National Semiconductor

March 2003
Revision 1.1

Geode™ SC1100
Information Appliance On a Chip

General Description

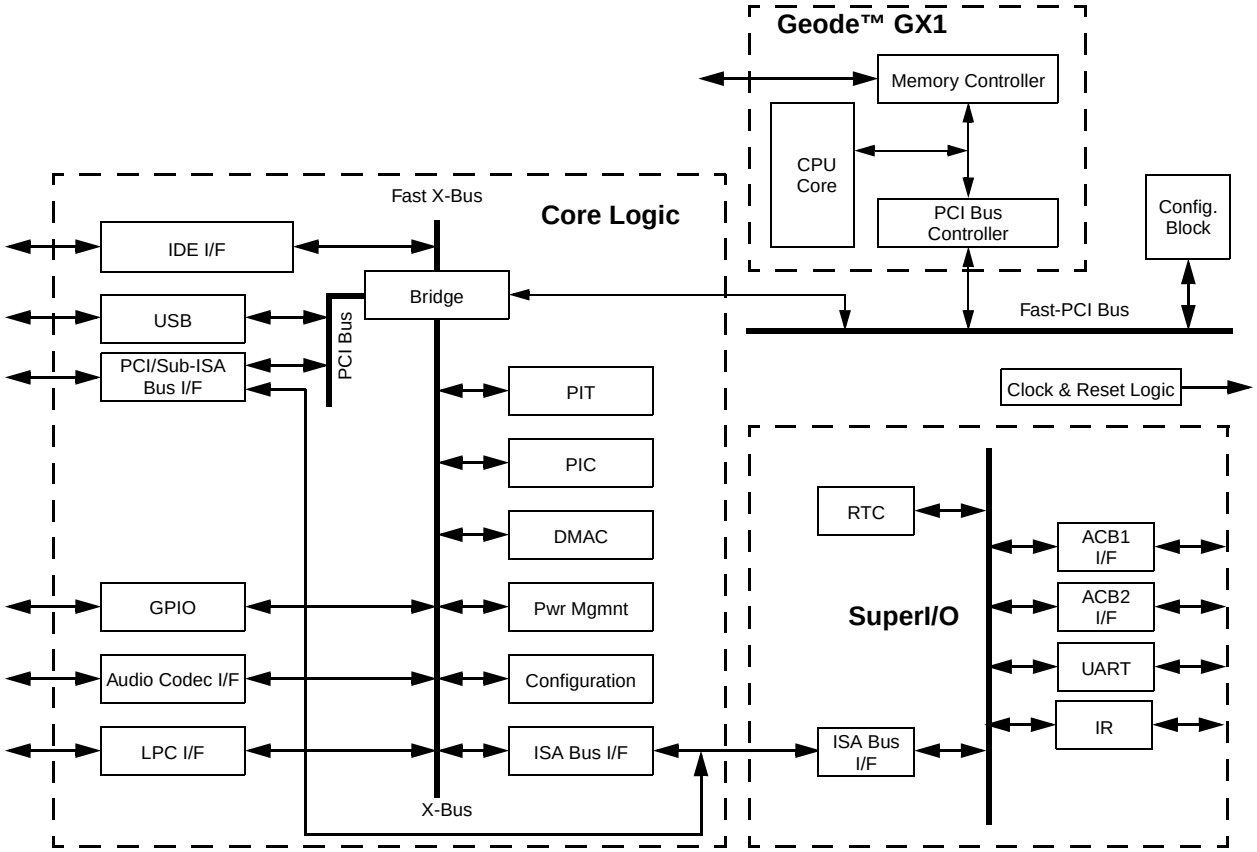
The Geode™ SC1100 device is a member of the National Semiconductor® Information Appliance On a Chip (IAOC) family of fully integrated x86 system chips. The main modules of the Geode SC1100 are:

- Geode GX1 processor module - Combines advanced CPU performance with Intel MMX support, a 64-bit synchronous DRAM (SDRAM) interface, and a PCI bus controller.
- Core Logic module - Includes PC/AT functionality, a USB interface, an IDE interface, a PCI bus interface, an LPC bus interface, Advanced Configuration Power Interface (ACPI) version 1.0 compliant power management, and an audio codec interface.
- SuperI/O module - Has a Serial Port (UART), an Infrared (IR) interface, two ACCESS.bus (ACB) interfaces, and a Real-Time Clock (RTC).

The block diagram shows the relationships between the modules.

These features, combined with the device's small form factor and low power consumption, make it ideal as the core for an advanced set-top box, consumer access device, residential gateway, thin client, or embedded system.

Block Diagram



National Semiconductor and Virtual System Architecture are registered trademarks of National Semiconductor Corporation.
Geode and VSA are trademarks of National Semiconductor Corporation.
For a complete listing of National Semiconductor trademarks, please visit www.national.com/trademarks.

© 2003 National Semiconductor Corporation

Features

General Features

- 32-bit x86 processor, up to 300 MHz, with MMX instruction set support
- Memory controller with 64-bit SDRAM interface
- PC/AT functionality
- PCI bus controller
- IDE interface, two channels
- USB, three ports, OHCI (OpenHost Controller Interface) version 1.0 compliant
- Audio, AC97/AMC97 version 2.0 compliant
- National's Virtual System Architecture® technology (VSA™) support
- Power management, ACPI (Advanced Configuration Power Interface) version 1.0 compliant
- Package:
  - 388-Terminal TEPBGA (Thermally Enhanced Plastic Ball Grid Array)

GX1 Processor Module

- CPU Core:
  - 32-bit x86 processor, up to 300 MHz, with MMX instruction set support
  - 16 KB unified L1 cache
  - Integrated Floating Point Unit (FPU)
  - Re-entrant SMM (System Management Mode) enhanced for VSA
- Memory Controller:
  - 64-bit SDRAM interface
  - 78 to 100 MHz frequency range
  - Direct interface with CPU/cache
  - Supports clock suspend and power-down/self-refresh
  - Up to 8 SDRAM devices or one DIMM/SODIMM

Core Logic Module

- Audio Codec Interface:
  - AC97/AMC97 (Rev. 2.0) codec interface
  - Legacy audio emulation using XpressAUDIO
  - Six DMA channels
- PC/AT Functionality:
  - Programmable Interrupt Controller (PIC), 8259A-equivalent
  - Programmable Interval Timer (PIT), 8254-equivalent
  - DMA Controller (DMAC), 8237-equivalent
- Power Management:
  - ACPI 1.0 compliant
  - Sx state control of three power planes
  - Cx/Sx state control of clocks and PLLs
  - Thermal event input
  - Wakeup event support:
    - Three general-purpose events
    - UART RI# signal
    - Infrared (IR) event
- General Purpose I/Os (GPIOs):
  - Six (6) dedicated GPIO signals
  - 24 multiplexed GPIO signals
- Low Pin Count (LPC) Bus Interface:
  - Specification version 1.0 compatible
- PCI Bus Interface:
  - PCI version 2.1 compliant with wakeup capability
  - 32-bit data path, up to 33 MHz
  - Glueless interface for an external PCI device
  - Supports four PCI bus master devices
  - Supports four PCI interrupts
  - Rotating priority
  - 3.3V signal support only
- Sub-ISA Bus Interface:
  - Supports up to four chip selects for external memory devices
    - Up to 16 MB addressing
    - Supports a chip select for ROM or Flash EPROM boot device, up to 16 MB
    - A chip select for one of:
      - M-Systems' DiskOnChip DOC2000 Flash file system
      - NAND EEPROM
  - Supports up to two chip selects for external I/O devices
    - 8-bit (optional 16-bit) data bus width
    - Cycle multiplexed with PCI signals
    - Is not the subtractive decode agent
- IDE Interface:
  - Two IDE channels for up to four external IDE devices
  - Supports ATA-33 synchronous DMA mode transfers, up to 33 MB/s
- Universal Serial Bus (USB):
  - USB OpenHCI 1.0 compliant
  - Three ports

SuperI/O Module

- Real-Time Clock (RTC):
  - DS1287, MC146818 and PC87911 compatible
  - Multi-century calendar
- ACCESS.bus (ACB) Interface:
  - Two ACB interface ports
- Serial Port (UART):
  - Enhanced UART
- Infrared (IR) Port
  - IrDA1.1 and 1.0 compatible
  - Sharp-IR options ASK-IR and DASK-IR
  - Consumer Remote Control supports RC-5, RC-6, NEC, RCA and RECS80
  - DMA support

Other Features

- High Resolution Timer:
  - 32-bit counter with 1 μs count interval
- Watchdog:
  - Interfaces to INTR, SMI, Reset
- Clocks:
  - Input (external crystals):
    - 32.768 KHz (internal clock oscillator)
    - 27 MHz (internal clock oscillator)
  - Output:
    - AC97 clock (24.576 MHz)
    - Memory controller clock (78 to 100 MHz)
    - PCI clock (33 MHz)
- JTAG Testability:
  - Bypass, Extest, Sample/Preload, IDcode, Clamp, HiZ
- Voltages
  - Internal logic: 1.8V (233 MHz); 2.0V (266 MHz) or TBD (300 MHz) ± 5%
  - Battery: 3V ± 20%
  - I/O: 3.3V ± 5%
  - Standby: 3.3V ± 5% and 1.8V (233 MHz); 2.0V (266 MHz) or 2.0V (300 MHz) ± 5%

Table of Contents

1.0 Architecture Overview 9

1.1 GX1 MODULE 10
1.1.1 Integer Unit 11
1.1.2 Floating Point Unit 11
1.1.3 Write-Back Cache Unit 11
1.1.4 Memory Management Unit 11
1.1.5 Internal Bus Interface Unit 11
1.1.6 Integrated Functions 12
1.1.6.1 Memory Subsystem 12
1.1.6.2 PCI Controller 12
1.1.6.3 Fast-PCI Bus 12
1.1.7 Differences from Standalone GX1 12
1.1.7.1 Device ID 12
1.1.7.2 SDRAM 12
1.1.7.3 Slave PCI Burst Length Control 13
1.1.7.4 Scratchpad Size Control 13
1.1.7.5 Disable Virtual VGA 13
1.1.7.6 Disable Display Controller 13
1.2 CORE LOGIC MODULE 14
1.2.1 Other Interfaces of the Core Logic Module 14
1.3 SUPERI/O MODULE 14
1.4 CLOCK, TIMERS, AND RESET LOGIC 14
1.4.1 Reset Logic 14
1.4.1.1 Power-On Reset 15
1.4.1.2 System Reset 15

2.0 Signal Definitions 16

2.1 BALL ASSIGNMENTS 17
2.2 STRAP OPTIONS 31
2.3 MULTIPLEXING CONFIGURATION 32
2.4 SIGNAL DESCRIPTIONS 36
2.4.1 System Interface 36
2.4.2 Memory Interface Signals 37
2.4.3 ACCESS.bus Interface Signals 38
2.4.4 PCI Bus Interface Signals 38
2.4.5 Sub-ISA Interface Signals 41
2.4.6 Low Pin Count (LPC) Bus Interface Signals 42
2.4.7 IDE Interface Signals 42
2.4.8 Universal Serial Bus (USB) Interface Signals 43
2.4.9 Serial Port (UART) and Infrared (IR) Interface Signals 43
2.4.10 AC97 Audio Interface Signals 44
2.4.11 Power Management Interface Signals 44
2.4.12 GPIO Interface Signals 45
2.4.13 Debug Monitoring Interface Signals 46
2.4.14 JTAG Interface Signals 47
2.4.15 Test and Measurement Interface Signals 47
2.4.16 Power, Ground and No Connections 48

Table of Contents (Continued)

3.0 General Configuration Block 49

3.1 CONFIGURATION BLOCK ADDRESSES 49
3.2 MULTIPLEXING, INTERRUPT SELECTION, AND BASE ADDRESS REGISTERS 50
3.3 WATCHDOG 55
3.3.1 Functional Description 55
3.3.1.1 WATCHDOG Timer 55
3.3.2 WATCHDOG Registers 56
3.3.2.1 Usage Hints 56
3.4 HIGH-RESOLUTION TIMER 57
3.4.1 Functional Description 57
3.4.2 High-Resolution Timer Registers 57
3.4.2.1 Usage Hints 57
3.5 CLOCK GENERATORS AND PLLS 59
3.5.1 27 MHz Crystal Oscillator 60
3.5.2 GX1 Module Core Clock 61
3.5.3 Internal Fast-PCI Clock 61
3.5.4 SuperI/O Clocks 62
3.5.5 Core Logic Module Clocks 62
3.5.6 Clock Registers 63

4.0 SuperI/O Module 65

4.1 FEATURES 66
4.2 MODULE ARCHITECTURE 67
4.3 CONFIGURATION STRUCTURE / ACCESS 68
4.3.1 Index-Data Register Pair 68
4.3.2 Banked Logical Device Registers 68
4.3.3 Default Configuration Setup 69
4.3.4 Address Decoding 69
4.4 STANDARD CONFIGURATION REGISTERS 70
4.4.1 SIO Control and Configuration Registers 73
4.4.2 Logical Device Control and Configuration 74
4.4.2.1 LDN 00h - Real-Time Clock 74
4.4.2.2 LDN 01h - System Wakeup Control 76
4.4.2.3 LDN 02h - Infrared Communication Port 77
4.4.2.4 LDN 05h and 06h - ACCESS.bus Ports 1 and 2 78
4.4.2.5 LDN 08h - SP Configuration 79
4.5 REAL-TIME CLOCK (RTC) 80
4.5.1 Bus Interface 80
4.5.2 RTC Clock Generation 80
4.5.2.1 Internal Oscillator 80
4.5.2.2 External Oscillator 81
4.5.3 Timing Generation 81
4.5.3.1 Timekeeping 82
4.5.3.2 Alarms 82
4.5.3.3 Power Supply 83
4.5.3.4 System Power States 84
4.5.3.5 Oscillator Activity 84
4.5.3.6 Interrupt Handling 85
4.5.3.7 Battery-Backed RAMs and Registers 85
4.5.4 RTC Registers 86
4.5.4.1 Usage Hints 91
4.5.5 RTC General-Purpose RAM Map 91

Table of Contents (Continued)

4.6 SYSTEM WAKEUP CONTROL (SWC) 92
4.6.1 Event Detection 92
4.6.2 SWC Registers 93
4.7 ACCESS.BUS INTERFACE 97
4.7.1 Data Transactions 97
4.7.2 Start and Stop Conditions 97
4.7.3 Acknowledge (ACK) Cycle 98
4.7.4 Acknowledge After Every Byte Rule 99
4.7.5 Addressing Transfer Formats 99
4.7.6 Arbitration on the Bus 99
4.7.7 Master Mode 99
4.7.7.1 Master Stop 100
4.7.8 Slave Mode 101
4.7.9 Configuration 101
4.7.10 ACB Registers 102
4.8 LEGACY FUNCTIONAL BLOCKS 105
4.8.1 UART Functionality (SP) 105
4.8.1.1 UART Mode Register Bank Overview 105
4.8.1.2 SP Register and Bit Maps for UART Functionality 105
4.8.2 IR Communications Port (IRCP) Functionality 108
4.8.2.1 IRCP Mode Register Bank Overview 108
4.8.2.2 IRCP Register and Bit Maps 109

5.0 Core Logic Module 116

5.1 FEATURE LIST 116
5.2 MODULE ARCHITECTURE AND CONFIGURATION 117
5.2.1 Fast-PCI Interface to External PCI Bus 118
5.2.1.1 Processor Mastered Cycles 118
5.2.1.2 External PCI Mastered Cycles 118
5.2.1.3 Core Logic Internal or Sub-ISA Mastered Cycles 118
5.2.1.4 External PCI Bus 118
5.2.2 PSERIAL Interface 118
5.2.3 IDE Controller 119
5.2.3.1 IDE Configuration Registers 119
5.2.3.2 PIO Mode 119
5.2.3.3 Bus Master Mode 119
5.2.3.4 UltraDMA/33 Mode 121
5.2.4 Universal Serial Bus 122
5.2.5 Sub-ISA Bus Interface 122
5.2.5.1 Sub-ISA Bus Cycles 123
5.2.5.2 Sub-ISA Support of Delayed PCI Transactions 123
5.2.5.3 Sub-ISA Bus Data Steering 124
5.2.5.4 I/O Recovery Delays 124
5.2.5.5 ISA DMA 125
5.2.5.6 ROM Interface 126
5.2.5.7 PCI and Sub-ISA Signal Cycle Multiplexing 126
5.2.6 AT Compatibility Logic 127
5.2.6.1 DMA Controller 127
5.2.6.2 Programmable Interval Timer 129
5.2.6.3 Programmable Interrupt Controller 130
5.2.7 I/O Ports 092h and 061h System Control 132
5.2.7.1 I/O Port 092h System Control 132
5.2.7.2 I/O Port 061h System Control 132
5.2.7.3 SMI Generation for NMI 132

Table of Contents (Continued)

5.2.8 Keyboard Support . . . 133
5.2.8.1 Fast Keyboard Gate Address 20 and CPU Reset . . . 133
5.2.9 Power Management Logic . . . 134
5.2.9.1 CPU States . . . 134
5.2.9.2 Sleep States . . . 134
5.2.9.3 Power Planes Control . . . 136
5.2.9.4 Power Management Events . . . 136
5.2.9.5 Usage Hints . . . 137
5.2.10 Power Management Programming . . . 137
5.2.10.1 APM Support . . . 138
5.2.10.2 CPU Power Management . . . 138
5.2.10.3 Peripheral Power Management . . . 139
5.2.10.4 Power Management Programming Summary . . . 142
5.2.11 GPIO Interface . . . 143
5.2.12 Integrated Audio . . . 143
5.2.12.1 Data Transport Hardware . . . 143
5.2.12.2 AC97 Codec Interface . . . 146
5.2.12.3 VSA Technology Support Hardware . . . 147
5.2.12.4 IRQ Configuration Registers . . . 149
5.2.13 LPC Interface . . . 149
5.2.13.1 LPC Interface Signal Definitions . . . 150
5.2.13.2 Cycle Types . . . 150
5.2.13.3 LPC Interface Support . . . 150
5.3 REGISTER DESCRIPTIONS . . . 151
5.3.1 PCI Configuration Space and Access Methods . . . 151
5.3.2 Register Summary . . . 152
5.4 CHIPSET REGISTER SPACE . . . 166
5.4.1 Bridge, GPIO, and LPC Registers - Function 0 . . . 166
5.4.1.1 GPIO Support Registers . . . 196
5.4.1.2 LPC Support Registers . . . 200
5.4.2 SMI Status and ACPI Registers - Function 1 . . . 207
5.4.2.1 SMI Status Support Registers . . . 208
5.4.2.2 ACPI Support Registers . . . 216
5.4.3 IDE Controller Registers - Function 2 . . . 225
5.4.3.1 IDE Controller Support Registers . . . 229
5.4.4 XpressAUDIO Registers - Function 3 . . . 231
5.4.4.1 XpressAUDIO Support Registers . . . 232
5.4.5 X-Bus Expansion Interface - Function 5 . . . 246
5.4.5.1 X-Bus Expansion Support Registers . . . 250
5.4.6 USB Controller Registers - PCIUSB . . . 251
5.4.7 ISA Legacy Register Space . . . 265

6.0 Debugging and Monitoring . . . 279

6.1 TESTABILITY (JTAG) . . . 279
6.1.1 Mandatory Instruction Support . . . 279
6.1.2 Optional Instruction Support . . . 279
6.1.3 JTAG Chain . . . 279

Table of Contents (Continued)

7.0 Electrical Specifications 280

7.1 GENERAL SPECIFICATIONS 280
7.1.1 Power/Ground Connections and Decoupling 280
7.1.2 Absolute Maximum Ratings 280
7.1.3 Operating Conditions 281
7.1.4 DC Current 282
7.1.4.1 Power State Parameter Definitions 282
7.1.4.2 Definition and Measurement Techniques of SC1100 Current Parameters 282
7.1.4.3 Current Consumption 283
7.1.5 Ball Capacitance and Inductance 284
7.1.6 Pull-Up and Pull-Down Resistors 285
7.2 DC CHARACTERISTICS 286
7.2.1 INAB DC Characteristics 287
7.2.2 INBTN DC Characteristics 287
7.2.3 INPCI DC Characteristics 287
7.2.4 INT DC Characteristics 287
7.2.5 INTS DC Characteristics 288
7.2.6 INTS1 DC Characteristics 288
7.2.7 INUSB DC Characteristics 289
7.2.8 ODn DC Characteristics 289
7.2.9 ODPCI DC Characteristics 289
7.2.10 Op/n DC Characteristics 289
7.2.11 OPCI DC Characteristics 290
7.2.12 OUSB DC Characteristics 290
7.2.13 TSp/n DC Characteristics 290
7.2.13.1 Exceptions 290
7.3 AC CHARACTERISTICS 291
7.3.1 Memory Interface 292
7.3.2 ACCESS.bus Interface 295
7.3.3 PCI Bus 298
7.3.3.1 Measurement and Test Conditions 301
7.3.4 Sub-ISA Interface 304
7.3.5 LPC Interface 308
7.3.6 IDE Interface Timing 309
7.3.7 Universal Serial Bus (USB) 327
7.3.8 Serial Port (UART) 331
7.3.9 Fast IR Port Timing 332
7.3.10 Audio Interface Timing (AC97) 333
7.3.11 Power Management 338
7.3.12 Power-Up Sequencing 339
7.3.13 JTAG Timing 341

8.0 Package Specifications 343

8.1 THERMAL CHARACTERISTICS 343
8.1.1 Heatsink Considerations 344
8.2 PHYSICAL DIMENSIONS 346

Appendix A Support Documentation 347

A.1 ORDER INFORMATION 347
A.2 DATASHEET REVISION HISTORY 347

1.0 Architecture Overview

As illustrated in Figure 1-1, the SC1100 contains the following modules in one integrated device:

- GX1 Module:
  - — Combines advanced CPU performance with MMX support, a 64-bit synchronous DRAM (SDRAM) interface and a PCI bus controller. Integrates GX1 silicon revision 8.1.1.
- Core Logic Module:
  - — Includes PC/AT functionality, an IDE interface, a Universal Serial Bus (USB) interface, ACPI 1.0 compliant power management, and an audio codec interface.
- SuperI/O Module:
  - — Includes a Serial Port (UART), an Infrared (IR) interface, two ACCESS.bus (ACB) interfaces, and a Real-Time Clock (RTC).

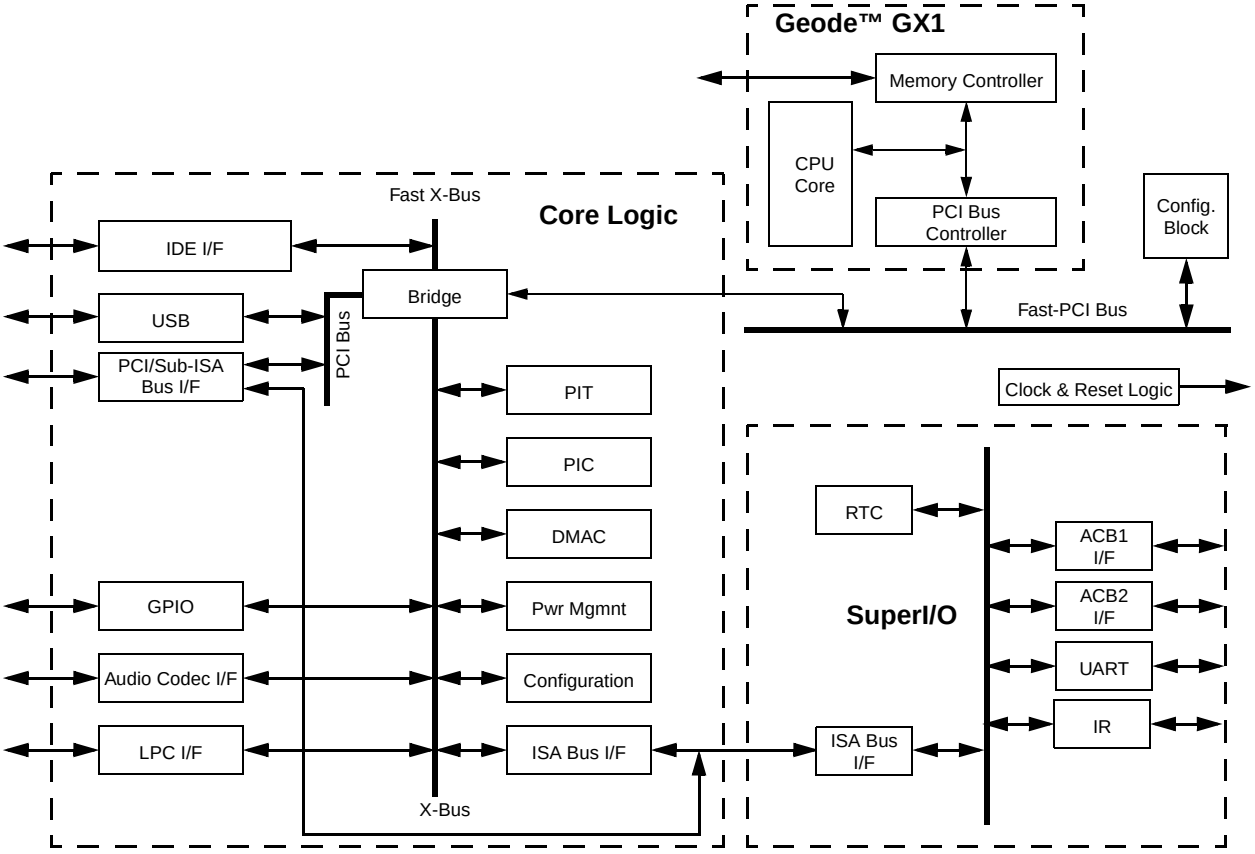


Figure 1-1. SC1100 Block Diagram

Architecture Overview (Continued)

1.1 GX1 MODULE

The GX1 module (based upon silicon revision 8.1.1) is the central module of the SC1100. For detailed information regarding the GX1 module, refer to the G*eode GX1 Processor Series datasheet* and the *Geode GX1 Processor Series Silicon Revision 8.1.1 errata*. The Geode GX1 module represents the sixth generation of x86-compatible 32-bit processors with sixth-generation features. The decoupled load/store unit allows reordering of load/store traffic to achieve higher performance. Other features include single-cycle execution, single-cycle instruction decode, 16 KB write-back cache, and clock rates up to 300 MHz. These features are made possible by the use of advanced-process technologies and pipelining.

The GX1 module has low power consumption at all clock frequencies. Where additional power savings are required, designers can make use of Suspend Mode, Stop Clock capability, and System Management Mode (SMM).

The GX1 module is divided into major functional blocks (as shown in Figure 1-2):

- Integer Unit
- Floating Point Unit (FPU)
- Write-Back Cache Unit
- Memory Management Unit (MMU)
- Internal Bus Interface Unit
- Integrated Functions

Instructions are executed in the integer unit and in the floating point unit. The cache unit stores the most recently used data and instructions and provides fast access to this information for the integer and floating point units.

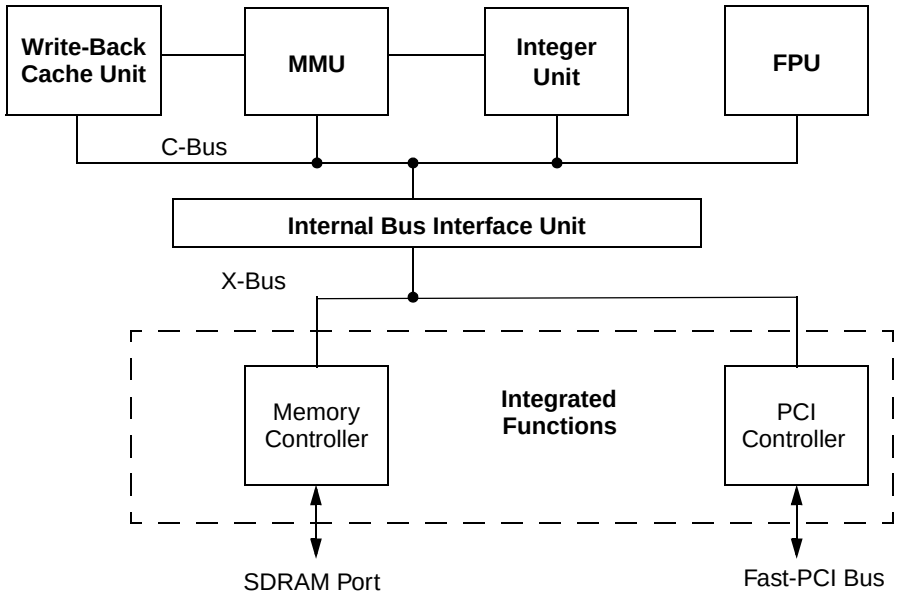


Figure 1-2. Internal Block Diagram

Architecture Overview (Continued)

1.1.1 Integer Unit

The integer unit consists of:

- Instruction Buffer
- Instruction Fetch
- Instruction Decoder and Execution

The pipelined integer unit fetches, decodes, and executes x86 instructions through the use of a five-stage integer pipeline.

The instruction fetch pipeline stage generates, from the on-chip cache, a continuous high-speed instruction stream for use by the module. Up to 128 bits of code are read during a single clock cycle.

Branch prediction logic within the prefetch unit generates a predicted target address for unconditional or conditional branch instructions. When a branch instruction is detected, the instruction fetch stage starts loading instructions at the predicted address within a single clock cycle. Up to 48 bytes of code are queued prior to the instruction decode stage.

The instruction decode stage evaluates the code stream provided by the instruction fetch stage and determines the number of bytes in each instruction and the instruction type. Instructions are processed and decoded at a maximum rate of one instruction per clock.

The address calculation function is pipelined and contains two stages, AC1 and AC2. If the instruction refers to a memory operand, AC1 calculates a linear memory address for the instruction.

The AC2 stage performs any required memory management functions, cache accesses, and register file accesses. If a floating point instruction is detected by AC2, the instruction is sent to the floating point unit for processing.

The execution stage, under control of microcode, executes instructions using the operands provided by the address calculation stage.

Write-back, the last stage of the integer unit, updates the register file within the integer unit or writes to the load/store unit within the memory management unit.

1.1.2 Floating Point Unit

The floating point unit (FPU) interfaces to the integer unit and the cache unit through a 64-bit bus. The FPU is x87-instruction-set compatible and adheres to the IEEE-754 standard. Because almost all applications that contain FPU instructions also contain integer instructions, the GX1 module's FPU achieves high performance by completing integer and FPU operations in parallel.

FPU instructions are dispatched to the pipeline within the integer unit. The address calculation stage of the pipeline checks for memory management exceptions and accesses memory operands for use by the FPU. Once the instructions and operands have been provided to the FPU, the FPU completes instruction execution independently of the integer unit.

1.1.3 Write-Back Cache Unit

The 16 KB write-back unified (data/instruction) cache is configured as four-way set associative. The cache stores up to 16 KB of code and data in 1024 cache lines.

The GX1 module provides the ability to allocate a portion of the L1 cache as a scratchpad, which is used to accelerate the Virtual Systems Architecture technology algorithms.

1.1.4 Memory Management Unit

The memory management unit (MMU) translates the linear address supplied by the integer unit into a physical address to be used by the cache unit and the internal bus interface unit. Memory management procedures are x86-compatible, adhering to standard paging mechanisms.

The MMU also contains a load/store unit that is responsible for scheduling cache and external memory accesses. The load/store unit incorporates two performance-enhancing features:

- **Load-store reordering** that gives memory reads, required by the integer unit, priority over writes to external memory.
- **Memory-read bypassing** that eliminates unnecessary memory reads by using valid data from the execution unit.

1.1.5 Internal Bus Interface Unit

The internal bus interface unit provides a bridge from the GX1 module to the integrated system functions and the Fast-PCI bus interface.

When an external memory access is required, the physical address is calculated by the memory management unit and then passed to the internal bus interface unit, which translates the cycle to an X-Bus cycle (the X-Bus is a proprietary internal bus which provides a common interface for all of the integrated functions). The X-Bus memory cycle is arbitrated between other pending X-Bus memory requests to the SDRAM controller before completing.

In addition, the internal bus interface unit provides configuration control for up to 20 different regions within system memory with separate controls for read access, write access, cacheability, and PCI access.

Architecture Overview (Continued)

1.1.6 Integrated Functions

The GX1 module integrates the following functions traditionally implemented using external devices:

- SDRAM memory controller
- PCI bridge

The module has also been enhanced to support VSA technology implementation.

1.1.6.1 Memory Subsystem

The memory controller drives a 64-bit SDRAM port directly. Up to two module banks of SDRAM are supported. Each module bank can have two or four component banks depending on the memory size and organization. The maximum configuration is two module banks with four component banks, each providing a total of 16 open banks. The maximum memory size is 512 MB.

The memory controller handles multiple requests for memory data from the GX1 module, and other sources via Fast-PCI. The memory controller contains extensive buffering logic that helps minimize contention for memory bandwidth. The memory controller cooperates with the internal bus controller to determine the cacheability of all memory references.

1.1.6.2 PCI Controller

The GX1 module incorporates a full-function PCI interface module that includes the PCI arbiter. All accesses to external I/O devices are sent over the Fast-PCI bus, although most memory accesses are serviced by the SDRAM controller. The internal bus interface unit contains address mapping logic that determines if memory accesses are targeted for the SDRAM or for the PCI bus. The PCI bus in an SC1100- based system is 3.3 volt only. Do not connect 5 volt devices on this bus.

1.1.6.3 Fast-PCI Bus

The GX1 module communicates with the Core Logic module via a Fast-PCI bus that can work at up to 66 MHz. The Fast-PCI bus is internal for the SC1100 and is connected to the Configuration Block.

1.1.7 Differences from Standalone GX1

1.1.7.1 Device ID

The SC1100's device ID is contained within the GX1 module. Software can detect the revision by reading the DIR0 and DIR1 Configuration registers (see Configuration registers in the G*eode GX1 Processor Series datasheet*). The SC1100 device errata contains the specific values.

1.1.7.2 SDRAM

The GX1 module is connected to external SDRAM DIMMs. For more information see Section 2.4.2 "Memory Interface Signals" on page 37, and the "Memory Controller" chapter in the *GX1 Processor Series datasheet*. The drive strength/slew control in the memory controller block (see Table 1-1) is different from the standalone GX1 processor.

Table 1-1. SC1100 GX1 Module System Memory Buffer Strength Control

Bit	Description
GX_BASE+Memory Offset 8400h **MC_MEM_CNTRL1 (R/W)** **Reset Value: 248C0040h** Width: DWORD	
31:30	**MD[63:0] Drive Strength:** 11 is strongest, 00 is weakest.
28:27	**MA[12:0] and BA[1:0] Drive Strength:** 11 is strongest, 00 is weakest.
25:24	**RASA#, CASA#, WEA#, CS[1:0]#, CKEA, DQM[7:0] Drive Strength:** 11 is strongest, 00 is weakest.
3	**X-Bus Round Robin.** Must be written with 1.
GX_BASE+Memory Offset 8404h **MC_MEM_CNTRL2 (R/W)** **Reset Value: 00000801h** Width: DWORD	
13:12	**SCLK High Drive/Slew Control:** Controls the high drive and slew rate of SDCLK[3:0] and SDCLK_OUT. 11 is strongest, 00 is weakest.

Architecture Overview (Continued)

1.1.7.3 Slave PCI Burst Length Control

The name of this bit differs from that described in the *GX1 Processor Series datasheet*. There, it is called SDBE. Otherwise, the functionality is the same as the standalone GX1 processor, See Table 1-2.

1.1.7.4 Scratchpad Size Control

The *GX1 Processor Series datasheet* allows additional sizes of 3 KB and 4 KB, which are disallowed in the SC1100, see Table 1-3.

1.1.7.5 Disable Virtual VGA

The registers detailed in Table 1-4 must be set as indicated in order for the SC1100 to function properly.

1.1.7.6 Disable Display Controller

In order for the SC1100 to function correctly, the Display Controller Configuration and Status registers (GX_BASE+Offset 8300h-830Fh) must all be set with 0s.

Table 1-2. Slave PCI Burst Length Control

Bit	Description
Index 41h Width: Byte	**PCI Control Function 2 Register (R/W)** — **Reset Value: 96h**
1	**Slave Disconnect Boundary Enable (SDBE1):** GX1 as a PCI slave: Works in conjunction with bit 0 of the MCR register. For details see Section 3.2 on page 50. **Note:** When Slave Disconnect Boundary is disabled for Write, the cache should use Write Through Mode instead of Write Back Mode. The Write Through Mode implies some overall performance degradation since all Writes go to Memory. If the Write back Mode is used in this case, the cache coherency cannot be guaranteed.

Table 1-3. Scratchpad Size Control

Bit	Description
Index B8h Width: Byte	**Graphics Control Register (R/W)** — **Reset Value: 00h**
3:2	**Scratchpad Size:** Specifies the size of the scratchpad cache. Either a 0 KB or a 2 KB Scratchpad Size must be chosen. For details, see the GX1 Processor Series datasheet.

Table 1-4. Disable Virtual VGA

Bit	Description
Index 20h Width: Byte	**PCR0: Performance Control 0 Register (R/W)** — **Reset Value: 07h**
5	**VGA Memory Write SMI Generation (VGAMWSI).** Must be set to 0.
Index B9h Width: Byte	**VGACTL Register (R/W)** — **Reset Value: 00h**
2	SMI generation for VGA memory range B8000h to BFFFFh. Must be set to 0.
1	SMI generation for VGA memory range B0000h to B7FFFh. Must be set to 0.
0	SMI generation for VGA memory range A0000h to AFFFFh. Must be set to 0.
Index BAh-BDh Width: DWORD	**VGAM Register (R/W)** — **Reset Value: xxxxxxxxh**
31:0	SMI generation for address range A0000h to AFFFFh. Must be set to all 0s.
Gx Based + Offset 8004h - 8007h Width: DWORD	**BC_XMAP_1 Register (R/W)** — **Reset Value: 00000000h**
28	Graphics Enable for B8 Region (GEB8). Must be set to 0.
20	Graphics Enable for B0 Region (GEB0). Must be set to 0.
15	SMID: All I/O accesses for address range 3D0h to 3DFh generate an SMI. Must be set to 0.
14	SMIC: All I/O accesses for address range 3C0h to 3CFh generate an SMI. Must be set to 0.
13	SMIB: All I/O accesses for address range 3B0h to 3BFh generate an SMI. Must be set to 0.
4	Graphics Enable for A Region (GEA). Must be set to 0.

1.2 CORE LOGIC MODULE

The Core Logic module is described in detail in Section 5.0 "Core Logic Module" on page 116.

The Core Logic module is connected to the Fast-PCI bus. It uses signal AD28 as the IDSEL for all PCI configuration functions except for USB which uses AD29.

1.2.1 Other Interfaces of the Core Logic Module

All the following interfaces of the Core Logic module are implemented via external pins of the SC1100. Each interface is listed below with a reference to the descriptions of the relevant pins.

- IDE: See Section 2.4.7 "IDE Interface Signals" on page 42.
- AC97: See Section 2.4.10 "AC97 Audio Interface Signals" on page 44.
- PCI: See Section 2.4.4 "PCI Bus Interface Signals" on page 38.
- USB: See Section 2.4.8 "Universal Serial Bus (USB) Interface Signals" on page 43. The USB function uses signal AD29 as the IDSEL for PCI configuration.
- LPC: See Section 2.4.6 "Low Pin Count (LPC) Bus Interface Signals" on page 42.
- Sub-ISA: See Section 2.4.5 "Sub-ISA Interface Signals" on page 41, Section 5.2.5 "Sub-ISA Bus Interface" on page 122, and Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50.
- GPIO: See Section 2.4.12 "GPIO Interface Signals" on page 45.
- More detailed information about each of these interfaces is provided in Section 5.2 "Module Architecture and Configuration" on page 117.
- Super/IO Block Interfaces: See Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50, Section 2.4.3 "ACCESS.bus Interface Signals" on page 38, Section 2.4.9 "Serial Port (UART) and Infrared (IR) Interface Signals" on page 43.

The Core Logic module interface to the GX1 module consists of seven miscellaneous connections, and the Fast-PCI bus interface signals. Note that the PC/AT legacy pins NMI, WM_RST, and A20M are all virtual functions executed in SMM (System Management Mode) by the BIOS.

- PSERIAL is a one-way serial bus from the GX1 to the Core Logic module used to communicate power management states.
- IRQ13 is an input from the processor indicating that a floating point error was detected and that INTR should be asserted.
- INTR is the level output from the integrated 8259A PICs and is asserted if an unmasked interrupt request (IRQn) is sampled active.
- SMI# is a level-sensitive interrupt to the GX1 that can be configured to assert on a number of different system events. After an SMI# assertion, SMM is entered and program execution begins at the base of the SMM address space. Once asserted, SMI# remains active until the SMI source is cleared.
- SUSP# and SUSPA# are handshake pins for implementing GX1 module Clock Stop and clock throttling.
- CPU_RST resets the GX1 module and is asserted for approximately 100 µs after the negation of POR#.
- Fast-PCI bus interface signals.

1.3 SUPERI/O MODULE

The SuperI/O (SIO) module is a member of National Semiconductor's SuperI/O family of integrated PC peripherals. It is a PC98 and ACPI compliant SIO that offers a single-cell solution to the most commonly used ISA peripherals.

The SIO module incorporates: a Serial Port, an Infrared Communication Port that supports FIR, MIR, HP-SIR, Sharp-IR, and Consumer Electronics-IR, two ACCESS.bus Interface (ACB) ports, System Wakeup Control (SWC), and a Real-Time Clock (RTC) that provides RTC timekeeping.

1.4 CLOCK, TIMERS, AND RESET LOGIC

In addition to the three main modules (i.e., GX1, Core Logic, and SIO) that make up the SC1100, the following blocks of logic have also been integrated into the SC1100:

- Clock Generators as described in Section 3.5 "Clock Generators and PLLs" on page 59.
- Configuration Registers as described in Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50.
- A WATCHDOG timer as described in Section 3.3 "WATCHDOG" on page 55.
- A High Resolution timer as described in Section 3.4 "High-Resolution Timer" on page 57.

1.4.1 Reset Logic

This section provides a description of the reset flow of the SC1100.

Architecture Overview (Continued)

1.4.1.1 Power-On Reset

Power-on reset is triggered by assertion of the POR# signal. Upon power-on reset, the following things happen:

- Strap pins are sampled.
- PLL4, FMUL3, PLL5 are reset, disabling their output. When the POR# signal is negated, FMUL3 performs coarse locking of clocks, after which each FMUL outputs its clock. FMUL3 is the last clock generator to output a clock. See Section 3.5 "Clock Generators and PLLs" on page 59.
- Certain WATCHDOG and High Resolution timer register bits are cleared.

1.4.1.2 System Reset

System reset causes signal PCIRST# to be issued, thus triggering reset of all PCI and LPC agents. A system reset is triggered by any of the following events:

- Power-on, as indicated by POR# signal assertion.
- A WATCHDOG reset event (see Section 3.3.2 "WATCHDOG Registers" on page 56).
- Software initiated system reset.

2.0 Signal Definitions

This section defines the signals and describes the external interface of the SC1100. Figure 2-1 shows the signals organized by their functional groups. Where signals are multiplexed, the default 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., cycle multiplexed).

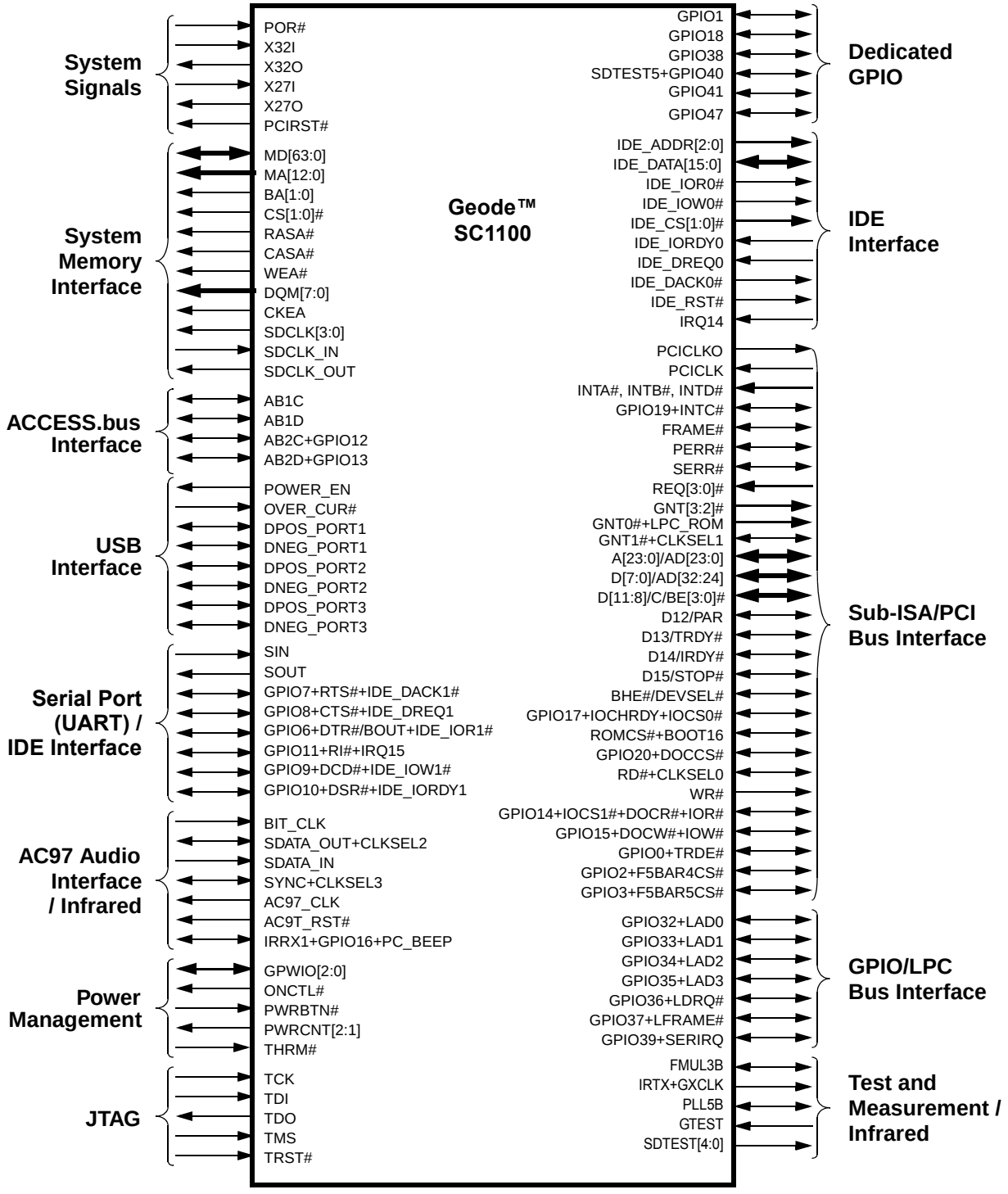


Figure 2-1. Signal Groups

Signal Definitions (Continued)

The remaining subsections of this chapter describe:

- Section 2.1 "Ball Assignments": Provides a ball assignment diagram and tables listing the signals sorted according to ball number and alphabetically by signal name.
- Section 2.2 "Strap Options": Several balls are read at power-up that set up the state of the SC1100. This section provides details regarding those balls.
- Section 2.3 "Multiplexing Configuration": Lists multiplexing options and their configurations.
- Section 2.4 "Signal Descriptions": Detailed descriptions of each signal according to functional group.

2.1 BALL ASSIGNMENTS

The SC1100 is configurable. Strap options and register programming are used to set various modes of operation and specific signals on specific balls. This section describes which signals are available on which balls and provides configuration information:

- Figure 2-2 "388-Terminal Ball Assignment Diagram (Top View)" on page 18: Illustration of ball assignment.
- Table 2-2 "Ball Assignment - Sorted by Ball Number" on page 19: Lists signals according to ball number. Power Rail, Signal Type, Buffer Type and, where relevant, Pull-Up or Pull-Down resistors are indicated for each ball in this table. For multiplexed balls, the necessary configuration for each signal is listed as well.
- Table 2-3 "Ball Assignment - Sorted Alphabetically by Signal Name" on page 27: Quick reference list sorted alphabetically listing all signal names.

The tables in this chapter use several common abbreviations. Table 2-1 lists the mnemonics and their meanings.

Notes:

1) For each GPIO signal, there is an optional pull-up resistor on the relevant ball. After system reset, the pull-up is present.

   This pull-up resistor can be disabled via registers in the Core Logic module. The configuration is without regard to the selected ball function (except for GPIO12, GPIO13, and GPIO16). Alternate functions for GPIO12, GPIO13, and GPIO16 control pull-up resistors.

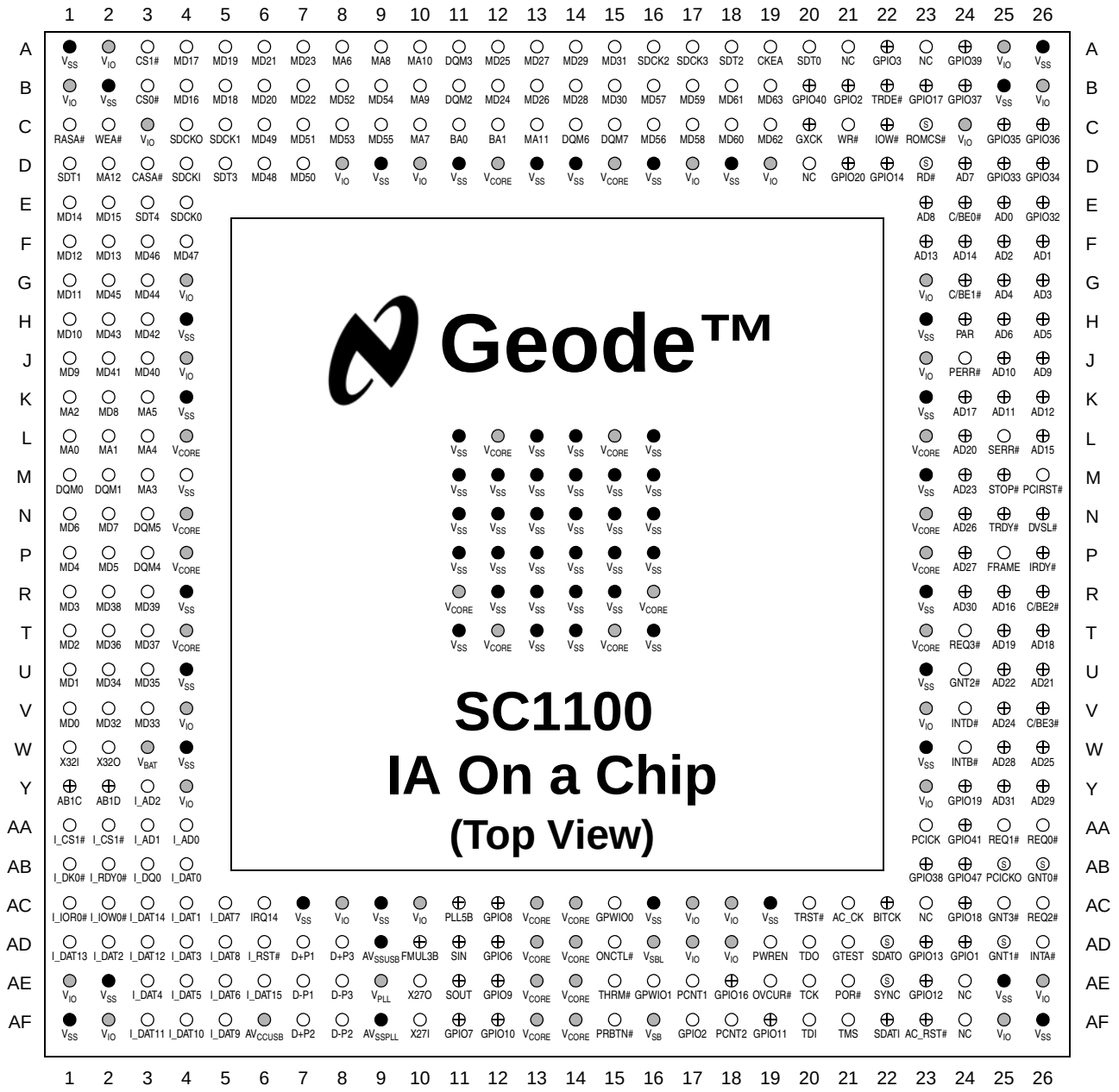
   For more information, see Section 5.4.1 "Bridge, GPIO, and LPC Registers - Function 0" on page 166.

2) Configuration settings listed in Table 2-2 are with regard to the Pin Multiplexing Register (PMR). See Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50 for a detailed description of this register.

Table 2-1. Signal Definitions Legend

Mnemonic	Definition
A	Analog
AV_{SS}	Ground ball: Analog
AV_{DD}	Power ball: Analog
GCB	General Configuration Block registers. Refer to Section 3.0 "General Configuration Block" on page 49. Location of the General Configuration Block cannot be determined by software. See *SC1100 Information Appliance On a Chip device errata.*
I	Input ball
I/O	Bidirectional ball
MCR[x]	Miscellaneous Configuration Register Bit x: A register, located in the GCB. Refer to Section 3.1 "Configuration Block Addresses" on page 49 for further details.
O	Output ball
OD	Open-drain
PD	Pull-down
PMR[x]	Pin Multiplexing Register Bit x: A register, located in the GCB, used to configure balls with multiple functions. Refer to Section 3.1 "Configuration Block Addresses" on page 49 for further details.
PU	Pull-up
TS	TRI-STATE
V_{CORE}	Power ball: 1.8V or 2.0V (speed grade dependent)
V_{IO}	Power ball: 3.3V
V_{SS}	Ground ball
#	The # symbol in a signal name indicates that the active or asserted state occurs when the signal is at a low voltage level. Otherwise, the signal is asserted when at a high voltage level.
/	A / in a signal name indicates both functions are always enabled (i.e., cycle multiplexed).
+	A + in 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.

Signal Definitions (Continued)



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

- ● = GND Ball
- ◉ = PWR Ball
- Ⓢ = Strap Option Ball
- ⊕ = Multiplexed Ball

Figure 2-2. 388-Terminal Ball Assignment Diagram (Top View)

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
A1	V_{SS}	GND	---	---	---
A2	V_{IO}	PWR	---	---	---
A3	CS1#	O	$O_{2/5}$	V_{IO}	---
A4[2]	MD17	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A5[2]	MD19	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A6[2]	MD21	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A7[2]	MD23	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A8	MA6	O	$O_{2/5}$	V_{IO}	---
A9	MA8	O	$O_{2/5}$	V_{IO}	---
A10	MA10	O	$O_{2/5}$	V_{IO}	---
A11	DQM3	O	$O_{2/5}$	V_{IO}	---
A12[2]	MD25	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A13[2]	MD27	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A14[2]	MD29	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A15[2]	MD31	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
A16	SDCLK2	O	$O_{2/5}$	V_{IO}	---
A17	SDCLK3	O	$O_{2/5}$	V_{IO}	---
A18	SDTEST2	O	$O_{2/5}$	V_{IO}	---
A19	CKEA	O	$O_{2/5}$	V_{IO}	---
A20	SDTEST0	O	$O_{2/5}$	V_{IO}	---
A21	NC	---	---	V_{IO}	---
A22	GPIO3	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[3] = 0
	F5BAR5CS#	O	$O_{3/5}$		PMR[3] = 1
A23	NC	---	---	V_{IO}	---
A24	GPIO39	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[16] = 0
	SERIRQ	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[16] = 1
A25	V_{IO}	PWR	---	---	---
A26	V_{SS}	GND	---	---	---
B1	V_{IO}	PWR	---	---	---
B2	V_{SS}	GND	---	---	---
B3	CS0#	O	$O_{2/5}$	V_{IO}	---
B4[2]	MD16	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B5[2]	MD18	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B6[2]	MD20	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B7[2]	MD22	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B8[2]	MD52	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B9[2]	MD54	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B10	MA9	O	$O_{2/5}$	V_{IO}	---
B11	DQM2	O	$O_{2/5}$	V_{IO}	---
B12[2]	MD24	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B13[2]	MD26	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B14[2]	MD28	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B15[2]	MD30	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B16[2]	MD57	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B17[2]	MD59	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B18[2]	MD61	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B19[2]	MD63	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
B20	GPIO40	I/O ($PU_{22.5}$)	IN_{TS}, $O_{2/5}$	V_{IO}	PMR[28] = 0 and PMR[27] = 0 and FPCI_MON = 0
	SDTEST5	O ($PU_{22.5}$)	$O_{2/5}$		PMR[28] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_AD6	O ($PU_{22.5}$)	$O_{2/5}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[28])
B21	GPIO2	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[1] = 0
	F5BAR4CS#	O	$O_{3/5}$		PMR[1] = 1
B22	TRDE#	O	$O_{3/5}$	V_{IO}	PMR[12] = 0
	GPIO0	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$		PMR[12] = 1
B23	GPIO17	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[5] = 0 and PMR[9] = 0
	IOCS0#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[5] = 1 and PMR[9] = 0
	IOCHRDY	I	IN_{TS}		PMR[5] = 1 and PMR[9] = 1
B24	GPIO37	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LFRAME#	O	O_{PCI}		PMR[14] = 1
B25	V_{SS}	GND	---	---	---
B26	V_{IO}	PWR	---	---	---
C1	RASA#	O	$O_{2/5}$	V_{IO}	---
C2	WEA#	O	$O_{2/5}$	V_{IO}	---
C3	V_{IO}	PWR	---	---	---
C4	SDCLK_OUT	O	$O_{2/5}$	V_{IO}	---
C5	SDCLK1	O	$O_{2/5}$	V_{IO}	---
C6[2]	MD49	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C7[2]	MD51	I/O	IN_T, $TS_{2/5}$	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
C8[2]	MD53	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C9[2]	MD55	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C10	MA7	O	$O_{2/5}$	V_{IO}	---
C11	BA0	O	$O_{2/5}$	V_{IO}	---
C12	BA1	O	$O_{2/5}$	V_{IO}	---
C13	MA11	O	$O_{2/5}$	V_{IO}	---
C14	DQM6	O	$O_{2/5}$	V_{IO}	---
C15	DQM7	O	$O_{2/5}$	V_{IO}	---
C16[2]	MD56	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C17[2]	MD58	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C18[2]	MD60	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C19[2]	MD62	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
C20	GXCLK	O	$O_{2/5}$	V_{IO}	PMR[29] = 0 and PMR[6] = 1
	IRTX	O	$O_{2/5}$		PMR[29] = 0 and PMR[6] = 0
	TEST3	O	$O_{2/5}$		PMR[29] = 1 and PMR[6] = 1
C21	WR#	O	$O_{3/5}$	V_{IO}	---
C22	IOW#	O	$O_{3/5}$	V_{IO}	PMR[21] = x and PMR[2] = 0
	DOCW#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
	GPIO15	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$		PMR[21] = 0 and PMR[2] = 1
C23	ROMCS#	O	$O_{3/5}$	V_{IO}	---
	BOOT16	I (PD_{100})	IN_{TS}	V_{IO}	Strap
C24	V_{IO}	PWR	---	---	---
C25	GPIO35	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LAD3	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14] = 1
C26	GPIO36	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LDRQ#	I ($PU_{22.5}$)	IN_{PCI}		PMR[14] = 1
D1	SDTEST1	O	$O_{2/5}$	V_{IO}	---
D2	MA12	O	$O_{2/5}$	V_{IO}	---
D3	CASA#	O	$O_{2/5}$	V_{IO}	---
D4	SDCLK_IN	I	IN_T	V_{IO}	---
D5	SDTEST3	O	$O_{2/5}$	V_{IO}	---
D6[2]	MD48	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
D7[2]	MD50	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
D8	V_{IO}	PWR	---	---	---
D9	V_{SS}	GND	---	---	---
D10	V_{IO}	PWR	---	---	---
D11	V_{SS}	GND	---	---	---
D12	V_{CORE}	PWR	---	---	---
D13	V_{SS}	GND	---	---	---
D14	V_{SS}	GND	---	---	---
D15	V_{CORE}	PWR	---	---	---
D16	V_{SS}	GND	---	---	---
D17	V_{IO}	PWR	---	---	---
D18	V_{SS}	GND	---	---	---
D19	V_{IO}	PWR	---	---	---
D20	NC	---	---	---	---
D21	GPIO20	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[7] = 0
	DOCCS#	O	$O_{3/5}$		PMR[7] = 1
D22	GPIO14	I/O $PU_{22.5)}$	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[21] = 0 and PMR[2] = 0
	IOCS1#	O ($PU_{22.5}$)	$O_{3/5}$		PMR[21] = 0 and PMR[2] = 1
	IOR#	O	$O_{3/5}$		PMR[21] = 1 and PMR[2] = 0
	DOCR#	O	$O_{3/5}$		PMR[21] = 1 and PMR[2] = 1
D23	RD#	O	$O_{3/5}$	V_{IO}	---
	CLKSEL0	I (PD_{100})	IN_{TS}		Strap
D24	AD7	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A7	O	O_{PCI}		
D25	GPIO33	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LAD1	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14] = 1
D26	GPIO34	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LAD2	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14] = 1
E1[2]	MD14	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
E2[2]	MD15	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
E3	SDTEST4	O	$O_{2/5}$	V_{IO}	---
E4	SDCLK0	O	$O_{2/5}$	V_{IO}	---
E23	AD8	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A8	O	O_{PCI}		
E24	C/BE0#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D8	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
E25	AD0	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A0	O	O_{PCI}		

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
E26	GPIO32	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[14] = 0
	LAD0	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		PMR[14] = 1
F1[2]	MD12	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
F2[2]	MD13	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
F3[2]	MD46	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
F4[2]	MD47	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
F23	AD13	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A13	O	O_{PCI}		
F24	AD14	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A14	O	O_{PCI}		
F25	AD2	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A2	O	O_{PCI}		
F26	AD1	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A1	O	O_{PCI}		
G1[2]	MD11	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
G2[2]	MD45	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
G3[2]	MD44	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
G4	V_{IO}	PWR	---	---	---
G23	V_{IO}	PWR	---	---	---
G24	C/BE1#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D9	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
G25	AD4	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A4	O	O_{PCI}		
G26	AD3	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A3	O	O_{PCI}		
H1[2]	MD10	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
H2[2]	MD43	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
H3[2]	MD42	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
H4	V_{SS}	GND	---	---	---
H23	V_{SS}	GND	---	---	---
H24	PAR	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D12	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
H25	AD6	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A6	O	O_{PCI}		
H26	AD5	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A5	O	O_{PCI}		
J1[2]	MD9	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
J2[2]	MD41	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
J3[2]	MD40	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
J4	V_{IO}	PWR	---	---	---
J23	V_{IO}	PWR	---	---	---
J24	PERR#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
J25	AD10	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A10	O	O_{PCI}		
J26	AD9	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A9	O	O_{PCI}		
K1	MA2	O	$O_{2/5}$	V_{IO}	---
K2[2]	MD8	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
K3	MA5	O	$O_{2/5}$	V_{IO}	---
K4	V_{SS}	GND	---	---	---
K23	V_{SS}	GND	---	---	---
K24	AD17	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A17	O	O_{PCI}		
K25	AD11	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A11	O	O_{PCI}		
K26	AD12	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A12	O	O_{PCI}		
L1	MA0	O	$O_{2/5}$	V_{IO}	---
L2	MA1	O	$O_{2/5}$	V_{IO}	---
L3	MA4	O	$O_{2/5}$	V_{IO}	---
L4	V_{CORE}	PWR	---	---	---
L11	V_{SS}	GND	---	---	---
L12	V_{CORE}	PWR	---	---	---
L13	V_{SS}	GND	---	---	---
L14	V_{SS}	GND	---	---	---
L15	V_{CORE}	PWR	---	---	---
L16	V_{SS}	GND	---	---	---

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
L23	V_{CORE}	PWR	---	---	---
L24	AD20	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A20	O	O_{PCI}		
L25	SERR#	I/O ($PU_{22.5}$)	IN_{PCI}, OD_{PCI}	V_{IO}	---
L26	AD15	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A15	O	O_{PCI}		
M1	DQM0	O	$O_{2/5}$	V_{IO}	---
M2	DQM1	O	$O_{2/5}$	V_{IO}	---
M3	MA3	O	$O_{2/5}$	V_{IO}	---
M4	V_{SS}	GND	---	---	---
M11	V_{SS}	GND	---	---	---
M12	V_{SS}	GND	---	---	---
M13	V_{SS}	GND	---	---	---
M14	V_{SS}	GND	---	---	---
M15	V_{SS}	GND	---	---	---
M16	V_{SS}	GND	---	---	---
M23	V_{SS}	GND	---	---	---
M24	AD23	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A23	O	O_{PCI}		
M25	STOP#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D15	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
M26	PCIRST#	O	O_{PCI}	V_{IO}	---
N1[2]	MD6	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
N2[2]	MD7	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
N3	DQM5	O	$O_{2/5}$	V_{IO}	---
N4	V_{CORE}	PWR	---	---	---
N11	V_{SS}	GND	---	---	---
N12	V_{SS}	GND	---	---	---
N13	V_{SS}	GND	---	---	---
N14	V_{SS}	GND	---	---	---
N15	V_{SS}	GND	---	---	---
N16	V_{SS}	GND	---	---	---
N23	V_{CORE}	PWR	---	---	---
N24	AD26	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D2	I/O	IN_{PCI}, O_{PCI}		
N25	TRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D13	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
N26	DEVSEL#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	BHE#	O ($PU_{22.5}$)	O_{PCI}		
P1[2]	MD4	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
P2[2]	MD5	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
P3	DQM4	O	$O_{2/5}$	V_{IO}	---
P4	V_{CORE}	PWR	---	---	---
P11	V_{SS}	GND	---	---	---
P12	V_{SS}	GND	---	---	---
P13	V_{SS}	GND	---	---	---
P14	V_{SS}	GND	---	---	---
P15	V_{SS}	GND	---	---	---
P16	V_{SS}	GND	---	---	---
P23	V_{CORE}	PWR	---	---	---
P24	AD27	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D3	I/O	IN_{PCI}, O_{PCI}		
P25	FRAME#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	---
P26	IRDY#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D14	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
R1[2]	MD3	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
R2[2]	MD38	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
R3[2]	MD39	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
R4	V_{SS}	GND	---	---	---
R11	V_{CORE}	PWR	---	---	---
R12	V_{SS}	GND	---	---	---
R13	V_{SS}	GND	---	---	---
R14	V_{SS}	GND	---	---	---
R15	V_{SS}	GND	---	---	---
R16	V_{CORE}	PWR	---	---	---
R23	V_{SS}	GND	---	---	---
R24	AD30	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D6	I/O	IN_{PCI}, O_{PCI}		
R25	AD16	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A16	O	O_{PCI}		
R26	C/BE2#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D10	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
T1[2]	MD2	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
T2[2]	MD36	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
T3[2]	MD37	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
T4	V_{CORE}	PWR	---	---	---
T11	V_{SS}	GND	---	---	---
T12	V_{CORE}	PWR	---	---	---
T13	V_{SS}	GND	---	---	---
T14	V_{SS}	GND	---	---	---
T15	V_{CORE}	PWR	---	---	---
T16	V_{SS}	GND	---	---	---
T23	V_{CORE}	PWR	---	---	---
T24	REQ3#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
T25	AD19	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A19	O	O_{PCI}		
T26	AD18	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A18	O	O_{PCI}		
U1[2]	MD1	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
U2[2]	MD34	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
U3[2]	MD35	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
U4	V_{SS}	GND	---	---	---
U23	V_{SS}	GND	---	---	---
U24	GNT2#	O	O_{PCI}	V_{IO}	---
U25	AD22	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A22	O	O_{PCI}		
U26	AD21	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	A21	O	O_{PCI}		
V1[2]	MD0	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V2[2]	MD32	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V3[2]	MD33	I/O	IN_T, $TS_{2/5}$	V_{IO}	---
V4	V_{IO}	PWR	---	---	---
V23	V_{IO}	PWR	---	---	---
V24	INTD#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
V25	AD24	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D0	I/O	IN_{PCI}, O_{PCI}		
V26	C/BE3#	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D11	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}		
W1	X32I	I	WIRE	V_{BAT}	---
W2	X32O	O	WIRE	V_{BAT}	---
W3	V_{BAT}	PWR	---	---	---
W4	V_{SS}	GND	---	---	---
W23	V_{SS}	GND	---	---	---
W24	INTB#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
W25	AD28	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D4	I/O	IN_{PCI}, O_{PCI}		
W26	AD25	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D1	I/O	IN_{PCI}, O_{PCI}		
Y1	AB1C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_AD1	O	$O_{2/8}$		PMR[27] = 1 or FPCI_MON = 1
Y2	AB1D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_AD2	O	$O_{2/8}$		PMR[27] = 1 or FPCI_MON = 1
Y3	IDE_ADDR2	O	$O_{1/4}$	V_{IO}	---
Y4	V_{IO}	PWR	---	---	---
Y23	V_{IO}	PWR	---	---	---
Y24	GPIO19	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[4] = 0
	INTC#	I ($PU_{22.5}$)	IN_{PCI}		PMR[4] = 1
Y25	AD31	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D7	I/O	IN_{PCI}, O_{PCI}		
Y26	AD29	I/O	IN_{PCI}, O_{PCI}	V_{IO}	Cycle Multiplexed
	D5	I/O	IN_{PCI}, O_{PCI}		
AA1	IDE_CS1#	O	$O_{1/4}$	V_{IO}	---
AA2	IDE_CS0#	O	$O_{1/4}$	V_{IO}	---
AA3	IDE_ADDR1	O	$O_{1/4}$	V_{IO}	---
AA4	IDE_ADDR0	O	$O_{1/4}$	V_{IO}	---
AA23	PCICLK	I	IN_T	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AA24	GPIO41	I/O ($PU_{22.5}$)	IN_{TS}, $O_{2/5}$	V_{IO}	PMR[29] = 0 and PMR[27] = 0 and FPCI_MON = 0
	TEST0	O ($PU_{22.5}$)	$O_{2/5}$		PMR[29] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_C/BE0#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[29])
AA25	REQ1#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AA26	REQ0#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AB1	IDE_DACK0#	O	$O_{1/4}$	V_{IO}	---
AB2	IDE_IORDY0	I	IN_{TS1}	V_{IO}	---
AB3	IDE_DREQ0	I	IN_{TS1}	V_{IO}	---
AB4	IDE_DATA0	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AB23	GPIO38	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_AD5	O	O_{PCI}		PMR[27] = 1 or FPCI_MON = 1
AB24	GPIO47	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_AD7	O	$O_{8/8}$		PMR[27] = 1 or FPCI_MON = 1
AB25	PCICLKO	O	O_{PCI}	V_{IO}	---
	FPCI_MON	I (PD_{100})	IN_{PCI}		Strap
AB26	GNT0#	O	O_{PCI}	V_{IO}	---
	LPC_ROM	I (PD_{100})	IN_{PCI}		Strap
AC1	IDE_IOR0#	O	$O_{1/4}$	V_{IO}	---
AC2	IDE_IOW0#	O	$O_{1/4}$	V_{IO}	---
AC3	IDE_DATA14	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AC4	IDE_DATA1	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AC5	IDE_DATA7	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AC6	IRQ14	I	IN_{TS1}	V_{IO}	---
AC7	V_{SS}	GND	---	---	---
AC8	V_{IO}	PWR	---	---	---
AC9	V_{SS}	GND	---	---	---
AC10	V_{IO}	PWR	---	---	---
AC11	PLL5B	I/O	IN_T, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST2	O	$O_{2/5}$		PMR[29] = 1
AC12	GPIO8	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	CTS#	I	IN_{TS}		PMR[17] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	IDE_DREQ1	I ($PU_{22.5}$)	IN_{TS1}		PMR[17] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0
	SMI_O	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[17] and PMR[8])
AC13	V_{CORE}	PWR	---	---	---
AC14	V_{CORE}	PWR	---	---	---
AC15	GPWIO0	I/O (PU_{100})	IN_{BTN}, $TS_{2/14}$	V_{SB}	---
AC16	V_{SS}	GND	---	---	---
AC17	V_{IO}	PWR	---	---	---
AC18	V_{IO}	PWR	---	---	---
AC19	V_{SS}	GND	---	---	---
AC20	TRST#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AC21	AC97_CLK	O	$O_{2/5}$	V_{IO}	PMR[25] = 1
AC22	BIT_CLK	I	IN_T	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_TRDY#	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1
AC23	NC	---	---	V_{IO}	---
AC24	GPIO18	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_AD0	O	$O_{8/8}$		PMR[27] = 1 or FPCI_MON = 1)
AC25	GNT3#	O	O_{PCI}	V_{IO}	---
AC26	REQ2#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	
AD1	IDE_DATA13	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AD2	IDE_DATA2	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AD3	IDE_DATA12	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AD4	IDE_DATA3	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AD5	IDE_DATA8	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AD6	IDE_RST#	O	$O_{1/4}$	V_{IO}	---
AD7	DPOS_PORT1	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AD8	DPOS_PORT3	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AD9	AV_{SSUSB}	GND	---	---	---

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AD10	FMUL3B	I/O	IN_{TS}, $TS_{2/5}$	V_{IO}	PMR[29] = 0
	TEST1	O	$O_{2/5}$		PMR[29] = 1
AD11	SIN	I	IN_{TS}	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_C/BE1#	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1
AD12	GPIO6	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	DTR#/BOUT	O	$O_{1/4}$		PMR[18] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	IDE_IOR1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0
	INTR_O	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[18] and PMR[8])
AD13	V_{CORE}	PWR	---	---	---
AD14	V_{CORE}	PWR	---	---	---
AD15[2,3]	ONCTL#	O	OD_{14}	V_{SB}	---
AD16	V_{SBL}	PWR	---	---	---
AD17	V_{IO}	PWR	---	---	---
AD18	V_{IO}	PWR	---	---	---
AD19	POWER_EN	O	$O_{1/4}$	V_{IO}	---
AD20	TDO	O ($PU_{22.5}$)	O_{PCI}	V_{IO}	---
AD21	GTEST	I ($PD_{22.5}$)	IN_{AB}	V_{IO}	---
AD22	SDATA_OUT	O	$O_{2/5}$	V_{IO}	---
	CLKSEL2	I (PD_{100})	IN_{AB}		Strap
AD23	GPIO13	I/O ($PU_{22.5}$)	IN_{AB}, $O_{2/8}$	V_{IO}	PMR[19] = 0 and PMR[27] = 0 and FPCI_MON = 0
	AB2D	I/O ($PU_{22.5}$)	IN_{AB}, OD_8		PMR[19] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_AD4	O ($PU_{22.5}$)	$O_{2/8}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[19])
AD24	GPIO1	I/O ($PU_{22.5}$)	IN_{PCI}, O_{PCI}	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	FPCICLK	O	O_{PCI}		PMR[27] = 1 or FPCI_MON = 1
AD25	GNT1#	O	O_{PCI}	V_{IO}	---
	CLKSEL1	I (PD_{100})	IN_{PCI}		Strap
AD26	INTA#	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AE1	V_{IO}	PWR	---	---	---
AE2	V_{SS}	GND	---	---	---
AE3	IDE_DATA4	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AE4	IDE_DATA5	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AE5	IDE_DATA6	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AE6	IDE_DATA15	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AE7	DNEG_PORT1	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AE8	DNEG_PORT3	I/O	IN_{USB}, O_{USB}	$AV_{C\text{-}CUSB}$	---
AE9	V_{PLL}	PWR	---	---	---
AE10	X27O	O	WIRE	V_{IO}	---
AE11	SOUT	O	$O_{8/8}$	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_C/BE2#	O	$O_{8/8}$		PMR[27] = 1 or FPCI_MON = 1
AE12	GPIO9	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	DCD#	I	IN_{TS}		PMR[18] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	IDE_IOW1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[18] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_IRDY#	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[18] and PMR[8])
AE13	V_{CORE}	PWR	---	---	---
AE14	V_{CORE}	PWR	---	---	---
AE15	THRM#	I	IN_{BTN}	V_{SB}	---
AE16	GPWIO1	I/O (PU_{100})	IN_{BTN}, $TS_{2/14}$	V_{SB}	---
AE17[2,3]	PWRCNT1	O	OD_{14}	V_{SB}	---
AE18	GPIO16	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[6] = 1 and PMR[0] = 0 and PMR[27] = 0 and FPCI_MON = 0
	PC_BEEP	O ($PU_{22.5}$)	$O_{2/5}$		PMR[6] = 1 and PMR[0] = 1 and PMR[27] = 0 and FPCI_MON = 0
	IRRX1	I ($PU_{22.5}$)	IN_{TS}	V_{SB}/V_{IO}	PMR[6] = 0 and PMR[0] = x and PMR[27] = 0 and FPCI_MON = 0 (Note: Power rail controlled by SW)
	F_DEVSEL#	O ($PU_{22.5}$)	$O_{2/5}$	V_{IO}	PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[6] and PMR[0])
AE19	OVER_CUR#	I	IN_{TS1}	V_{IO}	---
AE20	TCK	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AE21	POR#	I	IN_{TS}	V_{IO}	---

Signal Definitions (Continued)

Table 2-2. Ball Assignment - Sorted by Ball Number (Continued)

Ball No.	Signal Name	I/O (PU/PD)	Buffer[1] Type	Power Rail	Configuration
AE22	SYNC	O	$O_{2/5}$	V_{IO}	---
	CLKSEL3	I (PD_{100})	IN_{AB}		Strap
AE23	GPIO12	I/O ($PU_{22.5}$)	IN_{AB}, $O_{2/8}$	V_{IO}	PMR[19] = 0 and PMR[27] = 0 and FPCI_MON = 0
	AB2C	I/O ($PU_{22.5}$)	IN_{AB}, OD_8		PMR[19] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_AD3	O ($PU_{22.5}$)	$O_{2/8}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[19])
AE24	NC	---	---	---	---
AE25	V_{SS}	GND	---	---	---
AE26	V_{IO}	PWR	---	---	---
AF1	V_{SS}	GND	---	---	---
AF2	V_{IO}	PWR	---	---	---
AF3	IDE_DATA11	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AF4	IDE_DATA10	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AF5	IDE_DATA9	I/O	IN_{TS1}, $TS_{1/4}$	V_{IO}	---
AF6	AV_{CCUSB}	PWR	---	---	---
AF7	DPOS_PORT2	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
AF8	DNEG_PORT2	I/O	IN_{USB}, O_{USB}	AV_{C-CUSB}	---
AF9	AV_{SSPLL}	GND	---	---	---
AF10	X27I	I	WIRE	V_{IO}	---
AF11	GPIO7	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[17] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	RTS#	O	$O_{1/4}$		PMR[17] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	IDE_DACK1#	O ($PU_{22.5}$)	$O_{1/4}$		PMR[17] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_C/BE3#	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[17] and PMR[8])
AF12	GPIO10	I/O ($PU_{22.5}$)	IN_{TS}, $O_{1/4}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	DSR#	I	IN_{TS}		PMR[18] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0
	IDE_IORDY1	I ($PU_{22.5}$)	IN_{TS1}		PMR[18] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0
	F_FRAME#	O	$O_{1/4}$		PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[18] and PMR[8])
AF13	V_{CORE}	PWR	---	---	---
AF14	V_{CORE}	PWR	---	---	---
AF15	PWRBTN#	I (PU_{100})	IN_{BTN}	V_{SB}	---
AF16	V_{SB}	PWR	---	---	---
AF17	GPWIO2	I/O (PU_{100})	IN_{BTN}, $TS_{2/14}$	V_{SB}	---
AF18[2,3]	PWRCNT2	O	OD_{14}	V_{SB}	---
AF19	GPIO11	I/O ($PU_{22.5}$)	IN_{TS}, $O_{8/8}$	V_{IO}	PMR[18] = 0 and PMR[8] = 0
	RI#	I ($PU_{22.5}$)	IN_{TS}	V_{SB}/V_{IO}	PMR[18] = 1 and PMR[8] = 0 (Note: Power rail controlled by SW)
	IRQ15	I ($PU_{22.5}$)	IN_{TS}	V_{IO}	PMR[18] = 0 and PMR[8] = 1
AF20	TDI	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AF21	TMS	I ($PU_{22.5}$)	IN_{PCI}	V_{IO}	---
AF22	SDATA_IN	I	IN_T	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_GNT0#	O	$O_{2/5}$		PMR[27] = 1 or FPCI_MON = 1
AF23	AC97_RST#	O	$O_{2/5}$	V_{IO}	PMR[27] = 0 and FPCI_MON = 0
	F_STOP#	O	$O_{2/5}$		PMR[27] = 1 or FPCI_MON = 1
AF24	NC	---	---	---	---
AF25	V_{IO}	PWR	---	---	---
AF26	V_{SS}	GND	---	---	---

1. For Buffer Type definitions, refer to Table 7-7 "Buffer Types" on page 286.
2. Is back-drive protected (MD[63:0], DOS_PORT1, DNEG_PORT1, DOS_PORT2, DNEG_PORT2, DOS_PORT3, DNEG_PORT3, ONCTL#, and PWRCNT[2:1]).
3. Is 5V tolerant (ONCTL# and PWRCNT[2:1])

Signal Definitions (Continued)

Table 2-3. Ball Assignment - Sorted Alphabetically by Signal Name

Signal Name	Ball No.
A0	E25
A1	F26
A2	F25
A3	G26
A4	G25
A5	H26
A6	H25
A7	D24
A8	E23
A9	J26
A10	J25
A11	K25
A12	K26
A13	F23
A14	F24
A15	L26
A16	R25
A17	K24
A18	T26
A19	T25
A20	L24
A21	U26
A22	U25
A23	M24
AB1C	Y1
AB1D	Y2
AB2C	AE23
AB2D	AD23
AC97_CLK	AC21
AC97_RST#	AF23
AD0	E25
AD1	F26
AD2	F25
AD3	G26
AD4	G25
AD5	H26
AD6	H25
AD7	D24
AD8	E23
AD9	J26
AD10	J25
AD11	K25
AD12	K26
AD13	F23
AD14	F24
AD15	L26
AD16	R25
AD17	K24
AD18	T26
AD19	T25
AD20	L24
AD21	U26
AD22	U25
AD23	M24
AD24	V25
AD25	W26
AD26	N24
AD27	P24
AD28	W25
AD29	Y26
AD30	R24
AD31	Y25
AV_{CCUSB}	AF6
AV_{SSPLL}	AF9
AV_{SSUSB}	AD9
BA0	C11
BA1	C12
BHE#	N26
BIT_CLK	AC22
BOOT16	C23
C/BE0#	E24
C/BE1#	G24
C/BE2#	R26
C/BE3#	V26
CASA#	D3
CKEA	A19
CLKSEL0	D23
CLKSEL1	AD25
CLKSEL2	AD22
CLKSEL3	AE22
CS0#	B3
CS1#	A3
CTS#	AC12
D0	V25
D1	W26
D10	R26
D11	V26
D12	H24
D13	N25
D14	P26
D15	M25
D2	N24
D3	P24
D4	W25
D5	Y26
D6	R24
D7	Y25
D8	E24
D9	G24
DCD#	AE12
DEVSEL#	N26
DNEG_PORT1	AE7
DNEG_PORT2	AF8
DNEG_PORT3	AE8
DOCCS#	D21
DOCR#	D22
DOCW#	C22
DPOS_PORT1	AD7
DPOS_PORT2	AF7
DPOS_PORT3	AD8
DQM0	M1
DQM1	M2
DQM2	B11
DQM3	A11
DQM4	P3
DQM5	N3
DQM6	C14
DQM7	C15
DSR#	AF12

Signal Definitions (Continued)

Table 2-3. Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
DTR#/BOUT	AD12
F_AD0	AC24
F_AD1	Y1
F_AD2	Y2
F_AD3	AE23
F_AD4	AD23
F_AD5	AB23
F_AD6	B20
F_AD7	AB24
F_C/BE0#	AA24
F_C/BE1#	AD11
F_C/BE2#	AE11
F_C/BE3#	AF11
F_DEVSEL	AE18
F_FRAME#	AF12
F_GNT0#	AF22
F_IRDY#	AE12
F_STOP#	AF23
F_TRDY#	AC22
F5BAR4CS#	B21
F5BAR5CS#	A22
FMUL3B	AD10
FPCI_MON	AB25
FPCICLK	AD24
FRAME#	P25
GNT0#	AB26
GNT1#	AD25
GNT2#	U24
GNT3#	AC25
GPIO0	B22
GPIO1	AD24
GPIO2	B21
GPIO3	A22
GPIO6	AD12
GPIO7	AF11
GPIO8	AC12
GPIO9	AE12
GPIO10	AF12
GPIO11	AF19
GPIO12	AE23

Signal Name	Ball No.
GPIO13	AD23
GPIO14	D22
GPIO15	C22
GPIO16	AE18
GPIO17	B23
GPIO18	AC24
GPIO19	Y24
GPIO20	D21
GPIO32	E26
GPIO33	D25
GPIO34	D26
GPIO35	C25
GPIO36	C26
GPIO37	B24
GPIO38	AB23
GPIO39	A24
GPIO40	B20
GPIO41	AA24
GPIO47	AB24
GPWIO0	AC15
GPWIO1	AE16
GPWIO2	AF17
GTEST	AD21
GXCLK	C20
IDE_ADDR0	AA4
IDE_ADDR1	AA3
IDE_ADDR2	Y3
IDE_CS0#	AA2
IDE_CS1#	AA1
IDE_DACK0#	AB1
IDE_DACK1#	AF11
IDE_DATA0	AB4
IDE_DATA1	AC4
IDE_DATA2	AD2
IDE_DATA3	AD4
IDE_DATA4	AE3
IDE_DATA5	AE4
IDE_DATA6	AE5
IDE_DATA7	AC5
IDE_DATA8	AD5

Signal Name	Ball No.
IDE_DATA9	AF5
IDE_DATA10	AF4
IDE_DATA11	AF3
IDE_DATA12	AD3
IDE_DATA13	AD1
IDE_DATA14	AC3
IDE_DATA15	AE6
IDE_DREQ0	AB3
IDE_DREQ1	AC12
IDE_IOR0#	AC1
IDE_IOR1#	AD12
IDE_IORDY0	AB2
IDE_IORDY1	AF12
IDE_IOW0#	AC2
IDE_IOW1#	AE12
IDE_RST#	AD6
INTA#	AD26
INTB#	W24
INTC#	Y24
INTD#	V24
INTR_O	AD12
IOCHRDY	B23
IOCS0#	B23
IOCS1#	D22
IOR#	D22
IOW#	C22
IRDY#	P26
IRQ14	AC6
IRQ15	AF19
IRRX1	AE18
IRTX	C20
LAD0	E26
LAD1	D25
LAD2	D26
LAD3	C25
LDRQ#	C26
LFRAME#	B24
LPC_ROM	AB26
MA0	L1
MA1	L2

Signal Definitions (Continued)

Table 2-3. Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
MA2	K1
MA3	M3
MA4	L3
MA5	K3
MA6	A8
MA7	C10
MA8	A9
MA9	B10
MA10	A10
MA11	C13
MA12	D2
MD0	V1
MD1	U1
MD2	T1
MD3	R1
MD4	P1
MD5	P2
MD6	N1
MD7	N2
MD8	K2
MD9	J1
MD10	H1
MD11	G1
MD12	F1
MD13	F2
MD14	E1
MD15	E2
MD16	B4
MD17	A4
MD18	B5
MD19	A5
MD20	B6
MD21	A6
MD22	B7
MD23	A7
MD24	B12
MD25	A12
MD26	B13
MD27	A13
MD28	B14

Signal Name	Ball No.
MD29	A14
MD30	B15
MD31	A15
MD32	V2
MD33	V3
MD34	U2
MD35	U3
MD36	T2
MD37	T3
MD38	R2
MD39	R3
MD40	J3
MD41	J2
MD42	H3
MD43	H2
MD44	G3
MD45	G2
MD46	F3
MD47	F4
MD48	D6
MD49	C6
MD50	D7
MD51	C7
MD52	B8
MD53	C8
MD54	B9
MD55	C9
MD56	C16
MD57	B16
MD58	C17
MD59	B17
MD60	C18
MD61	B18
MD62	C19
MD63	B19
NC	A21
NC	A23
NC	D20
NC	AC23
NC	AE24

Signal Name	Ball No.
NC	AF24
ONCTL#	AD15
OVER_CUR#	AE19
PAR	H24
PC_BEEP	AE18
PCICLK	AA23
PCICLKO	AB25
PCIRST#	M26
PERR#	J24
PLL5B	AC11
POR#	AE21
POWER_EN	AD19
PWRBTN#	AF15
PWRCNT1	AE17
PWRCNT2	AF18
RASA#	C1
RD#	D23
REQ0#	AA26
REQ1#	AA25
REQ2#	AC26
REQ3#	T24
RI#	AF19
ROMCS#	C23
RTS#	AF11
SDATA_IN	AF22
SDATA_OUT	AD22
SDCLK_IN	D4
SDCLK_OUT	C4
SDCLK0	E4
SDCLK1	C5
SDCLK2	A16
SDCLK3	A17
SDTEST0	A20
SDTEST1	D1
SDTEST2	A18
SDTEST3	D5
SDTEST4	E3
SDTEST5	B20
SERIRQ	A24
SERR#	L25

Signal Definitions (Continued)

Table 2-3. Ball Assignment - Sorted Alphabetically by Signal Name (Continued)

Signal Name	Ball No.
SIN	AD11
SMI_O	AC12
SOUT	AE11
STOP#	M25
SYNC	AE22
TCK	AE20
TDI	AF20
TDO	AD20
TEST0	AA24
TEST1	AD10
TEST2	AC11
TEST3	C20
THRM#	AE15
TMS	AF21
TRDE#	B22
TRDY#	N25
TRST#	AC20
V_{BAT}	W3,
V_{CORE} (Total of 24)	D12, D15, L4, L12, L15, L23, N4, N23, P4, P23, R11, R16, T4, T12, T15, T23, AC13, AC14, AD13, AD14, AE13, AE14, AF13, AF14

Signal Name	Ball No.
V_{IO} (Total of 28)	A2, A25, B1, B26, C3, C24, D8, D10, D17, D19, G4, G23, J4, J23, V4, V23, Y4, Y23, AC8, AC10, AC17, AC18, AD17, AD18, AE1, AE26, AF2, AF25
V_{PLL}	AE9
V_{SB}	AF16
V_{SBL}	AD16

Signal Name	Ball No.
V_{SS} (Total of 60)	A1, A26, B2, B25, D9, D11, D13, D14, D16, D18, H4, H23, K4, K23, L11, L13, L14, L16, M4, M11, M12, M13, M14, M15, M16, M23, N11, N12, N13, N14, N15, N16, P11, P12, P13, P14, P15, P16, R4, R12, R13, R14, R15, R23, T11, T13, T14, T16, U4, U23, W4, W23, AC7, AC9, AC16, AC19, AE2, AE25, AF1, AF26
WEA#	C2
WR#	C21
X27I	AF10
X27O	AE10
X32I	W1
X32O	W2

Signal Definitions (Continued)

2.2 STRAP OPTIONS

Several balls are read at power-up that set up the state of the SC1100. These balls are typically multiplexed with other functions that are outputs after the power-up sequence is complete. The SC1100 must read the state of the balls at power-up and the internal PU or PD resistors do not guarantee the correct state will be read. Therefore, it is required that an external PU or PD resistor with a value of 1.5 KΩ be placed on the balls listed in Table 2-4. The value of the resistor is important to ensure that the proper state is read during the power-up sequence. If the ball is not read correctly at power-up, the SC1100 may default to a state that causes it to function improperly, possibly resulting in application failure.

Table 2-4. Strap Options

Strap Option	Muxed With	Ball #	Nominal Internal PU or PD	External PU/PD Strap Settings		Register References
				Strap = 0 (PD)	Strap = 1 (PU)	
CLKSEL0	RD#	D23	PD_{100}	See Section 2.4.1 on page 36 for CLKSEL strap options.		GCB+I/O Offset 1Eh[9:8] (RO): Value programmed at reset by CLKSEL[1:0]. GCB+I/O Offset 10h[3:0] (RO): Value programmed at reset by CLKSEL[3:0]. GCB+I/O Offset 1Eh[3:0] (R/W, but write not recommended): Value programmed at reset by CLKSEL[3:0]. **Note:** Values for GCB+I/O Offset 10h[3:0] and 1Eh[3:0] are not the same.
CLKSEL1	GNT1#	AD25	PD_{100}			
CLKSEL2	SDATA_OUT	AD22	PD_{100}			
CLKSEL3	SYNC	AE22	PD_{100}			
BOOT16	ROMCS#	C23	PD_{100}	Enable boot from 8-bit ROM	Enable boot from 16-bit ROM	GCB+I/O Offset 34h[3] (RO): Reads back strap setting. GCB+I/O Offset 34h[14] (R/W): Used to allow the ROMCS# width to be changed under program control.
FPCI_MON	PCICLKO	AB25	PD_{100}	Disable Fast-PCI, INTR_O, and SMI_O monitoring signals.	Enable Fast-PCI, INTR_O, and SMI_O monitoring signals. (Useful during debug.)	GCB+I/O Offset 34h[30] (MCR[30]) (RO): Reads back strap setting. **Note:** For normal operation, strap this signal low using a 1.5 KΩ resistor.
LPC_ROM	GNT0#	AB26	PD_{100}	Disable LPC ROM boot	Enable boot from LPC ROM device	This strap signal, when pulled high, sets bit F0BAR1+I/O Offset 10h[15], LPC ROM Addressing Enable.

Note: Accuracy of internal PU/PD resistors: 80K to 250K.

Location of the GCB (General Configuration Block) cannot be determined by software. See the *SC1100 Information Appliance On a Chip device errata* document.

Signal Definitions (Continued)

2.3 MULTIPLEXING CONFIGURATION

The tables that follow list multiplexing options and their configurations. Certain multiplexing options may be chosen per signal; others are available only for a group of signals.

A pull-up resistor is optional on the balls multiplexed with GPIO11, GPIO16, GPIO40, and GPIO41 regardless of their selected function. For other GPIO pins, the pull-up resistor is optional with regards to the selected function on the relevant ball (see Table 2-2 on page 19 or Table 7-6 on page 285). The pull-up resistor can be disabled by writing Core Logic registers.

Table 2-5. Two-Signal/Group Multiplexing

Ball No.	Default		Alternate	
	Signal	**Configuration**	**Signal**	**Configuration**
---	**Sub-ISA**		**GPIO**	
B22	TRDE#	PMR[12] = 0	GPIO0	PMR[12] = 1
	Internal Test			
AD10	FMUL3B	PMR[29] = 0	TEST1	PMR[29] = 1
AC11	PLL5B		TEST2	
---	**GPIO**		**LPC**	
E26	GPIO32	PMR[14] = 0	LAD0[1]	PMR[14] = 1
D25	GPIO33		LAD1[1]	
D26	GPIO34		LAD2[1]	
C25	GPIO35		LAD3[1]	
C26	GPIO36		LDRQ#[1]	
B24	GPIO37		LFRAME#[1]	
A24	GPIO39	PMR[16] = 0	SERIRQ[1]	PMR[16] = 1
---	**GPIO**		**PCI**	
Y24	GPIO19	PMR[4] = 0	INTC#[2]	PMR[4] = 1
---	**GPIO**		**Sub-ISA**	
B21	GPIO2	PMR[1] = 0	F5BAR4CS#	PMR[1] = 1
A22	GPIO3	PMR[3] = 0	F5BAR5CS#	PMR[3] = 1
D21	GPIO20	PMR[7] = 0	DOCCS#	PMR[7] = 1

Signal Definitions (Continued)

Table 2-5. Two-Signal/Group Multiplexing (Continued)

Ball No.	Default		Alternate	
	Signal	Configuration	Signal	Configuration
---	AC97, GPIO, ACB, UART		FPCI Monitoring [3]	
AF23	AC97_RST#	PMR[27] = 0 and FPCI_MON = 0	F_STOP#	PMR[27] = 1 or FPCI_MON = 1
AF22	SDATA_IN[4]		F_GNT0#	
AC22	BIT_CLK[5]		F_TRDY#	
AD24	GPIO1		FPCICLK	
AD11	SIN[6]		F_C/BE1#	
AC24	GPIO18		F_AD0	
Y1	AB1C[7]		F_AD1	
Y2	AB1D[7]		F_AD2	
AB23	GPIO38		F_AD5	
AB24	GPIO47		F_AD7	
AE11	SOUT		F_C/BE2#	

1. All LPC inputs (including SERIRQ), if selected but not used, should be tied high.
2. INTC# input, if selected but not used, should be tied high.
3. AC97 interface is not functional.
4. SDATA_IN input, if selected but not used, should be tied low.
5. BIT_CLK input, if selected but not used, should be tied low.
6. SIN input, if selected but not used, should be tied high.
7. AB1C, AB1D inputs, if selected but not used, should be tied high.

Signal Definitions (Continued)

Table 2-6. Three-Signal/Group Multiplexing

Ball No.	Default		Alternate1		Alternate2	
	Signal	Configuration	Signal	Configuration	Signal	Configuration
---	Sub-ISA		GPIO		Sub-ISA	
C22	IOW#	PMR[2] = 0 and PMR[21] = 0	GPIO15	PMR[2] =1 and PMR[21] = 0	DOCW#	PMR[2] = 1 and PMR[21] = 1
---	Infrared		Internal Test			
C20	IRTX	PMR[6] = 0 and PMR[29] = x	GXCLK	PMR[6] = 1 and PMR[29] = 0	TEST3	PMR[6] = 1 and PMR[29] = 1
---	GPIO		Sub-ISA		Sub-ISA	
B23	GPIO17	PMR[5] = 0 and PMR[9] = 0	IOCS0#	PMR[5] = 1 and PMR[9] = 0	IOCHRDY[1]	PMR[5] = 1 and PMR[9] = 1
AF19	GPIO11	PMR[18] = 0 and PMR[8] = 0	RI[2]	PMR[18] = 1 and PMR[8] = 0	IRQ15[3]	PMR[18] = 0 and PMR[8] = 1
---	GPIO		ACCESS.bus		FPCI Monitoring	
AE23	GPIO12[4]	PMR[19] = 0	AB2C[5]	PMR[19] = 1	F_AD3	PMR[27] = 1
AD23	GPIO13		AB2D[5]		F_AD4	
---	GPIO		Internal Test		FPCI Monitoring	
B20	GPIO40	PMR[28] = 0 and PMR[27] = 1 or FPC_MON = 1	SDTEST5	PMR[28] = 1 and PMR[27] = 1 or FPC_MON = 1	F_AD6	PMR[27] = 1 or FPC_MON = 1 (overrides PMR[28])
AA24	GPIO41	PMR[29] = 0 and PMR[27] = 1 or FPC_MON = 1	TEST0	PMR[29] = 1 and PMR[27] = 1 or FPC_MON = 1	F_C/BE0#	PMR[27] = 1 or FPC_MON = 1 (overrides PMR[29])

1. IOCHRDY input: If selected but not used, should be tied high.
2. RI# input: If selected but not used, should be tied high.
3. IRQ15 input: If selected but not used, should be tied low.
4. GPIO12 input: If selected but not used, should be tied low. This is true for all GPIOs in this device.
5. AB2C, AB2D inputs: If selected but not used, should be tied high.

Signal Definitions (Continued)

Table 2-7. Four-Signal /Group Multiplexing

Ball No.	Default		Alternate1		Alternate2		Alternate3	
	Signal	Configuration	Signal	Configuration	Signal	Configuration	Signal	Configuration
---	GPIO		Sub-ISA		Sub-ISA		Sub-ISA	
D22	GPIO14	PMR[2] = 0 and PMR[21] = 0	IOCS1#	PMR[2] = 1 and PMR[21] = 0	IOR#	PMR[2] = 0 and PMR[21] = 1	DOCR#	PMR[2] = 1 and PMR[21] = 1
---	Infrared		GPIO		AC97		FPCI Monitoring	
AE18	IRRX1[1]	PMR[6] = 0 PMR[0] = x and PMR[27] = 0 and FPCI_MON = 0	GPIO16	PMR[6] = 1 and PMR[0] = 0 and PMR[27] = 0 and FPCI_MON = 0	PC_BEEP	PMR[6] = 1 and PMR[0] = 1 and PMR[27] = 0 and FPCI_MON = 0	F_DEVSEL#	PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[6] and PMR[0])
	GPIO		UART		IDE Channel 1		FPCI Monitoring	
AF11	GPIO7	PMR[17] = 0 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0	RTS#	PMR[17] = 1 and PMR[8] = 0 and PMR[27] = 0 and FPCI_MON = 0	IDE_DACK1	PMR[17] = 0 and PMR[8] = 1 and PMR[27] = 0 and FPCI_MON = 0	F_C/BE3#	PMR[27] = 1 or FPCI_MON = 1 (overrides PMR[17], PMR[8], and PMR[18])
AC12	GPIO8		CTS#[2]		IDE_DREQ1[3]		SMI_O#	
AD12	GPIO6	PMR[18] = 0 and PMR[8] = 0	DTR#/ BOUT	PMR[18] = 1 and PMR[8] = 0	IDE_IOR1	PMR[18] = 0 and PMR[8] = 1	INTR_O	
AE12	GPIO9		DCD#[4]		IDE_IOW1		F_IRDY	
AF12	GPIO10		DSR#[5]		IDE_IORDY1[6]		F_FRAME#	

1. IRRX1 input: If selected but not used, should be tied high.
2. CTS# input: If selected but not used, should be tied low.
3. IDE_DREQ1 input: If selected but not used, should be tied low.
4. DCD# input: If selected but not used, should be tied high.
5. DSR# input: If selected but not used, should be tied low.
6. IDE_IORDY1 input: If selected but not used, should be tied high.

Signal Definitions (Continued)

2.4 SIGNAL DESCRIPTIONS

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

2.4.1 System Interface

Signal Name	Ball No.	Type	Description	Mux
CLKSEL1	AD25	I	**Fast-PCI Clock Selects.** These strap signals are used to set the internal Fast-PCI clock. 00 = 33.3 MHz 01 = Reserved 10 = 66.7 MHz 11 = 33.3 MHz During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	GNT1#
CLKSEL0	D23			RD#
CLKSEL3	AE22	I	**Maximum Core Clock Multiplier.** These strap signals are used to set the maximum allowed multiplier value for the core clock. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	SYNC
CLKSEL2	AD22			SDATA_OUT
BOOT16	C23	I	**Boot ROM is 16 Bits Wide.** This strap signal enables the optional 16-bit wide Sub-ISA bus. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	ROMCS#
LPC_ROM	AB26	I	**LPC_ROM.** If pulled high during reset, this strap signal forces selection of the LPC bus and sets bit F0BAR1+I/O Offset 10h[15], LPC ROM Addressing Enable. It enables the SC1100 to boot from a ROM connected to the LPC bus. During system reset, an internal pull-down resistor of 100 KΩ exists on these balls. An external pull-up or pull-down resistor of 1.5 KΩ must be used.	GNT0#
FPCI_MON	AB25	I	**Fast-PCI Monitoring.** The strap on this ball forces selection of Fast-PCI monitoring signals. For normal operation, strap this signal low using a 1.5 KΩ resistor. The value of this strap can be read on the MCR[30].	PCICLKO
POR#	AE21	I	**Power On Reset.** POR# is the system reset signal generated from the power supply to indicate that the system should be reset.	---
X32I	W1	I	**Crystal Connections.** Connected directly to a 32.768 KHz crystal. This clock input is required even if the internal RTC is not being used. Some of the internal clocks are derived from this clock. If an external clock is used, it should be connected to X32I, using a voltage level of 0V to V_{CORE} +10% maximum. X32O should remain unconnected.	---
X32O	W2	O		---
X27I	AF10	I	**Crystal Connections.** Connected directly to a 27.000 MHz crystal. Some of the internal clocks are derived from this clock. If an external clock is used, it should be connected to X27I, using a voltage level of 0V to V_{IO} and X27O should be remain unconnected.	---
X27O	AE10	O		---

Signal Definitions (Continued)

2.4.1 System Interface (Continued)

Signal Name	Ball No.	Type	Description	Mux
PCIRST#	M26	O	**PCI and System Reset.** PCIRST# is the reset signal for the PCI bus and system. It is asserted for approximately 100 µs after POR# is negated.	---

2.4.2 Memory Interface Signals

Signal Name	Ball No.	Type	Description	Mux
MD[63:0]	See Table 2-3 on page 27	I/O	**Memory Data Bus.** The data bus lines driven to/from system memory.	---
MA[12:0]	See Table 2-3 on page 27	O	**Memory Address Bus.** The multiplexed row/column address lines driven to the system memory. Supports 256-Mbit SDRAM.	---
BA1	C12	O	**Bank Address Bits.** These bits are used to select the component bank within the SDRAM.	---
BA0	C11			---
CS1#	A3	O	**Chip Selects.** These bits are used to select the module bank within system memory. Each chip select corresponds to a specific module bank. If CS# is high, the bank(s) do not respond to RAS#, CAS#, and WE# until the bank is selected again.	---
CS0#	B3			---
RASA#	C1	O	**Row Address Strobe.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. RASA# is used with CS[1:0]#.	---
CASA#	D3	O	**Column Address Strobe.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. CASA# is used with CS[1:0]#.	---
WEA#	C2	O	**Write Enable.** RAS#, CAS#, WE# and CKE are encoded to support the different SDRAM commands. WEA# is used with CS[1:0]#.	---
DQM[7:0]	C15, C14, N3, P3, A11, B11, M2, M1	O	**Data Mask Control Bits.** During memory read cycles, these outputs control whether SDRAM output buffers are driven on the MD bus or not. All DQM signals are asserted during read cycles. During memory write cycles, these outputs control whether or not MD data is written into SDRAM. DQM[7:0] connect directly to the [DQM7:0] pins of each DIMM connector.	---
CKEA	A19	O	**Clock Enable.** These signals are used to enter Suspend/power-down mode. CKEA is used with CS[1:0]#. If CKE goes low when no read or write cycle is in progress, the SDRAM enters power-down mode. To ensure that SDRAM data remains valid, the self-refresh command is executed. To exit this mode, and return to normal operation, drive CKE high. These signals should have an external pull-down resistor of 33 KΩ.	---

Signal Definitions (Continued)

2.4.2 Memory Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
SDCLK3	A17	O	**SDRAM Clocks.** SDRAM uses these clocks to sample all control, address, and data lines. To ensure that the Suspend mode functions correctly, SDCLK3 and SDCLK1 should be used with CS1#. SDCLK2 and SDCLK0 should be used together with CS0#.	---
SDCLK2	A16			---
SDCLK1	C5			---
SDCLK0	E4			---
SDCLK_IN	D4	I	**SDRAM Clock Input.** The SC1100 samples the memory read data on this clock. Works in conjunction with the SDCLK_OUT signal.	---
SDCLK_OUT	C4	O	**SDRAM Clock Output.** This output is routed back to SDCLK_IN. The board designer should vary the length of the board trace to control skew between SDCLK_IN and SDCLK.	---

2.4.3 ACCESS.bus Interface Signals

Signal Name	Ball No.	Type	Description	Mux
AB1C	Y1	I/O	**ACCESS.bus 1 Serial Clock.** This is the serial clock for the interface.	F_AD1
AB1D	Y2	I/O	**ACCESS.bus 1 Serial Data.** This is the bidirectional serial data signal for the interface.	F_AD2
AB2C	AE23	I/O	**ACCESS.bus 2 Serial Clock.** This is the serial clock for the interface.	GPIO12+F_AD3
AB2D	AD23	I/O	**ACCESS.bus 2 Serial Data.** This is the bidirectional serial data signal for the interface.	GPIO13+F_AD4

2.4.4 PCI Bus Interface Signals

Signal Name	Ball No.	Type	Description	Mux
PCICLK	AA23	I	**PCI Clock.** PCICLK provides timing for all transactions on the PCI bus. All other PCI signals are sampled on the rising edge of PCICLK, and all timing parameters are defined with respect to this edge.	---
PCICLKO	AB25	O	**PCI Clock Output.** Provides a clock for the system at 33 MHz. This clock is asynchronous to PCI signals. It should be connected to a low skew buffer with multiple outputs, of which one is connected to the PCICLK input. All PCI clock users in the system (including PCICLK) should receive the clock with as low a skew as possible. **Note:** Only a CMOS load should be connected to this signal.	FPCI_MON (Strap)
AD[31:24]	See Table 2-3 on page 27	I/O	**Multiplexed Address and Data.** A bus transaction consists of an address phase in the cycle in which FRAME# is asserted followed by one or more data phases. During the address phase, AD[31:0] contain a physical 32-bit address. For I/O, this is a byte address. For configuration and memory, it is a DWORD address. During data phases, AD[7:0] contain the least significant byte (LSB) and AD[31:24] contain the most significant byte (MSB).	D[7:0]
AD[23:0]				A[23:0]

Signal Definitions (Continued)

2.4.4 PCI Bus Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
C/BE3#	V26	I/O	**Multiplexed Command and Byte Enables.** During the address phase of a transaction when FRAME# is active, C/BE[3:0]# define the bus command. During the data phase, C/BE[3:0]# are used as byte enables. The byte enables are valid for the entire data phase and determine which byte lanes carry meaningful data. C/BE0# applies to byte 0 (LSB) and C/BE3# applies to byte 3 (MSB).	D11
C/BE2#	R26			D10
C/BE1#	G24			D9
C/BE0#	E24			D8
INTA#	AD26	I	**PCI Interrupts.** The SC1100 provides inputs for the optional “level-sensitive” PCI interrupts (also known in industry terms as PIRQx#). These interrupts can be mapped to IRQs of the internal 8259A interrupt controllers using PCI Interrupt Steering Registers 1 and 2 (F0 Index 5Ch and 5Dh).	---
INTB#	W24			---
INTC#	Y24			GPIO19
INTD#	V24			---
PAR	H24	I/O	**Parity.** Parity generation is required by all PCI agents. The master drives PAR for address- and write-data phases. The target drives PAR for read-data phases. Parity is even across AD[31:0] and C/BE[3:0]#. For address phases, PAR is stable and valid one PCI clock after the address phase. It has the same timing as AD[31:0] but is delayed by one PCI clock. For data phases, PAR is stable and valid one PCI clock after either IRDY# is asserted on a write transaction or after TRDY# is asserted on a read transaction. Once PAR is valid, it remains valid until one PCI clock after the completion of the data phase. (Also see PERR#.)	D12
FRAME#	P25	I/O	**Frame Cycle.** Frame is driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate the beginning of a bus transaction. While FRAME# is asserted, data transfers continue. FRAME# is deasserted when the transaction is in the final data phase.	---
IRDY#	P26	I/O	**Initiator Ready.** IRDY# is asserted to indicate that the bus master is able to complete the current data phase of the transaction. IRDY# is used in conjunction with TRDY#. A data phase is completed on any PCI clock in which both IRDY# and TRDY# are sampled as asserted. During a write, IRDY# indicates that valid data is present on AD[31:0]. During a read, it indicates that the master is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.	D14
TRDY#	N25	I/O	**Target Ready.** TRDY# is asserted to indicate that the target agent is able to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is complete on any PCI clock in which both TRDY# and IRDY# are sampled as asserted. During a read, TRDY# indicates that valid data is present on AD[31:0]. During a write, it indicates that the target is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.	D13

Signal Definitions (Continued)

2.4.4 PCI Bus Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
STOP#	M25	I/O	**Target Stop.** STOP# is asserted to indicate that the current target is requesting that the master stop the current transaction. This signal is used with DEVSEL# to indicate retry, disconnect, or target abort. If STOP# is sampled active by the master, FRAME# is deasserted and the cycle is stopped within three PCI clock cycles. As an input, STOP# can be asserted in the following cases: 1) If the PCI write buffers are full or if a previously buffered cycle has not completed. 2) On read cycles that cross cache line boundaries. This is conditional based upon the programming of GX1 module's PCI Configuration Register, Index 41h[1].	D15
DEVSEL#	N26	I/O	**Device Select.** DEVSEL# indicates that the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected. DEVSEL# is also driven by any agent that has the ability to accept cycles on a subtractive decode basis. As a master, if no DEVSEL# is detected within and up to the subtractive decode clock, a master abort cycle is initiated (except for special cycles which do not expect a DEVSEL# returned).	BHE#
PERR#	J24	I/O	**Parity Error.** PERR# is used for reporting data parity errors during all PCI transactions except a Special Cycle. The PERR# line is driven two PCI clocks after the data in which the error was detected. This is one PCI clock after the PAR that is attached to the data. The minimum duration of PERR# is one PCI clock for each data phase in which a data parity error is detected. PERR# must be driven high for one PCI clock before being placed in TRI-STATE. A target asserts PERR# on write cycles if it has claimed the cycle with DEVSEL#. The master asserts PERR# on read cycles.	---
SERR#	L25	I/O	**System Error.** SERR# can be asserted by any agent for reporting errors other than PCI parity, so that the PCI central agent notifies the processor. When the Parity Enable bit is set in the Memory Controller Configuration register, SERR# is asserted upon detection of a parity error in read operations from DRAM.	---
REQ3#	T24	I	**Request Lines.** Indicates to the arbiter that an agent requires the bus. Each master has its own REQ# line. REQ# priorities are based on the specified arbitration scheme.	---
REQ2#	AC26			---
REQ1#	AA25			---
REQ0#	AA26			---
GNT3#	AC25	O	**Grant Lines.** Indicate to the requesting master that it has been granted access to the bus. Each master has its own GNT# line. GNT# can be retracted at any time a higher REQ# is received or if the master does not begin a cycle within a minimum period of time (16 PCI clocks).	---
GNT2#	U24			---
GNT1#	AD25			CLKSEL1 (Strap)
GNT0#	AB26			LPC_ROM (Strap)

Signal Definitions (Continued)

2.4.5 Sub-ISA Interface Signals

Signal Name	Ball No.	Type	Description	Mux
A[23:0]	See Table 2-3 on page 27	O	**Address Lines**	AD[23:0]
D15	M25	I/O	**Data Bus**	STOP#
D14	P26			IRDY#
D13	N25			TRDY#
D12	H24			PAR
D11	V26			C/BE3#
D10	R26			C/BE2#
D9	G24			C/BE1#
D8	E24			C/BE0#
D[7:0]	Y25, R24, Y26, W25, P24, N24, W26, V25	I/O		AD[31:24]
BHE#	N26	O	**Byte High Enable.** With A0, defines byte accessed for 16 bit wide bus cycles.	DEVSEL#
IOCS1#	D22	O	**I/O Chip Selects**	GPIO14+IOR#+ DOCR#
IOCS0#	B23	O		GPIO17+ IOCHRDY
ROMCS#	C23	O	**ROM or Flash ROM Chip Select**	BOOT16 (Strap)
DOCCS#	D21	O	**DiskOnChip or NAND Flash Chip Select**	GPIO20
F5BAR4CS#	B21	O	**ROM or Flash ROM Chip Selects**	GPIO2
F5BAR5CS#	A22	O		GPIO3
TRDE#	B22	O	**Transceiver Data Enable Control.** Active low for Sub-ISA data transfers. The signal timing is as follows: • In a read cycle, TRDE# has the same timing as RD#. • In a write cycle, TRDE# is asserted (to active low) at the time WR# is asserted. It continues being asserted for one PCI clock cycle after WR# has been negated, then it is negated.	GPIO0
RD#	D23	O	**Memory or I/O Read.** Active on any read cycle.	CLKSEL0 (Strap)
WR#	C21	O	**Memory or I/O Write.** Active on any write cycle.	---
IOR#	D22	O	**I/O Read.** Active on any I/O read cycle.	GPIO14+IOCS1# +DOCR#
IOW#	C22	O	**I/O Write.** Active on any I/O write cycle.	DOCW#+GPIO15
DOCR#	D22	O	**DiskOnChip or NAND Flash Read.** Active on any memory read cycle to DiskOnChip.	GPIO14+IOCS1# +IOR#
DOCW#	C22	O	**DiskOnChip or NAND Flash Write.** Active on any memory write cycle to DiskOnChip.	IOW#+GPIO15
IOCHRDY	B23	I	**I/O Channel Ready**	GPIO17+IOCS0#

Signal Definitions (Continued)

2.4.6 Low Pin Count (LPC) Bus Interface Signals

Signal Name	Ball No.	Type	Description	Mux
LAD[3:0]	C25, D26, D25, E26	I/O	**LPC Address-Data.** Multiplexed command, address, bidirectional data, and cycle status.	GPIO[35:32]
LDRQ#	C26	I	**LPC DMA Request.** Encoded DMA request for LPC interface.	GPIO36
LFRAME#	B24	O	**LPC Frame.** A low pulse indicates the beginning of a new LPC cycle or termination of a broken cycle.	GPIO37
SERIRQ	A24	I/O	**Serial IRQ.** The interrupt requests are serialized over a single signal, where each IRQ level is delivered during a designated time slot.	GPIO39

2.4.7 IDE Interface Signals

Signal Name	Ball No.	Type	Description	Mux
IDE_RST#	AD6	O	**IDE Reset.** This signal resets all the devices that are attached to the IDE interface.	---
IDE_ADDR[2:0]	Y3, AA3, AA4	O	**IDE Address Bits.** These address bits are used to access a register or data port in a device on the IDE bus.	---
IDE_DATA[15:0]	See Table 2-3 on page 27	I/O	**IDE Data Lines.** IDE_DATA[15:0] transfers data to/from the IDE devices.	---
IDE_IOR0#	AC1	O	**IDE I/O Read Channels 0 and 1.** IDE_IOR0# is the read signal for Channel 0 and IDE_IOR1# is the read signal for Channel 1. Each signal is asserted at read accesses to the corresponding IDE port addresses.	---
IDE_IOR1#	AD12	O		GPIO6+DTR#/ BOUT+INTR_O
IDE_IOW0#	AC2	O	**IDE I/O Write Channels 0 and 1.** IDE_IOW0# is the write signal for Channel 0. IDE_IOW1# is the write signal for Channel 1. Each signal is asserted at write accesses to corresponding IDE port addresses.	---
IDE_IOW1#	AE12	O		GPIO9+DCD#+F _IRDY#
IDE_CS0#	AA2	O	**IDE Chip Selects 0 and 1.** These signals are used to select the command block registers in an IDE device.	---
IDE_CS1#	AA1	O		---
IDE_IORDY0	AB2	I	**I/O Ready Channels 0 and 1.** When deasserted, these signals extend the transfer cycle of any host register access if the required device is not ready to respond to the data transfer request.	---
IDE_IORDY1	AF12	I		GPIO10+DSR#+ F_FRAME#
IDE_DREQ0	AB3	I	**DMA Request Channels 0 and 1.** The IDE_DREQ signals are used to request a DMA transfer from the SC1100. The direction of transfer is determined by the IDE_IOR#/IOW# signals.	---
IDE_DREQ1	AC12	I		GPIO8+CTS# +SMI_O
IDE_DACK0#	AB1	O	**DMA Acknowledge Channels 0 and 1.** The IDE_DACK# signals acknowledge the DREQ request to initiate DMA transfers.	---
IDE_DACK1#	AF11	O		GPIO7+RTS# F_C/BE3#
IRQ14	AC6	I	**Interrupt Request Channels 0 and 1.** These input signals are edge-sensitive interrupts that indicate when the IDE device is requesting a CPU interrupt service.	---
IRQ15	AF19	I		GPIO11+RI#

Signal Definitions (Continued)

2.4.8 Universal Serial Bus (USB) Interface Signals

Signal Name	Ball No.	Type	Description	Mux
POWER_EN	AD19	O	**Power Enable.** This signal enables the power to a self-powered USB hub.	---
OVER_CUR#	AE19	I	**Overcurrent.** This signal indicates that the USB hub has detected an overcurrent on the USB.	---
DPOS_PORT1	AD7	I/O	**USB Port 1 Data Positive.** This signal is the Universal Serial Bus Data Positive for port 1.	---
DNEG_PORT1	AE7	I/O	**USB Port 1 Data Negative.** This signal is the Universal Serial Bus Data Negative for port 1.	---
DPOS_PORT2	AF7	I/O	**USB Port 2 Data Positive.** This signal is the Universal Serial Bus Data Positive for port 2.	---
DNEG_PORT2	AF8	I/O	**USB Port 2 Data Negative.** This signal is the Universal Serial Bus Data Negative for port 2.	---
DPOS_PORT3	AD8	I/O	**USB Port 3 Data Positive.** This signal is the Universal Serial Bus Data Positive for port 3.	---
DNEG_PORT3	AE8	I/O	**USB Port 3 Data Negative.** This signal is the Universal Serial Bus Data Negative for port 3.	---

2.4.9 Serial Port (UART) and Infrared (IR) Interface Signals

Signal Name	Ball No.	Type	Description	Mux
SIN	AD11	I	**Serial Input.** Receive composite serial data from the communications link (peripheral device, modem or other data transfer device).	F_C/BE1#
SOUT	AE11	O	**Serial Output.** Send composite serial data to the communications link (peripheral device, modem or other data transfer device). These signals are set active high after a system reset.	F_C/BE2#
RTS#	AF11	O	**Request to Send.** When low, indicates to the modem or other data transfer device that the corresponding UART is ready to exchange data. A system reset sets these signals to inactive high, and loopback operation holds them inactive.	GPIO7+ IDE_DACK1#+ F_C/BE3#
CTS#	AC12	I	**Clear to Send.** When low, indicates that the modem or other data transfer device is ready to exchange data.	GPIO8+ IDE_DREQ1+ SMI_O
DTR#/BOUT	AD12	O	**Data Terminal Ready Output.** When low, indicates to the modem or other data transfer device that the UART is ready to establish a communications link. After a system reset, this ball provides the DTR# function and sets this signal to inactive high. Loopback operation drive them inactive. **Baud Output.** Provides the associated serial channel baud rate generator output signal if test mode is selected (i.e., bit 7 of the EXCR1 Register is set).	GPIO6+ IDE_IOR1#+ INTR_O
RI#	AF19	I	**Ring Indicator.** When low, indicates to the modem that a telephone ring signal has been received by the modem. Monitored during power-off for wakeup event detection.	GPIO11+IRQ15

Signal Definitions (Continued)

2.4.9 Serial Port (UART) and Infrared (IR) Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
DCD#	AE12	I	**Data Carrier Detected.** When low, indicates that the data transfer device (e.g., modem) is ready to establish a communications link.	GPIO9+ IDE_IOW1#+ F_IRDY#
DSR#	AF12	I	**Data Set Ready.** When low, indicates that the data transfer device (e.g., modem) is ready to establish a communications link.	GPIO10+ IDE_IORDY1+ F_FRAME#
IRRX1	AE18	I	**IR Receive**. Primary input to receive serial data from the IR transceiver. Monitored during power-off for wakeup event detection.	GPIO16+ PC_BEEP+ F_DEVSEL
IRTX	C20	O	**IR Transmit**. IR serial output data.	GXCLK+TEST3

2.4.10 AC97 Audio Interface Signals

Signal Name	Ball No.	Type	Description	Mux
BIT_CLK	AC22	I	**Audio Bit Clock.** The serial bit clock from the codec.	F_TRDY#
SDATA_OUT	AD22	O	**Serial Data Output.** This output transmits audio serial data to the codec.	CLKSEL2 (Strap)
SDATA_IN	AF22	I	**Serial Data Input.** This input receives serial data from the primary codec.	F_GNT0#
SYNC	AE22	O	**Serial Bus Synchronization.** This bit is asserted to synchronize the transfer of data between the SC1100 and the AC97 codec.	CLKSEL3 (Strap)
AC97_CLK	AC21	O	**Codec Clock.** It is twice the frequency of the Audio Bit Clock.	---
AC97_RST#	AF23	O	**Codec Reset.** S3 to S5 wakeup is not supported because AC97_RST# is powered by V_{IO}. If wakeup from states S3 to S5 are needed, a circuit in the system board should be used to reset the AC97 codec.	F_STOP
PC_BEEP	AE18	O	**PC Beep.** Legacy PC/AT speaker output.	GPIO16+IRRX1+ F_DEVSEL#

2.4.11 Power Management Interface Signals

Signal Name	Ball No.	Type	Description	Mux
GPWIO0	AC15	I/O	**General Purpose Wakeup I/Os.** These signals each have an internal pull-up of 100 KΩ.	---
GPWIO1	AE16	I/O		---
GPWIO2	AF17	I/O		---
ONCTL#	AD15	O	**On / Off Control.** This signal indicates to the main power supply that power should be turned on. This signal is an open-drain output.	---

Signal Definitions (Continued)

2.4.11 Power Management Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
PWRBTN#	AF15	I	**Power Button.** An input used by the power management logic to monitor external system events, most typically a system on/off button or switch. This signal has an internal pull-up of 100 KΩ, a Schmitt-trigger input buffer and debounce protection of at least 16 ms. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low. **Note:** This signal can be pulled down internally by software control. This is done by enabling internal GPIO63 to be an output, for at least 16 ms (F0BAR0+I/O Offset 20h and 24h).	---
PWRCNT1	AE17	O	**Suspend Power Plane Control 1 and 2.** Control signals asserted during power management Suspend states. These signals are open-drain outputs.	---
PWRCNT2	AF18	O		---
THRM#	AE15	I	**Thermal Event.** An active low signal generated by external hardware indicating that the system temperature is too high.	---

2.4.12 GPIO Interface Signals

Name	Ball No.	Type	Description	Mux
GPIO0	B22	I/O	**GPIO Port 0.** Each signal is configured independently as an input or I/O, with or without static pull-up, and with either open-drain or totem-pole output type. A debouncer and an interrupt can be enabled or masked for each of signals GPIO[0:3] and GPIO[6:15] independently.	TRDE#
GPIO1	AD24			FPCICLK
GPIO2	B21			F5BAR4CS#
GPIO3	A22			F5BAR5CS#
GPIO6	AD12			DTR#/BOUTIDE_IOR1#+INTR_O
GPIO7	AF11			RTS#+IDE_DACK1#+F_C/BE3#
GPIO8	AC12			CTS#+IDE_DREQ1+SMI_O
GPIO9	AE12			DCD#+IDE_IOW1#+F_IRDY#
GPIO10	AF12			DSR#+IDE_IORDY1+F_FRAME#
GPIO11	AF19			RI#+IRQ15
GPIO12	AE23			AB2C+F_AD3
GPIO13	AD23			AB2D+F_AD4
GPIO14	D22			IOCS1#+IOR#+DOCR#
GPIO15	C22			IOW#+DOCW#
GPIO16	AE18			PC_BEEP+IRRX1+F_DEVSEL#
GPIO17	B23			IOCS0#+IOCHRDY
GPIO18	AC24			F_AD0
GPIO19	Y24			INTC#
GPIO20	D21			DOCCS#

Signal Definitions (Continued)

2.4.12 GPIO Interface Signals

Name	Ball No.	Type	Description	Mux
GPIO32	E26	I/O	**GPIO Port 1.** Same as Port 0. A debouncer and an interrupt can be enabled or masked for each of signals GPIO[32:41] and GPIO47 independently.	LAD0
GPIO33	D25			LAD1
GPIO34	D26			LAD2
GPIO35	C25			LAD3
GPIO36	C26			LDRQ#
GPIO37	B24			LFRAME#
GPIO38	AB23			F_AD5
GPIO39	A24			SERIRQ
GPIO40	B20			SDTEST5+F_AD6
GPIO41	AA24			TEST0+F_C/BE0#
GPIO47	AB24			F_AD7

2.4.13 Debug Monitoring Interface Signals

Signal Name	Ball No.	Type	Description	Mux
FPCICLK	AD24	O	**Fast-PCI Bus Monitoring Signals.** When enabled, this group of signals provides monitoring of the internal Fast-PCI bus for debug purposes. To enable, pull up FPCI_MON (ball AB25).	GPIO1
F_AD7	AB24	O		GPIO47
F_AD6	B20	O		GPIO40+SDTEST5#
F_AD5	AB23	O		GPIO38
F_AD4	AD23	O		AB2D+F_AD4
F_AD3	AE23	O		AB2C+F_AD3
F_AD2	Y2	O		AB1D
F_AD1	Y1	O		AB1C
F_AD0	AC24	O		GPIO18
F_C/BE3#	AF11	O		GPIO7+RTS#+IDE_DACK1#
F_C/BE2#	AE11	O		SOUT
F_C/BE1#	AD11	O		SIN
F_C/BE0#	AA24	O		GPIO41+TEST0
F_FRAME#	AF12	O		GPIO10+DSR#+IDE_IORDY1
F_IRDY#	AE12	O		GPIO9+DCD#+IDE_IOW1#
F_STOP#	AF23	O		AC97_RST#
F_DEVSEL#	AE18	O		GPIO16+PC_BEEP+IRRX1
F_GNT0#	AF22	O		SDATA_IN
F_TRDY#	AC22	O		BIT_CLK
INTR_O	AD12	O	**CPU Core Interrupt.** When enabled, this signal provides for monitoring of the internal GX1 core INTR signal for debug purposes. To enable, pull up FPCI_MON (ball AB25).	GPIO6+DTR#/BOUT+IDE_IOR1#

Signal Definitions (Continued)

2.4.13 Debug Monitoring Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
SMI_O	AC12	O	**System Management Interrupt.** This is the input to the GX1 core. When enabled, this signal provides for monitoring of the internal GX1 core SMI# signal for debug purposes. To enable, pull up FPCI_MON (ball AB25).	GPIO8+CTS#+IDE_DREQ1

2.4.14 JTAG Interface Signals

Signal Name	Ball No.	Type	Description	Mux
TCK	AE20	I	**JTAG Test Clock.** This signal has an internal weak pull-up resistor.	---
TDI	AF20	I	**JTAG Test Data Input.** This signal has an internal weak pull-up resistor.	---
TDO	AD20	O	**JTAG Test Data Output**	---
TMS	AF21	I	**JTAG Test Mode Select.** This signal has an internal weak pull-up resistor.	---
TRST#	AC20	I	**JTAG Test Reset.** This signal has an internal weak pull-up resistor. For normal JTAG operation, this signal should be active at power-up. If the JTAG interface is not being used, this signal can be tied low.	---

2.4.15 Test and Measurement Interface Signals

Signal Name	Ball No.	Type	Description	Mux
FMUL3B	AD10	I/O	**FMUL3 Bypass.** This signal is used for internal testing only. For normal operation, leave this signal unconnected.	TEST1
GXCLK	C20	O	**GX Clock.** This signal is used for internal testing only. For normal operation, program as IRTX or leave unconnected.	IRTX+TEST3
PLL5B	AC11	I/O	**PLL5 Bypass.** This signal is for internal testing only. For normal operation, leave this signal unconnected.	TEST2
TEST3	C20	O	**Internal Test Signals.** These signals are used for internal testing only. For normal operation, leave these signals unconnected unless programmed as one of their muxed options.	GXCLK+IRTX
TEST2	AC11	O		PLL5B
TEST1	AD10	O		FMUL3B
TEST0	AA24	O		GPIO41+ F_C/BE0#
SDTEST5	B20	O	**Memory Internal Test Signals.** These signals are used for internal testing only. For normal operation, leave SDTEST[4:1] unconnected and program SDTEST5 to function as GPIO40.	GPIO40+F_AD6
SDTEST4	E3	O		---
SDTEST3	D5	O		---
SDTEST2	A18	O		---
SDTEST1	D1	O		---
SDTEST0	A20	O		---

Signal Definitions (Continued)

2.4.15 Test and Measurement Interface Signals (Continued)

Signal Name	Ball No.	Type	Description	Mux
GTEST	AD21	I	**Global Test.** This signal is used for internal testing only. For normal operation, tie this signal low.	---

2.4.16 Power[1], Ground and No Connections

Signal Name	Ball No.	Type	Description
AV_{SSPLL}	AF9	GND	**Analog PLL Ground Connection**
AV_{CCUSB}	AF6	PWR	**3.3V Analog USB Power Connection** (Low noise power)
AV_{SSUSB}	AD9	GND	**Analog USB Ground Connection**
V_{BAT}	W3	PWR	**Battery.** Provides battery back-up to the RTC and ACPI registers, when V_{SB} is lower than the minimum value (see Table 7-2 on page 281). The ball is connected to the internal logic through a series resistor for UL protection.
V_{CORE}	Refer to Table 2-3 on page 27 (Total of 24)	PWR	**1.8V or 2.0V Core Processor Power Connections**
V_{IO}	Refer to Table 2-3 on page 27 (Total of 28)	PWR	**3.3V I/O Power Connections**
V_{PLL}	AE9	PWR	**3.3V PLL2 Analog Power Connection** (Low noise power)
V_{SB}	AF16	PWR	**3.3V Standby Power Supply.** Provides power to the RTC and ACPI circuitry while the main power supply is turned off.
V_{SBL}	AD16	PWR	**1.8V, 1.9V or 2.0V Standby Power Supply.** Provides power to the internal logic while the main power supply is turned off. This signal requires a 0.1 µF bypass capacitor to V_{SS}. This supply must be present when V_{SB} is present.
V_{SS}	Refer to Table 2-3 on page 27 (Total of 60)	GND	**Ground Connections**
NC	A21, A23, D20, AC23, AE24, AF24	---	**No Connections.** For normal operation, leave these signals unconnected.

1. All power sources must be connected, even if the corresponding function is not used

3.0 General Configuration Block

The General Configuration Block (GCB) includes registers for:

- Pin Multiplexing and Miscellaneous Configuration
- WATCHDOG Timer
- High-Resolution Timer
- Clock Generators

A selectable interrupt is shared by all these functions.

3.1 CONFIGURATION BLOCK ADDRESSES

Registers of the GCB are mapped in a 64-byte address range which is mapped to I/O space. These registers are physically connected to the internal Fast-PCI bus, but do not have a register block in PCI configuration space (i.e., they do not appear to software as PCI registers).

After system reset, the Base Address register is located at I/O address 02EAh. This address can be used only once. Before accessing any PCI registers, the BOOT code must program this 16-bit register to the I/O base address for the General Configuration Block registers. All subsequent writes to this address, are ignored until system reset.

Note: Location of the General Configuration Block cannot be determined by software. See the *SC1100 Information Appliance On a Chip device errata* document.

Reserved bits in the General Configuration block should read as written unless otherwise specified.

Table 3-1. Configuration Space Register Map

Offset	+0	+1	+2	+3
00h	WATCHDOG Timeout (WDTO)		WATCHDOG Configuration (WDCNFG)	
04h	WATCHDOG Status (WDSTS)	Reserved		
08h	TIMER Value (TMVALUE)			
0Ch	TIMER Status (TMSTS)	TIMER Config (TMCNFG)	Reserved	
10h	Maximum Core Clock Multiplier (MCCM)	Reserved	PLL Power Control (PPCR)	Reserved
14h	Reserved			
18h	Clocks: PLL3C			
1Ch	Reserved	Reserved	Core Clock Frequency (CCFC)	
20h	Reserved			
24h				
28h				
2Ch				
30h	Pin Multiplexing (PMR)			
34h	Miscellaneous Configuration (MCR)			
38h	Interrupt Selection (INTSEL)	Reserved		
3Ch	IA On a Chip ID (IID)	Revision (REV)	Configuration Base Address (CBA)	

General Configuration Block (Continued)

3.2 MULTIPLEXING, INTERRUPT SELECTION, AND BASE ADDRESS REGISTERS

The registers described in the Table 3-2 are used to determine general configuration for the Geode SC1100. These registers also indicate which multiplexed signals are issued via balls from which more than one signal may be output. For more information about multiplexed signals and the appropriate configurations, see Section 2.1 "Ball Assignments" on page 17.

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td colspan="2">Offset 30h Pin Multiplexing Register - PMR (R/W) Reset Value: 00000000h
Width: DWORD
This register configures pins with multiple functions. See Section 2.1 "Ball Assignments" on page 17 for more information about multiplexing information.</td></tr>
<tr><td>31:30</td><td>Reserved. Always write 0.</td></tr>
<tr><td>29</td><td>Test Signals. Selects ball functions.
<table>
<tr><th></th><th colspan="2">0: Internal Test/Serial Port Signal</th><th colspan="2">1: Internal Test Signal</th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>AA24</td><td>GPIO41
F_C/BE0#</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td><td>TEST0
F_C/BE0#</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AD10</td><td>FMUL3B</td><td>None</td><td>TEST1</td><td>None</td></tr>
<tr><td>AC11</td><td>PLL5B</td><td>None</td><td>TEST2</td><td>None</td></tr>
<tr><td>C20</td><td>IRTX+GXCLK</td><td>See PMR[6]</td><td>TEST3</td><td>PMR[6] = 1</td></tr>
</table></td></tr>
<tr><td>28</td><td>Test Signals. Selects ball function.
<table>
<tr><th></th><th colspan="2">0: GPIO Signal</th><th colspan="2">1: Internal Test Signal</th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>B20</td><td>GPIO40
F_AD6</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td><td>SDTEST5
F_AD6</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td></tr>
</table></td></tr>
<tr><td>27</td><td>FPCI_MON (Fast-PCI Monitoring). Fast-PCI monitoring output signals can be enabled in two ways: by setting this bit to 1 or by strapping FPCI_MON (ball AB25) high. (The strapped value can be read back at MCR[30].) Listed below is how these two options work together and the signals that are enabled (enabling overrides add’l dependencies).
<table>
<tr><th>PMR[27]</th><th>FPCI_MON</th><th></th></tr>
<tr><td>0</td><td>0</td><td>Disable Fast-PCI monitoring signals</td></tr>
<tr><td>0</td><td>1</td><td>Enable Fast-PCI monitoring signals</td></tr>
<tr><td>1</td><td>0</td><td>Enable Fast-PCI monitoring signals</td></tr>
<tr><td>1</td><td>1</td><td>Enable Fast-PCI monitoring signals</td></tr>
</table>
<table>
<tr><th>Ball #</th><th>FPCI_MON Signal</th><th>Other Signal</th><th>Add’l Dependencies</th></tr>
<tr><td>AD24</td><td>FPCICLK</td><td>GPIO1</td><td>None</td></tr>
<tr><td>AB24</td><td>F_AD7</td><td>GPIO47</td><td>None</td></tr>
<tr><td>B20</td><td>F_AD6</td><td>GPIO40+SDTEST5</td><td>See PMR[28]</td></tr>
<tr><td>AB23</td><td>F_AD5</td><td>GPIO38</td><td>None</td></tr>
<tr><td>AD23</td><td>F_AD4</td><td>GPIO13+AB2D</td><td>See PMR[19]</td></tr>
<tr><td>AE23</td><td>F_AD3</td><td>GPIO12+AB2C</td><td>See PMR[19]</td></tr>
<tr><td>Y2</td><td>F_AD2</td><td>AB1D</td><td>None</td></tr>
<tr><td>Y1</td><td>F_AD1</td><td>AB1C</td><td>None</td></tr>
<tr><td>AC24</td><td>F_AD0</td><td>GPIO18</td><td>None</td></tr>
<tr><td>AF11</td><td>F_C/BE3#</td><td>GPIO7+RTS#+IDE_DACK1#</td><td>See PMR[17]</td></tr>
<tr><td>AE11</td><td>F_C/BE2#</td><td>SOUT</td><td>None</td></tr>
<tr><td>AD11</td><td>F_C/BE1#</td><td>SIN</td><td>None</td></tr>
<tr><td>AA24</td><td>F_C/BE0#</td><td>GPIO41+TEST0</td><td>See PMR[29]</td></tr>
<tr><td>AF12</td><td>F_FRAME#</td><td>GPIO10+DSR#+IDE_IORDY1</td><td>See PMR[18]</td></tr>
<tr><td>AE12</td><td>F_IRDY#</td><td>GPIO9+DCD#+IDE_IOW1#</td><td>See PMR[18]</td></tr>
<tr><td>AE18</td><td>F_DEVSEL#</td><td>GPIO16+IRRX1+PC_BEEP</td><td>See PMR[6]</td></tr>
<tr><td>AF23</td><td>F_STOP#</td><td>AC97_RST#</td><td>None</td></tr>
<tr><td>AF22</td><td>F_GNT0#</td><td>SDATA_IN</td><td>None</td></tr>
<tr><td>AC22</td><td>F_TRDY#</td><td>BIT_CLK</td><td>None</td></tr>
<tr><td>AE18</td><td>F_DEVSEL#</td><td>GPIO16+PC_BEEP+IRRX1</td><td>See PMR[6]</td></tr>
<tr><td>AD12</td><td>INTR_O</td><td>GPIO6+DTR#/BOUTIDE_IOR1#</td><td>See PMR[18]</td></tr>
<tr><td>AC12</td><td>SMI_O</td><td>GPIO8+CTS#+IDE_DREQ1</td><td>See PMR[17]</td></tr>
</table></td></tr>
<tr><td>26</td><td>Reserved: Always write 0.</td></tr>
</table>

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>25</td><td>AC97CKEN (Enable AC97_CLK Output). This bit enables the output drive of AC97_CLK (ball AC21).
0: AC97_CLK output is HiZ.
1: AC97_CLK output is enabled.</td></tr>
<tr><td>24:22</td><td>Reserved: Always write 0.</td></tr>
<tr><td>21</td><td>GPSEL1 (Select Functions/Commands). Selects ball functions.
<table>
<tr><th></th><th>0: GPIO/IO Command Signal</th><th></th><th>1: DOC/NAND Signal</th><th></th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>D22</td><td>GPIO14
IOCS1#</td><td>PMR[2] = 0
PMR[2] = 1</td><td>IOR#
DOCR#</td><td>PMR[2] = 0
PMR[2] = 1</td></tr>
<tr><td>C22</td><td>IOW#
GPIO15</td><td>PMR[2] = 0
PMR[2] = 1</td><td>IOW#
DOCR#</td><td>PMR[2] = 0
PMR[2] = 1</td></tr>
</table></td></tr>
<tr><td>20</td><td>Reserved. Must be set to 0.</td></tr>
<tr><td>19</td><td>AB2SEL (Select ACCESS.bus 2). Selects ball functions.
<table>
<tr><th></th><th>0: GPIO Signal</th><th></th><th>1: ACCESS.bus 2 Signal</th><th></th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>AE23</td><td>GPIO12
F_AD3</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td><td>AB2C
F_AD3</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AD23</td><td>GPIO13
F_AD4</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td><td>AB2D
F_AD4</td><td>PMR[27] and FPCI_MON = 0
PMR[27] or FPCI_MON = 1</td></tr>
</table></td></tr>
<tr><td>18</td><td>SPSEL (Select SP Functions). Selects ball functions.
<table>
<tr><th></th><th>0: GPIO/IDE Signal</th><th></th><th>1: Serial Port Signal</th><th></th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>AD12</td><td>GPIO6
IDE_IOR1#
INTR_O</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td><td>DTR#/BOUT
Undefined
INTR_O</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AE12</td><td>GPIO9
IDE_IOW1#
F_IRDY#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td><td>DCD#
Undefined
F_IRDY#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AF12</td><td>GPIO10
IDE_IORDY1
F_FRAME#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td><td>DSR#
Undefined
F_FRAME#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AF19</td><td>GPIO11
IRQ15</td><td>PMR[8] = 0
PMR[8] = 1</td><td>RI#
Undefined</td><td>PMR[8] = 0
PMR[8] = 1</td></tr>
</table>
Note: PMR[27] and FPCI_MON = 0.</td></tr>
<tr><td>17</td><td>SPCRSEL (Select SP2 Flow Control). Selects ball functions.
<table>
<tr><th></th><th>0: GPIO/IDE Signal</th><th></th><th>1: Serial Port Signal</th><th></th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>AF11</td><td>GPIO7
IDE_DACK1#
F_C/BE3#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td><td>RTS
Undefined
F_C/BE3#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td></tr>
<tr><td>AC12</td><td>GPIO8
IDE_DREQ1
SMI_O</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td><td>CTS#
Undefined
F_C/BE3#</td><td>PMR[8] = 0 and Note
PMR[8] = 1 and Note
PMR[27] or FPCI_MON = 1</td></tr>
</table>
Note: PMR[27] and FPCI_MON = 0.</td></tr>
<tr><td>16</td><td>SERISEL (Select SERIRQ Function). Selects ball function.
<table>
<tr><th></th><th>0: GPIO Signal</th><th></th><th>1: Serial IRQ Signal</th><th></th></tr>
<tr><th>Ball #</th><th>Name</th><th>Add’l Dependencies</th><th>Name</th><th>Add’l Dependencies</th></tr>
<tr><td>A24</td><td>GPIO39</td><td>None</td><td>SERIRQ</td><td>None</td></tr>
</table></td></tr>
<tr><td>15</td><td>Reserved: Always write 0.</td></tr>
</table>

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
14	**LPCSEL (Select LPC Bus).** Selects ball functions. **0: GPIO Signal** / **1: LPC Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** E26 / GPIO32 / None / LAD0 / None D25 / GPIO33 / None / LAD1 / None D26 / GPIO34 / None / LAD2 / None C25 / GPIO35 / None / LAD3 / None C26 / GPIO36 / None / LDRQ# / None B24 / GPIO37 / None / LFRAME# / None
13	**Reserved:** Write as read.
12	**TRDESEL (Select TRDE#).** Selects ball function. **0: Sub-ISA Signal** / **1: GPIO Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** B22 / TRDE# / None / GPIO0 / None
11:10	**Reserved:** Write as read.
9	**IOCHRDY (Select IOCHRDY).** Selects function for ball B23. **0: GPIO/Sub-ISA Signal** / **1: Sub-ISA Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** B23 / GPIO17 / PMR[5] = 0 / IOCHRDY / PMR[5] = 1 IOCS0# / PMR[5] = 1 / Undefined / PMR[5] = 0
8	**IDE1SEL (Select IDE Channel 1).** Selects ball functions. Works in conjunction with PMR[18] and PMR[17]. See PMR[18] and PMR[17] for definitions.
7	**DOCCSSEL (Select DOCCS#).** Selects ball function. **0: GPIO Signal** / **1: Sub-ISA Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** D21 / GPIO20 / None / DOCCS# / None
6	**Internal Test Signals 0.** Selects functions for balls AE18 and C20. **0: Serial Port Signal** / **1: GPIO/Audio/Internal Test Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** AE18 / IRRX1 / Note / GPIO16 / PMR[0] = 0 and Note PC_BEEP / PMR[0] = 1 and Note F_DEVSEL# / PMR[27] or FPCI_MON = 1 / F_DEVSEL# / PMR[27] or FPCI_MON = 1 C20 / IRTX / PMR[29] = 0 / GXCLK / PMR[29] = 0 Undefined / PMR[29] = 1 / TEST3 / PMR[29] = 1 **Note:** PMR[27] and FPCI_MON = 0.
5	**IOCS0SEL (Select IOCS0#).** Selects ball function. Works in conjunction with PMR[9]. See PMR[9] description.
4	**INTCSEL (Select INTC#).** Selects ball function. **0: GPIO Signal** / **1: PCI Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** Y24 / GPIO19 / None / INTC# / None
3	**F5BAR5SEL (Select F5BAR5CS#).** Selects ball function for ball A22. **0: GPIO Signal** / **1: Sub-ISA Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** A22 / GPIO3 / None / F5BAR5CS# / None
2	**GPSEL0 (Select Functions/Commands).** Works in conjunction with PMR[21]. See PMR[21] for definition.
1	**F5BAR4SEL (Select F5BAR4CS#).** Selects function for ball B21. **0: GPIO Signal** / **1: Sub-ISA Signal** **Ball #** / **Name** / **Add'l Dependencies** / **Name** / **Add'l Dependencies** B21 / GPIO2 / None / F5BAR4CS# / None
0	**PCBEEPSEL (Select PC_BEEP)**. Works in conjunction with PMR[6]. See PMR[6] for definition.

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
Offset 34h **Miscellaneous Configuration Register - MCR (R/W)** **Reset Value: 0000001h** Width: DWORD Power-on reset value: The BOOT16 strap pin selects "Enable 16-Bit Wide Boot Memory".	
31	**Reserved.** Always write 0.
30	**FPCI_MON (Ball AB25) Strap Status. (Read Only)** Represents the value of the strap that is latched after power-on reset. Indicates if Fast-PCI monitoring output signals are enabled via strapping option. 0: Disable (FPCI_MON = 0). 1: Enable (FPCI_MON = 1). In addition to the strapping option, Fast-PCI monitoring output signals can also be enabled via PMR[27]. See PMR[27] bit description for a list of signals that are enabled when either option is used.
29:16	**Reserved.** Always write 0.
15	**F5B5_16 (Enable 16-Bit Wide F5BAR5CS# Access).** Enables the 16-line access to F5BAR5CS# in the Sub-ISA interface. 0: 8-bit wide F5BAR5CS# access is used. 1: 16-bit wide F5BAR5CS# access is used.
14	**IBUS16 (Invert BUS16).** This bit inverts the meaning of MCR[3] (bit 3 of this register). 0: BUS16 is as described for MCR[3]. 1: BUS16 meaning is inverted: if MCR[3] = 0, ROMCS# access is 16 bits wide; if MCR[3] = 1, ROMCS# access is 8 bits wide.
13	**F5B4ZWS (Enable ZWS for F5BAR4CS# Access).** Enables internal activation of ZWS# (Zero Wait States) control for F5BAR4CS# access. 0: ZWS is not active for F5BAR4CS# access. 1: ZWS is active for F5BAR4CS# access.
12	**IO1ZWS (Enable ZWS# for IOCS1# Access).** Enables internal activation of ZWS# (Zero Wait States) control for IOCS1# access. 0: ZWS# is not active for IOCS1# access. 1: ZWS# is active for IOCS1# access.
11	**IO0ZWS (Enable ZWS# for IOCS0# Access).** Enables internal activation of ZWS# (Zero Wait States) control for IOCS0# access. 0: ZWS# is not active for IOCS0# access. 1: ZWS# is active for IOCS0# access.
10	**DOCZWS (Enable ZWS# for DOCCS# Access).** Enables internal activation of ZWS# (Zero Wait States) control for DOCCS# access. 0: ZWS# is not active for DOCCS# access. 1: ZWS# is active for DOCCS# access.
9	**ROMZWS (Enable ZWS# for ROMCS# Access).** Enables internal activation of ZWS# (Zero Wait States) control for ROMCS# access. 0: ZWS# is not active for ROMCS# access. 1: ZWS# is active for ROMCS# access.
8	**IO1_16 (Enable 16-Bit Wide IOCS1# Access).** Enables the16-line access to IOCS1# in the Sub-ISA interface. 0: 8-bit wide IOCS1# access is used. 1: 16-bit wide IOCS1# access is used.
7	**IO0_16 (Enable 16-Bit Wide IOCS0# Access).** Enables the 16-line access to IOCS0# in the Sub-ISA interface. 0: 8-bit wide IOCS0# access is used. 1: 16-bit wide IOCS0# access is used.
6	**DOC16 (Enable 16-Bit Wide DOCCS# Access).** Enables the 16-line access to DOCCS# in the Sub-ISA interface. 0: 8-bit wide DOCCS# access is used. 1: 16-bit wide DOCCS# access is used.
5	**F5B4_16 (Enable 16-Bit Wide F5BAR4CS# Access).** Enables the 16-line access to F5BAR4CS# in the Sub-ISA interface. 0: 8-bit wide F5BAR4CS# access is used. 1: 16-bit wide F5BAR4CS# access is used.

General Configuration Block (Continued)

Table 3-2. Multiplexing, Interrupt Selection, and Base Address Registers (Continued)

Bit	Description
4	**IRTXEN (Infrared Transmitter Enable).** This bit enables the drive of Infrared transmitter's output. 0: IRTX+GXCLK+TEST3 line (ball C20) is HiZ. 1: IRTX+GXCLK+TEST3 line (ball C20) is enabled.
3	**BUS16 (16-Bit Wide Boot Memory). (Read Only)** This bit reports the status of the BOOT16 strap (ball C23). If the BOOT16 strap is pulled high, at reset 16-bit access to ROM in the Sub-ISA interface is enabled. MCR[14] = 1 inverts the meaning of this register, however, this bit reflects the value of the BOOT16 strap regardless of the setting of MCR[14]. 0: 8-bit wide ROM. 1: 16-bit wide ROM.
2	**F5B5ZWS (Enable ZWS for F5BAR5CS# Access).** Enables internal activation of ZWS# (Zero Wait States) control for F5BAR5CS# access. 0: ZWS is not active for F5BAR5CS# access. 1: ZWS is active for F5BAR5CS# access.
1	**Reserved.** Write as read.
0	**SDBE0 (Slave Disconnect Boundary Enable).** Works in conjunction with the GX1 module's PCI Control Function 2 Register (Index 41h), bit 1 (SDBE1). Sets boundaries for when the GX1 module is a PCI slave. SDBE[1:0] 00: Read and Write disconnect on boundaries set by bits [3:2] of the GX1 module's PCI Control Function 2 register (Index 41h). 01: Write disconnects on boundaries set by bits [3:2] of the GX1 module's PCI Control Function 2 register. Read disconnects on cache line boundary of 16 bytes. 1x: Read and Write disconnect on cache line boundary of 16 bytes. This bit is reset to 1. All PCI bus masters (including SC1100's on-chip PCI bus masters, e.g., the USB Controller) must be disabled while modifying this bit. When accessing this register while any PCI bus master is enabled, use read-modify-write to ensure this bit contents is unchanged. **Note:** When Slave Disconnect Boundary is disabled for Write, the cache should use Write Through Mode instead of Write Back Mode. The Write Through Mode implies some overall performance degradation since all Writes go to Memory. If the Write back Mode is used in this case, the cache coherency cannot be guaranteed.

Offset 38h **Interrupt Selection Register - INTSEL (R/W)** **Reset Value: 00h**
Width: Byte

This register selects the IRQ signal of the combined WATCHDOG and High-Resolution Timer interrupt. This interrupt is shareable with other interrupt sources.

7:4	**Reserved.** Write as read.			
3:0	**CBIRQ.** Configuration Block Interrupt.			
	0000: Disable	0100: IRQ4	1000: IRQ8#	1100: IRQ12
	0001: IRQ1	0101: IRQ5	1001: IRQ9	1101: Reserved
	0010: Reserved	0110: IRQ6	1010: IRQ10	1110: IRQ14
	0011: IRQ3	0111: IRQ7	1011: IRQ11	1111: IRQ15

Offset 3Ch **IA On a Chip Identification Number Register - IID (RO)** **Reset Value: 06h**
Width: Byte

This register identifies the IA On a Chip device.

Offset 3Dh **Revision Register - REV (RO)** **Reset Value: xxh**
Width: Byte

This register identifies the device revision. See device errata for value.

Offset 3Eh **Configuration Base Address Register - CBA (RO)** **Reset Value: xxxxh**
Width: WORD

This register sets the base address of the Configuration block.

15:6	**Configuration Base Address.** These bits are the high bits of the Configuration Base Address.
5:0	**Configuration Base Address.** These bits are the low bits of the Configuration Base Address. These bits are set to 0.

3.3 WATCHDOG

The SC1100 includes a WATCHDOG function to serve as a fail-safe mechanism in case the system becomes hung. When triggered, the WATCHDOG mechanism returns the system to a known state by generating an interrupt, an SMI, or a system reset (depending on configuration).

3.3.1 Functional Description

WATCHDOG is enabled when the WATCHDOG Timeout (WDTO) register (Offset 00h) is set to a non-zero value. The WATCHDOG timer starts with this value and counts down until either the count reaches 0, or a trigger event restarts the count (with the WDTO register value).

The WATCHDOG timer is restarted in any of the following cases:

- The WDTO register is set with a non-zero value.
- The WATCHDOG timer reaches 0 and the WATCHDOG Overflow bit (WDOVF, Offset 04h[0]) is 0.

The WATCHDOG function is disabled in any of the following cases:

- System reset occurs.
- The WDTO register is set to 0.
- The WDOVF bit is already 1 when the timer reaches 0.

3.3.1.1 WATCHDOG Timer

The WATCHDOG timer is a 16-bit down counter. Its input clock is a 32 KHz clock divided by a predefined value (see WDPRES field, Offset 02h[3:0]). The 32 KHz input clock is enabled when either:

- The GX1 module's internal SUSPA# signal is 1.

or

- The GX1 module's internal SUSPA# signal is 0 and the WD32KPD bit (Offset 02h[8]) is 0.

The 32 KHz input clock is disabled, when:

- The GX1 module's internal SUSPA# signal is 0 and the WD32KPD bit is 1.

For more information about signal SUSPA#, refer to the *GX1 Processor Series Datasheet*.

When the WATCHDOG timer reaches 0:

- If the WDOVF bit in the WDSTS register (Offset 04h[0]) is 0, an interrupt, an SMI or a system reset is generated, depending on the value of the WDTYPE1 field in the WDCNFG register (Offset 02h[5:4]).
- If the WDOVF bit in the WDSTS register is already 1 when the WATCHDOG timer reaches 0, an interrupt, an SMI or a system reset is generated according to the WDTYPE2 field (Offset 02h[7:6]), and the timer is disabled. The WATCHDOG timer is re-enabled when a non-zero value is written to the WDTO register (Offset 00h).

The interrupt or SMI is deasserted when the WDOVF bit is set to 0. The reset generated by the WATCHDOG functions is used to trigger a system reset via the Core Logic module. The value of the WDOVF bit, the WDTYPE1 field, and the WDTYPE2 field are not affected by a system reset (except when generated by power-on reset).

The SC1100 also allows no action to be taken when the timer reaches 0 (according to WDTYPE1 field and WDTYPE2 field). In this case only the WDOVF bit is set to 1.

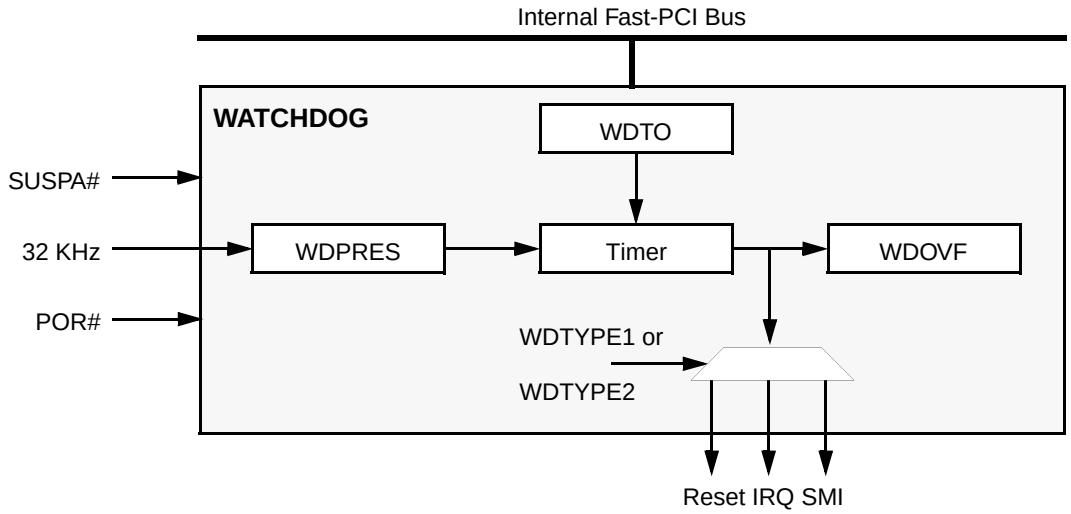


Figure 3-1. WATCHDOG Block Diagram

General Configuration Block (Continued)

WATCHDOG Interrupt

The WATCHDOG interrupt (if configured and enabled) is routed to an IRQ signal. The IRQ signal is programmable via the INTSEL register (Offset 38h, described in Table 3-2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50). The WATCHDOG interrupt is a shareable, active low, level interrupt.

WATCHDOG SMI

The WATCHDOG SMI is recognized by the Core Logic module as internal input signal EXT_SMI0#. To use the WATCHDOG SMI, Core Logic registers must be configured appropriately.

3.3.2 WATCHDOG Registers

Table 3-3 describes the WATCHDOG registers.

3.3.2.1 Usage Hints

- SMM code should set bit 8 of the WDCNFG register to 1 when entering ACPI C3 state, if the WATCHDOG timer is to be suspended. If this is not done, the WATCHDOG timer is functional during C3 state.
- SMM code should set bit 8 of the WDCNFG register to 1, when entering ACPI S1 and S2 states if the WATCHDOG timer is to be suspended. If this is not done, the WATCHDOG timer is functional during S1 and S2 states.

Table 3-3. WATCHDOG Registers

Bit	Description
Offset 00h **WATCHDOG Timeout Register - WDTO (R/W)** **Reset Value: 0000h** Width: WORD This register specifies the programmed WATCHDOG timeout period.	
15:0	Programmed timeout period.
Offset 02h **WATCHDOG Configuration Register - WDCNFG (R/W)** **Reset Value: 0000h** Width: WORD This register selects the signal to be generated when the timer reaches 0, whether or not to disable the 32 KHz input clock during low power states, and the prescaler value of the clock input.	
15:9	**Reserved.** Write as read.
8	**WD32KPD (WATCHDOG 32 KHz Power Down).** 0: 32 KHz clock is enabled. 1: 32 KHz clock is disabled, when the GX1 module asserts its internal SUSPA# signal. This bit is cleared to 0, when POR# is asserted or when the GX1 module deasserts its internal SUSPA# signal (i.e., on SUSPA# rising edge). See Section 3.3.2.1 "Usage Hints" on page 56.
7:6	**WDTYPE2 (WATCHDOG Event Type 2)**. 00: No action 01: Interrupt 10: SMI 11: System reset This field is reset to 0 when POR# is asserted. Other system resets do not affect this field.
5:4	**WDTYPE1 (WATCHDOG Event Type 1).** 00: No action 01: Interrupt 10: SMI 11: System reset This field is reset to 0 when POR# is asserted. Other system resets do not affect this field.
3:0	**WDPRES (WATCHDOG Timer Prescaler).** Divide 32 KHz by: 0000: 1 0100: 16 1000: 256 1100: 4096 0001: 2 0101: 32 1001: 512 1101: 8192 0010: 4 0110: 64 1010: 1024 1110: Reserved 0011: 8 0111: 128 1011: 2048 1111: Reserved

Table 3-3. WATCHDOG Registers (Continued)

Bit	Description
Offset 04h **WATCHDOG Status Register - WDSTS (R/WC)** **Reset Value: 00h** Width: Byte This register contains WATCHDOG status information.	
7:4	**Reserved.** Write as read.
3	**WDRST (WATCHDOG Reset Asserted). (Read Only)** This bit is set to 1 when WATCHDOG Reset is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.
2	**WDSMI (WATCHDOG SMI Asserted). (Read Only)** This bit is set to 1 when WATCHDOG SMI is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.
1	**WDINT (WATCHDOG Interrupt Asserted). (Read Only)** This bit is set to 1 when the WATCHDOG Interrupt is asserted. It is set to 0 when POR# is asserted, or when the WDOVF bit is set to 0.
0	**WDOVF (WATCHDOG Overflow).** This bit is set to 1 when the WATCHDOG Timer reaches 0. It is set to 0 when POR# is asserted, or when a 1 is written to this bit by software. Other system reset sources do not affect this bit.

3.4 HIGH-RESOLUTION TIMER

The SC1100 provides an accurate time value that can be used as a time stamp by system software. This time is called the High-Resolution Timer. The length of the timer value can be extended via software. It is normally enabled while the system is in the C0 and C1 states. Optionally, software can be programmed to enable use of the High-Resolution Timer during C3 state and/or S1 state as well. In all other power states the High-Resolution Timer is disabled.

3.4.1 Functional Description

The High-Resolution Timer is a 32-bit free-running count-up timer that uses the oscillator clock or the oscillator clock divided by 27. Bit TMCLKSEL of the TMCNFG register (Offset 0Dh[1]) can be set via software to determine which clock should be used for the High-Resolution Timer.

When the most significant bit (bit 31) of the timer changes from 1 to 0, bit TMSTS of the TMSTS register (Offset 0Ch[0]) is set to 1. When both bit TMSTS and bit TMEN (Offset 0Dh[0]) are 1, an interrupt is asserted. Otherwise, the interrupt is deasserted. This interrupt enables software emulation of a larger timer.

The High-Resolution Timer interrupt is routed to an IRQ signal. The IRQ signal is programmable via the INTSEL register (Offset 38h). For more information about this register, see section Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50.

System software uses the read-only TMVALUE register (Offset 08h[31:0]) to read the current value of the timer. The TMVALUE register has no default value.

The input clock (derived from the 27 MHz crystal oscillator) is enabled when:

- The GX1 module's internal SUSPA# signal is 1.

or

- The GX1 module's internal SUSPA# signal is 0 and bit TM27MPD (Offset 0Dh[2]) is 0.

The input clock is disabled, when the GX1 module's internal SUSPA# signal is 0 and the TM27MPD bit is 1.

For more information about signal SUSPA# see Section 3.4.2.1 "Usage Hints" and the *GX1 Processor Series Datasheet*.

The High-Resolution Timer function resides on the internal Fast-PCI bus and its registers are in General Configuration Block address space. Only one complete register should be accessed at a *time (e.g., DWORD access should be used for DWORD wide registers and byte access should be used for byte-wide registers).

3.4.2 High-Resolution Timer Registers

Table 3-4 on page 58 describes the registers for the High-Resolution Timer.

3.4.2.1 Usage Hints

- SMM code should set bit 2 of the TMCNFG register to 1 when entering ACPI C3 state if the High-Resolution Timer should be disabled. If this is not done, the High-Resolution Timer is functional during C3 state.
- SMM code should set bit 2 of the TMCNFG register to 1 when entering ACPI S1 state if the High-Resolution Timer should be disabled. If this is not done, the High-Resolution Timer is functional during S1 state.

General Configuration Block (Continued)

Table 3-4. High-Resolution Timer Registers

Bit	Description
Offset: 08h **TIMER Value Register - TMVALUE (RO)** **Reset Value: xxxxxxxxh** Width: DWORD This register contains the current value of the High-Resolution Timer.	
31:0	**Current Timer Value.**
Offset: 0Ch **TIMER Status Register - TMSTS (R/W)** **Reset Value: 00h** Width: Byte This register supplies the High-Resolution Timer status information.	
7:1	**Reserved**
0	**TMSTS (TIMER Status).** This bit is set to 1 when the most significant bit (bit 31) of the timer changes from 1 to 0. It is cleared to 0 upon system reset or when 1 is written by software to this bit.
Offset: 0Dh **TIMER Configuration Register - TMCNFG (R/W)** **Reset Value: 00h** Width: Byte This register enables the High-Resolution Timer interrupt; selects the Timer clock; and disables the 27 MHz internal clock during low power states.	
7:3	**Reserved.**
2	**TM27MPD (TIMER 27 MHz Power Down).** 0: 27 MHz input clock is enabled. 1: 27 MHz input clock is disabled when the GX1 module asserts its internal SUSPA# signal. This bit is cleared to 0 when POR# is asserted or when the GX1 module deasserts its internal SUSPA# signal (i.e., on SUSPA# rising edge). See Section 3.4.2.1 "Usage Hints".
1	**TMCLKSEL (TIMER Clock Select).** 0: Count-up timer uses the oscillator clock divided by 27. 1: Count-up timer uses the oscillator clock, 27 MHz clock.
0	**TMEN (TIMER Interrupt Enable).** 0: High-Resolution Timer interrupt is disabled. 1: High-Resolution Timer interrupt is enabled.

General Configuration Block (Continued)

3.5 CLOCK GENERATORS AND PLLS

This section describes the registers for the clocks required by the GX1 and Core Logic modules, and how these clocks are generated. See Figure 3-2 for a clock generation diagram.

The clock generators are based on 32.768 KHz and 27.000 MHz crystal oscillators. The 32.768 KHz crystal oscillator is described in Section 4.5.2 "RTC Clock Generation" on page 80 (functional description of the RTC).

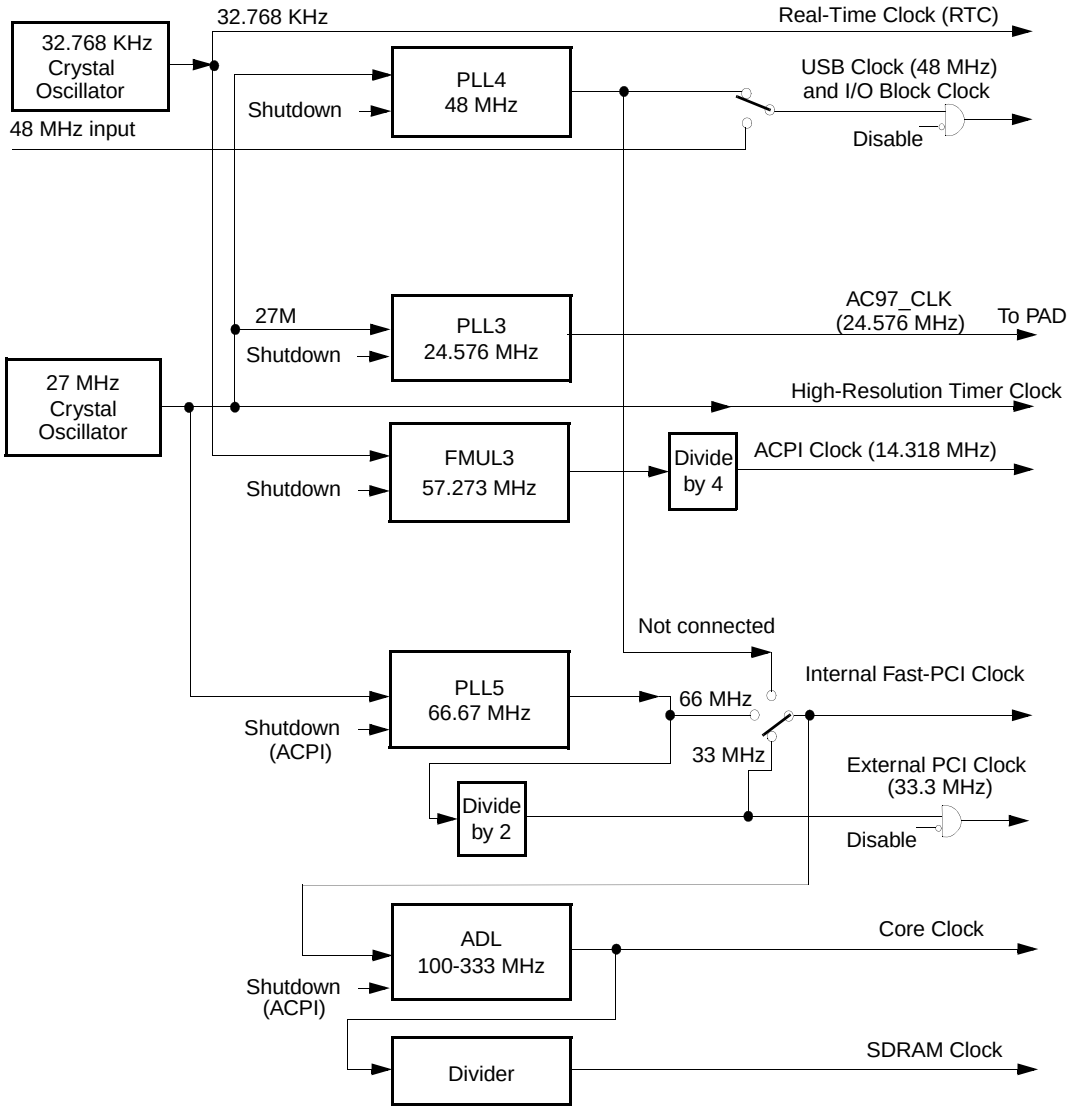


Figure 3-2. Clock Generation Block Diagram

General Configuration Block (Continued)

3.5.1 27 MHz Crystal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the X27I input (ball AF10) and the X27O output (ball AE10) signals. See Figure 3-3 for the recommended external circuit and Table 3-5 for a list of the circuit components.

Choose C_1 and C_2 capacitors to match the crystal's load capacitance. The load capacitance C_L "seen" by crystal Y is comprised of C_1 in series with C_2 and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 10 pF. The rule of thumb in choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

Example 1:

Crystal C_L = 10 pF, $C_{PARASITIC}$ = 8.2 pF
C_1 = 3.6 pF, C_2 = 3.6 pF

Example 2:

Crystal C_L = 20 pF, $C_{PARASITIC}$ = 8 pF
C_1 = 24 pF, C_2 = 24 pF

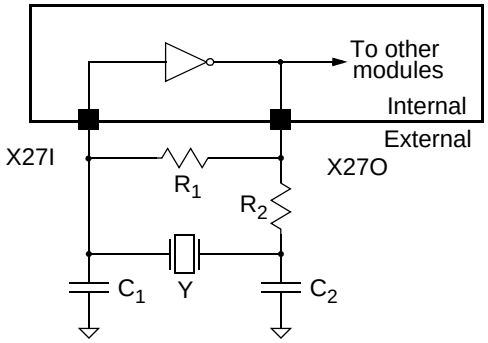


Figure 3-3. Recommended Oscillator External Circuitry

Table 3-5. Crystal Oscillator Circuit Components

Component	Parameters	Values	Tolerance
Crystal	Resonance Frequency	27.00 MHz Parallel Mode	50 PPM or better
	Type	AT-Cut or BT-cut	
	Serial Resistance	40 Ω	Max
	Shunt Capacitance	7 pF	Max
	Load Capacitance, C_L	10-20 pF	
	Temperature Coefficient	User-defined	
Resistor R_1	Resistance	20 MΩ	5%
Resistor R_2[1]	Resistance	100 Ω	5%
Capacitor C_1[1]	Capacitance	3-24 pF	5%
Capacitor C_2[1]	Capacitance	3-24 pF	5%

1. The value of these components is recommended. It should be tuned according to crystal and board parameters.

General Configuration Block (Continued)

3.5.2 GX1 Module Core Clock

The core clock is generated by an Analog Delay Loop (ADL) clock generator from the internal Fast-PCI clock. The clock can be any whole-number multiple of the input clock between 4 and 10. Possible values are listed in Table 3-6.

At power-on reset, the core clock multiplier value is set according to the value of four strapped pins - CLKSEL[3:0] (balls AE22, AD22, AD25, D23). These pins also select the clock which is used as input to the multiplier.

3.5.3 Internal Fast-PCI Clock

The internal Fast-PCI clock can be configured to 33 or 66 MHz via strap pins CLKSEL1 and CLKSEL0. These can be read in the internal Fast-PCI Clock field in the CCFC register (GCB+I/O Offset 1Eh[9:8]). (See Table 3-2 on page 50 details on the CCFC register.)

Table 3-6. Core Clock Frequency

ADL Multiplier Value	Internal Fast-PCI Clock Freq. (MHz)	
	33.33	**66.67**
4	---	266.7
7	233.3	–
8	266.7	–
9	300	–

Table 3-7. Strapped Core Clock Frequency

CLKSEL[3:0] Straps	Internal Fast-PCI Clock Freq. (MHz) (GCB+I/O Offset 1Eh[9:8])	Default ADL Multiplier		Maximum Core Clock Freq. (MHz)
		Multiply By	Multiplier Value (GCB+I/O Offset 1Eh[3:0])	
0000	33.33	7	0111	233
0100		8	1000	266
1000		9	1001	300
0110	66.67	4	0100	266

General Configuration Block (Continued)

3.5.4 SuperI/O Clocks

The SuperI/O module requires a 48 MHz input to the UART and other functions. This clock is supplied by PLL4 using a multiplier value of 576/(108x3) to generate 48 MHz.

3.5.5 Core Logic Module Clocks

The Core Logic module requires the following clock sources:

Real Time (RTC)

RTC requires a 32.768 KHz clock which is supplied directly from an internal low-power crystal oscillator. This oscillator uses battery power and has very low current consumption.

USB

The USB requires a 48 MHz input which is supplied by PLL4. The required total frequency accuracy and slow jitter for USB is 500 PPM; edge to edge jitter is ±1.2%.

ACPI

The ACPI logic block uses a 14.32 MHz clock supplied by FMUL3. FMUL3 creates this clock from the 32.768 KHz clock, with a multiplier value of 6992/4 to output a 57.278 MHz clock that is divided by 4.

External PCI

The PCI Interface uses a 33.3 MHz clock that is created by PLL5 and divided by 2. PLL5 uses the 27 MHz clock, to output a 66.67 MHz clock. PLL5 has a frequency accuracy of ± 0.1%.

AC97

The SC1100 generates the 24.576 MHz clock required by the audio codec. Therefore, no crystal need be included for the audio codec on the system board.

PLL3 uses the crystal oscillator clock to generate a 24.576 MHz clock. This clock is driven on the AC97_CLK signal (ball AC21). The accuracy of the clock supplied by the SC1100 is 50 PPM.

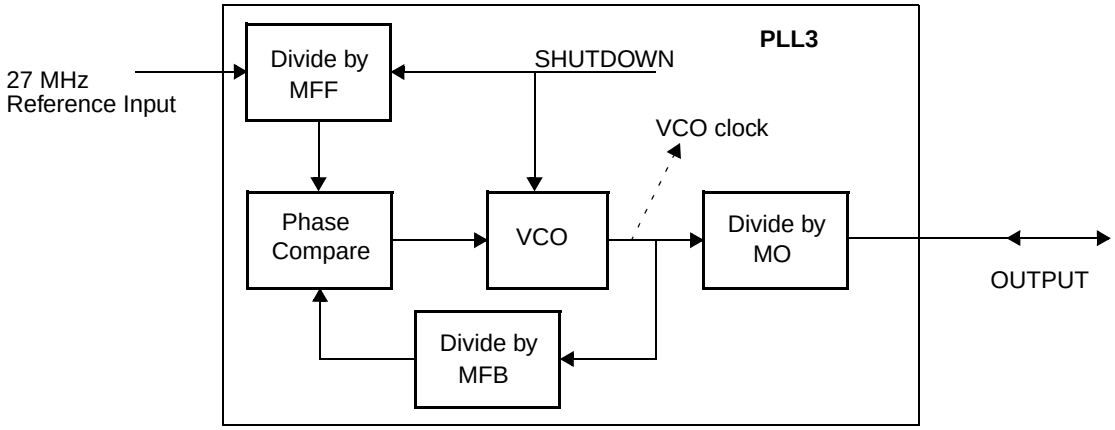


Figure 3-4. PLL3 and Dividers Block Diagram

General Configuration Block (Continued)

3.5.6 Clock Registers

Table 3-8. Clock Generator Configuration

Bit	Description
Offset: 10h **Maximum Core Clock Multiplier Register - MCCM (RO)** **Reset Value: Strapped Value** Width: Byte This register holds the maximum core clock multiplier value. The maximum clock frequency allowed by the core, is the Fast-PCI clock multiplied by this value.	
7:4	**Reserved.**
3:0	**MCM (Maximum Clock Multiplier).** This 4-bit value is the maximum multiplier value allowed for the core clock generator. It is derived from strap options CLKSEL[3:0] (balls AE22, AD22, AD25, D23).
Offset: 12h **PLL Power Control Register - PPCR (R/W)** **Reset Value: 2Fh** Width: Byte This register controls operation of the PLLs.	
7	**Reserved.** Write as read.
6	**EXPCID (Disable External PCI Clock).** 0: External PCI clock is enabled. 1: External PCI clock is disabled.
5	**Reserved.** Must be written as 1.
4	**Reserved.** Write as read.
3	**PLL3SD (Shut Down PLL3).** AC97 codec clock. 0: PLL3 is enabled. 1: PLL3 is shutdown.
2	**FM1SD (Shut Down PLL4).** 0: PLL4 is enabled. 1: PLL4 is shutdown.
1	**C48MD (Disable SuperI/O and USB Clock).** 0: USB and SuperI/O clock is enabled. 1: USB and SuperI/O clock is disabled.
0	**Reserved.** Write as read.
Offset: 18h **PLL3 Configuration Register - PLL3C (R/W)** **Reset Value: E1040005h** Width: DWORD	
31:24	**MFFC (MFF Counter Value).** F_{VCO} = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz
23:19	**Reserved.** Write as read.
18:8	**MFBC (MFB Counter Value).** F_{VCO} = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz
7	**Reserved.** Write as read.
6	**Reserved.** Must be set to 0.
5:0	**MOC (MO Counter Value).** F_{VCO} = OSCCLK * MFBC / (MFFC * MOC) OSCCLK = 27 MHz

General Configuration Block (Continued)

Table 3-8. Clock Generator Configuration (Continued)

Bit	Description
Offset: 1Eh **Core Clock Frequency Control Register - CCFC (R/W)** **Reset Value: Strapped Value** Width: WORD This register controls the configuration of the core clock multiplier and the reference clocks.	
15:14	**Reserved.** Write as read.
13	**Reserved.** Must be cleared to 0.
12	**Reserved.** Must be cleared to 0.
11:10	**Reserved.** Write as read.
9:8	**FPCICK (Internal Fast-PCI Clock). (Read Only)** Reflects the internal Fast-PCI clock and is the input to the GX1 module that is used to generate the core clock. These bits reflect the value of strap pins CLKSEL[1:0] (balls D23, AD25). 00: 33.3 MHz 01: Reserved 10: 66.7 MHz 11: 33.3 MHz
7:4	**Reserved.** Write as read.
3:0	**MVAL (Multiplier Value).** This 4-bit value controls the multiplier in ADL. The value is set according to the Maximum Clock Multiplier bits of the MCCM register (Offset 10h). The multiplier value should never be written with a multiplier which is different from the multiplier indicated in the MCCM register. 0100: Multiply by 4 0111: Multiply by 7 1000: Multiply by 8 Other: Reserved

4.0 SuperI/O Module

The SuperI/O (SIO) module is a member of National Semiconductor's SuperI/O family of integrated PC peripherals. It is a PC98 and ACPI compliant SIO that offers a single-cell solution to the most commonly used ISA peripherals.

The SIO module incorporates: one Serial Port, an Infrared Communication Port that supports FIR, MIR, HP-SIR, Sharp-IR, and Consumer Electronics-IR, two ACCESS.bus Interface (ACB) ports, System Wakeup Control (SWC), and a Real-Time Clock (RTC) that provides RTC timekeeping.

Outstanding Features

- Full compatibility with ACPI Revision 1.0 requirements.
- System Wakeup Control powered by V_{SB}, generates power-up request and a PME (power management event) in response to a pre-programmed CEIR, or a RI# (serial port ring indicate) event.
- Advanced RTC, Y2K compliant.

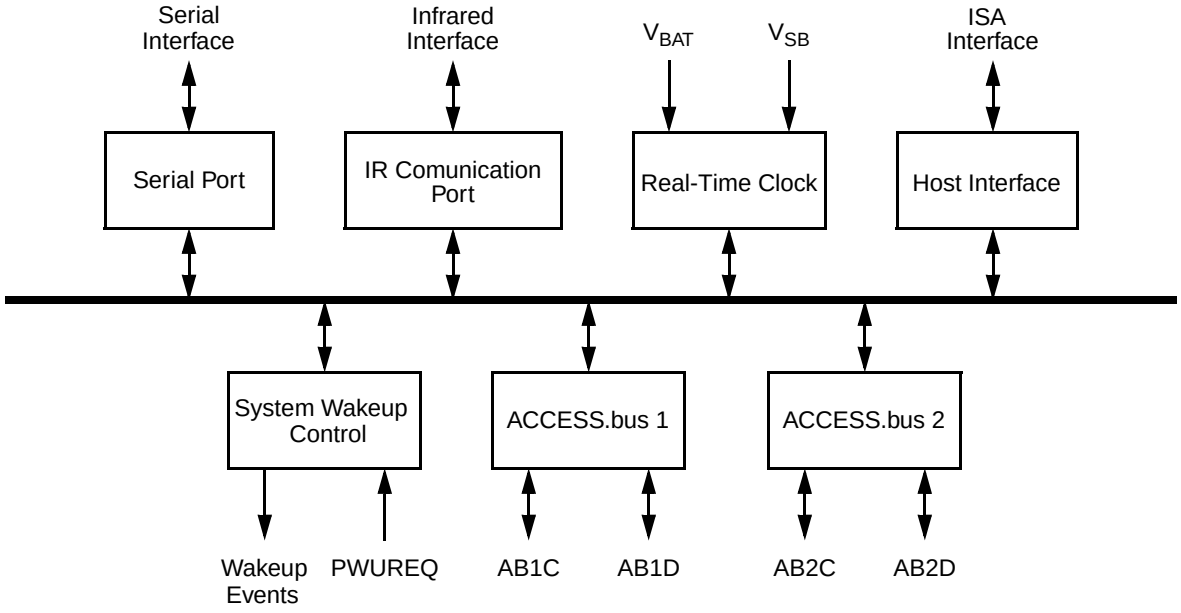


Figure 4-1. SIO Block Diagram

SuperI/O Module (Continued)

4.1 FEATURES

PC98 and ACPI Compliant

- PnP Configuration Register structure
- Flexible resource allocation for all logical devices:
  - — Relocatable base address
  - — 9 parallel IRQ routing options
  - — 3 optional 8-bit DMA channels (where applicable)

Serial Port

- 16550A compatible

Infrared Communication Port

- IrDA 1.1 and 1.0 compatible
- Data rate of 1.152 Mbps (MIR)
- Data rate of 4.0 Mbps (FIR)
- Data rate of up to 115.2 Kbps (HP-SIR)
- Selectable internal or external modulation/demodulation (ASK-IR and DASK-IR options of SHARP-IR)
- Consumer-IR (TV-Remote) mode
- Consumer Remote Control supports RC-5, RC-6, NEC, RCA and RECS 80
- DMA support

System WakeUp Control (SWC)

- Power-up request upon detection of RI# or CEIR activity:
  - — Optional routing of power-up request on IRQ line
- Pre-programmed CEIR address in a pre-selected standard (any NEC, RCA or RC-5)
- Powered by V_{SB}
- Battery-backed wakeup setup
- Power-fail recovery support

Real-Time Clock

- A modifiable address that is referenced by a 16-bit programmable register
- DS1287, MC146818 and PC87911 compatibility
- 242 bytes of battery backed up CMOS RAM in two banks
- Selective lock mechanisms for the 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
- Three different maskable interrupt flags:
  - — Periodic interrupts - At intervals from 122 msec to 500 msec
  - — Time-of-Month alarm - At intervals from once per second to once per month
  - — Update Ended Interrupt - Once per second upon completion of update
- Separate battery pin, 3.0V operation that includes an internal UL protection resistor
- 7 µA maximum power consumption during power down
- Double-buffer time registers
- Y2K Compliant

Clock Sources

- 48 MHz clock input
- On-chip low frequency clock generator for wakeup
- 32.768 KHz crystal with an internal frequency multiplier to generate all required internal frequencies

SuperI/O Module (Continued)

4.2 MODULE ARCHITECTURE

The SIO module comprises a collection of generic functional blocks. Each functional block is described in detail later in this chapter. The beginning of this chapter describes the SIO structure and provides all device specific information, including special implementation of generic blocks, system interface, and device configuration.

The SIO module is based on six logical devices, the host interface, and a central configuration register set, all built around a central, internal 8-bit bus.

The host interface serves as a bridge between the external ISA interface and the internal bus. It supports 8-bit I/O read, 8-bit I/O write and 8-bit DMA transactions, as defined in *Personal Computer Bus Standard P996*.

The central configuration register set supports ACPI compliant PnP configuration. The configuration registers are structured as a subset of the Plug and Play Standard Registers, defined in Appendix A of the *Plug and Play ISA Specification* Version 1.0a by Intel and Microsoft. All system resources assigned to the functional blocks (I/O address space, DMA channels and IRQ lines) are configured in, and managed by, the central configuration register set. In addition, some function-specific parameters are configurable through this unit and distributed to the functional blocks through special control signals.

The source of the device internal clocks is the 48 MHz clock signal or through the 32.768 KHz crystal with an internal frequency multiplier. The RTC operates on a 32.768 KHz clock.

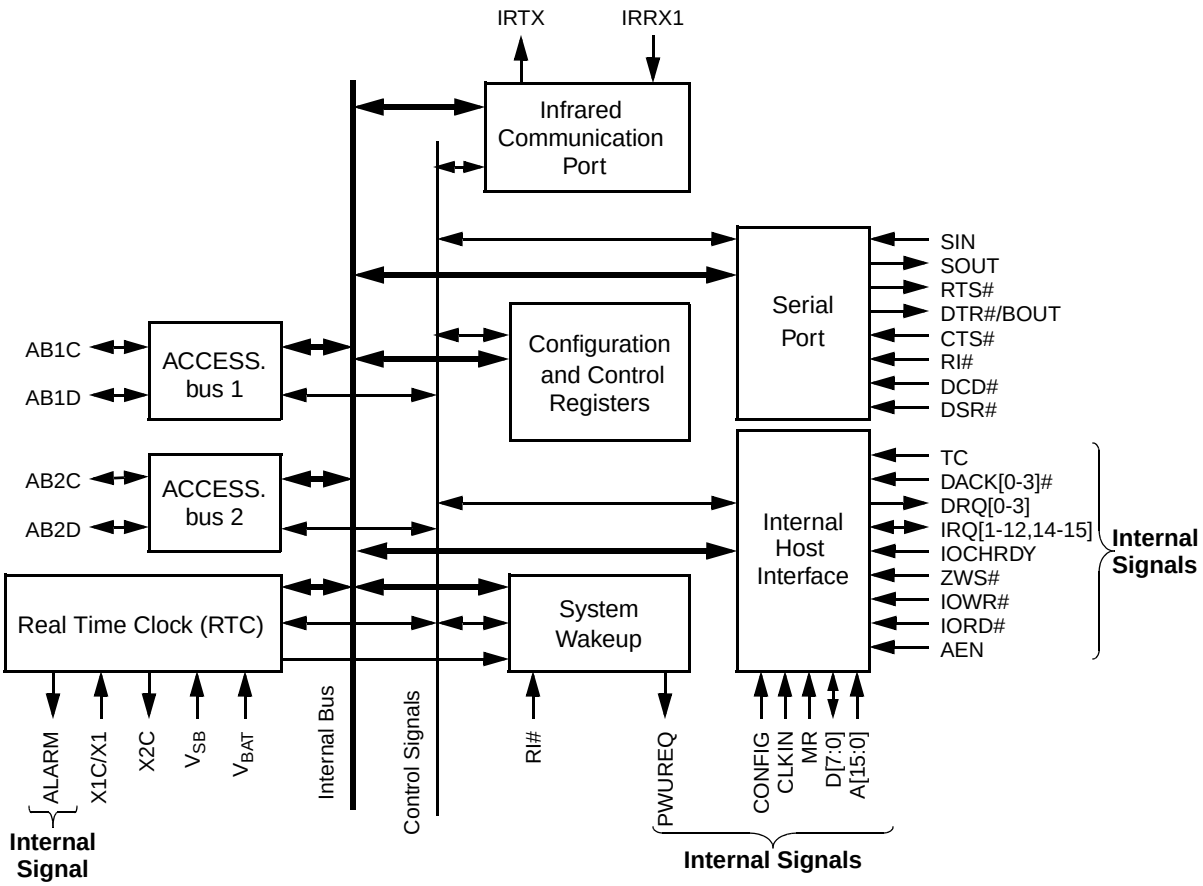


Figure 4-2. Detailed SIO Block Diagram

SuperI/O Module (Continued)

4.3 CONFIGURATION STRUCTURE / ACCESS

This section describes the structure of the configuration register file, and the method of accessing the configuration registers.

4.3.1 Index-Data Register Pair

The SIO configuration access is performed via an Index-Data register pair, using only two system I/O byte locations. The base address of this register pair is determined according to the state of the IO_SIOCFG_IN bit field of the Core Logic module (F5BAR0+I/O Offset 00h[26:25]). Table 4-1 shows the selected base addresses as a function of the IO_SIOCFG_IN bit field.

Table 4-1. SIO Configuration Options

IO_SIOCFG_IN Settings	I/O Address		Description
	Index Register	Data Register	
00	-	-	SIO disabled
01	-	-	Configuration access disabled
10	002Eh	002Fh	Base address 1 selected
11	015Ch	015Dh	Base address 2 selected

The Index register is an 8-bit R/W register located at the selected base address (Base+0). It is used as a pointer to the configuration register file, and holds the index of the configuration register that is currently accessible via the Data register. Reading the Index register returns the last value written to it (or the default of 00h after reset).

The Data register is an 8-bit virtual register, used as a data path to any configuration register. Accessing the data register results with physically accessing the configuration register that is currently pointed by the Index register.

4.3.2 Banked Logical Device Registers

Each functional block is associated with a Logical Device Number (LDN). The configuration registers are grouped into banks, where each bank holds the standard configuration registers of the corresponding logical device. Table 4-2 shows the LDNs of the device functional blocks.

Table 4-2. LDN Assignments

LDN	Functional Block
00h	Real Time Clock (RTC)
01h	System Wakeup Control (SWC)
02h	Infrared Communication Port (IRCP)
05h	ACCESS.bus 1 (ACB1)
06h	ACCESS.bus 2 (ACB2)
08h	Serial Port

Figure 4-3 shows the structure of the standard PnP configuration register file. The SIO Control and Configuration registers are not banked and are accessed by the Index-Data register pair only (as described above). However, the Logical Device Control and Configuration registers are duplicated over four banks for four logical devices. Therefore, accessing a specific register in a specific bank is performed by two-dimensional indexing, where the LDN register selects the bank (or logical device), and the Index register selects the register within the bank. Accessing the Data register while the Index register holds a value of 30h or higher results in a physical access to the Logical Device Configuration registers currently pointed to by the Index register, within the logical device bank currently selected by the LDN register.

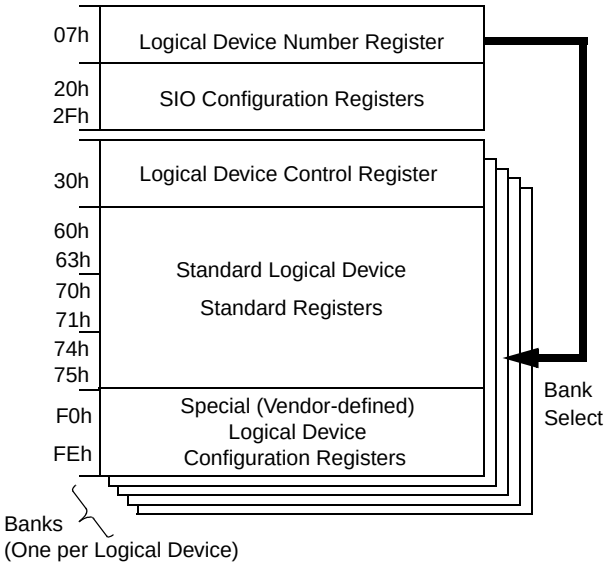


Figure 4-3. Structure of the Standard Configuration Register File

SuperI/O Module (Continued)

Write accesses to unimplemented registers (i.e., accessing the Data register while the Index register points to a non-existing register or the LDN is higher than 08h), are ignored and a read returns 00h on all addresses except for 74h and 75h (DMA configuration registers) which returns 04h (indicating no DMA channel is active). The configuration registers are accessible immediately after reset.

4.3.3 Default Configuration Setup

The device has four reset types:

Software Reset

This reset is generated by bit 1 of the SIOCF1 register, which resets all logical devices. A software reset also resets most bits in the SIO Configuration and Control registers (see Section 4.4.1 on page 73 for the bits not affected). This reset does not affect register bits that are locked for write access.

Hardware Reset

This reset is activated by the system reset signal. This resets all logical devices, with the exception of the RTC and the SWC, and all SIO Configuration and Control registers, with the exception of the SIOCF2 register. It also resets all SIO Control and Configuration registers, except for those that are battery-backed.

V_{PP} Power-Up Reset

This reset is activated when either V_{SB} or V_{BAT} is powered on after both have been off. V_{PP} is an internal voltage which is a combination of V_{SB} and V_{BAT}. V_{PP} is taken from V_{SB} if V_{SB} is greater than the minimum (Min) value defined in Section 7.1.3 "Operating Conditions" on page 281; otherwise, V_{BAT} is used as the V_{PP} source. This reset resets all registers whose values are retained by V_{PP}.

V_{SB} Power-Up Reset

This is an internally generated reset that resets the SWC, excluding those SWC registers whose values are retained by V_{PP}. This reset is activated after V_{SB} is powered up.

The SIO module wakes up with the default setup, as follows:

- When a hardware reset occurs:
  - — The configuration base address is 2Eh, 15Ch or None, according to the IO_SIOCFG_IN bit values, as shown in Table 4-1 on page 68.
  - — All Logical devices are disabled, with the exception of the RTC and the SWC, which remains functional but whose registers cannot be accessed.
- When either a hardware or a software reset occurs:
  - — The legacy devices are assigned with their legacy system resource allocation.
  - — The National proprietary functions are not assigned with any default resources and the default values of their base addresses are all 00h.

4.3.4 Address Decoding

A full 16-bit address decoding is applied when accessing the configuration I/O space, as well as the registers of the functional blocks. However, the number of configurable bits in the base address registers vary for each device.

The lower 1, 2, 3 or 4 address bits are decoded within the functional block to determine the offset of the accessed register, within the device's I/O range of 2, 4, 8 or 16 bytes, respectively. The rest of the bits are matched with the base address register to decode the entire I/O range allocated to the device. Therefore the lower bits of the base address register are forced to 0 (RO), and the base address is forced to be 2, 4, 8 or 16 byte aligned, according to the size of the I/O range.

The base address of the RTC, Serial Port, and the Infrared Communication Port are limited to the I/O address range of 00h to 7Fxh only (bits [15:11] are forced to 0).The addresses of the non-legacy devices are configurable within the full 16-bit address range (up to FFFxh).

In some special cases, other address bits are used for internal decoding. For more details, please see the detailed description of the base address register for each specific logical device.

SuperI/O Module (Continued)

4.4 STANDARD CONFIGURATION REGISTERS

As illustrated in Figure 4-4, the Standard Configuration registers are broadly divided into two categories: SIO Control and Configuration registers and Logical Device Control and Configuration registers (one per logical device, some are optional).

SIO Control and Configuration Registers

The only PnP control register in the SIO module is the Logical Device Number register at Index 07h. All other standard PnP control registers are associated with PnP protocol for ISA add-in cards, and are not supported by the SIO module.

The SIO Configuration registers at Index 20h-27h are mainly used for part identification. (See Section 4.4.1 "SIO Control and Configuration Registers" on page 73 for further details.)

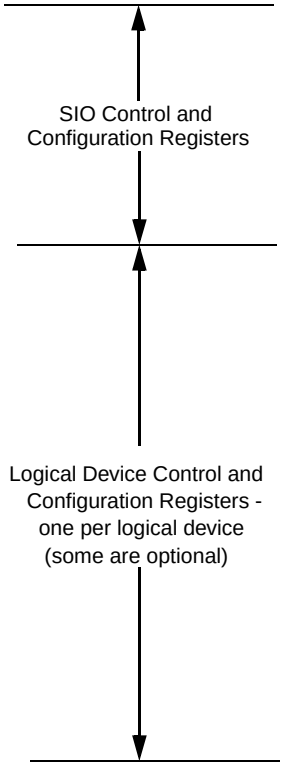
Logical Device Control and Configuration Registers

A subset of these registers is implemented for each logical device. (See Table 4-2 on page 68 for LDN assignment and Section 4.4.2 "Logical Device Control and Configuration" on page 74 for register details.)

Logical Device Control Register (Index 30h): The only implemented Logical Device Control register is the Activate register at Index 30. Bit 0 of the Activate register and bit 0 of the SIO Configuration 1 register (Global Device Enable bit) control the activation of the associated function block (except for the RTC and the SWC). Activation of the block enables access to the block's registers, and attaches its system resources, which are unused as long as the block is not activated. Activation of the block may also result in other effects (e.g., clock enable and active signaling), for certain functions.

Standard Logical Device Configuration Registers (Index 60h-75h): These registers are used to manage the resource allocation to the functional blocks. The I/O port base address descriptor 0 is a pair of registers at Index 60h-61h, holding the (first or only) 16-bit base address for the register set of the functional block. An optional second base-address (descriptor 1) at Index 62h-63h is used for devices with more than one continuous register set. Interrupt Number Select (Index 70h) and Interrupt Type Select (Index 71h) allocate an IRQ line to the block and control its type. DMA Channel Select 0 (Index 74h) allocates a DMA channel to the block, where applicable. DMA Channel Select 1 (Index 75h) allocates a second DMA channel, where applicable.

Special Logical Device Configuration Registers (F0h-F3h): The vendor-defined registers, starting at Index F0h are used to control function-specific parameters such as operation modes, power saving modes, pin TRI-STATE, clock rate selection, and non-standard extensions to generic functions.



Index	Register Name
07h	Logical Device Number
20h	SIO ID
21h	SIO Configuration 1
22h	SIO Configuration 2
27h	SIO Revision ID
2Eh	Reserved exclusively for National use
30h	Logical Device Control (Activate)
60h	I/O Port Base Address Descriptor 0 Bits [15:8]
61h	I/O Port Base Address Descriptor 0 Bits [7:0]
62h	I/O Port Base Address Descriptor 1 Bits [15:8]
63h	I/O Port Base Address Descriptor 1 Bits [7:0]
70h	Interrupt Number Select
71h	Interrupt Type Select
74h	DMA Channel Select 0
75h	DMA Channel Select 1
F0h	Device Specific Logical Device Configuration 1
F1h	Device Specific Logical Device Configuration 2
F2h	Device Specific Logical Device Configuration 3
F3h	Device Specific Logical Device Configuration 4

Figure 4-4. Standard Configuration Registers Map

SuperI/O Module (Continued)

Table 4-3 provides the bit definitions for the Standard Configuration registers.

- All reserved bits return 0 on reads, except where noted otherwise. They must not be modified as such modification may cause unpredictable results. Use read-modify-write to prevent the values of reserved bits from being changed during write.
- Write only registers should not use read-modify-write during updates.

Table 4-3. Standard Configuration Registers

Bit	Description
Index 07h	**Logical Device Number (R/W)**
Width: Byte	
This register selects the current logical device. See Table 4-2 for valid numbers. All other values are reserved.	
7:0	**Logical Device Number.**
Index 20h-2Fh	**SIO Configuration (R/W)**
Width: Byte	
SIO configuration and ID registers. See Section 4.4.1 "SIO Control and Configuration Registers" on page 73 for register/bit details.	
Index 30h	**Activate (R/W)**
Width: Byte	
7:1	**Reserved.**
0	**Logical Device Activation Control.** 0: Disable 1: Enable
Index 60h	**I/O Port Base Address Bits [15:8] Descriptor 0 (R/W)**
Width: Byte	
Indicates selected I/O lower limit address bits [15:8] for I/O Descriptor 0.	
Index 61h	**I/O Port Base Address Bits [7:0] Descriptor 0 (R/W)**
Width: Byte	
Indicates selected I/O lower limit address bits [7:0] for I/O Descriptor 0.	
Index 62h	**I/O Port Base Address Bits [15:8] Descriptor 1 (R/W)**
Width: Byte	
Indicates selected I/O lower limit address bits [15:8] for I/O Descriptor 1.	
Index 63h	**I/O Port Base Address Bits [7:0] Descriptor 1 (R/W)**
Width: Byte	
Indicates selected I/O lower limit address bits [7:0] for I/O Descriptor 1.	
Index 70h	**Interrupt Number (R/W)**
Width: Byte	
7:4	**Reserved.**
3:0	**Interrupt Number.** These bits select the interrupt number. A value of 1 selects IRQ1, a value of 2 selects IRQ2, etc. (up to IRQ12). **Note:** IRQ0 is not a valid interrupt selection.
Index 71h	**Interrupt Request Type Select (R/W)**
Width: Byte	
Selects the type and level of the interrupt request number selected in the previous register.	
7:2	**Reserved.**
1	**Interrupt Level Requested.** Level of interrupt request selected in previous register. 0: Low polarity. 1: High polarity. This bit must be set to 1 (high polarity), except for IRQ8#, that must be low polarity.
0	**Interrupt Type Requested.** Type of interrupt request selected in previous register. 0: Edge. 1: Level.

SuperI/O Module (Continued)

Table 4-3. Standard Configuration Registers (Continued)

Bit	Description
Index 74h	**DMA Channel Select 0 (R/W)**
Width: Byte	
Selected DMA channel for DMA 0 of the logical device (0 - the first DMA channel when using more than one DMA channel).	
7:3	**Reserved.**
2:0	**DMA 0 Channel Select.** This bit field selects the DMA channel for DMA 0. The valid choices are 0-3, where a value of 0 selects DMA channel 0, 1 selects channel 1, etc. A value of 4 indicates that no DMA channel is active. Values 5-7 are reserved.
Index 75h	**DMA Channel Select 1 (R/W)**
Width: Byte	
Indicates selected DMA channel for DMA 1 of the logical device (1 - the second DMA channel when using more than one DMA channel).	
7:3	**Reserved.**
2:0	**DMA 1 Channel Select:** This bit field selects the DMA channel for DMA 1. The valid choices are 0-3, where a value of 0 selects DMA channel 0, 1 selects channel 1, etc. A value of 4 indicates that no DMA channel is active. Values 5-7 are reserved.
Index F0h-FEh	**Logical Device Configuration (R/W)**
Width: Byte	
Special (vendor-defined) configuration options.	

SuperI/O Module (Continued)

4.4.1 SIO Control and Configuration Registers

Table 4-4 lists the SIO Control and Configuration registers and Table 4-5 provides their bit formats.

Table 4-4. SIO Control and Configuration Register Map

Index	Type	Name	Power Rail	Reset Value
20h	RO	**SID.** SIO ID	V_{CORE}	F5h
21h	R/W	**SIOCF1.** SIO Configuration 1	V_{CORE}	01h
22h	R/W	**SIOCF2.** SIO Configuration 2	V_{PP}	02h
27h	RO	**SRID.** SIO Revision ID	V_{CORE}	01h
2Eh	---	**RSVD.** Reserved exclusively for National use.	---	---

Table 4-5. SIO Control and Configuration Registers

Bit	Description
Index 20h	**SIO ID Register - SID (RO)** **Reset Value: F5h**
7:0	**Chip ID.** Contains the identity number of the module. The SIO module is identified by the value F5h.
Index 21h	**SIO Configuration 1 Register - SIOCF1 (RW)** **Reset Value: 01h**
7:6	**General Purpose Scratch.** When bit 5 is set to 1, these bits are RO. After reset, these bits can be R/W. Once changed to RO, the bits can be changed back to R/W only by a hardware reset.
5	**Lock Scratch.** This bit controls bits 7 and 6 of this register. Once this bit is set to 1 by software, it can be cleared to 0 only by a hardware reset. 0: Bits 7 and 6 of this register are R/W bits. (Default) 1: Bits 7 and 6 of this register are RO bits.
4:2	**Reserved.**
1	**SW Reset**. Read always returns 0. 0: Ignored. (Default) 1: Resets all devices that are reset by MR (with the exception of the lock bits) and the registers of the SWC.
0	**Global Device Enable.** This bit controls the function enable of all the logical devices in the SIO module, except the SWC and the RTC. It allows them to be disabled simultaneously by writing to a single bit. 0: All logical devices in the SIO module are disabled, except the SWC and the RTC. 1: Each logical device is enabled according to its Activate register at Index 30h. (Default)
Index 22h	**SIO Configuration 2 Register - SIOCF2 (R/W)** **Reset Value: 02h** **Note:** This register is reset only when V_{PP} is first applied.
7	**Reserved.**
6:4	**General Purpose Scratch.** Battery-backed.
3:2	**Reserved.**
1	**Reserved.**
0	**Reserved. (RO)**
Index 27h	**SIO Revision ID Register - SRID (RO)** **Reset Value: 01h**
7:0	**SIO Revision ID. (RO)** This RO register contains the identity number of the chip revision. SRID is incremented on each revision.

SuperI/O Module (Continued)

4.4.2 Logical Device Control and Configuration

As described in Section 4.3.2 "Banked Logical Device Registers" on page 68, each functional block is associated with a Logical Device Number (LDN). This section provides the register descriptions for each LDN.

The register descriptions in this subsection use the following abbreviations for Type:

- R/W = Read/Write
- R = Read from a specific address returns the value of a specific register. Write to the same address is to a different register.
- W = Write
- RO = Read Only
- R/W1C = Read/Write 1 to Clear. Writing 1 to a bit clears it to 0. Writing 0 has no effect.

4.4.2.1 LDN 00h - Real-Time Clock

Table 4-6 lists the registers which are relevant to configuration of the Real-Time Clock (RTC). Only the last registers (F0h-F3h) are described here (Table 4-7). See Table 4-3 "Standard Configuration Registers" on page 71 for descriptions of the other registers.

Table 4-6. Relevant RTC Configuration Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** When bit 0 is cleared, the registers of this logical device are not accessible.[1]	00h
60h	R/W	**Standard Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	00h
61h	R/W	**Standard Base Address LSB register.** Bit 0 (for A0) is RO, 0b.	70h
62h	R/W	**Extended Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	00h
63h	R/W	**Extended Base Address LSB register.** Bit 0 (for A0) is RO, 0b.	72h
70h	R/W	**Interrupt Number.**	08h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	00h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h
F0h	R/W	**RAM Lock register (RLR).**	00h
F1h	R/W	**Date of Month Alarm Offset register (DOMAO).** Sets index of Date of Month Alarm register in the standard base address.	00h
F2h	R/W	**Month Alarm Offset register (MONAO).** Sets index of Month Alarm register in the standard base address.	00h
F3h	R/W	**Century Offset register (CENO).** Sets index of Century register in the standard base address.	00h

1. The logical device registers are maintained, and all RTC mechanisms are functional.

SuperI/O Module (Continued)

Table 4-7. RTC Configuration Registers

Bit	Description
Index F0h	**RAM Lock Register - RLR (R/W)** **Reset Value: 00h** Width: Byte When any non-reserved bit in this register is set to 1, it can be cleared only by hardware reset.
7	**Block Standard RAM.** 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	**Block RAM Write.** 0: No effect on RAM access. (Default) 1: Writes to RAM (Standard and Extended) are ignored.
5	**Block Extended RAM Write.** This bit controls writes to bytes 00h-1Fh of the Extended RAM. 0: No effect on the Extended RAM access. (Default) 1: Writes to bytes 00h-1Fh of the Extended RAM are ignored.
4	**Block Extended RAM Read.** This bit controls read from bytes 00h-1Fh of the Extended RAM. 0: No effect on Extended RAM access. (Default) 1: Reads to bytes 00h-1Fh of the Extended RAM are ignored.
3	**Block Extended RAM.** 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	**Reserved.**
Index F1h	**Date Of Month Alarm Offset Register - DOMAO (R/W)** **Reset Value: 00h**
7	**Reserved.**
6:0	**Date of Month Alarm Register Offset Value.**
Index F2h	**Month Alarm Offset Register - MANAO (R/W)** **Reset Value: 00h**
7	**Reserved.**
6:0	**Month Alarm Register Offset Value.**
Index F3h	**Century Offset Register - CENO (R/W)** **Reset Value: 00h**
7	**Reserved.**
6:0	**Century Register Offset Value.**

SuperI/O Module (Continued)

4.4.2.2 LDN 01h - System Wakeup Control

Table 4-8 lists registers that are relevant to the configuration of System Wakeup Control (SWC). These registers are described earlier in Table 4-3 "Standard Configuration Registers" on page 71.

Table 4-8. Relevant SWC Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** When bit 0 is cleared, the registers of this logical device are not accessible.[1]	00h
60h	R/W	**Base Address MSB register.**	00h
61h	R/W	**Base Address LSB register.** Bits [3:0] (for A[3:0]) are RO, 0000b.	00h
70h	R/W	**Interrupt Number.** (For routing the internal PWUREQ signal.)	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W. Other bits are RO.	03h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h

1. The logical device registers are maintained, and all wakeup detection mechanisms are functional.

SuperI/O Module (Continued)

4.4.2.3 LDN 02h - Infrared Communication Port

Table 4-9 lists the configuration registers which affect the Infrared Communication Port (IRCP). Only the last register (F0h) is described here (Table 4-10). See Table 4-3 "Standard Configuration Registers" on page 71 for descriptions of the other registers listed.

Table 4-9. Relevant IRCP Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register.	00h
60h	R/W	**Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	03h
61h	R/W	**Base Address LSB register.** Bit [2:0] (for A[2:0]) are RO, 000b.	E8h
70h	R/W	**Interrupt Number.**	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	03h
74h	R/W	**DMA Channel Select 0 (RX_DMA)**.	04h
75h	R/W	**DMA Channel Select 1 (TX_DMA).**	04h
F0h	R/W	**Infrared Communication Port Configuration register.**	02h

Table 4-10. IRCP Configuration Register

Bit	Description
Index F0h	**Infrared Communication Port/Serial Port 3 Configuration Register (R/W) — Reset Value: 02h**
7	**Bank Select Enable.** Enables bank switching. 0: All attempts to access the extended registers are ignored. (Default) 1: Enables bank switching.
6:3	**Reserved.**
2	**Busy Indicator.** (RO) This bit can be used by power management software to decide when to power-down the device. 0: No transfer in progress. (Default) 1: Transfer in progress.
1	**Power Mode Control.** When the logical device is active in: 0: Low power mode - Clock disabled. The output signals are set to their default states. Registers are maintained. (Unlike Active bit in Index 30h that also prevents access to device registers.) 1: Normal power mode - Clock enabled. The device is functional when the logical device is active. (Default)
0	**TRI-STATE Control**. When enabled and the device is inactive, the logical device output pins are in TRI-STATE. One exception is the IRTX signal, which is driven to 0 when the Infrared Communication Port is inactive and is not affected by this bit. 0: Disabled. (Default) 1: Enabled when the device is inactive.

SuperI/O Module (Continued)

4.4.2.4 LDN 05h and 06h - ACCESS.bus Ports 1 and 2

ACCESS.bus ports 1 and 2 (ACB1 and ACB2) are identical. Each ACB is a two-wire synchronous serial interface compatible with the ACCESS.bus physical layer. ACB1 and ACB2 use a 24 MHz internal clock. Six runtime registers for each ACCESS.bus are described in Section 4.7 "ACCESS.bus Interface" on page 97.

ACB1 is designated as LDN 05h and ACB2 as LDN 06h. Table 4-11 lists the configuration registers which affect the ACCESS.bus ports. Only the last register (F0h) is described here (Table 4-12). See Table 4-3 "Standard Configuration Registers" on page 71 for descriptions of the others.

Table 4-11. Relevant ACB1 and ACB2 Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register	00h
60h	R/W	**Base Address MSB register.**	00h
61h	R/W	**Base Address LSB register.** Bits [2:0] (for A[2:0]) are RO, 000b.	00h
70h	R/W	**Interrupt Number.**	00h
71h	R/W	**Interrupt Type.** Bit 1 is R/W. Other bits are RO.	03h
74h	RO	**Report no DMA assignment.**	04h
75h	RO	**Report no DMA assignment.**	04h
F0h	R/W	**ACB1 and ACB2 Configuration register.**	00h

Table 4-12. ACB1 and ACB2 Configuration Register

Bit	Description
Index F0h **ACB1 and ACB2 Configuration Register (R/W)** **Reset Value: 00h** Width: Byte This register is reset by hardware to 00h.	
7:3	**Reserved.**
2	**Internal Pull-Up Enable.** 0: No internal pull-up resistors on AB1C/AB2C and AB1D/AB2D. (Default) 1: Internal pull-up resistors on AB1C/AB2C and AB1D/AB2D.
1:0	**Reserved.**

SuperI/O Module (Continued)

4.4.2.5 LDN 08h - SP Configuration

Table 4-13 lists the configuration registers which affect the Serial Port (SP). Only the last register (F0h) is described here (Table 4-14). See Table 4-3 "Standard Configuration Registers" on page 71 for descriptions of the others.

Table 4-13. SP Relevant Registers

Index	Type	Configuration Register or Action	Reset Value
30h	R/W	**Activate.** See also bit 0 of the SIOCF1 register.	00h
60h	R/W	**Base Address MSB register.** Bits [7:3] (for A[15:11]) are RO, 00000b.	02h
61h	R/W	**Base Address LSB register**. Bits [2:0] (for A[2:0]) are RO, 000b.	F8h
70h	R/W	**Interrupt Number.**	03h
71h	R/W	**Interrupt Type.** Bit 1 is R/W; other bits are RO.	03h
74h	RO	**Report no DMA assignment**.	04h
75h	RO	**Report no DMA assignment.**	04h
F0h	R/W	**Serial Port Configuration register.** (See Table 4-14.)	02h

Table 4-14. SP Configuration Register

Bit	Description
Index F0h	**Serial Port Configuration Register (R/W)** **Reset Value: 02h**
7	**Bank Select Enable**. Enables bank switching. 0: Disabled. (Default) 1: Enabled.
6:3	**Reserved.**
2	**Busy Indicator. (RO)** This bit can be used by power management software to decide when to power-down the logical device. 0: No transfer in progress. (Default) 1: Transfer in progress.
1	**Power Mode Control.** When the logical device is active in: 0: Low power mode - Serial Port clock disabled. The output signals are set to their default states. Registers are maintained. (Unlike Active bit in Index 30h that also prevents access to Serial Port registers.) 1: Normal power mode - Serial Port clock enabled. Serial Port is functional when the respective logical device is active. (Default)
0	**TRI-STATE Control**. This bit controls the TRI-STATE status of the device output pins when it is inactive (disabled). 0: Disabled. (Default) 1: Enabled when device inactive.

SuperI/O Module (Continued)

4.5 REAL-TIME CLOCK (RTC)

The RTC provides timekeeping and calendar management capabilities. The RTC uses a 32.768 KHz signal as the basic clock for timekeeping. It also includes 242 bytes of battery-backed RAM for general-purpose use.

The RTC provides the following functions:

- 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
- PnP support:
  — Relocatable Index and Data registers
  — Module access enable/disable option
  — Host interrupt enable/disable option
- 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

4.5.1 Bus Interface

The RTC function is initially mapped to the default SuperI/O locations at Indexes 70h to 73h (two Index/Data pairs). These locations may be reassigned, in compliance with Plug and Play requirements.

4.5.2 RTC Clock Generation

The RTC uses a 32.768 KHz clock signal as the basic clock for timekeeping. The 32.768 KHz clock can be supplied by the internal oscillator circuit, or by an external oscillator (see Section 4.5.2.2 "External Oscillator" on page 81).

4.5.2.1 Internal Oscillator

The internal oscillator employs an external crystal connected to the on-chip amplifier. The on-chip amplifier is accessible on the X32I input and X32O output. See Figure 4-5 for the recommended external circuit and Table 4-5 for a listing of the circuit components. The oscillator may be disabled in certain conditions. See Section 4.5.3.5 "Oscillator Activity" on page 84 for more details.

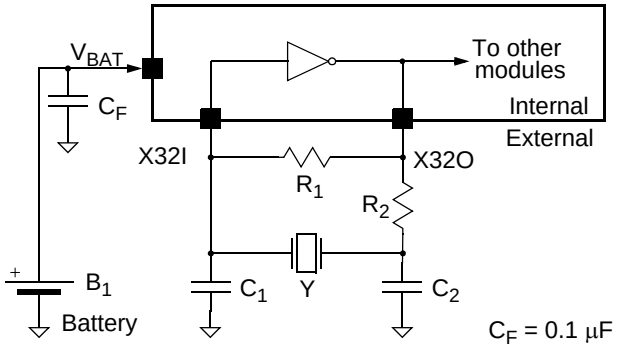


Figure 4-5. Recommended Crystal External Circuitry

Table 4-15. Crystal Oscillator Circuit Components

Component	Parameters	Values	Tolerance
Crystal	Resonance Frequency	32.768 KHz Parallel Mode	User-defined
	Type	N-Cut or XY-bar	
	Serial Resistance	40 KΩ	Max
	Quality Factor, Q	35000	Min
	Shunt Capacitance	2 pF	Max
	Load Capacitance, C_L	9-13 pF	
	Temperature Coefficient	User-defined	
Resistor R_1	Resistance	20 MΩ	5%
Resistor R_2	Resistance	180 KΩ	5%
Capacitor C_1	Capacitance	3 to 10 pF	5%
Capacitor C_2	Capacitance	3 to 10 pF	5%

SuperI/O Module (Continued)

External Elements

Choose C_1 and C_2 capacitors (see Figure 4-5 on page 80) to match the crystal's load capacitance. The load capacitance C_L "seen" by crystal Y is comprised of C_1 in series with C_2 and in parallel with the parasitic capacitance of the circuit. The parasitic capacitance is caused by the chip package, board layout and socket (if any), and can vary from 0 to 10 pF. The rule of thumb in choosing these capacitors is:

$$C_L = (C_1 * C_2) / (C_1 + C_2) + C_{PARASITIC}$$

Example:

Crystal C_L = 10 pF, $C_{PARASITIC}$ = 8.2 pF
C_1 = 3.6 pF, C_2 = 3.6 pF

Oscillator Tuning

The oscillator starts to generate 32.768 KHz pulses to the RTC after about 100 msec from when V_{BAT} is higher than V_{BATMIN} (2.4V) or V_{SB} is higher than V_{SBMIN} (3.0V). The oscillation amplitude on the X32O pin stabilizes to its final value (approximately 0.4V peak-to-peak around 0.7V DC) in about 1 s.

C_1 can be trimmed to achieve precisely 32.768 KHz. To achieve a high time accuracy, use crystal and capacitors with low tolerance and temperature coefficients.

4.5.2.2 External Oscillator

32.768 KHz can be applied from an external clock source, as shown in Figure 4-6.

Connections

Connect the clock to the X32I ball, leaving the oscillator output, X32O, unconnected.

Signal Parameters

The signal levels should conform to the voltage level requirements for X32I, of square or sine wave of 0.0V to V_{CORE} amplitude. The signal should have a duty cycle of approximately 50%. It should be sourced from a battery-backed source in order to oscillate during power-down. This assures that the RTC delivers updated time/calendar information.

4.5.3 Timing Generation

The timing generation function divides the 32.768 KHz clock by 2^{15} to derive a 1 Hz signal, which serves as the input for the seconds counter. This is performed by a divider chain composed of 15 divide-by-two latches, as shown in Figure 4-7.

Bits [6:4] (DV[2:0]) of the CRA register control the following functions:

- Normal operation of the divider chain (counting).
- Divider chain reset to 0.
- Oscillator activity when only V_{BAT} power is present (backup state).

The divider chain can be activated by setting normal operational mode (bits [6:4] of CRA = 01x or 100). The first update occurs 500 msec after divider chain activation.

Bits [3:0] of the CRA register select one of the fifteen taps from the divider chain to be used as a periodic interrupt. The periodic flag becomes active after half of the programmed period has elapsed, following divider chain activation.

See Table 4-18 on page 87 for more details.

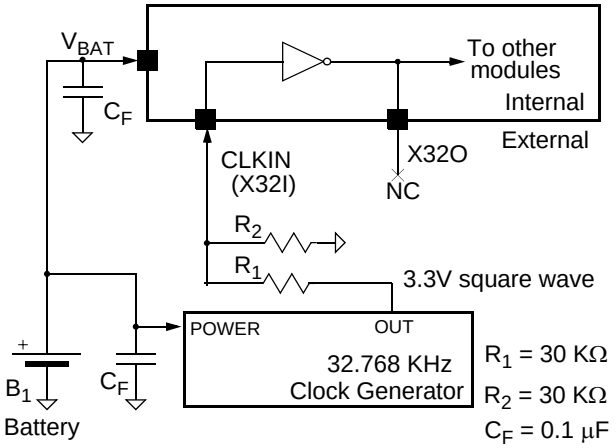


Figure 4-6. External Oscillator Connections

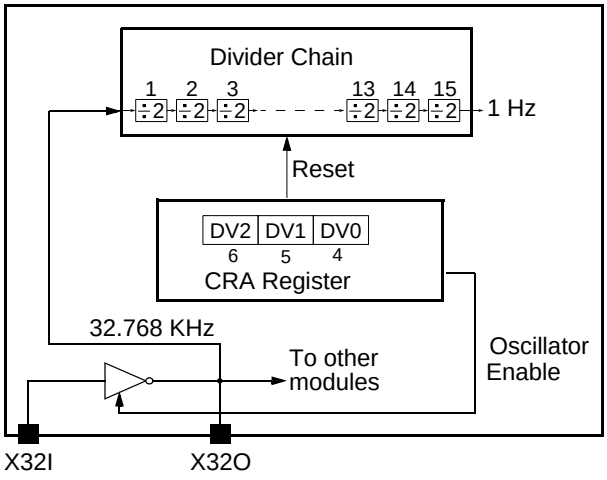


Figure 4-7. Divider Chain Control

SuperI/O Module (Continued)

4.5.3.1 Timekeeping

Data Format

Time is kept in BCD (Binary Coded Decimal) or binary format, as determined by bit 2 (DM) of Control Register B (CRB), and in either 12 or 24-hour format, as determined by bit 1 of this register.

Note: When changing the above formats, re-initialize all the time registers.

Daylight Saving

Daylight saving time exceptions are handled automatically, as described in Table 4-18 on page 87.

Leap Years

Leap year exceptions are handled automatically by the internal calendar function. Every four years, February is extended to 29 days.

Updating

The time and calendar registers are updated once per second regardless of bit 7 (SET) of CRB. Since the time and calendar registers are updated serially, unpredictable results may occur if they are accessed during the update. Therefore, you must ensure that reading or writing to the time storage locations does not coincide with a system update of these locations. There are several methods to avoid this contention.

Method 1

1) Set bit 7 of CRB to 1. This takes a “snapshot” of the internal time registers and loads them into the user copy registers. The user copy registers are seen when accessing the RTC from outside, and are part of the double buffering mechanism. You may keep this bit set for up to 1 second, since the time/calendar chain continues to be updated once per second.
2) Read or write the required registers (since bit 1 is set, you are accessing the user copy registers). If you perform a read operation, the information you read is correct from the time when bit 1 was set. If you perform a write operation, you write only to the user copy registers.
3) Reset bit 1 to 0. During the transition, the user copy registers update the internal registers, using the double buffering mechanism to ensure that the update is performed between two regular time updates. This mechanism enables new time parameters to be loaded in the RTC.

Method 2

1) Access the RTC registers after detection of an Update Ended interrupt. This implies that an update has just been completed and 999 msec remain until the next update.
2) To detect an Update Ended interrupt, you may either:
   - Poll bit 4 of CRC.
   - Use the following interrupt routine:
     - Set bit 4 of CRB.
     - Wait for an interrupt from interrupt pin.
     - Clear the IRQF flag of CRC before exiting the interrupt routine.

Method 3

Poll bit 7 of CRA. The update occurs 244 μs after this bit goes high. Therefore, if a 0 is read, the time registers remain stable for at least 244 μs.

Method 4

Use a periodic interrupt routine to determine if an update cycle is in progress, as follows:

1) Set the periodic interrupt to the desired period.
2) Set bit 6 of CRB to enable the interrupt from periodic interrupt.
3) Wait for the periodic interrupt appearance. This indicates that the period represented by the following expression remains until another update occurs:
   [(Period of periodic interrupt / 2) + 244 μs].

4.5.3.2 Alarms

The timekeeping function can be set to generate an alarm when the current time reaches a stored alarm time. After each RTC time update (every 1 second), the seconds, minutes, hours, date of month and month counters are compared with their corresponding registers in the alarm settings. If equal, bit 5 of CRC is set. If the Alarm Interrupt Enable bit was previously set (CRB bit 5), the interrupt request pin is also active.

Any alarm register may be set to “Unconditional Match” by setting bits [7:6] to 11. This combination, not used by any BCD or binary time codes, results in a periodic alarm. The rate of this periodic alarm is determined by the registers that were set to “Unconditional Match”.

For example, if all but the seconds and minutes alarm registers are set to “Unconditional Match”, an interrupt is generated every hour at the specified minute and second. If all but the seconds, minutes and hours alarm registers are set to “Unconditional Match”, an interrupt is generated every day at the specified hour, minute, and second.

SuperI/O Module (Continued)

4.5.3.3 Power Supply

The device is supplied from two supply voltages, as shown in Figure 4-8:

- System standby power supply voltage, V_{SB}
- Backup voltage, from low capacity Lithium battery

A standby voltage, V_{SB}, from the external AC/DC power supply powers the RTC under normal conditions.

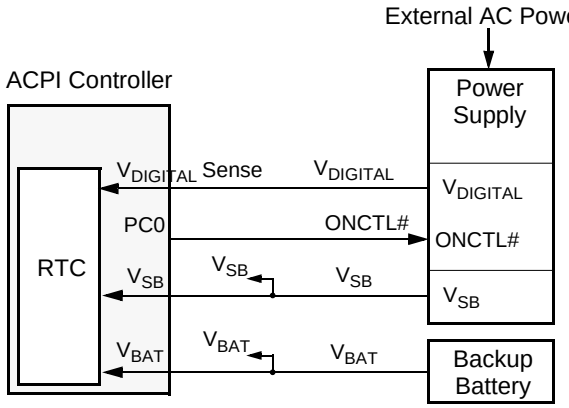
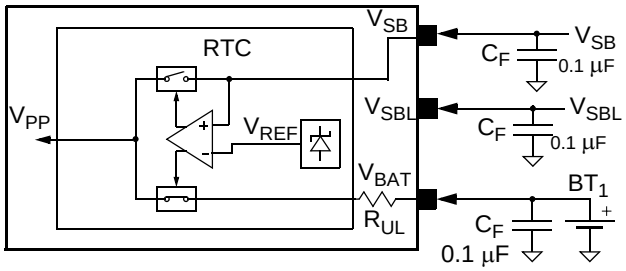


Figure 4-8. Power Supply Connections

Figure 4-9 represents a typical battery configuration. No external diode is required to meet the UL standard due to the internal switch and internal serial resistor, R_{UL}.



Note: Place a 0.1 µF capacitor on each V_{SB} and V_{SBL} power supply pin as close as possible to the pin, and also on V_{BAT}.

Figure 4-9. Typical Battery Configuration

The RTC is supplied from one of two power supplies, V_{SB} or V_{BAT}, according to their levels. An internal voltage comparator delivers the control signals to a pair of switches. Battery backup voltage V_{BAT} maintains the correct time and saves the CMOS memory when the V_{SB} voltage is absent, due to power failure or disconnection of the external AC/DC input power supply or V_{SB} main battery.

To assure that the module uses power from V_{SB} and not from V_{BAT}, the V_{SB} voltage should be maintained above its minimum, as detailed in Section 7.0 "Electrical Specifications" on page 280.

The actual voltage point where the module switches from V_{BAT} to V_{SB} is lower than the minimum workable battery voltage, but high enough to guarantee the correct functionality of the oscillator and the CMOS RAM.

Figure 4-10 shows typical battery current consumption during battery-backed operation, and Figure 4-11 during normal operation.

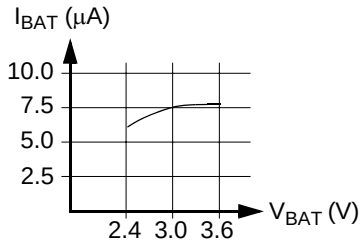
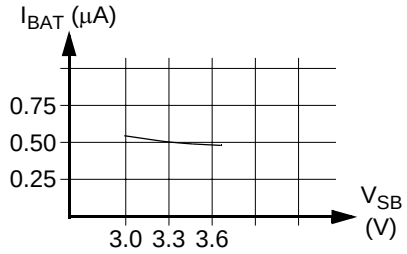


Figure 4-10. Typical Battery Current: Battery Backed Power Mode



Note: Battery voltage in this test is 3.0V.

Figure 4-11. Typical Battery Current: Normal Operation Mode

SuperI/O Module (Continued)

4.5.3.4 System Power States

The system power state may be No Power, Power On, Power Off, or Power Failure. Table 4-16 indicates the power-source combinations for each state. No other power-source combinations are valid.

In addition, the power sources and distribution for the entire system are illustrated in Figure 4-8 on page 83.

Table 4-16. System Power States

$V_{DIGITAL}$	V_{SB}	V_{BAT}	Power State
–	–	–	No Power
–	–	+	Power Failure
–	+	+ or -	Power Off
+	+	+ or -	Power On

No Power

This state exists when no external or battery power is connected to the device. This condition does not occur once a backup battery has been connected, except in the case of a malfunction.

Power On

This is the normal state when the system is active. This state may be initiated by various events in addition to the normal physical switching on of the system. In this state, the system power supply is powered by external AC power and produces $V_{DIGITAL}$ and V_{SB}. The system and the part are powered by $V_{DIGITAL}$, with the exception of the RTC logical device, which is powered by V_{SB}.

Power Off (Suspended)

This is the normal state when the system has been switched off and is not required to be active, but is still connected to a live external AC input power source. This state may be initiated directly or by software. The system is powered down. The RTC logical device remains active, powered by V_{SB}.

Power Failure

This state occurs when the external power source to the system stops supplying power, due to disconnection or power failure on the external AC input power source. The RTC continues to maintain timekeeping and RAM data under battery power (V_{BAT}), unless the oscillator stop bit was set in the RTC. In this case, the oscillator stops functioning if the system goes to battery power, and timekeeping data becomes invalid.

System Bus Lockout

During power on or power off, spurious bus transactions from the host may occur. To protect the RTC internal registers from corruption, all inputs are automatically locked out. The lockout condition is asserted when V_{SB} is lower than V_{SBON}.

Power-Up Detection

When system power is restored after a power failure or power off state (V_{SB} = 0), the lockout condition continues for a delay of 62 msec (minimum) to 125 msec (maximum) after the RTC switches from battery to system power.

The lockout condition is switched off immediately in the following situations:

- If the Divider Chain Control bits, DV[2:0], (CRA bits [6:4]) specify a normal operation mode (01x or 100), all input signals are enabled immediately upon detection of system voltage above V_{SBON}.
- When battery voltage is below V_{BATDCT} and HMR is 1, all input signals are enabled immediately upon detection of system voltage above V_{SBON}. This also initializes registers at offsets 00h through 0Dh.
- If bit 7 (VRT) of CRD is 0, all input signals are enabled immediately upon detection of system voltage above V_{SBON}.

4.5.3.5 Oscillator Activity

The RTC oscillator is active if:

- V_{SB} power supply is higher than V_{SBON}, independent of the battery voltage, V_{BAT}.

-or-

- V_{BAT} power supply is higher than V_{BATMIN}, regardless if V_{SB} is present or not.

The RTC oscillator is disabled if:

- During power-down (V_{BAT} only), the battery voltage drops below V_{BATMIN}. When this occurs, the oscillator may be disabled and its functionality cannot be guaranteed.

-or-

- Software wrote 00x to DV[2:0] bits of the CRA register and V_{SB} is removed. This disables the oscillator and decreases the power consumption from the battery connected to V_{BAT}. When disabling the oscillator, the CMOS RAM is not affected as long as the battery is present at a correct voltage level.

If the RTC oscillator becomes inactive, the following features are dysfunctional/disabled:

- Timekeeping
- Periodic interrupt
- Alarm

SuperI/O Module (Continued)

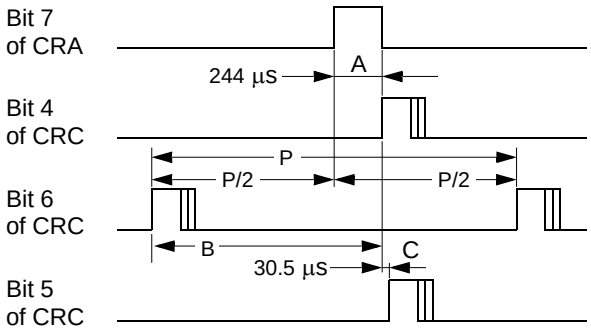
4.5.3.6 Interrupt Handling

The RTC has a single Interrupt Request line which handles the following three interrupt conditions:

- Periodic interrupt
- Alarm interrupt
- Update end interrupt

The interrupts are generated if the respective enable bits in the CRB register are set prior to an interrupt event occurrence. Reading the CRC register clears all interrupt flags. Thus, when multiple interrupts are enabled, the interrupt service routine should first read and store the CRC register, and then deal with all pending interrupts by referring to this stored status.

If an interrupt is not serviced before a second occurrence of the same interrupt condition, the second interrupt event is lost. Figure 4-12 illustrates the interrupt timing in the RTC.



Flags (and IRQ) are reset at the conclusion of CRC read or by reset.

A = Update In Progress bit high before update occurs = 244 µs

B = Periodic interrupt to update = Period (periodic int) / 2 + 244 µs

C = Update to Alarm Interrupt = 30.5 µs

P = Period is programmed by RS[3:0] of CRA

Figure 4-12. Interrupt/Status Timing

4.5.3.7 Battery-Backed RAMs and Registers

The RTC has two battery-backed RAMs and 17 registers, used by the logical units themselves. Battery-backup power enables information retention during system power-down.

The RAMs are:

- Standard RAM
- Extended RAM

The memory maps and register content of the RAMs is provided in Section 4.5.5 "RTC General-Purpose RAM Map" on page 91.

The first 14 bytes and 3 programmable bytes of the Standard RAM are overlaid by time, alarm data, and control registers. The remaining 111 bytes are general-purpose memory.

Registers with reserved bits should be written using the read-modify-write method.

All register locations within the device are accessed by the RTC Index and Data registers (at base address and base address+1). The Index register points to the register location being accessed, and the Data register contains the data to be transferred to or from the location. An additional 128 bytes of battery-backed RAM (also called Extended RAM) may be accessed via a second pair of Index and Data registers.

Access to the two RAMs may be locked. For details see Table 4-7 on page 75.

SuperI/O Module (Continued)

4.5.4 RTC Registers

The RTC registers can be accessed (see Section 4.4.2.1 "LDN 00h - Real-Time Clock" on page 74) at any time during normal operation mode (i.e.,when V_{SB} is within the recommended operation range). This access is disabled during battery-backed operation. The write operation to these registers is also disabled if bit 7 of the CRD register is 0.

Note: Before attempting to perform any start-up procedures, read about bit 7 (VRT) of the CRD register.

This section describes the RTC Timing and Control registers that control basic RTC functionality.

Table 4-17. RTC Register Map

Index	Type	Name	Reset Type
00h	R/W	**SEC.** Seconds Register	V_{PP} PUR
01h	R/W	**SECA.** Seconds Alarm Register	V_{PP} PUR
02h	R/W	**MIN.** Minutes Register	V_{PP} PUR
03h	R/W	**MINA.** Minutes Alarm Register	V_{PP} PUR
04h	R/W	**HOR.** Hours Register	V_{PP} PUR
05h	R/W	**HORA.** Hours Alarm Register	V_{PP} PUR
06h	R/W	**DOW.** Day Of Week Register	V_{PP} PUR
07h	R/W	**DOM.** Date Of Month Register	V_{PP} PUR
08h	R/W	**MON.** Month Register	V_{PP} PUR
09h	R/W	**YER.** Year Register	V_{PP} PUR
0Ah	R/W	**CRA.** RTC Control Register A	Bit specific
0Bh	R/W	**CRB.** RTC Control Register B	Bit specific
0Ch	RO	**CRC.** RTC Control Register C	Bit specific
0Dh	RO	**CRD.** RTC Control Register D	V_{PP} PUR
Programmable[1]	R/W	**DOMA.** Date of Month Alarm Register	V_{PP} PUR
Programmable[1]	R/W	**MONA.** Month Alarm Register	V_{PP} PUR
Programmable[1]	R/W	**CEN.** Century Register	V_{PP} PUR

1. Overlaid on RAM bytes in range 0Eh-7Fh. See Section 4.4.2.1 "LDN 00h - Real-Time Clock" on page 74.

SuperI/O Module (Continued)

Table 4-18. RTC Registers

Bit	Description
Index 00h Width: Byte	**Seconds Register - SEC (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Seconds Data.** Values may be 00 to 59 in BCD format or 00 to 3B in binary format.
Index 01h Width: Byte	**Seconds Alarm Register - SECA (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Seconds Alarm Data.** 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 (“11”), unconditional match is selected.
Index 02h Width: Byte	**Minutes Register - MIN (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Minutes Data.** Values can be 00 to 59 in BCD format, or 00 to 3B in binary format.
Index 03h Width: Byte	**Minutes Alarm Register - MINA (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Minutes Alarm Data.** Values can be 00 to 59 in BCD format, or 00 to 3B in binary format. When bits 7 and 6 are both set to 1, unconditional match is selected. See Section 4.5.3.2 "Alarms" on page 82 for more information about “unconditional” matches.
Index 04h Width: Byte	**Hours Register - HOR (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Hours Data.** 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.
Index 05h Width: Byte	**Hours Alarm Register - HORA (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Hours Alarm Data.** For 12-hour mode, values may 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 may be 0 to 23 in BCD format or 00 to 17 in Binary format. When bits 7 and 6 are both set to one (“11”), unconditional match is selected.
Index 06h Width: Byte	**Day of Week Register - DOW (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Day Of Week Data.** Values may be 01 to 07 in BCD format or 01 to 07 in binary format.
Index 07h Width: Byte	**Date of Month Register - DOM (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Date Of Month Data.** Values may be 01 to 31 in BCD format or 01 to 1F in binary format.
Index 08h Width: Byte	**Month Register - MON (R/W)** **Reset Type: V_{PP} PUR**
7-0	**Month Data.** Values may be 01 to 12 in BCD format or 01 to 0C in binary format.
Index 09h Width: Byte	**Year Register - YER (R/W)** **Reset Type: V_{PP} PUR**
7:0	**Year Data.** Values may be 00 to 99 in BCD format or 00 to 63 in binary format.
Index 0Ah Width: Byte This register controls test selection, among other functions. This register cannot be written before reading bit 7 of CRD.	**RTC Control Register A - CRA (R/W)** **Reset Type: Bit Specific**
7	**Update in Progress. (RO)** This bit is not affected by reset. This bit reads 0 when bit 7 of the CRB register is 1. 0: Timing registers not updated within 244 μs. 1: Timing registers updated within 244 μs.
6:4	**Divider Chain Control.** These bits control the configuration of the divider chain for timing generation and register bank selection. See Table 4-19 on page 89. They are cleared to 000 as long as bit 7 of CRD is 0.
3:0	**Periodic Interrupt Rate Select.** These bits select one of fifteen output taps from the clock divider chain to control the rate of the periodic interrupt. See Table 4-20 on page 89 and Figure 4-7 on page 81. They are cleared to 000 as long as bit 7 of CRD is 0.

SuperI/O Module (Continued)

Table 4-18. RTC Registers (Continued)

Bit	Description
Index 0Bh Width: Byte	**RTC Control Register B - CRB (R/W)** **Reset Type: Bit Specific**
7	**Set Mode.** This bit is reset at V_{PP} 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	**Periodic Interrupt.** Bits [3:0] of the CRA register determine the rate at which this interrupt is generated. It is cleared to 0 on RTC reset (i.e., hardware or software reset) or when RTC is disable. 0: Disable. 1: Enable.
5	**Alarm Interrupt.** 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 the CRD register reads 0. 0: Disable. 1: Enable.
4	**Update Ended Interrupt.** This interrupt is generated when an update occurs. It is cleared to 0 on RTC reset (i.e., hardware or software reset) or when the RTC is disable. 0: Disable. 1: Enable.
3	**Reserved.** 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.** This bit is reset at V_{PP} power-up reset only. 0: Enable BCD format. 1: Enable Binary format.
1	**Hour Mode.** This bit is reset at V_{PP} power-up reset only. 0: Enable 12-hour format. 1: Enable 24-hour format.
0	**Daylight Saving.** This bit is reset at V_{PP} power-up reset only. 0: Disable. 1: Enable: - In the spring, time advances from1:59:59 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.
Index 0Ch Width: Byte	**RTC Control Register C - CRC (RO)** **Reset Type: Bit Specific**
7	**IRQ Flag.** Mirrors the value on the interrupt output signal. When interrupt is active, IRQF is 1. To clear this bit (and deactivate the interrupt pin), read the CRC register as the flag bits UF, AF and PF are cleared after reading this register. 0: IRQ inactive. 1: Logic equation is true: ((UIE and UF) or (AIE and AF) or (PIE and PF)).
6	**Periodic Interrupt Flag.** Cleared to 0 on RTC reset (i.e., hardware or software reset) or the RTC 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	**Alarm Interrupt Flag.** Cleared to 0 as long as bit 7 of the CRD register reads 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	**Update Ended Interrupt Flag.** Cleared to 0 on RTC reset (i.e., hardware or software reset) or the RTC 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	**Reserved.**

SuperI/O Module (Continued)

Table 4-18. RTC Registers (Continued)

Bit	Description
Index 0Dh **RTC Control Register D - CRD (RO)** **Reset Type: V_{PP} PUR** Width: Byte	
7	**Valid RAM and Time.** 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) are not valid. 0: The voltage that feeds the RTC was too low. 1: RTC contents (time/calendar registers and CMOS RAM) are valid.
6:0	**Reserved.**
Index Programmable **Date of Month Alarm Register - DOMA (R/W)** **Reset Type: V_{PP} PUR** Width: Byte	
7:0	**Date of Month Alarm Data**. 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 (“11”), unconditional match is selected. (Default)
Index Programmable **Month Alarm Register - MONA (R/W)** **Reset Type: V_{PP} PUR** Width: Byte	
7:0	**Month Alarm Data.** 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 (“11”), unconditional match is selected. (Default)
Index Programmable **Century Register - CEN (R/W)** **Reset Type: V_{PP} PUR** Width: Byte	
7:0	**Century Data.** Values may be 00 to 99 in BCD format or 00 to 63 in Binary format.

Table 4-19. Divider Chain Control / Test Selection

DV2	DV1	DV0	
CRA6	**CRA5**	**CRA4**	**Configuration**
0	0	X	Oscillator Disabled
0	1	0	Normal Operation
0	1	1	Test
1	0	X	
1	1	X	Divider Chain Reset

Table 4-20. Periodic Interrupt Rate Encoding

Rate Select 3 2 1 0	Periodic Interrupt Rate (msec)	Divider Chain Output
0 0 0 0	No interrupts	
0 0 0 1	3.906250	7
0 0 1 0	7.812500	8
0 0 1 1	0.122070	2
0 1 0 0	0.244141	3
0 1 0 1	0.488281	4
0 1 1 0	0.976562	5
0 1 1 1	1.953125	6
1 0 0 0	3.906250	7
1 0 0 1	7.812500	8
1 0 1 0	15.625000	9
1 0 1 1	31.250000	10
1 1 0 0	62.500000	11
1 1 0 1	125.000000	12
1 1 1 0	250.000000	13
1 1 1 1	500.000000	14

SuperI/O Module (Continued)

Table 4-21. BCD and Binary Formats

Parameter	BCD Format	Binary Format
Seconds	00 to 59	00 to 3B
Minutes	00 to 59	00 to 3B
Hours	12-hour mode: 01 to 12 (AM) 81 to 92 (PM) 24-hour mode: 00 to 23	12-hour mode: 01 to 0C (AM) 81 to 8C (PM) 24-hour mode: 00 to 17
Day	01 to 07 (Sunday = 01)	01 to 07
Date	01 to 31	01 to 1F
Month	01 to 12 (January = 01)	01 to 0C
Year	00 to 99	00 to 63
Century	00 to 99	00 to 63

4.5.4.1 Usage Hints

1) Read bit 7 of CRD at each system power-up to validate the contents of the RTC registers and the CMOS RAM. When this bit is 0, the contents of these registers and the CMOS RAM are questionable. This bit is reset when the backup battery voltage is too low. The voltage level at which this bit is reset is below the minimum recommended battery voltage, 2.4V. Although the RTC oscillator may function properly and the register contents may be correct at lower than 2.4V, this bit is reset since correct functionality cannot be guaranteed. System BIOS may use a checksum method to revalidate the contents of the CMOS-RAM. The checksum byte should be stored in the same CMOS RAM.
2) Change the backup battery while normal operating power is present, and not in backup mode, to maintain valid time and register information. If a low leakage capacitor is connected to V_{BAT}, the battery may be changed in backup mode.
3) A rechargeable NiCd battery may be used instead of a non-rechargeable Lithium battery. This is a preferred solution for portable systems, where small size components is essential.
4) A supercap capacitor may be used instead of the normal Lithium battery. In a portable system usually the V_{SB} voltage is always present since the power management stops the system before its voltage falls too low. The supercap capacitor in the range of 0.047-0.470 F should supply the power during the battery replacement.

4.5.5 RTC General-Purpose RAM Map

Table 4-22. Standard RAM Map

Index	Description
0Eh-7Fh	Battery-backed general-purpose 111-byte RAM.

Table 4-23. Extended RAM Map

Index	Description
00h-7Fh	Battery-backed general-purpose 128-byte RAM.

4.6 SYSTEM WAKEUP CONTROL (SWC)

The SWC wakes up the system by sending a power-up request to the ACPI controller in response to the following maskable system events:

- Modem ring (RI#)
- Programmable Consumer Electronics IR (CEIR) address

Each system event that is monitored by the SWC is fed into a dedicated detector that decides when the event is active, according to predetermined (either fixed or programmable) criteria. A set of dedicated registers is used to determine the wakeup criteria, including the CEIR address.

A Wakeup Events Status Register (WKSR) and a Wakeup Events Control Register (WKCR) hold a Status bit and Enable bit, respectively, for each possible wakeup event.

Upon detection of an active event, the corresponding Status bit is set to 1. If the event is enabled (the corresponding Enable bit is set to 1), a power-up request is issued to the ACPI controller. In addition, detection of an active wakeup event may be also routed to an arbitrary IRQ.

Disabling an event prevents it from issuing power-up requests, but does not affect the Status bits. A power-up reset is issued to the ACPI controller when both the Status and Enable bits are set to 1 for at least one event type.

SWC logic is powered by V_{SB}. The SWC control and configuration registers are battery backed, powered by V_{PP}. The setup of the wakeup events, including programmable sequences, is retained throughout power failures (no V_{SB}) as long as the battery is connected. V_{PP} is taken from V_{SB} if $V_{SB} > 2.0$; otherwise, V_{BAT} is used as the V_{PP} source.

Hardware reset does not affect the SWC registers. They are reset only by a SIO software reset or power-up of V_{PP}.

4.6.1 Event Detection

CEIR Address

A CEIR transmission received on IRRX1 in a pre-selected standard (NEC, RCA or RC-5) is matched against a programmable CEIR address. Detection of matching can be used as a wakeup event. The CEIR address detection operates independently of the IR port, which is powered down with the rest of the system.

Whenever an IR signal is detected, the receiver immediately enters the Active state. When this happens, 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 received bit string is de-serialized and assembled into 8-bit characters.

The expected CEIR protocol of the received signal should be configured through bits [5:4] of the CEIR Wakeup Control register (IRWCR) (see Table 4-28 on page 95).

The CEIR Wakeup Address register (IRWAD) holds the unique address to be compared with the address contained in the incoming CEIR message. If CEIR is enabled (IRWCR[0] = 1) and an address match occurs, then the CEIR Event Status bit of WKSR is set to 1.

The CEIR Address Shift register (ADSR) holds the received address which is compared with the address contained in the IRWAD. The comparison is affected also by the CEIR Wakeup Address Mask register (IRWAM) in which each bit determines whether to ignore the corresponding bit in the IRWAD.

If CEIR routing to interrupt request is enabled, the assigned SWC interrupt request can be used to indicate that a complete address has been received. To get this interrupt when the address is completely received, IRWAM should be written with FFh. Once the interrupt is received, the value of the address can be read from ADSR.

Another parameter that is used to determine whether a CEIR signal is to be considered valid is the bit cell time width. There are four time ranges for the different protocols and carrier frequencies. Four pairs of registers (IRWTRxL and IRWTRxH) define the low and high limits of each time range. Table 4-24 lists the recommended time range limits for the different protocols and their applicable ranges. The values are represented in hexadecimal code where the units are of 0.1 msec.

Table 4-24. Time Range Limits for CEIR Protocols

Time Range	RC-5		NEC		RCA	
	Low Limit	High Limit	Low Limit	High Limit	Low Limit	High Limit
0	10h	14h	09h	0Dh	0Ch	12h
1	07h	0Bh	14h	19h	16h	1Ch
2	-	-	50h	64h	B4h	DCh
3	-	-	28h	32h	23h	2Dh

SuperI/O Module (Continued)

4.6.2 SWC Registers

The SWC registers are organized in two banks. The offsets are related to a base address that is determined by the SWC Base Address Register in the logical device configuration. The lower three registers are common to the two banks while the upper registers (03h-0Fh) are divided as follows:

- Bank 0 holds reserved registers.
- Bank 1 holds the CEIR Control registers.

The active bank is selected through the Configuration Bank Select field (bits [1:0]) in the Wakeup Configuration register (WKCFG). See Table 4-27 on page 94.

The tables that follow provide register maps and bit definitions for Banks 0 and 1.

Table 4-25. Banks 0 and 1 - Common Control and Status Register Map

Offset	Type	Name	Reset Value
00h	R/W1C	**WKSR.** Wakeup Events Status Register	00h
01h	R/W	**WKCR.** Wakeup Events Control Register	03h
02h	R/W	**WKCFG.** Wakeup Configuration Register	00h

Table 4-26. Bank 1 - CEIR Wakeup Configuration and Control Register Map

Offset	Type	Name	Reset Value
03h	R/W	**IRWCR.** CEIR Wakeup Control Register	00h
04h	---	**RSVD.** Reserved	---
05h	R/W	**IRWAD.** CEIR Wakeup Address Register	00h
06h	R/W	**IRWAM.** CEIR Wakeup Address Mask Register	E0h
07h	RO	**ADSR.** CEIR Address Shift Register	00h
08h	R/W	**IRWTR0L.** CEIR Wakeup, Range 0, Low Limit Register	10h
09h	R/W	**IRWTR0H.** CEIR Wakeup, Range 0, High Limit Register	14h
0Ah	R/W	**IRWTR1L.** CEIR Wakeup, Range 1, Low Limit Register	07h
0Bh	R/W	**IRWTR1H.** CEIR Wakeup, Range 1, High Limit Register	0Bh
0Ch	R/W	**IRWTR2L.** CEIR Wakeup, Range 2, Low Limit Register	50h
0Dh	R/W	**IRWTR2H.** CEIR Wakeup, Range 2, High Limit Register	64h
0Eh	R/W	**IRWTR3L.** CEIR Wakeup, Range 3, Low Limit Register	28h
0Fh	R/W	**IRWTR3H.** CEIR Wakeup, Range 3, High Limit Register	32h

SuperI/O Module (Continued)

Table 4-27. Banks 0 and 1 - Common Control and Status Registers

Bit	Description
Offset 00h **Wakeup Events Status Register - WKSR (R/W1C)** **Reset Value: 00h** Width: Byte This register is set to 00h on power-up of V_{PP} or software reset. It indicates which wakeup event occurred. (See Section 5.2.9.4 "Power Management Events" on page 136.)	
7	**Reserved.**
6	**Reserved.**
5	**IRRX1 (CEIR) Event Status.** This sticky bit shows the status of the CEIR event detection. This bit may be enabled only when IRRX1 is selected on the ball, otherwise results are undefined. 0: Event not detected. (Default) 1: Event detected.
4:2	**Reserved.**
1	**RI# Event Status.** This sticky bit shows the status of RI# event detection. This bit may be enabled only when RI# is selected on the ball, otherwise results are undefined. 0: Event not detected. (Default) 1: Event detected.
0	**Reserved.**
Offset 01h **Wakeup Events Control Register - WKCR (R/W)** **Reset Value: 03h** Width: Byte This register is set to 03h on power-up of V_{PP} or software reset. Detected wakeup events that are enabled issue a power-up request the ACPI controller and/or a PME to the Core Logic module. (See Section 5.2.9.4 "Power Management Events" on page 136.)	
7	**Reserved.**
6	**Reserved.** Must be set to 0.
5	**IRRX1 (CEIR) Event Enable.** This bit may be enabled only when IRRX1 is selected on the ball, otherwise results are undefined. 0: Disable. (Default) 1: Enable.
4:2	**Reserved.**
1	**RI# Event Enable.** This sticky bit shows the status of RI# event detection. This bit may be enabled only when RI# is selected on the ball, otherwise results are undefined. 0: Disable. 1: Enable. (Default)
0	**Reserved.**
Offset 02h **Wakeup Configuration Register - WKCFG (R/W)** **Reset Value: 00h** This register is set to 00h on power-up of V_{PP} or software reset. It enables access to CEIR registers.	
7:5	**Reserved.**
4	**Reserved.** Must be set to 0.
3	**Reserved.** Must be set to 0.
2	**Reserved.**
1:0	**Configuration Bank Select Bits.** 00: Only shared registers are accessible. 01: Shared registers and Bank 1 (CEIR) registers are accessible. 10: Bank selected. 1x: Reserved.

SuperI/O Module (Continued)

Table 4-28. Bank 1 - CEIR Wakeup Configuration and Control Registers

Bit	Description
Bank 1, Offset 03h	**CEIR Wakeup Control Register - IRWCR (R/W)**	**Reset Value: 00h**
Width: Byte This register is set to 00h on power-up of V_{PP} or software reset.	
7:6	**Reserved.**
5:4	**CEIR Protocol Select.** 00: RC5 01: NEC/RCA 1x: Reserved
3	**Reserved.**
2	**Invert IRRX1 Input.** 0: Not inverted. (Default) 1: Inverted.
1	**Reserved.**
0	**CEIR Enable.** 0: Disable. (Default) 1: Enable.
Bank 1, Offset 04h	**Reserved**
Bank 1, Offset 05h	**CEIR Wakeup Address Register - IRWAD (R/W)**	**Reset Value: 00h**
This register defines the station address to be compared with the address contained in the incoming CEIR message. If CEIR is enabled (bit 0 of the IRWCR register is 1) and an address match occurs, then bit 5 of the WKSR register is set to 1. This register is set to 00h on power-up of V_{PP} or software reset.	
7:0	**CEIR Wakeup Address.**
Bank 1, Offset 06h	**CEIR Wakeup Mask Register - IRWAM (R/W)**	**Reset Value: E0h**
Each bit in this register determines whether the corresponding bit in the IRWAD register takes part in the address comparison. Bits 5, 6, and 7 must be set to 1 if the RC-5 protocol is selected. This register is set to E0h on power-up of V_{PP} or software reset.	
7:0	**CEIR Wakeup Address Mask.** • If the corresponding bit is 0, the address bit is not masked (enabled for compare). • If the corresponding bit is 1, the address bit is masked (ignored during compare).
Bank 1, Offset 07h	**CEIR Address Shift Register - ADSR (RO)**	**Reset Value: 00h**
This register holds the received address to be compared with the address contained in the IRWAD register. This register is set to 00h on power-up of V_{PP} or software reset.	
7:0	**CEIR Address.**
CEIR Wakeup Range 0 Registers These two registers (IRWTR0L and IRWTR0H) define the low and high limits of time range 0 (see Table 4-24 on page 92). The values are represented in units of 0.1 msec. • RC-5 protocol: The bit cell width must fall within this range for the cell to be considered valid. The nominal cell width is 1.778 msec for a 36 KHz carrier. IRWTR0L and IRWTR0H should be set to 10h and 14h, respectively. (Default) • NEC protocol: The time distance between two consecutive CEIR pulses that encodes a bit value of 0 must fall within this range. The nominal distance for a 0 is 1.125 msec for a 38 KHz carrier. IRWTR0L and IRWTR0H should be set to 09h and 0Dh, respectively.	
Bank 1, Offset 08h	**IRWTR0L Register (R/W)**	**Reset Value: 10h**
This register is set to 10h on power-up of V_{PP} or software reset.	
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 0, Low Limit.**
Bank 1, Offset 09h	**IRWTR0H Register (R/W)**	**Reset Value: 14h**
This register is set to 14h on power-up of V_{PP} or software reset.	
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 0, High Limit.**

SuperI/O Module (Continued)

Table 4-28. Bank 1 - CEIR Wakeup Configuration and Control Registers (Continued)

Bit	Description
	CEIR Wakeup Range 1 Registers These two registers (IRWTR1L and IRWTR1H) define the low and high limits of time range 1 (see Table 4-24 on page 92). The values are represented in units of 0.1 msec. • RC-5 protocol: The pulse width defining a half-bit cell must fall within this range in order for the cell to be considered valid. The nominal pulse width is 0.889 for a 38 KHz carrier. IRWTR1L and IRWTR1H should be set to 07h and 0Bh, respectively. (Default) • NEC protocol: The time between two consecutive CEIR pulses that encodes a bit value of 1 must fall within this range. The nominal time for a 1 is 2.25 msec for a 36 KHz carrier. IRWTR1L and IRWTR1H should be set to 14h and 19h, respectively.
Bank 1, Offset 0Ah	**IRWTR1L Register (R/W)** — **Reset Value: 07h** This register is set to 07h on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 1, Low Limit.**
Bank 1, Offset 0Bh	**IRWTR1H Register (R/W)** — **Reset Value: 0Bh** This register is set to 0Bh on power-up of V_{PP} or software reset.
7:5	**Reserved.**
4:0	**CEIR Pulse Change, Range 1, High Limit.**
	CEIR Wakeup Range 2 Registers These two registers (IRWTR2L and IRWTR2H) define the low and high limits of time range 2 (see Table 4-24 on page 92). The values are represented in units of 0.1 msec. • RC-5 protocol: These registers are not used when the RC-5 protocol is selected. • NEC protocol: The header pulse width must fall within this range in order for the header to be considered valid. The nominal value is 9 msec for a 38 KHz carrier. IRWTR2L and IRWTR2H should be set to 50h and 64h, respectively. (Default)
Bank 1, Offset 0Ch	**IRWTR2L Register (R/W)** — **Reset Value: 50h** This register is set to 50h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 2, Low Limit.**
Bank 1, Offset 0Dh	**IRWTR2H Register (R/W)** — **Reset Value: 64h** This register is set to 64h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 2, High Limit.**
	CEIR Wakeup Range 3 Registers These two registers (IRWTR3L and IRWTR3H) define the low and high limits of time range 3 (see Table 4-24 on page 92). The values are represented in units of 0.1 msec. • RC-5 protocol: These registers are not used when the RC-5 protocol is selected. • NEC protocol: The post header gap width must fall within this range in order for the gap to be considered valid. The nominal value is 4.5 msec for a 36 KHz carrier. IRWTR3L and IRWTR3H should be set to 28h and 32h, respectively. (Default)
Bank 1, Offset 0Eh	**IRWTR3L Register (R/W)** — **Reset Value: 28h** This register is set to 28h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 3, Low Limit.**
Bank 1, Offset 0Fh	**IRWTR3H Register (R/W)** — **Reset Value: 32h** This register is set to 32h on power-up of V_{PP} or software reset.
7:0	**CEIR Pulse Change, Range 3, High Limit.**

4.7 ACCESS.BUS INTERFACE

The SC1100 has two ACCESS.bus (ACB) controllers. ACB is a two-wire synchronous serial interface compatible with the ACCESS.bus physical layer, Intel's SMBus, and Philips' I^2C. The ACB 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 ACB may issue a request to become the bus master.

The ACB 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.

The ACCESS.bus 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 (AB1D and AB2D) and the Serial Clock Line (AB1C and AB2C). (Here after referred to as ABD and ABC unless otherwise specified.)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 ACB initiates a data transaction with an attached ACCESS.bus compliant peripheral, the ACB becomes the master. When the peripheral responds and transmits data to the ACB, their master/slave (data transaction initiator and clock generator) relationship is unchanged, even though their transmitter/receiver functions are reversed.

This section describes the general ACB functional block. A device may include a different implementation. For device specific implementation, see Section 4.4.2.4 "LDN 05h and 06h - ACCESS.bus Ports 1 and 2" on page 78.

4.7.1 Data Transactions

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (ABC). Consequently, throughout the clock's high period, the data should remain stable (see Figure 4-13). Any changes on the ABD line during the high state of the ABC and in the middle of a transaction aborts the current transaction. New data should be sent during the low ABC 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 sections 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 ABC 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 ACCESS.bus, extend the access after each bit, thus allowing the software to handle this bit.

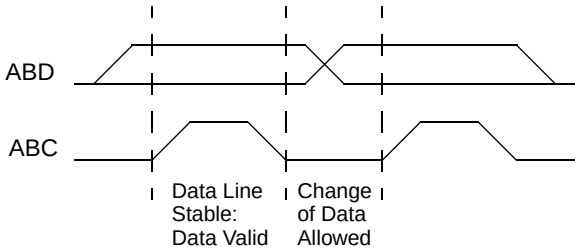


Figure 4-13. Bit Transfer

4.7.2 Start and Stop Conditions

The ACCESS.bus 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 (ABD) while the clock (ABC) is high indicates a Start Condition. A low-to-high transition of the ABD line while the ABC is high indicates a Stop Condition (Figure 4-14).

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 be accessed, or a change in the direction of data transfer.

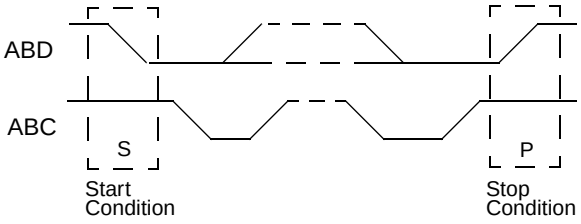


Figure 4-14. Start and Stop Conditions

SuperI/O Module (Continued)

4.7.3 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 4-15).

The master generates the ACK clock pulse on the ninth clock pulse of the byte transfer. The transmitter releases the ABD line (permits it to go high) to allow the receiver to send the ACK signal. The receiver must pull down the ABD 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 4-16 illustrates the ACK cycle.

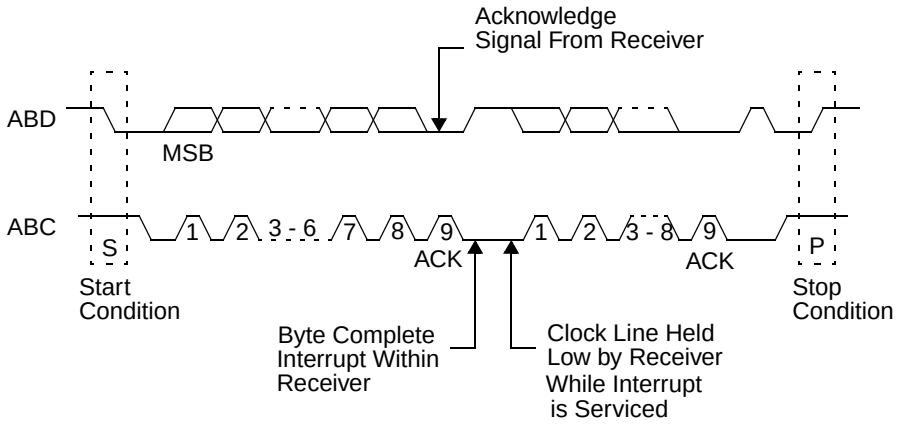


Figure 4-15. ACCESS.bus Data Transaction

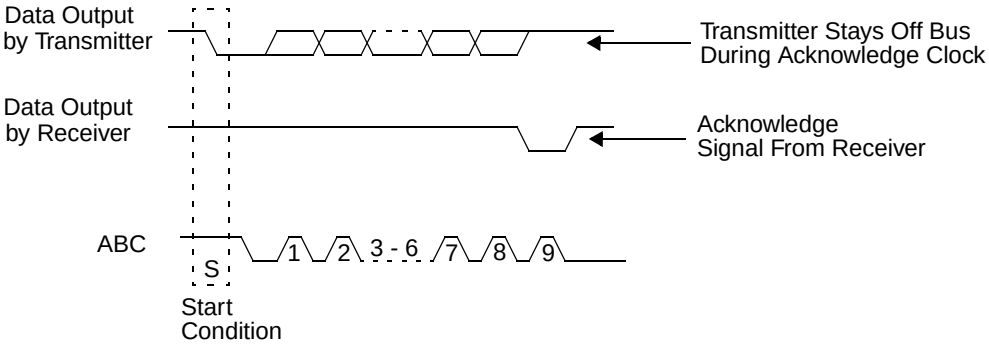


Figure 4-16. ACCESS.bus Acknowledge Cycle

SuperI/O Module (Continued)

4.7.4 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 ABD 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.

4.7.5 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 ABD line, once it recognizes its address.

The address consists of the first 7 bits after a Start Condition. The direction of the data transfer (R/W#) depends on the bit sent after the address, the eighth bit. A low-to-high transition during a ABC high period indicates the Stop Condition, and ends the transaction of ABD (see Figure 4-17).

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/W# bit (1 = Read, 0 = Write), the device acts either as a transmitter or a receiver.

The I^2C bus 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.

4.7.6 Arbitration on the Bus

Multiple master devices on the bus require arbitration between their conflicting bus access 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 ABD line differs from the value driven by the device. (An exception to this rule is ABD while receiving data. The lines may be driven low by the slave without causing an abort.)

The ABC 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 ABD to check if it is being addressed by the winning master on the bus.

4.7.7 Master Mode

Requesting Bus Mastership

An ACCESS.bus transaction starts with a master device requesting bus mastership. It asserts a Start Condition, followed by the address of the device it wants to access. 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 ACBCTL1[2] to the desired operation mode. (Polling or Interrupt) and set the ACBCTL1[0]. This causes the ACB to issue a Start Condition on the ACCESS.bus when the ACCESS.bus becomes free (ACBCST[1] is cleared, or other conditions that can delay start). It then stalls the bus by holding ABC low.
2) If a bus conflict is detected (i.e., another device pulls down the ABC signal), the ACBST[5] is set.
3) If there is no bus conflict, ACBST[1] and ACBST[6] are set.
4) If the ACBCTL1[2] is set and either ACBST[5] or ACBST[6] is set, an interrupt is issued.

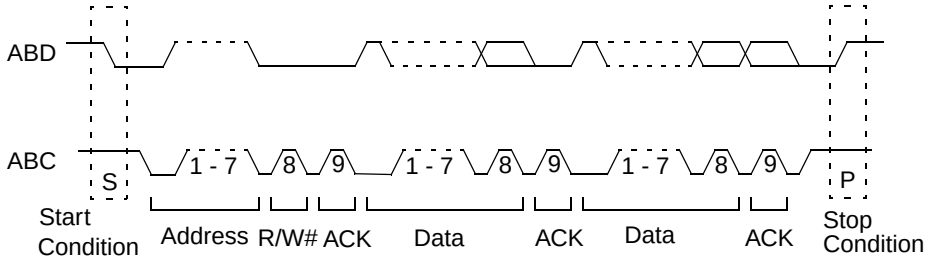


Figure 4-17. A Complete ACCESS.bus Data Transaction

SuperI/O Module (Continued)

Sending the Address Byte

When the device is the active master of the ACCESS.bus (ACBST[1] is set), it can send the address on the bus.

The address sent should not be the device's own address, as defined by ACBADDR[6:0] if ACBADDR[7] is set, nor should it be the global call address if ACBST[3] 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 ACBCTL1[4]. If only an address needs to be sent or if the device requires stall for some other reason, set ACBCTL1[7].
2) Write the address byte (7-bit target device address) and the direction bit to the ACBSDA register. This causes the ACB to generate a transaction. At the end of this transaction, the acknowledge bit received is copied to ACBST[4]. During the transaction, the ABD and ABC lines are continuously checked for conflict with other devices. If a conflict is detected, the transaction is aborted, ACBST[5] is set and ACBST[1] is cleared.
3) If ACBCTL1[7] is set and the transaction was successfully completed (i.e., both ACBST[5] and ACBST[4] are cleared), ACBST[3] is set. In this case, the ACB stalls any further ACCESS.bus operations (i.e., holds ABC low). If ACBCTL1[2] 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 ACBST[5] nor ACBST[4] is set, and no other master has accessed the device), ACBST[6] is set to indicate that the ACB awaits attention.
5) If the requested direction is receive, the start transaction was completed successfully and ACBCTL1[7] is cleared, the ACB starts receiving the first byte automatically.
6) Check that both ACBST[5] and ACBST[4] are cleared. If ACBCTL1[2] is set, an interrupt is generated when ACBST[5] or ACBST[4] is set.

Master Transmit

After becoming the bus master, the device can start transmitting data on the ACCESS.bus.

To transmit a byte in an interrupt or polling controlled operation, the software should:

1) Check that both ACBST[5] and ACBST[4] are cleared, and that ACBST[6] is set. If ACBCTL1[7] is set, also check that ACBST[3] is cleared (and clear it if required).
2) Write the data byte to be transmitted to the ACBSDA.

When either ACBST[5] or ACBST[4] is set, an interrupt is generated. When the slave responds with a negative acknowledge, ACBST[4] is set and ACBST[6] remains cleared. In this case, if ACBCTL1[2] is set, an interrupt is issued.

Master Receive

After becoming the bus master, the device can start receiving data on the ACCESS.bus.

To receive a byte in an interrupt or polling operation, the software should:

1) Check that ACBST[6] is set and that ACBST[5] is cleared. If ACBCTL1[7] is set, also check that the ACBST[3] is cleared (and clear it if required).
2) Set ACBCTL1[4] to 1, 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 the ACBSDA.

Before receiving the last byte of data, set ACBCTL1[4].

4.7.7.1 Master Stop

To end a transaction, set the ACBCTL1[1] before clearing the current stall flag (i.e., ACBST[6], ACBST[4], or ACBST[3]). This causes the ACB to send a Stop Condition immediately, and to clear ACBCTL1[1]. A Stop Condition may be issued only when the device is the active bus master (i.e., ACBST[1] is set).

Master Bus Stall

The ACB can stall the ACCESS.bus between transfers while waiting for the host response. The ACCESS.bus is stalled by holding the AB1C 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 (ACBST[4] = 1).
- ACBST[6] bit is set.
- ACBCTL1[7] = 1, after a successful start (ACBST[3] = 1).

Repeated Start

A repeated start is performed when the device is already the bus master (ACBST[1] is set). In this case, the ACCESS.bus is stalled and the ACB awaits host handling due to: negative acknowledge (ACBST[4] = 1), empty buffer (ACBST[6] = 1) and/or a stall after start (ACBST[3] 1).

For a repeated start:

1) Set ACBCTL1[0] to 1.
2) In master receive mode, read the last data item from ACBSDA.
3) Follow the address send sequence, as described in Section "Sending the Address Byte". If the ACB was awaiting handling due to ACBST[3] = 1, clear it only after writing the requested address and direction to ACBSDA.

SuperI/O Module (Continued)

Master Error Detection

The ACB 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 ACCESS.bus. If an illegal condition is detected, ACBST[5] is set, and master mode is exited (ACBST[1] is cleared).

Bus Idle Error Recovery

When a request to become the active bus master or a restart operation fails, ACBST[5] 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 ACCESS.bus may remain deadlocked.

To recover from deadlock, use the following sequence:

1) Clear ACBST[5] and ACBCST[1].
2) Wait for a timeout period to check that there is no other active master on the bus (i.e., ACBCST[1] remains cleared).
3) Disable, and re-enable the ACB 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 ACB becomes the bus master: it asserts a Start Condition, sends an address byte, then asserts a Stop Condition which synchronizes all the slaves.

4.7.8 Slave Mode

A slave device waits in idle mode for a master to initiate a bus transaction. Whenever the ACB is enabled and it is not acting as a master (i.e., ACBST[1] 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 ACBADDR[6:0] value if ACBADDR[7] = 1.

or

- The general call address if ACBCTL1[5] = 1.

This match is checked even when ACBST[1] is set. If a bus conflict (on ABD or ABC) is detected, ACBST[5] is set, ACBST[1] 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 ABD pin during the acknowledge cycle.
2) ACBCST[2] and ACBST[2] are set. If ACBST[0] = 1 (i.e., slave transmit mode) ACBST[6] is set to indicate that the buffer is empty.
3) If ACBCTL1[2] is set, an interrupt is generated if both ACBCTL1[2] and ACBCTL16 are set.
4) The software then reads ACBST[0] to identify the direction requested by the master device. It clears ACBST[2] 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 ACB extends the acknowledge clock until the software reads or writes ACBSDA. 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 ACCESS.bus by extending the first clock cycle of a transaction in the following cases:

- ACBST[6] is set.
- ACBST[2] and ACBCTL1[6] are set.

Slave Error Detection

The ACB detects illegal Start and Stop Conditions on the ACCESS.bus (i.e., a Start or Stop Condition within the data transfer or the acknowledge cycle). When this occurs, ACBST[5] is set and ACBCST[3:2] are cleared, setting the ACB as an unaddressed slave.

4.7.9 Configuration

ABD and ABC Signals

The ABD and ABC 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.

ACB Clock Frequency

The ACB permits the user to set the clock frequency for the ACCESS.bus clock. The clock is set by the ACBCTL2[7:1], which determines the ABC clock period used by the device. This clock low period may be extended by stall periods initiated by the ACB or by another ACCESS.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.

SuperI/O Module (Continued)

4.7.10 ACB Registers

Each functional block is associated with a Logical Device Number (LDN) (see Section 4.3.2 "Banked Logical Device Registers" on page 68). ACCESS.Bus Port 1 is assigned as LDN 05h and ACCESS.bus Port 2 as LDN 06h. In addition to the registers listed here, there are additional configuration registers listed in Section 4.4.2.4 "LDN 05h and 06h - ACCESS.bus Ports 1 and 2" on page 78.

Table 4-29. ACB Register Map

Offset	Type	Name	Reset Value
00h	R/W	**ACBSDA.** ACB Serial Data	xxh
01h	R/W	**ACBST.** ACB Status	00h
02h	R/W	**ACBCST.** ACB Control Status	00h
03h	R/W	**ACBCTL1.** ACB Control 1	00h
04h	R/W	**ACBADDR.** ACB Own Address	xxh
05h	R/W	**ACBCTL2.** ACB Control 2	00h

Table 4-30. ACB Registers

Bit	Description
Offset 00h Width: Byte	**ACB Serial Data Register - ACBSDA (R/W)** **Reset Value: xxh**
7:0	**ACB Serial Data.** 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 ACBSDA is allowed only when ACBST[6] is set, or for repeated starts after setting the ACBCTL1[0]. An attempt to access the register in other cases may produce unpredictable results.
Offset 01h Width: Byte	**ACB Status Register - ACBST (R/W)** **Reset Value: 00h**
This is a read register with a special clear. Some of its bits may be cleared by software, as described below. This register maintains the current ACB status. On reset, and when the ACB is disabled, ACBST is cleared (00h).	
7	**SLVSTP (Slave Stop). (R/W1C)** Writing 0 to SLVSTP is ignored. 0: Writing 1 or ACB disabled. 1: Stop Condition detected after a slave transfer in which ACBCST[2] or ACBCST[3] was set.
6	**SDAST (SDA Status). (RO)** 0: Reading from ACBSDA during a receive, or when writing to it during a transmit. When ACBCTL1[0] is set, reading ACBSDA does not clear SDAST. This enables ACB to send a repeated start in master receive mode. 1: SDA Data register awaiting data (transmit - master or slave) or holds data that should be read (receive - master or slave).
5	**BER (Bus Error). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: 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.
4	**NEGACK (Negative Acknowledge). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: Transmission not acknowledged on the ninth clock (In this case, SDAST (bit 6) is not set).
3	**STASTR (Stall After Start). (R/W1C)** Writing 0 to this bit is ignored. 0: Writing 1 or ACB disabled. 1: Address sent successfully (i.e., a Start Condition sent without a bus error, or Negative Acknowledge), if ACBCTL1[7] is set. This bit is ignored in slave mode. When STASTR is set, it stalls the ACCESS.bus by pulling down the ABC line, and suspends any further action on the bus (e.g., receive of first byte in master receive mode). In addition, if ACBCTL1[1] is set, it also causes the ACB to send an interrupt.

SuperI/O Module (Continued)

Table 4-30. ACB Registers (Continued)

Bit	Description
2	**NMATCH (New Match). (R/W1C)** Writing 0 to this bit is ignored. If ACBCTL1[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)** 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 (Transmit). (RO)** Direction bit. 0: Master/slave transmit mode not active. 1: Master/slave transmit mode active.

Offset 02h **ACB Control Status Register - ACBCST (R/W)** **Reset Value: 00h**

Width: Byte

This register configures and controls the ACB functional block. It maintains the current ACB status and controls several ACB functions. On reset and when the ACB is disabled, the non-reserved bits of ACBCST are cleared.

7:6	**Reserved.**
5	**TGABC (Toggle ABC Line). (R/W)** Enables toggling the ABC line during error recovery. 0: Clock toggle completed. 1: When the ABD line is low, writing 1 to this bit toggles the ABC line for one cycle. Writing 1 to TGABC while ABD is high is ignored.
4	**TSDA (Test ABD Line). (RO)** Reads the current value of the ABD line. It can be used while recovering from an error condition in which the ABD line is constantly pulled low by an out-of-sync slave. Data written to this bit is ignored.
3	**GCMTCH (Global Call Match). (RO)** 0: Start Condition or repeated Start and a Stop Condition (including illegal Start or Stop Condition). 1: In slave mode, ACBCTL1.GCMEN is set and the address byte (the first byte transferred after a Start Condition) is 00h.
2	**MATCH (Address Match). (RO)** 0: Start Condition or repeated Start and a Stop Condition (including illegal Start or Stop Condition). 1: ACBADDR[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 ACBADDR.
1	**BB (Bus Busy). (R/W1C)** 0: Writing 1, ACB disabled, or Stop Condition detected. 1: Bus active (a low level on either ABD or ABC), or Start Condition.
0	**BUSY. (RO)** This bit should always be written 0. This bit indicates the period between detecting a Start Condition and completing receipt of the address byte. After this, the ACB is either free or enters slave mode. 0: Completion of any state below or ACB disabled. 1: ACB is in one of the following states: -Generating a Start Condition -Master mode (ACBST[1] is set) -Slave mode (ACBCST[2] or ACBCST[3] set).

Offset 03h **ACB Control Register 1 - ACBCTL1 (R/W)** **Reset Value: 00h**

Width: Byte

7	**STASTRE (Stall After Start Enable).** 0: When cleared, ACBST[3] can not be set. However, if ACBST[3] is set, clearing STASTRE does not clear ACBST[3]. 1: Stall after start mechanism enabled, and ACB stalls the bus after the address byte.
6	**NMINTE (New Match Interrupt Enable).** 0: No interrupt issued on a new match. 1: Interrupt issued on a new match only if ACBCTL1[2] set.
5	**GCMEN (Global Call Match Enable).** 0: Global call match disabled. 1: Global call match enabled.

SuperI/O Module (Continued)

Table 4-30. ACB Registers (Continued)

Bit	Description
4	**ACK (Acknowledge).** This bit is ignored in transmit mode. When the device acts as a receiver (slave or master), this bit holds the stop transmitting instruction that is transmitted during the next acknowledge cycle. 0: Cleared after acknowledge cycle. 1: Negative acknowledge issued on next received byte.
3	**Reserved.**
2	**INTEN (Interrupt Enable).** 0: ACB interrupt disabled. 1: ACB interrupt enabled. An interrupt is generated in response to one of the following events: -Detection of an address match (ACBST[2] = 1) and ACBCTL1[6] = 1. -Receipt of Bus Error (ACBST[5] = 1). -Receipt of Negative Acknowledge after sending a byte (ACBST[4] = 1). -Acknowledge of each transaction (same as the hardware set of the ACBST[6]). -In master mode if ACBCTL1[7] = 1, after a successful start (ACBST[3] = 1). -Detection of a Stop Condition while in slave mode (ACBST[7] = 1).
1	**STOP (Stop).** 0: Automatically cleared after Stop issued. 1: Setting this bit in master mode generates a Stop Condition to complete or abort current message transfer.
0	**START (Start).** Set this bit only when in master mode or when requesting master mode. 0: Cleared after Start Condition sent or Bus Error (ACBST[5] = 1) detected. 1: Single or repeated Start Condition generated on the ACCESS.bus. If the device is not the active master of the bus (ACBST[1] = 0), setting START generates a Start Condition when the ACCESS.bus becomes free (ACBCST[1] = 0). An address transmission sequence should then be performed. If the device is the active master of the bus (ACBST[1] = 1), setting START and then writing to ACBSDA 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.

Offset 04h	**ACB Own Address Register - ACBADDR (R/W)**	**Reset Value: xxh**
Width: Byte		
7	**SAEN (Slave Address Enable).** 0: ACB does not check for an address match with ACBADDR[6:0]. 1: ACBADDR[6:0] holds a valid address and enables the match of ADDR to an incoming address byte.	
6:0	**ADDR (Address).** These bits hold the 7-bit device address of the SC1100. 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 ACBADDR[7] is 1, a match is declared.	

Offset 05h	**ACB Control Register 2 - ACBCTL2 (R/W)**	**Reset Value: 00h**
Width: Byte This register enables/disables the functional block and determines the ACB clock rate.		
7:1	**ABCFRQ (ABC Frequency).** This field defines the ABC period (low and high time) when the device serves as a bus master. The clock low and high times are defined as follows: tABCl = tABCh = 2*ABCFRQ*tCLK where tCLK is the module input clock cycle, as defined in the Section 4.2 "Module Architecture" on page 67. ABCFRQ can be programmed to values in the range of 001000b through 1111111b. Using any other value has unpredictable results.	
0	**EN (Enable).** 0: ACB is disabled, ACBCTL1, ACBST and ACBCST registers are cleared, and clocks are halted. 1: ACB is enabled.	

4.8 LEGACY FUNCTIONAL BLOCKS

This section briefly describes the following blocks that provide legacy device functions:

- Serial Port (SP), UART functionality.
- Infrared Communication Port.

Notes

- The Serial Port is similar to SCC1 in the National PC87338 device.
- The IR Communications Port is similar to SCC2 in the National PC87338 device.

The description of each Legacy block includes a general description, register maps, and bit maps. For more information about legacy blocks, contact your National Semiconductor representative.

4.8.1 UART Functionality (SP)

The generic SP supports serial data communication with a remote peripheral device or modem using a wired interface. The functional blocks can function as a standard 16450, 16550, or as an Extended UART.

4.8.1.1 UART Mode Register Bank Overview

Four register banks, each containing eight registers, control UART operation. All registers use the same 8-byte address space to indicate offsets 00h through 07h. The BSR register selects the active bank and is common to all banks. See Figure 4-18.

4.8.1.2 SP Register and Bit Maps for UART Functionality

The tables in this subsection provide register and bit maps for Banks 0 through 3.

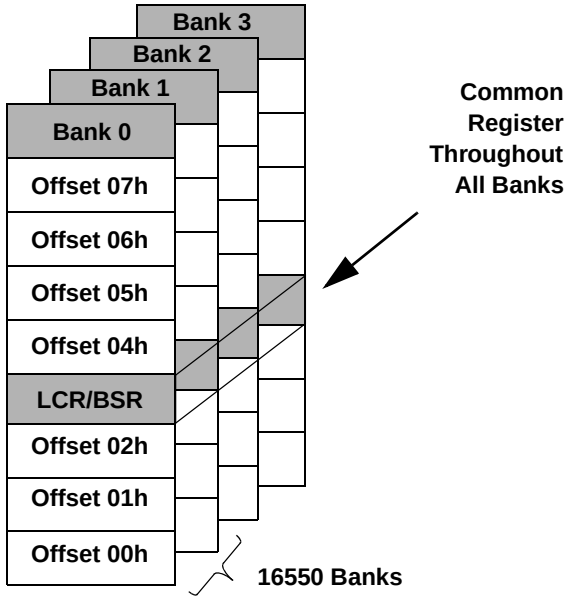


Figure 4-18. UART Mode Register Bank Architecture

Table 4-31. Bank 0 Register Map

Offset	Type	Name
00h	RO	**RXD.** Receiver Data Port
	W	**TXD.** Transmitter Data Port
01h	R/W	**IER.** Interrupt Enable
02h	RO	**EIR.** Event Identification (Read Cycles)
	W	**FCR.** FIFO Control (Write Cycles)
03h	W	**LCR**[1]**.** Line Control
	R/W	**BSR**[1]**.** Bank Select
04h	R/W	**MCR.** Modem/Mode Control
05h	RO	**LSR.** Link Status
06h	RO	**MSR.** Modem Status
07h	R/W	**SPR.** Scratchpad
	RO	**ASCR.** Auxiliary Status and Control

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-32.

SuperI/O Module (Continued)

Table 4-32. Bank Selection Encoding

BSR Bits								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Bank Selected**
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

Table 4-33. Bank 1 Register Map

Offset	Type	Name
00h	R/W	**LBGD(L).** Legacy Baud Generator Divisor Port (Low Byte)
01h	R/W	**LBGD(H).** Legacy Baud Generator Divisor Port (High Byte)
02h	---	**RSVD.** Reserved
03h	W	**LCR**[1]**.** Line Control
	R/W	**BSR**[1]**.** Bank Select
04h-07h	---	**RSVD.** Reserved

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-32 on page 106.

Table 4-34. Bank 2 Register Map

Offset	Type	Name
00h	R/W	**BGD(L).** Baud Generator Divisor Port (Low Byte)
01h	R/W	**BGD(H).** Baud Generator Divisor Port (High Byte)
02h	R/W	**EXCR1.** Extended Control 1
03h	R/W	**LCR/BSR.** Link Control/Bank Select
04h	R/W	**EXCR2.** Extended Control 2
05h	---	**RSVD.** Reserved
06h	RO	**TXFLV.** TX_FIFO Level
07h	RO	**RXFLV.** RX_FIFO Level

Table 4-35. Bank 3 Register Map

Offset	Type	Name
00h	RO	**MRID.** Module and Revision ID
01h	RO	**SH_LCR.** Shadow of LCR
02h	RO	**SH_FCR.** Shadow of FIFO Control
03h	R/W	**LCR/BSR.** Link Control/Bank Select
04h-07h	---	**RSVD.** Reserved

SuperI/O Module (Continued)

Table 4-36. Bank 0 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	RXD	RXD[7:0] (Receiver Data Bits)							
	TXD	TXD[7:0] (Transmitter Data Bits)							
01h	IER[1]	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER[2]	RSVD		TXEMP_IE	DMA_IE	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR[1]	FEN[1:0]		RSVD		RXFT	IPR1	IPR0	IPF
	EIR[2]	RSVD		TXEMP_EV	DMA_EV	MS_EV	LS_EV or TXHLT_EV	TXLDL_EV	RXHDL_EV
	FCR	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR[3]	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR[3]	BKSE	BSR[6:0] (Bank Select)						
04h	MCR[1]	RSVD			LOOP	ISEN or DCDLP	RILP	RTS	DTR
	MCR[2]	RSVD				TX_DFR	RSVD	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[1]	Scratch Data							
	ASCR[2]	RSVD	TXUR	RXACT	RXWDG	RSVD	S_OET	RSVD	RXF_TOUT

1. Non-Extended Mode.
2. Extended Mode.
3. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-32 on page 106.

Table 4-37. Bank 1 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	LBGD(L)	LBGD[7:0] (Low Byte)							
01h	LBGD(H)	LBGD[15:8] (High Byte)							
02h	RSVD	Reserved							
03h	LCR[1]	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR[1]	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	Reserved							

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-32 on page 106.

SuperI/O Module (Continued)

Table 4-38. Bank 2 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	BGD(L)	BGD[7:0] (Low Byte)							
01h	BGD(H)	BGD [15:8] (High Byte)							
02h	EXCR1	BTEST	RSVD	ETDLBK	LOOP	RSVD			EXT_SL
03h	LCR/BSR	BKSE	BSR[6:0] (Bank Select)						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RSVD			
05h	RSVD	Reserved							
06h	TXFLV	RSVD			TFL[4:0]				
07h	RXFLV	RSVD			RFL[4:0]				

Table 4-39. Bank 3 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	MRID	MID[3:0]				RID[3:0]			
01h	SH_LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
02h	SH_FCR	RXFTH[1:0]		TXFHT[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	LCR/BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	RSVD							

4.8.2 IR Communications Port (IRCP) Functionality

This section describes the IRCP support registers. The IRCP functional block provides advanced, versatile serial communications features with IR capabilities.

The IRCP also supports two DMA channels; the functional block can use either one or both of them. One channel is required for IR-based applications, since IR communication works in half duplex fashion. Two channels would normally be needed to handle high-speed full duplex IR based applications.

The IRCP signals are chosen via bit 6 of the PMR register (see Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50).

4.8.2.1 IRCP Mode Register Bank Overview

Eight register banks, each containing eight registers, control IRCP operation. All registers use the same 8-byte address space to indicate offsets 00h through 07h. The BSR register selects the active bank and is common to all banks. See Figure 4-19.

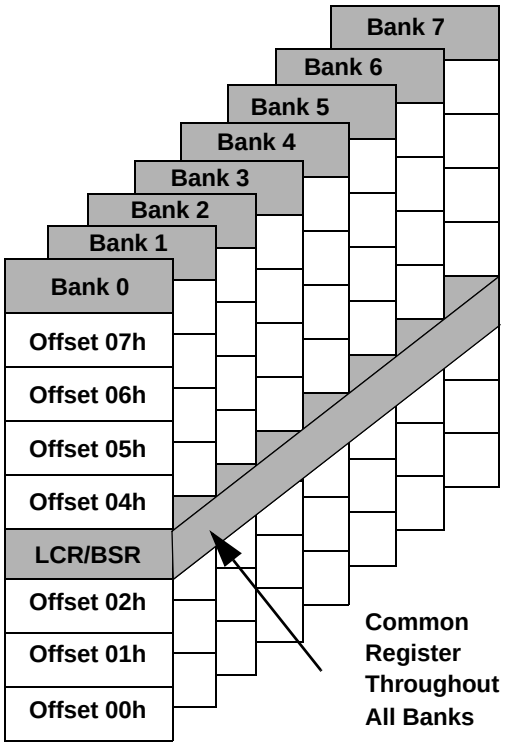


Figure 4-19. IRCP Register Bank Architecture

SuperI/O Module (Continued)

4.8.2.2 IRCP Register and Bit Maps

The tables in this subsection provide register and bit maps for Banks 0 through 7.

Table 4-40. Bank 0 Register Map

Offset	Type	Name
00h	RO	**RXD.** Receive Data Port
	W	**TXD.** Transmit Data Port
01h	R/W	**IER.** Interrupt Enable
02h	RO	**EIR.** Event Identification
	W	**FCR.** FIFO Control
03h	W	**LCR**[1]**.** Link Control
	R/W	**BSR**[1]**.** Bank Select
04h	R/W	**MCR.** Modem/Mode Control
05h	RO	**LSR.** Link Status
06h	RO	**MSR.** Modem Status
07h	R/W	**SPR.** Scratchpad
	R/W	**ASCR.** Auxiliary Status and Control

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-41.

Table 4-41. Bank Selection Encoding

BSR Bits								Bank Selected	Functionality
7	6	5	4	3	2	1	0		
0	x	x	x	x	x	x	x	0	UART + IR
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	IR Only
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	

SuperI/O Module (Continued)

Table 4-42. Bank 1 Register Map

Offset	Type	Name
00h	R/W	**LBGD(L).** Legacy Baud Generator Divisor Port (Low Byte)
01h	R/W	**LBGD(H).** Legacy Baud Generator Divisor Port (High Byte)
02h	---	**RSVD.** Reserved
03h	W	**LCR**[1]**.** Link Control
	R/W	**BSR**[1]**.** Bank Select
04h-07h	---	**RSVD.** Reserved

1. When bit 7 of this register is set to 1, bits [6:0] of BSR select the bank, as shown in Table 4-41.

Table 4-43. Bank 2 Register Map

Offset	Type	Name
00h	R/W	**BGD(L).** Baud Generator Divisor Port (Low Byte)
01h	R/W	**BGD(H).** Baud Generator Divisor Port (High Byte)
02h	R/W	**EXCR1.** Extended Control 1
03h	R/W	**BSR.** Bank Select
04h	R/W	**EXCR2.** Extended Control 2
05h	---	**RSVD.** Reserved
06h	RO	**TXFLV.** TX FIFO Level
07h	RO	**RXFLV.** RX FIFO Level

Table 4-44. Bank 3 Register Map

Offset	Type	Name
00h	RO	**MRID.** Module and Revision Identification
01h	RO	**SH_LCR.** Link Control Shadow
02h	RO	**SH_FCR.** FIFO Control Shadow
03h	R/W	**BSR.** Bank Select
04h-07h	---	**RSVD.** Reserved

SuperI/O Module (Continued)

Table 4-45. Bank 4 Register Map

Offset	Type	Name
00h	R/W	**TMR(L).** Timer (Low Byte)
01h	R/W	**TMR(H).** Timer (High Byte)
02h	R/W	**IRCR1.** IR Control 1
03h	R/W	**BSR.** Bank Select
04h	R/W	**TFRL(L).** Transmission Frame Length (Low Byte)
	RO	**TFRCC(L).** Transmission Current Count (Low Byte)
05h	R/W	**TFRL(H).** Transmission Frame Length (High Byte)
	RO	**TFRCC(H).** Transmission Current Count (High Byte)
06h	R/W	**RFRML(L).** Reception Frame Maximum Length (Low Byte)
	RO	**RFRCC(L).** Reception Frame Current Count (Low Byte)
07h	R/W	**RFRML(H).** Reception Frame Maximum Length (High Byte)
	RO	**RFRCC(H).** Reception Frame Current Count (High Byte)

Table 4-46. Bank 5 Register Map

Offset	Type	Name
00h	R/W	**SPR2.** Scratchpad 2
01h	R/W	**SPR3.** Scratchpad 3
02h	R/W	**RSVD.** Reserved
03h	R/W	**BSR.** Bank Select
04h	R/W	**IRCR2.** IR Control 2
05h	RO	**FRM_ST.** Frame Status
06h	RO	**RFRL(L).** Received Frame Length (Low Byte)
	RO	**LSTFRC.** Lost Frame Count
07h	RO	**RFRL(H).** Received Frame Length (High Byte)

Table 4-47. Bank 6 Register Map

Offset	Type	Name
00h	R/W	**IRCR3.** IR Control 3
01h	R/W	**MIR_PW.** MIR Pulse Width
02h	R/W	**SIR_PW.** SIR Pulse Width
03h	R/W	**BSR.** Bank Select
04h	R/W	**BFPL.** Beginning Flags/Preamble Length
05h-07h	---	**RSVD.** Reserved

SuperI/O Module (Continued)

Table 4-48. Bank 7 Register Map

Offset	Type	Name
00h	R/W	**IRRXDC.** IR Receiver Demodulator Control
01h	R/W	**IRTXMC.** IR Transmitter Modulator Control
02h	R/W	**RCCFG.** Consumer IR (CEIR) Configuration
03h	R/W	**BSR.** Bank Select
04h	R/W	**IRCFG1.** IR Interface Configuration 1
05h-06h	---	**RSVD.** Reserved
07h	R/W	**IRCFG4.** IR Interface Configuration 4

Table 4-49. Bank 0 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	RXD	RXD[7:0] (Receive Data)							
	TXD	TXD[7:0] (Transmit Data)							
01h	IER[1]	RSVD				MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
	IER[2]	TMR_IE	SFIF_IE	TXEMP_ IE/PLD_IE	DMA_IE	MS_IE	LS_IE	TXLDL_IE	RXHDL_IE
02h	EIR[1]	FEN[1:0]		RSVD		RXFT	IPR[1:0]		IPF
	EIR[2]	TMR_EV	SFIF_EV	TXEMP_EV/ PLD_EV	DMA_EV	MS_EV	LS_EV/ TXHLT_EV	TXLDL_EV	RXHDL_EV
	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] (Bank Select)						
04h	MCR[1]	RSVD			LOOP	ISEN/ DCDLP	RILP	RTS	DTR
	MCR[2]	MDSL[2:0]			IR_PLS	TX_DFR	DMA_EN	RTS	DTR
05h	LSR	ER_INF/ FR_END	TXEMP	TXRDY	BRK/ MAX_LEN	FE/ PHY_ERR	PE/ BAD_CRC	OE	RXDA
06h	MSR	DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS
07h	SPR[1]	Scratch Data							
	ASCR[2]	CTE/PLD	TXUR	RXACT/ RXBSY	RXWDG/ LOST_FR	TXHFE	S_EOT	FEND_INF	RXF_TOUT

1. Non-extended mode.
2. Extended mode.

SuperI/O Module (Continued)

Table 4-50. Bank 1 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	LBGD(L)	LBGD[7:0] (Low Byte Data)							
01h	LBGD(H)	LBGD[15:8] (High Byte Data)							
02h	RSVD	RSVD							
03h	LCR	BKSE	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
	BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	RSVD							

Table 4-51. Bank 2 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	BGD(L)	BGD[7:0] (Low Byte Data)							
01h	BGD(H)	BGD[15:8] (High Byte Data)							
02h	EXCR1	BTEST	RSVD	ETDLBK	LOOP	DMASWP	DMATH	DMANF	EXT_SL
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	EXCR2	LOCK	RSVD	PRESL[1:0]		RF_SIZ[1:0]		TF_SIZ[1:0]	
05h	RSVD	RSVD							
06h	TXFLV	RSVD		TFL[5:0]					
07h	RXFLV	RSVD		RFL[5:0]					

Table 4-52. Bank 3 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	MRID	MID[3:0]				RID[3:0]			
01h	SH_LCR[1]	RSVD	SBRK	STKP	EPS	PEN	STB	WLS[1:0]	
02h	SH_FCR[2]	RXFTH[1:0]		TXFTH[1:0]		RSVD	TXSR	RXSR	FIFO_EN
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h-07h	RSVD	Reserved							

1. LCR register Value.
2. FCR register Value.

SuperI/O Module (Continued)

Table 4-53. Bank 4 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	TMR(L)	TMR[7:0] (Low Byte Data)							
01h	TMR(H)	RSVD				TMR[11:8] (High Byte Data)			
02h	IRCR1	RSVD				IR_SL[1:0]		CTEST	TMR_EN
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	TFRL(L)/ TFRCC(L)	TFRL[7:0] / TFRCC[7:0] (Low Byte Data)							
05h	TFRL(H)/ TFRCC(H)	RSVD			TFRL[12:8] / TFRCC[12:8] (High Byte Data)				
06h	RFRML(L)/ RFRCC(L)	RFRML[7:0] / RFRCC[7:0] (Low Byte Data)							
07h	RFRML(H)/ RFRCC(H)	RSVD			RFRML[12:8] / RFRCC[12:8] (High Byte Data)				

Table 4-54. Bank 5 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	SPR2	Scratchpad 2							
01h	SPR3	Scratchpad 3							
02h	RSVD	RSVD							
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	IRCR2	RSVD	SFTSL	FEND_MD	AUX_IRRX	TX_MS	MDRS	IRMSSL	IR_FDPLX
05h	FRM_ST	VLD	LOST_FR	RSVD	MAX_LEN	PHY_ERR	BAD_CRC	OVR1	OVR2
06h	RFRL(L)/ LSTFRC	RFRL[7:0] (Low Byte Data) / LSTFRC[7:0]							
07h	RFRL(H)	RFRL[15:8] (High Byte Data)							

Table 4-55. Bank 6 Bit Map

Register		Bits							
Offset	**Name**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00h	IRCR3	SHDM_DS	SHMD_DS	FIR_CRC	MIR_CRC	RSVD	TXCRC_INV	TXCRC_DS	RSVD
01h	MIR_PW	RSVD				MPW[3:0]			
02h	SIR_PW	RSVD				SPW[3:0]			
03h	BSR	BKSE	BSR[6:0] (Bank Select)						
04h	BFPL	MBF[3:0]				FPL[3:0]			
05h-07h	RSVD	RSVD							

SuperI/O Module (Continued)

Table 4-56. Bank 7 Bit Map

Register		Bits							
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] (Bank Select)						
04h	IRCFG1	STRV_MS	SIRC[2:0]			IRID3	IRIC[2:0]		
05h-06h	RSVD	RSVD							
07h	IRCFG4	RSVD	IRRX_MD	IRSL0_DS	RXINV	IRSL21_DS	RSVD		

5.0 Core Logic Module

The Core Logic module is an enhanced PCI-to-Sub-ISA bridge (South Bridge), this module is ACPI-compliant, and provides AT/Sub-ISA functionality. The Core Logic module also contains state-of-the-art power management. Two bus mastering IDE controllers are included for support of up to four ATA-compliant devices. A three-port Universal Serial Bus (USB) provides high speed, and Plug & Play expansion for a variety of new consumer peripheral devices.

5.1 FEATURE LIST

Internal Fast-PCI Interface

The internal Fast-PCI bus interface is used to connect the Core Logic and GX1 modules of the SC1100. This interface includes the following features:

- PCI protocol for transfers on Fast-PCI bus
- Up to 66 MHz operation
- Subtractive decode handled internally in conjunction with external PCI bus

Bus Mastering IDE Controllers

- Two controllers with support for up to four IDE devices
- Independent timing for master and slave devices for both channels
- PCI bus master burst reads and writes
- Multiword DMA support
- Programmed I/O (PIO) Modes 0-4 support

Universal Serial Bus

- Three independent USB interfaces
- Open Host Controller Interface (OpenHCI) specification compliant

PCI Interface

- PCI 2.1 compliant
- PCI master for AC97 and IDE controllers
- Subtractive agent for unclaimed transactions
- Supports PCI initiator-to-Sub-ISA cycle translations
- PCI-to-Sub-ISA interrupt mapper/translator
- External PCI bus
  - — Devices internal to the Core Logic (IDE, Audio, USB, Sub-ISA, etc.) cannot master to memory through the external PCI bus.
  - — Legacy DMA is not supported to memory located on external PCI bus.
  - — Core Logic does not transfer subtractively decoded I/O cycles originating from external PCI bus.

AT Compatibility

- 8259A-equivalent interrupt controllers
- 8254-equivalent timer
- 8237-equivalent DMA controllers
- Port A, B, and NMI logic
- Positive decode for AT I/O space

Sub-ISA Interface

- Boot ROM chip select
- Extended ROM to 16 MB
- Two chipselects for ROM or Flash EPROM, each up to 16 MB
- Two general-purpose chip selects
- NAND Flash support
- M-Systems' DiskOnChip support
- Is not the subtractive decode agent

Power Management

- Automated CPU 0V Suspend modulation
- I/O Traps and Idle Timers for peripheral power management
- Software SMI and Stop Clock for APM support
- ACPI-compliant timer and register set
- Up to 30 GPIOs of which all can generate Power Management Interrupts (PMEs)
- Three Dedicated GPWIOs powered by V_{SBL} and V_{SB}
- Shadow register support for legacy controllers for 0V Suspend

Integrated Audio

- AC97 Version 2.0 compliant interface to audio codecs
- AMC97 codec support

Low Pin Count (LPC) Interface

- Based on Intel LPC Interface Specification Revision 1.0
- Serial IRQ support

Core Logic Module (Continued)

5.2 MODULE ARCHITECTURE AND CONFIGURATION

The Core Logic architecture provides the internal functional blocks shown in Figure 5-1.

- Fast-PCI interface to external PCI bus
- IDE controllers (UDMA-33)
- USB controllers
- Sub-ISA bus interface
- AT compatibility logic (legacy)
- ACPI compliant power management (includes GPIO interfaces, such as joystick)
- Integrated audio controller
- Low Pin Count (LPC) Interface

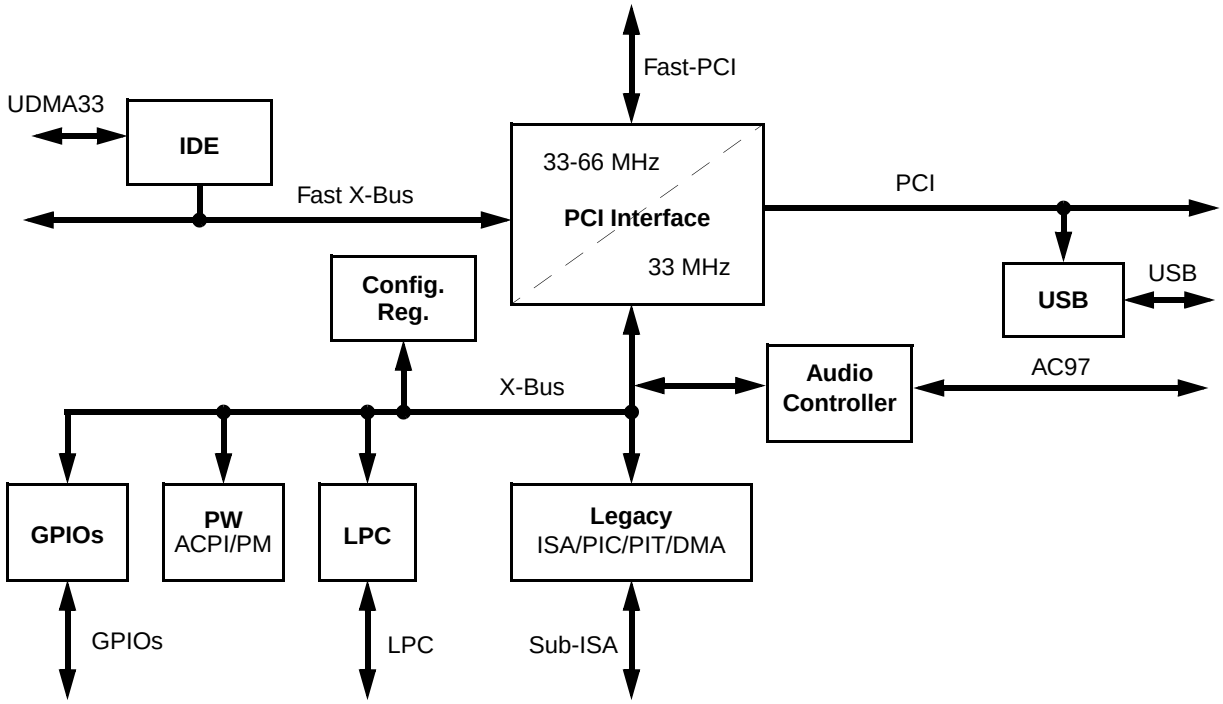


Figure 5-1. Core Logic Module Block Diagram

Core Logic Module (Continued)

5.2.1 Fast-PCI Interface to External PCI Bus

The Core Logic module provides a PCI bus interface that is both a slave for PCI cycles initiated by the GX1 module or other PCI master devices, and a non-preemptive master for DMA transfer cycles. It is also a standard PCI master for the IDE controllers and audio I/O logic. The Core Logic supports positive decode for configurable memory and I/O regions, and implements a subtractive decode option for unclaimed PCI accesses. It also generates address and data parity, and performs parity checking. The arbiter for the Fast-PCI interface is located in the GX1 module.

Configuration registers are accessed through the PCI interface using the PCI Bus Type 1 configuration mechanism as described in the PCI Specification.

5.2.1.1 Processor Mastered Cycles

The Core Logic module acts on all processor initiated cycles according to PCI rules for active/subtractive decode using DEVSEL#. Memory writes are automatically posted. Reads are retried if they are *not* destined for actively decoded (i.e., positive decode) devices on the high speed X-Bus or the 33 MHz X-Bus. This means that reads to external PCI, LPC, or Sub-ISA devices are automatically treated as delayed transactions through the PCI retry mechanism. This allows the high bandwidth devices access to the Fast-PCI interface while the response from a slow device is accumulated.

Bursting from the host is not supported.

All types of configuration cycles are supported and handled appropriately according to the PCI specification.

5.2.1.2 External PCI Mastered Cycles

Memory cycles mastered by external PCI devices on the external PCI bus are actively taken if they are within the system memory address range. Memory cycles to system memory are forwarded to the Fast-PCI interface. Burst transfers are stopped on every cache line boundary to allow efficient buffering in the Fast-PCI interface block.

I/O and configuration cycles mastered by external PCI devices which are subtractively decoded by the Core Logic module, are not handled.

5.2.1.3 Core Logic Internal or Sub-ISA Mastered Cycles

Only memory cycles (not I/O cycles) are supported by the internal Sub-ISA or legacy DMA masters. These memory cycles are always forwarded to the Fast-PCI interface.

5.2.1.4 External PCI Bus

The external PCI bus is a fully-compliant PCI bus. PCI slots are connected to this bus. Support for up to four bus masters is provided. The arbiter is in the Core Logic module.

5.2.2 PSERIAL Interface

The majority of the system power management logic is implemented in the Core Logic module, but a minimal amount of logic is contained within the GX1 module to provide information that is not externally visible.

The GX1 module implements a simple serial communications mechanism to transmit the CPU status to the Core Logic module via internal signal, PSERIAL. The GX1 module accumulates CPU events in an 8-bit register which it transmits serially every 1 to 10 µs.

The packet transmitter holds the serial output internal signal (PSERIAL) low until the transmission interval counter has elapsed. Once the counter has elapsed, the PSERIAL signal is held high for two clocks to indicate the start of packet transmission. The contents of the Serial Packet register are then shifted out starting from bit 7 down to bit 0. The PSERIAL signal is held high for one clock to indicate the end of packet transmission and then remains low until the next transmission interval. After the packet transmission is complete, the GX1 module's Serial Packet register's contents are cleared.

The GX1 module's input clock is used as the clock reference for the serial packet transmitter.

Once a bit in the register is set, it remains set until the completion of the next packet transmission. Successive events of the same type that occur between packet transmissions are ignored. Multiple unique events between packet transmissions accumulate in this register. The GX1 module transmits the contents of the serial packet only when a bit in the Serial Packet register is set and the interval counter has elapsed.

The Core Logic module decodes the serial packet after each transmission and performs the power management tasks.

For more information on the Serial Packet register refer to the *GX1 Processor Series Datasheet*.

Core Logic Module (Continued)

5.2.3 IDE Controller

The Core Logic module integrates a PCI bus mastering, ATA-4 compatible IDE controller. This controller supports UltraDMA, Multiword DMA and Programmed I/O (PIO) modes. Two devices are supported on the IDE controller. The data-transfer speed for each device can be independently programmed. This allows high-speed IDE peripherals to coexist on the same channel as lower speed devices.

The Core Logic module supports two IDE channels, a primary channel and a secondary channel.

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

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 Core Logic module also provides a software accessible buffered reset signal to the IDE drive, F0 Index 44h[3:2]. The IDE_RST# signal (ball AD6) is driven low during reset to the Core Logic module and can be driven low or high as needed for device-power-off conditions.

5.2.3.1 IDE Configuration Registers

Registers for configuring Channels 0 and 1 are located in the PCI register space designated as Function 2 (F2 Index 40h-5Ch). Table 5-35 on page 225 provides the bit formats for these registers. The IDE bus master configuration registers are accessed via F2 Index 20h which is Base Address Register 4 in Function 2 (F2BAR4). See Table 5-36 on page 229 for register/bit formats.

The following subsections discuss Core Logic operational/programming details concerning PIO, Bus Master, and UltraDMA/33 modes.

5.2.3.2 PIO Mode

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 Programmed I/O Register (F2 Index 40h)
- Channel 0 Drive 1 Programmed I/O Register (F2 Index 48h)
- Channel 1 Drive 0 Programmed I/O Register (F2 Index 50h)
- Channel 1 Drive 1 Programmed I/O Register (F2 Index 58h)

The IDE channels 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 inTable 5-36 on page 229. 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:

- F2 Index 44h[31] (Channel 0 Drive 0 — DMA Control Register) 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]) per channel 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 are only recommended settings and are not 100% tested.

5.2.3.3 Bus Master Mode

Two IDE bus masters are provided to perform the data transfers for the primary and secondary channels. The IDE controller of the Core Logic module off-loads the CPU and improves system performance in multitasking environments.

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 masters use 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.

Core Logic Module (Continued)

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.

Primary and Secondary IDE Bus Master Registers

The IDE bus master registers for each channel (primary and secondary) have an IDE Bus Master Command register and Bus Master Status register. These registers and bit formats are described in Table 5-36 on page 229.

Physical Region Descriptor Format

Each physical memory region to be transferred is described by a Physical Region Descriptor (PRD) as illustrated in Table 5-1. 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. The second DWORD contains the size (16 bits) of the buffer and the EOT flag. The EOT bit (bit 31) must be set to indicate the last PRD in the PRD table.

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. At the completion of each PRD, the bus master's next response depends on the settings of the EOT flag in the PRD. If the EOT bit is set, then the IDE bus master clears the Bus Master Active bit (Status register bit 0) and stops. If any errors occurred during the transfer, the bus master sets the Bus Master Error bit Status register bit 1).

Table 5-1. Physical Region Descriptor Format

	Byte 3								Byte 2								Byte 1								Byte 0							
DWORD	**31**	**31**	**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	E O T	Reserved															Size [15:1]															0

Core Logic Module (Continued)

5.2.3.4 UltraDMA/33 Mode

The IDE controller of the Core Logic module supports UltraDMA/33. It utilizes the standard IDE Bus Master functionality to interface, initiate, and control the transfer.The UltraDMA/33 definition also incorporates a Cyclic Redundancy Checking (CRC) error checking protocol to detect errors.

The UltraDMA/33 protocol requires no extra signal pins on the IDE connector. The IDE controller redefines three standard IDE control signals when in UltraDMA/33 mode. These definitions are shown in Table 5-2.

Table 5-2. UltraDMA/33 Signal Definitions

IDE Controller Channel Signal	UltraDMA/33 Read Cycle	UltraDMA/33 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 UltraDMA/33 operation.

IDE_IOW# is redefined 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 IDE controller. It is used by the IDE controller 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 IDE controller to the IDE device. It is the data strobe signal driven by the IDE controller 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 IDE controller 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 IDE controller 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.

UltraDMA/33 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 IDE controller asserts IDE_DACK#. When IDE_DACK# is asserted, the IDE controller drives IDE_CS0# and IDE_CS1# asserted, and IDE_ADDR[2:0] low. For write cycles, the IDE controller negates STOP, waits for the IDE device to assert DMARDY#, and then drives the first data WORD and STROBE signal. For read cycles, the IDE controller 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 Core Logic and the IDE via providing data, toggling STROBE and DMARDY#. The IDE_DATA[15:0] is latched by 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 the receiver. A burst cycle must first be paused as described above before it can be terminated. The IDE controller can then stop the burst cycle by asserting STOP, with the IDE device acknowledging by negating IDE_DREQ. The IDE device then stops the burst cycle by negating IDE_DREQ and the IDE controller acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The IDE controller then puts the result of the CRC calculation onto IDE_DATA[15:0] while deasserting 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 UltraDMA/33 transfers. The CRC value is calculated for all data by both the IDE controller and the IDE device during the UltraDMA/33 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 IDE controller drives the result of the CRC calculation onto IDE_DATA[15:0] which is then strobed by the deassertion of IDE_DACK#. The IDE device compares the CRC result of the IDE controller to its own and reports an error if there is a mismatch.

The timings for UltraDMA/33 are programmed into the DMA control registers:

- Channel 0 Drive 0 DMA Control Register (F2 Index 44h)
- Channel 0 Drive 1 DMA Control Register (F2 Index 4Ch)
- Channel 1 Drive 0 DMA Control Register (F2 Index 54h)
- Channel 1 Drive 1 DMA Control Register (F2 Index 5Ch)

The bit formats for these registers are described in Table 5-35 on page 225. Note that F2 Index 44h[20] is used to select either Multiword or UltraDMA mode. Bit 20 = 0 selects Multiword DMA mode. If bit 20 = 1, then UltraDMA/33 mode is selected. Once mode selection is made using this bit, the remaining DMA Control registers also operate in the selected mode.

Also listed in the bit formats are recommended values for both Multiword DMA Modes 0-2 and UltraDMA/33 Modes 0-2. Note that these are only recommended settings and are not 100% tested.

Core Logic Module (Continued)

5.2.4 Universal Serial Bus

The Core Logic module provides three complete, independent USB ports. Each port has a Data "Negative" and a Data "Positive" signal.

The USB ports are Open Host Controller Interface (OpenHCI) compliant. The OpenHCI specification provides a register-level description for a host controller, as well as common industry hardware/software interface and drivers.

5.2.5 Sub-ISA Bus Interface

The Sub-ISA interface of the Core Logic module is an ISA-like bus interface that is used by SC1100 to interface with Boot Flash, M-Systems' DiskOnChip or NAND EEPROM and other I/O devices. The Core Logic module is the default subtractive decoding agent and forwards all unclaimed memory and I/O cycles to the internal ISA bus. However, the Core Logic can be configured to ignore either I/O, memory, or all unclaimed cycles (subtractive decode disabled).

The Core Logic module does not support Sub-ISA refresh cycles. The refresh toggle bit in Port B still exists for software compatibility reasons.

The Sub-ISA interface includes the followings signals in addition to the signals used for an ISA interface:

- IOCS0#/IOCS1#
  - — Asserted on I/O read/write transactions from/to a programmable address range.
- DOCCS#
  - — Asserted on memory read/write transactions from/to a programmable window.
- ROMCS#
  - — Asserted on memory read/write to upper 16 MB of address space. Configurable via the ROM Mask register (F0 Index 6Eh[7:4]).
- F5BAR4CS#
  - — Asserted on memory read/write of up to 16 MB of address space. Configurable via the F5BAR4 Register (F5 Index 20h), F5BAR4 Mask Address Register (F5 Index 50h) and F5BARx Directed to Sub-ISA Register (F5 Index 59h).
- F5BAR5CS#
  - — Asserted on memory read/write of up to 16 MB of address space. Configurable via the F5BAR5 Register (F5 Index 24h), F5BAR5 Mask Address Register (F5 Index 54h) and F5BARx Directed to Sub-ISA Register (F5 Index 59h).
- DOCR#
  - — DOCR# is asserted on memory read transactions from DOCCS# window (i.e., when both DOCCS# and MEMR# are active, DOCR# is active; otherwise, it is inactive).
- DOCW
  - — DOCW is asserted on memory write transactions to DOCCS window (i.e., when both DOCCS# and MEMW# are active, DOCW is active; otherwise, it is inactive).
- RD#, WR#
  - — The signals IOR#, IOW#, MEMR#, and MEMW# are combined into two signals: RD# is asserted on I/O read or memory read; WR# is asserted on I/O write or memory write.

Memory devices that use ROMCS#, DOCCS#, F5BAR4CS#, or F5BAR5CS# as their chip select signal can be configured to support an 8-bit or 16-bit data bus. Such devices can also be configured as zero wait state devices (regardless of the data bus width). Programming of these features is via the Miscellaneous Configuration Register (MCR) in the General Configuration Block. For MCR register bit descriptions, see Table 3-2 on page 50.

I/O peripherals that use IOCS0# or IOCS1# as their chip select signal can be configured to support an 8-bit or 16-bit data bus. Such devices can also be configured as zero wait state devices (for 8-bit peripherals) via the MCR register. For MCR register bit descriptions, see Table 3-2 on page 50.

Other memory devices and I/O peripherals must be 8-bit devices; their transactions can not be with zero wait states

The Boot Flash supported by the SC1100 can be up to 16 MB. It is supported with the ROMCS# signal.

All unclaimed memory and I/O cycles are forwarded to the Internal ISA bus if subtractive decode is enabled.

The DiskOnChip chip select signal (DOCCS#) is asserted on any memory read or memory write transaction from/to a programmable address range. The address range is programmable via the DOCCS# Base Address and Control registers (F0 Index 78h and 7Ch). The base address must be on an address boundary, the size of the range.

Signal DOCCS# can also be used to interface to NAND Flash devices together with signals DOCW# and DOCR#. See application note *Geode™ SC1200/SC2200/SC3200 IAOC Devices: External NAND Flash Memory Circuit* for details.

Core Logic Module (Continued)

5.2.5.1 Sub-ISA Bus Cycles

The ISA bus controller issues multiple ISA cycles to satisfy PCI transactions that are larger than 16 bits. A full 32-bit read or write results in two 16-bit ISA transactions or four 8-bit ISA transactions. The ISA controller gathers the data from multiple ISA read cycles and returns TRDY# to the PCI bus.

SA[23:0] are a concatenation of ISA LA[23:17] and SA[19:0] and perform equivalent functionality at a reduced pin count.

Figure 5-2 shows the relationship between a PCI cycle and the corresponding ISA cycle generated.

Note: Not all signals described in Figure 5-2 are available externally. See Section 2.4.5 "Sub-ISA Interface Signals" on page 41 for more information about which Sub-ISA signals are externally available on the SC1100.

5.2.5.2 Sub-ISA Support of Delayed PCI Transactions

Multiple PCI cycles occur for every slower ISA cycle. This prevents slow PCI cycles from occupying too much bandwidth and allows access to other PCI traffic. Figure 5-3 on page 124 shows the relationship of PCI cycles to an ISA cycle with PCI delayed transactions enabled.

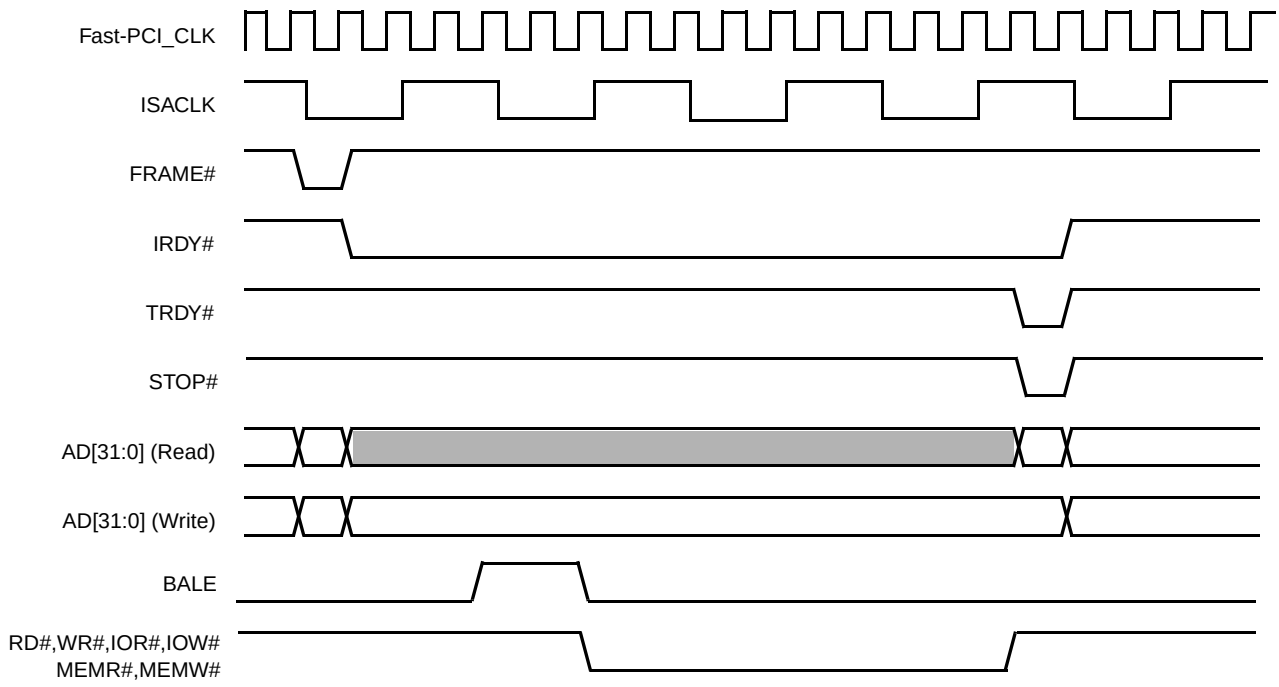


Figure 5-2. Non-Posted Fast-PCI to ISA Access

Core Logic Module (Continued)

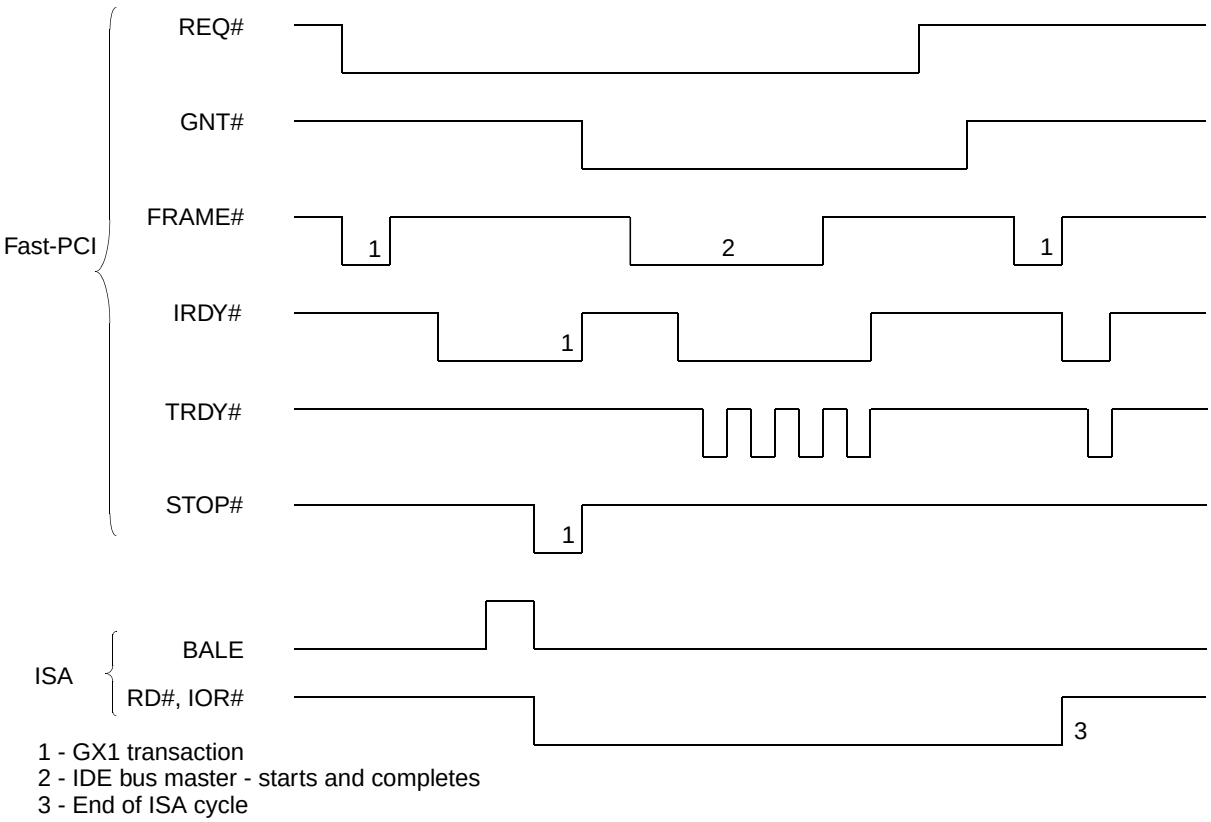


Figure 5-3. PCI to ISA Cycles with Delayed Transaction Enabled

5.2.5.3 Sub-ISA Bus Data Steering

The Core Logic module performs all of the required data steering from SD[7:0] to SD[15:0] during normal 8-bit ISA cycles, as well as during DMA and ISA master cycles. It handles data transfers between the 32-bit PCI data bus and the ISA bus. 8/16-bit devices can reside on the ISA bus. Various PC-compatible I/O registers, DMA controller registers, interrupt controller registers, and counter/timer registers lie on the on-chip I/O data bus. Either the PCI bus master or the DMA controllers can become the bus owner.

When the PCI bus master is the bus owner, the Core Logic module data steering logic provides data conversion necessary for 8/16/32-bit transfers to and from 8/16-bit devices on either the Sub-ISA bus or the 8-bit registers on the on-chip I/O data bus. When PCI data bus drivers of the Core Logic module are in TRI-STATE, data transfers between the PCI bus master and PCI bus devices are handled directly via the PCI data bus.

When the DMA requestor is the bus owner, the Core Logic module allows 8/16-bit data transfer between the Sub-ISA bus and the PCI data bus.

5.2.5.4 I/O Recovery Delays

In normal operation, the Core Logic module inserts a delay between back-to-back ISA I/O cycles that originate on the PCI bus. The default delay is four ISACLK cycles. Thus, the second of consecutive I/O cycles is held in the ISA bus controller until this delay count has expired. The delay is measured between the rising edge of IOR#/IOW# and the falling edge of BALE. This delay can be adjusted to a greater delay through the ISA I/O Recovery Control register (F0 Index 51h).

Note: This delay is not inserted for a 16-bit Sub-ISA I/O access that is split into two 8-bit I/O accesses.

Core Logic Module (Continued)

5.2.5.5 ISA DMA

DMA transfers occur between ISA I/O peripherals and system memory (i.e., not available externally). The data width can be either 8 or 16 bits. Out of the seven DMA channels available, four are used for 8-bit transfers while the remaining three are used for 16-bit transfers. One byte or WORD is transferred in each DMA cycle.

Note: The Core Logic module does not support DMA transfers to ISA memory.

The ISA DMA device initiates a DMA request by asserting one of the DRQ[7:5, 3:0] signals. When the Core Logic module receives this request, it sends a bus grant request to the PCI arbiter. After the PCI bus has been granted, the respective DACK# is driven active.

The Core Logic module generates PCI memory read or write cycles in response to a DMA cycle. Figure 5-4 and Figure 5-5 are examples of DMA memory read and memory write cycles. Upon detection of the DMA controller's MEMR# or MEMW# active, the Core Logic module starts the PCI cycle, asserts FRAME#, and negates an internal IOCHRDY. This assures the DMA cycle does not complete before the PCI cycle has provided or accepted the data. IOCHRDY is internally asserted when IRDY# and TRDY# are sampled active.

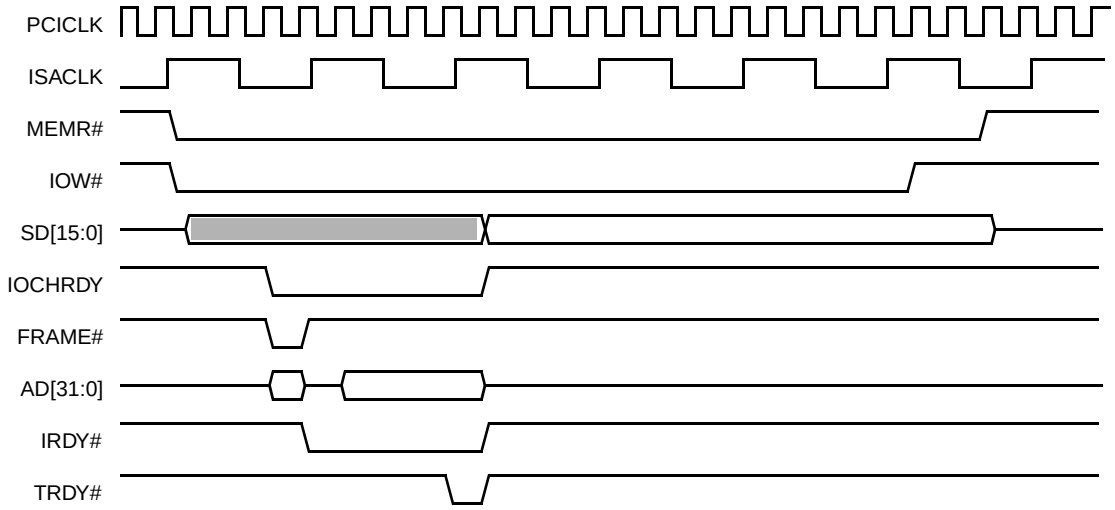


Figure 5-4. ISA DMA Read from PCI Memory

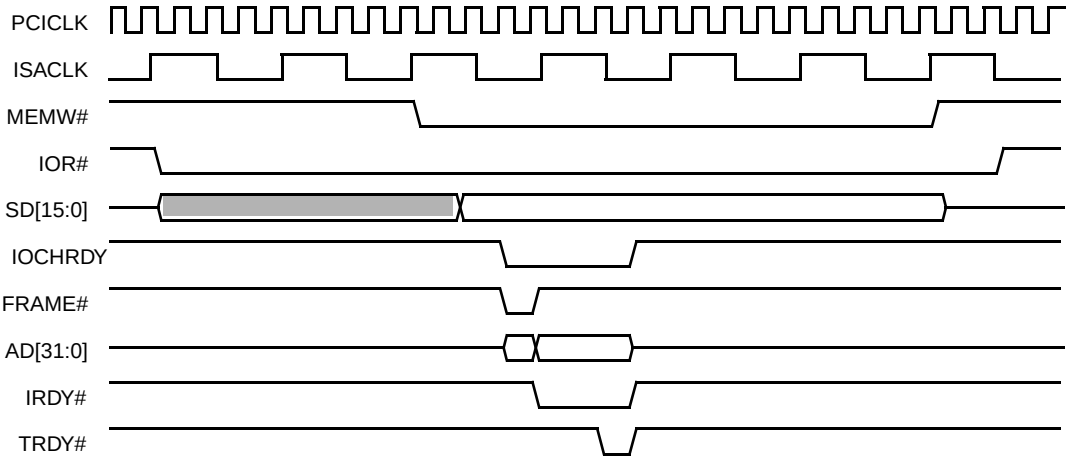


Figure 5-5. ISA DMA Write to PCI Memory

Core Logic Module (Continued)

5.2.5.6 ROM Interface

The Core Logic module positively decodes memory addresses 000F0000h-000FFFFFh (64 KB) and FFFC0000h-FFFFFFFFh (256 KB) at reset. These memory cycles cause the Core Logic module to claim the cycle, and generate an ISA bus memory cycle with ROMCS# asserted. The Core Logic module can also be configured to respond to memory addresses FF000000h-FFFFFFFFh (16 MB) and 000E0000h-000FFFFFh (128 KB).

8- or 16-bit wide ROM is supported. BOOT16 strap determines the width after reset. MCR[14,3] (Offset 34h) in the General Configuration Block (see Table 3-2 on page 50 for bit details) allows program control of the width.

Flash ROM is supported in the Core Logic module by enabling the ROMCS# signal on write accesses to the ROM region. Normally only read cycles are passed to the ISA bus, and the ROMCS# signal is suppressed for write cycles. When the ROM Write Enable bit (F0 Index 52h[1]) is set, a write access to the ROM address region causes a write cycle to occur with MEMW#, WR# and ROMCS# asserted.

5.2.5.7 PCI and Sub-ISA Signal Cycle Multiplexing

The SC1100 multiplexes most PCI and Sub-ISA signals on the balls listed in Table 5-3, in order to reduce the number of balls on the device. Cycle multiplexing is on a bus-cycle by bus-cycle basis (see Figure 5-6 on page 127), where the internal Core Logic PCI bridge arbitrates between PCI cycles and Sub-ISA cycles. Other PCI and Sub-ISA signals remain non-shared, however, some Sub-ISA signals may be muxed with GPIO.

Sub-ISA cycles are only generated as a result of GX1 module accesses to the following addresses or conditions:

- ROMCS# address range.
- DOCCS# address range.
- F5BAR4CS address range.
- F5BAR5CS address range.
- IOCS0# address range.
- IOCS1# address range.
- An I/O write to address 80h or to 84h.

If the Sub-ISA and PCI bus have more than four components, the Sub-ISA components can be buffered using 74HCT245 or 74FCT245 type transceivers. The RD# (an AND of IOR#, MEMR#) signal can be used as DIR control while TRDE# is used as enable control.

Table 5-3. Cycle Multiplexed PCI / Sub-ISA Balls

PCI	Sub-ISA	Ball No.
AD0	A0	E25
AD1	A1	F26
AD2	A2	F25
AD3	A3	G26
AD4	A4	G25
AD5	A5	H26
AD6	A6	H25
AD7	A7	D24
AD8	A8	E23
AD9	A9	J26
AD10	A10	J25
AD11	A11	K25
AD12	A12	K26
AD13	A13	F23
AD14	A14	F24
AD15	A15	L26
AD16	A16	R25
AD17	A17	K24
AD18	A18	T26
AD19	A19	T25
AD20	A20	L24
AD21	A21	U26
AD22	A22	U25
AD23	A23	M24
AD24	D0	V25
AD25	D1	W26
AD26	D2	N24
AD27	D3	P24
AD28	D4	W25
AD29	D5	Y26
AD30	D6	R24
AD31	D7	Y25
C/BE0#	D8	E24
C/BE1#	D9	G24
C/BE2#	D10	R26
C/BE3#	D11	V26
PAR	D12	H24
TRDY#	D13	N25
IRDY#	D14	P26
STOP#	D15	M25
DEVSEL#	BHE#	N26

Core Logic Module (Continued)

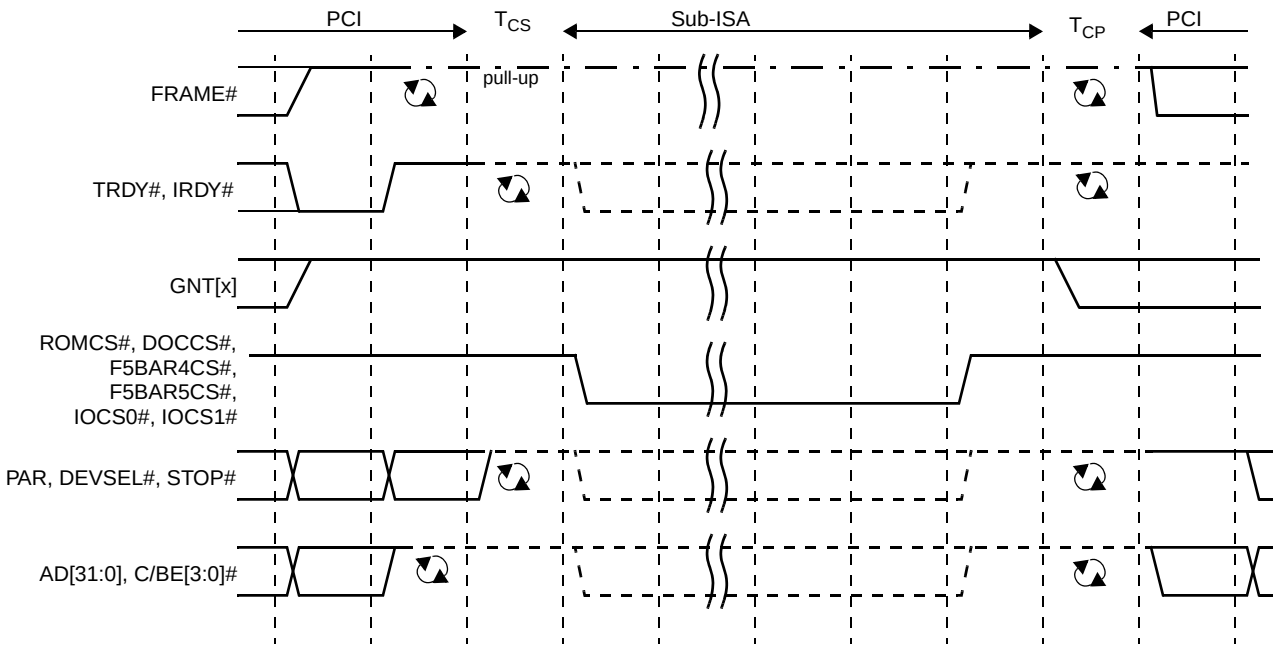


Figure 5-6. PCI Change to Sub-ISA and Back

5.2.6 AT Compatibility Logic

The Core Logic module integrates:

- Two 8237-equivalent DMA controllers with full 32-bit addressing
- Two 8259A-equivalent interrupt controllers providing 13 individually programmable external interrupts
- An 8254-equivalent timer for refresh, timer, and speaker logic
- NMI control and generation for PCI system errors and all parity errors
- Support for standard AT keyboard controllers
- Positive decode for the AT I/O register space
- Reset control

5.2.6.1 DMA Controller

The Core Logic module supports industry standard DMA architecture using two 8237-compatible DMA controllers in cascaded configuration. The DMA functions supported by the Core Logic module include:

- Standard seven-channel DMA support
- 32-bit address range support via high page registers
- IOCHRDY extended cycles for compatible timing transfers
- Internal Sub-ISA bus master device support using cascade mode
- NMI control and generation for PCI system errors and all parity errors.

Note: DMA interface signals are not available externally.

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 PCI memory or 8/16-bit ISA memory. Using the high and low page address registers, a full 32-bit PCI 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 contains Channels 4 through 7. Channel 4 is used to cascade DMA Controller 1, so it is not available externally. Channels 5 through 7 support 16-bit I/O adapters to transfer data between 16-bit I/O adapters and 16-bit system memory. Using the high and low page address registers, a full 32-bit PCI 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 128 KB pages. Channels 5, 6, and 7 transfer 16-bit WORDs on even byte boundaries only.

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.

Core Logic Module (Continued)

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 *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.

DMA Controller Registers

The DMA controller can be programmed with standard I/O cycles to the standard register space for DMA. The I/O addresses for the DMA controller registers are listed Table 5-43 on page 265.

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.

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 Core Logic module reads data from memory and writes it to the I/O device associated with the DMA channel.

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

The *verify* transfer type causes the Core Logic module 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.

DMA Priority

The DMA controller may be programmed for two types of priority schemes: *fixed* and *rotate* (I/O Ports 008h[4] and 0D0h[4] - see Table 5-43 on page 265).

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.

When programming the 16-bit channels (Channels 5, 6, and 7), the address which is written to the base address register must be the real address divided by two. Also, the base WORD Count for the 16-bit channels is the number of 16-bit WORDs to be transferred, not the number of bytes as is the case for the 8-bit channels.

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.

DMA Shadow Registers

The Core Logic module contains a shadow register located at F0 Index B8h (Table 5-29 on page 166) for reading the configuration of the DMA controllers. This read only register can sequence to read through all of the DMA registers.

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.

Core Logic Module (Continued)

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.

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.

BHE# is configured as an output during all DMA operations. It is driven as the inversion of AD0 during 8-bit DMA cycles and forced low for all 16-bit DMA cycles.

5.2.6.2 Programmable Interval Timer

The Core Logic module contains an 8254-equivalent Programmable Interval Timer (PIT) configured as shown in Figure 5-7. The PIT has three timers/counters, each with an input frequency of 1.19318 MHz (OSC divided by 12), and individually programmable to different modes.

The gates of Counter 0 and 1 are usually enabled, however, they can be controlled via F0 Index 50h. The gate of Counter 2 is connected to I/O Port 061h[0]. The output of Counter 0 is connected internally to IRQ0. This timer is typically configured in Mode 3 (square wave output), and used to generate IRQ0 at a periodic rate to be used as a system timer function. The output of Counter 1 is connected to I/O Port 061h[4]. The reset state of I/O Port 061h[4] is 0 and every falling edge of Counter 1 output causes I/O Port 061h[4] to flip states. The output of Counter 2 is brought out to the PC_BEEP output. This output is gated with I/O Port 061h[1].

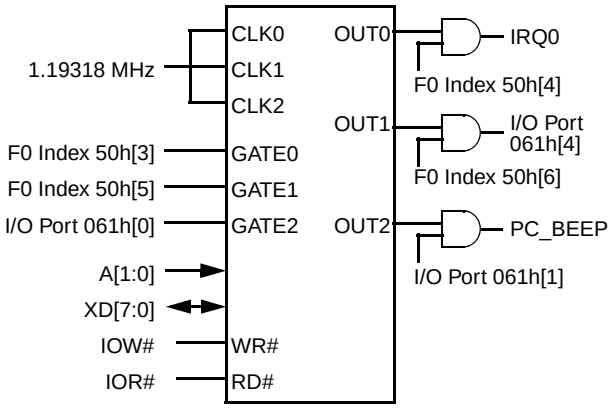


Figure 5-7. PIT Timer

PIT Shadow Register

The PIT registers are shadowed to allow for 0V Suspend to save/restore the PIT state by reading the PIT's counter and *write only* registers. The read sequence for the shadow register is listed in F0 Index BAh (see Table 5-29 on page 166).

Core Logic Module (Continued)

5.2.6.3 Programmable Interrupt Controller

The Core Logic module contains two 8259A-equivalent programmable interrupt controllers, with eight interrupt request lines each, for a total of 16 interrupts. The two controllers are cascaded internally, and two of the interrupt request inputs are connected to the internal circuitry. This allows a total of 13 externally available interrupt requests. See Figure 5-9.

Each Core Logic IRQ signal can be individually selected as edge- or level-sensitive. The four PCI interrupt signals may be routed internally to any PIC IRQ.

.

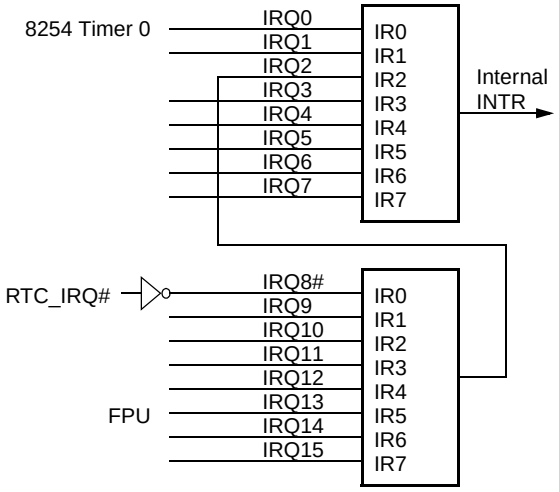


Figure 5-8. PIC Interrupt Controllers

Two interrupts are available externally depending upon selected ball multiplexing:

1) IRQ15 (ball AF19, muxed with GPIO11+RI#),

2) IRQ14 (ball AC6)

More of the IRQs are available through the use of SERIRQ (ball A24, muxed with GPIO39) function. See Table 5-4.

Table 5-4. PIC Interrupt Mapping

Master IRQ	Mapping
IRQ0	Connected to the OUT0 (system timer) of the internal 8254 PIT.
IRQ2	Connected to the slave's INTR for a cascaded configuration.
IRQ8#	Connected to internal real-time clock.
IRQ13	Connected to the FPU interface of the GX1 module.
IRQ15	Interrupts available to other functions
IRQ14	
IRQ12	
IRQ11	
IRQ10	
IRQ9	
IRQ7	
IRQ6	
IRQ5	
IRQ4	
IRQ3	
IRQ1	

The Core Logic module allows PCI interrupt signals INTA# (ball AD26), INTB# (ball W24), INTC# (ball Y24, muxed with GPIO19) and INTD# (ball V24) to be routed internally to any IRQ signal. The routing can be modified through Core Logic module's configuration registers. If this is done, the IRQ input must be configured to be level- rather than edge-sensitive. IRQ inputs may be individually programmed to be active low, level-sensitive with the Interrupt Sensitivity configuration registers at I/O address space 4D0h and 4D1h. PCI interrupt configuration is discussed in further detail in "PCI Compatible Interrupts" on page 131.

Core Logic Module (Continued)

PIC Interrupt Sequence

A typical AT-compatible interrupt sequence is as follows. Any unmasked interrupt generates the internal INTR signal to the CPU. The interrupt controller then responds to the interrupt acknowledge (INTA) cycles from the CPU. On the first INTA cycle the cascading priority is resolved to determine which of the two 8259A controllers output the interrupt vector onto the data bus. On the second INTA cycle the appropriate 8259A controller drives the data bus with the correct interrupt vector for the highest priority interrupt.

By default, the Core Logic module responds to PCI INTA cycles because the system interrupt controller is located within the Core Logic module. This may be disabled with F0 Index 40h[0]. When the Core Logic module responds to a PCI INTA cycle, it holds the PCI bus and internally generates the two INTA cycles to obtain the correct interrupt vector. It then asserts TRDY# and returns the interrupt vector.

PIC I/O Registers

Each PIC contains registers located in the standard I/O address locations, as shown in Table 5-46 "Programmable Interrupt Controller Registers" on page 273.

An initialization sequence must be followed to program the interrupt controllers. The sequence is started by writing Initialization Command Word 1 (ICW1). After ICW1 has been written, the controller expects the next writes to follow in the sequence ICW2, ICW3, and ICW4 if it is needed. The Operation Control Words (OCW) can be written after initialization. The PIC must be programmed before operation begins.

Since the controllers are operating in cascade mode, ICW3 of the master controller should be programmed with a value indicating that the IRQ2 input of the master interrupt controller is connected to the slave interrupt controller rather than an I/O device as part of the system initialization code. In addition, ICW3 of the slave interrupt controller should be programmed with the value 02h (slave ID) and corresponds to the input on the master controller.

PIC Shadow Register

The PIC registers are shadowed to allow for 0V Suspend to save/restore the PIC state by reading the PICs *write only* registers. A write to this register resets the read sequence to the first register. The read sequence for the shadow register is listed in F0 Index B9h.

PCI Compatible Interrupts

The Core Logic module allows the PCI interrupt signals INTA#, INTB#, INTC#, and INTD# (also known in industry terms as PIRQx#) to be mapped internally to any IRQ signal with the PCI Interrupt Steering registers 1 and 2, F0 Index 5Ch and 5Dh.

PCI interrupts are low-level sensitive, whereas PC/AT interrupts are positive-edge sensitive; therefore, the PCI interrupts are inverted before being connected to the 8259A.

Although the controllers default to the PC/AT-compatible mode (positive-edge sensitive), each IRQ may be individually programmed to be edge or level sensitive using the Interrupt Edge/Level Sensitivity registers in I/O Port 4D0h and 4D1h. However, if the controllers are programmed to be level-sensitive via ICW1, all interrupts must be level-sensitive. Figure 5-9 shows the PCI interrupt mapping for the master/slave 8259A interrupt controller.

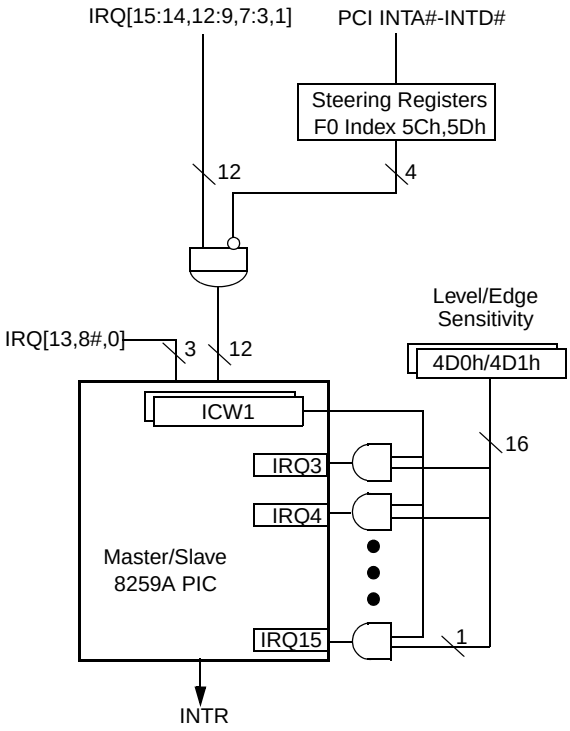


Figure 5-9. PCI and IRQ Interrupt Mapping

Core Logic Module (Continued)

5.2.7 I/O Ports 092h and 061h System Control

The Core Logic module supports control functions of I/O Ports 092h (Port A) and 061h (Port B) for PS/2 compatibility. I/O Port 092h allows a fast assertion of the A20M# or CPU_RST. (CPU_RST is an internal signal that resets the CPU. It is asserted for 100 µs after the negation of POR#.) I/O Port 061h controls NMI generation and reports system status.The Core Logic module generates an SMI for every internal change of the A20M# state and the SMI handler sets the A20M# state inside the GX1 module. This method is used for both the Port 092h (PS/2) and Port 061h (keyboard) methods of controlling A20M#.

5.2.7.1 I/O Port 092h System Control

I/O Port 092h allows for a fast keyboard assertion of an A20# SMI and a fast keyboard CPU reset. Decoding for this register may be disabled via F0 Index 52h[3].

The assertion of a fast keyboard A20# SMI is controlled by either I/O Port 092h or by monitoring for the keyboard command sequence (see Section 5.2.8.1 "Fast Keyboard Gate Address 20 and CPU Reset" on page 133). If bit 1 of I/O Port 092h is cleared, the Core Logic module internally asserts an A20M#, which in turn causes an SMI to the GX1 module. If bit 1 is set, A20M# is internally deasserted, again causing an SMI.

The assertion of a fast keyboard reset (WM_RST SMI) is controlled by bit 0 in I/O Port 092h or by monitoring for the keyboard command sequence (write data = FEh to I/O port 64h). If bit 0 is changed from 0 to 1, the Core Logic module generates a reset to the GX1 module by generating a WM_RST SMI. When the WM_RST SMI occurs, the BIOS jumps to the Warm Reset vector. Note that Warm Reset is not a pin, it is under SMI control.

5.2.7.2 I/O Port 061h System Control

Through I/O Port 061h, the speaker output can be enabled, the status of IOCHK and SERR can be read, and the state of the speaker data (Timer2 output) and refresh toggle (Timer1 output) can be read back.

5.2.7.3 SMI Generation for NMI

Figure 5-10 shows how the Core Logic module can generate an SMI for an NMI. Note that NMI is not a pin.

Core Logic Module (Continued)

5.2.8 Keyboard Support

The Core Logic module can actively decode the keyboard controller I/O Ports 060h, 062h, 064h and 066h, and generate an LPC bus cycle. Keyboard positive decoding can be disabled if F0 Index 5Ah[1] is cleared (i.e., subtractive decoding enabled).

Access to I/O Ports 060h and 064h on Sub-ISA can be enabled with ROMCS# asserted, by setting bit F0 Index 53h[7]. The Core Logic module will also actively decode the keyboard controller I/O Ports 062h and 066h if F0 Index 5Bh[7] is set.

5.2.8.1 Fast Keyboard Gate Address 20 and CPU Reset

The Core Logic module monitors the keyboard I/O Ports 064h and 060h for the fast keyboard A20M# and CPU reset control sequences. If a write to I/O Port 060h[1] = 1 after a write takes place to I/O Port 064h with data of D1h, then the Core Logic module asserts the A20M# signal. A20M# remains asserted until cleared by any one of the following:

- A write to bit 1 of I/O Port 092h.
- A CPU reset of some kind.
- A write to I/O Port 060h[1] = 0 following a write to I/O Port 064h with data of D1h.

The fast keyboard A20M# and CPU reset can be disabled through F0 Index 52h[7]. By default, bit 7 is set, and the fast keyboard A20M# and CPU reset monitor logic is active. If bit 7 is clear, the Core Logic module forwards the commands to the keyboard controller.

By default, the Core Logic module forces the deassertion of A20M# during a warm reset. This action may be disabled if F0 Index 52h[4] is cleared.

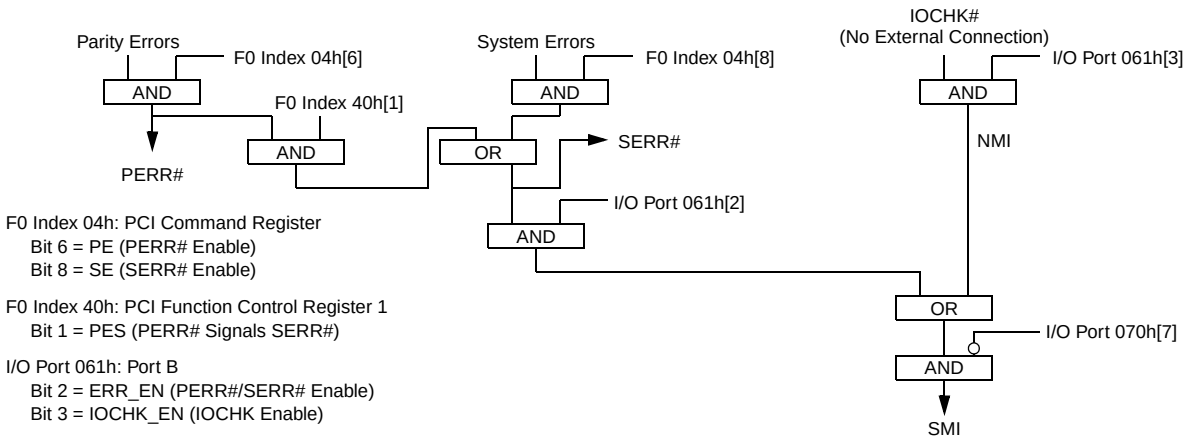


Figure 5-10. SMI Generation for NMI

Core Logic Module (Continued)

5.2.9 Power Management Logic

The Core Logic module integrates advanced power management features including idle timers for common system peripherals, address trap registers for programmable address ranges for I/O or memory accesses, four programmable general purpose external inputs, clock throttling with automatic speedup for the GX1 clock, software GX1 stop clock, 0V Suspend/Resume with peripheral shadow registers, and a dedicated serial bus to/from the GX1 module providing power management status.

The Core Logic module is ACPI (Advanced Configuration Power Interface) compliant. An ACPI-compliant system is one whose underlying BIOS, device drivers, chipset and peripherals conform to revision 1.0 of the ACPI specification. The Core Logic also supports Advanced Power Management (APM).

The SC1100 provides the following support of ACPI states:

- CPU States: C0, C1, and C3.
- Sleep States:
  - SL1/SL2 - ACPI S1 equivalent
  - SL3 - ACPI S3 equivalent
  - SL4 - ACPI S4 equivalent
  - SL5 - ACPI S5 equivalent
- General Purpose Events: Fully programmable GPE0 Event Block registers.
- Wakeup Events: Supported through GPWIO[2:0] which are powered by standby voltage and generate SMIs. See registers at F1BAR1+I/O Offset 0Ah and F1BAR1+I/O Offset 12h. Also see Section 4.6 "System Wakeup Control (SWC)" on page 92 and Table 5-5 "Wakeup Events Capability" on page 135.

SC1100 device power management is highly tuned for low power systems. It allows the system designer to implement a wide range of power saving modes using a wide range of capabilities and configuration options.

SC1100 controls the following functions directly:

- The system clocks.
- Core processor power states.
- Wakeup/resume event detection, including general purpose events.
- Power supply and power planes.

It also supports systems with an external micro controller that is used as a power management controller.

5.2.9.1 CPU States

The SC1100 supports three CPU states: C0, C1 and C3 (the Core Logic C2 CPU state is not supported). These states are fully compliant with the ACPI specification, revision 1.0. These states occur in the Working state only (S0/G0). They have no meaning when the system transitions into a Sleep state. For details on the various Sleep states, see Section 5.2.9.2 "Sleep States" on page 134.

C0 Power State - On

In this state the GX1 module executes code. This state has two substates: Full Speed or Throttling; selected via the THT_EN bit (F1BAR1+I/O Offset 00h[4]).

C1 Power State - Active Idle

The SC1100 enters the C1 state, when the Halt Instruction (HLT) is executed. It exits this state back to the C0 state upon an NMI, an unmasked interrupt, or an SMI. The Halt instruction stops program execution and generates a special Halt bus cycle. (See “Usage Hints” on page 137.)

Bus masters are supported in the C1 state and the SC1100 will temporarily exit C1 to perform a bus master transaction.

C2 Power State

The SC1100 does not support the C2 power state. All relevant registers and bit fields in the Core Logic are reserved.

C3 Power State

The SC1100 enters the C3 state, when the P_LVL3 register (F1BAR1+I/O Offset 05h) is read. It exits this state back to the C0 state (Full Speed or Throttling, depending on the THT_EN bit) upon:

- An NMI, an unmasked interrupt, or an SMI.
- A bus master request, if enabled via the BM_RLD bit (F1BAR1+I/O Offset 0Ch[1]).

In this state, the GX1 module is in Suspend Refresh mode (for details, see the Power Management section of the *GX1 Processor Series Datasheet*, and Section 5.2.9.5 "Usage Hints" on page 137).

PCI arbitration should be disabled prior entering the C3 state via the ARB_DIS bit in the PM2_CNT register (F1BAR1+Offset 20h[0]) because a PCI arbitration event could start after P_LVL3 has been read. After wakeup ARB_DIS needs to be cleared.

5.2.9.2 Sleep States

The SC1100 supports four Sleep states (SL1-SL4) and the Soft Off state (G2/S5). These states are fully compliant with the ACPI specification, revision 1.0.

When the SLP_EN bit (FABAR1+IO Offset 0Ch[13]) is set to 1, the SC1100 enters an SLx state according to the SLP_TYP field (F1BAR1+I/O Offset 0Ch[12:10]). It exits the Sleep state back to the S0 state (C0 state - Full Speed or Throttling, depending on the THT_EN bit) upon an enabled power management event. Table 5-5 on page 135 lists wakeup events from the various Sleep states.

Core Logic Module (Continued)

SL1 Sleep State (ACPI S1)

In this state the core processor is in 3V Suspend mode (all its clocks are stopped, including the memory controller). The SDRAM is placed in self-refresh mode. All other SC1100 and system clocks are running. The frequency multipliers (FMUL) and the PLLs are all running. All devices are powered up (PWRCNT[2:1] and ONCTL# are all asserted). See Section 5.2.9.5 "Usage Hints" on page 137.

No reset is performed, when exiting this state. The SC1100 keeps all context in this state. This state corresponds to ACPI Sleep state S1.

SL2 Sleep State (ACPI S1)

In this state, all of the SC1100 clocks are stopped including the FMULs and the PLLs, but not the 32 KHz oscillator. Selected clocks from the FMULs and PLLs can be stopped. The SDRAM is placed in self-refresh mode. The PWRCNT1 pin is deasserted. The SC1100 itself is powered up. The system designer can decide which other system devices to power off with the PWRCNT1 pin.

No reset is performed, when exiting this state. The SC1100 keeps all context in this state. This state corresponds to ACPI sleep state S1, with lower power and longer wake time than in SL1.

SL3 Sleep State (ACPI S3)

In this state, the SDRAM is placed in self-refresh mode, and PWRCNT[2:1] are deasserted. PWRCNT[2:1] should be used to power off most of the system (except for the SDRAM). If the Save-to-RAM feature is used, external circuitry in the SDRAM interface is required to guarantee data integrity. All SC1100 signals powered by V_{SB}, V_{SBL} or V_{BAT} are still functional to allow wakeup and to maintain the real-time clock.

The power-up sequence is performed, when exiting this state. This state corresponds to ACPI sleep state SL3.

SL4 and SL5 Sleep States (ACPI S4 and S5)

The SL4 and SL5 states are similar from the hardware perspective. In these states, the SC1100 deasserts PWRCNT[2:1] and ONCTL#. PWRCNT[2:1] and ONCTL# should be used to power off the system. All signals powered by V_{SB}, V_{SBL} or V_{BAT} are still functional to allow wakeup and to keep the real-time clock.

The power-up sequence is performed when exiting this state. This state corresponds to ACPI Sleep states SL4 and SL5.

Table 5-5. Wakeup Events Capability

Event	S0/C1	S0/C3	SL1	SL2	SL3	SL4, SL5
Enabled Interrupts	Yes	Yes	Yes	-	-	-
SMI according to Table 5-8	Yes	Yes	Yes	-	-	-
SCI according to Table 5-8	Yes	Yes	Yes	-	-	-
GPIO[47:32], GPIO[15:0]	Yes	Yes	Yes	-	-	-
Power Button	Yes	Yes	Yes	Yes	Yes	Yes
Power Button Override	Yes	Yes	Yes	Yes	Yes	Yes
Bus Master Request	Yes[1]	Yes	Yes	-	-	-
Thermal Monitoring	Yes	Yes	Yes	Yes	Yes	Yes
USB	Yes	Yes	Yes	Yes	-	-
IRRX1 (Infrared)	Yes	Yes	Yes	Yes	-	-
GPWIO[2:0]	Yes	Yes	Yes	Yes	Yes	Yes
RI# (UART)	Yes	Yes	Yes	Yes	-	-
RTC	Yes	Yes	Yes	Yes	Yes	Yes

1. Temporarily exits state.

Core Logic Module (Continued)

5.2.9.3 Power Planes Control

The SC1100 supports up to three power planes. Three signals are used to control these power planes. Table 5-6 describes the signals and when each is asserted.

Table 5-6. Power Planes Control Signals vs. Sleep States

Signal	S0	SL1	SL2	SL3	SL4 and SL5
PWRCNT1	1	1	0	0	0
PWRCNT2	1	1	1	0	0
ONCTL#	0	0	0	0	1

These signals allow control of the power of system devices and the SC1100 itself. Table 5-7 describes the SC1100 power planes with respect to the different Sleep and Global states.

Table 5-7. Power Planes vs. Sleep/Global States

Sleep/ Global State	V_{CORE}, $V_{I/O}$, V_{PLL}	V_{SB}, V_{SBL}	V_{BAT}
S0, SL1 and SL2	On	On	On or Off
SL3, SL4 and SL5	Off	On	On or Off
G3	Off	Off	On
No Power	Off	Off	Off
Illegal	On	Off	On or Off

The SC1100 power planes are controlled externally by the three signals (i.e., the system designer should make sure the system design is such that Table 5-7 is met) for all supported Sleep states.

V_{SB} and V_{BAT} are not controlled by any control signal. V_{SB} exists as long as the AC power is plugged in (for desktop systems) or the main battery is charged (for mobile systems). V_{BAT} exists as long as the RTC battery is charged.

The case in which V_{SB} does not exist is called Mechanical Off (G3).

5.2.9.4 Power Management Events

The SC1100 supports power management events that can manage:

- Transition of the system from a Sleep state to a Work state. This is done by the hardware. These events are defined as wakeup events.
- Enabled wakeup events to set the WAK_STS bit (F1BAR1+I/O Offset 08h[15]) to 1, when transitioning the system back to the working state.
- Generation of an interrupt. This invokes the relevant software driver. The interrupt can either be an SMI or SCI (selected by the SCI_EN bit, F1BAR1+I/O Offset 0Ch[0]). These events are defined as interrupt events.

Table 5-8 lists the power management events that can generate an SCI or SMI.

Table 5-8. Power Management Events

Event	SCI	SMI
Power Button	Yes	Yes
Power Button Override	Yes	-
Bus Master Request	Yes	-
Thermal Monitoring	Yes	Yes
USB	Yes	Yes
RTC	Yes	Yes
ACPI Timer	Yes	Yes
GPIO	Yes	Yes
IRRX1	Yes	Yes
RI2#	Yes	Yes
GPWIO	Yes	Yes
Internal SMI signal	Yes	-

Core Logic Module (Continued)

Power Button

The power button (PWRBTN#, ball AF15) input provides two events: a wake request, and a sleep request. For both these events, the PWRBTN# signal is debounced (i.e., the signal state is transferred only after 14 to 16 msec without transitions, to ensure that the signal is no longer bouncing).

ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 196). GPIO63 must be pulsed low for at least 16 msec and not more than 4 sec. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low.

Power Button Wake Event - Detection of a high-to-low transition on the debounced PWRBTN# input signal when in SL1 to SL5 Sleep states. The system is considered in the Sleep state, only after it actually transitioned into the state and not only according to the SLP_TYP field.

In reaction to this event, the PWRBTN_STS bit (F1BAR1+I/O Offset 08h[8]) is set to 1 and a wakeup event or an interrupt is generated (note that this is regardless of the PWRBTN_EN bit, F1BAR1+I/O Offset 0Ah[8]).

Power Button Sleep Event - Detection of a high-to-low transition on the debounced PWRBTN# input signal, when in the Working state (S0).

In reaction to this event, the PWRBTN_STS bit is set to 1.

- When both the PWRBTN_STS bit and the PWRBTN_EN bit are set to 1, an SCI interrupt is generated.
- When SCI_EN bit is 0, ONCTL# and PWRCNT[2:1] are deasserted immediately regardless of the PWRBTN_EN bit.

Power Button Override

When PWRBTN# is 0 for more than four seconds, ONCTL# and PWRCNT[2:1] are deasserted (i.e., the system transitions to the SL5 state, "Soft Off"). This power management event is called the power button override event.

In reaction to this event, the PWRBTN_STS bit is cleared to 0 and the PWRBTNOR_STS bit (F1BAR1+I/O Offset 08h[11]) is set to 1.

Thermal Monitoring

The thermal monitoring event (THRM#, ball AE15) enables control of ACPI-OS Control.

When the THRM# signal transitions from high-to-low, the THRM_STS bit (F1BAR1+I/O Offset 10h[5]) is set to 1. If the THRM_EN bit (F1BAR1+I/O Offset 12h[5]) is also set to 1, an interrupt is generated.

IRRX1, RI#

See Section 4.4.1 "SIO Control and Configuration Registers" on page 73 for control and operation.

5.2.9.5 Usage Hints

- During initialization, the BIOS should:
  - — Clear the SUSP_HLT bit in CCR2 (GX1 module, Index C2h[3]) to 0. This is needed for compliance with C0 definition of ACPI, when the Halt instruction (HLT) is executed.
  - — Disable the SUSP_3V option in C3 power state (F0 Index 60h[2]).
  - — Disable the SUSP_3V option in SL1 sleep state (F0 Index 60h[1]).
- SMM code should clear the CLK_STP bit in the PM Clock Stop Control register (GX_BASE+Memory Offset 8500h[0]) to 0 when entering C3 state.
- SMM code should correctly set the CLK_STP bit in the PM Clock Stop Control register (GX_BASE+Memory Offset 8500h[0]) when entering the SL1, SL2, and SL3 states.

5.2.10 Power Management Programming

The power management resources provided by a combined GX1 module and Core Logic module based system supports a high efficiency power management implementation. The following explanations pertain to a full-featured “notebook” power management system. The extent to which these resources are employed depends on the application and on the discretion of the system designer.

Power management resources can be grouped according to the function they enable or support. The major functions are as follows:

- APM Support
- CPU Power Management
  - — Suspend Modulation
  - — 3V Suspend
  - — Save-to-Disk
- Peripheral Power Management
  - — Device Idle Timers and Traps
  - — General Purpose Timers
  - — ACPI Timer Register
  - — Power Management SMI Status Reporting Registers

Included in the following subsections are details regarding the registers used for configuring power management features. The majority of these registers are directly accessed through the PCI configuration register space designated as Function 0 (F0). However, included in the discussions are references to F1BARx+I/O Offset xxh. This refers to registers accessed through base address registers in Function 1 (F1) at Index 10h (F1BAR0) and Index 40h (F1BAR1).

Core Logic Module (Continued)

5.2.10.1 APM Support

Many notebook computers rely solely on an Advanced Power Management (APM) driver for enabling the operating system to power-manage the CPU. APM provides several services which enhance the system power management; but in its current form, APM is imperfect for the following reasons:

- 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.

The Core Logic module provides two entry points for APM support:

- Software CPU Suspend control via the CPU Suspend Command register (F0 Index AEh).
- Software SMI entry via the Software SMI register (F0 Index D0h). This allows the APM BIOS to be part of the SMI handler.

5.2.10.2 CPU Power Management

The three greatest power consumers in a system are the display, the hard drive, and the CPU. The power management of the first two is relatively straightforward and is discussed in Section 5.2.10.3 "Peripheral Power Management" on page 139.

APM, if available, is used primarily by CPU power management since the operating system is most capable of reporting the Idle condition. Additional resources provided by the Core Logic module supplement APM by monitoring external activity and power managing the CPU based on the system demands. The two processes for power managing the CPU are Suspend Modulation and 3V Suspend.

Suspend Modulation

Suspend Modulation works by asserting and deasserting the internal SUSP# signal to the GX1 module for configurable durations. When SUSP# is asserted to the GX1 module, it enters an Idle state during which time the power consumption is significantly reduced. Even though the PCI clock is still running, the GX1 module stops the clocks to its core when SUSP# is asserted. By modulating SUSP# a reduced frequency of operation is achieved.

The Suspend Modulation feature works by assuming that the GX1 module is Idle unless external activity indicates otherwise. This approach effectively slows down the GX1 module until external activity indicates a need to run at full speed, thereby reducing power consumption. This approach is the opposite of that taken by most power management schemes in the industry, which run the system at full speed until a period of inactivity is detected, and then slows down. Suspend Modulation, the more aggressive approach, yields lower power consumption.

Suspend Modulation serves as the primary CPU power management mechanism when APM is not present. It also acts as a backup for situations where APM does not correctly detect an Idle condition in the system.

To provide high-speed performance when needed, SUSP# modulation is temporarily disabled any time system activity is detected. When this happens, the GX1 module is “instantly” converted to full speed for a programmed duration. System activities in the Core Logic module are asserted as: any unmasked IRQ, accessing Port 061h, any asserted SMI.

The automatic speedup events (IRQ) for Suspend Modulation should be used together with software-controlled speedup registers for major I/O events such as any access to the FDC, HDD, or parallel/serial ports, since these are indications of major system activities. When major I/O events occur, Suspend Modulation should be temporarily disabled using the procedures described in the Power Management registers in the following subsections.

If a bus master (UltraDMA/33, Audio, USB) request occurs, the GX1 module automatically deasserts SUSPA# and grants the bus to the requesting bus master. When the bus master deasserts REQ#, SUSPA# reasserts. This does not directly affect the Suspend Modulation programming.

Configuring Suspend Modulation: Control of the Suspend Modulation feature is accomplished using the Suspend Modulation and Suspend Configuration registers (F0 Index 94h and 96h, respectively).

The Power Management Enable Register 1 at F0 Index 80h on page 177 contains the Global Power Management bit (Bit 0). The global power management bit must be enabled for Suspend Modulation and all other power management resources to function.

Bit 0 of the Suspend Configuration register enables the Suspend Modulation feature. Bit 1 controls how SMI events affect the Suspend Modulation feature. In general this bit should be set to 1, which causes SMIs to disable Suspend Modulation until it is re-enabled by the SMI handler.

The Suspend Modulation register controls two 8-bit counters that represent the number of 32 µs intervals that the internal SUSP# signal is asserted and then deasserted to the GX1 module. These counters define a ratio which is the effective frequency of operation of the system while Suspend Modulation is enabled.

$$F_{eff} = F_{GX1} \times \frac{\text{Asserted Count}}{\text{Asserted Count + Deasserted Count}}$$

The IRQ Speedup Timer Count register (F0 Index 8Ch) configures the amount of time which Suspend Modulation is disabled when the event occurs.

Core Logic Module (Continued)

SMI Speedup Disable: If the Suspend Modulation feature is being used for CPU power management, the occurrence of an SMI disables Suspend Modulation so that the system operates at full speed while in SMM. There are two methods used to invoke this via bit 1 of the Suspend Configuration register.

1) If F0 Index 96h[1] = 0: Use the IRQ Speedup Timer (F0 Index 8Ch) to temporarily disable Suspend Modulation when an SMI occurs.

2) If F0 Index 96h[1] = 1: Disable Suspend Modulation when an SMI occurs until a read to the SMI Speedup Disable register (F1BAR0+I/O Offset 08h).

The SMI Speedup Disable register prevents VSA software from entering Suspend Modulation while operating in SMM. The data read from this register can be ignored. If the Suspend Modulation feature is disabled, reading this I/O location has no effect.

3 Volt Suspend

The Core Logic module supports the stopping of the CPU and system clocks for a 3V Suspend state. If appropriately configured, via the Clock Stop Control register (F0 Index BCh), the Core Logic module asserts internal SUSP_3V after it has gone through the SUSP#/SUSPA# handshake. SUSP_3V is a state indicator, indicating that the system is in a low-activity state and Suspend Modulation is active. This indicator can be used to put the system into a low-power state (the system clock can be turned off).

Internal SUSP_3V is connected to the enable control of the clock generators, so that the clocks to the CPU and the Core Logic module (and most other system devices) are stopped. The Core Logic module continues to decrement all of its device timers and respond to external SMI interrupts after the input clock has been stopped, as long as the 32 KHz clock continues to oscillate. Any SMI event or unmasked interrupt causes the Core Logic module to deassert SUSP_3V, restarting the system clocks. As the CPU or other device might include a PLL, the Core Logic module holds SUSP# active for a pre-programmed period of delay (the PLL re-sync delay) that varies from 0 to 15 ms. After this period has expired, the Core Logic module deasserts SUSP#, stopping Suspend. SMI# is held active for the entire period, so that the CPU reenters SMM when the clocks are restarted.

Save-to-Disk

Save-to-Disk is supported by the Core Logic module. In this state, the power is typically removed from the Core Logic module and from the entire SC1100, causing the state of the legacy peripheral devices to be lost. Shadow registers are provided for devices which allow their state to be saved prior to removing power. This is necessary because the legacy AT peripheral devices used several write only registers. To restore the exact state of these devices on resume, the write only register values are “shadowed” so that the values can be saved by the power management software.

The PC/AT compatible keyboard controller (KBC) and floppy port (FDC) do not exist in the SC1100. However, it is possible that one is attached on the ISA bus or the LPC bus (e.g., in a SuperI/O device). Some FDC registers are shadowed because they cannot be safely read. Additional shadow registers for other functions are described in Table 5-29 "F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support" on page 166.

5.2.10.3 Peripheral Power Management

The Core Logic module provides peripheral power management using a combination of device idle timers, address traps, and general purpose I/O pins. Idle timers are used in conjunction with traps to support powering down peripheral devices.

Device Idle Timers and Traps

Idle timers are used to power manage a peripheral by determining when the peripheral has been inactive for a specified period of time, and removing power from the peripheral at the end of that time period.

Idle timers are provided for the commonly-used peripherals (FDC, IDE, Parallel/Serial Ports, and Mouse/Keyboard). In addition, there are three user-defined timers that can be configured for either I/O or memory ranges.

The idle timers are 16-bit countdown timers with a one second time base, providing a timeout range of 1 to 65536 seconds (1092 minutes) (18 hours).

When the idle timer count registers are loaded with a non-zero value and enabled, the timers decrement until one of two possibilities happens: a bus cycle occurs at that I/O or memory range, or the timer decrements to zero.

If a bus cycle occurs, the timer is reloaded and begins decrementing again. If the timer decrements to zero, and power management is enabled (F0 Index 80h[0] = 1), the timer generates an SMI.

When an idle timer generates an SMI, the SMI handler manages the peripheral power, disables the timer, and enables the trap. The next time an event occurs, the trap generates an SMI. This time, the SMI handler applies power to the peripheral, resets the timer, and disables the trap.

Relevant registers for controlling Device Idle Timers are: F0 Index 80h, 81h, 82h, 93h, 98h-9Eh, and ACh.

Relevant registers for controlling User Defined Device Idle Timers are: F0 Index 81h, 82h, A0h, A2, A4h, C0h, C4h, C8h, CCh, CDh, and CEh.

Core Logic Module (Continued)

General Purpose Timers

The Core Logic module contains two general purpose idle timers, General Purpose Timer 1 (F0 Index 88h) and General Purpose Timer 2 (F0 Index 8Ah). These two timers are similar to the Device Idle Timers in that they count down to zero unless re-triggered, and generate an SMI when they reach zero. However, these are 8-bit timers instead of 16 bits, they have a programmable timebase, and the events which reload these timers are configurable. These timers are typically used for an indication of system inactivity for Suspend determination.

General Purpose Timer 1 can be re-triggered by activity to any of the configured User Defined Devices, Keyboard and Mouse, Parallel and Serial, Floppy disk, or Hard disk.

General Purpose Timer 2 can be re-triggered by a transition on the GPIO7 signal (if GPIO7 is properly configured).

When a General Purpose Timer is enabled or when an event reloads the timer, the timer is loaded with the configured count value. Upon expiration of the timer an SMI is generated and a status flag is set. Once expired, this counter must be re-initialized by disabling and enabling it.

The timebase for both General Purpose Timers can be configured as either 1 second (default) or 1 millisecond. The registers at F0 Index 89h and 8Bh are the control registers for the General Purpose Timers.

ACPI Timer Register

The ACPI Timer register (F1BAR0+I/O Offset 1Ch or at F1BAR1+I/O Offset 1Ch) provides the ACPI counter. The counter counts at 14.31818/4 MHz (3.579545 MHz). If SMI generation is enabled (F0 Index 83h[5] = 1), an SMI or SCI is generated when bit 23 of the ACPI Timer Register toggles.

Power Management SMI Status Reporting Registers

The Core Logic module updates status registers to reflect the SMI sources. Power management SMI sources are the device idle timers, address traps, and general purpose I/O pins.

Power management events are reported to the GX1 module through the active low SMI# signal. When an SMI is initiated, the SMI# signal is asserted low and is held low until all SMI sources are cleared. At that time, SMI# is deasserted.

All SMI sources report to the Top Level SMI Status register (F1BAR0+I/O Offset 02h) and the Top Level SMI Status Mirror register (F1BAR0+I/O Offset 00h). The Top SMI Status and Status Mirror registers are the top level of hierarchy for the SMI Handler in determining the source of an SMI. These two registers are identical except that reading the register at F1BAR0+I/O Offset 02h clears the status.

Since all SMI sources report to the Top Level SMI Status register, many of its bits combine a large number of events requiring a second level of SMI status reporting. The second level of SMI status reporting is set up very much like the top level. There are two status reporting registers, one “read only” (mirror) and one “read to clear”. The data returned by reading either offset is the same, the difference between the two being that the SMI can not be cleared by reading the mirror register.

Figure 5-11 shows an example SMI tree for checking and clearing the source of General Purpose Timers and the User Defined Trap generated SMI.

Core Logic Module (Continued)

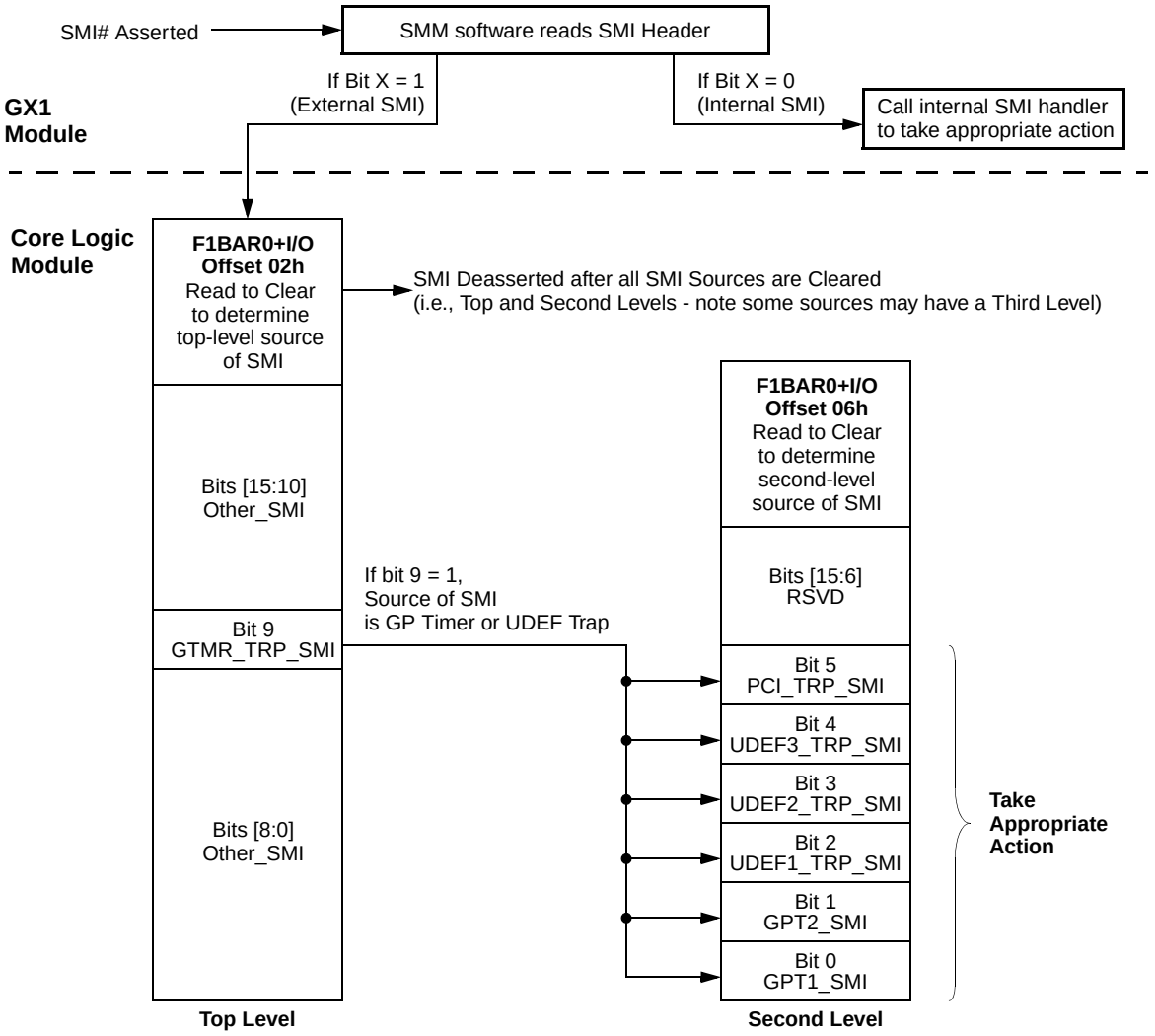


Figure 5-11. General Purpose Timer and UDEF Trap SMI Tree Example

Core Logic Module (Continued)

5.2.10.4 Power Management Programming Summary

Table 5-9 provides a programming register summary for the power management timers, traps, and functions. For complete bit information regarding the registers listed in Table 5-9, refer to Section 5.4.1 "Bridge, GPIO, and LPC Registers - Function 0" on page 166.

Table 5-9. Device Power Management Programming Summary

Device Power Management Resource	Located at F0 Index xxh Unless Otherwise Noted			
	Enable	Configuration	Second Level SMI Status/No Clear	Second Level SMI Status/With Clear
Traps	80h[2]	N/A	N/A	N/A
Idle Timers	80h[1]	N/A	N/A	N/A
Power Management	80h[0]	N/A	N/A	N/A
Keyboard / Mouse Idle Timer	81h[3]	93h[1:0]	85h[3]	F5h[3]
Parallel / Serial Idle Timer	81h[2]	93h[1:0]	85h[2]	F5h[2]
Floppy Disk Idle Timer	81h[1]	9Ah[15:0], 93h[7]	85h[1]	F5h[1]
Primary Hard Disk Idle Timer	81h[0]	98h[15:0], 93h[5]	85h[0]	F5h[0]
Secondary Hard Disk Idle Timer	83h[7]	ACh[15:0], 93h[4]	86h[4]	F6h[4]
User Defined Device 1 Idle Timer	81h[4]	A0h[15:0], C0h[31:0], CCh[7:0]	85h[4]	F5h[4]
User Defined Device 2 Idle Timer	81h[5]	A2h[15:0], C4h[31:0], CDh[7:0]	85h[5]	F5h[5]
User Defined Device 3 Idle Timer	81h[6]	A4h[15:0], C8h[31:0], CEh[7:0]	85h[6]	F5h[6]
Global Trap Enable	80h[2]	N/A	N/A	N/A
Keyboard / Mouse Trap	82h[3]	9Eh[15:0] 93h[1:0]	86h[3]	F6h[3]
Parallel / Serial Trap	82h[2]	9Ch[15:0], 93h[1:0]	86h[2]	F6h[2]
Floppy Disk Trap	82h[1]	93h[7]	86h[1]	F6h[1]
Primary Hard Disk Trap	82h[0]	93h[5]	86h[0]	F6h[0]
Secondary Hard Disk Trap	83h[6]	93h[4]	86h[5]	F6h[5]
User Defined Device 1 Trap	82h[4]	C0h[31:0], CCh[7:0]	F1BAR0+I/O Offset 04h[2]	F1BAR0+I/O Offset 06h[2]
User Defined Device 2 Trap	82h[5]	C4h[31:0], CDh[7:0]	F1BAR0+I/O Offset 04h[3]	F1BAR0+I/O Offset 06h[3]
User Defined Device 3 Trap	82h[6]	C8h[31:0], CEh[7:0]	F1BAR0+I/O Offset 04h[4]	F1BAR0+I/O Offset 06h[4]
General Purpose Timer 1	83h[0]	88h[7:0], 89h[7:0], 8Bh[4]	F1BAR0+I/O Offset 04h[0]	F1BAR0+I/O Offset 06h[0]
General Purpose Timer 2	83h[1]	8Ah[7:0], 8Bh[5,3,2]	F1BAR0+I/O Offset 04h[1]	F1BAR0+I/O Offset 06h[1]
Suspend Modulation	96h[0]	94h[15:0], 96h[2:0]	N/A	N/A
IRQ Speedup	80h[3]	8Ch[7:0]	N/A	N/A

Core Logic Module (Continued)

5.2.11 GPIO Interface

Up to 64 GPIOs in the Core Logic module are provided for system control. For further information, see Section 3.2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50 and Table 5-30 "F0BAR0+I/O Offset: GPIO Configuration Registers" on page 196.

Note: Not all GPIOs are available on SC1100 balls. GPIOs [63:48], [46:42], [31:21], and [5:4] are reserved.

5.2.12 Integrated Audio

The Core Logic module provides hardware support for the Virtual (soft) Audio subsystem as part of the Virtual System Architecture® (VSA™) technology for capture and playback of audio using an external codec. This eliminates much of the hardware traditionally associated with audio functions.

This hardware support includes:

- XpressAUDIO with 16-bit stereo FM synthesis and OPL3 emulation.
- Six-channel buffered PCI bus mastering interface.
- AC97 version 2.0 compatible interface to the codec. Any codec, which supports an independent input and output sample rate conversion interface, can be used with the Core Logic module.

Additional hardware provides the necessary functionality for VSA. This hardware includes the ability to:

- Generate an SMI to alert software to update required data. An SMI is generated when either audio buffer is half empty or full. If the buffers become completely empty or full, the Empty bit is asserted.
- Generate an SMI on I/O traps.
- Trap accesses for sound card compatibility at either I/O Port 220h-22Fh, 240h-24Fh, 260h-26Fh, or 280h-28Fh.
- Trap accesses for FM compatibility at I/O Port 388h-38Bh.
- Trap accesses for MIDI UART interface at I/O Port 300h-301h or 330h-331h.
- Trap accesses for serial input and output at COM2 (I/O Port 2F8h-2FFh) or COM4 (I/O Port 2E8h-2EFh).
- Support trapping for low (I/O Port 00h-0Fh) and/or high (I/O Port C0h-DFh) DMA accesses.
- Support hardware status register reads in Core Logic module, minimizing SMI overhead.
- Support is provided for software-generated IRQs on IRQ 2, 3, 5, 7, 10, 11, 12, 13, 14, and 15.

The following subsections include details of the registers used for configuring the audio interface. The registers are accessed through F3 Index 10h, the Base Address Register (F3BAR0) in Function 3. F3BAR0 sets the base address for the XpressAUDIO support registers as shown in Table 5-37 "F3: PCI Header Registers for XpressAUDIO Audio Configuration" on page 231.

5.2.12.1 Data Transport Hardware

The data transport hardware can be broadly divided into two sections: bus mastering and the codec interface.

Audio Bus Masters

The Core Logic module audio hardware includes six PCI bus masters (three for input and three for output) for transferring digitized audio between memory and the external codec. With these bus master engines, the Core Logic module off-loads the CPU and improves system performance.

The programming interface defines 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.

The six bus masters that directly drive specific slots on the AC97 interface are described in Table 5-10.

Table 5-10. Bus Masters That Drive Specific Slots of the AC97 Interface

Audio Bus Master #	Slots	Description
0	3 and 4	32-Bit output to codec. Left and right channels.
1	3 and 4	32-Bit input from codec. Left and right channels.
2	5	16-Bit output to codec.
3	5	16-Bit input from codec.
4	6 or 11	16-Bit output to codec. Slot in use is determined by F3BAR0+Memory Offset 08h[19].
5	6 or 11	16-Bit input from codec. Slot in use is determined by F3BAR0+Memory Offset 08h[20].

Core Logic Module (Continued)

Physical Region Descriptor Table Address

Before the bus master starts a master transfer it must be programmed with a pointer (PRD Table Address register) 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 descriptor table entries must be aligned on a 32-byte boundary and the table cannot cross a 64 KB boundary in memory.

Physical Region Descriptor Format

Each physical memory region to be transferred is described by a Physical Region Descriptor (PRD) as illustrated in Table 5-11. 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. The second DWORD contains the size (16 bits) of the buffer and flags (EOT, EOP, JMP). The description of the flags are as follows:

- EOT bit - If set in a PRD, this bit indicates the last entry in the PRD table (bit 31). 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.
- EOP bit - If set in a PRD and the bus master has completed the PRD's transfer, the End of Page bit is set (Status register bit 0 = 1) and an SMI 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 (Status register bit 1 = 1) and the bus master pauses. In this paused condition, reading the Status register clears both the Bus Master Error and the End of Page bits and the bus master continues.
- JMP bit - This PRD is special. If set, the Memory Region Physical Base Address is now the target address of the JMP. The target must be on a 32-byte boundary so bits[4:0] must be written to 0. There is no 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.

Programming Model

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 listed below, the reference to "Example" refers to Figure 5-12 "PRD Table Example" on page 145.

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 32-byte boundary. The last PRD 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.

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 SMI when a PRD is empty.

   Example - Fill Audio Buffer_1 and Audio Buffer_2. The SMI generated by the EOP from the first PRD allows the software to refill Audio Buffer_1. The second SMI refills Audio Buffer_2. The third SMI refills Audio Buffer_1 and so on.

Table 5-11. Physical Region Descriptor Format

	Byte 3								Byte 2								Byte 1								Byte 0							
DWORD	**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 [31:1] (Audio Data Buffer)																															0
1	EOT	EOP	JMP	Reserved													Size [15:1]															0

Core Logic Module (Continued)

4) Read the SMI Status register to clear the Bus Master Error and End of Page bits (bits 1 and 0).

   Set the correct direction to the Read or Write Control bit (Command register bit 3). Note that the direction of the data transfer of a particular bus master is fixed and therefore the direction bit must be programmed accordingly. It is assumed that the codec has been properly programmed to receive the audio data.

   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.

   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 to/from memory responding to bus master requests from the AC97 interface. At the completion of each PRD, the bus master’s next response depends on the settings of the flags in the PRD.

   Example - At the completion of PRD_1 an SMI is generated because the EOP bit is set while the bus master continues on to PRD_2. The address in the PRD Table Address register is incremented by 08h and is now pointing to PRD_3. The SMI Status register is read to clear the End of Page status flag. Since Audio Buffer_1 is now empty, the software can refill it.

   At the completion of PRD_2 an SMI is generated because the EOP bit is set. The bus master then continues on to PRD_3. The address in the PRD Table Address register is incremented by 08h. The DMA SMI Status register is read to clear the End of Page status flag. Since Audio Buffer_2 is now empty, the software can refill it. Audio Buffer_1 has been refilled from the previous SMI.

   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.

Stopping the bus master can be accomplished by not reading the SMI Status register End of Page status flag. This leads to a second EOP which causes a Bus Master Error and pauses the bus master. In effect, once a bus master has been enabled it never has to be disabled, just paused. The bus master cannot be disabled unless the bus master has been paused or has reached an EOT.

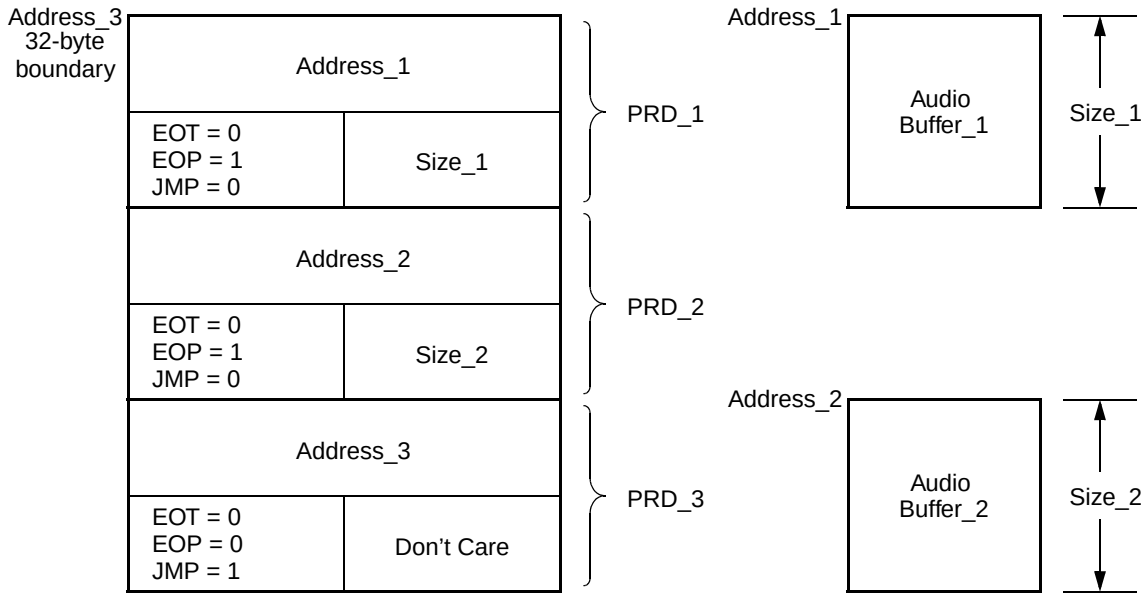


Figure 5-12. PRD Table Example

Core Logic Module (Continued)

5.2.12.2 AC97 Codec Interface

The AC97 codec (e.g., LM4548) is the master of the serial interface and generates the clocks to Core Logic module. Figure 5-13 shows the signal connections between a codec and the SC1100:

For PC speaker synthesis, the Core Logic module outputs the PC speaker signal on the PC_BEEP ball which is connected to the PC_BEEP input of the AC97 codec. Note that PC_BEEP is muxed on ball AE18 with GPIO16+IRRX1 and must be programmed via PMR[0] (see Table 3-2 on page 50).

Codec Configuration/Control Registers

The codec 32-bit related registers:

- Codec GPIO Status Register (F3BAR0+Memory Offset 00h)
- Codec GPIO Control Register (F3BAR0+Memory Offset 04h)
- Codec Status Register (F3BAR0+Memory Offset 08h)
- Codec Command Register (F3BAR0+Memory Offset 0Ch)

Codec GPIO Status and Control Registers:

The Codec GPIO Status and Control registers are used for codec GPIO related tasks such as enabling a codec GPIO interrupt to cause an SMI.

Codec Status Register:

The Codec Status register stores the codec status WORD. It is updated every valid Status Word slot.

Codec Command Register:

The Codec Command register writes the control WORD to the codec. By writing the appropriate control WORDs to this port, the features of the codec can be controlled. The contents of this register are written to the codec during the Control Word slot.

The bit formats for these registers are given in Table 5-38 "F3BAR0+Memory Offset: XpressAUDIO Configuration Registers" on page 232.

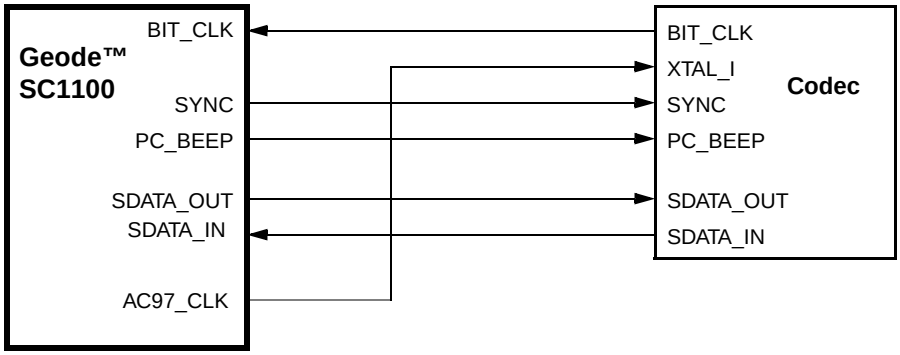


Figure 5-13. AC97 V2.0 Codec Signal Connections

Core Logic Module (Continued)

5.2.12.3 VSA Technology Support Hardware

The Core Logic module incorporates the required hardware in order to support the Virtual System Architecture® (VSA™) technology for capture and playback of audio using an external codec. This eliminates much of the hardware traditionally associated with industry standard audio functions.

XpressAUDIO software provides 16-bit compatible sound. This software is available to OEMs for incorporation into the system BIOS ROM.

VSA Technology

VSA technology provides a framework to enable software implementation of traditionally hardware-only components. VSA software executes in System Management Mode (SMM), enabling it to execute transparently to the operating system, drivers, and applications.

The VSA design is based upon a simple model for replacing hardware components with software. Hardware to be virtualized is merely replaced with simple access detection circuitry which asserts the SMI# (System Management Interrupt) internal signal when hardware accesses are detected. The current execution stream is immediately preempted, and the processor enters SMM. The SMM system software then saves the processor state, initializes the VSA execution environment, decodes the SMI source, and dispatches handler routines which have registered requests to service the decoded SMI source. Once all handler routines have completed, the processor state is restored and normal execution resumes. In this manner, hardware accesses are transparently replaced with the execution of SMM handler software.

Historically, SMM software was used primarily for the single purpose of facilitating active power management for notebook designs. That software's only function was to manage the power-up and down of devices to save power. With high performance processors now available, it is feasible to implement, primarily in SMM software, PC capabilities traditionally provided by hardware. In contrast to power management code, this virtualization software generally has strict performance requirements to prevent application performance from being significantly impacted.

Audio SMI Related Registers

The SMI related registers consist of:

- Audio SMI Status Reporting Registers:
  - — Top Level SMI Mirror and Status Registers (F1BAR0+Memory Offset 00h/02h)
  - — Second Level SMI Status Registers (F3BAR0+Memory Offset 10h/12h)
- I/O Trap SMI and Fast Write Status Register (F3BAR0+Memory Offset 14h)
- I/O Trap SMI Enable Register (F3BAR0+Memory Offset 18h)

Audio SMI Status Reporting Registers:

The Top SMI Status Mirror and Status registers are the top level of hierarchy for the SMI handler in determining the source of an SMI. These two registers are at F1BAR0+Memory Offset 00h (Status Mirror) and 02h (Status). The registers are identical except that reading the register at F1BAR0+Memory Offset 02h clears the status.

The second level of audio SMI status reporting is set up very much like the top level. There are two status reporting registers, one "read only" (mirror) and one "read to clear". The data returned by reading either offset is the same (i.e., SMI was caused by an audio related event). The difference between F3BAR0+Memory Offset 12h (Status Mirror) and 10h (Status) is in the ability to clear the SMI source at 10h.

Figure 5-14 shows an SMI tree for checking and clearing the source of an audio SMI. Only the audio SMI bit is detailed here. For details regarding the remaining bits in the Top SMI Status Mirror and Status registers refer to Table 5-33 "F1BAR0+I/O Offset: SMI Status Registers" on page 208.

I/O Trap SMI and Fast Write Status Register:

This 32-bit read-only register (F3BAR0+Memory Offset 14h) not only indicates if the enabled I/O trap generated an SMI, but also contains Fast Path Write related bits.

I/O Trap SMI Enable Register:

The I/O Trap SMI Enable register (F3BAR0+Memory Offset 18h) allows traps for specified I/O addresses and configures generation for I/O events. It also contains the enabling bit for Fast Path Read/Write features.

Status Fast Path Read/Write

Status Fast Path Read – If enabled, the Core Logic module intercepts and responds to reads to several status registers. This speeds up operations, and prevents SMI generation for reads to these registers. This process is called Status Fast Path Read. Status Fast Path Read is enabled via F3BAR0+Memory Offset 18h[4].

In Status Fast Path Read the Core Logic module responds to reads of the following addresses:

388h-38Bh, 2x0h, 2x1h, 2x2h, 2x3h, 2x8h and 2x9h

Note that if neither sound card or FM I/O mapping is enabled, then status read trapping is not possible.

Fast Path Write – If enabled, the Core Logic module captures certain writes to several I/O locations. This feature prevents two SMIs from being asserted for write operations that are known to take two accesses (the first access is an index and the second is data). This process is called Fast Path Write. Fast Path Write is enabled via F3BAR0+Memory Offset 18h[11].

Core Logic Module (Continued)

Fast Path Write captures the data and address bit 1 (A1) of the first access, but does not generate an SMI. A1 is stored in F3BAR0+Memory Offset 14h[15]. The second access causes an SMI, and the data and address are captured as in a normal trapped I/O.

In Fast Path Write, the Core Logic module responds to writes to the following addresses:

388h, 38Ah, 38Bh, 2x0h, 2x2h, and 2x8h

Table 5-38 on page 232 shows the bit formats of the second level SMI status reporting registers and the Fast Path Read/Write programming bits.

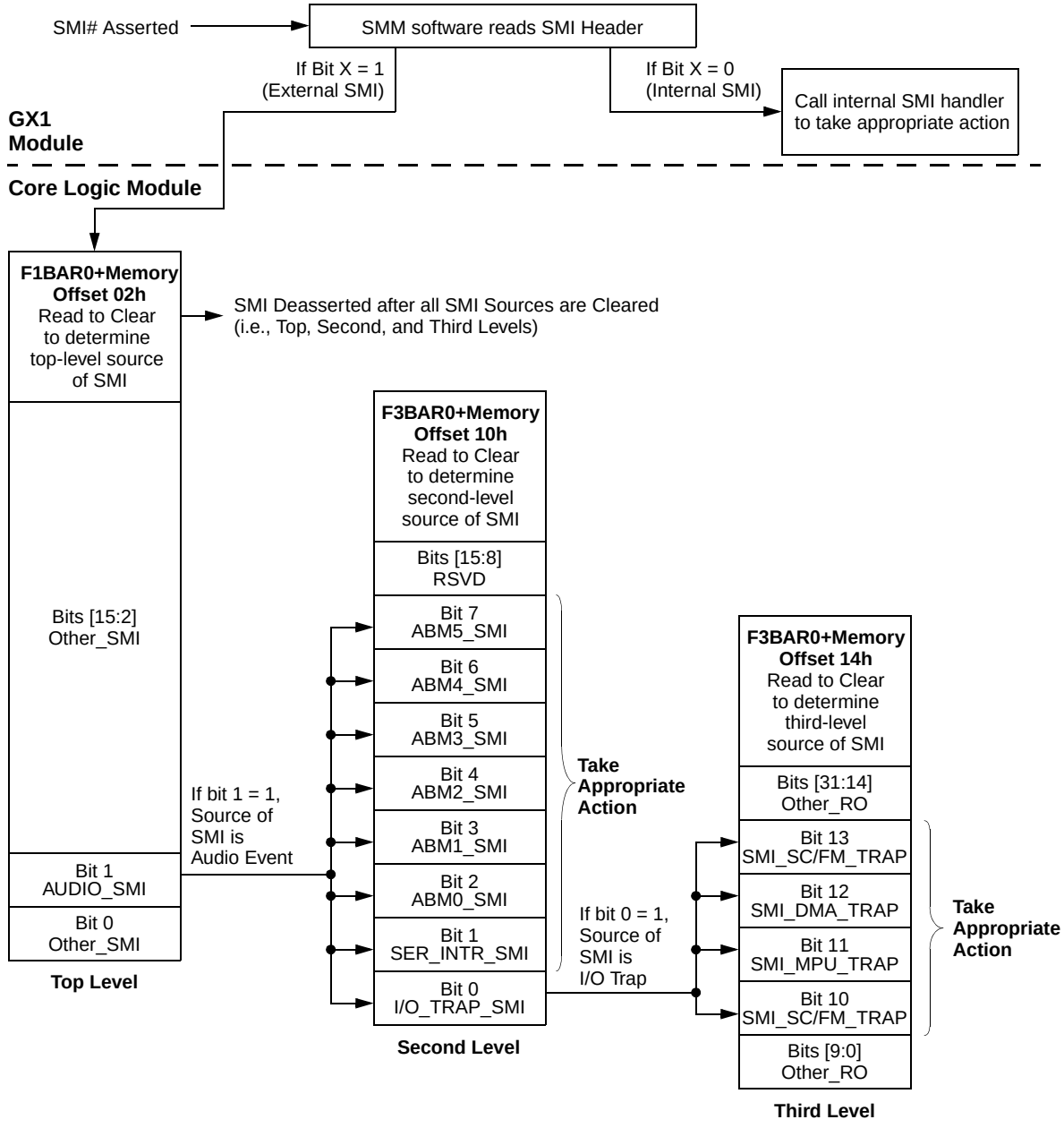


Figure 5-14. Audio SMI Tree Example

Core Logic Module (Continued)

5.2.12.4 IRQ Configuration Registers

The Core Logic module provides the ability to set and clear IRQs internally through software control. If the IRQs are configured for software control, they do not respond to external hardware. There are two registers provided for this feature:

- Internal IRQ Enable Register (F3BAR0+Memory Offset 1Ah)
- Internal IRQ Control Register (F3BAR0+Memory Offset 1Ch)

Internal IRQ Enable Register

The Internal IRQ Enable register configures the IRQs as internal (software) interrupts or external (hardware) interrupts. Any IRQ used as an internal software driven source must be configured as internal.

Internal IRQ Control Register

The Internal IRQ Control register allows individual software assertion/deassertion of the IRQs that are enabled as internal. These bits are used as masks when attempting to write a particular IRQ bit. If the mask bit is set, it can then be asserted/deasserted according to the value in the low-order 16 bits. Otherwise the assertion/deassertion values of the particular IRQ can not be changed.

5.2.13 LPC Interface

The LPC interface of the Core Logic module is based on the Intel Low Pin Count (LPC) Interface specification, revision 1.0. In addition to the required pins specified in the Intel LPC Interface specification, the Core Logic module also supports two optional pins: LDRQ# and SERIRQ.

The following subsections briefly describe some sections of the specification. However, for full details refer to the LPC specification directly.

The goals of the LPC interface are to:

- Enable a system without an ISA bus.
- Reduce the cost of traditional ISA bus devices.
- Use on a motherboard only.
- Perform the same cycle types as the ISA bus: memory, I/O, DMA, and Bus Master.
- Increase the memory space from 16 MB to 4 GB to allow BIOS sizes much greater.
- Provide synchronous design. Much of the challenge of an ISA design is meeting the different, and in some cases conflicting, ISA timings. Make the timings synchronous to a reference well known to component designers, such as PCI.
- Support software transparency: do not require special drivers or configuration for this interface. The motherboard BIOS should be able to configure all devices at boot.
- Support desktop and mobile implementations.
- Enable support of a variable number of wait states.
- Enable I/O memory cycle retries in SMM handler.
- Enable support of wakeup and other power state transitions.

Assumptions and functionality requirements of the LPC interface are:

- Only the following class of devices may be connected to the LPC interface:
  - — SuperI/O (FDC, SP, PP, IR, KBC) - I/O slave, DMA, bus master (for IR, PP).
  - — Audio, including AC97 style design - I/O slave, DMA, bus master.
  - — Generic Memory, including BIOS - Memory slave.
  - — System Management Controller - I/O slave, bus master.
- Interrupts are communicated with the serial interrupt (SERIRQ) protocol.
- The LPC interface does not need to support high-speed buses (such as CardBus, 1394, etc.) downstream, nor does it need to support low-latency buses such as USB.

Figure 5-15 shows a typical setup. In this setup, the LPC is connected through the Core Logic module to a PCI or host bus.

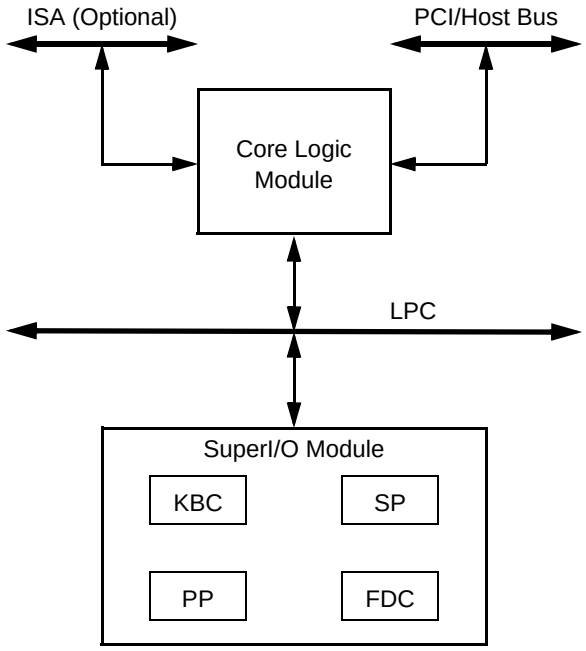


Figure 5-15. Typical Setup

Core Logic Module (Continued)

5.2.13.1 LPC Interface Signal Definitions

The LPC specification lists seven required and six optional signals for supporting the LPC interface. Many of the signals are the same signals found on the PCI interface and do not require any new pins on the host. Required signals must be implemented by both hosts and peripherals. Optional signals may or may not be present on particular hosts or peripherals.

The Core Logic module incorporates all the required LPC interface signals and two of the optional signals:

- Required LPC signals:
  - LAD[3:0] - Multiplexed Command, Address and Data.
  - LFRAME# - Frame: Indicates start of a new cycle, termination of broken cycle.
  - LRESET# - Reset: This signal is not available. Use PCI Reset signal PCIRST# instead.
  - LCLK - Clock: This signal is not available. Use PCI 33 MHz clock signal PCICLK (ball AA23) instead.
- Core Logic module optional LPC signals:
  - LDRQ# - Encoded DMA/Bus Master Request: Only needed by peripheral that need DMA or bus mastering. Peripherals may not share the LDRQ# signal.
  - SERIRQ - Serialized IRQ: Only needed by peripherals that need interrupt support.

5.2.13.2 Cycle Types

Table 5-12 shows the various types of cycles that are supported by the Core Logic module.

Table 5-12. Cycle Types

Cycle Type	Supported Sizes (Bytes)
Memory Read	1
Memory Write	1
I/O Read	1
I/O Write	1
DMA Read	1 or 2
DMA Write	1 or 2
Bus Master Memory Read	1, 2, or 4
Bus Master Memory Write	1, 2, or 4

5.2.13.3 LPC Interface Support

The LPC interface supports all the features described in the LPC Bus Interface specification, revision 1.0, with the following exceptions:

- Only 8- or 16-bit DMA, depending on channel number. Does not support the optional larger transfer sizes.
- Only one external DRQ pin.

Core Logic Module (Continued)

5.3 REGISTER DESCRIPTIONS

The Core Logic module is a multi-function module. Its register space can be broadly divided into three categories in which specific types of registers are located:

1) Chipset Register Space (F0-F3, F5; Note that F4 is Reserved): Comprised of five separate functions, each with its own register space, consisting of PCI header registers and configuration registers.

   The PCI header is a 256-byte region used for configuring a PCI device or function. The first 64 bytes are the same for all PCI devices and are predefined by the PCI specification. These registers are used to configure the PCI for the device. The rest of the 256-byte region is used to configure the device or function itself.

2) USB Controller Register Space (PCIUSB): Consists of the standard PCI header registers. The USB controller supports three ports and is OpenHCI compliant.

3) ISA Legacy Register Space (I/O Ports): Contains all the legacy compatibility I/O ports that are internal, trapped, shadowed, or snooped.

The following subsections provide:

- A brief discussion on how to access the registers located in PCI Configuration Space.
- Core Logic module register summaries.
- Bit formats for Core Logic module registers.

5.3.1 PCI Configuration Space and Access Methods

Configuration cycles are generated in the processor. All configuration registers in the Core Logic module are accessed through the PCI interface using the PCI Type One Configuration Mechanism. This mechanism uses two DWORD I/O locations at 0CF8h and 0CFCh. The first location (0CF8h) references the Configuration Address register. The second location (0CFCh) references the Configuration Data Register (CDR).

To access PCI configuration space, write the Configuration Address (0CF8h) Register with data that specifies the Core Logic module as the device on PCI being accessed, along with the configuration register offset. On the following cycle, a read or write to the Configuration Data Register (CDR) causes a PCI configuration cycle to the Core Logic module. Byte, WORD, or DWORD accesses are allowed to CDR at 0CFCh, 0CFDh, 0CFEh, or 0CFFh.

The Core Logic module has seven PCI configuration register sets, one for each function F0-F3, F4 (Reserved), F5, and USB (PCIUSB). Base Address Registers (BARx) in F0-F3, F5 and PCIUSB set the base addresses for additional I/O or memory mapped configuration registers for each function.

Table 5-13 shows the PCI Configuration Address Register (0CF8h) and how to access the PCI header registers.

Table 5-13. PCI Configuration Address Register (0CF8h)

31	30 24	23 16	15 11	10 8	7 2	1 0
Configuration Space Mapping	**Reserved**	**Bus Number**	**Device Number**	**Function**	**Index**	**DoubleWord 00**
1 (Enable)	000 000	0000 0000	xxxx x **(Note)**	xxx	xxxx xx	00 (Always)
Function 0 (F0): Bridge Configuration, GPIO and LPC Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	000	Index	
Function 1 (F1): SMI Status and ACPI Timer Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	001	Index	
Function 2 (F2): IDE Controller Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	010	Index	
Function 3 (F3): XpressAUDIO Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	011	Index	
Function 4 (F4): Reserved						
80h		0000 0000	1001 0 or 1000 0	100	Index	
Function 5 (F5): X-Bus Expansion Configuration Register Space						
80h		0000 0000	1001 0 or 1000 0	101	Index	
PCIUSB: USB Controller Configuration Register Space						
80h		0000 0000	1001 1 or 1000 1	000	Index	

Note: The device number depends upon the IDSEL Strap Override bit (F5BAR0+I/O Offset 04h[0]). This bit allows selection of the address lines to be used as the IDSEL. By Default: IDSEL = AD28 (1001 0) for F0-F3, F5; AD29 (1001 1) for PCIUSB.

Core Logic Module (Continued)

5.3.2 Register Summary

The tables in this subsection summarize the registers of the Core Logic module. Included in the tables are the register's reset values and page references where the bit formats are found.

Note: Function 4 (F4) is reserved.

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 166
02h-03h	16	RO	Device Identification Register	0510h	Page 166
04h-05h	16	R/W	PCI Command Register	000Fh	Page 166
06h-07h	16	R/W	PCI Status Register	0280h	Page 167
08h	8	RO	Device Revision ID Register	00h	Page 167
09h-0Bh	24	RO	PCI Class Code Register	060100h	Page 167
0Ch	8	R/W	PCI Cache Line Size Register	00h	Page 168
0Dh	8	R/W	PCI Latency Timer Register	00h	Page 168
0Eh	8	RO	PCI Header Type Register	80h	Page 168
0Fh	8	RO	PCI BIST Register	00h	Page 168
10h-13h	32	R/W	Base Address Register 0 (F0BAR0) — Sets the base address for the I/O mapped GPIO Runtime and Configuration Registers (summarized in Table 5-15).	00000001h	Page 168
14h-17h	32	R/W	Base Address Register 1 (F0BAR1) — Sets the base address for the I/O mapped LPC Configuration Registers (summarized in Table 5-16)	00000001h	Page 168
18h-2Bh	--	--	Reserved	00h	Page 168
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 168
2Eh-2Fh	16	RO	Subsystem ID	0500h	Page 168
30h-3Fh	--	--	Reserved	00h	Page 168
40h	8	R/W	PCI Function Control Register 1	39h	Page 169
41h	8	R/W	PCI Function Control Register 2	00h	Page 169
42h	--	--	Reserved	00h	Page 169
43h	8	R/W	PIT Delayed Transactions Register	02h	Page 170
44h	8	R/W	Reset Control Register	01h	Page 170
45h	--	--	Reserved	00h	Page 170
46h	8	R/W	PCI Functions Enable Register	FEh	Page 171
47h	8	R/W	Miscellaneous Enable Register	00h	Page 171
48h-4Bh	--	--	Reserved	00h	Page 171
4Ch-4Fh	32	R/W	Top of System Memory	FFFFFFFFh	Page 171
50h	8	R/W	PIT Control/ISA CLK Divider	7Bh	Page 171
51h	8	R/W	ISA I/O Recovery Control Register	40h	Page 172
52h	8	R/W	ROM/AT Logic Control Register	98h	Page 172
53h	8	R/W	Alternate CPU Support Register	00h	Page 173
54h-59h	--	--	Reserved	00h	Page 173
5Ah	8	R/W	Decode Control Register 1	01h	Page 173
5Bh	8	R/W	Decode Control Register 2	20h	Page 174
5Ch	8	R/W	PCI Interrupt Steering Register 1	00h	Page 174
5Dh	8	R/W	PCI Interrupt Steering Register 2	00h	Page 175
5Eh-5Fh	--	--	Reserved	00h	Page 175
60h-63h	32	R/W	ACPI Control Register	00000000h	Page 175
64h-6Dh	--	--	Reserved	00h	Page 175
6Eh-6Fh	16	R/W	ROM Mask Register	FFF0h	Page 176

Core Logic Module (Continued)

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary (Continued)

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
70h-71h	16	R/W	IOCS1# Base Address Register	0000h	Page 176
72h	8	R/W	IOCS1# Control Register	00h	Page 176
73h	8	---	Reserved	00h	Page 176
74h-75h	16	R/W	IOCS0 Base Address Register	0000h	Page 176
76h	8	R/W	IOCS0 Control Register	00h	Page 176
77h	--	--	Reserved	00h	Page 177
78h-7Bh	32	R/W	DOCCS Base Address Register	00000000h	Page 177
7Ch-7Fh	32	R/W	DOCCS Control Register	00000000h	Page 177
80h	8	R/W	Power Management Enable Register 1	00h	Page 177
81h	8	R/W	Power Management Enable Register 2	00h	Page 178
82h	8	R/W	Power Management Enable Register 3	00h	Page 179
83h	8	R/W	Power Management Enable Register 4	00h	Page 180
84h	8	RO	Second Level PME/SMI Status Mirror Register 1	00h	Page 181
85h	8	RO	Second Level PME/SMI Status Mirror Register 2	00h	Page 182
86h	8	RO	Second Level PME/SMI Status Mirror Register 3	00h	Page 182
87h	8	RO	Second Level PME/SMI Status Mirror Register 4	00h	Page 183
88h	8	R/W	General Purpose Timer 1 Count Register	00h	Page 184
89h	8	R/W	General Purpose Timer 1 Control Register	00h	Page 184
8Ah	8	R/W	General Purpose Timer 2 Count Register	00h	Page 185
8Bh	8	R/W	General Purpose Timer 2 Control Register	00h	Page 185
8Ch	8	R/W	IRQ Speedup Timer Count Register	00h	Page 186
8Dh-92h	--	--	Reserved	00h	Page 186
93h	8	R/W	Miscellaneous Device Control Register	00h	Page 186
94h-95h	16	R/W	Suspend Modulation Register	0000h	Page 186
96h	8	R/W	Suspend Configuration Register	00h	Page 187
97h	--	--	Reserved	00h	Page 187
98h-99h	16	R/W	Hard Disk Idle Timer Count Register — Primary Channel	0000h	Page 187
9Ah-9Bh	16	R/W	Floppy Disk Idle Timer Count Register	0000h	Page 187
9Ch-9Dh	16	R/W	Parallel / Serial Idle Timer Count Register	0000h	Page 188
9Eh-9Fh	16	R/W	Keyboard / Mouse Idle Timer Count Register	0000h	Page 188
A0h-A1h	16	R/W	User Defined Device 1 Idle Timer Count Register	0000h	Page 188
A2h-A3h	16	R/W	User Defined Device 2 Idle Timer Count Register	0000h	Page 188
A4h-A5h	16	R/W	User Defined Device 3 Idle Timer Count Register	0000h	Page 188
AAh-ABh	--	--	Reserved	00h	Page 188
ACh-ADh	16	R/W	Hard Disk Idle Timer Count Register — Secondary Channel	0000h	Page 188
AEh	8	WO	CPU Suspend Command Register	00h	Page 189
AFh	8	WO	Suspend Notebook Command Register	00h	Page 189
B0h-B3h	--	--	Reserved	00h	Page 189
B4h	8	RO	Floppy Port 3F2h Shadow Register	xxh	Page 189
B5h	8	RO	Floppy Port 3F7h Shadow Register	xxh	Page 189
B6h	8	RO	Floppy Port 1F2h Shadow Register	xxh	Page 189
B7h	8	RO	Floppy Port 1F7h Shadow Register	xxh	Page 189
B8h	8	RO	DMA Shadow Register	xxh	Page 189
B9h	8	RO	PIC Shadow Register	xxh	Page 190
BAh	8	RO	PIT Shadow Register	xxh	Page 190
BBh	8	RO	RTC Index Shadow Register	xxh	Page 190
BCh	8	R/W	Clock Stop Control Register	00h	Page 190
BDh-BFh	--	--	Reserved	00h	Page 191

Core Logic Module (Continued)

Table 5-14. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support Summary (Continued)

F0 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-29)
C0h-C3h	32	R/W	User Defined Device 1 Base Address Register	00000000h	Page 191
C4h-C7h	32	R/W	User Defined Device 2 Base Address Register	00000000h	Page 191
C8h-CBh	32	R/W	User Defined Device 3 Base Address Register	00000000h	Page 191
CCh	8	R/W	User Defined Device 1 Control Register	00h	Page 191
CDh	8	R/W	User Defined Device 2 Control Register	00h	Page 191
CEh	8	R/W	User Defined Device 3 Control Register	00h	Page 192
CFh	--	--	Reserved	00h	Page 192
D0h	8	WO	Software SMI Register	00h	Page 192
D1h-EBh	16	--	Reserved	00h	Page 192
ECh	8	R/W	Timer Test Register	00h	Page 192
EDh-F3h	--	--	Reserved	00h	Page 192
F4h	8	RC	Second Level PME/SMI Status Register 1	00h	Page 192
F5h	8	RC	Second Level PME/SMI Status Register 2	00h	Page 193
F6h	8	RC	Second Level PME/SMI Status Register 3	00h	Page 194
F7h	8	RC	Second Level PME/SMI Status Register 4	00h	Page 195
F8h-FFh	--	--	Reserved	00h	Page 195

Table 5-15. F0BAR0: GPIO Support Registers Summary

F0BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-30)
00h-03h	32	R/W	GPDO0 — GPIO Data Out 0 Register	FFFFFFFFh	Page 196
04h-07h	32	RO	GPDI0 — GPIO Data In 0 Register	FFFFFFFFh	Page 196
08h-0Bh	32	R/W	GPIEN0 — GPIO Interrupt Enable 0 Register	00000000h	Page 196
0Ch-0Fh	32	R/W1C	GPST0 — GPIO Status 0 Register	00000000h	Page 196
10h-13h	32	R/W	GPDO1 — GPIO Data Out 1 Register	FFFFFFFFh	Page 197
14h-17h	32	RO	GPDI1 — GPIO Data In 1 Register	FFFFFFFFh	Page 197
18h-1Bh	32	R/W	GPIEN1 — GPIO Interrupt Enable 1 Register	00000000h	Page 197
1Ch-1Fh	32	R/W1C	GPST1 — GPIO Status 1 Register	00000000h	Page 197
20h-23h	32	R/W	GPIO Signal Configuration Select Register	00000000h	Page 197
24h-27h	32	R/W	GPIO Signal Configuration Access Register	00000044h	Page 198
28h-2Bh	32	R/W	GPIO Reset Control Register	00000000h	Page 199

Table 5-16. F0BAR1: LPC Support Registers Summary

F0BAR1+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-31)
00h-03h	32	R/W	SERIRQ_SRC — Serial IRQ Source Register	00000000h	Page 200
04h-07h	32	R/W	SERIRQ_LVL — Serial IRQ Level Control Register	00000000h	Page 201
08h-0Bh	32	R/W	SERIRQ_CNT — Serial IRQ Control Register	00000000h	Page 203
0Ch-0Fh	32	R/W	DRQ_SRC — DRQ Source Register	00000000h	Page 203
10h-13h	32	R/W	LAD_EN — LPC Address Enable Register	00000000h	Page 204
14h-17h	32	R/W	LAD_D0 — LPC Address Decode 0 Register	00080020h	Page 204
18h-1Bh	32	R/W	LAD_D1 — LPC Address Decode 1 Register	00000000h	Page 205
1Ch-1Fh	32	R/W	LPC_ERR_SMI — LPC Error SMI Register	00000080h	Page 205
20h-23h	32	RO	LPC_ERR_ADD — LPC Error Address Register	00000000h	Page 206

Core Logic Module (Continued)

Table 5-17. F1: PCI Header Registers for SMI Status and ACPI Support Summary

F1 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-32)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 207
02h-03h	16	RO	Device Identification Register	0511h	Page 207
04h-05h	16	R/W	PCI Command Register	0000h	Page 207
06h-07h	16	RO	PCI Status Register	0280h	Page 207
08h	8	RO	Device Revision ID Register	00h	Page 207
09h-0Bh	24	RO	PCI Class Code Register	068000h	Page 207
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 207
0Dh	8	RO	PCI Latency Timer Register	00h	Page 207
0Eh	8	RO	PCI Header Type Register	00h	Page 207
0Fh	8	RO	PCI BIST Register	00h	Page 207
10h-13h	32	R/W	Base Address Register 0 (F1BAR0) — Sets the base address for the I/O mapped SMI Status Registers (summarized in Table 5-18).	00000001h	Page 207
14h-2Bh	--	--	Reserved	00h	Page 207
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 207
2Eh-2Fh	16	RO	Subsystem ID	0501h	Page 207
30h-3Fh	--	--	Reserved	00h	Page 207
40h-43h	32	R/W	Base Address Register 1 (F1BAR1) — Sets the base address for the I/O mapped ACPI Support Registers (summarized in Table 5-19)	00000001h	Page 207
44h-FFh	--	--	Reserved	00h	Page 207

Table 5-18. F1BAR0: SMI Status Registers Summary

F1BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-33)
00h-01h	16	RO	Top Level PME/SMI Status Mirror Register	0000h	Page 208
02h-03h	16	RO/RC	Top Level PME/SMI Status Register	0000h	Page 209
04h-05h	16	RO	Second Level General Traps & Timers PME/SMI Status Mirror Register	0000h	Page 210
06h-07h	16	RC	Second Level General Traps & Timers PME/SMI Status Register	0000h	Page 211
08h-09h	16	Read to Enable	SMI Speedup Disable Register	0000h	Page 212
0Ah-1Bh	--	--	Reserved	00h	Page 212
1Ch-1Fh	32	RO	ACPI Timer Register	xxxxxxxxh	Page 212
20h-21h	16	RO	Second Level ACPI PME/SMI Status Mirror Register	0000h	Page 212
22h-23h	16	RC	Second Level ACPI PME/SMI Status Register	0000h	Page 213
24h-27h	32	R/W	External SMI Register	00000000h	Page 213
28h-4Fh	--	--	Not Used	00h	Page 215
50h-FFh	--	--	The I/O mapped registers located here (F1BAR0+I/O Offset 50h-FFh) are also accessible at F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space.		

Core Logic Module (Continued)

Table 5-19. F1BAR1: ACPI Support Registers Summary

F1BAR1+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-34)
00h-03h	32	R/W	P_CNT — Processor Control Register	00000000h	Page 216
04h	8	RO	Reserved, do not read	00h	Page 216
05h	8	RO	P_LVL3 — Enter C3 Power State Register	xxh	Page 216
06h	8	R/W	SMI_CMD — OS/BIOS Requests Register	00h	Page 216
07h	8	R/W	ACPI_FUN_CNT — ACPI Function Control Register	00h	Page 216
08h-09h	16	R/W	PM1A_STS — PM1A Status Register	0000h	Page 217
0Ah-0Bh	16	R/W	PM1A_EN — PM1A Enable Register	0000h	Page 218
0Ch-0Dh	16	R/W	PM1A_CNT — PM1A Control Register	0000h	Page 218
0Eh	8	R/W	ACPI_BIOS_STS Register	00h	Page 219
0Fh	8	R/W	ACPI_BIOS_EN Register	00h	Page 219
10h-11h	16	R/W	GPE0_STS — General Purpose Event 0 Status Register	xxxxh	Page 219
12h-13h	16	R/W	GPE0_EN — General Purpose Event 0 Enable Register	0000h	Page 221
14h	8	R/W	GPWIO Control Register 1	00h	Page 222
15h	8	R/W	GPWIO Control Register 2	00h	Page 222
16h	8	R/W	GPWIO Data Register	00h	Page 223
17h	--	--	Reserved	00h	Page 223
18h-1Bh	32	R/W	ACPI SCI_ROUTING Register	00000F00h	Page 223
1Ch-1Fh	32	RO	PM_TMR — PM Timer Register	xxxxxxxxh	Page 224
20h	8	R/W	PM2_CNT — PM2 Control Register	00h	Page 224
21h-FFh	--	--	Not Used	00h	Page 224

Core Logic Module (Continued)

Table 5-20. F2: PCI Header Registers for IDE Controller Support Summary

F2 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-35)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 225
02h-03h	16	RO	Device Identification Register	0502h	Page 225
04h-05h	16	R/W	PCI Command Register	0000h	Page 225
06h-07h	16	RO	PCI Status Register	0280h	Page 225
08h	8	RO	Device Revision ID Register	01h	Page 225
09h-0Bh	24	RO	PCI Class Code Register	010180h	Page 225
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 225
0Dh	8	RO	PCI Latency Timer Register	00h	Page 225
0Eh	8	RO	PCI Header Type Register	00h	Page 225
0Fh	8	RO	PCI BIST Register	00h	Page 225
10h-13h	32	RO	Base Address Register 0 (F2BAR0) — Reserved for possible future use by the Core Logic module.	00000000h	Page 225
14h-17h	32	RO	Base Address Register 1 (F2BAR1) — Reserved for possible future use by the Core Logic module.	00000000h	Page 225
18h-1Bh	32	RO	Base Address Register 2 (F2BAR2) — Reserved for possible future use by the Core Logic module.	00000000h	Page 225
1Ch-1Fh	32	RO	Base Address Register 3 (F2BAR3) — Reserved for possible future use by the Core Logic module.	00000000h	Page 225
20h-23h	32	R/W	Base Address Register 4 (F2BAR4) — Sets the base address for the I/O mapped Bus Master IDE Registers (summarized in Table 5-21)	00000001h	Page 225
24h-2Bh	--	--	Reserved	00h	Page 225
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 225
2Eh-2Fh	16	RO	Subsystem ID	0502h	Page 225
30h-3Fh	--	--	Reserved	00h	Page 225
40h-43h	32	R/W	Channel 0 Drive 0 PIO Register	00009172h	Page 226
44h-47h	32	R/W	Channel 0 Drive 0 DMA Control Register	00077771h	Page 227
48h-4Bh	32	R/W	Channel 0 Drive 1 PIO Register	00009172h	Page 227
4Ch-4Fh	32	R/W	Channel 0 Drive 1 DMA Control Register	00077771h	Page 227
50h-FFh	--	--	Reserved	00h	Page 228

Table 5-21. F2BAR4: IDE Controller Support Registers Summary

F2BAR4+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-36)
00h	8	R/W	IDE Bus Master 0 Command Register — Primary	00h	Page 229
01h	--	--	Not Used	--	Page 229
02h	8	R/W	IDE Bus Master 0 Status Register — Primary	00h	Page 229
03h	--	--	Not Used	--	Page 229
04h-07h	32	R/W	IDE Bus Master 0 PRD Table Address — Primary	00000000h	Page 229

Core Logic Module (Continued)

Table 5-22. F3: PCI Header Registers for XpressAUDIO Support Summary

F3 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-37)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 231
02h-03h	16	RO	Device Identification Register	0503h	Page 231
04h-05h	16	R/W	PCI Command Register	0000h	Page 231
06h-07h	16	RO	PCI Status Register	0280h	Page 231
08h	8	RO	Device Revision ID Register	00h	Page 231
09h-0Bh	24	RO	PCI Class Code Register	040100h	Page 231
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 231
0Dh	8	RO	PCI Latency Timer Register	00h	Page 231
0Eh	8	RO	PCI Header Type Register	00h	Page 231
0Fh	8	RO	PCI BIST Register	00h	Page 231
10h-13h	32	R/W	Base Address Register 0 (F3BAR0) — Sets the base address for the memory mapped VSA audio interface control register block (summarized in Table 5-23).	00000000h	Page 231
14h-2Bh	--	--	Reserved	00h	Page 231
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 231
2Eh-2Fh	16	RO	Subsystem ID	0503h	Page 231
30h-FFh	--	--	Reserved	00h	Page 231

Table 5-23. F3BAR0: XpressAUDIO Support Registers Summary

F3BAR0+ Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-38)
00h-03h	32	R/W	Codec GPIO Status Register	00000000h	Page 232
04h-07h	32	R/W	Codec GPIO Control Register	00000000h	Page 232
08h-0Bh	32	R/W	Codec Status Register	00000000h	Page 232
0Ch-0Fh	32	R/W	Codec Command Register	00000000h	Page 233
10h-11h	16	RC	Second Level Audio SMI Status Register	0000h	Page 233
12h-13h	16	RO	Second Level Audio SMI Status Mirror Register	0000h	Page 234
14h-17h	32	RO	I/O Trap SMI and Fast Write Status Register	00000000h	Page 235
18h-19h	16	R/W	I/O Trap SMI Enable Register	0000h	Page 236
1Ah-1Bh	16	R/W	Internal IRQ Enable Register	0000h	Page 237
1Ch-1Fh	32	R/W	Internal IRQ Control Register	00000000h	Page 238
20h	8	R/W	Audio Bus Master 0 Command Register	00h	Page 239
21h	8	RC	Audio Bus Master 0 SMI Status Register	00h	Page 240
22h-23h	--	--	Not Used	---	Page 240
24h-27h	--	R/W	Audio Bus Master 0 PRD Table Address	00000000h	Page 240
28h	8	R/W	Audio Bus Master 1 Command Register	00h	Page 240
29h	8	RC	Audio Bus Master 1 SMI Status Register	00h	Page 241
2Ah-2Bh	--	--	Not Used	---	Page 241
2Ch-2Fh	32	R/W	Audio Bus Master 1 PRD Table Address	00000000h	Page 241
30h	8	R/W	Audio Bus Master 2 Command Register	00h	Page 241
31h	8	RC	Audio Bus Master 2 SMI Status Register	00h	Page 242
32h-33h	--	--	Not Used	00h	Page 242
34h-37h	32	R/W	Audio Bus Master 2 PRD Table Address	00000000h	Page 242
38h	8	R/W	Audio Bus Master 3 Command Register	00h	Page 242
39h	8	RC	Audio Bus Master 3 SMI Status Register	00h	Page 243
3Ah-3Bh	--	--	Not Used	---	Page 243

Core Logic Module (Continued)

Table 5-23. F3BAR0: XpressAUDIO Support Registers Summary (Continued)

F3BAR0+ Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-38)
3Ch-3Fh	32	R/W	Audio Bus Master 3 PRD Table Address	00000000h	Page 243
40h	8	R/W	Audio Bus Master 4 Command Register	00h	Page 243
41h	8	RC	Audio Bus Master 4 SMI Status Register	00h	Page 244
42h-43h	--	--	Not Used	---	Page 244
44h-47h	32	R/W	Audio Bus Master 4 PRD Table Address	00000000h	Page 244
48h	8	R/W	Audio Bus Master 5 Command Register	00h	Page 244
49h	8	RC	Audio Bus Master 5 SMI Status Register	00h	Page 245
4Ah-4Bh	--	--	Not Used	---	Page 245
4Ch-4Fh	32	R/W	Audio Bus Master 5 PRD Table Address	00000000h	Page 245

Core Logic Module (Continued)

Table 5-24. F5: PCI Header Registers for X-Bus Expansion Support Summary

F5 Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-39)
00h-01h	16	RO	Vendor Identification Register	100Bh	Page 246
02h-03h	16	RO	Device Identification Register	0515h	Page 246
04h-05h	16	R/W	PCI Command Register	0000h	Page 246
06h-07h	16	RO	PCI Status Register	0280h	Page 246
08h	8	RO	Device Revision ID Register	00h	Page 246
09h-0Bh	24	RO	PCI Class Code Register	068000h	Page 246
0Ch	8	RO	PCI Cache Line Size Register	00h	Page 246
0Dh	8	RO	PCI Latency Timer Register	00h	Page 246
0Eh	8	RO	PCI Header Type Register	00h	Page 246
0Fh	8	RO	PCI BIST Register	00h	Page 246
10h-13h	32	R/W	Base Address Register 0 (F5BAR0) — Sets the base address for the X-Bus Expansion support registers (summarized in Table 5-25.)	00000000h	Page 246
14h-17h	32	R/W	Base Address Register 1 (F5BAR1) — Reserved for possible future use by the Core Logic module.	00000000h	Page 246
18h-1Bh	32	R/W	Base Address Register 2 (F5BAR2) — Reserved for possible future use by the Core Logic module.	00000000h	Page 246
1Ch-1Fh	32	R/W	Base Address Register 3 (F5BAR3) — Reserved for possible future use by the Core Logic module.	00000000h	Page 246
20h-23h	32	R/W	Base Address Register 4 (F5BAR4) — Reserved for possible future use by the Core Logic module.	00000000h	Page 247
24h-27h	32	R/W	Base Address Register 5 (F5BAR5) — Reserved for possible future use by the Core Logic module.	00000000h	Page 247
28h-2Bh	--	--	Reserved	00h	Page 247
2Ch-2Dh	16	RO	Subsystem Vendor ID	100Bh	Page 247
2Eh-2Fh	16	RO	Subsystem ID	0505h	Page 247
30h-3Fh	--	--	Reserved	00h	Page 247
40h-43h	32	R/W	F5BAR0 Base Address Register Mask	FFFFFFC1h	Page 247
44h-47h	32	R/W	F5BAR1 Base Address Register Mask	00000000h	Page 248
48h-4Bh	32	R/W	F5BAR2 Base Address Register Mask	00000000h	Page 248
4Ch-4Fh	32	R/W	F5BAR3 Base Address Register Mask	00000000h	Page 248
50h-53h	32	R/W	F5BAR4 Base Address Register Mask	00000000h	Page 248
54h-57h	32	R/W	F5BAR5 Base Address Register Mask	00000000h	Page 248
58h	8	R/W	F5BARx Initialized Register	00h	Page 249
59h-FFh	--	--	Reserved	xxh	Page 249
60h-63h	32	R/W	Scratchpad for Chip Number	00000000h	Page 249
64h-67h	32	R/W	Scratchpad for Configuration Block Address	00000000h	Page 249
68h-FFh	--	--	Reserved	---	Page 249

Table 5-25. F5BAR0: I/O Control Support Registers Summary

F5BAR0+ I/O Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-40)
00h-03h	32	R/W	I/O Control Register 1	010C0007h	Page 250
04h-07h	32	R/W	I/O Control Register 2	00000002h	Page 250
08h-0Bh	32	R/W	I/O Control Register 3	00009000h	Page 250

Core Logic Module (Continued)

Table 5-26. PCIUSB: USB PCI Configuration Register Summary

PCIUSB Index	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-41)
00h-01h	16	RO	Vendor Identification	0E11h	Page 251
02h-03h	16	RO	Device Identification	A0F8h	Page 251
04h-05h	16	R/W	Command Register	00h	Page 251
06h-07h	16	R/W	Status Register	0280h	Page 252
08h	8	RO	Device Revision ID	08h	Page 252
09h-0Bh	24	RO	Class Code	0C0310h	Page 252
0Ch	8	R/W	Cache Line Size	00h	Page 252
0Dh	8	R/W	Latency Timer	00h	Page 252
0Eh	8	RO	Header Type	00h	Page 252
0Fh	8	RO	BIST Register	00h	Page 252
10h-13h	32	R/W	Base Address 0	00000000h	Page 252
14h-2Bh	--	--	Reserved	00h	Page 253
2Ch-2Dh	16	RO	Subsystem Vendor ID	0E11h	Page 253
2Eh-2Fh	16	RO	Subsystem ID	A0F8h	Page 253
30h-3Bh	--	--	Reserved	00h	Page 253
3Ch	8	R/W	Interrupt Line Register	00h	Page 253
3Dh	8	R/W	Interrupt Pin Register	01h	Page 253
3Eh	8	RO	Min. Grant Register	00h	Page 253
3Fh	8	RO	Max. Latency Register	50h	Page 253
40h-43h	32	R/W	ASIC Test Mode Enable Register	000F0000h	Page 253
44h	8	R/W	ASIC Operational Mode Enable	00h	Page 253
45h-FFh	--	--	Reserved	00h	Page 253

Core Logic Module (Continued)

Table 5-27. USB_BAR: USB Controller Registers Summary

USB_BAR0 +Memory Offset	Width (Bits)	Type	Name	Reset Value	Reference (Table 5-42)
00h-03h	32	R/W	HcRevision	00000110h	Page 254
04h-07h	32	R/W	HcControl	00000000h	Page 254
08h-0Bh	32	R/W	HcCommandStatus	00000000h	Page 254
0Ch-0Fh	32	R/W	HcInterruptStatus	00000000h	Page 255
10h-13h	32	R/W	HcInterruptEnable	00000000h	Page 255
14h-17h	32	R/W	HcInterruptDisable	00000000h	Page 256
18h-1Bh	32	R/W	HcHCCA	00000000h	Page 256
1Ch-1Fh	32	R/W	HcPeriodCurrentED	00000000h	Page 256
20h-23h	32	R/W	HcControlHeadED	00000000h	Page 256
24h-27h	32	R/W	HcControlCurrentED	00000000h	Page 256
28h-2Bh	32	R/W	HcBulkHeadED	00000000h	Page 256
2Ch-2Fh	32	R/W	HcBulkCurrentED	00000000h	Page 257
30h-33h	32	R/W	HcDoneHead	00000000h	Page 257
34h-37h	32	R/W	HcFmInterval	00002EDFh	Page 257
38h-3Bh	32	RO	HcFrameRemaining	00000000h	Page 257
3Ch-3Fh	32	RO	HcFmNumber	00000000h	Page 257
40h-43h	32	R/W	HcPeriodicStart	00000000h	Page 257
44h-47h	32	R/W	HcLSThreshold	00000628h	Page 257
48h-4Bh	32	R/W	HcRhDescriptorA	01000002h	Page 258
4Ch-4Fh	32	R/W	HcRhDescriptorB	00000000h	Page 258
50h-53h	32	R/W	HcRhStatus	00000000h	Page 259
54h-57h	32	R/W	HcRhPortStatus[1]	00000000h	Page 259
58h-5Bh	32	R/W	HcRhPortStatus[2]	00000000h	Page 260
5Ch-5Fh	32	R/W	HcRhPortStatus[3]	00000000h	Page 262
60h-9Fh	--	--	Reserved	xxxxxxxxh	Page 263
100h-103h	32	R/W	HceControl	00000000h	Page 263
104h-107h	32	R/W	HceInput	000000xxh	Page 263
108h-10Dh	32	R/W	HceOutput	000000xxh	Page 263
10Ch-10Fh	32	R/W	HceStatus	00000000h	Page 264

Core Logic Module (Continued)

Table 5-28. ISA Legacy I/O Register Summary

I/O Port	Type	Name	Reference
DMA Channel Control Registers (Table 5-43)			
000h	R/W	DMA Channel 0 Address Register	Page 265
001h	R/W	DMA Channel 0 Transfer Count Register	Page 265
002h	R/W	DMA Channel 1 Address Register	Page 265
003h	R/W	DMA Channel 1 Transfer Count Register	Page 265
004h	R/W	DMA Channel 2 Address Register	Page 265
005h	R/W	DMA Channel 2 Transfer Count Register	Page 265
006h	R/W	DMA Channel 3 Address Register	Page 265
007h	R/W	DMA Channel 3 Transfer Count Register	Page 265
008h	Read	DMA Status Register, Channels 3:0	Page 265
	Write	DMA Command Register, Channels 3:0	Page 266
009h	WO	Software DMA Request Register, Channels 3:0	Page 266
00Ah	W	DMA Channel Mask Register, Channels 3:0	Page 267
00Bh	WO	DMA Channel Mode Register, Channels 3:0	Page 267
00Ch	WO	DMA Clear Byte Pointer Command, Channels 3:0	Page 267
00Dh	WO	DMA Master Clear Command, Channels 3:0	Page 267
00Eh	WO	DMA Clear Mask Register Command, Channels 3:0	Page 267
00Fh	WO	DMA Write Mask Register Command, Channels 3:0	Page 267
0C0h	R/W	DMA Channel 4 Address Register (Not used)	Page 267
0C2h	R/W	DMA Channel 4 Transfer Count Register (Not Used)	Page 267
0C4h	R/W	DMA Channel 5 Address Register	Page 267
0C6h	R/W	DMA Channel 5 Transfer Count Register	Page 267
0C8h	R/W	DMA Channel 6 Address Register	Page 267
0CAh	R/W	DMA Channel 6 Transfer Count Register	Page 268
0CCh	R/W	DMA Channel 7 Address Register	Page 268
0CEh	R/W	DMA Channel 7 Transfer Count Register	Page 268
0D0h	Read	DMA Status Register, Channels 7:4	Page 268
	Write	DMA Command Register, Channels 7:4	Page 268
0D2h	WO	Software DMA Request Register, Channels 7:4	Page 269
0D4h	W	DMA Channel Mask Register, Channels 7:4	Page 269
0D6h	WO	DMA Channel Mode Register, Channels 7:4	Page 269
0D8h	WO	DMA Clear Byte Pointer Command, Channels 7:4	Page 269
0DAh	WO	DMA Master Clear Command, Channels 7:4	Page 269
0DCh	WO	DMA Clear Mask Register Command, Channels 7:4	Page 269
0DEh	WO	DMA Write Mask Register Command, Channels 7:4	Page 269
DMA Page Registers (Table 5-44)			
081h	R/W	DMA Channel 2 Low Page Register	Page 270
082h	R/W	DMA Channel 3 Low Page Register	Page 270
083h	R/W	DMA Channel 1 Low Page Register	Page 270
087h	R/W	DMA Channel 0 Low Page Register	Page 270
089h	R/W	DMA Channel 6 Low Page Register	Page 270
08Ah	R/W	DMA Channel 7 Low Page Register	Page 270
08Bh	R/W	DMA Channel 5 Low Page Register	Page 270
08Fh	R/W	Sub-ISA Refresh Low Page Register	Page 270
481h	R/W	DMA Channel 2 High Page Register	Page 270
482h	R/W	DMA Channel 3 High Page Register	Page 270
483h	R/W	DMA Channel 1 High Page Register	Page 270
487h	R/W	DMA Channel 0 High Page Register	Page 270

Core Logic Module (Continued)

Table 5-28. ISA Legacy I/O Register Summary (Continued)

I/O Port	Type	Name	Reference
489h	R/W	DMA Channel 6 High Page Register	Page 270
48Ah	R/W	DMA Channel 7 High Page Register	Page 270
48Bh	R/W	DMA Channel 5 High Page Register	Page 270
Programmable Interval Timer Registers (Table 5-45)			
040h	W	PIT Timer 0 Counter	Page 271
	R	PIT Timer 0 Status	Page 271
041h	W	PIT Timer 1 Counter (Refresh)	Page 271
	R	PIT Timer 1 Status (Refresh)	Page 271
042h	W	PIT Timer 2 Counter (Speaker)	Page 271
	R	PIT Timer 2 Status (Speaker)	Page 271
043h	R/W	PIT Mode Control Word Register	Page 272
		Read Status Command	
		Counter Latch Command	
Programmable Interrupt Controller Registers (Table 5-46)			
020h / 0A0h	WO	Master / Slave PCI ICW1	Page 273
021h / 0A1h	WO	Master / Slave PIC ICW2	Page 273
021h / 0A1h	WO	Master / Slave PIC ICW3	Page 273
021h / 0A1h	WO	Master / Slave PIC ICW4	Page 273
021h / 0A1h	R/W	Master / Slave PIC OCW1	Page 273
020h / 0A0h	WO	Master / Slave PIC OCW2	Page 274
020h / 0A0h	WO	Master / Slave PIC OCW3	Page 274
020h / 0A0h	RO	Master / Slave PIC Interrupt Request and Service Registers for OCW3 Commands	Page 274
Keyboard Controller Registers (Table 5-47)			
060h	R/W	External Keyboard Controller Data Register	Page 276
061h	R/W	Port B Control Register	Page 276
062h	R/W	External Keyboard Controller Mailbox Register	Page 276
064h	R/W	External Keyboard Controller Command Register	Page 276
066h	R/W	External Keyboard Controller Mailbox Register	Page 276
092h	R/W	Port A Control Register	Page 276
Real-Time Clock Registers (Table 5-48)			
070h	WO	RTC Address Register	Page 277
071h	R/W	RTC Data Register	Page 277
072h	WO	RTC Extended Address Register	Page 277
073h	R/W	RTC Extended Data Register	Page 277
Miscellaneous Registers (Table 5-49)			
0F0h, 0F1h	WO	Coprocessor Error Register	Page 277
1F0-1F7h/ 3F6h-3F7h	R/W	Primary IDE Registers	Page 277
4D0h	R/W	Interrupt Edge/Level Select Register 1	Page 277
4D1h	R/W	Interrupt Edge/Level Select Register 2	Page 278

Core Logic Module (Continued)

5.4 CHIPSET REGISTER SPACE

The Chipset Register Space of the Core Logic module is comprised of five separate functions (F0-F3 and F5, note that F4 is reserved), each with its own register space. Base Address Registers (BARs) in each PCI header register space set the base address for the configuration registers for each respective function. The configuration registers accessed through BARs are I/O or memory mapped. The PCI header registers in all functions are very similar.

1) Function 0 (F0): PCI Header/Bridge Configuration Registers for GPIO, and LPC Support (see Section 5.4.1).
2) Function 1 (F1): PCI Header Registers for SMI Status and ACPI Support (see Section 5.4.2 on page 207).
3) Function 2 (F2): PCI Header/Channel 0 and 1 Configuration Registers for IDE Controller Support (see Section 5.4.3 on page 225).
4) Function 3 (F3): PCI Header Registers for XpressAUDIO Audio Support (see Section 5.4.4 on page 231).
5) Function 4 (F4): Reserved.
6) Function 5 (F5): PCI Header Registers for X-Bus Expansion Support (see Section 5.4.5 on page 246).

   Function 5 contains six BARs in their standard PCI header locations (i.e., Index 10h, 14h, 18h, 1Ch, 20h, and 24h). In addition there are six mask registers that allow the six BARs to be fully programmable from 4 GB to 16 bytes for memory and from 4 GB to 4 bytes for I/O.

General Remarks:

- Reserved bits that are defined as "must be set to 0 or 1" should be written with that value.
- Reserved bits that are not defined as "must be set to 0 or 1" should be written with a value that is read from them.
- "Read to Clear" registers that are wider than one byte should be read in one read operation. If they are read a byte at a time, status bits may be lost, or not cleared.

5.4.1 Bridge, GPIO, and LPC Registers - Function 0

The register space designated as Function 0 (F0) is used to configure Bridge features and functionality unique to the Core Logic module. In addition, it configures the PCI portion of support hardware for the GPIO and LPC support registers. The bit formats for the PCI Header and Bridge Configuration registers are given in Table 5-29.

Note: The registers at F0 Index 50h-FFh can also be accessed at F1BAR0+I/O Offset 50h-FFh. However, the preferred method is to program these registers through the F0 register space.

Located in the PCI Header registers of F0, are two Base Address Registers (F0BARx) used for pointing to the register spaces designated for GPIO and LPC configuration (described in Section 5.4.1.1 "GPIO Support Registers" on page 196 and Section 5.4.1.2 "LPC Support Registers" on page 200).

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0510h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 000Fh**
15:10	**Reserved.** Must be set to 0.
9	**Fast Back-to-Back Enable.** This function is not supported when the Core Logic module is a master. It must always be disabled (i.e., must be set to 0).
8	**SERR#.** Allow SERR# assertion on detection of special errors. 0: Disable. (Default) 1: Enable.
7	**Wait Cycle Control (Read Only).** This function is not supported in the Core Logic module. It is always disabled (always reads 0, hardwired).
6	**Parity Error.** Allow the Core Logic module to check for parity errors on PCI cycles for which it is a target and to assert PERR# when a parity error is detected. 0: Disable. (Default) 1: Enable.
5	**VGA Palette Snoop Enable. (Read Only)** This function is not supported in the Core Logic module. It is always disabled (always reads 0, hardwired).
4	**Memory Write and Invalidate.** Allow the Core Logic module to do memory write and invalidate cycles, if the PCI Cache Line register (F0 Index 0Ch) is set to 32 bytes (08h). 0: Disable. (Default) 1: Enable.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
3	**Special Cycles.** Allow the Core Logic module to respond to special cycles. 0: Disable. 1: Enable. (Default) This bit must be enabled to allow the internal CPU Warm Reset signal to be triggered from a CPU Shutdown cycle.
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. (Default)
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus: 0: Disable. 1: Enable. (Default) This bit must be set to 1 to access I/O offsets through F0BAR0 and F0BAR1 (see F0 Index 10h and 14h).
Index 06h-07h	**PCI Status Register (R/W)** — **Reset Value: 0280h**
15	**Detected Parity Error.** This bit is set whenever a parity error is detected. Write 1 to clear.
14	**Signaled System Error.** This bit is set whenever the Core Logic module asserts SERR# active. Write 1 to clear.
13	**Received Master Abort.** This bit is set whenever a master abort cycle occurs. A master abort occurs when a PCI cycle is not claimed, except for special cycles. Write 1 to clear.
12	**Received Target Abort.** This bit is set whenever a target abort is received while the Core Logic module is the master for the PCI cycle. Write 1 to clear.
11	**Signaled Target Abort.** This bit is set whenever the Core Logic module signals a target abort. This occurs when an address parity error occurs for an address that hits in the active address decode space of the Core Logic module. Write 1 to clear.
10:9	**DEVSEL# Timing. (Read Only)** These bits are always 01, as the Core Logic module always responds to cycles for which it is an active target with medium DEVSEL# timing. 00: Fast. 01: Medium. 10: Slow. 11: Reserved.
8	**Data Parity Detected.** This bit is set when: 1) The Core Logic module asserts PERR# or observed PERR# asserted. 2) The Core Logic module is the master for the cycle in which the PERR# occurred, and PE is set (F0 Index 04h[6] = 1). Write 1 to clear.
7	**Fast Back-to-Back Capable. (Read Only)** Enables the Core Logic module, as a target, to accept fast back-to-back transactions. 0: Disable. 1: Enable. This bit is always set to 1.
6:0	**Reserved. (Read Only)** Must be set to 0 for future use.
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 060100h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 0Ch	**PCI Cache Line Size Register (R/W)** — **Reset Value: 00h**
7:0	**PCI Cache Line Size Register.** This register sets the size of the PCI cache line, in increments of four bytes. For memory write and invalidate cycles, the PCI cache line size must be set to 32 bytes (08h) and the Memory Write and Invalidate bit (F0 Index 04h[4]) must be set to 1.
Index 0Dh	**PCI Latency Timer Register (R/W)** — **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0.
3:0	**PCI Latency Timer Value.** The PCI Latency Timer register prevents system lockup when a slave does not respond to a cycle that the Core Logic module masters. If the value is set to 00h (default), the timer is disabled. If the timer is written with any other value, bits [3:0] become the four most significant bits in a timer that counts PCI clocks for slave response. The timer is reset on each valid data transfer. If the counter expires before the next assertion of TRDY# is received, the Core Logic module stops the transaction with a master abort and asserts SERR#, if enabled to do so (via F0 Index 04h[8]).
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 80h**
7:0	**PCI Header Type Register.** This register defines the format of this header. This header has a format of type 0. (For more information about this format, see the PCI Local Bus specification, revision 2.2.) Additionally, bit 7 of this register defines whether this PCI device is a multifunction device (bit 7 = 1) or not (bit 7 = 0).
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
This register indicates various information about the PCI Built-In Self-Test (BIST) mechanism. **Note:** This mechanism is not supported in the Core Logic module in the SC1100.	
7	**BIST Capable.** Indicates if the device can run a Built-In Self-Test (BIST). 0: The device has no BIST functionality. 1: The device can run a BIST.
6	**Start BIST.** Setting this bit to 1 starts up a BIST on the device. The device resets this bit when the BIST is completed. (Not supported.)
5:4	**Reserved.**
3:0	**BIST Completion Code.** Upon completion of the BIST, the completion code is stored in these bits. A completion code of 0000 indicates that the BIST was successfully completed. Any other value indicates a BIST failure.
Index 10h-13h	**Base Address Register 0 - F0BAR0 (R/W)** — **Reset Value: 00000001h**
This register allows access to I/O mapped GPIO runtime and configuration Registers. Bits [5:0] are read only (000001), indicating a 64-byte aligned I/O address space. Refer to Table 5-30 on page 196 for the GPIO register bit formats and reset values.	
31:6	**GPIO Base Address.**
5:0	**Address Range. (Read Only)**
Index 14h-17h	**Base Address Register 1 - F0BAR1 (R/W)** — **Reset Value: 00000001h**
This register allows access to I/O mapped LPC configuration registers. Bits [5:0] are read only (000001), indicating a 64-byte aligned I/O address space. Refer to Table 5-31 on page 200 for the bit formats and reset values of the LPC registers.	
31:6	**LPC Base Address.**
5:0	**Address Range. (Read Only)**
Index 18h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0500h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 40h	**PCI Function Control Register 1 (R/W)** — **Reset Value: 39h**
7:6	**Reserved.** Must be set to 0.
5	**Reserved.** Must be set to 0.
4	**Reserved.** Must be set to 1.
3	**Reserved**. Must be set to 1.
2	**Reserved**. Must be set to 0.
1	**PERR# Signals SERR#.** Assert SERR# when PERR# is asserted or detected as active by the Core Logic module (allows PERR# assertion to be cascaded to NMI (SMI) generation in the system). 0: Disable. 1: Enable.
0	**PCI Interrupt Acknowledge Cycle Response.** The Core Logic module responds to PCI interrupt acknowledge cycles. 0: Disable. 1: Enable.
Index 41h	**PCI Function Control Register 2 (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**X-Bus Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 5 (F5) register space, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
4	**Reserved**. Must be set to 0.
3	**XpressAUDIO Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 3 (F3) register space, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
2	**IDE Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 2 (F2) register space, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
1	**Power Management Configuration Trap.** If this bit is set to 1 and an access occurs to one of the configuration registers in PCI Function 1 (F1) register space, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
0	**Legacy Configuration SMI.** If this bit is set to 1 and an access occurs to one of the configuration registers in the ISA Legacy I/O register space, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[5].
Index 42h	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 43h	**Delayed Transactions Register (R/W)** — **Reset Value: 02h**
7:5	**Reserved.** Must be set to 0.
4	**Enable PCI Delayed Transactions for Access to I/O Address 170h-177h (Secondary IDE Channel).** PIO mode uses repeated I/O transactions that are faster when non-delayed transactions are used. 0: I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to Secondary IDE channel I/O addresses are delayed transactions on PCI.
3	**Enable PCI Delayed Transactions for Access to I/O Address 1F0h-1F7h (Primary IDE Channel).** PIO mode uses repeated I/O transactions that are faster when non-delayed transactions are used. 0: I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to Primary IDE channel I/O addresses are delayed transactions on PCI.
2	**Enable PCI Delayed Transactions for AT Legacy PIC I/O Addresses.** Some PIC status reads are long. Enabling delayed transactions help reduce DMA latency for high bandwidth devices like VIP. 0: PIC I/O addresses complete as fast as possible on PCI. (Default) 1: Accesses to PIC I/O addresses are delayed transactions on PCI.
1	**Enable PCI Delayed Transactions for AT Legacy PIT I/O Addresses.** Some x86 programs (certain benchmarks/diagnostics) assume a particular latency for PIT accesses; this bit allows that code to work. 0: PIT I/O addresses complete as fast as possible on PCI. 1: Accesses to PIT I/O addresses are delayed transactions on PCI. (Default) For best performance (e.g., when running Microsoft Windows), this bit should be set to 0.
0	**Reserved.** Must be set to 0.
Index 44h	**Reset Control Register (R/W)** — **Reset Value: 01h**
7	**AC97 Soft Reset.** Active high reset for the AC97 codec interface. 0: AC97_RST# (ball AF23) is driven high. (Default) 1: AC97_RST# (ball AF23) is driven low.
6:4	**Reserved.** Must be set to 0.
3	**IDE Controller Reset.** Reset the IDE controller. 0: Disable. 1: Enable. Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled.
2	**IDE Reset.** Reset IDE bus. 0: Disable. 1: Enable (drives outputs to zero). Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled.
1	**PCI Reset.** Reset PCI bus. 0: Disable. 1: Enable. When this bit is set to 1, the Core Logic module output signal PCIRST# is asserted and all devices on the PCI bus (including PCIUSB) are reset. No other function within the Core Logic module is affected by this bit. Write 0 to clear this bit. This bit is level-sensitive and must be cleared after the reset is enabled.
0	**X-Bus Warm Start.** Writing and reading this bit each have different meanings. When reading this bit, it indicates whether or not a warm start occurred since power-up: 0: A warm start occurred. 1: No warm start has occurred. When writing this bit, it can be used to trigger a system-wide reset: 0: No effect. 1: Execute system-wide reset (used only for clock configuration at power-up).
Index 45h	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 46h	**PCI Functions Enable Register (R/W)** — **Reset Value: FEh**
7:6	**Reserved.** Resets to 11.
5	**F5 (PCI Function 5).** When asserted (set to 1), enables the register space designated as F5. This bit must always be set to 1. (Default)
4	**Reserved.** Must be set to 0.
3	**F3 (PCI Function 3).** When asserted (set to 1), enables the register space designated as F3. This bit must always be set to 1. (Default)
2	**F2 (PCI Function 2).** When asserted (set to 1), enables the register space designated as F2. This bit must always be set to 1. (Default)
1	**F1 (PCI Function 1).** When asserted (set to 1), enables the register space designated as F1. This bit must always be set to 1. (Default)
0	**Reserved.** Must be set to 0.
Index 47h	**Miscellaneous Enable Register (R/W)** — **Reset Value: 00h**
7:3	**Reserved.** Must be set to 0.
2	**F0BAR1 (PCI Function 0, Base Address Register 1).** F0BAR1, pointer to I/O mapped LPC configuration registers. 0: Disable. 1: Enable.
1	**F0BAR0 (PCI Function 0, Base Address Register 0).** F0BAR0, pointer to I/O mapped GPIO configuration registers. 0: Disable. 1: Enable.
0	**Reserved.** Must be set to 0.
Index 48h-4Bh	**Reserved** — **Reset Value: 00h**
Index 4Ch-4Fh	**Top of System Memory (R/W)** — **Reset Value: FFFFFFFFh**
31:0	**Top of System Memory.** Highest address in system used to determine active decode for external PCI mastered memory cycles. If an external PCI master requests a memory address below the value programmed in this register, the cycle is transferred from the external PCI bus interface to the Fast-PCI interface for servicing by the GX1 module. **Note:** The 4 least significant bits must be set to 1.
Index 50h	**PIT Control/ISA CLK Divider (R/W)** — **Reset Value: 7Bh**
7	**PIT Software Reset.** 0: Disable. 1: Enable.
6	**PIT Counter 1.** 0: Forces Counter 1 output (OUT1) to zero. 1: Allows Counter 1 output (OUT1) to pass to the Port 061h[4].
5	**PIT Counter 1 Enable.** 0: Sets GATE1 input low. 1: Sets GATE1 input high.
4	**PIT Counter 0.** 0: Forces Counter 0 output (OUT0) to zero. 1: Allows Counter 0 output (OUT0) to pass to IRQ0.
3	**PIT Counter 0 Enable.** 0: Sets GATE0 input low. 1: Sets GATE0 input high.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2:0	**ISA Clock Divisor.** Determines the divisor of the PCI clock used to make the ISA clock, which is typically programmed for approximately 8 MHz: 000: Divide by 1 — 100: Divide by 5 001: Divide by 2 — 101: Divide by 6 010: Divide by 3 — 110: Divide by 7 011: Divide by 4 — 111: Divide by 8 If PCI clock = 25 MHz, use setting of 010 (divide by 3). If PCI clock = 30 or 33 MHz, use a setting of 011 (divide by 4).
Index 51h	**ISA I/O Recovery Control Register (R/W)** — **Reset Value: 40h**
7:4	**8-Bit I/O Recovery.** These bits determine the number of ISA bus clocks between back-to-back 8-bit I/O read cycles. This count is in addition to a preset one-clock delay built into the controller. 0000: 1 PCI clock 0001: 2 PCI clocks ::: ::: ::: 1111: 16 PCI clocks
3:0	**16-Bit I/O Recovery.** These bits determine the number of ISA bus clocks between back-to-back 16-bit I/O cycles. This count is in addition to a preset one-clock delay built into the controller. 0000: 1 PCI clock 0001: 2 PCI clocks ::: ::: ::: 1111: 16 PCI clocks
Index 52h	**ROM/AT Logic Control Register (R/W)** — **Reset Value: 98h**
7	**Snoop Fast Keyboard Gate A20 and Fast Reset.** Enables the snoop logic associated with keyboard commands for A20 Mask and Reset. 0: Disable snooping. The keyboard controller handles the commands. 1: Enable snooping.
6:5	**Reserved**.
4	**Enable A20M# Deassertion on Warm Reset.** Force A20M# high during a Warm Reset (guarantees that A20M# is deasserted regardless of the state of A20). 0: Disable. 1: Enable.
3	**Enable Port 092h (Port A).** Port 092h decode and the logical functions. 0: Disable. 1: Enable.
2	**Upper ROM Size.** Selects upper ROM addressing size. 0: 256K (FFFC0000h-FFFFFFFFh). 1: Use ROM Mask register (F0 Index 6Eh). ROMCS# goes active for the above ranges whether strapped for ISA or LPC. (Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.) The selected range can then be either positively or subtractively decoded through F0 Index 5Bh[5].
1	**ROM Write Enable**. When asserted, enables writes to ROM space, allowing Flash programming. If strapped for ISA and this bit is set to 1, writes to the configured ROM space asserts ROMCS#, enabling the write cycle to the Flash device on the ISA bus. Otherwise, ROMCS# is inhibited for writes. If strapped for LPC and this bit is set to 1, the cycle runs on the LPC bus. Otherwise, the LPC bus cycle is inhibited for writes. Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
0	**Lower ROM Size.** Selects lower ROM addressing size in which ROMCS# goes active. 0: Lower ROM access are 000F0000h-000FFFFFh (64 KB) (Default). 1: Lower ROM accesses are 000E0000h-000FFFFFh (128 KB). ROMCS# goes active for the above ranges whether strapped for ISA or LPC. (Refer to F0BAR1+I/O Offset 10h[15] for further strapping/programming details.) The selected range can then be either positively or subtractively decoded through F0 Index 5Bh[5].
Index 53h	**Alternate CPU Support Register (R/W)** — **Reset Value: 00h**
7	**Reserved.** Must be set to 0.
6	**Reserved.** Must be set to 0.
5	**Bidirectional SMI Enable.** 0: Disable. 1: Enable. This bit must be set to 0.
4:2	**Reserved.** Must be set to 0.
1	**IRQ13 Function Selection.** Selects function of the internal IRQ13/FERR# signal. 0: FERR#. 1: IRQ13. This bit must be set to 1.
0	**Generate SMI on A20M# Toggle.** 0: Disable. 1: Enable. This bit must be set to 1. SMI status is reported at F1BAR0+I/O Offset 00h/02h[7].
Index 54h-59h	**Reserved** — **Reset Value: 00h**
Index 5Ah	**Decode Control Register 1 (R/W)** — **Reset Value: 01h**
Indicates PCI positive or negative decoding for various I/O ports on the ISA bus. **Note:** Positive decoding by the Core Logic module speeds up I/O cycle time. The I/O ports mentioned in the bit descriptions below, do not exist in the Core Logic module. It is assumed that if positive decode is enabled for a port, the port exists on the ISA bus.	
7	**Secondary Floppy Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 372h-375h and 377h. 0: Subtractive. 1: Positive.
6	**Primary Floppy Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3F2h-3F5h and 3F7h. 0: Subtractive. 1: Positive.
5	**COM4 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 2E8h-2EFh. 0: Subtractive. 1: Positive.
4	**COM3 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3E8h-3EFh. 0: Subtractive. 1: Positive.
3	**COM2 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 2F8h-2FFh. 0: Subtractive. 1: Positive.
2	**COM1 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3F8h-3FFh. 0: Subtractive. 1: Positive.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**Keyboard Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O Ports 060h and 064h (as well as 062h and 066h, if enabled - F4 Index 5Bh[7] = 1). 0: Subtractive. 1: Positive. **Note:** If F0BAR1+I/O Offset 10h bits 10 = 0 and 16 = 1, then this bit must be written 0.
0	**Real -Time Clock Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O Ports 070h-073h. 0: Subtractive. 1: Positive.

Index 5Bh	Decode Control Register 2 (R/W)	Reset Value: 20h

Note: Positive decoding by the Core Logic module speeds up the I/O cycle time. The Keyboard, LPT3, LPT2, and LPT1 I/O ports do not exist in the Core Logic module. It is assumed that if positive decoding is enabled for any of these ports, the port exists on the ISA bus.

7	**Keyboard I/O Port 062h/066h Positive Decode.** This alternate port to the keyboard controller is provided in support of power management features. 0: Disable. 1: Enable.
6	**Reserved.** Must be set to 0.
5	**BIOS ROM Positive Decode.** Selects PCI positive or subtractive decoding for accesses to the configured ROM space. 0: Subtractive. 1: Positive. ROM configuration is at F0 Index 52h[2:0].
4	**Secondary IDE Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 170h-177h and 376h-377h (excluding writes to 377h). 0: Subtractive. Subtractively decoded IDE addresses are forwarded to the PCI slot bus. If a master abort occurs, they are then forwarded to ISA. 1: Positive. Positively decoded IDE addresses are forwarded to the internal IDE controller and then to the IDE bus.
3	**Primary IDE Controller Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 1F0h-1F7h and 3F6h-3F7h (excluding writes to 3F7h). 0: Subtractive. Subtractively decoded IDE addresses are forwarded to the PCI slot bus. If a master abort occurs, they are then forwarded to ISA. 1: Positive. Positively decoded IDE addresses are forwarded to the internal IDE controller and then to the IDE bus.
2	**LPT3 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 278h-27Fh. 0: Subtractive. 1: Positive.
1	**LPT2 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 378h-37Fh. 0: Subtractive. 1: Positive.
0	**LPT1 Positive Decode.** Selects PCI positive or subtractive decoding for accesses to I/O ports 3BCh-3BFh 0: Subtractive. 1: Positive.

Index 5Ch	PCI Interrupt Steering Register 1 (R/W)	Reset Value: 00h

Indicates target interrupts for signals INTB# (ball W24) and INTA# (ball AD26).

Note: The target interrupt must first be configured as level sensitive via I/O Ports 4D0h and 4D1h in order to maintain PCI interrupt compatibility.

7:4	**INTB# (Ball W24) Target Interrupt.**			
	0000: Disable	0100: IRQ4	1000: Reserved	1100: IRQ12
	0001: IRQ1	0101: IRQ5	1001: IRQ9	1101: Reserved
	0010: Reserved	0110: IRQ6	1010: IRQ10	1110: IRQ14
	0011: IRQ3	0111: IRQ7	1011: IRQ11	1111: IRQ15

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
3:0	**INTA# (Ball AD26) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15

Index 5Dh — PCI Interrupt Steering Register 2 (R/W) — Reset Value: 00h

Indicates target interrupts for signals INTD# (ball V24) and INTC# (ball Y24). Note that INTC# is muxed with GPIO19 (selection made via PMR[4], see Table 3-2 on page 50 for PMR[4] bit description).

Note: The target interrupt must first be configured as level sensitive via I/O Ports 4D0h and 4D1h in order to maintain PCI interrupt compatibility.

7:4	**INTD# (Ball V24) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15
3:0	**INTC (Ball Y24) Target Interrupt.** 0000: Disable 0100: IRQ4 1000: Reserved 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: Reserved 0010: Reserved 0110: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0111: IRQ7 1011: IRQ11 1111: IRQ15

Index 5Eh-5Fh — Reserved — Reset Value: 00h

Index 60h-63h — ACPI Control Register (R/W) — Reset Value: 00000000h

31:8	**Reserved.** Must be set to 0.
7	**Internal SUSP_3V Shut Down PLL5.** Allow internal SUSP_3V to shut down PLL5. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
6	**SUSP_3V Shut Down PLL4.** Allow internal SUSP_3V to shut down PLL4. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
5	**SUSP_3V Shut Down PLL3.** Allow internal SUSP_3V to shut down PLL3. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
4	**Reserved.**
3	**SUSP_3V Shut Down FMUL3.** Allow internal SUSP_3V to shut down FMUL3. 0: Clock generator is stopped when internal SUSP_3V is active. 1: Clock generator continues working when internal SUSP_3V is active.
2	**ACPI C3 SUSP_3V Enable.** Allow internal SUSP_3V to be active during C3 state. 0: Disable. 1: Enable.
1	**ACPI SL1 SUSP_3V Enable.** Allow internal SUSP_3V to be active during SL1 sleep state. 0: Disable. 1: Enable.
0	**ACPI C3 Support Enable.** Allow support of C3 states. 0: Disable. 1: Enable.

Index 64h-6Dh — Reserved — Reset Value: 00h

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 6Eh-6Fh	**ROM Mask Register (R/W)** — **Reset Value: FFF0h**
15:8	**Reserved.** Must be set to FFh.
7:4	**ROM Size.** If F0 Index 52h[2] = 1: 0000: 16 MB = FF000000h-FFFFFFFFh 1000: 8 MB = FF800000h-FFFFFFFFh 1100: 4 MB = FFC00000h-FFFFFFFFh 1110: 2 MB = FFE00000h-FFFFFFFFh 1111: 1 MB = FFF00000h-FFFFFFFFh All other settings for these bits are reserved.
3:0	**Reserved.** Must be set to 0.
Index 70h-71h	**IOCS1# Base Address Register (R/W)** — **Reset Value: 0000h**
15:0	**I/O Chip Select 1 Base Address.** This 16-bit value represents the I/O base address used to enable assertion of IOCS1# (ball D22, muxed with GPIO14+IOR#+DOCR# see PMR[21,2] in Table 3-2 on page 50 for ball function selection). This register is used in conjunction with F0 Index 72h (IOCS1# Control register).
Index 72h	**IOCS1# Control Register (R/W)** — **Reset Value: 00h** This register is used in conjunction with F0 Index 70h (IOCS1# Base Address register).
7	**I/O Chip Select 1 Positive Decode (IOCS1#).** 0: Disable. 1: Enable.
6	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured I/O address (base address configured in F0 Index 70h; range configured in bits [4:0]) cause IOCS1# to be asserted. 0: Disable. 1: Enable.
5	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured I/O address (base address configured in F0 Index 70h; range configured in bits [4:0]) cause IOCS1# to be asserted. 0: Disable. 1: Enable.
4:0	**IOCS1# I/O Address Range.** This 5-bit field is used to select the range of IOCS1#. 00000: 1 Byte 00001: 2 Bytes 00011: 4 Bytes 00111: 8 Bytes 01111: 16 Bytes 11111: 32 Bytes All other combinations are reserved.
Index 73h	**Reserved** — **Reset Value: 00h**
Index 74h-75h	**IOCS0# Base Address Register (R/W)** — **Reset Value: 0000h**
15:0	**I/O Chip Select 0 Base Address.** This 16-bit value represents the I/O base address used to enable the assertion of IOCS0# (ball B23, muxed with GPIO17+IOCHRDY, see PMR[9,5] in Table 3-2 on page 50 for ball function selection). This register is used in conjunction with F0 Index 76h (IOCS0# Control register).
Index 76h	**IOCS0# Control Register (R/W)** — **Reset Value: 00h** This register is used in conjunction with F0 Index 74h (IOCS0# Base Address register).
7	**I/O Chip Select 0 Positive Decode (IOCS0#).** 0: Disable. 1: Enable.
6	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured I/O address (base address configured in F0 Index 74h; range configured in bits [4:0]) cause IOCS0# to be asserted. 0: Disable. 1: Enable.
5	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured I/O address (base address configured in F0 Index 74h; range configured in bits [4:0]) cause IOCS0# to be asserted. 0: Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4:0	**IOCS0# I/O Address Range.** This 5-bit field is used to select the range of IOCS0#. 00000: 1 Byte 00001: 2 Bytes 00011: 4 Bytes 00111: 8 Bytes 01111: 16 Bytes 11111: 32 Bytes All other combinations are reserved.
Index 77h	**Reserved** — **Reset Value: 00h**
Index 78h-7Bh	**DOCCS# Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**DiskOnChip Chip Select Base Address.** This 32-bit value represents the memory base address used to enable assertion of DOCCS# (ball D21, muxed with GPIO20, see PMR[7] in Table 3-2 on page 50 for ball function selection). This register is used in conjunction with F0 Index 7Ch (DOCCS# Control register).
Index 7Ch-7Fh	**DOCCS# Control Register (R/W)** — **Reset Value: 00000000h** This register is used in conjunction with F0 Index 78h (DOCCS# Base Address register).
31:27	**Reserved.** Must be set to 0.
26	**DiskOnChip Chip Select Positive Decode (DOCCS#).** 0: Disable. 1: Enable.
25	**Writes Result in Chip Select.** When this bit is set to 1, writes to configured memory address (base address configured in F0 Index 78h; range configured in bits [18:0]) cause DOCCS# to be asserted. 0: Disable. 1: Enable.
24	**Reads Result in Chip Select.** When this bit is set to 1, reads from configured memory address (base address configured in F0 Index 78h; range configured in bits [18:0]) cause DOCCS# to be asserted. 0: Disable. 1: Enable.
23:19	**Reserved.** Must be set to 0.
18:0	**DOCCS# Memory Address Range.** This 19-bit mask is used to qualify accesses on which DOCCS# is asserted by masking the upper 19 bits of the incoming PCI address (AD[31:13]).
Index 80h	**Power Management Enable Register 1 (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**Codec SDATA_IN SMI.** When set to 1, this bit allows an SMI to be generated in response to an AC97 codec producing a positive edge on SDATA_IN (ball AF22). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[2].
4	**Reserved.** Must be set to 0.
3	**IRQ Speedup.** Any unmasked IRQ (per I/O Ports 021h/0A1h) or SMI disables clock throttling (via internal SUSP#/SUSPA# handshake) for a configurable duration when system is power-managed using CPU Suspend modulation. 0: Disable. 1: Enable. The duration of the speedup is configured in the IRQ Speedup Timer Count Register (F0 Index 8Ch).
2	**Traps.** Globally enable all power management I/O traps. 0: Disable. 1: Enable. This excludes the XpressAUDIO I/O traps, which are enabled via F3BAR0+Memory Offset 18h.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**Idle Timers.** Device idle timers. 0: Disable. 1: Enable. **Note:** Disable at this level does not reload the timers on the enable. The timers are disabled at their current counts. This bit has no affect on the Suspend Modulation Counters (F0 Index 94h). Only applicable when in APM mode (F1BAR1+I/O Offset 0Ch[0] = 0) and not ACPI mode.
0	**Power Management.** Global power management. 0: Disable. 1: Enable. This bit must be set to 1 immediately after POST for power management resources to function.
Index 81h	**Power Management Enable Register 2 (R/W)** — **Reset Value: 00h**
7	**Reserved.** Must be set to 0.
6	**User Defined Device 3 (UDEF3) Idle Timer Enable.** Turn on UDEF3 Idle Timer Count Register (F0 Index A4h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF3 address programming is at F0 Index C8h (base address register) and CEh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[6].
5	**User Defined Device 2 (UDEF2) Idle Timer Enable.** Turn on UDEF2 Idle Timer Count Register (F0 Index A2h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF2 address programming is at F0 Index C4h (base address register) and CDh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[5].
4	**User Defined Device 1 (UDEF1) Idle Timer Enable.** Turn on UDEF1 Idle Timer Count Register (F0 Index A0h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the programmed address range, the timer is reloaded with the programmed count. UDEF1 address programming is at F0 Index C0h (base address register) and CCh (control register). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[4].
3	**Keyboard/Mouse Idle Timer Enable.** Turn on Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges listed below, the timer is reloaded with the programmed count: — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[3].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Parallel/Serial Idle Timer Enable.** Turn on Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges listed below, the timer is reloaded with the programmed count. — LPT3: I/O Port 278h-27Fh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[2].
1	**Floppy Disk Idle Timer Enable.** Turn on Floppy Disk Idle Timer Count Register (F0 Index 9Ah) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges listed below, the timer is reloaded with the programmed count. — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h — Secondary floppy disk: I/O Port 372h-375h, 377h Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[1].
0	**Primary Hard Disk Idle Timer Enable.** Turn on Primary Hard Disk Idle Timer Count Register (F0 Index 98h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges selected in F0 Index 93h[5], the timer is reloaded with the programmed count. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[0].
Index 82h	**Power Management Enable Register 3 (R/W)** **Reset Value: 00h**
7	**Reserved.** Must be set to 0.
6	**User Defined Device 3 (UDEF3) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF3 address programming is at F0 Index C8h (Base Address register) and CEh (Control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[4].
5	**User Defined Device 2 (UDEF2) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF2 address programming is at F0 Index C4h (Base Address register) and CDh (Control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[3].
4	**User Defined Device 1 (UDEF1) Access Trap.** If this bit is enabled and an access occurs in the programmed address range, an SMI is generated. UDEF1 address programming is at F0 Index C0h (base address register), and CCh (control register). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[2].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
3	**Keyboard/Mouse Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included). Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[3].
2	**Parallel/Serial Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — LPT3: I/O Port 3BCh-3BEh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[2].
1	**Floppy Disk Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges listed below, an SMI is generated. — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h. — Secondary floppy disk: I/O Port 372h-375h, 377h. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[1].
0	**Primary Hard Disk Access Trap.** 0: Disable. 1: Enable. If this bit is enabled and an access occurs in the address ranges selected in F0 Index 93h[5], an SMI is generated. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[0].
Index 83h	**Power Management Enable Register 4 (R/W)** — **Reset Value: 00h**
7	**Secondary Hard Disk Idle Timer Enable.** Turn on Secondary Hard Disk Idle Timer Count Register (F0 Index ACh) and generate an SMI when the timer expires. 0: Disable. 1: Enable. If an access occurs in the address ranges selected in F0 Index 93h[4], the timer is reloaded with the programmed count. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[4].
6	**Secondary Hard Disk Access Trap.** If this bit is enabled and an access occurs in the address ranges selected in F0 Index 93h[4], an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[5].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
5	**ACPI Timer SMI.** Allow SMI generation for MSB toggles on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[0].
4	**THRM# SMI.** Allow SMI generation on assertion of THRM# (ball AE15). 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 87h/F7h[6].
3	**Reserved.** Must be set to 0.
2	**Reserved.** Must be set to 0.
1	**General Purpose Timer 2 Enable.** Turn on GP Timer 2 Count Register (F0 Index 8Ah) and generate an SMI when the timer expires. 0: Disable. 1: Enable. This idle timer is reloaded from the assertion of GPIO7 (if programmed to do so). GP Timer 2 programming is at F0 Index 8Bh[5,3,2]. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[1].
0	**General Purpose Timer 1 Enable.** Turn on GP Timer 1 Count Register (F0 Index 88h) and generate an SMI when the timer expires. 0: Disable. 1: Enable. This idle timer's load is multi-sourced and gets reloaded any time an enabled event (F0 Index 89h[6:0]) occurs. GP Timer 1 programming is at F0 Index 8Bh[4]. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. Second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[0].

Index 84h — Second Level PME/SMI Status Mirror Register 1 (RO) — Reset Value: 00h

The bits in this register are used for the second level of status reporting. The top level is reported at F1BAR0+I/O Offset 00h/02h[0].

This register is called a "mirror" register since an identical register exists at F0 Index F4h. Reading this register does not clear the status, while reading its counterpart at F0 Index F4h clears the status at both the second and the top levels.

7:3	**Reserved.** Reads as 0.
2	**GPWIO2 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO2 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO2 is enabled as an input: F1BAR1+I/O Offset 15h[2] = 0. 2) Set F1BAR1+I/O Offset 15h[6] to 1.
1	**GPWIO1 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO1 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO1 is enabled as an input: F1BAR1+I/O Offset 15h[1] = 0. 2) Set F1BAR1+I/O Offset 15h[5] to 1.
0	**GPWIO0 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO0 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO0 is enabled as an input: F1BAR1+I/O Offset 15h[0] = 0. 2) Set F1BAR1+I/O Offset 15h[4] to 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 85h	**Second Level PME/SMI Status Mirror Register 2 (RO)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. This register is called a “Mirror” register since an identical register exists at F0 Index F5h. Reading this register does not clear the status, while reading its counterpart at F0 Index F5h clears the status at both the second and top levels.	
7	**Reserved.** Reads as 0.
6	**User Defined Device Idle Timer 3 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 3 Idle Timer Count Register (F0 Index A4h). 0: No 1: Yes To enable SMI generation, set F0 Index 81h[6] to 1.
5	**User Defined Device Idle Timer 2 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 2 Idle Timer Count Register (F0 Index A2h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[5] to 1.
4	**User Defined Device Idle Timer 1 Timeout.** Indicates whether or not an SMI was caused by expiration of User Defined Device 1 Idle Timer Count Register (F0 Index A0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[4] to 1.
3	**Keyboard/Mouse Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[3] to 1.
2	**Parallel/Serial Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[2] to 1.
1	**Floppy Disk Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Floppy Disk Idle Timer Count Register (F0 Index 9Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[1] to 1.
0	**Primary Hard Disk Idle Timer Timeout.** Indicates whether or not an SMI was caused by expiration of Primary Hard Disk Idle Timer Count Register (F0 Index 98h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[0] to 1.
Index 86h	**Second Level PME/SMI Status Mirror Register 3 (RO)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. This register is called a “Mirror” register since an identical register exists at F0 Index F6h. Reading this register does not clear the status, while reading its counterpart at F0 Index F6h clears the status at both the second and top levels.	
7	**Reserved.** Reads as 0.
6	**Reserved.** Reads as 0.
5	**Secondary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the secondary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[6] to 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
4	**Secondary Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Secondary Hard Disk Idle Timer Count register (F0 Index ACh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[7] to 1.
3	**Keyboard/Mouse Access Trap SMI Status.** Indicates whether or not an SMI was caused by an trapped I/O access to the keyboard or mouse. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[3] to 1.
2	**Parallel/Serial Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to either the serial or parallel ports. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[2] to 1.
1	**Floppy Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the floppy disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[1] to 1.
0	**Primary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the primary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[0] to 1.
Index 87h	**Second Level PME/SMI Status Mirror Register 4 (RO)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported at F1BAR0+I/O Offset 00h/02h[0]. This register is called a “Mirror” register since an identical register exists at F0 Index F7h. Reading this register does not clear the status, while reading its counterpart at F0 Index F7h clears the status at both the second and top levels except for bit 7 which has a third level of SMI status reporting at F0BAR0+I/O 0Ch/1Ch.	
7	**GPIO Event SMI Status.** Indicates whether or not an SMI was caused by a transition of any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] to 0. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME and setting F1BAR1+I/O Offset 0Ch[0] = 0 enables the PME to generate an SMI. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h). The next level (third level) of SMI status is at F0BAR0+I/O 0Ch/1Ch[15:0].
6	**Thermal Override SMI Status.** Indicates whether or not an SMI was caused by the assertion of THRM# (ball AE15). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[4] to 1.
5:4	**Reserved.** Reads as 0.
3	**SIO PWUREQ SMI Status.** Indicates whether or not an SMI was caused by a power-up event from the SIO. 0: No. 1: Yes. A power-up event is defined as any of the following events/activities: — RI# — IRRX1 (CEIR) To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] to 0.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Codec SDATA_IN SMI Status.** Indicates whether or not an SMI was caused by AC97 codec producing a positive edge on SDATA_IN. 0: No. 1: Yes. To enable SMI generation, set F0 Index 80h[5] to 1.
1	**RTC Alarm (IRQ8#) SMI Status.** Indicates whether or not an SMI was caused by an RTC interrupt. 0: No. 1: Yes. This SMI event can only occur while in 3V Suspend and an RTC interrupt occurs with F1BAR1+I/O Offset 0Ch[0] = 0.
0	**ACPI Timer SMI Status.** Indicates whether or not an SMI was caused by an ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch) MSB toggle. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[5] to 1.
Index 88h	**General Purpose Timer 1 Count Register (R/W)** **Reset Value: 00h**
7:0	**GPT1_COUNT.** This field represents the load value for General Purpose Timer 1. This value can represent either an 8-bit counter or a 16-bit counter (selected in F0 Index 8Bh[4]). It is loaded into the counter when the timer is enabled (F0 Index 83h[0] = 1). Once enabled, an enabled event (configured in F0 Index 89h[6:0]) reloads the timer. The counter is decremented with each clock of the configured timebase (1 msec or 1 sec selected at F0 Index 89h[7]). Upon expiration of the counter, an SMI is generated, and the top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[9]. The second level SMI status is reported at F1BAR0+I/O Offset 04h/06h[0]. Once expired, this counter must be re-initialized by either disabling and enabling it, or writing a new count value in this register.
Index 89h	**General Purpose Timer 1 Control Register (R/W)** **Reset Value: 00h**
7	**General Purpose Timer 1 Timebase.** Selects timebase for General Purpose Timer 1 (F0 Index 88h). 0: 1 second. 1: 1 millisecond.
6	**Re-trigger General Purpose Timer 1 on User Defined Device 3 (UDEF3) Activity.** 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF3 (configured in F0 Index C8h and CEh) reloads General Purpose Timer 1.
5	**Re-trigger General Purpose Timer 1 on User Defined Device 2 (UDEF2) Activity.** 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF2 (configured in F0 Index C4h and CDh) reloads General Purpose Timer 1.
4	**Re-trigger General Purpose Timer 1 on User Defined Device 1 (UDEF1) Activity**. 0: Disable. 1: Enable. Any access to the configured (memory or I/O) address range for UDEF1 (configured in F0 Index C0h and CCh) reloads General Purpose Timer 1.
3	**Re-trigger General Purpose Timer 1 on Keyboard or Mouse Activity.** 0: Disable. 1: Enable. Any access to the keyboard or mouse I/O address range listed below reloads General Purpose Timer 1: — Keyboard Controller: I/O Ports 060h/064h. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is included). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is included).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Re-trigger General Purpose Timer 1 on Parallel/Serial Port Activity.** 0: Disable. 1: Enable. Any access to the parallel or serial port I/O address range listed below reloads the General Purpose Timer 1: — LPT3: I/O Port 3BCh-3BEh. — LPT2: I/O Port 378h-37Fh. — COM1: I/O Port 3F8h-3FFh (if F0 Index 93h[1:0] = 10 this range is excluded). — COM2: I/O Port 2F8h-2FFh (if F0 Index 93h[1:0] = 11 this range is excluded). — COM3: I/O Port 3E8h-3EFh. — COM4: I/O Port 2E8h-2EFh.
1	**Re-trigger General Purpose Timer 1 on Floppy Disk Activity.** 0: Disable. 1: Enable. Any access to the floppy disk drive address ranges listed below reloads General Purpose Timer 1: — Primary floppy disk: I/O Port 3F2h-3F5h, 3F7h — Secondary floppy disk: I/O Port 372h-375h, 377h The active floppy disk drive is configured via F0 Index 93h[7].
0	**Re-trigger General Purpose Timer 1 on Primary Hard Disk Activity.** 0: Disable. 1: Enable. Any access to the primary hard disk address range selected in F0 Index 93h[5], reloads General Purpose Timer 1.

Index 8Ah	General Purpose Timer 2 Count Register (R/W)	Reset Value: 00h
7:0	**GPT2_COUNT.** This field represents the load value for General Purpose Timer 2. This value can represent either an 8-bit or 16-bit counter (configured in F0 Index 8Bh[5]). It is loaded into the counter when the timer is enabled (F0 Index 83h[1] = 1). Once the timer is enabled and a transition occurs on GPIO7, the timer is re-loaded. The counter is decremented with each clock of the configured timebase (1 msec or 1 sec selected at F0 Index 8Bh[3]). Upon expiration of the counter, an SMI is generated and the top level of status is F1BAR0+I/O Offset 00h/02h[9]. The second level of status is reported at F1BAR0+I/O Offset 04h/06h[1]). Once expired, this counter must be re-initialized by either disabling and enabling it, or by writing a new count value in this register. For GPIO7 to act as the reload for this counter, it must be enabled as such (F0 Index 8Bh[2]) and be configured as an input. (GPIO pin programming is at F0BAR0+I/O Offset 20h and 24h.)	

Index 8Bh	General Purpose Timer 2 Control Register (R/W)	Reset Value: 00h
7	**Re-trigger General Purpose Timer 1 (GP Timer 1) on Secondary Hard Disk Activity.** 0: Disable. 1: Enable. Any access to the secondary hard disk address range selected in F0 Index 93h[4] reloads GP Timer 1.	
6	**Reserved.** Should be set to 0.	
5	**General Purpose Timer 2 (GP Timer 2) Shift.** GP Timer 2 is treated as an 8-bit or 16-bit timer. 0: 8-bit. The count value is loaded into GP Timer 2 Count Register (F0 Index 8Ah). 1: 16-bit. The value loaded into GP Timer 2 Count Register is shifted left by eight bits, the lower eight bits become zero, and this 16-bit value is used as the count for GP Timer 2.	
4	**General Purpose Timer 1 (GP Timer 1) Shift.** GP Timer 1 is treated as an 8-bit or 16-bit timer. 0: 8-bit. The count value is that loaded into GP Timer 1 Count Register (F0 Index 88h). 1: 16-bit. The value loaded into GP Timer 1 Count Register is shifted left by eight bit, the lower eight bits become zero, and this 16-bit value is used as the count for GP Timer 1.	
3	**General Purpose Timer 2 (GP Timer 2) Timebase.** Selects timebase for GP Timer 2 (F0 Index 8Ah). 0: 1 second. 1: 1 millisecond.	

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
2	**Re-trigger Timer on GPIO7 Pin Transition.** A rising-edge transition on the GPIO7 pin reloads GP Timer 2 (F0 Index 8Ah). 0: Disable. 1: Enable. For GPIO7 to work here, it must first be configured as an input. (GPIO pin programming is at F0BAR0+I/O Offset 20h and 24h.)
1:0	**Reserved.** Set to 0.
Index 8Ch	**IRQ Speedup Timer Count Register (R/W)** — **Reset Value: 00h**
7:0	**IRQ Speedup Timer Load Value.** This field represents the load value for the IRQ speedup timer. It is loaded into the counter when Suspend Modulation is enabled (F0 Index 96h[0] = 1) and an INTR or an access to I/O Port 061h occurs. When the event occurs, the Suspend Modulation logic is inhibited, permitting full performance operation of the GX1 module. Upon expiration, no SMI is generated; the Suspend Modulation begins again. The IRQ speedup timer's timebase is 1 msec. This speedup mechanism allows instantaneous response to system interrupts for full-speed interrupt processing. A typical value here would be 2 to 4 msec.
Index 8Dh-92h	**Reserved** — **Reset Value: 00h**
Index 93h	**Miscellaneous Device Control Register (R/W)** — **Reset Value: 00h**
7	**Floppy Drive Port Select.** Indicates whether all system resources used to power manage the floppy drive use the primary, or secondary FDC addresses for decode. 0: Secondary. 1: Primary.
6	**Reserved.** Must be set to 1.
5	**Partial Primary Hard Disk Decode.** This bit is used to restrict the addresses which are decoded as primary hard disk accesses. 0: Power management monitors all reads and writes to I/O Port 1F0h-1F7h, 3F6h-3F7h (excludes writes to 3F7h), and 170h-177h, 376h-377h (excludes writes to 377h). 1: Power management monitors only writes to I/O Port 1F6h and 1F7h.
4	**Partial Secondary Hard Disk Decode.** This bit is used to restrict the addresses which are decoded as secondary hard disk accesses. 0: Power management monitors all reads and writes to I/O Port 170h-177h, 376h-377h (excludes writes to 377h). 1: Power management monitors only writes to I/O Port 176h and 177h.
3:2	**Reserved.** Must be set to 0.
1	**Mouse on Serial Enable.** Mouse is present on a Serial Port. 0: No. 1: Yes. If a mouse is attached to a serial port (i.e., this bit is set to 1), that port is removed from the serial device list being used to monitor serial port access for power management purposes and added to the keyboard/mouse decode. This is done because a mouse, along with the keyboard, is considered an input device and is used only to determine when to blank the screen. This bit and bit 0 of this register determine the decode used for the Keyboard/Mouse Idle Timer Count Register (F0 Index 9Eh) as well as the Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch).
0	**Mouse Port Select.** Selects which serial port the mouse is attached to: 0: COM1 1: COM2. For more information see the description of bit 1 in this register (above).
Index 94h-95h	**Suspend Modulation Counter Register (R/W)** — **Reset Value: 0000h**
15:8	**Suspend Signal Asserted Counter.** This 8-bit counter represents the number of 32 µs intervals that the internal SUSP# signal is asserted to the GX1 module. Together with bits [7:0], perform the Suspend Modulation function for CPU power management. The ratio of SUSP# asserted-to-deasserted sets up an effective (emulated) clock frequency, allowing the power manager to reduce GX1 module power consumption. This counter is prematurely reset if an enabled speedup event occurs (i.e., IRQ speedups).

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
7:0	**Suspend Signal Deasserted Counter.** This 8-bit counter represents the number of 32 µs intervals that the internal SUSP# signal is deasserted to the GX1 module. Together with bits [15:8], perform the Suspend Modulation function for CPU power management. The ratio of SUSP# asserted-to-deasserted sets up an effective (emulated) clock frequency, allowing the power manager to reduce GX1 module power consumption. This counter is prematurely reset if an enabled speedup event occurs (i.e., IRQ speedups).
Index 96h	**Suspend Configuration Register (R/W)** — **Reset Value: 00h**
7:3	**Reserved.** Must be set to 0.
2	**Suspend Mode Configuration.** Special 3V Suspend mode to support powering down the GX1 module during Suspend. 0: Disable. 1: Enable.
1	**SMI Speedup Configuration.** Selects how the Suspend Modulation function should react when an SMI occurs. 0: Use the IRQ Speedup Timer Count Register (F0 Index 8Ch) to temporarily disable Suspend Modulation when an SMI occurs. 1: Disable Suspend Modulation when an SMI occurs until a read to the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h). The purpose of this bit is to disable Suspend Modulation while the GX1 module is in the System Management Mode so that VSA and Power Management operations occur at full speed. Two methods for accomplishing this are: Map the SMI into the IRQ Speedup Timer Count Register (F0 Index 8Ch). - or - Have the SMI disable Suspend Modulation until the SMI handler reads the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h). This the preferred method. This bit has no affect if the Suspend Modulation feature is disabled (bit 0 = 0).
0	**Suspend Modulation Feature Enable.** This bit is used to enable/disable the Suspend Modulation feature. 0: Disable. 1: Enable. When enabled, the internal SUSP# signal is asserted and deasserted for the durations programmed in the Suspend Modulation register (F0 Index 94h). The setting of this bit is mirrored in the Top Level PME/SMI Status register (F1BAR0+I/O Offset 00h/02h[15]. It is used by the SMI handler to determine if the SMI Speedup Disable register (F1BAR0+I/O Offset 08h) must be cleared on exit.
Index 97h	**Reserved** — **Reset Value: 00h**
Index 98h-99h	**Primary Hard Disk Idle Timer Count Register (Primary Channel) (R/W)** — **Reset Value: 0000h**
15:0	**Primary Hard Disk Idle Timer Count.** This idle timer is used to determine when the primary hard disk is not in use so that it can be powered down. The 16-bit value programmed here represents the period of hard disk inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured hard disk's data port (I/O port 1F0h or 3F6h). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[0] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[0].
Index 9Ah-9Bh	**Floppy Disk Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Floppy Disk Idle Timer Count.** This idle timer is used to determine when the floppy disk drive is not in use so that it can be powered down. The 16-bit value programmed here represents the period of floppy disk drive inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured floppy drive's data port (I/O port 3F5h or 375h). This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[1] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[1].

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index 9Ch-9Dh	**Parallel / Serial Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Parallel / Serial Idle Timer Count.** This idle timer is used to determine when the parallel and serial ports are not in use so that the ports can be power managed. The 16-bit value programmed in this register represents the period of inactivity for these ports after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the parallel (LPT) or serial (COM) I/O address spaces. If the mouse is enabled on a serial port, that port is not considered here. This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[2] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[2].
Index 9Eh-9Fh	**Keyboard / Mouse Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Keyboard / Mouse Idle Timer Count.** This idle timer determines when the keyboard and mouse are not in use so that the LCD screen can be blanked. The 16-bit value programmed in this register represents the period of inactivity for these ports after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to either the keyboard or mouse I/O address spaces (including the mouse serial port address space when a mouse is enabled on a serial port.) This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[3] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[3].
Index A0h-A1h	**User Defined Device 1 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 1 (UDEF1) Idle Timer Count.** This idle timer determines when the device configured as User Defined Device 1 (UDEF1) is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the F0 Index C0h (Base Address register) and F0 Index CCh (Control register). This counter uses a 1 second time base. To enable this timer, set F0 Index 81h[4] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[4].
Index A2h-A3h	**User Defined Device 2 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 2 (UDEF2) Idle Timer Count.** This idle timer determines when the device configured as UDEF2 is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the F0 Index C4h (Base Address register) and F0 Index CDh (Control register). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[5] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[5].
Index A4h-A5h	**User Defined Device 3 Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**User Defined Device 3 (UDEF3) Idle Timer Count.** This idle timer determines when the device configured as UDEF3 is not in use so that it can be power managed. The 16-bit value programmed in this register represents the period of inactivity for this device after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to memory or I/O address space configured in the UDEF3 Base Address Register (F0 Index C8h) and UDEF3 Control Register (F0 Index CEh). This counter uses a 1 second timebase. To enable this timer, set F0 Index 81h[6] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 85h/F5h[6].
Index A6h-ABh	**Reserved** — **Reset Value: 00h**
Index ACh-ADh	**Secondary Hard Disk Idle Timer Count Register (R/W)** — **Reset Value: 0000h**
15:0	**Secondary Hard Disk Idle Timer Count.** This idle timer is used to determine when the secondary hard disk is not in use so that it can be powered down. The 16-bit value programmed in this register represents the period of hard disk inactivity after which the system is alerted via an SMI. The timer is automatically reloaded with the count value whenever an access occurs to the configured hard disk's data port (I/O port 1F0h or 170h). This counter uses a 1 second timebase. To enable this timer, set F0 Index 83h[7] = 1. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[0]. Second level SMI status is reported at F0 Index 86h/F6h[4].

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index AEh	**CPU Suspend Command Register (WO)** **Reset Value: 00h**
7:0	**Software CPU Suspend Command.** If bit 0 in the Clock Stop Control register is set low (F0 Index BCh[0] = 0), a write to this register causes an internal SUSP#/SUSPA# handshake with the GX1 module, placing the GX1 module in a low-power state. The actual data written is irrelevant. Once in this state, any unmasked IRQ or SMI releases the GX1 module halt condition. If F0 Index BCh[0] = 1, writing to this register invokes a full system Suspend. In this case, the internal SUSP_3V signal is asserted after the SUSP#/SUSPA# halt. Upon a Resume event, the PLL delay programmed in the F0 Index BCh[7:4] is invoked, allowing the clock chip and GX1 module PLL to stabilize before deasserting SUSP#.
Index AFh	**Suspend Notebook Command Register (WO)** **Reset Value: 00h**
7:0	**Software CPU Stop Clock Suspend.** A write to this register causes a SUSP#/SUSPA# handshake with the CPU, placing the GX1 module in a low-power state. Following this handshake, the SUSP_3V signal is asserted. The SUSP_3V signal is intended to be used to stop all system clocks. Upon a Resume event, the internal SUSP_3V signal is deasserted. After a slight delay, the Core Logic module deasserts the SUSP# signal. Once the clocks are stable, the GX1 module deasserts SUSPA# and system operation resumes.
Index B0h-B3h	**Reserved** **Reset Value: 00h**
Index B4h	**Floppy Port 3F2h Shadow Register (RO)** **Reset Value: xxh**
7:0	**Floppy Port 3F2h Shadow.** Last written value of I/O Port 3F2h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B5h	**Floppy Port 3F7h Shadow Register (RO)** **Reset Value: xxh**
7:0	**Floppy Port 3F7h Shadow.** Last written value of I/O Port 3F7h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B6h	**Floppy Port 372h Shadow Register (RO)** **Reset Value: xxh**
7:0	**Floppy Port 372h Shadow.** Last written value of I/O Port 372h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B7h	**Floppy Port 377h Shadow Register (RO)** **Reset Value: xxh**
7:0	**Floppy Port 377h Shadow.** Last written value of I/O Port 377h. Required for support of FDC power On/Off and 0V Suspend/Resume coherency. This register is a copy of an I/O register which cannot safely be directly read. The value in this register is not deterministic of when the register is being read. It is provided here to assist in a Suspend-to-Disk operation.
Index B8h	**DMA Shadow Register (RO)** **Reset Value: xxh**
7:0	**DMA Shadow.** This 8-bit port sequences through the following list of shadowed DMA Controller registers. At power on, a pointer starts at the first register in the list and continuing through the other registers in subsequent reads according to the read sequence. 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: 1. DMA Channel 0 Mode Register 2. DMA Channel 1 Mode Register 3. DMA Channel 2 Mode Register 4. DMA Channel 3 Mode Register 5. DMA Channel 4 Mode Register 6. DMA Channel 5 Mode Register 7. DMA Channel 6 Mode Register 8. DMA Channel 7 Mode Register 9. DMA Channel Mask Register (bit 0 is channel 0 mask, etc.) 10. DMA Busy Register (bit 0 or 1 means a DMA occurred within last 1 msec, all other bits are 0)

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index B9h	**PIC Shadow Register (RO)** — **Reset Value: xxh**
7:0	**PIC Shadow.** This 8-bit port sequences through the following list of shadowed Interrupt Controller registers. At power on, a pointer starts at the first register in the list and continuing through the other registers in subsequent reads according to the read sequence. 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: 1. PIC1 ICW1 2. PIC1 ICW2 3. PIC1 ICW3 4. PIC1 ICW4 - Bits [7:5] of ICW4 are always 0. 5. PIC1 OCW2 - Bits [6:3] of OCW2 are always 0 (See Note). 6. PIC1 OCW3 - Bits [7:4] are 0 and bits [6:3] are 1. 7. PIC2 ICW1 8. PIC2 ICW2 9. PIC2 ICW3 10. PIC2 ICW4 - Bits [7:5] of ICW4 are always 0. 11. PIC2 OCW2 - Bits [6:3] of OCW2 are always 0 (See Note). 12. PIC2 OCW3 - Bits [7:4] are 0 and bits [6:3] are 1. **Note:** To restore OCW2 to the shadow register value, write the appropriate address twice. First with the shadow register value, then with the shadow register value ORed with C0h.
Index BAh	**PIT Shadow Register (RO)** — **Reset Value: xxh**
7:0	**PIT Shadow.** 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 continuing through the other registers in subsequent reads according to the read sequence. 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: 1. Counter 0 LSB (least significant byte) 2. Counter 0 MSB 3. Counter 1 LSB 4. Counter 1 MSB 5. Counter 2 LSB 6. Counter 2 MSB 7. Counter 0 Command Word 8. Counter 1 Command Word 9. Counter 2 Command Word **Note:** The LSB/MSB of the count is the Counter base value, not the current value. Bits [7:6] of the command words are not used.
Index BBh	**RTC Index Shadow Register (RO)** — **Reset Value: xxh**
7:0	**RTC Index Shadow.** The RTC Shadow register contains the last written value of the RTC Index register (I/O Port 070h).
Index BCh	**Clock Stop Control Register (R/W)** — **Reset Value: 00h**
7:4	**PLL Delay.** The programmed value in this field sets the delay (in milliseconds) after a break event occurs before the internal SUSP# signal is deasserted to the GX1 module. This delay is designed to allow the clock chip and CPU PLL to stabilize before starting execution. This delay is only invoked if the STP_CLK bit was set. The 4-bit field allows values from 0 to 15 msec. 0000: 0 msec — 0100: 4 msec — 1000: 8 msec — 1100: 12 msec 0001: 1 msec — 0101: 5 msec — 1001: 9 msec — 1101: 13 msec 0010: 2 msec — 0110: 6 msec — 1010: 10 msec — 1110: 14 msec 0011: 3 msec — 0111: 7 msec — 1011: 11 msec — 1111: 15 msec
3:1	**Reserved.** Set to 0.
0	**CPU Clock Stop.** 0: Normal internal SUSP#/SUSPA# handshake. 1: Full system Suspend.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Note:	This register configures the Core Logic module to support a 3V Suspend mode. Setting bit 0 causes the SUSP_3V signal to assert after the appropriate conditions, stopping the system clocks. A delay of 0-15 msec is programmable (bits [7:4]) to allow for a delay for the clock chip and CPU PLL to stabilize when an event Resumes the system. A write to the CPU Suspend Command register (F0 Index AEh) with bit 0 written as: 0: Internal SUSP#/SUSPA# handshake occurs. The GX1 module is put into a low-power state, and the system clocks are not stopped. When a break/resume event occurs, it releases the CPU halt condition. 1: Internal SUSP#/SUSPA# handshake occurs and the SUSP_3V signal is asserted, thus invoking a full system Suspend (both GX1 module and system clocks are stopped). When a break event occurs, the SUSP_3V signal is deasserted, the PLL delay programmed in bits [7:4] are invoked which allows the clock chip and GX1 module PLL to stabilize before deasserting the internal SUSP# signal.
Index BDh-BFh	**Reserved** — **Reset Value: 00h**
Index C0h-C3h	**User Defined Device 1 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 1 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CCh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.
Index C4h-C7h	**User Defined Device 2 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 2 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CDh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.
Index C8h-CBh	**User Defined Device 3 Base Address Register (R/W)** — **Reset Value: 00000000h**
31:0	**User Defined Device 3 Base Address.** This 32-bit register supports power management (Trap and Idle timer resources) for a PCMCIA slot or some other device in the system. The value in this register is used as the address comparator for the device trap/timer logic. The device can be memory or I/O mapped (configured in F0 Index CEh). The Core Logic module cannot snoop addresses on the Fast-PCI bus unless the it actually claims the cycle. Therefore, Traps and Idle timers cannot support power management of devices on the Fast-PCI bus.
Index CCh	**User Defined Device 1 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** Determines how User Defined Device 1 is mapped. 0: I/O. 1: Memory.
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 — 0: Disable write cycle tracking. 1: Enable write cycle tracking. Bit 5 — 0: Disable read cycle tracking. 1: Enable read cycle tracking. Bits [4:0] — Mask for address bits A[4:0]. If bit 7 = 1 (Memory): Bits [6:0] — Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.
Index CDh	**User Defined Device 2 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** Determines how User Defined Device 2 is mapped. 0: I/O 1: Memory

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 0: Disable write cycle tracking. 1: Enable write cycle tracking. Bit 5 0: Disable read cycle tracking. 1: Enable read cycle tracking. Bits [4:0] Mask for address bits A[4:0]. If bit 7 = 1 (Memory): Bits [6:0] Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.
Index CEh	**User Defined Device 3 Control Register (R/W)** — **Reset Value: 00h**
7	**Memory or I/O Mapped.** Determines how User Defined Device 3 is mapped. 0: I/O. 1: Memory.
6:0	**Mask.** If bit 7 = 0 (I/O): Bit 6 0: Disable write cycle tracking. 1: Enable write cycle tracking. Bit 5 0: Disable read cycle tracking. 1: Enable read cycle tracking. Bits [4:0] Mask for address bits A[4:0]. If bit 7 = 1 (Memory): Bits [6:0] Mask for address memory bits A[15:9] (512 bytes min. and 64 KB max.) A[8:0] are ignored. **Note:** A "1" in a mask bit means that the address bit is ignored for comparison.
Index CFh	**Reserved** — **Reset Value: 00h**
Index D0h	**Software SMI Register (WO)** — **Reset Value: 00h**
7:0	**Software SMI.** A write to this location generates an SMI. The data written is irrelevant. This register allows software entry into SMM via normal bus access instructions.
Index D1h-EBh	**Reserved** — **Reset Value: 00h**
Index ECh	**Timer Test Register (R/W)** — **Reset Value: 00h**
7:0	**Timer Test Value.** The Timer Test register is intended only for test and debug purposes. It is not intended for setting operational timebases.
Index EDh-F3h	**Reserved** — **Reset Value: 00h**
Index F4h	**Second Level PME/SMI Status Register 1 (RC)** — **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F0 Index 84h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F0 Index 84h can be read instead.	
7:3	**Reserved.** Reads as 0.
2	**GPWIO2 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO2 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO2 is enabled as an input: F1BAR1+I/O Offset 15h[2] = 0. 2) Set F1BAR1+I/O Offset 15h[6] = 1 to allow SMI generation.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
1	**GPWIO1 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO1 pin. 0: No. 1: Yes. To enable SMI generation: 1) Ensure that GPWIO1 is enabled as an input: F1BAR1+I/O Offset 15h[1] = 0. 2) Set F1BAR1+I/O Offset 15h[5] to 1 to allow SMI generation.
0	**GPWIO0 SMI Status.** Indicates whether or not an SMI was caused by a transition on the GPWIO0 pin. 0: No 1: Yes To enable SMI generation: 1) Ensure that GPWIO0 is enabled as an input: F1BAR1+I/O Offset 15h[0] = 0. 2) Set F1BAR1+I/O Offset 15h[4] to 1 to allow SMI generation.

Index F5h — Second Level PME/SMI Status Register 2 (RC) — Reset Value: 00h

The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0].

Reading this register clears the status at both the second and top levels.

A read-only “Mirror” version of this register exists at F0 Index 85h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F0 Index 85h can be read instead.

7	**Reserved.** Reads as 0.
6	**User Defined Device Idle Timer 3 (UDEF3) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 3 (UDEF3) Idle Timer Count Register (F0 Index A4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[6] = 1.
5	**User Defined Device Idle Timer 2 (UDEF2) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 2 (UDEF2) Idle Timer Count Register (F0 Index A2h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[5] = 1.
4	**User Defined Device Idle Timer 1 (UDEF1) SMI Status.** Indicates whether or not an SMI was caused by expiration of User Defined Device 1 (UDEF1) Idle Timer Count Register (F0 Index A0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[4] = 1.
3	**Keyboard/Mouse Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Keyboard/ Mouse Idle Timer Count Register (F0 Index 9Eh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[3] = 1.
2	**Parallel/Serial Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Parallel/Serial Port Idle Timer Count Register (F0 Index 9Ch). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[2] = 1.
1	**Floppy Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Floppy Disk Idle Timer Count Register (F0 Index 9Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[1] = 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
0	**Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Hard Disk Idle Timer Count Register (F0 Index 98h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 81h[0] = 1.
Index F6h	**Second Level PME/SMI Status Register 3 (RC)** **Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels. A read-only “Mirror” version of this register exists at F0 Index 86h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F0 Index 86h can be read instead.	
7	**Reserved.** Reads as 0.
6	**Reserved. (Read Only)**
5	**Secondary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the secondary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[6] = 1.
4	**Secondary Hard Disk Idle Timer SMI Status.** Indicates whether or not an SMI was caused by expiration of Secondary Hard Disk Idle Timer Count register (F0 Index ACh). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[7] = 1.
3	**Keyboard/Mouse Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the keyboard or mouse. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[3] = 1.
2	**Parallel/Serial Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to either the serial or parallel ports. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[2] =1.
1	**Floppy Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the floppy disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[1] = 1.
0	**Primary Hard Disk Access Trap SMI Status.** Indicates whether or not an SMI was caused by a trapped I/O access to the primary hard disk. 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[0] = 1.

Core Logic Module (Continued)

Table 5-29. F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support (Continued)

Bit	Description
Index F7h	**Second Level PME/SMI Status Register 4 (RC)**	**Reset Value: 00h**
The bits in this register contain second level status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[0]. Reading this register clears the status at both the second and top levels except for bit 7 which has a third level of status reporting at F0BAR0+I/O 0Ch/1Ch. A read-only “Mirror” version of this register exists at F0 Index 87h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F0 Index 87h can be read instead.	
7	**GPIO Event SMI Status (Read Only, Read does not Clear).** Indicates whether or not an SMI was caused by a transition of any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] = 0. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME and setting F1BAR1+I/O Offset 0Ch[0] = 0 enables the PME to generate an SMI. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h). The next level (third level) of SMI status is at F0BAR0+I/O 0Ch/1Ch.
6	**Thermal Override SMI Status.** Indicates whether or not an SMI was caused by an assertion of the THRM# signal. 0: No. 1: Yes. To enable SMI generation set F0 Index 83h[4] = 1.
5:4	**Reserved.** Reads as 0.
3	**SIO PWUREQ SMI Status.** Indicates whether or not an SMI was caused by a power-up event from the SIO. 0: No. 1: Yes. A power-up event is defined as any of the following events/activities: — RI# — IRRX1 (CEIR) To enable SMI generation, set F1BAR1+I/O Offset 0Ch[0] = 0.
2	**Codec SDATA_IN SMI Status.** Indicates whether or not an SMI was caused by AC97 codec producing a positive edge on SDATA_IN. 0: No. 1: Yes. To enable SMI generation, set F0 Index 80h[5] = 1.
1	**RTC Alarm (IRQ8#) SMI Status.** Indicates whether or not an SMI was caused by an RTC interrupt. 0: No. 1: Yes. This SMI event can only occur while in 3V Suspend and an RTC interrupt occurs and F1BAR1+I/O Offset 0Ch[0] = 0.
0	**ACPI Timer SMI Status.** Indicates whether or not an SMI was caused by an ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch) MSB toggle. 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[5] = 1.
Index F8h-FFh	**Reserved**	**Reset Value: 00h**

Core Logic Module (Continued)

5.4.1.1 GPIO Support Registers

F0 Index 10h, Base Address Register 0 (F0BAR0) points to the base address of where the GPIO runtime and configuration registers are located. Table 5-29 gives the bit formats of I/O mapped registers accessed through F0BAR0.

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers

Bit	Description
Offset 00h-03h	**GPDO0 — GPIO Data Out 0 Register (R/W)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data Out.** Bits [31:0] of this register correspond to GPIO31-GPIO0 signals, respectively. The value of each bit determines the value driven on the corresponding GPIO signal when its output buffer is enabled. Writing to the bit latches the written data unless the bit is locked by the GPIO Configuration register Lock bit (F0BAR0+I/O Offset 24h[3]). Reading the bit returns the value, regardless of the signal value and configuration. 0: Corresponding GPIO signal is driven to low when output enabled. 1: Corresponding GPIO signal is driven or released to high (according to buffer type and static pull-up selection) when output is enabled.
Offset 04h-07h	**GPDI0 — GPIO Data In 0 Register (RO)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data In.** Bits [31:0] of this register correspond to GPIO31-GPIO0 signals, respectively. Reading each bit returns the value of the corresponding GPIO signal, regardless of the signal configuration and the GPDO0 register (F0BAR0+I/O Offset 00h) value. Writes to this register are ignored. 0: Corresponding GPIO signal level is low. 1: Corresponding GPIO signal level is high.
Offset 08h-0Bh	**GPIEN0 — GPIO Interrupt Enable 0 Register (R/W)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Power Management Event (PME) Enable.** Bits [15:0] correspond to GPIO15-GPIO0 signals, respectively. Each bit allows PME generation by the corresponding GPIO signal. 0: Disable PME generation. 1: Enable PME generation. **Notes:** 1) All of the enabled GPIO PMEs are always reported at F1BAR1+I/O Offset 10h[3]. 2) Any enabled GPIO PME can be selected to generate an SCI or SMI at F1BAR1+I/O Offset 0Ch[0]. If SCI is selected, then the individually selected GPIO PMEs are globally enabled for SCI generation at F1BAR1+I/O Offset 12h[3] and the status is reported at F1BAR1+I/O Offset 10h[3]. If SMI is selected, the individually selected GPIO PMEs generate an SMI and the status is reported at F1BAR0+I/O Offset 00h/02h[0].
Offset 0Ch-0Fh	**GPST0 — GPIO Status 0 Register (R/W1C)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Status**. Bits [15:0] correspond to GPIO15-GPIO0 signals, respectively. Each bit reports a 1 when hardware detects the edge (rising/falling on the GPIO signal) that is programmed in F0BAR0+I/O Offset 24h[5]. If the corresponding bit in F0BAR0+I/O Offset 08h is set, this edge generates a PME. 0: No active edge detected since the bit was last cleared. 1: Active edge detected. Writing 1 to the Status bit clears it to 0. This is the third level of SMI status reporting to the second level at F0 Index 87h/F7h[7] and the top level at F1BAR0+I/O Offset 00h/02h[0]. Clearing the third level also clears the second and top levels. This is the second level of SCI status reporting to the top level at F1BAR1+Offset 10h[3]. The status must be cleared at both the this level and the top level (i.e., the top level is not automatically cleared when a bit in this register is cleared).

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
Offset 10h-13h	**GPDO1 — GPIO Data Out 1 Register (R/W)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data Out.** Bits [31:0] of this register correspond to GPIO63-GPIO32 signals, respectively. The value of each bit determines the value driven on the corresponding GPIO signal when its output buffer is enabled. Writing to the bit latches the written data unless the bit is locked by the GPIO Configuration register Lock bit (F0BAR0+I/O Offset 24h[3]). Reading the bit returns the value, regardless of the signal value and configuration. 0: Corresponding GPIO signal driven to low when output enabled. 1: Corresponding GPIO signal driven or released to high (according to buffer type and static pull-up selection) when output enabled.
Offset 14h-17h	**GPDI1 — GPIO Data In 1 Register (RO)** **Reset Value: FFFFFFFFh**
31:0	**GPIO Data In.** Bits [31:0] of this register correspond to GPIO63-GPIO32 signals, respectively. Reading each bit returns the value of the corresponding GPIO signal, regardless of the signal configuration and the GPDO1 register (F0BAR0+I/O Offset 10h) value. Writes to this register are ignored. 0: Corresponding GPIO signal level low. 1: Corresponding GPIO signal level high.
Offset 18h-1Bh	**GPIEN1 — GPIO Interrupt Enable 1 Register (R/W)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Power Management Event (PME) Enable.** Bits [15:0] of this register correspond to GPIO47-GPIO32 signals, respectively. Each bit allows PME generation by the corresponding GPIO signal. 0: Disable PME generation. 1: Enable PME generation. **Notes:** 1) All of the enabled GPIO PMEs are always reported at F1BAR1+I/O Offset 10h[3]. 2) Any enabled GPIO PME can be selected to generate an SCI or SMI at F1BAR1+I/O Offset 0Ch[0]. If SCI is selected, the individually selected GPIO PMEs are globally enabled for SCI generation at F1BAR1+I/O Offset 12h[3] and the status is reported at F1BAR1+I/O Offset 10h[3]. If SMI is selected, the individually selected GPIO PMEs generate an SMI and the status is reported at F1BAR0+I/O Offset 00h/02h[0].
Offset 1Ch-1Fh	**GPST1 — GPIO Status 1 Register (R/W1C)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:0	**GPIO Status**. Bits [15:0] correspond to GPIO47-GPIO32 signals, respectively. Each bit reports a 1 when hardware detects the edge (rising/falling on the GPIO signal) that is programmed in F0BAR0+I/O Offset 24h[5]. If the corresponding bit in F0BAR0+I/O Offset 18h is set, this edge generates a PME. 0: No active edge detected since the bit was last cleared. 1: Active edge detected. Writing 1 to the a Status bit clears it to 0. This is the third level of SMI status reporting to the second level at F0 Index 87h/F7h[7] and the top level at F1BAR0+I/O Offset 00h/02h[0]. Clearing the third level also clears the second and top levels. This is the second level of SCI status reporting to the top level at F1BAR1+Offset 10h[3]. The status must be cleared at both the this level and the top level (i.e., the top level is not automatically cleared when a bit in this register is cleared).
Offset 20h-23h	**GPIO Signal Configuration Select Register (R/W)** **Reset Value: 00000000h**
31:6	**Reserved.** Must be set to 0.
5	**Bank Select.** Selects the GPIO bank to be configured. 0: Bank 0 for GPIO0-GPIO31 signals. 1: Bank 1 for GPIO32-GPIO63 signals.

Core Logic Module (Continued)

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
4:0	**Signal Select.** Selects the GPIO signal to be configured in the Bank selected via bit 5 setting (i.e., Bank 0 or Bank 1). See Table 3-2 "Multiplexing, Interrupt Selection, and Base Address Registers" on page 50 for GPIO ball muxing options. GPIOs without an associated ball number are not available externally. **If bit 5 = 0; Bank 0** 00000 = GPIO0 (ball B22) 00001 = GPIO1 (ball AD24) 00010 = GPIO2 (ball B21) 00011 = GPIO3 (ball A22) 00100 = GPIO4 00101 = GPIO5 00110 = GPIO6 (ball AD12) 00111 = GPIO7 (ball AF11) 01000 = GPIO8 (ball AC12) 01001 = GPIO9 (ball AE12) 01010 = GPIO10 (ball AF12) 01011 = GPIO11 (ball AF19) 01100 = GPIO12 (ball AE23) 01101 = GPIO13 (ball AD23) 01110 = GPIO14 (ball D22) 01111 = GPIO15 (ball C22 10000 = GPIO16 (ball AE18) 10001 = GPIO17 (ball B23) 10010 = GPIO18 (ball AC24) 10011 = GPIO19 (ball Y24) 10100 = GPIO20 (ball D21) 10101 = GPIO21 10110 = GPIO22 10111 = GPIO23 11000 = GPIO24 11001 = GPIO25 11010 = GPIO26 11011 = GPIO27 11100 = GPIO28 11101 = GPIO29 11110 = GPIO30 11111 = GPIO31 **If bit 5 = 1; Bank 1** 00000 = GPIO32 (ball E26) 00001 = GPIO33 (ball D25 00010 = GPIO34 (ball D26) 00011 = GPIO35 (ball C25) 00100 = GPIO36 (ball C26) 00101 = GPIO37 (ball B24) 00110 = GPIO38 (ball AB23) 00111 = GPIO39 (ball A24) 01000 = GPIO40 (ball B20) 01001 = GPIO41 (ball AA24) 01010 = GPIO42 01011 = GPIO43 01100 = GPIO44 01101 = GPIO45 01110 = GPIO46 01111 = GPIO47 (ball AB24) 10000 = GPIO48 10001 = GPIO49 10010 = GPIO50 10011 = GPIO51 10100 = GPIO52 10101 = GPIO53 10110 = GPIO54 10111 = GPIO55 11000 = GPIO56 11001 = GPIO57 11010 = GPIO58 11011 = GPIO59 11100 = GPIO60 11101 = GPIO61 11110 = GPIO62 11111 = GPIO63

Offset 24h-27h **GPIO Signal Configuration Access Register (R/W)** **Reset Value: 00000044h**

This register is used to indicate configuration for the GPIO signal that is selected in the GPIO Signal Configuration Select Register (above).

Note: PME debouncing, polarity, and edge/level configuration is only applicable on GPIO0-GPIO15 signals (Bank 0 = 00000 to 01111) and on GPIO32-GPIO47 signals (Bank 1 settings of 00000 to 01111). The remaining GPIOs (GPIO16-GPIO31 and GPIO48-GPIO63) can not generate PMEs, therefore these bits have no function and read 0.

31:7	**Reserved.** Must be set to 0.
6	**PME Debounce Enable.** Enables/disables IRQ debounce (debounce period = 16 ms). 0: Disable. 1: Enable. (Default). See the note in the description of this register for more information about the default value of this bit.
5	**PME Polarity.** Selects the polarity of the signal that issues a PME from the selected GPIO signal (falling/low or rising/high). 0: Falling edge or low level input. (Default) 1: Rising edge or high level input. See the note in the description of this register for more information about the default value of this bit.
4	**PME Edge/Level Select.** Selects the type (edge or level) of the signal that issues a PME from the selected GPIO signal. 0: Edge input. (Default) 1: Level input. For normal operation, always set this bit to 0 (edge input). Erratic system behavior results if this bit is set to 1. See the note in the description of this register for more information about the default value of this bit.

Table 5-30. F0BAR0+I/O Offset: GPIO Configuration Registers (Continued)

Bit	Description
3	**Lock.** This bit locks the selected GPIO signal. Once this bit is set to 1 by software, it can only be cleared to 0 by power-on reset or by WATCHDOG reset. 0: No effect. (Default) 1: Direction, output type, pull-up and output value locked.
2	**Pull-Up Control.** Enables/disables the internal pull-up capability of the selected GPIO signal. It supports open-drain output signals with internal pull-ups and TTL input signals. 0: Disable. 1: Enable. (Default) Bits [1:0] of this register must = 01 for this bit to have effect.
1	**Output Type.** Controls the output buffer type (open-drain or push-pull) of the selected GPIO signal. 0: Open-drain. (Default) 1: Push-pull. Bit 0 of this register must be set to 1 for this bit to have effect.
0	**Output Enable.** Indicates the GPIO signal output state. It has no effect on input. 0: TRI-STATE. (Default) 1: Output enabled.
Offset 28h-2Bh	**GPIO Reset Control Register (R/W)** **Reset Value: 00000000h**
31:1	**Reserved.** Must be set to 0.
0	**GPIO Reset.** Reset the GPIO logic. 0: Disable. 1: Enable. Write 0 to clear. This bit is level-sensitive and must be cleared after the reset is enabled (normal operation requires this bit to be 0).

Core Logic Module (Continued)

5.4.1.2 LPC Support Registers

F0 Index 14h, Base Address Register 1 (F0BAR1) points to the base address of the register space that contains the configuration registers for LPC support. Table 5-31 gives the bit formats of the I/O mapped registers accessed through F0BAR1.

The LPC Interface supports all features described in the LPC bus specification 1.0, with the following exceptions:

- Only 8- or 16-bit DMA, depending on channel number. Does not support the optional larger transfer sizes.
- Only one external DRQ pin.

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers

Bit	Description
Offset 00h-03h	**SERIRQ_SRC — Serial IRQ Source Register (R/W)** **Reset Value: 00000000h**
31:21	**Reserved.**
20	**INTD Source.** Selects the interface source of the INTD# signal. 0: PCI (INTD#, ball V24). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
19	**INTC Source.** Selects the interface source of the INTC# signal. 0: PCI (INTC#, ball Y24) and program PMR[4] =1. 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
18	**INTB Source.** Selects the interface source of the INTB# signal. 0: PCI (INTB#, ball W24). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
17	**INTA Source.** Selects the interface source of the INTA# signal. 0: PCI (INTA#, ball AD26). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
16	**Reserved.** Set to 0.
15	**IRQ15 Source.** Selects the interface source of the IRQ15 signal. 0: ISA (IRQ15, ball AF19) and program PMR[18] = 0 and PMR[8] = 1. 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
14	**IRQ14 Source.** Selects the interface source of the IRQ14 signal. 0: ISA (IRQ14, ball AC6). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
13	**IRQ13 Source.** Selects the interface source of the internal IRQ13 signal. 0: ISA (IRQ13 internal signal - An input from the CPU indicating that a floating point error has been detected and that internal INTR should be asserted). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
12	**IRQ12 Source.** Selects the interface source of the IRQ12 signal. 0: ISA (IRQ12, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
11	**IRQ11 Source.** Selects the interface source of the IRQ11 signal. 0: ISA (IRQ11, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
10	**IRQ10 Source.** Selects the interface source of the IRQ10 signal. 0: ISA (IRQ10, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
9	**IRQ9 Source.** Selects the interface source of the IRQ9 signal. 0: ISA (IRQ9, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
8	**IRQ8# Source.** Selects the interface source of the IRQ8# signal. 0: ISA (IRQ8# internal signal - Connected to internal RTC). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
7	**IRQ7 Source.** Selects the interface source of the IRQ7 signal. 0: ISA (IRQ7, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
6	**IRQ6 Source.** Selects the interface source of the IRQ6 signal. 0: ISA (IRQ6, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
5	**IRQ5 Source.** Selects the interface source of the IRQ5 signal. 0: ISA (IRQ5, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
4	**IRQ4 Source.** Selects the interface source of the IRQ4 signal. 0: ISA (IRQ4, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
3	**IRQ3 Source.** Selects the interface source of the IRQ3 signal. 0: ISA (IRQ3, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
2	**Reserved.** Must be set to 0.
1	**IRQ1 Source.** Selects the interface source of the IRQ1 signal. 0: ISA (IRQ1, unavailable externally). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
0	**IRQ0 Source.** Selects the interface source of the IRQ0 signal. 0: ISA (IRQ0 internal signal - Connected to OUT0, System Timer, of the internal 8254 PIT). 1: LPC (SERIRQ, ball A24) and program PMR[16] = 1.
Offset 04h-07h	**SERIRQ_LVL — Serial IRQ Level Control Register (R/W)** **Reset Value: 00000000h**
31:21	**Reserved**
20	**INTD# Polarity.** If LPC is selected as the interface source for INTD# (F0BAR1+I/O Offset 00h[20] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
19	**INTC# Polarity.** If LPC is selected as the interface source for INTC# (F0BAR1+I/O Offset 00h[19] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
18	**INTB# Polarity.** If LPC is selected as the interface source for INTB# (F0BAR1+I/O Offset 00h[18] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
17	**INTA# Polarity.** If LPC is selected as the interface source for INTA# (F0BAR1+I/O Offset 00h[17] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
16	**Reserved.** Must be set to 0.
15	**IRQ15 Polarity.** If LPC is selected as the interface source for IRQ15 (F0BAR1+I/O Offset 00h[15] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
14	**IRQ14 Polarity.** If LPC is selected as the interface source for IRQ14 (F0BAR1+I/O Offset 00h[14] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
13	**IRQ13 Polarity.** If LPC is selected as the interface source for IQR13 (F0BAR1+I/O Offset 00h[13] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
12	**IRQ12 Polarity.** If LPC is selected as the interface source for IRQ12 (F0BAR1+I/O Offset 00h[12] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
11	**IRQ11 Polarity.** If LPC is selected as the interface source for IRQ11 (F0BAR1+I/O Offset 00h[11] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
10	**IRQ10 Polarity.** If LPC is selected as the interface source for IRQ10 (F0BAR1+I/O Offset 00h[10] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
9	**IRQ9 Polarity.** If LPC is selected as the interface source for IRQ9 (F0BAR1+I/O Offset 00h[9] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
8	**IRQ8# Polarity.** If LPC is selected as the interface source for IRQ8# (F0BAR1+I/O Offset 00h[8] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
7	**IRQ7 Polarity.** If LPC is selected as the interface source for IRQ7 (F0BAR1+I/O Offset 00h[7] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
6	**IRQ6 Polarity.** If LPC is selected as the interface source for IRQ6 (F0BAR1+I/O Offset 00h[6] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
5	**IRQ5 Polarity.** If LPC is selected as the interface source for IRQ5 (F0BAR1+I/O Offset 00h[5] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
4	**IRQ4 Polarity.** If SERIRQ is selected as the interface source for IRQ4 (F0BAR1+I/O Offset 00h[4] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
3	**IRQ3 Polarity.** If LPC is selected as the interface source for IRQ3 (F0BAR1+I/O Offset 00h[3] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
2	**SMI# Polarity.** This bit allows signal polarity selection of the SMI# generated from SERIRQ. 0: Active high. 1: Active low.
1	**IRQ1 Polarity.** If LPC is selected as the interface source for IRQ1 (F0BAR1+I/O Offset 00h[1] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.
0	**IRQ0 Polarity.** If LPC is selected as the interface source for IRQ0 (F0BAR1+I/O Offset 00h[0] = 1), this bit allows signal polarity selection. 0: Active high. 1: Active low.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 08h-0Bh	**SERIRQ_CNT — Serial IRQ Control Register (R/W)** — **Reset Value: 00000000h**
31:8	**Reserved.**
7	**Serial IRQ Enable.** 0: Disable. 1: Enable.
6	**Serial IRQ Interface Mode.** 0: Continuous. 1: Quiet.
5:2	**Number of IRQ Data Frames.** 0000: 17 frames — 0100: 21 frames — 1000: 25 frames — 1100: 29 frames 0001: 18 frames — 0101: 22 frames — 1001: 26 frames — 1101: 30 frames 0010: 19 frames — 0110: 23 frames — 1010: 27 frames — 1110: 31 frames 0011: 20 frames — 0111: 24 frames — 1011: 28 frames — 1111: 32 frames
1:0	**Start Frame Pulse Width.** 00: 4 Clocks 01: 6 Clocks 10: 8 Clocks 11: Reserved
Offset 0Ch-0Fh	**DRQ_SRC — DRQ Source Register (R/W)** — **Reset Value: 00000000h** DRQx are internal signals between the Core Logic and SuperI/O modules.
31:8	**Reserved.**
7	**DRQ7 Source.** Selects the interface source of the DRQ7 signal. 0: ISA (DRQ7, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
6	**DRQ6 Source.** Selects the interface source of the DRQ6 signal. 0: ISA (DRQ6, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
5	**DRQ5 Source.** Selects the interface source of the DRQ5 signal. 0: ISA (DRQ5, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
4	**LPC BM0 Cycles.** Allow LPC Bus Master 0 Cycles. 0: Enable. 1: Disable.
3	**DRQ3 Source.** Selects the interface source of the DRQ3 signal. 0: ISA (DRQ3, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
2	**DRQ2 Source.** Selects the interface source of the DRQ2 signal. 0: ISA (DRQ2, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
1	**DRQ1 Source.** Selects the interface source of the DRQ1 signal. 0: ISA (DRQ1, unavailable externally). 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.
0	**DRQ0 Source.** Selects the interface source of the DRQ0 signal. 0: ISA (DRQ0), unavailable externally. 1: LPC (LDRQ#, ball C26) and program PMR[14] = 1.

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
Offset 10h-13h	**LAD_EN — LPC Address Enable Register (R/W)** **Reset Value: 00000000h**
31:18	**Reserved.**
17	**LPC RTC.** RTC addresses I/O Ports 070h-073h. See bit 16 for decode.
16	**LPC/ISA Default Mapping.** Works in conjunction with bits 17 and [14:0] of this register to enable mapping of specific peripherals to LPC or internal ISA interfaces. If bit [x] = 0 and bit 16 = 0 then: Transaction routed to internal ISA bus. If bit [x] = 0 and bit 16 = 1 then: Transaction routed to LPC interface. If bit [x] = 1 and bit 16 = 0 then: Transaction routed to LPC interface. If bit [x] = 1 and bit 16 = 1 then: Transaction routed to internal ISA bus. Bit [x] is defined as bits 17 and [14:0].
15	**LPC ROM Addressing.** Depends upon F0 Index 52h[2,0]. 0: Disable. 1: Enable.
14	**LPC Alternate SuperI/O Addressing.** Alternate SuperI/O control addresses 4Eh-4Fh. See bit 16 for decode.
13	**LPC SuperI/O Addressing.** SuperI/O control addresses I/O Ports 2Eh-2Fh. See bit 16 for decode. **Note:** This bit should not be enabled when using the internal SuperI/O module and if IO_SIOCFG_IN (F5BAR0+I/O Offset 00h[26:25]) = 11.
12	**LPC Ad-Lib Addressing.** Ad-Lib addresses I/O Ports 388h-389h. See bit 16 for decode.
11	**LPC ACPI Addressing.** ACPI microcontroller addresses I/O Ports 62h and 66h. See bit 16 for decode.
10	**LPC Keyboard Controller Addressing.** KBC addresses I/O Ports 60h and 64h. **Note:** If this bit = 0 and bit 16 = 1, then F0 Index 5Ah[1] must be written 0.
9	**LPC Wide Generic Addressing.** Wide generic addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 18h[15:9] **Note:** The selected range must not overlap any address range that is positively decoded by F0BAR1+I/O Offset 10h bits [17], [14:10], and [8:0].
8	**LPC Game Port 1 Addressing.** Game Port 1 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[22:19]
7	**LPC Game Port 0 Addressing.** Game Port 0 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[18:15].
6	**LPC Floppy Disk Controller Addressing.** FDC addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[14]
5	**LPC Microsoft Sound System (MSS) Addressing.** MSS addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[13:12].
4	**LPC MIDI Addressing.** MIDI addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[11:10].
3	**LPC Audio Addressing.** Audio addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[9:8].
2	**LPC Serial Port 1 Addressing.** Serial Port 1 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[7:5].
1	**LPC Serial Port 0 Addressing.** Serial Port 0 addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[4:2].
0	**LPC Parallel Port Addressing.** Parallel Port addresses. See bit 16 for decode. Address selection made via F0BAR1+I/O Offset 14h[1:0].
Offset 14h-17h	**LAD_D0 — LPC Address Decode 0 Register (R/W)** **Reset Value: 00080020h**
31:15	**Reserved.**
14	**LPC Floppy Disk Controller Address Select.** Selects I/O Port: 0: 3F0h-3F7h. 1: 370h-377h. Selected address range is enabled via F0BAR1+I/O Offset 10h[6].

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
13:12	**LPC Microsoft Sound System (MSS) Address Select.** Selects I/O Port: 00: 530h-537h 10: E80h-E87h 01: 604h-60Bh 11: F40h-F47h Selected address range is enabled via F0BAR1+I/O Offset 10h[5].
11:10	**LPC MIDI Address Select.** Selects I/O Port: 00: 300h-301h 10: 320h-321h 01: 310h-311h 11: 330h-331h Selected address range is enabled via F0BAR1+I/O Offset 10h[4].
9:8	**LPC Audio Address Select.** Selects I/O Port: 00: 220h-233h 10: 260h-273h 01: 240h-253h 11: 280h-293h Selected address range is enabled via F0BAR1+I/O Offset 10h[3].
7:5	**LPC Serial Port 1 Address Select.** Selects I/O Port: 000: 3F8h-3FFh 010: 220h-227h 100: 238h-23Fh 110: 338h-33Fh 001: 2F8h-2FFh 011: 228h-22Fh 101: 2E8h-2EFh 111: 3E8h-3EFh Selected address range is enabled via F0BAR1+I/O Offset 10h[2].
4:2	**LPC Serial Port 0 Address Select.** Selects I/O Port: 000: 3F8h-3FFh 010: 220h-227h 100: 238h-23Fh 110: 338h-33Fh 001: 2F8h-2FFh 011: 228h-22Fh 101: 2E8h-2EFh 111: 3E8h-3EFh Selected address range is enabled via F0BAR1+I/O Offset 10h[1].
1:0	**LPC Parallel Port Address Select.** Selects I/O Port: 00: 378h-37Fh (+778h-77Fh for ECP) 01: 278h-27Fh (+678h-67Fh for ECP) (Note) 10: 3BCh-3BFh (+7BCh-7BFh for ECP) 11: Reserved Selected address range is enabled via F0BAR1+I/O Offset 10h[0]. **Note:** 279h is read only, writes are forwarded to ISA for PnP.
Offset 18h-1Bh	**LAD_D1 — LPC Address Decode 1 Register (R/W)** **Reset Value: 00000000h**
31:16	**Reserved.** Must be set to 0.
15:9	**Wide Generic Base Address Select.** Defines a 512 byte space. Can be mapped anywhere in the 64 KB I/O space. AC97 and other configuration registers are expected to be mapped to this range. It is wide enough to allow many unforeseen devices to be supported. Enabled at F0BAR1+I/O Offset 10h[9]. **Note:** The selected range must not overlap any address range that is positively decoded by F0BAR1+I/O Offset 10h bits [17], [14:10], and [8:0].
8:0	**Reserved.** Must be set to 0.
Offset 1Ch-1Fh	**LPC_ERR_SMI — LPC Error SMI Register (R/W)** **Reset Value: 00000080h**
31:10	**Reserved.** Must be set to 0.
9	**SMI Serial IRQ Enable.** Allows serial IRQ to generate an SMI. 0: Disable. 1: Enable. Top Level SMI status is reported at F1BAR0+I/O Offset 02h[3]. Second level status is reported at bit 6 of this register.
8	**SMI Configuration for LPC Error Enable.** Allows LPC errors to generate an SMI. 0: Disable. 1: Enable. Top Level SMI status is reported at F1BAR0+I/O Offset 02h[3]. Second level status is reported at bit 5 of this register.
7	**Reserved. (Read Only).**

Core Logic Module (Continued)

Table 5-31. F0BAR1+I/O Offset xxh: LPC Interface Configuration Registers (Continued)

Bit	Description
6	**SMI Source is Serial IRQ.** Indicates whether or not an SMI was generated by an SERIRQ. 0: No. 1: Yes. Write 1 to clear. To enable SMI generation, set bit 9 of this register to 1. This is the second level of status reporting. The top level status is reported in F1BAR0+I/O Offset 02h[3]. Writing a 1 to this bit also clears the top level status bit as long as bit 5 of this register is cleared.
5	**LPC Error Status.** Indicates whether or not an SMI was generated by an error that occurred on LPC. 0: No. 1: Yes. Write 1 to clear. To enable SMI generation, set bit 8 of this register to 1. This is the second level of status reporting. The top level status is reported in F1BAR0+I/O Offset 02h[3]. Writing a 1 to this bit also clears the top level status bit as long as bit 6 of this register is cleared.
4	**LPC Multiple Errors Status.** Indicates whether or not multiple errors have occurred on LPC. 0: No. 1: Yes. Write 1 to clear.
3	**LPC Timeout Error Status.** Indicates whether or not an error was generated by a timeout on LPC. 0: No. 1: Yes. Write 1 to clear.
2	**LPC Error Write Status.** Indicates whether or not an error was generated during a write operation on LPC. 0: No. 1: Yes. Write 1 to clear.
1	**LPC Error DMA Status.** Indicates whether or not an error was generated during a DMA operation on LPC. 0: No. 1: Yes. Write 1 to clear.
0	**LPC Error Memory Status.** Indicates whether or not an error was generated during a memory operation on LPC. 0: No. 1: Yes. Write 1 to clear.
Offset 20h-23h	**LPC_ERR_ADD — LPC Error Address Register (RO)** **Reset Value: 00000000h**
31:0	**LPC Error Address.**

Core Logic Module (Continued)

5.4.2 SMI Status and ACPI Registers - Function 1

The register space designated as Function 1 (F1) is used to configure the PCI portion of support hardware for the SMI Status and ACPI Support registers. The bit formats for the PCI Header registers are given in Table 5-32.

Located in the PCI Header registers of F1 are two Base Address Registers (F1BARx) used for pointing to the register spaces designated for SMI status and ACPI support, described later in this section.

Table 5-32. F1: PCI Header Registers for SMI Status and ACPI Support

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0511h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:1	**Reserved. (Read Only)**
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled to access I/O offsets through F1BAR0 and F1BAR1 (see F1 Index 10h and 40h).
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 068000h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F1BAR0 (R/W)** — **Reset Value: 00000001h** This register allows access to I/O mapped SMI status related registers. Bits [7:0] are read only (0000 0001), indicating a 256-byte I/O address range. Refer to Table 5-33 on page 208 for bit formats and reset values of the SMI status registers.
31:8	**SMI Status Base Address.**
7:0	**Address Range. (Read Only)**
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0501h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**
Index 40h-43h	**Base Address Register 1 - F1BAR1 (R/W)** — **Reset Value: 00000001h** This register allows access to I/O mapped ACPI related registers. Bits [7:0] are read only (0000 0001), indicating a 256 byte address range. Refer to Table 5-34 on page 216 for bit formats and reset values of the ACPI registers. **Note:** This Base Address register moved from its normal PCI Header Space (F1 Index 14h) to prevent plug and play software from relocating it after an FACP table is built.
31:8	**ACPI Base Address.**
7:1	**Address Range. (Read Only)**
0	**Enable. (Write Only)** This bit must be set to 1 to enable access to ACPI Support registers.
Index 44h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.2.1 SMI Status Support Registers

F1 Index 10h, Base Address Register 0 (F1BAR0), points to the base address for SMI Status register locations. Table 5-33 gives the bit formats of I/O mapped SMI Status registers accessed through F1BAR0.

The registers at F1BAR0+I/O Offset 50h-FFh can also be accessed F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space.

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers

Bit	Description
Offset 00h-01h	**Top Level PME/SMI Status Mirror Register (RO)** **Reset Value: 0000h**
Note:	Reading this register does not clear the status bits. For more information, see F1BAR0+I/O Offset 02h.
15	**Suspend Modulation Enable Mirror.** This bit mirrors the Suspend Mode Configuration bit (F0 Index 96h[0]). It is used by the SMI handler to determine if the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h) must be cleared on exit.
14	**SMI Source is USB.** Indicates whether or not an SMI was caused by USB activity 0: No. 1: Yes. To enable SMI generation, set F5BAR0+I/O Offset 00h[20:19] to 11.
13	**SMI Source is Warm Reset Command.** Indicates whether or not an SMI was caused by a Warm Reset command. 0: No. 1: Yes.
12	**SMI Source is NMI.** Indicates whether or not an SMI was caused by NMI activity. 0: No. 1: Yes.
11	**SMI Source is IRQ2 of SuperI/O.** Indicates whether or not an SMI was caused by SuperI/O IRQ2. 0: No. 1: Yes. The next level (second level) of SMI status is reported in the relevant SuperI/O module (configured to use IRQ2 via Index 70h, see Section 4.4 "Standard Configuration Registers" on page 70). For more information, see Table 4-27 "Banks 0 and 1 - Common Control and Status Registers" on page 94, Offset 00h.
10	**SMI Source is EXT_SMI[7:0].** Indicates whether or not an SMI was caused by a negative-edge event on EXT_SMI[7:0]. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 24h[23:8].
9	**SMI Source is GP Timers/UDEF/PCI/ISA Function Trap.** Indicates if an SMI was caused by: — Expiration of GP Timer 1 or 2. — Trapped access to UDEF1, 2, or 3. — Trapped access to F1-F3, F5, or ISA Legacy register space. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 04h/06h.
8	**SMI Source is Software Generated.** Indicates whether or not an SMI was caused by software. 0: No. 1: Yes.
7	**SMI on an A20M# Toggle.** Indicates whether or not an SMI was caused by a write access to either Port 92h or the keyboard command which initiates an A20M# SMI. 0: No. 1: Yes. This method of controlling the internal A20M# in the GX1 module is used instead of a pin. To enable SMI generation, set F0 Index 53h[0] to 1.
6:4	**Reserved.** Reads as 0.
3	**SMI Source is LPC.** Indicates whether or not an SMI was caused by the LPC interface. 0: No. 1: Yes. The next level (second level) of SMI status is at F0BAR1+I/O Offset 1Ch[6:5].

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
2	**SMI Source is ACPI.** Indicates whether or not an SMI was caused by an access (read or write) to one of the ACPI registers (F1BAR1). 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 20h.
1	**SMI Source is XpressAUDIO Subsystem.** Indicates whether or not an SMI was caused by the audio subsystem. 0: No. 1: Yes. The next level (second level) of SMI status is at F3BAR0+Memory Offset 10h/12h.
0	**SMI Source is Power Management Event.** Indicates whether or not an SMI was caused by one of the power management resources (except for GP timers, UDEFx and PCI/ISA function traps that are reported in bit 9). 0: No. 1: Yes. The next level (second level) of SMI status is at F0 Index 84h-F4h/87h-F7h.

Offset 02h-03h **Top Level PME/SMI Status Register (RO/RC)** **Reset Value: 0000h**

Note: Reading this register clears all the SMI status bits except for the “read only” bits, because they have a second level of status reporting. Clearing the second level status bits also clears the top level (except for GPIOs).

GPIO SMIs have a third level of SMI status reporting at F0BAR0+I/O Offset 0Ch/1Ch. Clearing the third level GPIO status bits also clears the second and top levels.

A read-only “Mirror” version of this register exists at F1BAR0+I/O Offset 00h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F1BAR0+I/O Offset 00h can be read instead.

15	**Suspend Modulation Enable Mirror. (Read to Clear)** This bit mirrors the Suspend Mode Configuration bit (F0 Index 96h[0]). It is used by the SMI handler to determine if the SMI Speedup Disable Register (F1BAR0+I/O Offset 08h) must be cleared on exit.
14	**SMI Source is USB. (Read to Clear)** Indicates whether or not an SMI was caused by USB activity. 0: No. 1: Yes. To enable SMI generation, set F5BAR0+I/O Offset 00h[20:19] to 11.
13	**SMI Source is Warm Reset Command. (Read to Clear)** Indicates whether or not an SMI was caused by Warm Reset command 0: No. 1: Yes.
12	**SMI Source is NMI. (Read to Clear)** Indicates whether or not an SMI was caused by NMI activity. 0: No. 1: Yes.
11	**SMI Source is IRQ2 of SuperI/O.** Indicates whether or not an SMI was caused by SuperI/O IRQ2. 0: No. 1: Yes. The next level (second level) of SMI status is reported in the relevant SuperI/O module (configured to use IRQ2 via Index 70h, see Section 4.4 "Standard Configuration Registers" on page 70). For more information, see Table 4-27 "Banks 0 and 1 - Common Control and Status Registers" on page 94, Offset 00h.
10	**SMI Source is EXT_SMI[7:0]. (Read Only. Read Does Not Clear)** Indicates whether or not an SMI was caused by a negative-edge event on EXT_SMI[7:0]. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 24h[23:8].
9	**SMI Source is General Timers/Traps. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the expiration of one of the General Purpose Timers or one of the User Defined Traps. 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 04h/06h.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
8	**SMI Source is Software Generated. (Read to Clear)** Indicates whether or not an SMI was caused by software. 0: No. 1: Yes.
7	**SMI on an A20M# Toggle. (Read to Clear)** Indicates whether or not an SMI was caused by an access to either Port 92h or the keyboard command which initiates an A20M# SMI. 0: No. 1: Yes. This method of controlling the internal A20M# in the GX1 module is used instead of a pin. To enable SMI generation, set F0 Index 53h[0] to 1.
6:4	**Reserved.** Reads as 0.
3	**SMI Source is LPC. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the LPC interface. 0: No. 1: Yes. The next level (second level) of SMI status is at F0BAR1+I/O Offset 1Ch[6:5].
2	**SMI Source is ACPI. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by an access (read or write) to one of the ACPI registers (F1BAR1). 0: No. 1: Yes. The next level (second level) of SMI status is at F1BAR0+I/O Offset 20h.
1	**SMI Source is XpressAUDIO Subsystem. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by the audio subsystem. 0: No. 1: Yes. The second level of status is found in F3BAR0+Memory Offset 10h/12h.
0	**SMI Source is Power Management Event. (Read Only, Read Does Not Clear)** Indicates whether or not an SMI was caused by one of the power management resources (except for GP timers, UDEFx and PCI/ISA function traps which are reported in bit 9). 0: No. 1: Yes. The next level (second level) of SMI status is at F0 Index 84h/F4h-87h/F7h.

Offset 04h-05h — **Second Level General Traps & Timers PME/SMI Status Mirror Register (RO)** — **Reset Value: 0000h**

The bits in this register contain second level status reporting. Top level status is reported at F1BAR0+I/O Offset 00h/02h[9].

Reading this register does not clear the SMI. For more information, see F1BAR0+I/O Offset 06h.

15:6	**Reserved.**
5	**PCI/ISA Function Trap.** Indicates whether or not an SMI was caused by a trapped PCI/ISA configuration cycle. 0: No. 1: Yes. To enable SMI generation for: — Trapped access to ISA Legacy I/O register space set F0 Index 41h[0] = 1. — Trapped access to F1 register space set F0 Index 41h[1] = 1. — Trapped access to F2 register space set F0 Index 41h[2] = 1. — Trapped access to F3 register space set F0 Index 41h[3] = 1. — Trapped access to F5 register space set F0 Index 41h[5] = 1.
4	**SMI Source is Trapped Access to User Defined Device 3.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 3 (F0 Index C8h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[6] = 1.

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
3	**SMI Source is Trapped Access to User Defined Device 2.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 2 (F0 Index C4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[5] = 1.
2	**SMI Source is Trapped Access to User Defined Device 1.** Indicates whether or not an SMI was caused by a trapped I/O or memory access to the User Defined Device 1 (F0 Index C0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[4] = 1.
1	**SMI Source is Expired General Purpose Timer 2.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 2 (F0 Index 8Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[1] = 1.
0	**SMI Source is Expired General Purpose Timer 1.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 1 (F0 Index 88h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[0] = 1.

Offset 06h-07h **Second Level General Traps & Timers Status Register (RC)** **Reset Value: 0000h**

The bits in this register contain second level of status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[9]. Reading this register clears the status at both the second and top levels.

A read-only “Mirror” version of this register exists at F1BAR0+I/O Offset 04h. If the value of this register must be read without clearing the SMI source (and consequently deasserting SMI), F1BAR0+I/O Offset 04h can be read instead.

15:6	**Reserved.**
5	**PCI/ISA Function Trap.** Indicates whether or not an SMI was caused by a trapped PCI/ISA configuration cycle 0: No. 1: Yes. To enable SMI generation for: — Trapped access to ISA Legacy I/O register space set F0 Index 41h[0] = 1. — Trapped access to F1 register space set F0 Index 41h[1] = 1. — Trapped access to F2 register space set F0 Index 41h[2] = 1. — Trapped access to F3 register space set F0 Index 41h[3] = 1. — Trapped access to F5 register space set F0 Index 41h[5] = 1.
4	**SMI Source is Trapped Access to User Defined Device 3 (UDEF3).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 3 (F0 Index C8h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[6] = 1.
3	**SMI Source is Trapped Access to User Defined Device 2 (UDEF2).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 2 (F0 Index C4h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[5] = 1.
2	**SMI Source is Trapped Access to User Defined Device 1 (UDEF1).** Indicates whether or not an SMI was caused by a trapped I/O or memory access to User Defined Device 1 (F0 Index C0h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 82h[4] = 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
1	**SMI Source is Expired General Purpose Timer 2.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 2 (F0 Index 8Ah). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[1] = 1.
0	**SMI Source is Expired General Purpose Timer 1.** Indicates whether or not an SMI was caused by the expiration of General Purpose Timer 1 (F0 Index 88h). 0: No. 1: Yes. To enable SMI generation, set F0 Index 83h[0] = 1.
Offset 08h-09h	**SMI Speedup Disable Register (Read to Enable)** — **Reset Value: 0000h**
15:0	**SMI Speedup Disable.** If bit 1 in the Suspend Configuration Register is set (F0 Index 96h[1] = 1), a read of this register invokes the SMI handler to re-enable Suspend Modulation. The data read from this register can be ignored. If the Suspend Modulation feature is disabled, reading this I/O location has no effect.
Offset 0Ah-1Bh	**Reserved** — **Reset Value: 00h**
These addresses should not be written.	
Offset 1Ch-1Fh	**ACPI Timer Register (RO)** — **Reset Value: xxxxxxxxh**
Note: This register can also be read at F1BAR1+I/O Offset 1Ch.	
31:24	**Reserved.**
23:0	**TMR_VAL.** This field returns the running count of the power management timer.
Offset 20h-21h	**Second Level ACPI PME/SMI Status Mirror Register (RO)** — **Reset Value: 0000h**
The bits in this register contain second level SMI status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[2]. Reading this register does not clear the SMI. For more information, see F1BAR0+I/O Offset 22h.	
15:6	**Reserved.** Always reads 0.
5	**ACPI BIOS SMI Status.** Indicates whether or not an SMI was caused by ACPI software raising an event to BIOS software. 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[2] to 1, and F1BAR1+I/O Offset 0Fh[0] to 1.
4	**PLVL3 SMI Status.** Indicates whether or not an SMI was caused by a read of the ACPI PLVL3 register (F1BAR1+I/O Offset 05h). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[11] to 1 (default).
3	**Reserved.**
2	**SLP_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[9] to 1 (default).
1	**THT_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI THT_EN bit (F1BAR1+I/O Offset 00h[4]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[8] to 1 (default).

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
0	**SMI_CMD SMI Status.** Indicates whether or not an SMI was caused by a write to the ACPI SMI_CMD register (F1BAR1+I/O Offset 06h). 0: No. 1: Yes. A write to the ACPI SMI_CMD register always generates an SMI.

Offset 22h-23h **Second Level ACPI PME/SMI Status Register (RC)** **Reset Value: 0000h**

The bits in this register contain second level of SMI status reporting. Top level is reported in F1BAR0+I/O Offset 00h/02h[2].

Reading this register clears the status at both the second and top levels.

A read-only "Mirror" version of this register exists at F1BAR0+I/O Offset 20h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F1BAR0+I/O Offset 20h can be read instead.

Bit	Description
15:6	**Reserved.** Always reads 0.
5	**ACPI BIOS SMI Status.** Indicates whether or not an SMI was caused by ACPI software raising an event to BIOS software. 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 0Ch[2] to 1, and F1BAR1+I/O Offset 0Fh[0] to 1.
4	**PLVL3 SMI Status.** Indicates whether or not an SMI was caused by a read of the ACPI PLVL3 register (F1BAR1+I/O Offset 05h). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[11] to 1 (default).
3	**Reserved.**
2	**SLP_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[9] to 1 (default).
1	**THT_EN SMI Status.** Indicates whether or not an SMI was caused by a write of 1 to the ACPI THT_EN bit (F1BAR1+I/O Offset 00h[4]). 0: No. 1: Yes. To enable SMI generation, set F1BAR1+I/O Offset 18h[8] to 1 (default).
0	**SMI_CMD SMI Status.** Indicates whether or not an SMI was caused by a write to the ACPI SMI_CMD register (F1BAR1+I/O Offset 06h). 0: No. 1: Yes. A write to the ACPI SMI_CMD register always generates an SMI.

Offset 24h-27h **External SMI Register (R/W)** **Reset Value: 00000000h**

Note: EXT_SMI[7:0] are external SMIs, meaning external to the Core Logic module.

Bits [23:8] of this register contain second level of SMI status reporting. Top level status is reported in F1BAR0+I/O Offset 00h/02h[10]. Reading bits [23:16] clears the second and top levels. If the value of the status bits must be read without clearing the SMI source (and consequently deasserting SMI), bits [15:8] can be read instead.

Bit	Description
31:24	**Reserved.** Must be set to 0.
23	**EXT_SMI7 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by assertion of EXT_SMI7. 0: No. 1: Yes. To enable SMI generation, set bit 7 to 1.
22	**EXT_SMI6 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI6. 0: No. 1: Yes. To enable SMI generation, set bit 6 to 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
21	**EXT_SMI5 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI5. 0: No. 1: Yes. To enable SMI generation, set bit 5 to 1.
20	**EXT_SMI4 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI4. 0: No. 1: Yes. To enable SMI generation, set bit 4 to 1.
19	**EXT_SMI3 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI3. 0: No. 1: Yes. To enable SMI generation, set bit 3 to 1.
18	**EXT_SMI2 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI2. 0: No. 1: Yes. To enable SMI generation, set bit 2 to 1.
17	**EXT_SMI1 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI1. 0: No. 1: Yes. To enable SMI generation, set bit 1 to 1.
16	**EXT_SMI0 SMI Status. (Read to Clear)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI0. 0: No. 1: Yes. To enable SMI generation, set bit 0 to 1.
15	**EXT_SMI7 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI7. 0: No. 1: Yes. To enable SMI generation, set bit 7 to 1.
14	**EXT_SMI6 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI6. 0: No. 1: Yes. To enable SMI generation, set bit 6 to 1.
13	**EXT_SMI5 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI5. 0: No. 1: Yes. To enable SMI generation, set bit 5 to 1.
12	**EXT_SMI4 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI4. 0: No. 1: Yes. To enable SMI generation, set bit 4 to 1.
11	**EXT_SMI3 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI3. 0: No. 1: Yes. To enable SMI generation, set bit 3 to 1.
10	**EXT_SMI2 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI2. 0: No. 1: Yes. To enable SMI generation, set bit 2 to 1.

Core Logic Module (Continued)

Table 5-33. F1BAR0+I/O Offset: SMI Status Registers (Continued)

Bit	Description
9	**EXT_SMI1 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI1. 0: No. 1: Yes. To enable SMI generation, set bit 1 to 1.
8	**EXT_SMI0 SMI Status. (Read Only)** Indicates whether or not an SMI was caused by an assertion of EXT_SMI0. 0: No. 1: Yes. To enable SMI generation, set bit 0 to 1.
7	**EXT_SMI7 SMI Enable.** When this bit is asserted, allow EXT_SMI7 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 23 (RC) and 15 (RO).
6	**EXT_SMI6 SMI Enable.** When this bit is asserted, allow EXT_SMI6 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 22 (RC) and 14 (RO).
5	**EXT_SMI5 SMI Enable.** When this bit is asserted, allow EXT_SMI5 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 21 (RC) and 13 (RO).
4	**EXT_SMI4 SMI Enable.** When this bit is asserted, allows EXT_SMI4 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 20 (RC) and 12 (RO).
3	**EXT_SMI3 SMI Enable.** When this bit is asserted, allow EXT_SMI3 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 19 (RC) and 11 (RO).
2	**EXT_SMI2 SMI Enable.** When this bit is asserted, allow EXT_SMI2 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 18 (RC) and 10 (RO).
1	**EXT_SMI1 SMI Enable.** When this bit is asserted, allow EXT_SMI1 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 17 (RC) and 9 (RO).
0	**EXT_SMI0 SMI Enable.** When this bit is asserted, allow EXT_SMI0 to generate an SMI on negative-edge events. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+00h/02h[10]. Second level SMI status is reported at bits 16 (RC) and 8 (RO).
Offset 28h-4Fh	**Not Used** — **Reset Value: 00h**
Offset 50h-FFh	The I/O mapped registers located here (F1BAR0+I/O Offset 50h-FFh) can also be accessed at F0 Index 50h-FFh. The preferred method is to program these registers through the F0 register space. Refer to Table 5-29 "F0: PCI Header and Bridge Configuration Registers for GPIO and LPC Support" on page 166 for more information about these registers.

Core Logic Module (Continued)

5.4.2.2 ACPI Support Registers

F1 Index 40h, Base Address Register 1 (F1BAR1), points to the base address of where the ACPI Support registers are located. Table 5-34 shows the I/O mapped ACPI Support registers accessed through F1BAR1.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers

Bit	Description
Offset 00h-03h	**P_CNT — Processor Control Register (R/W)** — **Reset Value: 00000000h**
31:5	**Reserved.** Always reads 0.
4	**THT_EN (Throttle Enable).** When this bit is asserted, it enables throttling of the clock based on the CLK_VAL field (bits [2:0] of this register). 0: Disable. 1: Enable. If F1BAR1+I/O Offset 18h[8] =1, an SMI is generated when this bit is set. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[1].
3	**Reserved.** Always reads 0.
2:0	**CLK_VAL (Clock Throttling Value).** CPU duty cycle: 000: Reserved — 010: 25% — 100: 50% — 110: 75% 001: 12.5% — 011: 37.5% — 101: 62.5% — 111: 87.5%
Offset 04h	**Reserved** — **Reset Value: 00h**
Note:	**This register should not be read.** It controls a reserved function of power management logic.
Offset 05h	**P_LVL3 — Enter C3 Power State Register (RO)** — **Reset Value: xxh**
7:0	**P_LVL3 (Power Level 3).** Reading this 8-bit read only register causes the processor to enter the C3 power state. Reads of P_LVL3 return 0. Writes have no effect. The ACPI state machine always waits for an SMI (any SMI) to be generated and serviced before transfer into C3 power state. A read of this register causes an SMI if enabled: F1BAR1+I/O Offset 18h[11] = 1. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[4].
Offset 06h	**SMI_CMD — OS/BIOS Requests Register (R/W)** — **Reset Value: 00h**
7:0	**SMI_CMD (SMI Command and OS / BIOS Requests).** A write to this register stores data and a read returns the last data written. In addition, a write to this register always generates an SMI. A read of this register does not generate an SMI. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[0].
Offset 07h	**ACPI_FUN_CNT — ACPI Function Control Register (R/W)** — **Reset Value: 00h**
7:5	**Reserved.** Must be set to 0.
4	**INTR_WU_SL1.** Enables wakeup on enabled interrupts in sleep state SL1. 0: Disable wakeup from SL1, when an enabled interrupt is active. 1: Enable wakeup from SL1, when an enabled interrupt is active.
3	**GPWIO_DBNC_DIS (GPWIO0 and GPWIO1 Debouncers).** Debounce settings for GPWIO0 (ball AC15) and GPWIO1 (ball AE16). Selects the time that a high-to-low or low-to-high transition (debounce period) must be for GPWIO0 to be recognized. 0: Debounce period is 15.8 msec. (Default) 1: Debounce period is 31 µs. GPWIO2 is fixed at 31 µs.
2:1	**Reserved.** Must be set to 0.
0	**PWRBTN_DBNC_DIS (Power Button Debouncer).** Allow a high-to-low or low-to-high transition of greater than 15.8 msec (debounce period) on PWRBTN# (ball AF15) before it is recognized. 0: Enable. (Default) 1: Disable.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
Offset 08h-09h	**PM1A_STS — PM1A Top Level PME/SCI Status Register (R/W)** **Reset Value: 0000h**
Notes: 1. This is the top level of PME/SCI status reporting for these events. There is no second level. 2. If SCI generation is not desired, the status bits are still set by the described conditions and can be used for monitoring purposes.	
15	**WAK_STS (Wakeup Status).** Indicates whether or not an SCI was caused by the occurrence of an enabled wakeup event. 0: No. 1: Yes. This bit is set when the system is in any Sleep state and an enabled wakeup event occurs (wakeup events are configured at F1BAR1+I/O Offset 0Ah and 12h). After this bit is set, the system transitions to a Working state. SCI generation is always enabled. Write 1 to clear.
14:12	**Reserved.** Must be set to 0.
11	**PWRBTNOR_STS (Power Button Override Status).** Indicates whether or not an SCI was caused by the power button being active for greater than 4 seconds. 0: No. 1: Yes. SCI generation is always enabled. Write 1 to clear.
10	**RTC_STS (Real-Time Clock Status).** Indicates if a Power Management Event (PME) was caused by the RTC generating an alarm (RTC IRQ signal is asserted). 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[10] to 1 and F1BAR1+I/O Offset 0Ch[0] to 1. (See Note 2 in the general description of this register.) Write 1 to clear.
9	**Reserved.** Must be set to 0.
8	**PWRBTN_STS (Power Button Status).** Indicates if PME was caused by the PWRBTN# (ball AF15) going low while the system is in a Working state. 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[8] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) In a Sleep state or the Soft-Off state, a wakeup event is generated when the power button is pressed (regardless of the PWRBTN_EN bit, F1BAR1+I/O Offset 0Ah[8], setting). Write 1 to clear.
7:6	**Reserved.** Must be set to 0.
5	**GBL_STS (Global Lock Status).** Indicates if PME was caused by the BIOS releasing control of the global lock. 0: No. 1: Yes. This bit is used by the BIOS to generate an SCI. BIOS writes the BIOS_RLS bit (F1BAR1+I/O Offset 0Fh[1]) which in turns sets the GBL_STS bit and raises a PME. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[5] to 1 and F1BAR1+I/O Offset 0Ch[0] to 1. (See Note 2 in the general description of this register.) Write 1 to clear.
4	**BM_STS (Bus Master Status).** Indicates if PME was caused by a system bus master requesting the system bus. 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ch[1] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) Write 1 to clear.
3:1	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
0	**TMR_STS (Timer Carry Status).** Indicates if SCI was caused by an MSB toggle (MSB changes from low-to-high or high-to-low) on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: No. 1: Yes. For the PME to generate an SCI, set F1BAR1+I/O Offset 0Ah[0] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register.) Write 1 to clear.
Offset 0Ah-0Bh	**PM1A_EN — PM1A PME/SCI Enable Register (R/W)** **Reset Value: 0000h**
In order for the ACPI events described below to generate an SCI, the SCI_EN bit must also be set (F1BAR1+I/O Offset 0Ch[0] = 1). The SCIs enabled via this register are globally enabled by setting F1BAR1+I/O Offset 08h. There is no second level of SCI status reporting for these bits.	
15:11	**Reserved.** Must be set to 0.
10	**RTC_EN (Real-Time Clock Enable).** Allow SCI generation when the RTC generates an alarm (RTC IRQ signal is asserted). 0: Disable. 1: Enable
9	**Reserved.** Must be set to 0.
8	**PWRBTN_EN (Power Button Enable).** Allow SCI generation when the PWRBTN# (ball AF15) goes low while the system is in a Working state. 0: Disable. 1: Enable
7:6	**Reserved.** Must be set to 0.
5	**GBL_EN (Global Lock Enable).** Allow SCI generation when the BIOS releases control of the global lock via the BIOS_RLS (F1BAR1+I/O Offset 0Fh[1] and GBL_STS (F1BAR1+I/O Offset 08h[5]) bits. 0: Disable. 1: Enable
4:1	**Reserved.** Must be set to 0.
0	**TMR_EN (ACPI Timer Enable).** Allow SCI generation for MSB toggles (MSB changes from low-to-high or high-to-low) on the ACPI Timer (F1BAR0+I/O Offset 1Ch or F1BAR1+I/O Offset 1Ch). 0: Disable. 1: Enable
Offset 0Ch-0Dh	**PM1A_CNT — PM1A Control Register (R/W)** **Reset Value: 0000h**
15:14	**Reserved.** Must be set to 0.
13	**SLP_EN (Sleep Enable). (Write Only)** Allow the system to sequence into the sleeping state associated with the SLP_TYPx (bits [12:10]). 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. The ACPI state machine always waits for an SMI (any SMI) to be generated and serviced before transitioning into a Sleep state. If F1BAR1+I/O Offset 18h[9] = 1, an SMI is generated when SLP_EN is set. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[2].
12:10	**SLP_TYPx (Sleep Type).** Defines the type of Sleep state the system enters when SLP_EN (bit 13) is set. 000: Sleep State S0 (Full on) — 100: Sleep State SL4 001: Sleep State SL1 — 101: Sleep State SL5 (Soft off) 010: Sleep State SL2 — 110: Reserved 011: Sleep State SL3 — 111: Reserved
9:3	**Reserved.** Set to 0.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
2	**GBL_RLS (Global Release). (Write Only)** This write only bit is used by ACPI software to raise an event to the BIOS software (i.e., it generates an SMI to pass execution control to the BIOS). 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. To generate an SMI, ACPI software writes the GBL_RLS bit which in turn sets the BIOS_STS bit (F1BAR1+I/O Offset 0Eh[0]) and raises a PME. For the PME to generate an SMI, set BIOS_EN (F1BAR1+I/O Offset 0Fh[0] to 1). The top level SMI status is reported at F1BAR0+I/O offset 00h/02h. Second level status is at F1BAR0+I/O Offset 22h[5].
1	**BM_RLD (Bus Master RLD).** If the processor is in the C3 state and a bus master request is generated, force the processor to transition to the C0 state. 0: Disable. 1: Enable.
0	**SCI_EN (System Control Interrupt Enable).** Globally selects power management events (PMEs) reported in PM1A_STS and GPE0_STS (F1BAR1+I/O Offset 08h and 10h) to be either an SCI or SMI type of interrupt. 0: APM Mode, generates an SMI and status is reported at F1BAR0+I/O Offset 00h/02h[0]. 1: ACPI Mode, generates an SCI if the corresponding PME enable bit is set and status is reported at F1BAR1+I/O Offset 08h and 10h. **Note:** This bit enables the ACPI state machine.
Offset 0Eh	**ACPI_BIOS_STS Register (R/W)** — **Reset Value: 00h**
7:1	**Reserved.** Must be set to 0.
0	**BIOS_STS (BIOS Status Release).** When 1 is written to the GLB_RLS bit (F1BAR1+I/O Offset 0Ch[2]), this bit is also set to 1. Write 1 to clear.
Offset 0Fh	**ACPI_BIOS_EN Register (R/W)** — **Reset Value: 00h**
7:2	**Reserved.** Must be set to 0.
1	**BIOS_RLS (BIOS Release). (Write Only)** When this bit is asserted, allow the BIOS to release control of the global lock. 0: Disable. 1: Enable. This is a write only bit and reads of this bit always return a 0. To generate an SCI, the BIOS writes the BIOS_RLS bit which in turn sets the GBL_STS bit (F1BAR1+I/O Offset 08h[5]) and raises a PME. For the PME to generate an SCI, set GBL_EN (F1BAR1+I/O Offset 0Ah[5] to 1).
0	**BIOS_EN (BIOS Enable).** When this bit is asserted, allow SMI generation by ACPI software via writes to GBL_RLS (F1BAR1+I/O Offset 0Ch[2]). 0: Disable. 1: Enable.
Offset 10h-11h	**GPE0_STS — General Purpose Event 0 PME/SCI Status Register (R/W)** — **Reset Value: xxxxh**
Notes: 1) This is the top level of PME/SCI status reporting. There is no second level except for bit 3 (GPIOs) where the next level of status is reported at F0BAR0+I/O Offset 0Ch/1Ch. 2) If SCI generation is not desired, the status bits are still set by the described conditions and can be used for monitoring purposes.	
15:12	**Reserved.** Must be set to 0.
11	**Reserved.**

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
10	**GPWIO2_STS.** Indicates if PME was caused by activity on GPWIO2 (ball AF17). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO2 is enabled as an input (F1BAR1+I/O Offset 15h[2] = 0) 2) Set F1BAR1+I/O Offset 12h[10] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.) If F1BAR1+I/O Offset 15h[6] = 1 it overrides these settings and GPWIO2 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
9	**GPWIO1_STS.** Indicates if PME was caused by activity on GPWIO1 (ball AE16). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO1 is enabled as an input (F1BAR1+I/O Offset 15h[1] = 0) 2) Set F1BAR1+I/O Offset 12h[9] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.) If F1BAR1+I/O Offset 15h[5] = 1 it overrides these settings and GPWIO1 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
8	**GPWIO0_STS.** Indicates if PME was caused by activity on GPWIO0 (ball AC15). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI: 1) Ensure that GPWIO0 is enabled as an input (F1BAR1+I/O Offset 15h[0] = 0) 2) Set F1BAR1+I/O Offset 12h[8] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above). If F1BAR1+I/O Offset 15h[4] = 1 it overrides these settings and GPWIO0 generates an SMI and the status is reported in F1BAR0+00h/02h[0].
7	**Reserved.** Must be set to 0.
6	**USB_STS.** Indicates if PME was caused by a USB interrupt event. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[6] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)
5	**THRM_STS.** Indicates if PME was caused by activity on THRM# (ball AE15). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[5] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1, (See Note 2 in the general description of this register above,)
4	**SMI_STS.** Indicates if PME was caused by activity on the internal SMI# signal. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[4] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
3	**GPIO_STS.** Indicates if PME was caused by activity on any of the GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0). 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[3] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above). F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled to generate a PME. In addition, the selected GPIO must be enabled as an input (F0BAR0+I/O Offset 20h and 24h).
2:1	**Reserved.** Reads as 0.
0	**PWR_U_REQ_STS.** Indicates if PME was caused by a power-up request event from the SuperI/O module. 0: No. 1: Yes. Write 1 to clear. For the PME to generate an SCI, set F1BAR1+I/O Offset 12h[0] = 1 and F1BAR1+I/O Offset 0Ch[0] = 1. (See Note 2 in the general description of this register above.)

Offset 12h-13h GPE0_EN — General Purpose Event 0 Enable Register (R/W) Reset Value: 0000h

In order for the ACPI events described below to generate an SCI, the SCI_EN bit must also be set (F1BAR1+I/O Offset 0Ch[0] = 1).

The SCIs enabled in this register are globally enabled by setting F1BAR1+I/O Offset 0Ch[0] to 1. The status of the SCIs is reported in F1BAR1+I/O Offset 10h.

15:12	**Reserved.**
11	**Reserved.**
10	**GPWIO2_EN.** Allow GPWIO2 (ball AF17) to generate an SCI. 0: Disable. 1: Enable. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[6] to force an SMI.
9	**GPWIO1_EN.** Allow GPWIO1 (ball AE16) to generate an SCI. 0: Disable. 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[5] to force an SMI.
8	**GPWIO0_EN.** Allow GPWIO0 (ball AC15) to generate an SCI. 0: Disable. 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. The setting of this bit can be overridden via F1BAR1+I/O Offset 15h[4] to force an SMI.
7	**Reserved.** Must be set to 0
6	**USB_EN.** Allow USB events to generate a SCI. 0: Disable. 1: Enable
5	**THRM_EN.** Allow THRM# (ball AE15) to generate an SCI. 0: Disable. 1: Enable.
4	**SMI_EN.** Allow SMI events to generate an SCI. 0: Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
3	**GPIO_EN.** Allow GPIOs (GPIO47-GPIO32 and GPIO15-GPIO0) to generate an SCI. 0: Disable. 1: Enable. F0BAR0+I/O Offset 08h/18h selects which GPIOs are enabled for PME generation. This bit (GPIO_EN) globally enables those selected GPIOs for generation of an SCI.
2:1	**Reserved.** Must be set to 0.
0	**PWR_U_REQ_EN.** Allow power-up request events from the SuperI/O module to generate an SCI. 0: Disable. 1: Enable. A power-up request event is defined as any of the following events/activities: Modem, Telephone, Keyboard, Mouse, CEIR (Consumer Electronic Infrared)
Offset 14h	**GPWIO Control Register 1 (R/W)** — **Reset Value: 00h**
7:3	**Reserved.** Must be set to 0.
2	**GPWIO2_POL.** Select GPWIO2 (ball AF17) polarity. 0: Active high. 1: Active low.
1	**GPWIO1_POL.** Select GPWIO1 (ball AE16) polarity. 0: Active high. 1: Active low.
0	**GPWIO0_POL.** Select GPWIO0 (ball AC15) polarity. 0: Active high. 1: Active low.
Offset 15h	**GPWIO Control Register 2 (R/W)** — **Reset Value: 00h**
7	**Reserved.** Must be set to 0.
6	**GPWIO_SMIEN2.** Allow GPWIO2 (ball AF17) to generate an SMI. 0: Disable. (Default) 1: Enable. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized. Bit 2 of this register must be set to 0 (input) for GPWIO2 to be able to generate an SMI. If asserted, this bit overrides the setting of F1BAR1+I/O Offset 12h[10] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
5	**GPWIO_SMIEN1.** Allow GPWIO1 (ball AE16) to generate an SMI. 0: Disable. (Default) 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. Bit 1 of this register must be set to 0 (input) for GPWIO1 to be able to generate an SMI. If asserted, this bit overrides the setting of F1BAR1+I/O Offset 12h[9] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
4	**GPWIO_SMIEN0.** Allow GPWIO0 (ball AC15) to generate an SMI. 0: Disable. (Default) 1: Enable. See F1BAR1+I/O Offset 07h[3] for debounce information. Bit 0 of this register must be set to 0 (input) for GPWIO0 to be able to generate an SMI. If enabled, this bit overrides the setting of F1BAR1+I/O Offset 12h[8] and its status is reported in F1BAR0+I/O Offset 00h/02h[0].
3	**Reserved.** Set to 0.
2	**GPWIO2_DIR.** Selects the direction of GPWIO2 (ball AF17). 0: Input. 1: Output.

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
1	**GPWIO1_DIR.** Selects the direction of GPWIO1 (ball AE16). 0: Input. 1: Output.
0	**GPWIO0_DIR.** Selects the direction of the GPWIO0 (ball AC15). 0: Input. 1: Output.
Offset 16h	**GPWIO Data Register (R/W)** — **Reset Value: 00h**
This register contains the direct values of the GPWIO2-GPWIO0 pins. Write operations are valid only for bits defined as outputs. Reads from this register read the last written value if the pin is an output. The pins are configured as inputs or outputs in F1BAR1+I/O Offset 15h.	
7:3	**Reserved.** Must be set to 0.
2	**GPWIO2_DATA.** Reflects the level of GPWIO2 (ball AF17). 0: Low. 1: High. A fixed high-to-low or low-to-high transition (debounce period) of 31 µs exists in order for GPWIO2 to be recognized.
1	**GPWIO1_DATA.** Reflects the level of GPWIO1 (ball AE16). 0: Low. 1: High. See F1BAR1+I/O Offset 07h[3] for debounce information.
0	**GPWIO0_DATA.** Reflects the level of GPWIO0 (ball AC15). 0: Low. 1: High. See F1BAR1+I/O Offset 07h[3] for debounce information.
Offset 17h	**Reserved** — **Reset Value: 00h**
Offset 18h-1Bh	**ACPI SCI_ROUTING Register (R/W)** — **Reset Value: 00000F00h**
31:17	**Reserved.**
16	**PCTL_DELAYEN.** Allow staggered delays on the activation and deactivation of the power control pins PWRCNT1, PWRCNT2, and ONCTL# by 2 msec each. 0: Disable. (Default) 1: Enable.
15:12	**Reserved.**
11	**PLVL3_SMIEN.** Allow SMI generation when the PLVL3 Register (F1BAR1+I/O Offset 05h) is read. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[4].
10	**Reserved**.
9	**SLP_SMIEN.** Allow SMI generation when the SLP_EN bit (F1BAR1+I/O Offset 0Ch[13]) is set. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[2].
8	**THT_SMIEN.** Allow SMI generation when the THT_EN bit (F1BAR1+I/O Offset 00h[4]) is set. 0: Disable. 1: Enable. (Default) Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[2]. Second level SMI status is reported at F1BAR0+I/O Offset 20h/22h[1].
7:4	**Reserved.**

Core Logic Module (Continued)

Table 5-34. F1BAR1+I/O Offset: ACPI Support Registers (Continued)

Bit	Description
3:0	**SCI_IRQ_ROUTE.** SCI is routed to: 0000: Disable 0100: IRQ4 1000: IRQ8# 1100: IRQ12 0001: IRQ1 0101: IRQ5 1001: IRQ9 1101: IRQ13 0010: Reserved 0010: IRQ6 1010: IRQ10 1110: IRQ14 0011: IRQ3 0011: IRQ7 1011: IRQ11 1111: IRQ15 For more details see Section 5.2.6.3 "Programmable Interrupt Controller" on page 130.
Offset 1Ch-1Fh	**ACPI Timer Register (RO)** — **Reset Value: xxxxxxxxh**
Note: This register can also be read at F1BAR0+I/O Offset 1Ch.	
31:24	**Reserved.**
23:0	**TMR_VAL. (Read Only)** This bit field contains the running count of the power management timer.
Offset 20h	**PM2_CNT — PM2 Control Register (R/W)** — **Reset Value: 00h**
7:1	**Reserved.**
0	**Arbiter Disable.** Disables the PCI arbiter when set by the OS. Used during C3 transition. 0: Arbiter not disabled. (Default) 1: Disable arbiter.
Offset 21h-FFh	**Reserved** — **Reset Value: 00h**
The read value for these registers is undefined.	

Core Logic Module (Continued)

5.4.3 IDE Controller Registers - Function 2

The register space designated as Function 2 (F2) is used to configure Channels 0 and 1 and the PCI portion of support hardware for the IDE controllers. The bit formats for the PCI Header/Channels 0 and 1 Registers are given in Table 5-35.

Located in the PCI Header Registers of F2 is a Base Address Register (F2BAR4) used for pointing to the register space designated for support of the IDE controllers, described later in this section.

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0502h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:3	**Reserved. (Read Only)**
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Reserved. (Read Only)**
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled, in order to access I/O offsets through F2BAR4 (for more information see F2 Index 20h).
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 01h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 010180h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F2BAR0 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 14h-17h	**Base Address Register 1 - F2BAR1 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 18h-1Bh	**Base Address Register 2 - F2BAR2 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 1Ch-1Fh	**Base Address Register 3 - F2BAR3 (RO)** — **Reset Value: 00000000h** Reserved. Reserved for possible future use by the Core Logic module.
Index 20h-23h	**Base Address Register 4 - F2BAR4 (R/W)** — **Reset Value: 00000001h** **Base Address 0 Register.** This register allows access to I/O mapped Bus Mastering IDE registers. Bits [3:0] are read only (0001), indicating a 16-byte I/O address range. Refer to Table 5-36 on page 229 for the IDE controller register bit formats and reset values.
31:4	**Bus Mastering IDE Base Address.**
3:0	**Address Range. (Read Only)**
Index 24h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0502h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 40h-43h	**Channel 0 Drive 0 PIO Register (R/W)** — **Reset Value: 00009172h**
If Index 44h[31] = 0, Format 0. Then bits, [15:0], in this register configure the same timing for both command and data. Format 0 settings for a Fast-PCI clock frequency of 33.3 MHz: — PIO Mode 0 = 00009172h — PIO Mode 1 = 00012171h — PIO Mode 2 = 00020080h — PIO Mode 3 = 00032010h — PIO Mode 4 = 00040010h Format 0 settings for a Fast-PCI clock frequency of 66.7 MHz: — PIO Mode 0 = 0000F8E4h — PIO Mode 1 = 000153F3h — PIO Mode 2 = 000213F1h — PIO Mode 3 = 00034231h — PIO Mode 4 = 00041131h **Note:** All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.	
31:20	**Reserved.** Must be set to 0.
19:16	**PIOMODE.** PIO mode.
15:12	**t2I.** Recovery time (value + 1 cycle).
11:8	**t3.** IDE_IOW# data setup time (value + 1 cycle).
7:4	**t2W.** IDE_IOW# width minus t3 (value + 1 cycle).
3:0	**t1.** Address Setup Time (value + 1 cycle).
If Index 44h[31] = 1, Format 1. Then the bits in this register allow independent control of command and data. Format 1 settings for: This PIO format must be used if the Fast-PCI clock is higher than 33 MHz. Format 1 settings for a Fast-PCI clock frequency of 33.3 MHz: — PIO Mode 0 = 9172D132h — PIO Mode 1 = 21717121h — PIO Mode 2 = 00803020h — PIO Mode 3 = 20102010h — PIO Mode 4 = 00100010h Format 1 settings for a Fast-PCI clock frequency of 66.7 MHz: — PIO Mode 0 = F8E4F8E4h — PIO Mode 1 = 53F3F353h — PIO Mode 2 = 13F18141h — PIO Mode 3 = 42314231h — PIO Mode 4 = 11311131h **Note:** All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.	
31:28	**t2IC.** Command cycle recovery time (value + 1 cycle).
27:24	**t3C.** Command cycle IDE_IOW# data setup (value + 1 cycle).
23:20	**t2WC.** Command cycle IDE_IOW# pulse width minus t3 (value + 1 cycle).
19:16	**t1C.** Command cycle address setup time (value + 1 cycle).
15:12	**t2ID.** Data cycle recovery time (value + 1 cycle).
11:8	**t3D.** Data cycle IDE_IOW# data setup (value + 1 cycle).
7:4	**t2WD.** Data cycle IDE_IOW# pulse width minus t3 (value + 1 cycle).
3:0	**t1D.** Data cycle address Setup Time (value + 1 cycle).

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 44h-47h	**Channel 0 Drive 0 DMA Control Register (R/W)** — **Reset Value: 00077771h**
The structure of this register depends on the value of bit 20.	
If bit 20 = 0, Multiword DMA	
Settings for a Fast-PCI clock frequency of 33.3 MHz: — Multiword DMA Mode 0 = 00077771h — Multiword DMA Mode 1 = 00012121h — Multiword DMA Mode 2 = 00002020h Settings for a Fast-PCI clock frequency of 66.7 MHz: — Multiword DMA Mode 0 = 000FFFF3h — Multiword DMA Mode 1 = 00035352h — Multiword DMA Mode 2 = 00015151h	
Note:	All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.
31	**PIO Mode Format.** This bit sets the PIO mode format for all channels and drives. Bit 31 of Offsets 2Ch, 34h, and 3Ch are R/W, but have no function so are defined as reserved. 0: Format 0. 1 Format 1.
30:21	**Reserved.** Must be set to 0.
20	**DMA Select.** Selects type of DMA operation. 0: Multiword DMA
19:16	**tKR.** IDE_IOR# recovery time (4-bit) (value + 1 cycle).
15:12	**tDR.** IDE_IOR# pulse width (value + 1 cycle).
11:8	**tKW**. IDE_IOW# recovery time (4-bit) (value + 1 cycle).
7:4	**tDW**. IDE_IOW# pulse width (value + 1 cycle).
3:0	**tM.** IDE_CS[1:0]# to IDE_IOR#/IOW# setup; IDE_CS[1:0]# setup to IDE_DACK0#/DACK1#.
If bit 20 = 1, UltraDMA	
Settings for a Fast-PCI clock frequency of 33.3 MHz: — UltraDMA Mode 0 = 00921250h — UltraDMA Mode 1 = 00911140h — UltraDMA Mode 2 = 00911030h Settings for a Fast-PCI clock frequency of 66.7 MHz: — UltraDMA Mode 0 = 009436A1h — UltraDMA Mode 1 = 00933481h — UltraDMA Mode 2 = 00923261h	
Note:	All references to “cycle” in the following bit descriptions are to a Fast-PCI clock cycle.
31	**PIO Mode Format.** This bit sets the PIO mode format for all channels and drives. Bit 31 of Offsets 2Ch, 34h, and 3Ch are R/W, but have no function so are defined as reserved. 0: Format 0 1: Format 1
30:24	**Reserved.** Must be set to 0.
23:21	**BSIZE.** Input buffer threshold.
20	**DMA Select.** Selects type of DMA operation. 1: UltraDMA.
19:16	**tCRC.** CRC setup UDMA in IDE_DACK# (value + 1 cycle) (for host terminate CRC setup = tMLI + tSS).
15:12	**tSS.** UDMA out (value + 1 cycle).
11:8	**tCYC.** Data setup and cycle time UDMA out (value + 2 cycles).
7:4	**tRP.** Ready to pause time (value + 1 cycle). Note: tRFS + 1 tRP on next clock.
3:0	**tACK.** IDE_CS[1:0]# setup to IDE_DACK0#/DACK1# (value + 1 cycle).
Index 48h-4Bh	**Channel 0 Drive 1 PIO Register (R/W)** — **Reset Value: 00009172h**
Channel 0 Drive 1 Programmed I/O Control Register. See F2 Index 40h for bit descriptions.	
Index 4Ch-4Fh	**Channel 0 Drive 1 DMA Control Register (R/W)** — **Reset Value: 00077771h**
Channel 0 Drive 1 MDMA/UDMA Control Register. See F2 Index 44h for bit descriptions.	
Note:	The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.
Index 50h-53h	**Channel 1 Drive 0 PIO Register (R/W)** — **Reset Value: 00009172h**
Channel 1 Drive 0 Programmed I/O Control Register. See F2 Index 40h for bit descriptions.	

Core Logic Module (Continued)

Table 5-35. F2: PCI Header/Channels 0 and 1 Registers for IDE Controller Configuration (Continued)

Bit	Description
Index 54h-57h	**Channel 1 Drive 0 DMA Control Register (R/W)** — **Reset Value: 00077771h**
Channel 1 Drive 0 MDMA/UDMA Control Register. See F2 Index 44h for bit descriptions.	
Note: The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.	
Index 58h-5Bh	**Channel 1 Drive 1 PIO Register (R/W)** — **Reset Value: 00009172h**
Channel 1 Drive 1 Programmed I/O Control Register. See F2 Index 40h for bit descriptions.	
Index 5Ch-5Fh	**Channel 1 Drive 1 DMA Control Register (R/W)** — **Reset Value: 00077771h**
Channel 1 Drive 1 MDMA/UDMA Control Register. See F2 Index 44h for bit descriptions.	
Note: The PIO Mode format is selected in F2 Index 44h[31], bit 31 of this register is defined as reserved.	
Index 60h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.3.1 IDE Controller Support Registers

F2 Index 20h, Base Address Register 4 (F2BAR4), points to the base address of where the registers for IDE controller configuration are located. Table 5-36 gives the bit formats of the I/O mapped IDE Controller Configuration registers that are accessed through F2BAR4.

Table 5-36. F2BAR4+I/O Offset: IDE Controller Configuration Registers

Bit	Description
Offset 00h	**IDE Bus Master 0 Command Register — Primary (R/W)** **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** 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	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the bus master. 0: Disable master. 1: Enable master. Bus master operations can be halted by setting this bit to 0. Once an operation has been halted, it cannot be resumed. If this bit 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.
Offset 01h	**Not Used**
Offset 02h	**IDE Bus Master 0 Status Register — Primary (R/W)** **Reset Value: 00h**
7	**Simplex Mode. (Read Only)** Indicates if both the primary and secondary channel operate independently. 0: Yes. 1: No (simplex mode).
6	**Drive 1 DMA Enable.** When asserted, allows Drive 1 to perform DMA transfers. 0: Disable. 1: Enable.
5	**Drive 0 DMA Enable.** When asserted, allows Drive 0 to perform DMA transfers. 0: Disable. 1: Enable.
4:3	**Reserved.** Must be set to 0. Must return 0 on reads.
2	**Bus Master Interrupt.** Indicates if the bus master detected an interrupt. 0: No. 1: Yes. Write 1 to clear.
1	**Bus Master Error.** Indicates if the bus master detected an error during data transfer. 0: No. 1: Yes. Write 1 to clear.
0	**Bus Master Active.** Indicates if the bus master is active. 0: No. 1: Yes.
Offset 03h	**Not Used**
Offset 04h-07h	**IDE Bus Master 0 PRD Table Address — Primary (R/W)** **Reset Value: 00000000h**
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for IDE Bus Master 0. When written, this field 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 field (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-36. F2BAR4+I/O Offset: IDE Controller Configuration Registers (Continued)

Bit	Description
Offset 08h	**IDE Bus Master 1 Command Register — Secondary (R/W)** **Reset Value: 00h**
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the direction of bus master transfers. 0: PCI reads are performed. 1: PCI writes are performed. This bit should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the bus master. 0: Disable master. 1: Enable master. Bus master operations can be halted by setting this bit to 0. Once an operation has been halted, it cannot be resumed. If this bit 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.
Offset 09h	**Not Used**
Offset 0Ah	**IDE Bus Master 1 Status Register — Secondary (R/W)** **Reset Value: 00h**
7	**Reserved. (Read Only)**
6	**Drive 1 DMA Capable.** Allow Drive 1 to perform DMA transfers. 0: Disable. 1: Enable.
5	**Drive 0 DMA Capable.** Allow Drive 0 to perform DMA transfers. 0: Disable. 1: Enable.
4:3	**Reserved.** Must be set to 0. Must return 0 on reads.
2	**Bus Master Interrupt.** Indicates if the bus master detected an interrupt. 0: No. 1: Yes. Write 1 to clear.
1	**Bus Master Error.** Indicates if the bus master detected an error during data transfer. 0: No. 1: Yes. Write 1 to clear.
0	**Bus Master Active.** Indicates if the bus master is active. 0: No. 1: Yes.
Offset 0Bh	**Not Used**
Offset 0Ch-0Fh	**IDE Bus Master 1 PRD Table Address — Secondary (R/W)** **Reset Value: 00000000h**
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for IDE Bus Master 1. When written, this field points to the first entry in a PRD table. Once IDE Bus Master 1 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this field (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

5.4.4 XpressAUDIO Registers - Function 3

The register designated as Function 3 (F3) is used to configure the PCI portion of support hardware for the XpressAUDIO registers. The bit formats for the PCI Header registers are given in Table 5-37.

A Base Address register (F3BAR0), located in the PCI Header registers of F3, is used for pointing to the register space designated for support of XpressAUDIO, described later in this section.

Table 5-37. F3: PCI Header Registers for XpressAUDIO Audio Configuration

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0503h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:3	**Reserved. (Read Only)**
2	**Bus Master.** Allow the Core Logic module bus mastering capabilities. 0: Disable. 1: Enable. (Default) This bit must be set to 1.
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. This bit must be enabled to access memory offsets through F3BAR0 (See F3 Index 10h).
0	**Reserved. (Read Only)**
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 040100h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register - F3BAR0 (R/W)** — **Reset Value: 00000000h** This register sets the base address of the memory mapped audio interface control register block. This is a 128-byte block of registers used to control the audio FIFO and codec interface, as well as to support VSA SMIs. Bits [11:0] are read only (0000 0000 0000), indicating a 4 KB memory address range. Refer to Table 5-38 on page 232 for the XpressAUDIO configuration register bit formats and reset values.
31:12	**XpressAUDIO Interface Base Address.**
11:0	**Address Range. (Read Only)**
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0503h**
Index 30h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

5.4.4.1 XpressAUDIO Support Registers

F3 Index 10h, Base Address Register 0 (F3BAR0), points to the base address of where the registers for XpressAUDIO support are located. Table 5-38 gives the bit formats of the memory mapped XpressAUDIO configuration registers that are accessed through F3BAR0.

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers

Bit	Description
Offset 00h-03h	**Codec GPIO Status Register (R/W)** — **Reset Value: 00000000h**
31	**Codec GPIO Interface.** 0: Disable. 1: Enable.
30	**Codec GPIO SMI.** When asserted, allows codec GPIO interrupt to generate an SMI. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[1].
29:21	**Reserved.** Must be set to 0.
20	**Codec GPIO Status Valid. (Read Only)** Indicates if the status read is valid. 0: Yes. 1: No.
19:0	**Codec GPIO Pin Status. (Read Only)** This field indicates the GPIO pin status that is received from the codec in slot 12 on the SDATA_IN (ball AF22) signal.
Offset 04h-07h	**Codec GPIO Control Register (R/W)** — **Reset Value: 00000000h**
31:20	**Reserved.** Must be set to 0.
19:0	**Codec GPIO Pin Data.** This field indicates the GPIO pin data that is sent to the codec in slot 12 on the SDATA_OUT (ball AD22) signal.
Offset 08h-0Bh	**Codec Status Register (R/W)** — **Reset Value: 00000000h**
31:24	**Codec Status Address. (Read Only)** Address of the register for which status is being returned. This address comes from slot 1 bits [19:12].
23	**Codec Serial INT Enable.** When asserted, allows codec serial interrupt to cause an SMI. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[1].
22	**SYNC Signal.** Sets SYNC (ball AE22) high or low. 0: Low. 1: High.
21	**Reserved.** Must be set to 0.
20	**Audio Bus Master 5 AC97 Slot Select.** Selects slot for Audio Bus Master 5 to receive data. 0: Slot 6. 1: Slot 11.
19	**Audio Bus Master 4 AC97 Slot Select.** Selects slot for Audio Bus Master 4 to transmit data. 0: Slot 6. 1: Slot 11.
18	**Reserved.** Must be set to 0.
17	**Status Tag. (Read Only)** The codec status data in bits [15:0] of this register is updated in the current AC97 frame. (codec ready, slot1 and slot2 bits in tag slot are all set in current AC97 frame). 0: Not new. 1: New, updated in current frame.
16	**Codec Status Valid. (Read Only)** Indicates if the status in bits [15:0] of this register is valid. This bit is high during slots 3 to 11 of the AC97 frame (i.e., for approximately 14.5 µs), for every frame. 0: No. 1: Yes.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
15:0	**Codec Status. (Read Only)** This is the codec status data that is received from the codec in slot 2 on SDATA_IN. Only bits [19:4] are used from slot 2. If this register is read with both bits 16 and 17, this register is set to 1, this field is updated in the current AC97 frame, and codec status data is valid. This bit field is updated only if the codec sent status data.
Offset 0Ch-0Fh	**Codec Command Register (R/W)** **Reset Value: 00000000h**
31:24	**Codec Command Address.** Address of the codec control register for which the command is being sent. This address goes in slot, 1 bits [19:12] on SDATA_OUT (ball AD22).
23:22	**Codec Communication.** Indicates the codec that the Core Logic module is communicating with. 00: Primary codec. 01: Secondary codec. 10: Third codec. 11: Fourth codec. Only 00 and 01 are valid settings for this bit field.
21:17	**Reserved.** Must be set to 0.
16	**Codec Command Valid. (Read Only)** Indicates if the command in bits [15:0] of this register is valid. 0: No. 1: Yes. This bit is set by hardware when a command is loaded. It remains set until the command has been sent to the codec. This bit is also set to 1 when written to.
15:0	**Codec Command.** This is the command being sent to the codec in bits [19:4] of slot 2 on SDATA_OUT (ball AD22).
Offset 10h-11h	**Second Level Audio SMI Status Register (RC)** **Reset Value: 0000h**
The bits in this register contain second level SMI status reporting. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. Reading this register clears the status bits at both the second and top levels. Note that bit 0 has a third level of status reporting which also must be "read to clear". A read-only “Mirror” version of this register exists at F3BAR0+I/O Memory Offset 12h. If the value of the register must be read without clearing the SMI source (and consequently deasserting SMI), F3BAR0+Memory Offset 12h can be read instead.	
15:8	**Reserved.** Must be set to 0.
7	**Audio Bus Master 5 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 5. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 5 is enabled (F3BAR0+Memory Offset 48h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 5 SMI Status Register (F3BAR0+Memory Offset 49h[0] = 1).
6	**Audio Bus Master 4 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 4. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 4 is enabled (F3BAR0+Memory Offset 40h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 4 SMI Status Register (F3BAR0+Memory Offset 41h[0] = 1).
5	**Audio Bus Master 3 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 3. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 3 is enabled (F3BAR0+Memory Offset 38h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 3 SMI Status Register (F3BAR0+Memory Offset 39h[0] = 1).
4	**Audio Bus Master 2 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 2. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 2 is enabled (F3BAR0+Memory Offset 30h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 2 SMI Status Register (F3BAR0+Memory Offset 31h[0] = 1).

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
3	**Audio Bus Master 1 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 1. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 1 is enabled (F3BAR0+Memory Offset 28h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 1 SMI Status Register (F3BAR0+Memory Offset 29h[0] = 1).
2	**Audio Bus Master 0 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 0. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 0 is enabled (F3BAR0+Memory Offset 20h[0] = 1). An SMI is then generated when the End of Page bit is set in the Audio Bus Master 0 SMI Status Register (F3BAR0+Memory Offset 21h[0] = 1).
1	**Codec Serial or GPIO Interrupt SMI Status.** Indicates if an SMI was caused by a serial or GPIO interrupt from codec. 0: No. 1: Yes. SMI generation enabling for codec serial interrupt: F3BAR0+Memory Offset 08h[23] = 1. SMI generation enabling for codec GPIO interrupt: F3BAR0+Memory Offset 00h[30] = 1.
0	**I/O Trap SMI Status.** Indicates if an SMI was caused by an I/O trap. 0: No. 1: Yes. The next level (third level) of SMI status reporting is at F3BAR0+Memory Offset 14h.

Offset 12h-13h **Second Level Audio SMI Status Mirror Register (RO)** **Reset Value: 0000h**

Note: The bits in this register contain second level SMI status reporting. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. Reading this register does not clear the status bits. See F3BAR0+Memory Offset 10h.

15:8	**Reserved.** Must be set to 0.
7	**Audio Bus Master 5 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 5. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 5 is enabled (F3BAR0+Memory Offset 48h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 49h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
6	**Audio Bus Master 4 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 4. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 4 is enabled (F3BAR0+Memory Offset 40h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 41h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
5	**Audio Bus Master 3 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 3. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 3 is enabled (F3BAR0+Memory Offset 38h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 39h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
4	**Audio Bus Master 2 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 2. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 2 is enabled (F3BAR0+Memory Offset 30h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 31h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
3	**Audio Bus Master 1 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 1. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 1 is enabled (F3BAR0+Memory Offset 28h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 29h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
2	**Audio Bus Master 0 SMI Status.** Indicates if an SMI was caused by an event occurring on Audio Bus Master 0. 0: No. 1: Yes. SMI generation is enabled when Audio Bus Master 0 is enabled (F3BAR0+Memory Offset 20h[0] = 1). An SMI is then generated when the End of Page bit is set in the SMI Status Register (F3BAR0+Memory Offset 21h[0] = 1). The End of Page bit must be cleared before this bit can be cleared.
1	**Codec Serial or GPIO Interrupt SMI Status.** Indicates if an SMI was caused by a serial or GPIO interrupt from codec. 0: No. 1: Yes. SMI generation enabling for codec serial interrupt: F3BAR0+Memory Offset 08h[23] = 1. SMI generation enabling for codec GPIO interrupt: F3BAR0+Memory Offset 00h[30] = 1.
0	**I/O Trap SMI Status.** Indicates if an SMI was caused by an I/O trap. 0: No. 1: Yes. The next level (third level) of SMI status reporting is at F3BAR0+Memory Offset 14h.

Offset 14h-17h **I/O Trap SMI and Fast Write Status Register (RO/RC)** **Reset Value: 00000000h**

Note: For the four SMI status bits (bits [13:10]), if the activity was a fast write to an even address, no SMI is generated regardless of the DMA, MPU, or Sound Card status. If the activity was a fast write to an odd address, an SMI is generated but bit 13 is set to a 1.

31:24	**Fast Path Write Even Access Data. (Read Only)** This bit field contains the data from the last Fast Path Write Even access. These bits change only on a fast write to an even address.
23:16	**Fast Path Write Odd Access Data. (Read Only)** This bit field contains the data from the last Fast Path Write Odd access. These bits change on a fast write to an odd address, and also on any non-fast write.
15	**Fast Write A1. (Read Only)** This bit contains the A1 value for the last Fast Write access.
14	**Read or Write I/O Access. (Read Only)** Indicates if the last trapped I/O access was a read or a write. 0: Read. 1: Write.
13	**Sound Card or FM Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the Sound Card or FM I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) Fast Path Write must be enabled, F3BAR0+Memory Offset 18h[11] = 1, for the SMI to be reported here. If Fast Path Write is disabled, the SMI is reported in bit 10 of this register. This is the third level of SMI status reporting. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[2].
12	**DMA Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the DMA I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) This is the third level of SMI status reporting. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[8:7].

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
11	**MPU Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the MPU I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) This is the third level of SMI status reporting. Second level of SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[6:5].
10	**Sound Card or FM Trap SMI Status. (Read to Clear)** Indicates if an SMI was caused by a trapped I/O access to the Sound Card or FM I/O Trap. 0: No. 1: Yes. (See the note included in the general description of this register above.) Fast Path Write must be disabled, F3BAR0+Memory Offset 18h[11] = 0, for the SMI to be reported here. If Fast Path Write is enabled, the SMI is reported in bit 13 of this register. This is the third level of SMI status reporting. Second level of SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Top level is reported at F1BAR0+I/O Offset 00h/02h[1]. SMI generation enabling is at F3BAR0+Memory Offset 18h[2].
9:0	**X-Bus Address (Read Only).** This bit field contains the captured ten bits of X-Bus address.
Offset 18h-19h	**I/O Trap SMI Enable Register (R/W)** — **Reset Value: 0000h**
15:12	**Reserved.** Must be set to 0.
11	**Fast Path Write Enable.** Fast Path Write (an SMI is not generated on certain writes to specified addresses). 0: Disable. 1: Enable. In Fast Path Write, the Core Logic module responds to writes to addresses: 388h, 38Ah, 38B, 2x0h, 2x2h, and 2x8h.
10:9	**Fast Read.** These two bits hold part of the response that the Core Logic module returns for reads to several I/O locations.
8	**High DMA I/O Trap.** If this bit is enabled and an access occurs at I/O Port C0h-DFh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[12].
7	**Low DMA I/O Trap.** If this bit is enabled and an access occurs at I/O Port 00h-0Fh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[12].
6	**High MPU I/O Trap.** If this bit is enabled and an access occurs at I/O Port 330h-331h, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[11].
5	**Low MPU I/O Trap.** If this bit is enabled and an access occurs at I/O Port 300h-301h, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[11].

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
4	**Fast Path Read Enable/SMI Disable.** When asserted, read Fast Path (an SMI is not generated on reads from specified addresses). 0: Disable. 1: Enable. In Fast Path Read the Core Logic module responds to reads of addresses: 388h-38Bh; 2x0h, 2x1, 2x2h, 2x3, 2x8 and 2x9h. If neither sound card nor FM I/O mapping is enabled, then status read trapping is not possible.
3	**FM I/O Trap.** If this bit is enabled and an access occurs at I/O Port 388h-38Bh, an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0].
2	**Sound Card I/O Trap.** If this bit is enabled and an access occurs in the address ranges selected by bits [1:0], an SMI is generated. 0: Disable. 1: Enable. Top level SMI status is reported at F1BAR0+I/O Offset 00h/02h[1]. Second level SMI status is reported at F3BAR0+Memory Offset 10h/12h[0]. Third level SMI status is reported at F3BAR0+Memory Offset 14h[10].
1:0	**Sound Card Address Range Select.** These bits select the address range for the sound card I/O trap. 00: I/O Port 220h-22Fh 01: I/O Port 240h-24Fh 10: I/O Port 260h-26Fh 11: I/O Port 280h-28Fh

Offset 1Ah-1Bh	Internal IRQ Enable Register (R/W)	Reset Value: 0000h
15	**IRQ15 Internal.** Configures IRQ15 for internal (software) or external (hardware) use. 0: External. 1: Internal.
14	**IRQ14 Internal.** Configures IRQ14 for internal (software) or external (hardware) use. 0: External. 1: Internal.
13	**Reserved.** Must be set to 0.
12	**IRQ12 Internal.** Configures IRQ12 for internal (software) or external (hardware) use. 0: External. 1: Internal.
11	**IRQ11 Internal.** Configures IRQ11 for internal (software) or external (hardware) use. 0: External. 1: Internal.
10	**IRQ10 Internal.** Configures IRQ10 for internal (software) or external (hardware) use. 0: External. 1: Internal.
9	**IRQ9 Internal.** Configures IRQ9 for internal (software) or external (hardware) use. 0: External. 1: Internal.
8	**Reserved.** Must be set to 0.
7	**IRQ7 Internal.** Configures IRQ7 for internal (software) or external (hardware) use. 0: External. 1: Internal.
6	**Reserved.** Must be set to 0.
5	**IRQ5 Internal.** Configures IRQ5 for internal (software) or external (hardware) use. 0: External. 1: Internal.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
4	**IRQ4 Internal.** Configures IRQ4 for internal (software) or external (hardware) use. 0: External. 1: Internal.
3	**IRQ3 Internal.** Configures IRQ3 for internal (software) or external (hardware) use. 0: External. 1: Internal.
2:0	**Reserved.** Must be set to 0.
Offset 1Ch-1Fh	**Internal IRQ Control Register (R/W)** **Reset Value: 00000000h** **Note:** Bits 31:16 of this register are Write Only. Reads to these bits always return a value of 0.
31	**Mask Internal IRQ15. (Write Only)** 0: Disable. 1: Enable.
30	**Mask Internal IRQ14. (Write Only)** 0: Disable. 1: Enable.
29	**Reserved. (Write Only)** Must be set to 0.
28	**Mask Internal IRQ12. (Write Only)** 0: Disable. 1: Enable.
27	**Mask Internal IRQ11. (Write Only)** 0: Disable. 1: Enable.
26	**Mask Internal IRQ10. (Write Only)** 0: Disable. 1: Enable.
25	**Mask Internal IRQ9. (Write Only)** 0: Disable. 1: Enable.
24	**Reserved. (Write Only)** Must be set to 0.
23	**Mask Internal IRQ7. (Write Only)** 0: Disable. 1: Enable.
22	**Reserved. (Write Only)** Must be set to 0.
21	**Mask Internal IRQ5. (Write Only)** 0: Disable. 1: Enable.
20	**Mask Internal IRQ4. (Write Only)** 0: Disable. 1: Enable.
19	**Mask Internal IRQ3. (Write Only)** 0: Disable. 1: Enable.
18:16	**Reserved. (Write Only)** Must be set to 0.
15	**Assert Masked Internal IRQ15.** 0: Disable. 1: Enable.
14	**Assert Masked Internal IRQ14.** 0: Disable. 1: Enable.

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
13	**Reserved.** Set to 0.
12	**Assert Masked Internal IRQ12.** 0: Disable. 1: Enable.
11	**Assert masked internal IRQ11.** 0: Disable. 1: Enable.
10	**Assert Masked Internal IRQ10.** 0: Disable. 1: Enable.
9	**Assert Masked Internal IRQ9.** 0: Disable. 1: Enable.
8	**Reserved.** Set to 0.
7	**Assert Masked Internal IRQ7.** 0: Disable. 1: Enable.
6	**Reserved.** Set to 0.
5	**Assert Masked Internal IRQ5.** 0: Disable. 1: Enable.
4	**Assert Masked Internal IRQ4.** 0: Disable. 1: Enable.
3	**Assert Masked Internal IRQ3.** 0: Disable. 1: Enable.
2:0	**Reserved.** Must be set to 0.

Offset 20h	**Audio Bus Master 0 Command Register (R/W)**	**Reset Value: 00h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.		
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.	
3	**Read or Write Control.** Sets the transfer direction of the Audio Bus Master. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read), and should not be changed when the bus master is active.	
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.	
0	**Bus Master Control.** Controls the state of the Audio Bus Master. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must either be paused, or reach EOT. Writing 0 to this bit while the bus master is operating may result in unpredictable behavior (and may crash the bus master state machine). The only recovery from such unpredictable behavior is a PCI reset.	

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 21h	**Audio Bus Master 0 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 22h-23h	**Not Used**
Offset 24h-27h	**Audio Bus Master 0 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 0: Output to codec; 32-bit; Left and Right Channels; Slots 3 and 4.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 0. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 0 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note:	The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)
Offset 28h	**Audio Bus Master 1 Command Register (R/W)** **Reset Value: 00h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 1. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 1. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing this bit to 0, the bus master must be either paused or reached EOT. Writing this bit to 0 while the bus master is operating results in unpredictable behavior (and may cause a crash of the bus master state machine). The only recovery from this condition is a PCI reset.

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 29h	**Audio Bus Master 1 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 2Ah-2Bh	**Not Used**
Offset 2Ch-2Fh	**Audio Bus Master 1 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 1: Input from codec; 32-Bit; Left and Right Channels; Slots 3 and 4.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field is a PRD table pointer for Audio Bus Master 1. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 1 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 30h	**Audio Bus Master 2 Command Register (R/W)** **Reset Value: 00h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the transfer direction of Audio Bus Master 2. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 2. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or reached EOT. Writing 0 to this bit while the bus master is operating results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 31h	**Audio Bus Master 2 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
7:2	**Reserved**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software has cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software has cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 32h-33h	**Not Used** — **Reset Value: 00h**
Offset 34h-37h	**Audio Bus Master 2 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 2: Output to codec; 16-Bit; Slot 5.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 2. When written, this field points to the first entry in a PRD table. Once Audio Bus Master 2 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 38h	**Audio Bus Master 3 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Sets the transfer direction of Audio Bus Master 3. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 3. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 39h	**Audio Bus Master 3 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 3Ah-3Bh	**Not Used**
Offset 3Ch-3Fh	**Audio Bus Master 3 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 3: Input from codec; 16-Bit; Slot 5.	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains is a PRD table pointer for Audio Bus Master 3. When written, this field points to the first entry in a PRD table. Once Audio Bus Master 3 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 40h	**Audio Bus Master 4 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 4. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 0 (read) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 4. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating, results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 41h	**Audio Bus Master 4 SMI Status Register (RC)** — **Reset Value: 00h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
7:2	**Reserved.**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 42h-43h	**Not Used**
Offset 44h-47h	**Audio Bus Master 4 PRD Table Address (R/W)** — **Reset Value: 00000000h**
Audio Bus Master 4: Output to codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[19] selects slot).	
31:2	**Pointer to the Physical Region Descriptor Table.** This register is a PRD table pointer for Audio Bus Master 4. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 4 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.
Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs. DWORD 0: [31:0] = Memory Region Physical Base Address DWORD 1: 31 = End of Table Flag 30 = End of Page Flag 29 = Loop Flag (JMP) [28:16] = Reserved (0) [15:0] = Byte Count of the Region (Size)	
Offset 48h	**Audio Bus Master 5 Command Register (R/W)** — **Reset Value: 00h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
7:4	**Reserved.** Must be set to 0. Must return 0 on reads.
3	**Read or Write Control.** Set the transfer direction of Audio Bus Master 5. 0: PCI reads are performed. 1: PCI writes are performed. This bit must be set to 1 (write) and should not be changed when the bus master is active.
2:1	**Reserved.** Must be set to 0. Must return 0 on reads.
0	**Bus Master Control.** Controls the state of the Audio Bus Master 5. 0: Disable. 1: Enable. Setting this bit to 1 enables the bus master to begin data transfers. When writing 0 to this bit, the bus master must be either paused or have reached EOT. Writing 0 to this bit while the bus master is operating, results in unpredictable behavior (and may crash the bus master state machine). The only recovery from this condition is a PCI reset.

Core Logic Module (Continued)

Table 5-38. F3BAR0+Memory Offset: XpressAUDIO Configuration Registers (Continued)

Bit	Description
Offset 49h	**Audio Bus Master 5 SMI Status Register (RC)** **Reset Value: 00h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
7:2	**Reserved**
1	**Bus Master Error.** Indicates if hardware encountered a second EOP before software cleared the first. 0: No. 1: Yes. If hardware encounters a second EOP (end of page) before software cleared the first, it causes the bus master to pause until this register is read to clear the error.
0	**End of Page.** Indicates if the Bus master transferred data which is marked by the EOP bit in the PRD table (bit 30). 0: No. 1: Yes.
Offset 4Ah-4Bh	**Not Used**
Offset 4Ch-4Fh	**Audio Bus Master 5 PRD Table Address (R/W)** **Reset Value: 00000000h**
Audio Bus Master 5: Input from codec; 16-Bit; Slot 6 or 11 (F3BAR0+Memory Offset 08h[20] selects slot).	
31:2	**Pointer to the Physical Region Descriptor Table.** This bit field contains a PRD table pointer for Audio Bus Master 5. When written, this register points to the first entry in a PRD table. Once Audio Bus Master 5 is enabled (Command Register bit 0 = 1), it loads the pointer and updates this register (by adding 08h) so that it points to the next PRD. When read, this register points to the next PRD.
1:0	**Reserved.** Must be set to 0.

Note: The Physical Region Descriptor (PRD) table consists of one or more entries - each describing a memory region to or from which data is to be transferred. Each entry consists of two DWORDs.

DWORD 0:	[31:0]	= Memory Region Physical Base Address
DWORD 1:	31	= End of Table Flag
	30	= End of Page Flag
	29	= Loop Flag (JMP)
	[28:16]	= Reserved (0)
	[15:0]	= Byte Count of the Region (Size)

Core Logic Module (Continued)

5.4.5 X-Bus Expansion Interface - Function 5

The register space designated as Function 5 (F5) is used to configure the PCI portion of support hardware for accessing the X-Bus Expansion support registers. The bit formats for the PCI Header Registers are given in Table 5-39.

Located in the PCI Header Registers of F5 are six Base Address Registers (F5BARx) used for pointing to the register spaces designated for X-Bus Expansion support, described later in this section.

Table 5-39. F5: PCI Header Registers for X-Bus Expansion

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 100Bh**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: 0515h**
Index 04h-05h	**PCI Command Register (R/W)** — **Reset Value: 0000h**
15:2	**Reserved. (Read Only)**
1	**Memory Space.** Allow the Core Logic module to respond to memory cycles from the PCI bus. 0: Disable. 1: Enable. If F5BAR0, F5BAR1, F5BAR2, F5BAR3, F5BAR4, and F5BAR5 (F5 Index 10h, 14h, 18h, 1Ch, 20h, and 24h) are defined as allowing access to memory mapped registers, this bit must be set to 1. BAR configuration is programmed through the corresponding mask register (see F5 Index 40h, 44h, 48h, 4Ch, 50h, and 54h)
0	**I/O Space.** Allow the Core Logic module to respond to I/O cycle from the PCI bus. 0: Disable. 1: Enable. If F5BAR0, F5BAR1, F5BAR2, F5BAR3, F5BAR4, and F5BAR5 (F5 Index 10h, 14h, 18h, 1Ch, 20h, and 24h) are defined as allowing access to I/O mapped registers, this bit must be set to 1. BAR configuration is programmed through the corresponding mask register (see F5 Index 40h, 44h, 48h, 4Ch, 50h, and 54h)
Index 06h-07h	**PCI Status Register (RO)** — **Reset Value: 0280h**
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 00h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 068000h**
Index 0Ch	**PCI Cache Line Size Register (RO)** — **Reset Value: 00h**
Index 0Dh	**PCI Latency Timer Register (RO)** — **Reset Value: 00h**
Index 0Eh	**PCI Header Type (RO)** — **Reset Value: 00h**
Index 0Fh	**PCI BIST Register (RO)** — **Reset Value: 00h**
Index 10h-13h	**Base Address Register 0 - F5BAR0 (R/W)** — **Reset Value: 00000000h**
X-Bus Expansion Address Space. This register allows PCI access to I/O mapped X-Bus Expansion support registers. Bits [5:0] must be set to 000001, indicating a 64-byte aligned I/O address space. Refer to Table 5-40 on page 250 for the X-Bus Expansion configuration register bit formats and reset values. **Note:** The size and type of accessed offsets can be reprogrammed through F5BAR0 Mask Register (F5 Index 40h).	
31:6	**X-Bus Expansion Base Address.**
5:0	**Address Range.** This bit field must be set to 000001 for this register to operate correctly.
Index 14h-17h	**Base Address Register 1 - F5BAR1 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR1 Mask Register (F5 Index 44h)	
Index 18h-1Bh	**Base Address Register 2 - F5BAR2 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR1 Mask Register (F5 Index 48h)	
Index 1Ch-1Fh	**Base Address Register 3 - F5BAR3 (R/W)** — **Reset Value: 00000000h**
Reserved. Reserved for possible future use by the Core Logic module. Configuration of this register is programmed through the F5BAR3 Mask Register (F5 Index 4Ch).	

Core Logic Module (Continued)

Table 5-39. F5: PCI Header Registers for X-Bus Expansion (Continued)

Bit	Description
Index 20h-23h	**Base Address Register 4 - F5BAR4 (R/W)** — **Reset Value: 00000000h**
F5BAR4CS# Address Space. This register allows PCI access to memory mapped devices on the Sub-ISA. Bit 0 must be set to 0, indicating memory address space. Access to maximum address space of 16 MB is supported. Configuration of this register is programmed through the F5BAR4 Mask Register (F5 Index 50h).	
31:24	**F5BAR4CS# Base Address.**
23:4	**F5BAR4CS# Base Address and Address Range.**
3	**Prefetchable.**
2:1	**Type.**
0	**Memory Space Indicator.** Must be set to 0.
Index 24h-27h	**Base Address Register 5 - F5BAR5 (R/W)** — **Reset Value: 00000000h**
F5BAR5CS# Address Space. This register allows PCI access to memory mapped devices on the Sub-ISA. Bit 0 must be set to 0, indicating memory address space. Access to maximum address space of 16 MB is supported. Configuration of this register is programmed through the F5BAR5 Mask Register (F5 Index 54h).	
31:24	**F5BAR5CS# Base Address.**
23:4	**F5BAR5CS# Base Address and Address Range.**
3	**Prefetchable.**
2:1	**Type.**
0	**Memory Space Indicator.** Must be set to 0.
Index 28h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 100Bh**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: 0505h**
Index 30h-3Fh	**Reserved** — **Reset Value: 00h**
Index 40h-43h	**F5BAR0 Mask Address Register (R/W)** — **Reset Value: FFFFFFC1h**
To use F5BAR0, the mask register should be programmed first. The mask register defines the size of F5BAR0 and whether the accessed offset registers are memory or I/O mapped. **Note:** Whenever a value is written to this mask register, F5BAR0 must also be written (even if the value for F5BAR0 has not changed).	
Memory Base Address Register (Bit 0 = 0)	
31:4	**Address Mask.** Determines the size of the BAR. — Every bit that is a 1 is programmable in the BAR. — Every bit that is a 0 is fixed 0 in the BAR. Since the address mask goes down to bit 4, the smallest memory region is 16 bytes, however, the PCI specification suggests not using less than a 4 KB address range.
3	**Prefetchable.** Indicates whether or not the data in memory is prefetchable. This bit should be set to 1 only if all the following are true: — There are no side-effects from reads (i.e., the data at the location is not changed as a result of the read). — The device returns all bytes regardless of the byte enables. — Host bridges can merge processor writes into this range without causing errors. — The memory is not cached from the host processor. 0: Data is not prefetchable. This value is recommended if one or more of the above listed conditions is not true. 1: Data is prefetchable.
2:1	**Type.** 00: Located anywhere in the 32-bit address space 01: Located below 1 MB 10: Located anywhere in the 64-bit address space 11: Reserved
0	**This bit must be set to 0, to indicate memory base address register.**

Core Logic Module (Continued)

Table 5-39. F5: PCI Header Registers for X-Bus Expansion (Continued)

Bit	Description
I/O Base Address Register (Bit 0 = 1)	
31:2	**Address Mask.** Determines the size of the BAR. — Every bit that is a 1 is programmable in the BAR. — Every bit that is a 0 is fixed 0 in the BAR. Since the address mask goes down to bit 2, the smallest I/O region is 4 bytes, however, the PCI Specification suggests not using less than a 4 KB address range.
1	**Reserved.** Must be set to 0.
0	**This bit must be set to 1, to indicate an I/O base address register.**
Index 44h-47h	**F5BAR1 Mask Address Register (R/W)** **Reset Value: 00000000h**
To use F5BAR1, the mask register should be programmed first. The mask register defines the size of F5BAR1 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR1 must also be written (even if the value for F5BAR1 has not changed).	
Index 48h-4Bh	**F5BAR2 Mask Address Register (R/W)** **Reset Value: 00000000h**
To use F5BAR2, the mask register should be programmed first. The mask register defines the size of F5BAR2 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR2 must also be written (even if the value for F5BAR2 has not changed).	
Index 4Ch-4Fh	**F5BAR3 Mask Address Register (R/W)** **Reset Value: 00000000h**
To use F5BAR3, the mask register should be programmed first. The mask register defines the size of F5BAR3 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR3 must also be written (even if the value for F5BAR3 has not changed).	
Index 50h-53h	**F5BAR4 Mask Address Register (R/W)** **Reset Value: 00000000h**
To use F5BAR4, the mask register should be programmed first. The mask register defines the size of F5BAR4 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR4 must also be written (even if the value for F5BAR4 has not changed).	
31:4	**Address Mask.** Bits [31:24] must be set to FFh. Other bits should be set according to the specific system configuration.
3	**Prefetchable.** Should be set according to the specific system configuration.
2:1	**Type.** Should be set according to the specific system configuration.
0	**Memory Space Indicator.** Must be set to 0.
Index 54h-57h	**F5BAR5 Mask Address Register (R/W)** **Reset Value: 00000000h**
To use F5BAR5, the mask register should be programmed first. The mask register defines the size of F5BAR5 and whether the accessed offset registers are memory or I/O mapped. See F5 Index 40h (F5BAR0 Mask Address Register) above for bit descriptions. **Note:** Whenever a value is written to this mask register, F5BAR5 must also be written (even if the value for F5BAR5 has not changed).	
31:4	**Address Mask.** Bits [31:24] must be set to FFh. Other bits should be set according to the specific system configuration.
3	**Prefetchable.** Should be set according to the specific system configuration.
2:1	**Type.** Should be set according to the specific system configuration.
0	**Memory Space Indicator.** Must be set to 0.

Core Logic Module (Continued)

Table 5-39. F5: PCI Header Registers for X-Bus Expansion (Continued)

Bit	Description
Index 58h	**F5BARx Initialized Register (R/W)** — **Reset Value: 00h**
7:6	**Reserved.** Must be set to 0.
5	**F5BAR5 Initialized.** This bit indicates if F5BAR5 (F5 Index 24h) has been initialized. At reset this bit is cleared (0). Writing F5BAR5 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR5 is disabled until either this bit is set to 1 or F5BAR5 is written (which causes this bit to be set to 1).
4	**F5BAR4 Initialized.** This bit indicates if F5BAR4 (F5 Index 28h) has been initialized. At reset this bit is cleared (0). Writing F5BAR4 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR4 is disabled until either this bit is set to 1 or F5BAR4 is written (which causes this bit to be set to 1).
3	**F5BAR3 Initialized.** This bit indicates if F5BAR3 (F5 Index 1Ch) has been initialized. At reset this bit is cleared (0). Writing F5BAR3 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR3 is disabled until either this bit is set to 1 or F5BAR3 is written (which causes this bit to be set to 1).
2	**F5BAR2 Initialized.** This bit indicates if F5BAR2 (F5 Index 18h) has been initialized. At reset this bit is cleared (0). Writing F5BAR2 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR2 is disabled until either this bit is set to 1 or F5BAR2 is written (which causes this bit to be set to 1).
1	**F5BAR1 Initialized.** This bit indicates if F5BAR1 (F5 Index 14h) has been initialized. At reset this bit is cleared (0). Writing F5BAR1 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR1 is disabled until either this bit is set to 1 or F5BAR1 is written (which causes this bit to be set to 1).
0	**F5BAR0 Initialized.** This bit indicates if F5BAR0 (F5 Index 10h) has been initialized. At reset this bit is cleared (0). Writing F5BAR0 sets this bit to 1. If this bit is programmed to 0, the decoding of F5BAR0 is disabled until either this bit is set to 1 or F5BAR0 is written (which causes this bit to be set to 1).
Index 59h	**F5BARx Directed to Sub-ISA (R/W)** — **Reset: 00h**
7:6	**Reserved.** Must be set to 0.
5	**F5BAR5 Directed to Sub-ISA.** Enables F5BAR5CS# to Sub-ISA. 0: F5BAR5 address range is not directed to Sub-ISA. 1: F5BAR5 address range is directed to Sub-ISA.
4	**F5BAR4 Directed to Sub-ISA.** Enables F5BAR4CS# to Sub-ISA. 0: F5BAR4 address range is not directed to Sub-ISA. 1: F5BAR4 address range is directed to Sub-ISA.
3	**Reserved.** Must be set to 0.
2	**Reserved.** Must be set to 0.
1	**Reserved.** Must be set to 0.
0	**Reserved.** Must be set to 0.
Index 5Ah-5Fh	**Reserved** — **Reset Value: xxh**
Index 60h-63h	**Scratchpad: Usually used for Device Number (R/W)** — **Reset Value: 00000000h** BIOS writes a value, of the Device number. Expected value: 00001100h.
Index 64h-67h	**Scratchpad: Usually used for Configuration Block Address (R/W)** — **Reset Value: 00000000h** BIOS writes a value, of the Configuration Block Address.
Index 68h-FFh	**Reserved**

Core Logic Module (Continued)

5.4.5.1 X-Bus Expansion Support Registers

F5 Index 10h, Base Address Register 0 (F5BAR0) sets the base address that allows PCI access to additional I/O Control Support registers. Table 5-40 shows the support registers accessed through F5BAR0.

Table 5-40. F5BAR0+I/O Offset: X-Bus Expansion Registers

Offset 00h-03h	I/O Control Register 1 (RW)	Reset Value: 010C0007h
31:28	**Reserved.**	
27	**IO_ENABLE_SIO_IR (Enable Integrated SIO Infrared).** 0: Disable. 1: Enable.	
26:25	**IO_SIOCFG_IN (Integrated SIO Input Configuration)**. These two bits can be used to disable the integrated SIO totally or limit/control the base address. 00: Integrated SIO disable. 01: Integrated SIO configuration access disable. 10: Integrated SIO base address 02Eh/02Fh enable. 11: Integrated SIO base address 015Ch/015Dh enable.	
24	**IO_ENABLE_SIO_DRIVING_ISA_BUS (Enable Integrated SIO ISA Bus Control)**. Allow the integrated SIO to drive the internal ISA bus. 0: Disable. 1: Enable. (Default)	
23:21	**Reserved**. Set to 0.	
20	**IO_USB_SMI_PWM_EN (USB SMI Configuration)**. Route USB-generated SMI to SMI Status Register in F1BAR0+I/O Offset 00h/02h[14]. 0: Disable. 1: Enable.	
19	**IO_USB_SMI_EN (USB SMI Configuration)**. Allow USB-generated SMIs. 0: Disable. 1: Enable. If bits 10 and 20 are enabled, the SMI generated by the USB is reported via the Top Level SMI status register at F1BAR0+I/O Offset 00h/02h[14]. If only bit 19 is enabled, the USB can generate an SMI but there is no status reporting.	
18	**IO_USB_PCI_EN (USB).** Enables USB ports. 0: Disable. 1: Enable.	
17:0	**Reserved**.	

Offset 04h-07h	I/O Control Register 2 (R/W)	Reset Value: 00000002h
31:2	**Reserved.** Write as read.	
1	**Reserved.** Must be set to 0. **Note:**	
0	**IO_STRAP_IDSEL_SELECT (IDSEL Strap Override).** 0: IDSEL: AD28 for Chipset Register Space (F0-F3, F5), AD29 for USB Register Space (PCIUSB). 1: IDSEL: AD26 for Chipset Register Space (F0-F3, F5), AD27 for USB Register Space (PCIUSB).	

Offset 08h-0Bh	I/O Control Register 3 (R/W)	Reset Value 00009000h
31:16	**Reserved**. Write as read.	
15:13	**IO_USB_XCVR_VADJ (USB Voltage Adjustment Connection).** These bits connect to the voltage adjustment interface on the three USB transceivers. Default = 100.	
12:8	**IO_USB_XCVT_CADJ (USB Current Adjustment)**. These bits connect to the current adjustment interface on the three USB transceivers. Default = 10000.	
7	**IO_TEST_PORT_EN (Debug Test Port Enable)**. 0: Disable. 1: Enable.	
6:0	**IO_TEST_PORT_REG (Debug Port Pointer).** These bits are used to point to the 16-bit slice of the test port bus.	

Core Logic Module (Continued)

5.4.6 USB Controller Registers - PCIUSB

The registers designated as PCIUSB are 32-bit registers decoded from the PCI address bits [7:2] and C/BE[3:0]#, when IDSEL is high, AD[10:8] select the appropriate function, and AD[1:0] are 00.

The PCI Configuration registers are listed in Table 5-41. They can be accessed as any number of bytes within a single 32-bit aligned unit. They are selected by the PCI-standard Index and Byte-Enable method.

In the PCI Configuration space, there is one Base Address Register (BAR), at Index 10h, which is used to map the USB Host Controller's operational register set into a 4K memory space. Once the BAR register has been initialized, and the PCI Command register at Index 04h has been set to enable the Memory space decoder, these “USB Controller” registers are accessible.

The memory-mapped USB Controller registers are listed in Table 5-42. They follow the Open Host Controller Interface (OHCI) specification. Registers marked as “Reserved”, and reserved bits within a register, should not be changed by software.

Table 5-41. PCIUSB: USB PCI Configuration Registers

Bit	Description
Index 00h-01h	**Vendor Identification Register (RO)** — **Reset Value: 0E11h**
Index 02h-03h	**Device Identification Register (RO)** — **Reset Value: A0F8h**
Index 04h-05h	**Command Register (R/W)** — **Reset Value: 0000h**
15:10	**Reserved.** Must be set to 0.
9	**Fast Back-to-Back Enable. (Read Only)** USB only acts as a master to a single device, so this functionality is not needed. It is always disabled (i.e., this bit must always be set to 0).
8	**SERR#.** When this bit is enabled, USB asserts SERR# when it detects an address parity error. 0: Disable. 1: Enable.
7	**Wait Cycle Control.** USB does not need to insert a wait state between the address and data on the AD lines. It is always disabled (i.e., this bit is set to 0).
6	**Parity Error.** USB asserts PERR# when it is the agent receiving data and it detects a data parity error. 0: Disable. 1: Enable.
5	**VGA Palette Snoop Enable. (Read Only)** USB does not support this function. It is always disabled (i.e., this bit is set to 0).
4	**Memory Write and Invalidate.** Allow USB to run Memory Write and Invalidate commands. 0: Disable. 1: Enable. The Memory Write and Invalidate Command only occurs if the cache-line size is set to 32 bytes and the memory write is exactly one cache line. This bit must be set to 0.
3	**Special Cycles.** USB does not run special cycles on PCI. It is always disabled (i.e., this bit is set to 0).
2	**PCI Master Enable**. Allow the USB to run PCI master cycles. 0: Disable. 1: Enable.
1	**Memory Space.** Allow the USB to respond as a target to memory cycles from the PCI bus. 0: Disable. 1: Enable.
0	**I/O Space.** Allow the USB to respond as a target to I/O cycles from the PCI bus. 0: Disable. 1: Enable.

Core Logic Module (Continued)

Table 5-41. PCIUSB: USB PCI Configuration Registers (Continued)

Bit	Description
Index 06h-07h	**Status Register (R/W)** — **Reset Value: 0280h**
The PCI specification defines this register to record status information for PCI related events. This is a read/write register. However, writes can only reset bits. A bit is reset whenever the register is written and the data in the corresponding bit location is a 1.	
15	**Detected Parity Error**. This bit is set to 1 whenever the USB detects a parity error, even if the Parity Error (Response) Detection Enable Bit (Command Register, bit 6) is disabled. Write 1 to clear.
14	**SERR# Status.** This bit is set whenever the USB detects a PCI address error. Write 1 to clear.
13	**Received Master Abort Status.** This bit is set when the USB, acting as a PCI master, aborts a PCI bus memory cycle. Write 1 to clear.
12	**Received Target Abort Status.** 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	**Signaled Target Abort Status.** This bit is set whenever the USB signals a target abort. Write 1 to clear.
10:9	**DEVSEL# Timing. (Read Only)** 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 01b.
8	**Data Parity Reported. (Read Only)** This bit is set to 1 if the Parity Error Response bit (Command Register bit 6) is set, and the USB detects PERR# asserted while acting as PCI master (whether or not PERR# was driven by USB).
7	**Fast Back-to-Back Capable.** The USB supports fast back-to-back transactions when the transactions are not to the same agent. This bit is always 1.
6:0	**Reserved**. Must be set to 0.
Index 08h	**Device Revision ID Register (RO)** — **Reset Value: 08h**
Index 09h-0Bh	**PCI Class Code Register (RO)** — **Reset Value: 0C0310h**
This register identifies the generic function of the USB the specific register level programming interface. The Base Class is 0Ch (Serial Bus Controller). The Sub Class is 03h (Universal Serial Bus). The Programming Interface is 10h (OpenHCI).	
Index 0Ch	**Cache Line Size Register (R/W)** — **Reset Value: 00h**
This register identifies the system cache-line size in units of 32-bit WORDs. The USB only stores the value of bit 3 in this register since the cache-line size of 32 bytes is the only value applicable to the design. Any value other than 08h written to this register is read back as 00h.	
Index 0Dh	**Latency Timer Register (R/W)** — **Reset Value: 00h**
This register identifies the value of the latency timer in PCI clocks for PCI bus master cycles. Bits [2:0] of this register are always set to 0.	
Index 0Eh	**Header Type Register (RO)** — **Reset Value: 00h**
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.	
Index 0Fh	**BIST Register (RO)** — **Reset Value: 00h**
This register identifies the control and status of Built-In Self-Test (BIST). The USB does not implement BIST, so this register is read only.	
Index 10h-13h	**Base Address Register- USB_BAR0 (R/W)** — **Reset Value: 00000000h**
31:12	**Base Address.** POST writes the value of the memory base address to this register.
11:4	**Always 0.** Indicates that a 4 KB address range is requested.
3	**Always 0.** Indicates that there is no support for prefetchable memory.
2:1	**Always 0.** Indicates that the base register is 32-bits wide and can be placed anywhere in 32-bit memory space.
0	**Always 0.** Indicates that the operational registers are mapped into memory space.

Core Logic Module (Continued)

Table 5-41. PCIUSB: USB PCI Configuration Registers (Continued)

Bit	Description
Index 14h-2Bh	**Reserved** — **Reset Value: 00h**
Index 2Ch-2Dh	**Subsystem Vendor ID (RO)** — **Reset Value: 0E11h**
Index 2Eh-2Fh	**Subsystem ID (RO)** — **Reset Value: A0F8h**
Index 30h-3Bh	**Reserved** — **Reset Value: 00h**
Index 3Ch	**Interrupt Line Register (R/W)** — **Reset Value: 00h** This register identifies the system interrupt controllers to which the device's interrupt pin is connected. The value of this register is used by device drivers and has no direct meaning to USB.
Index 3Dh	**Interrupt Pin Register (R/W)** — **Reset Value: 01h** This register selects which interrupt pin the device uses. USB uses INTA# after reset. INTB#, INTC# or INTD# can be selected by writing 2, 3 or 4, respectively.
Index 3Eh	**Min. Grant Register (RO)** — **Reset Value: 00h** This register specifies how long a burst is needed by the USB, assuming a clock rate of 33 MHz. The value in this register specifies a period of time in units of 1/4 microsecond.
Index 3Fh	**Max. Latency Register (RO)** — **Reset Value: 50h** This register specifies how often (in units of 1/4 microsecond) the USB needs access to the PCI bus assuming a clock rate of 33 MHz.
Index 40h-43h	**ASIC Test Mode Enable Register (R/W)** — **Reset Value: 000F0000h** Used for internal debug and test purposes only.
Index 44h	**ASIC Operational Mode Enable Register (R/W)** — **Reset Value: 00h**
7:1	**Write Only.** Read as 0s.
0	**Data Buffer Region 16** 0: The size of the region for the data buffer is 32 bytes. 1: The size of the region for the data buffer is 16 bytes.
Index 45h-FFh	**Reserved** — **Reset Value: 00h**

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers

Bit	Description
Offset 00h-03h	**HcRevision Register (RO)** — **Reset Value = 00000110h**
31:8	**Reserved.** Read/Write 0s.
7:0	**Revision (Read Only).** Indicates the Open HCI Specification revision number implemented by the Hardware. USB supports 1.0 specification. (X.Y = XYh).
Offset 04h-07h	**HcControl Register (R/W)** — **Reset Value = 00000000h**
31:11	**Reserved.** Read/Write 0s.
10	**RemoteWakeupConnectedEnable.** If a remote wakeup signal is supported, this bit enables that operation. Since there is no remote wakeup signal supported, this bit is ignored.
9	**RemoteWakeupConnected (Read Only).** This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to 0.
8	**InterruptRouting.** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
7:6	**HostControllerFunctionalState.** 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: 00: UsbReset. 01: UsbResume. 10: UsbOperational. 11: UsbSuspend.
5	**BulkListEnable.** When set, this bit enables processing of the Bulk list.
4	**ControlListEnable.** When set, this bit enables processing of the Control list.
3	**IsochronousEnable.** 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.
2	**PeriodicListEnable.** 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.
1:0	**ControlBulkServiceRatio.** 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).
Offset 08h-0Bh	**HcCommandStatus Register (R/W)** — **Reset Value = 00000000h**
31:18	**Reserved.** Read/Write 0s.
17:16	**ScheduleOverrunCount.** This field increments every time the SchedulingOverrun bit in HcInterruptStatus is set. The count wraps from 11 to 00.
15:4	**Reserved.** Read/Write 0s.
3	**OwnershipChangeRequest.** When set by software, this bit sets the OwnershipChange field in HcInterruptStatus. The bit is cleared by software.
2	**BulkListFilled.** 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.** 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.** This bit is set to initiate a software reset. This bit is cleared by the HC upon completion of the reset operation.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 0Ch-0Fh	**HcInterruptStatus Register (R/W)** **Reset Value = 00000000h**
31	**Reserved.** Read/Write 0s.
30	**OwnershipChange.** This bit is set when the OwnershipChangeRequest bit of HcCommandStatus is set.
29:7	**Reserved.** Read/Write 0s.
6	**RootHubStatusChange.** This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed.
5	**FrameNumberOverflow.** Set when bit 15 of FrameNumber changes value.
4	**UnrecoverableError (Read Only).** This event is not implemented and is hard-coded to 0. Writes are ignored.
3	**ResumeDetected.** Set when HC detects resume signaling on a downstream port.
2	**StartOfFrame.** Set when the Frame Management block signals a Start of Frame event.
1	**WritebackDoneHead.** Set after the HC has written HcDoneHead to HccaDoneHead.
0	**SchedulingOverrun.** Set when the List Processor determines a Schedule Overrun has occurred.
Note:	All bits are set by hardware and cleared by software.
Offset 10h-13h	**HcInterruptEnable Register (R/W)** **Reset Value = 00000000h**
31	**MasterInterruptEnable.** This bit is a global interrupt enable. A write of 1 allows interrupts to be enabled via the specific enable bits listed above.
30	**OwnershipChangeEnable.** 0: Ignore. 1: Enable interrupt generation due to Ownership Change.
29:7	**Reserved.** Read/Write 0s.
6	**RootHubStatusChangeEnable.** 0: Ignore. 1: Enable interrupt generation due to Root Hub Status Change.
5	**FrameNumberOverflowEnable.** 0: Ignore. 1: Enable interrupt generation due to Frame Number Overflow.
4	**UnrecoverableErrorEnable.** This event is not implemented. All writes to this bit are ignored.
3	**ResumeDetectedEnable.** 0: Ignore. 1: Enable interrupt generation due to Resume Detected.
2	**StartOfFrameEnable.** 0: Ignore. 1: Enable interrupt generation due to Start of Frame.
1	**WritebackDoneHeadEnable.** 0: Ignore. 1: Enable interrupt generation due to Writeback Done Head.
0	**SchedulingOverrunEnable.** 0: Ignore. 1: Enable interrupt generation due to Scheduling Overrun.
Note:	Writing a 1 to a bit in this register sets the corresponding bit, while writing a 0 leaves the bit unchanged.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 14h-17h	**HcInterruptDisable Register (R/W)** — **Reset Value = 00000000h**
31	**MasterInterruptEnable.** Global interrupt disable. A write of 1 disables all interrupts.
30	**OwnershipChangeEnable.** 0: Ignore. 1: Disable interrupt generation due to Ownership Change.
29:7	**Reserved.** Read/Write 0s.
6	**RootHubStatusChangeEnable.** 0: Ignore. 1: Disable interrupt generation due to Root Hub Status Change.
5	**FrameNumberOverflowEnable.** 0: Ignore. 1: Disable interrupt generation due to Frame Number Overflow.
4	**UnrecoverableErrorEnable.** This event is not implemented. All writes to this bit will be ignored.
3	**ResumeDetectedEnable.** 0: Ignore. 1: Disable interrupt generation due to Resume Detected.
2	**StartOfFrameEnable.** 0: Ignore. 1: Disable interrupt generation due to Start of Frame.
1	**WritebackDoneHeadEnable.** 0: Ignore. 1: Disable interrupt generation due to Writeback Done Head.
0	**SchedulingOverrunEnable.** 0: Ignore. 1: Disable interrupt generation due to Scheduling Overrun.
Note:	Writing a 1 to a bit in this register clears the corresponding bit, while writing a 0 to a bit leaves the bit unchanged.
Offset 18h-1Bh	**HcHCCA Register (R/W)** — **Reset Value = 00000000h**
31:8	**HCCA.** Pointer to HCCA base address.
7:0	**Reserved.** Read/Write 0s.
Offset 1Ch-1Ch	**HcPeriodCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**PeriodCurrentED.** Pointer to the current Periodic List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 20h-23h	**HcControlHeadED Register (R/W)** — **Reset Value = 00000000h**
31:4	**ControlHeadED.** Pointer to the Control List Head ED.
3:0	**Reserved.** Read/Write 0s.
Offset 24h-27h	**HcControlCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**ControlCurrentED.** Pointer to the current Control List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 28h-2Bh	**HcBulkHeadED Register (R/W)** — **Reset Value = 00000000h**
31:4	**BulkHeadED.** Pointer to the Bulk List Head ED.
3:0	**Reserved.** Read/Write 0s.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 2Ch-2Fh	**HcBulkCurrentED Register (R/W)** — **Reset Value = 00000000h**
31:4	**BulkCurrentED.** Pointer to the current Bulk List ED.
3:0	**Reserved.** Read/Write 0s.
Offset 30h-33h	**HcDoneHead Register (R/W)** — **Reset Value = 00000000h**
31:4	**DoneHead.** Pointer to the current Done List Head ED.
3:0	**Reserved.** Read/Write 0s.
Offset 34h-37h	**HcFmInterval Register (R/W)** — **Reset Value = 00002EDFh**
31	**FrameIntervalToggle (Read Only).** This bit is toggled by HCD when it loads a new value into FrameInterval.
30:16	**FSLargestDataPacket (Read Only).** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.
15:14	**Reserved.** Read/Write 0s.
13:0	**FrameInterval.** 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.
Offset 38h-3Bh	**HcFrameRemaining Register (RO)** — **Reset Value = 00000000h**
31	**FrameRemainingToggle (Read Only).** Loaded with FrameIntervalToggle when FrameRemaining is loaded.
30:14	**Reserved.** Read 0s.
13:0	**FrameRemaining (Read Only).** 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.
Offset 3Ch-3Fh	**HcFmNumber Register (RO)** — **Reset Value = 00000000h**
31:16	**Reserved.** Read 0s.
15:0	**FrameNumber (Read Only).** This 16-bit incrementing counter field is incremented coincident with the loading of FrameRemaining. The count rolls over from FFFFh to 0h.
Offset 40h-43h	**HcPeriodicStart Register (R/W)** — **Reset Value = 00000000h**
31:14	**Reserved.** Read/Write 0s.
13:0	**PeriodicStart.** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.
Offset 44h-47h	**HcLSThreshold Register (R/W)** — **Reset Value = 00000628h**
31:12	**Reserved.** Read/Write 0s.
11:0	**LSThreshold.** 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.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 48h-4Bh	**HcRhDescriptorA Register (R/W)** — **Reset Value = 01000002h**
31:24	**PowerOnToPowerGoodTime.** 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	**Reserved.** Read/Write 0s.
12	**NoOverCurrentProtection.** 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.** This bit should be written 0 and is only valid when NoOverCurrentProtection is cleared. 0: Global Over-Current. 1: Individual Over-Current.
10	**DeviceType (Read Only).** USB is not a compound device.
9	**NoPowerSwitching.** 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.** This bit is only valid when NoPowerSwitching is cleared. This bit should be written 0. 0: Global Switching. 1: Individual Switching.
7:0	**NumberDownstreamPorts (Read Only).** USB supports three downstream ports.
Note:	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.
Offset 4Ch-4Fh	**HcRhDescriptorB Register (R/W)** — **Reset Value = 00000000h**
31:16	**PortPowerControlMask.** Global-power switching. This field is only valid if NoPowerSwitching is cleared and Power-SwitchingMode 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). 0: Device not removable. 1: Global-power mask. Port Bit relationship - Unimplemented ports are reserved, read/write 0. 0 = Reserved 1 = Port 1 2 = Port 2 ... 15 = Port 15
15:0	**DeviceRemoveable.** USB ports default to removable devices. 0: Device not removable. 1: Device removable. Port Bit relationship 0 = Reserved 1 = Port 1 2 = Port 2 ... 15 = Port 15 Unimplemented ports are reserved, read/write 0.
Note:	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.

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 50h-53h	**HcRhStatus Register (R/W)** — **Reset Value = 00000000h**
31	**ClearRemoteWakeupEnable (Write Only).** Writing a 1 to this bit clears DeviceRemoteWakeupEnable. Writing a 0 has no effect.
30:18	**Reserved.** Read/Write 0s.
17	**OverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
16	**Read: LocalPowerStatusChange.** Not supported. Always read 0. **Write: SetGlobalPower.** Write a 1 issues a SetGlobalPower command to the ports. Writing a 0 has no effect.
15	**Read: DeviceRemoteWakeupEnable.** This bit enables ports' ConnectStatusChange as a remote wakeup event. 0: Disabled. 1: Enabled. **Write: SetRemoteWakeupEnable.** Writing a 1 sets DeviceRemoteWakeupEnable. Writing a 0 has no effect.
14:2	**Reserved.** Read/Write 0s.
1	**OverCurrentIndicator.** This bit reflects the state of the OVRCUR pin. This field is only valid if NoOverCurrentProtection and OverCurrentProtectionMode are cleared. 0: No over-current condition. 1: Over-current condition.
0	**Read: LocalPowerStatus.** Not Supported. Always read 0. **Write: ClearGlobalPower.** Writing a 1 issues a ClearGlobalPower command to the ports. Writing a 0 has no effect.
Note:	This register is reset by the UsbReset state.
Offset 54h-57h	**HcRhPortStatus[1] Register (R/W)** — **Reset Value = 00000000h**
31:21	**Reserved.** Read/Write 0s.
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	**PortEnableStatusChange.** 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	**ConnectStatusChange.** 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	**Reserved.** Read/Write 0s.
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full Speed device. 1: Low Speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.
7:5	**Reserved.** Read/Write 0s.
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note: This register is reset by the UsbReset state.	

Offset 58h-5Bh	**HcRhPortStatus[2] Register (R/W)**	**Reset Value = 00000000h**
31:21	**Reserved.** Read/Write 0s.	
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.	
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.	
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.	
17	**PortEnableStatusChange.** 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.	

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
16	**ConnectStatusChange.** 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	**Reserved.** Read/Write 0s.
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full speed device. 1: Low speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.
7:5	**Reserved.** Read/Write 0s.
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note: This register is reset by the UsbReset state.	

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 5Ch-5Fh	**HcRhPortStatus[3] Register (R/W)** — **Reset Value = 00000000h**
31:21	**Reserved.** Read/Write 0s.
20	**PortResetStatusChange.** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	**PortOverCurrentIndicatorChange.** This bit is set when OverCurrentIndicator changes. Writing a 1 clears this bit. Writing a 0 has no effect.
18	**PortSuspendStatusChange.** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	**PortEnableStatusChange.** 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	**ConnectStatusChange.** 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	**Reserved.** Read/Write 0s.
9	**Read: LowSpeedDeviceAttached.** This bit defines the speed (and bud idle) of the attached device. It is only valid when CurrentConnectStatus is set. 0: Full speed device. 1: Low speed device. **Write: ClearPortPower.** Writing a 1 clears PortPowerStatus. Writing a 0 has no effect.
8	**Read: PortPowerStatus.** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. If NoPowerSwitching is set, this bit is always read as 1. **Write: SetPortPower.** Writing a 1 sets PortPowerStatus. Writing a 0 has no effect.
7:5	**Reserved.** Read/Write 0s.
4	**Read: PortResetStatus.** 0: Port reset signal is not active. 1: Port reset signal is active. **Write: SetPortReset.** Writing a 1 sets PortResetStatus. Writing a 0 has no effect.
3	**Read: PortOverCurrentIndicator.** This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition. 1: Over-current condition. **Write: ClearPortSuspend.** Writing a 1 initiates the selective resume sequence for the port. Writing a 0 has no effect.
2	**Read: PortSuspendStatus.** 0: Port is not suspended. 1: Port is selectively suspended. **Write: SetPortSuspend.** Writing a 1 sets PortSuspendStatus. Writing a 0 has no effect.
1	**Read: PortEnableStatus.** 0: Port disabled. 1: Port enabled. **Write: SetPortEnable.** Writing a 1 sets PortEnableStatus. Writing a 0 has no effect.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
0	**Read: CurrentConnectStatus.** 0: No device connected. 1: Device connected. If DeviceRemoveable is set (not removable) this bit is always 1. **Write: ClearPortEnable.** Writing 1 a clears PortEnableStatus. Writing a 0 has no effect.
Note:	This register is reset by the UsbReset state.

Offset 60h-9Fh	Reserved	Reset Value = xxh

Offset 100h-103h	HceControl Register (R/W)	Reset Value = 00000000h

31:9	**Reserved.** Read/Write 0s.
8	**A20State.** Indicates current state of Gate A20 on keyboard controller. Compared against value written to 60h when GateA20Sequence is active.
7	**IRQ12Active.** Indicates a positive transition on IRQ12 from keyboard controller occurred. Software writes this bit to 1 to clear it (set it to 0); a 0 write has no effect.
6	**IRQ1Active.** Indicates a positive transition on IRQ1 from keyboard controller occurred. Software writes this bit to 1 to clear it (set it to 0); a 0 write has no effect.
5	**GateA20Sequence.** Set by HC when a data value of D1h is written to I/O port 64h. Cleared by HC on write to I/O port 64h of any value other than D1h.
4	**ExternalIRQEn.** When set to 1, IRQ1 and IRQ12 from the keyboard controller cause an emulation interrupt. The function controlled by this bit is independent of the setting of the EmulationEnable bit in this register.
3	**IRQEn.** When set, the HC generates IRQ1 or IRQ12 as long as the OutputFull bit in HceStatus is set to 1. If the AuxOutputFull bit of HceStatus is 0, IRQ1 is generated: if 1, then an IRQ12 is generated.
2	**CharacterPending.** When set, an emulation interrupt will be generated when the OutputFull bit of the HceStatus register is set to 0.
1	**EmulationInterrupt (Read Only).** This bit is a static decode of the emulation interrupt condition.
0	**EmulationEnable.** When set to 1 the HC is enabled for legacy emulation and will decode accesses to I/O registers 60h and 64h and generate IRQ1 and/or IRQ12 when appropriate. The HC also generates an emulation interrupt at appropriate times to invoke the emulation software.
Note:	This register is used to enable and control the emulation hardware and report various status information.

Offset 104h-107h	HceInput Register (R/W)	Reset Value = 000000xxh

31:8	**Reserved.** Read/Write 0s.
7:0	**InputData.** This register holds data written to I/O ports 60h and 64h.
Note:	This register is the emulation side of the legacy Input Buffer register.

Offset 108h-10Bh	HceOutput Register (R/W)	Reset Value = 000000xxh

31:8	**Reserved.** Read/Write 0s.
7:0	**OutputData.** This register hosts data that is returned when an I/O read of port 60h is performed by application software.
Note:	This register is the emulation side of the legacy Output Buffer register where keyboard and mouse data is to be written by software.

Core Logic Module (Continued)

Table 5-42. USB_BAR+Memory Offset: USB Controller Registers (Continued)

Bit	Description
Offset 10Ch-10Fh	**HceStatus Register (R/W)** **Reset Value = 00000000h**
31:8	**Reserved.** Read/Write 0s.
7	**Parity.** Indicates parity error on keyboard/mouse data.
6	**Timeout.** Used to indicate a time-out
5	**AuxOutputFull.** IRQ12 is asserted whenever this bit is set to 1 and OutputFull is set to 1 and the IRQEn bit is set.
4	**Inhibit Switch.** This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.
3	**CmdData.** The HC will set this bit to 0 on an I/O write to port 60h and on an I/O write to port 64h the HC will set this bit to 1.
2	**Flag.** Nominally used as a system flag by software to indicate a warm or cold boot.
1	**InputFull.** Except for the case of a Gate A20 sequence, this bit is set to 1 on an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists.
0	**OutputFull.** The HC will set this bit to 0 on a read of I/O port 60h. If IRQEn is set and AuxOutputFull is set to 0 then an IRQ1 is generated as long as this bit is set to 1. If IRQEn is set and AuxOutputFull is set to 1 then and IRQ12 will be generated a long as this bit is set to 1. While this bit is 0 and CharacterPending in HceControl is set to 1, an emulation interrupt condition exists.
Note:	This register is the emulation side of the legacy Status register.

Core Logic Module (Continued)

5.4.7 ISA Legacy Register Space

The ISA Legacy registers reside in the ISA I/O address space in the address range from 000h to FFFh and are accessed through typical input/output instructions (i.e., CPU direct R/W) with the designated I/O port address and 8-bit data.

The bit formats for the ISA Legacy I/O Registers plus two chipset-specific configuration registers used for interrupt mapping in the Core Logic module are given in this section. The ISA Legacy registers are separated into the following categories:

- DMA Channel Control Registers, see Table 5-43
- DMA Page Registers, see Table 5-44
- Programmable Interval Timer Registers, see Table 5-45
- Programmable Interrupt Controller Registers, see Table 5-46
- Keyboard Controller Registers, see Table 5-47
- Real Time Clock Registers, see Table 5-48
- Miscellaneous Registers, see Table 5-49 (includes 4D0h and 4D1h Interrupt Edge/Level Select Registers)

Table 5-43. DMA Channel Control Registers

Bit	Description
I/O Port 000h	**DMA Channel 0 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 001h	**DMA Channel 0 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 002h	**DMA Channel 1 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 003h	**DMA Channel 1 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 004h	**DMA Channel 2 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 005h	**DMA Channel 2 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 006h	**DMA Channel 3 Address Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 007h	**DMA Channel 3 Transfer Count Register (R/W)**
Written as two successive bytes, byte 0, 1.	
I/O Port 008h (R/W)	
Read	**DMA Status Register, Channels 3:0**
7	**Channel 3 Request.** Indicates if a request is pending. 0: No. 1: Yes.
6	**Channel 2 Request.** Indicates if a request is pending. 0: No. 1: Yes.
5	**Channel 1 Request.** Indicates if a request is pending. 0: No. 1: Yes.
4	**Channel 0 Request.** Indicates if a request is pending. 0: No. 1: Yes.
3	**Channel 3 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
2	**Channel 2 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
1	**Channel 1 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
0	**Channel 0 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
Write	**DMA Command Register, Channels 3:0**
7	**DACK Sense.** 0: Active high. 1: Active low.
6	**DREQ Sense.** 0: Active high. 1: Active low.
5	**Write Selection.** 0: Late write. 1: Extended write.
4	**Priority Mode.** 0: Fixed. 1: Rotating.
3	**Timing Mode.** 0: Normal. 1: Compressed.
2	**Channels 3:0.** 0: Disable. 1: Enable.
1:0	**Reserved.** Must be set to 0.
I/O Port 009h	**Software DMA Request Register, Channels 3:0 (W)**
7:3	**Reserved.** Must be set to 0.
2	**Request Type.** 0: Reset. 1: Set.
1:0	**Channel Number Request Select** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
I/O Port 00Ah	**DMA Channel Mask Register, Channels 3:0 (WO)**
7:3	**Reserved.** Must be set to 0.
2	**Channel Mask.** 0: Not masked. 1: Masked.
1:0	**Channel Number Mask Select.** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.
I/O Port 00Bh	**DMA Channel Mode Register, Channels 3:0 (WO)**
7:6	**Transfer Mode.** 00: Demand. 01: Single. 10: Block. 11: Cascade.
5	**Address Direction.** 0: Increment. 1: Decrement.
4	**Auto-initialize.** 0: Disable. 1: Enable.
3:2	**Transfer Type.** 00: Verify. 01: Memory read. 10: Memory write. 11: Reserved.
1:0	**Channel Number Mode Select.** 00: Channel 0. 01: Channel 1. 10: Channel 2. 11: Channel 3.
I/O Port 00Ch	**DMA Clear Byte Pointer Command, Channels 3:0 (W)**
I/O Port 00Dh	**DMA Master Clear Command, Channels 3:0 (W)**
I/O Port 00Eh	**DMA Clear Mask Register Command, Channels 3:0 (W)**
I/O Port 00Fh	**DMA Write Mask Register Command, Channels 3:0 (W)**
I/O Port 0C0h	**DMA Channel 4 Address Register (R/W)**
Not used.	
I/O Port 0C2h	**DMA Channel 4 Transfer Count Register (R/W)**
Not used.	
I/O Port 0C4h	**DMA Channel 5 Address Register (R/W)**
Memory address bytes 1 and 0.	
I/O Port 0C6h	**DMA Channel 5 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0C8h	**DMA Channel 6 Address Register (R/W)**
Memory address bytes 1 and 0.	

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
I/O Port 0CAh	**DMA Channel 6 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0CCh	**DMA Channel 7 Address Register (R/W)**
Memory address bytes 1 and 0.	
I/O Port 0CEh	**DMA Channel 7 Transfer Count Register (R/W)**
Transfer count bytes 1 and 0.	
I/O Port 0D0h (R/W)	
Read	**DMA Status Register, Channels 7:4**
7	**Channel 7 Request.** Indicates if a request is pending. 0: No. 1: Yes.
6	**Channel 6 Request.** Indicates if a request is pending. 0: No. 1: Yes.
5	**Channel 5 Request.** Indicates if a request is pending. 0: No. 1: Yes.
4	**Undefined**
3	**Channel 7 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
2	**Channel 6 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
1	**Channel 5 Terminal Count.** Indicates if TC was reached. 0: No. 1: Yes.
0	**Undefined.**
Write	**DMA Command Register, Channels 7:4**
7	**DACK Sense.** 0: Active high. 1: Active low.
6	**DREQ Sense.** 0: Active high. 1: Active low.
5	**Write Selection.** 0: Late write. 1: Extended write.
4	**Priority Mode.** 0: Fixed. 1: Rotating.
3	**Timing Mode.** 0: Normal. 1: Compressed.
2	**Channels 7:4.** 0: Disable. 1: Enable.
1:0	**Reserved.** Must be set to 0.

Core Logic Module (Continued)

Table 5-43. DMA Channel Control Registers (Continued)

Bit	Description
I/O Port 0D2h	**Software DMA Request Register, Channels 7:4 (W)**
7:3	**Reserved.** Must be set to 0.
2	**Request Type.** 0: Reset. 1: Set.
1:0	**Channel Number Request Select.** 00: Illegal. 01: Channel 5. 10: Channel 6. 11: Channel 7.
I/O Port 0D4h	**DMA Channel Mask Register, Channels 7:4 (WO)**
7:3	**Reserved.** Must be set to 0.
2	**Channel Mask.** 0: Not masked. 1: Masked.
1:0	**Channel Number Mask Select.** 00: Channel 4. 01: Channel 5. 10: Channel 6. 11: Channel 7.
I/O Port 0D6h	**DMA Channel Mode Register, Channels 7:4 (WO)**
7:6	**Transfer Mode.** 00: Demand. 01: Single. 10: Block. 11: Cascade.
5	**Address Direction.** 0: Increment. 1: Decrement.
4	**Auto-initialize.** 0: Disabled. 1: Enable.
3:2	**Transfer Type.** 00: Verify. 01: Memory read. 10: Memory write. 11: Reserved.
1:0	**Channel Number Mode Select.** 00: Channel 4. 01: Channel 5. 10: Channel 6. 11: Channel 7. Channel 4 must be programmed in cascade mode. This mode is not the default.
I/O Port 0D8h	**DMA Clear Byte Pointer Command, Channels 7:4 (W)**
I/O Port 0DAh	**DMA Master Clear Command, Channels 7:4 (W)**
I/O Port 0DCh	**DMA Clear Mask Register Command, Channels 7:4 (W)**
I/O Port 0DEh	**DMA Write Mask Register Command, Channels 7:4 (W)**

Core Logic Module (Continued)

Table 5-44. DMA Page Registers

Bit	Description
I/O Port 081h	**DMA Channel 2 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 082h	**DMA Channel 3 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 083h	**DMA Channel 1 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 087h	**DMA Channel 0 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 089h	**DMA Channel 6 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Ah	**DMA Channel 7 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Bh	**DMA Channel 5 Low Page Register (R/W)**
Address bits [23:16] (byte 2).	
I/O Port 08Fh	**ISA Refresh Low Page Register (R/W)**
Refresh address.	
I/O Port 481h	**DMA Channel 2 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 081h.	
I/O Port 482h	**DMA Channel 3 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 082h.	
I/O Port 483h	**DMA Channel 1 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 083h.	
I/O Port 487h	**DMA Channel 0 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 087h.	
I/O Port 489h	**DMA Channel 6 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 089h.	
I/O Port 48Ah	**DMA Channel 7 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 08Ah.	
I/O Port 48Bh	**DMA Channel 5 High Page Register (R/W)**
Address bits [31:24] (byte 3).	
Note: This register is reset to 00h on any access to Port 08Bh.	

Core Logic Module (Continued)

Table 5-45. Programmable Interval Timer Registers

Bit	Description
I/O Port 040h	
Write	**PIT Timer 0 Counter**
7:0	Counter Value.
Read	**PIT Timer 0 Status**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).
I/O Port 041h	
Write	**PIT Timer 1 Counter (Refresh)**
7:0	Counter Value.
Read	**PIT Timer 1 Status (Refresh)**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).
I/O Port 042h	
Write	**PIT Timer 2 Counter (Speaker)**
7:0	Counter Value.
Read	**PIT Timer 2 Status (Speaker)**
7	**Counter Output.** State of counter output signal.
6	**Counter Loaded.** Indicates if the last count written is loaded. 0: Yes. 1: No.

Core Logic Module (Continued)

Table 5-45. Programmable Interval Timer Registers (Continued)

Bit	Description
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).
I/O Port 043h (R/W)	**PIT Mode Control Word Register**
Notes: 1.	If bits [7:6] = 11: Register functions as Read Status Command and: Bit 5 = Latch Count Bit 4 = Latch Status Bit 3 = Select Counter 2 Bit 2 = Select Counter 1 Bit 1 = Select Counter 0 Bit 0 = Reserved
2.	If bits [5:4] = 00: Register functions as Counter Latch Command and: Bits [7:6] = Selects Counter Bits [3:0] = Don't care
7:6	**Counter Select.** 00: Counter 0. 01: Counter 1. 10: Counter 2. 11: Read-back command (Note 1).
5:4	**Current Read/Write Mode.** 00: Counter latch command. 01: R/W LSB only. 10: R/W MSB only. 11: R/W LSB, followed by MSB.
3:1	**Current Counter Mode.** 0-5.
0	**BCD Mode.** 0: Binary. 1: BCD (Binary Coded Decimal).

Table 5-46. Programmable Interrupt Controller Registers

Bit	Description
I/O Port 020h / 0A0h	**Master / Slave PIC ICW1 (WO)**
7:5	**Reserved.** Must be set to 0.
4	**Reserved.** Must be set to 1.
3	**Trigger Mode.** 0: Edge. 1: Level.
2	**Vector Address Interval.** 0: 8 byte intervals. 1: 4 byte intervals.
1	**Reserved.** Must be set to 0 (cascade mode).
0	**Reserved.** Must be set to 1 (ICW4 must be programmed).
I/O Port 021h / 0A1h	**Master / Slave PIC ICW2 (after ICW1 is written) (WO)**
7:3	**A[7:3].** Address lines [7:3] for base vector for interrupt controller.
2:0	**Reserved.** Must be set to 0.
I/O Port 021h / 0A1h	**Master / Slave PIC ICW3 (after ICW2 is written) (WO)**
Master PIC ICW3	
7:0	**Cascade IRQ.** Must be 04h.
Slave PIC ICW3	
7:0	**Slave ID.** Must be 02h.
I/O Port 021h / 0A1h	**Master / Slave PIC ICW4 (after ICW3 is written) (WO)**
7:5	**Reserved.** Must be set to 0.
4	**Special Fully Nested Mode.** 0: Disable. 1: Enable.
3:2	**Reserved.** Must be set to 0.
1	**Auto EOI.** 0: Normal EOI. 1: Auto EOI.
0	**Reserved.** Must be set to 1 (8086/8088 mode).
I/O Port 021h / 0A1h (R/W)	**Master / Slave PIC OCW1 (except immediately after ICW1 is written)**
7	**IRQ7 / IRQ15 Mask.** 0: Not Masked. 1: Mask.
6	**IRQ6 / IRQ14 Mask.** 0: Not Masked. 1: Mask.
5	**IRQ5 / IRQ13 Mask.** 0: Not Masked. 1: Mask.
4	**IRQ4 / IRQ12 Mask.** 0: Not Masked. 1: Mask.
3	**IRQ3 / IRQ11 Mask.** 0: Not Masked. 1: Mask.

Core Logic Module (Continued)

Table 5-46. Programmable Interrupt Controller Registers (Continued)

Bit	Description
2	**IRQ2 / IRQ10 Mask.** 0: Not Masked. 1: Mask.
1	**IRQ1 / IRQ9 Mask.** 0: Not Masked. 1: Mask.
0	**IRQ0 / IRQ8# Mask.** 0: Not Masked. 1: Mask.
I/O Port 020h / 0A0h	**Master / Slave PIC OCW2 (WO)**
7:5	**Rotate/EOI Codes.** 000: Clear rotate in Auto EOI mode 001: Non-specific EOI 010: No operation 011: Specific EOI (bits [2:0] must be valid) 100: Set rotate in Auto EOI mode 101: Rotate on non-specific EOI command 110: Set priority command (bits [2:0] must be valid) 111: Rotate on specific EOI command
4:3	**Reserved.** Must be set to 0.
2:0	**IRQ Number (000-111).**
I/O Port 020h / 0A0h	**Master / Slave PIC OCW3 (WO)**
7	**Reserved.** Must be set to 0.
6:5	**Special Mask Mode.** 00: No operation. 01: No operation. 10: Reset Special Mask Mode. 11: Set Special Mask Mode.
4	**Reserved.** Must be set to 0.
3	**Reserved.** Must be set to 1.
2	**Poll Command.** 0: Disable. 1: Enable.
1:0	**Register Read Mode.** 00: No operation. 01: No operation. 10: Read interrupt request register on next read of Port 20h. 11: Read interrupt service register on next read of Port 20h.
I/O Port 020h / 0A0h	**Master / Slave PIC Interrupt Request and Service Registers for OCW3 Commands (RO)**
The function of this register is set with bits [1:0] in a write to 020h.	
Interrupt Request Register	
7	**IRQ7 / IRQ15 Pending.** 0: Yes. 1: No.
6	**IRQ6 / IRQ14 Pending.** 0: Yes. 1: No.
5	**IRQ5 / IRQ13 Pending.** 0: Yes. 1: No.

Table 5-46. Programmable Interrupt Controller Registers (Continued)

Bit	Description
4	**IRQ4 / IRQ12 Pending.** 0: Yes. 1: No.
3	**IRQ3 / IRQ11 Pending.** 0: Yes. 1: No.
2	**IRQ2 / IRQ10 Pending.** 0: Yes. 1: No.
1	**IRQ1 / IRQ9 Pending.** 0: Yes. 1: No.
0	**IRQ0 / IRQ8# Pending.** 0: Yes. 1: No.
Interrupt Service Register	
7	**IRQ7 / IRQ15 In-Service.** 0: No. 1: Yes.
6	**IRQ6 / IRQ14 In-Service.** 0: No. 1: Yes.
5	**IRQ5 / IRQ13 In-Service.** 0: No. 1: Yes.
4	**IRQ4 / IRQ12 In-Service.** 0: No. 1: Yes.
3	**IRQ3 / IRQ11 In-Service.** 0: No. 1: Yes.
2	**IRQ2 / IRQ10 In-Service.** 0: No. 1: Yes.
1	**IRQ1 / IRQ9 In-Service.** 0: No. 1: Yes.
0	**IRQ0 / IRQ8# In-Service.** 0: No. 1: Yes.

Core Logic Module (Continued)

Table 5-47. Keyboard Controller Registers

Bit	Description
I/O Port 060h	**External Keyboard Controller Data Register (R/W)**
Keyboard Controller Data Register. All accesses to this port are passed to the ISA bus. If the fast keyboard gate A20 and reset features are enabled through bit 7 of the ROM/AT Logic Control Register (F0 Index 52h[7]), the respective sequences of writes to this port assert the A20M# signal or cause a warm CPU reset.	
I/O Port 061h	**Port B Control Register (R/W)** **Reset Value: 00x01100b**
7	**PERR#/SERR# Status. (Read Only)** Indicates if a PCI bus error (PERR#/SERR#) was asserted by a PCI device or by the SC1100. 0: No. 1: Yes. This bit can only be set if ERR_EN (bit 2) is set 0. This bit is set 0 after a write to ERR_EN with a 1 or after reset.
6	**IOCHK# Status. (Read Only)** Indicates if an I/O device is reporting an error to the SC1100. 0: No. 1: Yes. This bit can only be set if IOCHK_EN (bit 3) is set 0. This bit is set 0 after a write to IOCHK_EN with a 1 or after reset.
5	**PIT OUT2 State. (Read Only)** This bit reflects the current status of the of the PIT Counter 2 (OUT2).
4	**Toggle. (Read Only)** This bit toggles on every falling edge of Counter 1 (OUT1).
3	**IOCHK# Enable.** 0: Generates an NMI if IOCHK# is driven low by an I/O device to report an error. Note that NMI is under SMI control. 1: Ignores the IOCHK# input signal and does not generate NMI.
2	**PERR/ SERR Enable.** Generate an NMI if PERR#/SERR# is driven active to report an error. 0: Enable. 1: Disable.
1	**PIT Counter2 (SPKR).** 0: Forces Counter 2 output (OUT2) to zero. 1: Allows Counter 2 output (OUT2) to pass to the speaker.
0	**PIT Counter2 Enable.** 0: Sets GATE2 input low. 1: Sets GATE2 input high.
I/O Port 062h	**External Keyboard Controller Mailbox Register (R/W)**
Keyboard Controller Mailbox Register.	
I/O Port 064h	**External Keyboard Controller Command Register (R/W)**
Keyboard Controller Command Register. All accesses to this port are passed to the ISA bus. If the fast keyboard gate A20 and reset features are enabled through bit 7 of the ROM/AT Logic Control Register (F0 Index 52h[7]), the respective sequences of writes to this port assert the A20M# signal or cause a warm CPU reset.	
I/O Port 066h	**External Keyboard Controller Mailbox Register (R/W)**
Keyboard Controller Mailbox Register.	
I/O Port 092h	**Port A Control Register (R/W)** **Reset Value: 02h**
7:2	**Reserved.** Must be set to 0.
1	**A20M# Assertion.** Assert A20# (internally). 0: Enable. 1: Disable. This bit reflects A20# status and can be changed by keyboard command monitoring. An SMI event is generated when this bit is changed, if enabled by F0 index 53h[0]. The SMI status is reported in F1BAR0+I/O Offset 00h/02h[7].
0	**Fast CPU Reset.** WM_RST SMI is asserted to the BIOS. 0: Disable. 1: Enable. This bit must be cleared before the generation of another reset.

Core Logic Module (Continued)

Table 5-48. Real-Time Clock Registers

Bit	Description
I/O Port 070h	**RTC Address Register (WO)**
This register is shadowed within the Core Logic module and is read through the RTC Shadow Register (F0 Index BBh).	
7	**NMI Mask**. 0: Enable. 1: Mask.
6:0	**RTC Register Index.** A write of this register sends the data out on the ISA bus and also causes RTCALE to be triggered. (RTCALE is an internal signal between the Core Logic module and the internal real-time clock controller.)
I/O Port 071h	**RTC Data Register (R/W)**
A read of this register returns the value of the register indexed by the RTC Address Register. A write of this register sets the value into the register indexed by the RTC Address Register	
I/O Port 072h	**RTC Extended Address Register (WO)**
7	**Reserved.**
6:0	**RTC Register Index.** A write of this register sends the data out on the ISA bus and also causes RTCALE to be triggered. (RTCALE is an internal signal between the Core Logic module and the internal real-time clock controller.)
I/O Port 073h	**RTC Data Register (R/W)**
A read of this register returns the value of the register indexed by the RTC Extended Address Register. A write of this register sets the value into the register indexed by the RTC Extended Address Register	

Table 5-49. Miscellaneous Registers

Bit	Description
I/O Port 0F0h, 0F1h	**Coprocessor Error Register (W)** **Reset Value: F0h**
A write to either port when the FERR# signal is asserted causes the Core Logic Module to assert IGNNE#. IGNNE# remains asserted until the FERR# deasserts.	
I/O Ports 170h-177h/376h-377h	**Secondary IDE Registers (R/W)**
When the local IDE functions are enabled, reads or writes to these registers cause the local IDE interface signals to operate according to their configuration rather than generating standard ISA bus cycles.	
I/O Ports 1F0h-1F7h/3F6h-3F7h	**Primary IDE Registers (R/W)**
When the local IDE functions are enabled, reads or writes to these registers cause the local IDE interface signals to operate according to their configuration rather than generating standard ISA bus cycles.	
I/O Port 4D0h	**Interrupt Edge/Level Select Register 1 (R/W)** **Reset Value: 00h**
Notes: 1. If ICW1 - bit 3 in the PIC is set as level, it overrides the setting for bits [7:3] in this register. 2. Bits [7:3] in this register are used to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).	
7	**IRQ7 Edge or Level Sensitive Select.** Selects PIC IRQ7 sensitivity configuration. 0: Edge. 1: Level.
6	**IRQ6 Edge or Level Sensitive Select.** Selects PIC IRQ6 sensitivity configuration. 0: Edge. 1: Level.
5	**IRQ5 Edge or Level Sensitive Select.** Selects PIC IRQ5 sensitivity configuration. 0: Edge. 1: Level.
4	**IRQ4 Edge or Level Sensitive Select.** Selects PIC IRQ4 sensitivity configuration. 0: Edge. 1: Level.

Core Logic Module (Continued)

Table 5-49. Miscellaneous Registers (Continued)

Bit	Description
3	**IRQ3 Edge or Level Sensitive Select.** Selects PIC IRQ3 sensitivity configuration. 0: Edge. 1: Level.
2:0	**Reserved**. Must be set to 0.
I/O Port 4D1h	**Interrupt Edge/Level Select Register 2 (R/W)** **Reset Value: 00h**
Notes: 1. If ICW1 - bit 3 in the PIC is set as level, it overrides the setting for bits 7:6 and 4:1 in this register. 2. Bits [7:6] and [4:1] in this register are used to configure a PCI interrupt mapped to IRQ[x] on the PIC as level-sensitive (shared).	
7	**IRQ15 Edge or Level Sensitive Select.** Selects PIC IRQ15 sensitivity configuration. 0: Edge. 1: Level.
6	**IRQ14 Edge or Level Sensitive Select.** Selects PIC IRQ14 sensitivity configuration. 0: Edge. 1: Level.
5	**Reserved**. Must be set to 0.
4	**IRQ12 Edge or Level Sensitive Select.** Selects PIC IRQ12 sensitivity configuration. 0: Edge. 1: Level.
3	**IRQ11 Edge or Level Sensitive Select.** Selects PIC IRQ11 sensitivity configuration. 0: Edge. 1: Level.
2	**IRQ10 Edge or Level Sensitive Select.** Selects PIC IRQ10 sensitivity configuration. 0: Edge. 1: Level.
1	**IRQ9 Edge or Level Sensitive Select.** Selects PIC IRQ9 sensitivity configuration. 0: Edge. 1: Level.
0	**Reserved.** Must be set to 0.

6.0 Debugging and Monitoring

6.1 TESTABILITY (JTAG)

The Test Access Port (TAP) allows board level interconnection verification and chip production tests. An IEEE-1149.1a compliant test interface, TAP supports all IEEE mandatory instructions as well as several optional instructions for added functionality. See Table 6-1 for a summary of all instruction support. For further information on JTAG, refer to IEEE Standard 1149.1a-1993 Test Access Port and Boundary-Scan Architecture.

6.1.1 Mandatory Instruction Support

The TAP supports all IEEE mandatory instructions, including:

- BYPASS
  Presents the shortest path through a given chip (a 1-bit shift register).
- EXTEST
  Drives data loaded into the JTAG path (possibly with a SAMPLE/PRELOAD instruction) to output pins.
- SAMPLE/PRELOAD
  Captures chip inputs and outputs.

6.1.2 Optional Instruction Support

The TAP supports the following IEEE optional instructions:

- IDCODE
  Presents the contents of the Device Identification register in serial format.
- CLAMP
  Ensures that the Bypass register is connected between TDI and TDO, and then drives data that was loaded into the Boundary Scan register (e.g., via SAMPLE-PRELOAD instruction) to output signals. These signals do not change while the CLAMP instruction is selected.
- HIGHZ
  Puts all chip outputs in inactive (floating) state (including all pins that do not require a TRI-STATE output for normal functionality). Note that not all pull-up resistors are disabled in this state.

6.1.3 JTAG Chain

Pins that are not part of the JTAG chain:

- USB I/Os

Table 6-1. JTAG Mode Instruction Support

Code	Instruction	Activity
000	EXTEST	Drives shifted data to output pins.
001	SAMPLE/PRELOAD	Captures inputs and system outputs.
010	IDCODE	Scans out device identifier.
011	HIGHZ	Puts all output and bidirectional pins in TRI-STATE.
100	CLAMP	Drives fixed data from Boundary Scan register.
101	Reserved	
110	Reserved	
111	BYPASS	Presents shortest external path through device.

7.0 Electrical Specifications

This chapter provides information about:

- General electrical specifications
- DC characteristics
- AC characteristics

All voltage values in this chapter are with respect to V_{SS} unless otherwise noted.

7.1 GENERAL SPECIFICATIONS

7.1.1 Power/Ground Connections and Decoupling

When designing in the SC1100, use standard high frequency design techniques to reduce parasitic effects. For example:

- Filter the DC power leads with low-inductance decoupling capacitors.
- Use low-impedance traces.

National Semiconductor's application note "Geode SC1100 Information Appliance On A Chip: Layout Recommendations" provides detailed guidelines for producing optimal PCB layouts.

7.1.2 Absolute Maximum Ratings

Stresses beyond those indicated in the following table may cause permanent damage to the SC1100, 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 following table are stress ratings only. They do not imply that operation under other conditions is impossible.

Table 7-1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit	Comments
T_{CASE}	Operating case temperature[1]	-10	110	^oC	
$T_{STORAGE}$	Storage temperature[2]	-45	125	^oC	
V_{CC}	Supply voltage		See Table 7-2	V	
V_{MAX}	Voltage on				
	5V tolerant balls	-0.5	6.0	V	
	Others[3 4]	-0.5	4.2	V	
I_{IK}	Input clamp current[1]	-0.5	10	mA	
I_{OK}[1]	Output clamp current		25	mA	

1. Power applied - no clocks.
2. No bias.
3. Voltage min is -0.8V with a transient voltage of 20 ns or less.
4. Voltage max is 4.0V with a transient voltage of 20 ns or less.

Electrical Specifications (Continued)

7.1.3 Operating Conditions

Table 7-2 lists the power supplies of the SC1100 and provides the device operating conditions.

Table 7-2. Operating Conditions

Symbol[1]	Parameter	Min	Typ	Max	Unit	Comments
T_C	Operating case temperature.	0	-	85	^oC	
AV_{CCUSB}	Analog power supply. Powers internal analog circuits and some external signals (see Table 7-3).	3.135	3.3	3.465	V	
V_{BAT}	Battery supply voltage. Powers RTC and ACPI when V_{BAT} is greater than V_{SB} (by at least 0.5V), and some external signals (see Table 7-3).	2.4	3.0	3.6	V	
V_{IO}	I/O buffer power supply. Powers most of the external signals (see Table 7-3); certain signals within this power plane are 5V tolerant.	3.135	3.3	3.465	V	
V_{CORE}	Core processor and internal digital power supply. Powers internal digital logic, including internal frequency multipliers.					
	233 MHz Core clock frequency.	1.71	1.8	1.89	V	
	266 MHz Core clock frequency.	1.9	2.0	2.1	V	
	300 MHz Core clock frequency.	TBD	TBD	TBD	V	
V_{PLL}	PLL. Internal Phase Locked Loop (PLL) power supply.	3.135	3.3	3.465	V	
V_{SB}	Standby power supply. Powers RTC and ACPI when V_{SB} is greater than V_{BAT} -0.5V, and some external signals (see Table 7-3).	3.135	3.3	3.465	V	
V_{SBL}	Standby logic. Powers internal logic needed to support Standby V_{SB}. V_{SBL} (ball AD16) requires a 0.1 µF bypass capacitor to V_{SS}.					
	233 MHz Core clock frequency.	1.71	1.8	1.89	V	
	266 MHz Core clock frequency.	1.9	2.0	2.1	V	
	300 MHz Core clock frequency.	TBD	TBD	TBD	V	

1. For V_{IH} (Input High Voltage), V_{IL} (Input Low Voltage), I_{OH} (Output High Current), and I_{OL} (Output Low Current) operating conditions refer to Section 7.2 "DC Characteristics" on page 286.

Note:

1) All power sources must be connected to the SC1100, even if the function is not used.

2) Voltages must be applied according to the sequence set forth in Section 7.3.12 on page 339.

3) V_{SBL} and V_{CORE} must adhere at all times to the following requirement: $V_{SBL} \geq V_{CORE}$.

4) The power planes of the SC1100 can be turned on or off. For more information, see Section 5.2.9 "Power Management Logic" on page 134.

5) It is recommended that the voltage difference between V_{CORE} and V_{SBL} be less than 0.25V in order to reduce leakage current. If the voltage difference exceeds 0.25V, excessive leakage current is used in gates that are connected on the boundary between voltage domains.

6) It is recommended that the voltage difference between V_{IO} and V_{SB} be less than 0.25V in order to reduce leakage current. If the voltage difference exceeds 0.25V, excessive leakage current is used in gates that are connected on the boundary between voltage domains.

7) V_{SB}, V_{SBL} and V_{BAT} must be on if any other voltage is applied. V_{SB} and V_{BAT} voltages can be applied separately. See Section 7.3.12 "Power-Up Sequencing" on page 339.

Electrical Specifications (Continued)

Table 7-3 indicates which power rails are used for each signal of the SC1100 external interface. Power planes not listed in this table are internal, and are not related to signals of the external interface.

Table 7-3. Power Planes of External Interface Signals

Power Plane	Signal Names	V_{CC} Balls	V_{SS} Balls
Standby	GPWIO[0:2], ONCTL#, PWRBTN#, PWRCNT[1:2], THRM#, IRRX1, RI#	V_{SB}	V_{SS}
Battery	X32I, X32O	V_{BAT}	V_{SS}
USB	DPOS_PORT1, DNEG_PORT1, DPOS_PORT2, DNEG_PORT2, DPOS_PORT3, DNEG_PORT3	AV_{CCUSB}	AV_{SSUSB}
I/O	All other external interface signals	V_{IO}	V_{SS}

7.1.4 DC Current

DC current is not a simple measurement. Three of the SC1100 power states (On, Active Idle, Sleep) were selected for measurement. For each power state measured, two functional characteristics (Typical Average, Absolute Maximum) are used to determine how much current the SC1100 uses.

7.1.4.1 Power State Parameter Definitions

The DC characteristics tables in this section list Core and I/O current for three of the power states. For more explanation on the SC1100 power states see Section 5.2.9 "Power Management Logic" on page 134.

- **On (C0)**: All internal and external clocks with respect to the SC1100 are running and all functional blocks inside the GX1 module (CPU Core, Memory Controller, etc.) are actively generating cycles. This is equivalent to the ACPI specification's "S0,C0" state.
- **Active Idle (C1)**: The CPU Core has been halted, all other functional blocks are actively generating cycles. This state is entered when a HLT instruction is executed by the CPU Core. From a user's perspective, this state is indistinguishable from the On state and is equivalent to the ACPI specification's "S0,C1" state.
- **Sleep (SL2)**: This is the lowest power state the SC1100 can be in with voltage still applied to the device's core and I/O supply pins. This is equivalent to the ACPI specification's "S1" state.

7.1.4.2 Definition and Measurement Techniques of SC1100 Current Parameters

The following two parameters describe the SC1100 current while in the On state:

- **Typical Average**: Indicates the average current used by the SC1100 while in the On state. This is measured by running typical Windows applications with power management disabled (to guarantee that the SC1100 never goes into the Active Idle state). This number is provided for reference only since it can vary greatly depending on the usage model of the system.

Note: This typical average should not be confused with the typical power numbers shown in Table 7-2 on page 281. The numbers in Table 7-2 are based on a combination of On (Typical Average) and Active Idle states.

- **Absolute Maximum**: Indicates the maximum instantaneous current used by the SC1100. CPU Core current is measured by running the Landmark Speed 200 benchmark test (with power management disabled) and measuring the peak current at any given instant during the test. I/O current is measured by running Microsoft Windows 98.

Electrical Specifications (Continued)

7.1.4.3 Current Consumption

The following table contains **PRELIMINARY** information that is **SUBJECT TO CHANGE**.

Table 7-4. Current Consumption

Symbol	Parameter	Min	Typ[1]	Max	Unit	Conditions
I_{CC0}	V_{CORE} power supply current, C0 power state - Full speed[2]					
	233 MHz		820	1100	mA	
	266 MHz		940	1260	mA	
	300 MHz		TBD	TBD	mA	
I_{CC1}	V_{CORE} power supply current, C1 power state					
	233 MHz		360	450	mA	
	266 MHz		380	475	mA	
	300 MHz		TBD	TBD	mA	
I_{CC3}	V_{CORE} power supply current, C3 power state					
	233 MHz		TBD	TBD	mA	
	266 MHz		TBD	TBD	mA	
	300 MHz		TBD	TBD	mA	
I_{CCSL1}	V_{CORE} power supply current, SL1 Sleep state		70	110	mA	
I_{CCSL2}	V_{CORE} power supply current, SL2 Sleep state		10	20	mA	
I_{CCSL3}	V_{CORE} power supply current, SL3 Sleep state		-	0	mA	
I_{CCSL45}	V_{CORE} power supply current, SL4 and SL5 Sleep state		-	0	mA	
I_{BAT}	V_{BAT} battery supply current		7	50	µA	V_{BAT} = 3V @25°C other supplies at 0V
I_{SBL}	V_{SBL} average current supply		1.5	2	mA	
I_{SB}	V_{SB} average current supply		-	1.5	mA	$V_{IL} = V_{SS}$, $V_{IH} = V_{SB}$ No Load
I_{IO}	V_{IO} power supply current					
	233 MHz		200	220	mA	
	266 MHz		210	230	mA	
	300 MHz		TBD	TBD	mA	
I_{PLL}	V_{PLL} average current supply		13.5	15	mA	

1. Typical conditions: Nominal power supply, and $T_A = 25^oC$.
2. Throttling current depends on% performance and therefore cannot be specified.

Electrical Specifications (Continued)

7.1.5 Ball Capacitance and Inductance

Table 7-5 gives ball capacitance and inductance values.

Table 7-5. Ball Capacitance and Inductance

Symbol	Parameter	Min	Typ	Max	Unit
C_{IN}[1]	Input Pin Capacitance		4	7	pF
C_{IN}[1]	Clock Input Capacitance	5	8	12	pF
C_{IO}[1]	I/O Pin Capacitance		10	12	pF
C_{O}[1]	Output Pin Capacitance		6	8	pF
L_{PIN}[2]	Pin Inductance			20	nH

1. T_A = 25°C, f = 1 MHz. All capacitances are not 100% tested.
2. Not 100% tested.

Electrical Specifications (Continued)

7.1.6 Pull-Up and Pull-Down Resistors

The following table lists input balls that are internally connected to a pull-up or pull-down resistor. If these balls are not used, they do not require connection to an external pull-up or pull-down resistor.

Note: The resistors described in this table are implemented as transistors.

Table 7-6. Balls with PU/PD Resistors

Signal Name	Ball No.	PU/PD	Typ[1] Value [Ω]	Conditions
PCI				
FRAME#	P25	PU	22.5K	@ $V_{IN} = V_{SS}$
C/BE[3:0]#	V26, R26, G24, E24	PU	22.5K	@ $V_{IN} = V_{SS}$
PAR	H24	PU	22.5K	@ $V_{IN} = V_{SS}$
IRDY#	P26	PU	22.5K	@ $V_{IN} = V_{SS}$
TRDY#	N25	PU	22.5K	@ $V_{IN} = V_{SS}$
STOP#	M25	PU	22.5K	@ $V_{IN} = V_{SS}$
DEVSEL#	N26	PU	22.5K	@ $V_{IN} = V_{SS}$
PERR#	J24	PU	22.5K	@ $V_{IN} = V_{SS}$
SERR#	L25	PU	22.5K	@ $V_{IN} = V_{SS}$
REQ3#	T24	PU	22.5K	@ $V_{IN} = V_{SS}$
REQ2#	AC26	PU	22.5K	@ $V_{IN} = V_{SS}$
REQ1#	AA25	PU	22.5K	@ $V_{IN} = V_{SS}$
REQ0#	AA26	PU	22.5K	@ $V_{IN} = V_{SS}$
INTA#	AD26	PU	22.5K	@ $V_{IN} = V_{SS}$
INTB#	W24	PU	22.5K	@ $V_{IN} = V_{SS}$
INTC#	Y24	PU	22.5K	@ $V_{IN} = V_{SS}$
INTD#	V24	PU	22.5K	@ $V_{IN} = V_{SS}$
Low Pin Count (LPC)				
LAD[3:0]	C25, D26, D25, E26	PU	22.5K	@ $V_{IN} = V_{SS}$
LDRQ#	C26	PU	22.5K	@ $V_{IN} = V_{SS}$
SERIRQ	A24	PU	22.5K	@ $V_{IN} = V_{SS}$
System (Straps)				
CLKSEL[3:0]	AE22, AD22, AD25, D23	PD	100K	@ $V_{IN} = V_{IO}$
BOOT16	C23	PD	100K	@ $V_{IN} = V_{IO}$
FPCI_MON	AB25	PD	100K	@ $V_{IN} = V_{IO}$
LPC_ROM	AB26	PD	100K	@ $V_{IN} = V_{IO}$
ACCESS.bus[2]				
AB1C	Y1	PU	22.5K	@ $V_{IN} = V_{SS}$
AB1D	Y2	PU	22.5K	@ $V_{IN} = V_{SS}$
AB2C	AE23	PU	22.5K	@ $V_{IN} = V_{SS}$
AB2D	AD23	PU	22.5K	@ $V_{IN} = V_{SS}$
JTAG				
TCK	AE20	PU	22.5K	@ $V_{IN} = V_{SS}$
TMS	AF21	PU	22.5K	@ $V_{IN} = V_{SS}$
TDI	AF20	PU	22.5K	@ $V_{IN} = V_{SS}$
TRST#	AC20	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO[2]				
GPIO0	B22	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO1	AD24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO2	B21	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO3	A22	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO6	AD12	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO7	AF11	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO8	AC12	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO9	AE12	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO10	AF12	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO11	AF19	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO12	AE23	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO13	AD23	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO14	D22	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO15	C22	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO16	AE18	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO17	B23	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO18	AC24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO19	Y24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO20	D21	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO32	E26	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO33	D25	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO34	D26	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO35	C25	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO36	C26	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO37	B24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO38	AB23	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO39	A24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO40	B20	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO41	AA24	PU	22.5K	@ $V_{IN} = V_{SS}$
GPIO47	AB24	PU	22.5K	@ $V_{IN} = V_{SS}$
Power Management				
PWRBTN#	AF15	PU	100K	@ $V_{IN} = V_{SS}$
GPWIO[2:0]	AF17, AE16, AC15	PU	100K	@ $V_{IN} = V_{SS}$
Test and Measurement				
GTEST	AD21	PD	22.5K	@ $V_{IN} = V_{IO}$

1. Accuracy is: 22.5 KΩ resistors are within a range of 20 KΩ to 50 KΩ. 100 KΩ resistors are within a range of 90 KΩ to 250 KΩ.
2. Controlled by software.

Electrical Specifications (Continued)

7.2 DC CHARACTERISTICS

Table 7-7 describes the signal buffer types of the SC1100. (See Table 2-2 on page 19 for each signal's buffer type.) The subsections that follows provide detailed DC characteristics according to buffer type.

Table 7-7. Buffer Types

Symbol	Description	Reference
Diode	Diodes only, no buffer	---
IN_{AB}	Input, ACCESS.bus compatible with Schmitt Trigger	Section 7.2.1
IN_{BTN}	Input, TTL compatible with Schmitt Trigger, low leakage	Section 7.2.2
IN_{PCI}	Input, PCI compatible	Section 7.2.3
IN_{T}	Input, TTL compatible	Section 7.2.4
IN_{TS}	Input, TTL compatible with Schmitt Trigger type 200 mV	Section 7.2.5
IN_{TS1}	Input, with Schmitt Trigger type 200 mV	Section 7.2.6
IN_{USB}	Input, USB compatible	Section 7.2.7
OD_{n}	Output, Open-Drain, capable of sinking *n* mA.[1]	Section 7.2.8
OD_{PCI}	Output, Open-Drain, PCI compatible	Section 7.2.9
$O_{p/n}$	Output, Totem-Pole, capable of sourcing *p* mA and sinking *n* mA	Section 7.2.10
O_{PCI}	Output, PCI compatible, TRI-STATE	Section 7.2.11
O_{USB}	Output, USB compatible	Section 7.2.12
$TS_{p/n}$	Output, TRI-STATE, capable of sourcing *p* mA and sinking *n* mA	Section 7.2.13
WIRE	Wire, no buffer	---

1. Output from these signals is open-drain and cannot be forced high.

Electrical Specifications (Continued)

7.2.1 IN_{AB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	1.4		V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input hysteresis	150		mV	

1. Not 100% tested.

7.2.2 IN_{BTN} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{SB}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		5	µA	$V_{IN} = V_{SB}$
			-36	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input Hysteresis[1]	200		mV	

1. Not 100% tested.

7.2.3 IN_{PCI} DC Characteristics

(Note that the buffer type for PCICLK, ball AA23, is IN_T - not IN_{PCI}.)

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	$0.5V_{IO}$	V_{IO}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	$0.3V_{IO}$	V	
V_{IPU}	Input Pull-up Voltage[2]	$0.7V_{IO}$		V	
I_{IL}	Input Leakage Current[3,4]		+/-10	µs	$0 < V_{IN} < V_{IO}$

1. Not 100% tested.
2. Not 100% tested. This parameter indicates the minimum voltage to which pull-up resistors are calculated in order to pull a floated network.
3. Input leakage currents include Hi-Z output leakage for all bidirectional buffers with TRI-STATE outputs.
4. See Exceptions 2 and 3 in Section 7.2.13.1 on page 290.

7.2.4 IN_T DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µs	$V_{IN} = V_{IO}$
			–10	µs	$V_{IN} = V_{SS}$

1. Not 100% tested.

Electrical Specifications (Continued)

7.2.5 IN_{TS} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µs	$V_{IN} = V_{IO}$
			-10	µs	$V_{IN} = V_{SS}$
V_H	Input Hysteresis	200		mV	

1. Not 100% tested.

7.2.6 IN_{TS1} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	$0.5V_{IO}$	V_{IO}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	$0.3V_{IO}$	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{HIS}	Input Hysteresis[1]	200		mV	

1. Not 100% tested.

Electrical Specifications (Continued)

7.2.7 IN_{USB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{IH}	Input High Voltage	2.0	V_{IO}+0.5[1]	V	
V_{IL}	Input Low Voltage	-0.5[1]	0.8	V	
I_{IL}	Input Leakage Current		10	µA	$V_{IN} = V_{IO}$
			-10	µA	$V_{IN} = V_{SS}$
V_{DI}	Differential Input Sensitivity	0.2		V	|(D+)-(D-)| and Figure 7-1
V_{CM}	Differential Common Mode Range	0.8	2.5	V	Includes V_{DI} Range
V_{SE}	Single Ended Receiver Threshold	0.8	2.0	V	
V_{HIS}	Input Hysteresis[1]	150		mV	

1. Not 100% tested.

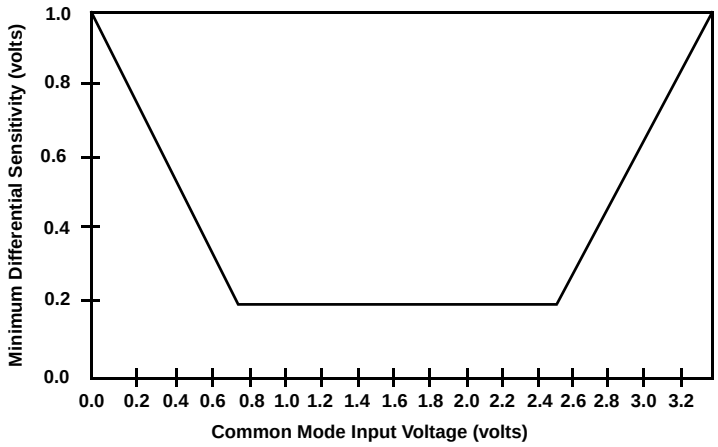


Figure 7-1. Differential Input Sensitivity for Common Mode Range

7.2.8 OD_n DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

7.2.9 OD_{PCI} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OL}	Output Low Voltage		$0.1V_{IO}$	V	I_{OL} = 1500 µA

7.2.10 $O_{p/n}$ DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -*p* mA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

Electrical Specifications (Continued)

7.2.11 O_{PCI} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	$0.9V_{IO}$		V	I_{OH} = -500 µA
V_{OL}	Output Low Voltage		$0.1V_{IO}$	V	I_{OL} =1500 µA

7.2.12 O_{USB} DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{USB_OH}	High-level output voltage	2.8	3.6[1]	V	I_{OH} = -0.25 mA R_L = 15 KΩ to GND
V_{USB_OL}	Low-level output voltage		0.3	V	I_{OL} = 2.5 mA R_L = 1.5 KΩ to 3.6V
t_{USB_CRS}	Output signal crossover voltage	1.3	2.0	V	

1. Tested by characterization.

7.2.13 $TS_{p/n}$ DC Characteristics

Symbol	Parameter	Min	Max	Unit	Comments
V_{OH}	Output High Voltage	2.4		V	I_{OH} = *-p* mA
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = *n* mA

7.2.13.1 Exceptions

1) I_{OH} is valid for a GPIO pin only when it is not configured as open-drain.
2) Signals with internal pull-ups have a maximum input leakage current of: $-\left(\frac{V_{power} - V_{IN}}{R(pull-up)}\right)$
   Where V_{POWER} is V_{IO}, or V_{SB}.
3) Signals with internal pull-downs have a maximum input leakage current of: $+\left(\frac{V_{IN} - V_{SS}}{R(pull-down)}\right)$

Electrical Specifications (Continued)

7.3 AC CHARACTERISTICS

The tables in this section list the following AC characteristics:

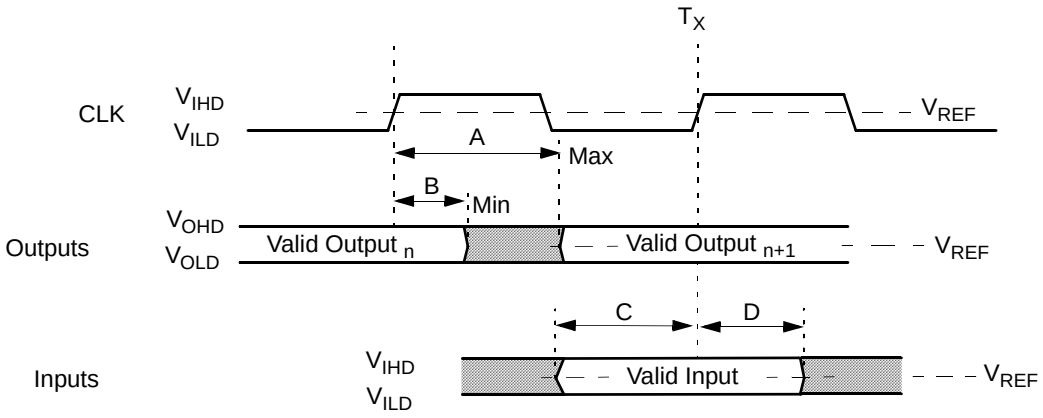
- Output delays
- Input setup requirements
- Input hold requirements
- Output float delays
- Power-up sequencing requirements

The default levels for measurement of the rising clock edge reference voltage (V_{REF}), and other voltages are shown in Table 7-8. Input or output signals must cross these levels during testing. Unless otherwise specified, all measurement points in this section conform to these default levels.

Table 7-8. Default Levels for Measurement of Switching Parameters

Symbol	Parameter	Value (V)
V_{REF}	Reference Voltage	1.5
V_{IHD}	Input High Drive Voltage	2.0
V_{ILD}	Input Low Drive Voltage	0.8
V_{OHD}	Output High Drive Voltage	2.4
V_{OLD}	Output Low Drive Voltage	0.4

All AC tests are at V_{IO} = 3.14V to 3.46V (3.3V nominal), T_C = 0 °C to 70 °C, C_L = 50 pF, unless otherwise specified.



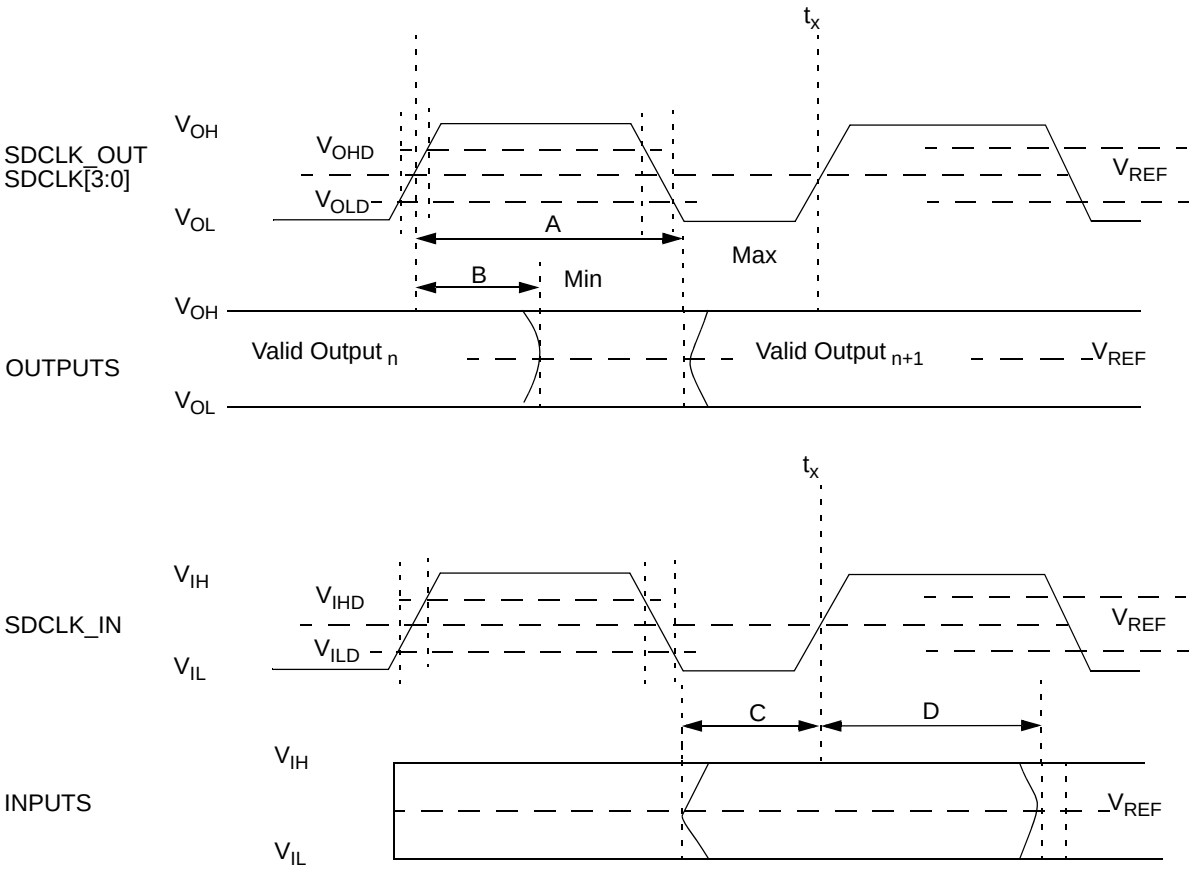
Legend: A = Maximum Output or Float Delay Specification
B = Minimum Output or Float Delay Specification
C = Minimum Input Setup Specification
D = Minimum Input Hold Specification

Figure 7-2. Drive level and Measurement Points for Switching Parameters

Electrical Specifications (Continued)

7.3.1 Memory Interface

The minimum input setup and hold times described in Figure 7-3 (legend C and D) define the smallest acceptable sampling window during which a synchronous input signal must be stable to ensure correct operation.



Legend: A = Maximum Output Delay

B = Minimum Output Delay

C = Minimum Input Setup

D = Minimum Input Hold

Figure 7-3. Drive Level and Measurement Points for Switching Characters

Electrical Specifications (Continued)

Table 7-9. SDRAM Interface Signals

Symbol	Parameter	Min	Max	Unit
t1	Control Output[1,2] Valid from SDCLK[3:0]	-3.9 + (x*y)	0.1 + (x*y)	ns
t2	MA[12:0], BA[1.0] Output[2] Valid from SDCLK[3:0]	-3.9 + (x*y)	0.1 + (x*y)	ns
t3	MD[63.0] Output[2] Valid from SDCLK[3:0]	-3.9 + (x*y)	0.7 + (x*y)	ns
t4	MD[63:0] Read Data in Setup to SDCLK_IN	1.3		ns
t5	MD[63:0] Read Data Hold to SDCLK_IN	2.0		ns
t6	SDCLK[3:0], SDCLK_OUT cycle time			
	233 MHz / 3.0	10.0	14.0	ns
	266 MHz / 3.0	8.3	13.5	
	300 MHz / 3.0	7.3	12.5	
t7	SDCLK, SDCLK_OUT fall/rise time between (V_{OLD}-V_{OHD})		1	ns
t8	SDCLK[3:0], SDCLK_OUT high time			
	233 MHz / 3.0	4.0		ns
	266 MHz / 3.0	3.0		
	300 MHz / 3.0	2.5		
t9	SDCLK[3:0], SDCLK_OUT low time			
	233 MHz / 3.0	4.0		ns
	266 MHz / 3.0	2.5		
	300 MHz / 3.0	2.5		
t10	SDCLK_IN fall/rise time between (V_{ILD}-V_{IHD})		1	ns

1. Control output includes all the following signals: RASA#, CASA#, WEA#, CKEA, DQM[7:0], and CS[1:0]#. Load = 50 pF, V_{CORE} = 1.8V - 2.1V, V_{IO} = 3.3V, @25°C.
2. Use the Min/Max equations [value+(x * y)] to calculate the actual value.
   x is the shift value which is applied to the SHFTSDCLK field, and y is 0.5 the core clock period.
   Note that the SHFTSDCLK field = GX_BASE+8404h[5:3]. See the GX1 Processor Series datasheet.
   For example, for a 233 MHz SC1100 processor running a 78 MHz SDRAM clock, with a shift value of 3:
   t1 Min = -3.0 + (3 * (4.29 * 0.5)) = 3.435 ns
   t1 Max = 0.1 + (3 * (4.29 * 0.5)) = 6.535 ns

Electrical Specifications (Continued)

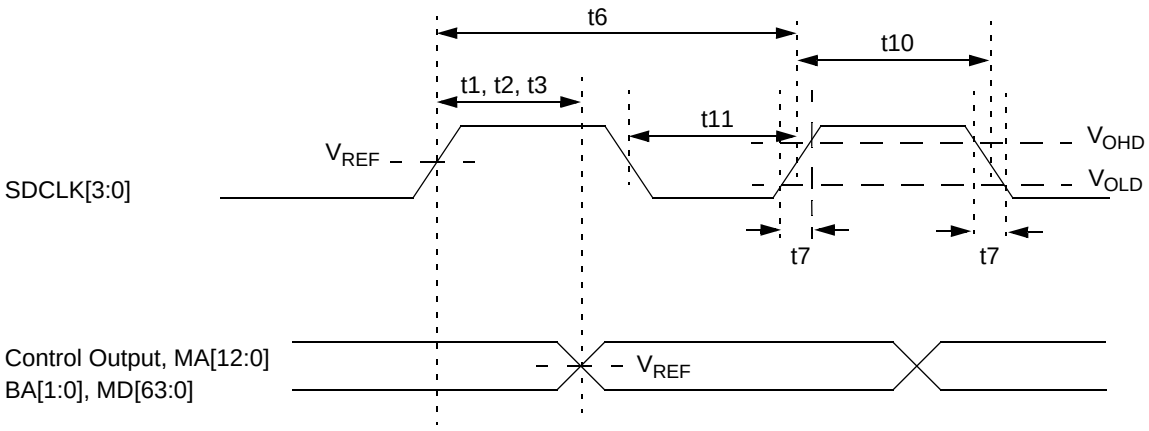


Figure 7-4. Output Valid Timing

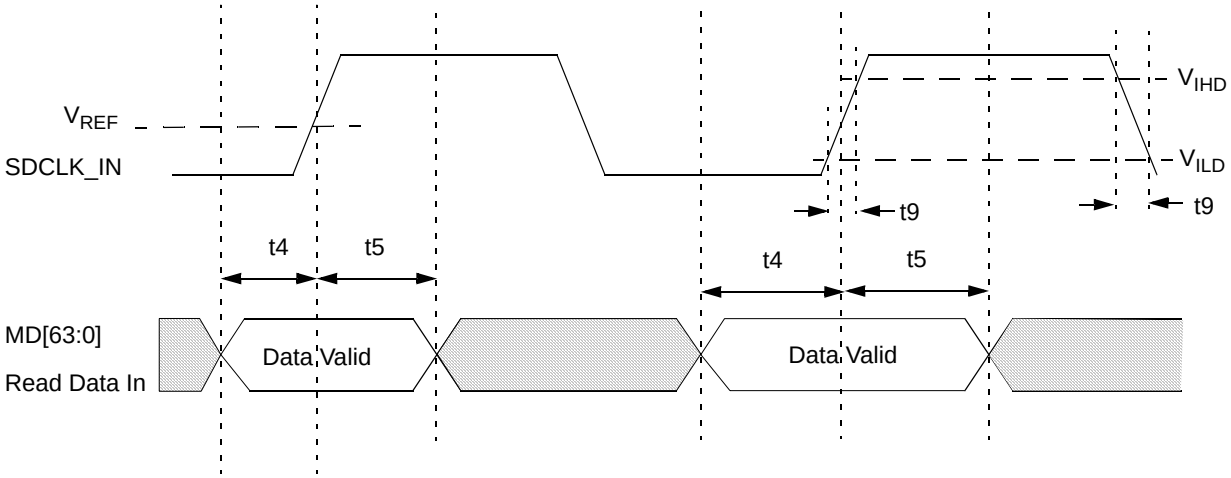


Figure 7-5. Setup and Hold Timing - Read Data In

Electrical Specifications (Continued)

7.3.2 ACCESS.bus Interface

The following tables describe the timing for all ACCESS.bus signals.

Notes: 1) All ACCESS.bus timing is not 100% tested.

2) In this table t_{CLK} = 1/24 MHz = 41.7 ns.

Table 7-10. ACCESS.bus Input Signals

Symbol	Parameter	Min	Max	Unit	Comments
t_{BUFi}	Bus free time between Stop and Start condition	$t_{SCLhigho}$			
t_{CSTOsi}	AB1C/AB2C setup time	$8 * t_{CLK} - t_{SCLri}$			Before Stop condition
t_{CSTRhi}	AB1C/AB2C hold time	$8 * t_{CLK} - t_{SCLri}$			After Start condition
t_{CSTRsi}	AB1C/AB2C setup time	$8 * t_{CLK} - t_{SCLri}$			Before Start condition
t_{DHCsi}	Data high setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge
t_{DLCsi}	Data low setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge
t_{SCLfi}	AB1D/AB2D fall time		300	ns	
t_{SCLri}	AB1D/AB2D rise time		1	µs	
$t_{SCLlowi}$	AB1C/AB2C low time	$16 * t_{CLK}$			After AB1C/AB2C falling edge
$t_{SCLhighi}$	AB1C/AB2C high time	$16 * t_{CLK}$			After AB1C/AB2C rising edge
t_{SDAfi}	AB1D/AB2D fall time		300	ns	
t_{SDAri}	AB1D/AB2D rise time		1	µs	
t_{SDAhi}	AB1D/AB2D hold time	0			After AB1C/AB2C falling edge
t_{SDAsi}	AB1D/AB2D setup time	$2 * t_{CLK}$			Before AB1C/AB2C rising edge

Table 7-11. ACCESS.bus Output Signals

Symbol	Parameter	Min	Max	Unit	Comments
$t_{SCLhigho}$	AB1C/AB2C high time	$K * t_{CLK}$ - 1 µs			After AB1C/AB2C rising edge[1]
$t_{SCLlowo}$	AB1C/AB2C low time	$K * t_{CLK}$ - 1 µs			After AB1C/AB2C falling edge
t_{BUFo}	Bus free time between Stop and Start condition	$t_{SCLhigho}$[2]	1	µs	
t_{CSTOso}	AB1C/AB2C setup time	$t_{SCLhigho}$[2]	1	µs	Before Stop condition
t_{CSTRho}	AB1C/AB2C hold time	$t_{SCLhigho}$[2]	1	µs	After Start condition
t_{CSTRso}	AB1C/AB2C setup time	$t_{SCLhigho}$[2]	1	µs	Before Start condition
t_{DHCso}	Data high setup time	$t_{SCLhigho}$[2] - t_{SDAro}	1	µs	Before AB1C/AB2C rising edge
t_{DLCso}	Data low setup time	$t_{SCLhigho}$[2] - t_{SDAfo}	1	µs	Before AB1C/AB2C rising edge
t_{SCLfo}	AB1D/AB2D signal fall time		300	ns	
t_{SCLro}	AB1D/AB2D signal rise time		1	µs	

Electrical Specifications (Continued)

Table 7-11. ACCESS.bus Output Signals (Continued)

Symbol	Parameter	Min	Max	Unit	Comments
t_{SDAfo}	AB1D/AB2D signal fall time		300	ns	
t_{SDAro}	AB1D/AB2D signal rise time		1	µs	
t_{SDAho}	AB1D/AB2D hold time	$7 * t_{CLK} - t_{SCLfo}$			After AB1C/AB2C falling edge
t_{SDAvo}	AB1D/AB2D valid time		$7 * t_{CLK} + t_{RD}$		After AB1C/AB2C falling edge

1. K is determined by bits [7:1] of the ACBCTL2 register (LDN 05h/06h, Offset 05h).
2. This value depends on the signal capacitance and the pull-up value of the relevant pin.

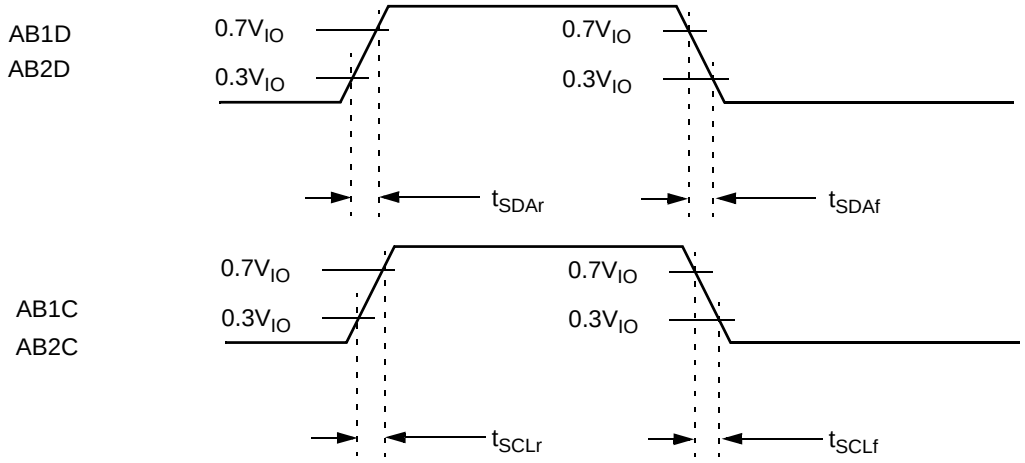


Figure 7-6. ACB Signals (AB1D, AB2D, AB1C and AB2C) Rising Time and Falling Times

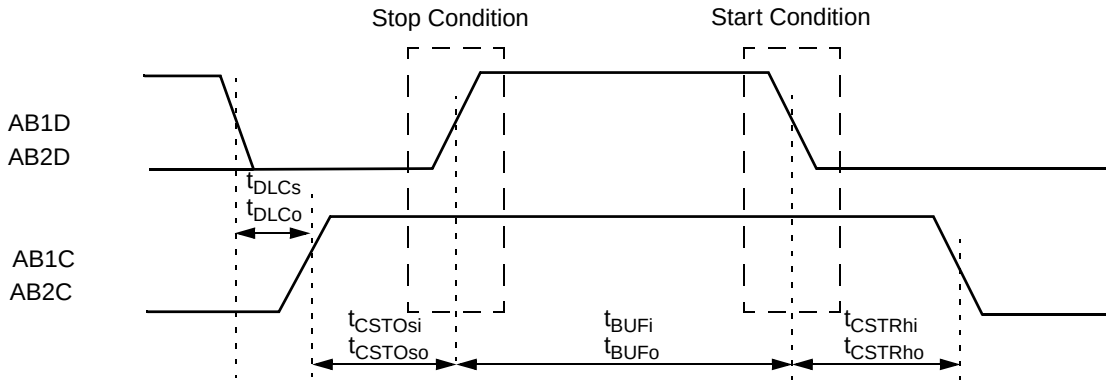


Figure 7-7. ACB Start and Stop Condition Timing

Electrical Specifications (Continued)

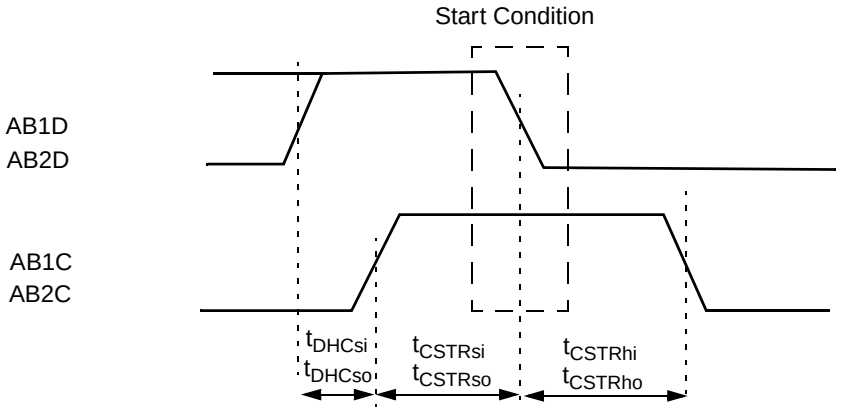


Figure 7-8. ACB Start Condition TIming

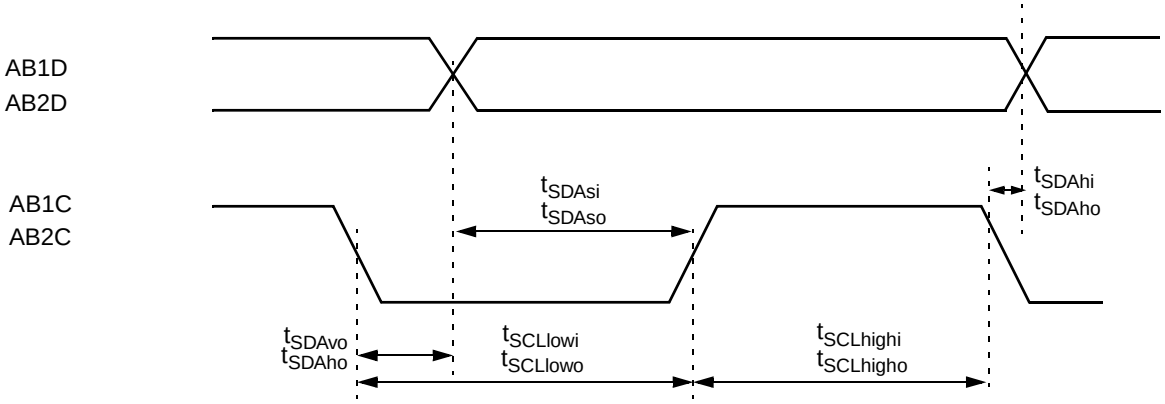


Figure 7-9. ACB Data Bit Timing

Electrical Specifications (Continued)

7.3.3 PCI Bus

The SC1100 is compliant with PCI Bus Rev. 2.1 specifications. Relevant information from the PCI Bus specifications is provided below. Not all parameters in Table 7-12 are 100% tested. The parameters in this table are further described in Figure 7-11.

Table 7-12. AC Specifications

Symbol	Parameter	Min	Max	Unit	Comments
$I_{OH}(AC)$[1, 2]	Switching Current High	$-12V_{IO}$		mA	$0 < V_{OUT} \leq 0.3V_{IO}$
		$-17.1(V_{IO}-V_{OUT})$		mA	$0.3V_{IO} < V_{OUT} < 0.9V_{IO}$
			Equation A (Figure 7-11)		$0.7V_{IO} < V_{OUT} < V_{IO}$
	Test Point[2]		$-32V_{IO}$	mA	$V_{OUT} = 0.7V_{IO}$
$I_{OL}(AC)$[1]	Switching Current Low	$16V_{IO}$		mA	$V_{IO} > V_{OUT} \geq 0.6V_{IO}$
		$26.7V_{OUT}$		mA	$0.6V_{IO} > V_{OUT} > 0.1V_{IO}$[1]
			Equation B (Figure 7-11)		$0.18V_{IO} > V_{OUT} > 0$[1, 2]
	Test Point[2]		$38V_{IO}$	mA	$V_{OUT} = 0.18V_{IO}$
I_{CL}	Low Clamp Current	$-25+(V_{IN}+1)/0.015$		mA	$-3 < V_{IN} \leq -1$
I_{CH}	High Clamp Current	$25+(V_{IN}-V_{IO}-1)/0.015$		mA	$V_{IO}+4 > V_{IN} > V_{IO}+1$
$SLEW_R$[3]	Output Rise Slew Rate	1	4	V/ns	$0.2V_{IO}$ - $0.6V_{IO}$ Load
$SLEW_F$[3]	Output Fall Slew Rate	1	4	V/ns	$0.6V_{IO}$ - $0.2V_{IO}$ Load

1. Refer to the V/I curves in Figure 7-11. This specification does not apply to PCICLKO and PCIRST# which are system outputs.
2. Maximum current requirements are met when drivers pull beyond the first step voltage. Equations which define these maximum values (A and B) are provided with relevant diagrams in Figure 7-11. These maximum values are guaranteed by design.
3. Rise slew rate does not apply to open-drain outputs. This parameter is interpreted as the cumulative edge rate across the specified range, according to the test circuit in Figure 7-10.

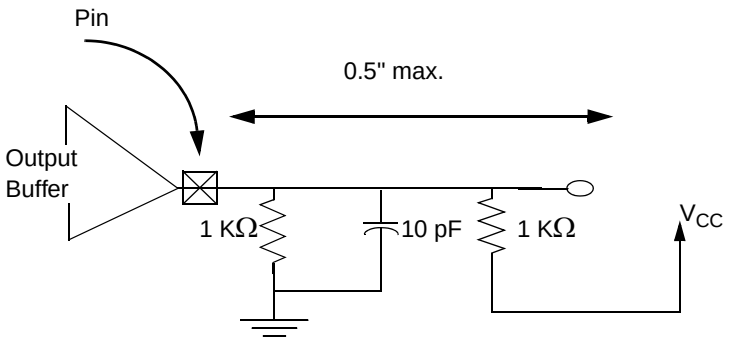
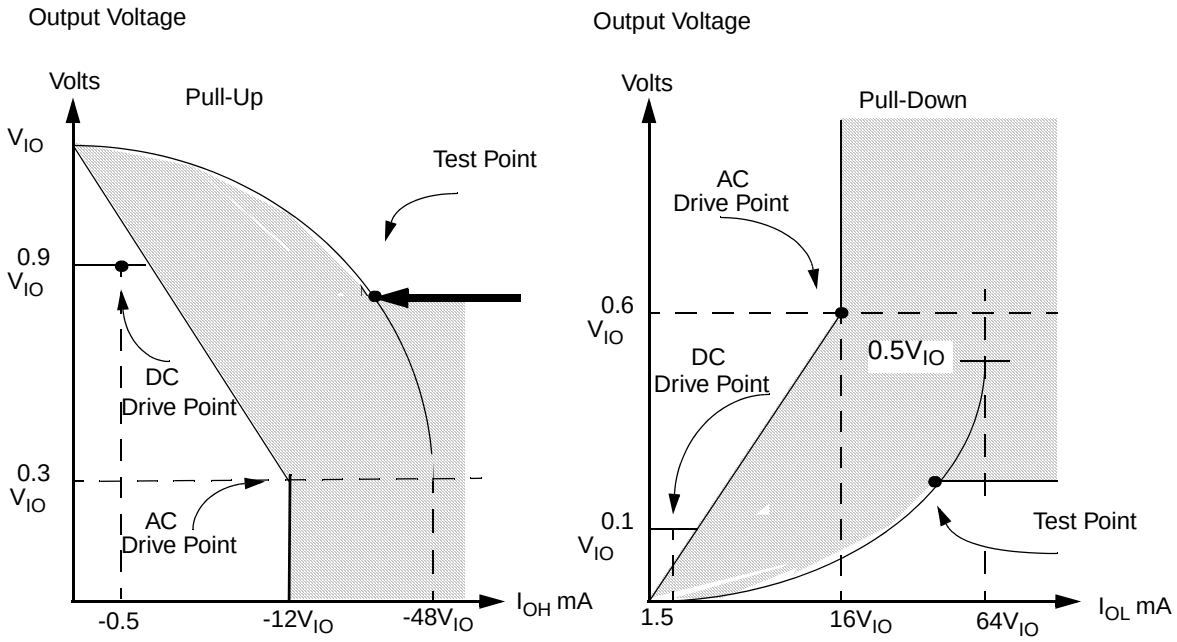


Figure 7-10. Testing Setup for Slew Rate and Minimum Timing

Electrical Specifications (Continued)



Equation A

$I_{OH} = (98.0/V_{IO})*(V_{OUT}-V_{IO})*(V_{OUT}+0.4V_{IO})$
for $V_{IO}>V_{OUT}>0.7V_{IO}$

Equation B

$I_{OL} = (256/V_{IO})*V_{OUT}*(V_{IO}-V_{OUT})$
for $0V<V_{OUT}<0.18V_{IO}$

Figure 7-11. V/I Curves for PCI Output Signals

Table 7-13. PCI Clock Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{CYC}	PCICLK Cycle time[1]	30		ns	
t_{HIGH}	PCICLK High time[2]	11		ns	
t_{LOW}	PCICLK Low time[2]	11		ns	
$PCICLK_{sr}$	PCICLK Slew Rate[3]	1	4	V/ns	
$PCIRST_{sr}$	PCIRST# Slew Rate[4]	50	-	mV/ns	

1. Clock frequency is between nominal DC and 33 MHz. Device operational parameters at frequencies under 16 MHz are not 100% tested. The clock can only be stopped in a low state.
2. Guaranteed by characterization.
3. Slew rate must be met across the minimum peak-to-peak portion of the clock waveform (see Figure 7-12).
4. The minimum PCIRST# slew rate applies only to the rising (deassertion) edge of the reset signal. See Figure 7-16 for PCIRST# timing.

Electrical Specifications (Continued)

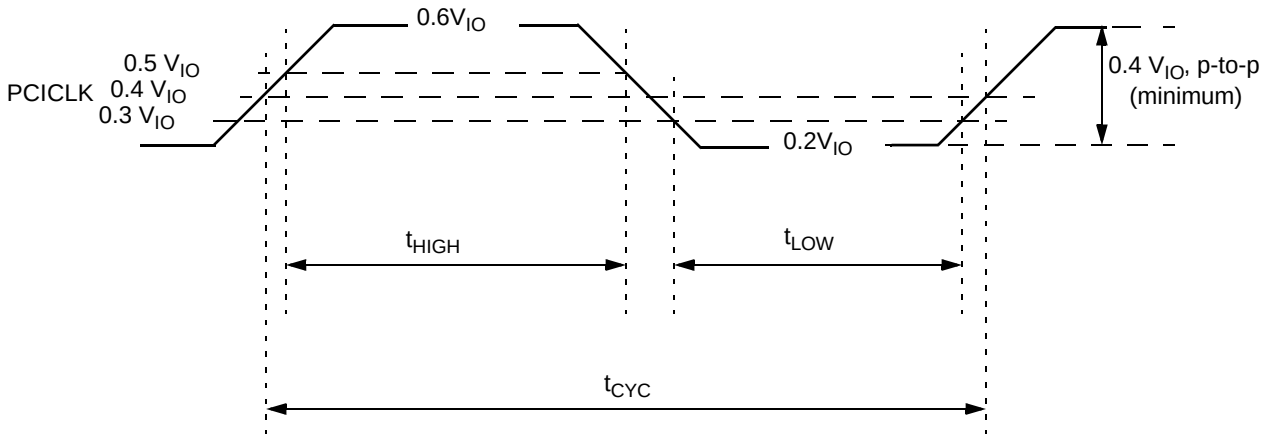


Figure 7-12. PCICLK Timing and Measurement Points

Table 7-14. PCI Bus Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{VAL}	PCICLK to Signal Valid Delay[1,2,4] (on the bus)	2	11	ns	
t_{VAL}(ptp)	PCICLK to Signal Valid Delay[1,2,4] (point-to-point)	2	12	ns	
t_{ON}	Float to Active Delay[1,3]	2		ns	
t_{OFF}	Active to Float Delay[1,3]		28	ns	
t_{SU}	Input Setup Time to PCICLK[4,5] (on the bus)	7		ns	
t_{SU}(ptp)	Input Setup Time to PCICLK[4,5] (point-to-point)	10,12		ns	
t_{H}	Input Hold Time from PCICLK[5]	0		ns	
t_{RST}	PCIRST# Active Time After Power Stable[6,3]	1		ms	
$t_{RST-CLK}$	PCIRST# Active Time After PCICLK Stable[6,3]	100		µs	
$t_{RST-OFF}$	PCIRST# Active to Output Float Delay[3,6,7,]		40	ns	

1. See the timing measurement conditions in Figure 7-14.
2. Minimum times are evaluated with same load used for slew rate measurement (as shown in note 3 of Table); maximum times are evaluated with the load circuits shown in Figure 7-13, for high-going and low-going edges respectively.
3. Not 100% tested.
4. REQ# and GNT# are point-to-point signals, and have different output valid delay and input setup times than do signals on the bus. GNT# has a setup time of 10 ns; REQ# has a setup time of 12 ns. All other signals are sent via the bus.
5. See the timing measurement conditions in Figure 7-15.
6. PCIRST# is asserted and deasserted asynchronously with respect to PCICLK (see Figure 7-16).
7. All output drivers are asynchronously floated when PCIRST# is active.

Electrical Specifications (Continued)

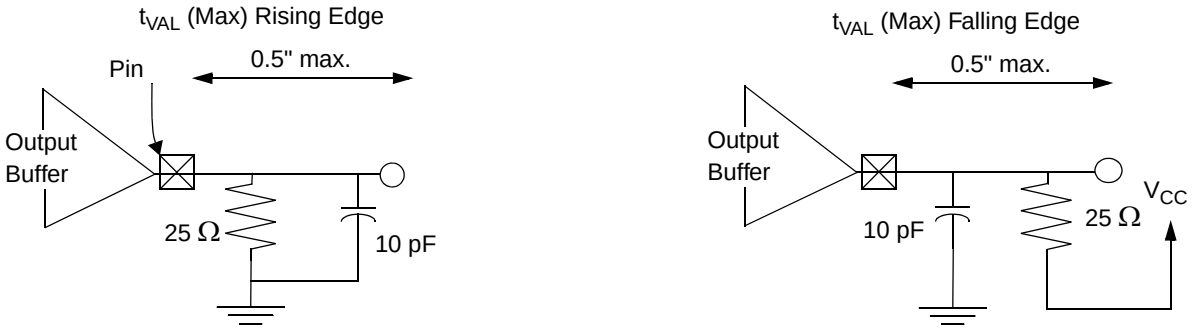


Figure 7-13. Load Circuits for Maximum Time Measurements

7.3.3.1 Measurement and Test Conditions

Table 7-15. Measurement Condition Parameters

Symbol	Value	Unit
V_{TH}[1]	0.6 V_{IO}	V
V_{TL}[1]	0.2 V_{IO}	V
V_{TEST}	0.4 V_{IO}	V
V_{STEP} (Rising Edge)	0.285 V_{IO}	V
V_{STEP} (Falling Edge)	0.615 V_{IO}	V
V_{MAX}[2]	0.4 V_{IO}	V
Input Signal Edge Rate	1	V/ns

1. The input test is performed with 0.1 V_{IO} of overdrive. Timing parameters must not exceed this overdrive.
2. V_{MAX} specifies the maximum peak-to-peak waveform allowed for measuring input timing.

Electrical Specifications (Continued)

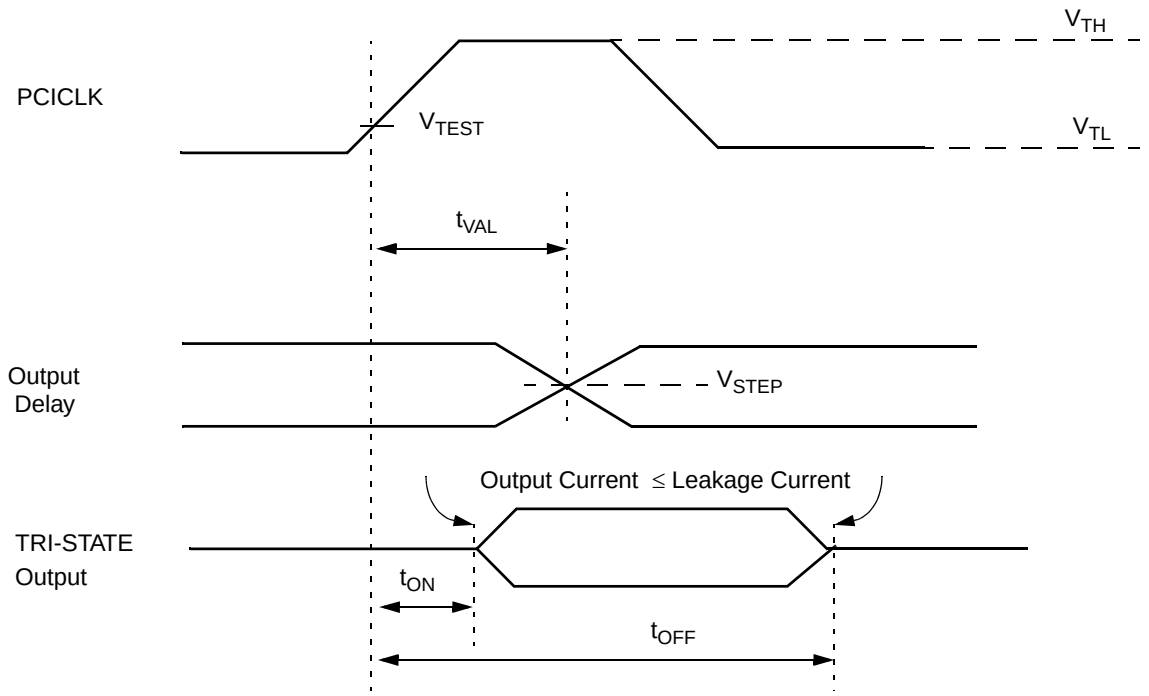


Figure 7-14. Output Timing Measurement Conditions

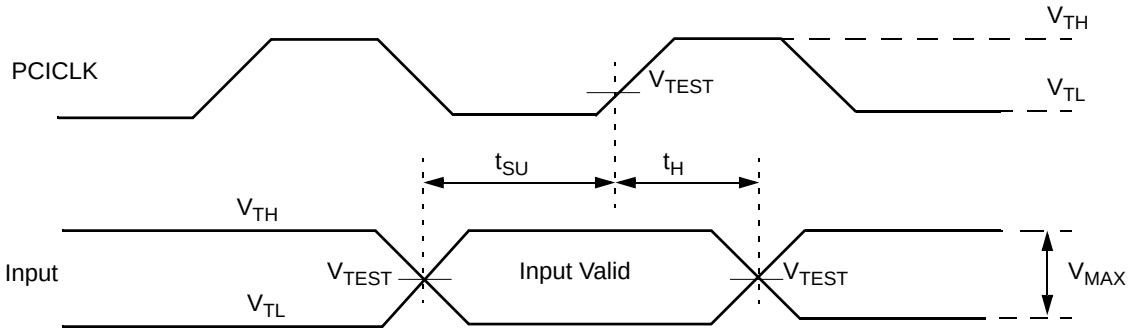
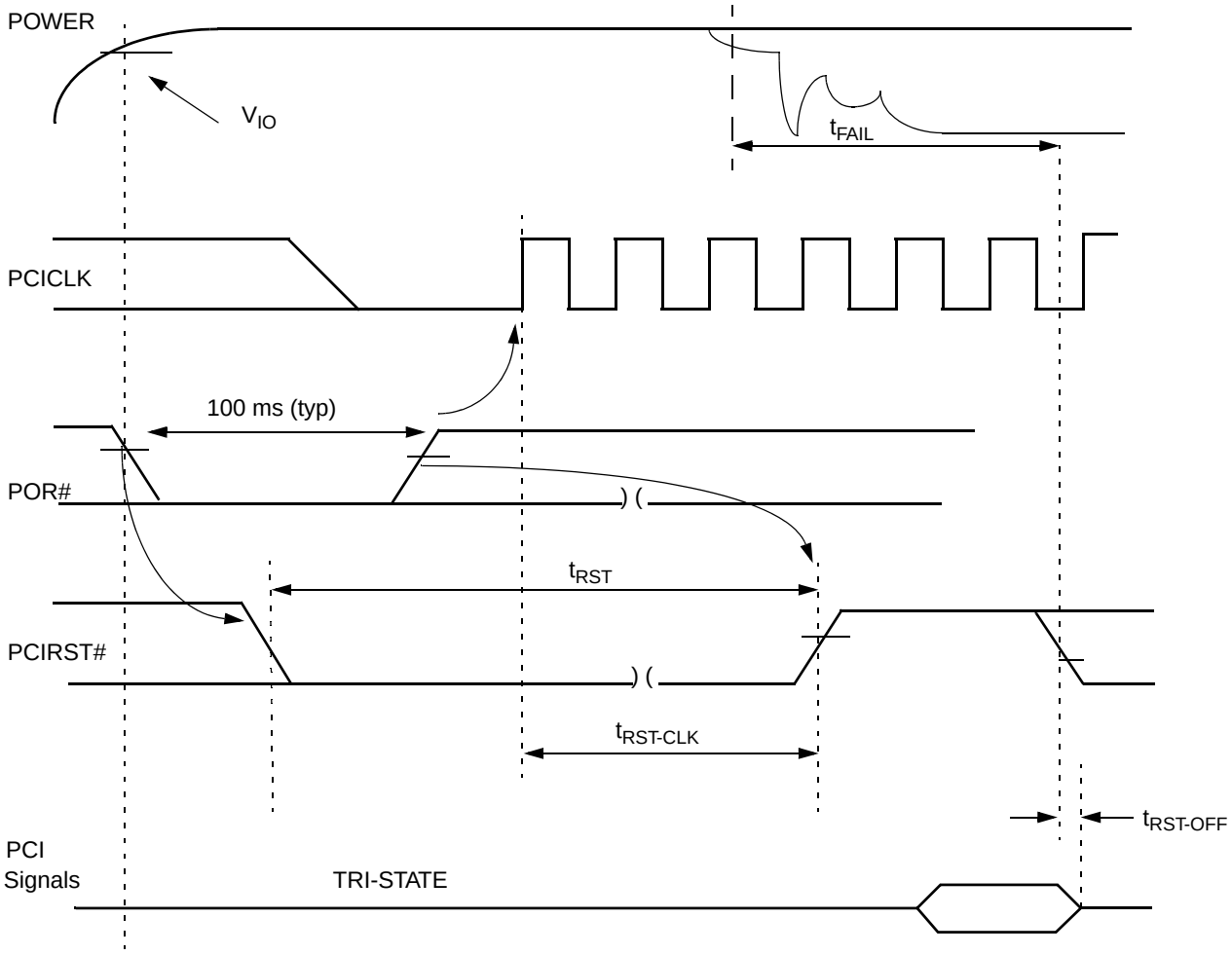


Figure 7-15. Input Timing Measurement Conditions

Electrical Specifications (Continued)



Note: The value of t_{FAIL} is 500 ns (maximum) from the power rail which exceeds specified tolerance by more than 500 mV.

Figure 7-16. Reset Timing

Electrical Specifications (Continued)

7.3.4 Sub-ISA Interface

All output timing is guaranteed for 50 pF load, unless otherwise specified.

The ISA Clock divisor (defined in F0 Index 50h[2:0] of the Core Logic module) is 011.

Table 7-16. Output Signals

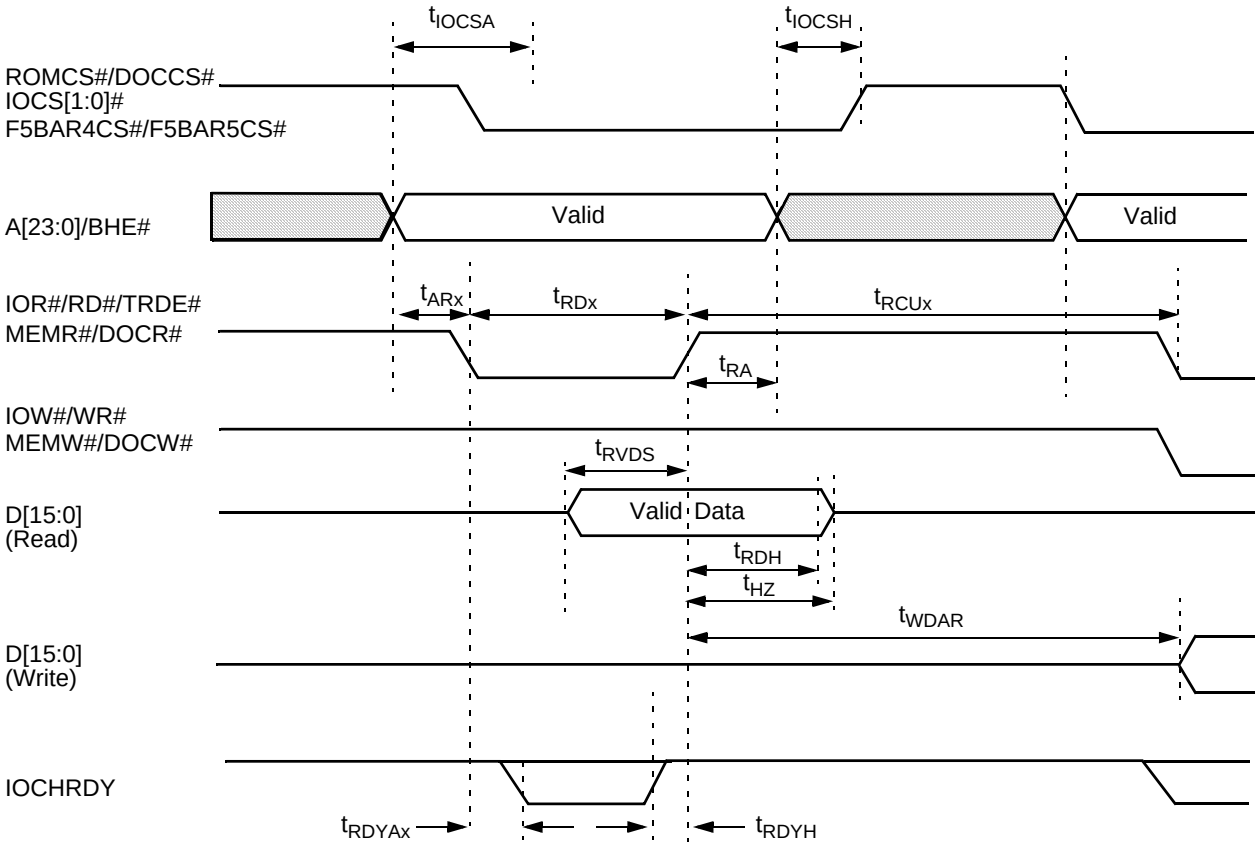
Symbol	Parameter	Bus Width (Bits)	Type	Min (ns)	Max (ns)	Figure	Comments
t_{RD1}	MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	16	M	225		7-17	Standard
t_{RD2}	MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	16	M	105		7-17	Zero wait state
t_{RD3}	IOR#/RD#/TRDE# Read active pulse width FE to RE	16	I/O	160		7-17	Standard
t_{RD4}	IOR#/MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	8	M, I/O	520		7-17	Standard
t_{RD5}	IOR#/MEMR#/DOCR#/RD#/TRDE# Read active pulse width FE to RE	8	M, I/O	160		7-17	Zero wait state
t_{RCU1}	MEMR#/DOCR#/RD#/TRDE# inactive pulse width	16	M	103		7-17	
t_{RCU2}	MEMR#/DOCR#/RD#/TRDE# inactive pulse width	8	M	163		7-17	
t_{RCU3}	IOR#/RD#/TRDE# inactive pulse width	8, 16	I/O	163		7-17	
t_{WR1}	MEMW#/WR# Write active pulse width FE to RE	16	M	225		7-18	Standard
t_{WR2}	MEMW#/DOCW#/WR# Write active pulse width FE to RE	16	M	105		7-18	Zero wait state
t_{WR3}	IOW#/WR# Write active pulse width FE to RE	16	I/O	160		7-18	Standard
t_{WR4}	IOW#/MEMW#/DOCW#/WR# Write active pulse width FE to RE	8	M, I/O	520		7-18	Standard
t_{WR5}	IOW#/MEMW#/DOCW#/WR# Write active pulse width FE to RE	8	M, I/O	160		7-18	Zero wait state
t_{WCU1}	MEMW#/WR#/DOCW# inactive pulse width	16	M	103		7-18	
t_{WCU2}	MEMW#/WR#/DOCW# inactive pulse width	8	M	163		7-18	
t_{WCU3}	IOW#/WR# inactive pulse width	8, 16	I/O	163		7-18	
t_{RDYH}	IOR#/MEMR#/RD#/DOCR#/IOW#/ MEMW#/WR#/DOCW# Hold after IOCHRDY RE	8, 16	M, I/O	120		7-17 7-18	
t_{RDYA1}	IOCHRDY valid after IOR#/MEMR#/ RD#/DOCR#/IOW#/MEMW#/WR#/ DOCW# FE	16	M, I/O		78	7-17 7-18	

Electrical Specifications (Continued)

Table 7-16. Output Signals (Continued)

Symbol	Parameter	Bus Width (Bits)	Type	Min (ns)	Max (ns)	Figure	Comments
t_{RDYA2}	IOCHRDY valid after IOR#/MEMR#/ RD#/DOCR#/IOW#/MEMW#/WR#/ DOCW# FE	8	M, I/O		366	7-17 7-18	
t_{IOCSA}	IOCS[1:0]#/DOCS#/ROMCS#/F5BAR 4CS#/ F5BAR5CS# driven active from A[23:0] valid	8, 16	M, I/O		34	7-17 7-18	
t_{IOCSH}	IOCS[1:0]#/DOCS#/ROMCS#/F5BAR 4CS#/ F5BAR5CS# valid Hold after A[23:0] invalid	8, 16	M, I/O	0		7-17 7-18	
t_{AR1}	A[23:0]/BHE# valid before MEMR#/DOCR# active	16	M	34		7-17	
t_{AR2}	A[23:0]/BHE# valid before IOR# active	16	I/O	100		7-17	
t_{AR3}	A[23:0]/BHE# valid before MEMR#/DOCR#/IOR# active	8	M, I/O	100		7-17	
t_{RA}	A[23:0]/BHE# valid Hold after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	25		7-17	
t_{RVDS}	Read data D[15:0] valid setup before MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	24		7-17	
t_{RDH}	Read data D[15:0] valid Hold after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O	0		7-17	
t_{HZ}	Read data floating after MEMR#/DOCR#/IOR# inactive	8, 16	M, I/O		41	7-17	
t_{AW1}	A[23:0]/BHE# valid before MEMW#/DOCW# active	16	M	34		7-18	
t_{AW2}	A[23:0]/BHE# valid before IOW# active	16	I/O	100		7-18	
t_{AW3}	A[23:0]/BHE# valid before MEMW#/DOCW#/IOW# active	8	M, I/O	100		7-18	
t_{WA}	A[23:0]/BHE# valid Hold after MEMW#/DOCW#/IOW# invalid	8, 16	M, I/O	25		7-18	
t_{DV1}	Write data D[15:0] valid after MEMW#/DOCW# active	8, 16	M	40		7-18	
t_{DV2}	Write data D[15:0] valid after IOW# active	8	I/O	40		7-18	
t_{DV3}	Write data D[15:0] valid after IOW# active	16	I/O	-23		7-18	
t_{WTR}	TRDE# inactive after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O	20		7-18	
t_{DH}	Write data D[15:0] after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O	45		7-18	
t_{DF}	Write data D[15:0] goes TRI-STATE after MEMW#/DOCW#/IOW# inactive	8, 16	M, I/O		105	7-18	
t_{WDAR}	Write data D[15:0] after read MEMR#/DOCR#/IOR#	8, 16	M, I/O	41		7-17	

Electrical Specifications (Continued)



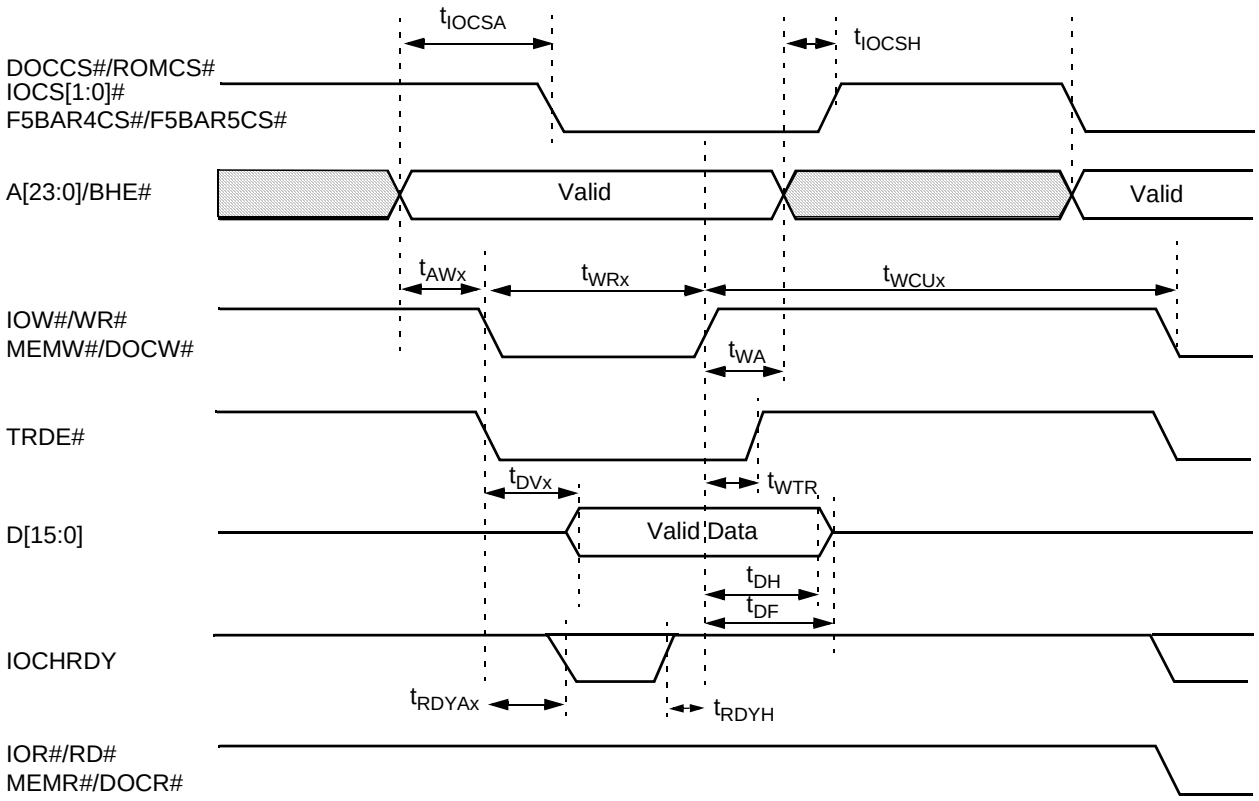
Notes:

1) x indicates a numeric index for the relevant symbol.

2) IOCHRDY is not available externally.

Figure 7-17. Sub-ISA Read Operation

Electrical Specifications (Continued)



Notes:

1) x indicates a numeric index for the relevant symbol.

2) IOCHRDY is not available externally.

Figure 7-18. Sub-ISA Write Operation

Electrical Specifications (Continued)

7.3.5 LPC Interface

Table 7-17. LPC and SERIRQ Signals

Symbol	Parameter	Min	Max	Unit	Comments
t_{VAL}	Output Valid delay	0	17	ns	After PCICLK rising edge
t_{ON}	Float to Active delay	2		ns	After PCICLK rising edge
t_{OFF}	Active to Float delay		28	ns	After PCICLK rising edge
t_{SU}	Input Setup time	7		ns	Before PCICLK rising edge
t_{HI}	Input Hold time	0		ns	After PCICLK rising edge

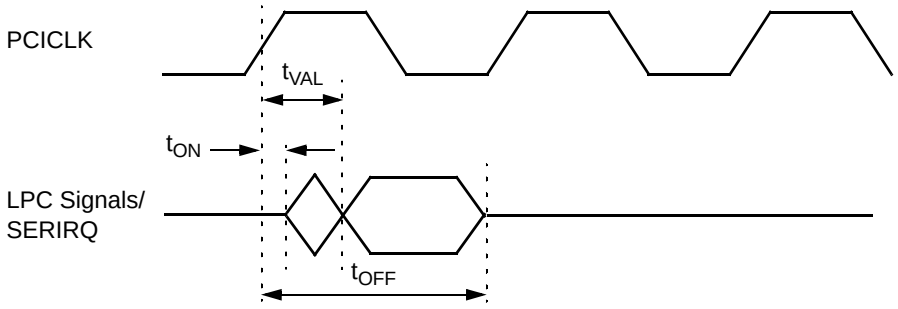


Figure 7-19. LPC Output Timing

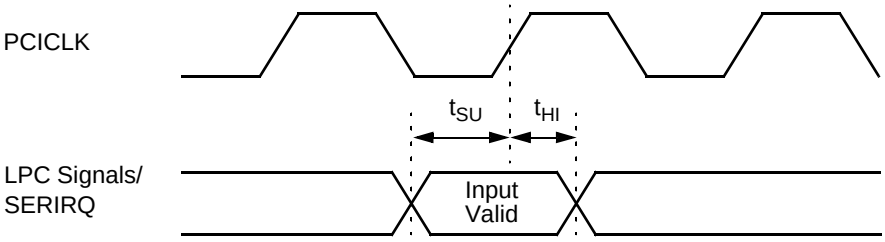


Figure 7-20. LPC Input Timing

Electrical Specifications (Continued)

7.3.6 IDE Interface Timing

Table 7-18. General Timing of the IDE Interface

Symbol	Parameter	Min	Max	Unit	Comments
t_{IDE_FALL}	Fall time of all IDE signals. From $0.9V_{IO}$ to $0.1V_{IO}$	5		ns	C_L = 40 pF
t_{IDE_RISE}	Rise time of all IDE signals. From $0.1V_{IO}$ to $0.9V_{IO}$	5		ns	C_L = 40 pF
$t_{IDE_RST_PW}$	IDE_RST# pulse width	25		µs	

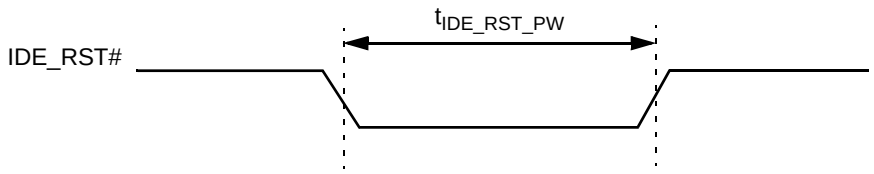


Figure 7-21. IDE Reset Timing

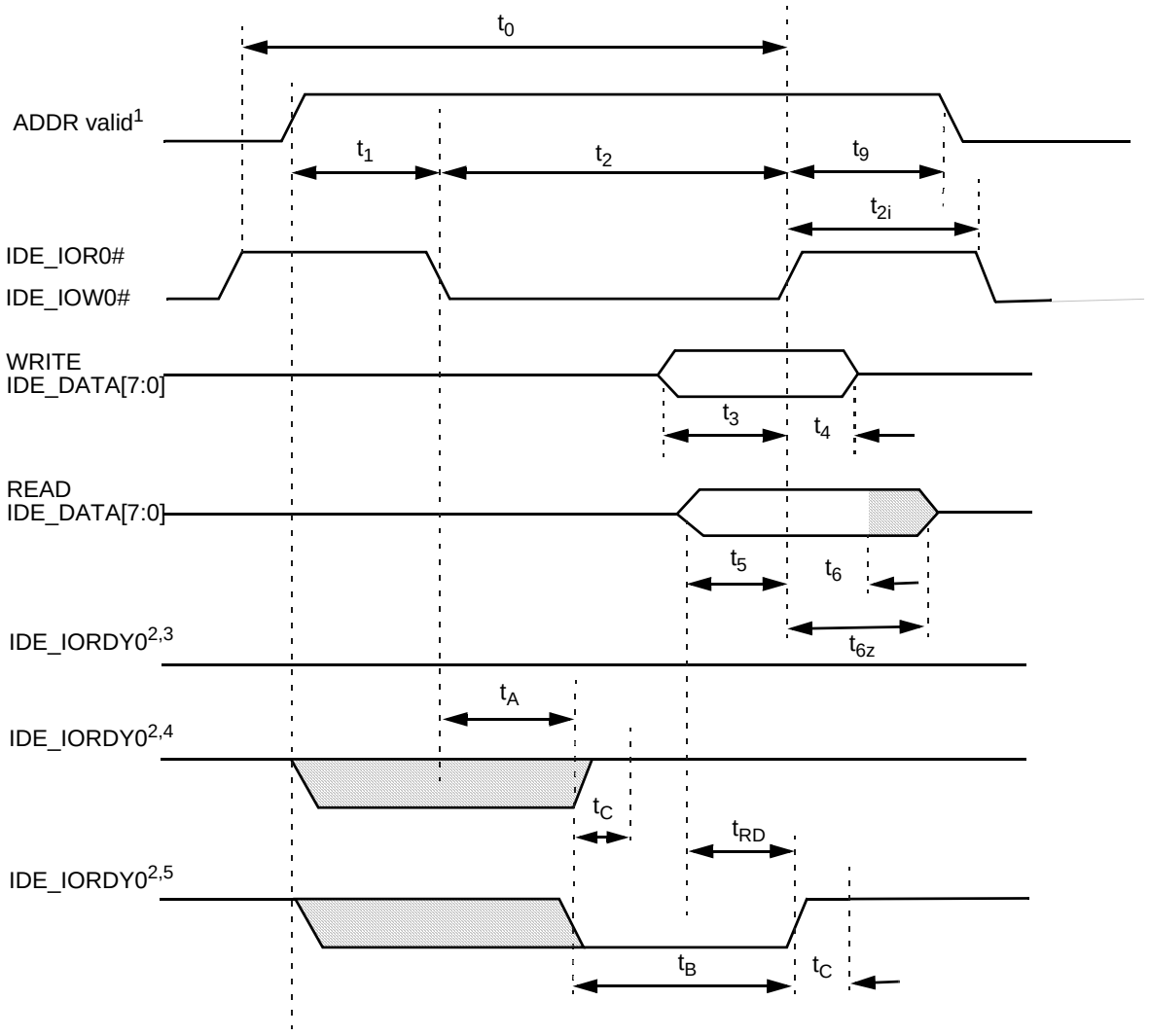
Electrical Specifications (Continued)

Table 7-19. Register Transfer to/from Device

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)	Mode 3 (ns)	Mode 4 (ns)
t_0	Cycle time[1] (min)	600	383	240	180	120
t_1	Address valid to IDE_IOR[0:1]#/ IDE_IOW[0:1]# setup (min)	70	50	30	30	25
t_2	IDE_IOR[0:1]#/IDE_IOW[0:1]# pulse width 8-bit[1] (min)	290	290	290	80	70
t_{2i}	IDE_IOR[0:1]#/IDE_IOW[0:1]# recovery time[1] (min)	-	-	-	70	25
t_3	IDE_IOW[0:1]# data setup (min)	60	45	30	30	20
t_4	IDE_IOW[0:1]# data hold (min)	30	20	15	10	10
t_5	IDE_IOR[0:1]# data setup (min)	50	35	20	20	20
t_6	IDE_IOR[0:1]# data hold (min)	5	5	5	5	5
t_{6Z}	IDE_IOR[0:1]# data TRI-STATE[2] (max)	30	30	30	30	30
t_9	IDE_IOR[0:1]#/IDE_IOW[0:1]# to address valid hold (min)	20	15	10	10	10
t_{RD}	Read Data Valid to IDE_IORDY[0:1] active (if IDE_IORDY[0:1] initially low after t_A (min)	0	0	0	0	0
t_A	IDE_IORDY[0:1] Setup time[3]	35	35	35	35	35
t_B	IDE_IORDY[0:1] Pulse Width (max)	1250	1250	1250	1250	1250
t_C	IDE_IORDY[0:1] Assertion to release (max)	5	5	5	5	5

1. t_0 is the minimum total cycle time, t_2 is the minimum command active time, and t_{2i} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_2, and t_{2i} are met. The minimum total cycle time requirements is greater than the sum of t_2 and t_{2i}. (This means that a host implementation can lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)
2. This parameter specifies the time from the rising edge of IDE_IOR[0:1]# to the time that the data bus is no longer driven by the device (TRI-STATE).
3. The delay from the activation of IDE_IOR[0:1]# or IDE_IOW[0:1]# until the state of IDE_IORDY[0,1] is first sampled. If IDE_IORDY[0:1] is inactive, then the host waits until IDE_IORDY[0:1] is active before the PIO cycle is completed. If the device is not driving IDE_IORDY[0:1] negated after activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_5 is met and t_{RD} is not applicable. If the device is driving IDE_IORDY[0:1] negated after activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_{RD} is met and t_5 is not applicable.

Electrical Specifications (Continued)



Notes:

1) Device address consists of signals IDE_CS[0:1]# and IDE_ADDR[2:0].

2) Negation of IDE_IORDY0,1 is used to extend the PIO cycle. The determination of whether or not the cycle is to be extended is made by the host after t_A from the assertion of IDE_IOR[0:1]# or IDE_IOW[0:1]#.

3) Device never negates IDE_IORDY[0:1]. Device keeps IDE_IORDY[0:1] released, and no wait is generated.

4) Device negates IDE_IORDY[0:1] before t_A but causes IDE_IORDY[0:1] to be asserted before t_A. IDE_IORDY[0:1] is released, and no wait is generated.

5) Device negates IDE_IORDY[0:1] before t_A. IDE_IORDY[0:1] is released prior to negation and may be asserted for no more than 5 ns before release. A wait is generated.

6) The cycle completes after IDE_IORDY[0:1] is reasserted. For cycles where a wait is generated and IDE_IOR[0:1] is asserted, the device places read data on IDE_DATA[15:0] for t_{RD} before asserting IDE_IORDY[0:1].

Figure 7-22. Register Transfer to/from Device

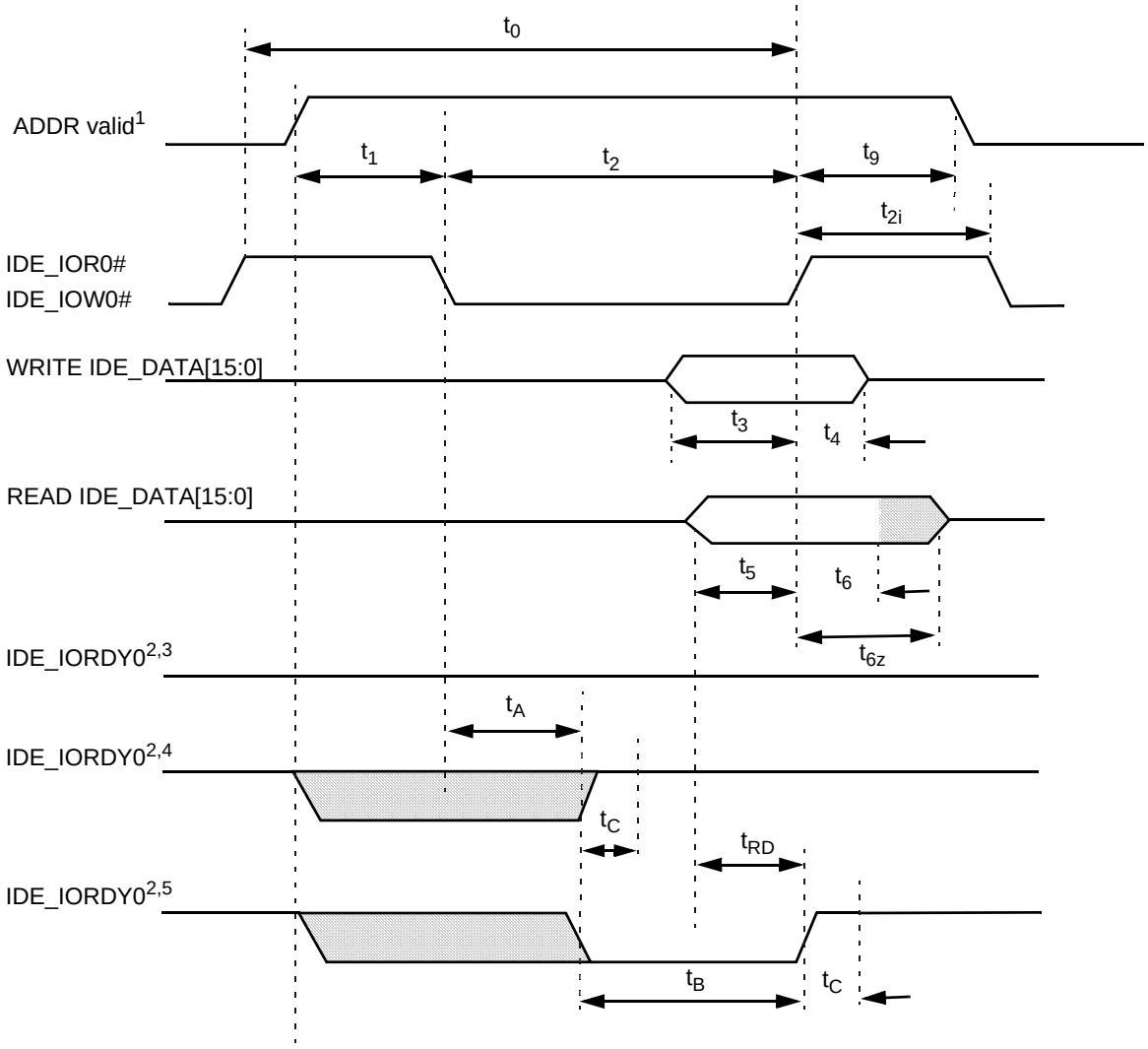
Electrical Specifications (Continued)

Table 7-20. PIO Data Transfer to/from Device

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)	Mode 3 (ns)	Mode 4 (ns)
t_0	Cycle time[1] (min)	600	383	240	180	120
t_1	Address valid to IDE_IOR[0:1]#/IDE_IOW[0:1]# setup (min)	70	50	30	30	25
t_2	IDE_IOR[0:1]#/IDE_IOW[0:1]# 16-bit[1] (min)	165	125	100	80	70
t_{2i}	IDE_IOR[0:1]#/IDE_IOW[0:1]# recovery time[1] (min)	-	-	-	70	25
t_3	IDE_IOW[0:1]# data setup (min)	60	45	30	30	20
t_4	IDE_IOW[0:1]# data hold (min)	30	20	15	10	10
t_5	IDE_IOR[0:1]# data setup (min)	50	35	20	20	20
t_6	IDE_IOR[0:1]# data hold (min)	5	5	5	5	5
t_{6Z}	IDE_IOR[0:1]# data TRI-STATE[2] (max)	30	30	30	30	30
t_9	IDE_IOR[0:1]#/IDE_IOW[0:1]# to address valid hold (min)	20	15	10	10	10
t_{RD}	Read Data Valid to IDE_IORDY[0,1] active (if IDE_IORDY[0:1] initially low after t_A) (min)	0	0	0	0	0
t_A	IDE_IORDY[0:1] Setup time[3]	35	35	35	35	35
t_B	IDE_IORDY[0:1] Pulse Width (max)	1250	1250	1250	1250	1250
t_C	IDE_IORDY[0:1] assertion to release (max)	5	5	5	5	5

1. t_0 is the minimum total cycle time, t_2 is the minimum command active time, and t_{2i} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_2, and t_{2i} are met. The minimum total cycle time requirement is greater than the sum of t_2 and t_{2i}. (This means that a host implementation may lengthen t_2 and/or t_{2i} to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)
2. This parameter specifies the time from the rising edge of IDE_IOR[0:1]# to the time that the data bus is no longer driven by the device (TRI-STATE).
3. The delay from the activation of IDE_IOR[0:1]# or IDE_IOW[0:1]# until the state of IDE_IORDY[0:1] is first sampled. If IDE_IORDY[0:1] is inactive, then the host waits until IDE_IORDY[0:1] is active before the PIO cycle is completed. If the device is not driving IDE_IORDY[0:1] negated after the activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_5 is met and t_{RD} is not applicable. If the device is driving IDE_IORDY[0:1] negated after the activation (t_A) of IDE_IOR[0:1]# or IDE_IOW[0:1]#, then t_{RD} is met and t_5 is not applicable.

Electrical Specifications (Continued)



Notes:

1) Device address consists of signals IDE_CS[0:1]# and IDE_ADDR[2:0].

2) Negation of IDE_IORDY[0:1] is used to extend the PIO cycle. The determination of whether or not the cycle is to be extended is made by the host after t_A from the assertion of IDE_IOR[0:1]# or IDE_IOW[0:1]#.

3) Device never negates IDE_IORDY[0:1]. Devices keep IDE_IORDY[0:1] released, and no wait is generated.

4) Device negates IDE_IORDY[0:1] before t_A but causes IDE_IORDY[0:1] to be asserted before t_A. IDE_IORDY[0:1] is released, and no wait is generated.

5) Device negates IDE_IORDY[0:1] before t_A. IDE_IORDY[0:1] is released prior to negation and may be asserted for no more than 5 ns before release. A wait is generated.

6) The cycle completes after IDE_IORDY[0:1] is reasserted. For cycles where a wait is generated and IDE_IOR[0:1]# is asserted, the device places read data on IDE_DATA[15:0] for t_{RD} before asserting IDE_IORDY[0:1].

Figure 7-23. PIO Data Transfer to/from Device

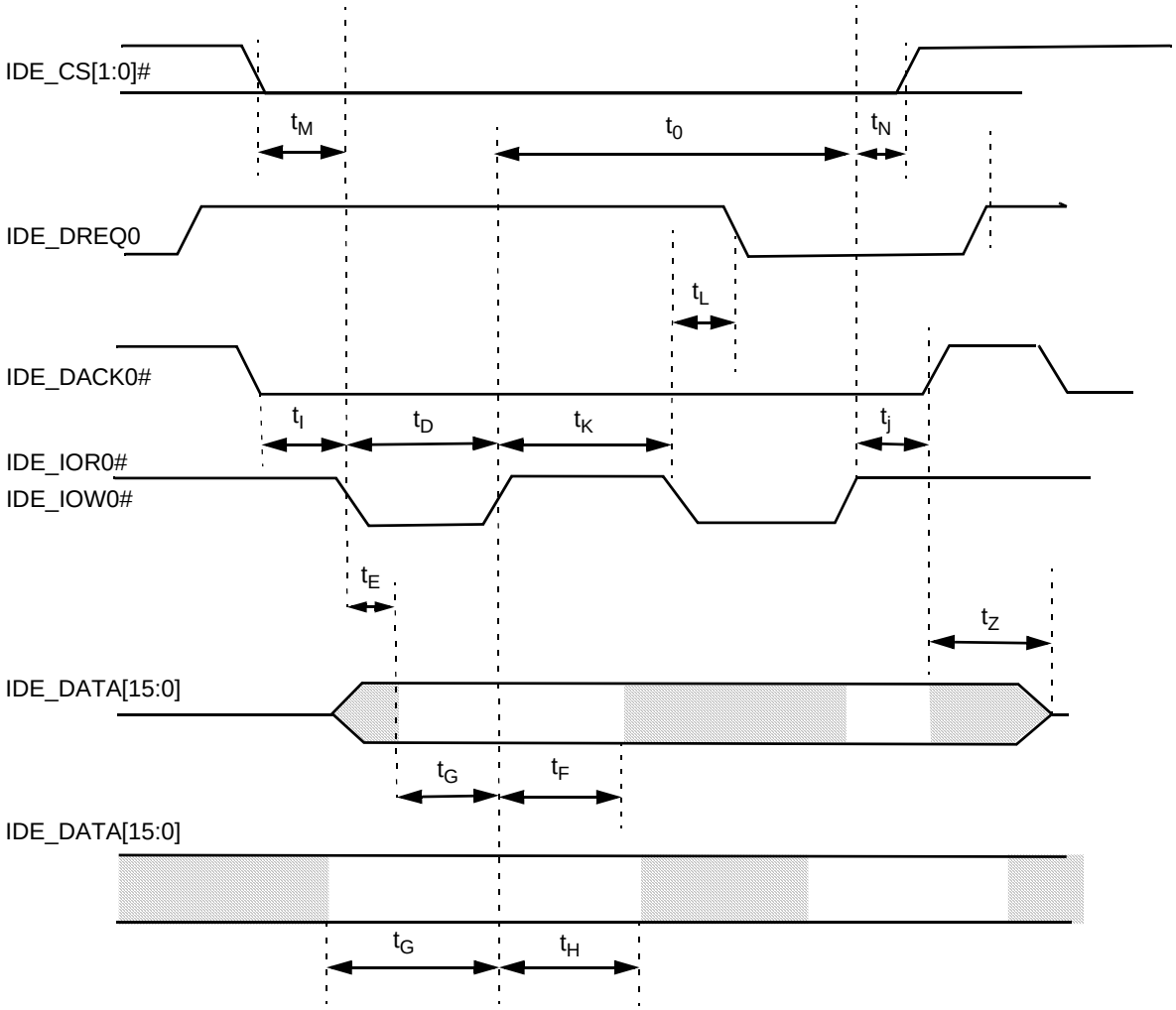
Electrical Specifications (Continued)

Table 7-21. Multiword DMA Data Transfer

Symbol	Parameter	Mode 0 (ns)	Mode 1 (ns)	Mode 2 (ns)
t_0	Cycle time[1] (min)	480	150	120
t_D	IDE_IOR[0:1]#/IDE_IOW[0:1]# (min)	215	80	70
t_E	IDE_IOR[0:1]# data access (max)	150	60	50
t_F	IDE_IOR[0:1]# data hold (min)	5	5	5
t_G	IDE_IOW[0:1]#/IDE_IOW[0:1]# data setup (min)	100	30	20
t_H	IDE_IOW[0:1]# data hold (min)	20	15	10
t_I	IDE_DACK[0:1]# to IDE_IOR[0:1]#/IDE_IOW[0:1]# setup (min)	0	0	0
t_J	IDE_IOR[0:1]#/IDE_IOW[0:1]# to IDE_DACK[0:1]# hold (min)	20	5	5
t_{KR}	IDE_IOR[0:1]# negated pulse width (min)	50	50	25
t_{KW}	IDE_IOW[0:1]# negated pulse width (min)	215	50	25
t_{LR}	IDE_IOR[0:1]# to IDE_DREQ[0:1] delay (max)	120	40	35
t_{LW}	IDE_IOW[0:1]# to IDE_DREQ0,1 delay (max)	40	40	35
t_M	IDE_CS[0:1]# valid to IDE_IOR[0:1]#/IDE_IOW[0:1]# (min)	50	30	25
t_N	IDE_CS[0:1]# hold	15	10	10
t_Z	IDE_DACK[0:1]# to TRI-STATE	20	25	25

1. t_0 is the minimum total cycle time, t_D is the minimum command active time, and t_{KR} or t_{KW} is the minimum command recovery time or command inactive time. The actual cycle time equals the sum of the command active time and the command inactive time. The three timing requirements of t_0, t_D and $t_{KR/KW}$, are met. The minimum total cycle time requirement t_0 is greater than the sum of t_D and $t_{KR/KW}$. (This means that a host implementation can lengthen t_D and/or $t_{KR/KW}$ to ensure that t_0 is equal to or greater than the value reported in the device's IDENTIFY DEVICE data.)

Electrical Specifications (Continued)



Notes:

1) For Multiword DMA transfers, the Device may negate IDE_DREQ[0:1] within the t_L specified time once IDE_DACK[0:1] is asserted, and reassert it again at a later time to resume the DMA operation. Alternatively, if the device is able to continue the transfer of data, the device may leave IDE_DREQ[0:1] asserted and wait for the host to reassert IDE_DACK[0:1]#.

2) This signal can be negated by the host to Suspend the DMA transfer in process.

Figure 7-24. Multiword DMA Data Transfer

Electrical Specifications (Continued)

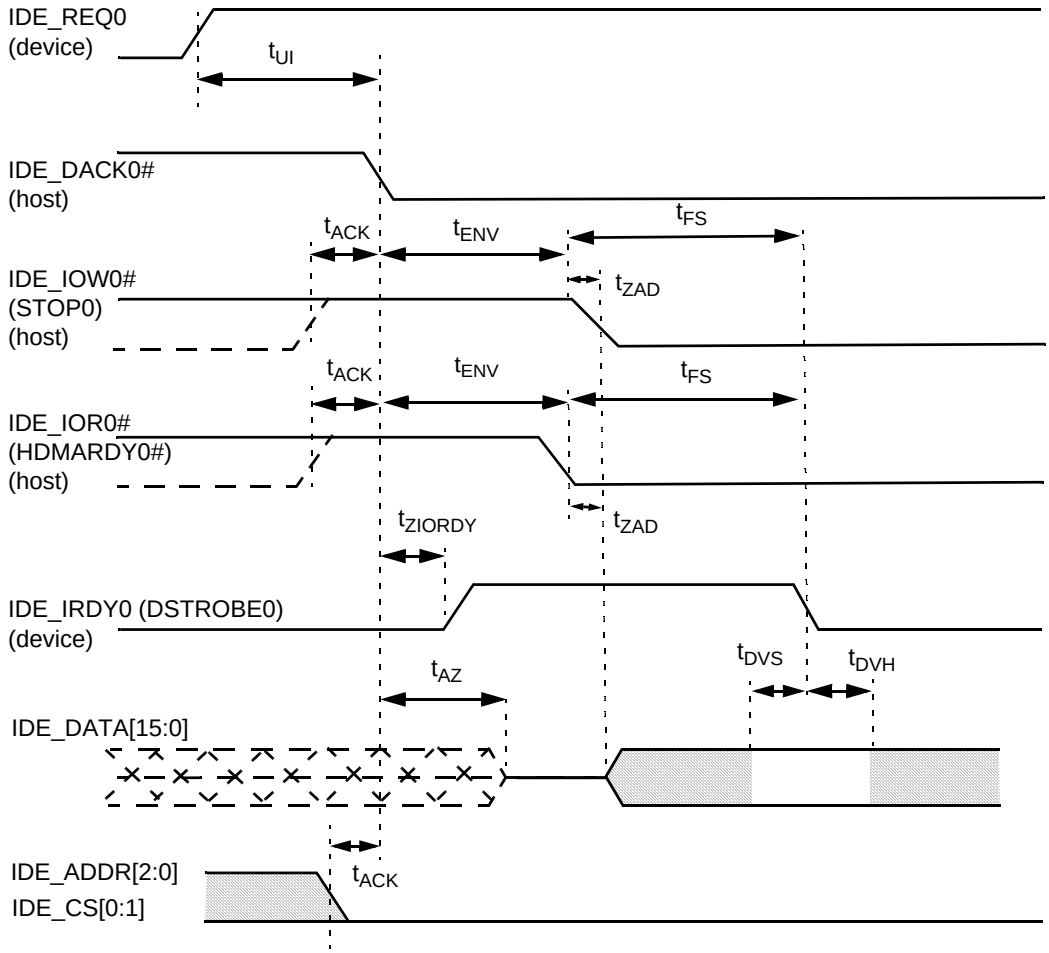
Table 7-22. UltraDMA Data Burst Timing Requirements

Symbol	Parameter	Mode 0 (ns)		Mode 1 (ns)		Mode 2 (ns)	
		Min	Max	Min	Max	Min	Max
t_{2CYC}	Typical sustained average two cycle time	240		160		120	
	Two cycle time allowing for clock variations (from rising edge to next rising edge or from falling edge to next falling edge of STROBE)	235		156		117	
t_{CYC}	Cycle time allowing for asymmetry and clock variations (from STROBE edge to STROBE edge)	114		75		55	
t_{DS}	Data setup time (at recipient)	15		10		7	
t_{DH}	Data hold time (at recipient)	5		5		5	
t_{DVS}	Data valid setup time at sender (from data bus being valid until STROBE edge)	70		48		34	
t_{DVH}	Data valid hold time at sender (from STROBE edge until data may become invalid)	6		6		6	
t_{FS}	First STROBE time (for device to first negate IDE_IRDY[0:1] (DSTROBE[0:1]) from IDE_IOW[0:1]# (STOP[0:1]) during a data in burst)	0	230	0	200	0	170
t_{LI}	Limited interlock time[1]	0	150	0	150	0	150
t_{MLI}	Interlock time with minimum[1]	20		20		20	
t_{UI}	Unlimited interlock time[1]	0		0		0	
t_{AZ}	Maximum time allowed for output drivers to release (from being asserted or negated)		10		10		10
t_{ZAH}	Minimum delay time required for output drivers to assert or negate (from released state)	20		20		20	
t_{ZAD}		0		0		0	
t_{ENV}	Envelope time (from IDE_DACK[0:1]# to IDE_IOW[0:1]# (STOP[0:1]) and IDE_IOR[0:1]# (HDMARDY[0:1]#) during data out burst initiation)	20	70	20	70	20	70
t_{SR}	STROBE to DMARDY time (if DMARDY# is negated before this long after STROBE edge, the recipient shall receive no more than one additional data WORD)		50		30		20
t_{RFS}	Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY#)		75		60		50
t_{RP}	Ready-to-pause time (time that recipient shall wait to initiate pause after negating DMARDY#)	160		125		100	
t_{IORDYZ}	Pull-up time before allowing IDE_IORDY[0:1] to be released		20		20		20
t_{ZIORDY}	Minimum time device shall wait before driving IDE_IORDY[0:1]	0		0		0	
T_{ACK}	Setup and hold times for IDE_DACK[0:1]# (before assertion or negation)	20		20		20	
T_{SS}	Time from STROBE edge to negation of IDE_DREQ[0:1] or assertion of IDE_IOW[0:1]# (STOP[0:1]) (when sender terminates a burst)	50		50		50	

1. t_{UI}, t_{MLI}, and t_{LI} indicate sender-to-recipient or recipient-to-sender interlocks, that is, one agent (either sender or recipient) is waiting for the other agent to respond with a signal before proceeding. t_{UI} is an unlimited interlock with no maximum time value. t_{MLI} is a limited time-out with a defined minimum. t_{LI} is a limited time-out with a defined maximum.

Electrical Specifications (Continued)

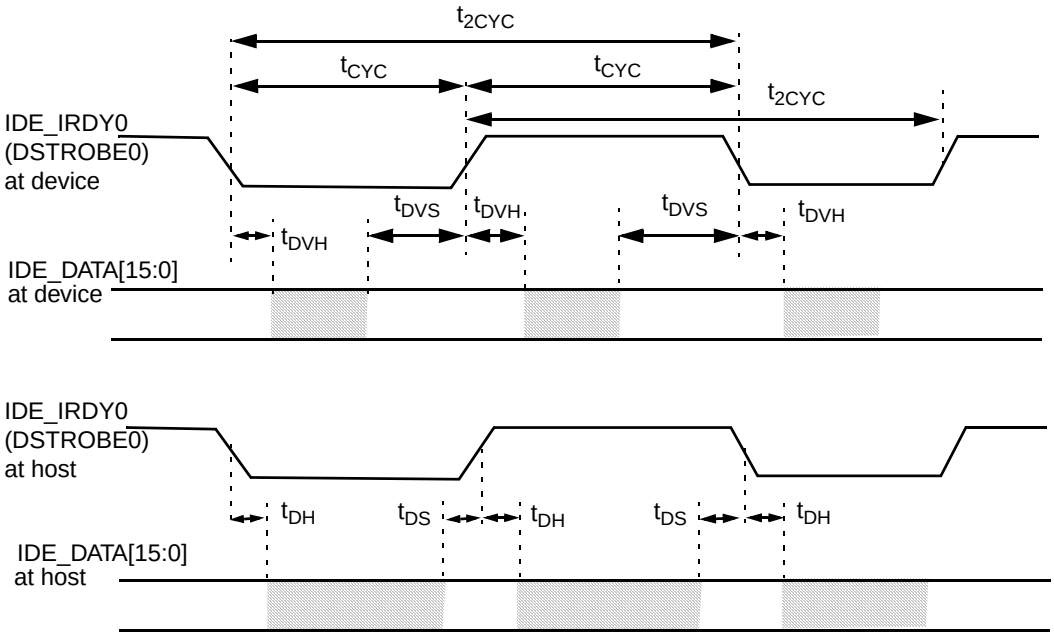
All timing parameters are measured at the connector of the device to which the parameter applies. For example, the sender stops generating STROBE edges t_{RFS} after the negation of DMARDY. Both STROBE and DMARDY timing measurements are taken at the connector of the sender.



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]), IDE_IOR[0:1]# (HDMARDY[0:1]#) and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are not in effect until IDE_REQ[0:1] and IDE_DACK[0:1]# are asserted.

Figure 7-25. Initiating an UltraDMA Data in Burst

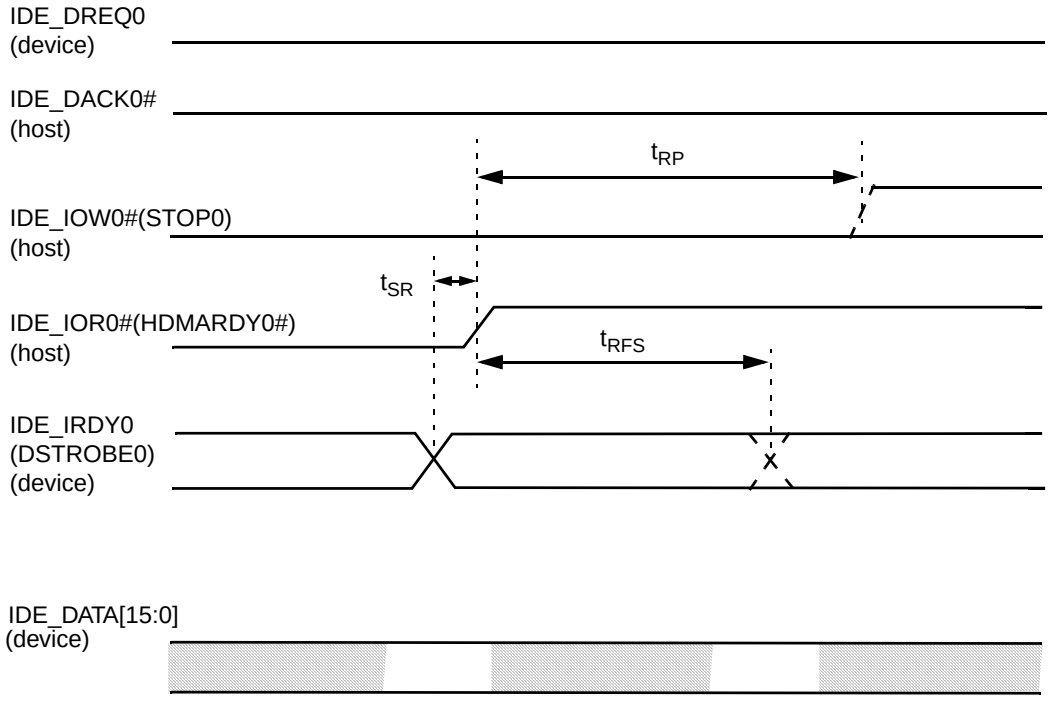
Electrical Specifications (Continued)



Note: IDE_DATA[15:0] and IDE_IRDY[0:1] (DSTROBE[0:1]) signals are shown at both the host and the device to emphasize that cable settling time and cable propagation delay do not allow the data signals to be considered stable at the host until a certain amount of time after they are driven by the device.

Figure 7-26. Sustained UltraDMA Data In Burst

Electrical Specifications (Continued)



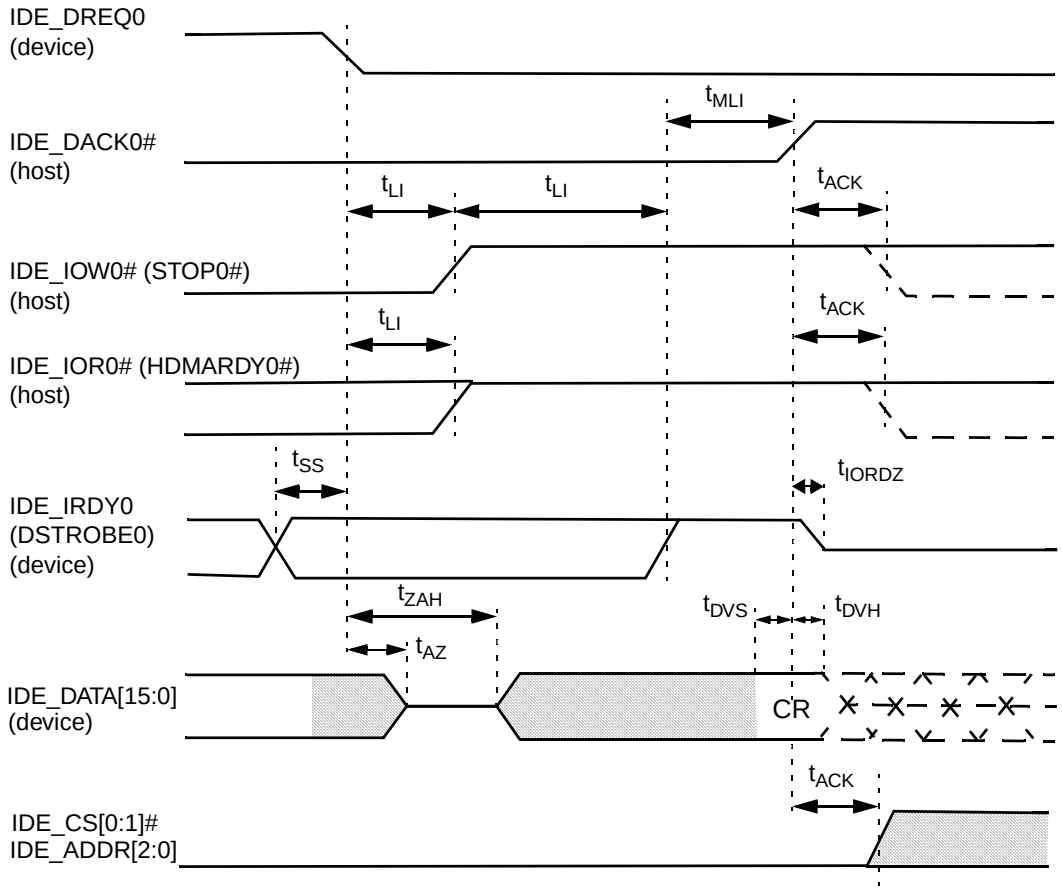
Notes:

1) The host can assert IDE_IOW[0:1]# (STOP[0:1]#) to request termination of the UltraDMA burst no sooner than t_{RP} after IDE_IOR[0:1]# (HDMARDY[0:1]#) is deasserted.

2) If the t_{SR} timing is not satisfied, the host may receive up to two additional data WORDs from the device.

Figure 7-27. Host Pausing an UltraDMA Data In Burst

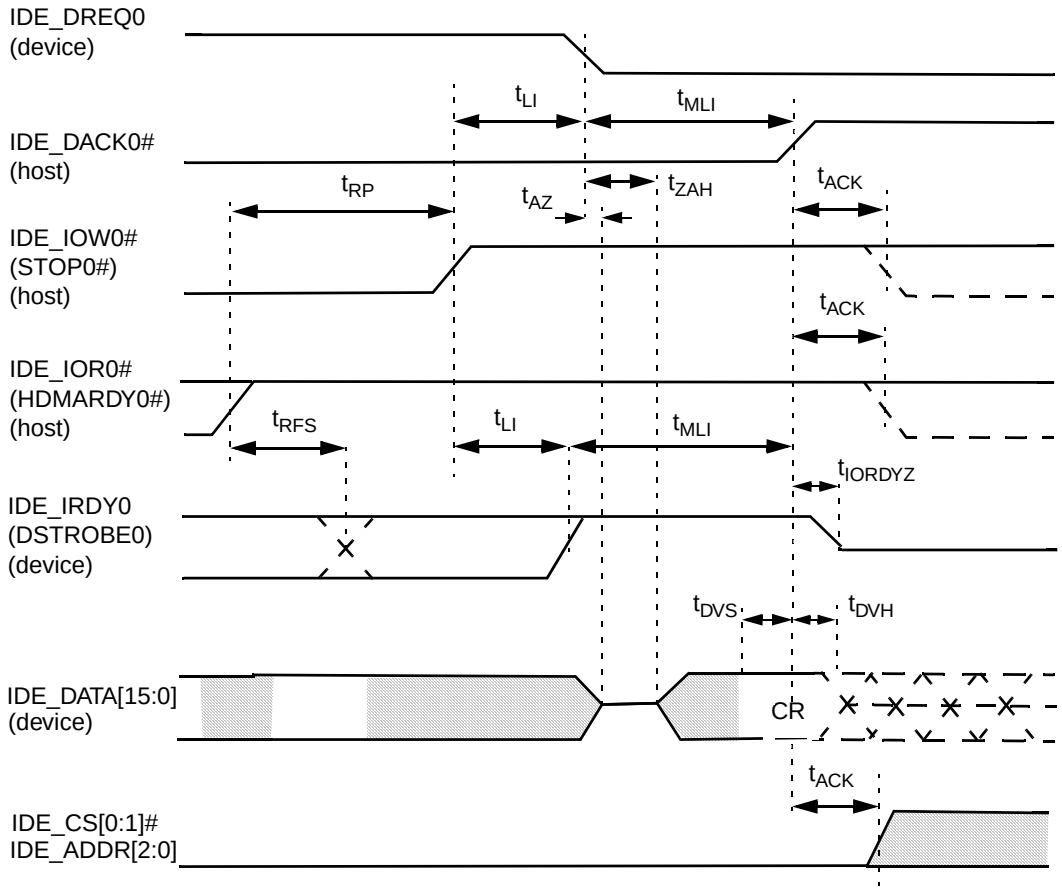
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IOR[0:1]# (HDMARDY[0:1]#), and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are deasserted.

Figure 7-28. Device Terminating an UltraDMA Data In Burst

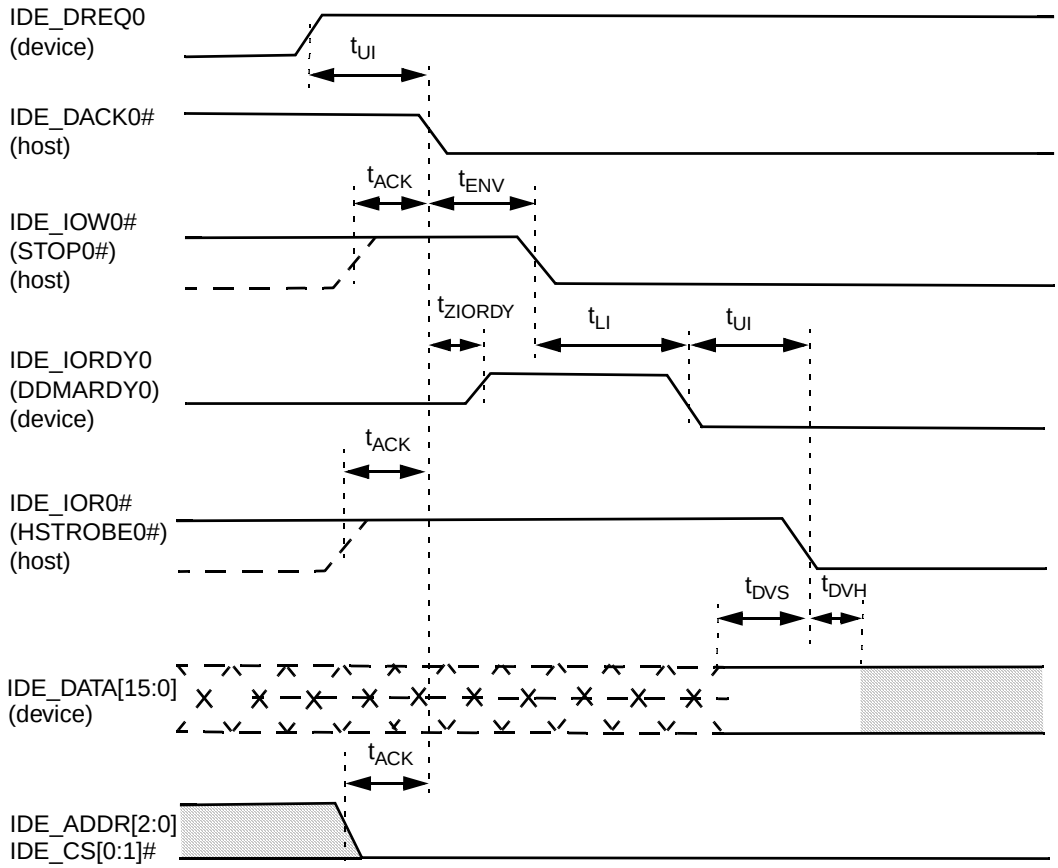
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IOR[0:1]# (HDMARDY[0:1]#), and IDE_IRDY[0:1] (DSTROBE[0:1]) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1] are deasserted.

Figure 7-29. Host Terminating an UltraDMA Data In Burst

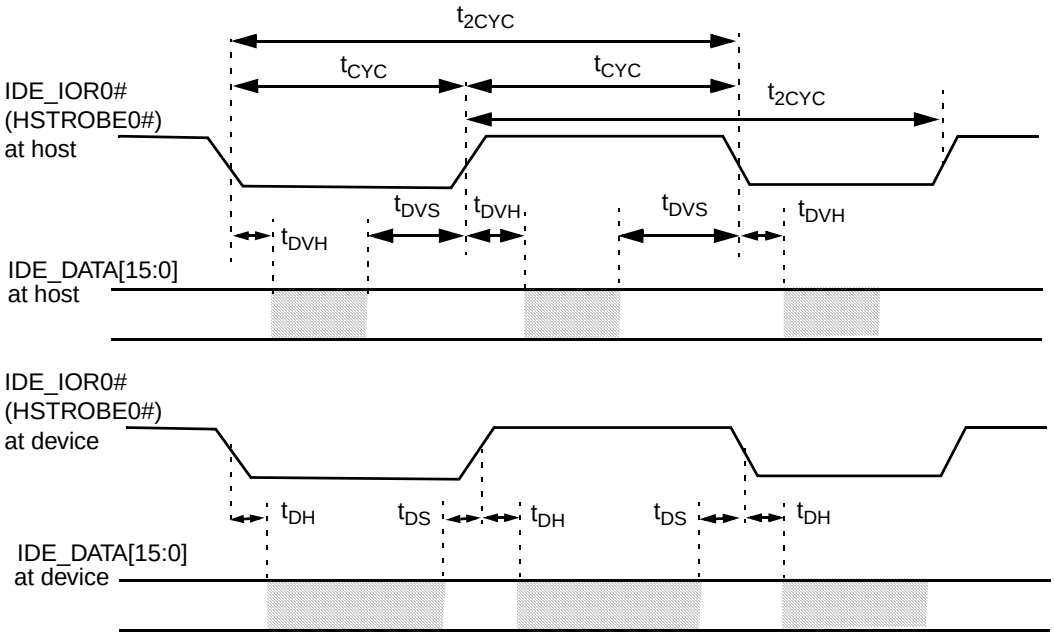
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]]# (STOP[0:1]#), IDE_IORDY[0:1]# (DDMARDY[0:1]) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are not in effect until IDE_DREQ[0:1] and IDE_DACK[0:1]# are asserted.

Figure 7-30. Initiating an UltraDMA Data Out Burst

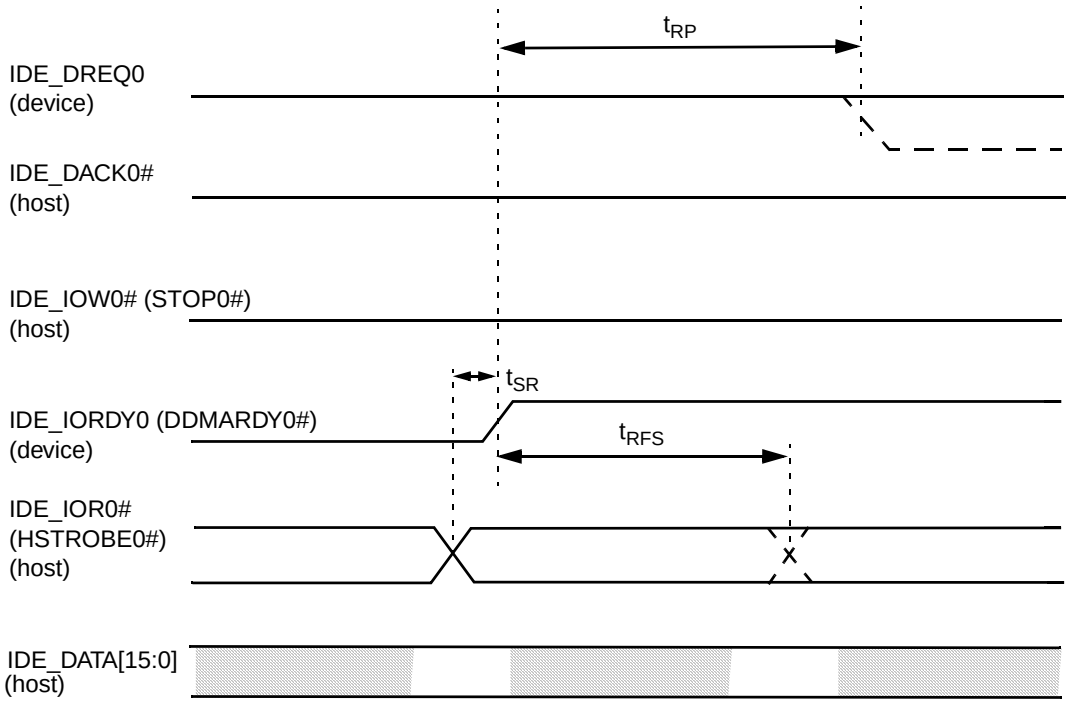
Electrical Specifications (Continued)



Note: IDE_DATA[15:0] and IDE_IOR[0:1]# (HSTROBE[0:1]#) signals are shown at both the device and the host to emphasize that cable settling time and cable propagation delay do not allow the data signals to be considered stable at the device until a certain amount of time after they are driven by the device.

Figure 7-31. Sustained UltraDMA Data Out Burst

Electrical Specifications (Continued)



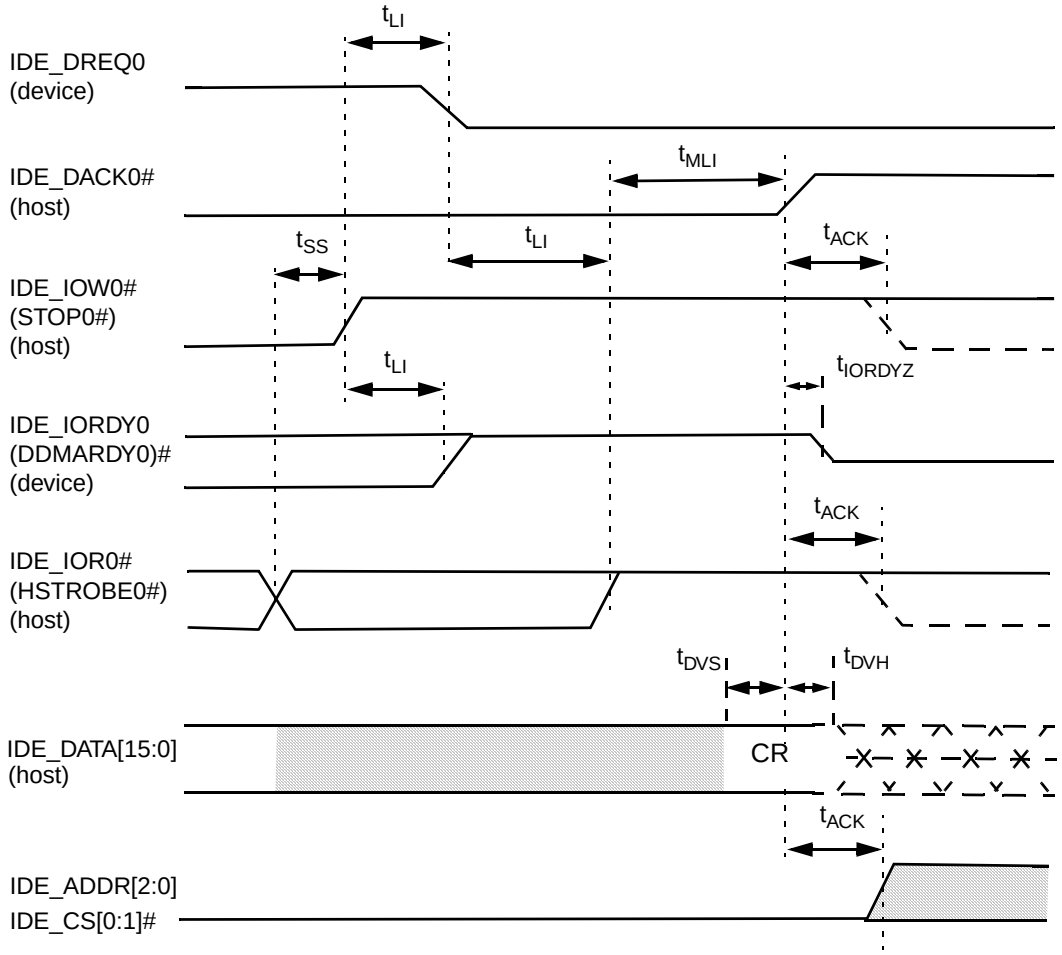
Notes:

1) The device can deassert IDE_DREQ[0:1] to request termination of the UltraDMA burst no sooner than t_{RP} after IDE_IORDY[0:1]# (DDMARDY[0:1]#) is deasserted.

2) If the t_{SR} timing is not satisfied, the device may receive up to two additional datawords from the host.

Figure 7-32. Device Pausing an UltraDMA Data Out Burst

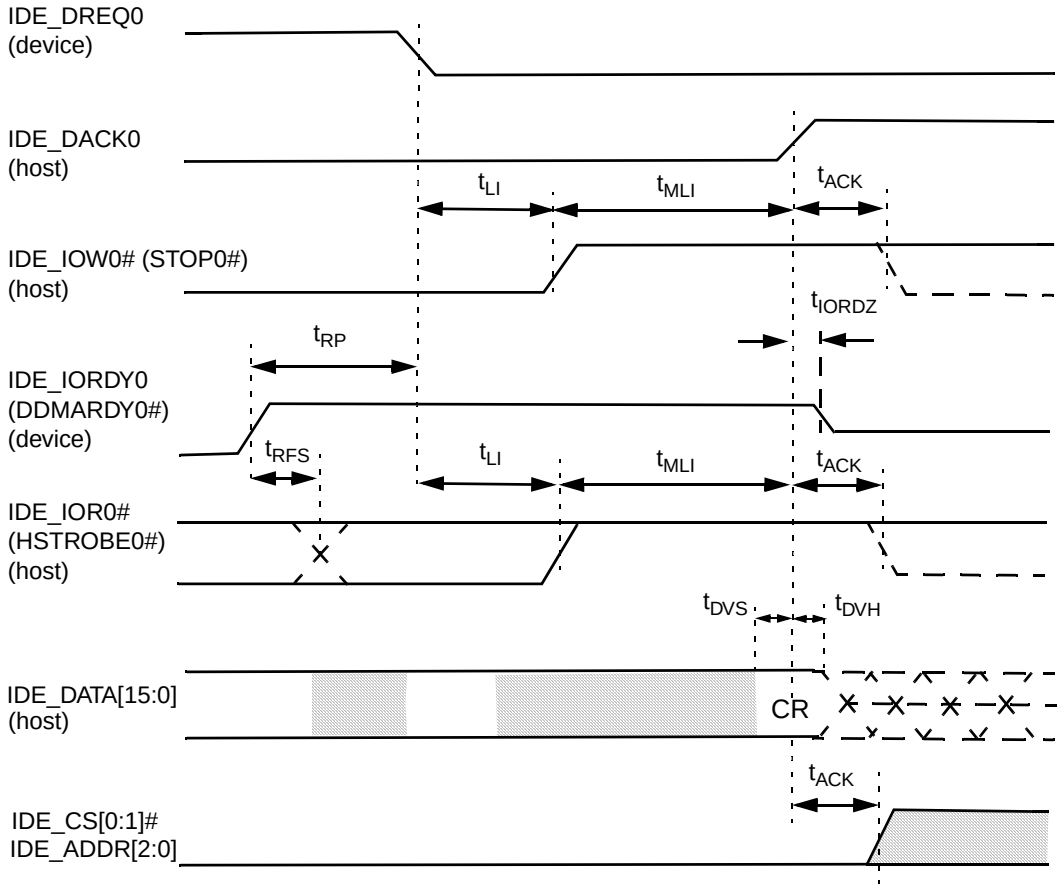
Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IORDY[0,1]# (DDMARDY[0:1]#) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are deasserted.

Figure 7-33. Host Terminating an UltraDMA Data Out Burst

Electrical Specifications (Continued)



Note: The definitions for the IDE_IOW[0:1]# (STOP[0:1]#), IDE_IORDY[0:1]# (DDMARDY[0:1]#) and IDE_IOR[0:1]# (HSTROBE[0:1]#) signal lines are no longer in effect after IDE_DREQ[0:1] and IDE_DACK[0:1]# are deasserted.

Figure 7-34. Device Terminating an UltraDMA Data Out Burst

Electrical Specifications (Continued)

7.3.7 Universal Serial Bus (USB)

Table 7-23. USB Interface Signals

Symbol	Parameter	Min	Max	Unit	Figure	Comments
Full Speed Source[1, 2]						
t_{USB_R1}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Rise Time	4	20	ns	7-35	(Monotonic) from 10% to 90% of the D_Port lines
t_{USB_F1}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Fall Time	4	20	ns	7-35	(Monotonic) from 90% to 10% of the D_Port lines
t_{USB_FRFM}	Rise/Fall time matching	90	110	%		
t_{USB_FSDR}	Full-speed data rate	11.97	12.03	Mbps		Average bit rate 12 Mbps ± 0.25%
t_{USB_FSF}	Full-speed frame interval	0.9995	1.0005	ms		1.0 ms ± 0.05%
t_{period_F}	Full-speed period between data bits	83.1	83.5	ns		Average bit rate 12 Mbps
$t_{USB\ DOR}$	Driver-output resistance	28	43	W		Steady-state drive
t_{USB_DJ11}	Source differential driver jitter[3, 4] for consecutive transition	–3.5	3.5	ns	7-36	
t_{USB_DJ12}	Source differential driver jitter[3, 4] for paired transitions	–4.0	4.0	ns	7-36	
t_{USB_SE1}	Source EOP width[4, 5]	160	175	ns	7-36	
t_{USB_DE1}	Differential to EOP transition skew[4, 5]	–2	5	ns	7-37	
t_{USB_RJ11}	Receiver data jitter tolerance[4] for consecutive transition	–18.5	18.5	ns	7-38	
t_{USB_RJ12}	Receiver data jitter tolerance[4] for paired transitions	–9	9	ns	7-38	
Full Speed Receiver EOP Width[4]						
t_{USB_RE11}	Must reject as EOP[5]		40	ns	7-37	
t_{USB_RE12}	Must accept as EOP[5]	82		ns	7-37	
Low Speed Source[1, 6]						
t_{USB_R2}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Rise Time	75	300[6]	ns	7-35	(Monotonic) from 10% to 90% of the D_Port lines
t_{USB_F2}	DPOS_Port1,2,3, DNEG_Port1,2,3 Driver Fall Time	75	300[6]	ns	7-35	(Monotonic) from 90% to 10% of the D_Port lines
t_{USB_LRFM}	Low-speed Rise/Fall time matching	80	120	%		
t_{USB_LSDR}	Low-speed data rate	1.4775	1.5225	Mbps		Average bit rate 1.5 Mbps ± 1.5%
t_{PERIOD_L}	Low-speed period	0.657	0.677	µs		at 1.5 Mbps
t_{USB_DJD21}	Source differential driver jitter[4] for consecutive transactions	–75	75	ns		Host (downstream)
t_{USB_DJD22}	Source differential driver jitter[4] for paired transactions	–45	45	ns	7-36	Host (downstream)
t_{USB_DJU21}	Source differential driver jitter[4] for consecutive transaction	–95	95	ns	7-36	Function (downstream)

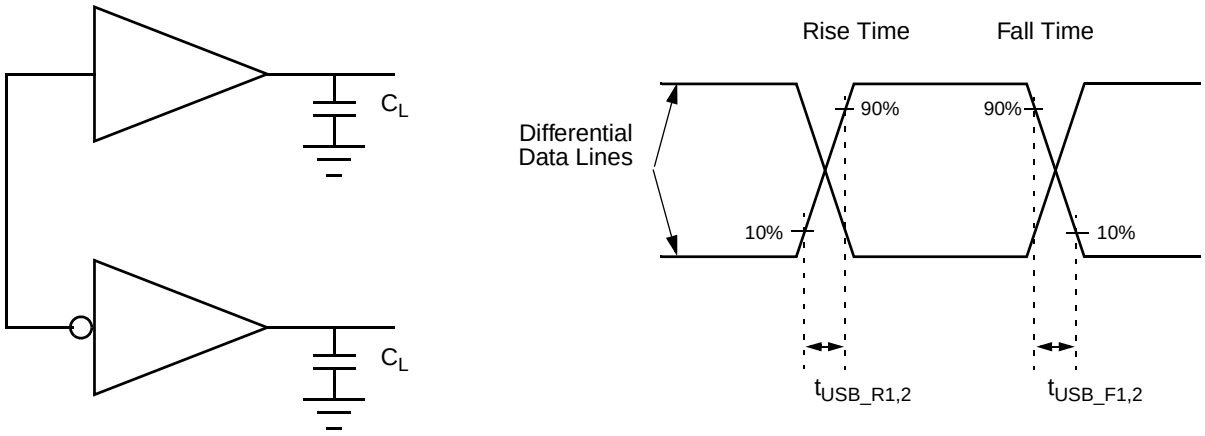
Electrical Specifications (Continued)

Table 7-23. USB Interface Signals (Continued)

Symbol	Parameter	Min	Max	Unit	Figure	Comments
t_{USB_DJU22}	Source differential driver jitter[4] for paired transactions	–150	150	ns	7-36	Function (downstream)
t_{USB_SE2}	Source EOP width[4, 5]	1.25	1.5	µs	7-37	
t_{USB_DE2}	Differential to EOP [5] transition skew	–40	100	ns	7-37	
t_{USB_RJD21}	Receiver Data Jitter Tolerance[4] for consecutive transactions	–152	152	ns	7-38	Host (upstream)
t_{USB_RJD22}	Receiver Data Jitter Tolerance[4] for paired transactions	–200	200	ns	7-38	Host (upstream)
t_{USB_RJU21}	Receiver Data Jitter Tolerance[4] for consecutive transactions	–75	75	ns	7-38	Function (downstream)
t_{USB_RJU22}	Receiver Data Jitter Tolerance[4] for paired transactions	–45	45	ns	7-38	Function (downstream)
Low Speed Receiver EOP Width[5]						
t_{USB_RE21}	Must reject as EOP		330	ns	7-36	
t_{USB_RE22}	Must accept as EOP	675		ns	7-36	

1. Unless otherwise specified, all timings use a 50 pF capacitive load (C_L) to ground.
2. Full-speed timing has a 1.5 KΩ pull-up to 2.8 V on the DPOS_Port1,2,3 lines.
3. Timing difference between the differential data signals (DPOS_PORT1,2,3 and DNEG_PORT1,2,3).
4. Measured at the crossover point of differential data signals (DPOS_PORT1,2,3 and DNEG_PORT1,2,3).
5. EOP is the End of Packet where DPOS_PORT^t = DNEG_PORT = SE0. SE0 occurs when output level voltage ≤ V_{SE} (Min).
6. C_L = 350 pF.

Electrical Specifications (Continued)



Full Speed: 4 to 20 ns at C_L = 50 pF
Low Speed: 75 ns at C_L = 50 pF, 300 ns at C_L = 350 pF

Figure 7-35. Data Signal Rise and Fall Time

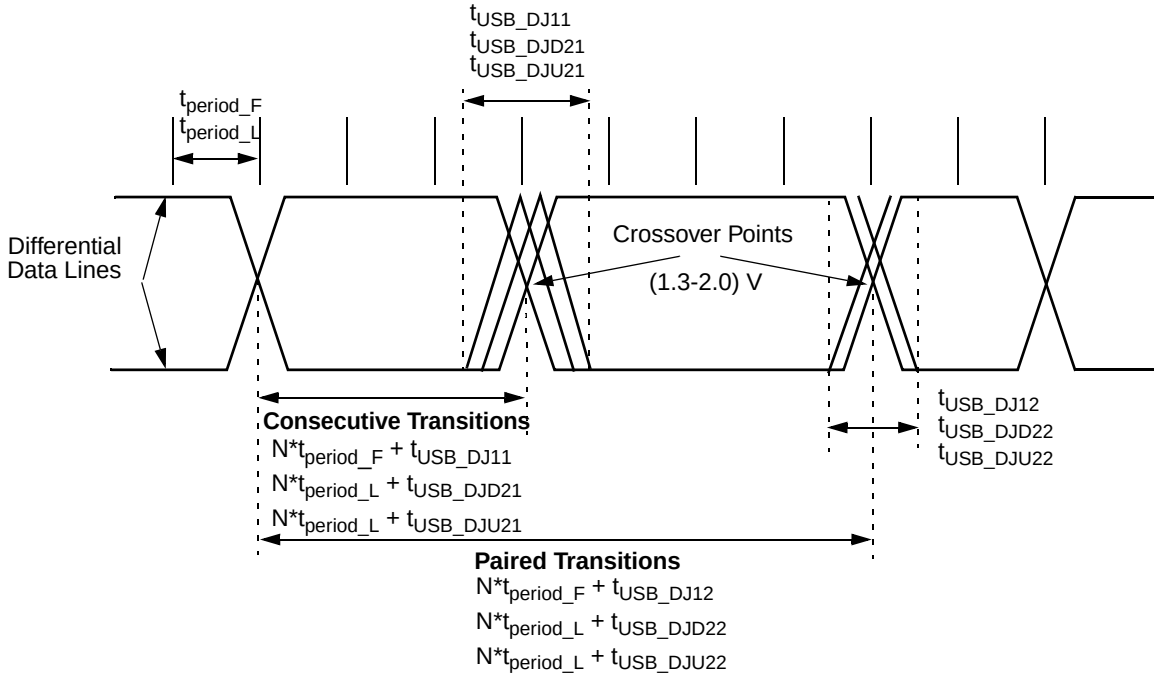


Figure 7-36. Source Differential Data Jitter

Electrical Specifications (Continued)

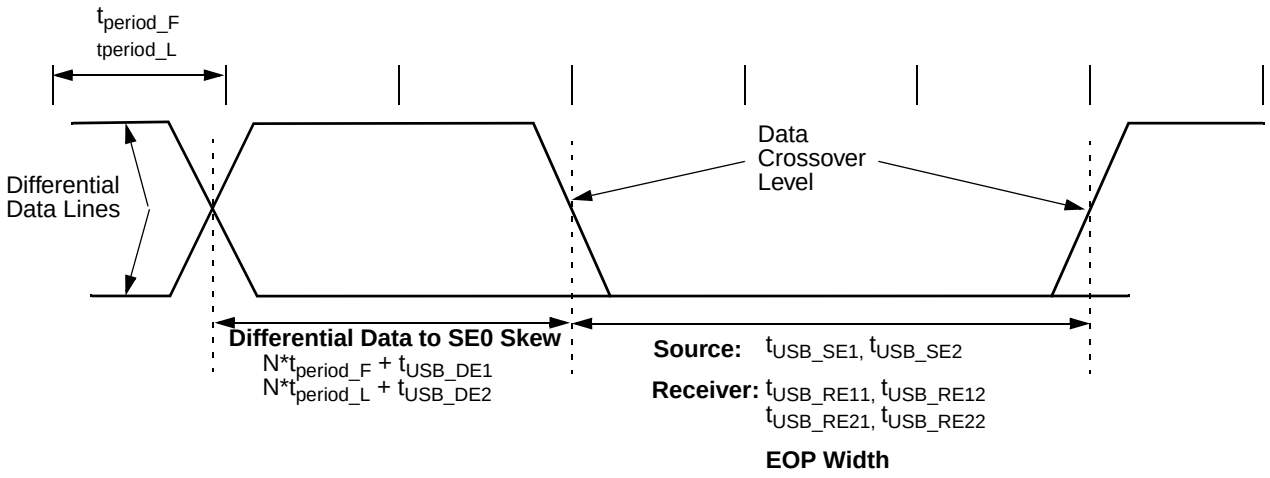


Figure 7-37. EOP Width Timing

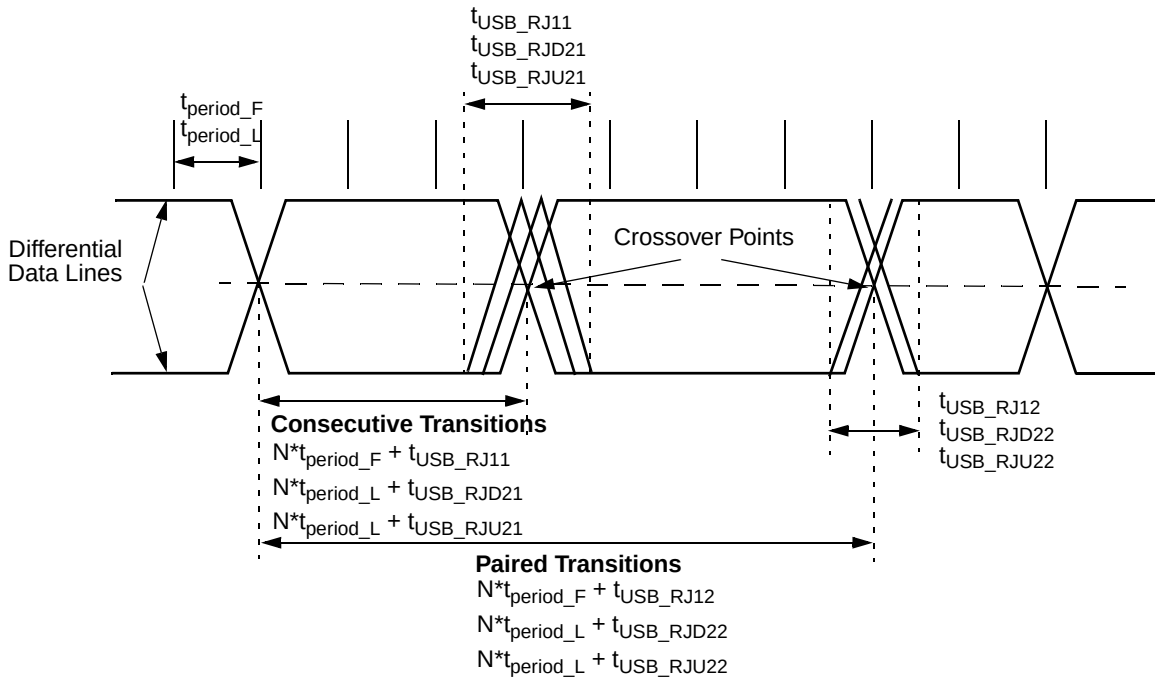


Figure 7-38. Receiver Jitter Tolerance

Electrical Specifications (Continued)

7.3.8 Serial Port (UART)

Table 7-24. UART, Sharp-IR, SIR, and Consumer Remote Control Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{BT}	Single Bit Time in UART and Sharp-IR	t_{BTN} - 25 [1]	t_{BTN} + 25	ns	Transmitter
		t_{BTN} - 2%	t_{BTN} + 2%	ns	Receiver
t_{CMW}	Modulation Signal Pulse Width in Sharp-IR and Consumer Remote Control	t_{CWN} - 25 [2]	t_{CWN} + 25	ns	Transmitter
		500		ns	Receiver
t_{CMP}	Modulation Signal Period in Sharp-IR and Consumer Remote Control	t_{CPN} - 25 [3]	t_{CPN} + 25	ns	Transmitter
		t_{MMIN} [4]	t_{MMAX} [4]	ns	Receiver
t_{SPW}	SIR Signal Pulse Width	$(^3/_{16})$ x t_{BTN} - 15 [1]	$(^3/_{16})$ x t_{BTN} + 15 [1]	ns	Transmitter, Variable
		1.48	1.78	µs	Transmitter, Fixed
		1		µs	Receiver
S_{DRT}	SIR Data Rate Tolerance % of Nominal Data Rate		± 0.87%		Transmitter
			± 2.0%		Receiver
t_{SJT}	SIR Leading Edge Jitter % of Nominal Bit Duration		± 2.5%		Transmitter
			± 6.5%		Receiver

1. t_{BTN} is the nominal bit time in UART, Sharp-IR, SIR and Consumer Remote Control modes. It is determined by the setting of the Baud Generator Divisor registers.
2. t_{CWN} is the nominal pulse width of the modulation signal for Sharp-IR and Consumer Remote Control modes. It is determined by the MCPW field (bits [7:5]) of the IRTXMC register and the TXHSC bit (bit 2) of the RCCFG register.
3. t_{CPN} is the nominal period of the modulation signal for Sharp-IR and Consumer Remote Control modes. It is determined by the MCFR field (bits [4:0]) of the IRTXMC register and the TXHSC bit (bit 2) of the RCCFG register.
4. t_{MMIN} and t_{MMAX} define the time range within which the period of the incoming subcarrier signal has to fall in order for the signal to be accepted by the receiver. These time values are determined by the contents of register IRRXDC and the setting of the RXHSC bit (bit 5) of the RCCFG register.

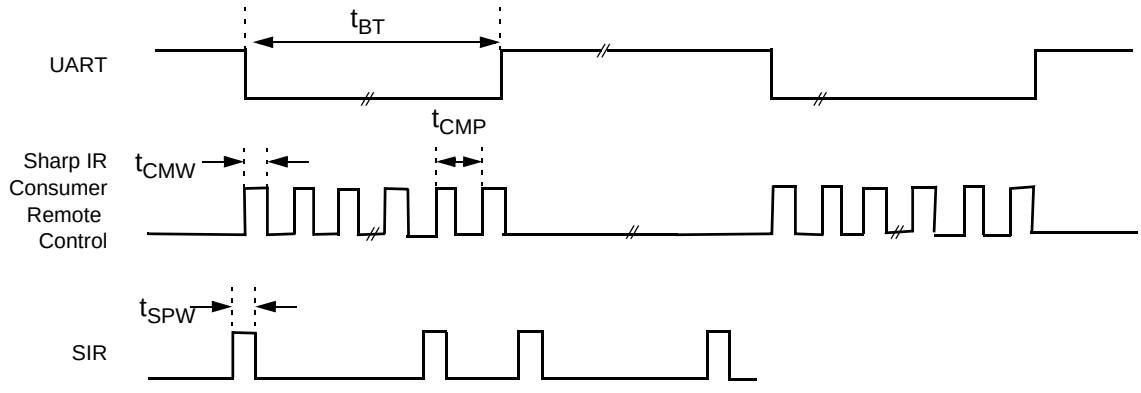


Figure 7-39. UART, Sharp-IR, SIR, and Consumer Remote Control Timing

Electrical Specifications (Continued)

7.3.9 Fast IR Port Timing

Table 7-25. Fast IR Port Timing Parameters

Symbol	Parameter	Min	Max	Unit	Comments
t_{MPW}	MIR Signal Pulse Width	t_{MWN}-25[1]	t_{MWN}+25	ns	Transmitter
		60		ns	Receiver
M_{DRT}	MIR Transmitter Data Rate Tolerance		± 0.1%		
t_{MJT}	MIR Receiver Edge Jitter, % of Nominal Bit Duration		± 2.9%		
t_{FPW}	FIR Signal Pulse Width	120	130	ns	Transmitter
		90	160	ns	Receiver
t_{FDPW}	FIR Signal Double Pulse Width	245	255	ns	Transmitter
		215	285	ns	Receiver
F_{DRT}	FIR Transmitter Data Rate Tolerance		± 0.01%		
t_{FJT}	FIR Receiver Edge Jitter, % of Nominal Bit Duration		± 4.0%		

1. t_{MWN} is the nominal pulse width for MIR mode. It is determined by the M_PWID field (bits [4:0]) in the MIR_PW register at offset 01h in bank 6 of logical device 5.

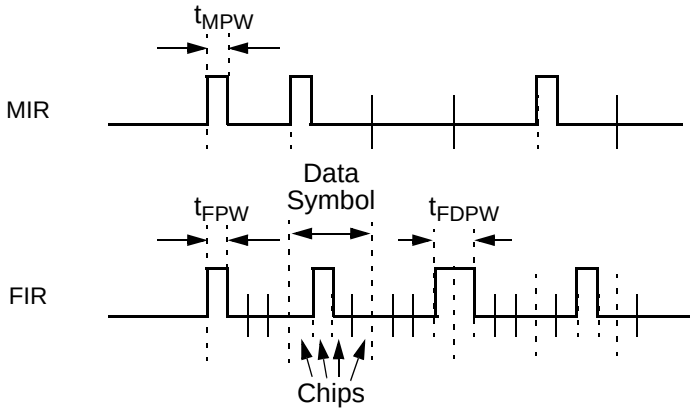


Figure 7-40. Fast IR Timing (MIR and FIR)

Electrical Specifications (Continued)

7.3.10 Audio Interface Timing (AC97)

Table 7-26. Cold Reset

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{RST_LOW}	AC97_RST# active low pulse width	1.0			µs	
$t_{RST2CLK}$	AC97_RST# inactive to BIT_CLK startup delay	162.8			ns	

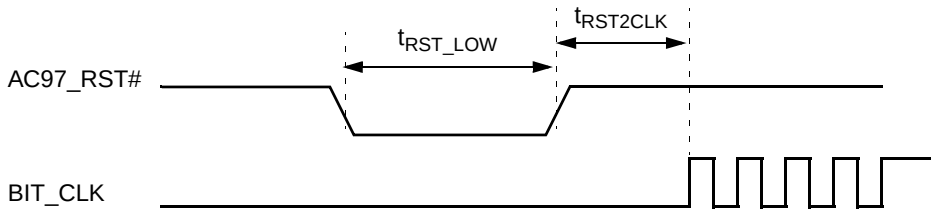


Figure 7-41. AC97 Reset Timing

Table 7-27. Warm Reset

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{SYNC_HIGH}	SYNC active high pulse width		1.3		µs	
t_{SYNC_IA}	SYNC inactive to BIT_CLK startup delay	162.8			ns	

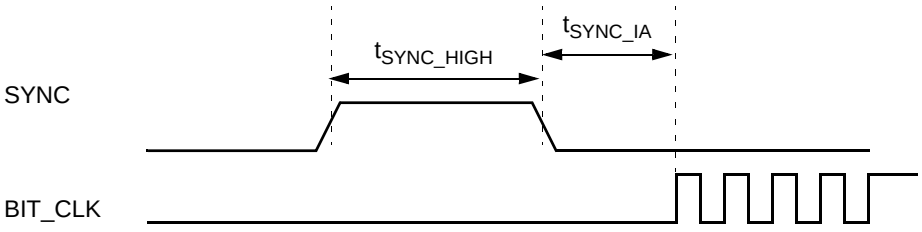


Figure 7-42. AC97 Sync Timing

Electrical Specifications (Continued)

Table 7-28. Clocks

Symbol	Parameter	Min	Typ	Max	Unit	Comments
F_{BIT_CLK}	BIT_CLK frequency		12.288		MHz	
t_{CLK_PD}	BIT_CLK period		81.4		ns	
t_{CLK_J}	BIT_CLK output jitter			750	ps	
t_{CLK_H}	BIT_CLK high pulse width[1]	32.56	40.7	48.84	ns	
t_{CLK_L}	BIT_CLK low pulse width[1]	32.56	40.7	48.84	ns	
F_{SYNC}	SYNC frequency		48.0		KHz	
t_{SYNC_PD}	SYNC period		20.8		µs	
t_{SYNC_H}	SYNC high pulse width		1.3		µs	
t_{SYNC_L}	SYNC low pulse width		19.5		µs	
F_{AC97_CLK}	AC97_CLK Frequency		24.576		MHz	
$t_{AC97_CLK_PD}$	AC97_CLK Period		40.7		ns	
$t_{AC97_CLK_D}$	AC97_CLK Duty Cycle	45		55	%	
$t_{AC97_CLK_FR}$	AC97_CLK Fall/Rise time	2		5	ns	
$t_{AC97_CLK_J}$	AC97_CLK output edge-to-edge jitter			100	ps	Measured from edge to edge

1. Worst case duty cycle restricted to 40/60.

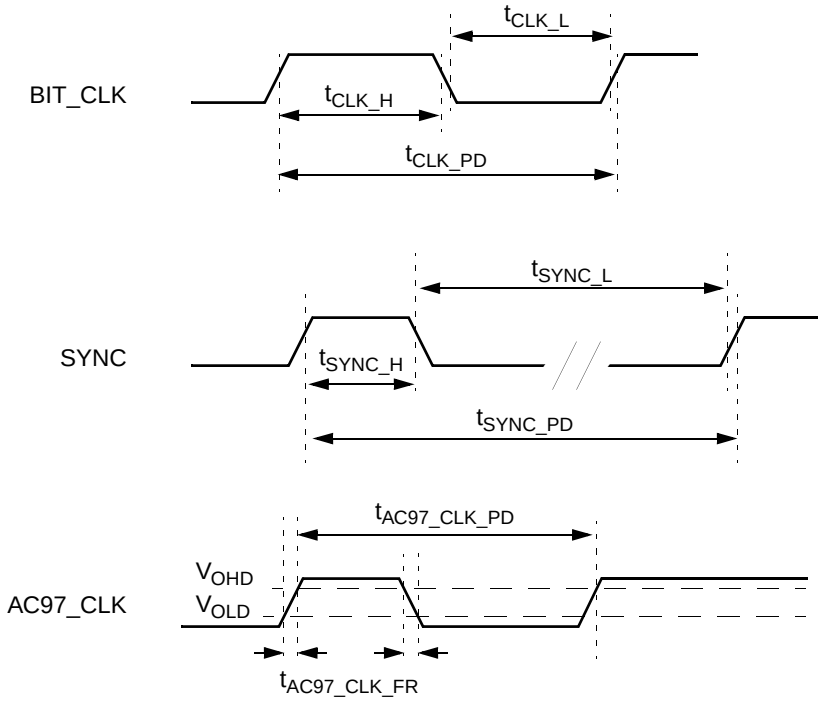


Figure 7-43. AC97 Clocks

Electrical Specifications (Continued)

Table 7-29. I/O Timing

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{AC97_S}	Input setup to falling edge of BIT_CLK	15.0			ns	
t_{AC97_H}	Hold from falling edge of BIT_CLK	10.0			ns	
t_{AC97_OV}	SDATA_OUT or SYNC valid after rising edge of BIT_CLK			15	ns	
t_{AC97_OH}	SDATA_OUT or SYNC hold time after falling edge of BIT_CLK	5			ns	
t_{AC97_SV}	Sync out valid after rising edge of BIT_CLK			15	ns	
t_{AC97_SH}	Sync out hold after falling edge of BIT_CLK	5			ns	

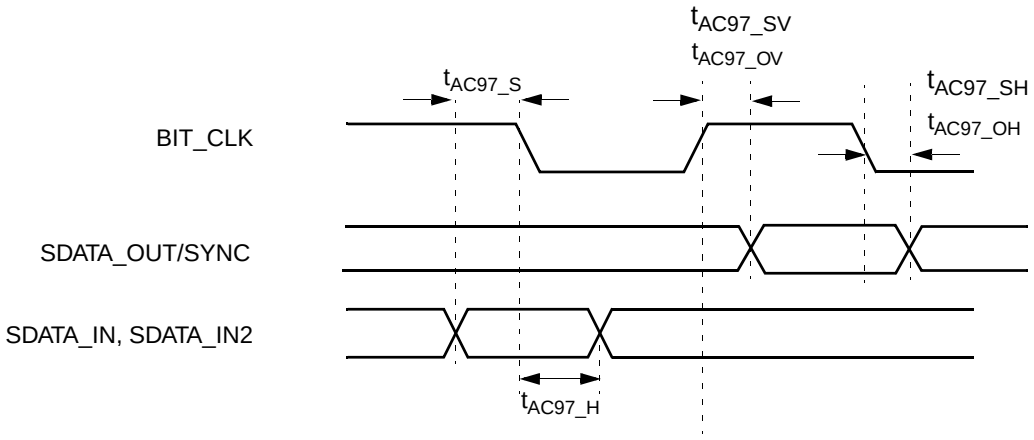


Figure 7-44. AC97 Data TIming

Electrical Specifications (Continued)

Table 7-30. Signal Rise and Fall Times

Symbol	Parameter	Min	Typ	Max	Unit	Comments
$trise_{CLK}$	BIT_CLK rise time	2		6	ns	
$tfall_{CLK}$	BIT_CLK fall time	2		6	ns	
$trise_{SYNC}$	SYNC rise time	2		6	ns	C_L = 50 pF
$tfall_{SYNC}$	SYNC fall time	2		6	ns	C_L = 50 pF
$trise_{DIN}$	SDATA_IN rise time	2		6	ns	
$tfall_{DIN}$	SDATA_IN fall time	2		6	ns	
$trise_{DOUT}$	SDATA_OUT rise time	2		6	ns	C_L = 50 pF
$tfall_{DOUT}$	SDATA_OUT fall time	2		6	ns	C_L = 50 pF

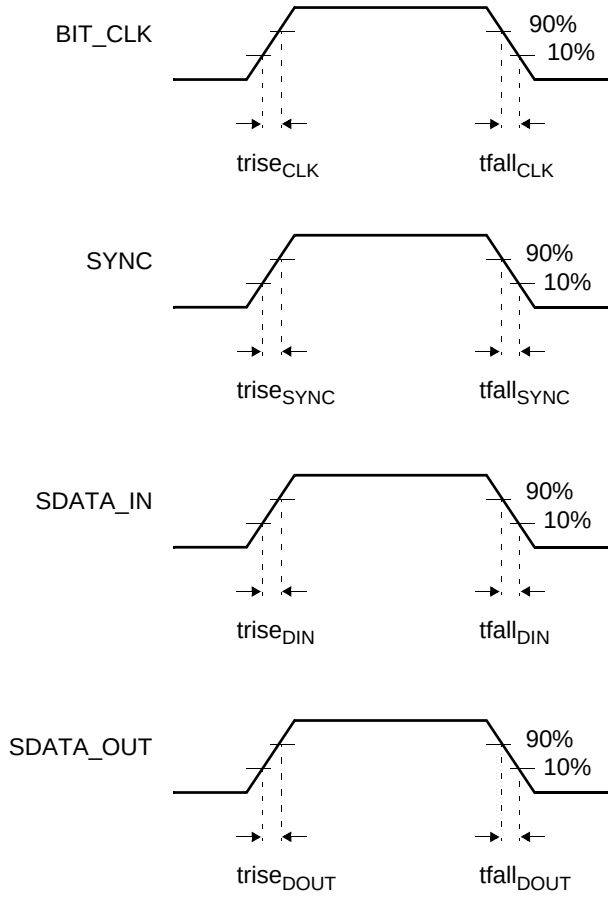
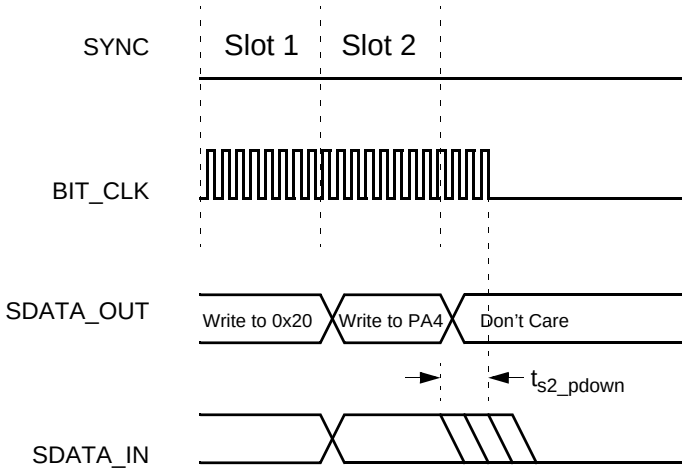


Figure 7-45. Rise and Fall Times

Electrical Specifications (Continued)

Table 7-31. AC-link Low Power Mode Timing

Symbol	Parameter	Min	Typ	Max	Unit	Comments
t_{s2_pdown}	End of Slot 2 to BIT_CLK, SDATA_IN low			1.0	µs	



Note: BIT_CLK is not to scale

Figure 7-46. Low Power Mode Timing

Electrical Specifications (Continued)

7.3.11 Power Management

Table 7-32. PWRBTN# Timing Parameters

Symbol	Parameter	Min	Max	Unit	Conditions
t_{PBTNP}	PWRBTN# Pulse Width[1]	16		ms	
t_{PBTNE}	Delay from PWRBTN# Events to ONCTL#	14	16	ms	

1. Not 100% tested.

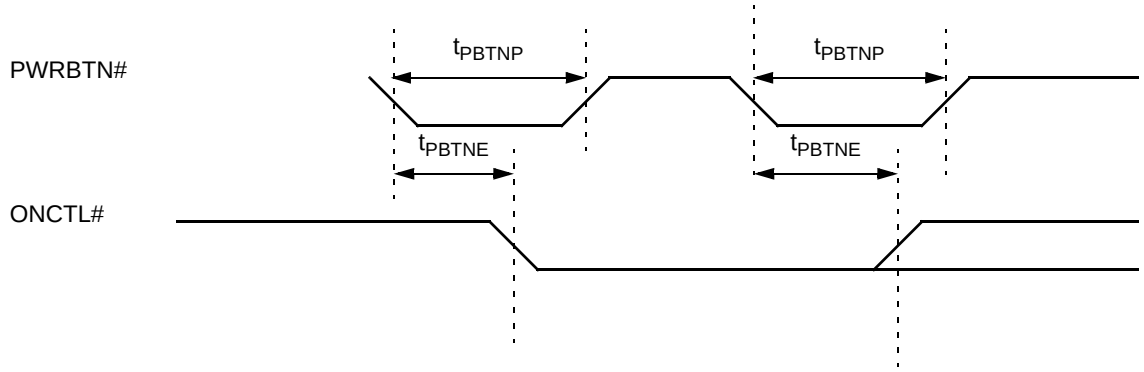


Figure 7-47. PWRBTN# Trigger and ONCTL# Timing

Table 7-33. Power Management Event (GPWIO) and ONCTL# Timing Parameters

Symbol	Parameter	Min	Max	Unit	Conditions
t_{PM}	Power Management Event to ONCTL# assertion		45	ns	

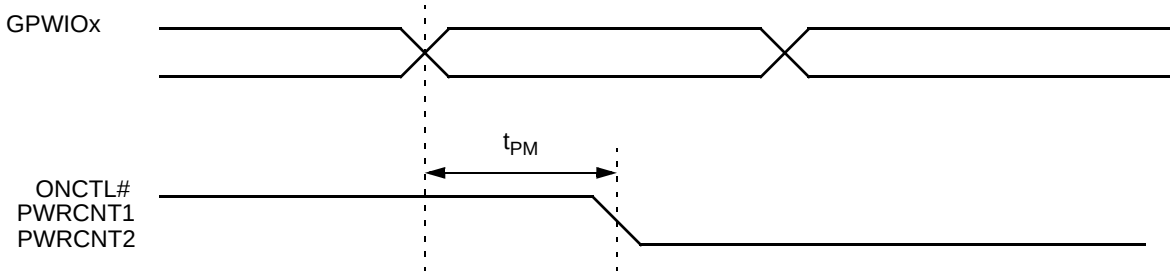


Figure 7-48. GPWIO and ONCTL# Timing

Electrical Specifications (Continued)

7.3.12 Power-Up Sequencing

Table 7-34. Power-Up Sequence Using the Power Button

Symbol	Parameter	Min	Max	Unit	Comments
t_1	Voltage sequence	-10	10	ms	
t_2	PWRBTN# inactive after V_{SB} or V_{SBL} applied, whichever is applied last	0	1	µs	PWRBTN# is an input and must be powered by V_{SB}.
t_3	PWRBTN# active pulse width	16	4000	ms	If PWRBTN# max is exceeded, ONCTL# will go inactive.
t_4	ONCTL# inactive after $V_{SB\ or}$ V_{SBL} applied, which ever is applied last	0	1	ms	
t_5	Signal active after PWRBTN active	16	32	ms	
t_6	V_{CORE} and V_{IO} applied after ONCTL# active	0		ms	System determines when V_{CORE} and V_{IO} are applied, hence there is no maximum constraint.
t_7	POR# inactive after V_{CORE} and V_{IO} applied	50		ms	POR# must not glitch during active time.

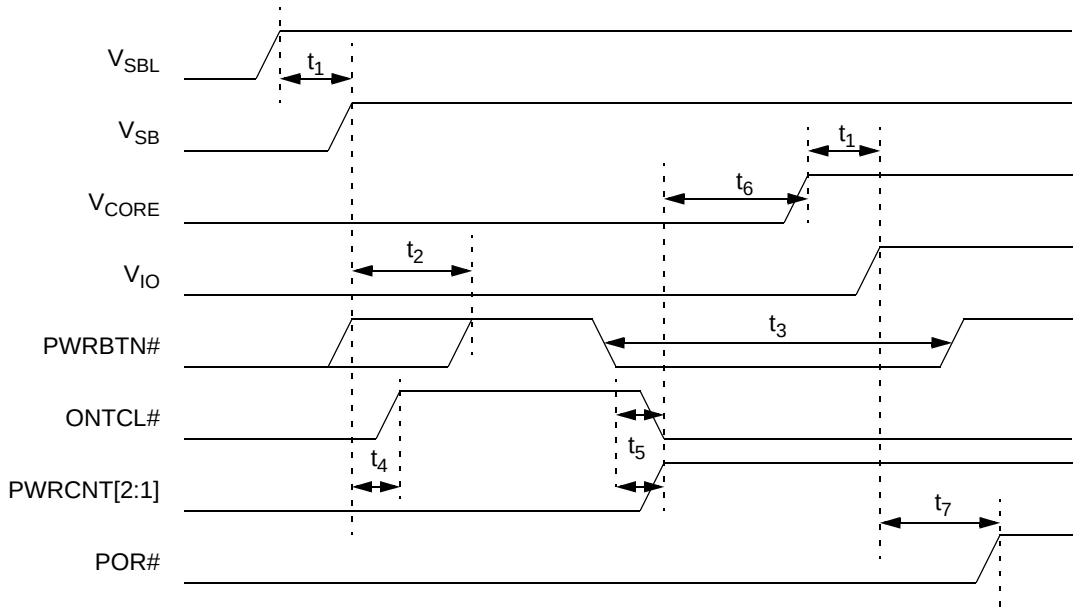


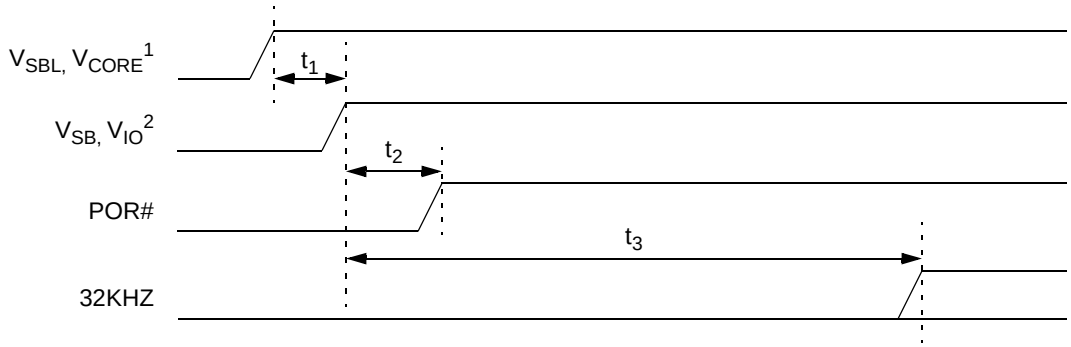
Figure 7-49. Power-Up Sequencing Timing

ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 196). GPIO63 must be pulsed low for at least 16 msec and not more than 4 sec.

Electrical Specifications (Continued)

Table 7-35. Power-Up Sequence Not Using the Power Button

Symbol	Parameter	Min	Max	Unit	Comments
t_1	Voltage sequence	-10	10	ms	
t_2	POR# inactive after V_{SBL}, V_{CORE}, V_{SB}, and V_{IO} applied	50		ms	POR# must not glitch during active time.
t_3	32KHZ startup time		1	s	Time required for 32 KHz oscillator to become stable at which time the RTC can reliably count.



1) V_{SBL} and V_{CORE} should be tied together.

2) V_{SB} and V_{IO} should be tied together.

Figure 7-50. Power-Up Sequencing Timing

ACPI is non-functional when the power-up sequence does not include using the power button. If ACPI functionality is desired, the power button must be toggled. This can be done externally or internally. GPIO63 is internally connected to PWRBTN#. To toggle the power button with software, GPIO63 must be programmed as an output using the normal GPIO programming protocol (see Section 5.4.1.1 "GPIO Support Registers" on page 196). GPIO63 must be pulsed low for at least 16 msec and not more than 4 sec. Asserting POR# has no effect on ACPI. If POR# is asserted and ACPI was active prior to POR#, then ACPI will remain active after POR#. Therefore, BIOS must ensure that ACPI is inactive before GPIO63 is pulsed low.

Electrical Specifications (Continued)

7.3.13 JTAG Timing

Table 7-36. JTAG Signals

Symbol	Parameter	Min	Max	Unit	Comments
	TCK Frequency (MHz)		25	MHz	
t_1	TCK Period	40		ns	
t_2	TCK High time	10		ns	
t_3	TCK Low time	10		ns	
t_4	TCK Rise time		4	ns	
t_5	TCK Fall Time		4	ns	
t_6	TDO Valid delay	3	25	ns	
t_7	Non-test outputs Valid delay	3	25	ns	50 pF load
t_8	TDO Float delay		30	ns	
t_9	Non-test outputs Float delay		36	ns	
t_{10}	TDI, TMS Setup time	8		ns	
t_{11}	Non-test inputs Setup time	8		ns	
t_{12}	TDI, TMS Hold time	7		ns	
t_{13}	Non-test inputs Hold time	7		ns	

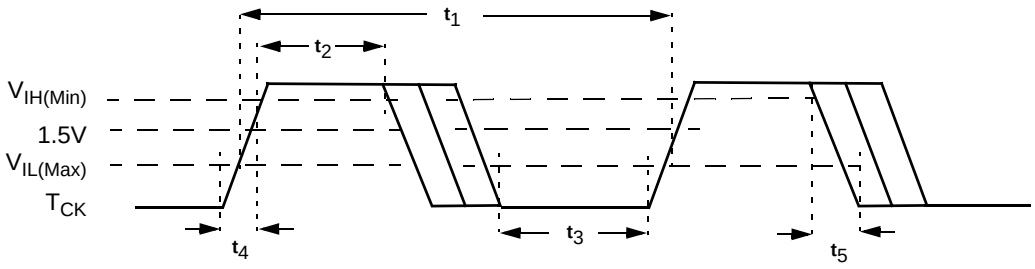


Figure 7-51. TCK Timing and Measurement Points

Electrical Specifications (Continued)

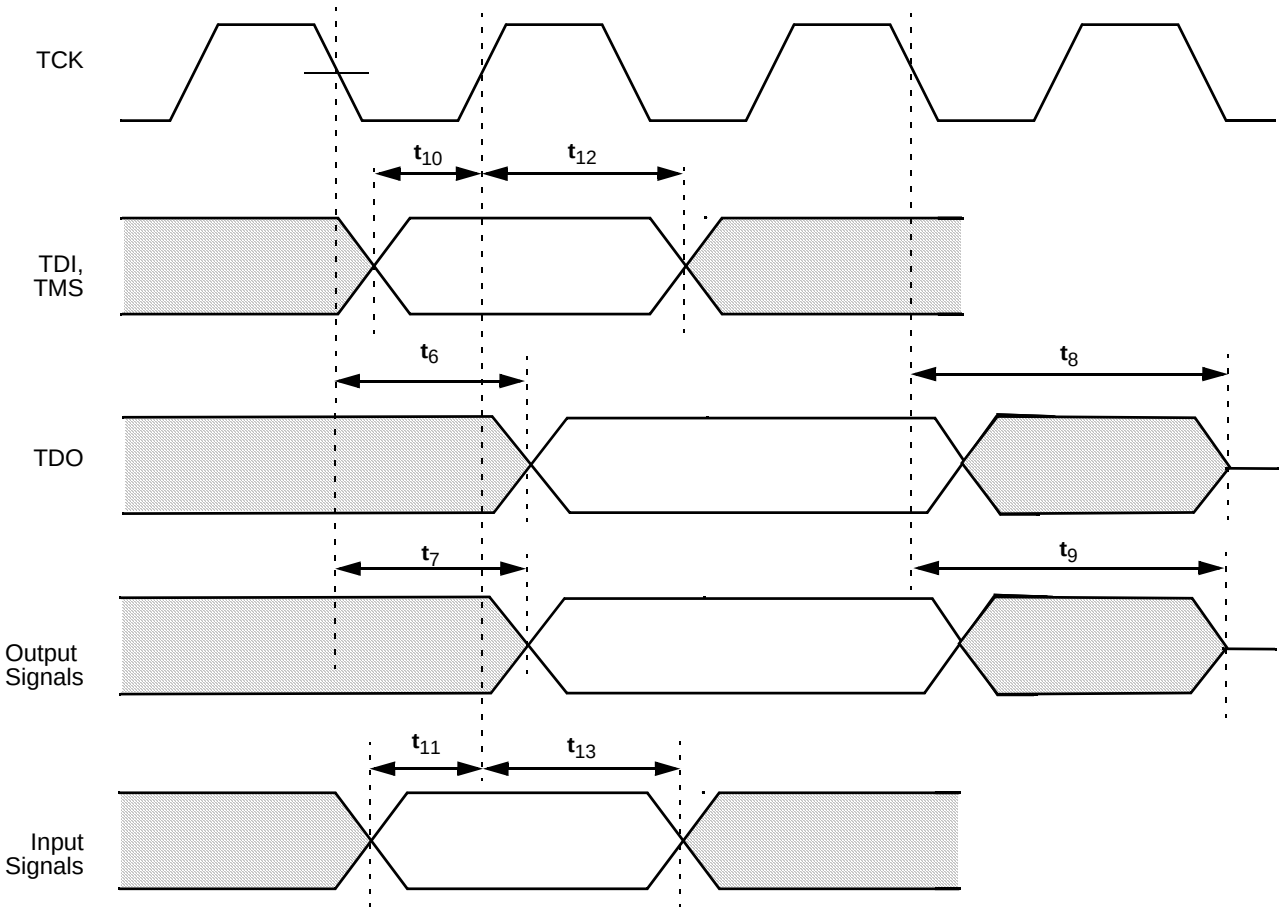


Figure 7-52. JTAG Test Timings

8.0 Package Specifications

8.1 THERMAL CHARACTERISTICS

The junction-to-case thermal resistance (θ_{JC}) of the TEPBGA package shown in Table 8-1 can be used to calculate the junction (die) temperature under any given circumstance.

Table 8-1. θ_{JC} (×C/W) for SC1100

Package	Max (°C/W)
TEPBGA without copper heat spreader	7
TEPBGA with copper heat spreader	5

Note that there is no specification for maximum junction temperature given since the operation of the TEPBGA device is guaranteed to a case temperature range of 0°C to 85°C (see Table 7-2 on page 281). As long as the case temperature of the device is maintained within this range, the junction temperature of the die will also be maintained within its allowable operating range. However, the die (junction) temperature under a given operating condition can be calculated by using the following equation:

$$T_J = T_C + (P * \theta_{JC})$$

where:

T_J = Junction temperature (°C)

T_C = Case temperature at top center of package (°C)

P = Maximum power dissipation (W)

θ_{JC} = Junction-to-case thermal resistance (°C/W)

These examples are given for reference only. The actual value used for maximum power (P) and ambient temperature (T_A) is determined by the system designer based on system configuration, extremes of the operating environment, and whether active thermal management (via Suspend Modulation) of the GX1 processor is employed.

A maximum junction temperature is not specified since a maximum case temperature is. Therefore, the following equation can be used to calculate the maximum thermal resistance required of the thermal solution for a given maximum ambient temperature:

$$\theta_{CS} + \theta_{SA} = \frac{T_C - T_A}{P}$$

where:

θ_{CS} = Max case-to-heatsink thermal resistance (°C/W) allowed for thermal solution

θ_{SA} = Max heatsink-to-ambient thermal resistance (°C/W) allowed for thermal solution

T_A = Max ambient temperature (°C)

T_C = Max case temperature at top center of package (°C)

P = Maximum power dissipation (W)

If thermal grease is used between the case and heatsink, θ_{CS} will reduce to about 0.01 °C/W. Therefore, the above equation can be simplified to:

$$\theta_{CA} = \frac{T_C - T_A}{P}$$

where:

$\theta_{CA} = \theta_{SA}$ = Max heatsink-to-ambient thermal resistance (°C/W) allowed for thermal solution

The calculated θ_{CA} value (examples shown in Table 8-2 on page 344) represents the maximum allowed thermal resistance of the selected cooling solution which is required to maintain the maximum T_C (shown in Table 7-2 on page 281) for the application in which the device is used.

Package Specifications (Continued)

Table 8-2. Case-to-Ambient Thermal Resistance Examples @ 85°C

Core Voltage (V_{CORE}) (Nominal)	Core Frequency	Maximum Power (W)	θ_{CA} for Different Ambient Temperatures (°C/W)				
			20°C	25°C	30°C	35°C	40°C
TBD	300 MHz	TBD	TBD	TBD	TBD	TBD	TBD
2.0V	266 MHz	2.98	21.84	20.16	18.48	16.8	15.12
1.8V	233 MHz	2.77	23.52	21.71	19.9	18.09	16.28

8.1.1 Heatsink Considerations

Table 8-2 shows the maximum allowed thermal resistance of a heatsink for particular operating environments. The calculated values, defined as θ_{CA}, represent the required ability of a particular heatsink to transfer heat generated by the SC1100 from its case into the air, thereby maintaining the case temperature at or below 85°C. Because θ_{CA} is a measure of thermal resistivity, it is inversely proportional to the heatsinks ability to dissipate heat or its thermal conductivity.

Note: A “perfect” heatsink would be able to maintain a case temperature equal to that of the ambient air inside the system chassis.

Looking at Table 8-2, it can be seen that as ambient temperature (T_A) increases, θ_{CA} decreases, and that as power consumption of the processor (P) increases, θ_{CA} decreases. Thus, the ability of the heatsink to dissipate thermal energy must increase as the processor power increases and as the temperature inside the enclosure increases.

While θ_{CA} is a useful parameter to calculate, heatsinks are not typically specified in terms of a single θ_{CA}.This is because the thermal resistivity of a heatsink is not constant across power or temperature. In fact, heatsinks become slightly less efficient as the amount of heat they are trying to dissipate increases. For this reason, heatsinks are typically specified by graphs that plot heat dissipation (in watts) vs. mounting surface (case) temperature rise above ambient (in °C). This method is necessary because ambient and case temperatures fluctuate constantly during normal operation of the system. The system designer must be careful to choose the proper heatsink by matching the required θ_{CA} with the thermal dissipation curve of the device under the entire range of operating conditions in order to make sure that the maximum case temperature (from Table 7-2 on page 281) is never exceeded. To choose the proper heatsink, the system designer must make sure that the calculated θ_{CA} falls above the curve (shaded area). The curve itself defines the minimum temperature rise above ambient that the heatsink can maintain.

Package Specifications (Continued)

Figure 8-1 is an example of a particular heatsink under consideration

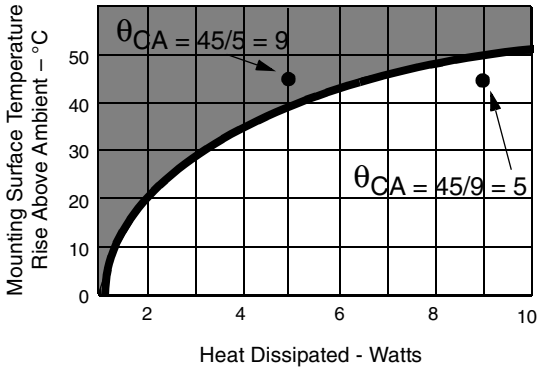


Figure 8-1. Heatsink Example

Example 1
Assume P (max) = 5W and T_A (max) = 40°C.

Therefore:

$$\theta_{CA} = \frac{T_C - T_A}{P}$$

$$\theta_{CA} = \frac{85 - 40}{5}$$

$$\theta_{CA} = 9$$

The heatsink must provide a thermal resistance below 9°C/W. In this case, the heatsink under consideration is more than adequate since at 5W worst case, it can limit the case temperature rise above ambient to 40°C (θ_{CA} =8).

Example 2
Assume P (max) = 9W and T_A (max) = 40°C.

Therefore:

$$\theta_{CA} = \frac{T_C - T_A}{P}$$

$$\theta_{CA} = \frac{85 - 40}{9}$$

$$\theta_{CA} = 5$$

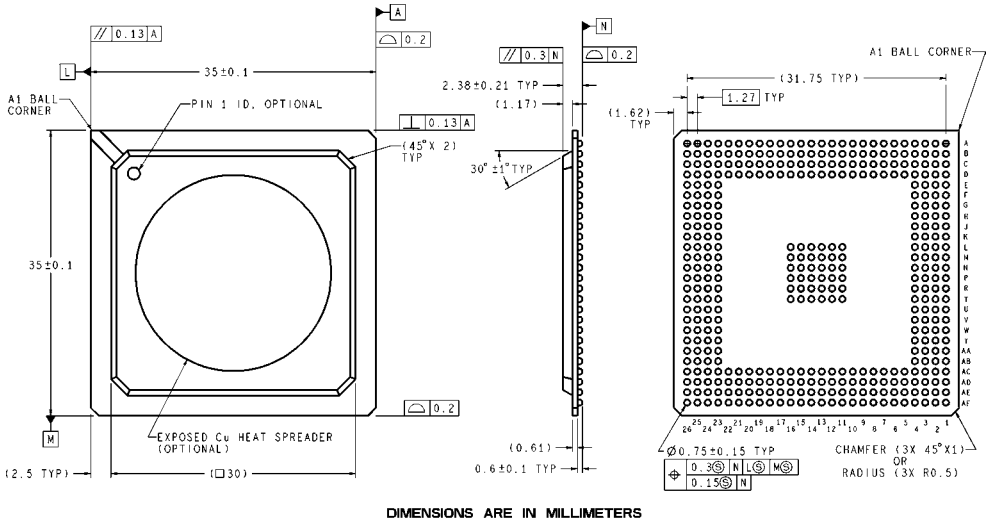
In this case, the heatsink under consideration is NOT adequate to limit the case temperature rise above ambient to 45°C for a 9W processor.

For more information on thermal design considerations or heatsink properties, refer to the Product Selection Guide of any leading vendor of thermal engineering solutions.

Note: The power dissipations P used in these examples are not representative of the power dissipation of the SC1100, which is always less than 4 Watts.

Package Specifications (Continued)

8.2 PHYSICAL DIMENSIONS



Notes:

1) Solder ball composition: Sn 63%, Pb 37%
2) Dimension is measured at the minimum solder ball diameter, parallel to primary datum N
3) The mold surface area may include dimple for A1 ball corner identification and mold ejector pin marks at each corner of molded package surface
4) Reference JEDEC registration MS-034, Variation BAR-2
5) The copper heat spreader is optional

Figure 8-2. 388-Terminal TEPBGA

Appendix A Support Documentation

A.1 ORDER INFORMATION

Order Number (NSID)	Part Marking	Core Frequency (MHz)	Core Voltage (V_{CORE})	Temperature (Degree C)	Package
SC1100UFH-233	SC1100UFH-233	233	1.8V	85	TEPBGA
SC1100UFH-266	SC1100UFH-266	266	2.0V	85	TEPBGA
SC1100UFH-300	SC1100UFH-300	300	TBD	85	TEPBGA

A.2 DATASHEET REVISION HISTORY

This document is a report of the revision/creation process of the datasheet for the Geode SC1100. Any revisions (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table below.

Table A-1. Revision History

Revision # (PDF Date)	Revisions / Comments
0.8 (June 2002)	First release of preliminary datasheet.
0.81 (July 2002)	Many edits to all sections
1.0 (November 2002)	Minor engineering edits.
1.1 (March 2003)	See Table A-2 "Edits to Current Revision" for details.

Table A-2. Edits to Current Revision

Section	Revision
Section "Features"	• Added “Is not the subtractive decode agent” under Sub-ISA Bus Interface. • Under Voltages, Internal Logic changed voltage for 266 MHz to 2.0V and changed voltage for 300 MHz to TBD.
Section 5.0 "Core Logic Module"	• Added “Is not the subtractive decode agent” under Sub-ISA Bus Interface. • Changed all references to 4-byte boundary in audio to 32-byte boundary. • Added sentence to third bullet under Physical Region Descriptor Format on page 144.
Section 7.0 "Electrical Specifications"	• Changed Min, Typ, and Max frequency values for 266 MHz Core clock frequency to 1.9, 2.0, and 2.1V respectively in Table 7-2 on page 281. • Changed many values in Table 7-4 on page 283.
Section 8.0 "Package Specifications"	• Removed Min column from Table 8-1 on page 343. • Made row for 300 MHz engineering text and changed Core Voltage (Nominal) to 2.0V for 266 MHz in Table 8-2 on page 344. • Changed the watts to 4 in the Note in Example 2 on page 345.
Section Appendix A "Support Documentation"	• Changed row for 300 MHz to engineering text and changed V_{CORE} for 266 MHz to 2.0V in Table A.1.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Email: new.feedback@nsc.com
Tel: 1-800-272-9959

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 87 90

National Semiconductor Asia Pacific Customer Response Group
Tel: +65-62544466
Fax: +65-62504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507
Email: jpn.feedback@nsc.com

www.national.com

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.