# Pentium PCI System I/O Chipset

# SiS5595

## *Preliminary*

Rev. 2.0

Nov. 02, 1998

# Contents

# Figures

# Tables

# 1 FEATURES

## 1.1 SiS5595 PCI SYSTEM I/O

- **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.
- **Supports both Desktop and Mobile Advanced Power Management Logic**
  - Meets ACPI 1.0 Requirements.
  - Supports Both ACPI and Legacy PMU.
  - Supports Suspend to RAM.
  - Supports Suspend to Hard Disk.
  - Optionally Tri-state ISA bus in low power state.
  - Supports Battery Management and LB/LLB/AC Indicator.
  - Supports CPU's SMM Mode Interface.
  - Supports CPU Stop Clock.
  - Supports Power Button of ACPI.
  - Supports three system timers and SMI# watchdog timer.
  - Supports Automatic Power Control.
  - Supports Modem Ring-in, RTC Alarm Wake up.
  - Supports Thermal Detection.
  - Supports GPIOs, and GPOs for External Devices Control.
  - Supports Programmable Chip Select.
  - Supports PCI Bus Power Management Interface Spec. 1.0 .
  - Supports Pentium II Sleep State.

- **Enhanced DMA Functions**
  - 8-, 16- bit DMA Data Transfer.
  - Two 8237A Compatible DMA Controllers with Seven Independent Programmable Channels.
  - Provide the Readability of the two 8237 Associated Registers.
  - Support Distributed DMA.
  - Support PC/PCI DMA.
  - Per DMA channel programmable in legacy, DDMA or PC/PCI DMA mode operation.
- **Integrated Two 8259A Interrupt Controllers**
  - 14 Independently Programmable Channels for Level- or Edge-triggered Interrupts.
  - Provide the Readability of the two 8259A Associated Registers.
  - Support Serial IRQ.
  - Support the Reroutability for the PCI Interrupts.
- **Three Programmable 16-bit Counters compatible with 8254**
  - System Timer Interrupt.
  - Generate Refresh Request.
  - Speaker Tone Output.
  - Provide the Readability of the 8254 Associated Registers.
- **Integrated Keyboard Controller**
  - Hardwired Logic Provides Instant Response.
  - Supports PS/2 Mouse Interface.
  - Supports Keyboard Password Security or Hot Key Power On Function.
  - Supports Hot Key "Sleep" Function.
  - Programmable Enable and Disable for Keyboard Controller and PS/2 Mouse.
- **Integrated Real Time Clock(RTC) with 256B CMOS SRAM**
  - Supports ACPI Day of Month Alarm/Month Alarm.
  - Supports various Power Up events, such as Button Up, Alarm Up, Ring Up, GPIO5/PME0# Up, GPIO10/ PME1# Up, Password Security Up, and Hotkey Up.

- Supports various Power Down Events, like Software Power-down, Button Power-down, and ACPI S3 Power-down.
- Supports Power Supply '98.
- Provides RTC year 2000 solution.

- **Integrated Frequency Ratio Control Logic for Pentium II CPU**
- **Universal Serial Bus Host Controller**
  - Open HCI Host Controller with Root Hub.
  - Two USB Ports.
  - Supports Legacy Devices.
  - Supports Over Current Detection.
- **Integrated Hardware Monitor Logic**
  - Up to 5 Positive Voltage Monitoring Inputs.
  - Two Fan Speed Monitoring Inputs.
  - One Temperature Sensings.
  - Supports thermister- or diode- temperature sensing for Pentium II CPU.
  - Threshold Comparison of all Monitored Values.
- **Supports $I^2C$ Serial Bus/ SMBUS**
- **Supports 2MB Flash ROM Interface**
- **208 pins PQFP Package**
- **5V CMOS Technology**

## 1.2 FUNCTIONAL BLOCK DIAGRAM

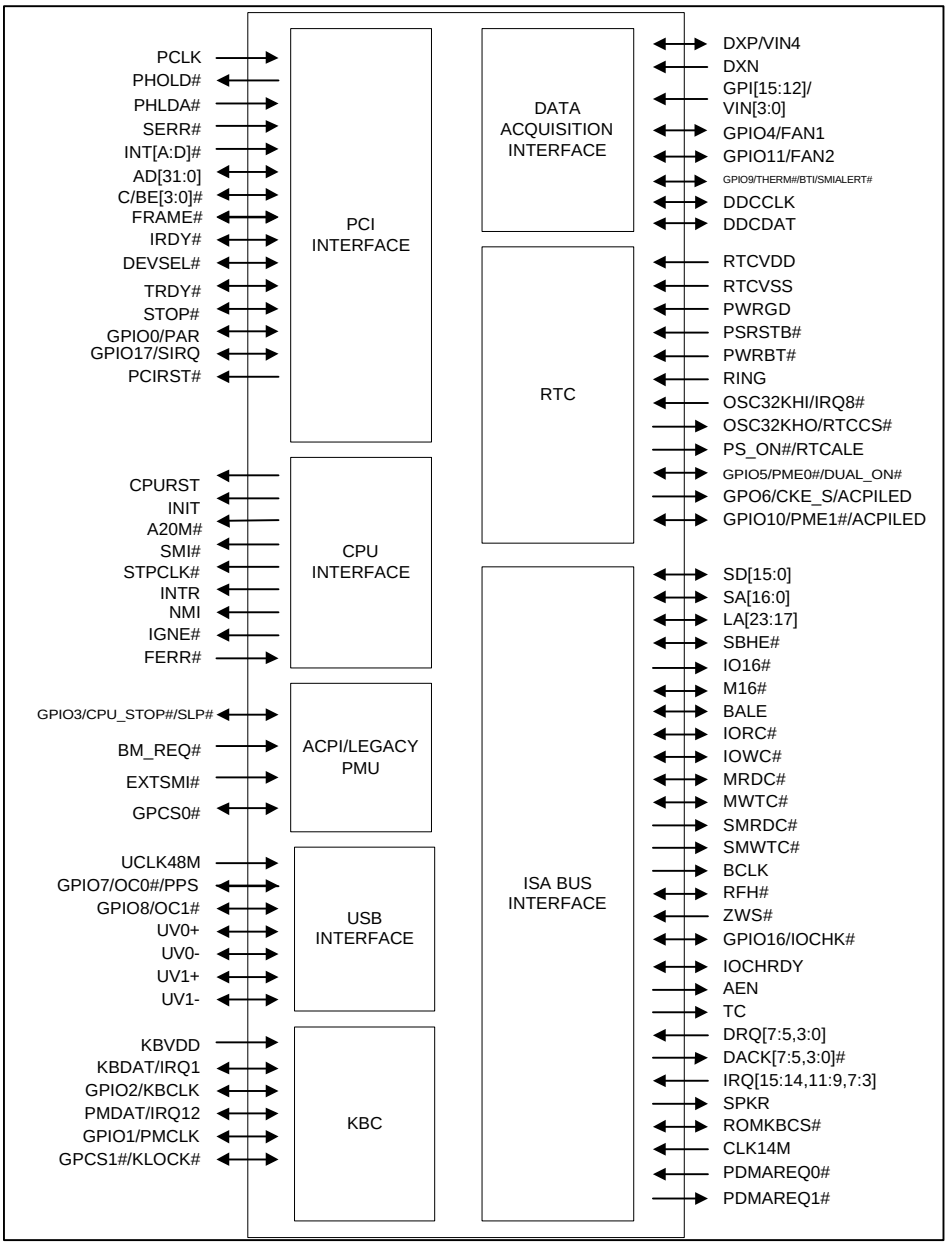


**Figure 1.2-1 Functional Block Diagram**

**Table 1.2-1 Multi-Function Pins**

| | | |
|---|---|---|
| GPIO0/PAR | GPIO8/OC1# | GPIO16/IOCHK# |
| GPIO1/PMCLK | GPIO9/THERM#/ BTI/SMBALERT# | GPIO17/SIRQ |
| GPIO2/KBCLK | GPIO10/PME1#/ACPILED | GPCS1#/KLOCK# |
| GPIO3/CPU_STOP#/SLP# | GPIO11/FAN2 | OSC32KHI/IRQ8# |
| GPIO4/FAN1 | GPI12/VIN0 | OSC32KHO/RTCCS# |
| GPIO5/PME0#/DUAL_ON# | GPI13/VIN1 | PS_ON#/RTCALE |
| GPO6/CKE_S/ACPILED | GPI14/VIN2 | KBDAT/IRQ1 |
| GPIO7/OC0#/PPS | GPI15/VIN3 | PMDAT/IRQ12 |
| DXP/VIN4 | | |

# 2 PIN ASSIGNMENT (TOP VIEW)

## 2.1 SiS5595 PIN ASSIGNMENT (TOP VIEW)

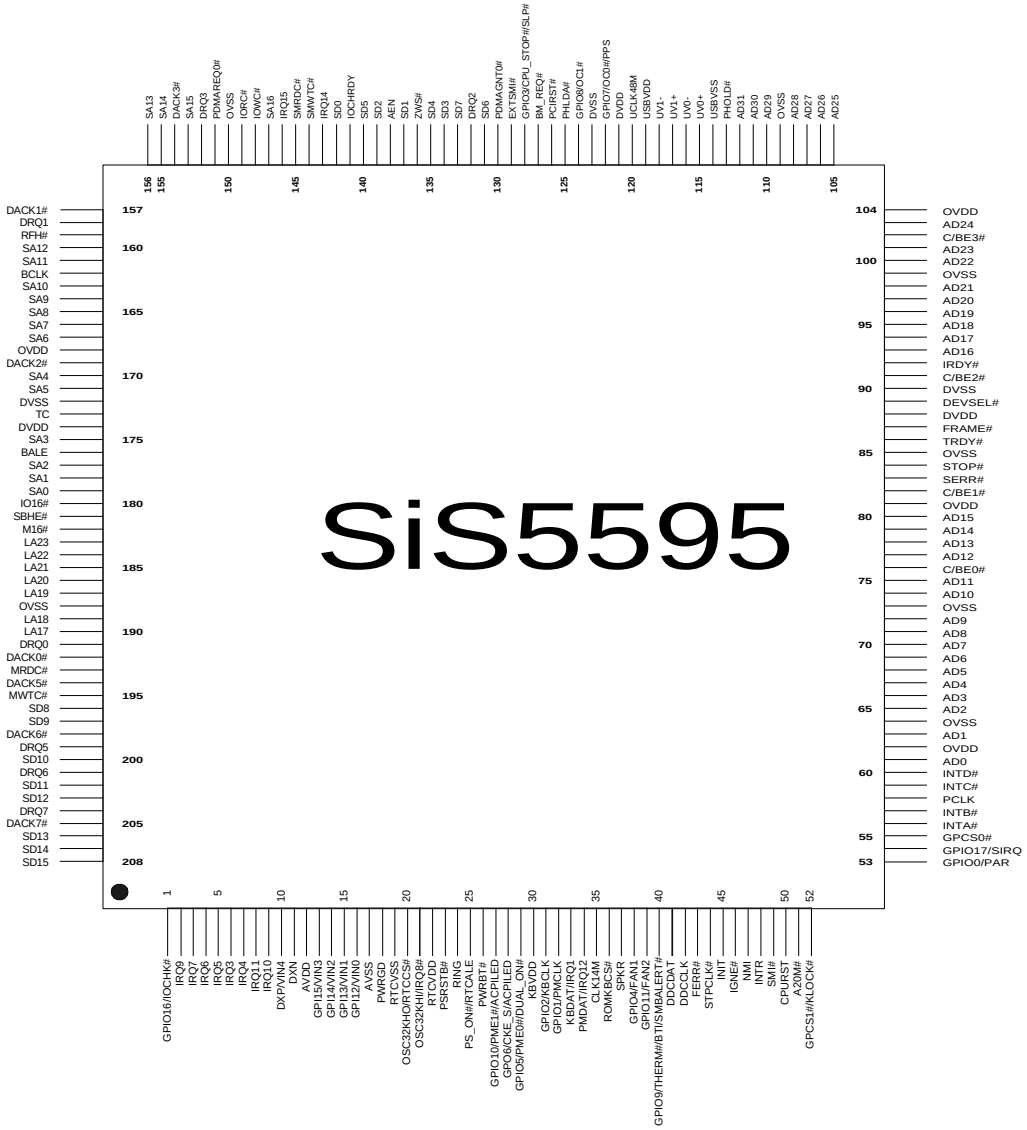


**Figure 2.1-1 pin assignment**

## 2.2 SiS5595 CHIP ALPHABETICAL PIN LIST

| SIGNAL NAME | SiS5595 PIN NO. |
|---|---|
| A20M# | 51 |
| AD0 | 61 |
| AD1 | 63 |
| AD2 | 65 |
| AD3 | 66 |
| AD4 | 67 |
| AD5 | 68 |
| AD6 | 69 |
| AD7 | 70 |
| AD8 | 71 |
| AD9 | 72 |
| AD10 | 74 |
| AD11 | 75 |
| AD12 | 77 |
| AD13 | 78 |
| AD14 | 79 |
| AD15 | 80 |
| AD16 | 93 |
| AD17 | 94 |
| AD18 | 95 |
| AD19 | 96 |
| AD20 | 97 |
| AD21 | 98 |
| AD22 | 100 |
| AD23 | 101 |
| AD24 | 103 |
| AD25 | 105 |
| AD26 | 106 |
| AD27 | 107 |
| AD28 | 108 |
| AD29 | 110 |
| AD30 | 111 |
| AD31 | 112 |
| AEN | 138 |
| AVDD | 12 |
| BALE | 176 |
| BCLK | 162 |
| BM_REQ# | 127 |
| C/BE0# | 76 |
| C/BE1# | 82 |
| C/BE2# | 91 |
| C/BE3# | 102 |
| CLK14M | 35 |
| CPURST | 50 |
| DACK0# | 192 |
| DACK1# | 157 |
| DACK2# | 169 |
| DACK3# | 154 |
| DACK5# | 194 |
| DACK6# | 198 |
| DACK7# | 205 |
| DDCCLK | 42 |
| DDCDAT | 41 |
| DEVSEL# | 89 |
| DRQ0 | 191 |
| DRQ1 | 158 |
| DRQ2 | 132 |
| DRQ3 | 152 |
| DRQ5 | 199 |
| DRQ6 | 201 |
| DRQ7 | 204 |
| DVDD | 88 |
| DVDD | 121 |
| DVDD | 174 |
| DVSS | 17 |
| DVSS | 90 |
| DVSS | 123 |
| DVSS | 172 |
| DXP/VIN4 | 10 |
| DXN | 11 |
| EXTSMI# | 129 |
| FERR# | 43 |
| FRAME# | 87 |
| GPCS0# | 55 |
| GPI12/VIN0 | 16 |
| GPI13/VIN1 | 15 |
| GPI14/VIN2 | 14 |
| GPI15/VIN3 | 13 |
| GPIO0/PAR | 53 |
| GPIO1/PMCLK | 32 |
| GPIO2/KBCLK | 31 |
| GPIO3/CPU_STOP#/SLP# | 128 |
| GPIO4/ FAN1 | 38 |
| GPIO5/PME0#/ DUAL_ON# | 29 |
| GPIO7/OC0#/PPS | 122 |
| GPIO8/OC1# | 124 |
| GPIO9/THERM#/BTI/SMBALERT# | 40 |
| GPIO10/PME1#/ACPILED | 27 |
| GPIO11/FAN2 | 39 |
| GPIO16/IOCHK# | 1 |
| GPIO17/SIRQ | 54 |
| GPO6/CKE_S/ ACPILED | 28 |
| IGNE# | 46 |
| INIT | 45 |
| INITA# | 56 |
| INITB# | 57 |
| INITC# | 59 |
| INITD# | 60 |
| INTR | 48 |
| IO16# | 180 |
| IOCHRDY | 141 |
| IORC# | 149 |
| IOWC# | 148 |
| IRDY# | 92 |
| IRQ3 | 6 |
| IRQ4 | 7 |
| IRQ5 | 5 |
| IRQ6 | 4 |
| IRQ7 | 3 |
| IRQ9 | 2 |
| IRQ10 | 9 |
| IRQ11 | 8 |
| IRQ14 | 143 |

| SIGNAL NAME | SiS5595 PIN NO. |
|---|---|
| IRQ15 | 146 |
| KBDAT/ IRQ1 | 33 |
| KBVDD | 30 |
| KLOCK#/CPCS1# | 52 |
| LA17 | 190 |
| LA18 | 189 |
| LA19 | 187 |
| LA20 | 186 |
| LA21 | 185 |
| LA22 | 184 |
| LA23 | 183 |
| M16# | 182 |
| MRDC# | 193 |
| MWTC# | 195 |
| NMI | 47 |
| OSC32KHI/IRQ8# | 21 |
| OSC32KHO/ RTCCS# | 20 |
| OVDD | 62 |
| OVDD | 81 |
| OVDD | 104 |
| OVDD | 168 |
| OVSS | 64 |
| OVSS | 73 |
| OVSS | 85 |
| OVSS | 99 |
| OVSS | 109 |
| OVSS | 150 |
| OVSS | 188 |
| PCIRST# | 126 |
| PCLK | 58 |
| PDMAGNT0# | 130 |
| PDMAREQ0# | 151 |
| PHLDA# | 125 |
| PHOLD# | 113 |
| PMDAT/ IRQ12 | 34 |
| PS_ON#/ RTCALE | 25 |
| PSRSTB# | 23 |
| RFH# | 159 |
| RING | 24 |
| PWRBT# | 26 |
| PWRGD | 18 |

| SIGNAL NAME | SiS5595 PIN NO. |
|---|---|
| ROMKBCS# | 36 |
| RTCVDD | 22 |
| RTCVSS | 19 |
| SA0 | 179 |
| SA1 | 178 |
| SA2 | 177 |
| SA3 | 175 |
| SA4 | 170 |
| SA5 | 171 |
| SA6 | 167 |
| SA7 | 166 |
| SA8 | 165 |
| SA9 | 164 |
| SA10 | 163 |
| SA11 | 161 |
| SA12 | 160 |
| SA13 | 156 |
| SA14 | 155 |
| SA15 | 153 |
| SA16 | 147 |
| SBHE# | 181 |
| SD0 | 142 |
| SD1 | 137 |
| SD2 | 139 |
| SD3 | 134 |
| SD4 | 135 |
| SD5 | 140 |
| SD6 | 131 |
| SD7 | 133 |
| SD8 | 196 |
| SD9 | 197 |
| SD10 | 200 |
| SD11 | 202 |
| SD12 | 203 |
| SD13 | 206 |
| SD14 | 207 |
| SD15 | 208 |
| SERR# | 83 |
| SMI# | 49 |
| SMRDC# | 145 |
| SMWTC# | 144 |
| SPKR | 37 |
| STOP# | 84 |

| SIGNAL NAME | SiS5595 PIN NO. |
|---|---|
| STPCLK# | 44 |
| TC | 173 |
| TRDY# | 86 |
| UCLK48M | 120 |
| USBVDD | 119 |
| USBVSS | 114 |
| UV0- | 116 |
| UV0+ | 115 |
| UV1- | 118 |
| UV1+ | 117 |
| ZWS# | 136 |

# 3 FUNCTIONAL DESCRIPTION

SiS5595 is a highly integrated system I/O that constitutes a high performance, rich featured, yet glueless solution for both Pentium and Pentium II systems.

The SiS5595 PCI system I/O integrates the PCI-to-ISA bridge with the DDMA and PC/PCI DMA, Serial IRQ capability, the ACPI/Legacy PMU, the Data Acquisition Interface, the Universal Serial Bus host/hub interface, and the ISA bus interface, which contains the ISA bus controller, the DMA controllers, the interrupt controllers, and the Timers. It also integrates the Keyboard controller, and the Real Time Clock (RTC). The built-in USB controller, which is fully compliant to OHCI (Open Host Controller Interface), provides two USB ports capable of running full/low speed USB devices. The Data Acquisition Interface offers the ability of monitoring and reporting the environmental condition of the PC. It could monitor 5 positive analog voltage inputs, 2 Fan speed inputs, and one external temperature inputs. It also integrates the automatic power control logic to control the power ON/OFF for ATX power supply. In addition, SiS5595 also integrates the thermal detection and frequency ratio control logic for Pentium II CPU.

## 3.1 PCI BUS INTERFACE

### 3.1.1 PCI TO ISA BUS BRIDGE

As a PCI slave device, the PCI-to-ISA Bridge responds to both I/O and memory transfers. It always target-terminates after the first data phase for any bursting cycle.

The PCI-to-ISA Bridge 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 agents. In reality, it only subtractively responds to low 64K I/O or low 16M memory accesses. It also positively decodes BIOS memory space by asserting DEVSEL# signal on the medium timing. It is optional to do positive or subtractive decode on I/O addresses for some internal registers.

As a PCI master device, the PCI master bridge on behalf of DMA devices or ISA Master devices drives the AD bus, C/BE[3:0]# and PAR signals. When MEMR# or MEMW# is asserted, the PCI-to-ISA bridge 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. It always activates 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 forward to PCI.

Delayed transaction is a mechanism used when the target, like PCI-to-ISA bridge in SiS5595 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.

- **DMA/ISA Master Cycles**

ISA devices or DMA controller embedded in the PCI-to-ISA Bridge of SiS5595 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, the actions of an ISA master cycle will be described first, 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.

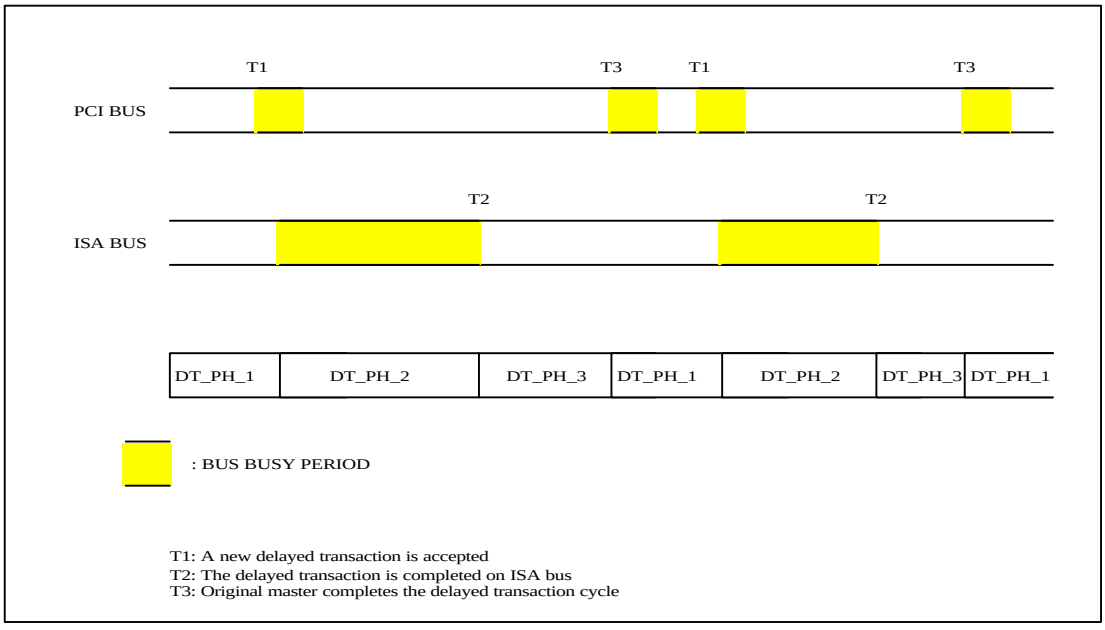


**Figure 3.1-1 PCI ISA Delay Transaction**

**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 DRQs to DMA controller embedded in the PCI-to-ISA bridge. The PCI-to-ISA bridge will then generate PHOLD# to system arbiter to request for PCI bus. The PHOLD# will be asserted as long as DRQ 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. When ISA master receives DACK#, it can then start its cycles transferring data to or from PCI (ISA) bus. When ISA master finishes its cycles, it de-asserts DRQ 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.

- **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, i.e., 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 delayed 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.

### 3.1.2 DISTRIBUTED DMA (DDMA)

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 SiS5595 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 SiS5595 will take control of this cycle by driving DEVSEL# active, asserting PHOLD# and issuing a PCI retry to terminate this cycle.

2) When granted the PCI bus, the SiS5595 will run up to 4 PCI I/O read/write cycles. 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 SiS5595 will set an internal flag indicating the completion and will de-assert PHOLD# and wait for the retried PCI I/O read/write from the CPU bridge.

The PCI I/O read/write will be retried. If it was a read, the SiS5595 will return the data. If it was a write, the SiS5595 will simply terminate the cycle. Then the SiS5595 will reset the internal flag.

### 3.1.3 PC/PCI DMA

SiS5595B supports one PC/PCI PDMAREQ0#/PDMAGNT0# pairs. PCI devices plugged in the PCI slot may initiate PC/PCI DMA transfer cycles through the PDMAREQ0#/ PDMAGNT0# pair. For DMA operation, three types of transfer cycles are supported: Memory-to-I/O, I/O-to-Memory and Verify. SiS5595B also supports ISA master operation through PC/PCI DMA channels, on which a PCI device may request the PCI bus through the PDMAREQ0#/PDMAGNT0# pair. Each of the seven DMA channels can be individually programmed to be in Legacy, DDMA or PC/PCI DMA mode. Care must be taken to ensure only one of the three operation modes is enabled for a particular DMA channel. The legacy 8237 compatible registers will be used to control the operation of PC/PCI DMA, for software backward compatibility.

### 3.1.4 SERIAL IRQ (SIRQ)

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 a 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.

#### 3.1.4.1 Timing Diagrams For Serial IRQ Cycle

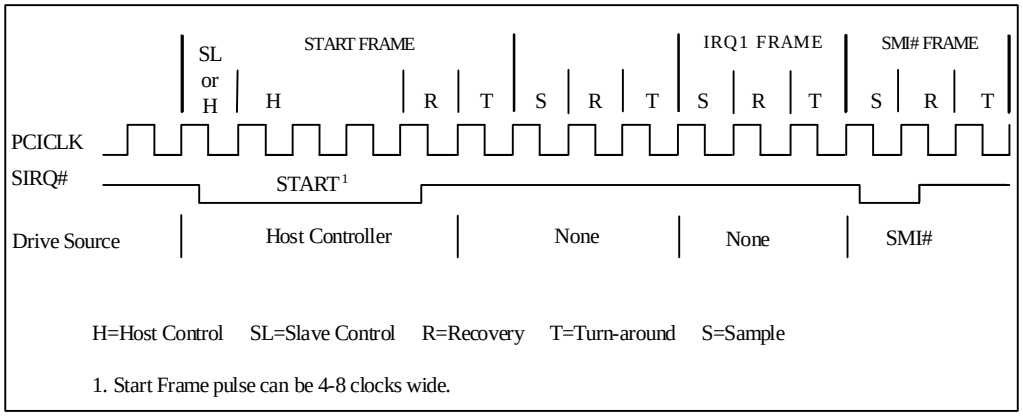


**Figure 3.1-2 Start Frame Timing with Source Sampled a Low Pulse on SMI#**

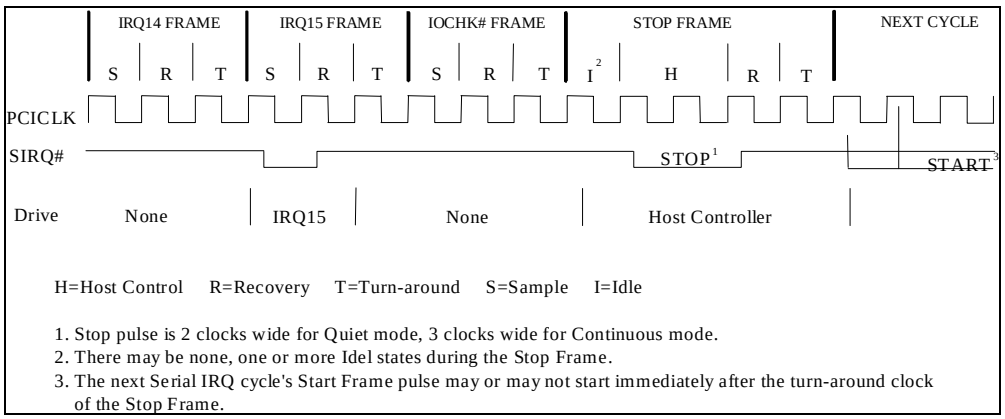


**Figure 3.1-3 Stop Frame Timing with Host using 17 SIRQ# Sampling Period**

### 3.1.4.2 Serial IRQ Cycle Control

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

1) **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 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 SiS5595 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, SiS5595 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 SiS5595 unless the

SIRQ# is already in a serial IRQ cycle and the IRQ/Data transition can be delivered in the serial IRQ cycle.

2) **Continuous (Idle) Mode:** Only the SiS5595 can initiate a Start Frame to update SIRQ# line information. All other serial IRQ agents become passive and may not initiate a Start Frame. SIRQ# will be driven low for four to eight clocks by the SiS5595. There are two functions in this mode. It can be used to stop or idle the SIRQ# or the SiS5595 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 occur during the Stop Frame. Upon reset, the Serial IRQ bus is defaulted to continuous mode, therefore only the SiS5595 can initiate the first Start Frame. Slave must continuously sample the Stop Frames pulse width to determine the next serial IRQ cycle's mode.

#### 3.1.4.3 IRQ/Data Frame

Once a Start Frame has been initiated, all serial IRQ devices must detect the rising edge 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).

**Table 3.1-1 IRQs Mapping in Serial IRQ Periods**

| SERIAL IRQ SAMPLING PERIODS | | |
|---|---|---|
| **IRQ/Data Frame** | **Signal Sampled** | **# of clocks past Start** |
| 1 | Reserved | 2 |
| 2 | IRQ1 | 5 |
| 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 |

| SERIAL IRQ SAMPLING PERIODS | | |
|---|---|---|
| **IRQ/Data Frame** | **Signal Sampled** | **# of clocks past Start** |
| 14 | IRQ13 | 41 |
| 15 | IRQ14 | 44 |
| 16 | IRQ15 | 47 |
| 17 | IOCHCK# | 50 |
| 18 | INTA# | 53 |
| 19 | INTB# | 56 |
| 20 | INTC# | 59 |
| 21 | INTD# | 62 |
| 32:22 | Unassigned | 95 |

At the end of each Sample phase, the SiS5595 will sample the state of the SIRQ# line and replicate the status on the original IRQ/Data line at the input to the 8259s Interrupt Controller.

#### 3.1.4.4 Stop Cycle Control

Once all IRQ/Data frames have completed, the SiS5595 will terminate SIRQ# activity by driving Stop cycle. Only the SiS5595 can initiate the stop frame. A Stop Frame is indicated by the SiS5595 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 SiS5595 may initiate a Start frame in the third clock or more after the rising edge of the Stop frame pulse.

## 3.2 ACPI /LEGACY PMU

### 3.2.1 ADVANCED CONFIGURATION AND POWER INTERFACE (ACPI)

Advanced Configuration and Power Interface (ACPI) is PC 97/98 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 is set.

The ACPI architecture in SiS5595 consists of the Fixed Features logic, Generic Features logic, Legacy Features logic, and the configuration registers to ensure fluent communication with the ACPI-compliant OS. The Fixed Features meet ACPI SPEC 1.0 requirement. The SCI/SMI# generating logic in ACPI senses external environmental changes or requests and interrupts, the OS to take some action. The wakeup logic will sequence the system from the S1/S2 to G0/S0 working state. Besides, sequencing the system from S3/S5 to G0/S0 can only be achieved through the power up events processed in the RTC/APC module.

#### 3.2.1.1 Fixed Features

From ACPI specification, SiS5595 supports the four global system states (G0-G3), and the traditional Legacy system state as shown in the Figure 3.2-1 Global System State Diagram. 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 re-routability 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 I/O write command to the SMI# Command Port (35h).

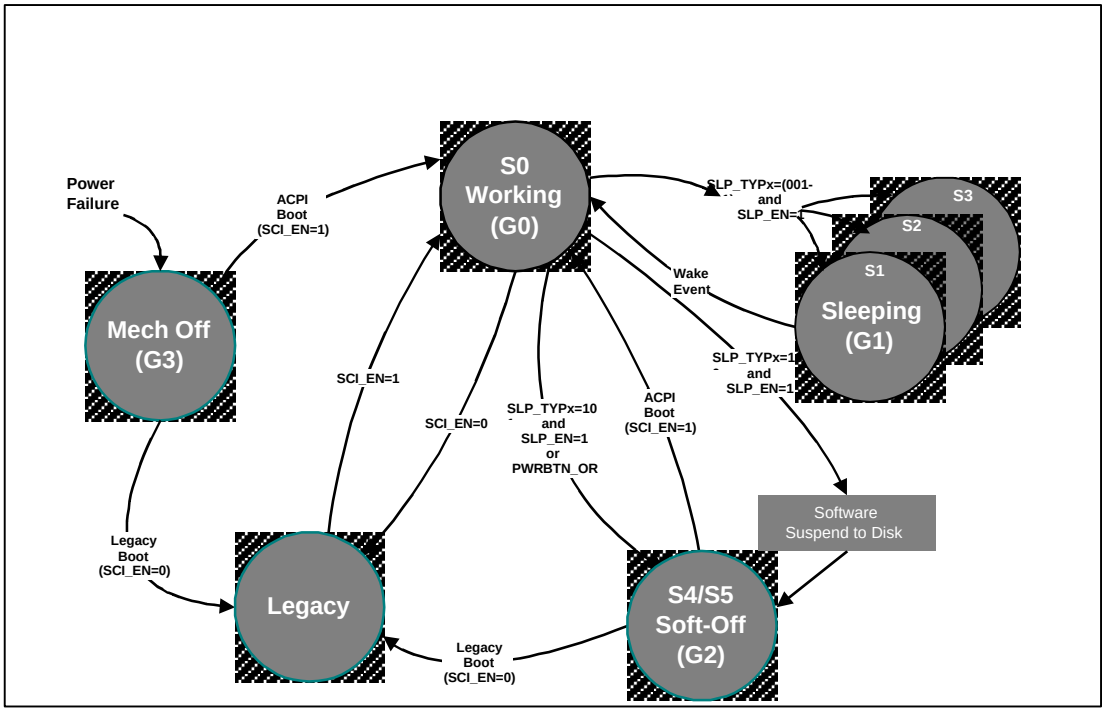


**Figure 3.2-1 Global System State Diagram**

- **Sleeping State**

ACPI state 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 reality, the system G1 State consists of S1, S2, S3, and S4 State hierarchically in the sense of sleeping states. The O.S. can initiate the sleeping state transition by programming the SLP_TYPx field with the appropriate value and then setting the SLP_ENx bit high. Table 3.2-1 Sleeping State and Clock State summarizes the recommended arrangement of CLK; Table 3.2-2 Sleeping State and Power State describes the recommended arrangement of the power state for the major components at the three sleeping states. In S2 state, the CLK to the CPU can be further stopped in SiS cored based Pentium system. However, all the clocks in S2 State must keep running in SiS cored based Pentium II system due to the nature of the clock scheme used. It is optional to put memory subsystem into the self-refresh

mode while the system enters S2 State. However, it is a MUST for a system to place the memory subsystem into self-refresh mode when it enters S3 State.

**Table 3.2-1 Sleeping State and Clock State**

| SYSTEM STATE | CLOCK STATE |
|---|---|
| S1 | Assert STPCLK# ; |
| S2 | Assert STPCLK# ;<br>Use GPIO3 to stop CPUCLK ;<br>Use GPIO3 as SLP# to put Pentium II into SLEEP state |
| S3 | Disable all the clocks except that for the wake-up logic (*1) |

(*1) Only 5595/32KHz is alive. Wake-up logic in SiS5595 chipset family normally relates to RTC module.

**Table 3.2-2 Sleeping State and Power State**

| SYSTEM STATE | POWER STATE |
|---|---|
| S1 | Keep all the power supplied |
| S2 | Can turn off the power for ISA devices |
| S3 | Can turn off all the power except for the SDRAM and the wake-up logic |
| S4/S5 | Turn off all the power except for the wake-up logic. |

The following statement details the running steps of S1, S2 and S3 State:

- **S1 state**

Entering S1 state is achieved by setting SLP_EN bit HIGH and SLP_TPY=001. All system power is still alive in this state. A set of wake-up Events can be enabled, before entering S1, to wake up the system back to G0 State.

- **S2 state**

The North Bridge in the following context may stand for SiS530 used in Pentium system or SiS5600/SiS620 used in Pentium II system, respectively.

1) Disable PCI system Arbiter & AGP arbiter

2) Program into S2 mode with the consequence that STPCLK# is asserted.

3) North Bridge keeps processing whatever is requested until CPU generates a stop grant (stpgnt) cycle. The North Bridge then forwards the stpgnt cycle via PCI special cycle to the PCI bus. While intercepting the stpgnt special cycle on the PCI bus, 5595 will optionally first mask the ISA commands (IORC#, IOWC#, MRDC#, and MWTC#), float all the ISA signals, and then assert GPIO3 in 32 PCICLKs later. GPIO3 can be used to disable the CPUCLK in

Pentium system. However, it is recommended that GPIO3 is connected to SLP# of Pentium II processor to place it into very low power SLEEP STATE in which the processor maintains its context, and PLL, and has stopped all internal clocks.

4) When the system is awaken, the WAK_STS is set, re-enable the ISA bus, unmask the ISA commands, and finally de-assert the STPCLK# in about 4ms. A period of 4ms is reserved to allow the CPU/PLL stabilization. A set of wake up events illustrated in Figure 3.2-2 Wake up events in S1 / S2 are supported to wake up the system in the S2 state.

Enabling "Mask the ISA command" and "Float the ISA signals" upon waking up from S2 State can be achieved by setting bit 3 of ACPI Reg. 13h. No ISA cycles should be initiated before RSTDRV is de-asserted. Setting bit 2 of ACPI reg. 13h can select the GPIO3 multi-function pin to function as CPU_STOP#/SLP#.

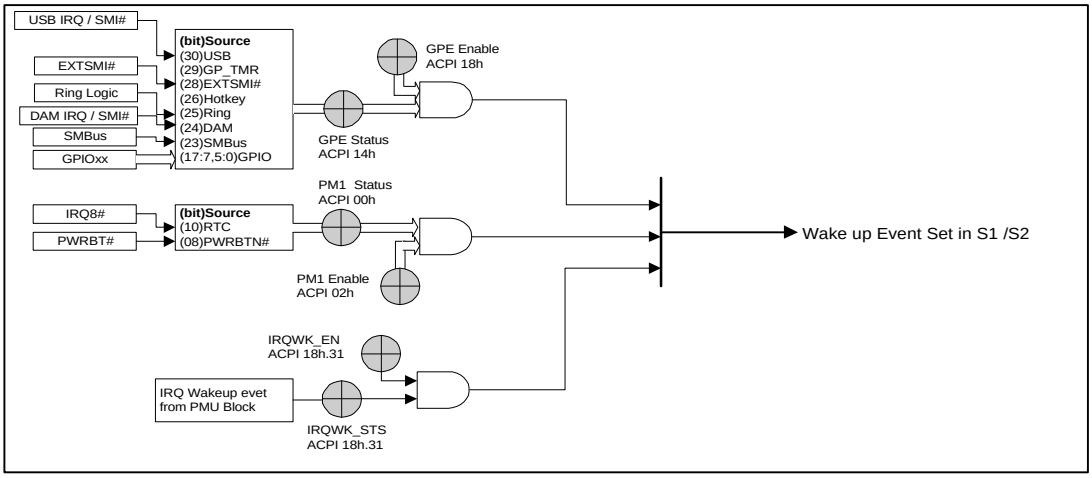


**Figure 3.2-2 Wake up events in S1 / S2**

Note : DAM means Data Acquisition Module.

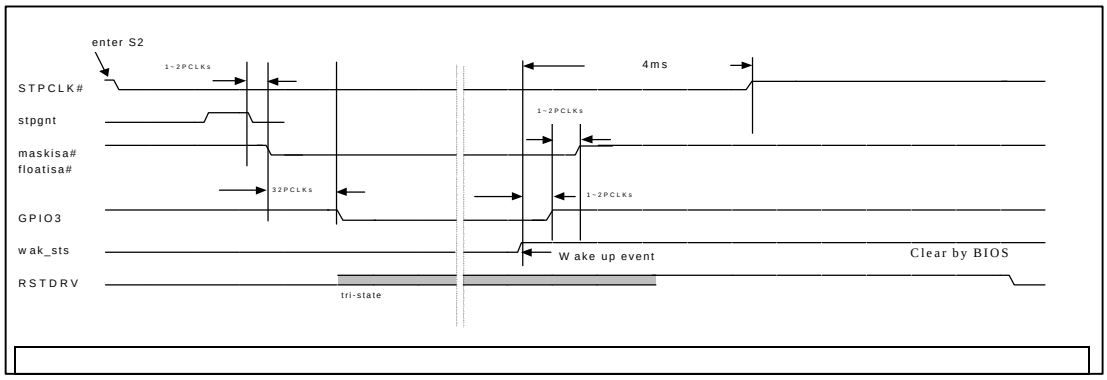


**Figure 3.2-3 5595's Timing Diagram in S2 State**

Note : stpgnt, maskisa, floatisa, and wak_sts are internal signals.

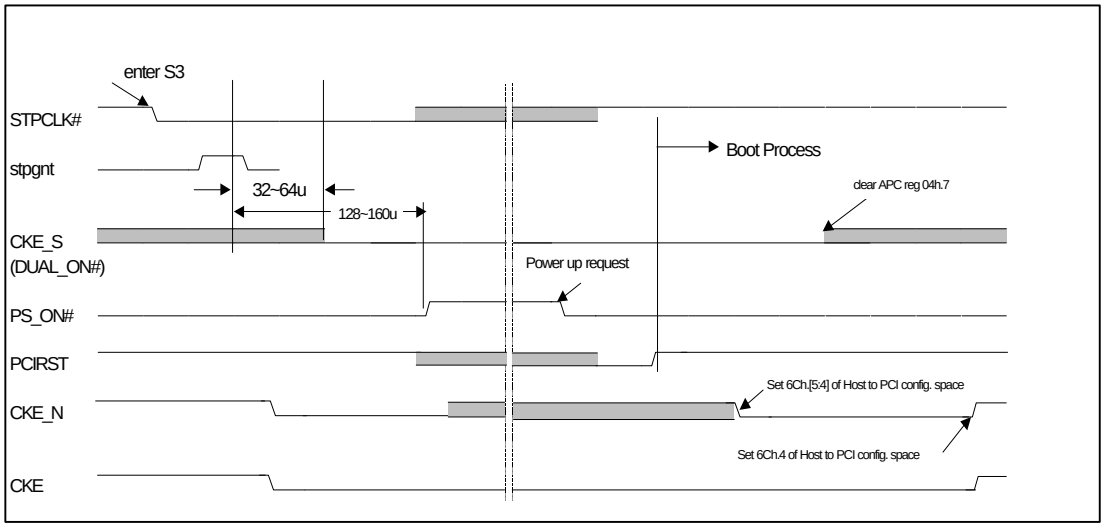


**Figure 3.2-4 North Bridge/5595's Timing Diagram in S3 State**

Note : “stpgnt” is an internal signal. CKE_N and CKE are driven by SiS North Bridge.

- **S3 state (Suspend To RAM)**

Disable PCI system Arbiter & AGP arbiter.

Similar to the S2 state, STPCLK# is asserted as a result of transiting the system into S3 state by writing 011(binary digit) to SLP_TYP field of register 04h in the ACPI IO space.

Having intercepted the Stop_Grant special cycle on the PCI bus, the SiS5595 will enable and drive CKE_S low in 32us to 64 us, and then negate PS_ON# in another 128us to 160us later.

Figure 3.2-4 North Bridge/5595's Timing Diagram in S3 State is the timing diagram showing the sequence happening on the SiS5595 while entering/exiting S3 State.

To provide a fast wake-up latency, it is highly recommended to place the system into the Suspend to RAM (STR) state in S3 state. Placing the system into STR is normally achieved through keeping the system image in the main memory (not in the hard disk). Except the memory subsystem, and the wake-up logic, the power to the rest of the M/B components is removed. Moreover, the memory subsystem is put into the low power state. For today's mainstream memory like EDO or SDRAM, placing it into low power state can be performed by programming it for self-refresh mode.

Per the SDRAM specification, it is required to keep CKE low as long as it is maintained in self-refresh mode. There are two classes of CKE signals generated in SiS chipset. One is CKE_N from the North Bridge, and the other is CKE_S from SiS5595. For a system which does not support Suspend to RAM (STR), CKE can be simply connected to a pull high resistor. To support Suspend to RAM function, an external LVT 245 is employed to generate 6 CKE signal lines to support 3 DIMM configuration. Please refer to the SiS5600/ SiS620/SiS5595 design guide or the associated application circuit for more detail. The

following details the protocol of CKE_N, and CKE_S while entering/exiting the self-refresh mode.

When Battery is first plugged in, CKE_S is put into HI_Z State. In fact, CKE line is driven high by external pull up resistor at this moment. Note that the pull-up resistor should be connected to the power supplying the SDRAM. CKE_N is also put into HI-Z State upon power up every time. Upon power up or wake-up, the BIOS should read the STR_STS bit stored in RTC APC 06h bit 5 to determine if it needs to alter the CKE_N, and CKE_S state.

If it is, the BIOS should program the North Bridge to drive CKE_N low, put the CKE_S into high impedance state, and then drive the CKE_N high to exit SDRAM from self-refresh mode. If STR_STS bit is not set, leave CKE_N and CKE_S in the high impedance state, and have the external pull-up resistor driving CKE high.

On the other hand, CKE_N and CKE_S perform in the following way while entering into S3 State.

1) The North Bridge enables CKE_N low after the completion of emptying the write buffer.

2) The SiS5595 drives CKE_S low in 32 us later after having intercepted the Stop_Grant special cycle on the PCI bus.

3) CKE_N stops driving CKE line in 128us to 160us later as a result of the negation of PS_ON# to turn off the power. Starting from this point, SDRAM and the wake-up logic is powered by the AUX3V (or AUX5V), or Vdual from Power Supply '98. The CKE_S keeps the CKE line low from this moment. CKE_S can also be connected to DUAL_ON# of the power supply '98.

The following example to represent routine recommended for regulating the CKE_S and CKE_N in the S3 State for SiS5595 working with SiS530 or SiS5600/SiS620.

***{Power up or resuming sequence for SiS530 or SiS5600/SiS620 core based system}***

*...*

***If (STR_STS)***

***/* Drive CKE_N low with the following two steps */***

*Set bit 4 of reg. 6Ch in the Host to PCI config. space ;*

*Set bit 5 of reg. 6Ch in the Host to PCI config. space ;*

***/* Place CKE_S into high impedance state */***

*Clear bit 7 of APC Reg 04h in the SiS5595;*

***/* Drive CKE_N high */***

*Clear bit 4 of reg. 6Ch in the Host to PCI configuration space ;*

***If (!STR_STS)***

***If (the system would like to support STR)***

***/* Drive CKE_N high */***

*Set bit 5 of reg. 6Ch in the Host to PCI config. Space ;*

***Else Leave CKE_N, and CKE_S as they are ;***

While in the G1 State, a set of Wake_Up Events can be enabled to transit the system state back to G0. Please refer to " **RTC APC "** illustrating the supported wake up events in S3 State.

- **G2---S4/S5(Suspend To Disk/Soft Off)**

The G2 soft-off state is an O.S. 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 de-assert PS_ON# eventually from hardware point of view.

In the G2 State, only the RTC power is alive. While in the G2 state, the SiS5595 could sense the following seven power up events to transit the system to the Legacy system state by asserting the PS_ON#. They are RTC Alarm On event, Power Button Up (PWRBT#) event, Ring Up event, PME0# event, PME1# event, Hotkey Match event, and Password Match event. Please see the APC portion of the RTC module for more details.

- **Processor Power State**

- **STPCLK# Throttling (C0)**

SiS5595 supports the four power states in the G0/S0 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 maxima performance. This can be achieved by programming the Throttling Duty Cycle Control field (ACPI: 0Ch[3:1]) with desired value, and setting Throttling Function Enable bit (ACPI: 0Ch[4]) to HIGH.

- **CPU Power State Level 1 (C1)**

The C1 State is supported through the HLT instruction. For instance, the execution of a HALT instruction will cause 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. CPU would not recognize AHOLD, BOFF#, and EADS# for cache invalidation or write-back. 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.

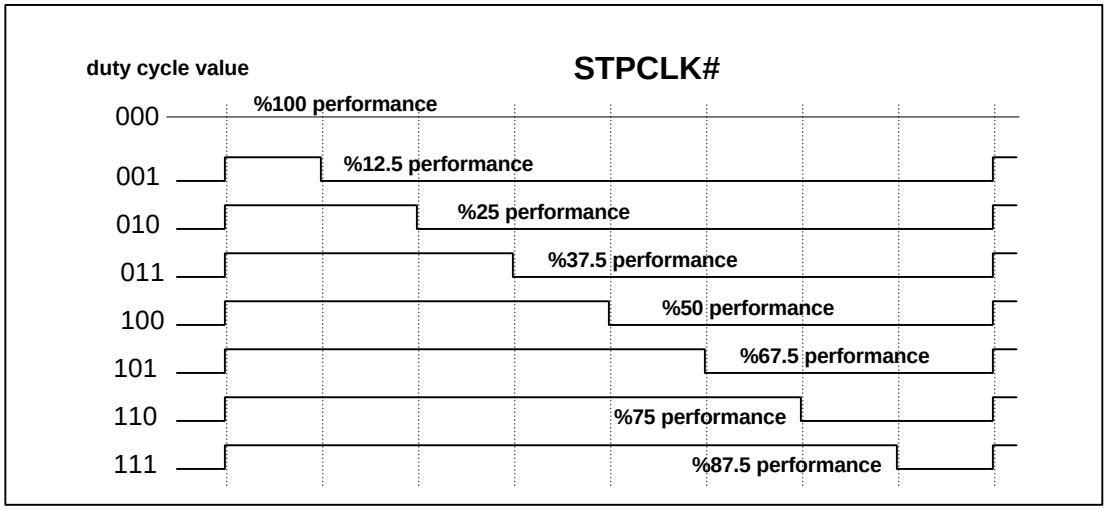


**Figure 3.2-5 STPCLK# Throttling & Performance**

- **CPU Power State Level 2(C2)**

In the C2 power state, SiS5595 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 Register 10h) as it is defined in the ACPI specification. Exiting C2 is effective when any interrupt is asserted. Register 68h, and 69h in the PMU configuration space defines which interrupt requests among the IRQ15-3, 1-0, and NMI are enabled to exit C2. Beside, the PMU configuration space register 50h bit 24 should be enabled.

- **CPU Power State Level 3(C3)**

As a more rigid or flexible alternative to the handling of bus master in the CPU low power state, SiS5595 supports the C3 power state by also keeping STPCLK# low, which can be entered by reading P_LVL3 register (ACPI Register 11h). Optionally, GPIO3 used as SLP# can be asserted to place the Pentium II processor into SLEEP State. The main difference between C3 and C2 state is that the bus masters are prevented 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 Register 12h[0]) to 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 Register 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 de-asserting STPCLK# while it is not needed for C2 state. Any interrupt will also bring the processor out of C3 power state. The BM_STS is set whenever any bus master request (REQ#) is asserted. Figure 3.2-6 Processor Power State Diagram illustrates the processor power state diagrams supported by SiS5595.

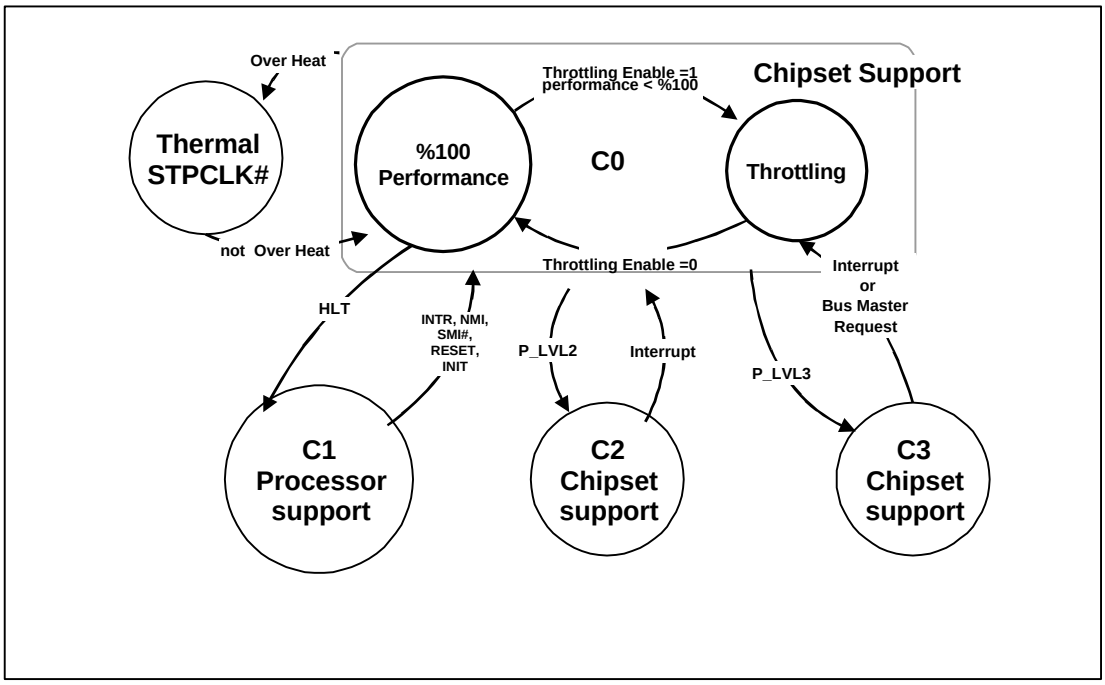


**Figure 3.2-6 Processor Power State Diagram**

- **System Timer Event (IRQ0)**

In Legacy mode, SiS5595 supports special IRQ0 function. When in C2 or C3, the IRQ0 (system timer) can de-assert STPCLK# (&SLP#) 125us and does not exit C2 or C3. This function is enabled by setting ACPI Register 13h bit 0, and you should disable IRQ0 wake up event in PMU configuration space register 68-69h. Legacy PMU handler can use 125us to update system time or do something else. After 125us, STPCLK# is asserted again.

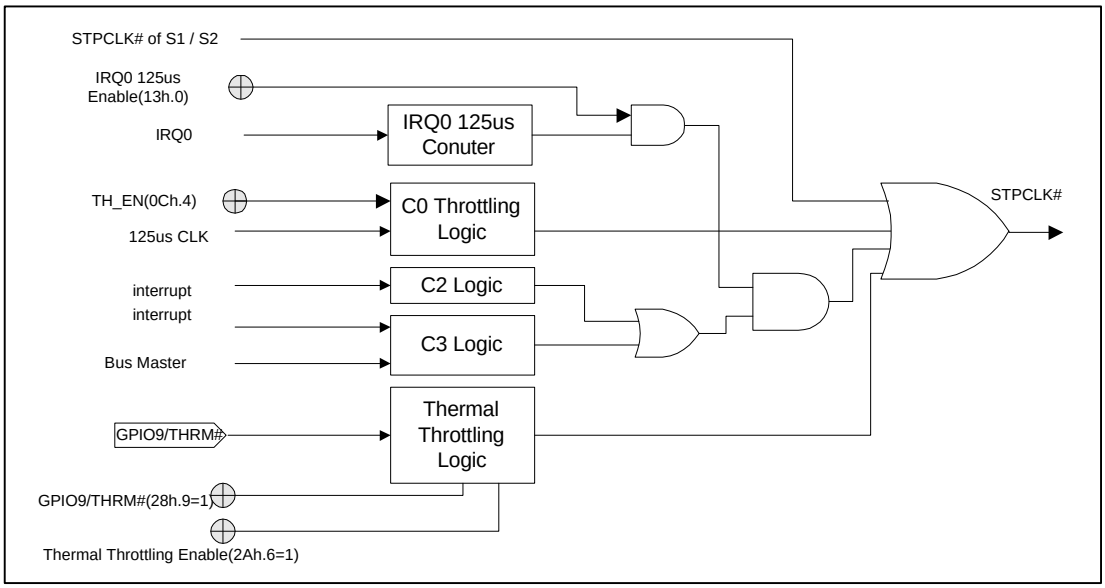


**Figure 3.2-7 STPCLK# Source**

- **Thermal STPCLK# Throttling**

SiS5595 can support thermal detection by selecting GPIO9 as the THERM# pin. A 1ms de-bouncer is used to sense the status of THERM#. When the logic of the THERM# matches the programming active level, the STPCLK# is throttled if the Thermal Throttling Enable (bit 6 of ACPI register 2Ah) is set. The throttling will be stopped if the THERM# 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 Throttling Function Enable bit (ACPI IO 0Ch bit 4) for throttling the STPCLK# in response to the THERM# request. If THERM# is asserted, the system will enter the throttling mode directly (whose duty cycle is defined by ACPI I/O 2Ah, bit 4~2) or generate an SCI/SMI# by the thermal throttling function/GPIO9 selection bit (ACPI: 28h[9]). Please refer to Figure 3.2-8 Thermal Detection Logic.

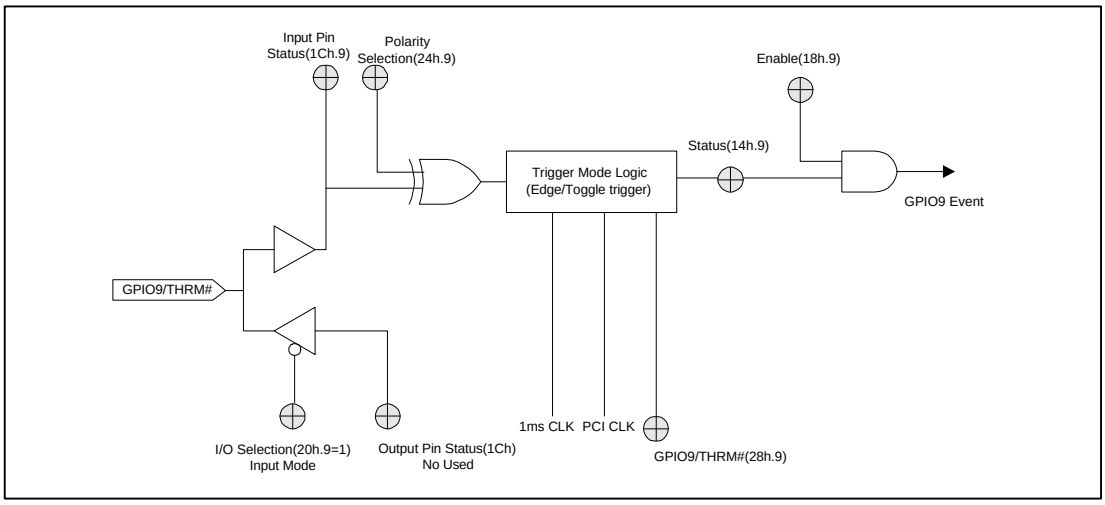


**Figure 3.2-8 Thermal Detection Logic**

- **Power Button**

This button is a user interface control instead of the traditional power supplier switch. It can be used to cycle the system between the G0 and G1 state, as one of the power management events. Besides, the power button provides user a mechanism to force the system to enter the G2 State (Soft-Off) when the system has hung. This is called as power button over-ride.

Normally, the SiS5595 generates a power button event in the form of SCI, or SMI# in the G0 working state while detecting the Press-and-Release sequence on the PWRBT# pin. However, by setting bit 4 of ACPI register 13 to 1, the power button event can be generated simply upon that the button is pressed. Upon the instant that power button is pressed to the instant that the power button override event is recognized, the specific routine can be invoked, and executed (due to SMI#, or SCI) to do any housekeeping before the power is removed.

A 1ms de-bouncer associated with the power button is used to recognize and respond to the active low logic presented on the pin. If the PWRBT# is pressed for more than 4 seconds, the SiS5595 will turn off the system power by de-asserting the PS_ON#.

If the PWRBT# is released within 4 seconds, 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. There is a power button over-ride enable bit in ACPI Register 13h bit 5, which is enabled by default. Although the ACPI SPEC 1.0 no longer requires the support of this power button over-ride enable bit, SiS5595 keeps the bit in other location to allow the application software to disable the function just in case.

- **Power Management Timer**

The SiS5595 supports a 24-bit power management timer, based on a 3.579545MHz clock, which provides an accurate time value used by system software to measure and profile system idleness (along with other tasks) while the system is in the working (G0) state. To allow software to extend the number of bits in the timer, the power management timer

generates an interrupt when the last bit of the timer changes (from 0 to 1 or 1 to 0). The PM Timer can be accessed directly by the ACPI driver or software. The TMR_STS status bit is set any time the last bit of the timer bit 23 goes from HIGH to LOW or LOW to HIGH. If the TMR_EN bit is set, the TMR_STS being set will generate an ACPI event (SCI/SMI#). The timer is reset when system wake-up from sleeping states (S1, S2) or hardware reset.

- **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 extend the alarm bytes, SiS5595 supports both the Day of the Month Alarm, and Month Alarm function. The Day of the Month Alarm byte, and Month Alarm byte are located in the 7Eh, and 7Fh of the Standard Bank of the RTC CMOS RAM, respectively. The RTC_STS bit will be set once IRQ8# is asserted from the internal RTC module. The application software can set any specific time to generate an SCI or SMI# if the system is in the working state, or a wake-up event if the system is in the sleeping state S1, or S2.

- **SCI /SMI# Generation**

When SCI_EN=1, most of the ACPI events can generate SCI; when SCI_EN=0 (Legacy mode) they can generate SMI#. Normally, ACPI events can generate SCI or SMI# only when the system is in the G0 working state. However, an option is reserved to generate SMI# in response to the ACPI events while the system is in one of the sleeping states. Figure 3.2-9 SCI / SMI# Events Overview summarizes all the supported ACPI events in SiS5595 to generate SCI/SMI#. System designer should take care that some events only generate SMI# or wake up system. For example, Hotkey can only generate SCI/SMI#, IRQWK can only generate wake-up events in S1 or S2 state, and SIRQ can only generate SMI#.

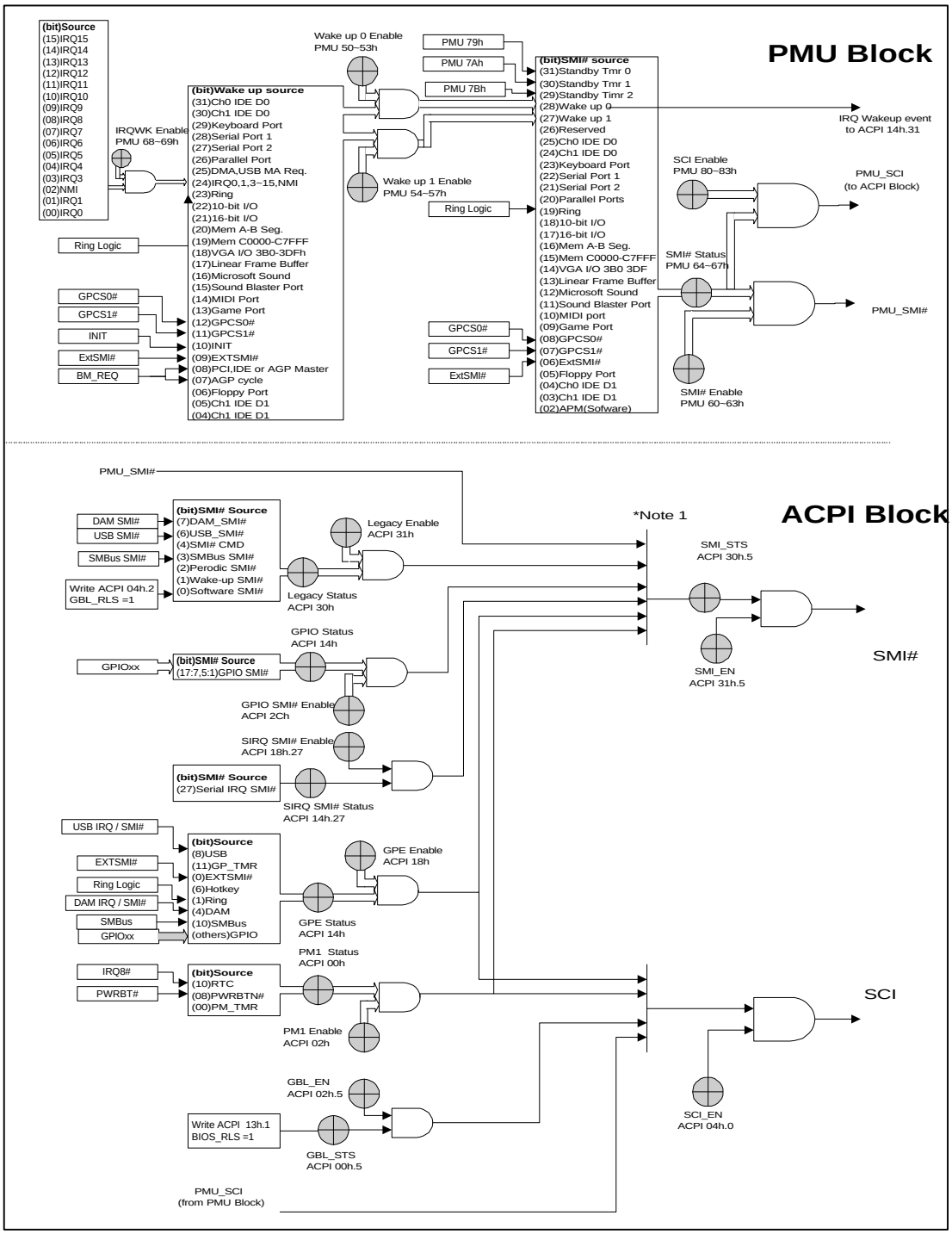


**Figure 3.2-9 SCI / SMI# Events Overview**

Note 1: The “line” means OR logic.

### 3.2.1.2 Generic Features

Generic Features in SiS5595 offers a variety of additional events such as USB event, GP_TMR event, EXT_SMI# event, Hotkey event, Ring event, Data Acquisition Module event, and GPIO events. These events are all categorized as ACPI events. Hence, they can either generate SCI or SMI# in the working state or serve as the wake-up events in the sleeping state.

- **Wake up IRQ (IRQ_WK) Event**

IRQ_WK event is one of GPE Bank event. A status bit located in ACPI register 14h bit 31 is set when IRQ_WK is asserted if its enable bit locating in ACPI Reg.18h bit 31 is set. Note that IRQ_WK only generates wake-up event in S1 or S2 State. It cannot generate SCI or SMI#. To correctly function, additional registers are to be programmed:

1) Specify IRQ Enable in PMU Configuration Space 68~69h,
2) Mask other wake-up source in PMU Configuration in PMU Configuration Space 50~53h**.**

Besides, the IRQ_WK (internal signal) is also used to exit CPU Power State from C2 or C3.

- **General Purpose Timer (GP_TMR) Event**

General-purpose timer is an 8 bits down counter. Its resolution is 1us or 1 min. General-purpose timer can be programmed as Suspend timer or BIOS Timer (ACPI: 1Ch[7]). Loading values into the counter values initiates the counting immediately. If there is not any reload events (PMU Configuration Register: 40h-43h) detected during counting down period, the timer will expire with the result of generating a power management event (SCI, SMI#, or wake up event). Suspend timer is functionally similar to System Standby Timer. However, expiration of Suspend Timer can assert SCI or SMI# but expiration of System Standby Timer can only assert SMI#.

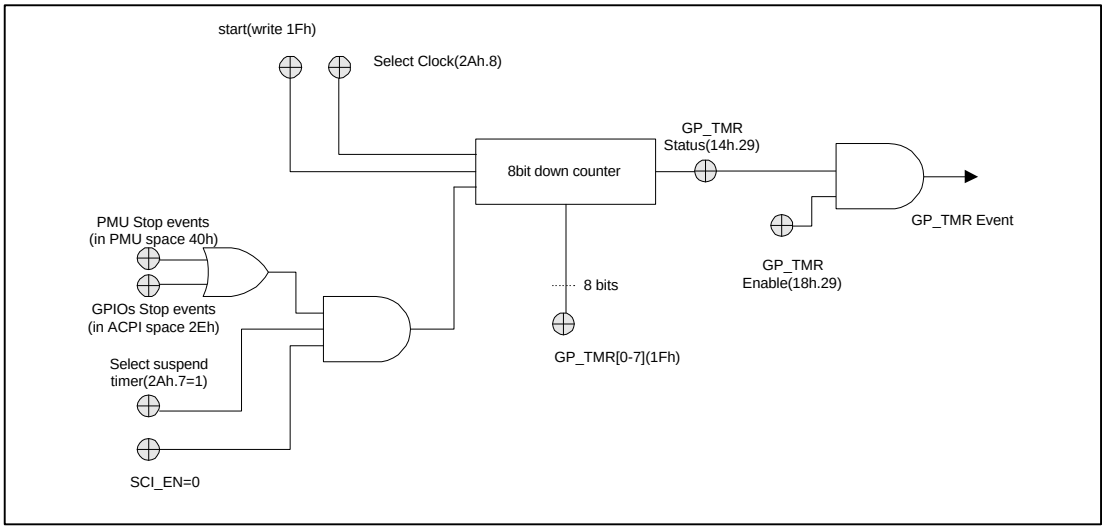


**Figure 3.2-10 General Purpose Timer Logic**

- **HotKey Event**

When internal keyboard controller is enabled, the HOTKY_STS (ACPI: 14h[6]) will be set if the"CTRL+ALT+Backspace" is recognized. Then SCI/SMI# is generated if HOTKY_EN (ACPI: 18h[6]) is set.

- **Ring Event**

There are two de-bouncers 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, depending on the value of ACPI Register 2Ah bit 5.

For ring detection, the RI_STS bit (ACPI 14h bit 1) will be set if the ring signal is sensed asserted. If the RI_EN bit (ACPI 18h bit 1) is also enabled, the power management event will be generated.

Ring function can only used with internal RTC.

- **GPIOx Events**

SiS5595 provides eighteen pins to support general purpose I/O function. GPO6 is output only. GPI[12:15] are input only. The rest (GPIO[5:0], GPIO[11:7], GPIO[17:16]) are bi-directional. Beside, GPIO5, GPO6 GPIO10 are in RTC power plant, so that the control register is in RTC APC space. The input/output attribute of these pins can be programmed through setting or resetting the corresponding bits of the **GPE I/O Selection** register in ACPI 20h. By default, all the GPIO pins are input. While in the input mode, the active logic level can be programmed through **GPE Polarity Selection** registers in ACPI 24h. The default active level is low. Some GPIOs have trigger mode selection, including GPIO1, GPIO11~15. User can set rising/falling edge trigger by **GPE Control** in ACPI space 24h and 2Ah. Please refer Figure 3.2-11 GPIO Logic.

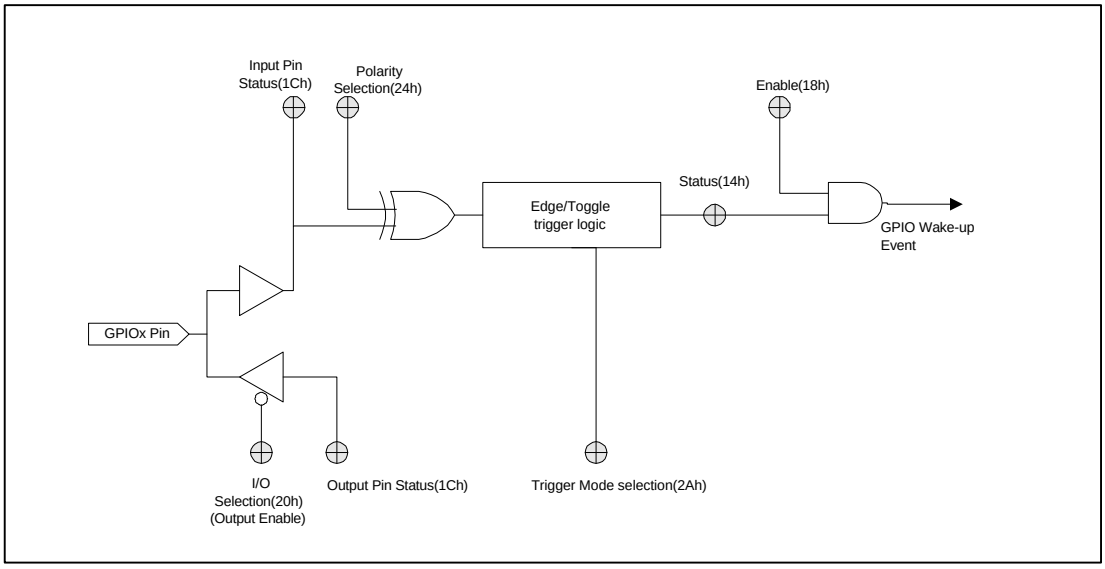


**Figure 3.2-11 GPIO Logic**

When the corresponding enable and status bits are set, SCI or SMI# or Wake-up event will be generated. 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 Pin Status register (in ACPI 1Ch). 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 Pin Status registers to control the peripheral device power, for instance.

- **GPIOx Logic**

The SiS5595 supports a variety of General-Purpose Input/ Output pins that are also MUXed with other signals as they are shown below with a couple of properties (Table 3.2-3):

**Table 3.2-3 GPIOx Relational Data Table**

| PIN NIMAE | MUX FUNCTION | DEFAULT FUNCTIO N | DEFAULT I/O | POWER PLANE | GPI MODE(*1) | FUNC SELECT REG. | IO SELECT REG. | GPI MODE SELECT | GPO LEVEL |
|---|---|---|---|---|---|---|---|---|---|
| GPIO0 | PAR | PAR | follow PCI | VDD | Level | ACPI 28.1 | ACPI 20.0 | ACPI 24.0 | ACPI 1C.0 |
| GPIO1 | PMCLK | GPIO1 | IN | KBVDD | Toggle/Edge | SIO CFG 60.3 | ACPI 20.1 | ACPI 24.1,2A.1 | ACPI 1C.1 |
| GPIO2 | KBCLK | GPIO2 | IN | KBVDD | Level | SIO CFG 60.3 | ACPI 20.2 | ACPI 24.2 | ACPI 1C.2 |
| GPIO3 | CPU_STP#/ SLP# | GPIO3 | IN | VDD | Level | ACPI 28.3 | ACPI 20.3 | ACPI 24.3 | ACPI 1C.3 |
| GPIO4 | FAN1 | GPIO4 | IN | VDD | Level | None | ACPI 20.4 ACPI 28.4 | ACPI 24.4 | ACPI 1C.4 |
| GPIO5 | PME0#/ DUAL_ON# | GPIO5 | IN | RTCVDD | Level | APC 03.3 APC 04.5 | APC 04.4 | ACPI 24.5 | ACPI 1C.5 |
| GPO6 | CKE_S/ ACPILED | CKE_S | OUTPUT | RTCVDD | Level | APC 04.2 APC 07.6 | None | ACPI 24.6 | ACPI 1C.6 |
| GPIO7 | OC0#/PPS | GPIO7 | IN | VDD | Level | ACPI 28.6~7 | ACPI 20.7 | ACPI 24.7(*2) | ACPI 1C.7 |
| GPIO8 | OC1# | GPIO8 | IN | VDD | Level | ACPI 28.8 | ACPI 20.8 | ACPI 24.8 | ACPI 1C.8 |
| GPIO9 | THERM#/ /SMBALERT#/ BTI | GPIO9 | IN | VDD | Level | ACPI 28.9 | ACPI 20.9 | ACPI 24.9 | ACPI 1C.9 |
| GPIO10 | ACPILED/ PME1# | GPIO10 | IN | RTCVDD | Level | APC 04.1 APC 07.7 | APC 04.3 | ACPI 24.10 | ACPI 1C.10 |
| GPIO11 | FAN2 | GPIO11 | IN | VDD | Toggle/Edge | None | ACPI 20.11 | ACPI 24.11, 2A.11 | ACPI 1C.11 |
| GPI12 | VIN0 | GPI12 | IN | VDD | Toggle/Edge | None | None | ACPI 24.12,2A.12 | ACPI 1C.12 |
| GPI13 | VIN1 | GPI13 | IN | VDD | Toggle/Edge | None | None | ACPI 24.13,2A.13 | ACPI 1C.13 |
| GPI14 | VIN2 | GPI14 | IN | VDD | Toggle/Edge | None | None | ACPI 24.14,2A.14 | ACPI 1C.14 |
| GPI15 | VIN3 | GPI15 | IN | VDD | Toggle/Edge | None | None | ACPI 24.15,2A.15 | ACPI 1C.15 |
| GPIO16 | IOCHK# | GPIO16 | IN | VDD | Level | ACPI 28.12 ACPI 28.13 | ACPI 20.16 | ACPI 24.16 | ACPI 1C.16 |
| GPIO17 | SIRQ | GPIO17 | IN | VDD | Level | ACPI 28.5 | ACPI 20.17 | ACPI 24.17 | ACPI 1C.17 |

*1 : When GPIO is configured as the INPUT "Toggle" mode, the "Polarity (GPI Mode)" bit in ACPI Reg. 24[1&:0] is ignored.

*2: GPIO7's Polarity, while configured in output mode, can be defined by programming ACPI Reg24[7]. This function is special in USB Application.

*3: The details can be referred in ACPI & APC SPEC of 5595.

Since these signals are multiplexed with other signals, the BIOS need to program the associated bit to select the usage first. Then, the input/output modes for these pins are further determined if they are selected as GPIO function. While the GPIOx pins are inputs for the SiS5595, GPI [15:12], GPIO11, and GPIO1 can be set as either edge (falling or rising) triggered or toggle mode by programming bit [15:11,1] of register 2Ah respectively. The rising or falling edge depends on the programming of bit [15:11,1] of reg. 24h respectively. Except these pins, the rest of the GPIO pins are sensitive to level trigger only. The high or low sensitive mode depends on the programming of bit [17:0] of register 24h respectively. The default active level is low. While in the input mode, the input level of each GPIO pin can be directly read back by reading the corresponding bit in the GPIO Pin Status Register Located in ACPI Register 1Ch. Besides, 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 Pin Status registers. When configured in the input mode, activation of any of the GPIO[17:7,5:0] pin can set the corresponding status bit in GPE_STS reg. with a consequence of generating a Power Management Event (SCI, SMI#, or wake-up event) if the corresponding enable bit of GPE_EN reg. is set.

For GPIO5, GPO6, and GPIO10, the following rules are supposed to be followed specifically:

1) If GPIO5 and GPIO10 are not adopted, the two pins can not be left open. One external pull-up or pull-down resistor is required. While GPO6 is not adopted, this pin can be left open.

2) The three pins are put in high impedance state upon the battery is plugged. The logic is retained unless it is programmed to serve as other functions. If they are designed as GPIO function, they can be controllable by appropriate programming, while power is on. While the system power is down and these pins are programmed to be output mode, they are rendered to output low state. As a summary of these GPIO pins working at the GPIO mode, we have to:

   - Select them working in GPIO function.
   - Select them functioning in the input/output mode.
   - Drive output high/or low depending on application.

Note that the bits control the functionality of these multifunction pins, and their input/output mode are stored in the APC registers. Their logic values are retained as long as the RTC power exists. For instance, if DUAL_ON#/GPIO5/PME0# is selected to function as GPIO with input mode, GPIO5 logic will be controlled by setting bit5 of Register 24h in the ACPI I/O space while the system power is on. While the system power is down, GPIO5 is enforced high no mater what value has been programmed before. Later when the system power is on again, GPIO5 stays at output mode with logic controlled by bit 5 of ACPI Register 24h, which by default is high. For more detail information, please refer to S3 State Related Signals in the RTC module.

- **GPIOx Stop and Reload Timers Events**

GPIO[17:16, 11:7, 5:1], GPI[15:12] events also can be used to reload PMU standby timer and stop General Purpose Timer in ACPI. Application software can specify which GPIO events are able to reload or stop timers (PMU and ACPI) in ACPI registers 2E~2Fh. For reloading PMU standby timers, bit 3 of 40h, 44h, 48h registers should enable for each timer in PMU block.

#### 3.2.1.3 Legacy Features

- **Legacy SMI# Events**

Legacy SMI# include DAM SMI#, USB SMI#, Periodic SMI#, Wake up SMI# from S1 /S2, Software SMI#, Serial IRQ SMI#, and GPIOx SMI#. All information are presented in Figure 3.2-9 SCI / SMI# Events Overview

- **SMI# Command**

ACPI register 35h is a SMI# command port. A SMI# will be generated if the APCI register 31h, bit 4 (SMICMD_EN) is enabled and an I/O write to SMI command port is detected. The SMI# handeler can check the SMI command port to determine which action should be taken.

- **Periodic SMI# Event**

Periodic SMI# can generate SMI# every 16 sec if the PERSMI_EN (ACPI 31h bit 2) is set. This allows the SMI# handler to periodically give warning to the user for delivering the low battery message, for instance.

### 3.2.2 POWER MANAGEMENT UNIT

Basically, the legacy PMU provides three main functions as follows:

#### 3.2.2.1 System Activity Monitoring

The system activity monitor watches, within a specified monitoring period, at the system events to decide when and to which green state the system will transit into. SiS5595 provides three independent system activity monitors to fulfill the requirement of flexible, and wide range of today's Green PC application.

Basically, the system activity monitor consists of a system standby timer specifying the "Monitor -Period", and an Event Recognizer to detect the enabled "Reload Events". Counting down the timer until expiration mechanizes the "Monitor Period". However, the timer is reloaded with its initial count, and re-counted down once any of the enabled Reloaded Events is detected. If the specified Monitor Period is elapsed (certainly without any reload event detected), the timer is expired with the result of generating SMI# to invoke the SMI# routine to do any preferred action, such as turning off the screen, transiting the system into throttling state or sleeping state by throttling the STPCLK# or keeping the STPCLK# asserted, respectively.

Each Standby Timer is 8-bit wide with 14.318MHz as the Clock source, and provides 4 levels of granularity, namely 17.8us, 4.58ms, 1.17us, and 5min. Register 79h, 7Ah, and 7Bh of the PMU configuration space defines the initial count for the system standby timer 0, 1, and 2, respectively. Bit[7:6], Bit[5:4], and Bit[3:2] of the PMU register 7Ch defines the granularity for the system standby timer 0, 1, and 2, respectively. ***Table 3.2-4 Reload Events for Each Timer*** summarizes the Reload Events that the Event Recognizers in SiS5595 can support. There are three independent Event Recognizers provided, and their Reload Events can be defined by programming PMU Register 40h-43h, 44h-47h, and 48h-4Bh, respectively.

**Table 3.2-4 Reload Events for Each Timer**

| TIMER | RELOAD EVENT |
|---|---|
| System Standby Timer 0 (PMU) | Defined on PMU Configuration Register 40h~43h<br>Hard Disk Primary Channel Drive 0/1: 1F0h~1F7h, 3F6h, Bus master IDE port,<br>Hard Disk Secondary Channel Drive 0/1 : 170h~177h, 376h, Bus Master IDE port,<br>Keyboard Port : 60h, 64h<br>Serial Port 1 : 3E8h~3Efh, 3F8h~3FFh<br>Serial Port 2 : 2E8h~2Efh, 2F8h~2FFh<br>Parallel Port :278h~27Fh, 378h~37Fh, 3BCh~3BEh<br>IRQ[15:3,1:0], NMI (Please refer to Register 68h~69h, 6Ah~6Bh, 6Ch~6Dh)<br>RING IN<br>Master request from USB, ISA Master or DMA cycle<br>Programmable 10 bit I/O Port (Please refer to Register5Ch~5Dh)<br>Programmable 16 bit I/O Port (Please refer to Register5Eh~5Fh)<br>Memory Address A0000h~BFFFFh<br>Memory Address C0000h~C7FFFh<br>VGA I/O Port 3B0h~3DFh<br>Linear Frame Buffer Memory Address (Please refer to Register 4Eh~4Fh)<br>Microsoft Sound System Port, one of the 530h~537h, 604h~60Bh, E80h~E87h, F40h~F47h (Please refer to Register 4Ch~4Dh)<br>Sound Blaster Port, one of the (220h~22Fh, 230h~233h), (240h~24Fh, 250h~253h), (260h~26Fh, 270h~273h), (280h~28Fh, 290h~293h)<br>MIDI Port, one of the 300h~333h, 310h~313h, 320h~323h, 330h~333h<br>Game Port 200h~207h, 388h~38Bh<br>GPCS0# (Please refer to Register 70h~73h)<br>GPCS1# (Please refer to Register 74h~77h, 78h)<br>INIT<br>EXTSMI#<br>Master request from 530 or 5600/620 (BM_REQ#)<br>AGP cycle from 530 or 5600/620 (BM_REQ#)<br>GPI[17, 15:7, 5:1] |
| System Standby Timer 1 (PMU) | Defined on PMU Configuration Register 44h~47h<br>The reload event is same as System Standby Timer 0 |
| System Standby Timer 2 (PMU) | Defined on PMU Configuration register 48h~4Bh<br>The reload event is same as System Standby Timer 0 |

Upon power up, the three standby timers are frozen which means they keep reloading the initial count from their associated system timer register (default 00h). Setting "System Standby Timer X SMI# Enable (bit 31, 30, and 29 of PMU Register 60h-63h) will de-freeze the system standby timer 0, 1, and 2, respectively. Upon expiration, the corresponding System Standby Timer X SMI# Request Status Bit is set, and the SMI# is generated. Note that the standby timer stays at "reloading" status upon expiration.

The SMI# Request Status allows the SMI# routine to identify which SMI# source is coming. Bit31, 30, and 29 of Register PMU 64h-67h correspond to the SMI# Request Status Bit for the system standby timer 0, 1, and 2, respectively. Writing a logic 1 to the SMI# Request Bit clears the status bit, and enables the standby timer count down again if its corresponding SMI# Enable bit is not cleared. It is recommended to disable the SMI# Enable bit before programming the initial count to make a clean start, meaning that the standby timer really counts down starting from the initial count. Before the timer is expired, it can be reloaded if any of the enabled reload events is identified, and then counts down again.

#### 3.2.2.2 Wake Up Event Recognizers

Two independent Wake Up Event Recognizers are provided to allow the system designers to flexibly meet the Green PC application. The main mission of the WUER (Wake Up Event Recognizers) is to generate SMI# upon detecting the wake up events, if the corresponding SMI# Enable bit is set.

**Table 3.2-5 Wakeup Event from PMU**

| TIMER | RELOAD EVENT |
|---|---|
| WAKEUP 0 (PMU) | Defined on PMU Configuration Register 50h~53h<br>The wakeup event is same as System Standby Timer 0 except GPI [17, 15:7, 5:1]. |
| WAKEUP 1 (PMU) | Defined on PMU Configuration Register 54h~57h<br>The wakeup event is same as System Standby Timer 0 except GPI [17, 15:7, 5:1]. |

To summarize the supported Wake UP Event Set, namely WakeUp0, and WakeUp1 which can be defined by programming the wake up event control registers locating on 50h-53h and 54h-57h of PMU configuration space, respectively. Bit 28, and 27 of the SMI# Enable Register enables/disables the generation of SMI# while wake up event is recognized. The WUER is designed to be quite independent on the system state (Sleep, or Throttling state). The wake up event can be recognized, and thus SMI# be generated as long as its corresponding enable bit in the wake event control register is set.

#### 3.2.2.3 SMI# Generation Logic

SiS5595 PMU allows a versatile event that could give rise to the generation of SMI#. Two sets of registers, namely SMI# Enable Registers, and SMI# Status Register relate to the SMI# generating logic. The SMI# Enable register locating on Register 60h to 63h enables the generation of SMI# upon that any of the enabled events is recognized. SMI# Request Status register, locating on Register 64h to 67h reflects the event(s) producing the SMI#.

## 3.3 SMBUS FUNCTIONAL DESCRIPTIONS

The System Management Bus (SMBus) is a subset of the Phillips $I^2C$ protocol. It is a two-wire interface for system to obtain chips' information, to indicate device model or part number, and so on. It also can be applied to control devices such as system suspend/wakeup, return device status or extend I/O for control purposes.

The SMBus host controller contains a Host Master and a Host Slave. The slave address may contain an alias address. The host master and slave are fully independent, which conveys that the host master can communicate with the host slave via the SMBUS protocol. SiS5595B also supports SMBALERT# signal for slave devices to assert request. The SMBALERT# pin is multiplexed with GPIO9 and BTI.

Access the SMBus register can be achieved by issuing an I/O write to ACPI offset 38H with an INDEX, then followed by a I/O read or write to APCI offset 39h with Data.

The SMBus Interrupt Request can be routed to any IRQ or SMI#/SCI.

**Table 3.3-1 SMBus Interrupt Table**

| INTERRUPT TYPE | ASSOCIATED REGISTER |
|---|---|
| IRQ | SIO.7E.7 : Data Acquisition Module and SMBus IRQ Mapping First Enable<br>SIO.7E.5 : SMBus IRQ Mapping Second Enable<br>SIO.7E.3~0 : IRQ Routing table |
| SCI/SMI# | APCI.14.23 : SMB_STS<br>APCI.18.23 : SMB_EN |
| SMI# | APCI.30.3 : SMBSMI_STS<br>APCI.31.3 : SMBSMI_EN |

### 3.3.1 SMBUS HOST MASTER INTERFACE

SiS5595B SMBus Host Master supports full SMBus protocol, including Quick command, Send/Receive Byte, Read/Write Byte/Word, Read/Write Block, and Process Call. The SiS5595B supports Read/Write Block command by an 8-byte buffer instead of 32-byte. For block transfer size larger than 8 bytes, software manipulation is required. Once the eight bytes block data have been transferred, the Sub-Block Request Status is employed to tell service routine some data not processed yet. For the maximun of 32 bytes block data transfer, four SMBus Interrupts will be raised by 5595 to complete the transfer.

When a Host transfer is initiated, the HOST_BUSY status bit will be set. The software is not allowed to start a new command protocol till the HOST_BUSY reset to 0. The currently HOST Master transfer cycle can be stopped by writting a ‘1’ to SMB_Kill bit. When a ‘1’ is written to SMB_Kill bit, all status bit for host master and host slave will be reset. After HOST_BUSY status bit becomes zero, the software is allowed to initiate a new transfer.

### 3.3.2 SMBUS HOST SLAVE INTERFACE

The Host has a Reserved Address in 0001000xb (x is R/W bit for command protocol) and an Alias Address defined in SMB_Alias (SMBus Reg.13h). An idividual enable and status bit is implemented for both address decoder. If a modified Write Word has been received by Host

Slave with the Slave Address hit the Reserved Address or Alias Address, a SMBus Interrupt will be raised. The device address, high data byte and low data byte are stored in SMBus register 10h ~ 12h. Once the status bit HIT_HAA or HIT_HRA is set to '1', the Host Slave is unable to receive other modified Write Command until this status bit is reset.

### 3.4 USB INTERFACE

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.

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:

- Provides USB Host Controller function to meet the Universal Serial Bus Specification version 1.0, with full compatibility to the Open Host Controller Interface Specification for USB Release 1.0
- Provides 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 Overcurrent Protection on the USB ports.

The following figure illustrates the USB System Block Diagram.

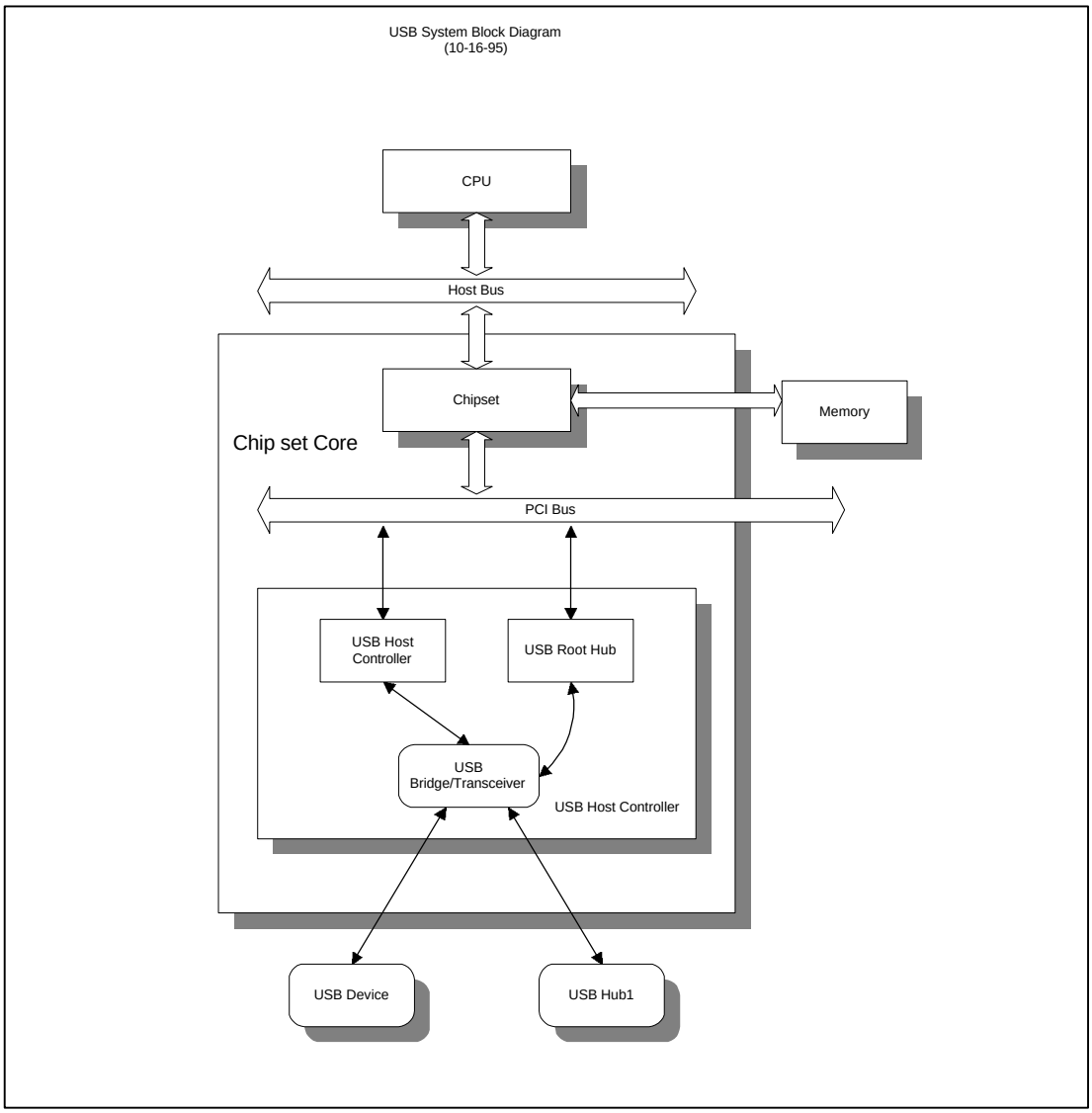


**Figure 3.4-1 USB System Block Diagram**

## 3.5 DATA ACQUISITION MODULE (DAM)

### 3.5.1 GENERAL DESCRIPTION

The Data Acquisition Module provides PC system hardware environment monitoring including power supply voltages and cooling fan rotational speed. In addition, up to eight external temperature sensors (such as LM75 device) can be deployed on the motherboard with their outputs connected together through the BTI line into the SiS5595 to optionally generate SMI# or IRQ. The most typical application will be putting one temperature sensor chip or one thermistor under the CPU to monitor the ambient temperature of CPU.

The Data Acquisition Module internally contains an Analog-to-Digital Converter (ADC) to continuously transform analog signals from power supply voltages or temperature sensor into 8-bit digital binary numbers. The rotational speed of fans can be calculated by counting the cycle time of digital pulses from tachometer output based on an internally generated 22.5KHz clock. For every input sources to be monitored, the actual readings which have been calculated or transformed will be compared with those upper and lower limits as programmed in their associated registers. Upon a limit being exceeded, a SMI# or IRQ can be optionally generated to inform the operating software to take proper steps to protect the system from critical failure.

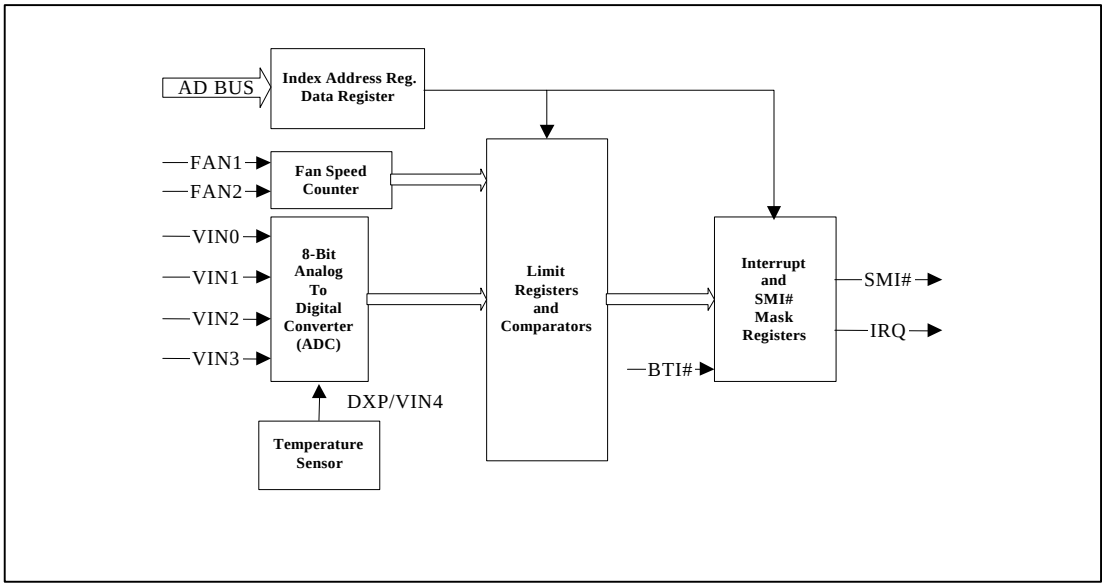


**Figure 3.5-1 Data Acquisition Module Block Diagram**

### 3.5.2 POWER SUPPLY VOLTAGE MONITORING

The SiS5595 Data Acquisition Module supports up to five positive voltage inputs monitoring. Analog inputs from power supply voltages, in term, will be converted into 8-bit unsigned binary numbers by the internal Analog-to-Digital Converter (ADC) with a resolution of 16mV. The resulting measurable voltage range is therefore from 0V to 4.096V. For voltages higher than 4.096V, an external voltage attenuator circuit consisting of two properly selected resistors can be used to scale down the voltage into the measurable range. Typically, the four voltages to be monitored in a PC system are +12V, +5V, +3.3V and +2.5V respectively. The +12V and +5V require external attenuators. The SiS5595 application notes contain the recommended values of the resistors for the attenuators.

### 3.5.3 FAN SPEED MONITORING

The fan inputs are from fans equipped with tachometer pin, which carries continuous digital pulses, indicating its current rotational speed. The longer the pulse's cycle time, the slower the fan rotates. The tachometer output requires a pull-up resistor connected to +5V. The SiS5595 Data Acquisition Module supports two fan inputs monitoring. The typical application will be connecting one to CPU fan and the other one to the case fan.

The cycle time of the pulses from the tachometer output is measured by a counter counting up at 22.5KHz clock. The reading of counter can be derived from the following formula. At the nominal speed, a reading of 153 is expected.

**Count = 22500 * 60 / RPM * Divisor**

Where:

RPM is the fan's nominal rotations per minute

Divisor is the normalized factor programmed in the fan divisor register in order to get a reading of 153 at nominal speed for every RPM. It should be:

8800RPM : 1

4400RPM : 2

2200RPM : 4

1100RPM : 8

If the speed of fan decreases, the reading should go up until the maximum reading of 255 is reached. This may indicate that the fan speed is very slow or has already stopped. There is no upper limit for the fan speed; only lower limit can be programmed. Once the lower limit is exceeded, a SMI# or IRQ can be generated to inform the operating software to take proper steps such as throttling CPU activities by asserting the STPCLK#.

### 3.5.4 THE ROUND-ROBIN SAMPLING CYCLE

Once the monitoring process is started by setting a 1 to the Start bit at Configuration register, the Data Acquisition Module will start sampling the inputs from each sources in a round-robin manner according to the following order:

(1) Reserved (2) VIN0 (3) VIN1 (4) VIN2 (5) VIN3 (6) DXP/VIN4 (7) FAN1 (8) FAN2

A complete round-robin cycle will be completed with the rate of approximately 8/7second, and the switch time for the eight input sources is evenly distributed within one cycle, i.e., 1/7 second. This gives the ADC enough time to provide a stable value before switching to the next input. The readings for each input sources will be updated approximately once every 8/7 second and can be read from these offset registers.

| OFFSET REGISTERS | DESCRIPTION |
|---|---|
| 20h/60h | VIN0 reading |
| 21h/61h | VIN1 reading |
| 22h/62h | VIN2 reading |
| 23h/63h | VIN3 reading |
| 24h/64h | DXP/VIN4 reading |
| 27h/67h | Reserved |
| 28h/68h | Fan1 reading |
| 29h/69h | Fan2 reading |

The Analog-to-Digital Converter can be calibrated through PCI-ISA configuration Registers 7Ch~7Dh. Users are advised NOT to program these calibration registers.

### 3.5.5 INTERFACE REGISTER BLOCK

The Data Acquisition Module internal registers can be accessed through Interface Register Block located in I/O address space, with its base address relocatable anywhere within the low 64K I/O space by programming PCI-ISA configuration Register 68h~69h and must be 8-byte aligned. The Interface Register Block consists of two registers—Index Address Pointer register and Data register located at base address +05h and base address +06h respectively.

| BASE ADDRESS | REGISTER |
|---|---|
| +05h | Index Address Pointer |
| +06h | Data |

All accesses to the Data Acquisition Module internal registers must be through the two registers. First the Index Address Pointer should be written with the offset address of the internal register, then the actual data read or write transaction can be carried out by reading or writing to Data register.

### 3.5.6 INTERNAL REGISTERS

Limit registers set the lower and higher limits for voltages, temperature and fan speed. Note that there is only lower limit for fan speed. A lower limit is considered to be exceeded if the reading is less than lower limit, while a higher limit is considered to be exceeded if the reading is greater than or equal to the higher limit. When the round-robin monitoring process updates a reading which has exceeded a limit, an IRQ or SMI# can be generated, if the corresponding Mask bit in NMI Mask register or SMI# Mask register is not disabled. At the same time, the corresponding interrupt status bit at the Interrupt Status register will be set. Reading the Interrupt Status register will reset all bits as well as clear outstanding interrupts.

## 3.6 INTEGRATED REAL TIME CLOCK (RTC)

### 3.6.1 REAL TIME CLOCK MODULE

The Real Time Clock module in the SiS5595 contains the industrial standard Real Time Clock, which is compatible to MC146818, and the Automatic Power Control (APC) 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 generator, 112 Bytes of standard CMOS SRAM, and 128 Byes of extended CMOS SRAM. The Automatic Power Control part provides the software/hardware power up/down functions. Figure 3.6-1 shows the block diagram of the RTC module.

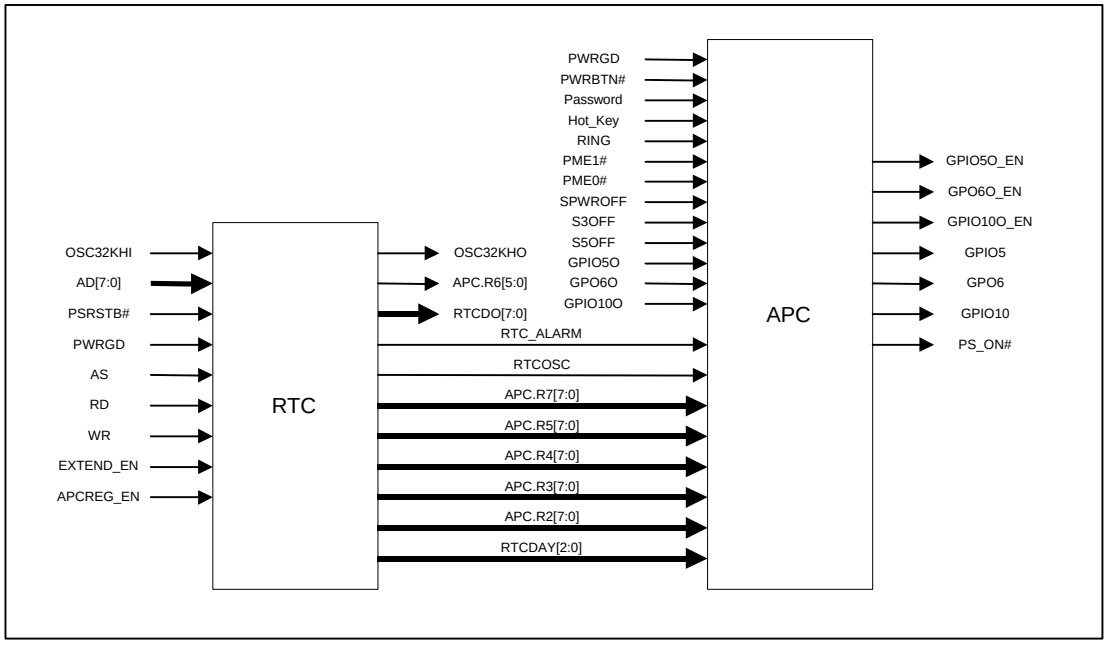


**Figure 3.6-1 RTC Module Block Diagram**

### 3.6.1.1 RTC Registers & RAM

Three separate RTC registers & RAMs are provided in the SiS5595. One is called the Standard Bank, another is the Extended Bank, and the other is the APC registers. All of these registers are referenced through the same address and data port, i.e. Port 70h and 71h, respectively. The access control with which the three portions of registers can be appropriately addressed are stored in PCI-ISA: 45h[3] (EXTEND_EN bit) and PCI-ISA: 45h[1] (APCREG_EN bit). Figure 3.6-2 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 0Ah, 0Bh, 0Ch, and 0Dh contain the RTC control and status bytes. The last two bytes (7Eh, 7Fh) are the Day of the 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 remaining 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.

**Table 3.6-1 The Access of Standard/Extend Bank and APC Registers**

| | STANDARD BANK | EXTEND BANK | APC REGISTERS |
|---|---|---|---|
| EXTEND_EN | 0 | 1 | 0 |
| APCREG_EN | 0 | 0 | 1 |

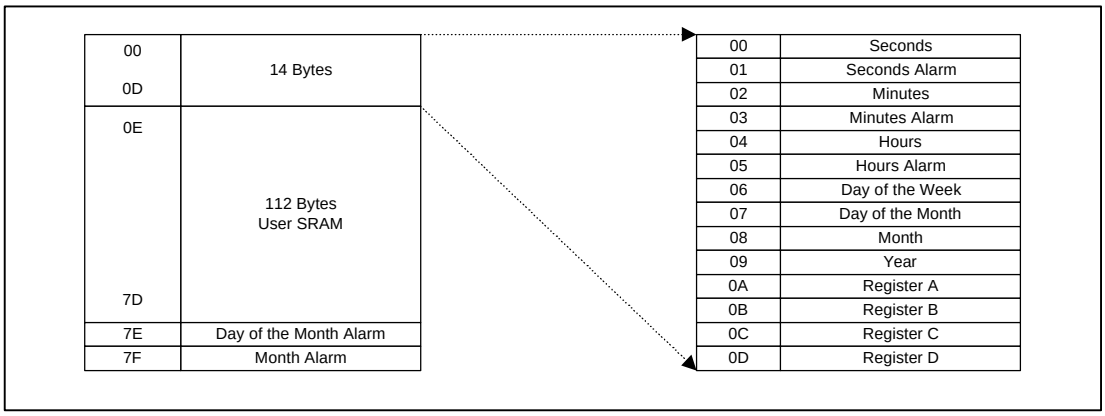


**Figure 3.6-2 Address Map of the Standard Bank**

### 3.6.1.2 RTC Update Cycle and RTC Alarm Event

The primary function of the update cycle is to increase the *Seconds* byte, update the other time bytes, and compare each alarm byte with the corresponding time byte. When the alarm time is written in the *Month Alarm, Day of the Month Alarm, Hours Alarm, Minutes Alarm* and *Seconds Alarm*, a RTC Alarm Event will be activated at the time specified in those alarm registers. A “don’t care” code can be set in one or more of the five alarm bytes by writing the two most significant bits to 1 (11XX XXXX). For example, if the “don’t care” code is set in the *Month Alarm*, the RTC Alarm Event will be generated at the time specified in the other four alarm registers in every month. Similarly, the RTC Alarm Event will be generated at the time specified in *Hours Alarm*, *Minutes Alarm* and *Seconds Alarm* every day if the *Month Alarm* and the *Day of the Month Alarm* are both set to “don’t care” code. Note that the *Day of the Month Alarm* and *Month Alarm* are “don’t care” by default. Figure 3.6-3 shows the block diagram of the RTC.

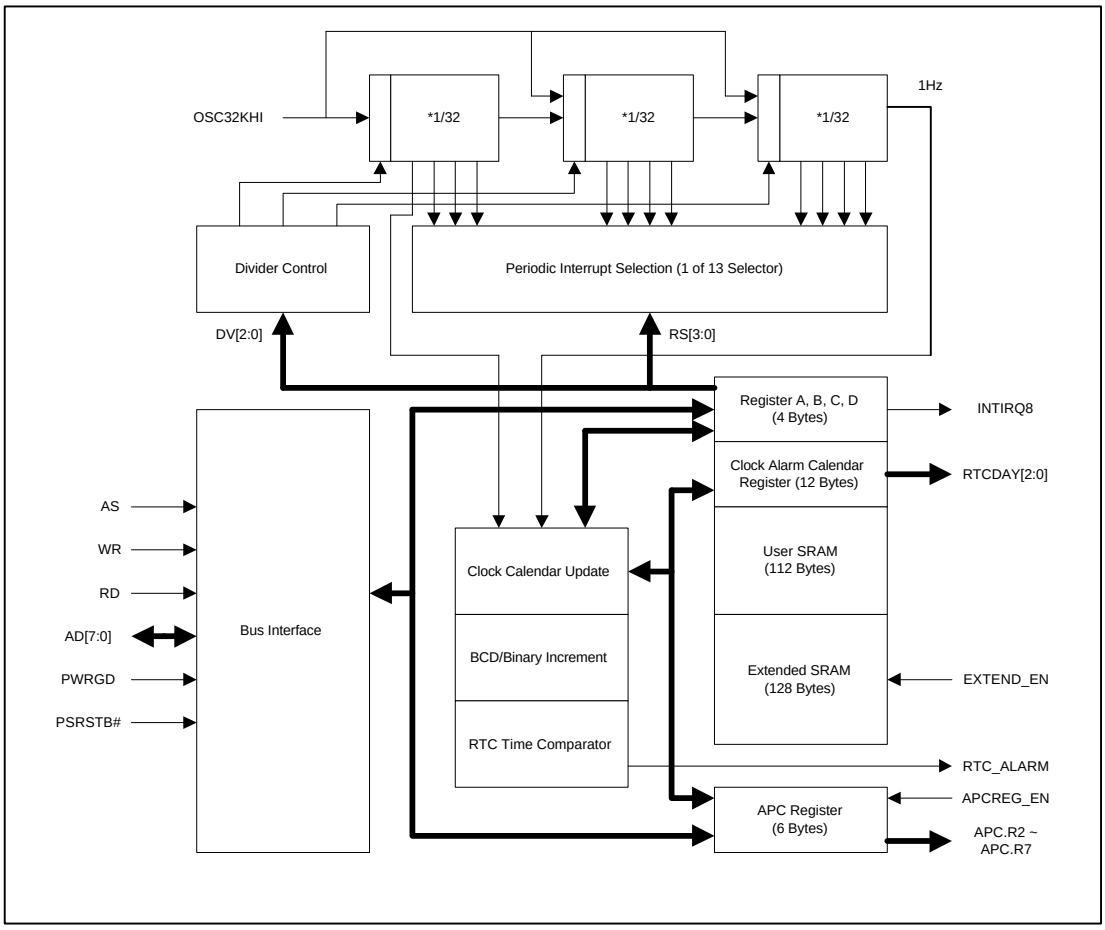


**Figure 3.6-3 Block Diagram of RTC**

### 3.6.1.3 RTC External Connection Requirement

The RTC is powered by RTCVDD and RTCVSS. In reality, not only the internal circuitry of the RTC, but also the pins associated with the RTC module are also powered by this specific power planes. They are PS_ON#, PWRGD, PSRSTB#, PWRBT#, RING, OSC32KHI, OSC32KHO, GPIO5/PME0#/DUAL_ON#, GPO6/CKE_S/ACPILED and GPIO10/PME1#/ ACPILED. RING is an input pin by default, which should be pulled low while not used.

## 3.7 AUTOMATIC POWER CONTROL MODULE

APC module is only functional while internal RTC is used. If an external RTC is used, all Automatic Power Control functions will be disabled automatically. ATX power supply has a control signal PS_ON# and two set of VDD named VDD5V and AUX5V. The AUX5V will output +5V as long as the AC power is applied to the ATX power supply, while the VDD5V will only be activated when the PS_ON# signal is output low. APC controls the signal PS_ON# to turn on or turn off VDD5V of ATX power supply. SiS5595 also supports PC'98 power supply, which will be discussed in "APC Functions" section.

### 3.7.1 APC REGISTERS

The APC registers are provided to support the Auto Power Control Function. The register 02h defines the "Day of the Week Power Up" setting byte. The 03h, 04h, 05h, 06h, 07h, and 08h registers contain the control information for the Automatic Power Control functions. Table 3.7-1 Address Map of the APC Control Registers shows the address map of the APC registers.

**Table 3.7-1 Address Map of the APC Control Registers**

| ADDRESS | REGISGER NAME |
|---|---|
| 00h | Reserved |
| 01h | Reserved |
| 02h | APC Register 02h |
| 03h | APC Register 03h |
| 04h | APC Register 04h |
| 05h | APC Register 05h |
| 06h | APC Register 06h |
| 07h | APC Register 07h |
| 08h | APC Register 08h |

### 3.7.2 APC FUNCTIONS

Let us take the following power up/down sequence as an example to illustrate the APC functions. Figure 3.7-1 Typical Timing Sequence on the Power Control Related Signals shows the typical timing sequence of the power control related signals.

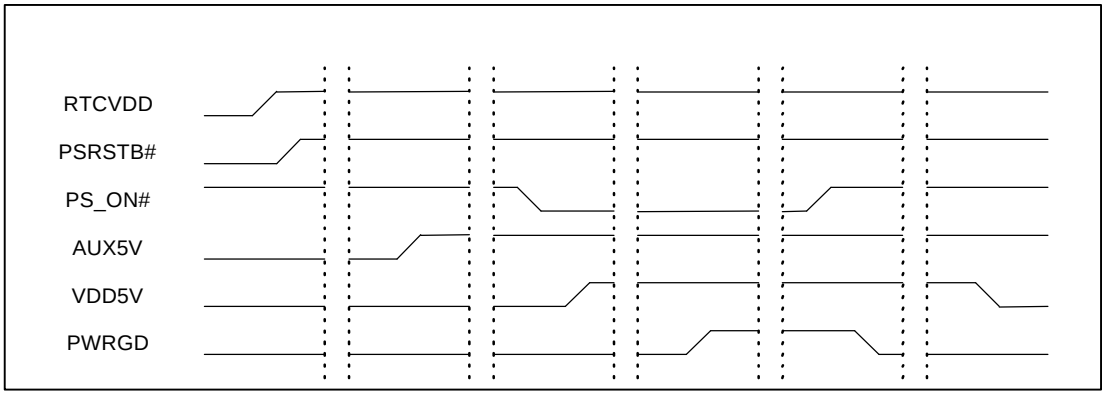


**Figure 3.7-1 Typical Timing Sequence on the Power Control Related Signals**

#### 3.7.2.1 APC Power Source Connection Requirement

The PSRSTB# is traditionally used to convey the battery life status to the RTC module. In SiS core based system, the signal is also used to determine if the internal RTC is to be used. A logic high on PSRSTB# inform SiS5595 that the internal RTC is selected.

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 equipped with the ATX power supply, it is recommended to design the power conversion circuitry powered by both the AUX5V and battery. In practical application, the PSRSTB# is low only when both the battery and the AUX5V is low. That is, PSRSTB# is low whenever the battery happens 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 battery. As long as the PSRSTB# is high and the system is in power down state (PWRGD is low), the power up events can be recognized and results in the assertion of the PS_ON# to have the ATX power supply provide VDD5V 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 keep working while VDD5V is removed, and can sense the "Power Up Request Events" to wake up the system by activating PS_ON#. In a word, 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.

#### 3.7.2.2 Power Up Request Events

During the power down period, the following events can power up the PC main board by the assertion of PS_ON#. They are Power Button On event (via PWRBT#), Hot Key Match event (via Keyboard Controller), Password Match event (via Keyboard Controller), RTC Alarm On event (via RTC Alarm), Ring Up event (via RING), Power Management Event 1 (via PME1#), Power Management Event 0 (via PME0#). Each power up function has its enable/disable bit specified in APC Registers. Except Power Button On event, all the rest power up events are enabled when APC_EN is set. APC module also provides five status bits to record the power up request events, they are MNUP_STS, ALMUP_STS, RNUP_STS, PME1_STS, and PME0_STS.

Note that PME0# and PME1# are low active logic and must be pulled high by external resistors. If they are pulled high by AUX5V and the system AC power source is suddenly off (that is, VDD5V and AUX5V are both disappeared), APC may recognize this as a power on event and will activate PS_ON#. If the AC power source is recovered within 4 seconds, the ATX power supply will be activated by this event. If PS_ON# is asserted due to an AC power source suddenly off event and the AC power source is not recovered within 4 seconds, APC will dessert PS_ON# automatically. In other words, APC will sample PWRGD while PS_ON# is asserted. If PS_ON# is asserted and PWRGD is not asserted within 4 seconds, APC will recognize this as a power failure event and will de-assert PS_ON# automatically. APC module provides a PSON_RSM bit (APC 07h[5]) to disable this power failure detecting function. While programming a 1 to this bit, PS_ON# can only be de-asserted by all power down events (Power Button Override, Software Power Off, S3 Power Off, and S5 Power Off). If RING is set to active low mode and the *System Powered-up by Ring* function is enabled, it will show the same characters, too.

The following is the detail description of these power-up events:

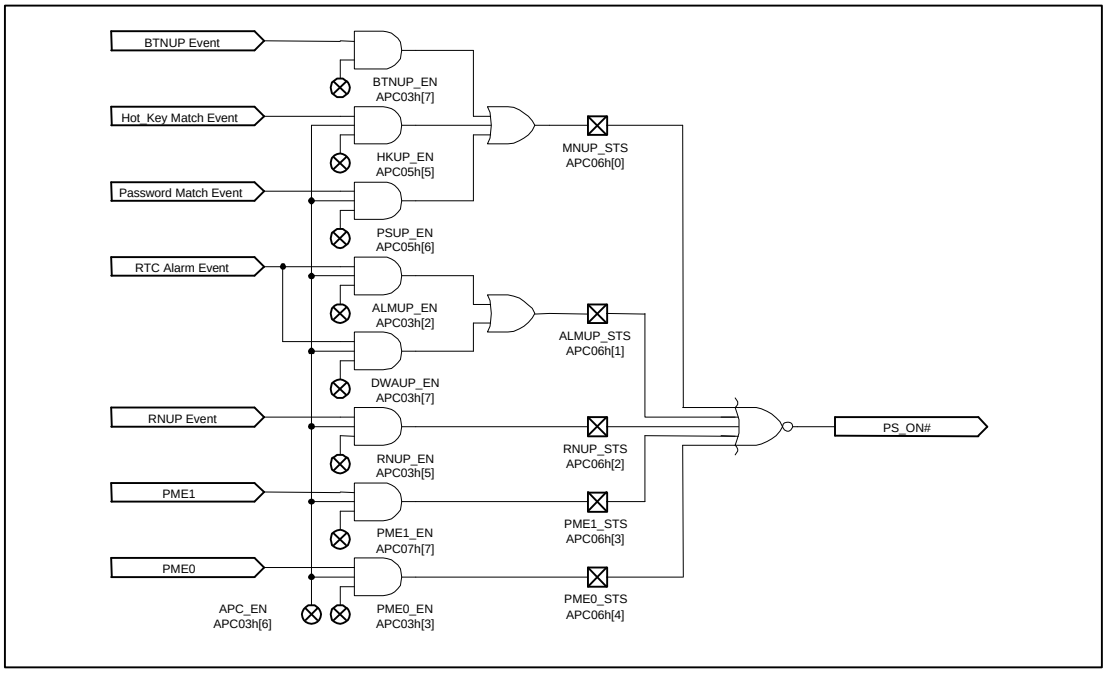


**Figure 3.7-2 Power Up Request Events**

- **Power Button On Event :**

SiS5595 provides a power button control pin, PWRBT#, to power up the system. 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 PS_ON#. While PWRGD is high, the assertion of PWRBT# less than 4 seconds results in an SCI/SMI event, and the assertion of PWRBT# more than 4 seconds will turn off the system. Power Button On function is enabled by default and can be disabled by writing a 0 to BTNUP_EN. If the AC power of the system is suddenly off (that is, VDD5V and AUX5V are both disappeared), the function will be enabled automatically to ensure that the user can power up the system by power button. Note that when PWRGD is low, Power Button on Event is controlled by APC. After PWRGD going high, this event will be recognized and responded by ACPI.

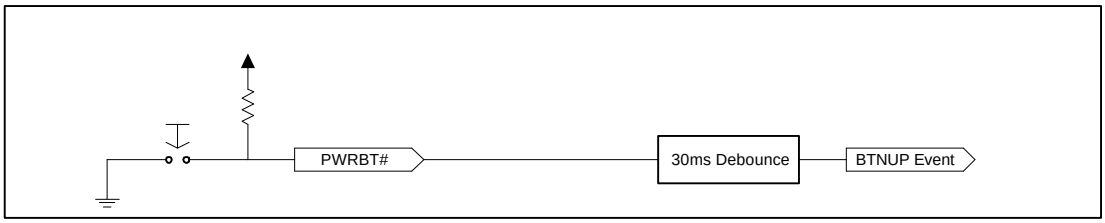


**Figure 3.7-3 Power Button On Event**

- **Hot Key Match Event and Password Match Event :**

SiS5595 build-in Keyboard Controller can generate two special events to power up the system. They are Hot Key Match Event and Password Match Event. These two functions will be disabled automatically while internal keyboard controller's power is suddenly off. Please refer to Keyboard Controller for more detail.

- **RTC Alarm On Event :**

When the time value of RTC matches the corresponding alarm bytes, the RTC would send an "RTC Alarm Event" to the APC module. RTC alarm event can be created form once per second to once per year. Beside the standard RTC Alarm power up function, SiS5595 also provides an additional power up function called the "Day of the Week Alarm Power Up". Following are the detail description of these two alarm power-up functions:

- **RTC Alarm Power Up Function :**

If APC_EN and ALMUP_EN are enabled and the system is in power down state, an RTC Alarm Event will power up the system.

- **Day of the Week Alarm Power Up Function:**

The *Day of the Week Alarm Power Up* Function can be enabled to power up the system on selected days within a week. With this additional feature, SiS5595 allows the user to power up the system, say, at 08:00 on each working day, and to stay at power off state on weekend. The Day of the Week Alarm Power Up byte is located in APC register 02h. Note that this feature is enabled when APC_EN and DWAUP_EN bits are both enabled, and ALMUP_EN is disabled.

- **Ring Up Event :**

While PWRGD is low, the detection of an active RING pulse lasting for more than 4ms would activate the PS_ON#. Note that the active high/low logic of the RING can be defined through programming RN_POL. While PWRGD is high, the detection of RING pulse would generate an SCI/SMI# event, which will be recognized and responded by ACPI. Please refer to ACPI section for more details.

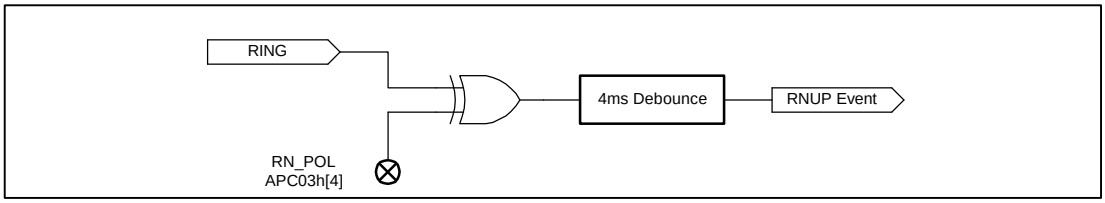


**Figure 3.7-4 Ring Up Event**

- **Power Manage Event 1 :**

When the power is removed, a high to low transition on PME1# indicates a power management event which will activate PS_ON#. When this function is used, APC 04h[3] must be set to 1 to configure this pin in input mode. This power up function can be used to accept PCI or AGP power management event (PME#) when system is in power down state.

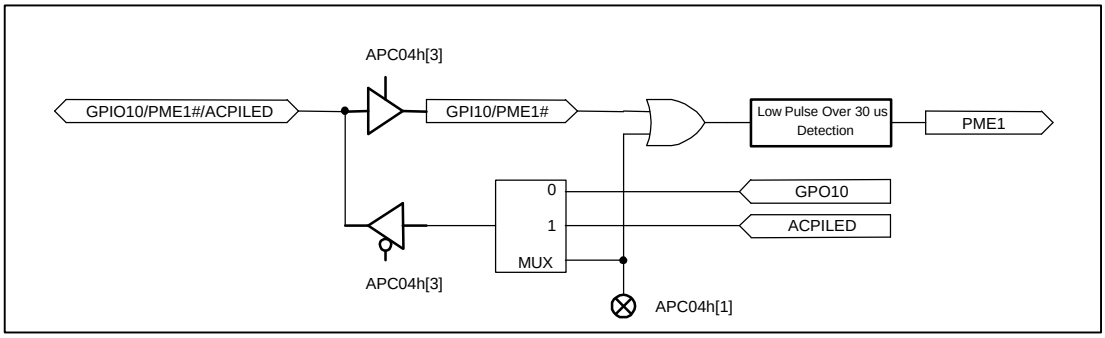


**Figure 3.7-5 Power Manage Event 1**

- **Power Manage Event 0 :**

When power is removed, a high to low transition on PME0# also indicates a power management event which will activate PS_ON#. When this function is used, APC 04h[4] must be set to 1 to configure this pin in input mode. This power up function can also be used to accept PCI or AGP power management event PME# when system is in power down state.

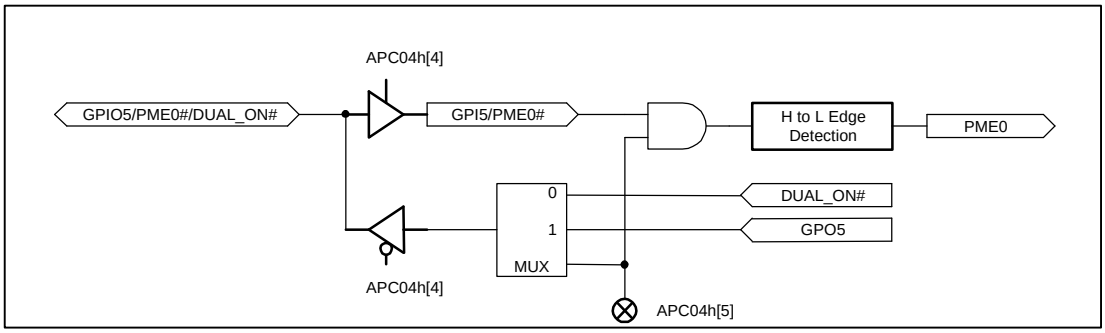


**Figure 3.7-6 Power Manage Event 0**

### 3.7.2.3 Power Down Request Events

While in the power up state, the following events can power down the PC main board by the de-assertion of PS_ON#. They are Power-Button-Over-Ride Event (via PWRBT#), S3 Power Off Event, S5 Power Off Event and Software Power Off Event (via PCI-ISA: 63h[2]). Following is the detailed description of these events:

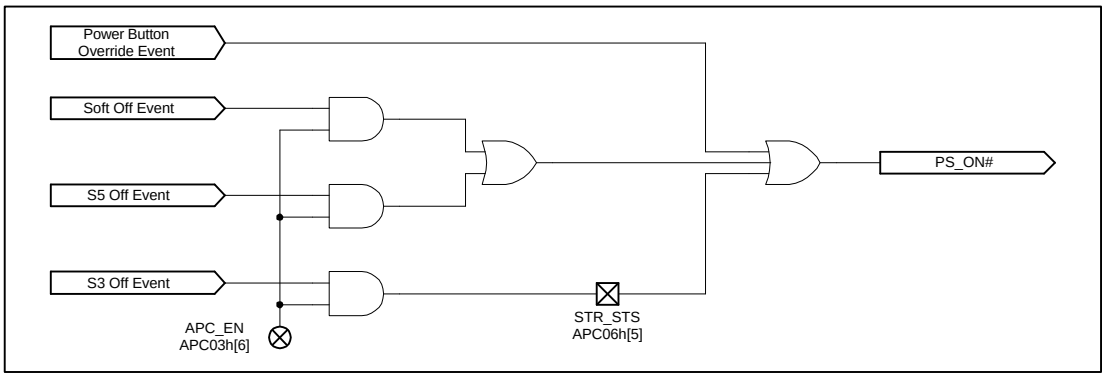


**Figure 3.7-7 Power Down Request Eve**

- **Power-Button-Over-Ride Event :**

Power-Button-Over-Ride event is generated when the Power Button has been pushed for more than 4 seconds. The SiS5595 will de-assert the PS_ON# if Power-Button-Over-Ride event occurs. Please note that this is purely done by hardware. No special support from BIOS is required.

- **Software Power Off Event :**

When APC_EN is enabled and the system is in power up state, writing a 1 to Power Off System Control bit (PCI-ISA: 63h[2]) can de-assert PS_ON#.

- **S5 Power Off Event :**

SiS5595 also provides another alternative to de-assert PS_ON# signal. When system is in power up state and APC_EN is enabled, programming a '1' to SLP_EN bit (ACPI 04h[13]) and '101' (S5 state) to SLP_TYP bits (ACPI 04h[12:10]) can sequence the system into S5 state with the consequence that PS_ON# is negated eventually.

- **S3 Off Event :**

When system is in power up state and APC_EN is enabled, programming a 1 to SLP_EN (ACPI 04h[13]) and '011' (S3 state) to SLP_TYP bits (ACPI 04h[12:10]) can force the system enter S3 state with the consequence that PS_ON# is negated eventually. This will be discussed in next section and ACPI section.

### 3.7.3 S3 (STR) FUNCTION RELATED SIGNALS

SiS5595 supports the ACPI S3 state or equivalently the 'Suspend to RAM (STR)' state which essentially turns off the system power except the power for the memory subsystem (mainly SDRAM). During the S3 State, the SDRAM memory subsystem should be put into self-refresh mode. The following illustrates the three multi-function pins, namely GPIO5, GPO6, and GPIO10 that are mainly used to support the S3 state.

#### 3.7.3.1 GPIO5/PME0#/DUAL_ON# :

| APC 04h[5] | APC 03h[3] | Function Select |
|---|---|---|
| 1 | 0 | GPIO5 |
| 1 | 1 | PME0# |
| 0 | X | DUAL_ON# |
| *APC 04h[4] : I/O Mode selection bit (0:output mode)<br>*APC 05h[7] : DUAL_ON# active level (1:active low) | | |

- **DUAL_ON# :**

While operating in this mode, the pin can be connected to the DUAL_ON# signal of the PC'98 power supply. APC 05h[7] determines the logic of this pin running in this mode. Note that: in addition to configure this pin as the DUAL_ON# function, and CKE_S can also be done as it will explain later. Following shows the output level of DUAL_ON# while programming APC 05h[7] to 0 or 1.

| | APC 05h[7] = 0 | APC 05h[7] = 1 |
|---|---|---|
| Normal State | 1 | 0 |
| Suspend State | 1 | 0 |
| Soft Off State | 0 | 1 |

#### 3.7.3.2 GPO6/CKE_S/ACPILED :

| APC 04h[2] | APC 07h[6] | Function Select |
|---|---|---|
| 1 | X | GPO6 |
| 0 | 0 | CKE_S |
| 0 | 1 | ACPILED |
| *APC 04h[0] : ACPILED supports 1 Hz blinking (0:disable) | | |

- **CKE_S :**

While entering S3 State, the SiS core based system can keep the SDRAM in the self-refresh mode by driving CKE_S low. Please refer to ACPI for more detail on the CKE_S function. Besides, when supporting PC'98 power supply, the CKE_S signal can be connected to the DUAL_ON# of power supply, too. The following summarizes the power supply subsystem operating states supported by the power supply '98.

| PS_ON# | DUAL_ON# | CKE_S | SYSTEM STATE | STANDBY OUTPUTS | DUAL OUTPUTS | MAIN OUTPUTS |
|---|---|---|---|---|---|---|
| 1 | 1 | Z | Soft Off | ON | OFF | OFF |

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | Suspend (S3) | ON | ON | OFF |
| 0 | X | Z | Normal | ON | ON | ON |
| 1.When system wakes up from S3 state, APC 04h[7] must be programmed to 1 to reset CKE_S output level.<br>* CKE_S must be pulled high by an external resistor. | | | | | | |

#### 3.7.3.3 GPIO10/PME1#/ACPILED :

| APC 04h[1] | APC 07h[7] | Function Select |
|---|---|---|
| 0 | 0 | GPIO10 |
| 0 | 1 | PME1# |
| 1 | X | ACPILED |
| *APC 04h[3] : Pin I/O Mode selection bit (0:output mode)<br>*APC 04h[0] : ACPILED support 1 Hz blinking (0:disable) | | |

- **ACPILED :**

It is up to the application to configure either GPO6 or GPIO10 as the ACPILED signal. ACPILED is driven low to turn on the LED while the system is in the S0, S1, or S2 states. It is floated in all other system states. Optionally, the LED can be put in the blinking state by setting APC 04h[0] while in the S0, S1, or S2 states.

### 3.8 INTEGRATED KEYBOARD CONTROLLER

The built-in KBC (Keyboard Controller) module uses hardwired methodology instead of software implementation as the traditional 8042 keyboard BIOS commands. In this way, the built-in KBC can have instant response to all the commands. It also has Fast Gate-20 and Fast Reset Features. Besides, the built-in KBC supports the industrial standard PS/2 mouse optionally. Moreover, the password security and the hot-key <CTRL+ALT+ Backspace> power up functions are implemented.

#### 3.8.1 STATUS REGISTER

The status register is an 8-bit read-only register located at address hex 64 of the I/O space and can be read at any time. It provides the state information of the built-in KBC and the interface. The definitions of each bit are described in the following:

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **Parity Error**<br>0: The last byte received from the keyboard had Odd Parity (No parity error).<br>1: The last byte received from the keyboard had Even Parity (Parity error). |
| 6 | RO | **Time-out error**<br>0: No Transmission Time-out Error is occurred.<br>1: Transmission Time-out Error is occurred. |

| | | |
|---|---|---|
| 5 | RO | **Auxiliary Output Buffer Full**<br>0: The Output Buffer's data is from the Keyboard.<br>1: The Output Buffer's data is from the Mouse. |
| 4 | RO | **Inhibit Switch**<br>0: The Keyboard is Inhibited.<br>1: The Keyboard is not Inhibited. |
| 3 | RO | **Command/Data**<br>0: The data is written to the I/O 60h, it is interpreted as a 'data byte'.<br>1: The data is written to the I/O 64h, it is interpreted as a 'command byte'. |
| 2 | RO | **System Flag**<br>This bit may be set to 0 or 1 by writing the system flag bit in the command byte. After a power on reset, it is cleared to 0. |
| 1 | RO | **Input Buffer Full**<br>0: The Input Buffer is Empty.<br>1: The Input Buffer is Full. Data has been written into the input buffer but the controller has not read the data. |
| 0 | RO | **Output Buffer Full**<br>0: The Output Buffer is Empty.<br>1: The Output Buffer is Full. The controller has placed data into the output buffer but the system has not yet read data. |

### 3.8.2 INPUT/OUTPUT BUFFER

#### 3.8.2.1 Input Buffer

The input buffer is an 8-bit write-only register located at address hex 60 or 64 of the I/O space. Writing to address hex 60 would set the bit 3 of status register to '0', which indicates a 'data byte'; writing to address hex 64 would set the bit 3 of status register to '1', which indicates a 'command byte'. 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's BIOS command. Data should be written to the input buffer only if the input buffer's full bit in the status register equals to '0'.

#### 3.8.2.2 Output Buffer

The output buffer is an 8-bit read-only register only located at address hex 60 of the I/O space. The keyboard controller would place the codes received from the keyboard or the return values of the KBC BIOS commands into the output buffer. The output buffer should be read only when output buffer's full bit in the status register equals to '1'.

### 3.8.3 INTERNAL OUTPUT PORT DEFINITIONS

The Internal Output Port includes two signals, P20 and P21, which are defined as following:

| NAME | DEFINITIONS |
|---|---|
| P20 | **RC – Keyboard Hardware Reset Control Signal** |

| | |
|---|---|
| | 1 : Inactive ( default )<br>0 : Active |
| P21 | **GA20 – Gate 20 Control Signal**<br>1 : Inactive ( default )<br>0 : Active |

### 3.8.4 KEYBOARD INTERNAL BIOS COMMANDS ( I/O PORT 64H )

| Command | Keyboard Mode | Keyboard PS/2 Mode |
|---|---|---|
| 01 – 1F | Read Internal RAM – The controller sends value of RAM to output buffer. | |
| 20 (00) | Read Keyboard Controller's Command byte – The controller sends the current Command Byte to its output buffer. | |
| 21 – 3F | Read Internal RAM – The controller sends value of RAM to output buffer. | |
| 41 – 5F | Write Internal RAM – The next byte of data written to I/O 60h is placed into Internal RAM. | |
| 60 (40) | 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 | 1 – Enable Keyboard Output-Buffer-Full Interrupt<br>The KBC would generate an interrupt when it places keyboard data into its output buffer. |
| | 1 | 1 – Enable Mouse-Buffer-Full Interrupt<br>The KBC would generate an interrupt when it places mouse data into its output buffer. |
| | 2 | 1 – The controller generates a System Flag.<br>The value written to this bit is placed in the system flag bit of the controller's status register. |
| | 3 | 1– Disable Keyboard lock Switch "KLOCK#" (Even when the KLOCK# enable bit equals 1). |
| | 4 | 1 – Disable Keyboard.<br>Disable the Keyboard interface by driving the 'keyboard clock' line low. Data won't be sent or received. |
| | 5 | 1 – Disable Mouse.<br>Disable the Mouse interface by driving the 'mouse clock' line low. Data won't be sent or received. |
| | 6 | 1 – IBM Personal Computer Compatibility Mode.<br>Convert the scan codes received from keyboard to IBM PC. This includes converting a two-byte sequence to the one-bye IBM Personal Computer format. |
| | 7 | 0 – Reserved. |
| 61 – 7F | Write Internal RAM – The next byte of data written to I/O 60h is placed into Internal RAM. | |

| | | |
|---|---|---|
| A0 | Read Internal ROM – The controller would convert the scan codes 'E0h' and 'FFh' into Kscan codes, and place the result in the output buffer. | |
| A1 | Read Keyboard Controller's Version – The version code (48h) will be placed to the output buffer. | |
| A4 | Reset Internal Register B to 00h. | Test Password Installed – The controller would send test result to its output buffer.<br>FAh – The Password has been installed.<br>F1h – The Password is not installed. |
| A5 | Reset Internal Register B to 01h. | Load Password – The next byte of data written to I/O 60h is placed into password stream (Max 15 bytes) that ends with "00h". The password is stored in scan code format. |
| A6 | Read Internal Register B – The controller sends value to the output buffer. | Enable Password Security – The keyboard data won't be sent to output buffer until the input character string matches the pre-loaded password. |
| A7 | Set Internal Register C to 00h. | Disable Mouse Device – The mouse interface is disabled by driving the mouse clock line low. |
| A8 | Set Internal Register C to 01h. | Enable Mouse Device – The mouse interface is enabled by driving the mouse clock line floating. |
| A9 | Read Internal Register C – The controller sends value to the output buffer. | ( Not Implemented in 5595)<br>Reference: Mouse Device Interface Test – This command tests the controller's mouse clock and data line, and place the result to output buffer as follows :<br>00 – No error detected.<br>01 – The 'Mouse Clock" line is stuck low.<br>02 – The 'Mouse Clock' line is stuck high.<br>03 – The 'Mouse data' line is stuck low.<br>04 – The 'Mouse data' line is stuck high. |
| AA | Self-Test – This command would result in the internal reset and test of KBC. If the test is successful, the value 55h will be placed in the output buffer. | |

| | | |
|---|---|---|
| AB | ( Not Implemented in 5595)<br>Reference: Keyboard Interface Test – This command tests the keyboard clock and data line, and the test result is placed in the output buffer as follows :<br>00 – No Error detected.<br>01 – The 'Keyboard Clock' line is stuck low.<br>02 – The 'Keyboard Clock' line is stuck high.<br>03 – The 'Keyboard Data' line is stuck low.<br>04 – The 'Keyboard Data' line is stuck high. | |
| AD | Disable Keyboard Interface – This command would set the bit 4 of the controller's command byte, and disable the keyboard interface by driving the clock line low. Data will not be sent or received. | |
| AE | Enable Keyboard Interface – This command would clear the bit 4 of command byte and release the keyboard interface. | |
| 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 Internal Output Port – The value of P20 and P21 would be placed in the bit 0 and bit 1 of the output buffer respectively. This command should be issued only if the output buffer is empty. | |
| D1 | Write Internal Output Port – The bit 0 and bit 1 of the next byte written to I/O 60h would be placed in the P20 and P21 respectively. The other bits are reserved and should be written to '1'. | |
| D2 | Write Keyboard Output Buffer – The next byte of data written to I/O port 60h is placed in output buffer as it received from keyboard. | |
| D3 | Not Valid | Write Mouse Output Buffer – The next byte of data written to I/O port 60h is placed in output buffer as it received from mouse. |
| D4 | Not Valid | Write Mouse Device – The next byte written to I/O 60h is transmitted to mouse device. |
| D6 | Enable Keyboard Lock < KLOCK#> Switch (Default: Enable) – The KLOCK# function will work if the KLOCK# function enable bit is enabled. | |
| D7 | Disable Keyboard Lock Switch < KLOCK# > – The KLOCK# function won't work even when the KLOCK# function enable bit is enabled. | |

### 3.8.5 PASSWORD SECURITY AND KEYBOARD POWER UP FUNCTIONS

Traditional password security function in the KBC has provided the ability of protecting your PC from being invaded. SiS5595/KBC supports the function with registers to store the password character string, and a pattern recognition circuitry to identify if an input character string matches the pre-loaded password.

Writing A4h to I/O port 64h directs the SiS5595/KBC to store the successive data written through I/O port 60h into the password string register until character 00h is input. '00h' is regarded as the end code of the password. Enabling the password security is done by programming I/O port 64h with data A6h. Once being initiated in this mode, the KBC won't issue IRQ1 to invoke the KB interrupt service routine until the input string matches the password. Thus, the system can be protected from invading to some degree.

Moreover, a more user friendly interface or higher security service has led to the request of the password security power up function. A specific power pin, called KBVDD is allocated for the whole KBC circuitry in the SiS5595 to support the function. A power conversion circuit is required to forward power to the SiS5595's KBC, and the keyboard. During the powered-down state of the power rails (including 3.3V, ±5V, ±12V), AUX5V supplies power to the KBC, and the keyboard. Upon the moment that VDD is gone away, the pattern recognition circuitry inside KBC is automatically configured to process the input keyboard string. Once the input string matches the pre-loaded password, the PS_ON# is asserted to turn on the ATX power supply if the password security power up function is enabled.

The following illustrates a procedure to regulate the password security and hot-key power up functions in application. It is highly recommended that the BIOS programmers could follow the sequence in the procedure. For the simplicity and readability, the events that the power up function is altered in the sequence are not included in our procedure. For example, the system was powered up by pressing password and the user want to power up the system by hot-key next time. However, such events are practicable.

#### 3.8.5.1 Procedure Regulating the Password Security and Hot-key Power up Functions

```
If (!CMOS_Valid)                                    /* Initialize all CMOS setting */
        Initialize( );

else { if PSUP_SET                                  /* User has enabled KB password security
            Password_Check( );                        power up function */
        else if HKUP_SET                            /* User has enabled KB hot-key power up
            Hotkey_Check( );                          function */
        else                                        /* PSUP_SET and HK_SET are software
            Password_or_Hotkey_Setting( );            variable used to regulate the functions */
    }
Continue BIOS Posting;

Initialize( )
 {Clear PSUP_SET;
  Clear HKUP_SET;
  Password_or_Hotkey_Setting( );
 }

Password_Check( )
{If KBP_LST                                         /* Keyboard power had ever lost */
   {Clear KPR_LST;                                  /* Enter second level security protection */
        Request the user to enter the password string [*1] and identify it;
        Reload password string into KBC;
        Enable PSUP_EN ;
        Disable BTNUP_EN; [*2]
   }
        else  Reload password string into KBC ; /* Keyboard power is not lost and the
                                                   system is powered up by KBC */
}
```

```
Hotkey_Check ( )
{If KPR_LST
        {Clear KPR_LST;
        Enable HKUP_EN;
        Disable BTNUP_EN;
        }
}

Password_or_Hotkey_Setting( )
{Clear KPR_LST;
If (password security power up function is selected [*3])
        {Set and load KB password into KBC;
        Enable PSUP_EN;
        Enable PSUP_SET;
        Disable BTNUP_EN;
        }
if ( hot-key power up function is selected [*3] )
        {Enable HKUP_EN;
         Enable HKUP_SET;
         Disable BTNUP_SET;
         }
}
```

[*1] Typically, the password string is stored in the RTC CMOS RAM.

[*2] It is apparent that BTNUP_EN should be disabled if password security or hot-key power up function is enabled. Otherwise, the system can be powered up simply by pressing the Power-Button for over 30ms. However, BTNUP_EN will be automatically enabled to allow manually power up the system if the KBC power has ever lost. The KBP_LST locating in the RTC APC register file is designed to reflect the KBC power status. When KBC power has ever lost, KBP_LST is set with the result that BTNUP_EN is set, and that KBC is held in “CLEAR” state when keyboard power is restored later. Both PSUP_EN and HKUP_EN are cleared also. Ending the KBC clear state can be done by writing logic 0 to KPR_LST.

[*3] User can select the password security or hot-key power up function normally through setting the corresponding option in the BIOS setup menu. As a result, PSUP_SET or HKUP_SET is set. If the user wants to change the enabled power up function in the sequence, this setting menu can do it also.

The following bits are used to support the password security or hot-key power up function. They are placed in the RTC module, and thus are retained as long as RTC power is there.

KBP_LST : APC 07h[0]

BTNUP_EN : APC 03h[7]

PSUP_EN : APC 05h[6]

HKUP_EN : APC 05h[5]

### 3.8.6 RELATED PCI TO ISA BRIDGE CONFIGURATION REGISTERS

R64h bit 0: Enable Keyboard Hardware Reset

R6Dh bit 1: Keyboard Hot-key Status

R6Dh bit 0: Keyboard Hot-key Control

R70h bit 4: Enable KLOCK# Function

R70h bit3: Enable Built-in Keyboard Controller

R70h bit2: Enable PS/2 Mouse Mode

## 3.9 ISA BUS INTERFACE

### 3.9.1 ISA BUS CONTROLLER

The SiS5595'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 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 buffering are also integrated in SiS5595. The SiS5595 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 refresh 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.

### 3.9.2 DMA CONTROLLER

The SiS5595 contains a seven-channel DMA controller. Channels 0 to 3 are for 8-bit DMA devices while channels 5 to 7 are 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 SiS5595 can support 32-bit address for DMA devices.

### 3.9.3 INTERRUPT CONTROLLER

The SiS5595 provides an ISA compatible interrupt controller that incorporates the functionality of two 8259 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 provides IRQ<15:8>. The two internal interrupts 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.

**Table 3.9-1 8259 IRQs Mapping**

| 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.

### 3.9.4 INTERRUPT STEERING

For each interrupt channel, an interrupt router is associated, serving as an interface between bunches of the interrupt request lines and the 8259 interrupt controller as shown in figures below. These routers can be classified into two categories, one for the IRQ [7:3], IRQ [12:9], and IRQ [15:14], and one for the IRQ0, IRQ1, IRQ8, and IRQ13. The following interrupt request lines can be routed to the IRQx of the first category.

Illustrates the structure of the Interrupt Router.

1) PCI Interrupt INT [A:D]#, and SIRQ [A:D]# through programming PCI to ISA register 41-44h,
2) IDE Interrupt request line through programming PCI to ISA register 61h,
3) USB Interrupt request line through programming register PCI to ISA 62h,
4) ACPI/SCI interrupt request line through programming register PCI to ISA 6Ah,
5) Data Acquisition and SMBus Interrupt request line via programming register PCI to ISA 7Eh,
6) SIRQx interrupt request line via programming PCI to ISA registers 89h, 8Ah, where x can be [7:3], [12:9], and [15:14].

Except the SIRQx, the rest of the interrupt requests are regarded to be sharable. That is, more than one line of interrupt request can be routed to the same IRQx. While supporting the shared interrupts, the associated IRQ channel must be set in the level sensitive mode. Enabling any of the routing registers will automatically mask the ISA Interrupt request to the corresponding IRQ. SIRQ [A:D]# are regarded to work in the level sensitive mode, while SIRQx in the level or edge sensitive mode. While configuring the SIRQ [A:D]# to any of the

IRQx, in addition to the enabling of PCI to ISA Register 41h to 44h, the BIOS should also set bit 3 to bit 6 of PCI to ISA Register 8Ch for each SIRQ, respectively.

SIRQ1 can be routed to IRQ1 while bit 0 of PCI to ISA register 89h is set.

SIRQ8 can be routed to IRQ8 while bit 7 of PCI to ISA register 89h is set.

SIRQ13 can be routed to IRQ13 if bit 4 of PCI to ISA register 8Ah is set.

ACPI/SCI can also be routed to IRQ13 via programming PCI to ISA register 6Ah.

To support some super I/O devices that don't keep their SIRQ1 (or 12) active until the INTR to the CPU is driven active, SIRQ1 (or SIRQ12) will be internally latched on its rising edge if bit 7 (or bit 6) of register 64h is set, respectively. Reading port 60h clears the latch. Optionally, setting the SIRQ bit of IRQ1 (or IRQ12) can clear the latch if bit 7 of Register 67h is set.

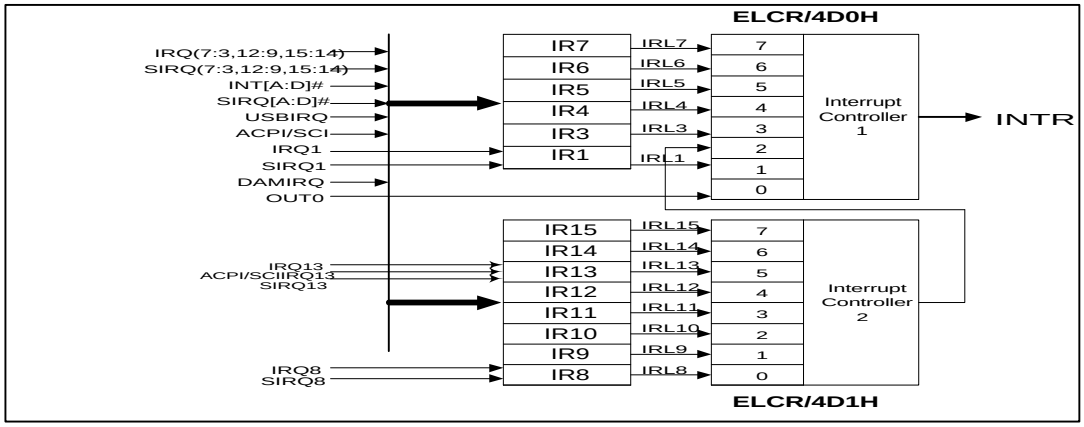


**Figure 3.9-1 Interrupt Router IRx**

Note: ELCR-Edge/Level-triggered Control Register

### 3.9.5 TIMER/COUNTER

The SiS5595 contains 3 counter/timers that are equivalent to those found in the 8254 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.

## 3.10 SYSTEM RESET

### Power-on Reset

When the system is initially powered, the power supply must wait until all voltages are stable for at least one millisecond, and then assert PWRGD signal. While PWRGD is deasserted, chipset must hold its PCI bus in reset. While power-on reset is asserted, chipset will reset and initialize their internal registers. Chipset must also initialize their PCI busses by asserting PCIRST# for a minimum of one millisecond. For Pentiumn II processor in power-on

configuration, the core logic must assert BREQ0# before the clock in which CPURST# is deasserted. BREQ0# must be deasserted by core logic in the clock after CPURST# is sampled deasserted. CPU agent 0 must delay BREQ0# assertion for a minimum of three clocks after the clock in which CPURST# is deasserted to guarantee wire-or glitch free operation. This sequence of events is shown following.

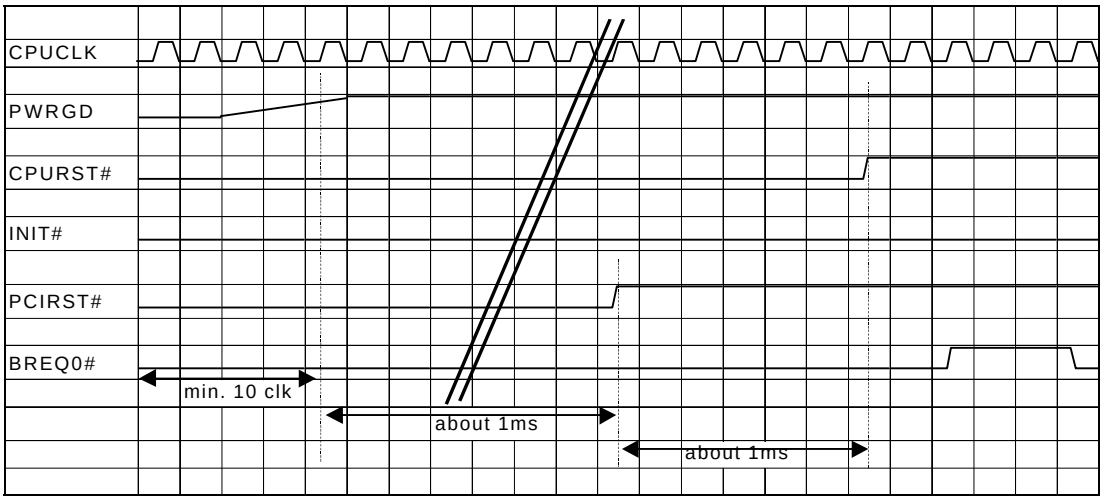


**Figure 3.10-1 Timing Sequence for Power-on Process**

### Soft Reset (INIT)

The SiS5595 can be programmed to deliver a soft reset to processors through the Reset control register. A soft reset asserts the INIT signal for 2us. The INIT signal resets all processors without effecting their internal caches or bus state machines. After soft reset, the processors begin to execute from address 00_FFFF_FFF0h. SiS5595 devices are not effected by the INIT signal. SiS5595 also can be programmed to deliver a CPU reset to processors through the Reset control register. The timing is the same as INIT.

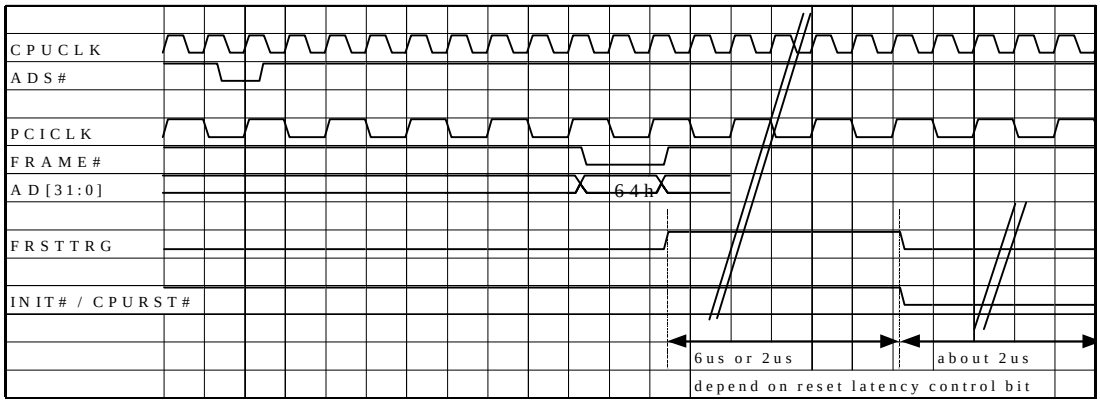


**Figure 3.10-2 Timing for Generating INIT#/CPURST#**

# 4 PIN DESCRIPTIONS

## 4.1 PCI/ISA BUS INTERFACE

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 142, 137, 139, 134, 135, 140, 131, 133, 196, 197, 200, 202, 203, 206, 207, 208 | SD[15:0] | I/O | **ISA Data bus:**<br>SD[7:0] is the lower byte of ISA data bus.<br>SD[15:8] is the upper byte of ISA data bus. |
| 179, 178, 177, 175, 170, 171, 167, 166, 165, 164, 163, 161, 160, 156, 155, 153, 147 | SA[16:0] | I/O | **ISA Address bus:**<br>External SA[19:17] with SA[16:0] allow memory space addressing up to 1 Mbytes.<br>SA[15:0] allow IO space addressing up to 64Kbyte. |
| 190, 189, 187, 186, 185, 184, 183 | LA[23:17] | I/O | **ISA LA[23:17] address bus:**<br>LA[23:17] allow memory space addressing up to 16 Mbytes. |
| 181 | SBHE# | I/O | **System Byte High Enable:**<br>When asserted, it indicates the upper byte of the ISA data bus, i.e. SD[15:8], contains valid data. |
| 149 | IORC# | I/O | **ISA bus I/O Read Command:**<br>IORC# is asserted to indicate the addressed IO device should drive its data onto the ISA data bus. This pin is input during ISA master cycles and output otherwise. |
| 148 | IOWC# | I/O | **ISA bus I/O Write Command:**<br>IOWC# is asserted to strobe data into the addressed IO device. This pin is input during ISA master cycles and output otherwise. |
| 193 | MRDC# | I/O | **ISA bus Memory Read Command:**<br>MRDC# is asserted to indicate the addressed memory device should drive its data onto the ISA data bus. This pin is input during ISA master cycles and output otherwise. |

| | | | |
|---|---|---|---|
| 195 | MWTC# | I/O | **ISA bus Memory Write Command:**<br>MWTC# is asserted to strobe data into the addressed memory device. This pin is input during ISA master cycles and output otherwise. |
| 145 | SMRDC# | O | **ISA bus Memory Read Command—address below 1Mbyte:**<br>For memory addresses within 000000h-0FFFFFh, it is asserted to indicate the addressed memory device should drive its data onto the ISA data bus. |
| 144 | SMWTC# | O | **ISA bus Memory Write Command—address below 1 Mbytes:**<br>For memory addresses within 000000h-0FFFFFh, it is asserted to strobe data into the addressed memory device. |
| 176 | BALE | O | **Bus Address Latch Enable:**<br>BALE indicates ISA address bus, LA[23:17], SA[16:0] and SBHE#, contain valid ISA address. |
| 180 | IO16# | I | **16-bit I/O Chip Select:**<br>IO16# is asserted by the addressed IO device to indicate that it is capable of doing 16-bit data transfers. |
| 182 | M16# | I/O | **16-bit Memory Chip Select:**<br>M16# is asserted by the addressed memory device to indicate that it is capable of doing 16-bit data transfers. |
| 159 | RFH# | I/O | **Refresh:**<br>RFH# is asserted during ISA refresh cycles. This pin is input during ISA master cycles and output otherwise. |
| 136 | ZWS# | I | **Zero Wait State:**<br>ZWS# is asserted by the addressed device to indicate that the current ISA cycle can be terminated without any additional wait-states. |
| 141 | IOCHRDY | I/O | **I/O Channel Ready:**<br>IOCHRDY is de-asserted by the addressed slave device to indicate more wait-states are required to complete the current ISA transaction. This pin is output during ISA master and DMA cycles and input otherwise. |
| 138 | AEN | O | **Address Enable:**<br>Address Enable is driven high during DMA and refresh cycles. |

| | | | |
|---|---|---|---|
| 162 | BCLK | O | **ISA Bus Clock:**<br>The ISA bus clock can be optionally derived from PCI clock or the CLK14M input. |
| 173 | TC | O | **Terminal Count of DMA:**<br>A high pulse on TC during DMA cycles indicates one of the DMA channels has completed its transfers. |
| 191, 158, 132, 152, 199, 201, 204 | DRQ[3:0]<br>DRQ[7:5] | I | **DMA Request:**<br>They will be asserted by external ISA devices to request for DMA service or become ISA masters. |
| 192, 157, 169, 154, 194, 198, 205 | DACK[3:0]#<br>DACK[7:5]# | O | **DMA Acknowledge:**<br>When one of them is asserted, it indicates the corresponding DMA request has been granted. |
| 143, 146, 8, 9, 2, 3, 4, 5, 7, 6 | IRQ[15:14],<br>IRQ[11:9],<br>IRQ[7:3] | I | **Interrupt Request:**<br>These are interrupt requests input to the internal 8259-compatible Interrupt Controller. |

### 4.2 PCI BUS + CPU INTERFACE

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 102, 91, 82, 76 | C/BE[3:0]# | I/O | **PCI Bus Command and Byte Enables:**<br>Comply with PCI specification 2.1 |
| 87 | FRAME# | I/O | **PCI FRAME#: Comply with PCI specification 2.1** |
| 112, 111, 110, 108, 107, 106, 105, 103, 101, 100, 98, 97, 96, 95, 94, 93, 80, 79, 78, 77, 75, 74, 72, 71, 70, 69, 68, 67, 66, 65, 63, 61 | AD[31:0] | I/O | **PCI Address/Data Bus:**<br>Comply with PCI specification 2.1 |
| 92 | IRDY# | I/O | **PCI IRDY#: Comply with PCI specification 2.1** |
| 86 | TRDY# | I/O | **PCI TRDY#: Comply with PCI specification 2.1** |
| 58 | PCLK | I | **PCI Clock: Comply with PCI specification 2.1** |
| 83 | SERR# | I | **System Error:**<br>When sampled active low, a non-maskable interrupt (NMI) can be generated to CPU if enabled. |

| | | | |
|---|---|---|---|
| 89 | DEVSEL# | I/O | **Device Select:**<br>SiS5595 will do positive decoding with medium timing to:<br>- SiS5595 PCI configuration registers access<br>- Built-in legacy ISA bus embedded controller registers access<br>- BIOS ROM memory access<br>- Interrupt acknowledge cycle<br>In addition, SiS5595 will do subtractive decoding to:<br>- I/O address range 00000000h-0000FFFFh.(Low 64 Kbytes)<br>- Memory address range 00000000h-00FFFFFFh (Low 16 Mbytes). |
| 84 | STOP# | I/O | **PCI STOP#: Comply with PCI specification 2.1** |
| 113 | PHOLD# | O | **PCI Bus Hold Request:**<br>PHOLD# is asserted to inform the PCI system arbiter located at north-bridge chipset that SiS5595 is intending to become PCI bus master. SiS5595 asserts PHOLD# on behalf of three local devices including ISA/DMA master, Distributed DMA master and USB master. |
| 125 | PHLDA# | I | **PCI Bus Hold Acknowledge:**<br>The PCI system arbiter responds to the assertion of PHOLD# by driving PHLDA# low, indicating SiS5595 can start its PCI master cycles. |
| 56, 57, 59, 60 | INT[A:D]# | I | **PCI interrupt A,B,C,D:**<br>The PCI interrupts will be connected to the inputs of the internal Interrupt controller through the rerouting logic associated with each PCI interrupt. |
| 126 | PCIRST# | OD | **PCI Bus Reset:**<br>PCIRST# will be asserted during the period when PWRGD is low, and will be kept on asserting until about 1 ms after PWRGD goes high. |
| 151 | PDMAREQ0# | I | **PC/PCI DMA Request 0:** This signal is used by PCI agent to request DMA services. |
| 130 | PDMAGNT0# | O | **PC/PCI DMA Grant 0:** This signal is used by 5595 to indicate which DMA channel is granted. |
| 47 | NMI | OD | **Non-Maskable Interrupt:**<br>A rising edge on NMI will trigger a non-maskable interrupt to CPU. This signal requires an external pull-up resistor tied to 3.3V. |

| | | | |
|---|---|---|---|
| 48 | INTR | OD | **Interrupt Request:**<br>At high-level voltage on this signal indicates to the CPU that there is outstanding interrupt(s) need to be serviced. This signal requires an external pull-up resistor tied to 3.3V. |
| 46 | IGNE# | OD | **Ignore Error:**<br>IGNE# is asserted to inform CPU to ignore a numeric error. This signal requires an external pull-up resistor tied to 3.3V. |
| 43 | FERR# | I | **Floating Point Error:**<br>CPU will assert this signal upon a floating point error occurs. |
| 44 | STPCLK# | OD | **Stop Clock:**<br>STPCLK# will be asserted to inhibit or throttle CPU activities upon a pre-defined power management event occurs. It requires an external pull-up resistor tied to 3.3V. |
| 49 | SMI# | OD | **System Management Interrupt:**<br>SMI# will be asserted upon a pre-defined power management event occurs. It requires an external pull-up resistor tied to 3.3V. |
| 50 | CPURST | OD | **CPU Reset:**<br>Active high signal to reset CPU. It requires an external pull-up resistor tied to 3.3V. |
| 45 | INIT | OD | **Initialization:**<br>INIT is used to re-start the CPU without flushing its internal caches and registers. In Pentium platform it is active low, while in Pentium II platform it is active high. This signal requires an external pull-up resistor tied to 3.3V. |
| 51 | A20M# | OD | **Address 20 Mask:**<br>When A20M# is asserted, the CPU A20 signal will be forced to '0'. It requires an external pull-up resistor tied to 3.3V. |

Note: "OD" means open drain signal.

### 4.3 KBC + MISC.

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 31 | GPIO2/<br>KBCLK | I/O | **General Purpose Input/ Output 2:**<br>When the internal keyboard controller is disabled, this pin is used as GPIO2. As input, it can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**Keyboard Clock:**<br>When the internal keyboard controller is enabled, this pin is used as the keyboard clock signal. |
| 33 | KBDAT/<br>IRQ1 | I/O | **Keyboard Dada:**<br>When the internal keyboard controller is enabled, this pin is used as the keyboard data signal.<br>**IRQ1:**<br>When the internal keyboard controller is disabled, this pin is used as the IRQ1 signal. |
| 32 | GPIO1/<br>PMCLK | I/O | **General Purpose Input/ Output 1:**<br>When the internal PS2 mouse controller is disabled, this pin is used as GPIO1. As input, it can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**PS2 Mouse Clock:**<br>When the internal keyboard and PS2 mouse controllers are enabled, this pin is used as the PS2 mouse clock signal. |
| 34 | PMDAT/<br>IRQ12 | I/O | **PS2 Mouse Data:**<br>When the internal keyboard and PS2 mouse controllers are enabled, this pin is used as PS2 mouse data signal.<br>**Interrupt Request 12:**<br>When the internal PS2 mouse controller is disabled, this pin is used as the IRQ12 signal. |

| | | | |
|---|---|---|---|
| 128 | GPIO3/ CPU_STOP# / SLP# | I/O | **General Purpose Input/ Output 3:**<br>As input, GPIO3 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**CPU_STOP#: (For Pentium platform)**<br>It is asserted to inform the system Clock Generator chip to stop the CPU and SDRAM clock outputs as well as to disconnect ISA bus from power supply, as defined in the power management S2 state.<br>**SLP#: (For Pentium-II platform)**<br>It is asserted to put the Pentium II processor into SLEEP state. |
| 53 | GPIO0/PAR | I/O | **General Purpose Input/ Output 0:**<br>As input, GPIO0 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**PCI Parity:**<br>SiS5595 always generates even-parity on PAR and ignores the parity driven by other PCI agents. |
| 40 | GPIO9/ THERM#/ BTI/ SMBALERT# | I/O | **General Purpose Input/ Output 9:**<br>As input, GPIO9 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**Thermal Detect:**<br>THERM# is connected to the internal ACPI-compatible power management unit as an indication of outstanding thermal event. In response, the STPCLK# signal will be asserted to throttle CPU activities.<br>**Board Temperature Interrupt:**<br>BTI is connected to the internal Data Acquisition Module for interrupt or SMI# generation. It is driven by temperature monitoring chips located on the motherboard.<br>**SMBUS Alert Interrupt:**<br>The SMBALERT# is an interrupt signal for SMBUS device to notify the host it wants to talk. |

| | | | |
|---|---|---|---|
| 127 | BM_REQ# | I | **Bus Mater Request:**<br>This is a serial link from SiS North Bridge chipset carrying the current AGP and PCI bus master information to SiS5595 for power management purpose. Two types of events are defined and can be enabled as a wakeup or reload event to the power management unit:<br>Event 1: AGP activity<br>Event 2: AGP/PCI/IDE master requesting for PCI bus<br>Event 1 can be sampled by SiS55595 at the 1,3,5…Nth PCI clock after FRAME# asserted. Event 2 can be sampled at the 2,4,6…Nth PCI clock after FRAME# asserted. |
| 55 | GPCS0# | I/O | **General Programmable Chip Select 0:**<br>This pin can be programmed through Reg_73h/PMU as one of the GPI, GPO, GPCS# or GPCSW# functions:<br>1) GPI: can be used to generate a reload, wakeup or SMI event<br>2) GPO: its output logic state can be controlled via a register bit.<br>3) GPCS#: active-low output can be used as a chip select signal.<br>4) GPCSW#: similar to GPCS#, the GPCS with Write function further qualifies the IOWC# signal so that it can be used to control an external 8-bit latch connected to SD[7:0], thereby expanding the number of general purpose outputs up to 8. |
| 52 | GPCS1#/<br>KLOCK# | I/O | **General Programmable Chip Select 1:**<br>Refer to GPCS0#.<br>**Keyboard Lock:**<br>When KLOCK# is tied low, the internal keyboard controller will not respond to any key-strikes. |
| 54 | GPIO17/<br>SIRQ | I/O | **General Purpose Input/ Output 17:**<br>As input, GPIO17 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**Serial IRQ:**<br>This signal is used as the Serial IRQ line signal. |

| | | | |
|---|---|---|---|
| 129 | EXTSMI# | I | **External SMI#:**<br>EXTSMI# can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. |
| 35 | CLK14M | I | **14.318 MHz clock input:**<br>This signal provides the fundamental clock for the 8254-compatible timer, PMU System Standby Timer, ACPI Timer and BCLK. |
| 36 | ROMKBCS# | O | **Keyboard or System ROM Chip Select:**<br>ROMKBCS# will be asserted during external keyboard controller or ROM access cycles. |
| 37 | SPKR | O | **Speaker output:**<br>The SPKR is connected to the system speaker. |
| 41 | DDCDAT | OD | **SMBus/$I^2C$ Bus Data:**<br>An external pull-up resister tied to 3.3V is required. |
| 42 | DDCCLK | OD | **SMBus/$I^2C$ Bus Clock:**<br>An external pull-up resister tied to 3.3V is required. |
| 122 | GPIO7/<br>OC0#/<br>PPS | I/O | **General Purpose Input/Output 7:**<br>As input, GPIO7 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**USB Over Current Detection:**<br>OC0# is used to detect the over current condition.<br>**USB Power Switch:**<br>PPS function is used to control the external Power-Distribution Switches logic to power off the USB power supply lines. |
| 124 | GPIO8/<br>OC1# | I/O | **General Purpose Input/Output 8:**<br>As input, GPIO8 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**USB Over Current Detection:**<br>OC1# is used to detect the over current condition. |

| | | | |
|---|---|---|---|
| 38 | GPIO4/<br>FAN1 | I/O | **General Purpose Input/Output 4:**<br>As input, GPIO4 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**Fan Tachometer Input 1:**<br>It is driven from the fan's tachometer output and is connected to the internal Data Acquisition Module for fan speed monitoring. |
| 39 | GPIO11/<br>FAN2 | I/O | **General Purpose Input/Output 11:**<br>As input, GPIO11 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**Fan Tachometer Input 2:**<br>It is driven from the fan's tachometer output and is connected to the internal Data Acquisition Module for fan speed monitoring. |
| 1 | GPIO16/<br>IOCHK# | I/O | **General Purpose Input/Output 16:**<br>As input, GPIO16 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**I/O Channel Check:**<br>IOCHK# can be issued by ISA devices to indicate an error event. In response, SiS5595 will assert the NMI if enabled. |

Note: "OD" means open drain signal.

### 4.4 USB CONTROLLER

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 115<br>116 | UV0+,<br>UV0- | I/O | **USB Data port 0:**<br>Used as the differential USB data bus pair of port 0. |
| 117<br>118 | UV1+,<br>UV1- | I/O | **USB Data port 1:**<br>Used as the differential USB data bus pair of port 1. |
| 120 | UCLK48M | I | **48 MHz USB Clock Input .** |

### 4.5 RTC

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 21 | OSC32KHI/ IRQ8# | I | **32.768 KHz Input:**<br>When internal RTC is enabled, this pin provides the 32.768 KHz clock signal from external crystal or oscillator.<br>**Interrupt Request 8:**<br>When external RTC is enabled, this pin is used as IRQ8#. |
| 20 | OSC32KHO/RTCCS# | O | **32.768 KHz Output:**<br>When internal RTC is enabled, this pin should be connected the other end of the 32.768 KHz crystal or left unconnected if an oscillator is used.<br>**RTC Chip Select:**<br>When external RTC is enabled, this pin is used as the chip select signal for the external RTC. |
| 18 | PWRGD | I | **Power Good:**<br>A high-level input to this signal indicates the power being supplied to the system is in stable operating state. During the period of PWRGD being low, CPURST and PCIRST# will all be asserted until after PWRGD goes to high for 1~2 ms. |
| 23 | PSRSTB# | I | **RTC Power Strobe:**<br>When the internal RTC is enabled, this signal is used as the power strobe signal to initialize RTC internal registers when power is first applied to the system. If the internal RTC is disabled, this signal should be tied low. |
| 25 | PS_ON#/ RTCALE | OD | **ATX Power ON/OFF control:**<br>PS_ON# is internally powered by RTCVDD and is used to control the on/off state of the ATX power supply. When the ATX power supply is in the OFF state, several power management events can be defined to generate a low output on this signal and hence switch the power supply to ON state. These events are PWRBT #, RING, PME0 #, PME1 #, RTC alarm, keyboard password matched and hotkey being pressed.<br>**RTC Address Latch Enabled:**<br>When external RTC is used, this signal can be used to latch the address of RTC internal register being accessed on the ISA SD [7:0] data bus. |

| | | | |
|---|---|---|---|
| 29 | GPIO5/<br>PME0#/<br>DUAL_ON# | I/O | **General Purpose Input/Output 5:**<br>When the system is in power-on mode, this pin is GPIO5. As input, GPIO5 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**PME0#:**<br>When the system is in power-down mode, a high-to-low transition on PME0# will cause the PS_ON# to go low and hence turn on the power supply.<br>**Dual Power On/Off Control:**<br>When the system is in suspend mode, the DUAL_ON# signal can be used to control PC98-compatible power supply. |
| 28 | GPO6/<br>CKE_S/<br>ACPILED | O | **General Purpose Output 6:**<br>The logic state of GPO6 can be controlled via a register bit located at the internal Automatic Power Control (APC) unit.<br>**SDRAM Clock Enable:**<br>When the system is in suspending mode (suspend to DRAM), the CKE_S is driven low to enable the self-refresh mode of SDRAM.<br>**ACPILED:**<br>ACPILED can be used to control the blinking of an LED at the frequency of 1 Hz to indicate the system is at power saving mode. |
| 27 | GPIO10/<br>PME1#/<br>ACPILED | I/O | **General Purpose Input/Output 10:**<br>When the system is in power-on mode, this pin is GPIO10. As input, GPIO10 can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit. As output, its output logic state can be controlled via a register bit.<br>**PME1#:**<br>When the system is in power-down mode, a low pulse over 30 us on PME1# will cause the PS_ON# to go low and hence turn on the power supply.<br>**ACPI LED:**<br>Refer to Pin 28. |

| | | | |
|---|---|---|---|
| 24 | RING | I | **Ring In:**<br>RING is connected to the internal Automatic Power Control (APC) unit and the ACPI-compatible power management unit (PMU). For the APC part, a 4ms active pulse detected on RING will cause the PS_ON# signal to go low and hence turn on the power supply. For the PMU part, it can be configured to generate a SMI#/SCI event, a wake up event or a system standby timer reload event. |
| 26 | PWRBT# | I | **Power Button:**<br>This signal is from the power button switch and will be monitored by the ACPI-compatible power management unit to switch the system between working and sleeping states. |

### 4.6 DATA ACQUISITION INTERFACE

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 13,14,15,16 | GPI[15:12]/ VIN[3:0] | I | **General Purpose Input [15:12]:**<br>GPI [15:12] each can be configured to generate a SMI#/SCI event, a Wakeup event or a Standby Timer reload event to the ACPI-compatible power management unit.<br>**VIN [3:0]:**<br>VIN [3:0] are connected to the Data Acquisition Module for system supplied voltages monitoring. They also can be served as temperture detection by connecting to one external thermister. |
| 10 | DXP/VIN4 | A | **DXP:**<br>When the transistor 2N3904 is employed as temperature sensor, this pin should be connected to the base and collector of 2N3904.<br>When thermister sensor is used, this pin should be connected to the one end of the thermister. The other end of the thermister should be connected to AGND (pin 17).<br>**VIN4:**<br>VIN4 are connected to the Data Acquisition Module for system supplied voltages monitoring. |
| 11 | DXN | A | **DXN:**<br>For 2N3904 sensor, this pin should be tied to emitter of the transistor.<br>For thermister sensor, this pin should be connected to VSS. |

Note: “A” means Analog signal.

### 4.7 POWER PINS

| SiS5595 PIN NO. | NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|---|
| 119 | USBVDD | PWR | **+3.3V DC Power for USB circuit.** |
| 114 | USBVSS | PWR | **Ground pin for USB circuit** |
| 22 | RTCVDD | PWR | **Power pin for internal RTC and APC.** |
| 19 | RTCVSS | PWR | **Ground pin for internal RTC and APC** |
| 12 | AVDD | PWR | **5V DC power for the Data Acquisition circuitry.** |
| 17 | DVSS | PWR | **Ground pin for the Data Acquisition circuitry.** |
| 30 | KBVDD | PWR | **Keyboard Controller Power Input.** If keyboard password security or hot key power-up function is enabled, the KBVDD should be connected to AUX-5V of the ATX power supply. The system designers are advised to ensure the power supply in use will provide enough current on the AUX-5V line to the keyboard in order for these functions to work successfully. Typically, a keyboard will consume up to 200~300mA of current and the actual values may be varied for different keyboard manufacturers. |
| 62, 81, 104, 168 | OVDD | PWR | **+5V DC I/O PAD power.** |
| 64, 73, 85, 99,109,150, 188 | OVSS | PWR | **Ground pin for I/O DC PAD power.** |
| 88,121, 174 | DVDD | PWR | **+5V DC main power supply.** |
| 90,123, 172 | DVSS | PWR | **Ground pin for main voltage supply.** |

## 5 HARDWARE TRAP

The ROMKBCS#, PSRSTB# and PHLDA# pins can be used to configure SiS5595 during system boot-up. The SiS5595's operating mode will be determined by the voltage-level being applied to these pins when the PWRGD signal is going from low to high, known as Hardware Trap. A logic “1” will be recognized and trapped into internal control circuitry if an external pull-up resistor is connected to the trap pin, while a logic “0” will be trapped if a pull-down resistor is connected. The PHLDA# is driven high by North Bridge when North Bridge is 530 during the rising edge of PWRGD, but a low is driven by 5600/620 (North Bridge) instead. The PHLDA# is also used as a selection for Pentiumn II processor core frequency configuration during the falling edge of CPURST.

| SiS5595 PIN NO. | SYMBOL | DESCRIPTION |
|---|---|---|
| 36 | ROMKBCS# | **Enable/Disable Internal PCI Clock DLL Circuitry to Improve Timing**<br>Pull-up: Disable<br>Pull-down: Enable |
| 23 | PSRSTB# | **Enable/Disable the Built-in RTC**<br>Pull-up: Use Internal RTC.<br>Pull-down: Use External RTC |
| 125 | PHLDA# | **CPURST and INIT active level selection**<br>Driven-high: Both are high level active, for Pentiumn processor<br>Driven-low: Both are low level active, for Pentiumn II processor |

For Pentium-II platform, there are four output pins of SiS5595--A20M#, INTR, IGNE# and NMI, are used to provide core operating frequency information to the CPU when CPURST is going from high to low. To achive this, the other four pins BM_REQ#, GPCS0#, PHOLD# and PHLDA# are used for A20M#, INTR, IGNE# and NMI trapping control. The trapping circuit is active when CPURST is high and till 7 PCI clocks later after CPURST goes from high-to-low. The PHOLD# and GPCS0# pins will be forced to input mode during this period, while BM_REQ# and PHLDA# are input only.

The 5595B also provides another method to control A20M#, INTR, IGNE# and NMI. It employed the APC register 08h, bit 7~4 to control these four pins. The APC register 08h, bit 3 is utilized to determine which method is applied. The APC Reg. 08h will be cleared to default value if the BIOS is unable to set the PMU Reg. 4Ch, bit 0 to ‘1’ in 4.68 seconds after CPURST deassertion.

Method I: APC Reg. 08h, bit 3 is set to 0

| SiS5595 PIN NO. | PIN NAME | DESCRIPTION |
|---|---|---|
| 127 | BM_REQ# | **Routed to A20M# for Pentium-II Core Frequency Selection** |
| 55 | GPCS0# | **Routed to INTR for Pentium-II Core Frequency Selection** |
| 113 | PHOLD# | **Routed to IGNE# for Pentium-II Core Frequency Selection** |
| 125 | PHLDA# | **Routed to NMI for Pentium-II Core Frequency Selection** |

Method II: APC Reg. 08h, bit 3 is set to 1

| Bit# of APC Reg. 08h | DESCRIPTION |
|---|---|
| 5 | **Routed to A20M# for Pentium-II Core Frequency Selection** |
| 6 | **Routed to INTR for Pentium-II Core Frequency Selection** |
| 4 | **Routed to IGNE# for Pentium-II Core Frequency Selection** |
| 7 | **Routed to NMI for Pentium-II Core Frequency Selection** |

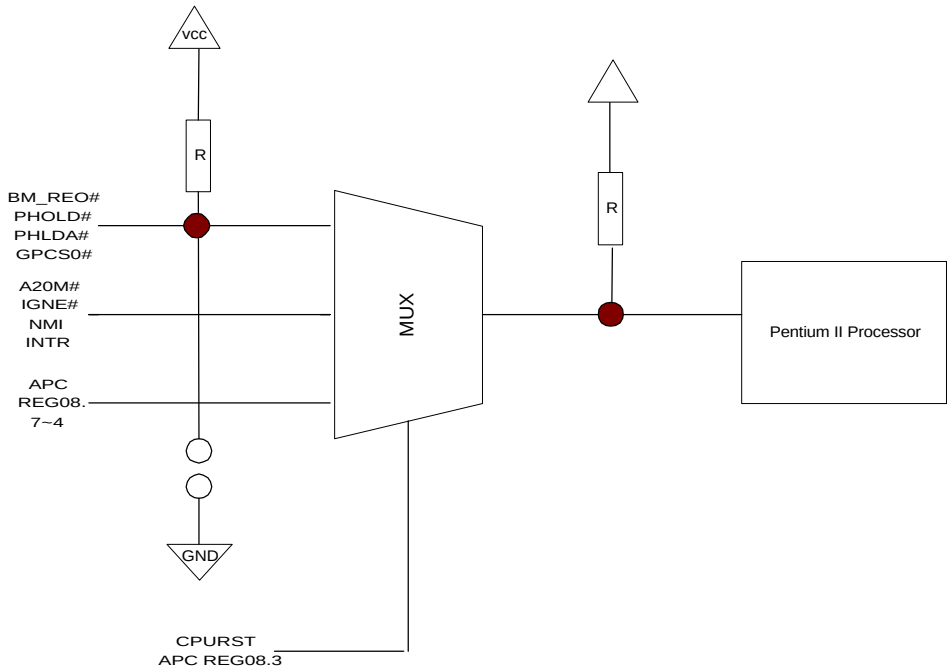


**Figure 5-1 Block Diagram for Generating Core Frequency**

# 6 REGISTER SUMMARY

## 6.1 PCI TO ISA BRIDGE CONFIGURATION REGISTERS

| ADDRESS | ACCESS | REGISTER NAME |
|---|---|---|
| 00-01h | RO | Vendor ID |
| 02-03h | RO | Device ID |
| 04-05h | R/W | Command Register |
| 06-07h | R/W | Status register |
| 08h | RO | Revision ID |
| 09-0Bh | RO | Class Code |
| 0Ch | RO | Cache Line Size |
| 0Dh | R/W | Master Latency Timer |
| 0Eh | RO | Header Type |
| 0Fh | RO | Built-in Self Test |
| 10-3Ch | RO | - Reserved |
| 40h | R/W | BIOS Control Register |
| 41-44h | R/W | PCI INTA#/B#/C#/D# Remapping Register |
| 45h-46h | R/W | ISA Bus Control Register I, II |
| 47h | R/W | DMA Clock and Wait State Control Register |
| 48h | R/W | ISA to PCI Top of Memory Region Register |
| 49h | R/W | ISA to PCI Memory Region Enable Register |
| 4Ah | R/W | ISA to PCI Memory Hole Bottom Address Register |
| 4Bh | R/W | ISA to PCI Memory Hole Top Address Register |
| 4C-4Fh | RO | Shadow Register of ICW1 to ICW4 of the INT1 |
| 50-53h | RO | Shadow Register of ICW1 to ICW4 of the INT2 |
| 54-55h | RO | Shadow Register of OCW 2&3 of INT1 |
| 56-57h | RO | Shadow Register of OCW 2&3 of the INT2 |
| 58h-5Fh | RO | CTC Shadow Registers 1 to 8 |
| 60h | RO | Shadow Register for ISA Port 70 |
| 61h | R/W | IDEIRQ Remapping Register |
| 62h | R/W | USBIRQ Remapping Register |
| 63h | R/W | PCI Output Buffer Current Strength Register |
| 64h | R/W | INIT Enable Register |
| 65h | R/W | PHOLD# Timer |
| 66h | R/W | Priority Timer |
| 67h | R/W | Respond to C/D Segment |
| 68-69h | R/W | Data Acquisition Module Base Address Register |
| 6Ah | R/W | ACPI/SCI IRQ Remapping Register |
| 6B-6Ch | R/W | Test Mode Register I, II |

| | | |
|---|---|---|
| 6Dh | R/W | $I^2C$ Bus Control Register |
| 6E-6Fh | R/W | Software-Controlled Interrupt Requests |
| 70h | R/W | Misc. Control Register |
| 71h | R/W | - Reserved |
| 72h | R/W | Individual PC/PCI DMA Channel Enable Register |
| 74-79h | R/W | - Reserved |
| 7Ah | R/W | Data Acquisition Module Function Selection Register |
| 7Bh | R/W | Data Acquisition Module Control Register |
| 7C-7Dh | R/W | Data Acquisition Module ADC Calibration Register |
| 7Eh | R/W | Data Acquisition Module IRQ Remapping Register |
| 7Fh | RO | - Reserved |
| 80-81h | R/W | Distributed DMA Master Configuration Register |
| 82-83h | RO | - Reserved |
| 84h | R/W | Individual Distributed DMA Channel Enable Register |
| 85-87h | RO | - Reserved |
| 88h | R/W | Serial Interrupt Control Register |
| 89-8Ah | R/W | Serial Interrupt Request Enable Register 1,2 |
| 8Bh | RO | - Reserved |
| 8Ch | R/W | Serial Interrupt Request Enable Register 3 |
| 8D-8Fh | RO | - Reserved |
| 90-91h | R/W | ACPI Base Address Register |

## 6.2 LEGACY ISA REGISTERS

### 6.2.1 DMA REGISTERS

(These registers can be accessed from PCI bus and ISA bus)

| ADDRESS | ACCESS | 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 | R/W | DMA1 Request Register |
| 000Ah | R/W | DMA1 Command(r) Write Single Mask Bit (w) Register |
| 000Bh | R/W | DMA1 Mode DMA 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(r) Register |
| 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 | R/W | DMA2 Request Register |
| 00D4h | R/W | DMA2 Command(r) Write Single Mask Bit(w) Register |
| 00D6h | R/W | 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 | ACCESS | 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 |

(These registers can be accessed from PCI bus or ISA bus)

| ADDRESS | ACCESS | REGISTER NAME |
|---|---|---|
| 00480h | R/W | Reserved |
| 00481h | R/W | DMA Channel 2 High Page Register |
| 00482h | R/W | DMA Channel 3 High Page Register |
| 00483h | R/W | DMA Channel 1 High Page Register |
| 00484h | R/W | Reserved |
| 00485h | R/W | Reserved |
| 00486h | R/W | Reserved |
| 00487h | R/W | DMA Channel 0 High Page Register |
| 00488h | R/W | Reserved |
| 00489h | R/W | DMA Channel 6 High Page Register |
| 0048Ah | R/W | DMA Channel 7 High Page Register |
| 0048Bh | R/W | DMA Channel 5 High Page Register |
| 0048Ch | R/W | Reserved |
| 0048Dh | R/W | Reserved |
| 0048Eh | R/W | Reserved |
| 0048Fh | R/W | Reserved |

### 6.2.2 INTERRUPT CONTROLLER REGISTERS

(These registers can be accessed from PCI bus or ISA bus.)

| ADDRESS | ACCESS | 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 |

### 6.2.3 TIMER REGISTERS

(These registers can be accessed from PCI bus or ISA bus.)

| ADDRESS | ACCESS | 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 |

### 6.2.4 OTHER REGISTERS

(These registers can be accessed from PCI bus or ISA bus.)

| ADDRESS | ACCESS | REGISTER NAME |
|---|---|---|
| 0061h | R/W | NMI Status Register |
| 0070h | WO | CMOS RAM Address and NMI Mask Register |

| | | |
|---|---|---|
| 00F0h | WO | Coprocessor Error Register |
| 04D0h | R/W | IRQ Edge/Level Control Register 1 |
| 04D1h | R/W | IRQ Edge/Level Control Register 2 |

### 6.3 PMU CONFIGURATION REGISTERS

| ADDRESS | ACCESS | REGISTER NAME |
|---|---|---|
| 40h ~ 43h | R/W | System Standby Timer 0 Reload Event |
| 44h ~ 47h | R/W | System Standby Timer 1 Reload Event |
| 48h ~ 4Bh | R/W | System Standby Timer 2 Reload Event |
| 4Ch ~ 4Dh | R/W | Sound Port Trap and Address Mask for Programmable 16-bit I/O |
| 4Eh ~ 4Fh | R/W | Linear Frame Buffer Trap |
| 50h ~ 53h | R/W | Wake-up 0 |
| 54h ~ 57h | R/W | Wake-up 1 |
| 58h ~ 5Bh | R/W | Reserved |
| 5Ch ~ 5Dh | R/W | Programmable 10-bit I/O |
| 5Eh ~ 5Fh | R/W | Programmable 16-bit I/O |
| 60h ~ 63h | R/W | SMI# Enable |
| 64h ~ 67h | R/W | SMI#/SCI Request Status |
| 68h ~ 69h | R/W | IRQ and NMI Enable for Wake-up0 and System Standby Timer 0 |
| 6Ah ~ 6Bh | R/W | IRQ and NMI Enable for Wake-up1 and System Standby Timer 1 |
| 6Ch ~ 6Dh | R/W | IRQ and NMI Enable for System Standby Timer 2 |
| 6Eh ~ 6Fh | R/W | Reserved |
| 70h ~ 71h | R/W | GPCS0# Base Address |
| 72h | R/W | GPCS0# Address Mask |
| 73h | R/W | GPCS0# Control |
| 74h ~ 75h | R/W | GPCS1# Base Address |
| 76h | R/W | GPCS1# Address Mask |
| 77h | R/W | GPCS1# Control |
| 78h | R/W | GPCS0# and GPCS1# De-bounce Counter and GPCS0# Address A[11:8] Mask |
| 79h | R/W | System Standby Timer 0 |
| 7Ah | R/W | System Standby Timer 1 |
| 7Bh | R/W | System Standby Timer 2 |
| 7Ch | R/W | System Standby Timer Granularity |
| 7Dh | R/W | Auto Power Off Timer and PMU Test Mode |
| 7Eh ~ 7Fh | R/W | IDE Bus Master Port Trap |
| 80h ~ 83h | R/W | SCI Enable |

## 6.4 ACPI CONFIGURATION REGISTERS

| OFFSET | BYTE LENGTH | ACCESS | NAME | ABBREVIATE | CLASS |
|---|---|---|---|---|---|
| 00 | 2 | R/W | PM1 Status | PM1_STS | Fixed Features |
| 02 | 2 | R/W | PM1 Enable | PM1_EN | Fixed Features |
| 04 | 2 | R/W | PM1 Control | PM1_CTL | Fixed Features |
| 06 | 2 | R/W | Reserved | | Fixed Features |
| 08 | 4 | R/W | PM Timer | PM_TMR | Fixed Features |
| 0C | 4 | R/W | CPU Control | P_CTL | Fixed Features |
| 10 | 1 | RO | CPU Power State Level 2 | P_LVL2 | Fixed Features |
| 11 | 1 | RO | CPU Power State Level 3 | P_LVL3 | Fixed Features |
| 12 | 1 | R/W | PM2 Control | PM2_CTL | Fixed Features |
| 13 | 1 | R/W | Fixed Features Control | FIX_CTL | Fixed Features |
| 14 | 4 | R/W | GPE Status | GPE_STS | Generic Features |
| 18 | 4 | R/W | GPE Enable | GPE_EN | Generic Features |
| 1C | 3 | R/W | GPE Pin Status | GPE_Pin | Generic Features |
| 1F | 1 | R/W | GP Timer | GP_TMR | Generic Features |
| 20 | 4 | R/W | GPE I/O Selection | GPE_IO | Generic Features |
| 24 | 4 | R/W | GPE Polarity Selection | GPE_Pol | Generic Features |
| 28 | 2 | R/W | GPE Multi-definition Pins Selection | GPE_Mul | Generic Features |
| 2A | 2 | R/W | GPE Control | GPE_CTL | Generic Features |
| 2C | 2 | R/W | GPIO Events SMI# Enable | GPE_SMI | Generic Features |
| 2E | 2 | R/W | GPIO Events Stop GPTMR and Reload PMU Timers Enable | GPE_RL | Generic Features |
| 30 | 1 | R/W | Legacy Status | LEG_STS | Legacy |
| 31 | 1 | R/W | Legacy Enable | LEG_EN | Legacy |
| 32 | 1 | R/O | Reserved | | |
| 33 | 1 | R/W | Test Control | TST_CTL | Legacy |
| 34 | 1 | R/W | Reserved | | |
| 35 | 1 | R/W | SMI# Command Port | SMI_CMD | Legacy |
| 36 | 1 | R/W | Free Read/Write Byte | Free_Byte | Legacy |
| 37 | 1 | R/O | Reserved | | |

| | | | | | |
|---|---|---|---|---|---|
| 38 | 1 | R/W | SMBus IO Index | SMB_INDEX | SMBus |
| 39 | 1 | R/W | SMBus IO Data | SMB_DAT | SMBus |
| Note: ALL ACPI registers can be accessed by byte, word, or D-word in length. | | | | | |

### 6.5 SMBUS IO REGISTERS

| OFFSET | BYTE LENGTH | ACCESS | NAME | ABBREVIATE | COMMENT |
|---|---|---|---|---|---|
| 00~01 | 2 | R/W | SMBus Host Status | SMB_STS | |
| 02~03 | 2 | R/W | SMBus Host Control | SMB_CTL | |
| 04 | 1 | R/W | SMBus Address Field | SMB_ADS | |
| 05 | 1 | R/W | SMBus Command Field | SMB_CMD | |
| 06 | 1 | RO | SMBus Data Processed Count | SMB_PCNT | |
| 07 | 1 | R/W | SMBus Data Byte Count | SMB_CNT | |
| 08 | 1 | R/W | SMBus Data Byte 0 | SMB_BYTE0 | |
| 09 | 1 | R/W | SMBus Data Byte 1 | SMB_BYTE1 | |
| 0A | 1 | R/W | SMBus Data Byte 2 | SMB_BYTE2 | |
| 0B | 1 | R/W | SMBus Data Byte 3 | SMB_BYTE3 | |
| 0C | 1 | R/W | SMBus Data Byte 4 | SMB_BYTE4 | |
| 0D | 1 | R/W | SMBus Data Byte 5 | SMB_BYTE5 | |
| 0E | 1 | R/W | SMBus Data Byte 6 | SMB_BYTE6 | |
| 0F | 1 | R/W | SMBus Data Byte 7 | SMB_BYTE7 | |
| 10 | 1 | R/W | SMBus Device Address | SMB_DEV | Device Master Address |
| 11 | 1 | R/W | SMBus Device Byte0 | SMB_DB0 | Low Byte |
| 12 | 1 | R/W | SMBus Device Byte1 | SMB_DB1 | High Byte |
| 13 | 1 | R/W | SMBus Host Alias Address | SMB_HAA | Another Host Address |

### 6.6 USB OPENHCI HOST CONTROLLER CONFIGURATION SPACE

#### 6.6.1 USB CONFIGURATION SPACE (FUNCTION 2)

| CONFIGURATION. OFFSET | ACCESS | MNEMONIC REGISTER |
|---|---|---|
| 00-01h | RO | VID Vendor ID |
| 02-03h | RO | DID Device ID |

| | | |
|---|---|---|
| 04-05h | R/W | CMD Command Register |
| 06-07h | R/W | STS Status register |
| 08h | RO | RID Revision ID |
| 09-0Bh | RO | CD Class Code |
| 0Ch | RO | CL Cache Line Size |
| 0Dh | R/W | MLT Master Latency Timer |
| 0Eh | RO | HT Header Type |
| 0Fh | RO | BIST Built-in Self Test |
| 10-13h | R/W | Base address |
| 13-3Bh | RO | - Reserved |
| 3Ch | R/W | INTL Interrupt line |
| 3Dh | RO | INTP Interrupt pin |
| 3Eh | RO | MINGNT Min Gnt |
| 3Fh | RO | MAXLAT Max Latency |

### 6.6.2 HOST CONTROLLER OPERATIONAL REGISTERS

| OFFSET | 3100 |
|---|---|
| 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 |

## 6.7 AUTOMATIC POWER CONTROL (APC) REGISTERS

(The following register can be accessed when PCI to ISA Register 45, bit 1 is set to 1.)

| ADDRESS | ACCESS | REGISTER NAME |
|---|---|---|
| 00h | RO | Reserved |
| 01h | RO | Reserved |
| 02h | R/W | APC Register 02h |
| 03h | R/W | APC Register 03h |
| 04h | R/W | APC Register 04h |
| 05h | R/W | APC Register 05h |
| 06h | RO | APC Register 06h |
| 07h | R/W | APC Register 07h |
| 08h | R/W | APC Register 08h |

### 6.7.1 RTC REGISTERS

| ADDRESS | ACCESS | 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 the Week |
| 07h | R/W | Day of the 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 the Month Alarm |
| 7Fh | R/W | Month Alarm |

## 6.8 DATA ACQUISITION MODULE (DAM) INTERNAL REGISTERS

| OFFSET | REGISTER | ACCESS |
|---|---|---|
| 40h | Configuration | R/W |
| 41h | Interrupt Status I | RO |
| 42h | Interrupt Status II | RO |
| 43h | SMI# Mask I | R/W |
| 44h | SMI# Mask II | RO |
| 45h | NMI Mask I | RO |
| 46h | NMI Mask II | RO |
| 47h | Fan Divisor | R/W |
| 48h | - Reserved | RO |
| 20h/60h | VIN0 Reading | RO |
| 21h/61h | VIN1 Reading | RO |
| 22h/62h | VIN2 Reading | RO |
| 23h/63h | VIN3 Reading | RO |
| 24h/64h | DXP/VIN4 Reading | RO |
| 25h~27h/65~67h | - Reserved | RO |
| 28h/68h | Fan 1 Reading | RO |
| 29h/69h | Fan 2 Reading | RO |
| 2Ah/6Ah | - Reserved | RO |
| 2Bh/6Bh | VIN0 High Limit | R/W |
| 2Ch/6Ch | VIN0 Low Limit | R/W |
| 2Dh/6Dh | VIN1 High Limit | R/W |
| 2Eh/6Eh | VIN1 Low Limit | R/W |
| 2Fh/6Fh | VIN2 High Limit | R/W |
| 30h/70h | VIN2 Low Limit | R/W |
| 31h/71h | VIN3 High Limit | R/W |
| 32h/72h | VIN3 Low Limit | R/W |
| 33h/73h | DXP/VIN4 High Limit | R/W |
| 34h/74h | DXP/VIN4 Low Limit | R/W |
| 35h~3Ah/75h~7Ah | - Reserved | R/W |
| 3Bh/7Bh | FAN1 fan count limit | R/W |
| 3Ch/7Ch | FAN2 fan count limit | R/W |
| 3Dh/7Dh | - Reserved | RO |

# 7 REGISTER DESCRIPTION

## 7.1 PCI TO ISA BRIDGE CONFIGURATION REGISTERS

| DEVICE | IDSEL | FUNCTION NUMBER |
|---|---|---|
| PCI to ISA bridge | AD12 | 0000b |

**Registers 00h~01h Vendor ID**

Default Value: 1039h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Vendor Identification Number**<br>Default value is 1039h |

**Registers 02h~03h Device ID**

Default Value: 0008h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Device Identification Number**<br>Default value is 0008h |

**Registers 04h~ 05h Command Register**

Default Value: 000Ch

Access: Read/Write, Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | RO | **Reserved.**<br>Read as 0 |
| 3 | RO | **Read as 1 to indicate the device is allowed to monitor special cycles.** |
| 2 | RO | **Read as 1 to indicate the device is able to become PCI bus master.** |
| 1 | R/W | **Response to Memory Space Accesses (default=0)**<br>This bit should be written to 1 after reset. |
| 0 | R/W | **Response to I/O Space Accesses (default =0)**<br>This bit should be written to 1 after reset. |

**Registers 06h~07h Status**

Default Value: 0200h

Access: Read/Write, Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:14 | RO | **Reserved.**<br>Read as 0. |
| 13 | RO | **Received Master-Abort**<br>This bit will be set to 1 when the current transaction is terminated with master-abort. This bit can be cleared to 0 by writing a 1. |
| 12 | RO | **Received Target-Abort**<br>This bit will be set to 1 when the current transaction is terminated with target-abort. This bit can be cleared to 0 by writing a 1. |
| 11 | RO | **Reserved.**<br>Read as 0. |
| 10:9 | R/W | **DEVSEL# Timing**<br>The two bits are hardwired to 01 to indicate positive decode with medium timing. |
| 8:0 | RO | **Reserved.**<br>Read as 0. |

**Register 08h Revision ID**

Default Value: B0h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Revision Identification Number** |

**Register 09h~0Bh Class Code**

Default Value: 060100h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 23:0 | RO | **Class Code**<br>Default value is 060100h |

**Register 0Ch Cache Line Size**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Cache Line Size** |

**Register 0Dh Master Latency Timer**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Master Latency Timer.** |

**Register 0Eh Header Type**

Default Value: 80h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Header Type**<br>Default value is 80h |

**Register 0Fh BIST**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **BIST**<br>Default value is 00h |

**Register 10h~3Ch Reserved. Read as 0.**

**Register 40h BIOS Control Register**

Default Value: 08h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **ACPI Enable**<br>0 : Disable (default)<br>1 : Enable<br>When enabled, ACPI registers located at IO space address as defined in ACPI Base registers (Reg. 90h~91h) can be accessed. |
| 6 | R/W | **PC/PCI DMA ISA Master Retry Mode Selection**<br>When this bit is programmed to 0, the PC/PCI DMA controller will not release the PCI bus as the PC/PCI DMA ISA Master cycle is retried. Inversely, the PC/PCI DMA controller will release the PCI bus. |
| 5 | R/W | **PCI Delayed Transaction Enable**<br>0 : Disable (default)<br>1 : Enable |
| 4 | R/W | **PCI Posted Write Buffer Enable**<br>0 : Disable (default)<br>1 : Enable |

| | | |
|---|---|---|
| 3 | R/W | **Positive Decode of Upper 64K BYTE BIOS Enable.** |
| 2 | R/W | **BIOS Subtractive Decode Enable.** |
| 1 | R/W | **Lower BIOS Enable.**<br>BIT[3:2] will only be effective when this bit is enabled. |
| 0 | R/W | **Extended BIOS Enable. (FFF80000~FFFDFFFF)**<br>When enabled, the device will positively respond to PCI cycles toward the Extended segment. |

| BITS [3:2] | F SEGMENT | | E SEGMENT | | COMMENT |
|---|---|---|---|---|---|
| | + | - | + | - | |
| 00 | | | v* | | Chip positively responds to E segment access. |
| 10 | v | | v* | | Chip positively responds to E and F segment access. |
| Others | | v | | | Chip subtractively responds to F segment access. |

Note *: Enabled if bit 1 is set.

**Register 41h/42h/43h/44h PCI INTA#/B#/C#/D# Remapping Register**

Default Value: 80/80/80/80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Remapping Enable**<br>0 : Enable<br>1 : Disable (default)<br>When enabled, PCI INTA#/B#/C#/D# will be remapped to the IRQ channel specified below. |
| 6:4 | RO | **Reserved.**<br>Read as 0 |
| 3:0 | R/W | **IRQx Remapping Table** |

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

Note: More than one of INT[A:D]# can be remapped to the same IRQ line, but that IRQ line should be programmed to level-triggered mode in Register 4D0h/4D1h.

**Register 45h ISA Bus Control Register I**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **ISA Bus Clock Selection**<br>00: 7.159MHz<br>01: PCICLK/4<br>10: PCICLK/3<br>11:Reserved |
| 5 | R/W | **Flash EPROM Control bit 0**<br>Refer to bit 2 of this register. |
| 4 | R/W | **Test bit for internal use only**<br>0 : Normal mode<br>1 : Test mode |
| 3 | R/W | **RTC Extended Bank Enable (EXTEND_EN)**<br>0 : Disable<br>1 : Enable<br>When this bit is enabled, the upper 128 bytes of RTC SRAM can be accessed. |
| 2 | R/W | **Flash EPROM Control Bit 1**<br>If bit 5 or 2 is not set to '1' after CPURST de-asserted, EPROM can be flashed. Once bit 5 or bit 2 is set to 1, EPROM can only be flashed when bit 5, 2 = 01 |
| 1 | R/W | **Automatic Power Control Registers (APCREG_EN) Enable**<br>0 : Disable<br>1 : Enable<br>When this bit is enable, APC registers can be accessed. |
| 0 | R/W | **Reserved. This bit must be programmed to 0** |

**Register 46h ISA Bus Control Register II**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **16-Bit I/O Cycle Command Recovery Time**<br>00: 5 BUSCLK (default)<br>01: 4 BUSCLK<br>10: 3 BUSCLK<br>11: 2 BUSCLK |

| | | |
|---|---|---|
| 5:4 | R/W | **8-Bit I/O Cycle Command Recovery Time**<br>00: 8 BUSCLK (default)<br>01: 5 BUSCLK<br>10: 4 BUSCLK<br>11: 3 BUSCLK |
| 3 | R/W | **ROM Cycle Wait State Selection**<br>0 : 4 wait states (default)<br>1 : 1 wait state |
| 2:0 | R/W | **Test bit for internal use only**<br>0 : Normal Mode (default)<br>1 : Test Mode |

**Register 47h DMA Clock and Wait State Control Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **PC/PCI DMA Function Enable**<br>0 : Disable<br>1 : Enable |
| 6 | R/W | **EXTEND DACK# Enable**<br>0 : Disable<br>1 : Enable (recommended)<br>When enabled, the assertion time of DACK# will be extended by 1/2 BCLK. |
| 5:4 | R/W | **16-Bit DMA Cycle Wait State**<br>00 : 1 DMACLK (default)<br>01 : 2 DMACLK<br>10 : 3 DMACLK<br>11 : 4 DMACLK |
| 3:2 | R/W | **8-Bit DMA Cycle Wait State**<br>00 : 1 DMACLK (default)<br>01 : 2 DMACLK<br>10 : 3 DMACLK<br>11 : 4 DMACLK |
| 1 | R/W | **Reserved.** |
| 0 | R/W | **DMA Clock Selection**<br>0: ISA Bus clock divided by 2 (recommended)<br>1 : Same as ISA Bus clock |

**Register 48h ISA to PCI Top of Memory Region Register**

Default Value: 01h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **Top of Memory region**<br>0000 : 000FFFFFh<br>0001 : 001FFFFFh<br>0010 : 002FFFFFh<br>0011 : 003FFFFFh<br>:<br>:<br>1101 : 00DFFFFFh<br>1110 : 00EFFFFFh<br>1111 : 00FFFFFFh<br>ISA master or DMA memory cycles will be qualified to be forwarded to PCI bus when their address in memory space fall within a certain memory region. The bottom address of the memory region is always at 00100000h. The top address can be programmed in bits[7:4] of this register. All memory cycles in this region will be forwarded to PCI bus except the ones that fall within a memory hole defined in register 4Ah~4Bh. |
| 3 | R/W | **E0000h-EFFFFh Memory Region forwarded to PCI Enable**<br>0 : Disable (default)<br>1 : Enable<br>ISA master and DMA memory cycles fall within this memory region will also be forwarded to PCI bus when enabled. |
| 2 | R/W | **A0000h-BFFFFh memory Region forwarded to PCI Enable** |
| 1 | R/W | **80000h-9FFFFh Memory Region forwarded to PCI Enable** |
| 0 | R/W | **00000h-7FFFFh Memory Region forwarded to PCI Enable** |

**Register 49h ISA to PCI Memory Region Enable Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **DC000h-DFFFFh Memory Region forwarded to PCI Enable** |
| 6 | R/W | **D8000h-DBFFFh Memory Region forwarded to PCI Enable** |
| 5 | R/W | **D4000h-D7FFFh Memory Region forwarded to PCI Enable** |
| 4 | R/W | **D0000h-D3FFFh Memory Region forwarded to PCI Enable** |
| 3 | R/W | **CC000h-CFFFFh Memory Region forwarded to PCI Enable** |
| 2 | R/W | **C8000h-CBFFFh memory Region forwarded to PCI Enable** |

| | | |
|---|---|---|
| 1 | R/W | **C4000h-C7FFFh Memory Region forwarded to PCI Enable** |
| 0 | R/W | **C0000h-C3FFFh Memory Region forwarded to PCI Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 4Ah ISA to PCI Memory Hole Bottom Address Register**

Default Value: 10h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Bottom address bits A[23:16] of the memory hole**<br>Register 4Ah and register 4Bh define a memory hole located between 00100000h and 00FFFFFFh. ISA master and DMA memory cycles fall within this hole will not be forwarded to PCI bus. Both the bottom and the top addresses must be larger than or equal to 00100000h. If the bottom address is greater than the top address, then the memory hole is disabled. |

**Register 4Bh ISA to PCI Memory Hole Top Address Register**

Default Value: 0Fh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Top address bits A[23:16] of the memory hole** |

**Register 4Ch~4Fh Shadow Register of ICW1 to ICW4 of INT1**

Default Value: 00000000h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect ICW1 to ICW4 of the master interrupt controller** |

**Register 50h~53h Shadow Register of ICW1 to ICW4 of INT2**

Default Value: 00000000h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect ICW1 to ICW4 of the slave interrupt controller** |

**Register 54h~55h Shadow Register of OCW2 & OCW3 of INT1**

Default Value: 0000h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect OCW2 and OCW3 of the master interrupt controller** |

**Register 56h~57h Shadow Register of OCW2 & OCW3 of INT2**

Default Value: 0000h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect OCW2 and OCW3 of the slave interrupt controller** |

**Register 58h CTC Shadow Register 1**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect low byte of the initial count number of CTC Counter 0** |

**Register 59h CTC Shadow Register 2**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect high byte of the initial count number of CTC Counter 0** |

**Register 5Ah CTC Shadow Register 3**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect low byte of the initial count number of CTC Counter 1** |

**Register 5Bh CTC Shadow Register 4**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect high byte of the initial count number of CTC Counter 1** |

**Register 5Ch CTC Shadow Register 5**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect low byte of the initial count number of CTC Counter 2** |

**Register 5Dh CTC Shadow Register 6**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect high byte of the initial count number of CTC Counter 2** |

**Register 5Eh CTC Shadow Register 7**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect Control word (43h) of the built-in CTC** |

**Register 5Fh Shadow Register 8**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | RO | **Reserved** |
| 5 | RO | **CTC counter2 Write count pointer status** |
| 4 | RO | **CTC counter1 Write count pointer status** |
| 3 | RO | **CTC counter0 Write count pointer status** |
| 2 | RO | **CTC counter2 Read count pointer status** |
| 1 | RO | **CTC counter1 Read count pointer status** |
| 0 | RO | **CTC counter0 Read count pointer status**<br>0 : LSB<br>1 : MSB |

**Register 60h Shadow Register for ISA port 70h**

Default Value: FFh

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reflect the content of ISA port 70h register** |

**Register 61h IDEIRQ Remapping Register**

Default Value: 80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **IDEIRQ Remapping Enable**<br>0 : Enable<br>1 : Disable (default) |
| 6 | R/W | **Control the output data value sampled from SIRQ and this function only for IRQ data in SIRQ.**<br>0: output value is the same as the sampled data.<br>1: output value is the inverted value of the sampled data. |
| 5 | R/W | **Reserved.** |
| 4 | R/W | **IDE Channel Remapping Selection**<br>0 : Primary IDE channel<br>1 : Secondary IDE channel |
| 3:0 | R/W | **IRQ Remapping Table** |

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

**Register 62h USBIRQ Remapping Register**

Default Value: 80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **USBIRQ Remapping Enable**<br>0 : Enable<br>1 : Disable (default) |
| 6 | R/W | **Integrated USB Enable**<br>0 : Disable<br>1 : Enable |
| 5:4 | R/W | **Reserved. This bit must be programmed to 0.** |
| 3:0 | R/W | **IRQ Remapping Table** |

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

**Register 63h PCI Output Buffer Current Strength Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | R/W | **Reserved.**<br>These bits must be programmed to 0 |
| 4:3 | R/W | **Hardware reset initiated by software**<br>When both set to 1, hardware reset will be generated to CPU. |
| 2 | R/W | **Software Power Off System Control(SPWROFF)**<br>Before enabling this function, the bit6 at APC Register 03h should be enabled. Once writing a 1 to this bit, system will be powered off. |
| 1:0 | R/W | **PCI Output Buffer Current Strength Selection**<br>00: 4mA (recommended)<br>11: 8mA<br>Others: Reserved |

**Register 64h INIT Enable Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **SMC37C673 Super I/O Compatible Mode**<br>The two bits should be programmed to 1, if a SMC37C673 Super IO chip is connected to SiS5595 via serial IRQ line. Bit_7 enables the chipset to latch SIRQ1, while Bit_6 enables the chipset to latch SIRQ12. For all other super IO chips, the two bits should be programmed to 0 |
| 5 | R/W | **INIT Enable**<br>0: Drives CPURST during S/W reset and INIT is inactive.<br>1: Drives INIT during S/W reset |
| 4 | R/W | **Fast Gate 20 Emulation**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 3 | R/W | **Fast Reset Latency Control**<br>0: 2us<br>1: 6us |
| 2 | R/W | **Fast Reset Emulation**<br>0: Disable<br>1: Enable |
| 1 | R/W | **A20M# Output Control**<br>0: Enable the assertion of A20M# if applicable.<br>1: Disable the assertion of A20M#, i.e., A20M# will be high at all times. |
| 0 | R/W | **Enable Keyboard Hardware Reset**<br>0: Disable<br>1: Enable |

**Register 65h PHOLD# Timer**

Default Value: 01h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **PHOLD# Timer**<br>The PHOLD# timer sets an upper limit in terms of PCI clock for the assertion time of PHOLD# initiated by USB or DDMA master. The timer starts and continues the counting when USB or DDMA master is asserting PHOLD#. Upon expiration, USB or DDMA master will be forced to de-assert PHOLD#. The maximum allowable value is FFh and the minimum allowable value is 01h. If a larger value is programmed, USB or DDMA master will be able to complete more PCI transactions by preventing the system arbiter from issuing GNT# to other PCI master candidates. The PCI bus bandwidth can be fairly shared by all PCI master candidates by properly program this timer. |

**Register 66h Priority Timer**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Priority Timer**<br>There are three PCI maser candidates inside the south-bridge chip competing for the PCI bus. They are- ISA/DMA master, DDMA and USB. A local arbiter with rotating arbitration scheme is adopted to coordinate their requests to become PCI master. The candidate that issues request to the arbiter with a higher priority is the winner and is eligible to become PCI master when PHLDA# is received. The priority timer is used to set a lower limit in terms of PCI clock for the wining candidates to continue its PCI transactions. The timer will start counting as soon as the winning candidate receives PHLDA#. Upon expiration, the winning candidate's priority will become the lowest among the three and, if there are other requests are outstanding, lose the ownership of PHLDA#. The maximum allowable value is FFh and the minimum allowable value is 00h. |

**Register 67h Respond to C/D Segments Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **ISR bits clear SIRQ1 and SIRQ 12 Latches Enable**<br>When set to 1, the internal latches for SIRQ1 and SIRQ12 will be cleared when the corresponding ISR bits are set in Interrupt Controller. The latches only take effective when either register 64h bit 7 or bit 6 is set to 1.The latches will always be cleared by a IO read cycle with address=60h. |
| 6:5 | R/W | **Reserved. These bits must be programmed to 0.** |
| 4:3 | R/W | **Test Mode Bits. These bits must be programmed to 0.** |
| 2 | R/W | **Subtractive Decode to Internal registers Enable**<br>0 : Disable<br>1 : Enable<br>When this bit is enabled , the SiS5595 will do subtractive decode on addresses for internal registers . |
| 1 | R/W | **Respond to C/D Segments Enable**<br>0 : Disable<br>1 : Enable<br>When enabled, the chip will positively responded to PCI memory cycles toward C segment and D segment; i.e., from FFC00000h to FFDFFFFFh. |
| 0 | R/W | **Test Mode Bit. This bit must be programmed to 0.** |

**Register 68~69h Data Acquisition Module Base Address**

Default Value: 0290h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:3 | R/W | **Correspond to Base Address A[15:3] in IO space**<br>Specify the base address for the internal registers of the Data Acquisition Module. |
| 2:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 6Ah ACPI/SCI IRQ Remapping Register**

Default Value: 80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **ACPI/SCI IRQ Remapping Enable**<br>1 : Disable<br>0 : Enable |
| 6:4 | R/W | **Reserved. This bit must be programmed to 0** |
| 3:0 | R/W | **Interrupt Remapping Table.** |

| Bits | IRQx# | Bits | IRQx# | Bits | IRQx# |
|---|---|---|---|---|---|
| 0000 | reserved | 0110 | IRQ6 | 1100 | IRQ12 |
| 0001 | reserved | 0111 | IRQ7 | 1101 | IRQ13 |
| 0010 | reserved | 1000 | Reserved | 1110 | IRQ14 |
| 0011 | IRQ3 | 1001 | IRQ9 | 1111 | IRQ15 |
| 0100 | IRQ4 | 1010 | IRQ10 | | |
| 0101 | IRQ5 | 1011 | IRQ11 | | |

**Register 6Bh Test Mode Register I**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Test bits. These bits should be programmed to 0.** |

**Register 6Ch Test Mode Register II**

Default Value: 20h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **Test bits. These bits should be programmed to 0.** |

| | | |
|---|---|---|
| 5 | R/W | **IRQ13 Control**<br>When set to 1, IRQ13 will be routed to FERR#. (default=1) |
| 4:2 | R/W | **Test bits. These bits should be programmed to 0.** |
| 1 | R/W | **Enable the Reading of all DMA base registers.**<br>0 : Disable<br>1 : Enable |
| 0 | R/W | **Reserved. This bit must be programmed to 0.** |

**Register 6Dh I²C Bus Control Register**

Default Value: 19h

Access: Read/Write, Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Internal Test Bit for Keyboard and PS/2 Mouse Pin Swapping**<br>0 : Normal Mode<br>1 : Test Mode |
| 6 | R/W | **Reserved.** |
| 5 | R/W | **Serial IRQ sampled IOCHK phase control**<br>0 : The sampled IOCHK on serial IRQ will be inverted.<br>1 : The sampled IOCHK on serial IRQ will not be inverted. (recommended) |
| 4 | R/W | **I2C Bus Data Active Level Control**<br>0 : Active Low<br>1 : Active High |
| 3 | R/W | **I2C Bus Clock Active Level Control**<br>0 : Active Low<br>1 : Active High |
| 2 | R/W | **I2C Bus Control**<br>0 : Disable<br>1 : Enable |
| 1 | RO | **Keyboard Hot Key Status**<br>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 enabled. |
| 0 | R/W | **Keyboard Hot Key Control**<br>0 : Disable<br>1 : Enable<br>This bit is meaningful only when internal KBC is enabled. |

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

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Interrupt Channel 7** |
| 6 | R/W | **Interrupt Channel 6** |
| 5 | R/W | **Interrupt Channel 5** |
| 4 | R/W | **Interrupt Channel 4** |
| 3 | R/W | **Interrupt Channel 3** |
| 2 | R/W | **Interrupt Channel 2** |
| 1 | R/W | **Interrupt Channel 1** |
| 0 | R/W | **Interrupt Channel 0**<br>Writing a 1 to these bits will cause the corresponding interrupt requests to be outstanding. This register defaults to all 0. |

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

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Interrupt Channel 15** |
| 6 | R/W | **Interrupt Channel 14** |
| 5 | R/W | **Interrupt Channel 13** |
| 4 | R/W | **Interrupt Channel 12** |
| 3 | R/W | **Interrupt Channel 11** |
| 2 | R/W | **Interrupt Channel 10** |
| 1 | R/W | **Interrupt Channel 9** |
| 0 | R/W | **Interrupt Channel 8**<br>Writing a 1 to these bits will cause the corresponding interrupt requests to be outstanding. |

**Register 70h Misc. Controller Register**

Default Value: 12h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **Reserved and should be programmed to 0.** |
| 5 | R/W | **Test bit. This bit should be programmed to 0** |

| | | |
|---|---|---|
| 4 | R/W | **Keyboard Lock Enable**<br>0 : Disable<br>1 : Enable(default) |
| 3 | R/W | **Integrated Keyboard Controller Enable**<br>0 : Disable<br>1 : Enable |
| 2 | R/W | **Integrated PS/2 Mouse Enable**<br>0 : Disable (default)<br>1 : Enable<br>This bit is meaningful only when Bit3 is enabled. |
| 1 | R/W | **Internal RTC Status**<br>0 : Disable<br>1 : Enable |
| 0 | R/W | **Test bit. This bit should be programmed to 0.** |

**Register 71h Reserved**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Reserved and should be programmed to 0.** |

**Register 72h Individual PC/PCI DMA Channel Enable**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Channel 7 PC/PCI DMA Enable** |
| 6 | R/W | **Channel 6 PC/PCI DMA Enable** |
| 5 | R/W | **Channel 5 PC/PCI DMA Enable** |
| 4 | R/W | **Reserved. This bit must be programmed to 0.** |
| 3 | R/W | **Channel 3 PC/PCI DMA Enable** |
| 2 | R/W | **Channel 2 PC/PCI DMA Enable** |
| 1 | R/W | **Channel 1 PC/PCI DMA Enable** |
| 0 | R/W | **Channel 0 PC/PCI DMA Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 73h~79h Reserved**

**Register 7Ah Data Acquisition Module Function Selection Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **DXP/VIN4 Mode Selection**<br>0 : VIN4 Mode<br>1 : DXP Mode |
| 6:5 | R/W | **Reserved** |
| 4 | R/W | **ADC Calibration Function Enable**<br>0 : Disable<br>1 : Enable.<br>Before setting this bit, the bit 0 of index register 40h ( start ) should be reset to 0, i.e., stops the round-robin monitoring, or the ADC Calibration Function would not start. This bit would be cleared after the ADC Calibration function is finished. |
| 3:0 | R/W | **Reserved** |

**Register 7Bh Data Acquisition Module Control Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Data Acquisition Module Enable**<br>0 : Disable<br>1 : Enable |
| 6:0 | R/W | **Test bits.**<br>These bits should be programmed to 0. |

**Register 7C~7D Data Acquisition Module ADC Calibration**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | This register is used as the calibration register for the Analog-to-Digital Converter at the Data Acquisition Module. |

**Register 7Eh Data Acquisition Module and SMBUS IRQ Remapping Register**

To activate DAM Remapping function, bit 7 and bit 6 have to be enabled. To activate SMBus Remapping function, bit 7 and bit 5 have to be enabled. In other words, bit [7:5] have to be enabled at the same time to enable both DAM and SMBus Remapping function.

Default Value: 80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Enable Both Data Acquisition Module and SMBUS IRQ Remapping with bit [6:5] enabled.**<br>0 : Enable<br>1 : Disable |
| 6 | R/W | **Enable Data Acquisition Module IRQ Remapping with bit 7 enabled.**<br>0 : Enable<br>1 : Disable |
| 5 | R/W | **Enable SMBus IRQ Remapping with bit 7 enabled.**<br>0 : Enable<br>1 : Disable |
| 4 | RO | **Data Acquisition Module IRQ and SMBus IRQ Status**<br>This bit reflects the IRQ status for Data Acquisition Module or SMBus if their associated IRQ Second Enable bit is enabled.<br>0: Indicates neither Data Acquisition Module IRQ nor SMBus IRQ is active.<br>1: Indicates either Data Acquisition Module IRQ or SMBus IRQ is active. |
| 3:0 | R/W | **IRQ Remapping Table** |

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

**Register 7Fh Reserved**

**Register 80h~81h Distributed DMA Master Configuration Register**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **DDMA slave base address bits[15:4]**<br>The DMA slave channels must be grouped into a 128 bytes block with 16 bytes per channel. The DMA slave channel 0 will be located at the base address specified here. |

| | | |
|---|---|---|
| 3:1 | R/W | **Reserved.**<br>This bit must be programmed to 0. |
| 0 | R/W | **DDMA Function Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 82~83h Reserved.**

**Register 84h Individual Distributed DMA Channel Enable**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Channel 7 DDMA Enable** |
| 6 | R/W | **Channel 6 DDMA Enable** |
| 5 | R/W | **Channel 5 DDMA Enable** |
| 4 | R/W | **Reserved. This bit must be programmed to 0.** |
| 3 | R/W | **Channel 3 DDMA Enable** |
| 2 | R/W | **Channel 2 DDMA Enable** |
| 1 | R/W | **Channel 1 DDMA Enable** |
| 0 | R/W | **Channel 0 DDMA Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 85h~87h Reserved**

**Register 88h Serial Interrupt Control Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Serial Interrupt (SIRQ) Control**<br>0 : Disable (default)<br>1 : Enable |
| 6 | R/W | **Quiet/Continuous Mode**<br>0 : Continuous (default)<br>1 : Quiet |
| 5:2 | R/W | **SIRQ Sample Period**<br>0000: 17 slots (default)<br>0001: 18 slots<br>0010: 19 slots<br>::::::::::::<br>1111: 32 slots |

| | | |
|---|---|---|
| 1:0 | R/W | **Start Cycle length**<br>00: 4 PCI clocks (default)<br>01: 6 PCI clocks<br>10: 8 PCI clocks<br>11: Reserved |

**Register 89h Serial Interrupt Enable Register 1**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Serial IRQ8 Enable** |
| 6 | R/W | **Serial IRQ7 Enable** |
| 5 | R/W | **Serial IRQ6 Enable** |
| 4 | R/W | **Serial IRQ5 Enable** |
| 3 | R/W | **Serial IRQ4 Enable** |
| 2 | R/W | **Serial IRQ3 Enable** |
| 1 | R/W | **Serial SMI# Enable** |
| 0 | R/W | **Serial IRQ1 Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 8Ah Serial Interrupt Enable Register 2**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Serial IOCHCK# Enable** |
| 6 | R/W | **Serial IRQ15 Enable** |
| 5 | R/W | **Serial IRQ14 Enable** |
| 4 | R/W | **Serial IRQ13 Enable** |
| 3 | R/W | **Serial IRQ12 Enable** |
| 2 | R/W | **Serial IRQ11 Enable** |
| 1 | R/W | **Serial IRQ10 Enable** |
| 0 | R/W | **Serial IRQ 9 Enable**<br>0 : Disable (default)<br>1 : Enable |

**Register 8Bh Reserved. Read Only Register**

**Register 8Ch Serial Interrupt Enable Register 3**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Reserved. These bits must be programmed to 0.** |
| 6 | R/W | **Serial INTA Enable**<br>0 : Disable (default)<br>1 : Enable |
| 5 | R/W | **Serial INTB Enable**<br>0 : Disable (default)<br>1 : Enable |
| 4 | R/W | **Serial INTC Enable**<br>0 : Disable (default)<br>1 : Enable |
| 3 | R/W | **Serial INTD Enable**<br>0 : Disable (default)<br>1 : Enable |
| 2:0 | R/W | **Reserved. These bits must be programmed to 0.** |

**Register 90h~91h ACPI Base Address Register**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:6 | R/W | **ACPI Base Address A[15:6]**<br>ACPI registers will be located at the base address specified here. |
| 5:0 | RO | **Reserved.**<br>Read as 0. |

## 7.2 PMU CONFIGURATION REGISTERS

| DEVICE | IDSEL | FUNCTION NUMBER |
|---|---|---|
| PMU | AD12 | 0001h |

**Register 00h~01h Vendor ID**

Default Value: 1039h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Vendor Identification Number** |

**Register 02h~03h Device ID**

Default Value: 0009h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Device Identification Number** |

**Register 04h~ 05h Command Port**

Default Value : 0000h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Reserved.**<br>Read as 0. |

**Register 06h~07h Status**

Default Value : 0200h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:14 | RO | **Reserved.**<br>Read as 0. |
| 13 | RO | **Received Master-Abort.**<br>Read as 0. |
| 12 | RO | **Received Target-Abort.**<br>Read as 0. |
| 11 | RO | **Reserved.**<br>Read as 0. |
| 10:9 | RO | **DEVSEL# Timing**<br>The two bits are hardwired to 01 to indicate positive decode with medium timing. |
| 8:0 | RO | **Reserved.**<br>Read as 0. |

**Register 08h Revision ID**

Default Value: 00h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Revision Identification Number** |

**Register 09h~0Bh Class Code**

Default Value : FF0000h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 23:0 | RO | **Class Code** |

**Register 0Ch Cache Line Size**

Default Value : 00h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Cache Line Size** |

**Register 0Dh Master Latency Timer**

Default Value : 00h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Master Latency Timer** |

**Register 0Eh Header Type**

Default Value : 80h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Header Type** |

**Register 0Fh BIST**

Default Value : 00h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **BIST** |

**Register 10h~3Fh**

Default Value : 00000000h

Access : Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:0 | RO | **Reserved.**<br>Read as 0. |

**Register 40h~43h System Standby Timer 0 Reload Event Control register**

**Register 44h~47h System Standby Timer 1 Reload Event Control register**

**Register 48h~4Bh System Standby Timer 2 Reload Event Control register**

Default Value : 00000000h

Access : Read/Write

The three register sets are used to define the System Standby Timer reload events as described below:

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **Reload Enable—Primary IDE Channel Device 0 Access**<br>When set, any I/O access cycle to the primary IDE ports 1F0-1F7h, 3F6h or IDE bus master ports with DEV=0 in Device/Head Register will reload the System Standby Timer. |
| 30 | R/W | **Reload Enable—Secondary IDE Channel Device 0 Access**<br>When set, any I/O access cycle to the secondary IDE ports 170-177h, 376h, or IDE bus master ports with DEV=0 in Device/Head Register will reload the System Standby Timer. |
| 29 | R/W | **Reload Enable—Keyboard Ports Access**<br>When set, any I/O access to the keyboard ports (60h or 64h) will reload the System Standby Timer. |
| 28 | R/W | **Reload Enable—Serial Port 1 Access**<br>When set, any I/O access to the Serial Ports (3F8-3FFh or 3E8-3EFh) will reload the System Standby Timer. |
| 27 | R/W | **Reload Enable—Serial Port 2 Access**<br>When set, any I/O access to the Serial Ports (2F8-2FFh, 2E8-2EFh) will reload the System Standby Timer. |
| 26 | R/W | **Reload Enable—Parallel Port Access**<br>When set, any I/O access to the Parallel ports (278-27Fh, 378-37Fh or 3BC-3BEh) will reload the System Standby Timer. |
| 25 | R/W | **Reload Enable—DMA or USB Master Request Event**<br>When set, either DMA or USB master raising a PCI request event will reload the System Standby Timer. |
| 24 | R/W | **Reload Enable--IRQ 0, 1, 3~15, NMI Events**<br>When set, any of the IRQ0,1,3-15 or NMI events as enabled in Reg. 68h~6Bh will reload the System Standby Timer. |
| 23 | R/W | **Reload Enable--Ring In Event**<br>When set, an active RING signal will reload the System Standby Timer. |
| 22 | R/W | **Reload Enable—Programmable 10-bit I/O Port Access**<br>When set, any access to I/O ports defined in Registers 5Ch ~ 5Dh will reload the System Standby Timer. |

| | | |
|---|---|---|
| 21 | R/W | **Reload Enable—Programmable 16-bit I/O Port Access**<br>When set, any I/O access to the address range defined in Registers 5Eh ~ 5Fh will reload the System Standby Timer. |
| 20 | R/W | **Reload Enable—Memory Address A0000h-AFFFFh, B0000-BFFFFh Access**<br>When set, any memory access which falls within A0000h ~ AFFFFh, B0000h ~ BFFFFh will reload the System Standby Timer. |
| 19 | R/W | **Reload Enable—Memory Address C0000h – C7FFFh Access**<br>When set, any memory access which falls within C000h-C7FFFh will reload the System Standby Timer. |
| 18 | R/W | **Reload Enable--VGA I/O Port 3B0-3BFh, 3C0-3CFh, 3D0-3DFh Access**<br>When set, any I/O access to ports 3B0-3BFh, 3C0-3CFh, 3D0-3DFh will reload the System Standby Timer. |
| 17 | R/W | **Reload Enable—Linear Frame Buffer Address**<br>When set, any memory access to the Linear Frame Buffer address range defined at register 4Eh~4Fh will reload the System Standby Timer. |
| 16 | R/W | **Reload Enable—Microsoft Sound Port Access**<br>When set, any I/O access to the address range defined in Register 4Ch~4Dh, bit 6~7 will reload the System Standby Timer. |
| 15 | R/W | **Reload Enable—Sound Blaster Port Access**<br>When set, any I/O access to the address range defined in Register 4Ch~4Dh, bit 4~5 will reload the System Standby Timer. |
| 14 | R/W | **Reload Enable--MIDI Port Access**<br>When set, any I/O access to the address range defined by Register 4Ch~4Dh bit 2~3 will reload the System Standby Timer. |
| 13 | R/W | **Reload Enable--Game Port Access**<br>When set, any I/O access to address 200-207h, 388-38Bh will reload the System Standby Timer. |
| 12 | R/W | **Reload Enable--GPCS0# Active Event**<br>The GPCS0# and GPCS1# can be programmed as input or output mode. An active event in input mode is defined as an active-level signal being applied on the pin. An active event in output mode is when GPCS#/GPCSW# function is chosen, and an active-low pulse is being generated on the pin. When this bit is set to 1, an active event will reload the System Standby Timer. |
| 11 | R/W | **Reload Enable—GPCS1# Active Event**<br>Refer to GPCS0#. |
| 10 | R/W | **Reload Enable—INIT Active Event**<br>When set, an active INIT will reload the System Standby Timer. |

| | | |
|---|---|---|
| 9 | R/W | **Reload Enable—EXTSMI# Event**<br>When set, an active EXTSMI# will reload the System Standby Timer. |
| 8 | R/W | **Reload Enable--PCI/AGP/IDE Master Request Events**<br>When set, any PCI/AGP/IDE master requests from North Bridge will reload the System Standby Timer. The PCI/AGP/IDE Master request information is carried on the BM_REQ# signal from North Bridge. |
| 7 | R/W | **Reload Enable--AGP cycle**<br>When set, any AGP activity events from North Bridge will reload the System Standby Timer. |
| 6 | R/W | **Reload Enable—Floppy Ports Access**<br>When set, any I/O access to the ports 3F0-3F7h, 370-377h, or an active DRQ2 will reload the System Standby Timer. |
| 5 | R/W | **Reload Enable—Primary IDE Channel Device 1 Access**<br>When set, any I/O access to the primary IDE ports 1F0-1F7h, 3F6h, IDE bus master ports with DEV=1 in Device/Head Register will reload the System Standby Timer. |
| 4 | R/W | **Reload Enable—Secondary IDE Channel Device 1 Access**<br>When set, any I/O access to the secondary IDE related ports 170-177h, 376h, or IDE bus master ports with DEV=1 in Device/Head Register will reload the System Standby Timer. |
| 3 | R/W | **Reload Enable--GPIO Active Event**<br>When set, any active GPIO signal enabled by ACPI Reg. 2Eh and 2Fh will reload the System Standby Timer. |
| 2:0 | R/W | **Reserved. These bits must be programmed to 0.** |

**Register 4C ~ 4Dh Sound Port Trap and Address Mask for the Programmable 16-bit IO**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:8 | R/W | **Mask Bits for the Programmable 16-bit I/O Port.**<br>These eight bits are mask bits for A[7:0] of the Programmable 16-bit I/O port address defined in Reg. 5Eh~5Fh. A '1' will cause the associated address bits to be masked (ignored) when doing address comparison. |
| 7:6 | R/W | **Microsoft Sound Port Trap Selection**<br>00 : F40h-F47h<br>01 : E80h-E87h<br>10 : 604h-60Bh<br>11 : 530h-537h |

| | | |
|---|---|---|
| 5:4 | R/W | **Sound Blaster Port Trap Selection**<br>00 : 280h-28Fh, 290h-293h<br>01 : 260h-26Fh, 270h-273h<br>10 : 240h-24Fh, 250h-253h<br>11 : 220h-22Fh, 230h-233h |
| 3:2 | R/W | **MIDI Port Trap Selection**<br>00 : 330h-333h<br>01 : 320h-323h<br>10 : 310h-313h<br>11 : 300h-303h |
| 1 | R/W | **Reserved.**<br>These bits must be programmed to 0. |
| 0 | R/W | **APC Register 08h cleared control**<br>0 : Enable<br>1 : Disable<br>If this bit is not set to 1 in 4.68 seconds after CPURST de-assertion, the APC Reg.08h will be cleared to default value. |

**Register 4Eh ~ 4Fh Linear Frame Buffer Trap Address**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:14 | R/W | **Reserved.**<br>These bits must be programmed to 0. |
| 13:12 | R/W | **Linear Frame Buffer Mask**<br>The two bits are used for the masking of A21 and A20 respectively of the Linear Frame Buffer. When set to 1 will enable the mask. |
| 11:0 | R/W | **Linear Frame Buffer Base Address**<br>Correspond to A[31:20] of the Base Address for the Linear Frame Buffer. |

**Register 50h ~ 53h Wake-up Event 0 Control register**

**Register 54h ~ 57h Wake-up Event 1 Control register**

Default Value : 00000000h

Access : Read/Write

Active events enabled by these two register sets will cause the corresponding status bits (Bit28 or 29) at Reg. 64h~67h to be set. Furthermore, active events enabled in Reg. 50h~53h will cause the system to exit C2, C3, S1 or S2 states.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **Wake-up Enable—Primary IDE Channel Device 0 Access**<br>When set, any I/O access to the primary IDE channel device 0 associated ports will cause the wake-up status bit to be set. |
| 30 | R/W | **Wake-up Enable—Secondary IDE Channel Device 0 Access**<br>When set, any I/O access to the secondary IDE channel device 0 associated ports will cause the wake-up status bit to be set. |
| 29 | R/W | **Wake-up Enable—Keyboard port Access**<br>When set, any I/O access to the keyboard ports (60h, 64h) will cause the wake-up status bit to be set. |
| 28 | R/W | **Wake-up Enable—Serial Port 1 Access**<br>When set, any I/O access to the Serial ports (3F8-3FFh, 3E8-3EFh) will cause the wake-up status bit to be set. |
| 27 | R/W | **Wake-up Enable--Serial Port 2 Access**<br>When set, any I/O access to the Serial ports (2F8-2FFh, 2E8-2EFh) will cause the wake-up status bit to be set. |
| 26 | R/W | **Wake-up Enable—Parallel Port Access**<br>When set, any I/O access to the Parallel ports (278-27Fh, 378-37Fh or 3BC-3BEh) will cause the wake-up status bit to be set. |
| 25 | R/W | **Wake-up Enable—DMA or USB Master Request Event**<br>When set, a DMA or USB master request for PCI bus event will cause the wake-up status bit to be set. |
| 24 | R/W | **Wake-up Enable--IRQ 0, 1, 3~15, NMI Active Event**<br>When set, any events from IRQ0, 1, 3-15 or NMI, with their enable-bits in Register 68h~6Dh being enabled, will cause the wake-up status bit to be set. |
| 23 | R/W | **Wake-up Enable--Ring In Active Event**<br>When set, an active event at the RING input will cause the wake-up status bit to be set. |
| 22 | R/W | **Wake-up Enable--Programmable 10 bit I/O Port Access Event**<br>When set, any I/O access to the address defined in Reg. 5Ch ~ 5Dh will cause the wake-up status bit to be set. |
| 21 | R/W | **Wake-up Enable--Programmable 16 bit I/O Port Access Event**<br>When set, any I/O access to the address defined in Reg. 5Eh ~ 5Fh will cause the wake-up status bit to be set. |
| 20 | R/W | **Wake-up Enable--Memory Address A0000h~AFFFFh, B0000 ~BFFFFh Access**<br>When set, any memory access which falls within A0000h ~AFFFFh, B0000h~BFFFFh will cause the wake-up status bit to be set. |

| | | |
|---|---|---|
| 19 | R/W | **Wake-up Enable--Memory Address C0000h~C7FFFh Access**<br>When set, any memory access which falls within C0000h ~C7FFFFh will cause the wake-up status bit to be set. |
| 18 | R/W | **Wake-up Enable--VGA I/O Port 3B0-3BFh, 3C0-3CFh, 3D0-3DFh Access**<br>When set, any I/O access to VGA I/O Ports 3B0-3BFh, 3C0-3CFh, 3D0-3DFh will cause the wake-up status bit to be set. |
| 17 | R/W | **Wake-up Enable--Linear Frame Buffer Address Access**<br>When set, any memory access to Linear Frame Buffer address will cause the wake-up status bit to be set. |
| 16 | R/W | **Wake-up Enable—Microsoft Sound Port Access**<br>When set, any I/O access to the address defined in Register 4Ch ~ 4Dh, bit 7~6 will cause the wake-up status bit to be set. |
| 15 | R/W | **Wake-up Enable--Sound Blaster Port Access**<br>When set, any I/O access to the address defined in Register 4Ch ~ 4Dh, bit 5~4 will cause the wake-up status bit to be set. |
| 14 | R/W | **Wake-up Enable--MIDI Port Access**<br>When set, any I/O access to the address defined in Register 4Ch ~ 4Dh, bit 3~2 will cause the wake-up status bit to be set. |
| 13 | R/W | **Wake-up Enable—Game Port Access**<br>When set, any I/O access to 200-207h, 388-38Bh will cause the wake-up status bit to be set. |
| 12 | R/W | **Wake-up Enable--GPCS0# Event**<br>When set, an active input to GPCS0# or GPCS0# in GPCS# /GPCSW# mode being output-low will cause the wake-up status bit to be set. |
| 11 | R/W | **Wake-up Enable—GPCS1# Event**<br>When set, an active input to GPCS1# or GPCS1# in GPCS# /GPCSW# mode being output-low will cause the wake-up status bit to be set. |
| 10 | R/W | **Wake-up Enable—INIT Event**<br>When set, an active INIT will cause the wake-up status bit to be set. |
| 9 | R/W | **Wake-up Enable—EXTSMI# Event**<br>When set, an active EXTSMI# will cause the wake-up status bit to be set. |
| 8 | R/W | **Wake-up Enable—PCI or AGP Master Active Event**<br>When set, any PCI/AGP/IDE master requests from North Bridge will cause the wake-up status bit to be set. |
| 7 | R/W | **Wake-up Enable—AGP cycle**<br>When set, any AGP activity events from North Bridge will cause the wake-up status bit to be set. |

| | | |
|---|---|---|
| 6 | R/W | **Wake-up Enable—Floppy port Access**<br>When set, any I/O access to the ports 3F0-3F7h, 370-377h, or an active DRQ2 will cause the wake-up status bit to be set. |
| 5 | R/W | **Wake-up Enable—Primary IDE Channel Device 1**<br>When set, any I/O access to the primary IDE channel device 1 associated ports will cause the wake-up status bit to be set. |
| 4 | R/W | **Wake-up Enable—Secondary IDE Channel Device 1**<br>When set, any I/O access to the secondary IDE channel device 1 associated ports will cause the wake-up status bit to be set. |
| 3:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 58h ~ 5Bh Reserved**

Default Value : 00000000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 5Ch ~ 5Dh Programmable 10-bit I/O Port Trap Address**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:10 | R/W | **Programmable 10-bit I/O Port Trap Address Mask Bits**<br>These bits correspond to the mask bits for A[5:0] of the 10-bit I/O Port Trap Address. A 1 in the bit position indicates the corresponding address bit should be masked (ignored) when doing address comparison. |
| 9:0 | R/W | **Programmable 10-bit I/O Port Trap Address Base**<br>These bits correspond to A[9:0] of the base address. Note that a matched address must have its A[15:10] set to all 0s. |

**Register 5Eh ~ 5Fh Programmable 16-bit I/O Port Trap Address**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Correspond to A[15:0] for the Programmable 16-bit I/O port**<br>Note that there are five address mask bits defined in Reg. 4Ch~4Dh, bit [12:8] can be used to mask A[4:0] defined in this register. |

**Register 60h ~ 63h SMI# Enable**

Default Value : 00000000h

Access : Read/Write

A SMI# will be generated when the following SMI# control bit is enabled, the corresponding status bit at Reg. 64h~67h is set by associated event and Bit_5 of APC Reg. 31h (SMI# Enable) is set.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **SMI# Enable--System Standby Timer 0 Expires**<br>When set, the System Standby Timer 0 expire event will generate the SMI#. |
| 30 | R/W | **SMI# Enable--System Standby Timer 1 Expires**<br>When set, the System Standby Timer 1 expire event will generate the SMI#. |
| 29 | R/W | **SMI# Enable—System Standby Timer 2 Expires**<br>When set, the System Standby Timer 2 expire event will generate the SMI#. |
| 28 | R/W | **SMI# Enable--Wake-up 0 Event**<br>When set, any active events defined in Reg. 50h~53h will generate the SMI#. |
| 27 | R/W | **SMI# Enable--Wake-up 1 Event**<br>When set, any active events defined in Reg. 54h~57h will generated the SMI#. |
| 26 | R/W | **Reserved.**<br>This bit must be programmed to 0. |
| 25 | R/W | **SMI# Enable--Primary IDE Channel Device 0 Access**<br>When set, any I/O access to the port associated with the primary IDE channel device 0 will generate the SMI#. |
| 24 | R/W | **SMI# Enable—Secondary IDE Channel Device 0 Access**<br>When set, any I/O access to the ports associated with the secondary IDE channel device 0 will generate the SMI#. |
| 23 | R/W | **SMI# Enable--Keyboard Ports Access**<br>When set, any I/O access to the keyboard ports will generate the SMI#. |
| 22 | R/W | **SMI# Enable--Serial port 1 Access**<br>When set, any I/O access to the serial port 1 will generate the SMI#. |
| 21 | R/W | **SMI# Enable--Serial port 2 Access**<br>When set, any I/O access to the serial port 2 will generate the SMI#. |

| | | |
|---|---|---|
| 20 | R/W | **SMI# Enable--Parallel ports Access**<br>When set, any I/O access to the parallel ports will generate the SMI#. |
| 19 | R/W | **SMI# Enable--RING Active Event**<br>When set, an active RING event will generate the SMI#. |
| 18 | R/W | **SMI# Enable—Programmable 10-bit I/O Port Access**<br>When set, any I/O access to the port defined in the programmable 10-bit I/O will generate the SMI#. |
| 17 | R/W | **SMI# Enable—Programmable 16-bit I/O Port Access**<br>When set, any I/O access to the port defined in the programmable 16-bit I/O will generate the SMI#. |
| 16 | R/W | **SMI# Enable--Memory address A0000-BFFFFh Access**<br>When set, any memory access to these addresses will generate the SMI#. |
| 15 | R/W | **SMI# Enable--Memory address C0000-C7FFFh Access**<br>When set, any memory access to these addresses will generate the SMI#. |
| 14 | R/W | **SMI# Enable--VGA I/O port 3B0-3DFh Access**<br>When set, any I/O access to these ports will generate the SMI#. |
| 13 | R/W | **SMI# Enable--Linear Frame Buffer Address Access**<br>When set, any memory access to the linear frame buffer address will generate the SMI#. |
| 12 | R/W | **SMI# Enable--Microsoft Sound Port Access**<br>When set, any I/O access to the port defined by Register 4Ch ~ 4Dh bit 7~6 will generate the SMI#. |
| 11 | R/W | **SMI# Enable--Sound Blaster Port Access**<br>When set, any I/O access to the port defined by Register 4Ch ~ 4Dh bit 5~4 will generate the SMI#. |
| 10 | R/W | **SMI# Enable--MIDI Port Access**<br>When set, any I/O access to the port defined by Register 4Ch ~ 4Dh bit 3~2 will generate the SMI#. |
| 9 | R/W | **SMI# Enable--Game Port Access**<br>When set, any I/O access to 200h-207h, 388h-38Bh will generate the SMI#. |
| 8 | R/W | **SMI# Enable--GPCS0# Active**<br>When set, an active input to GPCS0# or GPCS0# in GPCS#/GPCSW# mode being output-low will generate the SMI#. |
| 7 | R/W | **SMI# Enable--GPCS1# Active**<br>When set, an active input to GPCS1# or GPCS1# in GPCS#/GPCSW# mode being output-low will generate SMI#. |
| 6 | R/W | **SMI# Enable--EXTSMI# Active**<br>When set, the active EXTSMI# will generate the SMI#. |

| | | |
|---|---|---|
| 5 | R/W | **SMI# Enable--Floppy port Access**<br>When set, any I/O access to the port 3F0-3F7h, 370-377h or an active DREQ2 event will generate the SMI#. |
| 4 | R/W | **SMI# Enable--Primary IDE Channel Device 1 Access**<br>When set, any I/O access to the address associated with the primary IDE channel device 1 will generate the SMI#. |
| 3 | R/W | **SMI# Enable--Secondary IDE Channel Device 1 Access**<br>When set, any I/O access to the address associated with the secondary IDE channel device 1 will generate the SMI#. |
| 2 | R/W | **SMI# Enable—APM**<br>When this bit and Reg. 64h~67h, bit 2 are both set, a SMI# will be generated. |
| 1:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 64h - 67h SMI#/SCI Request Status**

Default Value : 00000000h

Access : Read/Write

The SMI#/SCI request status bit will be set when the associated event is active. The status bit can be cleared by writing a 1.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **System Standby Timer 0 Status**<br>This bit is set when the System Standby Timer 0 expires. |
| 30 | R/W | **System Standby Timer 1 Status**<br>This bit is set when the System Standby Timer 1 expires. |
| 29 | R/W | **System Standby Timer 2 Status**<br>This bit is set when the System Standby Timer 2 expires. |
| 28 | R/W | **Wake-up 0 Status**<br>This bit is set when any un-masked wake-up events for Wake up 0 SMI# enable is active |
| 27 | R/W | **Wake-up 1 Status**<br>This bit is set when any un-masked wake-up events for Wake up 1 SMI# enable is active |
| 26 | R/W | **Reserved.**<br>This bit must be programmed to 0. |
| 25 | R/W | **Primary IDE channel Device 0 Status**<br>This bit is set when the primary IDE channel device 0 is accessed. |
| 24 | R/W | **Secondary IDE channel Device 0 Status**<br>This bit is set when the secondary IDE channel device 0 is accessed. |

| | | |
|---|---|---|
| 23 | R/W | **Keyboard Ports Status**<br>This bit is set when any of the keyboard port is accessed. |
| 22 | R/W | **Serial Port 1 Status**<br>This bit is set when any of the serial port 1 is accessed. |
| 21 | R/W | **Serial Port 2 Status**<br>This bit is set when any of the serial port 2 is accessed. |
| 20 | R/W | **Parallel Ports Status**<br>This bit is set when any of the parallel ports is accessed. |
| 19 | R/W | **RING SMI# request**<br>This bit is set when the RING is active. |
| 18 | R/W | **Programmable 10-bit I/O port Status**<br>This bit is set when there is an I/O access to the programmable 16-bit I/O port. |
| 17 | R/W | **Programmable 16-bit I/O Port Status**<br>This bit is set when there is an I/O access to the programmable 16-bit I/O port. |
| 16 | R/W | **Memory Address A0000-BFFFFh Status**<br>This bit is set when memory access to the address range occurs. |
| 15 | R/W | **Memory Address C0000-C7FFFh Status**<br>This bit is set when memory access to the address range occurs. |
| 14 | R/W | **VGA I/O Port 3B0-3DFh Status**<br>This bit is set when I/O access to these ports occurs. |
| 13 | R/W | **Linear Frame Buffer Address Status**<br>This bit is set when memory access to the linear frame buffer address occurs. |
| 12 | R/W | **Microsoft Sound Port Status**<br>This bit is set when I/O access to the address range defined by bit 7~6 of Reg. 4Ch~4Dh occurs. |
| 11 | R/W | **Sound Blaster Port Status**<br>This bit is set when I/O access to the address range defined by bit 5~4 of Reg. 4Ch~4Dh occurs. |
| 10 | R/W | **MIDI Port Status**<br>This bit is set when I/O access to the address range defined by bit 3~2 of Reg. 4Ch~4Dh occurs. |
| 9 | R/W | **Game Port Status**<br>This bit is set when I/O access to 200h-207h, 388h-38Bh occurs. |
| 8 | R/W | **GPCS0# Status**<br>This bit is set when an active input to GPCS0# (GPCS0# in input mode) or GPCS0# is output-low (GPCS0# in GPCS#/GPCSW# output mode). |

| | | |
|---|---|---|
| 7 | R/W | **GPCS1# Status**<br>This bit is set when an active input to GPCS1# (GPCS1# in input mode) or GPCS1# is output-low (GPCS1# in GPCS#/GPCSW# output mode). |
| 6 | R/W | **EXTSMI# Status**<br>This bit is set when the EXTSMI# is active. |
| 5 | R/W | **Floppy Port Status**<br>This bit is set when I/O access to the ports 3F0-3F7h, 370-377h or an active DREQ2 occurs. |
| 4 | R/W | **Primary IDE channel Device 1 Status**<br>This bit is set when any of the primary IDE channel device 1 associated ports is accessed. |
| 3 | R/W | **Secondary IDE channel Device 1 Status**<br>This bit is set when any of the secondary IDE channel device 1 associated ports is accessed. |
| 2 | R/W | **APM Status.**<br>This bit is set when a 1 is written to Bit_1 of Reg. 7Ch. |
| 1:0 | R/W | **Reserved.**<br>This bit must be programmed to 0. |

**Register 68h ~ 69h/6Ah~6Bh** **IRQ and NMI Enable for Wake-up 0/1, and System Standby Timer 0/1 reload event control**

Default Value : 0000h

Access : R/W

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:3 | R/W | **Correspond to the enable bits for IRQ15-3 to reload the System Standby Timer 0, 1 or to generate a wake-up event.**<br>0 : Enable<br>1 : Disable |
| 2 | R/W | **Corresponds to the enable bits for NMI to reload the System Standby Timer 0, 1 or to generate wakeup event.**<br>0 : Enable<br>1 : Disable |
| 1:0 | R/W | **Correspond to the enable bits for IRQ1-0 to reload the System Standby Timer 0, 1 or to generate a wakeup event.**<br>0 : Enable<br>1 : Disable |

**Register 6Ch ~ 6Dh** **IRQ and NMI to Reload System Standby Timer 2 Enable**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:3 | R/W | **Correspond to the enable bits for IRQ15-3 to reload the System Standby Timer 2**<br>0 : Enable<br>1 : Disable |
| 2 | R/W | **Corresponds to the enable bit for NMI to reload the System Standby Timer 2**<br>0 : Enable<br>1 : Disable |
| 1:0 | R/W | **Correspond to the enable bits for IRQ1-0 to reload the System Standby Timer 2**<br>0 : Enable<br>1 : Disable |

**Register 6E ~ 6Fh Reserved**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 70 ~ 71h GPCS0# Base Address Register**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Base Address A[15:0] for GPCS0# in GPCS# or GPCSW# mode.** |

**Register 72h GPCS0# Base Address Mask Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Mask bits for GPCS0# Base Address A[7:0].** |

**Register 73h GPCS0# Control Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **GPCS0# Mode Control**<br>0 : Input mode<br>1 : Output mode |
| 6 | R/W | **GPCS0# Input Polarity Control**<br>0 : Active low<br>1 : Active high |
| 5 | R/W | **GPCS0# Input De-Bounce Filter Control**<br>0 : Disable<br>1 : Enable<br>When set to 1, the internal de-bounce circuit for GPCS0# input is enabled. |
| 4 | R/W | **GPCS0# Output Status Control for GPO mode**<br>0 : Output low<br>1 : Output high |
| 3 | RO | **GPCS0# Status**<br>This bit reflects the status of GPCS0#. |
| 2:1 | R/W | **GPO/GPCS#/GPCSW# selection**<br>00: GPO<br>01: GPCS#<br>10: GPO<br>11: GPCSW#<br>When the ISA address fallen in the range defined by Reg. 70h~72h will cause GPCS0# to output low when GPCS# function is selected.<br>When IOWC# is active and the ISA address fallen in the range defined by Reg. 70h~72h will cause GPCS0# to output low when GPCSW# function is selected.<br>If GPCSW# is selected, this pin can be used to control an external 74LS374 TTL to latch the values on SD[7:0]. In this way, the number of general-purpose outputs can be expanded to 8.<br>The GPCS0# have another definition, refer to PMU Reg. 77h bit 0. |
| 0 | R/W | **GPCS0# and GPCS1# Test Mode**<br>This bit should be programmed to 0 for normal operation. |

**Register 74 ~ 75h GPCS1# Base Address Register**

Default Value : 0000h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Base Address A[15:0] for GPCS1# in GPCS# or GPCSW# mode.** |

**Register 76h GPCS1# Base Address Mask Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Mask bits for GPCS1# Base Address A[7:0].** |

**Register 77h GPCS1# Control Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **GPCS1# Mode Control**<br>0 : Input mode<br>1 : Output mode<br>Since the KLOCK# and GPCS1# function share the same pin, when the Key-Lock function is selected, this bit must be set to 0. |
| 6 | R/W | **GPCS1# Input Polarity Control**<br>0 : Active low<br>1 : Active high |
| 5 | R/W | **GPCS1# Input De-Bounce Filter Control**<br>0 : Disable<br>1 : Enable<br>When set to 1, the internal de-bounce circuit for the GPCS1# input is enabled. |
| 4 | R/W | **GPCS1# Output Status Control for GPO mode**<br>0 : Output low<br>1 : Output high |
| 3 | RO | **GPCS1#Status**<br>This bit reflects the status of GPCS1#. |

| | | |
|---|---|---|
| 2:1 | R/W | **GPO/GPCS#/GPCSW# selection**<br>00: GPO<br>01: GPCS#<br>10: GPO<br>11: GPCSW#<br>When the ISA address fall in the range defined by Reg. 74h~76h will cause GPCS1# to output low when GPCS# function is selected.<br>When IOWC# is active and the ISA address fall in the range defined by Reg. 74h~76h will cause GPCS1# to output low when GPCSW# function is selected.<br>If GPCSW# is selected, this pin can be used to control an external 74LS374 TTL to latch the values on SD[7:0]. In this way, the number of general-purpose outputs can be expanded to 8. |
| 0 | R/W | **GPCS0# quick control**<br>When this bit is enabled and GPCS0# is programmed as output mode, the GPCS0# is driven from ORGated ( SA1, SA12, SA13, SA14, SA15).<br>0 : Disable<br>1 : Enable |

**Register 78h GPCS0#/GPCS1# De-bounce Counter Control Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **GPCS0# address A[11:8] mask control** |
| 3:0 | R/W | **GPCS0# and GPCS1# De-bounce duration timer**<br>Bit[3:0] specify the duration of de-bounce in multiples of 4.58ms. |

**Register 79h/7Ah/7Bh System Standby Timer 0/1/2 Reading Register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | The maximum counting value for these timers is FFh. When the System Standby Timer expires, a SMI# or SCI will be generated.<br>The following three conditions will cause the System Standby Timer to stop counting and return initial count value when read.<br>1.Both the associated SMI# and SCI enable bits are disabled.<br>2.An enabled reload event is active.<br>3.The corresponding System Standby Timer status bit at Reg. 64h~67h is set. |

**Register 7Ch System Standby Timer Granularity Control register**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **System Standby Timer 0 Granularity Selection**<br>11 : 17.8 us<br>10 : 4.58 ms<br>01 : 1.17 s<br>00 : 5 min. |
| 5:4 | R/W | **System Standby Timer 1 Granularity Selection**<br>11 : 17.8 us<br>10 : 4.58 ms<br>01 : 1.17 s<br>00 : 5 min. |
| 3:2 | R/W | **System Standby Timer 2 Granularity Selection**<br>11 : 17.8 us<br>10 : 4.58 ms<br>01 : 1.17 s<br>00 : 5 min. |
| 1 | R/W | **APM_SMI# Function Control**<br>Writing a 1 to this bit will cause the APM status bit in Reg. 64h~67h to be set. |
| 0 | R/W | **Reserved.**<br>This bit must be programmed to 0. |

**Register 7Dh Auto Power-Off Timer and PMU Test Mode**

Default Value : 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **Auto Power-Off Timer**<br>The resolution of this timer is 4 minute.<br>The Auto Power-off Timer, located in APC, is accessible through this register. The Auto Power-off Timer can be used to turn off the power for SDRAM. Should the APC Register bits [6:5] set to '10' and the APO time is programmed, the following sequence of events will occur when the system enters S3 state.<br>1. Assert STPCLK#<br>2. Assert CKE_N<br>3. Assert CKE_S<br>4. Turn off main power by de-asserting PS_ON#.<br>If there is no power-up event occurred before the APO timer expires, the SDRAM power will be off by GPIO5 when the APO timer expires. |
| 3:2 | R/W | **Legacy PMU Test Mode**<br>00: Normal operation<br>01: Counter test mode<br>10: Fast test mode<br>11: Clear Counter and SPKR is used to monitor internal signal |
| 1 | R/W | **EXTSMI# Polarity Selection**<br>0 : active low<br>1 : active high |
| 0 | R/W | **PMU Test Mode**<br>0 : Normal Mode<br>1 : Test Mode |

**Register 7E ~ 7Fh IDE Bus Master Port Trap**

Default Value : 0000h

Access : Read/Write

This register is used to define whether an access to IDE Bus Master ports should be recognized as an active event at the PMU and thereby reload the System Standby Timer or generate the SMI#/SCI.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Define A[15:4] for IDE Bus Master Base Address trap**<br>These registers should be programmed with the same value as the base address for IDE bus master at the North bridge. |

| | | |
|---|---|---|
| 3 | R/W | **Primary IDE Bus Master Port Trap Control**<br>0 : Disable<br>1 : Enable<br>When enabled, any I/O access to the primary IDE Bus Master ports will be recognized as an active event on IDE primary channel. |
| 2 | R/W | **Secondary IDE Bus Master Port Trap Control**<br>0 : Disable<br>1 : Enable<br>When enabled, any I/O access to the secondary IDE Bus Master ports will be recognized as an active event on IDE secondary channel. |
| 1:0 | R/W | **Reserved.**<br>These bits must be programmed to 0. |

**Register 80h~ 83h SCI Enable**

Default Value : 0000h

Access : Read/Write

A SCI will be generated when the following SCI control bit is enabled together with the corresponding status bit in Reg. 64h~67h is set by the associated event.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **SCI Enable-System Standby Timer 0 Expire Event**<br>When set, the System Standby Timer 0 expire event will generate the SCI. |
| 30 | R/W | **SCI Enable-System Standby Timer 1 Expire Event**<br>When set, the System Standby Timer 1 expire event will generate the SCI. |
| 29 | R/W | **SCI Enable-System Standby Timer 2 Expire Event**<br>When set, the System Standby Timer 2 expire event will generate the SCI. |
| 28 | R/W | **SCI Enable-Wake up 0 Event**<br>When set, an active event defined in register 50h ~ 53h will generate the SCI. |
| 27 | R/W | **SCI Enable-Wake up 1 Event**<br>When set, an active event defined at register 54h ~ 57h will generate the SCI. |
| 26 | R/W | **SCI Enable-Reserved. This bit must be programmed to 0.** |
| 25 | R/W | **SCI Enable-Primary IDE Channel Device 0 Access**<br>When set, any I/O access to the address associated with the primary IDE channel device 0 will generate the SCI. |

| | | |
|---|---|---|
| 24 | R/W | **SCI Enable-Secondary IDE Channel Device 0 Access**<br>When set, any I/O access to the address associated with the secondary IDE channel device 0 will generate the SCI. |
| 23 | R/W | **SCI Enable-Keyboard Ports Access**<br>When set, any I/O access to the keyboard ports will generate the SCI. |
| 22 | R/W | **SCI Enable-Serial port 1 Access**<br>When set, any I/O access to the serial port 1 will generate the SCI. |
| 21 | R/W | **SCI Enable-Serial port 2 Access**<br>When set, any I/O access to the serial port 2 will generate the SCI. |
| 20 | R/W | **SCI Enable-Parallel Ports Access**<br>When set, any I/O access to the parallel ports will generate the SCI. |
| 19 | R/W | **SCI Enable-RING Active Event**<br>When set, an active RING event will generate the SCI. |
| 18 | R/W | **SCI Enable-Programmable 10-bit I/O Ports Access**<br>When set, any I/O access to the ports defined in the programmable 10-bit I/O will generate the SCI. |
| 17 | R/W | **SCI Enable-Programmable 16-bit I/O Ports Access**<br>When set, any I/O access to the port defined in the programmable 16-bit I/O will generate the SCI. |
| 16 | R/W | **SCI Enable-Memory address A0000h-BFFFFh Access**<br>When set, any memory access to these address will generate the SCI. |
| 15 | R/W | **SCI Enable-Memory address C0000h-C7FFFh Access**<br>When set, any memory access to these address will generate the SCI. |
| 14 | R/W | **SCI Enable-VGA I/O port Access**<br>When set, any I/O access to this port will generate the SCI. |
| 13 | R/W | **SCI Enable-Linear Frame Buffer Address Access**<br>When set, any memory access to the linear frame buffer will generate the SCI. |
| 12 | R/W | **SCI Enable-Microsoft Sound Port Access**<br>When set, any I/O access to the ports defined in bit 7~6 of Register 4Ch ~ 4Dh will generate the SCI. |
| 11 | R/W | **SCI Enable-Sound Blaster Port Access**<br>When set, any I/O access to the ports defined in bit 5~4 of Register 4Ch ~ 4Dh will generate the SCI. |
| 10 | R/W | **SCI Enable-MIDI Port Access**<br>When set, any I/O access to the ports defined in bit 3~2 of Register 4Ch ~ 4Dh will generate the SCI. |

| | | |
|---|---|---|
| 9 | R/W | **SCI Enable-Game Port Access**<br>When set, any I/O access to 200-207h, 388-38Bh will generate the SCI. |
| 8 | R/W | **SCI Enable-GPCS0# Active Event**<br>When set, an active input to GPCS0# or GPCS0# in GPCS# /GPCSW# mode being output-low will generate the SCI. |
| 7 | R/W | **SCI Enable-GPCS1# Active Event**<br>When set, an active input to GPCS1# or GPCS1# in GPCS# /GPCSW# mode being output-low will generate the SCI. |
| 6 | R/W | **SCI Enable-EXTSMI# Active Event**<br>When set, an active EXTSMI# will generate the SCI. |
| 5 | R/W | **SCI Enable-Floppy Ports Access**<br>When set, any I/O access to the port 3F0-3F7h, 370-377h, or active DREQ2 will generate the SCI. |
| 4 | R/W | **SCI Enable-Primary IDE Channel Device 1 Access**<br>When set, any I/O access to the address associated with the primary IDE channel device 1 will generate the SCI. |
| 3 | R/W | **SCI Enable-Secondary IDE Channel Device 1 Access**<br>When set, any I/O access to the address associated with the secondary IDE channel device 1 will generate the SCI. |
| 2:0 | R/W | **Reserved.**<br>This bit must be programmed to 0. |

### 7.3 ACPI CONFIGURATION REGISTERS

The following registers located at I/O base address <Base> + the indicated offset value <Offset>. The base address is programmed in the Register 90h~91h of PCI-to-ISA Configuration space.

**Register 00h Power Management Status Register(PM1_STS)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15 | R/W | **Wake up Status (WAK_STS)**<br>This bit is set when the system is in the sleeping state and an enabled wake-up event occurs or set by a specific programming sequenc to APCI register 38h and 39h. Upon setting this bit, the state machine will transition the system into the on state. This bit can only be set by hardware and only can be cleared by software writing a “1” to this bit position. |
| 14:12 | RO | **Reserved.** |
| 11 | R/W | **Ignored. (Power Button Over-ride Status)** |

| | | |
|---|---|---|
| 10 | R/W | **RTC Status (RTC_STS)**<br>This bit is set when the RTC generates an alarm or set by a specific programming sequenc to APCI register 38h and 39h. While both RTC_EN bit and RTC_STS bit are set, a power management event is raised(SCI, SMI# or WAKE event). This bit is only set by hardware and only reset by software writing a "1" to this bit position. |
| 9 | RO | **Reserved.** |
| 8 | R/W | **Power Button Status (PWRBTN_STS)**<br>This bit is set when the power button is pressed (the PWRBT# signal is asserted Low) or set by a specific programming sequenc to APCI register 38h and 39h. In the working state, while both PWRBTN_STS bit and PWRBTN_EN bit are set,then a SCI or SMI# is raised. In the sleeping state, while PWRBTN_STS bit is set then a WAKE event is generated. This bit is only set by hardware and can only be reset by software writing a "1" to this bit position. |
| 7:6 | RO | **Reserved.** |
| 5 | R/W | **Global Status (GBL_STS)**<br>This bit is set by a BIOS-initiated SCI. BIOS can initiate a SCI by programming Register 13h bit 1 to 1. |
| 4 | R/W | **Bus Master Status(BM_STS)**<br>This is the bus master status bit. This bit is set when a system bus master is requesting the system bus, and can only be cleared by writing a "1" to this bit position. |
| 3:1 | RO | **Reserved** |
| 0 | R/W | **Power Management Timer Status (TMR_STS)**<br>Power management timer status or DOZE timer status. The free running timer will be DOZE timer when the Offset Register 2A bit 9 is set to 1 and SCI_EN bit is set to 0. It will be the power management timer otherwise. If the most significant bit of the 24-bit 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 event (SCI or SMI#) is raised. It can only be cleared by writing a '1' to this bit position. |

**Register 02h Power Management Resume Enable Register(PM1_EN)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:11 | RO | **Reserved** |

| | | |
|---|---|---|
| 10 | R/W | **RTC Enable (RTC_EN)**<br>This bit is used to enable the assertion of the RTC_STS bit to generate a power management event. (SCI, SMI# or WAKE) |
| 9 | RO | **Reserved** |
| 8 | R/W | **Power Button Enable (PWRBTN_EN)**<br>This bit is used to enable the assertion of the PWRBTN_STS bit to generate a power management event (SCI, SMI#). For WAKE event, this bit is ignored, i.e., the system can be waked up from S1 or S2 by Power Button regardless of the value of this bit. |
| 7:6 | RO | **Reserved** |
| 5 | R/W | **Global Enable (GBL_EN)**<br>This bit is used to enable the assertion of the GBL_STS bit to generate a power management event ( SCI) |
| 4:1 | RO | **Reserved** |
| 0 | R/W | **Power Management Timer Enable (TMR_EN)**<br>This is the 24-bit 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(PM1_CTL)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:14 | RO | **Reserved** |
| 13 | WO | **Sleeping Enable (SLP_EN)**<br>This is a write-only bit and always returns a zero when read. Setting this bit to 1 will cause the system to enter the suspend state defined by the SLP_TYP field. |
| 12:10 | R/W | **Sleeping Type (SLP_TYP)**<br>Define the power-saving mode that the system should enter when the SLP_EN bit is set to one.<br>000 : S0 state (**Working**)<br>001 : S1 state ( **STPCLK#**)<br>010 : S2 state ( **Disable CPU Clock**)<br>011 : S3 state ( **Suspend To RAM**)<br>100 : S4 state ( **Suspend To Disk**)<br>101 : S5 state (**Soft_Off**) |
| 9 | RO | **Ignored** |
| 8:3 | RO | **Reserved** |

| | | |
|---|---|---|
| 2 | WO | **Global Release (GBL_RLS)**<br>This bit is used by the ACPI software to raise a SMI# to the BIOS software.<br>Also refer to Reg. 30 Bit 0 and Reg. 31 bit 0, which are used as the enable and status bits for the BIOS software to respond to ACPI events. |
| 1 | R/W | **Bus Master Reload Enable (BM_RLD)**<br>If enabled, a bus master request will cause any processor in the C3 state to transition to the C0 state.<br>0 : Disable<br>1 : Enable |
| 0 | R/W | **SCI Enable (SCI_EN)**<br>Selects the power management event to be either SCI or SMI#. When this bit is set, a power management event will generate SCI. When this bit is reset, a power management event will generate SMI#. |

**Register 06h Reserved**

Default Value: Reserved

Access: RO

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | RO | **Reserved** |

**Register 08h ACPI Power Management Timer Register(PM_TMR)**

Default Value: none

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:24 | RO | **Reserved** |
| 23:0 | RO | **Power Management Timer Value**<br>This read-only field reflects the current counting of the power management timer. The timer-expire interval is translated by the follow equation :<br>Timer-Expire Interval = (Timer Counter-1) x 0.28us |

**Register 0Ch CPU Control(P_CTL)**

Default Value: 0000 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:5 | RO | **Ignored** |
| 4 | R/W | **Throttling Function Enable**<br>This bit enables clock throttling function. |

| | | |
|---|---|---|
| 3:1 | R/W | **Throttling Duty Cycle Control**<br>This 3-bit field determines the duty cycle of the STPCLK# signal when the system in the throttling mode.<br>**Bits** — **Performance Rate**<br>000 — 100%<br>001 — 12.5%<br>010 — 25%<br>011 — 37.5%<br>100 — 50%<br>101 — 62.5%<br>110 — 75%<br>111 — 87.5% |
| 0 | RO | **Reserved** |

**Register 10h CPU Power State Level 2(P_LVL2)**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Enter C2 Power State Register**<br>Reads to this register return all zeros; writes to this register have no effect. Reads to this register will also generate a " Enter C2 power state " event. |

**Register 11h CPU Power State Level 3(P_LVL3)**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Enter C3 Power State Register**<br>Reads to this register return all zeros, writes to this register have no effect. Reads to this register also generate a " Enter C3 power state " event. |

**Register 12h PM2 Control(PM2_CTL)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:1 | RO | **Reserved** |

| | | |
|---|---|---|
| 0 | R/W | **Arbiter Disable**<br>In order to maintain the Cache coherency when CPU is in the C3 state, other masters 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 enabled and can grant the bus to other bus masters. When this bit is '1' the system arbiter is disabled, and the default CPU owns system bus at all times. |

**Register 13h Fixed Features Control(FIX_CTL)**

Default Value: 20h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **C3 State SLP#(GPIO3) Enable**<br>In Pentium II platform, the SLP# can be asserted upon the system enters the CPU C3 state if this bit is enabled.<br>1:Enable<br>0:Disable |
| 6 | R/W | **Reserved(should be programmed with 0)** |
| 5 | R/W | **Power Button Over-ride Enable**<br>1:Enable<br>0:Disable |
| 4 | R/W | **Power Button Status Generation Selection**<br>0: PWRBTN_STS will be set until the Power Button is released.<br>1: PWRBTN_STS will be set immediately when the Power Button is pressed. |
| 3 | R/W | **Disable CPU Clock State--ISA BUS Floating Enable**<br>This bit is set to enable the floating of ISA bus during the S2 sleeping state such that ISA power may be turn off.<br>0: Disable<br>1: Enable |
| 2 | R/W | **Disable CPU Clock State--Clock Generator Stop Enable**<br>This bit is set to stop clock generator in S2 sleeping state. When this function is enabled, GPIO3 should be programmed to output mode (20h bit 3=0) as well as activate CPU_STOP#(28h bit 3=1) at the same time.<br>0:Disable<br>1:Enable |
| 1 | WO | **BIOS Relationship (BIOS_RLS)**<br>BIOS can set GBL_STS (Register 00 bit 5)=1 by setting this bit. If SCI_EN and GBL_EN are both set to 1, a SCI will be generated. This bit is write only. |

| | | |
|---|---|---|
| 0 | R/W | **IRQ0 Enable**<br>When the system is in C2 or C3 state with the STPCLK# asserted, an IRQ0 Event can be enabled to de-assert STPCLK # for 125 ms; during which period. The PMU handler can be invoked to update system timer.<br>0:Disable<br>1:Enable |

**Register 14h GPE Status(GPE_STS)**
Default Value: 00h
Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **GPIO15 Status(GPIO15_STS)**<br>Refer to the Note. |
| 30 | R/W | **GPIO14 Status(GPIO14_STS)**<br>Refer to the Note. |
| 29 | R/W | **GPIO13 Status(GPIO13_STS)**<br>Refer to the Note. |
| 28 | R/W | **GPIO12 Status(GPIO12_STS)**<br>Refer to the Note. |
| 27 | R/W | **GPIO11 Status(GPIO11_STS)**<br>Refer to the Note. |
| 26 | R/W | **GPIO3 Status(GPIO3_STS)**<br>Refer to the Note. |
| 25 | R/W | **GPIO9 Status(GPIO9_STS)**<br>Refer to the Note. |
| 24 | R/W | **GPIO8 Status(GPIO8_STS)**<br>Refer to the Note. |
| 23 | R/W | **GPIO2 Status(GPIO2_STS)**<br>Refer to the Note. |
| 22:18 | RO | **Reserved** |
| 17 | R/W | **GPIO1 Status(GPIO1_STS)**<br>Refer to the Note. |
| 16 | R/W | **GPIO0 Status(GPIO0_STS)**<br>Refer to the Note. |
| 15 | R/W | **GPIO17 Status(GPIO17_STS)**<br>Refer to the Note. |
| 14 | R/W | **GPIO16 Status(GPIO16_STS)**<br>Refer to the Note. |
| 13 | R/W | **GPIO7 Status(GPIO7_STS)**<br>Refer to the Note. |

| | | |
|---|---|---|
| 12 | R/W | **PMU Status(PMU_STS)**<br>This bit is set when a event enabled by PMU register 80h ~ 83h is active. |
| 11 | R/W | **SMBus Status(SMB_STS)**<br>This bit is set when a SMBus Controller interrupt is active. While both SMB_STS and SMB_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |
| 10 | R/W | **General Purpose Timer Status(GPTMR_STS)**<br>This bit is set when the General purpose timer times out. While both GPTIMER_STS and GPTIMER_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |
| 9 | R/W | **SERIAL IRQ Status(SIRQ_STS)**<br>This bit is set to 1 when a serial IRQ SMI# event is generated. While both SIRQ_STS and SIRQ_EN are set to 1, a power management event is raised (SMI#). This bit can only be cleared by writing a '1' to this bit.<br>**Note**: SIRQ_STS can only assert SMI#. |
| 8 | R/W | **USB Status(USB_STS)**<br>This bit is set when an USB SMI#/IRQ is generated. While both USB_STS and USB_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |
| 7 | R/W | **Wakeup IRQ Status(IRQWK_STS)**<br>This bit is set when a PMU Wakeup 0 event (as defined in PMU Register 50~53h) is generated. When both IRQWK_STS and IRQWK_EN are set, system can wake up from S1 or S2 mode. This bit can only be cleared by writing a '1' to this bit.<br>**Note**: The IRQWK_STS and IRQWK_EN will not generate SMI# or SCI in working state. |
| 6 | R/W | **Hot Key Status(HOTKEY_STS)**<br>This bit is set when a HOTKEY event is generated. While both HOTKEY_STS and HOTKEY_EN are set to 1, a power management event is raised (SMI#, SCI). This bit can only be cleared by writing a '1' to this bit.<br>**Note**: Hotkey can't wake-up system from any of the Sx state! |
| 5 | R/W | **GPIO4 Status(GPIO4_STS)**<br>Refer to the description of GPIOX Status. |
| 4 | R/W | **Data Acquisition Module Status(DAM_STS)**<br>This bit is set when a DAM SMI#/IRQ is generated. While both DAM_STS and DAM_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |

| | | |
|---|---|---|
| 3 | R/W | **GPIO5 Status(GPIO5_STS)**<br>This bit is set when GPIO10 event is generated or set by a specific programming sequence to ACPI register 38h and 39h. |
| 2 | R/W | **GPIO10 Status(GPIO10_STS)**<br>This bit is set when GPIO10 event is generated or set by a specific programming sequence to ACPI register 38h and 39h. |
| 1 | R/W | **Ring Status(RI_STS)**<br>This bit is set when a MODEM ring event is generated or set by a specific programming sequence to ACPI register 38h and 39h. While both RI_STS and RI_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |
| 0 | R/W | **EXTSMI# Status(EXT_STS)**<br>This bit is set to 1 when an EXTSMI# event is generated. While both EXT_STS and EXT_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). This bit can only be cleared by writing a '1' to this bit. |

**Note: GPIOx Status (GPIOx_STS)**

The GPIOx_STS is set when any one of GPIOx event is generated and the corresponding GPIOx is programmed to be input mode. While both GPIOx_STS and GPIOx_EN are set to 1, a power management event is raised (SMI#, SCI or WAKE). The GPIOx_STS can only be cleared by writing a '1' to the corresponding bit position.

**Register 18h GPE Enable(GPE_EN)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **GPIO15 Enable(GPIO15_EN)**<br>The Enable bit of GPIO15, when GPIO15_STS and GPIO15_EN are set, a power management event is raised. |
| 30 | R/W | **GPIO14 Enable (GPIO14_EN)**<br>The Enable bit of GPIO14, when GPIO14_STS and GPIO14_EN are set, a power management event is raised. |
| 29 | R/W | **GPIO13 Enable (GPIO13_EN)**<br>The Enable bit of GPIO13, when GPIO13_STS and GPIO13_EN are set, a power management event is raised. |
| 28 | R/W | **GPIO12 Enable (GPIO12_EN)**<br>The Enable bit of GPIO12, when GPIO12_STS and GPIO12_EN are set, a power management event is raised. |
| 27 | R/W | **GPIO11 Enable (GPIO11_EN)**<br>The Enable bit of GPIO11, when GPIO11_STS and GPIO11_EN are set, a power management event is raised. |

| | | |
|---|---|---|
| 26 | R/W | **GPIO3 Enable (GPIO3_EN)**<br>The Enable bit of GPIO3, when GPIO3_STS and GPIO3_EN are set, a power management event is raised. |
| 25 | R/W | **GPIO9 Enable (GPIO9_EN)**<br>The Enable bit of GPIO9, when GPIO9_STS and GPIO9_EN are set, a power management event is raised. |
| 24 | R/W | **GPIO8 Enable (GPIO8_EN)**<br>The Enable bit of GPIO8, when GPIO8_STS and GPIO8_EN are set, a power management event is raised. |
| 23 | R/W | **GPIO2 Enable (GPIO2_EN)**<br>The Enable bit of GPIO2, when GPIO2_STS and GPIO2_EN are set, a power management event is raised. |
| 22:18 | R/W | **Reserved** |
| 17 | R/W | **GPIO1 Enable (GPIO1_EN)**<br>The Enable bit of GPIO1, when GPIO1_STS and GPIO1_EN are set, a power management event is raised. |
| 16 | R/W | **GPIO0 Enable (GPIO0_EN)**<br>The Enable bit of GPIO0, when GPIO0_STS and GPIO0_EN are set, a power management event is raised. |
| 15 | R/W | **GPIO17 Enable (GPIO17_EN)**<br>The Enable bit of GPIO17, when GPIO17_STS and GPIO17_EN are set, a power management event is raised. |
| 14 | R/W | **GPIO16 Enable (GPIO16_EN)**<br>The Enable bit of GPIO16, when GPIO16_STS and GPIO16_EN are set, a power management event is raised. |
| 13 | R/W | **GPIO7 Enable (GPIO7_EN)**<br>The Enable bit of GPIO7, when GPIO7_STS and GPIO7_EN are set, a power management event is raised. |
| 12 | R/W | **PMU Enable (PMU_EN)**<br>When a acitve power management event enabled by PMU register 80h~83h and this bit is set, a power management event will be raised. |
| 11 | R/W | **SMBus Enable (SMB_EN)**<br>When SMB_EN and SMB_STS are set, a power management event is raised. |
| 10 | R/W | **General Purpose Timer Enable (GPTMR_EN)**<br>When GPTIMER_STS and GPTIMER_EN are set, a power management event is raised. |
| 9 | R/W | **Serial IRQ SMI# Enable Enable (SIRQ_EN)**<br>When SIRQ_STS and SIRQ_EN are set, a **SMI#** event is raised. |
| 8 | R/W | **USB Enable (USB_EN)**<br>When USB_EN and USB_STS are set, a power management event is raised. |

| | | |
|---|---|---|
| 7 | R/W | **Wakeup IRQ Enable (IRQWK_EN)**<br>When IRQWK_EN and IRQWK_STS are set to 1 during sleeping state, WAK_STS will be set. |
| 6 | R/W | **Hotkey Enable (HOTKEY_EN)**<br>When HOTKEY_STS and HOTKEY_EN are set, a power management event is raised. |
| 5 | R/W | **GPIO4 Enable (GPIO4_EN)**<br>The Enable bit of GPIO4, when GPIO4_STS and GPIO4_EN are set, a power management event is raised. |
| 4 | R/W | **Data Acquisition Module Enable (DAM_EN)**<br>When DAM_EN and DAM_STS are set, a power management event is raised. |
| 3 | R/W | **GPIO10 Enable (GPIO10_EN)**<br>The Enable bit of GPIO10, when GPIO10_STS and GPIO10_EN are set, a power management event is raised. |
| 2 | R/W | **GPIO5 Enable (GPIO5_EN)**<br>The Enable bit of GPIO5, when GPIO5_STS and GPIO5_EN are set, a power management event is raised. |
| 1 | R/W | **Ring Enable (RI_EN)**<br>When RI_EN and RI_STS are set, a power management event is raised. |
| 0 | R/W | **EXTSMI# Enable (EXT_EN)**<br>When EXT_STS and EXT_EN are set, a power management event is raised. |

**Register 1Ch GPE Pin Status(GPE_Pin)**

Default Value: 000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 23:18 | RO | **Reserved** |
| 17:16, 11:7, 5:0 | R/W | **GPIO[17:16,11:7,5:0] Pin I/O Register**<br>When the corresponding GPIO[17:16,11:7,5:0] are programmed to input mode, their input value can be read from this register. When GPIO[17:16,11:7,5:0] are programmed to output mode, their output value can be written to this register. |
| 15:12 | RO | **GPI[15:12] Pin Register**<br>The corresponding GPI [15:12] input value can be read from this register. |
| 6 | R/W | **GPO6 Pin Register**<br>The GPO[6] output value can be written to this register. |

**Register 1Fh General Purpose Timer(GP_TMR)**

Default Value: FFh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **General Purpose Timer**<br>The General Purpose timer is a down-counting counter with time resolution of 1 µsec or 1 min (programmable via Register 2A bit 8). While a value is written to this timer, it begins to count. It raises a power management event when the counter reaches zero. The counter can be used as a suspend timer when Register 2A bit 7 is set to 1 and SCI_EN is 0. |

**Register 20h-23h GPE I/O Selection(GPE_IO)**

Default Value: 0003 0B9Fh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:18 | RO | **Reserved** |
| 17:16 | R/W | **GPIO[17:16] INPUT/OUTPUT Control**<br>1 : Input Mode<br>0 : Output Mode |
| 15:12 | RO | **Reserved**<br>(GPI15~GPI12 are input only.) |
| 11 | R/W | **GPIO11 INPUT/OUTPUT Control**<br>1 : Input Mode<br>0 : Output Mode |
| 10 | R/W | **Reserved**<br>(GPIO10 INPUT/OUTPUT Control is in APC Register 04 bit 3) |
| 9:7 | R/W | **GPIO[9:7] INPUT/OUTPUT Control**<br>1 : Input Mode<br>0 : Output Mode |
| 6 | RO | **Reserved**<br>(GPO6 is output only.) |
| 5 | RO | **Reserved**<br>(GPIO5 INPUT/OUTPUT Control is in APC Register 04 bit 4.) |
| 4:0 | R/W | **GPIO[4:0] INPUT/OUTPUT Control**<br>1 : Input Mode<br>0 : Output Mode |

**Register 24h-27h GPE Polarity Selection(GPE_Pol)**

Default Value: 0000 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:29 | RO | **Reserved** |
| 28 | R/W | **EXTSMI# Polarity**<br>0 : Low active<br>1 : High active |
| 27:18 | RO | **Reserved** |
| 17:6,<br>11:7,<br>5:0 | R/W | **GPIO[17:16,11:7, 5:0] Polarity in Input Mode**<br>0 : Low active<br>1 : High active<br>Note: This bit has no effect in toggle input mode. |
| 15:12 | R/W | **GPI[15:12] Polarity**<br>0 : Low active<br>1 : High active<br>Note: This bit has no effect in toggle input mode. |

**Register 28h GPE Multi-definition Pins Selection (GPE_Mul)**

Default Value: 1001h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:13 | RO | **Reserved** |
| 12 | R/W | **GPIO16/IOCHK# Pin Selection when GPIO16 at input mode**<br>0 : GPIO16<br>1 : IOCHK# |
| 11:10 | RO | **Reserved** |
| 9 | R/W | **GPIO9/THERM#(BTI)/SMBALERT# Pin Selection when GPIO9 at input mode**<br>0 : GPIO9<br>1 : THERM#(Thermal detect)/BTI(used by DAM)/SMBALERT# |
| 8 | R/W | **GPIO8/OC1# Pin Selection when GPIO8 at Input Mode**<br>0 : GPIO8<br>1 : OC1#(Used by USB) |
| 7 | R/W | **GPIO7/ Pin Selection when GPIO7 at Output Mode**<br>0 : GPIO7<br>1 : PPS(Port Power Status, Used by USB) |

| | | |
|---|---|---|
| 6 | R/W | **GPIO7/OC0# Pin Selection when GPIO7 at Input Mode**<br>0 : GPIO7<br>1 : OC0#/PPS(Used by USB) |
| 5 | R/W | **GPIO17/SIRQ Pin Selection**<br>0: GPIO17<br>1: SIRQ I/O |
| 4 | R/W | **Reserved**<br>This bit must be set to '0'. |
| 3 | R/W | **GPIO3/CPU_STOP#/SLP# Pin Selection**<br>0 : GPIO3<br>1 : CPU_STOP# / SLP# |
| 2:1 | RO | **Reserved**<br>Note: GPIO1, GPIO2 are multiplexed with PMCLK and KBCLK, the selection bit is located in PCI-to-ISA Configuration Registers 70h bit 3. When that bit is set to 1, the internal KB controller will be enabled and GPIO1, GPIO2 are both disabled. |
| 0 | R/W | **GPIO0/PAR Pin Selection when GPIO0 at Output Mode**<br>0 : GPIO0<br>1 : PAR |

**Register 2Ah GPE Control(GPE_CTL)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:11 | R/W | **GPIO[15:11] Input Trigger Mode**<br>0 : Toggle trigger<br>1 : Edge trigger<br>Toggle means input changes from either 0 to 1 or 1 to 0.<br>Edge polarity can be defined at Register 24h bits [15:11]. |
| 10 | R/W | **Reload DOZE and Stop SUSPEND Timer Enable**<br>When this bit is enable, monitored events from PMU as defined in Register 40~43h will reload the DOZE timer and the SUSPEND timer.<br>0: Disable<br>1: Enable |
| 9 | R/W | **Power Management Timer Function Selection**<br>0 : ACPI PM timer<br>1 : DOZE timer |
| 8 | R/W | **General Purpose Timer Resolution**<br>0 : 1 us<br>1 : 1 min |

| | | |
|---|---|---|
| 7 | R/W | **General Purpose Timer Function Selection**<br>0 : BIOS timer<br>1 : Suspend timer |
| 6 | R/W | **Thermal Event Throttling CPU STPCLK# Enable**<br>0 : Disable<br>1 : Enable<br>When set to 1, an active thermal event will always activate the throttling function to assert STPCLK# regardless of the state of the Throttling Enable bit (Register 0Ch bit 4). This function is enabled when GPIO9 is configured as THERM# input (28h.bit9=1). |
| 5 | R/W | **Ring In Detection Method**<br>0 : Ring input is asserted for more than150ms<br>1 : Ring input is between 14Hz and 70 Hz |
| 4:2 | R/W | **Thermal Throttling Duty Cycle Control**<br>This 3-bit field determines the duty cycle of the STPCLK# signal when Thermal throttling event (THERM#) is asserted.<br>**Bits** — **Performance Rate**<br>000 — 100%<br>001 — 12.5%<br>010 — 25%<br>011 — 37.5%<br>100 — 50%<br>101 — 62.5%<br>110 — 75%<br>111 — 87.5% |
| 1 | R/W | **GPIO1 Input Trigger Mode**<br>0 : Toggle trigger<br>1 : Edge trigger<br>Toggle means input changes from either 0 to 1 or 1 to 0.<br>Edge polarity can be defined at Register 24h bit 1. |
| 0 | R/W | **Reserved.**<br>This bit must be set to 0. |

**Register 2Ch-2Dh GPIO Events SMI# Enable(GPE_SMI)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:12 | R/W | **GPI[15:12] Events SMI# Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPI[15:12] events become SMI# source. |
| 11:7 | R/W | **GPIO[11:7] Events SMI# Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO[11:7] events become SMI# source. |
| 6 | R/W | **GPIO17 Event SMI# Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO17 event become SMI# source. |
| 5:1 | R/W | **GPIO[5:1] Events SMI# Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO[5:1] events become SMI# source. |
| 0 | R/W | **GPIO16 Event SMI# Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO16 event become SMI# source. |

**Register 2Eh-2Fh GPIO Events StopGPTMR and Reload PMU Timers Enable(GPE_RL)**

Default Value: 0000 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:12 | R/W | **GPI[15:12] Events Stop GPTMR and Reload PMU Timers Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPI[15:12] events become stop GPTMR events source and the System Standby Timer of PMU will be reloaded. |

| | | |
|---|---|---|
| 11:7 | R/W | **GPIO[11:7] Events Stop GPTMR and Reload PMU Timers Enable**<br>0:Disable<br>1:Enable<br>When enable, regardless SCI_EN bit, the GPIO[11:7] events become stop GPTMR events source and the System Standby Timer of PMU will be reloaded. |
| 6 | R/W | **GPIO17 Event Stop GPTMR and Reload PMU Timers Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO17 event become Stop GPTMR event source and the System Standby Timer of PMU will be reloaded. |
| 5:1 | R/W | **GPIO[5:1] Events Stop GPTMR and Reload PMU Timers Enable**<br>0:Disable<br>1:Enable<br>When enabled, regardless SCI_EN bit, the GPIO[5:1] events become Stop GPTMR events source and the System Standby Timer of PMU will be reloaded. |
| 0 | R/W | **GPIO16 Event Stop GPTMR and Reload PMU Timers Enable**<br>0:Disable<br>1:Enable<br>When enable, regardless SCI_EN bit, the GPIO16 event become Stop GPTMR events source and the System Standby Timer of PMU will be reloaded. |

**Register 30h Legacy Status(LEG_STS)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Data Acquisition Module SMI# Status(DAMSMI_STS)**<br>This bit is set when a Data Acquisition Module SMI# is generated and must be cleared before exiting SMI# handler routine. This bit can be cleared by writing a '1' to it |
| 6 | R/W | **USB SMI# Status(USBSMI_STS)**<br>This bit is set when an USB SMI# is generated and must be cleared before exiting USB SMI# handler routine. This bit can be cleared by writing a '1' to it. |
| 5 | R/W | **SMI# Status(SMI_STS)**<br>This bit is set when an SMI# is generated and must be cleared before exiting SMI# handler routine. This bit can be cleared by writing a '1'. |

| | | |
|---|---|---|
| 4 | R/W | **SMI# Command Status (SMICMD_STS)**<br>This bit is set when an I/O write cycle to SMI# command port is detected. While SMICMDDIS_STS and SMICMD_DIS are set to 1, a SMI# is raised. |
| 3 | R/W | **SMBus SMI# Status (SMBSMI_STS)**<br>This bit is set when SMBus Controller Interrupt is active. While SMBSMI_STS and SMBSMI_EN are set to 1, a SMI# is raised. |
| 2 | R/W | **Periodic SMI# Status (PERSMI_STS)**<br>When periodic SMI# is enabled, this bit will be set every 16 sec. |
| 1 | R/W | **System wake-up SMI# Status (LEGA_STS)**<br>This bit is set when system is Waked-up from S1 or S2. When both LEGA_STS and LEGA_EN are set, a SMI# is raised. It can only be cleared by writing a 1 to this bit position. |
| 0 | R/W | **Software SMI# Status(BIOS_STS)**<br>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 writing a 1 to this bit position. This bit corresponds to the writing GBL_RLS bit of PM1 Control Register. |

**Register 31h Legacy Enable(LEG_EN)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Data Acquisition Module SMI# Enable(DAMSMI_EN)**<br>When enable, a DAM SMI# can be generated.<br>0 : Disable<br>1 : Enable |
| 6 | R/W | **USB SMI# Enable(USBSMI_EN)**<br>When enable, an USB SMI# can be generated.<br>0 : Disable<br>1 : Enable |
| 5 | R/W | **SMI# Enable(SMI_EN)**<br>The system SMI# can only be generated when this bit is enabled.<br>0 : Disable<br>1 : Enable |
| 4 | R/W | **SMI# Command (SMICMD_EN)**<br>SMI# command disable bit. While SMICMD_STS and SMICMD_EN are set to 1, a SMI# is raised.<br>0 : Disable<br>1 : Enable |

| | | |
|---|---|---|
| 3 | R/W | **SMBus SMI# Enable (SMBSMI_EN)**<br>SMBus SMI# enable bit. While SMBSMI_STS and SMBSMI_EN are set to 1, a SMI# is raised.<br>0 : Disable<br>1 : Enable |
| 2 | R/W | **Period SMI# Enable (PERSMI_EN)**<br>If this bit is set to 1, a SMI# will be generated every 16 sec.<br>0 : Disable<br>1 : Enable |
| 1 | R/W | **System wake-up SMI# Enable (LEGA_EN)**<br>Legacy S1, S2 wake-up SMI# enable bit.<br>0 : Disable<br>1 : Enable |
| 0 | R/W | **Software SMI# Enable(BIOS_EN)**<br>BIOS_STS enable bit.<br>0 : Disable<br>1 : Enable |

**Register 32h Reserved**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Reserved**<br>Read as 0. |

**Register 33h Test Control(TST_CTL)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:1 | RO | **Reserved** |
| 0 | R/W | **ACPI Test Mode ( for internal use only)**<br>0 : Normal Mode<br>1 : Test Mode |

**Register 34h Reserved**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:1 | R/W | **Reserved (Should be programmed with 0).** |

| | | |
|---|---|---|
| 0 | R/W | **SCI Generation Mask Control**<br>When this bit is enabled and system is waken up from S1/S2, no SCI# will be raised except the enabled Global or Power Management Timer event.<br>0 : Disable (recommended)<br>1: Enable |

**Register 35h SMI# Command Port(SMI_CMD)**

Default Value: None

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMI# Command value.** |

**Register 36h Mail Box(Free_Byte)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **R/W Free bits** |

**Register 37h Reversed**

**Register 38h SMBus IO Index Register(SMB_INDEX)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus IO Address Port** |

**Register 39h SMBus IO Data Register(SMB_DATA)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus IO Data Port** |

## 7.4 SMBUS IO REGISTERS

The SMBus Register can be access by first writing a INDEX ADDRESS to ACPI I/O base 38h, then read/write DATA via ACPI I/O base 39h. All SMBus registers can only be accessed by byte.

**Register 00h~01h SMBus Status(SMB_STS)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:11 | RO | **Reserved.** |
| 10 | R/W | **SMBus Alert (SMB_ALRT)**<br>This bit is set when the SMBALERT# is active. It can be cleared by writing a '1' to this bit. |
| 9 | R/W | **HIT Host Alias Address (HIT_HAA)**<br>This bit is set when the Host received a modified Write Word from a device master with the address field equivalent to the content in Host Alias Address register. This bit can be reset by writing a '1' to this bit position. If this bit is set to 1, no more modified Write Word can be received by the Host until this bit is clear to 0. |
| 8 | R/W | **HIT Host Reserved Address (HIT_HRA)**<br>This bit is set when the Host received a modified Write Word from a device master and the address field is 08h and can only be reset by writing a '1' to this bit position. If this bit is set to 1, no more modified Write Word transaction can be received by the Host until this bit is clear to 0. |
| 7 | R/W | **Sub-Block Request(SBK_REQ)**<br>This bit is set when the SMBus Host has finished 8 bytes data transfer for Block Protocol and there are some data have not transferred yet. If the byte count of the Block protocol is 32, then total 4 Interrupt request will occur during the entire block transfer. For the first three Interrupt, this bit will be set and the service routine should process or prepare the data as soon as possible**,** or the total trasnfer time may violate SMBus SPEC 1.0 (Timeout < 10ms). Once this bit is cleared, the following transation will be started right away. For the last interrupt, the Host Master bit will be set instead.<br>This bit is set by hardware, and can only be reset by writing a '1' to this bit position. |
| 6 | R/W | **Host Master (HO_MAS)**<br>This bit is set when the SMBus Host Master tranfer is complete, and can only be reset by writing a '1' to this bit position. |
| 5 | R/W | **SMBus Collision (BUS_COL)**<br>This bit is set when a SMBus collision condition occur during the Host Master transfer cycle and the Host lose in the bus arbitration, and can only be reset by writing a '1' to this bit position. The software should clear this bit and re-start SMBus operation. |

| | | |
|---|---|---|
| 4 | R/W | **Device Error(DEV_ERR)**<br>This bit is set when a Device Error condition occur, and can only be reset by writing a '1' to this bit position. The Device Errors may cause by:<br>• No Acknowledge received by Host from Device<br>• Timeout when Host transmit message<br>• Timeout when Host receive message |
| 3 | RO | **Reserved.** |
| 2 | RO | **SMBus Status (SMBUS_STS)**<br>Indicate the SMBus is at idle or active state**.**<br>1: Active<br>0: Idle |
| 1 | RO | **Host Master Status(HM_STS)**<br>Indicate that the Host Master is at idle or active state**.** When Host Master is at IDLE state, the Host Master is available for software to control.<br>1: Active<br>0: Idle |
| 0 | RO | **SMBus Interrupt Status(SMB_INTR).**<br>A '1' incicates a SMBus interrupt is generated by any of above Interrupt source. |

**Register 02h~03h SMBus Host Control(SMB_CTL)**

A SMBus Interrupt can be generated if Register 02h, bit 0 is enabled and the Interrupt Status bit with associated enable bits are set to 1.

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:13 | RO | **Reserved.** |
| 12 | R/W | **SMBus Host Slave Timeout Enable(SMBHSTO_EN)**<br>When this bit is enabled and the time for Host as a slave to receive message is over specification, a SMBus Interrupt will be generated.<br>0:Disable<br>1:Enable |
| 11 | R/W | **SMBus Host Master Timeout Enable(SMBHMTO_EN)**<br>When this bit is enabled and the time for Host as a Master to transfer mesage is over specification, a SMBus Interrupt will be generated.<br>0:Disable<br>1:Enable |

| | | |
|---|---|---|
| 10 | R/W | **SMBus Alert Interrupt Enable (SMBAL_INTR_EN)**<br>When this bit is enabled, a SMBus Interrupt will be generated by the active SMBALERT#.<br>0:Disable<br>1:Enable |
| 9 | R/W | **SMBus Host Alias Address Interrupt Enable(SMBHAA_INTR_EN)**<br>When this bit is enabled and the Device Address field in the modified Write Word received by Host hit the Host Alias Address, a SMBus Interrupt will be generated.<br>0:Disable<br>1:Enable |
| 8 | R/W | **SMBus Host Reserved Address Interrupt Enable(SMBHRA_INTR_EN)**<br>When this bit is enabled and the Device Address field in the modified Write Word received by Host is 10h, a SMBus Interrupt will be generated.<br>0:Disable<br>1:Enable |
| 7 | R/W | **SMBus Sub-Block Interrupt Enable (SMBB_INTR_EN)**<br>When this bit is enabled and the Host Master has finished 8 bytes transfer for Block Protocol and there are some data have not transferred yet, a SMBus Interrupt will then be generated.<br>0:Disable<br>1:Enable |
| 6 | R/W | **SMBus Host Transmitter Clock Selection(SMBHT_CLK)**<br>0:28KHz<br>1:56KHz |
| 5 | WO | **Kill(SMB_Kill)**<br>When this bit is set, all SMBus operation controlled by Host will be stopped and cleared to initial state. This operation won't affect the values in Command, Data, and Address registers. |
| 4 | WO | **Start(SMB_START)**<br>The SMBus Host transfer is initiated by writing a 1 to this bit. Prior to setting this bit, the SMBus Command Protocaol bits (SMB_PTL) and the associated registers should be properly programmed. |

| | | |
|---|---|---|
| 3:1 | R/W | **SMBus Command Protocol(SMB_PTL)**<br>Selecting the Protocol that SMBus Host is going to execute. Read or Write transfer is determined by SMBus Address register bit 0.<br>**Bit[3:1]** **Protocol**<br>000 Quick command<br>001 Send/Receive Byte<br>010 Read/Write Byte Data<br>011 Read/Write Word Data<br>100 Process Call<br>101 Read/Write Block Data<br>110 Reserved<br>111 Reserved |
| 0 | R/W | **SMBus Interrupt Enable(SMB_INTR_EN).**<br>This bit is used to enable the SMBus interrupt generation.<br>0:Disable<br>1:Enable |

**Register 04h SMBus Address(SMB_ADDR)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:1 | R/W | **SMBus Address(SMB_ADDRESS)**<br>The field is the slave address to target device. |
| 0 | R/W | **SMBus Read/Write(SMB_RW)**<br>Define write or read protocol for all Host transfer except Process Call.<br>0:Write<br>1:Read |

**Register 05h SMBus Command(SMB_CMD)**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus Command(SMB_Command)**<br>This register contains the Command code which is sent to device. |

**Register 06h SMBus Processed Byte Count(SMB_PCNT)**

Default Value: 01h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | RO | **Reserved** |
| 4:0 | RO | **SMBus Block Byte Pointer(SMB_BBP)**<br>These bits provide imformation that how many bytes data have transferred for Block procotol interrupt service routine. A 'zero' indicates a maximun of 32 data bytes has transferred. |

**Register 07h SMBus Byte Count(SMB_CNT)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | RO | **Reserved** |
| 5:0 | R/W | **SMBus Byte Count(SMB_CNT)**<br>These bits contain the byte count of Block Read/Write protocol. The byte count can not be 0. |

**Register 08h ~ 0Fh SMBus Byte 0~7 (SMB_BYTE0~7)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus Byte[7:0](SMB_Byte[7:0])**<br>These eight bytes are the data byte field for Block Read/Write protocol. The Byte0 is also used as byte data for Byte protocol which include Send/Receive Byte, Read/Write Data Byte. In addition, the Byte0 (low byte) and Byte1(high byte) are combined as word data for Word protocol which include Read/Write Word, Process Call. |

**Register 10h SMBus Device Address(SMB_DEV)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus Device Address(SMB_Device)**<br>This field is used to save the Device Address when Host received a modified Write Word from other SMBus trasmitter and the Slave Address field hit the Host Reserved Address or Host Alias Address. |

**Register 11h SMBus Device Byte 0(SMB_DB0)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus Device Byte 0(SMB_Byte0)**<br>This register contains the Data Low Byte when Host receive a modified Write Word from other SMBus master. |

**Register 12h SMBus Device Byte 1(SMB_DB1)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **SMBus Device Byte 1(SMB_Byte1)**<br>This register contains the Data High Byte when Host receives a modified Write Word from other SMBus master. |

**Register 13h SMBus Host Alias Address(SMB_HAA)**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:1 | R/W | **SMBus Host Alias Address(SMB_Alias)**<br>When Host receives a modified Write Word and the Slave Address field is equivalent to the content of this register, an interrupt will be raised if Alias Interrupt is also enabled. |
| 0 | RO | Read as '0'. The Host Slave accepts master Write Word protocol only. |

**Register FFh Set Specific Status bit in GPE_STS and PM1_STS**

Default Value: 00h

Access: Write Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | RO | **Reserved** |
| 5 | WO | **Set GPIO5_STS of GPE_STS**<br>When a '1' is written to this bit, it will set the GPIO5_STS of PM1_STS. A '0' written to this bit has no effect. |
| 4 | WO | **Set GPIO10_STS of GPE_STS**<br>When a '1' is written to this bit, it will set the GPIO10_STS of PM1_STS. A '0' written to this bit has no effect. |
| 3 | WO | **Set RI_STS in GPE_STS**<br>When a '1' is written to this bit, it will set the RI_STS of PM1_STS. A '0' written to this bit has no effect. |
| 2 | WO | **Set WAK_STS of PM1_STS**<br>When a '1' is written to this bit, it will set the WAK_STS of PM1_STS. A '0' written to this bit has no effect. |
| 1 | WO | **Set RTC_STS of PM1_STS**<br>When a '1' is written to this bit, it will set the RTC_STS of PM1_STS. A '0' written to this bit has no effect. |
| 0 | WO | **Set PWRBTN_STS of PM1_STS**<br>When a '1' is written to this bit, it will set the PWRBTN_STS of PM1_STS. A '0' written to this bit has no effect. |

### 7.5 THE DATA ACQUISITION MODULE INTERNAL REGISTERS

**Base Address + 05h Index Address Pointer Register**

Default Value: 00h

Access: Read/Write, Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **Busy**<br>The bit is set when this register is being written and is reset when the data register is being written or read. |
| 6:0 | R/W | **Index Address Pointer**<br>Address pointer aims to the internal registers. |

**Base Address + 06h Data Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **8-bit data written to or Read from the internal register is pointed by the Index Address Pointer Register**<br>To communicate with the internal registers of Data Acquisition Module, first write the address of that register to the Index Address Pointer register then read or write data from or to that register via the data register. |

**Index Address 40h Configuration Register**

Default Value: 08h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Initialization**<br>Writing a 1 to this bit will restore power-on values to all registers including the bit itself. |
| 6 | R/W | **Reserved** |
| 5 | R/W | **NMI/IRQ select**<br>0 : IRQ<br>1 : NMI |
| 4 | R/W | **Reserved** |
| 3 | R/W | **INT_Clear**<br>Writing a 1 will disable the SMI# and NMI/IRQ outputs without affecting the contents of Interrupt Status Register, and the monitoring process will stop. The monitoring process will resume upon clearing this bit. |
| 2 | R/W | **NMI/IRQ Enable**<br>Writing a 1 will enable the NMI/IRQ interrupt output. |
| 1 | R/W | **SMI# Enable**<br>Writing a 1 will enable the SMI# interrupt output. |
| 0 | R/W | **Start**<br>Writing a 1 to this bit will start the round-robin monitoring, while writing a 0 will disable it. The outputs of the interrupt pins will not be cleared if the user writes a 0 to this bit after an interrupt has occurred an unlike ”INT_Clear” bit. |

**Index Address 41h Interrupt Status Register I**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **FAN2**<br>A 1 indicates the fan count limit has been exceeded. |
| 6 | RO | **FAN1**<br>A 1 indicates the fan count limit has been exceeded. |
| 5 | RO | **BTI**<br>A 1 indicates an interrupt has occurred from the BTI input. |
| 4 | RO | **Reserved** |
| 3 | RO | **VIN3**<br>A 1 indicates a High or Low limit has been exceeded. |
| 2 | RO | **VIN2**<br>A 1 indicates a High or Low limit has been exceeded. |
| 1 | RO | **VIN1**<br>A 1 indicates a High or Low limit has been exceeded. |
| 0 | RO | **VIN0**<br>A 1 indicates a High or Low limit has been exceeded. |

**Index Address 42h** **Interrupt Status Register II**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **DXP/VIN4**<br>A 1 indicates a High or Low limit has been exceeded. |
| 6:0 | RO | **Reserved** |

**Index Address 43h** **SMI# Mask Register I**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **FAN2** |
| 6 | R/W | **FAN1** |
| 5 | R/W | **BTI** |
| 4 | R/W | **Reserved** |
| 3 | R/W | **VIN3** |
| 2 | R/W | **VIN2** |
| 1 | R/W | **VIN1** |

| | | |
|---|---|---|
| 0 | R/W | **VIN0**<br>0 : Enabled<br>1 : Disabled |

**Index Address 44h SMI# Mask Register II**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **DXP/VIN4**<br>0 : Enabled<br>1 : Disabled |
| 6:0 | R/W | **Reserved** |

**Index Address 45h NMI Mask Register I**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **FAN2** |
| 6 | R/W | **FAN1** |
| 5 | R/W | **BTI** |
| 4 | R/W | **Reserved** |
| 3 | R/W | **VIN3** |
| 2 | R/W | **VIN2** |
| 1 | R/W | **VIN1** |
| 0 | R/W | **VIN0**<br>0 : Enabled<br>1 : Disabled |

**Index Address 46h NMI Mask Register II**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **DXP/VIN4**<br>0 : Enabled<br>1 : Disabled |
| 6:0 | R/W | **Reserved** |

**Index Address 47h Fan Divisor Register**

Default Value: 50h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **FAN2 RPM Control**<br>00 : divide by 1, applied for 8800rpm<br>01 : divide by 2, applied for 4400rpm<br>10 : divide by 4, applied for 2200rpm<br>11 : divide by 8, applied for 1100rpm |
| 5:4 | R/W | **FAN1 RPM Control**<br>00 : divide by 1, applied for 8800rpm<br>01 : divide by 2, applied for 4400rpm<br>10 : divide by 4, applied for 2200rpm<br>11 : divide by 8, applied for 1100rpm |
| 3:0 | R/W | **Reserved** |

**Index Address 48h Reserved**

**Index Address 20h/60h VIN0 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of VIN0** |

**Index Address 21h/61h VIN1 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of VIN1** |

**Index Address 22h/62h VIN2 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of VIN2** |

**Index Address 23h/63h VIN3 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of VIN3** |

**Index Address 24h/64h DXP/VIN4 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of DXP/VIN4** |

**Index Address 25h~27h/65h~67h Reserved. Read as 0.**

**Index Address 28h/68h FAN1 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of FAN1** |

**Index Address 29h/69h FAN2 Reading Register**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **The Reading of FAN2** |

**Index Address 2Ah/6Ah Reserved**

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | Reading of this register can be programmed via PCI-ISA configuration Register78. |

**Index Address 2Bh/6Bh VIN0 High Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of VIN0** |

**Index Address 2Ch/6Ch VIN0 Low Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low Limit of VIN0** |

**Index Address 2Dh/6Dh VIN1 High Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of VIN1** |

**Index Address 2Eh/6Eh VIN1 Low Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low Limit of VIN1** |

**Index Address 2Fh/6Fh VIN2 High Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of VIN2** |

**Index Address 30h/70h VIN2 Low Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of VIN2** |

**Index Address 31h/71h VIN3 High Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of VIN3** |

**Index Address 32h/72h VIN3 Low Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low Limit of VIN3** |

**Index Address 33h/73h DXP/VIN4 High Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The High Limit of DXP/VIN4** |

**Index Address 34h/74h DXP/VIN4 Low Limit Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low Limit of DXP/VIN4** |

**Index Address 35h~3Ah/75h~7Ah Reserved. Read as 0.**

**Index Address 3Bh/7Bh FAN1 Fan Count Limit**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low-Limit Counts based on Internal Clock(22.5KHz) for the Fan Speed** |

**Index Address 3Ch/7Ch FAN2 Fan Count Limit**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **The Low-Limit Counts based on Internal Clock (22.5KHz) for the Fan Speed** |

**Index Address 3Dh~3Fh/7Dh~7Fh Reserved**

## 7.6 APC CONTROL REGISTERS

**APC Register 00~01 h Reserved**

**APC Register 02h**

Default Value: 00h

Access: Read/Write

To enable the "day of the week alarm" function, both the APC_EN(03h.6) and DWAUP_EN(02h.0) bits should be set as well as the RTC Alarm bytes defined. Then to depend on which day(s) the system is intended to be alarmed-up, the following bits [7:0] can be programmed accordingly.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Automatic power up system on Saturday** |
| 6 | R/W | **Automatic power up system on Friday** |
| 5 | R/W | **Automatic power up system on Thursday** |
| 4 | R/W | **Automatic power up system on Wednesday** |
| 3 | R/W | **Automatic power up system on Tuesday** |
| 2 | R/W | **Automatic power up system on Monday** |
| 1 | R/W | **Automatic power up system on Sunday** |
| 0 | R/W | **Day of the Week power up system control (DWAUP_EN)**<br>0: Disable<br>1: Enable |

**APC Register 03h**

Default Value: 80h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **System Powered-up by Power Button Control (BTNUP_EN)**<br>0: Disable<br>1: Enable<br>This bit will be enabled automatically when KPR_LST(07h.0) is set by APC. |
| 6 | R/W | **APC Function Control (APC_EN)**<br>0: Disable<br>1: Enable<br>When set to 1, all power-off events, excluding the Power Button related events, are valid and will be processed. |
| 5 | R/W | **System Powered-up by Ring Control (RNUP_EN)**<br>0 : Disable<br>1 : Enable |

| | | |
|---|---|---|
| 4 | R/W | **RING Input Active Level Control (RN_PAR)**<br>0: Active high<br>1: Active low |
| 3 | R/W | **Pin Function Selection for GPIO5/PME0# (PME0_EN)**<br>0 : Select GPIO5<br>1 : Select PME0#<br>When PME0# is selected, a high to low edge on this pin will cause the PS_ON# to be asserted. |
| 2 | R/W | **System Powered-up by RTC Alarm (ALMUP_EN)**<br>0 : Disable<br>1 : Enable |
| 1:0 | R/W | **APC Test Mode**<br>00 : Normal Mode<br>01 : APC test mode 1<br>10 : APC test mode 2<br>11 : APC test mode 3 |

**APC Register 04h**

Default Value: 38h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Tri-state Control of CKE_S**<br>The CKE_S is in the high impedance state by default, and will be driven low while the system is sequenced into S3 state. Writing a 1 to this bit will put the CKE_S back to high impedance state. Writing a 0 to this bit has no effect. |
| 6 | R/W | **S3 Timer Function Control**<br>0: Disable<br>1: Enable<br>While this bit is set, a 4~60 minutes programmable timer will start to count down upon S3 state is entered. After the timer expires, the system will enter S4/S5 mode automatically. |
| 5 | R/W | **Pin Function Selection for GPIO5/PME0#/DUAL_ON#**<br>0: Select DUAL_ON#<br>1: Select GPIO5/PME0#<br>When DUAL_ON# is selected, this pin is forced to output mode. |
| 4 | R/W | **Input/Output Mode Selection for GPIO5/PME0#/DUAL_ON#**<br>0: Output mode<br>1: Input mode |
| 3 | R/W | **Input/Output Mode Selection for GPIO10/PME1#/ACPILED**<br>0: Output mode<br>1: Input mode |

| | | |
|---|---|---|
| 2 | R/W | **Pin Function Selection for GPO6/CKE_S/ACPILED**<br>0: Select CKE_S/ACPILED<br>1: Select GPO6 |
| 1 | R/W | **Pin Function Selection for GPIO10/PME1#/ACPILED**<br>0: Select GPIO10/PME1#<br>1: Select ACPILED |
| 0 | R/W | **ACPILED 1Hz Blinking Enable**<br>0: Disable<br>1: Enable |

**APC Register 05h**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **DUAL_ON# Active Level Selection**<br>0: Active high<br>1: Active low |
| 6 | R/W | **System Powered-up by Keyboard Password Control (PSUP_EN)**<br>0: Disable<br>1: Enable<br>This bit will be reset to 0 when keyboard controller's power is removed. |
| 5 | R/W | **System Powered-up by Keyboard Hot-Key Control (HKUP_EN)**<br>0: Disable<br>1: Enable<br>This bit will be reset to 0 when keyboard controller's power is removed. |
| 4 | R/W | **Enable the Internal 32KHz-to-8MHz Frequency Conversion Circuitry**<br>0: Enable<br>1: Disable<br>If password security power-up or hot-key power-up function is enabled, this bit must be enabled. |
| 3:0 | R/W | **Reserved.**<br>These bits set RTC test modes, and must be programmed to 0. |

**APC Register 06h**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **System powered-up by Keyboard Hot-Key Status (HKUP_STS)**<br>This bit will be set to '1' while "system powered-up by Keyboard Hot-Key" events occur and can be set to '0' when read. |
| 6 | RO | **System powered-up by Keyboard Password Status (PSUP_STS)**<br>This bit will be set to '1' while "system powered-up by Keyboard Password" events occur and can be set to '0' when read. |
| 5 | RO | **S3 State Status (STR_STS)**<br>This bit will be set to '1' after the system is waked up from S3 state and can be reset when read. |
| 4 | RO | **System Powered-up by PME0# Status (PME0_STS)**<br>This bit will be set to '1' while the "system powered-up by PME0# " event occurs and can be reset when read. |
| 3 | RO | **System Powered-up by PME1# Status (PME1_STS)**<br>This bit will be set to '1' while the "system powered-up by PME1#" event occurs and can be reset when read. |
| 2 | RO | **System Powered-up by RING Status (RNUP_STS)**<br>This bit will be set to '1' while "system powered-up by RING" event occurs and can be reset when read. |
| 1 | RO | **System powered-up by RTC Alarm Status (ALMUP_STS)**<br>This bit will be set to '1' while "system powered-up by RTC Alarm" event occurs and can be set to '0' when read. |
| 0 | RO | **System powered-up by PowerButtonStatus (BTNUP_STS)**<br>This bit will be set to '1' while "system powered-up by PowerButton" events occur and can be set to '0' when read. |

**APC Register 07h**

Default Value: 01h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Pin Function Selection for GPIO10/PME1# (PME1_EN)**<br>0: Select GPIO10<br>1: Select PME1#<br>When PME1# is selected, a low pulse longer than 30 us on this pin will cause the PS_ON# signal to be asserted. |
| 6 | R/W | **Pin Function Selection for CKE_S/ACPILED**<br>0: Select CKE_S<br>1: Select ACPILED |

| | | |
|---|---|---|
| 5 | R/W | **PS_ON# Keep Previous State (PSON_RSM)**<br>0: Disable<br>1: Enable<br>If this bit is set to '1', the PS_ON# will return to its previous state when the AC power is restored from a previous sudden-off for arbitrary period of time. If this bit is set to '0', system is terned on while power is recovered within 4 seconds. On the other hand, system remains off when power is recovered in excess of 4 seconds. |
| 4 | R/W | **Reserved**<br>Reserved and must be set to '0' |
| 3 | R/W | **CMOS Century Byte Set to 20h (2000_EN)**<br>If this bit is set to '1', the data in standard CMOS RAM 32h will be set to 20h while the year byte in RTC time register updates from '99' to '00'. |
| 2:1 | R/W | **Reserved**<br>Reserved and must be set to '0' |
| 0 | R/W | **KBVDD Status (KPR_LST)**<br>0: Power exist<br>1: Power had ever lost<br>This bit will be set to 1 if the power connected to the internal keyboard controller had ever lost. It can be cleared by writing a '0'. |

**APC Register 08h**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **Multiplication of Processor Core Frequency to System Bus Frequency**<br>0000 : 1 / 2 0100 : 2 / 5 1000 : 1 / 6 1100 : 2 / 13<br>0001 : 1 / 3 0101 : 2 / 7 1001 : 1 / 7 1101 : 2 / 15<br>0010 : 1 / 4 0110 : 2 / 9 1010 : 1 / 8 1110 : 2 / 3<br>0011 : 1 / 5 0111 : 2 / 11 1011 : Reversed 1111 : 1 / 2 |
| 3 | R/W | **CPU Core Frequency Ratio controll selection**<br>0: By external pin, PHLDA#, PHOLD#, GPCS0#, BM_REQ#<br>1: By bit 7 ~ 4 of this register |
| 2:0 | R/W | **Reserved**<br>Reserved and must be set to '0' |

## 7.7 OTHER REGISTERS

**Register 4D0h IRQ Edge/Level Control Register 1**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **IRQ7. 0: Edge sensitive 1: Level sensitive** |
| 6 | R/W | **IRQ6. 0: Edge sensitive 1: Level sensitive** |
| 5 | R/W | **IRQ5. 0: Edge sensitive 1: Level sensitive** |
| 4 | R/W | **IRQ4. 0: Edge sensitive 1: Level sensitive** |
| 3 | R/W | **IRQ3. 0: Edge sensitive 1: Level sensitive** |
| 2 | R/W | **IRQ2. This bit must be set to 0** |
| 1 | R/W | **IRQ1. This bit must be set to 0** |
| 0 | R/W | **IRQ0. This bit must be set to 0** |

**Register 4D1h IRQ Edge/Level Control Register 2**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **IRQ15. 0: Edge sensitive 1: Level sensitive** |
| 6 | R/W | **IRQ14. 0: Edge sensitive 1: Level sensitive** |
| 5 | R/W | **IRQ13. 0: Edge sensitive 1: Level sensitive** |
| 4 | R/W | **IRQ12. 0: Edge sensitive 1: Level sensitive** |
| 3 | R/W | **IRQ11. 0: Edge sensitive 1: Level sensitive** |
| 2 | R/W | **IRQ10. 0: Edge sensitive 1: Level sensitive** |
| 1 | R/W | **IRQ9. 0: Edge sensitive 1: Level sensitive** |
| 0 | R/W | **IRQ8. This bit must be set to 0** |

## 7.8 USB OPENHCI HOST CONTROLLER CONFIGURATION SPACE

The base address of these registers is programmable by the memory base address register (USB PCI configuration register offset 10-13h). These registers should be written as Dword, byte writes to these registers have unpredictable effects.

The OpenHCI Host Controller (HC) contains a set of on-chip operational registers, which are mapped into a non-cacheable portion of the system addressable space. These registers are used by the Host Controller Driver (HCD). According to the function of these registers, they are divided into four partitions, specifically for Control and Status, Memory Pointer, Frame Counter and Root Hub. All of the registers should be read and written as Dwords.

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.

### 7.8.1 CONTROL AND STATUS PARTITION

**Register 00h HcRevision Register**

Default Value: 00000110h

Access: Read

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:9 | | **Reserved** |
| 8 | RO | **Legacy**<br>This read-only field is 1 to indicate that the legacy support registers are present in this HC. |
| 7:0 | RO | **Revision**<br>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**

Default Value: 00000000h

Access: Read/Write

The HcControl register defines the operating modes for the Host Controller. Only the Host Controller Driver, except Host Controller Functional State and Remote Wakeup Connected modifies most of the fields in this register.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:11 | | **Reserved** |
| 10 | R/O | **RemoteWakeupEnable**<br>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 Resume Detected bit in Hc Interrupt Status is set, a remote wakeup is signaled to the host system. Setting this bit has no impact on the generation of hardware interrupt.<br>Since there is no remote wakeup supported, this bit is ignored. |

| | | |
|---|---|---|
| 9 | RO | **Remote Wakeup Connected**<br>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 software reset. Remote wakeup signaling of the host system is host-bus-specific and is not described in this specification.<br>This bit is hard-coded to '0'. |
| 8 | R/W | **Interrupt Routing**<br>This bit determines the routing of interrupts generated by events registered in Hc Interrupt Status. 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 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. |
| 7:6 | R/W | **HostControllerFunctionalState for USB**<br>00b: UsbReset<br>01b: UsbResume<br>10b: UsbOperational<br>11b: UsbSuspend<br>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.<br>This field may be changed by HC only in the UsbSuspend state. HC may move from the UsbSuspend state to the UsbResume state after detecting the resume signal from a downstream port.<br>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 signal to downstream ports. |
| 5 | R/W | **BulkListEnable**<br>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 the processing of the list. |

| | | |
|---|---|---|
| 4 | R/W | **ControlListEnable**<br>This bit is set to enable the processing of the Control list in the next Frame. If cleared by HCD, the 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 the processing of the list. |
| 3 | R/W | **IsochronousEnable**<br>This bit is used by HCD to enable/disable the 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). |
| 2 | R/W | **PeriodicListEnable**<br>This bit is set to enable the processing of the periodic list in the next Frame. If cleared by HCD, the processing of the periodic list does not occur after the next SOF. HC must check this bit before it starts processing the list. |
| 1:0 | R/W | **ControlBulkServiceRatio**<br>This specifies the service ratio between Control and Bulk EDs. Before processing any of the non-periodic lists, HC must compare the ratio specified with its internal count on how many non-empty 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.<br>**CBSR** — **No. of Control EDs Over Bulk EDs Served**<br>0 — 1:1<br>1 — 2:1<br>2 — 3:1<br>3 — 4:1 |

**Register 08h HcCommandStatus Register**

Default Value: 00000000h

Access: Read/Write

The HcCommandStatus register is used by the Host Controller to receive commands issued by the Host Controller Driver, as well as reflect 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 those bits written as '1' become set in the register while those 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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:18 | | **Reserved** |
| 17:16 | RO | **Scheduling Overrun Count**<br>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 Scheduling Overrun in Hc Interrupt Status has already been set. This is used by HCD to monitor any persistent scheduling problems. |
| 15:4 | | **Reserved** |
| 3 | R/W | **OwnershipChangeRequest**<br>This bit is set by an OS HCD to request a change of control of the HC. When set HC will set the Ownership Change field in HcInterrupt Status. After the changeover, this bit is cleared and remains so until the next request from OS HCD. |
| 2 | R/W | **BulkListFilled**<br>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 a ED in the Bulk list.<br>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. |

| | | |
|---|---|---|
| 1 | R/W | **ControlListFilled**<br>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.<br>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 |
| 0 | R/W | **HostControllerReset**<br>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 signal should be asserted to its downstream ports. |

**Register 0Ch *HcInterruptStatus* Register**

Default Value: 00000000h

Access: Read/Write

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 is 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 | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | | **Reserved** |
| 30 | R/W | **OwnershipChange Status**<br>This bit is set by HC when HCD sets the OwnershipChangeRequest field in HcCommandStatus. This event, when unmasked, will always generate a System Management Interrupt (SMI#) immediately. |
| 29:7 | | **Reserved** |

| | | |
|---|---|---|
| 6 | R/W | **RootHubStatusChange Status**<br>This bit is set when the content of HcRhStatus or the content of any of HcRhPortStatus[NumberofDownstreamPort] has changed. |
| 5 | R/W | **FrameNumberOverflow Status**<br>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. |
| 4 | RO | **UnrecoverableError Status**<br>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.<br>This event is not implemented and is hard-coded to '0'. |
| 3 | R/W | **ResumeDetected Status**<br>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. |
| 2 | R/W | **StartofFrame Status**<br>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. |
| 1 | R/W | **WritebackDoneHead Status**<br>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. |
| 0 | R/W | **SchedulingOverrun Status**<br>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**

Default Value: 00000000h

Access: Read/Write

Each enable bit in the *HcInterruptEnable* register corresponds to an associated interrupt bit in the *HcInterruptStatus* register. The *HcInterruptEnable* register is used to control those 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 | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | 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. |
| 30 | R/W | **Ownership Change Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Ownership Change. |
| 29:7 | | **Reserved** |
| 6 | R/W | **RootHub tatus Change Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Root Hub Status Change. |
| 5 | R/W | **FrameNumberOverflow Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Frame Number Overflow. |
| 4 | R/W | **UnrecoverableError Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Unrecoverable Error. |
| 3 | R/W | **ResumeDetected Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Resume Detect. |
| 2 | R/W | **StartFrame Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Start of Frame. |
| 1 | R/W | **WritebackDoneHead Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to HcDoneHead Writeback |
| 0 | R/W | **SchedulingOverrun Enable**<br>0 : Ignore<br>1 : Enable interrupt generation due to Scheduling Overrun. |

**Register 14h *HcInterruptDisable* Register**

Default Value: 00000000h

Access: Read/Write

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 | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | 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. |
| 30 | R/W | **OwnershipChange Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Ownership Change. |
| 29:7 | | **Reserved** |
| 6 | R/W | **RootHubStatusChange Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Root Hub Status Change. |
| 5 | R/W | **FrameNumberOverflow Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Frame Number Overflow. |
| 4 | R/W | **UnrecoverableError Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Unrecoverable Error. |
| 3 | R/W | **ResumeDetected Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Resume Detect. |
| 2 | R/W | **StartFrame Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Start of Frame. |
| 1 | R/W | **WritebackDoneHead Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to HcDoneHead Writeback. |
| 0 | R/W | **Scheduling Overrun Disable**<br>0 : Ignore<br>1 : Disable interrupt generation due to Scheduling Overrun. |

### 7.8.2 MEMORY POINTER PARTITION

**Register 18h HcHCCA Register**

Default Value: 00000000h

Access: Read/Write

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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:8 | R/W | This is the base address of the Host Controller Communication Area. |
| 7:0 | | **Reserved.** |

**Register 1Ch HcPeriodCurrentED Register**

Default Value: 00000000h

Access: Read/Write

The HcPeriodCurrentED register contains the physical address of the current Isochronous or Interrupt Endpoint Descriptor.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **PeriodCurrentED**<br>This is used by HC to point to the head of one of the Periodic lists that 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. |
| 3:0 | | **Reserved** |

**Register 20h *HcControlHeadED* Register**

Default Value: 00000000h

Access: Read/Write

The *HcControlHeadED* register contains the physical address of the first Endpoint Descriptor of the Control list.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **ControlHeadED**<br>HC traverses the Control list starting with the HcControlHeadED pointer. The content is loaded from HCCA during the initialization of HC. |
| 3:0 | | **Reserved.** |

**Register 24h HcControlCurrentED Register**

Default Value: 00000000h

Access: Read/Write

The HcControlCurrentED register contains the physical address of the current Endpoint Descriptor of the Control list.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **ControlCurrentED**<br>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. |
| 3:0 | | **Reserved.** |

**Register 28h HcBulkHeadED Register**

Default Value: 00000000h

Access: Read/Write

The HcBulkHeadED register contains the physical address of the first Endpoint Descriptor of the Bulk list.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **BulkHeadED**<br>HC traverses the Bulk list starting with the HcBulkHeadED pointer. The content is loaded from HCCA during the initialization of HC. |
| 3:0 | | **Reserved.** |

**Register 2Ch HcBulkCurrentED Register**

Default Value: 00000000h

Access: Read/Write

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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **BulkCurrentED**<br>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. |
| 3:0 | | **Reserved.** |

**Register 30h HcDoneHead Register**

Default Value: 00000000h

Access: Read/Write

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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:4 | R/W | **DoneHead**<br>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.<br>This is set to zero whenever HC writes the content of this register to HCCA. It also sets the WritebackDoneHead of HcInterruptStatus. |
| 3:0 | | **Reserved** |

### 7.8.3 BITS FRAME COUNTER PARTITION

**Register 34h HcFmInterval Register**

Default Value: 00002EDFh

Access: Read/Write

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 | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | R/W | **FrameIntervalToggle**<br>HCD toggles this bit whenever it loads a new value to FrameInterval. |
| 30:16 | R/W | **FSLargestDataPacket**<br>This field specifies a value that 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. |
| 15:14 | | **Reserved** |
| 13:0 | R/W | **FrameInterval**<br>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**

Default Value: 00000000h

Access: Read Only

The HcFmRemaining register is a 14-bit down counter showing the bit time remaining in the current Frame.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | RO | **FrameRemainingToggle**<br>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. |
| 30:14 | | **Reserved** |
| 13:0 | RO | **FrameRemaining**<br>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**

Default Value: 00000000h

Access: Read

The HcFmNumber register is a 16-bit counter. It provides a timing reference among events occurring 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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:16 | | **Reserved** |
| 15:0 | RO | **FrameNumber**<br>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**

Default Value: 00000000h

Access: Read/Write

The HcPeriodicStart register has a 14-bit programmable value which determines when is the earliest time HC should start processing the periodic list.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:14 | | **Reserved** |
| 13:0 | R/W | **PeriodicStart**<br>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**

Default Value: 00000000h

Access: Read/Write

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 is allowed to change this value.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:12 | | **Reserved** |
| 11:0 | R/W | **LSThreshold**<br>This field contains a value that 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. |

### 7.8.4 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 that 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 [2:1]. 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**

Default Value: 01000002h

Access: Read/Write

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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:24 | R/W | **PowerOnToPowerGoodTime**<br>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. |
| 23: 13 | | **Reserved** |

| | | |
|---|---|---|
| 12 | R/W | **NoOverCurrentProtection**<br>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.<br>0 : Over-current status is reported collectively for all downstream ports<br>1 : No overcurrent protection supported |
| 11 | R/W | **OverCurrentProtectionMode**<br>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.<br>0 : over-current status is reported collectively for all downstream ports<br>1 : over-current status is reported on a per-port basis |
| 10 | RO | **DeviceType**<br>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. |
| 9 | R/W | **NoPowerSwitching**<br>These bits are used to specify whether power switching is supported or port is always powered. SiS5595 USB HC supports global power switching mode. When this bit is cleared, the PowerSwitchingMode specifies global or per-port switching.<br>0 : Ports are power switched<br>1 : Ports are always powered on when the HC is powered on |
| 8 | R/W | **PowerSwitchingMode**<br>This bit is used to specify how the power switching of the Root Hub ports is controlled. SiS5595 USB HC supports global power switching mode. This field is only valid if the NoPowerSwitching field is cleared.<br>0 : all ports are powered at the same time.<br>1 : Each port is powered individually. This mode allows port power to be controlled by either the global switch or per-port switching.<br>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/ Clear Global Power). |
| 7:0 | RO | **NumberDownstreamPorts**<br>These bits specify the number of downstream ports supported by the Root Hub.<br>SiS5595 USB HC supports two downstream ports. |

**Register 4Ch HcRhDescriptorB Register**

Default Value: 00000000h

Access: Read/Write

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.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:16 | R/W | **PortPowerControlMask**<br>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.<br>SiS5595 USB HC implements global power switching.<br>bit 0: Reserved<br>bit 1: Ganged-power mask on Port #1<br>bit 2: Ganged-power mask on Port #2<br>...<br>bit15: Ganged-power mask on Port #15 |
| 15:0 | R/W | **DeviceRemovable**<br>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.<br>bit 0: Reserved<br>bit 1: Device attached to Port #1<br>bit 2: Device attached to Port #2<br>...<br>bit15: Device attached to Port #15 |

**Register 50h HcRhStatus Register**

Default Value: 00000000h

Access: Read/Write

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 | ACCESS | DESCRIPTION |
|---|---|---|
| 31 | WO | **ClearRemoteWakeupEnable(Write)**<br>Writing a '1' clears DeviceRemoveWakeupEnable. Writing a '0' has no effect. |
| 30:18 | | **Reserved** |

| | | |
|---|---|---|
| 17 | R/W | **OverCurrentIndicatorChange**<br>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. |
| 16 | R/W | **LocalPowerStatusChange(Read)**<br>The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.<br>**SetGlobalPower(Write)**<br>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. |
| 15 | R/W | **DeviceRemoteWakeupEnable(Read)**<br>This bit enables a ConnectStatusChange bit as a resume event, causing a UsbSuspend to UsbResume state transition and setting the ResumeDetected interrupt.<br>0 : **ConnectStatusChange** is not a remote wakeup event.<br>1 : **ConnectStatusChange** is a remote wakeup event.<br>**SetRemoteWakeupEnable(Write)**<br>Writing a '1' sets DeviceRemoveWakeupEnable. Writing a '0' has no effect. |
| 14:2 | | **Reserved** |
| 1 | RO | **OverCurrentIndicator**<br>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' |
| 0 | R/W | **LocalPowerStatus(Read)**<br>The Root Hub does not support the local power status feature; thus, this bit is always read as '0'.<br>**ClearGlobalPower(Write)**<br>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[2:1] Register**

Default Value: 00000000h

Access: Read/Write

The HcRhPortStatus[2:1] 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'.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:21 | | **Reserved** |
| 20 | R/W | **PortResetStatusChange**<br>This bit is set at the end of the 10-ms port reset signal.<br>The HCD writes a '1' to clear this bit. Writing a '0' has no effect.<br>0 : port reset is not complete<br>1 : port reset is complete |
| 19 | R/W | **PortOverCurrentIndicatorChange**<br>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.<br>0 : no change in PortOverCurrentIndicator<br>1 : PortOverCurrentIndicator has changed |
| 18 | R/W | **PortSuspendStatusChange**<br>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.<br>0 : resume is not completed<br>1 : resume completed |
| 17 | R/W | **PortEnableStatusChange**<br>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.<br>0 : no change in **PortEnableStatus**<br>1 : change in PortEnableStatus |

| | | |
|---|---|---|
| 16 | R/W | **ConnectStatusChange**<br>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.<br>0 : no change in CurrentConnectStatus<br>1 : change in CurrentConnectStatus<br>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. |
| 15:10 | | **Reserved** |
| 9 | R/W | **LowSpeedDeviceAttached((Read))**<br>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.<br>0 : full speed device attached<br>1 : low speed device attached |
| 8 | R/W | **Port Power Status((Read))**<br>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 Set Port Power or Set Global Power. HCD clears this bit by writing Clear Port Power or Clear Global Power. Which power control switches are enabled is determined by Power Switching Mode and Port Port Control Mask[NDP]. In global switching mode (Power Switching Mode=0), only Set/ClearGlobalPower controls this bit. In per-port power switching (Power Switching Mode=1), if the Port Power Control Mask[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ Clear Global Power commands are enabled. When port power is disabled, Current Connect Status, Port Enable Status, Port Suspend Status, and Port Reset Status should be reset.<br>0 : port power is off<br>1 : port power is on<br>**SetPortPower(Write)**<br>The HCD writes a '1' to set the PortPowerStatus bit. Writing a '0' has no effect.<br>Note: This bit is always reads '1b' if power switching is not supported. |
| 7:5 | | **Reserved** |

| | | |
|---|---|---|
| 4 | R/W | **PortResetStatus(Read)**<br>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.<br>0 : port reset signal is not active<br>1 : port reset signal is active<br>**SetPortReset(Write)**<br>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. |
| 3 | R/W | **PortOverCurrentIndicator(Read)**<br>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<br>0 : no overcurrent condition.<br>1 : overcurrent condition detected.<br>**ClearSuspendStatus(Write)**<br>The HCD writes a '1' to initiate a resume. Writing a '0' has no effect. A resume is initiated only if PortSuspendStatus is set. |
| 2 | R/W | **PortSuspendStatus(Read)**<br>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.<br>0 : port is not suspended<br>1 : port is suspended<br>**SetPortSuspend(Write)**<br>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. |

| | | |
|---|---|---|
| 1 | R/W | **PortEnableStatus(Read)**<br>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.<br>0 : port is disabled<br>1 : port is enabled<br>**SetPortEnable(Write)**<br>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. |
| 0 | R/W | **CurrentConnectStatus(Read)**<br>This bit reflects the current state of the downstream port.<br>0 : no device connected<br>1 : device connected<br>**ClearPortEnable(Write)**<br>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.<br>Note: This bit is always read '1b' when the attached device is nonremovable (DeviceRemoveable[NDP]). |

### 7.8.5 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 7.8-1 Legacy Support Registers**

| OFFSET | REGISTER | DESCRIPTION |
|---|---|---|
| 100h | HceControl | Used to enable and control the emulation hardware and report various status informations. |
| 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, and 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 7.8-2 Emulated Registers

**Table 7.8-2 Emulated Registers**

| 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**

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:9 | | **Reserved** |
| 8 | R/W | **A20State**<br>Indicates current state of Gate A20 on keyboard controller. Used to compare against value to 60h when GateA20Sequence is active. |
| 7 | R/W | **IRQ12Active**<br>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. |
| 6 | R/W | **IRQ1Active**<br>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. |
| 5 | R/W | **GateA20Sequence**<br>Set by HC when a data value of D1h is written to I/O port 64h. Cleared by HC on write to I/O port 64h of any value other than D1h. |
| 4 | R/W | **ExternalIRQEn**<br>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. |

| | | |
|---|---|---|
| 3 | R/W | **IRQEn**<br>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. |
| 2 | R/W | **CharacterPending**<br>When set, an emulation interrupt is generated when the **OutputFull** bit of the *HceStatus* register is set to 0. |
| 1 | RO | **EmulationInterrupt**<br>This bit is a static decode of the emulation interrupt condition |
| 0 | R/W | **EmulationEnable**<br>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**

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:8 | | **Reserved** |
| 7:0 | R/W | **InputData**<br>This register holds data that is written to I/O ports 60h and 64h.<br>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**

Default Value: 00000000h

Access: Read/write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:8 | | **Reserved** |
| 7:0 | R/W | **OutputData**<br>This register hosts data that is returned when an I/O read of port 60h is performed by application software.<br>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**

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:8 | | **Reserved** |
| 7 | R/W | **Parity**<br>Indicates parity error on keyboard/mouse data. |
| 6 | R/W | **Time-out**<br>Used to indicate a time-out |
| 5 | R/W | **AuxOutputFull**<br>IRQ12 is asserted whenever this bit is set to 1 and **OutputFull** is set to 1 and the **IRQEn** bit is set. |
| 4 | R/W | **Inhibit Switch**<br>This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited. |
| 3 | R/W | **CmdData**<br>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 |
| 2 | R/W | **Flag**<br>Nominally used as a system flag by software to indicate a warm or cold boot. |
| 1 | R/W | **InputFull**<br>Except for the case of a Gate A20 sequence, this bit is set to 1 on an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists. |
| 0 | R/W | **OutputFull**<br>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.<br>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. |

# 8 ELECTRICAL CHARACTERISTICS

## 8.1 SiS5595 INTERNAL POWER PLANES

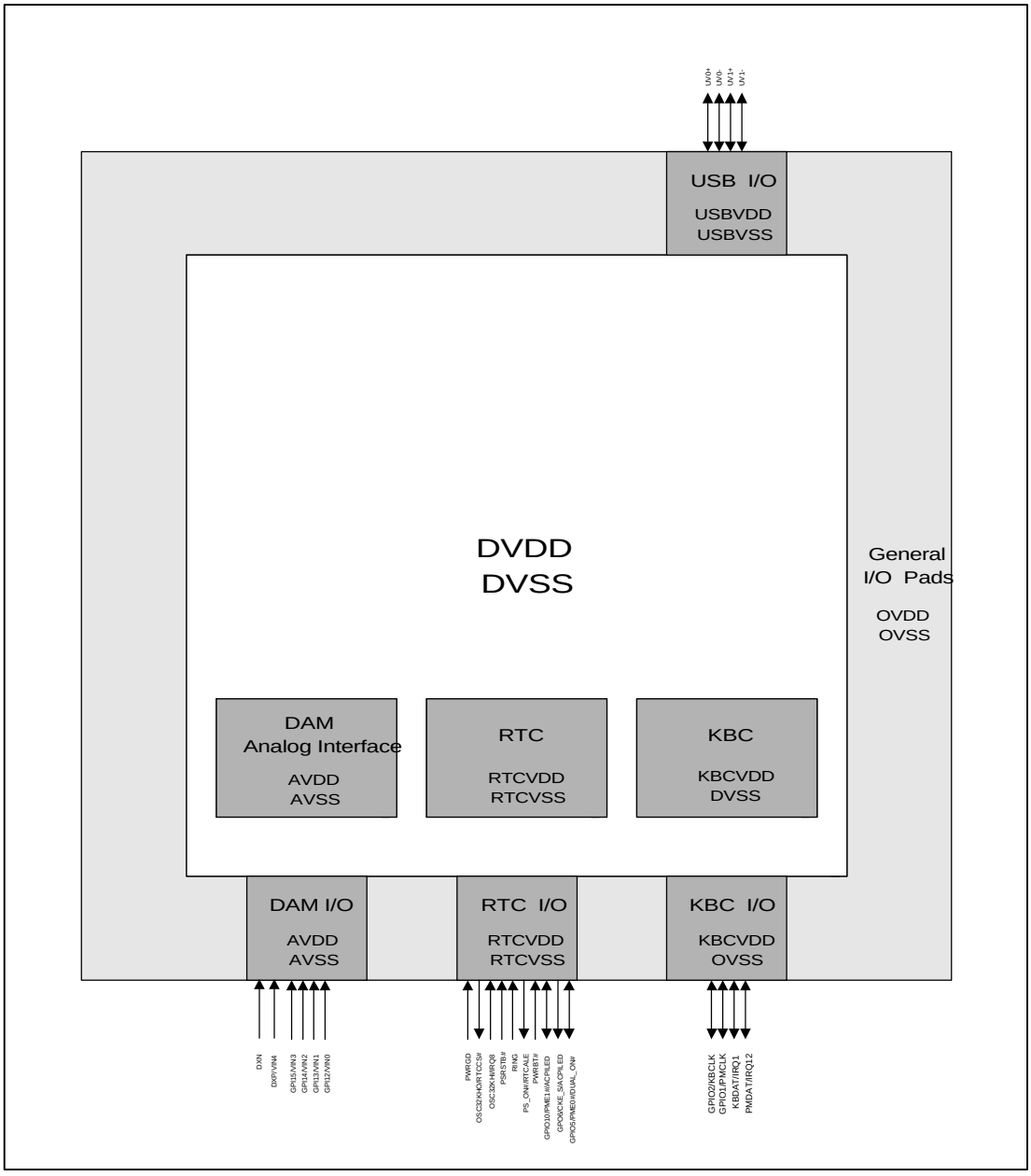


**Figure 8.1-1 SiS5595 Internal Power Planes**

**DAM:** Data Acquisition Module

## 8.2 ABSOLUTE MAXIMUM RATINGS

**Table 8.2-1 Min. and Max. Voltage and Temperature Table**

| PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|
| Ambient operating temperature | 0 | 70 | |
| Storage temperature | -40 | 125 | |
| Input voltage | -0.3 | 5.5 | V |
| Output voltage | -0.5 | 5.5 | V |

Note: The stress test condition listed above may cause permanent damage to device.

## 8.3 DC CHARACTERISTICS

**Table 8.3-1 SiS5595 DC Characteristics Table**

Ta=0-70 Celsius, VSS=0V, VDD=5V+/-5%

| SYMBOL | PARAMETER | MIN | MAX | NOTE |
|---|---|---|---|---|
| VIL1 | Input Low Voltage | -0.3V | 1.6V | Note1 |
| VIH1 | Input High Voltage | 3.2V | VDD+0.3V | Note1 |
| VIL2 | Input Low Voltage | -0.3V | 0.8V | Note1 |
| VIH2 | Input High Voltage | 1.8V | VDD+0.3V | Note1 |
| VIL3 | Input Low Voltage | -0.3V | 0.8V | Note1 |
| VIH3 | Input High Voltage | 2.0V | VDD+0.3V | Note1 |
| VOL | Output Low Voltage | | 0.45V | Note1 |
| VOH | Output High Voltage | 2.4V | | Note1 |
| IOL1 | Output Low Current | 0.7mA | | Note2 |
| IOL2 | Output Low Current | 4mA | | Note2 |
| IOH2 | Output High Current | 4mA | | Note2 |
| IOL3 | Output Low Current | 6mA | | Note2 |
| IOH3 | Output High Current | 6mA | | Note2 |
| IOL4 | Output Low Current | 8mA | | Note2 |
| IOH4 | Output High Current | 8mA | | Note2 |
| IOL5 | Output Low Current | 16mA | | Note2 |
| IOH5 | Output High Current | 16mA | | Note2 |
| IIH | Input Leakage Current | | -10uA | |
| IIL | Input Leakage Current | | 10uA | |
| CIN | Input Capacitance | | 12pF | Fc=1MHZ |
| COUT | Output Capacitance | | 12pF | Fc=1MHZ |
| CI/O | I/O Capacitance | | 12pF | Fc=1MHZ |

**Note 1 :**

- **VIL1/VIH1**

  Apply to following pins:

  PWRGD, PSRSTB#

- **VIL2/VIH2**

  Apply to following pins:

  GPI[15:12]/VIN[4:0],

  IRQ[3,4,5,6,7,9,10,14,15],

  GPIO3/CPU_STOP3/SLP#

  GPIO7/OC0#/PPS

  GPIO8/OC1#,GPIO2/KBCLK

  GPIO1/PMCLK

  GPIO16/IOCHK#

  GPIO4/FAN1

  GPIO11/FAN2

  GPIO5/PME0#/DUAL_ON#

  GPIO9/THERM#/BTI/SMBALERT#

  OSC32KHI/IRQ8#

  GPIO10/PMU1#/ACPILED

  PWRBT#

- **VIL3/VIH3**

  Apply to the rest input or bi-directional pins.

**Note 2 :**

- **IOL1**

  Apply to following pins:

  INIT, CPURST, NMI, INTR

- **IOL3/IOH3**

  Apply to following pins:

  IORC#, IOWC#, MWDC#, MWTC#, SMRDC#, SMWTC#

  IOCHRDY, RFH#, BCLK, BALE, AEN, SHBE#,

  UV0+, UV0-, UV1+, UV1-

- **IOL4/IOH4**

  Apply to following pins:

  DDCCLK, DDCDAT, PCIRST#, AD31~AD0,

  FRAME#, IRDY#, DEVSEL#,TRDY#,STOP#,

  GPIO0/PAR

- **IOL5/IOH5**

  Apply to following pins:

  KBDAT/IRQ1, GPIO2/KBCLK,

  PMDAT/IRQ12, GPIO1/PMCLK,

  GPO6/CKE_S#/ACPILED,

- **IOL2/IOH2**

  Apply to the rest output or bi-directional pins

## 8.4 INTERNAL RTC POWER CONSUMPTION

**Table 8.4-1 RTC Power Consumption Table**

| RTCVDD (V) | OPERATION CURRENT (uA) | POWER CONSUMPTION (uW) |
|---|---|---|
| 3.6 | 5.5 | 18 |
| 3.3 | 4.5 | 14.85 |
| 3.0 | 3.6 | 10.8 |
| 2.7 | 2.9 | 7.83 |
| 2.4 | 2.3 | 5.52 |
| 2.1 | 1.7 | 3.57 |
| 1.8 | 1.3 | 2.34 |

The minimum operation voltage of internal RTC (RTCVDD) is 2.0V, the recommended voltage is 2.2V to 2.7V.

## 8.5 AC CHARACTERISTICS

### 8.5.1 SiS5595 DMA CONTROLLER AC CHARACTERISTICS

**Table 8.5-1 DMA Controller AC Characteristics Table**

| SYM | PARAMETER | MAX | MIN | UNIT | CL |
|---|---|---|---|---|---|
| T1 | DACK# active to IORC# active | | 136.5 | ns | 35pf |
| T2 | DACK# active to IOWC# active | | 375.2 | ns | 35pf |
| T3 | DACK# active hold from IORC# inactive | | 112.6 | ns | 35pf |
| T4 | DACK# active hold from IOWC# inactive | | 170.7 | ns | 35pf |
| T5 | AEN active to IORC# active | | 1.192 | us | 120pf |
| T6 | AEN active to IOWC# active | | 1.430 | us | 120pf |
| T7 | AEN inactive from IORC# inactive | | 297.5 | ns | 120pf |
| T8 | AEN inactive from IOWC# inactive | | 355.6 | ns | 120pf |
| T9 | BALE active to IORC# active | | 367.1 | ns | 120pf |
| T10 | BALE active to IOWC# active | | 607.8 | ns | 120pf |
| T11 | BALE inactive from IORC# inactive | | 193.1 | ns | 120pf |
| T12 | BALE inactive from IOWC# inactive | | 251.2 | ns | 120pf |
| T13 | LA,SA,SBHE# valid setup time to MRDC# | | 239.3 | ns | 120pf |
| T14 | LA,SA,SBHE# valid setup time to MWTC# | | 477.9 | ns | 120pf |
| T15 | LA,SA,SBHE# valid hold time from MRDC# | | 75.7 | ns | 120pf |
| T16 | LA,SA,SBHE# valid hold time from MWTC# | | 133.7 | ns | 120pf |
| T17 | IORC# pulse width | | 767.8 | ns | 120pf |
| T18 | IOWC# pulse width | | 471.0 | ns | 120pf |
| T19 | MRDC# pulse width | | 766.9 | ns | 120pf |
| T20 | MWTC# pulse width | | 470.3 | ns | 120pf |
| T21 | MWTC# active from IORC# active | | 239.7 | ns | 120pf |
| T22 | IOWC# active from MRDC# active | | 237.1 | ns | 120pf |
| T23 | IORC# inactive from MWTC# inactive | | 57.8 | ns | 120pf |
| T24 | MRDC# inactive from IOWC# inactive | | 58.7 | ns | 120pf |
| T25 | IORC# inactive to active | | 192.2 | ns | 120pf |
| T26 | IOWC# inactive to active | | 489.0 | ns | 120pf |
| T27 | data valid setup to IOWC# inactive | | 505.6 | ns | 120pf |
| T28 | data valid hold from IOWC# inactive | | 89.1 | ns | 120pf |
| T29 | TC active setup to IOWC# inactive | | 480.5 | ns | 120pf |
| T30 | TC active hold from IOWC# inactive | | 127.8 | ns | 120pf |
| T31 | CHRDY inactive from MRDC# active | 31.9 | | ns | 120pf |
| T32 | CHRDY inactive width | | 344.4 | ns | 120pf |
| T33 | MRDC# active hold from CHRDY active | | 253.8 | ns | 120pf |
| T34 | MWTC# active hold from CHRDY active | | 223.8 | ns | 120pf |

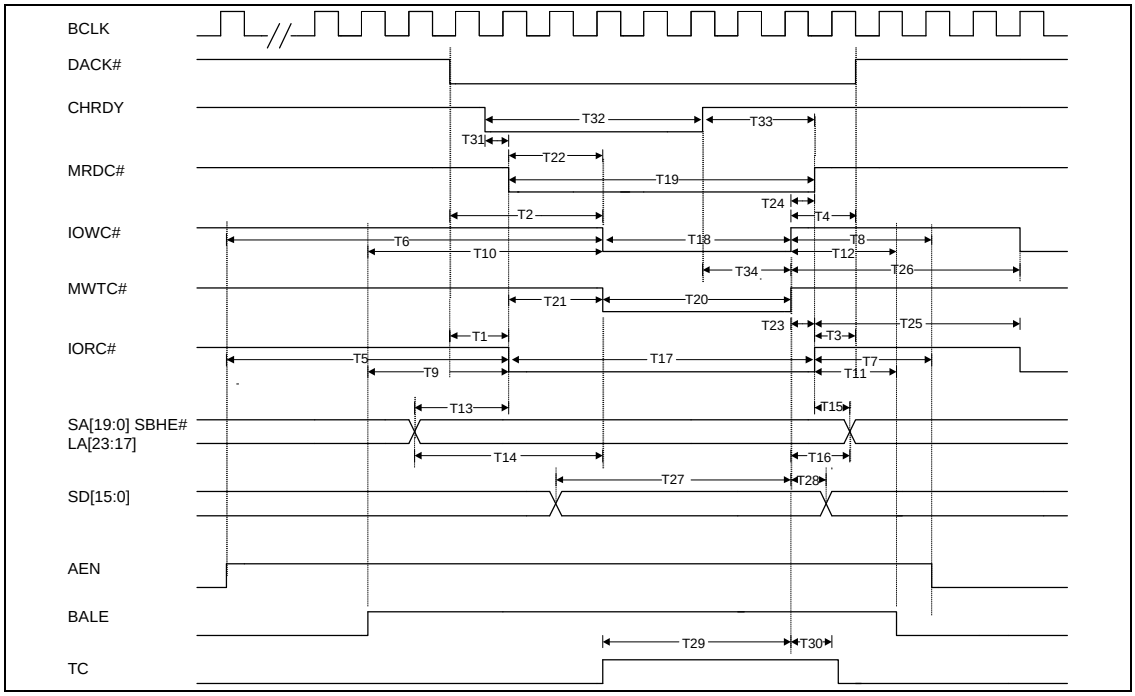


**Figure 8.5-1 DMA Cycles**

### 8.5.2 SiS5595 PCI-TO-ISA CYCLES AC CHARACTERISTICS

**Table 8.5-2 PCI-TO-ISA Cycles AC Characteristics Table**

| SYM | PARAMETER | MAX | MIN | UNIT | CL |
|---|---|---|---|---|---|
| T1 | BCLK cycle time | | 120 | ns | 120pf |
| T2 | IORC#<br>IOWC# inactive from BCLK rising<br>MRDC#<br>MWTC# | 4.6<br>4.4<br>4.8<br>4.9 | | ns | 120pf |
| T3 | LA[23:17] valid to M16# assertion | 202.2 | | ns | 120pf |
| T4 | SA[19:0] valid to IO16# assertion | 358.8 | | ns | 120pf |
| T5 | 16 bit IORC# pulse width 16 bit IOWC# pulse width | | 169.8<br>169.5 | ns | 120pf |
| T6 | 8-bit IORC# pulse width<br>8-bit IOWC# pulse width | | 529.8<br>529.5 | ns | 120pf |
| T7 | 16-bit MRDC# pulse width<br>16-bit MWTC# pulse width | | 230.1<br>229.5 | ns | 120pf |
| T8 | 8-bit MRDC# pulse width<br>8-bit MWTC# pulse width | | 529.7<br>529.1 | ns | 120pf |

| | | | | | |
|---|---|---|---|---|---|
| T9 | ROM MRDC# pulse width<br>ROM MWTC# pulse width | | 110.1<br>109.5 | ns | 120pf |
| T10 | SD data setup time to IORC#, MRDC# inactive | | 10 | ns | 120pf |
| T11 | SD data hold time to IORC#, MRDC# inactive | | 3 | ns | 120pf |
| T12 | SD[15:0] setup time to 16-bit IOWC# active<br>SD[15:0] setup time to 8-bit IOWC# active<br>SD[15:0] setup time to 16-bit MWCT# active<br>SD[15:0] setup time to 8-bit MWCT# active | | 214.9<br>214.9<br>157<br>215.9 | ns | 120pf |
| T13 | SD hold time from IOWC# inactive--disassembly cycle<br>SD hold time from MWTC# inactive--disassembly cycle | | 78<br>77.4 | ns | 120pf |
| T14 | SD[15:0] hold time from IOWC# inactive<br>SD[15:0] hold time from MWTC# inactive | | 8<br>171.6 | ns | 120pf |
| T15 | SA0, SA1, BXSBHE# hold time from the negation of IORC#, IOWC#, MWTC#, MRDC# | | 52.5 | ns | |
| T16 | ZWS# setup time to BCLK falling | | 10 | ns | 120pf |
| T17 | ZWS# hold time from BCLK falling | | 20 | ns | 120pf |
| T18 | LA[23:17] setup time to BALE falling | | 305.2 | ns | |
| T19 | LA[23:17] hold time to BALE falling | | 654.8 | ns | |
| T20 | LA[23:17] setup time to MWTC# falling | | 320.2 | ns | |
| T21 | SA[19:0] setup time to command active (8bits) | | 343.6 | ns | |
| T22 | SA[19:0] setup time to command active (16bits) | | 285.2 | ns | |
| T23 | SA[19:0] setup time to BALE falling | | 281.8 | ns | |

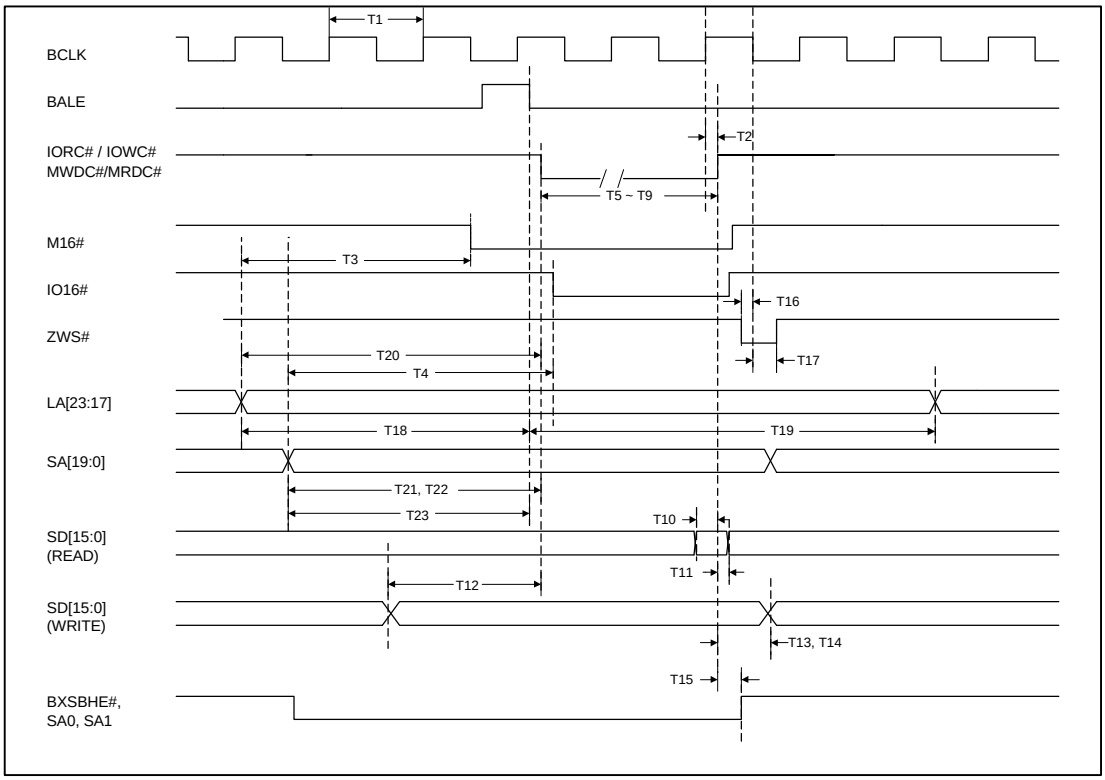


**Figure 8.5-2 The AC Timing Diagram of PCI to ISA Bus Cycles**

### 8.5.3 SiS5595 MISC. AC CHARACTERISTICS

**Table 8.5-3 MISC. AC Characteristics Table**

| SYM | PARAMETER | MAX | MIN | UNIT | CL |
|---|---|---|---|---|---|
| T1 | INIT active delay from PCICLK | | | ns | 35pf |
| T2 | INIT inactive delay from PCICLK | | | ns | 35pf |
| T3 | PCIRST# inactive delay from PCICLK | | | ns | 50pf |
| T4 | RFH# active setup to MRDC# active | | 341.79 | ns | 120pf |
| T5 | RFH# active hold from MRDC# inactive | | 30 | ns | 120pf |
| T6 | AEN active delay from RFH# active | 13.4 | | ns | 120pf |
| T7 | SERR#, IOCHK# active to NMI output floating active | 60.3 | | ns | |
| T8 | INT output floating delay from IRQ active | 20.2 | | ns | |
| T9 | IRQ active pulse width(in the edge trigger mode) | | 30 | ns | |
| T10 | IGNE # active delay from IOWC# active for port F0H access | | | ns | |
| T11 | IGNE# inactive delay from FERR# inactive | | | ns | |

| | | | | | |
|---|---|---|---|---|---|
| T12 | SPKR valid delay from 14Mhz | 12.57 | | ns | |
| T13 | RTCALE pulse width | 529.9 | | ns | |
| T14 | RTCALE active delay from IOWC# active | 11.96 | | ns | |
| T15 | PCIRST# active time after Power stable | | 1 | ms | |
| T16 | CPURST active time after Power stable | | 1 | ms | |
| T17 | Hard trap active time after CPURST inactive | | 7 | PCICLK | |
| T18 | INIT active time after shut down special cycle or keyboard fast reset cycle | | | | |
| T19 | INIT active time | | 15 | PCICLK | |
| T20 | CPURST active time after shut down special cycle or keyboard fast reset cycle | | 1 | ms | |

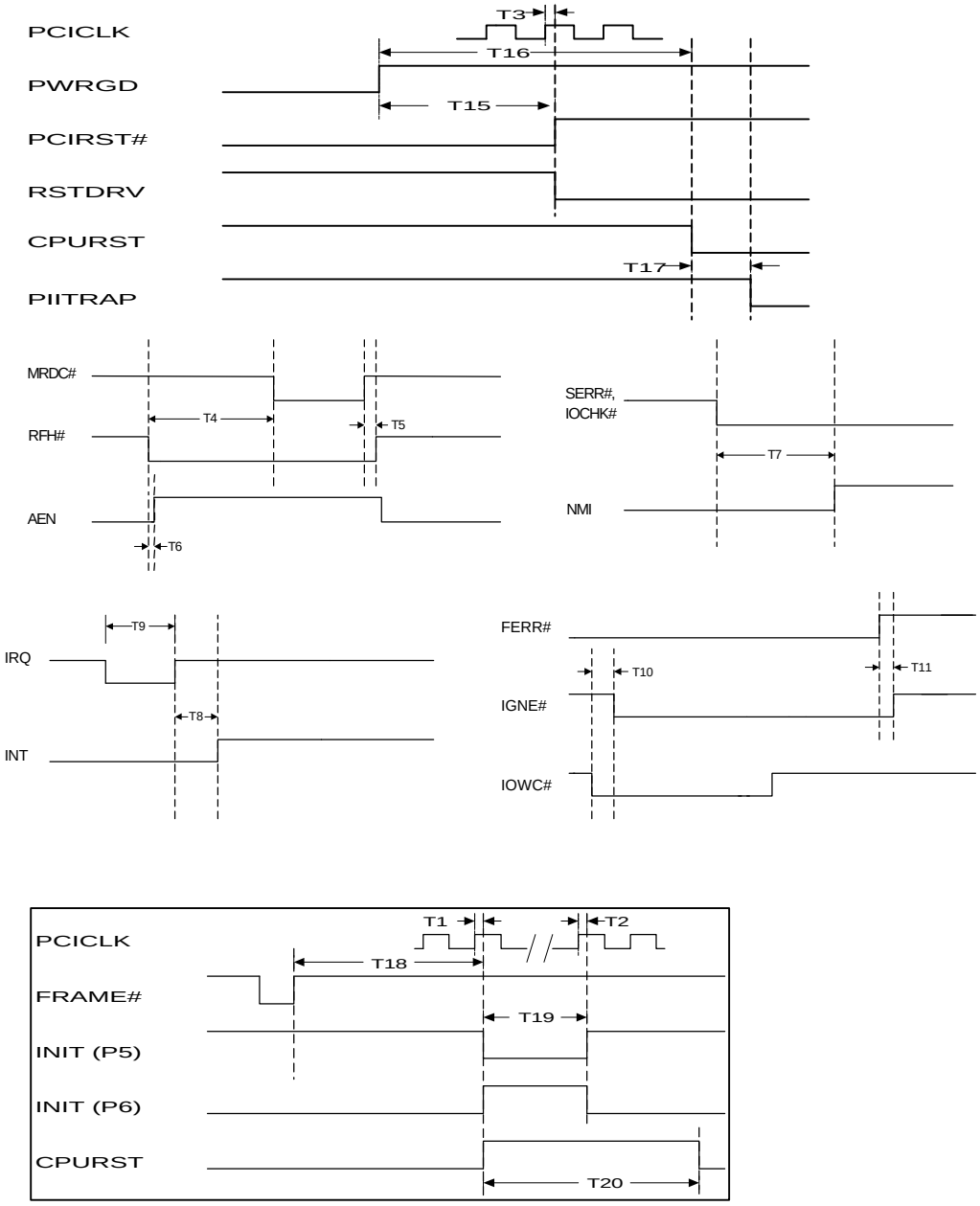


**Figure 8.5-3 Miscellaneous Timing**

# 9 MECHANICAL DIMENSION

## 9.1 SIS5595 (PQFP 208-PIN PLASTIC FLAT PACKAGE)

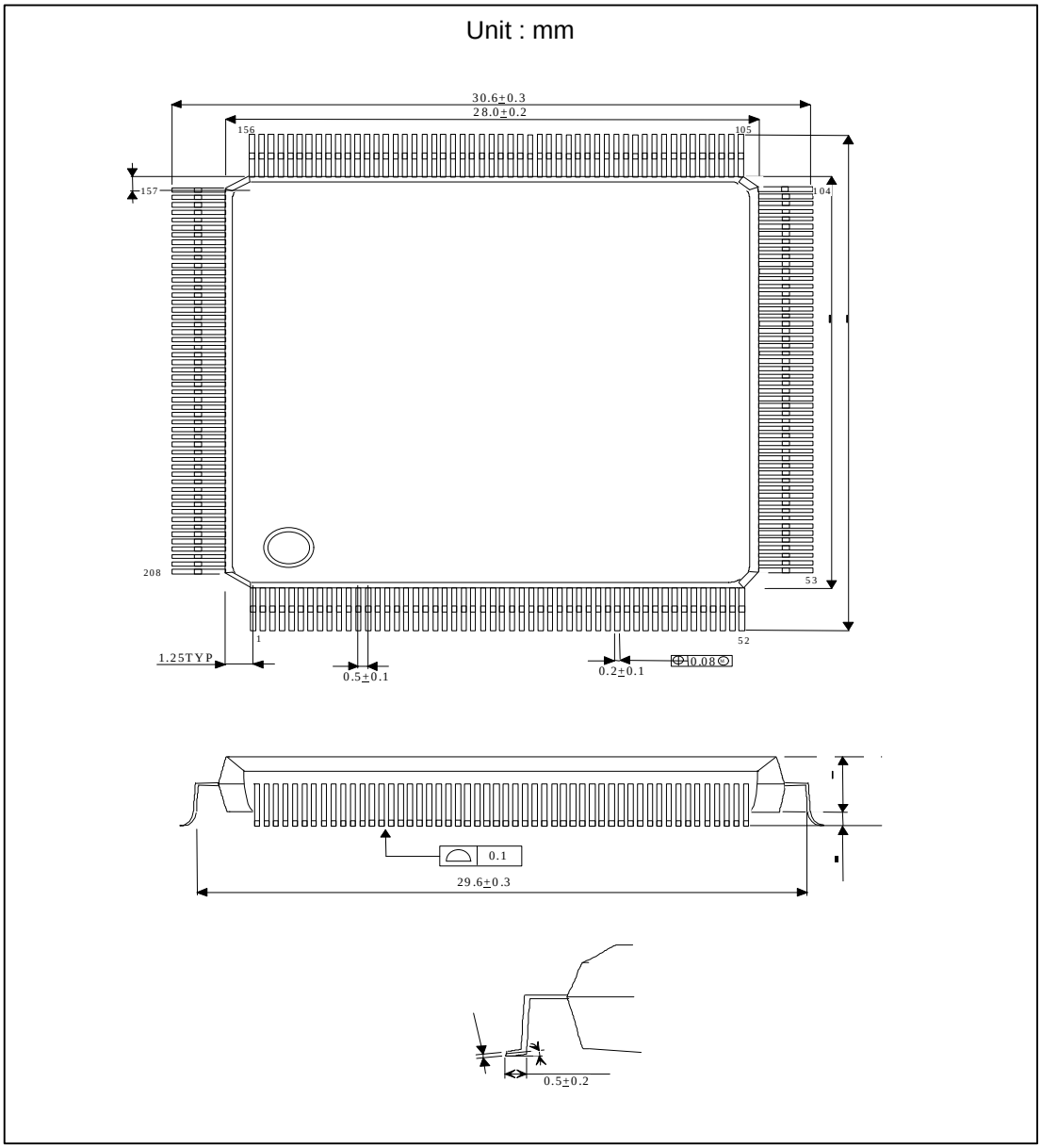


**Figure 9.1-1 SiS5595 Package Specification**

# 10 COPYRIGHT NOTICE



## TRADEMARKS



## DISCLAIMER



## RESTRICTED RIGHTS LEGEND