**CONTENTS**

## FIGURE



## TABLE

# 1. SiS600/SiS5595 OVERVIEW

***SiS600 PCI/ A.G.P./ Host/ Memory Controller***

***SiS5595 PCI SYSTEM I/O***

The SiS600/SiS5595 is a highly integrated Pentium II AGP chipset, which provides a high performance/cost index Desktop/Mobile solution for the Intel Pentium II AGP system.

The SiS600 AGP/PCI controller integrates the Host interface, Host-to-PCI bridge, the DRAM controller, the Accelerated Graphics Port interface, and the PCI IDE controller.

The host interface supports Pentium II bus for up to 100MHz. The DRAM controller can support EDO/FP/SDRAM memory up to 1.5GB with ECC function. The AGP 1.0 compliance interface supports both 1X, and 2X speed mode with side band address capability. The built-in fast PCI IDE controller supports the traditional ATA PIO/DMA, and the Ultra DMA/33 functionality.

The SiS5595 PCI system I/O integrates the PCI-to-ISA bridge with the DDMA, PC/PCI DMA and 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 temperature input. SiS5595 supports ACPI function to meet Advanced Configuration and Power Interface (ACPI) revision 1.0 specification for Windows 98 environment. It can support power-management timer, power button, power button over-ride, Real Time Clock (RTC) Alarm Wake up, LAN Wake up, more sleeping state, ACPI LED to indicate the system sleeping and working state.
In addition, SiS5595 also integrates the thermal detection and jumper free logic for Pentium II CPU to reduce the board cost.

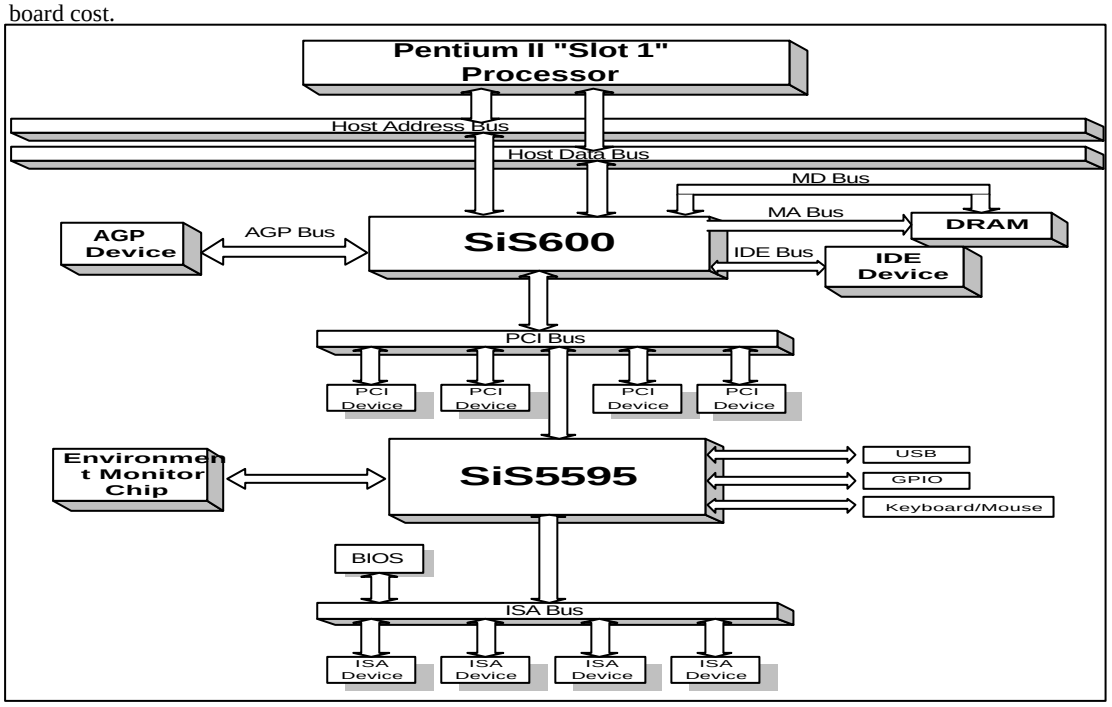



**Figure 1-1 SiS600/SiS5595 System Block Diagram**

## 2. FEATURES

### SiS600 PCI/A.G.P. BUS CONTROLLER

- **Supports Intel Pentium II CPU and host bus up to 100MHz**
- **Integrated DRAM Controller**
  - Supports 6/3 Banks(Single/Double sided) of FPM/EDO/SDRAM DIMM/SIMM
  - Supports 4M byes to 1.5G bytes of main memory
  - Provides X-4-4-4 to X-1-1-1 Burst Capability
  - Supports 512K/1M/2M/4M/8M/16M/32MxN DRAM devices
  - Auto Detection of FPM, EDO, and SDRAM
  - Supports Symmetric and Asymmetric DRAM
  - Configurable ECC Function Support
  - Supports ECC with Single Bit Error Correction and Multiple/nibble Bit Error Detection
  - Supports 3.3V DRAM
  - Supports CAS before RAS Refresh
  - Supports FP/EDO/SDRAM Self Refresh During Suspend Mode
  - Supports Relocation of System Management Memory
  - Programmable CAS#, RAS#, RAMWE#, SDRAS#, SDCAS#, CKE and MA Driving Current
  - Fully Configurable for the Characteristic of Shadow RAM ( 640K Bytes to 1M Bytes)
  - Supports FPM DRAM 8-3-3-3(-3-3-3-3) Burst Read Cycles without ECC
  - Supports FPM DRAM 8-3-3-3(-3-3-3-3) Burst Read Cycles with ECC
  - Supports EDO DRAM 8-2-2-2(-2-2-2-2) Burst Read Cycles without ECC
  - Supports EDO DRAM 8-2-2-2(-2-2-2-2) Burst Read Cycles with ECC
  - Supports SDRAM 8-1-1-1(-1-1-1-1) Burst Read Cycles without ECC
  - Supports SDRAM 9-1-1-1(-1-1-1-1) Burst Read Cycles with ECC
  - Supports SDRAM 2-2-2-2 for back to back single QW read cycles
  - Supports X-1-1-1/X-2-2-2/X-3-3-3 Burst Write Cycles
  - Supports SDRAM internal multi-bank operation (Up to 4 banks)
  - Two Programmable PCI-Hole Areas
  - Shadow RAM in Increments of 16 Kbytes
  - Page Table Cache Scheme for Mapping Graphical Texture Access to Physical Memory Address
  - Built-in 8 Way Associative/16 Entries GART cache to Minimize the Number of Memory Bus Cycles Required for Accessing Graphical Texture Memory
- **Fully Compliant to A.G.P. Revision 1.0 Specification**

- **Meets PC98 Requirements**
- **Supports PCI Revision 2.1 Specification**
- **Provides High Performance PCI Arbiter.**
  - Supports up to 4 PCI Masters
  - Supports Rotating Priority Mechanism
  - Hidden Arbitration Scheme Minimizes Arbitration Overhead.
  - Supports Concurrency between CPU to Memory and PCI to PCI.
  - Programmable Timers Ensure Guaranteed Minimum Access Time for PCI Bus Masters, and CPU
- **Integrated Host-to-PCI Bridge**
  - Supports Asynchronous PCI Clock
  - Translates the CPU Cycles into the PCI Bus Cycles
  - Zero Wait State Burst Cycles
  - Supports Pipeline Process in CPU-to-PCI Access
  - Maximum PCI Burst Transfer from 256 Bytes to 4 Kbytes
  - Supports Memory Remapping Function for PCI master accessing Graphical Texture Range
  - Reassembles PCI burst data size into optimized block size for the data snooping in the host bus to minimize interference on CPU
- **Integrated A.G.P. Compliant Target/66Mhz Host-to-PCI Bridge**
  - Supports Synchronous/ Asynchronous A.G.P. Clock
  - Supports 1X, and 2X Mode for A.G.P. 66/133 MHz 3.3V device
  - Translates the CPU Cycles into the A.G.P. Bus (PCI66) Cycles
  - Translates Sequential CPU-to-AGP Memory Write Cycles into A.G.P. Bus (PCI66) Burst Cycles
  - Zero Wait State Burst Cycles
  - Supports Pipeline Process in CPU-to-A.G.P. Access
  - Maximum PCI Burst Transfer from 256 Bytes to 4 Kbytes
  - Reassembles PCI burst data size into optimized block size for the data snooping in the host bus to minimize interference on CPU
  - Supports PCI-to-PCI bridge function for memory write from 33Mhz PCI bus to A.G.P. bus
- **Fast PCI IDE Master/Slave Controller**
  - Supports PCI Bus Mastering
  - Plug and Play Compatible
  - Supports Scatter and Gather
  - Supports Dual Mode Operation - Native Mode and Compatibility Mode
  - Supports IDE PIO Timing Mode 0, 1, 2 ,3 and 4
  - Supports Multiword DMA Mode 0, 1, 2

- Supports Ultra DMA/33
- Two 16 Dword FIFO for PCI Burst Transfers.

- **Integrated Posted Write Data Buffers and Read Prefetch Data Buffers to Increase System Performance**
  - CPU-to-Memory Posted Write Data Buffer with 8 QW Deep, Always Sustains 0 Wait Performance on CPU-to-Memory.
  - CPU-to-Memory Read Data Buffer with 8 QW Deep
  - CPU-to-PCI or CPU-to-AGP multi-purpose Write Data Buffer with 16QW Deep
  - CPU-to-PCI dedicates Posted Write Data Buffer with 4 DW Deep (cascade with multi-purpose Write FIFO)
  - PCI-to-Memory Posted Write Data Buffer with 16 QW Deep, Always Streams 0 Wait performance on PCI-to/from-Memory Access
  - PCI-to-Memory Read Prefetch Data Buffer with (16 QW + 4 DW) Deep
  - CPU-to-A.G.P. dedicates Posted Write Data Buffer with 4 DW Deep (cascade with multi-purpose Write FIFO)
  - CPU-to-Memory Posted Write Data Buffer with 8 QW Deep
  - Request Queue With the Depth of 32
  - High Priority Write Data Queue with 64 QW Deep
  - Low Priority Write Data Queue with 64 QW Deep
  - High Priority Read Data Return Queue with 64 QW Deep
  - Low Priority Read Data Return Queue with 64 QW Deep
- **Concurrent execution between CPU, AGP and PCI Transactions**
  - CPU-to- DRAM read/write and PCI-to-PCI read/write
  - CPU-to-DRAM read/write and PCI-to-DRAM read/write (these two types of transactions are mixed and arranged on host bus, dispatched to DRAM sequentially)
  - CPU-to-DRAM read/write and AGP-to-DRAM read/write (these two types of transactions are mixed and arranged on host bus, dispatched to DRAM sequentially)
  - CPU-to-DRAM read/write and AGP-to-PCI write (or PCI-to-AGP write), (these two types of transactions are mixed and arranged on host bus, dispatched to their destination respectively)
  - CPU-to-PCI write (or AGP-to-PCI write) and PCI-to-DRAM write (CPU-to-PCI, AGP-to-PCI write are temporary queued in multi-purpose FIFO if PCI-to-DRAM burst write is too long)
  - CPU-to-AGP write (or PCI-to-AGP write) and AGP-to-DRAM write (CPU to AGP write, PCI-to-AGP write are temporary queued in multi-purpose FIFO if AGP-to-DRAM burst write is too long)
- **Supports NAND Tree for Ball Connectivity Testing**
- **487-Balls BGA Package**
- **3.3V CMOS Technology**

**FUNCTIONAL BLOCK DIAGRAM**

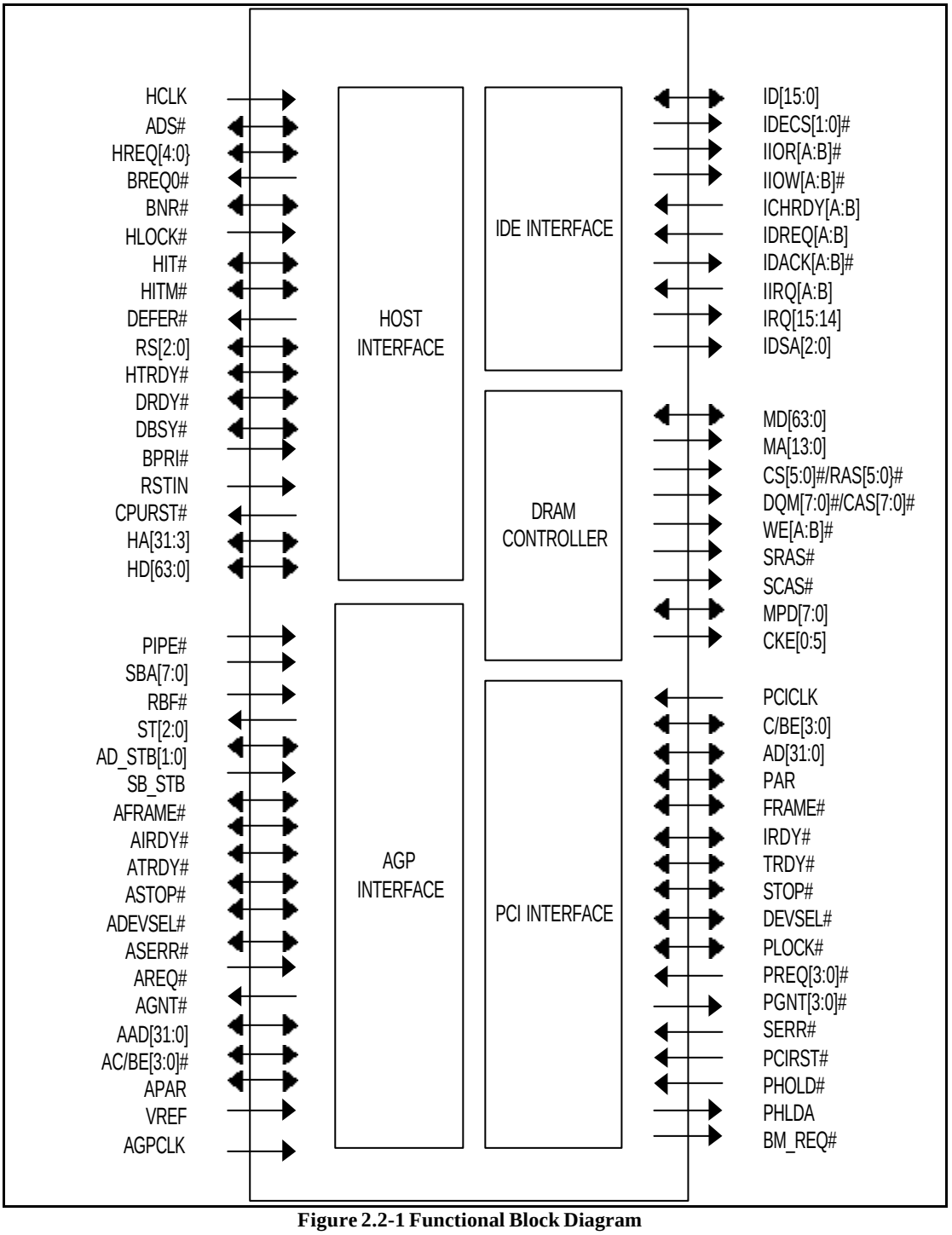



**Figure 2.2-1 Functional Block Diagram**

# 3. PIN ASSIGNMENT

## 3.1. SiS600 PIN ASSIGNMENT (TOP VIEW)

### 3.1.1. SiS600 PIN ASSIGNMENT (TOP VIEW-LEFT SIDE)

| | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ***A*** | | VSS | VSS | 5VDD | AAD4 | AD_STB0 | AAD12 | ASERR# | AIRDY# | VREFVSS | AAD20 | AD_STB1 | AAD28 | ***A*** |
| ***B*** | CKE1 | VSS | VSS | MD32 | AAD2 | AC/BE0# | AAD10 | AC/BE1# | ATRDY# | VREF | AAD18 | AC/BE3# | AAD26 | ***B*** |
| ***C*** | MD0 | CKE2 | VSS | VSS | AAD1 | AAD7 | AAD9 | AAD15 | ADEVSEL# | VREFVCC | AAD17 | AAD23 | AAD25 | ***C*** |
| ***D*** | MD2 | MD1 | MD33 | AAD3 | AAD0 | AAD6 | AAD8 | AAD14 | ASTOP# | AAD16 | AAD19 | AAD22 | AAD24 | ***D*** |
| ***E*** | MD36 | MD3 | MD35 | MD37 | VSS | AAD5 | AAD11 | AAD13 | APAR | AFRAME# | AC/BE2# | AAD21 | AAD27 | ***E*** |
| ***F*** | MD6 | MD38 | MD5 | MD4 | MD34 | | | | | | | | | ***F*** |
| ***G*** | MD41 | MD8 | MD7 | MD39 | MD40 | | | | | | | | | ***G*** |
| ***H*** | MD11 | MD43 | MD10 | MD9 | MD42 | | | | | | | | | ***H*** |
| ***J*** | MD14 | MD13 | MD45 | MD12 | MD44 | | | | | VCC | VCC | VCC | VCC | ***J*** |
| ***K*** | MPD0 | MPD4 | MD15 | MD47 | MD46 | | | | VCC | | | | | ***K*** |
| ***L*** | WEB#/ MA14 | WEA# | SCAS# | MPD1 | MPD5 | | | | VCC | | VSS | VSS | VSS | ***L*** |
| ***M*** | CS5#/ RAS5# | DQM1#/ CAS1# | DQM5#/ CAS5# | DQM4#/ CAS4# | DQM0#/ CAS0# | | | | VCC | | VSS | VSS | VSS | ***M*** |
| ***N*** | CS0#/ RAS0# | CS1#/ RAS1# | CS2#/ RAS2# | CS3#/ RAS3# | CS4#/ RAS4# | | | | VCC | | VSS | VSS | VSS | ***N*** |
| ***P*** | SRAS# | MA0 | MA1 | MA2 | MA4 | | | | VCC | | VSS | VSS | VSS | ***P*** |
| ***R*** | MA3 | MA5 | MA6 | MA7 | MA8 | | | | VCC | | VSS | VSS | VSS | ***R*** |
| ***T*** | MA9 | MA10 | MA11 | MA13 | MA12 | | | | VCC | | VSS | VSS | VSS | ***T*** |
| ***U*** | DQM6#/ CAS6# | DQM2#/ CAS2# | DQM3#/ CAS3# | CKE0 | DQM7#/ CAS7# | | | | VCC | | | | | ***U*** |
| ***V*** | MPD6 | MPD2 | MPD7 | MPD3 | MD16 | | | | VCC | VCC | VCC | VCC | VCC | ***V*** |
| ***W*** | MD48 | MD49 | MD50 | MD18 | MD17 | | | | | | | | | ***W*** |
| ***Y*** | MD51 | MD19 | MD52 | MD20 | MD21 | | | | | | | | | ***Y*** |
| ***AA*** | MD53 | MD54 | MD22 | MD55 | MD23 | | | | | | | | | ***AA*** |
| ***AB*** | MD56 | MD24 | MD57 | MD25 | MD28 | MD63 | IDACKB# | IIOWB | IDREQB | ID2 | ID4 | ID6 | RS1# | ***AB*** |
| ***AC*** | MD58 | MD59 | MD27 | MD26 | IDSA2 | IDECS0# | IIRQB | IIORB | ID15 | ID13 | ID10 | ID7 | RS2# | ***AC*** |
| ***AD*** | MD60 | 5VDD | MD61 | VSS | MD30 | IDSA0 | IDACKA# | IIORA | ID0 | ID12 | ID5 | IRQ15 | DBSY# | ***AD*** |
| ***AE*** | VSS | VSS | CKE4 | MD29 | MD31 | IDSA1 | ICHRDYA | IIOWA | ID14 | ID3 | ID9 | IRQ14 | HIT# | ***AE*** |
| ***AF*** | | CKE3 | CKE5 | MD62 | IDECS1# | IIRQA | ICHRDYB | IDREQA | ID1 | ID11 | ID8 | ADS# | HITM# | ***AF*** |
| | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | *11* | *12* | *13* | |

**Figure 3.1-1 SiS600 Pin Assignment (Top View-Left Side)**

### 3.1.2. SiS600 PIN ASSIGNMENT (TOP VIEW-RIGHT SIDE)

| | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | AAD29 | SBA5 | SBA1 | RBF# | HCLK | AVCC | AD5 | AD9 | AD14 | AGPCLK | AVDD | VSS | | A |
| B | AAD30 | SB_STB | SBA0 | PIPE# | RSTIN | AD1 | AD6 | AD10 | AD15 | VSS | VSS | C/BE1# | 5VDD | B |
| C | AAD31 | SBA4 | ST2 | AGNT# | PCIRST# | AD2 | AD7 | AD11 | VSS | AVSS | PAR | SERR# | PLOCK# | C |
| D | SBA7 | SBA2 | ST1 | AREQ# | AD0 | AD4 | C/BE0# | AD13 | AD12 | C/BE2# | DEVSEL# | TRDY# | IRDY# | D |
| E | SBA6 | SBA3 | ST0 | TEST_PIN# | AVSS | AD3 | AD8 | VSS | STOP# | FRAME# | AD16 | AD17 | AD18 | E |
| F | | | | | | | | | AD19 | AD20 | AD21 | AD22 | AD23 | F |
| G | | | | | | | | | AD24 | C/BE3# | AD25 | AD26 | AD27 | G |
| H | | | | | | | | | AD28 | AD29 | AD30 | AD31 | PCICLK | H |
| J | VCC | VCC | VCC | VCC | VCC | | | | PREQ3# | PREQ2# | PREQ1# | PREQ0# | PGNT3# | J |
| K | | | | | VCC | | | | PGNT2# | PGNT1# | PGNT0# | BM_REQ# | PHLDA# | K |
| L | VSS | VSS | VSS | | VCC | | | | HD62# | PHOLD# | HD58# | HD61# | HD55# | L |
| M | VSS | VSS | VSS | | VCC | | | | HD63# | HD56# | HD60# | HD50# | HD53# | M |
| N | VSS | VSS | VSS | | VCC | | | | HD57# | HD54# | HD59# | HD46# | HD48# | N |
| P | VSS | VSS | VSS | | VCC | | | | HD42# | HD41# | HD51# | HD52# | HD49# | P |
| R | VSS | VSS | VSS | | VCC | | | | HD36# | HD45# | HD39# | HD44# | HD47# | R |
| T | VSS | VSS | VSS | | VCC | | | | VSSREFB | HD43# | HD40# | GTLREFB | VTTB | T |
| U | | | | | VCC | | | | HD33# | HD32# | HD38# | HD37# | HD34# | U |
| V | VCC | VCC | VCC | VCC | VCC | | | | HD29# | HD30# | HD31# | HD28# | HD35# | V |
| W | | | | | | | | | HD24# | HD22# | HD25# | HD27# | HD26# | W |
| Y | | | | | | | | | HD18# | HD16# | HD21# | HD19# | HD23# | Y |
| AA | | | | | | | | | HD7# | HD15# | HD11# | HD17# | HD20# | AA |
| AB | HREQ2# | HREQ0# | HA6# | HA10# | VSSREFA | HA16# | HA15# | HA28# | HD0# | HD13# | HD14# | HD12# | HD10# | AB |
| AC | HLOCK# | HREQ1# | HA9# | HA3# | HA12# | HA18# | HA23# | HA24# | BREQ0# | HA30# | HD8# | HD6# | HD9# | AC |
| AD | DRDY# | HTRDY# | HA4# | HA7# | HA14# | HA17# | HA21# | HA27# | HA29# | VSS | VSS | HD5# | HD4# | AD |
| AE | HREQ3# | HREQ4# | BNR# | HA5# | HA8# | HA11# | HA25# | HA22# | HA26# | HD1# | VSS | VSS | HD2# | AE |
| AF | RS0# | DEFER# | BPRI# | GTLREFA | VTTA | HA13# | HA19# | HA20# | HA31# | CPURST# | HD3# | VSS | | AF |
| | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |

**Figure 3.1-2 SiS600 Pin Assignment (Top View-Right Side)**

### 3.2. ALPHABETICAL PIN LIST

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| 5VDD | AD2 |
| 5VDD | B26 |
| 5VDD | A4 |
| AAD0 | D5 |
| AAD1 | C5 |
| AAD2 | B5 |
| AAD3 | D4 |
| AAD4 | A5 |
| AAD5 | E6 |
| AAD6 | D6 |
| AAD7 | C6 |
| AAD8 | D7 |
| AAD9 | C7 |
| AAD10 | B7 |
| AAD11 | E7 |
| AAD12 | A7 |
| AAD13 | E8 |
| AAD14 | D8 |
| AAD15 | C8 |
| AAD16 | D10 |
| AAD17 | C11 |
| AAD18 | B11 |
| AAD19 | D11 |
| AAD20 | A11 |
| AAD21 | E12 |
| AAD22 | D12 |
| AAD23 | C12 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| AAD25 | C13 |
| AAD26 | B13 |
| AAD27 | E13 |
| AAD28 | A13 |
| AAD29 | A14 |
| AAD30 | B14 |
| AAD31 | C14 |
| AC/BE0# | B6 |
| AC/BE1# | B8 |
| AC/BE2# | E11 |
| AC/BE3# | B12 |
| AD_STB0 | A6 |
| AD_STB1 | A12 |
| AD0 | D18 |
| AD1 | B19 |
| AD2 | C19 |
| AD3 | E19 |
| AD4 | D19 |
| AD5 | A20 |
| AD6 | B20 |
| AD7 | C20 |
| AD8 | E20 |
| AD9 | A21 |
| AD10 | B21 |
| AD11 | C21 |
| AD12 | D22 |
| AD13 | D21 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| AD14 | A22 |
| AD16 | E24 |
| AD17 | E25 |
| AD18 | E26 |
| AD19 | F22 |
| AD20 | F23 |
| AD21 | F24 |
| AD22 | F25 |
| AD23 | F26 |
| AD24 | G22 |
| AD25 | G24 |
| AD26 | G25 |
| AD27 | G26 |
| AD28 | H22 |
| AD29 | H23 |
| AD30 | H24 |
| AD31 | H25 |
| ADEVSEL# | C9 |
| ADS# | AF12 |
| AFRAME# | E10 |
| AGNT# | C17 |
| AGPCLK | A23 |
| AIRDY# | A9 |
| APAR | E9 |
| AREQ# | D17 |
| ASERR# | A8 |
| ASTOP# | D9 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| ATRDY# | B9 |
| AVCC | A19 |
| AVDD | A24 |
| AVSS | C23 |
| AVSS | E18 |
| BM_REQ# | K25 |
| BNR# | AE16 |
| BPRI# | AF16 |
| BREQ0# | AC22 |
| C/BE0# | D20 |
| C/BE1# | B25 |
| C/BE2# | D23 |
| C/BE3# | G23 |
| CKE0 | U4 |
| CKE1 | B1 |
| CKE2 | C2 |
| CKE3 | AF2 |
| CKE4 | AE3 |
| CKE5 | AF3 |
| CPURST# | AF23 |
| CS0#/RAS0# | N1 |
| CS1#/RAS1# | N2 |
| CS2#/RAS2# | N3 |
| CS3#/RAS3# | N4 |
| CS4#/RAS4# | N5 |
| CS5#/RAS5# | M1 |
| DBSY# | AD13 |
| DEFER# | AF15 |
| DEVSEL# | D24 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| DQM0#/CAS0# | M5 |
| DQM1#/CAS1# | M2 |
| DQM2#/CAS2# | U2 |
| DQM3#/CAS3# | U3 |
| DQM4#/CAS4# | M4 |
| DQM5#/CAS5# | M3 |
| DQM7#/CAS7# | U5 |
| DRDY# | AD14 |
| FRAME# | E23 |
| GTLREFA | AF17 |
| GTLREFB | T25 |
| HA3# | AC17 |
| HA4# | AD16 |
| HA5# | AE17 |
| HA6# | AB16 |
| HA7# | AD17 |
| HA8# | AE18 |
| HA9# | AC16 |
| HA10# | AB17 |
| HA11# | AE19 |
| HA12# | AC18 |
| HA13# | AF19 |
| HA14# | AD18 |
| HA15# | AB20 |
| HA16# | AB19 |
| HA17# | AD19 |
| HA18# | AC19 |
| HA19# | AF20 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| HA20# | AF21 |
| HA21# | AD20 |
| HA22# | AE21 |
| HA23# | AC20 |
| HA24# | AC21 |
| HA25# | AE20 |
| HA26# | AE22 |
| HA27# | AD21 |
| HA28# | AB21 |
| HA29# | AD22 |
| HA31# | AF22 |
| HCLK | A18 |
| HD0# | AB22 |
| HD1# | AE23 |
| HD2# | AE26 |
| HD3# | AF24 |
| HD4# | AD26 |
| HD5# | AD25 |
| HD6# | AC25 |
| HD7# | AA22 |
| HD8# | AC24 |
| HD9# | AC26 |
| HD10# | AB26 |
| HD11# | AA24 |
| HD12# | AB25 |
| HD13# | AB23 |
| HD14# | AB24 |
| HD15# | AA23 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| HD16# | Y23 |
| HD17# | AA25 |
| HD18# | Y22 |
| HD19# | Y25 |
| HD20# | AA26 |
| HD21# | Y24 |
| HD22# | W23 |
| HD23# | Y26 |
| HD24# | W22 |
| HD25# | W24 |
| HD26# | W26 |
| HD28# | V25 |
| HD29# | V22 |
| HD30# | V23 |
| HD31# | V24 |
| HD32# | U23 |
| HD33# | U22 |
| HD34# | U26 |
| HD35# | V26 |
| HD36# | R22 |
| HD37# | U25 |
| HD38# | U24 |
| HD39# | R24 |
| HD40# | T24 |
| HD41# | P23 |
| HD42# | P22 |
| HD43# | T23 |
| HD44# | R25 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| HD45# | R23 |
| HD46# | N25 |
| HD47# | R26 |
| HD48# | N26 |
| HD49# | P26 |
| HD50# | M25 |
| HD51# | P24 |
| HD52# | P25 |
| HD53# | M26 |
| HD53# | M26 |
| HD54# | N23 |
| HD55# | L26 |
| HD56# | M23 |
| HD57# | N22 |
| HD58# | L24 |
| HD59# | N24 |
| HD60# | M24 |
| HD61# | L25 |
| HD62# | L22 |
| HD63# | M22 |
| HIT# | AE13 |
| HITM# | AF13 |
| HLOCK# | AC14 |
| HREQ0# | AB15 |
| HREQ1# | AC15 |
| HREQ2# | AB14 |
| HREQ3# | AE14 |
| HREQ4# | AE15 |

| SIGNAL NAME | SiS600 BALL NO |
|---|---|
| HTRDY# | AD15 |
| ICHRDYA | AE7 |
| ICHRDYB | AF7 |
| ID0 | AD9 |
| ID1 | AF9 |
| ID2 | AB10 |
| ID3 | AE10 |
| ID4 | AB11 |
| ID5 | AD11 |
| ID6 | AB12 |
| ID6 | AB12 |
| ID7 | AC12 |
| ID8 | AF11 |
| ID9 | AE11 |
| ID10 | AC11 |
| ID11 | AF10 |
| ID12 | AD10 |
| ID13 | AC10 |
| ID14 | AE9 |
| ID15 | AC9 |
| IDACKA# | AD7 |
| IDACKB# | AB7 |
| IDECS0# | AC6 |
| IDECS1# | AF5 |
| IDREQA | AF8 |
| IDREQB | AB9 |
| IDSA0 | AD6 |
| IDSA1 | AE6 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| IDSA2 | AC5 |
| IIORA | AD8 |
| IIORB | AC8 |
| IIOWA | AE8 |
| IIOWB | AB8 |
| IIRQA | AF6 |
| IIRQB | AC7 |
| IRDY# | D26 |
| IRQ14 | AE12 |
| IRQ15 | AD12 |
| MA0 | P2 |
| MA1 | P3 |
| MA2 | P4 |
| MA3 | R1 |
| MA4 | P5 |
| MA5 | R2 |
| MA6 | R3 |
| MA7 | R4 |
| MA8 | R5 |
| MA9 | T1 |
| MA10 | T2 |
| MA11 | T3 |
| MA12 | T5 |
| MA13 | T4 |
| MD0 | C1 |
| MD1 | D2 |
| MD2 | D1 |
| MD3 | E2 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| MD4 | F4 |
| MD5 | F3 |
| MD6 | F1 |
| MD7 | G3 |
| MD8 | G2 |
| MD9 | H4 |
| MD10 | H3 |
| MD11 | H1 |
| MD12 | J4 |
| MD13 | J2 |
| MD14 | J1 |
| MD15 | K3 |
| MD17 | W5 |
| MD18 | W4 |
| MD19 | Y2 |
| MD20 | Y4 |
| MD21 | Y5 |
| MD22 | AA3 |
| MD23 | AA5 |
| MD24 | AB2 |
| MD25 | AB4 |
| MD26 | AC4 |
| MD27 | AC3 |
| MD28 | AB5 |
| MD29 | AE4 |
| MD30 | AD5 |
| MD31 | AE5 |
| MD32 | B4 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| MD33 | D3 |
| MD34 | F5 |
| MD35 | E3 |
| MD36 | E1 |
| MD37 | E4 |
| MD38 | F2 |
| MD39 | G4 |
| MD40 | G5 |
| MD41 | G1 |
| MD42 | H5 |
| MD43 | H2 |
| MD44 | J5 |
| MD45 | J3 |
| MD45 | J3 |
| MD46 | K5 |
| MD47 | K4 |
| MD48 | W1 |
| MD49 | W2 |
| MD50 | W3 |
| MD51 | Y1 |
| MD52 | Y3 |
| MD53 | AA1 |
| MD54 | AA2 |
| MD55 | AA4 |
| MD56 | AB1 |
| MD57 | AB3 |
| MD58 | AC1 |
| MD59 | AC2 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| MD60 | AD1 |
| MD61 | AD3 |
| MD62 | AF4 |
| MD63 | AB6 |
| MPD0 | K1 |
| MPD1 | L4 |
| MPD2 | V2 |
| MPD3 | V4 |
| MPD4 | K2 |
| MPD5 | L5 |
| MPD6 | V1 |
| MPD7 | V3 |
| PAR | C24 |
| PCICLK | H26 |
| PCIRST# | C18 |
| PGNT0# | K24 |
| PGNT1# | K23 |
| PGNT2# | K22 |
| PGNT3# | J26 |
| PHLDA# | K26 |
| PHOLD# | L23 |
| PIPE# | B17 |
| PLOCK# | C26 |
| PREQ0# | J25 |
| PREQ1# | J24 |
| PREQ2# | J23 |
| PREQ3# | J22 |
| RBF# | A17 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| RS0# | AF14 |
| RS1# | AB13 |
| RS2# | AC13 |
| RSTIN | B18 |
| SB_STB | B15 |
| SBA0 | B16 |
| SBA1 | A16 |
| SBA2 | D15 |
| SBA3 | E15 |
| SBA4 | C15 |
| SBA5 | A15 |
| SBA6 | E14 |
| SBA7 | D14 |
| SCAS# | L3 |
| SERR# | C25 |
| SRAS# | P1 |
| ST0 | E16 |
| ST1 | D16 |
| ST2 | C16 |
| STOP# | E22 |
| TEST_PIN# | E17 |
| TRDY# | D25 |
| VCC | J10 |
| VCC | J11 |
| VCC | J12 |
| VCC | J13 |
| VCC | J14 |
| VCC | J15 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| VCC | J16 |
| VCC | J17 |
| VCC | J18 |
| VCC | V9 |
| VCC | V10 |
| VCC | V11 |
| VCC | V12 |
| VCC | V13 |
| VCC | V14 |
| VCC | V15 |
| VCC | V16 |
| VCC | V17 |
| VCC | V18 |
| VCC | K9 |
| VCC | K18 |
| VCC | L9 |
| VCC | L18 |
| VCC | M9 |
| VCC | M18 |
| VCC | M18 |
| VCC | N9 |
| VCC | N18 |
| VCC | P9 |
| VCC | P18 |
| VCC | R9 |
| VCC | R18 |
| VCC | T9 |
| VCC | T18 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| VCC | U9 |
| VCC | U18 |
| VREF | B10 |
| VREFVCC | C10 |
| VREFVSS | A10 |
| VSS | AE1 |
| VSS | AE2 |
| VSS | AD4 |
| VSS | AE24 |
| VSS | AD24 |
| VSS | AF25 |
| VSS | AE25 |
| VSS | A25 |
| VSS | B24 |
| VSS | B23 |
| VSS | E21 |
| VSS | C22 |
| VSS | C4 |
| VSS | E5 |
| VSS | A3 |
| VSS | B3 |
| VSS | C3 |
| VSS | A2 |
| VSS | B2 |
| VSS | L11 |
| VSS | L12 |
| VSS | L13 |
| VSS | L14 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| VSS | L15 |
| VSS | L16 |
| VSS | M11 |
| VSS | M12 |
| VSS | M13 |
| VSS | M14 |
| VSS | M15 |
| VSS | M16 |
| VSS | N11 |
| VSS | N12 |
| VSS | N13 |
| VSS | N14 |
| VSS | N15 |
| VSS | N15 |
| VSS | N16 |
| VSS | P11 |
| VSS | P12 |
| VSS | P13 |
| VSS | P14 |
| VSS | P15 |
| VSS | P16 |
| VSS | R11 |
| VSS | R12 |
| VSS | R13 |
| VSS | R14 |
| VSS | R15 |
| VSS | R16 |
| VSS | T11 |

| SIGNAL NAME | SiS600 BALL NO. |
|---|---|
| VSS | T12 |
| VSS | T13 |
| VSS | T14 |
| VSS | T15 |
| VSS | T16 |
| VSS | AD23 |
| VSSREFA | AB18 |
| VSSREFB | T22 |
| VTTA | AF18 |
| VTTB | T26 |
| WEA# | L2 |
| WEB#/MA14 | L1 |

# 4. PIN ASSIGNMENT

## 4.1. SiS600 PIN DESCRIPTION

### 4.1.1. HOST BUS INTERFACE

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| HCLK | I | **Host Clock :** Primary clock input drives the part. |
| ADS# | I/O GTL+ | **Address Status :** Address Status is driven by the CPU to indicate the start of a CPU bus cycle. |
| HREQ[4:0]# | I/O GTL+ | **Request Command:** HREQ[4:0]# are used to define each transaction type during the clock when ADS# is asserted and the clock after ADS# is asserted. |
| BREQ0# | O GTL+ | **Symmetric Agent Bus Request:** BREQ0# is driven by the symmetric agent that wants to request the bus. |
| BNR# | I/O GTL+ | **Block Next Request:** When system is busy, the bus agent can block further transaction by asserting BNR#. |
| HLOCK# | I GTL+ | **Host Lock :** When CPU asserts HLOCK# to indicate the current bus cycle is locked. |
| HIT# | I/O GTL+ | **Keeping a Non-Modified Cache Line:** |
| HITM# | I/O GTL+ | **Hits a Modified Cache Line:** Hit Modified indicates the snoop cycle hits a modified line in the L1 cache of CPU. |
| DEFER# | O GTL+ | **Defer Transaction Completion:** SiS600 will use this signal to indicate a retry response to the host bus. |

| | | |
|---|---|---|
| RS[2:0]# | O<br>GTL+ | **Response Status:**<br>RS[2:0]# are driven by the response agent to indicate the transaction response type. The following shows the response type.<br>**RS[2:0] Response Type RS[2:0] Response Type**<br>000 Idle State 100 Reserved<br>001 Retry 101 No data<br>010 Reserved 110 Implicit Write-back<br>011 Reserved 111 Normal Data |
| HTRDY# | I/O<br>GTL+ | **Target Retry:**<br>During write cycles, response agent will drive TRDY# to indicate the agent is ready to accept data. |
| DRDY# | I/O<br>GTL+ | **Data Ready:**<br>DRDY# is driven by the bus owner whenever the data is valid on the bus. |
| DBSY# | I/O<br>GTL+ | **Data Bus Busy:**<br>Whenever the data is not valid on the bus with DRDY# is deserted, DBSY# is asserted to hold the bus. |
| BPRI# | O<br>GTL+ | **Priority Agent Bus Request:**<br>BPRI# is driven by the priority agent that wants to request the bus.<br>BPRI# has higher priority than BREQ0# to access a bus. |
| RSTIN | I | **Reset:**<br>RSTIN is the reset pin generated from SiS5595, which is used to keep the host controller of SiS600 in the initial state, and also used to generate CPURST# to CPU. |
| CPURST# | O<br>GTL+ | **Host Bus Reset:**<br>CPURST# is used to keep all the bus agents in the same initial state before valid cycles issued. |
| HA[31:3] | I/O<br>GTL+ | **Host Address Bus :**<br>The CPU Address is driven by the CPU during CPU bus cycles. SiS600 forwards it to either the DRAM or the PCI bus depending on the address range. The address bus is driven by SiS600 during bus master cycles. |
| HD[63:0]# | I/O<br>GTL+ | **Host Data Bus :**<br>HD[63:0] contain 64-bit data between all bus agents. |

### 4.1.2. DRAM CONTROLLER

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| MD[63:0] | I/O | **Memory Data Bus :**<br>MD[63:0] contain 64-bit data between SiS600 and memory. |
| MA[13:0] | O | **Memory Address Lines 14-2:**<br>Memory address 14-2 are the row and column addresses for DRAM. |
| CS[5:0]#/RAS[5:0]# | O | **Chip Select (SDRAM)/ Row Address Strobe (EDO/FP):**<br>**Row Address Strobe (EDO/FP):**<br>DRAM Row address strobe 5-0 for DRAM banks 2-0.<br>**Chip Select (SDRAM) :**<br>These pins activate the SDRAM and accept any command when it is low. |
| DQM[7:0]#/<br>CAS[7:0]# | O | **Pin Mask/Output Enable (SDRAM)/Column address strobe (FPM/EDO) :**<br>**Column address strobe (FPM/EDO) :**<br>DRAM Column address strobe 7-0 for byte 7-0.<br>**Input / Output Data Mask (SDRAM) :**<br>SDRAM output enables during a read cycle and a byte mask during a write cycle. |
| WEA# | O | **Memory Write A:**<br>RAM Write is an active low output signal to enable local DRAM writes.<br>Two copies are provided for loading purposes. |
| WEB#/MA14 | O | **Memory Write B / Memory Address line 14 :**<br>RAM Write is an active low output signal to enable local DRAM writes.<br>Two copies are provided for loading purposes.<br>This pin also can serve as MA14 signal by programming the Register 52h bit4 to 1 to support the high density DRAM parts. |
| SRAS# | O | **SDRAM Row Address Strobe :**<br>It latches row address on the positive edge of the clock with SRAS# low. These signals enable row access and precharge. |

| | | |
|---|---|---|
| SCAS# | O | **SDRAM Column Address Strobe :**<br><br>SDRAM latches column address on the positive edge of the clock with SCAS# low. It is driven to low by SiS600 when column access. |
| MPD[7:0] | I/O | **Memory ECC Data Bus :**<br><br>These signals carry memory ECC data during read/write to DRAM. |
| CKE[0:5] | O | **SDRAM Clock Enable 0~5:**<br><br>While in ACPI S2 or S3 state, SiS600 can put the SDRAM in the self-refresh mode by CKE[0:5] signals. During power down mode, CKE[0:5] signals of DIMM module must be kept low by SiS5595's CKES signal. |

### 4.1.3. PCI INTERFACE

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| PCICLK | I | **PCI Clock :**<br><br>The PCICLK input provides the fundamental timing and the internal operating frequency for SiS600. It runs at the same frequency and skew of the PCI local bus. |
| C/BE[3:0]# | I/O | **PCI Bus Command and Byte Enables :**<br><br>PCI Bus Command and Byte Enables define the PCI command during the address phase of a PCI cycle, and the PCI byte enables during the data phases. C/BE[3:0]# are outputs when SiS600 is a PCI bus master and inputs when it is a PCI slave. |
| AD[31:0] | I/O | **PCI Address /Data Bus :**<br><br>In address phase:<br><br>1.When SiS600 is a PCI bus master, AD[31:0] are output signals.<br><br>When SiS600 is a PCI target, AD[31:0] are input signals.<br><br>In data phase:<br><br>1. When SiS600 is a target of a memory read/write cycle, AD[31:0] are floating.<br><br>2. When SiS600 is a target of a configuration or an I/O cycle, AD[31:0] are output signals in a read cycle, and input signals in a write cycle. |
| PAR | I/O | **Parity :**<br><br>Parity is an even parity which is generated across AD[31:0] and C/BE[3:0]#. |

| | | |
|---|---|---|
| FRAME# | I/O | **Frame :**<br><br>FRAME# is an output when SiS600 is a PCI bus master. SiS600 drives FRAME# to indicate the beginning and duration of an access. When SiS600 is a PCI slave, FRAME# is an input signal. |
| IRDY# | I/O | **Initiator Ready :**<br><br>IRDY# is an output when SiS600 is a PCI bus master. The assertion of IRDY# indicates the current PCI bus master's ability to complete the current data phase of the transaction. For a read cycle, IRDY# indicates that the PCI bus master is prepared to accept the read data on the following rising edge of the PCI clock. For a write cycle, IRDY# indicates that the bus master has driven valid data on the PCI bus. When SiS600 is a PCI slave, IRDY# is an input. |
| TRDY# | I/O | **Target Ready :**<br><br>TRDY# is an output when SiS600 is a PCI slave. The assertion of TRDY# indicates the target agent's ability to complete the current data phase of the transaction. For a read cycle, TRDY# indicates that the target has driven valid data onto the PCI bus. For a write cycle, TRDY# indicates that the target is prepared to accept data from the PCI bus. When SiS600 is a PCI master, it is an input signal. |
| STOP# | I/O | **Stop :**<br><br>STOP# indicates that the bus master must start terminating its current PCI bus cycle at the next clock edge and release control of the PCI bus. STOP# is used for disconnection, retry, and target-abortion sequences on the PCI bus. |
| DEVSEL# | I/O | **Device Select :**<br><br>As a PCI target, SiS600 asserts DEVSEL# by doing positive or subtractive decoding. SiS600 positively asserts DEVSEL# when the DRAM address is being accessed by a PCI master, PCI configuration registers or embedded controllers' registers are being addressed, or the BIOS memory space is being accessed. The low 16K I/O space and low 16M memory space are responded subtractively. The DEVESEL# is an input when SiS600 is acting as a PCI master. It is asserted by the addressed agent to claim the current transaction. |
| PLOCK# | I/O | **PCI Lock :**<br><br>PCI Lock indicates an exclusive bus operation that may require multiple transactions to complete. When PLOCK# is sampled asserted at the beginning of a PCI cycle, SiS600 considers itself a locked resource and remains in the locked state until PLOCK# is sampled negated on a new PCI cycle. |

| | | |
|---|---|---|
| PREQ[3:0]# | I | **PCI Bus Request :**<br>PCI Bus Request is used to indicate the PCI bus arbiter that an agent requires the use of PCI bus. |
| PGNT[3:0]# | O | **PCI Bus Grant :**<br>PCI Bus Grant signal indicates an agent that access to the PCI bus has been granted. |
| SERR# | I | **System Error :**<br>SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, SiS600 generates a non-maskable interrupt to the CPU. |
| PCIRST# | I | **PCI Bus Reset :**<br>The PCIRST# is used to reset the device on PCI bus. PCIRST# is driven low when PWRGD is sampled low and driven inactive about l ms after PWRGD is sampled high. |
| PHOLD# | I | **Bus Hold :**<br>PHOLD# is used to request the use of PCI bus. PHOLD# is asserted on behalf of the ISA master, DMA devices, or USB devices. PHOLD# is eventually connected to the PCI system arbiter (normally located in SiS600). |
| PHLDA# | O | **Bus Hold Acknowledge :**<br>The PCI system arbiter asserts this signal to acknowledge the grant of PCI bus access to the SiS5595. |
| BM_REQ# | O | **Bus Master Request Status :**<br>It will be used to carry the following two information :<br>1) AGP activity event to reload the system standby timer in the SiS5595, and<br>2) AGP/PCI/IDE bus master request event to exit from ACPI/C3 state. Upon power up, the clock after FRAME# is sampled asserted is defined as the slot containing the Bus master request event, the next clock containing the AGP activity event, ...etc. |

### 4.1.4. PCI IDE INTERFACE

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| ID[15:0] | I/O | **IDE Data Bus :**<br>These data signals are used to transfer data to/from the IDE device. |
| IDECS[1:0]# | O | **IDE Channel 0 Chip select signals :** |

| | | |
|---|---|---|
| IIOR[A:B] | O | **IDE Channel 0/1 I/O Read Cycle Command :** |
| IIOW[A:B] | O | **IDE Channel 0/1 I/O Write Cycle Command :** |
| ICHRDY[A:B] | I | **IDE Channel 0/1 I/O Channel Ready Signal :** |
| IDREQ[A:B] | I | **IDE Channel 0/1 DMA Request Signals :** |
| IDACK[A:B]# | O | **IDE Channel 0/1 DMA Acknowledge Signals :** |
| IIRQ[A:B] | I | **IDE Channel 0/1 Interrupt Request Signals :**<br>These are the synchronous interrupt request inputs from IDE device. |
| IRQ[15:14] | O | **IDE channel 1/0 Interrupt Request Signals :**<br>These are the synchronous interrupt request output to the SiS5595 internal 8259 controller. |
| IDSA[2:0] | O | **IDE Address [2:0] :** |

### 4.1.5. AGP INTERFACE

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| AGPCLK | I | **A.G.P. Clock :**<br>The AGPCLK input provides the fundamental timing and the internal operating frequency for SiS600. It runs at the same frequency and skew of the A.G.P. bus. |
| PIPE# | I | **Pipeline Operation :**<br>The AGP master asserts PIPE# to inform the target to enqueue a full width request. The master always enqueues one request on each rising edge of AGPCLK while PIPE# is asserted. |
| SBA[7:0] | I | **Sideband Address Port :**<br>These signals provide an optional path for the AGP master to pass the address and command to the target. |
| RBF# | I | **Read Buffer Full :**<br>AGP bus master asserts the signal to tell the AGP local arbiter not to initiate the return of any low priority read data since the AGP master is not affordable to buffer it. |

| | | |
|---|---|---|
| ST[2:0] | O | **Status Bus :**<br>Status bus passes the information to the AGP master what SiS600 may do.<br>**ST[2:0] Description**<br>000 Indicates that previously requested low priority read data is being return to the master.<br>001 Indicates that previously requested high priority read data is being return to the master.<br>010 Indicates that the master is to provide low priority write data for a previous enqueued write command.<br>011 Indicates that the master is to provide high priority write data for a previous enqueued write command.<br>111 Indicates that the master has been given permission to start a bus transaction. |
| AD_STB[1:0] | I/O | **AD bus Strobe 1 :**<br>AD bus Strobe 1 provides timing for 2x transfer mode on the AD[31:16]. SiS600 drives this signal while it is providing data. |
| SB_STB | I | **Side Band Strobe :**<br>Side Band Strobe provides timing for SiS600 to strobe SBA[7:0]. |
| AFRAME# | I/O | **Frame (Same as PCI) :**<br>AFRAME# remains deserted by the external pull up resistor in the AGP operation., bus behaves the same as in the PCI bus during PCI operations on the AGP bus. |
| AIRDY# | I/O | **AGP IRDY# :**<br>Assertion of AIRDY# indicates the AGP master is ready to provide all write data for the current transaction. While in the AGP read cycle, it indicates that the master is ready to transfer a subsequent block of data. |
| ATRDY# | I/O | **AGP TRDY# :**<br>Assertion of ATRDY# indicates that SiS600 is ready to provide read data for the entire transaction or is ready to transfer a block of data. |
| ASTOP# | I/O | **Stop (Same as PCI) :**<br>ASTOP# is only used during PCI operation on the AGP bus. |
| ADEVSEL# | I/O | **Device Select (Same as PCI) :**<br>ADEVSEL# is only used during PCI operation on the AGP bus. |

| | | |
|---|---|---|
| ASERR# | I/O | **AGP SERR# :**<br>It is provided for the AGP master to report any error. |
| AREQ# | I | **Request :**<br>AGP master asserts this signal to request the access of the AGP bus. |
| AGNT# | O | **Grant :**<br>SiS600 grants the AGP bus authority to the AGP bus master to initiate a bus transaction. Together with ST[2:0], the assertion of GNT# can also indicates that the master is the recipient of the previously requested read data, or that the master can provide write data for a previously enqueued write command. |
| AAD[31:0] | I/O | **AGP/PCI Address/Data Bus :** |
| AC/BE[3:0]# | I/O | **AGP/PCI Command/Bye Enables Bus :** |
| APAR | I/O | **Parity :**<br>APAR is only used during PCI operation on the AGP bus. |
| VREF | I | **AGP Reference Voltage :**<br>This pin should be guarded by VREFVDD/ VREFVSS on the motherboard layout. |
| VREFVDD | PWR | **3.3V DC power signal for AGP VREF.** |
| VREFVSS | PWR | **Ground signal for AGP VREF.** |

#### 4.1.6. POWER PINS

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|
| VCC | PWR | **+3.3V DC Power Source.** |
| VSS | PWR | **Ground signals for core logic.** |
| 5VCC | PWR | **5V DC Power Source.** |
| AVDD, AVCC, AVSS | PWR | **Phase Lock Loop circuit Power and Ground.** |
| GTLREF[A:B] | PWR | **Reference Voltage for GTL+ interface.** |
| VSSREF[A:B] | PWR | **Ground signals for reference voltages.** |
| VTT[A:B] | PWR | **Termination to Termination Voltage.** |

#### 4.1.7. MISC. PINS

| NAME | TYPE ATTR | DESCRIPTION |
|---|---|---|

| | | |
|---|---|---|
| TEST_PIN# | I | **Test Mode Select for NAND Tree function.**<br>Ball connectivity test mode<br>Pull-up : Disable<br>Pull-down : Enable |

# 5. FUNCTIONAL DESCRIPTION

## 5.1. HOST INTERFACE

### 5.1.1. HOST INTERFACE BLOCK DIAGRAM

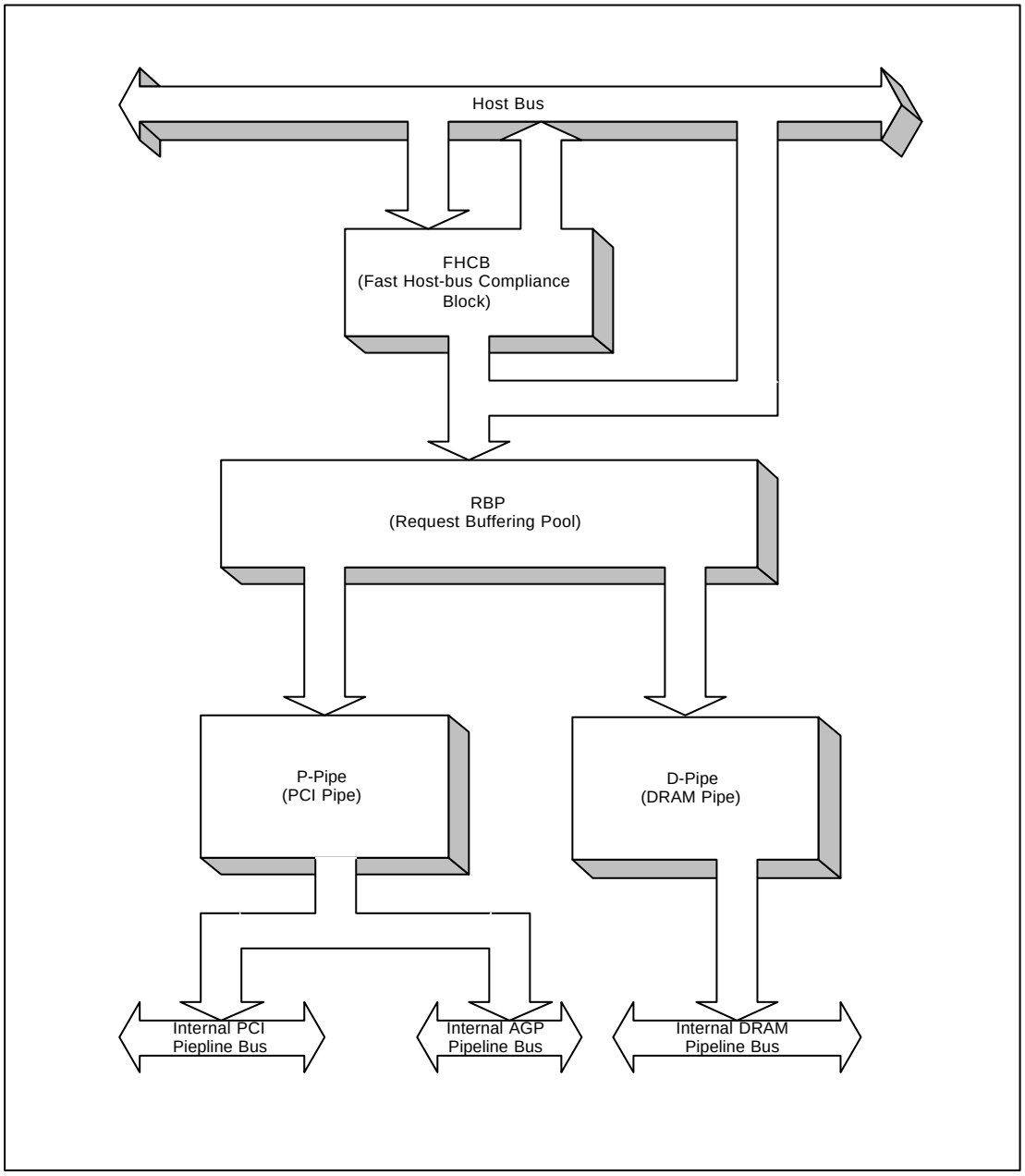



**Figure 5.1-1 Block Diagram for Host Interface**

The SiS600 is designed to support Pentium Pro/Pentium II CPU with a 100/66/60 MHz pipeline bus. The host interface is comprised of a “Fast Host-bus Compliant Block”(FHCB), a “Request Buffering Pool”(RBP), a “DRAM Pipe”(D-Pipe) and a “PCI Pipe”(P-Pipe). The FHCB is mainly designed for the hand shaking with CPU and the compliance of the Pentium II host bus protocol. This block can assert the control signals at the fastest time such that host bus can be pipelined to the most efficient stage. The RBP is capable of buffering eight consecutive requests, either from CPU or from SiS600 itself. There are two pipes constructed in SiS600 in such a way that requests from host bus can be serviced concurrently. The first pipe that can call it “DRAM Pipe.” This pipe is used to pipeline all requests toward onboard DRAM. These requests are arranged close enough that all DRAM bandwidth are exploited without idle state. The other pipe is named as “PCI Pipe.” This pipe is designed to service requests toward PCI bus or toward AGP bus. As cycle is executed in “DRAM pipe”, “PCI pipe” is also optimized to reduce all zero wait state and concurrent execution between CPU-PCI cycles and CPU-AGP cycles.

## 5.2. DRAM CONTROLLER

The SiS600 can support up to 1.5GBytes of DRAM and each bank could be single or double sided 72-bit (64-bit data and 8-bit ECC code) Fast Page Mode (FPM) DRAM, Extended Data Output (EDO) DRAM, and Synchronous DRAM (SDRAM). The SiS600 supports industry standard SIMM/DIMM modules. Six RAS#/CS# lines permit up to six rows (3 double sided banks) of DRAM, and mixing the different type of DRAM bank by bank is acceptable.

The installed EDO/FPM DRAM type can be 256K, 512k, 1M, 2M, 4M, 8M or 16M bit deep by n (n = 4, 8, 16,or 32) bit wide, and both symmetrical and asymmetrical addressing modes are supported. The installed SDRAM type can be 1M, 2M, 4M, 8M, 16M, 32M bit deep by n (n = 4, 8, 16,or 32) bit wide. The SiS600 DRAM Controller operates synchronously to the CPU clock.

### 5.2.1. DRAM CONFIGURATION

SiS600 supports three banks (double sided DRAM) of DRAM each 64/72-bit wide. The three banks may be configured in three banks of EDO/FPM SIMM, three banks of EDO/SDRAM DIMM or any other combinations as required. Access to the banks are not interleaved and need not to be populated starting from row 0 or in consecutive sequence.

The SiS600 can support EDO, FPM and SDRAM. These different types of DRAM can be mixed for each bank, it must contain only one type of DRAM in each bank.

The basic configurations are shown as the following sections:

#### 5.2.1.2. EDO/FPM DRAM CONFIGURATION (4 SIMM/6 SIMM):

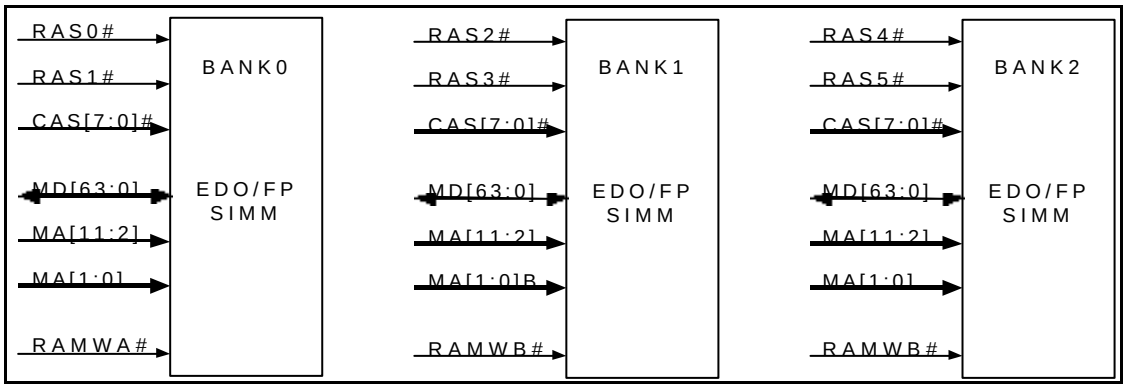



**Figure 5.2-1 EDO/FPM DRAM Configuration**

#### 5.2.1.3. SDRAM CONFIGURATION (2 DIMM/3 DIMM):

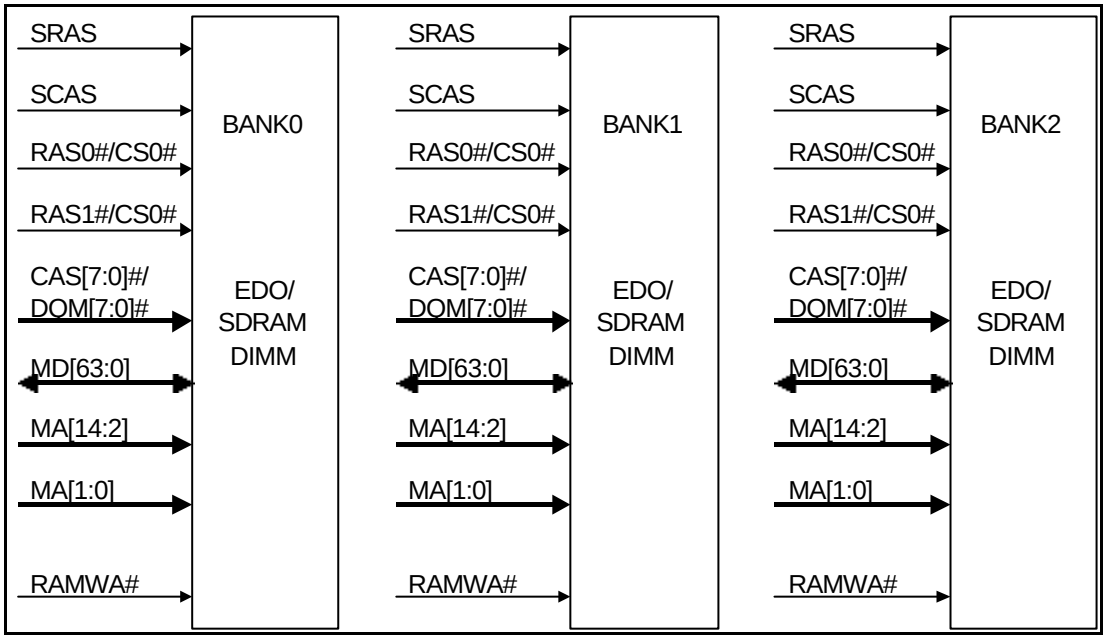



**Figure 5.2-2 SDRAM Configuration**

#### 5.2.1.4. DRAM TYPE MIXED CONFIGURATION: EDO/FPM + SDRAM (4 SIMM + 2 DIMM)

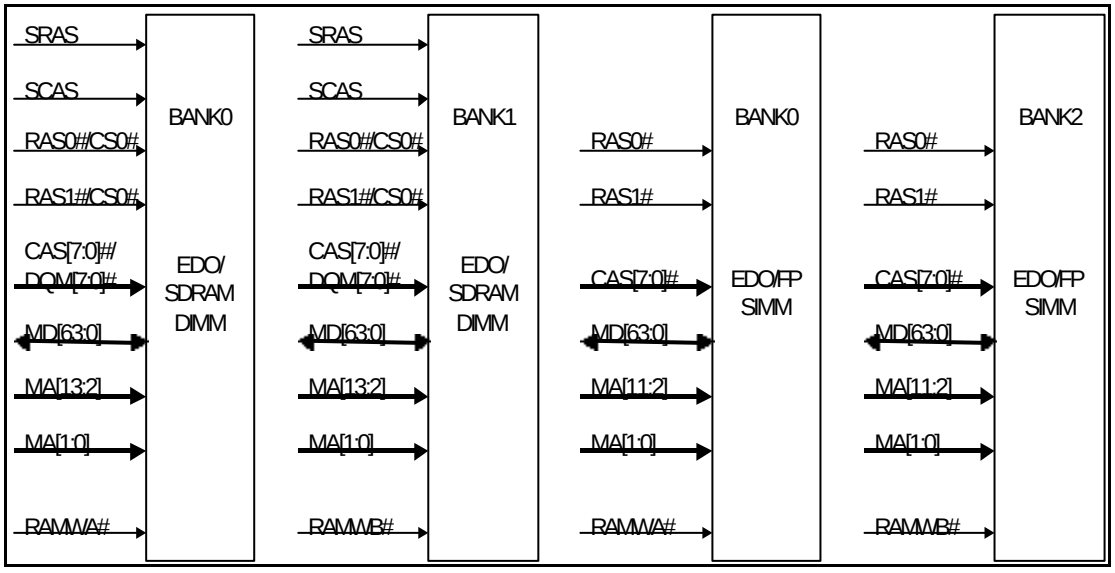



**Figure 5.2-3 Mixed DRAM Configuration**

**Note:**

1. SiS600 only supports six rows (3 banks) DRAM.
2. It is recommended that board designer must follow DC characteristics of each type DRAM (SDRAM, EDO, FPM) to design the portion of DRAM in DRAM mode mixed configuration.

### 5.2.2. DRAM SCRAMBLE TABLE

The DRAM scramble table contains information for memory address mapping. These tables provide the translation between CPU host address and memory Row and Column address.

There are several memory address mappings: MA mapping for FPM/EDO DRAM, 2Bank and 4Bank mapping for SDRAM that SiS600 supports:

#### 5.2.2.1. MA MAPPING TABLE FOR FPM/EDO DRAM

a. **Symmetric:**

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

**b. Asymmetric:**

| **Type** | **512K (10x9)** | | **1M (11x9)** | | **2M (11x10)** | |
|---|---|---|---|---|---|---|
| Address | Row | Column | Row | Column | Row | Column |
| MA0 | 15 | 3 | 15 | 3 | 15 | 3 |
| MA1 | 16 | 4 | 16 | 4 | 16 | 4 |
| MA2 | 17 | 5 | 17 | 5 | 17 | 5 |
| MA3 | 18 | 6 | 18 | 6 | 18 | 6 |
| MA4 | 19 | 7 | 19 | 7 | 19 | 7 |

| | | | | | | |
|---|---|---|---|---|---|---|
| MA5 | 20 | 8 | 20 | 8 | 20 | 8 |
| MA6 | 21 | 9 | 21 | 9 | 21 | 9 |
| MA7 | 13 | 10 | 22 | 10 | 22 | 10 |
| MA8 | 14 | 11 | 14 | 11 | 23 | 11 |
| MA9 | 12 | NA | 12 | NA | 13 | 12 |
| MA10 | NA | NA | 13 | NA | 14 | NA |
| MA11 | NA | NA | NA | NA | NA | NA |

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

NOTE : “X” MEANS DO NOT CARE.

#### 5.2.2.2. MA MAPPING TABLE FOR SDRAM

**a. 2 Banks Device SDRAM Type:**

| **Type** | **1M (1x11x8)** | | **2M (1x11x9)** | | **4M (1x11x10)** | |
|---|---|---|---|---|---|---|
| Address | Row | Column | Row | Column | Row | Column |
| MA0 | 15 | 3 | 15 | 3 | 15 | 3 |
| MA1 | 16 | 4 | 16 | 4 | 16 | 4 |
| MA2 | 17 | 5 | 17 | 5 | 17 | 5 |
| MA3 | 18 | 6 | 18 | 6 | 18 | 6 |

| | | | | | | |
|---|---|---|---|---|---|---|
| MA4 | 19 | 7 | 19 | 7 | 19 | 7 |
| MA5 | 20 | 8 | 20 | 8 | 20 | 8 |
| MA6 | 21 | 9 | 21 | 9 | 21 | 9 |
| MA7 | 22 | 10 | 22 | 10 | 22 | 10 |
| MA8 | 12 | NA | 23 | 11 | 23 | 11 |
| MA9 | 13 | NA | 13 | NA | 24 | 12 |
| MA10 | 14 | NA | 14 | NA | 14 | NA |
| MA11 | 11 | 11 | 12 | 12 | 13 | 13 |
| MA12 | NA | NA | NA | NA | NA | NA |
| MA13 | NA | NA | NA | NA | NA | NA |
| MA14 | NA | NA | NA | NA | NA | NA |

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

**b. 4 banks Device SDRAM Type:**

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

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

| | | | | | | |
|---|---|---|---|---|---|---|
| MA12 | 12 | 12 | 13 | 13 | 14 | 14 |
| MA13 | 24 | NA | 25 | NA | 26 | NA |
| MA14 | 25 | NA | 26 | NA | 27 | NA |

### 5.2.3. DRAM AUTO-DETECTION

SiS600 supports three banks DRAM for SIMM/DIMM from bank0 to bank2. The DRAM detection sequence is a bank-based detection sequence, it is performed by the BIOS bank by bank and fulfilled the DRAM configuration information into the corresponding DRAM configuration registers. The following steps will be described the DRAM detection sequence.

Step 1. To detect if there is any DRAM populated in bank N, SiS600 sets this bank with maximum DRAM size, then writes/reads the same address with test pattern by the normal DRAM read/write timing and compares the data. If the read data is the same as the write pattern, then there are exists DRAM in the bank N; otherwise, proceeds the SDRAM detection from step 3.

Step 2. If the DRAM is detected in the bank N by step 1, SiS600 treats it as EDO or FPM DRAM. SiS600 first writes test pattern into DRAM, then sets register 55h bit 6 (EDO Detection Bit) to be "1" in HOST-to-PCI bridge configuration space, then reads the same DRAM location and compares with the test pattern. The EDO detection bit will delay the data forward to CPU after 4096 CPU clock. If the CPU still get the right data, then EDO mode DRAM is set to this row; otherwise, the FP mode DRAM is set. Go to step 8.

Step 3. If the DRAM is detected not populated in bank N by normal write/read procedure, SiS600 checks if there is SDRAM existing in this bank or not. SiS600 first assumes the DRAM mode is SDRAM (set bit [7:6] of register 60h/61h/62h to be "11" in HOST-to-PCI bridge configuration space, it depends on which bank is under detection), and then does the SDRAM initialization procedure from step 4 to step 6.

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

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

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

Step 7. Write/Read the test pattern into SDRAM, then compare the data. If the data is correct, SDRAM is detected, and set bank N as SDRAM; otherwise, bank N is no DRAM populated.

Step 8. After DRAM mode is set, SiS600 does DRAM sizing by write/read test pattern based on the MA mapping table.

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

**Note:** The value of N is from 0 to 2.

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

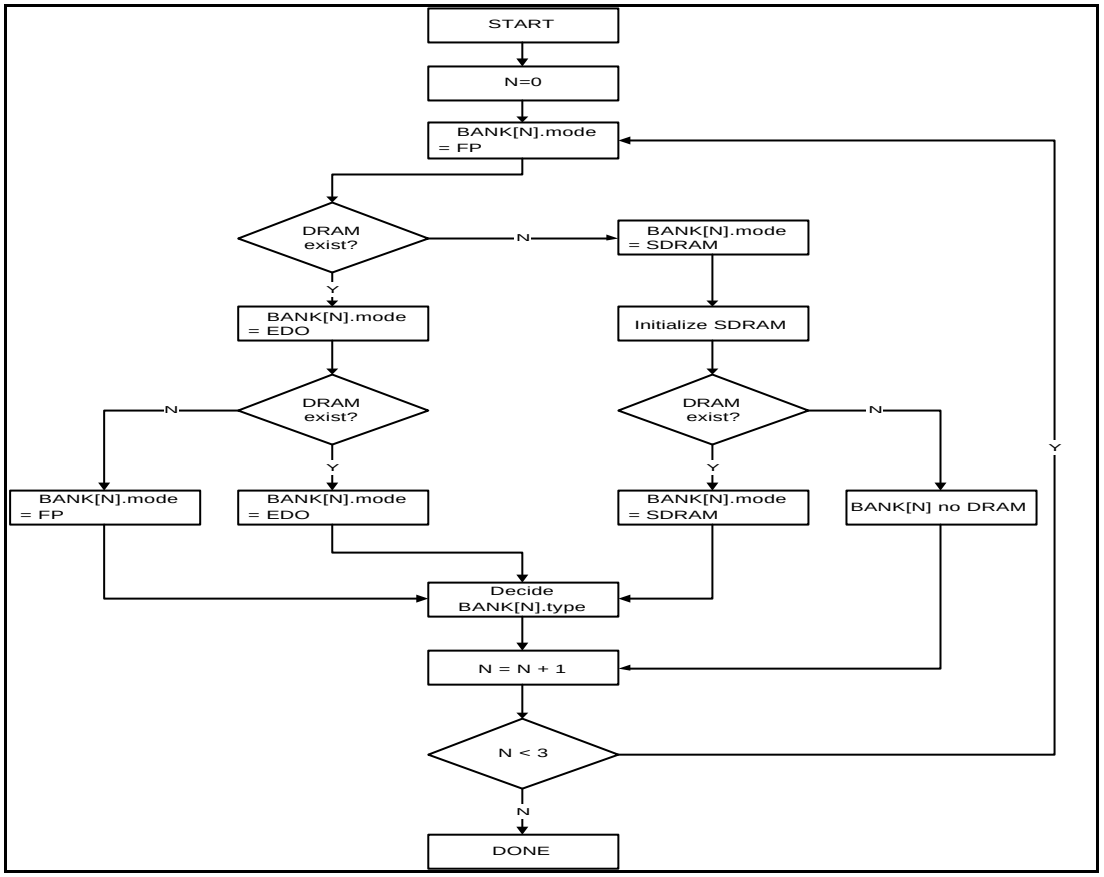



**Figure 5.2-4 DRAM Detection Sequence**

### 5.2.4. ARBITER

The arbiter is the interface between the DRAM controller and the host interface which can access DRAMs. In addition to passing or translating the information from outside to DRAM controller, arbiter is also responsible for which master has higher priority to access DRAMs. The arbiter treats different DRAM access request as DRAM master, and that makes there be 6 masters which are trying to access DRAMs by sending their request to the arbiter. After one of them gets the grant from the arbiter, it owns DRAM bus and begins to do memory data transaction. The masters are: Refresh high request, AGP high access request, CPU read request, CPU write request, AGP low access request, Refresh low request. The order of these masters shown above also stands for their priority to access memory.

### 5.2.5. REFRESH CYCLE

The refresh cycle will occur every 15.6us. It is timed by a counter of 14Mhz input. The CAS[7:0]# will be asserted at the same time, and the RAS[5:0]# are asserted sequentially.

### 5.2.6. GRAPHIC WINDOW RE-MAPPING

For supporting the A.G.P. bus, SiS600 supports an address range, Graphic Window (GW), which is virtual, contiguous, programmable range and can be re-mapped by Graphic Address Re-mapping Table (GART) to non-contiguous pages of the system memory. Graphic Window is defined by Graphic Window Base Address (GWBA) configuration register. The size of Graphic Window is

allowed the selection of 4M, 8M, 16M, 32M, 64M, 128M and 256M. It is defined by Graphic Window Size Register.

When an access is addressing to the Graphic Window, the memory controller will firstly read the re-mapping relation from GART, then translate to the physical address and read the data from system memory.

*Figure 5.2-5* shows the graphic address re-mapping function.

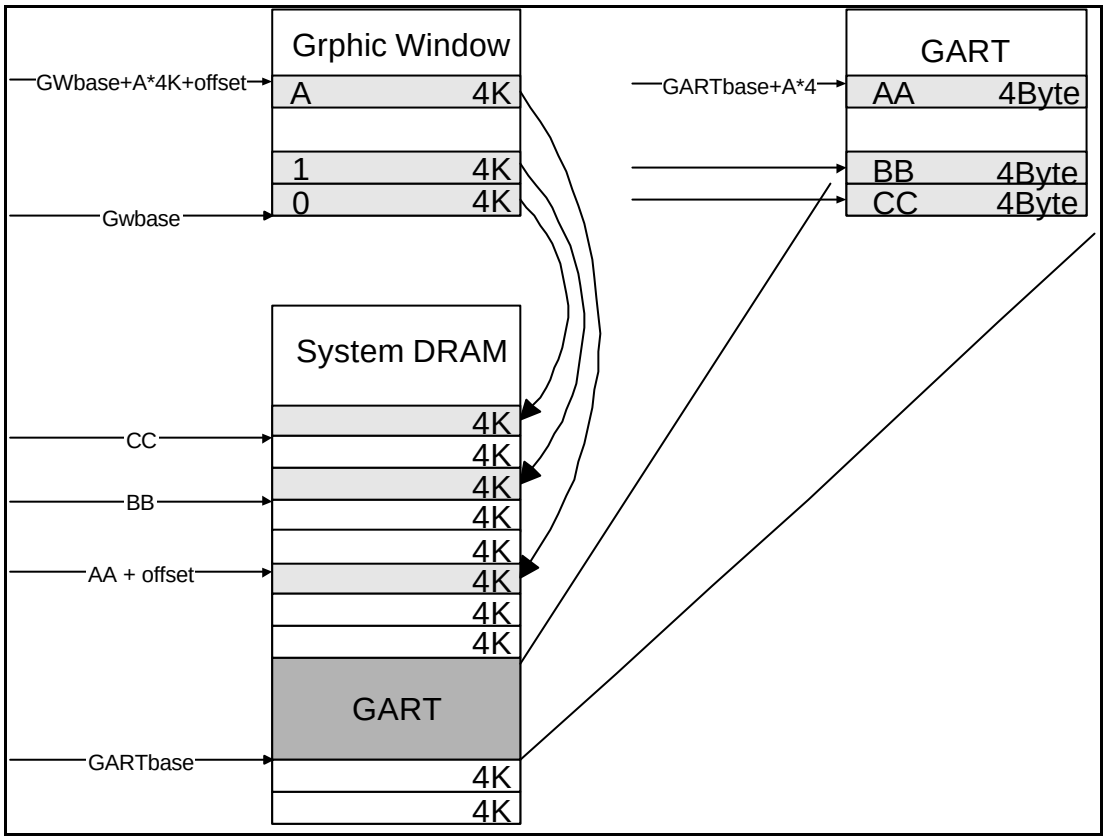



**Figure 5.2-5 Graphic Address Re-mapping Function**

In order to improve the performance of accessing graphic window, SiS600 builds a page table cache inside. The page table stores re-mapping relations of the most frequently used pages. It is set-associated with 8 ways and each way has 2 entries. If the page table stores the re-mapping relation of one specific page, then access within this page will directly be translated to the physical address without looking up the GART and be read by the memory controller.

**DATA INTEGRITY**

SiS600 supports ECC data integrity feature on the 64-bit DRAM interface. This feature provides single-error correction and double-error detection. After system reset, the DRAM controller is set in non-ECC mode. In this mode there is no provision for protecting the integrity of data within the DRAM. After the BIOS configuration procedure detects that all memory modules support 72-bit ECC mode operation, the DRAM controller can be set in ECC mode.

**INITIALIZATION**

When ECC is enabled, the whole DRAM modules must be first initialized by doing writes before the DRAM read operations to establish the correlation between 64-bit data and associated 8-bit ECC code which does not exist after power-on.

**ECC DETECTION AND CORRECTION**

During DRAM read operations, the full Qword of data and the associated ECC code are transferred simultaneously from DRAM to the SiS600. If there is a single-bit error in the 72-bit signals (64-bit data plus 8-bit code), the ECC mechanism will automatically recover the correct 64-bit data. Note that the recovered data is transferred to the requesting master (Host, PCI, or AGP interface), but the DRAM controller does not initiate a DRAM write cycle to fix the single-bit error in DRAM.

If the ECC mechanism detects a two-bit error, SiS600 can report this error by SERR# or record it on the Error Status Register optionally.

**ECC GENERATION**

During DRAM write operations, SiS600 automatically generates an 8-bit ECC code for the 64-bit data. If the requested write operation transfers single or multiple Qword of data, the ECC DRAM write can be completed without any overhead. If the write request only transfers less than 64 bits of data, the DRAM controller performs a read-merge-write operation.

## 5.3. PCI BRIDGE

The PCI bridge of SiS600 consists of three parts: PCI arbiter, PCI master bridge and PCI target bridge. The PCI arbiter controls the assignment of PCI bus ownership among all PCI masters. The PCI master bridge forwards the transactions from host bus. The PCI target bridge claims PCI cycles toward system memory or AGP bus as required by PCI master devices.

### 5.3.1. PCI ARBITER

The main function of PCI arbiter takes charge of the PCI bus ownership assignment. This PCI arbiter supports at most 4 external PCI masters using standard PCI REQ#/GNT# mechanism and 1 PCI master using PHOLD#/PHLDA# mechanism. The master that uses PHOLD#/PHLDA mechanism is not pre-emptive. That means the master can own the bus as long as it wishes after it gains the control of PCI bus. The master that use PHOLD#/PHLDA# mechanism to access PCI bus is typically SiS5595 chip.

**Arbitration Algorithm**

**PCI Masters (Agent 0~6, SIO) Requests**
Figure6.4-1 Arbitration Tree shows the arbitration tree in arbiter design. Whenever a PCI cycle occurs, priority status will be changed. The initial priority for master 0-7 to own PCI bus is 4 -> 0->SIO->2->5->1->6->3->4...

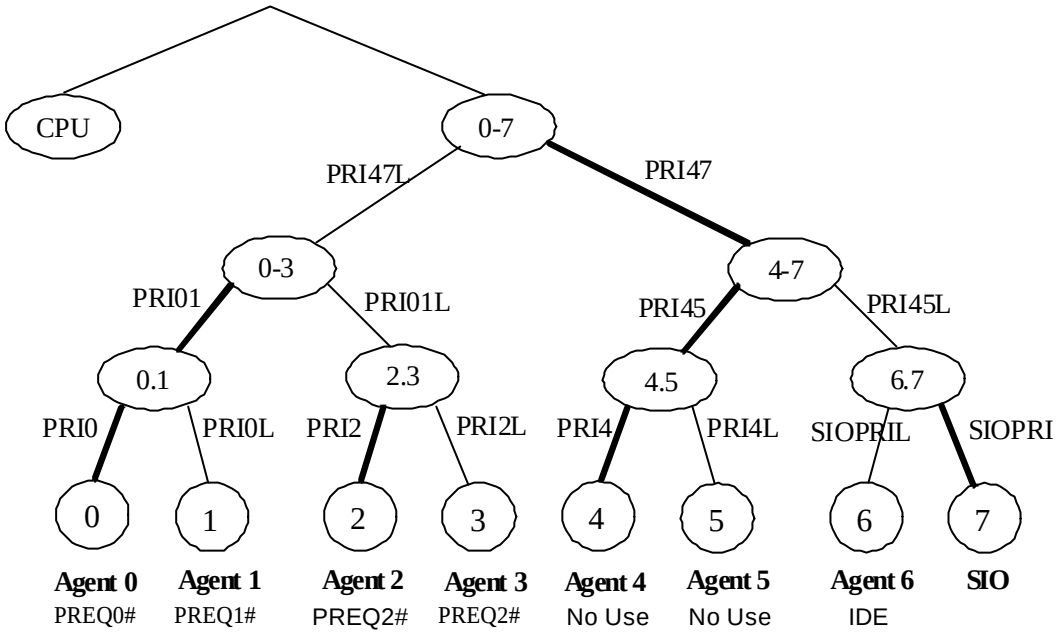



**Figure 5.3-1 Arbitration Tree**

**NOTE:** 1. "SIO" means the System I/O for PCI to ISA bridge (SiS5595).

2. The arbiter will treat PHOLD# as Agent SIO.

**CPU Request**

To avoid CPU being constantly held for a long time while PCI masters continuously deliver requests to the arbiter, SiS600 implemented a timer-based algorithm to reserve PCI bandwidth for CPU. Three timers, PCI Grant Timer (PGT)/ Master Latency Timer (MLT)/ CPU Idle Timer (CIT), are included in the host bridge for this purpose.

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

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

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

To prevent the host bridge from capturing PCI bus too long while CPU actually has nothing to do at all, the third timer, CPU Idle Timer (CIT) is included in our design. CIT starts to count when the host bridge get a chance to

start a transaction on PCI bus, but is reloaded with its initial value whenever the host bus leaves idle state. CIT actually keeps track on how long the CPU is in idle state. After CIT is expired, the host bridge de-asserts its request signal just in the same manner as the case of MLT's expiration.

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

### 5.3.2. PCI BUS INTERFACE

The bridge performs medium address decoding and supports all memory cycles (including memory write, memory write and invalidate, memory read, memory read multiple and memory read line) and configuration cycles targeting the bridge.

While the PCI master creating a resource lock on the bridge, the bridge will translate this exclusive access by keeping the Pentium II bus ownership continuously.

### 5.3.3. TARGET INITIATED TERMINATION

In general, the bridge is able to source or sink the data requested by the master until the master terminates the transaction. But sometimes when the bridge is unable to complete the request, it may use the STOP# signal to initiate termination of the transaction.

Target retry may occur for one of four reasons.

- The bridge cannot meet the initial latency requirement.
- The bridge is currently locked by another master.
- The posting buffers become unavailable during PCI write.
- The posting buffers are not flushed during PCI read.

Disconnect without data may occur for one of two reasons.

- The bridge cannot meet the subsequent latency requirement.
- The posting buffers become unavailable during PCI write.

Disconnect with data may occur for one of three reasons.

- The burst length reaches the resource boundary.
- The prefetch is not enabled during PCI read.
- The posting buffers become unavailable during PCI write.

### 5.3.4. PCI MASTER CONTROLLER (PMR)

- PMR locally has 4 level deep 3DW wide circular buffer.
- Supports Asynchronous PCI clock.
- Supports post write, burst read/write.
- Zero wait state burst cycles up to 512 bytes.
- Supports fast back-to-back transfer.
- Supports PCI LOCK# transaction.

- Fast reset emulation.
- Fast A20M# emulation.
- CPU involves PCI master arbitration.
- Supports CPU idle timer, PCI grant timer and master latency timer.
- Supports configuration access format type I, II.
- Using configuration mechanism #1.
- When CPU issues a line read cycle, converts the toggle mode address to the linear mode address automatically, then forwards to the PCI side.

The PCI Master Controller forwards the CPU cycles not targeting the local memory to the PCI bus. In the case of a 64-bit CPU request or a misaligned 32-bit CPU request the PMR assumes the read assembly and write disassembly control. A 4 level local posted write buffer is implemented to improve the CPU to PCI memory write performance. For PCI memory write cycles, the CPU data are pushed into the buffer as soon as FIFO is empty and then forwarded to PCI bus. If a line read cycle forwarded to the PCI, PCI master controller will convert the toggle mode address to the linear mode address automatically, then forward to the PCI side. If the consecutive written data is in DW incremental sequence, they will be transferred to the PCI bus in a burst manner.

The PMR provides a mechanism for converting standard I/O cycles on the CPU bus to Configuration cycles on the PCI bus.

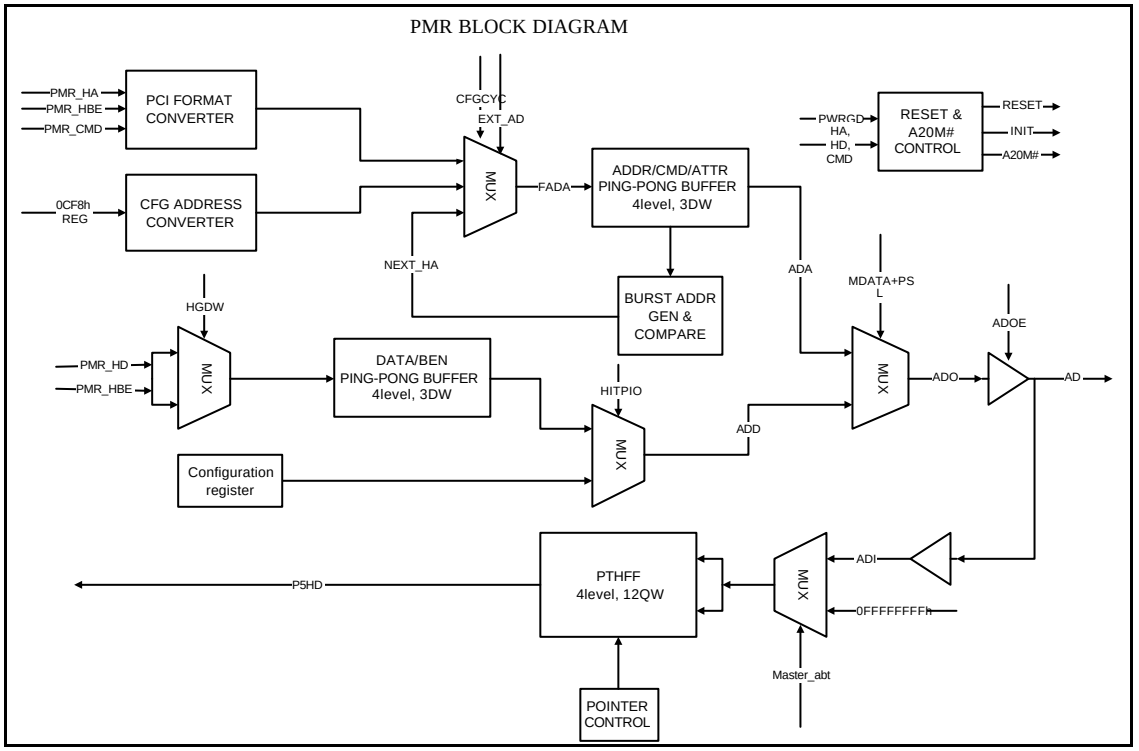



**Figure 5.3-2 Block Diagram for PMR**

### 5.3.5. PCI BURST AND POST MODE

If the BIOS turn on the burst feature, SiS chip will forward the cycle to PCI side and progress in burst mode. If this is a line read cycle, because CPU deliver a burst cycle in toggle mode, PMR will translate it to linear mode and start from “00” QW.

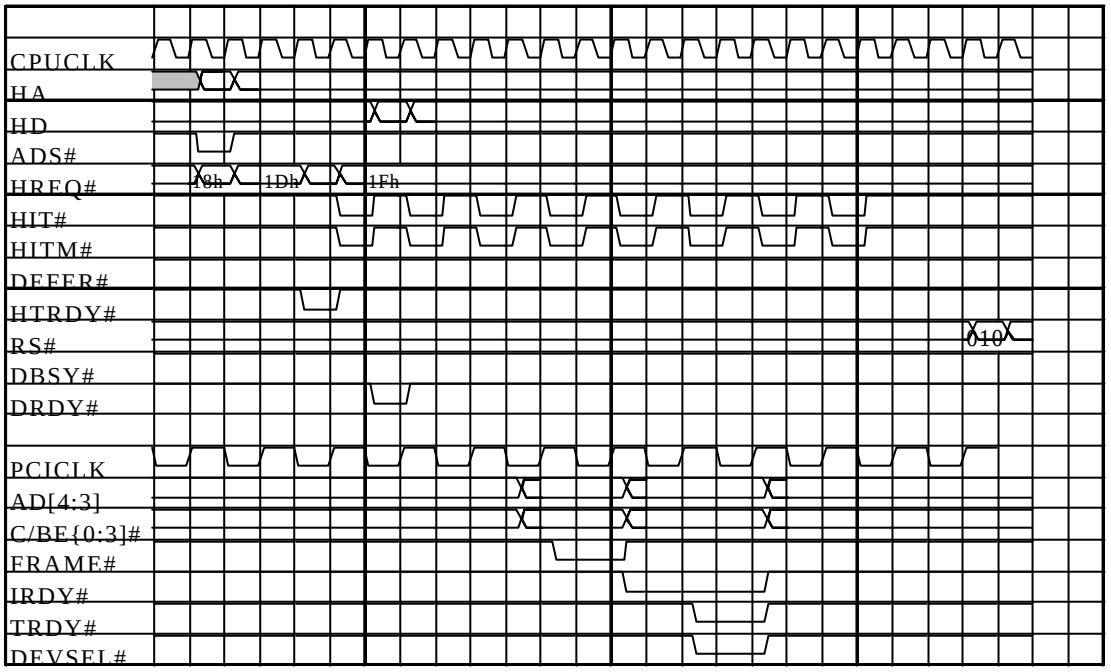



**Figure 5.3-3 Line Read Cycle**

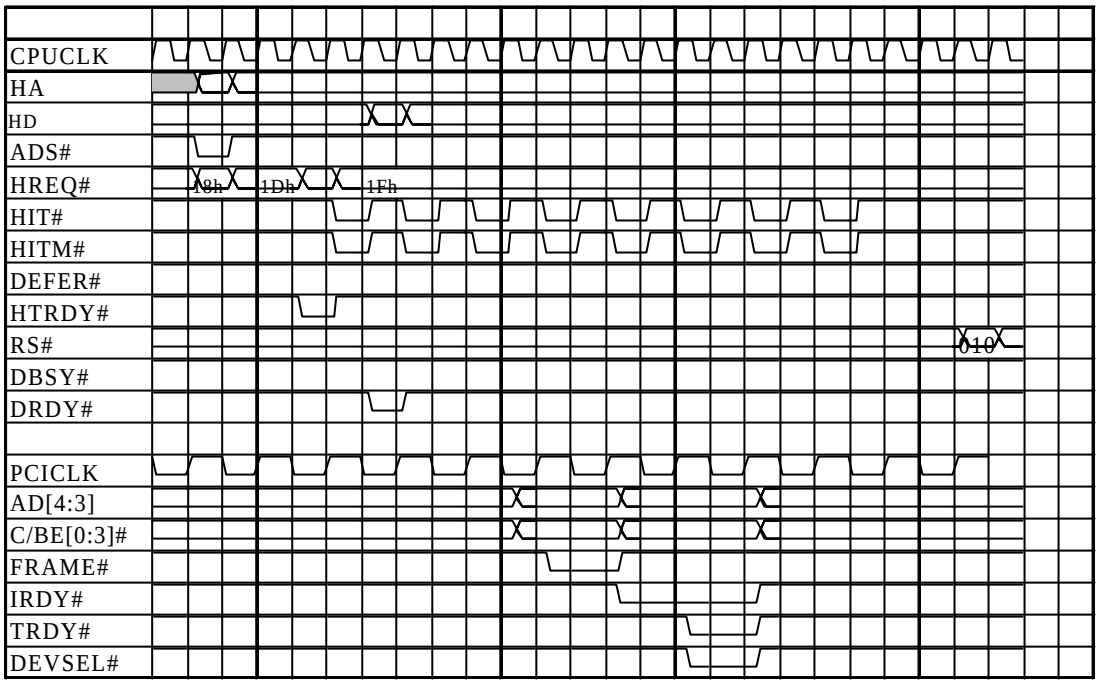



**Figure 5.3-4 Non-Post Cycle**

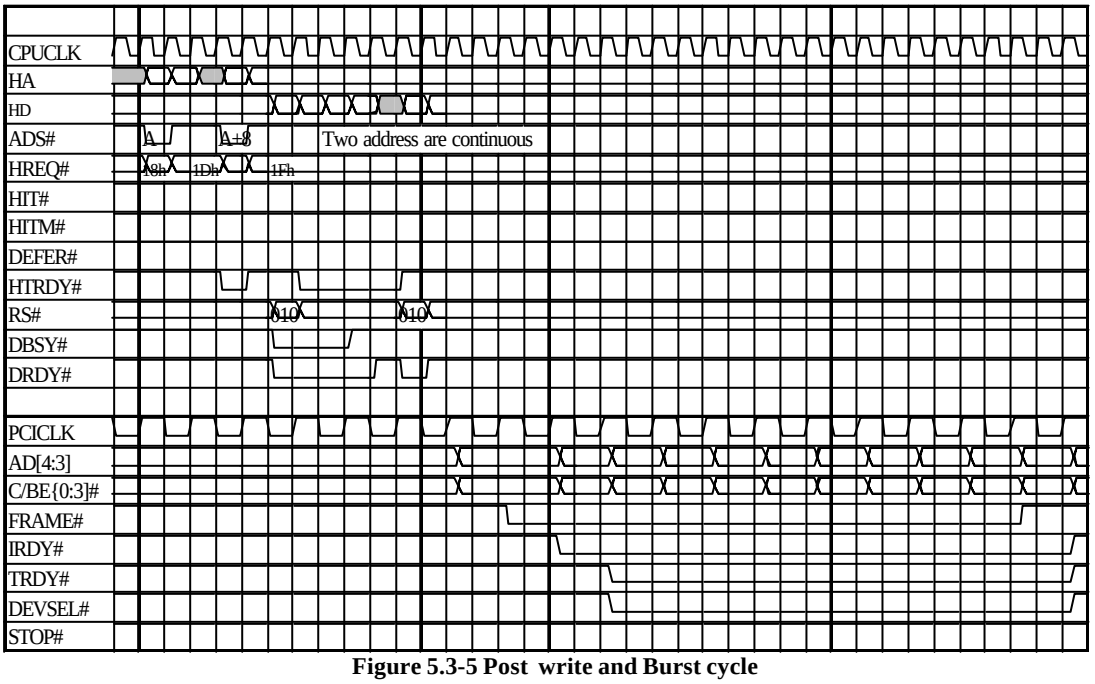



**Figure 5.3-5 Post write and Burst cycle**

### 5.3.6. 66MHZ PCI MASTER BRIDGE (PMR66)

The 66MHz PCI Master Bridge mainly forwards memory and I/O read/write cycles from host bus destining to AGP bus. To determine whether it is an AGP transaction or not, the decoding circuit of host interface refers to the Address Base and Address Limit in the configuration space of virtual PCI-to-PCI bridge (Bus 0, Device 2). However, PMR66 is responsible for generating configuration cycles for A.G.P. device.

Basically, the PMR66 owns the same function as PMR33 does, but they work with different bus segments. As a result, the mechanism of CPU Idle Timer on AGP (CITA), AGP Grant Timer (AGT) and CPU Latency Timer on AGP (CLTA) provide a fair and efficient arbitration mechanism on AGP bus as CIT, PGT and MLT do on PCI bus. Except that, for memory cycles, the PMR66 is able to generate burst cycles. And, for memory write cycles, it is capable to translate them into post-write cycles based on two kinds of lead-off timing.

Since the operating frequency of AGP bus is higher than traditional 33Mhz PCI bus and even the same as host bus. Thereafter, the consuming of post-write cycles would be at a great rate and cause more non-burst transactions on AGP bus. To have good bus utilization and gain higher performance whenever CPU issues post-write cycles, the PMR66 will postpone the post-write cycles and try to wait for a period. Then, the PMR66 combines them into a single burst transaction if they have continuous addresses.

Beside, SiS600 allows PCI masters to write transactions of 33Mhz PCI bus toward AGP bus. Subsequently, additional data path from PSL33 is implemented to transfer such cycles for PCI masters as depicted in Figure 5.3-3. Block diagram of 66Mhz PCI Master Bridge.

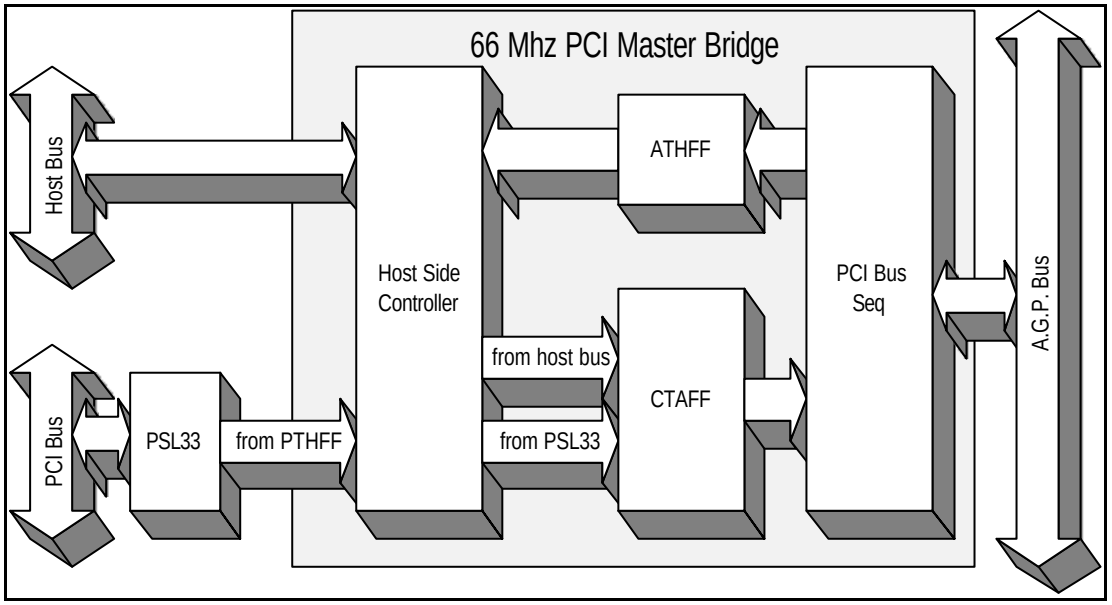



**Figure 5.3-6 Block diagram of 66Mhz PCI Master Bridge**

### 5.3.7. 66MHZ PCI TARGET BRIDGE (PSL66)

PSL66 claims PCI transactions that are destining system memory or 33Mhz PCI bus. All memory write accesses with PCI protocol initiated by A.G.P. compliant master are claimed by PSL66.

For memory read cycles, PSL66 only claims transactions that are targeting system memory.

The system memory referred above includes main memory range and Graphics Window.

PSL66 would not respond to I/O, PCI configuration, Interrupt acknowledge and special cycles. If the A.G.P. compliant master generates any of these cycles, the cycle will be ended with master-abort.

When PSL66 is processing accesses toward system DRAM, it checks whether the CPU is held or not. If the CPU is not held, PSL66 generates hold request to host interface. Host interface responds to this hold signal by asserting HLDA66. After the HLDA is returned, PSL66 has totally control over the host bus and then the accesses can be continued.

PSL66 and PMR66 share the same FIFO (CTAFF and ATHFF). The usage of the shared FIFO is mutual exclusive. PSL66's operation toward system memory will be suspended until all PMR66's data are flushed out of CTAFF.

In general, the functions of PSL66 are the same with the ones of PSL33. For further details of functional descriptions, make reference to the paragraph of PSL33.

## 5.4. A.G.P. COMPLIANT TARGET/HOST-TO-PCI66 BRIDGE

A.G.P. Compliant Target/Host-to-PCI66 Bridge is constituted by 4 blocks: A.G.P. arbiter, A.G.P. Target, PMR66 and PSL66. The A.G.P. arbiter coordinates the operation of A.G.P. Target, PMR66 and A.G.P. compliant master. The A.G.P. target is response for transferring data between A.G.P. bus and dram controller when the A.G.P. compliant target initiate bus transactions with A.G.P. protocol. PMR66 forward cycles from host bus or 33Mhz PCI bus to A.G.P. bus with PCI protocol. PSL66 claims PCI transactions which are destining to system memory.

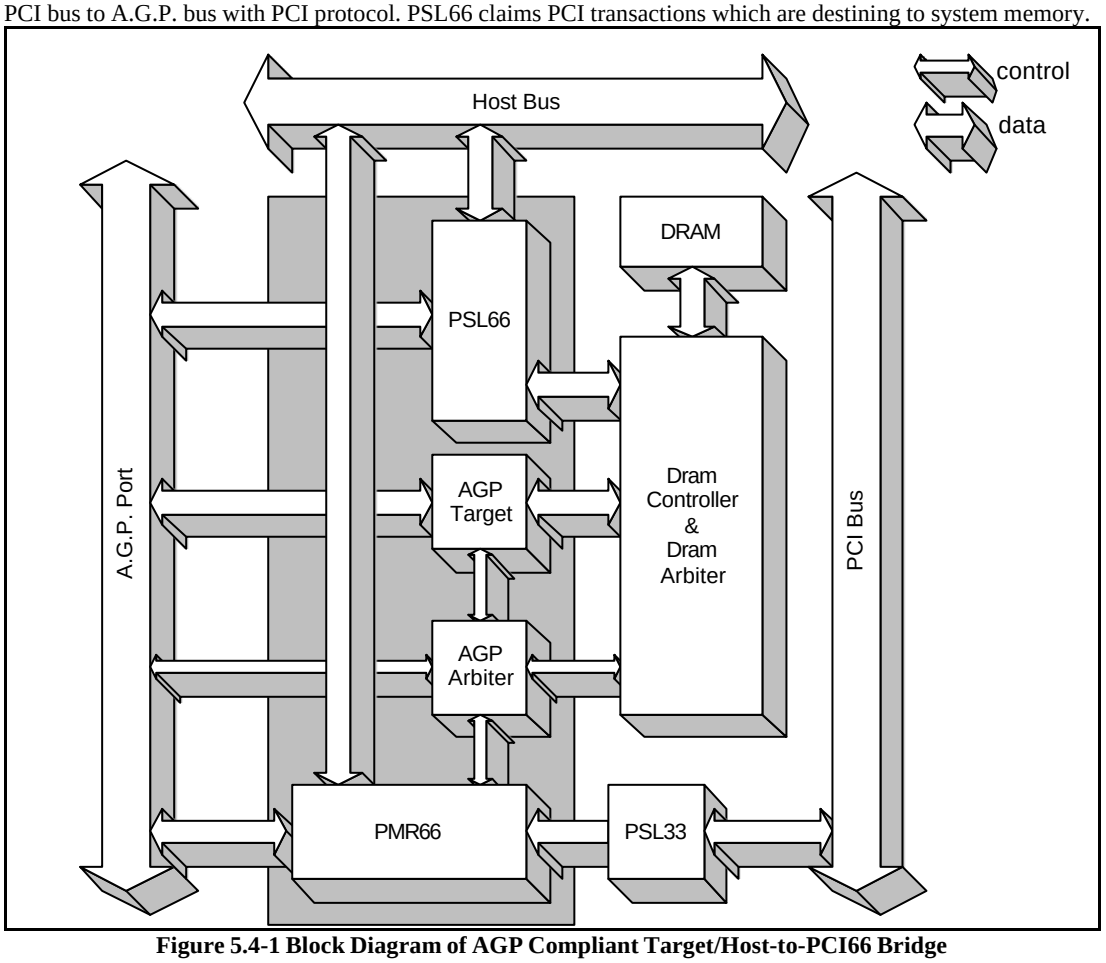



**Figure 5.4-1 Block Diagram of AGP Compliant Target/Host-to-PCI66 Bridge**

## 5.5. POWER MANAGEMENT SUPPORT

To support power management feature of SiS5595, SiS600 reports all PCI masters ¡ ¦activities and optionally reports activities on A.G.P. bus. SiS600 to SiS5595 via the pin BMREQ#. The information contained by BMREQ# can be classified into two categories. The first category is CPU operation and the second category is master operation. Each category occupies different time slot. If BMREQ# is reporting CPU operation in this clock, then BMREQ# will report master operation in the next clock and vice versa. Configuration Register 6Bh of host bridge is used to define what kinds of operations should be reported.

SiS600 may also trap ACPI I/O port accesses to support ACPI S3 and S2 states and to disable system arbiter. Configuration register 68~69h is used to define the ACPI I/O space base address and SiS600 uses this register to trap the ACPI I/O port accesses.

## 5.6. INTEGRATED PCI MASTER/SLAVE IDE CONTROLLER

***OVERVIEW***

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

The IDE Controller block diagram is shown as below:

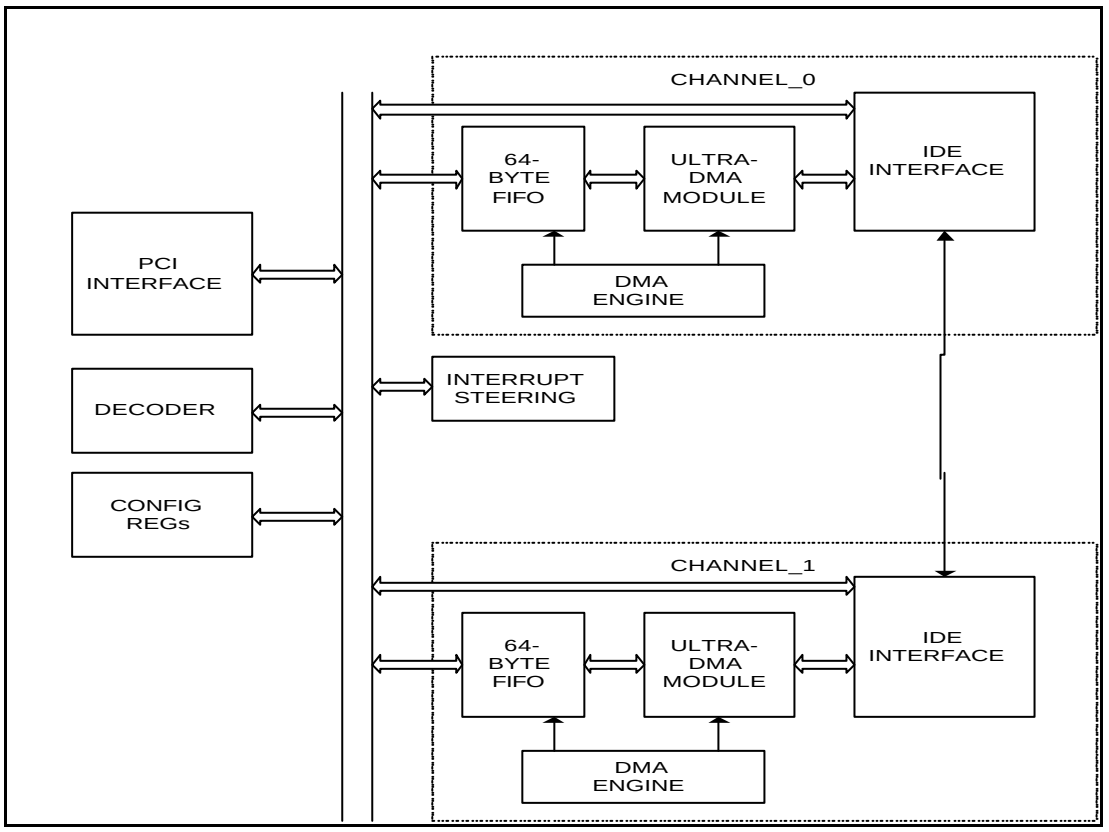



**Figure 5.6-1 Block Diagram for IDE Controller**

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

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

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

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

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

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

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

**Primary Channel:**

**Table 5.6-1 Accessing Method to I/O Ports for Primary Channel**

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

**Secondary Channel:**

**Table 5.6-2 Accessing Method to I/O Ports for Secondary Channel**

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

In Native mode, the interrupt requests of both channels (channel 0 and channel 1) share the same PCI interrupt pin. The interrupt pin may be re-routed to any one of eleven ISA compatible interrupts (IRQ[15:14], IRQ[12:9], and IRQ[7:3]) via programming Register 61h bits 3:0 in SiS5595 PCI to ISA bridge Configure space.

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

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

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

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

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

### PIO mode operation

The IDE controller is capable of doing prefetch or postwrite in PIO mode. The count(in bytes) of prefetch length for each channel can be programmed in Prefetch Count Registers 4Ch~4Dh and 4Eh~4Fh in PCI IDE Configuration space. Normally, the count will be programmed as 512($2^9$), which is the size of a single sector. The prefetch and postwrite functions can be enabled or disabled independently through control bits in Register 4Bh of PCI IDE configuration space. When prefetch is enabled, the controller will start prefetching when the first read data port command is received. It will keep prefetching until the FIFO is full or when prefetch count is reached. Whenever the FIFO becomes non-empty again, the prefetch will automatically resume until the prefetch count is reached.

When post write is enabled, the host can write data to FIFO in word- or Dword- increment. The IDE controller will automatically start IDE write cycles as long as FIFO is non-empty. When the fast post write function is

enabled, the write IDE data port command on PCI bus will last for 3 PCI clocks only. When disabled, the PCI command will be 5 PCI clocks.

**DMA mode operation**

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

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

**When the DMA engine is writing IDE**

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

**When the DMA engine is reading IDE**

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

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

**Ultra-DMA/33 Operation**

Ultra DMA is a fast data transfer protocol used on IDE bus. By utilizing both the rising edge and the falling edge of the data strobe signal to latch data from DD[15:0], the data transfer rate is effectively doubled than that of the traditional multi-word DMA while the highest fundamental frequency on the cable is the same. In view of the faster transfer rate on IDE bus may easily fill the FIFO up when reading IDE device, in such condition the IDE bus will be idle and result in system performance degradation, SiS Chip lengthens the internal FIFO for each channel (channel 0/channel 1) to 16-Dword to improve system performance. When the FIFO is half-full (or 3/4-full, programmable), the DMA engine should request for PCI bus by asserting an internal request signal to system arbiter. The system arbiter, based on an algorithm described in the previous sections, shall grant the PCI bus to DMA engine by asserting an internal grant signal to it. Ideally, the FIFO should never be full during data-in operation so that the burst data transfers on IDE will not be suspended. When the IDE controller is transferring

data from system memory to IDE, the DMA engine will initiate PCI burst cycles to read data from memory into FIFO until FIFO is full. The FIFO will decrease at the rate of the selected Ultra DMA mode as the IDE controller doing data-out operation. In the best situation, the FIFO should not be empty during data-out operation otherwise the burst data transfer on IDE will be suspended.

The Ultra-DMA mode can be enabled on a per-device basis and all three timing modes(0-2) are supported by programming the corresponding configuration registers. For Ultra-DMA operations, the following signal lines shall change to their new definition when IDACK# is asserted. These signals will revert back to their old definitions right after IDACK# is de-asserted.

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

**Table 5.6-3 Table for Different Command Definition**

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

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

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

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

1. During the burst data transfer, the host(CPU) is trying to access the command/control block registers. Since the command/control block access cycle is assigned to have higher priority than data transfer cycles, the SiS Chip must first terminate the burst, de-asserts the IDACK# signal, generate the corresponding DA[2:0] and CS[1:0] on IDE bus, and then complete the command/control block register access cycle. After that, the burst can be resumed by entering the Burst Initiation phase when the device re-asserts IDREQ.

## 5.7. BALL CONNECTIVITY TESTING

SiS Chip will provide a NAND chain Test Mode by TEST_PIN# signal is pull low. In order to ensure the connections of balls to tracks of main board, SiS Chip provides a simple way to do connective measurements. Basically, an additional 2-input-NAND gate is added into the I/O buffer cells. And, one of inputs of NAND gate is connected to input pin of I/O buffer as test input port in test mode. To monitor the test result at test output port, the output of the NAND gate is connected to the other input of the next NAND gate. Such that, the test result could be propagated and it forms a NAND tree, as depicted in Figure 5.7-1. To adapt to the scheme, all output buffers of SiS Chip are changed to bi-direction buffers to accept test signals.

### 5.7.1. TEST SCHEME

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

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

First of all, logic LOW is driven into TESTIN1 pin from track on main board. If logic HIGH could be observed at TESTOUT pin, it means that the connection of TESTIN1 pin to track is good, as shown in Figure 5.7-2. To test TESTIN2 pin, TESTIN2 pin should be driven LOW also. And, TESTIN1 pin should be kept at logic HIGH, such that the test result could be passed to TESTOUT pin and so on. Although SiS Chip operates at 3.3V, all input buffers of SiS Chip are 5V-input tolerance. Hence, all test signal could go up to 5V.

### 5.7.2. MEASUREMENTS

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

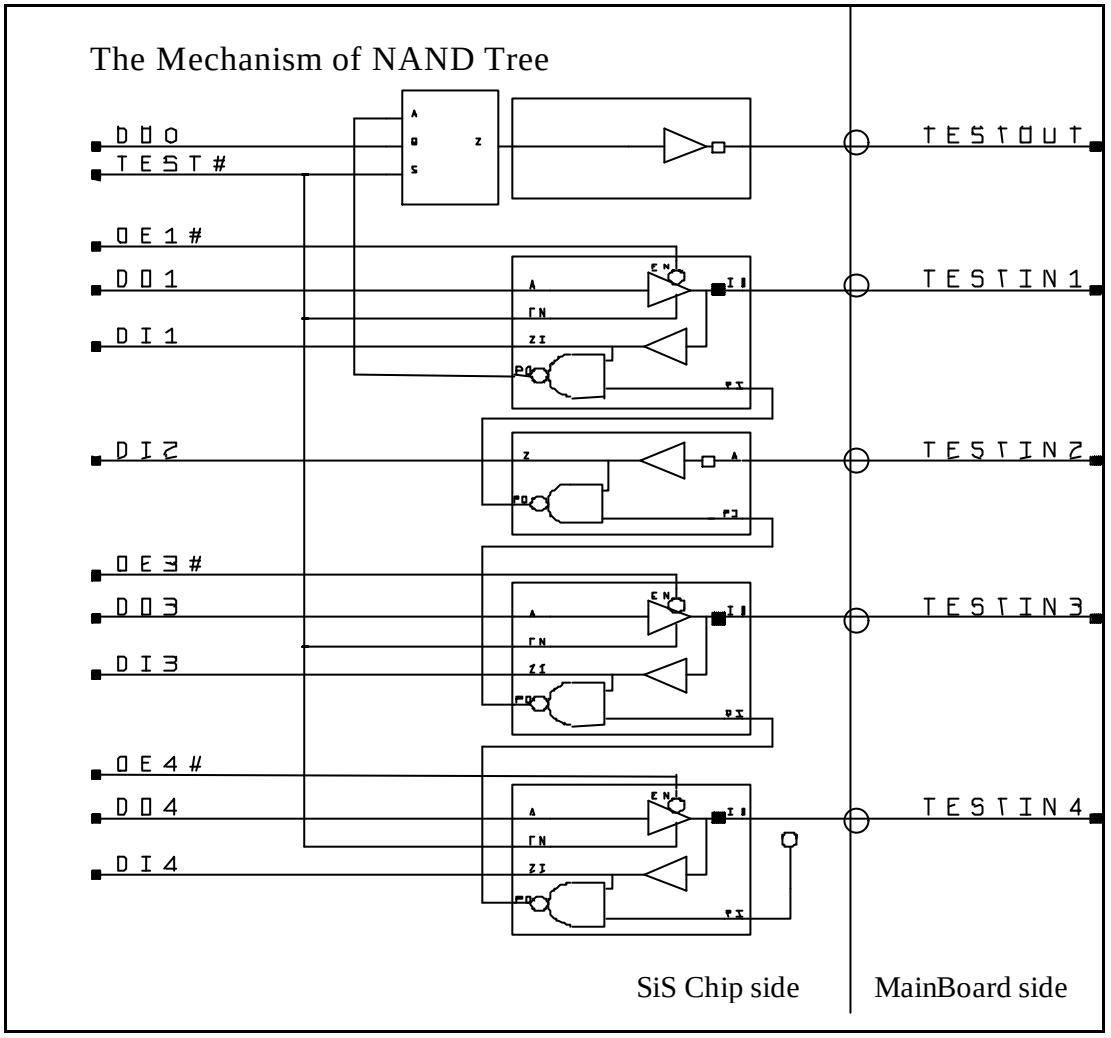



**Figure 5.7-1 The Mechanism of NAND Tree**

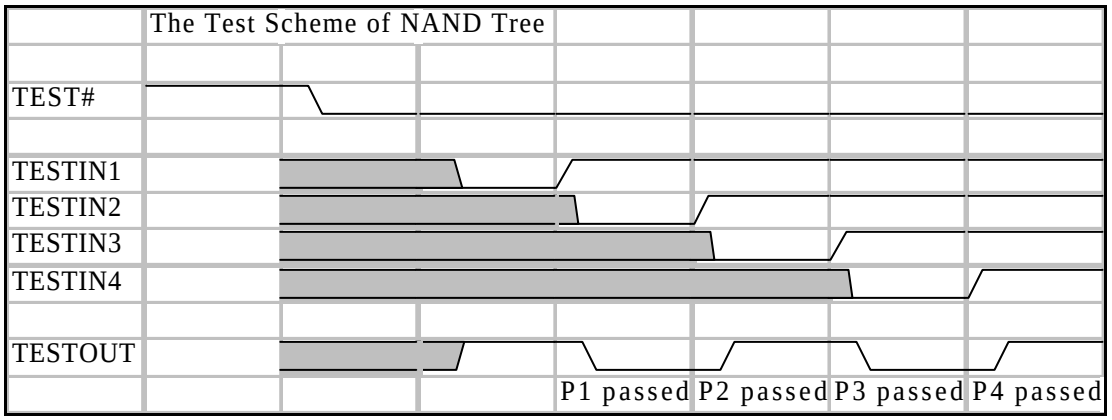


**Figure 5.7-2 The Test Scheme of NAND Tree**

**Table 5.7-1 NAND Tree List for SiS600**

| TEST VECTOR | BALL NAME LIST | TEST OUTPUT BALL NAME |
|---|---|---|
| TESTIN0[1:52]<br>(NAND Tree 1) | AAD0, AAD1, AAD2, AAD3, AAD4, AAD5, AAD6, AAD7, AC/BE0#, AD_STB0, AAD8, AAD9, AAD10, AAD11, AAD12, AAD13, AAD14, AAD15, AC/BE1#, ASERR#, ASTOP#, APAR, ADEVSEL#, ATRDY#, AIRDY#, AFRAME#, AC/BE2#, AAD16, AAD17, AAD18, AAD19, AAD20, AAD21, AAD22, AAD23, AC/BE3#, AD_STB1, AAD24, AAD25, AAD26, AAD27, AAD28, AAD29, AAD30, AAD31, SBA7, SBA6, SBA5, SBA4, SB_STB, SBA3, SBA2. | PLHDA# |
| TESTIN1[1:68]<br>(NAND Tree 2) | SBA1, SBA0, ST2, ST1, ST0, RBF#, PIPE#, AGNT#, AREQ#, RSTI, PCIRST#, XHCLK, AD0, AD1, AD2, AD3, AD4, AD5, AD6, AD7, C/BE0#, AD8, AD9, AD10, AD11, AD12, AD13, AD14, AD15, AGPCLK, C/BE1#, PAR, SERR#, STOP#, PLOCK#, DEVSEL#, TRDY#, IRDY#, FRAME#, C/BE2#, AD16, AD17, AD18, AD19, AD20, AD21, AD22, AD23, CBE3#, AD24, AD25, AD26, AD27, AD28, AD29, AD30, AD31, REQ3#, REQ2#, PCICLK, REQ1#, REQ0#, PHOLD#, HD62#, HD58#, HD61#, HD63#, HD55#. | BMREQ# |
| TESTIN2[1:54]<br>(NAND Tree 3) | HD56#, HD60#, HD50#, HD53#, HD54#, HD57#, HD59#, HD46#, HD48#, HD49#, HD52#, HD51#, HD41#, HD42#, HD47#, HD45#, HD44#, HD39#, HD36#, HD40#, HD43#, HD34#, HD37#, HD38#, HD33#, HD32#, HD35#, HD28#, HD31#, HD29#, HD30#, HD26#, HD27#, HD25#, HD24#, HD22#, HD23#, HD19#, HD21#, HD18#, HD16#, HD20#, HD13#, HD17#, HD11#, HD15#, HD10#, HD12#, HD14#, HD7#, HD9#, HD6#, HD8#, HD4#. | GNT3# |
| TESTIN3[1:62]<br>(NAND Tree 4) | HD5#, HD2#, HD3#, HD0#, HD1#, CPURST#, BREQ0#, HA29#, HA30#, HA26#, HA31#, HA24#, HA27#, HA28#, HA22#, HA20#, HA23#, HA21#, HA25#, HA19#, HA15#, | GNT2# |

| | | |
|---|---|---|
| | HA18#, HA17#, HA16#, HA11#, HA13#, HA12#, HA14#, HA8#, HA10#, HA7#, HA5#, HA3#, HA9#, HA6#, HA4#, BNR#, HREQO0#, BPRI#, HREQO1#, HTRDY#, HREQO4#, DEFER#, HLOCK#, HREQO2#, DRDY#, HREQO3#, RS0#, HITM#, HIT#, DBSY#, RS2#, RS1#, ADS#, IRQ14, IRQ15, IDA7, IDA8, IDA6, IDA9, IDA5, IDA10. | |
| TESTIN4[1:70]<br>(NAND Tree 5) | IDA11, IDA3, IDA12, IDA2, IDA13, IDA1, IDA14, IDA0, IDA15, IBDRQ, IADRQ, IAIOWC#, IAIORC#, IBIOWC#, IBIORC#, IBCHRDY, IACHRDY, IADACK, IBIRQ, IAIRQ, IADSA1, IADSA0, IADSA2, IACS0#, IACS1#, MD31, MD63, MD30, MD62, MD29, CKE5, CKE4, CKE3, MD61, MD28, MD60, MD27, MD59, MD26, MD58, MD25, MD57, MD24, MD56, MD23, MD55, MD22, MD54, MD21, MD53, MD20, MD52, MD19, MD51, MD18, MD50, MD17, MD49, MD16, MD48, MPD3, MPD7, MPD2, MPD6, CKE0, DQM3, DQM7, DQM2, DQM6, MA13. | GNT1# |
| TESTIN5[1:65]<br>(NAND Tree 6) | MA12, MA11, MA10, MA9, MA8, MA7, MA6, MA5, MA4, MA3, MA2, MA1, MA0, SRAS#, CS0#, CS1#, CS2#, CS3#, CS4#, CS5#, DQM1, DQM5, DQM0, DQM4, RAMWB#, RAMWA#, SCAS#, MPD1, MPD5, MPD0, MPD4, MD15, MD47, MD14, MD46, MD13, MD45, MD12, MD44, MD11, MD43, MD10, MD42, MD9, MD41, MD8, MD40, MD7, MD39, MD6, MD38, MD5, MD37, MD4, MD36, MD3, MD35, MD2, MD34, MD1, MD33, MD0, CKE2, CKE1, MD32. | GNT0# |

## 6. HARDWARE TRAP

There are three pins are used for trapping purpose to identify the hardware configurations at the power-up stage. These pins will be recognized as "1" if pull-up resistors are used; and will be recognized as "0" if pull-down resistors are used. The following table is a summary of all the Hardware Trap pins in SiS Chip.

| SYMBOL | DESCRIPTION |
|---|---|
| MD63 | **Internal DLL Circuits for PCI clock to optimize timing control**<br>Pull-up: Disable<br>Pull-down: Enable |
| MD62 | **Internal PLL circuit for AGP clock to optimize timing control**<br>Pull-up: Disable<br>Pull-down: Enable |
| MD60 | **Internal PLL Circuits for CPU clock to optimize timing control**<br>Pull-up: Disable<br>Pull-down: Enable |
| MD50 | **Asynchronous/Synchronous Clock support for CPU and A.G.P. Clocks**<br>Pull-up: Disable<br>Pull-down: Enable<br>Programming the Register C8h bit2 in the Host Bridge Configuration Space also can control this function. |

***Note: there are pull-down resistors on MD lines.***

# 7. CONFIGURATION REGISTER

## 7.1. DEVICE 0, FUNCTION 0 ( HOST-TO-PCI BRIDGE)

### 7.1.1. CONFIGURATION SPACE HEADER

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 00-01h | Vendor ID | 1039h | RO |
| 02-03h | Device ID | 5600h | RO |
| 04-05h | PCI Command Register | 0005h | RO<br>R/W |
| 06-07h | PCI Status Register | 0210h | RO<br>R/W<br>WC (*) |
| 08h | Revision ID | 10h | RO |
| 09h | Programming Interface | 00h | RO |
| 0Ah | Sub-Class Code | 00h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Master Latency timer | FFh | R/W |
| 0Eh | Header Type | 80h | RO |
| 0Fh | BIST | 00h | RO |
| 10-13h | Graphic Window Base Address | 00000000h | RO<br>R/W |
| 14-33h | Reserved | | |
| 34h | Capability Pointer | C0h | RO |

- WC stands for Write Clear. A WC bit will be reset to 0 by being written a “1”, and will still be 0 by being written a “0”. For instance, to write value 0100_0000_0000_0000b to the register will clear bit 14 and the other bits keep the same value.

### 7.1.2. REGISTERS FOR HOST & DRAM

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 50h | Host Interface Control I | 02h | R/W |
| 51h | Host Interface Control II | 00h | R/W |
| 52h | DRAM MISC control 1 | 00h | R/W |
| 53h | DRAM Timing control | 00h | R/W |
| 54h | DRAM RAS# Timing Control | 00h | R/W |
| 55h | DRAM MISC Control 2 | 00h | R/W |
| 56h | DRAM MISC Control 3 | 00h | R/W |
| 57h | SDRAM Control | 00h | R/W |
| 58h | FP/EDO DRAM CAS# Timing Control | 00h | R/W |
| 59h | A.G.P./PCI Buffer Strength and Current Rating | 00h | R/W |
| 5Ah | DRAM Buffer Strength and Current Rating | 00h | R/W |
| 5F~5Bh | Reserved | 00h | R/W |
| 60/61/62h | DRAM Type Registers of Bank 0/1/2 | 00h | R/W |
| 63h | DRAM Status Register(Bit-x = Bank-x ) | FFh | R/W |
| 67~64h | ECC Status Register | 00000000h | R/W |
| 69~68 | ACPI I/O Space Base Address Register | 0000h | R/W |
| 6Ah | SMRAM Access Control | 00h | R/W |
| 6Bh | System Event Monitor control for Power Management | 00h | R/W |
| 6Ch | DRAM Self-Refresh Control for Power Management | 00h | R/W |

### 7.1.3. SHADOW RAM & PCI-HOLE AREA

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 73~72h/71~70h | Shadow RAM attribute & Read/Write Control | 0000h | R/W |
| 74/75h | ECC Function Control | 00h | R/W |
| 76h | Reserved | 00h | R/W |
| 77h | Characteristics of PCI-Hole area | 00h | R/W |
| 79~78h | Allocation of PCI-Hole area #1 | 0000h | R/W |

| | | | |
|---|---|---|---|
| 7B~7Ah | Allocation of PCI-Hole area #2 | 0000h | R/W |

### 7.1.4. 33MHZ HOST BRIDGE PREFETCHABLE ADDRESS BASE & LIMIT

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 7D~7Ch | PCI 33 Prefetchable Memory Address Base | 0000h | R/W |
| 7F~7Eh | PCI 33 Prefetchable Memory Address Limit | 0000h | R/W |

### 7.1.5. TARGET BRIDGE TO DRAM CHARACTERISTICS

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 80h | Target Bridge to DRAM Characteristics | 00h | R/W |

### 7.1.6. 33MHZ HOST BRIDGE & PCI ARBITER

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 81h | Reserved | 00h | R/W |
| 82h | PCI33 Target Bridge Bus Characteristics | 00h | R/W |
| 83h | CPU to PCI33 Characteristics | 00h | R/W |
| 85~84h | PCI33 Grant Timer | FFFFh | R/W |
| 86h | CPU Idle Timer for PCI | FFh | R/W |
| 87h | General Purpose Register I | 00h | R/W |
| 89~88h | Base Address of Frame Buffer Area for fast back-to-back Transaction | 0000h | R/W |
| 8B~8Ah | Size of Frame Buffer Area | 0000h | R/W |

### 7.1.7. DLL CONTROL

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 8Ch | DLL Control | 00h | R/W |
| 8F~8Dh | General Purpose Register II | 0000h | R/W |

### 7.1.8. GART AND PAGE TABLE REGISTERS

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 93~90h | GART Base Address for 1-level Re-mapping | 00000000h | R/W |
| 94h | Graphic Window Control | 00h | R/W |

| | | | |
|---|---|---|---|
| 96~95h | Reserved | | |
| 97h | Page Table Cache Control | 00h | R/W |
| 9B~98h | Invalid Page Table Cache Control | 00000000h | R/W |
| AF~9Ch | Reserved | 00000000h | R/W |

### 7.1.9. A.G.P. AND 66MHZ HOST BRIDGE

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| C3~C0h | A.G.P. Capability Identify Register | 00100002h | RO |
| C7~C4h | A.G.P. Status Register | 1F000203h | RO |
| CB~C8h | A.G.P. Command Register | 00000000h | R/W |
| D3~CCh | Reserved | | |
| D4h | A.G.P. Interface Arbitration Timer (1) | 00h | R/W |
| D5h | A.G.P. Interface Arbitration Timer (2) | 00h | R/W |
| D6h | Data Transfer Counter | 00h | R/W |
| D7h | PCI 33/66 Idle Timer | 00h | R/W |
| D8h | AD_STB timing control | 00h | R/W |
| D9h | Reserved | 0000h | R/W |
| Dah | PCI 33/66 Balance Timer | 00h | R/W |
| DF~DBh | Reserved | | |
| E0h | A.G.P. Compliant Target / Arbiter Control Register. | 00h | R/W |
| E1h | Reserved | 00h | R/W |
| E2h | PCI66 Target Bridge Characteristics | 00h | R/W |
| E3h | CPU to PCI66 Bridge Characteristics | 00h | R/W |

## 7.2. DEVICE 2, FUNCTION 0 (VIRTUAL PCI-TO-PCI BRIDGE)

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 00-01h | Vendor ID | 1039h | RO |
| 02-03h | Device ID | 0001h | RO |
| 04-05h | PCI Command Register | 0000h | RO<br>R/W |

| | | | |
|---|---|---|---|
| 06-07h | PCI Status Register | 0000h | RO |
| 08h | Revision ID | 00h | RO |
| 09h | Programming Interface | 00h | RO |
| 0Ah | Sub-Class Code | 04h | RO |
| 0Bh | Base Class Code | 06h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Master latency timer | 00h | RO |
| 0Eh | Header Type | 01h | RO |
| 0Fh | BIST | 00h | RO |
| 19h | Secondary Bus Number | 00h | R/W |
| 1Ah | Subordinate Bus Number | 00h | R/W |
| 1Bh | Secondary Master latency Timer | 00h | R/W |
| 1Ch | I/O Base | F0h | R/W<br>RO |
| 1Dh | I/O Limit | 00h | R/W<br>RO |
| 1Eh | Secondary PCI-PCI Status | 0000h | R/W<br>RO |
| 20~21h | Non-prefetchable Memory Base Address | FFF0h | R/W<br>RO |
| 22~23h | Non-prefetchable Memory Limit Address | 0000h | R/W<br>RO |
| 24~25h | Prefetchable Memory Base Address | FFF0h | R/W<br>RO |
| 26~27h | Prefetchable Memory Limit Address | 0000h | R/W<br>RO |
| 28~3Dh | Reserved | | |
| 3Eh | PCI to PCI Bridge Control | 0000h | RW<br>RO |

### 7.3. PCI IDE CONFIGURATION SPACE

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 01~00h | Vendor ID | 1039h | RO |
| 03~02h | Device ID | 5513h | RO |
| 05~04h | PCI Command Port | 0000h | RO<br>R/W |
| 07~06h | PCI Status Register | 0000h | RO |
| 08h | Revision ID | 00h | RO |
| 0B~09h | Class Code | 000000h | RO |
| 0Ch | Cache Line Size | 00h | RO |
| 0Dh | Master latency timer | 00h | RO |
| 0Eh | Header Type | 01h | RO |
| 0Fh | BIST | 00h | RO |
| 1F~10h | Primary & Secondary Channel Command /Control Block Base Address Register | | R/W |
| 23~20h | Bus Master IDE Control Register Base Address | | R/W |
| 2B~24h | Reserved | | RO |
| 2Ch | Subsystem ID | | R/W<br>RO |
| 2F~2Dh | Reserved | 00h | R/W<br>RO |
| 33~30h | Expansion ROM Base Address | | |
| 47~40h | IDE Primary /Second Channel, Master/Slave Drive Timing Control | 0000h | R/W<br>RO |
| 48h | IDE Command Recovery Time Control | FFF0h | R/W<br>RO |
| 49h | IDE Command Active Time Control | 0000h | R/W<br>RO |
| 4B~4Ah | IDE General Control Register 0 &1 | | |

| | | | |
|---|---|---|---|
| 4F~4Ch | Prefetch Count of Primary /Secondary Channel | 0000h | RW<br>RO |
| 51~50h | Reserved | | |
| 52h | IDE Miscellaneous Control Register | | |

### 7.3.1. OFFSET REGISTER FOR PCI BUS MASTER IDE CONTROL REGISTERS

| REGISTER ADDRESS | REGISTER NAME | DEFAULT VALUE | ACCESS TYPE |
|---|---|---|---|
| 00h | Bus Master IDE Command Register (Primary) | 00 | R/W |
| 01h | Reserved | 00 | R/W |
| 02h | Bus Master IDE Status Register (Primary ) | 00 | R/W |
| 03h | Reserved | 00 | R/W |
| 04~07h | Bus Master IDE PRD Table Pointer Register (Primary) | 00 | R/W |
| 08h | Bus Master IDE Command Register (Secondary) | 00 | R/W |
| 09h | Reserved | 00 | R/W |
| 0Ah | Bus Master IDE Status Register (Secondary) | 00 | R/W |
| 0Bh | Reserved | 00 | R/W |
| 0C~0Fh | Bus Master IDE PRD Table Pointer Register (Secondary) | 00 | R/W |

# 8. REGISTER DESCRIPTION

The SiS600 contain the programmer visible registers located in the I/O space. These registers are listed in the following.

| I/O SPACE ADDRESS | CONFIGURATION REGISTER FUNCTION |
|---|---|
| 0CF8h | CONFIG_ADDRESS register (only valid for Dword access) |
| 0CFCh | CONFIG_DATA register (only valid if enable bit is set in the CONFIG_ADDRESS register) |

## 8.1. HOST BRIDGE REGISTERS (FUNCTION 0)

### 8.1.1. CONFIGURATION SPACE HEADER

**Register 00h Vendor ID**

Default Value: 1039h

Access: Read Only

The register identifies the manufacturer of the device. SiS is allocated as 1039h by PCI SIG.

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

**Register 02h Device ID**

Default Value: 5600h

Access: Read Only

The device identifier is allocated as 600h by Silicon Integrated Systems Corp.

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

**Register 04h Command**

Default Value: 0005h

Access: Read/Write, Read Only

The Command register provides coarse control over a device's ability to generate and respond to PCI cycles.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:10 | RO | **Reserved** |
| 9 | R/W | **Fast Back-to-Back Enable**<br>This bit controls whether or not the host bridge can do fast back-to-back transactions to different devices. Please note that even when this bit is disabled, the host bridge may still do fast back-to-back transactions if the host bridge can guarantee that the adjacent transactions are destined to the same device. The register that controls fast back-to-back transactions to the same agent is located in Register 83h.<br>0: Disable<br>1: Enable |
| 8 | R/W | **SERR# enable**<br>This bit controls the SERR# driver. When this bit is disabled, SiS600 would not drive SERR# under any condition. When this bit is enabled, SiS600 may drive SERR# in responding to ECC error or the assertion of ASERR# on A.G.P. bus.<br>0: Disable<br>1: Enable |
| 7:3 | RO | **Reserved** |
| 2 | RO | **Bus Master**<br>Always read as 1. |
| 1 | R/W | **Memory Space**<br>This bit controls the response to memory space accesses. When the bit is disabled, the host bridge ignores all access from PCI masters.<br>0: Disable<br>1: Enable |
| 0 | RO | **I/O Space**<br>Default value is 1. The host bridge only respond to the addresses 0CF8h and 0CFCh in the I/O space and the I/O transaction must be generated by the host bridge itself. |

**Register 06h Status**

Default Value: 0210h

Access: Read/Write, Read Only, Write Clear

The Status register is used to record status information for PCI bus related events. Reads to this register behave normally. Writes are slightly different in that each bit in this register can only be reset(Write Clear from 1 to 0), but not set. Each bit will be reset whenever the register is written, and the corresponding bit contains “1”. For instance, to write value 0100_0000_0000_0000b to the register will clear bit 14 and not affect any other bits.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15 | RO | **Detected Parity Error**<br>This bit is always 0, SiS600 does not support parity checking on the PCI bus. |
| 14 | WC | **Signaled System Error**<br>This bit is set whenever SiS600 drives SERR#. It is cleared by writing a 1 to it. |
| 13 | WC | **Received Master Abort**<br>This bit is set when SiS600 terminates a transaction with master abort. This bit is cleared by being written a 1. |
| 12 | WC | **Received Target Abort.**<br>This bit is set when SiS600 terminates a transaction with target abort. This bit is cleared by being written a 1. |
| 11 | RO | **Signaled Target Abort**<br>This bit is always 0 since SiS600 will never respond a transaction with target abort. |
| 10:9 | RO | **DEVSEL# Timing DEVT.**<br>These two bits define the timing to assert DEVSEL#. SiS600 always asserts DEVSEL# within two clocks after the assertion of FRAME#. |
| 8 | R/W | **Reserved** |
| 7:5 | RO | **Reserved** |
| 4 | RO | **CAP_LIST**<br>Read as "1" and it indicates the configuration space implements a list of capabilities. |
| 3:0 | RO | **Reserved** |

**Register 08h Revision ID**

Default Value: 10h

Access: Read Only

The Revision ID is 00h for B0 stepping.

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

**Register 09h Programming Interface**

Default Value: 00h

Access: Read Only

The default value is 00h since no specific register-level programming interface is provided.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Programming Interface** |

**Register 0Ah Sub Class Code**

Default Value: 00h

Access: Read Only

The Sub Class Code is 00h for host bridge.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Sub Class Code** |

**Register 0Bh Base Class Code**

Default Value: 06h

Access: Read Only

The value of 06h in this field identifies a bridge device.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Base Class Code** |

**Register 0Ch Cache Line Size**

Default Value: 00h

Access: Read Only

The value of this register is always 00h since the host bridge would not generate the Memory Write and Invalidate command.

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

**Register 0Dh Master latency Timer (MLT)**

Default Value: FFh

Access: Read/Write

MLT is used in conjunction with PGT(Register84h) and CIT(Register 86h) to provide a fair and efficient system arbitration mechanism. MLT guarantees a minimum system bandwidth for CPU when CPU and PCI masters are competing for system resources (system memory or PCI bus). This bit is valid for both PCI 33/66 buses.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for Master latency Timer**<br>Power-on default value is FFh but the recommended value is 20h. Unit: PCI clock |

**Register 0Eh Header Type**

Default Value: 80h

Access: Read Only

The value of 80h implies that SiS600 is a multiple function device.

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

**Register 0Fh BIST**

Default Value: 00h

Access: Read Only

The value is 00h since SiS600 do not support Build-in Self Test function.

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

**Register 10h Graphic Window Base Address (GWBA)**

Default Value: 00000000h

Access: Read/Write, Read Only

The register defines the starting address of the graphic window for A.G.P. Accessibility and effectiveness of this register is controlled by the Graphic Window Control Register(Register 94h).

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:22 | R/W<br>RO | **Define A[31:22] of Graphic window base address**<br>The accessibility of bits[31:22] are controlled by graphic window size(Bits[6:4], Register 94h). |
| 21:0 | RO | **Reserved and read as 000000h** |

| Bit31 | Bit30 | Bit29 | Bit28 | Bit27 | Bit26 | Bit25 | Bit24 | Bit23 | Bit22 | Size |
|---|---|---|---|---|---|---|---|---|---|---|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 4M |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 0 | 8M |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 0 | 0 | 16M |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | 0 | 0 | 0 | 32M |
| R/W | R/W | R/W | R/W | R/W | R/W | 0 | 0 | 0 | 0 | 64M |
| R/W | R/W | R/W | R/W | R/W | 0 | 0 | 0 | 0 | 0 | 128M |
| R/W | R/W | R/W | R/W | 0 | 0 | 0 | 0 | 0 | 0 | 256M |

**Register 14~33h Reserved**

**Register 34h Capability Pointer (CAPPTR)**

Default Value: C0h

Access: Read Only

The value of C0h indicates that the A.G.P. standard register block is located at Register C0h.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Capability Pointer**<br>Pointer to the starting location of A.G.P. standard register block. |

### 8.1.2. HOST CONTROL REGISTERS

**Register 50h Host Bus Interface control I**

default Value: 02h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **Reserved** |
| 3 | R/W | **CPU-to-Memory and PCI-to-Memory Concurrency Enable**<br>When enabled, CPU-to-Memory cycles and PCI-to-Memory cycles can take place concurrently on the host bus and PCI bus. These two kinds of memory access cycles will then be rearranged on the host bus and take place on the memory bus sequentially. Bus utilization on both the host bus and PCI bus can be improved. When disabled, the two kinds of cycles can not occur on CPU or PCI concurrently.<br>0: Disable<br>1: Enable |
| 2 | R/W | **CPU-to-PCI and PCI-to Masters Concurrently Access PCI Bus Function**<br>When enabled, CPU-to-PCI cycles and PCI-to-Memory can take place concurrently on the host bus and PCI bus respectively, such that their respective cycles will be forwarded to PCI bus and memory bus at the same time. This bit is valid only when bit 3 is set. When disabled, the two kinds of cycles can not occur on the host bus and PCI bus concurrently.<br>0: Disable<br>1: Enable |
| 1 | R/W | **CPU Pipeline Function**<br>When this bit is 0, only one pending cycle is allowed at one time. When this bit is 1, more than two pending cycles at one time are allowed subject to CPU's behavior.<br>0: no pipeline<br>1: pipeline enable |

| | | |
|---|---|---|
| 0 | R/W | **PCI 33/PCI 66 Masters Concurrently Access Memory Function**<br>When enabled, PCI33 and PCI66 masters requesting for memory access cycles will be granted simultaneously, such that their memory access cycle can be optimized. When disabled, only one of PCI33 and PCI66 masters can be granted. In case their requests are asserted at the same time, PCI66 masters have higher priority than PCI33.<br>0: Disabled<br>1: Enabled |

**Register 51h Host Bus Interface control II**

default Value: 02h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **Reserved** |
| 5 | R/W | **Support Asynchronous Clock Mode between Host and PCI clock**<br>0: Disable<br>1: Enable |
| 4 | R/W | **Reserved** |
| 3 | R/W | **Host to PCI33/66 concurrency Control**<br>0: Enable<br>1: Disable |
| 2 | R/W | **Host-to-SDRAM Pipelined Read Timing Control**<br>When set to 1, there will be no wait state between SDRAM back-to-back read cycles; when set to 0, there will be one wait state inserted in between.<br>0: x-1-1-1-2-1-1-1.....<br>1: x-1-1-1-1-1-1-1..... |
| 1 | R/W | **Host-to-PCI Cycle Timing Control**<br>When set to 1, the Host-to-PCI bridge will start the translation of Host-to-PCI cycle once the cycle appears on the host bus. When set to 0, the start of the Host-to-PCI translation will be delayed by one CPU clock.<br>0: Delay 1 CPU clock<br>1: Without Delay |

| | | |
|---|---|---|
| 0 | R/W | **Host-to-Memory Cycle Lead-off Time Control**<br><br>When set to 1, the host controller will start the translation of Host-to-Memory access cycle once the cycle appears on the host bus. The resulting Lead-off time for memory access cycle will be 8T. When set to 0, the start of translation will be delayed one CPU clock and the resulting Lead-off time is 9T.<br><br>0: Lead-off time = 9T<br><br>1: Lead-off time = 8T |

### 8.1.3. DRAM CONTROL REGISTERS

**Register 52h DRAM MISC Control 1**

default Value: 00h

Access: Read/Write

The register defines timing for Refresh cycles.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Test bit for DRAM Controller**<br>0: Normal Mode<br>1: Test Mode |
| 6:5 | R/W | **Reserved** |
| 4 | R/W | **Multi-function (Pin L1) for WEB#/MA14 Selection**<br>0: WEB#<br>1: MA14 |
| 3 | R/W | **Reserved** |
| 2 | R/W | **SDRAM Refresh Command to Different ROW Control**<br>When set to 0, refresh commands to different SDRAM rows initiated by SiS600 will be staggered one clock apart, such that the simultaneous-switching noise can be reduced, i.e. CS[5:0]# will be asserted one after another. When set to 1, SiS600 will issue refresh commands to different SDRAM rows simultaneously.<br>0: Staggered one clock apart<br>1: Simultaneous |
| 1 | R/W | **DRAM Refresh Test Mode**<br>This bit is a test mode bit used for internal refresh circuit testing.<br>0: Normal Mode<br>1: Test Mode |
| 0 | R/W | **CS# Behavior when SDRAM Refresh Cycles**<br>This bit controls the behavior of CS# for SDRAM during refresh cycles. If there is any SDRAM populated in the system, this bit should be programmed as 1.<br>0:Long pulse (CS# Behaves like RAS# during EDO/FP mode refresh cycles)<br>1:1T command pulse |

**Register 53h DRAM Timing Control**

default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **Implemented DRAM performance Control**<br>**Bits[7:6] Insert Wait State**<br>00 1 Wait State<br>01 2 Wait States<br>10 4 Wait States<br>11 8 Wait States |
| 5:3 | R/W | **Reserved** |
| 2:1 | R/W | **Refresh Queue Depth**<br>These two bits control the depth of refresh queue. To minimize the impact to performance caused by refresh cycles, the concept of refresh queue is introduced. Refresh request is arbitrated with other DRAM request, if a refresh request does not get serviced, it enters the refresh queue. The priority of refresh request will be promoted to highest when the refresh queue is full.<br>**Bits[2:1] Depth**<br>00 0<br>01 4<br>10 8<br>11 12 |
| 0 | R/W | **Improved the DRAM performance for ECC function**<br>0: Disabled<br>1: Enabled |

**Register 54h DRAM RAS#/CS# Timing Control**

default Value: 00h

Access: Read/Write

This register controls the RAS#/CS# timing for EDO, FP DRAM and SDRAM.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **RAS Pulse Width**<br>Bits[7:6] defines the RAS# pulse width for refresh cycles<br>**Bits[7:6] Pulse Width**<br>00 4T<br>01 5T<br>10 6T<br>11 7T |
| 5:4 | R/W | **RAS# Precharge Time**<br>**Bits[5:4] Description**<br>00 2T<br>01 3T<br>10 4T<br>11 5T |
| 3:2 | R/W | **RAS to CAS Delay**<br>**Bits[5:4] Description**<br>00 2T<br>01 3T<br>10 4T<br>11 5T |
| 1:0 | R/W | **Reserved** |

**Register 55h DRAM MISC Control 2**

default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **RAMW# Assertion Timing for EDO/FP Mode DRAM**<br>This bit is only valid for EDO/FP DRAM.<br>0: Normal<br>1: Faster |
| 6 | R/W | **EDO Detection Bit for BIOS**<br>This bit is used by BIOS to detect whether the DRAM type is EDO or FP DRAM. When set to 1, SiS600 will return BRDY# about $2^{12}$ CPU clocks later. In such case, BIOS can read correct data for EDO since data will still be valid on MD bus. For FP, data will be wrong. After BIOS finishes DRAM detection procedure, this bit should be programmed to "0".<br>0: Disable<br>1: Enable |
| 5 | R/W | **Turn Around Time Control between SDRAM Read and Write Cycles**<br>This bit is used to control memory data bus (MD) turn around time between two consecutive SDRAM read and write cycles. The turn around time in normal condition should be 1T. For certain SDRAMs that will not halt the driving of MD in time, the turn around time will be 2T to avoid contention.<br>0: 1T<br>1: 2T |
| 4 | R/W | **SDRAM RAS# Precharge Time Control**<br>When this bit is enabled, SiS600 will optimize the RAS# precharge time by comparing the memory address of every consecutive DRAM cycles to check whether the addresses are located at the same row or not. If the cycles are destined to the same row, the minimum RAS# precharge time for SDRAM should be met before issuing a row active command. However, if they are destined to the different rows, the RAS# precharge time is met automatically, and row active command can be issued immediately.<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 3 | R/W | **Memory Data Output Timing Control for EDO DRAM**<br>This bit is used to control the timing to drive memory data (MD) onto MD bus during EDO write memory cycles. When heavy loading EDO memory is used which will let RAMW# control signal delay more than 1 clock, the 1T delay control bit will avoid MD bus contention.<br>0: Normal<br>1: Delay 1T |
| 2 | R/W | **Reserved** |
| 1 | R/W | **Lead-off Time Control for DRAM Read Cycles**<br>0:Normal<br>1:Slow |
| 0 | R/W | **Lead-off Time Control for DRAM Row Start Cycles**<br>0:Normal<br>1:Slow |

**Register 56h DRAM MISC Control 3**

default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **MDLE Delay Control for EDO/FP Mode DRAM**<br>These bits control the timing for internal latch signal MDLE, which is used to latch MD after DRAM stops driving the MD line to guarantee DRAM masters will read correct data from MD bus. The value of delay time depends on the specification and the load of DRAM.<br>**Bits[7:4] Delay**<br>0000 0ns<br>0001 1ns<br>0010 2ns<br>0011 3ns<br>0100 4ns<br>0101 5ns<br>0110 6ns<br>0111 7ns<br>1000 8ns<br>1001 9ns<br>1010 10ns<br>1011 11ns<br>1100 12ns<br>1101 13ns<br>1110 14ns<br>1111 15ns |
| 3 | R/W | **SDRAM Initialization Mode Selection**<br>This bit controls whether the command specified in register 57h bits[7:5] should be applied to one row only or to all rows. If this bit is 0, the DRAM Status Register (Register 63h) will be to specify which row the command should be applied to.<br>0: One Row<br>1: All Rows |
| 2 | R/W | **Reserved** |

| | | |
|---|---|---|
| 1:0 | R/W | **Control Bits for SDRAM Read Clock Adjustment**<br>These control bits are used to adjust the internal SDRAM clock used to latch MD[63:0] driven out by SDRAM. The values being programmed will be subject to SDRAM tAC specification.<br>00: Normal<br>01: Delay 0.5ns<br>10: Delay 1.0ns<br>11: Delay 1.5ns |

**Register 57h SDRAM Control**

default Value: 00h

Access: Read/Write

This register controls SDRAM initialization process and timing.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Precharge Command**<br>When set, SiS600 will issue precharge command to SDRAM. This bit is automatically cleared after the precharge command is completed.<br>0: Disable<br>1: Enable |
| 6 | R/W | **Mode Register Set Command**<br>When set, SiS600 will issue mode register setting command to SDRAM. This bit is automatically cleared after the mode register setting command is completed.<br>0: Disable<br>1: Enable |
| 5 | R/W | **Refresh Command**<br>When set, SiS600 will issue refresh command to SDRAM. This bit is automatically cleared after the refresh command is completed.<br>0: Disable<br>1: Enable |
| 4 | R/W | **CAS# Latency (CL) Setting**<br>This bit contains the information for SDRAM during initialization.<br>0: 2T<br>1: 3T |

| | | |
|---|---|---|
| 3 | R/W | **SDRAM Write Retire Rate**<br>This bit controls the timing that SiS600 writes data into SDRAM during burst cycles.<br>0: X-2-2-2<br>1: X-1-1-1 |
| 2 | R/W | **Precharge Command Timing Control.**<br>This bit controls the timing for asserting precharge command when the address of the next memory access cycle is located at different page than that of the current one.<br>0: One wait state<br>1: Zero wait state |
| 1 | R/W | **SDRAM Multi-bank Function control**<br>When enable, SiS600 supports SDRAM internal multi-bank function up to 4 banks.<br>0: Disable<br>1: Enable |
| 0 | R/W | **NOP Command**<br>When set, SiS600 will issue NOP command to SDRAM. This bit is automatically cleared after the NOP command is completed.<br>0: Disable<br>1: Enable |

**Register 58h EDO/FP DRAM CAS# Timing Control**

default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **CAS# Precharge Time for FP DRAM**<br>**Bits[7:6] Description**<br>00 1T<br>01 1T during burst cycles, 2T for different cycles.<br>(One wait state between cycles)<br>10 2T<br>11 Reserved |

| | | |
|---|---|---|
| 5:4 | R/W | **CAS# Precharge Time for EDO DRAM**<br>**Bits[7:6] Description**<br>00 1T<br>01 1T during burst cycles, 2T for different cycles.<br>(One wait state between cycles)<br>10 2T<br>11 Reserved |
| 3:2 | R/W | **CAS# Pulse Width for FP DRAM**<br>**Bits[3:2] Description**<br>00 3T<br>01 2T<br>10 1T<br>11 Reserved |
| 1 | R/W | **CAS# Pulse Width for EDO DRAM**<br>0: 2T<br>1: 1T |
| 0 | R/W | **MD to HD Timing Control for EDO/FP Mode DRAM**<br>This bit controls the timing when SiS600 drives the data read from DRAM(MD) onto HD for CPU memory read cycles. This setting is only effective when CPU read FIFO is empty and ECC function is disabled. When set to 0, SiS600 will drive MD onto Host data bus immediately after MD from DRAM is being latched internally. For some DRAM with slower timing or heavier loading, MD valid time will be lagging behind and this bit should be set to 1 to insert one wait state.<br>0: Zero wait state<br>1: One wait state |

**Register 59h A.G.P./PCI Buffer Strength and Current Rating**

Default Value : 00h

Access: Read/Write

This register controls the buffer strength of A.G.P. and PCI bus related signals.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **Reserved** |
| 3:2 | R/W | **A.G.P. Current Rating**<br>This field specifies the buffer strength of all signals on the A.G.P.<br>**Bit[3:2] Buffer Strength**<br>00 3mA<br>01 2mA<br>10 2mA<br>11 4mA |
| 1 | R/W | **AD[31:0] Current Rating**<br>This bit controls the buffer strength of AD[31:0] on PCI bus.<br>0: 4mA<br>1: 8mA |
| 0 | R/W | **PCI Control Signals Current Rating**<br>This bit controls the buffer strength of FRAME#, IRDY#, TRDY#, DEVSEL#, STOP#, PAR, C/BE[3:0]# and PGNT[3:0]#.<br>0: 4mA<br>1: 8mA |

**Register 5Ah Memory Buffer Strength and Current Rating**

Default Value: 00h

Access: Read/Write

This register controls the buffer strength of DRAM related signals.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **RAS[5:0]#/CS[5:0]# Driving Rating**<br>0: 8mA<br>1: 12mA |

| | | |
|---|---|---|
| 6 | R/W | **CAS[7:0]#/DQM[7:0]# Driving Rating**<br>0: 8mA<br>1: 16mA |
| 5 | R/W | **CKE Driving Rating**<br>0: 12mA<br>1: 16mA |
| 4 | R/W | **MA[13:2] Driving Rating**<br>0: 12mA<br>1: 16mA |
| 3 | R/W | **MA[1:0] Driving Rating**<br>0: 12mA<br>1: 16mA |
| 2 | R/W | **RAMWA# Driving Rating**<br>0: 12mA<br>1: 16mA |
| 1 | R/W | **RAMWB# Driving Rating**<br>0: 12mA<br>1: 16mA |
| 0 | R/W | **SRAS#/SCAS# Driving Rating**<br>0: 12mA<br>1: 16mA |

**Register 5Bh~5Fh Reserved**

**Register 60h/61h/62h for bank x=0..2 DRAM Type Registers**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **DRAM Mode Selection**<br>00: FP Mode,<br>01: EDO Mode<br>10: Reserved<br>11: SDRAM |

| | | |
|---|---|---|
| 5 | R/W | **DRAM Configuration Selection**<br>0: Single sided<br>1: Double sided |
| 4 | R/W | **Reserved** |
| | | **RAM Type Selection**<br>**FPM/EDO DRAM**<br>0000: 256K Symmetric 9x9 0001: 1M Symmetric 10x10<br>0010: 4M Symmetric 11x11 0011: 16M Symmetric 12x12<br>0100: 12x8 1M Asymmetric 0101: 12x9 2M Asymmetric<br>0110: 12x10 4M Asymmetric 0111: 12x11 8M Asymmetric<br>1000: 10x9 512K Asymmetric 1001: 11x9 1M Asymmetric<br>1010: 11x10 2M Asymmetric Others: Reserved<br>**SDRAM (NBAxNRAxNCA)**<br>0000: 1x11x8(1M) 0001: 1x13x8(4M)<br>0010: 2x12x8(4M) 0011: 2x13x8(8M)<br>0100: 1x11x9(2M) 0101: 1x13x9(8M)<br>0110: 2x12x9(8M) 0111: 2x13x9(16M)<br>1000: 1x11x10(4M) 1001: 1x13x10(16M)<br>1010: 2x12x10(16M) 1011: 2x13x10(32M)<br>1100: 2x11x8(2M) Others: Reserved |

**Register 63h DRAM Status Register (bit-x =bank-x)**

Default Value : FFh

Access: Read/Write

This register is used to specify which DRAM banks are populated with DRAM.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | R/W | **Reserved** |
| 2:0 | R/W | **DRAM Status**<br>0: Absent<br>1: Installed |

**Register 64~67h ECC Status Register**

Default Value: 00000000h

Access: Read/Write

This register contains results reported by ECC function.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:16 | R/W | **ECC Error Specific Location**<br>ECC Parity Error Address [31:16] |
| 15:12 | R/W | **ECC Error Specific Location**<br>This 36-bit register specifies the specific address information where ECC occurs.<br>ECC Parity Error Address [15:12] |
| 11:8 | R/W | **Reserved** |
| 7:5 | R/W | **ECC Error Location**<br>The ECC function supported by SiS600 is on per-row basis. When an ECC error is being detected, these two bits indicate the specific row on which the error occurs.<br>Bits[7:5] Description<br>000 ECC error occurs in row 0<br>001 ECC error occurs in row 1<br>010 ECC error occurs in row 2<br>011 ECC error occurs in row 3<br>100 ECC error occurs in row 4<br>101 ECC error occurs in row 5<br>Others Reserved |
| 4 | R/W | **Unrecoverable ECC Error Status Bit**<br>When this bit is 1, there is unrecoverable ECC error being detected. |
| 3 | R/W | **ECC 1-Bit Recoverable Error Status Bit**<br>When this bit is 1, there is recoverable 1-bit ECC error being detected. |
| 2:0 | R/W | **Reserved** |

### 8.1.4. POWER MANAGEMENT

**Register 68h~69h ACPI I/O Space Base Address Register**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:5 | R/W | **A[15:5] for ACPI I/O Space base Address**<br>This register specifies A[15:5] of the starting address of the ACPI I/O space. |
| 4:3 | R/W | **Reserved** |
| 2 | R/W | **A.G.P. Request Enable**<br>This bit controls the A.G.P. request during ACPI cycles. When disabled, no A.G.P. request is acceptable during ACPI cycles.<br>0: Disable<br>1: Enable |
| 1 | R/W | **Reserved** |
| 0 | R/W | **Validity Bit**<br>When set to 1, the base address contain in Bit[15:5] in valid. Otherwise the base address defined in Bit[15:5] is ignored.<br>0: Invalid<br>1: Valid |

**Register 6Ah SMRAM Access Control Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **SMRAM Area Re-mapping Control**<br>This field controls how the address in the host bus is mapped to the system memory address when the SMARM access control bit is enabled or CPU is in the system management mode.<br>**Bits[7:6] Host Address System Memory Address**<br>00 E0000h~E7FFFh E0000h~E7FFFh (32K)<br>01 E0000h~E7FFFh A0000h~A7FFFh (32K)<br>10 E0000h~E7FFFh B0000h~B7FFFh (32K)<br>11 A0000h~AFFFFh A0000h~AFFFFh (64K) |
| 5 | R/W | **Reserved** |

| | | |
|---|---|---|
| 4 | R/W | **SMRAM Access Control**<br>When enabled, SMRAM area can be accessed without asserting SMIACT#. This function is useful for BIOS to initialize SMRAM. When disabled, SMRAM area can only be accessed during the SMI handler.<br>0: Disable<br>1: Enable |
| 3:0 | R/W | **Reserved** |

**Register 6Bh System Event Monitor Control for Power Management**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Monitoring A.G.P. I/O Access**<br>When enabled, CPU-to-AGP I/O accesses fall within the range defined by base address register will be reported to the SiS5595 via BM_REQ# .<br>0: Disable<br>1: Enable |
| 6 | R/W | **Monitoring A.G.P. Non-prefetchable Memory Access**<br>When enabled, CPU-to-AGP memory accesses fall within the non-prefetchable memory area will be reported to the SiS5595 via BM_REQ#.<br>0: Disable<br>1: Enable |
| 5 | R/W | **Monitoring A.G.P. Prefetchable Memory Access**<br>When enabled, CPU-to-AGP memory accesses fall within the prefetchable memory area will be reported to the SiS5595 via BM_REQ#.<br>0: Disable<br>1: Enable |
| 4 | R/W | **Monitoring VGA Compatible IO Access toward A.G.P.**<br>When enabled, I/O accesses fall within VGA compatible I/O addresses (3B0h ~ 3BBh, 3C0 ~ 3DFh) toward AGP will be reported to the SiS5595 via BM_REQ#.<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 3 | R/W | **Monitoring VGA Compatible Memory Access toward A.G.P.**<br>When enabled, memory accesses fall within VGA compatible memory region (A0000h~BFFFFh) toward A.G.P. will be reported to the SiS5595 via BM_REQ#.<br>0: Disable<br>1: Enable |
| 2 | R/W | **Monitoring A.G.P. Bus Master Activity**<br>When this bit is enabled, any A.G.P. bus master activity will be reported to the SiS5595 via BM_REQ#. When this bit is disable, No bus master activity will be reported.<br>0: Disable<br>1: Enable |
| 1:0 | R/W | **Reserved** |

**Register 6Ch DRAM Self-Refresh Control for Power Management**

Default Value: 00h

Access: Read/Write

This register controls ACPI sleep states supported by SiS600 and CKE behavior.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **ACPI S3 State Support**<br>0: Disabled<br>1: Enabled |
| 6 | R/W | **ACPI S2 State Support**<br>0: Disabled<br>1: Enabled |
| 5 | R/W | **CKE Output Enable Control**<br>When enabled, SiS600 drives CKE. When disabled, SiS600 floats its CKE output.<br>0: Disabled<br>1: Enabled |
| 4 | R/W | **CKE Selection**<br>This bit is controlled by BIOS during power management mode, and is only valid when the CKE Output Enable Control bit is enabled. When set to 1, SiS600 always drives CKE to low. When set to 0, SiS600 drives CKE to low only when it enters self-refresh mode (S2 or S3 state and stop grant cycle issued).<br>0: Normal Mode<br>1: Force Low |

| | | |
|---|---|---|
| 3:0 | R/W | **CKE Timing Control**<br>These bits control the timing of CKE. When the value of this field is 0000, CKE is driven out from flip-flop, otherwise, it is driven out from combinatorial logic. Various delay options are provided to ensure that CKE can meet SDRAM setup time and hold time specification when CKE is driven out from combinatorial logic.<br>**Bits Description**<br>0000 Flip-flop Output<br>0001 Delay 2ns<br>0010 Delay 3ns<br>0011 Delay 4ns<br>0100 Delay 5ns<br>0101 Delay 6ns<br>0110 Delay 7ns<br>0111 Delay 8ns<br>1000 Delay 9ns<br>1001 Delay 10ns<br>1010 Delay 11ns<br>1011 Delay 12ns<br>1100 Delay 13ns<br>1101 Delay 14ns<br>1110 Delay 1ns<br>1111 No Delay |

### 8.1.5. SHADOW RAM AREA

**Register 70h~71h Shadow RAM Read Attribute Control**

Default Value: 000000h

Access: Read/Write

Register 70h and register 71h define the attribute of the Shadow RAM from 640 KByte to 1 MByte.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15 | R/W | **Shadow RAM Enable for PCI Master Access**<br>When enable, shadow RAM area can be accessed from PCI masters.<br>0: Disable<br>1: Enable |
| 14:13 | R/W | **Reserved** |

| | | |
|---|---|---|
| 12:0 | R/W | **Bit Shadow RAM Segment**<br>0 0C0000h~0C3FFFh<br>1 0C4000h~0C7FFFh<br>2 0C8000h~0CBFFFh<br>3 0CC000h~0CFFFFh<br>4 0D0000h~0D3FFFh<br>5 0D4000h~0D7FFFh<br>6 0D8000h~0DBFFFh<br>7 0DC000h~0DFFFFh<br>8 0E0000h~0E3FFFh<br>9 0E4000h~0E7FFFh<br>10 0E8000h~0EBFFFh<br>11 0EC000h~0EFFFFh<br>12 0F0000h~0FFFFFh<br>0: Read disable<br>1 : Read enable |

**Register 72h~73h Shadow RAM Write Attribute Control**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:13 | R/W | **Reserved** |

| | | |
|---|---|---|
| 12:0 | R/W | Bit Shadow RAM Segment<br>0 0C0000h~0C3FFFh<br>1 0C4000h~0C7FFFh<br>2 0C8000h~0CBFFFh<br>3 0CC000h~0CFFFFh<br>4 0D0000h~0D3FFFh<br>5 0D4000h~0D7FFFh<br>6 0D8000h~0DBFFFh<br>7 0DC000h~0DFFFFh<br>8 0E0000h~0E3FFFh<br>9 0E4000h~0E7FFFh<br>10 0E8000h~0EBFFFh<br>11 0EC000h~0EFFFFh<br>12 0F0000h~0FFFFFh<br>0: Write disable<br>1: Write enable |

### 8.1.6. ECC FUNCTION CONTROL REGISTER

**Register 74h ECC Function Control I**

Default Value: 00h

Access: Read/Write

These registers control the ECC function supported by SiS600.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **ECC Initial Control**<br>When enabled, host memory write cycles will write host data to memory directly without ECC correction, even when the ECC function is enabled which is defined in bit[5:0]. This function is used by BIOS during its initialization procedure.<br>0: Disable<br>1: Enable |
| 6 | R/W | **ECC Test Mode**<br>When enabled, SiS600 may check the internal ECC function. This bit must work together with Register 75h bit 7.<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 5:0 | R/W | **ECC Function Control**<br>The ECC function supported by SiS600 is on a per-row basis. Bits[5:0] provide 5 enable bits for each of the six DRAM row respectively.<br>**Bit Description**<br>5 row5<br>4 row4<br>3 row3<br>2 row2<br>1 row1<br>0 row0<br>0: Disable<br>1: Enable |

**Register 75h ECC Function Control II**

Default Value 00h

Access Read/Write

These registers control the ECC function supported by SiS600.

| | | |
|---|---|---|
| 7 | R/W | **ECC Test Data for Test Mode**<br>This bit provides test data for SiS600 to check the internal ECC function during ECC test mode.<br>0: Internal 64-bit data return "00000000"<br>1: Internal 64-bit data return "FFFFFFFF" |
| 6 | R/W | **Reserved** |
| 5 | R/W | **BRDY# Assertion Timing for EDO Read Cycles**<br>This bit is used to control BRDY# assertion time during host masters read EDO memory cycles. For EDO modules with lighter CAS# loading, this bit can be set to 0, such that BRDY# will be asserted immediately after CAS# signals are issued. The resultant lead-off time is 7T. When this bit is 1, there is one wait state inserted such that the lead-off time will be 8T.<br>0: Lead-off time 7T<br>1: Lead-off time 8T |

| | | |
|---|---|---|
| 4 | R/W | **Refresh Cycle Enable**<br>When disabled, the normal refresh cycle issued from SiS600 will be disabled. This function is used by BIOS to perform SDRAM initialization, during which period SDRAM can still be refreshed by programming register 57h bit 5. For normal operation, this bit should be programmed with 1.<br>0: Disable<br>1: Enable |
| 3:0 | R/W | **Reserved** |

**Register 76h Reserved**

### 8.1.7. PCI HOLE AREA

**Register 77h Characteristics of PCI-Hole Area**

Default Value: 00h

Access: Read/Write

This register controls the PCI Hole area support.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | R/W | **Reserved** |
| 2 | R/W | **PCI-Hole Area I Enable**<br>0: Disable<br>1: Enable |
| 1 | R/W | **Reserved** |
| 0 | R/W | **PCI-Hole Area II Enable**<br>0: Disable<br>1: Enable |

**Register 78~79h Allocation of PCI-Hole Area #1**

Default Value: 0000h

Access: Read/Write

Register 78h and 79h define the size and the base address of the first PCI-Hole area.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:13 | R/W | **Size of PCI-Hole Area I (within 512 Mbytes)**<br>**Bits[15:13] Size**<br>000 64KB<br>001 128KB<br>010 256KB<br>011 512KB<br>100 1MB<br>101 2MB<br>110 4MB<br>111 8MB |
| 12:0 | R/W | **Base Address of PCI-Hole Area I**<br>This field specifies A[28:16] for the base address of the PCI-Hole area. |

**Register 7A~7Bh Allocation of PCI-Hole Area #1I**

Default Value 0000h

Access Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:13 | R/W | **Size of PCI-Hole Area II (within 512 Mbytes)**<br>**Bits[15:13] Size**<br>000 64KB<br>001 128KB<br>010 256KB<br>011 512KB<br>100 1MB<br>101 2MB<br>110 4MB<br>111 8MB |
| 12:0 | R/W | **Base Address of PCI-Hole Area II**<br>This field specifies A[28:16] for the base address of the PCI-Hole area. |

### 8.1.8. PCI33 PREFETCHABLE FUNCTION

**Register 7Ch~7Dh 33Mhz Host Bridge Prefetchable Address Base**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **PCI 33 Prefetchable Memory Address Base** |
| 3:0 | R/W | **Reserved** |

**Register 7Eh~7Fh 33Mhz Host Bridge Prefetchable Address Limit**

Default Value 00h

Access Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **PCI 33 Prefetchable Memory Address Limit** |
| 3:0 | R/W | **Reserved** |

### 8.1.9. TARGET BRIDGE CHARACTERISTICS

**Register 80h Target Bridge to DRAM Characteristics**

Default Value 0000h

Access Read/Write

This register controls the characteristics for 33Mhz PCI target bridge to access DRAM.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | R/W | **Address Boundary Alignment for PCI Bursting**<br>This field controls the alignment of address boundaries. PCI burst cycles generated by PCI masters can not across an address boundary defined by this field. Otherwise, the cycle will be terminated with disconnect.<br>**Bits[7:5] Boundary Alignment**<br>000 256 Bytes<br>001 512 Bytes<br>010 1K Bytes<br>011 2K Bytes<br>100 4K Bytes<br>Others Reserved |
| 4:2 | R/W | **Reserved** |
| 1 | R/W | **PCI Peer Concurrency**<br>When enabled, CPU-to-DRAM and PCI-to-PCI concurrency is allowed.<br>0: Disable<br>1: Enable |
| 0 | R/W | **Reserved** |

**Register 81h Reserved**

**Register 82h Target Bridge Characteristics**

Default Value: 00h

Access : Read/Write

This register controls the characteristics for 33Mhz PCI target bridge.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Improved the performance for PCI Master Write Cycle**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 6 | R/W | **PCI 33/66 Master Read Cycle followed by Master Write Cycle Control**<br>When disabled, PCI master initiated write cycle will not take place on the destination until PCI read FIFO is empty. When enabled, PCI master initiated write cycles will be forwarded to the destination immediately after the previous PCI read cycle, regardless of the status of PCI read FIFO.<br>0: Disable<br>1: Enable |
| 5 | R/W | **PCI33 Memory Read Line or Memory Read Multiple Command Prefetch Enable**<br>This bit enables data prefetching for Memory Read Line or Memory Read Multiple commands.<br>0: Disable<br>1: Enable |
| 4 | R/W | **PCI33 Memory Read Command Prefetch Enable**<br>This bit enables data prefetching for Memory Read command.<br>0: Disable<br>1: Enable |
| 3:2 | R/W | **Initial Latency Control**<br>This field controls the target initial latency of the 33Mhz target bridge. If SiS600 is unable to assert TRDY# for a transaction within the target initial latency defined by this field, SiS600 asserts STOP# to retry this cycle.<br>00: Disable<br>01: 16 PCI33 Clocks<br>10: 24 PCI33 Clocks<br>11: 32 PCI33 Clocks |
| 1 | R/W | **Subsequent Latency Control**<br>When enabled, SiS600 terminates a transaction with STOP# if it is unable to assert TRDY# for subsequent data transfer within 8 clocks.<br>0: Disable<br>1: Enable |
| 0 | R/W | **Reserved** |

**Register 83h CPU to PCI33 Characteristics and Arbitration Option**

Default Value: 00h

Access : Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | R/W | **Reserved** |
| 4 | R/W | **Fast Back-to-Back to Same Agent Control.**<br>When enabled, SiS600 enables fast back-to-back timing for consecutive IDE data port write cycles and consecutive CPU-to-PCI33 frame buffer memory write cycles. The base address and the size of the frame buffer area are defined by Register 88h and Register 8Ah. Register 88h and Register 8Ah must be initialized before this control bit is enabled.<br>0: Disable<br>1: Enable |
| 3 | R/W | **CPU Involved Arbitration on PCI**<br>PGT(Register 84h), CIT(Register 86h) and MLT(Register 0Dh) are effective only when this bit is enable. When enabled, SiS600 does not block CPU from operation longer than the period defined by PGT to serve PCI masters, and minimum access time for CPU is guaranteed by MLT.<br>0: Disable<br>1: Enable |
| 2 | R/W | **Non-Post Cycle Retry Behavior Control**<br>When enabled, CPU-to-PCI non-post cycle being retried by PCI target will be regenerated once, before SiS600 backs off CPU. When disabled, SiS600 will back off CPU immediately when the current CPU-to-PCI non-post cycle is being retried.<br>0: Disable<br>1: Enable |
| 1 | R/W | **Memory Burst Control**<br>This bit enables the host bridge to generate memory burst cycles.<br>0: Disable<br>1: Enable |
| 0 | R/W | **Memory Post Write Control**<br>When enabled, all CPU to PCI memory write cycles are posted.<br>0: Disable<br>1: Enable |

**Register 84~85h PCI 33/66 Grant Timer**

Default Value: FFFFh

Access: Read/Write

The timer is used to prevent PCI masters from seizing the PCI bus too long. When the timer expires, PCI arbiter forces the master that is currently occupying PCI bus to relinquish PCI bus by removing its grant.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **PCI 33/66 Grant Timer**<br>The setting of this register and MLT (register 1Dh) should be determined from the perspective of overall system performance. For a system with many PCI master devices, the value should be higher. For a system with fewer master devices, the value should be smaller. Typical value of this timer is 60h if MLT is set to 20h.<br>Unit: PCI clock |

**Register 86h CPU Idle Timer for PCI**

Default Value: FFh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **CPU Idle Timer**<br>Recommended value for this timer is 03h.<br>Unit: PCI clock |

**Register 87h General Purpose Register I**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **General Purpose Register**<br>This register provides the storage information for BIOS programming. |

**Register88h~89h Base address of fast back-to-back area**

Default Value: 0000h

Access: Read/Write

PCI cycles fall within this area will be carried out with fast back-to-back cycles, provided PCI spec is not violated.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Frame Buffer Base Address**<br>Bits[15:4] correspond to A[31:20] of the base address. The frame buffer base address must be 1MByte aligned. |
| 3:0 | R/W | **Reserved** |

**Register 8Ah Size of Frame Buffer Area**

Default Value: 0000h

Access: Read/Write

This register defines the size of the frame buffer area where fast back-to-back cycles can be applied.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Size of the frame buffer area**<br>**Bits[15:4] Size**<br>000000000000b 4GB<br>100000000000b 2GB<br>110000000000b 1GB<br>111000000000b 512MB<br>111100000000b 256MB<br>111110000000b 128MB<br>111111000000b 64MB<br>111111100000b 32MB<br>111111110000b 16MB<br>111111111000b 8MB<br>111111111100b 4MB<br>111111111110b 2MB<br>111111111111b 1MB |
| 3:0 | R/W | **Reserved** |

### 8.1.10. CPU/PCI CLOCKS DLL CONTROL REGISTERS

**Register 8Ch DLL Control**

Default Value: 00h

Access: Read/Write

DLL is used to minimize the clock skew between internal and external clocks. This register controls the time period that DLL circuit locks the reference clock again. This register applies to CPU clock and PCI clock.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **DLL Lock Period Control**<br>This field controls the period that DLL adjust the internal clock to the reference clock source.<br>**Bit[7:6] Period**<br>00 4T<br>01 8T<br>10 16T<br>11 32T<br>T is the reference clock source for DLL. |
| 5:3 | R/W | **Reserved** |
| 2 | R/W | **Test bit for internal clock**<br>0: Normal Mode<br>1: Test Mode |
| 1 | R/W | **Test bit for DRAM clock**<br>0: Normal Mode<br>1: Test Mode |
| 5:0 | R/W | **Test bit for PCI DLL function**<br>0: Normal Mode<br>1: Test Mode |

**Register 8D Timing Control for DRAM Controller**

Default Value: 62h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Reserved** |

| | | |
|---|---|---|
| 6:4 | R/W | **Clock Source Control for DRAM Output Signals**<br>These bits provide the clock source for DRAM output signals and this clock source is reference (lead/delay) to 600's CCLK input clock.<br>**Bits[6:4] Clock Source**<br>000 Lead 3.0ns<br>001 Lead 2.5ns<br>010 Lead 2.0ns<br>011 Lead 1.5ns<br>100 Lead 1.0ns<br>101 Lead 0.5ns<br>110 Lead 0.0ns<br>111 Delay 0.5ns |
| 3 | R/W | **Reserved** |
| 2:0 | R/W | **Clock Source Control for DRAM Input Signals**<br>These bits provide the clock source for DRAM input signals and this clock source is reference (lead/delay) to 600's CCLK input clock.<br>**Bits[6:4] Clock Source**<br>000 Lead 0.5ns<br>001 Lead 0.0ns<br>010 Delay 0.5ns<br>011 Delay 1.0ns<br>100 Delay 1.5ns<br>101 Delay 2.0ns<br>110 Delay 2.5ns<br>111 Delay 3.0ns |

**Register 8E/8F General Purpose Register II**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **General Purpose Register**<br>This register provides the storage information for BIOS programming. |

### 8.1.11. A.G.P. GART AND PAGE TABLE CONTROL REGISTERS

**Register 90h GART Base Address for Re-mapping**

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:12 | R/W | **A[31:12] for GART Base Address**<br>This register specifies the starting address of the Graphics Address Re-mapping Table (GART) base address located in main memory, which is always 4KB aligned. |
| 11:0 | R/W | **Reserved** |

**Register 94h Graphic Window Control**

Default Value: 00h

Access: Read/Write

This register specifies the size of the graphic window and indicates whether the Graphic Window Base Address Register and Re-mapping GART Base Address Register contain valid information or not.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Reserved** |
| 6:4 | R/W | **Graphic Window Size**<br>This field defines the size of the graphic window. The accessibility of GWBA register (Register 10h) is also controlled by this field.<br>**Bits[6:4] Size**<br>000 4M<br>001 8M<br>010 16M<br>011 32M<br>100 64M<br>101 128M<br>110 256M<br>111 Reserved |
| 3:2 | R/W | **Reserved** |

| | | |
|---|---|---|
| 1 | R/W | **Graphic Window Base Address (Register 10[31:22]) Validation**<br>The bit indicates the Graphic Window Base Address specified in GWBA Register (Register 10h) is valid or not.<br>0: Invalid<br>1: Valid |
| 0 | R/W | **GART Base Address for Re-mapping (Register 90[31:12]) Validation**<br>This bit indicates the Re-mapping GART Base Address specified in Register 90h is valid or not.<br>0: Invalid<br>1: Valid |

**Register 97h Page Table Cache Control**

Default Value: 00h

Access: Read/Write

Page Table Cache is used to speedup the address translation process from graphic address to system memory address. It stores recently used GART entries such that system memory access cycles can be reduced. This register controls the characteristic of the page table cache and the address translation mechanism.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | R/W | **Reserved** |
| 2 | R/W | **Page Table Cache Invalidation Control**<br>When set to 0, write cycles to GART entries will automatically invalidate the entire page table cache by hardware immediately. When set to 1, the mini-port driver will be responsible for writing the Page Table Cache Invalidation Control Register (Register 98h) to invalidate the page table cache when GART is being updated.<br>0: Detect writing GART entry<br>1: Write Configuration Register 98h |
| 1 | R/W | **Reserved** |
| 0 | R/W | **Page Table Cache Enable**<br>When enabled, page table cache will be used for accelerating the address translation process. When disabled, no GART entries are cached in the page table cache.<br>0: Disable<br>1: Enable |

**Register 98h Page Table Cache Invalidation Control**

Default Value: 00000000h

Access: Read/Write

This register controls the invalidation of page table cache. The invalidation can be applied to only one entry or to all entries.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:12 | R/W | **Graphic Window Memory Address**<br>Bits[31:12] define A[31:12] of a specific page of 4K A.G.P. graphic window memory address. There is a certain GART entry to handle address translation of the page. If the GART entry is resided in the page table cache and "1" is being written to bit 0 of this register, it is going to be invalidated. |
| 11:2 | R/W | **Reserved** |
| 1 | R/W | **Invalidate All**<br>Invalidate all page table cache entries when a "1" is being written to this bit. This bit is cleared after the invalidation is completed. |
| 0 | R/W | **Invalidate Entry**<br>Invalidate the corresponding page table cache entry specified in Bits[31:12] when a "1" is being written to this bit. This bit will be cleared after the invalidation is completed. |

**Register 9Ch ~ AFh Reserved**

### 8.1.12. A.G.P AND 66MHZ HOST BRIDGE CONTROL REGISTERS

**Register C0h AGP Capability Identify Register(ACAPID)**

Default Value: 00100002h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:24 | RO | **Reserved** |
| 23:20 | RO | **AGP revision Major**<br>Default value is "0001b" to indicate that SiS600 conforms to the major revision 1 of A.G.P. interface specification. |
| 19:16 | RO | **AGP revision Minor**<br>Default value is "0000b" to indicate that SiS600 conforms to the minor revision 0 of A.G.P. interface specification.. |

| | | |
|---|---|---|
| 15:8 | RO | **Next Capability**<br>Default value is “00h” to indicate the final item. |
| 7:0 | RO | **A.G.P. Capability ID**<br>Default value is “02h”. to indicate the list item as pertaining to A.G.P. registers. |

**Register C4h A.G.P. Status Register**

Default Value: 1F000203h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:24 | RO | **RQ Field**<br>The RQ field contains the maximum number of AGP command requests that SiS600 can manage. Default value is “1Fh” and it means 32. |
| 23:10 | RO | **Reserved** |
| 9 | RO | **SBA**<br>Default value is 1 to indicate that SiS600 supports side band addressing. |
| 8:2 | RO | **Reserved** |
| 1:0 | RO | **Data Rate**<br>The RATE field indicates the supported data transfer rates.<br>Default value is “11b” to indicate SiS600 support both 1X and 2X mode. |

**Register C8h A.G.P. Command Register**

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:10 | R/W | **Reserved** |
| 9 | R/W | **SBA_ENABLE.**<br>When set, the side band address mechanism is enabled. |
| 8 | R/W | **AGP_ENABLE.**<br>Setting the bit allows the target to accept A.G.P. bus operations. When cleared, the target ignores incoming A.G.P. bus operations. Please note that the target must be enabled before the master. |
| 7:2 | R/W | **Reserved** |

| | | |
|---|---|---|
| 2 | R/W | **Asynchronous/Synchronous Clock Support for CPU/AGP Clock**<br>0: Asynchronous Mode<br>1: Synchronous Mode |
| 1:0 | R/W | **Data Rate**<br>One(and only one) bit in the DATA_RATE field must be set to indicate the desired data transfer rate. <Bit 0: 1X, Bit 1: 2X>. The same bit must be set on both master and target. The DATA_RATE field applies to AD and SBA buses.<br>**Bits[1:0] Data Rate**<br>00 Reserved<br>01 1X mode<br>10 2X mode<br>11 Reserved |

**Register CC~D3h Reserved**

**Register D4h A.G.P. Interface Arbitration Timer (1)**

Default Value: 00h

Access: Read Write

This timer controls the maximum amount of time that is allocated to a master using PCI protocol to perform multiple back-to-back transactions on the A.G.P. The timer applies to both the host bridge and the A.G.P. masters.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for Timer (1)**<br>Unit: A.G.P. clock |

**Register D5h A.G.P. Interface Arbitration Timer (2)**

Default Value: 00h

Access: Read Write

This timer controls the minimum amount of time allocated to low priority data transaction.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for Timer (2)**<br>Unit: A.G.P. clock |

**Register D6h Data Transfer Counter (DTC)**

Default Value: 00h

Access: Read Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|

| | | |
|---|---|---|
| 7:4 | R/W | **WBFC (Write Buffer Flush Counter)**<br>WBFC counter controls the number of write transactions to be flushed into system memory after write buffer is full. The counter applies both to low and high priority transactions. |
| 3:0 | R/W | **RWDTC (R/W Data Transfer Counter Register)**<br>DTC controls the number of grants allocated to read and write data. DTC applies both to low and high priority transaction and is used to allow back-to-back transaction on A.G.P. bus. |

**Register D7h PCI 33/66 Idle Timer (PIT)**

Default Value: 00h

Access: Read Write

The timer is used to prevent PCI33 or PCI66 master from idling too long while outstanding PCI requests cannot be serviced.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for PCI Idle Timer**<br>Unit: CPU clock |

**Register D8h AD_STB Output Control**

Default Value: 00h

Access: Read Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:5 | R/W | **Extend delay time control for AD_STB Output Buffer**<br>**Bit[7:5] Delay**<br>000 0.5ns<br>001 1.5ns<br>010 2.5ns<br>011 3.5ns<br>100 4.5ns<br>101 5.5ns<br>110 6.5ns<br>111 7.0ns |

| | | |
|---|---|---|
| 4:3 | R/W | **AD_STB Adjusted delay time (stage1)**<br>**Bit[4:3] Delay**<br>00 0.3ns<br>01 1.5ns<br>10 2.5ns<br>11 3.5ns |
| 2:1 | R/W | **AD_STB Adjusted delay time(stage2)**<br>**Bit[2:1] Delay**<br>00 0.5ns<br>01 1.5ns<br>10 2.5ns<br>11 3.5ns<br>Using the inverse 66Mhz clock to generate AD_STB, we need some adjusting delay time on stage1 and stage2. |
| 0 | R/W | **Reserved** |

**Register D9h Reserved**

**Register DAh PBT (PCI 33/66 Balance Timer)**

Default Value: 00h

Access: Read/Write

This register controls a fair arbitration scheme between PCI33 and PCI66 masters. The timer is used to prevent PCI masters from seizing the PCI bus too long. When the timer expires, PCI arbiter forces the master that is currently occupying PCI bus to relinquish PCI bus by removing its grant.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for PCI balance timer**<br>Unit: CPU clock |

**Register DF~DBh Reserved**

**Register E0h** **A.G.P. Compliant Target/Arbiter Control Register (ACTACR)**

Default Value: 00h

Access: Read/Write

The 8-bit register contains various options for A.G.P. compliant target and arbiter.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:6 | R/W | **Read Buffer Threshold Point for Read GNT**<br>The read grant for low-priority AGP master will not be asserted on the A.G.P. bus until certain threshold amount of data is available in the internal read buffer. A.G.P. bus utilization will be improved if larger amount of data is transferred in a single burst. Please note that high-priority read grant is not affected by this setting.<br>**Bits[7:6] Threshold**<br>11 8 blocks<br>10 4 blocks<br>01 2 blocks<br>00 1 block |
| 5 | R/W | **Pipeline Option for Read Grant**<br>0: Fast<br>1: Slow |
| 4 | R/W | **A.G.P. Flush Option**<br>0: Flush only low priority write<br>1: Flush both high and low priority write |
| 3 | R/W | **Assert GNT# in response to PIPE# or FRAME# without Holding CPU**<br>When disabled, SiS600 can only asserts GNT# in response to A.G.P. master requests by PIPE# or FRAME# when CPU is being held. When enabled, SiS600 may assert GNT# when CPU is not being held. This bit is recommended to set to 0 when a pure PCI device or an A.G.P. device using SBA is plugged in A.G.P. This bit recommended to set to 1 when an A.G.P. device that will use PIPE# to its requests is plugged in the system.<br>0: Disable<br>1: Enable |
| 2 | R/W | **A.G.P. Counter Test Mode.**<br>For internal test mode use only.<br>0: Normal Mode<br>1: Test Mode |

| | | |
|---|---|---|
| 1 | R/W | **Synchronous Mode.**<br>This bit can only be enabled when A.G.P. clock and CPU clock are generated from the same source and the external skew between these two clocks is less than 2ns.<br>0: Disable<br>1: Enable |
| 0 | R/W | **TRDY# Delay for Data Transfer**<br>When set to 0, TRDY# is asserted in the fastest timing for read data transfer. When set to 1, the assertion of TRDY# for read data transfer is delayed 1 clock.<br>0: Minimum Delay<br>1: Maximum Delay |

**Register E1h Reserved**

**Register E2h PCI66 Target Bridge Characteristics**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Reserved** |
| 6 | R/W | **PCI 33/66 Memory Read Multiple Lines Control**<br>This bit is only valid when PCI master memory read prefetch function is enabled. (Controlled by Register 82h bits[5:4] and Register E2h bits[5:4] for PCI33 and PCI66, respectively). When this bit is 0, one more pending memory read prefetch cycle will be issued by SiS600 for PCI masters; when this bit is 1, two more pending memory read prefetch cycles can be generated.<br>0: 1 more pending cycle<br>1: 2 more pending cycles |
| 5 | R/W | **PCI66 Memory Read Line or Memory Read Multiple Command Prefetch Enable**<br>This bit enables data prefetching for Memory Read Line or Memory Read Multiple commands.<br>0: Disable<br>1: Enable |
| 4 | R/W | **PCI66 Memory Read Command Prefetch Enable**<br>This bit enables data prefetching for Memory Read command.<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 3:2 | R/W | **Initial Latency Control**<br>This field controls the target initial latency of the PCI66 target bridge. If SiS600 is unable to assert TRDY# for a transaction within the target initial latency defined by this field, SiS600 asserts STOP# to retry this cycle.<br>00: Disable<br>01: 16 PCI33 Clocks<br>10: 32 PCI33 Clocks<br>11: 48 PCI33 Clocks |
| 1 | R/W | **Subsequent Latency Control**<br>When this bit is enabled, SiS600 terminates a transaction with STOP# if it is unable to assert TRDY# for the subsequent data transfers within 8 clocks.<br>0: Disable<br>1: Enable |
| 0 | R/W | **Test Mode**<br>This bit controls the test mode for PCI66.<br>0: Test mode<br>1: Normal mode |

**Register E3h CPU to PCI66 Bridge characteristics**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | R/W | **Reserved** |
| 2 | R/W | **Non-Post Cycle Retry Behavior Control**<br>When enabled, CPU-to-PCI non-post cycle being retried by PCI target will be regenerated once, before SiS600 backs off CPU. When disabled, SiS600 will back off CPU immediately when the current CPU-to-PCI non-post cycle is being retried.<br>1: Enable<br>0: Disable |
| 1 | R/W | **Memory Burst Control**<br>This bit enables the host bridge to generate memory burst cycles.<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 0 | R/W | **Memory Post Write Control**<br>When enabled, all CPU to PCI memory-write cycles are posted.<br>0: Disable<br>1: Enable |

## 8.2. VIRTUAL PCI-TO-PCI BRIDGE REGISTERS (DEVICE 2)

**Register 00h Vendor ID**

Default Value: 1039h

Access: Read Only

The register identifies the manufacturer of the device. SiS is allocated as 1039h by PCI SIG.

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

**Register 02h Device ID**

Default Value: 0001h

Access: Read Only

The device identifier is allocated as 0001h by Silicon Integrated Systems Corp.

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

**Register 04h Command**

Default Value: 00h

Access: Read/Write, Read Only

The Command register provides coarse control over a device's ability to generate and respond to PCI cycles.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:10 | RO | **Reserved** |
| 9 | R/W | **Fast Back-to-Back Enable**<br>This bit is not used by the internal logic as a control signal, though it is implemented to be PCI compatible. The generation of fast back-to-back cycle is always controlled by the Fast back-to-back Control bit located in Register E3h of function 0, device0. |

| | | |
|---|---|---|
| 8 | R/W | **SERR# Enable**<br>This bit and bit 1 of the bridge control register (offset 3Fh) control the forwarding of ASERR# on A.G.P. bus to SERR# on PCI bus. When both of these two bits are enabled, SiS600 asserts SERR# when it detects the assertion of ASERR# on A.G.P. bus. When any one of these two bits is disabled, the assertion of ASERR# would not be forwarded to SERR#.<br>0: Disable<br>1: Enable |
| 7:6 | RO | **Reserved**<br>Default value is 00b |
| 5 | R/W | **VGA Palette Snoop Enable**<br>This bit controls the forwarding of CPU initiated access cycle toward VGA compatible addresses, which include 3C6h, 3C8h and 3C9h. When enabled, these I/O write cycles will be forwarded to both PCI bus and A.G.P bus. When disabled, these I/O write cycles will be forwarded only to PCI bus or A.G.P. bus that subject to VGA enable bit.<br>0: Disable<br>1: Enable |
| 4:3 | RO | **Reserved** |
| 2 | R/W | **Bus Master Enable**<br>This bit controls the SiS600's response to PCI cycles on A.G.P. bus. When disabled, the bridge does not respond to any transaction on A.G.P. bus.<br>0: Disable<br>1: Enable |
| 1 | R/W | **Memory Space Enable**<br>This bit controls the forwarding of memory accesses from CPU to A.G.P. bus. When disabled, the bridge would not forward any memory accesses to A.G.P. bus. When enabled, the bridge will forward CPU memory cycles toward A.G.P. bus according to standard PCI-to-PCI bridge forwarding rule.<br>0: Disable<br>1: Enable |
| 0 | R/W | **I/O Space Enable**<br>This bit controls the forwarding of I/O accesses from CPU to A.G.P. When disabled, the bridge would not forward any I/O accesses to A.G.P. When enabled, the bridge will forward CPU I/O cycles toward A.G.P. bus according to standard PCI-to-PCI bridge forwarding rule.<br>0: Disable<br>1: Enable |

**Register 06h Status**

Default Value: 00h

Access: Read Only

This register is reserved since the status information of the primary bus is stored in the status register of Device 0.

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

**Register 08h Revision ID**

Default Value: 00h

Access: Read Only

The Revision ID is 00h for our first Revision.

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

**Register 09h Programming Interface**

Default Value: 00h

Access: Read Only

The default value is 00h since no specific register-level programming interface is provided.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Programming Interface** |

**Register 0Ah Sub Class Code**

Default Value: 04h

Access: Read Only

The Sub Class Code is 04h for PCI-to-PCI bridge.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Sub Class Code** |

**Register 0Bh Base Class Code**

Default Value: 06h

Access: Read Only

The value of 06h in this field identifies a bridge device.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Base Class Code** |

**Register 0Ch Cache Line Size**

Default Value: 00h

Access: Read Only

The value of this register is always 00h and it indicates the host bridge would not generate the Memory Write and Invalidate command.

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

**Register 0Dh Master Latency Timer (MLT)**

Default Value: 00h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Initial Value for Master Latency Timer**<br>Unit: A.G.P. clock |

**Register 0Eh Header Type**

Default Value: 01h

Access: Read Only

The value of 01h identifies PCI-to-PCI bridge header is being used.

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

**Register 0Fh BIST**

Default Value: 00h

Access: Read Only

The value is 00h since SiS600 does not support Build-in Self Test function.

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

**Register 19h Secondary Bus Number (SBUSN)**

Default Value: 00h

Access: Read/Write

This register identifies the bus number assigned to the second bus side of the “virtual” PCI-to-PCI Bridge. This field is programmed by the PCI configuration software to allow mapping of configuration cycles to AGP.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Secondary Bus Number** |

**Register 1Ah Subordinate Bus Number (SUBUSN)**

Default Value: 00h

Access: Read/Write

This register is used to record the number of the highest numbered PCI bus that is behind A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Subordinate Bus Number**<br>Default value is 00h. |

**Register 1Bh Secondary Master Latency Timer (SMLT)**

Default Value: 00h

Access: Read/Write, Read Only

This register adheres to the definition of the Latency Timer in the PCI Local Bus Specification but applies only to A.G.P. interface.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | R/W | **Secondary Master Latency Timer**<br>Unit: 8 * A.G.P clock |
| 2:0 | RO | **Reserved** |

**Register 1Ch I/O Base**

Default Value: 0Fh

Access: Read/Write, Read Only

The I/O Base register defines the bottom address of an address range that is used by SiS600 to determine when to forward I/O transactions from CPU to A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **I/O Address Base A[15:12]**<br>Bits[7:4] controls the CPU to A.G.P. I/O access. 600 forward I/O cycle initiated by CPU to A.G.P. if the address of the cycle meets the following requirement.<br>IO_BASE ≤ address ≤ IO_LIMIT |
| 3:0 | RO | **Reserved** |

**Register 1Dh I/O Limit**

Default Value: 00h

Access: Read/Write, Read Only

The I/O Limit register defines the top address of an address range that is used SiS600 to determine when to forward I/O transactions from CPU to A.G.P.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | R/W | **I/O Address Limit A[15:12]**<br>Bits[7:4] controls the CPU to A.G.P. bus I/O access. SiS600 forward I/O cycle initiated by CPU to A.G.P. bus if the address of the cycle meets the following requirement.<br>IO_BASE ≤ address ≤ IO_LIMIT |
| 3:0 | RO | **Reserved** |

**Register 1Eh Secondary PCI-PCI Status (SSTS)**

Default Value: 0000h

Access: Read/Write, Read Only

The Secondary Status register is similar in function and bit definition to the Status register of device 0 function 0 of SiS600.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15 | RO | **Reserved** |
| 14 | WC | **Receiver System Error.**<br>This bit is set when ASERR# assertion is detected on A.G.P. bus. This bit can be cleared by writing a 1 to it. |
| 13 | WC | **Receiver Master Abort**<br>When 600 terminates a cycle on A.G.P. bus with master abort. This bit is set to 1. This bit can be cleared by writing a 1 to it. |
| 12 | WC | **Receiver Target Abort**<br>When a 600 initiated cycle on A.G.P. is terminated with a target abort. This bit is set to 1. This bit can be cleared by writing a 1 to it. |
| 11:0 | RO | **Reserved** |

**Register 20h** **Non-prefetchable Memory Base Address (MBASE)**

Default Value: FFF0h

Access: Read/Write, Read Only

The register defines the base address of a non-prefetchable memory address range that is used by SiS600 to determine when to forward memory transactions from CPU to A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Memory Address Base A[31:20]**<br>Bits[15:4] controls the CPU to A.G.P. memory access. SiS600 forward I/O cycle initiated by CPU to A.G.P. bus if the address of the cycle meets the following requirement.<br>MBASE ≤ address ≤ MLIMIT |
| 3:0 | RO | **Reserved** |

**Register 22h** **Non-prefetchable Memory Limit Address (MLIMIT)**

Default Value: 0000h

Access: Read/Write, Read Only

The register defines the top address of a non-prefetchable memory address range that is used by SiS600 to determine when to forward memory transactions from CPU to A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Memory Address Limit A[31:20].**<br>Bits[15:4] controls the CPU to A.G.P. memory access. SiS600 forward I/O cycle initiated by CPU to A.G.P. bus if the address of the cycle meets the following requirement.<br>MBASE ≤ address ≤ MLIMIT |
| 3:0 | RO | **Reserved** |

**Register 24h** **Prefetchable Memory Base Address (PMBASE)**

Default Value: FFF0h

Access: Read/Write, Read Only

The register defines the base address of a prefetchable memory address range that is used by SiS600 to determine when to forward memory transactions from CPU to A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Memory Address Base A[31:20].**<br>Bits[15:4] controls the CPU to A.G.P. memory access. SiS600 forward I/O cycle initiated by CPU to A.G.P. bus if the address of the cycle meets the following requirement.<br>MBASE ≤ address ≤ MLIMIT |
| 3:0 | RO | **Reserved** |

**Register 26h** **Prefetchable Memory Limit Address (PMLIMIT)**

Default Value: 0000h

Access: Read/Write, Read Only

The register defines the top address of a prefetchable memory address range that is used by SiS600 to determine when to forward memory transactions from CPU to A.G.P. bus.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | R/W | **Memory Address Limit A[31:20].**<br>Bits[15:4] controls the CPU to A.G.P. memory access. SiS600 forward I/O cycle initiated by CPU to A.G.P. bus if the address of the cycle meets the following requirement.<br>MBASE ≤ address ≤ MLIMIT |
| 3:0 | RO | **Reserved** |

**Register 3Eh** **PCI to PCI Bridge Control (BCTRL)**

Default Value: 0000h

Access: Read/Write, Read Only

The Bridge Control register provides control extensions to the Command register.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:4 | RO | **Reserved.** |

| | | |
|---|---|---|
| 3 | R/W | **VGA Enable**<br>The bit controls the forwarding of transactions initiated by CPU. When the bit is enabled, 600 forwards CPU-initiated cycles with the following address to A.G.P. bus.<br>Memory Address: 0A0000h ~ 0BFFFFh<br>I/O Address: 3B0h ~ 3BBh, 3C0 ~ 3DFh<br>0 : Disable<br>1 : Enable |
| 2 | R/W | **ISA Enable**<br>When enable, I/O cycles with addresses fall within the upper 768 bytes in each 1KB block (i.e., A9 or A8 = 1) will be forwarded to Primary PCI even if the address is within the range defined by the IOBASE and IOLIMIT.<br>0 : Disable<br>1 : Enable |
| 1 | R/W | **System Error Enable**<br>This bit and bit 1 of the bridge control register (offset 3Fh) control the forwarding of ASERR# on A.G.P. to SERR# on PCI bus. When both of these two bits are enabled, SiS600 asserts SERR# when it detects the assertion of ASERR# on A.G.P. When any one of these two bits is disabled, the assertion of ASERR# won't affect SERR#.<br>0: Disable<br>1: Enable |
| 0 | RO | **Reserved** |

## 8.3. PCI IDE CONFIGURATION SPACE REGISTER

| DEVICE | IDSEL | FUNCTION NUMBER |
|---|---|---|
| IDE | AD11 | 0001b |

**Register 00h Vendor ID**

Default Value: 1039h

Access: Read Only

The register identifies the manufacturer of the device. SiS is allocated as 1039h by PCI SIG.

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

**Register 02h Device ID**

Default Value: 5513h

Access: Read Only

The device identifier is allocated as 5513h by Silicon Integrated Systems Corp.

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

**Register 04h Command**

Default Value: 0000h

Access: Read/Write, Read Only

The Command register provides coarse control over a device ability to generate and respond to PCI cycles.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:3 | RO | **Reserved** |
| 2 | RO | **Bus Master**<br>When set, the Bus master function is enabled. It is disabled by default. |
| 1 | R/W | **Memory Space**<br>The bit controls the response to memory space accesses. This bit should be programmed as "0". |
| 0 | RO | **IO Space**<br>When enabled, the built-in IDE will respond to any access of the IDE legacy ports in the compatibility mode, or to any access of the IDE relocatable ports in the native mode. Also, any access to the PCI bus master IDE registers are allowed. This bit is zero (disabled) on reset. |

**Register 06h Status**

Default Value: 0000h

Access: Read/Write, Read Only, Write Clear

The Status register is used to record status information for PCI bus related events. Reads to this register behave normally. Writes are slightly different in that bits can be reset, but not set. A bit is reset whenever the register is written, and the data in the corresponding bit location is a 1. For instance, to clear bit 14 and not affect any other bits, write the value 0100_0000_0000_0000b to the register.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:14 | RO | **Reserved**<br>These bits are hardware to zero. |
| 13 | WC | **Master Abort Asserted**<br>This bit is set when a PCI bus master IDE transaction is terminated by master abort. While this bit is set, IDE will issue an interrupt request. This bit can be cleared by writing a 1 to it. |
| 12 | WC | **Received Target Abort**<br>The bit is set whenever PCI bus master IDE transaction is terminated with target abort. |
| 11 | RO | **Signaled Target Abort**<br>The bit will be asserted when IDE terminates a transaction with target abort. |
| 11:9 | RO | **DEVSEL# Timing DEVT.**<br>These two bits define the timing of asserting DEVSEL#. The built-in IDE always asserts DEVSEL# in fast timing, and thus the two bits are hardwired to 0 per PCI Spec. |
| 8 | R/W | **Reserved, Read as "0".** |
| 7:0 | RO | **Reserved**<br>Default value is 00h |

**Register 08h Revision ID**

Default Value: D0h

Access: Read Only

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

**Register 09h Programming Interface**

Default Value: 00h

Access: Read Only

The default value is 00h since no specific register-level programming interface is provided.

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **Master IDE Device**<br>This bit is hardwired to one to indicate that the built-in IDE is capable of supporting bus master function. |
| 6:4 | RO | **Reserved** |
| 3 | RO | **Secondary IDE Programmable Indicator**<br>When the bit is programmed as "1", it means that the primary channel can be programmed to operate in compatible or native mode. When the bit is programmed as "0"the mode is fixed and is determined by the value of bit 2. This bit should be programmed as ¡ 1¢ during the BIOS boot up procedures. |
| 2 | RO | **Secondary IDE Operating Mode**<br>This bit defines the mode that the secondary IDE channel is operating in. Zero corresponds to 'compatibility' while one means native mode. By default, this bit is 0 and is programmable. |
| 1 | RO | **Primary IDE Programmable Indicator**<br>When the bit is programmed as '1', it means that the primary channel can be programmed to operate in compatible or native mode. When the bit is programmed as ¡ 0¢ , the mode is fixed and is determined by the value of bit 0. This bit should be programmed as ¡ 1¢ during the BIOS boot up procedures. |
| 0 | RO | **Primary IDE Operating Mode**<br>This bit defines the mode that the primary IDE channel is operating in. Zero corresponds to 'compatibility' while one means native mode. By default, this bit is 0 and is programmable. |

**Register 0Ah Sub Class Code**

Default Value: 01h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Sub Class Code** |

**Register 0Bh Base Class Code**

Default Value: 01h

Access: Read Only

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | RO | **Base Class Code** |

**Register 0Ch Cache Line Size**

Default Value: 00h

Access: Read Only

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

**Register 0Dh Latency Timer**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:0 | R/W | **Initial Value for Latency Timer**<br>The default value is 0.<br>Unit: PCI clock |

**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~13h Primary Channel Command Block Base Address Register**

**Register 14h~17h Primary Channel Control Block Base Address Register**

**Register 18h~1Bh Secondary Channel Command Block Base Address Register**

**Register 1Ch~1Fh Secondary Channel Control Block Base Address Register**

In the native mode, above four registers define the IDE base address for each of the two IDE devices in both the primary and secondary channels respectively. In the compatible mode, the four registers can still be programmed and read out, but it does not affect the IDE address decoding.

**Register 20h~23h Bus Master IDE Control Register Base Address**

| OFFSET REGISTER | REGISTER ACCESS |
|---|---|
| 00H | **Bus Master IDE Command Register (Primary)** |
| 01H | **Reserved** |
| 02H | **Bus Master IDE Status Register(Primary)** |
| 03H | **Reserved** |
| 04-07H | **Bus Master IDE PRD (*) Table Pointer (Primary)** |
| 08H | **Bus Master IDE Command Register (Secondary)** |
| 09H | **Reserved** |
| 0AH | **Bus Master IDE Status Register (Secondary)** |
| 0BH | **Reserved** |
| 0C-0FH | **Bus Master IDE PRD (*) Table Pointer (Secondary)** |

*PRD: Physical Region Descriptor

**Register 24h~2Bh Reserved**

Default Value: 00h

Access: RO

**Register 2C~2Dh Subsystem Vendor ID**

Default Value: 0000h

Access: Read/Write

This register can be written once and is used to identify vendor of the subsystem.

**Register 2Eh~2Fh Subsystem Vendor ID**

Default Value: 0000h

Access: Read/Write

This register can be written once and is used to identify subsystem ID.

**Register 30h~33h Expansion ROM Base Address**

Default Value: 00000000h

Access: Read/Write

**Register 34h~3Fh Reserved**

**Register 40h IDE Primary Channel/Master Drive Data Recovery Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Test mode for internal use only**<br>0: Normal mode<br>1: Test mode<br>The test mode recovery and active timer counter. |
| 6 | R/W | **Test mode for internal use only**<br>0: Normal mode<br>1: Test mode<br>The test mode of prefetch byte counter. |
| 5:4 | R/W | **Reserved** |
| 3:0 | R/W | **Recovery Time**<br>0000: 12 PCICLK 0001: 1 PCICLK<br>0010: 2 PCICLK 0011: 3 PCICLK<br>0100: 4 PCICLK 0101: 5 PCICLK<br>0110: 6 PCICLK 0111: 7 PCICLK<br>1000: 8 PCICLK 1001: 9 PCICLK<br>1010: 10 PCICLK 1011: 11 PCICLK<br>1100: 13 PCICLK 1101: 14 PCICLK<br>1110: 15 PCICLK 1111: 15 PCICLK |

**Register 41h IDE Primary Channel/Master Drive Data Active Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Ultra DMA Mode Control**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 6:5 | R/W | **Ultra DMA/33 cycle time Select**<br>00: Reserved<br>01: Cycle time of 2 PCI clocks for data out<br>10: Cycle time of 3 PCI clocks for data out<br>11: Cycle time of 4 PCI clocks for data out |
| 4:3 | RO | **Reserved** |
| 2:0 | R/W | **Data Active Time Control**<br>000: 8 PCICLK 001: 1 PCICLK<br>010: 2 PCICLK 011: 3 PCICLK<br>100: 4 PCICLK 101: 5 PCICLK<br>110: 6 PCICLK 111: 12 PCICLK |

**Register 42h IDE Primary Channel/Slave Drive Data Recovery Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved** |
| 3:0 | R/W | **Recovery Time**<br>0000: 12 PCICLK 0001: 1 PCICLK<br>0010: 2 PCICLK 0011: 3 PCICLK<br>0100: 4 PCICLK 0101: 5 PCICLK<br>0110: 6 PCICLK 0111: 7 PCICLK<br>1000: 8 PCICLK 1001: 9 PCICLK<br>1010: 10 PCICLK 1011: 11 PCICLK<br>1100: 13 PCICLK 1101: 14 PCICLK<br>1110: 15 PCICLK 1111: 15 PCICLK |

**Register 43h IDE Primary Channel/Slave Drive Data Active Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Ultra DMA Mode Control**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 6:5 | R/W | **Ultra DMA/33 cycle time Select**<br>00: Reserved<br>01: Cycle time of 2 PCI clocks for data out<br>10: Cycle time of 3 PCI clocks for data out<br>11: Cycle time of 4 PCI clocks for data out |
| 4:3 | RO | **Reserved** |
| 2:0 | R/W | **Data Active Time Control**<br>000: 8 PCICLK 001: 1 PCICLK<br>010: 2 PCICLK 011: 3 PCICLK<br>100: 4 PCICLK 101: 5 PCICLK<br>110: 6 PCICLK 111: 12 PCICLK |

**Register 44h IDE Secondary Channel/Master Drive Data Recovery Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved** |
| 3:0 | R/W | **Recovery Time**<br>0000: 12 PCICLK 0001: 1 PCICLK<br>0010: 2 PCICLK 0011: 3 PCICLK<br>0100: 4 PCICLK 0101: 5 PCICLK<br>0110: 6 PCICLK 0111: 7 PCICLK<br>1000: 8 PCICLK 1001: 9 PCICLK<br>1010: 10 PCICLK 1011: 11 PCICLK<br>1100: 13 PCICLK 1101: 14 PCICLK<br>1110: 15 PCICLK 1111: 15 PCICLK |

**Register 45h IDE Secondary Channel/Master Drive Data Active Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Ultra DMA Mode Control**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 6:5 | R/W | **Ultra DMA/33 cycle time Select**<br>00: Reserved<br>01: Cycle time of 2 PCI clocks for data out<br>10: Cycle time of 3 PCI clocks for data out<br>11: Cycle time of 4 PCI clocks for data out |
| 4:3 | RO | **Reserved** |
| 2:0 | R/W | **Data Active Time Control**<br>000: 8 PCICLK 001: 1 PCICLK<br>010: 2 PCICLK 011: 3 PCICLK<br>100: 4 PCICLK 101: 5 PCICLK<br>110: 6 PCICLK 111: 12 PCICLK |

**Register 46h IDE Secondary Channel/Slave Drive Data Recovery Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved** |
| 3:0 | R/W | **Recovery Time**<br>0000: 12 PCICLK 0001: 1 PCICLK<br>0010: 2 PCICLK 0011: 3 PCICLK<br>0100: 4 PCICLK 0101: 5 PCICLK<br>0110: 6 PCICLK 0111: 7 PCICLK<br>1000: 8 PCICLK 1001: 9 PCICLK<br>1010: 10 PCICLK 1011: 11 PCICLK<br>1100: 13 PCICLK 1101: 14 PCICLK<br>1110: 15 PCICLK 1111: 15 PCICLK |

**Register 47h IDE Secondary Channel/Slave Drive Data Active Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Ultra DMA Mode Control**<br>0: Disable<br>1: Enable |

| | | |
|---|---|---|
| 6:5 | R/W | **Ultra DMA/33 cycle time Select**<br>00: Reserved<br>01: Cycle time of 2 PCI clocks for data out<br>10: Cycle time of 3 PCI clocks for data out<br>11: Cycle time of 4 PCI clocks for data out |
| 4:3 | RO | **Reserved** |
| 2:0 | R/W | **Data Active Time Control**<br>000: 8 PCICLK 001: 1 PCICLK<br>010: 2 PCICLK 011: 3 PCICLK<br>100: 4 PCICLK 101: 5 PCICLK<br>110: 6 PCICLK 111: 12 PCICLK |

**Register 48h IDE Command Recovery Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved** |
| 3:0 | R/W | **Recovery Time**<br>0000: 12 PCICLK 0001: 1 PCICLK<br>0010: 2 PCICLK 0011: 3 PCICLK<br>0100: 4 PCICLK 0101: 5 PCICLK<br>0110: 6 PCICLK 0111: 7 PCICLK<br>1000: 8 PCICLK 1001: 9 PCICLK<br>1010: 10 PCICLK 1011: 11 PCICLK<br>1100: 13 PCICLK 1101: 14 PCICLK<br>1110: 15 PCICLK 1111: 15 PCICLK |

**Register 49h IDE Command Active Time Control**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:3 | RO | **Reserved** |

| | | |
|---|---|---|
| 2:0 | R/W | **Data Active Time Control**<br>000: 8 PCICLK 001: 1 PCICLK<br>010: 2 PCICLK 011: 3 PCICLK<br>100: 4 PCICLK 101: 5 PCICLK<br>110: 6 PCICLK 111: 12 PCICLK |

**Register 4Ah IDE General Control Register 0**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Bus Master generates PCI burst cycles Control**<br>0: Disable<br>1: Enable |
| 6 | R/W | **Test Mode for internal use only**<br>0: Test Mode<br>1: Normal Mode<br>The test mode of PCI FRAME signal: 1 flip-flop output 0 combination output. |
| 5 | R/W | **Fast post-write control**<br>0: Disabled<br>1: Enabled (Recommended) |
| 4 | R/W | **Test Mode for internal use only**<br>0: Normal Mode<br>1: Test Mode<br>When this bit is set 1, the IRQ of HD drive would pass direct to 8259. On the others hand, IDE would *gate IRQ until IDE FIFO is empty under ab-normal operation. |
| 3 | R/W | **Bus Master requests PCI bus ownership timing control**<br>0: PCI Request asserted when FIFO is 75% full during prefetch cycles.<br>1: PCI Request asserted when FIFO is 50% full during prefetch cycles.<br>The default value is '0'. |
| 2 | R/W | **IDE Channel 1 Enable Bit**<br>0: Disabled<br>1: Enabled |

| | | |
|---|---|---|
| 1 | R/W | **IDE Channel 0 Enable Bit**<br>0: Disabled<br>1: Enabled |
| 0 | R/W | **Reserved** |

**Register 4Bh IDE General Control register 1**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | R/W | **Enable Post-write of the Slave Drive in Channel 1**<br>0: Disabled<br>1: Enabled |
| 6 | R/W | **Enable Post-write of the Master Drive in Channel 1**<br>0: Disabled<br>1: Enabled |
| 5 | R/W | **Enable Post-write of the Slave Drive in Channel 0**<br>0: Disabled<br>1: Enabled |
| 4 | R/W | **Enable Post-write of the Master Drive in Channel 0**<br>0: Disabled<br>1: Enabled |
| 3 | R/W | **Enable Prefetch of the Slave Drive in Channel 1**<br>0: Disabled<br>1: Enabled |
| 2 | R/W | **Enable Prefetch of the Master Drive in Channel 1**<br>0: Disabled<br>1: Enabled |
| 1 | R/W | **Enable Prefetch of the Slave Drive in Channel 0**<br>0: Disabled<br>1: Enabled |

| | | |
|---|---|---|
| 0 | R/W | **Enable Prefetch of the Master Drive in Channel 0**<br>0: Disabled<br>1: Enabled |

(Following two 16-bit wide registers define the prefetching length of each IDE channel respectively.)

**Register 4Ch~4Dh Prefetch Count of Primary Channel**

Default Value: FFFFh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Prefetch Count of Primary Channel**<br>The Count (in bytes) of IDE prefetch. The maximum value can be programmed is 512. (Default value is 512) |

**Register 4Eh~4Fh Prefetch Count of Secondary Channel**

Default Value: FFFFh

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Prefetch Count of Secondary Channel**<br>The Count (in bytes) of IDE prefetch. The maximum value can be programmed is 512. (Default value is 512) |

**Register 50h~51h Reserved**

Default Value: 0000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 15:0 | R/W | **Reserved** |

**Register 52h IDE Miscellaneous Control Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved** |
| 3 | R/W | **IDE Command Timing Select**<br>0: The recovery and active time programmed in register 48h-49h will be applied to command cycles for all IDE devices<br>1: The recovery and active time programmed in register 40h-47h will be applied to command cycles for their associated IDE devices. |

| | | |
|---|---|---|
| 2 | R/W | **Control of IDE Programmable Indicator register(09 bit 1 and 3)**<br>0: IDE register 09 bit 1 and 3 would be read as ¡ 0§ ¨<br>1: IDE register 09 bit 1 and 3 would be programmable |
| 1 | R/W | **Test Mode for internal use only**<br>0 : Normal Mode<br>1 : Test Mode<br>If this bit is set 1, IDE would reset IDE FIFO pointer when 8 bit command is forward to HDDs' driver. This bit would work on the condition that the transferring byte count of OS is not equal to the byte count received by HDDs' driver. |
| 0 | R/W | **IDE FIFO Size Select**<br>0: 32 Bytes FIFO<br>1: 64 Bytes FIFO(Recommended) |

### 8.3.1. OFFSET REGISTERS FOR PCI BUS MASTER IDE CONTROL REGISTERS

The PCI Bus master IDE Registers use 16 bytes of I/O Space. These registers can be accessed through I/O R/W to the address defined in the Bus Master IDE control register Base Address in the PCI IDE Configuration space. The base address is also defined in Register 20h~23h of PCI IDE configuration space.

**Register 00h Bus Master Primary IDE Command Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved. Return 0 on reads.** |
| 6:5 | R/W | **Read or Write Control.**<br>This bit defines the R/W control of the bus master transfer. When set to zero, PCI bus master reads are conducted. When set to one, PCI bus master writes are conducted. |
| 2:1 | RO | **Reserved** |
| 0 | R/W | **Start/Stop Bus Master**<br>The SiS Chip built-in IDE Controller enables its bus master operation whenever it detects this bit changing from a zero to a one. The operation can be halted by writing a zero to this bit. |

**Register 01h Reserved**

Default Value: 00h

Access: RO

**Register 02h Bus Master Primary IDE Status Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **Simplex Only**<br>This bit is hardwired to zero to indicate that both bus master channels can be operated at a time. |
| 6 | R/W | **Drive 1 DMA Capable**<br>This R/W bit can be set by BIOS or driver to indicate that drive 1 for this channel is capable of DMA transfers. |
| 5 | R/W | **Drive 0 DMA Capable**<br>This R/W bit can be set by BIOS or driver to indicate that drive 0 for this channel is capable of DMA transfers. |
| 4:3 | RO | **Reserved. Return 0 on reads** |
| 2 | R/W | **Interrupt**<br>The bit is set by the rising edge of the IDE interrupt line to indicate that all data transferred from the drive is visible in the system memory. Writing a '1' to this bit can reset it. |
| 1 | RO | **Error**<br>This bit is set when the IDE controller encounters an error during data transferring to/from memory. |
| 0 | R/W | **Bus Master IDE Device Active**<br>This bit is set when the start bit in the command register is set. It can be cleared when the last transfer of a region is performed, or the start bit is reset. |

**Register 03h Reserved**

Default Value: 00h

**Register 04h~07h Bus Master Primary IDE PRD Table Pointer Register**

This 32-bit register contains address pointing to the starting address of the PRD table.

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:2 | R/W | **Base Address of the PRD Table** |
| 1:0 | R/W | **Reserved** |

***PRD: Physical Region Descriptor**

**Register 08h Bus Master Secondary IDE Command Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7:4 | RO | **Reserved. Return 0 on reads.** |
| 3 | R/W | **Read or Write Control.**<br>This bit defines the R/W control of the bus master transfer. When set to zero, PCI bus master reads are conducted. When set to one, PCI bus master writes are conducted. |
| 2:1 | RO | **Reserved** |
| 0 | R/W | **Start/Stop Bus Master**<br>The SiS chip built-in IDE Controller enables its bus master operation whenever it detects this bit changing from a zero to a one. The operation can be halted by writing a zero to this bit. |

**Register 09h Reserved**

Default Value: 00h

Access: RO

**Register 0Ah Bus Master Secondary IDE Status Register**

Default Value: 00h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 7 | RO | **Simplex Only**<br>This bit is hardwired to zero to indicate that both bus master channels can be operated at a time. |
| 6 | R/W | **Drive 1 DMA Capable**<br>This R/W bit can be set by BIOS or driver to indicate that drive 1 for this channel is capable of DMA transfers. |
| 5 | R/W | **Drive 0 DMA Capable**<br>This R/W bit can be set by BIOS or driver to indicate that drive 0 for this channel is capable of DMA transfers**.** |
| 4:3 | RO | **Reserved. Return 0 on reads** |
| 2 | R/W | **Interrupt**<br>The bit is set by the rising edge of the IDE interrupt line to indicate that all data transferred from the drive is visible in the system memory. Writing a '1' to this bit can reset it. |

| | | |
|---|---|---|
| 1 | RO | **Error**<br>This bit is set when the IDE controller encounters an error during data transferring to/from memory. |
| 0 | R/W | **Bus Master IDE Device Active**<br>This bit is set when the start bit in the command register is set. It can be cleared when the last transfer of a region is performed, or the start bit is reset. |

**Register 0Bh Reserved**

Default Value: 00h

Access: RO

**Register 0Ch~0Fh Bus Master Secondary IDE PRD Table Pointer Register**

This 32-bit register contains address pointing to the starting address of the PRD table.

Default Value: 00000000h

Access: Read/Write

| BIT | ACCESS | DESCRIPTION |
|---|---|---|
| 31:2 | R/W | **Base Address of the PRD Table** |
| 1:0 | R/W | **Reserved** |

*PRD: Physical Region Descriptor

# 9. ELECTRICAL CHARACTERISTICS

## 9.1. ABSOLUTE MAXIMUM RATINGS

**Table 9.1-1 Absolute Maximum Ratings**

| PARAMETER | MIN. | MAX. | UNIT |
|---|---|---|---|
| Ambient operation temperature | 0 | 70 | $^0$C |
| Storage temperature | -40 | 125 | $^0$C |
| Input voltage | -0.3 | Vcc+0.3 | V |
| Output voltage | -0.5 | 3.3 | V |

NOTE:

Stress above these listed may cause permanent damage to device. Functional operation of this device should be restricted to the conditions described under operating conditions.

## 9.2. DC CHARACTERISTICS

### 9.2.1. DC CHARACTERISTICS OF HOST, DRAM, PCI AND IDE INTERFACE

**TA=0-70$^0$C, GND=0V, VCC=3.3V± 5%, $V_{TT}$ = 1.5V± 10%,**

**Table 9.2-1 DC Characteristics of Host, DRAM, PCI and IDE Interface**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | NOTES |
|---|---|---|---|---|---|
| $V_{IH\_GTL}$ | GTL+ Input High Voltage | 2/3$V_{TT}$ +0.2 | | V | |
| $V_{IL\_GTL}$ | GTL+ Input High Voltage | | 2/3$V_{TT}$ -0.2 | V | |
| $V_{IH\_TTL}$ | TTL Input High Voltage | 2 | VCC+0.3 | V | |
| $V_{IL\_TTL}$ | TTL Input Low Voltage | -0.3 | 0.8 | V | |
| $V_{OL\_GTL}$ | GTL+ Output Low Voltage | | 0.6 | V | |
| $V_{OL\_TTL}$ | TTL Output Low Voltage | | 0.45 | V | |
| $V_{REF}$ | GTL+ Reference Voltage | 2/3$V_{TT}$ -2% | 2/3$V_{TT}$+2% | V | |
| $I_{OL\_GTL}$ | GTL+ Output Low Current | | 36 | mA | |
| $I_{OH\_TTL}$ | TTL Output High Current | -2 | | mA | |
| $I_{OL\_TTL}$ | TTL Output Low Current | | 3 | mA | |
| $I_{IL}$ | Input Leakage Current | | ±10 | mA | |
| $C_{IN}$, $C_{OUT}$ | PCI Input Capacitance | | 5~8 | pF | Fc=1 MHz |
| | DRAM Input Capacitance | | 5~8 | pF | Fc=1 MHz |
| | GTL+ Input Capacitance | | 3~6 | pF | Fc=1 MHz |

### 9.2.2. DC CHARACTERISTICS OF A.G.P. INTERFACE

**TA=0-70$^0$C, GND=0V, VCC5=5V± 5%, VCC=3.3V± 5 %**

**Table 9.2-2 DC Characteristics of A.G.P. Interface**

| SYMBOL | PARAMETER | MIN | MAX | UNIT | NOTES |
|---|---|---|---|---|---|
| $V_{DD}$ | I/O Supply Voltage | 3.0 | 3.6 | V | |
| $V_{REF}$ | Input Reference Voltage | $0.39V_{DD}$ | $0.41V_{DD}$ | V | |
| $V_{IL1}$ | Input Low Voltage for 1X Signaling | -0.5 | $0.3\ V_{DD}$ | V | |
| $V_{IH1}$ | Input High Voltage for 1X Signaling | $0.5V_{DD}$ | $V_{DD}$ +0.5 | V | |
| $V_{IL2}$ | Input Low Voltage for 2X Sinaling | | $V_{REF}$ -0.2 | V | |
| $V_{IH2}$ | Input High Voltage for 2X Signaling | $V_{REF}$+0.2 | | V | |
| $V_{OL}$ | Output Low Voltage | | $0.1\ V_{DD}$ | V | |
| $V_{OH}$ | Output High Voltage | $0.9V_{DD}$ | | V | |
| $I_{REF}$ | $V_{REF}$ Pin Input Current | | +10 | mA | |
| $I_{OL}$ | Output Low Current | 2,3,4 | | mA | Note1 |
| $I_{OH}$ | Output High Current | -2,-3,-4 | | mA | Note1 |
| $I_{IH}$ | Input High Leakage Current | | 70 | µA | |
| $I_{IL}$ | Input Low Leakage Current | | ±10 | µA | |
| $C_{IN}$ | Input Capacitance | | 5~8 | pF | |
| $C_{OUT}$ | Output Capacitance | | 12 | pF | |
| $\Delta_{CIN}$ | Strobe to Data Pin Capacitance Delta | -1 | 2 | pF | |

Note:

1. The driving current is programmed. Please refer to register description.

## 10. THERMAL ANALYSIS

### 10.1 CHIP THERMAL ANALYSIS WITHOUT HEAT SINK

Room Temp. =
25C (No flow)

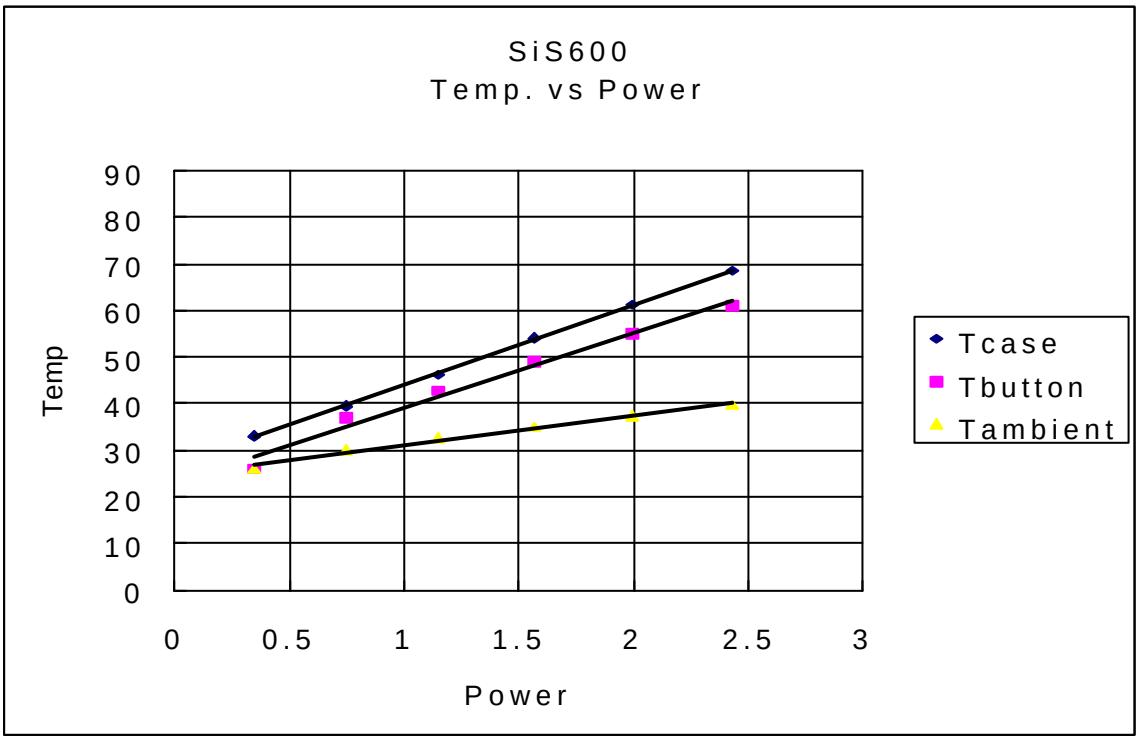
| Power | 0.352 | 0.744 | 1.152 | 1.57 | 1.995 | 2.43 |
|---|---|---|---|---|---|---|
| Tcase | 33.1 | 39.5 | 46.2 | 54.1 | 61.3 | 68.4 |
| Tbutton | 26.1 | 36.9 | 42.8 | 48.9 | 54.9 | 60.9 |
| Tambient | 26.0 | 30.2 | 32.6 | 35 | 37.3 | 39.8 |

**NOTE :**

Tcase : Temperature at the of molding compound Surface

Tbutton : Temperature at the back side of PCB where thermal balls are directly attached

Tambient : Temperature at PCB near the side of the BGA package



### 10.2 CHIP THERMAL ANALYSIS WITH HEAT SINK

Room Temp. = 25 °C

(No flow)

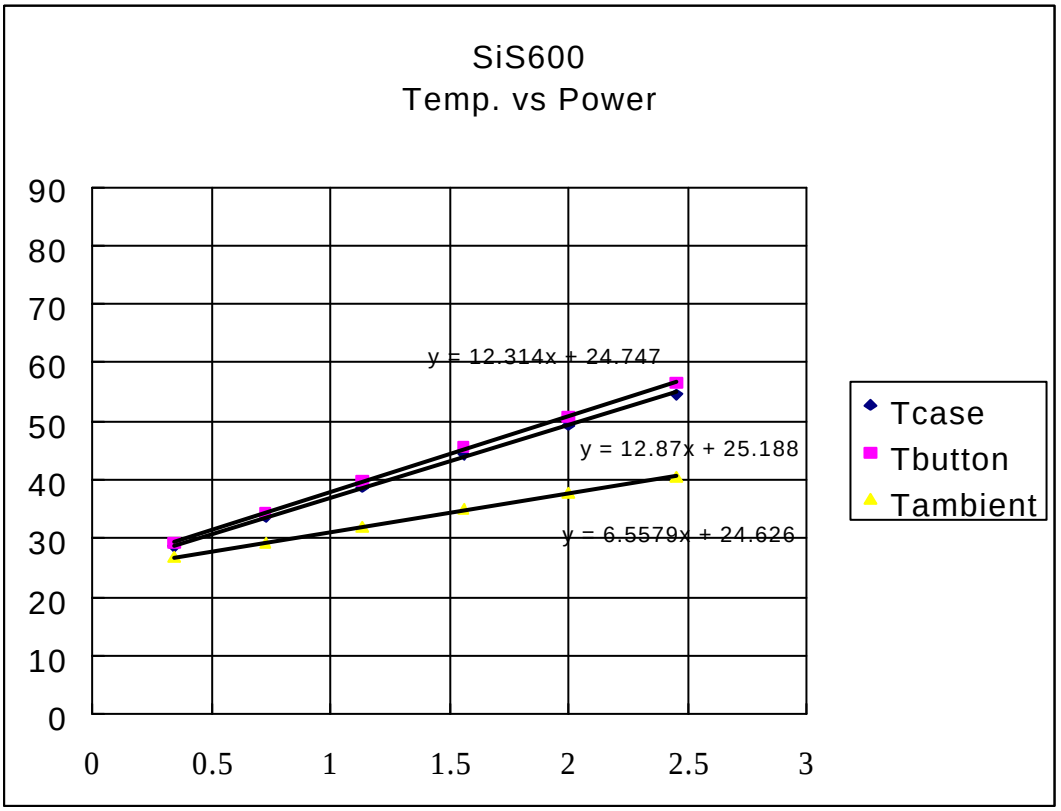
| Power (W) | 0.3385 | 0.724 | 1.131 | 1.556 | 1.995 | 2.451 |
|---|---|---|---|---|---|---|
| Tcase (°C) | 28.6 | 33.7 | 38.8 | 44.4 | 49.3 | 54.6 |
| Tbutton (°C) | 29.3 | 34.5 | 39.9 | 45.6 | 50.8 | 56.5 |
| Tambient (°C) | 26.8 | 29.4 | 31.9 | 35 | 37.9 | 40.5 |

**NOTE :**

Tcase : Temperature at the of molding compound Surface

Tbutton : Temperature at the back side of PCB where thermal balls are directly attached

Tambient : Temperature at PCB near the side of the BGA package



## 11. MECHANICAL DIMENSION

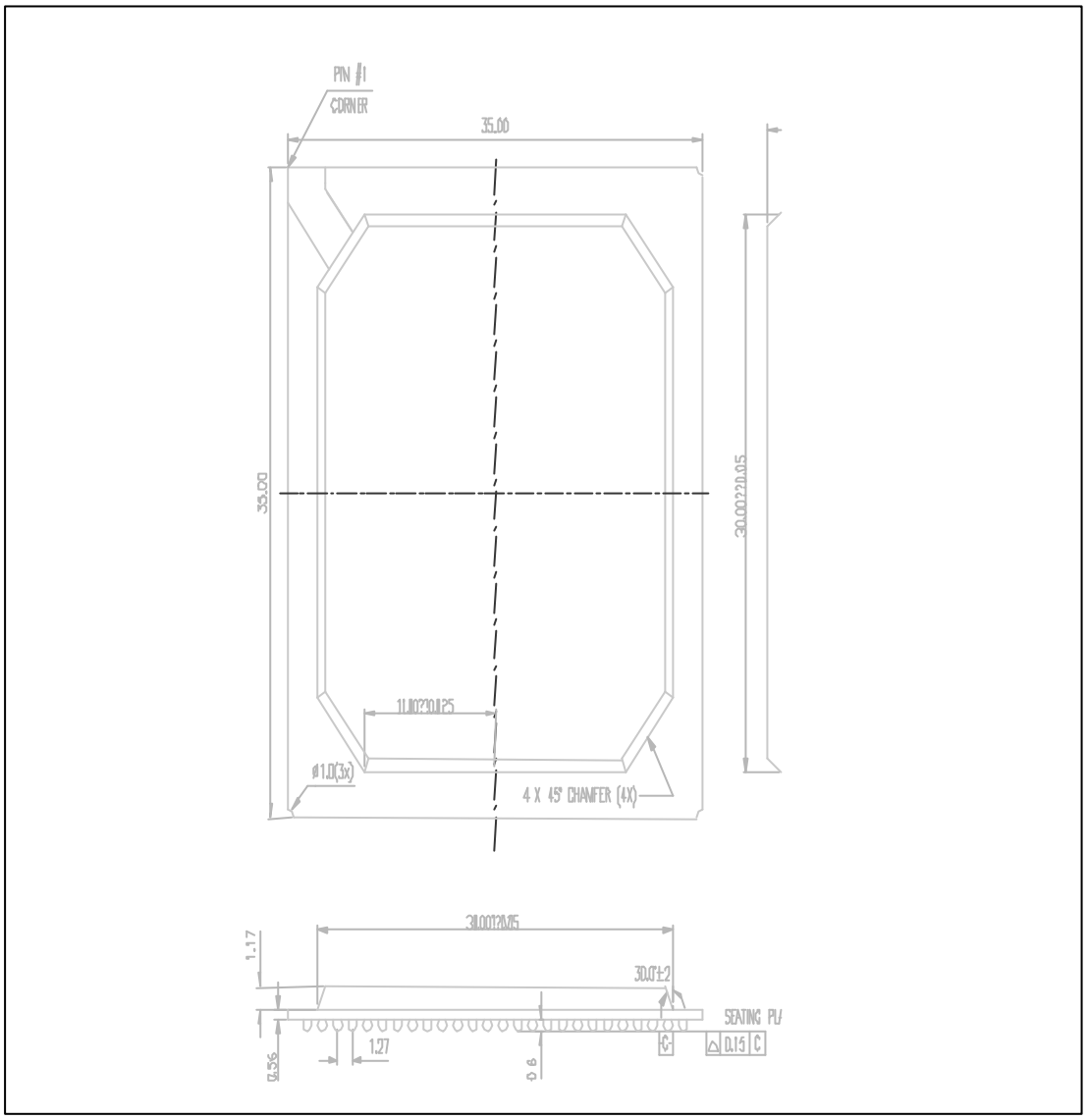



**Figure 10-1 Mechanical Dimension**

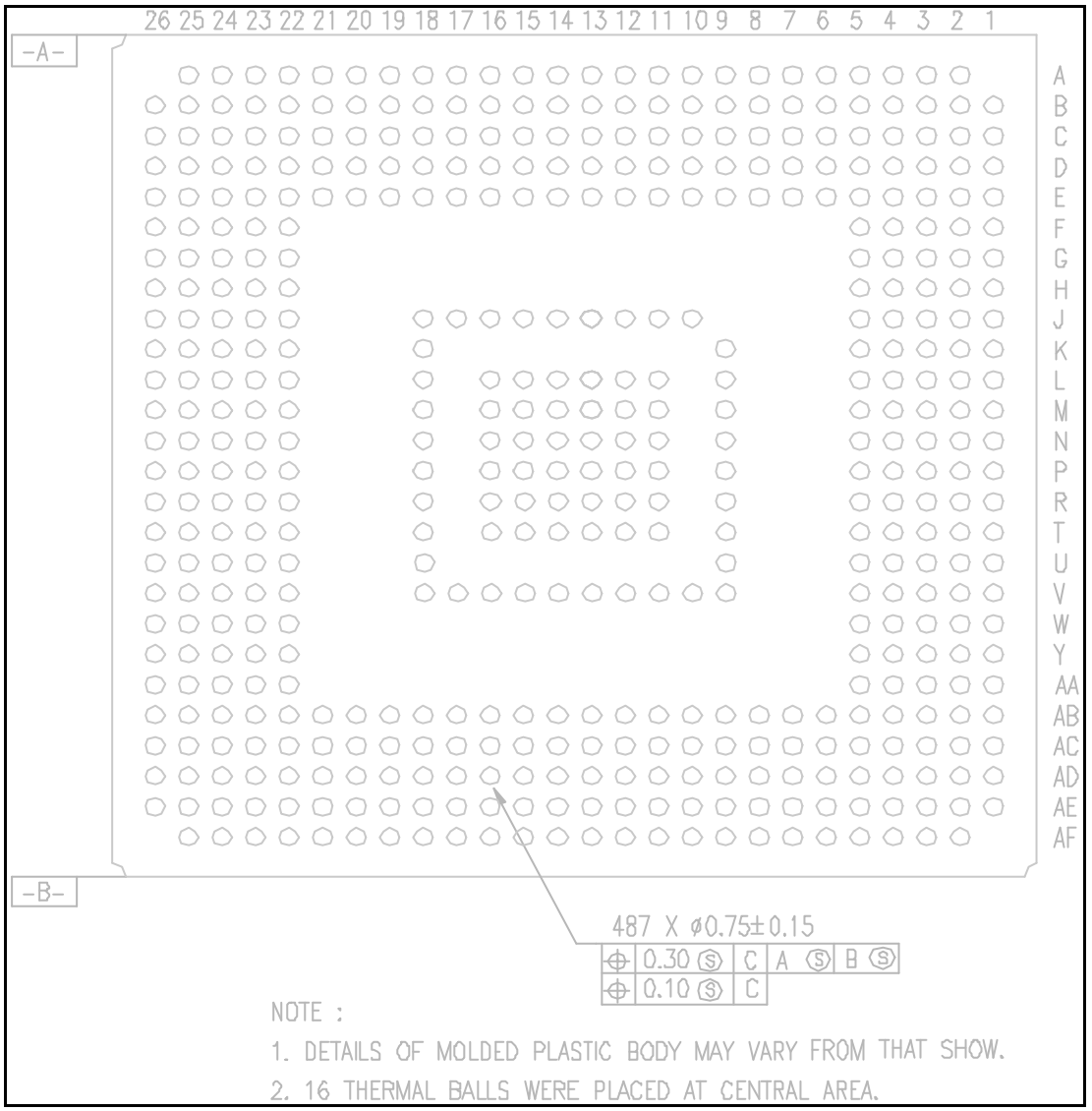



**Figure 10-2 600 Ball Assignment**

# 12. COPYRIGHT NOTICE



**TRADEMARKS**

SiS is a registered trademark of Silicon Integrated Systems Corp.

All brand or product names mentioned are trademarks or registered trademarks of their respective holders.

**DISCLAIMER**

Silicon Integrated Systems Corp. makes no representations or warranties regarding the contents of this manual. We reserve the right to revise the manual or make changes in the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

The information contained in this manual is provided for the general use by our customers. Our customers should be aware that the personal computer field is the subject of many patents. Our customers should ensure that they take appropriate action so that their use of our products does not infringe upon any patents. It is the policy of Silicon Integrated Systems Corp. To respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.

**RESTRICTED RIGHTS LEGEND**

Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.277-7013.

# Pentium II PCI/AGP Chipset

# SiS600

## *Preliminary*

Rev. 1.0

Jan. 25, 1999