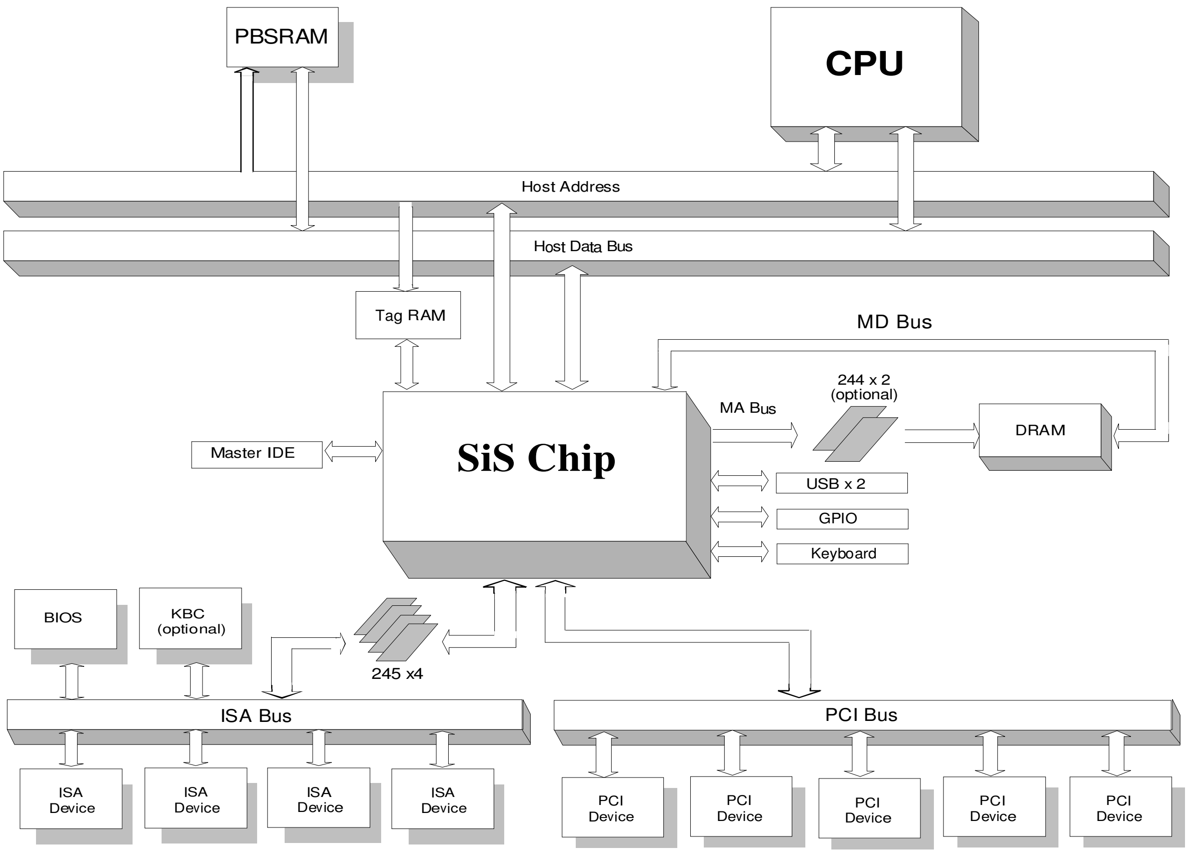
1. Introduction

Nowadays, several PC form factors exist in the PC board market, such as NLX, LPX, ATX and Baby-AT form factors. Due to the different placements of the form factor, PC chipsets should be prepared for different board layouts. As a result, SiS chips based on compatible logic design provide two series of chipsets, SiS 5581 and SiS 5582, to assist board designers for their board layouts.

SiS 5581's pin assignment is based on NLX, and LPX form factor, while SiS 5582's is defined on the basis of ATX and Baby-AT form factors. In the next few chapters, you will read "SiS Chip" which indicates either SiS 5581 or 5582 chipsets, decided by the placements of form factors on PC boards of customers.
The SiS Chip consists of Host-to-PCI bridge function, PCI to ISA bridge function, PCI IDE function, Universal Serial Bus host/hub function, Integrated RTC and Integrated Keyboard Controller.

SiS Chip supports Enhanced Power Management, including legacy Power Management Unit and Advanced Configuration and Power Interface (ACPI). It also supports ATA Synchronous DMA transfer protocol to improve the IDE performance and Common Architecture for moving ISA function to PCI to improve system performance. The system block diagram is shown below :



2. Features

- **Support Intel Pentium CPU and other compatible CPU host bus at 50/55/60/66/75 MHz**
- **Support CPU with MMX feature**
- **Support the Pipelined Address Mode of Pentium CPU**
- **Support the Full 64-bit Pentium Processor data Bus**
- **Meet PC97 Requirements**
- **Integrated Second Level (L2) Cache Controller**
  - Write Back Cache Modes
  - 8 bits or 7 bits Tag with Direct Mapped Cache Organization
  - Integrated 16K bits Dirty RAM
  - Support Pipelined Burst SRAM
  - Support 256 KBytes and 512 KBytes Cache Sizes
  - Cache Hit Read/Write Cycle of 3-1-1-1
  - Cache Back-to-Back Read/Write Cycle of 3-1-1-1-1-1-1-1
- **Integrated DRAM Controller**
  - Support 6 RAS Line (3 Banks) of FPM/EDO/SDRAM DIMMs/SIMMs
  - Support 2Mbytes to 384Mbytes of main memory
  - Support Cacheable DRAM Sizes up to 128 MBytes.
  - Support 256K/512K/1M/2M/4M/8M/16M/32M x N FPM/EDO/SDRAM DRAM
  - Support 64 Mb DRAM Technology
  - Support 3.3V or 5V DRAM.
  - Supports Symmetrical and Asymmetrical DRAM.
  - Support 32 bits/64 bits mixed mode configuration
  - Support Concurrent Write Back
  - Support CAS before RAS Refresh
  - Support Relocation of System Management Memory
  - Programmable CAS#, RAS#, RAMWE# and MA Driving Current.
  - Fully Configurable for the Characteristic of Shadow RAM (640 KBytes to 1 MBytes)
  - Support FPM DRAM 5-3-3-3(-3-3-3-3) Burst Read Cycles
  - Support EDO DRAM 5-2-2-2(-2-2-2-2) Burst Read Cycles
  - Support SDRAM 6-1-1-1(-2-1-1-1) Burst Read Cycles
  - Support X-1-1-1/X-2-2-2/X-3-3-3 Burst Write Cycles
  - Support 8 Qword Deep Buffer for Read/Write Reordering, Dword Merging and 3/2-1-1-1 Post write Cycles
  - Two Programmable Non-Cacheable Regions
  - Option to Disable Local Memory in Non-Cacheable Regions
  - Shadow RAM in Increments of 16 KBytes

- **Integrated PMU Controller**
  - Meet ACPI Requirements
  - Support Both ACPI and Legacy PMU
  - Support Suspend to Disk
  - Support SMM Mode of CPU
  - Support CPU Stop Clock
  - Support Power Button for ACPI function
  - Support Automatic Power Control for system power off function
  - Support Modem Ring-in, RTC Alarm Wake up
  - Support Thermal Detection
  - Support GPIOs, and GPOs for External Devices Control
  - Support Programmable Chip Select
- **Provides High Performance PCI Arbiter.**
  - Support up to 5 PCI Masters
  - Support Rotating Priority Mechanism
  - Hidden Arbitration Scheme Minimizes Arbitration Overhead.
  - Support Concurrency between CPU to Memory and PCI to PCI.
- **Integrated Host-to-PCI Bridge**
  - Support Asynchronous and Synchronous PCI Clock
  - Translates the CPU Cycles into the PCI Bus Cycles
  - Provides CPU-to-PCI Read Assembly and Write Disassembly Mechanism
  - Translates Sequential CPU-to-PCI Memory Write Cycles into PCI Burst Cycles
  - Zero Wait State Burst Cycles
  - Support IDE Posted Write
  - Support Pipelined Process in CPU-to-PCI Access
  - Support Advance Snooping for PCI Master Bursting
  - Maximum PCI Burst Transfer from 256 Bytes to 4 KBytes
- **Integrated Posted Write Buffers and Read Prefetch Buffers to Increase System Performance**
  - CPU-to-Memory Posted Write Buffer (CTMFF) with 8 QW Deep, Always Sustains 0 Wait Performance on CPU-to-Memory.
  - CPU-to-Memory Read Buffer with 4 QW Deep
  - CPU-to-PCI Posted Write Buffer(CTPFF) with 8 DW Deep
  - PCI-to-Memory Posted Write Buffer(PTHFF) with 8 QW Deep, Always Streams 0 Wait Performance on PCI-to/from-Memory Access
  - PCI-to-Memory Read Prefetch Buffer(CTPFF) with 8 QW Deep
- **Integrated PCI-to-ISA Bridge**
  - Translates PCI Bus Cycles into ISA Bus Cycles
  - Translates ISA Master or DMA Cycles into PCI Bus Cycles
  - Provides a Dword Post Buffer for PCI to ISA Memory cycles

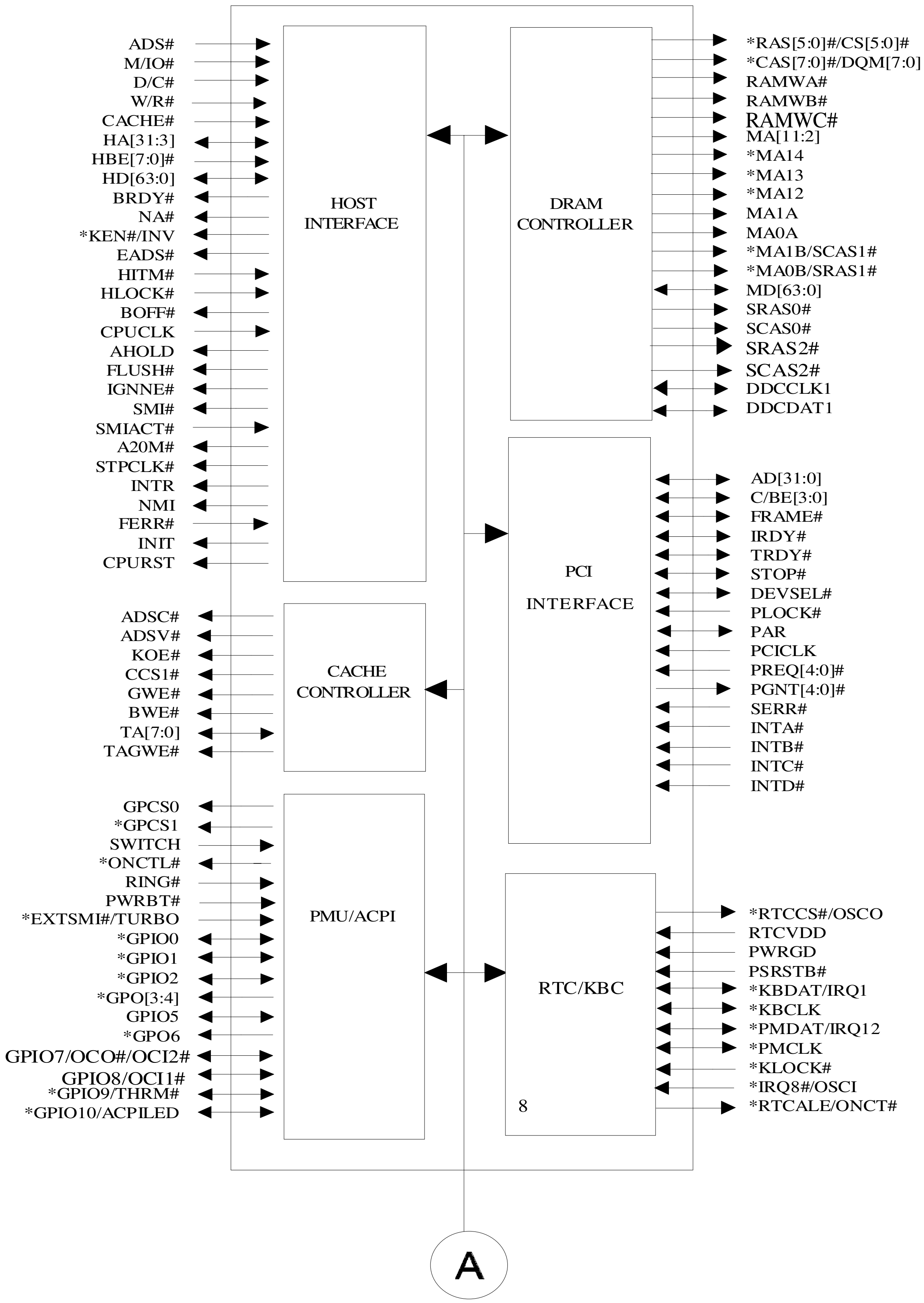
- Two 32 bit Prefetch/Post Buffers Enhance the DMA and ISA Master Performance
- Fully Compliant to PCI 2.1

- **Enhanced DMA Functions**
  - 8-, 16- bit DMA Data Transfer
  - ISA compatible, and Fast Type F DMA Cycles
  - Two 8237A Compatible DMA Controllers with Seven Independent Programmable Channels
  - Provides the Readability of the two 8237 Associated Registers
  - Support Distributed DMA
- **Built-in Two 8259A Interrupt Controllers**
  - 14 Independently Programmable Channels for Level- or Edge-triggered Interrupts
  - Provides the Readability of the two 8259A Associated Registers
  - Support Serial IRQ
- **Three Programmable 16-bit Counters compatible with 8254**
  - System Timer Interrupt
  - Generates Refresh Request
  - Speaker Tone Output
  - Provides the Readability of the 8254 Associated Registers
- **Built-in Keyboard Controller**
  - Hardwired Logic Provides Instant Response
  - Support PS/2 Mouse interface
  - Support Hot Key "Wake-up" Function
  - Capable of Enable/Disable Internal KBC and PS2 Mouse
- **Built-in Real Time Clock(RTC) with 256B CMOS SRAM**
  - Built-in up to one Month Alarm for ACPI
- **Fast PCI IDE Master/Slave Controller**
  - Bus Master Programming Interface for ATA Windows 95 Compliant Controller
  - Support PCI Bus Mastering
  - Plug and Play Compatible
  - Support Scatter and Gather
  - Support Dual Mode Operation - Native Mode and Compatibility Mode
  - Support IDE PIO Timing Mode 0, 1, 2 ,3 and 4
  - Support Multiword DMA Mode 0, 1, 2
  - Support Ultra DMA/33
  - Two Separate IDE Bus
  - Two 16 Dword FIFO for PCI Burst Transfers.
- **Universal Serial Bus Host Controller**
  - OpenHCI Host Controller with Root Hub
  - Two USB ports
  - Support Legacy Devices

- Support Over Current Detection

- **Support I^2C serial Bus**
- **Support the Reroutibility of the four PCI Interrupts**
- **Support 2Mb Flash ROM Interface**
- **Support NAND Tree for ball connectivity testing**
- **553-Balls BGA Package**
- **0.35µm 3.3V Technology**

2.1 Functional Block Diagram



To be continued on next page.

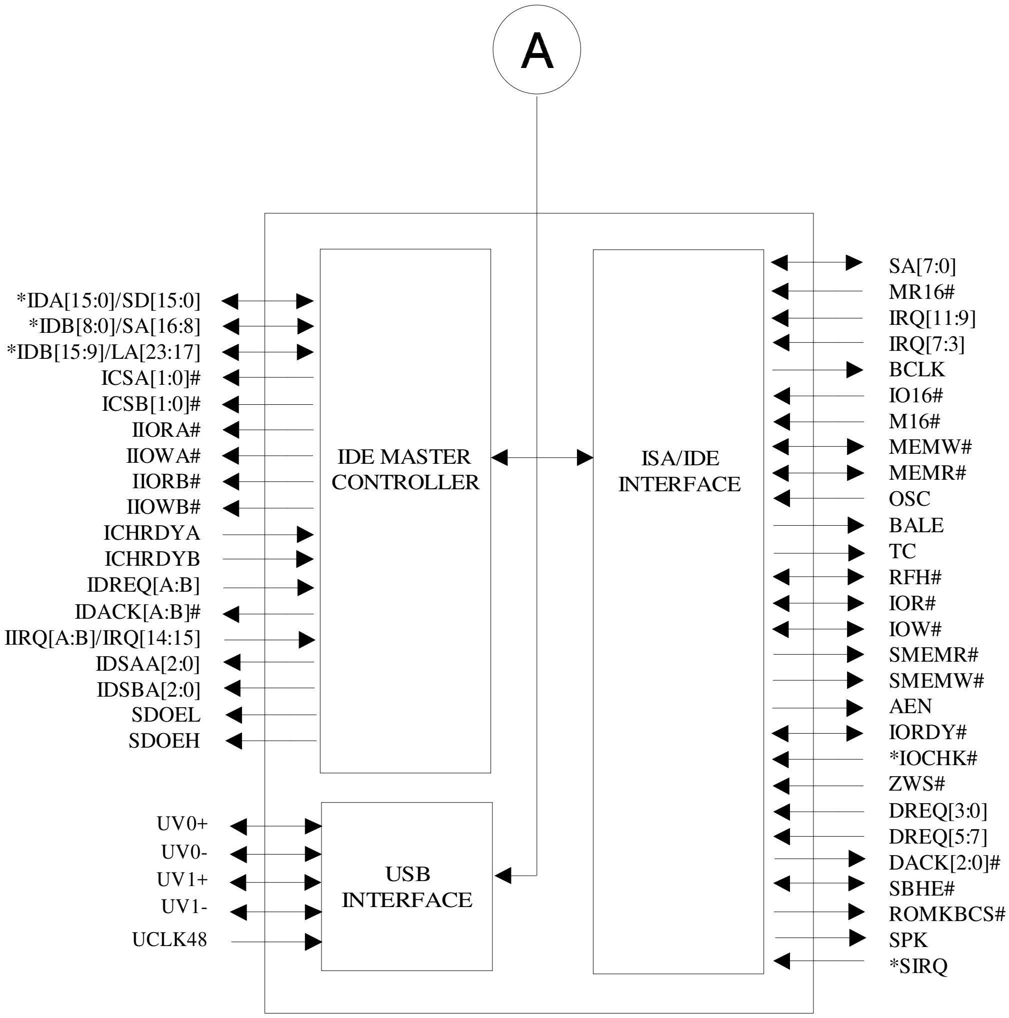


Figure 2-1

***Multi-function pins**

KEN#/INV	OSCI/IRQ8#	GPIO7/OCO#/OCI2#
RAS[5:0]#/CS[5:0]#	OSCO/RTCCS#	GPIO8/OCI1#
CAS[7:0]#/DQM[7:0]	ONCTL#/RTCALE	GPIO9/THRM#/IOCHK#
MA12/GPO3	KBDAT/IRQ1	GPIO10/ACPILED
MA13/GPO4	KBCLK/GPIO2	EXTSMI#/TURBO
MA14/GPO6	KLOCK#/GPIO0/RAMWC#	GPCS1/SIRQ
MA1B/SCAS1#	PMCLK/GPIO1	IDA[15:0]/SD[15:0]
MA0B/SRAS1#	PMDAT/IRQ12	IDB[8:0]/SA[16:8]
	IIRQ[A:B]/IRQ[14:15]	IDB[15:9]/LA[23:17]

3. Functional Description

3.1 Host Interface

The SiS Chip is designed to support Pentium CPU host interface at 75/66.667/60/55/50MHz. The host data bus and the DRAM bus are 64-bit wide.

The SiS Chip supports the pipelined addressing mode of the Pentium CPU by issuing the next address signal, NA#. NA# signal is asserted except single read DRAM cycle.

The SiS Chip supports the CPU L1 write back (WB) or write through (WT) cache policies and the SiS Chip L2 WB cache policies. The L1 cache is snooped by the assertion of EADS# when the CPU is put in the HOLD state.

The SiS Chip issues AHOLD to the Pentium CPU in response to the assertion of PCI master requests. Once the AHOLD is asserted, SiS Chip does not immediately assert PGNT[3:0]# until both the CPU to PCI posted write buffer and the memory write buffer are empty. During inquire cycles, the AHOLD may be negated temporarily to allow the CPU to write back the inquired hit modified line to L2 or DRAM.

3.2 Cache Controller

The built-in L2 Cache Controller uses a direct-mapped scheme, which can be configured as in the write back mode. SiS chip also supports the write through mode, but this function only for the cache sizing. Pipelined burst SRAMs are supported.

SiS Chip supports SRAM types auto-detection and auto-sizing. Table 3-1 shows the cache sizes that are supported by the SiS Chip when using synchronous SRAM, with the corresponding TAG RAM sizes, data RAM sizes, and cacheable memory sizes.

Table 3-1 Cache Size with 8-bit tag

Cache Size	Data RAM	Tag RAM	Cacheable Size
256K	32Kx32x2	8Kx8	64MB
512K	32Kx32x4	16Kx8	128MB
512K	64Kx32x2	16Kx8	128MB

The SiS Chip also provides an alternative to save the dirty SRAM chip. This is accomplished by integrated 16Kb Dirty RAM.

3.3 DRAM Controller

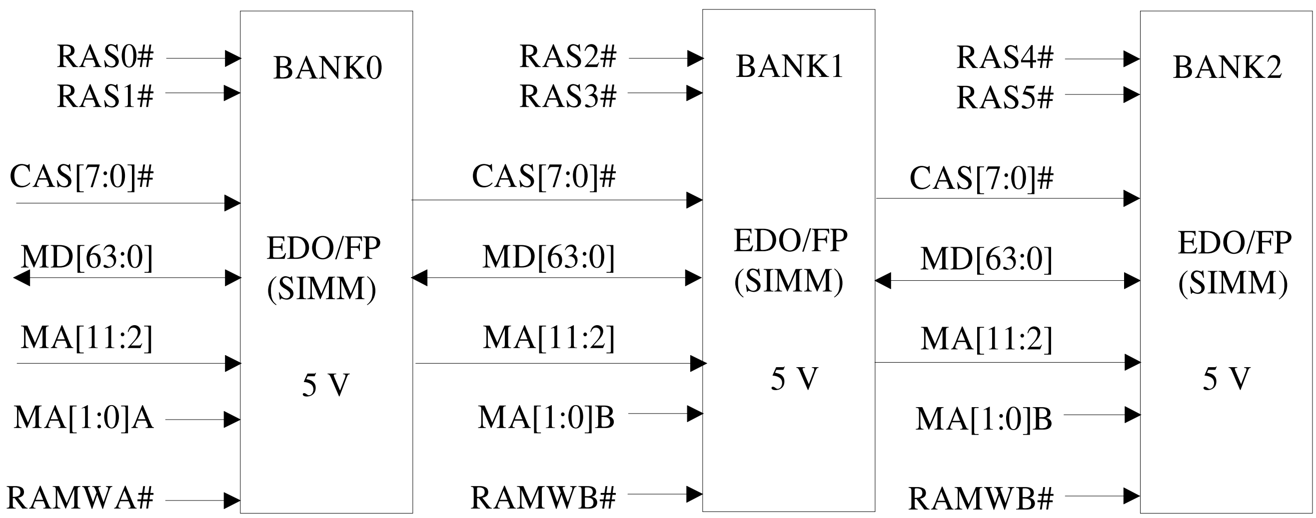
3.3.1 DRAM Type

The SiS Chip can support up to 384MBytes of DRAMs size and each bank could be single or double sided 64 bits FPM (Fast Page mode) DRAM, EDO (Extended Data Output) DRAM, and SDRAM (Synchronous DRAM) DRAM. Half populated bank(32-bit) is also supported. The installed EDO/FPM DRAM type can be 256K, 512k, 1M, 2M, 4M, 8M or 16M bit deep by n bit wide DRAMs, and both symmetrical and asymmetrical type DRAM are supported. It also supports SDRAM 1M, 2M, 4M, 8M, 16M or 32M bit deep by n bit wide DRAMs, and both single and double sided. It is also permissible to mix the DRAMs (FPM/EDO/SDRAM) bank by bank and the corresponding DRAM timing will be switched automatically according to register settings.

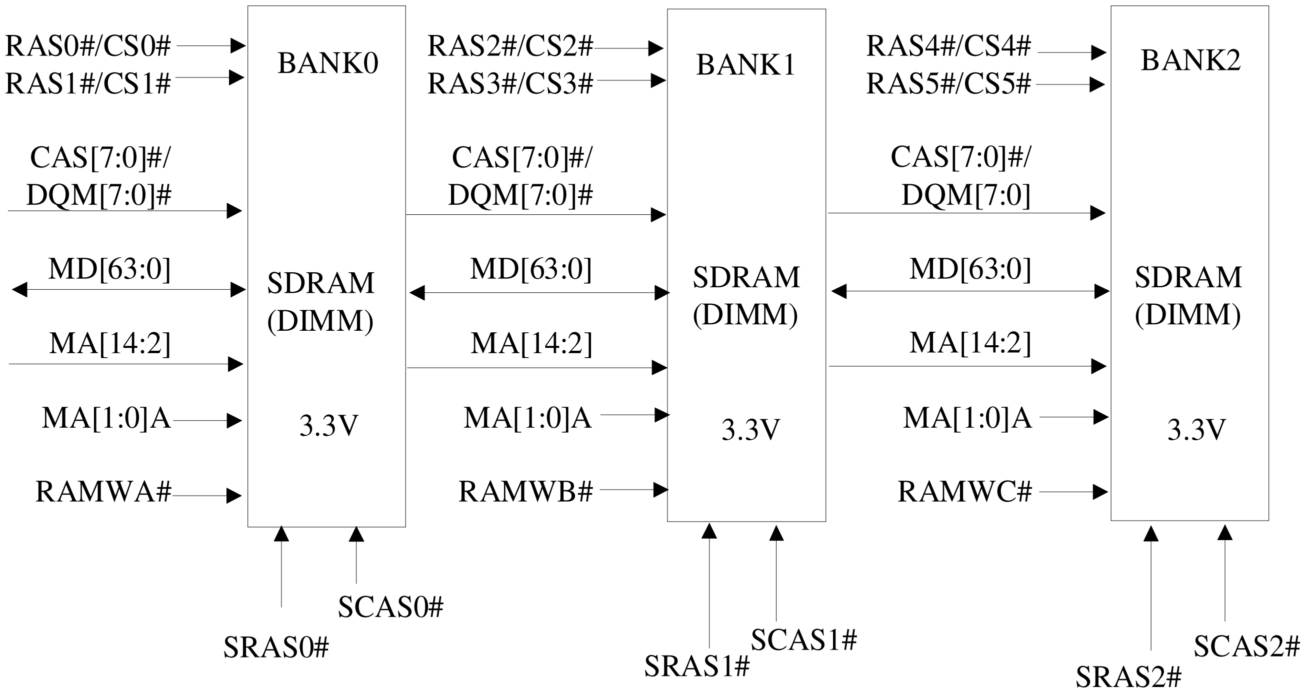
3.3.2 DRAM Configuration

SiS Chip supports six rows of DRAMs each 64 bits wide. Regarding to these six row lines, BANK0 use RAS[1:0], BANK1 use RAS[3:2], BANK2 use RAS[5:4]. The six rows of DRAMs may be implemented in three banks of single-sided SIMMs for FPM/EDO DRAM, three banks of double-sided SIMMs , three banks of SDRAM or any other combinations as required. Access to the rows are not interleaved and need not to be populated starting from row 0 or in consecutive sequence.
The SiS Chip can support EDO, FPM and SDRAM. SDRAM, EDO and FPM DRAMs can not be mixed in one bank, however, different banks can use different types of DRAM.
The basic configurations are shown as the following sections:

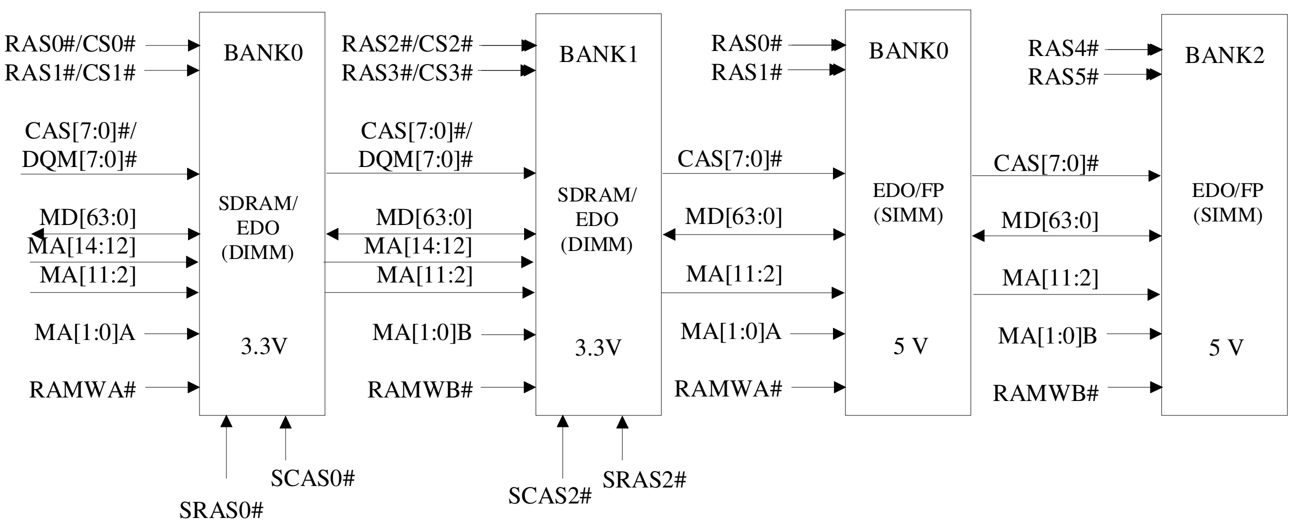
EDO/FPM DRAM Configuration (4 SIMM/6 SIMM):



SDRAM Configuration (2 DIMM/3 DIMM):



DRAM Type Mixed Configuration: EDO/FPM + SDRAM (4 SIMM + 2 DIMM)



NOTE :

1. SiS Chip only support six rows (3 banks) DRAMs.
2. It is recommended that board designer must follow DC characteristics of each type DRAM (SDRAM, EDO, FPM) to design the portion of DRAM in DRAM mode mixed configuration.
4. If the KLOCK# or GPIO function are not needed, the Bank2 DRAM can used the RAMWC# instead of RAMWB#.
5. Please refer to "Multiplexed pins" section to define the pin for each function.

3.3.3 DRAM Scramble Table

The DRAM scramble table contains information for memory address mapping. These tables provide the translation between CPU host address and memory Row and Column address.
There are several memory address mapping: 64-bit mapping and 32-bit mapping for FPM/EDO DRAM, 2 Banks and 4 Banks mapping for SDRAM that SiS Chip supports:

64-bit mapping table for FPM/EDO DRAM

a. **Symmetric:**

Type	256K (9x9)		1M (10x10)		4M (11x11)		16M (12x12)	
Address	Row	Column	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3	15	3
MA1	16	4	16	4	16	4	16	4
MA2	17	5	17	5	17	5	17	5
MA3	18	6	18	6	18	6	18	6
MA4	19	7	19	7	19	7	19	7
MA5	20	8	20	8	20	8	20	8
MA6	12	9	21	9	21	9	21	9
MA7	13	10	22	10	22	10	22	10
MA8	14	11	14	11	23	11	23	11
MA9	NA	NA	13	12	24	12	24	12
MA10	NA	NA	NA	NA	14	13	25	13
MA11	NA	NA	NA	NA	NA	NA	26	14

b. Asymmetric:

Type	512K (10x9)		1M (11x9)		2M (11x10)	
Address	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3
MA1	16	4	16	4	16	4
MA2	17	5	17	5	17	5
MA3	18	6	18	6	18	6
MA4	19	7	19	7	19	7
MA5	20	8	20	8	20	8
MA6	21	9	21	9	21	9
MA7	13	10	22	10	22	10
MA8	14	11	14	11	23	11
MA9	12	NA	12	NA	13	12
MA10	NA	NA	13	NA	14	NA
MA11	NA	NA	NA	NA	NA	NA

Type	1M (12x8)		2M (12x9)		4M (12x10)		8M (12x11)	
Address	Row	Column	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3	15	3
MA1	16	4	16	4	16	4	16	4
MA2	17	5	17	5	17	5	17	5
MA3	18	6	18	6	18	6	18	6
MA4	19	7	19	7	19	7	19	7
MA5	20	8	20	8	20	8	20	8
MA6	21	9	21	9	21	9	21	9
MA7	22	10	22	10	22	10	22	10
MA8	11	NA	23	11	23	11	23	11
MA9	12	NA	12	NA	24	12	24	12
MA10	13	NA	13	NA	13	NA	25	13
MA11	14	NA	14	NA	14	NA	14	NA

<u>32-bit mapping table for FPM/EDO DRAM</u>

a. Symmetric:

Type	256K (9x9)		1M (10x10)		4M (11x11)		16M (12x12)	
Address	Row	Column	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3	15	3
MA1	16	4	16	4	16	4	16	4
MA2	17	5	17	5	17	5	17	5
MA3	18	6	18	6	18	6	18	6
MA4	19	7	19	7	19	7	19	7
MA5	11	8	20	8	20	8	20	8
MA6	12	9	21	9	21	9	21	9
MA7	13	2	13	2	22	2	22	2
MA8	14	10	14	10	23	10	23	10
MA9	NA	NA	12	11	13	11	24	11
MA10	NA	NA	NA	NA	14	12	25	12
MA11	NA	NA	NA	NA	NA	NA	14	13

b. Asymmetric:

Type	512K (10x9)		1M (11x9)		2M (11x10)	
Address	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3
MA1	16	4	16	4	16	4
MA2	17	5	17	5	17	5
MA3	18	6	18	6	18	6
MA4	19	7	19	7	19	7
MA5	20	8	20	8	20	8
MA6	14	9	21	9	21	9
MA7	13	2	13	2	22	2
MA8	11	10	11	10	14	10
MA9	12	NA	12	NA	12	11
MA10	NA	NA	14	NA	13	NA
MA11	NA	NA	NA	NA	NA	NA

Type	1M (12x8)		2M (12x9)		4M (12x10)		8M (12x11)	
Address	Row	Column	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3	15	3
MA1	16	4	16	4	16	4	16	4
MA2	17	5	17	5	17	5	17	5
MA3	18	6	18	6	18	6	18	6
MA4	19	7	19	7	19	7	19	7
MA5	20	8	20	8	20	8	20	8
MA6	21	9	21	9	21	9	21	9
MA7	10	2	22	2	22	2	22	2
MA8	11	NA	11	10	23	10	23	10
MA9	12	NA	12	NA	12	11	24	11
MA10	13	NA	13	NA	13	NA	13	12
MA11	14	NA	14	NA	14	NA	14	NA

MA Mapping table for SDRAM

a. 2 Banks Device SDRAM Type:

Type	1M (1x11x8)		2M (1x11x9)		4M (1x11x10)	
Address	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3
MA1	16	4	16	4	16	4
MA2	17	5	17	5	17	5
MA3	18	6	18	6	18	6
MA4	19	7	19	7	19	7
MA5	20	8	20	8	20	8
MA6	21	9	21	9	21	9
MA7	22	10	22	10	22	10
MA8	12	NA	23	11	23	11
MA9	13	NA	13	NA	24	12
MA10	14	NA	14	NA	14	NA
MA11	11	11	12	12	13	13
MA12	NA	NA	NA	NA	NA	NA
MA13	NA	NA	NA	NA	NA	NA
MA14	NA	NA	NA	NA	NA	NA

Type	4M (1x13x8)		8M (1x13x9)		16M (1x13x10)	
Address	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3
MA1	16	4	16	4	16	4
MA2	17	5	17	5	17	5
MA3	18	6	18	6	18	6
MA4	19	7	19	7	19	7
MA5	20	8	20	8	20	8
MA6	21	9	21	9	21	9
MA7	22	10	22	10	22	10
MA8	12	NA	23	11	23	11
MA9	13	NA	13	NA	24	12
MA10	14	NA	14	NA	14	NA
MA11	11	11	12	12	13	13
MA12	NA	NA	NA	NA	NA	NA
MA13	23	NA	24	NA	25	NA
MA14	24	NA	25	NA	26	NA

b. 4 banks Device SDRAM Type:

Type	2M (2x11x8))		4M (2x12x8)		8M (2x12x9)		16M (2x12x10)	
Address	Row	Column	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3	15	3
MA1	16	4	16	4	16	4	16	4
MA2	17	5	17	5	17	5	17	5
MA3	18	6	18	6	18	6	18	6
MA4	19	7	19	7	19	7	19	7
MA5	20	8	20	8	20	8	20	8
MA6	21	9	21	9	21	9	21	9
MA7	22	10	22	10	22	10	22	10
MA8	23	NA	23	NA	23	11	23	11
MA9	13	NA	13	NA	24	NA	24	12
MA10	14	NA	14	NA	14	NA	25	NA
MA11	11	11	11	11	12	12	13	13
MA12	12	12	12	12	13	13	14	14
MA13	NA	NA	24	NA	25	NA	26	NA
MA14	NA	NA	NA	NA	NA	NA	NA	NA

Type	8M (2x13x8)		16M (2x13x9)		32M (2x13x10)	
Address	Row	Column	Row	Column	Row	Column
MA0	15	3	15	3	15	3
MA1	16	4	16	4	16	4
MA2	17	5	17	5	17	5
MA3	18	6	18	6	18	6
MA4	19	7	19	7	19	7
MA5	20	8	20	8	20	8
MA6	21	9	21	9	21	9
MA7	22	10	22	10	22	10
MA8	23	NA	23	11	23	11
MA9	13	NA	24	NA	24	12
MA10	14	NA	14	NA	25	NA
MA11	11	11	12	12	13	13
MA12	12	12	13	13	14	14
MA13	24	NA	25	NA	26	NA
MA14	25	NA	26	NA	27	NA

3.3.4 DRAM Detection Sequence

SiS Chip supports six rows (three banks) DRAMs for DRAM's SIMMs/DIMMs from row0 to row5. The DRAMs detection sequence is a row-based detection sequence, it is performed by the BIOS row by row and fulfill the DRAM configuration information into the corresponding DRAM configuration registers. The following steps will be described the DRAM detection sequence.

Step 1. To detect if there is any DRAM populated in row N, SiS Chip set this row with maximum DRAM size, then write/read the same address with test pattern by the normal DRAM read/write timing and compare the data. If the read data is the same as the write pattern, then there exists DRAM in the rowN; otherwise, proceed the SDRAM detection from step 3.

Step 2. If the DRAM is detected in the rowN by step 1, SiS Chip treat it as EDO or FPM DRAM. SiS Chip first write test pattern into DRAM, then set register 55h bit 6 (EDO test bit) to be "1" in PCI/memory bridge configuration register, and do the read, compare test pattern from the same DRAM location. The EDO test bit will delay the data forward to CPU after 4096 CPU clock. If the CPU still get the right data, then EDO mode DRAM is set to this row; otherwise, the FP mode DRAM is set. Go to step 8.

Step 3. If the DRAM is detected not populated in row N by normal write/read procedure, SiS Chip check if there is SDRAM exist in this row or not. SiS Chip first assume the DRAM mode is SDRAM (set bit [7:6] of register 60h/61h/62h to be "11" in Host to PCI bridge configuration register, it depends on which bank is under detection), and then do the SDRAM initialization procedure from step 4 to step 7.

Step 4. Set register 56h bit 3 to "1" to enable SDRAM sizing, then set register 57h bit 7 to be "1", register 57h bit 7 will drive a precharge command to SDRAM, then disable this bit (set to be "0").

Step 5. Set register 57h bit 6 to be "1", this bit will drive a "mode register set" (MRS) command to SDRAM. When SDRAM receive MRS command, it will load the needed information (Toggle/Linear mode, CAS Latency) into SDRAM. After doing MRS, disable this bit (set to be "0").

Step 6. Set register 57h bit 5 to be "1" at least two times, then SDRAM will perform refresh cycle at least two times before the normal operation. Disable this bit (set to be "0").

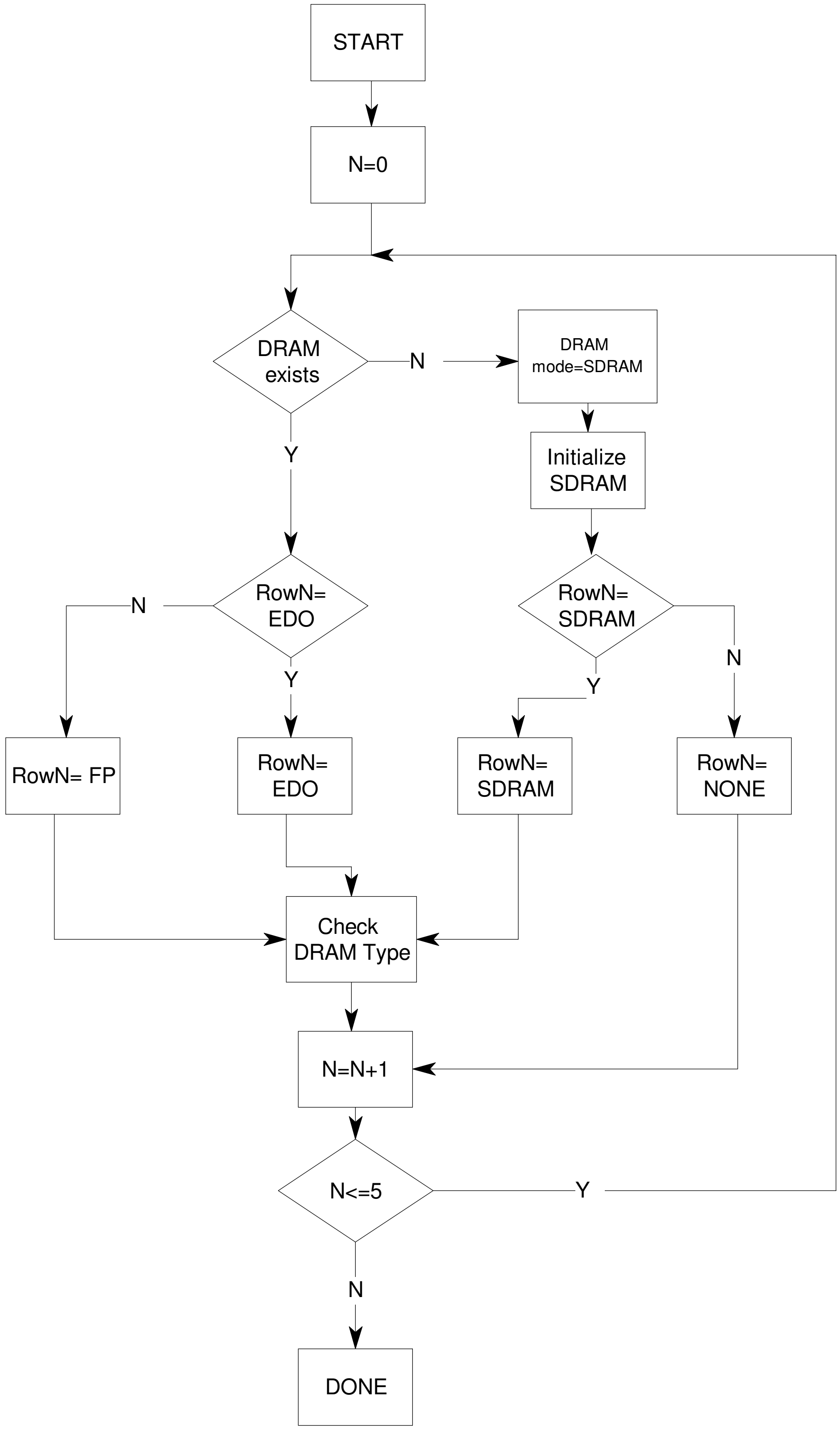
Step 7. Write/Read the test pattern into SDRAM, then compare the data. If the data is correct, SDRAM is detected, and set rowN as SDRAM; otherwise, rowN is no DRAM populated. Set Register 56h bit 3 to "0".

Step 8. After DRAM mode is set, SiS Chip do DRAM sizing by write/read test pattern based on the MA mapping table.

Step 9. Repeat from step 1 to step 8 to detect the other rows.

Note : The value of N is from 0 to 5.

The following will be shown the flow chart of DRAM Detection Sequence.



DRAM Detection Sequence

Figure 3-1

3.3.5 DRAM Performance

All the DRAM cycles are synchronous with the CPU clock. The following table shows the different possible speed settings that depend on different DRAM type, RAS# setting, CAS# setting, and so forth.

Cycle Type	DRAM type	75Mhz	66/60/50 Mhz	Note
Read Page Hit	EDO	5-2-2-2	5-2-2-2	
	FPM	5-3-3-3	5-3-3-3	*1
	SDRAM	6-1-1-1	6-1-1-1	CL=2
		7-1-1-1	7-1-1-1	CL=3
Read Row Start	EDO	9-2-2-2	8-2-2-2	*2
	FPM	9-3-3-3	8-3-3-3	*1
	SDRAM	10-1-1-1	9-1-1-1	CL=2
		11-1-1-1	10-1-1-1	CL=3
Read Page Miss	EDO	13-2-2-2	12-2-2-2	*3, *4
	FPM	13-3-3-3	12-3-3-3	*1 *3 *4
	SDRAM	11-1-1-1	10-1-1-1	CL=2, *4
		12-1-1-1	11-1-1-1	CL=3, *4
Back-to-Back Burst Read Page Hit	EDO	5-2-2-2-2-2-2-2...	5-2-2-2-2-2-2-2...	
	FPM	5-3-3-3-3-3-3-3...	5-3-3-3-3-3-3-3...	
	SDRAM	6-1-1-1-2-1-1-1...	6-1-1-1-2-1-1-1...	CL=2, *5
		7-1-1-1-3-1-1-1...	7-1-1-1-3-1-1-1...	CL=3, *5
Posted Write	EDO/FPM/ SDRAM	3-1-1-1	3-1-1-1	
Write Retire Rate (Buffer to DRAM)	EDO	2-2-2	2-2-2	
	FPM	3-3-3	3-3-3	*1
	SDRAM	2-2-2	1-1-1/2-2-2	CL=2
			1-1-1/2-2-2	CL=3
Write Page Hit	EDO	2	2	
	FPM	2	2	
	SDRAM	2	2	CL=2
		2	2	CL=3
Write Row Start	EDO	7	6	
	FPM	7	6	
	SDRAM	7	6	CL=2
		7	6	CL=3
Write Page Miss	EDO	10	9	
	FPM	10	9	
	SDRAM	10	9	CL=2
		10	9	CL=3

Note: EDO CAS# width=1T, FPM CAS# width=2T, CAS precharge time=1T, 60ns DRAM.

*1 X-4-4-4 is for both CAS pulse width and CAS precharge time are 2 CPU clocks.
*2 It is for RAS to CAS time of 3 CPU clocks.
*3 It is for RAS pre-charge time of 4 CPU clocks, RAS to CAS time of 3 CPU clocks.
*4 EDO : 9-2-2-2, FPM : 9-3-3-3 and SDRAM : 8-1-1-1 during pipelined cycle.
*5 6-1-1-1-4-1-1-1 and 7-1-1-1-5-1-1-1 for L2 Cache is populated.

3.3.6 CPU to DRAM Posted Write FIFOs

There is a built-in CPU to Memory posted write buffer with 8 QWord deep (CTMFF). All the write access to DRAM will be buffered. For the CPU read miss / Line fill cycles, the write-back data from the second level cache will be buffered first, and right after the data had been posted write into the FIFO, CPU can performs the read operation by the memory controller starting to read data from DRAMs. The buffered data are then written to DRAM whenever no any other read DRAM request comes. With this concurrent write back policy, many wait states are eliminated. If there comes a bunch of continuous DRAM write cycles, some ones will be pending if the CTMFF is full.

3.3.7 32-bit (Half-Populated) DRAM Access

For the read access, there will be either single or burst read cycle to access the DRAM which depends on the cacheability of the cycle. If the current DRAM configuration is half-populated bank, then the SiS Chip will assert 8 consecutive cycles to access DRAM for the burst cycle. For the single cycle that only accesses DRAM within a DWord, the SiS Chip will only issue one cycle to access DRAM. For the single cycle that accesses one Qword or cross DWord boundary, the SiS Chip will issue two consecutive cycles to access DRAM.

3.3.8 Arbiter

The arbiter is the interface between the DRAM controller and the host which can access DRAMs. In addition to pass or translate the information from outside to DRAM controller, arbiter is also responsible for which master has higher priority to access DRAMs. The arbiter treats different DRAM access request as DRAM master, and that makes there be 5 masters which are trying to access DRAMs by sending their request to the arbiter. After one of them get the grant from the arbiter, it owns DRAM bus and begins to do memory data transaction.
The masters are: CPU read request, PCI master, Posted write FIFO write request, and Refresh request. The order of these masters shown above also stands for their priority to access memory.

3.3.9 Refresh cycle

The refresh cycle will occur every 15.6 us, 62.4 us, 124.8 us and 187.2 us and depend on setting the register 53h bits[2:1] in Host to PCI bridge configuration space. It is timed by a counter of 14Mhz input. The CAS[7:0]# will be asserted at the same time, and the RAS[5:0]# are asserted sequentially.

3.4 PCI bridge

SiS Chip is able to operate at both asynchronous and synchronous PCI clocks. Synchronous mode is provided for those synchronous system to improve the overall system performance.
While in the PCI master write cycles, post-write is always performed. And function of Write Merge with CPU-to-DRAM post-write buffer is incorporated to eliminate the penalty of snooping write-back. On the other hand, prefetch is enabled for master read cycles by default, and such function could be disabled optionally. And, Direct-Read from CPU-to-DRAM post-write buffer is implemented to eliminate the overhead of snooping write-back also.
In addition to Write-Merge and Direct-Read, Snoop-Ahead also hides the overhead of inquiry cycles for master to main memory cycles. These key functions, Write-Merge, Direct-Read and Snoop-Ahead, achieve the purpose of zero wait for PCI burst transfer.
The post-write and prefetch buffers are both 16 Double-Word deep FIFOs .

3.4.1 Snooping Control

In order to maintain the cache consistency while PCI master accesses to main memory, SiS Chip performs inquiry cycle to snoop L1 and L2 caches before PCI masters really read from or write to memory. For the purpose of snooping, AHOLD is asserted to force the Pentium-like processors to float its address bus as soon as PCI master requests the PCI bus. Such host bus hold mechanism is completed by an AHOLD/BOFF# process and will be depicted later. Since the inquiry cycle is the major penalty for PCI master cycles, SiS Chip builds in a high performance snoop-ahead mechanism to incorporate the zero wait requirement of PCI bus transactions.
The main idea of “snoop-ahead” is to do memory operations and inquiry cycle simultaneously. For example, when transferring the Ln line of data, SiS Chip also performs the Ln+1 line of inquiry cycle in the mean while.

3.4.2 AHOLD/BOFF# Process and Arbiter Interface

In order to perform inquiry cycles, SiS Chip uses AHOLD to hold address bus of Pentium-like CPUs. While PCI master asserts PREQ#, SiS Chip will drive AHOLD firstly. And, if PCI master operates a peer-to-peer transaction, SiS Chip will deasserts AHOLD to permit CPU to do memory cycles concurrently. Otherwise, SiS Chip retains AHOLD signal until PREQ# is inactive and bus transaction completes.

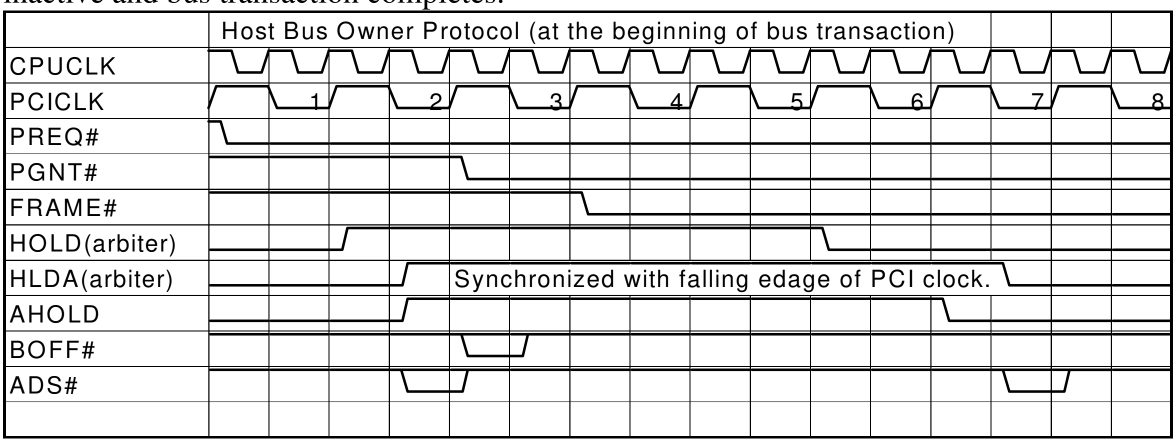
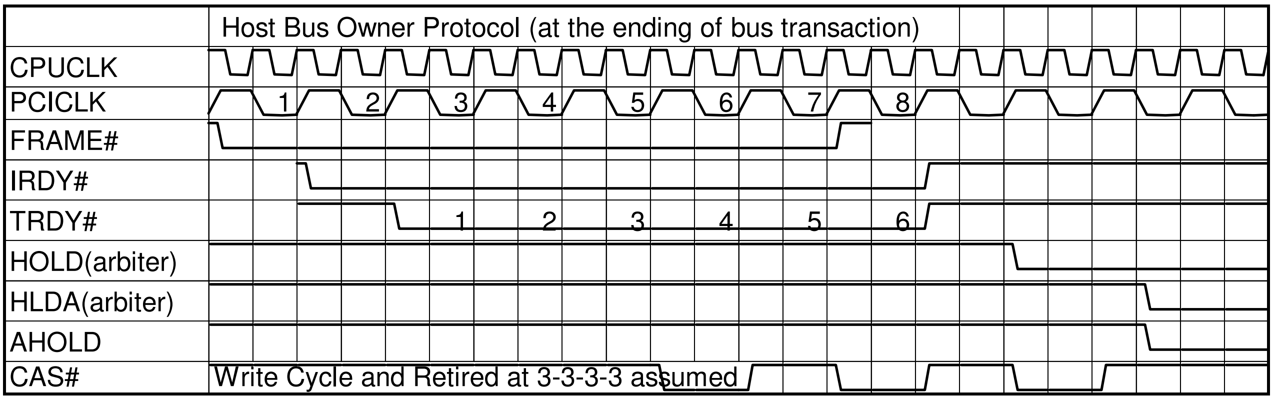


Figure 3-2



Note: HOLD & HLDA are internal signals.

Figure 3-3

3.4.3 Target Initiated Termination

In general, SiS Chip is capable to complete all the requests to access main memory from PCI masters until master terminates the transaction actively. Sometimes, as SiS Chip is unable to respond or is unable to burst, it will initiate to terminate bus transactions and STOP# will be issued by doing Retry or Disconnect.

3.4.4 Target Retry

SiS Chip may operate Target Retry for one of two reasons:

1. Whenever a PCI master tries to access main memory and SiS Chip is locked previously by another agent, Target Retry will be signaled.
2. Once SiS Chip can‘t meet the requirement of target initial latency, Target Retry is used and no data is transferred.

3.4.5 Disconnect With Data

In some situations, such as the burst crosses a resource boundary or a resource conflict, SiS Chip might be temporarily unable to continue bursting, and, therefore, SiS Chip concludes an active termination.

1. SiS Chip supports PCI burst transfers, the bursting length can be 256 bytes, 512 bytes, 1K bytes, 2K bytes, or 4K bytes and depend on setting the register 80h bits[7:5] in Host to PCI bridge configuration space. A burst will be terminated by doing Disconnect if the transfer goes across the programmed bursting length. In this way, at most 128 cache lines of data can be uninterruptedly transferred no matter what the status they are in L1 and L2 cache. One reason for the constraint is that page miss may occur only once at the beginning of the entire bursting transaction since the maximum bursting length is always within the page size in any of the used DRAM.
2. If advanced snoop function is disabled, PCI transaction will not cross the cache boundary and also causes a Disconnect operation. Since the heavy overhead of inquiry cycles is not preventable, and SiS Chip can‘t keep bursting transfer.

3.4.6 Disconnect Without Data

If Target Subsequent Latency timer expires, it causes SiS Chip to assert STOP# by doing Disconnect operation.

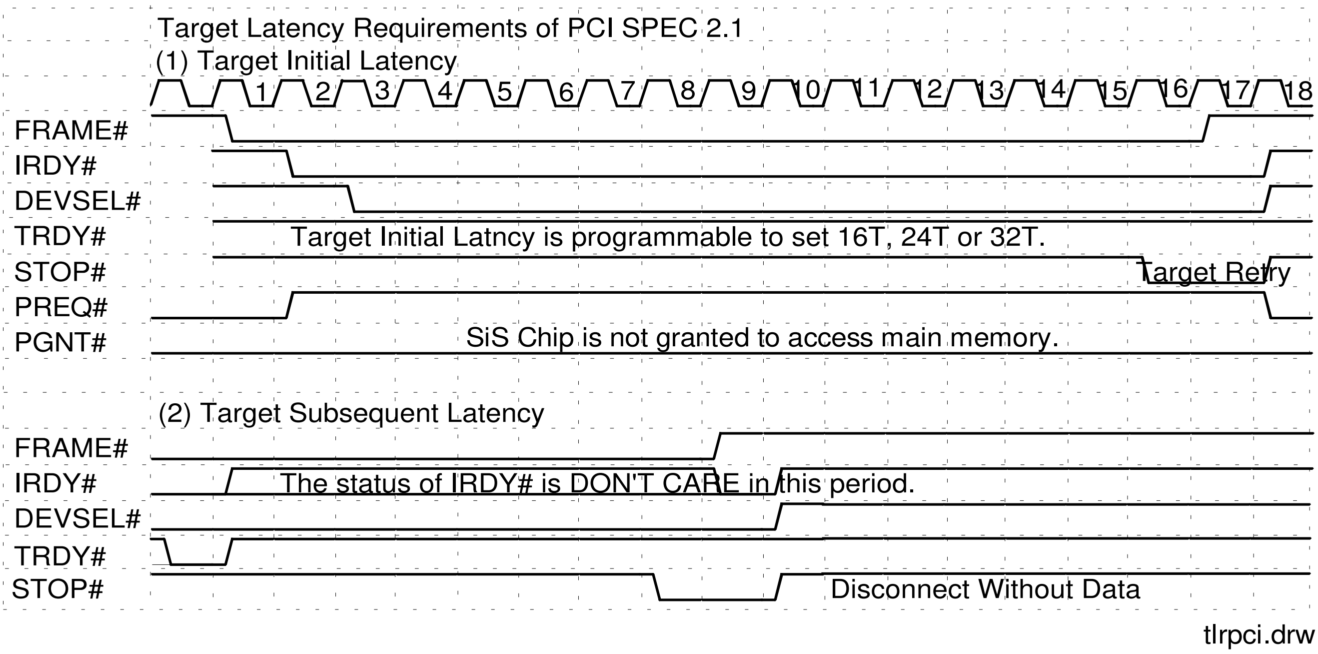


Figure 3-4

3.4.7 DATA Flow

The major two data paths are PCI->PTHFF->DRAM and DRAM->CTPFF->PCI for PCI master write DRAM cycles and read DRAM cycles, respectively. For cache system, if an inquiry cycle hits Pipeline Burst SRAM, SiS Chip would read from L2 directly, but write DRAM and L2 CACHE simultaneously.
Based on snooping result, there are additional data path that SiS Chip should perform.

Table Data Flow Based on Snooping Result

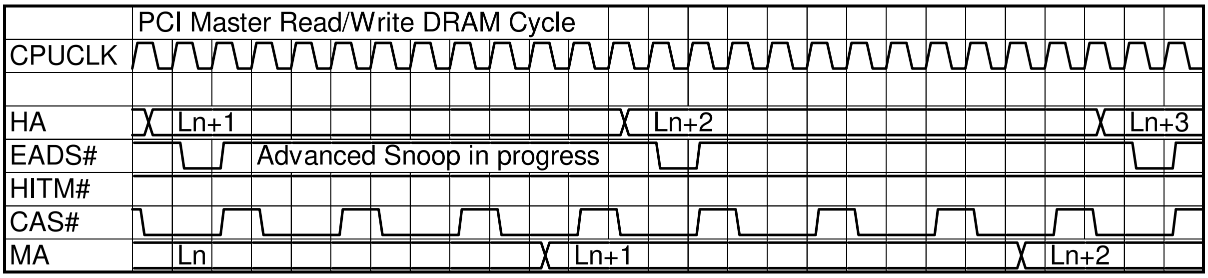
PCI Master Read Memory Cycle			
Result of Snoop			
Status of L1	Status of L2	Data Flow	Operation
Miss or Unmodified	Miss or None	DRAM -> CTPFF -> PCI	Read DRAM
	Hit and Not Dirty	DRAM -> CTPFF -> PCI	Read DRAM
	Hit and Dirty	L2 -> CTPFF -> PCI	Read L2
Hit Modified	Miss or None	L1 -> CTMFF & CTPFF CTPFF -> PCI	Direct Read
	Hit, Dirty or Not	L1 -> L2 & CTPFF CTPFF -> PCI	Direct Read

PCI Master Write Memory Cycle			
Result of Snoop			
Status of L1	Status of L2	Data Flow	PSL Operation
Miss or Unmodified	Miss or None	PCI -> PTHFF -> DRAM	Write DRAM
	Hit, Dirty or Not	PCI -> PTHFF -> L2&DRAM	Write DRAM&L2
Hit Modified	Miss or None	L1 -> CTMFF PTHFF & CTMFF -> DRAM	Write Merge
	Hit, Dirty or Not	L1 -> L2 PCI -> PTHFF -> L2&DRAM	Write DRAM&L2

Note: CTPFF means CPU-to-PCI Posted Write Buffer.
CTMFF means CPU-to-Memory Posted Write Buffer
PTMFF means PCI-to-Memory Posted Write Buffer

3.4.8 PCI Master Read/Write DRAM Cycle

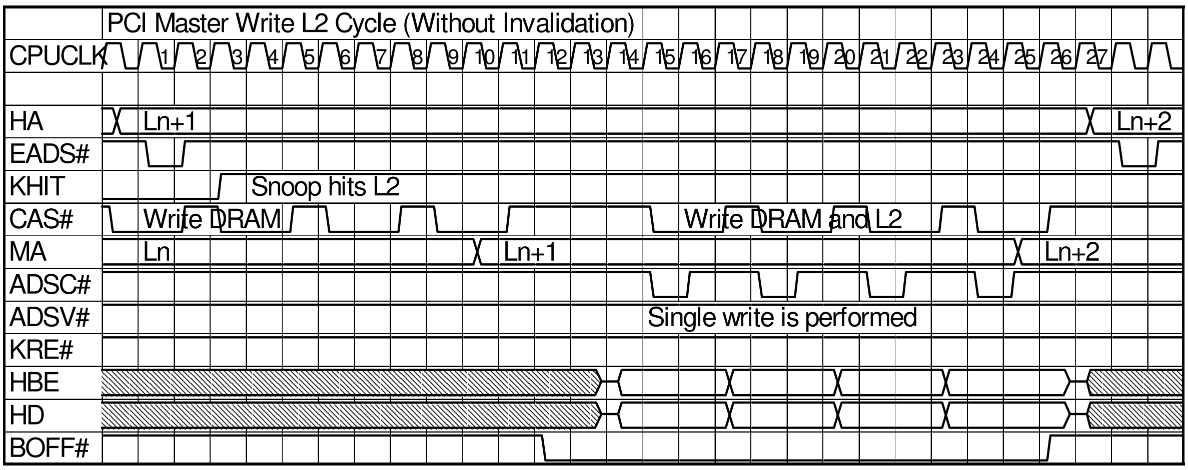
If inquiry cycle hits neither L1 nor L2 cache, SiS Chip could perform prefetching/retiring operation and inquiry cycles simultaneously.



3.4.9 PCI Master Write L2 CACHE and DRAM Cycles

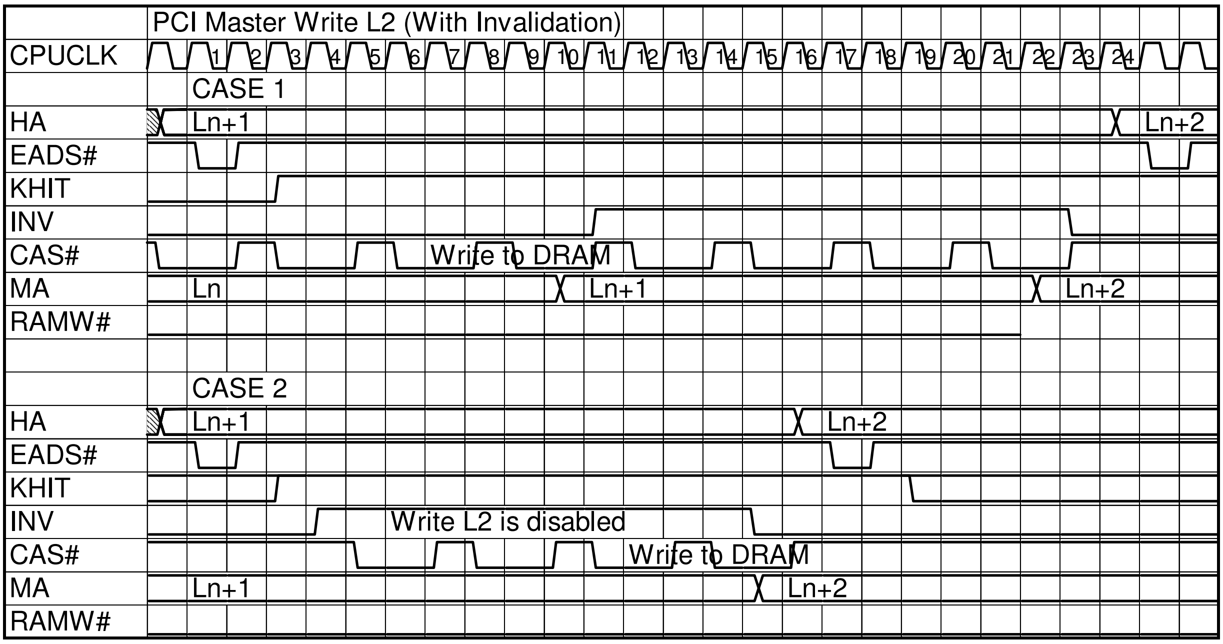
(1) Without Invalidation

For the purpose of writing L2 CACHE, PCI Slave controller (PSL) must drive the HBE[7:0]# and HD bus. Then, BOFF# is asserted to force CPU floats the host bus. And to retain the correct address on HA bus, advanced snoop is temporarily suspended.



Note: KHIT is an internal signal.

(2) With Invalidation



Note: KHIT is an internal signal.

3.4.10 PCI Arbiter

The main function of PCI arbiter takes charge of the PCI bus ownership assignment. This PCI arbiter supports at most 5 external PCI masters and 5 internal PCI masters. The arbitration operation is applied to the Host Bridge and CPU.

The arbitration scheme which we design is done at two layers. CPU has the highest priority, i.e., CPU will be the PCI bus owner if there is a request from the Host Bridge. If there is no request from the Host Bridge, rotational priority scheme will be applied to these masters.

Arbitration Algorithm

<u>**PCI Masters (Agent 0~6, SIO) Requests**</u>

Figure 3-5 shows the arbitration tree in arbiter design. Whenever a PCI cycle occurs, priority status will be changed. The initial priority for master 0-7 to own PCI bus is 4 -> 0->SIO->2->5->1->6->3->4.........

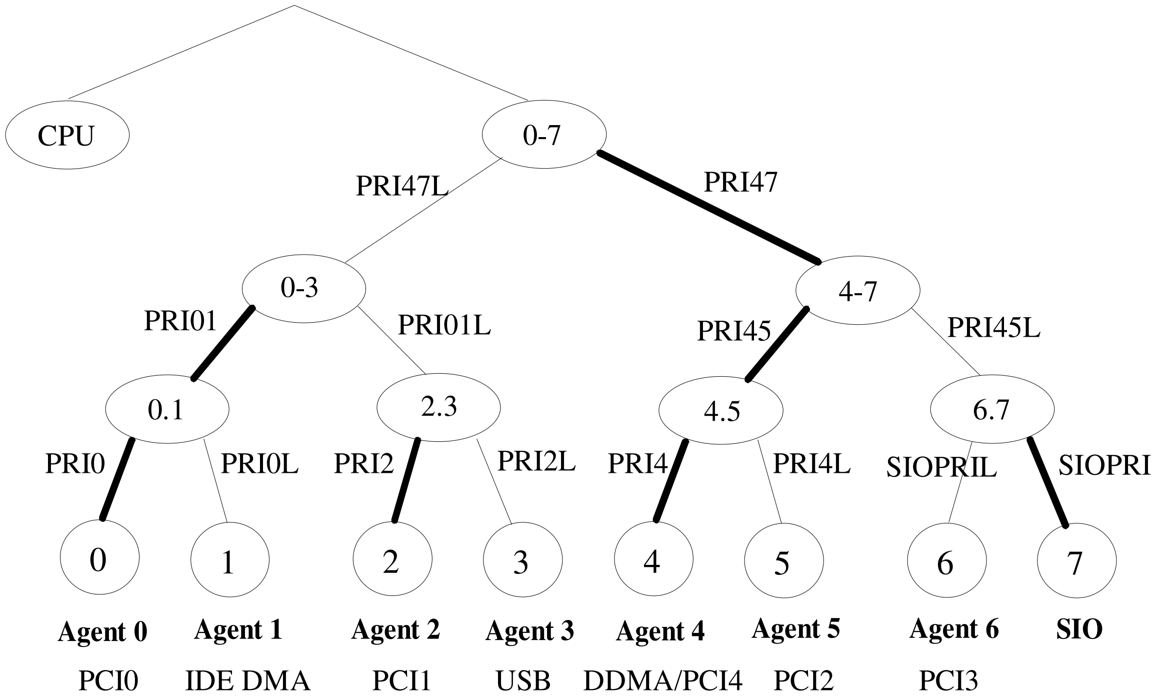


Figure 3-5 Arbitration Tree

NOTE: "SIO" means the System I/O for PCI to ISA bridge.

CPU Request

In our previous design, CPU will be constantly held if PCI masters continuously deliver requests to the arbiter. To address this problem in SiS Chip, we derived a timer-based algorithm to reserve PCI bandwidth for CPU. Three timers, PCI Grant Timer(PGT)/Master Latency Timer(MLT)/CPU Idle Timer(CIT), are included in the host bridge for this purpose.

Whenever the PCI bus is owned by any PCI device other than host bridge, PCI grant timer (PGT) starts to count. After the timer is expired, the host bridge asserts its request signal to ask for gaining the control of PCI bus. Since the host bridge has the highest priority, PCI arbiter grants the bus to the host bridge as soon as possible after it receives the request from the host bridge.

Once the host bridge get a chance to start a transaction on PCI bus, its master latency timer (MLT) begins to count. After MLT is expired, the host bridge deasserts its request signal to inform the arbiter that the host bridge no more needs the PCI bus. If there is any other PCI device that requests for the bus, arbiter grants the bus to the device and CPU is held again.

If there is no request from any PCI devices, the arbiter parks the bus on the host bridge. The ratio MLT/PGT approximately guarantees the minimum PCI bandwidth allocated to host bridge when CPU and PCI masters are contending for system resources, but it doesn't constrain CPU's highest utilization of PCI bus because of our bus parking policy.

To prevent the host bridge from capturing PCI bus too long while CPU actually has nothing to do at all, the third timer, CPU Idle Timer (CIT) is included in our design. CIT starts to count when the host bridge get a chance to start a transaction on PCI bus, but is reloaded with its initial value whenever the host bus leaves idle state. CIT actually keeps track on how long the CPU is in idle state. After CIT is expired, the host bridge deasserts its request signal just in the same manner as the case of MLT's expiration.

PGT is a 16-bit timer. MLT and CIT are both 8-bit timers. All of the initial values of the three timers are programmable and can be tuned according to the nature of the application. Although CIT & MLT are both 8-bit timers, the initial value of CIT is typically programmed much smaller than MLT.

PCI peer-to-peer access concurrent with CPU to L2/DRAM access

With this feature, a transaction initiated by a PCI master targeting a PCI target won't hold CPU. The CPU can still access L2 cache, system memory and PCI post-write buffers when PCI peer-to-peer activities are undergoing. With the enlarged 8 Dword deep PCI post-write buffers, it takes longer for CPU to halt while PCI peer-to-peer accesses are taking place.

Arbitration Parking

When no agent is currently using or requesting the bus, the arbiter will grant the bus ownership to the arbitration controller of SiS Chip.

CPU to PCI Bridge

The CPU to PCI bridge forwards the CPU cycles not targeting the local memory to the PCI bus, In the case of a 64-bit CPU request or a misaligned 32-bit CPU request, the bridge takes the duty of read assembly and write disassembly control, An 8 level post-write buffer is implemented to improve the performance of CPU to PCI memory write and CPU to IDE port write. Except for on-board memory write cycles, and any non-post write cycles forwarded to the PCI bus will be suspended until the post-write buffer is empty. For memory write cycles toward PCI or I/O write cycles towards IDE data port, the address and data from host bus are pushed into the post write buffer if it is not full. The push rate for a double word is 3 CPU clocks. The pushed data are, at later time, written to the PCI bus. If the addresses of consecutive written data are in double word incremental sequence and they are targeting memory space, they will be transferred to the PCI bus in a burst manner.

The bridge provides a mechanism for converting standard I/O cycles on the CPU bus to configuration cycles on the PCI bus. Configuration Mechanism#1 in PCI Specification is used to do the cycle conversion.

The bridge always intercepts the first interrupt acknowledge cycle from CPU bus, and forwards the second interrupt acknowledge cycle onto the PCI bus.

The bridge is designed to be able to handle asynchronous clock relationship between CPU and PCI. However, in order to enhance the performance of the bridge when PCI clock is lagging CPU clock by 2~4 ns, an optional synchronous mode is provided. The synchronous mode can averagely save two extra CPU clocks for a single non-post cycle.

3.5 Power Management Unit (PMU)

The function of PMU is to provide power management functions for the system to meet Green PC requirement. The main methodology of PMU is to generate SMI#, STPCLK# and FLUSH# signals to CPU for different situations. The PMU unit includes 3 major sub-blocks, Legacy PMU, APC and ACPI. Legacy PMU is the traditional PMU block and may be replaced by ACPI. APC(Auto Power Control) block is mainly responsible for the power supply control. For more information on APCI please refer to Section 0
Advanced Configuration and Power Interface (ACPI) on page 43.The following sections will introduce the legacy PMU function.

3.5.1 Legacy PMU Block Diagram

Legacy PMU block can be divided into several sub-blocks as shown in Figure 3-6 Legacy PMU Block Diagram. Events Catching Logic is responsible for recording the events that request SMI#. Time Base generation logic is to generate the clock for timer. Timers are responsible for timeout reporting. SMI generation Logic is for SMI# generation. STPCLK# generation Logic is for STPCLK# generation. FLUSH# generation Logic is for FLUSH# generation.

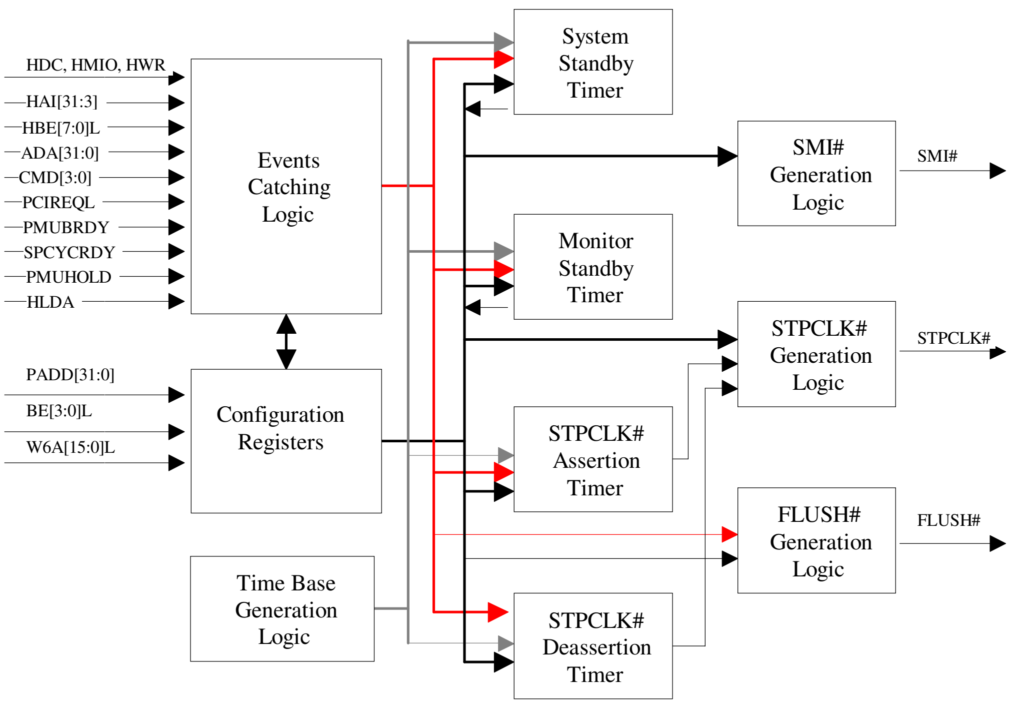


Figure 3-6 Legacy PMU Block Diagram

3.5.2 Time Base Generation Logic

All the clocks used in Legacy PMU timers are derived from 14.318 MHZ clock. To support different time slots, SiS Chip uses frequency divider to obtain the clock we require. The time slots SiS Chip support are divided in two classes. One is for monitor standby timer and the other is for system standby timer. The former includes 6.6sec, 0.84sec, 13.3ms and 1.6ms programmability in register 96h bits[7:6] of Host-to-PCI bridge configuration space while the latter includes 9 sec, 1.1 sec, 70ms and 8.85ms programmability in register 91h bits[1:0] of Host-to-PCI bridge configuration space. Besides, SiS Chip provide CPU clock for timer in test mode.

3.5.3 Timers

There are three kinds of timer defined in Legacy PMU. One is for monitor activity, another is for system activity and the other is for STPCLK# behavior generation. In order to save monitor power dissipation, we provide monitor standby timer to detect if there is any monitor-related activity. If there is any activity, monitor standby timer will be reloaded. Otherwise, monitor standby timer will continuously count down. If it count to zero, it will report timeout event. System standby timer has the same operation as monitor standby timer. STPCLK# assertion/deassertion timer is to toggle STPCLK# signal in Throttling mode.

3.5.4 Event Catching Logic

System Sleep Events

The timeout of system timer will request SMI# to enter Sleep state. If throttling mode is enabled, Legacy PMU will enter throttling mode. Otherwise, if STPCLK# mode is enabled, Legacy PMU will enter sleep mode. If both modes are disabled, Legacy PMU remains wakeup.

System Wakeup events

The following events will wakeup system from Standby state to Normal state.
Software Wakeup
RING
IRQ 1-15, NMI
INIT
PCI or ISA master request

Monitor Timeout event

If monitor timer expired, SMI# will be generated for the request of turn-off monitor power.

Monitor Wakeup Events

The following shows the events that can wakeup monitor from Standby to Normal state.
IRQ1-15,NMI
PCI master, ISA master activity
Ring Activity

SMI Sources

The following shows the sources to generate SMI request.
System Standby SMI
System Wakeup SMI

Throttling Wakeup SMI
Monitor Standby SMI
Monitor Wakeup SMI
Ring SMI
Keyboard Port SMI
Primary Hard Disk Port SMI
Secondary Hard Disk Port SMI
Primary Serial Port SMI
Secondary Serial Port SMI
Parallel Port SMI
APM SMI
Break Switch SMI (External SMI)
10-bit Programmable Port SMI
16-bit Programmable Port SMI
IRQ SMI
USB SMI

3.5.5 Output generation Logic

SMI# generation

When there is any event to request entering sleep/throttling/wakeup state, SMI# will be issued. When SMI# is recognized by CPU, SMI routine will handle the operation of state transition.

STPCLK# generation

STPCLK# generation is initialized by SMI sub-routine by writing Reg. 93 bit 3 to ‘1’ in Host to PCI bridge configuration space. The behavior of STPCLK# depends on the configuration register setting, i.e., non-throttling or throttling.

FLUSH# generation

FLUSH# is generated for DeTurbo mode. By issuing FLUSH#, CPU will write back all modified cachelines in the data cache and invalidate both internal code and data caches. Flush Acknowledge special cycle will be driven once flush operation is completed. Hence, CPU performance can be degraded.

3.5.6 Operation of Power Management

There are three states in PMU, i.e., Wakeup state, Sleep state and Throttling state. In wakeup state, system wakes up from sleep or throttling state. In sleep state, STPCLK# will always asserted until exiting Sleep state. In throttling state, STPCLK# will be asserted and deasserted periodically.

Once CPU recognizes a STPCLK# interrupt, CPU will perform the following:

1. Wait for all instructions being executed to complete.
2. Flush the instruction pipeline of any instructions waiting to be executed.
3. Wait for all pending bus cycles to complete and EWBE# to go active.

4. Drive a special bus cycle(stop grant bus cycle) to indicate that the clock is being stopped. Stop grant bus cycle is decoded as follows: M/IO#=0, D/C#=0, W/R#=1, Address Bus=00000010H (A4=1), BE7#-BE0#=11111011, Data bus=undefined.
5. Enter low power mode.

The rising edge of STPCLK# indicates that CPU can return to program execution at the instruction following the interrupted instruction

3.5.7 Hardware Limitation

If STPCLK# is configured as throttling mode. There is a possibility for SMI to break Configuration Register Access. To elaborate, there are two steps to access Configuration registers. The first step is to program I/O port CF8h and the second one is to program I/O port CFCh. If SMI routine begins to be executed between the two steps by CPU. There is possibility for PMU to cause mal-function.
The recommended solution for this problem is to read the CF8h data before executing other code in SMI routine and write back the data to CF8h before exiting SMI routine.

3.6 PCI/ISA System I/O (PSIO)

3.6.1 Functional Description

As a PCI slave device, PSIO responds to both I/O and memory transfers. PSIO always target-terminates after the first data phase for any bursting cycle.
The PSIO is assigned as the subtractive decoder in the Bus 0 of the PCI/ISA system by accepting all accesses not positively decoded by some other agent. In reality, the PSIO only subtractively responds to low 64K I/O or low 16M memory accesses. PSIO also positively decodes I/O addresses for internal registers, and BIOS memory space by asserting DEVSEL# signal on the medium timing.
As a PCI master device, the PCI master bridge on behalf of DMA devices or ISA Master devices start to drive the AD bus, C/BE[3:0]# and PAR signals. When MEMR# or MEMW# is asserted, the PSIO will generate FRAME#, and IRDY# to PCI bus if the targeted memory is not on the ISA side. The valid address and command are driven during the address phase, and PAR signal is asserted one clock after that phase. PSIO always activated FRAME# for 2 PCLKs because it does not conduct any bursting cycle.
The ISA address decoder is used to determine the destination of ISA master devices or DMA devices. This decoder provides the following options as they are defined in registers 48h to 4Bh of PCI to ISA Bridge configuration space.
a. Memory: 0-512K
b. Memory: 512K-640K
c. Memory: 640K-768K(video buffer)
d. Memory: 768K-896K in eight 16K sections(Expansion ROM)
e. Memory: 896K-960K(lower BIOS area)
f. Memory: 1M-XM-16M within which a hole can be opened. Access to the hole is not forwarded to PCI bus.
g. Memory:>16M automatically forwards to PCI.

3.6.2 ISA Bus Controller

The SiS Chip's ISA Bus Interface accepts those cycles from PCI bus interface and then translates them onto the ISA bus. It also requests the PCI master bridge to generate PCI cycle on behalf of DMA or ISA master devices. The ISA bus interface thus contains a standard ISA Bus Controller (IBC) and a Data Buffering logic. IBC provides all the ISA control, such as ISA command generation, I/O recovery control, wait-state insertion, and data buffer steering. The PCI to/from ISA address and data bus bufferings are also all integrated in SiS Chip. The SiS Chip can directly support 4 ISA slots without external data or address buffering.

Standard ISA bus refresh is requested by Counter 1, and then performed via the IBC. IBC generates the pertinent command and refreshes address to the ISA bus. Since the ISA refresh is transparent to the PCI bus and the DMA cycle, an arbiter is employed to resolve the possible conflicts among PCI cycles, refresh cycles, and DMA cycles.

3.6.3 DMA Controller

The SiS Chip contains a seven-channel DMA controller. The channel 0 to 3 is for 8-bit DMA devices while channel 5 to 7 is for 16-bit devices. The channels can also be programmed for any of the four transfer modes, which include single, demand, block, and cascade. Except in cascade mode, each of the three active transfer modes can perform three different types of transfers, which include read, write, and verify. The address generation circuitry in SiS Chip can only support 24-bit address for DMA devices.

Distributed DMA

Distributed DMA allows the individual DMA channels to be separated into different physical devices on the PCI bus. In distributed DMA, the DMA Master contains the addresses that were occupied by the traditional ISA DMA Controller(8237). This device will respond to any system read or write to the traditional ISA DMA address locations so the software will continue to think it is communicating with a standard DMA controller. The SiS Chip is the DMA Master and the protocol is as follows:

1). When the CPU bridge attempts to read/write a legacy DMA register, a PCI I/O cycle will be initiated on the PCI bus with a legacy DMA address. The SiS Chip will take control of this cycle by driving DEVSEL# active, driving the internal Request(requesting the PCI bus), and issuing a PCI retry to terminate this cycle.

2). When granted the PCI bus, the SiS Chip will run up to PCI I/O byte read/writes. The specific I/O addresses for each legacy DMA address are remappable. The purpose of these read/writes is to return/send the individual channel read/write information. DMA Slave devices must only respond to the slave address assigned to them and not any legacy DMA address.

3). At the end of the last read/write the SiS Chip will set an internal flag indicating completion and will drive its internal Request inactive (relinquishing the PCI bus) and wait for the retried PCI I/O read/write from the CPU bridge.

4). The PCI I/O read/write will be retried. If it was a read, theSiS Chip will return the data. If it was a write, the SiS Chip will simply terminate the cycle. Then the SiS Chip will reset the internal flag.

3.6.4 Interrupt Controller

The SiS Chip provides an ISA compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two controllers are cascaded so that 14 external and two internal interrupts are supported. The master interrupt controller provides IRQ<7:0> and the slave one provides IRQ<15:8>. The two internal interrupt are used for internal functions only and are not available externally. IRQ2 is used to cascade the two controllers together and IRQ0 is used as a system timer interrupt and is tied to interval Counter 0. The remaining 14 interrupt lines are available for external system interrupts.

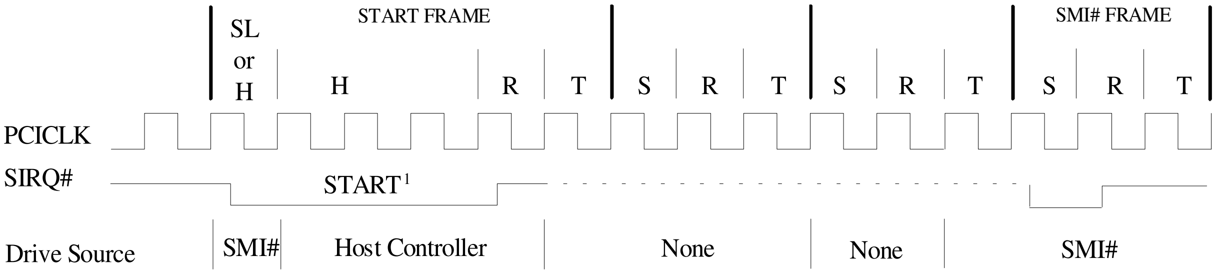
Priority	**Label**	**Controller**	**Typical Interrupt Source**
1	IRQ0	1	Timer/Counter 0 Out
2	IRQ1	1	Keyboard
3-10	IRQ2	1	Interrupt from Controller 2
3	IRQ8#	2	Real Time Clock
4	IRQ9	2	Expansion bus pin B04
5	IRQ10	2	Expansion bus pin D03
6	IRQ11	2	Expansion bus pin D04
7	IRQ12	2	Expansion bus pin D05
8	IRQ13	2	Coprocessor Error Ferr#
9	IRQ14	2	Fixed Disk Drive Controller Expansion bus pin D07
10	IRQ15	2	Expansion bus pin D06
11	IRQ3	1	Serial port 2, Expansion Bus B25
12	IRQ4	1	Serial port 1, Expansion Bus B24
13	IRQ5	1	Parallel Port 2, Expansion Bus B23
14	IRQ6	1	Diskette Controller, Expansion Bus B22
15	IRQ7	1	Parallel Port1, Expansion Bus B21

In addition to the ISA features, the ability to do interrupt sharing is included. Two registers located at 4D0h and 4D1h are defined to allow edge or level sense selection to be made on an individual channel by channel basis instead of on a complete bank of channels. Note that the default of IRQ0, IRQ1, IRQ2, IRQ8# and IRQ13 is edge sensitive, and can not be programmed. Also, each PCI Interrupt(INTx#) can be programmed independently to route to one of the eleven ISA compatible interrupts(IRQ<7:3>, IRQ<15:14>, and IRQ<12:9>) through PCI to ISA bridge configuration registers 41h to 44h.

Serial IRQ

The Serial IRQ provides a mechanism for communicating IRQ status between ISA legacy components, PCI components, and PCI system controllers. A serial interface is specified that provides a means for transferring IRQ and/or other information from one system component to a system host controller. A transfer, called an serial IRQ cycle, consists of three frame types: one Start Frame, several IRQ/Data Frames, and one Stop Frame. This protocol uses the PCI clock as its clock source and conforms to the PCI bus electrical specification.

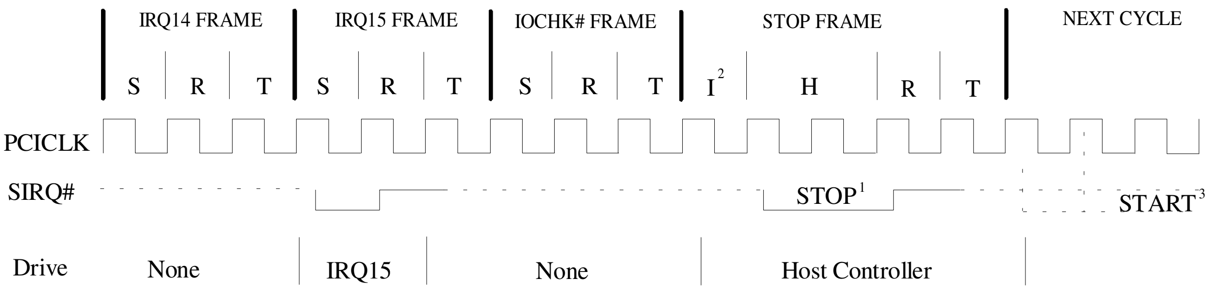
Timing Diagrams For Serial IRQ Cycle.



H=Host Control SL=Slave Control R=Recovery T=Turn-around S=Sample

1. Start Frame pulse can be 4-8 clocks wide.

Figure 3-7 Start Frame timing with source sampled a low pulse on SMI#



H=Host Control R=Recovery T=Turn-around S=Sample I=Idle

1. Stop pulse is 2 clocks wide for Quiet mode, 3 clocks wide for Continuous mode.
2. There may be none, one or more Idel states during the Stop Frame.
3. The next Serial IRQ cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

Figure 3-8 Stop Frame Timing with Host using 17 SIRQ# sampling period

Serial IRQ Cycle Control

There are two modes of operations for the serial IRQ start frame.

a) Quiet(Active) Mode: Any device may initiate a Start Frame by driving SIRQ# low for one clock, while SIRQ# is Idle. After driving low one clock the SIRQ# must immediately be tri-stated without at any time driving high. A Start Frame may not be initiated while the SIRQ# is Active. The SIRQ# is Idle between Stop and Start Frames. This mode of operation allows the SIRQ# to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated the SiS Chip will take over driving the SIRQ# low in the next clock and will continue driving the SIRQ# low for a programmable period of three to seven clocks more. This makes a total low pulse width of four to eight clocks. Finally, SiS Chip will drive the SIRQ# back high for one clock, then tri-state.

Any serial IRQ device which detects any transition on an SIRQ# line for which it is responsible must initiate a Start Frame in order to update the SiS Chip unless the SIRQ# is already in a serial IRQ cycle and the IRQ/Data transition can be delivered in the serial IRQ cycle.

b) Continuous(Idle) Mode: Only the SiS Chip can initiate a Start Frame to update SIRQ# line information. All other serial IRQ agents become passive may not initiate a Start Frame. SIRQ# will be driven low for four to eight clocks by the SiS Chip . This mode has two functions. It can be used to stop or idle the SIRQ# or the SiS Chip can operate SIRQ# in a continuous mode by initiating a Start Frame at the end of every Stop Frame. A serial IRQ mode transition can only occurs during the Stop Frame. Upon reset, the Serial IRQ bus is default to Continuous mode, therefore only the SiS Chip can initiate the first Start Frame. Slave must continuously sample the Stop Frames pulse width to determine the next serial IRQ cycle's mode.

IRQ/Data Frame

Once a Start Frame has been initiated, all serial IRQ devices must detect for the rising edges of the Start pulse and start counting IRQ/Data Frames from there. There are three clock phases for each IRQ/Data Frame:Sample phase, Recovery Phase, and Turn-around phase. During the Sample phase the serial IRQ device must drive the SIRQ# low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, SIRQ# must be left tri-stated. During the Recovery phase , a serial IRQ device will drive SIRQ# back high if it has driven the SIRQ# low in the previous clock. During the Turn-around phase all serial IRQ devices must be tri-stated. All serial IRQ devices will drive SIRQ# low at the appropriate sample point regardless of which device initiated the sample activity, if its associated IRQ/Data line is low.

The Sample phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one.(e.g. the IRQ5 sample clock is the sixth IRQ/Data frame, (6x3)-1=17th clock after the rising edge of the Start Pulse).

Serial IRQ Sampling Periods		
IRQ/Data Frame	**Signal Sampled**	**# of clocks past Start**

Serial IRQ Sampling Periods		
IRQ/Data Frame	**Signal Sampled**	**# of clocks past Start**
2:1	Reserved	2
3	SMI#	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8#	26
10	IRQ9	29
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47
17	IOCHCK#	50
21:18	Reserved	53
32:22	Unassigned	95

At the end of each Sample phase, the SiS Chip will sample the state the SIRQ# line and replicate the status the original IRQ/Data line at the input to the 8259s Interrupt Controller.

Stop Cycle Control

Once all IRQ/Data frames have completed, the SiS Chip will terminate SIRQ# activity by driving Stop cycle. Only the SiS Chip can initiate the stop frame. A Stop Frame is indicated by the SiS Chip driving SIRQ# low for two clocks(Quiet Mode) or 3 clocks(Continuous Mode), then back high for one clock. In the Quiet mode, any serial IRQ device may initiate a Start frame in the third clock or more after the rising edge of the Stop frame pulse. In the Continuous mode, only the SiS Chip may initiate a Start frame in the third clock or more after the rising edge of the Stop frame pulse.

EOI/ISR Latency

When a legacy interrupt is deasserted, it will start a serial interrupt frame. An EOI can occur after the legacy interrupt is deasserted, however, the 8259 may not detect the deasserted interrupt because it is still being serialized. This could cause the 8259 to generate interrupt as soon as the EOI is received. By delaying EOIs and ISR read to the 8259 in order to ensure that these latency issues are well covered. Note that, EOI indicates the End of Interrupt and ISR indicates the Interrupt Service Routine.

3.6.5 Timer/Counter

The SiS Chip contains 3 channel counter/timer that is equivalent to those found in the 82C54 programmable interval timer. The counters use a division of 14.318MHz OSC input as the clock source. The outputs of the timers are directed to key system functions. Counter 0 is connected to the interrupt controller IRQ0 and provides a system timer interrupt for a time-of-day, diskette time-out, or the other system timing function. Counter 1 generates a refresh-request signal and Counter 2 generates the tone for the speaker.

3.6.6 Integrated Real Time Clock (RTC)

Real Time Clock Module

The Real Time Clock module in the SiS Chip contain the industrial standard Real Time Clock which is compatible to MC146818, and the Auto Power Control circuitry mainly to support the ACPI power control functions. The Real Time Clock part provides a time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt, 114 Bytes of standard CMOS SRAM, and 128 Byes of extended CMOS SRAM. The Auto Power Control part provides the software/hardware power up/down functions. Figure 3-9 shows the block diagram of the RTC module.

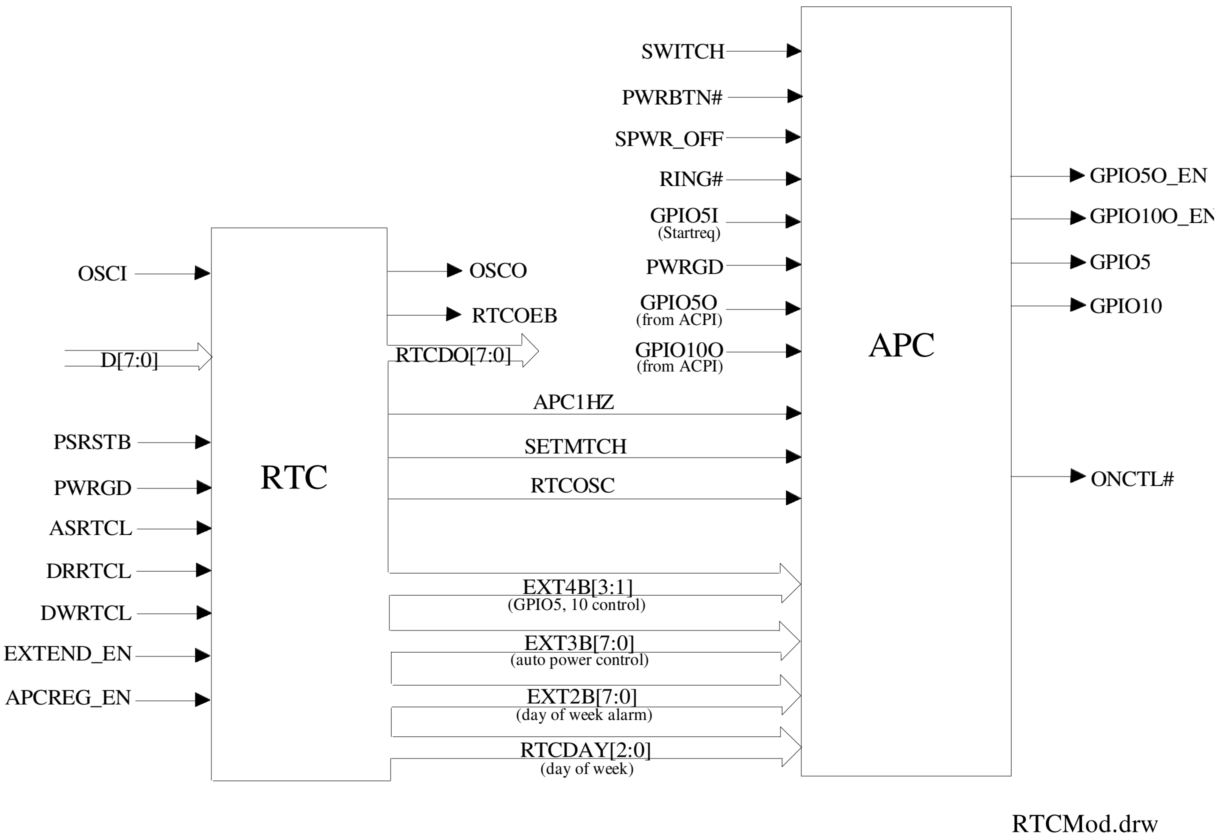


Figure 3-9 RTC module Block Diagram

RTC Registers & RAM

Three separate RTC registers & RAMs are provided in the SiS Chip. One is called the Standard Bank, another is the Extended Bank, and the other is the APC (Auto Power Control) registers. All of these registers are referenced through the same index, and data port, ie. port 70H and 71H, respectively. The access control with which that the three portions of registers are appropriately addressed are stored in PCI-ISA:45h[3] (EXTEND_EN bit) and PCI-ISA:44h[4] (APCREG_EN bit). EXTEND_EN bit enables the access of the Standard Bank while it is 0. The EXTEND_EN bit must be programmed to 1 to read/write the Extended Bank. The APCREG_EN bit toggle the 70H/71H access between the RTC registers and APC registers. The 70H/71H port will access the RTC Standard/Extended Bank if APCREG_EN bit is 0. The 70H/71H port will access the APC register if APCREG_EN bit is 1. Figure 3-10 shows the address map of the Standard Bank. In the Standard Bank, the lower 10 bytes contain the time, calendar, and alarm data. The registers Ah,Bh,Ch, and Dh contain the control and status bytes. The last two bytes (7Eh, 7Fh) are the Day of Month Alarm byte and the Month Alarm byte which are the extended alarm features requested by the ACPI. The Day of the Month Alarm selects the day within the month to generate a RTC alarm while the Month Alarm selects the month within the year to generate a RTC alarm. The remain 112 bytes in the Standard Bank are the general purpose RAM bytes. In the Extended Bank, another 128 bytes are also provided for the general purpose usage.

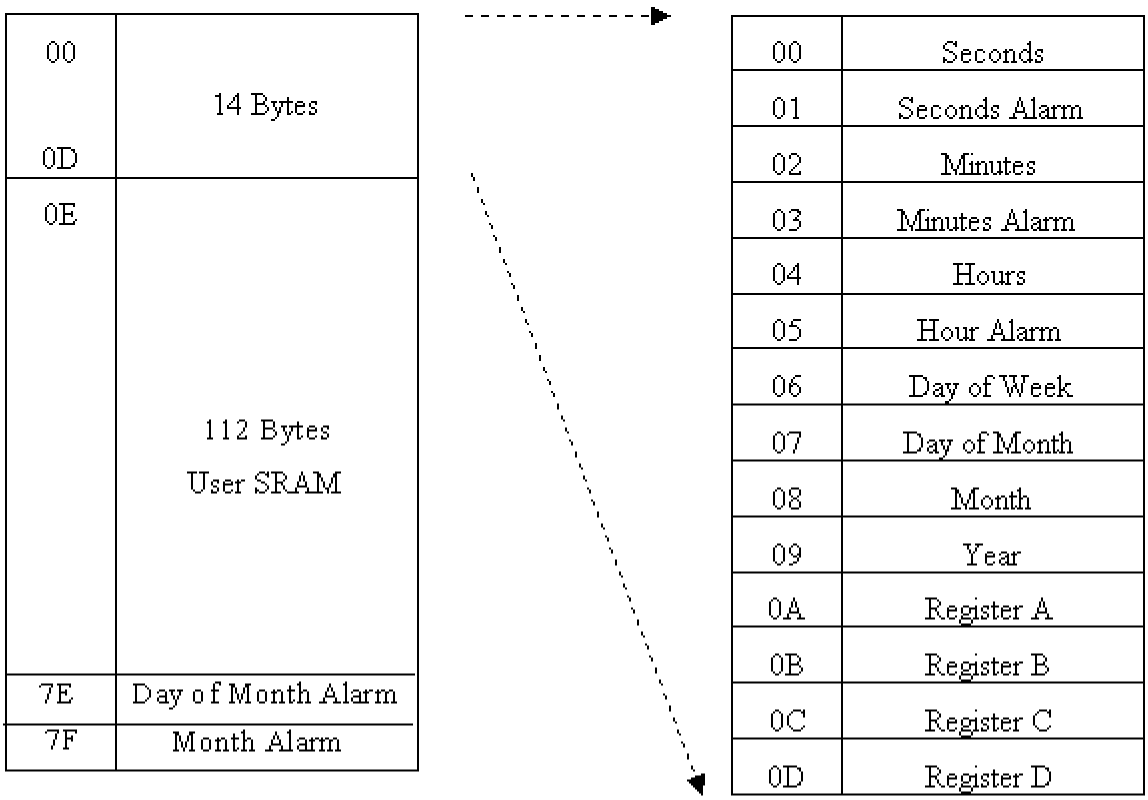


Figure 3-10 Address Map of the Standard Bank

The APC registers are provided to support the Auto Power Control Function. The register 02H defines the “Day of the Week” Alarm byte which can select the day within a week to generate a RTC alarm. The 03H, 04H registers contain the control information for the auto power control functions. Figure 3-11 shows the address map of the APC registers.

Address	Register Name
00h	Reserved
01h	Reserved
02h	Day of Week Alarm
03h	Auto Power Control Register I
04h	Auto Power Control Register II

Figure 3-11 Address Map of the APC Control Register

RTC Update Cycle

The primary function of the the update cycle is to increment the seconds bytes, the minutes bytes and so forth through to the year of the century byte. The update cycle also compares each alarm byte in the corresponding time byte and issues an alarm if a match or if a “Don’t care” code(11XXXXXX) is present in this time byte. Figure 3-12 shows the block diagram of the RTC.

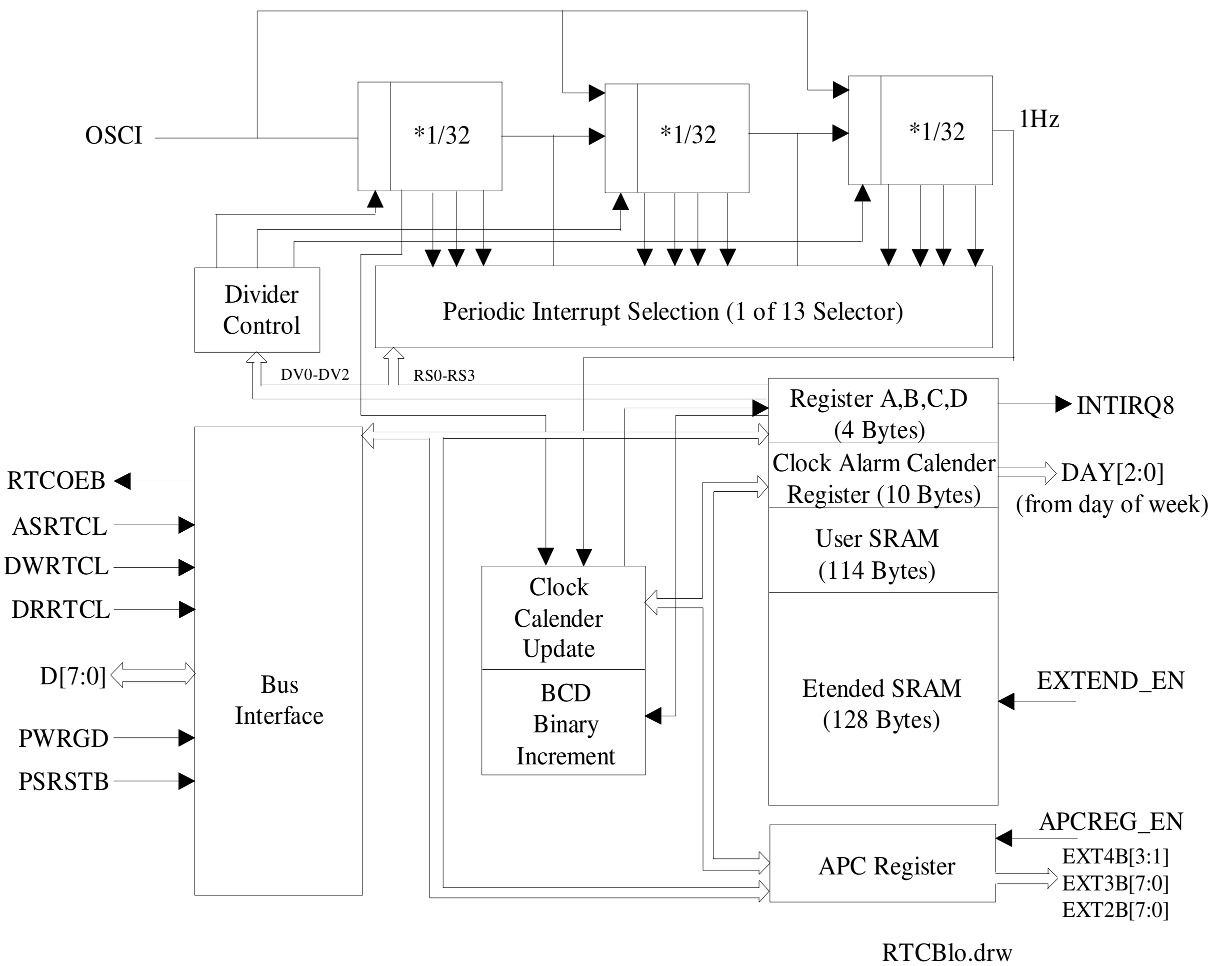


Figure 3-12 Block Diagram of RTC

RTC External Connection Requirement

The SiS Chip‘s RTC is powered by the RTCVDD, & RTCVSS. In reality, not only the internal circuitry of the RTC, the pins associated with the RTC module are also powered by the specific power planes. They are ONCTL#, PWRGD, PSRSTB#, SWITCH, PWRBT#, RING, OSCI, OSCO, GPIO5, and GPIO10.

SiS Chip is designed to support the 3V output with 5V input tolerant I/O buffers. The 5V tolerant capability is achieved by two 5V power pins(VCC5) sustaining the I/O associated well. The two pins must be connected to VCC5 if 5V input tolerance is required. The Ball NO. of these two pins are AF15, AB2 in SiS5581 and D15, H2 in SiS5582.

Please note that the RTC-related ten pins have no 5V tolerance since the associated well for the RTC is powered by the RTCVDD. The voltage level of RTCVDD is not allowed to be higher than 3.3V since SiS Chip employs the 3.3V process. Thus, DO NOT POWER THE RTCVDD HIGHER THAN 3.3V. Please ensure the ten pins related to the RTC follows this requirment. For instance, a voltage divider is requiried to clamp the PWRGD from the power supply to around 3.3V. These ten pins are : ONCTL#, RING, PWRGD, PSRSTB#, OSCI, OSCO, SWITCH, PWRBT#, GPIO5, GPIO10.

SiS Chip contains 3 wells if categorized by the driving power:
1) the internal circuitry excluding the RTC (powered by 3.3V OVDD)
2) the RTC(powered by RTCVDD), and
3) the I/O buffers(powered by 5V VCC5).

3.6.7 Auto Power Controller

An ATX power supply and the integrated RTC are needed at the same time to make the Auto Power Control function work. ATX power supply has a control signal ONCTL# and two set of VCC named VCC5 and AUX5V. When power is applied, then AUX5V exists but VCC5 does not until ONCTL# goes low. APC controls the signal ONCTL# to turn on or turn off VCC5 of ATX power supply.

Auto Power Control

Let us take the following power up/down sequence as an example to illustrate the APC functions. Please also refer to Figure 3-13 showing the typical timing sequence of the power control related pins. Also, Figure 3-14 outlines the main APC functions.

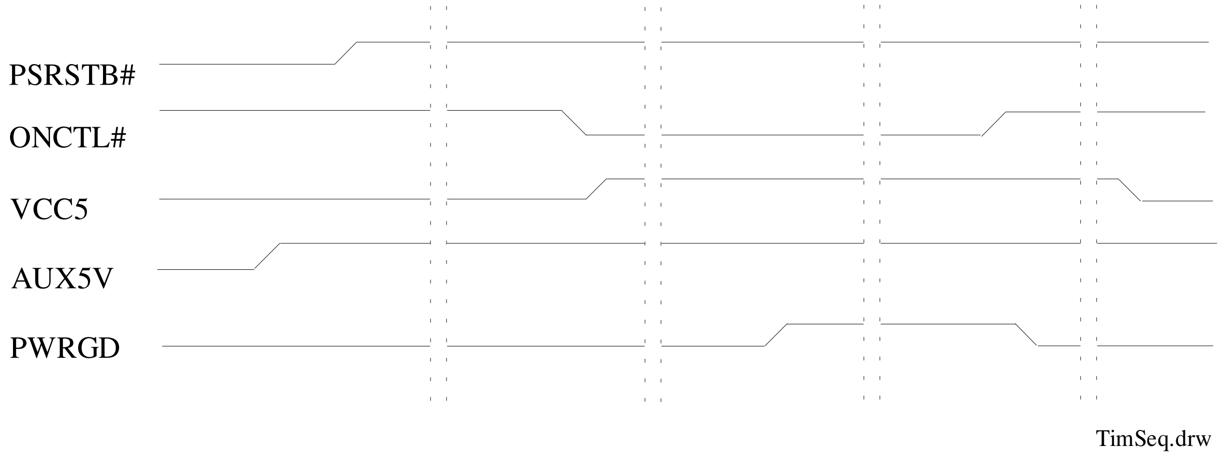


Figure 3-13 Typical Timing Sequence on the Power Control Related Signals

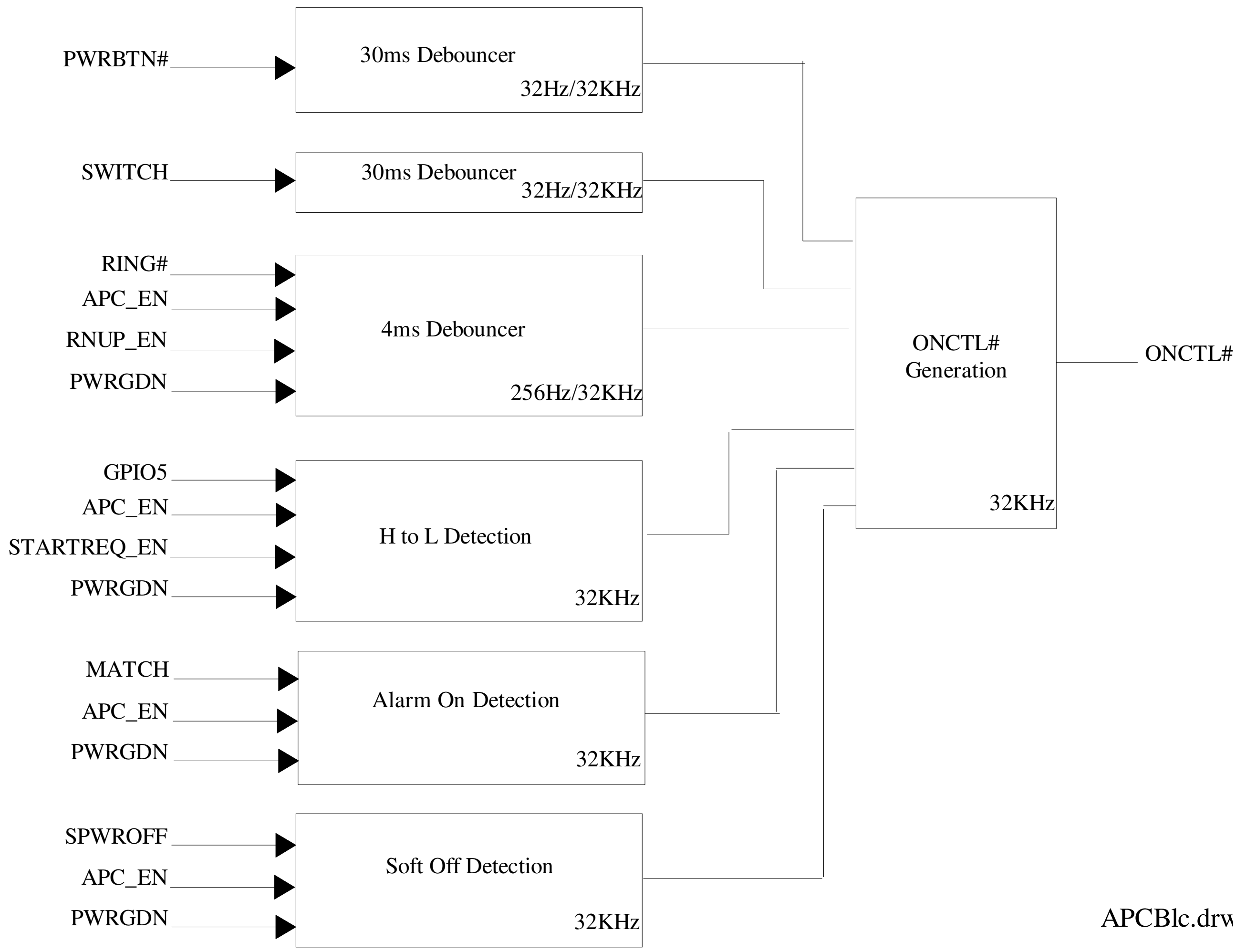


Figure 3-14 APC Block Diagram

(1) The PSRSTB# is used to determine whether the system is using internal RTC(PSRSTB# is high) or external RTC(PSRSTB# is low).The ONCTL# is set to high as long as the PSRSTB# is low. When the PSRSTB# is high and the power up events are asserted, the ONCTL# will control ATX power supply which can provide VCC5 for the motherboard.

Since the RTC must continue to count the time when the system power is removed, a conversion from the system power to an alternate power supply, usually a battery must be made. In a system powered by the ATX power supply, it is recommended to design the power conversion circuitry powered by both the AUX5V and battery. Please refer to "Application Circuit" for more details. In terms of this application, the PSRSTB# is low only when both the battery and the AUX5V is low. That is, PSRSTB# is low when the battery happened to be exhausted and the power supply is not plugged yet. Most of the cases in the application, the PSRSTB# is first restored to high by AUX5V from ATX power supply. As long as the PSRSTB# is high, the power up events can be recognized and results in the assertion of the ONCTL# to have the ATX power supply provide VCC5 for the system. It is now obvious why the conversion circuitry should use the AUX5V or battery for the power source. This ensures that the APC circuit block can have power to work and can sense the "Power Up" Request Events to wake up PC board's power from ATX power supply. In other words, RTC and APC controller must be powered by AUX5V/battery through RTCVDD, and PSRSTB# signal must be high, so that Power Up Request Events can wake up system power.

(2) During the power down period, the following events can power up the PC main board by the assertion of ONCTL#. They are Switch On event (via SWITCH), Power Button On event (via PWRBT#), Ring Up event (via RING), Start Request event (via GPIO5), and Alarm On event (via IRQ8#). Following is the detail description of these events:

1. Switch On Event:
   A schmitt trigger input buffer and a debounce protection of at least 30ms is associated with the SWITCH pin. When PWRGD is low, an active low on the SWITCH lasting for more than 30ms indicates a request from Switch On Request event, then the ONCTL# signal will be activated to low in order to power on the VCC5 of ATX power supply.

2. Power Button On Event:
   In addition to SWITCH, SiS Chip provides another power button control pin which is PWRBT#. While PWRGD is low, a high to low transition with the active low logic lasting for more than 30ms indicates the Power On Request Event which eventually activates the ONCTL#. While PWRGD is high, the assertion of PWRBT# less than 4 seconds results in a SCI/SMI event, and the assertion of PWRBT# more than 4 seconds will turn off the system. The ACPI module will take the appropriate action.

3. Ring Up Event:
   The function is enabled by setting APC_EN bit and RNUP_EN bit which are located in the bit 6 and bit 5 of the Auto Power Control Register I. Also, the active high/low logic of the RING can be programmed through bit 4 of this register. In reality, while PWRGD is low, the detection of RING pulse lasting for more than 4ms would activate the ONCTL#. While PWRGD is high, the detection of RING pulse would wake up the Legacy PMU or ACPI to put the system back to the higher activity mode. Please refer to ACPI section for more detail on this aspect.

4. Start Request Event (GPIO5 Event):
   While the power is removed, a high to low transition on the GPIO5 also indicates a power up request event which activates ONCTL#. Note that no debouncer is associated with this

signal. This function is enabled by setting APC_EN bit, and STARTREQ_EN bit which are registered in the bit 6, and bit 3 of Auto Power Control Register I, respectively.

5. Alarm On Event:

When the time value of RTC matches the corresponding time bytes which were set in advance, the RTC would send an “alarm” to the APC module to activate the ONCTL#. The SiS Chip supports the 24 hour alarm to a month alarm. Beside the ACPI Extended Alarm function, SiS chip also supplies an additional alarm function called the “Day of Week Alarm”. Following are the detail description of these two alarm functions:

(a) ACPI Extended Alarm Function: The ACPI Extended Alarm Function can set the alarm by the day within a month or the month within a year. The Day of the Month and the Month Alarms bytes are stored in the register 7Eh and 7Fh of RTC Standard Bank. To enable this function the APC_EN bit in APC Register I must be enabled.

(b) Day of Month Alarm Function: The Day of Month Alarm Function can set the alarm by the day within a week. With this additional feature, the SiS Chip allows the system to be alarmed up on, say, each Monday 08:00. The Day of the Week Alarm byte is located in the register 2h of the APC registers. Note that this feature is enabled when DayWeekAlarm_EN bit located in the bit 0 of the Day of the Week Alarm Register, and the APC_EN bit are both set. ACPI extended alarm function is the default alarm mode once the APC_EN bit is set. However, the ACPI extended alarm mode is replaced by the Day of the Week alarm mode once the DayWeekAlarm_EN bit is set.

(3) While in the power up state, the following events can power down the PC main board by the deassertion of ONCTL#. They are Switch Off event (via SWITCH), Software Power Off Request Event (via register bit of SiS Chip), and Power-Button-Over-Ride (via PWRBT#). Following is the detail description of these events:

1.Switch Off Event:

A schmitt trigger input buffer and a debounce protection of at least 30ms is associated with the SWITCH pin. When PWRGD is high, an active low pulse on the SWITCH signal for more than 30ms indicates a request from Switch Off Request Event, then the ONCTL# will be deasserted to high in order to power off the VCC5 of ATX power supply. An Switch on Request event asserts the ONCTL# while An Switch off request event deasserts ONCTL#.

2. Software Power Off Request Event:

SiS Chip also provide two software means to control the ONCTL# signal. One is to programming a “1” to SLP_EN bit (ACPI:04h[6]) and “100” (S5 state) to SLP_TYP bit (ACPI:04h[4:2]), the other is to program a “1” to APC_EN bit (APC: I[6]) and Power Off System Control bit (PCI-ISA:69h[4]). System will be powered off if any of these two settings was set. In other words, ONCTL# will be deasserted to high.

3. Power-Button-Over-Ride Event:

If push the Power Button over 4 seconds, a Power-Button-Over-Ride event happened. The SiS chip will deassert the ONCTL# if Power-Button-Over-Ride event happened, and turn off the system power eventually. Please note that this is a hardware implemented function, and do not need the special support from the BIOS.

(4) SiS chip supplies a special pin, GPIO10/ACPILED, to facilitate the Green PC denote-LED design. If this pin is in its output mode (APC II[3] = 0), and pin definition is ACPILED (APC II[1] = 1), and 1 Hz Function Support is enabled (APC II[0] = 1), and ACPI function is enabled (PCI to ISA:40h[7] = 1), then SiS chip will send out an 1 Hz signal on this pin to turn on/off the external LED whenever GPIO10 pin status register (ACPI:18h[10]) is set to 1

3.6.8 Advanced Configuration and Power Interface (ACPI)

Advanced Configuration and Power Interface (ACPI) is PC 97 specification. ACPI extends the portability for different platforms by moving the power management function into the OS. ACPI also releases the restriction of ROM BIOS capacity on the complexity of the advanced power management functions. The power management events of ACPI are initiated by the assertion of System Control Interrupt (SCI). System uses SCI to send ACPI-relevant notifications to the host OS, and then OS executes the specific service sub-routines according to which enable bit and status bit were set.

The ACPI architecture in SiS chip consists of the control logic, the SCI/SMI# generating logic, the wake up logic, and the configuration registers to ensure fluent communication with the ACPI-compliant OS. The control logic is used by OS to alternate some state transitions. The SCI/SMI# generating logic sensing the external environmental change or request is used to notify the OS to take some actions. The wakeup logic will sequence the system from the sleeping state (S1) to the G0 working state. Following table summaries the ACPI configuration space supported by SiS chip:

Name	Address	Register Description
PM1a_STS	00h	Power Management 1a Status Register
PM1b_STS	01h	Power Management 1b Status Register
PM1a_EN	02h	Power Management 1a Enable Register
PM1b_EN	03h	Power Management 1b Enable Register
PM1a_CNT	04h	Power Management 1a Control Register
PM1b_CNT	05h	Power Management 1b Control Register
PM_TMR	08h	Power Management Timer
P_CNT	0Ch	Processor Control
P_LVL2	10h	Processor LVL2 Register
P_LVL3	11h	Processor LVL3 register
PM2_CNT	12h	Power Management 2 Control
GP_TMR	13h	General Purpose Timer Register
GPIO_STS1	14h	Status register 1 for general purpose I/O pins
GPIO_STS2	15h	Status register 2 for general purpose I/O pins
GPIO_EN1	16h	Enable register 1 for general purpose I/O pins
GPIO_EN2	17h	Enable register 2 for general purpose I/O pins
GPIO_CNT1	18h	Control/Read back status register 1 for general purpose I/O pins
GPIO_CNT2	19h	Control/Read back status register 1 for general purpose I/O pins
GPIO_SEL1	1Ah	Input/Output Selection register 1 for general purpose I/O pins
GPIO_SEL2	1Bh	Input/Output Selection register 2 for general purpose I/O pins
GPIO_MUX1	1Ch	GPIO Mux. function selection register 1
GPIO_MUX2	1Dh	GPIO Mux function selection register 2
GPIO_LVL1	1Eh	Polarity selection register 1 for general purpose I/O pins

GPIO_LVL2	1Fh	Polarity selection register 2 for general purpose I/O pins
SMI_CMD	20h	SMI command port
MUX_REG	24h	Mux. function selection register
AUX_STS	25h	Auxiliary Control Status register
AUX_EN	26h	Auxiliary Control Enable register
SMI_PEN	28h	Programmable SMI command port enable value
SMI_PDIS	29h	Programmable SMI command port disable value
SMI_REG	2Ah	Scratch register for SMI or ACPI use
ACPI_TST	2Bh	APCI test mode register

Control Logic

From ACPI specification, SiS chip supports the four global system states (G0-G3), and the traditional Legacy system state as it is shown in the *Figure 3-15*. The ACPI-compliant OS assumes the responsibility of sequencing the platform between these various global states. The ACPI-compliant OS is invoked by the shareable interrupt to which SCI is routed. The reroutability of SCI is through the programming of register 6Ah of the PCI to ISA configuration space. Transition of Legacy to/from ACPI is through issuing ACPI activate/deactivate command to the SMI handler by doing OUT to the SMI_CMD port with the data equal to the value defined in SMI_PEN, or SMI_PDIS registers, respectively.

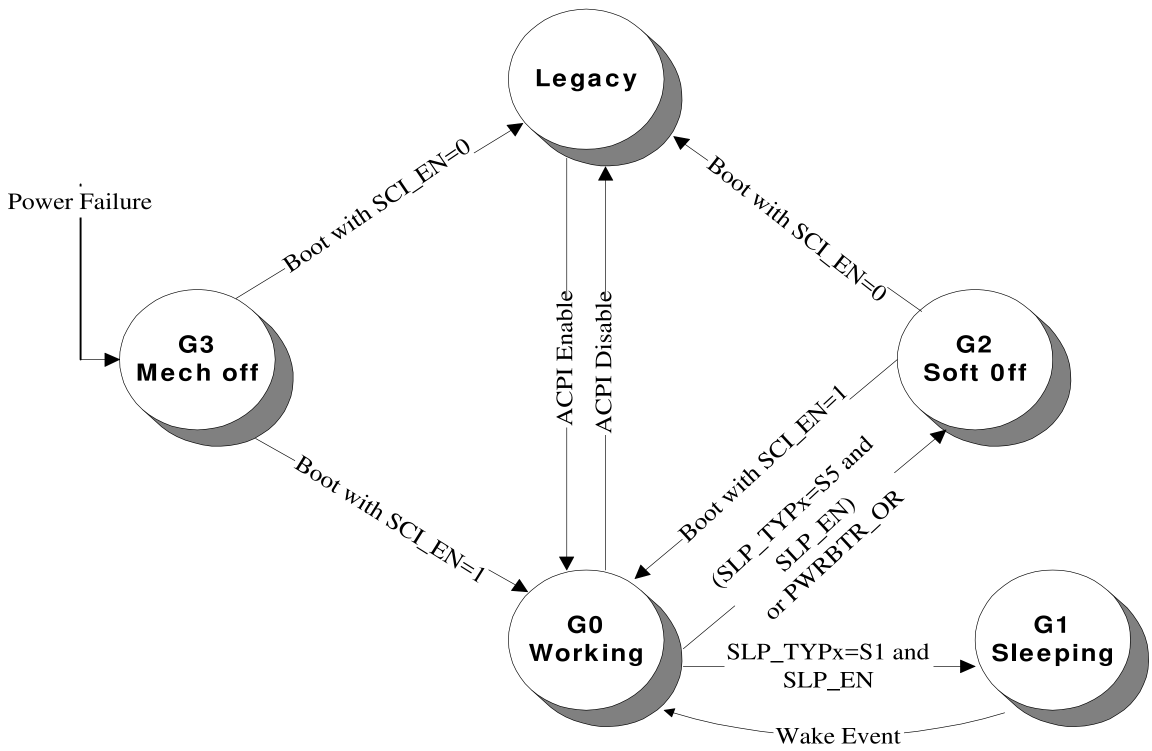


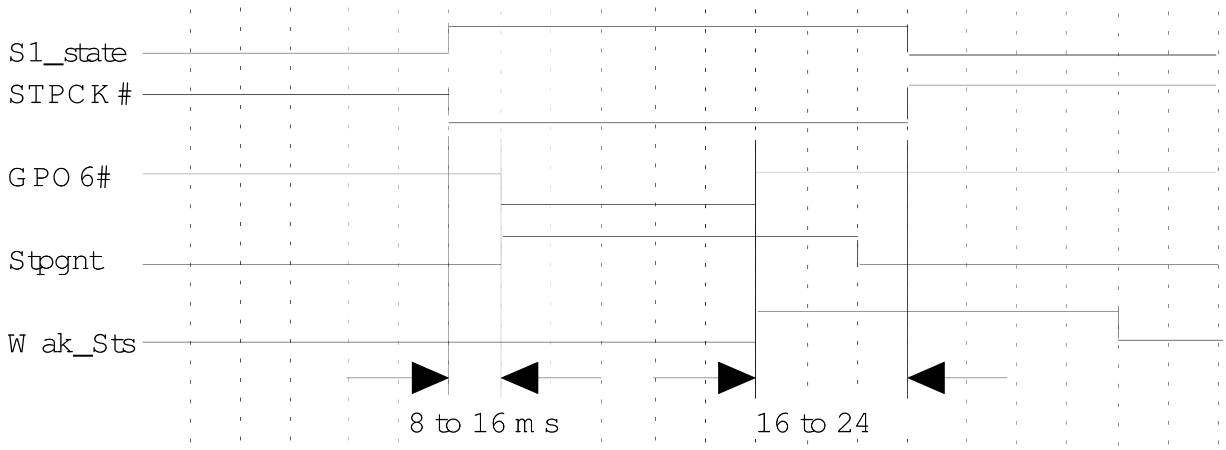
Figure 3-15 Global System State Diagram

ACPI machine would stay at G0 working state as the normal operating state in which different devices are dynamically transiting between the respective power states, and processors are dynamically transiting between their respective states (C0,C1,C2,C3) .In the

G0 state, two global state transition options are provided. One is the G1 sleeping state, and the other is the G2 soft -off state. SiS chip only supports two level of system sleeping states which are S1, and S5. In reality, the system G1 sleeping state only consists of one level (S1 only) of hierarchy in the sense of sleeping states. The OS can initiate the sleeping transition by programming the SLP_TYPx field with S1 value and then setting SLP_ENx bit high. While in the G1 state, a set of Wake_Up Events(explained in the Wakeup logic) can be enabled to transit the system state back to G0. The G2 soft-off state is an OS initiated system shutdown. The State can be initiated by programming the SLP_TYPx field with S5 value and setting the SLP_ENx bit high. Also, a hardware event which is driven by pressing the power button for more than four seconds can transit the system to the G2 state while it is in the G0 state. This hardware event is called a power button over-ride, and is mainly provided to turn off a hung system in case. Putting system in the G2 state will deassert ONCTL# eventually from hardware point of view.

G1 Sleeping System State --- S1

Transit to S1 state is by setting SLP_EN bit HIGH. All system power is still alive in this state. The external CPU clock can be disabled(or stopped) through GPO6. A set of Wake_UP Events can be enabled, before entering S1, to wake up the system back to G0 state. *Figure 3-16* is the typical sequence responding to enter S1 state. Details of the Wake_UP Events is illustrated in the Wakeup Logic.



1. Programming SLP_EN in SMI or SCI would put the PC in the system power state S1.
   Note : STPCLK# won't be activated until the SMIACT# is deasserted. That is, the STPCLK# is activated once the CPU exits the SMI handler.
2. Reserve 8 to 16ms for the emulation of Stpgnt latency.
3. While in the Stpgnt state, the wake up events would deassert GPO6.
4. Reserve 16 to 24ms from the exiting of Stpgnt to the deassertion of STPCLK#.
5. Wak_Sts could be cleared by writing a 1 to bit 15 of the base+0 register.

Figure 3-16 Typical GPO6 Timing

G2 Soft-off State:

In the G2 state, only the RTC power is alive. While in the G2 state, the SiS chip could sense the following five power up events to transit the system to the Legacy system state by asserting the ONCTL#. They are Alarm On event, Power On(PWRBTN#) event, Ring Up event, Start Request event(GPIO5), and the Switch On/Off event(SWITCH). Please see the APC portion of the RTC module for more details.

Processor Power State Control

SiS chip supports the four power states in the G0 working state. While in the C0 state, it provides programmable Throttling function to place the processor executing at a designated performance level relative to its max. performance. This can be achieved by programming the THTL_DUTY field (ACPI:0Ch[3:1]) with desired value, and setting THT_EN bit (ACPI:0Ch[4]) HIGH. The C1 state is supported through the HLT instruction. As an instance, the execution of a HALT instruction will cause the Pentium CPU to automatically enter the Auto HALT Power down state where Icc of the processor will be -20% of the Icc in the Normal state. In this state, the CPU will transit to the Normal state upon the occurrence of INTR, NMI, SMI#, RESET, or INIT. The Pentium won't recognize AHOLD, BOFF#, and EADS# for cache invalidation or writebacks. That is, the system is no longer able to allow bus master snooping, or memory access. As such, C2 low power state provides an alternative not to block bus master streaming while the CPU is put into the low power state.

In the C2 power state, SiS chip places the processor into the low power state by keeping STPCLK# low as long as no interrupt requests occur. Entering C2 state is through reading P_LVL2 register (ACPI:10h) as it is defined in the ACPI specification. Exiting C2 is effective when Wakeup1 is asserted (Wakeup1 is an internal signal, and will be asserted when any of the enabled interrupt(IRQ15-3, IRQ1,and NMI),GPCS0, or the GPCS1 is asserted.). Register 74, and 75 in the PCI to ISA configuration space defines which interrupt requests among the IRQ15-3,1, and NMI. are considered(or enabled) to generate Wakeup1. Bit 1 of register 63h, and bit 1 of register 64h enables/disables GPCS0, and GPCS1, respectively to be included in the generation of Wakeup1. The IRQ0 interrupt request is also allowed, if IRQ0SEN (ACPI:0Ch[6]) is set, to deassert STPCLK# for about 128us when in C2 or C3 state to refresh the system timer.

As a more rigid or flexible alternative to the handling of bus master in the CPU low power state, SiS chip supports the C3 power state by also keeping STPCLK# low, which can be entered by reading P_LVL3 register (ACPI:11h). The main difference between C3 and C2 state is that the bus masters are preventing from writing into the memory in the C3 state. This is, prior to entering the C3 state, done by setting the ARB_DIS bit (ACPI:12h[0]) HIGH which disables the system arbiter. Upon a bus master requesting an access, the CPU will awaken from C3 state if the BM_RLD bit (ACPI: 05h[1])is set, and re-enable bus master accesses by clearing the ARB_DIS to enable the system arbiter. If the BM_RLD bit is not set, the C3 power state is not exited upon bus master requests. From hardware point of view, in the C3 state, to serve bus master requests, it is needed to transit the CPU back to C0 state by deasserting STPCLK# while it is not needed for C2 state. Any interrupt(Wakeup1) will also bring the processor out of C3 power state. Specifically, SiS chip will bring the processor out of C2 or C3 state once G1 sleeping state is entered. Because, CPU processor state is only meaningful in the G0 Working state. To provide a running history for the ACPI driver to determine CPU power state policy, one last information BM_STS (ACPI: 01h[4]) is provided. The BM_STS is set whenever any bus master request (REQ#) is asserted. *Figure 3-17* illustrates the processor power state diagrams supported by SiS chip.

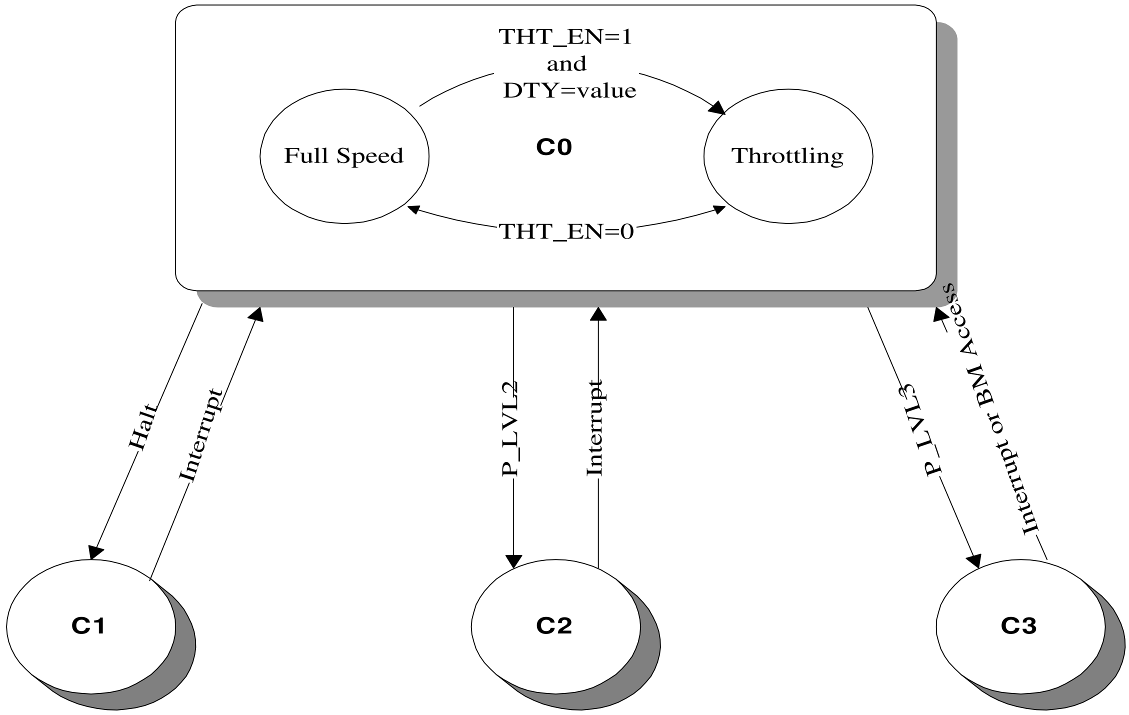


Figure 3-17 Processor Power State Diagram

SCI/SMI# Generation Logic

The SCI generation logic mainly issues the SCI interrupt to invoke the SCI interrupt handler to respond to the power management events. These power management events, instead of initiating SCI, will issue SMI# if the system works in Legacy mode with SCI_EN being 0. As a summary, the following events can raise SCI or SMI#, and can also be enabled as the wakeup events while in the G1 state.

1) Power management timer event,
2) Power button event,
3) Real time clock alarm event,
4) Ring event,
5) Hotkey event(or EXTSMI#),
6) General purpose timer event,
7) USB request event,
8) General purpose I/O events(GPIO0,
GPIO1,GPIO2,GPIO5/STARTREQ,GPIO7,
GPIO8,GPIO9/THRM,and GPIO10).

One specific event which is BIOS_SCI event only generates SCI. The BIOS_SCI event is provided to allow the BIOS to invoke SCI handler for some specific application. By writing a 1 to BIOS_RLS locating in the bit 10 of GPIO_MUX1, the GBL_STS status bit is set. The SCI will then be generated if GBL_EN bit (ACPI: 03h[5]) is set.

There are also five events that only generate SMI#. The generation of SMI# through these events is independent to the status of SCI_EN bit.

1) ACPI enable event,
2) ACPI disable event,
3) Serial IRQ event,
4) Periodic SMI event, and
5) ACPI_SMI event

While working in the Legacy system state, the events provided in the legacy PMU block are also the sources of generating SMI# as a backward compatibility. Therefore, the users can choose ACPI or Legacy PMU on their own need. Please refer to Figure 3-18 for more information.

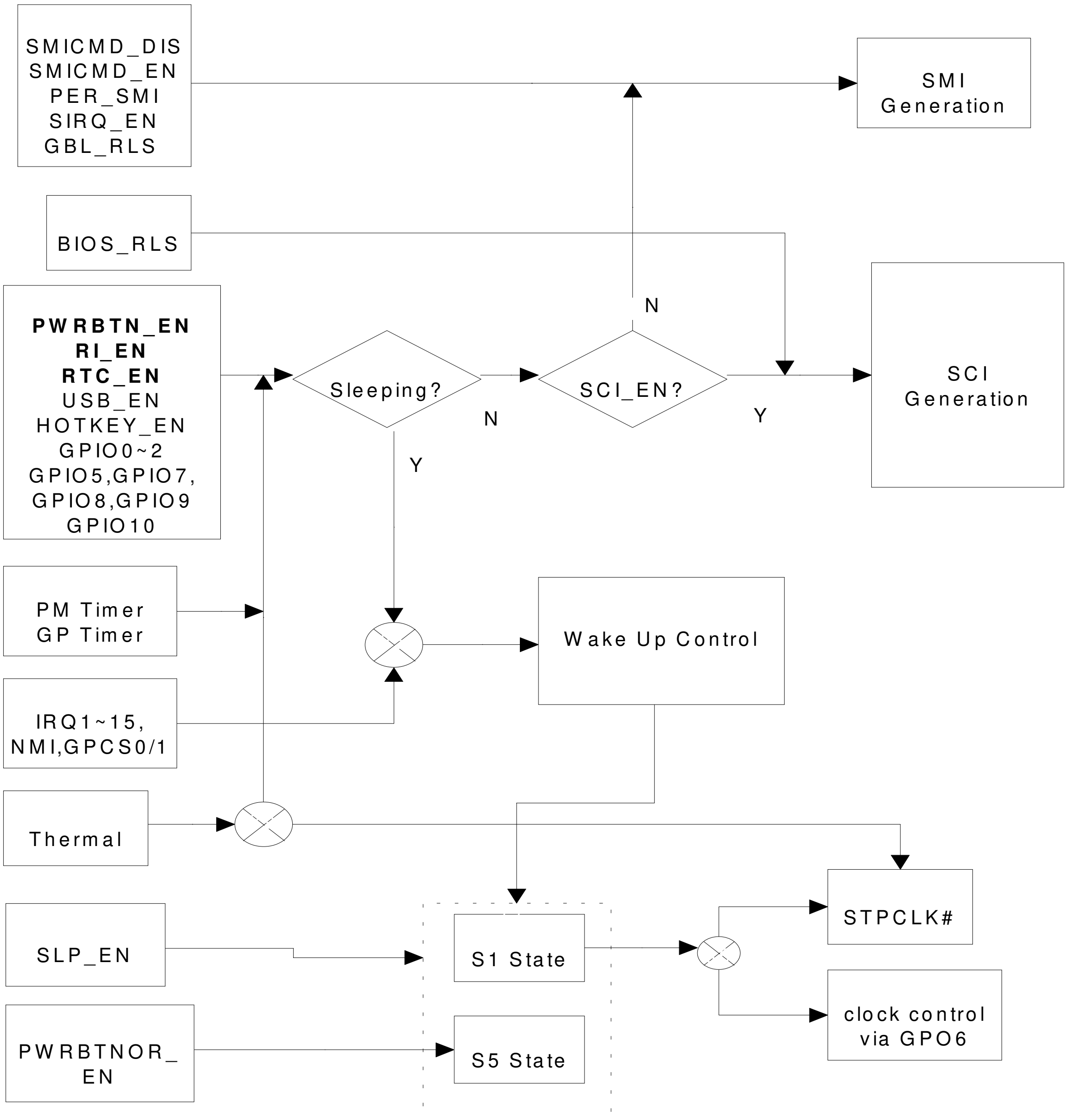


Figure 3-18 ACPI working mechanism overview

The following paragraphs describe some power management events:

Power management timer

The ACPI specification requires a power management timer (PM timer) which provides OS an accurate time value to monitor the system idle time. For instance, if CPU idle time is longer than the threshold value, OS may let CPU enter power saving state like C1, C2, C3. The PM timer is a 24-bit fixed rate free running count-up timer that runs off a 3.578545 Mhz clock in the SiS chipset. The status bit TMR_STS (ACPI:00h[0]) is set when the most significant bit of the timer (bit 23) is changed from “1” to “0” or from “0” to “1”. If the enable bit TMR_EN (ACPI:02h[0]) is set, then the setting of TMR_STS will raise an ACPI event. The current value of the timer can also be accessed by reading TMR_VAL register (ACPI: 09h-0Bh). The timer will be cleared as long as the system stays at S1 sleeping state. Please refer to the Figure 3-19 Power management Timer Flow Chart for the power management timer event flow chart.

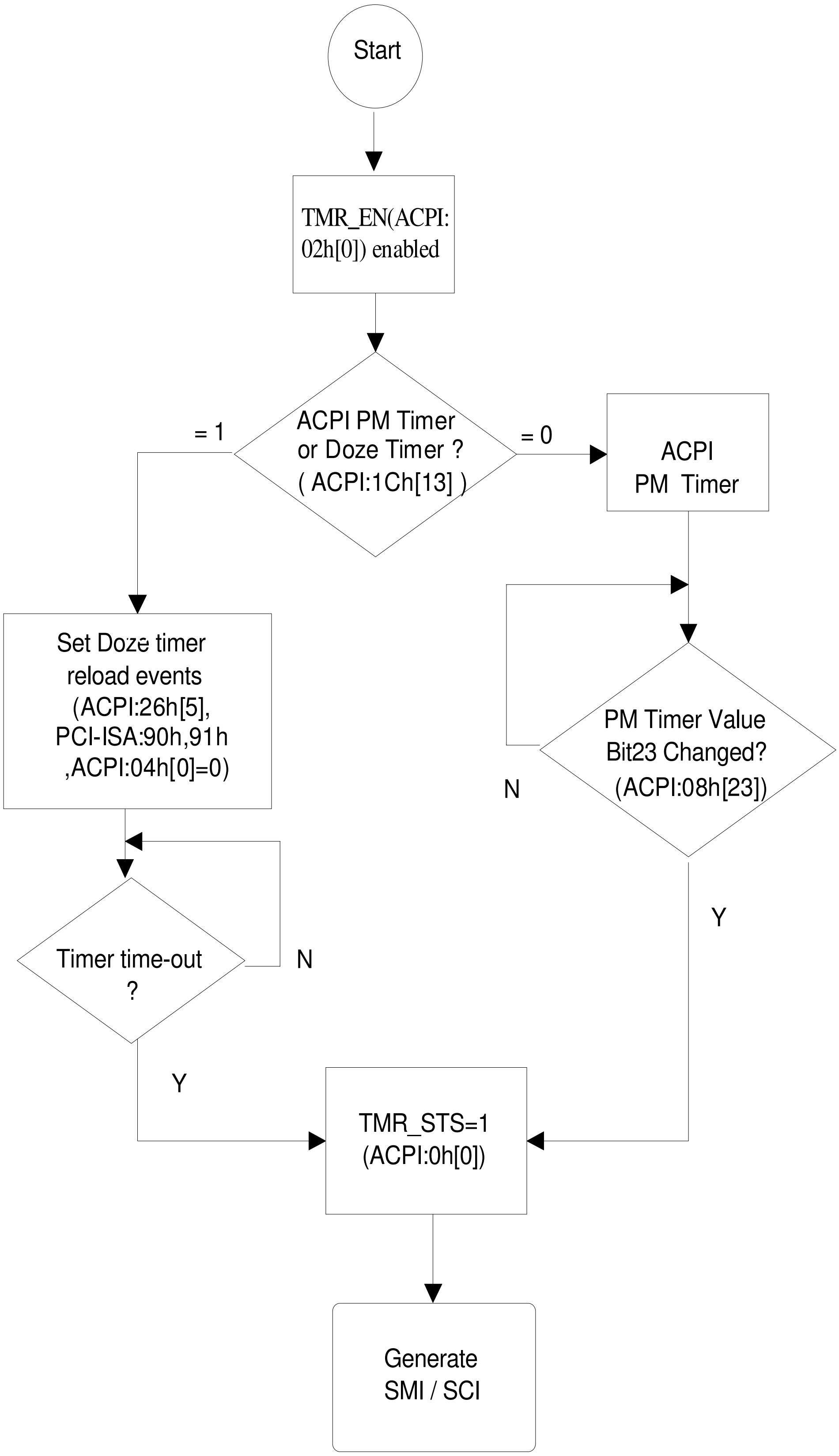


Figure 3-19 Power management Timer Flow Chart

Power Button switch

This switch is an user interface control instead of the traditional power supplier switch. It can be used to cycle the system between the G0 and G1 states through the power management event. Besides, the power button provides user a "Fail-safe" mechanism to force the system to enter the G2 state (Soft-Off) when the system has "Hung", called the power button override.

An 1ms debouncer associated with the power button is used to recognize and respond to the active low logic presented on the pin for more than 1ms. If the PWRBT# is pressed for more than 4 seconds, the SiS chip will turn off the system power by deasserting the ONCTL#.

If the PWRBT# is released within 4 seconds then only the PWRBTN_STS bit (ACPI:00h[8]) will be set. If the PWRBTN_EN bit (ACPI:02h[8]) is also enabled, an SMI or SCI will be raised. Please refer to the Figure 3-20 Power Button Switch Flow Chart for the power button event flow chart.

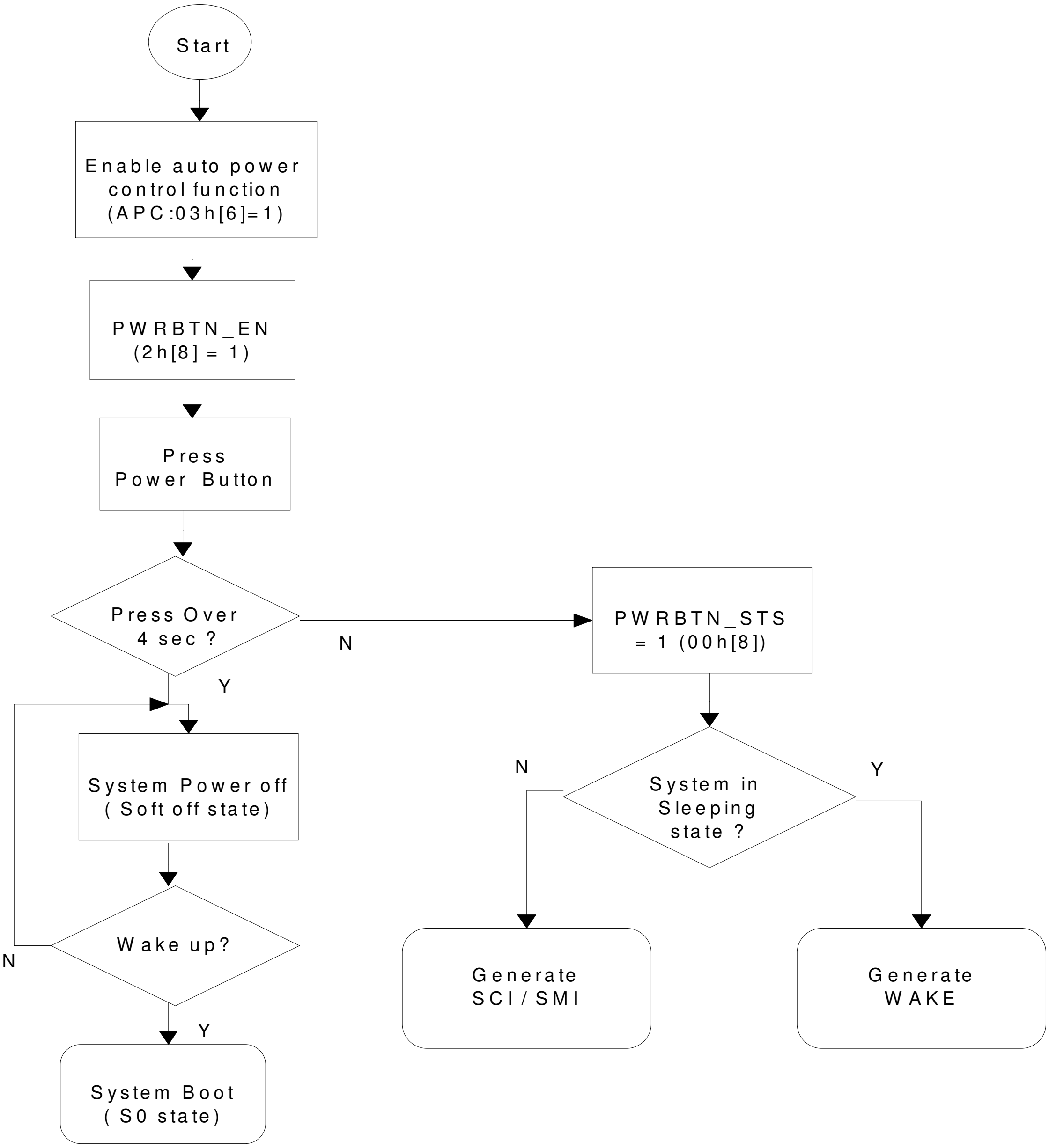


Figure 3-20 Power Button Switch Flow Chart

Real time clock alarm

It is required to extend the current RTC definition of a 24-hour alarm to a one-month-alarm in ACPI specification. To extending the alarm bytes, SiS chip supports both the "Day Alarm", and "Month Alarm" function. The Day Alarm byte, and Month Alarm byte are located in the 7Eh, and 7Fh of the Standard Bank of the RTC CMOS RAM, respectively. The OS will attempt to identify the RTC as a possible power management event source by checking the RTC_EN bit (ACPI:02h[10]) and RTC_STS bit (ACPI:00h[10]). Note that the RTC_STS bit will be set by the external IRQ8# if external RTC is used. Users can set any specific time to generate an SCI if system is in the working state, or a wake-up event if system is in the sleeping state. Please refer to Figure 3-21 RTC Alarm Flow Chart for the RTC alarm event flow chart.

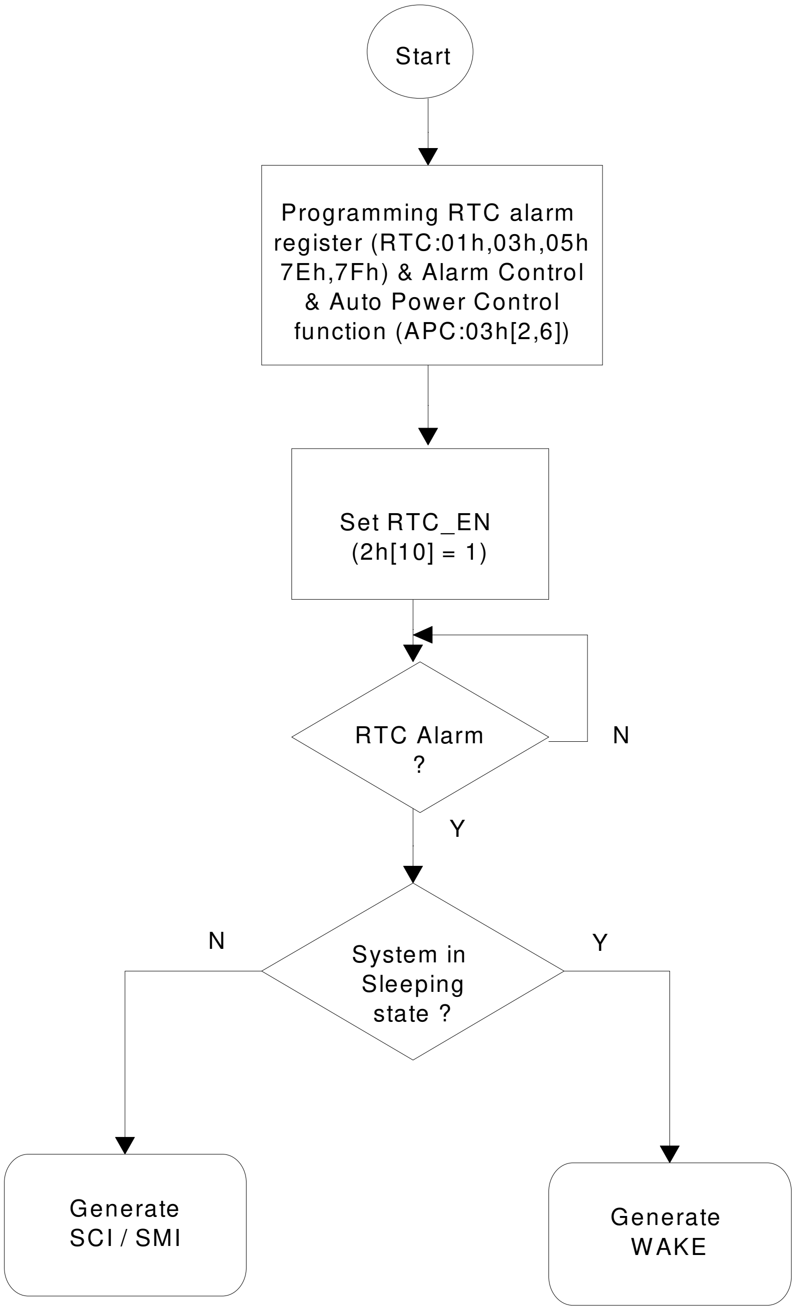


Figure 3-21 RTC Alarm Flow Chart

Ring, USB, Serial IRQ

Because the power management is controlled by OS in ACPI system, it is important to know that there is an event from the peripheral devices. The SiS chipset offers many hardware pins and internal detection to provide designers some flexible implementations. They are Ring, USB, Serial IRQ and GPIOs.
Two debouncers are associated with the RING to allow two possible modes of activation. The default mode supports 150ms detection on the RING signal while the other mode supports frequency between 14Hz to 70Hz. These two modes can be selected via ACPI:1Ch[0].
For ring detection, the RI_STS bit (ACPI:14h[0]) will be set if the ring signal is sensed asserted. If the RI_EN bit (ACPI:16h[0]) is also enabled, then the power management event will be generated. Please refer to Figure 3-22 Ring Flow Chart for the ring event flow chart.

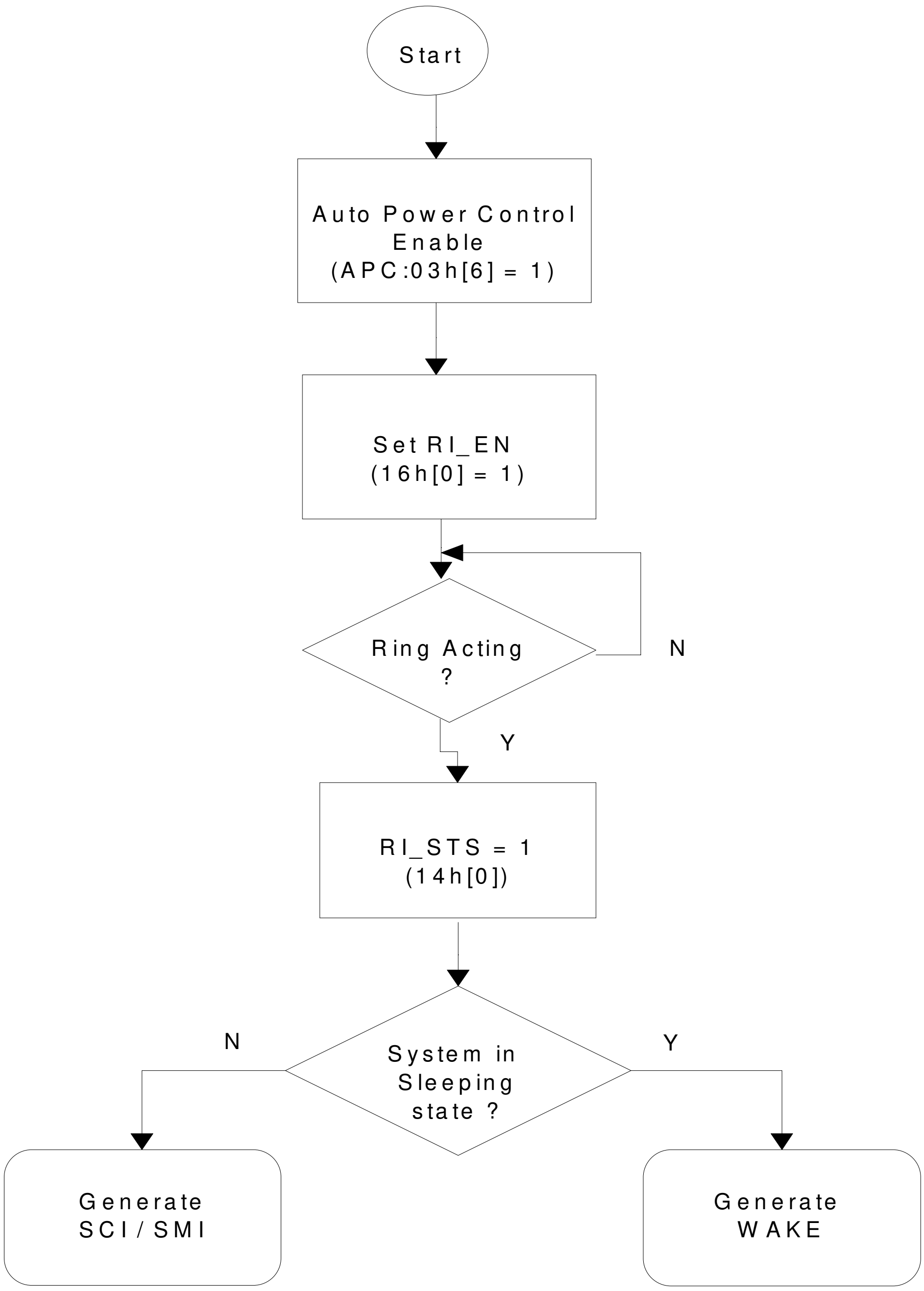


Figure 3-22 Ring Flow Chart

For USB or serial IRQ detection, a power management event is generated by chipset internal function through USB_EN (ACPI:16h[14]) & USB_STS (ACPI:14h[14]) or through SIRQ_EN (ACPI:16h[8]) & SIRQ_STS (ACPI:14h[8]) if there is an interrupt or an USB device changing. For more information please refer to Figure 3-23 USB Flow Chart and Figure 3-24 Serial IRQ Flow Chart.

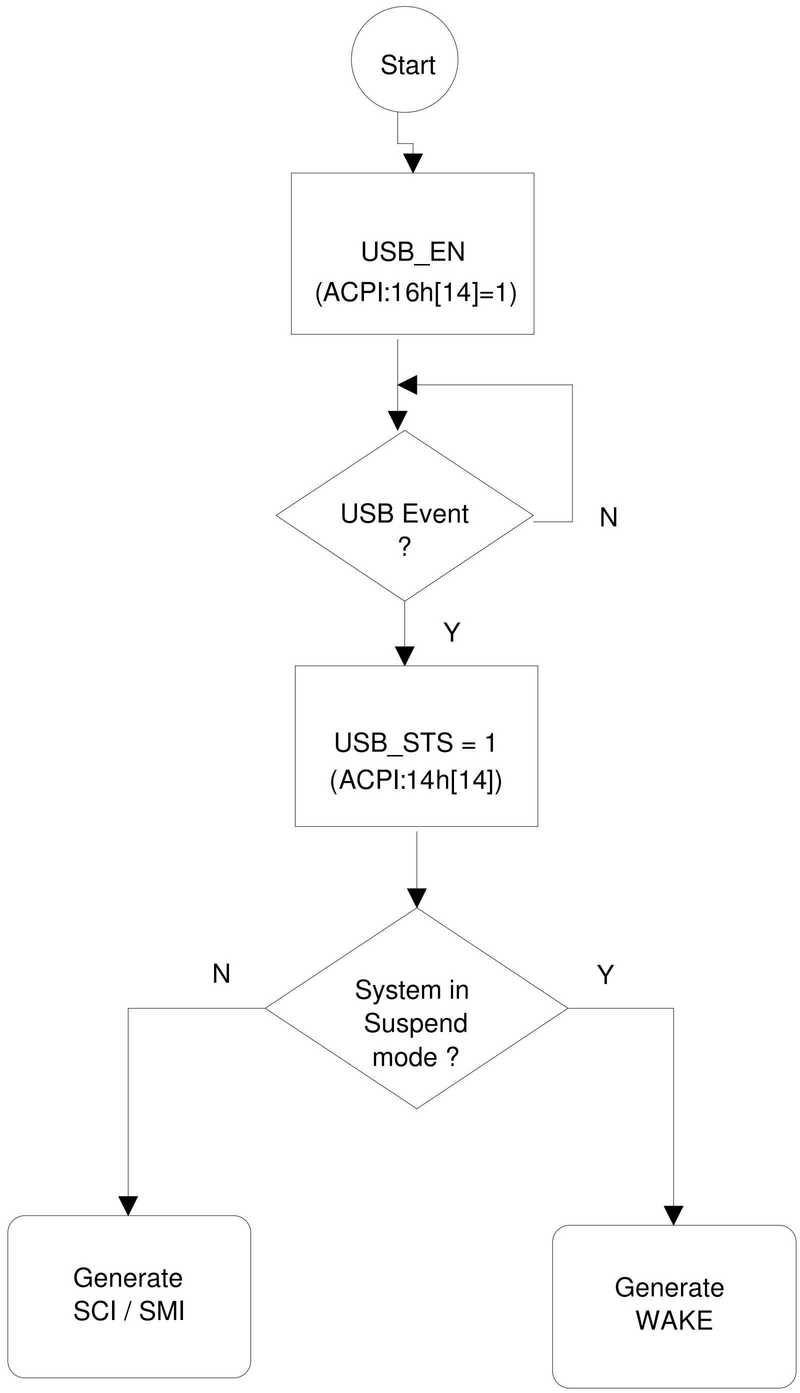


Figure 3-23 USB Flow Chart

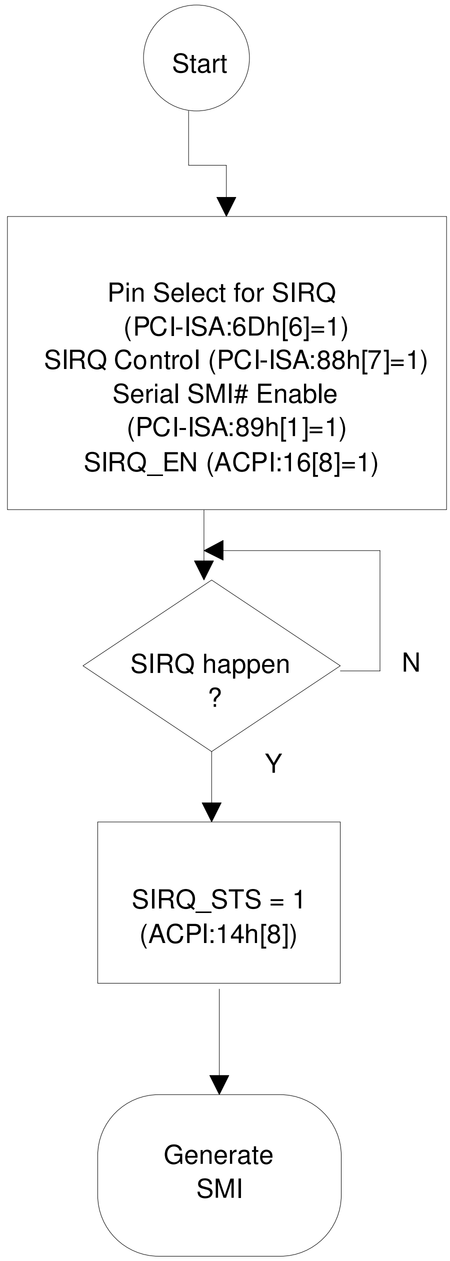


Figure 3-24 Serial IRQ Flow Chart

General Purpose I/O(GPIOx) Events:

SiS chip provides eleven pins to support general purpose I/O function. Three(GPO3, GPO4, and GPO6 are output only) The rest(GPIO[2:0],GPIO5,GPIO[10:7]) are bidirectional. The input/output attribute of these pins can be programmed through setting or resetting the corresponding bits of the GPIO_SEL1, and GPIO_SEL2 registers. By default, all the GPIO pins are input. While in the input mode, the "active logic" level can be programmed through GPIO_LVL1, and GPIO_LVL2 registers. The default active level is low. When the active level is sensed on the GPIO pin, the corresponding status bit in the GPIO_STS1, or GPIO_STS2 is set. Then, the SCI, or SMI# is generated if the corresponding enable bit in the GPIO_EN1, or GPIO_EN2 is set. Writing a 1 to the status bit can clear the bit. In addition, the input level of each GPIO pins can be directly read back by reading the corresponding bit in the

GPIO_CNT1 or GPIO_CNT2 registers. While in the output mode, the logic of each GPIO pin can be controlled by writing the desired value to the corresponding bit in the GPIO_CNT1 or GPIO_CNT2 registers to control the peripheral device power, for instance.

Hotkey Event

When internal keyboard controller is enabled, the HOTKY_STS will be set if the “CTRL+ALT+Backspace“ is recognized. Then SCI/SMI# is generated if HOTKY_EN (ACPI:17h[1]). is set. The HOTKY_STS (ACPI:15h[1]) can also be set once the TURBO/EXTSMI# is activated. The active level of TURBO/EXTSMI# can be programmed through bit 3 of GPIO_LVL1. register.

Thermal Detection

SiS chipset can program GPIO9 for the thermal detection input THRM#. A 1ms debouncer is used to sense the status of THRM#. When the logic of the THRM# matches the programming active level, the STPCLK# is throttled if the THRM_THEN bit (bit 3 of register 1Ch) is set. The throttling will be stopped if the THRM# goes back to the inactive state as a result of the system temperature may be cooled down. Note that it is not necessary to set the THT_EN bit for throttling the STPCLK# in response to the THRM# request. If THRM# is asserted, the system can be programmed to enter the throttling mode directly or generate an SCI/SMI instead. These two options can be selected by the thermal throttling function bit (ACPI:1Ch[3]). Please refer to Figure 3-25 Thermal Detection Flow Chart for the thermal detection flow chart.

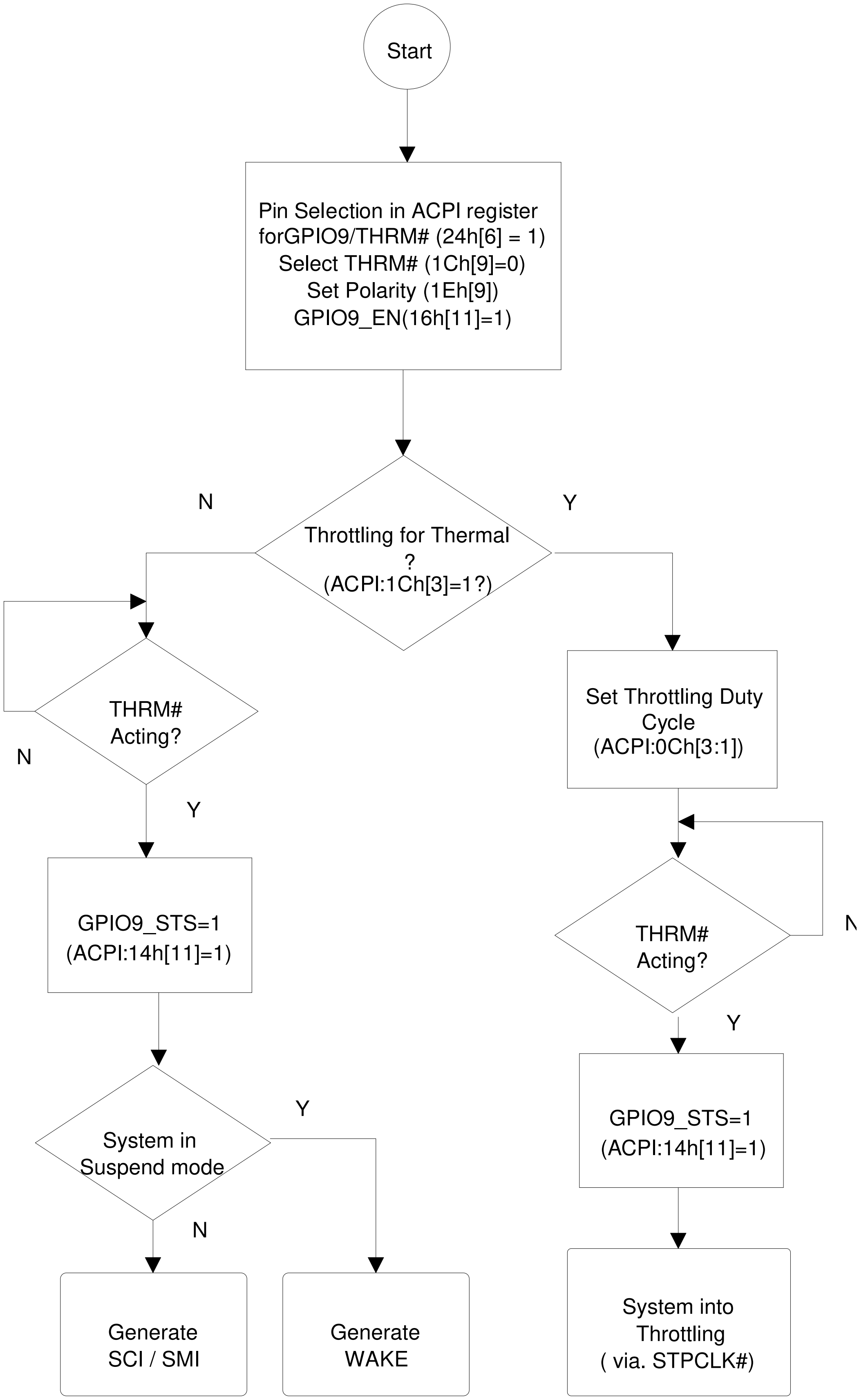


Figure 3-25 Thermal Detection Flow Chart

CPU low power state

The greatest power consumption component in a system is CPU. There is a great power saving if CPU enters the low power state or turns off when it is idle. The SiS chipset supports the C0-C3 CPU power states. In C1 state, system sends halt command to CPU through software in ACPI configuration register. In C2 and C3 state, system halt the CPU by asserting the STPCLK#. However, the Bus Master can still access the DRAM in the C2 state. In the C3 state, CPU should go back to C0 for the Bus Master accessing. Please refer to Figure 3-26 CPU Power State Flow Chart for the CPU power state flow chart.

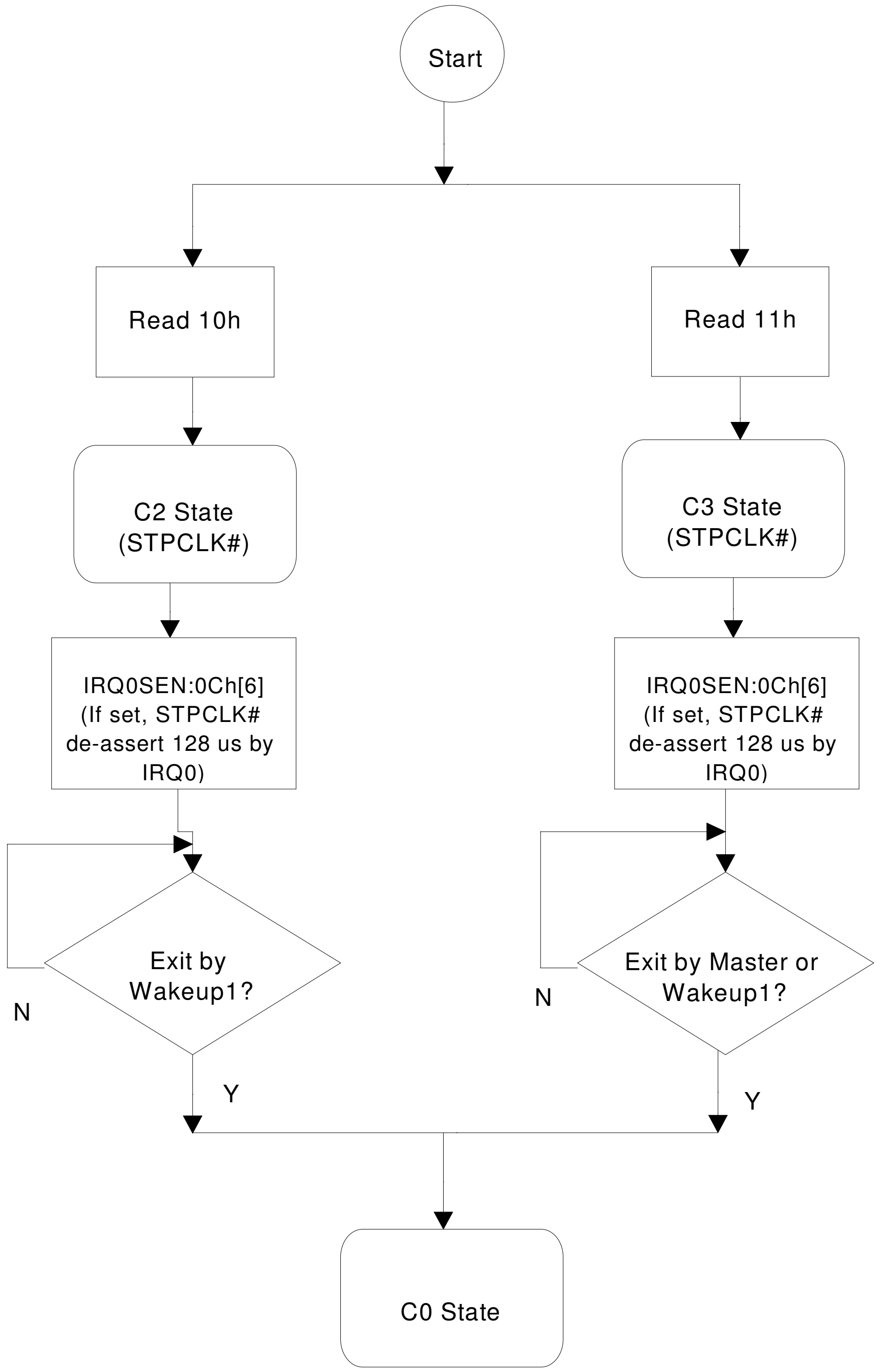


Figure 3-26 CPU Power State Flow Chart

The SiS Chip support S0, S1, S5 states. In S1, all system power is still working, but CPU clock can be stopped by GPO6. Please refer to Figure 3-27 System States Flow Chart. The wakeup event can be programmed by software in ACPI configuration register.
In S5 (soft-off) state, only the RTC power is working. To wake system in soft-off state, the SiS Chip provides RTC alarm, power button switch, APC switch and Modem Ring-in to wake up the system. Please refer to Figure 3-27 System States Flow Chart for the system state flow chart.

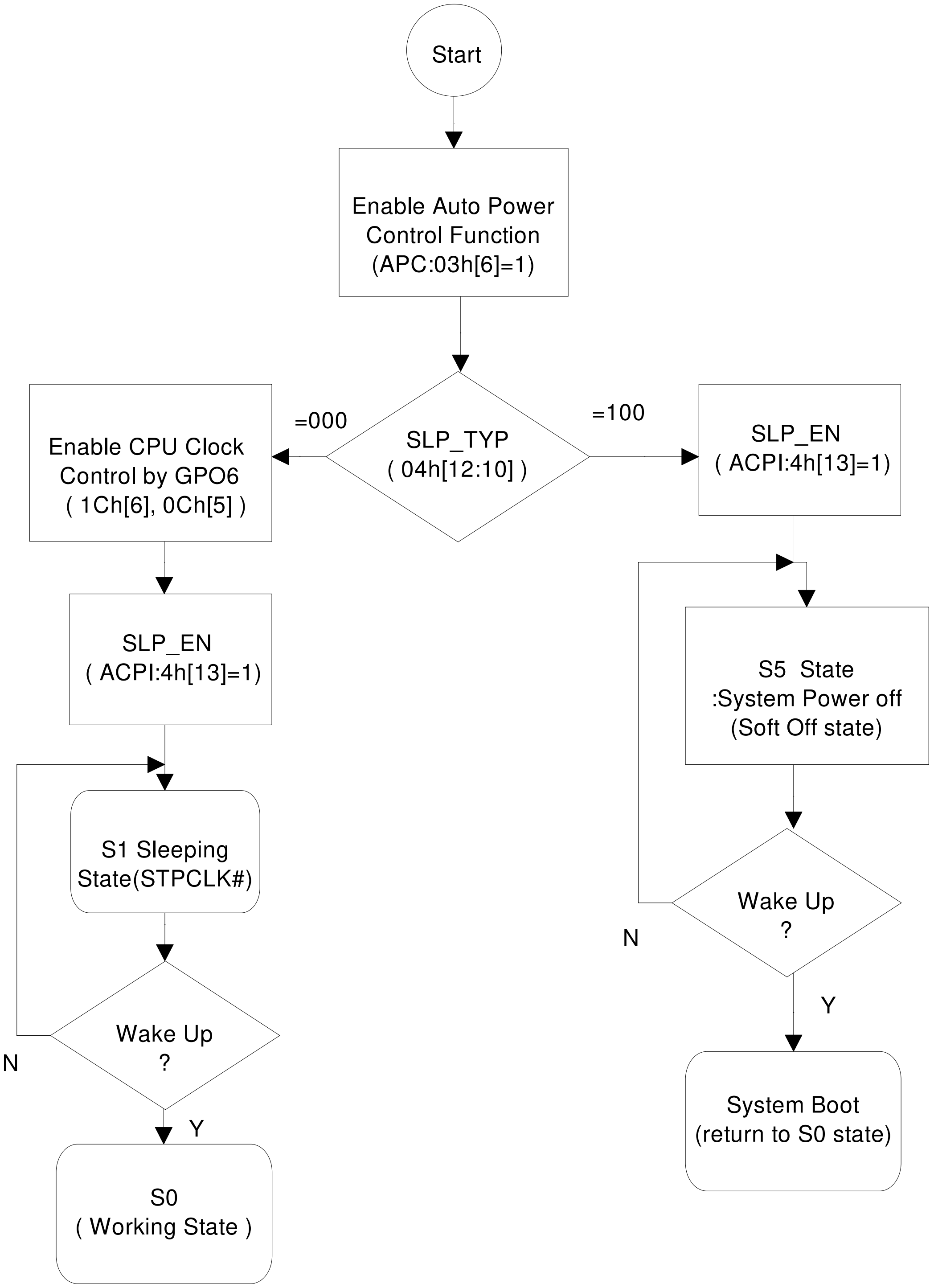


Figure 3-27 System States Flow Chart

General purpose timer and period SMI

General purpose timer is a 8 bits down counter. Its time slot is 1us or 1 min. General purpose timer can be programmed as Suspend timer or BIOS Timer(ACPI:1Ch[11]). After writing counter values, it begin to count. If using as Suspend timer and there is no any reload events (Host-PCI:90h,91h) happened, the timer will time out eventually. There is a power management event when timer is time out. The function of Suspend timer is similiar to System Standby Timer. However, Suspend Timer can assert SCI or SMI but System Standby Timer can only assert SMI.

Period SMI can generate SMI# every 16 sec if the PERSMI_EN (ACPI:26h[2]) is set. This allows the SMI# handler to periodically give warning to the user for delivering the "low battery" message, for instance.

These two functions do not belong to standard ACPI but for the convenience and variety of power management design. Please refer to *Figure 3-28* and Figure 3-29 for more information.

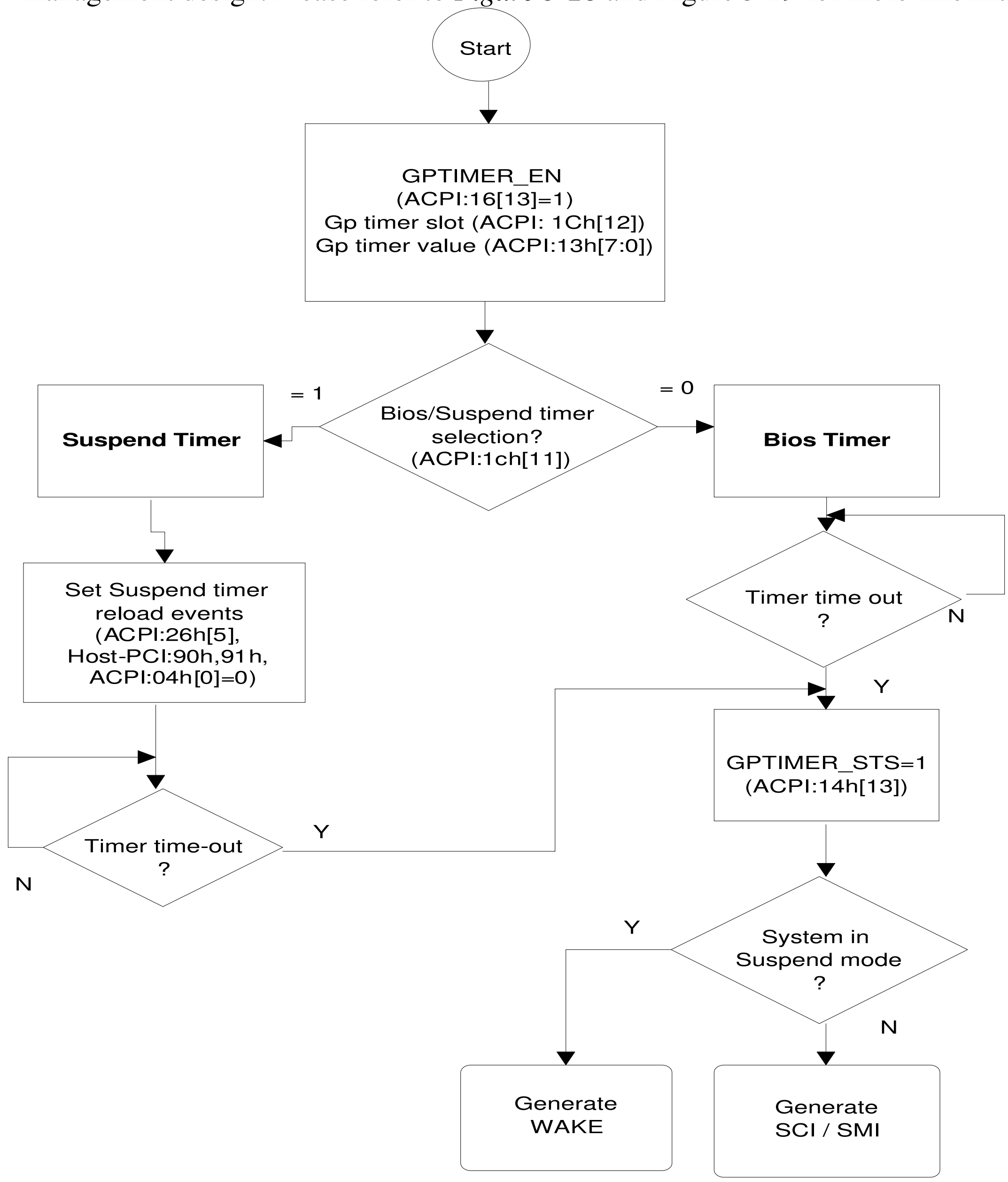


Figure 3-28 General Purpose Timer Flow Chart

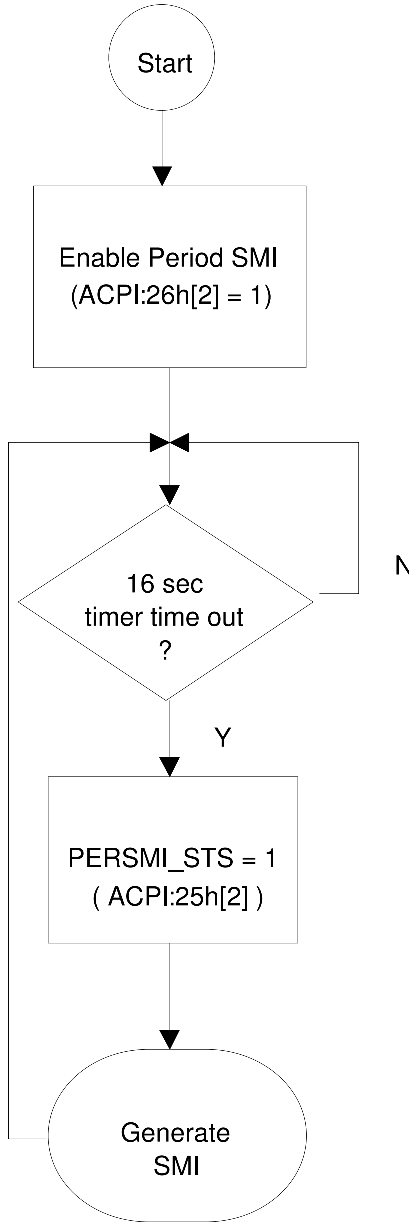


Figure 3-29 Period SMI Flow Chart

ACPI_SMI event:
The ACPI_SMI event, similar to the BIOS_SCI, is provided for the ACPI handler to invoke SMI# handler. Writing a 1 to GBL_RLS(bit 0 of AUX_STS1 register) will set the BIOS_STS bit, and generate SMI# if BIOS_EN bit is set..

Wakeup Logic
When in the G1(or S1) system state, the events that generates SCI or SMI# in the G0 state can be programmed to serve as the wakeup events while the system is in G1 state.
Any of the enabled wakeup events will set the wake status bit(WAK_STS) which allows the ACPI driver to separate sleeping from waking code. As mentioned in the preceding, activating Wakeup1 will also set the WAK_STS bit, and awake the system.

One difference among them is that any power button press will unconditionally set the PWRBTN_STS bit and awakens the system, regardless of the value of the PWRBTN_EN bit while in the G1 sleeping state.

SCI event source

Following are the sources which can generate the SCI.
Power button
RTC alarm
Global status bit (ACPI:00h[5])
Power management timer
Peripheral device IRQ
USB
RING
General purpose timer
SIRQ
Hot key (via EXTSMI#)
GPIO

3.7 Integrated PCI Master/Slave IDE Controller

Overview

SiS Chip supports a full function PCI IDE controller capable of PIO, DMA and Ultra DMA/33 mode operation. It can be supported by programming the internal registers to support PIO Mode 0 ~ 4, Single/Multi-Word DMA Mode 0 ~ 2 and Ultra DMA Mode 0 ~ 2 timing. The IDE Controller block diagram is shown as below:

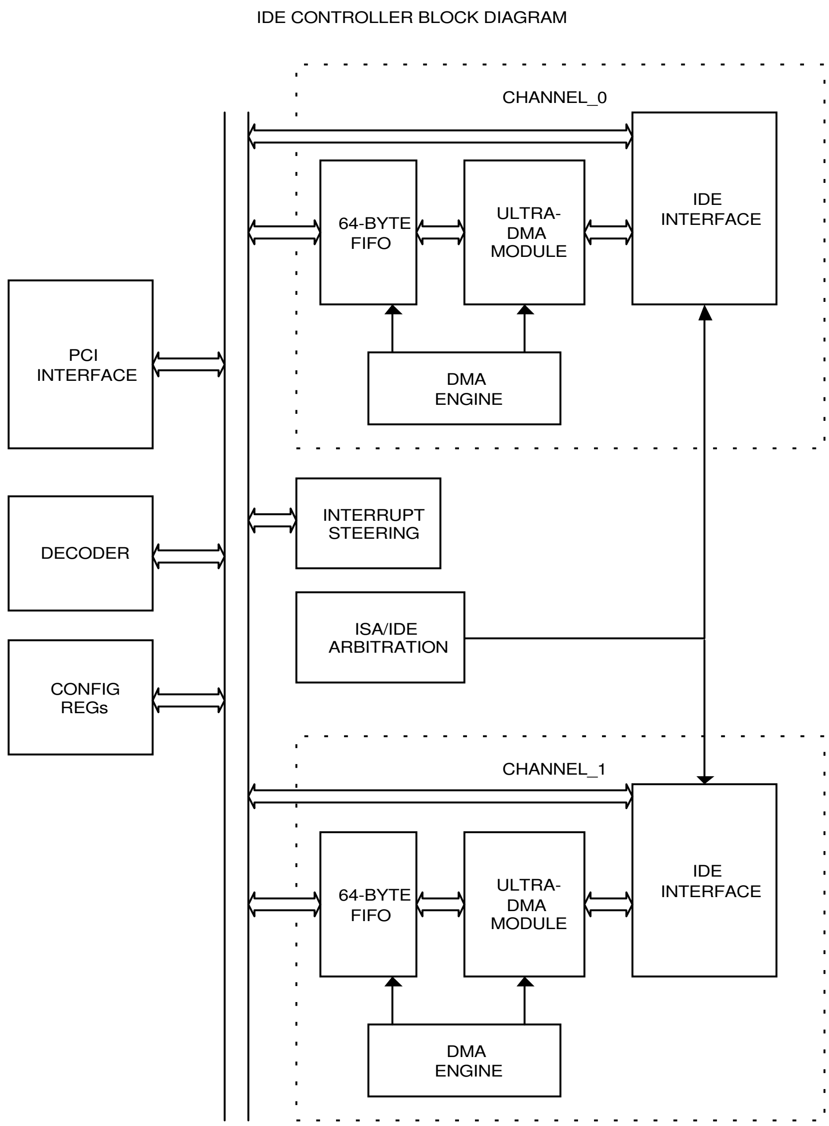


Figure 3-30 IDE Controller Block Diagram

There are two 64-byte FIFO associated with two IDE channels. The data can be popped into FIFO by the unit of word or double-word. All accesses to the IDE data port will go through FIFO, no matter prefetch/postwrite is enabled or not. Accesses to the command or control port will bypass FIFO. This mechanism allows the host to access command or control ports when FIFO is not empty. The FIFO has an option to be 32-byte in depth(from Register 52h bit 0 in PCI IDE configuration space), which is for backward compatibility only and is suggested not to be used. SiS Chip provides the 64-byte FIFO mainly to support Ultra-DMA. Because the Ultra-DMA can be operated at twice the speed of traditional DMA in mode-2, a small FIFO may easily become bottleneck and degrade system performance.

The host may need to access command or control ports when PIO mode or DMA mode data transfer is undergoing. The IDE controller provides a mechanism to complete the command/control port access without disrupting the operation of FIFO.

In PIO mode, when doing postwrite, the command/control port access is held-off until the FIFO is flushed to IDE. When doing prefetch, the command/control access is held-off until the FIFO is full. Before the command/control port access is actually carried out, the host will be keep waiting on PCI bus.

In DMA mode, the command/control access will go through a higher priority than the DMA data transfer cycles. When the command/control access cycle is first seen on the PCI bus, the controller will retry the cycle so that PCI bus will not be used by the host while it is only waiting. At the same time, the controller will suspend the DMA data transfer cycles by completing the current cycle successfully, then de-asserts IDACK# to inform IDE device to stop the DMA data transfer. The IDE device may or may not deassert its IDREQ at this moment. On the other hand, the host should keep retrying the command/control cycle on PCI bus. Eventually the cycle will be accepted and carried out when DMA data transfer is stopped. After the command/control cycle is completed, the controller resumes DMA data transfer cycles as soon as the IDE device asserts IDREQ.

Both primary and secondary channels may be programmed as Native mode or Compatibility mode via the Class Code Field in the controller's Configuration Space register.

In Compatibility mode, the interrupt requests for channel 0 and channel 1 are rerouted to IRQ 14 and IRQ 15 of the built-in Interrupt Controller.

Following table illustrates the accessing methods to the I/O ports in compatibility mode:

Primary Channel:

			READ		WRITE	
PORT	ICSA1#	ICSA0#	IIORA#	IIORB#	IIOWA#	IIOWB#
1F0	1	0	0	1	0	1
1F1	1	0	0	1	0	1
1F2	1	0	0	1	0	1
1F3	1	0	0	1	0	1
1F4	1	0	0	1	0	1
1F5	1	0	0	1	0	1
1F6	1	0	0	1	0	1
1F7	1	0	0	1	0	1
3F6	0	1	0	1	0	1

Secondary Channel:

			READ		WRITE	
PORT	ICSB1#	ICSB0#	IIORA#	IIORB#	IIOWA#	IIOWB#
170	1	0	1	0	1	0
171	1	0	1	0	1	0
172	1	0	1	0	1	0
173	1	0	1	0	1	0
174	1	0	1	0	1	0
175	1	0	1	0	1	0
176	1	0	1	0	1	0
177	1	0	1	0	1	0
376	0	1	1	0	1	0

In Native mode, the interrupt requests of both channels (channel 0 and channel 1) share the same PCI interrupt pin. The interrupt pin may be rerouted to any one of eleven ISA compatible interrupts (IRQ[15:14], IRQ[12:9], and IRQ[7:3]) via programming Register 61h bits 3:0 in PCI to ISA bridge Configure space.

Meanwhile, accessing of the I/O ports are via the addresses programmed in Base Address Registers 10h~13h, 14h~17h, 18h~1Bh and 1Ch~1Fh in PCI IDE configuration space.

While serving as a bus master device, the IDE controller may transfer data between IDE devices and main memory directly. By performing the DMA transfer, IDE offloads the CPU and improves system performance. Bus master DMA programming is according to the information specification "Programming Interface for Bus Master IDE Controller".

The integrated IDE controller contains PCI configuration header and registers to meet PCI 2.1 specifications. The integrated PCI IDE controller supports PCI type 0 configuration cycles of configuration mechanism #1.

Proper cycle timing is generated to meet PCI Bus speed and different modes of IDE drive. All cycle timing can be controlled by software programming from Register 40h to Register 49h in PCI IDE configuration space.

As a slave device, IDE decodes and interprets PCI cycles and generate signals to start and terminate IDE cycles. This block responds only to cycles that belong to IDE I/O address space. It supports both 16-bit and 32-bit I/O data transfer at address 1F0/170. All other IDE registers read or write operations are 8-bit only.

PIO mode operation

The IDE controller is capable of doing prefetch or postwrite in PIO mode. The count(in bytes) of prefetch length for each channel can be programmed in Prefetch Count Registers 4Ch~4Dh and 4Eh~4Fh in PCI IDE Configuration space. Normally, the count will be programmed as 512(2^9), which is the size of a single sector. The prefetch and postwrite functions can be enabled or disabled independently through control bits in Register 4Bh of PCI IDE

configuration space. When prefetch is enabled, the controller will start prefetching when the first read data port command is received. It will keep prefetching until the FIFO is full or when prefetch count is reached. Whenever the FIFO becomes non-empty again, the prefetch will automatically resume until the prefetch count is reached.
When postwrite is enabled, the host can write data to FIFO in word- or Dword- increment. The IDE controller will automatically start IDE write cycles as long as FIFO is non-empty. When the fast postwrite function is enabled, the write IDE data port command on PCI bus will last for 3 PCI clocks only. When disabled, the PCI command will be 5 PCI clocks.

DMA mode operation

There is a DMA engine associated with each channel. The DMA engine can be invoked by writing the start-bit in Bus Master command register. The DMA engine will first request for PCI bus to read the descriptor from memory, load the address pointer and byte-count. For IDE read operation, the controller will start prefetching data into FIFO at this moment. When FIFO is half-full (or 75% full, programmable), the DMA engine will request for PCI bus to flush the data in FIFO to memory. If the prefetch count is reached while the FIFO is not yet half-full, the DMA engine will also request for PCI bus to flush the FIFO. For write operation, after descriptor is read, the DMA engine will again request for PCI bus to read data from memory to FIFO. At the same time, when the FIFO becomes non-empty, the controller will automatically start IDE write cycles to flush data in FIFO to IDE device. When data in FIFO is less than eight bytes, the DMA engine will again request for PCI bus to re-fill the FIFO.

Normally, the byte-count loaded in IDE controller will be equal to IDE transfer size programmed to IDE devices. If the two values were programmed differently, the IDE controller and the software that driving IDE should work together to prevent system from failure.

When the DMA engine is writing IDE

If the byte-count was programmed to be greater than the IDE transfer size, the IDE device will de-assert IDREQ signal when the transfer size is reached and issue interrupt to IDE controller. The IDE controller will pass transparently the interrupt to host. When the host clears the start-bit in response to the interrupt, the IDE controller will simply discard the remaining data in FIFO. When the host read the status bit, it will see the interrupt bit set and active bit also set. This will be interpreted as a normal ending. If the byte-count was programmed to be less than the IDE transfer size, the controller will exhaust its data in FIFO while IDREQ signal is still asserting. The host should time-out because it does not receive any interrupt. When the host reads the status register, it will see the interrupt bit not set and the active bit set.

When the DMA engine is reading IDE

If the byte-count was programmed to be greater than the IDE transfer size, the IDE device will de-assert IDREQ signal when the transfer size is reached and issue interrupt to IDE controller. The IDE controller should mask the interrupt, request for PCI bus to flush all the data in FIFO to memory. After the FIFO is empty, the controller will unmask the interrupt to inform host that all data is visible in memory. The host, after received the interrupt, will read the status register and see the interrupt bit set and active bit also set. This will be interpreted as a normal ending.

If the byte count was programmed to be less than the IDE transfer size, the IDE controller will stop prefetching when its byte-count has reached while IDREQ signal is still asserted by device. The controller may or may not flush its data in FIFO to memory, depending on whether the FIFO has reached its request level or not. The host will eventually be time-out because it does not receive any interrupt. When the host reads the status register, it will see the interrupt bit not set and the active bit set. The remaining data in FIFO will be discarded when the host clears the start-bit.

Ultra-DMA/33 Operation

Ultra DMA is a fast data transfer protocol used on IDE bus. By utilizing both the rising edge and the falling edge of the data strobe signal to latch data from DD[15:0], the data transfer rate is effectively doubled than that of the traditional multi-word DMA while the highest fundamental frequency on the cable is the same. In view of the faster transfer rate on IDE bus may easily fill the FIFO up when reading IDE device, in such condition the IDE bus will be idle and result in system performance degradation, SiS Chip lengthens the internal FIFO for each channel (channel 0/channel 1) to 16-Dword to improve system performance. When the FIFO is half-full (or 3/4-full, programmable), the DMA engine should request for PCI bus by asserting an internal request signal to system arbiter. The system arbiter, based on an algorithm described in the previous sections, shall grant the PCI bus to DMA engine by asserting an internal grant signal to it. Ideally, the FIFO should never be full during data-in operation so that the burst data transfers on IDE will not be suspended. When the IDE controller is transferring data from system memory to IDE, the DMA engine will initiate PCI burst cycles to read data from memory into FIFO until FIFO is full. The FIFO will decrease at the rate of the selected Ultra DMA mode as the IDE controller doing data-out operation. In the best situation, the FIFO should not be empty during data-out operation otherwise the burst data transfer on IDE will be suspended.
The Ultra-DMA mode can be enabled on a per-device basis and all three timing modes(0-2) are supported by programming the corresponding configuration registers. For Ultra-DMA operations, the following signal lines shall change to their new definition when IDACK# is asserted. These signals will revert back to their old definitions right after IDACK# is de-asserted.

The following table shows the signal line difference between old definition and new definition (Ultra DMA).

Old Definition	New Definition
IIOW#	STOP#
IIOR#	HDMARDY# --- data in operation HSTROBE --- data out operation
ICHRDY#	DSTROBE --- data in operation DDMARDY# --- data out operation

There are three phases for an Ultra-DMA operation as defined in the protocol: Burst Initiation phase, Data Transfer phase and Burst Termination phase. The Burst Initiation phase is always initiated by the device when it asserts IDREQ. The SiS Chip will responds IDACK# after the base address and byte-count in the PRD table entry is read from system memory. During Data Transfer phase, either the sender or the receiver can pause a burst to allow for internal data processing and then resume the burst some time later. There are three situations that SiS Chip will pause a burst:

1. As a sender during data-out operation and the internal FIFO is empty. The burst will resume after the DMA engine re-fill the FIFO with data from system memory.
2. As a receiver during data-in operation and the internal FIFO is full. The burst will resume after the DMA engine dump the data in FIFO to system memory.
3. For a PRD table with multiple entries, the DMA engine will start the burst data transfer after base address and byte-count of one entry is read. When the data transfer for the current entry is completed and the next entry has not yet been read into the controller, the SiS Chip shall also initiate a pause. After the base address and byte-count for next entry is read, the burst resumes.

The Burst Termination phase can be initiated by either the SiS Chip or the device. In normal situations, when the data transfer has reached the byte-count as defined in the last entry of the PRD table, the SiS Chip will initiate a burst termination by asserting STOP#. After the termination is acknowledged by the device and HSTROBE signal return to the asserted state, the CRC will be sent on negation of IDACK#. There are two additional situations that the SiS Chip will also initiate a burst termination:

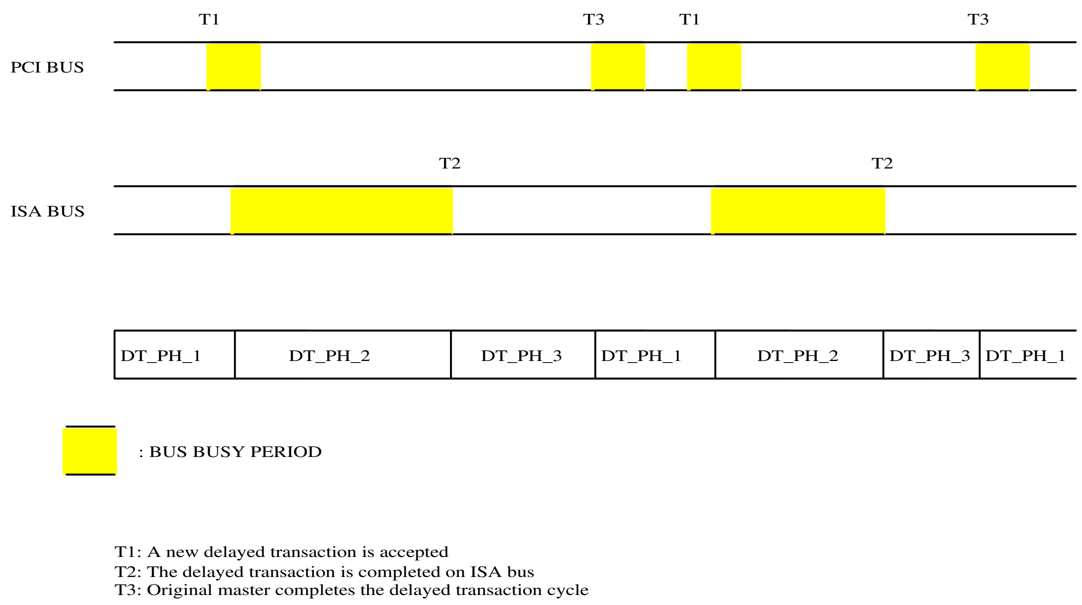
1. During the burst data transfer, the host(CPU) is trying to access the command/control block registers. Since the command/control block access cycle is assigned to have higher priority than data transfer cycles, the SiS Chip must first terminate the burst, de-asserts the IDACK# signal, generate the corresponding DA[2:0] and CS[1:0] on IDE bus, and then complete the command/control block register access cycle. After that, the burst can be resumed by entering the Burst Initiation phase when the device re-asserts IDREQ.
2. Since the usage of the IDE/ISA bus is arbitrated among PCI-to-ISA cycle, ISA masters and IDE controllers. Once the PCI-to-ISA cycle or the ISA masters gains higher priority on the bus and need to access the ISA bus, the IDE controller must yield. In such cases, when Ultra-DMA mode is operating, the controller will initiate a burst termination. After the preempting cycles are finished, the Ultra-DMA burst can be resumed.

3.8 Delayed Transaction

Delayed transaction is a mechanism used when the target, like PCI-to-ISA bridge in SiS Chip on behalf of the ISA devices, cannot complete the transaction within the initial latency of 16 PCI clocks. To support delayed transaction function, the PCI-to-ISA bridge would latch all the information required to complete the transaction and then terminate the master with a retry. The PCI-to-ISA bridge will then translate the request into ISA cycle to obtain the requested data for a read transaction or complete the actual request if a write request. During this period the original master would keep retrying the cycles while other PCI masters are also allowed to use the bus that would normally be wasted holding the original master in wait states. Eventually, the original master would get the latched data for read transaction, or complete the cycle for the write transaction when the PCI-to-ISA bridge completes the ISA cycles.

Delayed Transaction and ISA Master Cycle Arbitration

ISA devices or DMA controller embedded in the PCI-to-ISA bridge of SiS Chip may become ISA master and initiate cycles to access PCI bus. It is quite often that the ISA master may request for ISA bus while there is a delayed transaction undergoing. As a result, an arbitration rule is adopted in the PCI-to-ISA bridge to prevent conflict on the ISA bus. In this section, we will first describe the actions of an ISA master cycle, and next outline the arbitration rules. For convenience, the progress of a delayed transaction cycle will be divided into three phases: DT_PH_1, DT_PH_2 and DT_PH_3.



DT_PH_1: This is the period when there is no pending delayed transaction in progress

DT_PH_2: This is the period when the ISA cycle corresponding to the delayed transaction is undergoing on ISA bus.

DT_PH_3: From the end of ISA cycle up to the original PCI master successfully retries and completes the whole delayed transaction.

Note: the delayed transaction is said to be pending during DT_PH_2 and DT_PH_3.

Traditionally, ISA(DMA) masters request ISA bus by asserting their corresponding DREQs to DMA controller embedded in the PCI-to-ISA bridge. The PCI-to-ISA bridge, in turn, will generate PHOLD# to system arbiter to request for PCI bus. The PHOLD# will be asserted as long as DREQ is asserted by the ISA master. In response to PHOLD#, the system arbiter grants PCI bus to PCI-to-ISA bridge by asserting PHLDA#. The PCI-to-ISA bridge, upon receiving PHLDA#, will first check if ISA bus is busy or idle. If busy, it will defer the assertion of DACK# until ISA bus returns to idle. If idle, it will assert the corresponding DACK# immediately to inform ISA master to start. ISA master when received DACK#, can then start its cycles transferring data to or from PCI(ISA) bus. When ISA master finishes its cycles, it de-asserts DREQ and then PHOLD# will also be de-asserted immediately. The system arbiter, in response to the desertion of PHOLD#, will immediately de-assert PHLDA#. This completes the whole sequence of ISA master cycles.

Note: PHOLD# and PHLDA#, in SiS Chip, are internal signals interfaced between the system arbiter and the PCI-to-ISA bridge.

Delayed Transaction and ISA Master Arbitration Rule

1. When ISA master issues DREQ and there is no pending delayed transaction, this is the normal case that no arbitration is needed and the PCI-to-ISA bridge behaves exactly as that stated above.
2. When ISA master issues DREQ and there is currently a delayed transaction pending, the PCI-to-ISA bridge will disregard the pending delayed transaction and immediately generate PHOLD# to request for PCI bus.
3. When the system arbiter grants PCI bus to ISA master by asserting PHLDA#, and the delayed transaction is in DT_PH_2, i.e., the ISA bus is busy, the PCI-to-ISA bridge should defer the assertion of DACK# until DT_PH_3 is entered. Otherwise, ISA master will start its cycles as soon as DACK# is asserted and may result in ISA bus conflict.
4. If PHLDA# is asserted when the pending delayed transaction is already in DT_PH_3, ie., the ISA bus has returned to idle, the PCI-to-ISA bridge can assert DACK# immediately and hence ISA master may start its cycles even when the delayed transaction is not yet completed on PCI bus.
5. During the period that ISA master is active and delayed transaction is pending in DT_PH_3, the original PCI master that initiated the dalayed transaction will temporarily stop retrying on the PCI bus because PCI bus is now owned by ISA master.
6. After the ISA master finishes its data transfers, the original PCI master should eventually re-gain PCI bus and retry successfully.

3.9 The Architecture of ISA/IDE Multiplexed bus

SiS Chip interfaces to IDE bus and ISA bus through multiplexed pins. The data bus IDA[15:0] of IDE channel_0 share pins with SD[15:0] of ISA bus, while the data bus IDEB[15:0] of IDE channel_1 share pins with LA[23:17] and SA[16:8] of ISA bus. The resulting bus architecture interfaced with SiS Chip will be called IDE/ISA bus. The pin-sharing imposes limitation on the IDE/ISA bus such that IDE and ISA can not be operating simultaneously. As a result, when either of the IDE channels is operating, the ISA bus activities must be idle. Conversely, when the ISA bus is used by the PCI-to-ISA bridge or ISA masters, both the IDE channels must not be operating. There are two exceptions that ISA and IDE can both be operating. One is ISA refresh cycle initiated by refresh controller embedded in SiS Chip. Since only SA[7:0] and MEMR# are used during refresh cycles, it is apparent there will be no conflict between ISA and IDE. The other exception is when the internal registers of legacy ISA bus controllers (8259, 8237, 8254) are being accessed. These registers located inside the SiS Chip and hence no external AT cycles will be generated when they are being accessed. Therefore, these registers can be accessed when IDE is operating.

The IDE bus signals are driven directly by the chip, while the ISA bus signals are further buffered by 74LS245s. Two 74LS245 are used to interfaced with ISA address signals and two 74LS245 are used to interfaced with ISA data signals. The MR16# signal of ISA is used to control the direction of address signals to or from ISA slots. When a ISA master gains ISA bus ownership by asserting MR16#, the direction of address is from ISA to IDE. In all other cases, the direction of address is from IDE to ISA. During ISA refresh cycles, the ENABLE pins of the two 74LS245s interfaced with address signals are disabled by the RFH# signal such that ISA address signals will not appear on IDE and hence IDE operations will not be affected.

The SDOEL and SDOEH signals connect to the DIR pins of 74LS245s and are used to control the direction of ISA data flow. SDOEL is used to control low-byte, and SDOEH is used to control high-byte. When the two signals are high, the direction of ISA data flow is from IDE to ISA. When the two signals are low, the direction of ISA data flow is from ISA to IDE. When either of the IDE channels is operating, the SDOEL and SDOEH will be both high such that the data direction is from IDE to ISA. When PCI-to-ISA bridge or ISA master is active, SDOEL and SDOEH will be depending on the read/write status of the current transaction.

The above mechanism assumes that ISA devices located on ISA slots will not be affected by IDE signals propagate through the 74LS245s and appear on ISA address/data buses when IDE is operating, since the DIR signals will park the 74LS245s in the IDE-to-ISA direction.

To arbitrate the IDE and ISA bus, SiS Chip has developed an arbitration scheme on the IDE/ISA bus. By taking advantage of the arbitration scheme, IDE controller and ISA devices can each get a fair share of bus usage. The arbitration scheme will be described in the following section.

IDE/ISA Bus Limitation

1. The two IDE channels are fully separated and hence can be operating simultaneously without intervening each other.

2. Due to the limitation of multiplexed pins, when any one of the IDE channel is busy, ISA bus activities must remain idle. Conversely, when the ISA bus is busy, the two IDE channels must be idle.

There are three candidates compete for the IDE/ISA bus
1. PCI-to-ISA cycle
2. ISA master
3. IDE controllers (of the two channels)

Basic Rules

1. PCI-to-ISA cycle can preempt IDE cycles immediately
2. ISA master cycles cannot be preempted
3. A simple rotating-priority is adopted for IDE controllers and ISA masters
4. The minimum bandwidth of IDE controller can be guaranteed by programming the minimum accessed time register(50h~51h) in PCI IDE configuration space.
5. ISA master can preempt IDE controller only when its priority is larger than both IDE channels.

Arbitration Scheme

1. Since the PCI-to-ISA cycle and ISA master are already arbitrated by the system arbiter of SiS Chip, it is for sure that they will never be active simultaneously. Therefore, the IDE/ISA arbitration scheme can rule out this possibility.

2. PCI-to-ISA cycle can interrupt IDE controller immediately. When IDE controller of either channels detects a PCI-to-ISA cycle is requesting at the PCI-to-ISA bridge, it should suspend its operation immediately by completing the current IDE cycle. If in DMA mode, it should also deassert DACK#. The IDE controller should remain in idle state until the PCI-to-ISA cycle is complete and then resume its operation. The PCI-to-ISA bridge, on the other hand, should temporarily retry the PCI-to-ISA cycle on PCI bus when any one of the IDE channel is busy. It keeps retrying the cycle until both IDE channels are in idle state. It is obvious that this rule favors PCI-to-ISA cycles because IDE multi-sector data transfers are quite often and may last for a long period of time. If the PCI-to-ISA cycle can not preempt IDE, it may be waiting too long and result in system failure.

3. ISA master cycles cannot be suspended and then resumed later. Once the ISA master was granted to initiate its cycles, it must complete the whole process without being interrupted.

4. To solve the arbitration between IDE and ISA master, a rotating priority scheme is adopted to ensure each of the candidates will get a fair share of bus usage.

Since the ISA master can not be preempted, it can hold the bus as long as it desires. It is likely that IDE channels will not be able to get a fair share of bus usage when ISA master is heavily transferring data. As a supplement, the minimum accessed time for IDE channels can be guaranteed by programming the minimum accessed time register. This 16-bit register defines a minimum accessed time in terms of PCI clock for IDE. Every IDE data transfer is guaranteed not to be preempted by ISA master before IDE has used the bus for this amount of time. As such, the minimum bandwidth of IDE channels can be guaranteed. To count the amount of time that the bus is used by IDE, there is a granting timer associated with each IDE channel counting with PCI clock. Initially, the granting timer is loaded with the value of the minimum accessed time register. For every PCI clock, if the IDE/ISA bus is used by the associated IDE channel, the granting timer should count-down once. When the timer expires and ISA master is requesting for bus, the IDE channel should suspend its cycles and yield the bus to ISA master. The granting timer can be reloaded when ISA master finish using the bus.

Define:
PRIO_ISAM: the priority of ISA master
PRIO_IDE0: the priority of IDE channel_0
PRIO_IDE1: the priority of IDE channel_1

Operation rules for the rotating priority scheme:

- PRIO_ISAM will be the lowest when ISA master finishes its data transfer cycles.
- PRIO_IDE0 will be the lowest when the granting timer of IDE channel_0 expired
- PRIO_IDE1 will be the lowest when the granting timer of IDE channel_1 expired.
- ISA master can only preempt both IDE cycles when

PRIO_ISAM> PRIO_IDE0 and PRIO_ISAM> PRIO_IDE1

Consider the following sequence of events as an example.

Initially, after the system is reset:

PRIO_ISAM > PRIO_IDE0 and PRIO_ISAM > PRIO_IDE1

After the first ISA master cycle transfers:

PRIO_ISAM < PRIO_IDE0 and PRIO_ISAM < PRIO_IDE1

After IDE channel 0 data transfer and its granting timer expires

PRIO_ISAM > PRIO_IDE0 and PRIO_ISAM < PRIO_IDE1

After IDE channel 1 data transfer and its granting timer not yet expires

PRIO_ISAM > PRIO_IDE0 and PRIO_ISAM < PRIO_IDE1

After IDE channel 1 data transfer and its granting timer expires

PRIO_ISAM > PRIO_IDE0 and PRIO_ISAM > PRIO_IDE1

Note that the priority scheme is used to arbitrate bus usage when ISA master and IDE controller are competing for bus. If there is only one candidate requesting for bus at a time, it can get the bus immediately regardless of it priority.

3.10 USB Host Controller

The SiS USB Host Controller is developed to support the USB bus as the Host Controller with built-in Root Hub and 2 USB ports. The SiS USB Host Controller is implemented based on the OpenHCI, the Open Host Controller Interface Specification for USB Release 1.0.

In order to support the applications and drivers under non-USB aware environments (such as DOS environment), the SiS USB Host Controller implemented hardware to support the emulation of a PS/2 keyboard and mouse by their USB equivalents (to the USB keyboard and USB mouse). This emulation support is done by a set of registers that are controlled by code running in SMM. The hardware implementation is based on OpenHCI Legacy Support Interface Specification Release Version 1.01.

The SiS USB Host Controller provides the following major features.

- Provide USB Host Controller function to meet the Universal Serial Bus Specification version 1.0, with fully compatible to the Open Host Controller Interface Specification for USB Release 1.0
- Provide Legacy Support function based on OpenHCI Legacy Support Interface Specification Release Version 1.01.
- Built-in Root Hub, with two USB Ports integrated.
- Implement circuit and control for the Overcurrent Protection on the USB ports.

The following will be shown the USB System Block Diagram.

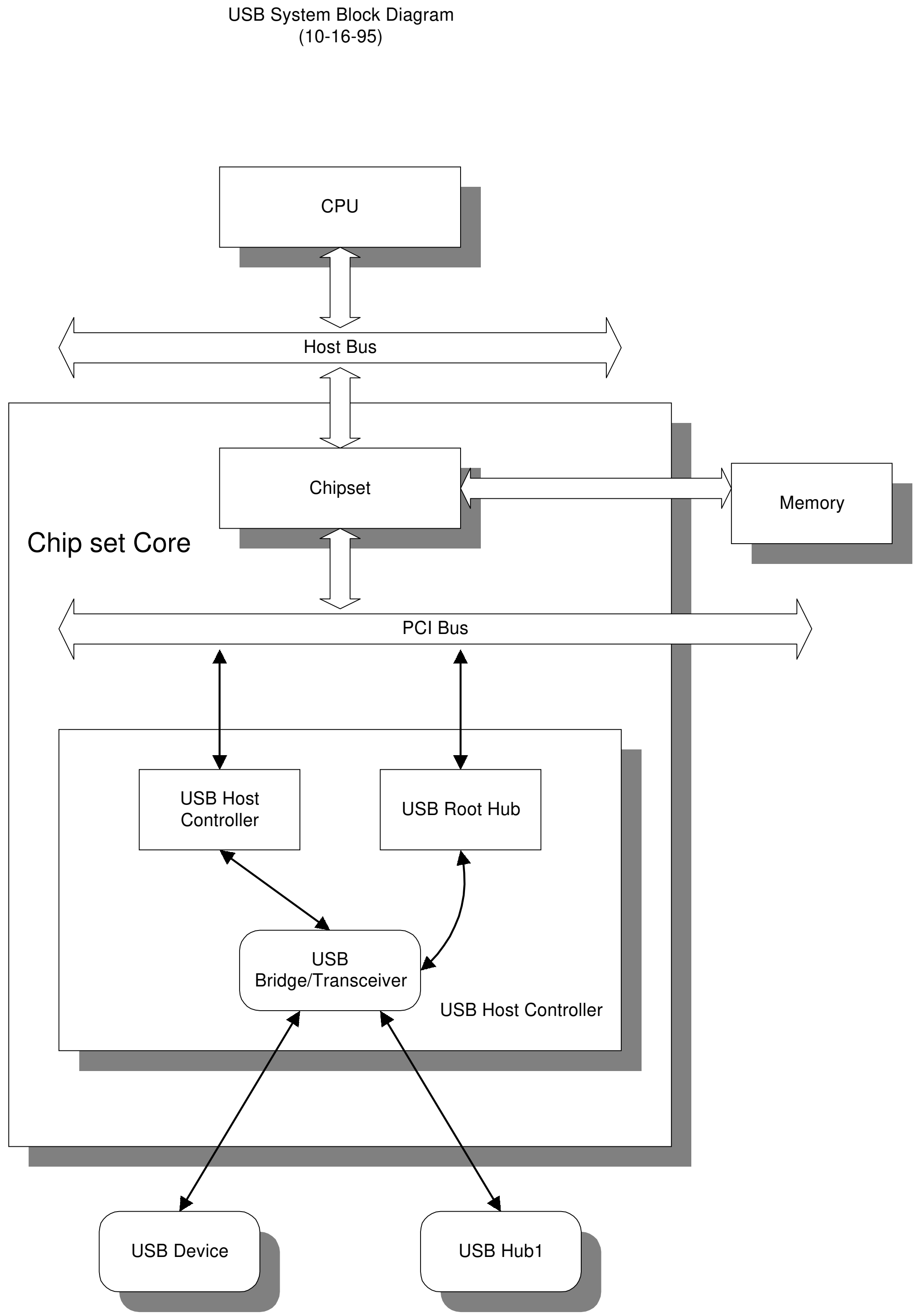


Figure 3-31 USB System Block Diagram

3.11 Integrated Keyboard Controller

The integrated KBC uses hardwired methodology instead of software implementation as the traditional 8042 keyboard BIOS. In this way, keyboard controller can have instant response to all the commands. It also supports Auto A20 gate and Auto Reset Features. Besides, the integrated KBC has a power control feature. After the [Ctrl]+[Alt]+[Backspace] hot keys are pressed, the system will enter the power saving mode. Moreover, the integrated KBC supports the industrial standard PS/2 mouse optionally.

Status Register

The status register is an 8 bits read only register located at I/O address hex 64. It has information about the state of the keyboard controller and interface. It may be read at any time.

Bit 7 **Parity Error**
0 : Odd Parity (No Parity Error)
1 : Even Parity (Parity Error)

Bit 6 **Time-out Error**
0 : No Transmission Time-out Error
1 : Transmission Time-out Error

Bit 5 **Auxiliary Output Buffer Full**
0 : Keyboard Data
1 : Mouse Data

Bit 4 **Inhibit Switch**
0 : Keyboard is Inhibited
1 : Keyboard is not Inhibited

Bit 3 **Command/Data**
0 : Data Byte. Writing to I/O 60h
1 : Command Byte. Writing to I/O 64h

Bit 2 **System Flag**
This bit may be set to 0 or 1 by writing to system flag bit in the keyboard controller's command byte. It is set to 0 after a power on reset.

Bit 1 **Input Buffer Full**
0 : Input Buffer Empty
1 : Input Buffer Full. Data has been written into the buffer but the controller has not read the data

Bit 0 **Output Buffer Full**
0 : Output Buffer Empty
1 : Output Buffer Full. The controller has placed data into its output buffer but the system has not yet read data

Input/Output Buffer

Input Buffer

The input buffer is an 8 bits write only register located at I/O address hex 60 or 64. Writing to address hex 60 sets a flag, that indicates a data write; writing to address hex 64 sets a flag, indicating a command write. Data written to I/O address hex 60 is sent to the keyboard, unless the keyboard controller is expecting a data byte following a controller command. Data should be written to the controller's input buffer only if the input buffer's full bit in the status register equal 0. The next command are valid keyboard controller commands.

Output Buffer

The output buffer is an 8 bits read only register at I/O address hex 60. The keyboard controller uses the output buffer to send scan codes received from the keyboard, and data bytes requested by command to the system. The output buffer should be read only when output buffer's full bit in the status register set to 1.

Commands (I/O Address 64H)

Write I/O Address 64h that is Keyboard BIOS Command:

<table>
<tr><th>Command</th><th>Keyboard Mode</th><th>Keyboard PS/2 Mode</th></tr>
<tr><td>00-1F</td><td colspan="2">Read Internal RAM -- The controller sends value of RAM to output buffer.</td></tr>
<tr><td>20</td><td colspan="2">Read Keyboard Controller's Command Byte -- The controller sends its current Command byte to its output buffer.</td></tr>
<tr><td>21-3F</td><td colspan="2">Read Internal RAM -- The controller sends value of RAM to output buffer.</td></tr>
<tr><td>40-5F</td><td colspan="2">Write Internal RAM -- The next byte of data written to I/O 60h is placed into Internal RAM.</td></tr>
<tr><td>60</td><td colspan="2">Write Keyboard Controller's Command Byte -- The next byte of data written to I/O 60h is placed in the controller's command byte.
Bit | Bit Definitions
0 | Enable Keyboard Output-Buffer-Full Interrupt.
Generates an interrupt when it places keyboard data into its output buffer.
1 | In Keyboard Mode: Reserved to 0.
In Keyboard PS/2 Mode:
1 -- Enable Mouse-Buffer-Full Interrupt.
Generates an interrupt when it places mouse data into its output buffer.
2 | 1 -- The controller generates an System Flag.
The value written to this bit is placed in the system flat bit of the controller's status register.
3 | 1 - Disable Keyboard Lock Switch “KBLOCK”.
4 | 1 - Disable Keyboard.
Disable the Keyboard interface by driving the 'clock' line low.
Data is not sent or received.
5 | 1 -- Disable Mouse.
Disable the mouse interface by driving the 'clock' line low.
Data is not sent or received.
6 | 1 -- IBM Personal Computer Compatibility Mode.
Convert the scan codes received from keyboard to IBM PC.
This includes converting a two-byte sequence to the one-byte IBM Personal Computer format.
7 | 0 -- Reserved.</td></tr>
<tr><td>61-7F</td><td colspan="2">Write Internal RAM -- The next byte of data written to I/O 60h is placed into Internal RAM.</td></tr>
<tr><td>A0</td><td colspan="2">Read Internal ROM -- The controller sends value to its output buffer that end with a "0".</td></tr>
<tr><td>A1</td><td colspan="2">Read Keyboard Controller's Version - The version code result will be placed to its output buffer.</td></tr>
<tr><td>A4</td><td>Reset Internal Register B to 0.</td><td>Not Valid</td></tr>
<tr><td>A5</td><td>Reset Internal Register B to 0.</td><td>Not Valid</td></tr>
<tr><td>A6</td><td>Read Internal Register B -- The controller sends value to its output buffer.</td><td>Not Valid</td></tr>
<tr><td>A7</td><td>Set Internal Register C to 0.</td><td>Disable Mouse Device -- This disable the mouse interface by driving the mouse clock line low.</td></tr>
<tr><td>A8</td><td>Set Internal Register C to 1.</td><td>Enable Mouse Device -- This enable the mouse interface by driving the mouse clock line float.</td></tr>
<tr><td>A9</td><td>Read Internal Register C -- The controller sends value to its output buffer.</td><td>Mouse Device Interface Test -- Test the controller's mouse clock and data line and place the result to output buffer as follows :

00 -- No error detected.
01 -- The 'Mouse Clock' line is stuck low.
02 -- The 'Mouse Clock' line is stuck high.
03 -- The 'Mouse Data' line is stuck low.
04 -- The 'Mouse Data' line is stuck high.</td></tr>
<tr><td>AA</td><td colspan="2">Self-Test - This commands the controller to perform internal diagnostic tests. A hex 55 is placed in the output buffer if no errors are detected.</td></tr>
</table>

AB	Keyboard Interface Test -- This commands the controller to test the keyboard clock and data line. The test result is placed in the output buffer as follows : 00 -- No error detected. 01 -- The 'Keyboard Clock' line is stuck low. 02 -- The 'Keyboard Clock' line is stuck high. 03 -- The 'Keyboard Data' line is stuck low. 04 -- The 'Keyboard Data' line is stuck high.	
AD	Disable Keyboard Feature -- This command sets bit 4 of the controller's command byte. This disable the keyboard interface by driving the clock line low. Data will not be sent or received.	
AE	Enable Keyboard Interface -- This command clears bit 4 of command byte which release the keyboard interface	
B0	Set P10 to 0	Not Valid
B1	Set P11 to 0	Not Valid
B8	Set P10 to 1 (Default)	Not Valid
B9	Set P11 to 1 (Default)	Not Valid
C0	Read Input Port -- This command the controller to read its input port and place the data in its output buffer. This command should be used only if the output buffer is empty.	
C1	Set Port P17 to 0 & KBLOCK disable	Set Port P17 to 0 & KBLOCK disable
C2	Not Valid	Place Bit 7-4 of Input Port to status register
C3	Not Valid	Place Bit 3-0 of Input Port to status register
C7	Set Port P17 to 1	Set Prot P17 to 1
CA	Read Internal Register D -- The Internal Register will be placed into its output buffer.	
CB	Write Internal Register D -- The next byte of data written to I/O 60h is placed in the controller's Register D.	
D0	Read Output Port -- This command causes the controller to read its output port and place data in its output buffer. This command should be issued only if the output buffer is empty.	
D1	Write Output Port -- The next byte of data written to I/O 60h is placed in the controller's output port.	
D2	Not Valid	Write Keyboard Output Buffer - The next byte of data written to I/O 60h is placed in output buffer as it receive from keyboard.
D3	Not Valid	Write Mouse Output Buffer - The next byte of data written to I/O 60h is placed in output buffer as it receive from mouse.
D4	Not Valid	Write Mouse Device - The next byte of data written to I/O 60h is transmitted mouse device.
D6	Enable P17(KBLOCK) Keyboard Lock Switch (Default)	
D7	Disable P17(KBLOCK) Keyboard Lock Switch, P17 define to I/O by C1 & C7 command	
F0-FF	Pulse Output Port -- Bits 0 through 3 of controller's output port may be pulsed low for approximately 6us. Bits 0 through 3 of this command indicate which bits are to be pulsed. A 0 indicates that the bit should be pulsed, and a 1 indicate the bit should not be modified.	

3.12 Multiplexed pins

Several SiS Chip I/O pins have multiple functions, the following table will provide the condition to define the pin for each function.

SiS5582 Ball No.	SiS5581 Ball No.	Pin Name	Description
N29	U29	MA0B	Set Register 57 bit 1 to “0” in Host to PCI Bridge Configuration Register
		SRAS1#	Set Register 57 bit 1 to “1” in Host to PCI Bridge Configuration Register
N25	U25	MA1B	Set Register 57 bit 1 to “0” in Host to PCI Bridge Configuration Register
		SCAS1#	Set Register 57 bit 1 to “1” in Host to PCI Bridge Configuration Register

J27	AA27	GPO3	Set ACPI/SCI Offset Register 1Ch bit 4 to “1” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		MA12	Set ACPI/SCI Offset Register 1Ch bit 4 to “0” in PCI to ISA Bridge Configuration Register
J28	AA28	GPO4	Set ACPI/SCI Offset Register 1Ch bit 5 to “1” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		MA13	Set ACPI/SCI Offset Register 1Ch bit 5 to “0” in PCI to ISA Bridge Configuration Register
J29	AA29	GPO6	Set ACPI/SCI Offset Register 1Ch bit 6 to “0” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		MA14	Set ACPI/SCI Offset Register 1C bit 6 to “1” in PCI to ISA Bridge Configuration Register
D26	AF26	GPIO7	Set Register 6Ah bit 4 to “0” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		OCO#	Set Register 6A bit 4 to “1” and 6A bit 6 to “1” in PCI to ISA Bridge Configuration Register
		OCI2#	Set Register 6A bit 4 to “1” and 6A bit 6 to “0” in PCI to ISA Bridge Configuration Register
H26	AB26	GPIO8	Set Register 6Ah bit 5 to “0” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		OCI1#	Set Register 6A bit 5 to “1” in PCI to ISA Bridge Configuration Register
D27	AF27	IOCHK#	Set ACPI/SCI Offset Register 24 bit 6 to “0” in PCI to ISA Bridge Configuration Register
		GPIO9	Set Register 40h bit7 to “1” and ACPI/SCI Offset Register 24h bit 6 to “1” in PCI to ISA Bridge Configuration Register and Set ACPI/SCI Offset Register 1C bit 9 to “1” in PCI to ISA Bridge Configuration Register.
		THRM#	Set ACPI/SCI Offset Register 24 bit 6 to “1” in PCI to ISA Bridge Configuration Register and Set ACPI/SCI Offset Register 1C bit 9 to “0” in PCI to ISA Bridge Configuration Register
B7	AH7	GPIO10	Set Auto Power Control Register II bit 1 to “0” in APC Control Registers and set Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register
		ACPILED	Set Auto Power Control Register II bit 1 to “1” in APC Control Registers
D6	AF6	OSCO	Connect PSRSTB# to Battery circuit
		RTCCS#	Pull-low resistor on PSRSTB# signal
E6	AE6	OSCI	Connect PSRSTB# to Battery circuit
		IRQ8#	Pull-low resistor on PSRSTB# signal
C8	AG8	RTCALE#	Connect PSRSTB# to Battery circuit

		ONCTL#	Pull-low resistor on PSRSTB# signal
B19	AH19	KBCLK	Set Register 70 bit 3 to “1” in PCI to ISA Bridge Configuration Register.
		GPIO2	Set Register 70 bit 3 to “1” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register, and set Register 1Dh bit 2 to ‘1’ in ACPI/SCI Offset Register.
A19	AJ19	KBDAT	Set Register 70h bit 3 to “1” in PCI to ISA Bridge Configuration Register.
		IRQ1	Set Register 70h bit 3 to “0” in PCI to ISA Bridge Configuration Register.
D19	AF19	PMCLK	Set Register 70h bit 3 to “1” and 70h bit 4 to “1” in PCI to ISA Bridge Configuration Register.
		GPIO1	Set Register 70 bit 3 to “0” and Register 40h bit7 to “1” in PCI to ISA Bridge Configuration Register, and set Register 1Dh bit 1 to ‘1’ in ACPI/SCI Offset Register.
B20	AH20	PMDAT	Set Register 70h bit 3 to “1” and 70h bit 4 to “1” in PCI to ISA Bridge Configuration Register.
		IRQ12	Set Register 70h bit 3 to “1” in PCI to ISA Bridge Configuration Register.
C20	AG20	KLOCK#	Set Register 70 bit 4 to “1” in PCI to ISA and 57 bit 0 to “1” in Host to PCI Configuration Register
		GPIO0	Set Register 70 bit 4 to “0” and 40h bit7 to “1” in PCI to ISA and 57 bit 0 to “1” in Host to PCI Bridge Configuration Register, and set Register 1Dh bit 0 to ‘1’ in ACPI/SCI Offset Register.
		RAMWC#	Set Register 57h bit 0 to “0” in Host to PCI Bridge Configuration Register.
B22	AH22	TURBO	Set Register 93 bit 2 to “0” in Host to PCI Bridge Configuration Register
		EXTSMI#	Set Register 93 bit 2 to “1” in Host to PCI Bridge Configuration Register.
N5	U5	IIRQA	If PCI IDE channel 0 operates in Native mode.
		IRQ14	If PCI IDE channel 0 operates in compatibility mode.
V5	M5	IIRQB	If PCI IDE channel 1 operates in Native mode.
		IRQ15	If PCI IDE channel 1 operates in compatibility mode.
D20	AF20	GPCS1	Set register 6D bit 6 to “0” in PCI to ISA Bridge Configuration Register.
		SIRQ	Set register 6D bit 6 to “1” in PCI to ISA Bridge Configuration Register.

3.13 Ball Connectivity Testing

SiS Chip will provide a NAND chain Test Mode. In order to ensure the connections of balls to tracks of mainboard, SiS Chip provides a simple way to do connective measurements. Basically, an additional 2-input-NAND gate is added into the I/O buffer cells. And, one of inputs of NAND gate is connected to input pin of I/O buffer as test input port in test mode. To monitor the test result at test output port, the output of the NAND gate is connected to the other input of the next NAND gate. Such that, the test result could be propagated and it forms a NAND tree, as depicted in Figure 3-32 on page 83. To adapt to the scheme, all output buffers of SiS Chip are changed to bidirection buffers to accept test signals.

3.13.1 Test Scheme

There are six NAND tree chains are provided by SiS Chip. Each NAND tree chain has several test-input pins and one output pin.

The following description is an example on 4-test-input pins to explain a NAND tree chain test scheme.

First of all, logic LOW is driven into TESTIN1 pin from track on mainboard. If logic HIGH could be observed at TESTOUT pin, it means that the connection of TESTIN1 pin to track is good, as shown in Figure 3-33 on page 83. To test TESTIN2 pin, TESTIN2 pin should be driven LOW also. And, TESTIN1 pin should be kept at logic HIGH, such that the test result could be passed to TESTOUT pin and so on. Although SiS Chip operates at 3.3V, all input buffers of SiS Chip are 5V-input tolerance. Hence, all test signal could go up to 5V.

3.13.2 Measurements

During test process, this scheme requires all test inputs to be driven simultaneously. To decrease the amount of test probes, SiS Chip divide pins into 6 branches. Meanwhile, some noise sensitive signals or analog signals, i.e. RTC, and power signals, are excluded. The final number of test-input probes is limited to 78 and these six NAND trees are listed in Table 3-2 NAND Tree List on page 83.

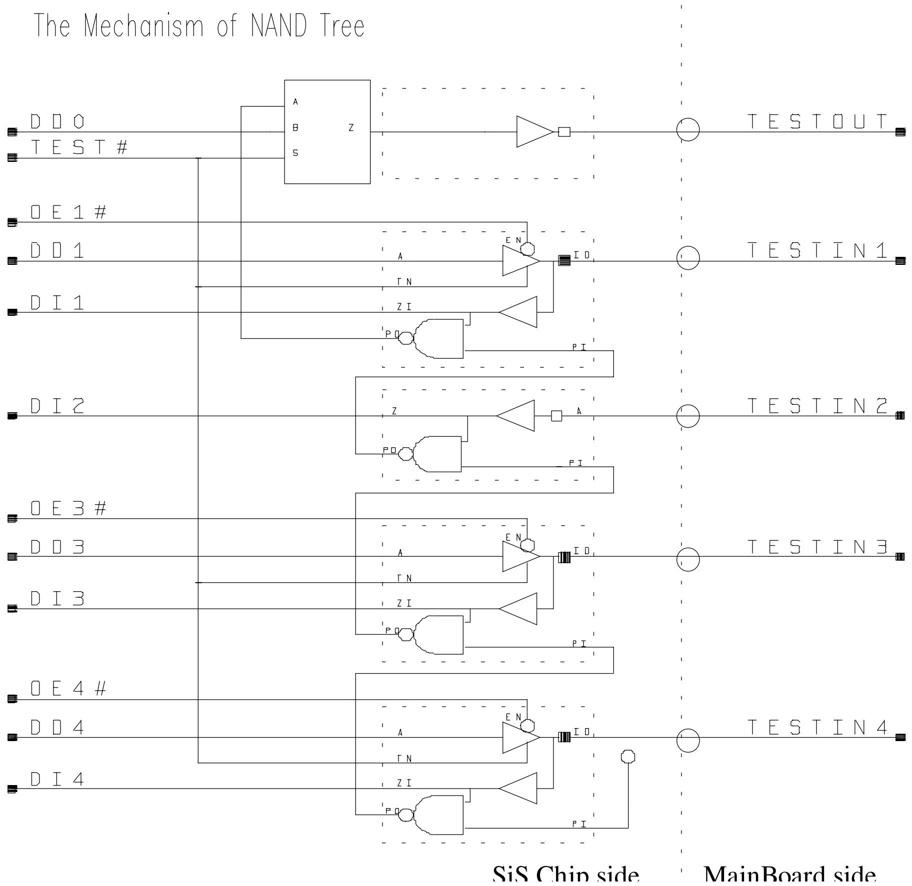


Figure 3-32

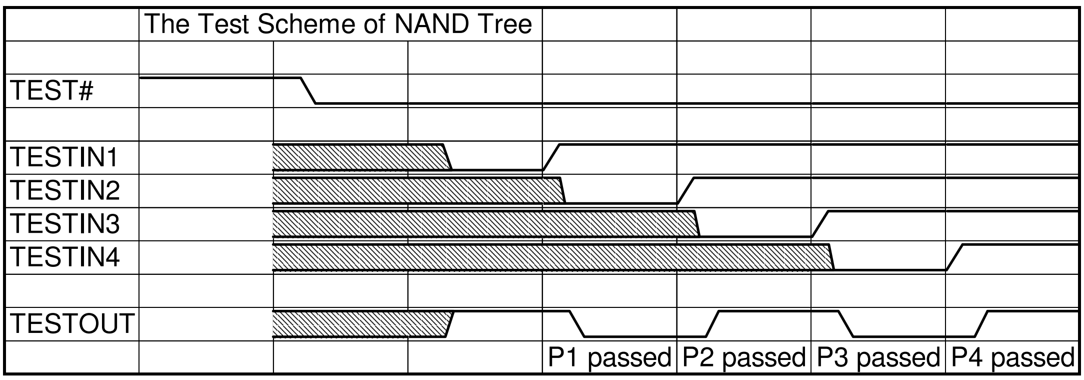


Figure 3-33

Table 3-2 NAND Tree List for SiS5581

TEST Vectors	Ball Number List	TEST Output Ball

TESTIN0[1:69] (NAND Tree 1)	N29, P26, N25, P27, P28, P24, P29, R26, P25, R27, R28, R29, T24, T26, T27, T25, T28, T29, U26, U24, U27, U28, U29, U25, V27, V28, V29, V25, W27, W28, W29, V26, Y27, Y28, Y29, W25, AA27, AA28, AA29, W26, AB27, AB28, AB29, Y25, AC26, AC27, Y26, AC28, AC29, AD27, AA25, AD28, AD29, AA26, AE26, AE27, AB25, AE28, AE29, AF26, AB26, AF27, AF28, AF29, AC25, AG28, AD26, AH28, AD25	AE05
TESTIN1[1:78] (NAND Tree 2)	B21, D19, A21, C22, B22, A22, C23, B23, A23, E24, D24, E21, C24, B24, A24, D21, E25, D25, E22, C25, B25, A25, D22, C26, B26, A26, E23, C27, D23, B27, G25, B28, C28, D26, G26, D27, D28, H25, D29, E26, E27, H26, E28, E29, F25, J25, F26, F27, F28, J26, F29, G27, K25, G28, G29, H27, H28, K26, H29, J27, J28, L25, J29, K27, K28, L26, K29, L27, M25, L28, L29, M27, M26, M28, M29, N26, N27, N28	AG05
TESTIN2[1:74] (NAND Tree 3)	C05, E09, B05, A05, E06, D09, D06, C06, B06, E10, A06, C07, B07, A07, C08, E11, B08, A08, C09, D11, B09, A09, E12, C10, B10, A10, D12, C11, B11, F13, A11, C12, B12, E13, A12, D13, C13, F14, B13, A13, E14, D14, C14, B14, A14, F16, D15, C15, B15, A15, E16, D16, C16, B16, F17, A16, D17, C17, E17, B17, A17, C18, B18, E18, A18, C19, B19, D18, A19, C20, B20, E19, A20, C21	AH05
TESTIN3[1:77] (NAND Tree 4)	T04, T03, T05, T02, T01, P06, R04, R03, R02, R01, P05, P04, P03, P02, N06, P01, N04, N03, N05, N02, N01, M03, M05, M02, M01, L03, M04, L02, L01, L05, K03, K02, L04, K01, J03, J02, K05, J01, H03, H02, K04, H01, G03, G02, J05, G01, F05, F04, J04, F03, F02, H05, F01, E04, E03, H04, E02, E01, D04, G05, D03, D02, D01, G04, C02, B02, E07, C03, D07, B03, E08, C04, B04, A04, D08, E05, D05	AG04

TESTIN4[1:63] (NAND Tree 5)	AG06, AH06, AJ06, AE09, AF05, AF09, AJ05, AE08, AH04, AF08, AJ04, AG03, AH03, AH02, AG02, AC05, AF04, AF03, AF02, AC04, AF01, AE04, AB05, AE03, AE02, AB04, AE01, AD05, AD04, AA05, AD03, AD02, AD01, AA04, AC03, AC02, AC01, AB03, Y04, AB01, AA03, W05, AA02, AA01, W04, Y03, Y02, Y01, V05, W03, W02, V04, W01, V03, U06, V02, V01, U04, U05, U03, U02, U01, T06	AH13
TESTIN5[1:63] (NAND Tree 6)	AF24, AG24, AH24, AF21, AE23, AF23, AG23, AE20, AH23, AJ23, AF20, AH22, AJ21, AG20, AF19, AH20, AJ20, AG19, AE18, AH19, AJ19, AG18, AH18, AJ18, AF17, AD17, AG17, AH17, AJ17, AE17, AF16, AG16, AD16, AH16, AJ16, AE16, AD14, AH15, AJ15, AF14, AG14, AE14, AH14, AJ14, AF13, AD13, AG13, AJ13, AE13, AG12, AH12, AJ12, AE12, AG11, AH11, AJ11, AF12, AG10, AH10, AJ10, AE11, AG09, AH09	AG15

Table 3-3 NAND Tree List for SiS5582

TEST Vectors	Ball Number List	TEST Output Ball
TESTIN0[1:69] (NAND Tree 1)	U29, T26, U25, T27, T28, T24, T29, R26, T25, R27, R28, R29, P24, P26, P27, P25, P28, P29, N26, N24, N27, N28, N29, N25, M27, M28, M29, M25, L27, L28, L29, M26, K27, K28, K29, L25, J27, J28, J29, L26, H27, H28, H29, K25, G26, G27, K26, G28, G29, F27, J25, F28, F29, J26, E26, E27, H25, E28, E29, D6, H26, D27, D28, D29, G25, C28, F26, B28, F25	E05

TESTIN1[1:78] (NAND Tree 2)	AH21, AF19, AJ21, AG22, AH22, AJ22, AG23, AH23, AJ23, AE24, AF24, AE21, AG24, AH24, AJ24, AF21, AE25, AF25, AE22, AG25, AH25, AJ25, AF22, AG26, AH26, AJ26, AE23, AG27, AF23, AH27, AC25, AH28, AG28, AF26, AC26, AF27, AF28, AB25, AF29, AE26, AE27, AB26, AE28, AE29, AD25, AA25, AD26, AD27, AD28, AA26, AD29, AC27, Y25, AC28, AC29, AB27, AB28, Y26, AB29, AA27, AA28, W25, AA29, Y27, Y28, W26, Y29, W27, V25, W28, W29, V27, V26, V28, V29, U26, U27, U28	C05
TESTIN2[1:74] (NAND Tree 3)	AG05, AE09, AH05, AJ05, AE06, AF09, AF06, AG06, AH06, AE10, AJ06, AG07, AH07, AJ07, AG08, AE11, AH08, AJ08, AG09, AF11, AH09, AJ09, AE12, AG10, AH10, AJ10, AF12, AG11, AH11, AD13, AJ11, AG12, AH12, AE13, AJ12, AF13, AG13, AD14, AH13, AJ13, AE14, AF14, AG14, AH14, AJ14, AD16, AF15, AG15, AH15, AJ15, AE16, AF16, AG16, AH16, AD17, AJ16, AF17, AG17, AE17, AH17, AJ17, AG18, AH18, AE18, AJ18, AG19, AH19, AF18, AJ19, AG20, AH20, AE19, AJ20, AG21	B05
TESTIN3[1:77] (NAND Tree 4)	P04, P03, P05, P02, P01, T06, R04, R03, R02, R01, T05, T04, T03, T02, U06, T01, U04, U03, U05, U02, U01, V03, V05, V02, V01, W03, V04, W02, W01, W05, Y03, Y02, W04, Y01, AA03, AA02, Y05, AA01, AB03, AB02, Y04, AB01, AC03, AC02, AA05, AC01, AD05, AD04, AA04, AD03, AD02, AB05, AD01, AE04, AE03, AB04, AE02, AE01, AF04, AC05, AF03, AF02, AF01, AC04, AG02, AH02, AE07, AG03, AF07, AH03, AE08, AG04, AH04, AJ04, AF08, AE05, AF05	C04

TESTIN4[1:63] (NAND Tree 5)	C06, B06, A06, E09, D05, D09, A05, E08, B04, D08, A04, C03, B03, B02, C02, G05, D04, D03, D02, G04, D01, E04, H05, E03, E02, H04, E01, F05, F04, J05, F03, F02, F01, J04, G03, G02, G01, H03, K04, H01, J03, L05, J02, J01, L04, K03, K02, K01, M05, L03, L02, M04, L01, M03, N06, M02, M01, N04, N05, N03, N02, N01, P06	B13
TESTIN5[1:63] (NAND Tree 6)	D24, C24, B24, D21, E23, D23, C23, E20, B23, A23, D20, B22, A21, C20, D19, B20, A20, C19, E18, B19, A19, C18, B18, A18, D17, F17, C17, B17, A17, E17, D16, C16, F16, B16, A16, E16, F14, B15, A15, D14, C14, E14, B14, A14, D13, F13, C13, A13, E13, C12, B12, A12, E12, C11, B11, A11, D12, C10, B10, A10, E11, C09, B09	C15

4. Pin Assignment and Description

To suit all kinds of PC main board form factor, SiS Chip provides two kinds of pin assignment for customized boards design.
SiS5582 pin assignment is based on ATX/traditional Baby-AT form factor, and SiS5581 is based on LPX/NPX/NLX form factor.

4.1 SiS5582 Pin Assignment(Top view)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
A			NC	AD30	PREQ2#	INTC#	RTCVDD	PSRSTB#	PWRBT#	DACK2	DREQ5	IRQ10	OSC	IRQ5	IOW#	ZWS#	SA7	SA2	KBDAT	SPK	UCLK48	UV1-	NC	NC	NC	NC	NC			**A**
B		PCICLK	AD28	C/BE3#	PGNT2#	INTB#	GPIO10	RING#	BCLK	DACK1	MEMW#	IO16#	BALE	IRQ6	SMEMR#	DREQ3	SA6	SA1	KBCLK	PMDAT	UV0+	TURBO	NC	NC	DVSS	DVSS	DVSS	PGNT4#		**B**
C	NC	AD27	AD29	PGNT3#	PGNT1#	INTA#	PWRGD	ONCTL#	MR16#	DACK0	MEMR#	M16#	TC	IRQ9	AEN	DREQ1	SA5	SA0	ROMKBCS#	KLOCK#	UV1+	TEST#	NC	NC	DVSS	DVSS	DVSS	NC	NC	**C**
D	AD21	AD23	AD24	AD25	PREQ0#	OCSO	DLLVDD1	AD31	PREQ1#	RTCVSS	SWITCH	DREQ6	IRQ4	IOR#	VCC5	DREQ0	SA3	PVDD	PMCLK	SIRQ	NC	NC	NC	NC	DVSS	GPIO7	IOCHK#	SRAS2#	SCAS2#	**D**
E	C/BE2#	AD17	AD18	AD20	PGNT0#	OCSI	DLLVSS1	PREQ3#	INTD#	GPIO5	DREQ7	IRQ11	SHBE#	IRQ7	OVDD	IORDY	RFH#	GPCS0	UV0-	NC	DVSS	DVSS	NC	DVSS	DVSS	CAS5#	CAS6#	DDCDAT1	DDCCLK1	**E**
F	PLOCK#	STOP#	DEVSEL#	IRDY#	FRAME#								IRQ3	SMEMW#	OVDD	DREQ2	SA4								PREQ4#	NC	CAS0#	CAS2#	CAS3#	**F**
G	AD15	C/BE1#	PAR	AD22	AD26																				NC	RAS4#	RAS0#	RAS2#	RAS3#	**G**
H	AD12	VCC5	AD14	AD16	AD19																				CAS7#	GPIO8	RAMWA#	SCAS0#	SRAS0#	**H**
J	AD8	AD9	AD11	SERR#	TRDY#																				CAS1#	CAS4#	MA12	MA13	MA14	**J**
K	AD5	AD6	AD7	AD13	PVDD							**OVDD**	**OVDD**				**OVDD**	**OVDD**							RAS5#	RAS1#	MA8	MA9	MA10	**K**
L	AD0	AD2	AD3	C/BE0#	AD10																				MA11	RAMWB#	MA4	MA5	MA6	**L**
M	IDSAA1	IDSAA0	SDOEH	AD1	AD4					**OVDD**		GND	GND	GND	GND	GND	GND	GND		**OVDD**					MA3	MA7	MA0A	MA1A	MA2	**M**
N	ICHRDYA	IDREQA	IDACKA#	IDSAA2	IIRQA	**SDOEL**				**OVDD**		GND	GND	GND	GND	GND	GND	GND		**OVDD**				**MD2**	MA1B	MD3	MD1	MD0	MA0B	**N**
P	IDA1	IDA0	ICSA0#	IIORA#	ICSA1#	**IIOWA#**						GND	GND	GND	GND	GND	GND	GND						**MD9**	MD6	MD8	MD7	MD5	MD4	**P**
R	IDA6	IDA5	IDA4	IDA3	OVDD	**OVDD**						GND	GND	GND	GND	GND	GND	GND						**OVDD**	OVDD	MD14	MD12	MD11	MD10	**R**
T	IDA12	IDA10	IDA9	IDA8	IDA7	**IDA2**						GND	GND	GND	GND	GND	GND	GND						**MD16**	MD13	MD20	MD18	MD17	MD15	**T**
U	IDSBA1	IDSBA0	IDA14	IDA13	IDA15	**IDA11**				**OVDD**		GND	GND	GND	GND	GND	GND	GND		**OVDD**				**PVDD**	MD19	MD24	MD23	MD22	MD21	**U**
V	IDRQB	IDACKB#	IDSBA2	IIOWB#	IIRQB					**OVDD**		GND	GND	GND	GND	GND	GND	GND		**OVDD**					MD31	MD27	MD28	MD26	MD25	**V**
W	ICSB0#	IIORB#	ICHRDYB	IDB13	ICSB1#																				MD38	MD34	MD32	MD30	MD29	**W**
Y	IDB12	IDB14	IDB15	IDB3	IDB9							**OVDD**	**OVDD**				**OVDD**	**OVDD**							MD47	MD42	MD36	MD35	MD33	**Y**
AA	IDB0	IDB10	IDB11	HA29	IDB7																				MD54	MD50	MD40	MD39	MD37	**AA**
AB	IDB4	IDB2	IDB1	HA22	HA26																				MD62	MD58	MD44	MD43	MD41	**AB**
AC	IDB8	IDB6	IDB5	HA14	HA18																				TA3	TA7	MD48	MD46	MD45	**AC**
AD	HA25	HA27	HA28	HA30	HA31								HD31	HD39	OVDD	HD47	HD56								MD55	MD53	MD52	MD51	MD49	**AD**
AE	HA20	HA21	HA23	HA24	HD0	HD6	HA11	HA7	HD3	HD11	HD17	HD24	HD35	HD42	OVDD	HD52	HD60	HBE0#	CPURST	DLLVSS0	AHOLD	D/C#	KOE#	FLUSH#	BRDY#	MD60	MD59	MD57	MD56	**AE**
AF	HA15	HA16	HA17	HA19	HD1	HD8	HA9	HA3	HD7	PVDD	HD21	HD28	HD37	HD43	HD48	HD53	HD58	HBE4#	STPCLK#	DLLVDD0	CACHE#	ADSC#	TA1	HLOCK#	W/R#	TA6	TAGW#	MD63	MD61	**AF**
AG	NC	HA13	HA10	HA6	HD2	HD9	HD13	HD16	HD20	HD25	HD29	HD33	HD38	HD44	HD49	HD54	HD59	HD63	HBE2#	HBE6#	NMI	SMIACT#	CPUCLK	BOFF#	M/IO#	BWE#	TA0	TA5	NC	**AG**
AH		HA12	HA8	HA5	HD4	HD10	HD14	HD18	HD22	HD26	HD30	HD34	HD40	HD45	HD50	HD55	HD61	FERR#	HBE3#	HBE7#	INTR	SMI#	HITM#	NA#	ADS#	GWE#	TA2	TA4		**AH**
AJ			NC	HA4	HD5	HD12	HD15	HD19	HD23	HD27	HD32	HD36	HD41	HD46	HD51	HD57	HD62	HBE1#	HBE5#	INIT	A20M#	IGNNE#	EADS#	KEN#	ADSV#	CCS1#	NC			**AJ**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	

4.2 SiS5581 Pin Assignment(Top view)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
A			NC	HA4	HD5	HD12	HD15	HD19	HD23	HD27	HD32	HD36	HD41	HD46	HD51	HD57	HD62	HBE1#	HBE5#	INIT	A20M#	IGNNE#	EADS#	KEN#	ADSV#	CCS1#	NC			A
B		HA12	HA8	HA5	HD4	HD10	HD14	HD18	HD22	HD26	HD30	HD34	HD40	HD45	HD50	HD55	HD61	FERR#	HBE3#	HBE7#	INTR	SMI#	HITM#	NA#	ADS#	GWE#	TA2	TA4		B
C	NC	HA13	HA10	HA6	HD2	HD9	HD13	HD16	HD20	HD25	HD29	HD33	HD38	HD44	HD49	HD54	HD59	HD63	HBE2#	HBE6#	NMI	SMIACT#	CPUCLK	BOFF#	M/IO#	BWE#	TA0	TA5	NC	C
D	HA15	HA16	HA17	HA19	HD1	HD8	HA9	HA3	HD7	PVDD	HD21	HD28	HD37	HD43	HD48	HD53	HD58	HBE4#	STPCLK#	DLLVDD0	CACHE#	ADSC#	TA1	HLOCK#	W/R#	TA6	TAGW#	MD63	MD61	D
E	HA20	HA21	HA23	HA24	HD0	HD6	HA11	HA7	HD3	HD11	HD17	HD24	HD35	HD42	OVDD	HD52	HD60	HBE0#	CPURST	DLLVSS0	AHOLD	D/C#	KOE#	FLUSH	BRDY#	MD60	MD59	MD57	MD56	E
F	HA25	HA27	HA28	HA30	HA31								HD31	HD39	OVDD	HD47	HD56								MD55	MD53	MD52	MD51	MD49	F
G	IDB8	IDB6	IDB5	HA14	HA18																				TA3	TA7	MD48	MD46	MD45	G
H	IDB4	IDB2	IDB1	HA22	HA26																				MD62	MD58	MD44	MD43	MD41	H
J	IDB0	IDB10	IDB11	HA29	IDB7																				MD54	MD50	MD40	MD39	MD37	J
K	IDB12	IDB14	IDB15	IDB3	IDB9							OVDD	OVDD				OVDD	OVDD							MD47	MD42	MD36	MD35	MD33	K
L	ICSB0#	IIORB#	ICHRDYB	IDB13	ICSB1#																				MD38	MD34	MD32	MD30	MD29	L
M	IDREQB	IDACKB#	IDSBA2	IIOWB#	IIRQB					OVDD		GND	GND	GND	GND	GND	GND	GND		OVDD					MD31	MD27	MD28	MD26	MD25	M
N	IDSBA1	IDSBA0	IDA14	IDA13	IDA15	IDA11				OVDD		GND	GND	GND	GND	GND	GND	GND		OVDD				PVDD	MD19	MD24	MD23	MD22	MD21	N
P	IDA12	IDA10	IDA9	IDA8	IDA7	IDA2						GND	GND	GND	GND	GND	GND	GND						MD16	MD13	MD20	MD18	MD17	MD15	P
R	IDA6	IDA5	IDA4	IDA3	OVDD	OVDD						GND	GND	GND	GND	GND	GND	GND						OVDD	OVDD	MD14	MD12	MD11	MD10	R
T	IDA1	IDA0	ICSA0#	IIORA#	ICSA1#	IIOWA#						GND	GND	GND	GND	GND	GND	GND						MD9	MD6	MD8	MD7	MD5	MD4	T
U	ICHRDYA	IDREQA	IDACKA#	IDSAA2	IIRQA	SDOEL				OVDD		GND	GND	GND	GND	GND	GND	GND		OVDD				MD2	MA1B	MD3	MD1	MD0	MA0B	U
V	IDSAA1	IDSAA0	SDOEH	AD1	AD4					OVDD		GND	GND	GND	GND	GND	GND	GND		OVDD					MA3	MA7	MA0A	MA1A	MA2	V
W	AD0	AD2	AD3	C/BE0#	AD10																				MA11	RAMWB#	MA4	MA5	MA6	W
Y	AD5	AD6	AD7	AD13	PVDD							OVDD	OVDD				OVDD	OVDD							RAS5#	RAS1#	MA8	MA9	MA10	Y
AA	AD8	AD9	AD11	SERR#	TRDY#																				CAS1#	CAS4#	MA12	MA13	MA14	AA
AB	AD12	VCC5	AD14	AD16	AD19																				CAS7#	GPIO8	RAMWA#	SCAS0#	SRAS0#	AB
AC	AD15	C/BE1#	PAR	AD22	AD26																				NC	RAS4#	RAS0#	RAS2#	RAS3#	AC
AD	PLOCK#	STOP#	DEVSEL#	IRDY#	FRAME#								IRQ3	SMEMW#	OVDD	DREQ2	SA4								PREQ4#	NC	CAS0#	CAS2#	CAS3#	AD
AE	C/BE2#	AD17	AD18	AD20	PGNT0#	OSCI	DLLVSS1	PREQ3#	INTD#	GPIO5	DREQ7	IRQ11	SHBE#	IRQ7	OVDD	IORDY	RFH#	GPCS0	UV0-	NC	DVSS	DVSS	NC	DVSS	DVSS	CAS5#	CAS6#	DDCDAT1	DDCCLK1	AE
AF	AD21	AD23	AD24	AD25	PREQ0#	OCSO	DLLVDD1	AD31	PREQ1#	RTCVSS	SWITCH	DREQ6	IRQ4	IOR#	VCC5	DREQ0	SA3	PVDD	PMCLK	SIRQ	NC	NC	NC	NC	DVSS	GPIO7	IOCHK#	SRAS2#	SCAS2#	AF
AG	NC	AD27	AD29	PGNT3#	PGNT1#	INTA#	PWRGD	ONCTL#	MR16#	DACK0	MEMR#	M16#	TC	IRQ9	AEN	DREQ1	SA5	SA0	ROMKBCS#	KLOCK#	UV1+	TEST#	NC	NC	DVSS	DVSS	DVSS	NC	NC	AG
AH		PCICLK	AD28	C/BE3#	PGNT2#	INTB#	GPIO10	RING#	BCLK	DACK1	MEMW#	IO16	BALE	IRQ6	SMEMR#	DREQ3	SA6	SA1	KBCLK	PMDAT	UV0+	TURBO	NC	NC	DVSS	DVSS	DVSS	PGNT4#		AH
AJ			NC	AD30	PREQ2#	INTC#	RTCVDD	PSRSTB#	PWRBT#	DACK2	DREQ5	IRQ10	OSC	IRQ5	IOW#	ZWS#	SA7	SA2	KBDAT	SPK	UCLK48	UV1-	NC	NC	NC	NC	NC			AJ
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	

4.3 SiS Chip Alphabetical Pin List

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
A20M#	AJ21	A21	BOFF#	AG24	C24
AD0	L01	W01	BRDY#	AE25	E25
AD1	M04	V04	BWE#	AG26	C26
AD2	L02	W02	C/BE0#	L04	W04
AD3	L03	W03	C/BE1#	G02	AC02
AD4	M05	V05	C/BE2#	E01	AE01
AD5	K01	Y01	C/BE3#	B04	AH04
AD6	K02	Y02	CACHE#	AF21	D21
AD7	K03	Y03	CAS0#/DQM0	F27	AD27
AD8	J01	AA01	CAS1#/DQM1	J25	AA25
AD9	J02	AA02	CAS2#/DQM2	F28	AD28
AD10	L05	W05	CAS3#/DQM3	F29	AD29
AD11	J03	AA03	CAS4#/DQM4	J26	AA26
AD12	H01	AB01	CAS5#/DQM5	E26	AE26
AD13	K04	Y04	CAS6#/DQM6	E27	AE27
AD14	H03	AB03	CAS7#/DQM7	H25	AB25
AD15	G01	AC01	CCS1#	AJ26	A26
AD16	H04	AB04	CPUCLK	AG23	C23
AD17	E02	AE02	CPURST	AE19	E19
AD18	E03	AE03	D/C#	AE22	E22
AD19	H05	AB05	DACK0#	C10	AG10
AD20	E04	AE04	DACK1#	B10	AH10
AD21	D01	AF01	DACK2#	A10	AJ10
AD22	G04	AC04	DDCCLK1	E29	AE29
AD23	D02	AF02	DDCDAT1	E28	AE28
AD24	D03	AF03	DEVSEL#	F03	AD03
AD25	D04	AF04	DLLVDD0	AF20	D20
AD26	G05	AC05	DLLVDD1	D07	AF07
AD27	C02	AG02	DLLVSS0	AE20	E20
AD28	B03	AH03	DLLVSS1	E07	AE07
AD29	C03	AG03	DREQ0	D16	AF16
AD30	A04	AJ04	DREQ1	C16	AG16
AD31	D08	AF08	DREQ2	F16	AD16
ADS#	AH25	B25	DREQ3	B16	AH16
ADSC#	AF22	D22	DREQ5	A11	AJ11
ADSV#	AJ25	A25	DREQ6	D12	AF12
AEN	C15	AG15	DREQ7	E11	AE11
AHOLD	AE21	E21	DVDD	AD15	F15
BALE	B13	AH13	DVDD	AE15	E15
BCLK	B09	AH09	DVDD	E15	AE15

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
DVDD	F15	AD15	GND	R12	R12
DVDD	R05	R05	GND	R13	R13
DVDD	R06	R06	GND	R14	R14
DVDD	R24	R24	GND	R15	R15
DVDD	R25	R25	GND	R16	R16
DVSS	B25	AH25	GND	R17	R17
DVSS	B26	AH26	GND	R18	R18
DVSS	B27	AH27	GND	T12	P12
DVSS	C25	AG25	GND	T13	P13
DVSS	C26	AG26	GND	T14	P14
DVSS	C27	AG27	GND	T15	P15
DVSS	D25	AF25	GND	T16	P16
DVSS	E21	AE21	GND	T17	P17
DVSS	E24	AE24	GND	T18	P18
DVSS	E25	AE25	GND	U12	N12
DVSS	E22	AE22	GND	U13	N13
EADS#	AJ23	A23	GND	U14	N14
FERR#	AH18	B18	GND	U15	N15
FLUSH#	AE24	E24	GND	U16	N16
FRAME#	F05	AD05	GND	U17	N17
GND	M12	V12	GND	U18	N18
GND	M13	V13	GND	V12	M12
GND	M14	V14	GND	V13	M13
GND	M15	V15	GND	V14	M14
GND	M16	V16	GND	V15	M15
GND	M17	V17	GND	V16	M16
GND	M18	V18	GND	V17	M17
GND	N12	U12	GND	V18	M18
GND	N13	U13	GPCS0	E18	AE18
GND	N14	U14	GPIO5	E10	AE10
GND	N15	U15	GPIO7/OCO#	D26	AF26
GND	N16	U16	GPIO8/OCI#	H26	AB26
GND	N17	U17	GPIO9/THRM#/IOCHK#	D27	AF27
GND	N18	U18	GPIO10/ACPILED	B07	AH07
GND	P12	T12	GWE#	AH26	B26
GND	P13	T13	HA03	AF08	D08
GND	P14	T14	HA04	AJ04	A04
GND	P15	T15	HA05	AH04	B04
GND	P16	T16	HA06	AG04	C04
GND	P17	T17	HA07	AE08	E08
GND	P18	T18	HA08	AH03	B03

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
HA09	AF07	D07	HD09	AG06	C06
HA10	AG03	C03	HD10	AH06	B06
HA11	AE07	E07	HD11	AE10	E10
HA12	AH02	B02	HD12	AJ06	A06
HA13	AG02	C02	HD13	AG07	C07
HA14	AC04	G04	HD14	AH07	B07
HA15	AF01	D01	HD15	AJ07	A07
HA16	AF02	D02	HD16	AG08	C08
HA17	AF03	D03	HD17	AE11	E11
HA18	AC05	G05	HD18	AH08	B08
HA19	AF04	D04	HD19	AJ08	A08
HA20	AE01	E01	HD20	AG09	C09
HA21	AE02	E02	HD21	AF11	D11
HA22	AB04	H04	HD22	AH09	B09
HA23	AE03	E03	HD23	AJ09	A09
HA24	AE04	E04	HD24	AE12	E12
HA25	AD01	F01	HD25	AG10	C10
HA26	AB05	H05	HD26	AH10	B10
HA27	AD02	F02	HD27	AJ10	A10
HA28	AD03	F03	HD28	AF12	D12
HA29	AA04	J04	HD29	AG11	C11
HA30	AD04	F04	HD30	AH11	B11
HA31	AD05	F05	HD31	AD13	F13
HBE0#	AE18	E18	HD32	AJ11	A11
HBE1#	AJ18	A18	HD33	AG12	C12
HBE2#	AG19	C19	HD34	AH12	B12
HBE3#	AH19	B19	HD35	AE13	E13
HBE4#	AF18	D18	HD36	AJ12	A12
HBE5#	AJ19	A19	HD37	AF13	D13
HBE6#	AG20	C20	HD38	AG13	C13
HBE7#	AH20	B20	HD39	AD14	F14
HD0	AE05	E05	HD40	AH13	B13
HD1	AF05	D05	HD41	AJ13	A13
HD02	AG05	C05	HD42	AE14	E14
HD03	AE09	E09	HD43	AF14	D14
HD04	AH05	B05	HD44	AG14	C14
HD05	AJ05	A05	HD45	AH14	B14
HD06	AE06	E06	HD46	AJ14	A14
HD07	AF09	D09	HD47	AD16	F16
HD08	AF06	D06	HD48	AF15	D15

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
HD49	AG15	C15	IDACKB#	V02	M02
HD50	AH15	B15	IDB0/SA8	AA01	J01
HD51	AJ15	A15	IDB1/SA9	AB03	H03
HD52	AE16	E16	IDB2/SA10	AB02	H02
HD53	AF16	D16	IDB3/SA11	Y04	K04
HD54	AG16	C16	IDB4/SA12	AB01	H01
HD55	AH16	B16	IDB5/SA13	AC03	G03
HD56	AD17	F17	IDB6/SA14	AC02	G02
HD57	AJ16	A16	IDB7/SA15	AA05	J05
HD58	AF17	D17	IDB8/SA16	AC01	G01
HD59	AG17	C17	IDB9/LA17	Y05	K05
HD60	AE17	E17	IDB10/LA18	AA02	J02
HD61	AH17	B17	IDB11/LA19	AA03	J03
HD62	AJ17	A17	IDB12/LA20	Y01	K01
HD63	AG18	C18	IDB13/LA21	W04	L04
HITM#	AH23	B23	IDB14/LA22	Y02	K02
HLOCK#	AF24	D24	IDB15/LA23	Y03	K03
ICHRDYA	N01	U01	IDREQA	N02	U02
ICHRDYB	W03	L03	IDREQB	V01	M01
ICSA0#	P03	T03	IDSAA0	M02	V02
ICSA1#	P05	T05	IDSAA1	M01	V01
ICSB0#	W01	L01	IDSAA2	N04	U04
ICSB1#	W05	L05	IDSBA0	U02	N02
IDA0/SD0	P02	T02	IDSBA1	U01	N01
IDA1/SD1	P01	T01	IDSBA2	V03	M03
IDA2/SD2	T06	P06	IGNNE#	AJ22	A22
IDA3/SD3	R04	R04	IIORA#	P04	T04
IDA4/SD4	R03	R03	IIORB#	W02	L02
IDA5/SD5	R02	R02	IIOWA#	P06	T06
IDA6/SD6	R01	R01	IIOWB#	V04	M04
IDA7/SD7	T05	P05	IIRQA/IRQ14	N05	U05
IDA8/SD8	T04	P04	IIRQB/IRQ15	V05	M05
IDA9/SD9	T03	P03	INIT	AJ20	A20
IDA10/SD10	T02	P02	INTA#	C06	AG06
IDA11/SD11	U06	N06	INTB#	B06	AH06
IDA12/SD12	T01	P01	INTC#	A06	AJ06
IDA13/SD13	U04	N04	INTD#	E09	AE09
IDA14/SD14	U03	N03	INTR	AH21	B21
IDA15/SD15	U05	N05	IO16#	B12	AH12
IDACKA#	N03	U03	IOR#	D14	AF14

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
IORDY	E16	AE16	MD05	P28	T28
IOW#	A15	AJ15	MD06	P25	T25
IRDY#	F04	AD04	MD07	P27	T27
IRQ03	F13	AD13	MD08	P26	T26
IRQ04	D13	AF13	MD09	P24	T24
IRQ05	A14	AJ14	MD10	R29	R29
IRQ06	B14	AH14	MD11	R28	R28
IRQ07	E14	AE14	MD12	R27	R27
IRQ09	C14	AG14	MD13	T25	P25
IRQ10	A12	AJ12	MD14	R26	R26
IRQ11	E12	AE12	MD15	T29	P29
KBCLK/GPIO2	B19	AH19	MD16	T24	P24
KBDAT/IRQ1	A19	AJ19	MD17	T28	P28
KEN#/INV	AJ24	A24	MD18	T27	P27
KLOCK#/GPIO0	C20	AG20	MD19	U25	N25
KOE#	AE23	E23	MD20	T26	P26
M/IO#	AG25	C25	MD21	U29	N29
M16#	C12	AG12	MD22	U28	N28
MA02	M29	V29	MD23	U27	N27
MA03	M25	V25	MD24	U26	N26
MA04	L27	W27	MD25	V29	M29
MA05	L28	W28	MD26	V28	M28
MA06	L29	W29	MD27	V26	M26
MA07	M26	V26	MD28	V27	M27
MA08	K27	Y27	MD29	W29	L29
MA09	K28	Y28	MD30	W28	L28
MA0A	M27	V27	MD31	V25	M25
MA0B/SRAS1#	N29	U29	MD32	W27	L27
MA10	K29	Y29	MD33	Y29	K29
MA11	L25	W25	MD34	W26	L26
MA12/GPO3	J27	AA27	MD35	Y28	K28
MA13/GPO4	J28	AA28	MD36	Y27	K27
MA14/GPO6	J29	AA29	MD37	AA29	J29
MA1A	M28	V28	MD38	W25	L25
MA1B/SCAS1#	N25	U25	MD39	AA28	J28
MD0	N28	U28	MD40	AA27	J27
MD1	N27	U27	MD41	AB29	H29
MD02	N24	U24	MD42	Y26	K26
MD03	N26	U26	MD43	AB28	H28
MD04	P29	T29	MD44	AB27	H27

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
MD45	AC29	G29	NC	D21	AF24
MD46	AC28	G28	NC	D22	AF22
MD47	Y25	K25	NC	D23	AG24
MD48	AC27	G27	NC	D24	AH24
MD49	AD29	F29	NC	E20	AE20
MD50	AA26	J26	NC	A03	AF21
MD51	AD28	F28	NC	E23	AE23
MD52	AD27	F27	NC	F26	AF23
MD53	AD26	F26	NC	G25	AG23
MD54	AA25	J25	NMI	AG21	C21
MD55	AD25	F25	ONCTL#/RTCALE	C08	AG08
MD56	AE29	E29	OSC	A13	AJ13
MD57	AE28	E28	OSCI/IRQ8#	E06	AE06
MD58	AB26	H26	OSCO/RTCCS#	D06	AF06
MD59	AE27	E27	OVDD	K12	Y12
MD60	AE26	E26	OVDD	K13	Y13
MD61	AF29	D29	OVDD	K17	Y17
MD62	AB25	H25	OVDD	K18	Y18
MD63	AF28	D28	OVDD	M10	V10
MEMR#	C11	AG11	OVDD	M20	V20
MEMW#	B11	AH11	OVDD	N10	U10
MR16#	C09	AG09	OVDD	N20	U20
NA#	AH24	B24	OVDD	U10	N10
NC	A23	AJ27	OVDD	U20	N20
NC	A24	C01	OVDD	V10	M10
NC	A25	C29	OVDD	V20	M20
NC	A26	A03	OVDD	Y12	K12
NC	A27	A27	OVDD	Y13	K13
NC	AG01	AG01	OVDD	Y17	K17
NC	AG29	AG29	OVDD	Y17	K17
NC	AJ03	AJ03	PAR	G03	AC03
NC	AJ27	AJ23	PCICLK	B02	AH02
NC	B23	AG28	PGNT0#	E05	AE05
NC	B24	AJ24	PGNT1#	C05	AG05
NC	B28	AJ25	PGNT2#	B05	AH05
NC	C01	AD26	PGNT3#	C04	AG04
NC	C23	AC25	PGNT4#	C24	AH28
NC	C28	AH23	PLOCK#	F01	AD01
NC	C29	AJ26	PMCLK/GPIO1	D19	AF19

Signal Name	SiS5582 Ball No.	SiS5581 Ball No.	Signal Name	SiS5582 Ball No.	SiS5581 Ball No.
PMDAT/IRQ12	B20	AH20	SDOEL	N06	U06
PREQ0#	D05	AF05	SERR#	J04	AA04
PREQ1#	D09	AF09	SHBE#	E13	AE13
PREQ2#	A05	AJ05	SIRQ/GPCS1	D20	AF20
PREQ3#	E08	AE08	SMEMR#	B15	AH15
PREQ4#	F25	AD25	SMEMW#	F14	AD14
PSRSTB#	A08	AJ08	SMI#	AH22	B22
PVDD	AF10	D10	SMIACT#	AG22	C22
PVDD	D18	AF18	SPK	A20	AJ20
PVDD	K05	Y05	SRAS0#	H29	AB29
PVDD	U24	N24	SRAS2#	D28	AF28
PWRBT#	A09	AJ09	STOP#	F02	AD02
PWRGD	C07	AG07	STPCLK#	AF19	D19
RAMWA#	H27	AB27	SWITCH	D11	AF11
RAMWB#	L26	W26	TA0	AG27	C27
RAS0#/CS0#	G27	AC27	TA1	AF23	D23
RAS1#/CS1#	K26	Y26	TA2	AH27	B27
RAS2#/CS2#	G28	AC28	TA3	AC25	G25
RAS3#/CS3#	G29	AC29	TA4	AH28	B28
RAS4#CS4#	G26	AC26	TA5	AG28	C28
RAS5#CS5#	K25	Y25	TA6	AF26	D26
RFH#	E17	AE17	TA7	AC26	G26
RING#	B08	AH08	TAGW#	AF27	D27
ROMKBCS#	C19	AG19	TC	C13	AG13
RTCVDD	A07	AJ07	TEST#	C22	AG22
RTCVSS	D10	AF10	TRDY#	J05	AA05
SA0	C18	AG18	TURBO/EXTSMI#	B22	AH22
SA1	B18	AH18	UCLK48	A21	AJ21
SA2	A18	AJ18	UV0+	B21	AH21
SA3	D17	AF17	UV0-	E19	AE19
SA4	F17	AD17	UV1+	C21	AG21
SA5	C17	AG17	UV1-	A22	AJ22
SA6	B17	AH17	VCC5	D15	AF15
SA7	A17	AJ17	VCC5	H02	AB02
SCAS0#	H28	AB28	W/R#	AF25	D25
SCAS2#	D29	AF29	ZWS#	A16	AJ16
SDOEH	M03	V03			

4.4 Pin Description

4.4.1 Host Bus Interface

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
AE19	E19	CPURST	O	Reset CPU is an active high output to reset the CPU.
AJ20	A20	INIT	O	The Initialization output forces the CPU to begin execution in a known state. The CPU state after INIT is the same as the state after CPURST except that the internal caches, model specific registers, and floating point registers retain the values they had prior to INIT.
AG23	C23	CPUCLK	I	Host clock. Primary clock input to drive the part.
AH25	B25	ADS#	I	Address Status is driven by the CPU to indicate the start of a CPU bus cycle.
AG25	C25	M/IO#	I	Memory I/O definition is an input to indicate an I/O cycle when low, or a memory cycle when high.
AE22	E22	D/C#	I	Data/Code is used to indicate whether the current cycle is a data or code access.
AF25	D25	W/R#	I	Write/Read from the CPU indicates whether the current cycle is a write or read access.
AE25	E25	BRDY#	O	Burst Ready indicates that data presented are valid during a burst cycle.
AF21	D21	CACHE#	I	The Cache pin indicates an L1 internally cacheable read cycle or a burst write-back cycle. If this pin is driven inactive during a read cycle, the CPU will not cache the returned data, regardless of the state of the KEN# pin.
AJ24	A24	KEN#/ INV	O	This function as both the KEN# signal during CPU read cycles, and the INV signal during L1 snoop cycles. During CPU cycles, KEN/INV is normally low. KEN#/INV will be driven high during the 1st BRDY# or NA# assertion of a non-L1-cacheable CPU read. KEN#/INV is driven high(low) during the EADS# assertion of a PCI master DRAM write(read) snoop cycle.

AH24	B24	NA#	O	The SiS Chip always asserts NA# no matter the burst, or pipelined burst SRAMs are used. This signal is connected to CPU and indicate to CPU that it is ready to process a second cycle.
AG24	C24	BOFF#	O	The SiS Chip asserts BOFF# to stop the current CPU cycle.
AE21	E21	AHOLD	O	The SiS Chip asserts AHOLD when a PCI master is performing a cycle to DRAM. AHOLD is held for the duration of PCI burst transfer. The SiS Chip negates AHOLD when the completion of PCI to DRAM read or write cycles complete and during PCI peer transfers.
AF24	D24	HLOCK#	I	When CPU asserts HLOCK# to indicate the current bus cycle is locked.
AE24	E24	FLUSH#	O	It is used to slow down the system in deturbo mode.
AJ23	A23	EADS#	O	The EADS# is driven to indicate that a valid external address has been driven to the CPU address pins to be used for an inquire cycle.
AH23	B23	HITM#	I	Hit Modified indicates the snoop cycle hits a modified line in the L1 cache of the CPU.
AH18	B18	FERR#	I	Floating point error from the CPU. It is driven active when a floating point error occurs.
AJ22	A22	IGNNE#	O	IGNNE# is normally in high impedance state, and is asserted to inform CPU to ignore a numeric error. A resistor connected to 3.3V is required to maintain a correct voltage level to CPU.
AH22	B22	SMI#	O	System Management Interrupt is used to indicate the occurrence of system management events. It is connected directly to the CPU SMI# input.
AG22	C22	SMIACT#	I	The SMIACT# pin is used as the SMI acknowledgment input from the CPU to indicate that the SMI# is being acknowledged and the processor is operating in System Management Mode(SMM).
AJ21	A21	A20M#	O	A20 Mask is the fast A20GATE output to the CPU. It remains high during power up and CPU reset period. It forces A20 to go low when active.

AF19	D19	STPCLK#	O	Stop Clock indicates a stop clock request to the CPU. When the CPU samples STPCLK# signal asserted it response by stopping its internal clock to get into the power saving state.
AH21	B21	INTR	O	Interrupt goes high whenever a valid interrupt request is asserted.
AG21	C21	NMI	O	Non-maskable interrupt is rising edge trigger signal to the CPU and is generated to invoke a non-maskable interrupt. Normally, this signal is low. It goes high state when a non-maskable interrupt source comes up.
AH20, AG20, AJ19, AF18, AH19, AG19, AJ18, AE18	B20, C20, A19, D18, B19, C19, A18, E18	HBE[7:0]#	I	CPU Byte Enables indicate which byte lanes on the CPU data bus carry valid data during the current bus cycle. HBE7# indicates that the most significant byte of the data bus is valid while HBE0# indicates that the least significant byte of the data bus is valid.
AD5, AD4, AA4, AD3, AD2, AB5, AD1, AE4, AE3, AB4, AE2, AE1, AF4, AC5, AF3, AF2, AF1, AC4, AG2, AH2, AE7, AG3, AF7, AH3, AE8, AG4, AH4, AJ4, AF8	F5, F4, J4, F3, F2, H5, F1, E4, E3, H4, E2, E1, D4, G5, D3, D2, D1. G4, C2, B2, E7, C3, D7, B3, E8, C4, B4, A4, D8	HA[31:3]	I/O	The CPU Address is driven by the CPU during CPU bus cycles. The SiS Chip forwards it to either the DRAM or the PCI bus depending on the address range. The address bus is driven by the SiS Chip during bus master cycles.

AG18, AJ17, AH17, AE17, AG17, AF17, AJ16, AD17, AH16,AG16, AF16, AE16, AJ15, AH15, AG15, AF15, AD16, AJ14, AH14,AG14, AF14, AE14, AJ13, AH13, AD14,AG13, AF13,AJ12, AE13, AH12, AG12, AJ11, AD13,AH11, AG11, AF12, AJ10, AH10, AG10, AE12, AJ9, AH9, AF11, AG9, AJ8, AH8, AE11, AG8, AJ7, AH7, AG7, AJ6, AE10, AH6, AG6, AF6, AF9, AE6, AJ5, AH5, AE9, AG5, AF5, AE5	C18, A17, B17, E17, C17, D17, A16, F17, B16, C16, D16, E16, A15, B15, C15, D15, F16, A14, B14, C14, D14, E14, A13, B13, F14, C13, D13, A12, E13, B12, C12, A11, F13, B11, C11, D12, A10, B10, C10, E12, A9, B9, D11, C9, A8, B8, E11, C8, A7, B7, C7, A6, E10, B6, C6, D6, D9, E6, A5, B5, E9, C5, D5, E5	HD[63:0]	I/O	CPU data bus.

4.4.2 L2 Cache Controller

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
AE23	E23	KOE#	O	Cache Output Enable for pipelined burst SRAM to enable data read.
AJ26	A26	CCS1#	O	A L2 cache consisting of burst SRAMs will power up, if necessary, and perform an access if this signal is asserted when ADSC# is asserted. A L2 cache consisting of burst SRAMs will power down if this signal is negated when ADSC# is asserted. When CCS1# is negated a L2 cache consisting of burst SRAMs ignores ADS#. If CCS1# is asserts when ADS# is asserted a L2 cache consisting burst SRAMs will power up, if necessary, and perform an access.
AH26	B26	GWE#	O	Global-write Enable. GWE# asserted causes a QWORD to be written into the L2 cache. It is used for L2 cache line fills.
AG26	C26	BWE#	O	Byte-write Enable. When GWE#=1, the assertion of BWE# causes the byte lanes that are enabled via the CPU‘s HBE[7:0]# signals to be written into the L2 cache, if they are powered up.
AF22	D22	ADSC#	O	Cache address strobe is for pipelined burst SRAM to load L2 cache address register from the SRAM address pins..
AJ25	A25	ADSV#	O	Cache address advance is for pipelined burst DRAM to advance to the next data into the cache line.
AF27	D27	TAGWE#	O	TAG RAM write enable output.
AC26, AF26, AG28,AH28, AC25,AH27, AF23, AG27	G26, D26, C28, B28, G25, B27, D23, C27	TA[7:0]	I/O	TAG RAM data bus lines. The voltage level must be the same as DRAM voltage level.

4.4.3 DRAM Controller

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
AF28, AB25, AF29, AE26, AE27,AB26, AE28,AE29, AD25,AA25, AD26,AD27, AD28,AA26, AD29,AC27, Y25, AC28, AC29,AB27, AB28,Y26, AB29,AA27, AA28, W25, AA29, Y27, Y28, W26, Y29, W27, V25, W28, W29, V27, V26, V28, V29, U26, U27, U28, U29, T26, U25, T27, T28, T24, T29, R26, T25, R27, R28, R29, P24, P26, P27, P25, P28, P29, N26, N24, N27, N28	D28, H25, D29, E26, E27, H26, E28, E29, F25, J25, F26, F27, F28, J26, F29, G27, K25, G28, G29, H27, H28, K26, H29, J27, J28, L25, J29, K27, K28, L26, K29, L27, M25, L28, L29, M27, M26, M28, M29, N26, N27, N28, N29, P26, N25, P27, P28, P24, P29, R26, P25, R27, R28, R29, T24, T26, T27, T25, T28, T29, U26, U24, U27, U28	MD[63:0]	I/O	Memory data bus.
M27	V27	MA0A	O	Memory address 0. Two copies are provided for loading purposes
N29	U29	MA0B/ SRAS1#	O	Memory address 0. Two copies are provided for loading purposes. If this function is not needed, then this signal can be used as SDRAM Row address strobe. SDRAM Row address strobe. It latch row address on the positive edge of the clock with SRAS[0:1]# low. These signals enable row access and precharge.

M28	V28	MA1A	O	Memory address 1. Two copies are provided for loading purposes
N25	U25	MA1B/ SCAS1#	O	Memory address 1. Two copies are provided for loading purposes. If this function is not needed, then this signal can be used as SDRAM Column address strobe. SDRAM Column address strobe. It latch column address on the positive edge of the clock with SCAS[0:1]# low. These signals enable column access.
L25, K29, K28, K27, M26, L29, L28, L27, M25, M29	W25, Y29, Y28, Y27, V26, W29, W28, W27, V25, V29	MA[11:2]	O	Memory address 11-2 are the row and column addresses for DRAM.
J27	AA27	MA12/ GPO3	O	Memory address 12 is the row and column addresses for DRAM. If this function is not needed, then this signal can be used as General Purpose Output. General Purpose Outputs can be used to control the external device and can be controlled via configuration register.
J28	AA28	MA13/ GPO4	O	Memory address 13 is the row and column addresses for DRAM. If this function is not needed, then this signal can be used as General Purpose Output. General Purpose Outputs can be used to control the external device and can be controlled via configuration register.
J29	AA29	MA14/ GPO6	O	Memory address 14 is the row and column addresses for DRAM. If this function is not needed, then this signal can be used as General Purpose Output. General Purpose Outputs can be used to control the external device and can be controlled via configuration register.
H27, L26	AB27, W26	RAMWA# RAMWB#	O	RAM Write is an active low output signal to enable local DRAM writes. Two copies are provided for loading purposes.

H25, E27, E26, J26, F29, F28, J25, F27	AB25, AE27, AE26, AA26, AD29, AD28, AA25, AD27	CAS[7:0]#/ DQM[7:0]	O	(FPM/EDO)DRAM Column address strobe 7-0 for byte 7-0. SDRAM output enables during a read cycle and a byte mask during a write cycle.
K25, G26, G29, G28, K26, G27	Y25, AC26, AC29, AC28, Y26, AC27	RAS[5:0]#/ CS[5:0]#	O	(FPM/EDO)DRAM Row address strobe 5-0 for DRAM banks 2-0. SDRAM chip select. These pins activate the SDRAM and accept any command when it is low.
H29 D28	AB29 AF28	SRAS0# SRAS2#	O	SDRAM Row address strobe. It latch row address on the positive edge of the clock with SRAS[0:2]# low. These signals enable row access and precharge.Third copies are provided for loading purposes.
H28 D29	AB28 AF29	SCAS0# SCAS2#	O	SDRAM Column address strobe. Third copies are provided for loading purposes.
E29	AE29	DDCCLK1	I/O	I^2C Bus Clock
E28	AE28	DDCDAT1	I/O	I^2C Bus Data

4.4.4 PCI Interface

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
B2	AH2	PCICLK	I	The PCICLK input provides the fundamental timing and the internal operating frequency for the SiS Chip. It runs at the same frequency and skew of the PCI local bus.
B4, E1, G2, L4	AH4, AE1, AC2, W4	C/BE[3:0]#	I/O	PCI Bus Command and Byte Enables define the PCI command during the address phase of a PCI cycle, and the PCI byte enables during the data phases. C/BE[3:0]# are outputs when the SiS Chip is a PCI bus master and inputs when it is a PCI slave.
D8, A4, C3, B3, C2, G5, D4, D3, D2, G4, D1, E4, H5, E3, E2, H4, G1, H3, K4, H1, J3, L5, J2, J1, K3, K2, K1, M5, L3, L2, M4, L1	AF8, AJ4, AG3, AH3, AG2, AC5, AF4, AF3, AF2, AC4, AF1, AE4, AB5, AE3, AE2, AB4, AC1, AB3, Y4, AB1, AA3, W5, AA2, AA1, Y3, Y2, Y1, V5, W3, W2, V4, L1	AD[31:0]	I/O	PCI Address /Data Bus *In address phase:* 1. When the SiS Chip is a PCI bus master, AD[31:0] are output signals. 2. When the SiS Chip is a PCI target, AD[31:0] are input signals. *In data phase:* 1. When the SiS Chip is a target of a memory read/write cycle, AD[31:0] are floating. 2. When the SiS Chip is a target of a configuration or an I/O cycle, AD[31:0] are output signals in a read cycle, and input signals in a write cycle.
G3	AC3	PAR	I/O	Parity is an even parity generated across AD[31:0] and C/BE[3:0]#.
F5	AD5	FRAME#	I/O	FRAME# is an output when the SiS Chip is a PCI bus master. The SiS Chip drives FRAME# to indicate the beginning and duration of an access. When the SiS Chip is a PCI slave, FRAME# is an input signal.

F4	AD4	IRDY#	I/O	IRDY# is an output when the SiS Chip is a PCI bus master. The assertion of IRDY# indicates the current PCI bus master's ability to complete the current data phase of the transaction. For a read cycle, IRDY# indicates that the PCI bus master is prepared to accept the read data on the following rising edge of the PCI clock. For a write cycle, IRDY# indicates that the bus master has driven valid data on the PCI bus. When the SiS Chip is a PCI slave, IRDY# is an input.
J5	AA5	TRDY#	I/O	TRDY# is an output when the SiS Chip is a PCI slave. The assertion of TRDY# indicates the target agent's ability to complete the current data phase of the transaction. For a read cycle, TRDY# indicates that the target has driven valid data onto the PCI bus. For a write cycle, TRDY# indicates that the target is prepared to accept data from the PCI bus. When the SiS Chip is a PCI master, it is an input.
F2	AD2	STOP#	I/O	STOP# indicates that the bus master must start terminating its current PCI bus cycle at the next clock edge and release control of the PCI bus. STOP# is used for disconnect, retry, and target-abort sequences on the PCI bus.
F3	AD3	DEVSEL#	I/O	As a PCI target, SiS Chip asserts DEVSEL# by doing positive or subtractive decoding. SiS Chip positively asserts DEVSEL# when the DRAM address is being access by a PCI master, PCI configuration registers or embedded controllers' registers are being addressed, or the BIOS memory space is being accessed. The low 16K IO space and low 16M memory space are subtractively responded. The DEVESEL# is an input when SiS Chip is acting as a PCI master. It is asserted by the addressed agent to claim the current transaction.

F1	AD1	PLOCK#	I	PCI Lock indicates an exclusive bus operation that may require multiple transactions to complete. When PLOCK# is sampled asserted at the beginning of a PCI cycle, the SiS Chip considers itself a locked resource and remains in the locked state until PLOCK# is sampled negated on a new PCI cycle.
E8, A5, D9, D5	AE8, AJ5, AF9, AF5	PREQ[4:0]#	I	PCI Bus Request is used to indicate to the PCI bus arbiter that an agent requires use of the PCI bus.
C4, B5, C5, E5	AG4, AH5, AG5, AE5	PGNT[4:0]#	O	PCI Bus Grant indicates to an agent that access to the PCI bus has been granted.
C6, B6, A6, E9	AG6, AH6, AJ6, AE9	INT[A:D]#	I	PCI Interrupt A to Interrupt D
J4	AA4	SERR#	I	SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the SiS Chip generates a non-maskable interrupt to the CPU.

4.4.5 PCI IDE/ISA Bus Interface

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
U5, U3, U4, T1, U6, T2, T3, T4, T5, R1, R2, R3, R4, T6, P1, P2	N5, N3, N4, P1, N6, P2, P3, P4, P5, R1, R2, R3, R4, P6, T1, T2	IDA[15:0]/ SD[15:0]	I/O	When IDE channel_0 is operating, these are IDE channel_0 data bus. Otherwise, these signals are ISA data bus and connect to ISA slots via two 74LS245s.
Y3, Y2, W4, Y1, AA3, AA2, Y5	K3, K2, L4, K1, J3, J2, K5	IDB[15:9]/ LA[23:17]	I/O	When IDE channel_1 is operating, these are IDE channel_1 data bus. Otherwise, these signals are ISA address bus and connect to ISA slots via 74LS245.
AC1, AA5, AC2, AC3, AB1, Y4, AB2, AB3, AA1	G1, J5, G2, G3, H1, K4, H2, H3, J1	IDB[8:0]/ SA[16:8]	I/O	When IDE channel_1 is operating, these are IDE channel_1 data bus. Otherwise, these signals are ISA address bus and connect to ISA slots via 74LS245.
P5, P3	T5, T3	ICSA[1:0]#	O	IDE channel 0 chip select signals.
W5, W1	L5, L1	ICSB[1:0]#	O	IDE channel 1 chip select signals.
P4, W2	T4, L2	IIOR[A:B]#	O	IDE channel 0/1 I/O read cycle command.

P6, V4	T6, M4	IIOW[A:B]#	O	IDE channel 0/1 I/O write cycle command
N1	U1	ICHRDYA	I	IDE channel 0 I/O channel ready signal.
W3	L3	ICHRDYB	I	IDE channel 1 I/O channel ready signal.
N2, V1	U2, M1	IDREQA IDREQB	I	IDE channel 0/1 DMA request signals.
N3, V2	U3, M2	IDACKA# IDACKB#	O	IDE channel 0/1 DMA acknowledge signals.
N5, V5	U5, M5	IIRQ[A:B]/ IRQ[14:15]	I	IDE channel 0/1 interrupt request signals. These are the synchronous interrupt request inputs to the 8259 controller.
N4, M1, M2	U4, V1, V2	IDSAA[2:0]	O	IDE channel 0 address [2:0].
V3, U1, U2	M3, N1, N2	IDSBA[2:0]	O	IDE channel 1 address [2:0].
M3	V3	SDOEH	O	This signal connects to the DIR pin of 74LS245 and is used to control the data flow of ISA highbyte data bus. When a “0” is output, the direction of data is going into SiS Chip, when a “1” is output, the direction of data is going out of SiS Chip. When the IDE/ISA bus is used by IDE, this signal is “1”.
N6	U6	SDOEL	O	This signal connects to the DIR pin of 74LS245 and is used to control the data flow of ISA lowbyte data bus. When a “0” is output, the direction of data is going into SiS Chip, when a “1” is output, the direction of data is going out of SiS Chip. When the IDE/ISA bus is used by IDE, this signal is “1”.
A13	AJ13	OSC	I	It is the buffered input of the external 14.318MHz oscillator.
B9	AH9	BCLK	O	ISA bus clock, for ISA bus controller, ISA bus interfaces and the DMA controller. It can be programmed to derive from the PCICLK or 7.159MHz from the 14MHz clock.
B12	AH12	IO16#	I	16-bit I/O chip select indicates that the AT bus cycle is a 16-bit I/O transfer when asserted or an 8-bit I/O transfer when it is negated.
C12	AG12	M16#	I	16-bit memory chip select indicates a 16-bit memory transfer when asserted or an 8-bit memory transfer when it is negated.

C15	AG15	AEN	O	Address Enable is driven high on the ISA bus to indicate the address lines are valid in DMA or ISA master cycles. It is low otherwise.
E16	AE16	IORDY	I/O	I/O channel ready is normally high. It can be pulled low by the slow devices on the AT bus to add wait states for the ISA memory or I/O cycles. When a DMA or an ISA master accesses a target, IORDY is an output to control the wait states.
D27	AF27	IOCHK#/ GPIO9/ THRM#	I/O	I/O Channel Check, or General Purpose Input/Output, or Thermal Detect. Please refer to "Multiplexed pins" section to define the pin function.
B13	AH13	BALE	O	Bus address latch enable is used on the ISA bus to latch valid address from the CPU. Its falling edge starts the ISA command cycles.
C9	AG9	MR16#	I	Master is an active low signal from AT bus. When active, it indicates that the ISA bus master has the control of the system. The address and control signals are all driven by the ISA bus master.
C11	AG11	MEMR#	I/O	AT bus memory read command signal is an output pin during AT/DMA/refresh cycles and is an input pin in ISA master cycles.
B11	AH11	MEMW#	I/O	AT bus memory write command signal is an output pin during AT/DMA cycles and is an input pin in ISA master cycles.
D14	AF14	IOR#	I/O	AT bus I/O read command signal is an output pin during AT or DMA cycles and is an input pin in ISA master cycles. When low, it strobes an I/O device to place data on the data bus.
A15	AJ15	IOW#	I/O	AT bus I/O write command signal is an output pin during AT or DMA cycles and is an input pin in ISA master cycles. When low, it strobes data on the data bus into a selected I/O device.
B15	AH15	SMEMR#	O	AT bus memory read. It instructs the memory devices to drive data onto the data bus. It is active only when the memory being accessed is within the lowest 1MB.

F14	AD14	SMEMW#	O	AT bus memory write. It instructs the memory devices to store the data presented on the data bus. It is active only when the memory being accessed is within the lowest 1MB.
A16	AJ16	ZWS#	I	Zero wait state is an active low signal. The system ignores the IORDY signal and terminates the AT bus cycle without additional wait state when it is asserted.
E17	AE17	RFH#	I/O	Refresh signal is used to initiate a refresh cycle. This signal is an input in ISA bus master cycles and is an output in other cycles.
E13	AE13	SBHE#	I/O	Byte high enable signal indicates that the high byte has valid data on the ISA 16-bit data bus. This signal is an output except during ISA master cycles.
C13	AG13	TC	O	Terminal Count of DMA. A pulse is generated by the DMA controller when the terminal count (TC) of any channel reaches 1.When a TC pulse occurs, the DMA controller will terminate the service, and if auto-initialize is enabled, the base registers will be written to the current registers of that channel. The mask bit and the TC bit in the Status word will be set for the currently active channel unless the channel is programmed for auto-initialize. In that case, the mask bit remains clear.
A17, B17, C17, F17, D17, A18, B18, C18	AJ17, AH17, AG17, AD17, AF17, AJ18, AH18, AG18	SA[7:0]	I/O	System address 7~0. They are inputs when an external bus master is in control and are outputs at all other times.
F13, D13, A14, B14, E14, C14, A12, E12	AD13, AF13, AJ14, AH14, AE14, AG14, AJ12, AE12	IRQ[3:7], IRQ[9:11]	I	These are the synchronous interrupt request inputs to the SiS Chip internal 8259 controller.

D16, C16, F16, B16, A11, D12, E11	AF16, AG16, AD16, AH16, AJ11, AF12, AE11	DREQ[0:3], DREQ[5:7]	I	DMA Request inputs are used by external devices to indicate when they need service from the internal DMA controllers.
C10, B10, A10	AG10. AH10, AJ10	DACK[0:2] #	O	DMA acknowledge output are used by external devices to indicate when they need service from the internal DMA controllers.
C19	AG19	ROMKBCS #	O	Keyboard or System ROM Chip Select. When asserted, it means the keyboard or ROM is to be accessed.
A20	AJ20	SPK	O	Speaker is the output for the speaker.

4.4.6 RTC/KBC

SiS5582 BALL No.	**SiS5581 BALL No.**	**NAME**	**TYPE ATTR**	**DESCRIPTION**
C20	AG20	KLOCK#/ GPIO0/ RAMWC#	I/O	This pin can used as the keyboard lock signal if internal KBC is enabled and Reg 70h bit 4 is set to “1” in PCI to ISA bridge configuration space. If this function is not needed, then it can be used as General Purpose Input/Output signal and can be control via configuration register.This pin can also use as the RAMWC# for the third DIMM RAM write enable.
B19	AH19	KBCLK/ GPIO2	I/O	When the internal KBC is enabled, this pin is used as the keyboard clock. If this function is not needed, then it can be used as General Purpose Input/Output signal and can be control via configuration register.
A19	AJ19	KBDAT/ IRQ1	I/O	When the internal KBC is enabled, this pin is used as the keyboard data. Otherwise, it is the IRQ1 signal use for external KBC.
D19	AF19	PMCLK/ GPIO1	I/O	When internal KBC is enabled, it can be served as PS2 mouse clock.
B20	AH20	PMDAT/ IRQ12	I/O	When in input mode, it functions as PMDAT if PS/2 mouse is enabled. If this function is not needed, then it can be used as ISA interrupt request 12.

C8	AG8	ONCTL#/ RTCALE	O	Power ON/OFF control. This open-drain output, powered by the RTCVDD, signals the main power supply that power should be turned on/off. When using external RTC: The signal is used to latch the address from the SD bus when CPU accesses RTC.
A8	AJ8	PSRSTB#	I	When using internal RTC: This signal is used as PSRSTB# (power strobe). PSRSTB# establishes the condition of the control register in RTC when power is first applied to the device.
D10	AF10	RTCVSS	PWR	RTC Ground.
A7	AJ7	RTCVDD	I	Power for internal RTC and APC.
E6	AE6	OSCI/ IRQ8#	I	When using internal RTC: This pin is used as the time base of the integrated RTC. This signal should be connected to 32.768 KHz crystal or oscillator input. When using external RTC: This pin is used as IRQ8#, which is the asynchronous interrupt request input to SiS Chip internal 8259 controller.
D6	AF6	OSCO/ RTCCS#	O	When using internal RTC: this pin should be connected the other end of the 32.768 KHz crystal or left unconnected if an oscillator is used. When using external RTC: This pin is used as chip select of RTC. It combine with IOR# and IOW# to generate RTCRD# and RTCWR#, that are used to store the data presented on the SD bus when CPU accesses the RTC.
C7	AG7	PWRGD	I	Power Good signal.

4.4.7 PMU/ACPI Controller

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
E18	AE18	GPCS0	I/O	General Programmable Chip Select 0 can be controlled via registers. This signal also can be programmed as GPO Write Enable 0 to latch enable signal to an external 74F374 for general purpose outputs from SD[7:0] bus.
E10	AE10	STARTREQ#/ GPIO5	I/O	General Purpose Input/Output. If used as STARTREQ#, a high to low transition indicates a power up event which activates ONCTL#.
D26	AF26	GPIO7/ OCO#/ OCI2#	I/O	General Purpose Input/Output. If this function is not needed, then it can be used as Global Power Enable Switch of USB port or Over Current Detect of USB port. Global Power Enable Switch is used to control the external Power-Distribution Switchs logic to power off the USB power supply lines.
H26	AB26	GPIO8/ OCI1#	I/O	General Purpose Input/Output. If this function is not needed, then it can be used as Over Current Detect of USB port and it must be programmed as input mode. Over Current Detect is used to detect the status of the USB power supply lines.
B7	AH7	GPIO10/ ACPILED	I/O	General Purpose Input/Output. If this function is not needed, then it can be used as ACPILED signal to control LED on/off in ACPI power saving state.
B22	AH22	TURBO/ EXTSMI#	I	When this signal be programmed as TURBO. This pin is used to slow down the system by connecting it to ground. When this signal be programmed as EXTSMI#. A signal from the break switch will cause the system enters the standby state. The pulse width of the EXTSMI# must greater than 4 CPUCLK.

D20	AF20	GPCS1/ SIRQ	I/O	General Programmable Chip Select 1 can be controlled via registers. This signal also can be programmed as GPO Write Enable 1 to latch enable signal to an external 74F374 for general purpose outputs from SD[7:0] bus. It is available as Serial Interrupt ReQuest function, it is a wired-OR signal and support ISA standard IRQs within PCI-based system.
B8	AH8	RING	I	When enable, detection of RING pulse or pulse train activates the ONCTL# pin. The pulse must be 4ms at least, and only one pulse in a sec. Engineering note: Input is blocked to reduce leakage current when either the APC‘s ring event is disabled or the APC is powered by RTCVDD.
A9	AJ9	PWRBT#	I	Power Button. This function provide the user interface control used to cycle the system between the sleeping and working states through Power Button switch. It also support the power-button-over function for system power off if it is pressed over 4 sec.
D11	AF11	SWITCH	I	Power On/Off switch. Indicated a Switch On/Off request. When PWRGD is low, an active low on the SWITCH indicates a SWITCH-on request event. When PWRGD is high, the logic indicates a SWITCH-off request event. The pin has a schmitt-trigger input buffer and a debounce protection of at least 30ms.

4.4.8 USB Controller

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
B21, E19	AH21, AE19	UV0+, UV0-	I/O	When USB function port 0. These two pins are used as the differential USB data bus pair of port 0.
C21, A22	AG21, AJ22	UV1+, UV1-	I/O	When USB function port 1. These two pins are used as the differential USB data bus pair of port 1.
A21	AJ21	UCLK48	I	48 Mhz USB clock .

4.4.9 Power Pins

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
AF20, D7	D20, AF7	DLLVDD0, DLLVDD1	PWR	+3.3V DC power for DLL circuit.
AE20, E7	E20, AE7	DLLVSS0, DLLVSS1	PWR	Ground for DLL circuit.
D15, H2	AF15, AB2	VCC5	PWR	+5V DC power for 5V safe input buffers in a system requiring 5V I/O tolerance.
E15, F15, R5, R6, R24, R25, AD15,AE15	E15, F15, R5,R6, R24, R25, AD15,AE15	DVDD	PWR	+3.3V I/O PAD DC power.
K12, K13, K17, K18, M10, M20, N10, N20, U10, U20, V10, V20, Y12, Y13, Y17, Y18,	K12, K13, K17, K18, M10, M20, N10, N20, U10, U20, V10, V20, Y12, Y13, Y17, Y18,	OVDD	PWR	+3.3V DC power for main voltage supply of SiS Chip.
D18, K5, U24, AF10	AF18, Y5, N24, D10	PVDD	PWR	+3.3V DC power
B25, B26, B27, D25, E22, E24, E25, E21, C25, C26, C27	AE22, AE24, AH25, AH26, AH27, AE25, AE21, AF25, AG25, AG26, AG27	DVSS	PWR	Ground.
M[12:18], N[12:18], P[12:18], R[12:18], T[12:18], U[12:18], V[12:18]	M[12:18], N[12:18], P[12:18], R[12:18], T[12:18], U[12:18], V[12:18]	GND	PWR	Ground

4.4.10 Misc. Pins

SiS5582 BALL No.	SiS5581 BALL No.	NAME	TYPE ATTR	DESCRIPTION
B28, F25, A23, B23, E20, C23, D23, E23, D21, B24, C24, D24, F26, G25, C28, A26, A25, A24, D22, A3, A27, C1, C29, AG1, AG29, AJ3, AJ27	AH28, AD25, AJ23, AH23, AE20, AG23, AF23, AE23, AF21, AH24, AG24, AF24, AD26, AC25, AG28, AJ26, AJ25, AJ24, AF22, AJ3, AJ27, AG1, AG29, C1, C29, A3, A27	NC	NC	Not Connect
C22	AG22	TEST#	I	Test mode Select for NAND Tree chain.

5. Hardware Trap

Several pins in the SiS Chip are used for trapping purpose to identify the hardware configurations at the power-up stage. These pins should be defined as “1” if pull-up resistors are used; and these pins should be “0” if pull-down resistors are used. The following table is a summary of all the Hardware Trap pins in SiS Chip.

SiS5582 Ball No	SiS5581 Ball No	Symbol	Description
AD26	F26	MD53	**Internal DLL circuit for CPU clock to optimize timing Control** Pull-up : Disable Pull-down : Enable
AA25	J25	MD54	**Internal DLL circuit for PCI clock to optimize timing Control** Pull-up : Disable Pull-down : Enable (Internal Test Mode)
Y29	K29	MD33	**Internal Test Mode.** This pin **MUST** be pull-up for SiS Chip normal operation. It is also defined the pins of PREQ4#/PGNT4# for 5th PCI Master device.
C22	AG22	TEST#	**Ball connectivity test mode** Pull-up : Disable Pull-down : Enable
A08	AJ08	PSRSTB#	**Connect to battery‘s power strobe: Select internal RTC**. Pull-down: Select external RTC.

6. Register Description

6.1 Host to PCI bridge configuration space

Device	IDSEL	Function Number
Host to PCI bridge	AD11	0000b

Register 00h~01h Vendor ID

Bits 15:0 1039h

Register 02h~03h Device ID

Bits 15:0 5597h

Register 04h~05h Command

Bits 15:10 Reserved

Bit 9 Fast Back-to-Back Enable

Bit 8 Reserved

Bits 7:2 Reserved and read as 01h.

Bit 1 Control a device's response to memory space accesses

0: Disable the device response

1: Allow the device response, state after PCIRST# (via CPURST) is 0

Bit 0 Reserved and should be set to 1

Register 06h~07h Status

Bit 15 Detected Parity Error

This bit is always 0, SiS Chip does not support parity checking on the PCI bus.

Bit 14 Reserved

Bit 13 Received Master Abort.

This bit is set by SiS Chip whenever it terminates a transaction with master abort. This bit is cleared by writing a 1 to it.

Bit 12 Received Target Abort.

This bit is set by SiS Chip whenever it terminates a transaction with target abort. This bit is cleared by writing a 1 to it.

Bit 11 Signaled Target Abort.

This bit is always 0 since SiS Chip will not terminate a transaction with target abort.

Bits 10:9 DEVSEL# Timing DEVT.

The two bits define the timing to assert DEVSEL#. The default value is DEVT=01. In fact, the SiS Chip always asserts DEVSEL# in medium timing.

Bit 8 Reserved

Bits 7:0 Reserved and read as "00h"

Register 08h Revision Identification.

Bits 7:0 02h

Register 09h Class ID

Bits 7:0 00h

Register 0Ah Sub-Class Code

Bits 7:0 00h

Register 0Bh Base Class Code

Bits 7:0 06h

Register 0Ch Cache Line Size

Bits 7:0 00h

Register 0Dh Master latency timer

Bits 7:0 Master latency timer
The default value is FFh, it means 255 PCI clocks.
Unit: PCI Clocks

Register 0Eh Header Type

Bits 7:0 00h (read only)

Register 0Fh BIST

Bits 7:0 00h (read only)

Register 50h Host Interface and DRAM arbiter (default = 00h)

Bit 7 NA# assert Control
0: Disable
1: Enable

Bit 6 NA# asserted on All Single Write Cycle (for internal dirty SRAM)
0: Enable (when using internal dirty SRAM)
1: Disable (When using AMD K6 write allocation function)

Bit 5 NA# Delay 1T on Burst read Hit L2 Cache Cycle

0: Disable

1: Enable (when using two banks P.B.SRAM)

Bit 4 **NA# Assert before the 1st BRDY# on Burst Read Miss Cycle (only available in cacheless system to improve the performance)**

0: Disable

1: Enable

Bit 3 **Write Merge Control**

0: Disable

1: Enable

Bit 2 **Read Write Reorder Control**

0: Disable

1: Enable

Bit 1 **Reserved, must be 0.**

Bit 0 **Reserved, must be 0.**

Register 51h L2 Cache Controller (default = 00h)

Bit 7 **L2 Cache exists Selection**

0: No L2 Cache

1: L2 Cache Exists

When no L2 exists, this bit should be programmed to 0.

Bit 6 **L2 Cache Enable**

0: Disable

1: Enable

Bits 5:4 **L2 Cache Size**

00: Reserved

01: 256K

10: 512K

11: Reserved

Bit 3 **L2 Cache WT/WB Policy**

0: Write Through Mode (only for cache sizing)

1: Write Back Mode

Bit 2 **L2 Cache Burst Addressing mode**

0: Toggle Mode

1: Linear Mode (For Cyrix CPU)

Bit 1 **L2 Cache Tag Size Selection**

0: 7 bits : TA7 is dirty bit and disable internal dirty SRAM

1: 8 bits : using internal dirty SRAM

Bit 0 **L2 Cache Sizing Enable**

0: Normal Operation

1: Always Cache Hit

Register 52h Control Register (default = 00h)

Bit 7 **CPU L1 Cache Write Back Mode Enable**
0: Disable
1: Enable

Bit 6 **Single read Allocation (L2 update) Control**
0: Disable
1: Enable

Bit 5 **Read FIFO Control**
0: Disable
1: Enable

Bit 4 **Reserved**

Bit 3 **Reserved**
This bit should be programmed to 0.

Bit 2 **Reserved**
This bit is programmed to 0.

Bit 1 **DRAM Refresh Mode (internal use only)**
0: Normal Mode
1: Test Mode

Bit 0 **Internal SRAM test mode (internal use only)**
0: Normal Mode
1: Test Mode

Register 53h DRAM Control Register (default = 38h)

Bits 7:6 **Starting point of Paging function select**
00: 1T (Recommended)
01: 2T
10: 4T
11: 8T
Note: This setting is used to control the start timing of the page operations which defined in bit[5:3].

Bit 5 **Always Page Miss After Write DRAM Cycles**
0: Disable
1: Enable

Bit 4 **Always Page Miss After Data Read DRAM Cycles**
0: Disable
1: Enable

Bit 3 **Always Page Miss After Code Read DRAM Cycles**
0: Disable
1: Enable

Bits 2:1 **Refresh cycle time**
00: 15.6 us

01: 62.4 us
10: 124.8 us
11: 187.2 us

Bit 0 **Reserved**

Register 54h DRAM Control Register 0(default = 54h)

Bits 7:6 RAS pulse width when refresh Cycle
00: 5T for EDO/FP DRAM and 4T for SDRAM
01: 6T for EDO/FP DRAM and 5T for SDRAM
10: 7T for EDO/FP DRAM and 6T for SDRAM
11: 8T for EDO/FP DRAM and 7T for SDRAM

Bits 5:4 RAS precharge time
00: 2T
01: 3T
10: 4T
11: 5T

Bits 3:2 RAS to CAS delay
00: 2T
01: 3T
10: 4T
11: 5T

Bit 1 CAS pulse width only for FPM DRAM
0: 2T
1: 1T

Bit 0 CAS pulse width only for EDO DRAM
0: 2T
1: 1T

Register 55h FPM/EDO DRAM Control Register 1(default = 00h)

Bit 7 RAMWA/B/C# assertion timing when read cycle followed by write cycle
0: 3T
1: 2T

Bit 6 EDO test mode
0: Normal mode
1: Test mode (for DRAM sizing)

Bit 5 SDRAM Read Cycle Delay 1T when Pipelined after Write Cycle
0: Disable(Normal)
1: Enabled(Recommended in 75 MHz or higher frequency)

Bits 4:3 CAS Precharge Time for FPM DRAM
00: 1T
01: 1T during burst cycles, 2T for different cycles (1 wait state between cycles)

10: 2T
11: Reserved

Bits 2:1 **CAS Precharge Time for EDO DRAM**
00: 1T
01: 1T during burst cycles, 2T for different cycles (1 wait state between cycles)
10: 2T
11: Reserved

Bit 0 **Reserved**

Register 56h Memory Data Latch Enable (MDLE) Delay Control Register

Bit 7 **SRAS#/SCAS#/RAMW# Pre-state Enable Bit (SDRAM)**
0: Disable
1: Enable (Recommended)

Bit 6 **Delay EDO/FPM DRAM Read Lead-off Cycle 1T**
0: Disable (Lead-off time = 5T)
1: Enable (Lead-off time = 6T)

Bits 5:4 **SDRAM Read Cycle Lead-off Time**
00: Normal (Lead-off time is 6T when CAS Latency is 2T or Lead-off time is 7T when CAS Latency is 3T)
01: Faster (Lead-off time is 5T when CAS Latency is 2T or Lead-off time is 6T when CAS Latency is 3T)
10: Slower (Lead-off time is 7T when CAS Latency is 2T or Lead-off time is 8T when CAS Latency is 3T)

Bit 3 **SDRAM Sizing Enable bit Control**
0: Disable
1: Enable
This bit must set to ‘1’ before initializing the SDRAM sizing. Once the SDRAM sizing completed, this bit must set to ‘0’.

Bits 2:0 **MDLE Delay**

000 : No delay	001 : delay 1 ns
010 : delay 2 ns	011 : delay 3 ns
100 : delay 4 ns	101 : delay 5 ns
110 : delay 6 ns	111 : delay 7 ns

Register 57h SDRAM Control Register

Bit 7 **Precharge Command**
0: Disable
1: Enable
(After the cycle completes, this bit will be cleared automatically.)

Bit 6 **Mode Register Set Command**

0: Disable Command Cycle
1: Enable Command Cycle

Bit 5 **For SDRAM sizing Refresh Command**
0: Disable
1: Enable
(After the cycle completes, this bit will be cleared automatically.)

Bit 4 **CAS Latency**
0: 2T
1: 3T

Bit 3 **SDRAM write retire rate**
0: X-2-2-2
1: X-1-1-1

Bit 2 **SDRAM wait state control during Precharge Command**
0: One wait state (Recommended)
1: Zero wait state

Bit 1 **Pin Definition Select for MA0B/SRAS1#, MA1B/SCAS1#**
0: MA0B, MA1B
1: SRAS1#, SCAS1#

Bit 0 **Pin Definition Select for KLOCK#/GPIO0/RAMWC#**
0: RAMWC#
1: KBLOCK#/GPIO0

Register 58h

Bit 7 **SDRAM ROWHIT RAS# Precharge Time Control**
0: Always ROWHIT(Recommended)
1: ROWHIT from Address Comparing Circuit

Bits 6:5 **When for DLL to Lock the Reference Clock Source**
00: 4T(T is the reference clock source for DLL)
01: 8T
10: 16T
11: 32T
Adjust the locking frequency by every x clocks.

Bit 4 **DLL Function Test Mode**
0: Normal Mode
1: Test Mode

Bit 3 **RAS#/CS# Assertion Timing When Refresh Cycles**
0: Normal
1: 1T Command Pulse
This bit must set to 1 if use SDRAM.

Bit 2 **SDRAM Back-to-Back Read Timing**

0: 5-1-1-1-2-1-1-1

1: 5-1-1-1-1-1-1-1

Only for cacheless systems, and Fast Read enabled (Host to PCI:56h[5:4]=01), and NA# 1T ahead BRDY# enabled (Host to PCI:50h[4]=1)

Bits 1:0 **Reserved**

Register 59h DRAM signals driving current Control (default = 00h)

Bit 7 **Selection of RAS[5:0]# Current Rating**

0: 4mA

1: 8mA

Bit 6 **Selection of CAS[7:0]# Current Rating**

0: 12mA

1: 16mA

Bit 5 **Selection of MA[14:2] Current Rating**

0: 6mA

1: 16mA

Bit 4 **Selection of MA[1:0]A Current Rating**

0: 6mA

1: 16mA

Bit 3 **Selection of MA[1:0]B Current Rating**

0: 6mA

1: 16mA

Bit 2 **Selection of RAMWA#/RAMWC# Current Rating**

0: 12mA

1: 16mA

Bit 1 **Selection of RAMWB# Current Rating**

0: 12mA

1: 16mA

Bit 0 **Selection of SRAS#/SCAS# Current Rating**

0: 12mA

1: 16mA

Register 5Ah PCI signals driving current Control

Bits 7:2 **Reserved**

Bit 1 **Selection of AD[31:0] Current Rating**

0: 4mA

1: 8mA

Bit 0 **Selection of FRAME#, IRDY#, TRDY#, DEVSEL#, STOP#, PAR, C/BE[3:0]# and PGNT[3:0]# Current Rating**

0: 4mA

1: 8mA

Register 5Bh~5Fh Reserved

Register 60h/61h/62h DRAM Bank 0/1/2 Register

Bits 7:6	**DRAM Mode Selection** 00: FPM DRAM 01: EDO DRAM 10: Reserved 11: SDRAM
Bit 5	**Double/Single Sided DRAM** 0: Single Sided 1: Double Sided
Bit 4	**Half/Full DRAM Populated** 0: 64 bits DRAM is Populated 1: 32 bits DRAM is Populated
Bits 3:0	**DRAM Type Selection**

For EDO/FPM DRAM :

0000 : 256K Symmetric 9x9	0001 : 1M Symmetric 10x10
0010 : 4M Symmetric11x11	0011 : 16M Symmetric12x12
0100 : 1M Asymmetric 12x8	0101 : 2M Asymmetric 12x9
0110 : 4M Asymmetric 12x10	0111 : 8M Asymmetric 12x11
1000 : 512K Asymmetric 10x9	1001 : 1M Asymmetric 11x9
1010: 2M Asymmetric 11x10	Others : Reserved

For SDRAM :

0000 : 1M 12x8 (2 banks)	0001 : 4M 14x8 (2 banks)
0010 : 4M 14x8 (4 banks)	0011 : 8M 15x8 (4 banks)
0100 : 2M 12x9 (2 banks)	0101 : 8M 14x9 (2 banks)
0110 : 8M 14x9 (4 banks)	0111 : 16M 15x9 (4 banks)
1000 : 4M 12x10 (2 banks)	1001 : 16M 14x10 (2 banks)
1010: 16M 14x10 (4 banks)	1011 : 32M 15x10 (4 banks)
1100: 2M 13x8 (4 banks)	Others: Reserved

Register 63h DRAM Status (Default=FFh)

Bits 7:3	**Reserved**
Bit 2	**DRAM Status for Bank2** 0: Absent 1: Installed
Bit 1	**DRAM Status for Bank1** 0: Absent 1: Installed

Bit 0 **DRAM Status for Bank0**
0: Absent
1: Installed

Register 64h~6Fh Reserved

Register 70h to register 76h define the attribute of the Shadow RAM from 640 KByte to 1 MByte. All of the registers 70h to 75h are defined as below, and each register defines the corresponding memory segment's attribute which are listed in the following table.

Register	Defined Range	Register	Defined Range
register 70h bits 7:5	0C0000h-0C3FFFh	register 73h bits 7:5	0D8000h-0DBFFFh
register 70h bits 3:1	0C4000h-0C7FFFh	register 73h bits 3:1	0DC000h-0DFFFFh
register 71h bits 7:5	0C8000h-0CBFFFh	register 74h bits 7:5	0E0000h-0E3FFFh
register 71h bits 3:1	0CC000h-0CFFFFh	register 74h bits 3:1	0E4000h-0E7FFFh
register 72h bits 7:5	0D0000h-0D3FFFh	register 75h bits 7:5	0E8000h-0EBFFFh
register 72h bits 3:1	0D4000h-0D7FFFh	register 75h bits 3:1	0EC000h-0EFFFFh

Register 70h shadow RAM Registers

Bit 7 **Read enable**
Bit 6 **L1/L2 cacheable**
Bit 5 **Write enable**
Bit 4 **Reserved**
Bit 3 **Read enable**
Bit 2 **L1/L2 cacheable**
Bit 1 **Write enable**
Bit 0 **Reserved**

Register 71h shadow RAM Registers

Bit 7 **Read enable**
Bit 6 **L1/L2 cacheable**
Bit 5 **Write enable**
Bit 4 **Reserved**
Bit 3 **Read enable**
Bit 2 **L1/L2 cacheable**
Bit 1 **Write enable**
Bit 0 **Reserved**

Register 72h shadow RAM Registers

Bit 7	**Read enable**
Bit 6	**L1/L2 cacheable**
Bit 5	**Write enable**
Bit 4	**Reserved**
Bit 3	**Read enable**
Bit 2	**L1/L2 cacheable**
Bit 1	**Write enable**
Bit 0	**Reserved**

Register 73h shadow RAM Registers

Bit 7	**Read enable**
Bit 6	**L1/L2 cacheable**
Bit 5	**Write enable**
Bit 4	**Reserved**
Bit 3	**Read enable**
Bit 2	**L1/L2 cacheable**
Bit 1	**Write enable**
Bit 0	**Reserved**

Register 74h/75h shadow RAM Registers

Bit 7	**Read enable**
Bit 6	**L1/L2 cacheable**
Bit 5	**Write enable**
Bit 4	**Reserved**
Bit 3	**Read enable**
Bit 2	**L1/L2 cacheable**
Bit 1	**Write enable**
Bit 0	**Reserved**

Register 76h Attribute of shadow RAM for BIOS area (default = 00h)

Bit 7	**Read enable for shadow RAM of BIOS area 0F0000-0FFFFFh**
Bit 6	**L1/L2 cacheable for shadow RAM of BIOS area 0F0000-0FFFFFh**
Bit 5	**Write enable for shadow RAM of BIOS area 0F0000-0FFFFFh**
Bit 4	**Reserved**
Bit 3	**Shadow RAM enable for PCI access**
Bits 2:0	**Reserved**

Register 77h Characteristics of non-cacheable area (default = 00h)

Bits 7:4 **Reserved**

Bit 3 **Allocation of Non-Cacheable Area I**
0: Local DRAM
1: PCI Bus. The local DRAM is disabled.

Bit 2 **Non-Cacheable Area I Enable**
0: Disable
1: Enable

Bit 1 **Allocation of Non-Cacheable Area II**
0: Local DRAM
1: PCI Bus. The local DRAM is disabled.

Bit 0 **Non-Cacheable Area II Enable**
0: Disable
1: Enable

Register 78h~79h Allocation of Non-Cacheable area I (default = 00h)

Bits 15:13 Size of Non-Cacheable Area I (within 384 MBytes)

000: 64KB	100: 1MB
001: 128KB	101: 2MB
010: 256KB	110: 4MB
011: 512KB	111: 8MB

Bits 12:0 A28~A16 Non-Cacheable Area I (within 384 MBytes)

Register 7Ah~7Bh Allocation of Non-Cacheable area II (default = 00h)

Bits 15:13 Size of Non-Cacheable Area II (within 384 MBytes)

000: 64KB	100: 1MB
001: 128KB	101: 2MB
010: 256KB	110: 4MB
011: 512KB	111: 8MB

Bits 12:0 A28~A16 Non-Cacheable Area II (within 384 MBytes)

Register 7Ch~7Fh Reserved

Register 80h PCI master characteristics

Bits 7:5 **Burstable Length Selection**
000: 256B
001: 512B
010: 1KB
011: 2KB

100: 4KB
Others: reserved
Maximum burstable address range in PCI master accessing main memory when 32-bit DRAM organization is employed with 256K or 512K type DRAM maximum burstable range reduces to 2KB only because the physical page size is 2KB in this situation. Thus, never program these bits to 4KB in 32 bit DRAM organization.

Bit 4 **TRDY# assertion timing in PCI master read cycle**
0: Assert TRDY# after prefetching 2 QWs
1: Assert TRDY# after prefetching 1 Qws

Bit 3 **Advanced snoop in PCI master write cycle**
0: Disable
1: Enable

Bit 2 **Advanced snoop in PCI master read cycle**
0: Disable
1: Enable

Bit 1 **PCI bus using synchronous mode**
0: Disable (default)
1: Enable

Bit 0 **Reserved**

Register 81h

Bit 7 **The timing for SiS Chip to prefetch FPM DRAM data to CTPFF (CPU to PCI FIFO)**
0: 1 CPUCLK delay from the assertion of CAS# (recommended in 50Mhz)
1: 2 CPUCLK delay from the assertion of CAS# (recommended in 60/66/75Mhz)

Bit 6 **The timing for SiS Chip to prefetch EDO DRAM data to CTPFF**
0: 1 CPUCLK delay from the assertion of CAS# (recommended in 50Mhz)
1: 2 CPUCLK delay from the assertion of CAS# (recommended in 60/66/75Mhz)

Bit 5 **Reserved**

Bit 4 **This bit must be programmed to “0”.**

Bit 3 **Synchronous DRAM burst read in PCI master read cycle**
0: Disable (default)
1: Enable

Bit 2 **Enable CPU to L2/DRAM and PCI Peer-to-Peer concurrency mode**
0: Disable
1: Enable

Bit 1 **Internal use only, must be 0.**

Bit 0 **Reserved**

Register 82h

Bit 7 **PCI master write main memory cycles**
0: Faster (default)
1: Slower (Recommended at 75 MHz)

Bit 6 **PEADS timing control in PCI master to main memory cycles**
0: Faster (default)
1: Slower (Recommended at 75 MHz)
When PCI master initiating memory cycle, SiS Chip will check ROW address on the CPU clock rising edge that PEADS is active. PEADS is expected as the first EADS# of every PCI bus transaction. Note that PEADS is internal signal.

Bit 5 **Enhanced performance for the *Memory Write and Invalidate* of PCI bus command**
0: Disable (default)
1: Enable
Note: This bit must set to 0 if using 512K cache size and TAG address is set to 8 bits.

Bit 4 **Read prefetch for the *Memory Read* of PCI bus command**
0: Enable (default)
1: Disable
If enabled, the *Memory Read Multiple and Memory Read Line* of PCI bus commands always do prefetch.

Bits 3:2 **PCI Target Bridge of SiS Chip Initial Latency Timer**
00: Disable (default)
01: 16 PCI Clocks
10: 24 PCI Clocks
11: 32 PCI Clocks

Bit 1 **PCI Target Bridge of SiS Chip Subsequent Latency Timer**
0: Disable (default)
1: Enable

Bit 0 **Propagation delay time of AD bus control**
0: Normal (Recommended)
1: Slower
When set, the SiS Chip timing is adjusted to serve those bus master agents that do not follow the PCI specification to have 12ns max. propagation delay time of AD in the address phase.

Register 83h CPU to PCI characteristics (default 00)

Bit 7 **Fast gate A20 emulation**
0: Disable
1: Enable (recommended)

Bit 6 **Fast reset emulation**
0: Disable

1: Enable (recommended)

Bit 5 **Fast reset latency control**
0: 2us
1: 6us

Bit 4 **Fast back-to-back function when the PCI cycle hit IDE or prefetchable area.**
0: Disable
1: Enable (recommended)

Bit 3 **CPU to PCI post write rate control**
0: 4T
1: 3T (recommended)

Bit 2 **IDE Data port post write function**
0: Disable
1: Enable (recommended)

Bit 1 **CPU to PCI burst memory write**
0: Disable
1: Enable (recommended)

Bit 0 **CPU to PCI post write function**
0: Disable
1: Enable (recommended)

Register 84h~85h PCI grant timer

Bits 15:0 **16 bits PCI Grant Timer**
The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1)
Note: Unit : PCI clock

Register 86h CPU idle timer

Bits 7:0 **8 bits CPU idle timer**
The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1)
Note: Unit : PCI clock

Register 87h Miscellaneous register (default 00)

Bit 7 **CPU to PCI Bridge Synchronous Mode**
0: Disable
1: Enable (recommended)

Bit 6 **CPU involve arbitration**
0: Disable
1: Enable (recommended)

Bit 5 **The latency of ADS# to FRAME#**

0: Normal
1: Fast

Bit 4 **CPU latency timer Testing Mode**
0: Disable
1: Enable

Bit 3 **2nd Half PCI Cycle of a 64-bit Access Retried Behavior**
0: Continue Retry (Recommended)
1: Back-Off CPU

Bit 2 **PCI Lock Function Enable**
0: Disable
1: Enable

Bits 1:0 **Reserved**

Register 88h~89h Base address of fast back-to-back area

Bits 15:0 **16 bits address A[31:16]**

Register 8Ah~8Bh Size of fast back-to-back area

Bits 15:0 **Mask bits**
0: Enable mask
1: Disable mask
The SiS Chip will compare the current address with the base address (Register 8A~8B) by using the mask bits to determine whether to execute the fast-back-to-back or not. If the corresponding mask bit is 1, SiS chip will compare the current address bit with the base address bit. If the corresponding mask bit is 0, SiS Chip will not make the comparison.

Register 8Ch/8Dh/8Eh/8Fh General Purpose Register

Bits 7:0 **Reserved**

Following two registers mainly defines the enable bits for the events monitored by System Standby timer. If any monitored event occurs during the programmed time, the System standby timer will be reloaded and starts to count down again.

Register 90h Legacy PMU control register

Bit 7 **Hard Disk Port 1 Enable**
When set, any I/O access to the Hard Disk port 1 (1F0-1F7h or 3F6h) will cause the System Standby timer be reloaded.

Bit 6 **Keyboard port Enable**
When set, any I/O access to the keyboard Ports (60h or 64h) will cause the System Standby timer be reloaded.

Bit 5 **Serial Port 1 Enable**

When set, any I/O access to the Serial Ports (3F8-3FFh or 3E8-3EFh) will cause the System Standby timer be reloaded.

Bit 4 **Serial Port 2 Enable**

When set, any I/O access to the Serial Ports (2F8-2FFh or 2E8-2EFh) will cause the System Standby timer be reloaded.

Bit 3 **Parallel Port Enable**

When set, any I/O access to the Parallel ports (278-27Fh, 378-37Fh or 3BC-3BEh) will cause the System Standby timer be reloaded.

Bit 2 **Hold Enable**

When set, any event from the ISA master or the PCI Local Master will cause the System Standby timer be reloaded.

Bit 1 **IRQ1~15, NMI**

When set, any event from the IRQ1-15 or NMI which is defined by PCI to ISA bridge configuration Register 74 and 75 will cause the System Standby timer be reloaded.

Bit 0 **Monitor Ring event enable**

If this bit is set, an event from the RING will cause the System Standby timer be reloaded.

Register 91h Address trap for Legacy PMU function

Bit 7 **Programmable 10 bit I/O Port Enable**

When set, any I/O access to the address will cause the System Standby timer be reloaded. The address is defined in Registers 96h and 97h.

Bit 6 **Programmable 16 bit I/O Port Enable**

When set, any I/O access to the address will cause the System Standby timer be reloaded. The address is defined in Registers 98h and 99h.

Bit 5 **A0000h - AFFFFh or B0000 - BFFFFh Address trap**

When set, any memory access to the address range will cause the System Standby timer to be reloaded.

Bit 4 **C0000h - C7FFFh Address trap**

When set, any memory access to the address range will cause the System Standby timer to be reloaded.

Bit 3 **3B0-3BFh, 3C0-3CFh, 3D0-3DFh Address trap**

When set, any I/O access to the I/O addresses will cause the System Standby timer to be reloaded.

Bit 2 **Secondary Drive port**

When set, any I/O access to the secondary drive port (170-177h, 376h) will reload the system standby timer.

Bits 1:0 **System Standby Timer Slot**

11 : 8.85 milli seconds
10 : 70 milli seconds
01 : 1.1 seconds
00 : 9 seconds

Register 92h

Bits 7:5 Define the Timer monitored events for the Monitor standby timer .
Bits 4:2 Define the wake-up events from System standby state.
Bits 1:0 Define the events to de-assert the STPCLK#.

Bit 7 **IRQ 1-15, NMI**

When set, any event from the IRQ1-15 or NMI which is defined by PCI to ISA bridge configuration Register 76 and 77 will cause the Monitor standby timer be reloaded.

Bit 6 **HOLD**

When set, any event from the ISA master or the PCI local master will cause the Monitor standby timer be reloaded.

Bit 5 **Reload Monitor Timer From RING signal**

When set, Monitor standby timer will be reloaded when RING is asserted.

Bit 4 **IRQ 1-15, NMI**

When enabled, any event from the IRQ1-15 or NMI which is defined by PCI to ISA bridge configuration Register 76 and 77 will bring the Monitor back to the Normal state from the Standby state.

Bit 3 **HOLD**

When enabled, any event from the ISA master or the PCI local master will bring the Monitor back to the Normal state from the Standby state.

Bit 2 **Ring Wakeup Enable**

If this bit is set, it will bring the Monitor back to the Normal state from the Standby state when RING is asserted.

Bit 1 **IRQ 1-15, NMI**

When enabled, any event from the IRQ1-15, GPIO or NMI which are defined by PCI to ISA bridge configuration Register 74h and 75h will de-assert the STPCLK#.

Bit 0 **HOLD**

When enabled, any event from the ISA master or the PCI local master will de-assert the STPCLK#.

Register 93h STPCLK# and APM SMI control

Bit 7 **INIT**

When enabled, an event from the INIT will de-assert the STPCLK#.

Bit 6 **Ring Wakeup Enable**

When enabled, system will wake up from standby mode to de-assert the STPCLK# when RING is asserted.

Bit 5 **STPCLK# Enable**

When set, writing a '1' to bit 3 of Register 93h will cause the STPCLK# to become active. This bit can be cleared.

Bit 4 **Throttling Enable**

When set, writing a '1' to bit 3 of Register 93h will cause the STPCLK# throttling state to become active. The throttling function can be disabled by clearing this bit.

Bit 3 **STPCLK# Control**

When this bit is set, the STPCLK# will be asserted or the Throttling function will be enabled depending on bits 5 and 4. If both bits 5 and 4 are enabled, the system will do the throttling function.

Bit 2 **Pin Definition Select for EXTSMI#/TURBO**

0: TURBO

1: EXTSMI#

The EXTSMI# function can be disabled by programming register 9Bh bit 1 to "0".

Bit 1 **APM SMI**

When Register 9Bh bit 0 is enabled, and a '1' is written to this bit, an SMI is generated. It is used by the software controlled SMI function like APM. This bit should be cleared at the end of the SMI handler.

Bit 0 **Deturbo function**

0: Disable

1: Enable

Register 94h Cyrix 6x86 and PMU function control

Bit 7 **Cyrix 6x86 SMAC access**

It must be set whenever the 6x86 CCR1 bit 2 is set and cleared if CCR1 bit 3 is cleared.

Bit 6 **Cyrix 6x86 MMAC access**

If set, access to address within SMM space is conducted to main memory instead of SMM area. It must be set whenever the 6x86 CCR1 bit 3 is set and cleared if CCR1 bit 3 is cleared.

In the 6x86's specification, the SMIACT# will be de-asserted when MMAC is set and re-asserted after it is cleared. This allows the SMI service routine to access normal memory area instead of SMM memory area.

Bit 5 **Cyrix 6x86 CPU**

It should be set if Cyrix 6x86 CPU is present.

Bit 4 **External SMI# Wakeup capability**

0: Disable

1: Enable

When enabled, an event from External SMI will de-assert the STPCLK#.

Bit 3 **Flush Function Block Mode**

0: Un-block

1: Block

It is suggested to block the FLUSH# (Deturbo Mode) when the STPCLK# is asserted.

Bit 2:0 **Reserved**

Register 95h

Bit 7 **IRQ SMI enable.**
When set, any unmasked event defined at PCI to ISA Bridge configuration Register 72h-73h will cause the SMI to be generated.

Bit 6 **IRQ SMI status.**
This bit is set when the bit 7 of this Register is enabled and the corresponding event is active.

Bit 5 **Throttling exit control**
When Register 9B, bit 5 (Throttling wake up SMI enable) is set and STPCLK# is at throttling mode, set this bit will cause the STPCLK# de-asserted and SMI# generated.

Bit 4 **USB SMI enable**
When this bit is set, a SMI# can be generated by USB controller.

Bit 3 **USB SMI request.**
This is an USB SMI Request start bit. When the bit 4 of this register is set and the USB controller asserts a control signal to generated SMI#, this bit is set.

Bit 2 **Reserved**

Bits 1:0 **Legacy PMU test mode**
00: Normal operation
01: Counter test mode
10: Fast test mode
11: Reserved

Register 96h Time slot and Programmable 10-bit I/O port definition

Bits 7:6 **Define the time slot of the Monitor Standby timer**
00 : 6.6 seconds
01 : 0.84 seconds
10 : 13.3 milli-seconds
11 : 1.6 milli-seconds

Bits 5:3 **Programmable 10-bit I/O port address mask bits**
000 : No mask
001 : A0 masked
010 : A1-A0 masked
011 : A2-A0 masked
100 : A3-A0 masked
101 : A4-A0 masked
110 : A5-A0 masked
111 : A6-A0 masked

Bit 2 **Reserved**

Bits 1:0 **Programmable 10-bit I/O port address bits A1, A0.**

Bits 1:0 correspond to the address bits A1 and A0.

Register 97h Programmable 10-bit I/O port address bits A9~A2

Bits 7:0 Define the programmable 10-bit I/O port address bits A[9:2].

Register 98h~99h Programmable 16-bit I/O port

Bits 15:0 Define the Programmable 16-bit I/O port.

Following two registers define the enable status of the devices in SMM. The bits are set when the devices are in standby state and cleared when the respective devices are in normal state.

Register 9Ah System Standby Timer events control

Bit 7 System Standby SMI Enable
When no non-masked event occurs during the programmed duration of the system standby timer, the timer expires. If this bit is enabled, the SMI# is generated and the system enters the System Standby state.

Bit 6 Programmable 10-bit I/O port wake up SMI Enable
When set, any I/O access to this port will be monitored to generate the SMI# to wake up this I/O port from the standby state to the Normal state. This bit is enabled only when the I/O port is in the Standby state.

Bit 5 Programmable 16-bit I/O port wake up SMI Enable
When set, any I/O access to this port will be monitored to generate the SMI# to wake up this I/O port from the standby state to the Normal state. This bit is enabled only when the I/O port is in the Standby state.

Bit 4 Parallel ports wake up SMI Enable
When set, any I/O access to the parallel ports will be monitored to generate the SMI# to wake up the parallel ports from the standby state to the Normal state. This bit is enabled only when the parallel ports are in the Standby state.

Bit 3 Serial port 1 wake up SMI Enable
When set, any I/O access to the serial port 1 will be monitored to generate the SMI# to wake up the serial ports from the standby state to the Normal state. This bit is enabled only when the serial port 1 are in the Standby state.

Bit 2 Serial port 2 wake up SMI Enable
When set, any I/O access to the serial port 2 will be monitored to generate the SMI# to wake up the serial ports from the standby state to the Normal state. This bit is enabled only when the serial port 2 are in the Standby state.

Bit 1 Hard Disk port 1 SMI Enable
When set, any I/O access to the hard disk port 1 will be monitored to generate the SMI# to wake up the hard disk from the standby state to the Normal state. This bit is enabled only when the hard disk port 1 is in the Standby state.

Bit 0 Hard Disk port 2 SMI Enable

When set, any I/O access to the hard disk port 2 will be monitored to generate the SMI# to wake up the hard disk from the standby state to the Normal state. This bit is enabled only when the hard disk port 2 is in the Standby state.

Register 9Bh Monitor Standdby Timer events control

Bit 7 **Monitor Standby SMI Enable**

0 : Disable

1 : Enable

When there is no access from the IRQ1-15, HOLD, RING and NMI during the programmed time of the Monitor Standby Timer, the timer expires. If this bit is set, an SMI is generated to bring the Monitor to the standby state.

Bit 6 **Monitor wake up SMI Enable**

When set, any event from the IRQ1-15, any bus master request, RING or NMI will be monitored to generate the SMI# to wake up the monitor from the standby state to the normal state.

Bit 5 **Throttling wake up SMI Enable**

When set, any unmasked event from the NMI, INIT, IRQ1-15, GPIO, EXTSMI#, RING or any bus master request will cause an SMI to be generated to bring the system back to the Normal state from the throttling state.

Bit 4 **System wake up SMI Enable**

When set, any unmasked event from the NMI, INIT, IRQ1-15, EXTSMI#, RING, GPIO or any bus master request will cause an SMI to be generated to bring the system back to the Normal state from the standby state.

Bit 3 **Keyboard wake up SMI Enable**

When set, any I/O access to the keyboard ports will be monitored to generate the SMI# to wake up the keyboard ports from the standby state to the Normal state. This bit is enabled only when the keyboard ports are in the Standby state.

Bit 2 **RING SMI Enable**

If this bit is set, it enables the SMI request from RING activity.

Bit 1 **External SMI Enable**

When set, the break switch (via EXTSMI#) can be pressed to generate the SMI# for the system to enter the Standby state.

Bit 0 **Software SMI Enable**

When set, an I/O write to bit 1 of register 93h will generate an SMI.

Following two registers define the SMI request status. If the respective SMI enable bit is set, each specific event will cause the respective bit to be set. The asserted bit should be cleared at the end of the SMI handler.

Register 9Ch SMI Request events status 0

Bit 7 **System Standby SMI Request**

This bit is set when the system standby timer expires.

Bit 6 **Programmable 10-bit I/O port wake up Request**

This bit is set when there is an I/O access to the port.

Bit 5 **Programmable 16-bit I/O port wake up Request**
This bit is set when there is an I/O access to the port.

Bit 4 **Parallel ports wake up Request**
This bit is set when the parallel ports are accessed.

Bit 3 **Serial port 1 wake up Request**
This bit is set when the serial port 1 are accessed.

Bit 2 **Serial port 2 wake up Request**
This bit is set when the serial port 2 are accessed.

Bit 1 **Hard Disk port 1 wake up Request**
This bit is set when the hard disk port 1 is accessed.

Bit 0 **Hard Disk port 2 wake up Request**
This bit is set when the hard disk port 2 is accessed.

Register 9Dh SMI Request events status 1

Bit 7 **Monitor Standby SMI Request**
This bit is set when the Monitor Standby Timer expires. This bit should be cleared at the end of the SMI handler.

Bit 6 **Monitor wake up Request**
This bit is set when there is an event from the IRQ1-15, any bus master request or NMI which are defined by bits 2, 3, 4 of Register 92, and the Monitor is in the standby state.

Bit 5 **Throttling wake up SMI Request**
This bit is set when there is any unmasked event from the NMI, INIT, IRQ1-15, RING, External SMI or any bus master request at the throttling state of the system.

Bit 4 **System wake up SMI Request**
This bit is set when there is any unmasked event from the NMI, INIT, IRQ1-15, or RING, External SMI or any bus master request at the standby state of the system.

Bit 3 **Keyboard ports wake up Request**
This bit is set when the keyboard ports are accessed.

Bit 2 **SMI request from RING**
This bit is set when there is RING activity.

Bit 1 **External SMI Request**
This bit is set when the break switch (via EXTSMI#) is pressed.

Bit 0 **Software SMI Request**
This bit is set when an I/O write to the bit 1 of register 93h and bit 0 of register is 1.

Register 9Eh STPCLK# Assertion Timer (default = FFh)

Bits 7:0 **This register defines the period of the STPCLK# assertion time.**

Bits[7:0] define the period of the STPCLK# assertion time when the STPCLK# enable bit is set. The timer will not start to count until the Stop Grant Special Cycle is received. The timer slot is 35 us.

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) x 35us

Register 9Fh STPCLK# De-assertion Timer (default = FFh)

Bits 7:0 This register defines the period of the STPCLK# de-assertion time.

Bits[7:0] define the period of the STPCLK# de-assertion time when the STPCLK# enable bit is set. The timer starts to count when the STPCLK# assertion timer expires. The timer slot is 35us.

When these two registers are read, the current values are returned.

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) x 35us

Register A0h~A1h Monitor Standby Timer (default = 00FFh)

Bits 15:0 Define the 16 bits Monitor standby timer.

It is a count-down timer and the time slot is programmable for 6.6s, 0.84s, 13.3 ms or 1.6ms. The value programmed to this register is loaded when the timer is enabled and the timer starts counting down. The timer is reloaded when an event from the IRQ1-15, HOLD or NMI occurs before the timer expires. When this register is read, the current value is returned.

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) x Time slot

NOTE: The setting of Time slot please refer to register 96h bits 7:6.

Register A2h System Standby Timer (default = FFh)

Bits 7:0 The register defines the duration of the System Standby Timer.

When the System Standby Timer expires, the system enters System Standby State. If any non-masked event occurs before the timer expires, the timer is reloaded with programmed number and the timer starts counting down again.

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) x Time slot

NOTE: The setting of Time slot please refer to register 91h bits 1:0.

Register A3h SMRAM access control and Power supply control (default = 00h)

Bits 7:6 SMRAM Area Remapping Control

00: EL to EL(32K)
01: EL to AL(32K)
10: EL to BL(32K)
11: A to A(64K)

Bit 5 Reserved

Bit 4 **SMRAM Access Control**

0: The SMRAM area can only be accessed during the SMI handler.

1: When set, the SMRAM area can be used. This bit can be set whenever it is necessary to access the SMRAM area. It is cleared after the access is finished.

Bits 3:0 **Reserved**

6.2 PCI to ISA Bridge Configuration Space

Device	IDSEL	Function Number
PCI to ISA bridge	AD12	0000b

Registers 00h~01h Vendor ID

Bits 15:0 = 1039h (Read Only)

Registers 02h~03h Device ID

Bits 15:0 = 0008h (Read Only)

Registers 04h~ 05h Command Port

Bits 15:4 **Reserved. Read as 0's**

Bit 3 **Monitor Special Cycle Enable**

Bit 2 **Behave as Bus Master Enable**

Bit 1 **Respond to Memory Space Accesses (Read/Writable)(Default=0)**

This bit is read/writable and should be set to 1.

Bit 0 **Respond to I/O Space Accesses (Read/Writable) (Default =0)**

This bit is read/writable and should be set to 1.

Registers 06h~07h Status

Bits 15:14 **Reserved. Read as 0's**

Bit 13 **Received Master-Abort**

When the SiS Chip generates a master-abort, this bit is set to a 1. This bit is cleared to 0 by writing a 1 to this bit.

Bit 12 **Received Target-Abort**

When the SiS Chip receives a target-abort, this bit is set to a 1. Software clears this bit to 0 by writing a 1 to this bit location.

Bit 11 **Reserved. Read as a 0**

Bits 10:9 **DEVSEL# Timing**

The SiS Chip always generates DEVSEL# with medium timing, these two bits are always set to 01.

Bits 8:0 **Reserved. Read as 0's.**

Register 08h Revision ID

Bits 7:0 **01h (Read Only)**

Register 09h~0Bh Class Code

Bits 23:0 **060100h (Read Only)**

Register 0Ch Cache Line Size

Bits 7:0 **00h**

Register 0Dh Master Latency Timer

Bits 7:0 **Master latency timer**
The default value is FFh, it menas 255 PCI clocks.
Unit: PCI Clocks

Register 0Eh Header Type

Bits 7:0 **80h (Read Only)**

Register 0Fh BIST

Bits 7:0 **80h (Read Only)**

Register 10h, 11h, 12h, 13h Reserved, always read as 0.

Register 40h BIOS Control Register

Bit 7 **Integrated ACPI Control**
0: Disable (Default)
1: Enable
When enabled, PCI to ISA bridge will response the access to ACPI registers according to the ACPI Base Address. The ACPI base address can be programmed in 90h, 91h.

Bit 6 **ISA Master action Control**
This bit must be set to “1” to prevent the deadlock which occuring in forwarding 64 bit memory read cycle to ISA side when delay transaction is enabled.

Bit 5 **Enable/Disable Delayed Transaction**
0: Disable (Default)
1: Enable

Bit 4 **PCI Posted Write Buffer Enable**
0: Disable (Default)

1: Enable

Bits [3:0] determine how the SiS Chip responds to F segment, E segment, and extended segment (FFF80000-FFFDFFFF) accesses. SiS Chip will positively respond to extended segment access when bit 0 is set. Bit 1, combining with bits [3:2], enables SiS Chip to respond to E segment access.

Bit 3 Positive Decode of Upper 64K BYTE BIOS Enable.

Bit 2 BIOS Subtractive Decode Enable.

Bits [3:2]	F segment		E segment		Comment
	+	-	+	-	
00			√*		Chip positively responds to E segment access.
10	√		√*		Chip positively responds to E and F segment access.
Others		√			Chip subtractively responds to F segment access.

Note: Enabled if bit 1 is set.

Bit 1 Lower BIOS Enable.

Bit 0 Extended BIOS Enable. (FFF80000~FFFDFFFF)

Register 41h/42h/43h INTA#/INTB#INTC# Remapping Control Register

Bit 7 Remapping Control

0: Enable

1: Disable (Default)

When enabled, INTA#/INTB#/INTC#, is remapped to the PC compatible interrupt signal specified in IRQ remapping table. This bit is set to 1 after reset.

Bits 6:4 Reserved. Read as 0's.

Bits 3:0 IRQx Remapping table.

Bits	**IRQx#**	**Bits**	**IRQx#**	**Bits**	**IRQx#**	**Bits**	**IRQx#**
0000	reserved	0101	IRQ5	1010	IRQ10	1111	IRQ15
0001	reserved	0110	IRQ6	1011	IRQ11		
0010	reserved	0111	IRQ7	1100	IRQ12		
0011	IRQ3	1000	reserved	1101	reserved		
0100	IRQ4	1001	IRQ9	1110	IRQ14		

Note: The difference INT[A:D]# can be remapped to the same IRQ signal, but this IRQ signal should be set to level sensitive.

Register 44h INTD# Remapping Control Register

Bit 7 Remapping Control

0: Enable

1: Disable (Default)

When enabled, INTD# is remapped to the PC compatible interrupt signal specified in IRQ remapping table. This bit is set to 1 after reset.

Bits 6:5 Reserved. Read as 0's.

Bit 4 Access APC Control Register(APCREG_EN)

0: Disable
1: Enable (Default)

Bits 3:0 **IRQ Remapping table.**

Bits	IRQx#	Bits	IRQx#	Bits	IRQx#	Bits	IRQx#
0000	reserved	0101	IRQ5	1010	IRQ10	1111	IRQ15
0001	reserved	0110	IRQ6	1011	IRQ11		
0010	reserved	0111	IRQ7	1100	IRQ12		
0011	IRQ3	1000	reserved	1101	reserved		
0100	IRQ4	1001	IRQ9	1110	IRQ14		

Note: The difference INT[A:D]# can be remapped to the same IRQ signal, but this IRQ signal should be set to level sensitive.

Register 45h (Default =00h)

Bits 7:6 **ISA Bus Clock Selection**
00: 7.159MHz
01: PCICLK/4
10: PCICLK/3

Bit 5 **Flash EPROM Control bit 0**

Bit 4 **Test bit for internal use only**
0: Normal mode
1: Test mode

Bit 3 **Access RTC Extended Bank Control(EXTEND_EN)**
0: Disable
1: Enable

Bit 2 **Flash EPROM Control Bit 1**

Bit 5	Bit 2	Operation
0	0	EPROM can be flashed
1	0	EPROM can't be flashed again
X	1	EPROM can be flashed whenever bit 5 is 0

Note: “X” means “Don’t care”.

Bits 1:0 **Reserved**

Register 46h (Default =00h)

Bits 7:6 **16-Bit I/O Cycle Command Recovery Time**
00: 5 BUSCLK
01: 4 BUSCLK
10: 3 BUSCLK
11: 2 BUSCLK

Bits 5:4 **8-Bit I/O Cycle Command Recovery Time**
00: 8 BUSCLK
01: 5 BUSCLK
10: 4 BUSCLK

11: 3 BUSCLK

Bit 3 **ROM Cycle Wait State Selection**

0: 4 wait states

1: 1 wait state

Bits 2:0 **Test bit for internal use only**

0: Normal Mode

1: Test Mode

Register 47h DMA Clock and Wait State Control Register (Default =00h)

Bit 7 **Reserved**

Bit 6 **Extended Terminal Count (TC)Hold Time**

0: The hold time of TC is compatible with Intel 8237

1: Extend the TC hold time by 1/2 DMACLK

Bits 5:4 **16-Bit DMA Cycle Wait State**

00 : 1 DMACLK

01 : 2 DMACLK

10 : 3 DMACLK

11 : 4 DMACLK

Bits 3:2 **8-Bit DMA Cycle Wait State**

00 : 1 DMACLK

01 : 2 DMACLK

10 : 3 DMACLK

11 : 4 DMACLK

Bit 1 **Extended DMAMEMR# Function**

0: Assertion of DMAMEMR# is delayed by one DMA clock cycle later than XIOR#

1: Assertion of DMAMEMR# is at the same time as XIOR#.

This bit is recommended to set to “1” to ensure that the assertion of DMAMEMR is earlier one DMA clock than the assertion of DMAIOW when the bit5 and bit3 of DMA command register are set to “0”.

Bit 0 **DMA Clock Selection**

0: 1/2 BUSCLK(Recommended)

1: BUSCLK

Register 48h ISA Master/DMA Memory Cycle Control Register 1 (Default =01h)

Bits 7:4 **Top of Memory size**

0000: 1 MByte

0001: 2 MByte

0010: 3 MByte

0011: 4 Mbyte

:

:
1101: 14 MByte
1110: 15 MByte
1111: 16 Mbyte
The ISA master or DMA memory access cycles will be forwarded to PCI bus when the address fall within the programmable region defined by bits[7:4]. The base address of the programmable region is 1Mbyte, and the top addresses is programmed in 1MByte increments from 1MByte to 16MByte. All memory cycles will be forwarded to PCI bus besides the cycle fall within memory hole defined in register 4Ah and 4Bh.

ISA master and DMA memory cycles to the following memory regions will be forwarded to PCI bus if they are enabled.

Bit 3 **E0000h-EFFFFh Memory Region**
0: Disable
1: Enable, the cycle is forwarded to PCI bus.

Bit 2 **A0000h-BFFFFh memory Region**
0: Disable
1: Enable (The cycle is forwarded to PCI bus.)

Bit 1 **80000h-9FFFFh Memory Region**
0: Disable
1: Enable (The cycle is forwarded to PCI bus.)

Bit 0 **00000h-7FFFFh Memory Region**
0: Disable
1: Enable(The cycle is forwarded to PCI bus.)

Register 49h ISA Master/DMA Memory Cycle Control Register 2 (Default =00h)

Bit 7	**DC000h-DFFFFh Memory Region**	**Bit 6**	**D8000h-DBFFFh Memory Region**
Bit 5	**D4000h-D7FFFh Memory Region**	**Bit 4**	**D0000h-D3FFFh Memory Region**
Bit 3	**CC000h-CFFFFh Memory Region**	**Bit 2**	**C8000h-CBFFFh Memory Region**
Bit 1	**C4000h-C7FFFh Memory Region**	**Bit 0**	**C0000h-C3FFFh Memory Region**

0: Disable
1: Enable
ISA master and DMA memory cycles to the following memory regions will be forwarded to PCI bus if they are enabled.

Register 4Ah and register 4Bh are used to define the ISA address hole. The ISA address hole is located between 1Mbyte and 16MByte, and sized in 64KByte increments. ISA master and DMA memory cycles fall within this hole will not be forwarded to PCI bus. Register 4Ah and 4Bh are used to define the bottom and top address of the hole respectively. The hole is located between top and bottom address, and the bottom and top address must be at or above 1MByte. If bottom address is greater than top address, the ISA address hole is disabled.

Register 4Ah ISA Master/DMA Memory Cycle Control Register 3 (Default =10h)

Bits 7:0 Bottom address of the ISA Address Hole [A23:A16]

Register 4Bh ISA Master/DMA Memory Cycle Control Register 4 (Default =0Fh)

Bits 7:0 Top address of the ISA Address hole [A23:A16]
This register is used to define the top address of the ISA Address hole

Registers 4Ch/4Dh/4Eh/4Fh Initialization Command Word 1/2/3/4 Mirror Register I

Bits 7:0 ICW1 to ICW4 of the built-in interrupt controller (master) can be read from 4Ch to 4Fh.

Registers 50h/51h/52h/53h Initialization Command Word 1/2/3/4 mirror Register II

Bits 7:0 ICW1 to ICW4 of the built-in interrupt controller (slave) can be read from 50h to 53h.

Registers 54h/55h Operational Control Word 2/3 Mirror Register I

Bits 7:0 OCW2 to OCW3 of the built-in interrupt controller (master) can be read from 54h to 55h.

Registers 56h/57h Operational Control Word 2/3 Mirror Register II

Bits 7:0 OCW2 to OCW3 of the built-in interrupt controller (slave) can be read from 56h to 57h.

Register 58h Counter Access Ports Mirror Register 0

Bits 7:0 Low byte of the initial count number of Counter 0 in the built-in CTC can be read from 58h.

Register 59h

Bits 7:0 High byte of the initial count number of Counter 0 in the built-in CTC can be read from 59h.

Register 5Ah

Bits 7:0 Low byte of the initial count number of Counter 1 in the built-in CTC can be read from 5Ah.

Register 5Bh

Bits 7:0 High byte of the initial count number of Counter 1 in the built-in CTC can be read from 5Bh.

Register 5Ch

Bits 7:0 **Low byte of the initial count number of Counter 2 in the built-in CTC can be read from 5Ch.**

Register 5Dh

Bits 7:0 **High byte of the initial count number of Counter 2 in the built-in CTC can be read from 5Dh.**

Register 5Eh

Bits 7:0 **Control word (43h) of the built-in CTC can be read from 5Eh.**

Register 5Fh

Bits 7:6 **Reserved**
Bit 5 **CTC write count pointer status for counter 2**
Bit 4 **CTC write count pointer status for counter 1**
Bit 3 **CTC write count pointer status for counter 0**
Bit 2 **CTC read count pointer status for counter 2**
Bit 1 **CTC read count pointer status for counter 1**
Bit 0 **CTC read count pointer status for counter 0**
0: LSB
1: MSB

Register 60h Mirror port
Bits 7:0 **The same value as ISA port 70h.**

Register 61h IDEIRQ Remapping Control Register

Bit 7 **IDEIRQ Remapping Control**
0: Enable
1: Disable (Default)
Bit 6 **Attribute of bits Control for Reg. 09h bit 1 and 3 in PCI IDE Configuration Space**
0: Read Only, read these two bits as ‘1’ and ‘1’. (Default)
1: Read/Writeable
Bits 5:4 **Reserved. Read as zero.**
Bits 3:0 **Interrupt Remapping Table**

Bits [3:0]	Remapped IRQ	Bits [3:0]	Remapped IRQ
0000	Reserved	1000	Reserved
0001	Reserved	1001	IRQ9
0010	Reserved	1010	IRQ10

0011	IRQ3	1011	IRQ11
0100	IRQ4	1100	IRQ12
0101	IRQ5	1101	Reserved
0110	IRQ6	1110	IRQ14
0111	IRQ7	1111	IRQ15

Register 62h USBIRQ Remapping Control Register

Bit 7 **USBIRQ Remapping Control**
0: Enable
1: Disable (Default)

Bit 6 **Integrated USB Control**
0: Disable
1: Enable

Bit 5 **USB Over_Current (OCI#) input polarity**
0: Low Active
1: High Active

Bit 4 **USB Power pin (OCO# or OCI2#,** corresponding to Reg. 6A bit6**) output/input polarity**
0: Low Active
1: High Active

Bits 3:0 **Interrupt Remapping Table**

Bits [3:0]	**Remapped IRQ**	**Bits [3:0]**	**Remapped IRQ**
0000	Reserved	1000	Reserved
0001	Reserved	1001	IRQ9
0010	Reserved	1010	IRQ10
0011	IRQ3	1011	IRQ11
0100	IRQ4	1100	IRQ12
0101	IRQ5	1101	Reserved
0110	IRQ6	1110	IRQ14
0111	IRQ7	1111	IRQ15

Register 63h GPCS0 Control Register

Bit 7 **GPCS0 Mode Control**
0: Output mode
1: Input mode (default)

Bit 6 **GPCS0 Input Active Level Control**
0: Active low (default)
1: Active high

Bit 5 **GPCS0 Input De-Bounce Filter Control**
0: Disable (default)
1: Enable
When this bit set to 1, the GPCS0 input goes through a de-bounce circuit.

Bit 4 **GPCS0 Output Status Control**
0: Output low (default)
1: Output high
When GPCS0 is programmed to a GPO pin function by Register 65h~66h bit 1:0 and set Register 63 bit 7 to "0" (output mode), this bit can be active.

Bit 3 **GPCS0 Status (When it is set to Input Mode)**
This bit is set when GPCS0 event is generated and it can be cleared by writting a "0" to this bit.

Bit 2 **Generated SMI# by GPCS0 Control**
0: Disable
1: Enable
Note: The Host to PCI configuration register 95h bit7 should be enabled.

Bit 1 **Control GPCS0 to reload system standby timer and exit system standby state**
0: Disable
1: Enable

Bit 0 **GPO Write Enable 0 Control**
0: Disable (GPCS0 signal)
1: Enable
If this bit is enabled, it controls the external 74LS374 TTL to buffer the external 8 GPOs signals for more peripheral devices control from the system data bus SD[7:0] by GPCS0 pin.

Register 64h GPCS1 Control Register

Bit 7 **GPCS1 Mode Control**
0: Output mode
1: Input mode (default)

Bit 6 **GPCS1 Input Active Level Control**
0: Active low (default)
1: Active high

Bit 5 **GPCS1 Input De-Bounce Filter Control**
0: Disable (default)
1: Enable
When this bit set to 1, the GPCS1 input goes through a de-bounce circuit.

Bit 4 **GPCS1 Output Status Control**
0: Output low (default)
1: Output high
When GPCS1 is programmed to a GPO pin function by Register 67h~68h bit 1:0 and set Register 64 bit 7 to "0" (output mode), this bit can be active.

Bit 3 **GPCS1 Status (When it is set to Input Mode)**
This bit is set when GPCS1 event is generated and it can be cleared by writting a "1" to this bit.

Bit 2 **Generated SMI# by GPCS1 Control**

0: Disable
1: Enable
Note: The Host to PCI configuration register 95h bit7 should be enabled.

Bit 1 **Control GPCS1 to reload system standby timer and exit system standby state**
0: Disable
1: Enable

Bit 0 **GPO Write Enable 1 Control**
0: Disable (GPCS1 signal)
1: Enable
If this bit is enabled, it controls the external 74LS374 TTL to buffer the external 8 GPOs signals for more peripheral devices control from the system data bus SD[7:0] by GPCS1 pin.

Register 65h~66h GPCS0 Output Mode Control Register

A 16-bit I/O space base address defined in bit[15:2] is used to cause GPCS0 to assert "active low" signal for subtractively decoded I/O cycles generated by PCI masters that fall in the range specified by this register. This register is available only when GPCS0 is set to output mode.

Bits 15:2 **A[15:2] of GPCS0 I/O Space Base Address**

Bits 1:0 **GPCS0 I/O Space Address Mask**
00: Mask A1, A0
01: Mask A2, A1, A0
10: Set GPCS0 to GPO function only (default)
11: Mask A3, A2, A1, A0

Registers 67h~68h GPCS1 Output Mode Control Register

A 16-bit I/O space base address defined in bit[15:2] is used to cause GPCS1 to assert "active low" signal for subtractively decoded I/O cycles generated by PCI masters that fall in the range specified by this register. This register is available only when GPCS1 is set to output mode.

Bits 15:2 **A[15:2] of GPCS1 I/O Space Base Address**

Bits 1:0 **GPCS1 I/O Space Address Mask**
00: Mask A1, A0
01: Mask A2, A1, A0
10: Set GPCS1 to GPO function only (default)
11: Mask A3, A2, A1, A0

Register 69h GPCS0/1 De-Bounce Control Register

Bits 7:6 **GPCS0 I/O Space Address Mask Control**
00: According to the setting of Reg. 66h bit[1:0]
01: Mask A0~A9
10: Mask A0~A10

11: Reserved

Note : This bit does not affect GPCS1

Bit 5 **Reserved**

Bit 4 **Power Off System Control**

Before enabling this function, Auto Power Control Register I bit 6 should be enabled. Once writing a '1' to this bit, system will be power off.

Bit 3:0 **De-bounce Count for GPCS0/1 De-Bounce Circuit**

The minimum value is 2. The timer-expire interval is calculated by the following equation : The timer-expire interval = (Counts-1)x0.6s

Register 6Ah ACPI/SCI IRQ Remapping Control Register

Bit 7 **ACPI/SCI IRQ Remapping Control**

1: Disable (default)

0: Enable

Bit 6 **Pin Definition Select for OCO#/OCI2#**

0: OCI2#

1: OCO#

Note: Register 62h bit4 can be used to select the polarity of OCI2# or OCO#.

Bit 5 **Pin Definition Select for GPIO8/USB Over_Current (OCI#)**

0: GPIO8

1: OCI#

Bit 4 **Pin Definition Select for GPIO7/OCO#/OCI2#**

0: GPIO7

1: OCO#/OCI2#

Note: Bit6 (Pin Definition Select for OCO#/OCI2#) have function only when Bit4 (Pin Definition Select for GPIO7/OCO#/OCI2#) is set to “1”.

Bits 3:0 **Interrupt Remapping Table**

Bits [3:0]	Remapped IRQ	Bits [3:0]	Remapped IRQ
0000	Reserved	1000	Reserved
0001	Reserved	1001	IRQ9
0010	Reserved	1010	IRQ10
0011	IRQ3	1011	IRQ11
0100	IRQ4	1100	IRQ12
0101	IRQ5	1101	Reserved
0110	IRQ6	1110	IRQ14
0111	IRQ7	1111	IRQ15

Register 6Bh

Bits 7:0 **Test bits. These bits should be programmed to all 0s.**

Register 6Ch

Bits 7:6	**Test bits. These bits should be programmed to all 0s.**
Bit 5	**IRQ13 Control** When this bit is set to “1”, IRQ13 will be routed to FERRN.(Default is 1)
Bit 4:2	**Test bits. These bits should be programmed to all 0s.**
Bit 1	**Enable/Disable The Reading Of All Base Registers In DMA Controller.** 0: Disable. (default) 1: Enable.
Bit 0	**Reserved.** This bit should be programmed to 0.

Register 6Dh (Default= 19h)

Bit 7	**Internal Test Bit for Keyboard and PS/2 Mouse Pin Swapping** 0: Normal Mode (Default) 1: Test Mode
Bit 6	**Pin Definition Select for GPCS1/SIRQ** 0: GPCS1 (default) 1: SIRQ
Bit 5	**Test bit, must be 0.**
Bit 4	**I^2C Bus Data Active Level Control** 0: Active Low 1: Active High (default)
Bit 3	**I^2C Bus Clock Active Level Control** 0: Active Low 1: Active High (default)
Bit 2	**I^2C Bus Control** 0: Disable (default) 1: Enable
Bit 1	**Hot Key Status** This bit is set when hot key (Ctrl+Alt+Backspace) is pressed and should be cleared at the end of SMI handler. This bit is meaningful only when internal KBC is used.
Bit 0	**Hot Key Control** 0: Disable 1: Enable (default) This bit is meaningful only when internal KBC is used.

Register 6Eh Software-Controlled Interrupt Request, Channels 7-0

Bit 7	**Interrupt Channel 7**
Bit 6	**Interrupt Channel 6**

Bit 5 **Interrupt Channel 5**

Bit 4 **Interrupt Channel 4**

Bit 3 **Interrupt Channel 3**

Bit 2 **Interrupt Channel 2**

Bit 1 **Interrupt Channel 1**

Bit 0 **Interrupt Channel 0**

Writing a 1 to these bits will cause the corresponding interrupt requests to be outstanding. This register defaults to all 0s.

Register 6Fh Software-Controlled Interrupt Request, channels 15-8

Bit 7 **Interrupt Channel 15**

Bit 6 **Interrupt Channel 14**

Bit 5 **Interrupt Channel 13**

Bit 4 **Interrupt Channel 12**

Bit 3 **Interrupt Channel 11**

Bit 2 **Interrupt Channel 10**

Bit 1 **Interrupt Channel 9**

Bit 0 **Interrupt Channel 8**

Writing a 1 to these bits will cause the corresponding interrupt requests to be outstanding.

Register 70h (Default=12h)

Bit 7 **Enable/Disable the prefetch/postwrite of the ISA master and DMA controller**

0: Disable.

1: Enable.

Bit 6 **Enable/Disable IOR# and MEMR# cycles extended by 1/2 BCLK**

0: Disable

1: Enable.

Bit 5 **Test bit. This bit should be programmed to 0**

Bit 4 **Pin Definition Select for KLOCK#/GPIO0**

0: GPIO0

1: KLOCK#

Bit 3 **Integrated Keyboard Controller Status Control**

0: Disable Integrated Keyboard Controller

1: Enable Integrated Keyboard Controller

Bit 2 **Integrated PS/2 Mouse Interface Status Control**

0: Disable Integrated PS/2 Mouse Interface

1: Enable Integrated PS/2 Mouse Interface

Note: This bit has function only when Bit3 (Integrated Keyboard Controller) is enabled.

Bit 1 **Built-in RTC Status (Read Only)**

0: Not used

1: Used

When built-in RTC is used, this bit is set to 1.

Bit 0 **Test bit. This bit should be programmed to 0.**

Register 71h Type-F DMA Control Register (Default= 00h)

This register is used to set which DMA channel can perform type-F DMA transfers. A “1” on any bit sets the corresponding DMA channel to perform type-F DMA transfers. This register is available only when the Register 70h bit 7 is enabled in PCI to ISA bridge configuration register.

Bit 7 **DMA Channel 7**

Bit 6 **DMA Channel 6**

Bit 5 **DMA Channel 5**

Bit 4 **Reserved**

Bit 3 **DMA Channel 3**

Bit 2 **DMA Channel 2**

Bit 1 **DMA Channel 1**

Bit 0 **DMA Channel 0**

Register 72h~73h SMI Triggered By IRQ Control

When disabled, any event from the corresponding IRQ will cause the system to generate SMI. This register is only meaningful when the Host to PCI bridge configuration register 95h, bit 7 is enabled.

Bits 15:3 **Corresponds To The Mask Bits Of IRQ 15-3**

Bit 2 **Reserved**

Bit 1 **Corresponds To The Mask Bit Of IRQ1**

Bit 0 **Reserved**

0: Disable (default)

1: Enable

Register 74h~75h System Standby Timer Reload, System Standby State Exit And Throttling State Exit Control

When disabled, any event from the corresponding IRQ and NMI will cause the system to exit the system standby state, exit the throttling state or reload the system standby timer, which are depended on Legacy PMU register setting.

Bits 15:3,1 **Corresponds To The Mask Bits Of IRQ 15-3,1**

Bit 2 **Reserved**

Bit 0 **Corresponds To The Mask Bit Of NMI**
0: Disable (default)
1: Enable

Register 76h~77h Monitor Standby Timer Reload And Monitor Standby State Exit Control

When disabled, any event from the corresponding IRQ/NMI will cause the system to exit the monitor standby state or reload the monitor standby timer, which are depended on Legacy PMU register setting.

Bits 15:2 **Corresponds To The Mask Bits of IRQ 15-2**

Bit 2 **Reserved**

Bit 0 **Corresponds To The Mask Bit of NMI**
0: Disable (default)
1: Enable

Register 80h~81h DDMA Control Register

Attribute: Write/Read
Default: 0000h

Bits 15:4 **DMA remap base address**
There is only one DMA Slave Base Address Register, and all of the legacy DMA channels will be grouped into 128 bytes block. (16 bytes times 8 channels).

Bits 3:1 **Reserved**

Bit 0 **DMA Master Enable**
0: DMA remapping disable. All accessed DMA legacy addresses are forwarded to the internal DMA controllers.
1: DMA remapping enable. Individual legacy DMA channels can be remapped.

Register 82h~83h Reserved

Register 84h Legacy DMA Slave Channel Enable

Attribute: Write/Read
Default: 00h

Bit 7 **Legacy DMA Slave Channel 7 Enable**
0: Disable
1: Enable

Bit 6 **Legacy DMA Slave Channel 6 Enable**
0: Disable
1: Enable

Bit 5 **Legacy DMA Slave Channel 5 Enable**
0: Disable

1: Enable

Bit 4 **Reserved**

Bit 3 **Legacy DMA Slave Channel 3 Enable**
0: Disable
1: Enable

Bit 2 **Legacy DMA Slave Channel 2 Enable**
0: Disable
1: Enable

Bit 1 **Legacy DMA Slave Channel 1 Enable**
0: Disable
1: Enable

Bit 0 **Legacy DMA Slave Channel 0 Enable**
0: Disable
1: Enable

Register 85h~87h Reserved

Register 88h Serial Interrupt Control Register

Attribute : Write/Read
Default Value : 00h

Bit 7 **Serial Interrupt (SIRQ) Control**
0: Disable
1: Enable

Bit 6 **Quiet/Continuous Mode**
0: Quiet
1: Continuous

Bits 5:2 **SIRQ Sample Period**
0000: 17 slots
0001: 18 slots
0010: 19 slots
::::::::::::
1111: 32 slots

Bits 1:0 **Start Cycle length**
00: 4 PCI clocks
01: 6 PCI clocks
10: 8 PCI clocks
11: Reserved

Register 89h Serial Interrupt Enable Register 1

Attribute : Write/Read
Default Value : 00h

Bit 7 **Reserved**

Bit 6 **Serial IRQ7 Enable**
0: Disable
1: Enable

Bit 5 **Serial IRQ6 Enable**
0: Disable
1: Enable

Bit 4 **Serial IRQ5 Enable**
0: Disable
1: Enable

Bit 3 **Serial IRQ4 Enable**
0: Disable
1: Enable

Bit 2 **Serial IRQ3 Enable**
0: Disable
1: Enable

Bit 1 **Serial SMI# Enable**
0: Disable
1: Enable

Bit 0 **Reserved**

Register 8Ah Serial Interrupt Enable Register 2

Attribute : Write/Read
Default Value : 00h

Bit 7 **Serial IOCHCK# Enable**
0: Disable
1: Enable

Bit 6 **Serial IRQ15 Enable**
0: Disable
1: Enable

Bit 5 **Serial IRQ14 Enable**
0: Disable
1: Enable

Bit 4 **Reserved**

Bit 3 **Serial IRQ12 Enable**
0: Disable
1: Enable

Bit 2 **Serial IRQ11 Enable**
0: Disable
1: Enable

Bit 1 **Serial IRQ10 Enable**

0: Disable
1: Enable

Bit 0 **Serial IRQ9 Enable**
0: Disable
1: Enable

Register 90h~91h ACPI Base Address Register

6.2.1 Offset Register for ACPI/SCI Base Address Register

The following Registers are shown the offset register of ACPI , i.e., Register 00h means the I/O address <Base> + 00h and the Base address is programmed in the Register 90h~91h of PCI to ISA bridge Configuration Register.

Register 00h Power Management Status Register

Bit 15 **Wake up status (WAK_STS)**
This bit is set when the system in the suspend state and an enable resume event occurs. Upon setting this bit, the state machine will transition the system to the on state. This bit can only be set by hardware and only can be cleared by software writing a one to this bit position.

Bits 14:11 **Reserved**

Bit 10 **RTC status (RTC_STS)**
This bit is set when the RTC generates an alarm. While both RTC_EN bit and RTC_STS bit are set, a power management event is raised(SCI, SMI or resume event). This bit is only set by hardware and only be reset by software writing a one to this bit position.

Bit 9 **Reserved**

Bit 8 **Power button status (PWRBTN_STS)**
This bit is set when the power button is pushed (The PWRBT# signal is asserted Low). In the working state, while PWRBTN_STS bit and PWRBTN_EN bit are both set then a SCI is raised. In the sleeping state, while PWRBTN_STS bit and PWRBTN_EN bit are both set then a wake-up event is generated. This bit is only set by hardware and can only be reset by software writing a one to this bit position.

Bits 7:6 **Reserved**

Bit 5 **Global status (GBL_STS)**
This bit is set when an SCI is generated due to BIOS wanting the attention of the SCI handler. BIOS will have a control bit which raise an SCI. (Register 1C bit 10)

Bit 4 **Bus master status**
This is the bus master status bit. This bit is set anytime a system bus master requests the system bus, and can only be cleared by writing a one to this bit position.

Bits 3:1 **Reserved**

Bit 0 **Power management timer status (TMR_STS)**

Power management timer status or DOZE timer status. Only the Offset Register 1C bit 13 is set to 1 and SCI_EN bit is set to 0, the free running timer (24 bit timer) is to be DOZE timer. If the most significant bit of 24 bits timer is changed from “1” to “0” or “0” to “1”, then the TMR_STS bit will be set. While TMR_STS bit and TMR_EN bit are set, a power management is raised. It can only be cleared by writing a one to this bit position.

Register 02h Power Management Resume Enable Register

Bits 15:11 Reserved

Bit 10 **RTC Enable (RTC_EN)**

This bit is used to enable the setting of the RTC_STS bit to generate a power management event. (SCI, SMI or WAKE)

Bit 9 **Reserved**

Bit 8 **Power Button Enable (PWRBTN_EN)**

This bit is used to enable the setting of the PWRBTN_STS bit to generate a power management event (SCI, SMI or WAKE)

Bits 7:6 **Reserved**

Bit 5 **Global Enable (GBL_EN)**

The global enable bit. When both the GBL_EN bit and GBL_STS bit are set then an SCI is raised.

Bits 4:1 **Reserved**

Bit 0 **Power management timer Enable (TMR_EN)**

This is the 24 bits free running timer enable bit. If this bit and TMR_STS bit are set then a power management event is raised. (SMI or SCI)

Register 04h Power Management Control Register

Bits 15:14 Reserved

Bit 13 **Sleeping Enable (SLP_EN)**

This is a write-only bit and reads it always return a zero. Setting this bit causes the system to sequence into the suspend state defined by the SLP_TYP field.

Bits 12:10 Sleeping Type (SLP_TYP)

Defines the type of suspend type that the system should enter power down mode when the SLP_EN bit is set to one.

000: S1 state

100: S5 state

Bit 9 **SiS Proprietary Bit, must be 1**

Bits 8:3 **Reserved**

Bit 2 **Global Release GBL_RLS**

This bit is used by the ACPI software to raise an SMI to the BIOS software. BIOS software has a corresponding enable and status to control its ability to receive ACPI events. (Register 25 bit 0 and Register 26 bit 0)

Bit 1 Bus Master Reload Enable (BM_RLD)

If enabled, this bit allows a bus master request to cause any processor in the C3 state to transition to the C0 state.

0: Disable

1: Enable

Bit 0 SCI Enable (SCI_EN)

Selects the power management event to be either an SCI or SMI interrupt. When this bit is set , then the power management events will generate an SCI interrupt. When this bit is reset power management events will generate an SMI interrupt.

Register 08h ACPI Power Timer Register

Bits 31:24 Reserved

Bits 23:0 Power management timer value

This read-only field returns the running count of the power management timer. The timer-expire interval is translated by the follow equation :

Timer-Expire Interval=(Timer Counter-1) x 0.28us

Register 0Ch

Bits 31:7 Reserved

Bit 6 IRQ0 Enable

This bit enables the de-assert STPCLK# a short time when IRQ0 happens during C2 and C3 state.

Bit 5 CPU Clock Control

This bit controls the clock generator control function via pin GPO6 during S1 state.

Bit 4 Throttling Function Enable

This bit enables clock throttling function.

Bits 3:1 Throttling Duty cycle Control

This 3-bit field determines the duty cycle of the STPCLK# signal when the system in the throttling mode.

000 RESERVED

001 7 : 1 (High : Low)

010 6 : 2

011 5 : 3

100 4 : 4

101 3 : 5

110 2 : 6

111 1 : 7

Bit 0 **Reserved**

Register 10h

Bits 7:0 **Enter C2 Power state register**

Reads to this register return all zeros, writes to this register have no effect. Reads to this register also generate a " Enter a C2 power state ".

Register 11h

Bits 7:0 **Enter C3 Power state register**

Reads to this register return all zeros, writes to this register have no effect. Reads to this register also generate a " Enter a C3 power state ".

Register 12h

Bits 7:1 **Reserved**

Bit 0 **Arbiter disable**

In order to maintain the Cache coherence when CPU is in the C3 state, the other master should not get the grant. This bit is used to enable and disable the system arbiter. When this bit is “0” the system arbiter is enable and can grant the bus to other bus masters bus. When this bit is “1” the system arbiter is disable, and the default CPU has ownership of the system bus.

Register 13h

Bits 7:0 **General Purpose Timer**

It is a down counter. It has the time resolution 1 µsec or 1 min. While a value is written to this timer, it begin to count. It raises a power management event when the counter is time out. In addition, it can be a suspend timer when Register 1C bit 11 is set to 1 and SCI_EN is 0.

Register 14h

Bit 15 **Wakeup IRQ status(WAKEIRQ_STS)**

This bit is set when IRQ[1-15] or NMI is generated. WAK_STS is set when both WAKEIRQ_STS and WAKEIRQ_EN are set at SUSPEND mode.

Bit 14 **USB status(USB_STS)**

This bit is set when USB event is generated. While both USB_STS and USB_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 13 **General purpose timer status(GPTIMER_STS)**

This bit is set when General purpose timer is time out. While both GPTIMER_STS and GPTIMER_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 12 **GPIO10 status(GPIO10_STS)**

This bit is set when GPIO10 event is generated and GPIO10 is to be input function. While both GPIO10_STS and GPIO10_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 11 **GPIO9/Thermal status(GPIO9_STS)**

This bit is set when GPIO9 event is generated and GPIO9 is to be input function. While both GPIO9_STS and GPIO9_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 10 **GPIO8 status(GPIO8_STS)**

This bit is set when GPIO8 event is generated and GPIO8 is to be input function. While both GPIO8_STS and GPIO8_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 9 **GPIO7 status(GPIO7_STS)**

This bit is set when GPIO7 event is generated and GPIO7 is to be input function. While both GPIO7_STS and GPIO7_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 8 **SERIAL IRQ status(SIRQ_STS)**

This bit is set when serial IRQ event is generated. While both SIRQ_STS and SIRQ_EN are set to 1, a power management event is raised.(SMI) It can only be cleared by writting a one to this bit position.

Bit 7 **GPIO5 status(GPIO5_STS)**

This bit is set when GPIO5 event is generated and GPIO5 is to be input function. While both GPIO5_STS and GPIO5_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bits 6:5 **Reserved**

Bit 4 **GPIO2 status(GPIO2_STS)**

This bit is set when GPIO2 event is generated and GPIO2 is to be input function. While both GPIO2_STS and GPIO2_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 3 **GPIO1 status(GPIO1_STS)**

This bit is set when GPIO1 event is generated and GPIO1 is to be input function. While both GPIO1_STS and GPIO1_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 2 **GPIO0 status(GPIO0_STS)**

This bit is set when GPIO0 event is generated and GPIO0 is to be input function. While both GPIO0_STS and GPIO0_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 1 **External SMI Status (HOTKEY_STS)**

This bit is set when HOTKEY (via EXTSMI#) event is generated. While both HOTKEY_STS and HOTKEY_EN are set to 1, a power management event is raised.(SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Bit 0 **Ring Status(RI_STS)**

This bit is set when MODEM ring event is generated. While both RI_STS and RI_EN are set to 1, a power management event is raised. (SMI, SCI or WAKE) It can only be cleared by writting a one to this bit position.

Register 16h

Bit 15 **Wake up IRQ Enable(WAKEIRQ_EN)**

The WAKEIRQ enable bit. When WAKEIRQ_EN and WAKEIRQ_STS are set during SUSPEND, WAK_STS will be set.

Bit 14 **USB Enable(USB_EN)**

The USB enable bit. When USB_EN and USB_STS are set, a power management is raised.

Bit 13 **General purpose timer Enable(GPTIMER_EN)**

The General Purpose timer enable bit. When GPTIMER_STS and GPTIMER_EN are set, a power management is raised.

Bit 12 **Reserved**

Bit 11 **GPIO9 Enable(GPIO9_EN)**

The GPIO9 enable bit. When GPIO9_STS and GPIO9_EN are set, a power management event is raised.

Bit 10 **GPIO8 Enable(GPIO8_EN)**

The GPIO8 enable bit. When GPIO8_STS and GPIO8_EN are set, a power management event is raised.

Bit 9 **GPIO7 Enable(GPIO7_EN)**

The GPIO7 enable bit. When GPIO7_STS and GPIO7_EN are set, a power management event is raised.

Bit 8 **Serial IRQ Enable(SIRQ_EN)**

The serial IRQ enable bit. When SIRQ_STS and SIRQ_EN are set, a power management event is raised.

Bits 7:5 **Reserved**

Bit 4 **GPIO2 Enable(GPIO2_EN)**

The GPIO2 enable bit. When GPIO2_STS and GPIO2_EN are set, a power management event is raised.

Bit 3 **GPIO1 Enable(GPIO1_EN)**

The GPIO1 enable bit. When GPIO1_STS and GPIO1_EN are set, a power management event is raised.

Bit 2 **GPIO0 Enable(GPIO0_EN)**

The GPIO0 enable bit. When GPIO0_STS and GPIO0_EN are set, a power management event is raised.

Bit 1 **Hotkey (via EXTSMI#) Enable (HOTKEY_EN)**

The HOTKEY enable bit. When HOTKEY_STS and HOTKEY_EN are set, a power management event is raised.

Bit 0 **Ring Enable(RI_EN)**

The MODEM ring enable bit. When RI_EN and RI_STS are set, a power management event is raised.

Register 18h

Bits 15:11 **Reserved**

Bit 10 **GPIO10 pin status register**

When GPIO10 is to be input function, it can read the input status via this register. When GPIO10 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 9 **GPIO9 pin status register**

When GPIO9 is to be input function, it can read the input status via this register. When GPIO9 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 8 **GPIO8 pin status register**

When GPIO8 is to be input function, it can read the input status via this register. When GPIO8 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 7 **GPIO7 pin status register**

When GPIO7 is to be input function, it can read the input status via this register. When GPIO7 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 6 **GPO6 pin status register**

It can write any value to system via this register to control the external peripheral device.

Bit 5 **GPIO5 pin status register**

When GPIO5 is to be input function, it can read the input status via this register. When GPIO5 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 4 **GPO4 pin status register**

It can write any value to system via this register to control the external peripheral device.

Bit 3 **GPO3 pin status register**

It can write any value to system via this register to control the external peripheral device.

Bit 2 **GPIO2 pin status register**

When GPIO2 is to be input function, it can read the input status via this register. When GPIO2 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 1 **GPIO1 pin status register**

When GPIO1 is to be input function, it can read the input status via this register. When GPIO1 is to be output function, it can write any value to system via this register to control the external peripheral device.

Bit 0 **GPIO0 pin status register**

When GPIO0 is to be input function, it can read the input status via this register. When GPIO0 is to be output function, it can write any value to system via this register to control the external peripheral device.

Register 1Ah

Bits 15:10 **Reserved**

Bit 9 **GPIO9 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Bit 8 **GPIO8 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Bit 7 **GPIO7 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Bits 6:3 **Reserved**

Bit 2 **GPIO2 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Bit 1 **GPIO1 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Bit 0 **GPIO0 INPUT/OUTPUT Control**

0 : Input Mode

1 : Output Mode

Register 1Ch

Bits15:14 **Reserved**

Bit 13 **Power Management timer functional selection**

0 : ACPI PM timer

1 : DOZE timer

Bit 12 **General Purpose timer of time slot**
0 : 1 us
1 : 1 min

Bit 11 **General Purpose timer functional Selection**
0 : BIOS timer
1 : Suspend timer

Bit 10 **BIOS relationship (BIOS_RLS)**
This bit is set by BIOS then the Global status bit (Register 00 bit 5) will be set.

Bit 9 **Pin Definition Select for THRM#/GPIO9**
0 : THRM#(Thermal detect)
1 : GPIO9

Bit 8 **Ring In detection method**
0 : Lasting low 150ms
1 : Between 14Hz and 70 Hz

Bit 7 **Reserved**

Bit 6 **Pin Definition Select for GPO6/MA14**
0: GPO6
1: MA14

Bit 5 **Pin Definition Select for GPO4/MA13**
0: MA13
1: GPO4

Bit 4 **Pin Definition Select for GPO3/MA12**
0: MA12
1: GPO3

Bit 3 **Throttling function for thermal**
0 : Disable
1 : Enable
If thermal is too high and asserted, throttling function will work. In this situation, it don't care the throttling enable bit.

Bit 2 **GPIO2 Pin Control**
0 : Not used (means NC pin)
1 : Used

Bit 1 **GPIO1 Pin Control**
0 : Not used (means NC pin)
1 : Used

Bit 0 **GPIO0 Pin Control**
0 : Not used (means NC pin)
1 : Used

Register 1Eh

Bits 15:12 **Reserved**

Bit 11 **Hot key polarity (via EXTSMI#)**
0 : Low activity
1 : High activity

Bit 10 **GPIO10 polarity in Input Mode**
0 : Low activity
1 : High activity

Bit 9 **GPIO9/Thermal polarity in Input Mode**
0 : Low activity
1 : High activity

Bit 8 **GPIO8 polarity in Input Mode**
0 : Low activity
1 : High activity

Bit 7 **GPIO7 polarity in Input Mode**
0 : Low activity
1 : High activity

Bit 6 **Reserved**

Bit 5 **GPIO5 polarity in Input Mode**
0 : Low activity
1 : High activity

Bits 4:3 **Reserved**

Bit 2 **GPIO2 polarity in Input Mode**
0 : Low activity
1 : High activity

Bit 1 **GPIO1 polarity in Input Mode**
0 : Low activate
1 : High activate

Bit 0 **GPIO0 polarity in Input Mode**
0 : Low activate
1 : High activate

Register 20h

Bit 7:0 **SMI Command Port**

Register 24h

Bit 7 **Reserved**

Bit 6 **Pin Definition Select for GPIO9/THRM#/IOCHK#**
0 : IOCHK#
1 : GPIO9/THRM#

Bits 5:2 **Reserved**

Bit 1 **Internal test bit**
0 : Normal mode
1 : Test mode

Bit 0 **Internal test bit**
0 : Normal mode
1 : Test mode

Register 25h

Bits 7:5 **Reserved**

Bit 4 **SMI command disable Status (SMICMDDIS_STS)**
This bit is set when OS write ACPI disable value to SMI command port. While SMICMDDIS_STS and SMICMD_DIS are set to 1, a SMI is raised.

Bit 3 **SMI command enable Status (SMICMDEN_STS)**
This bit is set when OS write ACPI enable value to SMI command port. While SMICMDEN_STS and SMICMD_EN are set to 1, a SMI is raised.

Bit 2 **Period SMI Status (PERSMI_STS)**
When period SMI is enable in legacy PMU, every 16 sec this bit will be set.

Bit 1 **LEGA_STS (only can be used for SMI generation)**
This bit is set when system wake up from suspend in legacy PMU. When both LEGA_STS and LEGA_EN are set, a SMI is raised. It can only be cleared by writting a one to this bit position.

Bit 0 **BIOS_STS(only can be used for SMI generation)**
This bit is set when a SMI is generated due to the ACPI wanting the attention of SMI handler. When both BIOS_STS and BIOS_EN are set, a SMI is raised. It can only be cleared by writting a one to this bit position.

Register 26h

Bits 7:6 **Reserved**

Bit 5 **Reload DOZE or SUSPEND timer bit**
When this bit is enable, monitor events of Register 90h and 91h of Host to PCI bridge configuration space will reload DOZE or SUSPEND timer.

Bit 4 **SMI Command Disable (SMICMD_DIS)**
SMI command disable bit. While SMICMDDIS_STS and SMICMD_DIS are set to 1, a SMI is raised.

Bit 3 **SMI Command Enable (SMICMD_EN)**
SMI command enable bit. While SMICMDEN_STS and SMICMD_EN are set to 1, a SMI is raised.

Bit 2 **PER_SMI (only can be used for SMI generation)**
If this bit is set to 1, every 16 sec sends a SMI.

Bit 1 **LEGA_EN (only can be used for SMI generation)**
Legacy PMU enable bit.

Bit 0 **BIOS_EN**

BIOS enable bit. This bit corresponds to BIOS_STS bit (Register 25, bit 0) in order to raise the SMI.

Register 28h

Bits 7:0 **Programming SMI command port enable value**

Register 29h

Bits 7:0 **Programming SMI command port disable value**

Register 2Ah Mail Box

Bits 7:0 **Free storage**

R/W register for BIOS or ACPI to use.

Register 2Bh

Bits 7:1 **Reserved**

Bit 0 **ACPI test mode (for internal use only)**

0 : Normal Mode

1: Test Mode

6.3 Non-Configuration Registers

DMA Registers

These registers can be accessed from PCI bus.

Address	Attribute	Register Name
0000h	R/W	DMA1 CH0 Base and Current Address Register
0001h	R/W	DMA1 CH0 Base and Current Count Register
0002h	R/W	DMA1 CH1 Base and Current Address Register
0003h	R/W	DMA1 CH1 Base and Current Count Register
0004h	R/W	DMA1 CH2 Base and Current Address Register
0005h	R/W	DMA1 CH2 Base and Current Count Register
0006h	R/W	DMA1 CH3 Base and Current Address Register
0007h	R/W	DMA1 CH3 Base and Current Count Register
0008h	R/W	DMA1 Status(r) Command(w) Register
0009h	WO	DMA1 Request Register
000Ah	WO	DMA1 Write Single Mask Bit
000Bh	WO	DMA1 Mode Register
000Ch	WO	DMA1 Clear Byte Pointer
000Dh	WO	DMA1 Master Clear
000Eh	WO	DMA1 Clear Mask Register

000Fh	R/W	DMA1 Write All Mask Bits(w) Mask Status Register(r)
00C0h	R/W	DMA2 CH0 Base and Current Address Register
00C2h	R/W	DMA2 CH0 Base and Current Count Register
00C4h	R/W	DMA2 CH1 Base and Current Address Register
00C6h	R/W	DMA2 CH1 Base and Current Count Register
00C8h	R/W	DMA2 CH2 Base and Current Address Register
00CAh	R/W	DMA2 CH2 Base and Current Count Register
00CCh	R/W	DMA2 CH3 Base and Current Address Register
00CEh	R/W	DMA2 CH3 Base and Current Count Register
00D0h	R/W	DMA2 Status(r) Command(w) Register
00D2h	WO	DMA2 Request Register
00D4h	WO	DMA2 Write Single Mask Bit Register
00D6h	WO	DMA2 Mode Register
00D8h	WO	DMA2 Clear Byte Pointer
00DAh	WO	DMA2 Master Clear
00DCh	WO	DMA2 Clear Mask Register
00DEh	R/W	DMA2 Write All Mask Bits(w) Mask Status Register(r)

These registers can be accessed from PCI bus or ISA bus.

Address	**Attribute**	**Register Name**
0080h	R/W	Reserved
0081h	R/W	DMA Channel 2 Low Page Register
0082h	R/W	DMA Channel 3 Low Page Register
0083h	R/W	DMA Channel 1 Low Page Register
0084h	R/W	Reserved
0085h	R/W	Reserved
0086h	R/W	Reserved
0087h	R/W	DMA Channel 0 Low Page Register
0088h	R/W	Reserved
0089h	R/W	DMA Channel 6 Low Page Register
008Ah	R/W	DMA Channel 7 Low Page Register
008Bh	R/W	DMA Channel 5 Low Page Register
008Ch	R/W	Reserved
008Dh	R/W	Reserved
008Eh	R/W	Reserved
008Fh	R/W	Refresh Low Page Register

Interrupt Controller Registers (These registers can be accessed from PCI bus or ISA bus.)

Address	**Attribute**	**Register Name**
0020h	R/W	INT 1 Base Address Register
0021h	R/W	INT 1 Mask Register
00A0h	R/W	INT 2 Base Address Register

00A1h	R/W	INT 2 Mask Register

Timer Registers (These registers can be accessed from PCI bus or ISA bus.)

Address	Attribute	Register Name
0040h	R/W	Interval Timer 1 - Counter 0
0041h	R/W	Interval Timer 1 - Counter 1
0042h	R/W	Interval Timer 1 - Counter 2
0043h	WO	Interval Timer 1 - Control Word Register

RTC Registers

Address	Attribute	Register Name
00h	R/W	Seconds
01h	R/W	Seconds Alarm
02h	R/W	Minutes
03h	R/W	Minutes Alarm
04h	R/W	Hours
05h	R/W	Hours Alarm
06h	R/W	Day of Week
07h	R/W	Day of Month
08h	R/W	Month
09h	R/W	Year
0Ah	R/W	Register A
0Bh	R/W	Register B (**bit 3 must be set to 0)**
0Ch	R/W	Register C
0Dh	R/W	Register D
7Eh	R/W	Day of Month Alarm
7Fh	R/W	Month Alarm

Note: Day of Month Alarm and Month Alarm on 7Eh/7Fh have function only when APC_EN is enabled (APC I:[6]=1)

APC Control Registers (Must set Register 44h bit 4 to 1 to access these registers)

Address	Attribute	Register Name
00h	R/W	Reserved
01h	R/W	Reserved
02h	R/W	Day of Week Alarm
03h	R/W	Auto Power Control Register I
04h	R/W	Auto Power Control Register II

Day of Week Alarm Register

Bit 7 Automatic Power Up System On Sat

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 6 **Automatic Power Up System On Fri**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 5 **Automatic Power Up System On Thu**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 4 **Automatic Power Up System On Wed**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 3 **Automatic Power Up System On Tue**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 2 **Automatic Power Up System On Mon**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 1 **Automatic Power Up System On Sun**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 and Day of Week Alarm Register bit 0 should be enabled and RTC Alarm should be programmed.

Bit 0 **Day of Week Alarm Control (DayWeekAlarm_EN)**

0: Disable

1: Enable

Before enabling this function, Auto Power Control Register I bit 6 should be enabled.

Auto Power Control Register I

Bit 7 **Reserved**

Bit 6 **Auto Power Control (APC) Function Control (APC_EN)**
0: Disable
1: Enable
When enabling this bit, functions of automatic power up system, power off system and a ring leads to power up system may work.

Bit 5 **RING Function Control (RNUP_EN)**
0 : Disable
1 : Enable
Before enabling this function, Auto Power Control Register I bit 6 should be enabled.

Bit 4 **RING Input Active Level Control**
0: Active high
1: Active low

Bit 3 **GPIO5 Leads To Power Up System Control (STARTREQ_EN)**
0 : Disable (default)
1 : Enable
A high to low transition on GPIO5 leads to activate the power up control, this bit is effective only when bit 6 of Auto Power Control Register I is set.

Bit 2 **ACPI Alarm Function Status Control**
0 : Disable
1 : Enable
This bit is used to enable the five alarm functions in RTC registers 01h, 03h, 05h, 7Eh, 7Fh.
Note: Before enabling this function, APC I[6] should be enabled.

Bit 1 **Test Mode for internal use only**
0 : Normal Mode
1 : Test Mode

Bit 0 **Test Mode for internal use only**
0 : Normal Mode
1 : Test Mode

Auto Power Control Register II

Bits 7:4 **Reserved**

Bit 3 **GPIO10 as Input/Output Mode Control**
0: Output Mode
1: Input Mode

Bit 2 **GPIO5 as Input/Output Mode Control**
0: Output Mode
1: Input Mode

Bit 1 **Pin Definition Select for GPIO10/ACPILED Selection**
0: GPIO10
1: ACPILED

Bit 0 **1 Hz function Support**
0: Disable
1: Enable

Other Registers (These registers can be accessed from PCI bus or ISA bus.)

Address	Attribute	Register Name
0061h	R/W	NMI Status Register
0070h	WO	CMOS RAM Address and NMI Mask Register
00F0h	WO	Coprocessor Error Register

Register 4D0h IRQ Edge/Level Control Register 1

Bit 7 **IRQ7**
0: Edge sensitive
1: Level sensitive

Bit 6 **IRQ6**
0: Edge sensitive
1: Level sensitive

Bit 5 **IRQ5**
0: Edge sensitive
1: Level sensitive

Bit 4 **IRQ4**
0: Edge sensitive
1: Level sensitive

Bit 3 **IRQ3**
0: Edge sensitive
1: Level sensitive

Bit 2 **IRQ2**
This bit must be set to 0. Read as 0.

Bit 1 **IRQ1**
This bit must be set to 0. Read as 0.

Bit 0 **IRQ0**
This bit must be set to 0. Read as 0.
After reset this register is set to 00h.

Register 4D1h IRQ Edge/Level Control Register 2

Bit 7 **IRQ15**
0: Edge sensitive
1: Level sensitive

Bit 6 **IRQ14**
0: Edge sensitive
1: Level sensitive

Bit 5 **IRQ13**
This bit must be set to 0. Read as 0.

Bit 4 **IRQ12**
0: Edge sensitive
1: Level sensitive

Bit 3 **IRQ11**
0: Edge sensitive
1: Level sensitive

Bit 2 **IRQ10**
0: Edge sensitive
1: Level sensitive

Bit 1 **IRQ9**
0: Edge sensitive
1: Level sensitive

Bit 0 **IRQ8#**
This bit must be set to 0. Read as zero.
After reset this register is set to 00h.

Register CF9h Reset control register

Bits 7:5 **Reserved**

Bit 4 **INIT Control**
0: Drive INIT during keyboard reset
1: Drive CPURST# during keyboard reset and INIT is inactive.

Bits 3:2 **Software Reset Control**
Writing “11” to these two bits at the same time will generate CPURST.

Bits 1:0 **Reserved**

6.4 PCI IDE Configuration Space

Device	IDSEL	Function Number
IDE	AD12	0001b

Register 00~01h - Vendor ID

Bits 15:0 **1039h(Read Only)**

Register 02~03h - Device ID

Bits 15:0 **5513h(Read only)**

Register 04h~05h Command port

Bits 15:8 **00h(Read Only)**

Bits 7:3 **These bits are hardwired to 0.**

Bit 2 **Bus Master Enable**

When set, the Bus master function is enabled. It is disabled by default.

Bit 1 **Memory Space Enable**

This bit should be programmed as “0”.

Bit 0 **I/O Space Enable**

When enabled, the built-in IDE will respond to any access of the IDE legacy ports in the compatibility mode, or to any access of the IDE relocatable ports in the native mode. Also, any access to the PCI bus master IDE registers are allowed. This bit is zero(disabled) on reset.

Register 06h~07h Status

Bits 15:14 **These bits are hardwired to zero.**

Bit 13 **Master Abort Asserted**

This bit is set when a PCI bus master IDE transaction is terminated by master abort. While this bit is set, IDE will issue an interrupt request. This bit can be cleared by writing a 1 to it.

Bit 12 **Received Target Abort**

The bit is set whenever PCI bus master IDE transaction is terminated with target abort.

Bit 11 **Signaled Target Abort.**

The bit will be asserted when IDE terminates a transaction with target abort.

Bits 10:9 **DEVSEL# Timing**

These two bits define the timing of asserting DEVSEL#. The built-in IDE always asserts DEVSEL# in fast timing, and thus the two bits are hardwired to 0 per PCI Spec.

Bit 8 **Reserved, Read as "0".**

Bits 7:6 **These bits are hardwired to zero.**

Bit 5 **This is a reserved bit, and is recommend to program 0.**

Bits 4:0 **These bits are hardwired to zero.**

Register 08h - Revision Identification

Bits 7:0 **D0h(Read Only)**

Register 09h - Programming Interface Byte

Bit 7 **Master IDE Device**

This bit is hardwired to one to indicate that the built-in IDE is capable of supporting bus master function.

Bits 6:4 **Reserved**

Bit 3 **Secondary IDE Programmable Indicator**

When the bit is programmed as '1', it means that the primary channel can be programmed to operate in compatible or native mode. When the bit is programmed as ‘0’, the mode is fixed and is determined by the value of bit 2. This bit should be programmed as ‘1’ during the BIOS boot up procedures.

Bit 2 **Secondary IDE Operating Mode**

This bit defines the mode that the secondary IDE channel is operating in. Zero corresponds to 'compatibility' while one means native mode. By default, this bit is 0 and is programmable.

Bit 1 **Primary IDE Programmable Indicator**

When the bit is programmed as '1', it means that the primary channel can be programmed to operate in compatible or native mode. When the bit is programmed as ‘0’, the mode is fixed and is determined by the value of bit 0. This bit should be programmed as ‘1’ during the BIOS boot up procedures.

Bit 0 **Primary IDE Operating Mode**

This bit defines the mode that the primary IDE channel is operating in. Zero corresponds to 'compatibility' while one means native mode. By default, this bit is 0 and is programmable.

Register 0Ah - Subclass ID

Bits 7:0 **01h**

Register 0Bh - Class ID

Bits 7:0 **01h**

Register 0Ch - Cache Line Size

Bits 7:0 **00h**

Register 0Dh- Latency Timer

Bits 7:0 **Programmable (from 0 to 255). The default value is 0.**

Register 0Eh - Header Type

Bits 7:0 **80h**

Register 0Fh - BIST

Bits 7:0 **00h**

Register 10h~13h Primary Channel Command Block Base Address Register

Register 14h~17h Primary Channel Control Block Base Address Register

Register 18h~1Bh Secondary Channel Command Block Base Address Register

Register 1Ch~1Fh Secondary Channel Control Block Base Address Register

In the native mode, above four registers define the IDE base address for each of the two IDE devices in both the primary and secondary channels respectively. In the compatible mode, the four registers can still be programmed and read out, but it does not affect the IDE address decoding.

Register 20h~23h Bus Master IDE Control Register Base Address

Offset Register	Register Access
00h	Bus Master IDE Command Register (Primary)
01h	Reserved
02h	Bus Master IDE Status Register(Primary)
03h	Reserved
04-07h	Bus Master IDE PRD (*) Table Pointer (Primary)
08h	Bus Master IDE Command Register (Secondary)
09h	Reserved
0Ah	Bus Master IDE Status Register (Secondary)
0Bh	Reserved
0C-0Fh	Bus Master IDE PRD (*) Table Pointer (Secondary)

*PRD: Physical Region Descriptor

Register 24h~2Bh Reserved

Register 2Ch Subsystem ID

This register can be written once and is used to identify vendor of the subsystem.

Register 2Dh~2Fh Reserved. Read as"0".

Register 30h~33h Expansion ROM Base Address

Register 40h IDE Primary Channel/Master Drive Data Recovery Time Control.

Bit 7 **Test mode for internal use only**

0: Normal mode

1: Test mode

Bit 6 **Test mode for internal use only**

0: Normal mode

1: Test mode

Bits 5:4 **Reserved**

Bits 3:0 **Recovery Time**

0000: 12 PCICLK	0001: 1 PCICLK
0010: 2 PCICLK	0011: 3 PCICLK
0100: 4 PCICLK	0101: 5 PCICLK
0110: 6 PCICLK	0111: 7 PCICLK
1000: 8 PCICLK	1001: 9 PCICLK
1010: 10 PCICLK	1011: 11 PCICLK
1100: 13 PCICLK	1101: 14 PCICLK
1110: 15 PCICLK	1111: 15 PCICLK

Register 41h IDE Primary Channel/Master Drive Control

Bit 7 **Ultra DMA Mode Control**

0: Disable

1: Enable

Bits 6:5 **Ultra DMA/33 cycle time Select**

00: Reserved

01: Cycle time of 2 PCI clocks for data out

10: Cycle time of 3 PCI clocks for data out

11: Cycle time of 4 PCI clocks for data out

Bits 4:3 **Reserved**

Bits 2:0 **Data Active Time Control**

000: 8 PCICLK	001: 1 PCICLK
010: 2 PCICLK	011: 3 PCICLK
100: 4 PCICLK	101: 5 PCICLK
110: 6 PCICLK	111: 12 PCICLK

Register 42h IDE Primary Channel/Slave Drive Data Recovery Time Control.

Bits 7:4 **Reserved**

Bits 3:0 **Recovery Time**

0000:	12 PCICLK	0001:	1 PCICLK
0010:	2 PCICLK	0011:	3 PCICLK
0100:	4 PCICLK	0101:	5 PCICLK
0110:	6 PCICLK	0111:	7 PCICLK
1000:	8 PCICLK	1001:	9 PCICLK
1010:	10 PCICLK	1011:	11 PCICLK

1100:	13 PCICLK	1101:	14 PCICLK
1110:	15 PCICLK	1111:	15 PCICLK

Register 43h IDE Primary Channel/Slave Drive Data Active Time Control

Bit 7 **Ultra DMA/33 Mode Control**
0: Disable
1: Enable

Bits 6:5 **Ultra DMA/33 Cycle time Select**
00: Reserved
01: Cycle time of 2 PCI clocks for data out
10: Cycle time of 3 PCI clocks for data out
11: Cycle time of 4 PCI clocks for data out

Bits 4:3 **Reserved**

Bits 2:0 **Data Active Time Control**

000: 8 PCICLK	001: 1 PCICLK
010: 2 PCICLK	011: 3 PCICLK
100: 4 PCICLK	101: 5 PCICLK
110: 6 PCICLK	111: 12 PCICLK

Register 44h IDE Secondary Channel/Master Drive Data Recovery Time Control.

Bits 7:4 **Reserved**

Bits 3:0 **Recovery Time**

0000:	12 PCICLK	0001:	1 PCICLK
0010:	2 PCICLK	0011:	3 PCICLK
0100:	4 PCICLK	0101:	5 PCICLK
0110:	6 PCICLK	0111:	7 PCICLK
1000:	8 PCICLK	1001:	9 PCICLK
1010:	10 PCICLK	1011:	11 PCICLK
1100:	13 PCICLK	1101:	14 PCICLK
1110:	15 PCICLK	1111:	15 PCICLK

Register 45h IDE Secondary Channel/Master Drive Data Active Time Control

Bit 7 **Ultra DMA/33 Mode Control**
0: Disable
1: Enable

Bits 6:5 **Ultra DMA/33 Cycle time Select**
00: Reserved
01: Cycle time of 2 PCI clocks for data out
10: Cycle time of 3 PCI clocks for data out
11: Cycle time of 4 PCI clocks for data out

Bits 4:3 **Reserved**

Bits 2:0 **Data Active Time Control**

000: 8 PCICLK	001: 1 PCICLK
010: 2 PCICLK	011: 3 PCICLK
100: 4 PCICLK	101: 5 PCICLK
110: 6 PCICLK	111: 12 PCICLK

Register 46h IDE Secondary Channel/Slave Drive Data Recovery Time Control.

Bits 7:4 **Reserved**

Bits 3:0 **Recovery Time**

0000:	12 PCICLK	0001:	1 PCICLK
0010:	2 PCICLK	0011:	3 PCICLK
0100:	4 PCICLK	0101:	5 PCICLK
0110:	6 PCICLK	0111:	7 PCICLK
1000:	8 PCICLK	1001:	9 PCICLK
1010:	10 PCICLK	1011:	11 PCICLK
1100:	13 PCICLK	1101:	14 PCICLK
1110:	15 PCICLK	1111:	15 PCICLK

Register 47h IDE Secondary Channel/Slave Drive Data Active Time Control

Bit 7 **Ultra DMA/33 Mode Control**

0: Disable

1: Enable

Bits 6:5 **Ultra DMA/33 Mode Select**

00: Reserved

01: Cycle time of 2 PCI clocks for data out

10: Cycle time of 3 PCI clocks for data out

11: Cycle time of 4 PCI clocks for data out

Bits 4:3 **Reserved**

Bits 2:0 **Data Active Time Control**

000: 8 PCICLK	001: 1 PCICLK
010: 2 PCICLK	011: 3 PCICLK
100: 4 PCICLK	101: 5 PCICLK
110: 6 PCICLK	111: 12 PCICLK

Register 48h IDE Command Recovery Time Control

Bits 7:4 **Reserved**

Bits 3:0 **Recovery Time**

0000:	12 PCICLK	0001:	1 PCICLK
0010:	2 PCICLK	0011:	3 PCICLK
0100:	4 PCICLK	0101:	5 PCICLK
0110:	6 PCICLK	0111:	7 PCICLK
1000:	8 PCICLK	1001:	9 PCICLK

1010:	10 PCICLK	1011:	11 PCICLK
1100:	13 PCICLK	1101:	14 PCICLK
1110:	15 PCICLK	1111:	15 PCICLK

Note: This bit is only meanful when bit3 of register 52h is “0”.

Register 49h IDE Command Active Time Control

Bits 7:3 **Reserved**

Bits 2:0 **Data Active Time Control**

000: 8 PCICLK 001: 1 PCICLK
010: 2 PCICLK 011: 3 PCICLK
100: 4 PCICLK 101: 5 PCICLK
110: 6 PCICLK 111: 12 PCICLK

Note: This bit is only meanful when bit3 of register 52h is “0”.

Register 4Ah IDE General Control Register 0

Bit 7 **Bus Master generates PCI burst cycles Control**
0: Disable
1: Enable (default)

Bit 6 **Test Mode for internal use only**
0: Test Mode
1: Normal Mode

Bit 5 **Fast post-write control**
0: Disabled
1: Enabled (Recommended)

Bit 4 **Test Mode for internal use only**
0: Normal Mode
1: Test Mode

Bit 3 **Bus Master requests PCI bus ownership timing control**
0: PCI Request asserted when FIFO is 75% full during prefetch cycles.
1: PCI Request asserted when FIFO is 50% full during prefetch cycles.
The default value is '0'.

Bit 2 **IDE Channel 0 Enable Bit**
0: Disabled (default)
1: Enabled

Bit 1 **IDE Channel 1 Enable Bit**
0: Disabled (default)
1: Enabled

Bit 0 **Test Mode for Internal Use**
0: Normal Mode (default)
1: Test mode

Register 4Bh IDE General Control register 1

Bit 7 **Enable Postwrite of the Slave Drive in Channel 1.**
0: Disabled. (default)
1: Enabled.

Bit 6 **Enable Postwrite of the Master Drive in Channel 1.**
0: Disabled. (default)
1: Enabled.

Bit 5 **Enable Postwrite of the Slave Drive in Channel 0.**
0: Disabled. (default)
1: Enabled.

Bit 4 **Enable Postwrite of the Master Drive in Channel 0.**
0: Disabled. (default)
1: Enabled.

Bit 3 **Enable Prefetch of the Slave Drive in Channel 1.**
0: Disabled. (default)
1: Enabled.

Bit 2 **Enable Prefetch of the Master Drive in Channel 1.**
0: Disabled. (default)
1: Enabled.

Bit 1 **Enable Prefetch of the Slave Drive in Channel 0.**
0: Disabled. (default)
1: Enabled.

Bit 0 **Enable Prefetch of the Master Drive in Channel 0.**
0: Disabled. (default)
1: Enabled.

(Following two 16-bit wide registers define the prefetching length of each IDE channel respectively.)

Register 4Ch~4Dh Prefetch Count of Primary Channel

Bits 15:0 The Count (in bytes) of IDE prefetch. The maximum value can be programmed is 512. (Default value is 512)

Register 4Eh~4Fh Prefetch Count of Secondary Channel

Bits 15:0 The Count (in bytes) of IDE prefetch. The maximum value can be programmed is 512. (Default value is 512)

Register 50h~51h IDE minimum accessed time register

Bits 15:0 **16 bits accessed time control**
This 16-bit value (in unit of PCLK) defines a minimum accessed time for IDE controller. When IDE controller and ISA master are competing for the ISA/IDE

bus, and Register 52h bit 2 is programmed as “1” to enable the IDE Granting Timer, the ISA master can preempt IDE only when IDE controller has used the bus for a minimum accessed time as define in this register. A granting timer associated with each IDE channel is used to count IDE controller’s term on the bus. This register is default to 0000h and it means 0 PCI clock.

Register 52h IDE Miscellaneous Control Register

Bits 7:4 Reserved

Bit 3 IDE Command Timing Select

0: The recovery and active time programmed in register 48h-49h will be applied to command cycles for all IDE devices

1: The recovery and active time programmed in register 40h-47h will be applied to command cycles for their associated IDE devices. (Default)

Bit 2 IDE Granting Timer Control

0: Disable

1: Enable

This bit is used together with IDE minimum access time (Register 50h~51h). When enabled, the minimum accessed time of IDE can be guaranteed by the programmed value in register 50h~51h. When disabled, the ISA master always has higher priority than IDE, and hence can preempt IDE any time.

Bit 1 Test Mode for internal use only

0 : Normal Mode

1 : Test Mode

Bit 0 IDE FIFO Size Select

0: 32 Bytes FIFO

1: 64 Bytes FIFO(Recommended)

6.4.1 Offset Registers for PCI Bus Master IDE Control Registers

The PCI Bus master IDE Registers use 16 bytes of I/O Space. These registers can be accessed through I/O R/W to the address defined in the Bus Master IDE control register Base Address in the PCI IDE Configuration space. The base address is also defined in Register 20h~23h of PCI IDE configuration space.

Register 00h Bus Master PrimaryIDE Command Register

Bits 7:4 Reserved. Return 0 on reads.

Bit 3 Read or Write Control.

This bit defines the R/W control of the bus master transfer. When set to zero, PCI bus master reads are conducted. When set to one, PCI bus master writes are conducted.

Bits 2:1 Reserved.

Bit 0 Start/Stop Bus Master

The SiS Chip built-in IDE Controller enables its bus master operation whenever it detects this bit changing from a zero to a one. The operation can be halted by writing a zero to this bit.

Register 01h Reserved

Register 02h Bus Master Primary IDE Status Register

Bit 7 Simplex Only

This bit is hardwired to zero to indicate that both bus master channels can be operated at a time.

Bit 6 Drive 1 DMA Capable

This R/W bit can be set by BIOS or driver to indicate that drive 1 for this channel is capable of DMA transfers.

Bit 5 Drive 0 DMA Capable

This R/W bit can be set by BIOS or driver to indicate that drive 0 for this channel is capable of DMA transfers.

Bits 4:3 Reserved. Return 0 on reads

Bit 2 Interrupt

The bit is set by the rising edge of the IDE interrupt line to indicate that all data transferred from the drive is visible in the system memory. Writing a '1' to this bit can reset it.

Bit 1 Error

This bit is set when the IDE controller encounters an error during data transferring to/from memory.

Bit 0 Bus Master IDE Device Active

This bit is set when the start bit in the command register is set. It can be cleared when the last transfer of a region is performed, or the start bit is reset.

Register 03h Reserved

Register 04h~07h Bus Master Primary IDE PRD Table Pointer Register

This 32-bit register contains address pointing to the starting address of the PRD table.

Bits 31:2 Base Address of the PRD Table

Bits 1:0 Reserved

Register 08h Bus Master Secondary IDE Command Register

Bits 7:4 Reserved. Return 0 on reads.

Bit 3 Read or Write Control.

This bit defines the R/W control of the bus master transfer. When set to zero, PCI bus master reads are conducted. When set to one, PCI bus master writes are conducted.

Bits 2:1 **Reserved.**

Bit 0 **Start/Stop Bus Master**

The SiS Chip built-in IDE Controller enables its bus master operation whenever it detects this bit changing from a zero to a one. The operation can be halted by writing a zero to this bit.

Register 09h **Reserved**

Register 0Ah **Bus Master Secondary IDE Status Register**

Bit 7 **Simplex Only**

This bit is hardwired to zero to indicate that both bus master channels can be operated at a time.

Bit 6 **Drive 1 DMA Capable**

This R/W bit can be set by BIOS or driver to indicate that drive 1 for this channel is capable of DMA transfers.

Bit 5 **Drive 0 DMA Capable**

This R/W bit can be set by BIOS or driver to indicate that drive 0 for this channel is capable of DMA transfers.

Bits 4:3 **Reserved. Return 0 on reads**

Bit 2 **Interrupt**

The bit is set by the rising edge of the IDE interrupt line to indicate that all data transferred from the drive is visible in the system memory. Writing a '1' to this bit can reset it.

Bit 1 **Error**

This bit is set when the IDE controller encounters an error during data transferring to/from memory.

Bit 0 **Bus Master IDE Device Active**

This bit is set when the start bit in the command register is set. It can be cleared when the last transfer of a region is performed, or the start bit is reset.

Register 0Bh **Reserved**

Register 0Ch~0Fh **Bus Master Secondary IDE PRD Table Pointer Register**

This 32-bit register contains address pointing to the starting address of the PRD table.

Bits 31:2 **Base Address of the PRD Table**

Bits 1:0 Reserved

6.5 USB Configuration Registers

The USB Configuration Registers are located in two spaces: the USB PCI Configuration Register Space as defined by PCI specification 2.1 and the USB OpenHCI Host Controller Operational Register Space as defined by OHCI specification 1.0.

6.5.1 USB PCI Configuration Register

Device	IDSEL	Function Number
USB	AD12	0002b

Configuration Offset	Register	Register Access
00-01h	Vendor ID	RO
02-03h	Device ID	RO
04-05h	Command	R/W
06-07h	Status	R/WC
08h	Revision ID	RO
09-0Bh	Class Code	RO
0Ch	Reserved	-
0Dh	Latency Timer	R/W
0Eh	Header Type	RO
0Fh	Reserved	-
10-13h	USB Memory Space Base Address	R/W
14-3Bh	Reserved	-
3Ch	Interrupt Line	R/W
3Dh	Interrupt Pin	RO
3Eh	Min. Grant	R/W
3Fh	Max. Latency	R/W
40-FFh	Reserved	-

Register 00h~01h Vendor ID

Bits 15:0 Vendor ID

This is a 16-bit value assigned to SiS. The default Value is 1039h.

Register 02~03h Device ID

Bits 15:0 Device ID.

This is a 16-bit value assigned to SiS USB Host Controller. The Default Value is 7001h.

Register 04h~05h Command (The default value is 00h)

Bits 15:10 Reserved

Bit 9 Fast Back To Back.

0 : Always disabled; not supported.

Bit 8 **SERR# Enable.**
1 : Enable.
0 : Disable.

Bit 7 **Wait Cycle Control.**
0 : Always disabled; not supported.

Bit 6 **Parity Error Response.**
1 : Enable.
0 : Disable.

Bit 5 **VGA Palette Snoop.**
0: Always disabled; not supported.

Bit 4 **Memory write and invalidate enable.**
1 : Enable.
0 : Disable.

Bit 3 **Special cycle.**
0 : Always disabled; not supported.

Bit 2 **Bus master.**
1: Enable.
0 : Disable.

Bit 1 **Memory space.**
1 : Enable.
0 : Disable.

Bit 0 **IO space.**
1 : Enable.
0 : Disable.

Register 06h~07h Status(The default value is 0280h)

Bit 15 **Detected Parity Error.**
This bit is set when parity error is detected. This bit is cleared by writing a 1 to it.

Bit 14 **Signaled System Error(SERR#).**
This bit is set when SERR# is asserted. This bit is cleared by writing a 1 to it.

Bit 13 **Received Master Abort.**
This bit is set when a master cycle is terminated by master abort. This bit is cleared by writing a 1 to it.

Bit 12 **Received Target Abort.**
This bit is set when a master cycle is terminated by target abort. This bit is cleared by writing a 1 to it.

Bit 11 **Signaled Target Abort.**
This bit is set when a target cycle is terminated by target abort. This bit is cleared by writing a 1 to it. 0 : always disabled; not supported.

Bits 10:9 **DEVSEL Timing.**

01 : medium.

Bit 8 **Data Parity Error Detected.**
This bit is set when
(1) PERR# is asserted.
(2) acting as bus master.
(3) Parity Error Response bit is set. This bit is cleared by writing a 1 to it.

Bit 7 **Fast back to back capable.**
1 : Always enabled.

Bit 6 **UDF support.**
0 : Always disabled; not supported.

Bit 5 **66 MHz capable.**
0 : Always disabled; not supported.

Bits 4:0 **Reserved.**

Register 08h Revision ID

Bits 7:0 **Revision ID.**
This register is hardwired to the default value of 0E0h. The default value is 0E0h.

Register 09~0Bh Class Code (The default value is 0C0310H.)

Bits 23:16 **Base Class.**
A constant value of ‘0Ch’ identifies the device being a Serial Bus Controller.

Bits 15:8 **SUB Class.**
A constant value of ‘03h’ identifies the device being of Universal Serial Bus.

Bits 7:0 **Programming Interface.**
A constant value of ‘10h’ identifies the device being an OpenHCI Host Controller.

Register 0Ch Reserved

Register 0Dh Latency Timer

Bits 7:0 **Latency Timer.**
The default Value is 00h.

Register 0Eh Header Type

Bits 7:0 **Multiple Function Device**
The default value is 10h.

Register 0Fh Reserved

Register 10h~13h USB Memory Space Base Address Register (Default=00h)

Bits 31:12 Base Address

Bits 11:0 Reserved and hardwired to “0”.

Register 14h~3Bh Reserved

Register 3Ch Interrupt Line

Bit 7:0 Interrupt Line
The default value is 00h.

Register 3Dh Interrupt Pin

Bit 7:0 Interrupt Pin.
The default value is 01h.

Register 3Eh Minimum Grant Time

Bit 7:0 Minimum Grant Time
The default value is 00h.

Register 3Fh Maximum Latency Time

Bit 7:0 Maximum Latency Time
The default value is 00h.

6.5.2 USB OpenHCI Host Controller Operational Register

The base address of these registers are programmable by the memory base address register (USB PCI configuration register offset 10-13h). These registers should be written as Dword, byte write to these registers have unpredictable effects.

The OpenHCI Host Controller (HC) contains a set of on-chip operational registers which are mapped into a noncacheable portion of the system addressable space. These registers are used by the Host Controller Driver (HCD). According to the function of these registers, they are divided into four partitions, specifically for Control and Status, Memory Pointer, Frame Counter and Root Hub. All of the registers should be read and written as Dwords.

Reserved bits may be allocated in future releases of this specification. To ensure interoperability, the Host Controller Driver that does not use a reserved field should not assume that the reserved field contains 0. Furthermore, the Host Controller Driver should always preserve the value(s) of the reserved field. When a R/W register is modified, the Host Controller Driver should first read the register, modify the bits desired, then write the register with the reserved bits still containing the read value. Alternatively, the Host Controller Driver can maintain an in-memory copy of previously written values that can be modified and then written to the Host Controller register. When a write to set/clear register is written, bits written to reserved fields should be 0.

Host Controller Operational Registers

Offset	31 — 00
0	HcRevision
4	HcControl
8	HcCommandStatus
C	HcInterruptStatus
10	HcInterruptEnable
14	HcInterruptDisable
18	HcHCCA
1C	HcPeriodCurrentED
20	HcControlHeadED
24	HcControlCurrentED
28	HcBulkHeadED
2C	HcBulkCurrentED
30	HcDoneHead
34	HcFmInterval
38	HcFmRemaining
3C	HcFmNumber
40	HcPeriodicStart
44	HcLSThreshold
48	HcRhDescriptorA
4C	HcRhDescriptorB
50	HcRhStatus
54	HcRhPortStatus[1]
58	HcRhPortStatus[2]
100	HceControl
104	HceInput
108	HceOutput
10C	HceStatus

6.5.2.1 Control and Status Partition

Register 00h HcRevision Register

Bits 31:9 Reserved

Bit 8 Legacy

This read-only field is 1 to indicate that the legacy support registers are present in this HC.

Bits 7:0 Revision

This read-only field contains the BCD representation of the version of the HCI specification that is implemented by this HC. For example, a value of 11h corresponds to version 1.1. All of the HC implementations that are compliant with current OpenHCI 1.0 specification will have a value of 10h.

Register 04h HcControl Register

The *HcControl* register defines the operating modes for the Host Controller. Most of the fields in this register are modified only by the Host Controller Driver, except HostControllerFunctionalState and RemoteWakeupConnected.

Bits 31:11 Reserved

Bit 10 RemoteWakeupEnable

This bit is used by HCD to enable or disable the remote wakeup feature upon the detection of upstream resume signaling. When this bit is set and the **ResumeDetected** bit in *HcInterruptStatus* is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.

Since there is no remote wakeup supported, this bit is ignored.

Bit 9 RemoteWakeupConnected

This bit indicates whether HC supports remote wakeup signaling. If remote wakeup is supported and used by the system it is the responsibility of system firmware to set this bit during POST. HC clears the bit upon a hardware reset but does not alter it upon a software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.

This bit is hard-coded to ‘0’.

Bit 8 InterruptRouting

This bit determines the routing of interrupts generated by events registered in *HcInterruptStatus*. If clear, all interrupts are routed to the normal host bus interrupt mechanism. If set, interrupts are routed to the System Management Interrupt. HCD clears this bit upon a hardware reset, but it does not alter this bit upon a software reset. HCD uses this bit as a tag to indicate the ownership of HC.

Bits 7:6 HostControllerFunctionalState for USB

00b: UsbReset
01b: UsbResume
10b: UsbOperational
11b: UsbSuspend

A transition to UsbOperational from another state causes SOF generation to begin 1 ms later. HCD may determine whether HC has begun sending SOFs by reading the **StartofFrame** field of *HcInterruptStatus*.

This field may be changed by HC only when in the UsbSuspend state. HC may move from the UsbSuspend state to the UsbResume state after detecting the resume signaling from a downstream port.

HC enters USBSUSPEND after a software reset, whereas it enters USBRESET after a hardware reset. The latter also resets the Root Hub and asserts subsequent reset signaling to downstream ports.

Bit 5 **BulkListEnable**

This bit is set to enable the processing of the Bulk list in the next Frame. If cleared by HCD, processing of the Bulk list does not occur after the next SOF. HC checks this bit whenever it determines to process the list. When disabled, HCD may modify the list. If *HcBulkCurrentED* is pointing to an ED to be removed, HCD must advance the pointer by updating *HcBulkCurrentED* before re-enabling processing of the list.

Bit 4 **ControlListEnable**

This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, processing of the Control list does not occur after the next SOF. HC must check this bit whenever it determines to process the list. When disabled, HCD may modify the list. If *HcControlCurrentED* is pointing to an ED to be removed, HCD must advance the pointer by updating *HcControlCurrentED* before re-enabling processing of the list.

Bit 3 **IsochronousEnable**

This bit is used by HCD to enable/disable processing of isochronous EDs. While processing the periodic list in a Frame, HC checks the status of this bit when it finds an Isochronous ED (F=1). If set (enabled), HC continues processing the EDs. If cleared (disabled), HC halts processing of the periodic list (which now contains only isochronous EDs) and begins processing the Bulk/Control lists. Setting this bit is guaranteed to take effect in the next Frame (not the current Frame).

Bit 2 **PeriodicListEnable**

This bit is set to enable the processing of the periodic list in the next Frame. If cleared by HCD, processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list.

Bits 1:0 **ControlBulkServiceRatio**

This specifies the service ratio between Control and Bulk EDs. Before processing any of the nonperiodic lists, HC must compare the ratio specified with its internal count on how many nonempty Control EDs have been processed, in determining whether to continue serving another Control ED or switching to Bulk EDs. The internal count will be retained when crossing the frame boundary. In case of reset, HCD is responsible for restoring this value.

CBSR	No. of Control EDs Over Bulk EDs Served
0	1 : 1
1	2 : 1
2	3 : 1
3	4 : 1

Register 08h ***HcCommandStatus*** **Register**

The *HcCommandStatus* register is used by the Host Controller to receive commands issued by the Host Controller Driver, as well as reflecting the current status of the Host Controller. To the Host Controller Driver, it appears to be a "write to set" register. The Host Controller must ensure that bits written as ‘1’ become set in the register while bits written as ‘0’ remain unchanged in the register. The Host Controller Driver may issue multiple distinct commands to the

Host Controller without concern for corrupting previously issued commands. The Host Controller Driver has normal read access to all bits.

The **SchedulingOverrunCount** field indicates the number of frames with which the Host Controller has detected the scheduling overrun error. This occurs when the Periodic list does not complete before EOF. When a scheduling overrun error is detected, the Host Controller increments the counter and sets the **SchedulingOverrun** field in the *HcInterruptStatus* register.

Bits 31:18 Reserved

Bits 17:16 SchedulingOverrunCount

These bits are incremented on each scheduling overrun error. It is initialized to 00b and wraps around at 11b. This will be incremented when a scheduling overrun is detected even if **SchedulingOverrun** in *HcInterruptStatus* has already been set. This is used by HCD to monitor any persistent scheduling problems.

Bits 15:4 Reserved

Bit 3 OwnershipChangeRequest

This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the **OwnershipChange** field in *HcInterruptStatus*. After the changeover, this bit is cleared and remains so until the next request from OS HCD.

Bit 2 BulkListFilled

This bit is used to indicate whether there are any TDs on the Bulk list. It is set by HCD whenever it adds a TD to an ED in the Bulk list.

When HC begins to process the head of the Bulk list, it checks BF. As long as **BulkListFilled** is 0, HC will not start processing the Bulk list. If **BulkListFilled** is 1, HC will start processing the Bulk list and will set BF to 0. If HC finds a TD on the list, then HC will set **BulkListFilled** to 1 causing the Bulk list processing to continue. If no TD is found on the Bulk list, and if HCD does not set **BulkListFilled**, then **BulkListFilled** will still be 0 when HC completes processing the Bulk list and Bulk list processing will stop.

Bit 1 ControlListFilled

This bit is used to indicate whether there are any TDs on the Control list. It is set by HCD whenever it adds a TD to an ED in the Control list.

When HC begins to process the head of the Control list, it checks CLF. As long as **ControlListFilled** is 0, HC will not start processing the Control list. If CF is 1, HC will start processing the Control list and will set **ControlListFilled** to 0. If HC finds a TD on the list, then HC will set **ControlListFilled** to 1 causing the Control list processing to continue. If no TD is found on the Control list, and if the HCD does not set **ControlListFilled**, then **ControlListFilled** will still be 0 when HC completes processing the Control list and Control list processing will stop.

Bit 0 HostControllerReset

This bit is set by HCD to initiate a software reset of HC. Regardless of the functional state of HC, it moves to the USBSUSPEND state in which most of the operational registers are reset except those stated otherwise; e.g., the

InterruptRouting field of *HcControl*, and no Host bus accesses are allowed. This bit is cleared by HC upon the completion of the reset operation. The reset operation must be completed within 10 µs. This bit, when set, should not cause a reset to the Root Hub and no subsequent reset signaling should be asserted to its downstream ports.

Register 0Ch *HcInterruptStatus* Register

This register provides status on various events that cause hardware interrupts. When an event occurs, Host Controller sets the corresponding bit in this register. When a bit becomes set, a hardware interrupt is generated if the interrupt is enabled in the *HcInterruptEnable* register and the **MasterInterruptEnable** bit is set. The Host Controller Driver may clear specific bits in this register by writing '1' to bit positions to be cleared. The Host Controller Driver may not set any of these bits. The Host Controller will never clear the bit.

Bit 31 **Reserved and read as 0.**

Bit 30 **OwnershipChange Status**

This bit is set by HC when HCD sets the **OwnershipChangeRequest** field in *HcCommandStatus*. This event, when unmasked, will always generate an System Management Interrupt (SMI) immediately.

This bit is tied to 0b when the SMI pin is not implemented.

Bits 29:7 **Reserved**

Bit 6 **RootHubStatusChange Status**

This bit is set when the content of *HcRhStatus* or the content of any of *HcRhPortStatus*[**NumberofDownstreamPort**] has changed.

Bit 5 **FrameNumberOverflow Status**

This bit is set when the MSb of *HcFmNumber* (bit 15) changes value, from 0 to 1 or from 1 to 0, and after *HccaFrameNumber* has been updated.

Bit 4 **UnrecoverableError Status**

This bit is set when HC detects a system error not related to USB. HC should not proceed with any processing nor signaling before the system error has been corrected. HCD clears this bit after HC has been reset.

This event is not implemented and is hard-coded to '0'.

Bit 3 **ResumeDetected Status**

This bit is set when HC detects that a device on the USB is asserting resume signaling. It is the transition from no resume signaling to resume signaling causing this bit to be set. This bit is not set when HCD sets the USBRESUME state.

Bit 2 **StartofFrame Status**

This bit is set by HC at each start of a frame and after the update of *HccaFrameNumber*. HC also generates a SOF token at the same time.

Bit 1 **WritebackDoneHead Status**

This bit is set immediately after HC has written *HcDoneHead* to *HccaDoneHead.* Further updates of the *HccaDoneHead* will not occur until this bit has been cleared. HCD should only clear this bit after it has saved the content of *HccaDoneHead*.

Bit 0 **SchedulingOverrun Status**

This bit is set when the USB schedule for the current Frame overruns and after the update of *HccaFrameNumber*. A scheduling overrun will also cause the **SchedulingOverrunCount** of *HcCommandStatus* to be incremented.

Register 10h ***HcInterruptEnable* Register**

Each enable bit in the *HcInterruptEnable* register corresponds to an associated interrupt bit in the *HcInterruptStatus* register. The *HcInterruptEnable* register is used to control which events generate a hardware interrupt. When a bit is set in the *HcInterruptStatus* register AND the corresponding bit in the *HcInterruptEnable* register is set AND the **MasterInterruptEnable** bit is set, then a hardware interrupt is requested on the host bus.

Writing a '1' to a bit in this register sets the corresponding bit, whereas writing a '0' to a bit in this register leaves the corresponding bit unchanged. On read, the current value of this register is returned.

Bit 31 **A ‘0’ written to this field is ignored by HC. A '1' written to this field enables interrupt generation due to events specified in the other bits of this register. This is used by HCD as a Master Interrupt Enable.**

Bit 30 **OwnershipChange Enable**

0: Ignore

1: Enable interrupt generation due to Ownership Change.

Bits 29:7 **Reserved**

Bit 6 **RootHubStatusChange Enable**

0: Ignore

1: Enable interrupt generation due to Root Hub Status Change.

Bit 5 **FrameNumberOverflow Enable**

0: Ignore

1: Enable interrupt generation due to Frame Number Overflow.

Bit 4 **UnrecoverableError Enable**

0: Ignore

1: Enable interrupt generation due to Unrecoverable Error.

Bit 3 **ResumeDetected Enable**

0: Ignore

1: Enable interrupt generation due to Resume Detect.

Bit 2 **StartFrame Enable**

0: Ignore

1: Enable interrupt generation due to Start of Frame.

Bit 1 **WritebackDoneHead Enable**

0: Ignore

1: Enable interrupt generation due to HcDoneHead Writeback.

Bit 0 **SchedulingOverrun Enable**

0: Ignore

1: Enable interrupt generation due to Scheduling Overrun.

Register 14h ***HcInterruptDisable*** **Register**

Each disable bit in the *HcInterruptDisable* register corresponds to an associated interrupt bit in the *HcInterruptStatus* register. The *HcInterruptDisable* register is coupled with the *HcInterruptEnable* register. Thus, writing a '1' to a bit in this register clears the corresponding bit in the *HcInterruptEnable* register, whereas writing a '0' to a bit in this register leaves the corresponding bit in the *HcInterruptEnable* register unchanged. On read, the current value of the *HcInterruptEnable* register is returned.

Bit 31 **A '0' written to this field is ignored by HC. A '1' written to this field disables interrupt generation due to events specified in the other bits of this register. This field is set after a hardware or software reset.**

Bit 30 **OwnershipChange Disable**

0: Ignore

1: Disable interrupt generation due to Ownership Change.

Bits 29:7 **Reserved**

Bit 6 **RootHubStatusChange Disable**

0: Ignore

1: Disable interrupt generation due to Root Hub Status Change.

Bit 5 **FrameNumberOverflow Disable**

0: Ignore

1: Disable interrupt generation due to Frame Number Overflow.

Bit 4 **UnrecoverableError Disable**

0: Ignore

1: Disable interrupt generation due to Unrecoverable Error.

Bit 3 **ResumeDetected Disable**

0: Ignore

1: Disable interrupt generation due to Resume Detect.

Bit 2 **StartFrame Disable**

0: Ignore

1: Disable interrupt generation due to Start of Frame.

Bit 1 **WritebackDoneHead Disable**

0: Ignore

1: Disable interrupt generation due to HcDoneHead Writeback.

Bit 0 **Scheduling Overrun Disable**

0: Ignore

1: Disable interrupt generation due to Scheduling Overrun.

6.5.2.2 Memory Pointer Partition

Register 18h ***HcHCCA*** **Register**

The *HcHCCA* register contains the physical address of the Host Controller Communication Area. The Host Controller Driver determines the alignment restrictions by writing all 1s to *HcHCCA* and reading the content of *HcHCCA*. The alignment is evaluated by examining the number of zeroes in the lower order bits. The minimum alignment is 256 bytes; therefore, bits 0 through 7 must always return '0' when read. This area is used to hold the control structures and the Interrupt table that are accessed by both the Host Controller and the Host Controller Driver.

Bits 31:8 This is the base address of the Host Controller Communication Area.

Bits 7:0 Reserved and read as ‘0’.

Register 1Ch *HcPeriodCurrentED* Register

The *HcPeriodCurrentED* register contains the physical address of the current Isochronous or Interrupt Endpoint Descriptor.

Bits 31:4 PeriodCurrentED

This is used by HC to point to the head of one of the Periodic lists which will be processed in the current Frame. The content of this register is updated by HC after a periodic ED has been processed. HCD may read the content in determining which ED is currently being processed at the time of reading.

Bits 3:0 Reserved and read as “0”.

Register 20h *HcControlHeadED* Register

The *HcControlHeadED* register contains the physical address of the first Endpoint Descriptor of the Control list.

Bits 31:4 ControlHeadED

HC traverses the Control list starting with the *HcControlHeadED* pointer. The content is loaded from HCCA during the initialization of HC.

Bits 3:0 Reserved and read as “0”.

Register 24h *HcControlCurrentED* Register

The *HcControlCurrentED* register contains the physical address of the current Endpoint Descriptor of the Control list.

Bits 31:4 ControlCurrentED

This pointer is advanced to the next ED after serving the present one. HC will continue processing the list from where it left off in the last Frame. When it reaches the end of the Control list, HC checks the **ControlListFilled** of in *HcCommandStatus*. If set, it copies the content of *HcControlHeadED* to *HcControlCurrentED* and clears the bit. If not set, it does nothing. HCD is allowed to modify this register only when the **ControlListEnable** of *HcControl* is cleared. When set, HCD only reads the instantaneous value of this register. Initially, this is set to zero to indicate the end of the Control list.

Bits 3:0 Reserved and read as “0”.

Register 28h ***HcBulkHeadED*** **Register**

The *HcBulkHeadED* register contains the physical address of the first Endpoint Descriptor of the Bulk list.

Bits 31:4 BulkHeadED

HC traverses the Bulk list starting with the *HcBulkHeadED* pointer. The content is loaded from HCCA during the initialization of HC.

Bits 3:0 **Reserved and read as "0".**

Register 2Ch ***HcBulkCurrentED*** **Register**

The *HcBulkCurrentED* register contains the physical address of the current endpoint of the Bulk list. As the Bulk list will be served in a round-robin fashion, the endpoints will be ordered according to their insertion to the list.

Bits 31:4 **BulkCurrentED**

This is advanced to the next ED after the HC has served the present one. HC continues processing the list from where it left off in the last Frame. When it reaches the end of the Bulk list, HC checks the **ControlListFilled** of HcControl. If set, it copies the content of *HcBulkHeadED* to *HcBulkCurrentED* and clears the bit. If it is not set, it does nothing. HCD is only allowed to modify this register when the **BulkListEnable** of *HcControl* is cleared. When set, the HCD only reads the instantaneous value of this register. This is initially set to zero to indicate the end of the Bulk list.

Bits 3:0 **Reserved and read as "0".**

Register 30h ***HcDoneHead*** **Register**

The *HcDoneHead* register contains the physical address of the last completed Transfer Descriptor that was added to the Done queue. In normal operation, the Host Controller Driver should not need to read this register as its content is periodically written to the HCCA.

Bits 31:4 DoneHead

When a TD is completed, HC writes the content of *HcDoneHead* to the NextTD field of the TD. HC then overwrites the content of *HcDoneHead* with the address of this TD.

This is set to zero whenever HC writes the content of this register to HCCA. It also sets the **WritebackDoneHead** of *HcInterruptStatus*.

Bits 3:0 **Reserved and read as "0".**

6.5.2.3 Frame Counter Partition

Register 34h ***HcFmInterval*** **Register**

The *HcFmInterval* register contains a 14-bit value which indicates the bit time interval in a Frame, (i.e., between two consecutive SOFs), and a 15-bit value indicating the Full Speed maximum packet size that the Host Controller may transmit or receive without causing scheduling overrun. The Host Controller Driver may carry out minor adjustment on the **FrameInterval** by writing a new value over the present one at each SOF. This provides the programmability necessary for the Host Controller to synchronize with an external clocking resource and to adjust any unknown local clock offset.

Bit 31 FrameIntervalToggle

HCD toggles this bit whenever it loads a new value to **FrameInterval**.

Bits 30:16 FSLargestDataPacket

This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the HCD.

Bits 15:14 Reserved

Bits 13:0 FrameInterval

This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999.

HCD should store the current value of this field before resetting HC. By setting the **HostControllerReset** field of *HcCommandStatus* as this will cause the HC to reset this field to its nominal value. HCD may choose to restore the stored value upon the completion of the Reset sequence.

Register 38h *HcFmRemaining* Register

The *HcFmRemaining* register is a 14-bit down counter showing the bit time remaining in the current Frame.

Bit 31 FrameRemainingToggle

This bit is loaded from the **FrameIntervalToggle** field of *HcFmInterval* whenever **FrameRemaining** reaches 0. This bit is used by HCD for the synchronization between **FrameInterval** and **FrameRemaining**.

Bits 30:14 Reserved

Bits 13:0 FrameRemaining

This counter is decremented at each bit time. When it reaches zero, it is reset by loading the **FrameInterval** value specified in *HcFmInterval* at the next bit time boundary. When entering the USBOPERATIONAL state, HC re-loads the content with the **FrameInterval** of *HcFmInterval* and uses the updated value from the next SOF.

Register 3Ch *HcFmNumber* Register

The *HcFmNumber* register is a 16-bit counter. It provides a timing reference among events happening in the Host Controller and the Host Controller Driver. The Host Controller Driver may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.

Bits 31:16 Reserved

Bits 15:0 FrameNumber

This is incremented when *HcFmRemaining* is re-loaded. It will be rolled over to 0h after ffffh. When entering the USBOPERATIONAL state, this will be incremented automatically. The content will be written to HCCA after HC has incremented the **FrameNumber** at each frame boundary and sent a SOF but before HC reads the first ED in that Frame. After writing to HCCA, HC will set the **StartofFrame** in *HcInterruptStatus*.

Register 40h *HcPeriodicStart* Register

The *HcPeriodicStart* register has a 14-bit programmable value which determines when is the earliest time HC should start processing the periodic list.

Bits 31:14 Reserved

Bits 13:0 PeriodicStart

After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10% off from *HcFmInterval*. A typical value will be 3E67h. When *HcFmRemaining* reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

Register 44h *HcLSThreshold* Register

The *HcLSThreshold* register contains an 11-bit value used by the Host Controller to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. Neither the Host Controller nor the Host Controller Driver are allowed to change this value.

Bits 31:12 Reserved

Bits 11:0 LSThreshold

This field contains a value which is compared to the **FrameRemaining** field prior to initiating a Low Speed transaction. The transaction is started only if **FrameRemaining** ≥ this field. The value is calculated by HCD with the consideration of transmission and setup overhead.

6.5.2.4 Root Hub Partition

All registers included in this partition are dedicated to the USB Root Hub which is an integral part of the Host Controller though still a functionally separate entity. The HCD emulates USBD accesses to the Root Hub via a register interface. The HCD maintains many USB-defined hub features which are not required to be supported in hardware. For example, the Hub's Device, Configuration, Interface, and Endpoint Descriptors are maintained only in the HCD as well as some static fields of the Class Descriptor. The HCD also maintains and decodes the Root Hub's device address as well as other trivial operations which are better suited to software than hardware.

The Root Hub register interface is otherwise developed to maintain similarity of bit organization and operation to typical hubs which are found in the system. Below are four register definitions: *HcRhDescriptorA*, *HcRhDescriptorB*, *HcRhStatus*, and *HcRhPortStatus[1:2]*. Each register is read and written as a Dword. These registers are only written during initialization to correspond with the system implementation. The *HcRhDescriptorA* and *HcRhDescriptorB* registers should be implemented such that they are writeable regardless of the HC USB state. *HcRhStatus* and *HcRhPortStatus* must be writeable during the USBOPERATIONAL state.

Register 48h *HcRhDescriptorA* Register

The *HcRhDescriptorA* register is the first register of two describing the characteristics of the Root Hub. Reset values are implementation-specific. The descriptor length (11), descriptor type (TBD), and hub controller current (0) fields of the hub Class Descriptor are emulated by the HCD. All other fields are located in the *HcRhDescriptorA* and *HcRhDescriptorB* registers.

Bits 31:24 PowerOnToPowerGoodTime

This byte specifies the duration HCD has to wait before accessing a powered-on port of the Root Hub. It is implementation-specific. The unit of time is 2 ms. The duration is calculated as **POTPGT** * 2 ms.

Bits 23:13 Reserved

Bit 12 NoOverCurrentProtection

This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the **OverCurrentProtectionMode** field specifies global or per-port reporting.

0: Over-current status is reported collectively for all downstream ports

1: No overcurrent protection supported

Bit 11 OverCurrentProtectionMode

This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, this fields should reflect the same mode as **PowerSwitchingMode**. This field is valid only if the **NoOverCurrentProtection** field is cleared.

0: over-current status is reported collectively for all downstream ports

1: over-current status is reported on a per-port basis

Bit 10 DeviceType

This bit specifies that the Root Hub is not a compound device. The Root Hub is not permitted to be a compound device. This field should always read/write 0.

Bit 9 NoPowerSwitching

These bits are used to specify whether power switching is supported or port are always powered. SiS Chip USB HC supports global power switching mode. When this bit is cleared, the **PowerSwitchingMode** specifies global or per-port switching.

0: Ports are power switched

1: Ports are always powered on when the HC is powered on

Bit 8 PowerSwitchingMode

This bit is used to specify how the power switching of the Root Hub ports is controlled. SiS Chip USB HC supports global power switching mode. This field is only valid if the **NoPowerSwitching** field is cleared.

0: all ports are powered at the same time.

1: Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching. If the PortPowerControlMask bit is set, the port responds only to port power commands (Set/ClearPortPower). If the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).

Bits 7:0 NumberDownstreamPorts

These bits specify the number of downstream ports supported by the Root Hub.

SiS Chip USB HC supports two downstream ports.

Register 4Ch *HcRhDescriptorB* Register

The *HcRhDescriptorB* register is the second register of two describing the characteristics of the Root Hub. These fields are written during initialization to configure the Root Hub.

Bits 31:16 PortPowerControlMask

Each bit indicates if a port is affected by a global power control command when **PowerSwitchingMode** is set. When set, the port's power state is only affected by per-port power control (**Set/ClearPortPower**). When cleared, the port is controlled by the global power switch (**Set/ClearGlobalPower**). If the device is configured to global switching mode (**PowerSwitchingMode**=0), this field is not valid.

SiS Chip USB HC implements global power switching.

bit 0: Reserved

bit 1: Ganged-power mask on Port #1

bit 2: Ganged-power mask on Port #2

...

bit15: Ganged-power mask on Port #15

Bits 15:0 DeviceRemovable

Each bit is dedicated to a port of the Root Hub. When cleared, the attached device is removable. When set, the attached device is not removable.

bit 0: Reserved

bit 1: Device attached to Port #1

bit 2: Device attached to Port #2

...

bit15: Device attached to Port #15

Register 50h *HcRhStatus* Register

The *HcRhStatus* register is divided into two parts. The lower word of a Dword represents the Hub Status field and the upper word represents the Hub Status Change field. Reserved bits should always be written '0'.

Bit 31 ClearRemoteWakeupEnable(write)

Writing a '1' clears DeviceRemoveWakeupEnable. Writing a '0' has no effect.

Bits 30:18 Reserved

Bit 17 OverCurrentIndicatorChange

This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a ‘1’. Writing a ‘0’ has no effect.

Bit 16 LocalPowerStatusChange(read)

The Root Hub does not support the local power status feature; thus, this bit is always read as ‘0’.

SetGlobalPower(write)

In global power mode (**PowerSwitchingMode**=0), This bit is written to ‘1’ to turn on power to all ports (clear **PortPowerStatus**). In per-port power mode, it sets **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a ‘0’ has no effect.

Bit 15 DeviceRemoteWakeupEnable(read)

This bit enables a **ConnectStatusChange** bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the **ResumeDetected** interrupt.

0: **ConnectStatusChange** is not a remote wakeup event.

1: **ConnectStatusChange** is a remote wakeup event.

SetRemoteWakeupEnable(write)

Writing a '1' sets **DeviceRemoveWakeupEnable**. Writing a '0' has no effect.

Bits 14:2 Reserved

Bit 1 OverCurrentIndicator

This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented this bit is always ‘0’

Bit 0 LocalPowerStatus(read)

The Root Hub does not support the local power status feature; thus, this bit is always read as ‘0’.

ClearGlobalPower(write)

In global power mode (**PowerSwitchingMode**=0), This bit is written to ‘1’ to turn off power to all ports (clear **PortPowerStatus**). In per-port power mode, it clears **PortPowerStatus** only on ports whose **PortPowerControlMask** bit is not set. Writing a ‘0’ has no effect.

Register 54h/58h *HcRhPortStatus*[1:2] Register

The *HcRhPortStatus*[1:2] register is used to control and report port events on a per-port basis. Two *HcRhPortStatus* registers that are implemented in hardware. The lower word is used to reflect the port status, whereas the upper word reflects the status change bits. Some status bits are implemented with special write behavior (see below). If a transaction (token through handshake) is in progress when a write to change port status occurs, the resulting port status change must be postponed until the transaction completes. Reserved bits should always be written '0'.

Bits 31:21 Reserved

Bit 20 PortResetStatusChange

This bit is set at the end of the 10-ms port reset signal.

The HCD writes a ‘1’ to clear this bit. Writing a ‘0’ has no effect.

0: port reset is not complete

1: port reset is complete

Bit 19 PortOverCurrentIndicatorChange

This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the **PortOverCurrentIndicator** bit. The HCD writes a ‘1’ to clear this bit. Writing a ‘0’ has no effect.

0: no change in PortOverCurrentIndicator

1: PortOverCurrentIndicator has changed

Bit 18 PortSuspendStatusChange

This bit is set when the full resume sequence has been completed. This sequence includes the 20-s resume pulse, LS EOP, and 3-ms resychronization delay. The HCD writes a ‘1’ to clear this bit. Writing a ‘0’ has no effect. This bit is also cleared when **ResetStatusChange** is set.

0: resume is not completed

1: resume completed

Bit 17 PortEnableStatusChange

This bit is set when hardware events cause the **PortEnableStatus** bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a ‘1’ to clear this bit. Writing a ‘0’ has no effect.

0: no change in **PortEnableStatus**

1: change in PortEnableStatus

Bit 16 ConnectStatusChange

This bit is set whenever a connect or disconnect event occurs. The HCD writes a ‘1’ to clear this bit. Writing a ‘0’ has no effect. If **CurrentConnectStatus** is cleared when a **SetPortReset, SetPortEnable**, or **SetPortSuspend** write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected.

0: no change in CurrentConnectStatus

1: change in CurrentConnectStatus

Note: If the **DeviceRemovable[NDP]** bit is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.

Bits 15:10 Reserved

Bit 9 LowSpeedDeviceAttached(read)

This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the **CurrentConnectStatus** is set.

0: full speed device attached

1: low speed device attached

ClearPortPower(write)

The HCD clears the **PortPowerStatus** bit by writing a ‘1’ to this bit. Writing a ‘0’ has no effect.

Bit 8 **PortPowerStatus(read)**

This bit reflects the port’s power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing **SetPortPower** or **SetGlobalPower**. HCD clears this bit by writing **ClearPortPower** or **ClearGlobalPower**. Which power control switches are enabled is determined by **PowerSwitchingMode** and **PortPortControlMask[NDP]**. In global switching mode (**PowerSwitchingMode**=0), only **Set/ClearGlobalPower** controls this bit. In per-port power switching (**PowerSwitchingMode**=1), if the **PortPowerControlMask[NDP]** bit for the port is set, only **Set/ClearPortPower** commands are enabled. If the mask is not set, only **Set/ClearGlobalPower** commands are enabled. When port power is disabled, **CurrentConnectStatus**, **PortEnableStatus**, **PortSuspendStatus**, and **PortResetStatus** should be reset.

0: port power is off

1: port power is on

SetPortPower(write)

The HCD writes a ‘1’ to set the **PortPowerStatus** bit. Writing a ‘0’ has no effect.
Note: This bit is always reads ‘1b’ if power switching is not supported.

Bits 7:5 **Reserved**

Bit 4 **PortResetStatus(read)**

When this bit is set by a write to **SetPortReset**, port reset signaling is asserted. When reset is completed, this bit is cleared when **PortResetStatusChange** is set. This bit cannot be set if **CurrentConnectStatus** is cleared.

0: port reset signal is not active

1: port reset signal is active

SetPortReset(write)

The HCD sets the port reset signaling by writing a ‘1’ to this bit. Writing a ‘0’ has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortResetStatus**, but instead sets **ConnectStatusChange**. This informs the driver that it attempted to reset a disconnected port.

Bit 3 **PortOverCurrentIndicator(read)**

This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal

0: no overcurrent condition.

1: overcurrent condition detected.

ClearSuspendStatus(write)

The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if **PortSuspendStatus** is set.

Bit 2 **PortSuspendStatus(read)**

This bit indicates the port is suspended or in the resume sequence. It is set by a **SetSuspendState** write and cleared when **PortSuspendStatusChange** is set at the end of the resume interval. This bit cannot be set if **CurrentConnectStatus** is cleared. This bit is also cleared when **PortResetStatusChange** is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC.

0: port is not suspended

1: port is suspended

SetPortSuspend(write)

The HCD sets the **PortSuspendStatus** bit by writing a '1' to this bit. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortSuspendStatus**; instead it sets **ConnectStatusChange**. This informs the driver that it attempted to suspend a disconnected port.

Bit 1 **PortEnableStatus(read)**

This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes **PortEnabledStatusChange** to be set. HCD sets this bit by writing **SetPortEnable** and clears it by writing **ClearPortEnable**. This bit cannot be set when **CurrentConnectStatus** is cleared. This bit is also set, if not already, at the completion of a port reset when **ResetStatusChange** is set or port suspend when **SuspendStatusChange** is set.

0: port is disabled

1: port is enabled

SetPortEnable(write)

The HCD sets **PortEnableStatus** by writing a '1'. Writing a '0' has no effect. If **CurrentConnectStatus** is cleared, this write does not set **PortEnableStatus**, but instead sets **ConnectStatusChange**. This informs the driver that it attempted to enable a disconnected port.

Bit 0 **CurrentConnectStatus(read)**

This bit reflects the current state of the downstream port.

0: no device connected

1: device connected

ClearPortEnable(write)

The HCD writes a '1' to this bit to clear the **PortEnableStatus** bit. Writing a '0' has no effect. The **CurrentConnectStatus** is not affected by any write.

Note: This bit is always read '1b' when the attached device is nonremovable (**DeviceRemoveable[NDP]**).

6.5.2.5 Legacy Support Registers

Four operational registers are used to provide the legacy support . Each of these registers is located on a 32-bit boundary. The offset of these registers is relative to the base address of the Host Controller operational registers with *HceControl* located at offset 100h.

Table 6-1

Offset	Register	Description
100h	*HceControl*	Used to enable and control the emulation hardware and report various status information.
104h	*HceInput*	Emulation side of the legacy Input Buffer register.
108h	*HceOutput*	Emulation side of the legacy Output Buffer register where keyboard and mouse data is to be written by software.
10Ch	*HceStatus*	Emulation side of the legacy Status register.

Three of the operational registers (*HceStatus*, *HceInput*, *HceOutput*) are accessible at I/O address 60h and 64h when emulation is enabled. Reads and writes to the registers using I/O addresses have side effects as outlined in the Table 6-2.

Table 6-2

I/O Address	Cycle Type	Register Contents Accessed/Modified	Side Effects
60h	IN	*HceOutput*	IN from port 60h will set **OutputFull** in *HceStatus* to 0
60h	OUT	*HceInput*	OUT to port 60h will set **InputFull** to 1 and **CmdData** to 0 in *HceStatus*.
64h	IN	*HceStatus*	IN from port 64h returns current value of *HceStatus* with no other side effect.
64h	OUT	*HceInput*	OUT to port 64h will set **InputFull** to 0 and CmdData in *HceStatus* to 1.

Register 100h HceControl Register

Bits 31:9 Reserved.

and read as 0s.

Bit 8 A20State

Indicates current state of Gate A20 on keyboard controller. Used to compare against value to 60h when GateA20Sequence is active.

Bit 7 IRQ12Active

Indicates that a positive transition on IRQ12 from keyboard controller has occurred. SW may write a 1 to this bit to clear it (set it to 0). SW write of a 0 to this bit has no effect.

Bit 6 **IRQ1Active**

Indicates that a positive transition on IRQ1 from keyboard controller has occurred. SW may write a 1 to this bit to clear it (set it to 0). SW write of a 0 to this bit has no effect.

Bit 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.

Bit 4 **ExternalIRQEn**

When set to 1, IRQ1 and IRQ12 from the keyboard controller causes an emulation interrupt. The function controlled by this bit is independent of the setting of the **EmulationEnable** bit in this register.

Bit 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, then IRQ1 is generated; if it is 1, then an IRQ12 is generated.

Bit 2 **CharacterPending**

When set, an emulation interrupt is generated when the **OutputFull** bit of the *HceStatus* register is set to 0.

Bit 1 **EmulationInterrupt**

This bit is a static decode of the emulation interrupt condition.

Bit 0 **EmulationEnable**

When set to 1, the HC is enabled for legacy emulation. The HC decodes accesses to I/O registers 60h and 64h and generates IRQ1 and/or IRQ12 when appropriate. Additionally, the HC generate s an emulation interrupt at appropriate times to invoke the emulation software.

Register 104h ***HceInput*** **Register**

Bits 31:8 **Reserved**

Bits 7:0 **InputData**

This register holds data that is written to I/O ports 60h and 64h.

I/O data that is written to ports 60h and 64h is captured in this register when emulation is enabled. This register may be read or written directly by accessing it with its memory address in the Host Controller’s operational register space. When accessed directly with a memory cycle, reads and writes of this register have no side effects.

Register 108h ***HceOutput*** **Register**

Bits 31:8 **Reserved**

Bits 7:0 **OutputData**

This register hosts data that is returned when an I/O read of port 60h is performed by application software.

The data placed in this register by the emulation software is returned when I/O port 60h is read and emulation is enabled. On a read of this location, the OutputFull bit in *HceStatus* is set to 0.

Register 10Ch ***HceStatus* Register**

Bits 31:8 **Reserved**

Bit 7 **Parity**

Indicates parity error on keyboard/mouse data.

Bit 6 **Time-out**

Used to indicate a time-out

Bit 5 **AuxOutputFull**

IRQ12 is asserted whenever this bit is set to 1 and **OutputFull** is set to 1 and the **IRQEn** bit is set.

Bit 4 **Inhibit Switch**

This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.

Bit 3 **CmdData**

The HC sets this bit to 0 on an I/O write to port 60h and to 1 on an I/O write to port 64h.

Bit 2 **Flag**

Nominally used as a system flag by software to indicate a warm or cold boot.

Bit 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.

Bit 0 **OutputFull**

The HC sets 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 an IRQ12 is generated as 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.

The contents of the *HceStatus* Register are returned on an I/O Read of port 64h when emulation is enabled. Reads and writes of port 60h and writes to port 64h can cause changes in this register. Emulation software can directly access this register through its memory address in the Host Controller's operational register space. Accessing this register through its memory address produces no side effects.

7. Electrical Characteristics

7.1 Absolute Maximum Ratings

Parameter	Min.	Max.	Unit
Ambient operation temperature	0	70	^{0}C
Storage temperature	-40	125	^{0}C
Input voltage	-0.3	Vcc+0.3	V
Output voltage	-0.5	3.3	V

NOTE:
Stress above these listed may cause permanent damage to device. Functional operation of this device should be restricted to the conditions described under operating conditions.

7.2 DC Characteristics

Table 7-1 DC Characteristics
Ta= 0 - 70^{o}C, Gnd= 0V, Vcc5= 5V$\pm$5%, Vcc=3.3V$\pm$5%

Symbol	Parameter	Min	Max	Unit	Notes
V_{IL1}	Input Low Voltage	-0.3	0.8	V	Note 1
V_{IH1}	Input High Voltage	2.2	V_{CC}+0.3	V	
V_{T1-}	Schmitt Trigger Threshold			V	Note 2
	Voltage Falling Edge	1.18	1.88		
V_{T1+}	Schmitt Trigger Threshold				Note 2
	Voltage Rising Edge	1.63	1.88		
V_{H1}	Hysteresis Voltage	0.45		V	Note 2
V_{OL2}	Output Low Voltage		0.4	V	
V_{OH2}	Output High Voltage	2.0	2.4	V	
I_{OL1}	Output Low Current	8		mA	Note 3, 8
I_{OH1}	Output High Current	-8		mA	Note 3, 8
I_{OL2}	Output Low Current	8, 16		mA	Note 4, 8
I_{OH2}	Output High Current	-8, 16		mA	Note 4, 8
I_{OL3}	Output Low Current	12, 16		mA	Note 5, 8
I_{OH3}	Output High Current	-12, -16		mA	Note 5, 8
I_{OL4}	Output Low Current	4, 8		mA	Note 6, 8
I_{OH4}	Output High Current	-4, -8		mA	Note 6, 8
I_{OL5}	Output Low Current	4		mA	Note 7
I_{OH5}	Output high Current	-4		mA	Note 7
I_{IH}	Input Leakage Current		-10	µA	
I_{IL}	Input Leakage Current		+10	µA	
C_{IN}	Input Capacitance		12	pF	Fc=1 Mhz
C_{OUT}	Output Capacitance		12	pF	Fc=1 Mhz
$C_{I/O}$	I/O Capacitance		12	pF	Fc=1 Mhz

NOTE:

1. The RTC-related ten pins only have 3.3V input tolerance. They are ONCTL#, RING, PWRGD, PSRSTB#, OSCI, OSCO, SWITCH#, PWRBT#, GPIO5, GPIO10.
2. V_{T1-}, V_{T1+} and V_{H1} are applicable to PWRGD
3. I_{OL1} and I_{OH1} are applicable to the following signals: AD[31:0], C/BE[3:0]#, GNT[3:0]#, STOP#, DEVSEL#, TRDY#, IRDY#, FRAME#, PHLDA#, GPO, PAR, PCIRST
4. I_{OL2} and I_{OH2} are applicable to the following signals: CAS[7:0]#
5. I_{OL3} and I_{OH3} are applicable to the following signals: MA[14:0], RAMW#A/B, SRAS#, SCAS#
6. I_{OL4} and I_{OH4} are applicable to the following signals: RAS[3:0]#, ADSC#, ADSV#
7. I_{OL5} and I_{OH5} are applicable to the following signals: KRE#, STPCLK#, INIT, SMI#, HA[31:3], W/R#, EADS#, NA#, BRDY#, KEN#, A20M#,BOFF#, CPURST, MD, HD, HBE[7:0]#
8. The driving current is programmed. Please refer to register description.

7.3 AC Characteristics

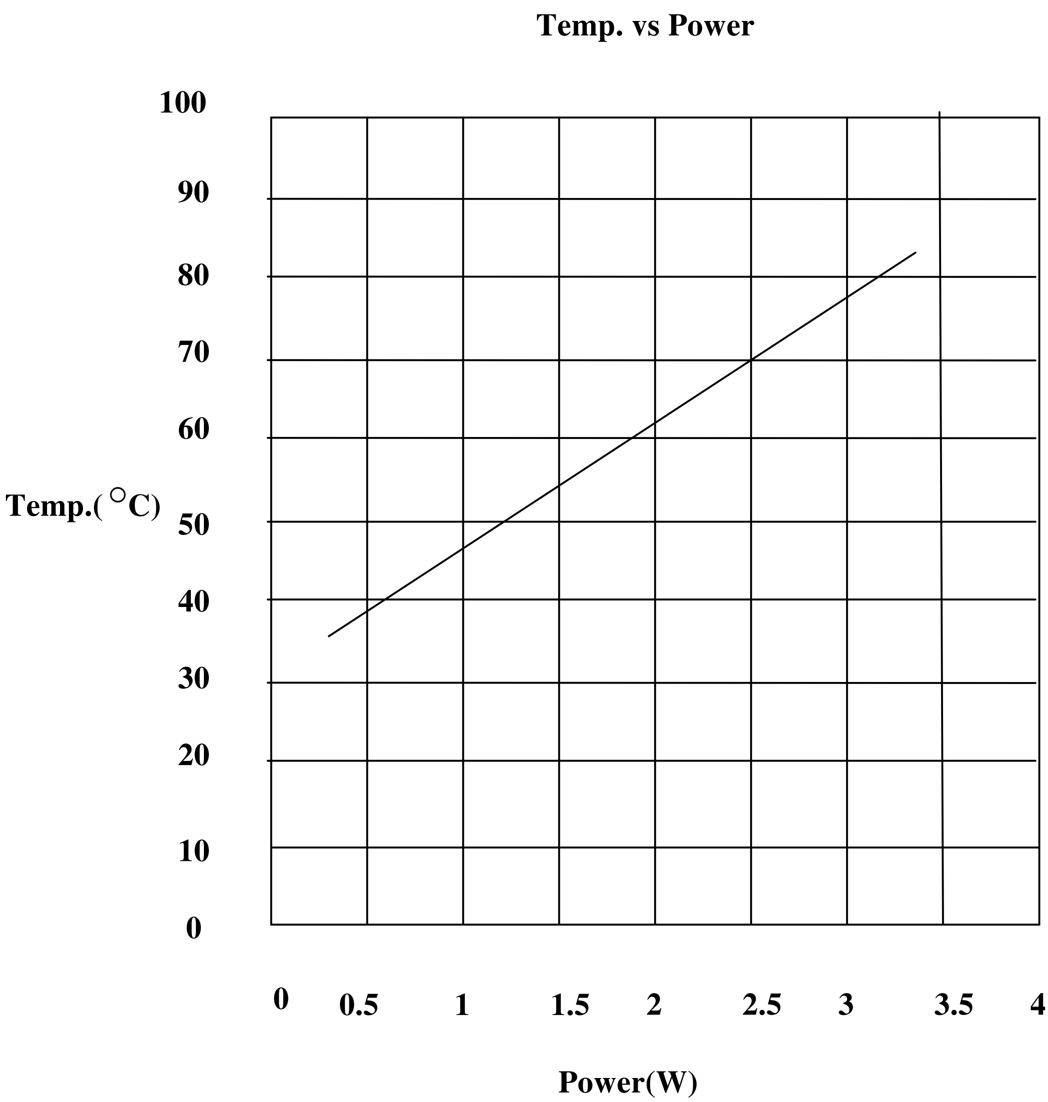
Reserved

8. Thermal Analysis and RTC Power Consumption

8.1 Chip Thermal Analysis

Room Temperature : 25°C
Location of thermal probe : center of the chip
Result :

Consumed Power (Walt)	0.37	0.84	1.39	2	2.68	3.43
Temp. (°C)	32.8	38.6	47.4	59	71.6	82.8



The formula of the characteristics line is
Temp. = 25.27 + 16.9 * **Power**(°**C)**

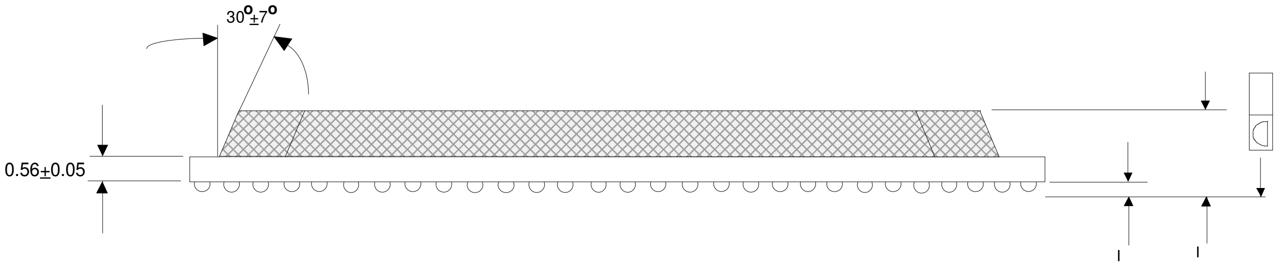
8.2 Internal RTC Power Consumption

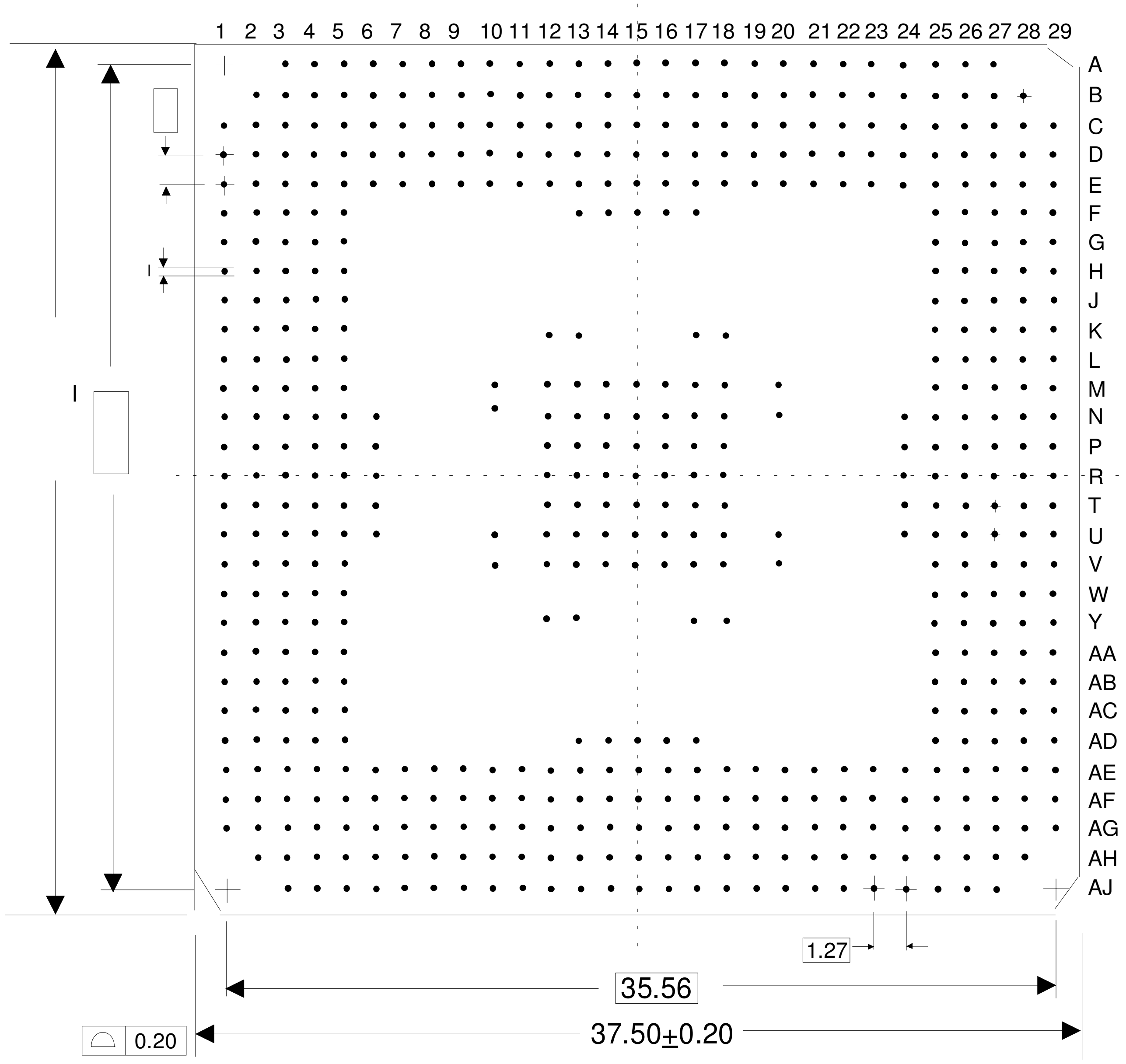
RTCVDD (V)	Operation Current (uA)	Power Consumption (uW)
3.22	5.0	16.1
2.93	4.2	12.31
2.64	3.5	9.24
2.36	2.9	6.84
2.07	2.3	4.76

The minimun operation voltage of internal RTC (RTCVDD) is 2.0V, the recommended voltage is 2.2V to 2.7V.

9. Mechanical Dimension (Top view)

(Unit: mm)





10. COPYRIGHT NOTICE

TRADEMARKS

SiS is a registered trademark of Silicon Integrated Systems Corp.

All brand or product names mentioned are trademarks or registered trademarks of their respective holders.

DISCLAIMER

Silicon Integrated Systems Corp. makes no representations or warranties regarding the contents of this manual. We reserve the right to revise the manual or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

The information contained in this manual is provided for general use by our customers. Our customers should be aware that the personal computer field ins the subject of many patents. Our customers should ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Silicon Integrated Systems Corp. To respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013.

Table of Contents