



# *AMD Geode™ CS5535 Companion Device Data Book*

---

*February 2005*

Publication ID: 31506B

---

© 2005 Advanced Micro Devices, Inc. All rights reserved.

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

## **Contacts**

[www.amd.com](http://www.amd.com)

## **Trademarks**

AMD, the AMD Arrow logo, and combinations thereof, and Geode, GeodeLink, WebPAD, and Virtual System Architecture are trademarks of Advanced Micro Devices, Inc.

Microsoft is a registered trademark of Microsoft Corporation in the United States and/or other jurisdictions.

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

# Contents

---

<b>List of Figures</b> .....	<b>7</b>
<b>List of Tables</b> .....	<b>9</b>
<b>1.0 Overview</b> .....	<b>13</b>
1.1 General Description .....	13
1.2 Features .....	14
<b>2.0 Architecture Overview</b> .....	<b>17</b>
2.1 GeodeLink™ PCI South Bridge .....	17
2.2 GeodeLink™ Control Processor .....	17
2.3 ATA-5 Controller .....	18
2.4 Universal Serial Bus Controllers .....	18
2.5 Audio Codec 97 (AC97) Controller .....	18
2.6 Diverse Device .....	18
2.7 GeodeLink™ Interface Unit .....	20
2.8 Low Voltage Detect .....	20
2.9 Processor Support .....	21
<b>3.0 Signal Definitions</b> .....	<b>23</b>
3.1 Ball Assignments .....	25
3.2 Signal Descriptions .....	36
<b>4.0 Global Concepts and Features</b> .....	<b>55</b>
4.1 GeodeLink™ Architecture Overview .....	55
4.2 MSR Addressing .....	59
4.3 Typical GeodeLink™ Device .....	61
4.4 Embedded PCI Adapter .....	62
4.5 Clock Considerations .....	63
4.6 Reset Considerations .....	65
4.7 Memory and I/O Map Overview .....	67
4.8 Standard GeodeLink™ Device MSRs .....	74
4.9 Power Management .....	79
4.10 Component Revision ID .....	80

<b>5.0</b>	<b>Module Functional Descriptions</b>	<b>81</b>
5.1	GeodeLink™ Interface Unit	82
5.2	GeodeLink™ PCI South Bridge	84
5.3	AC97 Audio Codec Controller	89
5.4	ATA-5 Controller	99
5.5	Universal Serial Bus Controller	102
5.6	Diverse Integration Logic	104
5.7	Programmable Interval Timer	108
5.8	Programmable Interrupt Control	111
5.9	Direct Memory Access Module	119
5.10	Keyboard Emulation Logic	122
5.11	System Management Bus Controller	127
5.12	UART and IR Port	137
5.13	Low Pin Count Port	146
5.14	Real-Time Clock Features	157
5.15	General Purpose Input/Output	159
5.16	Multi-Function General Purpose Timer	166
5.17	Power Management Control	172
5.18	Flash Controller	183
5.19	GeodeLink™ Control Processor	191
5.20	TAP Controller	193
<b>6.0</b>	<b>Register Descriptions</b>	<b>199</b>
6.1	GeodeLink™ Interface Unit Register Descriptions	201
6.2	GeodeLink™ PCI South Bridge Register Descriptions	228
6.3	AC97 Audio Codec Controller Register Descriptions	245
6.4	ATA-5 Controller Register Descriptions	262
6.5	USB Controller Register Descriptions	276
6.6	Diverse Integration Logic Register Descriptions	312
6.7	Floppy Port Register Descriptions	339
6.8	Programmable Interval Timer Register Descriptions	341
6.9	Programmable Interrupt Controller Register Descriptions	349
6.10	Keyboard Emulation Logic Register Descriptions	364
6.11	System Management Bus Register Descriptions	370
6.12	UART and IR Port Register Descriptions	377
6.13	Direct Memory Access Register Descriptions	417
6.14	Low Pin Count Register Descriptions	435
6.15	Real-Time Clock Register Descriptions	438
6.16	GPIO Subsystem Register Descriptions	450
6.17	Multi-Function General Purpose Timer Register Descriptions	483
6.18	Power Management Controller Register Descriptions	494
6.19	Flash Controller Register Descriptions	519
6.20	GeodeLink™ Control Processor Register Descriptions	531

---

<b>7.0</b>	<b>Electrical Specifications</b>	<b>545</b>
7.1	General Specifications	545
7.2	DC Characteristics	547
7.3	AC Characteristics	550
7.4	Power Supply Sequence Requirements	571
7.5	Low Voltage Detect (LVD) Parameters	572
7.6	Skip Parameter	573
<b>8.0</b>	<b>Package Specifications</b>	<b>575</b>
<b>Appendix A</b>	<b>Support Documentation</b>	<b>577</b>
A.1	Order Information	577
A.2	Data Book Revision History	577



# List of Figures

Figure 1-1.	Block Diagram	13
Figure 2-1.	Typical WebPAD™ System Block Diagram	21
Figure 2-2.	Thin Client Application System Block Diagram	22
Figure 3-1.	Typical Signal Groups	23
Figure 3-2.	208-PBGA Ball Assignment Diagram	26
Figure 4-1.	Simplified GLIU with Generic GeodeLink™ Devices	55
Figure 4-2.	GeodeLink™ Architecture ASMI and Error Routing	57
Figure 4-3.	AMD Geode™ CS5535 Companion Device GeodeLink™ Architecture Topology	58
Figure 4-4.	Typical GeodeLink™ Device	61
Figure 4-5.	GeodeLink™ Device With Embedded PCI Bus	62
Figure 4-6.	Reset Logic	66
Figure 4-7.	Direct ASMI Behavioral Model	75
Figure 4-8.	In-Direct ASMI Behavioral Model	76
Figure 5-1.	Module Block Diagram	81
Figure 5-2.	GLPCI_SB Block Diagram	84
Figure 5-3.	ACC Block Diagram	89
Figure 5-4.	AC Link Slot Scheme	91
Figure 5-5.	AC Link Output Frame	91
Figure 5-6.	AC Link Input Frame	91
Figure 5-7.	ACC PRD Table Example	96
Figure 5-8.	USB Core Block Diagram	102
Figure 5-9.	USB Host Controller FSM	103
Figure 5-10.	Diverse Logic Block Diagram	104
Figure 5-11.	I/O Space LBAR - Fixed Target Size	105
Figure 5-12.	I/O Space LBAR - Variable Target Size	106
Figure 5-13.	Memory Space LBAR	106
Figure 5-14.	PIT Block Diagram	108
Figure 5-15.	PIT Counter Latch Command Format	109
Figure 5-16.	PIT Read-Back Command Format	110
Figure 5-17.	PIT Status Byte Format	110
Figure 5-18.	PIC Subsystem	111
Figure 5-19.	Cascading 8259As for LPIC	114
Figure 5-20.	PIC 8259A Block Diagram	115
Figure 5-21.	DMA Module Block Diagram	119
Figure 5-22.	KEL Block Diagram	122
Figure 5-23.	SMB Block Diagram	127
Figure 5-24.	SMB Bit Transfer	129
Figure 5-25.	SMB START and STOP Conditions	129
Figure 5-26.	SMB Data Transaction	130
Figure 5-27.	SMB Acknowledge Cycle	130
Figure 5-28.	SMB Complete Data Transaction	131
Figure 5-29.	SMB Byte Write	135
Figure 5-30.	SMB Page Write	135
Figure 5-31.	SMB Current Address Read	135
Figure 5-32.	SMB Random Read	136

Figure 5-33.	SMB Sequential Reads . . . . .	136
Figure 5-34.	UART/IR Overview Diagram . . . . .	137
Figure 5-35.	UART Serial Data Stream Format . . . . .	140
Figure 5-36.	Real Dongle Interface . . . . .	144
Figure 5-37.	Virtual Dongle Interface . . . . .	145
Figure 5-38.	LPC Block Diagram . . . . .	146
Figure 5-39.	Start of Cycle Timing Diagram . . . . .	150
Figure 5-40.	Abort Mechanism Timing Diagram . . . . .	150
Figure 5-41.	DMA Cycle Timing Diagram . . . . .	151
Figure 5-42.	Start Frame Waveform . . . . .	153
Figure 5-43.	IRQ Frame Waveform . . . . .	153
Figure 5-44.	Stop Frame Waveform . . . . .	153
Figure 5-45.	FWH Read Cycle . . . . .	154
Figure 5-46.	FWH Write Cycle . . . . .	155
Figure 5-47.	RTC Block Diagram . . . . .	157
Figure 5-48.	Recommended External Component Connections . . . . .	158
Figure 5-49.	GPIO Configuration . . . . .	162
Figure 5-50.	MFGPT Top Level Block Diagram . . . . .	166
Figure 5-51.	MFGPT Block Diagram . . . . .	169
Figure 5-52.	MFGPT Bit Reverse Logic . . . . .	170
Figure 5-53.	PMC Power Management Elements . . . . .	174
Figure 5-54.	PMC System Sleep Sequence . . . . .	175
Figure 5-55.	PMC System Wakeup Sequence . . . . .	176
Figure 5-56.	Flash Controller NAND Read Cycle . . . . .	184
Figure 5-57.	NOR Flash Basic Timing . . . . .	187
Figure 5-58.	NOR Flash with Wait States Timing . . . . .	188
Figure 5-59.	NAND Flash Command/Address Timing . . . . .	189
Figure 5-60.	NAND Data Timing with No Wait States and No Prefetch (for the first data read) . . . . .	190
Figure 5-61.	NAND Data Timing with Wait States . . . . .	190
Figure 5-62.	GLCP Block Diagram . . . . .	191
Figure 5-63.	TAP Controller, Boundary Scan Block Diagram . . . . .	193
Figure 5-64.	TAP Controller and Parallel Scan Mode . . . . .	196
Figure 5-65.	TAP Controller and TAPSCAN Mode . . . . .	197
Figure 6-1.	UART Register Bank Architecture . . . . .	383
Figure 6-2.	DMA Control Signals Routing . . . . .	402
Figure 7-1.	Clock Reference Definition . . . . .	550
Figure 7-2.	AC Reference Timing and Test Definition . . . . .	550
Figure 7-3.	Output Reference Timing and Test Definition . . . . .	551
Figure 7-4.	Input Reference Timing and Test Definition . . . . .	551
Figure 7-5.	IDE Data In Timing Non-UltraDMA . . . . .	556
Figure 7-6.	IDE UltraDMA Data Out Timing . . . . .	557
Figure 7-7.	IDE UltraDMA Data In Timing . . . . .	558
Figure 7-8.	USB Output Parameter Definition . . . . .	560
Figure 7-9.	USB Input Parameter Definition . . . . .	561
Figure 7-10.	USB Differential Input Sensitivity Range . . . . .	561
Figure 7-11.	LVD Electrical Parameters . . . . .	572
Figure 7-12.	Skip Electrical Parameters . . . . .	573
Figure 8-1.	208-Terminal PBGA Package (Body Size: 23x23x2.13 mm; Pitch: 1.27 mm) . . . . .	575



# List of Tables

Table 3-1.	Abbreviations/Definitions	25
Table 3-2.	Ball Assignments: Sorted by Ball Number	27
Table 3-3.	Ball Assignments: Sorted Alphabetically by Signal Name	31
Table 3-4.	Buffer Type Characteristics	33
Table 3-5.	Boot Options Selection	34
Table 3-6.	DIVIL_BALL_OPT (MSR 51400015h)	34
Table 3-7.	IDE and Flash Ball Multiplexing	40
Table 3-8.	GPIO Options	47
Table 3-9.	GPIOx Available Functions Descriptions	49
Table 4-1.	MSR Routing Conventions	59
Table 4-2.	MSR Addresses from AMD Geode™ GX Processor	60
Table 4-3.	Clock Sources and Clock Domains	63
Table 4-4.	Register Space Map Except Diverse Device	68
Table 4-5.	Diverse Device Register Space Map Except Legacy I/O	69
Table 4-6.	Legacy I/O: 000h-4FFh	70
Table 4-7.	GLD_MSR_CAP Bit Descriptions	74
Table 4-8.	Standard GLD_MSR_SMI Format	74
Table 4-9.	GLD_MSR_SMIs Summary	77
Table 4-10.	MSR Power Management Model	78
Table 4-11.	Sleep Driven PCI Signals	79
Table 4-12.	Sleep Driven IDE Signals	79
Table 5-1.	GLIU Port Connections	82
Table 5-2.	GLIU Descriptors Reserved for GeodeLink™ Devices	83
Table 5-3.	CIS Serial Bits Assignment and Descriptions	87
Table 5-4.	Audio Bus Master Descriptions	90
Table 5-5.	SLOTREQ to Output Slot Mapping	93
Table 5-6.	Physical Region Descriptor (PRD) Format	95
Table 5-7.	PCM Data format (Byte and Channel Ordering)	96
Table 5-8.	Physical Region Descriptor Format	100
Table 5-9.	UDMA/66 Signal Definitions	101
Table 5-10.	Special Cycle Decodes	107
Table 5-11.	8254 PIT Register Ports	109
Table 5-12.	IRQ Map - Primary and LPC	112
Table 5-13.	IRQ Map - Unrestricted Sources Y	113
Table 5-14.	IRQ Map - Unrestricted Sources Z	113
Table 5-15.	8259A PIC I/O Addresses and I/O Data	116
Table 5-16.	DMA Source Selection	121
Table 5-17.	KEL Mixed Environment	124
Table 5-18.	Comparison of SMB, I2C, and ACCESS.bus	128
Table 5-19.	SMB Native Registers Map	132
Table 5-20.	UART/IR Controller Native Register Bit Map	138
Table 5-21.	Real Dongle Interface Signals	144
Table 5-22.	Cycle Types Supported	147
Table 5-23.	Cycle Field Definitions: Target Memory, I/O, and DMA	148
Table 5-24.	Host Initiated Cycles	151

Table 5-25.	DMA Initiated Cycles	151
Table 5-26.	IRQ Data Frames	152
Table 5-27.	FWH Read Cycle	155
Table 5-28.	FWH Write Cycle	156
Table 5-29.	External Component Recommended Specifications	158
Table 5-30.	Effect on Feature Bit	161
Table 5-31.	16-Bit GPIO Control Register Example	161
Table 5-32.	MFGPT Prescaler Clock Divider	170
Table 5-33.	MFGPT Pulse Density Modulation Example	171
Table 5-34.	Supported ACPI Power Management States	172
Table 5-35.	PM Events and Functions	180
Table 5-36.	ECC Parity/Bit Address Relationship	185
Table 5-37.	NAND Flash External Timing Parameters	189
Table 5-38.	TAP Controller Instructions	194
Table 5-39.	TAP Controller Instruction Bits	195
Table 6-1.	Standard GeodeLink™ Device MSR Summary	201
Table 6-2.	P2D Descriptor MSR Summary	201
Table 6-3.	GLIU Specific MSR Summary	202
Table 6-4.	IOD Descriptor MSR Summary	203
Table 6-5.	Standard GeodeLink™ Device MSR Summary	228
Table 6-6.	GLPCI_SB Specific MSR Summary	228
Table 6-7.	PCI Configuration Registers	229
Table 6-8.	Standard GeodeLink™ Device MSR Summary	245
Table 6-9.	ACC Native Registers Summary	246
Table 6-10.	Standard GeodeLink™ Device MSR Summary	262
Table 6-11.	ATAC Specific MSR Summary	262
Table 6-12.	ATAC Native Registers Summary	263
Table 6-13.	Standard GeodeLink™ Device MSR Summary	276
Table 6-14.	USB Specific MSR Summary	277
Table 6-15.	USB Embedded PCI Configuration Registers Summary	277
Table 6-16.	USB Host Controller Native Registers Summary	277
Table 6-17.	Standard GeodeLink™ Device MSR Summary	312
Table 6-18.	DIVIL Specific MSR Summary	312
Table 6-19.	Floppy Port Specific MSR Summary	339
Table 6-20.	PIT Specific MSR Summary	341
Table 6-21.	PIT Native Registers Summary	341
Table 6-22.	PIC Specific MSR Summary	349
Table 6-23.	PIC Native Registers Summary	349
Table 6-24.	KEL Specific MSR Summary	364
Table 6-25.	KEL Native Registers Summary	364
Table 6-26.	KEL Legacy Registers Emulated Summary	364
Table 6-27.	SMB Native Registers Summary	370
Table 6-28.	UART/IR Controller Specific MSR Summary	377
Table 6-29.	UART/IR Controller Native Registers Summary	377
Table 6-30.	Secondary ID Encoding	381
Table 6-31.	Register Bank Summary	383
Table 6-32.	Bank 0 Register Bit Map	384
Table 6-33.	EIR Non-Extended Mode Interrupt Priorities	388
Table 6-34.	Bit Settings for Parity Control	391
Table 6-35.	Bank Select Encoding	392
Table 6-36.	Bank 1 Bit Map	398
Table 6-37.	Bits Cleared on Fallback	398
Table 6-38.	Baud Generator Divisor Settings	399
Table 6-39.	Bank 2 Bit Map	400
Table 6-40.	DMA Threshold Levels	402

Table 6-41.	Bank 3 Bit Map . . . . .	405
Table 6-42.	Bank 4 Bit Map . . . . .	406
Table 6-43.	Bank 5 Bit Map . . . . .	407
Table 6-44.	Bank 6 Bit Map . . . . .	408
Table 6-45.	Bank 7 Bit Map . . . . .	410
Table 6-46.	CEIR, Low Speed Demodulator (RXHSC = 0) (Frequency Ranges in kHz) . . . . .	411
Table 6-47.	Consumer IR High Speed Demodulator (RXHSC = 1) (Frequency Ranges in kHz) . . . . .	411
Table 6-48.	Sharp-IR Demodulator (Frequency Ranges in kHz) . . . . .	411
Table 6-49.	Modulation Carrier Pulse Width . . . . .	412
Table 6-50.	CEIR Carrier Frequency Encoding . . . . .	413
Table 6-51.	IR Receive Input Selection . . . . .	416
Table 6-52.	DMA Specific MSRs Summary . . . . .	417
Table 6-53.	DMA Native Registers Summary . . . . .	417
Table 6-54.	LPC Specific MSRs Summary . . . . .	435
Table 6-55.	RTC Specific MSRs Summary . . . . .	438
Table 6-56.	RTC Native Registers Summary . . . . .	438
Table 6-57.	GPIO Low/High Bank Feature Bit Registers Summary . . . . .	450
Table 6-58.	GPIO Input Conditioning Function Registers Summary . . . . .	452
Table 6-59.	GPIO Interrupt and PME Mapper Registers Summary . . . . .	453
Table 6-60.	Low Bank Atomic Register Map . . . . .	454
Table 6-61.	Low Bank Atomic Bit Descriptions . . . . .	454
Table 6-62.	High Bank Atomic Register Map Format . . . . .	455
Table 6-63.	High Bank Atomic Bit Descriptions Format . . . . .	455
Table 6-64.	MFGPT Specific MSRs Summary . . . . .	483
Table 6-65.	MFGPT Native Registers Summary . . . . .	483
Table 6-66.	PMC Specific MSRs Summary . . . . .	494
Table 6-67.	ACPI Registers Summary . . . . .	494
Table 6-68.	PM Support Registers Summary . . . . .	495
Table 6-69.	Flash Controller Specific MSRs Summary . . . . .	519
Table 6-70.	Flash Controller Native Registers Summary . . . . .	520
Table 6-71.	Standard GeodeLink™ Device MSRs Summary . . . . .	531
Table 6-72.	GLPC Specific MSRs Summary . . . . .	531
Table 6-73.	Clock Mapping / Operational Settings . . . . .	536
Table 7-1.	Electro Static Discharge (ESD) . . . . .	545
Table 7-2.	Absolute Maximum Ratings . . . . .	545
Table 7-3.	Recommended Operating Conditions . . . . .	546
Table 7-4.	Current Consumption . . . . .	546
Table 7-5.	DC Characteristics . . . . .	547
Table 7-6.	Other USB DC Parameters . . . . .	549
Table 7-7.	Clock Timing Parameters . . . . .	552
Table 7-8.	Reset and Test Timing Parameters . . . . .	553
Table 7-9.	PCI, SUSP#, SUSPA#, and RESET_OUT# Timing Parameters . . . . .	554
Table 7-10.	IDE Register, PIO, and Multiword DMA Timing Parameters . . . . .	555
Table 7-11.	IDE UltraDMA Data Out Timing Parameters . . . . .	557
Table 7-12.	IDE UltraDMA Data In Timing Parameters . . . . .	558
Table 7-13.	Flash Timing Parameters . . . . .	559
Table 7-14.	USB Timing Parameters . . . . .	560
Table 7-15.	SMB Timing Parameters . . . . .	562
Table 7-16.	AC97 Codec Timing Parameters . . . . .	563
Table 7-17.	LPC Timing Parameters . . . . .	564
Table 7-18.	Power Management and Processor Control Timing Parameters . . . . .	565
Table 7-19.	Miscellaneous Signals Except UART Timing Parameters . . . . .	566
Table 7-20.	UART Timing Parameters . . . . .	567
Table 7-21.	GPIO Signaling . . . . .	568
Table 7-22.	MFGPT Signaling . . . . .	569

---

Table 7-23.	JTAG Timing Parameters	570
Table 7-24.	LVD Parameters	572
Table 7-25.	Skip Parameters	573
Table A-1.	Revision History	577
Table A-2.	Edits to Current Revision	578



Audio is supported by an internal controller, designed to connect to multiple AC97 compatible codecs. An IR (infrared) port supports all popular IR communication protocols. The IR port is shared with one of two industry-standard serial ports that can reach speeds of 115.2 kbps. An LPC (low pin count) port is provided to facilitate connections to a SuperI/O should additional expansion, such as a floppy drive, be necessary, and/or to an LPC ROM for the system BIOS.

The hard disk controller is an ATA-5 compatible bus mastering IDE controller; includes support for two ATA-compliant devices on one channel. Two dual-port USBs (universal serial buses, USB specification v1.1 compliant) provide four ports with both low and full-speed capabilities for Plug & Play expansion for a variety of consumer peripheral devices such as a keyboard, mouse, printer, and digital camera. A battery-backed RTC (real-time clock) keeps track of time and provides calendar functions.

A suite of 82xx devices provide the legacy PC functionality required by most designs, including two PICs (programmable interrupt controllers), one PIT (programmable interval timer) with three channels, and DMA (direct memory access) functions. The Geode CS5535 companion device contains eight MFGPTs (multi-function general purpose timers) that can be used for a variety of functions. A number of GPIOs (general purpose input/outputs) are provided, and are assigned to system functions on power-up (i.e., LPC port); each of these may be reassigned and given different I/O characteristics such as debounce, edge-triggering, etc.

State-of-the-art power management features are attained with the division of the device into two internal power domains. The GPIOs and multi-function timers are distributed into each of the two domains to allow these to act as wakeup sources for the device. In addition to full ACPI (Advanced Configuration Power Interface) compliance and support of industry-standard Wakeup and Sleep modes, the device automatically disables clocks from internal blocks when they are not being used.

## 1.2 Features

### General Features

- Designed for use with AMD's Geode GX processor
- 208-Terminal PBGA (plastic ball grid array) package
- 3.3V I/O and 1.2V or 1.5V (nominal) Core operation
- Low power operation: 150 mW Typ in Working state
- Working and Standby power domains
- IEEE 1149.1 compliant TAP and boundary scan

### GeodeLink™ PCI Bridge (South Bridge)

- Provides a PCI interface for GeodeLink devices:
  - PCI specification v2.2 compliant
  - 32-Bit, 33/66 MHz operation
  - Transaction FIFOs (first in/first out)
  - Bus master or slave
  - Converts selected PCI configuration bus cycles to internal MSR (Model Specific Register) cycles
  - Capable of handling in-bound transactions immediately after reset - no setup
  - Mapping of PCI virtual configuration space to MSR space is done completely in Virtual System Architecture™ (VSA) code
  - Serialized processor control interface

### GeodeLink™ Control Processor

- SUSP#/SUPA# handshake with power management logic provides Sleep control of all GeodeLink devices
- System software debug support using built-in "logic analyzer" with:
  - 8192-bit capture memory
  - Capture memory can be organized wide or narrow
  - "Analyzer" can be connected to thousands of possible internal nodes
  - Synchronous operation with Geode GX processor GeodeLink Control Processor
  - JTAG interface and system bus interfaces
  - For debug use, able to conduct any GeodeLink transaction via the JTAG interface
  - Manufacturing test support

### ATA-5 Controller

- 66 MB per second IDE Controller in UDMA mode per the ATA-5 specification
- 3.3V interface
- Legacy and Enhanced PIO (Programmable I/O), MDMA (Multi DMA), and UDMA (Ultra DMA) modes
- One channel with two devices
- Multiplexed with Flash interface

### Flash Interface

- Multiplexed with IDE interface
- Connects to array of industry standard NAND Flash and/or NOR Flash
- NOR optional execute-in-place boot source
- NAND optional file system
- General purpose ISA bus slave-like devices supported with configurable chip selects
- Hardware support for SmartMedia type ECC (Error Correcting Code) calculation off loading software intensive algorithm

### USB Controllers 1 and 2

- Two independent host USB controllers; doubles the throughput of a single controller
- Each controller has two ports; total of four ports
- USB specification v1.1 compliant, with external crimp protection diodes
- OHCI (Open Host Controller Interface) specification v1.0 compliant
- Supports wakeup events
- Second generation proven core design
- Over-current and power control support
- GeodeLink master burst reads and writes

### Audio Codec 97 (AC97) Controller

- AC97 specification v2.1 compliant interface to multiple audio codecs: Serial In, Serial Out, Sync Out, Bit Clock In
- Legacy “PC Beep” support
- Eight-channel buffered GeodeLink mastering interface
- ASMI and IRQ support
- Multiple codec support
- Surround sound support

### Diverse Device

- 82xx Legacy Devices:
  - Two 8259A-equivalent PICs:
    - Shadow registers allow reading of internal registers
  - One 8254-equivalent PIT
  - Two 8237A-equivalent DMA controllers:
    - 8-bit DMA supported (only)
  - Serial Ports 1 and 2:
    - Port 1 is shared with an IR port
    - 16550A and 16450 software compatible
    - Shadow register support for write-only bit monitoring
    - UART data rates up to 115.2 kbps
- IR (Infrared) Communication Port:
  - Shared with Serial Port 1
  - 16550A and 16450 software compatible
  - Shadow register support for write-only bit monitoring
  - Consumer-IR (TV-Remote) mode
  - Data rate up to 115.2 kbps (SIR)
  - HP-SIR (same as SIR above)
  - Selectable internal or external modulation/demodulation (Sharp-IR)
  - ASK-IR option of SHARP-IR
  - DASK-IR option of SHARP-IR
  - Consumer Remote Control supports RC-5, RC-6, NEC, RCA, and RECS 80

- System Management Bus (SMB) Controller:
  - Compatible with Intel System Management Bus, Phillips I<sup>2</sup>C, and ACCESS.bus
  - Bus master and slave operation
- LPC (Low Pin Count) Port:
  - Based on Intel LPC Interface specification v1.0
  - Serial IRQ support
  - Serial DMA support (8-bit only)
  - Boot source typically off external LPC
  - Supports firmware hub protocol
  - External bus masters not supported
- General Purpose I/Os (GPIOs):
  - Programmable: In, Out, I/O, Open-Drain, Pull-Up/Down, and Invert
  - Parallel bit read and write
  - Individual bit access eliminates Read-Modify-Write cycles
  - Input Conditioning Functions (ICF):
    - Input debounce/filter
  - Input event counter
  - Input edge detect
- Multi-Function General Purpose Timers (MFGPTs):
  - Eight MFGPTs - two are multiplexed with GPIOs for external usage
  - Two MFGPTs are powered by Standby power and can be used as wakeups
  - Watchdog timer generates reset, IRQ, ASMI, or NMI
  - Pulse Width Modulation (PWM)
  - Pulse Density Modulation (PDM)
  - Blink
- Real-Time Clock (RTC) with CMOS RAM:
  - Battery backed-up century calendar in days, day of the week, date of month, months, years and century, with automatic leap-year adjustment
  - Battery backed-up time of day in seconds, minutes, and hours that allows a 12 or 24 hour format and adjustments for daylight savings time
  - BCD or binary format for time keeping
  - DS1287, MC146818, and PC87911 compatibility
  - Selective lock mechanisms for the RTC RAM
  - Real-time alarm
  - V<sub>BAT</sub> or V<sub>STANDBY</sub> power sources with automatic switching between them
  - 242 bytes of battery-backed CMOS RAM in two banks

**■ Power Management Controller:**

- ACPI (Advanced Configuration Power Interface) specification v2.0 compliant timer and register set
- Supports APM (Advanced Power Management) and Legacy PM
- PME (power management event) from GPIOs and/or on-chip sources
- Working, Sleep, and Standby states
- Wakeup circuits powered by Standby power rails while rest of component and system powered off
- Automatic clock-off gating reduces power to inactive blocks
- Flexible power supply controls including On/Off and Sleep button inputs
- Generic Sleep output controls
- ACPI-compliant four second fail-safe off
- Low-voltage detect function for battery-powered applications
- Suspend/Acknowledge handshake with Geode GX processor
- System over-temperature support
- Low Voltage Detect (LVD) provides Power On Reset (POR) as well as continuous voltage monitoring for automatic system reset on a low voltage condition

**GeodeLink™ Interface Unit**

- 64-Bit, 66 MHz operation
- Transparent to applications software and BIOS due to PCI VSM (virtual system module) implementation
- Non-blocking arbitration and routing of request and data packets
- Programmable routing descriptors
- Programmable use and activity monitors that generate optional ASMI (asynchronous system management interrupts) for legacy power management purposes
- Programmable SSMI (synchronous system management interrupt) generators for selected range of addresses; intended for virtual device emulation (future support, if needed)
- ATA-5 Controller, GLIU, and Diverse Device are the only SSMI sources



# Architecture Overview

# 2

The AMD Geode™ CS5535 companion device provides interfaces for all the common peripherals of an information appliance, plus offers expansion for additional needs, if required. Featuring a 33/66 MHz PCI interface to the AMD Geode GX processor, the Geode CS5535 companion device is internally connected using the GeodeLink™ packet architecture. This architecture supports multiple simultaneous transactions and is totally transparent to all application software. GeodeLink architecture related operations are managed via Model Specific Registers (MSRs) that are detailed in Section 4.1.6 "Address Spaces and MSRs" on page 58.

As shown in Figure 1-1 on page 13, the Geode CS5535 companion device is implemented with one GeodeLink Interface Unit (GLIU) that connects to the:

- GeodeLink PCI South Bridge
- GeodeLink Control Processor
- ATA-5 Controller (IDE Controller multiplexed with Flash Interface)
- Universal Serial Bus 1 Controller with Ports 1-1 and 1-2
- Universal Serial Bus 2 Controller with Ports 2-1 and 2-2
- Audio Codec 97 (AC97) Controller
- Diverse Device:
  - Legacy DMA, Timer, and Interrupt (82xx PC Legacy)
  - UARTs (2) and IR (1) Port (shared with UART1)
  - System Management Bus (SMB) Controller
  - Low Pin Count (LPC) Controller
  - General Purpose I/O (GPIO) with Input Conditioning Functions (ICF)
  - Multi-Function General Purpose Timers (MFGPTs)
  - Flash Interface (multiplexed with IDE interface)
- Real-Time Clock (RTC) with CMOS RAM
- Power Management Controller (PMC)

The Low Voltage Detect (LVD) circuit is not a GeodeLink Device, but is connected to the Power Management Controller for voltage monitoring support.

## 2.1 GeodeLink™ PCI South Bridge

The GeodeLink PCI South Bridge (GLPCI\_SB) provides a PCI interface for the Geode CS5535 companion device. It acts as a PCI master or slave in providing PCI transactions to and from the Geode CS5535 companion device and the PCI bus. A special serial interface to the Geode GX processor, the CPU Interface Serial (CIS), is provided that assists in the transfer of information between the Geode CS5535 companion device and the Geode GX processor.

The interface is compliant to PCI specification v2.2 and may operate at up to 66 MHz. Optional bus signals PERR#, SERR#, LOCK#, and CLKRUN are not implemented. Within a PCI burst, zero wait state operation is achieved. The PCI interface supports programmable IDSEL selection, and can handle inbound transactions immediately after system reset.

## 2.2 GeodeLink™ Control Processor

The GeodeLink Control Processor is responsible for debug support and monitors system clocks in support of PMC operations.

The GLCP interfaces with a JTAG compatible Test Access Port (TAP) Controller that is IEEE 1149.1 compliant. During debug, it can be used to pass GeodeLink packets to/from the GeodeLink Interface Unit (GLIU). It is also used to sup-

## 2.3 ATA-5 Controller

The Geode CS5535 companion device integrates a fully-buffered, ATA-5 compliant (UDMA/66) IDE interface. The IDE interface supports one channel, which in turn supports two devices that can operate in PIO modes 1 to 4, MDMA modes 0 to 2, or UDMA/66 modes 0 to 4.

This interface is shared with the Flash interface, using the same balls. The interface usage, immediately after reset, is defined by the boot options selected (see Table 3-5 "Boot Options Selection" on page 34). After reset, the interface may be dynamically altered using the Ball Options MSR (see Table 3-6 on page 34).

The IDE interface provides a variety of features to optimize system performance, including 32-bit disk access, post write buffers, bus master, MDMA, look-ahead read buffer, and prefetch mechanism.

The IDE interface timing is completely programmable. Timing control covers the command active and recover pulse widths, and command block register accesses. The IDE data transfer speed for each device on each channel can be independently programmed allowing high-speed IDE peripherals to coexist on the same channel as older, compatible devices.

The Geode CS5535 companion device also provides a software accessible buffered reset signal to the IDE drive. The IDE\_RST# signal is driven low during reset to the Geode CS5535 companion device and can be driven low or high as needed for device-power-off conditions.

## 2.4 Universal Serial Bus Controllers

The Geode CS5535 companion device provides four USB ports, controlled by two independent controllers (USBC1 and USBC2) for 2x enhanced system performance. There are two ports associated with each controller for a total of four. Separate power and ground pins for the transceivers are provided to accommodate various system designs and provide superior noise immunity. Each pair of ports has an associated power control line, and there is a common over-current sense line for all four ports, compatible with the National Semiconductor LM3526 dual-port USB power switch. The controllers are OpenHost Controller Interface (OHCI) v1.0 compliant, and the ports adhere to the USB v1.1 specification, with external crimp protection diodes.

## 2.5 Audio Codec 97 (AC97) Controller

The audio subsection of the Geode CS5535 companion device consists of three 32-bit stereo-buffered bus masters (two for output, one for input) and five 16-bit mono-buffered bus masters (three for output, two for input), whose function is to transport audio data between system memory and external AC97 codecs.

This arrangement is capable of producing multi-channel 5.1 surround sound (left, center, right, left rear, right rear, and low frequency effects).

The codec interface is AC97 v2.1 compliant and contains Serial In (x2), Serial Out, Sync Out, and Bit Clock allowing support for any AC97 codec with Sample Rate Conversion (SRC). Additionally, the interface supports the industry-standard 16-bit pulse code modulated (PCM) format.

## 2.6 Diverse Device

A suite of 82xx devices provide all the legacy PC functionality required by most designs, including two programmable interrupt controllers (PICs), one Programmable Interval Timer (PIT) with three channels, and Direct Memory Access (DMA) functions. The Geode CS5535 companion device contains eight Multi-Function General Purpose Timers (MFGPTs) that can be used for a variety of functions. A number of General-purpose Input/Outputs (GPIOs) are provided, and are assigned to system functions on power-up (i.e., LPC port); each of these may be reassigned and given different I/O characteristics such as debounce, edge-triggering, and so forth.

The Diverse Integration Logic (DIVIL) holds the devices together and provides overall control and management via MSRs.

### 2.6.1 Legacy DMA

The Geode CS5535 companion device DMA controller consists of two cascaded 8237A-type DMA controllers that together support four 8-bit channels. The DMA controller is used to provide high speed transfers between internal chip sources. It has full 32-bit address range support via high-page registers. An internal mapper allows routing of any of seven internal DMA sources to the four 8-bit DMA channels.

### 2.6.2 Programmable Interval Timer, Legacy Timer

The Programmable Interval Timer (PIT) generates programmable time intervals from the divided clock of an external clock input. The PIT is an 8254-style timer that contains three 16-bit independently programmable counters. A 14.318 MHz external clock signal (from a crystal oscillator or a clock chip) is divided by 12 to generate 1.19 MHz for the clocking reference of all three counters.

### 2.6.3 Programmable Interrupt Controller, Legacy Interrupt

The Programmable Interrupt Controller (PIC) consists of two 8259A-compatible programmable interrupt controllers connected in cascade mode through interrupt number two. Request mask capability and edge-level controls are provided for each of the 15 channels along with a 15-level priority controller. The PIC devices support all x86 modes of operation except Special Fully Nested mode.

An IRQ mapper takes up to 62 discrete interrupt request (IRQ) inputs and maps or masks them to the 15 PIC inputs and to one ASMI (asynchronous system management interrupt). All 62 inputs are individually maskable and status readable.

In addition to the above 8259A features, there are shadow registers to obtain the values of legacy 8259A registers that have not been historically readable.

### 2.6.4 Keyboard Emulation Logic, Legacy Support Interface

The PS2 Keyboard Emulation Logic (KEL) provides a virtual 8042 keyboard controller interface that may be used to map non-legacy keyboard and mouse sources to this traditional interface. Flexible keyboard emulation logic allows PS2 keyboard emulation traditionally used for USB legacy keyboard emulation. For example, USB sources may be 'connected' to this interface via SMM (System Management Mode) software. It also allows mixed environments with one LPC legacy device and one USB device.

### 2.6.5 Universal Asynchronous Receiver Transmitter and IR Port

Two Universal Asynchronous Receiver Transmitters (UARTs) provide a system interface to the industry standard serial interface consisting of the basic transmit and receive signals. One of the UARTs can be coupled with infrared logic and be connected to an infrared sensor.

The UARTs are both 16550A and 16450 software-compatible and contain shadow register support for write-only bit monitoring. The ports have data rates up to 115.2 kbps.

Serial port 1 can be configured as an infrared communications port that supports Sharp-IR, Consumer-IR, and HP-SIR as well as many popular consumer remote-control protocols.

### 2.6.6 System Management Bus Controller

The System Management Bus (SMB) Controller provides a system interface to the industry standard SMB. The SMB allows easy interfacing to a wide range of low-cost memory and I/O devices, including: EEPROMs, SRAMs, timers, ADC, DAC, clock chips, and peripheral drivers. These lines are shared with two GPIOs and must be configured as SMB ports in order for this interface to be functional.

The SMB is a two-wire synchronous serial interface compatible with the System Management Bus physical layer. The SMB Controller can be configured as a bus master or slave, and can maintain bidirectional communication with both multiple master and slave devices. As a slave device, the SMB Controller may issue a request to become the bus master.

### 2.6.7 Low Pin Count Port

This port provides a system interface to the industry standard Low Pin Count (LPC) bus. The controller can convert an internal Local bus memory or I/O cycle to an external LPC cycle. It receives serial IRQs from the LPC and converts them to parallel form so they can be routed to the IRQ mapper. Lastly, it interacts with Legacy DMA logic to perform DMA between on-chip or off-chip DMA devices.

The LPC interface is based on the Intel's Low Pin Count (LPC) Interface specification v1.0. In addition to the required signals/pins specified in the Intel specification, it also supports two optional signals:

- LPC\_DRQ# - LPC DMA Request
- LPC\_SERIRQ - LPC Serial encoded IRQ

The LPC interface supports memory, I/O, DMA, and Intel's firmware hub interfaces.

### 2.6.8 General Purpose I/Os with Input Conditioning Functions (ICF)

There are 32 GPIOs in the Geode CS5535 companion device, 28 are externally available, that offer a variety of user-selectable configurations including accessing auxiliary functions within the chip, and input conditioning such as debounce and edge detect. Register access is configured in such a way as to avoid Read-Modify-Write operations; each GPIO may be directly and independently configured.

Several groups of GPIOs are multiplexed between the LPC Controller, the SMB Controller, access to the UARTs and MFGPTs, and power management controls including system power and Sleep buttons. Six of the GPIOs are in the Standby power domain, giving them increased versatility as wakeup event sources when only Standby power is applied.

A GPIO interrupt and power management event (PME) mapper can map any subset of GPIOs to the PICs (eight interrupts available) or Power Management Subsystem (eight events available).

Versatile input filtering is available for each GPIO input. Each preliminary input is optionally connected to a digital filter circuit that is optionally followed by an event counter. Lastly followed by an edge detector that together provide eight different ICFs (input conditioning functions), plus an auto-sense feature for determining the initial condition of the pin.

### 2.6.9 Multi-Function General Purpose Timers

This module contains eight Multi-Function General Purpose Timers (MFGPTs), six are in the normal  $V_{DD}$  Working power domain, while the other two are in the Standby power domain. The timers are very versatile and can be configured to provide a Watchdog timer (trigger GPIO output, interrupt or reset), perform Pulse Width Modulation (PWM) or Pulse Density Modulation (PDM), create Blink (low frequency pulse for LED), generate GPIO outputs, or act as general purpose timers.

Each MFGPT operates independently and has the following features:

- 32 kHz or 14.318 MHz clock selectable by software (applies to MFGPT0 to MFGPT5, in Working power domain, only).
- MFGPT6 and MFGPT7, in Standby power domain, use 32 kHz clock.
- Programmable input clock prescaler divisor to divide input clock by  $2^i$ , where  $i = 0$  to 15.
- Provide outputs for generating reset (limited to MFGPT0 to MFGPT5), IRQs, NMI, and ASMLs (indirectly through PICs).

### 2.6.10 Flash Interface

The Geode CS5535 companion device has a Flash device interface that supports popular NOR Flash and inexpensive NAND Flash devices. This interface is shared with the IDE interface (ATA-5 Controller (ATAC)), using the same balls. NOR or NAND Flash may co-exist with IDE devices using PIO (Programmed I/O) mode. The 8-bit interface supports up to four "lanes" of byte-wide Flash devices through use of four independent chip selects, and allows for booting from the array. Hardware support is present for SmartMedia-type ECC (Error Correction Code) calculations, off-loading software from having to support this task.

All four independent chip selects may be used as general purpose chip selects to support other ISA-like slave devices. Up to 1 kB of address space (without external latches) may be supported using these signals.

### 2.6.11 Real-Time Clock with CMOS RAM

The Geode CS5535 companion device maintains a real-time clock for system use. The clock is powered by an external battery and so continues to keep accurate time even when system power is removed. The clock can be set to make automatic Daylight Savings Time changes in the spring and fall without user intervention. There are sepa-

rate registers for seconds, minutes, hours, days (both day of the week and day of the month), months, and years. Alarms can be set for any time within the range of these registers, which have a 100-year capability. The clock uses an external 32 kHz oscillator or crystal as the timing element.

The same battery that keeps the clock continuously powered also provides power to a block of 242 bytes of CMOS RAM, used for storing non-volatile system parameters.

### 2.6.12 Power Management Controller

The Geode CS5535 companion device has state-of-the-art power management capabilities designed into every module. Independent clock controls automatically turn clocks off to sections of the chip that are not being used, saving considerable power. In addition, the chip supports full Sleep and Wakeup states with multiple methods of inducement. A suite of external signals support power management of devices on the system board. Legacy Power Management (PM), Advanced Power Management (APM), and Advanced Configuration and Power Interface (ACPI) techniques and requirements are supported. The GPIO subsystem can be configured to transmit any of several wakeup events into the system.

The Geode CS5535 companion device is divided into two main power domains: Working and Standby, plus circuits such as the real-time clock and CMOS RAM that are battery-backed. Most of the Geode CS5535 companion device is in the Working power domain, except for GPIO[31:24] and MFGPT[7:6]. This allows these devices to be used for wakeup events or output controls.

## 2.7 GeodeLink™ Interface Unit

The GeodeLink Interface Unit (GLIU) makes up the internal bus derived from the GeodeLink architecture. It has eight ports, one of which is dedicated to itself, leaving seven for use by internal GeodeLink devices. Figure 1-1 "Block Diagram" on page 13 shows this device as the central element of the architecture, though its presence is basically transparent to the end user.

## 2.8 Low Voltage Detect

The Low Voltage Detect (LVD) circuit monitors: Standby I/O voltage, Standby Core voltage, and Working Core voltage. Working I/O voltage is not monitored and is assumed to track with Working Core voltage. The LVD monitors these voltages to provide Working and Standby power-good signals (resets) for the respective working and standby power domains. Additionally, the PMC monitors the working power-good signal to shut-down and/or re-start the system as appropriate.

## 2.9 Processor Support

As previously stated, the Geode CS5535 companion device was designed to interface with the Geode GX processor. Figure 2-1 and Figure 2-2 on page 22 show typical

block diagrams for a WebPAD™ system and thin client application based on the Geode GX processor and Geode CS5535 companion device.

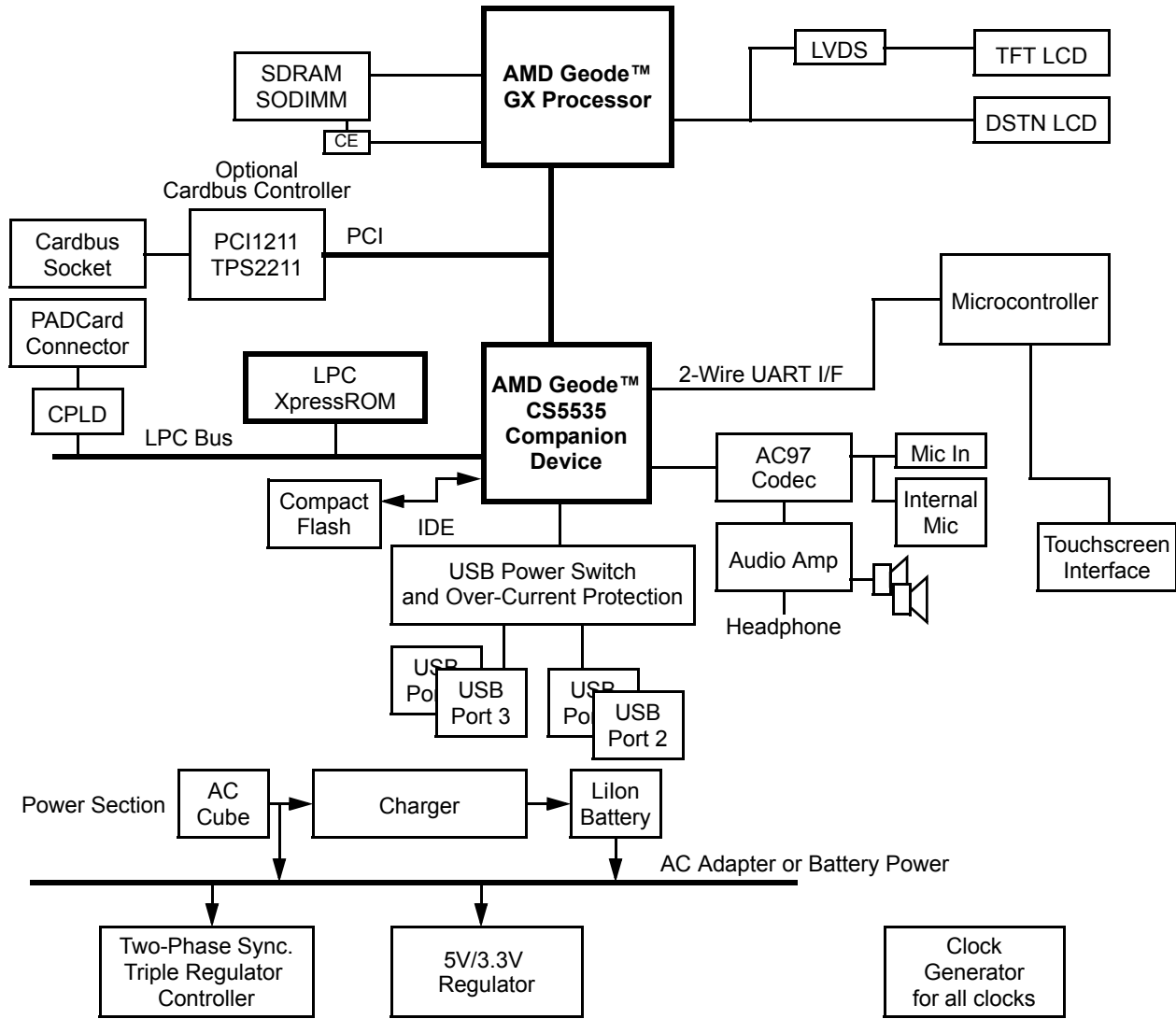


Figure 2-1. Typical WebPAD™ System Block Diagram

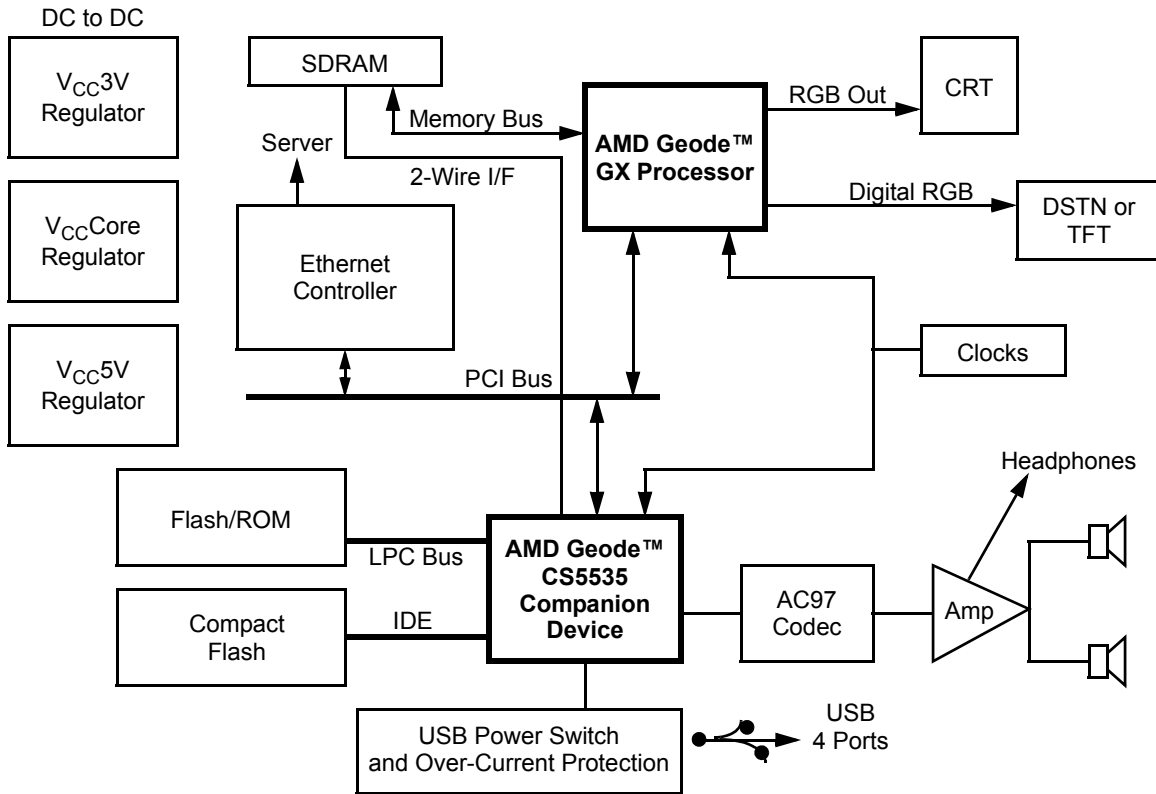


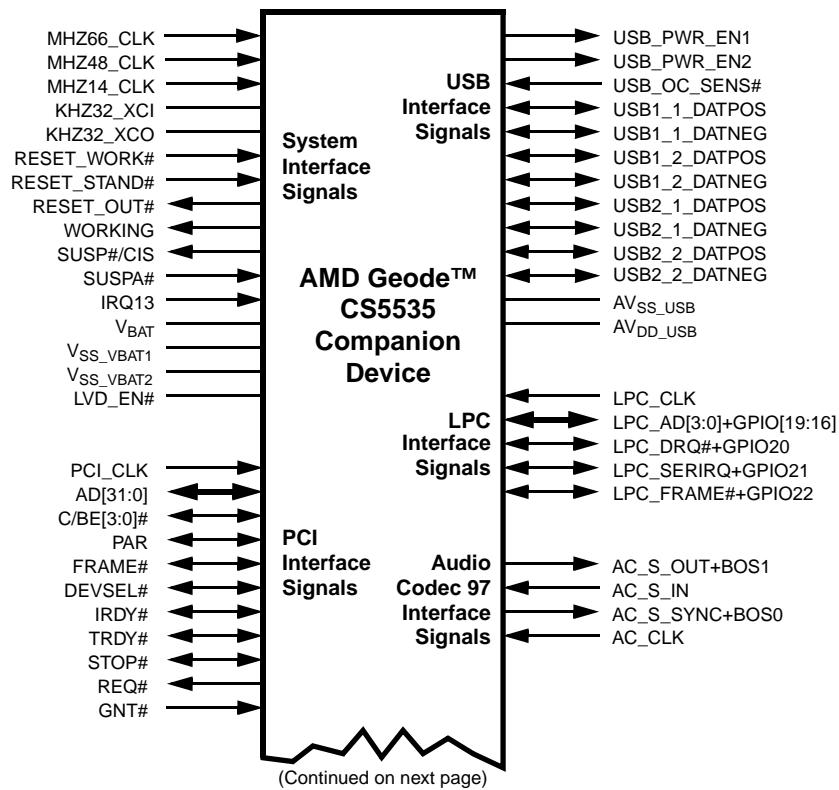
Figure 2-2. Thin Client Application System Block Diagram

# Signal Definitions 3

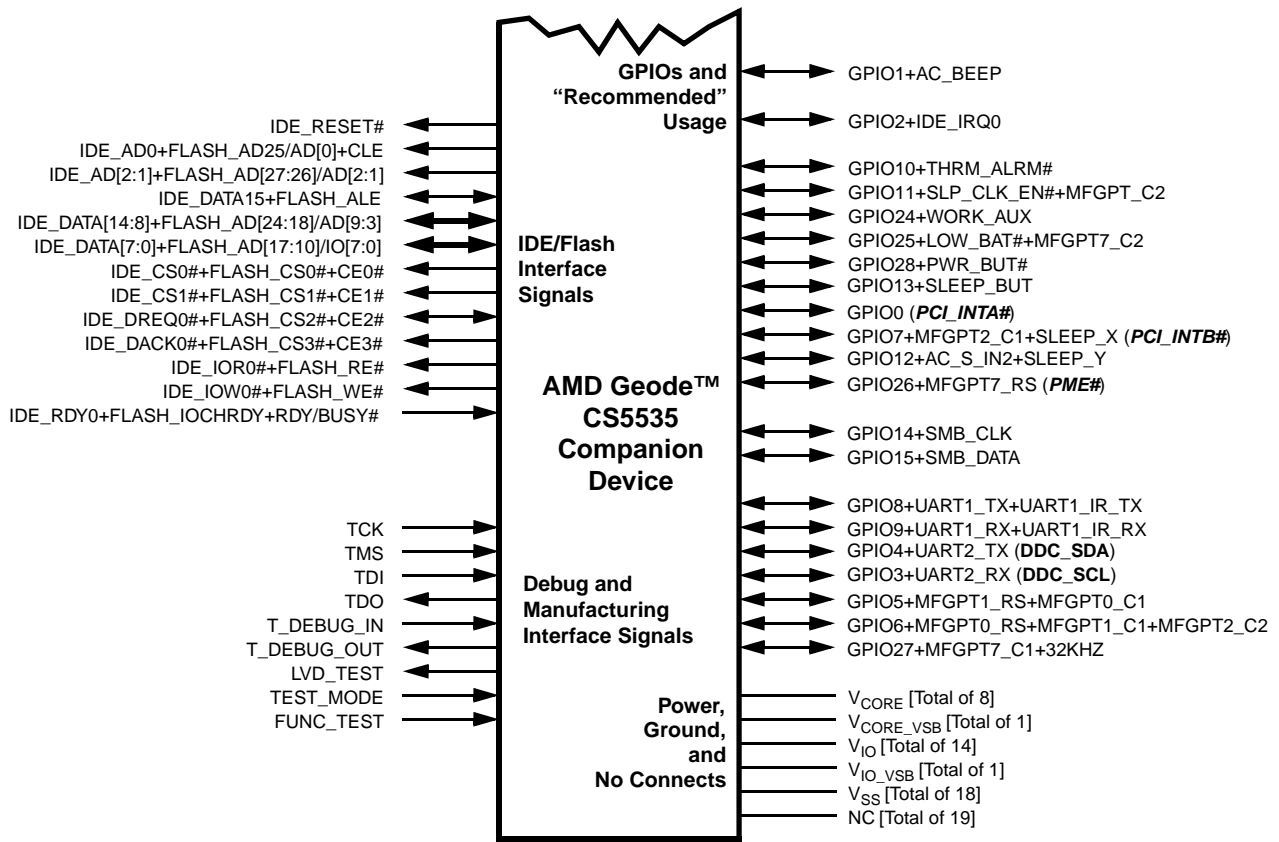
This section defines the signals and describes the external interface of the AMD Geode™ CS5535 companion device. Signal multiplexing has been utilized to a high degree. For example, the IDE and Flash interfaces are multiplexed on the same balls. Configuration is dependent upon the boot options selected (see Table 3-5 "Boot Options Selection" on page 34). If Flash is selected, the user has the option of using NOR and/or NAND Flash devices.

The GPIOs are configurable (e.g., any GPIO input can be mapped to an interrupt, ASMI, or PME). Figure 3-1 shows the signals organized in typical functional groups - not all possible multiplexing is shown.

Where signals are multiplexed, the primary signal name is listed first and is separated by a plus sign (+). A slash (/) in a signal name means that the function is always enabled and available (i.e., time multiplexed).



**Figure 3-1. Typical Signal Groups**



**Note:** *Bold-italicized* signal names in parenthesis denote a “recommended” use for a specific GPIO. See Table 3-8 “GPIO Options” on page 47 for additional details.

**Figure 3-1. Typical Signal Groups (Continued)**



### 3.1 Ball Assignments

As illustrated in Figure 3-1 on page 23, the Geode CS5535 companion device is configurable. Boot options and register programming are used to set various modes of operation and specific signals on specific balls.

This section describes the ball assignments and interface options:

- Figure 3-2 "208-PBGA Ball Assignment Diagram" on page 26:
  - Top view looking through package.
- Table 3-2 "Ball Assignments: Sorted by Ball Number" on page 27:
  - Primary signal name is listed first.
  - Includes a column labeled *Buffer Type*. See Section 3.1.1 "Buffer Types" on page 33 for details.

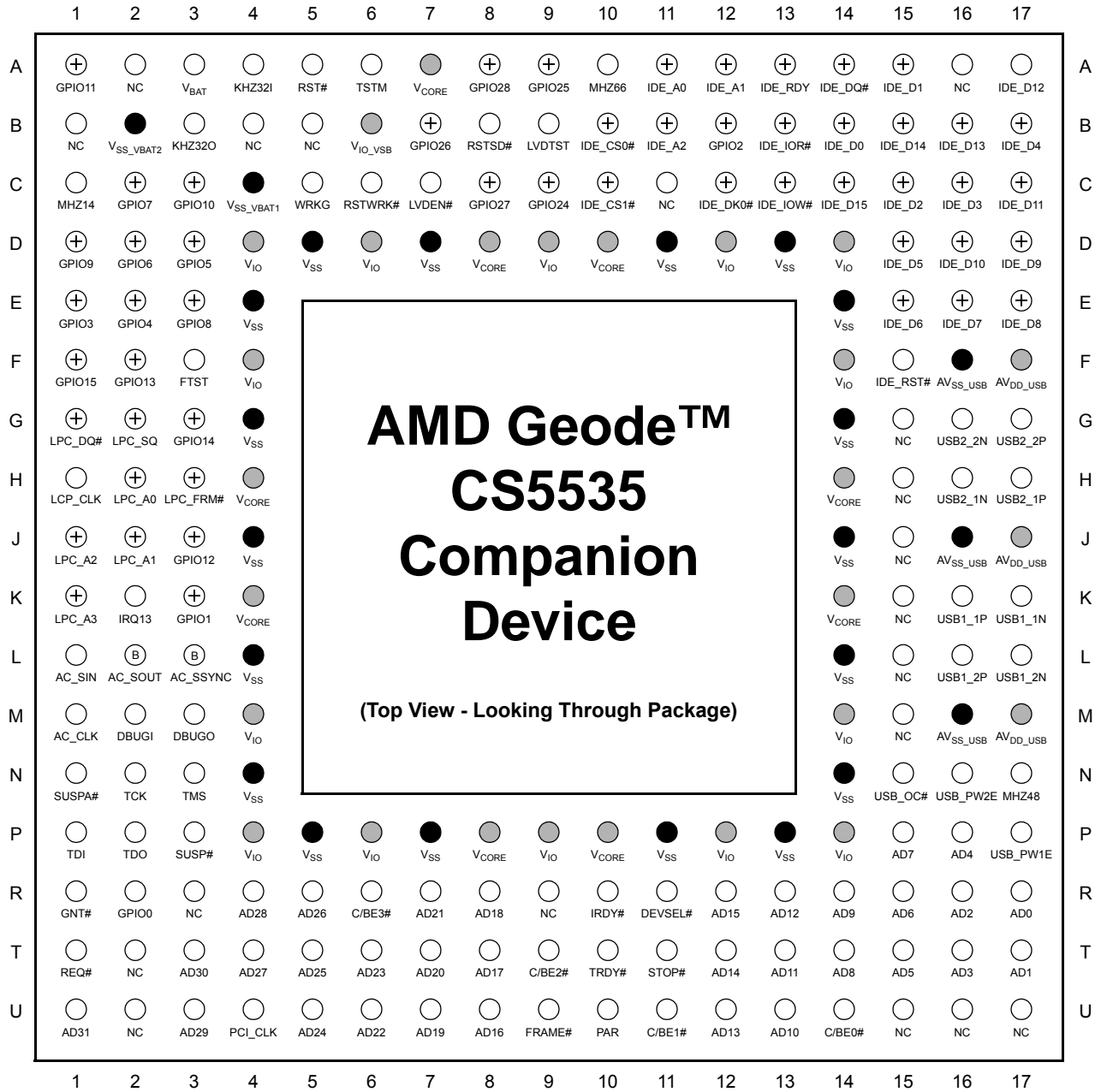
- Includes a column labeled *Configuration* with references to:
  - BOS[1:0] - See Section 3.1.2 "Boot Options" on page 34.
  - Ball Opt MSR - See Section 3.1.3 "Ball Options" on page 34.
  - IN\_AUX1, OUT\_AUX1, and OUT\_AUX2 - See Section 3.2.8 "GPIOs" on page 47.

- Table 3-3 "Ball Assignments: Sorted Alphabetically by Signal Name" on page 31:
  - Quick-reference list, sorted alphabetically with primary signal listed first.

The tables in this section use several abbreviations. Table 3-1 lists the mnemonics and their meanings.

**Table 3-1. Abbreviations/Definitions**

Mnemonic	Definition
A	Analog
AV <sub>SS</sub>	Analog Ground Connection
AV <sub>DD</sub>	Analog Power Connection
GND	Ground
I	Input
I/O	Bidirectional
O	Output
OD	Open-drain
Ball Opt MSR	Model Specific Register Ball Options: A register is used to configure balls with multiple functions. Refer to Section 3.1.3 "Ball Options" on page 34 for further details.
PD	Pull-down resistor
PWR	Power
PU	Pull-up resistor
TS	TRI-STATE
V <sub>CORE</sub>	1.2V or 1.5V Core Power Working Connection
V <sub>CORE_VSB</sub>	1.2V or 1.5V Core Power Standby Connection
V <sub>IO</sub>	3.3V I/O Power Working Connection
V <sub>IO_VSB</sub>	3.3V I/O Power Standby Connection
V <sub>SS</sub>	Ground
#	The "#" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present after the signal name, the signal is asserted when at a high voltage level.
/	A "/" in a signal name indicates the function is always enabled (i.e., time multiplexed - available when needed).
+	A "+" in a signal name indicates the function is available on the ball, but that either strapping options or register programming is required to select the desired function.



**Note:** Signal names have been abbreviated in this figure due to space constraints.

- = GND terminal
- = PWR terminal
- ⊕ = Multiplexed signal
- ⊕ = BOS (Boot Option Select)

**Figure 3-2. 208-PBGA Ball Assignment Diagram**

Table 3-2. Ball Assignments: Sorted by Ball Number

Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration
A1	GPIO11	I/O	Q7	
	SLP_CLK_EN#	O		OUT_AUX1
	MFGPT1_C2	O		OUT_AUX2
A2	NC	---	---	
A3	V <sub>BAT</sub>	Wire	Bare_Wire_BP	
A4	KHZ32_XCI	Wire	Bare_Wire	
A5	RESET_OUT#	O	Q7	
A6	TEST_MODE	Wire	Bare_Wire	
A7	V <sub>CORE_VSB</sub>	PWR	---	
A8	GPIO28	I/O	Q7	
	PWR_BUT#	I		IN_AUX1
A9	GPIO25	I/O	Q7	
	LOW_BAT#	I		IN_AUX1
	MFGPT7_C2	O		OUT_AUX2
A10	MHZ66_CLK	I	Q3	
A11	IDE_AD0	O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD25/AD0	O		BOS[1:0] = 10
	FLASH_CLE	O		
A12	IDE_AD1	O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD26/AD1	O		BOS[1:0] = 10
A13	IDE_RDY0	I	IDE	BOS[1:0] = 00 or 11
	FLASH_IOCHRDY	I		BOS[1:0] = 10
	FLASH_RDY/BUSY#	I		
A14	IDE_DREQ0	I	IDE	BOS[1:0] = 00 or 11
	FLASH_CS2#	O		BOS[1:0] = 10
	FLASH_CE2#	O		
A15	IDE_DATA1	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD11/IO1	I/O		BOS[1:0] = 10
A16	NC	---	---	
A17	IDE_DATA12	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD22/AD7	O		BOS[1:0] = 10
B1	NC	---	---	
B2	V <sub>SS_VBAT2</sub>	AGND	AVSS_USB	
B3	KHZ32_XCO	Wire	Bare_Wire	
B4	NC	---	---	
B5	NC	---	---	
B6	V <sub>IO_VSB</sub>	PWR	---	
B7	GPIO26	I/O	Q7	
	MFGPT7_RS	I		IN_AUX1
B8	RESET_STAND#	I	Bare_Wire	
B9	LVD_TEST	Wire (O)	Bare_Wire	
B10	IDE_CS0#	O	IDE	BOS[1:0] = 00 or 11
	FLASH_CS0#	O		BOS[1:0] = 10
	FLASH_CE0#	O		
B11	IDE_AD2	O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD27/AD2	O		BOS[1:0] = 10
B12	GPIO2	I/O	IDE	
	IDE_IRQ0	I		IN_AUX1
B13	IDE_IOR0#	O	IDE	BOS[1:0] = 00 or 11
	FLASH_RE#	O		BOS[1:0] = 10
B14	IDE_DATA0	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD10/IO0	I/O		BOS[1:0] = 10
B15	IDE_DATA14	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD24/AD9	O		BOS[1:0] = 10
B16	IDE_DATA13	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD23/AD8	O		BOS[1:0] = 10
B17	IDE_DATA4	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD14/IO4	I/O		BOS[1:0] = 10
C1	MHZ14_CLK	I	Q7	
C2	GPIO7	I/O	PCI	
	MFGPT2_C1	O		OUT_AUX1
	SLEEP_X	O		OUT_AUX2
C3	GPIO10	I/O	Q7	
	THRM_ALRM#	I		IN_AUX1
C4	V <sub>SS_VBAT1</sub>	AGND	AVSS_USB	
C5	WORKING	O	SMB	
C6	RESET_WORK#	I	Q7	
C7	LVD_EN#	Wire	Bare_Wire	
C8	GPIO27	I/O	Q7	
	MFGPT7_C1	O		OUT_AUX1
	32KHZ	O		OUT_AUX2
C9	GPIO24	I/O	SMB	
	WORK_AUX	O		OUT_AUX1
C10	IDE_CS1#	O	IDE	BOS[1:0] = 00 or 11
	FLASH_CS1#	O		BOS[1:0] = 10
	FLASH_CE1#	O		
C11	NC	---	---	
C12	IDE_DACK0#	O	IDE	BOS[1:0] = 00 or 11
	FLASH_CS3#	O		BOS[1:0] = 10
	FLASH_CE3#	O		
C13	IDE_IOW0#	O	IDE	BOS[1:0] = 00 or 11
	FLASH_WE#	O		BOS[1:0] = 10
C14	IDE_DATA15	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_ALE	O		BOS[1:0] = 10
C15	IDE_DATA2	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD12/IO2	I/O		BOS[1:0] = 10
C16	IDE_DATA3	I/O	IDE	BOS[1:0] = 00 or 11
	FLASH_AD13/IO3	I/O		BOS[1:0] = 10

Table 3-2. Ball Assignments: Sorted by Ball Number (Continued)

Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration	Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration	
C17	IDE_DATA11	I/O	IDE	BOS[1:0] = 00 or 11	F1	GPIO15	I/O	SMB	IN_AUX1 and OUT_AUX1	
	FLASH_AD21/AD6	O		BOS[1:0] = 10		SMB_DATA	I/O			
D1	GPIO9	I/O	Q7	IN_AUX1	F2	GPIO13	I/O	Q7	IN_AUX1	
	UART1_RX	I				SLEEP_BUT	I			
	UART1_IR_RX	I								
D2	GPIO6	I/O	Q7	IN_AUX1 OUT_AUX1 OUT_AUX2	F3	FUNC_TEST	I	SMB		
	MFGPT0_RS	I								
	MFGPT1_C1	O								
	MFGPT2_C2	O								
D3	GPIO5	I/O	Q7	IN_AUX1 OUT_AUX1	F4	V <sub>IO</sub>	PWR	---		
	MFGPT1_RS	I								
	MFGPT0_C1	O								
D4	V <sub>IO</sub>	PWR	---		F14	V <sub>IO</sub>	PWR	---		
D5	V <sub>SS</sub>	GND	---		F15	IDE_RESET#	O	IDE		
D6	V <sub>IO</sub>	PWR	---		F16	AV <sub>SS_USB</sub>	AGND	AV <sub>SS_USB</sub>		
D7	V <sub>SS</sub>	GND	---		F17	AV <sub>DD_USB</sub>	APWR	AV <sub>DD_USB</sub>		
D8	V <sub>CORE</sub>	PWR	---		G1	LPC_DRQ#	I	PCI	Ball Opt MSR [6,4] = 1,1	
D9	V <sub>IO</sub>	PWR	---			GPIO20	I/O			Ball Opt MSR [6] = 0
D10	V <sub>CORE</sub>	PWR	---		G2	LPC_SERIRQ	I/O	PCI	Ball Opt MSR [6,5] = 1,1	
D11	V <sub>SS</sub>	GND	---			GPIO21	I/O			Ball Opt MSR [6] = 0
D12	V <sub>IO</sub>	PWR	---			MFGPT2_RS	I			IN_AUX1
D13	V <sub>SS</sub>	GND	---		G3	GPIO14	I/O	SMB	IN_AUX1 and OUT_AUX1	
D14	V <sub>IO</sub>	PWR	---			SMB_CLK	I/O			
D15	IDE_DATA5	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	G4	V <sub>SS</sub>	GND	---		
		FLASH_AD15/IO5			I/O					
D16	IDE_DATA10	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	G14	V <sub>SS</sub>	GND	---		
	FLASH_AD20/AD5	O								
D17	IDE_DATA9	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	G15	NC	---	---		
	FLASH_AD19/AD4	O								
E1	GPIO3	I/O	SMB	IN_AUX1	G16	USB2_2_DATNEG	I/O	Bare_Wire		
	UART2_RX	I								
E2	GPIO4	I/O	SMB	OUT_AUX1	G17	USB2_2_DATPOS	I/O	Bare_Wire		
	UART2_TX	O								
E3	GPIO8	I/O	Q7	OUT_AUX1 OUT_AUX2	H1	LPC_CLK	I	Q7		
	UART1_TX	O								
	UART1_IR_TX	O								
E4	V <sub>SS</sub>	GND	---		H2	LPC_AD0	I/O	PCI	Ball Opt MSR [6] = 1	
E14	V <sub>SS</sub>	GND	---		GPIO16	I/O			Ball Opt MSR [6] = 0	
E15	IDE_DATA6	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	H3	LPC_FRAME#	O	PCI	Ball Opt MSR [6] = 1	
	FLASH_AD16/IO6	I/O								
E16	IDE_DATA7	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	H4	V <sub>CORE</sub>	PWR	---		
	FLASH_AD17/IO7	I/O								
E17	IDE_DATA8	I/O	IDE	BOS[1:0] = 00 or 11 BOS[1:0] = 10	H14	V <sub>CORE</sub>	PWR	---		
	FLASH_AD18/AD3	O								
					H15	NC	---	---		
					H16	USB2_1_DATNEG	I/O	Bare_Wire		
					H17	USB2_1_DATPOS	I/O	Bare_Wire		
					J1	LPC_AD2	I/O	PCI	Ball Opt MSR [6] = 1	
						GPIO18	I/O			Ball Opt MSR [6] = 0
					J2	LPC_AD1	I/O	PCI	Ball Opt MSR [6] = 1	
						GPIO17	I/O			Ball Opt MSR [6] = 0
					J3	GPIO12	I/O	Q7	IN_AUX1 OUT_AUX2	
						AC_S_IN2	I			
						SLEEP_Y	O			
J4	V <sub>SS</sub>	GND	---		J14	V <sub>SS</sub>	GND	---		

Table 3-2. Ball Assignments: Sorted by Ball Number (Continued)

Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration
J15	NC	---	---	
J16	AV <sub>SS_USB</sub>	AGND	AVSS_USB	
J17	AV <sub>DD_USB</sub>	APWR	AVDD_USB	
K1	LPC_AD3	I/O	PCI	Ball Opt MSR [6] = 1
	GPIO19	I/O		Ball Opt MSR [6] = 0
K2	IRQ13	I	Q7	
K3	GPIO1	I/O	Q7	Default
	AC_BEEP	O		OUT_AUX1
	MFGPT0_C2	O		OUT_AUX2
K4	V <sub>CORE</sub>	PWR	---	
K14	V <sub>CORE</sub>	PWR	---	
K15	NC	---	---	
K16	USB1_1_DATPOS	I/O	Bare_Wire	
K17	USB1_1_DATNEG	I/O	Bare_Wire	
L1	AC_S_IN	I	Q7	
L2	AC_S_OUT	O	Q7	
	BOS1	I		
L3	AC_S_SYNC	O	Q7	
	BOS0	I		
L4	V <sub>SS</sub>	GND	---	
L14	V <sub>SS</sub>	GND	---	
L15	NC	---	---	
L16	USB1_2_DATPOS	I/O	Bare_Wire	
L17	USB1_2_DATNEG	I/O	Bare_Wire	
M1	AC_CLK	I	Q7	
M2	T_DEBUG_IN	I	Q5	
M3	T_DEBUG_OUT	O	Q5	
M4	V <sub>IO</sub>	PWR	---	
M14	V <sub>IO</sub>	PWR	---	
M15	NC	---	---	
M16	AV <sub>SS_USB</sub>	AGND	AVSS_USB	
M17	AV <sub>DD_USB</sub>	APWR	AVDD_USB	
N1	SUSPA#	I	Q7	
N2	TCK	I	Q7	
N3	TMS	I	Q7	
N4	VSS	GND	---	
N14	VSS	GND	---	
N15	USB_OC_SENS#	I	Q7	
N16	USB_PWR_EN2	O	Q7	
N17	MHZ48_CLK	I	Q3	
P1	TDI	I	Q5	
P2	TDO	O, TS	Q5	
P3	SUSP#	O	Q3	
	CIS	O		
P4	V <sub>IO</sub>	PWR	---	
P5	V <sub>SS</sub>	GND	---	
P6	V <sub>IO</sub>	PWR	---	
P7	V <sub>SS</sub>	GND	---	
P8	V <sub>CORE</sub>	PWR	---	
P9	V <sub>IO</sub>	PWR	---	
P10	V <sub>CORE</sub>	PWR	---	
P11	V <sub>SS</sub>	GND	---	
P12	V <sub>IO</sub>	PWR	---	
P13	V <sub>SS</sub>	GND	---	
P14	V <sub>IO</sub>	PWR	---	
P15	AD7	I/O	PCI	
P16	AD4	I/O	PCI	
P17	USB_PWR_EN1	O	Q7	
R1	GNT#	I	PCI	
R2	GPIO0	I/O	PCI	
R3	NC	---	---	
R4	AD28	I/O	PCI	
R5	AD26	I/O	PCI	
R6	C/BE3#	I/O	PCI	
R7	AD21	I/O	PCI	
R8	AD18	I/O	PCI	
R9	NC	---	---	
R10	IRDY#	I/O	PCI	
R11	DEVSEL#	I/O	PCI	
R12	AD15	I/O	PCI	
R13	AD12	I/O	PCI	
R14	AD9	I/O	PCI	
R15	AD6	I/O	PCI	
R16	AD2	I/O	PCI	
R17	AD0	I/O	PCI	
T1	REQ#	O	PCI	
T2	NC	---	---	
T3	AD30	I/O	PCI	
T4	AD27	I/O	PCI	
T5	AD25	I/O	PCI	
T6	AD23	I/O	PCI	
T7	AD20	I/O	PCI	
T8	AD17	I/O	PCI	
T9	C/BE2#	I/O	PCI	
T10	TRDY#	I/O	PCI	
T11	STOP#	I/O	PCI	
T12	AD14	I/O	PCI	
T13	AD11	I/O	PCI	
T14	AD8	I/O	PCI	
T15	AD5	I/O	PCI	
T16	AD3	I/O	PCI	
T17	AD1	I/O	PCI	
U1	AD31	I/O	PCI	
U2	NC	---	---	
U3	AD29	I/O	PCI	
U4	PCI_CLK	I	Q3	
U5	AD24	I/O	PCI	
U6	AD22	I/O	PCI	
U7	AD19	I/O	PCI	
U8	AD16	I/O	PCI	

**Table 3-2. Ball Assignments: Sorted by Ball Number (Continued)**

Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration
U9	FRAME#	I/O	PCI	
U10	PAR	I/O	PCI	
U11	C/BE1#	I/O	PCI	
U12	AD13	I/O	PCI	
U13	AD10	I/O	PCI	
U14	C/BE0#	I/O	PCI	

Ball No.	Signal Name (Note 1)	Type	Buffer Type (Note 2)	Configuration
U15	NC	---	---	
U16	NC	---	---	
U17	NC	---	---	

Note 1. The primary signal name is listed first.

Note 2. See Table 3-4 "Buffer Type Characteristics" on page 33 for buffer type definitions.

Table 3-3. Ball Assignments: Sorted Alphabetically by Signal Name

Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.
32KHZ	C8	FLASH_AD10/IO0	B14	GPIO17	J2	LVD_EN#	C7
AC_BEEP	K3	FLASH_AD11/IO1	A15	GPIO18	J1	LVD_TEST	B9
AC_CLK	M1	FLASH_AD12/IO2	C15	GPIO19	K1	MFGPT0_C1	D3
AC_S_IN	L1	FLASH_AD13/IO3	C16	GPIO20	G1	MFGPT0_C2	K3
AC_S_IN2	J3	FLASH_AD14/IO4	B17	GPIO21	G2	MFGPT0_RS	D2
AC_S_OUT	L2	FLASH_AD15/IO5	D15	GPIO22	H3	MFGPT1_C1	D2
AC_S_SYNC	L3	FLASH_AD16/IO6	E15	GPIO24	C9	MFGPT1_C2	A1
AD0	R17	FLASH_AD17/IO7	E16	GPIO25	A9	MFGPT1_RS	D3
AD1	T17	FLASH_AD18/AD3	E17	GPIO26	B7	MFGPT2_C1	C2
AD2	R16	FLASH_AD19/AD4	D17	GPIO27	C8	MFGPT2_C2	D2
AD3	T16	FLASH_AD20/AD5	D16	GPIO28	A8	MFGPT2_RS	G2
AD4	P16	FLASH_AD21/AD6	C17	IDE_AD0	A11	MFGPT7_C1	C8
AD5	T15	FLASH_AD22/AD7	A17	IDE_AD1	A12	MFGPT7_C2	A9
AD6	R15	FLASH_AD23/AD8	B16	IDE_AD2	B11	MFGPT7_RS	B7
AD7	P15	FLASH_AD24/AD9	B15	IDE_CS0#	B10	MHZ14_CLK	C1
AD8	T14	FLASH_AD25/AD0	A11	IDE_CS1#	C10	MHZ48_CLK	N17
AD9	R14	FLASH_AD26/AD1	A12	IDE_DACK0#	C12	MHZ66_CLK	A10
AD10	U13	FLASH_AD27/AD2	B11	IDE_DATA0	B14	NC (Total of 19)	A2, A16, B1, B4, B5, C11, G15, H15, J15, K15, L15, M15, R3, R9, T2, U2, U15, U16, U17
AD11	T13	FLASH_ALE	C14	IDE_DATA1	A15	PAR	U10
AD12	R13	FLASH_CE0#	B10	IDE_DATA2	C15	PCI_CLK	U4
AD13	U12	FLASH_CE1#	C10	IDE_DATA3	C16	PWR_BUT#	A8
AD14	T12	FLASH_CE2#	A14	IDE_DATA4	B17	REQ#	T1
AD15	R12	FLASH_CE3#	C12	IDE_DATA5	D15	RESET_OUT#	A5
AD16	U8	FLASH_CLE	A11	IDE_DATA6	E15	RESET_STAND#	B8
AD17	T8	FLASH_CS0#	B10	IDE_DATA7	E16	RESET_WORK#	C6
AD18	R8	FLASH_CS1#	C10	IDE_DATA8	E17	SLEEP_BUT	F2
AD19	U7	FLASH_CS2#	A14	IDE_DATA9	D17	SLEEP_X	C2
AD20	T7	FLASH_CS3#	C12	IDE_DATA10	D16	SLEEP_Y	J3
AD21	R7	FLASH_IOCHRDY	A13	IDE_DATA11	C17	SLP_CLK_EN#	A1
AD22	U6	FLASH_RDY/BUSY#	A13	IDE_DATA12	A17	SMB_CLK	G3
AD23	T6	FLASH_RE#	B13	IDE_DATA13	B16	SMB_DATA	F1
AD24	U5	FLASH_WE#	C13	IDE_DATA14	B15	STOP#	T11
AD25	T5	FRAME#	U9	IDE_DATA15	C14	SUSP#	P3
AD26	R5	FUNC_TEST	F3	IDE_DREQ0#	A14	SUSPA#	N1
AD27	T4	GNT#	R1	IDE_IOR0#	B13	T_DEBUG_IN	M2
AD28	R4	GPIO0	R2	IDE_IOW0#	C13	T_DEBUG_OUT	M3
AD29	U3	GPIO1	K3	IDE_IRQ0	B12	TCK	N2
AD30	T3	GPIO2	B12	IDE_RDY0	A13	TDI	P1
AD31	U1	GPIO3	E1	IDE_RESET#	F15		
AV <sub>DD</sub> _USB	F17, J17, M17	GPIO4	E2	IRDY#	R10		
AV <sub>SS</sub> _USB	F16, J16, M16	GPIO5	D3	IRQ13	K2		
BOS0	L3	GPIO6	D2	KHZ32_XCI	A4		
BOS1	L2	GPIO7	C2	KHZ32_XCO	B3		
C/BE0#	U14	GPIO8	E3	LOW_BAT#	A9		
C/BE1#	U11	GPIO9	D1	LPC_AD0	H2		
C/BE2#	T9	GPIO10	C3	LPC_AD1	J2		
C/BE3#	R6	GPIO11	A1	LPC_AD2	J1		
CIS	P3	GPIO12	J3	LPC_AD3	K1		
DEVSEL#	R11	GPIO13	F2	LPC_CLK	H1		
		GPIO14	G3	LPC_DRQ#	G1		
		GPIO15	F1	LPC_FRAME#	H3		
		GPIO16	H2	LPC_SERIRQ	G2		

**Table 3-3. Ball Assignments: Sorted Alphabetically by Signal Name (Continued)**

Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.	Signal Name	Ball No.
TDO	P2	V <sub>IO</sub> (Total of 14)	D4, D6, D9, D12, D14, F4, F14, M4, M14, P4, P6, P9, P12, P14	V <sub>SS</sub> (Total of 18)	D5, D7, D11, D13, E4, E14, G4, J4, L4, N4, N14, P5, P7, P11, P13	V <sub>SS_VBAT1</sub>	C4
TEST_MODE	A6					V <sub>SS_VBAT2</sub>	B2
THRM_ALRM#	C3					WORK_AUX	C9
TMS	N3					WORKING	C5
TRDY#	T10						
UART1_IR_RX	D1						
UART1_IR_TX	E3						
UART1_RX	D1						
UART1_TX	E3						
UART2_RX	E1						
UART2_TX	E2						
USB_OC_SENS#	N15						
USB_PWR_EN1	P17						
USB_PWR_EN2	N16						
USB1_1_DATNEG	K17						
USB1_1_DATPOS	K16						
USB1_2_DATNEG	L17						
USB1_2_DATPOS	L16						
USB2_1_DATNEG	H16						
USB2_1_DATPOS	H17						
USB2_2_DATNEG	G16						
USB2_2_DATPOS	G17						
VBAT	A3						
V <sub>CORE</sub> (Total of 8)	D8, D10, H4, H14, K4, K14, P8, P10						
V <sub>CORE_VSB</sub>	A7	V <sub>IO_VSB</sub>	B6				



### 3.1.1 Buffer Types

Table 3-2 "Ball Assignments: Sorted by Ball Number" on page 27 includes a column labeled "Buffer Type". The details of each buffer type listed in this column are given in Table 3-4. The column headings in Table 3-4 are identified as follows:

**TS:** Indicates whether the buffer may be put into the TRI-STATE mode. Note some pins that have buffer types that allow TRI-STATE may never actually enter the TRI-STATE mode in practice, since they may be inputs or provide other signals that are always driven. To determine if a particular signal can be put in the TRI-STATE mode, consult the individual signal descriptions in Section 3.2 "Signal Descriptions" on page 36.

**OD:** Indicates if the buffer is open-drain, or not. Open-drain outputs may be wire ORed together and require a discrete pull-up resistor to operate properly.

**5VT:** Indicates if the buffer is 5-volt tolerant, or not. If it is 5-volt tolerant, then 5 volt TTL signals may be safely applied to this pin.

**Backdrive Protected:** Indicates that the buffer may have active signals applied even when the Geode CS5535 companion device is powered down.

**PU/PD:** Indicates if an internal, programmable pull-up or pull-down resistor may be present.

**Current High/Low (mA):** This column gives the current source/sink capacities when the voltage at the pin is high, and low. The high and low values are separated by a "/" and values given are in milli-amps (mA).

**Rise/Fall @ Load:** This column indicates the rise and fall times for the different buffer types at the load capacitance indicated. These measurements are given in two ways: rise/fall time between the 20%-80% voltage levels, or, the rate of change the buffer is capable of, in volts-per-nano-second (V/ns). See Section 7.3 "AC Characteristics" on page 550 for details.

Note the presence of several "wire" type buffers in this table. Signals identified as one of the wire-types are not driven by a buffer, hence no rise/fall time or other measurements are given; these are marked "NA" in Table 3-4. The wire-type connection indicates a direct connection to internal circuits such as power, ground, and analog signals.

**Table 3-4. Buffer Type Characteristics**

Name	TS	OD	5VT	Backdrive Protected	PU/PD	Current High/Low (mA)	Rise/Fall @ Load
Q3	X				X	24/24	3 ns @ 50 pF
Q5	X				X	24/24	5 ns @ 50 pF
Q7	X				X	24/24	7 ns @ 50 pF
PCI	X					0.5/1.5	1-4 V/ns @ 10 pF
IDE	X					16/16	1.25 V/ns @ 40 pF
SMB		X	X	X			Rise: 1 $\mu$ s @ 400 pF Fall: 300 ns @ 400 pF
Bare_Wire	NA	NA			NA	NA	NA
Bare_Wire_BP	NA	NA		X	NA	NA	NA
AVDD_USB	NA	NA			NA	NA	NA
AVSS_USB	NA	NA			NA	NA	NA

### 3.1.2 Boot Options

Two balls on the device, L2 and L3, the Boot Options Select balls (BOS[1:0]), serve to specify the location of the boot device as the system undergoes a full reset. Since boot devices may reside in Flash or on an IDE device, the IDE/Flash interface is necessarily selected as operating in one of the two modes by the Boot Options. After Reset, the function of these interfaces may be changed with the Ball Options MSR (see Section 3.1.3 "Ball Options"). Both these balls are multiplexed with other functions (as identified in Section 3.2.7 "Audio Codec 97 Interface" on page 45) and function as BOS[1:0] only when RESET\_OUT# is asserted. Table 3-5 indicates how these two balls should be configured to select the desired boot device. Both balls contain an internal pull-up, active only during reset, so if a

ball is required to be high during this time, it may be left unconnected. If a ball is desired to be low during reset, a pull-down (i.e., not a hard tie to ground) should be added. During reset, both balls' output drivers are in the TRI-STATE mode.

### 3.1.3 Ball Options

Table 3-6 shows the Ball Options MSR (DIVIL MSR 51400015h), through which the function of certain groups of multiplexed balls may be dynamically changed after the reset period ends. Specifically, the functions LPC/GPIO and IDE/Flash groups are selected, and certain individual balls, as specified in the MSR, are controlled.

**Table 3-5. Boot Options Selection**

BOS1 (Ball L2)	BOS0 (Ball L3)	Description
0	0	<b>Boot from Memory Device on the LPC Bus.</b> IDE pins come up connected to IDE Controller (see Section 3.2.3 "IDE/Flash Interface Signals" on page 40 and Table 3-6 "DIVIL_BALL_OPT (MSR 51400015h)").
0	1	<b>Reserved.</b>
1	0	<b>Boot from NOR Flash on the IDE Bus.</b> IDE pins come up connected to Flash Controller (see Section 3.2.3 "IDE/Flash Interface Signals" on page 40 and Table 3-6 "DIVIL_BALL_OPT (MSR 51400015h)"). NOR Flash, ROM, or other random access devices must be connected to "FLASH_CS_3".
1	1	<b>Boot from Firmware Hub on the LPC Bus.</b> IDE pins come up connected to IDE Controller (see Section 3.2.3 "IDE/Flash Interface Signals" on page 40 and Table 3-6 "DIVIL_BALL_OPT (MSR 51400015h)").

**Table 3-6. DIVIL\_BALL\_OPT (MSR 51400015h)**

Bit	Name	Description
31:12	RSVD	<b>Reserved.</b> Reads always return 0. Writes have no effect; by convention, always write 0.
11:10	SEC_BOOT_LOC	<b>Secondary Boot Location.</b> Determines which chip select asserts for addresses in the range F00F0000h to F00F3FFFh. Defaults to the same value as boot option: 00: LPC ROM. 01: Reserved . 10: Flash. 11: FirmWare Hub.
9:8	BOOT_OP_LATCHED (RO)	<b>Latched Value of Boot Option (Read Only).</b> For values, see Table 3-5 "Boot Options Selection".
7	RSVD	<b>Reserved.</b> Reads return value written. By convention, always write 0. Defaults low.

Table 3-6. DIVIL BALL\_OPT (MSR 51400015h) (Continued)

Bit	Name	Description
6	PIN_OPT_LALL	<p><b>All LPC Pin Option Selection.</b></p> <p>0: All LPC pins become GPIOs including LPC_DRQ# and LPC_SERIRQ.  Ball H3 functions as GPIO22  Ball H2 functions as GPIO16  Ball J2 functions as GPIO17  Ball J1 functions as GPIO18  Ball K1 functions as GPIO19  Ball G1 functions as GPIO20  Ball G2 functions as GPIO21</p> <p>1: All LPC pins are controlled by the LPC controller except LPC_DRQ# and LPC_SERIRQ use are determined by bits [5:4]. <b>(Default)</b>  Ball H3 functions as LPC_FRAME#  Ball H2 functions as LPC_AD0  Ball J2 functions as LPC_AD1  Ball J1 functions as LPC_AD2  Ball K1 functions as LPC_AD3</p> <p>When this bit is low, there is an implied high for the LPC_DISABLE_MEM and LPC_DISABLE_IO bits in DIVIL_LEG_IO (DIVIL MSR 51400014h[25:24]).</p>
5	PIN_OPT_LIRQ	<p><b>LPC_SERIRQ or GPIO21 Pin Option Selection.</b></p> <p>0: Ball G2 is GPIO21.  1: Ball G2 functions as LPC_SERIRQ. <b>(Default)</b></p>
4	PIN_OPT_LDRQ	<p><b>LPC_DRQ# or GPIO20 Pin Option Selection.</b></p> <p>0: Ball G1 is GPIO20.  1: Ball G2 functions as LPC_DRQ#. <b>(Default)</b></p>
3:2	PRI_BOOT_LOC [1:0]	<p><b>Primary Boot Location.</b> Determines which chip select asserts for addresses at or above F0000000h, except those in the range specified by SEC_BOOT_LOC (bits [11:10]). Defaults to the same value as boot option.</p> <p>00: LPC ROM.  01: Reserved .  10: Flash.  11: FirmWare Hub.</p>
1	RSVD	<b>Reserved.</b> Reads return value written. By convention, always write 0. Defaults low.
0	PIN_OPT_IDE	<p><b>IDE or Flash Controller Pin Function Selection.</b></p> <p>0: All IDE pins associated with Flash Controller. Default if BOS[1:0] = 10.  1: All IDE pins associated with IDE Controller. Default if BOS[1:0] = 00 or 11.</p> <p>IDE_IRQ0 is multiplexed with GPIO2; therefore, this bit has no affect with regards to programming IDE_IRQ0. See Table 3-5 "Boot Options Selection" for BOS[1:0] programming values.</p>

## 3.2 Signal Descriptions

Information in the tables that follow may have duplicate information in multiple tables. Multiple references all contain identical information.

### 3.2.1 System Interface Signals

Signal Name	Ball No.	Type	Description
MHZ66_CLK	A10	I	<b>66 MHz Clock.</b> This is the main system clock. It is also used by the IDE interface.
MHZ48_CLK	N17	I	<b>USB Clock.</b> The 48 MHz clock for the UARTs and SMB Controller.
MHZ14_CLK	C1	I	<b>14.31818 MHz Timer Clock.</b> The input clock for power management functions and the Programmable Interval Timer (PIT).
KHZ32_XCI	A4	Wire	<p><b>32 kHz Input.</b> This input is used for the real-time clock (RTC), GPIOs, MFGPTs, and power management functions.</p> <p>This input may come from either an external oscillator or one side of a 32.768 kHz crystal. If an external oscillator is used, it should be powered by <math>V_{IO\_VSB}</math>.</p> <p>External oscillators often do not power up cleanly, causing the Standby logic to fail to reset properly. If used, it is required that RESET_STAND# be used to reset the Standby logic after the external oscillator has stabilized. This can be as long as one second.</p>
KHZ32_XCO	B3	Wire	<b>32 kHz Input 2.</b> This input is to be connected to the other side of the crystal oscillator connected to KHZ32_XCI, if used. Leave open (not connected) if an oscillator (not a crystal) is connected to KHZ32_XCI.
RESET_WORK#	C6	I	<p><b>Reset Working Power Domain.</b> This signal, when asserted, is the master reset for all Geode CS5535 companion device interfaces that are in the Working power domain. See Section 4.9.1 "Power Domains" on page 80 for a description of the Working power domain.</p> <p>RESET_WORK# must be asserted for at least 10 ns in order to be properly recognized.</p> <p>If LVD_EN# is enabled (tied low) use of this input is not required. See the LVD_EN# signal description for further details.</p>
RESET_STAND#	B8	I	<p><b>Reset Standby Power Domain.</b> This signal, when asserted, is the master reset for all Geode CS5535 companion device interfaces that are in the Standby power domain. See Section 4.9.1 "Power Domains" on page 80 for a description of the Standby power domain.</p> <p>To ensure the Skip feature is disabled, PWR_BUT# must be high before RESET_STAND# is de-asserted. See PWR_BUT# description in Table 3-9 "GPIOx Available Functions Descriptions" starting on page 49 for details on the Skip feature."</p> <p>If LVD_EN# is enabled (tied low) use of this input is not required. See the LVD_EN# discussion in this table.</p> <p>Tie directly to <math>V_{IO\_VSB}</math> if not used.</p>
RESET_OUT#	A5	O	<b>Reset Output.</b> This is the main system reset signal. RESET_OUT# is de-asserted synchronously with the low-to-high edge of PCI_CLK. The de-assertion is delayed from internal <i>reset</i> by up to 32 seconds, with an 8 ms default value, using a programmable counter driven by the 32 kHz clock. Note this counter default is established by RESET_STAND# and is not affected by RESET_WORK#. Therefore, the delay value may be changed and the system reset with the new value.
WORKING	C5	O	<b>Working State.</b> When high, indicates the chip is in the Working state. This signal is intended to be used to control power to off-chip devices in a system. Open-drain. External pull-up required.

## 3.2.1 System Interface Signals (Continued)

Signal Name	Ball No.	Type	Description
SUSP#	P3	O	<b>Suspend.</b> This signal goes low in response to events as determined by the power management logic. It requests the Geode GX processor to enter the Suspend state. This is the default state for this ball at reset. Not used in normal operation.
CIS		O	<b>CPU Interface Serial.</b> A 20-bit serial status word is output on this ball, synchronized to PCI_CLK. Data changes on the rising edge and is stable on the falling edge of PCI_CLK. This WORD is output whenever one of the internally-monitored signals changes states. See Section 5.2.14 "CPU Interface Serial (CIS)" on page 87 for details. Used in normal operation.
SUSPA#	N1	I	<b>Suspend Acknowledge.</b> This input signal is driven low by the Geode GX processor when it has successfully entered the Suspend state.
IRQ13	K2	I	<b>Interrupt Request Level 13.</b> Floating Point error. Connect directly to IRQ13 of the Geode GX processor.
V <sub>BAT</sub>	A3	Wire	<b>Real-Time Clock Battery Back-Up.</b> Battery voltage on this ball keeps the real-time clock and CMOS RAM circuits continuously powered.  If not used, tie to ground. 2.4-3.6V, typical 3.0V. 10 $\mu$ A max. 5 $\mu$ A typical.  This ball incorporates a reverse bias protection diode on-chip. There is no need for an external diode.
V <sub>SS_VBAT1</sub>	C4	AGND	<b>Real-Time Clock Battery Grounds 1 and 2</b>
V <sub>SS_VBAT2</sub>	B2		
LVD_EN#	C7	Wire	<b>Low Voltage Detect Enable.</b> LVD_EN# enables/disables the on-chip low voltage detection circuit. When disabled, the external subsystem must assert RESET_STAND# as Standby power is applied and must assert RESET_WORK# as Working power is applied. When LVD is enabled, use of these two resets are optional. Generally, RESET_STAND# would be tied high (not used) while RESET_WORK# would be tied to a reset output that is typically available from the power supply. However, a system could just have a simple regulator circuit and also tie RESET_WORK# high.  Tie to V <sub>SS</sub> to enable. Tie to V <sub>IO_VSB</sub> to disable.

## 3.2.2 PCI Interface Signals

Signal Name (Note 1)	Ball No.	Type	Description
PCI_CLK	U4	I	<b>PCI Clock.</b> 33 or 66 MHz.
AD[31:0]	U1, T3, U3, R4, T4, R5, T5, U5, T6, U6, R7, T7, U7, R8, T8, U8, R12, T12, U12, R13, T13, U13, R14, T14, P15, R15, T15, P16, T16, R16, T17, R17	I/O	<b>PCI Address/Data.</b> AD[31:0] is a physical address during the first clock of a PCI transaction; it is the data during subsequent clocks.  When the Geode CS5535 companion device is a PCI master, AD[31:0] are outputs during the address and write data phases, and are inputs during the read data phase of a transaction.  When the Geode CS5535 companion device is a PCI slave, AD[31:0] are inputs during the address and write data phases, and are outputs during the read data phase of a transaction.
C/BE[3:0]#	R6, T9, U11, U14	I/O	<b>PCI Bus Command and Byte Enables.</b> During the address phase of a PCI transaction, when FRAME# is active, C/BE[3:0]# define the bus command. During the data phase of a transaction, C/BE[3:0]# are the data byte enables.  C/BE[3:0]# are outputs when the Geode CS5535 companion device is a PCI master and inputs when it is a PCI slave.
PAR	U10	I/O	<b>PCI Parity.</b> PAR is the parity signal driven to maintain even parity across AD[31:0] and C/BE[3:0]#.  The Geode CS5535 companion device drives PAR one clock after the address phase and one clock after each completed data phase of write transactions as a PCI master. It also drives PAR one clock after each completed data phase of read transactions as a PCI slave.
FRAME#	U9	I/O	<b>PCI Cycle Frame.</b> FRAME# is asserted to indicate the start and duration of a transaction. It is de-asserted on the final data phase.  FRAME# is an input when the Geode CS5535 companion device is a PCI slave.  Normally connected to a 10k to 15k $\Omega$ external pull-up. This signal is in TRI-STATE mode after reset.
DEVSEL#	R11	I/O	<b>PCI Device Select.</b> DEVSEL# is asserted by a PCI slave, to indicate to a PCI master and subtractive decoder that it is the target of the current transaction.  As an input, DEVSEL# indicates a PCI slave has responded to the current address.  As an output, DEVSEL# is asserted one cycle after the assertion of FRAME# and remains asserted to the end of a transaction as the result of a positive decode. DEVSEL# is asserted four cycles after the assertion of FRAME# if DEVSEL# has not been asserted by another PCI device when the Geode CS5535 companion device is programmed to be the subtractive decode agent.  Normally connected to a 10k to 15k $\Omega$ external pull-up. This signal is in TRI-STATE mode after reset.

## 3.2.2 PCI Interface Signals (Continued)

Signal Name (Note 1)	Ball No.	Type	Description
IRDY#	R10	I/O	<p><b>PCI Initiator Ready.</b> IRDY# is driven by the master to indicate valid data on a write transaction, or that it is ready to receive data on a read transaction.</p> <p>When the Geode CS5535 companion device is a PCI slave, IRDY# is an input that can delay the beginning of a write transaction or the completion of a read transaction.</p> <p>Wait cycles are inserted until both IRDY# and TRDY# are asserted together.</p> <p>Normally connected to a 10k to15k <math>\Omega</math> external pull-up. This signal is in TRI-STATE mode after reset.</p>
TRDY#	T10	I/O	<p><b>PCI Target Ready.</b> TRDY# is asserted by a PCI slave to indicate it is ready to complete the current data transfer.</p> <p>TRDY# is an input that indicates a PCI slave has driven valid data on a read or a PCI slave is ready to accept data from the Geode CS5535 companion device on a write.</p> <p>TRDY# is an output that indicates the Geode CS5535 companion device has placed valid data on AD[31:0] during a read or is ready to accept the data from a PCI master on a write.</p> <p>Wait cycles are inserted until both IRDY# and TRDY# are asserted together.</p> <p>Normally connected to a 10k to15k <math>\Omega</math> external pull-up. This signal is in TRI-STATE mode after reset.</p>
STOP#	T11	I/O	<p><b>PCI Stop.</b> As an input, STOP# indicates that a PCI slave wants to terminate the current transfer. The transfer is either aborted or retried. STOP# is also used to end a burst.</p> <p>As an output, STOP# is asserted with TRDY# to indicate a target disconnect, or without TRDY# to indicate a target retry. The Geode CS5535 companion device asserts STOP# during any cache line crossings if in single transfer DMA mode or if busy.</p> <p>Normally connected to a 10k to15k <math>\Omega</math> external pull-up. This signal is in TRI-STATE mode after reset.</p>
REQ#	T1	O	<p><b>PCI Bus Request.</b> The Geode CS5535 companion device asserts REQ# to gain ownership of the PCI bus. The REQ# and GNT# signals are used to arbitrate for the PCI bus.</p> <p>REQ# should connect to the REQ2# of the Geode GX processor and function as the highest-priority PCI master.</p>
GNT#	R1	I	<p><b>PCI Bus Grant.</b> GNT# is asserted by an arbiter that indicates to the Geode CS5535 companion device that access to the PCI bus has been granted.</p> <p>GNT# should connect to GNT2# of the Geode GX processor and function as the highest-priority PCI master.</p>

Note 1. Use RESET\_OUT# for PCI reset.

For SMI, PME, INTA#, and INTB# functions, see Table 3-8 "GPIO Options" on page 47.

### 3.2.3 IDE/Flash Interface Signals

The IDE and Flash interface signals are multiplexed together on the same balls as shown in Table 3-7. Section 3.2.3.1 provides the names and functions of these signals when the interface is in the IDE mode and Section 3.2.3.2 when in Flash mode (NOR Flash/GPCS and NAND Flash modes).

**Table 3-7. IDE and Flash Ball Multiplexing**

Ball No.	IDE Mode	NOR Flash/GPCS Mode		NAND Flash Mode
		Address Phase	Data Phase	
B11, A12	IDE_AD[2:1]	FLASH_AD[27:26]	FLASH_AD[2:1]	---
A11	IDE_AD0	FLASH_AD25	FLASH_AD0	FLASH_CLE
B15, B16, A17, C17, D16, D17, E17	IDE_DATA[14:8]	FLASH_AD[24:18]	FLASH_AD[9:3]	---
E16, E15, D15, B17, C16, C15, A15, B14	IDE_DATA[7:0]	FLASH_AD[17:10]	FLASH_IO[7:0]	FLASH_IO[7:0]
C14	IDE_DATA15	FLASH_ALE		FLASH_ALE
B10	IDE_CS0#	FLASH_CS0#		FLASH_CE0#
C10	IDE_CS1#	FLASH_CS1#		FLASH_CE1#
B13	IDE_IOR0#	FLASH_RE#		FLASH_RE#
C13	IDE_IOW0#	FLASH_WE#		FLASH_WE#
A14 (Note 1)	IDE_DREQ0	FLASH_CS2#		FLASH_CE2#
C12	IDE_DACK0#	FLASH_CS3# (Boot Flash Chip Select)		FLASH_CE3#
A13	IDE_RDY0	FLASH_IOCHRDY		FLASH_RDY/BUSY#

Note 1. Ball A14 is the only ball that changes direction from IDE to Flash (input when in IDE mode, output when in Flash mode). If this interface is to be switched between IDE and Flash modes, then ball A14 needs an external pull-up to keep it high during IDE mode.

#### 3.2.3.1 IDE Interface Signals

Signal Name	Ball No.	Type	Description
IDE_IRQ0	B12	I	<b>IDE Interrupt Request Channel 0.</b> This signal is required for all IDE applications that use IDE DMA modes. It is available on GPIO2, which must be configured in the IN_AUX1 mode. If an IDE application will not use IDE DMA modes, or if the Flash interface will be used instead of the IDE interface, then this signal may be used as GPIO2.
IDE_RESET#	F15	O	<b>IDE Reset.</b> An internal reset that is the functional "OR" of inputs RESET_WORK# and RESET_STAND#. It may also be controlled directly via a MSR 51300010h (see Section 6.4.2.2 "Reset Decode (ATAC_RESET)" on page 269). This signal resets all the devices that are attached to the IDE interface.
IDE_AD[2:0]	B11, A12, A11	O	<b>IDE Address Bits.</b> These address bits are used to access a register or data port in a device on the IDE bus.



### 3.2.3.1 IDE Interface Signals (Continued)

Signal Name	Ball No.	Type	Description
IDE_DATA[15:0]	C14, B15, B16, A17, C17, D16, D17, E17, E16, E15, D15, B17, C16, C15, A15, B14	I/O	<b>IDE Data Lines.</b> IDE_DATA[15:0] transfers data to/from the IDE devices.
IDE_IOR0#	B13	O	<b>IDE I/O Read.</b> This output is asserted on read accesses to corresponding IDE port addresses. When in Ultra DMA/33 mode, this signal is redefined: <ul style="list-style-type: none"> <li>• IDE_HDMA_RDY. Host DMA Ready for Ultra DMA data-in bursts.</li> <li>• IDE_HDMA_DS. Host DMA Data Strobe for Ultra DMA data-out bursts.</li> </ul>
IDE_IOW0#	C13	O	<b>IDE I/O Write.</b> This output is asserted on write accesses to corresponding IDE port addresses. When in Ultra DMA/33 mode, this signal is redefined: <ul style="list-style-type: none"> <li>• IDE_STOP - Stop Ultra DMA data burst.</li> </ul>
IDE_CS0#	B10	O	<b>IDE Chip Select 0.</b> This chip select signal is used to select the Command Block registers in IDE Device 0.
IDE_CS1#	C10	O	<b>IDE Chip Select.</b> This chip select signal is used to select the Command Block registers in IDE Device 1.
IDE_DREQ0	A14	I	<b>DMA Request.</b> The DREQ input is used to request a DMA transfer from the Geode CS5535 companion device. The direction of the transfers are determined by the IDE_IOR0# and IDE_IOW0# signals. <b>Note:</b> Ball A14 is the only ball that changes direction from IDE to Flash (input when in IDE mode, output when in Flash mode). If this interface is to be switched between IDE and Flash modes, the IDE interface must only use PIO mode and ball A14 requires a pull-up resistor to keep FLASH_CS3# high when in IDE mode.
IDE_DACK0#	C12	O	<b>DMA Acknowledge.</b> The DACK# output acknowledges the IDE_DREQ0 request to initiate DMA transfers.
IDE_RDY0	A13	I	<b>I/O Ready.</b> When de-asserted, this signal extends the transfer cycle of any host register access when the device is not ready to respond to the data transfer request. When in Ultra DMA/33 mode, this signal is redefined: <ul style="list-style-type: none"> <li>• IDE_DDMA_DS - Device DMA Data Strobe for Ultra DMA data-in bursts.</li> <li>• IDE_DDMA_RDY - Device DMA Ready for Ultra DMA data-out bursts.</li> </ul>

## 3.2.3.2 Flash Controller Interface

Signal Name	Ball No.	Type	Description
<b>NOR Flash / GPCS Mode</b>			
FLASH_CS[3:0]#	C12, A14, C10, B10	O	<b>Chip Selects.</b> Combine with FLASH_RE#/WE# strobes to access external NOR Flash devices or some simple devices such as a UART. CS3# is dedicated to a boot Flash device.  <b>Note:</b> Ball A14 is the only ball that changes direction from IDE to Flash (input when in IDE mode, output when in Flash mode). If this interface is to be switched between IDE and Flash modes, the IDE interface must only use PIO mode and ball A14 requires a pull-up resistor to keep FLASH_CS3# high when in IDE mode.
FLASH_RE#	B13	O	<b>Read Enable Strobe.</b> This signal is asserted during read operations from the NOR array.
FLASH_WE#	C13	O	<b>Write Enable Strobe.</b> This signal is asserted during write operations to the NOR array.
FLASH_ALE	C14	O	<b>Address Latch Enable.</b> Controls external latch (e.g., 74x373) for latching the higher address bits in address phase.
FLASH_AD[27:26]/ AD[2:1], FLASH_AD25/ AD0, FLASH_AD[24:18]/ AD[9:3]	B11, A12, A11, B15, B16, A17, C17, D16, D17, E17	O	<b>Address Bus.</b> During the address phase, address [27:18] is put on the bus. During the data phase, address [9:0] is put on the bus.
FLASH_AD[17:10]/ IO[7:0]	E16, E15, D15, B17, C16, C15, A15, B14	I/O	<b>Multiplexed Address and I/O Bus.</b> During the address phase, NOR address [17:10] are placed on these lines. During the data phase, it is the NOR I/O data bus.
FLASH_IOCHRDY	A13	I	<b>I/O Channel Ready.</b> When a device on the bus wants to extend its current cycle, it pulls this signal low to insert the wait state.
<b>NAND Flash Mode</b>			
FLASH_CE[3:0]#	C12, A14, C10, B10	O	<b>Chip Enables.</b> These signals remain low during a NAND cycle.  <b>Note:</b> Ball A14 is the only ball that changes direction from IDE to Flash. Needs external pull-up for Flash use.
FLASH_RE#	B13	O	<b>Read Enable Strobe.</b> This signal is asserted during read operations from the NAND array.
FLASH_WE#	C13	O	<b>Write Enable Strobe.</b> This signal is asserted during write operations to the NAND array.
FLASH_ALE	C14	O	<b>Address Latch Enable.</b> Level signal to indicate an address byte is writing to the NAND Flash device.
FLASH_CLE	A11	O	<b>Command Latch Enable.</b> Indicates a command byte is being written to the device.
FLASH_IO[7:0]	E16, E15, D15, B17, C16, C15, A15, B14	I/O	<b>I/O Bus.</b> I/O bus for NAND Flash devices. Command, address, and data are sent on this bus. This bus is actively driven to zero with or without an LPC_CLK from and after reset.
FLASH_RDY/BUSY#	A13	I	<b>Ready/Busy#.</b> NAND Flash pulls this signal low to indicate it is busy with an internal operation. No further action is accepted except read status.

### 3.2.4 USB Interface

Signal Name	Ball No.	Type	Description
USB_PWR_EN1	P17	O	<b>USB Power Enable 1.</b> This signal is intended to be used to enable an external USB power source for Port 1, such as the National Semiconductor LM3526.  USB_PWR_EN1 is an active high signal. If low, it indicates that the external USB power source for Port 1 is turned off. Defaults off from reset.
USB_PWR_EN2	N16	O	<b>USB Power Enable 2.</b> This signal is intended to be used to enable an external USB power source for Port 2, such as the National Semiconductor LM3526.  USB_PWR_EN2 is an active high signal. If low, it indicates that the external USB power source for Port 2 is turned off. Defaults off from reset.
USB_OC_SENS#	N15	I	<b>USB Over Current Sense.</b> This signal is the logical OR or wired-OR from all external USB power supply devices, such as the National Semiconductor LM3526, and is shared by USB1 and USB2 (all four ports). When pulled low it causes both USB_PWR_EN1 and USB_PWR_EN2 to de-assert and generate an interrupt. Tie high if not used.
USB1_1_DATPOS	K16	I/O	<b>USB Port 1_1 Data Positive.</b> This is the positive differential side of the USB data for port 1_1. (Note 1, Note 2.)
USB1_1_DATNEG	K17	I/O	<b>USB Port 1_1 Data Negative.</b> This is the negative differential side of the USB data for port 1_1. (Note 1, Note 2.)
USB1_2_DATPOS	L16	I/O	<b>USB Port 1_2 Data Positive.</b> This is the positive differential side of the USB data for port 1_2. (Note 1, Note 2.)
USB1_2_DATNEG	L17	I/O	<b>USB Port 1_2 Data Negative.</b> This is the negative differential side of the USB data for port 1_2. (Note 1, Note 2.)
USB2_1_DATPOS	H17	I/O	<b>USB Port 2_1 Data Positive.</b> This is the positive differential side of the USB data for port 2_1. (Note 1, Note 2.)
USB2_1_DATNEG	H16	I/O	<b>USB Port 2_1 Data Negative.</b> This is the negative differential side of the USB data for port 2_1. (Note 1, Note 2.)
USB2_2_DATPOS	G17	I/O	<b>USB Port 2_2 Data Positive.</b> This is the positive differential side of the USB data for port 2_2. (Note 1, Note 2.)
USB2_2_DATNEG	G16	I/O	<b>USB Port 2_2 Data Negative.</b> This is the negative differential side of the USB data for port 2_2. (Note 1, Note 2.)
AV <sub>SS_USB</sub>	F16, J16, M16	AGND	<b>USB Analog Circuit Ground.</b> Total of three ground balls for the USB transceivers. Most applications should connect this to signal ground.
AV <sub>DD_USB</sub>	F17, J17, M17	APWR	<b>USB Analog Circuit Power.</b> Total of three power balls for the USB transceivers. Most applications should connect this to V <sub>IO</sub> .

Note 1. Use external 27  $\Omega$  series resistor on output. From reset, these outputs are in TRI-STATE. At board level, a 15 k $\Omega$  pull-down resistor is required per the USB specification.

Note 2. External clamping diodes may be needed to meet over voltage requirements.

### 3.2.5 System Management Bus (SMB) Interface

Signal Name	Ball No.	Type	Description
SMB_CLK	G3	I/O	<p><b>SMB Clock.</b> This is the clock for the System Management bus. It is initiated by the master of the current transaction. Data is sampled during the high state of the clock.</p> <p>An external pull-up resistor is required.</p> <p>Shared with GPIO14. Set GPIO14 to IN_AUX1 and OUT_AUX1 modes simultaneously to use as SMB_CLK. See Table 3-8 "GPIO Options" on page 47.</p> <p>External voltage applied to this ball should not exceed <math>V_{IO}</math>.</p>
SMB_DATA	F1	I/O	<p><b>SMB Data.</b> This is the bidirectional data line for the System Management bus. Data may change during the low state of the SMB clock and should remain stable during the high state.</p> <p>An external pull-up resistor is required.</p> <p>Shared with GPIO15. Set GPIO15 to IN_AUX1 and OUT_AUX1 modes simultaneously to use as SMB_DATA. See Table 3-8 "GPIO Options" on page 47.</p> <p>External voltage applied to this ball should not exceed <math>V_{IO}</math>.</p>

### 3.2.6 Low Pin Count (LPC) Interface

Signal Name (Note 1)	Ball No.	Type	Description
LPC_CLK	H1	I	<p><b>LPC Clock.</b> 33 MHz LPC bus shift clock.</p>
LPC_AD[3:0]	K1, J1, J2, H2	I/O	<p><b>LPC Address/Data Bus.</b> This is the 4-bit LPC bus. Address, control, and data are transferred on this bus between the Geode CS5535 companion device and LPC devices.</p> <p>An external pull-up of 100 k<math>\Omega</math> is required on these balls (if used in LPC mode) to maintain a high level when the signals are in TRI-STATE mode. From reset, these signals are not driven.</p> <p>LPC_AD3 is shared with GPIO19. LPC_AD2 is shared with GPIO18. LPC_AD1 is shared with GPIO17. LPC_AD0 is shared with GPIO16.</p> <p>See Table 3-8 "GPIO Options" on page 47 for further details.</p>
LPC_DRQ#	G1	I	<p><b>LPC DMA Request.</b> This is the LPC DMA request signal. Peripherals requiring service pull it low and then place a serially-encoded requested channel number on this line to initiate a DMA transfer.</p> <p>If the device wakes up from Sleep, at least six LPC_CLKs must occur before this input is asserted.</p> <p>Shared with GPIO20. See Table 3-8 "GPIO Options" on page 47. Tie high if selected as LPC_DRQ# but not used.</p>

### 3.2.6 Low Pin Count (LPC) Interface

Signal Name (Note 1)	Ball No.	Type	Description
LPC_SERIRQ	G2	I/O	<p><b>LPC Encoded IRQ.</b> This is the LPC serial interrupt request line, used to report ISA-style interrupt requests. It may be activated by either the Geode CS5535 companion device or an LPC peripheral.</p> <p>An external pull-up of 100 k<math>\Omega</math> is required if this ball is used in LPC mode to maintain a high level when the signal is in TRI-STATE. From reset, this signal is not driven.</p> <p>If the device wakes up from Sleep, at least six LPC_CLKs must occur before this input is asserted if operating in Quiet mode.</p> <p>Shared with GPIO21. See Table 3-8 "GPIO Options" on page 47.</p>
LPC_FRAME#	H3	O	<p><b>LPC Frame.</b> This signal provides the active-low LPC FRAME signal used to start and stop transfers on the LPC bus.</p> <p>Shared with GPIO22. See Table 3-8 "GPIO Options" on page 47.</p>

Note 1. All LPC signals, except LPC\_CLK are shared on GPIO balls (see Table 3-8 "GPIO Options" on page 47). The Geode CS5535 companion device powers up with this group of balls set to the LPC mode; to use them as GPIOs they must be explicitly reprogrammed. The LPC signals may be switched to GPIOs use via MSR 51400015h (see Section 6.6.2.11 "Ball Options Control (DIVIL\_BALL\_OPTS)" on page 335).

Use RESET\_OUT# for LPC reset.

Use any GPIO assigned as a PME for the LPC PME.

Use any GPIO assigned as an SMI for the LPC SMI.

Use general Sleep and Standby controls (SLEEP\_X, ball C2 and SLEEP\_Y, ball J3) in place of LPC\_PD# for LPC power-down.

### 3.2.7 Audio Codec 97 Interface

Signal Name (Note 1)	Ball No.	Type	Description
AC_CLK	M1	I	<p><b>Audio Bit Clock.</b> The serial bit clock from the codec. The frequency of the bit clock is 12.288 MHz and is derived from the 24.576 MHz crystal input to the external audio codec. Not required if audio not used; tie low.</p>
AC_S_OUT	L2	O	<p><b>Audio Controller Serial Data Out.</b> This output transmits audio data to the codec. This data stream contains both control data and the DAC audio data. The data is sent on the rising edge of the AC_CLK. Connect to the audio codec's serial data input pin.</p>
BOS1		I	<p><b>Boot Options Select Bit 1.</b> During system reset, this ball is the MSB of the 2-bit boot option (balls L2 and L3) and is used to determine the location of the system boot device. It should be pulled low if required by Table 3-5 "Boot Options Selection" on page 34, otherwise, an internal pull-up, asserted during reset, will pull it high. During reset, the ball output drivers are held in TRI-STATE mode, and the ball is sampled on the rising edge of RESET_OUT# (i.e., when external reset is de-asserted). After reset, this signal defaults low (off).</p>
AC_S_IN	L1	I	<p><b>Audio Controller Serial Data Input.</b> This input receives serial data from the audio codec. This data stream contains both control data and ADC audio data. This input data is sampled on the falling edge of AC_CLK. Connect to the audio codec's serial data output pin.</p>

## 3.2.7 Audio Codec 97 Interface (Continued)

Signal Name (Note 1)	Ball No.	Type	Description
AC_S_SYNC	L3	O	<b>Audio Controller Sync.</b> This is a 48 kHz sync pulse that signifies the beginning of a serial transfer on AC_S_OUT, AC_S_IN, and AC_S_IN2. AC_S_SYNC is synchronous to the rising edge of AC_CLK. Connect to the audio codec's SYNC pin.
BOS0		I	<b>Boot Options Select Bit 0.</b> During system reset, this ball is the LSB of the 2-bit boot option (balls L2 and L3), used to determine the location of the system boot device. It should be pulled low if required by Table 3-5 "Boot Options Selection" on page 34, otherwise, an internal pull up, asserted during reset, will pull it high. During reset, the ball drivers are held in TRI-STATE mode, and the ball is sampled on the rising edge of RESET_OUT# (i.e., when external reset is de-asserted). After reset, this signal defaults low (off).
AC_BEEP	K3	O	<b>Legacy PC/AT Speaker Beep.</b> Connect to codec's PC_BEEP. This function is only available when GPIO1 is programmed to OUT_AUX1. See Table 3-8 "GPIO Options" on page 47.
AC_S_IN2	J3	I	<b>Audio Controller Serial Data Input 2.</b> This input receives serial data from a second codec. This data stream contains both control data and ADC audio data. This input data is sampled on the falling edge of AC_CLK. If the codec's Ready bit is set in this stream (slot 0, bit 15), then it is functionally ORed with AC_S_IN. Connect to a second codec's serial data output. This function is only available when GPIO12 is programmed to IN_AUX1. See Table 3-8 "GPIO Options" on page 47.

Note 1. Use RESET\_OUT# for AC97 reset.

### 3.2.8 GPIOs

Table 3-8 gives the dedicated functions associated with each GPIO. These functions may be invoked by configuring the associated GPIO to the IN\_AUX1, OUT\_AUX1, or OUT\_AUX2 modes. (The functions themselves are described in Table 3-9 "GPIOx Available Functions Descriptions" on page 49.) The column "Recommended Use" is a guideline for system designers to assign GPIO functionality. Any GPIO input can be mapped to an interrupt, ASMI, or PME. Details of configuring the GPIOs are given in Section 6.16 "GPIO Subsystem Register Descriptions" on page 450. All GPIOs have selectable pull-up and/or pull-down resistors available on the output, except for those indicated by Note 3 in the "Weak PU/PD" column of Table 3-8.

**Table 3-8. GPIO Options**

GPIO	Ball No.	Power Domain	Buffer Type	Post Reset		Recommended Use	Function Programming Options		
				Weak PU/PD	I/O Config		IN_AUX1	OUT_AUX1 Note 1	OUT_AUX2 Note 2
GPIO0	R2	W	PCI	(Note 3)	Disabled	PCI_INTA# (Note 4)			
GPIO1	K3	W	Q7	PU	Disabled	---		AC_BEEP	MFGPT0_C2
GPIO2	B12	W	IDE	(Note 3)	Disabled	---	IDE_IRQ0		
GPIO3	E1	W	SMB	(Note 3)	Disabled	DDC_SCL (Note 5)	UART2_RX		
GPIO4	E2	W	SMB	(Note 3)	Disabled	DDC_SDA (Note 5)		UART2_TX	
GPIO5	D3	W	Q7	Auto-sense (Note 6)	Disabled	---	MFGPT1_RS	MFGPT0_C1	
GPIO6	D2	W	Q7	Auto-sense (Note 6)	Disabled	---	MFGPT0_RS	MFGPT1_C1	MFGPT2_C2
GPIO7	C2	W	PCI	(Note 3)	Disabled	PCI_INTB# (Note 4)		MFGPT2_C1	SLEEP_X
GPIO8	E3	W	Q7	PU	Disabled	---		UART1_TX	UART1_IR_TX
GPIO9	D1	W	Q7	PU	Disabled	---	UART1_RX or UART1_IR_RX		
GPIO10	C3	W	Q7	PU	Disabled	--- (Note 7)	THRM_ALRM#		
GPIO11	A1	W	Q7	PU	Disabled	---		SLP_CLK_EN#	MFGPT1_C2
GPIO12	J3	W	Q7	PD	Disabled	---	AC_S_IN2		SLEEP_Y
GPIO13	F2	W	Q7	PU	Disabled	--- (Note 7)	SLEEP_BUT		
GPIO14	G3	W	SMB	(Note 3)	Disabled	--- (Note 8)	SMB_CLK_IN	SMB_CLK_OUT	
GPIO15	F1	W	SMB	(Note 3)	Disabled	--- (Note 8)	SMB_DATA_IN	SMB_DATA_OUT	
GPIO16	H2	W	PCI	(Note 3)	LPC (Note 9)	LPC_AD0			
GPIO17	J2	W	PCI	(Note 3)	LPC (Note 9)	LPC_AD1			
GPIO18	J1	W	PCI	(Note 3)	LPC (Note 9)	LPC_AD2			
GPIO19	K1	W	PCI	(Note 3)	LPC (Note 9)	LPC_AD3			
GPIO20	G1	W	PCI	(Note 3)	LPC (Note 9)	LPC_DRQ#			
GPIO21	G2	W	PCI	(Note 3)	LPC (Note 9)	LPC_SERIRQ	MFGPT2_RS		
GPIO22	H3	W	PCI	(Note 3)	LPC (Note 9)	LPC_FRAME#			
GPIO24	C9	S	SMB	(Note 3)	Disabled	---		WORK_AUX	
GPIO25	A9	S	Q7	PU/PD	Disabled	---	LOW_BAT#		MFGPT7_C2
GPIO26	B7	S	Q7	PU/PD	Disabled	PME# (Note 10)	MFGPT7_RS		
GPIO27	C8	S	Q7	PU/PD	Disabled	---		MFGPT7_C1	32KHZ
GPIO28	A8	S	Q7	PU/PD	Input Enabled (Note 11)	PWR_BUT# (Note 12)	PWR_BUT# (Note 13)		

Note 1. See Section 6.16.2.5 on page 458 for ball configuration.

Note 2. See Section 6.16.2.6 on page 459 for ball configuration.

Note 3. No internal pull-up/down available. If not used, tie low.

- Note 4. Any GPIO can be used as an interrupt input without restriction. These particular GPIOs have PCI I/O buffer types for complete PCI bus compatibility. However, such strict compatibility is generally not required.
- Note 5. Applications incorporating a CRT often require support for the Display Data Channel (DDC) serial interface. These particular GPIOs have open collector SMB I/O buffer types required by the DDC interface specification. The DDC protocol supplied by AMD is provided via software implementation and defaults to these GPIOs. However, any design not needing strict DDC electrical support can use other GPIOs. Lastly, applications not incorporating DDC use at all may use these GPIOs without restriction.
- Note 6. See section 5.15.5.6 "Auto-sense" on page 165 for an explanation of auto-sense.
- Note 7. Internal signal is active high. Use GPIO invert for active low external.
- Note 8. When both IN\_AUX1 and AUX\_OUTx are enabled, I/O direction on this ball is controlled by the SMB Controller.
- Note 9. Defaults to LPC use. Use Ball Options MSR (see Table 3-6 on page 34) to switch this ball to GPIO control.
- Note 10. Any GPIO can be used as a Power Management Event (PME) wakeup input without restriction. PMEs are supported for both Sleep and Standby wakeup. However, if Standby wakeup is desired, a GPIO on the Standby power domain must be used. Only GPIO[24:28] are supplied via the Standby Power Rail and are typically used as follows:  
GPIO24 - Auxiliary Working Power Control  
GPIO25 - Low Battery Alarm  
GPIO26 - PME  
GPIO27 - MFGPT setup to provide a blink  
GPIO28 - Power Button  
Depending on application use, the PME function could be moved to GPIO[24:27]. If only external PME wakeup from Sleep is required, the PME function could be moved to GPIO[0:23]. Lastly, the PME function could simply not be used, making more GPIOs available for other uses.
- Note 11. GPIOH\_IN\_EN (GPIO Offset A0h) and GPIOH\_IN\_AUX1\_SEL (GPIO Offset B4h) are enabled.
- Note 12. Reset default.
- Note 13. If the GPIO28 function is desired, the power button functionality in the PMC must be disabled before the IN\_AUX1 function is disabled.



**3.2.8.1 GPIO Functions and Recommended Usage**

Functions listed in Table 3-9 are functions that may be assigned to specific GPIO balls. The “Ball No.” column

gives the ball that must be used if this function is selected, and the “GPIOx” column gives the GPIO that the function is associated with.

**Table 3-9. GPIOx Available Functions Descriptions**

Function Name	Ball No.	GPIO[x]	Type	Description
32KHZ	C8	GPIO27	O	<b>32 kHz Clock.</b> When invoked, this ball produces a buffered output of the 32 kHz clock provided on KHZ32_XCI and KHZ32_XCO (balls A4 and B3, respectively).  This option is invoked by selecting the OUT_AUX2 option of GPIO27. Note that since GPIO27 is in the Standby power domain, the 32 kHz clock output will continue in Sleep and Standby states.
AC_BEEP	K3	GPIO1	O	<b>Legacy PC/AT Speaker Beep.</b> Connect to codec's PC_BEEP.
DDC_SCL	E1	GPIO3	I/O	<b>DDC Serial Clock.</b> This is a “recommended use” for GPIO3, because this is one of the few GPIOs that have a high drive capacity, open-drain output. The serial clock function must be implemented in software to support DDC monitors. There is no dedicated DDC clock function within the Geode CS5535 companion device.
DDC_SDA	E2	GPIO4	I/O	<b>DDC Serial Data.</b> This is a “recommended use” for GPIO4, because this is one of the few GPIOs that have a high drive capacity, open-drain output. The serial data function must be implemented in software to support DDC monitors. There is no dedicated DDC data function within the Geode CS5535 companion device.
LOW_BAT#	A9	GPIO25	I	<b>Low Battery Detect.</b> This is a “recommended use” for GPIO25 in battery-powered systems. It is invoked by setting GPIO25 to the IN_AUX1 mode. The signal is intended to be driven low by an external circuit when the battery voltage falls below a preset value (determined by the external circuit). It could be used to generate a PME (interrupt) - connected to LowBat function in the Power Management Controller that would then de-assert WORKING and WORK_AUX, if no software action is taken within a programmable time.
MFGPT0_C1	D3	GPIO5	O	<b>Multi-Function General Purpose Counter #0 - Compare 1 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 1 registers.
MFGPT0_C2	K3	GPIO1	O	<b>Multi-Function General Purpose Counter #0 - Compare 2 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 2 registers.
MFGPT0_RS	D2	GPIO6	I	<b>Multi-Function General Purpose Counter #0 - Restart.</b> An input to the counter that causes it to be reset to initial conditions and then to resume counting.
MFGPT1_C1	D2	GPIO6	O	<b>Multi-Function General Purpose Counter #1 - Compare 1 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 1 registers.
MFGPT1_C2	A1	GPIO11	O	<b>Multi-Function General Purpose Counter #1 - Compare 2 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 2 registers.

Table 3-9. GPIOx Available Functions Descriptions (Continued)

Function Name	Ball No.	GPIO[x]	Type	Description
MFGPT1_RS	D3	GPIO5	I	<b>Multi-Function General Purpose Counter #1 - Restart.</b> An input to the counter that causes it to be reset to initial conditions and then to resume counting.
MFGPT2_C1	C2	GPIO7	O	<b>Multi-Function General Purpose Counter #2 - Compare 1 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 1 registers.
MFGPT2_C2	D2	GPIO6	O	<b>Multi-Function General Purpose Counter #2 - Compare 2 Out.</b> Output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 2 registers.
MFGPT2_RS	G2	GPIO21	I	<b>Multi-Function General Purpose Counter #2 - Restart.</b> An input to the counter that causes it to be reset to initial conditions and then to resume counting.
MFGPT7_C1	C8	GPIO27	O	<b>Multi-Function General Purpose Counter #7 - Compare 1 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 1 registers.
MFGPT7_C2	A9	GPIO25	O	<b>Multi-Function General Purpose Counter #7 - Compare 2 Out.</b> An output from the counter that, when asserted, indicates the counter has reached the conditions set up in the counter's Compare 2 registers.
MFGPT7_RS	B7	GPIO26	I	<b>Multi-Function General Purpose Counter #7 - Restart.</b> An input to the counter that causes it to be reset to initial conditions and then to resume counting.
PCI_INTA#	R2	GPIO0	I	<b>PCI Interrupt A.</b> This is a "recommended use" for GPIO0, because this GPIO has a PCI-compatible output type.
PCI_INTB#	C2	GPIO7	I	<b>PCI Interrupt B.</b> This is a "recommended use" for GPIO7, because this GPIO has a PCI-compatible output type.
PME#	B7	GPIO26	I	<b>Power Management Event.</b> This is a "recommended use" for GPIO26. By mapping this GPIO (or any other) to the PME# function, the Geode CS5535 companion device may be awakened from a Sleep state when the mapped ball (in the recommended case, ball B7) is pulled low.

Table 3-9. GPIOx Available Functions Descriptions (Continued)

Function Name	Ball No.	GPIO[x]	Type	Description
PWR_BUT#	A8	GPIO28	I	<p><b>Power Button.</b> This GPIO can be mapped to the PMC “button-push” event, that may be used to implement the power ON and the four-second-delay power OFF functions. Note that GPIO28 comes up in the IN_AUX1 mode after reset, enabling this feature.</p> <p>Any power button change on this input must be at least two KHZ32_XCI edges (approximately 62 <math>\mu</math>s) in duration to be correctly detected. If spurious transitions smaller than this are possible, then use on-chip GPIO input filter function to insure proper operation. Additionally, the rise or fall time on this input must be less than 10 <math>\mu</math>s. If transition times longer than this are possible, then use the on-chip GPIO input filter function to insure proper operation.</p> <p>From the first power-up of the Standby power domain under which no filter is enabled, spurious transitions on the first high-to-low power button push are acceptable as long as the input is eventually low at least two KHZ32_XCI edges. Additionally, transition times as slow as 1 ms are acceptable for the first push. Note that these relaxed requirements work because this input is effectively a “don’t care” at the hardware level after the first power-up until software enables use of the power button. Before enabling use, the software can setup the GPIO filter or other functions as needed.</p> <p>Per the discussion of the Skip feature in Section 5.17 “Power Management Control” on page 172, this input may be tied to ground in order for the system to come on immediately when Standby and Working power are available. Specifically, systems that do not incorporate a power button should tie this input to ground.</p> <p>One side effect of the Skip feature, is that the platform design must insure that this input is not low when Standby power is applied if the Skip feature is not desired. Specifically, systems that incorporate use of a power button must insure that this input ramps to a high no more than 1 <math>\mu</math>s behind <math>V_{IO\_VSB}</math> ramp-up. Failure to quickly establish a high on this input during power up could result in a spurious Skip. See Section 7.6 “Skip Parameter” on page 573 for AC timing.</p>
AC_S_IN2	J3	GPIO12	I	<p><b>Audio Controller Serial Data Input 2.</b> This input receives serial data from a second codec. This data stream contains both control data and ADC audio data. This input data is sampled on the falling edge of AC_CLK. If the codec’s Ready bit is set in this stream (slot 0, bit 15), then it is functionally ORed with AC_SDATA_IN1. Connect to second codec’s serial data output.</p>
SLP_CLK_EN#	A1	GPIO11	O	<p><b>Sleep Clock Enable.</b> This signal is a control that is intended to be connected to the external system clock generator chip. The intended use is, when high, the clock generator runs; when low, the clock generator turns off. From reset, a pull-up makes this GPIO high. The active state of this signal indicates that the Geode CS5535 companion device is in the Sleep state.</p>
SLEEP_BUT	F2	GPIO13	I	<p><b>Sleep Button.</b> This GPIO can be mapped to a PME “button-push” type event and used to request the system software to put the system to Sleep.</p>

Table 3-9. GPIOx Available Functions Descriptions (Continued)

Function Name	Ball No.	GPIO[x]	Type	Description
SLEEP_X	C2	GPIO7	O	<b>Sleep X.</b> This general purpose power control output becomes active as the Geode CS5535 companion device enters and exits various power management modes. It may be used by external devices to control their power states synchronous with power state changes in the Geode CS5535 companion device. It may be configured as active high or active low.
SLEEP_Y	J3	GPIO12	O	<b>Sleep Y.</b> This general purpose power control output becomes active as the Geode CS5535 companion device enters and exits various power management modes. It may be used by external devices to control their power states synchronous with power state changes in the Geode CS5535 companion device. It may be configured as active high or active low.
SMB_CLK_IN	G3	GPIO14	I	<b>SMB Clock In / SMB Clock Out.</b> This is the clock for the SMB. In order to use it properly, the associated GPIO (GPIO14) should be set to IN_AUX1 and OUT_AUX1 simultaneously. The SMB controller determines the direction (in or out) of the associated ball.
SMB_CLK_OUT			O	
SMB_DATA_IN	F1	GPIO15	I	<b>SMB Data In / SMB Data Out.</b> This is the data line for the SMB. In order to use it properly, the associated GPIO (GPIO15) should be set to IN_AUX1 and OUT_AUX1 simultaneously. The SMB controller determines the direction (in or out) of the associated ball.
SMB_DATA_OUT			O	
THRM_ALRM#	C3	GPIO10	I	<b>Thermal Alarm.</b> When connected to an external thermal monitor, this input can act as a thermal fail-safe to shut down power by signalling the power management controller to de-assert WORKING and WORK_AUX. Set GPIO10 to the IN_AUX1 mode to enable this feature.
IDE_IRQ0	B12	GPIO2	I	<b>IDE Interrupt.</b> Indicates the external IDE device has completed the DMA operation.
UART1_IR_TX	E3	GPIO8	O	<b>UART1 Infrared Transmit.</b> This signal is the data output (TX) from the infrared mode of UART1. It is available when GPIO8 is switched to the OUT_AUX2 mode
UART1_RX	D1	GPIO9	I	<b>UART1 Receive or UART1 Infrared Receive.</b> This signal is the data input (RX) to the UART1. It acts as the input in both IR and conventional modes of UART1. It is available when GPIO9 is switched to the IN_AUX1 mode.
UART1_IR_RX			I	
UART1_TX	E3	GPIO8	O	<b>UART1 Transmit.</b> This signal is the data output (TX) from the conventional mode of UART1. It is available when GPIO8 is switched to the OUT_AUX1 mode.
UART2_RX	E1	GPIO3	I	<b>UART2 Receive.</b> This signal is the data input (RX) to the UART2. It acts as the input of UART2. It is available when GPIO3 is switched to the IN_AUX1 mode.
UART2_TX	E2	GPIO4	O	<b>UART2 Transmit.</b> This signal is the data output (TX) from the conventional mode of UART2. It is available when GPIO4 is switched to the OUT_AUX1 mode.
WORK_AUX	C9	GPIO24	O	<b>Working Auxiliary.</b> This output is intended to be used to control external power sources to all devices except memory (which is intended to be controlled by WORKING). WORK_AUX de-asserts in synchronism with WORKING.

### 3.2.9 Debug and Manufacturing Test Interface

Signal Name	Ball No.	Type	Description
TCK	N2	I	<b>JTAG Test Clock.</b>
TMS	N3	I	<b>JTAG Test Mode Select.</b>
TDI	P1	I	<b>JTAG Test Data In.</b>
TDO	P2	O, TS	<b>JTAG Test Data Out.</b> From reset, this output is in TRI-STATE mode. It is only enabled and driven when commanded to output or pass-through data per JTAG standards.
T_DEBUG_IN	M2	I	<b>Test Debug Input.</b> Input to the GeodeLink Control Processor (GLCP) from the Geode GX processor.
T_DEBUG_OUT	M3	O	<b>Test Debug Out.</b> Output from the GeodeLink Control Processor (GLCP) to the Geode GX processor.
LVD_TEST	B9	O	<b>Low Voltage Detect Test.</b> Manufacturing test only. No operational use. Make no connection.
TEST_MODE	A6	I	<b>Test Mode.</b> Manufacturing test only. No operational use. Tie low.
FUNC_TEST	F3	I	<b>Functional Test.</b> Manufacturing test only. No operational use. Tie low.

### 3.2.10 Power, Ground, and No Connects

Signal Name (Note 1)	Ball No.	Type	Description
V <sub>CORE</sub>	D8, D10, H4, H14, K4, K14, P8, P10	PWR	<b>1.2V or 1.5V (Nominal) Core Power Working Connection (Total of 8)</b>
V <sub>CORE_VSB</sub>	A7	PWR	<b>1.2V or 1.5V (Nominal) Core Power Standby Connection</b>
V <sub>IO</sub>	D4, D6, D9, D12, D14, F4, F14, M4, M14, P4, P6, P9, P12, P14	PWR	<b>3.3V (Nominal) I/O Power Connection (Total of 14)</b>
V <sub>IO_VSB</sub>	B6	PWR	<b>3.3V (Nominal) I/O Power Standby Connection</b>
V <sub>SS</sub>	D5, D7, D11, D13, E4, E14, G4, G14, J4, J14, L4, L14, N4, N14, P5, P7, P11, P13	GND	<b>Ground Connection (Total of 18)</b>
NC	A2, A16, B1, B4, B5, C11, G15, H15, J15, K15, L15, M15, R3, R9, T2, U2, U15, U16, U17	---	<b>No Connection (Total of 19).</b> These lines must be left disconnected. Connecting any or these lines to a pull-up/down resistor, an active signal, power, or ground could cause unexpected results and possible malfunctions.

Note 1. For module specific power and ground signals see:  
 Section 3.2.1 "System Interface Signals" on page 36.  
 Section 3.2.4 "USB Interface" on page 43.

# 4

## Global Concepts and Features

### 4.1 GeodeLink™ Architecture Overview

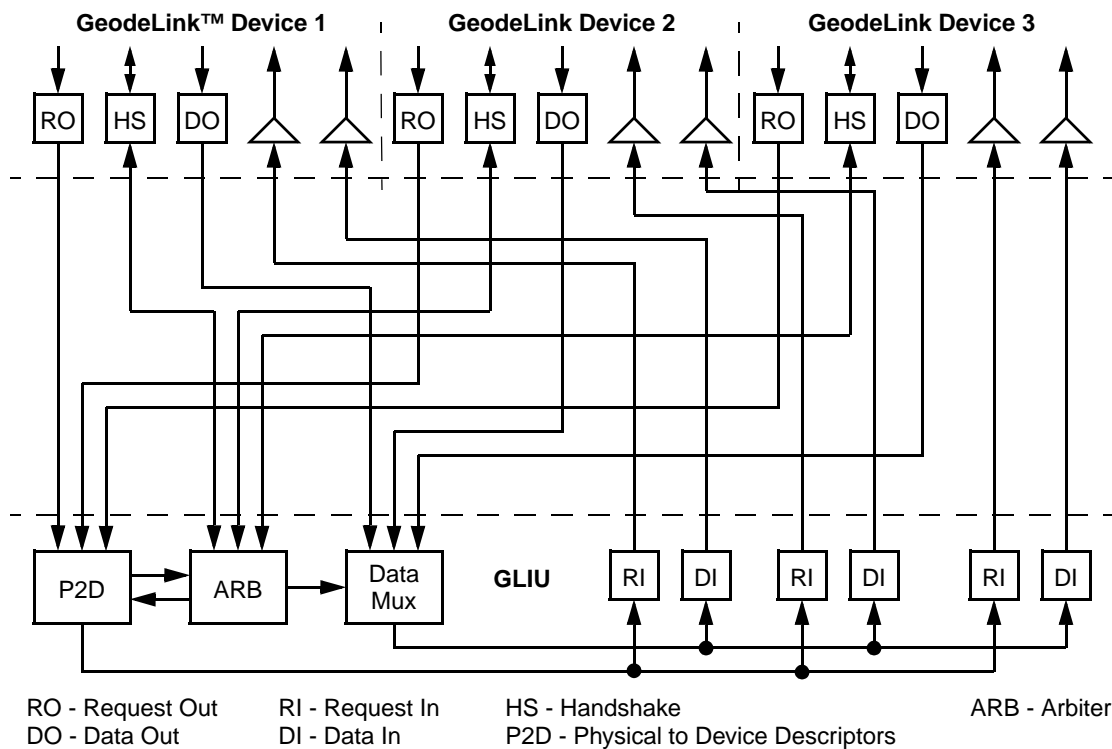
The information in this section provides a basic understanding of the architecture used to internally connect GeodeLink™ Devices. The actual existence of architecture is generally invisible to the user and the system programmer. AMD Core BIOS software provides all GeodeLink initialization and support, including related Model Specific Registers (MSRs). Additionally, this software provides a *Virtual PCI Configuration Space* that abstracts the architecture to industry standard interfaces. From this interface, all GeodeLink Devices appear in one PCI multi-function configuration space header on the external PCI bus.

A simplified view of a GLIU connected with three generic GeodeLink Devices is illustrated in Figure 4-1. The following points are relevant:

- All outputs from a GeodeLink Device to the GLIU are registered.
- All outputs from the GLIU to a GeodeLink Device are registered. Furthermore, there are dedicated output registers for each GeodeLink Device.
- GeodeLink Device inputs from the GLIU need not be registered, but they are buffered at the interface.
- All connections between the GeodeLink Devices and GLIU are dedicated point-to-point connections with one source and one load. There are no buses in TRI-STATE mode.
- The GLIU itself is a GeodeLink Device and is always Port 0.

#### 4.1.1 Introduction

This component is based on the GeodeLink packet architecture. It consists of a set of GeodeLink Devices and a GeodeLink Control Processor (GLCP) connected through the GeodeLink Interface Unit (GLIU).



**Figure 4-1. Simplified GLIU with Generic GeodeLink™ Devices**

The GLIU implements the “bus”. Transactions between GeodeLink Devices and GLIU are conducted with packets. The GLIU accepts request packets from masters and routes them to slaves. Similarly, slave response packets are routed back to the master. The bus is non-blocking. Several requests can be pending, but order is guaranteed. Broadcasts are not allowed. All packets have one source and one destination.

#### 4.1.2 Routing

The Physical to Device (P2D) descriptors control the routing of the packets. The descriptors are initialized by software at system setup. They establish the address map to be used by the system. They associate a memory or I/O address range with a GLIU port.

When a request packet arrives from a Request Out (RO) port, the address and other attributes in the packet are used to look up the destination port. If the port Request In (RI) is available, the request is passed. If there are multiple requests, priorities are used to establish which requestor and destination port utilize the transfer cycle. A transfer from an RO to an RI takes one clock edge.

#### 4.1.3 Response Packets

Earlier in this section, it was indicated that an RO can be used to present a write data packet or a read response packet. The use and need of a read response packet for a read request is obvious. However, there is also an optional write response packet. This tells the requestor that the write has completed. This is used to hold a processor I/O write instruction until the response is received, that is, I/O writes are never posted. Memory writes are always posted.

The response packet is also used to generate Synchronous System Management Interrupts (SSMIs). System Management Mode (SMM) is used for hardware emulation and other traps. An SSMI can be generated by a GeodeLink Device or via special GLIU descriptors. When the response arrives back at the processor, interface circuits generate an SMI to invoke the SMM software. Lastly, all response packets contain an *exception* flag that can be set to indicate an error.



### 4.1.4 ASMI and Error

Two additional signals are needed to complete this GeodeLink architecture overview: Asynchronous System Management Interrupt (ASMI) and Error. Each GeodeLink Device outputs these ASMI and Error signals.

An ASMI is much like a legacy interrupt, except it invokes the SMM handler. As the name suggests, an ASMI is an asynchronous event, while an SSMI is synchronous to the instruction that generated it.

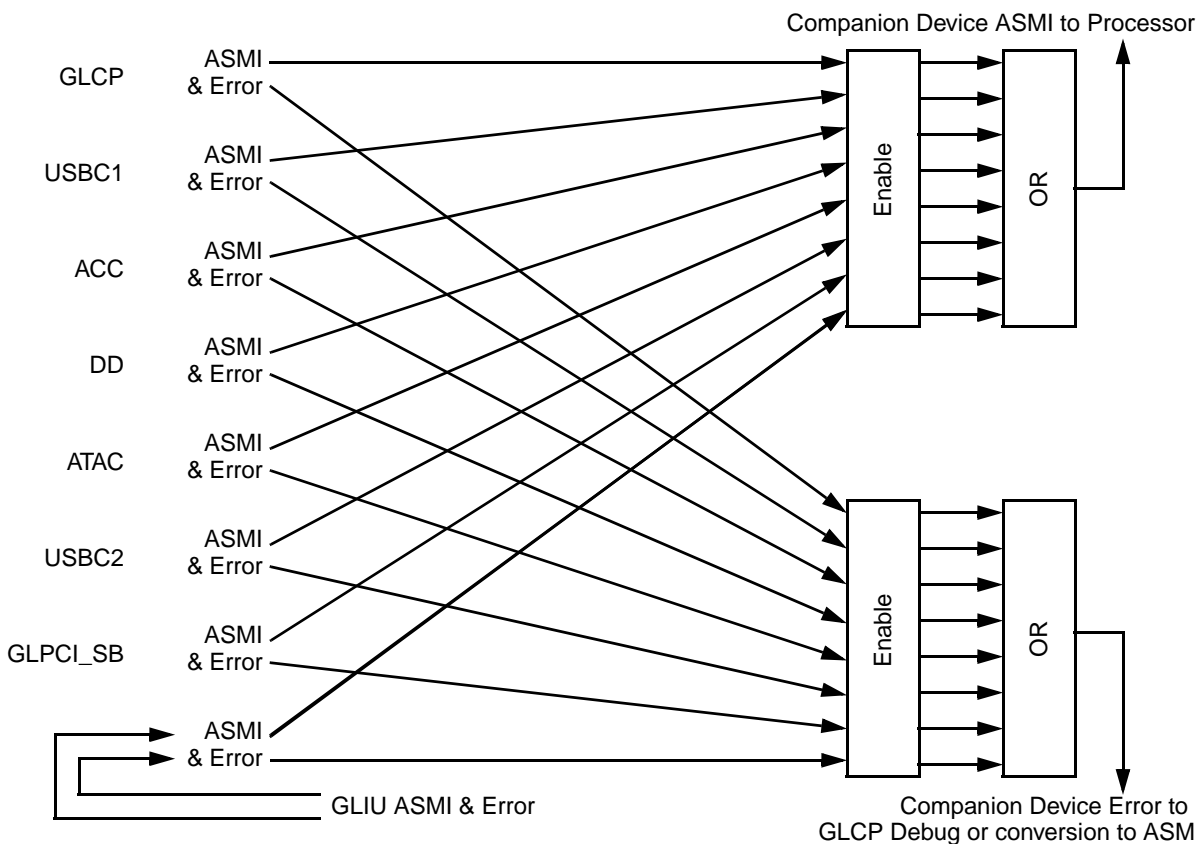
The Error signal simply indicates some type of unexpected error has occurred. A device asserts this signal when an unexpected error occurs. In a normal operating system, this would not be asserted. For example, a disk read error or ethernet network error would be signaled using normal GeodeLink packet mechanisms. This signal is reserved for the truly unexpected.

Each GeodeLink Device has mechanisms for enabling and mapping multiple internal sources down to these singular outputs. The mechanism consists of the logical “OR” of all enabled sources. The GLIU receives the ASMI and Error

pair from each GeodeLink Device. It has the same “OR” and enable mechanism that finally results in a single ASMI and Error pair for the whole component (see Figure 4-2). The ASMI is routed to the processor, while the Error is routed to the GLCP. Within the GLCP, the Error signal can be mapped into an ASMI for routing back into the GLIU.

### 4.1.5 Topology

The connection of the GLIU to the seven GeodeLink Devices of the Geode CS5535 companion device is illustrated in Figure 4-3. Note the port number of the GeodeLink Device; by design convention, the GLIU is always Port 0. Part of the Physical to Device (P2D) descriptor is a port number. When there is a hit on the descriptor address, the port number indicates which GeodeLink Device to route the packet to. If there is no hit, then the packet is routed to the default port. For the Geode CS5535 companion device, the default is always Port 4, that is, the Diverse Device (DD).



**Figure 4-2. GeodeLink™ Architecture ASMI and Error Routing**

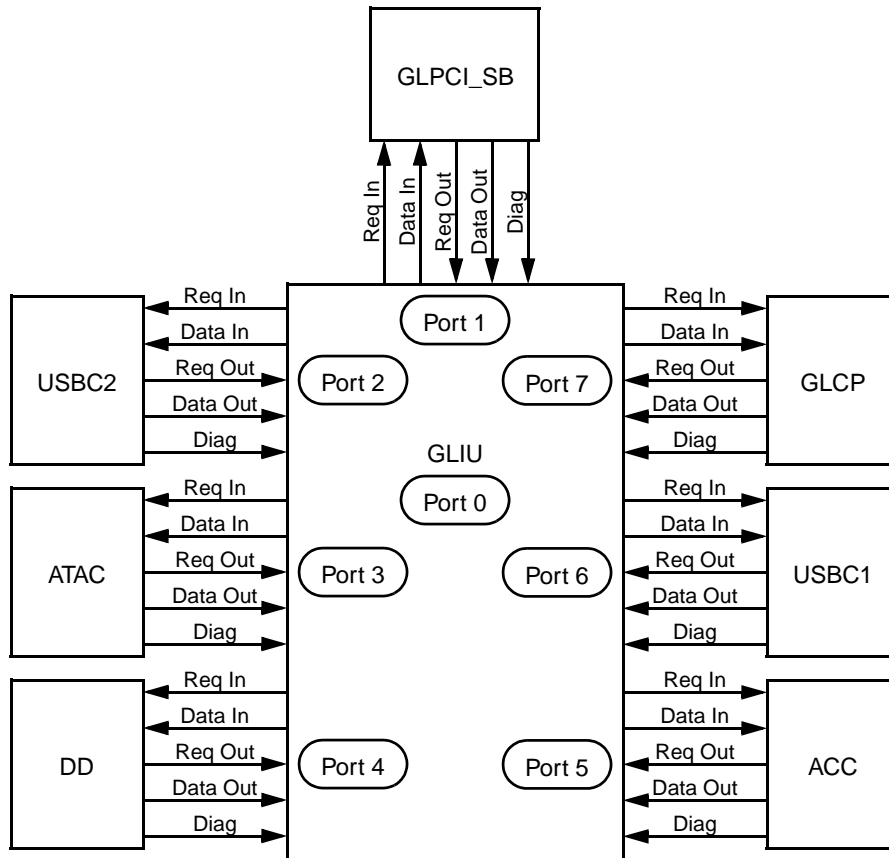
**4.1.6 Address Spaces and MSRs**

The GLIU and GeodeLink Devices support the traditional memory and I/O spaces. The memory space supports a traditional 32-bit byte address with associated byte enables. The I/O space is a 20-bit byte address with byte enables. I/O registers can be 8, 16, or 32 bits. The GLIU has both memory and I/O P2Ds for routing.

In addition to the above spaces, there is a Model Specific Register (MSR) space that is tied to the GeodeLink topology. As introduced in the previous section, the GLIU has eight ports with Port 0 assigned to the GLIU. An MSR “address” is relative to the device making a request to it and the topology between the requestor and the MSR. Thus, for the Geode GX processor to address an MSR in the Geode CS5535 companion device, it specifies a series of ports that must be traversed to get there. Once a specific device port is identified, additional address bits are available to select a specific MSR within a given device.

MSR space is functionally similar to PCI configuration space. At boot time system initialization, the Core BIOS (see Section 4.1 “GeodeLink™ Architecture Overview” on page 55) traverses the topology of the system to determine what is present. By convention, the first MSR at each port is an ID register that indicates a specific device. Once the Core BIOS knows what is present, it assigns devices to specific locations in the appropriate memory or I/O address space using MSRs. Generally, MSRs are used to configure and set up GeodeLink Devices, but are not used for ongoing operations.

The “assignment” MSRs are located in the GLIUs as “descriptors”. The “assignment descriptor” basically says: “route a request packet containing address X to port Y”. Port Y can be the final device or another GLIU. This second GLIU must have assignments to route address X to port Z. This process continues until the final device port is specified.



**Figure 4-3. AMD Geode™ CS5535 Companion Device GeodeLink™ Architecture Topology**

In addition to the “positive” address decode above, each GLIU has a subtractive port that takes all addresses not assigned to a specific port. There is always a default subtractive port path to the boot ROM to allow the central processor to start executing code from time zero. Thus, from system reset, there is a default memory address path that allows the first processor instruction fetch to:

- 1) Proceed down through the two Geode GX processor GLIUs;
- 2) cross the PCI bus to the Geode CS5535 companion device;
- 3) proceed down through the Geode CS5535 companion device GLIU to the default port connected to the DD; and
- 4) access the boot device connected to the DD.

#### 4.1.7 Special Cycles and BIZZARO Flag

The BIZZARO flag is used to indicate special cycles and exceptions to normal packet operation. All special cycles traverse the GLIU system as I/O packets with the BIZZARO flag set. The special cycles are:

- 1) Interrupt Acknowledge: I/O read from address zero.
- 2) Shutdown: I/O write to address zero.
- 3) Halt: I/O write to address one.

## 4.2 MSR Addressing

An MSR address consists of the fields shown in Table 4-1.

When a GLIU receives an MSR packet, it routes the packet to the port specified in *Field 0* but shifts address bits [31:14] to the left by three bits and replaces bits [16:14] with zero. Thus, Field 1 is moved to Field 0, Field 2 is moved to Field 1, etc. The address field always remains unchanged and selects one 64-bit MSR per address value, that is, the address value 0 accesses one 64-bit register, the address value 1 accesses one 64-bit register, the address value 2 accesses one 64-bit register, etc. There are no MSR byte enables. All 64 bits are always written and read.

Many Geode CS5535 companion device MSRs are only 32 bits in physical size. In these cases, interface logic discards the upper 32 bits on writes and pads the upper 32 bits on reads. Read padding is undefined. Lastly, Geode CS5535 companion device GeodeLink Devices only decode enough bits of the address to select one of *N* MSRs, where *N* is the total number of MSRs in the device. For example, if a GeodeLink Device has only 16 MSRs, then the addresses 0x2001, 0x0201, 0x0021, and 0x0x0001 all access MSR number 1, while the addresses 0x200F, 0x020F, 0x002F, and 0x0x000F all access MSR number 15.

To access a given GeodeLink Device, use Table 4-2 "MSR Addresses from AMD Geode™ GX Processor" on page 60. Note the target device addresses:

GLPCI_SB	5100xxxxh
GLIU	5101xxxxh
USBC2	5120xxxxh
ATAC	5130xxxxh
DD	5140xxxxh
ACC	5150xxxxh
USBC1	5160xxxxh
GLCP	5170xxxxh

The xxxx portion refers to the MSR addresses as they appear any place within Section 6.0 "Register Descriptions" on page 199. To form a complete MSR address, “OR” an address provided in a register description section with the appropriate address above.

Table 4-1. MSR Routing Conventions

Routing Field 0	Routing Field 1	Routing Field 2	Routing Field 3	Routing Field 4	Routing Field 5	Address Field
Bits [31:29]	Bits [28:26]	Bits [25:23]	Bits [22:20]	Bits [19:17]	Bits [16:14]	Bits [13:0]

Table 4-2. MSR Addresses from AMD Geode™ GX Processor

Routing Field 0	Routing Field 1	Routing Field 2	Routing Field 3	Routing Field 4	Routing Field 5	GeodeLink™ Device Target Name & Address	Comment
Bits [31:29]	Bits [28:26]	Bits [25:23]	Bits [22:20]	Bits [19:17]	Bits [16:14]		
These bits are shifted off to the left and never enter the Geode CS5535 companion device.			These bits are shifted into positions [31:23] by the time they reach the Geode CS5535 companion device. Bits in positions [22:14] are always 0 after shifting.				
010	100	010	000	000	000	GLPCI_SB 5100xxxxh	This all-zero convention indicates to the GLPCI_SB that the MSR packet coming across the PCI bus is actually for the GLCPI_SB.
010	100	010	000	Non-zero value		GLIU 5101xxxxh	This non-zero convention indicates to the GLPCI_SB that the MSR packet coming across the PCI bus should be forwarded to the GLIU. The GLIU only looks at [22:20] and hence, keeps the packet.
010	100	010	001	Any value		Illegal	The GLIU can not send any packets back to the port it came from.
010	100	010	010	Any value		USBC2 5120xxxxh	
010	100	010	011	Any value		ATAC 5130xxxxh	
010	100	010	100	Any value		DD 5140xxxxh	
010	100	010	101	Any value		ACC 5150xxxxh	
010	100	010	110	Any value		USBC1 5160xxxxh	
010	100	010	111	Any value		GLCP 5170xxxxh	

### 4.3 Typical GeodeLink™ Device

A typical or “generic” Geode CS5535 companion GeodeLink Device is illustrated in Figure 4-4 along with internal and external connections. The GeodeLink Device consists of the Native Block, GeodeLink Adapter, MSRs, and Clock Control Units (CCU). Each of these is discussed in the following paragraphs.

Before going into the blocks of the typical device, it should be noted that the following modules in the Geode CS5535 companion device follow this model very closely:

- AC97 Controller (ACC)
- ATA-5 Controller (ATAC)
- Diverse Device (DD)

Specifically, they all use the GeodeLink Adapter.

The Native Block performs the “useful” work for the device. For example, in a serial port device, the transmit parallel to serial shift register is located in this block. The Native Block connects to the outside world, that is, external devices, via the I/O cells and pads. The Native Block contains registers that are manipulated by software to perform the “work”. These are operational registers that are typically manipulated by device drivers. The Native Blocks are covered in detail in the corresponding module’s register descriptions.

The GeodeLink Adapter sits between the GLIU and the Local bus. The Local bus is a traditional address/data bus supporting GeodeLink Adapter to Native Block transactions (slave transactions) and Native Block to GeodeLink

Adapter transactions (master transactions). However, it is a single transaction bus in that any given slave or master transaction runs to completion before another transaction can start. This is compatible with the Native Blocks listed above (i.e., ACC, ATAC, and DD), which are all single transaction devices. As suggested by Figure 4-4, the GeodeLink Adapter contains no registers and is strictly speaking, just a bridge.

The MSRs are conceptually separate from the Native Block and GeodeLink Adapter and generally provide overall GeodeLink Device configuration and control. In most designs they are physically separated as shown. There are six standard MSRs that are detailed in Section 4.8 “Standard GeodeLink™ Device MSRs”. All GeodeLink Devices have these standard MSRs. GeodeLink Devices may also incorporate additional MSRs as appropriate.

On the upper right of the figure, the connections between the GeodeLink Adapter and GLIU are illustrated. All of these signals were covered previously in Section 4.1 “GeodeLink™ Architecture Overview”.

The *Clock Control Units (CCU)* are key components in the *Active Hardware Clock Gating (AHCG)* infrastructure. They provide the mechanism for turning off clocks to sections of logic that are *Not Busy*. Furthermore, they take an asynchronous global reset signal and synchronize it to the applicable clock domain.

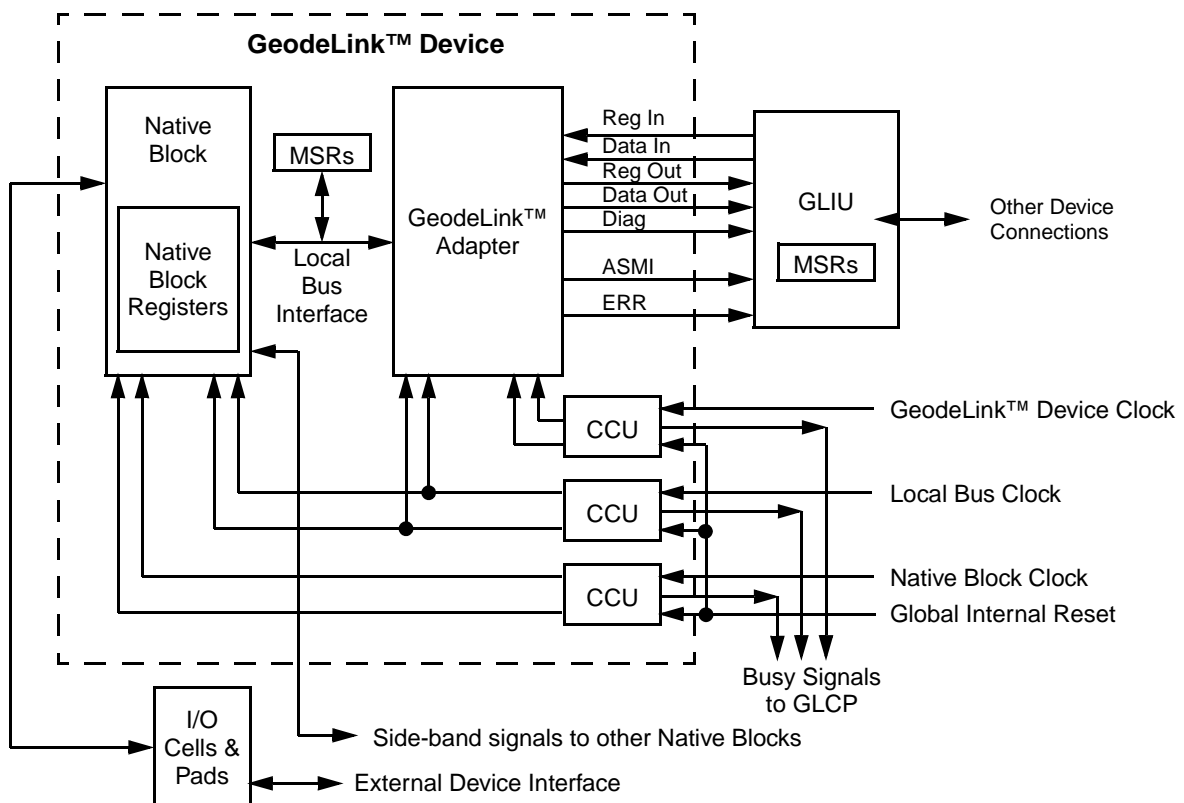


Figure 4-4. Typical GeodeLink™ Device

### 4.4 Embedded PCI Adapter

A GeodeLink Device with an embedded PCI bus is illustrated in Figure 4-5. Note the similarity with Figure 4-4 on page 61. The only difference is the PCI Adapter located in the center of the figure. The PCI Adapter allows a module designed for the PCI bus to be easily embedded in a GeodeLink architecture. It is used to embed the USB core into the GeodeLink architecture.

The PCI Adapter provides the following features:

- Converts GeodeLink Adapter Local bus transactions to/from PCI bus transactions.
- Provides PCI bus configuration transactions from GeodeLink Device MSR transactions.
- Performs PCI bus protocol checking and sets the GeodeLink Device Error Flag if appropriate.

- Capable of responding to both single data phase as well as burst transactions.
- Closely follows the functioning of the GeodeLink Adapter.
- Capable of issuing retries to the master when the Local bus side is taking too long to complete the current transaction.
- Safely handles all the error conditions possible during normal and configuration PCI transactions.

Using the Virtual PCI Configuration Space (see Section 4.1 "GeodeLink™ Architecture Overview" on page 55), the Core BIOS lifts this embedded physical interface into the multi-function PCI configuration space header.

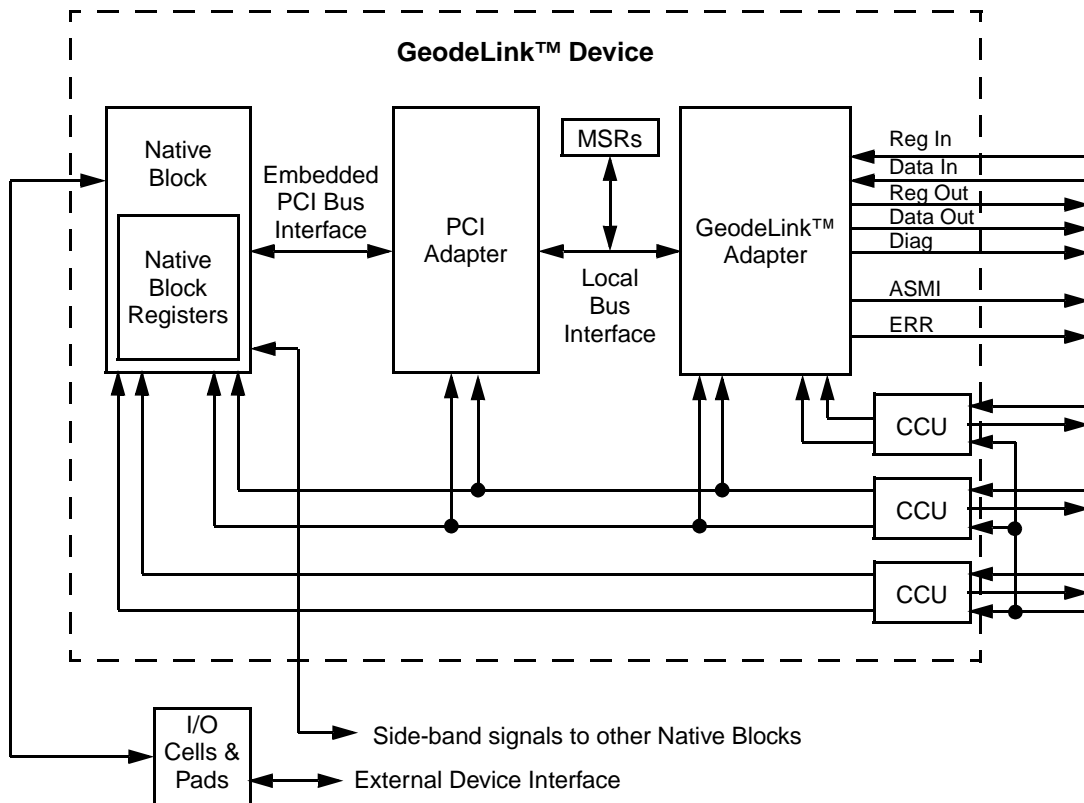


Figure 4-5. GeodeLink™ Device With Embedded PCI Bus

## 4.5 Clock Considerations

### 4.5.1 Clock Domain Definitions

Table 4-3 lists the clock sources and domains for the Geode CS5535 companion device.

**Table 4-3. Clock Sources and Clock Domains**

Component Pin	Domain Name	Description
MHZ66_CLK	ATAC_LB	ATAC Local bus and ATAC core
Inverted MHZ66_CLK (Note 1)	GLIU_GLD	GLIU GeodeLink Device interface and related logic
	GLIU_STAT	GLIU Statistics Counters
	GLPCI_GLD	GLPCI_SB GeodeLink Device interface and related logic
	USBC2_GLD	USBC2 GeodeLink Device interface and related logic
	ATAC_GLD	ATAC GeodeLink Device interface and related logic
	DD_GLD	DD GeodeLink Device interface and related logic
	ACC_GLD	ACC GeodeLink Device interface and related logic
	USBC1_GLD	USBC1 GeodeLink Device interface and related logic
	GLCP_GLD	GLCP GeodeLink Device interface and related logic
MHZ66_CLK divided by two (Note 2)	USBC2_LB	USBC2 Local bus interface and related logic
	ACC_LB	ACC Local bus interface and related logic
	USBC1_LB	USBC1 Local bus interface and related logic
PCI_CLK	GLPCI_TRNA	GLPCI_SB transaction processing
	GLPCI_INTF	GLPCI_SB interface to PCI bus
	GLCP_PCI	GLCP PCI related logic
MHZ48_CLK	USBC2_COR	USBC2 core logic
	USBC1_COR	USBC1 core logic
MHZ48_CLK divided by two (Note 2)	SMB_COR	SMB Controller core logic
	UART1_COR	UART1 core logic
	UART2_COR	UART2 core logic
LPC_CLK	DD_LB	DD Local bus interface and related logic; Includes PIC
	LPC_COR	LPC Controller core logic
	PIT_COR	PIT core logic
	DMA_COR	8237 DMA core logic
AC_CLK	ACC_COR	ACC core logic
MHZ14_CLK (Note 3)	MFGPT_COR_14M	MFGPT core logic 14 MHz clock
	PMC_SLP	Power Management Controller Sleep logic
	PIT_REF	Programmable Interval Timer reference clock
KHZ32_XCI (Note 3)	RTC_COR	RTC core logic
	MFGPT_COR_32K	MFGPT core logic 32 kHz clock
	MFGPT_COR_32K_S	MFGPT 32kHz clock source; Standby power domain
	PMC_STB	PMC Standby logic; Standby power domain
	GPIO_COR	GPIO core logic
	GPIO_COR_S	GPIO core logic; Standby power domain
TCK (Note 3)	TAP_CNTRL	JTAG TAP Controller clock source
(Note 4)	GLCP_DBG	GLCP debug logic

Note 1. The MHZ66\_CLK is first inverted and then fed to all these domains.

Note 2. Each domain receives the referenced clock and performs the divide just before the CCU.

Note 3. This clock differs from other clocks in this table in that this clock does not utilize a CCU nor is it subject to GLCP control or power management control.

Note 4. This logic does not have a fixed clock source. During debug it is switched to the clock domain of interest. It does have a CCU.

## 4.5.2 Clock Controls and Setup

Each of the clock domains listed in Table 4-3 on page 63 is subject to various GLCP controls and status registers except those with “Note 3”. These registers and a brief description of each is provided:

- **GLCP Clock Active (GLCP\_CLKACTIVE), MSR 51700011h:** A 1 indicates the corresponding clock is active. This is a read only register.
- **GLCP Clock Control (GLCP\_CLKOFF), MSR 51700010h:** A 1 indicates the corresponding clock is to be disabled immediately and unconditionally. Not normally used operationally. Debug only.
- **GLCP Clock Mask for Debug Clock Stop Action (GLCP\_CLKDISABLE), MSR 51700012h:** A 1 indicates the corresponding clock is to be disabled by debug logic via a debug event or trigger. Not normally used operationally. Debug only.
- **GLCP Clock Active Mask for Suspend Acknowledge (GLCP\_CLK4ACK), MSR 51700013h:** A 1 indicates the corresponding clock is to be monitored during a power management Sleep operation. When all the clocks with associated 1s go inactive, the GLCP sends a Sleep Acknowledge to the Power Management Controller. This register is used during Sleep sequences and requires the CLK\_DLY\_EN bit in GLCP\_GLB\_PM (MSR 5170000Bh[1]) to be 0.
- **GLCP Clock Mask for Sleep Request (GLCP\_PMCLKDISABLE), MSR 51700009h:** A 1 indicates the corresponding clock is to be disabled unconditionally during a power management Sleep operation. Clocks are disabled when the GLCP completes all of its Sleep Request operations and sends a Sleep Acknowledge to the Power Management Controller.

All of the registers above have the same layout, where each bit is associated with a clock domain. The layout and recommended operating values for the registers is provided in Table 6-73 "Clock Mapping / Operational Settings" on page 536.

### 4.5.2.1 Additional Setup Operations

- **GLCP Debug Clock Control (GLCP\_DBGCLKCTL), MSR 51700016h:** Set all bits to 0. This turns off all clocks to debug features; not needed during normal operation.
- **GLCP Global Power Management Control (GLCP\_GLB\_PM), MSR 5170000Bh:** Set all bits to 0. This disables the use of the fixed delay in GLCP\_CLK\_DIS\_DELAY and enables the use of GLCP\_CLK4ACK.
- **GLCP Clock Disable Delay Value (GLCP\_CLK\_DIS\_DELAY), MSR 51700008h:** Set all bits to 0. Since use of this register is disabled by setting all GLCP\_DBGCLKCTL bits to 0, the actual value of this register is a “don’t care”; it is set here for completeness. If use of GLCP\_CLK\_DIS\_DELAY is desired, set the CLK\_DLY\_EN bit in GLCP\_GLB\_PM (MSR 5170000Bh[1] = 1). This will disable the use of GLCP\_CLK4ACK and shut off the clocks in GLCP\_PMCLKDISABLE after the GLCP\_CLK\_DIS\_DELAY expires. This delay is measured in PCI clock edges.



## 4.6 Reset Considerations

The elements that effect “reset” within the Geode CS5535 companion device are illustrated in Figure 4-6 on page 66. The following points are significant:

- Signals denoted in upper case (i.e., all capitals) are external pins. Signals denoted in lower case are internal signals.
- There are separate resets for the Working power domain (RESET\_WORK#) and the Standby power domain (RESET\_STAND#).
- All elements in the figure are within the Standby power domain and operate off the KHZ32\_CLK.
- The TAP Controller is in the Working power domain, but it may be reset separately from the other Working domain logic.
- Any time the Geode CS5535 companion device is in the Standby state, the Working power domain is unconditionally and immediately driven into reset.
- Any faulted event or external reset input forces the Geode CS5535 companion device into the Standby state.
- External reset (RESET\_OUT#) is always asserted immediately with internal working domain reset. Deassertion is also immediate unless the event is a wakeup from a soft off condition (see Section 5.17.3.4 "Wakeup Events" on page 178 for details). RESET\_OUT# asserts without any clocks but requires the KHZ32\_CLK for the delay and the PCI\_CLK to de-assert.
- IDE\_RESET# is always asserted immediately with internal working domain reset and de-asserts when the ATAC comes out of reset, that is, within a few MHZ66\_CLK edges of internal reset de-assert.
- LVD monitors  $V_{CORE}$  and only asserts power\_good\_working when  $V_{CORE}$  is within normal operating range.
- LVD monitors  $V_{CORE\_VSB}$  and  $V_{IO\_VSB}$  along with RESET\_STAND#. The assertion of power\_good\_standby only occurs when the voltages are within normal operating range and RESET\_STAND# is high, that is, de-asserted.

When power is applied to the Geode CS5535 companion device from a completely cold start, that is, no Standby or Working power, both RESET\_STAND# and RESET\_WORK# are applied. Alternatively, one or both of the reset inputs may be tied to Standby I/O power ( $V_{IO\_VSB}$ ), and the LVD circuit will generate internal Power Good Working and internal Power Good Standby. Assuming the LVD circuit is enabled (LVD\_EN# pin tied low), Power Good Standby will assert until proper Standby voltages have been achieved and RESET\_STAND# has been de-asserted.

RESET\_OUT# is de-asserted synchronous with the low-to-high edge of PCI\_CLK. The de-assertion is delayed from internal\_reset using a counter in the Power Management Controller. This counter is driven by the 32 kHz clock and is located in the Standby power domain. The value of the counter is programmable but defaults to 0x0\_0100 (256 edges). 31.25  $\mu$ s per edge times 256 equals an 8 ms delay. Note this counter default is established by RESET\_STAND# and is not effected by RESET\_WORK#. Therefore, the delay value may be changed and then the system can be reset with the new value.

Note the special consideration for TAP Controller reset. When boundary scan is being performed, internal component operation is not possible due to the scanning signals on the I/Os. Under this condition, it is desirable to hold the component internals in reset while the boundary scan is being performed by the TAP Controller. However, under normal operation, it is desirable to reset the TAP Controller with the other logic in the Working domain during power management sequences.

Achieving these dual goals is accomplished as follows:

### For boundary scan:

- Assert RESET\_STAND#, causing internal power\_good\_standby to go low. This causes the complete component to reset, except for the TAP Controller. Keep this input held low throughout boundary scan operations.
- Assert and de-assert RESET\_WORK# as needed to reset the TAP Controller.

### For normal operation:

- The internal Power Good Standby will be high, meaning the TAP Controller reset asserts any time the Standby state is active or anytime RESET\_WORK# is active.

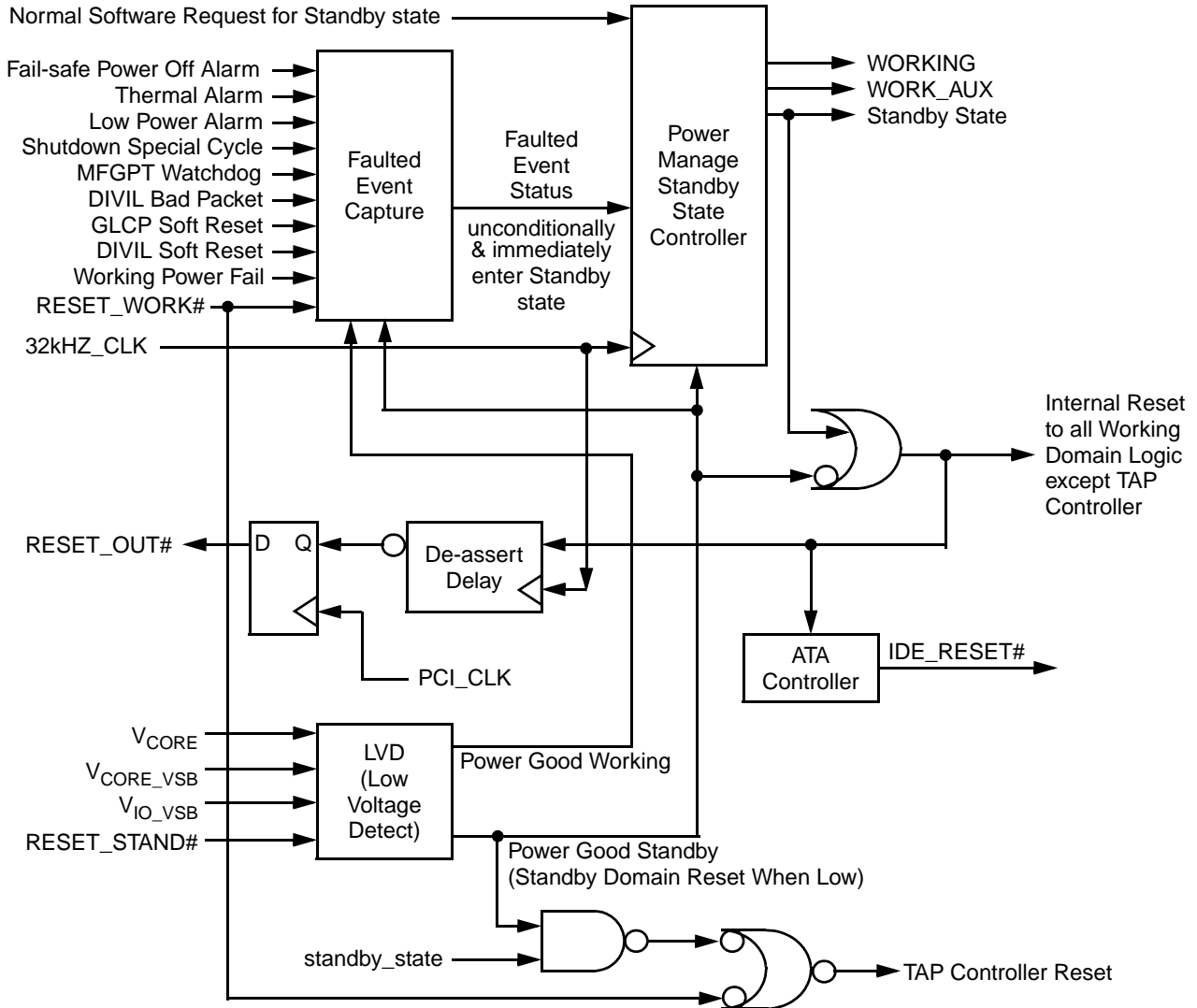


Figure 4-6. Reset Logic

## 4.7 Memory and I/O Map Overview

### 4.7.1 Introduction

There are several places in the Geode CS5535 companion device where addresses are decoded and routed:

- **Physical PCI Bus.** The GLPCI\_SB decodes PCI bus transactions and claims them with a “DEVSEL#” as appropriate. After claiming a transaction, the GLPCI\_SB converts it to a GLIU request packet. It then passes the request to the GLIU. It has no routing control or responsibility beyond this point.
- **GLIU.** The GLIU compares the request addresses against the descriptor settings. It passes the request to the port associated with the compare hit. Each port is connected to a specific GeodeLink Device (see Section 4.1.5 “Topology” on page 57 for port assignment). There are also specific legacy addresses that receive “special” routing beyond the standard descriptor routing mechanisms.
- **Typical GeodeLink Device.** For most GeodeLink Devices, further decoding is minimal. If a device contains only MSRs and a single Native Block (register group) in I/O or memory space, specific bits within the request packet can be used to easily select between the two. If a device contains more than one register group, a Local Base Address Register (LBAR) for each group is used. Like a PCI Base Address Register (BAR), an LBAR compare and hit operation is used to select the desired group.
- **Diverse Device.** The Diverse Device has the same decoding responsibilities as a typical GeodeLink Device. Beyond this programmable LBAR decoding, it has substantial fixed decoding associated with legacy addresses.

### 4.7.2 PCI Bus Decoding

From reset, the GLPCI\_SB does not actively decode any cycle. However, it does subtractively decode all cycles. From reset, any cycle not positively claimed on the PCI bus is converted to a GLIU request and passed to the GLIU.

Using appropriate setup registers, the GLPCI\_SB can be programmed to actively decode selected I/O and memory regions. Other than actively claiming, the “convert” and “pass” operation is the same.

There are control bits in GLPCI\_CTL (MSR 51700010h) to regulate behavior associated with legacy addresses:

- **Bits [12:11]: Legacy I/O Space Active Decode.** These bits control the degree to which the GLPCI\_SB actively claims I/O region 0000h through 03FFh:
  - 00: Subtractive – Claim on fourth clock. (Default.)
  - 01: Slow – Claim on third clock.
  - 10: Medium – Claim on second clock.
- **Bit 13: Reject Primary IDE.** If this bit is set, the GLPCI\_SB will not actively decode the primary IDE addresses of 01F0h/01F7h and 03F6h.

- **Bit 14: Reject Secondary IDE.** If this bit is set, the GLPCI\_SB will not actively decode the secondary IDE addresses of 0170h/0177h and 0376h.
- **Bit 15: Reject DMA High Page Active.** If this bit is set, the GLPCI\_SB will actively decode the I/O range 0480h/048Fh associated with the DMA High Page registers.

For further details on the GLPCI\_CTL register see Section 6.2.2.1 “Global Control (GLPCI\_CTRL)” on page 234.

Lastly, there is an “MSR Access Mailbox” located in PCI Configuration register space. It consists of the following 32-bit registers:

- **MSR Address (PCI Index F4h).** Full MSR routing path in the upper portion plus 14 device address bits in the lower portion.
- **MSR Data Low (PCI Index F8h).** Bits [31:0]: When read, an MSR cycle is generated. The 64-bit read returns the low 32 bits and saves the upper 32 bits for a read to “Data High”. A write holds the value written as the current “Data Low”.
- **MSR Data High (PCI Index FCh).** Bits [63:32]: Reads return the upper 32 bits of the last MSR value read. Writes generate an MSR write cycle using the current value and the “Data Low” value.

For further details on the MSR Access Mailbox see Section 6.2.3 “PCI Configuration Registers” on page 240.

### 4.7.3 GLIU Decoding

From reset, the GLIU passes all request packets to the Diverse Device, except for the legacy primary IDE addresses (01F0h/01F7h and 03F6h), these are passed to the IDE device in the ATAC. There is a GLIU IOD\_SC descriptor to control this primary IDE behavior and it defaults configured (see Section 6.1.4.2 “IOD Swiss Cheese Descriptors (GLIU\_IOD\_SC[x])” on page 226). If this descriptor is disabled, all requests pass to the Diverse Device.

Using appropriate MSR setup registers (descriptors), the GLIU can be programmed to route selected I/O and memory regions to specific GeodeLink Devices. Any memory or I/O address that does not hit one of these regions, subtractively routes to the Diverse Device. Unlike PCI, there is no performance loss associated with being the subtractive port.

Operationally, there are five bus masters within the Geode CS5535 companion device: ATAC, ACC, DD, USBC1, and USBC2. These masters only generate requests to access main memory off the Geode GX processor. Therefore, all their GLIU requests need to be routed to the GLPCI\_SB for presentation to the PCI bus. A set of GLIU P2D\_BM descriptors could be used for this purpose. However, the Geode CS5535 companion device GLIU is uniquely modified to route all requests for the listed masters to the

GLPCI\_SB unconditionally. Therefore, GLIU P2D\_BM settings do not affect packet routing from the listed masters. GLIU descriptors are only used to route requests from the GLPCI\_SB and GLCP.

#### 4.7.4 Legacy Keyboard Emulation

In the Geode CS5535 companion device, there are two USB Controllers and hence two copies of this hardware. The USB control registers are memory mapped. The memory region associated with these registers is relocatable via standard GLIU descriptor MSRs starting at an appropriate base address. The region size is 4 KB (1000h), that is, an offset range of 000h through FFFh. There are four registers called: HceControl, HceInput, HceOutput, and HceStatus. There are no USB control registers above this region.

Special consideration is given to the legacy keyboard emulation control registers normally associated with the USB Controller. This “normal” association is due to the fact that “normally”, the keyboard emulation hardware is physically located with the USB hardware, even though there is no logical association between the two at the hardware level. This “normal” association is driven by industry standard device drivers that group the two register sets in the same region. The keyboard emulation registers are located at the USB base address plus 0100h.

A single copy of the Keyboard Emulation Logic (KEL) hardware is located in the Diverse Device (DD) module, where it can be closely coordinated with a possible real keyboard controller in any of three locations: in either USB Controller

(USBC1 or USBC2), or on the LPC bus. This leaves the problem of the control registers that are physically in the DD, but logically (from the software perspective) in the USB Controller.

A descriptor type is incorporated into the Geode CS5535 companion device to deal with this keyboard issue. It is a variant of the standard “P2D Base Mask Descriptor” (P2D\_BM) called P2D\_BMK (keyboard). A P2D\_BMK descriptor does additional decoding based on Address bit 8. If this bit is low, the hit directs to the USB port. If this bit is high, the hit directs to the subtractive port. There are two P2D\_BMK descriptors in the Geode CS5535 companion device (see Section 6.1.2.2 “P2D Base Mask KEL Descriptors (GLIU\_P2D\_BMK[x])” on page 210).

#### 4.7.5 GeodeLink™ Device Decoding Except Diverse Device

Table 4-4 shows the register space map for all Geode CS5535 companion devices except the Diverse Device. There are no fixed addresses associated with these devices other than the MSRs and the legacy IDE I/O addresses as detailed in Section 4.7.3 “GLIU Decoding” on page 67.

#### 4.7.6 Diverse Device Decoding Except Legacy I/O

The Diverse Device register space map except legacy I/O is shown in Table 4-5 on page 69.

**Table 4-4. Register Space Map Except Diverse Device**

Device	MSR Space (Note 1)	I/O Space	Memory Space
GLPCI_SB	Standard GeodeLink Device MSRs plus GLPCI_SB setup. All MSRs also accessible from PCI Configuration space.	None.	None.
GLCP	Standard GeodeLink Device MSRs plus diagnostic and debug.	None.	None.
GLIU	Standard GeodeLink Device MSRs plus descriptor setup.	Programmable SSMLs.	Programmable SSMLs.
ACC	Standard GeodeLink Device MSRs.	16-byte codec interface plus a 48-byte master interface. All trap registers removed. Generates no SSMLs.	The register space can be here also.
ATAC	Standard GeodeLink Device MSRs plus timing parameters. Bus Master LBAR.	Legacy primary addresses. 16-byte master interface.	None.
USBC1	Standard GeodeLink Device MSRs plus PCI configuration emulation. Must set PCI BAR in USB Controller.	None.	4 KB, but less than 256 bytes used. Keyboard emulation registers to Diverse Device.
USBC2	Standard GeodeLink Device MSRs plus PCI configuration emulation. Must set PCI BAR in USB Controller.	None.	4 KB, but less than 256 bytes used. Keyboard emulation registers to Diverse Device.

Note 1. See Section 4.8 “Standard GeodeLink™ Device MSRs” on page 74 for register descriptions.

**Table 4-5. Diverse Device Register Space Map Except Legacy I/O**

Device	MSR Space (Note 1)	I/O Space	Memory Space
DD	Standard GeodeLink Device MSRs plus: SMB LBAR, ACPI LBAR, PM LBAR, GPIO LBAR, MFGPT LBAR, NAND LBAR, KEL LBAR, KEL LBAR, IRQ Mapper LBAR, Legacy Controls, DMA Mappers, Shadow Registers, LPC Controls, and Memory Mask. NOR Flash address control.	Located by associated LBAR. Defaults disabled. 008 Bytes SMB, 016 Bytes ACPI, 064 Bytes PM Support, 256 Bytes GPIO and ICFs, 064 Bytes MFGPTs, 016 Bytes NAND Flash, and 032 Bytes IRQ Mapper.  All I/O that does not hit one of the items above and does not hit a legacy address, is directed to the LPC bus.	16-Byte KEL Host Controller register set at LBAR. Defaults disabled.  NOR Flash per LBAR.  All other memory accesses are directed to the LPC bus.

Note 1. See Section 4.8 "Standard GeodeLink™ Device MSRs" on page 74 for register descriptions.

### 4.7.7 Legacy I/O Decoding

Table 4-6 details the legacy I/O range for 000h through 4FFh. Each I/O location has a read/write (R/W) capability. Note the following abbreviations:

--- Unknown or can not be determined.

Yes Read and write the register at the indicated location. No shadow required.

WO Write only. Value written can not be read back. Reads do not contain any useful information.

RO Read only. Writes have no effect.

Shw The value written to the register can not be read back via the same I/O location. Read back is accomplished via a “Shadow” register located in MSR space.

Shw@ Reads of the location return a constant or meaningless value.

Shw\$ Reads of the location return a status or some other meaningful information.

Rec Writes to the location are “recorded” and written to the LPC. Reads to the location return the recorded value. The LPC is not read.

**Table 4-6. Legacy I/O: 000h-4FFh**

I/O Addr.	Function	Size	R/W	Comment
000h	Slave DMA Address - Channel 0	8-bit	Yes	16-bit values in two transfers.
001h	Slave DMA Counter - Channel 0	8-bit	Yes	16-bit values in two transfers.
002h	Slave DMA Address - Channel 1	8-bit	Yes	16-bit values in two transfers.
003h	Slave DMA Counter - Channel 1	8-bit	Yes	16-bit values in two transfers.
004h	Slave DMA Address - Channel 2	8-bit	Yes	16-bit values in two transfers.
005h	Slave DMA Counter - Channel 2	8-bit	Yes	16-bit values in two transfers.
006h	Slave DMA Address - Channel 3	8-bit	Yes	16-bit values in two transfers.
007h	Slave DMA Counter - Channel 3	8-bit	Yes	16-bit values in two transfers.
008h	Slave DMA Command/Status - Channels [3:0]	8-bit	Shw\$	
009h	Slave DMA Request - Channels [3:0]	8-bit	WO	Reads return value B2h.
00Ah	Slave DMA Mask - Channels [3:0]	8-bit	Shw@	Reads return value B2h.
00Bh	Slave DMA Mode - Channels [3:0]	8-bit	Shw@	Reads return value B2h.
00Ch	Slave DMA Clear Pointer - Channels [3:0]	8-bit	WO	Reads return value B2h.
00Dh	Slave DMA Reset - Channels [3:0]	8-bit	WO	Reads return value B2h.
00Eh	Slave DMA Reset Mask - Channels [3:0]	8-bit	Shw@	Reads return value B2h.
00Fh	Slave DMA General Mask - Channels [3:0]	8-bit	Shw@	Reads return value B2h.
010h-01Fh	No Specific Usage	---	---	
020h	PIC Master - Command/Status	8-bit	Shw\$	
021h	PIC Master - Command/Status	8-bit	Shw\$	
022h-03Fh	No Specific Usage	---	---	
040h	PIT – System Timer	8-bit	Shw\$	
041h	PIT – Refresh Timer	8-bit	Shw\$	
042h	PIT – Speaker Timer	8-bit	Shw\$	
043h	PIT – Control	8-bit	Shw\$	
044h-05Fh	No Specific Usage	---	---	
060h	Keyboard/Mouse - Data Port	8-bit	Yes	If KEL Memory Offset 100h[0] = 1 (EmulationEnable bit). If MSR 5140001Fh[0] = 1 (SNOOP bit) and KEL Memory Offset 100h[0] = 0 (EmulationEnable bit).
061h	Port B Control	8-bit	Yes	

Table 4-6. Legacy I/O: 000h-4FFh (Continued)

I/O Addr.	Function	Size	R/W	Comment
062h-063h	No Specific Usage	---	---	
064h	Keyboard/Mouse - Command/ Status	8-bit	Yes	If KEL Memory Offset 100h[0] = 1 (EmulationEnable bit). If MSR 5140001Fh[0] = 1 (SNOOP bit) and KEL Memory Offset 100h[0] = 0 (EmulationEnable bit).
065h-06Fh	No Specific Usage	---	---	
070h-071h	RTC RAM Address/Data Port	8-bit	Yes	Options per MSR 51400014h[0]. (Note 1)
072h-073h	High RTC RAM Address/Data Port	8-bit	Yes	Options per MSR 51400014h[1].
074h-077h	No Specific Usage	---	---	
078h-07Fh	No Specific Usage	---	---	
080h	Post Code Display	8-bit	Rec	Write LPC and DMA. Read only DMA.
081h	DMA Channel 2 Low Page	8-bit	Rec	Upper addr bits [23:16]. Write LPC and DMA. Read only DMA.
082h	DMA Channel 3 Low Page			
083h	DMA Channel 1 Low Page			
084h-086h	No Specific Usage	8-bit	Rec	Write LPC and DMA. Read only DMA.
087h	DMA Channel 0 Low Page	8-bit	Rec	Upper addr bits [23:16]. Write LPC and DMA. Read only DMA.
088h	No Specific Usage	8-bit	Rec	Write LPC and DMA. Read only DMA.
089h	DMA Channel 6 Low Page	8-bit	Rec	Upper addr bits [23:16]. Write LPC and DMA. Read only DMA.
08Ah	DMA Channel 7 Low Page			
08B	DMA Channel 5 Low Page			
08Ch-08Eh	No Specific Usage	8-bit	Rec	Write LPC and DMA. Read only DMA.
08Fh	DMA C4 Low Page	8-bit	Rec	Upper addr bits [23:16]. See comment at 080h.
090h-091h	No Specific Usage	---	---	
092h	Port A	8-bit	Yes	If kel_porta_en is enabled, then access Port A; else access LPC.
093h-09Fh	No Specific Usage	---	---	
0A0h	PIC Slave - Command/Status	8-bit	Shw\$	
0A1h	PIC Slave - Command/Status	8-bit	Shw\$	
0A2h-0BFh	No Specific Usage	8-bit	---	
0C0h	Master DMA Address - Channel 4	8-bit	Yes	16-bit values in two transfers.
0C1h	No Specific Usage	8-bit	---	
0C2h	Master DMA Counter - Channel 4	8-bit	Yes	16-bit values in two transfers.
0C3h	No Specific Usage	8-bit	---	
0C4h	Master DMA Address - Channel 5	8-bit	Yes	16-bit values in two transfers.
0C6h	Master DMA Counter - Channel 5	8-bit	Yes	16-bit values in two transfers.

Table 4-6. Legacy I/O: 000h-4FFh (Continued)

I/O Addr.	Function	Size	R/W	Comment
0C7h	No Specific Usage	8-bit	---	
0C8h	Master DMA Address - Channel 6	8-bit	Yes	16-bit values in two transfers.
0CAh	Master DMA Counter - Channel 6	8-bit	Yes	16-bit values in two transfers.
0CBh	No Specific Usage	8-bit	---	
0CCh	Master DMA Address - Channel 7	8-bit	Yes	16-bit values in two transfers.
0CDh	No Specific Usage	8-bit	---	
0CEh	Master DMA Counter - Channel 7	8-bit	Yes	16-bit values in two transfers.
0CFh	No Specific Usage	8-bit	---	
0D0h	Master DMA Command/Status - Channels [7:4]	8-bit	Shw\$	
0D1h	No Specific Usage	8-bit	---	
0D2h	Master DMA Request - Channels [7:4]	8-bit	WO	
0D3h	No Specific Usage	8-bit	---	
0D4h	Master DMA Mask - Channels [7:4]	8-bit	Yes	
0D5h	No Specific Usage	8-bit	---	
0D6h	Master DMA Mode - Channels [7:4]	8-bit	Shw@	
0D7h	No Specific Usage	8-bit	---	
0D8h	Master DMA Clear Pointer - Channels [7:4]	8-bit	WO	
0D9h	No Specific Usage	8-bit	---	
0DAh	Master DMA Reset - Channels [7:4]	8-bit	WO	
0DBh	No Specific Usage	8-bit	---	
0DCh	Master DMA Reset Mask - Channels [7:4]	8-bit	WO	
0DDh	No Specific Usage	8-bit	---	
0DEh	Master DMA General Mask - Channels [7:4]	8-bit	Shw@	
0DFh	No Specific Usage	8-bit	---	
0E0h-2E7h	No Specific Usage	---	---	
2E8h-2EFh	UART/IR - COM4	8-bit	---	MSR bit enables/disables into I/O space.(UART1 MSR 51400014h[18:16], UART2 MSR 51400014h[22:20]). Defaults to LPC.
2F0h-2F7h	No Specific Usage	---	---	
2F8h-2FFh	UART/IR - COM2	8-bit	---	MSR bit enables/disables into I/O space.(UART1 MSR 51400014h[18:16], UART2 MSR 51400014h[22:20]). Defaults to LPC.
300h-36Fh	No Specific Usage	---	---	
370h	Floppy Status R A	8-bit	RO	Second Floppy.
371h	Floppy Status R B	8-bit	RO	Second Floppy.
372h	Floppy Digital Out	8-bit	Shw@	Second Floppy.
373h	No Specific Usage	8-bit	---	
374h	Floppy Cntrl Status	8-bit	RO	Second Floppy.
375h	Floppy Data	8-bit	Yes	Second Floppy.
376h	No Specific Usage	8-bit	---	



Table 4-6. Legacy I/O: 000h-4FFh (Continued)

I/O Addr.	Function	Size	R/W	Comment
377h	Floppy Conf Reg	8-bit	Shw\$	Second Floppy.
378h-3E7h	No Specific Usage	---	---	
3E8h-3EFh	UART/IR - COM3	8-bit	---	MSR bit enables/disables into I/O space.(UART1 MSR 51400014h[18:16], UART2 MSR 51400014h[22:20]). Defaults to LPC.
3F0h	Floppy Status R A	8-bit	RO	First Floppy.
3F1h	Floppy Status R B	8-bit	RO	First Floppy.
3F2h	Floppy Digital Out	8-bit	Shw@	First Floppy.
3F3h	No Specific Usage	8-bit	---	
3F4h	Floppy Cntrl Status	8-bit	RO	First Floppy.
3F5h	Floppy Data	8-bit	Yes	First Floppy.
3F6h	No Specific Usage	8-bit	---	
3F7h	Floppy Conf Reg	8-bit	Shw\$	First Floppy.
3F8h-3FFh	UART/IR - COM1	8-bit	---	MSR bit enables/disables into I/O space.(UART1 MSR 51400014h[18:16], UART2 MSR 51400014h[22:20]). Defaults to LPC.
400h-47Fh	No Specific Usage	---	---	
480h	No Specific Usage	8-bit	WO	Write LPC and DMA. Read only DMA.
481h	DMA Channel 2 High Page	8-bit	Rec	Upper addr bits [31:24]. Write LPC and DMA. Read only DMA.
482h	DMA Channel 3 High Page			
483h	DMA Channel 1 High Page			
484h-486h	No Specific Usage	8-bit	WO	Write LPC and DMA. Read only DMA.
487h	DMA Channel 0 High Page	8-bit	Rec	Upper addr bits [31:24]. Write LPC and DMA. Read only DMA.
488h	No Specific Usage	8-bit	WO	Write LPC and DMA. Read only DMA.
489h	DMA Channel 6 High Page	8-bit	Rec	Upper addr bits [31:24]. Write LPC and DMA. Read only DMA.
48Ah	DMA Channel 7 High Page			
48Bh	DMA Channel 5 High Page			
48Ch-48Eh	No Specific Usage	8-bit	WO	Write LPC and DMA. Read only DMA.
48Fh	DMA Channel 4 High Page	8-bit	Rec	Upper addr bits [31:24]. Write LPC and DMA. Read only DMA.
490h-4CFh	No Specific Usage	---	---	
4D0h	PIC Level/Edge	8-bit	Yes	IRQ0-IRQ 7.
4D1h	PIC Level/Edge	8-bit	Yes	IRQ8-IRQ15.
4D2h-4FFh	No Specific Usage	---	---	

Note 1. The Diverse Device snoops writes to this port and maintains the MSB as NMI enable. When low, NMI is enabled. When high, NMI is disabled. This bit defaults high. Reads of this port return bits [6:0] from the on-chip or off-chip target, while bit 7 is returned from the "maintained" value.

## 4.8 Standard GeodeLink™ Device MSR

All GeodeLink Devices have the following Standard MSRs and are always located at the addresses indicated below from the base address given in Table 4-2 "MSR Addresses from AMD Geode™ GX Processor" on page 60:

- MSR Address 0: GeodeLink Device Capabilities (GLD\_MSR\_CAP)
- MSR Address 1: GeodeLink Device Master Configuration (and GLA Prefetch) (GLD\_MSR\_CONFIG)
- MSR Address 2: GeodeLink Device System Management Interrupt Control (GLD\_MSR\_SMI)
- MSR Address 3: GeodeLink Device Error Control (GLD\_MSR\_ERROR)
- MSR Address 4: GeodeLink Device Power Management (GLD\_MSR\_PM)
- MSR Address 5: GeodeLink Device Diagnostic MSR (GLD\_MSR\_DIAG) (This register is reserved for internal use by AMD and should not be written to.)

### 4.8.1 MSR Address 0: Capabilities

The Capabilities MSR (GLD\_MSR\_CAP) is read only and provides identification information as illustrated Table 4-7.

### 4.8.2 MSR Address 1: Master Configuration

The defined fields in the GeodeLink Device Master Configuration MSR (GLD\_MSR\_CONF) vary depending upon the device. Refer to the appropriate GeodeLink Device register chapter starting in Section 6.0 "Register Descriptions" on page 199 for descriptions.

### 4.8.3 MSR Address 2: SMI Control

Each GeodeLink Device within the Geode CS5535 companion device incorporates System Management Interrupts (SMIs). These SMIs are controlled via the Standard GLD\_MSR\_SMI located at MSR Address 2 within each GeodeLink Device (see Table 4-8). The lower 32 bits of this register contain *Enable* (EN) bits, while the upper 32 bits contain *Flag* (FLAG) bits. The EN and FLAG bits are organized in pairs of (n, n+32). For example: (0,32); (1,33); (2,34); etc. The GLD\_MSR\_SMI is used to control and report *events*.

An *event* is any action or occurrence within the GeodeLink Device requiring processor attention. The FLAG bits are status bits that record events. The EN bits enable events to be recorded. An EN bit must be 1 for an event to be recorded (with the exception of the GLUI and the GLCP - the EN bit must be 0 for an event to be recorded). When an event is recorded, the associated FLAG bit is set to a 1. SMI events are of two types: Asynchronous SMI (ASMI) and Synchronous SMI (SSMI).

Table 4-7. GLD\_MSR\_CAP Bit Descriptions

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies the module.
7:0	REV_ID	<b>Revision ID.</b> Identifies the module revision.

Table 4-8. Standard GLD\_MSR\_SMI Format

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG	SMI_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN	SMI_EN



- 7) NESS remains high because Event[Y] has fired.
- 8) VSA sets EN[n] high. This action sets ASMI[n+1] high again and causes another Geode CS5535 companion device ASMI.
- 9) VSA begins to return to the process interrupted by the original ASMI, but notes SMI into the processor is still asserted and returns to step 3.
- 10) If there was no Event[Y] at any point above, return to the interrupted process.

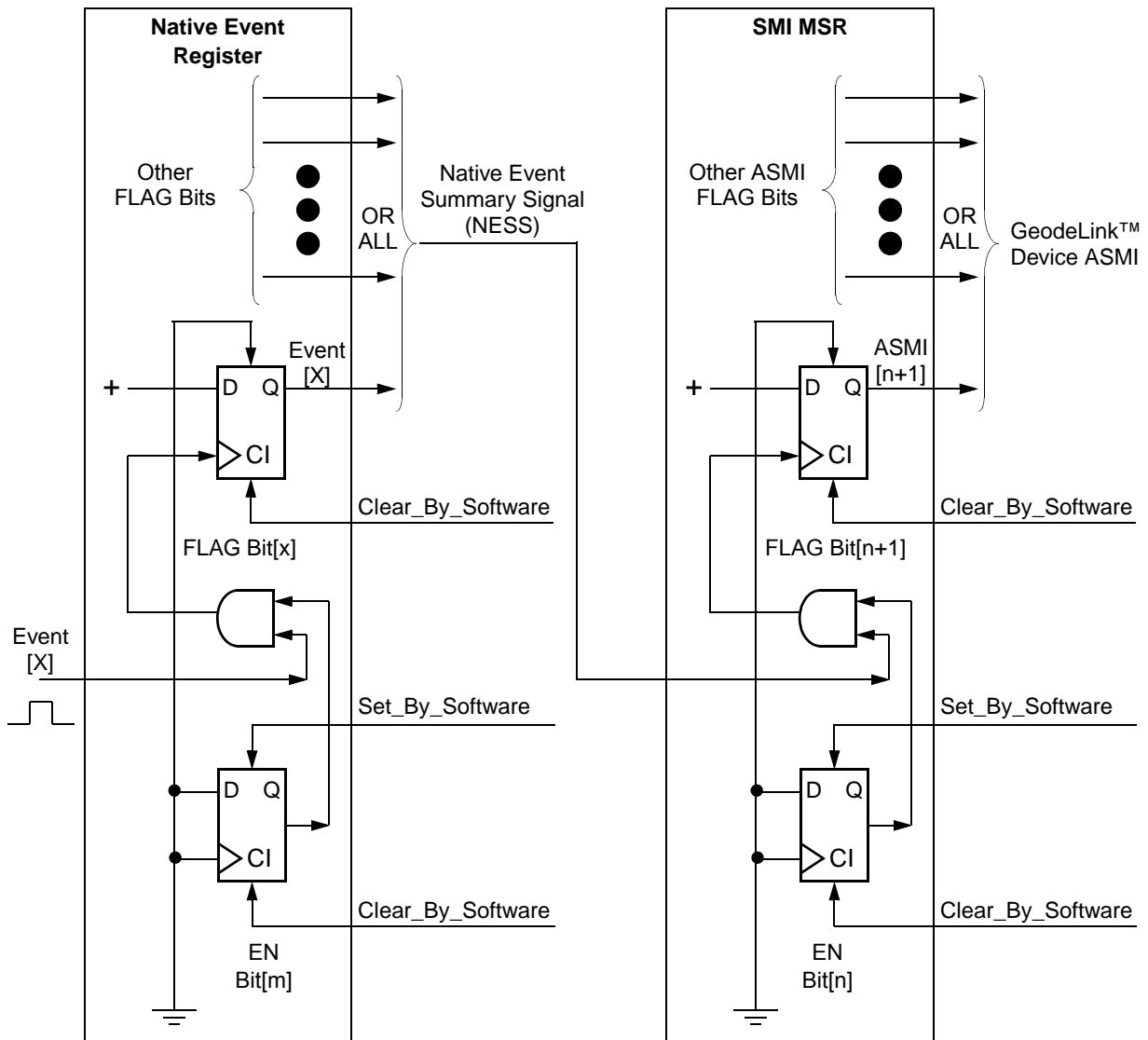
**Note:** Step 5 above could occur at any time between step 2 and step 9, or the Event[Y] could come after step 10. Regardless, the same VSA approach is used in order not to miss any events.

**4.8.3.2 Apparent SSMI**

An SSMI event is associated with an I/O space access to a specific address or range of addresses. If SSMIs are

enabled for the given address, then the hardware traps or blocks access to the target register. The actual register write and/or read operation is not performed. Generally, only write operations are trapped, but there are cases of trapping writes and reads. The Geode CS5535 companion device does not support SSMIs, however, the Geode CS5535 companion device supports a mechanism called “Apparent SSMI” using ASMIs. (Hereafter “Apparent SSMI” is referred to as “SSMI”.)

The Geode CS5535 companion device insures that the ASMI is taken on the I/O instruction boundary. The ASMI reaches the CPU before a target ready is signaled on the PCI bus. This action creates an SSMI because the I/O instruction will not complete before ASMI reaches the CPU. VSA software then examines the GLPCI\_SB\_GLD\_SMI\_MSR to determine if an SSMI has occurred from an I/O trap.



**Figure 4-8. In-Direct ASMI Behavioral Model**

SSMIs are primarily used for hardware emulation and extension. From the perspective of the code on which the trap occurred, everything is normal and done in hardware. However, VSA code generally performs a number of operations to achieve the desired result. This can include returning an appropriate read value to the trapped software. The GLIU is often used to implement SSMI traps.

Any I/O descriptor can be used for this purpose by setting the Destination ID to 0. On a descriptor hit, the GLIU traps the access and sets the SSMI bit in the response packet.

#### 4.8.3.3 ASMI/SSMI Summary

Table 4-9 provides a register summary for the Standard GLD\_SMI\_MSRs.

**Table 4-9. GLD\_MSR\_SMI Summary**

Port #, Device	FLAG Bit	EN Bit	SMI Type	Description
Port 0, GLIU (Note 1)	35	3	ASMI	Statistics Counter 2 Event
	34	2	ASMI	Statistics Counter 1 Event
	33	1	ASMI	Statistics Counter 0 Event
	32	0	SSMI	Descriptor Trap and Illegal Accesses
Port 1, GLPCI_SB	22	6	ASMI	Target Abort Signaled
	21	5	ASMI	Parity Error
	20	4	ASMI	System Error
	19	3	ASMI	EXECP Received
	18	2	ASMI	SSMI Received
	17	1	ASMI	Target Abort Received
	16	0	ASMI	Master Abort Received
Port 2, USBC2	33	1	ASMI	INT by the USB (see PIC for actual source)
	32	0	ASMI	ASMI for the USB (see USB for actual source)
Port 3, ATAC	33	1	ASMI	IRQ for IDE (see PIC for actual source)
	1	0	SSMI	IDE PIO
Port 4, DD (DIVIL)	47	15	SSMI	PMC PM1_CNT
	46	14	SSMI	PMC PM2_CNT
	45	13	ASMI	KEL A20 Keyboard
	44	12	SSMI	8237 DMA Controller access during legacy DMA
	43	11	SSMI	LPC access during legacy DMA
	41	9	SSMI	UART 2 access during legacy DMA
	40	8	SSMI	UART 1 access during legacy DMA
	39	7	ASMI	KEL INIT Port A
	38	6	ASMI	KEL A20 Port A
	37	5	ASMI	KEL INIT Keyboard
	36	4	ASMI	PMC Event (see PMC for actual source)
	35	3	ASMI	Extended PIC Mapper (see PIC for actual source)
	34	2	ASMI	KEL Emulation Event
	33	1	ASMI	Shutdown Special Cycle
	32	0	ASMI	Halt Special Cycle
Port 5, ACC	32	0	SSMI	IRQ from ACC
Port 6, USBC1	33	1	ASMI	INT by the USB (see PIC for actual source)
	32	0	ASMI	ASMI for the USB (see USB for actual source)
Port 7, GLCP (Note 1)	17	1	ASMI	Debug event
	16	0	ASMI	Convert Geode CS5535 companion Global GLIU_Error to ASMI

Note 1. For this device, the listed events are enabled when the EN bit is low. For all other devices, events are enabled when the associated EN bit is high.

**4.8.4 MSR Address 3: Error Control**

Each GeodeLink Device within the Geode CS5535 companion device can generate errors. Furthermore, these errors are controlled via the Standard GeodeLink Device Error MSR (GLD\_MSR\_ERROR) located at MSR Address 3 within each GeodeLink Device. The register is organized just like GLD\_MSR\_SMI, that is, the lower 32 bits contain Enable (EN) bits, while the upper 32 bits contain FLAG (FLAG) bits (see Table 4-8 on page 74). The EN and FLAG bits are organized in pairs of (n, n+32). For example: (0,32); (1,33); (2,34); etc. The Error MSR is used to control and report errors.

The SMI concepts of *direct* asynchronous and synchronous carry over into similar error concepts. However, there is no concept of an *in-direct* error. At each GeodeLink Device, all of the Error FLAG bits are ORed together to form the *Error* signal. The Error is routed through the GLIU where it is ORed with all other device Errors to form the Geode CS5535 companion device Error signal. This signal is routed to the GLCP for debug purposes.

Only the GLIU is capable of generating synchronous errors that utilize the *Exception* (EXCEP) bit of the associated response packet. All other Geode CS5535 companion device GeodeLink Devices only generate asynchronous errors.

**4.8.5 MSR Address 4: Power Management**

All the power management MSRs (GLD\_MSR\_PM) conform to the model illustrated in Table 4-10. The power and I/O mode functions are completely independent other than

sharing the same MSR. The GLD\_MSR\_PM fields have the following definitions:

- Power Mode for Clock Domains:
  - 00: Disable clock gating. Clocks are always on.
  - 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy.
  - 10: Reserved.
  - 11: Reserved.
- I/O Mode (Applies only to GLPCI\_SB and ATAC modules, see Table 4-11 and Table 4-12 for a list of controlled signals):
  - 00: No gating of I/O cells during a Sleep sequence (Default).
  - 01: During a power management Sleep sequence, force inputs to their non-asserted state when PM\_IN\_SLPCTL is enabled.
  - 10: During a power management Sleep sequence, force inputs to their non-asserted state when PM\_IN\_SLPCTL is enabled, and park (force) outputs low when PM\_OUT\_SLPCTL is enabled.
  - 11: Immediately and unconditionally, force inputs to their not asserted state, and park (force) outputs low.

The PMC controls when the PCI/IDE inputs and outputs (listed in Table 4-11 and Table 4-12) are asserted and de-asserted. The PM\_OUT\_SLPCTL (PMS I/O Offset 0Ch) and PM\_IN\_SLPCTL (PMS I/O Offset 20h) registers provide the global control of the PCI/IDE I/Os. The IO\_MODE bits individually control PCI (GLPCI\_SB GLD\_MSR\_PM (MSR 51000004h[49:48]) and IDE (ATAC GLD\_MSR\_PM (MSR 51300004h[49:48]) I/Os.

**Table 4-10. MSR Power Management Model**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IO MODE H	IO MODE G	IO MODE F	IO MODE E	IO MODE D	IO MODE C	IO MODE B	IO MODE A	RSVD																							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMODE15	PMODE14	PMODE13	PMODE12	PMODE11	PMODE10	PMODE9	PMODE8	PMODE7	PMODE6	PMODE5	PMODE4	PMODE3	PMODE2	PMODE1	PMODE0																

**Table 4-11. Sleep Driven PCI Signals**

Signal	Ball No.	Direction
C/BE[3:0]#	R6, T9, U11, U14	Pad driven to 0. Internal logic sees logic 1.
DEVSEL#	R11	Pad driven to 0. Internal logic sees logic 1.
FRAME#	U9	Pad driven to 0. Internal logic sees logic 1.
TRDY#	T10	Pad driven to 0. Internal logic sees logic 1.
IRDY#	R10	Pad driven to 0. Internal logic sees logic 1.
STOP#	T11	Pad driven to 0. Internal logic sees logic 1.
PAR	U10	Pad driven to 0. Internal logic sees logic 1.
REQ#	T1	Pad driven to 0.
GNT#	R1	Pad TRI-STATE. Internal logic sees logic 0.
AD[31:0]	U1, T3, U3, R4, T4, R5, T5, U5, T6, U6, R7, T7, U7, R8, T8, U8, R12, T12, U12, R13, T13, U13, R14, T14, P15, R15, T15, P16, T16, R16, T17, R17	Pad driven to 0.

**Table 4-12. Sleep Driven IDE Signals**

Signal	Ball No.	Direction
IDE_CS[1:0]#	C10, B10	Pad driven to 0.
IDE_IOR0#	B13	Pad driven to 0.
IDE_IOW0#	C13	Pad driven to 0.
IDE_AD[2:0]	B11, A12, A11	Pad driven to 0.
IDE_RESET#	F15	Pad driven to 0.
IDE_RDY0	A13	Pad TRI-STATE. Internal logic sees logic 0.
IDE_DREQ0	A14	Pad TRI-STATE. Internal logic sees logic 0.
IDE_DACK0#	C12	Pad driven to 0.
IDE_DATA[15:0]	C14, B15, B16, A17, C17, D16, D17, E17, E16, E15, D15, B17, C16, C15, A15, B14	Pad driven to 0.

## 4.9 Power Management

Typically the three greatest power consumers in a computing device are the display, the hard drive (if it has one) and the system electronics. The CPU usually consumes the most power of all the system electronic components. Managing power for the first two is relatively straightforward in the sense that they are simply turned on or off. Managing CPU power is more difficult since effective use clock control technology requires effective detection of inactivity, both at a system level and at a code processing level.

Power consumption in a Geode GX processor or other Geode processor based system is managed with the use of both hardware and software. The complete hardware solution is provided for only when the Geode GX processor is combined with the Geode CS5535 companion device.

The processor power consumption is managed primarily through a sophisticated clock stop management technology. The processor also provides the hardware enablers from which the complete power management solution depends on.

Basically two methods are supported to manage power during periods of inactivity. The first method, called activity based power management allows the hardware in the Geode CS5535 companion device to monitor activity to certain devices in the system and if a period of inactivity occurs take some form of power conservation action. This method does not require OS support because this support is handled by SMM software. Simple monitoring of external activity is imperfect as well as inefficient. The second method, called passive power management, requires the OS to take an active role in managing power. AMD supports two application programming interfaces (APIs) to enable power management by the OS: Advanced Power Management (APM) and Advanced Configuration and Power Interface (ACPI). These two methods can be used independent of one another or they can be used together. The extent to which these resources are employed depends on the application and the discretion of the system designer.

The Geode GX processor and Geode CS5535 companion device contain advanced power management features for reducing the power consumption of the processor, companion device and other devices in the system.

### 4.9.1 Power Domains

In order to support power management in periods of inactivity as well as “off” conditions, the Geode CS5535 companion device is divided into three power domains:

- Working Domain - Consists of  $V_{CORE}$  and  $V_{IO}$
- Standby Domain - Consists of  $V_{CORE\_VSB}$  and  $V_{IO\_VSB}$
- RTC Domain - Consists of  $V_{BAT}$

When the system is in an operational mode, all three of the domains are on. In general the power management techniques used while operating produce power savings without user awareness. The performance and usability of the system is unaffected.

When the system is “off” only the standby domain is powered. If desired, the operational design can allow returning the system to the operational point when the system was last “on”. This “instant on” feature is a requirement for many battery powered systems.

If the system has been removed from all power sources the Real Time Clock (RTC) can be kept operating with a small button battery.

All sections of the Geode CS5535 companion device use the Working domain except:

#### Standby Domain

- GPIO[31:24] and associated registers.
- GPIO Input Conditioning Functions 6 and 7.
- GPIO Power Management Events (PMEs) 6 and 7.
- MFGPT[7:6].
- Power Management Controller (PMC) Standby Controller and associated I/O consisting of: WORKING, WORK\_AUX, and RESET\_OUT.
- PMC Standby registers starting at PMS I/O Offset 30h. See Table 6-68 “PM Support Registers Summary” on page 495.

#### RTC Domain

- Real Time Clock

### 4.9.2 ACPI Power Management

ACPI power management is a standardized method to manage power. An overview of the standard is presented here. See Section 5.17 “Power Management Control” on page 172 for a more complete discussion of ACPI support in the Geode CS5535 companion device. See ACPI specification v2.0 for complete details on ACPI. A Geode GX processor/CS5535 companion device system solution can fully support all the requirements in the ACPI specification.

ACPI defines power states from a system perspective down to a device perspective. There are four global system states: G0 through G3. As a subset of the Global system states G0-G2 there are six Sleep states: S0 through S5. Within the sleep states S0-S1 there are five CPU states:

C0-C3 and CT, and three Device states: D0-D2. In a Geode GX processor/CS5535 companion device system design, the optional Sleep state S2, and the CPU states C3 and CT (CPU Throttling) are not supported. See Table 5-34 “Supported ACPI Power Management States” on page 172. Table 5-34 shows how the ACPI power states relate to each other. The table also shows the condition of the power domains and the logic within those domains with respect to the ACPI power states.

### 4.9.3 APM Power Management

Some systems rely solely on an APM (Advanced Power Management) driver for enabling the operating system to power-manage the CPU. APM provides several services which enhance the system power management. It is a reasonable approach to power management but APM has some limitations:

- APM is an OS-specific driver, and may not be available for some operating systems.
- Application support is inconsistent. Some applications in foreground may prevent Idle calls.
- APM does not help with Suspend determination or peripheral power management.

### 4.10 Component Revision ID

The revision ID number of the Geode CS5535 companion device may be read in any of the following places. All return the same value:

- 1) GLCP\_CHIP\_REV\_ID register: MSR 51700017h[7:0].
- 2) PCI Configuration Space Device Revision ID: PCI Index 08h[15:0].
- 3) TAP Controller Revision register: Instruction 8FFFFAh[7:0].

The revision is an 8-bit value. Bits [7:4] indicate major revisions while bits [3:0] indicate minor revisions. For example:

0x11	A1	Value assigned to first manufactured silicon of any new product.
0x12	A2	Minor update to first silicon.
0x21	B1	Major change to first silicon.

For listing of updates, refer to the *AMD Geode™ CS5535 Companion Device Specification Update* document.



# 5

## Module Functional Descriptions

The modules that make up the AMD Geode™ CS5535 companion device (shown in gray in Figure 5-1) are:

- GeodeLink™ Interface Unit
- GeodeLink PCI South Bridge
- GeodeLink Control Processor
- ATA-5 Controller (IDE Controller multiplexed with Flash Interface)
- Universal Serial Bus 1 Controller with Ports 1-1 and 1-2

- Universal Serial Bus 2 Controller with Ports 2-1 and 2-2
- Audio Codec 97 (AC97) Controller
- Diverse Device

The Low Voltage Detect (LVD) circuit is not a GeodeLink Device, but is connected to the Power Management Controller (PMC) for voltage monitoring support.

This section provides a functional description of each module.

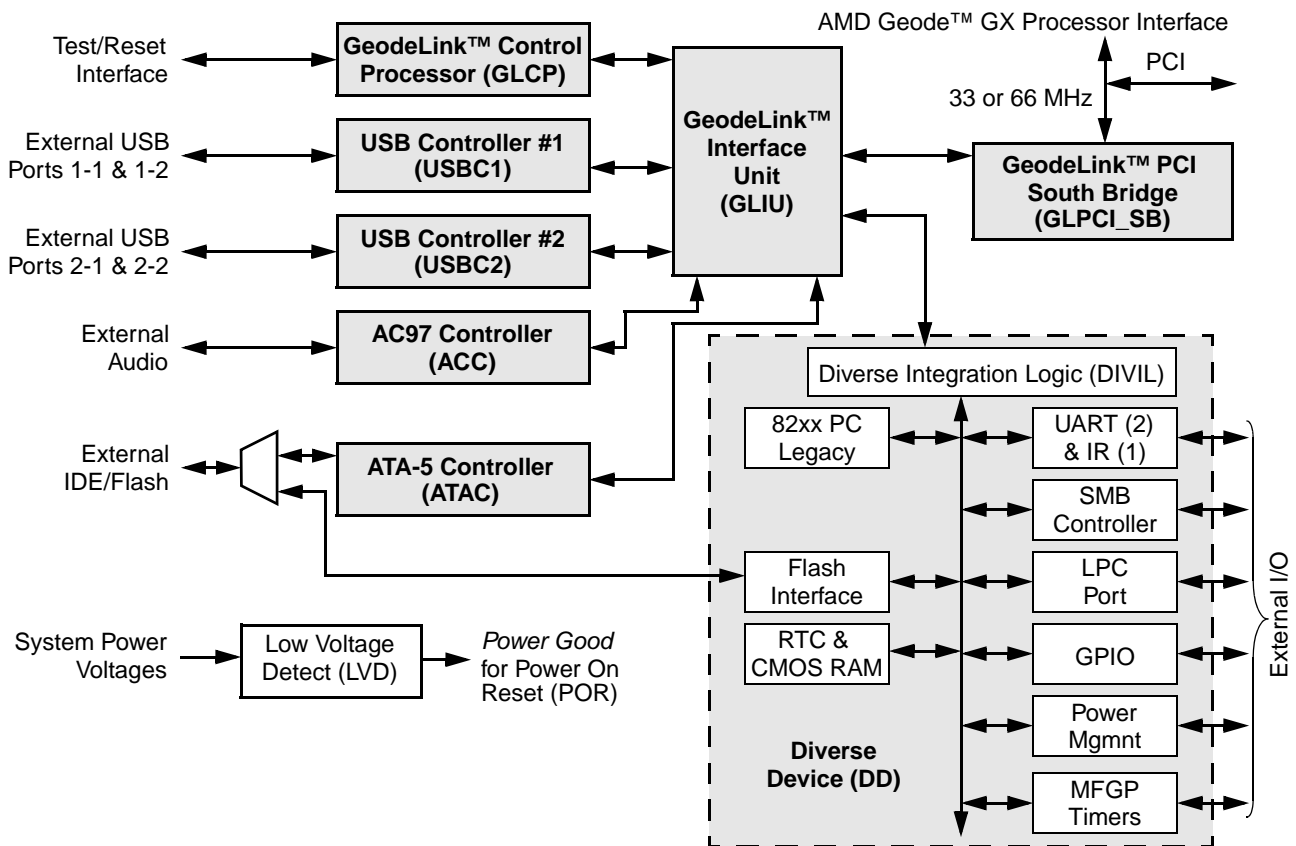


Figure 5-1. Module Block Diagram

## 5.1 GeodeLink™ Interface Unit

Many traditional architectures use buses to connect modules together, usually requiring unique addressing for each register in every module. This requires that some kind of house-keeping be done as new modules are designed and new devices are created from the module set. Module select signals can be used to create the unique addresses, but that can get cumbersome and it requires that the module selects be sourced from some centralized location.

To alleviate this issue, AMD developed an internal bus architecture called GeodeLink. The GeodeLink architecture connects the internal modules of a device using the data channels provided by GeodeLink Interface Units (GLIUs). Using GLIUs, all internal module port addresses are derived from the distinct port that the module is connected to. In this way, a module's Model Specific Registers (MSRs) do not have unique addresses until a device is defined. Also, as defined by the GeodeLink architecture, a module's port address depends on the location of the module sourcing the cycle, or source module.

The Geode CS5535 companion device incorporates one GLIU into its device architecture. Except for the configuration registers that are required for x86 compatibility, all internal registers are accessed through a Model Specific Register (MSR) set. MSRs have a 32-bit address space and a 64-bit data space. The full 64-bit data space is always read or written when accessed.

### 5.1.1 GLIU Port Connections

Table 5-1 shows the GeodeLink Devices connected to each of the seven GLIU ports on the Geode CS5535 companion device.

**Table 5-1. GLIU Port Connections**

Port #	GeodeLink™ Device
1	GeodeLink PCI South Bridge (GLPCI_SB)
2	USB Controller #2 (USBC2)
3	ATA-5 Controller (ATAC)
4	Diverse Device (DD)
5	AC97 Audio Controller (ACC)
6	USB Controller #1 (USBC1)
7	GeodeLink Control Processor (GLCP)

### 5.1.2 Descriptor Summary

Table 5-2 shows the descriptors reserved for each GeodeLink Device.

**Table 5-2. GLIU Descriptors Reserved for GeodeLink™ Devices**

GeodeLink™ Device	Descriptor Type	# of Descriptors	Usage
USBC1	P2D_BMK	1	Do not hit on keyboard emulation registers.
USBC2	P2D_BMK	1	Do not hit on keyboard emulation registers.
ATAC	IOD_BM	1	For IDE master registers.
	IOD_BM	1	Defaults to 1F6h.
	IOD_SC	1	Defaults to 3F6h.
DD	IOD_BM	3	COM ports legacy power management.
	IOD_BM	1	For secondary IDE trapping to 17xh.
	IOD_SC	1	For secondary IDE trapping to 376h.
	IOD_SC	1	Keyboard legacy power management.
	IOD_SC	3	LPC ports legacy power management.
	IOD_SC	1	Floppy legacy power management.
ACC	P2D_BM	1	For memory space registers.
	IOD_BM	1	For I/O space registers.
GLPCI_SB	P2D_BM	1	For master requests to Geode GX processor's GLPCI.
Spares	IOD_BM	2	Provides the possibility to virtualize NAND Flash accesses when the IDE/Flash shared pins are in the "IDE Position". For the "Flash Position", already dedicated IDE descriptors can be used to virtualize.
	IOD_BM	1	
	IOD_SC	1	
	P2D_BM	1	

## 5.2 GeodeLink™ PCI South Bridge

The GeodeLink™ PCI Bus South Bridge (GLPCI\_SB) provides a PCI interface for GeodeLink Device based designs. Its three major functions are:

- 1) Acting as a PCI slave and transforming PCI transactions to GLIU transactions as a GLIU master.
- 2) Acting as a GLIU slave and transforming GLIU transactions to PCI bus transactions as a PCI master.
- 3) Providing a CPU serial interface that conveys system information such as interrupts, SSMI, ASMI, etc.

**Features include:**

- PCI v2.2 compliance. Optional signals PERR#, SERR#, LOCK#, and CLKRUN are not implemented.
- 32-bit, 66 MHz PCI bus operation and 64-bit, 66 MHz GeodeLink Device operation.
- Target support for fast back-to-back transactions.
- Zero wait state operation within a PCI burst.
- MSR access mailbox in PCI configuration space.
- Capable of handling in-bound transactions after RESET\_OUT# + 2 clock cycles.
- Dynamic clock stop/start support for GeodeLink and PCI clock domains via power management features.

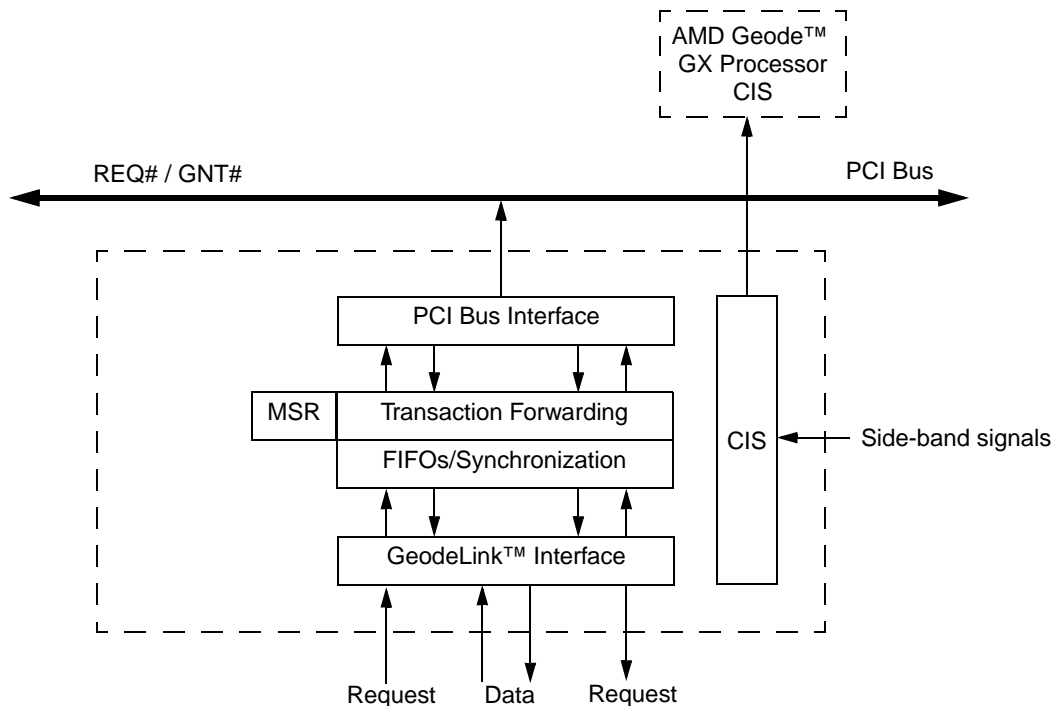
- Programmable IDSEL selection.
- Support active decoding for Legacy I/O space 000h to 3FFh and DMA High Page 480h to 48Fh.
- Support subtractive decode for memory and I/O space.
- Special performance enhancements for fast IDE PIO data transfers.

The GLPCI\_SB module is composed of the following major blocks:

- GeodeLink Interface
- FIFO/Synchronization
- Transaction Forwarding
- PCI Bus Interface
- CPU Interface Serial (CIS)

The GLIU and PCI bus interfaces provide adaptation to the respective buses. The Transaction Forwarding block provides bridging logic. The CIS block provides serial output to the CPU for any change in SSMI and the selected side-band signals. Little endian byte ordering is used on all signal buses.

Figure 5-2 is a block diagram of the GLPCI\_SB module.



**Figure 5-2. GLPCI\_SB Block Diagram**

### 5.2.1 GeodeLink™ Interface

The GeodeLink Interface block provides a thin protocol conversion layer between the transaction forwarding module and the GLIU. It is responsible for multiplexing in-bound write request data with out-bound read response data on the single GLIU data out bus.

### 5.2.2 FIFOs/Synchronization

The FIFO block consists of a collection of in-bound and out-bound FIFOs. Each FIFO provides simple, synchronous interfaces to the reader and to the writer.

The FIFO block also includes synchronization logic for a few non-FIFO related signals and clock gating logic.

### 5.2.3 Transaction Forwarding

The Transaction Forwarding block receives, processes, and forwards transaction requests and responses between the GeodeLink Interface and PCI Bus Interface blocks. It implements the transaction ordering rules. It performs write/read prefetching as needed. It also performs the necessary translation between GLIU and PCI commands. The Transaction Forwarding block also handles the conversion between 64-bit GLIU data paths and 32-bit PCI data paths.

Out-bound transactions are handled in a strongly ordered fashion. All out-bound memory writes are posted. The SEND\_RESPONSE flag is never expected to be set in a memory write and is ignored. Any queued out-bound requests are flushed prior to handling an in-bound read request.

All in-bound memory writes are posted. South bridge master out-bound read request data can pass in-bound writes. Thus, a pending out-bound read request need not be deferred while posted in-bound write data is waiting to be flushed or is in the process of being flushed. The out-bound read request may be performed on the PCI bus at the same time that the in-bound write data is flowing through the GLIU.

When handling an in-bound read request, the intended size of the transfer is unknown. In-bound read requests for non-prefetchable addresses will only fetch the data explicitly indicated in the PCI transaction. Thus, all in-bound read requests made to non-prefetchable addresses will return at most a single 32-bit WORD. In-bound read requests made to prefetchable memory may cause more than a 32-bit WORD to be prefetched. The amount of data prefetched is configurable via the read threshold fields of the Global Control (GLPCI\_CTRL), MSR 51000010h. Multiple read requests may be generated to satisfy the read threshold value.

In-bound read requests may pass posted in-bound write data when there is no address collision between the read request and the address range of the posted write data (different cache lines) and the read address is marked as being prefetchable.

Support IDE data port Read Prefetch when MSR Control register (MSR 51000010h[19:18]) is set to IDE prefetch for performance enhancement. I/O reads to address 1F0h can follow a prefetching behavior. When enabled, the GLPCI\_SB issues GLIU Read Request Packets for this specific address before receiving a request on the PCI bus for it. All PCI accesses to 1F0h must be DWORDs; that is, 4 bytes.

### 5.2.4 PCI Bus Interface

The PCI Bus Interface block provides a protocol conversion layer between the transaction forwarding module and the PCI bus. The master and target portions of this module operate independently. Thus, out-bound write requests and in-bound read responses are effectively multiplexed onto the PCI bus. It includes address decoding logic to recognize distinct address regions for slave operation. Each address region is defined by a starting address, an address mask, and some attached attributes (i.e., memory and/or I/O space indicator, prefetchable, retry/hold, postable memory write, region enable).

The PCIF is responsible for retrying out-bound requests when a slave termination without data is seen on the PCI bus. It also must restart transactions on the PCI that are prematurely ended with a slave termination. This module also always slave terminates in-bound read transactions issued to non-prefetchable regions after a single WORD has been transferred.

### 5.2.5 CPU Interface Serial

The CPU Interface Serial block provides a serial interface to the CPU for side-band signals. From reset, the GLPCI\_SB connects only the SUSP# signal to the serial output. All other signals must be added by programming the CIS mode (MSR 51000010h[4:3]). Any change of the signals selected from the 16 side-band signals will start shifting to the CPU all 20 bits of the CIS register including two START bits (00) and two padding STOP bits (11). Three different modes control the selection of the side-band signals to the CIS shift register.

### 5.2.6 Programmable ID Selection

An ID select register, IDSEL[31:0], is used for programmable ID selection. Only one bit in IDSEL[31:12] is set and used as a chip select (i.e., compared with AD[31:12]) during a PCI configuration write/read. The reset value of the IDSEL register is 02000000h. After reset, the first 32-bit I/O write PCI command (i.e., BE# = 0h) with address 00000000h and one bit set in AD[31:0] is assumed to initialize the IDSEL register. Only data with one bit set in AD[31:0] is considered valid. All other values are ignored and will not change the contents of IDSEL.

### 5.2.7 SSMI and EXCEP Support in GLIU Read/Write Response Packets

If the HCD (Hold for CIS Transfer Disable) bit in GLPCI\_MSR\_CTRL (MSR 51000010h[9]) is set, any inbound memory, I/O, or external MSR read/write response packet will be checked for SSMI and EXCEP bits. If the response packet has the EXCEP bit and/or SSMI bit set, then the GLPCI\_SB will not complete the transaction (it will either issue a Retry or Hold PCI bus) until the CIS transfer completes.

During an out-bound transaction, when the GLPCI\_SB issues a Master Abort, the EXCEP bit in the GLIU response packet is set.

### 5.2.8 Subtractive Decoding

If the SDOFF (Subtractive Decode Off) bit in the GLPCI\_MSR\_CTRL (MSR 51000010h[10]) is cleared (reset value), any PCI transaction, other than Configuration Read/Write, Interrupt Acknowledge, and Special Cycle transactions, not claimed by any device (i.e., not asserting DEVSEL#) within the default active decode cycles (three cycles immediately after FRAME# being asserted) will be accepted by GLPCI\_SB at the fourth clock edge. The Retry condition will be issued for Memory Read, Memory Read Line, Memory Read Multiple (after Initial Latency Timeout), and I/O Read/Write (immediately) and all the required information (command, address and byte enable bits) is stored for the following Delayed Transactions. During Delayed Transactions, the active decode scheme is used. Any address accessed through a subtractive decoding is assumed to be non-prefetchable.

### 5.2.9 Byte Enable Checking in I/O Address Decoding

In any in-bound I/O transaction, the byte enables BE[3:0]# are checked against address bits PCI\_AD[1:0] for valid combinations. If an illegal byte enable pattern is asserted, the GLPCI\_SB will issue a Target Abort. The only exception is when subtractive decode is used. During a subtractive decode, PCI\_AD[1:0] and BE# are passed to the GLIU as is. The IOED (I/O Addressing Error Checking Disable) bit in GLPCI\_MSR\_CTRL (MSR 51000010h[8]) can be set to disable the I/O addressing error checking, where AD[1:0] is ignored and the byte enables are passed to the GLIU.

### 5.2.10 IDE Data Port Read Prefetch

This algorithm issues multiple four byte reads to the IDE data register (1F0h) at the “beginning” of an IDE “read operation”. The hardware continues to read ahead of software read requests until a sector boundary is about to be crossed. The hardware does not read ahead over a sector boundary. Once a software read crosses a sector boundary, the hardware proceeds to read ahead again. Furthermore, the algorithm does not prefetch the last read of a sector because there is the potential that the last sector read will be the last read of the overall “read operation”. On the last read, the status will change to indicate the operation is complete. By not prefetching the last sector read, the

data and status never get out of sync with each other. In PIDE prefetch mode, hardware always makes four byte reads to the IDE data register (1F0h) and supplies the four bytes of read data to IDE read operations ignoring byte enables of the IDE read operation.

### 5.2.11 IDE Data Port Write Posting

The PPIDE (Post Primary IDE) in GLPCI\_MSR\_CTRL (MSR 51000010h[17]) controls post/write on confirmation for I/O writes of address 1F0h (part of primary IDE address). If bit 17 is set, a write is completed immediately on the PCI bus as soon as it is accepted by the GLPCI\_SB. If bit 17 is cleared, an I/O write is completed only after completing the write in the primary IDE space. Default behavior is write on confirmation.

### 5.2.12 Other Typical Slave Write Posting

For each GLPCI\_SB Region Configuration register (0 through 15), if the SPACE bit (bit 32) is programmed for I/O and bit 3 (PF, Prefetchable) is high, post all I/O writes to this region. (See Section 6.2.2.2 “Region 0-15 Configuration MSRs (GLPCI\_R[x])” on page 238 for further details.)

Use of this feature is most appropriate for GPIO “bit banging” in the Diverse Device module. Posting writes on the North Bridge side will not increase performance.

### 5.2.13 Memory Writes with Send Response

Normally memory writes are posted independent of region and independent of decode and legacy/non-legacy address. The USB registers are in memory space and can not be moved to I/O space due to driver compatibility issues. In a Geode GX processor/companion device system, a memory write is posted and a possibility exists that a subsequent I/O write will complete before the posted memory write completes. In order to prevent out of order execution, when a memory write is issued to the GLIU in the Geode CS5535 companion device, the request packet is issued with the send response bit set to serialize the request. I/O writes are not an issue, since the requests packet always has the send response bit set.

### 5.2.14 CPU Interface Serial (CIS)

The CIS provides the system interface between the Geode CS5535 companion device and Geode GX processor. The interface supports several modes to send different combinations of 16-bit side-band signals through the CIS signal (ball P3). The sideband signals are synchronized to the PCI clock through 2-stage latching. Whenever at least one of 16 signals is changed, the serial transfer (using the PCI clock) immediately starts to send the information from the South Bridge to the North Bridge. But, if any bit changes within 20 clocks of any previous change, the later change will not be transmitted during the transfer. Another transfer will start immediately after the conclusion of the transfer due to the subsequent change.

There are three modes of operation for the CIS signal (ball P3). Note that the transmitted polarity may be different than the “generally defined” polarity state:

- Mode A - Non-serialized mode with CIS equivalent to SUSP# (reset mode). Not used in normal operation.

- Mode B - Serialized mode with signals SUSP#, NMI#, Sleep#, and Delayed Sleep#. Not used in normal operation.
- Mode C - Serialized mode with Mode B signals plus SMI#, and INTR#. Used in normal operation.

If the GLPCI\_MSR\_CTRL bit HCD (MSR 51000010h[9]) is set, any in-bound transaction, except in-bound memory writes, will be held for any CIS transfer to complete before claiming completion.

Mode selection is programmed in the GLPCI\_MSR\_CTRL, bits [4:3] (MSR 51000010h).

Table 5-3 lists the serial data with corresponding side-band signals. The serial shift register takes the selected side-band signals as inputs. The signal SMI is the ORed result of the SSMI\_ASMI\_FLAG (SSMI Received Event) bit in GLPCI\_SB\_GLD\_MSR\_SMI (MSR 51000002h[18]) and the side-band signal ASMI. It also serves as a direct output to the processor.

**Table 5-3. CIS Serial Bits Assignment and Descriptions**

Bit Position	Mode B	Mode C	Comment
0 (START)	0	0	Start Bit 0
1 (START)	0	0	Start Bit 1
2	1	1	Reserved
3	1	1	Reserved
4	SUSP#	SUSP#	Sleep Request
5	NMI#	NMI#	Non-Maskable Interrupt
6	Sleep#	Sleep#	Power Management Input Disable
7	Delayed Sleep#	Delayed Sleep#	Power Management Output Disable
8	1	SMI#	Asynchronous SMI or Synchronous SMI
9	1	INTR#	Maskable Interrupt out
10	1	1	Reserved
11	1	1	Reserved
12	1	1	Reserved
13	1	1	Reserved
14	1	1	Reserved
15	1	1	Reserved
16	1	1	Reserved
17	1	1	Reserved
18 (END)	1	1	Stop Bit 0
19 (END)	1	1	Stop Bit 1
<b>Note:</b> Mode A is not listed since it is a non-serialized mode with CIS equivalent to SUSP# (reset mode).			

### 5.2.15 Exception Handling

This section describes how various errors are handled by the PCI Bus Interface block.

Since PERR# is not implemented on the Geode CS5535 companion device or the Geode GX processor, error reporting via this signal is not supported. In a Geode GX processor/CS5535 companion device system, other PCI devices that do have the PERR# pin must have a pull-up.

### 5.2.16 Out-Bound Write Exceptions

When performing an out-bound write on PCI, three errors may occur: master abort, target abort, and parity error. When a master or target abort occurs, the PCI Bus Interface block will flush any stored write data. If enabled, an ASMI is generated. ASMI generation is enabled and reported in GLPCI\_SB\_GLD\_MSR\_SMI (MSR 5100002h). Parity errors are detected and handled by the processor. The failed transaction will not be retried.

### 5.2.17 Out-Bound Read Exceptions

When performing an out-bound read on PCI, three errors may occur: master abort, target abort, and detected parity error. When a master or target abort occurs, the PCI Bus Interface block will return the expected amount of data. If enabled, an ASMI is generated. ASMI generation is enabled and reported in GLPCI\_SB\_GLD\_MSR\_SMI (MSR 5100002h). Parity errors are detected and handled by the processor. The failed transaction will not be retried.

### 5.2.18 In-Bound Write Exceptions

When performing an in-bound write from PCI, two errors may occur: a detected parity error and a GLIU exception. A GLIU exception cannot be relayed back to the originating PCI bus master, because in-bound PCI writes are always posted. When a parity error is detected, an ASMI is generated if it is enabled. ASMI generation is enabled and reported in GLPCI\_SB\_GLD\_MSR\_SMI (MSR 5100002h). However, the corrupted write data will be passed along to the GLIU.

### 5.2.19 In-Bound Read Exceptions

When performing an in-bound read from the GLIU, the EXCEP flag may be set on any received bus-WORD of data. This may be due to an address configuration error caused by software or by an error reported by the source of data. The asynchronous ERR and/or SMI bit will be set by the PCI Bus Interface block and the read data, valid or not, will be passed to the PCI Bus Interface block along with the associated exceptions. The PCI Bus Interface block should simply pass the read response data along to the PCI bus.



### 5.3 AC97 Audio Codec Controller

The primary purpose of the AC97 Audio Codec Controller (ACC) is to stream data between system memory and an AC97 codec (or codecs) using direct memory access (DMA). The AC97 codec supports several channels of digital audio input and output. Hence, the ACC contains several bus mastering DMA engines to support these channels. This method off-loads the CPU, improving system performance. The ACC is connected to the system through the GLIU and all accesses to and from system memory go through the GLIU. The AC97 codec is connected with a serial interface, and all communication with the codec occurs via that interface (see Figure 5-3).

#### Features include:

- AC97 version 2.1 compliant interface to codecs: serial in (x2), serial out, sync out, and bit clock in.
- Eight-channel buffered GLIU mastering interface.
- Support for industry standard 16-bit pulse code modulated (PCM) audio format.
- Support for any AC97 codec with Sample Rate Conversion (SRC).
- Transport for audio data to and from the system memory and AC97 codec.
- Capable of outputting multi-channel 5.1 surround sound (Left, Center, Right, Left Rear, Right Rear, and Low Frequency Effects).

#### Hardware Includes:

- GeodeLink™ Adapter.
- Three 32-bit stereo-buffered bus masters (two for output, one for input).
- Five 16-bit mono-buffered bus masters (three for output, two for input).
- AC Link Control block for interfacing with external AC97 codec(s).

The ACC logic controls the traffic to and from the AC97 codec. For input channels, serial data from the codec is buffered and written to system memory using DMA. For output channels, software-processed data is read from system memory and streamed out serially to the codec.

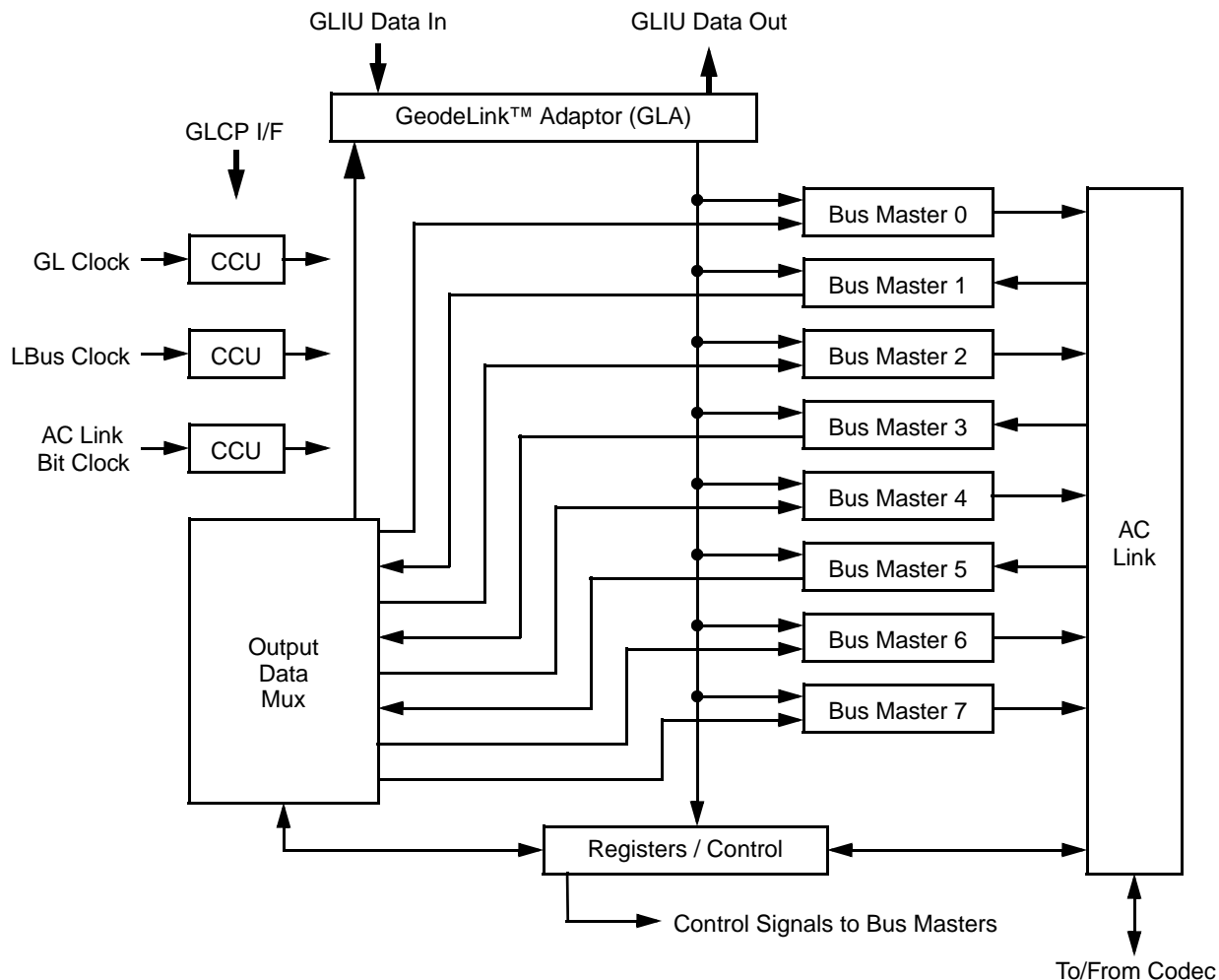


Figure 5-3. ACC Block Diagram

### 5.3.1 Audio Bus Masters

The ACC includes eight bus mastering units (three for input, five for output). Each bus master corresponds to one or two slots in the AC Link transfer protocol (see Section 5.3.4.1 "AC Link Serial Interface Protocol" on page 91). Table 5-4 lists the details for each bus master.

### 5.3.2 Bus Master Audio Configuration Registers

The bus masters must be programmed by software to configure how they transfer data. This is done using their configuration registers. These registers determine whether the bus master is active and what parts of memory they have been assigned to transfer. Status registers allow software to read back information on the state of the bus masters. (See Section 6.3.2 "ACC Native Registers" on page 252 for further details on the Bus Master Audio Configuration registers.)

### 5.3.3 AC Link Overview

The AC Link is the interface between the AC97 codec and the ACC. The interface is AC97 v2.1 compliant. Any AC97 codec that supports Sample Rate Conversion (SRC) can be used with the ACC. See Intel Corporation's "Audio Codec 97" Revision 2.1 component specification for more details.

The AC Link protocol defines an input and output frame consisting of 12 "slots" of data. Each slot contains 20 bits, except slot 0, it contains 16 bits. The SYNC signal is generated by the ACC and defines the beginning of an input and an output frame. The serial clock is generated by the AC97 codec. The AC Link is covered in depth in Section 5.3.4.1 "AC Link Serial Interface Protocol" on page 91. It is important to note that the AC97 codec has its own set of configuration registers that are separate from the ACC. These registers are accessible over the serial link. There are registers in the ACC that provide software with an interface to the AC97 codec registers. (See Section 6.3.2 "ACC Native Registers" on page 252 for register descriptions.)

**Table 5-4. Audio Bus Master Descriptions**

Bus Master	Size	Direction	AC Link Slot(s)	Channel Description
BM0	32-bit (16 bits/channel)	Output to codec	3 (left) and 4 (right)	Left and Right Stereo Main Playback
BM1	32-bit (16 bits/channel)	Input from codec	3 (left) and 4 (right)	Left and Right Stereo Recording
BM2	16-bit	Output to codec	5	Modem Line 1 DAC Output
BM3	16-bit	Input from codec	5	Modem Line 1 ADC Input
BM4	16-bit	Output to codec	6 or 11 (configurable)	Center Channel Playback (slot 6) or Headset Playback (slot 11)
BM5	16-bit	Input from codec	6 or 11 (configurable)	Microphone Record (slot 6) or Headset Record (slot 11)
BM6	32-bit (16 bits/channel)	Output to codec	7 (left) and 8 (right)	Left and Right Surround Playback
BM7	16-bit	Output to codec	9	Low Frequency Effects Playback (LFE)

### 5.3.4 Codec Interface

#### 5.3.4.1 AC Link Serial Interface Protocol

The following figures outline the slot definitions and timing scheme of the AC Link serial protocol.

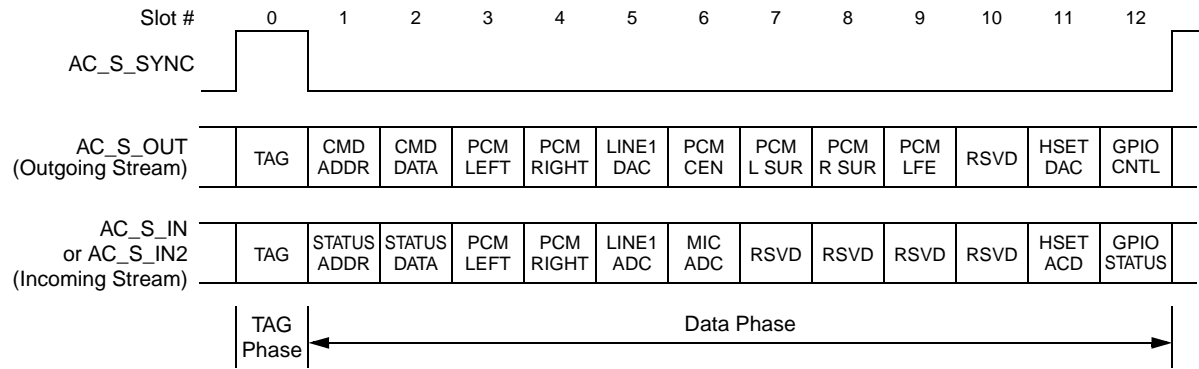


Figure 5-4. AC Link Slot Scheme

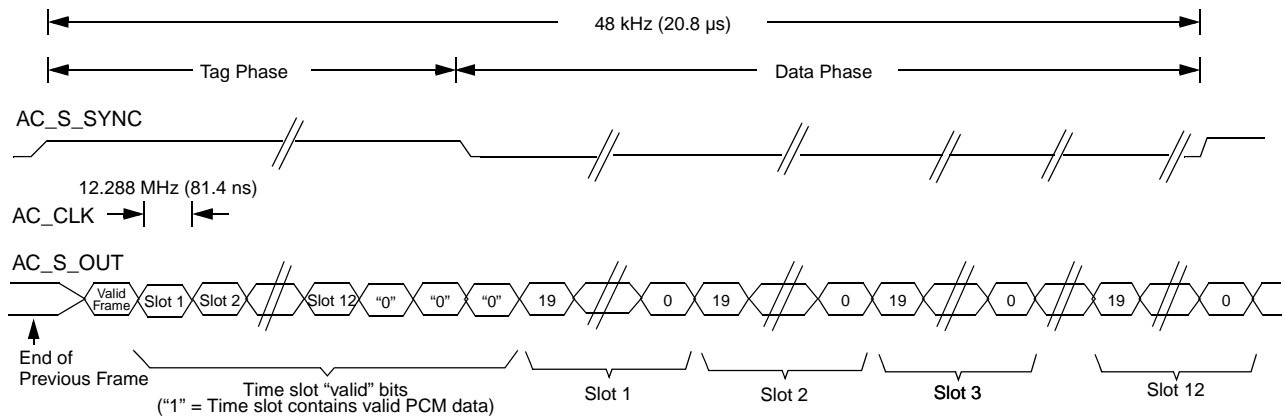


Figure 5-5. AC Link Output Frame

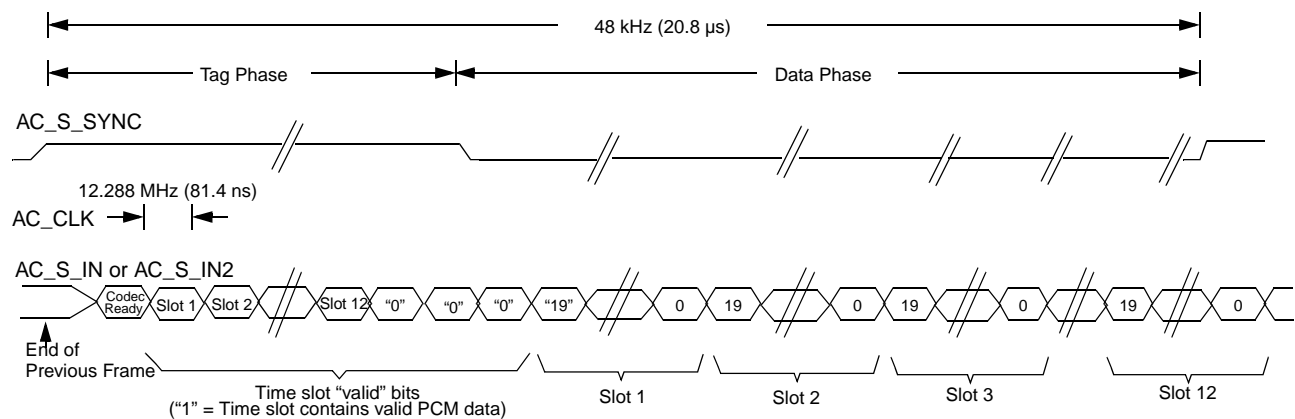


Figure 5-6. AC Link Input Frame

### 5.3.4.2 AC Link Output Frame (AC\_S\_OUT)

The audio output frame data stream corresponds to the time division multiplexed bundles of all digital output data targeting the AC97 codec's DAC inputs and control registers. Each audio output frame contains 13, 20-bit outgoing data slots, except for slot 0, it has 16 bits. Slot 0 is a dedicated slot used for the AC Link protocol.

An audio output frame begins with a low-to-high transition of the AC\_S\_SYNC signal. AC\_S\_SYNC is synchronous to the rising edge of AC\_CLK. The AC97 codec samples the AC\_S\_SYNC on the immediately following falling edge of AC\_CLK. AC\_S\_SYNC is held high for 16 bit clocks. The ACC transmits data on each rising edge of the bit clock, whereas the AC97 codec samples the data on the falling edge of AC\_CLK.

The serial output stream is MSB justified (MSB first) within each slot, and all non-valid bit positions are stuffed with 0s by the AC Link interface module.

#### Slot 0: TAG

This slot is used for AC Link protocol information. The first bit (bit 15) flags the validity of the entire audio frame as a whole. If this bit is 0, all of the remaining bits in the frame should be 0. The next 12 bits indicate the validity of the 12 following slots. The last two bits contain the codec ID for accessing registers of several codecs. When the codec ID is 01, 10, or 11, bits 13 and 14 must always be 0, even if slots 1 and 2 are valid. Slots that are marked invalid by slot 0 should be padded with all 0s (except for slots 1 and 2 while accessing registers of a secondary codec).

bit 15	Frame Valid
bit 14	Slot 1 Valid (primary codec only)
bit 13	Slot 2 Valid (primary codec only)
bits [12:3]	Slot 3-12 Valid bits (bit[12] -> slot 3, bit[11] -> slot 4, bit[10] -> slot 5, ... , bit[3] -> slot 12)
bit 2	Reserved
bits [1:0]	Codec ID field

#### Slot 1: Command Address

The command address is used to access registers within the AC97 codec. The AC97 registers control features and monitor status for AC97 codec functions, including mixer settings and power management as indicated in the *AC97 Codec* specifications.

The control interface architecture supports up to 64 16-bit read/write registers, addressable on even byte boundaries, and reserves support for 64 odd addresses. Audio output frame slot 1 communicates control register address, and write/read command information to the AC97 codec.

bit 19	Read/Write Command (1 = Read, 0 = Write)
bits [18:12]	Control Register Index (64 16-bit locations, addressed on even byte boundaries)
bits [11:0]	Reserved (Stuffed with 0s)

The first bit (MSB) indicates whether the current control transaction is a read or write operation. The following 7 bit positions communicate the targeted control register address.

#### Slot 2: Command Data

The command data slot carries 16-bit control register write data if the current command port operation is a write cycle as indicated by slot 1, bit 19.

bits [19:4]	Control Register Write Data (Stuffed with 0s if current operation is a read)
bits [3:0]	Reserved (Stuffed with 0s)

If the current command port operation is a read, then the entire slot is stuffed with 0s.

#### Slot 3: PCM Playback Left Channel

Outputs the front left audio DAC data (main output) (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 4: PCM Playback Right Channel

Outputs the front right audio DAC data (main output) (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 5: Modem Line 1 DAC

Outputs the modem line 1 DAC data (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 6: PCM Playback Center Channel

Outputs the center channel DAC data (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 7: PCM Playback Left Surround Channel

Outputs the left surround channel DAC data (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 8: PCM Playback Right Surround Channel

Outputs the right surround channel DAC data (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 9: PCM Playback LFE Channel

Outputs the low frequency effects channel DAC data (16-bit resolution, MSB first, unused LSBs = 0).

#### Slot 10: Not used

Slot 10 is not used by the ACC.

### Slot 11: Modem Headset DAC

Outputs the headset DAC data (16-bit resolution, MSB first, unused LSBs = 0).

### Slot 12: GPIO Control

This slot allows the ACC to set the value of the AC97 codec's GPIO output pins.

bits [19:4] Value of the GPIO pins  
(Up to 16 can be implemented)

bits [3:0] Reserved

#### 5.3.4.3 AC Link Input Frame (AC\_S\_IN, AC\_S\_IN2)

The audio input frame data streams correspond to the time division multiplexed bundles of all digital input data coming from the AC97 codec. Each input frame contains 13, 20-bit incoming data slots, except for slot 0; it is 16 bits. Slot 0 is a dedicated slot used for the AC Link protocol.

An audio input frame begins with a low-to-high transition of the AC\_S\_SYNC signal. AC\_S\_SYNC is synchronous to the rising edge of AC\_CLK. The AC97 codec samples the AC\_S\_SYNC signal on the immediately following falling edge of the bit clock. The AC97 codec transmits data on each following rising edge of AC\_CLK. The ACC samples the data on the falling edges of AC\_CLK.

The serial input stream is MSB justified (MSB first) within each slot, and all non-valid bit positions stuffed with zeroes by the AC97 codec.

### Slot 0: TAG

The first bit of the TAG slot (bit 15) is the Codec\_Ready bit. The next 12 bits indicate the validity of the next 12 data slots.

### Slot 1: Status Address / SLOTREQ Bits

The status address is the echo of the register address (index) that was sent to the codec on output slot 1 of the previous output frame. It indicates the address (index) of the register whose data is being returned in slot 2 of the input frame.

bit 19 Reserved  
(Stuffed with 0s)

bits [18:12] Control Register Index  
(Echo of register index for which data is being returned)

bits [11:2] SLOTREQ bits  
(For variable sampling rate)

bits [1:0] Reserved  
(Stuffed with 0s)

The SLOTREQ bits support the variable sample rate signaling protocol. With normal 48 kHz operation, these bits are always zero. When the AC97 codec is configured for a lower sample rate, some output frames will not contain samples because the AC Link always outputs frames at 48 kHz. The SLOTREQ bits serve as the codec's instrument to tell the ACC whether it needs a sample for a given slot

on the next output frame. For each bit: 0 = Send data; 1 = Do NOT send data. If the codec does not request data for a given slot, the ACC should tag that slot invalid and not send PCM data. The mapping between SLOTREQ bits and output slots is given in Table 5-5. The SLOTREQ bits are independent of the validity of slot 1, and slot 1 will only be tagged valid by the codec if it contains a register index.

**Table 5-5. SLOTREQ to Output Slot Mapping**

Bit	Slot Request	Notes
11	Slot 3	Left Channel Out (BM0)
10	Slot 4	Right Channel Out (BM0)
9	Slot 5	Modem Line 1 Out (BM2)
8	Slot 6	Center Out (BM4 if selected)
7	Slot 7	Left Surround Out (BM6)
6	Slot 8	Right Surround Out (BM6)
5	Slot 9	LFE Out (BM7)
4	Slot 10	Not Supported
3	Slot 11	Handset Out (BM4 if selected)
2	Slot 12	Not Supported

### Slot 2: Status Data

The status data slot delivers 16-bit control register read data.

bits [19:4] Control Register Read Data  
(Stuffed with 0s if slot 2 is tagged "invalid" by slot 0)

bits [3:0] Reserved  
(Stuffed with 0s)

### Slot 3: PCM Record Left Channel

Contains the left channel ADC input data (16-bit resolution, MSB first, unused LSBs = 0).

### Slot 4: PCM Record Right Channel

Contains the right channel ADC input data (16-bit resolution, MSB first, unused LSBs = 0).

### Slot 5: Modem Line 1 ADC

Contains the modem line 1 ADC input data (16-bit resolution, MSB first, unused LSBs = 0).

### Slot 6: Optional Microphone Record Data

Contains the microphone ADC input data (16-bit resolution, MSB first, unused LSBs = 0).

### Slots 7-10: Not Used

Slots 7-10 are reserved.

**Slot 11: Modem Headset ADC**

Contains the modem headset ADC input data (16-bit resolution, MSB first, unused LSBs = 0).

**Slot 12: GPIO Status**

This slot returns the pin status of the AC97 codec's GPIO pins (if implemented).

bits [19:4]	Value of the GPIO pins (Up to 16 can be implemented)
bits [3:1]	Reserved
bit 0	GPIO_INT input pin event interrupt (1 = Event; 0 = No Event)

Bit[0] indicates that there was a transition on one of the unmasked codec GPIO pins (see *AC97 Codec Specification v2.1* for details). If the Codec GPIO Interrupt Enable bit is set, then slot 12, bit[0] = 1 will trigger an IRQ and set the Codec GPIO Interrupt Flag bit.

**5.3.5 AC Link Power Management****5.3.5.1 AC Link Power-down**

The AC Link interface signals can be placed in a low power mode by programming the AC97 codec's Power-down Control/Status register. When this is performed, both the AC\_CLK and AC\_S\_IN are brought to a low voltage level by the AC97 codec. This happens immediately following the write to the AC97 codec's Power-down Control/Status register, so no data can be transmitted in slots 3-12 for the frame signaling power-down. After powering down the AC Link, the ACC must keep AC\_S\_SYNC and AC\_S\_OUT low; hence, all the AC Link signals (input and output) are driven low.

AC\_CLK is de-asserted at the same time that bit[4] of slot 2 is being transmitted on the AC Link. This is necessary because the precise time when the codec stops AC\_CLK is not known.

**5.3.5.2 AC Link Wakeup (Warm Reset)**

A warm reset re-activates the AC Link without altering the registers in the AC97 codec. The ACC signals the warm reset by driving AC\_S\_SYNC high for a minimum of 1  $\mu$ s in the absence of the AC\_CLK. This must not occur for a minimum of four audio frame periods following power-down (note that no bit clock is available during this time). AC\_S\_SYNC is normally a synchronous signal to AC\_CLK, but when the AC97 codec is powered down, it is treated as an asynchronous wakeup signal. During wakeup, the AC97 codec does not re-activate the bit clock until AC\_S\_SYNC is driven high (for 1  $\mu$ s minimum) and then low again by the ACC. Once AC\_S\_SYNC is driven low, AC\_CLK is re-asserted.

See "Audio Driver Power-up/down Programming Model" on page 98 for additional power management information and programming details.

**5.3.6 Bus Mastering Buffer Scheme**

Because the bus masters must feed data to the codec without interruption, they require a certain amount of data buffering.

The 32-bit bus masters (stereo) use 24 bytes of buffer space, and the 16-bit bus masters (mono) use 20 bytes of buffer space. A bus master will always do buffer fill/empty requests whenever it can transfer 16 bytes of data. It will attempt to do transfers of 16 bytes on a 16-byte boundary, whenever possible. A bus master may do a transfer of more (if it is just starting, and sufficient buffer space is available) or less than 16 bytes (to bring itself onto a 16-byte boundary). It may also do a transfer of less than 16 bytes if the size of the physical memory region causes it to end on a non-16 byte boundary.

Some important details on how a bus master behaves:

- When an outgoing bus master is enabled, it begins sending data over the AC Link as soon as data is available in its buffer. The slot valid tag for its slot will be asserted beginning with the first audio sample.
- When a bus master is disabled while operating, any data in its buffer is lost. Re-enabling the bus master begins by fetching a PRD.
- If the bus master is paused during recording or playback, the data in its buffer remains there in a frozen state. Once resumed, it continues as if nothing has occurred. If the bus master is playing back data, the output slots corresponding to the bus master are tagged invalid while it is in the paused state.
- If a buffer underrun occurs on an outgoing bus master, the output slots corresponding to the bus master are tagged invalid until data becomes available.
- If a buffer overrun occurs on an incoming bus master, samples coming in on the serial link are tossed away until space becomes available in the bus master's buffer.

### 5.3.7 ACC Software Programming

#### 5.3.7.1 Physical Region Descriptor (PRD) Table Address Register

Before a bus master starts a transfer it must be programmed with a pointer to a Physical Region Descriptor (PRD) table. This is done by writing to the bus master's PRD Table Address register. This pointer sets the starting memory location of the PRD table. The PRDs in the PRD table describe the areas of memory that are used in the data transfer. The table must be aligned on a 4-byte boundary (DWORD aligned).

#### 5.3.7.2 Physical Region Descriptor Format

Each physical memory region to be transferred is described by a PRD as illustrated in Table 5-6. The PRD table must be created in memory by software before the bus master can be activated. When the bus master is enabled by setting its Bus Master Enable bit, data transfer begins, with the PRD table serving as the bus master's "guide" for what to do. The bus master does not cache PRDs.

A PRD entry in the PRD table consists of two DWORDs. The first DWORD contains a 32-bit pointer to a buffer to be transferred (Memory Region Base Address). The second DWORD contains control flags and a 16-bit buffer size value. The maximum amount of audio data that can be transferred for a given PRD is 65534 bytes for mono streams and 65532 bytes for stereo streams.

For stereo streams (bus masters 0, 1, and 6):

Memory Region Base Address and Size should be a multiple of four (DWORD aligned). This ensures an equal number of left and right samples.

For mono streams (bus masters 2, 3, 4, 5, and 7):

Memory Region Base Address and Size should be a multiple of two (WORD aligned).

Descriptions of the control flags are:

- **End of Transfer (EOT)** - If set in a PRD, this bit indicates the last entry in the PRD table. The last entry in a PRD table must have either the EOT bit or the JMP bit set. A PRD can not have both the JMP and EOT bits set. When the bus master reaches an EOT, it stops and clears its Bus Master Enable bit. If software desires an IRQ to be generated with the EOT, it must set the EOP bit and the EOT bit on the last PRD entry.
- **End of Page (EOP)** - If set in a PRD and the bus master has completed the PRD's transfer, the End of Page bit is set (in the IRQ status register) and an IRQ is generated. If a second EOP is reached due to the completion of another PRD before the End of Page bit is cleared, the Bus Master Error bit is set (in the IRQ status register) and the bus master pauses. In this paused condition, reading the IRQ status register clears both the Bus Master Error and the End of Page bits, and the bus master continues.
- **Jump (JMP)** - This PRD is special. If set, the Memory Region Base Address is now the target address of the JMP. The target address of the JMP must point to another PRD. There is no audio data transfer with this PRD. This PRD allows the creation of a looping mechanism. If a PRD table is created with the JMP bit set in the last PRD, the PRD table does not need a PRD with the EOT bit set. A PRD can not have both the JMP and EOT bits set.

**Table 5-6. Physical Region Descriptor (PRD) Format**

DWORD	Byte 3								Byte 2								Byte 1								Byte 0										
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0	Memory Region Base Address [31:0] (Address of Audio Data Buffer)																																		
1	E	E	J	Reserved																								Size [15:0]							
	O	O	M																																
	T	P	P																																

**5.3.7.3 PCM Data Format and Byte Order**

Table 5-7 shows an example of the how PCM audio data is stored in memory (byte order and channel order). Each row represents a byte in memory, with increasing addresses as you go down. The byte order can be configured via the Bus Master Command Register for Intel (little endian) or Motorola (big endian) byte ordering. Changing the byte order ONLY affects how PCM data is interpreted. PRD entries and register contents are always little endian. The two channel format applies to the 32-bit bus masters handling left and right input and output. The single channel format applies to the 16-bit bus masters. The 32-bit bus masters always operate on stereo data, and the 16-bit bus masters always operate on mono data. Since there is no special mode for playing monaural sound through the main channels (left and right), it is the responsibility of the software to create stereo PCM data with identical samples for the left and right channels to effectively output monaural sound.

**5.3.7.4 Programming Model**

**Audio Playback/Record**

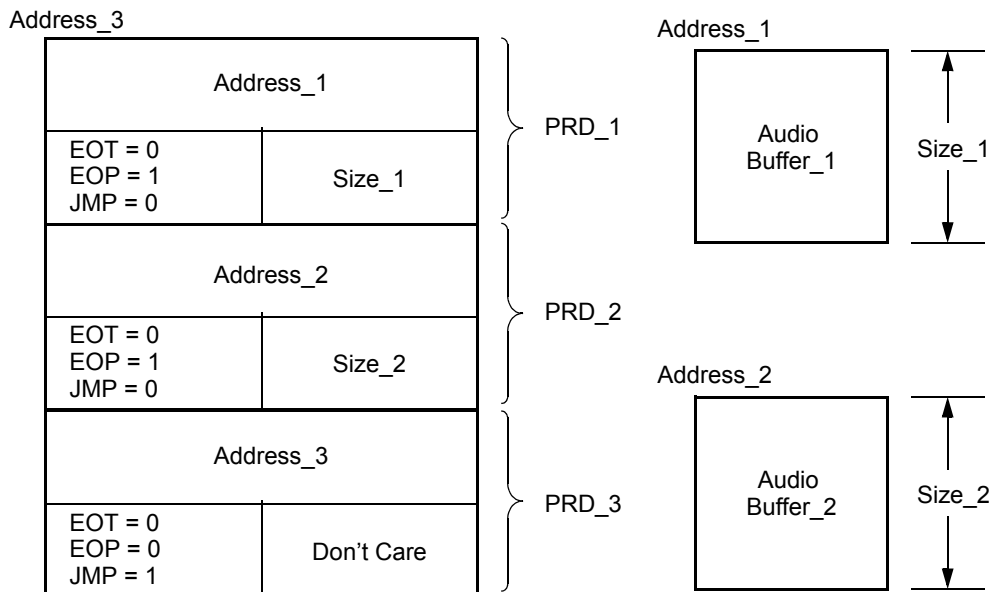
The following discussion explains, in steps, how to initiate and maintain a bus master transfer between memory and an audio slave device. In the steps, the reference to **Example** refers to Figure 5-7:

- 1) Software creates a PRD table in system memory. The last PRD entry in a PRD table must have the EOT or JMP bit set.

**Example** - Assume the data is outbound. There are three PRDs in the example PRD table. The first two PRDs (PRD\_1, PRD\_2) have only the EOP bit set. The last PRD (PRD\_3) has only the JMP bit set. This example creates a PRD loop.

**Table 5-7. PCM Data format (Byte and Channel Ordering)**

2 Channel, Little Endian			1 Channel, Little Endian		2 Channel, Big Endian			1 Channel, Big Endian	
Sample	Channel	Byte	Sample	Byte	Sample	Channel	Byte	Sample	Byte
0	Left	Low	0	Low	0	Left	High	0	High
0	Left	High	0	High	0	Left	Low	0	Low
0	Right	Low	1	Low	0	Right	High	1	High
0	Right	High	1	High	0	Right	Low	1	Low
1	Left	Low	2	Low	1	Left	High	2	High
1	Left	High	2	High	1	Left	Low	2	Low
1	Right	Low	3	Low	1	Right	High	3	High
1	Right	High	3	High	1	Right	Low	3	Low



**Figure 5-7. ACC PRD Table Example**



- 2) Software loads the starting address of the PRD table by programming the PRD Table Address register.

**Example** - Program the PRD Table Address register with Address\_3.

- 3) Software must fill the buffers pointed to by the PRDs with audio data. It is not absolutely necessary to fill the buffers; however, the buffer filling process must stay ahead of the buffer emptying. The simplest way to do this is by using the EOP flags to generate an interrupt when an audio buffer is empty.

**Example** - Fill Audio Buffer\_1 and Audio Buffer\_2. Ensure that an interrupt service routine is assigned to the audio interrupt.

- 4) Read the IRQ Status register to clear the Bus Master Error and End of Page flags (if set).

Program the AC97 codec properly to receive audio data (mixer settings, etc.).

Engage the bus master by setting the Bus Master Enable bit.

The bus master reads the PRD entry pointed to by the PRD Table Address register. Using the address from the PRD, it begins the audio transfer. The PRD Table Address register is incremented by eight.

**Example** - The bus master is now properly programmed to transfer Audio Buffer\_1 to a specific slot(s) in the AC97 interface.

- 5) The bus master transfers data from memory and sends it to the AC97 Codec. At the completion of each PRD, the bus master's next response depends on the settings of the flags in the PRD.

**Example** - After transferring the data described by PRD\_1, an interrupt is generated because the EOP bit is set, and the bus master continues on to PRD\_2. The interrupt service routine reads the Second Level Audio IRQ Status register to determine which bus master to service. It refills Audio Buffer\_1 and then reads the bus master's IRQ Status register to clear the End of Page flag and the interrupt.

After transferring the data described by PRD\_2, another interrupt is generated because the EOP bit is set, and the bus master continues on to PRD\_3. The interrupt service routine reads the Second Level Audio IRQ Status register to determine which bus master to service. It refills Audio Buffer\_2 and then reads the bus master's IRQ Status register to clear the End of Page flag and the interrupt.

PRD\_3 has the JMP bit set. This means the bus master uses the address stored in PRD\_3 (Address\_3) to locate the next PRD. It does not use the address in the PRD Table Address register to get the next PRD. Since Address\_3 is the location of PRD\_1, the bus master has looped the PRD table. No interrupt is generated for PRD\_3.

Pausing the bus master can be accomplished by setting the Bus Master Pause bit in its control register. The bus master stops immediately on the current sample being processed. Upon resuming, the bus master (clearing the Bus Master Pause bit), resumes on the exact sample where it left off.

The bus master can be stopped in the middle of a transfer by clearing the Bus Master Enable bit in its control register. In this case, the bus master will not remember what sample it left off on. If it is re-enabled, it will begin by reading the PRD entry pointed to by its PRD Table Address register. If software does not re-initialize this pointer, it will be pointing to the PRD entry immediately following the PRD entry that was being processed. This may be an invalid condition if the bus master was disabled while processing the last PRD in a PRD table (PRD Table Address register pointing to memory beyond the table).

Note that if the Bus Master Error bit is set, the interrupt service routine should refill two buffers instead of one, because a previous interrupt was missed (unless it was intentionally missed). For this to work correctly, the service routine should read the Second Level Audio IRQ Status register, fill the buffer of the bus master needing service, read the bus master's IRQ Status register, and then fill the next buffer if the Bus Master Error bit was set. Failing to fill the first buffer before reading the IRQ Status register would possibly resume the bus master too early and result in sound being played twice or data being overwritten (if recording).

### Codec Register Access

The ACC provides a set of registers that serve as an interface to the AC97 codec's registers. The Codec Command register allows software to initiate a read or a write of a codec register. The Codec Status register allows software to read back the data from the codec after a read operation has completed. Since the AC Link runs very slow relative to core CPU speed (and therefore software speed), it is necessary for software to wait between issuing commands to the codec.

For register reads, software specifies a command address and sets both the read/write flag and the Codec Command New flag in the Codec Control register. Software must then wait for the Codec Status New bit to be set before using the returned status data in the Codec Status register. Before issuing another read command, software must wait for the Codec Command New flag to be cleared by hardware. (Note: Codec Command New will clear before Codec Status New is set; therefore, a second read can be issued before the result of the current read is returned).

For register writes, software specifies a command address and command data using the Codec Control register. At the same time it must set the Codec Command New flag. Before issuing another read or write, software must wait for the Codec Command New flag to clear. See Section 6.3 "AC97 Audio Codec Controller Register Descriptions" for details on the Codec register interface.

### Audio Driver Power-up/down Programming Model

The ACC contains Machine Specific Registers (MSRs) that relate to a very low level power management scheme, but are discrete from the power management features of the codec and the device driver programming model. This section covers the power management features for the device driver.

See Section 5.3.5 "AC Link Power Management" on page 94 for power management hardware details.

The following sections outline how the device driver should perform power management.

#### Power-down Procedure

- 1) Disable or pause all bus masters using their bus master command register.
- 2) It may be necessary to determine if a second codec is being used, and if so, verify that the power-down Semaphore for Secondary Codec bit is set before proceeding (to insure that the modem driver has prepared the second codec for power-down, if necessary).
- 3) Using the Codec Control register, access the primary codec's registers and program the codec to power-down. Also, simultaneously write to the AC Link Shutdown bit in the Codec Control register (ACC I/O Offset 0Ch[18]).
- 4) The ACC and codec will power-down once the command is received by the codec. All of the contents of the ACC and codec registers are preserved during the power-down state.
- 5) If you wish to enable the GPIO wakeup interrupt, wait for an additional two audio frame periods (42  $\mu$ s) before setting the GPIO Wakeup Interrupt Enable bit (ACC I/O Offset 00h[29]). Failure to wait will cause false interrupt events to occur.

#### Power-up Procedure

- 1) If GPIO Wakeup Interrupt Enable (ACC I/O Offset 00h[29]) was set in the power-down procedure, it will automatically be disabled upon power-up.
- 2) Set the AC Link Warm Reset bit in the Codec Control register (ACC I/O Offset 0Ch[17]). This will initiate the warm reset sequence.
- 3) Wait for the Codec Ready bit(s) in the Codec Status register (ACC I/O Offset 08h[23:22]) to be asserted before accessing any codec features or enabling any bus masters.

#### Notes:

- 1) If the GPIO Wakeup Interrupt Enable (ACC I/O Offset 00h[29]) is set, and an interrupt occurs, it is detected and fired, but the interrupt does not wakeup the codec and ACC. The hardware will only wakeup if the software responds to the interrupt and performs the power-up procedure.
- 2) Once software has issued a power-down, it must not perform the power-up procedure for at least six audio frame periods (about 0.125 ms or 125  $\mu$ s). Doing so could lock up the codec or ACC.
- 3) If the system has cut off power to the codec and restarted it, it is not necessary to initiate a warm reset. The AC Link Shutdown should be cleared manually to restart the operation of the AC Link.

## 5.4 ATA-5 Controller

The hard disk controller is an ATA-5 compatible IDE controller (ATAC). This controller supports UDMA/66, MDMA, and PIO modes. The controller can support one channel (two devices).

The IDE interface provides a variety of features to optimize system performance, including 32-bit disk access, post write buffers, bus master, MDMA, look-ahead read buffer, and prefetch mechanism.

The IDE interface timing is completely programmable. Timing control covers the command active and recover pulse widths, and command block register accesses. The IDE data-transfer speed for each device on each channel can be independently programmed allowing high-speed IDE peripherals to coexist on the same channel as older, compatible devices. Faster devices must be ATA-5 compatible.

The ATAC also provides a software-accessible buffered reset signal to the IDE drive. The IDE\_RST# signal is driven low during system reset and can be driven low or high as needed for device power-off conditions.

### Features include:

- ATA5-compliant IDE controller
- Supports PIO (mode 0 to 4), MDMA (mode 0 to 2), and UDMA (mode 0 to 4)
- Supports one channel, two devices
- Allows independent timing programming for each device

### 5.4.1 PIO Modes

The IDE data port transaction latency consists of address latency, asserted latency, and recovery latency. Address latency occurs when a PCI master cycle targeting the IDE data port is decoded, and the IDE\_ADDR[2:0] and IDE\_CS# lines are not set up. Address latency provides the setup time for the IDE\_ADDR[2:0] and IDE\_CS# lines prior to IDE\_IOR# and IDE\_IOW#.

Asserted latency consists of the I/O command strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface without violating minimum cycle periods for the IDE interface.

If IDE\_IORDY is asserted when the initial sample point is reached, no wait states are added to the command strobe assertion length. If IDE\_IORDY is negated when the initial sample point is reached, additional wait states are added.

Recovery latency occurs after the IDE data port transactions have completed. It provides hold time on the IDE\_ADDR[2:0] and IDE\_CS# lines with respect to the read and write strobes (IDE\_IOR# and IDE\_IOW#).

The PIO portion of the IDE registers is enabled through:

- Channel 0 Drive 0 PIO (ATAC\_CH0D0\_PIO) (MSR 51300020h)

- Channel 0 Drive 1 PIO (ATAC\_CH0D1\_PIO) (MSR 51300022h)

The IDE channel and devices can be individually programmed to select the proper address setup time, asserted time, and recovery time.

The bit formats for these registers are shown in Section 6.4.3 "ATAC Native Registers" on page 274. Note that there are different bit formats for each of the PIO programming registers depending on the operating format selected: Format 0 or Format 1.

Channel 0 Drive 0/1 - The DMA register (MSR 51300021h/51300023h) sets the format of the PIO register. If bit 31 = 0, Format 0 is used and it selects the slowest PIO mode (bits [19:16]) for commands. If bit 31 = 1, Format 1 is used and it allows independent control of command and data.

Also listed in the bit formats are recommended values for the different PIO modes. Note that these values are only recommended settings and are not 100% tested. Maximum PIO data throughput is achieved when the PIO data port is read/written using 32-bit accesses.

### 5.4.2 Bus Master Mode

An IDE bus master is provided to perform the data transfers for the IDE channel. The ATAC off-loads the CPU and improves system performance.

The bus master mode programming interface is an extension of the standard IDE programming model. This means that devices can always be dealt with using the standard IDE programming model, with the master mode functionality used when the appropriate driver and devices are present. Master operation is designed to work with any IDE device that supports DMA transfers on the IDE bus. Devices that work in PIO mode can only use the standard IDE programming model.

The IDE bus master uses a simple scatter/gather mechanism, allowing large transfer blocks to be scattered to or gathered from memory. This cuts down on the number of interrupts to and interactions with the CPU.

#### 5.4.2.1 Physical Region Descriptor Table Address

Before the controller starts a master transfer it is given a pointer to a Physical Region Descriptor Table. This pointer sets the starting memory location of the Physical Region Descriptors (PRDs). The PRDs describe the areas of memory that are used in the data transfer. The PRDs must be aligned on a 4-byte boundary and the table cannot cross a 64 KB boundary in memory.

#### 5.4.2.2 IDE Bus Master Registers

The IDE Bus Master registers have an IDE Bus Master Command register and Bus Master Status register. These registers can be accessed by byte, WORD, or DWORD.

**5.4.2.3 Physical Region Descriptor Format**

Each physical memory region to be transferred is described by a Physical Region Descriptor (PRD) as illustrated in Table 5-8. When the bus master is enabled (Command register bit 0 = 1), data transfer proceeds until each PRD in the PRD table has been transferred. The bus master does not cache PRDs.

The PRD table consists of two DWORDs. The first DWORD contains a 32-bit pointer to a buffer to be transferred. This pointer must be WORD aligned. The second DWORD contains the size (16 bits) of the buffer and the EOT (End Of Table) flag. The size must be in multiples of one WORD (2 bytes) or zero (meaning a 64 KB transfer). The EOT bit (bit 31) must be set to indicate the last PRD in the PRD table.

**5.4.2.4 Programming Model**

The following steps explain how to initiate and maintain a bus master transfer between memory and an IDE device:

- 1) Software creates a PRD table in system memory. Each PRD entry is 8 bytes long, consisting of a base address pointer and buffer size. The maximum data that can be transferred from a PRD entry is 64 KB. A PRD table must be aligned on a 4-byte boundary. The last PRD in a PRD table must have the EOT bit set.

- 2) Software loads the starting address of the PRD table by programming the PRD Table Address register.
- 3) Software must fill the buffers pointed to by the PRDs with IDE data.
- 4) Write 1 to the Bus Master Interrupt bit and Bus Master Error (Status register bits 2 and 1) to clear the bits.
- 5) Set the correct direction to the Read or Write Control bit (Command register bit 3).

Engage the bus master by writing a 1 to the Bus Master Control bit (Command register bit 0).

The bus master reads the PRD entry pointed to by the PRD Table Address register and increments the address by 08h to point to the next PRD. The transfer begins.

- 6) The bus master transfers data to/from memory responding to bus master requests from the IDE device until all PRD entries are serviced.
- 7) The IDE device signals an interrupt once its programmed data count has been transferred.
- 8) In response to the interrupt, software resets the Bus Master Control bit in the Command register. It then reads the status of the controller and IDE device to determine if the transfer is successful.

**Table 5-8. Physical Region Descriptor Format**

DWORD	Byte 3								Byte 2								Byte 1								Byte 0							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Memory Region Physical Base Address [31:1] (IDE Data Buffer)																															0
1	EOT	Reserved															Size [15:1]															0

#### 5.4.2.5 UDMA/66 Mode

The ATAC supports UDMA/66. It utilizes the standard IDE bus master functionality to interface, initiate, and control the transfer. The UDMA/66 definition also incorporates a Cyclic Redundancy Checking (CRC) error checking protocol to detect errors.

The UDMA/66 protocol requires no extra signal pins on the IDE connector. The ATAC redefines three standard IDE control signals when in UDMA/66 mode. These definitions are shown in Table 5-9.

**Table 5-9. UDMA/66 Signal Definitions**

IDE Channel Signal	UDMA/66 Read Cycle	UDMA/66 Write Cycle
IDE_IOW#	STOP	STOP
IDE_IOR#	DMARDY#	STROBE
IDE_IORDY	STROBE	DMARDY#

All other signals on the IDE connector retain their functional definitions during the UDMA/66 operation.

IDE\_IOW# is defined as STOP for both read and write transfers to request to stop a transaction.

IDE\_IOR# is redefined as DMARDY# for transferring data from the IDE device to the ATAC. It is used by the ATAC to signal when it is ready to transfer data and to add wait states to the current transaction. The IDE\_IOR# signal is defined as STROBE for transferring data from the ATAC to the IDE device. It is the data strobe signal driven by the ATAC on which data is transferred during each rising and falling edge transition.

IDE\_IORDY is redefined as STROBE for transferring data from the IDE device to the ATAC during a read cycle. It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. IDE\_IORDY is defined as DMARDY# during a write cycle for transferring data from the ATAC to the IDE device. It is used by the IDE device to signal when it is ready to transfer data and to add wait states to the current transaction.

UDMA/66 data transfer consists of three phases: a startup phase, a data transfer phase, and a burst termination phase.

The IDE device begins the startup phase by asserting IDE\_DREQ. When ready to begin the transfer, the ATAC asserts IDE\_DACK#. When IDE\_DACK# is asserted, the ATAC drives IDE\_CS0# and IDE\_CS1# asserted, and IDE\_ADDR[2:0] low. For write cycles, the ATAC negates STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, the ATAC negates STOP and asserts DMARDY#. The IDE device then sends the first data word and asserts STROBE.

The data transfer phase continues the burst transfers with the ATAC and the IDE via providing data, toggling STROBE and DMARDY#. IDE\_DATA[15:0] is latched by the receiver on each rising and falling edge of STROBE. The transmitter can pause the burst cycle by holding STROBE high or low, and resume the burst cycle by again toggling STROBE. The receiver can pause the burst cycle by negating DMARDY# and resumes the burst cycle by asserting DMARDY#.

The current burst cycle can be terminated by either the transmitter or receiver. A burst cycle must first be paused, as described above, before it can be terminated. The ATAC can then stop the burst cycle by asserting STOP, with the IDE device acknowledging by negating IDE\_DREQ. The transmitter then drives the STROBE signal to a high level. The ATAC then puts the result of the CRC calculation onto IDE\_DATA[15:0] while de-asserting IDE\_DACK#. The IDE device latches the CRC value on the rising edge of IDE\_DACK#.

The CRC value is used for error checking on UDMA/66 transfers. The CRC value is calculated for all data by both the ATAC and the IDE device during the UDMA/66 burst transfer cycles. This result of the CRC calculation is defined as all data transferred with a valid STROBE edge while IDE\_DACK# is asserted. At the end of the burst transfer, the ATAC drives the result of the CRC calculation onto IDE\_DATA[15:0], which is then strobed by the de-assertion of IDE\_DACK#. The IDE device compares the CRC result of the ATAC to its own and reports an error if there is a mismatch.

The timings for UDMA/66 are programmed into the DMA control registers:

- Channel 0 Drive 0 DMA (ATAC\_CH0D0\_DMA) (MSR 51300021h)
- Channel 0 Drive 1 DMA (ATAC\_CH0D1\_DMA) (MSR 51300023h)

The bit formats for these registers are given in Section 6.4.3 "ATAC Native Registers" on page 274. Note that MSR 51300021h[20] is used to select either MDMA or UDMA mode. Bit 20 = 0 selects MDMA mode. If bit 20 = 1, then UDMA/66 mode is selected. Once mode selection is made using this bit, the remaining DMA registers also operate in the selected mode.

Also listed in the bit formats are recommended values for both MDMA modes 0-2 and UDMA/66 modes 0-4. Note that these values are only recommended settings and are not 100% tested.

## 5.5 Universal Serial Bus Controller

The two Universal Serial Bus Controllers (USBC) each contain a GeodeLink™ Adapter, PCI Adapter, and USB Core blocks. The functional descriptions of the blocks are described in the following subsections.

### 5.5.1 GeodeLink™ Adapter

The GeodeLink Adapter (GLA) translates GeodeLink transactions to/from Local bus transactions. The GLA interfaces to a 64-bit GLIU (GeodeLink Interface Unit) and a 32-bit Local bus. The GLA supports in-bound memory and I/O requests which are converted by the PCI Adapter (PA) into PCI memory and I/O requests that target the USBC. It also supports in-bound MSR transactions to the MSRs. These are located “between” the GLA and PA. Lastly, there is a special MSR used to pass PCI configuration requests to the PA. The GLA supports out-bound memory requests only. I/O and MSR transactions from the USBC never occur. USBC PCI master requests are converted by the PA into Local bus master requests. These requests may consist of a simple 4-byte read or write. Alternatively, a PCI burst transaction of any length may be converted to an

appropriate series of GLIU transactions by the GLA. Lastly, the GLA synchronizes GLIU transactions at the GLIU clock to the slower Local bus transaction at Local bus clock.

### 5.5.2 PCI Adapter

The PCI Adapter translates PCI signals to a specific Local bus transaction that is attached to the GLA, while the PCI signals are connected directly to a compatible PCI device. It also translates the Local bus transactions to PCI transactions.

### 5.5.3 USB Core

The USB Core is a PCI-based implementation of the Universal Serial Bus (USB) v1.1 Specification utilizing the Open Host Controller Interface (OHCI) standard developed by Compaq, Microsoft®, and National Semiconductor. The USB Core consists of the three blocks: Host Controller, USB Interface, and PCI Interface

The USB Core block diagram is shown in Figure 5-8.

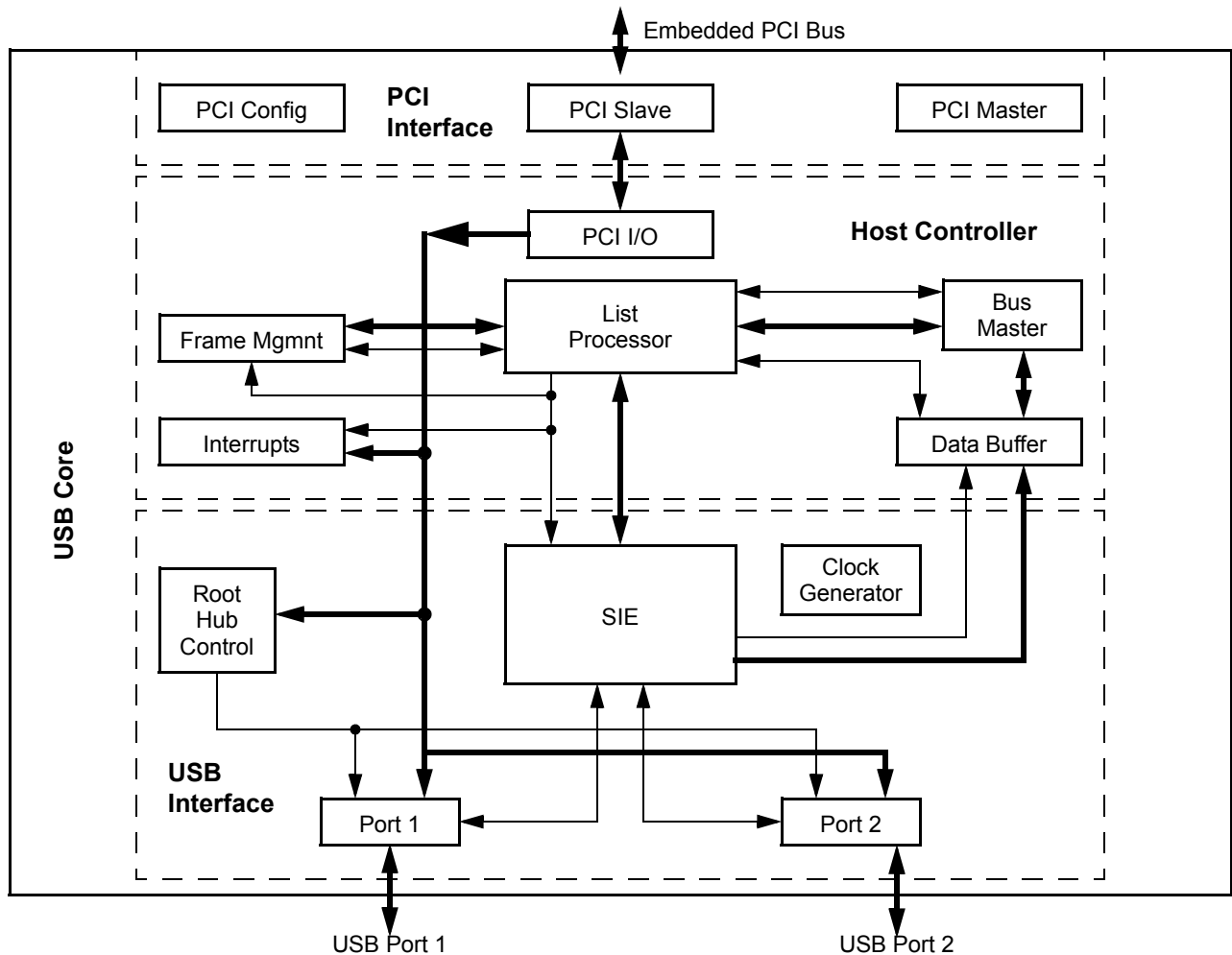


Figure 5-8. USB Core Block Diagram

### 5.5.4 Host Controller

The USB host interacts with USB devices through the Host Controller. The host is responsible for:

- Detecting the attachment and removal of USB devices.
- Managing control flow between the host and USB devices.
- Managing data flow between the host and USB devices.
- Collecting status and activity statistics.
- Providing power to attached USB devices.

The USB system software on the host manages interactions between USB devices and host-based device software. There are five areas of interactions between the USB system software and device software:

- 1) Device enumeration and configuration.
- 2) Isochronous data transfers.
- 3) Asynchronous data transfers.
- 4) Power management.
- 5) Device and bus management information.

Whenever possible, the USB system software uses existing host system interfaces to manage the above interactions.

The OHCI specification for the Universal Serial Bus is a register-level description of a host controller for the Univer-

sal Serial Bus, which in turn is described by the Universal Serial Bus specification. OHCI allows multiple host controller vendors to design and sell host controllers with a common software interface, freeing them from the burden of writing and distributing software drivers. The design goal has been to balance the complexity of the hardware and software so that OHCI is more than the simplest possible host controller for USB yet not the most complex possible.

The Host Controller has four USB states visible to the host controller driver via the operational registers: USBOPERATIONAL, USBRESET, USBRESUME, and USBRESUME. These states define the Host Controller responsibilities relating to USB signaling and bus states. The USB states are reflected in the HostControllerFunctionalState field of the HcControl register. The Host Controller may only perform a single state transition. During a remote wakeup event, the Host Controller may transition from USBSUSPEND to USBRESUME. The Host Controller interface registers are PCI memory mapped I/O. The functional state machine (FSM) is shown in Figure 5-9.

### 5.5.5 USB Interface

The USB Interface includes the integrated Root Hub with two external ports, Port 1 and Port 2, as well as the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the Host Controller as well as the hub and port management specified by USB.

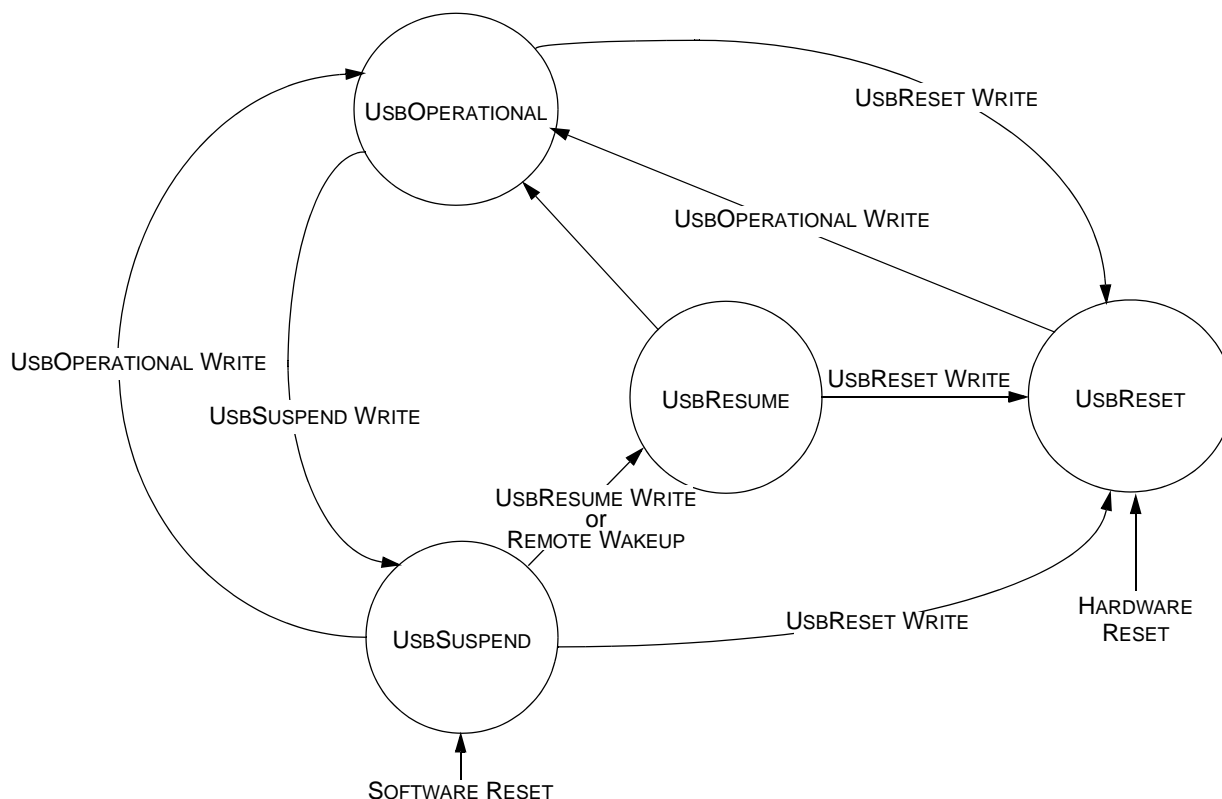
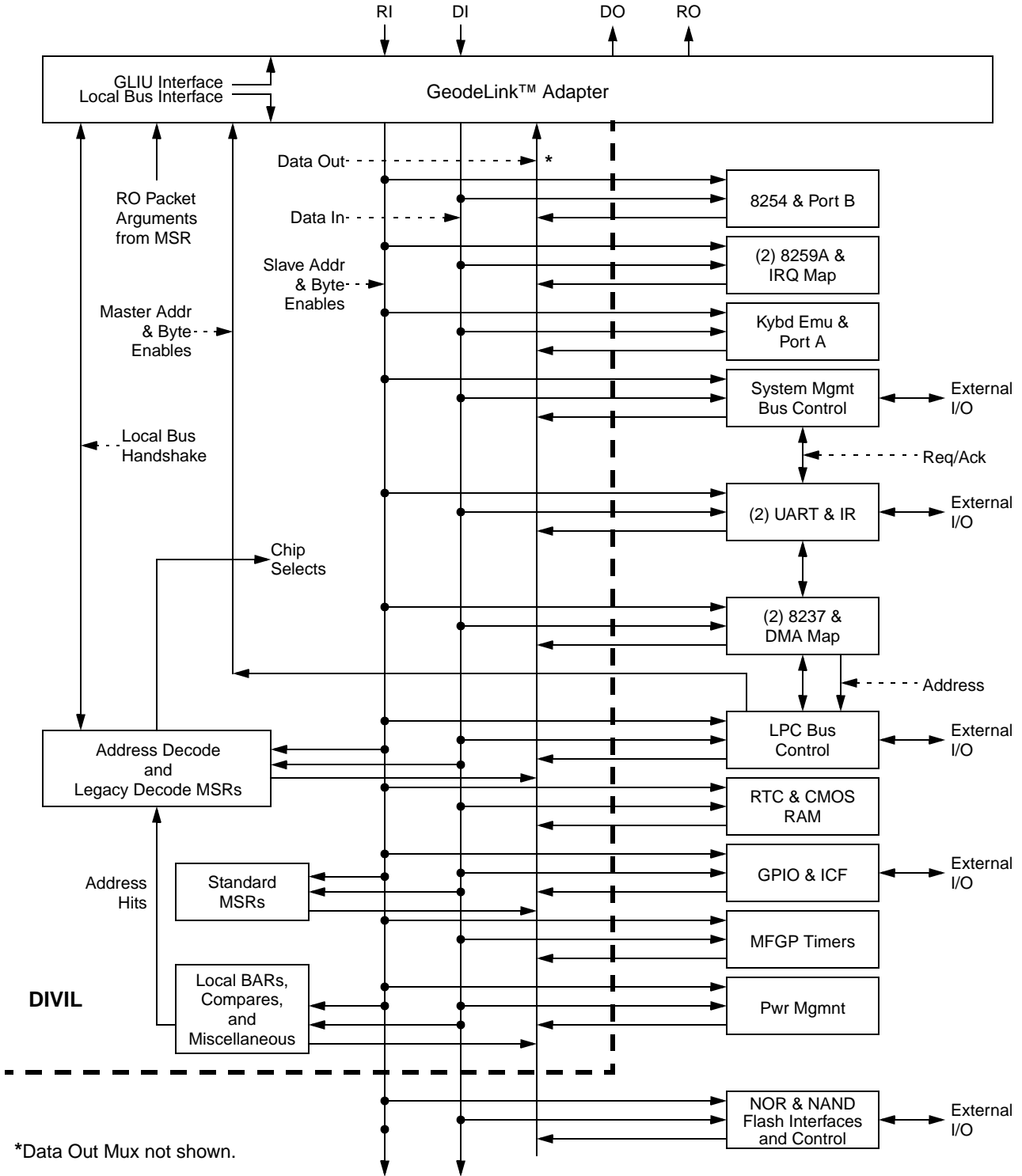


Figure 5-9. USB Host Controller FSM

### 5.6 Diverse Integration Logic

The Diverse Integration Logic (DIVIL) connects a series of Local bus devices to the GeodeLink™ architecture. Figure 5-10 illustrates how the DIVIL (within the dashed lines) interfaces with the other devices of the Diverse Device.

The main blocks of the DIVIL are: Address Decode, Standard MSRs, Local BARs, and Dataout Mux (DOM).



\*Data Out Mux not shown.

Figure 5-10. Diverse Logic Block Diagram



- **Address Decode** - Decodes the upper Local bus address bits to select a target slave. Most of the legacy devices have fixed addresses or are selectable between a small number of selectable I/O addresses. However, many of the functions are relocatable via a Local Base Address Register (LBAR); established via an MSR. Address Decode also detects special GLIU cycles, such as Shutdown, and takes appropriate action.
- **Standard MSRs** - Includes the Standard GeodeLink Device MSRs found in all GeodeLink Devices: Capabilities, Master Configuration, SMI Control, Error Control, Power Management, and Diagnostics.
- **Local BARs** - Local Base Address Registers (LBARs) establish the location of non-legacy functions within the Diverse Device. The module also includes logic to compare the current bus cycle address to the LBAR to detect a hit. For the I/O LBARs, the I/O address space 000h-4FFh is off limits. No I/O LBAR is allowed to point to this space.
- **Data Out Mux (DOM)** - This mux is not explicitly illustrated. Each function above produces a single output to the DIVIL. The DIVIL DOM has a port for each of the functions and is responsible for selecting between them.

mask and base address values are established via an MSR.

### 5.6.1.1 Fixed Target Size I/O LBARs

This discussion applies to the following LBARs:

- MSR 51400008h: IRQ Mapper (MSR\_LBAR\_IRQ)
- MSR 5140000Bh: SMB (MSR\_LBAR\_SMB)
- MSR 5140000Ch: GPIO and ICFs (MSR\_LBAR\_GPIO)
- MSR 5140000Dh: MFGPTs (MSR\_LBAR\_MFGPT)
- MSR 5140000Eh: ACPI (MSR\_LBAR\_ACPI)
- MSR 5140000Fh: Power Management Support (MSR\_LBAR\_PMS)

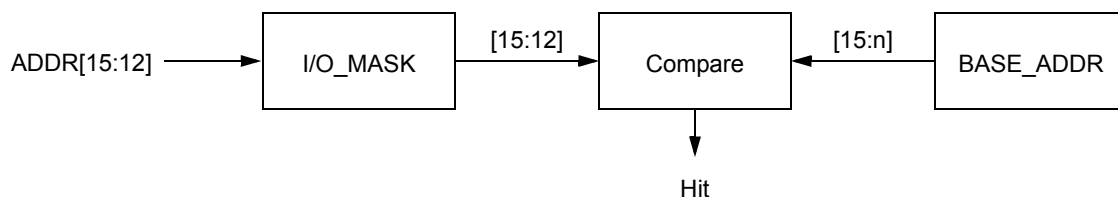
The IO\_MASK only applies to the upper bits [15:12] (see Figure 5-11). Normally, one would set all the mask bits (i.e., no mask on upper bits). One should only mask or clear bits if address wrapping or aliasing is desired.

- **Rule.** When a mask bit is cleared, the associated bit in the base address must also be cleared. Otherwise, the compare will not be equal on these bits. This rule applies to both memory and I/O LBARs.

The base size is fixed based on the target. For example, the GPIO takes 256 bytes of address space. Therefore, the base only applies to bits [15:8]. Base bits [7:0] are always forced by the hardware. Therefore, the base is always forced by hardware to be on a boundary the size of the target.

### 5.6.1 LBARs and Comparators

The LBARs are used to establish the address and hence, chip select location of all functions that do not have fixed legacy addresses. This block also has comparators to establish when a current bus cycle address hits an LBAR. A hit is passed to the address decode block and results in a chip select to the target device if there are no conflicts. The



**Notes:**

- 1) The I/O mask is always 4 bits.
- 2) The I/O base address is variable ([15:n]).

The value of “n” depends on the I/O space requirements of the target. For example, a device needing 4, 8, 16, 32, 64, 128, or 256 bytes of I/O space has “n” = 2, 3, 4, 5, 6, 7, 8, respectively. The value “n” for various functions is:

MSR_LBAR_IRQ	n = 5	MSR_LBAR_SMB	n = 3
MSR_LBAR_GPIO	n = 8	MSR_LBAR_MFGPT	n = 6
MSR_LBAR_ACPI	n = 5	MSR_LBAR_PMS	n = 7
MSR_LBAR_FLASH_IO	n = 4		

**Figure 5-11. I/O Space LBAR - Fixed Target Size**

**5.6.1.2 Variable Target Size I/O LBARs**

This discussion applies to the following LBARs:

- MSR 51400010h: Flash Chip Select 0 (MSR\_LBAR\_FLSH0) with bit 34 = 0 (I/O mapped)
- MSR 51400011h: Flash Chip Select 1 (MSR\_LBAR\_FLSH1) with bit 34 = 0 (I/O mapped)
- MSR 51400012h: Flash Chip Select 2 (MSR\_LBAR\_FLSH2) with bit 34 = 0 (I/O mapped)
- MSR 51400013h: Flash Chip Select 3 (MSR\_LBAR\_FLSH3) with bit 34 = 0 (I/O mapped)

**Note:** Flash Chip Selects [3:0] can be programmed for I/O or a memory space. See Section 5.6.1.3 "Memory LBARs".

The I/O LBAR works just like the Fixed style, except the size of the IO\_MASK has been expanded to cover the entire address range (see Figure 5-12). In the Fixed style, the IO\_MASK applies to bits [15:12] but for Variable style, the IO\_MASK applies to bits [15:4]. If all bits are set, then the target size is 16 bytes and base address bits [15:4] determine the base. Base bits [3:0] are a "don't care" and are effectively forced to zero by the hardware. Thus, the smallest I/O target is 16 bytes. As the LSBs of IO\_MASK are cleared, the "target space" expands. For example, assume a 64-byte device is desired in I/O space. The IO\_MASK = FFCh, base address bits [15:6] are programmed to the desired base, and base address bits [5:4] are cleared (see *Rule* on page 105).

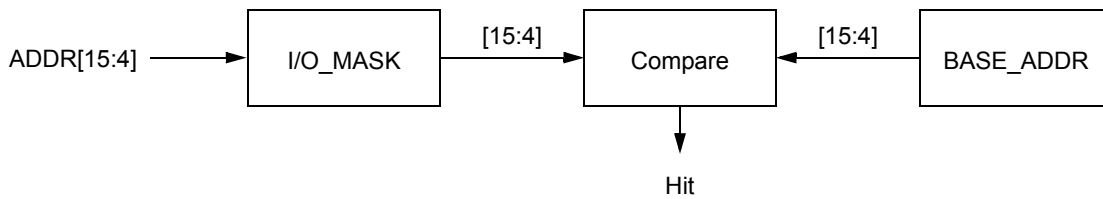
**5.6.1.3 Memory LBARs**

This discussion applies to the following LBARs:

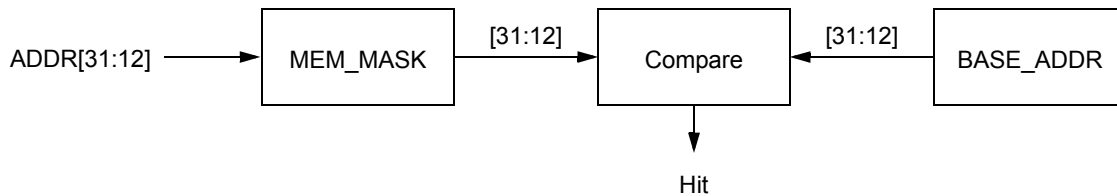
- MSR 51400009h: KEL from USB Host Controller 1 (MSR\_LBAR\_KEL1)
- MSR 51400010h: Flash Chip Select 0 (MSR\_LBAR\_FLSH0) with bit 34 = 1 (memory mapped)
- MSR 51400011h: Flash Chip Select 1 (MSR\_LBAR\_FLSH1) with bit 34 = 1 (memory mapped)
- MSR 51400012h: Flash Chip Select 2 (MSR\_LBAR\_FLSH2) with bit 34 = 1 (memory mapped)
- MSR 51400013h: Flash Chip Select 3 (MSR\_LBAR\_FLSH3) with bit 34 = 1 (memory mapped)
- MSR 5140000Ah: KEL from USB Host Controller 2 (MSR\_LBAR\_KEL2)

**Note:** The Flash Chip Selects [3:0] can be programmed for an I/O space or a Memory space.

For memory space, the LBAR works exactly like the Variable style (see Figure 5-13), except that clearing the LSBs of the MEM\_MASK begins to make sense. For example, assume there is a 64 KB external ROM that is going to be connected to Flash Chip Select 0. Such a device needs address bits [15:0]. The MEM\_MASK would normally be programmed to FFFF0h and base address bits [31:16] would be programmed to the desired base. The values in base address [15:12] would be cleared because the associated mask bits are cleared (see *Rule* on page 105). Lastly, the memory target can not be smaller than 4 KB.



**Figure 5-12. I/O Space LBAR - Variable Target Size**



**Note:** The memory mask is always 20 bits, which is equal to the number of memory base address bits.

**Figure 5-13. Memory Space LBAR**

#### 5.6.1.4 Miscellaneous Block

Special cycles are sent to the Miscellaneous block. They are decoded as given in Table 5-10. Note that the Halt special cycle depends on the value of the SPEC\_CYC\_MD bit in MSR\_LEG\_IO (MSR 51400014h[28]).

Soft IRQ and Soft Reset MSRs are decoded within the Miscellaneous block.

Each block in the Diverse Device generates an output to the DIVIL. The DIVIL DOM has a port for each of the functions and is responsible for selecting between them.

#### 5.6.2 Standard MSRs

This block contains the Standard GeodeLink Device MSRs and their associated logic. These standard MSRs are: Capabilities, Master Configuration, SMI Control, Error Control, Power Management, and Diagnostics. The Capabilities, Master Configuration, and Diagnostic MSRs are “passive” in that they contain values that have an effect elsewhere. The other MSRs have various “active” bits that are set and cleared via hardware/software interactions.

**Table 5-10. Special Cycle Decodes**

Cycle Type	Address	Function	Action
Write	00h	Shutdown	Send shutdown pulse to MSR_ERROR.
			Send shutdown pulse to MSR_SMI.
			If RESET_SHUT_EN (MSR 51400014h[31]) is high, send reset pulse to power management indicating shutdown reset.
<b>SPEC_CYC_MD (MSR 51400014h[28]) = 0</b>			
Write	01h	Halt	Send halt pulse to MSR_SMI.
<b>SPEC_CYC_MD (MSR 51400014h[28]) = 1</b>			
Write	02h	Halt	Send halt pulse to MSR_SMI.
	All other values	x86 Special	Discard with no side effects.
	All other values	Not Defined	Discard with no side effects.
Read	00h	Interrupt Ack	Send cycle to PIC.
			GeodeLink Adapter generates back-to-back bus cycles.
	All other values	Not Defined	Return zero with no side effects.

## 5.7 Programmable Interval Timer

The Programmable Interval Timer (PIT) generates programmable time intervals from the divided clock of an external clock signal of a crystal oscillator. The PIT (8254) has six modes of operation. Figure 5-14 shows the block diagram of the PIT and its connectivity to the Local bus.

The 8254 is comprised of three independently programmable counters. Each counter is 16 bits wide. A 14.318 MHz external clock signal (from a crystal oscillator or an external clock chip) is divided by 12 to generate 1.19 MHz, which is used as a clocking reference for these three counters.

Each counter is enabled or triggered with its GATE signal. Based on the counting mode, the counter concerned is activated by a high level or a low-to-high transition of its GATE signal.

Each counter has its output signal, whose shape is dependent upon the counter's operational mode. The Control register loads the counters and controls the various modes of operation. This Control register controls the operation mode of the control logic (counter state machine), which in turn controls the counter, the high-order and low-order output latches. A status latch is also present in the 8254 and is used to output status information.

### Features include:

- Comprised of three 16-bit wide counters.
- Supports read-back and counter latch commands.
- Supports six modes of counting.
- Allows several counter latch commands in parallel with the read-back command.

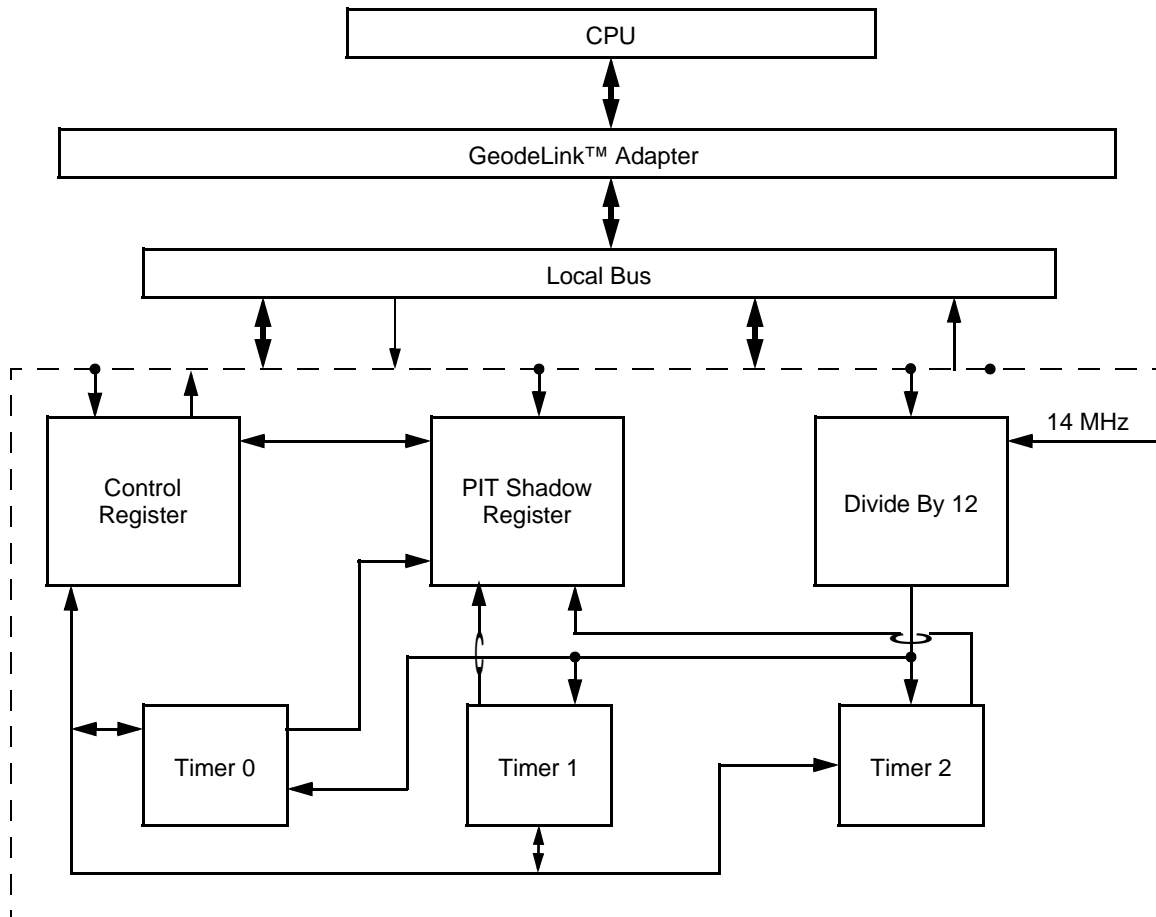


Figure 5-14. PIT Block Diagram

### 5.7.1 Programming the 8254 PIT

Programming of the 8254 PIT is initiated by first writing one control word via I/O Address 043h into the PIT Mode Control Word register. It is followed by writing one or two data bytes via the I/O address of the intended counter. If the Control register is loaded once, the counters may be overwritten with different values without accessing the Control register again. Table 5-11 lists the I/O addresses of the various registers.

**Table 5-11. 8254 PIT Register Ports**

I/O Address	Register	Access Type
040h	Counter 0	Read/ Write
041h	Counter 1	Read / Write
042h	Counter 2	Read / Write
043h	Control Word	Write

The Control register in the 8254 PIT is write-only, but certain control information can be determined by the read-back (read-status) command.

#### 5.7.1.1 Write to the Counters

To load a counter with new values, a control word needs to output that defines the intended counter, number and type of bytes to write, the counting mode and the counting format. Bits [5:4] of the Control Word register (see Section 6.8.2.7 on page 347) indicate whether low-order or high-order or both are going to be written. If low-order or high-order counter byte only is specified to be written, then only that byte can be read during a read access. According to bits [5:4] of the Control Word register, one needs to write either the low-order or the high-order or both into the counter after passing the control word. If bits [5:4] of the Control Word register is 11, then a low-order byte needs to be written first, followed by a high-order byte. For small counting values or counting values that are multiples of 256, it is sufficient to pass the low-order or high-order counter byte. Bits [3:1] of the Control Word register define

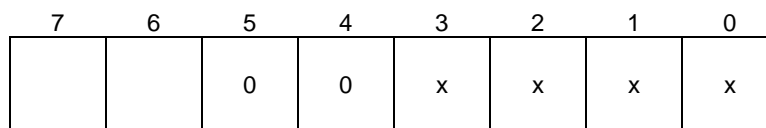
the counting mode of the counter selected by bits [5:4]. Bit 0 of the Control Word register defines the binary or BCD counting format. The maximum loadable count value is not FFFFh (binary counting) or 9999 (BCD counting), but 0. On the next CLK pulse the counter concerned jumps to FFFFh or 9999. Once the value is decreased to 0 again, it outputs a signal according to the programmed mode. Therefore, the value 0 corresponds to 2<sup>16</sup> for binary counting and 10<sup>4</sup> for BCD counting.

#### Read from the Counters

There are three options for reading a counter in the 8254 PIT:

- 1) Counter Latch command
- 2) Read-back (read-status) command
- 3) Direct Read

To read a counter, the third option (Direct Read) should not be used. The Counter Latch command or Read-back command should be used to transfer the current state of the counter into its output latches. One or two successive read counter instructions for the port address of the counter concerned reads these latches. If only the low-order or high-order byte was written when the counter was loaded with the initial counting value, then read the current counting value of the initially written byte by a single read counter instruction. If both the low-order and high-order counter bytes are written previously, then to read the current counter value, two read counter instructions are needed. The 8254 PIT returns the low-order byte of the 16-bit counter with the first read counter instruction, and then the high-order byte with the second read counter instruction. If the content of the counter has been transferred once by a counter latch command into the output latches, then this value is held there until the CPU executes one or two counter read instructions, or until the corresponding counter is reprogrammed. Successive counter latch commands are ignored if the output latches haven't been read before. Figure 5-15 shows the format of the control word for the counter latch command.



Bits [7:6] = Select counter to latch

**Figure 5-15. PIT Counter Latch Command Format**

The read-back command present in the 8254 PIT is used to determine the current counter value and its status like counting format, the counting mode, the low-order or high-order byte or both being read or written, and the status of its output. Figure 5-16 shows the format of the read-back command. The two most significant bits define the read-back command with their value 11b. CT and ST indicate that the value and the status of the counter are to be determined respectively. The C0-C2 bits define the counter whose value or status is to be determined. With the read-back command, several counter latch commands can be issued in parallel by indicating several counters simultaneously with the C0-C2 bits. The 8254 then behaves as if several counter latch commands have been issued individually, and transfers the individual count values into the output latches of each counter. All successive counter latch

commands, whether issued by its own counter or a next read-back command, are ignored if the counter concerned has not been read by counter read instructions. To determine the programmed mode of a particular counter, set CT = 1 and ST = 0.

The read-back command latches the current mode and provides a status byte (see Figure 5-17) at the port address of the counter concerned. This status byte is fetched by a counter read instruction. The PIN bit indicates the current status of the concerned counter's output pin. If PIN = 1, then the counter output is at logic 1, else at logic 0. Bit 0 shows whether the last written counter value has already been transferred to the counter. Not before zero = 0 is it meaningful to read back the counter value.

7	6	5	4	3	2	1	0
1	1	CT	ST	C2	C1	C0	x

CT: Determine count value of selected counter.

0 = Determine count value.

1 = Do not determine count value.

ST: Determine status of selected counter.

0 = Determine count status.

1 = Do not determine count status.

C2, C1, C0: Counter selection.

0 = Counter not selected.

1 = Counter selected.

**Figure 5-16. PIT Read-Back Command Format**

7	6	5	4	3	2	1	0
PIN	LOAD	LH		MODE			BCD

PIN: Status of counter output pin:

0 = Output pin low.

1 = Output pin high.

LOAD: Is counter loaded with a start value?

0 = Counter loaded, count value can be read.

1 = Counter not yet loaded, count value cannot be read.

LH: Corresponds to bits [5:4] of the Control Word register.

MODE: Corresponds to bits [3:1] of the Control Word register.

BCD: Corresponds to bit 0 of the Control Word register.

**Figure 5-17. PIT Status Byte Format**

## 5.8 Programmable Interrupt Control

The Programmable Interrupt Control subsystem (PIC) is illustrated in Figure 5-18. The major modules are the Mapper and Masks (MM), Extended PIC (XPIC), and Legacy 8259A PIC (LPIC).

### Features

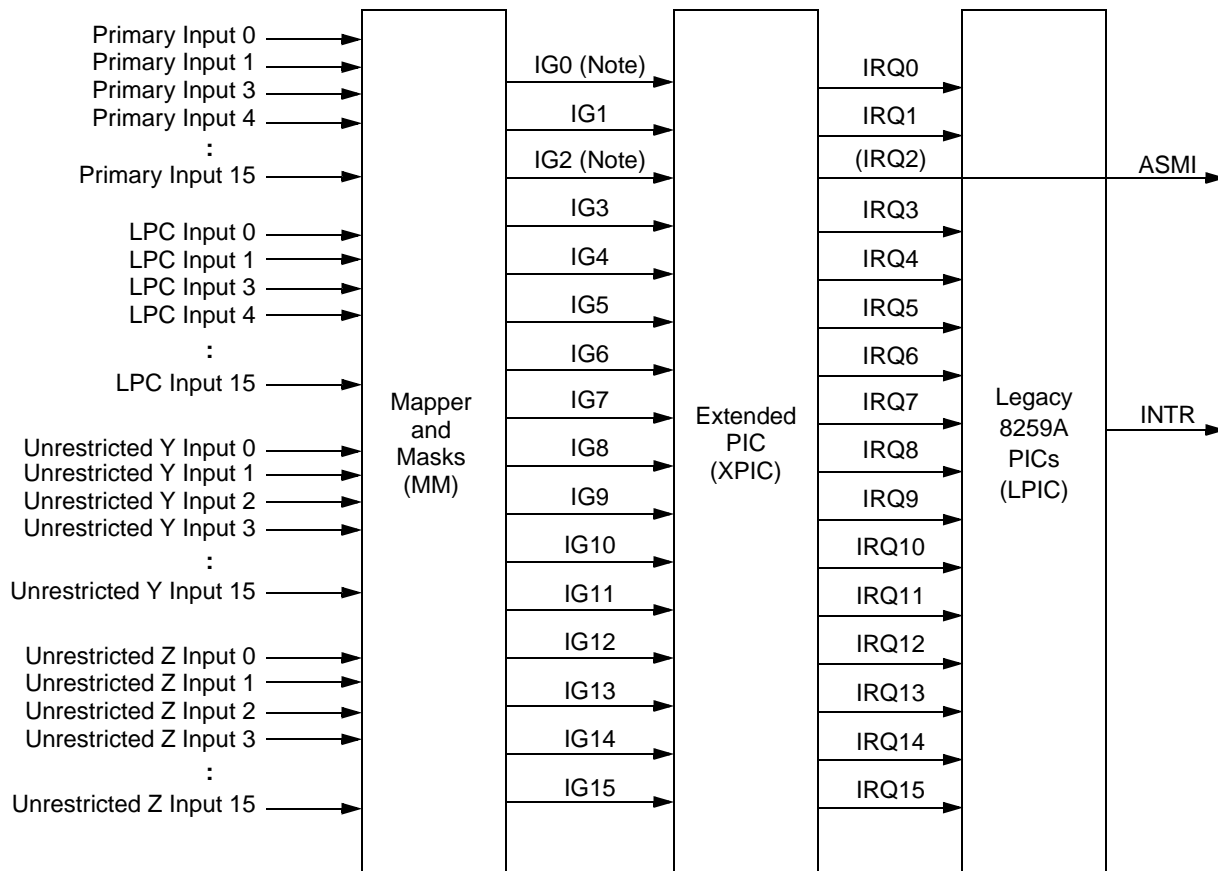
- Two x86 compatible 8259As
- 15-Level Priority Controller
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Individual Edge/Level Controls
- Complete 8259A state read back via shadow registers
- Mapper routes 62 inputs to 15 legacy interrupts and one Asynchronous System Management Interrupt (ASMI)
- All 62 inputs individually maskable and status readable (MSRs 51400020h-51400027h or PIC I/O Offsets 00h-1Ch, see Section 6.9.1 on page 350)

### 5.8.1 Mapper and Masks

This block maps and masks up to 62 interrupt sources to 60 discrete Extended PIC (XPIC) inputs. The sources are organized into four groups:

- 1) 15 Primary pre-defined inputs (see Table 5-12)
- 2) 15 LPC inputs pre-defined (see Table 5-12)
- 3) 16 Unrestricted Y inputs (see Table 5-13)
- 4) 16 Unrestricted Z inputs (see Table 5-14)

The outputs are organized into 16 groups of four signals each, except groups 0 and 2; they have two signals each. Each group is called an Interrupt Group (IG). Each pre-defined input is mapped to a specific IG. Each unrestricted input can be mapped to any IG except IG0. Regardless of mapping, any interrupt source can be masked to prevent participation in the interrupt process. Once the input to output map is established along with the mask values, signal flow from input to output is always completely combinatorial.



**Note:** The outputs are organized into 16 groups of four signals each, except IG0 and IG2; they have two signals each.

Figure 5-18. PIC Subsystem

Table 5-12. IRQ Map - Primary and LPC

Input #	Primary Sources	LPC Sources	Legacy IRQ
Input 0	8254 Timer IRQ	LPC IRQ0	8254 Timer
Input 1	KEL IRQ1	LPC IRQ1	Keyboard
N/A	None (Slave Controller)	None	None
Input 3	Reserved - Grounded	LPC IRQ3	UART
Input 4	Reserved - Grounded	LPC IRQ4	UART
Input 5	Reserved - Grounded	LPC IRQ5	Parallel Port 2
Input 6	Reserved - Grounded	LPC IRQ6	Floppy
Input 7	Reserved - Grounded	LPC IRQ7	Parallel Port1
Input 8	RTC Periodic IRQ	LPC IRQ8	RTC
Input 9	Reserved - Grounded	LPC IRQ9	Undefined
Input 10	Reserved - Grounded	LPC IRQ10	Undefined
Input 11	Reserved - Grounded	LPC IRQ11	Undefined
Input 12	KEL IRQ12	LPC IRQ12	Mouse
Input 13	Float Point Error IRQ	LPC IRQ13	FPU
Input 14	Primary IDE Channel IRQ	LPC IRQ14	Primary IDE
Input 15	Reserved - Grounded	LPC IRQ15	Secondary IDE



**Table 5-13. IRQ Map - Unrestricted Sources Y**

Unrestricted Y	Source	Comment
Input 0	Software Generated IRQ	
Input 1	USB1 IRQ	
Input 2	USB2 IRQ	
Input 3	RTC Alarm	This is a pulse from the RTC. Must use edge triggered interrupt, that is, level interrupt will not work.
Input 4	Audio IRQ	OR of all audio codec interrupts and master interrupts.
Input 5	Power Management SCI	OR of all possible power management System Control Interrupts (SCIs).
Input 6	NAND Flash Ready	Ready to perform NAND write or read.
Input 7	NAND Flash Distraction	NOR access occurred during NAND operation causing a NAND abort or distraction.
Input 8	Reserved, Grounded	
Input 9	Reserved, Grounded	
Input 10	Reserved, Grounded	
Input 11	Reserved, Grounded	
Input 12	SMB Controller IRQ	
Input 13	KEL Emulation IRQ	
Input 14	UART 1 IRQ	
Input 15	UART 2 IRQ	

**Table 5-14. IRQ Map - Unrestricted Sources Z**

Unrestricted Z	Source	Comment
Input 0	MFGPT_Comp_1A	OR of MFGPT_Comp_1 0 and 4.
Input 1	MFGPT_Comp_1B	OR of MFGPT_Comp_1 1 and 5.
Input 2	MFGPT_Comp_1C	OR of MFGPT_Comp_1 2 and 6.
Input 3	MFGPT_Comp_1D	OR of MFGPT_Comp_1 3 and 7.
Input 4	MFGPT_Comp_2A	OR of MFGPT_Comp_2 0 and 4.
Input 5	MFGPT_Comp_2B	OR of MFGPT_Comp_2 1 and 5.
Input 6	MFGPT_Comp_2C	OR of MFGPT_Comp_2 2 and 6.
Input 7	MFGPT_Comp_2D	OR of MFGPT_Comp_2 3 and 7.
Input 8	GPIO Interrupt 0	From GPIO Interrupt/PME Mapper.
Input 9	GPIO Interrupt 1	From GPIO Interrupt/PME Mapper.
Input 10	GPIO Interrupt 2	From GPIO Interrupt/PME Mapper.
Input 11	GPIO Interrupt 3	From GPIO Interrupt/PME Mapper.
Input 12	GPIO Interrupt 4	From GPIO Interrupt/PME Mapper.
Input 13	GPIO Interrupt 5	From GPIO Interrupt/PME Mapper.
Input 14	GPIO Interrupt 6	From GPIO Interrupt/PME Mapper.
Input 15	GPIO Interrupt 7	From GPIO Interrupt/PME Mapper.

**5.8.2 Extended PIC (XPIC)**

For each of 16 input IGs of four signals each (except IG0 and IG2 with two signals each), XPIC provides a four input "OR". Thus, 16 outputs are formed. A software readable XPIC Input Request Register (XIRR) is available to read the status of the 64 inputs. Outputs [0:1] and [3:15] are connected directly to the corresponding inputs on LPIC. Output 2 can be used as an ASMI.

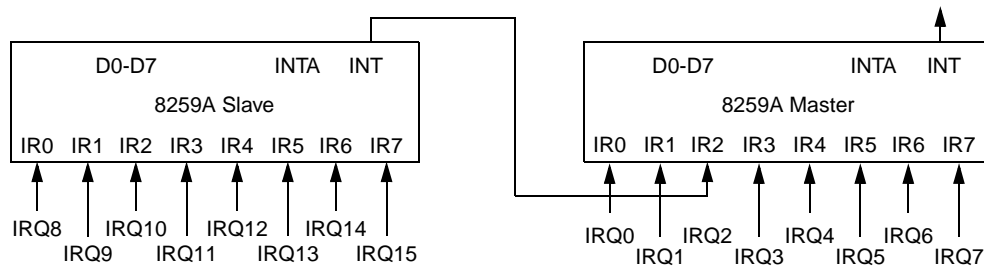
**5.8.3 Legacy PIC (LPIC)**

The LPIC consists of two 8259A compatible Programmable Interrupt Controllers (PICs) connected in Cascade mode through IR2 (see Figure 5-19). The LPIC devices support all x86 modes of operation except Special Fully Nested mode. LPIC contains mechanisms to:

- 1) Mask any of the 15 inputs via an Interrupt Mask Register (IMR).

- 2) Determine the input request status via an Interrupt Request Register (IRR).
- 3) Generate an interrupt request (INTR) to the processor when any of the unmasked requests are asserted.
- 4) Provide an interrupt vector to the processor as part of an interrupt acknowledge operation based on request priorities.
- 5) Determine which requests are acknowledged but not yet fully serviced, via an In-Service Register (ISR).

In addition to the above 8259A features, there are two registers to control edge/level mode for each of the interrupt inputs as well as shadow registers to obtain the values of legacy 8259A registers that have not been historically readable.



**Note:** Cascading the 8259A PICs. The INT output of the slave is connected to the IRQ2 input of the master.

**Figure 5-19. Cascading 8259As for LPIC**

As illustrated in Figure 5-20, the blocks that make up the 8259A PIC are:

- Read/Write Control Logic
- Interrupt Request Register (IRR)
- In-Service Register (ISR)
- Interrupt Mask Register (IMR)
- Priority Resolver
- Interrupt Sequence
- Data Bus Buffer
- Cascade Buffer/Comparator

**Read/Write Control Logic**

The function of this block is to accept commands from the CPU. It contains the four Initialization Command Word registers, ICW1-ICW4, and three Operation Command Word registers, OCW1-OCW3, that can be programmed to operate in various modes.

**IRR, ISR, and IMR**

Three registers are available to handle interrupts in the PIC: Interrupt Request Register (IRR), In-Service Register (ISR), and Interrupt Mask Register (IMR). Each of the three

registers is eight bits wide, where every bit corresponds to one of the IR0-IR7 input lines.

**Priority Resolver**

The priority resolver block manages the hardware requests according to their priority. As several bits may be set in the IRR simultaneously, the priority encoder passes only the highest priority bit; ordered in priority 0 through 7 (0 being the highest).

**Interrupt Sequence**

The INT output goes directly to the CPU interrupt input. When an INT signal is activated, the CPU responds with an Interrupt Acknowledge access that is translated to two pulses on the INTA input of the PIC. At the first INTA pulse the highest priority IRR bit is loaded into the corresponding ISR bit, and that IRR bit is reset. The second INTA pulse instructs the PIC to present the 8-bit vector of the interrupt handler onto the data bus.

**Data Bus Buffer**

Control words and status information are transferred through the data bus buffer.

**Cascade Buffer/Comparator**

This functional block stores and compares the IDs of the PICs.

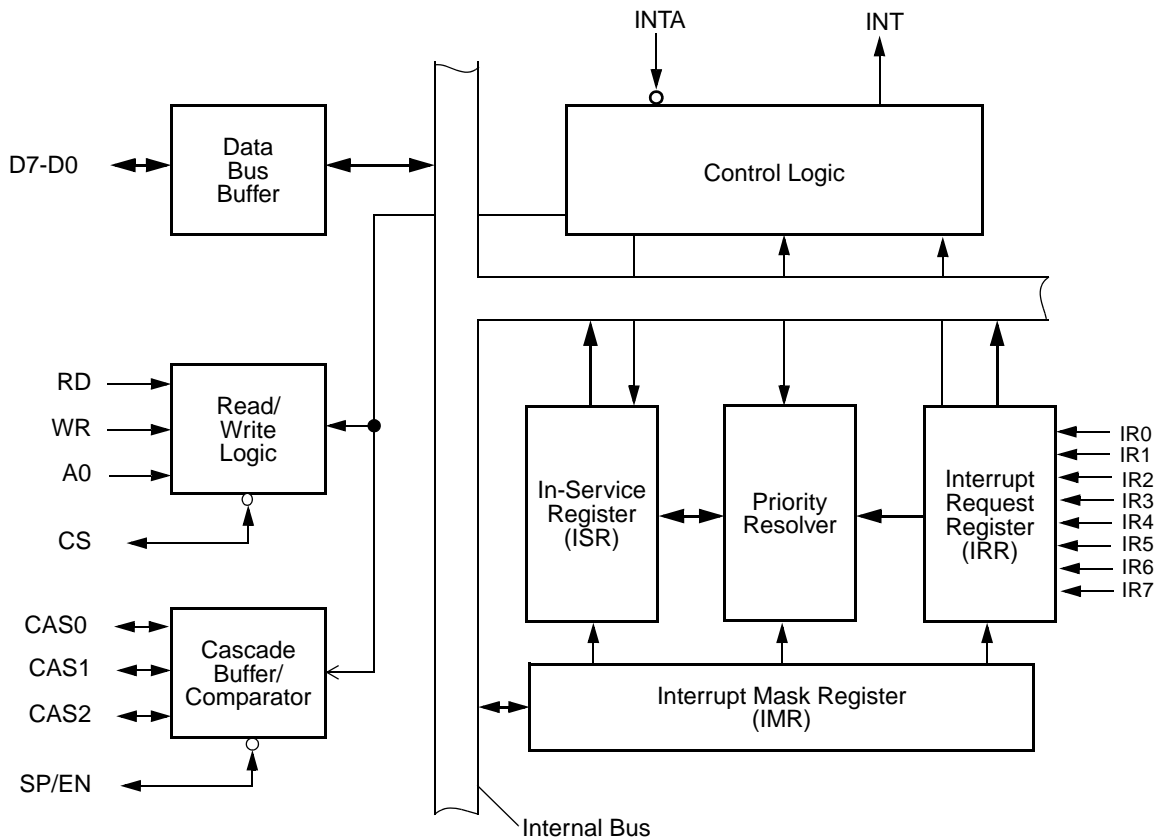


Figure 5-20. PIC 8259A Block Diagram

### 5.8.3.1 Interrupt Sequence

There are three registers in the PIC that control the interrupt requests: Interrupt Request Register (IRR), Interrupt Service Register (ISR), and Interrupt Mask Register (IMR). The eight interrupt lines IR0-IR7 are connected to the IRR.

The peripheral that requests an interrupt raises the signal at the corresponding IR0-IR7 inputs, which sets the corresponding bit in the IRR. Several peripheral devices can issue interrupt requests at the same time. The PIC gates these requests under the Interrupt Mask register and under the priority of any interrupt service routine already entered (using the ISR), and activates the PIC's output INTR to the CPU. The CPU acknowledges the INTR, generating two INTA pulses. On the first, the priority encoder transfers (clears) the highest-priority enabled bit in the IRR to the corresponding bit in the ISR (sets). Also, the two PICs use their Cascade connections to decide which one will be selected to respond further. On the second INTA pulse, the selected PIC presents the 8-bit pointer (called as vector data) onto the data bus. The CPU reads this pointer as the number of the interrupt handler to call.

Software writes a command (EOI) at the end of the interrupt subroutine, which clears the appropriate ISR bit.

### Initialization and Programming

Two types of command words are generated by the CPU to program the PIC:

- 1) **Initialization Command Word (ICW):** The PIC is first initialized by four ICWs (ICW1-ICW4) before any normal operation begins. The sequence is started by writing Initialization Command Word 1 (CW1). After ICW1 has been written, the controller expects the next writes to follow in the sequence ICW2, ICW3, and ICW4 if it is needed.
- 2) **Operation Command Word (OCW):** Using these three OCWs (OCW1-OCW3), the PIC is instructed to operate in various interrupt modes. These registers can be written after the initialization above.

ICWs and OCWs must be programmed before operation begins.

Since both the PICs are cascaded, the ICW3 of the master PIC should be programmed with the value 04h, indicating that the IRQ2 input of the master PIC is connected to the INT output of the slave PIC, rather than the I/O device. This is part of the system initialization code. Also, ICW3 of the slave PIC should be programmed with the value 02h (slave ID) as that corresponds to the input on the master PIC.

For accessing the PIC's registers, two ports are available for the master and slave. Table 5-15 lists the addresses and read/write data for these registers.

**Table 5-15. 8259A PIC I/O Addresses and I/O Data**

I/O Address IRQ0-IRQ7 (Master)	I/O Address IRQ8-IRQ15 (Slave)	Read Data	Write Data
020h	0A0h	IRR ISR	ICW1 OCW2 OCW3
021h	0A1h	IMR	ICW2 ICW3 ICW4 OCW1 (IMR)

### 5.8.3.2 Interrupt Modes

#### Fully Nested Mode

The interrupt requests are ordered in priority from 0 through 7.

The highest priority request is processed and its vector data placed on the bus.

The corresponding ISR bit is set until the trailing edge of the last INTA. While the ISR bit is set, all other interrupts of the same or lower priority are inhibited, while higher levels will be acknowledged only if the CPU's internal interrupt enable flip-flop has been re-enabled through software.

#### End of Interrupt (EOI) Mode

The ISR bit can be reset by a command word that must be issued to the PIC before returning from a service routine.

EOI must be issued twice if in cascade mode, once for the master and once for the slave.

There are two forms of EOI: Specific and Non-Specific.

When a non-specific EOI is issued, the PIC automatically resets the ISR bit corresponding to the highest priority level in service. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

A specific EOI is issued when a mode is used that may disturb the fully nested structure and the PIC might not be able to determine the last interrupt level acknowledged. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the ISR bit to be reset).

#### Automatic End of Interrupt (AEIO) Mode

The PIC automatically performs a non-specific EOI at the trailing edge of the last INTA pulse. This mode is not supported in the Geode CS5535 companion device.

### Automatic Rotation Mode

In cases where a number of IRQs have equal priority, the device that has been serviced, will receive the lowest priority. So now that device, if requesting another interrupt, must wait until the other seven devices have been serviced.

There are two ways to accomplish automatic rotation using OCW2:

- Rotation on the non-specific EOI command (R = 1, SL = 0, EOI = 1).
- Rotation in automatic EOI mode, which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

### Specific Rotation Mode

Priorities can be changed by programming the bottom priority, which fixes all other priorities. For example, if IR5 is programmed as the bottom priority device, then IR6 will have the highest priority. The command is issued to OCW2 (R = 1, SL = 1, and L0-L2 is the binary priority level code of the bottom priority device).

### Special Mask Mode

In this mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked. The special mask mode is set (SSMM = 1, SMM = 1) and cleared (SSMM = 1, SMM = 0) by OCW3.

## 5.8.4 PIC Subsystem Operation

From reset, the PIC subsystem comes up in legacy mode. The “Primary” mapper and mask inputs connect directly to LPIC and all other interrupt sources are masked off.

While there are a number of different ways to use the PIC Subsystem, the discussions that follow assume a mix of “level” and “edge” interrupt inputs. The first discussion assumes the OS schedules the “work” of the interrupt service after a brief interrupt service routine. The second discussion assumes the OS performs the “work” real-time in the interrupt service routine.

Assume the mapper and masks have been established as desired. “Level” interrupts can be shared, but “edge” interrupts cannot. This means an XPIC “level” output can be driven by up to four mapper and masks inputs. Further, this means an XPIC “edge” output can only be driven by one mapper and mask input.

Assume all edge interrupts generate a low-to-high edge to indicate an interrupt. Assume active low interrupts are inverted outside the PIC subsystem as needed; that is, all MM inputs are active high. An external PCI bus uses active low interrupts that can be shared in an open-collector wired “OR” fashion. This is OK. On-chip, the interrupt sense is inverted. Lastly, note that for the edge interrupts the edge must remain high until the interrupt acknowledge action.

Assume LPIC is initialized as follows:

```
;Set Initialization Command Words (ICWs)
;All values are in hex
;PIC #1 (Master)
out 20, 11      ; ICW1 - Edge, Master, ICW4 needed
out 21, 8       ; ICW2 - Interrupt vector table offset is 8
out 21, 4       ; ICW3 - Master level 2
out 21, 1       ; ICW4 - Master, 8086 mode
out 21, ff      ; mask all IRQs
;PIC #2 (Slave)
out a0, 11      ; ICW1 - Edge, Slave ICW4 needed
out a1, 70      ; ICW2 - Interrupt vector table offset 70
out a1, 2       ; ICW3 - Slave Level 2
out a1, 1       ; ICW4 - Slave, 8086 mode
out a1, ff      ; mask all IRQs
;Use Operation Control Words (OCWs) during interrupt
service
```

Thus, the LPIC 8259As all start in edge mode. This is followed by writes to the individual edge level registers at 4D0h (interrupts 0-7) and 4D1h (interrupts 8-15) to establish level mode for all level inputs. Note that IRQ0 and IRQ2 can not be put in level mode. Writing 0FFh to 4D0h will read back 0FAh.

### Scheduled Interrupts Approach

The following set of events would be typical. Assume the processor has maskable interrupts enabled:

- 1) One or more interrupts are generated in the system. These set the associated bits in the LPIC Interrupt Request Register (IRR).
- 2) The maskable interrupt signal (INTR) is asserted by the LPIC and interrupts the processor. INTR is an active high level.
- 3) The processor generates an interrupt acknowledge bus cycle that flows through the GeodeLink™ system as a single BIZZARO packet. When it reaches the Diverse Logic, it is converted to the two cycle interrupt acknowledge sequence expected by the LPIC.
- 4) The acknowledge operation returns an interrupt vector to the processor that is used to call the appropriate interrupt service routine. Processor interrupts are now disabled at the processor.
- 5) The acknowledge operation also selects the highest priority interrupt from the IRR and uses it to set one bit in the LPIC Interrupt Service Register (ISR). Each acknowledge operation always sets a single ISR bit.

- 6) The acknowledge operation generally de-asserts INTR if there are no higher priority interrupts. However, it is possible that another interrupt is generated in the system anytime after the acknowledge. Any new interrupts will appear in the IRR. If they are higher priority than the current interrupt, then the INTR is re-asserted. Since interrupts are disabled at the processor, INTR remaining high or going high during the interrupt service routine has no effect until interrupts are explicitly enabled again at the processor by the interrupt service routine or implicitly enabled when a return-from-interrupt is executed.
  - 7) The interrupt service routine masks off the interrupt in the LPIC Interrupt Mask Register (IMR). The interrupt service routine interacts with the OS to schedule calls to the drivers associated with the interrupt. If level, one or more drivers could be associated. If edge, only one driver could be associated. The service executes a return-from-interrupt.
  - 8) The OS calls the drivers associated with the interrupt as scheduled. Each driver checks its associated device to determine service needs. If no "need", the driver returns to the OS without any action. If "need", the driver performs the interrupt action, clears the interrupt source, and returns to the OS. When all the scheduled drivers have been called, the OS un-masks the interrupt at LPIC. Note that the individual drivers do not directly interact with LPIC.
- Note in the above procedure that there is not a need to handle "level" and "edge" types separately as long as "edge" types are not shared.
- 3) Eventually, the highest priority service routine is running and INTR is de-asserted. The service calls the driver(s) associated with the interrupt. The driver completes the interrupt "work", clears the interrupt at its system source, and returns to the interrupt service routine.
  - 4) The interrupt service routine disables interrupts at the processor and prepares to return to a lower priority service routine or the initially interrupted process. It writes an end-of-interrupt (EOI) command (020h) to the LPIC OCW2 register. This clears the highest priority ISR bit. One EOI always clears one ISR bit. The service routine executes a return-from-interrupt that enables interrupts again at the processor.
  - 5) It is possible for INTR to assert from the same interrupt as soon as EOI is written. The initial interrupt acknowledge action copies the bit to the ISR. For edge mode, the initial interrupt acknowledge action also clears the bit in the IRR. For level, IRR always reflects the level of the signal on the interrupt port. After the interrupt acknowledge for edge mode, another edge could set the bit in the IRR before the EOI. If in level mode, another shared interrupt could be keeping the input high or potentially the initial interrupt has occurred again, since the driver cleared the source but before the EOI. At any rate, if IRR is high at EOI, INTR will immediately assert again. Hence, the need to disable interrupts at the processor in step 10 above before writing the EOI.
  - 6) Eventually, all system events are serviced and control returns to the originally interrupted program.

### Real-Time Interrupts Approach

The following discussion assumes the "work" associated with the interrupt is performed in the interrupt service routine. The setup and steps 1 through 6 are the same:

- 1) If there is only one driver associated with the interrupt, it is called at this point. If more than one driver (shared), then they could be called in order to determine "need". Alternately, the XIRR could be read to directly identify the source.
- 2) Depending on the event being serviced and the OS policies, the processor will enable interrupts again at some point. Potentially, this will generate another higher priority interrupt causing the current service routine to nest with another interrupt acknowledge cycle. For a nest operation, an additional bit will be set in the ISR.

Note that the above procedure did not use the Interrupt Mask Register (IMR), but variations on the above could have. Lastly note, as in the first discussion, drivers do not directly interact with the LPIC.

## 5.9 Direct Memory Access Module

The Direct Memory Access (DMA) module supports industry standard DMA architecture using two 8237-compatible DMA controllers in cascaded configuration. Figure 5-21 shows the DMA module partitioning. It consists of two standard 8237 DMA controllers, a bus interface, address mapper, and source mapper.

### Features

- 32-bit address range support via high page registers.
- Supports the standard 7-channel DMA configuration, out of which the four 8-bit channels are used.
- DMA mapper to route DMA sources to the four 8-bit DMA channels.
- DMA sources from the LPC bus, and from transmit and receive buffers from the two on-chip UARTs are supported.

- Allow the data bus to be released in between DMA transfers during demand or bulk mode to allow transfers to the DMA module or the module doing DMA transfers.

### 5.9.1 DMA Controllers

The Core Logic module supports seven DMA channels using two standard 8237-equivalent controllers. DMA Controller 1 contains Channels 0 through 3 and supports 8-bit I/O adapters. These channels are used to transfer data between 8-bit peripherals and system memory. Using the high and low page address registers, a full 32-bit address is output for each channel so they can all transfer data throughout the entire 4 GB system address space. Each channel can transfer data in 64 KB pages.

DMA Controller 2 is unused.

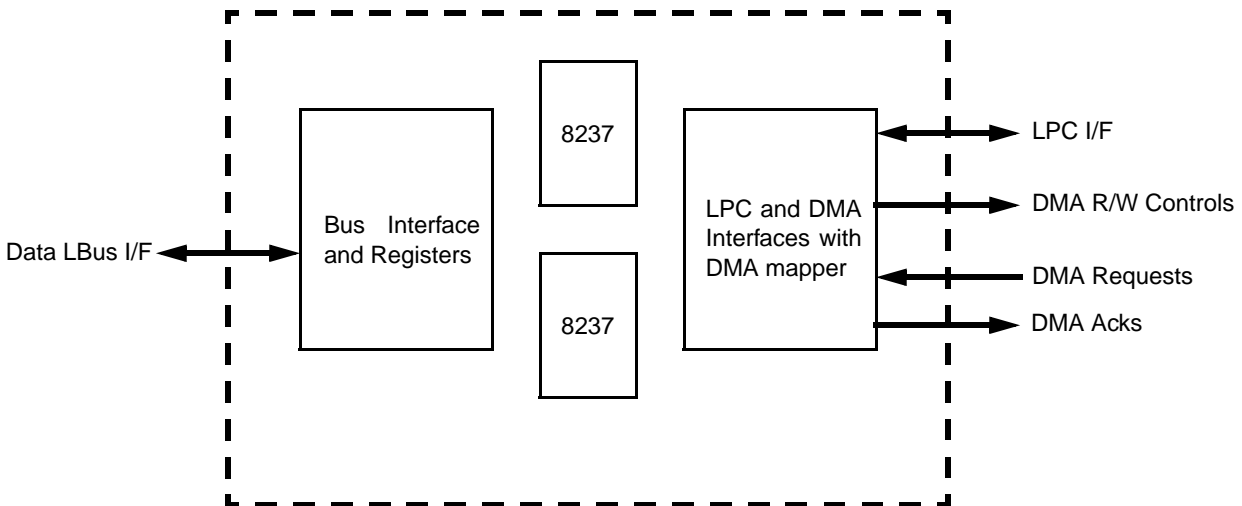


Figure 5-21. DMA Module Block Diagram

### 5.9.2 DMA Transfer Modes

Each DMA channel can be programmed for *single*, *block*, *demand* or *cascade* transfer modes. In the most commonly used mode, *single* transfer mode, one DMA cycle occurs per DRQ and the PCI bus is released after every cycle. This allows the Core Logic module to timeshare the PCI bus with the GX1 module. This is imperative, especially in cases involving large data transfers, because the GX1 module gets locked out for too long.

In *block* transfer mode, the DMA controller executes all of its transfers consecutively without releasing the PCI bus.

In *demand* transfer mode, DMA transfer cycles continue to occur as long as DRQ is high or terminal count is not reached. In this mode, the DMA controller continues to execute transfer cycles until the I/O device drops DRQ to indicate its inability to continue providing data. For this case, the PCI bus is held by the Core Logic module until a break in the transfers occurs.

In the Geode CS5535 companion device design, block and demand transfers behave much like single transfer mode to avoid the lockout problem.

In *cascade* mode, the channel is connected to another DMA controller or to an ISA bus master, rather than to an I/O device. In the Core Logic module, one of the 8237 controllers is designated as the master and the other as the slave. The HOLD output of the slave is tied to the DRQ0 input of the master (Channel 4), and the master's DACK0# output is tied to the slave's HLDA input.

In each of these modes, the DMA controller can be programmed for *read*, *write*, or *verify* transfers.

Both DMA controllers are reset at power-on reset (POR) to *fixed* priority. Since master Channel 0 is actually connected to the slave DMA controller, the slave's four DMA channels have the highest priority, with Channel 0 as highest and Channel 3 as the lowest. Immediately following slave Channel 3, master Channel 1 (Channel 5) is the next highest, followed by Channels 6 and 7.

### 5.9.3 DMA Controller Registers

The DMA controller can be programmed with standard I/O cycles to the standard register space for DMA.

When writing to a channel's address or WORD Count register, the data is written into both the base register and the current register simultaneously. When reading a channel address or WORD Count register, only the current address or WORD Count can be read. The base address and base WORD Count are not accessible for reading.

### 5.9.4 DMA Transfer Types

Each of the seven DMA channels may be programmed to perform one of three types of transfers: *read*, *write*, or *verify*. The transfer type selected defines the method used to transfer a byte or WORD during one DMA bus cycle.

For *read* transfer types, the DMA controller reads data from memory and writes it to the I/O device associated with the DMA channel.

For *write* transfer types, the DMA controller reads data from the I/O device associated with the DMA channel and writes to the memory.

The *verify* transfer type causes the DMA controller to execute DMA transfer bus cycles, including generation of memory addresses, but neither the READ nor WRITE command lines are activated. This transfer type was used by DMA Channel 0 to implement DRAM refresh in the original IBM PC and XT.

### 5.9.5 DMA Priority

The DMA controller may be programmed for two types of priority schemes: *fixed* and *rotate*.

In *fixed* priority, the channels are fixed in priority order based on the descending values of their numbers. Thus, Channel 0 has the highest priority. In *rotate* priority, the last channel to get service becomes the lowest-priority channel with the priority of the others rotating accordingly. This prevents a channel from dominating the system.

The address and WORD Count registers for each channel are 16-bit registers. The value on the data bus is written into the upper byte or lower byte, depending on the state of the internal addressing byte pointer. This pointer can be cleared by the Clear Byte Pointer command. After this command, the first read/write to an address or WORD-count register reads or writes to the low byte of the 16-bit register and the byte pointer points to the high byte. The next read/write to an address or WORD-count register reads or writes to the high byte of the 16-bit register and the byte pointer points back to the low byte.

The DMA controller allows the user to program the active level (low or high) of the DRQ and DACK# signals. Since the two controllers are cascaded together internally on the chip, these signals should always be programmed with the DRQ signal active high and the DACK# signal active low.



### 5.9.6 DMA Shadow Registers

The Diverse Integration Logic module contains shadow registers (see Section 6.13 on page 417) for reading the configuration of the DMA controllers.

### 5.9.7 DMA Addressing Capability

DMA transfers occur over the entire 32-bit address range of the PCI bus. This is accomplished by using the DMA controller's 16-bit memory address registers in conjunction with an 8-bit DMA Low Page register and an 8-bit DMA High Page register. These registers, associated with each channel, provide the 32-bit memory address capability. A write to the Low Page register clears the High Page register, for backward compatibility with the PC/AT standard. The starting address for the DMA transfer must be programmed into the DMA controller registers and the channel's respective Low and High Page registers prior to beginning the DMA transfer.

#### DMA Page Registers and Extended Addressing

The DMA Page registers provide the upper address bits during DMA cycles. DMA addresses do not increment or decrement across page boundaries. Page boundaries for the 8-bit channels (Channels 0 through 3) are every 64 KB and page boundaries for the 16-bit channels (Channels 5, 6, and 7) are every 128 KB.

Before any DMA operations are performed, the Page registers must be written at the I/O Port addresses in the DMA controller registers to select the correct page for each DMA channel. The other address locations between 080h and 08Fh and 480h and 48Fh are not used by the DMA channels, but can be read or written by a PCI bus master. These registers are reset to zero at POR. A write to the Low Page register clears the High Page register, for backward compatibility with the PC/AT standard.

For most DMA transfers, the High Page register is set to zeros and is driven onto PCI address bits AD[31:24] during DMA cycles. This mode is backward compatible with the PC/AT standard. For DMA extended transfers, the High Page register is programmed and the values are driven onto the PCI addresses AD[31:24] during DMA cycles to allow access to the full 4 GB PCI address space.

### 5.9.8 DMA Address Generation

The DMA addresses are formed such that there is an upper address, a middle address, and a lower address portion.

The upper address portion, which selects a specific page, is generated by the Page registers. The Page registers for each channel must be set up by the system before a DMA operation. The DMA Page register values are driven on PCI address bits AD[31:16] for 8-bit channels and AD[31:17] for 16-bit channels.

The middle address portion, which selects a block within the page, is generated by the DMA controller at the beginning of a DMA operation and any time the DMA address increments or decrements through a block boundary. Block sizes are 256 bytes for 8-bit channels (Channels 0 through 3) and 512 bytes for 16-bit channels (Channels 5, 6, and 7). The middle address bits are driven on PCI address bits AD[15:8] for 8-bit channels and AD[16:9] for 16-bit channels.

The lower address portion is generated directly by the DMA controller during DMA operations. The lower address bits are output on PCI address bits AD[7:0] for 8-bit channels and AD[8:1] for 16-bit channels.

### 5.9.9 DMA Mapper Source Selection

For each 8-bit DMA channel, the DMA mapper allows the DMA request to come from a number of sources. Table 5-16 shows how the DMA mapper register select field selects the appropriate DMA source.

When LPC is selected as the DMA source for DMA Channel 0, the source is LPC DMA Channel 0. Similarly, when LPC is selected as the source for DMA Channel 1, 2, or 3, then the DMA sources for those three DMA channels are respectively LPC DMA Channels 1, 2, and 3. Therefore, LPC DMA Channel 0 can only be mapped to DMA Channel 0, LPC DMA Channel 1 can only be mapped to DMA Channel 1, etc.

**Table 5-16. DMA Source Selection**

Source Selector Value from DMA Mapper	DMA Source
0	None (DMA channel off)
1	UART1 Transmit
2	UART1 Receive
3	UART2 Transmit
4	UART2 Receive
5	Reserved (not active)
6	Reserved (not active)
7	LPC

### 5.10 Keyboard Emulation Logic

The Keyboard Emulation Logic (KEL) provides a virtual 8042 keyboard controller interface that is used to map non-legacy keyboard and mouse sources to this traditional interface. For example, Universal Serial Bus (USB) sources are “connected” to this interface via System Management Mode (SMM) software. It also allows mixed environments with one LPC legacy device and multiple new (USB) devices. It produces IRQ and ASMI outputs.

#### Features

- Provides a virtual 8042 keyboard controller interface.
- Allows mixed environments.

- Produces IRQ and ASMI outputs.
- Employs a clock control logic for power management purposes.
- No USB controller required for KEL to operate.

#### 5.10.1 Keyboard Emulation and Port A

The Keyboard Emulation Logic (KEL) with Port A is illustrated in Figure 5-22. Strictly speaking, these are separate functions. However, since they both effect the FA20# (Force processor Address bit 20 to zero when low), the two functions are implemented together. The Keyboard Emulation Logic is the most complex and is discussed first.

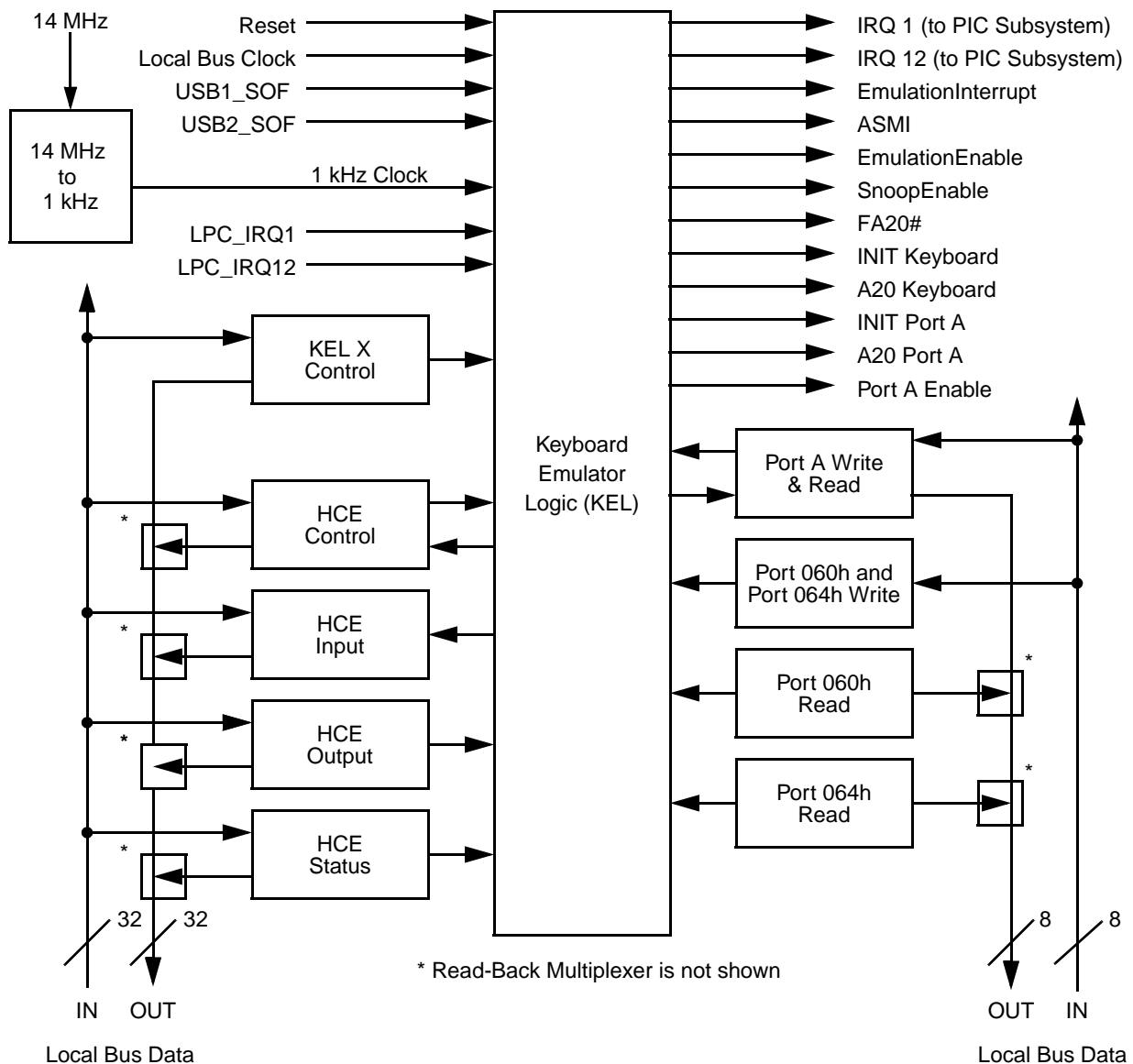


Figure 5-22. KEL Block Diagram

### 5.10.2 Keyboard Emulation Overview

The purpose of the KEL is to model the legacy 8042 keyboard/mouse controller interfaced via legacy I/O Addresses 060h and 064h (also known as Ports 60 and 64). This hardware and supporting processor System Management Mode (SMM) software are designed to support systems that do not have a true PS/2-compatible keyboard and/or mouse interface, but those that have alternative devices performing the equivalent function. Generally, the alternative device is a keyboard or mouse off a USB (Universal Serial Bus) port, but it need not be. Due to the origins of the hardware to be explained shortly, this discussion generally assumes a USB alternative device, but this is not a requirement from a hardware perspective.

The KEL closely models the keyboard emulation hardware detailed in the USB OpenHost Controller Interface specification (OHCI). It is specifically designed to be software compatible with this model. In the USB model, it is part of the USB “Host Controller”, but is logically separate from it. The discussion and description that follows is taken from the OHCI specification, but with modifications to reflect the Geode CS5535 companion device specific implementation.

To support applications and drivers in non-USB-aware environments (e.g., DOS), a peripheral subsystem needs to provide some hardware support for the emulation of a PS/2 keyboard and/or mouse by their USB equivalents (alternative devices). For OHCI, this emulation support is provided by a set of registers that are controlled by code running in SMM. Working in conjunction, this hardware and software produces approximately the same behavior-to-application code as would be produced by a PS/2-compatible keyboard and/or mouse interface.

When data is received from the alternative device, the emulation code is notified and becomes responsible for translating the alternative device keyboard/mouse data into a data sequence that is equivalent to what would be produced by a PS/2-compatible keyboard/mouse interface. The translated data is made available to the system through the legacy keyboard interface I/O Addresses 060h and 064h. Likewise, when data/control is to be sent to the alternative device (as indicated by the system writing to the legacy keyboard interface), the emulation code is notified and becomes responsible for translating the information into appropriate data to be sent to the alternative device.

On the PS/2 keyboard/mouse interface, a read of I/O Address 060h returns the current contents of the keyboard output buffer; a read of I/O Address 064h returns the contents of the keyboard status register. An I/O write to I/O Addresses 060h and 064h puts data into the keyboard input buffer (data is being input into the keyboard subsystem). When emulation is enabled, reads and writes of I/O Addresses 060h and 064h are captured in the KEL HceOutput, HceStatus, and/or HceInput operational registers.

The KEL described here supports a mixed environment in which either the keyboard or mouse is implemented as an

alternative device and the other device is attached to a standard PS/2 interface.

The following sub-sections use the term “alternate device interrupt”. This is an ASMI or IRQ as appropriate for the device; for example the USB can generate either an ASMI or IRQ. The KEL generates a separate ASMI or IRQ.

### 5.10.3 Theory - Keyboard / Mouse Input

When data is received from the alternative device, the emulation code is notified with an alternate device interrupt and translates the keyboard/mouse data into an equivalent PS/2-compatible sequence for presentation to the application software. For each byte of PS/2-compatible data that is to be presented to the applications software, the emulation code writes to the HCE\_Output register. The emulation code then sets the appropriate bits in the HCE\_Status register (normally, OutputFull is set for keyboard data and OutputFull plus AuxOutputFull for mouse data). If keyboard/mouse interrupts are enabled, setting the HCE\_Status register bits causes the generation of an IRQ1 for keyboard data and IRQ12 for mouse data. The emulation code then exits and waits for the next alternate device interrupt.

When the host CPU exits from SMM, it can service the pending IRQ1/IRQ12. This normally results in a read from I/O Address 060h. When I/O Address 060h is read, the KEL intercepts the access and returns the current contents of HCE\_Output. The KEL also clears the OutputFull bit in HCE\_Status and de-asserts IRQ1/IRQ12.

If the emulation software has multiple characters to send to the application software, it sets the CharacterPending bit in the HCE\_Control register. This causes the KEL to generate an ASMI at the beginning of the next frame a time after the application read from I/O Address 060h (HCE\_Output.).

### 5.10.4 Theory - Keyboard Output

Keyboard output is indicated by application software writing data to either I/O Address 060h or 064h. Upon a write to either address, the KEL captures the data in the HCE\_Input register and, except in the case of a FA20# (Force processor Address bit 20 to zero when low) sequence, updates the HCE\_Status register's InputFull and CmdData bits. When the InputFull bit is set, a KEL ASMI is generated at the beginning of the next frame. Upon receipt of the KEL ASMI, the emulation software reads HCE\_Control and HCE\_Status to determine the cause of the emulation interrupt and performs the operation indicated by the data. Generally, this means putting out data to the alternate device.

### 5.10.5 Emulation Events

Emulation Events (EEs) are caused by reads and writes of the emulation registers. EEs generated by the emulation hardware are steered by the KEL to either an ASMI or an Emulation Interrupt. Steering is determined by the EE Routing (EER) bit of the Keyboard Emulation Logic Control Register (KELX\_CTL) (MSR 5140001Fh[1]).

Historically, EEs for data coming from the keyboard/mouse are generated on USB frame boundaries. The KEL is independent of the USB logic, but uses USB frame boundaries for backward compatibility. Alternately, an independent 1 ms counter can be used (MSR 5140001Fh[3:2]). At the beginning of each frame, the conditions that define asynchronous EEs are checked and, if an EE condition exists, the EE is signaled to the host. This has the effect of reducing the number of EEs that are generated for legacy input to no more than 1,000 per second. The number of emulation interrupts is limited because the maximum rate of data delivery to an application cannot be more than 1,000 bytes per second. A benefit of this rule is that, for normal keyboard and mouse operations, only one EE is required for each data byte sent to the application. Additionally, delay of the EE until the next Start of Frame causes data persistence for keyboard input data that is equivalent to that provided by an 8042.

### 5.10.6 Theory - KEL EEs

There are three EEs that produce the signal KEL ASMI. These three EEs are: Character Pending, Input Full, and External IRQ. An A20 sequence is a possible Input Full EE. The A20Sequence bit in the HCE\_Control register (KEL Memory Offset 100h[5]) will be set in this case. The signal KEL ASMI is an active high pulse one Local bus clock in width and sent to the Diverse Integration Logic (DIVIL). This signal is only asserted when the EmulationEnable bit in HCE\_Control is high (KEL Memory Offset 100h[0] = 1), that is, emulation is enabled. For an EE, KEL also optionally produces an Emulation IRQ (KEL\_EMU\_IRQ). This signal is a level and is only asserted when the EE Routing (EER) bit in MSR\_KELX\_CTL (MSR 5140001Fh[1]) is low. De-asserting KEL\_EMU\_IRQ requires clearing the appropriate bit in the HCE\_Control or KEL HCE\_Status registers (KEL Memory Offset 100h and 10Ch).

For the keyboard A20Sequence, KEL sets the KEL\_A20\_ASMI\_FLAG if enabled in the DIVIL.

Keyboard INIT and A20 are generated as appropriate when emulation is enabled or snoop is enabled in MSR\_KELX\_CTL. KEL ASMI is generated as appropriate when emulation is enabled. KEL ASMI is not generated when emulation is disabled and snoop is enabled. Keyboard A20 under snoop does not require service beyond the DIVIL GLD\_MSR\_SMI (MSR 51400002h); that is, KEL does not need to be manipulated. The InputFull bit in

HCE\_Status (KEL Memory Offset 10Ch[1]) will set, but does not require service. Each new keyboard A20Sequence will set the KEL\_INIT\_ASMI\_FLAG if enabled.

If a write to Port A changes the value of bit 1, the KEL sets the PORTA\_A20\_ASMI\_FLAG if enabled in the DIVIL. If bit 0 of Port A is written to a 1, KEL sets the PORTA\_INIT\_ASMI\_FLAG if enabled in the DIVIL. It also sets Port A to the value 2; that is, only bit 1 is high. The A20State bit in HCE\_Control (KEL Memory Offset 100h[8]) is not effected.

The rate of application software reading of I/O Address 060h is dependent on the alternate device interrupt rate or SOFEVENT (Start of Frame Event, MSR 5140001Fh[3:2]) when the CharacterPending bit is used in the HCE\_Control register (KEL Memory Offset 100h[2]). There is one KEL EE per application software read of I/O Address 060h when CharacterPending is set.

The rate of application software writing of I/O Addresses 060h and 064h is no greater than SOFEVENT. Generally, there is one KEL EE per application write to I/O Address 060h.

SOFEVENT is used to emulate normal delays associated with a real 8042 controller and PS/2 device. Its source is established via MSR 5140001Fh[3:2]. Its value is 1 ms frame interval.

### 5.10.7 Theory - Mixed Environment

A mixed environment is one in which an alternate device and a PS/2 device are supported simultaneously (e.g., a USB keyboard and a PS/2 mouse). The mixed environment is supported by allowing the emulation software to control the PS/2 interface. Control of this interface includes capturing I/O accesses to I/O Addresses 060h and 064h and also includes capture of interrupts from the PS/2 keyboard controller off the LPC. IRQ1 and IRQ12 from the LPC keyboard controller are routed through the KEL. When External-IRQEn in HCE\_Control (KEL Memory Offset 100h[4]) is set, IRQ1 and IRQ12 from the legacy keyboard controller are blocked at the KEL and an ASMI is generated instead. This allows the emulation software to capture data coming from the legacy controller and presents it to the application through the emulated interface. The behavior of IRQ1 and IRQ12 with respect to ExternalIRQEn and IRQEN bits is summarized in Table 5-17.

**Table 5-17. KEL Mixed Environment**

Emulation Enable	External IRQEn	IRQEN	LPC_IRQ1	LPC_IRQ12	Output Full	Output FullAux	IRQ1 Active	IRQ12 Active	Action
1	0	1	0	0	1	0	0	0	IRQ1
1	0	1	0	0	0	1	0	0	IRQ12
x	1	0	0	1	0	0	0	1	EE
x	1	0	1	0	0	0	1	0	EE

### 5.10.8 Theory - Force A20 Low Sequence

The FA20 sequence occurs frequently in DOS applications. Mostly, the sequence is to set FA20 high; that is, do not force address bit 20 to a 0. High is the default state of this signal. To reduce the number of ASMI's caused by the A20 sequence, KEL generates an ASMI only if the GateA20 sequence would change the state of A20.

The A20 sequence is initiated with a write of D1h to I/O Address 064h. On detecting this write, the KEL sets the A20Sequence bit in HCE\_Control (KEL Memory Offset 100h[5]). It captures the data byte in HCE\_Input (KEL Memory Offset 104h[7:0]), but does not set the InputFull bit in HCE\_Status (KEL Memory Offset 10Ch[1]). When A20Sequence is set, a write of a value to I/O Address 060h that has bit 1 set to a value different than A20State in HCE\_Control (KEL Memory Offset 100h[8]) causes InputFull to be set and causes an ASMI. An ASMI with both InputFull and A20Sequence set indicates that the application is trying to change the setting of FA20 on the keyboard controller. However, when A20Sequence is set, and a write of a value to I/O Address 060h that has bit 1 set to the same value as A20State in HCE\_Control is detected, then no ASMI will occur.

As mentioned above, a write to I/O Address 064h of any value other than D1h causes A20Sequence to be cleared. If A20Sequence is active and a value of FFh is written to I/O Address 064h, A20Sequence is cleared but InputFull is not set. A write of any value other than D1h or FFh causes InputFull to be set, which then causes an ASMI. A write of FFh to I/O Address 064h when A20Sequence is not set causes InputFull to be set. The current value of the A20\_Mask is maintained in two unconnected places. The A20State bit in HCE\_Control and bit 1 in Port A. The value of A20State is only changed via a software write to HCE\_Control. It is set to 0 at reset. The value of bit 1 in Port A changes on any write to Port A. From reset PortA[1] is 1.

### 5.10.9 Theory - Processor Initialize Sequence

The processor initialization sequence is possible if either of the following cases is true:

- A write of a value fed to I/O Address 064h indicates processor initialization (INIT) or warm reset. This sets KEL\_INIT\_ASMI\_FLAG if enabled in the DIVIL. All HCE registers and Port A are not effected.
- Port A initialization, INIT will respond to: Write 01h to I/O Address 092h. (Refer to Section 5.10.10 "Port A".)

### 5.10.10 Port A

This register is at I/O Address 092h. It can also be used to change the state of A20 or to cause an INIT. When 8-bit data that has its bit 0 set to 1 is written, it causes an INIT. However, if bit 1 of the 8-bit data is set to 1, it causes a change in the state of A20 (A20 gets asserted). As above, an ASMI is only generated on an INIT or A20 event. The INIT operation always forces A20 high. Writes to bits 2 and

higher are a "don't care". Reads to Port A always return 00h or 02h depending on the state of the bit 1 of Port A.

Note that A20 can be changed with Port A or the GateA20 sequence. Another important point is that A20State in HCE\_Control and bit 1 in Port A are independent from each other. Writing a 1 to Port A bit 1 does not effect the A20State bit. Changing the state of the A20State bit does not effect the bit 1 of Port A.

Note that when A20 has a value of 0, it means that the second MB wraps to the first MB. However, a value of 1 means that A20 is not modified.

The following statements summarize the above INIT and A20 sequences :

INIT will respond to: Write 01h to I/O Address 092h or FEh to I/O Address 064h.

A20 toggle will respond to: Write 02h to Address 092h or Write 00h to Address 092h (bit 1 toggles, bit 0 held at 0), and Write D1h to I/O Address 064h then write a value to I/O Address 060h that has bit 1 set to a value different than the A20State in HceControl register. Trapping will insure the SMI is taken on the instruction boundary.

A Keyboard INIT will not respond to: Write D1h to I/O Address 064h followed by a write 02h to I/O Address 060h (set bit 0 to 0).

### 5.10.11 Keyboard Emulation Logic MSRs

In addition to HCE\_Control (KEL Memory Offset 100h), there is a KEL Extended Control MSR, MSR\_KELX\_CTL (MSR 5140001Fh), to provide additional features.

A "snoop" feature is used when an external LPC based keyboard controller is used (while the KEL is not enabled). All I/O accesses to I/O Addresses 060h and 064h proceed to the LPC, but the KEL snoops or watches for the A20 and INIT sequences. If these occur, KEL sets KEL\_A20\_ASMI\_FLAG or KEL\_INIT\_ASMI\_FLAG in the DIVIL if enabled.

EEs may be routed such that they generate an IRQ or ASMI. In the case of Emulation IRQ, the clearing is done by an operation on the appropriate HCE\_Control or HCE\_Status registers. Reading the EE routing bit is not required for emulation processing via IRQ. This bit does not effect ASMI's associated with A20 and INIT operations. All ASMI signals are a single clock pulse wide.

SOFEVENT (Start of Frame Event) is established with bits [3:2] of MSR 5140001Fh. These bits provide alternative sources for SOFEVENT. The SOFEVENT can be sourced from USB1, USB2, or the PIT. A 00 value selects the test mode.

The Port A enable bit is a mask bit for Port A and its default state is high.

### 5.10.12 Related Diverse Device Functions

FA20# and INIT are not passed directly to the processor. SSM code manipulates equivalent functions in the processor.

The HCE registers are considered part of the USB operational register set for some software and hence share the same memory mapped register space. The GLIU descriptor for the USBs, MSR\_LBAR\_KEL1, and MSR\_LBAR\_KEL2 must all be set to the same base. The GLIU will route accesses at memory offset 100h and above to the Diverse Device and accesses below 100h to the USB.

The address decoder in the DIVIL routes accesses to I/O Addresses 060h and 064h to the KEL or LPC based on the value of EmulationEnable in HCE\_Control (KEL Memory Offset 100h). If snoop mode is enabled and the Emulation-Enable bit is not set, writes are made directly to both the KEL and LPC.

The LPC IRQ1 and IRQ12 outputs are connected to both KEL and the PIC subsystem. Masking logic in the subsystem allows the LPC interrupts to be used directly or the KEL set can be used.

The KEL ASMI is routed through the Diverse Device's Standard GLD\_MSR\_SMI (MSR 51400002h). It may be masked off there, but it is only cleared via MSR\_KELX\_CTL (MSR 5140001Fh).

### 5.10.13 Emulation Event Decode

Emulation Events are of two types: frame synchronous and asynchronous. The conditions for a frame synchronous interrupt are sampled by the KEL at each SOF interval and, if an event condition exists, it is signaled at that time. For asynchronous events, the event is signaled as soon as the condition exists.

The equation for the synchronous Emulation Event condition is:

```
synchronousEvent =
HCE_Control.EmulationEnable
    (KEL Memory Offset 100h[0])
and
HCE_Control.CharacterPending
    (KEL Memory Offset 100h[2])
and not
HCE_Status.OutputFull
    (KEL Memory Offset 10Ch[0]).
```

When this decode is true, an Emulation Event is generated at the next SOF. The Event condition is latched until the decode becomes false.

The equation for the asynchronous Emulation Event condition is:

```
asynchronousEvent =
HCE_Control.EmulationEnable
    (KEL Memory Offset 100h[0])
and
HCE_Status.InputFull
    (KEL Memory Offset 10Ch[1]),
or
HCE_Control.ExternIRQEn
    (KEL Memory Offset 100h[4])
and
HCE_Control.IRQ1Active
    (KEL Memory Offset 100h[6])
or
HCE_Control.IRQ12Active
    (KEL Memory Offset 100h[7]).
```

## 5.11 System Management Bus Controller

The System Management Bus (SMB) Controller is a two-wire synchronous serial interface compatible with the System Management Bus physical layer. The SMB Controller is also compatible with Intel's SMBus and Philips' I<sup>2</sup>C. The SMB Controller can be configured as a bus master or slave, and can maintain bidirectional communication with both multiple master and slave devices. As a slave device, the SMB Controller may issue a request to become the bus master.

The SMB Controller allows easy interfacing to a wide range of low-cost memories and I/O devices, including: EEPROMs, SRAMs, timers, ADC, DAC, clock chips, and peripheral drivers.

This chapter describes the general SMB Controller functional block. A device may include a different implementation.

A block diagram of the System Management Bus (SMB) Controller is shown Figure 5-23.

The SMB Controller is upward compatible with previous industry standard two-wire interfaces as detailed in Table 5-18 on page 128.

The SMB Controller's protocol uses a two-wire interface for bidirectional communication between the ICs connected to the bus. The two interface lines are the Serial Data Line (SDL) and the Serial Clock Line (SCL). These lines should be connected to a positive supply via an internal or external pull-up resistor, and remain high even when the bus is idle.

Each IC has a unique address and can operate as a transmitter or a receiver (though some peripherals are only receivers).

During data transactions, the master device initiates the transaction, generates the clock signal, and terminates the transaction. For example, when the SMB Controller initiates a data transaction with an attached SMB compliant peripheral, the SMB Controller becomes the master. When the peripheral responds and transmits data to the SMB Controller, their master/slave (data transaction initiator and clock generator) relationship is unchanged, even though their transmitter/receiver functions are reversed.

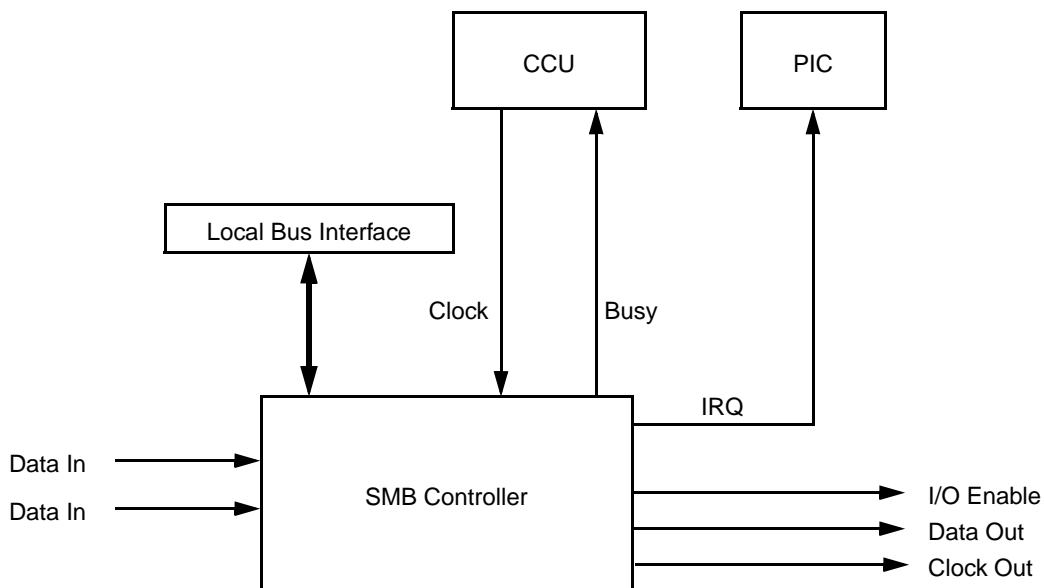


Figure 5-23. SMB Block Diagram

Table 5-18. Comparison of SMB, I<sup>2</sup>C, and ACCESS.bus

Symbol	Parameter	SMB		I <sup>2</sup> C		ACCESS.bus	
		Min	Max	Min	Max	Min	Max
F	Operating frequency	10 kHz	100 kHz	0 kHz	100 kHz	10 kHz	100 kHz
T <sub>BUF</sub>	Bus free time between STOP and START condition	4.7 μs		4.7 μs		4.7 μs	
T <sub>HD:STA</sub>	Hold time after (repeated) START condition. After this period the first clock is generated	4.0 μs		4.0 μs		4.0 μs	
T <sub>SU:STA</sub>	Repeated START condition setup time	4.7 μs		4.7 μs		4.7 μs	
T <sub>SU:STO</sub>	STOP condition setup time	4.0 μs		4.0 μs		4.0 μs	
T <sub>HD:DAT</sub>	Data hold time	300 ns		0 μs	3.45 μs	300 ns	
T <sub>SU:DAT</sub>	Data setup time	250 ns		250 ns		250 ns	
T <sub>Timeout</sub>	Detect clock low time-out	25 ms	35 ms			25 ms	35 ms
T <sub>LOW</sub>	Clock low period	4.7 μs		4.7 μs		4.7 μs	
T <sub>HIGH</sub>	Clock high period	4.0 μs	50 μs	4.0 μs		4.0 μs	50 μs
T <sub>LOW:SEXT</sub>	Cumulative clock low extend period (slave)		25 ms				25 ms
T <sub>LOW:MEXT</sub>	Cumulative clock low extend period (master)		10 ms				10 ms
T <sub>F</sub>	Clock/data fall time		300 ns		300 ns		300 ns
T <sub>R</sub>	Clock/data rise time		1000 ns		1000 ns		1000 ns
TPOR	Time that device must be operational after power-on reset		500 ms				
V <sub>IL</sub>	SMBus signal input low voltage		0.8V	-0.5V	1.5V	-0.5V	0.6V
V <sub>IH</sub>	SMBus signal input high voltage	2.1V	VDD	3.0V		1.4V	5.5V
V <sub>OL</sub>	SMBus signal output low voltage		0.4V	0V	0.4V	0V	0.4V
I <sub>LEAK_BUS</sub>	Input leakage per bus segment	-200 μA	200 μA				
I <sub>LEAK_PIN</sub>	Input leakage per device pin	-10 μA	10 μA	-10 μA	10 μA		10 μA
V <sub>DD</sub>	Nominal bus voltage	2.7V	5.5V			2.0V	5.0V
I <sub>PULLUP</sub>	Current sinking, V <sub>OL</sub> = 0.4V (SMBus)	4.0 mA				100 μA	350 μA
C <sub>BUS</sub>	Capacitive load per bus segment		400 pF				
C <sub>I</sub>	Capacitance for SMBDAT or SMBCLK pin		10 pF		10 pF		
V <sub>NOISE</sub>	Signal noise immunity from 10 to 100 MHz	300 mV p-p					



### 5.11.1 Data Transactions

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (SCL). Consequently, throughout the clock's high period, the data should remain stable (see Figure 5-24). Any changes on the SDA line during the high state of SCL and in the middle of a transaction aborts the current transaction. New data should be sent during the low SCL state. This protocol permits a single data line to transfer both command/control information and data, using the synchronous serial clock.

Each data transaction is composed of a START condition, a number of byte transfers (set by the software), and a STOP condition to terminate the transaction. Each byte is transferred with the most significant bit first, and after each byte (8 bits), an Acknowledge signal must follow. The following subsections provide further details of this process.

During each clock cycle, the slave can stall the master while it handles the previous data or prepares new data. This can be done for each bit transferred, or on a byte boundary, by the slave holding SCL low to extend the clock-low period. Typically, slaves extend the first clock cycle of a transfer if a byte read has not yet been stored, or if the next byte to be transmitted is not yet ready. Some microcontrollers, with limited hardware support for SMB, extend the SMB after each bit, thus allowing the software to handle this bit.

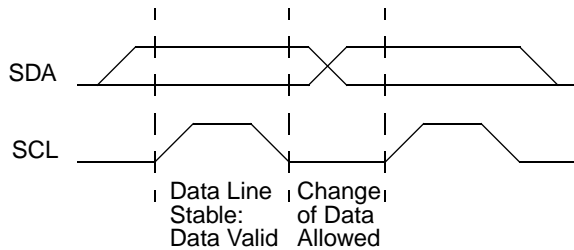


Figure 5-24. SMB Bit Transfer

#### 5.11.1.1 START and STOP Conditions

The SMB master generates START and STOP conditions (control codes). After a START condition is generated, the bus is considered busy and retains this status for a certain time after a STOP condition is generated. A high-to-low transition of the data line (SDA) while the clock (SCL) is high indicates a START condition. A low-to-high transition of the SDA line while the SCL is high indicates a STOP condition (see Figure 5-25).

In addition to the first START condition, a repeated START condition can be generated in the middle of a transaction. This allows another device to arbitrate the bus, or a change in the direction of data transfer.

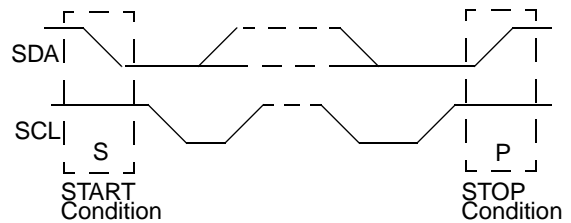


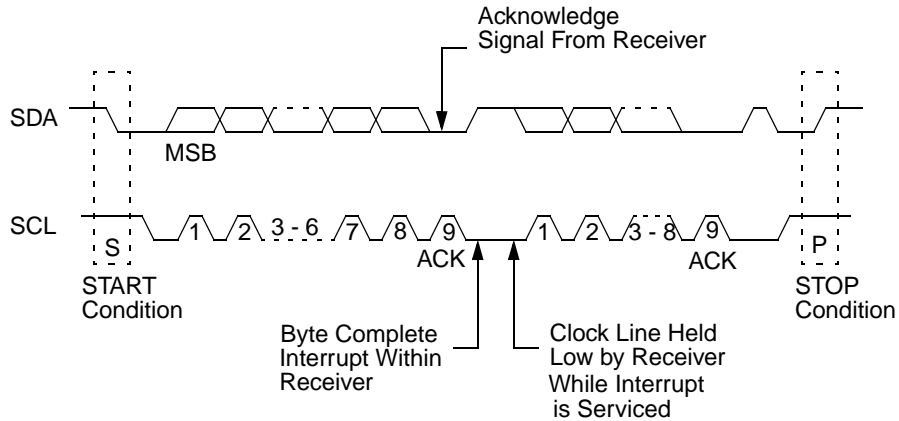
Figure 5-25. SMB START and STOP Conditions

**5.11.1.2 Acknowledge (ACK) Cycle**

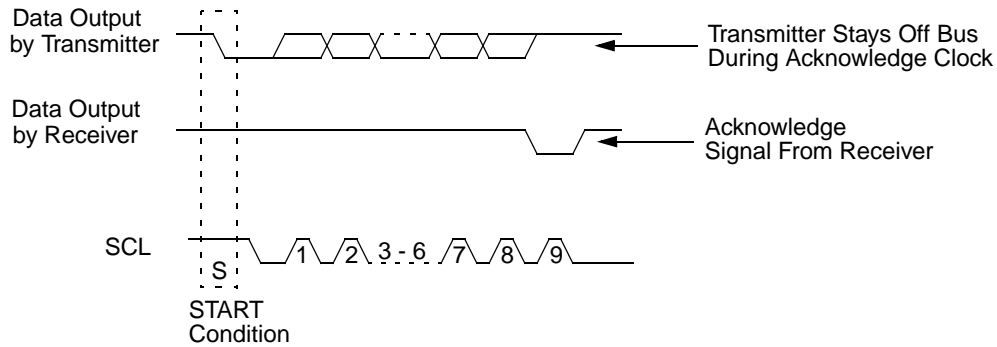
The ACK cycle consists of two signals: the ACK clock pulse sent by the master with each byte transferred, and the ACK signal sent by the receiving device (see Figure 5-26).

The master generates the ACK clock pulse on the ninth clock pulse of the byte transfer. The transmitter releases

the SDA line (permits it to go high) to allow the receiver to send the ACK signal. The receiver must pull down the SDA line during the ACK clock pulse, signalling that it has correctly received the last data byte and is ready to receive the next byte. Figure 5-27 on page 130 illustrates the ACK cycle.



**Figure 5-26. SMB Data Transaction**



**Figure 5-27. SMB Acknowledge Cycle**

### 5.11.1.3 Acknowledge After Every Byte Rule

According to this rule, the master generates an acknowledge clock pulse after each byte transfer, and the receiver sends an acknowledge signal after every byte received. There are two exceptions to this rule:

- When the master is the receiver, it must indicate to the transmitter the end of data by not acknowledging (negative acknowledge) the last byte clocked out of the slave. This negative acknowledge still includes the acknowledge clock pulse (generated by the master), but the SDA line is not pulled down.
- When the receiver is full, otherwise occupied, or a problem has occurred, it sends a negative acknowledge to indicate that it cannot accept additional data bytes.

### 5.11.1.4 Addressing Transfer Formats

Each device on the bus has a unique address. Before any data is transmitted, the master transmits the address of the slave being addressed. The slave device should send an acknowledge signal on the SDA line, once it recognizes its address.

The address consists of the first seven bits after a START condition. The direction of the data transfer ( $R/\bar{W}$ ) depends on the bit sent after the address, the eighth bit. A low-to-high transition during an SCL high period indicates the STOP condition, and ends the transaction of SDA (see Figure 5-28).

When the address is sent, each device in the system compares this address with its own. If there is a match, the device considers itself addressed and sends an acknowledge signal. Depending on the state of the  $R/\bar{W}$  bit (1 = Read, 0 = Write), the device acts either as a transmitter or a receiver.

The SMB protocol allows a general call address to be sent to all slaves connected to the bus. The first byte sent specifies the general call address (00h) and the second byte specifies the meaning of the general call (for example, write slave address by software only). Those slaves that require data acknowledge the call, and become slave receivers; other slaves ignore the call.

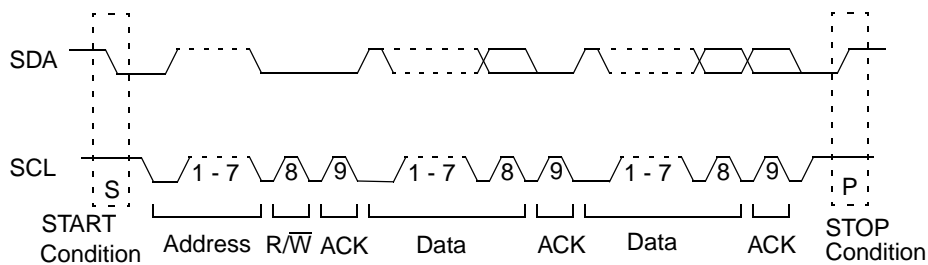


Figure 5-28. SMB Complete Data Transaction

### 5.11.1.5 Arbitration on the Bus

Multiple master devices on the bus require arbitration between their conflicting bus SMB demands. Control of the bus is initially determined according to address bits and clock cycle. If the masters are trying to address the same slave, data comparisons determine the outcome of this arbitration. In master mode, the device immediately aborts a transaction if the value sampled on the SDA line differs from the value driven by the device. (An exception to this rule is SDA while receiving data. The lines may be driven low by the slave without causing an abort.)

The SCL signal is monitored for clock synchronization and to allow the slave to stall the bus. The actual clock period is set by the master with the longest clock period, or by the slave stall period. The clock high period is determined by the master with the shortest clock high period.

When an abort occurs during the address transmission, a master that identifies the conflict should give up the bus, switch to slave mode, and continue to sample SDA to check if it is being addressed by the winning master on the bus.

### 5.11.1.6 Master Mode

This discussion and Section 5.11.1.7 "Slave Mode" reference several bits in the SMB Native register set (e.g., SMBCTL1.STASTRE, SMBST.MASTER, etc.). Table 5-19 provides the bit map for the SMB Native registers for the reader's convenience. For full bit descriptions, refer to Section 6.11.1 "SMB Native Registers" on page 371.

#### Requesting Bus Mastership

An SMB transaction starts with a master device requesting bus mastership. It asserts a START condition, followed by the address of the device that wants the bus. If this transaction is successfully completed, the software may assume that the device has become the bus master.

For the device to become the bus master, the software should perform the following steps:

- 1) Configure SMBCTL1.INTEN to the desired operation mode (Polling = 0 or Interrupt = 1) and set SMBCTL1.START. This causes the SMB Controller to issue a START condition on the bus when the bus becomes free (SMBCST.BB is cleared, or other conditions that can delay START). It then stalls the bus by holding SCL low.
- 2) If a bus conflict is detected (i.e., another device pulls down the SCL signal), SMBST.BER is set.
- 3) If there is no bus conflict, SMBST.SDAST and SMBST.MASTER are set.
- 4) If SMBCTL1.INTEN is set and either SMBST.BER or SMBST.SDAST is set, an interrupt is issued.

#### Sending the Address Byte

When the device is the active master of the bus (SMBST.MASTER is set), it can send the address on the bus.

The address sent should not be the device's own address, as defined by the ADDR bits of the SMBADDR register if the SMBADDR.SAEN is set, nor should it be the global call address if the SMBCST.GCMTCH is set.

To send the address byte, use the following sequence:

- 1) For a receive transaction, where the software wants only one byte of data, it should set SMBCTL1.ACK. If only an address needs to be sent or if the device requires stall for some other reason, set the SMBCTL1.STASTRE.
- 2) Write the address byte (7-bit target device address) and the direction bit to SMBSDA. This causes the SMB Controller to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to SMBST.NEGACK. During the transaction, the SDA and SCL lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, SMBST.BER is set, and SMBST.MASTER is cleared.

**Table 5-19. SMB Native Registers Map**

SMB I/O Offset	Name	7	6	5	4	3	2	1	0
00h	SMBSDA	SMBSDA							
01h	SMBST	SLVSTP	SDAST	BER	NEGACK	STASTR	NMATCH	MASTER	XMIT
02h	SMBCST	RSVD		TGSCL	TSDA	GCMTCH	MATCH	BB	BUSY
03h	SMBCTL1	STASTRE	NMINTE	GCMEN	ACK	RSVD	INTEN	STOP	START
04h	SMBADDR	SAEN	SMBADDR						
05h	SMBCTL2	SCLFRQ	EN						
06h	SMBCTL3	SCLFRQ	EN						

- 3) If SMBCTL1.STASTRE is set and the transaction was successfully completed (i.e., both SMBST.BER and SMBST.NEGACK are cleared), the STASTR bit is set. In this case, the SMB Controller stalls any further bus operations (i.e., holds SCL low). If SMBCTL1.INTEN is set, it also sends an interrupt request to the host.
- 4) If the requested direction is transmit and the START transaction was completed successfully (i.e., neither SMBST.NEGACK nor SMBST.BER is set, and no other master has arbitrated the bus), SMBST.SDAST is set to indicate that the SMB Controller awaits attention.
- 5) If the requested direction is receive, the START transaction was completed successfully and SMBCTL1.STASTRE is cleared, the SMB Controller starts receiving the first byte automatically.
- 6) Check that both SMBST.BER and SMBST.NEGACK are cleared. If SMBCTL1.INTEN is set, an interrupt is generated when either SMBST.BER or SMBST.NEGACK is set.

#### Master Transmit

After becoming the bus master, the device can start transmitting data on the bus.

To transmit a byte in an interrupt or polling controlled operation, the software should:

- 1) Check that both SMBST.BER and SMBST.NEGACK are cleared, and that SMBST.SDAST is set. If SMBCTL1.STASTRE is set, also check that the SMBST.STASTR is cleared (and clear it if required).
- 2) Write the data byte to be transmitted to SMBSDA.

When either SMBST.NEGACK or SMBST.BER is set, an interrupt is generated. When the slave responds with a negative acknowledge, SMBST.NEGACK is set and SMBST.SDAST remains cleared. In this case, if SMBCTL1.INTEN is set, an interrupt is issued.

#### Master Receive

After becoming the bus master, the device can start receiving data on the bus.

To receive a byte in an interrupt or polling operation, the software should:

- 1) Check that SMBST.SDAST is set and that SMBST.BER is cleared. If SMBCTL1.STASTRE is set, also check that SMBST.STASTR is cleared (and clear it if required).
- 2) Set SMBCTL1.ACK, if the next byte is the last byte that should be read. This causes a negative acknowledge to be sent.
- 3) Read the data byte from SMBSDA.

Before receiving the last byte of data, set SMBCTL1.ACK. Before generating a STOP condition or generating a repeated START condition, it is necessary to perform an SDA read and clear the SMBST.SDAST bit.

#### Master STOP

To end a transaction, set SMBCTL1.STOP before clearing the current stall flag (i.e., the SDAST, NEGACK, or STASTR bit of SMBST). This causes the SMB to send a STOP condition immediately, and to clear SMBCTL1.STOP. A STOP condition may be issued only when the device is the active bus master (SMBST.MASTER is set).

#### Master Bus Stall

The SMB Controller can stall the bus between transfers while waiting for the host response. The bus is stalled by holding the SCL signal low after the acknowledge cycle. Note that this is interpreted as the beginning of the following bus operation. The user must make sure that the next operation is prepared before the flag that causes the bus stall is cleared.

The flags that can cause a bus stall in master mode are:

- Negative acknowledge after sending a byte (SMBST.NEGACK = 1).
- SMBST.SDAST bit is set.
- SMBCTL1.STASTRE = 1, after a successful START (SMBST.STASTR = 1).

#### Repeated START

A repeated START is performed when the device is already the bus master (SMBST.MASTER is set). In this case, the bus is stalled and the SMB Controller awaits host handling due to: negative acknowledge (SMBST.NEGACK = 1), empty buffer (SMBST.SDAST = 1), and/or a stall after START (SMBST.STASTR = 1).

For a repeated START:

- 1) Set (1) SMBCTL1.START.
- 2) In master receive mode, read the last data item from SMBSDA.
- 3) Follow the address send sequence, as described in "Write the address byte (7-bit target device address) and the direction bit to SMBSDA. This causes the SMB Controller to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to SMBST.NEGACK. During the transaction, the SDA and SCL lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, SMBST.BER is set, and SMBST.MASTER is cleared.
- 4) If the SMB Controller was awaiting handling due to SMBST.STASTR = 1, clear it only after writing the requested address and direction to SMBSDA.

### Master Error Detection

The SMB Controller detects illegal START or STOP conditions (i.e., a START or STOP condition within the data transfer, or the acknowledge cycle) and a conflict on the data lines of the bus. If an illegal condition is detected, SMBST.BER is set and master mode is exited (SMBST.MASTER is cleared).

### Bus Idle Error Recovery

When a request to become the active bus master or a restart operation fails, SMBST.BER is set to indicate the error. In some cases, both the device and the other device may identify the failure and leave the bus idle. In this case, the START sequence may be incomplete and the bus may remain deadlocked.

To recover from deadlock, use the following sequence:

- 1) Clear SMBST.BER and SMBCST.BB.
- 2) Wait for a time-out period to check that there is no other active master on the bus (i.e., SMBCST.BB remains cleared).
- 3) Disable, and re-enable the SMB Controller to put it in the non-addressed slave mode. This completely resets the functional block.

At this point, some of the slaves may not identify the bus error. To recover, the SMB Controller becomes the bus master: it asserts a START condition, sends an address byte, then asserts a STOP condition that synchronizes all the slaves.

#### 5.11.1.7 Slave Mode

A slave device waits in Idle mode for a master to initiate a bus transaction. Whenever the SMB Controller is enabled and it is not acting as a master (i.e., SMBST.MASTER is cleared), it acts as a slave device.

Once a START condition on the bus is detected, the device checks whether the address sent by the current master matches either:

- The SMBADDR.ADDR value if SMBADDR.SAEN = 1, or
- The general call address if SMBCTL1.GCMEN = 1.

This match is checked even when SMBST.MASTER is set. If a bus conflict (on SDA or SCL) is detected, SMBST.BER is set, SMBST.MASTER is cleared and the device continues to search the received message for a match.

If an address match or a global match is detected:

- 1) The device asserts its SDA line during the acknowledge cycle.
- 2) SMBCST.MATCH and SMBST.NMATCH are set. If SMBST.XMIT = 1 (i.e., slave transmit mode) SMBST.SDAST is set to indicate that the buffer is empty.

- 3) If SMBCTL1.INTEN is set, an interrupt is generated if both SMBCTL1.INTEN and SMBCTL1.NMINTE are set.
- 4) The software then reads SMBST.XMIT to identify the direction requested by the master device. It clears SMBST.NMATCH so future byte transfers are identified as data bytes.

### Slave Receive and Transmit

Slave receive and transmit are performed after a match is detected and the data transfer direction is identified. After a byte transfer, the SMB Controller extends the acknowledge clock until the software reads or writes the SMBSDA register. The receive and transmit sequences are identical to those used in the master routine.

### Slave Bus Stall

When operating as a slave, the device stalls the bus by extending the first clock cycle of a transaction in the following cases:

- SMBST.SDAST is set.
- SMBST.NMATCH and SMBCTL1.NMINTE are set.

### Slave Error Detection

The SMB Controller detects an illegal START and STOP conditions on the bus (i.e., a START or STOP condition within the data transfer or the acknowledge cycle). When this occurs, SMBST.BER is set and SMBCST.MATCH and SMBCST.GMATCH are cleared, setting the SMB Controller as an unaddressed slave.

#### 5.11.1.8 Configuration

##### SDA and SCL Signals

The SDA and SCL are open-drain signals. The device permits the user to define whether to enable or disable the internal pull-up of each of these signals.

##### SMB Clock Frequency

The SMB permits the user to set the clock frequency for the System Management Bus clock. The clock is set by the SMBCTL2.SCLFRQ field and the SMBCTL3 register, which determines the SCL clock period used by the device. This clock low period may be extended by stall periods initiated by the SMB or by another System Management Bus device. In case of a conflict with another bus master, a shorter clock high period may be forced by the other bus master until the conflict is resolved.

**5.11.1.9 Transaction Types**

**Byte Write**

Sequence of events (see Figure 5-29):

- 1) START
- 2) Address phase
- 3) Acknowledge
- 4) Word address
- 5) Acknowledge
- 6) Data
- 7) Acknowledge
- 8) STOP

- 6) Data1
- 7) Acknowledge
- 8) Data(n)
- 9) Acknowledge
- 10) Data(n+1)
- 11) Acknowledge
- 12) Data(n+x)
- 13) Acknowledge
- 14) STOP

**Page Write**

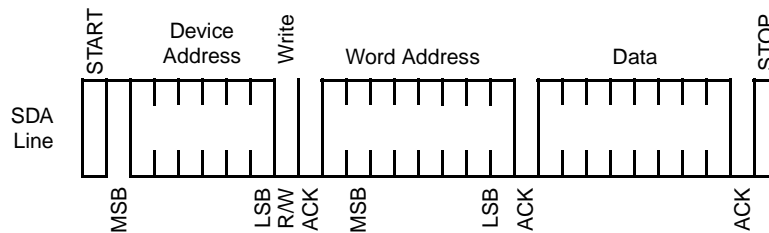
Sequence of Events (see Figure 5-30):

- 1) START
- 2) Address
- 3) Acknowledge
- 4) Word Address
- 5) Acknowledge

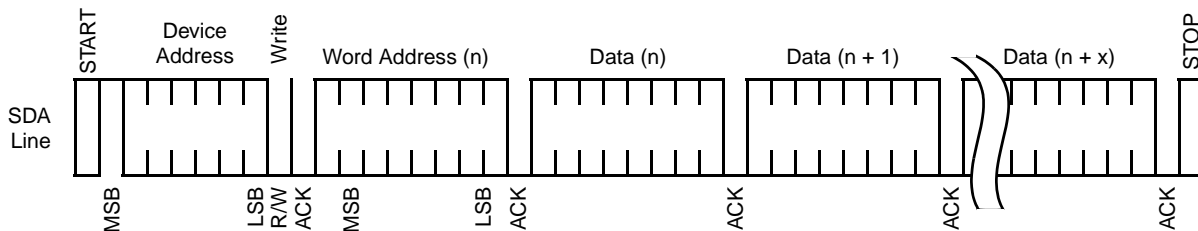
**Current Address Read**

Sequence of Events (see Figure 5-31):

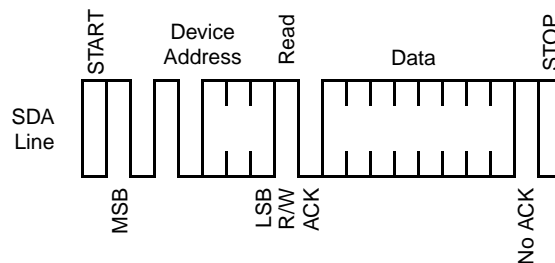
- 1) START
- 2) Device Address - 8 bit
- 3) Acknowledge
- 4) Data
- 5) No Acknowledge
- 6) STOP



**Figure 5-29. SMB Byte Write**



**Figure 5-30. SMB Page Write**



**Figure 5-31. SMB Current Address Read**

**Random Read**

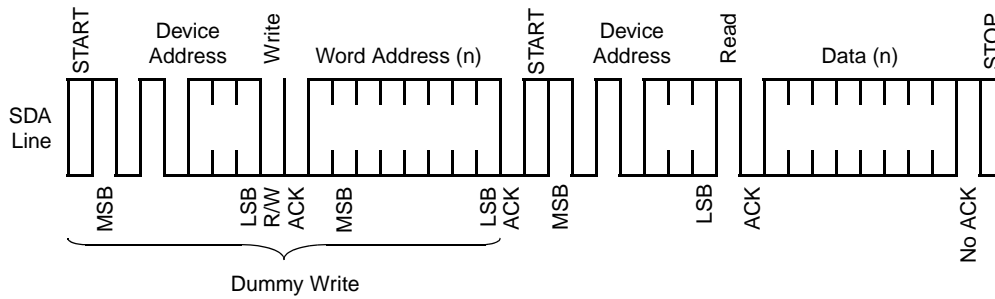
Sequence of Events (see Figure 5-32):

- 1) START
- 2) Device Address
- 3) Acknowledge
- 4) Word Address(n)
- 5) Acknowledge
- 6) START
- 7) Device Address
- 8) Acknowledge
- 9) Data(n)
- 10) No Acknowledge
- 11) STOP

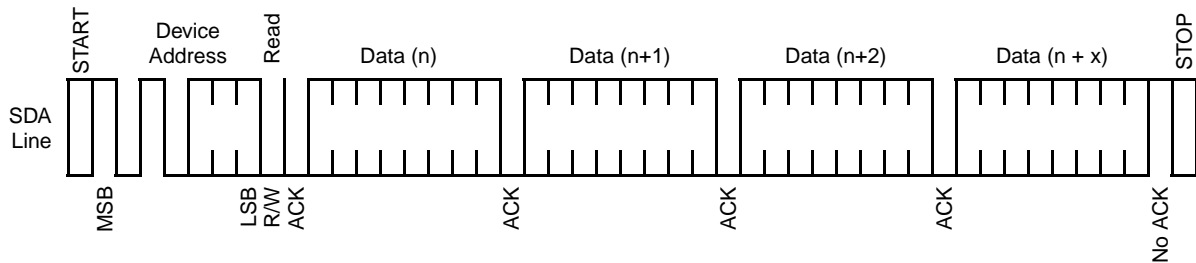
**Sequential Reads**

Sequence of Events (see Figure 5-33):

- 1) START
- 2) Device Address
- 3) Acknowledge
- 4) Data(n)
- 5) Acknowledge
- 6) Data(n+1)
- 7) Acknowledge
- 8) Data(n+2)
- 9) Acknowledge
- 10) Data(n+x)
- 11) No Acknowledge
- 12) STOP



**Figure 5-32. SMB Random Read**



**Figure 5-33. SMB Sequential Reads**



## 5.12 UART and IR Port

The UART and IR Port (UART/IR Controller) is an enhanced serial port with fast IR (infrared). The UART/IR Controller provides advanced, versatile serial communications features with IR capabilities and supports:

- UART (Section 5.12.1.1 "UART Mode" on page 140)
- Sharp-IR (Section 5.12.1.2 "Sharp-IR Mode" on page 140)
- IrDA 1.0 SIR (Section 5.12.1.3 "SIR Mode" on page 140)
- Consumer Electronic IR (CEIR); also called TV Remote or Consumer remote control (Section 5.12.1.4 "CEIR Mode" on page 141)

In UART mode, the functional block can act as a standard 16450 or 16550, or in extended mode.

Existing 16550-based legacy software is completely and transparently supported. Organization and specific fallback mechanisms switch the functional block to 16550 compatibility mode upon reset, or when initialized by 16550 software.

This functional block has two DMA channels, of which the device can use one or both. One channel is required for IR-based applications, since IR communication works in half-duplex fashion. Two channels are normally needed to handle high-speed, full duplex, UART-based applications.

Figure 5-34 shows the serial port connections to the peripheral devices and host, as well as the device configuration.

### Features

- Fully compatible with 16550 and 16450 devices (except modem)
- Extended UART mode
- Sharp-IR
- IrDA 1.0 SIR with up to 115.2 kbaud data rate
- Consumer-IR mode
- UART mode data rates up to 1.5 Mbps
- Full duplex infrared frame transmission and reception
- Transmit deferral
- Automatic fallback to 16550 compatibility mode
- Selectable 16 and 32 level FIFOs
- 12-bit timer for infrared protocol support
- DMA handshake signal routing for either 1 or 2 channels
- Support for power management
- Virtual dongle interface

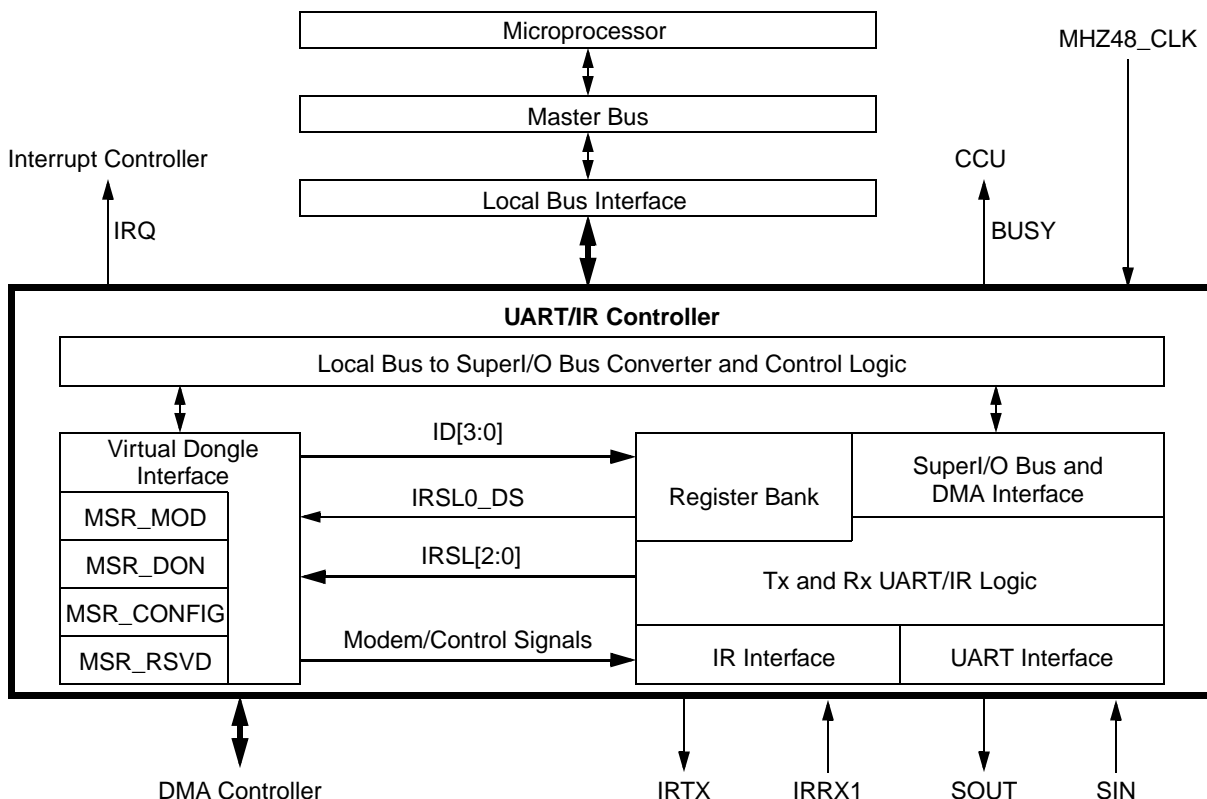


Figure 5-34. UART/IR Overview Diagram

### 5.12.1 Operational Modes

This section describes the operation modes of the UART/IR Controller. Although each mode is unique, certain system resources and features are common.

This discussion references several bits in the UART/IR Controller Native register set. Table 5-20 provides the bit map for the UART/IR Controller Native registers for the reader's convenience. For full bit descriptions, refer to Section 6.12.2 on page 383.

**Table 5-20. UART/IR Controller Native Register Bit Map**

I/O Offset	Name	7	6	5	4	3	2	1	0
<b>Bank 0</b>									
00h	RXD	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
00h	TXD	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
01h	IER (Note 1)	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER (Note 2)	RSVD		TXEMP_IE	DMA_IE	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR (Note 1)	FEN[1:0]		RSVD		RXFT	IPR[1:0]		IPF
	EIR (Note 2)	RSVD		TXEMP_EV	DMA_EV	MS_EV	LS_EV/ TXHLT_EV	TXLDL_EV	RXHDL_IE
	FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR	BKSE	BSR[6:0]						
04h	MCR (Note 1)	RSVD			LOOP	ISEN or DCDLP	RILP	RTS	DTR
	MCR (Note 2)	MDSL[2:0]			IR_PLS	TX_DFR	DMA_EN	RTS	DTR
05h	LSR	ER_INF	TXEMP	TXRDY	BRK	FE	PE	OE	RXDA
06h	MSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
07h	SPR (Note 1)	Scratch Data							
	ASCR (Note 2)	CTE	TXUR	RXACT	RXWDG	RSVD	S_OET	RSVD	RXF_TOUT
<b>Bank 1</b>									
00h	LBGD_L	LBGD[7:0]							
01h	LBGD_H	LBGD[15:8]							
02h	RSVD	RSVD							
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS1	WLS0
	BSR	BKSE	BSR[6:0]						
04-07h	RSVD	RSVD							
<b>Bank 2</b>									
00h	BGD_L	BGD[7:0]							
01h	BGD_H	BGD[15:8]							
02h	EXCR1	RSVD		EDTLBK	LOOP	DMASWP	DMATH	DMANF	EXT_SL
03h	BSR	BKSE	BSR[6:0]						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RF_SIZ[1:0]		TF_SIZ1[1:0]	
05h	RSVD	RSVD							
06h	TXFLV	RSVD			TFL[5:0]				
07h	RXFLV	RSVD			RFL[5:0]				

**Table 5-20. UART/IR Controller Native Register Bit Map (Continued)**

I/O Offset	Name	7	6	5	4	3	2	1	0
<b>Bank 3</b>									
00h	MRID	MID[3:0]			RID[3:0]				
01h	SH_LCR	RSVD	SBRK	STKP	EPS	PEN	STB	WLS1	WLS0
02h	SH_FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							
<b>Bank 4</b>									
00h-01h	RSVD	RSVD							
02h	IRCR1	RSVD				IR_SL[1:0]		RSVD	
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							
<b>Bank 5</b>									
00h-02h	RSVD	RSVD							
03h	BSR	BKSE	BSR[6:0]						
04h	IRCR2	RSVD	RSVD	RSVD	AUX_IRRX	RSVD	RSVD	IRMSSL	IR_FDPLX
05h-07h	RSVD	RSVD							
<b>Bank 6</b>									
00h	IRCR3	SHDM_DS	SHMD_DS	RSVD					
01h	RSVD	RSVD							
02h	SIR_PW	RSVD				SPW3	SPW2	SPW1	SPW0
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							
<b>Bank 7</b>									
00h	IRRXDC	DBW[2:0]			DFR[4:0]				
01h	IRTXMC	MCPW[2:0]			MCFR[4:0]				
02h	RCCFG	R_LEN	T_OV	RXHSC	RCDM_DS	RSVD	TXHSC	RC_MMD[1:0]	
03h	BSR	BKSE	BSR[6:0]						
04h	IRCFG1	STRV_MS	RSVD	SET_IRTX	IRRX1_LV	RSVD	IRIC[2:0]		
05h-06h	RSVD	RSVD							
07h	IRCFG4	RSVD		IRSL0_DS	RXINV	IRSL21_DS	RSVD		

Note 1. Non-Extended Mode.

Note 2. Extended Mode.

### 5.12.1.1 UART Mode

UART mode supports serial data communication with a remote peripheral device using a wired interface. This functional block provides receive and transmit channels that can operate concurrently in full-duplex mode. This functional block performs all functions required to conduct parallel data interchange with the system and composite serial data exchange with the external data channel.

It performs parallel-to-serial conversion on data characters received from the processor or a DMA controller, and serial-to-parallel conversion on data characters received from the serial interface. Figure 5-35 shows the serial data stream. A data character contains five to eight data bits. It is preceded by a START bit and is followed by an optional PARITY bit and a STOP bit. Data is transferred in Little Endian order (LSB first).

UART mode can be implemented in standard 16450 and 16550 compatibility (non-extended) and extended mode. UART 16450 compatibility mode is the default after power-up or reset. When extended mode is selected, the functional block architecture changes slightly and a variety of additional features are made available. The interrupt sources are no longer prioritized, and an Auxiliary Status and Control Register (ASCR) replaces the Scratch Pad Register (SPR). The additional features include: transmitter FIFO (TX\_FIFO) thresholding, DMA capability, and interrupts on transmitter empty states and DMA events.

The clock for both transmit and receive channels is provided by an internal baud generator that divides its input clock by any divisor value from 1 to  $2^{16}-1$ . The output clock frequency of the baud generator must be programmed to be 16 times the baud rate value. The baud generator input clock is derived from a 24 MHz clock through a programmable prescaler. The prescaler value is determined by the PRESL bits in the EXCR2 register. Its default value is 13. This allows all the standard baud rates, up to 115.2 Kbaud, to be obtained. Smaller prescaler values allow baud rates up to 921.6 Kbaud (standard) and 1.5 Kbaud (non-standard).

Before operation can begin, both the communication format and baud rate must be programmed by the software. The communication format is programmed by loading a control byte into the LCR (Link Control Register), while the baud rate is selected by loading an appropriate value into the Baud Generator Divisor Register. The software can read the status of the functional block at any time during operation. The status information includes Full/Empty states for both transmit and receive channels, and any other condition detected on the received data stream, such as a parity error, framing error, data overrun, or break event.

### 5.12.1.2 Sharp-IR Mode

This mode supports bidirectional data communication with a remote device, using IR radiation as the transmission medium. Sharp-IR uses Digital Amplitude Shift Keying (DASK) and allows serial communication at baud rates up to 38.4 Kbaud. The format of the serial data is similar to that of the UART data format. Each data word is sent serially, beginning with a 0 value START bit, followed by up to eight data bits (LSB first), an optional parity bit, and ending with at least one STOP bit, with a binary value of 1. A logical 0 is signalled by sending a 500 kHz continuous pulse train of IR radiation. A logical 1 is signalled by the absence of an IR signal. This functional block can perform the modulation and demodulation operations internally, or can rely on the external optical module to perform them.

Sharp-IR device operation is similar to operation in UART mode. The difference being that data transfer operations are normally performed in half-duplex fashion, and the modem control and status signals are not used. Selection of the Sharp-IR mode is controlled by the Mode Select (MDSL) bits in the MCR when the functional block is in extended mode, or by the IR\_SL bits in the IRCR1 register when the functional block is in non-extended mode.) This prevents legacy software, running in non-extended mode, from spuriously switching the functional block to UART mode when the software writes to the MCR.

### 5.12.1.3 SIR Mode

SIR mode supports bidirectional data communication with a remote device, using IR radiation as the transmit medium. SIR allows serial communication at baud rates up to 115.2 Kbaud. The serial data format is similar to that of the UART data format. Each data word is sent serially, beginning with a 0 value START bit, followed by eight data bits (LSB first), an optional PARITY bit, and ending with at least one STOP bit, with a binary value of 1. A 0 value is signalled by sending a single IR pulse. A 1 value is signalled by the absence of a pulse. The width of each pulse can be either 1.6  $\mu$ s (3/16 the time required to transmit a single bit at 115.2 kbps). This way, each word begins with a pulse at the START bit.

Operation in SIR is similar to that of the UART mode. The difference being that data transfer operations are normally performed in half-duplex fashion. Selection of the IrDA 1.0 SIR mode is controlled by the MDSL bits in the MCR when the UART is in extended mode, or by the IR\_SL bits in the IRCR1 register when the UART is in non-extended mode. This prevents legacy software, running in non-extended mode, from spuriously switching the functional block to UART mode when the software writes to the MCR.

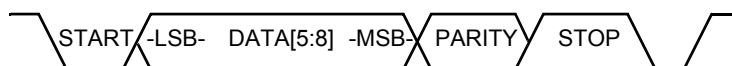


Figure 5-35. UART Serial Data Stream Format

#### 5.12.1.4 CEIR Mode

The Consumer Electronics IR circuitry is designed to optimally support all major protocols presently used in the following remote-controlled home entertainment equipment: RC-5, RC-6, RECS 80, NEC, and RCA. This module, in conjunction with an external optical device, provides the physical layer functions necessary to support these protocols. Such functions include: modulation, demodulation, serialization, de-serialization, data buffering, status reporting, interrupt generation, etc. The software is responsible for the generation of IR code transmitted, and the interpretation of received code.

#### CEIR Transmit Operation

The transmitted code consists of a sequence of bytes that represents either a bit string or a set of run-length codes. The number of bits or run-length codes needed to represent each IR code bit depends on the IR protocol used. The RC-5 protocol, for example, needs two bits or between one and two run-length codes to represent each IR code bit.

Transmission is initiated when the processor or DMA controller writes code bytes into the empty TX\_FIFO. Transmission is completed when the processor sets the S\_EOT bit of the ASCR, before writing the last byte, or when the DMA controller activates the terminal count (TC). Transmission also terminates if the processor simply stops transferring data and the transmitter becomes empty. In this case, however, a transmitter-underrun condition is generated that must be cleared in order to begin the next transmission.

The transmission bytes are either de-serialized or run-length encoded, and the resulting bit-string modulates a carrier signal that is sent to the transmitter LED. The transfer rate of this bit-string, like in UART mode, is determined by the value programmed in the Baud Generator Divisor Register. Unlike a UART transmission, START, STOP, and PARITY bits are not included in the transmitted data stream. A logic 1 in the bit-string keeps the LED off, so no IR signal is transmitted. A logic 0 generates a sequence of modulating pulses that turn on the transmitter LED. Frequency and pulse width of the modulating pulses are programmed by the MCFR and MCPW fields in the IRTXMC register, as well as the TXHSC bit of the RCCFG register.

The RC\_MMD field of RCCFG selects the transmitter modulation mode. If the C\_PLS mode is selected, modulating pulses are generated continuously for the entire logic 0 bit time. If 6\_PLS or 8\_PLS mode is selected, six or eight pulses are generated each time a logic 0 bit is transmitted following a logic 1 bit.

C\_PLS modulation mode is used for RC-5, RC-6, NEC, and RCA protocols. 8\_PLS or 6\_PLS modulation mode is used for the RECS 80 protocol. The 8\_PLS or 6\_PLS mode allows minimization of the number of bits needed to represent the RECS 80 IR code sequence. The current transmitter implementation supports only the modulated modes of the RECS 80 protocol; it does not support the Flash mode.

**Note:** The total transmission time for the logic 0 bits must be equal to or greater than six or eight times the period of the modulation subcarrier, otherwise fewer pulses will be transmitted.

#### CEIR Receive Operation

The CEIR receiver is significantly different from a UART receiver. The incoming IR signals are DASK modulated; therefore, demodulation may be necessary. Also, there are no START bits in the incoming data stream.

The operations performed by the receiver, whenever an IR signal is detected, are slightly different, depending on whether or not receiver demodulation is enabled. If demodulation is disabled, the receiver immediately becomes active. If demodulation is enabled, the receiver checks the carrier frequency of the incoming signal and becomes active only if the frequency is within the programmed range. Otherwise, the signal is ignored and no other action is taken.

When the receiver enters the Active state, the RXACT bit of the ASCR is set to 1. Once in the Active state, the receiver keeps sampling the IR input signal and generates a bit-string, where a logic 1 indicates an Idle condition and a logic 0 indicates the presence of IR energy. The IR input is sampled regardless of the presence of IR pulses at a rate determined by the value loaded into the Baud Generator Divisor Registers. The received bit-string is either de-serialized and assembled into 8-bit characters, or is converted to run-length encoded values. The resulting data bytes are then transferred into the receiver FIFO (RX\_FIFO).

The receiver also sets the RXWDG bit of the ASCR each time an IR pulse signal is detected. This bit is automatically cleared when the ASCR is read. It is intended to assist the software in determining when the IR link has been Idle for a period of time. The software can then stop data from being received by writing a 1 into the RXACT bit to clear it, and return the receiver to the inactive state.

The frequency bandwidth for the incoming modulated IR signal is selected by the DFR and DBW fields in the IRRXDC register. There are two CEIR receive data modes: Oversampled and Programmed T Period. For either mode, the sampling rate is determined by the setting of the Baud Generator Divisor Registers.

Oversampled mode can be used with the receiver demodulator either enabled or disabled. It should be used with the demodulator disabled when a detailed snapshot of the incoming signal is needed; for example, to determine the period of the carrier signal. If the demodulator is enabled, the stream of samples can be used to reconstruct the incoming bit-string. To obtain good resolution, a fairly high sampling rate should be selected.

Programmed T Period mode should be used with the receiver demodulator enabled. The T Period represents one-half bit time for protocols using biphasic encoding or the basic unit of pulse distance for protocols using pulse distance encoding. The baud is usually programmed to match the T Period. For long periods of logic low or high, the receiver samples the demodulated signal at the programmed sampling rate.

When a new IR energy pulse is detected, the receiver synchronizes the sampling process to the incoming signal timing. This reduces timing-related errors and eliminates the possibility of missing short IR pulse sequences, especially with the RECS 80 protocol. In addition, the Programmed T Period sampling minimizes the amount of data used to represent the incoming IR signal, therefore reducing the processing overhead in the host CPU.

#### 5.12.1.5 FIFO Timeouts

Timeout mechanisms are provided to prevent received data from remaining in the RX\_FIFO indefinitely, in case the programmed interrupt or DMA thresholds are not reached.

An RX\_FIFO timeout generates a Receiver Data Ready interrupt and/or a receiver DMA request if bit 0 of the IER register and/or bit 2 of the MCR register (in Extended mode) are set to 1, respectively. An RX\_FIFO timeout also sets bit 0 of the ASCR register to 1 if the RX\_FIFO is below the threshold. When a Receiver Data Ready interrupt occurs, this bit is tested by the software to determine whether a number of bytes indicated by the RX\_FIFO threshold can be read without checking bit 0 of the LSR register.

The conditions that must exist for a timeout to occur in the modes of operation are described below. When a timeout has occurred, it can only be reset when the FIFO is read by the processor or DMA controller.

#### Timeout Conditions for UART, SIR, and Sharp-IR Modes

RX\_FIFO timeout conditions:

- At least one byte is in the RX\_FIFO.
- More than four character times have elapsed since the last byte was loaded into the RX\_FIFO from the receiver logic.
- More than four character times have elapsed since the last byte was read from the RX\_FIFO by the processor or DMA controller.

#### Timeout Conditions for CEIR Mode

The RX\_FIFO timeout in CEIR mode is disabled while the receiver is active. The conditions for this timeout to occur are as follows:

- At least one byte has been in the RX\_FIFO for 64  $\mu$ s or more.
- The receiver has been inactive (RXACT = 0) for 64  $\mu$ s or more.
- More than 64  $\mu$ s have elapsed since the last byte was read from the RX\_FIFO by the processor or DMA controller.

#### 5.12.1.6 Transmit Deferral

This feature allows software to send short, high-speed data frames in PIO mode without the risk of generating a transmitter underrun.

Transmit deferral is available only in extended mode and when the TX\_FIFO is enabled. When transmit deferral is enabled (TX\_DFR bit of the MCR register set to 1) and the transmitter becomes empty, an internal flag is set and locks the transmitter. If the processor now writes data into the TX\_FIFO, the transmitter does not start sending the data until the TX\_FIFO level reaches either 14 for a 16-level TX\_FIFO or 30 for a 32-level TX\_FIFO, at which time the internal flag is cleared. The internal flag is also cleared and the transmitter starts transmitting when a timeout condition is reached. This prevents some bytes from being in the TX\_FIFO indefinitely if the threshold is not reached.

The timeout mechanism is implemented by a timer that is enabled when the internal flag is set and there is at least one byte in the TX\_FIFO. Whenever a byte is loaded into the TX\_FIFO, the timer is reloaded with the initial value. If no byte is loaded for a 64  $\mu$ s time, the timer times out and the internal flag is cleared, thus enabling the transmitter.

### 5.12.1.7 Automatic Fallback to 16550 Compatibility Mode

This feature is designed to support existing legacy software packages, using the 16550 serial port. For proper operation, many of these software packages require that the module look identical to a plain 16550, since they access the serial port registers directly. Because several extended features and new operational modes are provided, make sure the module is in the proper state before executing a legacy program.

The fallback mechanism eliminates the need to change the state when a legacy program is executed following completion of a program that used extended features. It automatically switches the module to 16550 compatibility mode and turns off any extended features whenever the Baud Generator Divisor Register is accessed through the LBGD\_L or LBGD\_H ports in register Bank 1.

In order to avoid spurious fallbacks, baud generator divisor ports are provided in Bank 2. Baud generator divisor access through these ports changes the baud rate setting but does not cause fallback.

New programs designed to take advantage of the extended features should not use LBGD\_L and LBGD\_H to change the baud rate. Instead, they should use BGD\_L and BGD\_H.

A fallback can occur in either extended or non-extended modes. If extended mode is selected, fallback is always enabled. In this case, when a fallback occurs, the following happens:

- TX\_FIFO and RX\_FIFO switch to 16 levels.
- A value of 13 is selected for the baud generator pre-scaler.
- ETDLBK and BTEST of the EXCR1 register are cleared.
- UART mode is selected.
- The functional block switches to non-extended mode.

When fallback occurs from non-extended mode, only the first three of the above actions occur. If either Sharp-IR or SIR infrared modes were selected, no switching to UART mode occurs. This prevents spurious switching to UART mode when a legacy program, running in Infrared mode, accesses the Baud Generator Divisor Register from Bank 1.

Fallback from non-extended mode can be disabled by setting LOCK in the EXCR2 register to 1. When LOCK is set and the functional block is in non-extended mode, two scratch pad registers overlaid with LBGD\_L and LBGD\_H are enabled. Any attempted processor access of the Baud Generator Divisor Register through LBGD\_L and LBGD\_H accesses the scratch pad registers, without affecting the baud rate setting. This feature allows existing legacy programs to run faster than 115.2 Kbaud, without realizing they are running at this speed.

### 5.12.2 Modem Support

An MSR (MSR\_UART[x]\_MOD) (UART1 MSR 51400038h and UART2 MSR 5140003Ch) mimics modem input signals for making it compatible with the software having modem support. The hardware of this module has all the required functionality for modem compatibility.

**5.12.3 Dongle Interface**

The dongle interface on the Geode CS5535 companion device is not a fully hardware compatible interface. The real dongle interface requires six external interface signals and the Geode CS5535 companion device only supports three. With only three signals, the dongle interface supports a subset of the real dongle interface through virtualization.

Configure mode: Change IRSL[2:0] as an output and configure the transceiver for the required mode. If two infrared inputs are required, change IRSLO to an input to give the second receiver channel IRRX2. The IRSL2 and IRSL1 are configured as outputs to keep the transceiver in the required mode.

**5.12.3.1 Real Dongle**

The real dongle interface uses six multiplexed pins for dongle identification, data transfer, and transceiver configuration. Figure 5-36 on page 144 illustrates the real dongle interface and Table 5-21 provides the interface signals and their descriptions.

**Table 5-21. Real Dongle Interface Signals**

Signal Name	Type	Description
IRTX	O	Infrared transmit data
IRRX	I	Infrared receive data
ID0/IRSL0/IRRX2	I/O	Identification signal 0 Infrared mode select 0 Infrared receive data for transceivers with two RX channels
ID1/IRSL1	I/O	Identification signal 1 Infrared mode select 1
ID2/IRSL2	I/O	Identification signal 2 Infrared mode select 2
ID3	I/O	Identification signal 2

Only three signals (IRTX, IRRX, and ID0/IRSL0/IRRX2) are used for the IR interface. It has three phases:

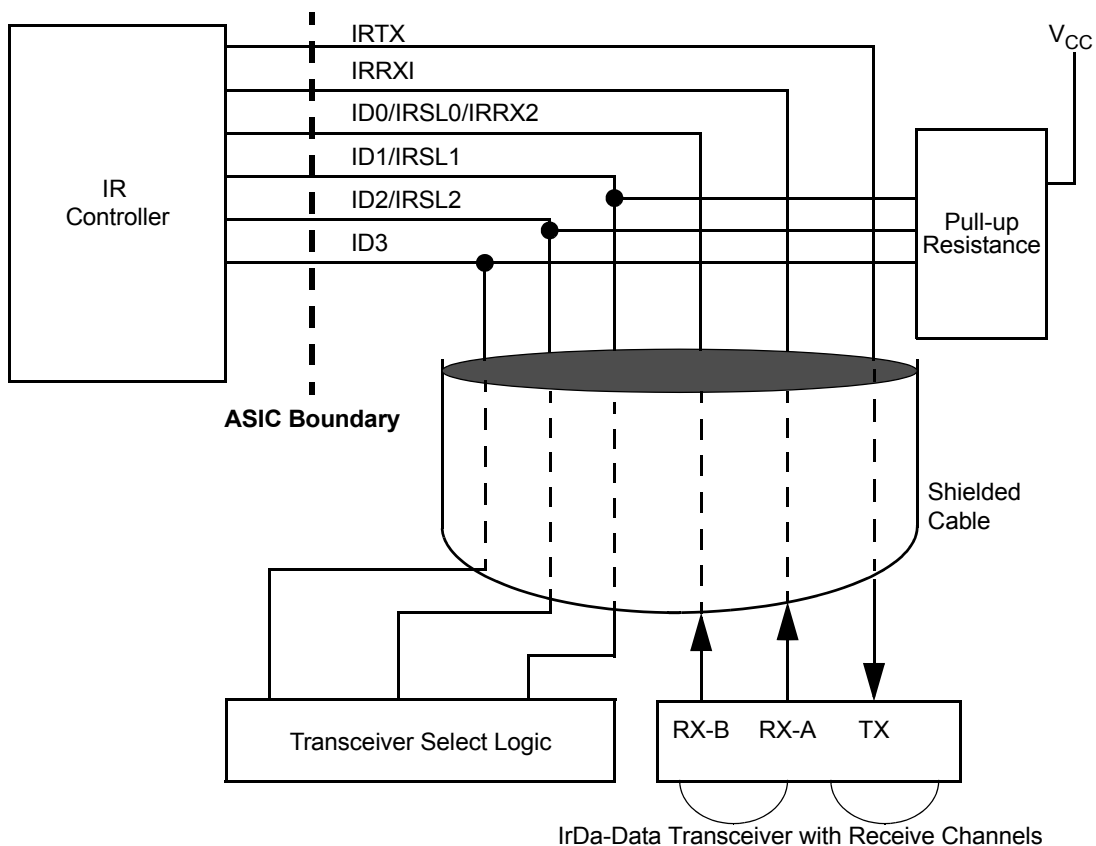
Phase 1:

Change the ID0-ID3 bits to input mode, and read the status to complete primary identification of the dongle.

Phase 2:

Change ID1 and ID2 as output and read the status of ID0 and ID3 to complete the secondary dongle identification phase. This phase provides information about the connected dongle.

Phase 3:



**Figure 5-36. Real Dongle Interface**



### 5.12.3.2 Virtual Dongle

The virtual dongle interface is used due to the unavailability of pins for dongle identification and configuration (see Figure 5-37).

The virtual dongle interface is a method used to run legacy software on the UART/IR Controller. The virtual dongle interface uses dedicated UART/IR MSRs. (See Section 6.12.1 "UART/IR Controller Specific MSRs" on page 380 for complete register and bit formats.) The virtual dongle imitates the real dongle as far as legacy software is concerned, and there are no plug-and-play requirements for IR transceivers.

- The software inputs the dongle's ID to the ID[0:3] bits of MSR\_UART[x]\_MOD (UART1 MSR 51400038h and UART2 MSR 5140003Ch) as the primary ID encoding.
- For dongles that use a non-serial transceiver, it identifies the Consumer IR capabilities. The software should switch ID1 and ID2 to output mode (so they become IRSL1 and IRSL2), IRSL1 and IRSL2 will or will not behave differently (i.e., INV [invert] or NCH [no change]) from the previous step and the software should respond by driving the appropriate level on ID0 and ID3 in the MSR\_UART[x]\_DONG register.

The operational mode of an infrared dongle that uses a non-serial transceiver is selected by the driving the IRSL[2:0] signals.

#### Features

- Uses only three pins to connect to IR transceiver.
- Fully supports legacy software written for real dongle, with some manual intervention.
- All real dongle modes can be supported by changing the MSR.

#### Limitations

- No Plug-and-Play features available.
- IRSL1 and IRSL2 pins need to be tied in the IR transceiver for the required mode.
- MSR contents must be changed when changing the transceiver mode. If BIOS is used to change the MSR contents, it must be a factory setting.
- If the legacy software supports IR transceiver configuration, the contents of IRSL[2:0] are to be read from the MSR and the required bit tying needs to be done in the transceiver board.

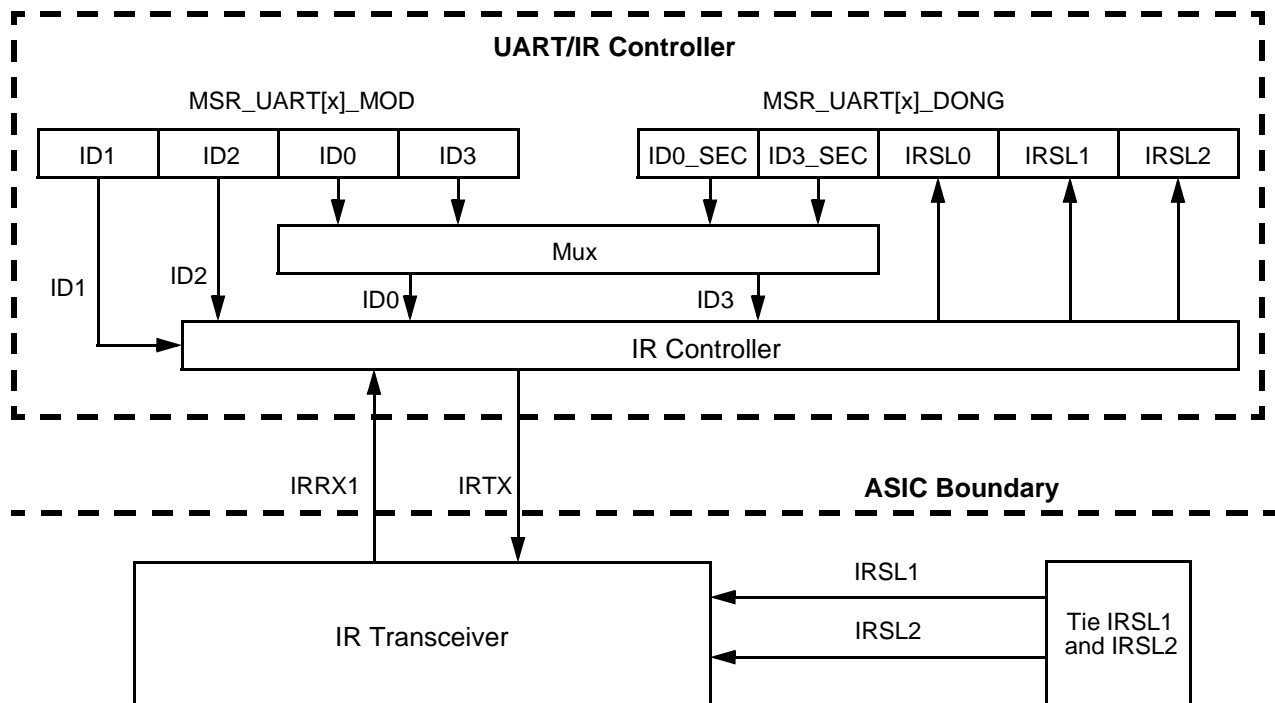


Figure 5-37. Virtual Dongle Interface

### 5.13 Low Pin Count Port

The Low Pin Count port is based on Intel's *Low Pin Count (LPC) Interface Specification v1.0*. In addition to the required pins, the Geode CS5535 companion device also supports two optional pins: LDRQ# and SERIRQ. The LPC interface supports memory, I/O, DMA, and Intel's Firmware Hub Interface. Figure 5-38 shows the block diagram of the LPC port.

#### Features

- Supports memory, I/O, and DMA cycle types.
  - Bus master cycles not supported.
  - CLKRUN# and LPCPD# not supported. SMI# and PME# supported via GPIOs.
  - On-chip DMA transfers through LPC.
  - Supports Intel's FirmWare Hub (FWH) Interface:
    - 5 Signal communication interface supporting byte-at-a-time reads and writes.
    - LAD[3:0] called as FWH0-FWH3 and LFRAME# as FWH4.
- Based on Intel's Low Pin Count (LPC) Specification v1.0.
  - Serial IRQ support.

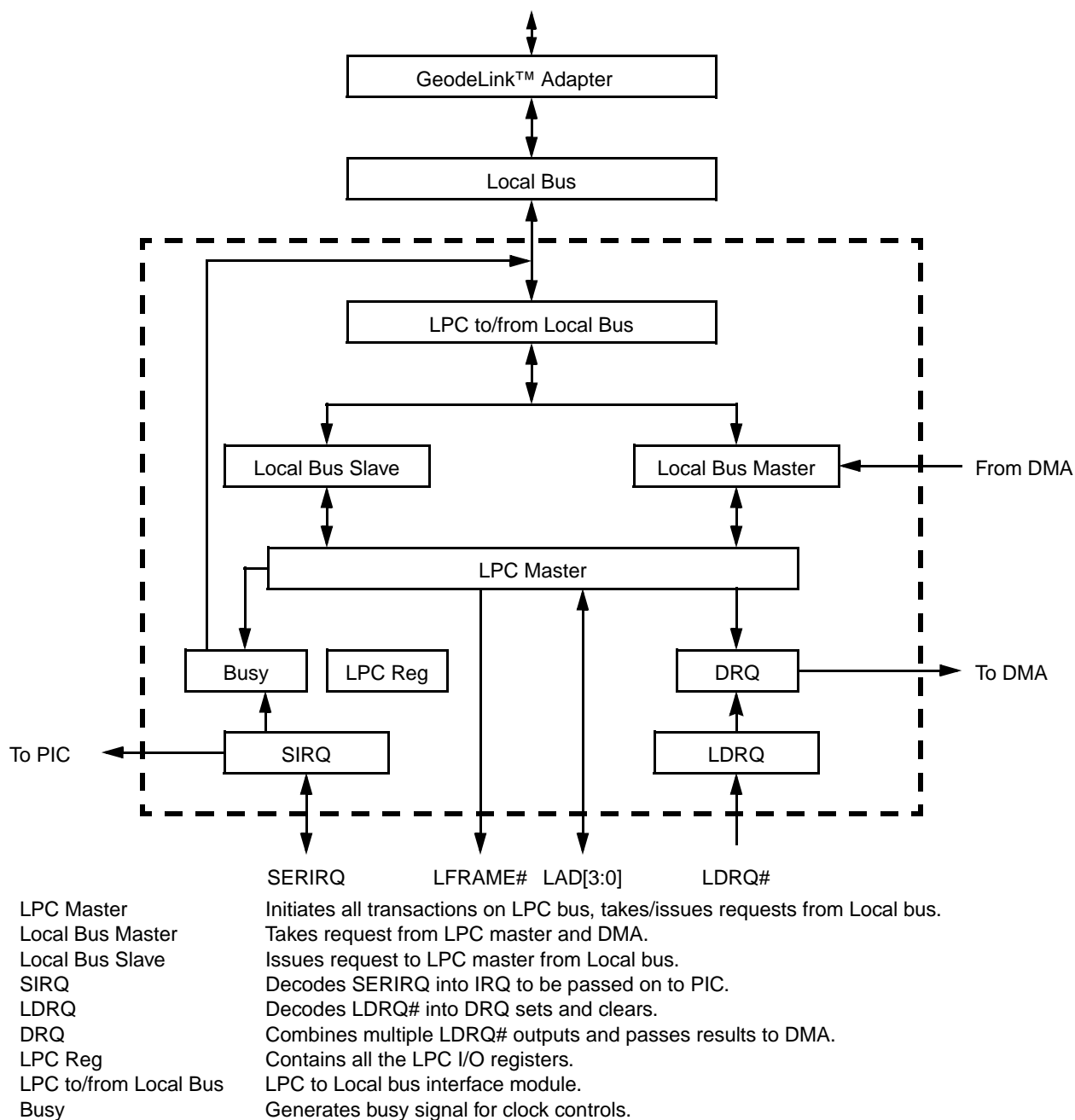


Figure 5-38. LPC Block Diagram

### 5.13.1 LPC Protocol

LPC supports memory read/write, I/O read/write, DMA read/write, and Firmware Hub Interface (see Table 5-22). Data transfers on the LPC bus are serialized over a 4-bit bus.

**Table 5-22. Cycle Types Supported**

Cycle Type	Size	Size Supported
Intel FWH Read	1 Byte	Yes
Intel FWH Write	1 Byte	Yes
Memory Read	1 Byte	Yes
Memory Write	1 Byte	Yes
I/O Read	1 Byte	Yes
I/O Write	1 Byte	Yes
DMA Read	1, 2, 4 Bytes	1 Byte Only
DMA Write	1, 2, 4 Bytes	1 Byte Only
Bus Master Mem Read	1, 2, 4 Bytes	No
Bus Master Mem Write	1, 2, 4 Bytes	No
Bus Master I/O Read	1, 2, 4 Bytes	No
Bus Master I/O Write	1, 2, 4 Bytes	No

LFRAME# is used by the host to start or stop transfers. No peripherals drive this signal. A cycle is started by the host when it drives LFRAME# active and puts information related to the cycle on the LAD[3:0] signals. The host drives information such as address or DMA channel number. For DMA and target cycles, the host drives cycle type (memory or I/O), read/write direction, and size of the transfer. The host optionally drives data, and turns around to monitor peripherals for completion of the cycle. The peripheral indicates the completion of the cycle by driving appropriate values on the LAD[3:0] signals.

The LAD[3:0] signals communicate address, control, and data information over the LPC bus between the host and the peripheral. The information carried on the LAD signals are: start, stop (abort a cycle), transfer type (memory, I/O, DMA), transfer direction (read/write), address, data, wait states, and DMA channel number. The following sections give an overview of fields used. Detailed field descriptions are provided in Table 5-23 on page 148.

**START:** This field indicates the start or stop of a transaction. The START field is valid on the last clock that LFRAME# is active. It is used to indicate a device number, or start/stop indication.

**CYCTYP:** The Cycle Type field is driven by the host when it is performing DMA or target accesses. Bits [3:2] are used for cycle type and bit 1 is used for direction. Bit 0 is reserved.

**SIZE:** This field is one clock. It is driven by the host on memory and DMA transfers to determine how many bytes are to be transferred. Bits [1:0] are used to determine size and bits [3:2] are reserved.

**TAR:** The Turn Around field is two clocks, and is driven by the host when it is turning control over to a peripheral and vice versa. In the first clock a host or a peripheral drives the LAD[3:0] lines to 1111b, on the second cycle the host or peripheral TRI-STATES the LAD[3:0] lines. These lines have weak pull-ups so they will remain at a logical high state.

**ADDR:** The Address field is four clocks for I/O cycles and eight clocks for memory cycles. It is driven by the host on target accesses. This field is not driven on DMA cycles. The most significant nibble is driven first.

**CHANNEL/Terminal Count:** The Channel field is one clock and driven by the host on DMA cycles to indicate the DMA channel. Only 8-bit channels are supported (0, 1, 2, 3). DMA channel is communicated on LAD[2:0] and Terminal Count (TC) is communicated through LAD3. TC indicates the last byte of transfer, based upon the size of the transfer. If an 8-bit transfer and TC is set, then this is the last byte.

**DATA:** This field is two clocks, representing one byte data. It is driven by the host on target and DMA cycles when data is flowing to the peripheral, and by the peripheral when data is flowing to the host. The lower nibble is driven first.

**SYNC:** This field can be several clocks in length and is used to add wait states. Driven by the peripheral on target or DMA cycles.

SYNC Timeout:

- 1) The host starts a cycle, but no device ever drives SYNC valid. If the host observes three consecutive clocks without a valid SYNC, it can abort the cycle.
- 2) The host starts a cycle, a device drives a SYNC valid to insert wait states (LAD[3:0] = 0101b or 0110b), but never completes it. This could happen if the peripheral locks up for some reason. The peripheral should be designed to prevent this case:
  - If the SYNC pattern is 0101b, then the maximum number of SYNC clocks is eight. If the host sees more than eight, it may abort the cycle.
  - If the SYNC pattern is 0110b, then no maximum number of SYNC clocks took place, the peripheral must have protection mechanisms to complete the cycle.

When the host is driving SYNC, it may insert a very large number of wait-states depending on PCI latencies. The peripheral must not assume any timeouts.

SYNC Error Indication: A peripheral can report an error via the LAD[3:0] = 1010b encoding. If the host was reading data from a peripheral, the data will still be transferred in the next two nibbles, even though this data is invalid, the

peripheral must transfer it. If the host was writing, data had already been transferred.

In DMA if it was a multiple byte cycle, an error SYNC terminates the cycle.

For more info on SYNC timeout and SYNC error details, refer to the *LPC Specification*.

**Table 5-23. Cycle Field Definitions: Target Memory, I/O, and DMA**

Field	# Clocks	Comment																						
START	1	<b>Start of Cycle.</b> 0000b indicates a start of a cycle.																						
CYCTYP	1	<p><b>Cycle Type.</b> Indicates the type of cycle.</p> <table border="0"> <tr> <td>Bits [3:0]</td> <td>Definition</td> </tr> <tr> <td>000x</td> <td>I/O Read</td> </tr> <tr> <td>001x</td> <td>I/O Write</td> </tr> <tr> <td>010x</td> <td>Memory Read</td> </tr> <tr> <td>011x</td> <td>Memory Write</td> </tr> <tr> <td>100x</td> <td>DMA Read</td> </tr> <tr> <td>101x</td> <td>DMA Write</td> </tr> <tr> <td>1100</td> <td>Reserved</td> </tr> <tr> <td>1101</td> <td>FWH Read</td> </tr> <tr> <td>1110</td> <td>FWH Write</td> </tr> <tr> <td>1111</td> <td>Reserved</td> </tr> </table>	Bits [3:0]	Definition	000x	I/O Read	001x	I/O Write	010x	Memory Read	011x	Memory Write	100x	DMA Read	101x	DMA Write	1100	Reserved	1101	FWH Read	1110	FWH Write	1111	Reserved
Bits [3:0]	Definition																							
000x	I/O Read																							
001x	I/O Write																							
010x	Memory Read																							
011x	Memory Write																							
100x	DMA Read																							
101x	DMA Write																							
1100	Reserved																							
1101	FWH Read																							
1110	FWH Write																							
1111	Reserved																							
CHANNEL	1	<p><b>Channel #.</b> Used only for DMA cycles to indicate channel number being granted. The LAD[2:0] bits indicate the channel number being granted, and LAD[3] indicates the TC bit. The encoding on LAD[2:0] for channel number is as follows:</p> <table border="0"> <tr> <td>LAD[2:0]</td> <td>Definition</td> </tr> <tr> <td>000</td> <td>I/O Read</td> </tr> <tr> <td>001</td> <td>I/O Write</td> </tr> <tr> <td>010</td> <td>Memory Read</td> </tr> <tr> <td>011</td> <td>Memory Write</td> </tr> <tr> <td>100-111</td> <td>Reserved</td> </tr> </table> <p>Only 8-bit channels are supported.</p>	LAD[2:0]	Definition	000	I/O Read	001	I/O Write	010	Memory Read	011	Memory Write	100-111	Reserved										
LAD[2:0]	Definition																							
000	I/O Read																							
001	I/O Write																							
010	Memory Read																							
011	Memory Write																							
100-111	Reserved																							
TAR	2	<b>Turn-Around.</b> The last component driving LAD[3:0] will drive it high during the first clock and TRI-STATE during the second clock.																						
SIZE	1	<p><b>Size of Transfer.</b> Used only for DMA cycles. Bits [3:0] are reserved and must be ignored by the peripheral.</p> <table border="0"> <tr> <td>LAD[1:0]</td> <td>Definition</td> </tr> <tr> <td>00</td> <td>8-Bit</td> </tr> <tr> <td>01-11</td> <td>Reserved</td> </tr> </table> <p>Only 8-bit is supported for all transfers.</p>	LAD[1:0]	Definition	00	8-Bit	01-11	Reserved																
LAD[1:0]	Definition																							
00	8-Bit																							
01-11	Reserved																							
DATA	1 Byte DMA: 1 Byte	<p><b>Data Phase.</b> The data byte is transferred with the least significant nibble first (D[3:0] on LAD[3:0], then D[7:4] on LAD[3:0]).</p> <p><b>DMA.</b> The data byte is transferred with the least significant nibble first (D[3:0] on LAD[3:0], then D[7:4] on LAD[3:0]). Only one byte data transfer is supported.</p>																						
ADDR	8 for Memory, 4 for I/O	<p><b>Address Phase.</b> Address is 32-bit for memory, 16-bit for I/O. It is transferred most significant nibble first.</p> <p>DMA cycles do not use the ADDR field.</p>																						

**Table 5-23. Cycle Field Definitions: Target Memory, I/O, and DMA (Continued)**

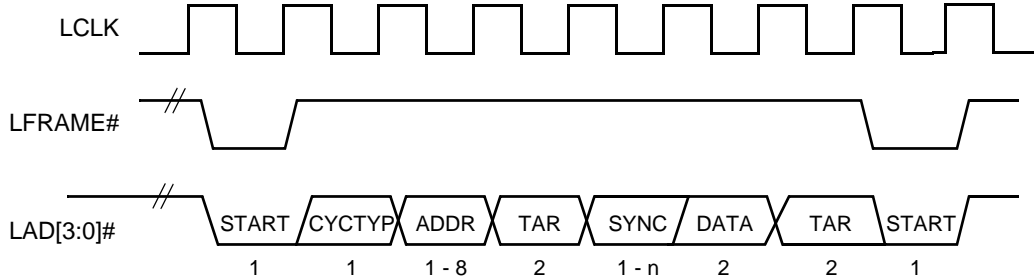
Field	# Clocks	Comment
SYNC	1-N	<p><b>Sync:</b> Allows peripheral or host to synchronize (add wait-states). Generally, the peripheral or host drives 0101 or 0110 until no more wait-states are needed. At that point it will drive 0000. All other combinations are reserved.</p> <p>0000 Sync achieved with no error. DMA. Sync achieved with no error. Also indicates no more transfer desired for that channel, and DMA request is de-asserted.</p> <p>0101 Indicates that Sync not achieved yet, but the part is driving the bus. DMA. Part indicating wait states.</p> <p>0110 Indicates that Sync not achieved yet, but the part is driving the bus, and expect long Sync. DMA. Part indicating wait states, and many wait states will be added.</p> <p>1010 Special case. Peripheral indicating errors, see sync section in protocol overview. DMA. Sync achieved with error. Also indicates no more transfers desired for that channel, and DMA request is de-asserted.</p> <p>1001 DMA (only). Sync achieved with no error and more DMA transfer desired to continue after this transfer.</p>

**5.13.2 Cycle Protocol**

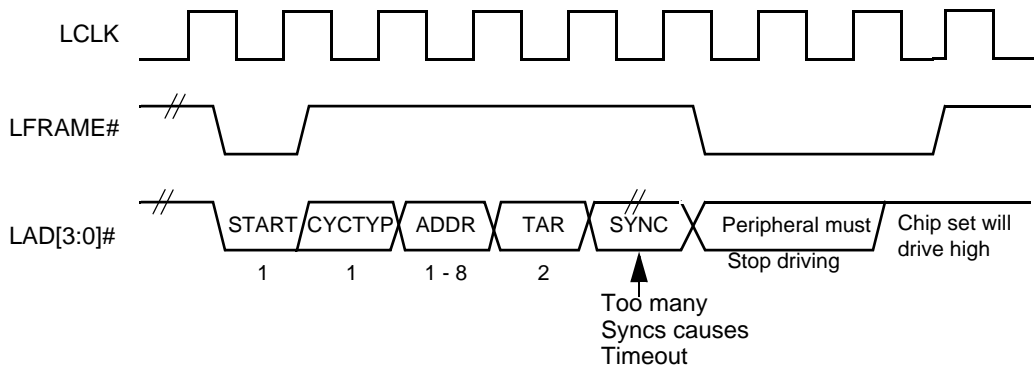
**Start of Cycle (see Figure 5-39):** The host asserts LFRAME# for one or more clocks and drives a START value on LAD[3:0], all peripherals stop driving the LAD[3:0] signals even if in the middle of a transfer. The peripheral must always use the last START value when LFRAME# was active. On the clock after the START value, the host de-asserts LFRAME#.

**Abort Mechanism (see Figure 5-40):** The host can cause an abort on the LPC interface by driving LFRAME# active with a START value of 1111b. The host must keep LFRAME# active for at least four consecutive clocks and drive LAD[3:0] to 1111b no later than the fourth clock after LFRAME# goes active. The host must drive LFRAME# inactive for at least one clock after an abort.

An abort typically occurs on SYNC timeouts.



**Figure 5-39. Start of Cycle Timing Diagram**



**Figure 5-40. Abort Mechanism Timing Diagram**

**5.13.2.1 Host Initiated Cycles**

**Memory Cycles:** Memory read or write cycles are intended for memory-mapped devices. The ADDR field is a full 32 bits, and transmitted with most significant nibble first. Typically a memory device supports much less addressing and ignores address bits above which it is capable of decoding.

**I/O Cycles:** I/O read or write cycles are intended for peripherals. These cycles are generally used for register or FIFO accesses and have minimal Sync times. Data transfers are assumed to be exactly 1 byte. The host is responsible for breaking up larger data transfers into 8-bit cycles. The minimum number of wait states between bytes is 1.

The host initiated cycles are shown in Table 5-24.

**Table 5-24. Host Initiated Cycles**

Memory or I/O	Driven By	
	Read Cycle	Write Cycle
START	Host	Host
CYCTYP + DIR	Host	Host
ADDR	Host	Host
TAR	Host	Host
SYNC	Peripheral	Peripheral
DATA	Peripheral	Host
TAR	Peripheral	Peripheral

**5.13.2.2 DMA Initiated Cycles**

DMA on LPC is handled through the LDRQ# line from peripherals and special encoding on LAD[3:0] for the host. Single, demand, verify, and increment mode are supported on the LPC interface. Block, decrement, and cascade are not supported. Channels 0 through 3 are 8-bit channels. Only 8-bit channels are supported.

**Asserting DMA Requests:** Peripherals need the DMA service to encode their request channel number on the LDRQ# signal. LDRQ# is synchronous with LCLK. Peripherals start the sequence by asserting LDRQ# low. The next 3 bits contain the encoded DMA channel number with the MSB first. And the next bit (ACT) indicates whether the

requested channel is active or not. The case where the ACT is low (inactive) will be rare, and is only used to indicate that a previous request for that channel is being abandoned. After indication, LDRQ# should go high for at least one clock. After that one clock LDRQ# can be brought low for next encoding sequence (see Figure 5-41.)

**DMA Transfer:** Arbitration for DMA channels is performed through the 8237 within the host. Once the host won the arbitration, it asserts LFRAME# on the LPC bus. The host starts a transfer by asserting 0000b on LAD[3:0] with LFRAME# asserted. The host's assert "cycle type" and direction is based on the DMA transfer. In the next cycle it asserts channel number and in the following cycle it indicates the size of the transfer.

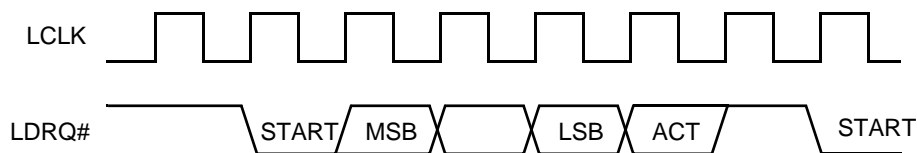
**DMA Reads:** The host drives 8 bits of data and turns the bus around, then the peripheral acknowledges the data with a valid SYNC.

**DMA Writes:** The host turns the bus around and waits for data, then the peripheral indicates data is ready through valid SYNC and transfer of the data.

The DMA initiated cycles are shown in Table 5-25.

**Table 5-25. DMA Initiated Cycles**

DMA	Driven By	
	Read Cycle (Host to Peripheral)	Write Cycle (Peripheral to Host)
START	Host	Host
CYCTYP	Host	Host
CHANNEL	Host	Host
SIZE	Host	Host
DATA	Host	Host
TAR	Host	Peripheral
SYNC	Peripheral	Peripheral
TAR	Peripheral	Peripheral



**Figure 5-41. DMA Cycle Timing Diagram**

### 5.13.3 Serial IRQ

The LPC supports a serial IRQ scheme. This allows a single signal to be used to report ISA-style interrupt requests. Because more than one device may need to share the single serial IRQ signal, an Open Collector signaling scheme is used.

Serial interrupt information is transferred using three types of frames: a Start frame, one or more IRQ Data frames, and one Stop frame (see Figure Figure 5-42, Figure 5-43, and Figure 5-44 on page 153). There are also two modes of operation. Quiet mode, initiated by the peripheral, and Continuous mode, initiated by the host:

- 1) **Quiet (Active) Mode:** To indicate an interrupt, the peripheral brings the SERIRQ signal active for one clock, and then places the signal in TRI-STATE mode. This brings all the state machines from the Idle state to the Active state.

The host then takes control of the SERIRQ signal by driving it low on the next clock, and continues driving it low for 3-7 clocks more (programmable). Thus, the total number of clocks low will be 4-8. After those clocks, the host drives SERIRQ high for one clock and then places SERIRQ into the TRI-STATE mode.

- 2) **Continuous (Idle) Mode:** In this mode, the host initiates the Start frame, rather than the peripherals. Typically, this is done to update IRQ status (acknowledges). The host drives SERIRQ low for 4-8 clocks. This is the default mode after reset; it can be used to enter the Quiet Mode.

#### Data Frame

Once the Start frame has been initiated, all of the serial interrupt peripherals must start counting frames based on the rising edge of the SERIRQ. Each of the IRQ/DATA frames has exactly three phases of one clock each: a Sample phase, a Recovery phase, and a Turn Around phase.

During the sample phase, the device drives SERIRQ low if the corresponding interrupt signals should be active. If the corresponding interrupt is inactive, then the devices should not drive the SERIRQ signal. It will remain high due to pull-up registers. During the other two phases (Turn Around and Recovery), no device should drive the SERIRQ signal. The IRQ/DATA frames have a specific order and usage as shown in Table 5-26.

#### Stop Frame

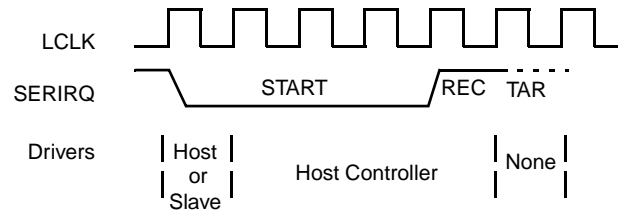
After all of the Data frames, a Stop frame is performed by the host. This is accomplished by driving SERIRQ low for two to three clocks. The number of clocks determines the next mode:

- If the SERIRQ is low for two clocks, the next mode is the Quiet mode. Any device may initiate a Start frame in the second clock (or more) after the rising edge of the Stop frame.
- If SERIRQ is low for three clocks, the next cycle is the Continuous mode. Only the host may initiate a Start frame in the second clock (or more) after the rising edge of the Stop frame.

**Table 5-26. IRQ Data Frames**

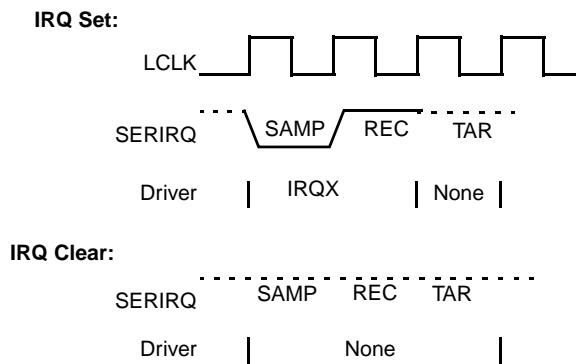
Date Frame Number	Usage
0	IRQ0
1	IRQ1
2	SMI# (Not Supported)
3	IRQ3
4	IRQ4
5	IRQ5
6	IRQ6
7	IRQ7
8	IRQ8
9	IRQ9
10	IRQ10
11	IRQ11
12	IRQ12
13	IRQ13
14	IRQ14
15	IRQ15
31-16	Unassigned





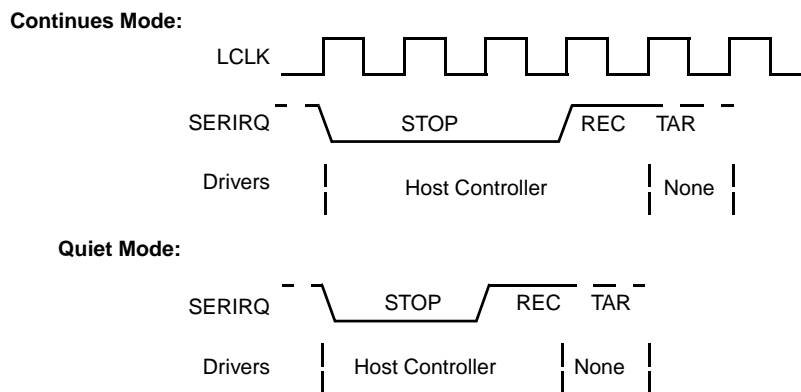
START: Start pulse width can be from 4-8 cycles, the width is determined by the value of START width.  
 REC: Recover, host actively drives SERIRQ high.  
 TAR: Turn Around Cycle. Dead cycle to prevent bus contention.

**Figure 5-42. Start Frame Waveform**



SAMP: Sample, slave drives low or leaves high.  
 REC: Recover, slave actively drives SERIRQ high if driven low during sample.  
 TAR: Turn Around Cycle. Dead cycle to prevent bus contention.

**Figure 5-43. IRQ Frame Waveform**



REC: Recover, host actively drives SERIRQ high.  
 TAR: Turn Around Cycle. Dead cycle to prevent bus contention.

**Figure 5-44. Stop Frame Waveform**

**5.13.4 Firmware Hub Interface**

The Firmware Hub (FWH) relies on the Intel Firmware Hub interface to communicate with the outside world. This interface consists of four bidirectional signals and one “control” input. The timing and the electrical parameters of the FWH interface are similar to those of the LPC interface. The Intel FWH interface is designed to use an LPC-compatible Start cycle, with a reserved cycle type code. This ensures that all LPC devices present on the shared interface will ignore cycles destined for the FWH, without becoming “confused” by the different protocols.

When the FWH interface is active, information is transferred to and from the FWH by a series of “fields” where each field contains four bits of data. Many fields are one clock cycle in length but can be of variable length, depending upon the nature of the field. Field sequences and contents are strictly defined for read and write operations.

**5.13.4.1 FWH Cycles**

A cycle is started on the rising edge of LCLK when LFRAME# is asserted and a valid cycle type is driven on LAD[3:0] by the host. Valid cycle types for the FWH are 1101b (read) and 1110b (write).

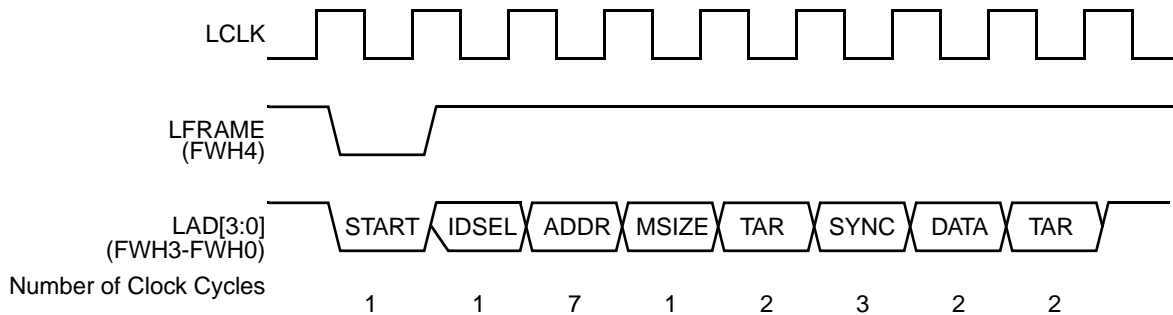
**FWH Read Cycles:** A read cycle is initiated by asserting 1101b on LAD[3:0] with LFRAME# low. All data transfers are valid on the rising edge of the LCLK. The cycle is illustrated in Figure 5-45 and described in Table 5-27.

**FWH Write Cycles:** A write cycle is initiated by asserting 1110b on LAD[3:0] with LFRAME# low. All data transfers are valid on the rising edge of the LCLK. The cycle is illustrated in Figure 5-46 and described in Table 5-28.

**Abort Operation:** LFRAME# (FWH4) active (low) indicates either that a Start cycle will eventually occur or that an abort is in progress. In either case, if LFRAME# (FWH4) is asserted, the Intel FWH will “immediately” TRI-STATE its outputs and the FWH state machine will reset.

During a write cycle, there is a possibility that an internal Flash write or erase operation is in progress (or has just been initiated). If LFRAME# (FWH4) is asserted during this frame, the internal operation will not abort. The software must send an explicit Flash command to terminate or Suspend the operation.

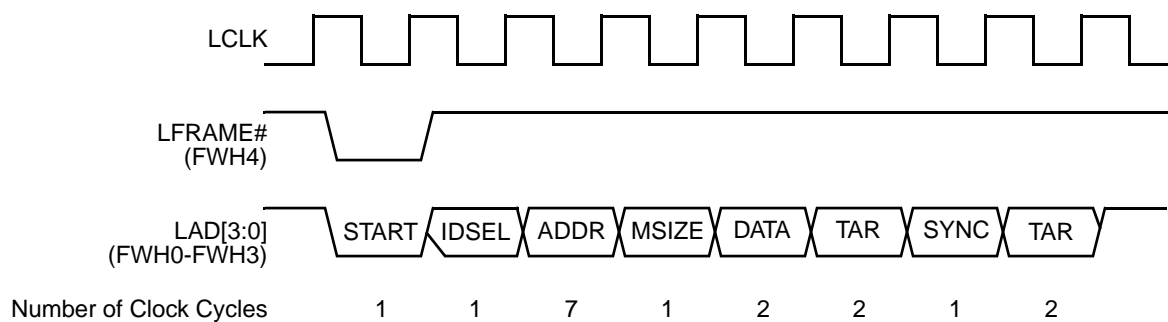
The internal FWH state machine will not initiate a Flash write or erase operation until it has received the last data nibble from the chip set. This means that LFRAME (FWH4) can be asserted as late as this cycle (“cycle 12”) and no internal Flash operation will be attempted. However, since the Intel FWH will start “processing” incoming data before it generates its SYNC field, it should be considered a non-buffered peripheral device.



**Figure 5-45. FWH Read Cycle**

**Table 5-27. FWH Read Cycle**

Signal	Clock Cycle	LAD[3:0]	Peripheral I/O	Description
START	1	1101b	I	On the rising edge of CLK with LFRAME# low, the contents of LAD[3:0] indicate the start of an FWH cycle.
IDSEL	1	0000	I	Indicates which FWH peripheral is selected. The value on the LAD[3:0] is compared to the IDSEL strapping on the FWH device pins to select which device is being addressed.  Note: From Intel <i>82802 Specification</i> - the boot device must have an ID (determined by ID strapping pins ID[3:0]) of 0. It is advisable that subsequent devices use incremental numbering.
ADDR	7	xxxx	I	A 28-bit address phase is transferred starting with the most significant nibble first.
MSIZE	1	0000b	I	Always 0000b (single byte transfer).
TAR	1	1111b	I	The LPC host drives LAD[3:0] to 1111b to indicate a turnaround cycle.
TAR	1	1111b (float)	O	The FWH device takes control of LAD[3:0] during this cycle.
WSYNC	2	0101b	O	The FWH device drives LAD[3:0] to 0101b (short wait-sync) for two clock cycles, indicating that the data is not yet available.
RSYNC	1	0000b	O	The FWH device drives LAD[3:0] to 0000b, indicating that data will be available during the next clock cycle.
DATA	2	xxxx	O	Data transfer is two cycles, starting with least significant nibble.
TAR	1	1111b	O	The FWH device drives LAD[3:0] to 1111b, to indicate a turnaround cycle.
TAR	1	1111b (float)	N/A	The FWH device floats its output and the LPC host takes control of LAD[3:0].



**Figure 5-46. FWH Write Cycle**

Table 5-28. FWH Write Cycle

Signal	Clock Cycle	LAD[3:0]	Peripheral I/O	Description
START	1	1110b	I	On the rising edge of CLK with LFRAME# Low, the contents of LAD[3:0] indicate the start of an FWH cycle.
IDSEL	1	0000	I	Indicates which FWH peripheral is selected. The value on the LAD[3:0] is compared to the IDSEL strapping on the FWH device pins to select which device is being addressed.  Note: From Intel(R) 82802 spec - the boot device must have an ID (determined by ID strapping Pins ID[3:0]) of 0. It is advisable that subsequent devices use incremental numbering.
ADDR	7	xxxx	I	A 28-bit address phase is transferred starting with the most significant nibble first.
MSIZE	1	0000b	I	Always 0000b (single byte transfer).
DATA	2	xxxx	I	Data transfer is two cycles, starting with least significant nibble.
TAR	1	1111b	I	The LPC host drives LAD[3:0] to 1111b to indicate a turnaround cycle.
TAR	1	1111b (float)	O	The FWH device takes control of LAD[3:0] during this cycle.
SYNC	1	0000b	O	The FWH device drives LAD[3:0] to 0000b to indicate it has received data or a command.
TAR	1	1111b	O	The FWH device drives LAD[3:0] to 1111b, indicating a turnaround cycle.
TAR	1	1111b (float)	N/A	The FWH device floats its output and the LPC host takes control of LAD[3:0].

### 5.14 Real-Time Clock Features

The Real-Time Clock (RTC) consists of three main blocks: the digital section, the analog section, and the level-shifter block (see Figure 5-47). The digital section contains the bus interface, RAM, voltage control, time generator, and the time keeper. The analog section contains the voltage switch and low power crystal oscillator. Finally, the level shifter block provides the appropriate voltage level translation of signals to and from the RTC block. Level shifters are needed because the RTC is powered by the  $V_{PP}$  (output of the analog section), which is different from the  $V_{CORE}$  and  $V_{CORE\_VSB}$  power domains.

**Features**

- Accurate timekeeping and calendar management
- Alarm at a predetermined time and/or date
- Three programmable interrupt sources

- Valid timekeeping during power-down, by utilizing external battery backup
- 242 bytes of battery-backed RAM
- RAM lock schemes to protect its content
- Internal oscillator circuit (the crystal itself is off-chip), or external clock supply for the 32.768 kHz clock
- A century counter
- Additional low-power features such as:
  - Automatic switching from battery to  $V_{SB}$
  - Internal power monitoring on the VRT bit
  - Oscillator disabling to save battery during storage
- Software compatible with the DS1287 and MC146818

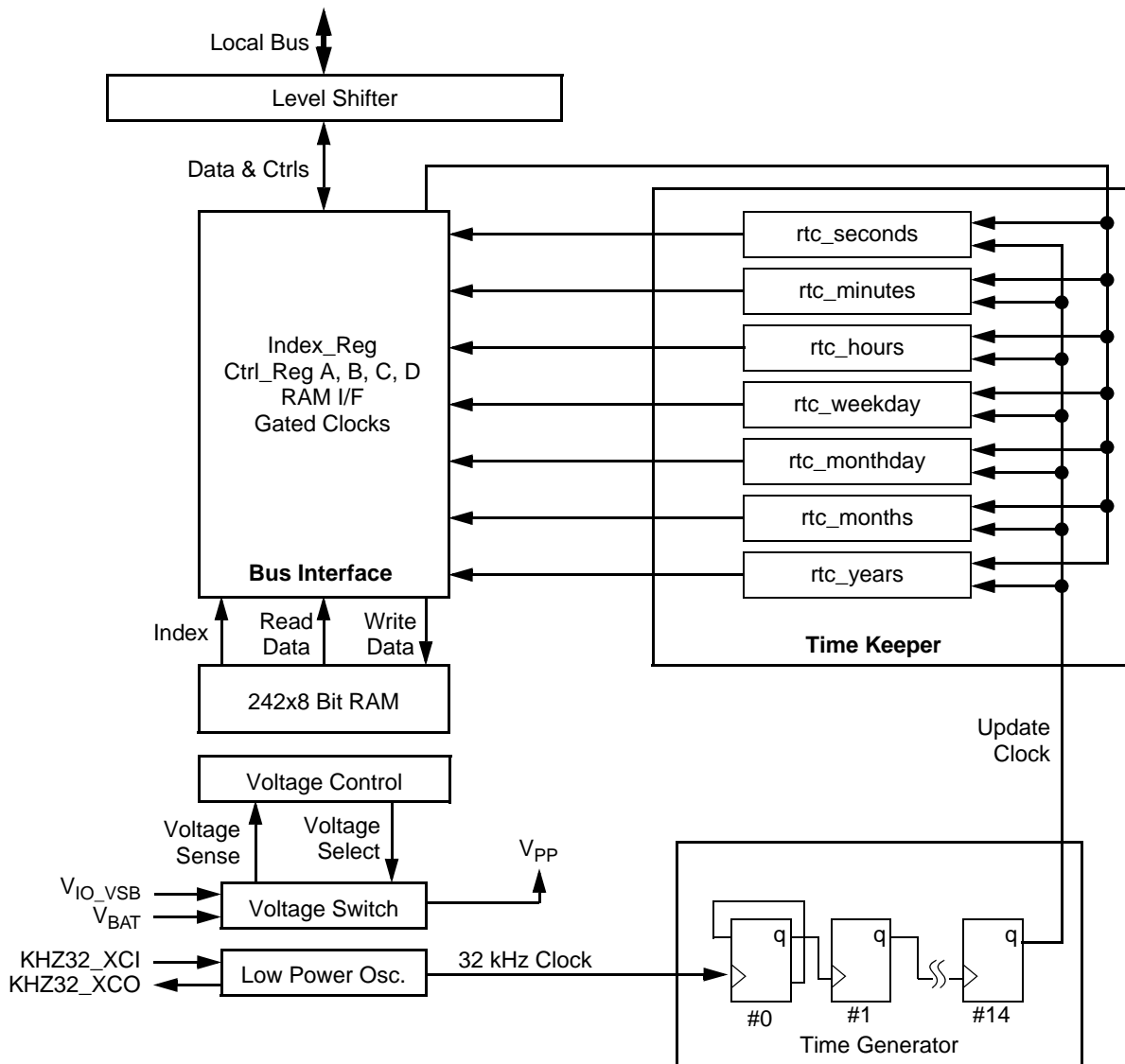


Figure 5-47. RTC Block Diagram

### 5.14.1 External Use Recommendations

It is recommended that the external components for the oscillator be connected as illustrated in Figure 5-48. The recommended specifications for those external components are listed in Table 5-29.

Capacitors C1 and C2 should be chosen to match the crystal's load capacitance. The load capacitance  $C_L$  "seen" by the crystal Y is comprised of C1 in series with C2 in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout, and socket (if any). The rule of thumb in choosing these capacitors is:

$$C_L = (C1 * C2)/(C1 + C2) + C_{PARASITIC}$$

C1 can be trimmed to achieve precisely 32.768 kHz. To achieve high time accuracy, use crystal and capacitors with low tolerance and temperature coefficients.

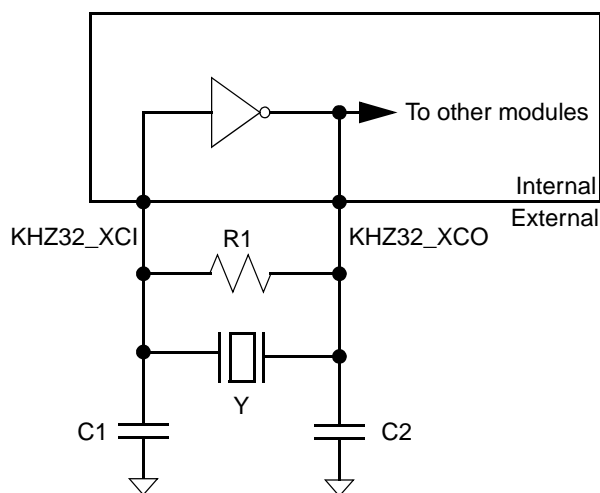


Figure 5-48. Recommended External Component Connections

Table 5-29. External Component Recommended Specifications

Component	Parameters	Values	Tolerance
Crystal	Resonance	32.768 kHz Parallel Mode	User-defined
	Type	N-Cut or XY-bar	
	Serial Resistance	40 k $\Omega$	Max
	Q Factor	35000	Min
	Shunt Capacitance	2 pF	Max
	Load Capacitance, $C_L$	9-13 pF	
	Temperature Coefficient	User-defined	
Resistor, R1	Resistor	20 M $\Omega$ (Note 1)	5%
Capacitor, C1	Capacitor	22 pF	5%
Capacitor, C2	Capacitor	22 pF	5%

Note 1. A single 20 M $\Omega$  resistor may be difficult to acquire. Substituting two 10 M $\Omega$  resistors in series is acceptable.

## 5.15 General Purpose Input/Output

Proper use and understanding of the General Purpose Input/Output (GPIO) subsystem is the key to applying the Geode CS5535 companion device in a custom system design. By totalizing the optional features of the Geode CS5535 companion device GPIOs, system functions such as soft buttons, DDC monitoring, timers, system interrupts, and others, may be implemented. The system designer should pay careful attention to the suite of features available through the GPIO subsystem and, because the GPIOs are multiplexed with other on-chip functions, must make careful trade-offs to obtain the features desired in the system.

The register space for control of the GPIO subsystem contains space for control of 32 GPIOs. Since only 28 GPIOs are realized, the control bits for the non-existent GPIO[31:29], and GPIO[23] are marked "Reserved". GPIO[22:16] are multiplexed with the LPC bus; therefore, if the system requires an LPC bus, GPIO[22:16] are not available as GPIOs. Likewise, GPIO[15:14] are multiplexed with the SMB (System Management Bus); if the system requires the SMB, GPIO[15:14] will be dedicated to this function and not available as GPIOs. Other GPIOs are multiplexed with individual functions as indicated in Table 3-8 "GPIO Options" on page 47.

### Features

- Input Features:
  - Each of the available GPIOs may be configured as an input. A block of eight Input Conditioning Functions, providing edge detection, event counting, and input filtering, may be configured for use by any eight of the 28 GPIOs, though all 28 may have edge detection. The optionally-conditioned input may then be fed to steering logic that can connect it to an interrupt, or power-management input event (PME).
- Output Features:
  - Each of the available 28 GPIOs has a configurable output cell. The output cell for each GPIO may be independently configured to provide a variety of interface options. The cell may be enabled or disabled, configured as a totem-pole or open-drain type, have internal pull-up or pull-down resistors applied, or be inverted.
  - As indicated in Table 3-8 "GPIO Options" on page 47, the GPIOs have differing output driver types and reset defaults. When choosing a GPIO for a given function, choose one with a compatible output driver type, and one that the use of, does not make another desired function inaccessible. Careful study of this table will assist the system designer in making proper selections of the desired functionality of the suite of GPIOs.
- Auxiliary Functions:
  - Most of the 28 GPIOs have additional hard-wired internally-connected functions that may be selected by choosing either the AUX\_1 or AUX\_2 outputs. Use of these allows internal functions to be accessed at the device pins. Table 3-8 "GPIO Options" on page 47 identifies these auxiliary functions, including access to the UARTS and multi-function timers, as well as certain power-management controls.
- Output Mapping:
  - After passing through the optional input conditioning circuits, any GPIO may be mapped (connected) to one of eight PIC-level interrupts, or to one of eight Power Management Event (PME) inputs. A given GPIO may not be simultaneously mapped to both an interrupt and a PME. The PIC subsystem interrupt inputs may be configured to cause the generation of an ASMI-type interrupt from any or all of the mapped GPIO signals.
- Power Domains:
  - The GPIO circuits are distributed into the Working and Standby power domains. Those circuits in the Standby power domain may be used for system wakeup events, since they remain powered when the Working power is removed. As indicated in Table 3-8 "GPIO Options" on page 47, GPIO[28:24] are located in the Standby power domain; all others are in the Working power domain. Event/Filter pairs 6 and 7 are located in the Standby domain; pairs [5:0] are in the Working power domain.
- Auto-sense:
  - GPIO5 and GPIO6 have a feature called Auto-sense. When reset is applied to the system, a weak internal pull-up is applied to the pad. When reset is de-asserted, the auto-sense value is used to establish the pull-up/down state on the de-assertion edge. If nothing pulls down the pad, then the weak pull-up continues to be applied. If the pad is pulled down, then pull-up is set to "no" and pull-down is set to "yes". The output driver does not actively drive the pad, that is, it remains in TRI-STATE mode. If an auto-sensed pull-down is desired, a diode between the reset signal and the GPIO pin will pull it down during the auto-sense operation but will have no effect during normal operation.
- Recommended Functions:
  - System designers at AMD have created a list of recommended uses for selected GPIOs, see Table 3-8 "GPIO Options" on page 47. The desired functions were matched up with GPIOs by selecting appropriate buffer types and multiplexing options to create an optimal list of recommended uses for the GPIOs. Designers may use these recommended functions as a starting point and make modifications to the list as needed to fit the particulars of their system.

### 5.15.1 Programming for Recommended Functions

Table 3-8 "GPIO Options" on page 47 includes an "Recommended Use" column. Shown below are the register settings to achieve the example.

Example Use	Getting Example Use	Note
PCI_INTA#	INPUT_ENABLE = 1 Setup GPIO Interrupt Mapper	
AC_BEEP	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
IDE_IRQ0	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	
DDC_SCL	OUT_ENABLE = 1	Software write OUT_VALUE
DDC_SDA	OUT_ENABLE = 1	Software write OUT_VALUE
MFGPT0	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
MFGPT1	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
PCI_INTB#	INPUT_ENABLE = 1 Setup GPIO Interrupt Mapper	
UART1_TX	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
UART1_RX	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	
THRM_ALRM#	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	
	INPUT_INVERT = 1	
SLP_CLK#	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
GPIO IN	INPUT_ENABLE = 1	Software read READ_BACK
GPIO IN	INPUT_ENABLE = 1	Software read READ_BACK
SMB_CLK	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	
	OUT_AUX1_SELECT = 1	
SMB_DATA	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	
	OUT_AUX1_SELECT = 1	
LPC_AD0 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_AD1 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_AD2 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_AD3 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_DRQ# 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_SERIRQ 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
LPC_FRAME# 34	Hardware default	Table 3-6 "DIVIL BALL_OPT (MSR 51400015h)" on page 34
WORK_AUX	OUT_ENABLE = 1	
	OUT_AUX1_SELECT = 1	
LOW_BAT#	INPUT_ENABLE = 1	
	IN_AUX1_SELECT = 1	



### 5.15.2 Register Strategy

The register set for the GPIO subsystem has been arranged in such a way as to eliminate the need for read-modify-write operations. Individual GPIO control bits may be directly and immediately altered without requiring knowledge of any other GPIO states or bit settings. Previous systems required the current settings for all GPIOs to be read, selected pins changed, and the result written back. If this read-modify-write operation was interrupted by another process that also used the GPIOs, then erroneous operation could result.

To avoid the read-modify-write operation, two data bits are used to control each GPIO feature bit, wherein a feature is enabled or disabled. One register bit is used to establish a logic 1, while a second register bit is used to establish a logic 0. A 1 in a register bit changes the feature bit's value, while a 0 does nothing. Since there are two register bits for each feature bit, for each GPIO, there are four combinations of register bits possible. The two control bits operate in an exclusive-OR pattern, as illustrated in Table 5-30.

**Table 5-30. Effect on Feature Bit**

Logic 0 Bit Position	Logic 1 Bit Position	Effect on Feature Bit
0	0	No change
1	0	Feature bit is cleared to 0
0	1	Feature bit is set to 1
1	1	No change

An example 16-bit register controlling a feature bit for eight GPIOs is illustrated in Table 5-31. Note that the real registers are 32 bits; 16 bits are used here as an illustrative example.

Assume that the register in Table 5-31 allows setting and clearing of an unspecified "feature bit" for GPIO[7:0]. Assume that the 16-bit value given in the example has just been written into the register. In this example, all four possible bit combinations from Table 5-30 are examined.

**Table 5-31. 16-Bit GPIO Control Register Example**

Bit No.	"1" Sets Control Bit to 0								"1" Sets Control Bit to 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0
GPIO #	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

GPIO5 has a 0 in the logic 0 bit position (register bit 13), and a 1 in the logic 1 bit position (register bit 5), so the GPIO5 feature bit would become a 1.

GPIO4 has a 1 in the logic 0 bit position (register bit 13), and a 0 in the logic 1 bit position (register bit 5), so the GPIO4 feature bit would become a 0.

GPIO7 has a 1 in both bit positions 15 and 7. Writing a 1 to both the logic 1 and logic 0 bit positions causes no change to the GPIO7 feature bit.

GPIO6 has a 0 in both bit positions 14 and 6. Writing a 0 to both the logic 1 and logic 0 bit positions causes no change to the GPIO6 control bit. GPIO[3:0], also have 0s in both bit positions, so they experience no change.

Reads produce a normal and an inverted value. For example, assume the Output Enable is set only for GPIO4 in the above register. A read would return the value EF10h.

Actual GPIO registers associated with feature bit settings are 32 bits wide and each handle 16 GPIOs. They are organized into low and high banks. The low bank deals with GPIO[15:0], while the high bank deals with GPIO[28:24] and GPIO[22:16].

In addition to these "bit registers", there are value registers for the Input Conditioning Functions.

### 5.15.3 Lock Bits

Many GPIO registers are protected against accidental changes by Lock Enable registers that prevent further changes. Once a LOCK bit is set, the associated register can not be changed until the corresponding LOCK bit is cleared. There are two Lock Bit registers, one for the high bank (GPIOH\_LOCK\_EN, GPIO I/O Offset 8Ch) and one for the low bank (GPIOH\_LOCK\_EN, GPIO I/O Offset 3Ch). All GPIO registers are protected by LOCK bits except the High and Low Bank Read Back registers, (GPIO[x]\_READ\_BACK), High and Low Bank Positive Edge Status registers (GPIO[x]POSEDGE\_STS), the High Low Bank Negative Edge Status registers (GPIO[x]NEGEDGE\_STS), and of course the Lock Enable registers themselves.

### 5.15.4 GPIO Basic I/O Configuration

The General Purpose Input and Output (GPIO) Interface is illustrated in Figure 5-49. The figure represents one of twenty-eight GPIOs potentially available. Note the GPIOs [31:29] and [23] are non-existent. Table 3-8 "GPIO Options" on page 47 provides a complete list of features for each GPIO and should be consulted when configuring a system.

Each GPIO has basic configuration options used to set up the characteristics of the GPIO for either input or output. Each of the functions in the list below follows the GPIO register strategy outlined in Section 5.15.2 "Register Strategy" on page 161 unless otherwise noted. This strategy allows individual GPIOs to be modified without accidentally changing the characteristics of unrelated GPIOs, and without requiring 'read-modify-write' cycles. All values are active high.

- **OUT\_EN.** When high, enables this GPIO/OUT\_AUX for output. A pad may be configured for output, input, or both.
- **IN\_EN.** Enables this GPIO/IN\_AUX for input. A pad may be configured for input, output, or both.
- **OUT\_VAL.** This will establish the value driven to the pad when it is selected as an Output, unless either OUT\_AUX1 or OUT\_AUX2 are selected. The value driven to the GPIO pad is subject to an optional inversion.

- **OUT\_INVRT\_EN.** When high, inverts the GPIO/OUT\_AUX Output Value.
- **IN\_INVRT\_EN.** Inverts the signal applied to the ball, and presents the inverted value to all follow-up circuitry (i.e., input conditioning functions). Affects both GPIO and IN\_AUX.
- **OUT\_OD\_EN.** Configures this GPIO/OUT\_AUX for open-drain operation. When the output pad is to be driven low, the pad is driven low. When the output pad is to be driven high, the pad is allowed to float and is not driven.
- **OUT\_AUX1\_SEL** and **OUT\_AUX2\_SEL.** Selects an internal auxiliary source for the Output Value. Table 3-8 "GPIO Options" on page 47 identifies all the possible internal connections for these two auxiliary sources.
- **IN\_AUX1\_SEL.** Selects an internal destination other than the GPIO for the ball. Table 3-8 "GPIO Options" on page 47 lists all the functions that may be connected in this manner.
- **PU\_EN.** Applies a weak pull-up to the pad. The effect of this control is independent of all other settings except PD\_EN. If PU\_EN is set by software, PD\_EN is automatically cleared. Affects both GPIO and IN\_AUX.
- **PD\_EN.** Applies a weak pull-down to the pad. The effect of this control is independent of all other settings except PU\_EN. If PD\_EN is set by software, PU\_EN is automatically cleared. Affects both GPIO and IN\_AUX.

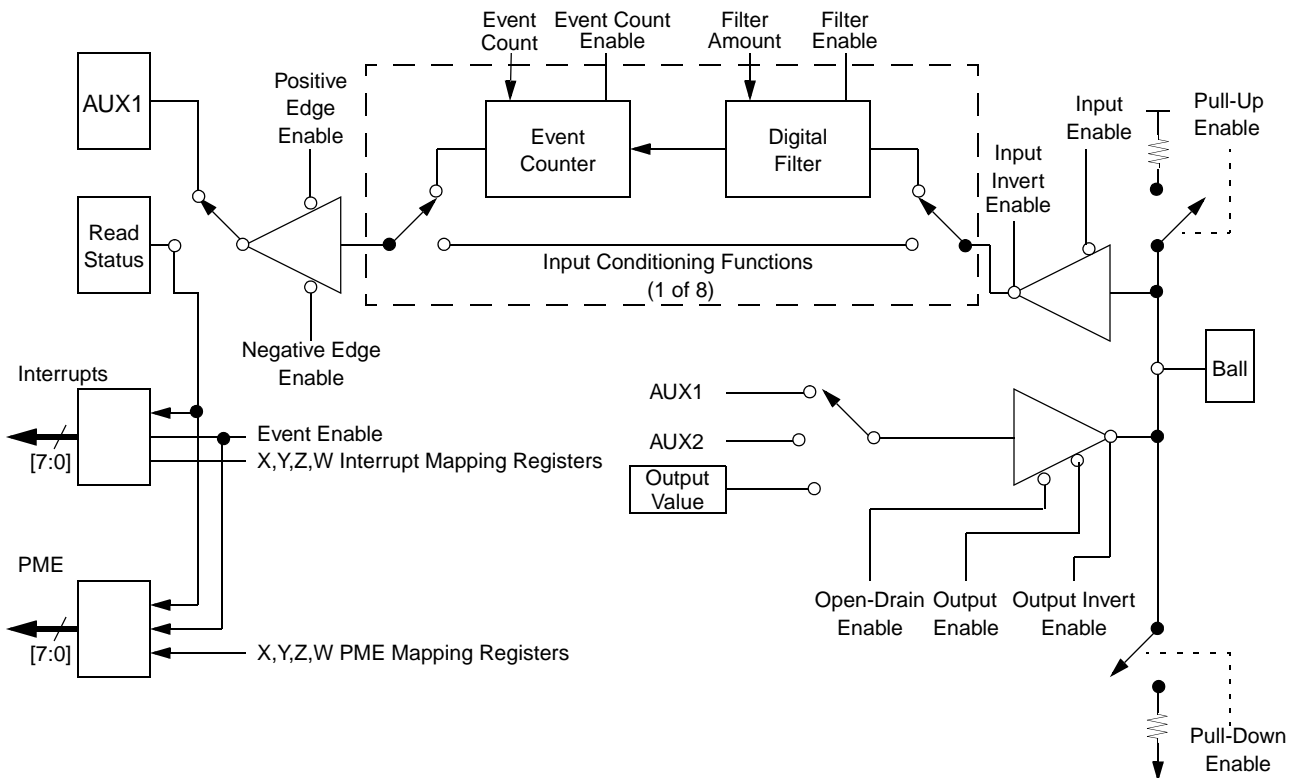


Figure 5-49. GPIO Configuration

### 5.15.5 Input Conditioning Functions

GPIOs and their corresponding IN\_AUX function in the Geode CS5535 companion device may have their inputs conditioned by configurable circuitry as illustrated in Figure 5-49 on page 162. Any GPIO may be connected to one of eight Input Conditioning functions, each consisting of a Digital Filter and an Event Counter (known as an Event/Filter pair). Each GPIO is followed by an edge detection function that may be set for either positive or negative going edges. As shown in Figure 5-49, the edge detection function may be used to monitor the output of the Event/Filter pair that has been associated with that particular GPIO, or it may be used independently of the Event /Filter pair.

These functions are enabled as follows:

- IN\_FLTR\_EN. Enables the input filter function of the associated GPIO.
- EVNTCNT\_EN. Enables the event counter function of the associated GPIO.
- IN\_POSEDGE\_EN and IN\_NEGEDGE\_EN. Enables the edge detection function and mode.

The final input value may be read back by a software accessible register (GPIO[x]\_READ\_BACK). It may also be used as an Interrupt or a Power Management Event.

There are a total of eight Digital Filter/Event Counter pairs that are shared by 28 GPIOs. There is a selection function to associate a given Filter/Counter pair with a given GPIO. All GPIOs incorporate edge detection.

#### 5.15.5.1 Input Filter Conditioning Function

The digital filter is one-half of a Filter/Event conditioning circuit. (The other half is the Event Counter.) The filter is used to produce a stable output from an unstable input. Mechanical switch de-bounce is a typical use.

To use one of the eight digital filters, it must first be assigned to one of the GPIO inputs using one of the GPIO\_FE[x]\_SEL registers (GPIO I/O Offsets F0h-F7h); where “x” is the number of the Filter/Event pair, 0 to 7. Then the filter function must be enabled through either the GPIOL\_IN\_FLTR\_EN (GPIO I/O Offset 28h) or the GPIOH\_IN\_FLTR\_EN (GPIO I/O Offset A8h) registers, depending on whether the selected GPIO is in the high [28:16] or low [15:0] bank. Finally, a GPIO\_FLTR[x]\_AMNT (GPIO I/O Offsets 50h, 58h, 60h, 68h, 70h, 78h, D0h, and D8h) must be determined and then programmed to establish the filter’s stability period.

The associated GPIO input must ultimately remain stable for a FLTR\_AMNT number of 32 kHz clock edges in order for the output to change. A FLTR\_AMNT of 0 effectively disables the filtering function, because the counter will not roll over from 0 to all 1s. The maximum FLTR\_AMNT is FFFFh.

The digital filter is based upon a 16-bit programmable down-counter. An initial count is loaded into the counter via the GPIO\_FLTR[x]\_AMNT register. When the associated GPIO input changes, the counter begins counting down from FLTR\_AMNT towards 0. If the associated GPIO input remains stable for the length of the count-down period, then the counter reaches 0 and produces an output pulse to whatever the GPIO is internally connected to. If the associated GPIO input changes during the count-down period, then the counter reloads the initial count from the GPIO\_FLTR[x]\_AMNT register and begins counting down towards 0 again.

Direct access to the counter’s state is provided by the R/W register GPIO\_FLTR[x]\_CNT, that may be read at any time to determine the current value of the counter. The GPIO\_FLTR[x]\_CNT register may also be written to at any time, thereby jamming the counter state forward or backward from the current count.

Reads and writes of the GPIO\_FLTR[x]\_CNT register are internally synchronized to avoid false read values and corrupted writes, that is, reads and writes may occur to a filter circuit without concern of the phasing or timing of the 32 kHz clock edges. When GPIO[x]\_IN\_FLTR\_EN is low the filter circuit is not clocked.

The filter circuit is used to produce a stable output from an unstable input. Mechanical switch de-bounce is a typical use. The default value for all flip-flops, the Down Counter, and the Filter Amount Register is zero. Software establishes the filter amount. As long as the preliminary input on the left matches the filtered input on the right, the circuit is stable and the counter continuously loads the filter amount value. When the preliminary input changes, the counter begins to count. If the input remains steady, then the counter reaches zero and enables loading the value flip-flop. This brings the circuit back to the stable point. If the input does not remain steady, then the counter reloads. The preliminary input on the left must remain steady the “filter amount” number of clock edges for the final input on the right to change. A filter amount of zero effectively disables the filtering function because the Down Counter will not roll over backwards to all ones. The maximum filter amount is FFFFh

### 5.15.5.2 Input Event Counter Conditioning Function

The event counter is one half of a filter/event conditioning circuit, and is in series with its associated filter. (The other half is the digital filter.) It counts events and can produce an output when a predefined count is reached. The event counter may be down-counted by writing to a particular address. It may be used as a rate counter that may be periodically read, and that produces no output at all.

To use one of the eight event counters, it must first be assigned to one of the GPIO inputs using one of the GPIO\_FE[x]\_SEL registers (where X is the number of the Filter/Event pair, 0 to 7). Then the associated digital filter must be enabled, through either the GPIOL\_IN\_FLTR\_EN or GPIOH\_IN\_FLTR\_EN registers, depending on whether the selected GPIO is in the high [28:16] or low [15:0] bank. If digital filtering is not required, program the associated GPIO\_FLTR[x]\_AMNT registers to 0000h. Finally, the desired “compare value” (GPIO\_EVNTCNT[x]\_COMPARE) must be determined and then programmed to establish the number of events that will produce an output when that count has been reached.

The event counter is based upon a 16-bit programmable up/down counter. The up-down counter counts positive edges of the selected GPIO input and produces a constant or level output when the GPIO\_EVNTCNT[x] (counter value) exceeds the GPIO\_EVNTCNT[x]\_COMPARE (compare value). The output can be read as the GPIO and/or used to drive an auxiliary input.

The counter may be counted down one count by writing to one of two addresses, depending on which bank (High or Low) the associated GPIO resides in. Knowledge of which GPIO is associated with the event counter is required, since these two decremter registers have a dedicated bit for each GPIO. When counted down, this counter, unlike the counter in the digital filter, will roll over from 0000h to FFFFh. Typically, decrementing is used to clear an interrupt or power management event as part of the associated service routine.

### 5.15.5.3 Uses of the Event Counter

Such an auxiliary input could be used to drive an ASMI or maskable interrupt. Assume the compare value is set to 0. The service routine clears the ASMI by decrementing the counter via the mechanism illustrated. If additional events have occurred, the count does not decrement to 0 and the ASMI remains asserted. The count up and down inputs are synchronized such that false values are not created if up and down pulses occur at or near the same instant in time. The counter will not decrement through 0.

Alternatively, the compare value could be set to a higher value to trigger an ASMI or interrupt when a certain number of events has occurred. In this case, the ASMI or interrupt is cleared by writing the counter to 0.

Lastly, the input value may be ignored and the event counter used as a rate indicator. If software reads the counter at a fixed periodic interval, an input pulse rate may be measured. Such an approach may be used to imple-

ment a tachometer function. The counter will increment past all Fs back to 0.

As suggested above, the counter may be read or written under software control. The read and write operations are synchronized such that false values are not created if count up pulses occur at or near the same instant in time.

### 5.15.5.4 Input Edge Conditioning Function

The Edge Detection function is illustrated as part of Figure 5-49 on page 162. It is normally used to generate an ASMI or maskable interrupt on each positive and/or negative edge of an input signal. Use of this function simultaneously with the event counter function is somewhat logically mutually exclusive, but is not prevented in hardware.

Each GPIO has the optional edge detection function.

The reset default for the detection circuit establishes a 0 level on GPIO[x]\_POSEDGE\_EN and GPIO[x]\_NEGEDGE\_EN. When both are set to 0, the edge detection function is disabled. If either a positive or negative edge detection is enabled, an active high output is produced when the appropriate edge occurs. This level must be cleared by writing to either the GPIO[x]\_POSEDGE\_STS or the GPIO[x]\_NEGEDGE\_STS registers, whichever is appropriate. If another edge occurs before clearing, the active high output is not affected. If the clear action occurs at the “same time” as another edge, the result is not defined.

Each edge detection function is controlled by four registers as follows:

- Positive Edge Enable (GPIO[x]\_POSEDGE\_EN). Enabled if feature bit is high.
- Negative Edge Enable (GPIO[x]\_NEGEDGE\_EN). Enabled if feature bit is high.
- Positive Edge Status (GPIO[x]\_POSEDGE\_STS). Set indicates edge. Write 1 to clear.
- Negative Edge Status (GPIO[x]\_NEGEDGE\_STS). Set indicates edge. Write 1 to clear.

### 5.15.5.5 Output Steering (Mapping)

Outputs from the internal GPIO circuits, driven by inputs to the Geode CS5535 companion device from the system, may be steered (or 'mapped') to either interrupts, or power management events (PME). Sufficient steering logic exists in the Geode CS5535 companion device to provide for eight independent interrupts and simultaneously for eight independent PMEs.

The eight GPIO interrupts are all in Working power domain; of the eight PMEs, [7:6] are in Standby power domain and [5:0] are in Working domain. Those in the Standby power domain are intended to be used to awaken the system when the Working power domain is off, however, they may also be used when the Working power domain is on. The interrupts are connected to the PIC subsystem, and the PMEs are connected to the Power Management subsystem.

Four 32-bit steering registers control the routing of the GPIOs' internal output (that produced by an input to the chip from an external source, or from one of the internally-connected AUX inputs) to either an interrupt or PME. The set of four registers taken together, contain a nibble for each GPIO. The upper bit of each nibble selects either a PME (if high) or an interrupt (if low). The remaining three bits of each nibble select which of the eight possible interrupts or PMEs the GPIO will be steered to.

The four registers are identified as GPIO Mapper X, Y, Z, and W. Their GPIO associations are as follows:

- GPIO\_MAP\_X = GPIO[7:0]
- GPIO\_MAP\_Y = GPIO[15:8]
- GPIO\_MAP\_Z = GPIO[23:16]
- GPIO\_MAP\_W = GPIO[31:24]

The steering logic does not prohibit mapping of two or more GPIOs to the same output, but it is impossible to create a single GPIO that functions simultaneously as both an interrupt and a PME. Registers X, Y, Z, and W default to all 0s, as do both the High and Low EVNT\_EN registers. Thus, all GPIOs are mapped to INT[0] after a reset, but none are enabled.

### 5.15.5.6 Auto-sense

Two GPIOs (GPIO5 and GPIO6) have a function called "Auto-sense". Auto-sense is a method of automatically determining whether or not to apply a pull-up or pull-down to the corresponding GPIO input.

Auto-sensed inputs behave as follows: when reset is applied to the system, a weak pull-up is applied to the pad. When reset is de-asserted, the sensed value is used to establish the pull-up/down state on the de-assertion edge. If nothing pulls down the pad, then the pull-up continues to be applied. If the pad is pulled down, then the pull-up is cleared to 0 and the pull-down is set to 1. If a pull-down is desired, a diode between the reset signal and the GPIO pin will pull it down during the auto-sense operation but have no effect during normal operation.

Disabling the auto-sensed pull-up or pull-down requires more program operation than just disabling the pull-up or pull-down through the GPIOL\_PU\_EN or GPIOL\_PD\_EN. If the pullup was enabled through auto-sense, the procedure to disable is:

- 1) Disable the pull-up in GPIOL\_PU\_EN.
- 2) Enable the pull-down in GPIOL\_PD\_EN.
- 3) Disable the pull-down in GPIOL\_PD\_EN.

If the pull-down was enabled through auto-sense, the procedure to disable is:

- 1) Disable the pull-down in GPIOL\_PD\_EN.
- 2) Enable the pull-up in GPIOL\_PU\_EN.
- 3) Disable the pull-up in GPIOL\_PU\_EN.

## 5.16 Multi-Function General Purpose Timer

The Multi-Function General Purpose Timer module contains eight multi-function general purpose timers (MFGPTs). Six of the eight MFGPTs are in the Working power domain running off a 32 kHz clock or a 14.318 MHz clock, while the other two are in the Standby power domain running off a 32 kHz clock.

The Working power domain contains the following blocks:

- Six MFGPTs each split into three blocks, one containing I/O registers, one containing the clock switch, and one containing the timer logic.
- 15-bit prescaler to divide down the 14.318 MHz clock and generate 15 carry-out signals.
- 15-bit prescaler to divide down the 32 kHz clock and generate 15 carry-out signals.
- Logic to implement Local Bus Interface, Control Logic, MSR Registers, and NMI, IRQ, and Reset Output Events.
- Two blocks containing I/O registers to write into the two MFGPTs in the Standby power domain.

The Standby power domain contains the following blocks:

- Two MFGPTs.
- 15-bit prescaler to divide down the 32 kHz clock and generate 15 carry-out signals.
- Interface for signals going between Standby and Working power domains.

Figure 5-50 shows the top level block diagram of the Multi-Function General Purpose Timer module.

### Features

Each MFGPT operates independently and can have the following features:

- 32 kHz or 14.318 MHz clock selectable by software (MFGPT0 to MFGPT5 only; MFGPT6 and MFGPT7 use 32 kHz clock).
- Programmable input clock prescaler divisor to divide input clock by  $2^i$ , where  $i = 0$  to 15.
- Watchdog timer (trigger GPIO output, interrupt, or reset).
- Pulse Width Modulation (PWM).
- Pulse Density Modulation (PDM).
- Blink (low frequency pulse for LED).
- General Purpose Timer.
- Generate GPIO outputs.
- Provide outputs for generating reset (limited to MFGPT0 to MFGPT5), IRQs, NMI, and ASMI (indirectly through PIC).

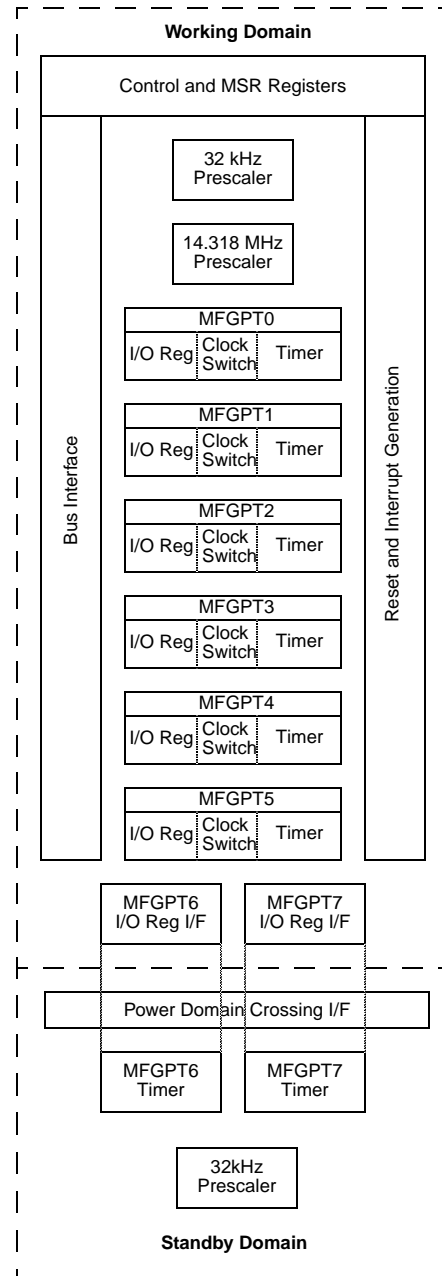


Figure 5-50. MFGPT Top Level Block Diagram

### 5.16.1 Prescaler

The 15-bit prescaler is a binary down counter, dividing down the incoming clock, and provides 15 outputs for the MFGPTs. The frequency of these outputs ranges from  $2^{-1}$  to  $2^{-15}$  of the input frequency and each pulse is one incoming clock high, so these outputs function as increment enables for the MFGPTs. The prescaler resets to  $0000_{16}$  and starts decrementing after reset. The prescaler output vector, `psclr_out[14:0]`, is based on prescaler counter `psclr_cnt[14:0]`, where `psclr_out[i] = &(~psclr_cnt[i:0])` (i.e., prescaler output bit *i* is asserted if the prescaler counter from bit *i* down to bit 0 are all low). When the prescaler reaches  $0000_{16}$ , all prescaler outputs are asserted at that time.

The external clock for the prescaler is activated if there is one or more MFGPTs activated using it as its clock source; it is also activated for MFGPT I/O register writes and synchronous counter reads (only for 14.318 MHz) when the MFGPT being written has already selected the 14.318 MHz clock as its clock source. Whenever the external clock is activated, the prescaler counts. Therefore, multiple MFGPTs and register access can affect the prescaler counting. From the point of view of the MFGPT, once the MFGPT is disabled and then re-enabled, it cannot be determined exactly when the prescaler carry-out occurs as it does not know how long the prescaler has been stopped, if at all.

### 5.16.2 I/O Registers Block

The I/O register write data is first stored in I/O register sub-modules before being transferred over to the MFGPTs. There are two types of I/O register sub-modules, one for the Working power domain and one for the Standby power domain. The main difference is that for the Working power domain, except for the counter register, the register values here and the register values in the timer are the same. For the Standby power domain, the register values in the I/O register sub-module cannot be relied upon except during write, as this logic could have been powered down in Standby mode and the register data is therefore invalid. For the Standby power domain, the read always comes from the timer directly.

#### 5.16.2.1 MFGPT Register Set

There are four software accessible I/O registers per MFGPT: Up Counter, Comparator 1 Value, Comparator 2 Value, and Setup registers. (See Section 6.17 "Multi-Function General Purpose Timer Register Descriptions" on page 483 for register details.) Writes to these registers are first stored here and then transferred to a separate copy of the register in the timer. For MFGPT0 to MFGPT5, read of these registers, except for the counter, comes from the registers here, while read of the counter register comes from the timer. For MFGPT6 and MFGPT7, reads of these registers comes from the copy inside the timer. (TW Note: Restate reason or be more precise in reference.)

#### 5.16.2.2 Setup Register

The Setup Register contains the following control fields that control the MFGPT operation:

- **Counter Enable.** Enables the Up Counter to count (it does not enable/disable other MFGPT functions).
- **Clock Select.** Instructs the clock switch logic to use the 32 kHz clock as the MFGPT clock if low or the 14.318 MHz clock if high, once this register has been written (only for MFGPT0 to MFGPT5).
- **Scale Factor.** Selects the prescaler divide scale factor for Up Counter to increment.
- **Stop Enable.** Enables the Up Counter to stop counting during a system power management Sleep mode (for MFGPT0 to MFGPT5) or Standby mode (for MFGPT6 and MFGPT7).
- **External Enable.** Enables the Up Counter to be cleared and restarted rather than performing the next increment each time there is a low to high transition detected on the GPIO input associated with the timer. An asynchronous edge-detector catches the transition; the signal is then synchronized and sent to clear the counter synchronously. Therefore, the clear does not occur immediately on the transition.
- **Reverse Enable.** Flips the order of the Up Counter outputs going to the Compare 1 circuit so that bit 0 becomes bit 15, bit 1 becomes bit 14, etc. This allows the timer logic to generate a PDM signal instead of a PWM signal. To properly generate a PDM signal, the Compare 2 Value should be set to  $FFFF_{16}$  to allow the Compare 1 Value to establish the density.
- **Compare 1 Mode.** Controls the Compare 1 output. There are four cases:
  - 00: Disabled. Output is low.
  - 01: Compare on Equal. The compare output goes high when the Up Counter value, after going through Bit Reverse logic, is the same value as the Compare 1 Value.
  - 10: Compare on GE. The compare output goes high when the Up Counter value, after going through Bit Reverse logic, is greater than or equal to the Compare 1 Value.
  - 11: Event. Same as "Compare on GE", but an event is also created. This event can be read and cleared via the MFGPT Setup Register and is used to generate interrupt and reset.
- **Compare 2 Mode.** Same as Compare 1 Mode, except this controls the Compare 2 output. The Up Counter is directly compared against the Compare 2 Value (i.e., without going through Bit Reverse logic).

All of the above fields, except Count Enable, are write-once only.

### Compare Status/Event Bits

The Setup register also contains two status bits: one from Compare 1 and one from Compare 2. If Event mode is selected, then these two status bits represent the events from the two Compare circuits, and writing a 1 to one of the bits would clear that particular event. If Event mode is not selected, then the status bits read back the compare outputs, and writing to those bits has no effect. Note that since this logic is in the Working power domain, MFGPT6 and MFGPT7 would lose these events when  $V_{CORE}$  is powered off. In order for events to be captured again, the chip has to have  $V_{CORE}$  powered up out of Standby mode and then come out of reset.

The Compare 1 and Compare 2 outputs may change simultaneously on the same MFGPT clock edge. However, when checking the outputs through the two status bits after this occurred, on rare occasions the read may find only one of the two outputs changed to the new value. This could occur when the two outputs change at about the same time they are synchronized, by separate synchronizers to the local bus clock domain, and one synchronizer captured the new value in time while the other one does not. A subsequent read can show that both outputs did change states.

#### 5.16.2.3 Register Initialization Sequence for Event Mode

If the Setup register is written before the other three I/O registers, and if Event mode is selected for Compare 1 mode or Compare 2 mode, then events will be triggered immediately. This is because the compare outputs will look for a Compare register value greater than or equal to the counter, and the result will be true as those registers are all 0. To avoid triggering these events on Setup register initialization, first initialize the Compare 1 Value and Compare 2 Value registers before initializing the Setup register.

#### 5.16.2.4 Register Data Transfer to/from MFGPT

Only WORD writes and DWORD writes are accepted for I/O register accesses; BYTE writes to I/O registers are ignored. The DWORD write would cause the two I/O registers located within the DWORD boundaries to be written in parallel. If Up Counter, Compare 1 Value, or Compare 2 Value registers are written while the MFGPT is running, it could cause the compare outputs to change in the middle of a prescaler period (i.e., not at a clock cycle where the prescaler signals a counter increment). For MFGPT0 to MFGPT5, the clock switch circuitry disables all clocks to MFGPT until the Setup register has been written. Therefore, even if the Up Counter, Compare 1 Value, and/or Compare 2 Value registers are written before the Setup register; these register values would get transferred to the timer at the same time as the Setup register values.

All reads and writes to MFGPT registers can be done by software at any time and are completed without requiring any additional software operation and without affecting the proper operation of the MFGPT as long as a clock to the MFGPT has been selected by writing to the Setup register.

On a write, the write transfer on the bus is considered complete when the write to the register in the I/O register submodule is complete. This occurs before the register data is transferred to the timer. However, a subsequent read or write to the same register will be held up until that first write transfer to the timer is complete.

The Setup register, except for bits 13 and 14, are handled in the same way as the Compare 1 Value and Compare 2 Value registers. Bits 13 and 14 write and read were discussed earlier, where the entire logic is in the Working power domain.

#### 5.16.2.5 Register Re-initialization

If it is necessary to re-initialize the Up Counter, Compare 1 Value, or Compare 2 Value, the following sequence should be followed to prevent any spurious reset, interrupt, or output pulses from being created:

- 1) Clear Counter Enable bit to 0.
- 2) Clear Interrupt Enable, NMI Enable, and Reset Enable bits in MSRs; disable GPIO inputs and outputs.
- 3) Update Up Counter, Compare 1 Value, and Compare 2 Value registers as desired.
- 4) When updates are completed, clear any event bits that are set.
- 5) Set up Interrupt Enable, NMI Enable, and Reset Enable bits in MSRs; enable desired GPIO inputs and outputs.
- 6) Set Counter Enable bit to 1.

### 5.16.3 Clock Switch

The clock switch output is disabled at reset and selection can only be done one time after reset, at the first write to Setup register.

#### Restriction on Register Read/Write Sequence Due to Clock Switch

Note that because the timer clock is stopped until the first write to the Setup register, a write to one of the other three I/O registers during this time could not complete its transfer to the timer. As a result, a second access, read or write, to the same register will cause the bus interface to hang, as the second access waits for the first access (the initial write) to complete before completing its own operation. But since the first access cannot complete without a clock, the second access is in limbo. This means no more accesses can occur, so there is no way to write to the Setup register to enable the timer clock. Care should be taken to see that this situation does not occur.



### 5.16.4 Single MFGPT

Figure 5-51 shows the functionality of one of these timers. There are two types of timers, one for the Working power domain and one for the Standby power domain.

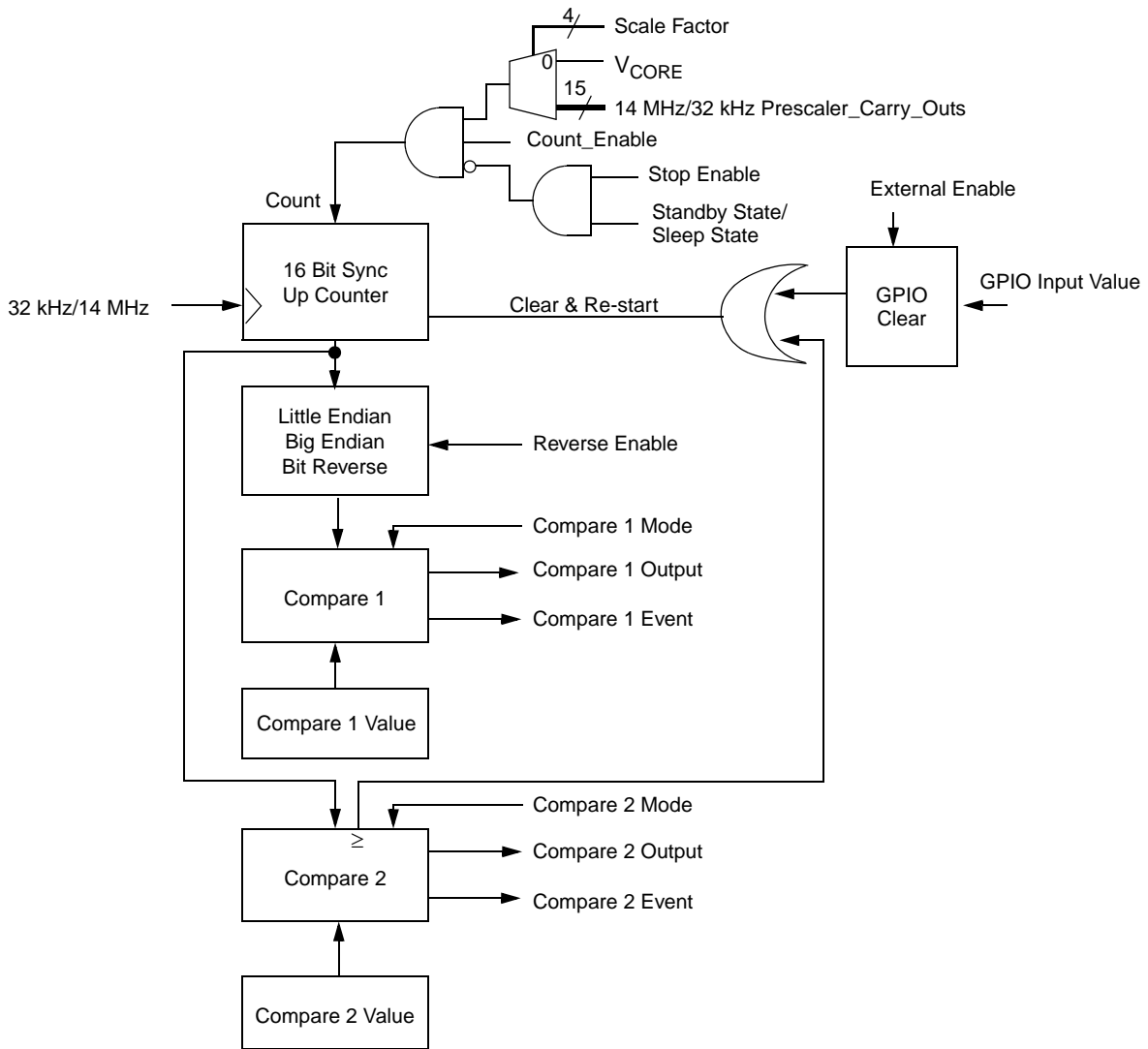


Figure 5-51. MFGPT Block Diagram

### 5.16.4.1 Clock Selection and Counter Increment

The MFGPT can use either the 32 kHz clock or 14.318 MHz clock as the clock source (MFGPT6 and MFGPT7 in Standby power domain are limited to the 32 kHz clock). When the Counter Enable bit is high, the MFGPT is activated and capable of counting. An actual increment is performed when the selected prescaler divide-by signals the increment; this is done through the Scale Factor selecting one of 16 signals. Table 5-32 shows how the Scale Factor effectively divides down the incoming clock.

**Table 5-32. MFGPT Prescaler Clock Divider**

Scale Factor	Input Clock Divide-By
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024
1011	2048
1100	4096
1101	8192
1110	16384
1111	32768

### 5.16.4.2 Compare 1 and Compare 2 Outputs

When the Up Counter reaches the Compare 1 Value, the Compare 1 Output is asserted. When the Compare 2 Value is reached, the Compare 2 Output is asserted, and the Up Counter then synchronously clears and restarts. The MFGPT outputs coming from Compare 1 and Compare 2 are all glitch-free outputs.

The compare outputs and events may change in the middle of a prescaler period if new values are written to the Up Counter, Compare 1 Value, or Compare 2 Value registers. These compare outputs can be used to trigger their respective events and drive GPIO outputs. The events are used to trigger interrupts, NMI, and reset.

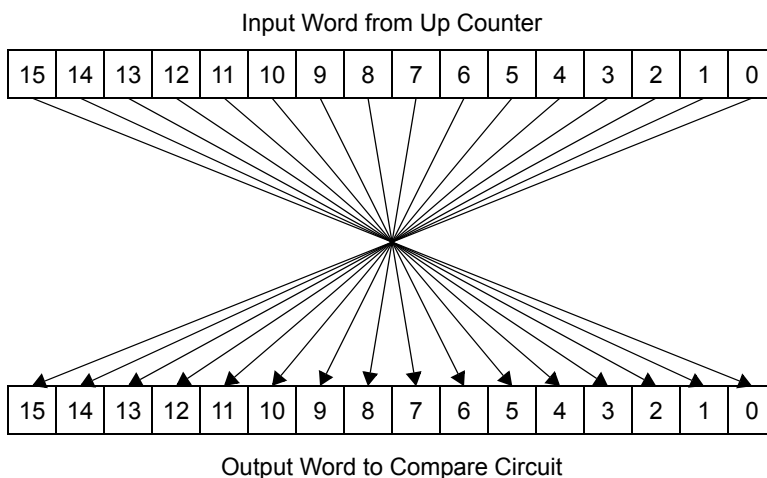
### 5.16.4.3 GPIO Input

The Up Counter could also be software selected to have a GPIO input positive edge as another source for the counter to clear and restart. The GPIO input signal is asynchronous to the timer and the timer uses a flip-flop to capture the GPIO rising edge. It takes up to one prescaler clock period plus two MFGPT clock periods from the GPIO rising edge for the clear to take effect. Once the counter is cleared, this edge detect circuit can then accept a new GPIO edge. Each individual pulse can be as short as a few nanoseconds wide for the rising edge to be captured. If this feature is not selected or the counter is disabled, the clear counter output and the edge detector are kept de-asserted.

### 5.16.4.4 Bit Reverse and Pulse Density Modulation

Figure 5-52 shows how the Little Endian/Big Endian Bit Reverse functions.

Table 5-33 on page 171 shows a 3-bit example of pulse density modulation; note that the MFGPT has a 16-bit implementation. If the desired pulse train is of the opposite polarity, this can be inverted in the GPIO or generated with a different Compare 1 value.



**Figure 5-52. MFGPT Bit Reverse Logic**

**Table 5-33. MFGPT Pulse Density Modulation Example**

Up Counter Output	Bit Reverse Output	Pulse Train Output for Given Compare 1 Value (Note 1)				
		1	2	4	5	6
000	000	0	0	0	0	0
001	100	1	1	1	0	0
010	010	1	1	0	0	0
011	110	1	1	1	1	1
100	001	1	0	0	0	0
101	101	1	1	1	1	0
110	011	1	1	0	0	0
111	111	1	1	1	1	1

Note 1. Compare 2 Value must be set to all 1s for pulse density modulation.

### 5.16.5 Working Power Domain Logic

The Working power domain logic consists of the Local Bus Interface, Control Logic, MSRs, and NMI, IRQ, and Reset Output Events.

When Event mode is enabled, the NMI, IRQ, and Reset Output Events logic gathers the event outputs of all eight MFGPTs and then generates the interrupt and resets outputs based on MSR settings. The interrupt outputs go to the PIC that can then trigger an IRQ or ASMI. Note that MFGPT6 and MFGPT7 cannot trigger reset. These outputs are controlled by MSR bits, and the NMI output can be further controlled by the MSB of I/O Address 070h.

### 5.16.6 Power Domain Crossing Interface Logic

The asynchronous internal Standby State signal will disable/enable the Working power domain interface immediately. Therefore, any bus operation active at that time will have an indeterminate result.

## 5.17 Power Management Control

The Power Management Control (PMC) module is a GeodeLink™ Device whose function is to control all aspects of power management. Power management is event driven meaning that, in general, any action that the PMC performs is predicated on some event. These events can come from other GeodeLink Devices, including the CPU module inside the Geode GX processor for example, or events coming from other off-chip sources.

The PMC is compatible with the industry standard power management capabilities as defined in the *Advanced Configuration and Power Interface (ACPI) v2.0 specification*. An OS that conforms to ACPI can take advantage of the Geode CS5535 companion device ACPI support hardware.

Advanced Power Management (APM) is another power management approach that the PMC supports. APM is a subset of ACPI and therefore will not be directly discussed.

Components in a GeodeLink architecture based system have hardware and software means of performing power management which the PMC controls. A high performance computing system consumes multiple watts of power when fully on. However, with GeodeLink architecture, system power consumption is significantly lower on average through the use of power states that reduce power needs when the system is idle.

### 5.17.1 Power Domains

The PMC module consists of three blocks: Working, Standby and RTC.

- The Working block contains all circuits and functions associated with the Working power domain. It includes

the Working state machine, Local bus interface, ACPI power management registers, and power management supporting logic (i.e., counters, timer, CCU, etc.). The main function of the Working block is to put the system into Sleep, that is, turn off clocks to the system and disable I/Os to reduce power consumption.

- The Standby block contains all circuits and functions associated with the Standby power domain. It includes the Standby state machine, ACPI registers, and power management supporting logic (i.e., counters, reset, CCU, etc.). The function of the Standby logic is to control power to the Working power domain. The PMC disables all interfaces between the Standby and Working domains while the Working power is off. A special feature of the Standby state machine is called Skip which allows the system to power up immediately when power is applied. See the signal description for the PWR\_BUT# in Table 3-9 "GPIOx Available Functions Descriptions" starting on page 49 for additional details.
- The RTC block contains the timing circuits for keeping real time. These circuits are powered by  $V_{BAT}$  (ball A3). It is not a device requirement that the RTC block be powered during Mechanical Off. If a system design does not require that real time be kept, then  $V_{BAT}$  should be tied to ground.

### 5.17.2 Power States

Table 5-34 shows the supported ACPI power states and how they relate to the Geode GX processor/CS5535 companion device system. ACPI power states not described are not supported.

**Table 5-34. Supported ACPI Power Management States**

ACPI States			Hardware States			
Global System State	Sleep State	C State	Geode GX Processor State	Geode CS5535 Companion Device Working Logic	System Main Memory	Geode CS5535 Companion Device Standby Logic
G0: Working	S0	C0	FO	FO	FO	On
			AHCG	AHCG	AHCG	
		C1	Suspend on Halt	AHCG	AHCG	
G1: Idle	S1: Sleeping	C2	Sleep	Sleep	Auto-refresh	On (Sleep)
	S3: Save-to-RAM	Off	Off	Off	Auto-refresh	Standby
	S4: Save-to-Disk	Off	Off	Off	Off	
G2: Soft Off	S5	Off	Off	Off	Off	Standby
G3: Mechanical Off	Off	Off	Off	Off	Off	Off

### 5.17.2.1 ACPI System Power States

- **G0/S0:** Not Sleeping. Software is executing code or could be halted waiting for a system event.
- **G1/S1:** Requires explicit software action to enter this state. All Geode GX processor, Geode CS5535 companion device, and main memory states maintained. All system clocks may be turned off except 32 kHz or selected additional clocks may be left on as required. The PMC provides generic controls SLEEP\_X and SLEEP\_Y that may be used to control the “D” states of external system devices (not described in this data book, see ACPI specification for details). Two additional internal signals control PCI and IDE input and outputs. A wakeup event brings the system back to the opcode following the one that initiated entry into S1. Context restore operation is not required on the Geode GX processor, Geode CS5535 companion device, or main memory.
- **G1/S3:** Save-to-RAM state. Requires explicit software action to enter this state. The Geode CS5535 companion device and other system context are lost. System state is saved in the main memory. To properly support this state, main memory power must be controlled by WORKING power while the Geode CS5535 companion device, Geode GX processor, and all other system components power must be controlled by WORK\_AUX power. Note that this applies only to the Working domain of the Geode CS5535 companion device. The Standby domain must be continuously supplied from Standby power.
- **G1/S4:** Suspend-to-Disk state. Requires explicit software action to enter this state. Same as S3 state, but the system state is “saved” on the hard drive or other mass storage device. Only Standby power is on while in this state.
- **G2/S5:** Requires explicit software action to enter this state. All system context is lost and not saved. Operating system re-boot is required. The 32 kHz clock is kept running for Standby PMC and selected GPIO and MFGPT circuits.
- **G3:** Software action is not required to enter this state. Working power and Standby power are removed. The only domain that may be powered is the RTC. It is not a requirement that the RTC be powered.

### 5.17.2.2 CPU Power States

- **G0/S0/C0:** Processor actively executing instructions and clock running. Cache snoops supported.
- **G0/S0/C1:** HLT instruction executed. Usually occurs in the Operating System’s idle loop. Operating System waiting for Power Management Event (PME), interrupt, or ASMI. Cache snoops are supported while in this state, so bus mastering activity can safely occur.
- **G1/S1/C2:** Processor is in the lowest power state that maintains context in a software invisible fashion. Entered as part of the S1 sequence. The

SUSP#/SUSPA# signaling protocol indicates entry. SUSP# is not an explicit external signal, it is part of the CIS packet. (See Section 5.2.14 “CPU Interface Serial (CIS)” on page 87 for further details.) No explicit software action required. However, this state can be entered by explicit software action by reading the ACPI P\_LVL2 register provided by the Geode GX processor’s GLCP.

### 5.17.2.3 Hardware Power States

- **FO (Full On):** From a hardware reset, all clocks come up Full On or always running. Generally, the system should not be left in this state. The AHCG state should be used.
- **AHCG (Active Hardware Clock Gating):** This is the desired mode of operation; it utilizes automatic hardware clock gating. Latency to turn on a clock is near 0. This hardware state should be established at system initialization by BIOS code; after initialization it needs no additional support. AHCG is invisible to the Operating System, ACPI, or other software based power management facilities.
- **Suspend on Halt:** See CPU power state G0/S0/C1.
- **Sleep:** See CPU power state G1/S1/C2.
- **Auto-refresh:** The memory controller issues an auto-refresh command to the DRAMs. In this state, the DRAMs perform refresh cycles on their own without any additional commands or activity from the memory controller or the interface. As long as power to the DRAMs is maintained, the memory contents are retained.

### 5.17.2.4 PMC Control

Under S3, S4, and S5 power management states, all Working domain circuits, as well as the Geode GX processor, are turned off to conserve power. Under S3, the system memory is powered by V<sub>IO\_VSB</sub> in Standby Auto-refresh mode but otherwise, all other system components are also turned off.

The PMC is used to establish overall system power states. Normally, the Standby domain voltages are present anytime the system is plugged into the wall; if portable, anytime the battery is plugged in. Generally, G3 Mechanical Off (see Table 5-34 on page 172) only applies during storage or maintenance. Therefore, operationally speaking, the PMC Standby controller is always available to manage power. There is a class of system designs that do not require G1 and G2 global power states. These systems usually power-up WORKING and STANDBY power domains simultaneously when power is applied.

For supporting “Save-to-RAM” (G1/S3) the WORKING output is used to switch off/on the Working domain sources for system memory while the WORK\_AUX output is used to switch off/on the Working domain sources for everything else. Thus, the PMC can completely control the system power states via these outputs.

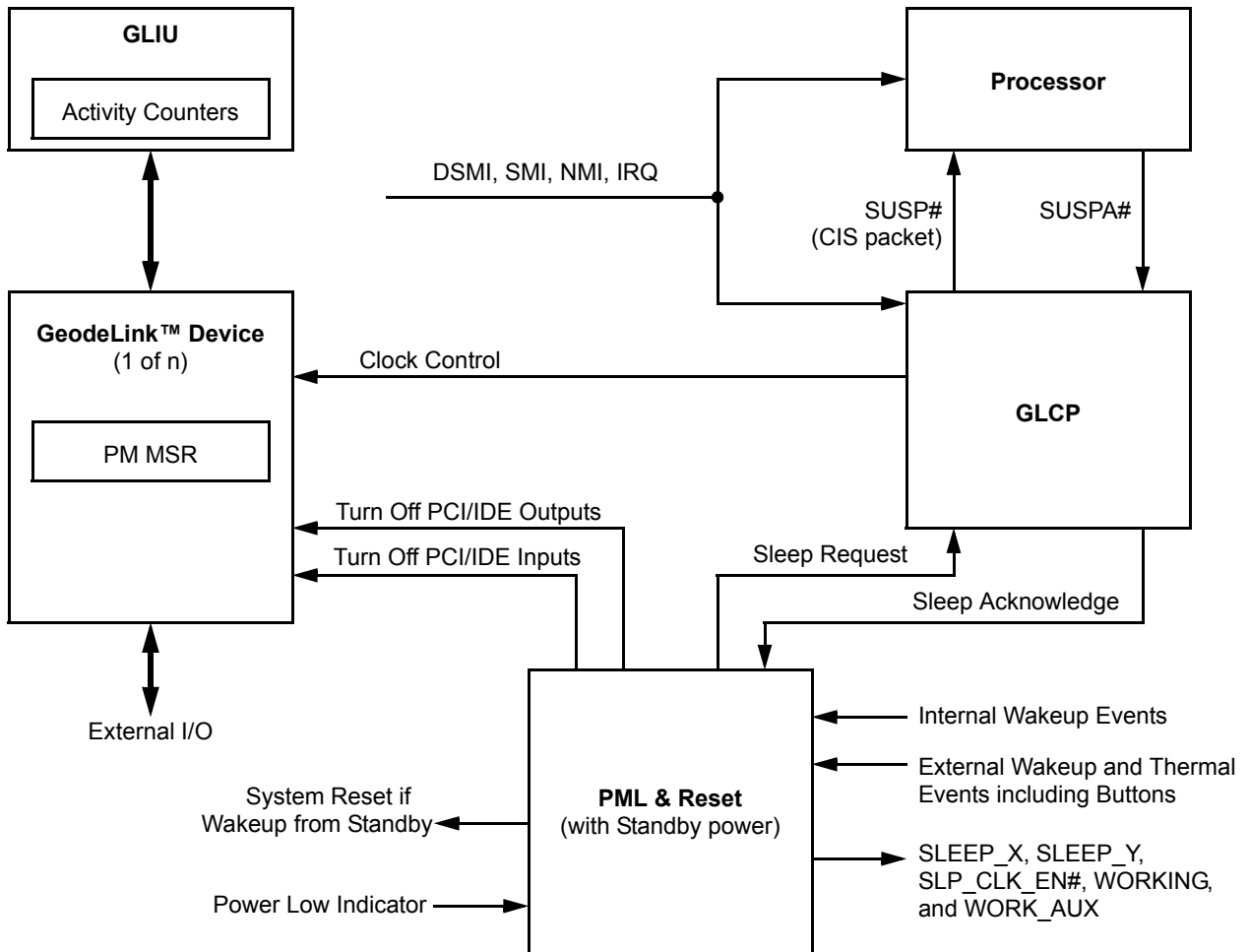
**5.17.3 Software Power Management Actions**

The hardware comes up from hardware system reset in the Full On (FO) state. As part of system initialization, the power management MSRs (see Section 6.18.1 on page 496) are written to establish the Active Hardware Clock Gating (AHCG) state. The AHCG state is the nominal operational state.

**5.17.3.1 Sleep/Standby Sequence**

Entering the states under G1 and G2 requires explicit software action. That action starts a hardware chain of events in which some of the chain is determined by registers that must be programmed previous to the start of the sequence. The block diagram of the hardware involved in this sequence is illustrated in Figure 5-53. Figure 5-54 and Figure 5-55 graphically show the Sleep/Standby sequence. The sequence is as follows:

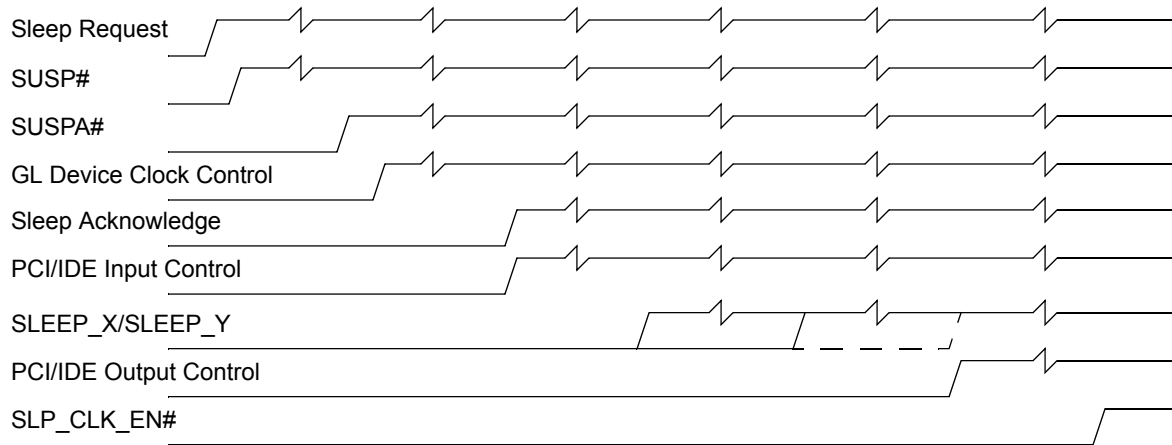
- 1) The "explicit software action" begins with a write to PM1\_CNT (ACPI I/O Offset 08h) starting the Sleep/Standby sequence.
- 2) The PMC issues a Sleep Request to the Geode CS5535 companion device GLCP and it passes the request as SUSP# to the Geode GX processor's GLCP.
- 3) The Geode GX processor's GLCP issues a suspend request to the processor. After the processor has shut-down operation it provides a suspend acknowledge back to the Geode GX processor's GLCP.
- 4) The Geode GX processor's GLCP processes a sleep sequence similar to that described in step 5 while issuing a SUSPA# to the Geode CS5535 companion device GLCP.



**Notes:** \*At least one per GeodeLink™ Device.  
#Global signal, one per system.

**Figure 5-53. PMC Power Management Elements**

- 5) The Geode CS5535 companion device GLCP processes a Sleep sequence. This is done in one of the three ways:
  - a) If the CLK\_DLY\_EN bit in GLCP\_GLB\_PM (MSR 5170000Bh[1]) is 0 and the CLK\_DELAY value in the GLCP\_CLK\_DIS\_DELAY (MSR 51700008h[23:0]) is 0, then wait until the CLK\_ACTIVE flags specified in GLCP\_CLK4ACK (MSR 51700013h[33:0]) have gone to 0.
  - b) If the CLK\_DLY\_EN bit is 1 and the CLK\_DELAY value is non-zero, then wait the amount of time of the CLK\_DELAY value.
  - c) If the CLK\_DLY\_EN is 0 and the CLK\_DELAY value is non-zero, then wait as in (a) but no longer than (b).
- 6) At the completion of the wait above, de-assert the CLK\_DIS bits specified in GLCP\_PMCLKDISABLE (MSR 51700009h[33:0]) and assert Sleep Acknowledge to the PMC.
- 7) When the Sleep Acknowledge is received, the PMC can optionally issue additional external generic controls SLEEP\_X and SLEEP\_Y as well as SLP\_CLK# to turn off external clocks. The completion of this step takes the system to S1. The system is now in Sleep.
- 8) If the Sleep Request was to enter S3 (Save-to-RAM) then the PMC moves beyond S1 and removes main power by de-asserting WORK\_AUX and leaving WORKING asserted. WORKING is used to power main memory, while WORK\_AUX is used for everything else in the system.
- 9) If the Sleep Request was to enter S4 (Save-to-Disk) or S5 (Soft Off) then the PMC moves beyond S1 and removes main power by de-asserting both WORK\_AUX and WORKING.
- 10) An external or internal wakeup event reverses the events above to bring the system back to the S0 state.

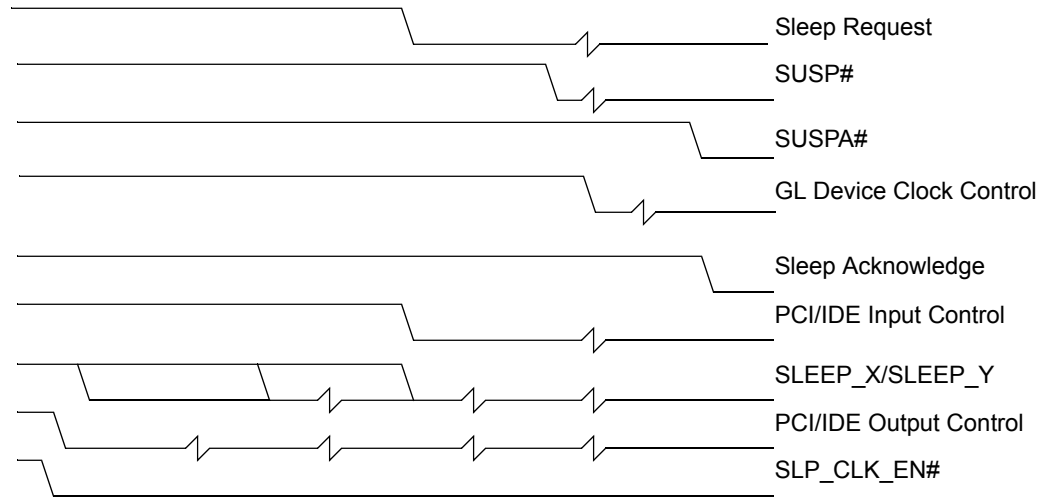


— Indicates a variable delay.

SLP\_CLK\_EN# signal must be the last control to assert because it turns off all system clocks.

**Note:** External signals are not necessarily active high. Shown as active high for clarity.

**Figure 5-54. PMC System Sleep Sequence**



—△— Indicates a variable delay.

SLP\_CLK\_EN# de-asserts at wakeup event and turns on system clocks.

SLEEP\_X/SLEEP\_Y Controls should de-assert between PCI/IDE input and output controls.

Wakeup sequence begins with a Sleep wakeup event.

**Note:** External signals are not necessarily active high. Shown as active high for clarity.

**Figure 5-55. PMC System Wakeup Sequence**



### 5.17.3.2 Sleep Controls

Sleep Request/Sleep Acknowledge handshake (see Figure 5-53 on page 174) between the GLCP and PMC controls the transitions into and out of the Sleep and Standby states. The PMC starts the Sleep sequence by asserting Sleep Request to the GLCP. The GLCP requests the processor to enter C2 by asserting the SUSP# signal. When SUSPA# from the processor is received, the GLCP informs the internal GeodeLink Devices of a pending shutdown and waits until the GeodeLink Devices' clock control indicates that they are ready. The length of time it takes for each device to respond is programmable (GLCP MSR 51700008h, 51700013h, and 5170000Bh). After all designated GeodeLink Devices have responded, the GLCP asserts Sleep Acknowledge to the PMC.

The PM\_IN\_SLPCTL (PMS I/O Offset 20h) register and the PM\_OUT\_SLPCTL (PMS I/O Offset 0Ch) are used to disable PCI/IDE inputs and outputs respectively during Sleep. Generally, they are asserted at the end of a Sleep sequence and de-asserted at the beginning of a Wakeup sequence. When "disabled", some of the outputs are forced to TRI-STATE with an active internal pull-down resistor while the rest are simply pulled low. See Section 4.8.5 "MSR Address 4: Power Management" on page 78 for specific details on PCI/IDE I/O controls during Sleep.

### 5.17.3.3 Power Controls

In response to Sleep Acknowledge from the GLCP, the PMC can assert five controls/enables: SLEEP\_X, SLEEP\_Y, SLP\_CLK\_EN#, WORKING, and WORK\_AUX. These can control external electronic power switches and enables. Each control's assertion and de-assertion is subject to an enable and a programmable delay (PMS I/O Offset 04h to 3Ch).

Controls SLEEP\_X and SLEEP\_Y are generic and have no specific use. Asserting control SLP\_CLK\_EN# is assumed to turn off the system (board) clocks. It is always de-asserted by the wakeup event. The following conditions apply to the timing of selected output control (see Section 5.17.3.2 "Sleep Controls"), SLEEP\_X, SLEEP\_Y and SLP\_CLK\_EN#.

#### When going to sleep:

- a) If not enabled, SLEEP\_X and SLEEP\_Y do not assert at all. If they are enabled, the delay should be set to occur between the delays programmed in the PM\_IN\_SLPCTL and PM\_OUT\_SLPCTL registers.
- b) If SLP\_CLK\_EN# is enabled, any delays associated with the PM\_OUT\_SLPCTL, SLEEP\_X, and SLEEP\_Y registers must be less than the SLP\_CLK\_EN# delay.
- c) If SLP\_CLK\_EN# is enabled, then Sleep wakeup is possible only after SLP\_CLK\_EN# asserts.
- d) If SLP\_CLK\_EN# is not enabled, and if at least one of the following PM\_OUT\_SLPCTL, SLEEP\_X, or SLEEP\_Y registers is enabled, then Sleep wakeup is possible only after the longest delay of the three. The delays could be zero.
- e) If SLP\_CLK\_EN# is not enabled, and the PM\_OUT\_SLPCTL, SLEEP\_X, or SLEEP\_Y registers are not enabled, then Sleep wakeup is possible immediately.
- f) If SLP\_CLK\_EN# is enabled and the delay associated with the PM\_OUT\_SLPCTL register is longer than or equal to the delay associated with SLP\_CLK\_EN#, then the PCI/IDE outputs will not be disabled.

If enabled, the de-assertion of WORKING is assumed to remove Working power and all clock sources except 32 kHz; that is, the Standby state is entered. In this state, the PMC, disables its interface to all circuits connected to Working power and asserts RESET\_OUT# before de-assertion of WORKING. RESET\_OUT# remains asserted throughout Standby.

WORK\_AUX is an auxiliary control for the Standby state with no specific use. It can be de-asserted any time before or after WORKING.

WORKING and WORK\_AUX are independent controls, but the use of either implies that Standby state is to be entered. In both cases, the PMC disables all circuits connected to Working power and asserts reset. However, since they are independent, one may be left on while the other is de-asserted.

### 5.17.3.4 Wakeup Events

If the system has been put to Sleep, only preprogrammed wakeup events can get the system running again. The PMC contains the controls that allow the system to respond to the selected wakeup events.

On wakeup from Sleep (not Standby, but Sleep Wakeup) (see Figure 5-53 on page 174), the PMC immediately de-asserts SLP\_CLK\_EN# to turn system clocks back on. It also re-enables PCI/IDE outputs to allow output drivers to return to their operational levels. Next it de-asserts SLEEP\_X and SLEEP\_Y based on programmable delays. Alternate SLEEP\_X and SLEEP\_Y interactions are shown as dotted lines. Lastly, the PMC, re-enables PCI/IDE inputs after a programmable delay and de-asserts Sleep Request. The GLCP starts any on-chip PLLs and waits for them to become stable. Then the GLCP de-asserts SUSP# to the processor. When the processor de-asserts SUSPA#, the GLCP de-asserts Sleep Acknowledge. The PMC allows the wakeup event to assert a System Control Interrupt (SCI).

#### After a wakeup event:

**a)** PCI/IDE outputs are re-enabled after SLP\_CLK\_EN# is de-asserted.

**b)** PCI/IDE inputs are re-enabled at Sleep wakeup or after a programmable delay. Generally, PCI/IDE inputs are normally used with a delay and that delay is longer than any de-assertion delay associated with SLEEP\_X and/or SLEEP\_Y. Re-enabling PCI/IDE inputs is generally not useful at the beginning of a wakeup sequence.

**c)** Sleep Request is de-asserted at Sleep wakeup or after a programmable delay. Sleep Request is kept de-asserted until the PCI/IDE inputs are re-enabled. Generally, the enable and delay values in PM\_SED (PMS I/O Offset 14h) and PM\_IN\_SLPCTL (PMS I/O Offset 20h) should always be the same.

**d)** If used, SLEEP\_X/SLEEP\_Y delay should be set to occur between the delays programmed in PM\_OUT\_SLPCTL (PMS I/O Offset 0Ch) and PM\_IN\_SLPCTL (PMS I/O Offset 20h). If the delays for SLEEP\_X/SLEEP\_Y are longer than the PM\_IN\_SLPCTL delay, then SLEEP\_X/SLEEP\_Y de-assert at the same time as the PCI/IDE inputs are re-enabled.

On wakeup from Standby (not Sleep, but Standby Wakeup) the PMC asserts WORKING and performs a system reset. RESET\_OUT# is de-asserted after a programmable delay and the normal software start-up sequence begins. However, early in the sequence, the software checks the PMC state to determine if waking from Standby (PMS I/O Offset 54h[0]). If yes, then the system state is potentially restored from non-volatile storage.

If enabled, WORK\_AUX may be asserted before or after RESET\_OUT# is de-asserted.

### 5.17.3.5 Fail-Safe Power Off

The PMC provides the support logic to implement an ACPI compliant fail-safe power off button. This logic unconditionally de-asserts the WORKING and WORK\_AUX signals if the On/Off button is held down for a programmable delay. For ACPI compliance, this delay should be set to four seconds.

### 5.17.3.6 Wake Events Status and SCI

When enabled, a wake event from the general wake events register (see Section 6.16.4 "GPIO Interrupt and PME Registers" on page 479) set its status bit and the "WAK\_STS" bit and causes a system control interrupt (SCI). The Sleep button, RTC alarm, and power button when asserted, always set their status bit. They set the "WAK\_STS" bit and generate an SCI only when their enable bit is set. When overflowed, the PM timer sets its status bit. This overflow condition does not cause a wakeup event but if enabled, it generates an SCI. The event's status is cleared by writing a one to it.

## 5.17.4 PMC Power Management States

The PMC state machines support the fundamental hardware states: Power Off, Reset Standby, Working, Sleep, and Controlled Standby.

- **Reset Standby State:** From Power Off, reset is applied to the Standby domain by the external input pin RESET\_STAND# or through the Low Voltage Detect (LVD) circuit. Once reset, and the Skip feature is not enabled, the Reset Standby state de-asserts WORKING and WORK\_AUX outputs and waits for a Reset Standby wakeup event. If the Skip feature is enabled, the state machine will proceed to the working state immediately. To enable Skip, see the signal description for the PWR\_BUTTON# in Table 3-9 "GPIOx Available Functions Descriptions" starting on page 49.
- **Working State:** The Working state can be entered from Reset Standby, Sleep, or Controlled Standby states. Working state is established when Working power is applied and all system clocks are enabled. Once in this state, registers and functions in the PMC can be initialized, programmed, enabled/disabled, and the potential exists for the system to proceed to the Sleep state or Standby state.
- **Sleep State:** The system initiates the entry to the Sleep state with a Sleep sequence. Under the Sleep state, Working and Standby power are maintained. PCI/IDE inputs are disabled when Sleep Acknowledge asserts. PCI/IDE outputs are disabled when Sleep Acknowledge asserts or after a programmable delay. SLEEP\_X, SLEEP\_Y, and SLP\_CLK\_EN# may be asserted if enabled. A Sleep wakeup event returns the system to Working state.

- **Controlled Standby State:** Can be entered “normally”, “fault condition”, or by a “restart”.

A normal entry is by way of a system initiated sequence as in the Sleep case. This method of entry requires the Standby state machine to monitor SLP\_CLK\_EN# and look for an enable of the “Working De-assert Delay and Enable” register (PM\_WKDD, PMS I/O Offset 30h[30]) or the Work\_aux De-assert Delay and Enable register (PM\_WKXD, PMS I/O Offset 34h[30]). This signals the Controlled Standby state normal entry. A Standby wakeup event returns the system to Working state after a programmable delay (PM\_NWKD, PMS I/O Offset 4Ch).

If enabled, a faulted entry can be initiated by a low power off, thermal off, or fail-safe off. It can also be initiated by Working power fail asserted. A default wakeup event returns the system to the Working state after a programmable delay (PM\_FWKD, PMS I/O Offset 50h).

If a power loss occurs and power is restored, the system can be returned to the Working state automatically by software setting the PI (Power Immediate bit, read, cleared, and set through PMS I/O Offset 54h[18:16]). The PI bit is powered by the RTC block, so it remains valid as long as V<sub>BAT</sub> remains connected. PI behaves similarly to Skip.

A re-start can be initiated by any of these resets: GLCP soft reset, soft reset, shutdown reset, watchdog reset, or bad packet type reset. The system returns to the Working state when reset is de-asserted and the faulted\_to\_work delay (PM\_FWKD) expired. WORKING and WORK\_AUX are not de-asserted.

When a Controlled Standby state is entered by a faulted condition or restart event, software control is assumed lost and the software established state is assumed to be potentially wrong. Therefore, the Standby domain returns to the state associated with “Standby State Entry from Power Off”; that is, Standby domain reset defaults are used. The only exceptions are registers from the following list; these are locked and not subject to change by software:

PM_RD	De-assert Reset Delay from Standby (PMS I/O Offset 38h)
PM_WKXA	WORK_AUX Assert Delay from Standby (PMS I/O Offset 3Ch)
PM_FSD	Fail-Safe Delay and Enable (PMS I/O Offset 40h)
PM_TSD	Thermal Safe Delay and Enable (PMS I/O Offset 44h)
PM_PSD	Power Safe Delay and Enable (PMS I/O Offset 48h)
PM_NWKD	Normal to Work Delay and Enable (PMS I/O Offset 4Ch)
PM_FWKD	Faulted to Work Delay and Enable (PMS I/O Offset 50h)

The Faulted to Work Delay and Enable (PM\_FWKD) register is the only one of the above registers that potentially applies during a re-start entry.

Lastly, note that any normal entry operation in process is aborted.

Wakeup from faulted entry is the same as that associated with Standby State Entry from Power Off; that is, it acts as if the power button has been pushed. Other possible wakeup events such as RTC Alarm and PMEs are ignored. However, the system can be held in the Standby state for the following reasons:

- 1) If enabled and locked, LOW\_BAT# is still asserted.
- 2) If LVD\_EN# is tied to ground and V<sub>CORE</sub> is not at a valid voltage, or if RESET\_WORK # is asserted.

**Note:** If enabled and locked, the thermal alarm does not keep the system in the Standby state if it is asserted. The thermal alarm circuitry resides in the Working domain, and its state is ignored by the Standby state. Once out of Standby, the thermal alarm again comes into play. If it is still asserted, its timer would start again.

The Power Management Control (PMC) has two state machines:

- **Working State Machine:** Operates under Working power and runs on a 14 MHz clock from the CCU. Its function is to generate control signals used to turn off/on systems clocks and I/Os based on events coming from on or off the chip.
- **Standby State Machine:** Operates under Standby power and runs on the 32 kHz clock. Its function is to power-up and down the Working power to the Working domain based on events coming from on or off the chip.

**5.17.5 PMC Power Management Events**

A large number of inputs to the PMC are used to monitor and create system power managements events. Some of these inputs apply the Working state machine while the remainder apply to the Standby state machine.

**5.17.5.1 PM Sleep Events**

- Sleep:
  - Sleep sequence initiated by software
- Wakeup:
  - Assertion of the Sleep Button (SLEEP\_BTN)
  - Assertion of the Power Button (PWR\_BTN#)
  - RTC alarm
  - Working power domain PMEs
  - Standby power domain PMEs

**5.17.5.2 PM Standby Events**

- Standby:
  - Sleep sequence initiated by software
  - LVD detection of low voltage on V<sub>CORE</sub> (system fault)

- Assertion of the Power Button for 4 seconds (PWR\_BTN#, system fault)
- Thermal Alarm (THRM\_ALARM#, system fault)
- Low battery (LOW\_BAT#, system fault)
- Hardware reset (system restart)
- Software initiated reset (system restart)
- Shutdown initiated reset, CPU triple fault (system restart)
- Watchdog initiated reset (system restart)
- GLCP software initiated reset (system restart)
- Bad packet type reset (system restart)
- Reset Standby state machine RESET\_STANDBY# (Standby)
- Wakeup:
  - Assertion of the Power Button (PWR\_BTN#)
  - RTC Alarm
  - Standby power domain PMEs

Table 5-35 provides a complete list of the power management inputs and describes their function. The system can only be in one of three states: Working, Sleep, or Standby. The activity of the inputs is to move the system from one state to another.

**Table 5-35. PM Events and Functions**

Event	Current State	Function
<b>The following events are Sleep and/or Standby wakeup events (except for ACPI Timer).</b>		
PWR_BTN# (also serves as a Standby event)	Working	Sets the status bit (PWRBTN_STS) in PM1_STS (ACPI I/O Offset 00h[8] = 1). If PWRBTN_EN is enabled (ACPI I/O Offset 02h[8] = 1), an SCI is generated.
	Sleep	Sets the status bit (PWRBTN_STS) in PM1_STS (ACPI I/O Offset 00h[8] = 1). If PWRBTN_EN is enabled (ACPI I/O Offset 02h[8] = 1), SCI generation and wakeup from this event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
	Standby	Sets the status bit (PWRBTN_STS) in PM1_STS (ACPI I/O Offset 00h[8] = 1). If PWRBTN_EN is enabled (ACPI I/O Offset 02h[8] = 1), SCI generation and wakeup from this event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
SLP_BTN	Working	Sets the status bit (SLPBTN_STS) in PM1_STS (ACPI I/O Offset 00h[9] = 1). If SLPBTN_EN is enabled (ACPI I/O Offset 02h[9] = 1), an SCI is generated.
	Sleep	Sets the status bit (SLPBTN_STS) in PM1_STS (ACPI I/O Offset 00h[9] = 1). If SLPBTN_EN is enabled (ACPI I/O Offset 02h[9] = 1), SCI generation and wakeup from this event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
RTC Alarm	Working	Sets the status bit (RTC_STS) in PM1_STS (ACPI I/O Offset 00h[10] = 1). If RTC_EN is enabled (ACPI I/O Offset 02h[10] = 1), an SCI is generated.
	Sleep	Sets the status bit (RTC_STS) in PM1_STS (ACPI I/O Offset 00h[10] = 1). If RTC_EN is enabled (ACPI I/O Offset 02h[10] = 1), SCI generation and wakeup from this event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
	Standby	Sets the status bit (RTC_STS) in PM1_STS (ACPI I/O Offset 00h[10] = 1). If RTC_EN is enabled (ACPI I/O Offset 02h[10] = 1), SCI generation and wakeup from this event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
ACPI Timer (internal timer)	Working	Sets the status bit (TMR_STS) in PM1_STS (ACPI I/O Offset 00h[0] = 1). If TMR_EN is enabled (ACPI I/O Offset 02h[0] = 1), an SCI is generated.

Table 5-35. PM Events and Functions (Continued)

Event	Current State	Function
GPE[23:0] (General Purpose Power Management Events in Working Domain)	Working	If GPE_EN[23:0] are enabled (ACPI I/O Offset 1Ch[23:0] = 1), the corresponding status bit (GPE_STS[23:0]) in GPE0_STS (ACPI I/O Offset 18h[23:0]) is set and an SCI is generated.
	Sleep	If GPE_EN[23:0] are enabled (ACPI I/O Offset 1Ch[23:0] = 1), SCI generation and wakeup from the event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
GPE[31:24] (General Purpose Power Management Events in Standby Domain)	Working	If GPE_EN[31:24] are enabled (ACPI I/O Offset 1Ch[31:24] = 1), the corresponding status bit (GPE_STS[31:24]) in GPE0_STS (ACPI I/O Offset 18h[23:0]) is set and an SCI is generated.
	Sleep	If GPE_EN[31:24] are enabled (ACPI I/O Offset 1Ch[23:0] = 1), SCI generation and wakeup from the event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
	Standby	If GPE_EN[31:24] are enabled (ACPI I/O Offset 1Ch[31:24] = 1), the corresponding status bit (GPE_STS[31:24]) in GPE0_STS (ACPI I/O Offset 18h[23:0]) is set, and SCI generation and wakeup from the event is enabled (i.e., sets the WAK_STS bit, ACPI I/O Offset 00h[15] = 1).
The following events caused a Standby state entry.		
RESET_STAND#	Working	If asserted, the corresponding status bit (OFF_FLAG) in PM_SSC (PMS I/O Offset 54h[0]) is set and causes a Reset Standby state entry. No Working or Standby power.
	Sleep	If asserted, the corresponding status bit (OFF_FLAG) in PM_SSC (PMS I/O Offset 54h[0]) is set and causes a Reset Standby state entry. No Working or Standby power.
	Standby	If asserted in Restart, or Normal or Faulted Standby state, the corresponding status bit (OFF_FLAG) in PM_SSC (PMS I/O Offset 54h[0]) is set and causes a Reset Standby state entry.
LVD circuit detects low voltage on V <sub>CORE</sub>	Working	If de-asserted, the status bit (LVD_FLAG) in PM_SSC (PMS I/O Offset 54h[2]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Sleep	If de-asserted, the status bit (LVD_FLAG) in PM_SSC (PMS I/O Offset 54h[2]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Standby	If de-asserted in Restart state, the status bit (LVD_FLAG) in PM_SSC (PMS I/O Offset 54h[2]) is set and causes an Faulted Standby state entry.
PWR_BUT#	Working	If enabled and asserted for four seconds (fail-safe), the status bit (PWRBUT_FLAG) in PM_SSC (PMS I/O Offset 54h[3]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Sleep	If enabled and asserted for four seconds (fail-safe), the status bit (PWRBUT_FLAG) in PM_SSC (PMS I/O Offset 54h[3]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Standby	If enabled and asserted for four seconds (fail-safe) while in Normal or Restart state, the status bit (PWRBUT_FLAG) in PM_SSC (PMS I/O Offset 54h[3]) is set and causes a Faulted Standby state entry.
THRM_ALRM#	Working	If enabled and asserted for a programmable amount of time, the status bit (THRM_FLAG) in PM_SSC (PMS I/O Offset 54h[4]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Sleep	If enabled and asserted for a programmable amount of time, the status bit (THRM_FLAG) in PM_SSC (PMS I/O Offset 54h[4]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Standby	Ignored.

Table 5-35. PM Events and Functions (Continued)

Event	Current State	Function
LOW_BAT#	Working	If enabled and asserted for a programmable amount of time, the status bit (LOWBAT_FLAG) in PM_SSC (PMS I/O Offset 54h[5]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Sleep	If enabled and asserted for a programmable amount of time, the status bit (LOWBAT_FLAG) in PM_SSC (PMS I/O Offset 54h[5]) is set and causes an Faulted Standby state entry. Working power is turned-off.
	Standby	If asserted in Normal or Re-start state, the status bit (LOWBAT_FLAG) in PM_SSC (PMS I/O Offset 54h[5]) is set and causes a Faulted state entry.
RESET_WORK#	Working	If asserted, the status bit (HRD_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[6]) is set and causes a Restart state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (HRD_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[6]) is set and causes a Restart state entry. Working power is not turned-off.
Software initiated reset	Working	If asserted, the status bit (SFT_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[8]) is set and causes a Restart state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (SFT_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[8]) is set and causes a Restart state entry. Working power is not turned-off.
Shutdown initiated reset (CPU triple fault)	Working	If asserted, the status bit (SHTDWN_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[9]) is set and causes a Re-start state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (SHTDWN_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[9]) is set and causes a Restart state entry. Working power is not turned-off.
Watchdog initiated reset	Working	If asserted, the status bit (WATCHDOG_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[10]) is set and causes a Restart state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (WATCHDOG_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[10]) is set and causes a Restart state entry. Working power is not turned-off.
GLCP Soft Reset	Working	If asserted, the status bit (GLCP_SFT_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[11]) is set and causes a Restart state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (GLCP_SFT_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[11]) is set and causes a Restart state entry. Working power is not turned-off.
Bad packet type reset	Working	If asserted, the status bit (BADPACK_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[12]) is set and causes a Restart state entry. Working power is not turned-off.
	Sleep	If asserted, the status bit (BADPACK_RST_FLAG) in PM_SSC (PMS I/O Offset 54h[12]) is set and causes a Restart state entry. Working power is not turned-off.

## 5.18 Flash Controller

The Geode CS5535 companion device has a Flash device interface that supports popular NOR Flash and inexpensive NAND Flash devices. This interface is shared with the IDE interface (ATA-5 Controller (ATAC)), using the same balls. NOR or NAND Flash may co-exist with IDE devices using PIO (Programmed I/O) mode. The 8-bit interface supports up to four “lanes” of byte-wide Flash devices through use of four independent chip selects, and allows for booting from the array. Hardware support is present for SmartMedia-type ECC (Error Correction Code) calculations, off-loading software from having to support this task. If Flash and IDE are both operational, an external pull-up (10K) to IDE\_DREQ0 (ball A14) is required and IDE\_DREQ0 must not be connected to the IDE device. To switch modes, explicit software actions must occur to disable one and enable the other.

### Features

- Supports popular NOR Flash and inexpensive NAND Flash devices on IDE interface. No extra pins needed.
- NOR Flash and NAND Flash co-exist with IDE devices with PIO (Programmed I/O) only mode.
- General purpose chip select pins support on-board ISA-like slave devices.
- Programmable timing supports a variety of Flash devices.
- Supports up to four byte-wide NOR Flash devices.
  - Address up to 256 KB boot ROMs using an external octal latch.
  - Address up to 256 MB linear Flash memory arrays using external latches.
  - Boot ROM capability.
  - Burst mode capability (DWORD read/write on PCI bus).
- Supports up to four byte-wide NAND Flash devices.
  - Hardware support for SmartMedia-type ECC (Error Correction Code) calculation off-loading software effort.
- Supports four programmable chip select pins with memory or I/O addressable.
  - Up to 1 KB of address space without external latch.

### 5.18.1 NAND Flash Controller

To understand the functioning of the NAND Flash Controller, an initialization sequence and a read sequence is provided in the following sub-sections. The NAND Flash Controller's registers can be mapped to memory or I/O space. The following example is based on memory mapped registers.

#### 5.18.1.1 Initialization

- 1) Program MSR\_LBAR\_FLSH0 (MSR 51400010h) to establish a base address (NAND\_START) and whether in memory or I/O space. The NAND Controller is memory mapped in this example and always occupies 4 KB of memory space.
- 2) Set the NAND timing MSRs to the appropriate values (MSRs 5140001Bh and 5140001Ch).

#### 5.18.1.2 Read

- 1) Allocate a memory buffer. Start at address BAh in system memory.
- 2) Fill the buffer with the following values:
  - BA: 02h (Assert CE#, CLE)
  - BA + 1: 00h (Command: Read mode)
  - BA + 2: 04h (Assert CE#, ALE, De-assert CLE)
  - BA + 3: CA (Start column address)
  - BA + 4: 04h
  - BA + 5: PA0 (Page address byte 0)
  - BA + 6: 04h
  - BA + 7: PA1 (Page address byte 1)
  - BA + 8: 04h
  - BA + 9: PA2 (Page address byte 2)
  - BA + 10: 08h (Assert CE#, De-assert ALE, Enable Interrupt)
- 3) For (i = 0; i < 11; i++), write the data in buffer [BA+i] to memory location [NAND\_START + 800h + i]. Generate the command and address phase on the NAND Flash interface.

- 4) NAND Flash device may pull down the RDY/BUSY# signal at this point. Software sets the EN\_INT bit and waits for the interrupt.
- 5) Memory byte writes 03h to memory location [NAND\_START + 815h] to clear ECC parity and Enable ECC engine.
- 6) For (i = 0; i < 256; i++), read data from [NAND\_START + i] to buffer [BA + i] (read data from NAND Flash to memory buffer. Can use DWORD read to save time).
- 7) Memory DWORD Reads [NAND\_START + 810h] to get ECC parity [ECC0] of first 256 byte data.
- 8) Memory byte writes 03h to memory location [NAND\_START + 815h] to clear ECC parity and enable ECC engine.
- 9) For (i = 256; i < 512; i++), read data from [NAND\_START + i] to buffer [BA + i] (read data from NAND Flash to memory buffer. Can use DWORD read to save time).
- 10) Memory DWORD reads [NAND\_START + 810h] to get ECC parity [ECC1] of second 256 byte data.
- 11) For (i = 512; i < 528; i++), read data from [NAND\_START + i] to buffer [BA + i] (read data from NAND Flash redundant data to memory buffer. Can use DWORD read to save time).
- 12) Write 01h to memory location [NAND\_START + 800h] (de-assert CE#, NAND Flash enters to Idle state).
- 13) Retrieve ECC parity data from redundant data area and compare them to ECC0 and ECC1.
- 14) Correct data if data error is detected and can be fixed.

Figure 5-56 on page 184 shows a basic NAND read cycle.

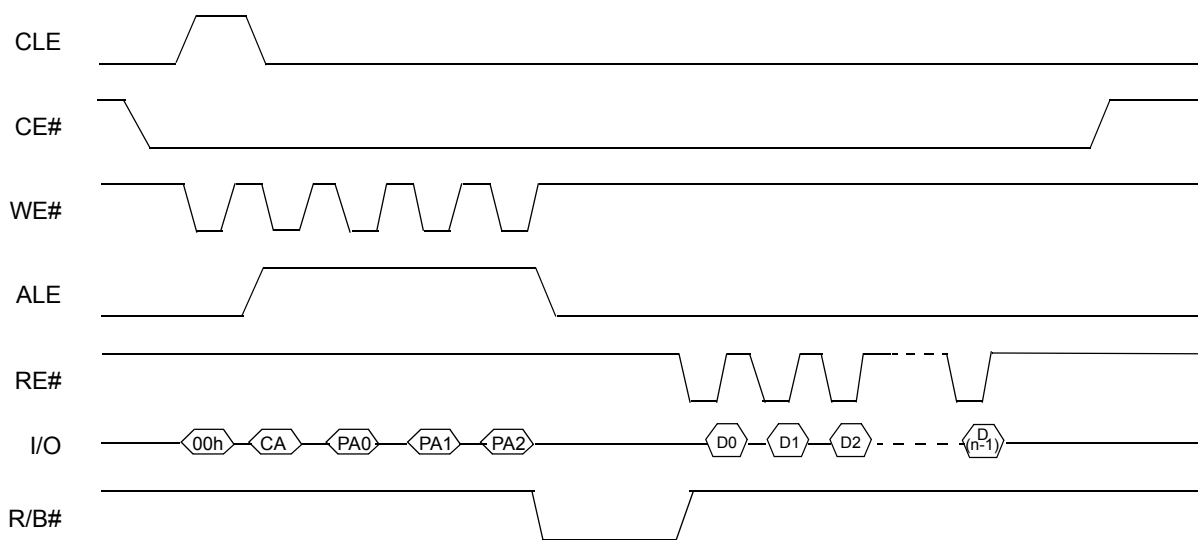


Figure 5-56. Flash Controller NAND Read Cycle



### 5.18.1.3 NAND ECC Control Module

The NAND ECC Control Module is part of the NAND Flash Controller. It calculates 22-bit ECC parity for each of the 256 bytes of the NAND Flash's data transferred on the Local bus. The ECC calculation algorithm follows the *SmartMedia Physical Format Specification*. The ECC algorithm is capable of single-bit correction and 2-bit random-error detection. ECCs are generated only for data areas and no ECC is generated for page-data redundant areas containing ECCs as the page-data redundant area is duplicated for reliability. For ECC calculations, 256 bytes are handled as a stream of 2048-bit serial data. In the event of an error, the error-correction feature can detect the bit location of the error based on the results of a parity check and correct the data.

#### Hardware Operation

The ECC engine treats 256-byte data as a block. Each byte has an 8-bit address called a Line Address (LA). Each bit in a byte has a 3-bit address called a Column Address (CA). Combining these two address fields forms an 11-bit unique address for every single bit in the 256-byte data block. The address uses the notation: LLLL\_LLLL, CCC. This module contains an 8-bit counter to keep track of the LA of each byte. Each ECC parity bit calculation in the ECC engine produces even parity of half of the data bits in the block. Different parity bits use different sets of the bits. For example, CP0 is the even parity bit of the bits with Column Address bit 0 equals 0. CP1 is the even parity bit of the bits with Column Address bit 0 equals 1. Both odd and even parity are supported for ECC. The ECC parity available in the NAND ECC column, LSB line, and MSB line parity registers is the inverted output of the ECC parity from the ECC engine in the case of odd ECC parity and the non-inverted output in the case of even parity. Table 5-36 lists the relationship between the parity bits and the corresponding bit addresses. The hardware ECC engine calculates 22-bit ECC parity whenever there is a data write or data read to/from the NAND Flash device. On power-up, the ECC engine is configured to be odd parity. Even or odd ECC parity is controlled by bit 2 of NAND ECC Control register (Flash Memory Offset 815h).

**Table 5-36. ECC Parity/Bit Address Relationship**

Parity	Bit Address
CP0	xxxx_xxxx, xx0
CP1	xxxx_xxxx, xx1
CP2	xxxx_xxxx, x0x
CP3	xxxx_xxxx, x1x
CP4	xxxx_xxxx, 0xx
CP5	xxxx_xxxx, 1xx
LP00	xxxx_xxx0, xxx
LP01	xxxx_xxx1, xxx
LP02	xxxx_xx0x, xxx
LP03	xxxx_xx1x, xxx
LP04	xxxx_x0xx, xxx
LP05	xxxx_x1xx, xxx
LP06	xxxx_0xxx, xxx
LP07	xxxx_1xxx, xxx
LP08	xxx0_xxxx, xxx
LP09	xxx1_xxxx, xxx
LP10	xx0x_xxxx, xxx
LP11	xx1x_xxxx, xxx
LP12	x0xx_xxxx, xxx
LP13	x1xx_xxxx, xxx
LP14	0xxx_xxxx, xxx
LP15	1xxx_xxxx, xxx

### Software Operation

The NAND Flash contains a redundant data area containing ECC fields. While writing to the NAND Flash, the hardware ECC engine calculates ECC parity, if it is enabled properly. Software can write the ECC parity bits to the ECC field after writing the data area. When software reads the data from NAND Flash, the hardware ECC engine calculates ECC parity. After the data is read from the NAND Flash, software can compare the ECC parity in the hardware ECC engine and the ECC parity in the ECC field of the NAND Flash to determine if the data block is correct.

Each data bit has 11 corresponding parity bits, which can be determined by the bit address. If one data bit is different from its original value, 11 ECC parity bits are changed from the original ECC parity bits. Take the ECC parity from the hardware ECC engine and perform bit-wise exclusive OR with it and the ECC parity field in NAND Flash. The result can be as follows.

- 1) All bits are 0. The data is correct.
- 2) Eleven bits are 1. One bit error has been detected. Use the eleven bits to identify the error bit position.
- 3) One bit is 1. One bit in ECC field is corrupt. Data area should be OK.
- 4) Otherwise, two or more data bits are corrupt. Cannot be corrected.

#### 5.18.2 NOR Flash Controller/General Purpose Chip Select

The NOR Flash Controller supports up to four independent chip selects that can be used for NOR Flash devices or General Purpose Chip Selects (GPCS). Up to 28 bits of address is supported for each chip select, allowing byte-wide linear arrays up to 256 MB in memory space. Chips selects may also be located in I/O space, but the usable address bits are limited by the overall limits of I/O space.

Each chip select is independently programmable:

- Address Setup: Defaults to seven Local bus clocks
- Read/Write Strobe Width: Defaults to seven Local bus clocks
- Address Hold: Defaults to seven Local bus clocks cycles
- Optional Wait State Insertion: Defaults off, driven by an external input (FLASH\_IOCHRDY) to be used by General Purpose devices.
- Optional Write Protect: Defaults protected

These settings are located in MSR space and on hard reset default to the settings listed above. Hence, virtually any NOR device can be used immediately out of reset for first instruction fetch. After booting, delays can be programmed as appropriate.

Special considerations must be made for NOR Flash write operations. Depending on the manufacturer and write mode, each write can take from a few microseconds to a few hundred microseconds. Specifically, the software performing the write must observe the following procedure:

- 1) Write to device.
- 2) Wait an amount of time dependent on manufacturer's specifications.
- 3) Repeat from #1 until all writes are completed.

The "wait" in step two is implemented using an appropriate time base reference. There is no reference within the Flash Controller subsystem.

Some NOR devices provide a ready line that de-asserts during the "wait" in step two. Direct use of this signal is not supported by the Flash Controller. The NOR write software should use an appropriate time base reference to determine when the device is ready, that is, determine how long to wait for the current write to complete before starting another write. Alternatively, the NOR device internal status may be read to determine when the write operation is complete. Refer to NOR Flash manufacturer's data sheets for additional write operation details.

### 5.18.3 Flash Controller Interface Timing Diagrams

#### 5.18.3.1 NOR/GPCS

The NOR/GPCS timing has two phases: address phase and data phase.

In the address phase, the address bus and data bus present a higher address, ADD[27:10]. Board designers can use external latches, such as 74x373, to latch the address bits.

In the data phase, the address bus presents ADD[9:0], and the data bus is for data read or write.

The Flash Controller is running off internal Local bus clock, which is at the highest frequency of 33 MHz. The address phase is always two clock periods. The ALE signal asserts high in the first-half clock period and de-asserts in the sec-

ond clock period. A 74LCX373 only needs 4 ns setup time and 2 ns hold time (worst case). This timing provides a lot of flexibility for the designing of the board. In the data phase, the address bus and write data bus are available in the first clock period. In the second clock period of the data phase, chip select goes low. After the required hold time, chip select goes high, and write data bus change. After one Local bus clock from chip select change (going high), address bus changes. The setup time, strobe pulse width, and hold time are programmable through the NOR timing registers. See Section 6.19.1.2 "NOR Flash Timing MSRs" on page 522.

Figure 5-57 and Figure 5-58 provides some NOR Flash timing examples.

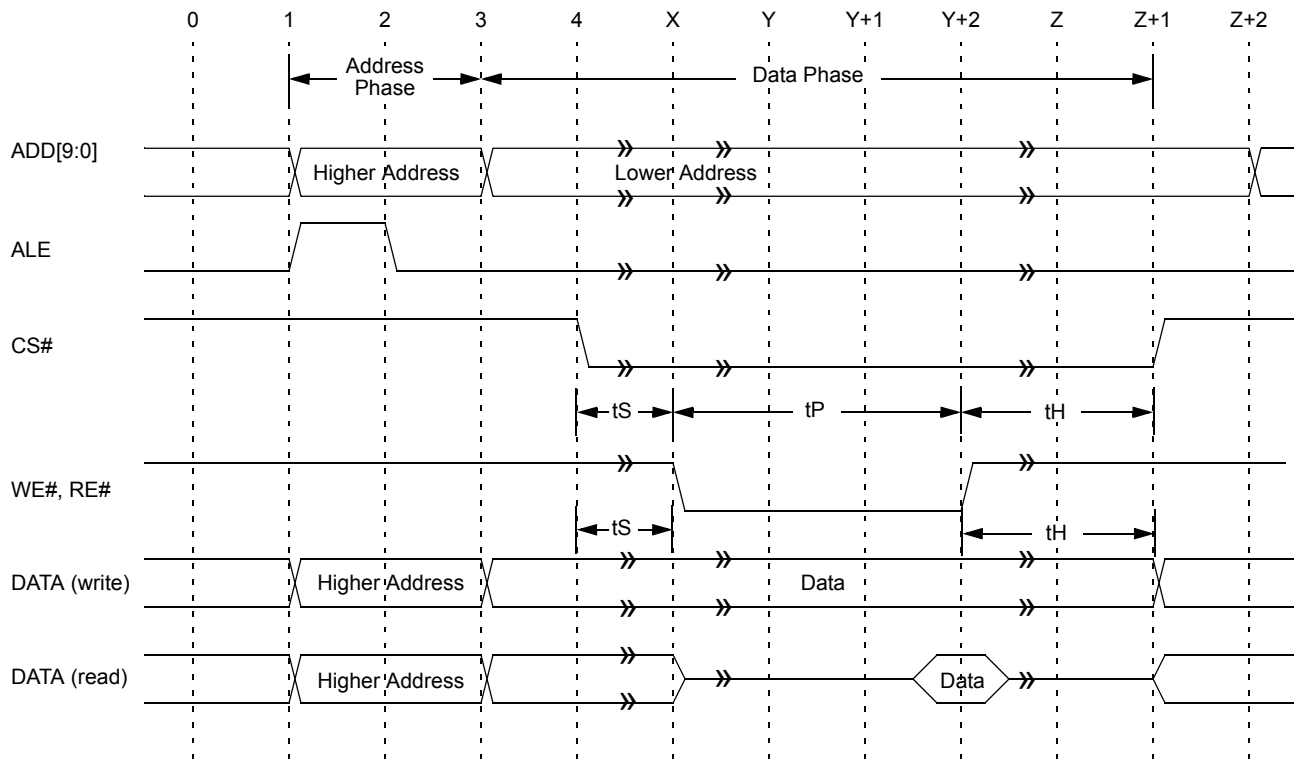


Figure 5-57. NOR Flash Basic Timing

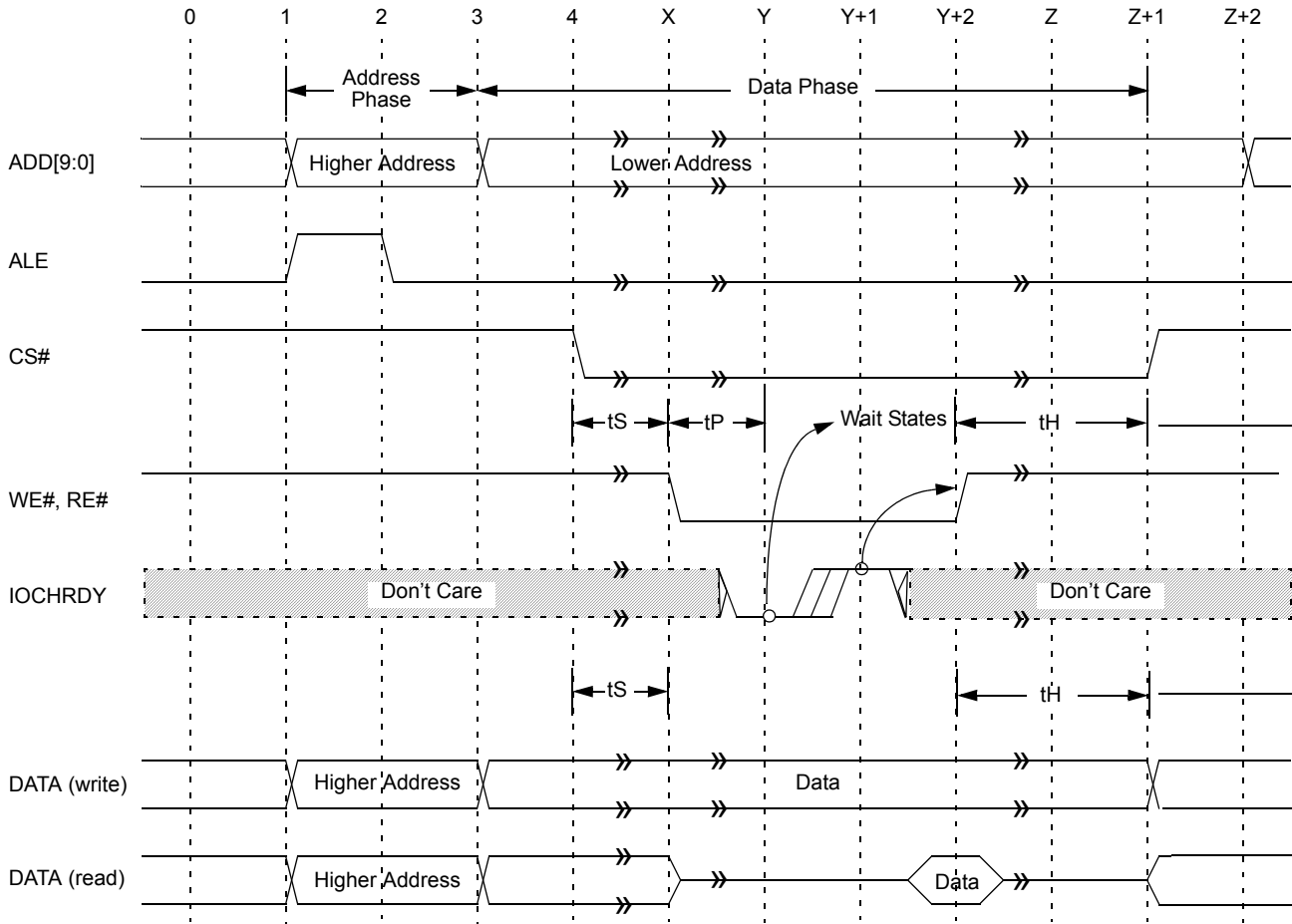


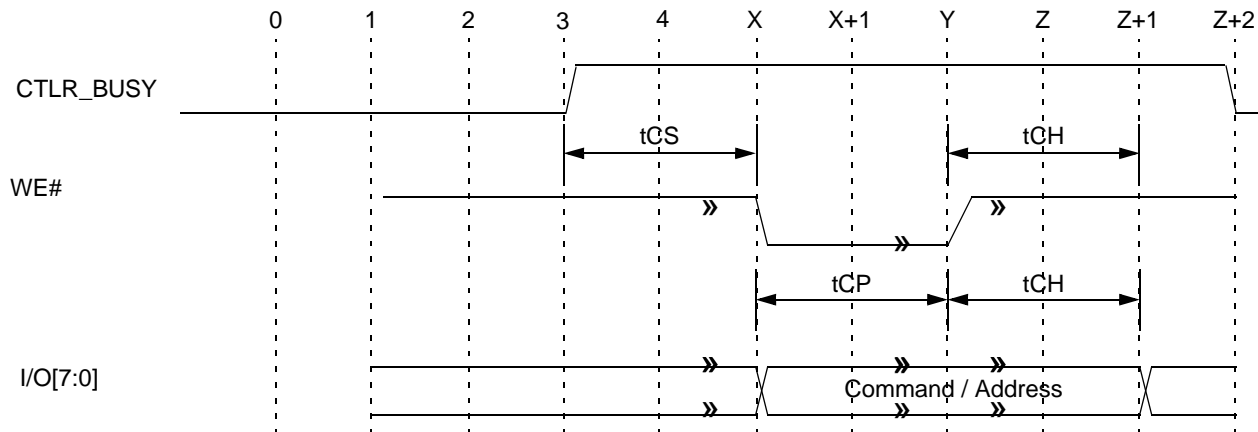
Figure 5-58. NOR Flash with Wait States Timing

### 5.18.3.2 NAND

The NAND Flash interface has three external timings that are controlled by nine timing registers. The timing parameters are described in Table 5-37 and illustrated in Figures 5-59 through 5-61.

**Table 5-37. NAND Flash External Timing Parameters**

Symbol	Description
tCS	Control Setup Time: The setup time from the toggle of the control signals to the falling edge of WE#.
tCP	Control Pulse Width: The WE# active pulse width in the Command/Address phase. Note that the command/address byte is put on the I/O bus at the same time that the WE# is asserted.
tCH	Control Hold Time: The hold time from the rising edge of WE# to the toggle of the control signals. Note that the I/O bus is turned off when the tCH expires.
tWS	Data Write Setup Time: This timing is just for the internal state machine; no external reference point. Can be set to 0 if the setup time is not needed.
tWP	Data Write Pulse Width: The WE# active pulse width in the data write phase. Note that the data byte is put on the I/O bus at the same time that WE# is asserted; no external reference point. Can be set to 0 if the hold time is not needed.
tRP	Data Read Pulse Width: The RE# active pulse width in the data read phase.
tRH	Data Read Hold Time: This timing is just for the internal state machine; no external reference point. Can be set to 0 if the hold time is not needed.



**Note:** CTLR\_BUSY is bit 2 of the NAND Status register (Flash Memory Offset 810h or Flash I/O Offset 06h).

**Figure 5-59. NAND Flash Command/Address Timing**

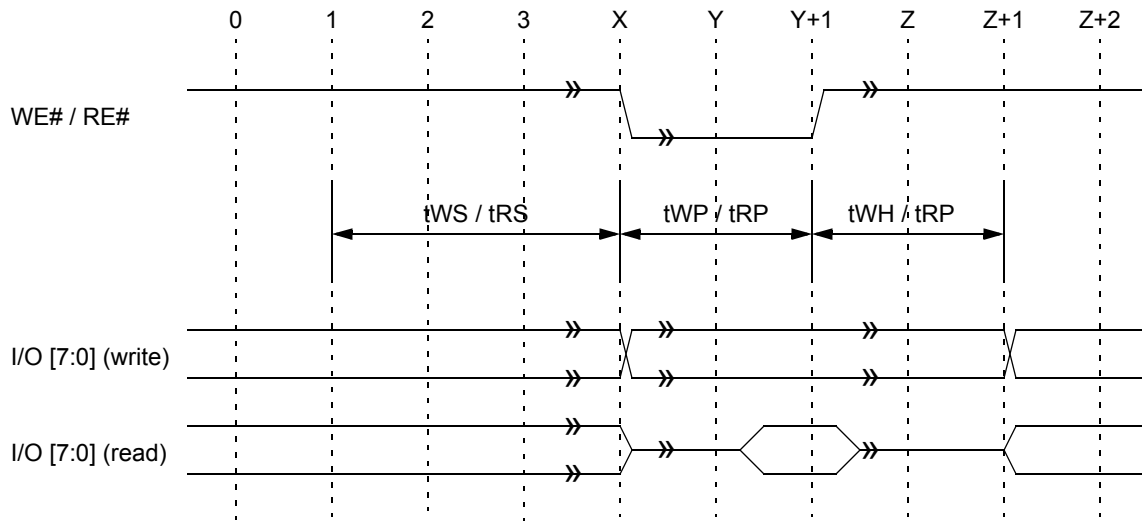


Figure 5-60. NAND Data Timing with No Wait States and No Prefetch (for the first data read)

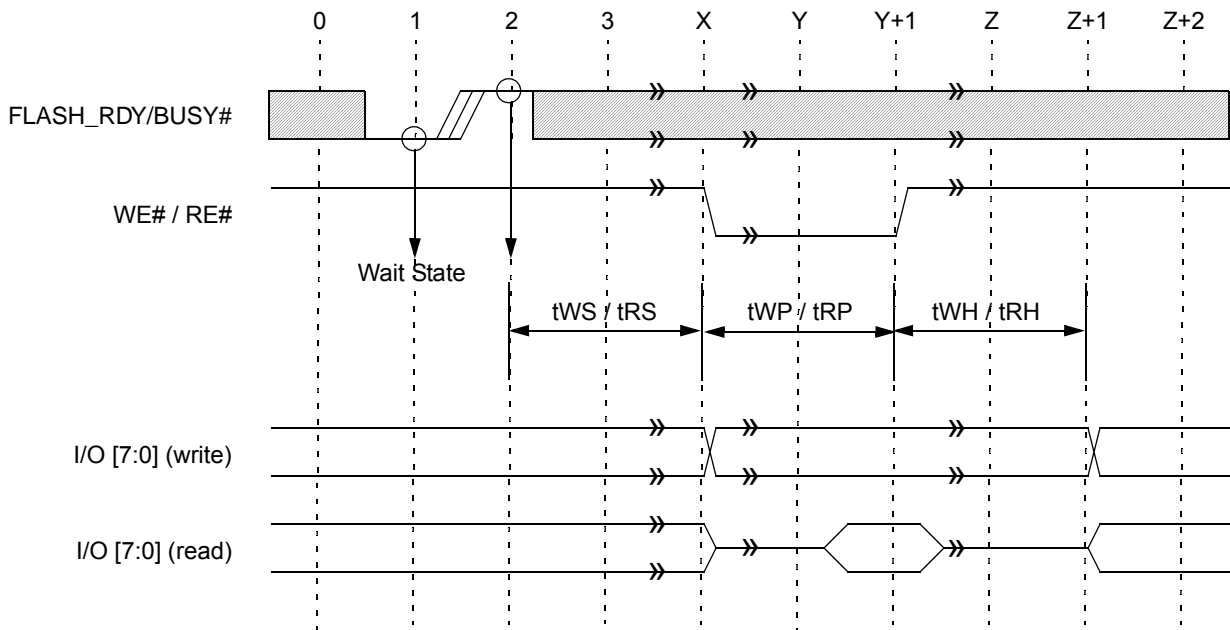


Figure 5-61. NAND Data Timing with Wait States

## 5.19 GeodeLink™ Control Processor

The GeodeLink™ Control Processor (GLCP) functionality is illustrated in Figure 5-62 and is summarized as:

- Serial to GeodeLink conversion to facilitate JTAG accesses to GeodeLink Devices
- Power management support (reset and clock control)
- MSRs

Together with a JTAG controller, the GLCP provides complete visibility of the register state that the chip is in. All registers are accessible via the JTAG interface.

How the JTAG controller interfaces with the GLCP is beyond the scope of this document and is not explained here.

The GLCP also works with the CCU (Clock Control Unit) blocks of other GeodeLink Devices to provide clock control via its relevant MSRs. The GLCP supplies the clock enable signals to all the CCUs, allowing clocks to be shut off if the power management logic generates a Sleep request or if a debug event triggers a clock disable situation.

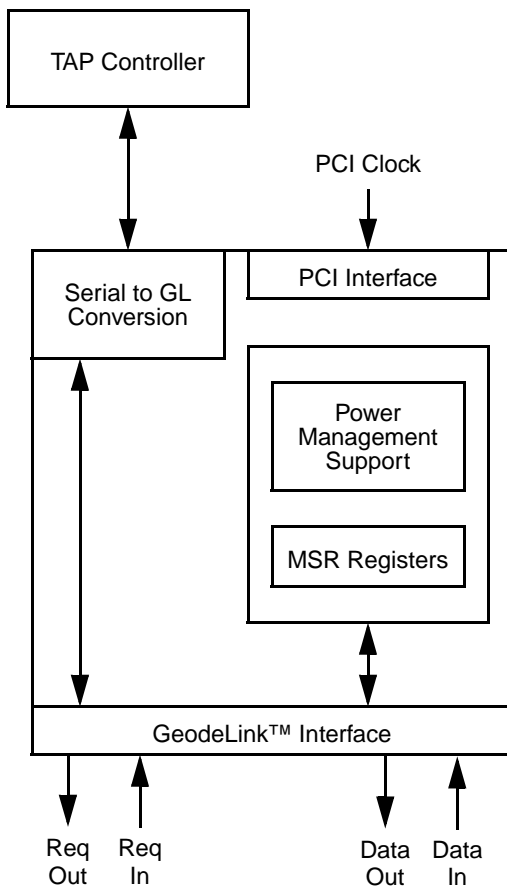


Figure 5-62. GLCP Block Diagram

### 5.19.1 GeodeLink™ Power Management Support

The main power management functions are performed by the Power Management Logic, with the GLCP playing a supporting role. (See Section 5.17 "Power Management Control" on page 172 for a complete understanding of power management.)

#### 5.19.1.1 Soft Reset

This is one of the active high soft reset sources going to the Power Management Logic. It resides in the GLCP\_SYS\_RST register. When active, all circuitry in the Geode CS5535 companion device is reset (including the GLCP\_SYS\_RST register itself).

#### 5.19.1.2 Clock Control

The GLCP provides a mechanism to shut off clocks. The busy signal from a module can control the clock gating in its CCU, however, clocks can also be enabled or disabled by the functional clock enable signals coming from the GLCP. These enable signals are asynchronous to the modules and need to be synchronized in the CCU blocks before being used to enable or disable the functional clocks.

The clocks can be disabled in one or a combination of the three ways below. All the MSRs mentioned can be found in Section 6.18 "Power Management Controller Register Descriptions" on page 494 and Section 6.20 "GeodeLink™ Control Processor Register Descriptions" on page 531.

- 1) The power management circuitry disables the clocks when going into Sleep. The Sleep sequence is started by the assertion of Sleep Request from the Power Management Logic. The GLCP asserts Sleep Request and waits for the assertion of Sleep Acknowledge, which indicates that the clocks should be disabled. There are two ways to do this:

A) If Sleep Acknowledge is asserted and the clock disable delay period has expired, disable the clocks specified in GLCP\_PMCLKDISABLE (MSR 51700009h). Each bit in GLCP\_PMCLKDISABLE corresponds to a CCU, and when set, indicates that the clock going to that CCU should be disabled during a Sleep sequence. The clock disable delay period is specified by the CLK\_DELAY bits in GLCP\_CLK\_DIS\_DELAY (MSR 51700008h), and is enabled by the CLK\_DLY\_EN bit in GLCP\_GLB\_PM (MSR 5170000Bh). It is clocked by the PCI functional clock.

B) If Sleep Acknowledge is asserted and the clock disable delay period is not enabled, check to see if all clocks specified by GLCP\_CLK4ACK (MSR 51700013h) have become inactive. If GLCP\_CLKACTIVE (MSR 51700011h) shows that those clocks are indeed inactive, disable the clocks specified in GLCP\_PMCLKDISABLE (MSR 51700009h).

Sleep Acknowledge is asserted after the clocks have been disabled. The wakeup sequence is triggered by the de-assertion of the Sleep Request, which turns on all the clocks.

- 2) If a debug event in the debug circuitry triggers a clock disable, disable all the clocks specified in GLCP\_CLKDISABLE. Each GLCP\_CLKDISABLE bit corresponds to a CCU, and when set, indicates that the clock going to that CCU should be disabled.
- 3) Each bit in GLCP\_CLKOFF (MSR 51700010h) corresponds to a CCU. When set, the bit indicates that the clock going to that CCU should be disabled. This is the simplest case.

### 5.19.2 GLCP Clocks

The GLCP has multiple clock domains, namely the GeodeLink clock and PCI clock. The GeodeLink clock is the clock source for the MSRs, the serial interface, and the GeodeLink interface. The PCI clock is used in the power management support for the clock disable delay timer. Both the GeodeLink and PCI functional clocks come from primary inputs. All these clocks are handled by a CCU. Even though the PCI clock is always running in functional mode, a CCU is needed to be able to perform reset synchronization and to turn off the internal clock to support TAPSCAN. The CCUs used by the GLCP are the asynchronous versions, since the GLCP outputs asynchronous busy signals.



## 5.20 TAP Controller

The TAP Controller is IEEE 1149.1 compliant. A block diagram of the TAP, boundary scan and Internal scan is shown in Figure 5-63. The JTAG pins TCK, TDI, TDO, TMS, and RESET\_STAND# are directly supported. The TAP is programmable by means of TAP control instructions. The meanings of the various instructions are shown in Table 5-38 on page 194 along with the length of the DR (Data register) that can be accessed once the instruction is entered. All Data registers shift in and out data LSB first. The Instruction register and all Data registers are shift registers, so if more bits are shifted in than the register can hold, only the last bits shifted in - the MSBs - will be used. This can be useful on systems that always shift in a multiple of 8 bits to the Data or Instruction registers. The Instruction register is 24 bits wide and defined in Table 5-39 on page 195.

The TAP Controller can be initialized synchronously or asynchronously. For a synchronous reset, holding TMS high and clocking TCK a minimum of five times will put the TAP state machine into the Test-Logic-Reset state. Asynchronous reset is available too by asserting RESET\_STAND# (TAP Controller Reset) (see Section 4.6 "Reset Considerations" on page 65). From

RESET\_STAND#, the TAP state machine will immediately enter the Test-Logic-Reset state.

The TAP has specific pre-assigned meanings to the bits in the 24-bit IR register. The meanings are summarized in Table 5-39. Note that the bits only affect the chip once the "Update-IR" JTAG state occurs in the JTAG Controller - shifting through these bits will not change the state of internal signals (e.g., test\_mode). The details on JTAG Controller states are covered in the IEEE 1149.1 standard.

### Features

TAP control/access to the following:

- Shift/capture of CCU scan chain
- GLIU access via Request-in, Request-out packets
- TAPSCAN access
- TRI-STATE mode control
- Memory BIST control
- ID code
- Configures component for JTAG bypass mode

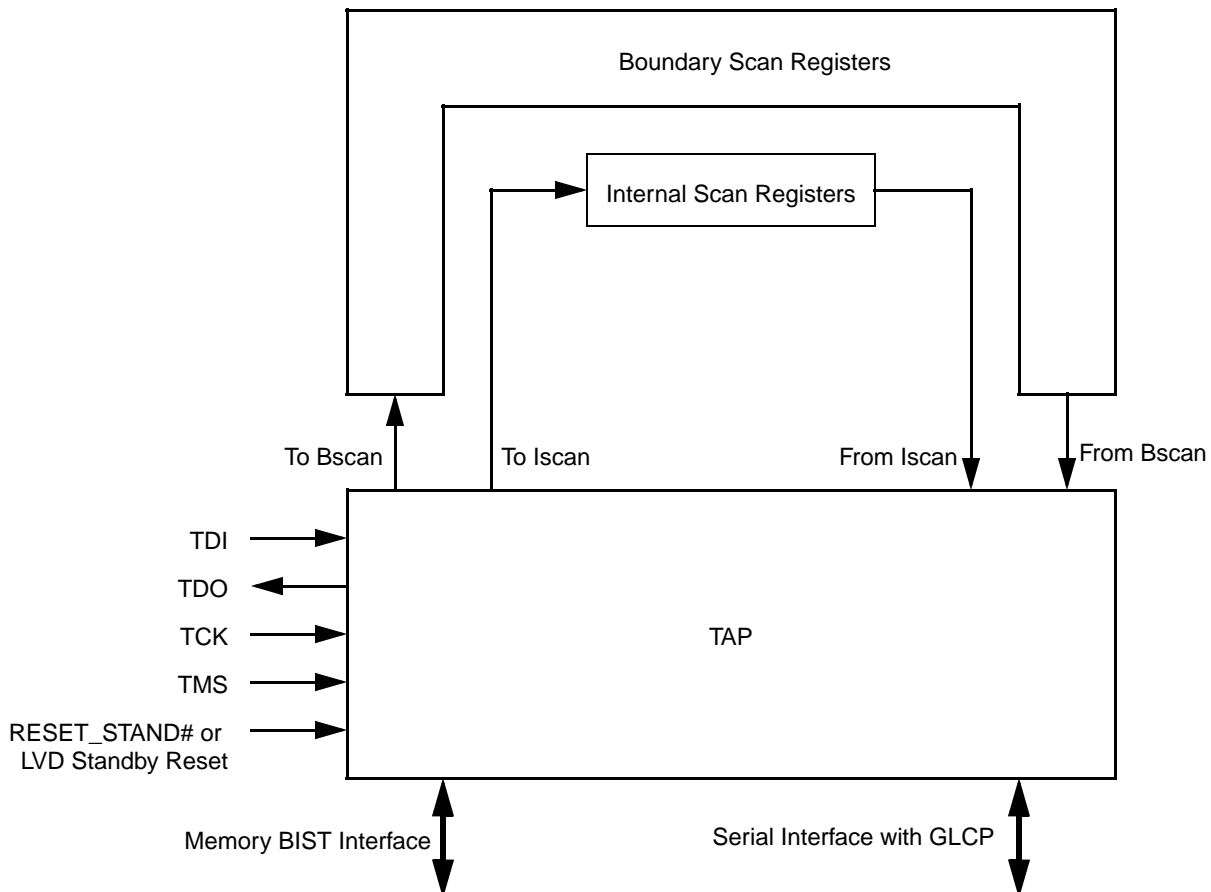


Figure 5-63. TAP Controller, Boundary Scan Block Diagram

Table 5-38. TAP Controller Instructions

Instruction	DR Length	IR Name	Description												
000000h and FFFFE8h	240	EXTEST	<b>Boundary Scan Ring.</b> IEEE 1149.1 specification compliant. (Mapped twice in IR address space.)												
01FFFAh through 1DFFFAh	Variable	TAPSCAN[0:28]	<b>TAP Scan Chain 0 through Chain 28.</b> These are parts of the internal scan chain subdivided according to a common CCU clock.												
01FFFFh through 1DFFFFh	1	TAPFUNC[0:28]	<b>TAP Function Chain 0 through Chain 28.</b> One capture cycle applied to the individual CCU scan chain clocked by the functional clock.												
81FFFAh	70	GL_ADDR	<b>GeodeLink Address.</b> Access GeodeLink request packet and data packet control bits.												
83FFFAh	66	GL_DATA	<b>GeodeLink Data.</b> Access GeodeLink data.												
85FFFAh	4	PADACC	<b>Pad Access.</b> Padtestmode for access to analog and memory BIST signals.												
87FFFAh	24	PROGMISR	<b>Program instruction.</b> This instruction provides direct access to an MSR used for the ROM memory BIST test. Not supported.												
8BFFFAh	70	GL_ADDR_ACT	<b>GeodeLink Address Action.</b> Same data register as GL_ADDR, but no GeodeLink transactions are triggered by the access - only by GLCP debug action.												
8DFFFAh	--	TST_IDDQ	<b>Test IDDQ.</b> Put chip in a mode for running IDDQ tests.												
8FFFAh	8	REVID	<b>Revision ID.</b> The TAP instruction used to access the current revision code (8 bits) for the chip.												
FFFFDFh	1	TRISTATE	<b>TRI-STATE.</b> Put chip into TRI-STATE and comparison mode.												
FFFFDh	21	BISTDR	<b>Parallel RAM BIST.</b> Internal data register (for chip test).												
FFFFEh	32	IDCODE	<b>ID Code.</b> 0FE1101Fh for the Geode CS5535 companion device. <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%;"></td> <td style="width: 25%; text-align: right;">LSB</td> </tr> <tr> <td>ID[31:28]</td> <td>ID27</td> <td>ID[12:11]</td> <td>ID[1:0]</td> </tr> <tr> <td>Version</td> <td>Part Number</td> <td>Manuf. ID</td> <td>1</td> </tr> </table>				LSB	ID[31:28]	ID27	ID[12:11]	ID[1:0]	Version	Part Number	Manuf. ID	1
			LSB												
ID[31:28]	ID27	ID[12:11]	ID[1:0]												
Version	Part Number	Manuf. ID	1												
FFFFFh	1	BYPASS	<b>Bypass.</b> IEEE 1149.1 specification requires all 1s to be bypassed.												

Table 5-39. TAP Controller Instruction Bits

Bit	Name	Description
23	TAPSCAN	<b>TAP Scan.</b> (Also USER[6].) This is a user bit added by AMD. Low indicates that an internal scan chain will be accessed by the TAP.
22:17	USER[5:0]	<b>User Bits 5 through 0.</b> User bits used to identify an internal scan chain or, if bit 23 is high, to access a special internal DR.
16	bistEnable[3]	<b>BIST Enable Bit 3.</b> Works in conjunction with bits [9:7]. See bits [9:7] description for decode.
15:10	RSVD	<b>Reserved.</b> Should always be high.
9:7	bistEnable[2:0]	<p><b>BIST Enable Bits 2 through 0.</b> Works in conjunction with bit 16.</p> <p>0000: bistEn12  0010: bistEn11  0100: bistEn10  0110: bistEn9  0001: bistEn8  0011: bistEn7  0101: bistEn6  0111: bistEn5  1000: bistEn4  1010: bistEn3  1100: bistEn2  1110: bistEn1  1011: bistEn0  1001: Does nothing [normally used for logic BIST]  1101: Parallel scan  1111: Scan through TAP</p>
6	RSVD	<b>Reserved.</b> Should always be high.
5	forceDis	<b>Force Disable.</b> Active low bit that places all output pins in TRI-STATE mode. See TRISTATE for details.
4	selectJtagOut	<b>Select JTAG Output.</b> Active low bit that allows boundary scan cells to control pads.
3	selectJtagIn	<b>Select JTAG Output.</b> Active low bit that allows boundary scan cells to drive data into core logic of chip.
2:0	OP[2:0]	<p><b>Operation Bits 2 through 0.</b> Selects for how the JTAG chains are wired together.</p> <p>000: TDI -&gt; Boundary Scan -&gt; TDO  001: TDI -&gt; Boundary Scan -&gt; Internal Scan -&gt; Device ID -&gt; Bypass -&gt; TDO  010: TDI -&gt; Internal Scan -&gt; TDO  011: TDI -&gt; Boundary Scan -&gt; Internal Scan -&gt; Device ID -&gt; Bypass -&gt; TDO  100: TDI -&gt; Internal Scan -&gt; TDO  101: TDI -&gt; Internal Data Register -&gt; TDO  110: TDI -&gt; Device ID -&gt; TDO  111: TDI -&gt; Bypass -&gt; TDO</p>

### 5.20.1 EXTEST

The EXTEST instruction accesses the boundary scan chain around the chip and controls the pad logic such that the boundary scan data will control the data and enable signals for the pads. IEEE 1149.1 requires that an all-zero instruction access the boundary scan chain.

### 5.20.2 TAPSCAN

These instructions enable JTAG access to the internal scan associated with a particular CCU clock. TCK will then provide the Scan clock to the CCU so that shifting occurs correctly during the Shift-DR state of the TAP. (See Figure 5-64 and Figure 5-65.)

### 5.20.3 TAPFUNC

These instructions connect TDI and TDO to the 1-bit bypass register during DR access. They are useful in that during the Capture-DR state, one functional clock can be applied to the specific CCU clock indicated by the instruction. This mode works well with the FS2 JTAG control software available with Geode CS5535 companion device. The Scan Enable signal to the block will be inactive during this clock.

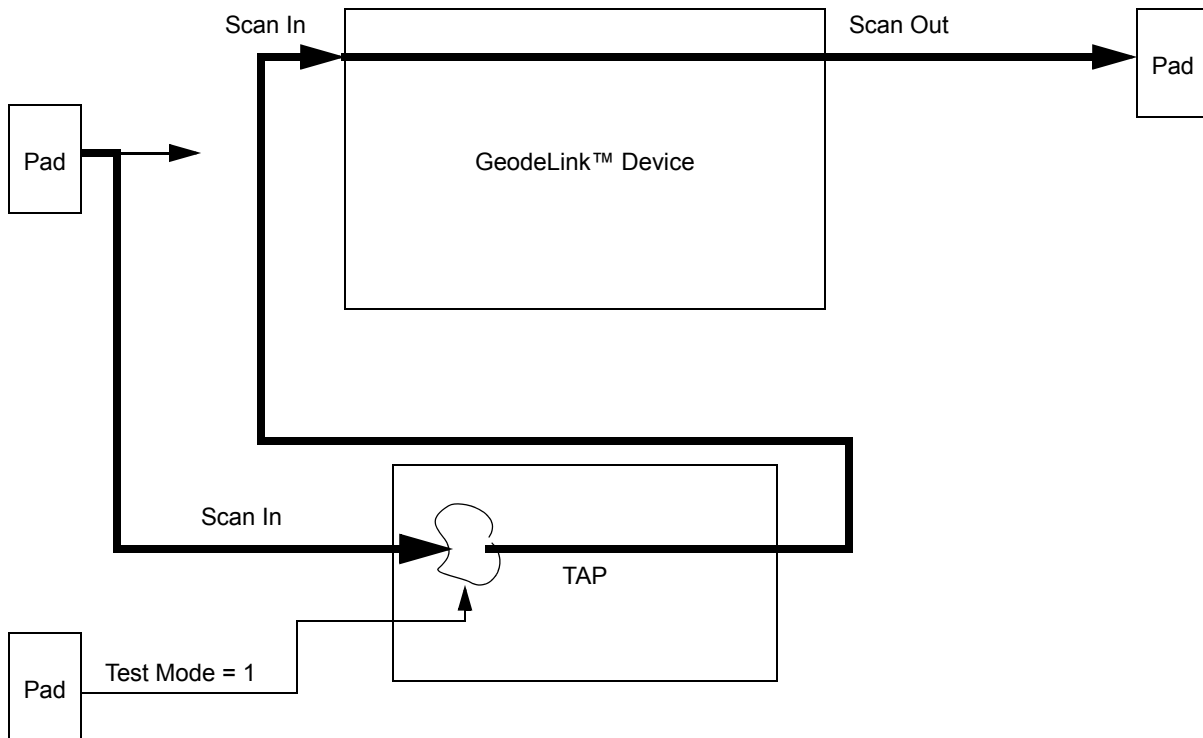


Figure 5-64. TAP Controller and Parallel Scan Mode

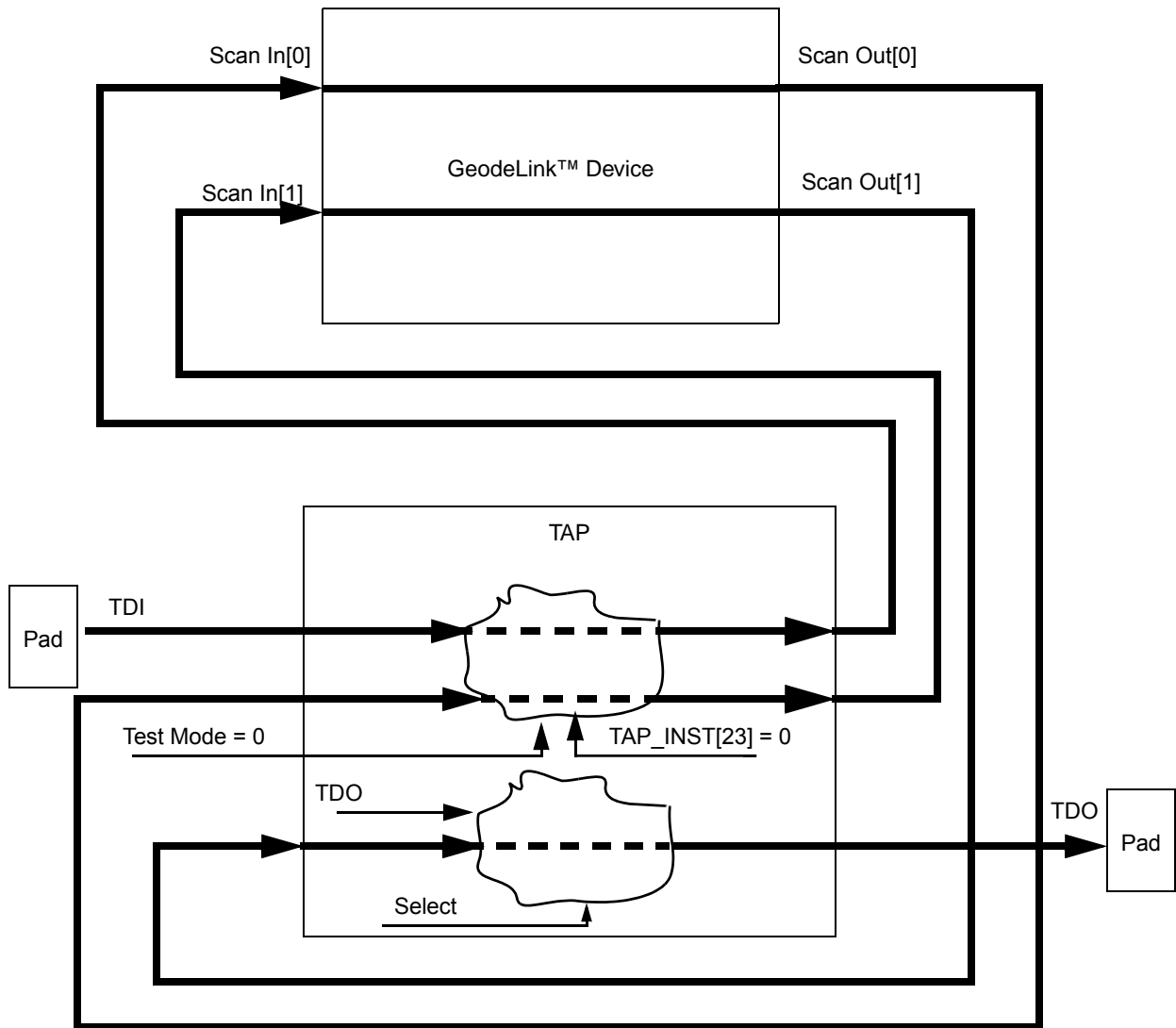


Figure 5-65. TAP Controller and TAPSCAN Mode

#### 5.20.4 GL\_ADDR

This register contains 53 bits for a GeodeLink™ control packet and the 17 bits for a GeodeLink data packet. The 17 bits from the data packet are updated if a GeodeLink read is requested and is available for shifting out. The GL\_DATA description discusses the various conditions under which a valid request packet is posted to the internal GeodeLink. Note that since only one GL\_ADDR request packet can be sequenced in with JTAG, the special “read with byte enable” 2-packet requests that GeodeLink supports cannot be triggered. Of course, 8, 16, 32, and 64-bit reads can still be performed and reads of less than 64-bit sizes will generate the appropriate byte enables at the device. As with GeodeLink traffic, reads of less than 64 bits must be to an aligned address, but the data will return in the GL\_DATA adjusted to 64-bit alignment (i.e., a 16-bit read to address 102h should have address bit 1 set and data will return in bits [31:16] of the 64-bit response). Writes of less than 64 bits must always have 64-bit aligned addresses and should use the byte enables in the data packet (part of the GL\_ADDR data register) to identify which specific bytes are to be written.

#### 5.20.5 GL\_DATA

The data transfer rate in and out of the JTAG port is limited to about 90% of the TCK frequency by the GLCP design. The GLCP is designed for up to 50 MHz TCKs, but typical TCK rates for industry interfaces are about 15 MHz. As such, the GLCP JTAG data rate is 14 Mbits/sec or 1.6 Mbytes/sec. Again, however, industry interface boxes will limit this rate to about 500 kbytes/sec.

GeodeLink requests packets are triggered at these specific moments:

- If GL\_ADDR has been accessed more recently than GL\_ADDR\_ACT and...
  - the TYPE of the request is a read and the Update-DR JTAG state is entered after loading the GL\_ADDR register.
  - the TYPE of the request is a write and the Update-DR JTAG state is entered after loading the GL\_DATA register.
  - the TYPE of the request is a read and the second TCK in the Shift-DR state for shifting out the GL\_DATA register is received and the first two bits shifted in (GL\_DATA DR bits 1 and 0) are non-zero and the first bit shifted out was non-zero.
- If GL\_ADDR\_ACT register has been accessed more recently than GL\_ADDR and...
  - the GLCP debug logic triggers the GeodeLink\_action due to a debug event occurring.

Note that if both MSR accesses from the GLIU and JTAG accesses are interfacing to these registers, the results will be non-deterministic.

#### 5.20.6 PADACC

Provides a test mode whereby USB interface signals or memory BIST signals can be accessed by input/output pads. This access is accomplished by writing to the Auxiliary Test Register.

#### 5.20.7 PROGMISR

This instruction provides direct access to an MSR used for the ROM memory BIST test. At the conclusion of the test, the resulting signature is then checked. A correct test will result in MBIST\_GO being logic-high.

#### 5.20.8 MB\_ADDR\_ACT

This is the same data register as GL\_ADDR, but it disables any GeodeLink transaction from occurring either on this access or a following access to the GL\_DATA register. Only the GLCP debug action that triggers a GeodeLink cycle will cause these bits to be used.

#### 5.20.9 TST\_IDDQ

Places the chip in a mode for running IDDQ tests (i.e., generates an internal signal to disable pull-ups and pull downs). Also the transceiver is powered off.

#### 5.20.10 REVID

The TAP instruction used to access the current 8-bit revision code of the chip.

#### 5.20.11 TRISTATE

This instruction will TRI-STATE all of the tri-statable primary outputs. The DR accessed is the BYPASS register.

#### 5.20.12 BISTDR

Can be used to run all memory BIST controllers in parallel.

#### 5.20.13 IDCODE

This instruction accesses the 32-bit IDCODE register during DR access.

#### 5.20.14 BYPASS

In the IEEE 1149.1 specification, shifting all 1s into the IR must connect the 1-bit BYPASS register. The register has no function except as a storage flip-flop. This instruction can also allow relatively easy connection of multiple GLCP JTAG interface chips. On a board with two GLCP chips, TMS and TCK of each chip should be wired together and TDO of one chip should connect to TDI of the other chip.

**Note:** In parallel scan mode, “input” pads provide data into the boundary scan cells (the boundary scan cells provide data into the core). “Cowrie” pads will behave as dictated by the internal core flops that normally control the pad; the output data and enable state will be latched into the boundary scan cells. “Cheroot” pads will drive out data as dictated by the internal core flop associated with the pad.

# Register Descriptions

# 6

This chapter provides detailed information regarding the registers of the AMD Geode™ CS5535 companion device. The register descriptions are documented at the module-level and briefly summarized below.

## GeodeLink™ Interface Unit (GLIU)

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- P2D Descriptor MSRs: Accessed via RDMSR and WRMSR instructions. (Memory base descriptor.)
- GLIU Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- IOD Descriptor MSRs: Accessed via RDMSR and WRMSR instructions. (I/O base descriptor.)

## GeodeLink PCI South Bridge (GLPCI\_SB)

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- GLPCI\_SB Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- PCI Configuration Registers: Index accessed via PCI configuration cycle.

## Audio Codec 97 Controller (ACC)

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- ACC Native Registers: Accessed as I/O offsets from a GLIU IOD descriptor.

## ATA5 Controller (ATAC)

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- ATAC Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- ATAC Native Registers: Accessed as I/O offsets from a GLIU IOD descriptor.

## Universal Serial Bus Controllers (USBC1, USBC2)

- Standard GeodeLink Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- USB Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- USB Embedded PCI Configuration Registers: Accessed via RDMSR and WRMSR instructions. Also requires GLIU P2D descriptor.

- Host Controller Native Registers: Accessed via a base address register at USB PCI Index 10h as memory offsets.

## Diverse Integration Logic (DIVIL)

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- DIVIL Specific MSRs: Accessed via RDMSR and WRMSR instructions.

## Floppy Port

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- Floppy Port Specific MSRs: Accessed via RDMSR and WRMSR instructions.

## Programmable Interval Timer (PIT)

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- PIT Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- PIT Native Registers: Accessed as I/O addresses.

## Programmable Interrupt Controller (PIC)

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- PIC Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- PIC Native Registers: Accessed as I/O addresses.

## Keyboard Emulation Logic (KEL)

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- KEL Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- KEL Native Registers: Accessed via a base address register, MSR\_LBAR\_KEL1 (MSR 51400009h) and/or MSR\_LBAR\_KEL2 (MSR 5140000Ah), as memory offsets.

## System Management Bus (SMB)

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- SMB Native Registers: Accessed via a base address register, MSR\_LBAR\_SMB (MSR 5140000Bh), as I/O offsets.

**Universal Asynchronous Receiver-Transmitter (UART)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- UART/IR Controller Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- UART/IR Controller Native Registers: Accessed via Banks 0 through 7 as I/O offsets. See MSR\_LEG\_IO (MSR 51400014h) bits [22:20] and bits [18:16] for setting base address.

**Direct Memory Access (DMA)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- DMA Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- DMA Native Registers: Accessed as I/O Addresses.

**Low Pin Count (LPC)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- LPC Specific MSRs: Accessed via RDMSR and WRMSR instructions.

**Real-Time Clock (RTC)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- RTC Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- RTC Native Registers: Accessed as I/O addresses.

**General Purpose Input Output (GPIO)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- GPIO Native Registers: Accessed via a base address register, MSR\_LBAR\_GPIO (MSR 5140000Ch), as I/O offsets.
  - GPIO Low/High Bank Feature Bit Registers
  - GPIO Input Conditioning Function Registers
  - GPIO Interrupt and PME Registers

**Multi-Function General Purpose Timer (MFGPT)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- MFGPT Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- MFGPT Native Registers: Accessed via a base address register, MSR\_LBAR\_MFGPT (MSR 5140000Dh), as I/O offsets.

**Power Management Controller (PMC)**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- PMC Specific MSRs: Accessed via RDMSR and WRMSR instructions.

- ACPI Registers: Accessed via a base address register, MSR\_LBAR\_ACPI (MSR 5140000Eh), as I/O offsets.
- PM Support Registers: Accessed via a base address register, MSR\_LBAR\_PMS (MSR 5140000Fh), as I/O offsets.

**Flash Controller**

- Standard GeodeLink Device MSRs: Accessed via RDMSR and WRMSR instructions. (Shared with DIVIL.)
- Flash Controller Specific MSRs: Accessed via RDMSR and WRMSR instructions.
- Flash Controller Native Registers: Accessed via a base address register as either memory or I/O offsets:
  - MSR\_LBAR\_FLSH0 (MSR 51400010h) for use with FLASH\_CS0#.
  - MSR\_LBAR\_FLSH1 (MSR 51400011h) for use with FLASH\_CS1#.
  - MSR\_LBAR\_FLSH2 (MSR 51400012h) for use with FLASH\_CS2#.
  - MSR\_LBAR\_FLSH3 (MSR 51400013h) for use with FLASH\_CS3#.

**GeodeLink Control Processor (GLCP)**

- Standard GeodeLink™ Device (GLD) MSRs: Accessed via RDMSR and WRMSR instructions.
- GLCP Specific MSRs: Accessed via RDMSR and WRMSR instructions.

Note that MSRs for the Floppy Port, PIT, PIC, KEL, SMB, UART, DMA, LPC, RTC, GPIO, MFGPT, and Flash Controller modules are part of the DIVIL (i.e., MSR 51400000h-514000FFh). Hence, the Standard GeodeLink Device MSRs (MSR 51400000h-51400007h) are documented in the DIVIL register description and the device Specific MSRs are documented in their appropriate register description chapter.

The tables in this chapter use the following abbreviations:

Type	Description
R/W	Read/Write.
R	Read from a specific address returns the value of a specific register.
W	Write to the same address is to a different register.
W	Write.
RO	Read Only.
WO	Write Only.
R/W1C	Read/Write 1 to clear. Writing 1 to a bit clears it to 0. Writing 0 has no effect.



## 6.1 GeodeLink™ Interface Unit Register Descriptions

The GeodeLink™ Interface Unit (GLIU) registers are Model Specific Registers (MSRs) and are accessed through the RDMSR and WRMSR instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The MSRs are split into the following groups:

- Standard GeodeLink Device (GLD) MSRs
- P2D Descriptor MSRs

- GLIU Specific MSRs
- IOD Descriptor MSRs

Tables 6-1 through 6-4 are GLIU register summary tables that include reset values and page references where the bit descriptions are provided.

Reserved (RSVD) fields do not have any meaningful storage elements. They always return 0.

**Table 6-1. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51010000h	RO	GLD Capabilities MSR (GLIU_GLD_MSR_CAP)	00000000_000010xxh	Page 204
51010001h	R/W	GLD Master Configuration MSR (GLIU_GLD_MSR_CONFIG)	00000000_00000004h	Page 204
51010002h	R/W	GLD SMI MSR (GLIU_GLD_MSR_SMI)	00000000_00000001h	Page 205
51010003h	R/W	GLD Error MSR (GLIU_GLD_MSR_ERROR)	00000000_00000001h	Page 206
51010004h	R/W	GLD Power Management MSR (GLIU_GLD_MSR_PM)	00000000_00000000h	Page 208
51010005h	R/W	GLD Diagnostic MSR (GLIU_GLD_MSR_DIAG)	00000000_00000000h	Page 208
51010006h-5101000Fh	R/W	GLIU Reserved MSRs (GLD_MSRS_RSVD)	00000000_00000000h	---

**Table 6-2. P2D Descriptor MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51010020h	R/W	P2D Base Mask Descriptor 0 (GLIU_P2D_BM0)	000000FF_FFF00000h	Page 209
51010021h	R/W	P2D Base Mask Descriptor 1 (GLIU_P2D_BM1)	000000FF_FFF00000h	Page 209
51010022h	R/W	P2D Base Mask Descriptor 2 (GLIU_P2D_BM2)	000000FF_FFF00000h	Page 209
51010023h	R/W	P2D Base Mask KEL Descriptor 0 (GLIU_P2D_BMK0)	000000FF_FFF00000h	Page 210
51010024h	R/W	P2D Base Mask KEL Descriptor 1 (GLIU_P2D_BMK1)	000000FF_FFF00000h	Page 210
51010025h-5101003Fh	R/W	P2D Reserved Descriptors (P2D_RSVD)	00000000_00000000h	---

Table 6-3. GLIU Specific MSRs Summary

MSR Address	Type	Register Name	Reset Value	Reference
51010080h	R/W	Coherency (GLIU_COH)	00000000_00000000h	Page 211
51010081h	R/W	Port Active Enable (GLIU_PAE)	00000000_0000FFFFh	Page 211
51010082h	R/W	Arbitration (GLIU_ARB)	00000000_00000000h	Page 212
51010083h	R/W	Asynchronous SMI (GLIU_ASMI)	00000000_00000000h	Page 212
51010084h	R/W	Asynchronous Error (GLIU_AERR)	00000000_00000000h	Page 213
51010085h	R/W	Debug (GLIU_DEBUG)	00000000_00000004h	Page 214
51010086h	RO	Physical Capabilities (GLIU_PHY_CAP)	327920A0_80000003h	Page 215
51010087h	RO	N Outstanding Response (GLIU_NOUT_RESP)	00000000_00000000h	Page 216
51010088h	RO	Number of Outstanding Write Data (GLIU_NOUT_WDATA)	00000000_00000000h	Page 216
51010089h-5101008Ah	R/W	Reserved (RSVD)	00000000_00000000h	---
5101008Bh	RO	WHO AM I (GLIU_WHOAMI)	Configuration Dependent	Page 216
5101008Ch	R/W	Slave Disable (GLIU_SLV_DIS)	00000000_00000000h	Page 217
5101008Dh-5101008Fh	R/W	Reserved (RSVD)	00000000_00000000h	---
510100A0h	WO	Descriptor Statistic Counter 0 (GLIU_STATISTIC_CNT0)	00000000_00000000h	Page 218
510100A1h	R/W	Descriptor Statistic Mask 0 (GLIU_STATISTIC_MASK0)	00000000_00000000h	Page 219
510100A2h	R/W	Descriptor Statistic Action 0 (GLIU_STATISTIC_ACTION0)	00000000_00000000h	Page 220
510100A3h	R/W	Reserved (RSVD)	00000000_00000000h	---
510100A4h	WO	Descriptor Statistic Counter 1 (GLIU_STATISTIC_CNT1)	00000000_00000000h	Page 218
510100A5h	R/W	Descriptor Statistic Mask 1 (GLIU_STATISTIC_MASK1)	00000000_00000000h	Page 219
510100A6h	R/W	Descriptor Statistic Action 1 (GLIU_STATISTIC_ACTION1)	00000000_00000000h	Page 220
510100A7h	R/W	Reserved (RSVD)	00000000_00000000h	---
510100A8h	WO	Descriptor Statistic Counter 2 (GLIU_STATISTIC_CNT2)	00000000_00000000h	Page 218
510100A9h	R/W	Descriptor Statistic Mask 2 (GLIU_STATISTIC_MASK2)	00000000_00000000h	Page 219
510100AAh	R/W	Descriptor Statistic Action 2 (GLIU_STATISTIC_ACTION2)	00000000_00000000h	Page 220
510100ABh-510100BFh	R/W	Reserved (RSVD)	00000000_00000000h	---
510100C0h	R/W	Request Compare Value (GLIU_RQ_COMP_VAL)	001FFFFFF_FFFFFFFFh	Page 221
510100C1h	R/W	Request Compare Mask (GLIU_RQ_COMP_MASK)	00000000_00000000h	Page 222

**Table 6-3. GLIU Specific MSRs Summary (Continued)**

MSR Address	Type	Register Name	Reset Value	Reference
510100C2h-510100CFh	R/W	Reserved (RSVD)	00000000_00000000h	---
510100D0h	R/W	Data Compare Value Low (GLIU_DA_COMP_VAL_LO)	00001FFF_FFFFFFFFh	Page 222
510100D1h	R/W	Data Compare Value High (GLIU_DA_COMP_VAL_HI)	0000000F_FFFFFFFFh	Page 223
510100D2h	R/W	Data Compare Mask Low (GLIU_DA_COMP_MASK_LO)	00000000_00000000h	Page 223
510100D3h	R/W	Data Compare Mask High (GLIU_DA_COMP_MASK_HI)	00000000_00000000h	Page 224
510100D4h-510100DFh	R/W	Reserved (RSVD)	00000000_00000000h	---

**Table 6-4. IOD Descriptor MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
510100E0h	R/W	IOD Base Mask 0 (GLIU_IOD_BM0); Reserved for ATA-5; Defaults to 1F <sub>x16</sub> .	60000000_1F0FFFF0h	Page 225
510100E1h	R/W	IOD Base Mask 1 (GLIU_IOD_BM1)	000000FF_FFF00000h	Page 225
510100E2h	R/W	IOD Base Mask 2 (GLIU_IOD_BM2)	000000FF_FFF00000h	Page 225
510100E3h	R/W	IOD Base Mask 3 (GLIU_IOD_BM3)	000000FF_FFF00000h	Page 225
510100E4h	R/W	IOD Base Mask 4 (GLIU_IOD_BM4)	000000FF_FFF00000h	Page 225
510100E5h	R/W	IOD Base Mask 5 (GLIU_IOD_BM5)	000000FF_FFF00000h	Page 225
510100E6h	R/W	IOD Base Mask 6 (GLIU_IOD_BM6)	000000FF_FFF00000h	Page 225
510100E7h	R/W	IOD Base Mask 7 (GLIU_IOD_BM7)	000000FF_FFF00000h	Page 225
510100E8h	R/W	IOD Base Mask 8 (GLIU_IOD_BM8)	000000FF_FFF00000h	Page 225
510100E9h	R/W	IOD Base Mask 9 (GLIU_IOD_BM9)	000000FF_FFF00000h	Page 225
510100EAh	R/W	IOD Swiss Cheese 0 (GLIU_IOD_SC0)	60000000_403003F0h	Page 226
510100EAh	R/W	IOD Swiss Cheese 0 (GLIU_IOD_SC0); Reserved for ATA-5; Defaults to 3F <sub>616</sub> .	60000000_403003F0h	Page 226
510100EBh	R/W	IOD Swiss Cheese 1 (GLIU_IOD_SC1)	00000000_00000000h	Page 226
510100ECh	R/W	IOD Swiss Cheese 2 (GLIU_IOD_SC2)	00000000_00000000h	Page 226
510100EDh	R/W	IOD Swiss Cheese 3 (GLIU_IOD_SC3)	00000000_00000000h	Page 226
510100EEh	R/W	IOD Swiss Cheese 4 (GLIU_IOD_SC4)	00000000_00000000h	Page 226
510100EFh	R/W	IOD Swiss Cheese 5 (GLIU_IOD_SC5)	00000000_00000000h	Page 226
510100F0h	R/W	IOD Swiss Cheese 6 (GLIU_IOD_SC6)	00000000_00000000h	Page 226
510100F1h	R/W	IOD Swiss Cheese 7 (GLIU_IOD_SC7)	00000000_00000000h	Page 226
510100F2h-510100FFh	R/W	Reserved (RSVD)	00000000_00000000h	---

## 6.1.1 Standard GeodeLink™ Device (GLD) MSRs

### 6.1.1.1 GLD Capabilities MSR (GLIU\_GLD\_MSR\_CAP)

MSR Address 51010000h  
 Type RO  
 Reset Value 00000000\_000010xxh

**GLIU\_GLD\_MSR\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID																REV_ID							

**GLIU\_GLD\_MSR\_CAP Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads return 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (0010h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> for value.

### 6.1.1.2 GLD Master Configuration MSR (GLIU\_GLD\_MSR\_CONFIG)

MSR Address 51010001h  
 Type R/W  
 Reset Value 00000000\_00000004h

**GLIU\_GLD\_MSR\_CONFIG Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												SUBP			

**GLD\_MSR\_CONFIG Bit Descriptions**

Bit	Name	Description
63:3	RSVD	<b>Reserved.</b> Write as read.
2:0	SUBP	<b>Subtractive Port.</b> For all negative decode requests. 000: Port 0 (GLIU)                      100: Port 4 (DD) 001: Port 1 (GLPCI_SB)                101: Port 5 (ACC) 010: Port 2 (USBC2)                    110: Port 6 (USBC1) 011: Port 3 (ATAC)                     111: Port 7 (GLCP) <b>Note:</b> The reset value of this register should not be changed.

**6.1.1.3 GLD SMI MSR (GLIU\_GLD\_MSR\_SMI)**

MSR Address 51010002h  
 Type R/W  
 Reset Value 00000000\_00000001h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 0. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further SMI and ASMI generation details.)

**GLIU\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32					
RSVD																																	STATCNT2_ASMI_FLAG	STATCNT1_ASMI_FLAG	STATCNT0_ASMI_FLAG	SSMI_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
RSVD																																	STATCNT2_ASMI_EN	STATCNT1_ASMI_EN	STATCNT0_ASMI_EN	SSMI_EN

**GLIU\_GLD\_MSR\_SMI Bit Descriptions**

Bit	Name	Description
63:36	RSVD	<b>Reserved.</b> Write as read.
35	STATCNT2_ASMI_FLAG	<b>Statistic Counter 2 ASMI Flag.</b> If high, records that an ASMI was generated due to a Statistic Counter 2 (MSR 510100A8h) event. Write 1 to clear; writing 0 has no effect. STATCNT2_ASMI_EN (bit 3) must be low to generate ASMI and set flag.
34	STATCNT1_ASMI_FLAG	<b>Statistic Counter 1 ASMI Flag.</b> If high, records that an ASMI was generated due to a Statistic Counter 1 (MSR 510100A4h) event. Write 1 to clear; writing 0 has no effect. STATCNT1_ASMI_EN (bit 2) must be low to generate ASMI and set flag.
33	STATCNT0_ASMI_FLAG	<b>Statistic Counter 0 SMI Flag.</b> If high, records that an ASMI was generated due to a Statistic Counter 0 (MSR 510100A0h) event. Write 1 to clear; writing 0 has no effect. STATCNT0_ASMI_EN (bit 1) must be low to generate ASMI and set flag.
32	SSMI_FLAG	<b>SSMI Flag.</b> If high, records that an SSMI was generated due to a received event. Event sources are: <ul style="list-style-type: none"> <li>Illegal request type to GLIU (Port 0), meaning anything other than MSR read/write, debug request, and null.</li> <li>A self-referencing packet (i.e., a packet sent to the GLIU that finds its destination port is the source port).</li> <li>The destination of the packet is to a port where the GLIU slave for that port has been disabled.</li> <li>Trap on a descriptor with device port set to 0. This is the typical operational use of this bit. The data returned with such a trap is the value 0.</li> </ul> Write 1 to clear; writing 0 has no effect. SSMI_EN (bit 0) must be low to generate SSMI and set flag.
31:4	RSVD	<b>Reserved.</b> Write as read.

## GLIU\_GLD\_MSR\_SMI Bit Descriptions (Continued)

Bit	Name	Description
3	STATCNT2_ASMI_EN	<b>Statistic Counter 2 ASMI Enable.</b> Write 0 to enable STATCNT2_ASMI_FLAG (bit 35) and to allow a Statistic Counter 2 (MSR 510100A8h) event to generate an ASMI.
2	STATCNT1_ASMI_EN	<b>Statistic Counter 1 ASMI Enable.</b> Write 0 to enable STATCNT1_ASMI_FLAG (bit 34) and to allow a Statistic Counter 1 (MSR 510100A4h) event to generate an ASMI.
1	STATCNT0_ASMI_EN	<b>Statistic Counter 0 ASMI Enable.</b> Write 0 to enable STATCNT0_ASMI_FLAG (bit 33) and to allow a Statistic Counter 0 (MSR 510100A0h) event to generate an ASMI.
0	SSMI_EN	<b>SSMI Enable.</b> Write 0 to enable SSMI_FLAG (bit 32) and to allow a received SSMI event to generate an SSMI. (See bit 32 description for SSMI event sources.)

## 6.1.1.4 GLD Error MSR (GLIU\_GLD\_MSR\_ERROR)

MSR Address 51010003h  
 Type R/W  
 Reset Value 00000000\_00000001h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 0. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.4 "MSR Address 3: Error Control" on page 78 for further details.)

## GLIU\_GLD\_MSR\_ERROR Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32														
RSVD																				DACMP_ERR_FLAG	RSVD				RQCOMP_ERR_FLAG	RSVD	STATCNT2_ERR_FLAG	STATCNT1_ERR_FLAG	STATCNT0_ERR_FLAG	SSMI_ERR_FLAG	UNEXP_ADD_ERR_FLAG	UNEXP_TYPE_ERR_FLAG													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
RSVD																				DACMP_ERR_EN	RSVD				RQCOMP_ERR_EN	RSVD	STATCNT2_ERR_EN	STATCNT1_ERR_EN	STATCNT0_ERR_EN	SSMI_ERR_EN	UNEXP_ADD_ERR_EN	UNEXP_TYPE_ERR_EN													

## GLIU\_GLD\_MSR\_ERROR Bit Descriptions

Bit	Name	Description
63:44	RSVD	<b>Reserved.</b> Write as read.
43	DACOMP_ERR_FLAG	<b>Data Comparator Error Flag.</b> If high, records that an ERR was generated due to a Data Comparator (DA_COMP_VAL_LO / DA_COMP_VAL_HI, MSR 510100D0h / 510100D1h) event. Write 1 to clear; writing 0 has no effect. DACOMP_ERR_EN (bit 11) must be low to generate ERR and set flag.
42:40	RSVD	<b>Reserved.</b> Write as read.

## GLIU\_GLD\_MSR\_ERROR Bit Descriptions (Continued)

Bit	Name	Description
39	RQCOMP_ERR_FLAG	<b>Request Comparator Error Flag.</b> If high, records that an ERR was generated due to a Request Comparator 0 (RQ_COMP_VAL, MSR 510100C0h) event. Write 1 to clear; writing 0 has no effect. RQCOMP_ERR_EN (bit 7) must be low to generate ERR and set flag.
38	RSVD	<b>Reserved.</b> Write as read.
37	STATCNT2_ERR_FLAG	<b>Statistic Counter 2 Error Flag.</b> If high, records that an ERR was generated due to a Statistic Counter 2 (MSR 510100A8h) event. Write 1 to clear; writing 0 has no effect. STATCNT2_ERR_EN (bit 5) must be low to generate ERR and set flag.
36	STATCNT1_ERR_FLAG	<b>Statistic Counter 1 Error Flag.</b> If high, records that an ERR was generated due to a Statistic Counter 1 (MSR 510100A4h) event. Write 1 to clear; writing 0 has no effect. STATCNT2_ERR_EN (bit 4) must be low to generate ERR and set flag.
35	STATCNT0_ERR_FLAG	<b>Statistic Counter 0 Error Flag.</b> If high, records that an ERR was generated due to a Statistic Counter 0 (MSR 510100A0h) event. Write 1 to clear; writing 0 has no effect. STATCNT0_ERR_EN (bit 3) must be low to generate ERR and set flag.
34	SSMI_ERR_FLAG	<b>SSMI Error Flag.</b> If high, records that an ERR was generated due an unhandled SSMI (synchronous error). Write 1 to clear; writing 0 has no effect. SSMI_ERR_EN (bit 2) must be low to generate ERR and set flag. (Note 1)
33	UNEXP_ADD_ERR_FLAG	<b>Unexpected Address Error Flag.</b> If high, records that an ERR was generated due an unexpected address (synchronous error). Write 1 to clear; writing 0 has no effect. UNEXP_ADD_ERR_EN (bit 1) must be low to generate ERR and set flag. (Note 1)
32	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Error Flag.</b> If high, records that an ERR was generated due an unexpected type (synchronous error). Write 1 to clear; writing 0 has no effect. UNEXP_TYPE_ERR_EN (bit 0) must be low to generate ERR and set flag. (Note 1)
31:12	RSVD	<b>Reserved.</b> Write as read.
11	DACOMP_ERR_EN	<b>Data Comparator Error Enable.</b> Write 0 to enable DACOMP_ERR_FLAG (bit 43) and to allow a Data Comparator (DA_COMP_VAL_LO / DA_COMP_VAL_HI, MSR 510100D0h / 510100D1h) event to generate an ERR and set flag.
10:8	RSVD	<b>Reserved.</b> Write as read.
7	RQCOMP_ERR_EN	<b>Request Comparator Error Enable.</b> Write 0 to enable RQCOMP_ERR_FLAG (bit 39) and to allow a Request Comparator (RQ_COMP_VAL, MSR 510100C0h) event to generate an ERR.
6	RSVD	<b>Reserved.</b> Write as read.
5	STATCNT2_ERR_EN	<b>Statistic Counter 2 Error Enable.</b> Write 0 to enable STATCNT2_ERR_FLAG (bit 37) and to allow a Statistic Counter 2 (MSR 510100A8h) event to generate an ERR.
4	STATCNT1_ERR_EN	<b>Statistic Counter 1 Error Enable.</b> Write 0 to enable STATCNT1_ERR_FLAG (bit 36) and to allow a Statistic Counter 1 (MSR 510100A4h) event to generate an ERR.
3	STATCNT0_ERR_EN	<b>Statistic Counter 0 Error Enable.</b> Write 0 to enable STATCNT0_ERR_FLAG (bit 35) and to allow a Statistic Counter 0 (MSR 510100A0h) event to generate an ERR.
2	SSMI_ERR_EN	<b>SSMI Error Enable.</b> Write 0 to enable SSMI_ERR_FLAG (bit 34) and to allow the unhandled SSMI (synchronous error) event to generate an ERR.
1	UNEXP_ADD_ERR_EN	<b>Unexpected Address Error Enable.</b> Write 0 to enable UNEXP_ADD_ERR_FLAG (bit 33) and to allow the unexpected address (synchronous error) event to generate an ERR.
0	UNEXP_TYPE_ERR_EN	<b>Unexpected Type Error Enable.</b> Write 0 to enable UNEXP_TYPE_ERR_FLAG (bit 32) and to allow the unexpected type (synchronous error) event to generate an ERR.

Note 1. These are synchronous errors, that is, they do not result in the assertion of the GL Error signal but instead set the Exception bit in the response packet.

**6.1.1.5 GLD Power Management MSR (GLIU\_GLD\_MSR\_PM)**

MSR Address 51010004h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLIU\_GLD\_MSR\_PM Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												PMODE1	PMODE0		

**GLIU\_GLD\_MSR\_PM Bit Descriptions**

Bit	Name	Description
63:4	RSVD	<b>Reserved.</b> Write as read.
3:2	PMODE1	<b>Power Mode 1.</b> Statistics and Time Slice Counters. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
1:0	PMODE0	<b>Power Mode 0.</b> Online GLIU logic. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.

**6.1.1.6 GLD Diagnostic MSR (GLIU\_GLD\_MSR\_DIAG)**

MSR Address 51010005h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.





6.1.2.2 P2D Base Mask KEL Descriptors (GLIU\_P2D\_BMK[x])

**P2D Base Mask KEL Descriptor 0 (GLIU\_P2D\_BMK0)**

MSR Address 51010023h  
 Type R/W  
 Reset Value 000000FF\_FFF00000h

**P2D Base Mask KEL Descriptor 1 (GLIU\_P2D\_BMK1)**

MSR Address 51010024h  
 Type R/W  
 Reset Value 000000FF\_FFF00000h

This is a special version of a P2D\_BM descriptor to support routing the USB Keyboard Emulation Logic (KEL) native registers to the default port. The default port device on the Geode CS5535 companion device contains the KEL.

**GLIU\_P2D\_BMK[x] Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
PDID1_BMK			PCMP_BIZ_BMK	RSVD																			PBASE_BMK								
31	30	29		28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
PBASE_BMK												PMASK_BMK																			

**GLIU\_P2D\_BMK[x] Bit Descriptions**

Bit	Name	Description
63:61	PDID1_BMK	<b>Physical Descriptor Destination ID. Descriptor Destination ID.</b> These bits define which port to route the request to if it is a hit based on the other settings in this register. 000: Port 0 (GLIU)                      100: Port 4 (DD) 001: Port 1 (GLPCI_SB)                101: Port 5 (ACC) 010: Port 2 (USBC2)                    110: Port 6 (USBC1) 011: Port 3 (ATAC)                      111: Port 7 (GLCP)
60	PCMP_BIZ_BMK	<b>Physical Compare BIZZARO Flag.</b> If set, bit 8 of the address must be low for a hit on this descriptor.
59:40	RSVD	<b>Reserved.</b> Write as read.
39:20	PBASE_BMK	<b>Physical Memory Address Base.</b> These bits form the matching value against which the masked value of the physical address bits [31:12] are directly compared. If a match is found, then a hit is declared, depending on the setting of the BIZZARO flag comparator.
19:0	PMASK_BMK	<b>Physical Memory Address Mask.</b> These bits are used to mask physical address bits [31:12] for the purposes of this hit detection.



**GLIU\_PAE Bit Descriptions (Continued)**

Bit	Name	Description
5:4	PAE3	<b>Port Active Enable for Port 3 (ATAC).</b> See bits [15:14] for decode.
3:2	PAE2	<b>Port Active Enable for Port 2 (USBC2).</b> See bits [15:14] for decode.
1:0	PAE1	<b>Port Active Enable for Port 1 (GLPCI_SB).</b>

**6.1.3.3 Arbitration (GLIU\_ARB)**

MSR Address 51010082h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLIU\_ARB Register**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD	PIPE_DIS	RSVD																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
RSVD																															

**GLIU\_ARB Bit Descriptions**

Bit	Name	Description
63	RSVD	<b>Reserved.</b> Write as read.
62	PIPE_DIS	<b>Pipelined Arbitration Disabled.</b> 0: Pipelined arbitration enabled and the GLIU is not limited to one outstanding transaction. 1: Limit the entire GLIU to one outstanding transaction.
61:0	RSVD	<b>Reserved.</b> Write as read.

**6.1.3.4 Asynchronous SMI (GLIU\_ASMI)**

MSR Address 51010083h  
 Type R/W  
 Reset Value 00000000\_00000000h

ASMI is a condensed version of the Port ASMI signals. The EN bits ([15:8]) can be used to prevent a device from issuing an ASMI. A write of 1 to the EN bit disables the device's ASMI. The FLAG bits ([7:0]) are status bits. If high, an ASMI was generated due to the associated device. (See Section 4.1.4 "ASMI and Error" on page 57 for further details.)

**GLIU\_ASMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																P7_ASMI_EN	P6_ASMI_EN	P5_ASMI_EN	P4_ASMI_EN	P3_ASMI_EN	P2_ASMI_EN	P1_ASMI_EN	P0_ASMI_EN	P7_ASMI_FLAG	P6_ASMI_FLAG	P5_ASMI_FLAG	P4_ASMI_FLAG	P3_ASMI_FLAG	P2_ASMI_FLAG	P1_ASMI_FLAG	P0_ASMI_FLAG

**GLIU\_ASMI Bit Descriptions**

Bit	Name	Description
63:16	RSVD	Reserved.
15	P7_ASMI_EN	Port 7 (GLCP) Asynchronous SMI Enable.
14	P6_ASMI_EN	Port 6 (USBC1) Asynchronous SMI Enable.
13	P5_ASMI_EN	Port 5 (ACC) Asynchronous SMI Enable.
12	P4_ASMI_EN	Port 4 (DD) Asynchronous SMI Enable.
11	P3_ASMI_EN	Port 3 (ATAC) Asynchronous SMI Enable.
10	P2_ASMI_EN	Port 2 (USBC2) Asynchronous SMI Enable.
9	P1_ASMI_EN	Port 1 (GLPCI_SB) Asynchronous SMI Enable.
8	P0_ASMI_EN	Port 0 (GLIU) Asynchronous SMI Enable.
7	P7_ASMI_FLAG (RO)	Port 7 (GLCP) Asynchronous SMI Flag (Read Only).
6	P6_ASMI_FLAG (RO)	Port 6 (USBC1) Asynchronous SMI Flag (Read Only).
5	P5_ASMI_FLAG (RO)	Port 5 (ACC) Asynchronous SMI Flag (Read Only).
4	P4_ASMI_FLAG (RO)	Port 4 (DD) Asynchronous SMI Flag (Read Only).
3	P3_ASMI_FLAG (RO)	Port 3 (ATAC) Asynchronous SMI Flag (Read Only).
2	P2_ASMI_FLAG (RO)	Port 2 (USBC2) Asynchronous SMI Flag (Read Only).
1	P1_ASMI_FLAG (RO)	Port 1 (GLPCI_SB) Asynchronous SMI Flag (Read Only).
0	P0_ASMI_FLAG (RO)	Port 0 (GLIU) Asynchronous SMI Flag (Read Only).

**6.1.3.5 Asynchronous Error (GLIU\_AERR)**

MSR Address 51010084h  
 Type R/W  
 Reset Value 00000000\_00000000h

ERR is a condensed version of the port (asynchronous) ERR signals. The EN bits ([15:8]) can be used to prevent a device from issuing an ERR. A write of 1 to the EN bit disables the device's ERR. The FLAG bits ([7:0]) are status bits. If high, an ERR was generated due to the associated device. (See Section 4.1.4 "ASMI and Error" on page 57 for further details.)

**GLIU\_AERR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																P7_AERR_EN	P6_AERR_EN	P5_AERR_EN	P4_AERR_EN	P3_AERR_EN	P2_AERR_EN	P1_AERR_EN	P0_AERR_EN	P7_AERR_FLAG	P6_AERR_FLAG	P5_AERR_FLAG	P4_AERR_FLAG	P3_AERR_FLAG	P2_AERR_FLAG	P1_AERR_FLAG	P0_AERR_FLAG

## GLIU\_AERR Bit Descriptions

Bit	Name	Description
63:16	RSVD	Reserved.
15	P7_AERR_EN	Port 7 (GLCP) Asynchronous Error Enable.
14	P6_AERR_EN	Port 6 (USBC1) Asynchronous Error Enable.
13	P5_AERR_EN	Port 5 (ACC) Asynchronous Error Enable.
12	P4_AERR_EN	Port 4 (DD) Asynchronous Error Enable.
11	P3_AERR_EN	Port 3 (ATAC) Asynchronous Error Enable.
10	P2_AERR_EN	Port 2 (USBC2) Asynchronous Error Enable.
9	P1_AERR_EN	Port 1 (GLPCI_SB) Asynchronous Error Enable.
8	P0_AERR_EN	Port 0 (GLIU) Asynchronous Error Enable.
7	P7_AERR_FLAG (RO)	Port 7 (GLCP) Asynchronous Error Flag (Read Only).
6	P6_AERR_FLAG (RO)	Port 6 (USBC1) Asynchronous Error Flag (Read Only).
5	P5_AERR_FLAG (RO)	Port 5 (ACC) Asynchronous Error Flag (Read Only).
4	P4_AERR_FLAG (RO)	Port 4 (DD) Asynchronous Error Flag (Read Only).
3	P3_AERR_FLAG (RO)	Port 3 (ATAC) Asynchronous Error Flag (Read Only).
2	P2_AERR_FLAG (RO)	Port 2 (USBC2) Asynchronous Error Flag (Read Only).
1	P1_AERR_FLAG (RO)	Port 1 (GLPCI_SB) Asynchronous Error Flag (Read Only).
0	P0_AERR_FLAG (RO)	Port 0 (GLIU) Asynchronous Error Flag (Read Only).

## 6.1.3.6 Debug (GLIU\_DEBUG)

MSR Address 51010085h  
 Type R/W  
 Reset Value 00000000\_00000004h

## GLIU\_DEBUG Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

## GLIU\_DEBUG Bit Descriptions

Bit	Name	Description
63:0	RSVD	Reserved. Write as read.

**6.1.3.7 Physical Capabilities (GLIU\_PHY\_CAP)**

MSR Address 51010086h  
 Type RO  
 Reset Value 327920A0\_80000003h

This register provides the resources available in the Geode CS5535 companion device.

**GLIU\_PHY\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD	NSTAT_CNT			NDBG_DA_CMP			NDBG_RQ_CMP			NPORTS			NCOH			NIOD_SC						NIOD_BM						NP2D_BMK			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
NP2D_BMK	NP2D_SC						NP2D_RO						NP2D_R						NP2D_BMO						NP2D_BM						

**GLIU\_PHY\_CAP Bit Descriptions**

Bit	Name	Description
63	RSVD	<b>Reserved.</b> Returns 0.
62:60	NSTAT_CNT	<b>Number Of Statistic Counters.</b> Provides the number of available Statistic Counters.
59:57	NDBG_DA_CMP	<b>Number Of Data Comparators.</b> Provides the number of available Data Comparators.
56:54	NDBG_RQ_CMP	<b>Number Of Request Comparators.</b> Provides the number of available Request Comparators.
53:51	NPORTS	<b>Number of Ports on the GLIU.</b> Provides the number of available ports on the GLIU.
50:48	NCOH	<b>Number of Coherent Devices.</b> Provides the number of available Coherent Devices.
47:42	NIOD_SC	<b>Number of IOD_SC Descriptors.</b> Provides the number of available IOD_SC Descriptors.
41:36	NIOD_BM	<b>Number of IOD_BM Descriptors.</b> Provides the number of available IOD_BM Descriptors.
35:30	NP2D_BMK	<b>Number of P2D_BMK Descriptors.</b> Provides the number of available P2D_BMK Descriptors.
29:24	NP2D_SC	<b>Number of P2D_SC Descriptors.</b> Provides the number of available P2D_SC Descriptors
23:18	NP2D_RO	<b>Number of P2D_RO Descriptors.</b> Provides the number of available P2D_RO Descriptors.
17:12	NP2D_R	<b>Number of P2D_R Descriptors.</b> Provides the number of available P2D_R Descriptors.
11:6	NP2D_BMO	<b>Number of P2D_BMO Descriptors.</b> Provides the number of available P2D_BMO Descriptors.
5:0	NP2D_BM	<b>Number of P2D_BM Descriptors.</b> Provides the number of available P2D_BM Descriptors.

**6.1.3.8 N Outstanding Response (GLIU\_NOUT\_RESP)**

MSR Address 51010087h  
 Type RO  
 Reset Value 00000000\_00000000h

**GLIU\_NOUT\_RESP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLIU\_NOUT\_RESP Bit Descriptions**

Bit	Name	Description
63:0	RSVD	Reserved. Returns 0.

**6.1.3.9 Number of Outstanding Write Data (GLIU\_NOUT\_WDATA)**

MSR Address 51010088h  
 Type RO  
 Reset Value 00000000\_00000000h

**GLIU\_NOUT\_WDATA Register Map (RO)**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLIU\_NOUT\_WDATA Bit Descriptions**

Bit	Name	Description
63:0	RSVD	Reserved. Returns 0.

**6.1.3.10 WHO AM I (GLIU\_WHOAMI)**

MSR Address 5101008Bh  
 Type RO  
 Reset Value Configuration Dependent

**GLIU\_WHOAMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLIU\_WHOAMI Bit Descriptions**

Bit	Name	Description
63:0	RSVD	Reserved. Returns 0.



**6.1.3.11 Slave Disable (GLIU\_SLV\_DIS)**

MSR Address 5101008Ch  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLIU\_SLV\_DIS Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLIU\_SLV\_DIS Bit Descriptions**

Bit	Name	Description
63:0	RSVD	<b>Reserved.</b> Write as read.

### 6.1.3.12 Descriptor Statistic Counters (GLIU\_STATISTIC\_CNT[x])

#### Descriptor Statistic Counter 0 (GLIU\_STATISTIC\_CNT0)

MSR Address 510100A0h  
Type WO  
Reset Value 00000000\_00000000h

#### Descriptor Statistic Counter 1 (GLIU\_STATISTIC\_CNT1)

MSR Address 510100A4h  
Type WO  
Reset Value 00000000\_00000000h

#### Descriptor Statistic Counter 2 (GLIU\_STATISTIC\_CNT2)

MSR Address 510100A8h  
Type WO  
Reset Value 00000000\_00000000h

These registers work in conjunction with the GLIU\_STATISTIC\_MASK[x] and the GLIU\_STATISTIC\_ACTION[x] registers. The counters count 'hits' on the P2D and IOD descriptors. The counter behaves as setup in the GLIU\_STATISTIC\_ACTION[x] register.

#### GLIU\_STATISTIC\_CNT[x] Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
LOAD_VAL																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT																															

#### GLIU\_STATISTIC\_CNT[x] Bit Descriptions

Bit	Name	Description
63:32	LOAD_VAL	<b>Counter Load Value.</b> A value loaded here will be used as the initial Statistics Counter value when a LOAD action occurs or is commanded.
31:0	CNT	<b>Counter Value.</b> These bits provide the current counter value when read.

**6.1.3.13 Descriptor Statistic Mask (GLIU\_STATISTIC\_MASK[x])****Descriptor Statistic Mask 0 (GLIU\_STATISTIC\_MASK0)**

MSR Address 510100A1h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Descriptor Statistic Mask 1 (GLIU\_STATISTIC\_MASK1)**

MSR Address 510100A5h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Descriptor Statistic Mask 2 (GLIU\_STATISTIC\_MASK2)**

MSR Address 510100A9h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLIU\_STATISTIC\_MASK[x] Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IOD_MASK																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P2D_MASK																															

**GLIU\_STATISTIC\_MASK[x] Bit Descriptions**

Bit	Name	Description
63:32	IOD_MASK	<p><b>Mask for Hits to each IOD.</b> Hits are determined after the request is arbitrated.</p> <p>A hit is determined by the following logical equation:</p> $\text{hit} =  (\text{IOD\_MASK}[n-1:0] \& \text{RQ\_DESC\_HIT}[n-1:0] \&\& \text{is\_io})    (\text{P2D\_MASK}[n-1:0] \& \text{RQ\_DESC\_HIT}[n-1:0] \&\& \text{is\_mem})$
31:0	P2D_MASK	<p><b>Mask for Hits to each P2D.</b></p> <p>A hit is determined by the following logical equation:</p> $\text{hit} =  (\text{IOD\_MASK}[n-1:0] \& \text{RQ\_DESC\_HIT}[n-1:0] \&\& \text{is\_io})    (\text{P2D\_MASK}[n-1:0] \& \text{RQ\_DESC\_HIT}[n-1:0] \&\& \text{is\_mem})$

### 6.1.3.14 Descriptor Statistic Action (GLIU\_STATISTIC\_ACTION[x])

#### Descriptor Statistic Action 0 (GLIU\_STATISTIC\_ACTION0)

MSR Address 510100A2h  
 Type R/W  
 Reset Value 00000000\_00000000h

#### Descriptor Statistic Action 1 (GLIU\_STATISTIC\_ACTION1)

MSR Address 510100A6h  
 Type R/W  
 Reset Value 00000000\_00000000h

#### Descriptor Statistic Action 2 (GLIU\_STATISTIC\_ACTION2)

MSR Address 510100AAh  
 Type R/W  
 Reset Value 00000000\_00000000h

### GLIU\_STATISTIC\_ACTION[x] Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								PREDIV														WRAP	ZERO_ERR	ZERO_SMI	ALWAYS_DEC	HIT_ERR	HIT_SMI	HIT_DEC	HIT_LDEN		

### GLIU\_STATISTIC\_ACTION[x] Bit Descriptions

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Write as read.
23:8	PREDIV	<b>Pre-divider used for ALWAYS_DEC.</b> The pre-divider is free running and extends the depth of the counter.
7	WRAP	<b>Decrement Counter Beyond Zero and Wrap.</b> 0: Disable wrap; counter stops when it reaches zero. 1: Enable wrap; counter decrements through 0 to all ones.
6	ZERO_ERR	<b>Asset AERR on Cnt = 0.</b> Assert AERR (internal GLIU_P_SERR) when STATISTIC_CNT[x] = 0. 0: Disable. 1: Enable.
5	ZERO_SMI	<b>Assert ASMI on Cnt = 0.</b> Assert ASMI (internal GLIU_P_ASMI) when STATISTIC_CNT[x] = 0. 0: Disable. 1: Enable.
4	ALWAYS_DEC	<b>Always Decrement Counter.</b> If enabled, the counter will decrement on every memory clock, subject to the prescaler value PREDIV (bits [23:8]). Decrementing will continue unless loading is occurring due to another action, or if the counter reaches zero and WRAP is disabled (bit[7]). 0: Disable. 1: Enable.

## GLIU\_STATISTIC\_ACTION[x] Bit Descriptions (Continued)

Bit	Name	Description
3	HIT_ERR	<b>Assert AERR on Descriptor Hit.</b> This bit causes an asynchronous error to be generated when a matching descriptor hit occurs, or not. The descriptor hits are ANDed with the masks and then all ORed together. 0: Disable. 1: Enable.
2	HIT_SMI	<b>Assert ASMI on Descriptor Hit.</b> This bit causes an ASMI to be generated when a matching descriptor hit occurs, or not. The descriptor hits are ANDed with the masks and then all ORed together. 0: Disable. 1: Enable.
1	HIT_DEC	<b>Decrement Counter on Descriptor Hit.</b> This bit causes the associated counter to decrement when a matching descriptor hit occurs, or not. The descriptor hits are ANDed with the masks and then all ORed together. 0: Disable. 1: Enable.
0	HIT_LDEN	<b>Load Counter on Descriptor Hit.</b> This bit causes the associated counter to reload its LOAD_VAL when a matching descriptor hit occurs, or not. The descriptor hits are ANDed with the masks and then all ORed together. 0: Disable. 1: Enable.

## 6.1.3.15 Request Compare Value (GLIU\_RQ\_COMP\_VAL)

MSR Address 510100C0h  
 Type R/W  
 Reset Value 001FFFFFF\_FFFFFFFFh

The RQ Compare Value and the RQ Compare Mask enable traps on specific transactions. A hit to the RQ Compare is determined by  $hit = (RQ\_IN \& RQ\_COMP\_MASK) == RQ\_COMP\_VAL$ . A hit can trigger the RQ\_COMP error sources when they are enabled. The value is compared only after the packet is arbitrated.

## GLIU\_RQ\_COMP\_VAL Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD											RQ_COMPVAL																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RQ_COMPVAL																															

## GLIU\_RQ\_COMP\_VAL Bit Descriptions

Bit	Name	Description
63:53	RSVD	<b>Reserved.</b> Write as read.
52:0	RQ_COMPVAL	<b>Request Packet Value.</b> This is the value compared against the logical bit-wise AND of the incoming request packet and the RQ_COMP_MASK in order to determine a hit.

**6.1.3.16 Request Compare Mask (GLIU\_RQ\_COMP\_MASK)**

MSR Address 510100C1h  
 Type R/W  
 Reset Value 00000000\_00000000h

The RQ Compare Value and the RQ Compare Mask enable traps on specific transactions. A hit to the RQ Compare is determined by  $\text{hit} = (\text{RQ\_IN} \& \text{RQ\_COMP\_MASK}) == \text{RQ\_COMP\_VAL}$ . A hit can trigger the RQ\_COMP error sources when they are enabled. The value is compared only after the packet is arbitrated.

**GLIU\_RQ\_COMP\_MASK Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD											RQ_COMPMASK																				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RQ_COMPMASK																															

**GLIU\_RQ\_COMP\_MASK Bit Descriptions**

Bit	Name	Description
63:53	RSVD	<b>Reserved.</b> Write as read.
52:0	RQ_COMPMASK	<b>Request Packet Mask.</b> This field is bit-wise logically ANDed with the incoming Request Packet before it is compared to the RQ_COMPVAL.

**6.1.3.17 Data Compare Value Low (GLIU\_DA\_COMP\_VAL\_LO)**

MSR Address 510100D0h  
 Type R/W  
 Reset Value 00001FFF\_FFFFFFFFh

The DA Compare Value and the DA Compare Mask enable traps on specific transactions. A hit to the DA Compare is determined by  $\text{hit} = (\text{DA\_IN} \& \text{DA\_COMP\_MASK}) == \text{DA\_COMP\_VAL}$ . A hit can trigger the DA\_CMP error sources when they are enabled. The value is compared only after the packet is arbitrated.

**GLIU\_DA\_COMP\_VAL\_LO Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																DALO_COMPVAL															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DALO_COMPVAL																															

**GLIU\_DA\_COMP\_VAL\_LO Bit Descriptions**

Bit	Name	Description
63:45	RSVD	<b>Reserved.</b> Write as read.
44:0	DALO_COMPVAL	<b>Data Packet Compare Value [44:0].</b> This field forms the lower portion of the data value that is compared to the logical bit-wise AND of the incoming data value and the data value compare mask in order to determine a hit. The “HI” and “LO” portions of the incoming data, the compare value, and the compare mask, are assembled into complete bit patterns before these operations occur.

**6.1.3.18 Data Compare Value High (GLIU\_DA\_COMP\_VAL\_HI)**

MSR Address 510100D1h  
 Type R/W  
 Reset Value 0000000F\_FFFFFFFFh

The DA Compare Value and the DA Compare Mask enable traps on specific transactions. A hit to the DA Compare is determined by  $hit = (DA\_IN \& DA\_COMP\_MASK) == DA\_COMP\_VAL$ . A hit can trigger the DA\_CMP error sources when they are enabled. The value is compared only after the packet is arbitrated.

**GLIU\_DA\_COMP\_VAL\_HI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
DAHI_COMPVAL																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAHI_COMPVAL																															

**GLIU\_DA\_COMP\_VAL\_HI Bit Descriptions**

Bit	Name	Description
63:36	RSVD	<b>Reserved.</b> Write as read.
35:0	DAHI_COMPVAL	<b>DA Packet Compare Value [80:45].</b> This field forms the upper portion of the data value that is compared to the logical bit-wise AND of the incoming data value and the data value compare mask in order to determine a hit. The “HI” and “LO” portions of the incoming data, the compare value, and the compare mask, are assembled into complete bit patterns before these operations occur.

**6.1.3.19 Data Compare Mask Low (GLIU\_DA\_COMP\_MASK\_LO)**

MSR Address 510100D2h  
 Type R/W  
 Reset Value 00000000\_00000000h

The DA Compare Value and the DA Compare Mask enable traps on specific transactions. A hit to the DA Compare is determined by  $hit = (DA\_IN \& DA\_COMP\_MASK) == DA\_COMP\_VAL$ . A hit can trigger the DA\_COMP error sources when they are enabled. The value is compared only after the packet is arbitrated.

**GLIU\_DA\_COMP\_MASK\_LO Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																DALO_COMPMASK															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DALO_COMPMASK																															

**GLIU\_DA\_COMP\_MASK\_LO Bit Descriptions**

Bit	Name	Description
63:45	RSVD	<b>Reserved.</b> Write as read.
44:0	DALO_COMPMASK	<b>Data Packet Compare Value [44:0].</b> This field is forms the lower portion of the data COMPMASK value, that is then bit-wise logically ANDed with the incoming data value before it is compared to the DA_COMPVAL. The “HI” and “LO” portions of the incoming data, the compare value, and the compare mask, are assembled into complete bit patterns before these operations occur.

**6.1.3.20 Data Compare Mask High (GLIU\_DA\_COMP\_MASK\_HI)**

MSR Address 510100D3h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLIU\_DA\_COMP\_MASK\_HI Register**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
DAHI_COMPMASK																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAHI_COMPMASK																															

**GLIU\_DA\_COMP\_MASK\_HI Bit Descriptions**

Bit	Name	Description
63:36	RSVD	<b>Reserved.</b> Write as read.
35:0	DAHI_COMPMASK	<b>DA Packet Compare Mask [80:45].</b> This field is forms the upper portion of the data COMPMASK value that is then bit-wise logically ANDed with the incoming data value before it is compared to the DA_COMP_VAL. The “HI” and “LO” portions of the incoming data (the compare value and compare mask) are assembled into complete bit patterns before these operations occur.





**GLIU\_IOD\_BM[x] Bit Descriptions (Continued)**

Bit	Name	Description
59:40	RSVD	<b>Reserved.</b> Write as read.
39:20	IBASE_BM	<b>Physical I/O Address Base.</b> These bits form the matching value against which the masked value of the physical address, bits [31:12] are directly compared. If a match is found, then a hit is declared, depending on the setting of the BIZZARO flag comparator.
19:0	IMASK_BM	<b>Physical I/O Address Mask.</b> These bits are used to mask address bits [31:12] for the purposes of this hit detection.

**6.1.4.2 IOD Swiss Cheese Descriptors (GLIU\_IOD\_SC[x])**

**IOD Swiss Cheese 0 (GLIU\_IOD\_SC0)**

MSR Address 510100EAh  
 Type R/W  
 Reset Value 60000000\_403003F0h

**IOD Swiss Cheese 4 (GLIU\_IOD\_SC4)**

MSR Address 510100EEh  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 1 (GLIU\_IOD\_SC1)**

MSR Address 510100EBh  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 5 (GLIU\_IOD\_SC5)**

MSR Address 510100EFh  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 2 (GLIU\_IOD\_SC2)**

MSR Address 510100ECh  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 6 (GLIU\_IOD\_SC6)**

MSR Address 510100F0h  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 3 (GLIU\_IOD\_SC3)**

MSR Address 510100EDh  
 Type R/W  
 Reset Value 00000000\_00000000h

**IOD Swiss Cheese 7 (GLIU\_IOD\_SC7)**

MSR Address 510100F1h  
 Type R/W  
 Reset Value 00000000\_00000000h

Each of these eight descriptors checks that the physical address supplied by the device's request on the address bits is equal to the IBASE\_SC field of descriptor register bits and that the enable write or read conditions given by the descriptor register fields WEN and REN respectively matches the request type and enable fields given on the physical address bits of the device's request. If the above matches, then the descriptor has a hit condition and routes the received address to the programmed destination ID, IDID1\_SC field of the descriptor register bits.

**GLIU\_IOD\_SC[x] Register**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
IDID_SC			ICMP_BIZ_SC	RSVD																											
31	30	29		28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
EN_SC								RSVD	WEN_SC	REN_SC	IBASE_SC																RSVD				



## 6.2 GeodeLink™ PCI South Bridge Register Descriptions

The GeodeLink™ PCI South Bridge (GLPCI\_SB) register set consists of:

- Standard GeodeLink Device (GLD) MSRs
- GLPCI\_SB Specific MSRs
- PCI Configuration Registers

The MSRs (both Standard and GLPCI\_SB Specific) are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details on MSR addressing.

Additionally, all GLPCI\_SB Specific MSRs can be accessed through both the PCI and GLIU interfaces. See

Section 6.2.3 "PCI Configuration Registers" on page 240 for details.

The PCI configuration registers can only be accessed through the PCI interface and include:

- The first 16 bytes of standard PCI configuration registers.
- MSR access registers:
  - PMCTRL
  - PMADDR
  - PMDATA0
  - PMDATA1

Tables 6-5 through 6-7 are register summary tables that include reset values and page references where the bit descriptions are provided.

**Table 6-5. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register	Reset Value	Reference
51000000h	RO	GLD Capabilities MSR (GLPCI_GLD_MSR_CAP)	00000000_002051xxh	Page 230
51000001h	R/W	GLD Master Configuration MSR (GLPCI_GLD_MSR_CONFIG)	00000000_00000000h	Page 230
51000002h	R/W	GLD SMI MSR (GLPCI_GLD_MSR_SMI)	00000000_00000000h	Page 231
51000003h	R/W	GLD Error MSR (GLPCI_GLD_MSR_ERROR)	00000000_00000000h	Page 232
51000004h	R/W	GLD Power Management MSR (GLPCI_GLD_MSR_PM)	00000000_00000000h	Page 233
51000005h	R/W	GLD Diagnostic MSR (GLPCI_GLD_MSR_DIAG)	00000000_00000000h	Page 234

**Table 6-6. GLPCI\_SB Specific MSRs Summary**

MSR Address	Type	Register	Reset Value	Reference
51000010h	R/W	Global Control (GLPCI_CTRL)	44000030_00000003h	Page 234
51000020h	R/W	Region 0 Configuration (GLPCI_R0)	00000000_00000000h	Page 238
51000021h	R/W	Region 1 Configuration (GLPCI_R1)	00000000_00000000h	Page 238
51000022h	R/W	Region 2 Configuration (GLPCI_R2)	00000000_00000000h	Page 238
51000023h	R/W	Region 3 Configuration (GLPCI_R3)	00000000_00000000h	Page 238
51000024h	R/W	Region 4 Configuration (GLPCI_R4)	00000000_00000000h	Page 238
51000025h	R/W	Region 5 Configuration (GLPCI_R5)	00000000_00000000h	Page 238
51000026h	R/W	Region 6 Configuration (GLPCI_R6)	00000000_00000000h	Page 238
51000027h	R/W	Region 7 Configuration (GLPCI_R7)	00000000_00000000h	Page 238
51000028h	R/W	Region 8 Configuration (GLPCI_R8)	00000000_00000000h	Page 238
51000029h	R/W	Region 9 Configuration (GLPCI_R9)	00000000_00000000h	Page 238
5100002Ah	R/W	Region 10 Configuration (GLPCI_R10)	00000000_00000000h	Page 238
5100002Bh	R/W	Region 11 Configuration (GLPCI_R11)	00000000_00000000h	Page 238
5100002Ch	R/W	Region 12 Configuration (GLPCI_R12)	00000000_00000000h	Page 238

**Table 6-6. GLPCI\_SB Specific MSRs Summary**

MSR Address	Type	Register	Reset Value	Reference
5100002Dh	R/W	Region 13 Configuration (GLPCI_R13)	00000000_00000000h	Page 238
5100002Eh	R/W	Region 14 Configuration (GLPCI_R14)	00000000_00000000h	Page 238
5100002Fh	R/W	Region 15 Configuration (GLPCI_R15)	00000000_00000000h	Page 238
51000030h	RO	PCI Configuration Space Header Byte 0-3 (GLPCI_PCIHEAD_BYTE0-3)	00000000_002A100Bh	Page 239
51000031h	RO	PCI Configuration Space Header Byte 4-7 (GLPCI_PCIHEAD_BYTE4-7)	00000000_00000000h	Page 239
51000032h	RO	PCI Configuration Space Header Byte 8-B (GLPCI_PCIHEAD_BYTE8-B)	00000000_00000000h	Page 239
51000033h	RO	PCI Configuration Space Header Byte C-F (GLPCI_PCIHEAD_BYTEC-F)	00000000_00000000h	Page 240

**Table 6-7. PCI Configuration Registers**

Index	Type	Width (Bits)	Name	Reset Value	Reference
00h	RO	32 (Note 1)	PCI Configuration Space Header Byte 0-3 (GLPCI_PCI_HEAD_BYTE0-3)	002A100Bh	Page 240
04h	RO	32 (Note 1)	PCI Configuration Space Header Byte 4-7 (GLPCI_PCI_HEAD_BYTE4-7)	00000000h	Page 241
08h	RO	32 (Note 1)	PCI Configuration Space Header Byte 8-B (GLPCI_PCI_HEAD_BYTE8-B)	FF0000xxh	Page 241
0Ch	RO	32 (Note 1)	PCI Configuration Space Header Byte C-F (GLPCI_PCI_HEAD_BYTEC-F)	00000000h	Page 242
F0h	R/W	32	PCI MSR Control (GLPCI_PMCTRL)	00000001h	Page 242
F4h	R/W	32	PCI MSR Address (GLPCI_PMADDR)	00000000h	Page 243
F8h	R/W	32	PCI MSR Data 0 (GLPCI_PMDATA0)	00000000h	Page 243
FCh	R/W	32	PCI MSR Data 1 (GLPCI_PMDATA1)	00000000h	Page 244

Note 1. Read address bits [1:0] are ignored and taken as 00.

## 6.2.1 Standard GeodeLink™ Device (GLD) MSRs

### 6.2.1.1 GLD Capabilities MSR (GLPCI\_GLD\_MSR\_CAP)

MSR Address 51000000h  
 Type RO  
 Reset Value 00000000\_002051xxh

**GLPCI\_GLD\_MSR\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID														REV_ID									

**GLPCI\_GLD\_MSR\_CAP Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2051h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

### 6.2.1.2 GLD Master Configuration MSR (GLPCI\_GLD\_MSR\_CONFIG)

MSR Address 51000001h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLPCI\_GLD\_MSR\_CONFIG Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							PRI			RSVD	PID				

**GLPCI\_GLD\_MSR\_CONFIG Bit Descriptions**

Bit	Name	Description
63:7	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
6:4	PRI	<b>Priority Level.</b> Always write 0.
3	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
2:0	PID	<b>Priority ID.</b> Always write 0.

**6.2.1.3 GLD SMI MSR (GLPCI\_GLD\_MSR\_SMI)**

MSR Address 51000002h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further SMI/ASMI generation details.)

**GLPCI\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									TAS_ASMI_FLAG	PAR_ASMI_FLAG	SYSE_ASMI_FLAG	EXCEP_ASMI_FLAG	SSMI_ASMI_FLAG	TAR_ASMI_FLAG	MAR_ASMI_FLAG	RSVD									TAS_ASMI_EN	PAR_ASMI_EN	SYSE_ASMI_EN	EXCEP_ASMI_EN	SSMI_ASMI_EN	TAR_ASMI_EN	MAR_ASMI_EN

**GLPCI\_GLD\_MSR\_SMI Bit Descriptions**

Bit	Name	Description
63:23	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
22	TAS_ASMI_FLAG	<b>Target Abort Signaled ASMI Flag.</b> If high, records that an ASMI was generated due the signaling of a target abort on the PCI bus. Write 1 to clear; writing 0 has no effect. TA_ASMI_EN (bit 6) must be high to generate ASMI and set flag.
21	PAR_ASMI_FLAG	<b>Parity Error ASMI Flag.</b> If high, records that an ASMI was generated due to the detection of a PCI bus parity error. Write 1 to clear; writing 0 has no effect. PAR_ASMI_EN (bit 5) must be high to generate ASMI and set flag.
20	SYSE_ASMI_FLAG	<b>System Error ASMI Flag.</b> If high, records that an ASMI was generated due to the detection of a PCI bus system error. Write 1 to clear; writing 0 has no effect. SYSE_ASMI_EN (bit 4) must be high to generate ASMI and set flag.
19	EXCEP_ASMI_FLAG	<b>Exception Bit Flag.</b> If high, records that an ASMI was generated due to the EXCEP bit being set in the received GLIU read or write response packet. Write 1 to clear; writing 0 has no effect. EXCEP_ASMI_EN (bit 3) must be set to enable this flag.
18	SSMI_ASMI_FLAG	<b>SSMI ASMI Flag.</b> If high, records that an ASMI was generated due to the SSMI bit being set in the received GLIU read or write response packet. Write 1 to clear; writing 0 has no effect. SSMI_ASMI_EN (bit 2) must be set to enable this flag.
17	TAR_ASMI_FLAG	<b>Target Abort Received ASMI Flag.</b> If high, records that an ASMI was generated due to the reception of a target abort on the PCI bus. Write 1 to clear; writing 0 has no effect. TAR_ASMI_EN (bit 1) must be high to generate ASMI and set flag.
16	MAR_ASMI_FLAG	<b>Master Abort Received ASMI Flag.</b> If high, records that an ASMI was generated due to the reception of a master abort on the PCI bus. Write 1 to clear; writing 0 has no effect. MAR_ASMI_EN (bit 0) be high to generate ASMI and set flag.
15:7	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
6	TAS_ASMI_EN	<b>Target Abort Signaled ASMI Enable.</b> Write 1 to enable TAS_ASMI_FLAG (bit 22) and to allow the event to generate an ASMI.
5	PAR_ASMI_EN	<b>Parity Error ASMI Enable.</b> Write 1 to enable PAR_ASMI_FLAG (bit 21) and to allow the event to generate an ASMI.
4	SYSE_ASMI_EN	<b>System Error SMI Enable.</b> Write 1 to enable SYSE_ASMI_FLAG (bit 20) and to allow the event to generate an ASMI.

## GLPCI\_GLD\_MSR\_SMI Bit Descriptions (Continued)

Bit	Name	Description
3	EXCEP_ASMI_EN	<b>Exception Bit Enable.</b> Write 1 to enable EXCEP_ASMI_FLAG (bit 19) and to allow the event.
2	SSMI_EN	<b>SSMI Enable.</b> Write 1 to enable SSMI_ASMI_FLAG bit (bit 18) and to allow the event.
1	TAR_ASMI_EN	<b>Target Abort Received ASMI Enable.</b> Write 1 to enable TAR_ASMI_FLAG (bit 17) and to allow the event to generate an ASMI.
0	MAR_ASMI_EN	<b>Master Abort Received ASMI Enable.</b> Write 1 to enable MAR_ASMI_FLAG (bit 16) and to allow the event to generate an ASMI.

## 6.2.1.4 GLD Error MSR (GLPCI\_GLD\_MSR\_ERROR)

MSR Address 51000003h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.4 "MSR Address 3: Error Control" on page 78 for further details.)

## GLPCI\_GLD\_MSR\_ERROR Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									TAS_ERR_FLAG	PARE_ERR_FLAG	SYSE_ERR_FLAG	EXCEP_ERR_FLAG	RSVD	TAR_ERR_FLAG	MAR_ERR_FLAG	RSVD									TAS_ERR_EN	PARE_ERR_EN	SYSE_ERR_EN	EXCEP_ERR_EN	RSVD	TAR_ERR_EN	MAR_ERR_EN

## GLPCI\_GLD\_MSR\_ERROR Bit Descriptions

Bit	Name	Description
63:23	RSVD	<b>Reserved (Read Only).</b> Returns 0.
22	TAS_ERR_FLAG	<b>Target Abort Signaled Error Flag.</b> If high, records that an ERR was generated due to signaling of a target abort on the PCI bus. Write 1 to clear; writing 0 has no effect. TAS_ERR_EN (bit 6) must be set to enable this event and set flag.
21	PARE_ERR_FLAG	<b>Parity Error Flag.</b> If high, records that an ERR was generated due to the detection of a PCI bus parity error. Write 1 to clear; writing 0 has no effect. PARE_ERR_EN (bit 5) must be set to enable this event and set flag.
20	SYSE_ERR_FLAG	<b>System Error Flag.</b> If high, records that an ERR was generated due to the detection of a PCI bus system error. Write 1 to clear; writing 0 has no effect. SYSE_ERR_EN (bit 4) must be set to enable this event and set flag.
19	EXCEP_ERR_FLAG	<b>Exception Bit Error Flag.</b> If high, records that the EXCP bit in the received GLIU read or write response packet is set. Write 1 to clear. EXCEP_ERR_EN (bit 3) must be set to enable this event and set flag.
18	RSVD	<b>Reserved (Read Only).</b> Returns 0.
17	TAR_ERR_FLAG	<b>Target Abort Received Error Flag.</b> If high, records that an ERR was generated due to the reception of a target abort on the PCI bus. Write 1 to clear; writing 0 has no effect. TAR_ERR_EN (bit 1) must be set to enable this event and set flag.



## GLPCI\_GLD\_MSR\_ERROR Bit Descriptions (Continued)

Bit	Name	Description
16	MAR_ERR_FLAG	<b>Master Abort Received Error Flag.</b> If high, records that an ERR was generated due to the reception of a master abort on the PCI bus. Write 1 to clear; writing 0 has no effect. MAR_ERR_EN (bit 0) must be set to enable this event and set flag.
15:7	RSVD	<b>Reserved (Read Only).</b> Returns 0.
6	TAS_ERR_EN	<b>Target Abort Signaled Error Enable.</b> Write 1 to enable TAS_ERR_FLAG (bit 22) and to allow the event to generate an ERR.
5	PARE_ERR_EN	<b>Parity Error Enable.</b> Write 1 to enable PAR_ERR_FLAG (bit 21) and to allow the event to generate an ERR.
4	SYSE_ERR_EN	<b>System Error Enable.</b> Write 1 to enable SYSE_ERR_FLAG (bit 20) and to allow the event to generate an ERR.
3	EXCEP_ERR_EN	<b>Exception Bit Error Enable.</b> Write 1 to enable EXCEP_ERR_FLAG (bit 19) and to allow the event to generate an ERR.
2	RSVD	<b>Reserved (Read Only).</b> Returns 0.
1	TAR_ERR_EN	<b>Target Abort Received Error Enable.</b> Write 1 to enable TAR_ERR_FLAG (bit 17) and to allow the event to generate an ERR.
0	MAR_ERR_EN	<b>Master Abort Received Enable.</b> Write 1 to enable MAR_ERR_FLAG (bit 16) and to allow the event to generate an ERR.

## 6.2.1.5 GLD Power Management MSR (GLPCI\_GLD\_MSR\_PM)

MSR Address 51000004h  
 Type R/W  
 Reset Value 00000000\_00000000h

## CLPCI\_GLD\_MSR\_PM Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD														IO MODEA		RSVD															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								P MODE2		P MODE1		P MODE0			

## GLPCI\_GLD\_MSR\_PM Bit Descriptions

Bit	Name	Description
63:50	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
49:48	IOMODEA	<p><b>I/O Mode A Control.</b> These bits determine how the associated PCI inputs and outputs will behave when the PMC asserts two internal signals that are controlled by PMS I/O Offset 20h and 0Ch. The list of affected signals is given in Table 4-11 "Sleep Driven PCI Signals" on page 79.</p> <p>00: No gating of I/O cells during a Sleep sequence (Default).</p> <p>01: During a power management Sleep sequence, force inputs to their non-asserted state when PM_IN_SLPCTL is enabled.</p> <p>10: During a power management Sleep sequence, force inputs to their non-asserted state when PM_IN_SLPCTL is enabled, and park (force) outputs low when PM_OUT_SLPCTL is enabled.</p> <p>11: Immediately and unconditionally, force inputs to their not asserted state, and park (force) outputs low.</p>

## GLPCI\_GLD\_MSR\_PM Bit Descriptions (Continued)

Bit	Name	Description
47:35	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
34:32	RSVD	<b>Reserved.</b> Write as read.
31:6	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
5:4	PMODE2	<b>Power Mode 2.</b> Power mode for PCI-fast clock domain. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
3:2	PMODE1	<b>Power Mode 1.</b> Power mode for PCI clock domain. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
1:0	PMODE0	<b>Power Mode 0.</b> Power mode for GLIU clock domain. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.

## 6.2.1.6 GLD Diagnostic MSR (GLPCI\_GLD\_MSR\_DIAG)

MSR Address 51000005h  
Type R/W  
Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.2.2 GLPCI\_SB Specific MSRs

## 6.2.2.1 Global Control (GLPCI\_CTRL)

MSR Address 51000010h  
Type R/W  
Reset Value 44000030\_00000003h

## GLPCI\_CTRL Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
FTH			RTH				RSVD					RTL				RSVD				SLTO	ILTO		LAT			0	0	0			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								SUS		RSVD	FPIDE	PPIDE	LRH	RDHP	PSIDE	RPIDE	LEGACT	SDOFF	HCD	IOED	RSVD			CISM		OD	IE	ME			

## GLPCI\_CTRL Bit Descriptions

Bit	Name	IB/OB	Description
63:60	FTH	IB	<b>In-Bound Flush Threshold.</b> Controls the timing for requesting new read data while concurrently flushing previously prefetched, stale read data. While flushing stale data, if the number of prefetched 64-bit WORDs reaches this level, then a new read request is made.
59:56	RTH	IB	<b>In-Bound Read Threshold.</b> Controls the timing for prefetching read data. If the number of prefetched 32-bit WORDs is decremented and reaches this threshold, a subsequent GLIU request is generated to fetch the next cache-line of read data.
55:52	RSVD (RO)	---	<b>Reserved (Read Only).</b> Returns 0.
51:49	RTL	OB	<b>Retry Transaction Limit.</b> Limits the number of out-bound retries. If a target signals retry indefinitely the PCI interface may be configured to abort the failing out-bound request. 000: No limit. 100: 64 retries. 001: 8 retries. 101: 128 retries. 010: 16 retries. 110: 256 retries. 011: 32 retries. 111: 512 retries.
48:43	RSVD (RO)	---	<b>Reserved (Read Only).</b> Returns 0.
42	SLTO	IB	<b>Subsequent Latency Time-Out Select.</b> Specifies the subsequent target latency time-out limit. If, within a burst, the GLPCI_SB module does not respond with the configured number of clock edges the PCI interface terminates the PCI bus cycle. 0: 8 PCI clock edges. 1: 4 PCI clock edges.
41:40	ILTO	IB	<b>Initial Latency Time-out Select.</b> Specifies the initial target latency time-out limit for the PCI interface. If the GLPCI_SB module does not respond with the first data phase within the configured number of clock edges the PCI interface terminates the PCI bus cycle. 00: 32 PCI clock edges. 10: 8 PCI clock edges. 01: 16 PCI clock edges. 11: 4 PCI clock edges.
39:35	LAT	IB/OB	<b>PCI Usage Timer.</b> Usage time-out value for limiting bus tenure.
34:32	0 (RO)	IB/OB	<b>Constant 0 (Read Only).</b> The three least significant bits of the PCI latency timer field are fixed as zeros. These bits are not used as part of the PCI latency timer comparison.
31:24	RSVD (RO)	---	<b>Reserved (Read Only).</b> Returns 0.
23:21	SUS	IB/OB	<b>Busy Sustain.</b> Controls the sustain time for keeping the clocks running after the internal busy signals indicate that the clocks may be gated. 000: No sustain. 100: 32 clock cycles. 001: 4 clock cycles. 101: 64 clock cycles. 010: 8 clock cycles. 110: 128 clock cycles. 011: 16 clock cycles. 111: 256 clock cycles.
20	RSVD (RO)	---	<b>Reserved (Read Only).</b> Returns 0.

## GLPCI\_CTRL Bit Descriptions (Continued)

Bit	Name	IB/OB	Description
19:18	FPIDE	IB	<p><b>Prefetch Primary IDE.</b> If these bits are set, I/O reads to address 1F0h conform to a prefetching behavior. Under this mode, the GLPCI_SB issues GLIU Read Request Packets for this specific address before receiving a request on the PCI bus for it. When IDE prefetch is enabled, all PCI accesses to 1F0h must be DWORDs; that is, 4 bytes. This setting can only be changed between PIO operations.</p> <p>00: Off. (Default)  01: At “beginning” initialize pipeline with two read requests.  10: At “beginning” initialize pipeline with three read requests.  11: Reserved.</p> <p>The prefetch only applies if the current command is "read". The current command is assumed from the last write to IDE Command Register at 1F7h. The following commands are considered "reads":</p> <p>Read sectors - 20h  Read multiple - C4h  Read buffer - E4h</p> <p>Prefetch does not cross sector boundaries; that is, 512-byte boundaries. Any prefetched data is discarded and the “boundary” set to 0 on any write to 1F7h.</p>
17	PPIDE	IB	<p><b>Post Primary IDE.</b> Defaults to 0. If this bit is set, I/O writes to address 1F0h are posted; that is, the “send response” flag is not set in the GLIU Write Request Packet. Effectively, an I/O write to this specific address is posted just like memory writes are posted. When IDE posting is enabled, single and double WORD writes may be mixed without restriction.</p>
16	LRH	IB	<p><b>Legacy I/O Retry/Hold.</b></p> <p>0: Legacy I/O retry.  1: Legacy I/O hold.</p> <p>Regardless of the above settings an I/O read or write to 1F0h always causes a retry if data can not be immediately transferred.</p>
15	RDHP	IB	<p><b>Reject DMA High Page.</b> Controls the decoding of I/O range associated with the DMA High Page registers (480h-48Fh).</p> <p>0: Considered part of legacy I/O.  1: Subtractive decode.</p>
14	RSIDE	IB	<p><b>Reject Secondary IDE.</b> Controls the decoding of I/O range associated with Secondary IDE address of 170h-177h and 376h.</p> <p>0: Considered part of legacy I/O.  1: Subtractive decode.</p>
13	RPIDE	IB	<p><b>Reject Primary IDE.</b> Controls the decoding of I/O range associated with Primary IDE address of 1F0h-1F7h and 3F6h.</p> <p>0: Considered part of legacy I/O.  1: Subtractive decode.</p>
12:11	LEGACT	IB	<p><b>Legacy I/O Space Active Decode.</b></p> <p>00: Subtractive decode (claim on fourth clock).  01: Slow decode (claim on third clock).  10: Medium decode (Claim on second clock).  11: Reserved (implemented as medium decode and returned 10 when read).</p>
10	SDOFF	OB	<p><b>Non Legacy Subtractive Decode Off.</b></p> <p>0: Subtractive decode enabled.  1: Subtractive decode disabled.</p>

## GLPCI\_CTRL Bit Descriptions (Continued)

Bit	Name	IB/OB	Description
9	HCD	IB	<b>Hold for CIS Transfer Disable.</b> 0: Hold for CIS transfer enabled. 1: Hold for CIS transfer disabled.
8	IOED	IB	<b>I/O Addressing Error Checking Disable.</b> 0: I/O addressing error checking enabled. 1: I/O addressing error checking disabled.
7:5	RSVD (RO)	---	<b>Reserved (Read Only).</b> Returns 0.
4:3	CISM	IB/OB	<b>CIS Mode.</b> 00: Mode A (Default). Not used in normal operation. 01: Mode B. Not used in normal operation. 10: Mode C. Used in normal operation. 11: Reserved.  See Section 5.2.14 "CPU Interface Serial (CIS)" on page 87 for details regarding operation modes.
2	OD	OB	<b>Out-Bound Request Disable.</b> 0: Out-bound request enabled. 1: Out-bound request disabled.  When an out-bound request is disabled, all outstanding out-bound requests are serviced before a read response packet with SSMI bit and all data bits cleared and EXCEP bit set is returned.
1	IE	IB	<b>I/O Enable.</b> Enable handling of in-bound I/O transactions from PCI. When set to 1 the PCI interface accepts all in-bound I/O transactions from PCI. This mode is only intended for design verification purposes. When cleared to 0 no in-bound I/O transactions are accepted.
0	ME	IB	<b>Memory Enable.</b> Enable handling of in-bound memory access transaction from PCI. When cleared to 0, the PCI interface does not accept any in-bound memory transactions from the PCI bus.

6.2.2.2 Region 0-15 Configuration MSRs (GLPCI\_R[x])

**Region 0 Configuration (GLPCI\_R0)**

MSR Address 51000020h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 8 Configuration (GLPCI\_R8)**

MSR Address 51000028h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 1 Configuration (GLPCI\_R1)**

MSR Address 51000021h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 9 Configuration (GLPCI\_R9)**

MSR Address 51000029h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 2 Configuration (GLPCI\_R2)**

MSR Address 51000022h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 10 Configuration (GLPCI\_R10)**

MSR Address 5100002Ah  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 3 Configuration (GLPCI\_R3)**

MSR Address 51000023h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 11 Configuration (GLPCI\_R11)**

MSR Address 5100002Bh  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 4 Configuration (GLPCI\_R4)**

MSR Address 51000024h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 12 Configuration (GLPCI\_R12)**

MSR Address 5100002Ch  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 5 Configuration (GLPCI\_R5)**

MSR Address 51000025h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 13 Configuration (GLPCI\_R13)**

MSR Address 5100002Dh  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 6 Configuration (GLPCI\_R6)**

MSR Address 51000026h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 14 Configuration (GLPCI\_R14)**

MSR Address 5100002Eh  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 7 Configuration (GLPCI\_R7)**

MSR Address 51000027h  
 Type R/W  
 Reset Value 00000000\_00000000h

**Region 15 Configuration (GLPCI\_R15)**

MSR Address 5100002Fh  
 Type R/W  
 Reset Value 00000000\_00000000h

GLPCI\_R[x] Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
TOP																	RSVD										SPACE				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE																	RSVD							PF	RSVD	RH	EN				

## GLPCI\_REGCONF[x] Bit Descriptions

Bit	Name	Description
63:44	TOP	<b>Top of Region.</b> For memory use [63:44] as top of address bits [31:12]. For I/O use [63:46] as top of address bits [19:2]. (Note 1)
43:33	RSVD (RO)	<b>Reserved (Read Only):</b> Returns 0.
32	SPACE	<b>Region Space Indicator.</b> 0: Memory space. 1: I/O space.
31:12	BASE	<b>Base of Region.</b> For memory use [31:12] as base of address bits [31:12]. For I/O use [31:14] as base of address bits [19:2]. (Note 1)
11:4	RSVD (RO)	<b>Reserved (Read Only):</b> Returns 0.
3	PF	<b>Prefetchable.</b> If region is memory and this bit is set, it indicates a prefetchable memory region. Reads to this region have no side-effects. If region is I/O and this bit is set, post all I/O writes to this region.
2	RSVD (RO)	<b>Reserved (Read Only):</b> Returns 0.
1	RH	<b>Retry/Hold.</b> Defines whether GLPCI_SB PCI slave generates a retry condition or holds the PCI bus until cycle completion. Note that even if hold is selected, the cycle will be terminated if initial latency time-out is reached. 0: Retry. 1: Hold.
0	EN	<b>Region Enable.</b> Set to 1 to enable access to this region.

Note 1. For memory, 4 KB granularity, inclusive: [63:44] <= address[31:12] <= [31:12].  
For I/O, 4B granularity, inclusive: [63:46] <= address[19:2] <= [31:14].

### 6.2.2.3 PCI Configuration Space Header Byte 0-3 (GLPCI\_PCIHEAD\_BYTE0-3)

MSR Address 51000030h  
Type RO  
Reset Value 00000000\_002A100Bh

Reads back the value of PCI Configuration Space Header Byte 0-3 (GLPCI\_PCI\_HEAD\_BYTE0-3). See Section 6.2.3.1 on page 240 for register map and bit definitions.

### 6.2.2.4 PCI Configuration Space Header Byte 4-7 (GLPCI\_PCIHEAD\_BYTE4-7)

MSR Address 51000031h  
Type RO  
Reset Value 00000000\_00000000h

Reads back the value of PCI Configuration Space Header Byte 4-7 (GLPCI\_PCI\_HEAD\_BYTE4-7). See Section 6.2.3.2 on page 241 for register map and bit definitions.

### 6.2.2.5 PCI Configuration Space Header Byte 8-B (GLPCI\_PCIHEAD\_BYTE8-B)

MSR Address 51000032h  
Type RO  
Reset Value 00000000\_00000000h

Reads back the value of PCI Configuration Space Header Byte 8-B (GLPCI\_PCI\_HEAD\_BYTE8-B). See Section 6.2.3.3 on page 241 for register map and bit definitions.

### 6.2.2.6 PCI Configuration Space Header Byte C-F (GLPCI\_PCIHEAD\_BYTEC-F)

MSR Address	51000033h
Type	RO
Reset Value	00000000_00000000h

Reads back the value of PCI Configuration Space Header Byte C-F (GLPCI\_PCI\_HEAD\_BYTEC-F). See Section 6.2.3.4 on page 242 for register map and bit definitions.

### 6.2.3 PCI Configuration Registers

The first 16 bytes of the PCI configuration register space consist of standard PCI header registers. An additional 32 bytes are used to implement a mailbox for giving access from the PCI bus to the internal MSR of the Geode CS5535 companion device.

#### MSR Access Mailbox

Upon reset, MSR access is enabled. That is, the PMC-TRL.EN bit is set. A PCI configuration (config) write to register F0h clearing the EN bit is required to disable MSR access.

An MSR read is accomplished by:

- A PCI configuration write to register F4h (PMADDR) with the appropriate address value. If the appropriate address value was previously written to register F4h, then this step is unnecessary.
- A PCI configuration read of register F8h (PMDATA0). This starts the GLIU MSR read. The PCI bus is held (i.e., no retry unless time-out) until the transaction completes.

- A PCI configuration read of register FCh (PMDATA1). The PCI bus is held (i.e., no retry unless time-out) until the transaction completes.

An MSR write is accomplished by:

- A PCI configuration write to register F4h (PMADDR) with the appropriate address value. If the appropriate address value was previously written to register F4h, then this step is unnecessary.
- A PCI configuration write to register F8h (PMDATA0).
- A PCI configuration write to register FCh (PMDATA1). This starts the GLIU MSR write. The PCI bus is held (i.e., no retry unless time-out) until the transaction completes.

Any PCI transaction interrupting an MSR read/write transaction is retried until the MSR transaction is complete.

The external MSR write request always has the SEND\_RESPONSE bit set. The returned MSR read or write response packet is checked for the SSMI and EXCEP bits.

### 6.2.3.1 PCI Configuration Space Header Byte 0-3 (GLPCI\_PCI\_HEAD\_BYTE0-3)

PCI Index	00h
Type	RO
Reset Value	002A100Bh

#### GLPCI\_PCI\_HEAD\_BYTE0-3 Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_ID																VEN_ID															

#### GLPCI\_PCI\_HEAD\_BYTE0-3 Bit Descriptions

Bit	Name	Description
31:16	DEV_ID	<b>Device Identification Register (Read Only).</b> Identifies Geode CS5535 companion as the device. Reads as 002Ah.
15:0	VEN_ID	<b>Vendor Identification Register (Read Only).</b> Identifies AMD as the vendor. Reads as 100Bh.



**6.2.3.2 PCI Configuration Space Header Byte 4-7 (GLPCI\_PCI\_HEAD\_BYTE4-7)**

PCI Index 04h  
 Type RO  
 Reset Value 00000000h

**GLPCI\_PCI\_HEAD\_BYTE4-7 Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCI_STS																PCI_CMD															

**GLPCI\_PCI\_HEAD\_BYTE4-7 Bit Descriptions**

Bit	Name	Description
31:16	PCI_STS	<b>PCI Status Register (Read Only).</b> Not implemented.
15:0	PCI_CMD	<b>PCI Command Register (Read Only).</b> Not implemented.

**6.2.3.3 PCI Configuration Space Header Byte 8-B (GLPCI\_PCI\_HEAD\_BYTE8-B)**

PCI Index 08h  
 Type RO  
 Reset Value FF0000xxh

**GLPCI\_PCI\_HEAD\_BYTE8-B Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCI_CLASS																								DEV_REV_ID							

**GLPCI\_PCI\_HEAD\_BYTE8-B Bit Descriptions**

Bit	Name	Description
31:16	PCI_CLASS	<b>PCI Class Code (Read Only).</b>
15:0	DEV_REV_ID	<b>Device Revision ID (Read Only).</b> Identifies the major and minor silicon revision of the Geode CS5535 companion device. Can also be read at MSR 51700017h[7:0]. See Section 6.20.2.13 "Chip Revision ID (GLCP_CHIP_REV_ID)" on page 543.

**6.2.3.4 PCI Configuration Space Header Byte C-F (GLPCI\_PCI\_HEAD\_BYTEC-F)**

PCI Index           0Ch  
 Type                RO  
 Reset Value        00000000h

**GLPCI\_PCI\_HEAD\_BYTEC-F Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCI_BIST								PCI_HEADER								PCI_LTNCY_TMR								PCI_CACHE							

**GLPCI\_PCI\_HEAD\_BYTEC-F Bit Descriptions**

Bit	Name	Description
31:24	PCI_BIST	<b>PCI BIST Register (Read Only).</b> Not implemented.
23:16	PCI_HEADER	<b>PCI Header Type Byte (Read Only).</b> This register defines the format of this header. This header is of type format 0, that is, this byte contains all zeroes.
16:8	PCI_LTNCY_TMR	<b>PCI Latency Timer Register (Read Only).</b> Not implemented. Writing these bits have no effect.
7:0	PCI_CACHE	<b>PCI Cache Line Size Register (Read Only).</b> Not implemented. Writing these bits have no effect.

**6.2.3.5 PCI MSR Control (GLPCI\_PMCTRL)**

PCI Index           F0h  
 Type                R/W  
 Reset Value        00000001h

**GLPCI\_PMCTRL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															MSR_EN

**GLPCI\_PMCTRL Bit Descriptions**

Bit	Name	Description
31:1	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
0	MSR_EN	<b>MSR Enable.</b> Set to 1 to enable access to Model Specific Registers (MSRs).

**6.2.3.6 PCI MSR Address (GLPCI\_PMADDR)**

PCI Index	F4h
Type	R/W
Reset Value	00000000h

**GLPCI\_PMADDR Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																															

**GLPCI\_PMADDR Bit Descriptions**

Bit	Name	Description
31:0	ADDRESS	<b>MSR Address.</b> Address field to use in GLIU MSR accessing. Addresses with the most significant 18 bits set to zero address the model specific registers of the GLPCI_SB module itself. If any of the 18 most significant bits are set to one, the GLPCI_SB forwards the MSR access to the GLIU without performing any address translation.

**6.2.3.7 PCI MSR Data 0 (GLPCI\_PMDATA0)**

PCI Index	F8h
Type	R/W
Reset Value	00000000h

**GLPCI\_PMDATA0 Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0																															

**GLPCI\_PMDATA0 Bit Descriptions**

Bit	Name	Description
31:0	DATA0	<b>MSR Data 0.</b> Least significant 32-bits of MSR data. DATA0 and DATA1 R/W are atomic in nature (i.e., if DATA0 access is made in GLPCI_SB then it must followed by DATA1 access). Until the DATA1 access, the GLPCI_SB retries all other transactions on the PCI bus for $2^{15}$ cycles. After the timeout expires, atomic nature of DATA0 and DATA1 expires and other transactions are accepted.

**6.2.3.8 PCI MSR Data 1 (GLPCI\_PMDATA1)**

PCI Index FCh  
 Type R/W  
 Reset Value 00000000h

**GLPCI\_PMDATA1 Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1																															

**GLPCI\_PMDATA1 Bit Descriptions**

Bit	Name	Description
31:0	DATA1	<b>MSR Data 1.</b> Most significant 32-bits of MSR data. DATA0 and DATA1 R/W are atomic in nature (i.e., if DATA0 access is made in GLPCI_SB then it must followed by DATA1 access). Until the DATA1 access, GLPCI_SB will retry all other transactions on PCI bus for 2 <sup>15</sup> cycles. After this timeout, atomic nature for DATA0 and DATA1 expires and other transactions are accepted.

### 6.3 AC97 Audio Codec Controller Register Descriptions

The control registers for the AC97 Audio Codec Controller (ACC) are divided into two register sets:

- Standard GeodeLink™ Device (GLD) MSRs
- ACC Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The ACC Native registers begin at ACC Offset 00h. The system automatically maps the ACC registers to a location in memory space or I/O space, but this is hidden from the module's point of view. At the audio block level, it does not matter if these registers are in memory or I/O space but at the system level, there are significant operational differences (see Section "Eliminating Race Conditions") Hereafter the ACC Address are called out as I/O Offsets, since I/O mapping is recommended.

For Native register access, only the lower seven bits of the address are decoded, so the register space is aliased. Accesses beyond 7Fh alias below 7Fh. Accesses to addresses that are not implemented or reserved are "don't cares" (i.e. writes do nothing, reads return 0s).

Tables 6-8 and 6-9 are ACC register summary tables that include reset values and page references where the bit descriptions are provided.

#### Eliminating Race Conditions

All I/O writes are sequence locked, that is, completion of the write at the target is confirmed before the executing processor proceeds to the next instruction. All memory writes are posted, that is, the executing processor proceeds to the next instruction immediately after the write whether or not the write has completed. Write posting can lead to out of order execution. Reading the register to which a write has been posted, forces any pending posted write to execute if it has not already done so.

Consider this example. Assume an audio master is performing an access to system memory and register access is temporarily blocked. If the processor was servicing an interrupt, a write to clear the interrupt would post to a memory mapped register but not execute immediately, that is, the interrupt would not immediately clear. If the processor then enabled the Programmable Interrupt Controller (PIC) for new interrupts, then the "not immediately cleared" interrupt would cause a false new interrupt, a form of a race condition.

This type of race condition can be eliminated by placing the audio registers in I/O space, or, by performing a register read to any register having a pending posted write that is capable of creating a race condition.

**Table 6-8. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51500000h	RO	GLD Capabilities MSR (ACC_GLD_MSR_CAP)	00000000_002330xxh	Page 248
51500001h	R/W	GLD Master Configuration MSR (ACC_GLD_MSR_CONFIG)	00000000_0000F000h	Page 248
51500002h	R/W	GLD SMI MSR (ACC_GLD_MSR_SMI)	00000000_00000000h	Page 249
51500003h	R/W	GLD Error MSR (ACC_GLD_MSR_ERROR)	00000000_00000000h	Page 250
51500004h	R/W	GLD Power Management MSR (ACC_GLD_MSR_PM)	00000000_00000000h	Page 251
51500005h	R/W	GLD Diagnostic MSR (ACC_GLD_MSR_DIAG)	00000000_00000000h	Page 251

Table 6-9. ACC Native Registers Summary

ACC I/O Offset	Type	Width (Bits)	Name	Reset Value	Reference
00h	R/W	32	Codec GPIO Status Register (ACC_GPIO_STATUS)	00000000h	Page 252
04h	R/W	32	Codec GPIO Control Register (ACC_GPIO_CNTL)	00000000h	Page 253
08h	R/W	32	Codec Status Register (ACC_CODEEC_STATUS)	00000000h	Page 253
0Ch	R/W	32	Codec Control Register (ACC_CODEEC_CNTL)	00000000h	Page 254
10h-11h	---	---	Not Used	---	---
12h	RO	16	Second Level Audio IRQ Status Register (ACC_IRQ_STATUS)	00000000h	Page 256
14h	R/W	32	Bus Master Engine Control Register (ACC_ENGINE_CNTL)	00000000h	Page 257
18h-1Fh	---	---	Not Used	---	---
20h	R/W	8	Bus Master 0 Command (ACC_BM0_CMD)	00h	Page 258
21h	RC	8	Bus Master 0 IRQ Status (ACC_BM0_STATUS)	00h	Page 259
22h-23h	---	---	Not Used	---	---
24h	R/W	32	Bus Master 0 PRD Table Address (ACC_BM0_PRD)	00000000h	Page 260
28h	R/W	8	Bus Master 1 Command (ACC_BM1_CMD)	08h	Page 258
29h	RC	8	Bus Master 1 IRQ Status (ACC_BM1_STATUS)	00h	Page 259
2Ah-2Bh	---	---	Not Used	---	---
2Ch	R/W	32	Bus Master 1 PRD Table Address (ACC_BM1_PRD)	00000000h	Page 260
30h	R/W	8	Bus Master 2 Command (ACC_BM2_CMD)	00h	Page 258
31h	RC	8	Bus Master 2 IRQ Status (ACC_BM2_STATUS)	00h	Page 259
32h-33h	---	---	Not Used	---	---
34h	R/W	32	Bus Master 2 PRD Table Address (ACC_BM2_PRD)	00000000h	Page 260
38h	R/W	8	Bus Master 3 Command (ACC_BM3_CMD)	08h	Page 258
39h	RC	8	Bus Master 3 IRQ Status (ACC_BM3_STATUS)	00h	Page 259
3Ah-3Bh	---	---	Not Used	---	---
3Ch	R/W	32	Bus Master 3 PRD Table Address (ACC_BM3_PRD)	00000000h	Page 260
40h	R/W	8	Bus Master 4 Command (ACC_BM4_CMD)	00h	Page 258

Table 6-9. ACC Native Registers Summary

ACC I/O Offset	Type	Width (Bits)	Name	Reset Value	Reference
41h	RC	8	Bus Master 4 IRQ Status (ACC_BM4_STATUS)	00h	Page 259
42h-43h	---	---	Not Used	---	---
44h	R/W	32	Bus Master 4 PRD Table Address (ACC_BM4_PRD)	00000000h	Page 260
48h	R/W	8	Bus Master 5 Command (ACC_BM5_CMD)	08h	Page 258
49h	RC	8	Bus Master 5 IRQ Status (ACC_BM5_STATUS)	00h	Page 259
4Ah-4Bh	---	---	Not Used	---	---
4Ch	R/W	32	Bus Master 5 PRD Table Address (ACC_BM5_PRD)	00000000h	Page 260
50h	R/W	8	Bus Master 6 Command (ACC_BM6_CMD)	00h	Page 258
51h	RC	8	Bus Master 6 IRQ Status (ACC_BM6_STATUS)	00h	Page 259
52h-53h	---	---	Not Used	---	---
54h	R/W	32	Bus Master 6 PRD Table Address (ACC_BM6_PRD)	00000000h	Page 260
58h	R/W	8	Bus Master 7 Command (ACC_BM7_CMD)	00h	Page 258
59h	RC	8	Bus Master 7 IRQ Status (ACC_BM7_STATUS)	00h	Page 259
5Ah-5Bh	---	---	Not Used	---	---
5Ch	R/W	32	Bus Master 7 PRD Table Address (ACC_BM7_PRD)	00000000h	Page 260
60h	RO	32	Bus Master 0 DMA Pointer (ACC_BM0_PNTR)	00000000h	Page 261
64h	RO	32	Bus Master 1 DMA Pointer (ACC_BM1_PNTR)	00000000h	Page 261
68h	RO	32	Bus Master 2 DMA Pointer (ACC_BM2_PNTR)	00000000h	Page 261
6Ch	RO	32	Bus Master 3 DMA Pointer (ACC_BM3_PNTR)	00000000h	Page 261
70h	RO	32	Bus Master 4 DMA Pointer (ACC_BM4_PNTR)	00000000h	Page 261
74h	RO	32	Bus Master 5 DMA Pointer (ACC_BM5_PNTR)	00000000h	Page 261
78h	RO	32	Bus Master 6 DMA Pointer (ACC_BM6_PNTR)	00000000h	Page 261
7Ch	RO	32	Bus Master 7 DMA Pointer (ACC_BM7_PNTR)	00000000h	Page 261

### 6.3.1 Standard GeodeLink™ Device (GLD) MSRs

#### 6.3.1.1 GLD Capabilities MSR (ACC\_GLD\_MSR\_CAP)

MSR Address 51500000h  
 Type RO  
 Reset Value 00000000\_002330xxh

**ACC\_GLD\_MSR\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID																REV_ID							

**ACC\_GLD\_MSR\_CAP Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads return 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2330h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

#### 6.3.1.2 GLD Master Configuration MSR (ACC\_GLD\_MSR\_CONFIG)

MSR Address 51500001h  
 Type R/W  
 Reset Value 00000000\_0000F000h

**ACC\_GLD\_MSR\_CONFIG Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												PREFETCH_SEL	PREFETCH		DISCARD	NON_COH_WR	NON_COH_RD	RSVD						PRI		RSVD	PID				

**ACC\_GLD\_MSR\_CONFIG Bit Descriptions**

Bit	Name	Description
63:20	RSVD	<b>Reserved.</b> Reads return 0.
19	PREFETCH_SEL	<b>Select Flexible Prefetch Policy.</b> 0: Fixed read prefetch policy is selected. (Default) 1: The ACC establishes prefetch policy.
18:16	FIX_PREFETCH	<b>Fixed Read Prefetch Policy.</b> 000: None. Each read takes a complete trip to memory. 001: Initial read 08 bytes. Read next 8 only when requested. 010: Initial read 16 bytes. Read next 16 only when requested. 011: Initial read 32 bytes. Read next 32 only when requested. 100: Initial read 32 bytes. Read next 32 when 16 bytes left. 101, 110, and 111: Reserved.



**ACC\_GLD\_MSR\_CONFIG Bit Descriptions (Continued)**

Bit	Name	Description
15:14	DISCARD	<b>Read Prefetch Discard Policy.</b> 00: Reserved. 01: Discard all data not taken under current local bus grant. 10: Discard all data on any local bus transaction. 11: Discard all data on any local bus write transaction. Always use this value.
13	NON_COH_WR	<b>Non-Coherent Write.</b> 0: Write requests are coherent. 1: Write requests are non-coherent. Always use this value.
12	NON_COH_RD	<b>Non-Coherent Read.</b> 0: Read requests are coherent. 1: Read requests are non-coherent. Always use this value.
11:7	RSVD	<b>Reserved.</b> Reads as 0.
6:4	PRI	<b>Priority Level.</b> Always write 0.
3	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
2:0	PID	<b>Priority ID.</b> Always write 0.

**6.3.1.3 GLD SMI MSR (ACC\_GLD\_MSR\_SMI)**

MSR Address 51500002h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further SMI/ASMI generation details.)

**ACC\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															IRQ_SSMI_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
RSVD																															IRQ_SSMI_EN

**ACC\_GLD\_MSR\_SMI Bit Descriptions**

Bit	Name	Description
63:33	RSVD	<b>Reserved.</b> Reads return 0.
32	IRQ_SSMI_FLAG	<b>IRQ SSMI Flag.</b> If high, records that an SSMI was generated because the ACC interrupt signal transitioned from 0 to 1. This bit is unaffected when the interrupt transitions from 1 to 0. Write 1 to clear; writing 0 has no effect. IRQ_SSMI_EN (bit 1) must be set to enable this event and set flag.
31:1	RSVD	<b>Reserved.</b> Reads return 0.
0	IRQ_SSMI_EN	<b>IRQ SSMI Enable.</b> Write 1 to enable IRQ_SSMI_FLAG (bit 32) and to allow the event to generate an SSMI.

**6.3.1.4 GLD Error MSR (ACC\_GLD\_MSR\_ERROR)**

MSR Address 51500003h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further details.)

**ACC\_GLD\_MSR\_ERROR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	UNEXP_TYPE_ERR_FLAG
RSVD																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	UNEXP_TYPE_ERR_EN
RSVD																																

## ACC\_GLD\_MSR\_ERROR Bit Descriptions

Bit	Name	Description
63:33	RSVD	<b>Reserved.</b> Reads return 0.
32	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Error Flag.</b> If high, records that an ERR was generated due to either an unexpected type event or a master response packet with the EXCEP bit set has been received. Write 1 to clear; writing 0 has no effect. UNEXP_TYPE_ERR_EN (bit 1) must be set to enable this event and set flag.
31:1	RSVD	<b>Reserved.</b> Reads return 0.
0	UNEXP_TYPE_ERR_EN	<b>Unexpected Type Error Enable.</b> Write 1 to enable UNEXP_TYPE_ERR_FLAG (bit 32) and to allow the event to generate an ERR.

## 6.3.1.5 GLD Power Management MSR (ACC\_GLD\_MSR\_PM)

MSR Address 51500004h  
 Type R/W  
 Reset Value 00000000\_00000000h

## ACC\_GLD\_MSR\_PM Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																										P	P				
																										MODE1	MODE0				

## ACC\_GLD\_MSR\_PM Bit Descriptions

Bit	Name	Description
63:62	RSVD	<b>Reserved.</b> Reads return value written.
61:32	RSVD	<b>Reserved.</b> Reads return 0.
31:4	RSVD	<b>Reserved.</b> Reads return 0.
3:2	PMODE1	<b>Power Mode 1.</b> Power mode for LBus clock. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever the LBus circuits are not busy. 10: Reserved. 11: Reserved.
1:0	PMODE0	<b>Power Mode 0.</b> Power mode for GLIU clock 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever the GLIU circuits are not busy. 10: Reserved. 11: Reserved.

## 6.3.1.6 GLD Diagnostic MSR (ACC\_GLD\_MSR\_DIAG)

MSR Address 51500005h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.3.2 ACC Native Registers

### 6.3.2.1 Codec GPIO Status Register (ACC\_GPIO\_STATUS)

ACC I/O Offset 00h  
 Type R/W  
 Reset Value 00000000h

**ACC\_GPIO\_STATUS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO_EN	INT_EN	WU_INT_EN	RSVD								INT_FLAG	WU_INT_FLAG	PIN_STS																		

**ACC\_GPIO\_STATUS Bit Descriptions**

Bit	Name	Description
31	GPIO_EN	<b>GPIO Enable.</b> This bit determines if the codec GPIO pin data is sent out in slot 12 of the serial output stream. 0: Send 0s and tag slot 12 as invalid. 1: Send GPIO pin data and tag slot valid.
30	INT_EN	<b>Codec GPIO Interrupt Enable.</b> Allow a codec GPIO interrupt to set the codec GPIO interrupt flag and generate an IRQ. 0: Disable. 1: Enable.  A GPIO interrupt is defined by serial data in slot 12, bit 0
29	WU_INT_EN	<b>Codec GPIO Wakeup Interrupt Enable.</b> Allow a codec GPIO wakeup interrupt to set the codec GPIO wakeup interrupt flag and generate an IRQ. 0: Disable. 1: Enable.  A codec GPIO wakeup interrupt is defined as a 0-to-1 transition of AC_S_IN or AC_S_IN2 while the codec is powered down. This bit can only be set after the codec(s) are powered down (See Audio Driver Power-up/down Programming Model on page 98).
28:22	RSVD	<b>Reserved.</b> Reads return 0.
21	INT_FLAG	<b>Codec GPIO Interrupt Flag (Read to Clear).</b> If the GPIO interrupt is enabled (bit 30 = 1) then this flag is set upon a codec GPIO interrupt event (serial data in slot 12, bit 0 = 1), and an IRQ is generated.
20	WU_INT_FLAG	<b>Codec GPIO Wakeup Interrupt Flag (Read to Clear).</b> If the GPIO wakeup interrupt is enabled (bit 29 = 1), then this flag is set when a GPIO wakeup interrupt occurs, and an IRQ is generated.
19:0	PIN_STS	<b>Codec GPIO Pin Status (Read Only).</b> This is the GPIO pin status that is received from the codec in slot 12 of the serial input stream. This is updated every time slot 12 of the input stream is tagged valid.  <b>Note:</b> All 20 bits of input slot 12 are visible in this register, including reserved bits within slot 12.

**6.3.2.2 Codec GPIO Control Register (ACC\_GPIO\_CNTL)**

ACC I/O Offset 04h  
 Type R/W  
 Reset Value 00000000h

**ACC\_GPIO\_CNTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												PIN_DATA																			

**ACC\_GPIO\_CNTL Bit Descriptions**

Bit	Name	Description
31:20	RSVD	<b>Reserved.</b> Reads return 0.
19:0	PIN_DATA	<b>Codec GPIO Pin Data.</b> This is the GPIO pin data that is sent to the codec in slot 12 of the serial output stream.  <b>Note:</b> All 20 bits of the output slot 12 are controllable through this register, even though some are reserved per the AC97 spec and should be set to zero.

**6.3.2.3 Codec Status Register (ACC\_CODEC\_STATUS)**

ACC I/O Offset 08h  
 Type R/W  
 Reset Value 00000000h

**ACC\_CODEC\_STATUS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STS_ADD								PRM_RDY_STS	SEC_RDY_STS	SDATAIN2_EN	BM5_SEL	BM4_SEL	RSVD	STS_NEW	RSVD	STS_DATA															

**ACC\_CODEC\_STATUS Bit Descriptions**

Bit	Name	Description
31:24	STS_ADD	<b>Codec Status Address (Read Only).</b> Address of the register for which status is being returned. This address comes from slot 1 bits [19:12] of the serial input stream.  <b>Note:</b> Bit 19 of slot 1 is reserved, but still observable by software.
23	PRM_RDY_STS	<b>Primary Codec Ready (Read Only).</b> Indicates the ready status of the primary codec (slot 0, bit 15). Software should not access the codec or enable any bus masters until this bit is set. This bit is cleared when the AC Link Shutdown bit is set in the Codec Control register (ACC I/O Offset 0Ch[18]).
22	SEC_RDY_STS	<b>Secondary Codec Ready (Read Only).</b> Indicates the ready status of the secondary codec (slot 0, bit 15). Software should not access the codec or enable any bus masters until this bit is set. This bit is cleared when the AC Link Shutdown bit is set in the Codec Control register (ACC I/O Offset 0Ch[18]).

## ACC\_CODEC\_STATUS Bit Descriptions (Continued)

Bit	Name	Description
21	SDATAIN2_EN	<b>Enable Second Serial Data Input (AC_S_IN2).</b> 0: Disable. 1: Enable.  For the second serial input to function, this bit must be set. This is functionally ANDed with the AC_S_IN2 port of the ACC. Often, it may be necessary to configure a corresponding I/O pin as an input on the chip containing the ACC.
20	BM5_SEL	<b>Audio Bus Master 5 AC97 Slot Select.</b> Selects the serial input slot for Audio Bus Master 5 to receive data. 0: Slot 6. 1: Slot 11.
19	BM4_SEL	<b>Audio Bus Master 4 AC97 Slot Select.</b> Selects slot for Audio Bus Master 4 to transmit data. 0: Slot 6. 1: Slot 11.
18	RSVD	<b>Reserved.</b> Reads return 0
17	STS_NEW	<b>Codec Status New (Read to Clear).</b> Indicates if the status data in bits [15:0] is new: 0: Not new. 1: New.  This bit is set by hardware after receiving valid codec status data in slot 2 of the input stream. Upon issuing a read to the codec registers, software should wait for this flag to indicate that the corresponding data has been returned.
16	RSVD	<b>Reserved.</b> Reads return 0.
15:0	STS_DATA	<b>Codec Status Data (Read Only).</b> This is the codec status data that is received from the codec in slot 2, bits [19:4] of the serial input stream. This is used for reading the contents of registers inside the AC97 codec.

## 6.3.2.4 Codec Control Register (ACC\_CODEC\_CNTL)

ACC I/O Offset    0Ch  
Type                R/W  
Reset Value        00000000h

Since this register could potentially be accessed by both an audio driver and a modem driver running at the same time, it is expected that all writes occur as atomic read-modify-write accesses.

## ACC\_CODEC\_CNTL Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW_CMD	CMD_ADD							COMM_SEL	PD_PRIM	PD_SEC	RSVD	LNK_SHTDWN	LNK_WRMIRST	CMD_NEW	CMD_DATA																

## ACC\_CODEC\_CNTL Bit Descriptions

Bit	Name	Description
31	RW_CMD	<p><b>Codec Read/Write Command.</b> This bit specifies a read or write operation targeting the AC97 codec's registers.</p> <p>0: Write. 1: Read.</p> <p>This bit determines whether slot 1, bit 19 of the serial output stream will be high or low.</p>
30:24	CMD_ADD	<p><b>Codec Command Address.</b> Address of the codec control register for which the command is being sent. This address goes in slot 1, bits [18:12] of the serial output stream. This is used for specifying the address of a register in the AC97 codec (for reading or writing).</p>
23:22	COMM_SEL	<p><b>Audio Codec Communication.</b> Selects which codec to communicate with (for register reads/writes):</p> <p>00: Codec 1 (Primary) 01: Codec 2 (Secondary) 10: Codec 3 11: Codec 4</p> <p>These bits determine output slot 0, bits [1:0]. When these bits are non-zero, bits [14:13] of output slot 0 must be set to zeros regardless of the validity of slot 1 and slot 2.</p>
21	PD_PRIM	<p><b>Power-down Semaphore for Primary Codec.</b> This bit is used by software in conjunction with bit 20 to coordinate the power-down of the two codecs. This bit is intended to be set by the audio driver to indicate to the modem driver that the audio codec has been prepared for power-down. Internally it does not control anything, and is simply a memory bit.</p>
20	PD_SEC	<p><b>Power-down Semaphore for Secondary Codec.</b> This bit is used by software in conjunction with bit 21 to coordinate the power-down of the two codecs. This bit is intended to be set by the modem driver to indicate to the audio driver that the modem codec has been prepared for power-down. Internally it does not control anything, and is simply a memory bit.</p>
19	RSVD	<p><b>Reserved.</b> Reads return 0.</p>
18	LNK_SHTDWN	<p><b>AC Link Shutdown.</b> Informs the Controller that the AC Link is being shutdown.</p> <p>This bit should be set at the same time that the codec power-down command is issued to the codec.</p> <p>Setting this bit also clears both Codec Ready bits in the Codec Status register (ACC I/O Offset 08h[23:22]).</p> <p>Issuing a warm reset via bit 17 clears this bit.</p> <p>If the codec has been powered off and back on, a warm reset is unnecessary, this bit should be cleared manually.</p>
17	LNK_WRM_RST	<p><b>AC Link Warm Reset.</b> Setting this bit initiates the AC Link/codec warm reset process. It is automatically cleared by hardware once the serial bit clock resumes. This should only be set when the codec(s) are powered down. Once set, software should then wait for "Codec Ready" before accessing the codec.</p>

## ACC\_CODEC\_CNTL Bit Descriptions (Continued)

Bit	Name	Description
16	CMD_NEW	<p><b>Codec Command New.</b> Indicates if the codec command in bits [31:22] (and [15:0] for writes) is new.</p> <p>0: Not new. 1: New.</p> <p>This bit is to be set by software when a new command is loaded. It is cleared by hardware when the command is sent to the codec. Software must wait for this bit to clear before loading another command.</p> <p>This bit can not be cleared by software. When the CODEC_CNTL register is written by software with bit 16 cleared, then bits [31:22] and [15:0] are unaffected. Thus, bit 16 is an "enable" allowing bits [31:22] and [15:0] to be changed.</p>
15:0	CMD_DATA	<p><b>Codec Command Data.</b> This is the command data being sent to the codec in slot 2, bits [19:12] of the serial output stream. This is used for writing data into one of the registers in the AC97 codec. The contents are only sent to the codec for write commands (bit [31] = 0). For reads slot 2, bits[19:12] are stuffed with 0s.</p>

## 6.3.2.5 Second Level Audio IRQ Status Register (ACC\_IRQ\_STATUS)

ACC I/O Offset 12h  
Type RO  
Reset Value 00000000h

## ACC\_IRQ\_STATUS Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BM7_IRQ_STS	BM6_IRQ_STS	BM5_IRQ_STS	BM4_IRQ_STS	BM3_IRQ_STS	BM2_IRQ_STS	BM1_IRQ_STS	BM0_IRQ_STS	WU_IRQ_STS	IRQ_STS

## ACC\_IRQ\_STATUS Bit Descriptions

Bit	Name	Description
15:10	RSVD	<b>Reserved.</b> Reads return 0.
9	BM7_IRQ_STS	<b>Audio Bus Master 7 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 7. Reading the Bus Master 7 IRQ Status Register clears this bit.
8	BM6_IRQ_STS	<b>Audio Bus Master 6 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 6. Reading the Bus Master 6 IRQ Status Register clears this bit.
7	BM5_IRQ_STS	<b>Audio Bus Master 5 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 5. Reading the Bus Master 5 IRQ Status Register clears this bit.
6	BM4_IRQ_STS	<b>Audio Bus Master 4 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 4. Reading the Bus Master 4 IRQ Status Register clears this bit.
5	BM3_IRQ_STS	<b>Audio Bus Master 3 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 3. Reading the Bus Master 3 IRQ Status Register clears this bit.



**ACC\_IRQ\_STATUS Bit Descriptions (Continued)**

Bit	Name	Description
4	BM2_IRQ_STS	<b>Audio Bus Master 2 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 2. Reading the Bus Master 2 IRQ Status Register clears this bit.
3	BM1_IRQ_STS	<b>Audio Bus Master 1 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 1. Reading the Bus Master 1 IRQ Status Register clears this bit.
2	BM0_IRQ_STS	<b>Audio Bus Master 0 IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by an event occurring on Audio Bus Master 0. Reading the Bus Master 0 IRQ Status Register clears this bit.
1	WU_IRQ_STS	<b>Codec GPIO Wakeup IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by a GPIO Wakeup Interrupt event (serial data in going high during power-down). Reading the Codec GPIO Status Register clears this bit.
0	IRQ_STS	<b>Codec GPIO IRQ Status.</b> If this bit is set, it indicates that an IRQ was caused by a GPIO event in the AC97 Codec (slot 12, bit 0). Reading the Codec GPIO Status Register clears this bit.

**6.3.2.6 Bus Master Engine Control Register (ACC\_ENGINE\_CNTL)**

ACC I/O Offset    14h  
 Type             R/W  
 Reset Value     00000000h

**ACC\_ENGINE\_CNTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															SSND_MODE

**ACC\_ENGINE\_CNTL Bit Descriptions**

Bit	Name	Description
31:1	RSVD	<b>Reserved.</b> Reads return 0.
0	SSND_MODE	<b>Surround Sound (5.1) Synchronization Mode.</b> Enables synchronization of Bus Masters 0, 4, 6, and 7. This bit should be set whenever playing back multi-channel surround sound. It ensures that the four bus masters stay synchronized and do not introduce any temporal skew between the separate channels.

### 6.3.2.7 Audio Bus Master 0-7 Command Registers (ACC\_BM[x]\_CMD)

#### Bus Master 0 Command (ACC\_BM0\_CMD)

ACC I/O Offset 20h  
Type R/W  
Reset Value 00h

#### Bus Master 4 Command (ACC\_BM4\_CMD)

ACC I/O Offset 40h  
Type R/W  
Reset Value 00h

#### Bus Master 1 Command (ACC\_BM1\_CMD)

ACC I/O Offset 28h  
Type R/W  
Reset Value 08h

#### Bus Master 5 Command (ACC\_BM5\_CMD)

ACC I/O Offset 48h  
Type R/W  
Reset Value 08h

#### Bus Master 2 Command (ACC\_BM2\_CMD)

ACC I/O Offset 30h  
Type R/W  
Reset Value 00h

#### Bus Master 6 Command (ACC\_BM6\_CMD)

ACC I/O Offset 50h  
Type R/W  
Reset Value 00h

#### Bus Master 3 Command (ACC\_BM3\_CMD)

ACC I/O Offset 38h  
Type R/W  
Reset Value 08h

#### Bus Master 7 Command (ACC\_BM7\_CMD)

ACC I/O Offset 58h  
Type R/W  
Reset Value 00h

### ACC\_BM[x]\_CMD Register Map

7	6	5	4	3	2	1	0
RSVD				RW	BYTE_ORD	BM_CTL	

### ACC\_BM[x]\_CMD Bit Descriptions

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Reads return 0
3	RW (RO)	<b>Read or Write (Read Only).</b> Indicates the transfer direction of the audio bus master. This bit always reads 0 for BM [0,2,4,6,7]. This bit always reads 1 for BM[1,3,5]. 0: Memory to codec. 1: Codec to memory.
2	BYTE_ORD	<b>Byte-Order.</b> Sets the byte order for 16-bit samples that this bus master uses. 0: Little Endian (Intel) byte-order (LSBs at lower address). 1: Big Endian (Motorola) byte-order (MSBs at lower address).
1:0	BM_CTL	<b>Bus Master Pause/Enable Control.</b> Enables, disables, or pauses the bus master. 00: Disable bus master. 01: Enable bus master. 10: Reserved. 11: Pause bus master (if currently enabled) or do nothing (if currently disabled). When the bus master is enabled by writing 01, the bus master starts up by using the address in its associated PRD Table Address Register. Writing 00 while the bus master is enabled causes the bus master to stop immediately. Upon resuming the bus master uses the address in its PRD Table Address Register. The PRD Table Address Register must be re-initialized by software before enabling the bus master, or there is a risk that the bus master may overstep the bounds of the PRD Table. <b>Note:</b> When the bus master reaches a PRD with the EOT bit set, these bits are set to 00.

## 6.3.2.8 Audio Bus Master 0-7 IRQ Status Registers (ACC\_BM[x]\_STATUS)

**Bus Master 0 IRQ Status (ACC\_BM0\_STATUS)**

ACC I/O Offset 21h  
 Type RC  
 Reset Value 00h

**Bus Master 4 IRQ Status (ACC\_BM4\_STATUS)**

ACC I/O Offset 41h  
 Type RC  
 Reset Value 00h

**Bus Master 1 IRQ Status (ACC\_BM1\_STATUS)**

ACC I/O Offset 29h  
 Type RC  
 Reset Value 00h

**Bus Master 5 IRQ Status (ACC\_BM5\_STATUS)**

ACC I/O Offset 49h  
 Type RC  
 Reset Value 00h

**Bus Master 2 IRQ Status (ACC\_BM2\_STATUS)**

ACC I/O Offset 31h  
 Type RC  
 Reset Value 00h

**Bus Master 6 IRQ Status (ACC\_BM6\_STATUS)**

ACC I/O Offset 51h  
 Type RC  
 Reset Value 00h

**Bus Master 3 IRQ Status (ACC\_BM3\_STATUS)**

ACC I/O Offset 39h  
 Type RC  
 Reset Value 00h

**Bus Master 7 IRQ Status (ACC\_BM7\_STATUS)**

ACC I/O Offset 59h  
 Type RC  
 Reset Value 00h

**ACC\_BM[x]\_STATUS Register Map**

7	6	5	4	3	2	1	0
RSVD						BM_EOP_ERR	EOP

**ACC\_BM[x]\_STATUS Bit Descriptions**

Bit	Name	Description
7:2	RSVD	<b>Reserved.</b> Reads return 0
1	BM_EOP_ERR	<b>Bus Master Error.</b> If this bit is set, it indicates that hardware encountered a second EOP before software cleared the first EOP.  If hardware encounters a second EOP (end of page) before software clears the first EOP, it causes the bus master to pause until this register is read to clear the error. Read to clear.
0	EOP	<b>End of Page.</b> If this bit is set, it indicates the bus master transferred data that is marked by the EOP bit in the PRD table (bit 30). Read to clear.

### 6.3.2.9 Audio Bus Master 7-0 PRD Table Address Registers (ACC\_BM[x]\_PRD)

#### Bus Master 0 PRD Table Address (ACC\_BM0\_PRD)

ACC I/O Offset 24h  
Type R/W  
Reset Value 00000000h

#### Bus Master 4 PRD Table Address (ACC\_BM4\_PRD)

ACC I/O Offset 44h  
Type R/W  
Reset Value 00000000h

#### Bus Master 1 PRD Table Address (ACC\_BM1\_PRD)

ACC I/O Offset 2Ch  
Type R/W  
Reset Value 00000000h

#### Bus Master 5 PRD Table Address (ACC\_BM5\_PRD)

ACC I/O Offset 4Ch  
Type R/W  
Reset Value 00000000h

#### Bus Master 2 PRD Table Address (ACC\_BM2\_PRD)

ACC I/O Offset 34h  
Type R/W  
Reset Value 00000000h

#### Bus Master 6 PRD Table Address (ACC\_BM6\_PRD)

ACC I/O Offset 54h  
Type R/W  
Reset Value 00000000h

#### Bus Master 3 PRD Table Address (ACC\_BM3\_PRD)

ACC I/O Offset 3Ch  
Type R/W  
Reset Value 00000000h

#### Bus Master 7 PRD Table Address (ACC\_BM7\_PRD)

ACC I/O Offset 5Ch  
Type R/W  
Reset Value 00000000h

### ACC\_BM[x]\_PRD Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRD_PNTR																	RSVD														

### ACC\_BM[x]\_PRD Bit Descriptions

Bit	Name	Description
31:2	PRD_PNTR	<p><b>Pointer to the Physical Region Descriptor Table.</b> This register is a PRD table pointer for Audio Bus Master [x].</p> <p>When written, this register points to the first entry in a PRD table. Once Audio Bus Master [x] is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register to the next PRD by adding 08h.</p> <p>When read, this register points to the next PRD.</p>
1:0	RSVD	<b>Reserved.</b> Reads return 0

**6.3.2.10 Bus Master 0-7 DMA Pointer Registers (ACC\_BM[x]\_PNTR)****Bus Master 0 DMA Pointer (ACC\_BM0\_PNTR)**

ACC I/O Offset 60h  
 Type RO  
 Reset Value 00000000h

**Bus Master 4 DMA Pointer (ACC\_BM4\_PNTR)**

ACC I/O Offset 70h  
 Type RO  
 Reset Value 00000000h

**Bus Master 1 DMA Pointer (ACC\_BM1\_PNTR)**

ACC I/O Offset 64h  
 Type RO  
 Reset Value 00000000h

**Bus Master 5 DMA Pointer (ACC\_BM5\_PNTR)**

ACC I/O Offset 74h  
 Type RO  
 Reset Value 00000000h

**Bus Master 2 DMA Pointer (ACC\_BM2\_PNTR)**

ACC I/O Offset 68h  
 Type RO  
 Reset Value 00000000h

**Bus Master 6 DMA Pointer (ACC\_BM6\_PNTR)**

ACC I/O Offset 78h  
 Type RO  
 Reset Value 00000000h

**Bus Master 3 DMA Pointer (ACC\_BM3\_PNTR)**

ACC I/O Offset 6Ch  
 Type RO  
 Reset Value 00000000h

**Bus Master 7 DMA Pointer (ACC\_BM7\_PNTR)**

ACC I/O Offset 7Ch  
 Type RO  
 Reset Value 00000000h

**ACC\_BM[x]\_PNTR Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_PNTR																															

**ACC\_BM[x]\_PNTR Bit Descriptions**

Bit	Name	Description
31:0	DMA_PNTR	<b>DMA Buffer Pointer.</b> Address of current sample being fetched (BM [0,2,4,6,7]) or written (BM [1,3,5]) by the DMA Bus Master [x].

## 6.4 ATA-5 Controller Register Descriptions

The control registers for the ATA-5 compatible IDE controller (ATAC) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs
- ATAC Specific MSRs
- ATAC Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the

perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The Native registers are accessed as I/O Offsets from a GLIU IOD Descriptor and are BYTE, WORD and DWORD accessible.

Tables 6-10 through 6-12 are ATAC register summary tables that include reset values and page references where the bit descriptions are provided.

**Table 6-10. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51300000h	RO	GLD Capabilities MSR (ATAC_GLD_MSR_CAP)	00000000_002470xxh	Page 263
51300001h	R/W	GLD Master Configuration MSR (ATAC_GLD_MSR_CONFIG)	00000000_0000F000h	Page 264
51300002h	R/W	GLD SMI MSR (ATAC_GLD_MSR_SMI)	00000000_00000000h	Page 265
51300003h	R/W	GLD Error MSR (ATAC_GLD_MSR_ERROR)	00000000_00000100h	Page 266
51300004h	R/W	GLD Power Management MSR (ATAC_GLD_MSR_PM)	00000000_00000000h	Page 267
51300005h	R/W	GLD Diagnostic MSR (ATAC_GLD_MSR_DIAG)	00000000_00000000h	Page 268

**Table 6-11. ATAC Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51300008h	R/W	I/O Base Address (ATAC_IO_BAR)	00000000_00000001h	Page 269
51300009h	---	Unused	---	---
51300010h	R/W	Reset Decode (ATAC_RESET)	00000000_00000000h	Page 269
51300011h- 5130001Fh	---	Unused	---	---
51300020h	R/W	Channel 0 Drive 0 PIO (ATAC_CH0D0_PIO)	00000000_00009172h	Page 270
51300021h	R/W	Channel 0 Drive 0 DMA (ATAC_CH0D0_DMA)	00000000_00077771h	Page 271
51300022h	R/W	Channel 0 Drive 1 PIO (ATAC_CH0D1_PIO)	00000000_00009172h	Page 273
51300023h	R/W	Channel 0 Drive 1 DMA (ATAC_CH0D1_DMA)	00000000_00077771h	Page 273
51300024h	R/W	PCI Abort Error (ATAC_PCI_ABRERR)	00000000_00000000h	Page 273

Table 6-12. ATAC Native Registers Summary

ATAC I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
00h	R/W	8	Bus Master 0 Command - Primary (ATAC_BM0_CMD_PRIM)	00h	Page 274
01h	---	---	Unused	---	---
02h	R/W	8	Bus Master 0 Status - Primary (ATAC_BM0_STS_PRIM)	00h	Page 274
03h	---	---	Unused	---	---
04h	R/W	32	Bus Master 0 PRD Table Address - Primary (ATAC_BM0_PRD)	00000000h	Page 275
08h-0Fh	---	---	Reserved. Write accesses are ignored, read accesses return 0.		

#### 6.4.1 Standard GeodeLink™ Device (GLD) MSRs

##### 6.4.1.1 GLD Capabilities MSR (ATAC\_GLD\_MSR\_CAP)

MSR Address 51300000h  
 Type RO  
 Reset Value 00000000\_002470xxh

##### ATAC\_GLD\_MSR\_CAP Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID																REV_ID							

##### ATAC\_GLD\_MSR\_CAP Bit Descriptions

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2470h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

### 6.4.1.2 GLD Master Configuration MSR (ATAC\_GLD\_MSR\_CONFIG)

MSR Address 51300001h  
 Type R/W  
 Reset Value 00000000\_0000F000h

#### ATAC\_GLD\_MSR\_CONFIG Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									PRG_PREFETCH			PREFETCH_SEL	FIX_PREFETCH			DISCARD		NON_COH_WR	NON_COH_RD	RSVD					PRI		RSVD		PID		

#### ATAC\_GLD\_MSR\_CONFIG Bit Descriptions

Bit	Name	Description
63:23	RSVD	<b>Reserved.</b> Reads as 0.
22:20	PRG_PREFETCH	<b>Programmable Prefetch Code.</b> When bit 19 is set, these three bits determine the prefetch policy for non-PRD read accesses. PRD read accesses use 001 setting. 000: No prefetch. 001: Prefetch 8 bytes. 010: Prefetch 16 bytes. 011: Prefetch 32 bytes. 100: Prefetch 32 bytes, additional 32 bytes prefetch after 16 bytes have been read. All other values are reserved.
19	PREFETCH_SEL	<b>Prefetch Select.</b> Select between fixed (when clear) or programmable prefetch policy codes (when set).
18:16	FIX_PREFETCH	<b>Fixed Prefetch Code.</b> When bit 19 is clear, these three bits determine the prefetch policy for all read accesses. 000: No prefetch. 001: Prefetch 8 bytes. 010: Prefetch 16 bytes. 011: Prefetch 32 bytes. 100: Prefetch 32 bytes, additional 32 bytes prefetch after 16 bytes have been read. All other values are reserved.
15:14	DISCARD	<b>Discard.</b> Read prefetch discard policy. 00: Reserved. 01: Discard all data not taken under current local bus grant. 10: Discard all data on any local bus transaction. 11: Discard all data on any local bus write transaction. Always use this value. (Default)
13	NON_COH_WR	<b>Non-Coherent Write.</b> 0: Write requests are coherent. 1: Write requests are non-coherent. Always use this value. (Default)
12	NON_COH_RD	<b>Non-Coherent Read.</b> 0: Read requests are coherent. 1: Read requests are non-coherent. Always use this value. (Default)
11:7	RSVD	<b>Reserved.</b> Reads as 0.
6:4	PRI	<b>Priority Level.</b> Always write 0.



### ATAC\_GLD\_MSR\_CONFIG Bit Descriptions (Continued)

Bit	Name	Description
3	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
2:0	PID	<b>Priority ID.</b> Always write 0.

#### 6.4.1.3 GLD SMI MSR (ATAC\_GLD\_MSR\_SMI)

MSR Address 51300002h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further details.)

### ATAC\_GLD\_MSR\_SMI Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32			
RSVD																																	IDE_CH0_IRQ_ASMI_FLAG	IDE_PIO_SSMI_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RSVD																																	IDE_CH0_IRQ_ASMI_EN	IDE_PIO_SSMI_EN

### ATAC\_GLD\_MSR\_SMI Bit Descriptions

Bit	Name	Description
63:34	RSVD	<b>Reserved.</b> Read returns 0.
33	IDE_CH0_IRQ_ASMI_FLAG	<b>IDE Channel 0 IRQ ASMI Flag.</b> If high, records that an ASMI was generated due to an IRQ event. Write 1 to clear; writing 0 has no effect. IDE_CH0_IRQ_ASMI_EN (bit 1) must be set to enable this event and set flag.
32	IDE_PIO_SSMI_FLAG	<b>IDE Programmable I/O SSMI Flag.</b> If high, records that an SSMI was generated due to a PIO transaction occurring during a DMA command. Write 1 to clear; writing 0 has no effect. IDE_PIO_SSMI_EN (bit 0) must be set to enable this event and set flag.
31:2	RSVD	<b>Reserved.</b> Read returns 0.
1	IDE_CH0_IRQ_ASMI_EN	<b>IDE Channel 0 IRQ ASMI Mask.</b> Write 1 to enable IDE_CH0_IRQ_ASMI_FLAG (bit 33) and to allow the event to generate an ASMI.
0	IDE_PIO_SSMI_EN	<b>IDE Programmable I/O SSMI Mask.</b> Write 1 to enable IDE_PIO_SSMI_FLAG (bit 32) and to allow the event to generate an SSMI.

#### 6.4.1.4 GLD Error MSR (ATAC\_GLD\_MSR\_ERROR)

MSR Address 51300003h  
 Type R/W  
 Reset Value 00000000\_00000100h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the FLAG; writing 0 has no effect.

**ATAC\_GLD\_MSR\_ERROR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
RSVD																															
RSVD																															
BLOCKIO																															
RSVD																															
IDE_PIO_ERR_FLAG																															
RESP_EXCEP_ERR_FLAG																															
SSMI_ERR_FLAG																															
BLOCKIO_SSMI_FLAG																															
UNEXP_TYPE_ERR_FLAG																															
IDE_PIO_ERR_EN																															
RESP_EXCEP_ERR_EN																															
SSMI_ERR_EN																															
BLOCKIO_SSMI_EN																															
UNEXP_TYPE_ERR_EN																															

**ATAC\_GLD\_MSR\_ERROR Bit Descriptions**

Bit	Name	Description
63:37	RSVD	<b>Reserved.</b> Write as read.
36	IDE_PIO_ERR_FLAG	<b>IDE PIO Error.</b> If high, records that an ERR was generated due to a PIO access during a DMA command. Write 1 to clear; writing 0 has no effect. IDE_PIO_ERR_EN (bit 4) must be high to generate ERR and set flag.  If IDE_PIO_ERR_EN = 1 and IDE_PIO_ASMI_EN (MSR 51300002h[0]) = 0 and the error occurs, this bit gets set, and the GLCP master error signal is asserted.  If both IDE_PIO_ERR_EN = 1 and IDE_PIO_ASMI_EN = 1 and the error occurs, this bit gets set, the GLCP master error signal is asserted, and an SSMI is generated.
35	RESP_EXCEP_ERR_FLAG	<b>Response Exception Error Flag.</b> If high, records that an ERR was generated and the GLCP master error signal was asserted due to the EXCEP bit being set in the response packet. Write 1 to clear; writing 0 has no effect. RESP_EXCEP_ERR_EN (bit 3) must be high to enable these events.
34	SSMI_ERR_FLAG	<b>SSMI Error Flag.</b> If high, records that an ERR was generated due to an uncleared SSMI. Write 1 to clear; writing 0 has no effect. SSMI_ERR_EN (bit 2) must be high to generate ERR and set flag.  If SSMI_ERR_EN and IDE_PIO_SSMI_FLAG (MSR 51300002h[32]) = 1 and the error occurs, this bit gets set, the GLCP master error signal is asserted, lb_slav_rdy is asserted, and an SSMI is generated.
33	BLOCKIO_SSMI_FLAG	<b>SSMI on I/O write during DMA.</b> If high, records that an I/O write during DMA had occurred. Write 1 to clear; writing 0 has no effect. BLOCKIO_SSMI_EN (bit 1) and BLOCKIO (bit 8) must be high to set flag.

## ATAC\_GLD\_MSR\_ERROR Bit Descriptions (Continued)

Bit	Name	Description
32	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Error Flag.</b> If high, records that ERR was generated and the GLCP master error signal was asserted due to an unexpected type occurring. Write 1 to clear; writing 0 has no effect. UNEXP_TYPE_ERR_FLAG (bit 0) must be high to generate ERR, set flag, and assert the GLCP master error signal. Once clear, the GLCP master error signal is de-asserted.
31:9	RSVD	<b>Reserved.</b> Write as read.
8	BLOCKIO	<b>IDE Device Register I/O in DMA Blocked.</b> When this bit is set, and if IDE_PIO_SSMI_EN (MSR 51300002h[0]) is 0 and if IDE_PIO_ERR_EN (bit 4) is 0, ignore all PIO writes during DMA, return 80h on all PIO reads during DMA, and generate an SSMI. The GLCP master error signal will not be asserted. When this bit is clear, IDE device register I/O during DMA will not be blocked.
7:5	RSVD	<b>Reserved.</b> Write as read.
4	IDE_PIO_ERR_EN	<b>IDE PIO Error Enable.</b> Write 1 to enable IDE_PIO_ERR_FLAG (bit 36) and to allow a PIO access during a DMA command event to generate an ERR.
3	RESP_EXCEP_ERR_EN	<b>Response Exception Enable.</b> Write 1 to enable RESP_EXCEP_ERR_FLAG (bit 35) and to allow when the EXCEP bit is set in the response packet to generate an ERR.
2	SSMI_ERR_EN	<b>Uncleared SSMI Enable.</b> Write 1 to enable SSMI_ERR_FLAG (bit 34) and to the allow an uncleared SSMI to generate an ERR.
1	BLOCKIO_SSMI_EN	<b>SSMI on I/O write Enable.</b> Write 1 to enable BLOCKIO_SSMI_FLAG (bit 33) when I/O writes during DMA with BLOCKIO (bit 8) set occur.
0	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Enable.</b> Write 1 to enable UNEXP_TYPE_ERR_FLAG (bit 32) and to the allow an unexpected type occurring to generate an ERR.

## 6.4.1.5 GLD Power Management MSR (ATAC\_GLD\_MSR\_PM)

MSR Address 51300004h  
 Type R/W  
 Reset Value 00000000\_00000000h

## ATAC\_GLD\_MSR\_PM Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD														IOMODEA	RSVD																
31	30	29	28	27	26	25	24	23	22	21	20	19	18		17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RSVD																												PMODE1	PMODE0		

### ATAC\_GLD\_MSR\_PM Bit Descriptions

Bit	Name	Description
63:50	RSVD	<b>Reserved.</b> Returns 0 on read.
49:48	IOMODEA	<p><b>I/O Mode A Control.</b> These bits determine how the associated IDE inputs and outputs behave when the PMC asserts two internal signals that are controlled by PMS I/O Offset 20h and 0Ch. The list of affected signals is in Table 4-12 "Sleep Driven IDE Signals" on page 79.</p> <p>00: No gating of I/O cells during a Sleep sequence (Default).</p> <p>01: During a power management Sleep sequence, force inputs to their non-asserted state when PM_IN_SLPCTL is enabled.</p> <p>10: During a power management Sleep sequence, force inputs to their non-asserted state when PM_IN_SLPCTL is enabled, and park (force) outputs low when PM_OUT_SLPCTL is enabled.</p> <p>11: Immediately and unconditionally, force inputs to their not asserted state, and park (force) outputs low.</p>
47:4	RSVD	<b>Reserved.</b> Returns 0 on read.
3:2	PMODE1	<p><b>Power Mode 1.</b> Power mode for Channel 1 clock domain.</p> <p>00: Disable clock gating. Clocks are always on.</p> <p>01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy.</p> <p>10: Reserved.</p> <p>11: Reserved.</p>
1:0	PMODE0	<p><b>Power Mode 0.</b> Power mode for Channel 0 clock domain.</p> <p>00: Disable clock gating. Clocks are always on.</p> <p>01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy.</p> <p>10: Reserved.</p> <p>11: Reserved.</p>

#### 6.4.1.6 GLD Diagnostic MSR (ATAC\_GLD\_MSR\_DIAG)

MSR Address     51300005h  
 Type            R/W  
 Reset Value     00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.4.2 ATAC Specific MSRs

### 6.4.2.1 I/O Base Address (ATAC\_IO\_BAR)

MSR Address 51300008h  
 Type R/W  
 Reset Value 00000000\_00000001h

This register sets the base address of the I/O mapped bus mastering IDE and controller registers. Bits [2:0] are read only (001), indicating an 8-byte I/O address range.

**ATAC\_IO\_BAR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															BM_IDE_BAR_EN
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
RSVD																BM_IDE_BAR											ADD_RNG				

**ATAC\_IO\_BAR Bit Descriptions**

Bit	Name	Description
63:33	RSVD	<b>Reserved.</b> Write as 0. Return 0 on read.
32	BM_IDE_BAR_EN	<b>Bus Mastering IDE Base Address Enable.</b> This bit should be set to enable access to the native register set after the base address has been set. 0: Disable (i.e., cannot access native registers in ATAC). 1: Enable (normal operation).
31:17	RSVD	<b>Reserved.</b> Write to 0. Return 0 on read.
16:3	BM_IDE_BAR	<b>Bus Mastering IDE Base Address.</b> These bits form the base address of the ATAC native register set. Users may write the full address, including bits [2:0], knowing that bits [2:0] will drop off and be assumed as 000.
2:0	ADD_RNG (RO)	<b>Address Range (Read Only).</b> Hard wired to 001. This indicates that the I/O base address is in units of bytes.

### 6.4.2.2 Reset Decode (ATAC\_RESET)

MSR Address 51300010h  
 Type R/W  
 Reset Value 00000000\_00000000h

**ATAC\_RESET Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														ATAC_RESET	IDE_RESET

### ATAC\_RESET Bit Descriptions

Bit	Name	Description
63:2	RSVD	<b>Reserved.</b> Set to 0. Return 0 on read.
1	ATAC_RESET	<b>IDE Controller Reset.</b> Reset the IDE Controller. 0: Normal state (operate). 1: Reset. Write 0 to clear. This bit is level-sensitive and must be cleared after reset is performed.
0	IDE_RESET	<b>IDE Reset.</b> Reset the IDE bus. 0: Normal state (operate). 1: Reset. Write 0 to clear. This bit is level-sensitive and must be cleared after reset is performed.

#### 6.4.2.3 Channel 0 Drive 0 PIO (ATAC\_CH0D0\_PIO)

MSR Address 51300020h  
Type R/W  
Reset Value 00000000\_00009172h

**Note:** The reset value of this register is not a valid PIO mode.

### ATAC\_CH0D0\_PIO Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												PIOMODE				t2I		t3		t2W			t1								
t2IC				t3C				t2WC				t1C				t2ID		t3D		t2WD			t1D								

### ATAC\_CH0D0\_PIO Bit Descriptions

Bit	Name	Description
63:32	RSVD	<b>Reserved.</b> Set to 0.
<p><b>If MSR 51300021h[31] = 0, Format 0.</b> Selects slowest PIOMODE per channel (by comparing bit [19:16] of MSR 50302020h and MSR 50302022h) for commands.</p> <p>Format 0 settings for:</p> <ul style="list-style-type: none"> <li>PIO Mode 0 = 0000F7F4h</li> <li>PIO Mode 1 = 000153F3h</li> <li>PIO Mode 2 = 000213F1h</li> <li>PIO Mode 3 = 00035131h</li> <li>PIO Mode 4 = 00041131h</li> </ul>		
31:20	RSVD	<b>Reserved.</b> Set to 0.
19:16	PIOMODE	<b>PIO Mode.</b> 0000: PIO Mode 0                      0011: PIO Mode 3 0001: PIO Mode 1                      0100: PIO Mode 4 0010: PIO Mode 2                      All Other Values are Reserved
15:12	t2I	<b>Recovery Time.</b> Value + 1 cycle.
11:8	t3	<b>IDE_IOW# Data Setup Time.</b> Value + 1 cycle.
7:4	t2W	<b>IDE_IOW# Width Minus t3.</b> Value + 1 cycle.
3:0	t1	<b>Address Setup Time.</b> Value + 1 cycle.

## ATAC\_CH0D0\_PIO Bit Descriptions (Continued)

Bit	Name	Description
<p><b>If MSR 50302021h[31] = 1, Format 1.</b> Allows independent control of command and data.</p> <p>Format 1 settings for:</p> <ul style="list-style-type: none"> <li>PIO Mode 0 = F7F4F7F4h</li> <li>PIO Mode 1 = 53F3F173h</li> <li>PIO Mode 2 = 13F18141h</li> <li>PIO Mode 3 = 51315131h</li> <li>PIO Mode 4 = 11311131h</li> </ul>		
31:28	t2IC	<b>Command Cycle Recovery Time.</b> Value + 1 cycle.
27:24	t3C	<b>Command Cycle IDE_IOW# Data Setup.</b> Value + 1 cycle.
23:20	t2WC	<b>Command Cycle IDE_IOW# Pulse Width Minus t3.</b> Value + 1 cycle.
19:16	t1C	<b>Command Cycle Address Setup Time.</b> Value + 1 cycle.
15:12	t2ID	<b>Data Cycle Recovery Time.</b> Value + 1 cycle.
11:8	t3D	<b>Data Cycle IDE_IOW# Data Setup.</b> Value + 1 cycle.
7:4	t2WD	<b>Data Cycle IDE_IOW# Pulse Width Minus t3.</b> Value + 1 cycle.
3:0	t1D	<b>Data Cycle Address Setup Time.</b> Value + 1 cycle.
<p><b>Note:</b> Register settings described as "Value + n cycle(s)" will produce timings for the indicated parameter as measured in 66 MHz clock cycles. The 'value' that is entered is the desired number of 66 MHz clock cycles in hexadecimal; the actual parameter timing generated by that entry is the entered 'value' plus the indicated number of 'cycles' ('n') as listed in the description of that parameter.</p>		

## 6.4.2.4 Channel 0 Drive 0 DMA (ATAC\_CH0D0\_DMA)

MSR Address 51300021h  
 Type R/W  
 Reset Value 00000000\_00077771h

## ATAC\_CH0D0\_DMA Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
RSVD																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PIO_FORM	MODE66_SEL							RSVD				DMA_SEL	tKR				tDR				tKW				tDW				tM			
	MODE66_SEL							RSVD					DMA_SEL	tCRC				tSS				tCYC				tRP				tACK		

### ATAC\_CH0D0\_DMA Bit Descriptions

Bit	Name	Description
63:32	RSVD	<b>Reserved.</b> Set to 0.
<b>If bit 20 = 0, Multi-word DMA.</b> Settings for: Multi-word DMA Mode 0 = 7F0FFFF3h Multi-word DMA Mode 1 = 7F035352h Multi-word DMA Mode 2 = 7F024241h		
31	PIO_FORM	<b>PIO Mode Format.</b> 0: Format 0. 1: Format 1.
30:24	MODE66_SEL	<b>Mode 66 Select:</b> Set to 7Fh for 66 MHz system clock.
23:21	RSVD	<b>Reserved.</b> Set to 0.
20	DMA_SEL	<b>DMA Select.</b> DMA operation. 0: Multi-word DMA. 1: Ultra DMA.
19:16	tKR	<b>IDE_IOR# Recovery Time (4-bit).</b> Value + 1 cycle.
15:12	tDR	<b>IDE_IOR# Pulse Width.</b> Value + 1 cycle.
11:8	tKW	<b>IDE_IOW# Recovery Time (4-bit).</b> Value + 1 cycle.
7:4	tDW	<b>IDE_IOW# Pulse Width.</b> Value + 1 cycle.
3:0	tM	<b>IDE_CS0#/CS1# to IDE_IOR#/IOW# Setup; IDE_CS0#/CS1# Setup to IDE_DACK0#/DACK1#.</b>
<b>If bit 20 = 1, Ultra DMA.</b> Settings for: Ultra DMA Mode 0 = 7F7436A1h Ultra DMA Mode 1 = 7F733481h Ultra DMA Mode 2 = 7F723261h Ultra DMA Mode 3 = 7F713161h Ultra DMA Mode 4 = 7F703061h		
31	PIO_FORM	<b>PIO Mode Format.</b> 0: Format 0. 1: Format 1.
30:24	MODE66_SEL	<b>Mode 66 Select:</b> Set to 7Fh for 66 MHz system clock.
23:21	RSVD	<b>Reserved.</b> Set to 011. Will read back as 011.
20	DMA_SEL	<b>DMA Select.</b> DMA operation. 0: Multi-word DMA. 1: Ultra DMA.
19:16	tCRC	<b>CRC Setup UDMA in IDE_DACK#.</b> Value + 1 cycle (for host terminate CRC setup = tMLI + tSS).
15:12	tSS	<b>UDMA Out.</b> Value + 1 cycle.
11:8	tCYC	<b>Data Setup and Cycle Time UDMA Out.</b> Value + 2 cycles.
7:4	tRP	<b>Ready to Pause Time.</b> Value + 1 cycle. Used only for UDMA-in. <b>Note:</b> tRFS + 1 tRP on next clock.
3:0	tACK	<b>IDE_CS0#/CS1# Setup to IDE_DACK0#/DACK1#.</b> Value + 1 cycle.
<b>Note:</b> Register settings described as "Value + n cycle(s)" produce timings for the indicated parameter as measured in 66 MHz clock cycles. The 'value' that is entered is the desired number of 66 MHz clock cycles in hexadecimal; the actual parameter timing generated by that entry is the entered 'value' plus the indicated number of 'cycles' ('n') as listed in the description of that parameter.		



**6.4.2.5 Channel 0 Drive 1 PIO (ATAC\_CH0D1\_PIO)**

MSR Address 51300022h  
 Type R/W  
 Reset Value 00000000\_00009172h

Refer to Section 6.4.2.3 "Channel 0 Drive 0 PIO (ATAC\_CH0D0\_PIO)" on page 270 for bit descriptions.

**6.4.2.6 Channel 0 Drive 1 DMA (ATAC\_CH0D1\_DMA)**

MSR Address 51300023h  
 Type R/W  
 Reset Value 00000000\_00077771h

Refer to Section 6.4.2.4 "Channel 0 Drive 0 DMA (ATAC\_CH0D0\_DMA)" on page 271 for bit descriptions.

**6.4.2.7 PCI Abort Error (ATAC\_PCI\_ABRERR)**

MSR Address 51300024h  
 Type R/W  
 Reset Value 00000000\_00000000h

**ATAC\_PCI\_ABRERR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Read: RSVD; Write: PCI_ABORT																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read: RSVD; Write: PCI_ABORT																															

**ATAC\_PCI\_ABRERR Bit Descriptions**

Bit	Name	Description
63:0	RSVD/PCI_ABORT	<b>Reserved.</b> Return 0 on read. <b>PCI Abort.</b> Write (of any value) to this register will set bit 1 of Bus Master 0 Status Register.

### 6.4.3 ATAC Native Registers

#### 6.4.3.1 Bus Master 0 Command - Primary (ATAC\_BM0\_CMD\_PRIM)

ATAC I/O Address 00h

Type R/W

Reset Value 00h

**ATAC\_BM0\_CMD\_PRIM Register Map**

7	6	5	4	3	2	1	0
RSVD				RWCTL	RSVD		BMCTL

**ATAC\_BM0\_CMD\_PRIM Bit Descriptions**

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Set to 0. Return 0 on read.
3	RWCTL	<b>Read or Write Control.</b> Sets the direction of bus master transfers. 0: PCI reads performed. 1: PCI writes performed. This bit should not be changed when the bus master is active.
2:1	RSVD	<b>Reserved.</b> Set to 0. Return 0 on read.
0	BMCTL	<b>Bus Master Control.</b> Controls the state of the bus master. 0: Disable master. 1: Enable master. Bus master operations can be halted by setting bit 0 to 0. Once an operation has been halted, it can not be resumed. If bit 0 is set to 0 while a bus master operation is active, the command is aborted and the data transferred from the drive is discarded. This bit should be reset after completion of data transfer.

#### 6.4.3.2 Bus Master 0 Status - Primary (ATAC\_BM0\_STS\_PRIM)

ATAC I/O Address 02h

Type R/W

Reset Value 00h

**ATAC\_BM0\_STS\_PRIM Register Map**

7	6	5	4	3	2	1	0
MODE	D1DMA	D0DMA	RSVD		BMINT	BMERR	BMSTS

**ATAC\_BM0\_STS\_PRIM Bit Descriptions**

Bit	Name	Description
7	MODE (RO)	<b>Simplex Mode (Read Only).</b> Can both the primary and secondary channel operate independently? 0: Yes. 1: No (simplex mode).
6	D1DMA	<b>Drive 1 DMA Capable.</b> Allows Drive 1 to be capable of DMA transfers. 0: Disable. 1: Enable.

**ATAC\_BM0\_STS\_PRIM Bit Descriptions (Continued)**

Bit	Name	Description
5	D0DMA	<b>Drive 0 DMA Capable.</b> Allows Drive 0 to be capable of DMA transfers. 0: Disable. 1: Enable.
4:3	RSVD	<b>Reserved:</b> Set to 0. Must return 0 on reads.
2	BMINT	<b>Bus Master Interrupt.</b> Has the bus master detected an interrupt? 0: No. 1: Yes. Write 1 to clear.
1	BMERR	<b>Bus Master Error.</b> Has the bus master detected an error during data transfer? (This bit is set by a write access to MSR 51300024h.) 0: No. 1: Yes. Write 1 to clear.
0	BMSTS (RO)	<b>Bus Master Status (Read Only).</b> Is the bus master active? 0: No. 1: Yes.

**6.4.3.3 Bus Master 0 PRD Table Address - Primary (ATAC\_BM0\_PRD)**

ATA-5 I/O Address 04h

Type R/W

Reset Value 00000000h

**ATAC\_BM0\_PRD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRD_PNTR																RSVD															

**ATAC\_BM0\_PRD Bit Descriptions**

Bit	Name	Description
31:2	PRD_PNTR	<b>Pointer to the Physical Region Descriptor Table.</b> This register is a PRD table pointer for IDE Bus Master 0.  When written, this register points to the first entry in a PRD table. Once IDE Bus Master 0 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register to the next PRD by adding 08h.  When read, this register points to the next PRD.
1:0	RSVD	<b>Reserved.</b> Set to 0.

## 6.5 USB Controller Register Descriptions

The control registers allow software to communicate with the USB Controller. These control registers can be broadly divided into four register sets:

- Standard GeodeLink (GLD) Device MSRs
- USB Specific MSRs
- USB Embedded PCI Configuration Registers
- Host Controller Native Registers

The MSRs (both Standard and Specific) are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details. The USB Transceiver Control MSR (MSR\_USBXCVR) is dedicated to the USB transceivers and allows software to control transceiver voltage and current settings. It also provides a mechanism by which the software can turn on or turn off the transmitter side of the transceiver for power management purposes.

The embedded PCI Configuration Registers are 32-bit registers decoded from the embedded PCI address bits 7 through 2 and C/BE[3:0]#, when IDSEL is high, AD[10:8] select the appropriate function, and AD[1:0] are 00. This embedded PCI bus is accessed via special GeodeLink Adapter MSR accesses. Bytes within a 32-bit address are selected with the valid byte enables. All registers can be accessed via 8, 16, or 32-bit cycles (i.e., each byte is individually selected by the byte enables.) Registers marked as reserved, and reserved bits within a register are not implemented and should return 0s when read. Writes have no effect for reserved registers.

The Host Controller (HC) contains a set of on-chip operational registers that are mapped into a non-cacheable portion of the system addressable space. These registers are used by the Host Controller Driver (HCD). According to the function of these registers, they are divided into four partitions, specifically for Control and Status, Memory Pointer,

Frame Counter and Root Hub. All of the registers should be read and written as DWORDs. To ensure interoperability, the Host Controller driver that does not use a reserved field should not assume that the reserved field contains 0. Furthermore, the Host Controller driver should always preserve the value(s) of the reserved field. When a R/W register is modified, the Host Controller driver should first read the register, modify the bits desired, then write the register with the reserved bits still containing the read value. Alternatively, the Host Controller Driver can maintain an in-memory copy of previously written values that can be modified and then written to the Host Controller register. When a write to set/clear register is written, bits written to reserved fields should be 0. These registers can be grouped into four functional groups: Host Controller control and status registers, memory pointers, frame counter and control registers, and Root Hub status and control.

- Host Controller control and status registers define the operating mode of the host controller. They reflect current status of the host controller, provide interrupt control and status, and reflect error status conditions.
- Memory pointers provide pointers to the data structure that are required to communicate with the host controller driver and perform transactions based on the transfer descriptors that reside in memory.
- Frame counter and control provide frame timing status and control. This set of registers also govern Start Of the Frame (SOF) timing and control events that are tied to frame timing intervals.
- Root Hub status and control registers are dedicated to the root hub function. Two sets of registers are included to control the two ports.

Tables 6-13 through 6-16 are register summary tables that include reset values and page references where the register maps and bit descriptions are provided.

**Table 6-13. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
USBC1: 51600000h USBC2: 51200000h	RO	GLD Capabilities MSR (USBC_GLD_MSR_CAP)	00000000_002420xxh	Page 279
USBC1: 51600001h USBC2: 51200001h	R/W	GLD Master Configuration MSR (USBC_GLD_MSR_CONFIG)	00000000_0000F000h	Page 279
USBC1: 51600002h USBC2: 51200002h	R/W	GLD SMI MSR (USBC_GLD_MSR_SMI)	00000000_00000000h	Page 281
USBC1: 51600003h USBC2: 51200003h	R/W	GLD Error MSR (USBC_GLD_MSR_ERROR)	00000000_00000000h	Page 282
USBC1: 51600004h USBC2: 51200004h	R/W	GLD Power Management MSR (USBC_GLD_MSR_PM)	00000000_00000000h	Page 284
USBC1: 51600005h USBC2: 51200005h	R/W	GLD Diagnostic MSR (USBC_GLD_MSR_DIAG)	00000000_00000000h	Page 284

**Table 6-14. USB Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
USBC1: 51600008h USBC2: 51200008h	R/W	USB Transceiver Control (USBC_XCVR_CNTRL)	00000000_00020100h	Page 285

**Table 6-15. USB Embedded PCI Configuration Registers Summary**

Index	Type	Width (Bits)	Register Name	Reset Value	Reference
00h	RO	16	Vendor Identification (USBC_PCI_VENID)	0E11h	Page 286
02h	RO	16	Device Identification (USBC_PCI_DEVID)	A0F8h	Page 286
04h	R/W	16	Command (USBC_PCI_CMD)	0000h	Page 287
06h	R/W	16	Status (USBC_PCI_STS)	0280h	Page 288
08h	RO	8	Device Revision Identification (USBC_PCI_DEVREVID)	06h	Page 289
09h	RO	24	PCI Class Code (USBC_PCI_CLASS)	0C0310h	Page 289
0Ch	R/W	8	Cache Line Size (USBC_PCI_CACHE)	00h	Page 289
0Dh	R/W	8	Latency Timer (USBC_PCI_LTNCY_TMR)	00h	Page 290
0Eh	R/W	8	Header Type (USBC_PCI_HEADER)	00h	Page 290
0Fh	RO	8	BIST (USBC_PCI_BIST)	00h	Page 290
10h	R/W	32	Base Address Register (USBC_PCI_BAR)	00000000h	Page 291
3Ch	R/W	8	Interrupt Line (USBC_PCI_INT_LINE)	00h	Page 291
3Dh	R/W	8	Interrupt Pin (USBC_PCI_INT_PIN)	01h	Page 292
3Eh	R/W	8	Minimum Grant (USBC_PCI_MIN_GNT)	00h	Page 292
3Fh	R/W	8	Maximum Latency (USBC_PCI_MAX_LTNCY)	50h	Page 292
40h	R/W	32	ASIC Test Mode Enable (USBC_PCI_ASIC_TEST)	000F0000h	Page 293
44h	R/W	16	ASIC Operational Mode Enable (USBC_PCI_ASIC_MODE)	0000h	Page 293

**Table 6-16. USB Host Controller Native Registers Summary**

USB Memory Offset	Width (Bits)	Type		Register Name	Reset Value	Reference
		HCD	HC			
00h	32	RO	RO	Host Controller Revision (USBC_HcRevision)	00000110h	Page 294
04h	32	R/W	R/W	Host Controller Control (USBC_HcControl)	00000000h	Page 294
08h	32	R/W	R/W	Host Controller Command Status (USBC_HcCommandStatus)	00000000h	Page 296
0Ch	32	R/W	R/W	Host Controller Interrupt Status (USBC_HcInterruptStatus)	00000000h	Page 297
10h	32	R/W	RO	Host Controller Interrupt Enable (USBC_HcInterruptEnable)	00000000h	Page 298

Table 6-16. USB Host Controller Native Registers Summary (Continued)

USB Memory Offset	Width (Bits)	Type		Register Name	Reset Value	Reference
		HCD	HC			
14h	32	R/W	RO	Host Controller Interrupt Disable (USBC_HcInterruptDisable)	00000000h	Page 299
18h	32	R/W	RO	Host Controller HCCA (USBC_HcHCCA)	00000000h	Page 300
1Ch	32	RO	R/W	Host Controller Current Period List ED (USBC_HcPeriodCurrentED)	00000000h	Page 300
20h	32	R/W	RO	Host Controller Control List Head ED (USBC_HcControlHeadED)	00000000h	Page 300
24h	32	R/W	R/W	Host Controller Current Control List ED (USBC_HcControlCurrentED)	00000000h	Page 301
28h	32	R/W	RO	Host Controller Bulk List Head ED (USBC_HcBulkHeadED)	00000000h	Page 301
2Ch	32	R/W	R/W	Host Controller Current Bulk List ED (USBC_HcBulkCurrentED)	00000000h	Page 301
30h	32	RO	R/W	Host Controller Current Done List Head ED (USBC_HcDoneHead)	00000000h	Page 302
34h	32	R/W	RO	Host Controller Frame Interval (USBC_HcFmInterval)	00002EDFh	Page 302
38h	32	RO	R/W	Host Controller Frame Remaining (USBC_HcFrameRemaining)	00000000h	Page 303
3Ch	32	RO	R/W	Host Controller Frame Number (USBC_HcFmNumber)	00000000h	Page 303
40h	32	R/W	RO	Host Controller Periodic Start (USBC_HcPeriodicStart)	00000000h	Page 304
44h	32	R/W	RO	Host Controller Low Speed Threshold (USBC_HcLSThreshold)	00000628h	Page 304
48h	32	R/W	RO	Host Controller Root Hub Descriptor A (USBC_HcRhDescriptorA)	01000002h	Page 305
4Ch	32	R/W	RO	Host Controller Root Hub Descriptor B (USBC_HcRhDescriptorB)	00000000h	Page 306
50h	32	R/W	R/W	Host Controller Root Hub Status (USBC_HcRhStatus)	00000000h	Page 307
54h	32	R/W	R/W	Host Controller Root Hub Port Status 1 (USBC_HcRhPortStatus[1])	00000000h	Page 308
58h	32	R/W	R/W	Host Controller Root Hub Port Status 2 (USBC_HcRhPortStatus[2])	00000000h	Page 310

### 6.5.1 Standard GeodeLink Device (GLD) MSRs

#### 6.5.1.1 GLD Capabilities MSR (USBC\_GLD\_MSR\_CAP)

MSR Address USBC1: 51600000h  
 USBC2: 51200000h  
 Type RO  
 Reset Value 00000000\_002420xxh

**USBC\_GLD\_MSR\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID																REV_ID							

**USBC\_GLD\_MSR\_CAP Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2420h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

#### 6.5.1.2 GLD Master Configuration MSR (USBC\_GLD\_MSR\_CONFIG)

MSR Address USBC1: 51600001h  
 USBC2: 51200001h  
 Type R/W  
 Reset Value 00000000\_0000F000h

**USBC\_GLD\_MSR\_CONFIG Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32		
RSVD																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD									PCI_MAST_STOP_EN		PDP		SEL		FIX_PREFETCH			RSVD		USB_LOPWR_EN		USB_GLITCHFIX_EN		RSVD									

## USBC\_GLD\_MSR\_CONFIG Bit Descriptions

Bit	Name	Description
63:23	RSVD	<b>Reserved.</b> Writes “don’t care” and reads return 0.
22	PCI_MAST_STOP_EN	<b>PCI Master Stop Enable.</b>
21:20	PDP	<b>PCI Master Prefetch Decode Policy.</b>
19	SEL	<b>Select.</b> 0: Use field [18:16]. 1: Use field [22:20].
18:16	FIX_PREFETCH	<b>Fixed Read Prefetch Policy.</b> 000: None. Each read takes a complete trip to memory. 001: Initial read 08 bytes. Read next 8 only when requested. 010: Initial read 16 bytes. Read next 16 only when requested. 011: Initial read 32 bytes. Read next 32 only when requested. 100: Initial read 32 bytes. Read next 32 when 16 bytes left. 101, 110, and 111: Reserved.
15:14	RSVD	<b>Reserved.</b> Read as written.
13	USB_LOPWR_EN	<b>USB Low Power Enable.</b> When 0, the transceiver is prevented from entering the low power Suspend state. This bit has no functional impact on the transceiver or USB Controller. Defaults to 1 (i.e., Suspend allowed).
12	USB_GLITCHFIX_EN	<b>USB Glitch Fix Enable:</b> When 0, enable transceiver de-glitch logic. This logic suppresses a glitch that can occur when: 1) The transceiver moves from the suspend state to any other state. and 2) A low speed device is connected. The state is defined by HcControl[7:6]. Defaults to 1 (i.e., no glitch suppression). Note that this bit represents a deviation from the standard MSR_CONFIG register in Section 4.8.2 "MSR Address 1: Master Configuration" on page 74.
11:0	RSVD	<b>Reserved.</b> Read as written.



**6.5.1.3 GLD SMI MSR (USBC\_GLD\_MSR\_SMI)**

MSR Address USBC1: 51600002h  
 USBC2: 51200002h  
 Type R/W  
 Reset Value 00000000\_00000000h

**USBC\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32			
RSVD																																	USB_INT_ASMI_FLAG	USB_ASMI_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RSVD																																	USB_INT_ASMI_EN	USB_ASMI_EN

**GLD\_MSR\_SMI Bit Descriptions**

Bit	Name	Description
63:34	RSVD	<b>Reserved.</b> Returns 0 on read.
33	USB_INT_ASMI_FLAG	<b>USB Interrupt Flag.</b> If high, records that an ASMI was generated due to an INT being generated by the USB module. The INT condition is cleared inside the USB module core. Write 1 to clear the flag; writing 0 has no effect. (Also see bit 1 description.)
32	USB_ASMI_FLAG	<b>USB ASMI Flag.</b> If high, records that an ASMI was generated by the USB module. The ASMI condition is cleared inside the USB module. Write 1 to clear the flag; writing 0 has no effect. (Also see bit 0 description.)
31:2	RSVD	<b>Reserved.</b> Returns 0 on read.
1	USB_INT_ASMI_EN	<b>USB Interrupt Enable.</b> If this bit is high, the INT generated by the USB module is used to generate a GeodeLink ASMI.
0	USB_ASMI_EN	<b>USB ASMI Enable.</b> If this bit is high, the ASMI generated by the USB module is used to generate a GeodeLink ASMI.

#### 6.5.1.4 GLD Error MSR (USBC\_GLD\_MSR\_ERROR)

MSR Address USBC1: 51600003h  
 USBC2: 51200003h  
 Type R/W  
 Reset Value 00000000\_00000000h

#### USBC\_GLD\_MSR\_ERROR Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																					CMD_ERR_FLAG	SYS_ERR_FLAG	PARIN_ERR_FLAG	PAROUT_ERR_FLAG	TIMEOUT_ERR_FLAG	PA_SVADRS_ERR_FLAG	PA_TABORT_ERR_FLAG	PA_MABORT_ERR_FLAG	EXCEP_ERR_FLAG	RSVD	UNEXP_TYPE_ERR_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																					CMD_ERR_EN	SYS_ERR_EN	PARIN_ERR_EN	PAROUT_ERR_EN	TIMEOUT_ERR_EN	PA_SVADRS_ERR_EN	PA_TABORT_ERR_EN	PA_MABORT_ERR_EN	EXCEP_ERR_EN	RSVD	UNEXP_TYPE_ERR_EN

#### USBC\_GLD\_MSR\_ERROR Bit Descriptions

Bit	Name	Description
63:44	RSVD	<b>Reserved.</b> Returns 0 on read.
43	CMD_ERR_FLAG	<b>Command Error Flag.</b> If high, records that an ERR was generated because a PCI device asserted the embedded PCI bus Parity Error signal. (The PCI Adapter slave interface generated a command error on a command received from the PCI device, or when the 2 LSBs of the address received from the PCI device in the address phase are non-zero.) Write 1 to clear the flag; writing 0 has no effect. CMD_ERR_EN (bit 11) must be high to generate ERR and set flag.
42	SYS_ERR_FLAG	<b>System Error Flag.</b> If high, records that an ERR was generated because a PCI device asserted the embedded PCI bus System Error signal. Write 1 to clear the flag; writing 0 has no effect. SYS_ERR_EN (bit 10) must be high to generate ERR and set flag.
41	PARIN_ERR_FLAG	<b>Parity Error In Error Flag.</b> If high, records that an ERR was generated because a Parity Error on a PCI device master to PCI Adapter target transaction occurred. Write 1 to clear the flag; writing 0 has no effect. PARIN_ERR_EN (bit 9) must be high to generate ERR and set flag.
40	PAROUT_ERR_FLAG	<b>Parity Error Out Error Flag.</b> If high, records that an ERR was generated because a Parity error on PCI Adapter master to PCI device target transaction occurred. Write 1 to clear the flag; writing 0 has no effect. PAROUT_ERR_EN (bit 8) must be high to generate ERR and set flag.
39	TIMEOUT_ERR_FLAG	<b>Master Timeout Error Flag.</b> If high, records that an ERR was generated because the PCI Adapter completed a master bus cycle to the GeodeLink Adapter when the PCI device failed to perform an expected retry after a disconnect. Write 1 to clear the flag; writing 0 has no effect. TIMEOUT_ERR_EN (bit 7) must be high to generate ERR and set flag.

**USBC\_GLD\_MSR\_ERROR Bit Descriptions (Continued)**

Bit	Name	Description
38	PA_SVADRS_ERR_FLAG	<b>PCI Adapter Slave Interface Address Parity Error Flag.</b> If high, records that an ERR was generated because the PCI Adapter slave interface generated a parity error on the address received from the PCI device in the address phase. Write 1 to clear the flag; writing 0 has no effect. PA_SVADRS_ERR_EN (bit 6) must be high to generate ERR and set flag.
37	PA_TABORT_ERR_FLAG	<b>PCI Adapter Target Abort Error Flag.</b> If high, records that an ERR was generated because the PCI Adapter, as a target, aborted the PCI bus cycle. The GeodeLink Adapter and the PCI Adapter complete the GeodeLink Transaction. Write 1 to clear the flag; writing 0 has no effect. PA_TABORT_ERR_EN (bit 5) must be high to generate ERR and set flag.
36	PA_MABORT_ERR_FLAG	<b>PCI Adapter Master Abort Error.</b> If high, records that an ERR was generated because the PCI device did not respond as a target. The PCI Adapter aborted the PCI bus cycle and indicated to the GeodeLink Adapter to abort the GeodeLink Adapter transaction. The GeodeLink Adapter completes the GeodeLink transaction and sets the EXCP bit in any response packet. Write 1 to clear the flag; writing 0 has no effect. PA_MABORT_ERR_EN (bit 4) must be high to generate ERR and set flag.
35	EXCEP_ERR_FLAG	<b>Exception Bit Error Flag.</b> If high, records that an ERR was generated because the EXCEP bit was set in THE response packet associated with a device master request. Write 1 to clear the flag; writing 0 has no effect. EXCEP_ERR_EN (bit 3) must be high to generate ERR and set flag.
34:33	RSVD	<b>Reserved.</b> Writes don't care. Reads return 0.
32	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Error Flag.</b> If high, records that an ERR was generated because an unexpected type or other "bad" GeodeLink transaction. This error is fatal and there is no system recovery. The device (GeodeLink Adapter) will hang. Write 1 to clear the flag; writing 0 has no effect. UNEXP_TYPE_ERR_EN (bit 0) must be high to generate ERR and set flag.
31:12	RSVD	<b>Reserved.</b> Writes don't care. Reads return 0.
11	CMD_ERR_EN	<b>Command Error Enable.</b> Write 1 to enable CMD_ERR_FLAG (bit 43) and to allow the event to generate an ERR.
10	SYS_ERR_EN	<b>System Error Enable.</b> Write 1 to enable SYS_ERR_FLAG (bit 42) and to allow the event to generate an ERR.
9	PARIN_ERR_EN	<b>Parity Error In Enable.</b> Write 1 to enable PARIN_ERR_FLAG (bit 41) and to allow the event to generate an ERR.
8	PAROUT_ERR_EN	<b>Parity Error Out Enable.</b> Write 1 to enable PAROUT_ERR_FLAG (bit 40) and to allow the event to generate an ERR.
7	TIMEOUT_ERR_EN	<b>Timeout Error Enable.</b> Write 1 to enable TIMEOUT_ERR_FLAG (bit 39) and to allow the event to generate an ERR.
6	PA_SVADRS_ERR_EN	<b>PCI Adapter Slave Interface Address Parity Error Enable.</b> Write 1 to enable PA_SVADRS_ERR_FLAG (bit 38) and to allow the event to generate an ERR.
5	PA_TABORT_ERR_EN	<b>PCI Adapter Target Abort Error Enable.</b> Write 1 to enable PA_TABORT_ERR_FLAG (bit 37) and to allow the event to generate an ERR.
4	PA_MABORT_ERR_EN	<b>PCI Adapter Master Abort Error Enable.</b> Write 1 to enable PA_MABORT_ERR_FLAG (bit 36) and to allow the event to generate an ERR.
3	EXCEP_ERR_EN	<b>Exception Bit Error Enable.</b> Write 1 to enable EXCEP_ERR_FLAG (bit 35) and to allow the event to generate an ERR.
2:1	RSVD	<b>Reserved.</b> Writes don't care. Reads return zero.
0	UNEXP_TYPE_ERR_EN	<b>Unexpected Type Error Enable.</b> Write 1 to enable UNEXP_TYPE_ERR_FLAG (bit 32) and to allow the event to generate an ERR.

**6.5.1.5 GLD Power Management MSR (USBC\_GLD\_MSR\_PM)**

MSR Address USBC1: 51600004h  
 USBC2: 51200004h  
 Type R/W  
 Reset Value 00000000\_00000000h

**USBC\_GLD\_MSR\_PM Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											PMODE2	PMODE1	PMODE0		

**USBC\_GLD\_MSR\_PM Bit Descriptions**

Bit	Name	Description
63:6	RSVD	<b>Reserved.</b> Returns 0 on read.
5:4	PMODE2	<b>Power Mode 2.</b> Controls clock behavior for the USB Core. 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
3:2	PMODE1	<b>Power Mode 1.</b> Controls clock behavior for the USB PCI Adapter 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
1:0	PMODE0	<b>Power Mode 0.</b> Controls clock behavior for the USB GeodeLink Adapter 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.

**6.5.1.6 GLD Diagnostic MSR (USBC\_GLD\_MSR\_DIAG)**

MSR Address USBC1: 51600005h  
 USBC2: 51200005h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.5.2 USB Specific MSRs

### 6.5.2.1 USB Transceiver Control (USBC\_XCVR\_CNTRL)

MSR Address USBC1: 51600008h  
 USBC2: 51200008h  
 Type R/W  
 Reset Value 00000000\_00020100h

**USBC\_XCVR\_CNTRL Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														XCEIV2_PM	RCVR2_VADJ		XMIT2_CADJ						XCEIV1_PM	RCVR1_VADJ		XMIT1_CADJ					

**USBC\_XCVR Bit Description**

Bit	Name	Description
63:18	RSVD	<b>Reserved.</b> Read as 0.
17	XCEIV2_PM	<b>Transceiver 2 Power Management Bit.</b> Disables transceiver #2 and puts the transceiver in a low power state. 0: Enable transceiver. 1: Disable transceiver.
16:14	RCVR2_VADJ	<b>Single Ended Receiver #2 Threshold Voltage Adjustment.</b> For proper operation this must be set to 100 (4h). See electrical specifications Section 7.3.6 "USB Signals" on page 560.
13:9	XMIT2_CADJ	<b>Differential Transmitter #2 Slew Rate Current Adjustment.</b> For proper operation this must be set to 010 (2h). See electrical specifications Section 7.3.6 "USB Signals" on page 560.
8	XCEIV1_PM	<b>Transceiver 1 Power Management Bit.</b> Disables transceiver #1 and puts the transceiver in a low power state. 0: Enable transceiver. 1: Disable transceiver.
7:5	RCVR1_VADJ	<b>Single Ended Receiver #1 Threshold Voltage Adjustment.</b> For proper operation this must be set to 100 (4h). See electrical specifications Section 7.3.6 "USB Signals" on page 560.
4:0	XMIT1_CADJ	<b>Differential Transmitter #1 Slew Rate Current Adjustment.</b> For proper operation this must be set to 010 (2h). See electrical specifications Section 7.3.6 "USB Signals" on page 560.

### 6.5.3 USB Embedded PCI Configuration Registers

#### 6.5.3.1 Vendor Identification (USBC\_PCI\_VENID)

Index            00h  
 Type            RO  
 Reset Value    0E11h

**USBC\_PCI\_VENID Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDOR_ID															

**USBC\_PCI\_VENID Bit Descriptions**

Bit	Name	Description
15:0	VENDOR_ID	Vendor ID.

#### 6.5.3.2 Device Identification (USBC\_PCI\_DEVID)

Index            02h  
 Type            RO  
 Reset Value    A0F8h

**USBC\_DEVID Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICE_ID															

**USBC\_DEVID Bit Descriptions**

Bit	Name	Description
15:0	DEVICE_ID	Device ID.

**6.5.3.3 Command (USBC\_PCI\_CMD)**

Index 04h  
 Type R/W  
 Reset Value 0000h

**USBC\_PCI\_CMD Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						FB2B_EN	SERR	WS_CTL	PAR	VGA_SNP_EN	MEMW_INV	SPC_CYC	MSTR_EN	MEM	IO

**USBC\_PCI\_CMD Bit Descriptions**

Bit	Name	Description
15:10	RSVD	<b>Reserved.</b> Set to 0.
9	FB2B_EN (RO)	<b>Fast Back-to-Back Enable (Read Only).</b> USB only acts as a master to a single device, so this functionality is not needed. It is always disabled (must always be set to 0).
8	SERR	<b>System Error.</b> USB asserts SERR# when it detects an address parity error. 0: Disable; 1: Enable.
7	WS_CTL	<b>Wait Cycle Control.</b> USB does not need to insert a wait state between the address and data on the AD lines. It is always disabled (bit is set to 0).
6	PAR	<b>Parity Error.</b> USB asserts PERR# when it is the agent receiving data and it detects a data parity error. 0: Disable; 1: Enable.
5	VGA_SNP_EN (RO)	<b>VGA Palette Snoop Enable (Read Only).</b> USB does not support this function. It is always disabled (bit is set to 0).
4	MEMW_INV	<b>Memory Write and Invalidate:</b> Allow USB to run Memory Write and Invalidate commands. 0: Disable; 1: Enable.  The Memory Write and Invalidate commands only occur if the cache line size is set to 32 bytes and the memory write is exactly one cache line.  This bit is ignored by the GeodeLink Adapter and hence the setting is a 'don't care'.
3	SPC_CYC	<b>Special Cycles.</b> USB does not run special cycles on PCI. It is always disabled (bit is set to 0).
2	MSTR_EN	<b>PCI Master Enable.</b> Allow USB to run PCI master cycles. 0: Disable; 1: Enable.
1	MEM	<b>Memory Space.</b> Allow USB to respond as a target to memory cycles. 0: Disable; 1: Enable.
0	IO	<b>I/O Space.</b> Allow USB to respond as a target to I/O cycles. 0: Disable; 1: Enable.

### 6.5.3.4 Status (USBC\_PCI\_STS)

Index 06h  
 Type R/W  
 Reset Value 0280h

#### USBC\_PCI\_STS Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR_ERR	SERR_STS	RCVD_MSTR_ABT_STS	RCVD_TRGT_ABT_STS	SIG_TRGT_ABT_STS	DEVSEL_TIME		DATA_PAR	FB2B	RSVD						

#### USBC\_PCI\_STS Bit Descriptions

Bit	Name	Description
15	PAR_ERR	<b>Detected Parity Error.</b> This bit is set whenever the USB detects a parity error, even if the Parity Error (response) detection enable bit (Index 04h[6]) is disabled. Write 1 to clear.
14	SERR_ERR	<b>SERR# Status.</b> This bit is set whenever the USB detects a PCI address error. Write 1 to clear.
13	RCVD_MSTR_ABT_STS	<b>Received Master Abort Status.</b> This bit is set when the USB, acting as a PCI master, aborts a PCI bus memory cycle. Write 1 to clear.
12	RCVD_TRGT_ABT_STS	<b>Received Target Abort Status.</b> This bit is set when a USB generated PCI cycle (USB is the PCI master) is aborted by a PCI target. Write 1 to clear.
11	SIG_TRGT_ABT_STS	<b>Signaled Target Abort Status.</b> This bit is set whenever the USB signals a target abort. Write 1 to clear.
10:9	DEVSEL_TIME	<b>DEVSEL# Timing (Read Only).</b> These bits indicate the DEVSEL# timing when performing a positive decode. Since DEVSEL# is asserted to meet the medium timing, these bits are encoded as 01.
8	DATA_PAR	<b>Data Parity Reported.</b> Set to 1 if the Parity Error bit (Command Register bit 6) is set, and USB detects PERR# asserted while acting as PCI master (whether PERR# was driven by USB or not).
7	FB2B	<b>Fast Back-to-Back Capable (Read Only).</b> USB does support fast back-to-back transactions when the transactions are not to the same agent. This bit is always 1.
6:0	RSVD	<b>Reserved.</b> Set to 0.



**6.5.3.5 Device Revision Identification (USBC\_PCI\_DEVREVID)**

Index 08h  
 Type RO  
 Reset Value 06h

**USBC\_PCI\_DEVREVID Register Map**

7	6	5	4	3	2	1	0
DEV_REV_ID							

**USBC\_PCI\_DEVREVID Bit Descriptions**

Bit	Name	Description
7:0	DEV_REV_ID	Device Revision ID.

**6.5.3.6 PCI Class Code (USBC\_PCI\_CLASS)**

Index 09h  
 Type RO  
 Reset Value 0C0310h

**USBC\_PCI\_CLASS Register Map**

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCI_CLASS																							

**USBC\_PCI\_CLASS Bit Descriptions**

Bit	Name	Description
23:0	PCI_CLASS	<b>PCI Class.</b> This register identifies this function as an OpenHCI device. The base class is 0Ch (serial bus controller). The sub class is 03h (universal serial bus). The programming interface is 10h (OpenHCI).

**6.5.3.7 Cache Line Size (USBC\_PCI\_CACHE)**

Index 0Ch  
 Type R/W  
 Reset Value 00h

**USBC\_PCI\_CACHE Register Map**

7	6	5	4	3	2	1	0
PCI_CACHE							

**USBC\_PCI\_CACHE Bit Descriptions**

Bit	Name	Description
7:0	PCI_CACHE	<b>PCI Cache.</b> The USB controller stores the system cache line size, in units of 32-bit WORDs in bit 3 of this register. A cache line size of 32 bytes is the only one applicable to this design. Users must set this register to 08h for proper operation. Any value other than 08h written to this register is read back as 00h.

**6.5.3.8 Latency Timer (USBC\_PCI\_LTNCY\_TMR)**

Index 0Dh  
 Type R/W  
 Reset Value 00h

**USBC\_PCI\_LTNCY\_TMR Register Map**

7	6	5	4	3	2	1	0
LTNCY_TMR							

**USBC\_PCI\_LTNCY\_TMR Bit Descriptions**

Bit	Name	Description
7:0	LTNCY_TMR	<b>Latency Timer.</b> This register identifies the value of the latency timer in PCI clocks for PCI bus master cycles.

**6.5.3.9 Header Type (USBC\_PCI\_HEADER)**

Index 0Eh  
 Type RO  
 Reset Value 00h

**USBC\_PCI\_HEADER Register Map**

7	6	5	4	3	2	1	0
PCI_HEADER							

**USBC\_PCI\_HEADER Bit Descriptions**

Bit	Name	Description
7:0	PCI_HEADER	<b>PCI Header Type.</b> This register identifies the type of the predefined header in the configuration space. Since the USB is a single function device and not a PCI-to-PCI bridge, this byte should be read as 00h.

**6.5.3.10 BIST (USBC\_PCI\_BIST)**

Index 0Fh  
 Type RO  
 Reset Value 00h

**USBC\_PCI\_BIST Register Map**

7	6	5	4	3	2	1	0
PCI_BIST							

**USBC\_PCI\_BIST Bit Descriptions**

Bit	Name	Description
7:0	PCI_BIST	<b>PCI BIST.</b> This register identifies the control and status of Built In Self Test. The USB does not implement BIST, so this register is read only.

**6.5.3.11 Base Address Register (USBC\_PCI\_BAR)**

Index 10h  
 Type R/W  
 Reset Value 00000000h

This BAR sets the base address of the memory mapped USB controller registers. Bits [11:0] are read only (0000 0000 0000), indicating a 4 KB memory address range. Refer to Section 6.5.4 "Host Controller Native Registers" on page 294 for the USB controller register bit formats and reset values.

**USBC\_PCI\_BAR Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE_ADDR											ADDR_RNG											PREFETCH	WIDTH	MEM							

**USBC\_PCI\_BAR Bit Descriptions**

Bit	Name	Description
31:12	BASE_ADDR	<b>Base Address.</b> POST writes the value of the memory base address to this register.
11:4	ADDR_RNG	<b>Address Range (Always 0.)</b> Indicates a 4k byte address range is requested.
3	PREFETCH	<b>Prefetch (Always 0).</b> Indicates there is no support for prefetchable memory.
2:1	WIDTH	<b>Width (Always 0).</b> Indicates that the base register is 32-bits wide and can be placed anywhere in 32-bit memory space.
0	MEM	<b>Memory (Always 0).</b> Indicates that the operational registers are mapped into memory space.

**6.5.3.12 Interrupt Line (USBC\_PCI\_INT\_LINE)**

Index 3Ch  
 Type R/W  
 Reset Value 00h

**USBC\_PCI\_INT\_LINE Register Map**

7	6	5	4	3	2	1	0
INT_LINE							

**USBC\_PCI\_INT\_LINE Bit Descriptions**

Bit	Name	Description
7:0	INT_LINE	<b>Interrupt Line.</b> This register identifies which of the system interrupt controllers the devices interrupt pin is connected to. The value of this register is used by device drivers and has no direct meaning to the USB.

**6.5.3.13 Interrupt Pin (USBC\_PCI\_INT\_PIN)**

Index            3Dh  
 Type            R/W  
 Reset Value    01h

**USBC\_PCI\_INT\_PIN Register Map**

7	6	5	4	3	2	1	0
INT_PIN							

**USBC\_PCI\_INT\_PIN Bit Descriptions**

Bit	Name	Description
7:0	INT_PIN	<b>Interrupt Pin.</b> This register identifies which interrupt pin a device uses. Since the USB uses INTA#, this value is set to 01h.

**6.5.3.14 Minimum Grant (USBC\_PCI\_MIN\_GNT)**

Index            3Eh  
 Type            R/W  
 Reset Value    00h

**USBC\_PCI\_MIN\_GNT Register Map**

7	6	5	4	3	2	1	0
MIN_GNT							

**USBC\_PCI\_MIN\_GNT Bit Descriptions**

Bit	Name	Description
7:0	MIN_GNT	<b>Minimum Grant.</b> This register specifies the desired settings for how long a burst the USB needs assuming a clock rate of 33 MHz. The value specifies a period of time in units of 1/4 microsecond.

**6.5.3.15 Maximum Latency (USBC\_PCI\_MAX\_LTNCY)**

Index            3Fh  
 Type            R/W  
 Reset Value    50h

**USBC\_PCI\_MAX\_LTNCY Register Map**

7	6	5	4	3	2	1	0
MAX_LTNCY							

**USBC\_PCI\_MAX\_LTNCY Bit Descriptions**

Bit	Name	Description
7:0	MAX_LTNCY	<b>Maximum Latency.</b> This register specifies the desired settings for how often the USB needs access to the PCI bus assuming a clock rate of 33 MHz. The value specifies a period of time in units of 1/4 microsecond.

**6.5.3.16 ASIC Test Mode Enable (USBC\_PCI\_ASIC\_TEST)**

Index 40h  
 Type R/W  
 Reset Value 000F0000h

**USBC\_PCI\_ASIC\_TEST Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**USBC\_PCI\_ASIC\_TEST Bit Descriptions**

Bit	Name	Description
31:0	RSVD	<b>Reserved.</b> These bits are reserved for internal testing only. These bits should not be written to.

**6.5.3.17 ASIC Operational Mode Enable (USBC\_PCI\_ASIC\_MODE)**

Index 44h  
 Type R/W  
 Reset Value 0000h

**USBC\_PCI\_ASIC\_MODE Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD							SIE_PIPE_DIS	RSVD							BUFF_SIZE

**USBC\_PCI\_ASIC\_MODE Bit Descriptions**

Bit	Reset	Description
15:9	RSVD	<b>Reserved.</b> Write 0s; reads undefined.
8	SIE_PIPE_DIS	<b>SIE Pipeline Disable.</b> When set, waits for all USB bus activity to complete prior to returning completion status to the List Processor. Should normally be cleared to 0. This is a fail-safe mechanism to avoid potential problems with the clk_dr transition between 1.5 MHz and 12 MHz.
7:1	RSVD	<b>Reserved.</b> Write 0s; reads undefined.
0	BUFF_SIZE	<b>Data Buffer Size.</b> When set, the size of the region for the data buffer is 16 bytes. When clear, the size is 32 bytes.

### 6.5.4 Host Controller Native Registers

#### 6.5.4.1 Host Controller Revision (USBC\_HcRevision)

USB Memory Offset 00h  
 Type HCD: RO  
 HC: RO  
 Reset Value 00000110h

USBC\_HcRevision Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														REV																	

USBC\_HcRevision Bit Descriptions

Bit	Name	HCD	HC	Description
31:8	RSVD	RO	RO	<b>Reserved.</b> Read/write 0s.
7:0	REV	RO	RO	<b>Revision (Read Only).</b> Indicates the Open HCI Specification revision number implemented by the Hardware. USB supports 1.0a specification. (X.Y = XYh).

#### 6.5.4.2 Host Controller Control (USBC\_HcControl)

USB Memory Offset 04h  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

USBC\_HcControl Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										RemoteWakeUpConnectedEnable	RemoteWakeUpConnected	InterruptRouting	HostControllerFunctionalState	BulkListEnable	ControlListEnable	IsynchronousEnable	PeriodicListEnable	ControlBulkServiceRatio													

## USBC\_HcControl Bit Descriptions

Bit	Name	HCD	HC	Description
31:11	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
10	RemoteWakeup Connected Enable	R/W	RO	<p><b>Remote Wakeup Connected Enable.</b> This bit is used by the HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the ResumeDetected bit in HcInterruptStatus is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.</p> <p>In the host controller, this bit is cleared to 0 on hardware reset and is write-read. Otherwise, it does nothing. It can be used as a flag by software as indicated above. Setting this bit will not enable or disable remote wakeup. That is covered in the Power Management Controller.</p>
9	RemoteWakeup Connected	R/W	R/W	<p><b>Remote Wakeup Connected.</b> This bit indicates whether the HC supports remote wakeup signaling. If remote wakeup is supported and used by the system it is the responsibility of system firmware to set this bit during POST. The HC clears the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.</p> <p>In the host controller, it is cleared to 0 on hardware reset and is write-read. Otherwise, it does nothing. It can be used as a flag by software as indicated above.</p>
8	InterruptRouting	R/W	RO	<p><b>Interrupt Routing.</b> This bit is used for interrupt routing.</p> <p>0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.</p>
7:6	HostController FunctionalState	R/W	R/W	<p><b>Host Controller Functional State.</b> This field sets the HC state. The HC may force a state change from UsbSuspend to UsbResume after detecting resume signaling from a downstream port. States are:</p> <p>00: UsbReset. 01: UsbResume. 10: UsbOperational. 11: UsbSuspend.</p>
5	BulkListEnable	R/W	RO	<p><b>Bulk List Enable.</b> When set, this bit enables processing of the Bulk list.</p>
4	ControlList Enable	R/W	RO	<p><b>Control List Enable.</b> When set, this bit enables processing of the Control list.</p>
3	Isochronous Enable	R/W	RO	<p><b>Isochronous Enable.</b> When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the HC will check this bit when it finds an isochronous ED.</p>
2	PeriodicList Enable	R/W	RO	<p><b>Periodic List Enable.</b> When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The HC checks this bit prior to attempting any periodic transfers in a frame.</p>
1:0	ControlBulk ServiceRatio	R/W	RO	<p><b>Control Bulk Service Ratio.</b> Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e., 00 = 1 Control Endpoint; 11 = 3 Control Endpoints).</p>

**6.5.4.3 Host Controller Command Status (USBC\_HcCommandStatus)**

USB Memory Offset 08h  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcCommandStatus Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														ScheduleOverrunCount	RSVD												OwnershipChangeRequest	BulkListFilled	ControlListFilled	HostControllerReset	

**USBC\_HcCommandStatus Bit Descriptions**

Bit	Name	HCD	HC	Description
31:18	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
17:16	Schedule OverrunCount	RO	R/W	<b>Schedule Overrun Count.</b> This field increments every time the SchedulingOverrun bit in HcInterruptStatus is set. The count wraps from 11 to 00.
15:4	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
3	Ownership ChangeRequest	R/W	R/W	<b>Ownership Change Request.</b> When set by software, this bit sets the OwnershipChange field in HcInterruptStatus. The bit is cleared by software. This field always reads back as 0.
2	BulkListFilled	R/W	R/W	<b>Bulk List Filled.</b> Set to indicate there is an active ED on the Bulk List. The bit may be set by either software or the HC and cleared by the HC each time it begins processing the head of the Bulk List.
1	ControlListFilled	R/W	R/W	<b>Control List Filled.</b> Set to indicate there is an active ED on the Control List. It may be set by either software or the HC and cleared by the HC each time it begins processing the head of the Control List.
0	HostControllerReset	R/W	R/W	<b>Host Controller Reset.</b> This bit is set to initiate a software reset. This bit is cleared by the HC upon completion of the reset operation. This field always reads back as 0.



**6.5.4.4 Host Controller Interrupt Status (USBC\_HcInterruptStatus)**

USB Memory Offset 0Ch  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

All bits are set by the hardware and cleared by software.

**USBC\_HcInterruptStatus Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD	OwnershipChange	RSVD																								RootHubStatusChange	FrameNumberOverflow	UnrecoverableError	ResumeDetected	StartOfFrame	WritebackDoneHead	SchedulingOverrun

**USBC\_HcInterruptStatus Bit Descriptions**

Bit	Name	HCD	HC	Description
31	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
30	Ownership Change	R/W	R/W	<b>Ownership Change.</b> This bit is set when the OwnershipChangeRequest bit of HcCommandStatus is set.
29:7	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
6	RootHubStatus Change	R/W	R/W	<b>Root Hub Status Change.</b> This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed.
5	FrameNumber Overflow	R/W	R/W	<b>Frame Number Overflow.</b> Set when bit 15 of FrameNumber changes value.
4	Unrecoverable Error	R/W	R/W	<b>Unrecoverable Error (Read Only).</b> This event is not implemented and is hard-coded to 0. HCD clears this bit.
3	Resume Detected	R/W	R/W	<b>Resume Detected.</b> Set when HC detects resume signaling on a downstream port.
2	StartOfFrame	R/W	R/W	<b>Start Of Frame.</b> Set when the Frame Management block signals a Start of Frame event.
1	WritebackDone Head	R/W	R/W	<b>Writeback Done Head.</b> Set after the HC has written HcDoneHead.
0	Scheduling Overrun	R/W	R/W	<b>Scheduling Overrun.</b> Set when the List Processor determines a Schedule Overrun has occurred.

**6.5.4.5 Host Controller Interrupt Enable (USBC\_HcInterruptEnable)**

USB Memory Offset 10h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

Writing a 1 to a bit in this register sets the corresponding bit, while writing a 0 leaves the bit unchanged.

**USBC\_HcInterruptEnable Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MasterInterruptEnable	OwnershipChangeEnable	RSVD																								RootHubStatusChangeEnable	FrameNumberOverflowEnable	UnrecoverableErrorEnable	ResumeDetectedEnable	StartOfFrameEnable	WritebackDoneHeadEnable	SchedulingOverrunEnable

**USBC\_HcInterruptEnable Bit Descriptions**

Bit	Name	HCD	HC	Description
31	MasterInterrupt Enable	R/W	RO	<b>Master Interrupt Enable.</b> This bit is a global interrupt enable. A write of 1 allows interrupts to be enabled via the specific enable bits listed above.
30	Ownership ChangeEnable	R/W	RO	<b>Ownership Change Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Ownership Change.
29:7	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
6	RootHubStatus ChangeEnable	R/W	RO	<b>RootHub Status Change Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Root Hub Status Change.
5	FrameNumber OverflowEnable	R/W	RO	<b>Frame Number Overflow Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Frame Number Overflow.
4	Unrecoverable ErrorEnable	R/W	RO	<b>Unrecoverable Error Enable.</b> This event is not implemented. All writes to this bit are ignored.
3	Resume DetectedEnable	R/W	RO	<b>Resume Detected Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Resume Detected.
2	StartOfFrame Enable	R/W	RO	<b>Start Of Frame Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Start of Frame.
1	WritebackDone HeadEnable	R/W	RO	<b>Writeback Done Head Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Writeback Done Head.
0	Scheduling OverrunEnable	R/W	RO	<b>Scheduling Overrun Enable.</b> 0: Ignore. 1: Enable interrupt generation due to Scheduling Overrun.

**6.5.4.6 Host Controller Interrupt Disable (USBC\_HcInterruptDisable)**

USB Memory Offset 14h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

Writing a 1 to a bit in this register clears the corresponding bit, while writing a 0 leaves the bit unchanged.

**USBC\_HcInterruptDisable Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MasterInterruptEnable	OwnershipChangeEnable	RSVD																								RootHubStatusChangeEnable	FrameNumberOverflowEnable	UnrecoverableErrorEnable	ResumeDetectedEnable	StartOfFrameEnable	WritebackDoneHeadEnable	SchedulingOverrunEnable

**USBC\_HcInterruptDisable Bit Descriptions**

Bit	Name	HCD	HC	Description
31	MasterInterrupt Enable	R/W	RO	<b>Master Interrupt Disable.</b> Global interrupt disable. A write of 1 disables all interrupts.
30	Ownership ChangeEnable	R/W	RO	<b>Ownership Change Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Ownership Change.
29:7	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
6	RootHubStatus ChangeEnable	R/W	RO	<b>Root Hub Status Change Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Root Hub Status Change.
5	FrameNumber OverflowEnable	R/W	RO	<b>Frame Number Overflow Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Frame Number Overflow.
4	Unrecoverable ErrorEnable	R/W	RO	<b>Unrecoverable Error Disable.</b> This event is not implemented. All writes to this bit will be ignored.
3	Resume DetectedEnable	R/W	RO	<b>Resume Detected Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Resume Detected.
2	StartOfFrame Enable	R/W	RO	<b>Start Of Frame Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Start of Frame.
1	WritebackDone HeadEnable	R/W	RO	<b>Writeback Done Head Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Writeback Done Head.
0	Scheduling OverrunEnable	R/W	RO	<b>Scheduling Overrun Disable.</b> 0: Ignore. 1: Disable interrupt generation due to Scheduling Overrun.

**6.5.4.7 Host Controller HCCA (USBC\_HcHCCA)**

USB Memory Offset 18h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

**USBC\_HcHCCA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCCA														RSVD																	

**USBC\_HcHCCA Bit Descriptions**

Bit	Name	HCD	HC	Description
31:8	HCCA	R/W	RO	<b>HCCA.</b> Pointer to HCCA base address.
7:0	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.

**6.5.4.8 Host Controller Current Period List ED (USBC\_HcPeriodCurrentED)**

USB Memory Offset 1Ch  
 Type HCD: RO  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcPeriodCurrentED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PeriodCurrentED																RSVD															

**USBC\_HcPeriodCurrentED Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	PeriodCurrentED	RO	R/W	<b>Period Current ED.</b> Pointer to the current Periodic List ED.
3:0	RSVD	RO	R/W	<b>Reserved.</b> Read/write 0s.

**6.5.4.9 Host Controller Control List Head ED (USBC\_HcControlHeadED)**

USB Memory Offset 20h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

**USBC\_HcControlHeadED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ControlHeadED																RSVD															

**USBC\_HcControlHeadED Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	ControlHeadED	R/W	RO	<b>Control Head ED.</b> Pointer to the Control List Head ED.
3:0	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.

**6.5.4.10 Host Controller Current Control List ED (USBC\_HcControlCurrentED)**

USB Memory Offset 24h  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcControlHeadED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ControlCurrentED																												RSVD			

**USBC\_HcControlHeadED Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	ControlCurrentED	R/W	R/W	<b>Control Current ED.</b> Pointer to the current Control List ED.
3:0	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.

**6.5.4.11 Host Controller Bulk List Head ED (USBC\_HcBulkHeadED)**

USB Memory Offset 28h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

**USBC\_HcBulkHeadED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BulkHeadED																												RSVD			

**USBC\_HcBulkHeadED Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	BulkHeadED	R/W	RO	<b>Bulk Head ED.</b> Pointer to the Bulk List Head ED.
3:0	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.

**6.5.4.12 Host Controller Current Bulk List ED (USBC\_HcBulkCurrentED)**

USB Memory Offset 2Ch  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcBulkCurrentED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BulkCurrentED																												RSVD			

**USBC\_HcBulkCurrentED Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	BulkCurrentED	R/W	R/W	<b>Bulk Current ED.</b> Pointer to the current Bulk List ED.
3:0	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.

**6.5.4.13 Host Controller Current Done List Head ED (USBC\_HcDoneHead)**

USB Memory Offset 30h  
 Type HCD: RO  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcDoneHead Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DoneHead																												RSVD			

**USBC\_HcDoneHead Bit Descriptions**

Bit	Name	HCD	HC	Description
31:4	DoneHead	RO	R/W	<b>Done Head.</b> Pointer to the current Done List Head ED.
3:0	RSVD	RO	R/W	<b>Reserved.</b> Read/write 0s.

**6.5.4.14 Host Controller Frame Interval (USBC\_HcFmInterval)**

USB Memory Offset 34h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00002EDFh

**USBC\_HcFmInterval Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrameIntervalToggle	FSLargestDataPacket															RSVD		FrameInterval													

**USB\_HcFmInterval Bit Descriptions**

Bit	Name	HCD	HC	Description
31	FrameIntervalToggle	R/W	RO	<b>Frame Interval Toggle</b> This bit is toggled by HCD when it loads a new value into FrameInterval.
30:16	FSLargestDataPacket	R/W	RO	<b>FS Largest Data Packet.</b> This field specifies a value that is loaded into the Largest Data Packet Counter at the beginning of each frame.
15:14	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
13:0	FrameInterval	R/W	RO	<b>Frame Interval.</b> This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

**6.5.4.15 Host Controller Frame Remaining (USBC\_HcFrameRemaining)**

USB Memory Offset 38h  
 Type HCD: RO  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcFrameRemaining Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FrameRemainingToggle	RSVD														FrameRemaining																

**USBC\_HcFrameRemaining Bit Descriptions**

Bit	Name	HCD	HC	Description
31	FrameRemainingToggle	RO	R/W	<b>Frame Remaining Toggle.</b> Loaded with FrameIntervalToggle when FrameRemaining is loaded.
30:14	RSVD	RO	R/W	<b>Reserved.</b> Read/write 0s.
13:0	FrameRemaining	RO	R/W	<b>Frame Remaining.</b> When the HC is in the UsbOperational state, this 14-bit field decrements each 12 MHz clock period. When the count reaches 0, (end of frame) the counter reloads with FrameInterval. In addition, the counter loads when the HC transitions into UsbOperational.

**6.5.4.16 Host Controller Frame Number (USBC\_HcFmNumber)**

USB Memory Offset 3Ch  
 Type HCD: RO  
 HC: R/W  
 Reset Value 00000000h

**USBC\_HcFmNumber Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																FrameNumber															

**USBC\_HcFmNumber Bit Descriptions**

Bit	Name	HCD	HC	Description
30:16	RSVD	RO	R/W	<b>Reserved.</b> Read/write 0s.
15:0	FrameNumber	RO	R/W	<b>Frame Number.</b> This 16-bit incrementing counter field is incremented coincident with the loading of FrameRemaining. The count rolls over from FFFFh to 0h.

**6.5.4.17 Host Controller Periodic Start (USBC\_HcPeriodicStart)**

USB Memory Offset 40h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

**USBC\_HcPeriodicStart Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														PeriodicStart																	

**USBC\_HcPeriodicStart Bit Descriptions**

Bit	Name	HCD	HC	Description
31:14	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
13:0	PeriodicStart	R/W	RO	<b>Periodic Start.</b> This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

**6.5.4.18 Host Controller Low Speed Threshold (USBC\_HcLSThreshold)**

USB Memory Offset 44h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000628h

**USBC\_HcLSThreshold Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														LSThreshold																	

**USBC\_HcLSThreshold Bit Descriptions**

Bit	Name	HCD	HC	Description
30:12	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
11:0	LSThreshold	R/W	RO	<b>LS Threshold.</b> This field contains a value used by the Frame Management block to determine whether or not a low speed transaction can be started in the current frame.



**6.5.4.19 Host Controller Root Hub Descriptor A (USBC\_HcRhDescriptorA)**

USB Memory Offset 48h  
 Type HCD: R/W  
 HC: RO  
 Reset Value 01000002h

This register is only reset by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.

**USBC\_HcRhDescriptorA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PowerOnToPowerGoodTime								RSVD												NumberDownstreamPorts											
												NoOverCurrentProtection	OverCurrentProtectionMode	DeviceType	NoPowerSwitching	PowerSwitchingMode															

**USBC\_HcRhDescriptorA Bit Descriptions**

Bit	Name	HCD	HC	Description
31:24	PowerOnToPowerGoodTime	R/W	RO	<b>PowerOn To PowerGood Time.</b> This field value is represented as the number of 2 ms intervals, ensuring that the power switching is effective within 2 ms. Only bits [25:24] are implemented as R/W. The remaining bits are read only as 0. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support system implementation. This field should always be written to a non-zero value.
23:13	RSVD	R/W	RO	<b>Reserved.</b> Read/write 0s.
12	NoOverCurrentProtection	R/W	RO	<b>No Over-Current Protection.</b> This bit should be written to support the external system port over-current implementation. 0: Over-current status is reported. 1: Over-current status is not reported.
11	OverCurrentProtectionMode	R/W	RO	<b>Over Current Protection Mode.</b> This bit should be written 0 and is only valid when NoOverCurrentProtection is cleared. 0: Global Over-Current. 1: Individual Over-Current
10	DeviceType	RO	RO	<b>Device Type (Read Only).</b> USB is not a compound device, therefore this field will return 0.
9	NoPowerSwitching	R/W	RO	<b>No Power Switching.</b> This bit should be written to support the external system port power switching implementation. 0: Ports are power switched. 1: Ports are always powered on.
8	PowerSwitchingMode	R/W	RO	<b>Power Switching Mode.</b> This bit is only valid when NoPowerSwitching is cleared. This bit should be written 0. 0: Global Switching. 1: Individual Switching
7:0	NumberDownstreamPorts	RO	RO	<b>Number Downstream Ports (Read Only).</b> USB supports two downstream ports, therefore this field will return 02h.

**6.5.4.20 Host Controller Root Hub Descriptor B (USBC\_HcRhDescriptorB)**

USB Memory Offset 4Ch  
 Type HCD: R/W  
 HC: RO  
 Reset Value 00000000h

This register is only reset by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.

**USBC\_HcRhDescriptorB Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PortPowerControlMask																DeviceRemoveable															

**USBC\_HcRhDescriptorB Bit Descriptions**

Bit	Name	HCD	HC	Description
31:16	PortPower ControlMask	R/W	RO	<p><b>Port Power Control Mask.</b> Global-power switching. This field is only valid if NoPowerSwitching is cleared and PowerSwitching-Mode is set (individual port switching). When set, the port only responds to individual port power switching commands (Set/ClearPortPower). When cleared, the port only responds to global power switching commands (Set/ClearGlobalPower).</p> <p>0: Device not removable.                      1: Global-power mask.</p> <p>Port Bit relationship - Unimplemented ports are reserved, read/write 0.</p> <p>0: Reserved                      1: Port 1                      2: Port 2                      ...                      15: Port 15</p>
15:0	Device Removeable	R/W	RO	<p><b>Device Removeable.</b> USB ports default to removable devices.</p> <p>0: Device not removable.                      1: Device removable.</p> <p>Port Bit relationship - Unimplemented ports are reserved, read/write 0.</p> <p>0: Reserved                      1: Port 1                      2: Port 2                      ...                      15: Port 15</p>

**6.5.4.21 Host Controller Root Hub Status (USBC\_HcRhStatus)**

USB Memory Offset 50h  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

This register is reset by the UsbReset state.

**Note:** Read back are 0s.

**USBC\_HcRhStatus Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ClearRemoteWakeupEnable	RSVD													OverCurrentIndicatorChange	Dual Function	Dual Function	RSVD													OverCurrentIndicator	Dual Function

**USBC\_HcRhStatus Bit Descriptions**

Bit	Name	HCD	HC	Description
31	ClearRemote WakeupEnable	WO	RO	<b>Clear Remote Wakeup Enable.</b> Writing a 1 to this bit clears DeviceRemoteWakeupEnable. Writing a 0 has no effect.
30:18	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
17	OverCurrent IndicatorChange	R/W	R/W	<b>Over Current Indicator Change.</b> This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
16	Dual Function	R/W	RO	<b>Read: Local Power Status Change.</b> Not supported. Always read 0. <b>Write: Set Global Power.</b> Write a 1 issues a SetGlobalPower command to the ports. Writing a 0 has no effect.
15	Dual Function	R/W	RO	<b>Read: Device Remote Wakeup Enable.</b> This bit enables ports' ConnectStatusChange as a remote wakeup event. 0: Disabled. 1: Enabled. <b>Write: Set Remote Wakeup Enable.</b> Writing a 1 sets DeviceRemoteWakeupEnable. Writing a 0 has no effect.
14:2	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
1	OverCurrent Indicator	R	R/W	<b>Over Current Indicator.</b> This bit reflects the state of the OVR-CUR pin. This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared. 0: No over-current condition. 1: Over-current condition.
0	Dual Function	R/W	RO	<b>Read: Local Power Status.</b> Not Supported. Always read 0. <b>Write: Clear Global Power.</b> Writing a 1 issues a ClearGlobalPower command to the ports. Writing a 0 has no effect.

**6.5.4.22 Host Controller Root Hub Port Status 1 (USBC\_HcRhPortStatus[1])**

USB Memory Offset 54h  
 Type HCD: R/W  
 HC: R/W  
 Reset Value 00000000h

This register is reset by the UsbReset state.

**Note:** There must be a Device Connect detected by the host to be able to set other bits.

**USBC\_HcRhPortStatus[1] Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											PortResetStatusChange	PortOverCurrentIndicatorChange	PortSuspendStatusChange	PortEnableStatusChange	ConnectStatusChange	RSVD						Dual Function	Dual Function	RSVD			Dual Function	Dual Function	Dual Function	Dual Function	Dual Function

**USBC\_HcRhPortStatus[1] Bit Descriptions**

Bit	Name	HCD	HC	Description
31:21	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
20	PortReset StatusChange	R/W	R/W	<b>Port Reset Status Change.</b> This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	PortOverCurrent IndicatorChange	R/W	R/W	<b>Port Over Current Indicator Change.</b> This bit is set when Over-CurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	PortSuspend StatusChange	R/W	R/W	<b>Port Suspend Status Change.</b> This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	PortEnable StatusChange	R/W	R/W	<b>Port Enable Status Change.</b> This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.
16	ConnectStatus Change	R/W	R/W	<b>Connect Status Change.</b> This bit indicates a connect or disconnect event has been detected. Writing a 1 clears this bit. Writing a 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. If DeviceRemoveable is set, this bit resets to 1.
15:10	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.

**USBC\_HcRhPortStatus[1] Bit Descriptions (Continued)**

Bit	Name	HCD	HC	Description
9	Dual Function	R/W	R/W	<p><b>Read: Low Speed Device Attached.</b> This bit defines the speed of the attached device. It is only valid when CurrentConnectStatus is set.</p> <p>0: Full Speed device. 1: Low Speed device.</p> <p><b>Write: Clear Port Power.</b> Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.</p>
8	Dual Function	R/W	R/W	<p><b>Read: Port Power Status.</b> This bit reflects the power state of the port regardless of the power switching mode.</p> <p>0: Port power is off. 1: Port power is on.</p> <p><b>Note:</b> If NoPowerSwitching is set, this bit is always read as 1.</p> <p><b>Write: Set Port Power.</b> Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.</p>
7:5	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
4	Dual Function	R/W	R/W	<p><b>Read: Port Reset Status.</b></p> <p>0: Port reset signal is not active. 1: Port reset signal is active.</p> <p><b>Write: Set Port Reset.</b> Writing a 1 sets PortResetStatus. Writing a 0 has no effect.</p>
3	Dual Function	R/W	R/W	<p><b>Read: Port Over Current Indicator.</b> This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtection-Mode is set.</p> <p>0: No over-current condition. 1: Over-current condition.</p> <p><b>Write: Clear Port Suspend.</b> Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.</p>
2	Dual Function	R/W	R/W	<p><b>Read: Port Suspend Status.</b></p> <p>0: Port is not suspended. 1: Port is selectively suspended.</p> <p><b>Write: Set Port Suspend.</b> Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.</p>
1	Dual Function	R/W	R/W	<p><b>Read: Port Enable Status.</b></p> <p>0: Port disabled. 1: Port enabled.</p> <p><b>Write: Set PortEnable.</b> Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.</p>
0	Dual Function	R/W	R/W	<p><b>Read: Current Connect Status.</b></p> <p>0: No device connected. 1: Device connected.</p> <p><b>Note:</b> If DeviceRemoveable is set (not removable) this bit is always 1.</p> <p><b>Write: Clear Port Enable.</b> Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.</p>

**6.5.4.23 Host Controller Root Hub Port Status 2 (USBC\_HcRhPortStatus[2])**

USB Memory Offset 58h

Type HCD: R/W

HC: R/W

Reset Value 00000000h

This register is reset by the UsbReset state.

**Note:** There must be a Device Connect detected by the host to be able to set other bits.**USBC\_HcRhPortStatus[2] Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											PortResetStatusChange	PortOverCurrentIndicatorChange	PortSuspendStatusChange	PortEnableStatusChange	ConnectStatusChange	RSVD						Dual Function	Dual Function	RSVD			Dual Function	Dual Function	Dual Function	Dual Function	Dual Function

**USBC\_HcRhPortStatus[2] Bit Descriptions**

Bit	Name	HCD	HC	Description
31:21	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
20	PortReset StatusChange	R/W	R/W	<b>Port Reset Status Change.</b> This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	PortOverCurrent IndicatorChange	R/W	R/W	<b>Port Over Current Indicator Change.</b> This bit is set when Over-CurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	PortSuspend StatusChange	R/W	R/W	<b>Port Suspend Status Change.</b> This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	PortEnable StatusChange	R/W	R/W	<b>Port Enable Status Change.</b> This bit indicates that the port has been disabled due to a hardware event (cleared PortEnableStatus). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.
16	ConnectStatus Change	R/W	R/W	<b>Connect Status Change.</b> This bit indicates a connect or disconnect event has been detected. Writing a 1 clears this bit. Writing a 0 has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. If DeviceRemoveable is set, this bit resets to 1.
15:10	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.

**USBC\_HcRhPortStatus[2] Bit Descriptions (Continued)**

Bit	Name	HCD	HC	Description
9	Dual Function	R/W	R/W	<p><b>Read: Low Speed Device Attached.</b> This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set.</p> <p>0: Full speed device. 1: Low Speed device.</p> <p><b>Write: Clear Port Power.</b> Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.</p>
8	Dual Function	R/W	R/W	<p><b>Read: Port Power Status.</b> This bit reflects the power state of the port regardless of the power switching mode.</p> <p>0: Port power is off. 1: Port power is on.</p> <p><b>Note:</b> If NoPowerSwitching is set, this bit is always read as 1.</p> <p><b>Write: Set Port Power.</b> Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.</p>
7:5	RSVD	R/W	R/W	<b>Reserved.</b> Read/write 0s.
4	Dual Function	R/W	R/W	<p><b>Read: Port Reset Status.</b></p> <p>0: Port reset signal is not active. 1: Port reset signal is active.</p> <p><b>Write: Set Port Reset.</b> Writing a 1 sets PortResetStatus. Writing a 0 has no effect.</p>
3	Dual Function	R/W	R/W	<p><b>Read: Port Over Current Indicator.</b> This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtection-Mode is set.</p> <p>0: No over-current condition. 1: Over-current condition.</p> <p><b>Write: Clear Port Suspend.</b> Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.</p>
2	Dual Function	R/W	R/W	<p><b>Read: Port Suspend Status.</b></p> <p>0: Port is not suspended. 1: Port is selectively suspended.</p> <p><b>Write: Set Port Suspend.</b> Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.</p>
1	Dual Function	R/W	R/W	<p><b>Read: Port Enable Status.</b></p> <p>0: Port disabled. 1: Port enabled.</p> <p><b>Write: Set Port Enable.</b> Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.</p>
0	Dual Function	R/W	R/W	<p><b>Read: Current Connect Status.</b></p> <p>0: No device connected. 1: Device connected.</p> <p>If DeviceRemoveable is set (not removable) this bit is always 1.</p> <p><b>Write: Clear Port Enable.</b> Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.</p>

## 6.6 Diverse Integration Logic Register Descriptions

All registers associated with Diverse Integration Logic (DIVIL) are MSRs:

- Standard GeodeLink Device (GLD) MSRs
- DIVIL Specific MSRs

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, some DIVIL MSRs are called out as 32 bits. The DIVIL (DD) treats writes to the upper 32 bits (i.e., bits [63:32]) of the 32-bit MSRs as don't cares and always returns 0 on these bits.

The Standard GeodeLink Device MSRs are summarized in Table 6-17 and the DIVIL Specific MSRs are summarized in Table 6-18. The reference column in the tables point to the page where the register maps and bit descriptions are listed. Some notations in the reference column also point to other chapters. These MSRs are physically located in the DIVIL, but the descriptions are documented with the associated module and are listed here only for completeness.

**Table 6-17. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400000h	RO	GLD Capabilities MSR (DIVIL_GLD_MSR_CAP)	00000000_002DF0xxh	Page 317
51400001h	R/W	GLD Master Configuration MSR (DIVIL_GLD_MSR_CONFIG)	00000000_0000F000h	Page 317
51400002h	R/W	GLD SMI MSR (DIVIL_GLD_MSR_SMI)	00000000_00000000h	Page 318
51400003h	R/W	GLD Error MSR (DIVIL_GLD_MSR_ERROR)	00000000_00000000h	Page 321
51400004h	R/W	GLD Power Management MSR (DIVIL_GLD_MSR_PM)	00000000_00000000h	Page 323
51400005h	R/W	GLD Diagnostic MSR (DIVIL_GLD_MSR_DIAG)	00000000_00000000h	Page 324
51400006h-51400007h	R/W	DD Reserved MSRs (DD_MSR_RSVD) (Reads return 1; writes have no effect.)	FFFFFFFF_FFFFFFFFh	---

**Table 6-18. DIVIL Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400008h	R/W	Local BAR - IRQ Mapper (DIVIL_LBAR_IRQ) I/O Space - Use of this LBAR is optional. IRQ Mapper is always accessible via MSR space.	00000000_00000000h	Page 325
51400009h	R/W	Local BAR - KEL from USB Host Controller 1 (DIVIL_LBAR_KEL1) Memory Space - First of two ways to access KEL. KEL operation is not dependent on KEL1 or KEL2 access.	00000000_00000000h	Page 326
5140000Ah	R/W	Local BAR - KEL from USB Host Controller 2 (DIVIL_LBAR_KEL2) Memory Space - Second of two ways to access KEL. KEL operation is not dependent on KEL1 or KEL2 access.	00000000_00000000h	Page 326
5140000Bh	R/W	Local BAR - SMB (DIVIL_LBAR_SMB) I/O Space - Local Base Address Register for SMB Controller native registers.	00000000_00000000h	Page 327



Table 6-18. DIVIL Specific MSRs Summary (Continued)

MSR Address	Type	Register Name	Reset Value	Reference
5140000Ch	R/W	Local BAR - GPIO and ICFs (DIVIL_LBAR_GPIO) I/O Space - Local Base Address Register for GPIOs and ICFs.	00000000_00000000h	Page 328
5140000Dh	R/W	Local BAR - MFGPTs (DIVIL_LBAR_MFGPT) I/O Space - Local Base Address Register for MFGPTs.	00000000_00000000h	Page 329
5140000Eh	R/W	Local BAR - ACPI (DIVIL_LBAR_ACPI) I/O Space - Local Base Address Register for MFGPTs.	00000000_00000000h	Page 330
5140000Fh	R/W	Local BAR - Power Management Support (DIVIL_LBAR_PMS) I/O Space - Local Base Address Register for Power Management Support registers.	00000000_00000000h	Page 331
51400010h	R/W	Local BAR - Flash Chip Select 0 (DIVIL_LBAR_FLSH0) Local Base Address Register for Flash Controller, Chip Select 0.	00000000_00000000h	Page 332
51400011h	R/W	Local BAR - Flash Chip Select 1 (DIVIL_LBAR_FLSH1) Local Base Address Register for Flash Controller, Chip Select 1.	00000000_00000000h	Page 332
51400012h	R/W	Local BAR - Flash Chip Select 2 (DIVIL_LBAR_FLSH2) Local Base Address Register for Flash Controller, Chip Select 2.	00000000_00000000h	Page 332
51400013h	R/W	Local BAR - Flash Chip Select 3 (DIVIL_LBAR_FLSH3) Local Base Address Register for Flash Controller, Chip Select 3.	00000000_00000000h	Page 332
51400014h	R/W	Legacy I/O Space Controls (DIVIL_LEG_IO) Legacy I/O space controls.	04000003h	Page 334
51400015h	R/W	Ball Options Control (DIVIL_BALL_OPTS) Controls IDE and LPC pin options.	00000x7xh	Page 335
51400016h	R/W	Soft IRQ (DIVIL_SOFT_IRQ) Software generated IRQ.	00000000h	Page 337
51400017h	R/W	Soft Reset (DIVIL_SOFT_RESET) Software generated RESET.	00000000h	Page 337
51400018h	R/W	NOR Flash Control (NORF_CTL)	00000000h	Page 520 (Flash spec)
51400019h	R/W	NOR Flash Timing for Chip Selects 0 and 1 (NORTF_T01)	07770777h	Page 522 (Flash spec)
5140001Ah	R/W	NOR Flash Timing for Chip Selects 2 and 3 (NORTF_T23)	07770777h	Page 523 (Flash spec)

Table 6-18. DIVIL Specific MSRs Summary (Continued)

MSR Address	Type	Register Name	Reset Value	Reference
5140001Bh	R/W	NAND Flash Data Timing MSR (NANDF_DATA)	07770777h	Page 524 (Flash spec)
5140001Ch	R/W	NAND Flash Control Timing (NANDF_CTL)	00000777h	Page 525 (Flash spec)
5140001Dh	R/W	Flash Reserved (NANDF_RSVD)	00000000h	Page 525 (Flash spec)
5140001Eh	R/W	Access Control DMA Request (DIVIL_AC_DMA)	00000000h	Page 338
5140001Fh	R/W	Keyboard Emulation Logic Control Register (KELX_CTL)	00000010h	Page 365 (KEL spec)
51400020h	R/W	IRQ Mapper Unrestricted Y and Z Select Low (PIC_[Y/Z]SEL_LOW)	00000000h	Page 350 (PIC spec)
51400021h	R/W	IRQ Mapper Unrestricted Y Select High (PIC_YSEL_HIGH)	00000000h	Page 351 (PIC spec)
51400022h	R/W	IRQ Mapper Unrestricted Z Select Low (PIC_ZSEL_LOW)	00000000h	Page 350 (PIC spec)
51400023h	R/W	IRQ Mapper Unrestricted Z Select High (PIC_ZSEL_HIGH)	00000000h	Page 351 (PIC spec)
51400024h	R/W	IRQ Mapper Primary Mask (PIC_IRQM_PRIM)	0000FFFFh	Page 352 (PIC spec)
51400025h	R/W	IRQ Mapper LPC Mask (PIC_IRQM_LPC)	00000000h	Page 352 (PIC spec)
51400026h	RO	IRQ Mapper Extended Interrupt Request Status Low (PIC_XIRR_STS_LOW)	xxxxxxxh	Page 353 (PIC spec)
51400027h	RO	IRQ Mapper Extended Interrupt Request Status High (PIC_XIRR_STS_HIGH)	xxxxxxxh	Page 354 (PIC spec)
51400028h	R/W	MFGPT IRQ Mask (MFGPT_IRQ)	00000000h	Page 484 (MFGPT spec)
51400029h	R/W	MFGPT NMI and Reset Mask (MFGPT_NR)	00000000h	Page 487 (MFGPT spec)
5140002Ah	R/W	MFGPT Reserved (MFGPT_RSVD)	00000000h	Page 488 (MFGPT spec)
5140002Bh	WO	MFGPT Clear Setup Test (MFGPT_SETUP)	00000000h	Page 488 (MFGPT spec)
5140002Ch- 5140002Fh	R/W	Reserved. Reads return 1. Writes have no effect.	FFFFFFFFh	---
51400030h	RO	Floppy Port 3F2h Shadow (FLPY_3F2_SHDW)	xxh	Page 339 (Floppy spec)
51400031h	RO	Floppy Port 3F7h Shadow (FLPY_3F7_SHDW)	xxh	Page 339 (Floppy spec)
51400032h	RO	Floppy Port 372h Shadow (FLPY_372_SHDW)	xxh	Page 340 (Floppy spec)
51400033h	RO	Floppy Port 377h Shadow (FLPY_377_SHDW)	xxh	Page 340 (Floppy spec)
51400034h	RO	PIC Shadow (PIC_SHDW)	xxh	Page 354 (PIC spec)
51400035h	R/W	Reserved. Reads return 1. Writes have no effect.	FFFFFFFFh	---

Table 6-18. DIVIL Specific MSRs Summary (Continued)

MSR Address	Type	Register Name	Reset Value	Reference
51400036h	RO	PIT Shadow (PIT_SHDW)	00h	Page 342 (PIT spec)
51400037h	R/W	PIT Count Enable (PIT_CNTRL)	03h	Page 342 (PIT spec)
51400038h	R/W	UART1 Primary Dongle and Modem Interface (UART[1]_MOD)	0xh	Page 380 (UART spec)
51400039h	R/W	UART1 Secondary Dongle and Status (UART[1]_DONG)	xxh	Page 381 (UART spec)
5140003Ah	R/W	UART1 Interface Configuration (UART[1]_CONF)	42h	Page 382 (UART spec)
5140003Bh	R/W	UART1 Reserved MSR (UART1_RSVD_MSR) - Reads return 1; writes have no effect.	11h	---
5140003Ch	R/W	UART2 Primary Dongle and Modem Interface (UART[2]_MOD)	0xh	Page 380 (UART spec)
5140003Dh	R/W	UART2 Secondary Dongle and Status (UART[2]_DONG)	xxh	Page 381 (UART spec)
5140003Eh	R/W	UART2 Interface Configuration (UART[2]_CONF)	42h	Page 382 (UART spec)
5140003Fh	R/W	UART2 Reserved MSR (UART2_RSVD_MSR) - Reads return 1; writes have no effect.	11h	---
51400040h	R/W	DMA Mapper (DMA_MAP)	0000h	Page 420 (DMA spec)
51400041h	RO	DMA Shadow Channel 0 Mode (DMA_SHDW_CH0)	00xxh	Page 421 (DMA spec)
51400042	RO	DMA Shadow Channel 1 Mode (DMA_SHDW_CH1)	00xxh	Page 421 (DMA spec)
51400043	RO	DMA Shadow Channel 2 Mode (DMA_SHDW_CH2)	00xxh	Page 421 (DMA spec)
51400044	RO	DMA Shadow Channel 3 Mode (DMA_SHDW_CH3)	00xxh	Page 421 (DMA spec)
51400045h	RO	DMA Shadow Channel 4 Mode (DMA_SHDW_CH4)	00xxh	Page 421 (DMA spec)
51400046h	RO	DMA Shadow Channel 5 Mode (DMA_SHDW_CH5)	00xxh	Page 421 (DMA spec)
51400047h	RO	DMA Shadow Channel 6 Mode (DMA_SHDW_CH6)	00xxh	Page 421 (DMA spec)
51400048h	RO	DMA Shadow Channel 7 Mode (DMA_SHDW_CH7)	00xxh	Page 421 (DMA spec)
51400049h	RO	DMA Shadow Mask (DMA_MSK_SHDW)	00FFh	Page 422 (DMA spec)
5140024Ah	R/W	Reserved MSR (RSVD_MSR) - Reads return 1; writes have no effect.	11h	---
5140024Bh	R/W	Reserved MSR (RSVD_MSR) - Reads return 1; writes have no effect.	11h	---
5140004Ch	RO	LPC Address Error (LPC_EADDR)	00000000h	Page 435 (LPC spec)

Table 6-18. DIVIL Specific MSRs Summary (Continued)

MSR Address	Type	Register Name	Reset Value	Reference
5140004Dh	RO	LPC Error Status (LPC_ESTAT)	00000000h	Page 436 (LPC spec)
5140004Eh	R/W	LPC Serial IRQ Control (LPC_SIRQ)	00000000h	Page 436 (LPC spec)
5140004Fh	R/W	LPC Reserved (LPC_RSVD)	00000000h	Page 437 (LPC spec)
51400050h	R/W	PMC Logic Timer (PMC_LTMR)	00000000h	Page 496 (PMC spec)
51400051h	R/W	PMC Reserved (PMC_RSVD)	00000000h	Page 496 (PMC spec)
51400052h- 51400053h	R/W	Reserved MSR (RSVD_MSR) - Reads return 1; writes have no effect.	11h	---
51400054h	R/W	RTC RAM Lock (RTC_RAM_LOCK)	00h	Page 439 (RTC spec)
51400055h	R/W	RTC Date of Month Alarm Offset (RTC_DOMA_OFFSET)	00h	Page 440 (RTC spec)
51400056h	R/W	RTC Month Alarm Offset (RTC_MONA_OFFSET)	00h	Page 440 (RTC spec)
51400057h	R/W	RTC Century Offset (RTC_CEN_OFFSET)	00h	Page 441 (RTC spec)
51400058h- 514000FFh	R/W	Reserved MSR (RSVD_MSR) - Reads return 1; writes have no effect.	11h	---

### 6.6.1 Standard GeodeLink™ Device (GLD) MSRs

#### 6.6.1.1 GLD Capabilities MSR (DIVIL\_GLD\_MSR\_CAP)

MSR Address 51400000h  
 Type RO  
 Reset Value 00000000\_002DF0xxh

**DIVIL\_GLD\_MSR\_CAP Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEV_ID														REV_ID									

**DIVIL\_GLD\_MSR\_CAP Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads return 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2DF0h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

#### 6.6.1.2 GLD Master Configuration MSR (DIVIL\_GLD\_MSR\_CONFIG)

MSR Address 51400001h  
 Type R/W  
 Reset Value 00000000\_0000F000h

**DIVIL\_GLD\_MSR\_CONFIG Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
RSVD																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD													FIX_PREFETCH		DISCARD		NON_COH_WR		NON_COH_RD		RSVD						PRI		RSVD		PID	

**DIVIL\_GLD\_MSR\_CONFIG Bit Descriptions**

Bit	Name	Description
63:19	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
18:16	FIX_PREFETCH	<b>Fixed Read Prefetch Policy.</b> 000: None. Each read takes a complete trip to memory. 001: Initial read 08 bytes. Read next 8 only when requested. 010: Initial read 16 bytes. Read next 16 only when requested. 011: Initial read 32 bytes. Read next 32 only when requested. 100: Initial read 32 bytes. Read next 32 when 16 bytes left. 101, 110, and 111: Reserved.

**DIVIL\_GLD\_MSR\_CONFIG Bit Descriptions (Continued)**

Bit	Name	Description
15:14	DISCARD	<b>Read Prefetch Discard Policy.</b> 00: Reserved. 01: Discard all data not taken under current local bus grant. 10: Discard all data on any local bus transaction. 11: Discard all data on any local bus write transaction. Always use this value.
13	NON_COH_WR	<b>Non-Coherent Write.</b> 0: Write requests are coherent. 1: Write requests are non-coherent. Always use this value.
12	NON_COH_RD	<b>Non-Coherent Read.</b> 0: Read requests are coherent. 1: Read requests are non-coherent. Always use this value.
11:7	RSVD	<b>Reserved.</b> Reads as 0.
6:4	PRI	<b>Priority Level.</b> Always write 0.
3	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0.
2:0	PID	<b>Priority ID.</b> Always write 0.

**6.6.1.3 GLD SMI MSR (DIVIL\_GLD\_MSR\_SMI)**

MSR Address 51400002h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further SMI/ASMI generation details.)

**DIVIL\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32																
RSVD																PM1_CNT_SSMI_FLAG	PM2_CNT_SSMI_FLAG	KEL_A20_ASMI_FLAG	DMA_SSMI_FLAG	LPC_SSMI_FLAG	RSVD	UART2_SSMI_FLAG	UART1_SSMI_FLAG	PORTA_INIT_ASMI_FLAG	PORTA_A20_ASMI_FLAG	KEL_INIT_ASMI_FLAG	PM_ASMI_FLAG	PIC_ASMI_FLAG	KEL_EE_ASMI_FLAG	SHTDWN_ASMI_FLAG	HLT_ASMI_FLAG																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	PM1_CNT_SSMI_EN	PM2_CNT_SSMI_EN	KEL_A20_ASMI_EN	DMA_SSMI_EN	LPC_SSMI_EN	RSVD	UART2_SSMI_EN	UART1_SSMI_EN	PORTA_INIT_ASMI_EN	PORTA_A20_ASMI_EN	KEL_INIT_ASMI_EN	PM_ASMI_EN	PIC_ASMI_EN	KEL_EE_ASMI_EN	SHTDWN_ASMI_EN	HLT_ASMI_EN																

## DIVIL\_GLD\_MSR\_SMI Bit Descriptions

Bit	Name	Description
63:48	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
47	PM1_CNT_SSMI_FLAG	<b>Power Management 1 Control Register SSMI Flag.</b> If high, records that an SSMI was generated on a write to PM1_CNT (ACPI I/O Offset 08h). Write 1 to clear; writing 0 has no effect. PM1_CNT_SSMI_EN (bit 15) must be high to generate SSMI and set flag.
46	PM2_CNT_SSMI_FLAG	<b>Power Management 2 Control Register SSMI Flag.</b> If high, records that an SSMI was generated on a write to PM2_CNT (ACPI I/O Offset 0Ch). Write 1 to clear; writing 0 has no effect. PM2_CNT_SSMI_EN (bit 14) must be high to generate SSMI and set flag.
45	KEL_A20_ASMI_FLAG	<b>KEL Gate A20 ASMI Flag.</b> If high, records that an ASMI was generated in the KEL due to keyboard gate A20 signal change. Write 1 to clear; writing 0 has no effect. KELA20_ASMI_EN (bit 13) must be high to generate ASMI and set flag.
44	DMA_SSMI_FLAG	<b>DMA SSMI Flag.</b> If high, records that an SSMI was generated on the 8237s during DMA. Write 1 to clear; writing 0 has no effect. DMA_SSMI_EN (bit 12) must be high to generate SSMI and set flag. (Also see Section 6.6.2.14 "Access Control DMA Request (DIVIL_AC_DMA)" on page 338.)
43	LPC_SSMI_FLAG	<b>LPC SSMI Flag.</b> If high, records that an SSMI was generated on the LPC during DMA. Write 1 to clear; writing 0 has no effect. LPC_SSMI_EN (bit 11) must be high to generate SSMI and set flag. (Also see Section 6.6.2.14 "Access Control DMA Request (DIVIL_AC_DMA)" on page 338.)
42	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
41	UART2_SSMI_FLAG	<b>UART2 SSMI Flag.</b> If high, records that an SSMI was generated on UART2 during DMA. Write 1 to clear; writing 0 has no effect. UART2_SSMI_EN (bit 9) must be high to generate SSMI and set flag. (Also see Section 6.6.2.14 "Access Control DMA Request (DIVIL_AC_DMA)" on page 338.)
40	UART1_SSMI_FLAG	<b>UART1 SSMI Flag.</b> If high, records that an SSMI was generated on UART1 during DMA. Write 1 to clear; writing 0 has no effect. UART1_SSMI_EN (bit 8) must be high to generate SSMI and set flag. (Also see Section 6.6.2.14 "Access Control DMA Request (DIVIL_AC_DMA)" on page 338.)
39	PORTA_INIT_ASMI_FLAG	<b>Port A INIT ASMI Flag.</b> If high, records that an ASMI was generated in the KEL due to an INIT on Port A (092h). Write 1 to clear; writing 0 has no effect. PORTA_INIT_ASMI_EN (bit 7) must be high to generate ASMI and set flag.
38	PORTA_A20_ASMI_FLAG	<b>Port A A20 ASMI Flag.</b> If high, records that an ASMI was generated in the KEL due to a A20 change on Port A (092h). Write 1 to clear; writing 0 has no effect. PORTA_A20_ASMI_EN (bit 6) must be high to generate ASMI and set flag.
37	KEL_INIT_ASMI_FLAG	<b>KEL INIT ASMI Flag.</b> If high, records that an ASMI was generated in the KEL due to a keyboard INIT sequence. Write 1 to clear; writing 0 has no effect. KEL_INIT_ASMI_EN (bit 5) must be high to generate ASMI and set flag.
36	PM_ASMI_FLAG	<b>Power Management ASMI Flag.</b> If high, records that an ASMI was generated in the Power Management Logic. Write 1 to clear; writing 0 has no effect. PM_ASMI_EN (bit 4) must be high to generate ASMI and set flag.
35	PIC_ASMI_FLAG	<b>PIC ASMI Flag.</b> If high, records that an ASMI was generated in the Extended PIC Mapper. Write 1 to clear; writing 0 has no effect. PIC_ASMI_EN (bit 3) must be high to generate ASMI and set flag.
34	KEL_EE_ASMI_FLAG	<b>KEL Emulation Event ASMI Flag.</b> If high, records that an ASMI was generated in the KEL due to a KEL emulation event. Write 1 to clear; writing 0 has no effect. KEL_EE_ASMI_EN (bit 2) must be high to generate ASMI and set flag.
33	SHTDWN_ASMI_FLAG	<b>Shutdown ASMI Flag.</b> If high, records that an ASMI was generated on a Shutdown special cycle. Write 1 to clear; writing 0 has no effect. SHTDWN_ASMI_EN (bit 1) must be high to generate ASMI and set flag.

## DIVIL\_GLD\_MSR\_SMI Bit Descriptions (Continued)

Bit	Name	Description
32	HLT_ASMI_FLAG	<b>Halt ASMI Flag.</b> If high, records that an ASMI was generated on a Halt special cycle. Write 1 to clear; writing 0 has no effect. HLT_ASMI_EN (bit 0) must be high to generate an ASMI and set flag.
31:16	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
15	PM1_CNT_SSMI_EN	<b>Power Management 1 Control Register SSMI Enable.</b> Write 1 to enable PM1_CNT_SSMI_FLAG (bit 47) and to allow writes to PM1_CNT (ACPI I/O Offset 08h) to generate an SSMI.
14	PM2_CNT_SSMI_EN	<b>Power Management 2 Control Register SSMI Enable.</b> Write 1 to enable PM2_CNT_SSMI_FLAG (bit 46) and to allow writes to PM2_CNT (ACPI I/O Offset 0Ch) to generate an SSMI.
13	KEL_A20_ASMI_EN	<b>KEL Gate A20 ASMI Enable.</b> Write 1 to enable KEL_A20_ASMI_FLAG (bit 45) and to allow a keyboard gate A20 signal change in the KEL to generate an ASMI.
12	DMA_SSMI_EN	<b>DMA SSMI Enable.</b> Write 1 to enable DMA_SSMI_FLAG (bit 44) and to allow 8237s during DMA to generate an SSMI.
11	LPC_SSMI_EN	<b>LPC SSMI Enable.</b> Write 1 to enable LPC_SSMI_FLAG (bit 43) and to allow the LPC to generate an SSMI.
10	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
9	UART2_SSMI_EN	<b>UART2 SSMI Enable.</b> Write 1 to enable UART2_SSMI_FLAG (bit 41) and to allow UART2 to generate an SSMI.
8	UART1_SSMI_EN	<b>UART1 SSMI Enable.</b> Write 1 to enable UART1_SSMI_FLAG (bit 40) and to allow UART1 to generate an SSMI.
7	PORTA_INIT_ASMI_EN	<b>Port A INIT ASMI Enable.</b> Write 1 to enable PORTA_INIT_ASMI_FLAG (bit 39) and to allow an INIT on Port A in the KEL to generate an ASMI.
6	PORTA_A20_ASMI_EN	<b>Port A A20 ASMI Enable.</b> Write 1 to enable PORTA_A20_ASMI_FLAG (bit 38) and to allow an A20 change on Port A in the KEL to generate an ASMI.
5	KEL_INIT_ASMI_EN	<b>KEL INIT ASMI Enable.</b> Write 1 to enable KEL_INIT_ASMI_FLAG (bit 37) and to allow a keyboard INIT sequence in the KEL to generate an ASMI.
4	PM_ASMI_EN	<b>Power Management ASMI Enable.</b> Write 1 to enable PM_ASMI_FLAG (bit 36) and to allow the Power Management Logic to generate an ASMI.
3	PIC_ASMI_EN	<b>PIC ASMI Enable.</b> Write 1 to enable PIC_ASMI_FLAG (bit 35) and to allow the Extended PIC Mapper to generate an ASMI.
2	KEL_EE_ASMI_EN	<b>KEL Emulation Event ASMI Enable.</b> Write 1 to enable KEL_EE_ASMI_FLAG (bit 34) and to allow the KEL to generate an ASMI.
1	SHTDWN_ASMI_EN	<b>Shutdown ASMI Enable.</b> Write 1 to enable SHTDWN_ASMI_FLAG (bit 33) and to allow a Shutdown special cycle to generate an ASMI.
0	HLT_ASMI_EN	<b>Halt ASMI Enable.</b> Write 1 to enable HLT_ASMI_FLAG (bit 32) and to allow a Halt special cycle to generate an ASMI.



**6.6.1.4 GLD Error MSR (DIVIL\_GLD\_MSR\_ERROR)**

MSR Address 51400003h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 1. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.4 "MSR Address 3: Error Control" on page 78 for further on ERR generation details.)

**DIVIL\_GLD\_MSR\_ERROR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32									
RSVD									UART1UART2_ERR_FLAG	MEM_LBAR_DECODE_ERR_FLAG	IO_LBAR_DECODE_ERR_FLAG	RSVD									SHTDWN_ERR_FLAG	NAND_DIST_ERR_FLAG	RSVD	DMA_DMA_ERR_FLAG	LPC_DMA_ERR_FLAG	RSVD	UART2_DMA_ERR_FLAG	UART1_DMA_ERR_FLAG	RSVD						LPC_MAST_ERR_FLAG	LPC_SLV_ERR_FLAG	MAST_RESP_EXCEP_FLAG	REPEAT_SSMI_ERR_FLAG	DECODE_ERR_FLAG	LB_ADAP_BAD_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
RSVD																LPC_EXCP_EN	SHTDWN_ERR_EN	NAND_DIST_ERR_EN	RSVD	DMA_DMA_ERR_EN	LPC_DMA_ERR_EN	RSVD	UART2_DMA_ERR_EN	UART1_DMA_ERR_EN	RSVD						LPC_MAST_ERR_EN	LPC_SLV_ERR_EN	MAST_RESP_EXCEP_EN	REPEAT_SSMI_ERR_EN	DECODE_ERR_EN	LB_ADAP_BAD_EN				

**DIVIL\_GLD\_MSR\_ERROR Bit Descriptions**

Bit	Name	Description
63:55	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
54	UART1UART2_ERR_FLAG	<b>UART1 and UART2 Error Flag.</b> If high, records that an ERR was generated due to a collision between the two UARTs. UART1 and UART2 are set to the same I/O address base. No chip selects are asserted and DECODE_ERR_FLAG (bit 33) is asserted. DECODE_ERR_EN (bit 1) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
53	MEM_LBAR_DECODE_ERR_FLAG	<b>Memory LBAR Decode Error Flag.</b> If high, records that an ERR was generated due to a collision between one memory LBAR and another memory LBAR hit. In this case, NO chip select is generated and DECODE_ERR_FLAG (bit 33) is asserted. DECODE_ERR_EN (bit 1) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
52	IO_LBAR_DECODE_ERR_FLAG	<b>I/O LBAR Decode Error Flag.</b> If high, records that an ERR was generated due to a collision between one I/O LBAR and another I/O LBAR hit. In this case, NO chip select is generated and DECODE_ERR_FLAG (bit 33) is asserted. DECODE_ERR_EN (bit 1) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
51:48	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
47	SHTDWN_ERR_FLAG	<b>Shutdown Error Flag.</b> If high, records that an ERR was generated due to a Shutdown cycle occurrence. SHTDWN_ERR_EN (bit 15) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.

## DIVIL\_GLD\_MSR\_ERROR Bit Descriptions (Continued)

Bit	Name	Description
46	NAND_DIST_ERR_FLAG	<b>NAND Distract Error Flag.</b> If high, records that an ERR was generated due to a NAND distract error. NAND_DIST_ERR_EN (bit 14) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
45	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
44	DMA_DMA_ERR_FLAG	<b>8237 DMA Error Flag.</b> If high, records that an ERR was generated due to an access on the 8237s during DMA. DMA_DMA_ERR_EN (bit 12) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
43	LPC_DMA_ERR_FLAG	<b>LPC DMA Error Flag.</b> If high, records that an ERR was generated due to an LPC access during DMA. LPC_DMA_ERR_EN (bit 11) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
42	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
41	UART2_DMA_ERR_FLAG	<b>UART2 DMA Error Flag.</b> If high, records that an ERR was generated due to a UART2 access during DMA. UART2_DMA_ERR_EN (bit 9) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
40	UART1_DMA_ERR_FLAG	<b>UART1 DMA Error Flag.</b> If high, records that an ERR was generated due to a UART1 access during DMA. UART1_DMA_ERR_EN (bit 8) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
39:38	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
37	LPC_MAST_ERR_FLAG	<b>LPC Master Error Flag.</b> If high, records that an ERR was generated on the LPC due to a Master <=> GeodeLink Adapter transaction. LPC_MAST_ERR_EN (bit 5) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
36	LPC_SLV_ERR_FLAG	<b>LPC Slave Error Flag.</b> If high, records that an ERR was generated on the LPC due to a Slave <=> GeodeLink Adapter transaction. LPC_SLV_ERR_EN (bit 4) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
35	MAST_RESP_EXCEP_FLAG	<b>Master Response Exception Flag.</b> If high, records that an ERR was generated due to the GeodeLink Adapter detecting the EXCEP bit set in a local bus master response packet. MAST_RESP_EXCP_EN (bit 3) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
34	REPEAT_SSMI_ERR_FLAG	<b>Repeat SSMI Error Flag.</b> If high, records that an ERR was generated due to a second SSMI occurring on an address before the first was cleared. REPEAT_SSMI_ERR_EN (bit 2) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
33	DECODE_ERR_FLAG	<b>Decode Error Flag.</b> If high, records that an ERR was generated during the address decode cycle due to one or more devices being decoded to the same address. Bits [54:52] record further information about this type of error. DECODE_ERR_EN (bit 1) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
32	LB_ADAP_BAD_FLAG	<b>LBUS Adapter Bad Flag.</b> If high, records that an ERR was generated due to the GeodeLink Adapter detected an error at the GeodeLink interface (e.g., packet type not supported). LB_ADAP_BAD_EN (bit 0) must be high to generate ERR and set flag. Write 1 to clear; writing 0 has no effect.
31:17	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
16	LPC_EXCP_EN	<b>LPC Exception Enable.</b> Write 1 to enable EXCEP bit in response packet for LPC address errors.
15	SHTDWN_ERR_EN	<b>Shutdown Error Enable.</b> Write 1 to enable SHTDWN_ERR_FLAG (bit 47) and to allow a Shutdown cycle to generate an ERR.
14	NAND_DIST_ERR_EN	<b>NAND Distract Error Enable.</b> Write 1 to enable NAND_DIST_ERR_FLAG (bit 46) and to allow a NAND distract error to generate an ERR.
13	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.

**DIVIL\_GLD\_MSR\_ERROR Bit Descriptions (Continued)**

Bit	Name	Description
12	DMA_DMA_ERR_EN	<b>8237 DMA Error Enable.</b> Write 1 to enable DMA_DMA_ERR_FLAG (bit 44) and to allow an access on the 8237s during DMA to generate an ERR.
11	LPC_DMA_ERR_EN	<b>LPC DMA Error Enable.</b> Write 1 to enable LPC_DMA_ERR_FLAG (bit 43) and to allow an LPC access during DMA an LPC access during DMA to generate an ERR.
10	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
9	UART2_DMA_ERR_EN	<b>UART2 DMA Error Enable.</b> Write 1 to enable UART2_DMA_ERR_FLAG (bit 41) and to allow UART2 accesses during DMA to generate an ERR.
8	UART1_DMA_ERR_EN	<b>UART1 DMA Error Enable.</b> Write 1 to enable UART1_DMA_ERR_FLAG (bit 40) and to allow UART1 accesses during DMA to generate an ERR.
7:6	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
5	LPC_MAST_ERR_EN	<b>LPC Master Error Enable.</b> Write 1 to enable LPC_MAST_ERR_FLAG (bit 37) and to allow Master <=> GeodeLink Adapter transactions to generate an ERR.
4	LPC_SLV_ERR_EN	<b>LPC Slave Error Enable.</b> Write 1 to enable LPC_SLV_ERR_FLAG (bit 36) and to allow Slave <=> GeodeLink Adapter transactions to generate an ERR.
3	MAST_RESP_EXCEP_EN	<b>Master Response Exception Enable.</b> Write 1 to enable MAST_RESP_EXCEP_FLAG (bit 35) and to allow when the GeodeLink Adapter detects the EXCEP bit set in a local bus master response packet to generate an ERR.
2	REPEAT_SSMI_ERR_EN	<b>Repeat SSMI Error Enable.</b> Write 1 to enable REPEAT_SSMI_ERR_FLAG (bit 34) and to allow when a second SSMI occurs on an address before the first was cleared to generate an ERR.
1	DECODE_ERR_EN	<b>Decode Error Enable.</b> Write 1 to enable FLAG bits [54:52] and to allow when one or more devices are decoded to the same address during the address decode cycle to generate an ERR.
0	LB_ADAP_BAD_EN	<b>LBus Adapter Bad Enable.</b> Write 1 to enable LB_ADAP_BAD_FLAG (bit 32) and to allow when the GeodeLink Adapter detects an error at the GeodeLink interface to generate an ERR.

**6.6.1.5 GLD Power Management MSR (DIVIL\_GLD\_MSR\_PM)**

MSR Address     51400004h  
 Type            R/W  
 Reset Value    00000000\_00000000h

**DIVIL\_GLD\_MSR\_PM Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMODE15	PMODE14	RSVD						PMODE11	PMODE10	PMODE9	PMODE8	PMODE7	PMODE6	PMODE5	PMODE4	RSVD		PMODE2	PMODE1	PMODE0											

## DIVIL\_GLD\_MSR\_PM Bit Descriptions

Bit	Name	Description
63:48	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
47:46	RSVD	<b>Reserved.</b> Do not write. Reads return 0.
45:44	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
43:36	RSVD	<b>Reserved.</b> Do not write. Reads return 0.
35	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
34:32	RSVD	<b>Reserved.</b> Do not write. Reads return 0.
31:30	PMODE15	<b>Power Mode for GPIO Standby Power Domain 32 kHz Clock Domain.</b> 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
29:28	PMODE14	<b>Power Mode for GPIO Working Power Domain 32 kHz Clock Domain.</b> See bits [31:30] for decode.
27:24	RSVD	<b>Reserved.</b> Reads will return the value written.
23:22	PMODE11	<b>Power Mode for MFGPT Standby Power Domain 32 kHz Clock Domain.</b> See bits [31:30] for decode.
21:20	PMODE10	<b>Power Mode for MFGPT Working Power Domain 32 kHz Clock Domain.</b> See bits [31:30] for decode.
19:18	PMODE9	<b>Power Mode for MFGPT Working power domain 14 MHz Clock Domain.</b> See bits [31:30] for decode.
17:16	PMODE8	<b>Power Mode for LPC.</b> See bits [31:30] for decode.
15:14	PMODE7	<b>Power Mode for UART2.</b> See bits [31:30] for decode.
13:12	PMODE6	<b>Power Mode for UART1.</b> See bits [31:30] for decode.
11:10	PMODE5	<b>Power Mode for System Management Bus Controller.</b> See bits [31:30] for decode.
9:8	PMODE4	<b>Power Mode for DMA (8237).</b> See bits [31:30] for decode.
7:6	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
5:4	PMODE2	<b>Power Mode for PIT (8254).</b> See bits [31:30] for decode.
3:2	PMODE1	<b>Power Mode for GeodeLink Adapter Local Bus Interface and Local Bus Clock.</b> See bits [31:30] for decode.
1:0	PMODE0	<b>Power Mode for GeodeLink Adapter GeodeLink Interface.</b> See bits [31:30] for decode.

## 6.6.1.6 GLD Diagnostic MSR (DIVIL\_GLD\_MSR\_DIAG)

MSR Address 51400005h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.6.2 DIVIL Specific MSRs

Refer to Section 5.6 "Diverse Integration Logic" on page 104 for an explanation and block diagram of the address comparison mechanism of the base address and the address lines.

Note that the I/O space 04FFh-0000h is off limits to I/O LBARs.

### 6.6.2.1 Local BAR - IRQ Mapper (DIVIL\_LBAR\_IRQ)

MSR Address 51400008h  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details.

The IRQ Mapper takes 32 bytes of I/O space. Use of this LBAR is optional. The IRQ Mapper is always available via MSR space.

**DIVIL\_LBAR\_IRQ Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD										LBAR_EN		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3		2	1
RSVD																BASE_ADDR										RSVD					

**DIVIL\_MSR\_LBAR\_IRQ Bit Descriptions**

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105.
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable address detection by this LBAR. 1: Enable address detection by this LBAR.
31:20	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:5	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105.
4:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.2 Local BAR - KEL from USB Host Controller 1 (DIVIL\_LBAR\_KEL1)**

MSR Address 51400009h  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details.

The KEL registers take 4 KB of memory space. However, only offsets 100h, 104h, 108h, and 10Ch contain registers. All other writes are "don't care" and reads return 0.

This is one of two KEL LBARs. Each LBAR "hits" to the same KEL. This allows USB Host Controllers at different addresses to be used, if desired. Both LBARs do NOT have to be used.

**DIVIL\_LBAR\_KEL1 Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
MEM_MASK														RSVD													LBAR_EN				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5		4	3	2	1
BASE_ADDR														RSVD																	

**DIVIL\_LBAR\_KEL1 Bit Descriptions**

Bit	Name	Description
63:44	MEM_MASK	<b>Memory Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:12	BASE_ADDR	<b>Base Address in Memory Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
11:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.3 Local BAR - KEL from USB Host Controller 2 (DIVIL\_LBAR\_KEL2)**

MSR Address 5140000Ah  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details.

The KEL registers take 4 KB of memory space. However, only offsets 100h, 104h, 108h, and 10Ch contain registers. All other writes are "don't care" and reads return 0.

This is one of two KEL LBARs. Each LBAR "hits" to the same KEL. This allows USB Host Controllers at different addresses to be used if desired. Both LBARs do NOT have to be used.

**DIVIL\_LBAR\_KEL2 Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
MEM_MASK														RSVD													LBAR_EN				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5		4	3	2	1
BASE_ADDR														RSVD																	

## DIVIL\_LBAR\_KEL2 Bit Descriptions

Bit	Name	Description
63:44	MEM_MASK	<b>Memory Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:12	BASE_ADDR	<b>Base Address in Memory Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
11:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

## 6.6.2.4 Local BAR - SMB (DIVIL\_LBAR\_SMB)

MSR Address 5140000Bh  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details. The SMB Controller takes 8 bytes of I/O space.

## DIVIL\_LBAR\_SMB Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD										LBAR_EN		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3		2	1
RSVD																BASE_ADDR										RSVD					

## DIVIL\_LBAR\_SMB

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:3	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
2:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.5 Local BAR - GPIO and ICFs (DIVIL\_LBAR\_GPIO)**

MSR Address 5140000Ch  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details. The GPIOs and ICFs take 256 bytes of I/O space.

**DIVIL\_LBAR\_GPIO Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD										LBAR_EN		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3		2	1
RSVD																BASE_ADDR						RSVD									

**DIVIL\_LBAR\_GPIO Bit Description**

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:8	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
7:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.



**6.6.2.6 Local BAR - MFGPTs (DIVIL\_LBAR\_MFGPT)**

MSR Address 5140000Dh  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details. The MFGPTs take 64 bytes of I/O space.

**DIVIL\_LBAR\_MFGPT Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD										LBAR_EN		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BASE_ADDR						RSVD									

**DIVIL\_LBAR\_MFGPT Bit Descriptions**

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:6	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
5	RSVD	<b>Reserved.</b> Reads return value written. Default value is 0. Note that this bit is reserved and performs no function.
4:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.7 Local BAR - ACPI (DIVIL\_LBAR ACPI)**

MSR Address 5140000Eh  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details.

The ACPI registers take 32 bytes of I/O space. Offsets are as follows:

00h PM1\_STS  
 02h PM1\_EN  
 08h PM1\_CNT  
 0Ch PM2\_CNT  
 10h PM\_TMR  
 14h Reserved  
 18h GPE0\_STS  
 1Ch GPE0\_EN

**DIVIL\_LBAR ACPI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD										LBAR_EN		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3		2	1
RSVD																BASE_ADDR										RSVD					

**DIVIL\_LBAR ACPI Bit Descriptions**

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:5	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
4:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.8 Local BAR - Power Management Support (DIVIL\_LBAR\_PMS)**

MSR Address 5140000Fh  
 Type R/W  
 Reset Value 00000000\_00000000h

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details. The Power Management Support registers take 128 bytes of I/O space.

**DIVIL\_LBAR\_PMS Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK			RSVD											LBAR_EN	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2		1
RSVD																BASE_ADDR						RSVD									

**DIVIL\_LBAR\_PMS Bit Descriptions**

Bit	Name	Description
63:49	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
48	RSVD	<b>Reserved.</b> Always write 0.
47:44	IO_MASK	<b>I/O Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:33	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
16	RSVD	<b>Reserved.</b> Always write 0.
15:7	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
6:0	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.

**6.6.2.9 Local BAR - Flash Chip Select (DIVIL\_LBAR\_FLSH[x])**

See Section 5.6.1 "LBARs and Comparators" on page 105 for operational details.

The NAND Flash control registers take 16 bytes of I/O space. NOR Flash maps into some multiple of 4k bytes.

There are two forms of this LBAR depending on the space, memory or I/O, Flash Device 0 is mapped into. Space is determined by bit 34 of the LBAR.

**Local BAR - Flash Chip Select 0 (DIVIL\_LBAR\_FLSH0)**

MSR Address 51400010h  
 Type R/W  
 Reset Value 00000000\_00000000h

Uses FLASH\_CS0# and FLASH\_CE0#.

**Local BAR - Flash Chip Select 1 (DIVIL\_LBAR\_FLSH1)**

MSR Address 51400011h  
 Type R/W  
 Reset Value 00000000\_00000000h

Uses FLASH\_CS1# and FLASH\_CE1#.

**Local BAR - Flash Chip Select 2 (DIVIL\_LBAR\_FLSH2)**

MSR Address 51400012h  
 Type R/W  
 Reset Value 00000000\_00000000h

Uses FLASH\_CS2# and FLASH\_CE2#.

**Local BAR - Flash Chip Select 3 (DIVIL\_LBAR\_FLSH3)**

MSR Address 51400013h  
 Type R/W  
 Reset Value 00000000\_00000000h

Uses FLASH\_CS3# and FLASH\_CE3#.

**DIVIL\_LBAR\_FLSH[x] Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																IO_MASK											RSVD	0	NOR_NAND	LBAR_EN	
MEM_MASK																RSVD											RSVD	1	NOR_NAND	LBAR_EN	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																BASE_ADDR											RSVD				
BASE_ADDR																RSVD															

**DIVIL\_LBAR\_FLSH[x] Bit Descriptions**

Bit	Name	Description
<b>If bit 34 = 0; I/O Mapped</b>		
63:49	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0
48	RSVD	<b>Reserved.</b> Always write 0.
47:36	IO_MASK	<b>I/O Address Mask Value.</b> For standard NAND Flash, bits [48:36] should be set to all 1s. Add 0s from the LSBs as needed for OEM specific devices that take more than 16 bytes. See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
35	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.
34	MEM_IO	<b>Memory or I/O Mapped.</b> 0: LBAR is I/O mapped. 1: LBAR is memory mapped.
33	NOR_NAND	<b>NOR or NAND.</b> 0: Use NOR chip select (FLASH_CS[x]#). 1: Use NAND chip select (FLASH_CE[x]#).

**DIVIL\_LBAR\_FLSH[x] Bit Descriptions (Continued)**

Bit	Name	Description
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:17	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.
16	RSVD	<b>Reserved.</b> Always write 0.
15:4	BASE_ADDR	<b>Base Address in I/O Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
3:0	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.
<b>If bit 34 = 1; Memory Mapped</b>		
63:44	MEM_MASK	<b>Memory Address Mask Value.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
43:35	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.
34	MEM_IO	<b>Memory or I/O Mapped.</b> 0: LBAR is I/O mapped. 1: LBAR is memory mapped.
33	NOR_NAND	<b>NOR or NAND.</b> 0: Use NOR chip select (FLASH_CS[x]#). 1: Use NAND chip select (FLASH_CE[x]#).
32	LBAR_EN	<b>LBAR Enable.</b> 0: Disable LBAR. 1: Enable LBAR.
31:12	BASE_ADDR	<b>Base Address in Memory Space.</b> See discussion in Section 5.6.1 "LBARs and Comparators" on page 105
11:0	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.

**6.6.2.10 Legacy I/O Space Controls (DIVIL\_LEG\_IO)**

MSR Address 51400014h  
 Type R/W  
 Reset Value 04000003h

**DIVIL\_LEG\_IO Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_SHUT_EN	RESET_BAD_EN	RSVD	SPEC_CYC_MD	0x000E_XXXX	0x000F_XXXX	LPC_DISABLE_MEM	LPC_DISABLE_IO	RSVD		UART2_ENABLE[2:0]		RSVD		UART1_ENABLE[2:0]		RSVD													RTC_ENABLE1	RTC_ENABLE0	

**DIVIL\_LEG\_IO Bit Description**

Bit	Name	Description
31	RESET_SHUT_EN	<b>Shutdown Reset Enable</b> If set, this bit will enable the issuance of the RESET_OUT# signal upon the detection of a PCI Shutdown cycle from the Geode GX processor (or any other PCI master). The reason for the reset is recorded in the PM_SSC register (PMS I/O Offset 54h[9], see Section 6.18.3.19 on page 517 for bit details). 0: Do not issue RESET_OUT# upon detection of Shutdown cycle. 1: Issue RESET_OUT# upon detection of Shutdown cycle.
30	RESET_BAD_EN	<b>Bad Transaction Reset Enable</b> If set, this bit will enable a system wide reset via the RESET_OUT# signal, if the GeodeLink Adapter detects a 'bad' GeodeLink transaction. The reason for the reset is recorded in the PM_SSC register (PMS I/O Offset 54h[12], see Section 6.18.3.19 on page 517 for bit details). 0: Do not issue RESET_OUT# upon detection of a "bad" transaction. 1: Issue RESET_OUT# upon detection of a "bad" transaction
29	RSVD	<b>Reserved:</b> This bit should always be written to 0.
28	SPEC_CYC_MD	<b>Special Cycle Mode.</b> Allows selection of how the DIVIL decodes Local bus address for GeodeLink special cycles. (Defaults 0.) 0: Decode is per the PCI spec: 00h: Shutdown. 01h: Halt. All other values ignored. 1: Decode is per the x86 standard: 00h: Shutdown. 02h: Halt. All other values ignored.
27	0X000E_XXXX	<b>000Exxxxh Remap.</b> If high, memory addresses in the range of 000Exxxxh are re-mapped to FFFExxxxh. Applies to addresses except the LBAR comparators and other address decode functions. (Defaults 0.)
26	0X000F_XXXX	<b>000Fxxxxh Remap.</b> If high, memory addresses in the range of 000Fxxxxh are re-mapped to FFFFxxxxh. Applies to addresses except the LBAR comparators and other address decode functions. (Defaults 1.)
25	LPC_DISABLE_MEM	<b>LPC Disable Memory.</b> If high, discard all memory writes which would otherwise go to the LPC by default. For reads, return all 1s. "Default" means any address not explicitly mapped into on-chip memory space or claimed by an LBAR hit.
24	LPC_DISABLE_IO	<b>LPC Disable I/O.</b> If high, discard all I/O writes which would otherwise go to the LPC by default. For reads, return all 1s. "Default" means any address not explicitly mapped into on-chip legacy I/O space or claimed by an LBAR hit.

**DIVIL\_LEG\_IO Bit Description (Continued)**

Bit	Name	Description
23	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0.
22:20	UART2_ENABLE [2:0]	<b>UART2 Enable.</b> 0xx: UART2 not enabled into DIVIL I/O space; use LPC. 100: UART2 enabled into I/O base 02E8h (COM4). 101: UART2 enabled into I/O base 02F8h (COM3). 110: UART2 enabled into I/O base 03E8h (COM2). 111: UART2 enabled into I/O base 03F8h (COM1).  If UART1 and UART2 are set to the same I/O base, a decode error is generated on access.
19	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0
18:16	UART1_ENABLE [2:0]	<b>UART1 Enable.</b> 0xx: UART1 not enabled into DIVIL I/O space; use LPC. 100: UART1 enabled into I/O base 02E8h (COM4). 101: UART1 enabled into I/O base 02F8h (COM3). 110: UART1 enabled into I/O base 03E8h (COM2). 111: UART1 enabled into I/O base 03F8h (COM1).  If UART1 and UART2 are set to the same I/O base, a decode error is generated on access.
15:2	RSVD	<b>Reserved.</b> Reads return value written. Defaults to 0
1	RTC_ENABLE1	<b>Real-Time Clock Map 1.</b> Routes I/O port locations 072h and 073h to the internal RTC high RAM or LPC.  0: RTC high RAM routed to LPC bus. 1: RTC high RAM routed to internal RTC. (Default)
0	RTC_ENABLE0	<b>Real-Time Clock Map 0.</b> Routes I/O port locations 070h and 071h internal RTC or LPC. Writes to port 070h (Index) are always routed internal. The MSB is used to establish the NMI enable state.  0: RTC routed to LPC bus. 1: RTC routed to internal RTC. (Default)

**6.6.2.11 Ball Options Control (DIVIL\_BALL\_OPTS)**

MSR Address 51400015h  
 Type R/W  
 Reset Value 00000x7xh

**DIVIL\_BALL\_OPTS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											SEC_BOOT_LOC	BOOT_OP_LATCHED		RSVD	PIN_OPT_LALL	PIN_OPT_LIRQ	PIN_OPT_LDRQ	PRI_BOOT_LOC[1:0]	RSVD	PIN_OPT_IDE											

## DIVIL BALL\_OPTS Bit Descriptions

Bit	Name	Description
31:12	RSVD	<b>Reserved.</b> Reads always return 0. Writes have no effect; by convention, always write 0.
11:10	SEC_BOOT_LOC	<b>Secondary Boot Location.</b> Determines which chip select asserts for addresses in the range F00F0000h to F00F3FFFh. Defaults to the same value as boot option: 00: LPC ROM. 01: Reserved. 10: Flash. 11: FirmWare Hub.
9:8	BOOT_OP_LATCHED (RO)	<b>Latched Value of Boot Option (Read Only).</b> For values, see Table 3-5 "Boot Options Selection" on page 34.
7	RSVD	<b>Reserved.</b> Reads return value written. By convention, always write 0. Defaults low.
6	PIN_OPT_LALL	<b>All LPC Pin Option Selection.</b> 0: All LPC pins become GPIOs including LPC_DRQ# and LPC_SERIRQ. Ball H3 functions as GPIO22 Ball H2 functions as GPIO16 Ball J2 functions as GPIO17 Ball J1 functions as GPIO18 Ball K1 functions as GPIO19 Ball G1 functions as GPIO20 Ball G2 functions as GPIO21  1: All LPC pins are controlled by the LPC controller except LPC_DRQ# and LPC_SERIRQ use are determined by bits [5:4]. <b>(Default)</b> Ball H3 functions as LPC_FRAME# Ball H2 functions as LPC_AD0 Ball J2 functions as LPC_AD1 Ball J1 functions as LPC_AD2 Ball K1 functions as LPC_AD3  When this bit is low, there is an implied high for the LPC_DISABLE_MEM and LPC_DISABLE_IO bits in DIVIL_LEG_IO (DIVIL MSR 51400014h[25:24]).
5	PIN_OPT_LIRQ	<b>LPC_SERIRQ or GPIO21 Pin Option Selection.</b> 0: Ball G2 is GPIO21. 1: Ball G2 functions as LPC_SERIRQ. <b>(Default)</b>
4	PIN_OPT_LDRQ	<b>LPC_DRQ# or GPIO20 Pin Option Selection.</b> 0: Ball G1 is GPIO20. 1: Ball G2 functions as LPC_DRQ#. <b>(Default)</b>
3:2	PRI_BOOT_LOC [1:0]	<b>Primary Boot Location.</b> Determines which chip select asserts for addresses at or above F0000000h, except those in the range specified by SEC_BOOT_LOC (bits [11:10]). Defaults to the same value as boot option. 00: LPC ROM. 01: Reserved. 10: Flash. 11: FirmWare Hub.
1	RSVD	<b>Reserved.</b> Reads return value written. By convention, always write 0. Defaults low.
0	PIN_OPT_IDE	<b>IDE or Flash Controller Pin Function Selection.</b> 0: All IDE pins associated with Flash Controller. Default if BOS[1:0] = 10. 1: All IDE pins associated with IDE Controller. Default if BOS[1:0] = 00 or 11.  IDE_IRQ0 is multiplexed with GPIO2; therefore, this bit has no affect with regards to programming IDE_IRQ0. See Table 3-5 "Boot Options Selection" on page 34 for BOS[1:0] programming values.



**6.6.2.12 Soft IRQ (DIVIL\_SOFT\_IRQ)**

MSR Address 51400016h  
 Type R/W  
 Reset Value 00000000h

**DIVIL\_SOFT\_IRQ Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											SOFT_IRQ				

**DIVIL\_SOFT\_IRQ Bit Descriptions**

Bit	Name	Description
31:1	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
0	SOFT_IRQ	<b>Soft IRQ.</b> This bit can be written high or low, and is connected to the soft IRQ input of the IRQ Mapper. Hence, writing high causes an interrupt while writing low clears it. Reads return the value written.

**6.6.2.13 Soft Reset (DIVIL\_SOFT\_RESET)**

MSR Address 51400017h  
 Type R/W  
 Reset Value 00000000h

**DIVIL\_SOFT\_RESET Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											SOFT_RESET				

**DIVIL\_SOFT\_RESET Bit Descriptions**

Bit	Name	Description
31:1	RSVD	<b>Reserved.</b> Reads return 0. Writes have no effect.
0	SOFT_RESET	<b>Soft Reset.</b> A bit that when written to a 1 causes the system to hard reset. Reads return 0.

### 6.6.2.14 Access Control DMA Request (DIVIL\_AC\_DMA)

MSR Address 5140001Eh  
 Type R/W  
 Reset Value 00000000h

The controls below only affect memory and I/O accesses to the target slaves. MSR accesses are not affected. However, MSR writes during DMA may have unintended side effects.

Note that when in demand or block mode, the UART reads and writes are disallowed; no corresponding mechanism exists to allow UART controller reads or writes during UART activity. If attempted, CPU writes have no effect and the CPU reads return all 1s.

The enables default to 0. If 0, reads or writes to the indicated device are blocked during activity. This may cause an SSMI or ERROR if enabled by the associated MSR. Thus, writes are discarded and reads return all Fs. If an enable is 0, a chip select for the indicated device is not asserted. If an enable is 1, the indicated device is available for access during activity.

**DIVIL\_AC\_DMA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD														AC_DMA_W	AC_DMA_R	RSVD														AC_DMA_LPC_IW	AC_DMA_LPC_IR	AC_DMA_LPC_MW	AC_DMA_LPC_MR

**DIVIL\_AC\_DMA Bit Descriptions**

Bit	Name	Description
31:18	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
17	AC_DMA_W	<b>Allow DMA Writes during DMA Activity.</b> If set, this bit allows writes to the DMA controller during DMA activity (data transfers). This mechanism may be used, among other things, to abort a hung DMA transfer. If clear, DMA controller writes are locked out during DMA activity.
16	AC_DMA_R	<b>Allow DMA Reads during DMA Activity.</b> If set, this bit allows reads from the DMA controller during DMA activity (data transfers). If clear, DMA controller reads are locked out during DMA activity.
15:4	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
3	AC_DMA_LPC_IW	<b>LPC I/O Writes during LPC DMA</b> If set, this bit allows I/O writes to the LPC bus during LPC DMA transfer. If clear, I/O writes are locked out during LPC DMA transfers.
2	AC_DMA_LPC_IR	<b>LPC I/O Reads during LPC DMA.</b> If set, this bit allows I/O reads to the LPC bus during LPC DMA transfer. If clear, I/O reads are locked out during LPC DMA transfers.
1	AC_DMA_LPC_MW	<b>LPC Memory Writes during LPC DMA.</b> If set, this bit allows memory writes to the LPC bus during LPC DMA transfer. If clear, memory writes are locked out during LPC DMA transfers.
0	AC_DMA_LPC_MR	<b>LPC Memory Reads during LPC DMA.</b> If set, this bit allows memory reads to the LPC bus during LPC DMA transfer. If clear, memory reads are locked out during LPC DMA transfers.

## 6.7 Floppy Port Register Descriptions

The registers for the Floppy Port are divided into two sets:

- Standard GeodeLink Device (GLD) MSR's (Shared with DIVIL, see Section 6.6.1 "Standard GeodeLink™ Device (GLD) MSR's" on page 317.)
- Floppy Port Specific MSR's

The MSR's are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the

perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSR's are 64 bits, however, the Floppy Port Specific MSR's are called out as 8 bits. The Floppy Port treats writes to the upper 56 bits (i.e., bits [63:8]) of the 8-bit MSR's as don't cares and always returns 0 on these bits. Table 6-19 summarizes the Floppy Port Specific MSR's.

The reference column in the tables point to the page where the register maps and bit descriptions are listed.

**Table 6-19. Floppy Port Specific MSR's Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400030h	RO	Floppy Port 3F2h Shadow (FLPY_3F2_SHDW)	xxh	Page 339
51400031h	RO	Floppy Port 3F7h Shadow (FLPY_3F7_SHDW)	xxh	Page 339
51400032h	RO	Floppy Port 372h Shadow (FLPY_372_SHDW)	xxh	Page 340
51400033h	RO	Floppy Port 377h Shadow (FLPY_377_SHDW)	xxh	Page 340

### 6.7.1 Floppy Port Specific MSR's

#### 6.7.1.1 Floppy Port 3F2h Shadow (FLPY\_3F2\_SHDW)

MSR Address 51400030h  
 Type RO  
 Reset Value xxh

**FLPY\_3F2\_SHDW Register Map**

7	6	5	4	3	2	1	0
FLPY_PORT_3F2_VAL							

**FLPY\_3F2\_SHDW Bit Descriptions**

Bit	Name	Description
7:0	FLPY_PORT_3F2_VAL	<b>Floppy Port Shadow Register Value Last Written to I/O Port 3F2h.</b> Required for support of FDC power ON/OFF and Zero Volt Suspend/Resume coherency.  This register is a copy of an I/O register that cannot safely be directly read. Value in register is not deterministic of when the register is being read. It is provided here to assist in a Save-to-Disk operation.

#### 6.7.1.2 Floppy Port 3F7h Shadow (FLPY\_3F7\_SHDW)

MSR Address 51400031h  
 Type RO  
 Reset Value xxh

**FLPY\_3F7\_SHDW Register Map**

7	6	5	4	3	2	1	0
FLPY_PORT_3F7_VAL							

## FLPY\_3F7\_SHDW Bit Descriptions

Bit	Name	Description
7:0	FLPY_PORT_3F7_VAL	<b>Floppy Port Shadow Register Value Last Written to I/O Port 3F7h.</b> Required for support of FDC power ON/OFF and Zero Volt Suspend/Resume coherency. This register is a copy of an I/O register that cannot safely be directly read. Value in register is not deterministic of when the register is being read. It is provided here to assist in a Save-to-Disk operation.

## 6.7.1.3 Floppy Port 372h Shadow (FLPY\_372\_SHDW)

MSR Address      51400032h  
Type              RO  
Reset Value      xxh

## FLPY\_372\_SHDW Register Map

7	6	5	4	3	2	1	0
FLPY_PORT_372_VAL							

## FLPY\_372\_SHDW Bit Descriptions

Bit	Name	Description
7:0	FLPY_PORT_372_VAL	<b>Floppy Port Shadow Register Value Last Written to I/O Port 372h.</b> Required for support of FDC power ON/OFF and Zero Volt Suspend/Resume coherency. This register is a copy of an I/O register that cannot safely be directly read. Value in register is not deterministic of when the register is being read. It is provided here to assist in a Save-to-Disk operation.

## 6.7.1.4 Floppy Port 377h Shadow (FLPY\_377\_SHDW)

MSR Address      51400033h  
Type              RO  
Reset Value      xxh

## FLPY\_377\_SHDW Register Map

7	6	5	4	3	2	1	0
FLPY_PORT_377_VAL							

## FLPY\_377\_SHDW Bit Descriptions

Bit	Name	Description
7:0	FLPY_PORT_377_VAL	<b>Floppy Port Shadow Register Value Last Written to I/O Port 377h.</b> Required for support of FDC power ON/OFF and Zero Volt Suspend/Resume coherency. This register is a copy of an I/O register that cannot safely be directly read. Value in register is not deterministic of when the register is being read. It is provided here to assist in a Save-to-Disk operation.

## 6.8 Programmable Interval Timer Register Descriptions

The registers for the Programmable Interval Timer (PIT) are divided into three sets:

- Standard GeodeLink Device (GLD) MSR's (Shared with DIVIL, see Section 6.6.1 on page 317.)
- PIT Specific MSR's
- PIT Native Registers

The MSR's are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSR's are 64 bits, however, the PIT Specific MSR's are called out as 8 bits. The PIT treats writes to the upper 56 bits (i.e., bits [63:8]) of the 64-bit MSR's as don't cares and always returns 0 on these bits. The PIT Specific MSR's are summarized in Table 6-20.

The Native registers associated with the PIT are summarized in Table 6-21 and are accessed as I/O Addresses.

The reference column in the tables point to the page where the register maps and bit descriptions are listed.

**Table 6-20. PIT Specific MSR's Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400036h	RO	PIT Shadow (PIT_SHDW)	00h	Page 342
51400037h	R/W	PIT Count Enable (PIT_CNTRL)	03h	Page 342

**Table 6-21. PIT Native Registers Summary**

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
40h	W	8	PIT Timer 0 Counter - System (PIT_TMR0_CNTR_SYS)	00h	Page 343
	R	8	PIT Timer 0 Status - System (PIT_TMR0_STS_SYS)	00h	Page 344
41h	W	8	PIT Timer 1 Counter - Refresh (PIT_TMR1_CNTR_RFSH)	00h	Page 345
	R	8	PIT Timer 1 Status - Refresh (PIT_TMR1_STS_RFSH)	00h	Page 345
42h	W	8	PIT Timer 2 Counter - Speaker (PIT_TMR2_CNTR_SPKR)	00h	Page 346
	R	8	PIT Timer 2 Status - Speaker (PIT_TMR2_STS_SPKR)	00h	Page 346
43h	R/W	8	PIT Mode Control Word (PIT_MODECTL_WORD)	00h	Page 347
61h	R/W	8	Port B Control (PIT_PORTBCTL)	00h	Page 348

## 6.8.1 PIT Specific MSRs

### 6.8.1.1 PIT Shadow (PIT\_SHDW)

MSR Address 51400036h

Type RO

Reset Value 00h

**PIT\_SHDW Register Map**

7	6	5	4	3	2	1	0
PIT_SHDW							

**PIT\_SHDW Bit Descriptions**

Bit	Name	Description
7:0	PIT_SHDW (RO)	<p><b>PIT Shadow (Read Only).</b> This 8-bit port sequences through the following list of shadowed Programmable Interval Timer registers. At power on, a pointer starts at the first register in the list and consecutively reads to increment through it. A write to this register resets the read sequence to the first register. Each shadow register in the sequence contains the last data written to that location.</p> <p>The read sequence for this register is:</p> <ol style="list-style-type: none"> <li>1. Counter 0 LSB (least significant byte)</li> <li>2. Counter 0 MSB</li> <li>3. Counter 1 LSB</li> <li>4. Counter 1 MSB</li> <li>5. Counter 2 LSB</li> <li>6. Counter 2 MSB</li> <li>7. Counter 0 Command Word</li> <li>8. Counter 1 Command Word</li> <li>9. Counter 2 Command Word</li> </ol> <p><b>Note:</b> The LSB / MSB of the count is the Counter base value, not the current value. In the case of counter mode 3, the LSB of the count is the Counter base value - 1 (even count value). Bits [7:6] of the command words are not used.</p>

### 6.8.1.2 PIT Count Enable (PIT\_CNTRL)

MSR Address 51400037h

Type R/W

Reset Value 03h

**PIT\_CNTRL Register Map**

7	6	5	4	3	2	1	0
RSVD			PIT_CNTR_ ACC_DLY_EN	RSVD		PIT_CNTR1_ EN	PIT_CNTR0_ EN

**PIT\_CNTRL Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Read zero. Write "don't care".

**PIT\_CNTRL Bit Descriptions (Continued)**

Bit	Name	Description
4	PIT_CNTR_ACC_DLY_EN	<b>PIT Counter Access Delay Enable.</b> Used as an access delay enable for the read and write operations of the PIT counters. This bit introduces a 1 $\mu$ s delay between successive reads and/or writes of the PIT counters. This bit is intended to ensure that older, DOS-based programs that rely on the PIT timing access to be 1 $\mu$ s still function properly. 0: Disable access delay. 1: Enable access delay.
3:2	RSVD	<b>Reserved.</b> Read zero. Write "don't care".
1	PIT_CNTR1_EN	<b>PIT Counter 1 Enable.</b> 0: Sets GATE1 input low. 1: Sets GATE1 input high.
0	PIT_CNTR0_EN	<b>PIT Counter 0 Enable.</b> 0: Sets GATE0 input low. 1: Sets GATE0 input high.
<b>Note:</b> PIT_CNTR2_EN (PIT Counter 2 Enable) bit is located at I/O Address 61h[0] (see Section 6.8.2.8 "Port B Control (PIT_PORTBCTL)" on page 348).		

**6.8.2 PIT Native Registers**

**6.8.2.1 PIT Timer 0 Counter - System (PIT\_TMR0\_CNTR\_SYS)**

I/O Address      40h  
 Type              W  
 Reset Value      00h

**PIT\_TMR0\_CNTR\_SYS Register Map**

7	6	5	4	3	2	1	0
CNTR0							

**PIT\_TMR0\_CNTR\_SYS Bit Description**

Bit	Name	Description
7:0	CNTR0	<b>Counter 0 Value.</b> Provides the base counter value.

**6.8.2.2 PIT Timer 0 Status - System (PIT\_TMR0\_STS\_SYS)**

I/O Address 40h  
 Type R  
 Reset Value 00h

**PIT\_TMR0\_STS\_SYS Register Map**

7	6	5	4	3	2	1	0
I/O Address 43h[7:0] = 1101xx10 or 0010xxxx							
CNTR0_CUR_COUNT							
I/O Address 43h[7:0] = 11110xx10							
CNTR0_OUT	CNTR0_LOAD	CNTR0_RW		CNTR0_MODE		BCD	

**PIT\_TMR0\_STS\_SYS Bit Descriptions**

Bit	Name	Description
I/O Address 43h[7:0] = 1101xx10 or 0010xxxx		
7:0	CNTR0_CUR_COUNT	<b>Counter 0 Current Count.</b> Reports the current count value in Counter 0.
I/O Address 43h[7:0] = 11110xx10		
7	CNTR0_OUT	<b>Counter 0 Output.</b> Returns current state of counter output signal.
6	CNTR0_LOAD	<b>Counter 0 Loaded.</b> Last count written is loaded? 0: Yes. 1: No.
5:4	CNTR0_RW	<b>Counter 0 Read /Write Mode.</b> 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	CNTR0_MODE	<b>Counter 0 Current Mode.</b> 000: Interrupt on terminal count. 001: Programmable one-shot. 010, 110: Rate generator. 011, 111: Square wave generator. 100: Software triggered pulse generator. 101: Hardware triggered pulse generator.
0	BCD	<b>BCD Mode.</b> 0: Binary. 1: BCD (binary coded decimal).



**6.8.2.3 PIT Timer 1 Counter - Refresh (PIT\_TMR1\_CNTR\_RFSH)**

I/O Address 41h  
 Type W  
 Reset Value 00h

**PIT\_TMR1\_CNTR\_RFSH Register Map**

7	6	5	4	3	2	1	0
CNTR1							

**PIT\_TMR1\_CNTR\_RFSH Bit Description**

Bit	Name	Description
7:0	CNTR1	<b>Counter 1 Value.</b> Provides the base counter value.

**6.8.2.4 PIT Timer 1 Status - Refresh (PIT\_TMR1\_STS\_RFSH)**

I/O Address 41h  
 Type R  
 Reset Value 00h

**PIT\_TMR1\_STS\_RFSH Register Map**

7	6	5	4	3	2	1	0
I/O Address 43h[7:0] = 1101x1x0 or 0110xxxx							
CNTR1_CUR_COUNT							
I/O Address 43h[7:0] = 1110x1x0							
CNTR1_OUT	CNTR1_LOAD	CNTR1_RW		CNTR1_MODE		BCD	

**PIT\_TMR1\_STS\_RFSH Bit Descriptions**

Bit	Name	Description
I/O Address 43h[7:0] = 1101x1x0 or 0110xxxx		
7:0	CNTR1_CUR_COUNT	<b>Counter 1 Current Count.</b> Reports the current count value in Counter 1.
I/O Address 43h[7:0] = 1110x1x0		
7	CNTR1_OUT	<b>Counter 1 Output.</b> Returns current state of counter output signal.
6	CNTR1_LOAD	<b>Counter 1 Loaded.</b> Last count written is loaded? 0: Yes. 1: No.
5:4	CNTR1_RW	<b>Counter 1 Read /Write Mode.</b> 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.

## PIT\_TMR1\_STS\_RFSH Bit Descriptions (Continued)

Bit	Name	Description
3:1	CNTR1_MODE	<b>Counter 1 Current Mode.</b> 000: Interrupt on terminal count. 001: Programmable one-shot. 010, 110: Rate generator. 011, 111: Square wave generator. 100: Software triggered pulse generator. 101: Hardware triggered pulse generator.
0	BCD	<b>BCD Mode.</b> 0: Binary. 1: BCD (binary coded decimal).

## 6.8.2.5 PIT Timer 2 Counter - Speaker (PIT\_TMR2\_CNTR\_SPKR)

I/O Address 42h  
Type W  
Reset Value 00h

## PIT\_TMR2\_CNTR\_SPKR Register Map

7	6	5	4	3	2	1	0
CNTR2							

## PIT\_TMR2\_CNTR\_SPKR Bit Description

Bit	Name	Description
7:0	CNTR2	<b>Counter 2 Value.</b> Provides the base counter value.

## 6.8.2.6 PIT Timer 2 Status - Speaker (PIT\_TMR2\_STS\_SPKR)

I/O Address 42h  
Type R  
Reset Value 00h

## PIT\_TMR2\_STS\_SPKR Register Map

7	6	5	4	3	2	1	0
I/O Address 43h[7:0] = 11011xx0 or 1000xxxx							
CNTR2_CUR_COUNT							
I/O Address 43h[7:0] = 11101xx0							
CNTR2_OUT	CNTR2_LOAD	CNTR2_RW		CNTR2_MODE		BCD	

## PIT\_TMR2\_STS\_SPKR Bit Descriptions

Bit	Name	Description
I/O Address 43h[7:0] = 11011xx0 or 1000xxxx		
7:0	CNTR2_CUR_COUNT	<b>Counter 2 Current Count.</b> Reports the current count value in Counter 2.

**PIT\_TMR2\_STS\_SPKR Bit Descriptions (Continued)**

Bit	Name	Description
<b>I/O Address 43h[7:0] = 11101xx0</b>		
7	CNTR2_OUT	<b>Counter 2 Output.</b> Returns current state of counter output signal.
6	CNTR2_LOAD	<b>Counter 2 Loaded.</b> Last count written is loaded? 0: Yes. 1: No.
5:4	CNTR2_RW	<b>Counter 2 Read /Write Mode.</b> 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	CNTR2_MODE	<b>Counter 2 Current Mode.</b> 000: Interrupt on terminal count. 001: Programmable one-shot. 010, 110: Rate generator. 011, 111: Square wave generator. 100: Software triggered pulse generator. 101: Hardware triggered pulse generator.
0	BCD	<b>BCD Mode.</b> 0: Binary. 1: BCD (binary coded decimal).

**6.8.2.7 PIT Mode Control Word (PIT\_MODECTL\_WORD)**

I/O Address 43h  
Type R/W  
Reset Value 00h

**PIT\_MODECTL\_WORD Register Map**

7	6	5	4	3	2	1	0
CNTR_SEL		RW_MODE		CNTR_MODE			BCD

**PIT\_MODECTL\_WORD Bit Descriptions**

Bit	Name	Description
7:6	CNTR_SEL	<b>Counter Select.</b> 00: Counter 0. 01: Counter 1. 10: Counter 2. 11: Read-back Command (Note 1).
5:4	RW_MODE	<b>Current Read/Write Mode.</b> 00: Counter latch command (Note 2). 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.

## PIT\_MODECTL\_WORD Bit Descriptions (Continued)

Bit	Name	Description
3:1	CNTR_MODE	<b>Current Counter Mode.</b> 000: Interrupt on terminal count. 001: Programmable one-shot. 010, 110: Rate generator. 011, 111: Square wave generator. 100: Software triggered pulse generator. 101: Hardware triggered pulse generator.
0	BCD	<b>BCD Mode.</b> 0: Binary. 1: BCD (binary coded decimal).

Note 1. If bits [7:6] = 11: Register functions as Read Status Command  
Bit 5 = Latch Count, Bit 4 = Latch Status, Bit 3 = Select Counter 2, Bit 2 = Select Counter 1, Bit 1 = Select Counter 0, and Bit 0 = Reserved.

Note 2. If bits [5:4] = 00: Register functions as Counter Latch Command  
Bits [7:6] = Selects Counter, and [3:0] = Don't care.

## 6.8.2.8 Port B Control (PIT\_PORTBCTL)

I/O Address 61h  
Type R/W  
Reset Value 00h

## PIT\_PORTBCTL Register Map

7	6	5	4	3	2	1	0
RSVD		OUT2_STS	TOGGLE	RSVD		PIT_CNTR2_SPKR	PIT_CNTR2_EN

## PIT\_PORTBCTL Bit Descriptions

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Read 0. Write "don't care".
5	OUT2_STS (RO)	<b>PIT Counter 2 Out State (Read Only).</b> This bit reflects the current status of the PIT Counter 2 output (OUT2). Write "don't care".
4	TOGGLE (RO)	<b>Toggle (Read Only).</b> This bit toggles on every falling edge of Counter 1 output (OUT1). Write "don't care".
3:2	RSVD	<b>Reserved.</b> Read 0. Write "don't care".
1	PIT_CNTR2_SPKR	<b>PIT Counter 2 (Speaker).</b> 0: Forces speaker output to 0. 1: Allows Counter 2 output (OUT2) to pass to the speaker (i.e., the AC_BEEP signal; a mux option on GPIO1).
0	PIT_CNTR2_EN	<b>PIT Counter 2 Enable.</b> 0: Sets GATE2 input low. 1: Sets GATE2 input high.

## 6.9 Programmable Interrupt Controller Register Descriptions

The registers for the Programmable Interrupt Controller (PIC) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- PIC Specific MSRs
- PIC Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the PIC Specific MSRs are called out as 32 and 8 bits. The PIC treats writes to the

upper 32/56 bits (i.e., bits [63:32/63:8]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The PIC Specific MSRs are also accessible in I/O space via MSR\_LBAR\_IRQ (MSR 51400008h), except for MSR\_PIC\_SHDW (MSR 51400034h). See Section 6.6.2.1 "Local BAR - IRQ Mapper (DIVIL\_LBAR\_IRQ)" on page 325.

The Native registers associated with the PIC are summarized in Table 6-23 on page 349 and are accessed as I/O Addresses.

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-22. PIC Specific MSRs Summary**

MSR Address	PIC I/O Offset	Type	Register Name	Reset Value	Reference
51400020h	00h	R/W	IRQ Mapper Unrestricted Y Select Low (PIC_YSEL_LOW)	00000000h	Page 350
51400021h	04h	R/W	IRQ Mapper Unrestricted Y Select High (PIC_YSEL_HIGH)	00000000h	Page 351
51400022h	08h	R/W	IRQ Mapper Unrestricted Z Select Low (PIC_ZSEL_LOW)	00000000h	Page 350
51400023h	0Ch	R/W	IRQ Mapper Unrestricted Z Select High (PIC_ZSEL_HIGH)	00000000h	Page 351
51400024h	10h	R/W	IRQ Mapper Primary Mask (PIC_IRQM_PRIM)	0000FFFFh	Page 352
51400025h	14h	R/W	IRQ Mapper LPC Mask (PIC_IRQM_LPC)	00000000h	Page 352
51400026h	18h	RO	IRQ Mapper Extended Interrupt Request Status Low (PIC_XIRR_STS_LOW)	xxxxxxxxh	Page 353
51400027h	1Ch	RO	IRQ Mapper Extended Interrupt Request Status High (PIC_XIRR_STS_HIGH)	xxxxxxxxh	Page 354
51400034h	---	RO	PIC Shadow (PIC_SHDW)	xxh	Page 355

**Table 6-23. PIC Native Registers Summary**

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
020h	WO	8	Initialization Command Word 1 (PIC_ICW1) - Master	00h	Page 357
0A0h	WO	8	Initialization Command Word 1 (PIC_ICW1) - Slave	00h	Page 357
021h	WO	8	Initialization Command Word 2 (PIC_ICW2) - Master	00h	Page 357
0A1h	WO	8	Initialization Command Word 2 (PIC_ICW2) - Slave	00h	Page 357
021h	WO	8	Initialization Command Word 3 (PIC_ICW3) - Master	00h	Page 358
0A1h	WO	8	Initialization Command Word 3 (PIC_ICW3) - Slave	00h	Page 358
021h	WO	8	Initialization Command Word 4 (PIC_ICW4) - Master	00h	Page 358
0A1h	WO	8	Initialization Command Word 4 (PIC_ICW4) - Slave	00h	Page 358

Table 6-23. PIC Native Registers Summary (Continued)

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
021h	R/W	8	Operation Command Word 1 / Interrupt Mask (PIC_OCW1/IMR) - Master	00h	Page 359
0A1h	R/W	8	Operation Command Word 1 / Interrupt Mask (PIC_OCW1/IMR) - Slave	00h	Page 359
020h	WO	8	Operation Command Word 2 (PIC_OCW2) - Master	00h	Page 359
0A0h	WO	8	Operation Command Word 2 (PIC_OCW2) - Slave	00h	Page 359
020h	WO	8	Operation Command Word 3 (PIC_OCW3) - Master	00h	Page 360
0A0h	WO	8	Operation Command Word 3 (PIC_OCW3) - Slave	00h	Page 360
020h	RO	8	Interrupt Request Register (PIC_IRR) - Master	00h	Page 361
0A0h	RO	8	Interrupt Request Register (PIC_IRR) - Slave	00h	Page 361
020h	RO	8	In-Service Register (PIC_ISR) - Master	00h	Page 361
0A0h	RO	8	In-Service Register (PIC_ISR) - Slave	00h	Page 361
4D0h	R/W	8	Interrupt Edge/Level Select 1 (PIC_INT_SEL1)	00h	Page 362
4D1h	R/W	8	Interrupt Edge/Level Select 2 (PIC_INT_SEL2)	00h	Page 363

## 6.9.1 PIC Specific MSRs

### 6.9.1.1 IRQ Mapper Unrestricted Y and Z Select Low (PIC\_[Y/Z]SEL\_LOW)

#### IRQ Mapper Unrestricted Y Select Low (PIC\_YSEL\_LOW)

MSR Address 51400020h  
 PIC I/O Offset 00h  
 Type R/W  
 Reset Value 00000000h

#### IRQ Mapper Unrestricted Z Select Low (PIC\_ZSEL\_LOW)

MSR Address 51400022h  
 PIC I/O Offset 08h  
 Type R/W  
 Reset Value 00000000h

#### PIC\_[Y/Z]SEL\_LOW Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAP_[Y/Z]7				MAP_[Y/Z]6				MAP_[Y/Z]5				MAP_[Y/Z]4				MAP_[Y/Z]3				MAP_[Y/Z]2				MAP_[Y/Z]1				MAP_[Y/Z]0			

#### PIC\_[Y/Z]SEL\_LOW Bit Descriptions

Bit	Name	Description
31:28	MAP_[Y/Z]7	<b>Map Unrestricted [Y/Z] Input 7.</b> 0000: Disable    0100: IG4    1000: IG8    1100: IG12 0001: IG1    0101: IG5    1001: IG9    1101: IG13 0010: IG2    0110: IG6    1010: IG10    1110: IG14 0011: IG3    0111: IG7    1011: IG11    1111: IG15 For Unrestricted Y and Z Inputs [7:0] sources, see Table 5-13 and Table 5-14 on page 113.
27:24	MAP_[Y/Z]6	<b>Map Unrestricted [Y/Z] Input 6.</b> See bits [31:28] for decode.
23:20	MAP_[Y/Z]5	<b>Map Unrestricted [Y/Z] Input 5.</b> See bits [31:28] for decode.
19:16	MAP_[Y/Z]4	<b>Map Unrestricted [Y/Z] Input 4.</b> See bits [31:28] for decode.
15:12	MAP_[Y/Z]3	<b>Map Unrestricted [Y/Z] Input 3.</b> See bits [31:28] for decode.

**PIC\_[Y/Z]SEL\_LOW Bit Descriptions**

Bit	Name	Description
11:8	MAP_[Y/Z]2	<b>Map Unrestricted [Y/Z] Input 2.</b> See bits [31:28] for decode.
7:4	MAP_[Y/Z]1	<b>Map Unrestricted [Y/Z] Input 1.</b> See bits [31:28] for decode.
3:0	MAP_[Y/Z]0	<b>Map Unrestricted [Y/Z] Input 0.</b> See bits [31:28] for decode.

**6.9.1.2 IRQ Mapper Unrestricted Y and Z Select High (PIC\_[Y/Z]SEL\_HIGH)**

**IRQ Mapper Unrestricted Y Select High (PIC\_YSEL\_HIGH)**

MSR Address 51400021h  
 PIC I/O Offset 04h  
 Type R/W  
 Reset Value 00000000h

**IRQ Mapper Unrestricted Z Select High (PIC\_ZSEL\_HIGH)**

MSR Address 51400023h  
 PIC I/O Offset 0Ch  
 Type R/W  
 Reset Value 00000000h

**PIC\_[Y/Z]SEL\_HIGH Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAP_[Y/Z]15				MAP_[Y/Z]14				MAP_[Y/Z]13				MAP_[Y/Z]12				MAP_[Y/Z]11				MAP_[Y/Z]10				MAP_[Y/Z]9				MAP_[Y/Z]8			

**PIC\_[Y/Z]SEL\_HIGH Bit Descriptions**

Bit	Name	Description
31:28	MAP_[Y/Z]_15	<b>Map Unrestricted [Y/Z] Input 15.</b>  0000: Disable    0100: IG4        1000: IG8        1100: IG12 0001: IG1        0101: IG5        1001: IG9        1101: IG13 0010: IG2        0110: IG6        1010: IG10       1110: IG14 0011: IG3        0111: IG7        1011: IG11       1111: IG15  For Unrestricted Y and Z Inputs [7:0] sources, see Table 5-13 and Table 5-14 on page 113.
27:24	MAP_[Y/Z]_14	<b>Map Unrestricted [Y/Z] Input 14.</b> See bits [31:28] for decode.
23:20	MAP_[Y/Z]_13	<b>Map Unrestricted [Y/Z] Input 13.</b> See bits [31:28] for decode.
19:16	MAP_[Y/Z]_12	<b>Map Unrestricted [Y/Z] Input 12.</b> See bits [31:28] for decode.
15:12	MAP_[Y/Z]_11	<b>Map Unrestricted [Y/Z] Input 11.</b> See bits [31:28] for decode.
11:8	MAP_[Y/Z]_10	<b>Map Unrestricted [Y/Z] Input 10.</b> See bits [31:28] for decode.
7:4	MAP_[Y/Z]_9	<b>Map Unrestricted [Y/Z] Input 9.</b> See bits [31:28] for decode.
3:0	MAP_[Y/Z]_8	<b>Map Unrestricted [Y/Z] Input 8.</b> See bits [31:28] for decode.

**6.9.1.3 IRQ Mapper Primary Mask (PIC\_IRQM\_PRIM)**

MSR Address 51400024h  
 PIC I/O Offset 10h  
 Type R/W  
 Reset Value 0000FFFFh

**PIC\_IRQM\_PRIM Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																PRIM15_MSK	PRIM14_MSK	PRIM13_MSK	PRIM12_MSK	PRIM11_MSK	PRIM10_MSK	PRIM9_MSK	PRIM8_MSK	PRIM7_MSK	PRIM6_MSK	PRIM5_MSK	PRIM4_MSK	PRIM3_MSK	RSVD	PRIM1_MSK	PRIM0_MSK

**PIC\_IRQM\_PRIM Bit Descriptions**

Bit	Name	Description
31:16	RSVD	<b>Reserved.</b> Set to 0.
15:0	PRIM[15:0]_MSK	<b>Primary Inputs [15:0] Mask.</b> Bits [15:0] correspond to Primary Inputs [15:0], bit 2 is reserved (i.e., no IRQ2). 0: Mask the interrupt source. 1: Do not mask the interrupt source. For Primary Inputs [15:0] sources, see Table 5-12 on page 112.

**6.9.1.4 IRQ Mapper LPC Mask (PIC\_IRQM\_LPC)**

MSR Address 51400025h  
 PIC I/O Offset 14h  
 Type R/W  
 Reset Value 00000000h

**PIC\_IRQM\_LPC Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LPC15_EN	LPC14_EN	LPC13_EN	LPC12_EN	LPC11_EN	LPC10_EN	LPC9_EN	LPC8_EN	LPC7_EN	LPC6_EN	LPC5_EN	LPC4_EN	LPC3_EN	RSVD	LPC1_EN	LPC0_EN

**PIC\_IRQM\_LPC Bit Descriptions**

Bit	Name	Description
31:16	RSVD	<b>Reserved.</b> Set to 0.
15:0	LPC[15:0]_EN	<b>LPC Inputs [15:0] Enable.</b> Bits [15:0] correspond to LPC Inputs [15:0], bit 2 is don't care (i.e., no IRQ2). 0: Disable interrupt source. 1: Enable interrupt source. For LPC Inputs [15:0] sources, see Table 5-12 on page 112.



**6.9.1.5 IRQ Mapper Extended Interrupt Request Status Low (PIC\_XIRR\_STS\_LOW)**

MSR Address 51400026h  
 PIC I/O Offset 18h  
 Type RO  
 Reset Value xxxxxxxxh

**PIC\_XIRR\_STS\_LOW Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IG7_STS				IG6_STS				IG5_STS				IG4_STS				IG3_STS				IG2_STS		RSVD		IG1_STS				RSVD		IG0_STS	

**PIC\_XIRR\_STS\_LOW Bit Descriptions**

Bit	Name	Description
31:28	IG7_STS	<b>Interrupt Group 7 Status.</b> Reports the status of the four interrupts in this group. bit 28: Primary Input 7. bit 29: LPC Input 7. bit 30: Unrestricted Y Input 7. bit 31: Unrestricted Source Z Input 7.
27:24	IG6_STS	<b>Interrupt Group 6 Status.</b> Reports the status of the four interrupts in this group. bit 24: Primary Input 6. bit 25: LPC Input 6. bit 26: Unrestricted Y Input 6. bit 27: Unrestricted Z Input 6.
23:20	IG5_STS	<b>Interrupt Group 5 Status.</b> Reports the status of the four interrupts in this group. bit 20: Primary Input 5. bit 21: LPC Input 5. bit 22: Unrestricted Y Input 5. bit 23: Unrestricted Z Input 5.
19:16	IG4_STS	<b>Interrupt Group 4 Status.</b> Reports the status of the four interrupts in this group. bit 16: Primary Input 4. bit 17: LPC Input 4. bit 18: Unrestricted Y Input 4. bit 19: Unrestricted Z Input 4.
15:12	IG3_STS	<b>Interrupt Group 3 Status.</b> Reports the status of the four interrupts in this group. bit 12: Primary Input 3. bit 13: LPC Input 3. bit 14: Unrestricted Y Input 3. bit 15: Unrestricted Z Input 3.
11:10	IG2_STS	<b>Interrupt Group 2 Status.</b> Reports the status of the two interrupts in this group. bit10: Unrestricted Y Input 2. bit 11: Unrestricted Z Input 2.
9:8	RSVD	<b>Reserved.</b> Always reads 0; no connection to any interrupts.
7:4	IG1_STS	<b>Interrupt Group 1 Status.</b> Reports the status of the four interrupts in this group. bit 4: Primary Input 1. bit 5: LPC Input 1. bit 6: Unrestricted Y Input 1. bit 7: Unrestricted Z Input 1.
3:2	RSVD	<b>Reserved.</b> Always reads 0; no connection to any interrupts.

**PIC\_XIRR\_STS\_LOW Bit Descriptions (Continued)**

Bit	Name	Description
1:0	IG0_STS	<b>Interrupt Group 0 Status.</b> Reports the status of the two interrupts in this group. bit 0: Primary Input 0. bit 1: LPC Input 0.

**6.9.1.6 IRQ Mapper Extended Interrupt Request Status High (PIC\_XIRR\_STS\_HIGH)**

MSR Address 51400027h  
PIC I/O Offset 1Ch  
Type RO  
Reset Value xxxxxxxxh

**PIC\_XIRR\_STS\_HIGH Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IG15_STS				IG14_STS				IG13_STS				IG12_STS				IG11_STS				IG10_STS				IG9_STS				IG8_STS			

**PIC\_XIRR\_STS\_HIGH Bit Descriptions**

Bit	Name	Description
31:28	IG15_STS	<b>Group 15 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 28: Primary Input 15. bit 29: LPC Input 15. bit 30: Unrestricted Y Input 15. bit 31: Unrestricted Z Input 15.
27:24	IG14_STS	<b>Group 14 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 24: Primary Input 14. bit 25: LPC Input 14. bit 26: Unrestricted Y Input 14. bit 27: Unrestricted Z Input 14.
23:20	IG13_STS	<b>Group 13 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 20: Primary Input 13. bit 21: LPC Input 13. bit 22: Unrestricted Y Input 13. bit 23: Unrestricted Z Input 13.
19:16	IG12_STS	<b>Group 12 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 16: Primary Input 12. bit 17: LPC Input 12. bit 18: Unrestricted Y Input 12. bit 19: Unrestricted Z Input 12.
15:12	IG11_STS	<b>Group 11 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 12: Primary Input 11. bit 13: LPC Input 11. bit 14: Unrestricted Y Input 11. bit 15: Unrestricted Z Input 11.
11:8	IG10_STS	<b>Group 10 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 08: Primary Input 10. bit 09: LPC Input 10. bit 10: Unrestricted Y Input 10. bit 11: Unrestricted Z Input 10.

**PIC\_XIRR\_STS\_HIGH Bit Descriptions (Continued)**

Bit	Name	Description
7:4	IG9_STS	<b>Group 9 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 4: Primary Input 9. bit 5: LPC Input 9. bit 6: Unrestricted Y Input 9. bit 7: Unrestricted Z Input 9.
3:0	IG8_STS	<b>Group 8 Interrupt Status.</b> Reports the status of the four interrupts in this group. bit 0: Primary Input 8. bit 1: LPC Input 8. bit 2: Unrestricted Y Input 8. bit 3: Unrestricted Z Input 8.

**6.9.1.7 PIC Shadow (PIC\_SHDW)**

MSR Address 51400034h  
 Type RO  
 Reset Value xxh

**PIC\_SHDW Register Map**

7	6	5	4	3	2	1	0
PIC_SHDW							

**PIC\_SHDW Bit Descriptions**

Bit	Name	Description
7:0	PIT_SHDW (RO)	<b>PIC Shadow (Read Only).</b> This 8-bit port sequences through the following list of shadowed Programmable Interrupt Controller registers. At power on, a pointer starts at the first register in the list and consecutively reads incrementally through it. A write to this register resets the read sequence to the first register. Each shadow register in the sequence contains the last data written to that location.  The read sequence for this register is: <ol style="list-style-type: none"> <li>1. PIC1 ICW1.</li> <li>2. PIC1 ICW2.</li> <li>3. PIC1 ICW3.</li> <li>4. PIC1 ICW4 - Bits [7:5] of ICW4 are always 0.</li> <li>5. PIC1 OCW2 - Bits [6:3] of OCW2 are always 0 (Note 1).</li> <li>6. PIC1 OCW3 - Bits [7, 4] are 0 and bit [6, 3] are 1.</li> <li>7. PIC2 ICW1.</li> <li>8. PIC2 ICW2.</li> <li>9. PIC2 ICW3.</li> <li>10. PIC2 ICW4 - Bits [7:5] of ICW4 are always 0.</li> <li>11. PIC2 OCW2 - Bits [6:3] of OCW2 are always 0 (Note 1).</li> <li>12. PIC2 OCW3 - Bits [7, 4] are 0 and bit [6, 3] are 1.</li> </ol>

Note 1. To restore OCW2 to shadow register value, write the appropriate address twice. First with the shadow register value, then with the shadow register value ORed with C0h.

## 6.9.2 PIC Native Registers

There are two separate PIC sub-blocks in the Geode CS5535 companion device, connected in a cascaded arrangement, as is required for a PC-compatible system. Each PIC has its own native register set, apart from the MSR registers (unique to the Geode CS5535 companion device architecture), which are common.

The master PIC occupies I/O Addresses 020h and 021h, and manages IRQ signals IRQ0 through IRQ7, with IRQ2 claimed as the cascade input for the slave PIC. The slave PIC occupies I/O Addresses 0A0h and 0A1h, and manages IRQ signals IRQ8 through IRQ15. In this description, the two addresses of a PIC are called the Even address (A[0] = 0) and the Odd address (A[0] = 1).

The PIC register set addressing is often confusing due to some very severe constraints the PIC had in its earliest history. When it was a separate chip, the package pinout limited it to only one address line. To make up for this, two bits of the data written (bits 3 and 4) sometimes serve an addressing function to select registers.

The chip functions in two fundamental modes with respect to register accesses: it is either in *Operation* mode (normal operation), or it is in *Initialization* mode (being initialized). Different sets of registers are selected in each mode.

### Operation Mode

When the PIC is in Operation mode, a set of registers may be accessed, called the Operation Command Words (OCWs).

- OCW1: The Interrupt Mask register (IMR), may be read or written at any time except during Initialization.
- OCW2: A write-only register that is given commands from software. For example, the End of Interrupt command is written here at the end of interrupt service to terminate the blocking of interrupts on the basis of priority.
- OCW3: A write-only register that is given a different set of commands from software. For example, it is through this register that software can request images of two internal registers:
  - IRR: Interrupt Request Register -- shows those IRQs with pending interrupts that have not yet received an Interrupt Acknowledge from the CPU.
  - ISR: In-Service Register -- shows those IRQs that have received Interrupt Acknowledge, but whose interrupt service routines have not yet completed.

### Initialization Mode

The PIC is placed into its Initialization mode by a write of a reserved value (xxx1xxxx) to the even-numbered address (master 0020h / slave 00A0h). This is the first of a sequence of writes to a special set of Initialization Control Word registers (ICW1, ICW2, ICW3 and ICW4) that hold permanent settings and are normally touched only while the operating system is booting.

## 6.9.2.1 Register Addressing Scheme

### Writing to the Even Address

Data Bit		Access Performed
4	3	
0	0	In Operation mode, writes to OCW2, asserting a routine command.
0	1	In Operation mode, writes to OCW3, asserting a special or diagnostic command. Commands written to OCW3 may request to examine an internal PIC register; if so, this (even) address must be immediately read to retrieve the requested value and terminate the command. See "Reading from the Even Address" below.
1	0 or 1	Triggers Initialization mode and writes to ICW1. Bit 3 is used as a data bit in this case.

Other write and read accesses do not depend directly on this form of addressing.

### Writing to the Odd Address

- Operation Mode: Writes to the Interrupt Mask Register: OCW1.
- Initialization Mode: Three successive writes to this address must immediately follow the write to ICW1 (above), before any other accesses are performed to the PIC. These writes load ICW2, ICW3 and ICW4 in succession, after which the PIC automatically transitions to Operation mode.

### Reading from the Even Address

(Operation mode only: the Initialization mode does not involve reading.)

Reads from this address are generally for special or diagnostic purposes. The read must be preceded by writing a command to OCW3 (above), to select an internal register to read. This will be the IRR or ISR.

(Another register, "POLL", historically part of the PIC architecture, is not provided in the Geode CS5535 companion device.)

Following that command, the read here will return the requested value.

### Reading from the Odd Address

(Operation mode only: the Initialization mode does not involve reading.)

Always reads from the Interrupt Mask Register: OCW1.

### Associated External Registers

Two directly addressable read/write registers, outside the addresses of the PICs themselves, have been added to allow individual control of which IRQs are level sensitive vs. edge sensitive.

- INT\_SEL1, I/O Address 04D0h, controls IRQ1, IRQ3-7
- INT\_SEL2, I/O Address 04D1h, controls IRQ8-15

**6.9.2.2 Initialization Command Word 1 (PIC\_ICW1)**

I/O Port            Master: 020h  
                       Slave: 0A0h  
 Type                WO  
 Reset Value        00h

**PIC\_ICW1 Register Map**

7	6	5	4	3	2	1	0
RSVD			1	TRIGGER	RSVD	RSVD	RSVD

**PIC\_ICW1 Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Write to 0.
4	1	<b>Write to 1.</b> Write to 1 to write ICW1 and enter Initialization mode. (See Section 6.9.2 "PIC Native Registers" on page 356.)
3	TRIGGER	<b>Trigger Mode.</b> 0: Edge. 1: Level.
2	RSVD	<b>Reserved.</b> Write to 0.
1	RSVD	<b>Reserved.</b> Write to 0.
0	RSVD	<b>Reserved.</b> Write to 1.

**6.9.2.3 Initialization Command Word 2 (PIC\_ICW2)**

I/O Port            Master: 021h  
                       Slave: 0A1h  
 Type                WO  
 Reset Value        00h

**PIC\_ICW2 Register Map**

7	6	5	4	3	2	1	0
A[7:3]					RSVD		

**PIC\_ICW2 Bit Descriptions**

Bit	Name	Description
7:3	A[7:3]	<b>Address Lines [7:3].</b> For base vector of interrupt controller.
2:0	RSVD	<b>Reserved.</b> Write to 0.

**6.9.2.4 Initialization Command Word 3 (PIC\_ICW3)**

I/O Port            Master: 021h  
                       Slave: 0A1h  
 Type                WO  
 Reset Value        00h

**PIC\_ICW3 Register Map**

7	6	5	4	3	2	1	0
Master = CASCADE_IRQ; Slave = SLAVE_ID							

**PIC\_ICW3 Bit Descriptions**

Bit	Name	Description
<b>Master</b>		
7:0	CASCADE_IRQ	<b>Cascade IRQ.</b> Must be written to 04h.
<b>Slave</b>		
7:0	SLAVE_ID	<b>Slave ID.</b> Must be written to 02h.

**6.9.2.5 Initialization Command Word 4 (PIC\_ICW4)**

I/O Port            Master: 021h  
                       Slave: 0A1h  
 Type                WO  
 Reset Value        00h

**PIC\_ICW4 Register Map**

7	6	5	4	3	2	1	0
RSVD			SPEC_NST	RSVD		AUTO_EOI	RSVD

**PIC\_ICW4 Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Write to 0.
4	SPEC_NST	<b>Reserved (Special Fully Nested mode is not supported).</b> Write to 0.
3:2	RSVD	<b>Reserved.</b> Write to 0.
1	AUTO_EOI	<b>Auto End of Interrupt.</b> This feature is present, but is not recommended for use. When set to 1, this bit causes the PIC to automatically issue an End of Interrupt internally, immediately after each Interrupt Acknowledge from the CPU. When cleared to 0 (the default), it requires software action (writing to the OCW2 register) to signal End of Interrupt.  0: Normal EOI. 1: Auto EOI (not recommended).
0	RSVD	<b>Reserved.</b> Write to 1 (8086/8088 mode).

**6.9.2.6 Operation Command Word 1 / Interrupt Mask (PIC\_OCW1/IMR)**

I/O Port            Master: 021h  
                       Slave: 0A1h  
 Type                R/W  
 Reset Value        00h

**PIC\_OCW1/IMR Register Map**

7	6	5	4	3	2	1	0
IRQ7_15M	IRQ6_14M	IRQ5_13M	IRQ4_12M	IRQ3_11M	IRQ2_10M	IRQ1_9M	IRQ0_8M

**PIC\_OCW1/IMR Bit Descriptions**

Bit	Name	Description
7	IRQ7_15M	<b>IRQ7 / IRQ15 Mask.</b> 0: Not Masked; 1: Masked.
6	IRQ6_14M	<b>IRQ6 / IRQ14 Mask.</b> 0: Not Masked; 1: Masked.
5	IRQ5_13M	<b>IRQ5 / IRQ13 Mask.</b> 0: Not Masked; 1: Masked.
4	IRQ4_12M	<b>IRQ4 / IRQ12 Mask.</b> 0: Not Masked; 1: Masked.
3	IRQ3_11M	<b>IRQ3 / IRQ11 Mask.</b> 0: Not Masked; 1: Masked.
2	IRQ2_10M	<b>IRQ2 / IRQ10 Mask.</b> 0: Not Masked; 1: Masked.
1	IRQ1_9M	<b>IRQ1 / IRQ9 Mask.</b> 0: Not Masked; 1: Masked.
0	IRQ0_8M	<b>IRQ0 / IRQ8 Mask.</b> 0: Not Masked; 1: Masked.

**6.9.2.7 Operation Command Word 2 (PIC\_OCW2)**

I/O Port            Master: 020h  
                       Slave: 0A0h  
 Type                WO  
 Reset Value        00h

**PIC\_OCW2 Register Map**

7	6	5	4	3	2	1	0
ROT_EOI			00		IRQ		

**PIC\_OCW2 Bit Descriptions**

Bit	Name	Description
7:5	ROT_EOI	<b>Rotate/EOI Codes.</b> 000: Clear rotate in Auto EOI mode. 100: Set rotate in Auto EOI mode. 001: Non-specific EOI. 101: Rotate on non-specific EOI command. 010: No operation. 110: Set priority command (bits [2:0] must be valid). 011: Specific EOI (bits [2:0] must be valid). 111: Rotate on specific EOI command (bits [2:0] must be valid)
4:3	00	<b>Write to 0.</b> Write to 00 to write OCW2 (rather than OCW3 or ICW1). (See Section 6.9.2 "PIC Native Registers" on page 356.)
2:0	IRQ	<b>IRQ Number (000-111).</b>

**6.9.2.8 Operation Command Word 3 (PIC\_OCW3)**

I/O Port            Master: 020h  
                       Slave: 0A0h  
 Type                WO  
 Reset Value        00h

**PIC\_OCW3 Register Map**

7	6	5	4	3	2	1	0
RSVD	SP_MASK		01		RSVD	REG_READ	

**PIC\_OCW3 Bit Descriptions**

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write to 0.
6:5	SP_MASK	<b>Special Mask Mode.</b> The internal SMM bit can be set or cleared using this 2-bit field. 0x: No change to the internal SMM bit. 10: Clears the internal SMM bit (i.e., value of SMM bit = 0). (Default after initialization.) 11: Sets the internal SMM bit (i.e., value of SMM bit = 1).  While the internal SMM bit is 1, interrupt blocking by priority is disabled, and only the Interrupt Mask Register (OCW1) is used to block interrupt requests to the CPU. While the internal SMM bit is 0 (the default), an unmasked IRQ must also be of higher priority than the IRQ of the currently running interrupt service routine. Regardless of the setting of this bit, the IRQ priority is still used to arbitrate among multiple allowed IRQ requests at the time of an Interrupt Acknowledge access from the CPU.
4:3	01	<b>Write to 01.</b> Write to 01 to write OCW3 (rather than OCW2 or ICW1). (See Section 6.9.2 "PIC Native Registers" on page 356.)
2	RSVD	<b>Reserved.</b> Write to 0. (Poll Command at this address is not supported.)
1:0	REG_READ	<b>Register Read Mode.</b> 00: No operation. 10: Read interrupt request register on next read of I/O Port 020h (master) or 0A0h (slave). 01: No operation. 11: Read interrupt service register on next read of I/O Port 020h (master) or 0A0h (slave).



### 6.9.2.9 Interrupt Request Register (PIC\_IRR)

I/O Port           Master: 020h  
                       Slave: 0A0h  
 Type                RO  
 Reset Value        00h

This register is accessible only after the appropriate command is written to OCW3.

**PIC\_IRR Register Map**

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
IRQ7_15STS	IRQ6_14STS	IRQ5_13STS	IRQ4_12STS	IRQ3_11STS	IRQ2_10STS	IRQ1_9STS	IRQ0_8STS

**PIC\_IRR Bit Descriptions**

Bit	Name	Description
7	IRQ7_15STS	<b>IRQ7 / IRQ15 Status (Pending).</b> 0: Yes; 1: No.
6	IRQ6_14STS	<b>IRQ6 / IRQ14 Status (Pending).</b> 0: Yes; 1: No.
5	IRQ5_13STS	<b>IRQ5 / IRQ13 Status (Pending).</b> 0: Yes; 1: No.
4	IRQ4_12STS	<b>IRQ4 / IRQ12 Status (Pending).</b> 0: Yes; 1: No.
3	IRQ3_11STS	<b>IRQ3 / IRQ11 Status (Pending).</b> 0: Yes; 1: No.
2	IRQ2_10STS	<b>IRQ2 / IRQ10 Status (Pending).</b> 0: Yes; 1: No.
1	IRQ1_9STS	<b>IRQ1 / IRQ9 Status (Pending).</b> 0: Yes; 1: No.
0	IRQ0_8STS	<b>IRQ0 / IRQ8 Status (Pending).</b> 0: Yes; 1: No.

### 6.9.2.10 In-Service Register (PIC\_ISR)

I/O Port           Master: 020h  
                       Slave: 0A0h  
 Type                RO  
 Reset Value        00h

This register is accessible only after the appropriate command is written to OCW3.

**PIC\_ISR Register Map**

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
IRQ7_15IS	IRQ6_14IS	IRQ5_13IS	IRQ4_12IS	IRQ3_11IS	IRQ2_10IS	IRQ1_9IS	IRQ0_8IS

**PIC\_ISR Bit Descriptions**

Bit	Name	Description
7	IRQ7_15IS	<b>IRQ7 / IRQ15 In-Service.</b> 0: No; 1: Yes.
6	IRQ6_14IS	<b>IRQ6 / IRQ14 In-Service.</b> 0: No; 1: Yes.
5	IRQ5_13IS	<b>IRQ5 / IRQ13 In-Service.</b> 0: No; 1: Yes.
4	IRQ4_12IS	<b>IRQ4 / IRQ12 In-Service.</b> 0: No; 1: Yes.
3	IRQ3_11IS	<b>IRQ3 / IRQ11 In-Service.</b> 0: No; 1: Yes.
2	IRQ2_10IS	<b>IRQ2 / IRQ10 In-Service.</b> 0: No; 1: Yes.
1	IRQ1_9IS	<b>IRQ1 / IRQ9 In-Service.</b> 0: No; 1: Yes.
0	IRQ0_8IS	<b>IRQ0 / IRQ8 In-Service.</b> 0: No; 1: Yes.

**6.9.2.11 Interrupt Edge/Level Select 1 (PIC\_INT\_SEL1)**

I/O Port 4D0h  
 Type R/W  
 Reset Value 00h

**PIC\_INT\_SEL1 Register Map**

7	6	5	4	3	2	1	0
IRQ7_SEL	IRQ6_SEL	IRQ5_SEL	IRQ4_SEL	IRQ3_SEL	RSVD	IRQ1_SEL	RSVD

**PIC\_INT\_SEL1 Bit Descriptions**

Bit	Name	Description
7	IRQ7_SEL	<b>IRQ7 Edge or Level Select.</b> Selects PIC IRQ7 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
6	IRQ6_SEL	<b>IRQ6 Edge or Level Select.</b> Selects PIC IRQ6 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
5	IRQ5_SEL	<b>IRQ5 Edge or Level Select.</b> Selects PIC IRQ5 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
4	IRQ4_SEL	<b>IRQ4 Edge or Level Select.</b> Selects PIC IRQ4 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
3	IRQ3_SEL	<b>IRQ3 Edge or Level Select.</b> Selects PIC IRQ7 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
2	RSVD	<b>Reserved.</b> Write to 0.
1	IRQ1_SEL	<b>IRQ1 Edge or Level Select.</b> Selects PIC IRQ1 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
0	RSVD	<b>Reserved.</b> Write to 0.

Note 1. If ICW1 bit 3 in the PIC is set as level, it overrides the settings of this bit. This bit is provided to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).

**6.9.2.12 Interrupt Edge/Level Select 2 (PIC\_INT\_SEL2)**

I/O Port                   4D1h  
 Type                       R/W  
 Reset Value               00h

**PIC\_INT\_SEL2 Register Map**

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
IRQ7_SEL	IRQ6_SEL	IRQ5_SEL	IRQ4_SEL	IRQ3_SEL	RSVD	IRQ1_SEL	RSVD

**PIC\_INT\_SEL2 Bit Descriptions**

Bit	Name	Description
7	IRQ15_SEL	<b>IRQ15 Edge or Level Select.</b> Selects PIC IRQ15 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
6	IRQ14_SEL	<b>IRQ14 Edge or Level Select.</b> Selects PIC IRQ14 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
5	RSVD	<b>Reserved.</b> Write to 0.
4	IRQ12_SEL	<b>IRQ12 Edge or Level Select.</b> Selects PIC IRQ12 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
3	IRQ11_SEL	<b>IRQ11 Edge or Level Select.</b> Selects PIC IRQ11 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
2	IRQ10_SEL	<b>IRQ10 Edge or Level Select.</b> Selects PIC IRQ10 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
1	IRQ9_SEL	<b>IRQ9 Edge or Level Select.</b> Selects PIC IRQ9 sensitivity configuration. 0: Edge; 1: Level. (Note 1)
0	RSVD	<b>Reserved.</b> Write to 0.

Note 1. If ICW1 bit 3 in the PIC is set as level, it overrides the settings of this bit. This bit is provided to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).

## 6.10 Keyboard Emulation Logic Register Descriptions

The registers for the Keyboard Emulation Logic (KEL) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- KEL Specific MSRs
- KEL Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the KEL Specific MSRs are called out as 32 bits. The KEL treats writes to the upper 32 bits (i.e., bits [63:32]) of the 64-bit MSRs as don't cares and always returns 0 on these bits. The KEL Specific MSRs are summarized in Table 6-24.

Four Native registers are used to provide the keyboard emulation support, summarized in Table 6-25:

- **KEL HCE Control Register:** Used to enable and control the emulation hardware and report various status information.

- **KEL HCE Input Register:** Emulation side of the legacy 8048 Controller Input Buffer register. Writes to I/O Port 060h and 064h are read here.
- **KEL HCE Output Register:** Emulation side of the legacy 8048 Controller Output Buffer register where keyboard and mouse data is to be written by software. Reads from I/O Port 060h are setup here.
- **KEL HCE Status Register:** Emulation side of the legacy 8048 Controller Status register. Reads from I/O Port 60h are setup here.

Each of the Native registers is located on a 32-bit boundary. The Offset of these registers is relative to the base address. (See Section 6.6.2.2 "Local BAR - KEL from USB Host Controller 1 (DIVIL\_LBAR\_KEL1)" on page 326 and Section 6.6.2.3 "Local BAR - KEL from USB Host Controller 2 (DIVIL\_LBAR\_KEL2)" on page 326.) Any writes to locations outside these offsets are a "don't care". Any reads to locations outside these offsets return zero.

Three of the operational registers (HCE\_Status, HCE\_Input, HCE\_Output), summarized in Table 6-25, are accessible at I/O Port 060h and 064h when emulation is enabled. Port A is at I/O Port 092h. Reads and writes to the registers using I/O addresses have side effects as outlined in Table 6-26.

**Table 6-24. KEL Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
5140001Fh	R/W	Keyboard Emulation Logic Control Register (KELX_CTL)	00000010h	Page 365

**Table 6-25. KEL Native Registers Summary**

KEL Memory Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
100h	R/W	32	KEL HCE Control Register (KEL_HCE_CTRL)	00000000h	Page 366
104h	R/W	32	KEL HCE Input (KEL_HCE_IN)	000000xxh	Page 367
108h	R/W	32	KEL HCE Output (KEL_HCE_OUT)	000000xxh	Page 367
10Ch	R/W	32	KEL HCE Status (KEL_HCE_STS)	00000000h	Page 368
092h	R/W	8	Port A (KEL_PORTA)	00h	Page 369

**Table 6-26. KEL Legacy Registers Emulated Summary**

I/O Port	I/O Cycle	Register Contents Accessed/Modified	Side Effects in Emulation Mode
060h	Read	HCE_Output	Read from Port 060h clears OutputFull in HCE_Status to 0.
060h	Write	HCE_Input	Write to Port 060h sets InputFull to 1 and CmdData to 0 in HCE_Status.
064h	Read	HCE_Status	Read from Port 064h returns current value of HCE_Status with no side effects.
064h	Write	HCE_Input	Write to Port 064h will set InputFull to 0 and CmdData in HCE_Status to 1.

## 6.10.1 KEL Specific MSRs

### 6.10.1.1 Keyboard Emulation Logic Control Register (KELX\_CTL)

MSR Address 5140001Fh  
 Type R/W  
 Reset Value 00000010h

Port A operation is not effected by Snoop or EmulationEnable settings in KELX\_CTL. Bits [31:5] are Reserved. Writes are “don’t care” and reads always return zero.

**KELX\_CTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											PRTA_EN	SOFEVENT	EER	SNOOP	

**KELX\_CTL Bit Descriptions**

Bit	Name	Description									
31:5	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.									
4	PRTA_EN	<b>Port A Enable.</b> Defaults high. If high, Port A is enabled and accesses to I/O Port 092h are processed by the Geode CS5535 companion device. If low, accesses to I/O Port 092h are passed on to the LPC bus (where Port A may exist inside a SuperI/O).									
3:2	SOFEVENT	<b>Start-Of-Frame (SOF) Event (SOFEVENT) Selection.</b> 00: Test Mode (No delays). 01: USB1 (Default). 10: USB2. 11: 1 ms from PIT.									
1	EER	<b>Emulation Event (EE) Routing.</b> 0: Emulation Interrupt and ASMI. 1: ASMI only. The EER bit controls keyboard emulation interrupt generation associated with EEs: <ul style="list-style-type: none"> <li>Character Pending - Clear EE by setting the Outputfull bit in the HCE_Status register or clearing the CharacterPending bit in HCE_Control.</li> <li>Input Full - Clear EE by clearing the InputFull bit in HCE_Status.</li> <li>External IRQ - Clear EE by clearing IRQ1Active or IRQ12Active as appropriate in HCE_Control.</li> </ul>									
0	SNOOP	<b>Snoop.</b> Only applies when EmulationEnable in HCE_Control is low. When high, indicates A20 and Init keyboard sequences are to be detected on I/O Port 060h/064h transactions to the LPC keyboard. KEL must generate ASMI upon detection of A20 and Init even though EmulationEnable in HCE_Control is low. Emulation Enable Snoop <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">0</td><td style="padding-right: 10px;">1</td><td>PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h.</td></tr> <tr> <td>1</td><td>x</td><td>PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. KEL_ASMI is generated on InputFull, External IQ, or Character Pending. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h For the A20 keyboard sequence, both KEL_ASMI and PORTA_A20_ASMI are signaled.</td></tr> <tr> <td>0</td><td>0</td><td>OFF.</td></tr> </table>	0	1	PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h.	1	x	PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. KEL_ASMI is generated on InputFull, External IQ, or Character Pending. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h For the A20 keyboard sequence, both KEL_ASMI and PORTA_A20_ASMI are signaled.	0	0	OFF.
0	1	PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h.									
1	x	PORTA_A20_ASMI or KEL_INIT_ASMI is generated when sequence is detected. KEL_ASMI is generated on InputFull, External IQ, or Character Pending. Status of ASMI Flag and Enable bits are in DIVIL MSR 51400002h For the A20 keyboard sequence, both KEL_ASMI and PORTA_A20_ASMI are signaled.									
0	0	OFF.									

## 6.10.2 KEL Native Registers

### 6.10.2.1 KEL HCE Control Register (KEL\_HCE\_CTRL)

KEL Memory Offset 100h

Type R/W

Reset Value 00000000h

**KEL\_HCE\_CTRL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							A20State	IRQ12Active	IRQ1Active	A20Sequence	ExternallIRQEn	IRQEn	CharacterPending	EmulationInterrupt	EmulationEnable

**KEL\_HCE\_CTRL Bit Descriptions**

Bit	Name	Description
31:9	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.
8	A20State	<b>A20 State.</b> Indicates current state of A20 on the LPC keyboard controller. Used to compare against value written to I/O Port 060h when A20Sequence is active. A20State is set and cleared only by software.
7	IRQ12Active	<b>IRQ12 Active.</b> Indicates that a positive transition on IRQ12 from the LPC keyboard controller has occurred. Software may write 1 to this bit to clear (0) it. A software write of 0 to this bit has no effect.
6	IRQ1Active	<b>IRQ1 Active.</b> Indicates that a positive transition on IRQ1 from the LPC keyboard controller has occurred. Software may write 1 to this bit to clear (0) it. A software write of a 0 to this bit has no effect.
5	A20Sequence	<b>A20 Sequence.</b> Set by KEL when a data value of D1h is written to I/O Port 064h. Cleared by KEL on write to I/O Port 064h of any value other than D1h.
4	ExternallIRQEn	<b>External Interrupt Request Enable.</b> When set to 1, IRQ1 and IRQ12 from the LPC keyboard controller causes an Emulation Event. The function controlled by this bit is independent of the setting of the EmulationEnable bit (bit 0).
3	IRQEn	<b>Interrupt Request Enable.</b> When set, the KEL generates IRQ1 or IRQ12 as long as the OutputFull bit in HCE_Status is set to 1. If the AuxOutputFull bit of HceStatus is 0, then IRQ1 is generated; if it is 1, then an IRQ12 is generated.
2	CharacterPending	<b>Character Pending,</b> When set, an EE is generated when the OutputFull bit of the HCE_Status register is cleared to 0.
1	Emulation Interrupt (RO)	<b>Emulation Interrupt (Read Only).</b> This bit is a static decode of the Emulation Enable state. Returns 1 if: CharacterPending = 1 OR InputFull = 1 OR ExternallIRQEn = 1 AND (IRQ1Active OR IRQ12Active = 1).
0	EmulationEnable	<b>Emulation Enable.</b> When set to 1, the KEL is enabled for legacy emulation. The KEL decodes accesses to I/O Port 060h and 064h and generates IRQ1 and/or IRQ12 when appropriate. Additionally, the KEL generates an ASMI at appropriate times to invoke the emulation software.

**6.10.2.2 KEL HCE Input (KEL\_HCE\_IN)**

KEL Memory Offset 104h  
 Type R/W  
 Reset Value 000000xxh

I/O data that is written to I/O Port 060h and 064h is captured in the HCE\_Input register when emulation is enabled. This register may be read or written directly by accessing it with its memory address in the KEL's operational register space. When accessed directly via the KEL's operational address space, reads and writes of this register have no side effects.

**KEL\_HCE\_IN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														Input_Data																	

**KEL\_HCE\_IN Bit Descriptions**

Bit	Name	Description
31:8	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.
7:0	Input_Data	<b>Input Data.</b> This register holds data normally read by SSM software. The register value is normally established by a write to I/O Port 060h or 064h. However, the value can also be established by a direct write. Such direct writes have no side effects.

**6.10.2.3 KEL HCE Output (KEL\_HCE\_OUT)**

KEL Memory Offset 108h  
 Type R/W  
 Reset Value 000000xxh

The data placed in the HCE\_Output register by the emulation software is returned when I/O Port 060h is read and emulation is enabled. On a read of this location, the OutputFull bit in HCE\_Status is cleared to 0.

**KEL\_HCE\_OUT Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														Output_Data																	

**KEL\_HCE\_OUT Bit Descriptions**

Bit	Name	Description
31:8	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.
7:0	Output_Data	<b>Output Data.</b> This register holds data normally written by SSM software. It is returned when an I/O read of I/O Port 060h is performed. Writes to this register have no side effects. After writing this register, SSM software normally sets OutputFull in HCE_Status.

#### 6.10.2.4 KEL HCE Status (KEL\_HCE\_STS)

KEL Memory Offset 10Ch  
 Type R/W  
 Reset Value 00000000h

The contents of the HCE\_Status register are returned on an I/O read of I/O Port 064h when emulation is enabled. Reads and writes of I/O Port 060h and writes to I/O Port 064h can cause changes in this register. Emulation software can directly access this register through its memory address in the KEL's operational register space. Accessing this register through its memory address produces no side effects.

**KEL\_HCE\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								Parity	Timeout	AuxOutputFull	InhibitSwitch	CmdData	Flag	InputFull	OutputFull

**KEL\_HCE\_STS Bit Descriptions**

Bit	Name	Description
31:8	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.
7	Parity	<b>Parity.</b> Indicates parity error on keyboard/mouse data. The value of this bit is only changed by a direct write to this register.
6	Timeout	<b>Timeout.</b> Used to indicate a timeout. The value of this bit is only changed by a direct write to this register.
5	AuxOutputFull	<b>Auxiliary Output Full.</b> IRQ12 is asserted whenever this bit is set, OutputFull is set, and IRQEn is set. The value of this bit is not affected by any hardware action, therefore, it can only be changed by a direct write to this register.
4	InhibitSwitch	<b>Inhibit Switch.</b> This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.
3	CmdData	<b>Command Data.</b> The KEL clears this bit on an I/O write to I/O Port 60h and sets it on an I/O write to I/O Port 064h.
2	Flag	<b>Flag.</b> Nominally used as a system flag by software to indicate a warm or cold boot.
1	InputFull	<b>Input Full.</b> Except for the case of a A20 sequence, this bit is set on an I/O write to I/O Port 060h or 064h. While this bit is set and emulation is enabled, an emulation interrupt occurs. This bit can only be cleared by a direct write of 0.
0	OutputFull	<b>Output Full.</b> The KEL clears this bit on a read of I/O Port 060h. If IRQEn is set and AuxOutputFull is clear, then an IRQ1 is generated as long as this bit is set. If IRQEn is set and AuxOutputFull is set, then an IRQ12 is generated as long as this bit is set. If this bit is clear and CharacterPending in HCE_Control is set, an emulation interrupt occurs. This bit can only be set by a direct write of this register.



**6.10.2.5 Port A (KEL\_PORTA)**

I/O Port 092h  
 Type R/W  
 Reset Value 00h

Bit 4 of KELX\_CTL (MSR 5140001Fh[4]) is the Port A Enable bit. It must be set to 1 to enable access to this register; otherwise, Port A accesses are directed to the LPC bus where a Port A register may exist in a SuperI/O device.

:

**KEL\_PORTA Register Map**

7	6	5	4	3	2	1	0
RSVD						A20_MASK	SYSR

**KEL\_PORTA Bit Descriptions**

Bit	Name	Description
7:2	RSVD	<b>Reserved.</b> Writes have no effect; reads return 0.
1	A20_MASK	<p><b>A20 Mask.</b> This bit is necessary for older programs that require Port A Address bit 20 emulation. It requires ASMIs to be enabled in order for software to recognize the changing of this bit, and subsequent proper emulation of the A20 address bit (see PORTA_A20_ASMI_EN bit in DIVIL_GLD_MSR_SMI, MSR 51400002h[6], for ASMI enabling details). Anytime this bit is changed an ASMI is generated.</p> <p>If this bit is 1 and a 0 is written, and ASMI generation is enabled, then proper VSA operation causes a rollover of address from A[19:0] = all 1s, to A[19:0] = all 0s. Address bit A20 is 0.</p> <p>If this bit is 0 and a 1 is written, and ASMI generation is enabled, then no legacy rollover occurs from address [19:0] = all 1s. The next incremental address is A20 = 1 and A[19:0] = all 0.</p>
0	SYSR	<p><b>Legacy System Reset.</b> Writing a 1 to this bit causes a system soft reset (INIT) that will only happen if ASMIs are enabled (see PORTA_INIT_ASMI_EN bit in DIVIL_GLD_MSR_SMI, MSR 51400002h[7], for ASMI enabling details). Writing a 0 to this bit has no effect. This bit always reads as 0.</p>

## 6.11 System Management Bus Register Descriptions

The registers for the System Management Bus (SMB) are divided into two sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- SMB Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the

perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The Native registers (summarized in Table 6-27) are accessed via Base Address Register MSR\_LBAR\_SMB (MSR 5140000Bh) as I/O Offsets. (See Section 6.6.2.4 on page 327 for bit descriptions of the Base Address Register.) The reference column in the summary table points to the page where the register maps and bit descriptions are listed.

**Table 6-27. SMB Native Registers Summary**

SMB I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
00h	R/W	8	SMB Serial Data (SMB_SDA)	00h	Page 371
01h	R/W	8	SMB Status (SMB_STS)	00h	Page 371
02h	R/W	8	SMB Control Status (SMB_CTRL_STS)	10h	Page 373
03h	R/W	8	SMB Control 1 (SMB_CTRL1)	00h	Page 374
04h	R/W	8	SMB Address (SMB_ADDR)	00h	Page 375
05h	R/W	8	SMB Control 2 (SMB_CTRL2)	00h	Page 376
06h	R/W	8	SMB Control 3 (SMB_CTRL3)	00h	Page 376
07h	R/W	8	SMB Reserved Register (SMBRSVD). Writes are "don't care" and reads return undefined value.	xxh	---

## 6.11.1 SMB Native Registers

### 6.11.1.1 SMB Serial Data (SMB\_SDA)

SMB I/O Offset 00h  
 Type R/W  
 Reset Value 00h

**SMB\_SDA Register Map**

7	6	5	4	3	2	1	0
SMBSDA							

**SMB\_SDA Bit Descriptions**

Bit	Name	Description
7:0	SMBSDA	<b>SMB Serial Data.</b> This shift register is used to transmit and receive data. The most significant bit is transmitted (received) first, and the least significant bit is transmitted last. Reading or writing to the SMBSDA register is allowed only when the SDAST bit (SMB I/O Offset 01h[6]) is set, or for repeated starts after setting the START bit (SMB I/O Offset 03h[0]). Any attempt to access the register in other cases may produce unpredictable results.

### 6.11.1.2 SMB Status (SMB\_STS)

SMB I/O Offset 01h  
 Type R/W  
 Reset Value 00h

This is a read/write register with a special clear. Some of its bits may be cleared by software, as described in the table below. This register maintains the current SMB status. On reset, and when the SMB is disabled, SMBST is cleared (00h).

**SMB\_STS Register Map**

7	6	5	4	3	2	1	0
SLVSTP	SDAST	BER	NEGACK	STASTR	NMATCH	MASTER	XMIT

**SMB\_STS Bit Descriptions**

Bit	Name	Description
7	SLVSTP (R/W1C)	<b>Slave Stop (Read/Write 1 to Clear).</b> Writing 0 to SLVSTP is ignored. 0: Writing 1 or SMB disabled. 1: Stop condition detected after a slave transfer in which MATCH (SMB I/O Offset 02h[2]) or GCMATCH (SMB I/O Offset 02h[3]) was set.
6	SDAST (RO)	<b>SMB_DATA Status (Read Only).</b> 0: Reading from SMBSDA (SMB I/O Offset 00h) during a receive, or when writing to it during a transmit. When START (SMB I/O Offset 03h[0]) is set, reading SMBSDA does not clear SDAST; enabling the SMB to send a repeated start in master receive mode. 1: SMBSDA awaiting data (transmit - master or slave) or holds data that should be read (receive - master or slave).

**SMB\_STS Bit Descriptions (Continued)**

Bit	Name	Description
5	BER (R/W1C)	<b>Bus Error (Read/Write 1 to Clear).</b> Writing 0 to BER is ignored. 0: Writing 1 or SMB disabled. 1: Invalid Start or Stop condition detected during data transfer (i.e., Start or Stop condition during the transfer of bits [8:2] and acknowledge cycle), or when an arbitration problem detected. If the SMBus loses an arbitration this bit is set.
4	NEGACK (R/W1C)	<b>Neg Acknowledge (Read/Write 1 to Clear).</b> Writing 0 to NEGACK is ignored. 0: Writing 1 or SMB disabled. 1: Transmission not acknowledged on the ninth clock. (In this case, SDAST (bit 6) is not set.)
3	STASTR (R/W1C)	<b>Stall After Start (Read/Write 1 to Clear).</b> Writing 0 to STASTR is ignored. When STASTR is set, it stalls the SMBus by pulling down the SMB_CLK line, and suspends any further action on the bus (e.g., receive of first byte in master receive mode). 0: Writing 1 or SMB disabled. 1: This bit is not set in the slave mode and is only set in the master transmit mode. When set, this bit indicates that the address was sent successfully and the bus is now stalled. Note that this mode of operation must be enabled with the STASTRE bit (SMB I/O Offset 03h[7]) in order for this bit to function this way. Also, if enabled with INTEN (SMB I/O Offset 03h[2]), an interrupt is sent when this bit is set. This bit is cleared by writing a 1 to it; writing a 0 has no effect.
2	NMATCH (R/W1C)	<b>New Match (Read/Write 1 to Clear).</b> Writing 0 to NMATCH is ignored. If INTEN (SMB I/O Offset 03h[2]) is set, an interrupt is sent when this bit is set. 0: Software writes 1 to this bit. 1: Address byte follows a Start condition or a Repeated Start, causing a match or a global-call match.
1	MASTER (RO)	<b>Master (Read Only).</b> 0: Arbitration loss (BER, bit 5, is set) or recognition of a Stop condition. 1: Bus master request succeeded and master mode active.
0	XMIT (RO)	<b>Transmit (Read Only).</b> Direction bit. 0: Master/slave transmit mode not active. 1: Master/slave transmit mode active.

**6.11.1.3 SMB Control Status (SMB\_CTRL\_STS)**

SMB I/O Offset 02h  
 Type R/W  
 Reset Value 10h

This register configures and controls the SMB functional block. It maintains the current SMB status and controls several SMB functions. On reset and when the SMB is disabled, the non-reserved bits of SMBCST are cleared.

**SMB\_CTRL\_STS Register Map**

7	6	5	4	3	2	1	0
RSVD		TGSCCL	TSDA	GCMTCH	MATCH	BB	BUSY

**SMB\_CTRL\_STS Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
5	TGSCCL	<b>Toggle SMB_CLK Line.</b> Enables toggling the SMB_CLK line during error recovery. 0: Clock toggle completed. 1: When the SMB_DATA line is low, writing 1 to this bit toggles the SMB_CLK line for one cycle. Writing 1 to TGSCCL while SMB_DATA is high is ignored.
4	TSDA (RO)	<b>Test SMB_DATA Line (Read Only).</b> This bit reads the current value of the SMB_DATA line. It can be used while recovering from an error condition in which the SMB_DATA line is constantly pulled low by an out-of-sync slave. Data written to this bit is ignored.
3	GCMTCH (RO)	<b>Global Call Match (Read Only).</b> 0: Start condition or Repeated Start and a Stop condition (including Illegal Start or Stop condition). 1: In slave mode, GCMEN (SMB I/O Offset 03h[5]) is set and the address byte (the first byte transferred after a Start condition) is 00h.
2	MATCH (RO)	<b>Address Match (Read Only).</b> 0: Start condition or Repeated Start and a Stop condition (including Illegal Start or Stop condition). 1: SAEN (SMB I/O Offset 04h[7]) is set and the first 7 bits of the address byte (the first byte transferred after a Start condition) match the 7-bit address in the SMBADR (SMB I/O Offset 04h[6:0]).
1	BB (R/W1C)	<b>Bus Busy (Read/Write 1 to Clear).</b> 0: Writing 1, SMB disabled, or Stop condition detected. 1: Bus active (a low level on either SMB_DATA or SMB_CLK), or Start condition.
0	BUSY (RO)	<b>Busy (Read Only).</b> This bit indicates the SMB is either in slave transmit/receive or master transmit/receive mode, or if there is an arbitration going on the bus. 0: SMB disabled or SMBus in Idle mode 1: SMB is in one of the following states: -Generating a Start condition. -Detects a Start condition. -Master mode (MASTER (SMB I/O Offset 01h[1]) is set). -Slave mode (MATCH (bit 2) or GCMTCH (bit 3) are set).

**6.11.1.4 SMB Control 1 (SMB\_CTRL1)**

SMB I/O Offset 03h  
 Type R/W  
 Reset Value 00h

This register configures and controls the SMB functional block. It maintains the current SMB status and controls several SMB functions. On reset and when the SMB is disabled, the non-reserved bits of SMB\_CTRL1 are cleared.

**SMB\_CTRL1 Register Map**

7	6	5	4	3	2	1	0
STASTRE	NMINTE	GCMEN	ACK	RSVD	INTEN	STOP	START

**SMB\_CTRL1 Bit Descriptions**

Bit	Name	Description
7	STASTRE	<b>Stall After Start Enable.</b> 0: When cleared, STASTR (SMB I/O Offset 01h[3]) can not be set. However, if STASTR is set, clearing STASTRE will not clear STASTR. 1: Stall after start mechanism enabled, and SMB stalls the bus after the address byte.
6	NMINTE	<b>New Match Interrupt Enable.</b> 0: No interrupt issued on a new match. 1: Interrupt issued on a new match only if INTEN (bit 2) is set.
5	GCMEN	<b>Global Call Match Enable.</b> 0: SMB not responding to global call. 1: Global call match enabled.
4	ACK	<b>Receive Acknowledge.</b> This bit is ignored in transmit mode. When the device acts as a receiver (slave or master), this bit holds the transmitting instruction that is transmitted during the next acknowledge cycle. 0: Cleared after acknowledge cycle. 1: Negative acknowledge issued on next received byte.
3	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect.
2	INTEN	<b>Interrupt Enable.</b> 0: SMB interrupt disabled. 1: SMB interrupt enabled. An interrupt is generated in response to one of the following events: -Detection of an address match (NMATCH, SMB I/O Offset 01h[2] = 1) and NMINTE (bit 6) = 1. -Receipt of bus error (BER, SMB I/O Offset 01h[5] = 1). -Receipt of Negative Acknowledge after sending a byte (NEGACK, SMB I/O Offset 01h[4] = 1). -Acknowledge of each transaction (same as the hardware set of the SDAST bit, SMB I/O Offset 01h[6]) when DMA not enabled. -In master mode if STASTRE = 1 (SMB I/O Offset 03h[7]), after a successful start (STASTR = 1, SMB I/O Offset 01h[3]). -Detection of a Stop condition while in slave mode (SLVSTP = 1, SMB I/O Offset 01h[7]).
1	STOP	<b>Stop.</b> 0: Automatically cleared after Stop issued. 1: Setting this bit in master mode generates a Stop condition to complete or abort current message transfer.

**SMB\_CTRL1 Bit Descriptions**

Bit	Name	Description
0	START	<p><b>Start.</b> Set this bit only when in master mode or when requesting master mode.</p> <p>0: Cleared after Start condition sent or Bus Error (BER (SMB I/O Offset 01h[5]) = 1) detected.</p> <p>1: Single or repeated Start condition generated on the SMB. If the device is not the active master of the bus (MASTER (SMB I/O Offset 01h[1]) = 0), setting START generates a Start condition when the SMB becomes free (BB (SMB I/O Offset 02h[1]) = 0). An address transmission sequence should then be performed.</p> <p>If the device is the active master of the bus (MASTER = 1), setting START and then writing to SMBSDA generates a Start condition. If a transmission is already in progress, a repeated Start condition is generated. This condition can be used to switch the direction of the data flow between the master and the slave, or to choose another slave device without separating them with a Stop condition.</p>

**6.11.1.5 SMB Address (SMB\_ADDR)**

SMB I/O Offset 04h  
 Type R/W  
 Reset Value 00h

**SMB\_ADDR Register Map**

7	6	5	4	3	2	1	0
SAEN	SMBADDR						

**SMB\_ADDR Bit Descriptions**

Bit	Name	Description
7	SAEN	<p><b>Slave Enable.</b></p> <p>0: SMB does not check for an address match with address field.</p> <p>1: Address field holds a valid address and enables the match of ADDR to an incoming address byte.</p>
6:0	SMBADDR	<p><b>Device address (SMB Own Address).</b> These bits hold the 7-bit device address. When in slave mode, the first 7 bits received after a Start condition are compared with this field (first bit received is compared with bit 6, and the last bit with bit 0). If the address field matches the received data and SAEN (bit 7) is 1, a match is declared.</p>

**6.11.1.6 SMB Control 2 (SMB\_CTRL2)**

SMB I/O Offset 05h  
 Type R/W  
 Reset Value 00h

This register enables/disables the functional block and determines the SMB clock rate.

**SMB\_CTRL2 Register Map**

7	6	5	4	3	2	1	0
SCLFRQ[6:0]							EN

**SMB\_CTRL2 Bit Descriptions**

Bit	Name	Description
7:1	SCLFRQ[6:0]	<p><b>SMB_CLK Frequency.</b> This field combined with SCLFRQ[14:7] in SMB_CTRL3 defines the SMB_CLK period (low and high time) when the device serves as a bus master. The clock low and high times are defined as follows:</p> $t_{SCLl} = t_{SCLh} = 2 * SCLFRQ * t_{CLK}$ <p>where <math>t_{CLK}</math> is the module input clock cycle.</p> <p>SCLFRQ can be programmed to values in the range of 0008h through 7FFFh. Using any other value has unpredictable results.</p> <p>The low and high time are generally equal unless two or more devices are driving the SCL line.</p>
0	EN	<p><b>Enable.</b></p> <p>0: SMB is disabled, all registers are cleared, and clocks are halted. In the SMBCST register all bits are cleared except the TSDA bit, it reflects the value of SMB_DATA.</p> <p>1: SMB is enabled.</p>

**6.11.1.7 SMB Control 3 (SMB\_CTRL3)**

SMB I/O Offset 06h  
 Type R/W  
 Reset Value 00h

This register enables/disables the functional block and determines the SMB clock rate.

**SMB\_CTRL3 Register Map**

7	6	5	4	3	2	1	0
SCLFRQ[14:7]							

**SMB\_CTRL3 Bit Descriptions**

Bit	Name	Description
7:0	SCLFRQ[14:7]	<p><b>SMB_CLK Frequency.</b> This field combined with SCLFRQ[6:0] in SMB_CTRL2 defines the SMB_CLK period (low and high time) when the device serves as a bus master. See SMB_CTRL2 for use of this register.</p>



## 6.12 UART and IR Port Register Descriptions

The registers for the UART/IR Controller are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- UART/IR Controller Specific MSRs
- UART/IR Controller Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the UART/IR Controller Specific MSRs (summarized in Table 6-28) are called out

as 8 bits. The UART/IR Controller treats writes to the upper 56 bits (i.e., bits [63:8]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The UART/IR Controller Native register set consists of eight register banks, each containing eight registers, to control UART operation. All registers use the same 8-byte address space to indicate I/O Offsets 00h-07h. The Native registers are accessed via Banks 0 through 7 as I/O Offsets. See MSR\_LEG\_IO (MSR 51400014h) bits [22:20] and bits [18:16] for setting base address. Each bank and its offsets are summarized in Table 6-29.

The register summary tables include reset values and page references where the bit descriptions are provided.

**Table 6-28. UART/IR Controller Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400038h	R/W	UART1 Primary Dongle and Modem Interface (UART[1]_MOD)	0xh	Page 380
51400039h	R/W	UART1 Secondary Dongle and Status (UART[1]_DONG)	xxh	Page 381
5140003Ah	R/W	UART1 Interface Configuration (UART[1]_CONF)	42h	Page 382
5140003Bh	R/W	UART1 Reserved MSR (UART[1]_RSVD_MSR) - Reads return 0; writes have no effect.	00h	---
5140003Ch	R/W	UART2 Primary Dongle and Modem Interface (UART[2]_MOD)	0xh	Page 380
5140003Dh	R/W	UART2 Secondary Dongle and Status (UART[2]_DONG)	xxh	Page 381
5140003Eh	R/W	UART2 Interface Configuration (UART[2]_CONF)	42h	Page 382
5140003Fh	R/W	UART2 Reserved MSR (UART[2]_RSVD_MSR) - Reads return 0; writes have no effect.	00h	---

**Table 6-29. UART/IR Controller Native Registers Summary**

I/O Offset	Type	Register Name	Reset Value	Reference
<b>Bank 0</b>				
00h	RO	Receive Data Port (RXD)	xxh	Page 384
	WO	Transmit Data Port (TXD)	xxh	Page 384
01h	R/W	Interrupt Enable Register (IER)	00h	Page 385
02h	RO	Event Identification Register (EIR)	Extended Mode: 22h	Page 386
			Non-Extended Mode: 01h	Page 388
	WO	FIFO Control Register (FCR)	00h	Page 389
03h	WO	Link Control Register (LCR)	00h	Page 390
	R/W	Bank Select Register (BSR)	00h	Page 392
04h	R/W	Modem/Mode Control Register (MCR)	00h	Page 393
05h	RO	Link Status Register (LSR)	60h	Page 394

**Table 6-29. UART/IR Controller Native Registers Summary (Continued)**

I/O Offset	Type	Register Name	Reset Value	Reference
06h	RO	Modem Status Register (MSR)	x0h	Page 396
07h	R/W	Scratchpad Register (SPR)	00h	Page 397
	R/W	Auxiliary Status and Control Register (ASCR)	00h	Page 397
<b>Bank 1</b>				
00h	R/W	Legacy Baud Generator Divisor Low Byte (LBDG_L)	xxh	Page 398
01h	R/W	Legacy Baud Generator Divisor High Byte (LBDG_H)	xxh	Page 398
02h	---	Reserved Register (RSVD)	---	---
03h	R/W	Link Control Register (LCR)	00h	Page 399
	R/W	Bank Selection Encoding Register (BSR)	00h	Page 399
04h-07h	---	Reserved (RSVD)	---	---
<b>Bank 2</b>				
00h	R/W	Baud Generator Divisor Low Byte (BGD_L)	xxh	Page 400
01h	R/W	Baud Generator Divisor High Byte (BGD_H)	xxh	Page 400
02h	R/W	Extended Control Register 1 (EXCR1)	00h	Page 400
03h	R/W	Bank Select Register (BSR)	00h	Page 403
04h	R/W	Extended Control Register 2 (EXCR2)	00h	Page 403
05h	---	Reserved Register (RSVD)	---	---
06h	RO	TX_FIFO Current Level Register (TXFLV)	00h	Page 404
07h	RO	RX_FIFO Current Level Register (RXFLV)	00h	Page 404
<b>Bank 3</b>				
00h	RO	Module Identification and Revision ID Register (MRID)	0xh	Page 405
01h	RO	Shadow of Link Control Register (SH_LCR)	00h	Page 405
02h	RO	Shadow of FIFO Control Register (SH_FCR)	00h	Page 406
03h	R/W	Bank Select Register (BSR)	00h	Page 406
04h-07h	---	Reserved Register (RSVD)	---	---
<b>Bank 4</b>				
00h-01h	---	Reserved Register (RSVD)	---	---
02h	R/W	IR Control Register 1 (IRCR1)	00h	Page 406
03h	R/W	Bank Select Register (BSR)	00h	Page 407
04h-07h	---	Reserved Register (RSVD)	---	---
<b>Bank 5</b>				
00h-02h	---	Reserved Register (RSVD)	---	---
03h	R/W	Bank Select Register (BSR)	00h	Page 407
04h	R/W	IR Control Register 2 (IRCR2)	02h	Page 407
05h-07h	---	Reserved Register (RSVD)	---	---

**Table 6-29. UART/IR Controller Native Registers Summary (Continued)**

I/O Offset	Type	Register Name	Reset Value	Reference
<b>Bank 6</b>				
00h	R/W	IR Control Register 3 (IRCR3)	20h	Page 408
01h	---	Reserved Register (RSVD)	---	---
02h	R/W	SIR Pulse Width Register (SIR_PW)	00h	Page 409
03h	R/W	Bank Select Register (BSR)	00h	Page 409
04h-07h	---	Reserved Register (RSVD)	---	---
<b>Bank 7</b>				
00h	R/W	IR Receiver Demodulator Control Register (IRRXDC)	29h	Page 410
01h	R/W	IR Transmitter Modulator Control Register (IRTXMC)	69h	Page 412
02h	R/W	CEIR Configuration Register (RCCFG)	00h	Page 414
03h	R/W	Bank Select Register (BSR)	00h	Page 415
04h	R/W	IR Interface Configuration Register 1 (IRCFG1)	xxh	Page 415
05h-06h	---	Reserved Register (RSVD)	---	---
07h	R/W	IR Interface Configuration 4 Register (IRCFG4)	00h	Page 416

## 6.12.1 UART/IR Controller Specific MSRs

### 6.12.1.1 UART[x] Primary Dongle and Modem Interface (UART[x]\_MOD)

#### UART1 Primary Dongle and Modem Interface (UART[1]\_MOD)

MSR Address	51400038h
Type	R/W
Reset Value	0xh

#### UART2 Primary Dongle and Modem Interface (UART[2]\_MOD)

MSR Address	5140003Ch
Type	R/W
Reset Value	0xh

MSR\_UART[x]\_MOD is used for primary identification of the dongle interface and also provides the device with a virtual modem interface. To run legacy software, the modem control bits are set such that the software sees the modem always ready for data transfer.

#### UART[x]\_MOD Register Map

7	6	5	4	3	2	1	0
MOD7	MOD6	MOD5	MOD4	ID0	ID1	ID2	ID3

#### UART[x]\_MOD Bit Descriptions

Bit	Name	Description
7	MOD7	<b>Modem 7.</b> This bit directly sets the value of the Ring Indicator (RI) bit of the Modem Status Register (Bank 0, I/O Offset 06h[6]).
6	MOD6	<b>Modem 6.</b> This bit directly sets the value of the Data Set Ready (DSR) bit of the Modem Status Register (Bank 0, I/O Offset 06h[5]).
5	MOD5	<b>Modem 5.</b> This bit directly sets the value of the Data Carrier Detect (DCD) bit of the Modem Status Register (Bank 0, I/O Offset 06h[7]).
4	MOD4	<b>Modem 4.</b> This bit directly sets the value of the Clear to Send (CTS) bit of the Modem Status Register (Bank 0, I/O Offset 06h[4]).
3:0	ID0-ID3	<p><b>Primary Dongle Control Signals.</b> This field selects the type of IR Dongle allowing software to imitate the functionality of a real dongle. These bits are unused in UART mode.</p> <p>x000: Infrared transceiver with serial interface and differential signalling.</p> <p>x100: Reserved.</p> <p>0010: IrDA-data transceiver is: Sharp RY5HD01 or Sharp RY5KD01.</p> <p>1010: Reserved.</p> <p>0110: Infrared transceiver with serial interface and single-ended signalling.</p> <p>1110: Infrared transceiver supports consumer IR modes only.</p> <p>0001: IrDA-data transceiver is: HP HSDL-2300 or HP HSDL-3600</p> <p>1001: IrDA-data transceiver is: IBM 31T1100, Vishay-Telefunken TFDS6000 or Siemens IRMS/T6400.</p> <p>0101: Reserved.</p> <p>1101: IrDA-data transceiver is: Vishay-Telefunken TFDS6500.</p> <p>x011: IrDA-data transceiver is: HP HSDL-1100, HP HSDL-2100, TI TSML 1100 or Sharp RY6FD11E.</p> <p>0111: IrDA-data transceiver supports SIR mode only.</p> <p>1111: No dongle connected.</p>

**6.12.1.2 UART[x] Secondary Dongle and Status (UART[x]\_DONG)**

**UART1 Secondary Dongle and Status (UART[1]\_DONG)**

MSR Address 51400039h  
 Type R/W  
 Reset Value xxh

**UART2 Secondary Dongle and Status (UART[2]\_DONG)**

MSR Address 5140003Dh  
 Type R/W  
 Reset Value xxh

UART[x]\_DONG is used for secondary identification of the dongle interface, and along with UART[x]\_MOD, constitutes a VDI (virtual dongle interface). The transceivers can also be configured for available mode by driving IRSL[2:0]. This support is not provided in the current design, but the values of these signals are written in VDI UART[x]\_MOD in case there is a need to know what legacy software writes on IRSL[2:0] bits (via IRCFG1 at I/O Offset 04h in Bank 7). The MSR is readable by the software so the status of IRSL[2:0] can be known.

**UART[x]\_DONG Register Map**

7	6	5	4	3	2	1	0
RSVD			IRSL2	IRSL1	IRSL0	ID3_SEC	ID0_SEC

**UART[x]\_DONG Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Write as 0.
4:2	IRSL2-IRSL0 (RO)	<b>IRSL[2:0] (Read Only).</b> These bits are reflections of settings in the Transceiver Identification and Control (IRIC) bits (Bank 7, I/O Offset 04h[2:0]). Software may read this field to determine the settings of the virtual IR interface (dongle). See Section 6.12.10.5 "IR Interface Configuration Register 1 (IRCFG1)" on page 415 for decode.
1	ID3_SEC	<b>Secondary Dongle Control Signal.</b> Secondary virtual dongle configuration bit. See Table 6-30. Refer to Section 5.12.3 "Dongle Interface" on page 144 for additional information.
0	ID0_SEC	<b>Secondary Dongle Control Signal.</b> Secondary virtual dongle configuration bit. See Table 6-30. Refer to Section 5.12.3 "Dongle Interface" on page 144 for additional information.

**Table 6-30. Secondary ID Encoding**

IRSL2	IRSL1	ID3	ID0
NCH	INV	NCH: No support INV: 36 kHz demodulation support (RC-5 and RC-6 protocols)	Reserved
INV	NCH	NCH: No support INV: 38 kHz demodulation support (NEC protocol)	NCH: No support INV: 40 kHz demodulation support (JVC, Panasonic protocols)
INV	INV	NCH: No support INV: 56.9 kHz demodulation support (RCA protocol)	Reserved

**Note:** NCH: No Change; INV: Invert.

### 6.12.1.3 UART[x] Interface Configuration (UART[x]\_CONF)

#### UART1 Interface Configuration (UART[1]\_CONF)

MSR Address 5140003Ah  
 Type R/W  
 Reset Value 42h

#### UART2 Interface Configuration (UART[2]\_CONF)

MSR Address 5140003Eh  
 Type R/W  
 Reset Value 42h

UART[x]\_CONF derives the control signals for mode and reset control.

#### UART[x]\_CONF Register Map

7	6	5	4	3	2	1	0
RSVD	BUSY	RESET2SIR	EN_BANKS	TEST	PWDN	DEVEN	MSR_SOFT_RESET

#### UART[x]\_CONF Bit Descriptions

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write as 0.
6	BUSY (RO)	<b>Busy (Read Only).</b> When high, this bit indicates that the UART busy signal is active.
5	RESET2SIR	<b>IR Transmitter and Receiver Reset.</b> Set to 1 to enable reset to the IR transmitter and receiver. (Default = 0.) Write 0 to bring it out of reset.
4	EN_BANKS	<b>Banks Enable.</b> When set to 1, enables access to upper banks (i.e., Banks 2 through 7). (Default = 0; that is, the UART boots in basic mode [16550 mode where access to upper banks is not needed]).
3	TEST	<b>Test.</b> When set to 1, the UART goes into test mode and generates a baud clock on the serial output pin UART[x]_TX (Default = 0.)
2	PWDN	<b>Power-down.</b> When set to 1, the UART clocks (24 MHz clock) are frozen, but interrupt remains enabled. (Default = 0.)
1	DEVEN	<b>Device Enable.</b> Set to 1 to enable the UART. Resetting this bit disables UART functionality and masks the interrupt. (Default = 1.)
0	MSR_SOFT_RESET	<b>MSR Software Reset.</b> Writing a 1 resets the UART. (Default = 0.)

### 6.12.2 UART/IR Controller Native Registers

Eight register banks, each containing eight registers, control UART/IR operation. All registers use the same 8-byte address space to indicate I/O Offsets 00h-07h. The active bank must be selected by the software.

The register bank organization enables access to the banks, as required for activation of all module modes, while maintaining transparent compatibility with 16450 or 16550 software. This activates only the registers and specific bits used in those devices.

The Bank Select Register (BSR) selects the active bank and is common to all banks. Therefore, each bank defines seven new registers. See Figure 6-1.

The default bank selection after the system reset is 0, placing the module in UART 16550 mode. Additionally, setting the baud in Bank 1 (as required to initialize the 16550 UART) switches the module to non-extended UART mode. This ensures that running existing 16550 software switches the system to the 16550 configuration without software modification.

Table 6-31 shows the main functions of the registers in each bank. Banks 0 to 3 control UART and IR modes of operation; Banks 4 to 7 control and configure the IR modes only. Banks 4 to 7 are reserved in UART2.

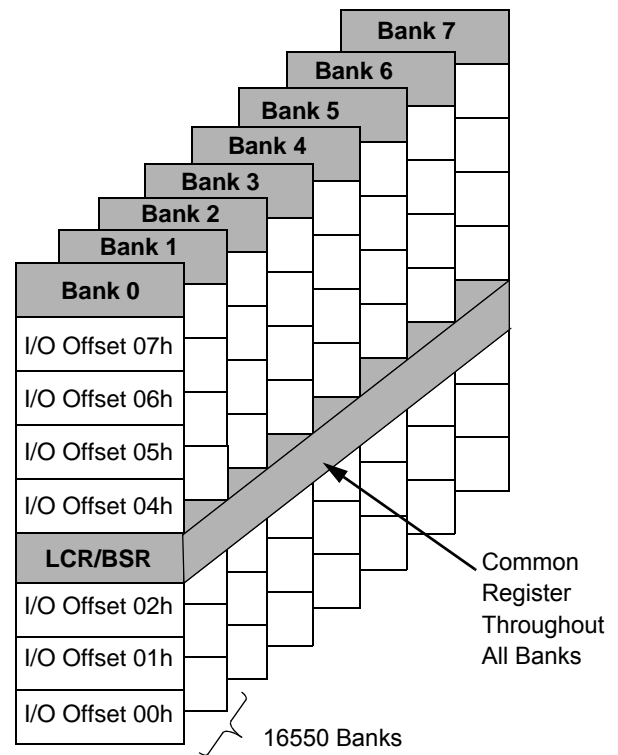


Figure 6-1. UART Register Bank Architecture

Table 6-31. Register Bank Summary

Bank	UART	IR Mode	Main Functions
0	x	x	Global control and status
1	x	x	Legacy bank
2	x	x	Alternative baud generator divisor, extended control, and status
3	x	x	Module revision ID and shadow registers
4		x	IR mode setup
5		x	IR control
6		x	IR physical layer configuration
7		x	CEIR and optical transceiver configuration

### 6.12.3 Bank 0 Register Descriptions

In non-extended modes of operation, Bank 0 is compatible with both the 16450 and 16550. Upon reset, this functional block defaults to the 16450 mode. In extended mode, all the registers (except RXD and TXD) offer additional features. The bit formats for the registers in Bank 0 are summarized in Table 6-32. Detailed descriptions of each register follow.

**Table 6-32. Bank 0 Register Bit Map**

I/O Offset	Name	7	6	5	4	3	2	1	0
00h	RXD	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0
00h	TXD	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
01h	IER (Note 1)	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER (Note 2)	RSVD		TXEMP_IE	DMA_IE	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR (Note 1)	FEN[1:0]		RSVD		RXFT	IPR[1:0]		IPF
	EIR (Note 2)	RSVD		TXEMP_EV	DMA_EV	MS_EV	LS_EV/ TXHLT_EV	TXLDL_EV	RXHDL_IE
	FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR	BKSE	BSR[6:0]						
04h	MCR (Note 1)	RSVD			LOOP	ISEN or DCDLP	RILP	RTS	DTR
	MCR (Note 2)	MDSL[2:0]			IR_PLS	TX_DFR	DMA_EN	RTS	DTR
05h	LSR	ER_INF	TXEMP	TXRDY	BRK	FE	PE	OE	RXDA
06h	MSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
07h	SPR (Note 1)	Scratch Data							
	ASCR (Note 2)	CTE	TXUR	RXACT	RXWDG	RSVD	S_OET	RSVD	RXF_TOUT

Note 1. Non-Extended Mode.

Note 2. Extended Mode.

#### 6.12.3.1 Receive/Transmit Data Ports

##### Receive Data Port (RXD)

I/O Offset      00h  
Type             RO  
Reset Value    xxh

##### Transmit Data Port (TXD)

I/O Offset      00h  
Type             WO  
Reset Value    xxh

The RXD (RO) and TXD (WO) ports share the same address.

RXD is accessed during CPU read cycles. It is used to receive incoming data when the FIFOs are disabled, or from the bottom of the RX\_FIFO when the FIFOs are enabled.

TXD is accessed during CPU write cycles. It is used to write data directly to the transmitter when the FIFOs are disabled, or to the TX\_FIFO when the FIFOs are enabled.

DMA cycles always access the TXD and RXD ports, regardless of the selected bank.



### 6.12.3.2 Interrupt Enable Register (IER)

I/O Offset	01h
Type	R/W
Reset Value	00h

IER controls the enabling of various interrupts. Some interrupts are common to all operating modes of the functional block, while others are mode-specific. Bits [7:4] can be set in extended mode only. They are cleared in non-extended mode. When a bit is set to 1, an interrupt is generated when the corresponding event occurs. In non-extended mode, most events can be identified by reading the LSR and MSR. The receiver high-data-level event can only be identified by reading the EIR register after the corresponding interrupt has been generated. In extended mode, events are identified by event flags in the EIR register.

The bitmap of the IER varies depending on the operating mode of the functional block. The modes can be divided into the two groups and is selected via the EXT\_SL bit in the EXCR1 register (Bank 2, I/O Offset 02h[0]):

- UART, Sharp-IR, SIR and CEIR in extended mode (EXT\_SL = 1)
- UART, Sharp-IR and SIR in non-extended mode (EXT\_SL = 0)

#### IER, Extended Mode: UART, SIR, Sharp-IR and CEIR (EXCR1.EXT\_SL = 1)

IER Extended Mode Register Map

7	6	5	4	3	2	1	0
RSVD		TXEMP_IE	DMA_IE	MS_IE	LS_IE/ TXHLT_IE	TXLDL_IE	RXHDL_IE

IER Extended Mode Bit Descriptions

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Write as 0.
5	TXEMP_IE	<b>Transmitter Empty Interrupt Enable.</b> Setting this bit to 1 enables transmitter empty interrupts (in all modes).
4	DMA_IE	<b>DMA Interrupt Enable.</b> Setting this bit to 1 enables the interrupt on terminal count when the DMA is enabled.
3	MS_IE	<b>Modem Status Interrupt Enable.</b> Setting this bit to 1 enables the interrupts on modem status events.
2	LS_IE/TXHLT_IE	<b>Link Status Interrupt Enable/Transmitter Halted Interrupt Enable.</b> Setting this bit enables link status interrupts and transmitter halted interrupts in CEIR.
1	TXLDL_IE	<b>Transmitter Low-Data-Level Interrupt Enable.</b> Setting this bit to 1 enables interrupts when the TX_FIFO is below the threshold level or the transmitter holding register is empty.
0	RXHDL_IE	<b>Receiver High-Data-Level Interrupt Enable.</b> Setting this bit to 1 enables interrupts when the RXD is full, or the RX_FIFO is equal to or above the RX_FIFO threshold level, or an RX_FIFO timeout occurs.

#### Notes (Extended Mode Only):

- 1) If the interrupt signal drives an edge-sensitive interrupt controller input, it is advisable to disable all interrupts by clearing all the IER bits upon entering the interrupt routine, and re-enable them just before exiting it. This guarantees proper interrupt triggering in the interrupt controller should one or more interrupt events occur during execution of the interrupt routine.
- 2) If an interrupt source must be disabled, the CPU can do so by clearing the corresponding bit of the IER register. However, if an interrupt event occurs just before the corresponding enable bit of the IER register is cleared, a spurious interrupt may be generated. To avoid this problem, clearing of any IER bit should be done during execution of the interrupt service routine. If the interrupt controller is programmed for level-sensitive interrupts, clearing IER bits can be performed outside the interrupt service routine, but with the CPU interrupt disabled.
- 3) If the LSR, MSR, or EIR registers are to be polled, the interrupt sources (identified via self-clearing bits) should have their corresponding IER bits set to 0. This prevents spurious pulses on the interrupt output pin.

**IER, Non-Extended Mode: UART, SIR or Sharp-IR (EXCR1.EXT\_SL = 0)**

Upon reset, the IER supports UART, SIR and Sharp-IR in non-extended modes.

**IER Non-Extended Mode Register Map**

7	6	5	4	3	2	1	0
RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE

**IER Non-Extended Mode Bit Descriptions**

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Write as 0.
3	MS_IE	<b>Modem Status Interrupt Enable.</b> Setting this bit to 1 enables the interrupts on modem status events. (EIR bits [3:0] are 0000.)
2	LS_IE	<b>Link Status Interrupt Enable.</b> Setting this bit to 1 enables interrupts on Link Status events. (EIR bits [3:0] are 0110)
1	TXLDL_IE	<b>Transmitter Low Data Level Interrupt Enable.</b> Setting this bit to 1 enables interrupts on transmitter low data level events. (EIR bits [3:0] are 0010.)
0	RXHDL_IE	<b>Receiver High Data Level Interrupt Enable.</b> Setting this bit to 1 enables interrupts on receiver high data level, or RX_FIFO timeout events. (EIR bits [3:0] are 0100 or 1100.)

**6.12.3.3 Event Identification (EIR)/FIFO Control Registers (FCR)**

The Event Identification Register (EIR) is a read-only register and shares the same address as the write-only FIFO Control Register (FCR).

**Event Identification Register (EIR)**

I/O Offset	02h
Type	RO
Reset Value	Extended Mode: 22h Non-Extended Mode: 01h

The EIR indicates the interrupt source, and operates in two modes, non-extended mode (EXT\_SL of the EXCR1 register = 0), and extended mode (EXT\_SL of the EXCR1 register = 1) (Bank 2, I/O Offset 02h[0]). In non-extended mode (default), this register functions the same as in the 16550 mode.

**EIR, Extended Mode (EXCR1.EXT\_SL = 1)**

In extended mode, each of the previously prioritized and encoded interrupt sources is broken down into individual bits. Each bit in this register acts as an interrupt pending flag, and is set to 1 when the corresponding event has occurred or is pending, regardless of the IER register bit setting. When this register is read, the DMA event (bit 4) is cleared if an 8237 type DMA is used. All other bits are cleared when the corresponding interrupts are acknowledged, by reading the relevant register (e.g., reading the MSR register clears MS\_EV).

**EIR Extended Mode Register Map**

7	6	5	4	3	2	1	0
RSVD		TXEMP_EV	DMA_EV	MS_EV	LS_EV/TXHLT_EV	TXLDL_EV	RXHDL_EV

**EIR Extended Mode Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Read as 0.
5	TXEMP_EV	<b>Transmitter Empty Event.</b> This bit is the same as bit 6 of the LSR (Length Status Register). It is set to 1 when the transmitter is empty. (Default = 1.)
4	DMA_EV	<b>DMA Event.</b> This bit is set to 1 when a DMA terminal count (TC) is activated. It is cleared upon read.
3	MS_EV	<p><b>Modem Status Event.</b></p> <ul style="list-style-type: none"> <li>In UART mode:                             <ul style="list-style-type: none"> <li>This bit is set to 1 when any of the 0 to 3 bits in the MSR is set to 1.</li> </ul> </li> <li>In any IR mode:                             <ul style="list-style-type: none"> <li>The function of this bit depends on the setting of IRMSSL of the IRCR2 register (see Section 6.12.8.2 "IR Control Register 2 (IRCR2)" on page 407). When IRMSSL is 0, the bit functions as a modem status interrupt event; when IRMSSL is set to 1, the bit is forced to 0.</li> </ul> </li> </ul>
2	LS_EV/TXHHLT_EV	<p><b>Link Status Event</b></p> <ul style="list-style-type: none"> <li>In UART, Sharp-IR and SIR:                             <ul style="list-style-type: none"> <li>This bit is set to 1 when a receiver error or break condition is reported. When FIFOs are enabled, the parity error, frame error and break conditions are reported only when the associated character reaches the bottom of the RX_FIFO. An overrun error is reported as soon as it occurs.</li> </ul> </li> </ul> <p><b>Link Status Event or Transmitter Halted Event</b></p> <ul style="list-style-type: none"> <li>In CEIR:                             <ul style="list-style-type: none"> <li>Set to 1 when the receiver is overrun or the transmitter underrun.</li> </ul> </li> </ul> <p><b>Note:</b> A high-speed CPU can service the interrupt generated by the last frame byte reaching the RX_FIFO bottom before that byte is transferred to memory by the DMA controller. This can happen when the CPU interrupt latency is shorter than the RX_FIFO timeout. A DMA request is generated only when the RX_FIFO level reaches the DMA threshold or when a FIFO timeout occurs, in order to minimize the performance degradation due to DMA signal handshake sequences. If the DMA controller must be set up before receiving each frame, the software in the interrupt routine should make sure that the last byte of the frame received has been transferred to memory before reinitializing the DMA controller, otherwise that byte could appear as the first byte of the next frame received.</p>
1	TXLDL_EV	<p><b>Transmitter Low-Data-Level Event.</b> (Default = 1.)</p> <ul style="list-style-type: none"> <li>FIFOs disabled:                             <ul style="list-style-type: none"> <li>Set to 1 when the transmitter holding register is empty.</li> </ul> </li> <li>FIFOs enabled:                             <ul style="list-style-type: none"> <li>Set to 1 when the TX_FIFO level is below the threshold level.</li> </ul> </li> </ul>
0	RXHDL_EV	<p><b>Receiver High-Data-Level Event.</b></p> <ul style="list-style-type: none"> <li>FIFOs disabled:                             <ul style="list-style-type: none"> <li>Set to 1 when a character is in the receiver holding register.</li> </ul> </li> <li>FIFOs enabled:                             <ul style="list-style-type: none"> <li>Set to 1 when the RX_FIFO is equal to or above threshold or an RX_FIFO timeout has occurred.</li> </ul> </li> </ul>

**EIR, Non-Extended Mode (EXCR1.EXT\_SL = 0)**

In non-extended UART mode, the functional block prioritizes interrupts into four levels. The EIR indicates the highest level of interrupt that is pending. See Table 6-33 for the encoding of these interrupts.

**EIR Non-Extended Mode Register Map**

7	6	5	4	3	2	1	0
FEN1	FEN0	RSVD		RXFT	IPR1	IPR0	IPF

**EIR Non-Extended Mode Bit Descriptions**

Bit	Name	Description
7:6	FEN[1:0]	<b>FIFOs Enabled.</b> 00: No FIFO enabled (Default). 11: FIFOs enabled (bit 0 of FCR = 1). 01, 10: Reserved.
5:4	RSVD	<b>Reserved.</b> Write to 0.
3	RXFT	<b>RX_FIFO Timeout.</b> In the 16450 mode, this bit is always 0. In the 16550 mode (FIFOs enabled), this bit is set to 1 when an RX_FIFO read timeout has occurred and the associated interrupt is currently the highest priority pending interrupt.
2:1	IPR[1:0]	<b>Interrupt Priority.</b> When bit 0 (IPF) is 0, these bits indicate the pending interrupt with the highest priority. (Default = 00). See Table 6-33.
0	IPF	<b>Interrupt Pending Flag.</b> 0: Interrupt pending. 1: No interrupt pending (Default).

**Table 6-33. EIR Non-Extended Mode Interrupt Priorities**

EIR Bits[3:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0001	N/A	None	None	N/A
0110	Highest	Link Status	Parity error, framing error, data overrun or break event.	Read Link Status Register (LSR).
0100	Second	Receiver High Data Level Event	Receiver Holding Register (RXD) is full, or RX_FIFO level is equal to or above the threshold.	Reading the RXD or RX_FIFO level drops below threshold.
1100	Second	RX_FIFO Timeout	At least one character is in the RX_FIFO, and no character has been input to or read from the RX_FIFO for four character times.	Reading the RXD port.
0010	Third	Transmitter Low Data Level Event	Transmitter Holding Register or TX_FIFO empty.	Reading the Event Identification Register (EIR) if this interrupt is currently the highest priority pending interrupt, or writing into the TXD port.
0000	Fourth	Modem Status	Any transition on CTS, DSR or DCD or a high-to-low transition on RI.	Reading the Modem Status Register (MSR).

**FIFO Control Register (FCR)**

I/O Offset      02h  
 Type            WO  
 Reset Value    00h

The FIFO Control Register (FCR) is used to enable the FIFOs, clear the FIFOs and set the interrupt threshold levels for the RX\_FIFO and TX\_FIFO. FCR can be read through SH\_FCR in Bank 3, I/O Offset 02h (see Section 6.12.6.3 on page 406).

**FCR Register Map**

7	6	5	4	3	2	1	0
RXFTH		TXFTH		RSVD	TXSR	RXSR	FIFO_EN

**FCR Bit Descriptions**

Bit	Name	Description																		
7:6	RXFTH	<p><b>RX_FIFO Interrupt Threshold Level.</b> These bits select the RX_FIFO interrupt threshold level. An interrupt is generated when the level of data in the RX_FIFO is equal to or above the encoded threshold.</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <th colspan="3" style="text-align: center;">RX_FIFO Interrupt Threshold Level</th> </tr> <tr> <th style="text-align: left;"></th> <th style="text-align: center;">(16-Level FIFO)</th> <th style="text-align: center;">(32-Level FIFO)</th> </tr> <tr> <td>00:</td> <td style="text-align: center;">1 (Default)</td> <td style="text-align: center;">1 (Default)</td> </tr> <tr> <td>01:</td> <td style="text-align: center;">4</td> <td style="text-align: center;">8</td> </tr> <tr> <td>10:</td> <td style="text-align: center;">8</td> <td style="text-align: center;">16</td> </tr> <tr> <td>11:</td> <td style="text-align: center;">14</td> <td style="text-align: center;">26</td> </tr> </table>	RX_FIFO Interrupt Threshold Level				(16-Level FIFO)	(32-Level FIFO)	00:	1 (Default)	1 (Default)	01:	4	8	10:	8	16	11:	14	26
RX_FIFO Interrupt Threshold Level																				
	(16-Level FIFO)	(32-Level FIFO)																		
00:	1 (Default)	1 (Default)																		
01:	4	8																		
10:	8	16																		
11:	14	26																		
5:4	TXFTH[1:0]	<p><b>TX_FIFO Interrupt Threshold Level.</b> In non-extended modes, these bits have no effect. In extended modes, these bits select the TX_FIFO interrupt threshold level. An interrupt is generated when the level of data in the TX_FIFO drops below the encoded threshold.</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <th colspan="3" style="text-align: center;">TX_FIFO Interrupt Threshold Level</th> </tr> <tr> <th style="text-align: left;"></th> <th style="text-align: center;">(16-Level FIFO)</th> <th style="text-align: center;">(32-Level FIFO)</th> </tr> <tr> <td>00:</td> <td style="text-align: center;">1 (Default)</td> <td style="text-align: center;">1 (Default)</td> </tr> <tr> <td>01:</td> <td style="text-align: center;">3</td> <td style="text-align: center;">7</td> </tr> <tr> <td>10:</td> <td style="text-align: center;">9</td> <td style="text-align: center;">17</td> </tr> <tr> <td>11:</td> <td style="text-align: center;">13</td> <td style="text-align: center;">25</td> </tr> </table>	TX_FIFO Interrupt Threshold Level				(16-Level FIFO)	(32-Level FIFO)	00:	1 (Default)	1 (Default)	01:	3	7	10:	9	17	11:	13	25
TX_FIFO Interrupt Threshold Level																				
	(16-Level FIFO)	(32-Level FIFO)																		
00:	1 (Default)	1 (Default)																		
01:	3	7																		
10:	9	17																		
11:	13	25																		
3	RSVD	<b>Reserved.</b> Write to 0.																		
2	TXSR	<b>Transmitter Soft Reset.</b> Writing a 1 to this bit generates a transmitter soft reset that clears the TX_FIFO and the transmitter logic. This bit is automatically cleared by the hardware.																		
1	RXSR	<b>Receiver Soft Reset.</b> Writing a 1 to this bit generates a receiver soft reset that clears the RX_FIFO and the receiver logic. This bit is automatically cleared by the hardware.																		
0	FIFO_EN	<b>FIFO_EN (FIFO Enable).</b> When set to 1, this bit enables both the TX_FIFO and RX_FIFOs. Resetting this bit clears both FIFOs. In CEIR mode, the FIFOs are always enabled and the setting of this bit is ignored.																		

### 6.12.3.4 Link Control/Bank Select Registers

These registers share the same address.

The Link Control Register (LCR) selects the communication format for data transfers in UART, SIR and Sharp-IR modes.

The Bank Select Register (BSR) is used to select the register bank to be accessed next.

Reading the register at this address location returns the content of the BSR. The content of LCR can be read from the Shadow of Link Control register (SH\_LCR) in Bank 3, I/O Offset 01h (see Section 6.12.6.2 on page 405). During a write operation to this register at this address location, setting of Bank Select Enable (BKSE, bit 7) determines the register to be accessed, as follows:

- If bit 7 is 0, both LCR and BSR are written into.
- If bit 7 is 1, only BSR is written into and LCR remains unchanged. This prevents the communication format from being spuriously affected when a bank other than Bank 0 or 1 is accessed.

#### Link Control Register (LCR)

I/O Offset        03h  
Type              WO  
Reset Value      00h

Bits 6 to 0 are only effective in UART, Sharp-IR and SIR modes. They are ignored in CEIR mode.

#### LCR Register Map

7	6	5	4	3	2	1	0
BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	

#### LCR Bit Descriptions

Bit	Name	Description
7	BKSE	<p><b>Bank Select Enable.</b></p> <p>0: Register functions as the Link Control register (LCR). 1: Register functions as the Bank Select register (BSR).</p>
6	SBRK	<p><b>Set Break.</b> Enables or disables a break. During the break, the transmitter can be used as a character timer to accurately establish the break duration. This bit acts only on the transmitter front end and has no effect on the rest of the transmitter logic. When set to 1, the following occurs:</p> <ul style="list-style-type: none"> <li>— If UART mode is selected, the UART[x]_TX pin is forced to a logic 0 state.</li> <li>— If SIR mode is selected, pulses are issued continuously on the UART[x]_IR_TX pin.</li> <li>— If Sharp-IR mode is selected and internal modulation is enabled, pulses are issued continuously on the UART[x]_IR_TX pin.</li> <li>— If Sharp-IR mode is selected and internal modulation is disabled, the UART[x]_IR_TX pin is forced to a logic 1 state.</li> </ul> <p>To avoid transmission of erroneous characters as a result of the break, use the following procedure:</p> <ul style="list-style-type: none"> <li>— Wait for the transmitter to be empty (TXEMP = 1).</li> <li>— Set SBRK to 1.</li> <li>— Wait for the transmitter to be empty, and clear SBRK to restore normal transmission.</li> </ul>
5	STKP	<p><b>Stick Parity.</b> When parity is enabled (PEN is 1), this bit and EPS (bit 4) control the parity bit (see Table 6-34). This bit has no effect when parity is not enabled.</p>
4	EPS	<p><b>Even Parity Select.</b> When parity is enabled (PEN is 1), this bit and STKP (bit 5) control the parity bit (see Table 6-34). This bit has no effect when parity is not enabled.</p>
3	PEN	<p><b>Parity Enable.</b> This bit enables the parity bit. The parity enable bit is used to produce an even or odd number of 1s when the data bits and parity bit are summed, as an error detection device.</p> <p>0: No parity bit is used (Default). 1: A parity bit is generated by the transmitter and checked by the receiver.</p>

**LCR Bit Descriptions (Continued)**

Bit	Name	Description
2	STB	<p><b>Stop Bits.</b> This bit specifies the number of stop bits transmitted with each serial character.</p> <p>0: One stop is bit generated (Default).</p> <p>1: If the data length is set to 5 bits via bits [1:0] (WLS[1:0]), 1.5 stop bits are generated. For 6-, 7- or 8-bit word lengths, 2 stop bits are transmitted. The receiver checks for 1 stop bit only, regardless of the number of stop bits selected.</p>
1:0	WLS[1:0]	<p><b>Character Length Select [1:0].</b> These bits specify the number of data bits in each transmitted or received serial character.</p> <p>00: 5 (Default)</p> <p>01: 6</p> <p>10: 7</p> <p>11: 8</p>

**Table 6-34. Bit Settings for Parity Control**

PEN	EPS	STKP	Selected Parity Bit
0	x	x	None
1	0	0	Odd
1	1	0	Even
1	0	1	Logic 1
1	1	1	Logic 0

**Bank Select Register (BSR)**

I/O Offset      03h  
 Type            R/W  
 Reset Value    00h

BSR selects the next register bank to be accessed. For details on how to access this register, see the description of BKSE (bit 7) of LCR (Section 6.12.3.4 on page 390). The register map and bit descriptions are applicable for all banks.

**BSR Register Map**

7	6	5	4	3	2	1	0
BKSE		BSR[6:0]					

**BSR Bit Descriptions**

Bit	Name	Description
7	BKSE	<b>Bank Select Enable</b> 0: Bank 0 selected. 1: Bits 6 to 0 specify the selected bank (see Table 6-35).
6:0	BSR[6:0]	<b>Bank Select [6:0]</b> . When BKSE (bit 7) is set to 1, these bits select the bank (see Table 6-35).

**Table 6-35. Bank Select Encoding**

BSR Bits								Bank Selected
7	6	5	4	3	2	1	0	
0	x	x	x	x	x	x	x	0
1	0	x	x	x	x	x	x	1
1	1	x	x	x	x	1	x	1
1	1	x	x	x	x	x	1	1
1	1	1	0	0	0	0	0	2
1	1	1	0	0	1	0	0	3
1	1	1	0	1	0	0	0	4
1	1	1	0	1	1	0	0	5
1	1	1	1	0	0	0	0	6
1	1	1	1	0	1	0	0	7
1	1	1	1	1	x	0	0	Reserved
1	1	0	x	x	x	0	0	Reserved



**6.12.3.5 Modem/Mode Control Register (MCR)**

I/O Offset        04h  
 Type             R/W  
 Reset Value     00h

MCR controls the virtual interface with the modem or data communications set in loopback mode, and the device operational mode when the device is in the extended mode. This register function differs for extended and non-extended modes. Modem control pins are not available and are replaced with the virtual interface (except RTS and DTR signals) controlled by software only through MSR\_UART[x]\_MOD (see Section 6.12.1.1 on page 380).

**MCR, Extended Mode (EXCR1.EXT\_SL = 1)**

In extended mode, this register is used to select the operation mode (IrDA, Sharp, etc.) of the device and enable the DMA interface. In these modes, the interrupt output signal is always enabled, and loopback can be enabled by setting bit 4 of EXCR1. Bits 2 to 7 should always be initialized when the operation mode is changed from non-extended to extended.

**MCR Extended Mode Register Map**

7	6	5	4	3	2	1	0
MDSL[2:0]			RSVD	TX_DFR	DMA_EN	RTS	DTR

**MCR Extended Mode Bit Descriptions**

Bit	Name	Description
7:5	MDSL[2:0]	<p><b>Mode Select [2:0].</b> These bits select the operation mode of the functional block when in extended mode. When the mode is changed, the TX_FIFO and RX_FIFOs are flushed, Link Status and Modem Status Interrupts are cleared, and all of the bits in the Auxiliary Status and Control register are cleared.</p> <p>000: UART (Default)                      001: Reserved                      010: Sharp-IR                      011: SIR                      100: Reserved                      101: Reserved                      110: CEIR                      111: Reserved</p>
4	RSVD	<b>Reserved.</b> Write as 0.
3	TX_DFR	<p><b>Transmit Deferral.</b> For a detailed description of the transmit deferral see Section 5.12.1.6 "Transmit Deferral" on page 142. This bit is effective only if the TX_FIFO is enabled (FCR bit 0 = 1).</p> <p>0: No transmit deferral enabled. (Default.)                      1: Transmit deferral enabled.</p>
2	DMA_EN	<p><b>DMA Enable.</b> When set to 1, DMA mode of operation is enabled. When DMA is selected, transmit and/or receive interrupts should be disabled to avoid spurious interrupts. DMA cycles always address the Data Holding registers or FIFOs, regardless of the selected bank.</p> <p>0: DMA mode disabled. (Default.)                      1: DMA mode enabled.</p>
1	RTS	<p><b>Request To Send.</b> When loopback is enabled (bit 4 of EXCR1 = 1, Bank 2, I/O Offset 02h), this bit internally drives both CTS and DCD. Otherwise it is unused.</p>
0	DTR	<p><b>Data Terminal Ready.</b> When loopback is enabled (bit 4 of EXCR1 = 1, Bank 2, I/O Offset 02h), this bit internally drives both DSR and RI. Otherwise it is unused.</p>

**MCR, Non-Extended Mode (EXCR1.EXT\_SL = 0)****MCR Non-Extended Mode Register Map**

7	6	5	4	3	2	1	0
RSVD			LOOP	ISEN or DCCLP	RILP	RTS	DTR

**MCR Non-Extended Mode Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Must written as 0.
4	LOOP	<b>Loopback Enable.</b> This bit accesses the same internal register as LOOP (bit 4) of the EXCR1 register. (See Section 6.12.5.2 on page 400 for more information on Loopback mode). 0: Loopback disabled. (Default.) 1: Loopback enabled.
3	ISEN or DCCLP	<b>Interrupt Signal Enable or DCD Loopback.</b> In normal operation (standard 16450 or 16550) mode, this bit controls the interrupt signal and must be set to 1 in order to enable the interrupt request signal. In loopback mode, this bit internally drives DCD, and the interrupt signal is always enabled.
2	RILP	<b>Ring Indicator in Loopback.</b> When loopback is enabled, this bit internally drives RI. Otherwise, it is unused.
1	RTS	<b>Request To Send.</b> When loopback is enabled, this bit drives CTS internally. Otherwise, it is unused.
0	DTR	<b>Data Terminal Ready.</b> When loopback is enabled, this bit internally drives DSR. Otherwise, it is unused.

**6.12.3.6 Link Status Register (LSR)**

I/O Offset	05h
Type	RO
Reset Value	60h

LSR provides status information concerning data transfer. Upon reset, this register assumes the value of 60h. The bit definitions change depending upon the operation mode of the functional block.

Bits 1 to 4 of the LSR indicate link status events. These bits are sticky (accumulate any conditions occurred since the last time the register was read). They are cleared when one of the following events occur:

- Hardware reset
- Receiver soft reset (via the FIFO Control register)
- LSR register read

The LSR is intended for read operations only. Writing to the LSR is not permitted.

**LSR Register Map**

7	6	5	4	3	2	1	0
ER_INF	TXEMP	TXRDY	BRK	FE	PE	OE	RXDA

## LSR Bit Descriptions

Bit	Name	Description
7	ER_INF	<b>Error in RX_FIFO.</b> In UART, Sharp-IR, and SIR modes, this bit is set to 1 if there is at least one framing error, parity error, or break indication in the RX_FIFO. This bit is always 0 in 16450 mode. It is cleared upon read or upon reset if there is no faulted byte in RX_FIFO.
6	TXEMP	<b>Transmitter Empty.</b> This bit is set to 1 when the transmitter holding register or the TX_FIFO is empty, and the transmitter front end is idle.
5	TXRDY	<b>Transmitter Ready.</b> This bit is set to 1 when the transmitter holding register or the TX_FIFO is empty. It is cleared when a data character is written to the TXD register.
4	BRK	<b>Break Event Detected.</b> In UART, Sharp-IR, and SIR modes, this bit is set to 1 when a break event is detected (i.e., when a sequence of logic 0 bits, equal or longer than a full character transmission, is received). If the FIFOs are enabled, the break condition is associated with the particular character in the RX_FIFO to which it applies. In this case, the BRK bit is set when the character reaches the bottom of the RX_FIFO. When a break event occurs, only one 0 character is transferred to the receiver holding register or the RX_FIFO. The next character transfer takes place after at least one logic 1 bit is received, followed by a valid start bit. This bit is cleared upon read.
3	FE	<b>Framing Error.</b> In UART, Sharp-IR and SIR modes, this bit is set to 1 when the received data character does not have a valid stop bit (i.e., the stop bit following the last data bit or parity bit is a 0). If the FIFOs are enabled, this Framing Error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the bottom of the RX_FIFO. After a framing error is detected, the receiver will try to resynchronize. The receiver assumes that framing error (the erroneous stop bit) is the next start bit (the erroneous stop bit) and shifts in the new character. This bit is cleared upon a read.
2	PE	<b>Parity Error.</b> In UART, Sharp-IR, and SIR modes, this bit is set to 1 if the received data character does not have the correct parity, even or odd, as selected by the parity control bits of the LCR. If the FIFOs are enabled, this error is associated with the particular character in the FIFO that it applies to. This error is revealed to the CPU when its associated character is at the bottom of the RX_FIFO. This bit is cleared upon read.
1	OE	<b>Overrun Error.</b> In UART, Sharp-IR, and SIR modes, this bit is set to 1 as soon as an overrun condition is detected by the receiver. It is cleared upon read.  FIFOs Disabled: An overrun occurs when a new character is completely received into the receiver front end section and the CPU has not yet read the previous character in the receiver holding register. The new character overwrites the previous character in the receiver holding register.  FIFOs Enabled: An overrun occurs when a new character is completely received into the receiver front end section and the RX_FIFO is full. The new character is discarded, and the RX_FIFO is not affected.
0	RXDA	<b>Receiver Data Available.</b> This bit is set to 1 when the receiver holding register is full. If the FIFOs are enabled, this bit is set when at least one character is in the RX_FIFO. It is cleared when the CPU reads all the data in the holding register or the RX_FIFO.

### 6.12.3.7 Modem Status Register (MSR)

I/O Offset	06h
Type	RO
Reset Value	x0h

The function of this register depends on the selected operational mode. When a UART mode is selected, this register provides the current state as well as state-change information of the status lines from the modem or data transmission module.

When any of the IR modes is selected, the register function is controlled by the setting of the IRMSSL bit of the IRCR2 (see Section 6.12.8.2 on page 407). If IRMSSL is 0, the MSR register works as in UART mode. If IRMSSL is 1, the MSR returns the value 30h, regardless of the state of the modem input lines.

When loopback is enabled, the MSR works similarly except that its status input signals are internally driven by appropriate bits in the MCR since the modem input lines are internally disconnected. Refer to bits 1 to 0 in MCR extended mode or bits 3 to 0 in MCR non-extended mode (see Section 6.12.3.5 on page 393) and to the LOOP and ETDLBK bits at the EXCR1 (see Section 6.12.5.2 on page 400).

A Modem Status Event (MS\_EV) is generated if the MS\_IE bit in IER is enabled and any of bits 0, 1, 2, or 3 in this register are set to 1.

Bits 0 to 3 are cleared to 0 as a result of any of the following events:

- A hardware reset occurs.
- The operational mode is changed and the IRMSSL bit is 0.
  - The MSR register is read.

In the reset state, bits 4 to 7 are indeterminate as they reflect their corresponding signal levels at MSR\_UART[x]\_MOD.

#### MSR (Modem Status Register) Register Map

7	6	5	4	3	2	1	0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

#### MSR (Modem Status Register) Bit Descriptions

Bit	Name	Description
7	DCD	<b>Data Carrier Detect.</b> This is the status of the MOD5 bit in MSR_UART[x]_MOD (see Section 6.12.1.1 on page 380).
6	RI	<b>Ring Indicator.</b> This is the status of the MOD7 bit in MSR_UART[x]_MOD (see Section 6.12.1.1 on page 380).
5	DSR	<b>Data Set Ready.</b> This is the status of the MOD6 bit in MSR_UART[x]_MOD (see Section 6.12.1.1 on page 380).
4	CTS	<b>Clear To Send.</b> This is the status of the MOD4 bit in MSR_UART[x]_MOD (see Section 6.12.1.1 on page 380).
3	DDCD	<b>Delta Data Carrier Detect.</b> This bit is the Delta Data Carrier Detect (DDCD) indicator. When high, this bit indicates that the DCD input has changed state. Reading this register causes this bit to be cleared.
2	TERI	<b>Trailing Edge Ring Indicator.</b> This bit is the Trailing Edge Ring Indicator (TERI) detector. When high, this bit indicates that the RI input has changed from a high to low state. Reading this register causes this bit to be cleared.
1	DDSR	<b>Delta Data Set Ready.</b> This bit is the Delta Data Set Ready (DDSR) indicator. When high, this bit indicates that the DSR input has changed state since the last time it was read by the CPU. Reading this register causes this bit to be cleared.
0	DCTS	<b>Delta Clear To Send.</b> This bit is the Delta Clear To Send (DCTS) indicator. When high, this bit indicates that the CTS input has changed state since the last time it was read by the CPU. Reading this register causes this bit to be cleared.

### 6.12.3.8 Scratchpad/Auxiliary Status and Control Registers

The Scratchpad Register (SPR) and Auxiliary Status and Control Register (ASCR) share the same address.

#### Scratchpad Register (SPR)

I/O Offset	07h
Type	R/W
Reset Value	00h

This register is accessed when the device is in non-extended mode (EXCR1.EXT\_SL = 0). This is a scratchpad register for temporary data storage.

#### Auxiliary Status and Control Register (ASCR)

I/O Offset	07h
Type	R/W
Reset Value	00h

ASCR is accessed when the extended mode (EXCR1.EXT\_SL = 1) of operation is selected. The definition of the bits in this case is dependent upon the mode selected in the MCR, bits 7 to 5. This register is cleared upon hardware reset. Bit 2 is also cleared when the transmitter is “soft reset” (via the FIFO Control register) or after the S\_EOT byte is transmitted. Bit 6 is also cleared when the transmitter is “soft reset” or by writing 1 into it. Bits 0, 1, 4, and 5 are also cleared when the receiver is “soft reset” (via the FIFO Control register).

#### ASCR Extended Mode Register Map

7	6	5	4	3	2	1	0
RSVD	TXUR	RXACT	RXWDG	RSVD	S_EOT	RSVD	RXF_TOUT

#### ASCR Extended Mode Bit Descriptions

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write as 0.
6	TXUR	<b>IR Transmitter Underrun.</b> For CEIR mode only. This bit is set to 1 when a transmitter underrun occurs. It is always cleared when a mode other than CEIR is selected. This bit must be cleared, by writing a 1 into it to re-enable transmission.
5	RXACT	<b>Receiver Active.</b> For CEIR mode only. This bit is set to 1 when an IR pulse or pulse-train is received. If a 1 is written into this bit position, the bit is cleared and the receiver deactivated. When this bit is set, the receiver samples the IR input continuously at the programmed baud and transfers the data to the RX_FIFO.
4	RXWDG	<b>Reception WATCHDOG.</b> For CEIR mode only. This bit is set to 1 each time a pulse or pulse-train (modulated pulse) is detected by the receiver. It can be used by the software to detect a receiver idle condition. It is cleared upon read.
3	RSVD	<b>Reserved.</b> Write as 0.
2	S_EOT	<b>Set End of Transmission.</b> For CEIR mode only. When a 1 is written into this bit position before writing the last character into the TX_FIFO, data transmission is properly completed. If the CPU simply stops writing data into the TX_FIFO at the end of the data stream, a transmitter underrun is generated and the transmitter stops. In this case this is not an error, but the software must clear the underrun before the next transmission can occur. This bit is automatically cleared by hardware when a character is written to the TX_FIFO.
1	RSVD	<b>Reserved.</b> Write as 0.
0	RXF_TOUT (RO)	<b>RX_FIFO Timeout (Read Only).</b> This bit is set to 1 if the RX_FIFO is below threshold and an RX_FIFO timeout occurs. It is cleared when a character is read from the RX_FIFO.

### 6.12.4 Bank 1 Register Descriptions

The bit formats of the registers in Bank 1 are summarized in Table 6-36. Detailed descriptions of each register follow.

**Table 6-36. Bank 1 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h	LBGD_L	LBGD[7:0]							
01h	LBGD_H	LBGD[15:8]							
02h	RSVD	RSVD							
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS1	WLS0
	BSR	BKSE	BSR[6:0]						
04-07h	RSVD	RSVD							

#### 6.12.4.1 Legacy Baud Generator Divisor Port

##### Legacy Baud Generator Divisor Low Byte (LBGD\_L)

I/O Offset      00h  
 Type            R/W  
 Reset Value    xxh

##### Legacy Baud Generator Divisor High Byte (LBGD\_H)

I/O Offset      01h  
 Type            R/W  
 Reset Value    xxh

The Legacy Baud Generator Divisor (LBGD) port provides an alternate data path to the Baud Divisor Generator register. LBGD is a 16-bit wide port split into two bytes, LBGD\_L and LBGD\_H, occupying consecutive address locations. This port is implemented to maintain compatibility with 16550 standard and to support existing legacy software packages. New software should use the BGD port in Bank 2 to access the baud generator divisor register.

The programmable baud rates in the non-extended mode are achieved by dividing a 24 MHz clock by a prescale value of 13, 1.625 or 1. This prescale value is selected by the PRESL field of EXCR2 (Section 6.12.5.4 on page 403).

Divisor values between 1 and  $2^{16}-1$  can be used (0 cannot be used, see Table 6-38 "Baud Generator Divisor Settings" on page 399). The baud generator divisor must be loaded during initialization to ensure proper operation of the baud generator. Upon loading either part of it, the baud generator counter is immediately loaded. Table 6-38 shows typical baud divisors. After reset, the divisor register contents are indeterminate.

If the UART is in extended mode, any access to the LBGD\_L or LBGD\_H causes a reset to the default non-extended mode (i.e., 16550 mode). To access a baud generator divisor when in extended mode, use the port pair in Bank 2 (see Section 6.12.5.1 "Baud Generator Divisor Port" on page 400).

Table 6-37 shows the bits that are cleared when fallback occurs during extended or non-extended modes. If the UART is in non-extended mode and the LOCK bit is 1, the content of the divisor (BGD) port is not be effected and no other action is taken.

**Table 6-37. Bits Cleared on Fallback**

Register Name	UART Mode and LOCK Bit before Fallback		
	Extended Mode LOCK = x	Non-Extended Mode LOCK = 0	Non-Extended Mode LOCK = 1
MCR	2 to 7	None	None
EXCR1	0, 5 and 7	5 and 7	None
EXCR2	0 to 5	0 to 5	None
IRCR1	2 and 3	None	None

**Table 6-38. Baud Generator Divisor Settings**

Prescaler Value	13		1.625		1	
	Baud	Divisor	% Error	Divisor	% Error	Divisor
50	2304	0.16%	18461	0.00%	30000	0.00%
75	1536	0.16%	12307	0.01%	20000	0.00%
110	1047	0.19%	8391	0.01%	13636	0.00%
134.5	857	0.10%	6863	0.00%	11150	0.02%
150	768	0.16%	6153	0.01%	10000	0.00%
300	384	0.16%	3076	0.03%	5000	0.00%
600	192	0.16%	1538	0.03%	2500	0.00%
1200	96	0.16%	769	0.03%	1250	0.00%
1800	64	0.16%	512	0.16%	833	0.04%
2000	58	0.53%	461	0.12%	750	0.00%
2400	48	0.16%	384	0.16%	625	0.00%
3600	32	0.16%	256	0.16%	416	0.16%
4800	24	0.16%	192	0.16%	312	0.16%
7200	16	0.16%	128	0.16%	208	0.16%
9600	12	0.16%	96	0.16%	156	0.16%
14400	8	0.16%	64	0.16%	104	0.16%
19200	6	0.16%	48	0.16%	78	0.16%
28800	4	0.16%	32	0.16%	52	0.16%
38400	3	0.16%	24	0.16%	39	0.16%
57600	2	0.16%	16	0.16%	26	0.16%
115200	1	0.16%	8	0.16%	13	0.16%
230400	---	---	4	0.16%	---	---
460800	---	---	2	0.16%	---	---
750000	---	---	---	---	2	0.00%
921600	---	---	1	0.16%	---	---
1500000	---	---	---	---	1	0.00%

**6.12.4.2 Link Control/Bank Select Register (LCR/BSR)**

These registers share the same address and are the same as the registers at I/O Offset 03h in Bank 0.

**Link Control Register (LCR)**

I/O Offset      03h  
 Type            RW  
 Reset Value    00h

See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

**Bank Selection Encoding Register (BSR)**

I/O Offset      03h  
 Type            R/W  
 Reset Value    00h

See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

### 6.12.5 Bank 2 Register Descriptions

The bit formats for the registers in Bank 2 are summarized in Table 6-39. Detailed descriptions of each register follow.

**Table 6-39. Bank 2 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h	BGD_L	BGD[7:0]							
01h	BGD_H	BGD[15:8]							
02h	EXCR1	RSVD		EDTLBK	LOOP	DMASWP	DMATH	DMANF	EXT_SL
03h	BSR	BKSE	BSR[6:0]						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RF_SIZ[1:0]		TF_SIZ1[1:0]	
05h	RSVD	RSVD							
06h	TXFLV	RSVD			TFL[5:0]				
07h	RXFLV	RSVD			RFL[5:0]				

#### 6.12.5.1 Baud Generator Divisor Port

##### Baud Generator Divisor Low Byte (BGD\_L)

I/O Offset      00h  
 Type            R/W  
 Reset Value    xxh

##### Baud Generator Divisor High Byte (BGD\_H)

I/O Offset      01h  
 Type            R/W  
 Reset Value    xxh

This port performs the same function as the Legacy Baud Divisor port in Bank 1 and is accessed identically, but does not change the operation mode of the functional block when accessed. See Section 6.12.4.1 "Legacy Baud Generator Divisor Port" on page 398 for more details.

Use this port to set the baud when operating in extended mode to avoid fallback to a non-extended operation mode (i.e., 16550 compatible). When programming the baud, writing to BGD\_H causes the baud to change immediately.

#### 6.12.5.2 Extended Control Register 1 (EXCR1)

I/O Offset      02h  
 Type            R/W  
 Reset Value    00h

Use this register to control operation in the extended mode. Upon reset, all bits are set to 0.

**EXCR1 Register Map**

7	6	5	4	3	2	1	0
RSVD		EDTLBK	LOOP	DMASWP	DMATH	DMANF	EXT_SL

**EXCR1 Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Write as 0.
5	EDTLBK	<b>Enable Transmitter During Loopback.</b> When this bit is set to 1, the transmitter serial output is enabled and functions normally when loopback is enabled.

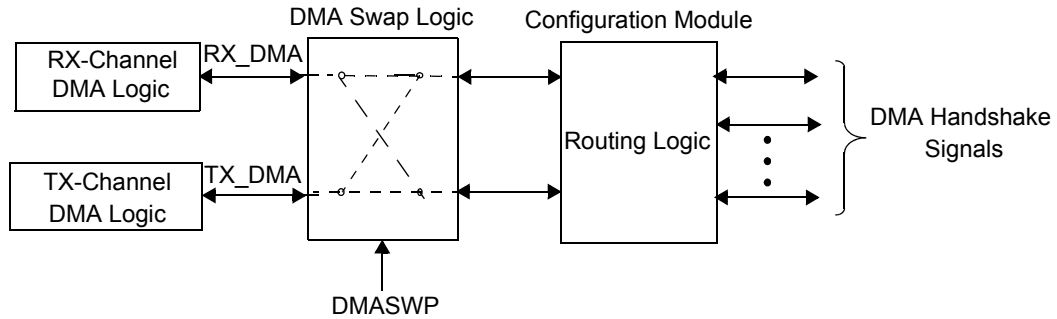


**EXCR1 Bit Descriptions (Continued)**

Bit	Name	Description
4	LOOP	<p><b>Loopback Enable.</b> During loopback, the transmitter output is connected internally to the receiver input, to enable system self-test of serial communication. In addition to the data signal, all additional signals within the UART are interconnected to enable real transmission and reception using the UART mechanisms. When this bit is set to 1, loopback is selected. This bit accesses the same internal register as bit 4 in the MCR, when the UART is in a non-extended mode. Loopback behaves similarly in both non-extended and extended modes. When extended mode is selected, the DTR bit of the MCR internally drives both DSR and RI, and the RTS bit drives CTS and DCD.</p> <p>During loopback, the following occurs:</p> <ul style="list-style-type: none"> <li>• The transmitter and receiver interrupts are fully operational. The modem status interrupts are also fully operational, but the interrupt sources are now the lower bits of the MCR. Modem interrupts in IR modes are disabled unless the IRMSSL bit of the IRCR2 is 0. Individual interrupts are still controlled by the IER bits.</li> <li>• The DMA control signals are fully operational.</li> <li>• UART and IR receiver serial input signals are disconnected. The internal receiver input signals are connected to the corresponding internal transmitter output signals.</li> <li>• The UART transmitter serial output is forced high and the IR transmitter serial output is forced low, unless the ETDLBK bit is set to 1, in which case they function normally.</li> <li>• The virtual modem signals of MSR_UART[x]_MOD register (DSR, CTS, RI and DCD) are disconnected. The internal modem status signals are driven by the lower bits of the MCR.</li> </ul>
3	DMASWP	<p><b>DMA Swap.</b> This bit selects the routing of the DMA control signals between the internal DMA logic and configuration module of the chip. When this bit is 0, the transmitter and receiver DMA control signals are not swapped. When it is 1, they are swapped. A block diagram illustrating the control signals routing is shown in Figure 6-2 "DMA Control Signals Routing" on page 402. The swap feature is particularly useful when only one 8237 DMA channel is used to serve both transmitter and receiver. In this case, only one external DMA Request/DMA Acknowledge pair is interconnected to the swap logic by the configuration module. Routing the external DMA channel to either the transmitter or receiver DMA logic is then controlled by the DMASWP bit. This way, the IR device drivers do not need to know the details of the configuration module.</p>
2	DMATH	<p><b>DMA FIFO Threshold.</b> This bit selects the TX_FIFO and RX_FIFO threshold levels used by the DMA request logic to support demand transfer mode. A transmission DMA request is generated when the TX_FIFO level is below the threshold. A reception DMA request is generated when the RX_FIFO level reaches the threshold or when an RX_FIFO timeout occurs. Table 6-40 lists the threshold levels for each FIFO.</p>
1	DMANF	<p><b>DMA Fairness Control.</b> This bit controls the maximum duration of DMA burst transfers.</p> <p>0: DMA requests forced inactive after approximately 10.5 <math>\mu</math>s of continuous transmitter and/or receiver DMA operation (default).</p> <p>1: TX-DMA request is deactivated when the TX_FIFO is full. An RX DMA request is deactivated when the RX_FIFO is empty.</p>
0	EXT_SL	<p><b>Extended Mode Select.</b> When set to 1, extended mode is selected.</p>

**Table 6-40. DMA Threshold Levels**

Bit Value	DMA Threshold for FIFO Type		
	RX_FIFO	TX_FIFO	
		16 Levels	32 Levels
0	4	13	29
1	10	7	23



**Figure 6-2. DMA Control Signals Routing**

**6.12.5.3 Bank Select Register (BSR)**

I/O Offset 03h  
 Type R/W  
 Reset Value 00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

**6.12.5.4 Extended Control Register 2 (EXCR2)**

I/O Offset 04h  
 Type R/W  
 Reset Value 00h

EXCR2 configures the RX\_FIFO and TX\_FIFO sizes and the value of the prescaler, and controls the baud divisor register lock. Upon reset, all bits are set to 0.

**EXCR2 Register Map**

7	6	5	4	3	2	1	0
LOCK	RSVD	PRESL[1:0]		RF_SIZ[1:0]		TF_SIZ[1:0]	

**EXCR2 Bit Descriptions**

Bit	Name	Description
7	LOCK	<b>Baud Divisor Register Lock.</b> When set to 1, any access to the baud generator divisor register through LBGD_L and LBGD_H, as well as fallback are disabled from non-extended mode. In this case, two scratchpad registers overlaid with LBGD_L and LBGD_H are enabled, and any attempted CPU access of the baud generator divisor register through LBGD_L and LBGD_H access the scratchpad registers instead. This bit must be set to 0 when extended mode is selected.
6	RSVD	<b>Reserved.</b> Write as 0.
5:4	PRESL[1:0]	<b>Prescaler Select.</b> The prescaler divides the 24 MHz input clock frequency to provide the clock for the baud generator.  00: 13 (Default) 01: 1.625 10: Reserved 11: 1.0
3:2	RF_SIZ[1:0]	<b>RX_FIFO Levels Select.</b> These bits select the number of levels for the RX_FIFO. They are effective only when the FIFOs are enabled. (FCR bit 0 = 1.)  00: 16 (Default) 01: 32 1x: Reserved
1:0	TF_SIZ[1:0]	<b>TX_FIFO Levels Select.</b> These bits select the number of levels for the TX_FIFO. They are effective only when the FIFOs are enabled. (FCR bit 0 = 1.)  00: 16 (Default) 01: 32 1x: Reserved

**6.12.5.5 TX\_FIFO Current Level Register (TXFLV)**

I/O Offset        06h  
 Type             RO  
 Reset Value     00h

TXFLV returns the number of bytes in the TX\_FIFO.

**TXFLV Register Map**

7	6	5	4	3	2	1	0
RSVD		TFL[5:0]					

**TXFLV Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Returns 0.
5:0	TFL[5:0]	<b>Number of Bytes in TX_FIFO.</b> These bits specify the number of bytes in the TX_FIFO.  <b>Note:</b> The contents of TXFLV and RXFLV are not frozen during CPU reads. Therefore, invalid data could be returned if the CPU reads these registers during normal transmitter and receiver operation. To obtain correct data, the software should perform three consecutive reads and then take the data from the second read if the first and second reads yield the same result. It can also be taken from the third read if the first and second reads yield different results.

**6.12.5.6 RX\_FIFO Current Level Register (RXFLV)**

I/O Offset        07h  
 Type             RO  
 Reset Value     00h

RXFLV returns the number of bytes in the RX\_FIFO. It can be used for software debugging.

**RXFLV Register Map**

7	6	5	4	3	2	1	0
RSVD		RFL[5:0]					

**RXFLV Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Return 0s.
5:0	RFL[5:0]	<b>Number of Bytes in RX_FIFO.</b> These bits specify the number of bytes in the RX_FIFO.  <b>Note:</b> The contents of TXFLV and RXFLV are not frozen during CPU reads. Therefore, invalid data could be returned if the CPU reads these registers during normal transmitter and receiver operation. To obtain correct data, the software should perform three consecutive reads and then take the data from the second read if the first and second reads yield the same result. It can also be taken from the third read if the first and second reads yield different results.

### 6.12.6 Bank 3 Register Descriptions

The bit formats for the registers in Bank 3 are summarized in Table 6-41. Detailed descriptions of each register follow.

**Table 6-41. Bank 3 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h	MRID	MID[3:0]				RID[3:0]			
01h	SH_LCR	RSVD	SBRK	STKP	EPS	PEN	STB	WLS1	WLS0
02h	SH_FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							

#### 6.12.6.1 Module Identification and Revision ID Register (MRID)

I/O Offset        00h  
 Type            RO  
 Reset Value     0xh

MRID identifies the revision of the module. When read, it returns the module ID and revision level in the format 0xh, where x indicates the revision number.

**MRID Register Map**

7	6	5	4	3	2	1	0
MID[3:0]				RID[3:0]			

**MRID Bit Descriptions**

Bit	Name	Description
7:4	MID[3:0]	<b>Module ID.</b> Identifies the module type.
3:0	RID[3:0]	<b>Revision ID.</b> Identifies the module revision level. For example, 0h = revision 0, 1h = revision 1, etc.

#### 6.12.6.2 Shadow of Link Control Register (SH\_LCR)

I/O Offset        01h  
 Type            RO  
 Reset Value     00h

SH\_LCR returns the value of the LCR. The LCR is written into when a byte value, with bit 7 set to 0, is written to the LCR/BSR registers location (at I/O Offset 03h) from any bank.

**SH\_LCR Register Map**

7	6	5	4	3	2	1	0
RSVD	SBRK	STKP	EPS	PEN	STB	WLS1	WLS0

### 6.12.6.3 Shadow of FIFO Control Register (SH\_FCR)

I/O Offset	02h
Type	RO
Reset Value	00h

This register returns the contents of the FCR in Bank 0.

**FCR Register Map**

7	6	5	4	3	2	1	0
RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN

### 6.12.6.4 Bank Select Register (BSR)

I/O Offset	03h
Type	R/W
Reset Value	00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

### 6.12.7 Bank 4 Register Descriptions

The bit formats for the registers in Bank 4 are summarized in Table 6-42. Detailed descriptions of each register follow.

**Table 6-42. Bank 4 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h-01h	RSVD	RSVD							
02h	IRCR1	RSVD				IR_SL[1:0]		RSVD	
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							

#### 6.12.7.1 IR Control Register 1 (IRCR1)

I/O Offset	02h
Type	R/W
Reset Value	00h

IRCR1 enables the Sharp-IR or SIR IR mode in non-extended mode of operation. Upon reset, all bits are set to 0.

**IRCR1 Register Map**

7	6	5	4	3	2	1	0
RSVD				IR_SL[1:0]		RSVD	

**IRCR1 Bit Descriptions**

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Write as 0.
3:2	IR_SL[1:0]	<b>Sharp-IR or SIR Mode Select.</b> These bits enable Sharp-IR and SIR modes in non-extended mode. They allow selection of the appropriate IR interface when extended mode is not selected. These bits are ignored when extended mode is selected. 00: UART (Default). 01: Reserved. 10: Sharp-IR. 11: SIR.
1:0	RSVD	<b>Reserved.</b> Write as 0.

**6.12.7.2 Bank Select Register (BSR)**

I/O Offset 03h  
 Type R/W  
 Reset Value 00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

**6.12.8 Bank 5 Register Descriptions**

The bit formats for the registers in Bank 5 are summarized in Table 6-43. Detailed descriptions of each register follow.

**Table 6-43. Bank 5 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h-02h	RSVD	RSVD							
03h	BSR	BKSE	BSR[6:0]						
04h	IRCR2	RSVD	RSVD	RSVD	AUX_IRRX	RSVD	RSVD	IRMSSL	IR_FDPLX
05h-07h	RSVD	RSVD							

**6.12.8.1 Bank Select Register (BSR)**

I/O Offset 03h  
 Type R/W  
 Reset Value 00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

**6.12.8.2 IR Control Register 2 (IRCR2)**

I/O Offset 04h  
 Type R/W  
 Reset Value 02h

IRCR2 controls the basic settings of the IR modes. Upon reset, the content of this register is 02h.

**IRCR2 Register Map**

7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	AUX_IRRX	RSVD	RSVD	IRMSSL	IR_FDPLX

**IRCR2 Bit Descriptions**

Bit	Name	Description
7:5	RSVD	<b>Reserved.</b> Write to 0.
4	AUX_IRRX	<b>Auxiliary IR Input Select.</b> When set to 1, the IR signal is received from the auxiliary input. See Table 6-51 "IR Receive Input Selection" on page 416.
3:2	RSVD	<b>Reserved.</b> Write to 0.
1	IRMSSL	<b>MSR Register Function Select in IR Mode.</b> This bit selects the behavior of the MSR and Modem Status Interrupt (MS_EV) when an IR mode is selected. When a UART mode is selected, the MSR and the MS_EV function normally and this bit is ignored. 0: MSR and MS_EV work in the IR modes as in the UART mode. 1: MSR returns 30h, and MS_EV is disabled (Default).
0	IR_FDPLX	<b>Enable IR Full Duplex Mode.</b> When set to 1, the IR receiver is not masked during transmission.

### 6.12.9 Bank 6 Register Descriptions

The bit formats for the registers in Bank 6 are summarized in Table 6-44. Detailed descriptions of each register follow.

**Table 6-44. Bank 6 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h	IRCR3	SHDM_DS	SHMD_DS	RSVD					
01h	RSVD	RSVD							
02h	SIR_PW	RSVD				SPW3	SPW2	SPW1	SPW0
03h	BSR	BKSE	BSR[6:0]						
04h-07h	RSVD	RSVD							

#### 6.12.9.1 IR Control Register 3 (IRCR3)

I/O Offset        00h  
 Type             R/W  
 Reset Value     20h

IRCR3 is used to select the operating mode of the Sharp-IR interface.

**IRCR3 Register Map**

7	6	5	4	3	2	1	0
SHDM_DS	SHMD_DS	RSVD					

**IRCR3 Bit Descriptions**

Bit	Name	Description
7	SHDM_DS	<b>Sharp-IR Demodulation Disable.</b> 0: Internal 500 kHz receiver demodulation enabled (Default). 1: Internal demodulation disabled.
6	SHMD_DS	<b>Sharp-IR Modulation Disable.</b> 0: Internal 500 kHz transmitter modulation enabled (Default). 1: Internal modulation disabled.
5:0	RSVD	<b>Reserved.</b> Read as written.



**6.12.9.2 SIR Pulse Width Register (SIR\_PW)**

I/O Offset        02h  
 Type             R/W  
 Reset Value     00h

SIR\_PW sets the pulse width for transmitted pulses in SIR operation mode. These settings do not affect the receiver. Upon reset, the content of this register is 00h, which defaults to a pulse width of 3/16 of the baud rate.

**SIR\_PW Register Map**

7	6	5	4	3	2	1	0
RSVD				SPW[3:0]			

**SIR\_PW Bit Descriptions**

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Write to 0.
3:0	SPW[3:0]	<b>SIR Pulse Width.</b> Two codes for setting the pulse width are available. All other values for this field are reserved.  0000: Pulse width = 3/16 of bit period (Default). 1101: Pulse width = 1.6 $\mu$ s.

**6.12.9.3 Bank Select Register (BSR)**

I/O Offset        03h  
 Type             R/W  
 Reset Value     00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

### 6.12.10 Bank 7 Register Descriptions

The bit formats for the registers in Bank 7 are summarized in Table 6-44. Detailed descriptions of each register follow.

**Table 6-45. Bank 7 Bit Map**

Register		Bits							
I/O Offset	Name	7	6	5	4	3	2	1	0
00h	IRRXDC	DBW[2:0]			DFR[4:0]				
01h	IRTXMC	MCPW[2:0]			MCFR[4:0]				
02h	RCCFG	R_LEN	T_OV	RXHSC	RCDM_DS	RSVD	TXHSC	RC_MMD[1:0]	
03h	BSR	BKSE	BSR[6:0]						
04h	IRCFG1	STRV_MS	RSVD	SET_IRTX	IRRX1_LV	RSVD	IRIC[2:0]		
05h-06h	RSVD	RSVD							
07h	IRCFG4	RSVD		IRSL0_DS	RXINV	IRSL21_DS	RSVD		

The CEIR utilizes two carrier frequency ranges (See Table 6-50 on page 413):

- Low range - spans from 30 kHz to 56 kHz, in 1 kHz increments.
- High range - includes three frequencies: 400 kHz, 450 kHz or 480 kHz.

High and low frequencies are specified independently to allow separate transmission and reception modulation settings. The transmitter uses the carrier frequency settings in Table 6-50 "CEIR Carrier Frequency Encoding" on page 413.

The two registers at I/O Offsets 04h and 07h (IR transceiver configuration registers) are provided to configure the virtual IR dongle interface via IRSL[2:0] bits (to allow legacy software writes on these bits).

#### 6.12.10.1 IR Receiver Demodulator Control Register (IRRXDC)

I/O Offset	00h
Type	R/W
Reset Value	29h

IRRXDC controls settings for Sharp-IR and CEIR reception. After reset, the content of this register is 29h. This setting selects a subcarrier frequency in a range between 34.61 kHz and 38.26 kHz for the CEIR mode, and from 480.0 kHz to 533.3 kHz for the Sharp-IR mode. The value of this register is ignored if the receiver demodulation for both modes is disabled (see bit 7 (SHDM\_DS) in Section 6.12.9.1 "IR Control Register 3 (IRCR3)" on page 408 and bit 4 (RCDM\_DS) in Section 6.12.10.3 "CEIR Configuration Register (RCCFG)" on page 414). The available frequency ranges for CEIR and Sharp-IR modes are given in Tables 6-46 through 6-48.

**IRRXDC Register Map**

7	6	5	4	3	2	1	0
DBW[2:0]			DFR[4:0]				

#### IRRXDC Bit Descriptions

Bit	Name	Description
7:5	DBW[2:0]	<b>Demodulator Bandwidth.</b> These bits set the demodulator bandwidth for the selected frequency range. The subcarrier signal frequency must fall within the specified frequency range in order to be accepted. Used for both Sharp-IR and CEIR modes. (Default = 001.)
4:0	DFR[4:0]	<b>Demodulator Frequency.</b> These bits select the subcarrier's center frequency for the CEIR mode. (Default = 01001.)

**Table 6-46. CEIR, Low Speed Demodulator (RXHSC = 0) (Frequency Ranges in kHz)**

DFR[4:0]	DBW[2:0] Bits (Bits [7:5] of IRRXDC)											
	001		010		011		100		101		110	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
00011	28.6	31.6	27.3	33.3	26.1	35.3	25.0	37.5	24.0	40.0	23.1	42.9
00100	29.3	32.4	28.0	34.2	26.7	36.2	25.6	38.4	24.6	41.0	23.7	43.9
00101	30.1	33.2	28.7	35.1	27.4	37.1	26.3	39.4	25.2	42.1	24.3	45.1
00110	31.7	35.1	30.3	37.0	29.0	39.2	27.8	41.7	26.7	44.4	25.6	47.6
00111	32.6	36.0	31.1	38.1	29.8	40.3	28.5	42.8	27.4	45.7	26.3	48.9
01000	33.6	37.1	32.0	39.2	30.7	41.5	29.4	44.1	28.2	47.0	27.1	50.4
01001	34.6	38.3	33.0	40.4	31.6	42.8	30.3	45.4	29.1	48.5	28.0	51.9
01011	35.7	39.5	34.1	41.7	32.6	44.1	31.3	46.9	30.0	50.0	28.8	53.6
01100	36.9	40.7	35.2	43.0	33.7	45.5	32.3	48.4	31.0	51.6	29.8	55.3
01101	38.1	42.1	36.4	44.4	34.8	47.1	33.3	50.0	32.0	53.3	30.8	57.1
01111	39.4	43.6	37.6	45.9	36.0	48.6	34.5	51.7	33.1	55.1	31.8	59.1
10000	40.8	45.1	39.0	47.6	37.3	50.4	35.7	53.6	34.3	57.1	33.0	61.2
10010	42.3	46.8	40.4	49.4	38.6	52.3	37.0	55.6	35.6	59.3	34.2	63.5
10011	44.0	48.6	42.0	51.3	40.1	54.3	38.5	57.7	36.9	61.5	35.5	65.9
10101	45.7	50.5	43.6	53.3	41.7	56.5	40.0	60.0	38.4	64.0	36.9	68.6
10111	47.6	52.6	45.5	55.6	43.5	58.8	41.7	62.5	40.0	66.7	38.5	71.4
11010	49.7	54.9	47.4	57.9	45.3	61.4	43.5	65.2	41.7	69.5	40.1	74.5
11011	51.9	57.4	49.5	60.6	47.4	64.1	45.4	68.1	43.6	72.7	41.9	77.9
11101	54.4	60.1	51.9	63.4	49.7	67.2	47.6	71.4	45.7	76.1	43.9	81.6

**Table 6-47. Consumer IR High Speed Demodulator (RXHSC = 1) (Frequency Ranges in kHz)**

DFR[4:0]	DBW[2:0] Bits (Bits [7:5] of IRRXDC)											
	001		010		011		100		101		110	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
00011	381.0	421.1	363.6	444.4	347.8	470.6	333.3	500.0	320.0	533.3	307.7	571.4
01000	436.4	480.0	417.4	505.3	400.0	533.3	384.0	564.7	369.2	600.0	355.6	640.0
01011	457.7	505.3	436.4	533.3	417.4	564.7	400.0	600.0	384.0	640.0	369.9	685.6

**Table 6-48. Sharp-IR Demodulator (Frequency Ranges in kHz)**

DFR[4:0]	DBW[2:0] Bits (Bits [7:5] of IRRXDC)											
	001		010		011		100		101		110	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
xxxxx	480.0	533.3	457.1	564.7	436.4	600.0	417.4	640.0	400.0	685.6	384.0	738.5

**6.12.10.2 IR Transmitter Modulator Control Register (IRTXMC)**

I/O Offset      01h  
 Type            R/W  
 Reset Value    69h

IRTXMC selects the modulation subcarrier parameters for CEIR and Sharp-IR mode transmission. For Sharp-IR, only the subcarrier pulse width is controlled by this register; the subcarrier frequency is fixed at 500 kHz.

After reset, the value of this register is 69h, selecting a carrier frequency of 36 kHz and an IR pulse width of 7  $\mu$ s for CEIR, or a pulse width of 0.8  $\mu$ s for Sharp-IR.

**IRTXMC Bit Map**

7	6	5	4	3	2	1	0
MCPW[2:0]			MCFR[4:0]				

**IRTXMC Bit Descriptions**

Bit	Name	Description
7:5	MCPW[2:0]	<b>Modulation Subcarrier Pulse Width.</b> Specifies the pulse width of the subcarrier clock, as shown in Table 6-49. (Default = 011.)
4:0	MCFR[4:0]	<b>Modulation Subcarrier Frequency.</b> These bits set the frequency for the CEIR modulation subcarrier. The encoding is defined in Table 6-50. (Default = 01001.)

**Table 6-49. Modulation Carrier Pulse Width**

MCPW[2:0]	Low Frequency (TXHSC = 0) (CEIR only)	High Frequency (TXHSC = 1) (CEIR or Sharp-IR)
0 0 0	Reserved	Reserved
0 0 1	Reserved	Reserved
0 1 0	6.0 $\mu$ s	0.7 $\mu$ s
0 1 1	7.0 $\mu$ s	0.8 $\mu$ s
1 0 0	9.0 $\mu$ s	0.9 $\mu$ s
1 0 1	10.6 $\mu$ s	Reserved
1 1 0	Reserved	Reserved
1 1 1	Reserved	Reserved

**Table 6-50. CEIR Carrier Frequency Encoding**

<b>MCFR[4:0]</b>	<b>Low Frequency (TXHSC = 0)</b>	<b>High Frequency (TXHSC = 1)</b>
00000	Reserved	Reserved
00001	Reserved	Reserved
00010	Reserved	Reserved
00011	30 kHz	400 kHz
00100	31 kHz	Reserved
00101	32 kHz	Reserved
00110	33 kHz	Reserved
00111	34 kHz	Reserved
01000	35 kHz	450 kHz
01001	36 kHz	Reserved
01010	37 kHz	Reserved
01011	38 kHz	480 kHz
01100	39 kHz	Reserved
01101	40 kHz	Reserved
01110	41 kHz	Reserved
01111	42 kHz	Reserved
10000	43 kHz	Reserved
10001	44 kHz	Reserved
10010	45 kHz	Reserved
10011	46 kHz	Reserved
10100	47 kHz	Reserved
10101	48 kHz	Reserved
10110	49 kHz	Reserved
10111	50 kHz	Reserved
11000	51 kHz	Reserved
11001	52 kHz	Reserved
11010	53 kHz	Reserved
11011	54 kHz	Reserved
11100	55 kHz	Reserved
11101	56 kHz	Reserved
11110	56.9 kHz	Reserved
11111	Reserved	Reserved

**6.12.10.3 CEIR Configuration Register (RCCFG)**

I/O Offset        02h  
 Type             R/W  
 Reset Value     00h

RCCFG controls the basic operation of the CEIR mode.

**RCCFG Register Map**

7	6	5	4	3	2	1	0
R_LEN	T_OV	RXHSC	RCDM_DS	RSVD	TXHSC	RC_MMD[1:0]	

**RCCFG Bit Descriptions**

Bit	Name	Description
7	R_LEN	<b>Run Length Control.</b> When set to 1, this bit enables run length encoding/decoding. The format of a run length code is:  Yxxxxxxx, where Y is the bit value and xxxxxxx is the number of bits minus 1 (selects from 1 to 128 bits).
6	T_OV	<b>Receiver Sampling Mode.</b> 0: Programmed T period sampling. 1: Oversampling mode.
5	RXHSC	<b>Receiver Carrier Frequency Select.</b> This bit selects the receiver demodulator frequency range.  0: Low frequency: 30.0-56.9 kHz. 1: High frequency: 400-480 kHz.
4	RCDM_DS	<b>Receiver Demodulation Disable.</b> When this bit is 1, the internal demodulator is disabled. The internal demodulator, when enabled, performs carrier frequency checking and envelope generation. This bit must be set to 1 (disabled) when the demodulation is performed externally, or when oversampling mode is selected to determine the carrier frequency.
3	RSVD	<b>Reserved.</b> Write as 0.
2	TXHSC	<b>Transmitter Subcarrier Frequency Select.</b> This bit selects the modulation carrier frequency range.  0: Low frequency: 30.0-56.9 kHz. 1: High frequency: 400-480 kHz.
1:0	RC_MMD[1:0]	<b>Transmitter Modulator Mode.</b> Determines how IR pulses are generated from the transmitted bit string.  00: C_PLS modulation mode. Pulses are generated continuously for the entire logic 0 bit time.  01: 8_PLS modulation mode. 8 pulses are generated each time one or more logic 0 bits are transmitted following a logic 1 bit.  10: 6_PLS Modulation Mode. 6 pulses are generated each time one or more logic 0 bits are transmitted following a logic 1 bit.  11: Reserved. Result is indeterminate.

**6.12.10.4 Bank Select Register (BSR)**

I/O Offset 03h  
 Type R/W  
 Reset Value 00h

BSR is the same as the BSR register at I/O Offset 03h in Bank 0. See Section 6.12.3.4 "Link Control/Bank Select Registers" on page 390 for bit descriptions.

**6.12.10.5 IR Interface Configuration Register 1 (IRCFG1)**

I/O Offset 04h  
 Type R/W  
 Reset Value xxh

IRCFG1 holds the transceiver configuration data for Sharp-IR and SIR modes. It is also used to directly control the transceiver operation mode when automatic configuration is not enabled. The two next-to-least significant bits are used to read the identification data of a plug-and-play IR interface adapter.

**IRCFG1 Register Map**

7	6	5	4	3	2	1	0
STRV_MS	RSVD	SET_RTX	IRRX1_LV	RSVD	IRIC[2:0]		

**IRCFG1 Bit Descriptions**

Bit	Name	Description
7	STRV_MS	<b>Special Transceiver Mode Selection.</b> When this R/W bit is set to 1, the UART[x]_IR_TX output signal is forced to active high and a timer is started. The timer times out after 64 micro-seconds, at which time the bit is reset and the UART[x]_IR_TX output signal becomes low again. The timer is restarted every time a 1 is written to this bit. Although it is possible to extend the period during which UART[x]_IR_TX remains high beyond 64 micro-seconds, this should be avoided to prevent damage to the transmitter LED. Writing a 0 to this bit has no effect.
6	RSVD	<b>Reserved.</b> Write as 0.
5	SET_IRTX	<b>Set IRTX.</b> When this bit is set to 1, it forces the UART[x]_IR_TX signal high. <b>Caution:</b> Indefinite HIGH output should be avoided as this condition can damage the transmitter LED.
4	IRRX1_LV	<b>IRRX1 Level (Read Only).</b> This bit reflects the value of the UART[x]_IR_RX input signal.
3	RSVD	<b>Reserved.</b> Write as 0.
2:0	IRIC[2:0]	<b>Transceiver Identification and Control Bits 2 through 0.</b> The function of IRIC0 depends on whether the UART[x]_IRSL0/IRRX2 signal is programmed as an input or an output. If programmed as an input (IRSL0_DS = 0, bit 5 of IRCFG4) then upon a read, this bit returns the logic level of the signal. Data written to this bit position is ignored. The other two signals (IRSL1, IRSL2) must be programmed as outputs only (IRSL21_DS = 1, bit 3 of IRCFG4).  If the UART[x]_IRSL0/IRRX2 signal is programmed as an output, IRIC[2:0] drives the IRSL[2:0] signals to select the operation mode of an infrared dongle. (These bits are reflected in bits [4:2] of MSR_UART[x]_DONG). When read, these bits return the values previously written.  Below is the operation mode encoding for non-serial transceivers.  00x: IrDA-data modes. 010: Reserved. 011: 36 kHz consumer IR. 100: 40 kHz consumer IR. 101: 38 kHz consumer IR 110: Reserved. 111: 56.9 kHz consumer IR.

**6.12.10.6 IR Interface Configuration 4 Register (IRCFG4)**

I/O Offset      07h  
 Type            R/W  
 Reset Value    00h

IRCFG4 configures the receiver data path.

**IRCFG4 Register Map**

7	6	5	4	3	2	1	0
RSVD		IRSL0_DS	RXINV	IRSL21_DS	RSVD		

**IRCFG4 Bit Descriptions**

Bit	Name	Description
7:6	RSVD	<b>Reserved.</b> Must be written 0.
5	IRSL0_DS	<b>IRSL0/IRRX2 Pin Direction Select.</b> This bit determines the direction of the UART[x]_IRSL0/IRRX2 pin. See Table 6-51. 0: Pin direction is input (UART[x]_IRRX2). 1: Pin direction is output (UART[x]_IRSL0).
4	RXINV	<b>IRRX Signal Invert.</b> This bit supports optical transceivers with receive signals of opposite polarity (active high instead of active low). When set to 1, an inverter is placed in the receiver input signal path.
3	IRSL21_DS	<b>Reserved.</b> Must be written 1.
2:0	RSVD	<b>Reserved.</b> Must be written 0.

**Table 6-51. IR Receive Input Selection**

IRSL0_DS (IRCFG4, bit 5)	AUX_IRRX (IRCR2, bit 4)	Selected IRRX
0	0	IRRX1
0	1	IRRX2
1	0	IRRX1
1	1	None Selected



## 6.13 Direct Memory Access Register Descriptions

The registers for the Direct Memory Access (DMA) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- DMA Specific MSRs
- DMA Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the DMA Specific MSRs (summarized in Table 6-52) are called out as 16 bits. The DMA module treats writes to the upper 48 bits (i.e., bits [63:16]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The Native registers associated with the DMA module are summarized in Table 6-53 and accessed as I/O Addresses.

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-52. DMA Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400040h	R/W	DMA Mapper (DMA_MAP)	0000h	Page 420
51400041h	RO	DMA Shadow Channel 0 Mode (DMA_SHDW_CH0)	00xxh	Page 421
51400042	RO	DMA Shadow Channel 1 Mode (DMA_SHDW_CH1)	00xxh	Page 421
51400043	RO	DMA Shadow Channel 2 Mode (DMA_SHDW_CH2)	00xxh	Page 421
51400044	RO	DMA Shadow Channel 3 Mode (DMA_SHDW_CH3)	00xxh	Page 421
51400045h	RO	DMA Shadow Channel 4 Mode (DMA_SHDW_CH4)	00xxh	Page 421
51400046h	RO	DMA Shadow Channel 5 Mode (DMA_SHDW_CH5)	00xxh	Page 421
51400047h	RO	DMA Shadow Channel 6 Mode (DMA_SHDW_CH6)	00xxh	Page 421
51400048h	RO	DMA Shadow Channel 7 Mode (DMA_SHDW_CH7)	00xxh	Page 421
51400049h	RO	DMA Shadow Mask (DMA_MSK_SHDW)	00FFh	Page 422

**Table 6-53. DMA Native Registers Summary**

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
000h	R/W	8	Slave DMA Channel 0 Memory Address (DMA_CH0_ADDR_BYTE)	xxh	Page 422
001h	R/W	8	Slave DMA Channel 0 Transfer Count (DMA_CH0_CNT_BYTE)	xxh	Page 423
002h	R/W	8	Slave DMA Channel 1 Memory Address (DMA_CH1_ADDR_BYTE)	xxh	Page 422
003h	R/W	8	Slave DMA Channel 1 Transfer Count (DMA_CH1_CNT_BYTE)	xxh	Page 423
004h	R/W	8	Slave DMA Channel 2 Memory Address (DMA_CH2_ADDR_BYTE)	xxh	Page 422

Table 6-53. DMA Native Registers Summary (Continued)

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
005h	R/W	8	Slave DMA Channel 2 Transfer Count (DMA_CH2_CNT_BYTE)	xxh	Page 423
006h	R/W	8	Slave DMA Channel 3 Memory Address (DMA_CH3_ADDR_BYTE)	xxh	Page 422
007h	R/W	8	Slave DMA Channel 3 Transfer Count (DMA_CH3_CNT_BYTE)	xxh	Page 423
008h	R	8	Slave DMA Channel [3:0] Status (DMA_CH3:0_STS)	00h	Page 423
	W	8	Slave DMA Channel [3:0] Command (DMA_CH3:0_CMD)	xxh	Page 424
009h	WO	8	Slave DMA Channel [3:0] Software Request (DMA_CH3:0_SFT_REQ)	xxh	Page 424
00Ah	WO	8	Slave DMA Channel [3:0] Channel Mask (DMA_CH3:0_CHMSK)	xxh	Page 425
00Bh	WO	8	Slave DMA Channel [3:0] Mode (DMA_CH3:0_MODE)	xxh	Page 425
00Ch	WO	8	Slave DMA Channel [3:0] Clear Byte Pointer (DMA_CH3:0_CLR_PNTR)	xxh	Page 426
00Dh	WO	8	Slave DMA Channel [3:0] Master Clear (DMA_CH3:0_MSTR_CLR)	xxh	Page 426
00Eh	WO	8	Slave DMA Channel [3:0] Clear Mask Register (DMA_CH3:0_CLR_MSK)	xxh	Page 426
00Fh	WO	8	Slave DMA Channel [3:0] Write Mask Register (DMA_CH3:0_WR_MSK)	0Fh	Page 427
0C0h	R/W	8	Master DMA Channel 4 Memory Address (DMA_CH4_ADDR_BYTE)	xxh	Page 427
0C2h	R/W	8	Master DMA Channel 4 Transfer Count (DMA_CH4_CNT_BYTE)	xxh	Page 428
0C4h	R/W	8	Master DMA Channel 5 Memory Address (DMA_CH5_ADDR_BYTE)	xxh	Page 427
0C6h	R/W	8	Master DMA Channel 5 Transfer Count (DMA_CH5_CNT_BYTE)	xxh	Page 428
0C8h	R/W	8	Master DMA Channel 6 Memory Address (DMA_CH6_ADDR_BYTE)	xxh	Page 427
0CAh	R/W	8	Master DMA Channel 6 Transfer Count (DMA_CH6_CNT_BYTE)	xxh	Page 428
0CCh	R/W	8	Master DMA Channel 7 Memory Address (DMA_CH7_ADDR_BYTE)	xxh	Page 427
0CEh	R/W	8	Master DMA Channel 7 Transfer Count (DMA_CH7_CNT_BYTE)	xxh	Page 428
0D0h	R	8	Master DMA Channel [7:4] Status (DMA_CH7:4_STS)	00h	Page 428
	W	8	Master DMA Channel [7:4] Command (DMA_CH7:4_CMD)	xxh	Page 429
0D2h	WO	8	Master DMA Channel [7:4] Software Request (DMA_CH7:4_SFT_REQ)	xxh	Page 429

Table 6-53. DMA Native Registers Summary (Continued)

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
0D4h	WO	8	Master DMA Channel [7:4] Channel Mask (DMA_CH7:4_CHMSK)	xxh	Page 430
0D6h	WO	8	Master DMA Channel [7:4] Mode (DMA_CH7:4_MODE)	xxh	Page 430
0D8h	WO	8	Master DMA Channel [7:4] Clear Byte Pointer (DMA_CH7:4_CLR_PNTR)	xxh	Page 431
0DAh	WO	8	Master DMA Channel [7:4] Master Clear (DMA_CH7:4_MSTR_CLR)	xxh	Page 431
0DCh	WO	8	Master DMA Channel [7:4] Clear Mask (DMA_CH7:4_CLR_MSK)	xxh	Page 431
0DEh	WO	8	Master DMA Channel [7:4] Write Mask (DMA_CH7:4_WR_MSK)	0Fh	Page 432
080h	R/W	8	Post Code Display Register (POST_DISPLAY)	00h	Page 432
081h	R/W	8	DMA Channel 2 Low Page (DMA_CH2_LO_PAGE)	00h	Page 433
082h	R/W	8	DMA Channel 3 Low Page (DMA_CH3_LO_PAGE)	00h	Page 433
083h	R/W	8	DMA Channel 1 Low Page (DMA_CH1_LO_PAGE)	00h	Page 433
087h	R/W	8	DMA Channel 0 Low Page (DMA_CH0_LO_PAGE)	00h	Page 433
089h	R/W	8	DMA Channel 6 Low Page (DMA_CH6_LO_PAGE)	00h	Page 433
08Ah	R/W	8	DMA Channel 7 Low Page (DMA_CH7_LO_PAGE)	00h	Page 433
08Bh	R/W	8	DMA Channel 5 Low Page (DMA_CH5_LO_PAGE)	00h	Page 433
08Fh	R/W	8	DMA Channel 4 Low Page (DMA_CH4_LO_PAGE)	00h	Page 433
481h	R/W	8	DMA Channel 2 High Page (DMA_CH2_HI_PAGE)	00h	Page 434
482h	R/W	8	DMA Channel 3 High Page (DMA_CH3_HI_PAGE)	00h	Page 434
483h	R/W	8	DMA Channel 1 High Page (DMA_CH1_HI_PAGE)	00h	Page 434
487h	R/W	8	DMA Channel 0 High Page (DMA_CH0_HI_PAGE)	00h	Page 434
489h	R/W	8	DMA Channel 6 High Page (DMA_CH6_HI_PAGE)	00h	Page 434
48Ah	R/W	8	DMA Channel 7 High Page (DMA_CH7_HI_PAGE)	00h	Page 434
48Bh	R/W	8	DMA Channel 5 High Page (DMA_CH5_HI_PAGE)	00h	Page 434
48Fh	R/W	8	DMA Channel 4 High Page (DMA_CH4_HI_PAGE)	00h	Page 434

### 6.13.1 DMA Specific MSRs

#### 6.13.1.1 DMA Mapper (DMA\_MAP)

MSR Address 51400040h

Type R/W

Reset Value 0000h

**DMA\_MAP Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	DMA_CH3_MAP			RSVD	DMA_CH2_MAP			RSVD	DMA_CH1_MAP			RSVD	DMA_CH0_MAP		

**DMA\_MAP Bit Descriptions**

Bit	Name	Description
15	RSVD	<b>Reserved.</b> No effect on DMA Mapper functionality; reads return value written.
14:12	DMA_CH3_MAP	<b>DMA Channel 3 Source Select.</b> 000: DMA Channel 3 off.                      100: UART2 receive. 001: UART1 transmit.                      101: Reserved; not active. 010: UART1 receive.                      110: Reserved; not active. 011: UART2 transmit.                      111: LPC DMA Channel 3.
11	RSVD	<b>Reserved.</b> No effect on DMA Mapper functionality; reads return value written.
10:8	DMA_CH2_MAP	<b>DMA Channel 2 Source Select.</b> 000: DMA Channel 2 off.                      100: UART2 receive. 001: UART1 transmit.                      101: Reserved; not active. 010: UART1 receive.                      110: Reserved; not active. 011: UART2 transmit.                      111: LPC DMA Channel 2.
7	RSVD	<b>Reserved.</b> No effect on DMA Mapper functionality; reads return value written.
6:4	DMA_CH1_MAP	<b>DMA Channel 1 Source Select.</b> 000: DMA Channel 1 off.                      100: UART2 receive. 001: UART1 transmit.                      101: Reserved; not active. 010: UART1 receive.                      110: Reserved; not active. 011: UART2 transmit.                      111: LPC DMA Channel 1.
3	RSVD	<b>Reserved.</b> No effect on DMA Mapper functionality; reads return value written.
2:0	DMA_CH0_MAP	<b>DMA Channel 0 Source Select.</b> 000: DMA Channel 0 off.                      100: UART2 receive. 001: UART1 transmit.                      101: Reserved; not active. 010: UART1 receive.                      110: Reserved; not active. 011: UART2 transmit.                      111: LPC DMA Channel 0.

## 6.13.1.2 DMA Shadow Channel [7:0] Mode (DMA\_SHDW\_CH[x])

**DMA Shadow Channel 0 Mode (DMA\_SHDW\_CH0)**

MSR Address 51400041h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 4 Mode (DMA\_SHDW\_CH4)**

MSR Address 51400045h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 1 Mode (DMA\_SHDW\_CH1)**

MSR Address 51400042h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 5 Mode (DMA\_SHDW\_CH5)**

MSR Address 51400046h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 2 Mode (DMA\_SHDW\_CH2)**

MSR Address 51400043h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 6 Mode (DMA\_SHDW\_CH6)**

MSR Address 51400047h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 3 Mode (DMA\_SHDW\_CH3)**

MSR Address 51400044h  
 Type RO  
 Reset Value 00xxh

**DMA Shadow Channel 7 Mode (DMA\_SHDW\_CH7)**

MSR Address 51400048h  
 Type RO  
 Reset Value 00xxh

**DMA\_SHDW\_CH[x] Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								TRANS_MODE		ADDR_DIR	AUTO_INIT	TRANS_TYPE		CH_NUM	

**DMA\_SHDW\_CH[x] Bit Descriptions**

Bit	Name	Description
15:8	RSVD	<b>Reserved.</b> Reads as 00h.
7:6	TRANS_MODE	<b>Data Transfer Mode.</b> 00: Demand. 10: Block. 01: Single. 11: Cascade
5	ADDR_DIR	<b>Address Direction.</b> 0: Increment; 1: Decrement.
4	AUTO_INIT	<b>Auto-Initialization Enable.</b> 0: Disabled; 1: Enabled.
3:2	TRANS_TYPE	<b>Transfer Type.</b> 00: Verify. 10: Memory read. 01: Memory write. 11: Reserved.
1:0	CH_NUM	<b>Channel Number [3:0].</b> 00: Channel 0. 10: Channel 2. 01: Channel 1. 11: Channel 3.
		<b>Channel Number [7:4].</b> 00: Channel 4. 10: Channel 6. 01: Channel 5. 11: Channel 7.

**6.13.1.3 DMA Shadow Mask (DMA\_MSK\_SHDW)**

MSR Address 51400049h  
 Type RO  
 Reset Value 00FFh

**DMA\_MSK\_SHDW Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CH7MASK	CH6MASK	CH5MASK	CH4MASK	CH3MASK	CH2MASK	CH1MASK	CH0MASK

**DMA\_MSK\_SHDW Bit Descriptions**

Bit	Name	Description
15:8	Reserved	<b>Reserved.</b> Reads as 00h.
7	CH7MASK	<b>Channel 7 Mask</b> Reflects value of Channel 7 Mask bit. 0: Not masked; 1: Masked.
6	CH6MASK	<b>Channel 6 Mask.</b> Reflects value of Channel 6 Mask bit. 0: Not masked; 1: Masked.
5	CH5MASK	<b>Channel 5 Mask.</b> Reflects value of Channel 5 Mask bit. 0: Not masked; 1: Masked.
4	CH4MASK	<b>Channel 4 Mask.</b> Reflects value of Channel 4 Mask bit. 0: Not masked; 1: Masked.
3	CH3MASK	<b>Channel 3 Mask.</b> Reflects value of Channel 3 Mask bit. 0: Not masked; 1: Masked.
2	CH2MASK	<b>Channel 2 Mask.</b> Reflects value of Channel 2 Mask bit. 0: Not masked; 1: Masked.
1	CH1MASK	<b>Channel 1 Mask.</b> Reflects value of Channel 1 Mask bit. 0: Not masked; 1: Masked.
0	CH0MASK	<b>Channel 0 Mask.</b> Reflects value of Channel 0 Mask bit. 0: Not masked; 1: Masked.

**6.13.2 DMA Native Registers**

These registers reside in the I/O address space.

**6.13.2.1 Slave DMA Channel [3:0] Memory Address (DMA\_CH[x]\_ADDR\_BYTE)****Slave DMA Channel 0 Memory Address (DMA\_CH0\_ADDR\_BYTE)**

I/O Address 000h  
 Type R/W  
 Reset Value xxh

**Slave DMA Channel 2 Memory Address (DMA\_CH2\_ADDR\_BYTE)**

I/O Address 004h  
 Type R/W  
 Reset Value xxh

**Slave DMA Channel 1 Memory Address (DMA\_CH1\_ADDR\_BYTE)**

I/O Address 002h  
 Type R/W  
 Reset Value xxh

**Slave DMA Channel 3 Memory Address (DMA\_CH3\_ADDR\_BYTE)**

I/O Address 006h  
 Type R/W  
 Reset Value xxh

**DMA\_CH[x]\_ADDR\_BYTE Register Map**

7	6	5	4	3	2	1	0
DMA_CH_ADDR_BYTE							

**DMA\_CH[x]\_ADDR\_BYTE Bit Descriptions**

Bit	Name	Description
7:0	DMA_CH_ADDR_BYTE	<b>DMA Channel Address.</b> Read/write in two successive bus cycles, low byte first.

### 6.13.2.2 Slave DMA Channel [3:0] Transfer Count (DMA\_CH[x]\_CNT\_BYTE)

#### Slave DMA Channel 0 Transfer Count (DMA\_CH0\_CNT\_BYTE)

I/O Address 001h  
Type R/W  
Reset Value xxh

#### Slave DMA Channel 2 Transfer Count (DMA\_CH2\_CNT\_BYTE)

I/O Address 005h  
Type R/W  
Reset Value xxh

#### Slave DMA Channel 1 Transfer Count (DMA\_CH1\_CNT\_BYTE)

I/O Address 003h  
Type R/W  
Reset Value xxh

#### Slave DMA Channel 3 Transfer Count (DMA\_CH3\_CNT\_BYTE)

I/O Address 007h  
Type R/W  
Reset Value xxh

#### DMA\_CH[x]\_CNT\_BYTE Register Map

7	6	5	4	3	2	1	0
DMA_CH_CNT_BYTE							

#### DMA\_CH[x]\_CNT\_BYTE Bit Descriptions

Bit	Name	Description
7:0	DMA_CH_CNT_BYTE	<b>DMA Channel Transfer Count.</b> Read/write in two successive bus cycles, low byte first.

### 6.13.2.3 DMA Channel [3:0] Status / Command

#### Slave DMA Channel [3:0] Status (DMA\_CH3:0\_STS)

I/O Address 008h  
Type R  
Reset Value 00h

#### DMA\_CH3:0\_STS Register Map

7	6	5	4	3	2	1	0
DMA_CH3_RQ	DMA_CH2_RQ	DMA_CH1_RQ	DMA_CH0_RQ	DMA_CH3_TC	DMA_CH2_TC	DMA_CH1_TC	DMA_CH0_TC

#### DMA\_CH3:0\_STATUS Bit Descriptions

Bit	Name	Description
7	DMA_CH3_RQ	<b>Channel 3 DMA Request.</b> 0: Not pending; 1: Pending.
6	DMA_CH2_RQ	<b>Channel 2 DMA Request.</b> 0: Not pending; 1: Pending.
5	DMA_CH1_RQ	<b>Channel 1 DMA Request.</b> 0: Not pending; 1: Pending.
4	DMA_CH0_RQ	<b>Channel 0 DMA Request.</b> 0: Not pending; 1: Pending.
3	DMA_CH3_TC	<b>Channel 3 Terminal Count.</b> 0: Count not reached; 1: Count reached.
2	DMA_CH2_TC	<b>Channel 2 Terminal Count.</b> 0: Count not reached; 1: Count reached.
1	DMA_CH1_TC	<b>Channel 1 Terminal Count.</b> 0: Count not reached; 1: Count reached.
0	DMA_CH0_TC	<b>Channel 0 Terminal Count.</b> 0: Count not reached; 1: Count reached.







**6.13.2.7 Slave DMA Channel [3:0] Clear Byte Pointer (DMA\_CH3:0\_CLR\_PNTR)**

I/O Address Offset 00Ch  
 Type WO  
 Reset Value xxh

**DMA\_CH3:0\_CLR\_PNTR Register Map**

7	6	5	4	3	2	1	0
CLR_PNTR (DUMMY_VAL)							

**DMA\_CH3:0\_CLR\_PNTR Bit Descriptions**

Bit	Name	Description
7:0	CLR_PNTR	<b>Clear Pointer.</b> A write with any data (dummy value) resets high/low byte pointer for Channels 3:0 memory address and terminal count registers.

**6.13.2.8 Slave DMA Channel [3:0] Master Clear (DMA\_CH3:0\_MSTR\_CLR)**

I/O Address 00Dh  
 Type WO  
 Reset Value xxh

**DMA\_CH3:0\_MSTR\_CLR Register Map**

7	6	5	4	3	2	1	0
MSTR_CLR (DUMMY_VAL)							

**DMA\_CH3:0\_MSTR\_CLR Bit Descriptions**

Bit	Name	Description
7:0	MSTR_CLR	<b>Master Clear.</b> A write with any data (dummy value) resets the 8237 DMA controller for Channels [3:0].

**6.13.2.9 Slave DMA Channel [3:0] Clear Mask Register (DMA\_CH3:0\_CLR\_MSK)**

I/O Address 00Eh  
 Type WO  
 Reset Value xxh

**DMA\_CH3:0\_CLR\_MSK Register Map**

7	6	5	4	3	2	1	0
CLR_MSK (DUMMY_VAL)							

**DMA\_CH3:0\_CLR\_MSK Bit Descriptions**

Bit	Name	Description
7:0	CLR_MSK	<b>Clear Mask.</b> A write with any data (dummy value) clears the mask bits for Channels [3:0].

**6.13.2.10 Slave DMA Channel [3:0] Write Mask Register (DMA\_CH3:0\_WR\_MSK)**

I/O Address 00Fh  
 Type WO  
 Reset Value 0Fh

**DMA\_CH3:0\_WR\_MSK Register Map**

7	6	5	4	3	2	1	0
RSVD				CH3_MASK	CH2_MASK	CH1_MASK	CH0_MASK

**DMA\_CH3:0\_WR\_MSK Bit Descriptions**

Bit	Name	Description
7:4	Reserved	<b>Reserved.</b> Write value is don't care.
3	CH3_MASK	<b>Channel 3 Mask Value.</b> 0: Not masked; 1: Masked.
2	CH2_MASK	<b>Channel 2 Mask Value.</b> 0: Not masked; 1: Masked.
1	CH1_MASK	<b>Channel 1 Mask Value.</b> 0: Not masked; 1: Masked.
0	CH0_MASK	<b>Channel 0 Mask Value.</b> 0: Not masked; 1: Masked.

**6.13.2.11 Master DMA Channel [7:4] Memory Address (DMA\_CH[x]\_ADDR\_BYTE)****Master DMA Channel 4 Memory Address (DMA\_CH4\_ADDR\_BYTE)**

I/O Address 0C0h  
 Type R/W  
 Reset Value xxh

**Master DMA Channel 6 Memory Address (DMA\_CH6\_ADDR\_BYTE)**

I/O Address 0C8h  
 Type R/W  
 Reset Value xxh

**Master DMA Channel 5 Memory Address (DMA\_CH5\_ADDR\_BYTE)**

I/O Address 0C4h  
 Type R/W  
 Reset Value xxh

**Master DMA Channel 7 Memory Address (DMA\_CH7\_ADDR\_BYTE)**

I/O Address 0CCh  
 Type R/W  
 Reset Value xxh

**DMA\_CH[x]\_ADDR\_BYTE Register Map**

7	6	5	4	3	2	1	0
DMA_CH_ADDR_BYTE							

**DMA\_CH[x]\_ADDR\_BYTE Bit Descriptions**

Bit	Name	Description
7:0	DMA_CH_ADDR_BYTE	<b>DMA Channel Address.</b> Read/write in two successive bus cycles, low byte first.

### 6.13.2.12 Master DMA Channel [7:4] Transfer Count (DMA\_CH[x]\_CNT\_BYTE)

#### Master DMA Channel 4 Transfer Count (DMA\_CH4\_CNT\_BYTE)

I/O Address 0C2h  
Type R/W  
Reset Value xxh

#### Master DMA Channel 6 Transfer Count (DMA\_CH6\_CNT\_BYTE)

I/O Address 0CAh  
Type R/W  
Reset Value xxh

#### Master DMA Channel 5 Transfer Count (DMA\_CH5\_CNT\_BYTE)

I/O Address 0C6h  
Type R/W  
Reset Value xxh

#### Master DMA Channel 7 Transfer Count (DMA\_CH7\_CNT\_BYTE)

I/O Address 0CEh  
Type R/W  
Reset Value xxh

#### DMA\_CH[x]\_CNT\_BYTE Register Map

7	6	5	4	3	2	1	0
DMA_CH_CNT_BYTE							

#### DMA\_CH[x]\_CNT\_BYTE Bit Descriptions

Bit	Name	Description
7:0	DMA_CH_CNT_BYTE	<b>DMA Channel Transfer Count.</b> Read/write in two successive bus cycles, low byte first.

### 6.13.2.13 DMA Channel [7:4] Status / Command

#### Master DMA Channel [7:4] Status (DMA\_CH7:4\_STS)

I/O Address 0D0h  
Type R  
Reset Value 00h

#### DMA\_CH7:4\_STS Register Map

7	6	5	4	3	2	1	0
DMA_CH7_RQ	DMA_CH6_RQ	DMA_CH5_RQ	DMA_CH4_RQ	DMA_CH7_TC	DMA_CH6_TC	DMA_CH5_TC	DMA_CH4_TC

#### DMA\_CH7:4\_STS Bit Descriptions

Bit	Name	Description
7	DMA_CH7_RQ	<b>Channel 7 DMA Request.</b> 0: Not pending; 1: Pending.
6	DMA_CH6_RQ	<b>Channel 6 DMA Request.</b> 0: Not pending; 1: Pending.
5	DMA_CH5_RQ	<b>Channel 5 DMA Request.</b> 0: Not pending; 1: Pending.
4	DMA_CH4_RQ	<b>Channel 4 DMA Request.</b> 0: Not pending; 1: Pending.
3	DMA_CH7_TC	<b>Channel 7 Terminal Count.</b> 0: Count not reached; 1: Count reached.
2	DMA_CH6_TC	<b>Channel 6 Terminal Count.</b> 0: Count not reached; 1: Count reached.
1	DMA_CH5_TC	<b>Channel 5 Terminal Count.</b> 0: Count not reached; 1: Count reached.
0	DMA_CH4_TC	<b>Channel 4 Terminal Count.</b> 0: Count not reached; 1: Count reached.





**6.13.2.17 Master DMA Channel [7:4] Clear Byte Pointer (DMA\_CH7:4\_CLR\_PNTR)**

I/O Address 0D8h  
 Type WO  
 Reset Value xxh

**DMA\_CH7:4\_CLR\_PNTR Register Map**

7	6	5	4	3	2	1	0
CLR_PNTR (DUMMY_VAL)							

**DMA\_CH7:4\_CLR\_PNTR Bit Descriptions**

Bit	Name	Description
7:0	CLR_PNTR	<b>Clear Pointer.</b> A write with any data (dummy value) resets high/low byte pointer for Channels [7:4] memory address and terminal count registers.

**6.13.2.18 Master DMA Channel [7:4] Master Clear (DMA\_CH7:4\_MSTR\_CLR)**

I/O Address Offset 0DAh  
 Type WO  
 Reset Value xxh

**DMA\_CH7:4\_MSTR\_CLR Register Descriptions**

7	6	5	4	3	2	1	0
MSTR_CLR (DUMMY_VAL)							

**DMA\_CH7:4\_MSTR\_CLR Bit Descriptions**

Bit	Name	Description
7:0	MSTR_CLR	<b>Master Clear.</b> A write with any data (dummy value) resets the 8237 DMA controller for Channels [7:4].

**6.13.2.19 Master DMA Channel [7:4] Clear Mask (DMA\_CH7:4\_CLR\_MSK)**

I/O Address Offset 0DCh  
 Type WO  
 Reset Value xxh

**DMA Clear Mask Register for Channels 7:4**

7	6	5	4	3	2	1	0
CLR_MSK (DUMMY_VAL)							

**DMA\_CH7:4\_CLR\_MSK Bit Descriptions**

Bit	Name	Description
7:0	CLR_MSK	<b>Clear Mask.</b> A write with any data (dummy value) clears the mask bits for Channels [7:4].

**6.13.2.20 Master DMA Channel [7:4] Write Mask (DMA\_CH7:4\_WR\_MSK)**

I/O Address 0DEh  
 Type WO  
 Reset Value 0Fh

**DMA\_CH7:4\_WR\_MSK Register Map**

7	6	5	4	3	2	1	0
RSVD				CH7_MASK	CH6_MASK	CH5_MASK	CH4_MASK

**DMA\_CH7:4\_WR\_MSK Bit Descriptions**

Bit	Name	Description
7:4	RSVD	<b>Reserved.</b> Write value is don't care.
3	CH7_MASK	<b>Channel 7 Mask Value.</b> 0: Not masked; 1: Masked.
2	CH6_MASK	<b>Channel 6 Mask Value.</b> 0: Not masked; 1: Masked.
1	CH5_MASK	<b>Channel 5 Mask Value.</b> 0: Not masked; 1: Masked.
0	CH4_MASK	<b>Channel 4 Mask Value.</b> 0: Not masked; 1: Masked.

**6.13.2.21 Post Code Display Register (POST\_DISPLAY)**

I/O Address 080h  
 Type R/W  
 Reset Value 00h

**POST\_DISPLAY Register Map**

7	6	5	4	3	2	1	0
POST_CODE							

**POST\_DISPLAY Bit Descriptions**

Bit	Name	Description
7:0	POST_CODE	<b>Post Code Display Value.</b> This register is the historical Port 80 that receives the POST (Power-On Self-Test) codes reported during initialization. Typically used by the BIOS, Port 80 may also be written to by any piece of executing software to provide status or other information. The most recent value written to Port 80 is recorded in this register.



**6.13.2.22 DMA Channel [3:0] Low Page (DMA\_CH[x]\_LO\_PAGE)****DMA Channel 2 Low Page (DMA\_CH2\_LO\_PAGE)**

I/O Address 081h  
 Type R/W  
 Reset Value 00h

**DMA Channel 1 Low Page (DMA\_CH1\_LO\_PAGE)**

I/O Address 083h  
 Type R/W  
 Reset Value 00h

**DMA Channel 3 Low Page (DMA\_CH3\_LO\_PAGE)**

I/O Address 082h  
 Type R/W  
 Reset Value 00h

**DMA Channel 0 Low Page (DMA\_CH0\_LO\_PAGE)**

I/O Address Offset 087h  
 Type R/W  
 Reset Value 00h

**DMA\_CH[x]\_LO\_PAGE Register Map**

7	6	5	4	3	2	1	0
DMA_CH_LO_PAGE							

**DMA\_CH[x]\_LO\_PAGE Bit Descriptions**

Bit	Name	Description
7:0	DMA_CH_LO_PAGE	<b>DMA Channel Low Page Value.</b> Address bits [23:16].

**6.13.2.23 DMA Channel [7:4] Low Page (DMA\_CH[x]\_LO\_PAGE)****DMA Channel 6 Low Page (DMA\_CH6\_LO\_PAGE)**

I/O Address 089h  
 Type R/W  
 Reset Value 00h

**DMA Channel 5 Low Page (DMA\_CH5\_LO\_PAGE)**

I/O Address 08Bh  
 Type R/W  
 Reset Value 00h

**DMA Channel 7 Low Page (DMA\_CH7\_LO\_PAGE)**

I/O Address 08Ah  
 Type R/W  
 Reset Value 00h

**DMA Channel 4 Low Page (DMA\_CH4\_LO\_PAGE)**

I/O Address 08Fh  
 Type R/W  
 Reset Value 00h

**DMA\_CH[x]\_LO\_PAGE Register Map**

7	6	5	4	3	2	1	0
DMA_CH_LO_PAGE							RSVD

**DMA\_CH[x]\_LO\_PAGE Bit Descriptions**

Bit	Name	Description
7:1	DMA_CH_LO_PAGE	<b>DMA Channel Low Page Value.</b> Address bits [23:17].
0	RSVD	<b>Reserved.</b> Not used to generate DMA address. Write value is don't care; reads return value written.

**6.13.2.24 DMA Channel [7:0] High Page (DMA\_CH[x]\_HI\_PAGE)****DMA Channel 2 High Page (DMA\_CH2\_HI\_PAGE)**

I/O Address 481h  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 81h.

**DMA Channel 3 High Page (DMA\_CH3\_HI\_PAGE)**

I/O Address 482h  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 82h.

**DMA Channel 1 High Page (DMA\_CH1\_HI\_PAGE)**

I/O Address 483h  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 83h.

**DMA Channel 0 High Page (DMA\_CH0\_HI\_PAGE)**

I/O Address 487h  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 87h.

**DMA Channel 6 High Page (DMA\_CH6\_HI\_PAGE)**

I/O Address 489h  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 89h.

**DMA Channel 7 High Page (DMA\_CH7\_HI\_PAGE)**

I/O Address Offset 48Ah  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 8Ah.

**DMA Channel 5 High Page (DMA\_CH5\_HI\_PAGE)**

I/O Address 48Bh  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 8Bh.

**DMA Channel 4 High Page (DMA\_CH4\_HI\_PAGE)**

I/O Address 48Fh  
 Type R/W  
 Reset Value 00h

This register is also cleared on any access to I/O Port 8Fh.

**DMA\_CH[x]\_HI\_PAGE Register Map**

7	6	5	4	3	2	1	0
DMA_CH_HI_PAGE							

**DMA\_CH[x]\_HI\_PAGE Bit Descriptions**

Bit	Name	Description
7:0	DMA_CH_HI_PAGE	<b>DMA Channel High Page Value.</b> Address bits [31:24].

## 6.14 Low Pin Count Register Descriptions

The registers for the Low Pin Count (LPC) port are divided into two sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- LPC Specific MSRs

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the LPC Specific MSRs are called out as 32 bits. The LPC module treats writes to the upper 32 bits (i.e., bits [63:32]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The LPC Specific MSRs are summarized in Table 6-54. The reference column in the summary table points to the page where the register maps and bit descriptions are listed.

**Table 6-54. LPC Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
5140004Ch	RO	LPC Address Error (LPC_EADDR)	00000000h	Page 435
5140004Dh	RO	LPC Error Status (LPC_ESTAT)	00000000h	Page 435
5140004Eh	R/W	LPC Serial IRQ Control (LPC_SIRQ)	00000000h	Page 435
5140004Fh	R/W	LPC Reserved (LPC_RSVD)	00000000h	Page 435

### 6.14.1 LPC Specific MSRs

The LPC Controller uses the MSR\_LPC\_EADDR and MSR\_LPC\_ESTAT to record the indicated information associated with any given LPC bus error. The recorded information can not be cleared or modified. Simultaneous with this recording, an error event is sent to the DIVIL\_GLD\_MSR\_ERROR (MSR 51400003h). If enabled at the DIVIL\_GLD\_MSR\_ERROR, any LPC error event is recorded with a flag. This flag can be cleared. The status of this flag should be used to determine if there is an outstanding error associated with the two MSRs below.

#### 6.14.1.1 LPC Address Error (LPC\_EADDR)

MSR Address      5140004Ch  
 Type              RO  
 Reset Value      00000000h

#### LPC\_EADDR Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPC_ERR_ADDR																															

#### LPC\_EADDR Bit Descriptions

Bit	Name	Description
31:0	LPC_ERR_ADDR	<b>LPC Error Address.</b> When an error occurs, this register captures the associated 32-bit address.

**6.14.1.2 LPC Error Status (LPC\_ESTAT)**

MSR Address 5140004Dh  
 Type RO  
 Reset Value 00000000h

**LPC\_ESTAT Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																											TIMEOUT	DMA	WRITE	MEMORY	

**ESTAT Bit Descriptions**

Bit	Name	Description
31:4	RSVD	<b>Reserved.</b> Reads as 0.
3	TIMEOUT	<b>Timeout.</b> If set, indicates an LPC error was caused by a timeout.
2	DMA	<b>DMA.</b> If set, indicates an LPC error occurred during a DMA transaction.
1	WRITE	<b>Write.</b> If set, indicates an LPC error occurred during an LPC write transaction.
0	MEMORY	<b>Memory.</b> If set, indicates an LPC error occurred during an LPC memory transaction.

**6.14.1.3 LPC Serial IRQ Control (LPC\_SIRQ)**

MSR Address 5140004Eh  
 Type R/W  
 Reset Value 00000000h

**LPC\_SIRQ Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INVERT[15:0]												RSVD						SIRQ_EN	SIRQ_MODE	IRQ_FRAME			START_FPW								



## 6.15 Real-Time Clock Register Descriptions

The registers for the Real-Time Clock (RTC) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- RTC Specific MSRs
- RTC Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the RTC Specific MSRs (summarized in Table 6-55) are called out as 8 bits. The RTC module treats writes to the upper 56 bits (i.e., bits [63:8]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The Native registers associated with the RTC module are summarized in Table 6-56 and are accessed as I/O Addresses.

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-55. RTC Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400054h	R/W	RTC RAM Lock (RTC_RAM_LOCK)	00h	Page 439
51400055h	R/W	RTC Date of Month Alarm Offset (RTC_DOMA_OFFSET)	00h	Page 440
51400056h	R/W	RTC Month Alarm Offset (RTC_MONA_OFFSET)	00h	Page 440
51400057h	R/W	RTC Century Offset (RTC_CEN_OFFSET)	00h	Page 441

**Table 6-56. RTC Native Registers Summary**

I/O Address	Type	Width (Bits)	Register Name	Reset Value	Reference
00h	R/W	8	Seconds (RTC_SEC)	00h	Page 441
01h	R/W	8	Seconds Alarm (RTC_SECA)	00h	Page 442
02h	R/W	8	Minutes (RTC_MIN)	00h	Page 442
03h	R/W	8	Minutes Alarm (RTC_MINA)	00h	Page 442
04h	R/W	8	Hours (RTC_HR)	00h	Page 443
05h	R/W	8	Hours Alarm	00h	Page 443
06h	R/W	8	Day of Week (RTC_DOW)	00h	Page 443
07h	R/W	8	Day of Month (RTC_DOM)	00h	Page 444
08h	R/W	8	Month (RTC_MONTH)	00h	Page 444
09h	R/W	8	Year (RTC_YEAR)	00h	Page 444
0Ah	R/W	8	RTC Control Register A (RTC_CRA)	20h	Page 445
0Bh	R/W	8	RTC Control Register B (RTC_CRB)	00h	Page 446
0Ch	RO	8	RTC Control Register C (RTC_CRC)	00h	Page 447
0Dh	RO	8	RTC Control Register D (RTC_CRD)	00h	Page 448
Programmable (Note 1)	R/W	8	Date of Month Alarm (RTC_DOMA)	00h	Page 448
Programmable (Note 1)	R/W	8	Month Alarm (RTC_MONA)	00h	Page 449
Programmable (Note 1)	R/W	8	Century (RTC_CEN)	00h	Page 449

Note 1. Register location is programmable (through the MSR registers) and overlay onto the lower RAM space.

## 6.15.1 RTC Specific MSRs

### 6.15.1.1 RTC RAM Lock (RTC\_RAM\_LOCK)

MSR Address 51400054h  
 Type R/W  
 Reset Value 00h

When a non-reserved bit is set to 1, it can only be cleared by hardware reset.

#### RTC\_RAM\_LOCK Register Map

7	6	5	4	3	2	1	0
BLK_STDRAM	BLK_RAM_WR	BLK_XRAM_WR	BLK_XRAM_RD	BLK_XRAM	RSVD		

#### RTC\_RAM\_LOCK Bit Descriptions

Bit	Name	Description
7	BLK_STDRAM	<b>Block Standard RAM.</b> 0: No effect on Standard RAM access (default). 1: Read and write to locations 38h-3Fh of the Standard RAM are blocked, writes ignored, and reads return FFh.
6	BLK_RAM_WR	<b>Block RAM Write.</b> 0: No effect on RAM access (default). 1: Write to RAM (Standard and Extended) are ignored.
5	BLK_XRAM_WR	<b>Block Extended RAM Write.</b> This bit controls write to bytes 00h-1Fh of the Extended RAM. 0: No effect on Extended RAM access (default). 1: Writes to byte 00h-1Fh of the Extended RAM are ignored.
4	BLK_XRAM_RD	<b>Block Extended RAM Read.</b> This bit controls read from bytes 00h-1Fh of the Extended RAM. 0: No effect on Extended RAM access (default). 1: Reads from byte 00h-1Fh of the Extended RAM are ignored.
3	BLK_XRAM	<b>Block Extended RAM.</b> This bit controls access to the Extended RAM 128 bytes. 0: No effect on Extended RAM access (default). 1: Read and write to the Extended RAM are blocked; writes are ignored and reads return FFh.
2:0	RSVD	<b>Reserved.</b> Write as 0.

**6.15.1.2 RTC Date of Month Alarm Offset (RTC\_DOMA\_OFFSET)**

MSR Address 51400055h  
 Type R/W  
 Reset Value 00h

**RTC\_DOMA\_OFFSET Register Map**

7	6	5	4	3	2	1	0
RSVD	DOMA_OFST						

**RTC\_DOMA\_OFFSET Bit Descriptions**

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write as 0.
6:0	DOMA_OFST	<b>Date of Month Alarm Register Offset Value.</b> This register sets the location in RAM space of the Date Of Month Alarm register. This register must be programmed after a hardware reset, otherwise the Day Of Month Alarm register will be on top of other RAM data. It is programmed as an offset from 0. Reset to 00h by hardware reset.

**6.15.1.3 RTC Month Alarm Offset (RTC\_MONA\_OFFSET)**

MSR Address 51400056h  
 Type R/W  
 Reset Value 00h

**RTC\_MONA\_OFFSET Register Map**

7	6	5	4	3	2	1	0
RSVD	MONA_OFST						

**RTC\_MONA\_OFFSET Bit Descriptions**

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write as 0.
6:0	MONA_OFST	<b>Month Alarm Register Offset Value.</b> This register sets the location in RAM space of the Month Alarm register. This register must be programmed after a hardware reset, otherwise the Month Alarm register will be on top of other RAM data. It is programmed as an offset from 0. Reset to 00h by hardware reset.



**6.15.1.4 RTC Century Offset (RTC\_CEN\_OFFSET)**

MSR Address 51400057h  
 Type R/W  
 Reset Value 00h

**RTC\_CEN\_OFFSET Register Map**

7	6	5	4	3	2	1	0
RSVD		CEN_OFST					

**RTC\_CEN\_OFFSET Bit Descriptions**

Bit	Name	Description
7	RSVD	<b>Reserved.</b> Write as 0.
6:0	CEN_OSFT	<b>Century Register Offset Value.</b> This register sets the location in RAM space of the Century register. This register must be programmed after a hardware reset, otherwise the Century register will be on top of other RAM data. It is programmed as an offset from 0. Reset to 00h by hardware reset.

**6.15.2 RTC Native Registers****6.15.2.1 Seconds (RTC\_SEC)**

I/O Address 00h  
 Type R/W  
 Reset Value 00h

**RTC\_SEC Register Map**

7	6	5	4	3	2	1	0
SEC_DATA							

**RTC\_SEC Bit Descriptions**

Bit	Name	Description
7:0	SEC_DATA	<b>Seconds Data.</b> Values may be 00 to 59 in BCD format or 00 to 3B in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.2 Seconds Alarm (RTC\_SECA)**

I/O Address 01h  
 Type R/W  
 Reset Value 00h

**RTC\_SECA Register Map**

7	6	5	4	3	2	1	0
SECA_DATA							

**RTC\_SECA Bit Descriptions**

Bit	Name	Description
7:0	SECA_DATA	<b>Seconds Alarm Data.</b> Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one, an unconditional match is selected. Reset by $V_{PP}$ power-up.

**6.15.2.3 Minutes (RTC\_MIN)**

I/O Address 02h  
 Type R/W  
 Reset Value 00h

**RTC\_MIN Register Map**

7	6	5	4	3	2	1	0
MIN_DATA							

**RTC\_MIN Bit Descriptions**

Bit	Name	Description
7:0	MIN_DATA	<b>Minutes Data.</b> Values may be 00 to 59 in BCD format or 00 to 3B in binary format. Reset by $V_{PP}$ power-up.

**6.15.2.4 Minutes Alarm (RTC\_MINA)**

I/O Address 03h  
 Type R/W  
 Reset Value 00h

**RTC\_MINA Register Map**

7	6	5	4	3	2	1	0
MINA_DATA							

**RTC\_MINA Bit Descriptions**

Bit	Name	Description
7:0	MINA_DATA	<b>Minutes Alarm Data.</b> Values may be 00 to 59 in BCD format or 00 to 3B in binary format. When bits 7 and 6 are both set to one, an unconditional match is selected. Reset by $V_{PP}$ power-up.

**6.15.2.5 Hours (RTC\_HR)**

I/O Address 04h  
 Type R/W  
 Reset Value 00h

**RTC\_HR Register Map**

7	6	5	4	3	2	1	0
HR_DATA							

**RTC\_HR Bit Descriptions**

Bit	Name	Description
7:0	HR_DATA	<b>Hours Data.</b> For 12-hour mode, values can be 01 to 12 (AM) and 81 to 92 (PM) in BCD format, or 01 to 0C (AM) and 81 to 8C (PM) in binary format. For 24-hour mode, values can be 0 to 23 in BCD format or 00 to 17 in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.6 Hours Alarm**

I/O Address 05h  
 Type R/W  
 Reset Value 00h

**RTC\_HRA Register Map**

7	6	5	4	3	2	1	0
HRA_DATA							

**RTC\_HRA Bit Descriptions**

Bit	Name	Description
7:0	HRA_DATA	<b>Hours Alarm Data.</b> For 12-hour mode, values can be 01 to 12 (AM) and 81 to 92 (PM) in BCD format, or 01 to 0C (AM) and 81 to 8C (PM) in binary format. For 24-hour mode, values can be 0 to 23 in BCD format or 00 to 17 in binary format. When bits 7 and 6 are both set to one, unconditional match is selected. Reset by V <sub>PP</sub> power-up.

**6.15.2.7 Day of Week (RTC\_DOW)**

I/O Address 06h  
 Type R/W  
 Reset Value 00h

**RTC\_DOW Register Map**

7	6	5	4	3	2	1	0
DOW_DATA							

**RTC\_DOW Bit Descriptions**

Bit	Name	Description
7:0	DOW_DATA	<b>Day of Week Data.</b> Values may be 01 to 07 in BCD format or 01 to 07 in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.8 Day of Month (RTC\_DOM)**

I/O Address 07h  
 Type R/W  
 Reset Value 00h

**RTC\_DOM Register Map**

7	6	5	4	3	2	1	0
DOM_DATA							

**RTC\_DOM Bit Descriptions**

Bit	Name	Description
7:0	DOW_DATA	<b>Day of Month Data.</b> Values may be 01 to 31 in BCD format or 01 to 0F in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.9 Month (RTC\_MONTH)**

I/O Address 08h  
 Type R/W  
 Reset Value 00h

**RTC\_MON Register Map**

7	6	5	4	3	2	1	0
MON_DATA							

**RTC\_MON Bit Descriptions**

Bit	Name	Description
7:0	MON_DATA	<b>Month Data.</b> Values may be 01 to 12 in BCD format or 01 to 0C in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.10 Year (RTC\_YEAR)**

I/O Address 09h  
 Type R/W  
 Reset Value 00h

**RTC\_YR Register Map**

7	6	5	4	3	2	1	0
YR_DATA							

**RTC\_YR Bit Descriptions**

Bit	Name	Description
7:0	YR_DATA	<b>Year Data.</b> This register holds the two least significant digits of a four-digit year. For example, if the year is 2007, this register would contain the equivalent of '07'. Values may be 00 to 99 in BCD format or 00 to 63 in binary format. Reset by V <sub>PP</sub> power-up.

**6.15.2.11 RTC Control Register A (RTC\_CRA)**

I/O Address      0Ah  
 Type              R/W  
 Reset Value      20h

This register controls test selection among other functions. This register cannot be written before reading bit 7 of RTC\_CRD (VRT bit).

**RTC\_CRA Register Map**

7	6	5	4	3	2	1	0
UIP	DIV_CHN_CTL			PIR_SEL			

**RTC\_CRA Bit Descriptions**

Bit	Name	Description																
7	UIP (RO)	<p><b>Update in Progress (Read Only).</b> This RO bit is not affected by reset. This bit reads 0 when bit 7 of RTC_CRB is 1.</p> <p>0: Timing registers not updated within 244 <math>\mu</math>s.            1: Timing registers updated within 244 <math>\mu</math>s.</p>																
6:4	DIV_CHN_CTL	<p><b>Divider Chain Control.</b> These R/W bits control the configuration of the divider chain for timing generation and register bank selection. They are cleared to 010 as long as bit 7 of RTC_CRD (VRT bit) is 0.</p> <p>00x: Oscillator Disabled.                      10x: Test.            010: Normal Operation.                      11x: Divider Chain Reset.            011: Test.</p>																
3:0	PIR_SEL	<p><b>Periodic Interrupt Rate Select:</b> These R/W bits select one of fifteen output taps from the clock divider chain to control the rate of the periodic interrupt. They are cleared to 000 as long as bit 7 of RTC_CRD (VRT bit) is 0.</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">0000: No interrupts</td> <td style="width: 50%;">1000: 3.906250 ms</td> </tr> <tr> <td>0001: 3.906250 ms</td> <td>1001: 7.812500 ms</td> </tr> <tr> <td>0010: 7.812500 ms</td> <td>1010: 15.625000 ms</td> </tr> <tr> <td>0011: 0.122070 ms</td> <td>1011: 31.250000 ms</td> </tr> <tr> <td>0100: 0.244141 ms</td> <td>1100: 62.500000 ms</td> </tr> <tr> <td>0101: 0.488281 ms</td> <td>1101: 125.000000 ms</td> </tr> <tr> <td>0110: 0.976562 ms</td> <td>1110: 250.000000 ms</td> </tr> <tr> <td>0111: 1.953125 ms</td> <td>1111: 500.000000 ms</td> </tr> </table>	0000: No interrupts	1000: 3.906250 ms	0001: 3.906250 ms	1001: 7.812500 ms	0010: 7.812500 ms	1010: 15.625000 ms	0011: 0.122070 ms	1011: 31.250000 ms	0100: 0.244141 ms	1100: 62.500000 ms	0101: 0.488281 ms	1101: 125.000000 ms	0110: 0.976562 ms	1110: 250.000000 ms	0111: 1.953125 ms	1111: 500.000000 ms
0000: No interrupts	1000: 3.906250 ms																	
0001: 3.906250 ms	1001: 7.812500 ms																	
0010: 7.812500 ms	1010: 15.625000 ms																	
0011: 0.122070 ms	1011: 31.250000 ms																	
0100: 0.244141 ms	1100: 62.500000 ms																	
0101: 0.488281 ms	1101: 125.000000 ms																	
0110: 0.976562 ms	1110: 250.000000 ms																	
0111: 1.953125 ms	1111: 500.000000 ms																	

**6.15.2.12 RTC Control Register B (RTC\_CRB)**

I/O Address 0Bh  
 Type R/W  
 Reset Value 00h

**RTC\_CRB Register Map**

7	6	5	4	3	2	1	0
SET_MODE	PI_EN	AI_EN	UEI_EN	RSVD	DATA_MODE	HR_MODE	DAY_SAVE

**RTC\_CRB Bit Descriptions**

Bit	Name	Description
7	SET_MODE	<b>Set Mode.</b> This bit is reset at V <sub>PP</sub> power-up reset only. 0: Timing updates occur normally 1: User copy of time is “frozen”, allowing the time registers to be accessed whether or not an update occurs.
6	PI_EN	<b>Periodic Interrupts Enable.</b> Bits [3:0] of RTC_CRA (PIR_SEL) determine the rate at which this interrupt is generated. It is cleared to 0 on an RTC reset (i.e., hardware reset) or when the RTC is disable. 0: Disable. 1: Enable.
5	AI_EN	<b>Alarm Interrupt Enable.</b> This interrupt is generated immediately after a time update in which the seconds, minutes, hours, date and month time equal their respective alarm counterparts. It is cleared to 0 as long as bit 7 of RTC_CRD (VRT bit) reads 0. 0: Disable. 1: Enable.
4	UEI_EN	<b>Update Ended Interrupts Enable.</b> This interrupt is generated when an update occurs. It is cleared to 0 on an RTC reset (i.e., hardware reset) or when the RTC is disabled. 0: Disable. 1: Enable.
3	RSVD	<b>Reserved.</b> This bit is defined as “Square Wave Enable” by the MC146818 and is not supported by the RTC. This bit is always read as 0.
2	DATA_MODE	<b>Data Mode.</b> Selects data mode. This bit is reset at V <sub>PP</sub> power-up reset only. 0: BCD format. 1: Binary format.
1	HR_MODE	<b>Hour Mode.</b> Selects hour mode. This bit is reset at V <sub>PP</sub> power-up reset only. 0: 12-hour format. 1: 24-hour format.
0	DAY_SAVE	<b>Daylight Saving.</b> Enables/disables daylight savings mode. This bit is reset at V <sub>PP</sub> power-up reset only. 0: Disable. 1: Enable.  In the spring, time advances from 1:59:59 AM to 3:00:00 AM on the first Sunday in April.  In the fall, time returns from 1:59:59 AM to 1:00:00 AM on the last Sunday in October.

**6.15.2.13 RTC Control Register C (RTC\_CRC)**

I/O Address      0Ch  
 Type              RO  
 Reset Value      00h

**RTC\_CRC Register Map**

<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
IRQF	PF	AF	UF	RSVD			

**RTC\_CRC Bit Descriptions**

Bit	Name	Description
7	IRQF (RO)	<b>IRQ Flag (Read Only).</b> This RO bit mirrors the value on the interrupt output signal. When interrupt is active, IRQF is 1. Reading this register clears this bit (and deactivates the interrupt pin) and clears the flag bits UF, AF, and PF.  0: IRQ inactive. 1: Logic equation is true: ((UIE and UF) or (AIE and AF) or (PIE and PF)).
6	PF (RO)	<b>Periodic Interrupts Flag (Read Only).</b> This RO bit is cleared to 0 on an RTC reset (i.e., hardware reset) or when the RTC is disabled. In addition, this bit is cleared to 0 when this register is read.  0: No transition occurred on the selected tap since the last read. 1: Transition occurred on the selected tap of the divider chain.
5	AF (RO)	<b>Alarm Interrupt Flag (Read Only).</b> This RO bit is cleared to 0 as long as bit 7 of RTC_CRD (VRT bit) is 0. In addition, this bit is cleared to 0 when this register is read.  0: No alarm detected since the last read. 1: Alarm condition detected.
4	UF (RO)	<b>Update Ended Interrupts Flag (Read Only).</b> This RO bit is cleared to 0 on an RTC reset (i.e., hardware reset) or when the RTC is disabled. In addition, this bit is cleared to 0 when this register is read.  0: No update occurred since the last read. 1: Time registers updated.
3:0	RSVD (RO)	<b>Reserved (Read Only).</b> Reads as 0.

**6.15.2.14 RTC Control Register D (RTC\_CRD)**

I/O Address 0Dh  
 Type RO  
 Reset Value 00h

**RTC\_CRD Register Map**

7	6	5	4	3	2	1	0
VRT		RSVD					

**RTC\_CRD Bit Descriptions**

Bit	Name	Description
7	VRT (RO)	<p><b>Valid RAM and Time (Read Only).</b> This bit senses the voltage that feeds the RTC (VSB or VBAT) and indicates whether or not it was too low since the last time this bit was read. If it was too low, the RTC contents (time/calendar registers and CMOS RAM) is not valid. It is clear on <math>V_{PP}</math> power-up.</p> <p>0: The voltage that feeds the RTC was too low.            1: RTC contents (time/calendar registers and CMOS RAM) valid.</p>
6:0	RSVD (RO)	<b>Reserved (Read Only).</b> Reads as 0.

**6.15.2.15 Date of Month Alarm (RTC\_DOMA)**

I/O Address Programmable  
 Type R/W  
 Reset Value 00h

**RTC\_DOMA Register Map**

7	6	5	4	3	2	1	0
DOMA_DATA							

**RTC\_DOMA Bit Descriptions**

Bit	Name	Description
7:0	DOMA_DATA	<p><b>Date of Month Alarm Data.</b> Values may be 01 to 31 in BCD format or 01 to 1F in binary format. When bits 7 and 6 are both set to one, an unconditional match is selected (default). Reset by <math>V_{PP}</math> power-up.</p>



**6.15.2.16 Month Alarm (RTC\_MONA)**

I/O Address Programmable  
 Type R/W  
 Reset Value 00h

**RTC\_MONA Register Map**

7	6	5	4	3	2	1	0
MONA_DATA							

**RTC\_MONA Bit Descriptions**

Bit	Name	Description
7:0	MONA_DATA	<b>Month Alarm Data.</b> Values may be 01 to 12 in BCD format or 01 to 0C in binary format. When bits 7 and 6 are both set to one, an unconditional match is selected (default). Reset by $V_{PP}$ power-up.

**6.15.2.17 Century (RTC\_CEN)**

I/O Address Programmable  
 Type R/W  
 Reset Value 00h

**RTC\_CEN Register Map**

7	6	5	4	3	2	1	0
CEN_DATA							

**RTC\_CEN Bit Descriptions**

Bit	Name	Description
7:0	CEN_DATA	<b>Century Data.</b> This register holds the two most significant digits of a four-digit year. For example, if the year is 2008, this register would contain the equivalent of '20'. Values may be 00 to 99 in BCD format or 00 to 63 in binary format. Reset by $V_{PP}$ power-up.

## 6.16 GPIO Subsystem Register Descriptions

The registers for the General Purpose Input Output (GPIO) are divided into two sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- GPIO Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The GPIO Native registers are accessed via a Base Address Register, MSR\_LBAR\_GPIO (MSR 5140000Ch), as I/O Offsets. (See Section 6.6.2.5 on page 328 for bit descriptions of the Base Address Register.)

The Native registers associated with GPIO configuration are broadly divided into three categories:

- 1) GPIO Low/High Bank Feature Bit Registers.  
These registers (summarized in Table 6-57) control basic GPIO features. The Feature Bit registers use the atomic programming model except where noted. See

Section 6.16.1 "Atomic Bit Programming Model" on page 454 for details.

- 2) Input Conditioning Function Registers.  
These registers (summarized in Table 6-58 on page 452) are associated with the eight digital filter/event counter pairs that can be shared with the 32 GPIOs. These registers are not based on the atomic bit programming model.
- 3) GPIO Interrupt and PME Mapper Registers.  
These registers (summarized in Table 6-59 on page 453) are used for mapping any GPIO to one of the eight PIC-level interrupts or to one of the eight PME (Power Management Event) inputs.

The reference column in the summary tables point to the page where the detailed register maps and bit descriptions are listed. The *Low Bank* refers to GPIO[15:0] while the *High Bank* refers to GPIO[31:16]

**Note:** All register bits dealing with GPIO31, GPIO30, GPIO29 and GPIO23 are reserved.

**Table 6-57. GPIO Low/High Bank Feature Bit Registers Summary**

GPIO I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
<b>GPIO Low Bank Feature Bit Registers</b>					
00h	R/W	32	GPIO Low Bank Output Value (GPIOL_OUT_VAL)	FFFF0000h	Page 456
04h	R/W	32	GPIO Low Bank Output Enable (GPIOL_OUT_EN)	FFFF0000h	Page 456
08h	R/W	32	GPIO Low Bank Output Open-Drain Enable (GPIOL_OUT_OD_EN)	FFFF0000h	Page 457
0Ch	R/W	32	GPIO Low Bank Output Invert Enable (GPIOL_OUT_INVRT_EN)	FFFF0000h	Page 457
10h	R/W	32	GPIO Low Bank Output Auxiliary 1 Select (GPIOL_OUT_AUX1_SEL)	FFFF0000h	Page 458
14h	R/W	32	GPIO Low Bank Output Auxiliary 2 Select (GPIOL_OUT_AUX2_SEL)	FFFF0000h	Page 459
18h	R/W	32	GPIO Low Bank Pull-Up Enable (GPIOL_PU_EN)	1000EFFFh	Page 459
1Ch	R/W	32	GPIO Low Bank Pull-Down Enable (GPIOL_PD_EN)	EFFF1000h	Page 460
20h	R/W	32	GPIO Low Bank Input Enable (GPIOL_IN_EN)	FFFF0000h	Page 460
24h	R/W	32	GPIO Low Bank Input Invert Enable (GPIOL_IN_INVRT_EN)	FFFF0000h	Page 461
28h	R/W	32	GPIO Low Bank Input Filter Enable (GPIOL_IN_FLTR_EN)	FFFF0000h	Page 461
2Ch	R/W	32	GPIO Low Bank Input Event Count Enable (GPIOL_IN_EVNTCNT_EN)	FFFF0000h	Page 462
30h (Note 1)	RO	32	GPIO Low Bank Read Back (GPIOL_READ_BACK)	00000000h	Page 467
34h	R/W	32	GPIO Low Bank Input Auxiliary 1 Select (GPIOL_IN_AUX1_SEL)	FFFF0000h	Page 463

Table 6-57. GPIO Low/High Bank Feature Bit Registers Summary (Continued)

GPIO I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
38h	R/W	32	GPIO Low Bank Events Enable (GPIOL_EVNT_EN)	FFFF0000h	Page 463
3Ch (Note 1)	R/W	32	GPIO Low Bank Lock Enable (GPIOL_LOCK_EN)	00000000h	Page 469
40h	R/W	32	GPIO Low Bank Input Positive Edge Enable (GPIOL_IN_POSEDGE_EN)	FFFF0000h	Page 464
44h	R/W	32	GPIO Low Bank Input Negative Edge Enable (GPIOL_IN_NEGEDGE_EN)	FFFF0000h	Page 464
48h	R/W	32	GPIO Low Bank Input Positive Edge Status (GPIOL_IN_POSEDGE_STS)	FFFF0000h	Page 465
4Ch	R/W	32	GPIO Low Bank Input Negative Edge Status (GPIOL_IN_NEGEDGE_STS)	FFFF0000h	Page 466
<b>GPIO High Bank Feature Bit Registers</b>					
80h	R/W	32	GPIO High Bank Output Value (GPIOH_OUT_VAL)	FFFF0000h	Page 456
84h	R/W	32	GPIO High Bank Output Enable (GPIOH_OUT_EN)	FFFF0000h	Page 456
88h	R/W	32	GPIO High Bank Output Open-Drain Enable (GPIOH_OUT_OD_EN)	FFFF0000h	Page 457
8Ch	R/W	32	GPIO High Bank Output Invert Enable (GPIOH_OUT_INVRT_EN)	FFFF0000h	Page 457
90h	R/W	32	GPIO High Bank Output Auxiliary 1 Select (GPIOH_OUT_AUX1_SEL)	FFFF0000h	Page 458
94h	R/W	32	GPIO High Bank Output Auxiliary 2 Select (GPIOH_OUT_AUX2_SEL)	FFFF0000h	Page 459
98h	R/W	32	GPIO High Bank Pull-Up Enable (GPIOH_PU_EN)	0000FFFFh	Page 459
9Ch	R/W	32	GPIO High Bank Pull-Down Enable (GPIOH_PD_EN)	FFFF0000h	Page 460
A0h	R/W	32	GPIO High Bank Input Enable (GPIOH_IN_EN)	EFFF1000h	Page 460
A4h	R/W	32	GPIO High Bank Input Invert Enable (GPIOH_IN_INV_EN)	FFFF0000h	Page 461
A8h	R/W	32	GPIO High Bank Input Filter Enable (GPIOH_IN_FLTER_EN)	FFFF0000h	Page 461
ACh	R/W	32	GPIO High Bank Input Event Count Enable (GPIOH_IN_EVNTCNT_EN)	FFFF0000h	Page 462
B0h (Note 1)	RO	32	GPIO High Bank Read Back (GPIOH_READ_BACK)	00000000h	Page 468
B4h	R/W	32	GPIO High Bank Input Auxiliary 1 Select (GPIOH_IN_AUX1_SEL)	EFFF1000h	Page 463
B8h	R/W	32	GPIO High Bank Events Enable (GPIOH_EVNT_EN)	FFFF0000h	Page 463
BCh (Note 1)	R/W	32	GPIO High Bank Lock Enable (GPIOH_LOCK_EN)	00000000h	Page 470
C0h	R/W	32	GPIO High Bank Input Positive Edge Enable (GPIOH_IN_POSEDGE_EN)	FFFF0000h	Page 464
C4h	R/W	32	GPIO High Bank Input Negative Edge Enable (GPIOH_IN_NEGEDGE_EN)	FFFF0000h	Page 464

**Table 6-57. GPIO Low/High Bank Feature Bit Registers Summary (Continued)**

GPIO I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
C8h	R/W	32	GPIO High Bank Input Positive Edge Status (GPIOH_IN_POSEDGE_STS)	FFFF0000h	Page 465
CCh	R/W	32	GPIO High Bank Input Negative Edge Status (GPIOH_IN_NEGEDGE_STS)	FFFF0000h	Page 466

Note 1. The GPIO[x]\_READ\_BACK and GPIO[x]\_LOCK\_EN registers are not based on the atomic programming model (i.e., only one bit for control as opposed to two bits). See Section 6.16.1 "Atomic Bit Programming Model" on page 454 for more information on atomic programming.

**Table 6-58. GPIO Input Conditioning Function Registers Summary**

GPIO I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
50h	R/W	16	GPIO Filter 0 Amount (GPIO_FLTR0_AMNT)	0000h	Page 471
52h	R/W	16	GPIO Filter 0 Count (GPIO_FLTR0_CNT)	0000h	Page 472
54h	R/W	16	GPIO Event Counter 0 (GPIO_EVNTCNT0)	0000h	Page 473
56h	R/W	16	GPIO Event Counter 0 Compare Value (GPIO_EVNTCNT0_COMP)	0000h	Page 474
58h	R/W	16	GPIO Filter 1 Amount (GPIO_FLTR1_AMNT)	0000h	Page 471
5Ah	R/W	16	GPIO Filter 1 Count (GPIO_FLTR1_CNT)	0000h	Page 472
5Ch	R/W	16	GPIO Event Counter 1 (GPIO_EVNTCNT1)	0000h	Page 473
5Eh	R/W	16	GPIO Event Counter 1 Compare Value (GPIO_EVNTCNT1_COMP)	0000h	Page 474
60h	R/W	16	GPIO Filter 2 Amount (GPIO_FLTR2_AMNT)	0000h	Page 471
62h	R/W	16	GPIO Filter 2 Count (GPIO_FLTR2_CNT)	0000h	Page 472
64h	R/W	16	GPIO Event Counter 2 (GPIO_EVNTCNT2)	0000h	Page 473
66h	R/W	16	GPIO Event Counter 2 Compare Value (GPIO_EVNTCNT2_COMP)	0000h	Page 474
68h	R/W	16	GPIO Filter 3 Amount (GPIO_FLTR3_AMNT)	0000h	Page 471
6Ah	R/W	16	GPIO Filter 3 Count (GPIO_FLTR3_CNT)	0000h	Page 472
6Ch	R/W	16	GPIO Event Counter 3 (GPIO_EVNTCNT3)	0000h	Page 473
6Eh	R/W	16	GPIO Event Counter 3 Compare Value (GPIO_EVNTCNT3_COMP)	0000h	Page 474
70h	R/W	16	GPIO Filter 4 Amount (GPIO_FLTR4_AMNT)	0000h	Page 471
72h	R/W	16	GPIO Filter 4 Count (GPIO_FLTR4_CNT)	0000h	Page 472
74h	R/W	16	GPIO Event Counter 4 (GPIO_EVNTCNT4)	0000h	Page 473
76h	R/W	16	GPIO Event Counter 4 Compare Value (GPIO_EVNTCNT4_COMP)	0000h	Page 474
78h	R/W	16	GPIO Filter 5 Amount (GPIO_FLTR5_AMNT)	0000h	Page 471
7Ah	R/W	16	GPIO Filter 5 Count (GPIO_FLTR5_CNT)	0000h	Page 472
7Ch	R/W	16	GPIO Event Counter 5 (GPIO_EVNTCNT5)	0000h	Page 473
7Eh	R/W	16	GPIO Event Counter 5 Compare Value (GPIO_EVNTCNT5_COMP)	0000h	Page 474

**Table 6-58. GPIO Input Conditioning Function Registers Summary (Continued)**

GPIO I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
D0h	R/W	16	GPIO Filter 6 Amount (GPIO_FLTR6_AMNT)	0000h	Page 471
D2h	R/W	16	GPIO Filter 6 Count (GPIO_FLTR6_CNT)	0000h	Page 472
D4h	R/W	16	GPIO Event Counter 6 (GPIO_EVNTCNT6)	0000h	Page 473
D6h	R/W	16	GPIO Event Counter 6 Compare Value (GPIO_EVNTCNT6_COMP)	0000h	Page 474
D8h	R/W	16	GPIO Filter 7 Amount (GPIO_FLTR7_AMNT)	0000h	Page 471
DAh	R/W	16	GPIO Filter 7 Count (GPIO_FLTR7_CNT)	0000h	Page 472
DCh	R/W	16	GPIO Event Counter 7 (GPIO_EVNTCNT7)	0000h	Page 473
DEh	R/W	16	GPIO Event Counter 7 Compare Value (GPIO_EVNTCNT7_COMP)	0000h	Page 474
F0h	R/W	8	GPIO Filter/Event Pair 0 Selection (GPIO_FE0_SEL)	00h	Page 475
F1h	R/W	8	GPIO Filter/Event Pair 1 Selection (GPIO_FE1_SEL)	00h	Page 475
F2h	R/W	8	GPIO Filter/Event Pair 2 Selection (GPIO_FE2_SEL)	00h	Page 475
F3h	R/W	8	GPIO Filter/Event Pair 3 Selection (GPIO_FE3_SEL)	00h	Page 475
F4h	R/W	8	GPIO Filter/Event Pair 4 Selection (GPIO_FE4_SEL)	00h	Page 475
F5h	R/W	8	GPIO Filter/Event Pair 5 Selection (GPIO_FE5_SEL)	00h	Page 475
F6h	R/W	8	GPIO Filter/Event Pair 6 Selection (GPIO_FE6_SEL)	00h	Page 475
F7h	R/W	8	GPIO Filter/Event Pair 7 Selection (GPIO_FE7_SEL)	00h	Page 475
F8h	R/W	32	GPIO Low Bank Event Counter Decrement (GPIOL_EVNTCNT_DEC)	00000000h	Page 477
FCh	R/W	32	GPIO High Bank Event Counter Decrement (GPIOL_EVNTCNT_DEC)	00000000h	Page 478

**Table 6-59. GPIO Interrupt and PME Mapper Registers Summary**

GPIO Address	Type	Width (Bits)	Register Name	Reset Value	Reference
E0h	R/W	32	GPIO Mapper X (GPIO_MAP_X)	00000000h	Page 482
E4h	R/W	32	GPIO Mapper Y (GPIO_MAP_Y)	00000000h	Page 481
E8h	R/W	32	GPIO Mapper Z (GPIO_MAP_Z)	00000000h	Page 480
ECh	R/W	32	GPIO Mapper W (GPIO_MAP_W)	00000000h	Page 479

### 6.16.1 Atomic Bit Programming Model

The registers in Section 6.16.2 "GPIO Low/High Bank Feature Bit Registers", starting on page 456, that are referred to as "atomic" all follow the same programming model (i.e., work exactly the same way) but each controls a different GPIO feature. Two data bits are used to control each GPIO Feature bit, each pair of bits operate in an exclusive-OR pattern. Refer to Section 5.15.2 "Register Strategy" on page 161 for further details.

The Low Bank registers control programming of GPIO15 through GPIO0 and the High Bank registers program GPIO31 through GPIO16. The tables that follow provide the register/bit formats for the Low and High Bank GPIO feature configuration registers.

**Note:** All register bits dealing with GPIO31, GPIO30, GPIO29 and GPIO23 are Reserved .

**Table 6-60. Low Bank Atomic Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0	GPIO15	GPIO14	GPIO13	GPIO12	GPIO11	GPIO10	GPIO9	GPIO8	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

**Table 6-61. Low Bank Atomic Bit Descriptions**

Bit	Name	Description
31,15	GPIO15	<b>GPIO15 Feature.</b> 00: No change. 01: Feature bit = 1. 10: Feature bit = 0. 11: No change.
30,14	GPIO14	<b>GPIO14 Feature.</b> See bits [31,15] for decode.
29,13	GPIO13	<b>GPIO13 Feature.</b> See bits [31,15] for decode.
28,12	GPIO12	<b>GPIO12 Feature.</b> See bits [31,15] for decode.
27,11	GPIO11	<b>GPIO11 Feature.</b> See bits [31,15] for decode.
26,10	GPIO10	<b>GPIO10 Feature.</b> See bits [31,15] for decode.
25,9	GPIO9	<b>GPIO9 Feature.</b> See bits [31,15] for decode.
24,8	GPIO8	<b>GPIO8 Feature.</b> See bits [31,15] for decode.
23,7	GPIO7	<b>GPIO7 Feature.</b> See bits [31,15] for decode.
22,6	GPIO6	<b>GPIO6 Feature.</b> See bits [31,15] for decode.
21,5	GPIO5	<b>GPIO5 Feature.</b> See bits [31,15] for decode.
20,4	GPIO4	<b>GPIO4 Feature.</b> See bits [31,15] for decode.
19,3	GPIO3	<b>GPIO3 Feature.</b> See bits [31,15] for decode.
18,2	GPIO2	<b>GPIO2 Feature.</b> See bits [31,15] for decode.
17,1	GPIO1	<b>GPIO1 Feature.</b> See bits [31,15] for decode.
16,0	GPIO0	<b>GPIO0 Feature.</b> See bits [31,15] for decode.

Table 6-62. High Bank Atomic Register Map Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	RSVD	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16	RSVD	RSVD	RSVD	GPIO28	GPIO27	GPIO26	GPIO25	GPIO24	RSVD	GPIO22	GPIO21	GPIO20	GPIO19	GPIO18	GPIO17	GPIO16

Table 6-63. High Bank Atomic Bit Descriptions Format

Bit	Name	Description
31,15	RSVD	<b>Reserved.</b> Write as 00.
30,14	RSVD	<b>Reserved.</b> Write as 00.
29,13	RSVD	<b>Reserved.</b> Write as 00.
28,12	GPIO28	<b>GPIO28 Feature.</b> 00: No change. 01: Feature bit = 1. 10: Feature bit = 0. 11: No change.
27,11	GPIO27	<b>GPIO27 Feature.</b> See bits [28,12] for decode.
26,10	GPIO26	<b>GPIO26 Feature.</b> See bits [28,12] for decode.
25,9	GPIO25	<b>GPIO25 Feature.</b> See bits [28,12] for decode.
24,8	GPIO24	<b>GPIO24 Feature.</b> See bits [28,12] for decode.
23,7	RSVD	<b>Reserved.</b> Write as 00.
22,6	GPIO22	<b>GPIO22 Feature.</b> See bits [28,12] for decode.
21,5	GPIO21	<b>GPIO21 Feature.</b> See bits [28,12] for decode.
20,4	GPIO20	<b>GPIO20 Feature.</b> See bits [28,12] for decode.
19,3	GPIO19	<b>GPIO19 Feature.</b> See bits [28,12] for decode.
18,2	GPIO18	<b>GPIO18 Feature.</b> See bits [28,12] for decode.
17,1	GPIO17	<b>GPIO17 Feature.</b> See bits [28,12] for decode.
16,0	GPIO16	<b>GPIO16 Feature.</b> See bits [28,12] for decode.

**6.16.2 GPIO Low/High Bank Feature Bit Registers**

**6.16.2.1 GPIO Output Value (GPIO[x]\_OUT\_VAL)**

These registers control the output value for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the output value = 1. If the feature bit is low, the output value = 0. The reset value forces all the output values to be initially set to 0. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

If OUT\_AUX1 (GPIO I/O Offset 10h/90h) and/or OUT\_AUX2 (GPIO I/O Offset 14h/94h) are selected, then their value overrides the OUT\_VAL settings. See Table 3-8 "GPIO Options" on page 47 for AUX programming details.

**GPIO Low Bank Output Value (GPIOL\_OUT\_VAL)**

GPIO I/O Offset 00h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Output Value (GPIOH\_OUT\_VAL)**

GPIO I/O Offset 80h  
 Type R/W  
 Reset Value FFFF0000h

**GPIOL\_OUT\_VAL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_VAL_15	OUT_VAL_14	OUT_VAL_13	OUT_VAL_12	OUT_VAL_11	OUT_VAL_10	OUT_VAL_9	OUT_VAL_8	OUT_VAL_7	OUT_VAL_6	OUT_VAL_5	OUT_VAL_4	OUT_VAL_3	OUT_VAL_2	OUT_VAL_1	OUT_VAL_0	OUT_VAL_15	OUT_VAL_14	OUT_VAL_13	OUT_VAL_12	OUT_VAL_11	OUT_VAL_10	OUT_VAL_9	OUT_VAL_8	OUT_VAL_7	OUT_VAL_6	OUT_VAL_5	OUT_VAL_4	OUT_VAL_3	OUT_VAL_2	OUT_VAL_1	OUT_VAL_0

**GPIOH\_OUT\_VAL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_VAL_28	OUT_VAL_27	OUT_VAL_26	OUT_VAL_25	OUT_VAL_24	RSVD	OUT_VAL_22	OUT_VAL_21	OUT_VAL_20	OUT_VAL_19	OUT_VAL_18	OUT_VAL_17	OUT_VAL_16	RSVD	RSVD	RSVD	OUT_VAL_28	OUT_VAL_27	OUT_VAL_26	OUT_VAL_25	OUT_VAL_24	RSVD	OUT_VAL_22	OUT_VAL_21	OUT_VAL_20	OUT_VAL_19	OUT_VAL_18	OUT_VAL_17	OUT_VAL_16

**6.16.2.2 GPIO/OUT\_AUX Output Enable (GPIO[x]\_OUT\_EN)**

These registers control the output enable for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit high, the output is enabled. If the feature bit is low, the output is disabled. The reset value forces all the outputs to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Output Enable (GPIOL\_OUT\_EN)**

GPIO I/O Offset 04h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Output Enable (GPIOH\_OUT\_EN)**

GPIO I/O Offset 84h  
 Type R/W  
 Reset Value FFFF0000h

**GPIOL\_OUT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_EN_15	OUT_EN_14	OUT_EN_13	OUT_EN_12	OUT_EN_11	OUT_EN_10	OUT_EN_9	OUT_EN_8	OUT_EN_7	OUT_EN_6	OUT_EN_5	OUT_EN_4	OUT_EN_3	OUT_EN_2	OUT_EN_1	OUT_EN_0	OUT_EN_15	OUT_EN_14	OUT_EN_13	OUT_EN_12	OUT_EN_11	OUT_EN_10	OUT_EN_9	OUT_EN_8	OUT_EN_7	OUT_EN_6	OUT_EN_5	OUT_EN_4	OUT_EN_3	OUT_EN_2	OUT_EN_1	OUT_EN_0

**GPIOH\_OUT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_EN_28	OUT_EN_27	OUT_EN_26	OUT_EN_25	OUT_EN_24	RSVD	OUT_EN_22	OUT_EN_21	OUT_EN_20	OUT_EN_19	OUT_EN_18	OUT_EN_17	OUT_EN_16	RSVD	RSVD	RSVD	OUT_EN_28	OUT_EN_27	OUT_EN_26	OUT_EN_25	OUT_EN_24	RSVD	OUT_EN_22	OUT_EN_21	OUT_EN_20	OUT_EN_19	OUT_EN_18	OUT_EN_17	OUT_EN_16



**6.16.2.3 GPIO/OUT\_AUX Output Open-Drain Enable (GPIO[x]\_OUT\_OD\_EN)**

These registers control the open-drain enable of the output for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The open-drain is enabled if the feature bit is high. The open-drain is disabled if the feature bit is low. The reset value forces all open-drains on the outputs to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Output Open-Drain Enable (GPIO\_L\_OUT\_OD\_EN)**

GPIO I/O Offset 08h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Output Open-Drain Enable (GPIO\_H\_OUT\_OD\_EN)**

GPIO I/O Offset 88h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_OUT\_OD\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_OD_15	OUT_OD_14	OUT_OD_13	OUT_OD_12	OUT_OD_11	OUT_OD_10	OUT_OD_9	OUT_OD_8	OUT_OD_7	OUT_OD_6	OUT_OD_5	OUT_OD_4	OUT_OD_3	OUT_OD_2	OUT_OD_1	OUT_OD_0	OUT_OD_15	OUT_OD_14	OUT_OD_13	OUT_OD_12	OUT_OD_11	OUT_OD_10	OUT_OD_9	OUT_OD_8	OUT_OD_7	OUT_OD_6	OUT_OD_5	OUT_OD_4	OUT_OD_3	OUT_OD_2	OUT_OD_1	OUT_OD_0

**GPIO\_H\_OUT\_OD\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_OD_28	OUT_OD_27	OUT_OD_26	OUT_OD_25	OUT_OD_24	RSVD	OUT_OD_22	OUT_OD_21	OUT_OD_20	OUT_OD_19	OUT_OD_18	OUT_OD_17	OUT_OD_16	RSVD	RSVD	RSVD	OUT_OD_28	OUT_OD_27	OUT_OD_26	OUT_OD_25	OUT_OD_24	RSVD	OUT_OD_22	OUT_OD_21	OUT_OD_20	OUT_OD_19	OUT_OD_18	OUT_OD_17	OUT_OD_16

**6.16.2.4 GPIO/OUT\_AUX Output Invert Enable (GPIO[x]\_OUT\_INVRT\_EN)**

These registers control the output invert enable for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The output is inverted if the feature bit is high. The output is not inverted if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Output Invert Enable (GPIO\_L\_OUT\_INVRT\_EN)**

GPIO I/O Offset 0Ch  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Output Invert Enable (GPIO\_H\_OUT\_INVRT\_EN)**

GPIO I/O Offset 8Ch  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_OUT\_INVRT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_INVRT_15	OUT_INVRT_14	OUT_INVRT_13	OUT_INVRT_12	OUT_INVRT_11	OUT_INVRT_10	OUT_INVRT_9	OUT_INVRT_8	OUT_INVRT_7	OUT_INVRT_6	OUT_INVRT_5	OUT_INVRT_4	OUT_INVRT_3	OUT_INVRT_2	OUT_INVRT_1	OUT_INVRT_0	OUT_INVRT_15	OUT_INVRT_14	OUT_INVRT_13	OUT_INVRT_12	OUT_INVRT_11	OUT_INVRT_10	OUT_INVRT_9	OUT_INVRT_8	OUT_INVRT_7	OUT_INVRT_6	OUT_INVRT_5	OUT_INVRT_4	OUT_INVRT_3	OUT_INVRT_2	OUT_INVRT_1	OUT_INVRT_0

**GPIO\_H\_OUT\_INVRT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_INVRT_28	OUT_INVRT_27	OUT_INVRT_26	OUT_INVRT_25	OUT_INVRT_24	RSVD	OUT_INVRT_22	OUT_INVRT_21	OUT_INVRT_20	OUT_INVRT_19	OUT_INVRT_18	OUT_INVRT_17	OUT_INVRT_16	RSVD	RSVD	RSVD	OUT_INVRT_28	OUT_INVRT_27	OUT_INVRT_26	OUT_INVRT_25	OUT_INVRT_24	RSVD	OUT_INVRT_22	OUT_INVRT_21	OUT_INVRT_20	OUT_INVRT_19	OUT_INVRT_18	OUT_INVRT_17	OUT_INVRT_16

**6.16.2.5 GPIO Output Auxiliary 1 Select (GPIO[x]\_OUT\_AUX1\_SEL)**

These registers select the Auxiliary 1 output of the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. Auxiliary 1 is selected as the output if the feature bit is high. Auxiliary 1 is not selected as the output if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

If OUT\_AUX1 and/or OUT\_AUX2 are selected, then their value overrides the OUT\_VAL (GPIO I/O Offset 00h/80h) settings. See Table 3-8 "GPIO Options" on page 47 for AUX programming details.

**GPIO Low Bank Output Auxiliary 1 Select (GPIO\_L\_OUT\_AUX1\_SEL)**

GPIO I/O Offset 10h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Output Auxiliary 1 Select (GPIO\_H\_OUT\_AUX1\_SEL)**

GPIO I/O Offset 90h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_OUT\_AUX1\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_AUX1_15	OUT_AUX1_14	OUT_AUX1_13	OUT_AUX1_12	OUT_AUX1_11	OUT_AUX1_10	OUT_AUX1_9	OUT_AUX1_8	OUT_AUX1_7	OUT_AUX1_6	OUT_AUX1_5	OUT_AUX1_4	OUT_AUX1_3	OUT_AUX1_2	OUT_AUX1_1	OUT_AUX1_0	OUT_AUX1_15	OUT_AUX1_14	OUT_AUX1_13	OUT_AUX1_12	OUT_AUX1_11	OUT_AUX1_10	OUT_AUX1_9	OUT_AUX1_8	OUT_AUX1_7	OUT_AUX1_6	OUT_AUX1_5	OUT_AUX1_4	OUT_AUX1_3	OUT_AUX1_2	OUT_AUX1_1	OUT_AUX1_0

**GPIO\_H\_OUT\_AUX1\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_AUX1_28	OUT_AUX1_27	OUT_AUX1_26	OUT_AUX1_25	OUT_AUX1_24	RSVD	OUT_AUX1_22	OUT_AUX1_21	OUT_AUX1_20	OUT_AUX1_19	OUT_AUX1_18	OUT_AUX1_17	OUT_AUX1_16	RSVD	RSVD	RSVD	OUT_AUX1_28	OUT_AUX1_27	OUT_AUX1_26	OUT_AUX1_25	OUT_AUX1_24	RSVD	OUT_AUX1_22	OUT_AUX1_21	OUT_AUX1_20	OUT_AUX1_19	OUT_AUX1_18	OUT_AUX1_17	OUT_AUX1_16

**6.16.2.6 GPIO Output Auxiliary 2 Select (GPIO[x]\_OUT\_AUX2\_SEL)**

These registers select the Auxiliary 2 output of the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. Auxiliary 2 is selected as the output if the feature bit is high. Auxiliary 2 is not selected as the output if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

If OUT\_AUX1 and/or OUT\_AUX2 are selected, then their value overrides the OUT\_VAL (GPIO I/O Offset 00h/80h) settings. See Table 3-8 "GPIO Options" on page 47 for AUX programming details.

**GPIO Low Bank Output Auxiliary 2 Select (GPIO\_L\_OUT\_AUX2\_SEL)**

GPIO I/O Offset 14h  
Type R/W  
Reset Value FFFF0000h

**GPIO High Bank Output Auxiliary 2 Select (GPIO\_H\_OUT\_AUX2\_SEL)**

GPIO I/O Offset 94h  
Type R/W  
Reset Value FFFF0000h

**GPIO\_L\_OUT\_AUX2\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_AUX2_15	OUT_AUX2_14	OUT_AUX2_13	OUT_AUX2_12	OUT_AUX2_11	OUT_AUX2_10	OUT_AUX2_9	OUT_AUX2_8	OUT_AUX2_7	OUT_AUX2_6	OUT_AUX2_5	OUT_AUX2_4	OUT_AUX2_3	OUT_AUX2_2	OUT_AUX2_1	OUT_AUX2_0	OUT_AUX2_15	OUT_AUX2_14	OUT_AUX2_13	OUT_AUX2_12	OUT_AUX2_11	OUT_AUX2_10	OUT_AUX2_9	OUT_AUX2_8	OUT_AUX2_7	OUT_AUX2_6	OUT_AUX2_5	OUT_AUX2_4	OUT_AUX2_3	OUT_AUX2_2	OUT_AUX2_1	OUT_AUX2_0

**GPIO\_H\_OUT\_AUX2\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	OUT_AUX2_28	OUT_AUX2_27	OUT_AUX2_26	OUT_AUX2_25	OUT_AUX2_24	RSVD	OUT_AUX2_22	OUT_AUX2_21	OUT_AUX2_20	OUT_AUX2_19	OUT_AUX2_18	OUT_AUX2_17	OUT_AUX2_16	RSVD	RSVD	RSVD	OUT_AUX2_28	OUT_AUX2_27	OUT_AUX2_26	OUT_AUX2_25	OUT_AUX2_24	RSVD	OUT_AUX2_22	OUT_AUX2_21	OUT_AUX2_20	OUT_AUX2_19	OUT_AUX2_18	OUT_AUX2_17	OUT_AUX2_16

**6.16.2.7 GPIO/AUX Pull-Up Enable (GPIO[x]\_PU\_EN)**

These registers control enabling of the pull-up on the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the pull-up is enabled. If the feature bit is low, the pull-up is disabled. The reset value forces all the pull-ups to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Pull-Up Enable (GPIO\_L\_PU\_EN)**

GPIO I/O Offset 18h  
Type R/W  
Reset Value 1000EFFFh

**GPIO High Bank Pull-Up Enable (GPIO\_H\_PU\_EN)**

GPIO I/O Offset 98h  
Type R/W  
Reset Value 0000FFFFh

**GPIO\_L\_PU\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PU_15	PU_14	PU_13	PU_12	PU_11	PU_10	PU_9	PU_8	PU_7	PU_6	PU_5	PU_4	PU_3	PU_2	PU_1	PU_0	PU_15	PU_14	PU_13	PU_12	PU_11	PU_10	PU_9	PU_8	PU_7	PU_6	PU_5	PU_4	PU_3	PU_2	PU_1	PU_0

For bits [6:5], see special disabling rules for the auto-sense GPIOs in Section 5.15.5.6 on page 165.

**GPIO\_H\_PU\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	PU_28	PU_27	PU_26	PU_25	PU_24	RSVD	PU_22	PU_21	PU_20	PU_19	PU_18	PU_17	PU_16	RSVD	RSVD	RSVD	PU_28	PU_27	PU_26	PU_25	PU_24	RSVD	PU_22	PU_21	PU_20	PU_19	PU_18	PU_17	PU_16

### 6.16.2.8 GPIO/AUX Pull-Down Enable (GPIO[x]\_PD\_EN)

These registers control enabling of the pull-down on the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the pull-down is enabled. If the feature bit is low, the pull-down is disabled. The reset value forces all the pull-downs to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

#### GPIO Low Bank Pull-Down Enable (GPIOL\_PD\_EN)

GPIO I/O Offset 1Ch  
Type R/W  
Reset Value EFFF1000h

#### GPIO High Bank Pull-Down Enable (GPIOH\_PD\_EN)

GPIO I/O Offset 9Ch  
Type R/W  
Reset Value FFFF0000h

#### GPIOL\_PD\_EN Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD_15	PD_14	PD_13	PD_12	PD_11	PD_10	PD_9	PD_8	PD_7	PD_6	PD_5	PD_4	PD_3	PD_2	PD_1	PD_0	PD_15	PD_14	PD_13	PD_12	PD_11	PD_10	PD_9	PD_8	PD_7	PD_6	PD_5	PD_4	PD_3	PD_2	PD_1	PD_0

For bits [6:5], see special disabling rules for the auto-sense GPIOs in Section 5.15.5.6 on page 165.

#### GPIOH\_PD\_EN Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	PD_28	PD_27	PD_26	PD_25	PD_24	RSVD	PD_22	PD_21	PD_20	PD_19	PD_18	PD_17	PD_16	RSVD	RSVD	RSVD	PD_28	PD_27	PD_26	PD_25	PD_24	RSVD	PD_22	PD_21	PD_20	PD_19	PD_18	PD_17	PD_16

### 6.16.2.9 GPIO/IN\_AUX Input Enable (GPIO[x]\_IN\_EN)

These registers control the input enable for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the input is enabled. If the feature bit is low, the input is disabled. The reset value forces all the inputs to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

#### GPIO Low Bank Input Enable (GPIOL\_IN\_EN)

GPIO I/O Offset 20h  
Type R/W  
Reset Value FFFF0000h

#### GPIO High Bank Input Enable (GPIOH\_IN\_EN)

GPIO I/O Offset A0h  
Type R/W  
Reset Value EFFF1000h

#### GPIOL\_IN\_EN Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_EN_15	IN_EN_14	IN_EN_13	IN_EN_12	IN_EN_11	IN_EN_10	IN_EN_9	IN_EN_8	IN_EN_7	IN_EN_6	IN_EN_5	IN_EN_4	IN_EN_3	IN_EN_2	IN_EN_1	IN_EN_0	IN_EN_15	IN_EN_14	IN_EN_13	IN_EN_12	IN_EN_11	IN_EN_10	IN_EN_9	IN_EN_8	IN_EN_7	IN_EN_6	IN_EN_5	IN_EN_4	IN_EN_3	IN_EN_2	IN_EN_1	IN_EN_0

#### GPIOH\_IN\_EN Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_EN_28	IN_EN_27	IN_EN_26	IN_EN_25	IN_EN_24	RSVD	IN_EN_22	IN_EN_21	IN_EN_20	IN_EN_19	IN_EN_18	IN_EN_17	IN_EN_16	RSVD	RSVD	RSVD	IN_EN_28	IN_EN_27	IN_EN_26	IN_EN_25	IN_EN_24	RSVD	IN_EN_22	IN_EN_21	IN_EN_20	IN_EN_19	IN_EN_18	IN_EN_17	IN_EN_16

**6.16.2.10 GPIO/IN\_AUX Input Invert Enable (GPIO[x]\_IN\_INVRT\_EN)**

These registers control the input invert enable for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The input is inverted if the feature bit is high. The input is not inverted if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Invert Enable (GPIO\_L\_IN\_INVRT\_EN)**

GPIO I/O Offset 24h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Input Invert Enable (GPIO\_H\_IN\_INVRT\_EN)**

GPIO I/O Offset A4h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_IN\_INVRT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_INVRT_15	IN_INVRT_14	IN_INVRT_13	IN_INVRT_12	IN_INVRT_11	IN_INVRT_10	IN_INVRT_9	IN_INVRT_8	IN_INVRT_7	IN_INVRT_6	IN_INVRT_5	IN_INVRT_4	IN_INVRT_3	IN_INVRT_2	IN_INVRT_1	IN_INVRT_0	IN_INVRT_15	IN_INVRT_14	IN_INVRT_13	IN_INVRT_12	IN_INVRT_11	IN_INVRT_10	IN_INVRT_9	IN_INVRT_8	IN_INVRT_7	IN_INVRT_6	IN_INVRT_5	IN_INVRT_4	IN_INVRT_3	IN_INVRT_2	IN_INVRT_1	IN_INVRT_0

**GPIO\_H\_IN\_INVRT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_INVRT_28	IN_INVRT_27	IN_INVRT_26	IN_INVRT_25	IN_INVRT_24	RSVD	IN_INVRT_22	IN_INVRT_21	IN_INVRT_20	IN_INVRT_19	IN_INVRT_18	IN_INVRT_17	IN_INVRT_16	RSVD	RSVD	RSVD	IN_INVRT_28	IN_INVRT_27	IN_INVRT_26	IN_INVRT_25	IN_INVRT_24	RSVD	IN_INVRT_22	IN_INVRT_21	IN_INVRT_20	IN_INVRT_19	IN_INVRT_18	IN_INVRT_17	IN_INVRT_16

**6.16.2.11 GPIO/IN\_AUX Input Filter Enable (GPIO[x]\_IN\_FLTR\_EN)**

These registers control the input filter function enable for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the filter function is enabled. If the feature bit is low, the filter function is disabled. The reset value forces all the filter functions to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Filter Enable (GPIO\_L\_IN\_FLTR\_EN)**

GPIO I/O Offset 28h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Input Filter Enable (GPIO\_H\_IN\_FLTR\_EN)**

GPIO I/O Offset A8h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_IN\_FLTR\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_FLTR_15	IN_FLTR_14	IN_FLTR_13	IN_FLTR_12	IN_FLTR_11	IN_FLTR_10	IN_FLTR_9	IN_FLTR_8	IN_FLTR_7	IN_FLTR_6	IN_FLTR_5	IN_FLTR_4	IN_FLTR_3	IN_FLTR_2	IN_FLTR_1	IN_FLTR_0	IN_FLTR_15	IN_FLTR_14	IN_FLTR_13	IN_FLTR_12	IN_FLTR_11	IN_FLTR_10	IN_FLTR_9	IN_FLTR_8	IN_FLTR_7	IN_FLTR_6	IN_FLTR_5	IN_FLTR_4	IN_FLTR_3	IN_FLTR_2	IN_FLTR_1	IN_FLTR_0

**GPIO\_H\_IN\_FLTR\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_FLTR_28	IN_FLTR_27	IN_FLTR_26	IN_FLTR_25	IN_FLTR_24	RSVD	IN_FLTR_22	IN_FLTR_21	IN_FLTR_20	IN_FLTR_19	IN_FLTR_18	IN_FLTR_17	IN_FLTR_16	RSVD	RSVD	RSVD	IN_FLTR_28	IN_FLTR_27	IN_FLTR_26	IN_FLTR_25	IN_FLTR_24	RSVD	IN_FLTR_22	IN_FLTR_21	IN_FLTR_20	IN_FLTR_19	IN_FLTR_18	IN_FLTR_17	IN_FLTR_16

**6.16.2.12 GPIO/IN\_AUX Input Event Count Enable (GPIO[x]\_IN\_EVNTCNT\_EN)**

These registers control the enabling of the input event counter function for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. If the feature bit is high, the event counter function is enabled on the input. If the feature bit is low, the event counter function is disabled on the input. The reset value forces all the filter functions to be disabled. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

When the event counter is enabled, the filter must also be enabled (GPIO I/O Offset 28h/A8h). If no filtering is desired, then program the GPIO\_FILTER[x]\_AMOUNT register to 0.

**GPIO Low Bank Input Event Count Enable (GPIO\_L\_IN\_EVNTCNT\_EN)**

GPIO I/O Offset 2Ch  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Input Event Count Enable (GPIO\_H\_IN\_EVNTCNT\_EN)**

GPIO I/O Offset ACh  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_IN\_EVNTCNT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_EVNTCNT_15	IN_EVNTCNT_14	IN_EVNTCNT_13	IN_EVNTCNT_12	IN_EVNTCNT_11	IN_EVNTCNT_10	IN_EVNTCNT_9	IN_EVNTCNT_8	IN_EVNTCNT_7	IN_EVNTCNT_6	IN_EVNTCNT_5	IN_EVNTCNT_4	IN_EVNTCNT_3	IN_EVNTCNT_2	IN_EVNTCNT_1	IN_EVNTCNT_0	IN_EVNTCNT_15	IN_EVNTCNT_14	IN_EVNTCNT_13	IN_EVNTCNT_12	IN_EVNTCNT_11	IN_EVNTCNT_10	IN_EVNTCNT_9	IN_EVNTCNT_8	IN_EVNTCNT_7	IN_EVNTCNT_6	IN_EVNTCNT_5	IN_EVNTCNT_4	IN_EVNTCNT_3	IN_EVNTCNT_2	IN_EVNTCNT_1	IN_EVNTCNT_0

**GPIO\_H\_IN\_EVNTCNT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_EVNTCNT_28	IN_EVNTCNT_27	IN_EVNTCNT_26	IN_EVNTCNT_25	IN_EVNTCNT_24	RSVD	IN_EVNTCNT_22	IN_EVNTCNT_21	IN_EVNTCNT_20	IN_EVNTCNT_19	IN_EVNTCNT_18	IN_EVNTCNT_17	IN_EVNTCNT_16	RSVD	RSVD	RSVD	IN_EVNTCNT_28	IN_EVNTCNT_27	IN_EVNTCNT_26	IN_EVNTCNT_25	IN_EVNTCNT_24	RSVD	IN_EVNTCNT_22	IN_EVNTCNT_21	IN_EVNTCNT_20	IN_EVNTCNT_19	IN_EVNTCNT_18	IN_EVNTCNT_17	IN_EVNTCNT_16

### 6.16.2.13 GPIO Input Auxiliary 1 Select (GPIO[x]\_IN\_AUX1\_SEL)

Each GPIO has a dedicated internal destination for the conditioned input from the component ball; these inputs are activated when Auxiliary 1 Input is selected. Table 3-8 "GPIO Options" on page 47 shows all the dedicated destinations. These registers select the Auxiliary 1 Input of the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. Auxiliary 1 Input is selected as the input if the feature bit is high. Auxiliary 1 is not selected as the input if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

#### GPIO Low Bank Input Auxiliary 1 Select (GPIO\_L\_IN\_AUX1\_SEL)

GPIO I/O Offset 34h  
 Type R/W  
 Reset Value FFFF0000h

#### GPIO High Bank Input Auxiliary 1 Select (GPIO\_H\_IN\_AUX1\_SEL)

GPIO I/O Offset B4h  
 Type R/W  
 Reset Value EFFF1000h

**GPIO\_L\_IN\_AUX1\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_AUX1_15	IN_AUX1_14	IN_AUX1_13	IN_AUX1_12	IN_AUX1_11	IN_AUX1_10	IN_AUX1_9	IN_AUX1_8	IN_AUX1_7	IN_AUX1_6	IN_AUX1_5	IN_AUX1_4	IN_AUX1_3	IN_AUX1_2	IN_AUX1_1	IN_AUX1_0	IN_AUX1_15	IN_AUX1_14	IN_AUX1_13	IN_AUX1_12	IN_AUX1_11	IN_AUX1_10	IN_AUX1_9	IN_AUX1_8	IN_AUX1_7	IN_AUX1_6	IN_AUX1_5	IN_AUX1_4	IN_AUX1_3	IN_AUX1_2	IN_AUX1_1	IN_AUX1_0

**GPIO\_H\_IN\_AUX1\_SEL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_AUX1_28	IN_AUX1_27	IN_AUX1_26	IN_AUX1_25	IN_AUX1_24	RSVD	IN_AUX1_22	IN_AUX1_21	IN_AUX1_20	IN_AUX1_19	IN_AUX1_18	IN_AUX1_17	IN_AUX1_16	RSVD	RSVD	RSVD	IN_AUX1_28	IN_AUX1_27	IN_AUX1_26	IN_AUX1_25	IN_AUX1_24	RSVD	IN_AUX1_22	IN_AUX1_21	IN_AUX1_20	IN_AUX1_19	IN_AUX1_18	IN_AUX1_17	IN_AUX1_16

### 6.16.2.14 GPIO/IN\_AUX Event Enable (GPIO[x]\_EVNT\_EN)

These registers control the Event Enable for INT (Interrupt) and PME (Power Management Event) mapping of the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The GPIO is enabled for mapping if the feature bit is high. The GPIO is disabled from mapping if the feature bit is low. Actual mapping is performed by the GPIO X, Y, Z, and W mapping registers, detailed on page 482 through page 479; the Event Enable registers simply enable/disable the associated GPIO for mapping. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

#### GPIO Low Bank Events Enable (GPIO\_L\_EVNT\_EN)

GPIO I/O Offset 38h  
 Type R/W  
 Reset Value FFFF0000h

#### GPIO High Bank Events Enable (GPIO\_H\_EVNT\_EN)

GPIO I/O Offset B8h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_EVNT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVNT_15	EVNT_14	EVNT_13	EVNT_12	EVNT_11	EVNT_10	EVNT_9	EVNT_8	EVNT_7	EVNT_6	EVNT_5	EVNT_4	EVNT_3	EVNT_2	EVNT_1	EVNT_0	EVNT_15	EVNT_14	EVNT_13	EVNT_12	EVNT_11	EVNT_10	EVNT_9	EVNT_8	EVNT_7	EVNT_6	EVNT_5	EVNT_4	EVNT_3	EVNT_2	EVNT_1	EVNT_0

**GPIO\_H\_EVNT\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	EVNT_28	EVNT_27	EVNT_26	EVNT_25	EVNT_24	RSVD	EVNT_22	EVNT_21	EVNT_20	EVNT_19	EVNT_18	EVNT_17	EVNT_16	RSVD	RSVD	RSVD	EVNT_28	EVNT_27	EVNT_26	EVNT_25	EVNT_24	RSVD	EVNT_22	EVNT_21	EVNT_20	EVNT_19	EVNT_18	EVNT_17	EVNT_16

**6.16.2.15 GPIO/IN\_AUX Input Positive Edge Enable (GPIO[x]\_IN\_POSEDGE\_EN)**

These registers control the enabling of the positive edge detector function for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The positive edge detector function is enabled if the feature bit is high. The positive edge detector function is disabled if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Positive Edge Enable (GPIO\_L\_IN\_POSEDGE\_EN)**

GPIO I/O Offset 40h  
Type R/W  
Reset Value FFFF0000h

**GPIO High Bank Input Positive Edge Enable (GPIO\_H\_IN\_POSEDGE\_EN)**

GPIO I/O Offset C0h  
Type R/W  
Reset Value FFFF0000h

**GPIO\_L\_IN\_POSEDGE\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_POS_15	IN_POS_14	IN_POS_13	IN_POS_12	IN_POS_11	IN_POS_10	IN_POS_9	IN_POS_8	IN_POS_7	IN_POS_6	IN_POS_5	IN_POS_4	IN_POS_3	IN_POS_2	IN_POS_1	IN_POS_0	IN_POS_15	IN_POS_14	IN_POS_13	IN_POS_12	IN_POS_11	IN_POS_10	IN_POS_9	IN_POS_8	IN_POS_7	IN_POS_6	IN_POS_5	IN_POS_4	IN_POS_3	IN_POS_2	IN_POS_1	IN_POS_0

**GPIO\_H\_IN\_POSEDGE\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_POS_28	IN_POS_27	IN_POS_26	IN_POS_25	IN_POS_24	RSVD	IN_POS_22	IN_POS_21	IN_POS_20	IN_POS_19	IN_POS_18	IN_POS_17	IN_POS_16	RSVD	RSVD	RSVD	IN_POS_28	IN_POS_27	IN_POS_26	IN_POS_25	IN_POS_24	RSVD	IN_POS_22	IN_POS_21	IN_POS_20	IN_POS_19	IN_POS_18	IN_POS_17	IN_POS_16

**6.16.2.16 GPIO/IN\_AUX Input Negative Edge Enable (GPIO[x]\_IN\_NEGEDGE\_EN)**

These registers control the enabling of the negative edge detector function in the inputs for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. The negative edge detector function is enabled if the feature bit is high. The negative edge detector function is disabled if the feature bit is low. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Negative Edge Enable (GPIO\_L\_IN\_NEGEDGE\_EN)**

GPIO I/O Offset 44h  
Type R/W  
Reset Value FFFF0000h

**GPIO High Bank Input Negative Edge Enable (GPIO\_H\_IN\_NEGEDGE\_EN)**

GPIO I/O Offset C4h  
Type R/W  
Reset Value FFFF0000h

**GPIO\_L\_IN\_NEGEDGE\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_NEG_15	IN_NEG_14	IN_NEG_13	IN_NEG_12	IN_NEG_11	IN_NEG_10	IN_NEG_9	IN_NEG_8	IN_NEG_7	IN_NEG_6	IN_NEG_5	IN_NEG_4	IN_NEG_3	IN_NEG_2	IN_NEG_1	IN_NEG_0	IN_NEG_15	IN_NEG_14	IN_NEG_13	IN_NEG_12	IN_NEG_11	IN_NEG_10	IN_NEG_9	IN_NEG_8	IN_NEG_7	IN_NEG_6	IN_NEG_5	IN_NEG_4	IN_NEG_3	IN_NEG_2	IN_NEG_1	IN_NEG_0

**GPIO\_H\_IN\_NEGEDGE\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_NEG_28	IN_NEG_27	IN_NEG_26	IN_NEG_25	IN_NEG_24	RSVD	IN_NEG_22	IN_NEG_21	IN_NEG_20	IN_NEG_19	IN_NEG_18	IN_NEG_17	IN_NEG_16	RSVD	RSVD	RSVD	IN_NEG_28	IN_NEG_27	IN_NEG_26	IN_NEG_25	IN_NEG_24	RSVD	IN_NEG_22	IN_NEG_21	IN_NEG_20	IN_NEG_19	IN_NEG_18	IN_NEG_17	IN_NEG_16



**6.16.2.17 GPIO/IN\_AUX Input Positive Edge Status (GPIO[x]\_IN\_POSEDGE\_STS)**

These registers report the status of the positive edge detection function for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. Writing a 1 clears the detected edge and reading returns the current status. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Positive Edge Status (GPIO\_L\_IN\_POSEDGE\_STS)**

GPIO I/O Offset 48h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Input Positive Edge Status (GPIO\_H\_IN\_POSEDGE\_STS)**

GPIO I/O Offset C8h  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_IN\_POSEDGE\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_POS_STS_15	IN_POS_STS_14	IN_POS_STS_13	IN_POS_STS_12	IN_POS_STS_11	IN_POS_STS_10	IN_POS_STS_9	IN_POS_STS_8	IN_POS_STS_7	IN_POS_STS_6	IN_POS_STS_5	IN_POS_STS_4	IN_POS_STS_3	IN_POS_STS_2	IN_POS_STS_1	IN_POS_STS_0	IN_POS_STS_15	IN_POS_STS_14	IN_POS_STS_13	IN_POS_STS_12	IN_POS_STS_11	IN_POS_STS_10	IN_POS_STS_9	IN_POS_STS_8	IN_POS_STS_7	IN_POS_STS_6	IN_POS_STS_5	IN_POS_STS_4	IN_POS_STS_3	IN_POS_STS_2	IN_POS_STS_1	IN_POS_STS_0

**GPIO\_H\_IN\_POSEDGE\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_POS_STS_28	IN_POS_STS_27	IN_POS_STS_26	IN_POS_STS_25	IN_POS_STS_24	RSVD	IN_POS_STS_22	IN_POS_STS_21	IN_POS_STS_20	IN_POS_STS_19	IN_POS_STS_18	IN_POS_STS_17	IN_POS_STS_16	RSVD	RSVD	RSVD	IN_POS_STS_28	IN_POS_STS_27	IN_POS_STS_26	IN_POS_STS_25	IN_POS_STS_24	RSVD	IN_POS_STS_22	IN_POS_STS_21	IN_POS_STS_20	IN_POS_STS_19	IN_POS_STS_18	IN_POS_STS_17	IN_POS_STS_16

**6.16.2.18 GPIO/IN\_AUX Input Negative Edge Status (GPIO[x]\_IN\_NEGEDGE\_STS)**

These registers report the status of the negative edge detection function for the low (GPIO[15:0]) and high (GPIO[31:16]) banks of GPIOs. Writing a 1 clears the detected edge and reading returns the current status. (These registers use atomic programming, see Section 6.16.1 on page 454 for details.)

**GPIO Low Bank Input Negative Edge Status (GPIO\_L\_IN\_NEGEDGE\_STS)**

GPIO I/O Offset 4Ch  
 Type R/W  
 Reset Value FFFF0000h

**GPIO High Bank Input Negative Edge Status (GPIO\_H\_IN\_NEGEDGE\_STS)**

GPIO I/O Offset CCh  
 Type R/W  
 Reset Value FFFF0000h

**GPIO\_L\_IN\_NEGEDGE\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IN_NEG_STS_15	IN_NEG_STS_14	IN_NEG_STS_13	IN_NEG_STS_12	IN_NEG_STS_11	IN_NEG_STS_10	IN_NEG_STS_9	IN_NEG_STS_8	IN_NEG_STS_7	IN_NEG_STS_6	IN_NEG_STS_5	IN_NEG_STS_4	IN_NEG_STS_3	IN_NEG_STS_2	IN_NEG_STS_1	IN_NEG_STS_0	IN_NEG_STS_15	IN_NEG_STS_14	IN_NEG_STS_13	IN_NEG_STS_12	IN_NEG_STS_11	IN_NEG_STS_10	IN_NEG_STS_9	IN_NEG_STS_8	IN_NEG_STS_7	IN_NEG_STS_6	IN_NEG_STS_5	IN_NEG_STS_4	IN_NEG_STS_3	IN_NEG_STS_2	IN_NEG_STS_1	IN_NEG_STS_0

**GPIO\_H\_IN\_NEGEDGE\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	RSVD	RSVD	IN_NEG_STS_28	IN_NEG_STS_27	IN_NEG_STS_26	IN_NEG_STS_25	IN_NEG_STS_24	RSVD	IN_NEG_STS_22	IN_NEG_STS_21	IN_NEG_STS_20	IN_NEG_STS_19	IN_NEG_STS_18	IN_NEG_STS_17	IN_NEG_STS_16	RSVD	RSVD	RSVD	IN_NEG_STS_28	IN_NEG_STS_27	IN_NEG_STS_26	IN_NEG_STS_25	IN_NEG_STS_24	RSVD	IN_NEG_STS_22	IN_NEG_STS_21	IN_NEG_STS_20	IN_NEG_STS_19	IN_NEG_STS_18	IN_NEG_STS_17	IN_NEG_STS_16

### 6.16.2.19 GPIO Read Back (GPIO[x]\_READ\_BACK)

The Read Back registers provide the current values of the states of each GPIO as sent to the ball. The GPIO[x]\_READ\_BACK registers are not based on the atomic programming model since these are not control registers.

#### GPIO Low Bank Read Back (GPIOL\_READ\_BACK)

GPIO I/O Offset 30h  
 Type RO  
 Reset Value 00000000h

#### GPIOL\_READ\_BACK Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RB_15	RB_14	RB_13	RB_12	RB_11	RB_10	RB_9	RB_8	RB_7	RB_6	RB_5	RB_4	RB_3	RB_2	RB_1	RB_0

#### GPIOL\_READ\_BACK Bit Descriptions

Bit	Name	Description
31:16	RSVD	<b>Reserved.</b> Reads back 0.
15	RB_15	<b>GPIO15 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
14	RB_14	<b>GPIO14 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
13	RB_13	<b>GPIO13 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
12	RB_12	<b>GPIO12 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
11	RB_11	<b>GPIO11 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
10	RB_10	<b>GPIO10 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
9	RB_9	<b>GPIO9 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
8	RB_8	<b>GPIO8 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
7	RB_7	<b>GPIO7 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
6	RB_6	<b>GPIO6 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
5	RB_5	<b>GPIO5 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
4	RB_4	<b>GPIO4 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
3	RB_3	<b>GPIO3 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
2	RB_2	<b>GPIO2 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
1	RB_1	<b>GPIO1 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
0	RB_0	<b>GPIO0 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.

**GPIO High Bank Read Back (GPIOH\_READ\_BACK)**

GPIO I/O Offset B0h  
 Type RO  
 Reset Value 00000000h

**GPIOH\_READ\_BACK Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RSVD	RSVD	RSVD	RB_28	RB_27	RB_26	RB_25	RB_24	RSVD	RB_22	RB_21	RB_20	RB_19	RB_18	RB_17	RB_16

**GPIOH\_READ\_BACK Bit Descriptions**

Bit	Name	Description
31:16	RSVD	<b>Reserved.</b> Reads back 0.
15	RSVD	<b>Reserved.</b> Reads back 0.
14	RSVD	<b>Reserved.</b> Reads back 0.
13	RSVD	<b>Reserved.</b> Reads back 0.
12	RB_28	<b>GPIO28 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
11	RB_27	<b>GPIO27 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
10	RB_26	<b>GPIO26 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
9	RB_25	<b>GPIO25 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
8	RB_24	<b>GPIO24 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
7	RSVD	<b>Reserved.</b> Reads back 0.
6	RB_22	<b>GPIO22 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
5	RB_21	<b>GPIO21 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
4	RB_20	<b>GPIO20 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
3	RB_19	<b>GPIO19 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
2	RB_18	<b>GPIO18 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
1	RB_17	<b>GPIO17 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.
0	RB_16	<b>GPIO16 Read Back Value.</b> Provides status (1/0) of the associated GPIO ball.

**6.16.2.20 GPIO/AUX Lock Enable (GPIO[x]\_LOCK\_EN)**

These registers lock the values of feature bit registers except the GPIO[x]\_READ\_BACK, GPIO[x]\_IN\_POSEDGE\_STS, and GPIO[x]\_IN\_NEGEDGE\_STS registers. When set, the indicated feature bits may not be changed. The GPIO[x]\_LOCK\_EN registers are not based on the atomic programming model (i.e., only one bit for control as opposed to two bits).

**GPIO Low Bank Lock Enable (GPIOL\_LOCK\_EN)**

GPIO I/O Offset 3Ch  
Type R/W  
Reset Value 00000000h

**GPIOL\_LOCK\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LKNE	LKPE	LKIP	LKIA	LKEE	LKFE	LKII	LKIE	LKPD	LKPU	LKA2	LKA1	LKOI	LKOD	LKOE	LKOV

**GPIOL\_LOCK\_ENABLE Bit Descriptions**

Bit	Name	Description
31:15	RSVD	<b>Reserved.</b> Write to 0.
15	LKNE	<b>Lock GPIOL_IN_NEGEDGE_EN.</b> When set, writing to the GPIO Low Bank Input Negative Edge Enable register (GPIO I/O Offset 44h) is prevented.
14	LKPE	<b>Lock GPIOL_IN_POSEDGE_EN.</b> When set, writing to the GPIO Low Bank Input Positive Edge Enable register (GPIO I/O Offset 40h) is prevented.
13	LKIP	<b>Lock GPIOL_EVENTS_EN.</b> When set, writing to the GPIO Low Bank Events Enable (interrupts & PMEs) register (GPIO I/O Offset 38h) is prevented.
12	LKIA	<b>Lock GPIOL_IN_AUX1_SEL.</b> When set, writing to the GPIO Low Bank Input Auxiliary 1 Select register (GPIO I/O Offset 34h) is prevented.
11	LKEE	<b>Lock GPIOL_IN_EVNTCNT_EN.</b> When set, writing to the GPIO Low Bank Input Event Count Enable register (GPIO I/O Offset 2Ch) is prevented.
10	LKFE	<b>Lock GPIOL_IN_FLTR_EN.</b> When set, writing to the GPIO Low Bank Input Filter Enable register (GPIO I/O Offset 28h) is prevented.
9	LKII	<b>Lock GPIOL_IN_INVRT_EN.</b> When set, writing to the GPIO Low Bank Input Invert Enable register (GPIO I/O Offset 24h) is prevented.
8	LKIE	<b>Lock GPIOL_IN_EN.</b> When set, writing to the GPIO Low Bank Input Enable register (GPIO I/O Offset 20h) is prevented.
7	LKPD	<b>Lock GPIOL_PU_EN.</b> When set, writing to the GPIO Low Bank Pull-Down Enable register (GPIO I/O Offset 1Ch) is prevented.
6	LKPU	<b>Lock GPIOL_PU_EN.</b> When set, writing to the GPIO Low Bank Pull-Up Enable register (GPIO I/O Offset 18h) is prevented.
5	LKA2	<b>Lock GPIOL_OUT_AUX2_SEL.</b> When set, writing to the GPIO Low Bank Output Auxiliary 2 Select register (GPIO I/O Offset 14h) is prevented.
4	LKA1	<b>Lock GPIOL_OUT_AUX1_SEL.</b> When set, writing to the GPIO Low Bank Output Auxiliary 1 Select register (GPIO I/O Offset 10h) is prevented.
3	LKOI	<b>Lock GPIOL_OUT_INVRT_EN.</b> When set, writing to the GPIO Low Bank Output Invert Enable register (GPIO I/O Offset 0Ch) is prevented.
2	LKOD	<b>Lock GPIOL_OUT_OD_EN.</b> When set, writing to the GPIO Low Bank Output Open-Drain Enable register (GPIO I/O Offset 08h) is prevented.
1	LKOE	<b>Lock GPIOL_OUT_EN.</b> When set, writing to the GPIO Low Bank Enable register (GPIO I/O Offset 04h) is prevented.
0	LKOV	<b>Lock GPIOL_OUT_VAL.</b> When set, writing to the GPIO Low Bank Output Value register (GPIO I/O Offset 00h) is prevented.

**GPIO High Bank Lock Enable (GPIOH\_LOCK\_EN)**

GPIO I/O Offset BCh  
 Type R/W  
 Reset Value 00000000h

**GPIOH\_LOCK\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																LKNE	LKPE	LKIP	LKIA	LKEE	LKFE	LKII	LKIE	LKPD	LKPU	LKA2	LKA1	LKOI	LKOD	LKOE	LKOV

**GPIOH\_LOCK\_EN Bit Descriptions**

Bit	Name	Description
31:15	RSVD	<b>Reserved.</b> Write to 0.
15	LKNE	<b>Lock GPIOH_IN_NEGEDGE_EN.</b> When set, writing to the GPIO Low Bank Input Negative Edge Enable register (GPIO I/O Offset C4h) is prevented.
14	LKPE	<b>Lock GPIOH_IN_POSEDGE_EN.</b> When set, writing to the GPIO Low Bank Input Positive Edge Enable register (GPIO I/O Offset C0h) is prevented.
13	LKIP	<b>Lock GPIOH_EVENTS_EN.</b> When set, writing to the GPIO Low Bank Events Enable (interrupts & PMEs) register (GPIO I/O Offset B8h) is prevented.
12	LKIA	<b>Lock GPIOH_IN_AUX1_SEL.</b> When set, writing to the GPIO Low Bank Input Auxiliary 1 Select register (GPIO I/O Offset B4h) is prevented.
11	LKEE	<b>Lock GPIOH_IN_EVNTCNT_EN.</b> When set, writing to the GPIO Low Bank Input Event Count Enable register (GPIO I/O Offset ACh) is prevented.
10	LKFE	<b>Lock GPIOH_IN_FLTR_EN.</b> When set, writing to the GPIO Low Bank Input Filter Enable register (GPIO I/O Offset A8h) is prevented.
9	LKII	<b>Lock GPIOH_IN_INVRT_EN.</b> When set, writing to the GPIO Low Bank Input Invert Enable register (GPIO I/O Offset A4h) is prevented.
8	LKIE	<b>Lock GPIOH_IN_EN.</b> When set, writing to the GPIO Low Bank Input Enable register (GPIO I/O Offset A0h) is prevented.
7	LKPD	<b>Lock GPIOH_PD_EN.</b> When set, writing to the GPIO Low Bank Pull-Down Enable register (GPIO I/O Offset 9Ch) is prevented.
6	LKPU	<b>Lock GPIOH_PU_EN.</b> When set, writing to the GPIO Low Bank Pull-Up Enable register (GPIO I/O Offset 98h) is prevented.
5	LKA2	<b>Lock GPIOH_OUT_AUX2_SEL.</b> When set, writing to the GPIO Low Bank Output Auxiliary 2 Select register (GPIO I/O Offset 94h) is prevented.
4	LKA1	<b>Lock GPIOH_OUT_AUX1_SEL.</b> When set, writing to the GPIO Low Bank Output Auxiliary 1 Select register (GPIO I/O Offset 90h) is prevented.
3	LKOI	<b>Lock GPIOH_OUT_INVRT_EN.</b> When set, writing to the GPIO Low Bank Output Invert Enable register (GPIO I/O Offset 8Ch) is prevented.
2	LKOD	<b>Lock GPIOH_OUT_OD_EN.</b> When set, writing to the GPIO Low Bank Output Open-Drain Enable register (GPIO I/O Offset 88h) is prevented.
1	LKOE	<b>Lock GPIOH_OUTPUT_ENABLE.</b> When set, writing to the GPIO Low Bank Enable register (GPIO I/O Offset 84h) is prevented.
0	LKOV	<b>Lock GPIOH_OUTPUT_VALUE.</b> When set, writing to the GPIO Low Bank Output Value register (GPIO I/O Offset 80h) is prevented.

### 6.16.3 GPIO Input Conditioning Function Registers

The Geode CS5535 companion device has eight digital filter/event counter pairs (numbered 0 through 7) that can be shared with 28 GPIOs. There are two 16-bit registers associated with digital filter (FILTER\_AMOUNT and FILTER\_COUNTER) and two 16-bit registers associated with event counter (EVENTCOUNT and EVENT\_COMP). The Input Conditioning Function registers are not based on the atomic programming model.

#### 6.16.3.1 GPIO Filter Amount (GPIO\_FLTR[x]\_AMNT)

GPIO\_FILTER[x]\_AMOUNT are 16-bit registers and programmed with a 16-bit filter count value.

##### GPIO Filter 0 Amount (GPIO\_FLTR0\_AMNT)

GPIO I/O Offset 50h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 4 Amount (GPIO\_FLTR4\_AMNT)

GPIO I/O Offset 70h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 1 Amount (GPIO\_FLTR1\_AMNT)

GPIO I/O Offset 58h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 5 Amount (GPIO\_FLTR5\_AMNT)

GPIO I/O Offset 78h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 2 Amount (GPIO\_FLTR2\_AMNT)

GPIO I/O Offset 60h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 6 Amount (GPIO\_FLTR6\_AMNT)

GPIO I/O Offset D0h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 3 Amount (GPIO\_FLTR3\_AMNT)

GPIO I/O Offset 68h  
Type R/W  
Reset Value 0000h

##### GPIO Filter 7 Amount (GPIO\_FLTR7\_AMNT)

GPIO I/O Offset D8h  
Type R/W  
Reset Value 0000h

#### GPIO\_FLTR[x]\_AMNT Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FILTER_AMOUNT															

#### GPIO\_FLTR[x]\_AMNT Bit Descriptions

Bit	Name	Description
15:0	FILTER_AMOUNT	<b>Filter Amount.</b> The associated GPIO input must remain stable for a FILTER_AMOUNT number of 32 kHz clock edges in order for the output to change. A FILTER_AMOUNT of 0 effectively disables the filtering function because the counter will not roll over from 0 to all 1s. The maximum FILTER_AMOUNT is FFFFh.

### 6.16.3.2 GPIO Filter Count (GPIO\_FILTER[x]\_COUNT)

Writing to these 16-bit registers programs the counter value. Reads provide current counter value.

#### GPIO Filter 0 Count (GPIO\_FLTR0\_CNT)

GPIO I/O Offset 52h  
Type R/W  
Reset Value 0000h

#### GPIO Filter 4 Count (GPIO\_FLTR4\_CNT)

GPIO I/O Offset 72h  
Type R/W  
Reset Value 0000h

#### GPIO Filter 1 Count (GPIO\_FLTR1\_CNT)

GPIO I/O Offset 5Ah  
Type R/W  
Reset Value 0000h

#### GPIO Filter 5 Count (GPIO\_FLTR5\_CNT)

GPIO I/O Offset 7Ah  
Type R/W  
Reset Value 0000h

#### GPIO Filter 2 Count (GPIO\_FLTR2\_CNT)

GPIO I/O Offset 62h  
Type R/W  
Reset Value 0000h

#### GPIO Filter 6 Count (GPIO\_FLTR6\_CNT)

GPIO I/O Offset D2h  
Type R/W  
Reset Value 0000h

#### GPIO Filter 3 Count (GPIO\_FLTR3\_CNT)

GPIO I/O Offset 6Ah  
Type R/W  
Reset Value 0000h

#### GPIO Filter 7 Count (GPIO\_FLTR7\_CNT)

GPIO I/O Offset DAh  
Type R/W  
Reset Value 0000h

#### GPIO\_FLTR[x]\_CNT Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FILTER_COUNT															

#### GPIO\_FLTR[x]\_CNT Bit Descriptions

Bit	Name	Description
15:0	FILTER_COUNT	<b>Filter Count.</b> An initial count is loaded into the FILTER_COUNT via the FILTER_AMOUNT register. Direct access to the counter's state is provided via the FILTER_COUNT register and may be read at any time to determine the current value of the counter. The FILTER_COUNT register may also be written to at any time, thereby jamming the counter state forward or backward from the current count.



**6.16.3.3 GPIO Event Counter (GPIO\_EVNTCNT[x])**

Writing to these 16-bit registers programs the counter value. Reads provide current counter value.

**GPIO Event Counter 0 (GPIO\_EVNTCNT0)**

GPIO I/O Offset 54h  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 4 (GPIO\_EVNTCNT4)**

GPIO I/O Offset 74h  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 1 (GPIO\_EVNTCNT1)**

GPIO I/O Offset 5Ch  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 5 (GPIO\_EVNTCNT5)**

GPIO I/O Offset 7Ch  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 2 (GPIO\_EVNTCNT2)**

GPIO I/O Offset 64h  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 6 (GPIO\_EVNTCNT6)**

GPIO I/O Offset D4h  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 3 (GPIO\_EVNTCNT3)**

GPIO I/O Offset 6Ch  
Type R/W  
Reset Value 0000h

**GPIO Event Counter 7 (GPIO\_EVNTCNT7)**

GPIO I/O Offset DCh  
Type R/W  
Reset Value 0000h

**GPIO\_EVNTCNT[x] Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT_COUNT															

**GPIO\_EVNTCNT\_[x] Bit Descriptions**

Bit	Name	Description
15:0	EVENT_COUNT	<b>Event Counter Status.</b> Direct access to the counter's state is provided via this register and may be read at any time to determine the current value of the counter. This register may also be written to at any time, thereby jamming the counter state forward or backward from the current count. Hardware provisions exist to ensure accurate readings even if a counter edge is in process.

### 6.16.3.4 GPIO Event Counter Compare Value (GPIO\_EVNTCNT[x]\_COMP)

These 16-bit registers are programmed with event count compare value.

#### GPIO Event Counter 0 Compare Value (GPIO\_EVNTCNT0\_COMP)

GPIO I/O Offset 56h  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 4 Compare Value (GPIO\_EVNTCNT4\_COMP)

GPIO I/O Offset 76h  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 1 Compare Value (GPIO\_EVNTCNT1\_COMP)

GPIO I/O Offset 5Eh  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 5 Compare Value (GPIO\_EVNTCNT5\_COMP)

GPIO I/O Offset 7Eh  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 2 Compare Value (GPIO\_EVNTCNT2\_COMP)

GPIO I/O Offset 66h  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 6 Compare Value (GPIO\_EVNTCNT6\_COMP)

GPIO I/O Offset D6h  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 3 Compare Value (GPIO\_EVNTCNT3\_COMP)

GPIO I/O Offset 6Eh  
Type R/W  
Reset Value 0000h

#### GPIO Event Counter 7 Compare Value (GPIO\_EVNTCNT7\_COMP)

GPIO I/O Offset DEh  
Type R/W  
Reset Value 0000h

#### GPIO\_EVNTCNT[x]\_COMP Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVNTCNT_COMP															

#### GPIO\_EVENTCOUNT\_[x] Bit Descriptions

Bit	Name	Description
15:0	EVNTCNT_COMP	<b>Event Counter Compare Value.</b> This register is used to set the event counter's compare value. The compare value, when exceeded by the event counter, causes the counter to produce a constant (level) output.

**6.16.3.5 GPIO Filter/Event Pair Selection (GPIO\_FE[x]\_SEL)**

These registers assign any GPIO to one of the eight filter/event pairs; part of the input conditioning functions.

**GPIO Filter/Event Pair 0 Selection (GPIO\_FE0\_SEL)**

GPIO I/O Offset F0h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 4 Selection (GPIO\_FE4\_SEL)**

GPIO I/O Offset F4h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 1 Selection (GPIO\_FE1\_SEL)**

GPIO I/O Offset F1h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 5 Selection (GPIO\_FE5\_SEL)**

GPIO I/O Offset F5h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 2 Selection (GPIO\_FE2\_SEL)**

GPIO I/O Offset F2h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 6 Selection (GPIO\_FE6\_SEL)**

GPIO I/O Offset F6h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 3 Selection (GPIO\_FE3\_SEL)**

GPIO I/O Offset F3h  
Type R/W  
Reset Value 00h

**GPIO Filter/Event Pair 7 Selection (GPIO\_FE7\_SEL)**

GPIO I/O Offset F7h  
Type R/W  
Reset Value 00h

**GPIO\_FE[x]\_SEL Register Map**

7	6	5	4	3	2	1	0
RSVD				FE_SEL			

**GPIO\_FE[x]\_SEL Bit Descriptions**

Bit	Name	Description
7:5	RSVD	Reserved.

## GPIO\_FE[x]\_SEL Bit Descriptions (Continued)

Bit	Name	Description
4:0	FE_SEL	<p><b>Filter/Event Pair Select.</b> Selects one of 32 GPIO inputs, Filter Enables, Event Enables, and Event Counter Decrements for Filter Event Pair [x].</p> <p>00000: GPIO0 is connected to Filter Event Pair [x].  00001: GPIO1 is connected to Filter Event Pair [x].  00010: GPIO2 is connected to Filter Event Pair [x].  00011: GPIO3 is connected to Filter Event Pair [x].  00100: GPIO4 is connected to Filter Event Pair [x].  00101: GPIO5 is connected to Filter Event Pair [x].  00110: GPIO6 is connected to Filter Event Pair [x].  00111: GPIO7 is connected to Filter Event Pair [x].  01000: GPIO8 is connected to Filter Event Pair [x].  01001: GPIO9 is connected to Filter Event Pair [x].  01010: GPIO10 is connected to Filter Event Pair [x].  01011: GPIO11 is connected to Filter Event Pair [x].  01100: GPIO12 is connected to Filter Event Pair [x].  01101: GPIO13 is connected to Filter Event Pair [x].  01110: GPIO14 is connected to Filter Event Pair [x].  01111: GPIO15 is connected to Filter Event Pair [x].  10000: GPIO16 is connected to Filter Event Pair [x].  10001: GPIO17 is connected to Filter Event Pair [x].  10010: GPIO18 is connected to Filter Event Pair [x].  10101: GPIO19 is connected to Filter Event Pair [x].  10100: GPIO20 is connected to Filter Event Pair [x].  10101: GPIO21 is connected to Filter Event Pair [x].  10110: GPIO22 is connected to Filter Event Pair [x].  10111: Reserved.  11000: GPIO24 is connected to Filter Event Pair [x].  11001: GPIO25 is connected to Filter Event Pair [x].  11010: GPIO26 is connected to Filter Event Pair [x].  11011: GPIO27 is connected to Filter Event Pair [x].  11100: GPIO28 is connected to Filter Event Pair [x].  11101: Reserved.  11110: Reserved.  11111: Reserved.</p>

### 6.16.3.6 GPIO Event Counter Decrement (GPIO[x]\_EVNTCNT\_DEC)

There are two 32-bit Event Counter Decrement registers one for the lower bank (GPIO[15:0]) and one for the higher bank (GPIO[31:16]) of GPIOs. These registers generate one 33 ns wide pulse when written to it, so multiple successive writes may be performed without waiting for the previous write to 'complete'; in addition, reading these registers always provides 0s.

#### GPIO Low Bank Event Counter Decrement (GPIOL\_EVNTCNT\_DEC)

GPIO I/O Offset F8h  
Type R/W  
Reset Value 00000000h

#### GPIOL\_EVNTCNT\_DEC Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																ECD_15	ECD_14	ECD_13	ECD_12	ECD_11	ECD_10	ECD_9	ECD_8	ECD_7	ECD_6	ECD_5	ECD_4	ECD_3	ECD_2	ECD_1	ECD_0

#### GPIOL\_EVNTCNT\_DEC Bit Descriptions

Bit	Name	Description
31:16	RSVD	Reserved. Write/read as 0.
15	ECD15	<b>GPIO15 Event Counter Decrement.</b> Writing this bit high generates a decrement pulse to the event counter that has been associated with this GPIO. There is no need to write the bit low again. This bit will always read as low. Event counters are associated with specific GPIOs via the GPIO_FE[x]_SEL register set.
14	ECD14	<b>GPIO14 Event Counter Decrement.</b> Same as EDC15 (bit 15)
13	ECD13	<b>GPIO13 Event Counter Decrement.</b> Same as EDC15 (bit 15)
12	ECD12	<b>GPIO12 Event Counter Decrement.</b> Same as EDC15 (bit 15).
11	ECD11	<b>GPIO11 Event Counter Decrement.</b> Same as EDC15 (bit 15).
10	ECD10	<b>GPIO10 Event Counter Decrement.</b> Same as EDC15 (bit 15).
9	ECD9	<b>GPIO9 Event Counter Decrement.</b> Same as EDC15 (bit 15).
8	ECD8	<b>GPIO8 Event Counter Decrement.</b> Same as EDC15 (bit 15).
7	ECD7	<b>GPIO7 Event Counter Decrement.</b> Same as EDC15 (bit 15).
6	ECD6	<b>GPIO6 Event Counter Decrement.</b> Same as EDC15 (bit 15).
5	ECD5	<b>GPIO5 Event Counter Decrement.</b> Same as EDC15 (bit 15).
4	ECD4	<b>GPIO4 Event Counter Decrement.</b> Same as EDC15 (bit 15).
3	ECD3	<b>GPIO3 Event Counter Decrement.</b> Same as EDC15 (bit 15).
2	ECD2	<b>GPIO2 Event Counter Decrement.</b> Same as EDC15 (bit 15).
1	ECD1	<b>GPIO1 Event Counter Decrement.</b> Same as EDC15 (bit 15).
0	ECD0	<b>GPIO0 Event Counter Decrement.</b> Same as EDC15 (bit 15).

**GPIO High Bank Event Counter Decrement  
(GPIOH\_EVNTCNT\_DEC)**

GPIO I/O Offset FCh  
 Type R/W  
 Reset Value 00000000h

**GPIOH\_EVNTCNT\_DEC Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																RSVD	RSVD	RSVD	ECD_28	ECD_27	ECD_26	ECD_25	ECD_24	RSVD	ECD_22	ECD_21	ECD_20	ECD_19	ECD_18	ECD_17	ECD_16

**GPIOH\_EVNTCNT\_DEC Bit Descriptions**

Bit	Name	Description
31:13	RSVD	<b>Reserved.</b> Write/read as 0.
12	ECD28	<b>GPIO28 Event Counter Decrement.</b> Writing this bit high generates a decrement pulse to the event counter that has been associated with this GPIO. There is no need to write the bit low again. This bit will always read as low. Event counters are associated with specific GPIOs via the GPIO_FE[x]_SEL register set.
11	ECD27	<b>GPIO27 Event Counter Decrement.</b> Same as EDC28 (bit 12).
10	ECD26	<b>GPIO26 Event Counter Decrement.</b> Same as EDC28 (bit 12).
9	ECD25	<b>GPIO25 Event Counter Decrement.</b> Same as EDC28 (bit 12).
8	ECD24	<b>GPIO24 Event Counter Decrement.</b> Same as EDC28 (bit 12).
7	RSVD	<b>Reserved.</b> Write/read as 0.
6	ECD22	<b>GPIO22 Event Counter Decrement.</b> Same as EDC28 (bit 12).
5	ECD21	<b>GPIO21 Event Counter Decrement.</b> Same as EDC28 (bit 12).
4	ECD20	<b>GPIO20 Event Counter Decrement.</b> Same as EDC28 (bit 12).
3	ECD19	<b>GPIO19 Event Counter Decrement.</b> Same as EDC28 (bit 12).
2	ECD18	<b>GPIO18 Event Counter Decrement.</b> Same as EDC28 (bit 12).
1	ECD17	<b>GPIO17 Event Counter Decrement.</b> Same as EDC28 (bit 12).
0	ECD16	<b>GPIO16 Event Counter Decrement.</b> Same as EDC28 (bit 12).

### 6.16.4 GPIO Interrupt and PME Registers

There are four 32-bit registers in the mapper used for GPIO INT (Interrupt) and PME (Power Management Event) mapping. These registers connect any GPIO to one of eight PIC interrupts or to one of eight PME inputs.

- 1) GPIO\_MAP\_W: Maps 8 final inputs ([31:24] of 32 final inputs).
- 2) GPIO\_MAP\_Z: Maps 8 final inputs ([23:16] of 32 final inputs).
- 3) GPIO\_MAP\_Y: Maps 8 final inputs ([15:8] of 32 final inputs).
- 4) GPIO\_MAP\_X: Maps 8 final inputs ([7:0] of 32 final inputs).

The MAP registers setup the routing of the final inputs to either GPIO\_INT[7:0] or GPIO\_PME[7:0]. The four registers contain 32 4-bit fields, that is a nibble for each final input. Each nibble contains the following control bits:

- PME\_SEL: Located in MSB of the nibble and directs the final input to INT when low. If high, the final input is directed to PME outputs.
- MAP\_SEL: These bits determine which bit in the output field the final input is directed to (i.e., either (GPIO\_INT[7:0]) or GPIO\_PME[7:0]).

#### 6.16.4.1 GPIO Mapper W (GPIO\_MAP\_W)

GPIO I/O Offset ECh  
 Type R/W  
 Reset Value 00000000h

**GPIO\_MAP\_W Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD												PME_SEL_28	MAP_SEL_28			PME_SEL_27	MAP_SEL_27			PME_SEL_26	MAP_SEL_26			PME_SEL_25	MAP_SEL_25			PME_SEL_24	MAP_SEL_24		

**GPIO\_MAP\_W Bit Descriptions**

Bit	Name	Description
31:20	RSVD	<b>Reserved.</b> Write as 0.
19	PME_SEL_28	<b>GPIO28 PME Select.</b> Selects where to map GPIO28. 0: INT (Interrupt). 1: PME (Power Management Event).
18:16	MAP_SEL_28	<b>GPIO28 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO28 should be mapped to. 000: Bit 0      010: Bit 2      100: Bit 4      110: Bit 6 001: Bit 1      011: Bit 3      101: Bit 5      111: Bit 7
15	PME_SEL_27	<b>GPIO27 PME Select.</b> Selects where to map GPIO27. See bit 19 for decode.
14:12	MAP_SEL_27	<b>GPIO27 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO27 should be mapped to. See bits [18:16] for decode.
11	PME_SEL_26	<b>GPIO26 PME Select.</b> Selects where to map GPIO26. See bit 19 for decode.
10:8	MAP_SEL_26	<b>GPIO26 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO26 should be mapped to. See bits [18:16] for decode.
7	PME_SEL_25	<b>GPIO25 PME Select.</b> Selects where to map GPIO25. See bit 19 for decode.
6:4	MAP_SEL_25	<b>GPIO25 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO25 should be mapped to. See bits [18:16] for decode.
3	PME_SEL_24	<b>GPIO24 PME Select.</b> Selects where to map GPIO24. See bit 19 for decode.
2:0	MAP_SEL_24	<b>GPIO24 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO24 should be mapped to. See bits [18:16] for decode.

**6.16.4.2 GPIO Mapper Z (GPIO\_MAP\_Z)**

GPIO I/O Offset E8h  
 Type R/W  
 Reset Value 00000000h

**GPIO\_MAP\_Z Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD				PME_SEL_22	MAP_SEL_22		PME_SEL_21	MAP_SEL_21		PME_SEL_20	MAP_SEL_20		PME_SEL_19	MAP_SEL_19		PME_SEL_18	MAP_SEL_18		PME_SEL_17	MAP_SEL_17		PME_SEL_16	MAP_SEL_16		PME_SEL_16	MAP_SEL_16		PME_SEL_16	MAP_SEL_16		PME_SEL_16	MAP_SEL_16	

**GPIO\_MAP\_Z Bit Descriptions**

Bit	Name	Description
31:28	RSVD	<b>Reserved.</b> Write as 0.
27	PME_SEL_22	<b>GPIO22 PME Select.</b> Selects where to map GPIO22. 0: INT (Interrupt). 1: PME (Power Management Event).
26:24	MAP_SEL_22	<b>GPIO22 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO22 should be mapped to. 000: Bit 0      010: Bit 2      100: Bit 4      110: Bit 6 001: Bit 1      011: Bit 3      101: Bit 5      111: Bit 7
23	PME_SEL_21	<b>GPIO21 PME Select.</b> Selects where to map GPIO21. See bit 27 for decode.
22:20	MAP_SEL_21	<b>GPIO21 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO21 should be mapped to. See bits [26:24] for decode.
19	PME_SEL_20	<b>GPIO20 PME Select.</b> Selects where to map GPIO20. See bit 27 for decode.
18:16	MAP_SEL_20	<b>GPIO20 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO20 should be mapped to. See bits [26:24] for decode.
15	PME_SEL_19	<b>GPIO19 PME Select.</b> Selects where to map GPIO19. See bit 27 for decode.
14:12	MAP_SEL_19	<b>GPIO19 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO19 should be mapped to. See bits [26:24] for decode.
11	PME_SEL_18	<b>GPIO18 PME Select.</b> Selects where to map GPIO18. See bit 27 for decode.
10:8	MAP_SEL_18	<b>GPIO18 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO18 should be mapped to. See bits [26:24] for decode.
7	PME_SEL_17	<b>GPIO17 PME Select.</b> Selects where to map GPIO17. See bit 27 for decode.
6:4	MAP_SEL_17	<b>GPIO17 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO17 should be mapped to. See bits [26:24] for decode.
3	PME_SEL_16	<b>GPIO16 PME Select.</b> Selects where to map GPIO16. See bit 27 for decode.
2:0	MAP_SEL_16	<b>GPIO16 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO16 should be mapped to. See bits [26:24] for decode.



### 6.16.4.3 GPIO Mapper Y (GPIO\_MAP\_Y)

GPIO I/O Offset E4h  
 Type R/W  
 Reset Value 00000000h

#### GPIO\_MAP\_Y Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PME_SEL_15	MAP_SEL_15		PME_SEL_14	MAP_SEL_14		PME_SEL_13	MAP_SEL_13		PME_SEL_12	MAP_SEL_12		PME_SEL_11	MAP_SEL_11		PME_SEL_10	MAP_SEL_10		PME_SEL_9	MAP_SEL_9		PME_SEL_8	MAP_SEL_8									

#### GPIO\_MAP\_Y Bit Descriptions

Bit	Name	Description
31	PME_SEL_15	<b>GPIO15 PME Select.</b> Selects where to map GPIO15. 0: INT (Interrupt). 1: PME (Power Management Event).
30:28	MAP_SEL_15	<b>GPIO15 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO15 should be mapped to. 000: Bit 0      010: Bit 2      100: Bit 4      110: Bit 6 001: Bit 1      011: Bit 3      101: Bit 5      111: Bit 7
27	PME_SEL_14	<b>GPIO14 PME Select.</b> Selects where to map GPIO14. See bit 31 for decode.
26:24	MAP_SEL_14	<b>GPIO14 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO14 should be mapped to. See bits [30:28] for decode.
23	PME_SEL_13	<b>GPIO13 PME Select.</b> Selects where to map GPIO13. See bit 31 for decode.
22:20	MAP_SEL_13	<b>GPIO13 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO13 should be mapped to. See bits [30:28] for decode.
19	PME_SEL_12	<b>GPIO12 PME Select.</b> Selects where to map GPIO12. See bit 31 for decode.
18:16	MAP_SEL_12	<b>GPIO12 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO12 should be mapped to. See bits [30:28] for decode.
15	PME_SEL_11	<b>GPIO11 PME Select.</b> Selects where to map GPIO11. See bit 31 for decode.
14:12	MAP_SEL_11	<b>GPIO11 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO11 should be mapped to. See bits [30:28] for decode.
11	PME_SEL_10	<b>GPIO10 PME Select.</b> Selects where to map GPIO10. See bit 31 for decode.
10:8	MAP_SEL_10	<b>GPIO10 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO10 should be mapped to. See bits [30:28] for decode.
7	PME_SEL_9	<b>GPIO9 PME Select.</b> Selects where to map GPIO9. See bit 31 for decode.
6:4	MAP_SEL_9	<b>GPIO9 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO9 should be mapped to. See bits [30:28] for decode.
3	PME_SEL_8	<b>GPIO8 PME Select.</b> Selects where to map GPIO8. See bit 31 for decode.
2:0	MAP_SEL_8	<b>GPIO8 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO8 should be mapped to. See bits [30:28] for decode.

**6.16.4.4 GPIO Mapper X (GPIO\_MAP\_X)**

GPIO I/O Offset E0h  
 Type R/W  
 Reset Value 00000000h

**GPIO\_MAP\_X Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PME_SEL_7	MAP_SEL_7		PME_SEL_6	MAP_SEL_6		PME_SEL_5	MAP_SEL_5		PME_SEL_4	MAP_SEL_4		PME_SEL_3	MAP_SEL_3		PME_SEL_2	MAP_SEL_2		PME_SEL_1	MAP_SEL_1		PME_SEL_0	MAP_SEL_0											

**GPIO\_MAP\_X Bit Descriptions**

Bit	Name	Description
31	PME_SEL_7	<b>GPIO7 PME Select.</b> Selects where to map GPIO7. 0: INT (Interrupt). 1: PME (Power Management Event).
30:28	MAP_SEL_7	<b>GPIO7 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO7 should be mapped to. 000: Bit 0      010: Bit 2      100: Bit 4      110: Bit 6 001: Bit 1      011: Bit 3      101: Bit 5      111: Bit 7
27	PME_SEL_6	<b>GPIO6 PME Select.</b> Selects where to map GPIO6. See bit 31 for decode.
26:24	MAP_SEL_6	<b>GPIO6 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO6 should be mapped to. See bits [30:28] for decode.
23	PME_SEL_5	<b>GPIO5 PME Select.</b> Selects where to map GPIO5. See bit 31 for decode.
22:20	MAP_SEL_5	<b>GPIO5 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO5 should be mapped to. See bits [30:28] for decode.
19	PME_SEL_4	<b>GPIO4 PME Select.</b> Selects where to map GPIO4. See bit 31 for decode.
18:16	MAP_SEL_4	<b>GPIO4 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO4 should be mapped to. See bits [30:28] for decode.
15	PME_SEL_3	<b>GPIO3 PME Select.</b> Selects where to map GPIO3. See bit 31 for decode.
14:12	MAP_SEL_3	<b>GPIO2 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO3 should be mapped to. See bits [30:28] for decode.
11	PME_SEL_2	<b>GPIO2 PME Select.</b> Selects where to map GPIO2. See bit 31 for decode.
10:8	MAP_SEL_2	<b>GPIO2 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO2 should be mapped to. See bits [30:28] for decode.
7	PME_SEL_1	<b>GPIO1 PME Select.</b> Selects where to map GPIO1. See bit 31 for decode.
6:4	MAP_SEL_1	<b>GPIO1 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO1 should be mapped to. See bits [30:28] for decode.
3	PME_SEL_0	<b>GPIO0 PME Select.</b> Selects where to map GPIO0. See bit 31 for decode.
2:0	MAP_SEL_0	<b>GPIO0 Map Select.</b> Selects which bit of in the output field (i.e., INT or PME) GPIO0 should be mapped to. See bits [30:28] for decode.

## 6.17 Multi-Function General Purpose Timer Register Descriptions

The registers for the Multi-Function General Purpose Timer (MFGPT) are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- MFGPT Specific MSRs
- MFGPT Native Registers.

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the MFGPT Specific MSRs (summarized in Table 6-64) are called out as 32 bits. The MFGPT module treats writes to the upper 32 bits (i.e., bits [63:32]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The Native registers associated with the MFGPT (summarized in Table 6-65) are accessed via a Base Address Register, MSR\_LBAR\_MFGPT (MSR 514000Dh), as I/O Offsets. (See Section 6.6.2.6 on page 329 for bit descriptions of the Base Address Register.)

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-64. MFGPT Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400028h	R/W	MFGPT IRQ Mask (MFGPT_IRQ)	00000000h	Page 484
51400029h	R/W	MFGPT NMI and Reset Mask (MFGPT_NR)	00000000h	Page 487
5140002Ah	R/W	MFGPT Reserved (MFGPT_RSVD)	00000000h	Page 488
5140002Bh	WO	MFGPT Clear Setup Test (MFGPT_SETUP)	00000000h	Page 488

**Table 6-65. MFGPT Native Registers Summary**

MFGPT I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
00h	R/W	16	MFGPT0 Comparator 1 (MFGPT0_CMP1)	0000h	Page 489
02h	R/W	16	MFGPT0 Comparator 2 (MFGPT0_CMP2)	0000h	Page 490
04h	R/W	16	MFGPT0 Up Counter (MFGPT0_CNT)	0000h	Page 491
06h	R/W	16	MFGPT0 Setup (MFGPT0_SETUP)	0000h	Page 492
08h	R/W	16	MFGPT1 Comparator 1 (MFGPT1_CMP1)	0000h	Page 489
0Ah	R/W	16	MFGPT1 Comparator 2 (MFGPT1_CMP2)	0000h	Page 490
0Ch	R/W	16	MFGPT1 Up Counter (MFGPT1_CNT)	0000h	Page 491
0Eh	R/W	16	MFGPT1 Setup (MFGPT1_SETUP)	0000h	Page 492
10h	R/W	16	MFGPT2 Comparator 1 (MFGPT2_CMP1)	0000h	Page 489
12h	R/W	16	MFGPT2 Comparator 2 (MFGPT2_CMP2)	0000h	Page 490
14h	R/W	16	MFGPT2 Up Counter (MFGPT2_CNT)	0000h	Page 491
16h	R/W	16	MFGPT2 Setup (MFGPT2_SETUP)	0000h	Page 492
18h	R/W	16	MFGPT3 Comparator 1 (MFGPT3_CMP1)	0000h	Page 489
1Ah	R/W	16	MFGPT3 Comparator 2 (MFGPT3_CMP2)	0000h	Page 490
1Ch	R/W	16	MFGPT3 Up Counter (MFGPT3_CNT)	0000h	Page 491
1Eh	R/W	16	MFGPT3 Setup (MFGPT3_SETUP)	0000h	Page 492
20h	R/W	16	MFGPT4 Comparator 1 (MFGPT4_CMP1)	0000h	Page 489
22h	R/W	16	MFGPT4 Comparator 2 (MFGPT4_CMP2)	0000h	Page 490
24h	R/W	16	MFGPT4 Up Counter (MFGPT4_CNT)	0000h	Page 491

Table 6-65. MFGPT Native Registers Summary (Continued)

MFGPT I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
26h	R/W	16	MFGPT4 Setup (MFGPT4_SETUP)	0000h	Page 492
28h	R/W	16	MFGPT5 Comparator 1 (MFGPT5_CMP1)	0000h	Page 489
2Ah	R/W	16	MFGPT5 Comparator 2 (MFGPT5_CMP2)	0000h	Page 490
2Ch	R/W	16	MFGPT5 Up Counter (MFGPT5_CNT)	0000h	Page 491
2Eh	R/W	16	MFGPT5 Setup (MFGPT5_SETUP)	0000h	Page 492
30h	R/W	16	MFGPT6 Comparator 1 (MFGPT6_CMP1)	0000h	Page 489
32h	R/W	16	MFGPT6 Comparator 2 (MFGPT6_CMP2)	0000h	Page 490
34h	R/W	16	MFGPT6 Up Counter (MFGPT6_CNT)	0000h	Page 491
36h	R/W	16	MFGPT6 Setup (MFGPT6_SETUP)	0000h	Page 492
38h	R/W	16	MFGPT7 Comparator 1 (MFGPT7_CMP1)	0000h	Page 489
3Ah	R/W	16	MFGPT7 Comparator 2 (MFGPT7_CMP2)	0000h	Page 490
3Ch	R/W	16	MFGPT7 Up Counter (MFGPT7_CNT)	0000h	Page 491
3Eh	R/W	16	MFGPT7 Setup (MFGPT7_SETUP)	0000h	Page 492

### 6.17.1 MFGPT Specific MSRs

This register connects the MFGPT Comparator 1 and 2 outputs to the Interrupt Mapper.

#### 6.17.1.1 MFGPT IRQ Mask (MFGPT\_IRQ)

MSR Address 51400028h  
 Type R/W  
 Reset Value 00000000h

MFGPT\_IRQ Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
RSVD																MFGPT7_C2_IRQM	MFGPT6_C2_IRQM	MFGPT5_C2_IRQM	MFGPT4_C2_IRQM	MFGPT3_C2_IRQM	MFGPT2_C2_IRQM	MFGPT1_C2_IRQM	MFGPT0_C2_IRQM	MFGPT7_C1_IRQM	MFGPT6_C1_IRQM	MFGPT5_C1_IRQM	MFGPT4_C1_IRQM	MFGPT3_C1_IRQM	MFGPT2_C1_IRQM	MFGPT1_C1_IRQM	MFGPT0_C1_IRQM																

MFGPT\_IRQ Bit Descriptions

Bit	Name	Description
31:16	RSVD	<b>Reserved.</b> Writes are don't cares.
15	MFGPT7_C2_IRQM	<b>Enable MFGPT7 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 7. The other bit in the ORed pair is bit 11, MFGPT3_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.

## MFGPT\_IRQ Bit Descriptions

Bit	Name	Description
14	MFGPT6_C2_IRQM	<b>Enable MFGPT6 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 6. The other bit in the ORed pair is bit 10, MFGPT2_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
13	MFGPT5_C2_IRQM	<b>Enable MFGPT5 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 5. The other bit in the ORed pair is bit 9, MFGPT1_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
12	MFGPT4_C2_IRQM	<b>Enable MFGPT4 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 4. The other bit in the ORed pair is bit 8, MFGPT0_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
11	MFGPT3_C2_IRQM	<b>Enable MFGPT3 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 7. The other bit in the ORed pair is bit 15, MFGPT7_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
10	MFGPT2_C2_IRQM	<b>Enable MFGPT2 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 6. The other bit in the ORed pair is bit 14, MFGPT6_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
9	MFGPT1_C2_IRQM	<b>Enable MFGPT1 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 5. The other bit in the ORed pair is bit 13, MFGPT5_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
8	MFGPT0_C2_IRQM	<b>Enable MFGPT0 Comparator 2 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 4. The other bit in the ORed pair is bit 12, MFGPT4_C2_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
7	MFGPT7_C1_IRQM	<b>Enable MFGPT7 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 3. The other bit in the ORed pair is bit 3, MFGPT3_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.

## MFGPT\_IRQ Bit Descriptions

Bit	Name	Description
6	MFGPT6_C1_IRQM	<b>Enable MFGPT6 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 2. The other bit in the ORed pair is bit 2, MFGPT2_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
5	MFGPT5_C1_IRQM	<b>Enable MFGPT5 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 1. The other bit in the ORed pair is bit 1, MFGPT1_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
4	MFGPT4_C1_IRQM	<b>Enable MFGPT4 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 0. The other bit in the ORed pair is bit 0, MFGPT0_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
3	MFGPT3_C1_IRQM	<b>Enable MFGPT3 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 3. The other bit in the ORed pair is bit 7, MFGPT7_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
2	MFGPT2_C1_IRQM	<b>Enable MFGPT2 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 2. The other bit in the ORed pair is bit 6, MFGPT6_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
1	MFGPT1_C1_IRQM	<b>Enable MFGPT1 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 1. The other bit in the ORed pair is bit 5, MFGPT5_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.
0	MFGPT0_C1_IRQM	<b>Enable MFGPT0 Comparator 1 Output to the Interrupt Mapper.</b> When set high, this input becomes one of two, ORed together, to form "Unrestricted Sources Z", bit 0. The other bit in the ORed pair is bit 4, MFGPT4_C1_IRQM. The Unrestricted Sources Z are detailed in Table 5-14 "IRQ Map - Unrestricted Sources Z" on page 113. When cleared low, this MFGPT output does not contribute to the Unrestricted Sources Z interrupt.

### 6.17.1.2 MFGPT NMI and Reset Mask (MFGPT\_NR)

This register enables the MFGPT Comparator 1 and 2 outputs to generate resets or NMIs.

MSR Address 51400029h

Type R/W

Reset Value 00000000h

**MFGPT\_NR Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD		MFGPT5_C2_RSTEN	MFGPT4_C2_RSTEN	MFGPT3_C2_RSTEN	MFGPT2_C2_RSTEN	MFGPT1_C2_RSTEN	MFGPT0_C2_RSTEN	RSVD								NMI_LEG	MFGPT7_C2_NMIM	MFGPT6_C2_NMIM	MFGPT5_C2_NMIM	MFGPT4_C2_NMIM	MFGPT3_C2_NMIM	MFGPT2_C2_NMIM	MFGPT1_C2_NMIM	MFGPT0_C2_NMIM	MFGPT7_C1_NMIM	MFGPT6_C1_NMIM	MFGPT5_C1_NMIM	MFGPT4_C1_NMIM	MFGPT3_C1_NMIM	MFGPT2_C1_NMIM	MFGPT1_C1_NMIM	MFGPT0_C1_NMIM

**MFGPT\_NR Bit Descriptions**

Bit	Name	Description
31:30	RSVD	<b>Reserved.</b> Writes are don't care; reads return 0.
29	MFGPT5_C2_RSTEN	<b>MFGPT5 Comparator 2 Reset Enable.</b> Allow MFGPT5 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
28	MFGPT4_C2_RSTEN	<b>MFGPT4 Comparator 2 Reset Enable.</b> Allow MFGPT4 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
27	MFGPT3_C2_RSTEN	<b>MFGPT3 Comparator 2 Reset Enable.</b> Allow MFGPT3 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
26	MFGPT2_C2_RSTEN	<b>MFGPT2 Comparator 2 Reset Enable.</b> Allow MFGPT2 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
25	MFGPT1_C2_RSTEN	<b>MFGPT1 Comparator 2 Reset Enable.</b> Allow MFGPT1 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
24	MFGPT0_C2_RSTEN	<b>MFGPT0 Comparator 2 Reset Enable.</b> Allow MFGPT0 Comparator 2 output to cause a hard reset. 0: Disable; 1: Enable.
23:17	RSVD	<b>Reserved.</b> Writes are don't care; reads return 0.
16	NMI_LEG	<b>Legacy NMI.</b> Allow legacy NMI mask bit (bit 7 of RTC register at I/O Address 070h) to gate NMI. 0: Disable; 1: Enable.
15	MFGPT7_C2_NMIM	<b>MFGPT7 Comparator 2 NMI Enable.</b> Allow MFGPT7 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
14	MFGPT6_C2_NMIM	<b>MFGPT6 Comparator 2 NMI Enable.</b> Allow MFGPT6 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
13	MFGPT5_C2_NMIM	<b>MFGPT5 Comparator 2 NMI Enable.</b> Allow MFGPT5 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
12	MFGPT4_C2_NMIM	<b>MFGPT4 Comparator 2 NMI Enable.</b> Allow MFGPT4 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
11	MFGPT3_C2_NMIM	<b>MFGPT3 Comparator 2 NMI Enable.</b> Allow MFGPT3 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
10	MFGPT2_C2_NMIM	<b>MFGPT2 Comparator 2 NMI Enable.</b> Allow MFGPT2 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
9	MFGPT1_C2_NMIM	<b>MFGPT1 Comparator 2 NMI Enable.</b> Allow MFGPT1 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.

## MFGPT\_NR Bit Descriptions

Bit	Name	Description
8	MFGPT0_C2_NMIM	<b>MFGPT0 Comparator 2 NMI Enable.</b> Allow MFGPT0 Comparator 2 output to cause an NMI. 0: Disable; 1: Enable.
7	MFGPT7_C1_NMIM	<b>MFGPT7 Comparator 1 NMI Enable.</b> Allow MFGPT7 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
6	MFGPT6_C1_NMIM	<b>MFGPT6 Comparator 1 NMI Enable.</b> Allow MFGPT6 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
5	MFGPT5_C1_NMIM	<b>MFGPT5 Comparator 1 NMI Enable.</b> Allow MFGPT5 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
4	MFGPT4_C1_NMIM	<b>MFGPT4 Comparator 1 NMI Enable.</b> Allow MFGPT4 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
3	MFGPT3_C1_NMIM	<b>MFGPT3 Comparator 1 NMI Enable.</b> Allow MFGPT3 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
2	MFGPT2_C1_NMIM	<b>MFGPT2 Comparator 1 NMI Enable.</b> Allow MFGPT2 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
1	MFGPT1_C1_NMIM	<b>MFGPT1 Comparator 1 NMI Enable.</b> Allow MFGPT1 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.
0	MFGPT0_C1_NMIM	<b>MFGPT0 Comparator 1 NMI Enable.</b> Allow MFGPT0 Comparator 1 output to cause an NMI. 0: Disable; 1: Enable.

## 6.17.1.3 MFGPT Reserved (MFGPT\_RSVD)

MSR Address 5140002Ah  
 Type R/W  
 Reset Value 00000000h

This register is reserved. Reads return 0. Writes have no effect.

## 6.17.1.4 MFGPT Clear Setup Test (MFGPT\_SETUP)

MSR Address 5140002Bh  
 Type WO  
 Reset Value 00000000h

## MFGPT\_SETUP Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

## MFGPT\_SETUP Bit Descriptions

Bit	Name	Description
31:0	RSVD	<b>Reserved.</b> These bits are reserved for internal testing only. These bits should not be written to.



## 6.17.2 MFGPT Native Registers

### 6.17.2.1 MFGPT[x] Comparator 1 (MFGPT[x]\_CMP1)

MFGPT0 to MFGPT5 CMP1 registers are in the Working power domain while MFGPT6 and MFGPT7 CMP1 registers are in the Standby power domain.

#### MFGPT0 Comparator 1 (MFGPT0\_CMP1)

MFGPT I/O Offset 00h  
Type R/W  
Reset Value 0000h

#### MFGPT4 Comparator 1 (MFGPT4\_CMP1)

MFGPT I/O Offset 20h  
Type R/W  
Reset Value 0000h

#### MFGPT1 Comparator 1 (MFGPT1\_CMP1)

MFGPT I/O Offset 08h  
Type R/W  
Reset Value 0000h

#### MFGPT5 Comparator 1 (MFGPT5\_CMP1)

MFGPT I/O Offset 28h  
Type R/W  
Reset Value 0000h

#### MFGPT2 Comparator 1 (MFGPT2\_CMP1)

MFGPT I/O Offset 10h  
Type R/W  
Reset Value 0000h

#### MFGPT6 Comparator 1 (MFGPT6\_CMP1)

MFGPT I/O Offset 30h  
Type R/W  
Reset Value 0000h

#### MFGPT3 Comparator 1 (MFGPT3\_CMP1)

MFGPT I/O Offset 18h  
Type R/W  
Reset Value 0000h

#### MFGPT7 Comparator 1 (MFGPT7\_CMP1)

MFGPT I/O Offset 38h  
Type R/W  
Reset Value 0000h

### MFGPT[x]\_CMP1 Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFGPT_CMP1_VAL															

### MFGPT[x]\_CMP1 Bit Descriptions

Bit	Name	Description
15:0	MFGPT_CMP1_VAL	<b>Comparator 1 Comparison Value.</b> This 16-bit value is the compare value for Comparator 1 of MFGPT[x].

**6.17.2.2 MFGPT[x] Comparator 2 (MFGPT[x]\_CMP2)**

MFGPT0 to MFGPT5 CMP2 registers are in the Working power domain while MFGPT6 and MFGPT7 CMP2 registers are in the Standby power domain.

**MFGPT0 Comparator 2 (MFGPT0\_CMP2)**

MFGPT I/O Offset 02h  
Type R/W  
Reset Value 0000h

**MFGPT4 Comparator 2 (MFGPT4\_CMP2)**

MFGPT I/O Offset 22h  
Type R/W  
Reset Value 0000h

**MFGPT1 Comparator 2 (MFGPT1\_CMP2)**

MFGPT I/O Offset 0Ah  
Type R/W  
Reset Value 0000h

**MFGPT5 Comparator 2 (MFGPT5\_CMP2)**

MFGPT I/O Offset 2Ah  
Type R/W  
Reset Value 0000h

**MFGPT2 Comparator 2 (MFGPT2\_CMP2)**

MFGPT I/O Offset 12h  
Type R/W  
Reset Value 0000h

**MFGPT6 Comparator 2 (MFGPT6\_CMP2)**

MFGPT I/O Offset 32h  
Type R/W  
Reset Value 0000h

**MFGPT3 Comparator 2 (MFGPT3\_CMP2)**

MFGPT I/O Offset 1Ah  
Type R/W  
Reset Value 0000h

**MFGPT7 Comparator 2 (MFGPT7\_CMP2)**

MFGPT I/O Offset 3Ah  
Type R/W  
Reset Value 0000h

**MFGPT[x]\_CMP2 Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFGPT_CMP2_VAL															

**MFGPT[x]\_CMP2 Bit Descriptions**

Bit	Name	Description
15:0	MFGPT_CMP2_VAL	<b>Comparator 2 Comparison Value.</b> This 16-bit value is the compare value for Comparator 2 of MFGPT[x].

**6.17.2.3 MFGPT[x] Up Counter (MFGPT[x]\_CNT)**

MFGPT0 to MFGPT5 Up Counter registers are in the Working power domain while MFGPT6 and MFGPT7 Up Counter registers are in the Standby power domain.

**MFGPT0 Up Counter (MFGPT0\_CNT)**

MFGPT I/O Offset 04h  
Type R/W  
Reset Value 0000h

**MFGPT4 Up Counter (MFGPT4\_CNT)**

MFGPT I/O Offset 24h  
Type R/W  
Reset Value 0000h

**MFGPT1 Up Counter (MFGPT1\_CNT)**

MFGPT I/O Offset 0Ch  
Type R/W  
Reset Value 0000h

**MFGPT5 Up Counter (MFGPT5\_CNT)**

MFGPT I/O Offset 2Ch  
Type R/W  
Reset Value 0000h

**MFGPT2 Up Counter (MFGPT2\_CNT)**

MFGPT I/O Offset 14h  
Type R/W  
Reset Value 0000h

**MFGPT6 Up Counter (MFGPT6\_CNT)**

MFGPT I/O Offset 34h  
Type R/W  
Reset Value 0000h

**MFGPT3 Up Counter (MFGPT3\_CNT)**

MFGPT I/O Offset 1Ch  
Type R/W  
Reset Value 0000h

**MFGPT7 Up Counter (MFGPT7\_CNT)**

MFGPT I/O Offset 3Ch  
Type R/W  
Reset Value 0000h

**MFGPT[x]\_CNT Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFGPT_CNT															

**MFGPT[x]\_CNT Bit Descriptions**

Bit	Name	Description
15:0	MFGPT_CNT	<b>Up Counter Value.</b> This register contains the current value of the counter of MFGPT[x]. Hardware guarantees that reading or writing may be performed at any time without experiencing aliasing or 'intermediate-value' problems.

### 6.17.2.4 MFGPT[x] Setup (MFGPT[x]\_SETUP)

MFGPT0 to MFGPT5 Setup registers are in the Working power domain while MFGPT6 and MFGPT7 Setup registers are in the Standby power domain. Bits [11:0] are write-once; bit 12 is read-only.

#### MFGPT0 Setup (MFGPT0\_SETUP)

MFGPT I/O Offset 06h  
Type R/W  
Reset Value 0000h

#### MFGPT4 Setup (MFGPT4\_SETUP)

MFGPT I/O Offset 26h  
Type R/W  
Reset Value 0000h

#### MFGPT1 Setup (MFGPT1\_SETUP)

MFGPT I/O Offset 0Eh  
Type R/W  
Reset Value 0000h

#### MFGPT5 Setup (MFGPT5\_SETUP)

MFGPT I/O Offset 2Eh  
Type R/W  
Reset Value 0000h

#### MFGPT2 Setup (MFGPT2\_SETUP)

MFGPT I/O Offset 16h  
Type R/W  
Reset Value 0000h

#### MFGPT6 Setup (MFGPT6\_SETUP)

MFGPT I/O Offset 36h  
Type R/W  
Reset Value 0000h

#### MFGPT3 Setup (MFGPT3\_SETUP)

MFGPT I/O Offset 1Eh  
Type R/W  
Reset Value 0000h

#### MFGPT7 Setup (MFGPT7\_SETUP)

MFGPT I/O Offset 3Eh  
Type R/W  
Reset Value 0000h

**MFGPT[x]\_SETUP Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFGPT_CNT_EN	MFGPT_CMP2	MFGPT_CMP1	MFGPT_SETUP	MFGPT_STOP_EN	MFGPT_EXT_EN	MFGPT_CMP2 MODE		MFGPT_CMP1 MODE		MFGPT_REV_EN	MFGPT_CLKSEL	MFGPT_SCALE			

**MFGPT[x]\_SETUP Bit Descriptions**

Bit	Name	Description
15	MFGPT_CNT_EN	<b>Counter Enable.</b> Enable MFGPT for counting. 0: Disable; 1: Enable.
14	MFGPT_CMP2	<b>Compare 2 Output Status.</b> If Conditioning Mode is set to Event, writing this bit to a 1 clears the event until the next time Compare 2 goes from low-to-high; reading returns the event status. For other modes, this bit follows current compare output values and writes to this bit have no effect. When Compare 2 value is met, the counter is reset and counting continues.
13	MFGPT_CMP1	<b>Compare 1 Output Status.</b> If Conditioning Mode is set to Event, writing this bit to a 1 clears the event until the next time Compare 1 goes from low-to-high; reading returns the event status. For other modes, this bit follows current compare output values and writes to this bit have no effect. When Compare 2 value is met, counting stops. Counter is reset and restarted only by external enable or disabling/enabling.
12	MFGPT_SETUP (RO)	<b>Setup (Read Only).</b> Any value written to this bit is a 'don't care'. From reset, this bit is low. If low, it indicates the MFGPT has not been setup and is currently disabled. On the first write to this register, bits [11:0] are established per the write and this bit is set to a 1. After this bit is set on the first write, bits [12:0] cannot be changed and subsequent writes are 'don't care'.
11	MFGPT_STOP_EN	<b>Stop Enable (Write Once).</b> Enable counter to Stop on Sleep state for MFGPT0 to MFGPT5, or Standby state for MFGPT6 and MFGPT7. 0: Disable; 1: Enable.



## 6.18 Power Management Controller Register Descriptions

The registers for the Power Management Controller (PMC) are divided into four sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- PMC Specific MSRs
- ACPI Registers
- PM Support Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the PMC Specific MSRs (summarized in Table 6-66) are called out as 32 bits. The PMC module treats writes to the upper 32 bits (i.e., bits [63:32]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The configuration registers associated with the PMC are divided into two categories: ACPI registers (summarized in Table 6-67) and PM Support registers (summarized in Table 6-68 on page 495):

- The ACPI registers are accessed via Base Address Register, MSR\_LBAR ACPI (MSR 5140000Eh), as I/O Offsets. (See Section 6.6.2.7 on page 330 for bit descriptions of the Base Address Register.)
- The PM Support registers are accessed via a Base Address Register, MSR\_LBAR\_PMS (MSR 5140000Fh), as I/O Offsets. (See Section 6.6.2.8 on page 331 for bit descriptions of the Base Address Register.)

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-66. PMC Specific MSRs Summary**

MSR Address	Type	Register Name	Power Domain	Reset Value	Reference
51400050h	R/W	PMC Logic Timer (PMC_LTMR)	Working	00000000h	Page 496
51400051h	R/W	PMC Reserved (PMC_RSVD)	No f/flops	00000000h	Page 496

**Table 6-67. ACPI Registers Summary**

ACPI I/O Offset	Type	Width (Bits)	Register Name	Power Domain	Reset Value	Reference
00h (WORD access only)	R/W	16	PM Status 1 (PM1_STS) (Note 1 and Note 2)	Standby	0000h	Page 496
04h	R/W	16	PM Enable 1 (PM1_EN) (Note 1 and Note 2)	Standby	0100h	Page 498
08h	R/W	16	PM Control 1 (PM1_CNT) (Note 1 and Note 3)	Working	0000h	Page 499
0Ch	R/W	16	PM Control 2 (PM2_CNT) (Note 4)	Working	0000h	Page 500
10h	RO	32	PM Timer (PM_TMR) (Note 1)	Working	0000h	Page 500
14h	R/W	32	PM Reserved (PM_RSVD)	No f/flops	0000h	---
18h	R/W	32	General Purpose Events Status 0 (PM_GPE0_STS) (Note 5)	Standby	00000000h	Page 501
1Ch	R/W	32	General Purpose Events Enable 0 (PM_GPE0_EN) (Note 5)	Standby	00000000h	Page 503

Note 1. Required ACPI register.

Note 2. Both PM1\_STS and PM1\_EN access Offset 00h when using 32-bit access.  
Only PM1\_STS with a 16-bit access to Offset 00h.  
Only PM1\_EN with a 16-bit access to Offset 02h.  
Offset 04h is reserved. Reads return 0.

Note 3. SSMI may be implemented on this register by decode hardware outside of PM module.

Note 4. Optional ACPI register. SSMI may be implemented on this register by decode hardware outside of PM module.

Note 5. Required ACPI register that can also be implemented via a control method.

Table 6-68. PM Support Registers Summary

PMS I/O Offset	Type	Width (Bits)	Register Name	Power Domain	Reset Value	Reference
00h	R/W	16	PM Sleep Start Delay (PM_SSD)	Working	0000h	Page 504
04h	R/W	32	PM Sleep Control X Assert Delay and Enable (PM_SCXA)	Working	00000000h	Page 505
08h	R/W	32	PM Sleep Control Y Assert Delay and Enable (PM_SCYA)	Working	00000000h	Page 505
0Ch	R/W	32	PM Sleep Output Disable Assert Delay and Enable (PM_OUT_SLPCTL)	Working	00000000h	Page 506
10h	R/W	32	PM Sleep Clock Delay and Enable (PM_SCLK)	Working	00000000h	Page 507
14h	R/W	32	PM Sleep End Delay (PM_SED)	Working	00000000h	Page 507
18h	R/W	32	PM Sleep Control X De-assert Delay (PM_SCXD)	Working	00000000h	Page 508
1Ch	R/W	32	PM Sleep Control Y De-assert Delay (PM_SCYD)	Working	00000000h	Page 509
20h	R/W	32	PM PCI and IDE Input Sleep Control (PM_IN_SLPCTL)	Working	00000000h	Page 510
24h-2Ch	R/W	32	PM Reserved (PM_RSVD) (Reads as 0.)	No f/flops	00000000h	---
30h	R/W	32	PM Working De-assert Delay and Enable (PM_WKD)	Standby	00000000h	Page 511
34h	R/W	32	PM Working Auxiliary De-assert Delay and Enable (PM_WKXD)	Standby	00000000h	Page 512
38h	R/W	32	PM De-assert Reset Delay from Standby (PM_RD)	Standby	40000100h	Page 513
3Ch	R/W	32	PM Working Auxiliary Assert Delay from Standby Wakeup (PM_WKXA)	Standby	00000000h	Page 513
40h	R/W	32	PM Fail-Safe Delay and Enable (PM_FSD)	Standby	00000000h	Page 514
44h	R/W	32	PM Thermal-Safe Delay and Enable (PM_TSD)	Standby	00000000h	Page 515
48h	R/W	32	PM Power-Safe Delay and Enable (PM_PSD)	Standby	00000000h	Page 515
4Ch	R/W	32	PM Normal Work Delay and Enable (PM_NWKD)	Standby	00000000h	Page 516
50h	R/W	32	PM Abnormal Work Delay and Enable (PM_AWKD)	Standby	00000000h	Page 517
54h	R/W	32	PM Standby Status and Control (PM_SSC)	Standby	00000001h	Page 517
58h-7Fh	R/W	32	PM Reserved (PM_RSVD) (Reads as 0.)	No f/flops	00000000h	---

## 6.18.1 PMC Specific MSRs

### 6.18.1.1 PMC Logic Timer (PMC\_LTMR)

MSR Address 51400050h  
 Type R/W  
 Reset Value 00000000h

#### PMC\_LTMR Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSR_PML_TMR																															

#### PMC\_LTMR Bit Descriptions

Bit	Name	Description
31:0	MSR_PML_TMR	<b>Legacy Power Management Timer.</b> 32-Bit write / read of timer counter. Writes initialize the counter value; reads return current timer counter value.

### 6.18.1.2 PMC Reserved (PMC\_RSVD)

MSR Address 51400051h  
 Type R/W  
 Reset Value 00000000h

#### PMC\_RSVD Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMC_RSVD																															

#### PMC\_RSVD Bit Descriptions

Bit	Name	Description
31:0	PMC_RSVD	<b>Reserved.</b> This is a reserved register and should not be accessed by user software. By convention write 0, but other values are "don't care". Reads always return 0.

## 6.18.2 ACPI Registers

### 6.18.2.1 PM Status 1 (PM1\_STS)

ACPI I/O Offset 00h (WORD access only)  
 Type R/W  
 Reset Value 0000h

PM1\_STS is the Status register for Timer Carry, Button, and RTC Alarm wakeup events. All bits in this register are cleared by the Standby state except bits 15, 10, and 8. They maintain their state through Standby. This register must be accessed as WORD only. Byte, non-aligned WORD, or DWORD access is not allowed.

#### PM1\_STS Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAK_FLAG	RSVD			IGNORE	RTC_FLAG	SLPBTN_FLAG	PWRBTN_FLAG	RSVD		GBL_FLAG	BM_FLAG	RSVD			TMR_FLAG



## PM1\_STS Bit Descriptions

Bit	Name	Description
15	WAK_FLAG (Note 1, Note 2)	<b>Wakeup Event Flag.</b> This bit is set high by the hardware when any wakeup event occurs. Write 1 to clear; writing 0 has no effect.
14:12	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads always return 0.
11	IGNORE	<b>Ignore.</b> By ACPI convention not used by software. Reads always return 0. To support the Global Status Lock Flag bit, writing a 1 to this bit sets bit 5. Writing a 0 has no effect.
10	RTC_FLAG	<b>Real-Time Clock Alarm Flag.</b> This bit is set high by the hardware when the RTC generates an alarm. If RTC_EN (ACPI I/O Offset 02h[10]) is high, an SCI is generated. Write 1 to clear; writing 0 has no effect.
9	SLPBTN_FLAG	<b>Sleep Button Flag.</b> This bit is set high by the hardware when the “sleep button” is pushed. If SLPBTN_EN (ACPI I/O Offset 02h[9]) is high, an SCI is generated. Write 1 to clear; writing 0 has no effect.
8	PWRBTN_FLAG	<b>Power Button Flag.</b> This bit is set high by the hardware when the “power button” is pushed. If PWRBTN_EN (ACPI I/O Offset 02h[8]) is high, an SCI is generated. Write 1 to clear; writing 0 has no effect.
7:6	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads always return 0.
5	GBL_FLAG	<b>Global Lock Flag.</b> If high, indicates that the BIOS released control of Global Lock Status bit. This bit is cleared by writing a 1 to it. This bit is set by writing a 1 to bit 11 (IGNORE). If GLB_EN (ACPI I/O Offset 02h[5]) is high, an SCI is generated.
4	BM_FLAG	<b>Bus Master Flag.</b> This bit indicates a master has requested the bus. Used to indicate a possible incoherent cache when the processor is in state C3. This function is not supported because the Geode CS5535 companion device does not support the C3 state. By convention write 0, but other values are “don’t care”. Reads return 0.
3:1	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads always return 0.
0	TMR_FLAG	<b>Timer Carry Flag.</b> This bit is set high by the hardware anytime the Power Management Timer rolls over from all 1s back to 0. If TMR_EN (ACPI I/O Offset 02h[0]) is high, an SCI is generated when the rollover occurs. Write 1 to clear; writing 0 has no effect.

Note 1. A wakeup event can come from any event enabled by PM1\_EN (ACPI I/O Offset 02h) or PM\_GPE0\_EN (ACPI I/O Offset 1Ch). A wakeup will occur even if the SCI is not mapped to an ASMI or IRQ.

Note 2. After starting a Sleep sequence, software would normally spin by entering a polling loop on the WAK\_FLAG. This bit is normally (software has cleared it from last Sleep) 0 before starting a Sleep sequence. The Sleep sequence puts the processor in Suspend while it is spinning. When the sequence brings the processor out of Suspend, the WAK\_FLAG bit is set. The Sleep sequence starts when SLP\_EN (ACPI I/O Offset 08h[13]) is written to a 1.

### 6.18.2.2 PM Enable 1 (PM1\_EN)

ACPI I/O Offset 04h  
 Type R/W  
 Reset Value 0100h

PM1\_EN is the Enable register for Timer Carry, Button, and RTC Alarm wakeup events. All bits in this register are cleared by the Standby state except bits 10 and 8. They maintain their state through Standby. All bits in this register return the value written when read, except for the Reserved bits.

If enabled, any of the SCIs cause a wakeup event if the system state is Sleep or Standby (except TMR and GBL).

#### PM1\_EN Register Map

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					RTC_EN	SLPBTN_EN	PWRBTN_EN	RSVD			GLB_EN	RSVD			TMR_EN

#### PM1\_EN Bit Descriptions

Bit	Name	Description
15:11	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads return 0 value.
10	RTC_EN	<b>Real-Time Clock SCI Enable.</b> Enables generating an SCI when RTC_FLAG (ACPI I/O Offset 00h[10]) gets set. Also enables wakeup from this event. 0: Disable; 1: Enable.
9	SLPBTN_EN	<b>Sleep Button SCI Enable.</b> Enables generating an SCI when SLPBTN_FLAG (ACPI I/O Offset 00h[9]) gets set. Also enables wakeup from this event. 0: Disable; 1: Enable.
8	PWRBTN_EN	<b>Power Button SCI Enable.</b> Enables generating an SCI when PWRBTN_FLAG (ACPI I/O Offset 00h[8]) gets set. Also enables wakeup from this event. 0: Disable; 1: Enable (Default).
7:6	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads return 0 value.
5	GLB_EN	<b>Global Enable.</b> Enables generating an SCI when GLB_FLAG (ACPI I/O Offset 00h[5]) gets set. There is no wakeup concept for this event. 0: Disable; 1: Enable.
4:1	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads return 0 value.
0	TMR_EN	<b>Timer SCI Enable.</b> Enables generating an SCI when TMR_FLAG (ACPI I/O Offset 00h[0]) gets set. There is no wakeup concept for this event 0: Disable; 1: Enable.

**6.18.2.3 PM Control 1 (PM1\_CNT)**

ACPI I/O Offset 08h  
 Type R/W  
 Reset Value 0000h

PM1\_CNT is the Control register for global and the Sleep state settings.

**PM1\_CNT Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		SLP_EN	SLP_TYPx			IGNORE	RSVD						GBL_RLS	BM_RLD	SCI_EN

**PM1\_CNT Bit Descriptions**

Bit	Name	Description
15:14	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads always return 0.
13	SLP_EN (WO)	<b>Sleep Enable (Write Only).</b> This is a write-only bit and reads to it always return 0. Setting this bit causes the system to sequence into the Sleep state defined by SLP_TYPx (bits [12:10]).  After the delay in SLP_DELAY (PMS I/O Offset 00h[11:0]), the system state begins the move from Working to Sleeping or Standby state. The Sleep Request/Sleep Acknowledge sequenced is started. The sequence may be aborted by writing SLP_EN_INDIC (PMS I/O Offset 00h[15]).
12:10	SLP_TYPx	<b>Sleep Type.</b> Defines the type of Sleep state the system enters when SLP_EN (bit 13) is set to 1. Reads always return the value written.  These bits do not directly affect the internal hardware, but are required by the ACPI specification. When this register is accessed, VSA code traps the access and transfers bits written here to the appropriate locations to set up the desired power management mode.  The Sleep type is directly controlled by GLCP settings, individual GeodeLink Device Power Management MSR settings, and PML settings.
9	IGNORE	<b>Ignore.</b> By convention not used by ACPI software. Software always writes 0.  If a 1 was written to bit 2 (GBL_RLS), this bit is set, that is, a read of this bit returns a 1. Write 1 to clear; writing 0 has no effect.
8:3	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads always return 0.
2	GBL_RLS	<b>Global Lock Release.</b> This is the “release of global lock” bit. The ACPI driver writes this bit to a 1 to rise an event to the BIOS. The write indicates the release of global lock. Reads always return 0.  Writing a 1 to this bit sets bit 9 (IGNORE).
1	BM_RLD	<b>Bus Master RLD.</b> When high, this bit allows the generation of a bus master request to cause any processor in the C3 state to transition to the C0 state. Reads return the value written. In the Geode CS5535 companion device, the C3 state is NOT supported. Other than serving as an indicator, this bit does nothing.
0	SCI_EN	<b>SCI Enable.</b> When low, indicates native power management mode. When high, indicates ACPI mode. Reads return the value written. Other than serving as an indicator, this bit does not directly affect the hardware.

**6.18.2.4 PM Control 2 (PM2\_CNT)**

ACPI I/O Offset 0Ch  
 Type R/W  
 Reset Value 0000h

PM2\_CNT is the Control register for enabling/disabling the system arbiter. This register is not implemented. Writes “don’t care”. Reads return 0. This register may be accessed with 8-bit or 16-bit I/O.

**PM2\_CNT Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								RSVD							ARB_DIS

**PM2\_CNT Bit Descriptions**

Bit	Name	Description
15:8	RSVD	<b>Reserved.</b> ACPI defines this as an 8-bit register. It has been extended so that all PML registers are at least 16 bits. Writes to these bits are a “don’t care”. Reads always return 0.
7:1	RSVD	<b>Reserved.</b> By convention write 0, but other values are “don’t care”. Reads return 0 value.
0	ARB_DIS	<b>System Arbiter Disable.</b> Disables when high. Reads return value written. This bit is required by the ACPI specification, but internally is not connected to any PM logic.

**6.18.2.5 PM Timer (PM\_TMR)**

ACPI I/O Offset 10h  
 Type RO  
 Reset Value 00000000h

PM\_TMR is the data value register for the 32-bit timer running from the 3.579 MHz clock.

**PM\_TMR Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR_VAL																															

**PM\_TMR Bit Descriptions**

Bit	Name	Description
31:0	TMR_VAL (RO)	<b>Timer Value (Read Only).</b> This read only counter is driven by the 3.579545 MHz clock. Writes are always a “don’t care”.  The counter runs continuously as long as the system is in the Working state; otherwise, counting is stopped. It stops counting when SUSP# is asserted and starts counting when SUSPA# has been de-asserted after having been asserted. The value in this register is lost in the Standby state.

### 6.18.2.6 General Purpose Events Status 0 (PM\_GPE0\_STS)

ACPI I/O Offset 18h  
 Type R/W  
 Reset Value 00000000h

PM\_GPE0\_STS is the Status register for General Purpose Events. Status events are cleared by writing a 1 to the appropriate FLAG bit. Writing 0 has no effect. By convention, bits [23:0] are associated with the Working domain while bits [31:24] are associated with Standby domain. During Standby, bits [23:0] are unconditionally cleared. These events are all individually enabled and then ORed together to form the System Control Interrupt (SCI).

**PM\_GPE0\_STS Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPIOM7_PME_FLAG	GPIOM6_PME_FLAG	RSVD									GPIOM5_PME_FLAG	GPIOM4_PME_FLAG	GPIOM3_PME_FLAG	GPIOM2_PME_FLAG	GPIOM1_PME_FLAG	GPIOM0_PME_FLAG	RSVD									USBC2_PME_FLAG	USBC1_PME_FLAG	UART2_PME_FLAG	UART1_PME_FLAG	SMB+PME_FLAG	PIC_ASMI_PME_FLAG	PIC_IRQ_PME_FLAG

**PM\_GPE0\_STS Bit Descriptions**

Bit	Name	Description
31	GPIOM7_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 7 PME Flag.</b> If high, this bit records that a PME occurred via bit 7 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[31]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
30	GPIOM6_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 6 PME Flag.</b> If high, this bit records that a PME occurred via bit 6 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[30]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
29:22	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect
21	GPIOM5_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 5 PME Flag.</b> If high, this bit records that a PME occurred via bit 5 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[21]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
20	GPIOM4_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 4 PME Flag.</b> If high, this bit records that a PME occurred via bit 4 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[20]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
19	GPIOM3_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 3 PME Flag.</b> If high, this bit records that a PME occurred via bit 3 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[19]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
18	GPIOM2_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 2 PME Flag.</b> If high, this bit records that a PME occurred via bit 2 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[18]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
17	GPIOM1_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 1 PME Flag.</b> If high, this bit records that a PME occurred via bit 1 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[17]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.

## PM\_GPE0\_STS Bit Descriptions

Bit	Name	Description
16	GPIOM0_PME_FLAG	<b>GPIO IRQ/PME Mapper Bit 0PME Flag.</b> If high, this bit records that a PME occurred via bit 0 of the GPIO IRQ/PME mapper. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[16]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
15:7	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect
6	USBC2_PME_FLAG	<b>USB Controller #2 PME Flag.</b> If high, this bit records that a PME occurred via USB Controller #2. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[6]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
5	USBC1_PME_FLAG	<b>USB Controller #1 PME Flag.</b> If high, this bit records that a PME occurred via USB Controller #1. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[5]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
4	UART2_PME_FLAG	<b>UART #2 PME Flag.</b> If high, this bit records that a PME occurred via UART #2. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[4]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
3	UART1_PME_FLAG	<b>UART #1 PME Flag.</b> If high, this bit records that a PME occurred via UART #1. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[3]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
2	SMB_PME_FLAG	<b>SMB PME Flag.</b> If high, this bit records that a PME occurred via the SMB. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[2]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
1	PIC_ASMI_PME_FLAG	<b>PIC ASMI PME Flag.</b> If high, this bit records that a PME occurred due to a PIC ASMI. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[1]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.
0	PIC_IRQ_PME_FLAG	<b>PIC Interrupt PME Flag.</b> If high, this bit records that a PME occurred due to a PIC Interrupt. Both this bit and the corresponding enable bit in PM_GPE0_EN (ACPI I/O Offset 1Ch[0]) must be high in order for this PME to be passed on to the system. Write 1 to clear; writing 0 has no effect.

**6.18.2.7 General Purpose Events Enable 0 (PM\_GPE0\_EN)**

ACPI I/O Offset 1Ch  
 Type R/W  
 Reset Value 00000000h

PM\_GPE0\_EN is the Enable register for General Purpose Events. Reads always return the value written. By convention, bits [23:0] are associated with the Working domain while bits [31:24] are associated with the Standby domain. During Standby, bits [23:0] are unconditionally cleared. PME status can be read via the corresponding FLAG bit in the PM\_GPE0\_STS register (ACPI I/O Offset 18h).

**PM\_GPE0\_EN Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPIOM7_PME_EN	GPIOM6_PME_EN	RSVD									GPIOM5_PME_EN	GPIOM4_PME_EN	GPIOM3_PME_EN	GPIOM2_PME_EN	GPIOM1_PME_EN	GPIOM0_PME_EN	RSVD									USBC2_PME_EN	USBC1_PME_EN	UART2_PME_EN	UART1_PME_EN	SMB+PME_EN	PIC_ASMI_PME_EN	PIC_IRQ_PME_EN

**PM\_GPE0\_EN Bit Descriptions**

Bit	Name	Description
31	GPIOM7_PME_EN	<b>GPIO IRQ/PME Mapper Bit 7 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 7 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
30	GPIOM6_PME_EN	<b>GPIO IRQ/PME Mapper Bit 6 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 6 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
29:22	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect
21	GPIOM5_PME_EN	<b>GPIO IRQ/PME Mapper Bit 5 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 5 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
20	GPIOM4_PME_EN	<b>GPIO IRQ/PME Mapper Bit 4 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 4 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
19	GPIOM3_PME_EN	<b>GPIO IRQ/PME Mapper Bit 3 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 3 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
18	GPIOM2_PME_EN	<b>GPIO IRQ/PME Mapper Bit 2 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 2 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
17	GPIOM1_PME_EN	<b>GPIO IRQ/PME Mapper Bit 1 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 1 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
16	GPIOM0_PME_EN	<b>GPIO IRQ/PME Mapper Bit 0 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via bit 0 of the GPIO IRQ/PME mapper. Write this bit low to disable the generation of a PME from this source.
15:7	RSVD	<b>Reserved.</b> Reads return 0; writes have no effect
6	USBC2_PME_EN	<b>USB Controller #2 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via USB Controller #2. Write this bit low to disable the generation of a PME from this source.
5	USBC1_PME_EN	<b>USB Controller #1 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via USB Controller #2. Write this bit low to disable the generation of a PME from this source.

**PM\_GPE0\_EN Bit Descriptions (Continued)**

Bit	Name	Description
4	UART2_PME_EN	<b>UART #2 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via UART #2. Write this bit low to disable the generation of a PME from this source.
3	UART1_PME_EN	<b>UART #1 PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via UART #2. Write this bit low to disable the generation of a PME from this source.
2	SMB_PME_EN	<b>SMB PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs via the SMB. Write this bit low to disable the generation of a PME from this source.
1	PIC_ASMI_PME_EN	<b>PIC ASMI PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs due to a PIC ASMI. Write this bit low to disable the generation of a PME from this source.
0	PIC_IRQ_PME_EN	<b>PIC Interrupt PME Enable.</b> When set high, this bit enables the generation of a PME to the system if a PME occurs due to a PIC Interrupt. Write this bit low to disable the generation of a PME from this source.

**6.18.3 PM Support Registers**

The registers listed in this sub-section are not ACPI registers, but are used to support power management implementation.

**6.18.3.1 PM Sleep Start Delay (PM\_SSD)**

PMS I/O Offset 00h  
 Type R/W  
 Reset Value 0000h

**PM\_SSD Register Map**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP_EN_INDIC	RSVD	SLP_WRT_EN	SLP_DELAY_EN	SLP_DELAY											

**PM\_SSD Bit Descriptions**

Bit	Name	Description
15	SLP_EN_INDIC	<b>Sleep Enable Indicator.</b> If SLP_EN (ACPI I/O Offset 08h[13]) was written to a 1, then this bit reads high. If this bit is written to a 1, the Sleep sequence is aborted. Writing a 0 to this bit has no effect. This bit always clears on a Sleep or Standby wakeup.
14	RSVD	<b>Reserved.</b> By convention write 0. Reads return value written.
13	SLP_WRT_EN	<b>Sleep Write Enable.</b> Must be high in order to change bits 12 and [11:0] (SLP_DELAY_EN and SLP_DELAY). Reads of this bit always return 0.
12	SLP_DELAY_EN	<b>Sleep Delay Enable.</b> Must be high to enable the delay specified in bits [11:0] (SLP_DELAY). Reads return value written.
11:0	SLP_DELAY	<b>Sleep Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait before beginning the Sleep or Standby process as defined by SLP_EN (ACPI I/O Offset 08h[13]). Bit 12 (SLP_DELAY_EN) must be high to enable this delay. Reads return the value written.



**6.18.3.2 PM Sleep Control X Assert Delay and Enable (PM\_SCXA)**

PMS I/O Offset 04h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_SCXA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLPX_EN	SLPX_DELAY																													

**PM\_SCXA Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPX_EN	<b>Sleep X Assert and Delay Enable.</b> Must be high to assert the SLEEP_X ball and to enable its assert delay specified in bits [29:0] (SLPX_DELAY).
29:0	SLPX_DELAY	<b>Sleep X Assert Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from the assertion of SUSPA# before asserting the SLEEP_X ball. Bit 30 (SLPX_EN) must be high to enable this delay.  SLEEP_X is not allowed to assert if this delay is larger than SLPCLK_DELAY (PMS I/O Offset 10h[29:0]). This is only true if SLPCLK_EN is enabled (PMS I/O Offset 10h[30] = 1).

**6.18.3.3 PM Sleep Control Y Assert Delay and Enable (PM\_SCYA)**

PMS I/O Offset 08h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_SCYA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLPY_EN	SLPY_DELAY																													

**PM\_SCYA Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPY_EN	<b>Sleep Y Assert and Delay Enable.</b> Must be high to assert SLEEP_Y and enable its assert delay specified in bits [29:0] (SLPY_DELAY).
29:0	SLPY_DELAY	<b>Sleep Y Assert Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from the assertion of SUSPA# before asserting the SLEEP_Y ball. Bit 30 (SLPY_EN) must be high to enable this delay.  SLEEP_Y is not allowed to assert if this delay is larger than SLPCLK_DELAY (PMS I/O Offset 10h[29:0]). This is only true if SLPCLK_EN is enabled (PMS I/O Offset 10h[30] = 1).

**6.18.3.4 PM Sleep Output Disable Assert Delay and Enable (PM\_OUT\_SLPCTL)**

PMS I/O Offset 0Ch  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_OUT\_SLPCTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	PCI_IDE_OUT_SLP	PCI_IDE_OUT_SLP_DELAY																													

**PM\_OUT\_SLPCTL Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	PCI_IDE_OUT_SLP	<b>PCI/IDE Output Sleep Control.</b> Allows the delay specified in bits [29:0] (PCI_IDE_OUT_SLP_DELAY) to turn off PCI/IDE outputs as listed in Table 4-11 "Sleep Driven PCI Signals" and Table 4-12 "Sleep Driven IDE Signals" on page 79. Individual enables exist for PCI (PCI_GLD_MSR_PM, MSR 51000004h[49:48]) and IDE (IDE_GLD_MSR_PM, MSR 51300004h[49:48]). Output control immediately enables the PCI/IDE outputs when SUSP# de-asserts.  0: Disable. 1: Enable.
29:0	PCI_IDE_OUT_SLP_DELAY	<b>PCI/IDE Output Sleep Control Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from Sleep wakeup before PCI/IDE outputs are disabled. Bit 30 (PCI_IDE_OUT_SLP) must be high to enable this delay.  The PCI/IDE outputs will not turn off if this delay is larger than SLPCLK_DELAY (PMS I/O Offset 10h[29:0]). This is only true if SLPCLK_EN is enabled (PMS I/O Offset 10h[30] = 1).

**6.18.3.5 PM Sleep Clock Delay and Enable (PM\_SCLK)**

PMS I/O Offset 10h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_SCLK Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLPCLK_EN	SLPCLK_DELAY																													

**PM\_SCLK Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPCLK_EN	<b>Sleep Clock Delay Enable.</b> Must be high to assert SLP_CLK_EN# and enable its assert delay specified in bits [29:0] (SLPCLK_DELAY). Use of this control is required but not sufficient to enter the Standby state.  <b>WARNING:</b> Using this control immediately turns off all system clocks except the 32 kHz RTC clock.
29:0	SLPCLK_DELAY	<b>Sleep Clock Assert Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from the assertion of SUSPA# before asserting SLP_CLK_EN#. Bit 30 (SLPCLK_EN) must be high to enable this delay.  There is NOT a de-assert delay. The wakeup event causes SLP_CLK_EN# to de-assert combinatorially from the wakeup event. This event is called Sleep wakeup. The concept of a Sleep wakeup applies even if Sleep Clock is not used.

**6.18.3.6 PM Sleep End Delay (PM\_SED)**

PMS I/O Offset 14h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_SED Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLPEND_EN	SLPEND_DELAY																													

## PM\_SED Bit Descriptions

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPEND_EN	<b>Sleep End Delay Enable.</b> Must be high to enable the delay specified in bits [29:0] (SLPEND_DELAY).
29:0	SLPEND_DELAY	<b>Sleep End Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from Sleep wakeup before de-asserting SUSP#. Bit 30 (SLPEND_EN) must be high to enable this delay.  If PCI_IDE_IN_SLP is not enabled (PMS I/O Offset 20h[30] = 0) or the delay is less than SLPEND_DELAY, SUSP# de-asserts at the same time the PCI/IDE inputs are re-enabled.

## 6.18.3.7 PM Sleep Control X De-assert Delay (PM\_SCXD)

PMS I/O Offset 18h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

## PM\_SCXD Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RSVD	SLPX_END_EN	SLPX_END_DELAY																																

## PM\_SCXD Bit Descriptions

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPX_END_EN	<b>Sleep X De-assert and Delay Enable.</b> Must be high to de-assert SLEEP_X and enable the delay specified in bits [29:0] (SLPX_END_DELAY).
29:0	SLPX_END_DELAY	<b>Sleep X De-assert Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from Sleep wakeup before de-asserting the SLEEP_X ball. Bit 30 (SLPX_END_EN) must be high to enable this delay.  If PCI_IDE_IN_SLP is not enabled (PMS I/O Offset 20h[30] = 0) or is less than the SLPX_END_DELAY, SLEEP_X de-asserts at the same time the PCI/IDE inputs are re-enabled.

**6.18.3.8 PM Sleep Control Y De-assert Delay (PM\_SCYD)**

PMS I/O Offset 1Ch  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_SCYD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	SLPY_END_EN	SLPY_END_DELAY																													

**PM\_SCYD Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	SLPY_END_EN	<b>Sleep Y De-assert and Delay Enable.</b> Must be high to de-assert SLEEP_Y and enable the delay specified in bits [29:0] (SLPY_END_DELAY).
29:0	SLPY_END_DELAY	<b>Sleep Control Y De-assert Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from Sleep wakeup before de-asserting the SLEEP_Y ball. Bit 30 (SLPY_END_EN) must be high to enable this delay.  If PCI_IDE_IN_SLP is not enabled (PMS I/O Offset 20h[30] = 0) or is less than the SLPY_END_DELAY, SLEEP_Y de-asserts at the same time the PCI/IDE inputs are re-enabled.

**6.18.3.9 PM PCI and IDE Input Sleep Control (PM\_IN\_SLPCTL)**

PMS I/O Offset 20h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written.

**PM\_IN\_SLPCTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	PCI_IDE_IN_SLP	PCI_IDE_IN_SLP_DELAY																													

**PM\_IN\_SLPCTL Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	PCI_IDE_IN_SLP	<b>PCI/IDE Input Sleep Control.</b> Allows the delay specified in bits [29:0] (PCI_IDE_IN_SLP_DELAY) to turn off PCI/IDE inputs as listed in Table 4-11 "Sleep Driven PCI Signals" and Table 4-12 "Sleep Driven IDE Signals" on page 79. Individual enables exist for PCI (PCI_GLD_MSR_PM, MSR 51000004h[49:48]) and IDE (IDE_GLD_MSR_PM, MSR 51300004h[49:48]).  0: Disable. 1: Enable.
29:0	PCI_IDE_IN_SLP_DELAY	<b>PCI/IDE Input Sleep Control Delay.</b> Indicates the number of 3.57954 MHz clock edges to wait from Sleep wakeup before PCI/IDE inputs are disabled. Bit 30 (PCI_IDE_IN_SLP) must be high to enable this delay.

**6.18.3.10 PM Working De-assert Delay and Enable (PM\_WKD)**

PMS I/O Offset 30h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_WKD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	WORKING_DEASSERT_EN	RSVD										WORKING_DEASSERT_DELAY																			

**PM\_WKD Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	WORKING_DEASSERT_EN	<p><b>Working De-assert and Delay Enable.</b> Must be high to de-assert the WORKING output and to enable its delay specified in bits [19:0] (WORKING_DEASSERT_DELAY). Use of this control implies a system sequence into the Standby state. The PMC disables its interfaces to non-Standby portions of the component and only considers wakeup events from Standby circuits. The PMC also immediately asserts system reset when SLP_CLK_EN# is asserted regardless of the value of WORKING_DEASSERT_DELAY (bits [19:0]). Reset remains asserted throughout the Standby state.</p> <p>There is NOT an assert delay. The wakeup event causes the WORKING output to assert. This event is called Standby wakeup.</p> <p>On wakeup, Reset will continue to be applied to all non-Standby circuits for the length of time specified in the RESET_DELAY (PMS I/O Offset 38h[19:0]).</p> <p>Enabling this function and/or the function in PM_WKXD (PMS I/O Offset 34h[30] = 1) causes the same Standby state events. Standby state is not entered unless SLP_CLK_EN# is asserted.</p>
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	WORKING_DEASSERT_DELAY	<p><b>Working De-assert Delay.</b> Indicates the number of 32 kHz clock edges to wait from the assertion of SLP_CLK_EN# before de-asserting the WORKING output. Bit 30 (WORKING_DEASSERT_EN) must be high to enable this delay.</p>

**6.18.3.11 PM Working Auxiliary De-assert Delay and Enable (PM\_WKXD)**

PMS I/O Offset 34h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_WKXD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	WORK_AUX_DEASSERT_EN	RSVD										WORK_AUX_DEASSERT_DELAY																			

**PM\_WKXD Bit Descriptions**

Bit	Name	Description
31	RSVD	<b>Reserved.</b> By convention write 0, but may write anything.
30	WORK_AUX_DEASSERT_EN	<p><b>Working Auxiliary De-assert and Delay Enable.</b> Must be high to de-assert the WORK_AUX output and enable its delay specified in bits [19:0] (WORK_AUX_DEASSERT_DELAY).</p> <p>Use of this control implies a system sequence into the Standby State. The PMC disables its interfaces to non-Standby portions of the component and only considers wakeup events from Standby circuits. The PMC also immediately asserts system reset when SLP_CLK_EN# is asserted regardless of the value of WORK_AUX_DEASSERT_DELAY (bits [19:0]). Reset remains asserted throughout the Standby state.</p> <p>There is NOT an assert delay. The wakeup event causes the WORK_AUX output to assert. This event is called Standby wakeup.</p> <p>On wakeup, Reset continues to be applied to all non-Standby circuits for the length of time specified in RESET_DELAY (PMS I/O Offset 38h[19:0]).</p> <p>Enabling this function and/or the function in PM_WKD (PMS I/O Offset 30h[30] = 1) causes the same Standby state events. Standby state is not entered unless SLP_CLK_EN# is asserted.</p>
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	WORK_AUX_DEASSERT_DELAY	<p><b>WORK_AUX De-assert Delay.</b> Indicates the number of 32 kHz clock edges to wait from the assertion of SLP_CLK_EN# before de-asserting the WORK_AUX output. Bit 30 (WORK_AUX_DEASSERT_EN) must be high to enable this delay.</p>



**6.18.3.12 PM De-assert Reset Delay from Standby (PM\_RD)**

PMS I/O Offset 38h  
 Type R/W  
 Reset Value 40000100h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_RD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET_LOCK	RESET_EN	RSVD										RESET_DELAY																			

**PM\_RD Bit Descriptions**

Bit	Name	Description
31	RESET_LOCK	<b>Reset Lock.</b> After this bit is set, the value of PM_RD can not be changed until RESET_STAND# is applied.
30	RESET_EN	<b>Reset Delay Enable.</b> Must be high for the RESET_OUT# output de-assert delay specified in bits [19:0] (RESET_DELAY) to be applied. (Default = 1.)
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	RESET_DELAY	<b>Reset De-assert Delay.</b> Indicates the number of 32 kHz clock edges to continue asserting RESET_OUT# from Standby Wakeup. Default value is 8 ms. This delay starts only if the RESET_WORK# input is de-asserted and the internal Low Voltage Detect circuit detects normal operating voltages on V <sub>CORE</sub> . (Default = 00100h.) (See Section 4.6 "Reset Considerations" on page 65 for further details regarding reset conditions.) Bit 30 (RESET_EN) must be high to enable this delay.  Reset will be applied to the system for the longer of this value or until the internal Low Voltage Detect circuit detects normal operating voltages on V <sub>CORE</sub>

**6.18.3.13 PM Working Auxiliary Assert Delay from Standby Wakeup (PM\_WKXA)**

PMS I/O Offset 3Ch  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_WKXA Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORK_AUX_LOCK	WORK_AUX_EN	RSVD										WORK_AUX_DELAY																			

## PM\_WKXA Bit Descriptions

Bit	Name	Description
31	WORK_AUX_LOCK	<b>Working Auxiliary Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	WORK_AUX_EN	<b>Working Auxiliary Delay Enable.</b> Must be high to enable the delay specified in bits [19:0] (WORK_AUX_DELAY). If this bit is low, the WORK_AUX output is unconditionally asserted at Standby wakeup. If WORK_AUX was not de-asserted going into Standby, then this control is a “don’t care”.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	WORK_AUX_DELAY	<b>Working Auxiliary Assert Delay.</b> Indicates the number of 32 kHz clock edges to wait from Standby wakeup before asserting the WORK_AUX output. Bit 30 (WORK_AUX_EN) must be high to enable this delay.  May be programmed to assert before or after RESET_OUT# de-asserts.  The Standby wakeup event is not recognized until Normal (NWKD) or Abnormal (AWKD) to Work Delay expires, if those delays are enabled. (See PMS I/O Offset 4Ch and 50h for details regarding the NWKD and AWKD registers.)

## 6.18.3.14 PM Fail-Safe Delay and Enable (PM\_FSD)

PMS I/O Offset    40h  
 Type                R/W  
 Reset Value        00000000h

Reads always return the value written, except for RSVD bits [29:20].

## PM\_FSD Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWRBUT_LOCK	PWRBUT_EN	RSVD										PWRBUT_DELAY																			

## PM\_FSD Bit Descriptions

Bit	Name	Description
31	PWRBUT_LOCK	<b>Power Button Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	PWRBUT_EN	<b>Power Button Enable.</b> Must be high to enable the fail-safe function.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	PWRBUT_DELAY	<b>Power Button Delay.</b> If the Power Button (PWR_BUT#) input signal is asserted for PWRBUT_DELAY number of 32 kHz clock edges, then unconditionally de-assert WORKING and WORK_AUX to remove Working power. If PWR_BUT# is still asserted at wakeup, hold in Standby state until de-asserted.  PWR_BUT# needs to be asserted for at least one 32 kHz clock edge for this function to work properly. A less than one 32 kHz clock edge pulse on PWR_BUT# may not be registered.  The delay restarts if PWR_BUT# de-asserts and then asserts again. If PWR_BUT# is already asserted, the delay restarts anytime PWRBUT_DELAY (this field) is written.

**6.18.3.15 PM Thermal-Safe Delay and Enable (PM\_TSD)**

PMS I/O Offset 44h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_TSD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THRM_LOCK	THRM_EN	RSVD										THRM_DELAY																			

**PM\_TSD Bit Descriptions**

Bit	Name	Description
31	THRM_LOCK	<b>Thermal Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	THRM_EN	<b>Thermal Enable.</b> Must be high to enable the thermal alarm function.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0 value.
19:0	THRM_DELAY	<p><b>Thermal Delay.</b> If the Thermal Alarm (THRM_ALARM#) input signal is asserted for THRM_DELAY number of 32 kHz clock edges, then unconditionally de-assert WORKING and WORK_AUX to remove Working power. If THRM_ALARM# is still asserted at wakeup, hold in Standby state until THRM_ALARM# is de-asserted.</p> <p>THRM_ALARM# needs to be asserted for at least one 32 kHz clock edge for this function to work properly. A less than one 32 kHz clock edge pulse on THRM_ALARM# may not be registered.</p> <p>The delay restarts if THRM_ALARM# de-asserts and then asserts again. If THRM_ALARM# is asserted, the delay restarts anytime THRM_DELAY (this bit) is written.</p> <p>Since the thermal alarm resides in the Working domain, the THRM_ALARM# input signal is blocked (de-asserted) when in Standby state. The result is the Standby state could not be held if the thermal alarm is still asserted at wakeup. Once out of Standby, the thermal alarm again comes into play. If it is still asserted, its timer would start again.</p>

**6.18.3.16 PM Power-Safe Delay and Enable (PM\_PSD)**

PMS I/O Offset 48h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20].

**PM\_PSD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOWBAT_LOCK	LOWBAT_EN	RSVD										LOWBAT_DELAY																			

## PM\_PSD Bit Descriptions

Bit	Name	Description
31	LOWBAT_LOCK	<b>Low Battery Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	LOWBAT_EN	<b>Low Battery Enable.</b> Must be high to enable this function.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	LOWBAT_DELAY	<b>Low Battery Delay.</b> If the Low Battery input LOW_BAT# (GPIO25) is asserted for LOWBAT_DELAY number of 32 kHz clock edges, then unconditionally de-assert WORKING and WORK_AUX to remove Working power. If LOW_BAT# is still asserted at wakeup, hold in Standby state until LOW_BAT# is de-asserted.  LOW_BAT# needs to be asserted for at least one 32 kHz clock edge for this function to work properly. A less than one 32 kHz clock edge pulse on the LOW_BAT# input may not be registered.  The delay restarts if LOW_BAT# de-asserts and then asserts again. If LOW_BAT# is already asserted, the delay restarts anytime LOWBAT_DELAY (this field) is written.

## 6.18.3.17 PM Normal Work Delay and Enable (PM\_NWKD)

PMS I/O Offset 4Ch  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20]. This register applies to Normal Standby state entry.

## PM\_NWKD Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NWKD_LOCK	NWKD_EN	RSVD										NWKD_DELAY																			

## PM\_NWKD Bit Descriptions

Bit	Name	Description
31	NWKD_LOCK	<b>Normal Work Delay Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	NWKD_EN	<b>Normal Work Delay Enable.</b> Must be high to enable this function.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	NWKD_DELAY	<b>Normal Work Delay Delay.</b> If the Standby state is entered normally, NWKD_DELAY number of 32 kHz clock edges must pass before the Working state is allowed to be entered again, that is, a Standby Wakeup recognized.

**6.18.3.18 PM Abnormal Work Delay and Enable (PM\_AWKD)**

PMS I/O Offset 50h  
 Type R/W  
 Reset Value 00000000h

Reads always return the value written, except for RSVD bits [29:20]. This register applies to Abnormal Standby state entry.

**PM\_AWKD Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWKD_LOCK	AWKD_EN	RSVD										AWKD_DELAY																			

**PM\_AWKD Bit Descriptions**

Bit	Name	Description
31	AWKD_LOCK	<b>Abnormal Work Delay Lock.</b> After this bit is set, the value in this register can not be changed until RESET_STAND# is applied.
30	AWKD_EN	<b>Abnormal Work Delay Enable.</b> Must be high to enable this function.
29:20	RSVD	<b>Reserved.</b> By convention write 0, but may write anything. Reads return 0.
19:0	AWKD_DELAY	<b>Abnormal Work Delay Delay.</b> If the standby state is entered abnormally, AWKD_DELAY number of 32 kHz clock edges must pass before the Working state is allowed to be entered again, that is, a Standby Wakeup recognized.

**6.18.3.19 PM Standby Status and Control (PM\_SSC)**

PMS I/O Offset 54h  
 Type R/W  
 Reset Value 00000001h

**PM\_SSC Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
RSVD													PI	CLEAR_PI	SET_PI	RSVD										BAD_PACK_RST_FLAG	GLCP_SFT_RST_FLAG	WATCHDOG_RST_FLAG	SHTDWN_RST_FLAG	SFT_RST_FLAG	RSVD	HRD_RST_FLAG	LOWBAT_FLAG	THRM_FLAG	PWRBTN_FLAG	LVD_FLAG	NORM_FLAG	OFF_FLAG

## PM\_SSC Bit Descriptions

Bit	Name	Description
31:19	RSVD	<b>Reserved.</b> Reads return 0; writes are don't care.
18	PI (RO)	<b>Power Immediate (Read Only).</b> Reads return current value of Power Immediate bit.
17	CLEAR_PI	<b>Clear Power Immediate.</b> Write 1 to clear the read only Power Immediate bit (bit 18). Writing 0 has no effect. Reads return 0.
16	SET_PI	<b>Set Power Immediate.</b> Write 1 to set the read only Power Immediate bit (bit 18). Writing 0 has no effect. Reads return 0.
15:13	RSVD	<b>Reserved.</b> Reads return 0; writes are don't care.
12	BADPACK_RST_FLAG	<b>Bad Packet Reset Flag.</b> If set, indicates that the last Standby state was entered from bad packet type reset. Returns to Working state when Abnormal Work Delay expires. (Note 1)
11	GLCP_SFT_RST_FLAG	<b>GLCP Soft Reset Flag.</b> If set, indicates that the last Standby state was entered from a GLCP soft reset. Returns to Working State when Abnormal Work Delay expires. (Note 1)
10	WATCHDOG_RST_FLAG	<b>Watchdog Reset Flag.</b> If set, indicates that the last Standby state was entered from a Watchdog reset. Returns to Working state when Abnormal Work Delay expires. (Note 1)
9	SHTDWN_RST_FLAG	<b>Shutdown Reset Flag.</b> If set, indicates that the last Standby state was entered from shutdown reset. Returns to Working state when Abnormal Work Delay expires. (Note 1)
8	SFT_RST_FLAG	<b>Soft Reset Flag.</b> If set, indicates that the last Standby state was entered from a software reset. Returns to Working state when Abnormal Work Delay expires. (Note 1)
7	RSVD	<b>Reserved.</b> Reads return 0; writes are don't care.
6	HRD_RST_FLAG	<b>Hard Reset Flag.</b> If set, indicates that the last Standby state was entered due to the unexpected assertion of Working reset. Returns to Working state when hard reset is de-asserted and Abnormal Work Delay expires. (Note 1)
5	LOWBAT_FLAG	<b>Low Battery Flag.</b> If set, indicates that the last Standby state was entered due to a low power shutdown. Returns to Working state due to default wakeup. (Note 1)
4	THRM_FLAG	<b>Thermal Flag.</b> If set, indicates that the last Standby state was entered due to a thermal shutdown. Returns to Working state due to default wakeup. (Note 1)
3	PWRBTN_FLAG	<b>Power Button Flag.</b> If set, indicates that the last Standby state was entered via a fail-safe power off sequence. User held down the power button. PM1_CNT was not used. Returns to Working state due to default wakeup. (Note 1)
2	LVD_FLAG	<b>Working Power Fail.</b> If set, indicates that the last Standby state was entered via an unexpected loss of Working power as detected with the on-chip Low Voltage Detect circuit. Returns to Working state due to default wakeup. (Note 1)
1	NORM_FLAG	<b>Normal Flag.</b> If set, indicates that the last Standby state was entered under program control through use of PM1_CNT. Returns to Working state due to programmed wakeup. See PM1_STS (ACPI I/O Offset 00h) and PM_GPE0_STS (ACPI I/O Offset 18h) for wakeup source. (Note 1)
0	OFF_FLAG	<b>Off Flag (No Previous).</b> If set, indicates that the circuits of the Standby power domain have been reset. Entry was from the Power Off state. Returns to Working state due to default wakeup. (Note 1)

Note 1. **Standby Status.** These bits are cleared each time the Power Management Logic enters the Standby state except for the bit that caused the entry. Write 1 to the bit to clear the bit; writing 0 has no effect. Bits [12:8,6] do not result in WORKING or WORK\_AUX being de-asserted.

## 6.19 Flash Controller Register Descriptions

The registers for the Flash Controller are divided into three sets:

- Standard GeodeLink Device (GLD) MSRs (Shared with DIVIL, see Section 6.6.1 on page 317.)
- Flash Controller Specific MSRs
- Flash Controller Native Registers

The MSRs are accessed via the RDMSR and WRMSR processor instructions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

All MSRs are 64 bits, however, the Flash Controller Specific MSRs (summarized in Table 6-69) are called out as 32 bits. The Flash Controller treats writes to the upper 32 bits (i.e., bits [63:32]) of the 64-bit MSRs as don't cares and always returns 0 on these bits.

The Native registers associated with the Flash Controller are NAND configuration registers, summarized in Table 6-70. The NAND native registers are 4-kbyte memory mapped or 16-byte I/O mapped. The base address is defined by LBAR in Diverse Device and can be located at any 4-kbyte boundaries if it is memory mapped, any 16-byte boundary if it is I/O mapped. The NAND Flash Controller is a 32-bit wide device present in Diverse Device without burst capability. To access the MSR registers in the NAND Flash Controller, a 32-bit wide bus is used as the LBus interface. For NAND Command/Address, data write

and read modes, the NAND Flash Controller provides the valid data on the least significant nibbles of the LBus data ports.

There are no NOR control registers located in I/O or memory space. All NOR timing control functions are located in the Flash Specific MSRs. Additionally, the Diverse Device LBAR MSRs associates up to four chip selects for four Flash devices (see Section 6.6.2.9 "Local BAR - Flash Chip Select (DIVIL\_LBAR\_FLSH[x])" on page 332 bit details).

- MSR\_LBAR\_FLSH0 (MSR 51400010h) for use with FLASH\_CS0#.
- MSR\_LBAR\_FLSH1 (MSR 51400011h) for use with FLASH\_CS1#.
- MSR\_LBAR\_FLSH2 (MSR 51400012h) for use with FLASH\_CS2#.
- MSR\_LBAR\_FLSH3 (MSR 51400013h) for use with FLASH\_CS3#.

After the MSR setup is complete, a NOR Flash device can be associated with a block of system memory using up to 28 address bits (A[27:0]).

The reference column in the summary tables point to the page where the register maps and bit descriptions are listed.

**Table 6-69. Flash Controller Specific MSRs Summary**

MSR Address	Type	Register Name	Reset Value	Reference
51400018h	R/W	NOR Flash Control (NORF_CTL)	00000000h	Page 520
51400019h	R/W	NOR Flash Timing for Chip Selects 0 and 1 (NORTF_T01)	07770777h	Page 522
5140001Ah	R/W	NOR Flash Timing for Chip Selects 2 and 3 (NORTF_T23)	07770777h	Page 523
5140001Bh	R/W	NAND Flash Data Timing MSR (NANDF_DATA)	07770777h	Page 523
5140001Ch	R/W	NAND Flash Control Timing (NANDF_CTL)	00000777h	Page 523
5140001Dh	R/W	Flash Reserved (NANDF_RSVD)	00000000h	Page 523

**Table 6-70. Flash Controller Native Registers Summary**

Flash Memory Offset	Flash I/O Offset	Type	Width (Bits)	Register Name	Reset Value	Reference
000h-7FFh	00h-03h	R/W	8	NAND Device Data (NAND_DATA)	Undefined	Page 526
Any Even Address between 800h-80Eh	04h	R/W	8	NAND Control Register (NAND_CTL)	01h	Page 526
Any Odd Address between 801h-80Fh	05h	R/W	8	NAND I/O (NAND_IO)	00h	Page 527
810h	06h	R/W	8	NAND Status (NAND_STS)	0xh	Page 527
815h	08h	R/W	8	NAND ECC Control (NAND_ECC_CTL)	04h	Page 528
811h	09h	R/W	8	NAND ECC LSB Line Parity (NAND_ECC_LSB)	FFh	Page 529
812h	0Ah	R/W	8	NAND ECC MSB Line Parity (NAND_ECC_MSB)	FFh	Page 529
813h	0Bh	R/W	8	NAND ECC Column Parity (NAND_ECC_COL)	FFh	Page 529
814h	0Ch	R/W	8	NAND Line Address Counter (NAND_LAC)	00h	Page 530
816h-FFFh	07h, 0Dh-0Fh	---	---	Reserved. Reads return 0. Writes have no effect.	---	---

**6.19.1 Flash Controller Specific MSRs**

**6.19.1.1 NOR Flash Control (NORF\_CTL)**

MSR Address 51400018h  
 Type R/W  
 Reset Value 00000000h

**NORF\_CTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																							CHK_IOCHRDY3	CHK_IOCHRDY2	CHK_IOCHRDY1	CHK_IOCHRDY0	WE_CS3	WE_CS2	WE_CS1	WE_CS0	



## NOR\_CTL Bit Descriptions

Bit	Name	Description
31:8	RSVD	<b>Reserved</b>
7	CHK_IOCHRDY3 (Note 1)	<b>Check I/O Channel Ready 3.</b> Check FLASH_IOCHRDY signal for NOR Chip Select #3 (FLASH_CS3#) 0: Ignore IOCHRDY signal. No wait states will be inserted. 1: Check IOCHRDY before finishing the chip select strobe pulse.
6	CHK_IOCHRDY2 (Note 1)	<b>Check I/O Channel Ready 2.</b> Check FLASH_IOCHRDY signal for NOR Chip Select #2 (FLASH_CS2#) 0: Ignore IOCHRDY signal. No wait states will be inserted. 1: Check IOCHRDY before finishing the chip select strobe pulse.
5	CHK_IOCHRDY1 (Note 1)	<b>Check I/O Channel Ready 1.</b> Check FLASH_IOCHRDY signal for NOR Chip Select #1 (FLASH_CS1#) 0: Ignore IOCHRDY signal. No wait states will be inserted. 1: Check IOCHRDY before finishing the chip select strobe pulse.
4	CHK_IOCHRDY0 (Note 1)	<b>Check I/O Channel Ready 0.</b> Check FLASH_IOCHRDY signal for NOR Chip Select #0 (FLASH_CS0#) 0: Ignore IOCHRDY signal. No wait states will be inserted. 1: Check IOCHRDY before finishing the chip select strobe pulse.
3	WE_CS3	<b>Write Enable for CS3#.</b> Write Enable for NOR Chip Select #3 (FLASH_CS3#) 0: No write cycles go out to NOR Flash interface via CS3#. 1: Allow write cycles to go out to NOR Flash interface.
2	WE_CS2	<b>Write Enable for CS2#.</b> Write Enable for NOR Chip Select #2 (FLASH_CS2#) 0: No write cycles go out to NOR Flash interface via CS2#. 1: Allow write cycles to go out to NOR Flash interface.
1	WE_CS1	<b>Write Enable for CS1#.</b> Write Enable for NOR Chip Select #1 (FLASH_CS1#) 0: No write cycles go out to NOR Flash interface via CS1#. 1: Allow write cycles to go out to NOR Flash interface.
0	WE_CS0	<b>Write Enable for CS0#.</b> Write Enable for NOR Chip Select #0 (FLASH_CS0#) 0: No write cycles go out to NOR Flash interface via CS0#. 1: Allow write cycles to go out to NOR Flash interface.

Note 1. If any CHK\_IOCHRDY[x] bit (bits [7:4]) is high, and the corresponding Chip Select (FLASH\_CS[x]#) is low, then signal FLASH\_IOCHRDY is checked to determine when to de-assert the RE# or WE# strobe. The RE# or WE# strobe pulse width will be the programmed value (MSR\_NORTF\_T01[22:20], MSR\_NORTF\_T01[6:4], MSR\_NORTF\_T23[22:20] and MSR\_NORTF\_T23[6:4]) increased by at least two local bus clock (33 MHz) cycles. If no CHK\_IOCHRDY[x] bit (bits [7:4]) is high, or if no CHK\_IORDY[x] bit is high that has a corresponding active (low) Chip Select (FLASH\_CS[x]#), then signal FLASH\_IOCHRDY is ignored and the NOR Controller's WE# and RE# strobe pulse widths will be the values programmed in the NOR MSR registers (MSR\_NORTF\_T01[22:20], MSR\_NORTF\_T01[6:4], MSR\_NORTF\_T23[22:20] and MSR\_NORTF\_T23[6:4]). In this case, if the pulse width of WE# and RE# in the NOR MSR registers is programmed as 0, then the NOR Controller's WE# and RE# generation will use 16 as the count value of NOR pulse width.

### 6.19.1.2 NOR Flash Timing MSRs

The NOR Flash controller is used for NOR Flash or GPCS. The timing is different from device to device, so separate timing registers are used for each device.

#### NOR Flash Timing for Chip Selects 0 and 1 (NORTF\_T01)

MSR Address 51400019h

Type R/W

Reset Value 07770777h

#### NORTF\_T01 Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				tH1		RSVD	tP1		RSVD	tS1			RSVD			tH0		RSVD	tP0		RSVD	tS0									

#### NORTF\_T01 Bit Descriptions

Bit	Name	Description
31:27	RSVD	<b>Reserved.</b> Reads return value written.
26:24	Th1 (Note 1)	<b>Hold Time for NOR Chip Select 1.</b> Hold from WE# or RE# rising edge to chip select. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
23	RSVD	<b>Reserved.</b> Reads return value written
22:20	Tp1	<b>Strobe Pulse Width for NOR Chip Select 1.</b> RE# and WE# strobe pulse width. At the end of the Tp, sample the IOCHRDY pin to see if a wait state is needed. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
19	RSVD	<b>Reserved.</b> Reads return value written.
18:16	Ts1 (Note 1)	<b>Setup Time for NOR Chip Select 1.</b> Chip select to WE# or RE# falling edge setup time. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
15:11	RSVD	<b>Reserved.</b> Reads return value written.
10:8	Th0 (Note 1)	<b>Hold time for NOR Chip Select 0.</b> Hold from WE# or RE# rising edge to chip select. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
7	RSVD	<b>Reserved.</b> Reads return value written.
6:4	Tp0	<b>Strobe Pulse Width for NOR Chip Select 0.</b> RE# and WE# strobe pulse width. At the end of the Tp, sample the IOCHRDY pin to see if a wait state is needed. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
3	RSVD	<b>Reserved.</b> Reads return value written.
2:0	Ts0 (Note 1)	<b>Setup Time for NOR Chip Select 0.</b> Chip select to WE# or RE# falling edge setup time. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.

Note 1. The valid range for the count values of setup time, and hold time in NOR MSR registers is 1 through 7 (Local Bus clock cycles or LPC clock cycles) when a General Purpose device is used (with NOR Controller). If signal FLASH\_IOCHRDY is not used by the General Purpose device, then the valid range for the count value of pulse width in NOR MSR registers is 1 through 7 (Local Bus clock cycles or LPC clock cycles). If FLASH\_IOCHRDY is used by the General Purpose device, then the valid range for the count value of pulse width in NOR MSR registers is 2 through 7 (Local Bus clock cycles or LPC clock cycles). In the case of NOR devices, as NOR Controller doesn't support the use of FLASH\_IOCHRDY (explained in Section 5.18.2 "NOR Flash Controller/General Purpose Chip Select" on page 186), the valid range for the count values of setup time, pulse width, and hold time in NOR MSR registers is 1 through 7 (Local Bus clock cycles or LPC clock cycles). In the case of NOR Controller's WE# and RE# strobe pulse widths, if NOR Control MSR register bits[7:4] (CHKRDY[3:0]) are enabled (active high) in the case of General Purpose devices then the generated pulse widths will be longer than programmed count value. If setup or hold time in NOR MSR registers is programmed as 0, then the NOR Controller's WE# and RE# generation will use 16 as the count value of NOR setup or hold time.

**NOR Flash Timing for Chip Selects 2 and 3 (NORTF\_T23)**

MSR Address 5140001Ah  
 Type R/W  
 Reset Value 07770777h

**NORTF\_T23 Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				tH3			RSVD	tP3			RSVD	tS3			RSVD				tH2		RSVD	tP2		RSVD	tS2						

**NORTF\_T23 Bit Descriptions**

Bit	Name	Description
31:27	RSVD	<b>Reserved.</b> Reads return value written.
26:24	tH3 (Note 1)	<b>Hold Time for NOR Chip Select 3.</b> Hold from WE# or RE# rising edge to chip select. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
23	RSVD	<b>Reserved.</b> Reads return value written.
22:20	tP3	<b>Strobe Pulse Width for NOR Chip Select 3.</b> RE# and WE# strobe pulse width. At the end of the tP, sample the FLASH_IOCHRDY signal to see if a wait state is needed. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
19	RSVD	<b>Reserved.</b> Reads return value written.
18:16	tS3 (Note 1)	<b>Setup Time for NOR Chip Select 3.</b> Chip select to WE# or RE# falling edge setup time. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
15:11	RSVD	<b>Reserved.</b> Reads return value written.
10:8	tH2 (Note 1)	<b>Hold Time for NOR Chip Select 2.</b> Hold from WE# or RE# rising edge to chip select. Refer to Figure 5-58.
7	RSVD	<b>Reserved.</b> Reads return value written.
6:4	tP2	<b>Strobe Pulse Width for NOR Chip Select 2.</b> RE# and WE# strobe pulse width. At the end of the tP, sample the FLASH_IOCHRDY signal to see if a wait state is needed. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.
3	RSVD	<b>Reserved.</b> Reads return value written.
2:0	tS2 (Note 1)	<b>Setup Time for NOR Chip Select 2.</b> Chip select to WE# or RE# falling edge setup time. Refer to Figure 5-58 "NOR Flash with Wait States Timing" on page 188.

Note 1. The valid range for the count values of setup time, and hold time in NOR MSR registers is 1 through 7 when a General Purpose device is used (with NOR Controller). If signal FLASH\_IOCHRDY is not used by the General Purpose device, then the valid range for the count value of pulse width in NOR MSR registers is 1 through 7. If FLASH\_IOCHRDY is used by the General Purpose device, then the valid range for the count value of pulse width in NOR MSR registers is 2 through 7. In the case of NOR devices, as NOR Controller doesn't support the use of FLASH\_IOCHRDY (explained in Section 5.18.2 "NOR Flash Controller/General Purpose Chip Select" on page 186), the valid range for the count values of setup time, pulse width, and hold time in NOR MSR registers is 1 through 7. In the case of NOR Controller's WE# and RE# strobe pulse widths, if NOR Control MSR register bits[7:4] (CHKRDY[3:0]) are enabled (active high) in the case of General Purpose devices then the generated pulse widths will be longer than programmed count value. If setup or hold time in NOR MSR registers is programmed as 0, then the NOR Controller's WE# and RE# generation will use 16 as the count value of NOR setup or hold time.

### 6.19.1.3 NAND Flash Data Timing MSR (NANDF\_DATA)

MSR Address 5140001Bh  
 Type R/W  
 Reset Value 07770777h

Most NAND devices have similar timing. All NAND devices share the timing registers. The valid range for the count values of setup time and hold time in NAND MSRs is 0 through 7 Local Bus clock ('lb\_c') cycles or LPC clock ('lpc\_c') cycles and the valid range for the count values of pulse width in NAND MSRs is 1 through 7 Local Bus clock ('lb\_c') cycles or LPC clock ('lpc\_c') cycles.

#### NANDF\_DATA Register Map

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				tRH		RSVD	tRP		RSVD	tRS		RSVD				tWH		RSVD	tWP		RSVD	tWS									

#### NANDF\_DATA Bit Descriptions

Bit	Name	Description
31:27	RSVD	<b>Reserved.</b> Reads return value written.
26:24	tRH	<b>Data Read Hold Time.</b> This timing is just for internal state machine; no external reference point. Can be set to 0 if the hold time is not needed. Range = 0h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.
23	RSVD	<b>Reserved.</b> Reads return value written.
22:20	tRP	<b>Data Read Pulse Width.</b> The RE# active pulse width in data read phase. Range = 1h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.
19	RSVD	<b>Reserved.</b> Reads return value written.
18:16	tRS	<b>Data Read Setup Time.</b> This timing is just for internal state machine; no external reference point. Can be set to 0 if the setup time is not needed. Range = 0h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.
15:11	RSVD	<b>Reserved.</b> Reads return value written.
10:8	tWH	<b>Data Write Hold Time.</b> The hold time from WE# rising edge to I/O bus is turned off. Range = 0h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.
7	RSVD	<b>Reserved.</b> Reads return value written.
6:4	tWP	<b>Data Write Pulse Width.</b> The WE# active pulse width in data write phase. Note that the data byte is put on the I/O bus at the same time the WE# is asserted. Range = 1h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.
3	RSVD	<b>Reserved.</b> Reads return value written.
2:0	tWS	<b>Data Write Setup Time.</b> This timing is just for internal state machine; no external reference point. Can be set to 0 if the setup time is not needed. Range = 0h to 7h. Refer to Figure 5-61 "NAND Data Timing with Wait States" on page 190.

**6.19.1.4 NAND Flash Control Timing (NANDF\_CTL)**

MSR Address 5140001Ch  
 Type R/W  
 Reset Value 00000777h

**NANDF\_CTL Register Map**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																						tCH		RSVD	tCP		RSVD	tCS			

**NANDF\_CTL Bit Descriptions**

Bit	Name	Description
31:11	RSVD	<b>Reserved.</b> Reads return value written.
10:8	tCH	<b>Control Hold Time.</b> The hold time from the rising edge of WE# to the toggle of control signals. Note that the I/O bus is turned off when the tCH expires. Range = 0h to 7h. Refer to Figure 5-59 "NAND Flash Command/Address Timing" on page 189.
7	RSVD	<b>Reserved.</b> Reads return value written.
6:4	tCP	<b>Control Pulse Width.</b> The WE# active pulse width in command/address phase. Note that the command/address byte is put on the I/O bus at the same time that the WE# is asserted. Range = 1h to 7h. Refer to Figure 5-59 "NAND Flash Command/Address Timing" on page 189.
3	RSVD	<b>Reserved.</b> Reads return value written.
2:0	tCS	<b>Control Setup Time.</b> The setup time from the toggle of the control signals to the WE# falling edge. Range = 0h to 7h. Refer to Figure 5-59 "NAND Flash Command/Address Timing" on page 189.

**6.19.1.5 Flash Reserved (NANDF\_RSVD)**

MSR Address 5140001Dh  
 Type R/W  
 Reset Value 00000000h

This register is reserved. Reads return 0. Writes have no effect.

## 6.19.2 Flash Controller Native Registers

### 6.19.2.1 NAND Device Data (NAND\_DATA)

Flash Memory Offset 000h-7FFh  
 Flash I/O Offset 00h-03h  
 Type R/W  
 Reset Value Undefined

Reading or writing to this range accesses the same NAND Device Data.

**NAND\_DATA Register Map**

7	6	5	4	3	2	1	0
DATA							

**NAND\_DATA Bit Descriptions**

Bit	Name	Description
7:0	DATA	<b>NAND Device Data.</b> No default value. This address space is not to be read from until data is read from the NAND Flash device. The system will hang if a read is done from this address space without a prior read from NAND Flash device.

### 6.19.2.2 NAND Control Register (NAND\_CTL)

Flash Memory Offset Any Even Address between 800h-80Eh  
 Flash I/O Offset 04h  
 Type R/W  
 Reset Value 01h

**NAND\_CTL Register Map**

7	6	5	4	3	2	1	0
RSVD			DIST_EN	RDY_INT_MASK	ALE	CLE	CE#

**NAND\_CTL Bit Descriptions**

Bit	Name	Description
7:5	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0 when read.
4	DIST_EN	<b>NAND Distract Interrupt Enable.</b> 0: Disables the generation of NAND Distract Interrupt. 1: Enables the generation of NAND Distract Interrupt.
3	RDY_INT_MASK	<b>NAND Ready Interrupt Mask.</b> 0: Interrupt is masked. 1: Enable NAND Flash device's RDY/BUSY# signal to generate an interrupt.
2	ALE	<b>Address Latch Enable.</b> FLASH_ALE output signal reflects the value of this bit.
1	CLE	<b>Command Latch Enable.</b> FLASH_CLE output signal reflects the value of this bit.
0	CE#	<b>Chip Enable.</b> CE# signal reflects the value of this bit. The NAND_CS signals from the Diverse Device determine which CE# is asserted. Keep this bit low during entire NAND cycle. Writing a 1 to this bit resets the NAND controller.

**6.19.2.3 NAND I/O (NAND\_IO)**

Flash Memory Offset Any Odd Address between 801h-80Fh  
Flash I/O Offset 05h  
Type R/W  
Reset Value 00h

**NAND\_IO Register Map**

7	6	5	4	3	2	1	0
IO							

**NAND\_IO Bit Descriptions**

Bit	Name	Description
7:0	IO	<p><b>I/O Register.</b> Writing to this register triggers a command/address phase sub-cycle on the NAND Flash interface. The data written to this register is put on the I/O bus during the sub-cycle. It returns previous written value when read.</p> <p><b>Note:</b> Before writing to this register check for CTLR_BUSY bit (Flash Memory Offset 810h[2]/Flash I/O Offset 06h[2]) in NAND_STS register to be 0.</p>

**6.19.2.4 NAND Status (NAND\_STS)**

Flash Memory Offset 810h  
Flash I/O Offset 06h  
Type R/W  
Reset Value 0xh

**NAND\_STS Register Map**

7	6	5	4	3	2	1	0
RSVD				FLASH_RDY	CTLR_BUSY	CMD_COMP	DIST_ST

**NAND\_STS Bit Descriptions**

Bit	Name	Description
7:4	RSVD (RO)	<b>Reserved (Read Only).</b> Returns 0 when read.
3	FLASH_RDY (RO)	<b>Flash Ready (Read Only).</b> Double synchronized output (with respect to local bus clock) of the NAND Flash device's RDY/BUSY#.
2	CTLR_BUSY (RO)	<b>NAND Controller Busy (Read Only).</b> When high, indicates that the NAND Controller's state machines are busy.
1	CMD_COMP	<b>NAND Command Complete.</b> When high, indicates that the most recent NAND command has completed. May be read anytime. Write 1 to clear this bit. Writing 0 has no effect.
0	DIST_ST	<p><b>NAND Distract Status.</b> Occurrence of a NOR interruption during a NAND transaction sets this bit. Write 1 to clear this bit. Writing 0 has no effect.</p> <p>A NAND transaction is started as soon as CE# goes low. It is stopped when CE# goes high. Typically, a NAND transaction needs multiple software commands (from 6 to ~500). Since the Flash Interface is shared between NAND and NOR Flash Controllers and the NOR Flash Controller gets priority to use the Flash Interface, a NAND transaction may be interrupted by a NOR transaction. DIST_ST bit is set to record this event. NAND Flash software must take necessary actions to recover the uncompleted transaction.</p>

**6.19.2.5 NAND ECC Control (NAND\_ECC\_CTL)**

Flash Memory Offset 815h  
 Flash I/O Offset 08h  
 Type R/W  
 Reset Value 04h

**NAND\_ECC\_CTL Register Map**

7	6	5	4	3	2	1	0
RSVD					PARITY	CLRECC	ENECC

**NAND\_ECC\_CTL Bit Descriptions**

Bit	Name	Description
7:3	RSVD	<b>Reserved.</b> Reads return value written.
2	PARITY	<p><b>Parity.</b></p> <p>0: ECC Parity registers are even parity.            1: ECC Parity registers are odd parity.</p> <p>In the case of odd ECC parity, the value read from NAND_ECC_LSB (Flash Memory Offset 811h/Flash I/O Offset 09h), NAND_ECC_MSB (Flash Memory Offset 812h/Flash I/O Offset 0Ah), and NAND_ECC_COL (Flash Memory Offset 813h/Flash I/O Offset 0Bh) parity registers will be complement of the value written into these registers (except for LSB two bits of the NAND_ECC_COL register, they are always 11b for odd parity).</p>
1	CLRECC	<p><b>Clear ECC Engine.</b> Write 1 to clear ECC parity registers (NAND_ECC_LSB, NAND_ECC_MSB, and NAND_ECC_COL), NAND Line Address Counter register (NAND_LAC) and reset the ECC engine. Writing 0 has no effect.</p> <p>The ECC engine contains an 8-bit Line Address Counter (LAC) to keep track of data that has been read from or written into the NAND Flash. Software has to reset the counter by writing a 1 to the CLRECC bit before transferring data to/from the NAND Flash. Every data byte transferred to/from the NAND Flash Controller increments the LAC. The NAND_LAC (Flash Memory Offset 814h/Flash I/O Offset 0Ch) register reports the current count of the LAC.</p>
0	ENECC	<p><b>Enable ECC Calculation Engine.</b></p> <p>0: Disable ECC Engine. ECC engine holds previous value.            1: Enable ECC Engine. Every data byte transferred to/from the NAND Flash Controller will be counted in ECC calculation.</p>



**6.19.2.6 NAND ECC Parity Registers**

ECC parity registers contain 22 parity bits. The bit location and definition follows the SmartMedia Physical Format Specifications.

**NAND ECC LSB Line Parity (NAND\_ECC\_LSB)**

Flash Memory Offset 811h  
Flash I/O Offset 09h  
Type R/W  
Reset Value FFh

**NAND\_ECC\_LSB Register Map**

7	6	5	4	3	2	1	0
LP[7:0]							

**NAND\_ECC\_LSB Bit Descriptions**

Bit	Name	Description
7:0	LP[7:0]	Line Parity Bits 7 through 0.

**NAND ECC MSB Line Parity (NAND\_ECC\_MSB)**

Flash Memory Offset 812h  
Flash I/O Offset 0Ah  
Type R/W  
Reset Value FFh

**NAND\_ECC\_MSB Register Map**

7	6	5	4	3	2	1	0
LP[15:8]							

**NAND\_ECC\_MSB Bit Descriptions**

Bit	Name	Description
7:0	LP[15:8]	Line Parity Bits 15 through 8.

**NAND ECC Column Parity (NAND\_ECC\_COL)**

Flash Memory Offset 813h  
Flash I/O Offset 0Bh  
Type R/W  
Reset Value FFh

**NAND\_ECC\_COL Register Map**

7	6	5	4	3	2	1	0
CP[5:0]						RSVD	

**NAND\_ECC\_COL Bit Descriptions**

Bit	Name	Description
7:2	CP[5:0]	Column Parity Bits 5 through 0.
1:0	RSVD (RO)	Reserved. Always returns 11 for odd ECC parity and 00 for even ECC parity.

**6.19.2.7 NAND Line Address Counter (NAND\_LAC)**

Flash Memory Offset 814h  
Flash I/O Offset 0Ch  
Type R/W  
Reset Value 00h

**NAND\_LAC Register Map**

7	6	5	4	3	2	1	0
LAC							

**NAND\_LAC Bit Descriptions**

Bit	Name	Description
7:0	LAC	<b>Line Address Counter Value.</b> The ECC engine contains an 8-bit Line Address Counter (LAC) to keep track of data that has been read from or written into the NAND Flash. Software has to reset the counter by writing a 1 to the CLRECC bit (Flash Memory Offset 815h[1]/Flash I/O Offset 08h[1]) before transferring data to/from the NAND Flash. Every data byte exchanged between NAND Flash controller and GeodeLink Adapter increments the LAC. The NAND_LAC register reports the current count of the LAC.

## 6.20 GeodeLink™ Control Processor Register Descriptions

The GeodeLink Control Processor's (GLPC) register set consists of:

- Standard GeodeLink Device (GLD) MSRs
- GLCP Specific MSRs

The MSRs (both Standard and GLPC Specific) are accessed via the RDMSR and WRMSR processor instruc-

tions. The MSR address is derived from the perspective of the CPU Core. See Section 4.2 "MSR Addressing" on page 59 for more details.

The tables that follow are register summary tables that include reset values and page references where the bit descriptions are provided.

**Table 6-71. Standard GeodeLink™ Device MSRs Summary**

MSR Address	Type	Register	Reset Value	Reference
51700000h	RO	GLD Capabilities MSR (GLCP_GLD_MSR_CAP)	00000000_002021xxh	Page 532
51700001h	R/W	GLD Master Configuration MSR (GLCP_GLD_MSR_CONFIG)	00000000_00000000h	Page 532
51700002h	R/W	GLD SMI MSR (GLCP_GLD_MSR_SMI)	00000000_00000003h	Page 532
51700003h	R/W	GLD Error MSR (GLCP_GLD_MSR_ERROR)	00000000_00000000h	Page 534
51700004h	R/W	GLD Power Management MSR (GLCP_GLD_MSR_PM)	00000000_00000000h	Page 535
51700005h	R/W	GLD Diagnostic MSR (GLCP_GLD_MSR_DIAG)	00000000_00000000h	Page 535

**Table 6-72. GLPC Specific MSRs Summary**

MSR Address	Type	Register	Reset Value	Reference
51700008h	R/W	GLCP Clock Disable Delay Value (GLCP_CLK_DIS_DELAY)	00000000_00000000h	Page 537
51700009h	R/W	GLCP Clock Mask for Sleep Request (GLCP_PMCLKDISABLE)	00000000_00000000h	Page 537
5170000Ah	RO	GLCP Fabrication (GLCP_FAB)	00000000_00000002h	Page 538
5170000Bh	R/W	GLCP Global Power Management Control (GLCP_GLB_PM)	00000000_00000000h	Page 538
5170000Ch	R/W	GLCP Debug Output from Chip (GLCP_DBGOUT)	00000000_00000000h	Page 539
5170000Dh	R/W	Reserved Registers (GLPC_RSVD)	00000000_00000000h	---
5170000Eh	R/W	Software Communication Register (GLCP_DOWSER)	00000000_00000000h	Page 539
5170000Fh	R/W	GLCP Reserved Register (GLPC_RSVD)	00000000_00000000h	---
51700010h	R/W	GLCP Clock Control (GLCP_CLKOFF)	00000000_00000000h	Page 540
51700011h	RO	GLCP Clock Active (GLCP_CLKACTIVE)	0000000x_xxxxxxxxh	Page 540
51700012h	R/W	GLCP Clock Mask for Debug Clock Stop Action (GLCP_CLKDISABLE)	00000000_00000000h	Page 540
51700013h	R/W	GLCP Clock Active Mask for Suspend Acknowledge (GLCP_CLK4ACK)	00000000_00000000h	Page 541
51700014h	R/W	GLCP System Reset Control (GLCP_SYS_RST)	00000000_00000000h	Page 542
51700015h	R/W	Reserved Registers (GLPC_RSVD)	00000000_00000000h	---
51700016h	R/W	GLCP Debug Clock Control (GLCP_DBGCLKCTL)	00000000_00000002h	Page 542
51700017h	RO	Chip Revision ID (GLCP_CHIP_REV_ID)	00000000_000000xxh	Page 543
51700018h-517000FFh	R/W	Reserved Registers (GLPC_RSVD) - Reserved for internal testing. Do not write to these registers.	xxxxxxxx_xxxxxxxxh	---

## 6.20.1 Standard GeodeLink™ Device (GLD) MSRs

### 6.20.1.1 GLD Capabilities MSR (GLCP\_GLD\_MSR\_CAP)

MSR Address 51700000h  
 Type RO  
 Reset Value 00000000\_002021xxh

GLCP\_GLD\_MSR\_CAP Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DEVID																REVID							

GLCP\_GLD\_MSR\_CAP Bit Descriptions

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:8	DEV_ID	<b>Device ID.</b> Identifies module (2021h).
7:0	REV_ID	<b>Revision ID.</b> Identifies module revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.

### 6.20.1.2 GLD Master Configuration MSR (GLCP\_GLD\_MSR\_CONFIG)

MSR Address 51700001h  
 Type R/W  
 Reset Value 00000000\_00000000h

GLCP\_GLD\_MSR\_CONFIG Register Map

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																												PID			

GLCP\_GLD\_MSR\_CONFIG Bit Descriptions

Bit	Name	Description
63:3	RSVD	<b>Reserved.</b> Always write 0.
2:0	PID	<b>Priority ID.</b> Always write 0.

### 6.20.1.3 GLD SMI MSR (GLCP\_GLD\_MSR\_SMI)

MSR Address 51700002h  
 Type R/W  
 Reset Value 00000000\_00000003h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 0. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.3 "MSR Address 2: SMI Control" on page 74 for further SMI/ASMI generation details.)

**GLCP\_GLD\_MSR\_SMI Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														DBG_ASMI_FLAG	ERR_ASMI_FLAG	RSVD														DBG_ASMI_EN	ERR_ASMI_EN

**GLCP\_GLD\_MSR\_SMI Bit Descriptions**

Bit	Name	Description
63:18	RSVD	<b>Reserved.</b> Reads as 0.
17	DBG_ASMI_FLAG	<b>Debug ASMI Flag.</b> If high, records that an ASMI was generated and applied to the system, due to a debug event. Write 1 to clear; writing 0 has no effect. DBG_ASMI_EN (bit 1) must be low to enable this flag.
16	ERR_ASMI_FLAG	<b>Error ASMI Flag.</b> If high, records that an ASMI was generated and applied to the system due to ERR signal. Write 1 to clear; writing 0 has no effect. ERR_ASMI_EN (bit 0) must be low to enable this flag.
15:2	RSVD	<b>Reserved.</b> Reads as 0.
1	DBG_ASMI_EN	<b>Debug ASMI Enable.</b> Write 0 to enable DBG_ASMI_FLAG (bit 17). Write 1 to disable the flag and ASMI generation.
0	ERR_ASMI_EN	<b>Error ASMI Enable.</b> Write 0 to enable ERR_ASMI_FLAG (bit 16). Write 1 to disable the flag and ASMI generation.

**6.20.1.4 GLD Error MSR (GLCP\_GLD\_MSR\_ERROR)**

MSR Address 51700003h  
 Type R/W  
 Reset Value 00000000\_00000000h

The flags are set by internal conditions. The internal conditions are enabled if the EN bit is 0. Reading the FLAG bit returns the value; writing 1 clears the flag; writing 0 has no effect. (See Section 4.8.4 "MSR Address 3: Error Control" on page 78 for further ERR generation details.)

**GLCP\_GLD\_MSR\_ERROR Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32			
RSVD																																	SIZE_ERR_FLAG	UNEXP_TYPE_ERR_FLAG
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RSVD																																	SIZE_ERR_EN	UNEXP_TYPE_ERR_EN

**GLCP\_GLD\_MSR\_ERROR Bit Descriptions**

Bit	Name	Description
63:34	RSVD	<b>Reserved.</b> Reads as 0.
33	SIZE_ERR_FLAG	<b>Size Error Flag.</b> The GLIU interface detected a read or write of more than 1 data packet (size = 16 or 32 bytes). If a response packet is expected, the EXCEP bit of the response packet will be set; in all cases the asynchronous error signal will be set. Write 1 to clear; writing 0 has no effect.
32	UNEXP_TYPE_ERR_FLAG	<b>Unexpected Type Error Flag.</b> An unexpected type was sent to the GLCP GeodeLink interface (start request with BEX type, snoop, peek_write, debug_req, or NULL type). If a response packet is expected, the EXCEP bit of the response packet will be set; in all cases the asynchronous error signal will be set. Write 1 to clear; writing 0 has no effect.
31:2	RSVD	<b>Reserved.</b> Reads as 0.
1	SIZE_ERR_EN	<b>Size Error Enable.</b> Write 0 to enable the flag (bit 33) and allow the size error event to generate an asynchronous error to the system.
0	UNEXP_TYPE_ERR_EN	<b>Unexpected Type Error Enable.</b> Write 0 to enable the flag (bit 32) and allow the unexpected type event to generate an asynchronous error to the system.

**6.20.1.5 GLD Power Management MSR (GLCP\_GLD\_MSR\_PM)**

MSR Address 51700004h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLCP\_GLD\_MSR\_PM Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32		
RSVD																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RSVD																														PMODE1		PMODE0	

**GLCP\_GLD\_MSR\_PM Bit Descriptions**

Bit	Name	Description
63:4	RSVD	<b>Reserved.</b> Reads as 0.
3:2	PMODE1	<b>Power Mode 1.</b> Power mode for Clock Domain 1 (Debug). 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.
1:0	PMODE0	<b>Power Mode 0.</b> Power mode for Clock Domain 0 (GLIU). 00: Disable clock gating. Clocks are always on. 01: Enable active hardware clock gating. Clock goes off whenever this module's circuits are not busy. 10: Reserved. 11: Reserved.

**6.20.1.6 GLD Diagnostic MSR (GLCP\_GLD\_MSR\_DIAG)**

MSR Address 51700005h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is reserved for internal use by AMD and should not be written to.

## 6.20.2 GLCP Specific MSRs

These registers are used for power management, and facilitate some clock and reset functions. The “CLK” associated registers (i.e., CLKACTIVE, CLKOFF, CLKDISABLE, CLK4ACK and PMCLKDISABLE) have the same layout where each bit is associated with a clock domain. The layout and recommended operating values for the “CLK” associated registers is shown in Table 6-73. For additional discussion on clock management considerations, see Section 4.5 “Clock Considerations” on page 63.

**Table 6-73. Clock Mapping / Operational Settings**

MSR Bit	Name/Description	GLCP Register				
		CLK ACTIVE	CLK OFF	CLK DISABLE	CLK 4ACK	PM CLK DISABLE
CLK[63:34]	RSVD. Reserved for future use by AMD.	RO	0	0	0	0
CLK33	GLCP_PCI. GLCP PCI Clock.	RO	0	0	0	0
CLK32	GLCP_DBG. GLCP DBG Logic Clock.	RO	0	0	0	0
CLK31	GLCP_GLIU. GLCP GeodeLink Clock.	RO	0	0	0	0
CLK30	DIVIL_MFGPT_32K_STD. MFGPT 32 kHz Standby Clock entering DIVIL.	RO	0	0	0	0
CLK29	DIVIL_MFGPT_14M. MFGPT 14 MHz Clock entering DIVIL.	RO	0	0	0	0
CLK28	DIVIL_MFGPT_32K. MFGPT 32 kHz Clock entering DIVIL.	RO	0	0	0	0
CLK27	DIVIL_GPIO_STD. GPIO Standby Clock entering DIVIL.	RO	0	0	0	0
CLK26	DIVIL_GPIO. GPIO Clock entering DIVIL.	RO	0	0	0	0
CLK25	DIVIL_PMC_STD. PMC Standby Clock.	RO	0	0	0	0
CLK24	DIVIL_PMC. PMC Working Logic Clock.	RO	0	0	0	0
CLK23	DIVIL_UART2. UART2 Clock entering DIVIL.	RO	0	0	1	0
CLK22	DIVIL_UART1. UART1 Clock entering DIVIL.	RO	0	0	1	0
CLK21	DIVIL_PIT. PIT Clock entering DIVIL.	RO	0	0	0	0
CLK20	DIVIL_SMB. SMB Clock entering DIVIL.	RO	0	0	1	0
CLK19	DIVIL_DMA. DMA Clock entering DIVIL.	RO	0	0	1	0
CLK18	DIVIL_LPC. LPC Clock entering DIVIL.	RO	0	0	1	0
CLK17	DIVIL_LB. LBus Clock entering DIVIL.	RO	0	0	1	0
CLK16	DIVIL_GLIU. GeodeLink (GLIU) Clock entering DIVIL.	RO	0	0	1	0
CLK15	ACC_BIT. AC97 Clock entering ACC.	RO	0	0	0	0
CLK14	ACC_LB. 33 MHz Clock entering ACC.	RO	0	0	1	0
CLK13	ACC_GLIU. GeodeLink (GLIU) Clock entering ACC.	RO	0	0	1	0
CLK12	ATAC_LB. 66 MHz Clock entering ATAC.	RO	0	0	1	0
CLK11	ATAC_GLIU. GeodeLink (GLIU) Clock entering ATAC.	RO	0	0	1	0
CLK10	USB2_48M. 48 MHz Clock entering USB (ports 3 & 4).	RO	0	0	1	0
CLK9	USB1_48M. 48 MHz Clock entering USB (ports 1 & 2).	RO	0	0	1	0
CLK8	USB2_LB. 33 MHz Clock entering USB (ports 3 & 4).	RO	0	0	1	0
CLK7	USB1_LB. 33 MHz Clock entering USB (ports 1 & 2).	RO	0	0	1	0
CLK6	USB2_GLIU. GeodeLink (GLIU) Clock entering USB (ports 3 & 4).	RO	0	0	1	0
CLK5	USB1_GLIU. GeodeLink (GLIU) Clock entering USB (ports 1 & 2).	RO	0	0	1	0
CLK4	GLPCI_PCIF. Fast PCI Clock for chip I/O interface.	RO	0	0	0	0
CLK3	GLPCI_PCI. Normal PCI Clock for GLPCI_SB logic.	RO	0	0	1	0
CLK2	GLPCI_GLIU. GeodeLink (GLIU) Clock entering GLPCI_SB.	RO	0	0	1	0
CLK1	GL0_1. GeodeLink (GLIU) operational logic clock.	RO	0	0	1	0
CLK0	GL0_0. GeodeLink (GLIU) clock to timer logic.	RO	0	0	0	0



**6.20.2.1 GLCP Clock Disable Delay Value (GLCP\_CLK\_DIS\_DELAY)**

MSR Address 51700008h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register has bits that, when set, disable clocks.

**GLCP\_CLK\_DIS\_DELAY Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CLK_DELAY																							

**GLCP\_CLK\_DIS\_DELAY Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0
23:0	CLK_DELAY	<b>Clock Disable Delay.</b> If enabled in GLCP_GLB_PM (MSR 5170000Bh[1] = 1), this field indicates the period to wait from the assertion of SUSPA# before gating-off clocks specified in GLCP_PMCLKDISABLE (MSR 51700009h). If this delay is enabled, it overrides or disables the function of GLCP_CLK4ACK (MSR 51700013h). If GLCP_GLB_PM enable bit is not set (MSR 5170000Bh[1] = 0), but this register (GLCP_CLK_DIS_DELAY) is non-zero, then this register behaves as a timeout for the CLK4ACK behavior. Note that this number is in terms of PCI clock cycles, divided by 16.

**6.20.2.2 GLCP Clock Mask for Sleep Request (GLCP\_PMCLKDISABLE)**

MSR Address 51700009h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLCP\_PMCLKDISABLE Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK31	CLK30	CLK29	CLK28	CLK27	CLK26	CLK25	CLK24	CLK23	CLK22	CLK21	CLK20	CLK19	CLK18	CLK17	CLK16	CLK15	CLK14	CLK13	CLK12	CLK11	CLK10	CLK9	CLK8	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0

**GLCP\_PMCLKDISABLE Bit Descriptions**

Bit	Name	Description
63:34	RSVD	<b>Reserved</b>
33:0	CLK_DIS	<b>Clock Disable.</b> The bits in this field correspond to the Clock Off (CLK_OFF) bits in GLCP_CLKOFF (MSR 51700010h). If a bit in this field is set, then the corresponding CLK_OFF bit will be set when the power management circuitry disables clocks when entering Sleep. For bit-to-clock correspondences and recommended operational settings see Table 6-73 on page 536.

**6.20.2.3 GLCP Fabrication (GLCP\_FAB)**

MSR Address 5170000Ah  
 Type RO  
 Reset Value 00000000\_00000002h

This is a read only register used to track various fab processes and product family parameters. It is meant for AMD internal use only.

**GLCP\_FAB Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLCP\_FAB Bit Descriptions**

Bit	Name	Description
63:0	RSVD	<b>Reserved.</b> Reads return reset value.

**6.20.2.4 GLCP Global Power Management Control (GLCP\_GLB\_PM)**

MSR Address 5170000Bh  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLCP\_GLB\_PM Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																														CLK_DLY_EN	RSVD

**GLCP\_GLB\_PM Bit Descriptions**

Bit	Name	Description
63:2	RSVD	<b>Reserved</b>
1	CLK_DLY_EN	<b>Clock Delay Enable.</b> Write 1 to enable gating-off clock enables from a delay rather than from GLCP_CLK4ACK (MSR 51700013h).
0	RSVD	<b>Reserved.</b> Must be written 0.

**6.20.2.5 GLCP Debug Output from Chip (GLCP\_DBGOUT)**

MSR Address 5170000Ch  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLCP\_DBGOUT Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLCP\_DBGOUT Bit Descriptions**

Bit	Name	Description
63:0	RSVD	<b>Reserved.</b> These bits are reserved for internal testing only. These bits should not be written to.

**6.20.2.6 Software Communication Register (GLCP\_DOWSER)**

MSR Address 5170000Eh  
 Type R/W  
 Reset Value 00000000\_00000000h

This register is a free 64-bit read/write register that can be used by software, for example, to store flags.

**GLCP\_DOWSER Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
VAL																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															

**GLCP\_DOWSER Bit Descriptions**

Bit	Name	Description
63:0	VAL	<b>Value.</b> This 64-bit scratchpad register was specifically added for SW debugger use (DOWSER). The register resets to 00000000_00000000h with both hard and soft resets.

**6.20.2.7 GLCP Clock Control (GLCP\_CLKOFF)**

MSR Address 51700010h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register has bits that, when set, disable clocks immediately. It is not intended for normal use, only as a debug tool.

**GLCP\_CLKOFF Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32			
RSVD																																	CLK33	CLK32
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
CLK31	CLK30	CLK29	CLK28	CLK27	CLK26	CLK25	CLK24	CLK23	CLK22	CLK21	CLK20	CLK19	CLK18	CLK17	CLK16	CLK15	CLK14	CLK13	CLK12	CLK11	CLK10	CLK9	CLK8	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0			

**GLCP\_CLKOFF Bit Descriptions**

Bit	Name	Description
63:34	RSVD	<b>Reserved</b>
33:0	CLK_OFF	<b>Clock Off.</b> A 1 in any bit position causes the corresponding clock to be immediately and unconditionally shut off. It is not intended for normal operational use, only as a debug tool. For bit-to-clock correspondences and recommended operational settings see Table 6-73 on page 536.

**6.20.2.8 GLCP Clock Active (GLCP\_CLKACTIVE)**

MSR Address 51700011h  
 Type RO  
 Reset Value 0000000x\_xxxxxxxxh

**GLCP\_CLKACTIVE Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32			
RSVD																																	CLK33	CLK32
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
CLK31	CLK30	CLK29	CLK28	CLK27	CLK26	CLK25	CLK24	CLK23	CLK22	CLK21	CLK20	CLK19	CLK18	CLK17	CLK16	CLK15	CLK14	CLK13	CLK12	CLK11	CLK10	CLK9	CLK8	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0			

**GLCP\_CLKACTIVE Bit Descriptions**

Bit	Name	Description
63:34	RSVD (RO)	<b>Reserved (Read Only).</b> Reads as 0.
33:0	CLK_ACT (RO)	<b>Clock Active (Read Only).</b> This register reports the status, active or inactive, of each clock. When set, each bit indicates that a block is internally enabling its own clock. The actual clock can be off even though the CLK_ACT bit is set, if the corresponding CLK_OFF bit is set in GLCP_CLKOFF (MSR 51700010h). For bit-to-clock correspondences and recommended operational settings see Table 6-73 on page 536.

**6.20.2.9 GLCP Clock Mask for Debug Clock Stop Action (GLCP\_CLKDISABLE)**

MSR Address 51700012h  
 Type R/W  
 Reset Value 00000000\_00000000h

**GLCP\_CLKDISABLE Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLCP\_CLKDISABLE Bit Descriptions**

Bit	Name	Description
63:0	RSVD	<b>Reserved.</b> This register is reserved for internal testing only. These bits should not be written to.

**6.20.2.10 GLCP Clock Active Mask for Suspend Acknowledge (GLCP\_CLK4ACK)**

MSR Address 51700013h  
 Type R/W  
 Reset Value 00000000\_00000000h

This register has bits that correspond to the Clock Active (CLK\_ACT) bits in GLCP\_CLKACTIVE (MSR 51700011h). If the bit in GLCP\_CLK4ACK is set, then the SUSPA# signal will not go low unless all the marked clocks are inactive.

**GLCP\_CLK4ACK Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK31	CLK30	CLK29	CLK28	CLK27	CLK26	CLK25	CLK24	CLK23	CLK22	CLK21	CLK20	CLK19	CLK18	CLK17	CLK16	CLK15	CLK14	CLK13	CLK12	CLK11	CLK10	CLK9	CLK8	CLK7	CLK6	CLK5	CLK4	CLK3	CLK2	CLK1	CLK0

**GLCP\_CLK4ACK Bit Descriptions**

Bit	Name	Description
63:34	RSVD	<b>Reserved</b>
33:0	CLKACT_EN_SLP	<b>Clock Active Enable for Sleep.</b> A 1 in any bit position indicates the corresponding clock is to be monitored during a power management Sleep operation. When all the clocks with associated 1s become inactive, the GLCP sends a Suspend Acknowledge (SUSPA#) to the power management logic to begin the transition to the Sleep state. Use of this register during Sleep sequences requires the CLK_DLY_EN bit (MSR 5170000Bh[1]) to be 0. For bit-to-clock correspondences and recommended operational settings see Table 6-73 on page 536.

**6.20.2.11 GLCP System Reset Control (GLCP\_SYS\_RST)**

MSR Address 51700014h  
 Type R/W  
 Reset Value 00000000\_00000000h

Writing 1 to the CHIP\_RESET bit creates a chip-wide reset and in turn, resets this register. Writing this register with the CHIP\_RESET bit set will never send a write-response over the GLIU Interface (this allows halting bus traffic before the reset occurs).

**GLCP\_SYS\_RST Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	
RSVD																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RSVD																RSVD																CHIP_RESET

**GLCP\_SYS\_RST Bit Descriptions**

Bit	Name	Description
63:24	RSVD	<b>Reserved.</b> Reads as 0.
23:1	RSVD	<b>Reserved.</b> These bits can be read/written but should not be used; write to 0.
0	CHIP_RESET	<b>Chip Reset.</b> When written to a 1, the Geode CS5535 companion device enters reset, which in turn resets this register. JTAG logic is not reset by CHIP_RESET, but otherwise the entire chip is reset. (Default = 0.)

**6.20.2.12 GLCP Debug Clock Control (GLCP\_DBGCLKCTL)**

MSR Address 51700016h  
 Type R/W  
 Reset Value 00000000\_00000002h

This register is reserved for internal testing only. These bits should not be written to.

**GLCP\_DBGCLKCTL Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																															

**GLCP\_DBGCLKCTL Bit Descriptions**

Bit	Name	Description
63:0	RSVD	<b>Reserved.</b> This register is reserved for internal testing only. These bits should not be written to.

**6.20.2.13 Chip Revision ID (GLCP\_CHIP\_REV\_ID)**

MSR Address 51700017h  
 Type RO  
 Reset Value 00000000\_000000xxh

**CHIP\_REV\_ID Register Map**

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
RSVD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								MAJ			MIN				

**CHIP\_REV\_ID Bit Descriptions**

Bit	Name	Description
63:8	RSVD	<b>Reserved</b>
7:4	MAJ	<b>Major Revision.</b> Identifies major silicon revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.
3:0	MIN	<b>Minor Revision.</b> Identifies minor silicon revision. See <i>AMD Geode™ CS5535 Companion Device Specification Update</i> document for value.





# Electrical Specifications

# 7

This chapter provides information about:

- General Specifications
- DC Characteristics
- AC Characteristics

Throughout this section, the following abbreviations apply:

C	degrees centigrade
mA	milli amps
MHz	mega hertz
ms	milli seconds
mV	milli volts
NA	not applicable
ns	nano seconds
pF	pico farads
t <sub>ENABLE</sub>	Time Enable
t <sub>HOLD</sub>	Time Hold
t <sub>SETUP</sub>	Time Setup
t <sub>VAL</sub>	Time Valid
V	volts
μA	micro amps
μs	micro seconds

## 7.1 General Specifications

### 7.1.1 Electro Static Discharge (ESD)

This device is a high performance integrated circuit and is ESD sensitive. Handling and assembly of this device should be performed at ESD free workstations. Table 7-1 lists the ESD ratings of the AMD Geode™ CS5535 companion device.

**Table 7-1. Electro Static Discharge (ESD)**

Parameter	Units
Human Body Model (HBM)	2000V ESD
Machine Model (MM)	200V ESD

### 7.1.2 Power/Ground Connections and Decoupling

When testing and operating this component, use standard high frequency techniques to reduce parasitic effects. For example:

- Filter the DC power leads with low-inductance decoupling capacitors.
- Use low-impedance wiring.
- Utilize all power and ground connections.

### 7.1.3 Absolute Maximum Ratings

Stresses beyond those indicated in Table 7-2 may cause permanent damage to the component, reduce device reliability, and result in premature failure, even when there is no immediately apparent sign of failure. Prolonged exposure to conditions at or near the absolute maximum ratings may also result in reduced device life span and reduced reliability.

**Note:** The values in the Table 7-2 are stress ratings only. They do not imply that operation under these conditions is possible.

**Table 7-2. Absolute Maximum Ratings**

Parameter	Min	Max	Units	Comment
Operating Case Temperature	-65	110	C	Power applied and no clocks.
Storage Temperature	-65	150	C	No power applied.
Core Supply		1.6	V	
I/O Supply		3.6	V	
Voltage on Non-5V Tolerant Balls	-0.5	3.6	V	
Voltage on 5V Tolerant Balls	-0.5	5.5	V	

### 7.1.4 Recommended Operating Conditions

**Table 7-3. Recommended Operating Conditions**

Symbol	Parameter	Min	Typ	Max	Units
V <sub>CORE</sub>	Core Supply Voltage, Working Domain (Note 1)	1.14	1.2 or 1.5	1.58	V
V <sub>CORE_VSB</sub>	Core Supply Voltage, Standby Domain (Note 1)	1.14	1.2 or 1.5	1.58	V
V <sub>IO</sub>	I/O Supply Voltage, Working Domain	3.14	3.3	3.46	V
V <sub>IO_VSB</sub>	I/O Supply Voltage, Standby Domain	3.14	3.3	3.46	V
A <sub>VDD_USB</sub>	USB Transceiver Supply Voltage	3.14	3.3	3.46	V
V <sub>BAT</sub>	Real-time Clock Battery	2.4	3.0	3.6	V
T <sub>CASE</sub>	Case Temperature of Package	0		85	C
Input Timing	Input Rise and Fall Times (unless otherwise indicated) See Figure 7-1 "Clock Reference Definition" on page 550 for rise and fall definition.	0.5		5	ns

Note 1. The Geode™ CS5535 companion device is designed to be used with the Geode GX processor; as such, their core voltage ranges are compatible.

### 7.1.5 Current Consumption

**Table 7-4. Current Consumption**

Symbol	Description	Typ Avg	Abs Max	Units	Comments
I <sub>CORE_ON</sub>	Core plus Standby Core Current	60	125	mA	CPU Mode = Full On Typ = Typical operating conditions, clock gating on Max =Maximum operating conditions, clock gating off
I <sub>IO_ON</sub>	I/O plus Standby I/O Current	12	25	mA	CPU Mode = Full On Typ = Typical operating conditions, clock gating on Max =Maximum operating conditions, clock gating off
I <sub>VDD_USB</sub>	USB Current	4	5	mA	
I <sub>CORE_Sleep</sub>	Core plus Standby Core Current		21	mA	SUSPA# active All clocks stopped
I <sub>IO_Sleep</sub>	I/O plus Standby I/O Current		5	mA	All I/Os driven low or TRI-STATE
I <sub>CORE_VSB_Standby</sub>	Standby Core Current		<1	mA	All operating conditions.
I <sub>IO_VSB_Standby</sub>	Standby I/O Current		<1	mA	No V <sub>IO_VSB</sub> powered output sourcing current.
I <sub>BAT</sub>	Battery Current @ V <sub>BAT</sub> = 3.0V (Nominal), 25°C		5	µA	Use for battery life calculation. When off, system quickly reaches ambient temperature.
	Battery Current @ V <sub>BAT</sub> = 3.0V (Max), 85°C		10	µA	

## 7.2 DC Characteristics

All DC parameters and current specifications in this section are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ) unless otherwise specified.

For a detailed explanation of buffer types for the parameters listed in Table 7-5 on page 547, see Table 3-4 "Buffer Type Characteristics" on page 33.

**Table 7-5. DC Characteristics**

Symbol	Buffer Type	Min	Typ	Max	Units	Comment/Condition
$V_{IL}$	Low Level Input Voltage					
	Q3, Q5, Q7	-0.3	0.5	0.8	V	
	PCI	-0.3	0.5	$0.3 \cdot V_{IO}$	V	
	IDE	-0.3	0.5	0.8	V	
	SMB	-0.3	0.5	0.8	V	
	USB Receiver	-0.3		0.8		Single ended operation. Note 1, and Note 2.
	RESET_STAND# (Bare_Wire)	-0.3	0.5	0.8	V	
$V_{IH}$	High Level Input Voltage					
	Q3, Q5, Q7	2.0	3.0	$V_{IO}+0.3$	V	
	PCI	$0.5 \cdot V_{IO}$	3.0	$V_{IO}+0.3$	V	
	IDE	2.0	3.0	$V_{IO}+0.3$	V	
	SMB	2.1	3.0	5.5	V	5V tolerant, backdrive (back-powered) protected. Note 3.
	USB Receiver	2.0		$A_{VDD\_USB}+0.3$	V	Single ended operation. Note 1 and Note 2.
	RESET_STAND# (Bare_Wire)	2.0	3.0	$V_{IO}+0.3$	V	
$V_{OL}$	Low Level Output Voltage					
	Q3, Q5, Q7			0.4	V	$I_{OL} = 24$ mA.
	PCI			$0.1 \cdot V_{IO}$	V	$I_{OL} = 1.5$ mA.
	IDE			0.4	V	$I_{OL} = 16$ mA.
	SMB			0.4	V	$I_{OL} = 4$ mA.
	USB Driver			0.3	V	1.5 k $\Omega$ resistor to $A_{VDD\_USB}$ dp or dn of each differential pair. Note 2
$V_{OH}$	High Level Output Voltage					
	Q3, Q5, Q7	2.4			V	at $I_{OH} = -24$ mA.
	PCI	$0.9 \cdot V_{IO}$			V	at $I_{OH} = -500$ $\mu$ A.
	IDE	2.4			V	at $I_{OH} = -16$ mA.
	SMB		NA			Open-drain.
	USB Driver	2.8			V	15 k $\Omega$ resistor to ground. dp or dn of each differential pair. Note 2

Table 7-5. DC Characteristics (Continued)

Symbol	Buffer Type	Min	Typ	Max	Units	Comment/Condition
$I_{LLeak}$ (Note 4)	Input Leakage Current					
	Q3, Q5, Q7	NA	+/- 3		$\mu A$	Driver output disabled. $V_{pad} = 0$ to $V_{IO}$ .
	PCI	NA	+/- 5		$\mu A$	Driver output disabled. $V_{pad} = 0$ to $V_{IO}$ .
	IDE	NA	+/- 2		$\mu A$	Driver output disabled. $V_{pad} = 0$ to $V_{IO}$ .
	SMB	NA	+/- 5		$\mu A$	Driver output disabled. $V_{pad} = 0$ to $V_{IO}$ .
		NA	5		$\mu A$	Driver output disabled. Note 3. $V_{pad} = V_{IO}$ to 5.5.
		NA	5		$\mu A$	Note 3 and Note 5. $V_{pad} = 0$ to 5.5.
	USB	NA	+/- 3		$\mu A$	Driver output disabled. Note 2 and Note 6. $V_{pad} = 0$ to $A_{VDD\_USB}$ .
RESET_STAND# (Bare_Wire)	NA	+/- 3		$\mu A$	$V_{pad} = 0$ to $V_{IO}$ .	
$I_{PU}$ (Note 7)	Pull-Up Current					
	Q3, Q5, Q7	-50	100	-150	$\mu A$	Pull-up on and pull-down off. Output TRI-STATE and $V_{pad} = 0$ .
$I_{PD}$ (Note 7)	Pull-Down Current					
	Q3, Q5, Q7	20	45	130	$\mu A$	Pull-up off and pull-down on. Output TRI-STATE and $V_{pad} = V_{IO}$ .

Note 1. Actual value is adjustable. Values in this table require  $VADJ[2:0] = 4$  (011b):

--RCVR2\_VADJ: USBC1 MSR 51600008h[16:14] and USBC2 MSR 51200008h[16:14];

--(RCVR1\_VADJ: USBC1 MSR 51600008h[7:5] and USBC2 MSR 51200008h[7:5].

The single ended receiver incorporates hysteresis for more reliable operation.

Under any mode of operation, the USB inputs do not meet the requirements of "Figure 7-1 Maximum Input Waveform For USB Signaling" from the *USB Specification v1.1*. The maximum voltage can not exceed the value above.

Note 2. USB v1.1 compliance is achieved with external crimp protection diodes.

Note 3. The following SMB I/Os are limited to an input high max of  $V_{IO}$ : FUNC\_TEST, SMB\_CLK, and SMB\_DATA.

Note 4. This parameter is sometimes referred to as TRI-STATE leakage.

Note 5.  $V_{CORE}$  and  $V_{CORE\_VSB} = 0$ ,  $V_{IO}$  and  $V_{IO\_VSB} = 0$ ,  $T_{CASE} = All$ .

Note 6.  $A_{VDD\_USB}$  from 0 to  $A_{VDD\_USB\_MAX}$ .

Note 7. No pull-ups/downs on PCI, IDE, and SMB I/O cell types.

Table 7-6. Other USB DC Parameters

Signal	Parameter	Min	Max	Units	Comment/Condition
Differential Pair: USB1_1_DATPOS USB1_1_DATNEG	Output $V_{CRS}$ (cross over)	1.3	2.0	V	Same for low and full speed. Note 1.
Differential Pair: USB1_2_DATPOS USB1_2_DATNEG  Differential Pair: USB2_1_DATPOS USB2_1_DATNEG  Differential Pair: USB2_2_DATPOS USB2_2_DATNEG	Input Differential Voltage Sensitivity	200		mV	Same for low and full speed. Note 2 and Note 3.

Note 1. Crossover voltage defined in Figure 7-8 "USB Output Parameter Definition" on page 560.

Note 2. See Figure 7-9 "USB Input Parameter Definition" on page 561.

Note 3. When both differential data inputs are in the Differential Input Voltage Common Mode Range ( $V_{CM}$ ) of 0.8V to 2.5V as illustrated in Figure 7-10 "USB Differential Input Sensitivity Range" on page 561.

### 7.3 AC Characteristics

Unless otherwise indicated, no inputs have a specified hysteresis. Figure 7-1 and Figure 7-2 provide reference definitions used in this section.

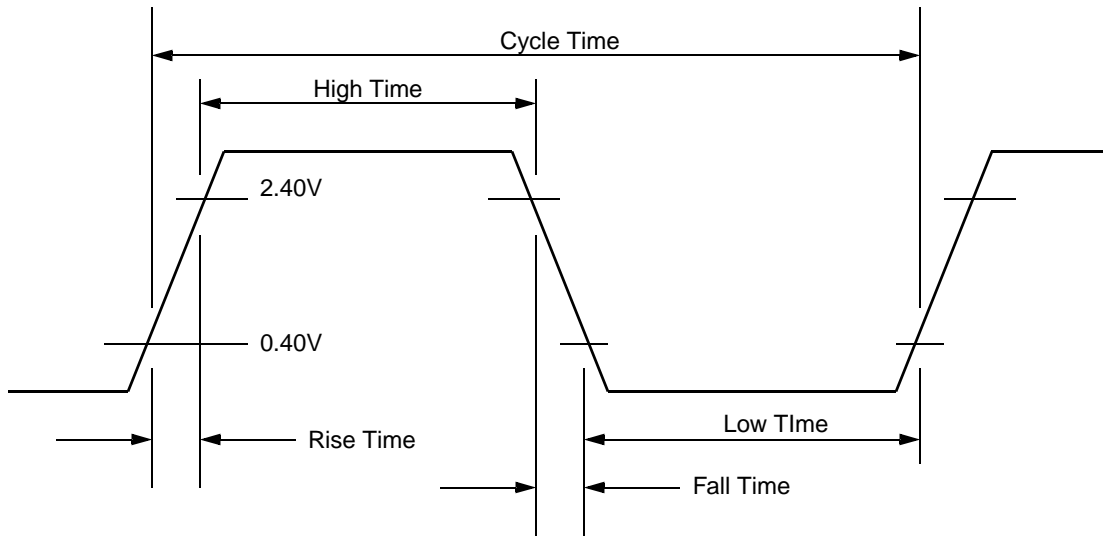


Figure 7-1. Clock Reference Definition

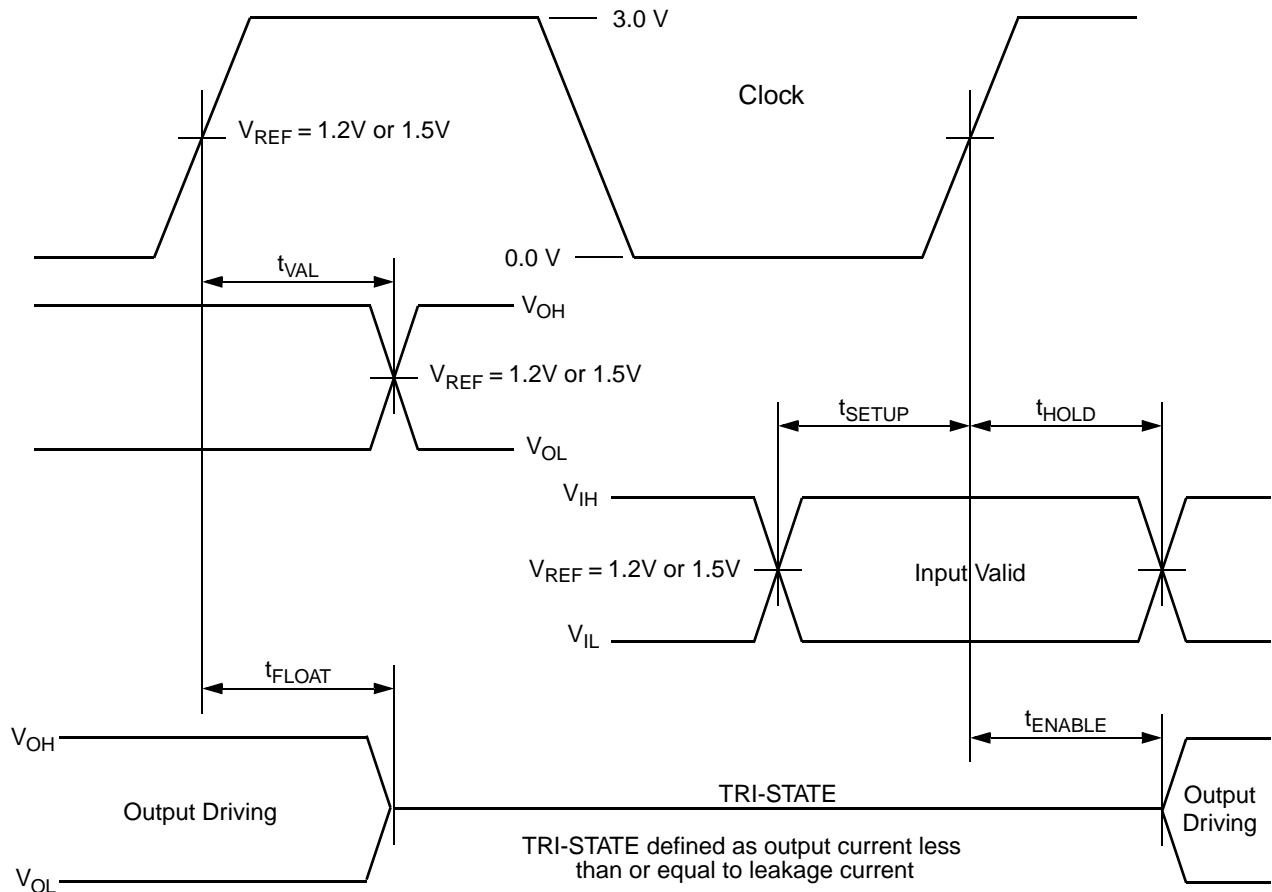
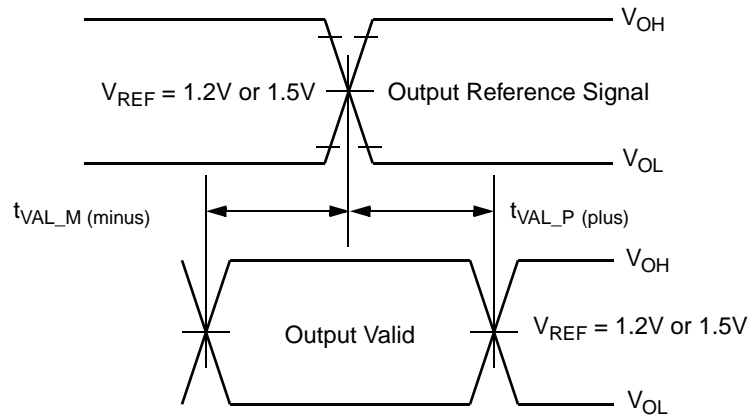
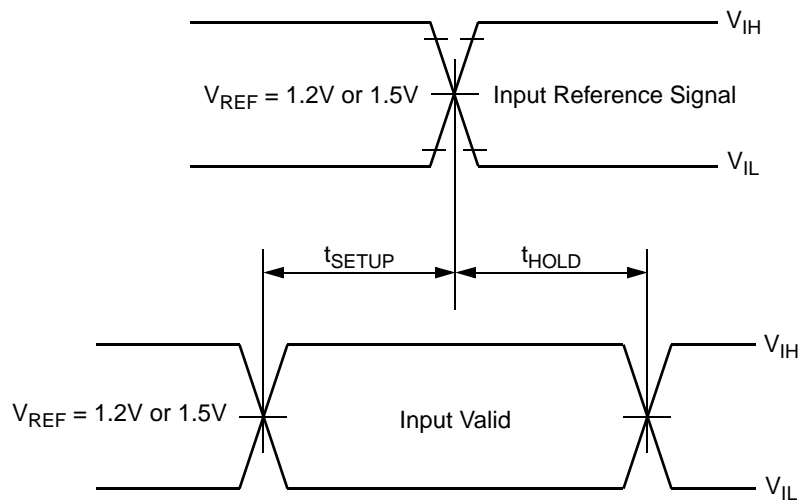


Figure 7-2. AC Reference Timing and Test Definition



**Figure 7-3. Output Reference Timing and Test Definition**



**Figure 7-4. Input Reference Timing and Test Definition**

### 7.3.1 Clock Inputs

All of the clocks in Table 7-7 indicate a minimum frequency of zero. Specifically, there are no dynamic circuits with minimum clock speeds. Additionally, *Active Hardware Clock Gating* (AHCG) (see Section 4.9 "Power Management" on page 79) will turn off all or some of the system clocks at selected points in time to save power. The

KHZ32\_XCI/KH32XCO being the one exception to "off all". It always runs.

While the above discussion is accurate, there are minimum clock requirements for proper system operation. These are indicated in the Notes for each clock in Table 7-7.

**Table 7-7. Clock Timing Parameters**

Signal	Parameter	Min	Typ	Max	Units	Comment/Condition
MHZ66_CLK	Frequency	0	66.00	66.50	MHz	Note 1 and Note 2.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
PCI_CLK	Frequency	0	33 or 66	67.50	MHz	Note 1 and Note 4.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
MHZ48_CLK	Frequency	0	48.00	49.25	MHz	Note 1 and Note 5.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
LPC_CLK	Frequency	0	33	34	MHz	Note 1, Note 4, and Note 6.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
MHZ14_CLK	Frequency	0	14.31818	15.00	MHz	Note 1 and Note 7.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
AC_CLK	Frequency	0	12.239	13.00	MHz	Note 1 and Note 8.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
TCK	Frequency	0	4.0	15.00	MHz	Note 1.
	Rise/Fall Time	0.5		2	ns	
	High/Low Time	6		None	ns	Note 3.
KHZ32_XCI KHZ32_XCO	Frequency	0	32.768	1000	kHz	Note 1 and Note 9.
	High/Low Time	0.5		None	μs	Note 3.

Note 1. Signal parameters are defined in Figure 7-1 "Clock Reference Definition" on page 550.

Note 2. Operationally, the minimum clock is 64.50 MHz. Operation out of the 66 MHz range indicated causes the Geode CS5535 companion device ATA Controller to operate out of ATA specification limits.

Note 3. Clock duty cycles not 100% tested. Guaranteed by design.

Note 4. For maximum system performance, this clock should be as high as possible up to the maximum indicated.

Note 5. The USB spec requires a controller accuracy of +/- 0.05% (500 ppm).

Note 6. Must be greater than half the MHZ14\_CLK frequency.

Must be at least four times faster than the KHZ32\_XCI frequency.

Note 7. This clock is used as the system time base and hence should have the "typical" frequency indicated.

Note 8. This clock should be connected to the external codec output that is half the codec input clock of 24.478 MHz.

Note 9. Typically, connect these pins to a 32.768 kHz crystal. However, an external oscillator may be connected to the KHZ32\_XCI pin and driven to the maximum rate shown for testing. When operated with an external oscillator, leave the KHZ32\_XCO pin open. When operating with an external oscillator, input voltage on KHZ32\_XCI (khz32\_xci) should swing rail-to-rail, that is, zero-to- $V_{IO\_VSB}$ . With external oscillator, the input voltage high should always be at least 3.0 volts.



## 7.3.2 Reset and Test Inputs

Table 7-8. Reset and Test Timing Parameters

Signal	Parameter	Min	Max	Units	Comment/Condition
RESET_WORK#	Rise/Fall Time	0.5	5	ns	Note 1.
	Low Time	3	None	ns	Time required to detect a reset.
	High Time	3	None	ns	
	Cycle Time does not apply. Once the component has started a reset operation, cycling this input generally does not apply.				
RESET_STAND#	Rise/Fall Time	0.5	5	ns	Note 1.
	Low Time	3	None	ns	Time required to detect a reset.
	High Time	3	None	ns	
	Cycle Time does not apply. Once the component has started a reset operation, cycling this input generally does not apply.				
LVD_EN#	Static signal. Tie high or low as indicated in the signal description.				
TEST_MODE	Static signal. Operationally, always tie low.				
FUNC_TEST	Static signal. Operationally, always tie low.				

Note 1. Signal parameters are defined in Figure 7-1 "Clock Reference Definition" on page 550.

### 7.3.3 PCI and Related Signals

The signals detailed in this sub-section use a "PCI" buffer type except SUSP#, SUSPA#, and RESET\_OUT# they use types "Q3" and "Q7" respectively. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- Signals are referenced to PCI\_CLK low-to-high edge.
- Signal and test parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-9. PCI, SUSP#, SUSPA#, and RESET\_OUT# Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
PCI_INTA# (GPIO0) PCI_INTB# (GPIO7)	Async Input	NA	NA	ns	No clock reference. See Section 7.3.13 "GPIO Signaling" on page 568.
REQ#	$t_{VAL}$	2	6	ns	Note 1.
GNT#	$t_{SETUP}$	5	NA	ns	
	$t_{HOLD}$	0	NA	ns	
CBE[3:0]#, DEVSEL#, FRAME#, TRDY#, IRDY#, STOP#, PAR, AD[31:0]	$t_{VAL}$	2	6	ns	Applies when signal is an output. Note 1.
	$t_{FLOAT}$	NA	8	ns	Applies when signal is an output. Note 2.
	$t_{ENABLE}$	2	NA	ns	Applies when signal is an output. Note 3.
	$t_{SETUP}$	3	NA	ns	Applies when signal is an output.
	$t_{HOLD}$	0	NA	ns	
SUSP#	$t_{VAL}$	1	6	ns	Note 1.
RESET_OUT#	$t_{VAL}$	3	10	ns	Note 4.
SUSPA#	$t_{SETUP}$	3	NA	ns	
	$t_{HOLD}$	0	NA	ns	

Note 1.  $t_{VAL}$  min times with load of: 10 pF cap to ground.  
 $t_{VAL}$  max times with load of: 35 pF cap to ground.

Note 2.  $t_{FLOAT}$  with the load of: 10 pF cap to ground.

Note 3.  $t_{ENABLE}$  with the load of: 10 pF cap to ground.

Note 4.  $t_{VAL}$  min times with load of: 10 pF cap to ground  
 $t_{VAL}$  max times with load of: 50 pF cap to ground.

### 7.3.4 IDE Signals in IDE Mode

The signals detailed in this sub-section use an "IDE" buffer type. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

- Signals are referenced to MHZ66\_CLK low-to-high edge.
- Signal and test parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-10. IDE Register, PIO, and Multiword DMA Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
IDE_CS[1:0]#, IDE_IOW0, IDE_AD[2:0], IDE_RST#, IDE_DACK0#	$t_{VAL}$	2	10	ns	IDE_DACK0# is only used for DMA. Note 1 and Note 2.
IDE_IOR0#	$t_{VAL}$	3•	13	ns	Note 1 and Note 2.
IDE_DATA[15:0] for Write					Note 3.
IDE_DATA[15:0] for Read					Note 4 and Note 5.
IDE_RDY0, IDE_IRQ0, IDE_DREQ0#	Async Input	NA	NA	ns	No clock reference. Note 6. IDE_IRQ0 and IDE_DREQ0# are only used for DMA.

Note 1. Per the ATA/ATAPI-5 spec, these signals utilize Output Reference Timing (see Figure 7-3 on page 551) relative to IDE\_IOR0# and IDE\_IOW0#. However, the IDE Controller uses the MHZ66\_CLK edges to make output changes, that when taken together, meet all the timing requirements of the referenced spec. Therefore,  $t_{VAL}$  times are specified and tested relative to the MHZ66\_CLK.

Note 2.  $t_{VAL}$  min times with load of: 15 pF cap to ground.  
 $t_{VAL}$  max times with load of: 40 pF cap to ground.

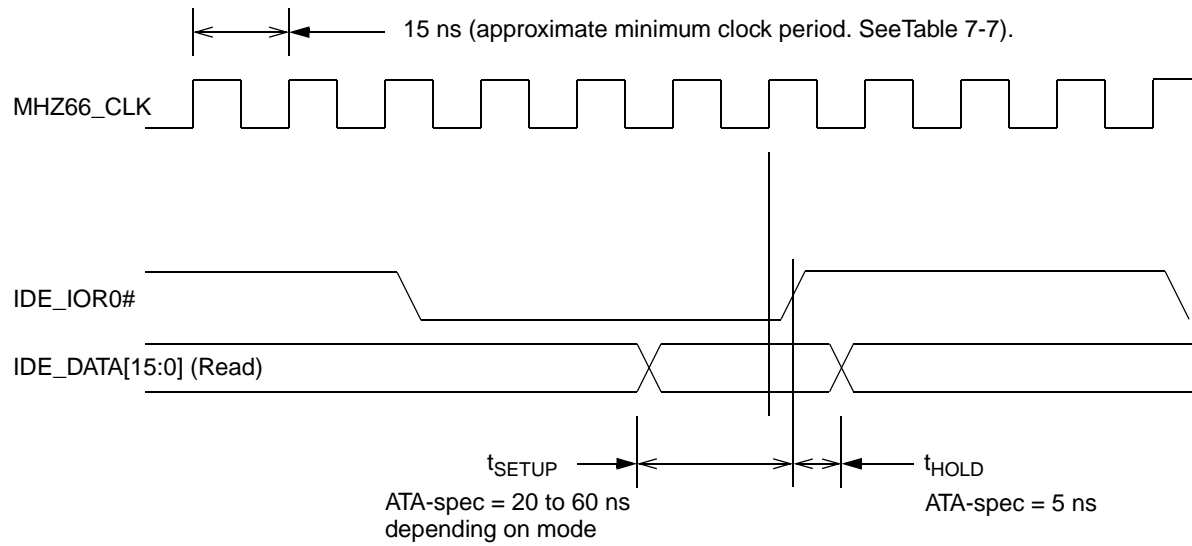
Note 3. Per the ATA/ATAPI-5 spec, IDE\_DATA write signals utilize *Output Reference Timing* (see Figure 7-3 on page 551) relative to IDE\_IOW0#. However, the IDE Controller uses the MHZ66\_CLK edges to make output changes, and when taken together, meet all the timing requirements of the referenced spec. See Figure 7-5 "IDE Data In Timing Non-UltraDMA" on page 556.

Note 4. Per the ATA/ATAPI-5 spec, these signals utilize *Input Reference Timing* (see Figure 7-4 on page 551) relative to IDE\_IOR0#. However, the IDE Controller samples the inputs with the MHZ66\_CLK at the appropriate points in time to meet all the timing requirements of the referenced spec.

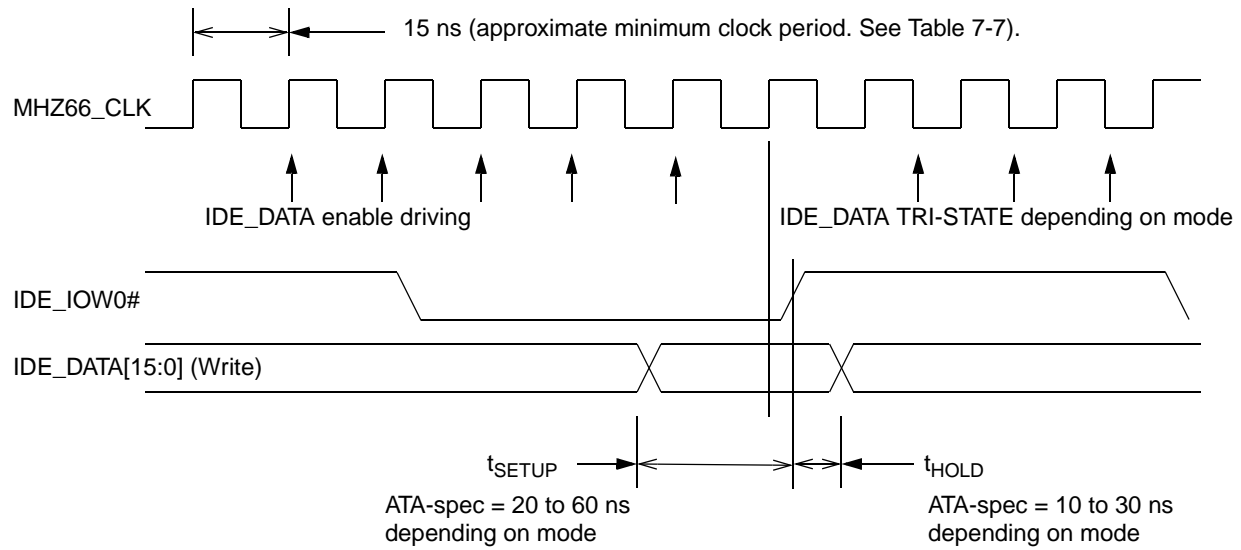
Note 5. See Figure 7-5 "IDE Data In Timing Non-UltraDMA". As indicated in Note 4, IDE\_DATA[15:0] read input is specified relative to the low-to-high edge of IDE\_IOR0#. However, the implementation meets a tighter test spec referenced to the low-to-high edge of the MHZ66\_CLK.

Note 6. For IDE\_IRQ0, GPIO2 configured: Aux In and Input Enable = 1; Output Enable, Aux Out 1, and Aux Out 2 = 0.

**Read Operations**



**Write Operations**

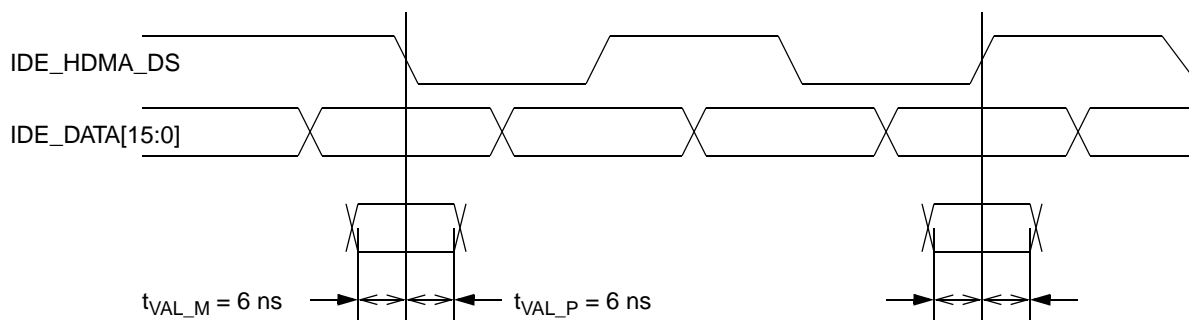


**Figure 7-5. IDE Data In Timing Non-UltraDMA**

**Table 7-11. IDE UltraDMA Data Out Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
IDE_DACK0# IDE_HDMA_DS, IDE_STOP, IDE_IOW0#	$t_{VAL}$	2	10	ns	In Ultra DMA/33 mode, the IDE_IOR0# signal is redefined as IDE_HDMA_DS. Note 1 and Note 2.
IDE_DATA[15:0]	$t_{VAL}$	See Notes		ns	Note 2 and Note 3.
IDE_DREQ0	Async Input	NA	NA	ns	No clock reference. Note 4.
IDE_DDMA_RDY	Async Input	NA	NA	ns	In Ultra DMA/33 mode, the IDE_RDY0 signal is redefined as IDE_DDMA_RDY. Note 4.
IDE_IRQ0	Async Input	NA	NA	ns	Note 4 and Note 5.

- Note 1. Per the ATA/ATAPI-5 spec, these signals utilize Output Reference Timing (see Figure 7-3 on page 551) relative to IDE\_IOR0# and IDE\_IOW0#. However, the IDE Controller uses the MHZ66\_CLK edges to make output changes, that when taken together, meet all the timing requirements of the referenced spec. Therefore,  $t_{VAL}$  times are specified and tested relative to the MHZ66\_CLK.
- Note 2.  $t_{VAL}$  min times with load of: 15 pF cap to ground.  
 $t_{VAL}$  max times with load of: 40 pF cap to ground.
- Note 3. This signal uses *Output Reference Timing* (see Figure 7-3 on page 551). Figure 7-6 illustrates IDE\_HDMA\_DS and IDE\_DATA[15:0] relationship.
- Note 4. Per the ATA/ATAPI-5 spec, these signals utilize *Input Reference Timing* (see Figure 7-4 on page 551) relative to IDE\_IOR0#. However, the IDE Controller samples the inputs with the MHZ66\_CLK at the appropriate points in time to meet all the timing requirements of the referenced spec.
- Note 5. For IDE\_IRQ0, GPIO2 configured: IN\_AUX1 and Input Enable = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

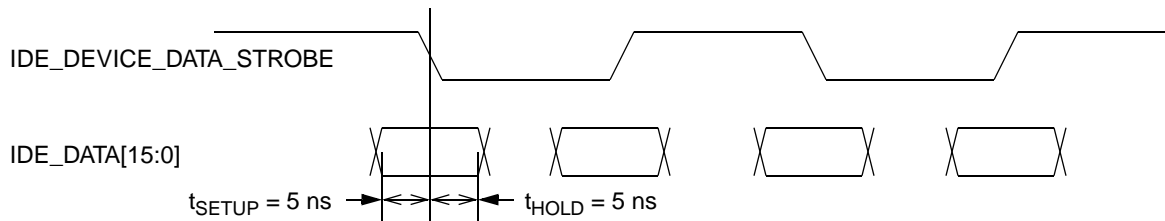


**Figure 7-6. IDE UltraDMA Data Out Timing**

**Table 7-12. IDE UltraDMA Data In Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
IDE_DACK0#, IDE_HDMA_RDY, IDE_STOP, IDE_IOW0	t <sub>VAL</sub>	2	10	ns	In Ultra DMA/33 mode, the IDE_IOR0# signal is redefined as IDE_HDMA_RDY. Note 1 and Note 2.
IDE_DDMA_DS	Async Input	NA	NA	ns	In Ultra DMA/33 mode, the IDE_RDY0 signal is redefined as IDE_DDMA_DS. Note 3.
IDE_DATA[15:0]	Sync to IDE_DDMA_DS	see Note		ns	Note 3.
IDE_DREQ0#	Async Input	NA	NA	ns	No clock reference. Note 4.
IDE_IRQ0	Async Input	NA	NA	ns	Note 4 and Note 5.

- Note 1. Per the ATA/ATAPI-5 spec, these signals utilize Output Reference Timing (see Figure 7-3 on page 551) relative to IDE\_IOR0# and IDE\_IOW0#. However, the IDE Controller uses the MHZ66\_CLK edges to make output changes, that when taken together, meet all the timing requirements of the referenced spec. Therefore, t<sub>VAL</sub> times are specified and tested relative to the MHZ66\_CLK.
- Note 2. t<sub>VAL</sub> min times with load of: 15 pF cap to ground.  
t<sub>VAL</sub> max times with load of: 40 pF cap to ground.
- Note 3. These signals use *Input Reference Timing* (see Figure 7-4 on page 551). Figure 7-7 illustrates their relationship and specified setup and hold times.
- Note 4. Per the ATA/ATAPI-5 spec, these signals utilize *Input Reference Timing* (see Figure 7-4 on page 551) relative to IDE\_IOR0#. However, the IDE Controller samples the inputs with the MHZ66\_CLK at the appropriate points in time to meet all the timing requirements of the referenced spec.
- Note 5. For IDE\_IRQ0, GPIO2 configured: IN\_AUX1 and Input Enable = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.



**Figure 7-7. IDE UltraDMA Data In Timing**

### 7.3.5 IDE Signals in Flash Mode

The signals detailed in this sub-section use an "IDE" buffer type. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{\text{CORE}}$  and  $V_{\text{CORE\_VSB}} = \text{All}$ ;  $V_{\text{IO}}$  and  $V_{\text{IO\_VSB}} = \text{All}$ ;  $T_{\text{CASE}} = \text{All}$ ).

- Signals are referenced to LPC\_CLK.
- Signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-13. Flash Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
FLASH_CS[3:0]#, FLASH_RE#, FLASH_WE, FLASH_ALE, FLASH_CLE, FLASH_AD[9:0], FLASH_IO[7:0] signals and I/Os in Flash output mode	$t_{\text{VAL}}$	2	9	ns	Note 1.
FLASH_IO[7:0] signals and I/Os in Flash input mode except FLASH_IOCHRDY	$t_{\text{SETUP}}$	8	NA	ns	
	$t_{\text{HOLD}}$	0	NA	ns	
FLASH_IOCHRDY	Async Input	NA	NA	ns	No clock reference.

Note 1.  $t_{\text{VAL}}$  min times with load of: 10 pF cap to ground.  
 $t_{\text{VAL}}$  max times with load of: 50 pF cap to ground.

**7.3.6 USB Signals**

All signals detailed in this sub-section use an "IDE" buffer type. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

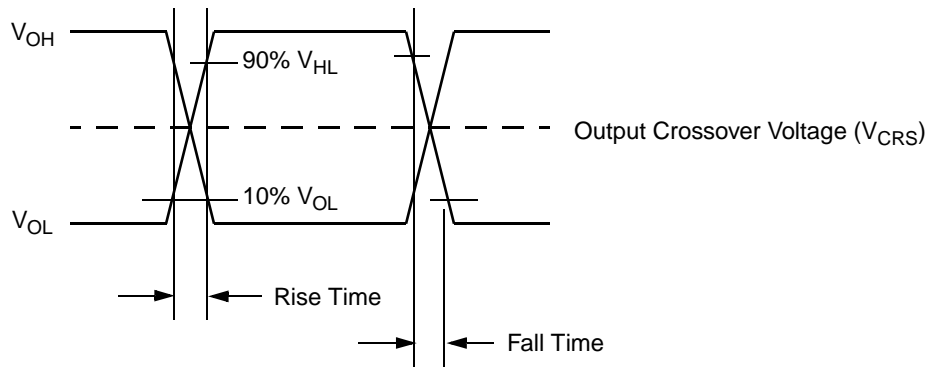
In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

**Table 7-14. USB Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
USB_PWR_EN1, USB_PWR_EN2	Async Output or Level	NA	NA	ns	No clock reference.
	Async Input	NA	NA	ns	No clock reference.
USB_OC_SENS#	Output Rise and Fall Times (full speed)	4	20	ns	Note 1, Note 2, and Note 3.
Differential Pair: USB1_1_DATPOS, USB1_1_DATNEG, Differential Pair: USB1_2_DATPOS, USB1_2_DATNEG, Differential Pair: USB2_1_DATPOS, USB2_1_DATNEG usb2_portdn1 Differential Pair: USB2_2_DATPOS, USB2_2_DATNEG	Output Rise and Fall Times (low speed)	75	300	ns	Note 1, Note 2, and Note 4. Measured at the downstream host end.

- Note 1. Rise time and fall time defined in Figure 7-8 "USB Output Parameter Definition" on page 560.  
 Note 2. The rise and fall time values for a given differential pair are matched within 10%. Edges are monotonic.  
 Note 3. Actual value is adjustable. Values in this table require CADJ[4:0] = 0x10b:  
 --XMIT2\_CADJ: USBC1 MSR 51600008h[13:9] and USBC2 MSR 51200008h[4:0];  
 --(XMIT1\_CADJ: USBC1 MSR 51600008h[7:5] and USBC2 MSR 51200008h[7:5].  
 Note 4. The rise and fall time values for a given differential pair are matched within 20%. Edges are monotonic.



**Figure 7-8. USB Output Parameter Definition**



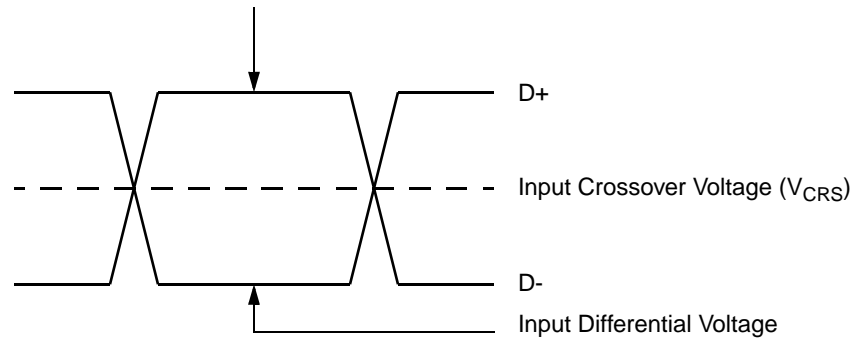


Figure 7-9. USB Input Parameter Definition

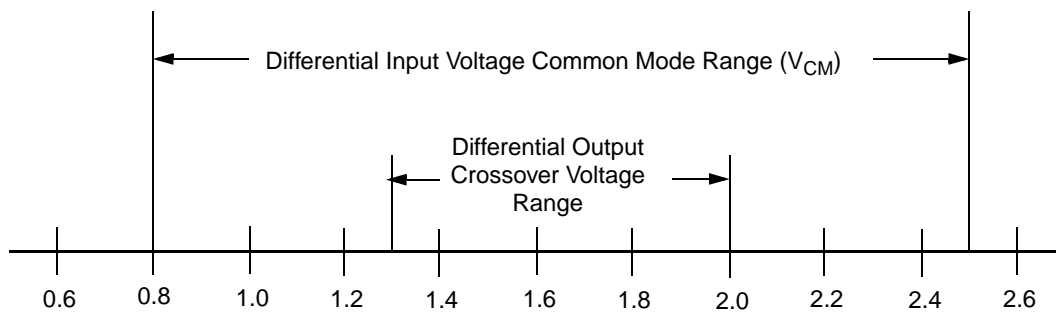


Figure 7-10. USB Differential Input Sensitivity Range

### 7.3.7 System Management Bus (SMB) Signals

The signals detailed in this sub-section use an “SMB” buffer type. For a detailed explanation of buffer types, see Table 3-4 “Buffer Type Characteristics” on page 33.

The SMB utilizes a two-wire asynchronous protocol. Master and slave devices are connected open-drain with an external pull-up resistor. The SMB\_CLK is not a free running fixed frequency signal, but rather a cooperatively generated signal with a minimum low time of 4.7  $\mu$ s, minimum high time of 4.0  $\mu$ s, and 100 kHz frequency limit. The minimum frequency is 10 kHz. The SMB\_DATA signal is also cooperatively driven. Communication on the SMB is via rel-

ative manipulation of these two signals. The SMB Controller and I/O cell incorporated within the Geode CS5535 companion device fully meets the requirements of the SMB specification version 2.0.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

**Table 7-15. SMB Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
Bidirectional SMB_CLK (GPIO14), SMB_DATA (GPIO15)	Async Output or Level	NA	NA	ns	No clock reference.
	Async Input	NA	NA	ns	No clock reference.

### 7.3.8 AC97 Codec Signals

The signals detailed in this sub-section use a "Q7" buffer type. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{\text{CORE}}$  and  $V_{\text{CORE\_VSB}} = \text{All}$ ;  $V_{\text{IO}}$  and  $V_{\text{IO\_VSB}} = \text{All}$ ;  $T_{\text{CASE}} = \text{All}$ ).

- Signals are referenced to AC\_CLK low-to-high edge.
- Signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-16. AC97 Codec Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
AC_S_OUT, AC_S_SYNC	$t_{\text{VAL}}$	2	15	ns	Note 1.
AC_S_IN, AC_S_IN2 (GPIO12)	$t_{\text{SETUP}}$	10	NA	ns	Note 2 and Note 3.
	$t_{\text{HOLD}}$	10	NA	ns	
AC_BEEP (GPIO1)	$t_{\text{VAL}}$	2	25	ns	Note 1, Note 4. and Note 5.

Note 1.  $t_{\text{VAL}}$  min times with load of: 10 pF cap to ground.  
 $t_{\text{VAL}}$  max times with load of: 50 pF cap ground.

Note 2. Signals are referenced to AC\_CLK high-to-low edge.

Note 3. For AC\_S\_IN2, GPIO12 configured: (Aux In) and Input Enable = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

Note 4. Signal is referenced to LPC\_CLK low-to-high edge.

Note 5. For AC\_BEEP, GPIO1 configured: OUT\_AUX1 and Output Enable = 1; Open-drain, OUT\_AUX2, and Input Enable = 0.

### 7.3.9 Low Pin Count (LPC) Signals

The signals detailed in this sub-section use a "PCI" buffer type except for LPC\_CLK. LPC\_CLK uses a "Q7" buffer. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

- Signals are referenced to LPC\_CLK low-to-high edge.
- Signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-17. LPC Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
LPC_FRAME#, LPC_AD[3:0], LPC_DRQ#, LPC_SERIRQ	$t_{VAL}$ when signal is an output	2	11	ns	Note 1 and Note 2.
	$t_{FLOAT}$ when signal is an output	NA	11	ns	Note 2 and Note 3.
	$t_{ENABLE}$ when signal is an output	2	NA	ns	Note 2 and Note 4.
	$t_{SETUP}$ when signal is an input	7	NA	ns	Note 2.
	$t_{HOLD}$ when signal is an input	0	NA	ns	Note 2.

Note 1.  $t_{VAL}$  min times with load of: 10 pF cap to ground.

$t_{VAL}$  max times with load of: 50 pF cap to ground.

Note 2. All information in this table applies when the following control bits are high in Table 3-6 "DIVIL BALL\_OPT (MSR 51400015h)" on page 34: PIN\_OPT\_LDRQ, PIN\_OPT\_LIRQ, and PIN\_OPT\_LALL.

Note 3.  $t_{FLOAT}$  with the load of: 10 pF cap to ground.

Note 4.  $t_{ENABLE}$  with the load of: 10 pF cap to ground.

### 7.3.10 Power Management and Processor Control Signals

The Power Management Controller (PMC) signals detailed in this sub-section use various types of buffers depending upon chip configuration. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- No clock reference.

**Table 7-18. Power Management and Processor Control Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
WORKING	Async Output or Level	NA	NA	ns	From PMC.
RESET_OUT#	Sync Output to PCI_CLK	NA	NA	ns	See RESET_OUT# in Table 7-9 on page 554.
THRM_ALARM# (GPIO10)	Async Input	NA	NA	ns	Note 1. To PMC.
SLP_CLK_EN (GPIO11)	Async Output or Level	NA	NA	ns	Note 2. From PMC.
WORK_AUX (GPIO24)	Async Output or Level	NA	NA	ns	Note 2. From PMC.
LOW_BAT# (GPIO25)	Async Input	NA	NA	ns	Note 1. To PMC.
PWR_BUT# (GPIO28)	Async Input	NA	NA	ns	Note 1. To PMC.
, SUSPA#	Sync to PCI_CLK	NA	NA	ns	See Table 7-9 on page 554 for , SUSPA# data.
IRQ13	Async Input	NA	NA	ns	To PIC subsystem.
SLEEP_X (GPIO7)	Async Output or Level	NA	NA	ns	Note 3. From PMC.
SLEEP_Y (GPIO12)	Async Output or Level	NA	NA	ns	Note 3. From PMC.
SLEEP_BUT (GPIO13)	Async Input	NA	NA	ns	Note 1. To PMC.

Note 1. GPIO configured: IN\_AUX1 and Input Enable = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

Note 2. GPIO configured: IN\_AUX1 and Input Enable = 0; Output Enable and OUT\_AUX1 = 1; Open-drain and OUT\_AUX2 = 0.

Note 3. GPIO configured: IN\_AUX1 and Input Enable = 0; Output Enable and OUT\_AUX2 = 1; Open-drain and OUT\_AUX1 = 0.

### 7.3.11 Miscellaneous Signals

The “recommended use” for GPIO3 and GPIO4 is DDC support signals DDC\_SCL and DDC\_SDA, because these two GPIOs have a high drive capacity, open-drain output. They use an “SMB” buffer type. The 32 kHz clock output is a mux option on GPIO27 and uses a “Q7” buffer. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- Signals are referenced to KHZ32\_XCI high-to-low edge.
- Signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-19. Miscellaneous Signals Except UART Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
DDC_SCL (GPIO3) DDC_SDA (GPIO4)	NA	NA	NA	ns	See Section 7.3.13 "GPIO Signaling" on page 568.
32KHZ (GPIO27)	$t_{VAL}$	2	15	ns	Note 1 and Note 2.

Note 1.  $t_{VAL}$  min times with load of: 10 pF cap to ground.  
 $t_{VAL}$  max times with load of: 50 pF cap to ground.

Note 2. GPIO27 configured: IN\_AUX1 and Input Enable = 0; Output Enable and OUT\_AUX2 = 1; Open-drain and OUT\_AUX1 = 0.

### 7.3.12 UART and IR Signaling

The UART support signals detailed in this sub-section use various types of buffers depending upon chip configuration. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- No clock reference.

**Table 7-20. UART Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
UART1_TX (GPIO8)	Async Output or Level	NA	NA	ns	Note 1.
UART1_IR_TX (GPIO8)	Async Output or Level	NA	NA	ns	Note 2.
UART1_RX (GPIO9)	Async Input	NA	NA	ns	Note 3.
UART2_TX (GPIO4)	Async Output or Level	NA	NA	ns	Note 1.
UART2_RX (GPIO3)	Async Input	NA	NA	ns	Note 3

Note 1. GPIO configured: IN\_AUX1 and Input Enable = 0; Output Enable and OUT\_AUX1 = 1; Open-drain and OUT\_AUX2 = 0.

Note 2. GPIO configured: IN\_AUX1 and Input Enable = 0; Output Enable and OUT\_AUX2 = 1; Open-drain and OUT\_AUX1 = 0.

Note 3. GPIO configured: IN\_AUX1 and Input Enable = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

### 7.3.13 GPIO Signaling

The GPIO signals detailed in this sub-section use various types of buffers depending upon chip configuration. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

- GPIO[22:0] signals are referenced to LPC\_CLK and GPIO[28:24] signals are referenced to KHZ32\_XCI. Use low-to-high edge if LPC Clock; use high-to-low edge if 32kHz clock.
- GPIO signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-21. GPIO Signaling**

Signal GPIO[28:24, 22:0]	Parameter	Min	Max	Units	Comment/Condition
GPIO Input [all]	Async Input	NA	NA	ns	No clock reference. Can be read via programmed I/O. Can be used as an interrupt or PME. Note 1 and Note 2.
GPIO Output [22:16, 13:5, 2:0]	$t_{VAL}$	2	20	ns	Note 2, Note 3, and Note 4. (LPC_CLK)
GPIO Output [28:25]	$t_{VAL}$	4	20	ns	Note 2, Note 3, and Note 4. (KHZ32_XCI)
GPIO OD Output high-to-low data [15:14, 4:3]	$t_{VAL}$	2	25	ns	Note 2, Note 4, and Note 5. (LPC_CLK)
GPIO OD Output high-to-low data [24]	$t_{VAL}$	4	25	ns	Note 2, Note 4, and Note 5. (KHZ32_XCI)
GPIO OD Output low-to-high data [15:14, 4:3]	$t_{VAL}$	NA	28	ns	Note 2, Note 4, and Note 5 (LPC_CLK)
GPIO OD Output low-to-high data [24]	$t_{VAL}$	NA	28	ns	Note 2, Note 4, and Note 5. (KHZ32_XCI)

Note 1. GPIO configured: Input Enable = 1; IN\_AUX1, Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

Note 2. PIN\_OPT\_LALL = 0 in Table 3-6 "DIVIL\_BALL\_OPT (MSR 51400015h)" on page 34.

Note 3.  $t_{VAL}$  min times with load of: 10 pF cap to ground.

$t_{VAL}$  max times with load of: 50 pF cap to ground.

Note 4. GPIO configured: IN\_AUX1, Input Enable, OUT\_AUX1, OUT\_AUX2 = 0; Output Enable = 1; Open-drain = 0.

Note 5. Load per Table 7-15 "SMB Timing Parameters" on page 562.



### 7.3.14 MFGPT Signaling

The MFGPT signals detailed in this sub-section use various types of buffers depending upon chip configuration. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- MFGPT signals are referenced to KHZ32\_XCI or MHZ14\_CLK depending on MFGPT clock configuration. Use low-to-high edge if MHZ14\_CLK; use high-to-low edge if KHZ32\_XCI. MFGPT7 supports KHZ32\_XCI only.
- MFGPT signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-22. MFGPT Signaling**

Signal	Parameter	Min	Max	Units	Comment/Condition
Inputs: MFGPT0 (GPIO6) MFGPT1 (GPIO5) MFGPT2 (GPIO21) MFGPT7 (GPIO26)	Async Input	NA	NA	ns	No clock reference. Restarts the MFGPT. Note 1 and Note 2.
Outputs: MFGPT0_C1 (GPIO5) MFGPT1_C1 (GPIO6) MFGPT2_C1 (GPIO7) MFGPT7_C1 (GPIO27)	$t_{VAL}$ if MHZ14_CLK	2	20	ns	Note 2, Note 3, and Note 4.
	$t_{VAL}$ if KHZ32_XCI	2	20	ns	
Outputs: MFGPT0_C2 (GPIO1) MFGPT1_C2 (GPIO11) MFGPT2_C2 (GPIO6) MFGPT7_C2 (GPIO25)	$t_{VAL}$ if MHZ14_CLK	2	20	ns	Note 2, Note 3, and Note 5.
	$t_{VAL}$ if KHZ32_XCI	2	20	ns	

Note 1. GPIO configured: Input Enable and IN\_AUX1 = 1; Output Enable, OUT\_AUX1, and OUT\_AUX2 = 0.

Note 2. PIN\_OPT\_LALL = 0 in Table 3-6 "DIVIL BALL\_OPT (MSR 51400015h)" on page 34.

Note 3.  $t_{VAL}$  min times with load of: 10 pF cap to ground.  
 $t_{VAL}$  max times with load of: 50 pF cap to ground.

Note 4. GPIO configured: IN\_AUX1, Input Enable = 0; Output Enable = 1; Open-drain = 0; OUT\_AUX1 = 1; OUT\_AUX2 = 0.

Note 5. GPIO configured: IN\_AUX1, Input Enable = 0; Output Enable = 1; Open-drain = 0; OUT\_AUX1 = 0; OUT\_AUX2 = 1.

### 7.3.15 JTAG Signals

The TDI and TDO JTAG signals use a “Q5” buffer type while the TMS signal uses a “Q7”. For a detailed explanation of buffer types, see Table 3-4 "Buffer Type Characteristics" on page 33.

In this subsection, unless otherwise noted:

- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,

$V_{\text{CORE}}$  and  $V_{\text{CORE\_VSB}} = \text{All}$ ;  $V_{\text{IO}}$  and  $V_{\text{IO\_VSB}} = \text{All}$ ;  $T_{\text{CASE}} = \text{All}$ ).

- Signals are referenced to TCK low-to-high edge.
- Signal parameters are defined in Figure 7-2 "AC Reference Timing and Test Definition" on page 550.

**Table 7-23. JTAG Timing Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
TDO	$t_{\text{VAL}}$	2	16	ns	Note 1 and Note 2.
TMS, TDI	$t_{\text{SETUP}}$	10	NA	ns	
	$t_{\text{HOLD}}$	2	NA	ns	

Note 1.  $t_{\text{VAL}}$  min times with load of: 10 pF cap to ground.

$t_{\text{VAL}}$  max times with load of: 50 pF cap to ground.

Note 2. Signal is referenced to TCK high-to-low edge.

## 7.4 Power Supply Sequence Requirements

The voltages applied to the Geode CS5535 companion device are subject to the requirements listed below as well as the requirements of Table 7-3 "Recommended Operating Conditions" on page 546. Reference values "minimum" and "maximum" below should be taken from Table 7-3.

Note that the information below is provided for completeness. Since the specified margins are very wide, the typical power supply implementation meets them without problem.

If these requirements are not observed, the RTC circuit and/or the LVD circuit may not operate correctly.

- 1)  $V_{CORE}$  and  $V_{IO}$  may come up in any order but must meet their respective minimum values within 100 ms of each other.
- 2) From zero volts,  $V_{CORE}$  and  $V_{IO}$  must ramp up monotonically and reach 90% of their respective minimum values no sooner than 10  $\mu$ s and no later than 1 second.
- 3) From their respective minimum values,  $V_{CORE}$  and  $V_{IO}$  must ramp down to within 0.4 volts of ground no sooner than 100 ms and no later than 5 seconds.
- 4) Same as requirement 1, but for  $V_{CORE\_VSB}$  and  $V_{IO\_VSB}$ .
- 5) Same as requirement 2, but for  $V_{CORE\_VSB}$  ( $VDD\_VSB$ ) and  $V_{IO\_VSB}$ .
- 6) Same as requirement 3 but for  $V_{CORE\_VSB}$  and  $V_{IO\_VSB}$ .
- 7) The absolute value of the delta between  $V_{CORE}$  and  $V_{CORE\_VSB}$  must not exceed 0.25 volts.
- 8) The absolute value of the delta between  $V_{IO}$  and  $V_{IO\_VSB}$  must not exceed 0.25 volts.

### 7.5 Low Voltage Detect (LVD) Parameters

The LVD electrical parameters are defined in Figure 7-11 and listed in Table 7-24. Use of internal signals Power Good Standby And Power Good Working is illustrated in Figure 4-6 "Reset Logic" on page 66.

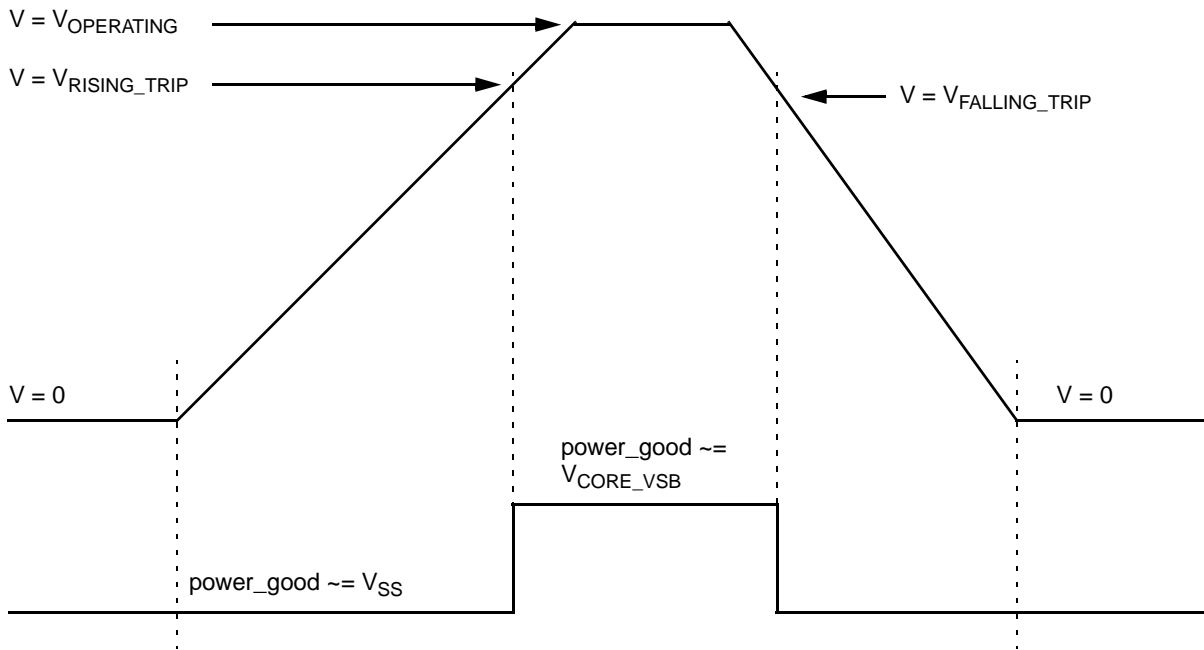
- All timing specifications are specified under the operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).
- The LVD circuit incorporates no clock signals.

In this subsection, unless otherwise noted:

**Table 7-24. LVD Parameters**

Signal	Parameter	Min	Max	Units	Comment/Condition
Power Good Standby	$V_{RISING\_TRIP\_CORE\_VSB}$	0.80	1.00	V	
	$V_{FALLING\_TRIP\_CORE\_VSB}$	0.72	0.90	V	
	$V_{HYSTERESIS\_CORE\_VSB}$	23	110	mV	Note 1.
	$V_{RISING\_TRIP\_IO\_VSB}$	2.15	3.05	V	
	$V_{FALLING\_TRIP\_IO\_VSB}$	2.32	3.05	V	
	$V_{HYSTERESIS\_IO\_VSB}$	100	580	mV	Note 1.
Power Good Working	$V_{RISING\_TRIP\_CORE}$	0.80	1.00	V	
	$V_{FALLING\_TRIP\_CORE}$	0.72	0.90	V	
	$V_{HYSTERESIS\_CORE}$	23	110	mV	Note 1.

Note 1. The resulting circuit has high noise immunity. Any glitches less than 150 ns are rejected. Hysteresis is guaranteed by design.



**Figure 7-11. LVD Electrical Parameters**

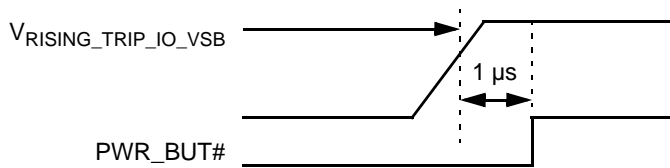
## 7.6 Skip Parameter

The Skip electrical parameters are defined in Figure 7-12 and listed in Table 7-25. In this subsection, unless otherwise noted: All timing specifications are specified under the

operating conditions listed in Table 7-3 on page 546 (i.e.,  $V_{CORE}$  and  $V_{CORE\_VSB} = All$ ;  $V_{IO}$  and  $V_{IO\_VSB} = All$ ;  $T_{CASE} = All$ ).

**Table 7-25. Skip Parameters**

Parameter	Min	Max	Units	Comment/Condition
$V_{RISING\_TRIP\_IO\_VSB}$	2.15	3.05	V	
PWR_BUT# High	0	1	$\mu s$	To enable Skip, tie PWR_BUT# to ground. To disable Skip, PWR_BUT# must be high.

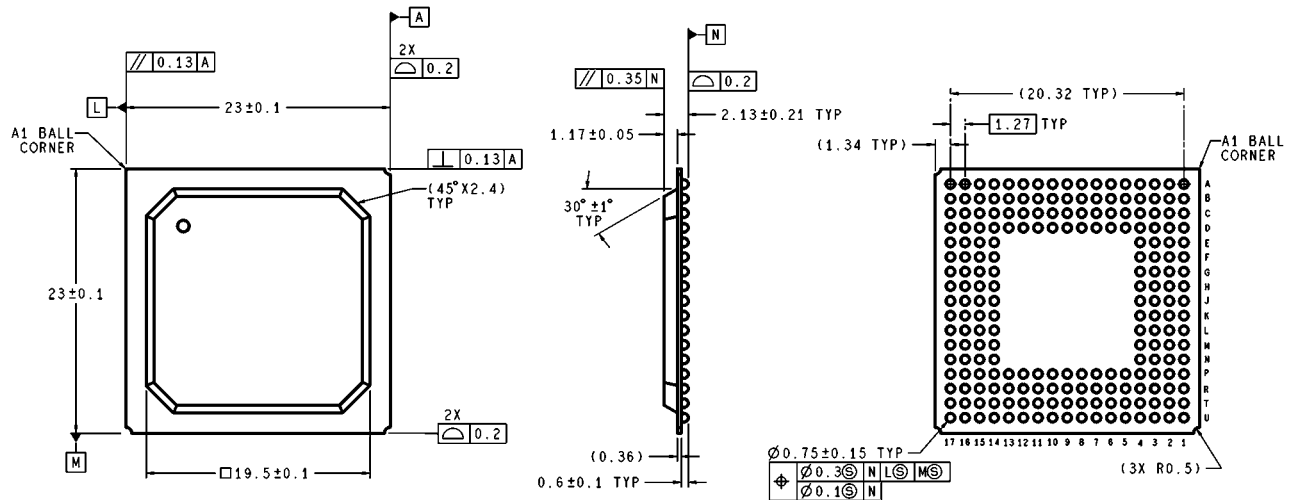


**Figure 7-12. Skip Electrical Parameters**



# Package Specifications

AMD Geode™ C5535 companion device is packaged in a 208-terminal PBGA (plastic ball grid array). Figure 8-1 provides the mechanical dimensions of the package.



DIMENSIONS ARE IN MILLIMETERS

UDC208B (Rev A)

**NOTES:** UNLESS OTHERWISE SPECIFIED.

- 1) PBGA WITH LEAD (PB):
  - a) SOLDER BALL COMPOSITION: SN 63%, PB 37%.
  - b) SOLDERING PROFILE: 220o C.
- 2) PBGA LEAD (PB) FREE:
  - a) SOLDER BALL COMPOSITION: SN 96.5%, AG 3.5%.
  - b) SOLDERING PROFILE: 260o C
- 3) DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM N.
- 4) THE MOLD SURFACE AREA MAY INCLUDE DIMPLE FOR A1 BALL CORNER IDENTIFICATION.
- 5) REFERENCE JEDEC REGISTRATION MO-151, VARIATION BAJ-2.

**Figure 8-1. 208-Terminal PBGA Package (Body Size: 23x23x2.13 mm; Pitch: 1.27 mm)**





# Support Documentation

## A.1 Order Information

Order Number	Core Voltage (V <sub>CORE</sub> )	Temperature (Degree C)	Package
CS5535-UDC	1.2V or 1.5V	0 - 85	PBGA
CS5535-UDCF			Lead Free PBGA <sup>1</sup>

Note 1. As of this publication date, a lead free package version may not be available. Check with your local AMD Sales Representative for availability. See Section 8.0 "Package Specifications" on page 575 for details on the lead free part.

## A.2 Data Book Revision History

This document is a report of the revision/creation process of the data book for the AMD Geode™ CS5535 companion device. Any revision (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table(s) below.

**Table A-1. Revision History**

Revision # (PDF Date)	Revisions / Comments
0.1 (10/23/01)	First-pass draft for team to review.
0.2 (11/18/01)	Incorporated teams edits. First-pass release to customers.
0.3 (03/04/02)	First-pass draft of complete data book for team review.
0.4 (03/12/02)	Second-pass of complete data book for team review.
0.41 (03/18/02)	Third-pass of complete data book for team review.
0.42 (03/21/02)	Fourth-pass of complete data book for team review.
0.43 (03/22/02)	Final corrections. This interim data book was released.
0.44 (06/18/02)	Corrections and updates following additional review and feedback.
0.5 (07/22/02)	Significant corrections and updates have been made to the Section 4.0 "Global Concepts and Features" and Section 7.0 "Electrical Specifications". All outstanding corrections and additions have been made to: Section 6.0 "Register Descriptions".
0.6 (12/23/02)	First-pass of functional and register chapters formatting/re-write by TME and Tech Writer. Text designated in blue denotes that the TW needs to discuss with the TME (or vice-versa) regarding consistency.
0.7 (2/21/03)	Changed Table 3-3 "Ball Assignments: Sorted Alphabetically by Signal Name" to be broken out more (i.e., all signals even if muxed). Focused on expanding/detailing Section 6.0 "Register Descriptions" and Section 4.0 "Global Concepts and Features" chapters. Text designated in blue denotes that the TW needs to discuss with the TME (or vice-versa) regarding consistency.

**Table A-1. Revision History (Continued)**

Revision # (PDF Date)	Revisions / Comments
0.8 (7/28/03)	Design team reviewed TME/Tech Writer edits in rev 0.70. Made adjustments accordingly - mostly expanding/correcting register descriptions. TME/Tech Writer rewrote many functional sections of the data book. This revision has changed considerably (no functional changes were made, just the description of the modules and registers were modified) and any old revisions should be discarded.
31506A (5/24/04)	Rev 0.9 was not released. Changed name of processor from GX2 to Geode GX processor and added engineering edits.
31506B (02/16/05)	Engineering edits. See Table A-2 for details.

**Table A-2. Edits to Current Revision**

Section	Revision
<b>All Sections</b>	<ul style="list-style-type: none"> <li>Changed <math>V_{CORE}</math> from 1.5V to 1.2V or 1.5V.</li> <li>Multi-Function General Purpose Timer was incorrectly abbreviated as MFPGT. Corrected all occurrences.</li> </ul>
<b>Section 3.0 "Signal Definitions"</b>	<ul style="list-style-type: none"> <li>Fixed typo for UART2_TX in Table 3-9 on page 49. Changed UART1 to UART2.</li> </ul>
<b>Section 5.0 "Module Functional Descriptions"</b>	
<b>Section 5.2 "GeodeLink™ PCI South Bridge"</b>	<ul style="list-style-type: none"> <li>Changed ASMI# to SMI# for Bit Position 8, Mode C in Table 5-3 on page 87 and in third bullet explaining Mode C.</li> <li>Added # signs to active low signals in Section 5.2.14 on page 87.</li> </ul>
<b>Section 6.0 "Register Descriptions"</b>	
<b>Section 6.4 "ATA-5 Controller Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Edited bit descriptions of bits [8, 1] in Section 6.4.1.4 on page 266.</li> </ul>
<b>Section 6.6 "Diverse Integration Logic Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Corrected typo in bit description for bit[22:20] in Section 6.6.2.10 on page 334. Changed UART1 to UART2.</li> </ul>
<b>Section 6.7 "Floppy Port Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Corrected register name for MSR Address 51400032h. Changed from FLPY_3F2_SHDW to FLPY_372_SHDW.</li> </ul>
<b>Section 6.13 "Direct Memory Access Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Corrected bit number for AUTO_INIT in bit descriptions table in Section 6.13.2.6 on page 425. Changed from 5 to 4.</li> </ul>
<b>Section 6.16 "GPIO Subsystem Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Corrected bit name in bit description for bit[10:8] in Section 6.16.4.1 on page 479. Changed from GPIO18 Map Select to GPIO26 Map Select.</li> </ul>
<b>Section 6.17 "Multi-Function General Purpose Timer Register Descriptions"</b>	<ul style="list-style-type: none"> <li>Section 6.17.2.4 on page 492: <ul style="list-style-type: none"> <li>Bits[14:13] - Edited bit descriptions.</li> <li>Added Write Once to descriptions of bits[11:0].</li> <li>Corrected typo for bits[9:6] - Had setting 00 twice, changed to 11.</li> </ul> </li> </ul>



[www.amd.com](http://www.amd.com)