SiI3512 PCI to Serial ATA Controller

Data Sheet

Document # DS-0102-D01

November 2010

Export Controlled Document

This document contains information subject to the Export Administration Regulations (EAR) and has a classification of EAR99 or is controlled for Anti-Terrorism (AT) purposes. Transfer of this information by any means to an EAR Country Group E:1 or foreign national thereof (whether in the U.S. or abroad) may require an export license or other approval from the U.S. Department of Commerce. For more information, contact the Silicon Image Director of Global Trade Compliance.

Further Information

To request other materials, documentation, and information, contact your local Silicon Image, Inc. sales office or visit the Silicon Image, Inc. web site at www.siliconimage.com.

Revision History

Revision	Date	Comment
A	01/2004	Derived from Preliminary datasheet Rev 0.98
B	06/2006	Removed SiI3512CT128 (128 pin TQFP standard package without an exposed pad) Corrected inconsistent sentences (minor fixes including mistyping). Updated legal boilerplate
C	11/2006	Datasheet is no longer under NDA, removed confidential markings
D	02/2007	Additional template updates; package markings changed.
D01	11/2010	Added Export Control Statement updated entire layout to the current TechPubs template.

Table of Contents

List of Figures

List of Tables

Overview

The SiI3512 PCI to Serial ATA Controller is a single-chip solution for a PCI to Serial ATA controller. It accepts host commands through the PCI bus, processes them, and transfers data between the host and Serial ATA devices. It can be used to control two independent Serial ATA channels. Each channel has its own Serial ATA bus and will support one Serial ATA device. The SiI3512 controller supports a 32-bit 66-MHz PCI bus and the Serial ATA Generation 1 transfer rate of 1.5 Gbit/s (150 Mbyte/s).

Key Benefits

The SiI3512 controller is the perfect single-chip solution for designs that need to accommodate storage peripherals with the new Serial ATA interface. Any system with a PCI bus interface can simply add the Serial ATA interface by adding a card with the SiI3512 controller and loading the driver into the system.

The SiI3512 controller comes complete with drivers for Windows 98, Windows Millennium, Windows NT 4.0, Windows 2000, XP, Windows 2003, Netware 5.1, 6.0, 6.5, Red Hat Linux 8.0, 9.0, SuSE Linux 8.1, 8.2 and United Linux 1.0.

Features

Overall Features

- Standalone PCI to Serial ATA host controller chip
- Compliant with *PCI Specification, revision 2.3*.
- Compliant with *Programming Interface for Bus Master IDE Controller, revision 1.0*.
- Driver support for Windows 98, Windows Millennium, Windows NT 4.0, Windows 2000, XP, Windows 2003, Netware 5.1, 6.0, 6.5, Red Hat Linux 8.0, 9.0, SuSE Linux 8.1, 8.2 and United Linux 1.0
- Supports up to 4-Mbit external flash or EPROM for BIOS expansion.
- Supports an external EEPROM, flash or EPROM for programmable device ID, subsystem vendor ID, subsystem product ID and PCI sub-class code.
- Supports the Silicon Image specific driver for special chip functions.
- Fabricated in a 0.18µ CMOS process with a 1.8-V core and 3.3-V (5-V tolerant) I/Os.
- Supports Plug and Play.
- Supports ATAPI device
- Supports Activity LEDs, one for each channel with 12-mA open drain driving capability.
- SiI3512ECTU128 is available in a 128-pin TQFP package with e-pad. **EPAD must be soldered to PCB GND.**

PCI Features

- Supports 66-MHz PCI with 32-bit data.
- Supports PCI PERR and SERR reporting.
- Supports PCI bus master operations: Memory Read, Memory Read Multiple, and Memory Write.
- Supports PCI bus target operations: Configuration Read, Configuration Write, I/O Read, I/O Write, Memory Read, Memory Write, Memory Read Line (Memory Read) and Memory Read Multiple (Memory Read)
- Supports byte alignment for odd-byte PCI address access.
- Supports jumper configurable PCI class code.
- Supports programmable and EEPROM, flash and EPROM loadable PCI class code.
- Supports Base Address Register 5 in memory space.

Serial ATA Features

- Integrated Serial ATA Link and PHY logic
- Compliant with Serial ATA 1.0 specifications
- Supports two independent Serial ATA channels and SATA Generation-1 transfer rate of 1.5 Gbit/s.
- Supports Spread Spectrum in receiver
- Single PLL architecture, 1 PLL for both ports
- Programmable drive strengths for Backplane applications

Other Features

- Features independent 256-byte FIFOs (32-bit x 64 deep) per Serial ATA channel for host reads and writes.
- Features Serial ATA to PCI interrupt masking.
- Features Watchdog Timer for fault resiliency.

Applications

- PC motherboards
- Serial ATA drive add-on cards
- Serial ATA RAID controllers

References

For more details about Serial ATA technology, the reader is referred to the following industry specifications:

- *Serial ATA / High Speed Serialized AT Attachment specification, Revision 1.0*
- *PCI Local Bus Specification Revision 2.3*
- *Advanced Power Management Specification Revision 1.0*
- *PCI IDE Controller Specification Revision 1.0*
- *Programming Interface for Bus Master IDE Controller, Revision 1.0*

Functional Description

The SiI3512 controller is a PCI to Serial ATA controller chip that transfers data between the PCI bus and storage media (e.g., hard disk drive). The SiI3512 controller consists of the following functional blocks:

- PCI Interface. Provides the interface to any system that has a PCI bus. Instructions and system clocks are based on this interface.
- Serial ATA Interface. Two separate channels (primary and secondary) to access storage media such as hard disk drive, floppy disk drive, and CD-ROM.

Functional Block Diagram

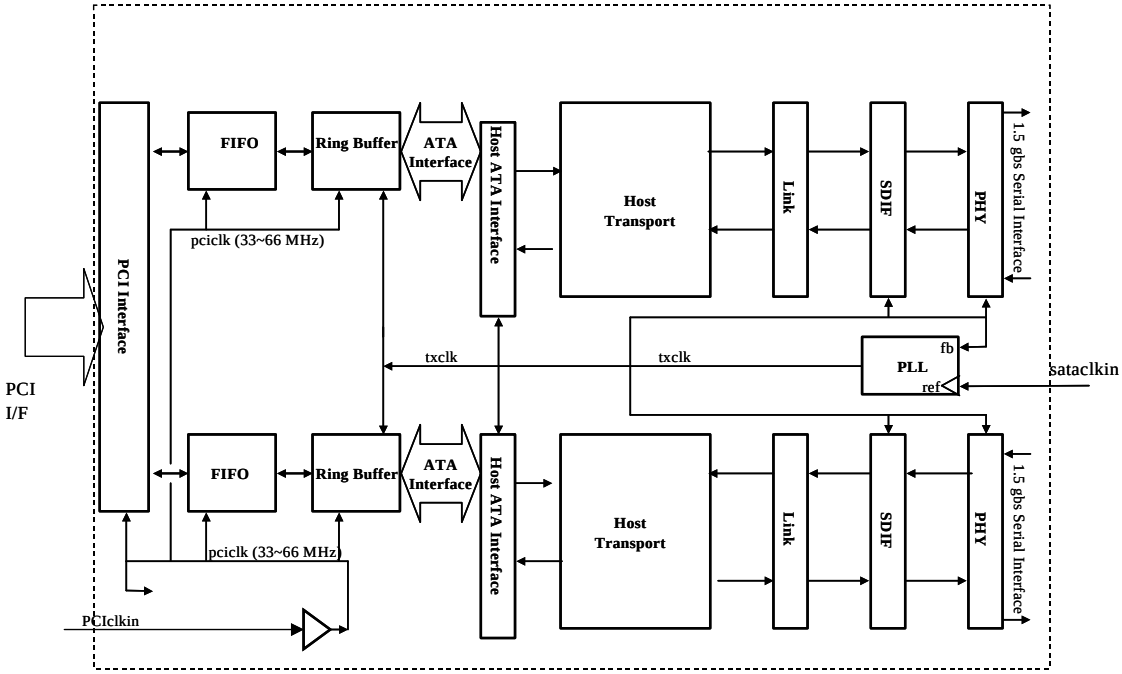


Figure 1. SiI3512 Block Diagram

PCI Interface

The SiI3512 controller PCI interface is compliant with the *PCI Local Bus Specification (Revision 2.3)*. The SiI3512 device can act as a PCI master and a PCI slave, and contains the SiI3512 controller PCI configuration space and internal registers. When the SiI3512 needs to access shared memory, it becomes the bus master of the PCI bus and completes the memory cycle without external intervention. In the mode when it acts as a bridge between the PCI bus and the Serial ATA bus, it will behave as a PCI slave.

PCI Initialization

Generally, when a system initializes a module containing a PCI device, the configuration manager reads the configuration space of each PCI device on the PCI bus. Hardware signals select a specific PCI device based on a bus number, a slot number, and a function number. If a device that is addressed (via signal lines) responds to the configuration cycle by claiming the bus, then that function's configuration space is read out from the device during the cycle. Because any PCI device can be a multifunction device, every supported function's configuration space needs to be read from the device. Based on the information read, the configuration manager will assign system resources to each supported function within the device. Sometimes new information needs to be written into the function's configuration space. This is accomplished with a configuration write cycle.

PCI Bus Operations

The SiI3512 controller behaves either as a PCI master or a PCI slave device at any time and switches between these modes as required during device operation.

As a PCI slave, the SiI3512 controller responds to the following PCI bus operations:

- I/O Read
- I/O Write
- Configuration Read
- Configuration Write
- Memory Read
- Memory Write

All other PCI cycles are ignored by the SiI3512 controller.

As a PCI master, the SiI3512 controller generates the following PCI bus operations:

- Memory Read Multiple
- Memory Read
- Memory Write

PCI Configuration Space

This section describes how the SiI3512 controller implements the required PCI configuration register space. The intent of the PCI configuration space definition is to provide an appropriate set of configuration registers that satisfy the needs of current and anticipated system configuration mechanisms, without specifying those mechanisms or otherwise placing constraints on their use. These registers allow for:

- Full device relocation (including interrupt binding)
- Installation, configurations, and booting without user interventions
- System address map construction by device-independent software

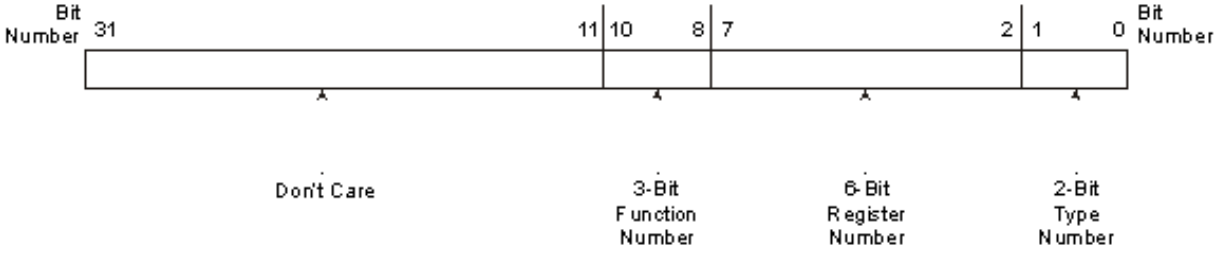


Figure 2. Address Lines During Configuration Cycle

The SiI3512 controller responds only to Type 0 configuration cycles. Type 1 cycles, which pass a configuration request on to another PCI bus, are ignored.

The address phase during a SiI3512 configuration cycle indicates the function number and register number being addressed which can be decoded by observing the status of the address lines AD[31:0] (see Figure 2 above).

The value of the signal lines AD[7:2] during the address phase of configuration cycles selects the register of the configuration space to access. Valid values are between 0 and 15, inclusive. Accessing registers outside this range results in an all-0s value being returned on reads, and no action being taken on writes.

The Class Code register contains the Class Code, Sub-Class Code, and Register-Level Programming Interface registers.

All writable bits in the configuration space are reset to 0 by the hardware reset, PCI RESET (RST#) asserted. After reset, the SiI3512 is disabled and will only respond to PCI configuration write and PCI configuration read cycles.

Deviations from the Specification

The SiI3512 product has been developed and tested to the specification listed in this document. As a result of testing and customer feedback, we may become aware of deviations to the specification that could affect the component's operation. It is important that you have the most current version of this specification. If there are any questions, please contact Silicon Image, Inc.

Electrical Characteristics

Device Electrical Characteristics

Specifications are for Commercial Temperature range, 0°C to +70°C, unless otherwise specified.

Table 1. Absolute Maximum Ratings

Symbol	Parameter	Ratings	Unit
VDDO, VDDX	I/O and Oscillator Supply Voltage	4.0	V
VDDI,VDDA,VDDP	Digital, Analog and PLL Supply Power	2.15	V
V_{PCI_IN}	Input Voltage for PCI signals	-0.3 ~ 6.0	V
V_{NONPCI_IN}	Input Voltage for Non-PCI signals	-0.3 ~ VDDO+0.3	V
V_{CLK_IN}	Input Voltage for CLKI	-0.3 ~ VDDX+0.3	V
I_{OUT}	DC Output Current	16	mA
θ_{JA}	Thermal Resistance (Junction to Ambient)	46.3[1]	°C/W
T_{STG}	Storage Temperature	-65 ~ 150	°C

Notes: This number will decrease by soldering an EPAD to PCB GND.

Table 2. DC Specifications

Symbol	Parameter	Condition	Type	Limits			Unit
				Min	Typ	Max	
VDDI, VDDA, VDDP	Supply Voltage (Digital, Analog, PLL)			1.71	1.8	1.89	V
VDDO	Supply Voltage(I/O)	-	-	3.0	3.3	3.6	V
$IDD_{1.8V}$	1.8-V Supply Current				236[1]	330[2]	mA
$IDD_{3.3V}$	3.3-V Supply Current	C_{LOAD}= 20pF			12[1]	30[2]	mA
V_{IH}	Input High Voltage	-	3.3-V PCI	0.5xVDDO	-	-	V
		-	Non-PCI	2.0	-	-	
V_{IL}	Input Low Voltage	-	3.3-V PCI	-	-	0.3xVDDO	V
		-	Non-PCI	-	-	0.8	
V_{OH}	Output High Voltage	I_{OUT} = -500µA	3.3-V PCI	0.9xVDDO	-	-	V
		-	Non-PCI	2.4	-	-	
V_{OL}	Output Low Voltage	I_{OUT} = 1500µA	3.3-V PCI	-	-	0.1xVDDO	V
		-	Non-PCI	-	-	0.4	
V+	Input High Voltage	-	Schmitt	-	1.8	2.3	V
V-	Input Low Voltage	-	Schmitt	0.5	0.9	-	V
V_H	Hysteresis Voltage	-	Schmitt	0.4	-	-	V
I_{IH}	Input High Current	V_{IN} = VDD	-	–10	-	10	µA
I_{IL}	Input Low Current	V_{IN} = VSS	-	–10	-	10	µA
I_{ILOD}	Open Drain output sync current					12	mA
I_{OZ}	3-State Leakage Current	-	-	–10	-	10	µA

Notes:

1. Using the random data pattern (read/write operation) at 1.8-V or 3.3-V power supply, PCI interface = 33 MHz
2. Using the maximum toggling data pattern (read/write operation) at 1.89-V or 3.6-V power supply, PCI interface = 66 MHz.

Table 3. SATA Interface DC Specifications

Symbol	Parameter	Condition	Limits			Unit
			Min	Typ	Max	
V_{DOUT_00}	TX+/TX- differential peak-to-peak voltage swing.	Terminated by 50 Ω. Tx Swing Value = 00	400	500	600	mV
V_{DOUT_01}	TX+/TX- differential peak-to-peak voltage swing.	Terminated by 50 Ω. Tx Swing Value = 01	500	600	700	mV
V_{DOUT_10}	TX+/TX- differential peak-to-peak voltage swing.	Terminated by 50 Ω. Tx Swing Value = 10	550	700	800	mV
V_{DOUT_11}	TX+/TX- differential peak-to-peak voltage swing.	Terminated by 50 Ω. Tx Swing Value = 11	650	800	900	mV
V_{DIN}	RX+/RX- differential peak-to-peak input sensitivity		325			mV
V_{DIH}	RX+/RX- differential Input common-mode voltage		200	300	450	mV
V_{DOH}	TX+/TX-differential Output common-mode voltage		200	300	450	mV
V_{SDT}	Squelch detector threshold		100	50	200	mV
Z_{DIN}	Differential input impedance	REXT = 1k 1% for 25-MHz SerDes Ref Clk REXT = 4.99k 1% for 100-MHz SerDes Ref Clk	85	100	115	Ω
Z_{DOUT}	Differential output impedance	REXT = 1k 1% for 25-MHz SerDes Ref Clk REXT = 4.99k 1% for 100-MHz SerDes Ref Clk	85	100	115	Ω

SATA Interface Timing Specifications

Table 4. SATA Interface Timing Specifications

Symbol	Parameter	Condition	Limits			Unit
			Min	Typ	Max	
$T_{TX\_RISE\_FALL}$	Rise and Fall time at transmitter	20%–80%	133		274	ps
T_{TX_SKEW}	Tx differential skew				20	ps
$T_{TX\_DC\_FREQ}$	Tx DC clock frequency skew		–350		+350	ppm
$T_{TX\_AC\_FREQ}$	Tx AC clock frequency skew	SerDes Ref Clk = SSC AC modulation, subject to the "Downspread SSC" triangular modulation (30–33 kHz) profile per 6.6.4.5 in SATA 1.0 specification	–5000		+0	ppm

SATA Interface Transmitter Output Jitter Characteristics

Table 5. SATA Interface Transmitter Output Jitter Characteristics – SiI3512ECTU128

Symbol	Parameter	Condition	Limits			Unit
			Min	Typ	Max	
RJ5UI	5UI later Random Jitter	Measured at Tx output pins 1-sigma deviation		7.0		ps rms
RJ250UI	250UI later Random Jitter	Measured at Tx output pins 1-sigma deviation		9.5		ps rms
DJ5UI	5UI later Deterministic Jitter	Measured at Tx output pins peak to peak phase variation Random data pattern		33		ps
DJ250UI	250UI later Deterministic Jitter	Measured at Tx output pins peak to peak phase variation Random data pattern		34		ps

Note: EPAD is soldered to a landing area on the PCB.

CLKI SerDes Reference Clock Input Requirements

Table 6. CLKI SerDes Reference Clock Input Requirements

Symbol	Parameter	Condition	Limits			Unit
			Min	Typ	Max	
T_{CLKI_FREQ}	Nominal Frequency	REXT = 1k 1% REXT = 4.99k 1%		25 100		MHz
T_{CLKI_J}	CLKI frequency tolerance		–100		+100	ppm
$T_{CLKI\_RISE\_FALL}$	Rise and Fall time at CLKI	25-MHz reference clock, 20%–80% 100-MHz reference clock, 20%–80%			4 2	ns
$T_{CLKI\_RC\_DUTY}$	CLKI duty cycle	20%–80%	40		60	%

PCI 33-MHz Timing Specifications

Table 7. PCI 33-MHz Timing Specifications

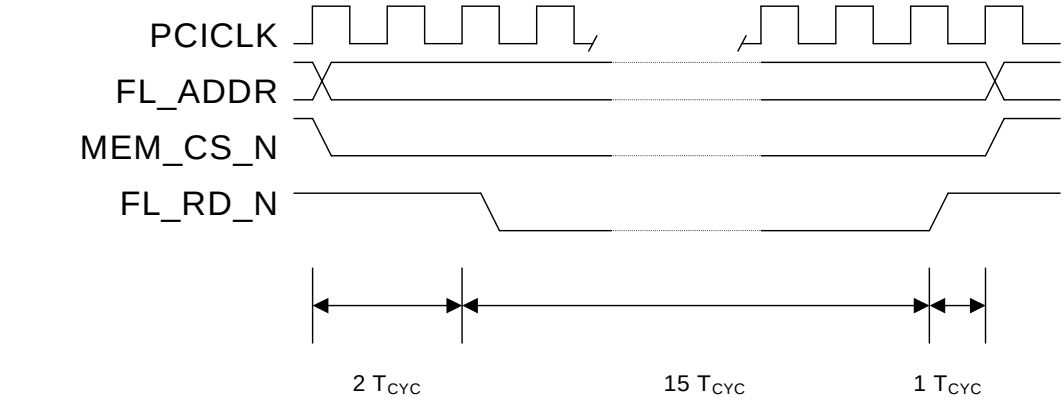
Symbol	Parameter	Limits		Unit
		Min	Max	
T_{VAL}	CLK to Signal Valid – Bussed Signals	2.0	11.0	ns
$T_{VAL\ (PTP)}$	CLK to Signal Valid – Point to Point	2.0	11.0	ns
T_{ON}	Float to Active Delay	2.0	-	ns
T_{OFF}	Active to Float Delay	-	28.0	ns
T_{SU}	Input Setup Time – Bussed Signals	7.0	-	ns
$T_{SU\ (PTP)}$	Input Setup Time – Point to Point	10.0	-	ns
T_{H}	Input Hold Time	0.0	-	ns

PCI 66-MHz Timing Specifications

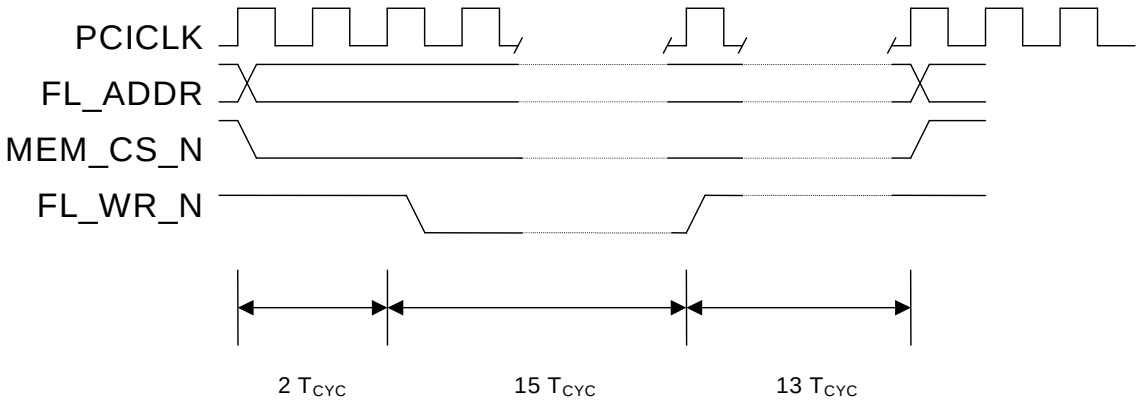
Table 8. PCI 66-MHz Timing Specifications

Symbol	Parameter	Limits		Unit
		Min	Max	
T_{VAL}	CLK to Signal Valid – Bussed Signals	2.0	6.0	ns
$T_{VAL\ (PTP)}$	CLK to Signal Valid – Point to Point	2.0	6.0	ns
T_{ON}	Float to Active Delay	2.0		ns
T_{OFF}	Active to Float Delay		14.0	ns
T_{SU}	Input Setup Time – Bussed Signals	3.0		ns
$T_{SU\ (PTP)}$	Input Setup Time – Point to Point	5.0		ns
T_{H}	Input Hold Time	0.0		ns

Flash Memory Timing Specifications



FLASH READ TIMING



FLASH WRITE TIMING

Figure 3. Flash Memory Timing

Pin Definitions

Pin Listing

This section describes the pin-out of the SiI3512 PCI to Serial ATA host controller. Table 9 gives the pin numbers, pin names, pin types, drive types where applicable, internal resistors where applicable, and descriptions.

Table 9. Pin Listing

Pin #	Pin Name	Type	Internal Resistor	Description
1	PCI_AD00	I/O	-	PCI Address/Data
2	EEPROM_SDAT	I/O	PU – 70k	EEPROM Serial Data
3	EEPROM_SCLK	I/O	PU – 70k	EEPROM Serial Clock
4	VDDA	PWR	-	1.8-V SerDes Power
5	RxP2	I		Channel 2 Differential Receive +ve
6	RxN2	I		Channel 2 Differential Receive -ve
7	GNDA	GND	-	Analog Ground
8	VDDA	PWR	-	1.8-V SerDes Power
9	GNDA	GND	-	Analog Ground
10	TxN2	O		Channel 2 Differential Transmit -ve
11	TxP2	O		Channel 2 Differential Transmit +ve
12	GNDA	GND	-	Analog Ground
13	VDDA	PWR	-	1.8-V SerDes Power
14	RxP1	I		Channel 1 Differential Receive +ve
15	RxN1	I		Channel 1 Differential Receive -ve
16	GNDA	GND	-	Analog Ground
17	VDDA	PWR	-	1.8-V SerDes Power
18	GNDA	GND	-	Analog Ground
19	TxN1	O		Channel 1 Differential Transmit -ve
20	TxP1	O		Channel 1 Differential Transmit +ve
21	GNDA	GND	-	Analog Ground
22	VDDP	PWR	-	1.8-V PLL Power
23	VDDP	PWR	-	1.8-V PLL Power
24	REXT	I		External Reference Resistor Input
25	GNDA	GND	-	Analog Ground
26	XTALI/CLKI	I		Crystal Oscillator Input or external clock input
27	XTALO	O		Crystal Oscillator Output
28	VDDO	PWR		3.3-V supply for Crystal Oscillator
29	SCAN_EN	I	PD – 60k	Internal Scan Enable
30	MEM_CS_N	O	PU – 70k	Memory Chip Select
31	FL_ADDR[00] / IDE_CFG	I/O	PU – 70k	Flash Memory Address 0 / IDE-RAID Configuration
32	FL_ADDR[01] / BA5_EN	I/O	PU – 70k	Flash Memory Address 1 / Base Address Register 5 Enable
33	FL_ADDR[02]	O	PU – 70k	Flash Memory Address 2
34	FL_WR_N	O	PU – 70k	Flash Memory Write Strobe
35	FL_RD_N	O	PU – 70k	Flash Memory Read Strobe
36	FL_ADDR[03]	O	PU – 70k	Flash Memory Address 3
37	FL_ADDR[04]	O	PU – 70k	Flash Memory Address 4
38	FL_ADDR[05]	O	PU – 70k	Flash Memory Address 5
39	FL_ADDR[06]	O	PU – 70k	Flash Memory Address 6
40	VDDO	PWR	-	3.3 Volt Power

Pin #	Pin Name	Type	Internal Resistor	Description
41	VSS	GND	-	Ground
42	FL_ADDR[07]	O	PU – 70k	Flash Memory Address 7
43	FL_ADDR[08]	O	PU – 70k	Flash Memory Address 8
44	FL_ADDR[09]	O	PU – 70k	Flash Memory Address 9
45	FL_ADDR[10]	O	PU – 70k	Flash Memory Address 10
46	FL_ADDR[11]	O	PU – 70k	Flash Memory Address 11
47	VDDI	PWR	-	1.8-V Internal core Power
48	VSS	GND	-	Ground
49	FL_ADDR[12]	O	PU – 70k	Flash Memory Address 12
50	FL_ADDR[13]	O	PU – 70k	Flash Memory Address 13
51	LED0	OD		Channel 0 activity LED indicator
52	FL_ADDR[14]	O	PU – 70k	Flash Memory Address 14
53	FL_ADDR[15]	O	PD – 60k	Flash Memory Address 15
54	FL_ADDR[16]	O	PD – 60k	Flash Memory Address 16
55	FL_ADDR[17]	O	PD – 60k	Flash Memory Address 17
56	FL_ADDR[18]	O	PD – 60k	Flash Memory Address 18
57	TEST_MODE	I	PD – 60k	Test Mode Enable
58	FL_DATA[00]	I/O	PU – 70k	Flash Memory Data 0
59	FL_DATA[01]	I/O	PU – 70k	Flash Memory Data 1
60	FL_DATA[02]	I/O	PU – 70k	Flash Memory Data 2
61	FL_DATA[03]	I/O	PU – 70k	Flash Memory Data 3
62	FL_DATA[04]	I/O	PU – 70k	Flash Memory Data 4
63	FL_DATA[05]	I/O	PU – 70k	Flash Memory Data 5
64	FL_DATA[06]	I/O	PU – 70k	Flash Memory Data 6
65	FL_DATA[07]	I/O	PU – 70k	Flash Memory Data 7
66	LED1	OD		Channel 1 activity LED indicator
67	PCI_INTA_N	OD	-	PCI Interrupt
68	PCI_RST_N	I-Schmitt	-	PCI Reset
69	PCI_CLK	I	-	PCI Clock
70	PCI_GNT_N	I	-	PCI Bus Grant
71	PCI_REQ_N	T	-	PCI Bus Request
72	PCI_AD31	I/O	-	PCI Address/Data
73	PCI_AD30	I/O	-	PCI Address/Data
74	PCI_AD29	I/O	-	PCI Address/Data
75	VDDO	PWR	-	3.3-V Power
76	VSS	GND	-	Ground
77	PCI_AD28	I/O	-	PCI Address/Data
78	PCI_AD27	I/O	-	PCI Address/Data
79	PCI_AD26	I/O	-	PCI Address/Data
80	VDDI	PWR	-	1.8-V Internal Core Power
81	VSS	GND	-	Ground
82	PCI_AD25	I/O	-	PCI Address/Data
83	PCI_AD24	I/O	-	PCI Address/Data
84	PCI_CBE3	I/O	-	PCI Command/Byte Enable
85	PCI_IDSEL	I	PU-70k	PCI ID Select
86	PCI_AD23	I/O	-	PCI Address/Data
87	PCI_AD22	I/O	-	PCI Address/Data
88	PCI_AD21	I/O	-	PCI Address/Data
89	PCI_AD20	I/O	-	PCI Address/Data
90	PCI_AD19	I/O	-	PCI Address/Data

Pin #	Pin Name	Type	Internal Resistor	Description
91	VDDO	PWR	-	3.3-V Power
92	VSS	GND	-	Ground
93	PCI_AD18	I/O	-	PCI Address/Data
94	PCI_AD17	I/O	-	PCI Address/Data
95	PCI_AD16	I/O	-	PCI Address/Data
96	PCI_CBE2	I/O	-	PCI Command/Byte Enable
97	PCI_FRAME_N	I/O	-	PCI Frame
98	PCI_IRDY_N	I/O	-	PCI Initiator Ready
99	PCI_PERR_N	I/O	-	PCI Parity Error
100	PCI_STOP_N	I/O	-	PCI Stop
101	PCI_DEVSEL_N	I/O	-	PCI Device Select
102	PCI_TRDY_N	I/O	-	PCI Target Ready
103	PCI_SERR_N	OD	-	PCI System Error
104	PCI_PAR	I/O	-	PCI Parity
105	PCI_CBE1	I/O	-	PCI Command/Byte Enable
106	VDDO	PWR	-	3.3-V Power
107	VSS	GND	-	Ground
108	PCI_AD15	I/O	-	PCI Address/Data
109	PCI_AD14	I/O	-	PCI Address/Data
110	PCI_AD13	I/O	-	PCI Address/Data
111	PCI_AD12	I/O	-	PCI Address/Data
112	VDDI	PWR	-	1.8-V Core Power
113	VSS	GND	-	Ground
114	PCI_AD11	I/O	-	PCI Address/Data
115	PCI_AD10	I/O	-	PCI Address/Data
116	PCI_M66EN	I	PU-70k	PCI 66-MHz Enable
117	PCI_AD09	I/O	-	PCI Address/Data
118	PCI_AD08	I/O	-	PCI Address/Data
119	PCI_CBE0	I/O	-	PCI Command/Byte Enable
120	PCI_AD07	I/O	-	PCI Address/Data
121	PCI_AD06	I/O	-	PCI Address/Data
122	VDDO	PWR	-	3.3-V Power
123	VSS	GND	-	Ground
124	PCI_AD05	I/O	-	PCI Address/Data
125	PCI_AD04	I/O	-	PCI Address/Data
126	PCI_AD03	I/O	-	PCI Address/Data
127	PCI_AD02	I/O	-	PCI Address/Data
128	PCI_AD01	I/O	-	PCI Address/Data

Table 10. Pin Types

Pin Type	Description
I	Input Pin with LVTTL Thresholds
I-Schmitt	Input Pin with Schmitt Trigger
O	Output Pin
T	Tri-state Output Pin
I/O	Bi-directional Pin
OD	Open Drain Output Pin
Note: PCI pins are 5-V tolerant.	

Pin Diagram

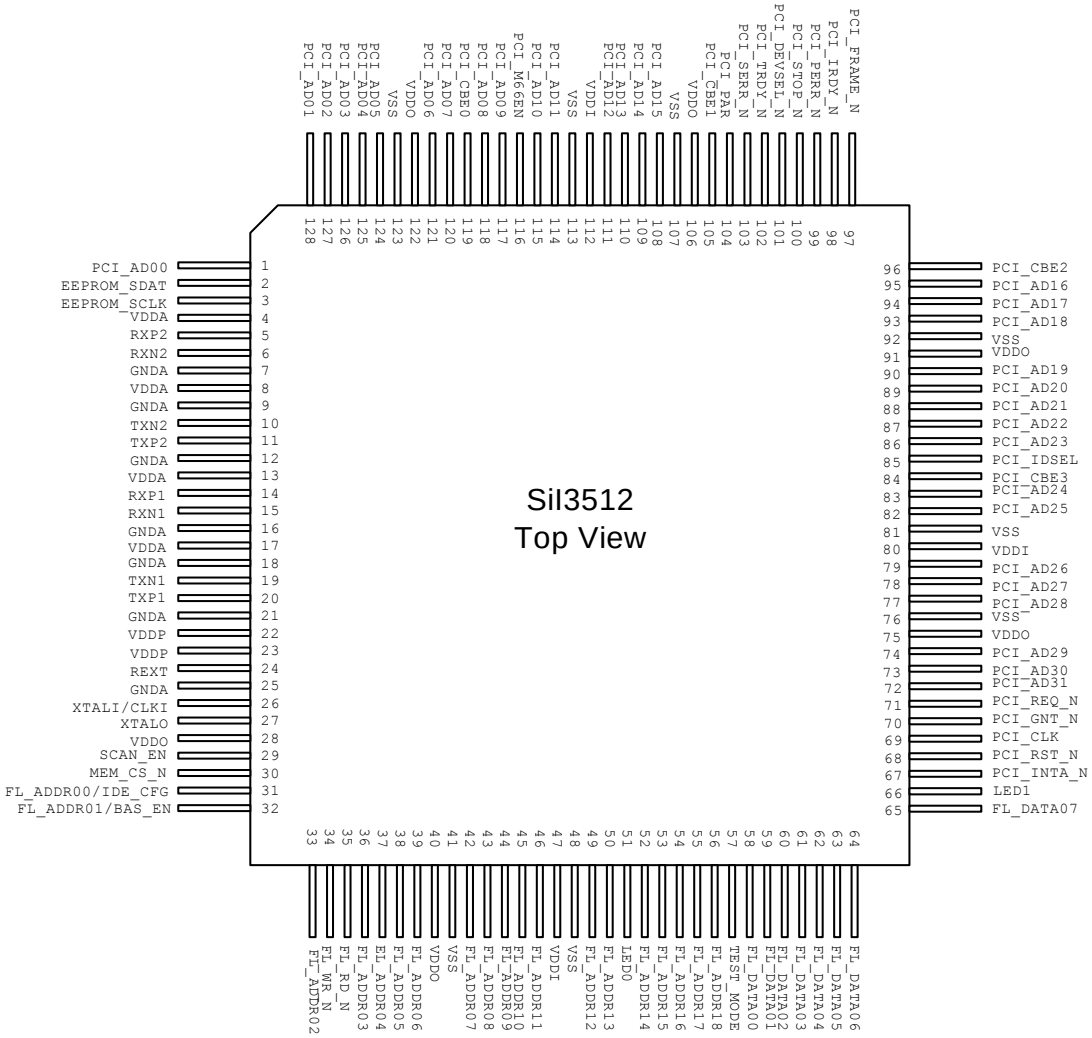


Figure 4. SiI3512 Pin Diagram

Pin Descriptions

66-MHz 32-bit PCI Pins

PCI Address and Data

Pin Names: PCI_AD[31..0]
Pin Numbers: 72~74, 77~79, 82~83, 86~90, 93~95, 108~111, 114~115, 117~118, 120~121, 124~128, 1
Address and Data buses are multiplexed on the same PCI pins. A bus transaction consists of an address phase followed by one or more data phases. PCI supports both read and write bursts. The address phase is the first clock cycle in which PCI_FRAME_N signal is asserted. During the address phase, PCI_AD[31:0] contain a physical address (32 bits). For I/O, this can be a byte address. For configuration and memory it is a dword address. During data phases, PCI_AD[7:0] contain the least significant byte (LSB) and PCI_AD[31:24] contain the most significant byte (MSB). Write data is stable and valid when PCI_IRDY_N is asserted; read data is stable and valid when PCI_TRDY_N is asserted. Data is transferred during those clocks where both PCI_IRDY_N and PCI_TRDY_N are asserted.

PCI Command and Byte Enables

Pin Names: PCI_CBE[3..0]
Pin Numbers: 84, 96, 105, 119
Command and Byte Enables are multiplexed on the same PCI pins. During the address phase of a transaction, PCI_CBE[3:0]_N define the bus command. During the data phase, PCI_CBE[3:0]_N are used as Byte Enables. Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data.

PCI ID Select

Pin Name: PCI_IDSEL
Pin Number: 85
This signal is used as a chip select during configuration read and write transactions.

PCI Frame Cycle

Pin Name: PCI_FRAME_N
Pin Number: 97
Cycle Frame is driven by the current master to indicate the beginning and duration of an access. PCI_FRAME_N is asserted to indicate that a bus transaction is beginning. While PCI_FRAME_N is asserted, data transfers continue. When PCI_FRAME_N is de-asserted, the transaction is in the final data phase or has completed.

PCI Initiator Ready

Pin Name: PCI_IRDY_N
Pin Number: 98
Initiator Ready indicates the initializing agent's (bus master's) ability to complete the current data phase of the transaction. This signal is used with PCI_TRDY_N. A data phase is completed on any clock when both PCI_IRDY_N and PCI_TRDY_N are sampled as asserted. Wait cycles are inserted until both PCI_IRDY_N and PCI_TRDY_N are asserted together.

PCI Target Ready

Pin Name: PCI_TRDY_N
Pin Number: 102
Target Ready indicates the target agent's ability to complete the current data phase of the transaction. PCI_TRDY_N is used with PCI_IRDY_N. A data phase is completed on any clock when both PCI_TRDY_N and PCI_IRDY_N are sampled asserted. During a read, PCI_TRDY_N indicates that valid data is present on PCI_AD[31:0]. During a write, it indicates the target is prepared to accept data.

PCI Device Select

Pin Name: PCI_DEVSEL_N
Pin Number: 101
Device Select, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, PCI_DEVSEL_N indicates to a master whether any device on the bus has been selected.

PCI Stop

Pin Name: PCI_STOP_N
Pin Number: 100
PCI_STOP_N indicates the current target is requesting that the master stop the current transaction.

PCI Parity Error

Pin Name: PCI_PERR_N
Pin Number: 99
PCI_PERR_N indicates a data parity error between the current master and target on PCI. On a write transaction, the target always signals data parity errors back to the master on PCI_PERR_N. On a read transaction, the master asserts PCI_PERR_N to indicate to the system that an error was detected.

PCI System Error

Pin Name: PCI_SERR_N
Pin Number: 103
System Error is for reporting address parity errors, data parity errors on Special Cycle Command, or any other system error where the result will be catastrophic. The PCI_SERR_N is a pure open drain and is actively driven for a single PCI clock by the agent reporting the error. The assertion of PCI_SERR_N is synchronous to the clock and meets the setup and hold times of all bused signals. However, the restoring of PCI_SERR_N to the de-asserted state is accomplished by a weak pull-up. Note that if an agent does not want a non-maskable interrupt (NMI) to be generated, a different reporting mechanism is required.

PCI Parity

Pin Name: PCI_PAR
Pin Number: 104
PCI_PAR is even parity across PCI_AD[31:0] and PCI_CBE[3:0]_N. Parity generation is required by all PCI agents. PCI_PAR is stable and valid one clock after the address phase. For data phases PCI_PAR is stable and valid one clock after either PCI_IRDY_N is asserted on a write transaction or PCI_TRDY_N is asserted on a read transaction. Once PCI_PAR is valid, it remains valid until one clock after the completion of the current data phase. (PCI_PAR has the same timing as PCI_AD[31:0] but delayed by one clock.)

PCI Request

Pin Name: PCI_REQ_N
Pin Number: 71
This signal indicates to the arbiter that this agent desires use of the PCI bus.

PCI Grant

Pin Name: PCI_GNT_N
Pin Number: 70
This signal indicates to the agent that access to the PCI bus has been granted. In response to a PCI request, this is a point-to-point signal. Every master has its own PCI_GNT_N, which must be ignored while PCI_RST_N is asserted.

PCI Interrupt A

Pin Name: PCI_INTA_N
Pin Number: 67
Interrupt A is used to request an interrupt on the PCI bus. PCI_INTA_N is open collector and is an open drain output.

PCI Clock Signal

Pin Names: PCI_CLK
Pin Number: 69
Clock Signal provides timing for all transactions on PCI and is an input to every PCI device. All other PCI signals (except PCI_RST_N, and PCI_INTA_N) are sampled on the rising edge of PCI_CLK. All other timing parameters are defined with respect to this edge.

PCI Reset

Pin Name: PCI_RST_N
Pin Number: 68
PCI_RST_N is an active low input that is used to set the internal registers to their initial state. PCI_RST_N is typically the system power-on reset signal as distributed on the PCI bus.

PCI M66EN

Pin Name: PCI_M66EN
Pin Number: 116
This pin configures the PCI bus operating frequency. When low, the PCI bus operates from 0 to 33 MHz. When high, the PCI bus operates from 33 MHz to 66 MHz.

Miscellaneous I/O Pins

Ground

Pin Name: VSS
Pin Number: 41, 48, 76, 81, 92, 107, 113, and 123
Logic Ground. This ground pins are connected with GNDA (SerDes Ground) with an EPAD.

TEST

Pin Name: TEST_MODE
Pin Number: 57
This pin is used, in conjunction with other pins, to enable various test functions within the device.

Power Supply

Pin Name(s): VDDO
Pin Number(s): 28, 40, 75, 91, 106, and 122
Power Supply Input.

Pin Name(s): VDDI
Pin Number(s): 47, 80, and 112
Power Supply Input for internal core.

Internal Scan Test

Pin Name: SCAN_EN
Pin Number: 29
This pin, when active (high), will place all scan flip-flops into scan mode for chip testing. This pin must be left open or tied to ground for normal operation.

LED Drivers

Pin Names: LED[0..1]
Pin Numbers: 51, 66
These are 12-mA open-drain outputs to drive Activity LEDs for IDE channels 0 and 1, respectively.

Flash Signals

Pin Name: FL_ADDR[00] / IDE_CFG

Pin Number: 31
When PCI_RST_N is de-asserted, this pin is an output and represents flash memory address bit 0. During reset, it is sampled to configure Mass Storage class or RAID mode in the PCI Class Code register. A high on this pin sets Mass Storage class, a low sets RAID mode. The configuration state is latched internally when PCI_RST_N is de-asserted. This pad is internally pulled high to enable Mass Storage class if left unconnected.

Pin Name: FL_ADDR[01] / BA5_EN
Pin Number: 32
When PCI_RST_N is de-asserted, this pin is an output and represents flash memory address bit 1. During reset, it is sampled to configure Base Address Register 5. A high on this pin enables Base Address Register 5, a low disables Base Address Register 5. The configuration state is latched internally when PCI_RST_N is de-asserted. This pin is internally pulled high to enable Base address register 5 when left unconnected.

Pin Name: FL_ADDR[02-18]
Pin Numbers: 33, 36~39, 42~46, 49~50, 52~56
Flash Memory address bits; 19 total for 512k address space. Flash address pins 15 to 18 are used to select internal test modes in conjunction with the TEST_MODE pin; they have internal pull-downs and must be unconnected or pulled down.

Pin Name: FL_DATA[00-07]
Pin Numbers: 58~65
8-bit flash memory data bus.

Pin Name: FL_RD_N
Pin Number: 35
Flash read enable signal, active low.

Pin Name: FL_WR_N
Pin Number: 34
Flash write enable signal, active low.

Memory Chip Select

Pin Name: MEM_CS_N
Pin Number: 30
This pin is used to select and enable the external memory. It is active low.

Serial Interface Signals

Pin Name: EEPROM_SDAT
Pin Number: 2
Serial Interface data line.

Pin Name: EEPROM_SCLK
Pin Number: 3
Serial Interface clock.

Serial ATA Signals

Power Supply & Ground

Pin Name: VDDA
Pin Numbers: 4, 8, 13, 17
SerDes 1.8-V power supply pins.

Pin Name: VDDP
Pin Numbers: 22, 23
PLL 1.8-V power supply pins.

Pin Name: GNDA
Pin Numbers: 7, 9, 12, 16, 18, 21, and 25
SerDes Ground. These ground pins are connected with VSS (Logic Ground) with an EPAD.

High-Speed Serial Signals

Pin Name: RxN1
Pin Number: 15
Channel 1 high-speed differential receive negative side.

Pin Name: RxP1
Pin Number: 14
Channel 1 high-speed differential receive positive side. Loading an internal register through the flash or EEPROM during the initialization sequence could reverse RxP1 and RxN1 pinouts.

Pin Name: TxN1
Pin Number: 19
Channel 1 high-speed differential transmit negative side.

Pin Name: TxP1
Pin Number: 20
Channel 1 high-speed differential transmit positive side.

Pin Name: RxN2
Pin Number: 6
Channel 2 high-speed differential receive negative side.

Pin Name: RxP2
Pin Number: 5
Channel 2 high-speed differential receive positive side. Loading an internal register through the flash or EEPROM during the initialization sequence could reverse RxP2 and RxN2 pinouts.

Pin Name: TxN2
Pin Number: 10
Channel 2 high-speed differential transmit negative side.

Pin Name: TxP2
Pin Number: 11
Channel 2 high-speed differential transmit positive side.

Other SerDes Signals

Pin Name: XTALO
Pin Number: 27
Crystal oscillator pin for SerDes reference clock. A 25-MHz crystal must be used.

Pin Name: XTALI/CLKI
Pin Number: 26
Crystal oscillator pin for SerDes reference clock. When external clock source is selected, the external clock (either 25 MHz or 100 MHz) will come in through this pin. The clock precision requirement is ±100ppm.

Pin Name: REXT
Pin Number: 24
External reference resistor pin for termination calibration. This pin provides the addition function of selecting frequency of the clock source. For 25-MHz crystal/external clock, a 1k, 1% resistor is connected to ground. To use 100-MHz external clock, a 4.99K, 1% resistor is connected to ground.

Package Drawing

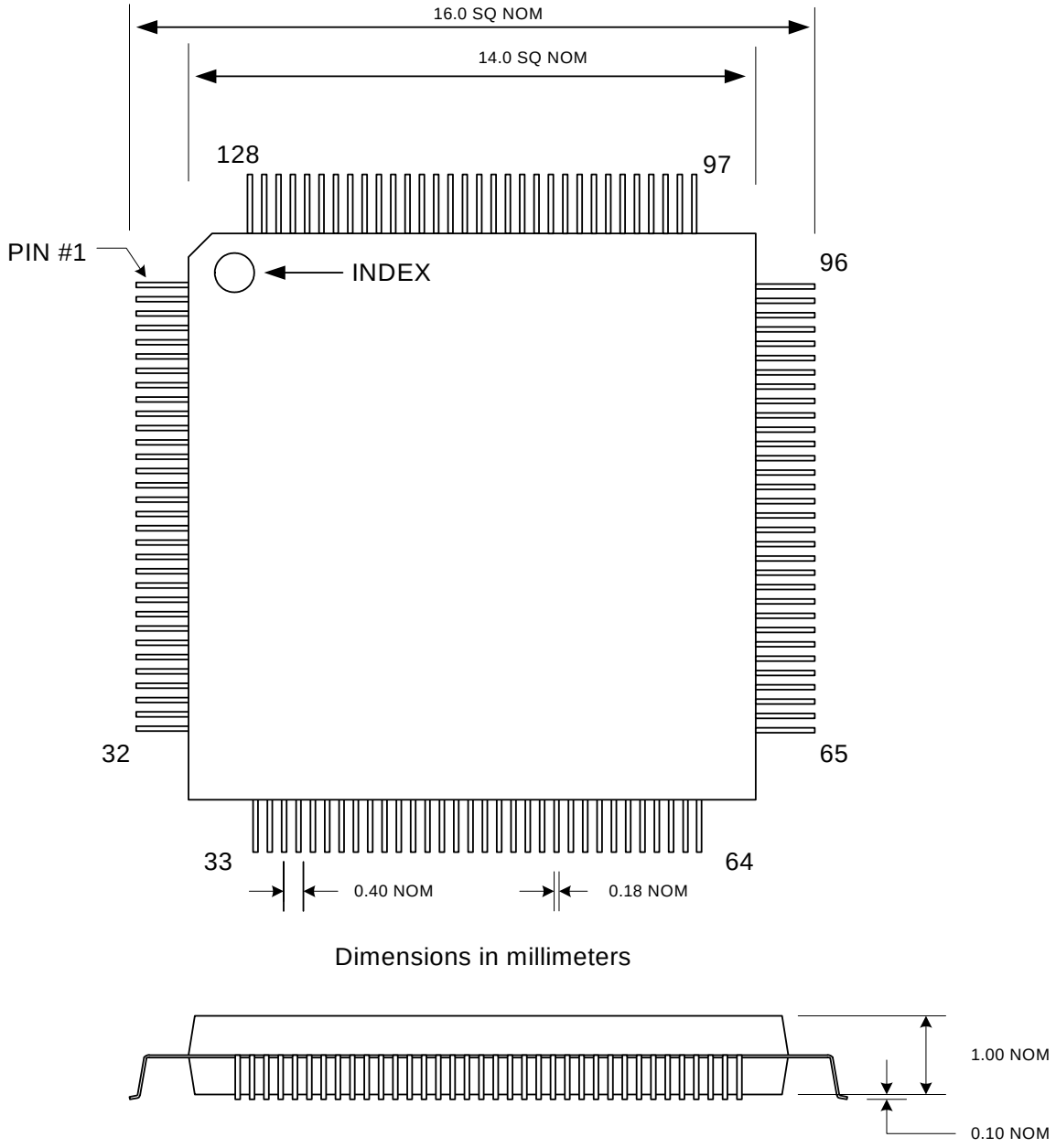


Figure 5. Package Drawing – 128 TQFP

Part Ordering Number:
SiI3512ECTU128 (128-pin TQFP lead-free package with an exposed pad)

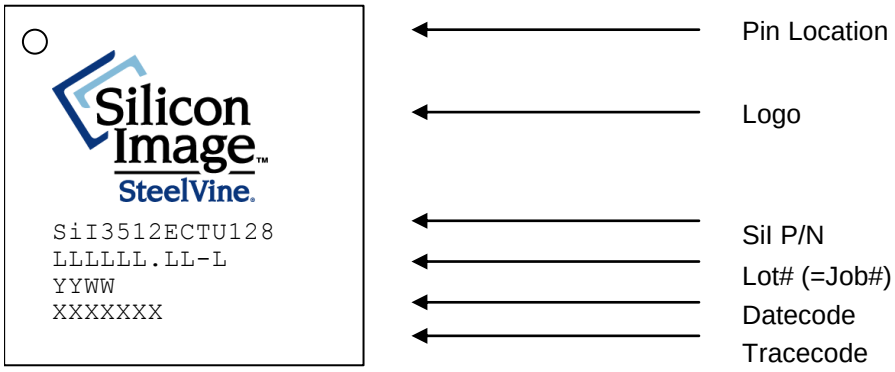


Figure 6. Package Markings

Enhanced Packaging (SiI3512ECTU128)

The SiI3512ECTU128 is packaged in a 128-pin TQFP with an exposed metal pad (5.385x5.385 mm) on the package designed to improve signal quality by having a low-inductance ground connection. The exposed pad should be soldered to a landing area on the PCB, as described below. The characterization results show improved signal quality by going above 25% coverage of the exposed pad area. A poor connection of the exposed pad to the landing area on the PCB may result in CRC or primitive error if the environments are noisy.

PCB Design Requirements

In order to improve the signal quality, it is required that landing area be incorporated on the PCB within the footprint of the package corresponding to the exposed metal pad on the package, as shown in Figure 8. Although the size of this landing area can be larger than the exposed pad on the package, the solderable area, as defined by the solder mask, should be at least the same as the exposed pad area on the package. A clearance of at least 0.5 mm should be designed on the PCB between the outer edges of the landing area and the inner edges of pad pattern for the leads to avoid any shorts.

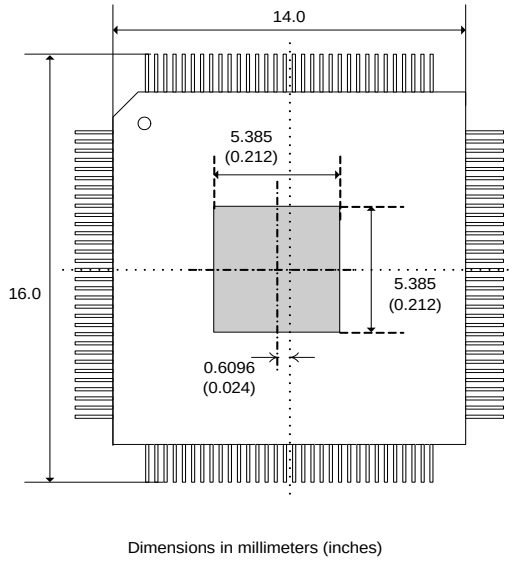


Figure 7. Top View of Enhanced 128-pin TQFP Package

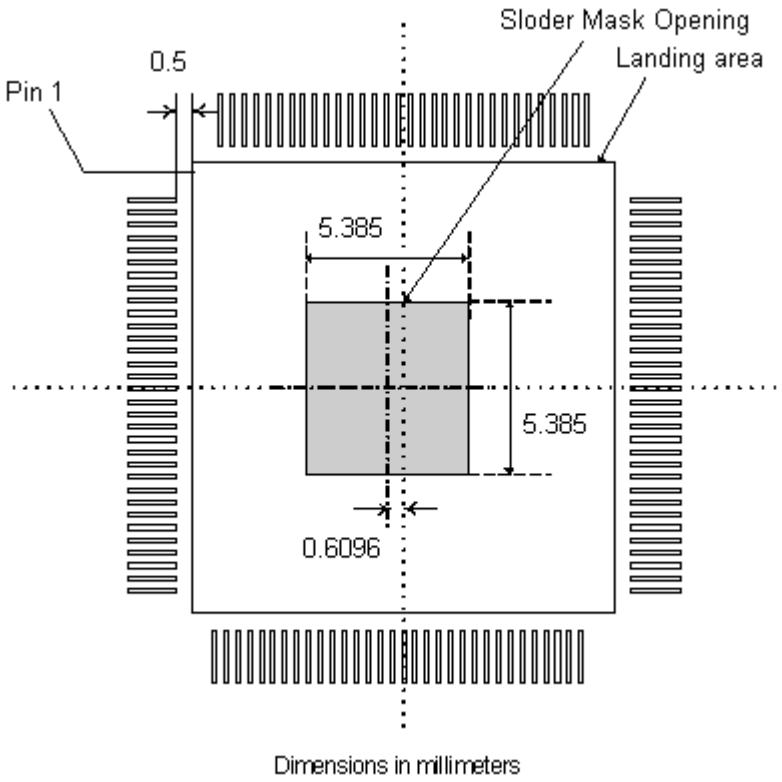
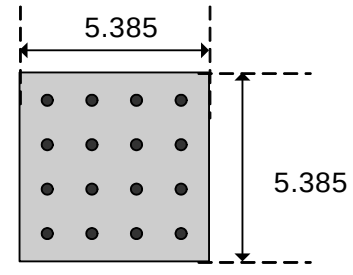


Figure 8. Top View of TQFP Landing Area Design on PCB

Ground vias are required in the metal land to provide a low-impedance connection to ground. An array of vias should be incorporated in the ground pad at a 1.2-mm pitch grid, as shown in Figure 9. The via diameter should be approximately 0.30 mm (12 mils). It is also desirable to avoid any solder wicking inside the via during the soldering process, which may result in voids in solder between the exposed pad and the PCB landing area. The ground vias can be “tented” with solder mask on the top surface of the PCB to avoid solder wicking inside the via during assembly. The solder mask diameter should be at least 0.1 mm (4 mils) larger than the via diameter.



0.3-mm diameter with 1.2-mm pitch Ground Via

Dimensions in millimeters

Figure 9. Ground Pad Via Grid

Board Mounting Guidelines

The following are general recommendations for mounting exposed pad lead-frame devices on the PCB. This should serve as the starting point in assembly process development and it is recommended that the process should be developed based on past experience in mounting standard, non-ePad packages.

Stencil Design

For proper ground connection, it is required that the exposed pad on the package be soldered to the landing area on the PCB. This requires solder paste application not only on the pad pattern for lead attachment but also on the landing area using the stencil. While for standard (non-ePad) lead-frame based packages the stencil thickness depends on the lead pitch and package co-planarity only, the package standoff also needs to be considered for the ground enhanced packages to determine the stencil thickness. For a nominal standoff of 0.1 mm, the stencil thickness of 5 to 8 mils (depending upon the pitch) should still provide good solder joint between the exposed pad and the landing area. The aperture openings should be the same as the solder mask opening on the ground land. Because a large stencil opening may result in poor release, the aperture opening can be subdivided into an array of smaller openings, similar to the ground land pattern shown in Figure 10. The above guidelines will result in the solder joint area to be about 80 to 90% of the exposed pad area.

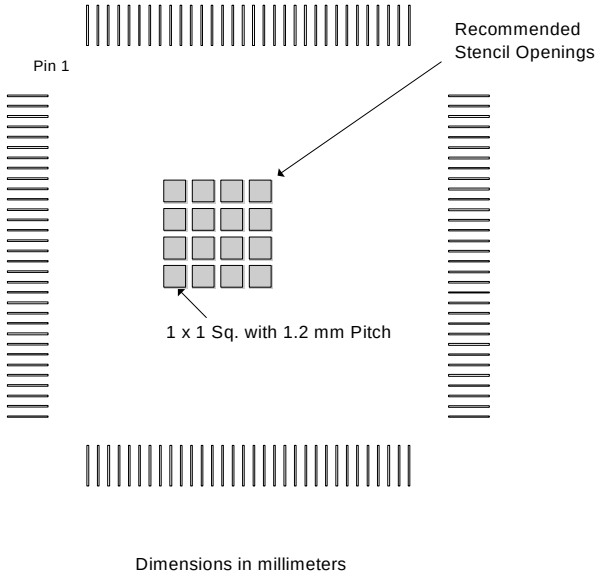


Figure 10. Top View of Recommended Stencil Design

Block Diagram

The SiI3512 controller contains the major logic modules shown below.

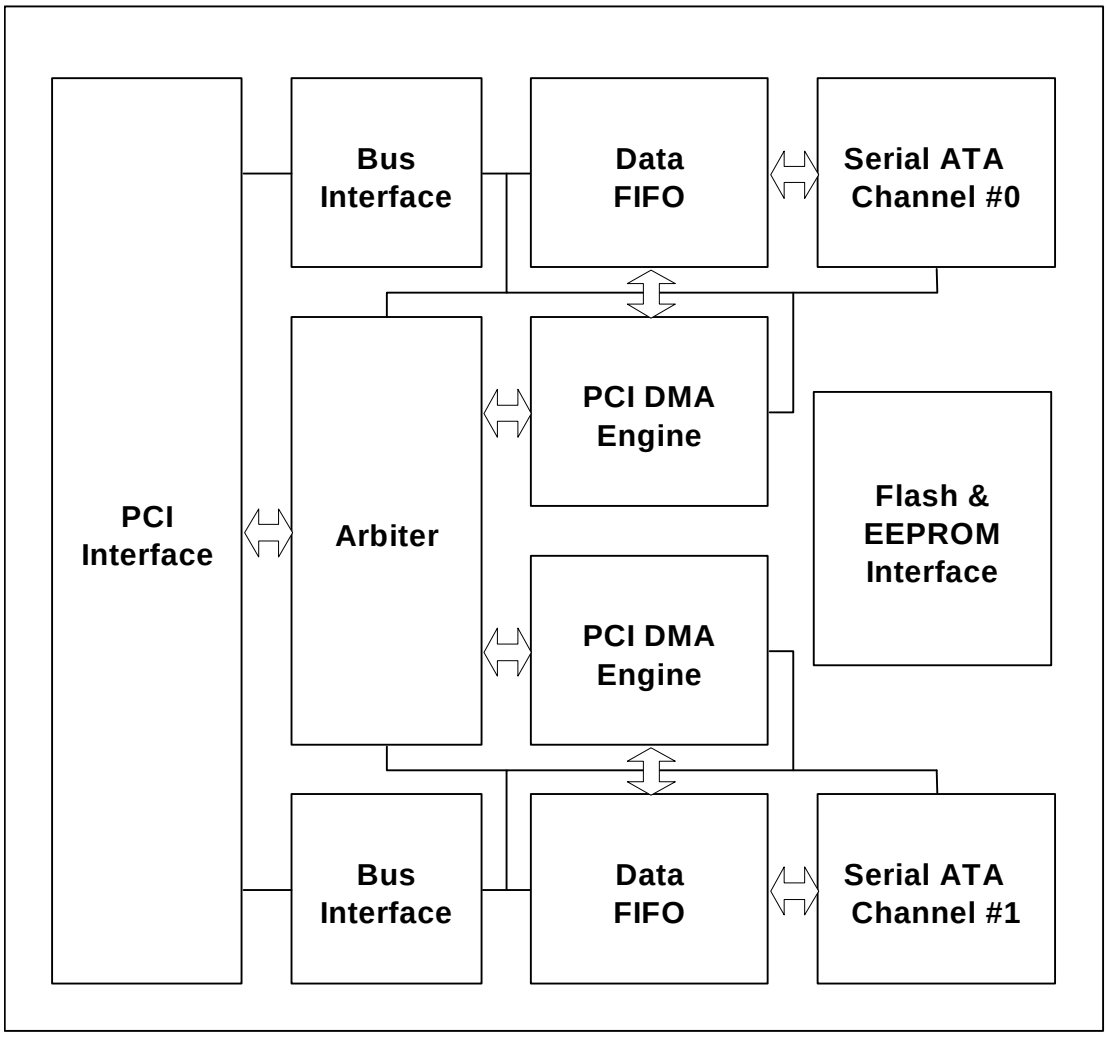


Figure 11. SiI3512 Block Diagram

Auto-Initialization

The SiI3512 controller supports an external flash and/or EEPROM device for BIOS extensions and user-defined PCI configuration header data.

Auto-Initialization from Flash

The SiI3512 controller initiates the flash detection and configuration space loading sequence upon the release of PCI_RST_N. It begins by reading the highest two addresses (0x7FFFF and 0x7FFFE), checking for the correct data signature pattern — 0xAA and 0x55, respectively. If the data signature pattern is correct, the SiI3512 device continues to sequence the address downward, reading a total of sixteen bytes. If the Data Signature is correct (0x55 at 0x7FFFC), the last 12 bytes are loaded into the PCI Configuration Space registers.

Note: If both flash and EEPROM are installed, the PCI Configuration Space registers will be loaded with the EEPROM's data. While the sequence is active, the SiI3512 controller responds to all PCI bus accesses with a Target Retry.

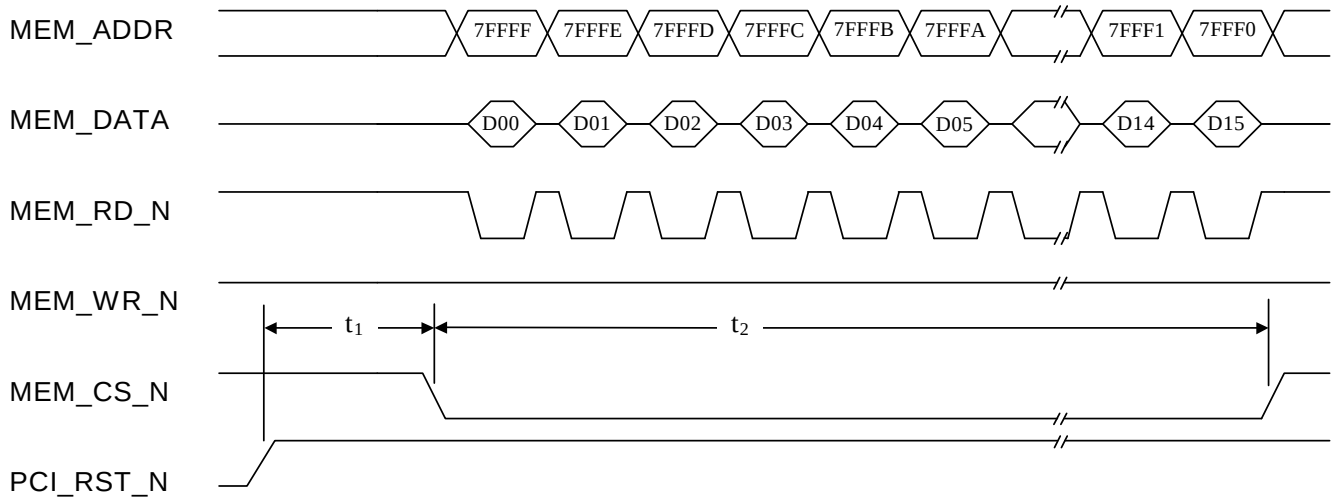


Figure 12. Auto-Initialization from Flash Timing

Table 11. Auto-Initialization from Flash Timing

Parameter	Value	Description
t_1	660 ns	PCI reset to Flash Auto-Initialization cycle begin
t_2	9600 ns	Flash Auto-Initialization cycle time

Table 12. Flash Data Description

Address	Data Byte	Description
0x7FFFF	D00	Data Signature = 0xAA
0x7FFFE	D01	Data Signature = 0x55
0x7FFFD	D02	AA = 120-ns flash device / else 240-ns flash device
0x7FFFC	D03	Data Signature = 0x55
0x7FFFB	D04	PCI Device ID [23:16]
0x7FFFA	D05	PCI Device ID [31:24]
0x7FFF9	D06	PCI Class Code [15:08]
0x7FFF8	D07	PCI Class Code [23:16]
0x7FFF7	D08	PCI Sub-System Vendor ID [07:00]
0x7FFF6	D09	PCI Sub-System Vendor ID [15:08]
0x7FFF5	D10	PCI Sub-System ID [23:16]
0x7FFF4	D11	PCI Sub-System ID [31:24]
0x7FFF3	D12	SerialATA PHY Config [07:00]
0x7FFF2	D13	SerialATA PHY Config [15:08]
0x7FFF1	D14	SerialATA PHY Config [23:16]
0x7FFF0	D15	SerialATA PHY Config [31:24]

Auto-Initialization from EEPROM

The SiI3512 controller initiates the EEPROM detection and configuration space loading sequence after the flash read sequence. The SiI3512 device supports up to 256-byte EEPROM with a 2-wire serial interface. The sequence of operations is as follows:

1. START condition defined as a high-to-low transition on SDAT while SCLK is high
2. Control byte = 1010 (Control Code) + 000 (Chip Select) + 0 (Write Address)
3. Acknowledge
4. Starting address field = 00000000
5. Acknowledge
6. Sequential data bytes separated by Acknowledges
7. STOP condition.

While the sequence is active, the SiI3512 controller responds to all PCI bus accesses with a Target Retry.

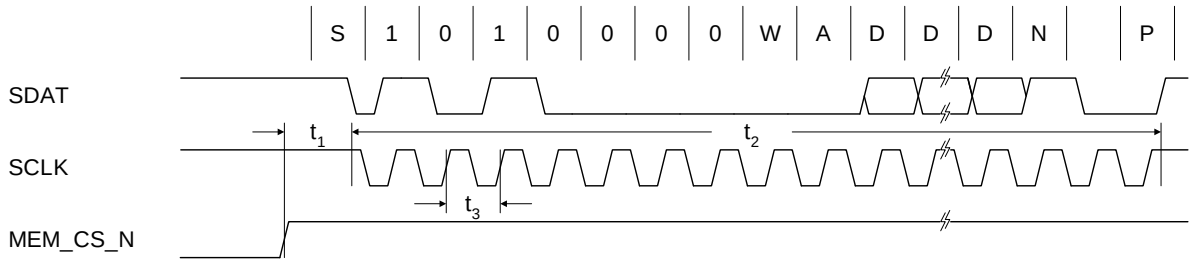


Figure 13. Auto-Initialization from EEPROM Timing

Table 13. Auto-Initialization from EEPROM Timing

Parameter	Value	Description
t_1	26.00 µs	End of Auto-Initialization from flash to start of Auto-Initialization from EEPROM
t_2	2.66 ms	Auto-Initialization from EEPROM cycle time
t_3	19.26 µs	EEPROM serial clock period

Table 14. Auto-Initialization from EEPROM Timing Symbols

Parameter	Description
S	START condition
W	R/W 0 = Write Command, 1 = Read Command
A	Acknowledge
D	Serial data
N	No-Acknowledge
P	STOP condition

Table 15. EEPROM Data Description

Address	Data Byte	Description
0x00	D00	Memory Present Pattern = 0xAA
0x01	D01	Memory Present Pattern = 0x55
0x02	D02	Data Signature = 0xAA
0x03	D03	Data Signature = 0x55
0x04	D04	PCI Device ID [23:16]
0x05	D05	PCI Device ID [31:24]
0x06	D06	PCI Class Code [15:08]
0x07	D07	PCI Class Code [23:16]
0x08	D08	PCI Sub-System Vendor ID [07:00]
0x09	D09	PCI Sub-System Vendor ID [15:08]
0x0A	D10	PCI Sub-System ID [23:16]
0x0B	D11	PCI Sub-System ID [31:24]
0x0C	D12	SerialATA PHY Config [07:00]
0x0D	D13	SerialATA PHY Config [15:08]
0x0E	D14	SerialATA PHY Config [23:16]
0x0F	D15	SerialATA PHY Config [31:24]

Register Definitions

This section describes the registers within the SiI3512 controller.

PCI Configuration Space

The PCI Configuration Space registers define the operation of the SiI3512 device on the PCI bus. These registers are accessible only when the SiI3512 controller detects a Configuration Read or Write operation, with its IDSEL asserted, on the 32-bit PCI bus.

Table 16 outlines the PCI Configuration space for the SiI3512 controller.

Table 16. PCI Configuration Space

Address Offset	Register Name				Access Type
	31 16		15 00		
0x00	Device ID		Vendor ID		R/W
0x04	PCI Status		PCI Command		R/W
0x08	PCI Class Code			Revision ID	R/W
0x0C	BIST	Header Type	Latency Timer	Cache Line Size	R/W
0x10	Base Address Register 0				R/W
0x14	Base Address Register 1				R/W
0x18	Base Address Register 2				R/W
0x1C	Base Address Register 3				R/W
0x20	Base Address Register 4				R/W
0x24	Base Address Register 5				R/W
0x28	Reserved				-
0x2C	Subsystem ID		Subsystem Vendor ID		R/W
0x30	Expansion ROM Base Address				R/W
0x34	Reserved			Capabilities Ptr	R
0x38	Reserved				R/W
0x3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line	R/W
0x40	Reserved			Configuration	R/W
0x44	Software Data Register				R/W
0x48	Reserved				-
0x4C	Reserved				-
0x50	Reserved				-
0x54	Reserved				-
0x58	Reserved				-
0x5C	Reserved				-
0x60	Power Management Capabilities		Next Item Pointer	Capability ID	R/W
0x64	Data	Reserved	Functions Control and Status		R/W
0x68	Reserved				-
0x6C	Reserved				-
0x70	Reserved	PCI Bus Master Status – IDE0	Reserved	PCI Bus Master Command – IDE0	R/W
0x74	PRD Table Address – IDE0				R/W
0x78	Reserved	PCI Bus Master Status – IDE1	Reserved	PCI Bus Master Command – IDE1	R/W
0x7C	PRD Table Address – IDE1				R/W
0x80	Reserved			IDE0 Data Transfer Mode	R/W
0x84	Reserved			IDE1 Data Transfer Mode	R/W

Address Offset	Register Name				Access Type
	31	16	15	00	
0x88	System Configuration Status		System Command		R/W
0x8C	System Software Data				R/W
0x90	Flash Memory Address – Command + Status				R/W
0x94	Reserved			Flash Memory Data	R/W
0x98	EEPROM Memory Address – Command + Status				R/W
0x9C	Reserved			EEPROM Memory Data	R/W
0xA0	Reserved		IDE0 Config + Status	IDE0 Cmd + Status	R/W
0xA4	Reserved				R/W
0xA8	Reserved				R/W
0xAC	Reserved				R/W
0xB0	Reserved		IDE1 Config + Status	IDE1 Cmd + Status	R/W
0xB4	Reserved				R/W
0xB8	Reserved				R/W
0xBC	Reserved				R/W
0xC0	BA5 Indirect Address				R/W
0xC4	BA5 Indirect Access				R/W

Device ID – Vendor ID

Address Offset: 0x00
Access Type: Read /Write
Reset Value: 0x3512_1095

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Device ID																Vendor ID															

This register defines the Device ID and Vendor ID associated with the SiI3512 controller. The register bits are defined below.

- **Bit [31:16]**: Device ID (R/W) – Device ID. This value in this bit field is determined by any one of three options:
    1. This field defaults to 0x3512 to identify the device as a Silicon Image SiI3512 controller.
    2. Loaded from an external memory device: If an external memory device – flash or EEPROM – is present with the correct signature, the Device ID is loaded from that device after reset. See Auto-Initialization section on page 25 for additional information.
    3. System programmable: If Bit 0 of the Configuration register (0x40) is set, the three are system programmable.
- **Bit [15:00]**: Vendor ID (R) – Vendor ID. This field defaults to 0x1095 to identify the vendor as Silicon Image.

PCI Status – PCI Command

Address Offset: 0x04
Access Type: Read/Write/Write-One-to-Clear
Reset Value: 0x02B0_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Det Par Err	Sig Sys Err	Rcvd M Abort	Rcvd T Abort	Sig T Abort	Devsel Timing		Det M Data Par Err	Fast B-to-B Capable	Reserved	66 -MHz Capable	Capabilities List	Int Status	Reserved								Int Disable	Fast B-to-B Enable	SERR Enable	Address Stepping	Par Error Response	VGA Palette	Memory Wr & Inv	Special Cycles	Bus Master	Memory Space	IO Space

This register defines the various control functions associated with the PCI bus. The register bits are defined below.

- **Bit 31**: Det. Par Err (R/W1C) – Detected Parity Error. This bit set indicates that the SiI3512 device detected a parity error on the PCI bus-address or data parity error-while responding as a PCI target.
- **Bit 30**: Sig. Sys Err (R/W1C) – Signaled System Error. This bit set indicates that the SiI3512 controller signaled SERR on the PCI bus.
- **Bit 29**: Rcvd M Abort (R/W1C) – Received Master Abort. This bit set indicates that the SiI3512 controller terminated a PCI bus operation with a Master Abort.
- **Bit 28**: Rcvd T Abort (R/W1C) – Received Target Abort. This bit set indicates that the SiI3512 controller received a Target Abort termination.
- **Bit 27**: Sig. T Abort (R/W1C) – Signaled Target Abort. This bit set indicates that the SiI3512 controller terminated a PCI bus operation with a Target Abort.
- **Bit [26:25]**: Devsel Timing (R) – Device Select Timing. This bit field indicates the DEVSEL timing supported by the SiI3512 device. The hardwired value is 0b01 for Medium decode timing.
- **Bit 24**: Det M Data Par Err (R/W1C) – Detected Master Data Parity Error. This bit set indicates that the SiI3512 controller, as bus master, detected a parity error on the PCI bus. The parity error may be either reported by the target device via PERR# on a write operation or by the SiI3512 controller on a read operation.
- **Bit 23**: Fast B-to-B Capable (R) – Fast Back-to-Back Capable. This bit is hardwired to 1 to indicate that the SiI3512 controller is Fast Back-to-Back capable as a PCI target.
- **Bit 22**: Reserved (R).
- **Bit 21**: 66-MHz Capable (R) – 66-MHz PCI Operation Capable. This bit is hardwired to 1 to indicate that the SiI3512 device is 66-MHz capable.
- **Bit 20**: Capabilities List (R) – PCI Capabilities List. This bit is hardwired to 1 to indicate that the SiI3512 controller has a PCI Power Management Capabilities register linked at offset 0x34.
- **Bit 19**: Interrupt Status (R).
- **Bit [18:11]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit 10**: Interrupt Disable (R/W)
- **Bit 09**: Fast B-to-B Enable (R) – Fast Back-to-Back Enable. This bit is hardwired to 0 to indicate that the SiI3512 controller does not support Fast Back-to-Back operations as bus master.
- **Bit 08**: SERR Enable (R/W) – SERR Output Enable. This bit set enables the SiI3512 controller to drive the PCI SERR# pin when it detects an address parity error. The Parity Error Response bit (06) must also be set to enable SERR# reporting.
- **Bit 07**: Address Stepping (R) – Address Stepping Enable. This bit is hardwired to 0 to indicate that the SiI3512 controller does not support Address Stepping.
- **Bit 06**: Par Error Response (R/W) – Parity Error Response Enable. This bit set enables the SiI3512 controller to respond to parity errors on the PCI bus. If this bit is cleared, the SiI3512 controller will ignore PCI parity errors.
- **Bit 05**: VGA Palette (R) – VGA Palette Snoop Enable. This bit is hardwired to 0 to indicate that the SiI3512 controller does not support VGA Palette Snooping.
- **Bit 04**: Mem Wr & Inv (R) – Memory Write and Invalidate Enable. This bit is hardwired to 0 to indicate that the SiI3512 device does not support Memory Write and Invalidate.
- **Bit 03**: Special Cycles (R) – Special Cycles Enable. This bit is hardwired to 0 to indicate that the SiI3512 controller does not respond to Special Cycles.
- **Bit 02**: Bus Master (R/W) – Bus Master Enable. This bit set enables the SiI3512 device to act as PCI bus master.

- **Bit 01**: Memory Space (R/W) – Memory Space Enable. This bit set enables the SiI3512 device to respond to PCI memory space access.
- **Bit 00**: IO Space (R/W) – IO Space Enable. This bit set enables the SiI3512 controller to respond to PCI IO space access.

PCI Class Code – Revision ID

Address Offset: 0x08
Access Type: Read/Write
Reset Value: 0x0180_0001 or 0x0104_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PCI Class Code																PCI Prog Int				IDE1 Mode Prog	IDE1 Pwr-Up Mode	IDE0 Mode Prog	IDE0 Pwr-Up Mode	Revision ID							

This register defines the various control functions associated with the PCI bus. The register bits are defined below.

- **Bit [31:08]**: PCI Class Code (R) – PCI Class Code. This value in this bit field is determined by any one of three options:
  1) The default value, set by an external jumper on the FL_ADDR[00]/IDE_CFG pin:
     1. If IDE_CFG = 0, the value is 0x010400 for RAID mode
     2. If IDE_CFG = 1, the value is 0x018000 for Mass Storage class
  2) Loaded from an external memory device: If an external memory device — flash or EEPROM — is present with the correct signature, the PCI Class Code is loaded from that device after reset. See Auto-Initialization section on page 25 for additional information.
  3) System programmable: If Bit 0 of the Configuration register (0x40) is set the three bytes are system programmable.
- **Bit [07:00]**: Revision ID (R) – Chip Revision ID. This bit field is hardwired to indicate the revision level of the chip design; revision 0x01 is defined for the production version.

BIST – Header Type – Latency Timer – Cache Line Size

Address Offset: 0x0C
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BIST								Header Type								Latency Timer								Cache Line Size							

This register defines the various control functions associated with the PCI bus. The register bits are defined below.

- **Bit [31:24]**: BIST (R) – This bit field is hardwired to 0x00.
- **Bit [23:16]**: Header Type (R) – This bit field is hardwired to 0x00.
- **Bit [15:08]**: Latency Timer (R/W). This bit field is used to specify the time in number of PCI clocks, the SiI3512 device as a master is still allowed to control the PCI bus after its GRANT_L is de-asserted. The lower four bits [0B:08] are hardwired to 0x0, resulting in a time granularity of 16 clocks.
- **Bit [07:00]**: Cache Line Size (R/W). This bit field is used to specify the system cache line size in terms of 32-bit words. The upper 2 bits are not used, resulting in a maximum size of 64 32-bit words. With the SiI3512 controller as a master, initiating a read transaction, it issues PCI command Read Multiple in place when empty space in its FIFO is larger than the value programmed in this register. If this value is set to 0x00, SiI3512 controller will disable PCI command Read Multiple.

Base Address Register 0

Address Offset: 0x10
Access Type: Read/Write
Reset Value: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 0																													Not Used		

This register defines the addressing of various control functions within the SiI3512 controller. The register bits are defined below.

- **Bit [31:03]**: Base Address Register 0 (R/W). This register defines the I/O Space base address for the IDE Channel #0 task file registers.
- **Bit [02:00]**: Base Address Register 0 (R) – This bit field is not used and is hardwired to 0b001.

Base Address Register 1

Address Offset: 0x14
Access Type: Read/Write
Reset Value: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 1																														Not Used	

This register defines the addressing of various control functions within the SiI3512 controller. The register bits are defined below.

- **Bit [31:02]**: Base Address Register 1 (R/W). This register defines the I/O Space base address for the IDE Channel #0 Device Control- Alternate Status register.
- **Bit [01:00]**: Base Address Register 1 (R) – This bit field is not used and is hardwired to 0b01.

Base Address Register 2

Address Offset: 0x18
Access Type: Read/Write
Reset Value: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 2																													Not Used		

This register defines the addressing of various control functions within the SiI3512 controller. The register bits are defined below.

- **Bit [31:03]**: Base Address Register 2 (R/W). This register defines the I/O Space base address for the IDE Channel #1 task file registers.
- **Bit [02:00]**: Base Address Register 2 (R) – This bit field is not used and is hardwired to 0b001.

Base Address Register 3

Address Offset: 0x1C
Access Type: Read/Write
Reset Value: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 3																														Not Used	

This register defines the addressing of various control functions within the SiI3512 controller. The register bits are defined below.

- **Bit [31:02]**: Base Address Register 3 (R/W). This register defines the I/O Space base address for the IDE Channel #1 Device Control- Alternate Status register.
- **Bit [01:00]**: Base Address Register 3 (R). This bit field is not used and is hardwired to 0b01.

Base Address Register 4

Address Offset: 0x20
Access Type: Read/Write
Reset Value: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 4																												Not Used			

This register defines the addressing of various control functions within the SiI3512 controller. The register bits are defined below.

- **Bit [31:04]**: Base Address Register 4 (R/W). This register defines the I/O Space base address for the PCI bus master registers.
- **Bit [03:00]**: Base Address Register 4 (R). This bit field is not used and is hardwired to 0b0001.

Base Address Register 5

Address Offset: 0x24
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Base Address Register 5																															

This register defines the addressing of various control functions within the SiI3512 device. This register is enabled when input BA5_EN is set to one (see description for pin FL_ADDR[01]/BA5_EN in Miscellaneous I/O Pins section on page 15. The register bits are defined below.

- **Bit [31:09]**: Base Address Register 5 (R/W). This register defines the Memory Space base address for all Silicon Image driver specific functions.
- **Bit [08:00]**: Base Address Register 5 (R). This bit field is not used and is hardwired to 0x00.

Subsystem ID – Subsystem Vendor ID

Address Offset: 0x2C
Access Type: Read/Write
Reset Value: 0x3512_1095

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Subsystem ID																Subsystem Vendor ID															

This register defines the Subsystem ID fields associated with the PCI bus. The register bits are defined below.

- **Bit [31:16]**: Subsystem ID (R). The value in this bit field is determined by any one of three options:
  1) The default value of 0x3512.
  2) Loaded from an external memory device: If an external memory device – flash or EEPROM – is present with the correct signature, the Subsystem ID is loaded from that device after reset. See Auto-Initialization section on page 25 for additional information.
  3) System programmable: If Bit 0 of the Configuration register (0x40) is set, the two bytes are system programmable.
- **Bit [15:00]**: Subsystem Vendor ID (R). The value in this bit field is determined by any one of three options:
  1) The default value of 0x1095.
  2) Loaded from an external memory device: If an external memory device – flash or EEPROM – is present with the correct signature, the Subsystem Vendor ID is loaded from that device after reset. See Auto-Initialization section on page 25 for additional information.
  3) System programmable: If Bit 0 of the Configuration register (0x40) is set, the two bytes are system programmable.

Expansion ROM Base Address

Address Offset: 0x30
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Expansion ROM Base Address													Not Used																		Exp ROM Enable

This register defines the Expansion ROM base address associated with the PCI bus. The register bits are defined below.

- **Bit [31:19]**: Expansion ROM Base Address (R/W) – Expansion ROM Base Address. This bit field defines the upper bits of the Expansion ROM base address.
- **Bit [18:01]**: Not Used (R). This bit field is hardwired to 0x00000. The minimum Expansion ROM address range is 512K bytes.
- **Bit [00]**: Exp ROM Enable (R/W) – Expansion ROM Enable. This bit is set to enable the Expansion ROM access.

Capabilities Pointer

Address Offset: 0x34
Access Type: Read
Reset Value: 0x0000_0060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Capabilities Pointer							

This register defines the link to a list of new capabilities associated with the PCI bus. The register bits are defined below.

- **Bit [31:08]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [07:00]**: Capabilities Pointer (R) – Capabilities Pointer. This bit field defaults to 0x60 to define the address for the 1st entry in a list of PCI Power Management capabilities.

Max Latency – Min Grant – Interrupt Pin – Interrupt Line

Address Offset: 0x3C
Access Type: Read/Write
Reset Value: 0x0000_0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Max Latency								Min Grant								Interrupt Pin								Interrupt Line							

This register defines the various control functions associated with the PCI bus. The register bits are defined below.

- **Bit [31:24]**: Max Latency (R) – Maximum Latency. This bit field is hardwired to 0x00.
- **Bit [23:16]**: Min Grant (R) – Minimum Grant. This bit field is hardwired to 0x00.
- **Bit [15:08]**: Interrupt Pin (R) – Interrupt Pin Used. This bit field is hardwired to 0x01 to indicate that the SiI3512 controller uses the INTA# interrupt.
- **Bit [07:00]**: Interrupt Line (R/W) – Interrupt Line. This bit field is used by the system to indicate interrupt line routing information. The SiI3512 device does not use this information.

Configuration

Address Offset: 0x40
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																														BA5 Ind Acc Ena	PCI Hdr Wr Ena

This register defines the various control functions associated with the PCI bus. The register bits are defined below.

- **Bit [31:02]**: Reserved (R). This bit field is hardwired to 0x00000000.
- **Bit [01]**: BA5 Ind Acc Ena (R/W) – BA5 Indirect Access Enable. This bit is set to enable indirect access to BA5 address space using Configuration Space registers 0xC0 and 0xC4 (BA5 Indirect Address and BA5 Indirect Access).
- **Bit [00]**: PCI Hdr Wr Ena (R/W) – PCI Configuration Header Write Enable. This bit is set to enable write access to the following registers in the PCI Configuration Header: Device ID (0x03–0x02), PCI Class Code (0x09–0x0B), Subsystem Vendor ID (0x2D–0x2C), and Subsystem ID (0x2F–0x2E).

Software Data Register

Address Offset: 0x44
Access Type: Read/Write
Reset Value: Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Software Data																															

This register is used by the software for non-resettable data storage. The contents are unknown on power-up and are never cleared by any type of reset.

Power Management Capabilities

Address Offset: 0x60
Access Type: Read Only
Reset Value: 0x0622_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PME Support					PPM D2 Support	PPM D1 Support	Auxiliary Current			Dev Special Init	Reserved	PME Clock	PPM Rev			Next Item Pointer								Capability ID							

This register defines the power management capabilities associated with the PCI bus. The register bits are defined below.

- **Bit [31:27]**: PME Support (R) – Power Management Event Support. This bit field is hardwired to 0x00 to indicate that the SiI3512 device does not support PME.
- **Bit [26]**: PPM D2 Support (R) – PCI Power Management D2 Support. This bit is hardwired to 1 to indicate support for the D2 Power Management State.
- **Bit [25]**: PPM D1 Support (R) – PCI Power Management D1 Support. This bit is hardwired to 1 to indicate support for the D1 Power Management State.
- **Bit [24:22]**: Auxiliary Current (R) – Auxiliary Current. This bit field is hardwired to 0b000.
- **Bit [21]**: Dev Special Init (R) – Device Special Initialization. This bit is hardwired to 1 to indicate that the SiI3512 controller requires special initialization
- **Bit [20]**: Reserved (R). This bit is reserved and returns zero on a read.
- **Bit [19]**: PME Clock (R) – Power Management Event Clock. This bit is hardwired to 0. The SiI3512 controller does not support PME.
- **Bit [18:16]**: PPM Rev (R) – PCI Power Management Revision. This bit field is hardwired to 0b010 to indicate compliance with the *PCI Power Management Interface Specification revision 1.1*.
- **Bit [15:08]**: Next Item Pointer (R) – PCI Additional Capability Next Item Pointer. This bit field is hardwired to 0x00 to indicate that there are no additional items on the Capabilities List.
- **Bit [07:00]**: Capability ID (R) – PCI Additional Capability ID. This bit field is hardwired to 0x01 to indicate that this Capabilities List is a PCI Power Management definition.

Power Management Control + Status

Address Offset: 0x64
Access Type: Read/Write
Reset Value: 0x6400_4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PPM Data								Reserved								PME Status	PPM Data Scale		PPM Data Sel				PME Ena	Reserved						PPM Power State	

This register defines the power management capabilities associated with the PCI bus. The register bits are defined below.

- **Bit [31:24]**: PPM Data (R) – PCI Power Management Data. This bit field is hardwired to 0x64.
- **Bit [23:16]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [15]**: PME Status (R) – PME Status. This bit is hardwired to 0. The SiI3512 device does not support PME.
- **Bit [14:13]**: PPM Data Scale (R) – PCI Power Management Data Scale. This bit field is hardwired to 0b10 to indicate a scaling factor of 10 mW.
- **Bit [12:09]**: PPM Data Sel (R/W) – PCI Power Management Data Select. This bit field is set by the system to indicate which data field is to be reported through the PPM Data bits (although current implementation hardwires the PPM Data to indicate 1 Watt).
- **Bit [08]**: PME Ena (R) – PME Enable. This bit is hardwired to 0. The SiI3512 controller does not support PME.
- **Bit [07:02]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [01:00]**: PPM Power State (R/W) – PCI Power Management Power State. This bit field is set by the system to dictate the current Power State: 00 = D0 (Normal Operation), 01 = D1, 10 = D2, and 11 = D3 (Hot).

PCI Bus Master – IDE0

Address Offset: 0x70
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	Reserved		IDE0 DMA Comp	PBM Error	PBM Active	Reserved								Reserved				PBM Rd-Wr	Reserved		PBM Enable

This register defines the PCI bus master register for IDE Channel #0 in the SiI3512 controller. The register bits are also mapped to Base Address 4, Offset 0x00, Base Address 5, Offset 0x00, and Base Address 5, Offset 0x10. See PCI Bus Master – IDE0 section on page 51 for bit definitions.

PRD Table Address – IDE0

Address Offset: 0x74
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Table Address – IDE0																														Reserved	

This register defines the PRD Table Address register for IDE Channel #0 in the SiI3512 controller. The register bits are also mapped to Base Address 4, Offset 0x04 and Base Address 5, Offset 0x04. See PRD Table Address – IDE0 section on page 52 for bit definitions.

PCI Bus Master – IDE1

Address Offset: 0x78
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	Reserved		IDE1 DMA Comp	PBM Error	PBM Active	Reserved								Reserved					PBM Rd-Wr	Reserved		PBM Enable

This register defines the PCI bus master register for IDE Channel #1 in the SiI3512 device. The register bits are also mapped to Base Address 4, Offset 0x08, Base Address 5, Offset 0x08, and Base Address 5, Offset 0x18. See PCI Bus Master – IDE1 section on page 52 for bit definitions.

PRD Table Address – IDE1

Address Offset: 0x7C
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Table Address – IDE1																														Reserved	

This register defines the PRD Table Address register for IDE Channel #1 in the SiI3512 controller. The register bits are also mapped to Base Address 4, Offset 0x0C and Base Address 5, Offset 0x0C. See PRD Table Address – IDE1 section on page 53 for bit definitions.

Data Transfer Mode – IDE0

Address Offset: 0x80
Access Type: Read/Write
Reset Value: 0x0000_0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Reserved		IDE0 Device 1 Transfer Mode		Reserved		IDE0 Device 0 Transfer Mode	

This register defines the transfer mode register for IDE Channel #0 in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0xB4. See Data Transfer Mode – IDE0 section on page 67 for bit definitions.

Data Transfer Mode – IDE1

Address Offset: 0x84
Access Type: Read/Write
Reset Value: 0x0000_0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Reserved		IDE1 Device 1 Transfer Mode		Reserved		IDE1 Device 0 Transfer Mode	

This register defines the transfer mode register for IDE Channel #1 in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0xF4. See Data Transfer Mode – IDE1 section on page 72 for bit definitions.

System Configuration Status – Command

Address Offset: 0x88
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								IDE1 Int Block	IDE0 Int Block	Reserved					BA5_EN	Reserved								IDE0 Module Rst	IDE1 Module Rst	FF0 Module Rst	FF1 Module Rst	Reserved		ARB Module Rst	PBM Module Rst

This register defines the system configuration status and command register for the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x48. See System Configuration Status – Command section on page 58 for bit definitions.

System Software Data Register

Address Offset: 0x8C
Access Type: Read/Write
Reset Value: Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
System Software Data																															

This register is used by the software for non-resettable data storage. The contents are unknown on power-up and are never cleared by any type of reset. The register bits are also mapped to Base Address 5, Offset 0x4C. See System Software Data Register section on page 59 for bit definitions.

Flash Memory Address – Command + Status

Address Offset: 0x90
Access Type: Read/Write
Reset Value: 0x0800_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved				Mem Init Done	Mem Init	Mem Access Start	Mem Access Type	Reserved					Memory Address																		

This register defines the address and command/status register for flash memory interface in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x50. See Flash Memory Address – Command + Status section on page 59 for bit definitions.

Flash Memory Data

Address Offset: 0x94
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Memory Data							

This register defines the data register for flash memory interface in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x54. See Flash Memory Data section on page 60 for bit definitions.

EEPROM Memory Address – Command + Status

Address Offset: 0x98
Access Type: Read/Write
Reset Value: 0x0800_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved			Mem Error	Mem Init Done	Mem Init	Mem Access Start	Mem Access Type	Reserved																Mem Address							

This register defines the address and command/status register for EEPROM memory interface in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x58. See EEPROM Memory Address – Command + Status section on page 60 for bit definitions.

EEPROM Memory Data

Address Offset: 0x9C
Access Type: Read/Write
Reset Value: 0x0000_00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Memory Data							

This register defines the data register for EEPROM memory interface in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x5C. See EEPROM Memory Data section on page 61 for bit definitions.

IDE0 Task File Configuration + Status

Address Offset: 0xA0
Access Type: Read/Write
Reset Value: 0x6515_0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																Reserved	Watchdog Int Ena	Watchdog Ena	Watchdog Timeout	Interrupt Status	Virtual DMA Int	IORDY Monitoring	Reserved						Channel Rst	Buffered Cmd	Reserved

This register defines the task file configuration and status register for IDE Channel #0 in the SiI3512 device. The register bits are also mapped to Base Address 5, Offset 0xA0. See IDE0 Task File Configuration + Status section on page 66 for bit definitions.

IDE1 Task File Configuration + Status

Address Offset: 0xB0
Access Type: Read/Write
Reset Value: 0x6515_0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																Reserved	Watchdog Int Ena	Watchdog Ena	Watchdog Timeout	Interrupt Status	Virtual DMA Int	IORDY Monitoring	Reserved						Channel Rst	Buffered Cmd	Reserved

This register defines the task file configuration and status register for IDE Channel #1 in the SiI3512 device. The register bits are also mapped to Base Address 5, Offset 0xE0. See IDE1 Task File Configuration + Status section on page 71 for bit definitions.

BA5 Indirect Address

Address Offset: 0xC0
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																							Address								

This register permits the indirect addressing of registers normally referenced using Base Address 5. Any register that is not accessible by any means other than via Base Address 5 is indirectly addressable. The following BA5 address ranges are not indirectly accessible, but are accessible either in Configuration Space or via other Base Address registers: 0x00–0x1C, 0x80–0x8C, 0xC0–0xCC.

BA5 Indirect Access

Address Offset: 0xC4
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
As defined for indirectly accessed register																															

This register provides the indirect access addressed by the BA5 Indirect Address register. The use of indirect access must be enabled by setting bit 1 of the Configuration register (0x40).

Internal Register Space – Base Address 0

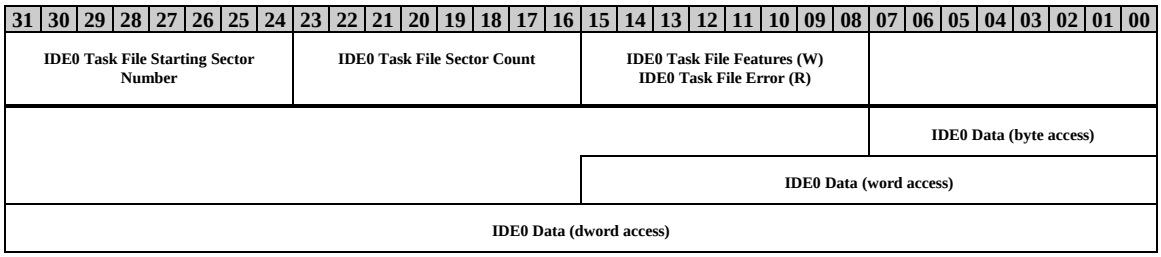
These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI I/O space.

Table 17. SiI3512 Internal Register Space – Base Address 0

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	IDE0 TF Starting Sector Number	IDE0 TF Sector Count	IDE0 TF Features IDE0 TF Error	IDE0 TF Data	R/W
0x04	IDE0 TF Command+Status	IDE0 TF Device+Head	IDE0 TF Cylinder High	IDE0 TF Cylinder Low	R/W

IDE0 Task File Register 0

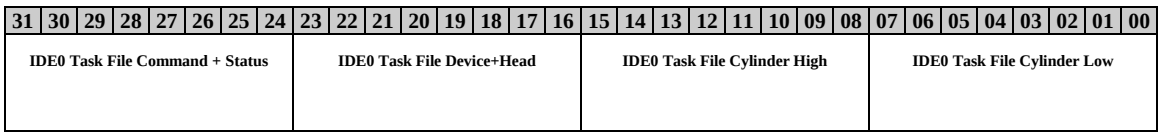
Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines four of the IDE Channel #0 Task File registers in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x80. See IDE0 Task File Register 0 section on page 63 for bit definitions.

IDE0 Task File Register 1

Address Offset: 0x04
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines four of the IDE Channel #0 Task File registers in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0x84. See IDE0 Task File Register 1 section on page 64 for bit definitions.

Internal Register Space – Base Address 1

These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI I/O space.

Table 18. SiI3512 Internal Register Space – Base Address 1

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	Reserved	IDE0 TF Device Control Auxiliary Status	Reserved	Reserved	R/W

IDE0 Task File Register 2

Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								IDE0 Task File Device Control IDE0 Task File Auxiliary Status								Reserved								Reserved							

This register defines one of the IDE Channel #0 Task File registers in the SiI3512 device. The register bits are also mapped to Base Address 5, Offset 0x88. See IDE0 Task File Register 2 section on page 64 for bit definitions.

Internal Register Space – Base Address 2

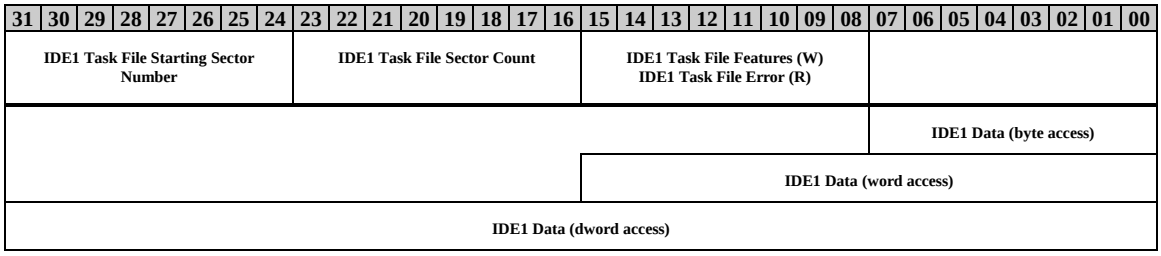
These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI I/O space.

Table 19. SiI3512 Internal Register Space – Base Address 2

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	IDE1 TF Starting Sector Number	IDE1 TF Sector Count	IDE1 TF Features IDE1 TF Error	IDE1 TF Data	R/W
0x04	IDE1 TF Command+Status	IDE1 TF Device+Head	IDE1 TF Cylinder High	IDE1 TF Cylinder Low	R/W

IDE1 Task File Register 0

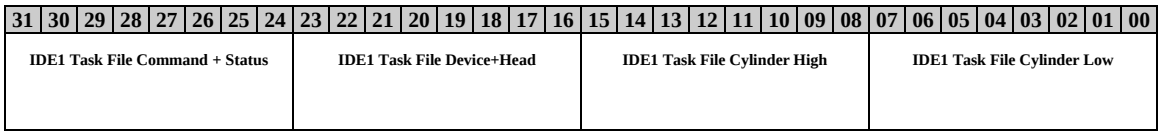
Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines four of the IDE Channel #1 Task File registers in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0xC0. See IDE1 Task File Register 0 section on page 68 for bit definitions.

IDE1 Task File Register 1

Address Offset: 0x04
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines four of the IDE Channel #1 Task File registers in the SiI3512 controller. The register bits are also mapped to Base Address 5, Offset 0xC4. See IDE1 Task File Register 1 section on page 68 for bit definitions.

Internal Register Space – Base Address 3

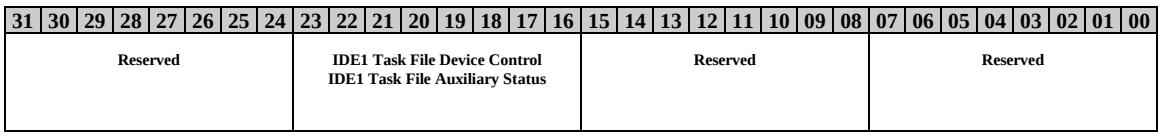
These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI I/O space.

Table 20. SiI3512 Internal Register Space – Base Address 3

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	Reserved	IDE1 TF Device Control Auxiliary Status	Reserved	Reserved	R/W

IDE1 Task File Register 2

Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines one of the IDE Channel #1 Task File registers in the SiI3512 device. The register bits are also mapped to Base Address 5, Offset 0xC8. See IDE1 Task File Register 2 section on page 69 for bit definitions.

Internal Register Space – Base Address 4

These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI I/O space.

Table 21. SiI3512 Internal Register Space – Base Address 4

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	Reserved	PCI Bus Master Status – IDE0	Software Data	PCI Bus Master Command – IDE0	R/W
0x04	PRD Table Address – IDE0				R/W
0x08	Reserved	PCI Bus Master Status – IDE1	Reserved	PCI Bus Master Command – IDE1	R/W
0x0C	PRD Table Address – IDE1				R/W

PCI Bus Master – IDE0

Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_XX00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	Reserved		IDE0 DMA Comp	PBM Error	PBM Active	IDE Watchdog	IDE1 DMA Comp	Software						Reserved				PBM Rd-Wr	Reserved		PBM Enable

This register defines the PCI bus master register for IDE Channel #0 in the SiI3512 controller. See PCI Bus Master – IDE0 section on page 51 for bit definitions.

PRD Table Address – IDE0

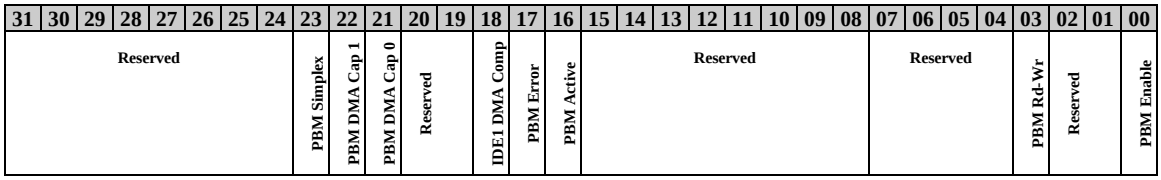
Address Offset: 0x04
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Table Address – IDE0																														Reserved	

This register defines the PRD Table Address register for IDE Channel #0 in the SiI3512 controller. The register bits are also mapped to PCI Configuration Space, Offset 0x74 and Base Address 5, Offset 0x04. See PRD Table Address – IDE0 section on page 52 for bit definitions.

PCI Bus Master – IDE1

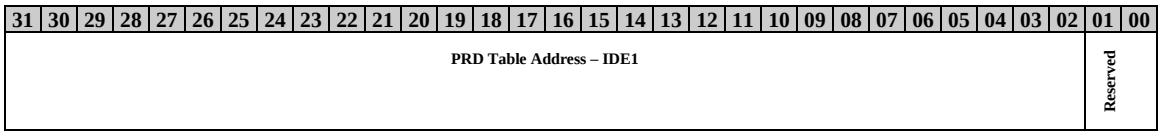
Address Offset: 0x08
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines the PCI bus master register for IDE Channel #1 in the SiI3512 controller. See PCI Bus Master – IDE1 section on page 52 for bit definitions.

PRD Table Address – IDE1

Address Offset: 0x0C
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines the PRD Table Address register for IDE Channel #1 in the SiI3512 controller. The register bits are also mapped to PCI Configuration Space, Offset 0x7C and Base Address 5, Offset 0x0C. See PRD Table Address – IDE1 section on page 53 for bit definitions.

Internal Register Space – Base Address 5

These registers are 32 bits wide and define the internal operation of the SiI3512 controller. The access types are defined as follows: R=read, W=write, and C=clearable by some write operation. Access to this register is through the PCI Memory space. The Base Address 5 can be disabled by setting input BA5_EN to low.

Table 22. SiI3512 Internal Register Space – Base Address 5

Address Offset	Register Name				Access Type
	31	16	15	00	
0x00	Reserved	PCI Bus Master Status – IDE0	Software Data	PCI Bus Master Command – IDE0	R/W
0x04	PRD Table Address – IDE0				R/W
0x08	Reserved	PCI Bus Master Status – IDE1	Reserved	PCI Bus Master Command – IDE1	R/W
0x0C	PRD Table Address – IDE1				R/W
0x10	PCI Bus Master Status – IDE1	PCI Bus Master Status2 – IDE0	Software Data	PCI Bus Master Command2 – IDE0	R/W
0x14	Reserved				-
0x18	Reserved	PCI Bus Master Status2 – IDE1	Reserved	PCI Bus Master Command2 – IDE1	R/W
0x1C	Reserved				-
0x20	PRD Address – IDE0				R
0x24	PCI Bus Master Byte Count – IDE0				R
0x28	PRD Address – IDE1				R
0x2C	PCI Bus Master Byte Count – IDE1				R
0x30	Reserved				-

Address Offset	Register Name				Access Type
	31 16		15 00		
0x34	Reserved				-
0x38	Reserved				-
0x3C	Reserved				-
0x40	FIFO Valid Byte Count – IDE0		FIFO Wr Request Control – IDE0	FIFO Rd Request Control – IDE0	R/W
0x44	FIFO Valid Byte Count – IDE1		FIFO Wr Request Control – IDE1	FIFO Rd Request Control – IDE1	R/W
0x48	System Configuration Status		System Command		R/W
0x4C	System Software Data				R/W
0x50	Flash Memory Address – Command and Status				R/W
0x54	Reserved		Reserved	Flash Memory Data	R/W
0x58	EEPROM Memory Address – Command and Status				R/W
0x5C	Reserved			EEPROM Memory Data	R/W
0x60	FIFO Port – IDE0				R/W
0x64	Reserved				-
0x68	FIFO Byte1 Write Pointer – IDE0	FIFO Byte1 Read Pointer – IDE0	FIFO Byte0 Write Pointer – IDE0	FIFO Byte0 Read Pointer – IDE0	R
0x6C	FIFO Byte3 Write Pointer – IDE0	FIFO Byte3 Read Pointer – IDE0	FIFO Byte2 Write Pointer – IDE0	FIFO Byte2 Read Pointer – IDE0	R
0x70	FIFO Port – IDE1				R/W
0x74	Reserved				-
0x78	FIFO Byte1 Write Pointer – IDE1	FIFO Byte1 Read Pointer – IDE1	FIFO Byte0 Write Pointer – IDE1	FIFO Byte0 Read Pointer – IDE1	R
0x7C	FIFO Byte3 Write Pointer – IDE1	FIFO Byte3 Read Pointer – IDE1	FIFO Byte2 Write Pointer – IDE1	FIFO Byte2 Read Pointer – IDE1	R
0x80	IDE0 TF Starting Sector Number	IDE0 TF Sector Count	IDE0 TF Features IDE0 TF Error	IDE0 TF Data	R/W
0x84	IDE0 TF Command+Status	IDE0 TF Device+Head	IDE0 TF Cylinder High	IDE0 TF Cylinder Low	R/W
0x88	Reserved	IDE0 TF Device Control Auxiliary Status	Reserved	Reserved	R/W
0x8C	IDE0 Read Ahead Data				R/W
0x90	IDE0 TF Starting Sector Number2	IDE0 TF Sector Count2	IDE0 TF Features2 IDE0 TF Error2	Reserved	R/W
0x94	IDE0 TF Cmd	IDE0 TF Device+Head2	IDE0 TF Cylinder High2	IDE0 TF Cylinder Low2	R/W
0x98	IDE0 TF Cylinder High 2 Ext	IDE0 TF Cylinder Low 2 Ext	IDE0 TF Starting Sector 2 Ext	IDE0 TF Sector Count 2 Ext	R/W
0x9C	IDE0 Virtual DMA/PIO Read Ahead Byte Count				R/W
0xA0	Reserved		IDE0 Config + Status	IDE0 Cmd + Status	R/W
0xA4	Reserved				R/W
0xA8	Reserved				R/W
0xAC	Reserved				R/W
0xB0	IDE0 Test Register				R/W
0xB4	Reserved			IDE0 Data Transfer Mode	R/W
0xB8	Reserved				-
0xBC	Reserved				-

Address Offset	Register Name				Access Type
	31	16	15	00	
0xC0	IDE1 TF Starting Sector Number	IDE1 TF Sector Count	IDE1 TF Features IDE1 TF Error	IDE1 TF Data	R/W
0xC4	IDE1 TF Command+Status	IDE1 TF Device+Head	IDE1 TF Cylinder High	IDE1 TF Cylinder Low	R/W
0xC8	Reserved	IDE1 TF Device Control Auxiliary Status	Reserved		R/W
0xCC	IDE1 Read Ahead Data				R/W
0xD0	IDE1 TF Starting Sector Number2	IDE1 TF Sector Count2	IDE1 TF Features2 IDE1 TF Error2	Reserved	R/W
0xD4	IDE1 TF Cmd	IDE1 TF Device+Head2	IDE1 TF Cylinder High2	IDE1 TF Cylinder Low2	R/W
0xD8	IDE1 TF Cylinder High Ext 2	IDE1 TF Cylinder Low Ext 2	IDE1 TF Starting Sector Ext 2	IDE1 TF Sector Count Ext 2	R/W
0xDC	IDE1 Virtual DMA/PIO Read Ahead Byte Count				R/W
0xE0	Reserved		IDE1 Config + Status	IDE1 Cmd + Status	R/W
0xE4	Reserved				R/W
0xE8	Reserved				R/W
0xEC	Reserved				R/W
0xF0	IDE1 Test Register				R/W
0xF4	Reserved			IDE1 Data Transfer Mode	R/W
0xF8	Reserved				-
0xFC	Reserved				-
0x100	SControl (Channel 0)				R/W
0x104	SStatus (Channel 0)				R/W
0x108	SError (Channel 0)				R/C
0x10C	SActive (Channel 0)				-
0x110	Reserved				-
0x114	Reserved				-
0x118	Reserved				-
0x11C	Reserved				-
0x120	Reserved				-
0x124	Reserved				-
0x128	Reserved				-
0x12C	Reserved				-
0x130	Reserved				-
0x134	Reserved				-
0x138	Reserved				-
0x13C	Reserved				-
0x140	SMisc (Channel 0)				R/W
0x144	PHY Configuration				R/W
0x148	SIEN (Channel 0)				R/W
0x14C	SFISCfg (Channel 0)				R/W
0x150	Reserved				-
0x154	Reserved				-
0x158	Reserved				-
0x15C	Reserved				-
0x160	RxFIS0 (Channel 0)				R
0x164	RxFIS1 (Channel 0)				R

Address Offset	Register Name	Access Type
	31 16 | 15 00	
0x168	RxFIS2 (Channel 0)	R
0x16C	RxFIS3 (Channel 0)	R
0x170	RxFIS4 (Channel 0)	R
0x174	RxFIS5 (Channel 0)	R
0x178	RxFIS6 (Channel 0)	R
0x17C	Reserved	-
0x180	SControl (Channel 1)	R/W
0x184	SStatus (Channel 1)	R/W
0x188	SError (Channel 1)	R/C
0x18C	SActive (Channel 1)	-
0x190	Reserved	-
0x194	Reserved	-
0x198	Reserved	-
0x19C	Reserved	-
0x1A0	Reserved	-
0x1A4	Reserved	-
0x1A8	Reserved	-
0x1AC	Reserved	-
0x1B0	Reserved	-
0x1B4	Reserved	-
0x1B8	Reserved	-
0x1BC	Reserved	-
0x1C0	SMisc (Channel 1)	R/W
0x1C4	Reserved	R/W
0x1C8	SIEN (Channel 1)	R/W
0x1CC	SFISCfg (Channel 1)	R/W
0x1D0	Reserved	-
0x1D4	Reserved	-
0x1D8	Reserved	-
0x1DC	Reserved	-
0x1E0	RxFIS0 (Channel 1)	R
0x1E4	RxFIS1 (Channel 1)	R
0x1E8	RxFIS2 (Channel 1)	R
0x1EC	RxFIS3 (Channel 1)	R
0x1F0	RxFIS4 (Channel 1)	R
0x1F4	RxFIS5 (Channel 1)	R
0x1F8	RxFIS6 (Channel 1)	R
0x1FC	Reserved	-

PCI Bus Master – IDE0

Address Offset: 0x00
Access Type: Read/Write
Reset Value: 0x0000_XX00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	Reserved		IDE0 DMA Comp	PBM Error	PBM Active	IDE Watchdog	IDE1 DMA Comp	Software						Reserved				PBM Rd-Wr	Reserved		PBM Enable

This register defines the PCI bus master register for IDE Channel #0 in the SiI3512 device. The register bits are defined below.

- **Bit [31:24]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [23]**: PBM Simplex (R) – PCI Bus Master Simplex Only. This read-only bit field is hardwired to zero to indicate that both IDE channels can operate as PCI bus master at any time.
- **Bit [22]**: PBM DMA Cap 1 (R/W) – PCI Bus Master DMA Capable – Device 1. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [21]**: PBM DMA Cap 0 (R/W) – PCI Bus Master DMA Capable – Device 0. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [20:19]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [18]**: IDE0 DMA Comp (R/W1C) – IDE0 DMA Completion Interrupt. During write DMA operation, this bit set indicates that the IDE0 interrupt has been asserted and all data has been written to system memory. During Read DMA, this bit set indicates that the IDE0 interrupt has been asserted. This bit must be W1C by software when set during DMA operation (bit 0 is set). During normal operation, this bit reflects IDE0 interrupt line.
- **Bit [17]**: PBM Error (R/W1C) – PCI Bus Master Error – IDE0. This bit set indicates that a PCI bus error occurred while the SiI3512 controller was bus master. Additional information is available in the PCI Status register in PCI Configuration space.
- **Bit [16]**: PBM Active (R) – PCI Bus Master Active – IDE0. This bit set indicates that the SiI3512 controller is currently active in a data transfer as PCI bus master. This bit is cleared by the hardware when all data transfers have completed or PBM Enable bit is not set.
- **Bit[15]**: IDE Watchdog Timer Status (R) – This bit is an ORed result of bit 12 in IDE0 Task File Timing + Configuration + Status and bit 12 of IDE1 Task File Timing + Configuration + Status registers. When set indicates that either IDE0 or IDE1 Watchdog timer has expired.
- **Bit[14]**: IDE1 Interrupt Status (R) – This bit is a copy of Bit[18] IDE1 DMA Completion Interrupt in PCI Bus Master – IDE1.
- **Bit [13:08]**: Software Data (R/W) – System Software Data Storage. This bit field is used for read/write data storage by the system. The properties of this bit field are detailed below.

Table 23. Software Data Byte, Base Address 5, Offset 0x00

Bit Location	Default	Description
[13:12]	0bXX	Not cleared by any reset
[11:10]	0b00	Cleared by PCI reset
[09:08]	0bXX	Cleared only by a D0–D3 power state change

- **Bit [07:04]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [03]**: PBM Rd-Wr (R/W) – PCI Bus Master Read-Write Control. This bit is set to specify a DMA write operation from IDE0 to system memory. This bit is cleared to specify a DMA read operation from system memory to an IDE0 device.
- **Bit [02:01]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [00]**: PBM Enable (R/W) – PCI Bus Master Enable – IDE0. This bit is set to enable PCI bus master operations for IDE Channel #0. PCI bus master operations can be halted by clearing this bit, but will erase all state information in the control logic. If this bit is cleared while the PCI bus master is active, the operation will be

aborted and the data discarded. While this bit is set, accessing IDE0 Task File or PIO data registers will be terminated with Target-Abort.

PRD Table Address – IDE0

Address Offset: 0x04
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Table Address – IDE0																														Reserved	

This register defines the PRD Table Address register for IDE Channel #0 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:02]**: PRD Table Address (R/W) – Physical Region Descriptor Table Address. This bit field defines the Descriptor Table base address.
- **Bit [01:00]**: Reserved (R). This bit field is reserved and returns zeros on a read.

PCI Bus Master – IDE1

Address Offset: 0x08
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	Reserved		IDE1 DMA Comp	PBM Error	PBM Active	Reserved								Reserved					PBM Rd-Wr	Reserved		PBM Enable

This register defines the PCI bus master register for IDE Channel #1 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:24]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [23]**: PBM Simplex (R) – PCI Bus Master Simplex Only. This read-only bit field is hardwired to zero to indicate that both IDE channels can operate as PCI bus master at any time.
- **Bit [22]**: PBM DMA Cap 1 (R/W) – PCI Bus Master DMA Capable – Device 1. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [21]**: PBM DMA Cap 0 (R/W) – PCI Bus Master DMA Capable – Device 0. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [20:19]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [18]**: IDE1 DMA Comp (R/W1C) – IDE1 DMA Completion Interrupt. During write DMA operation, this bit set indicates that the IDE1 interrupt has been asserted and all data has been written to system memory. During Read DMA, this bit set indicates that the IDE1 interrupt has been asserted. This bit must be W1C by software when set during DMA operation (bit 0 is set). During normal operation, this bit reflects IDE1 interrupt line.
- **Bit [17]**: PBM Error (R/W1C) – PCI Bus Master Error – IDE1. This bit set indicates that a PCI bus error occurred while the SiI3512 controller was bus master. Additional information is available in the PCI Status register in PCI Configuration space.
- **Bit [16]**: PBM Active (R) – PCI Bus Master Active – IDE1. This bit set indicates that the SiI3512 controller is currently active in a data transfer as PCI bus master. This bit is cleared by the hardware when all data transfers have completed or PBM Enable bit is not set.
- **Bit [15:08]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [07:04]**: Reserved (R). This bit field is reserved and returns zeros on a read.

- **Bit [03]**: PBM Rd-Wr (R/W) – PCI Bus Master Read-Write Control. This bit is set to specify a DMA write operation from IDE1 to system memory. This bit is cleared to specify a DMA read operation from system memory to an IDE1 device.
- **Bit [02:01]**: Reserved (R). This bit field is reserved and returns zeros on a read.
- **Bit [00]**: PBM Enable (R/W) – PCI Bus Master Enable – IDE1. This bit is set to enable PCI bus master operations for IDE Channel #1. PCI bus master operations can be halted by clearing this bit, but will erase all state information in the control logic. If this bit is cleared while the PCI bus master is active, the operation will be aborted and the data discarded. While this bit is set, accessing IDE1 Task File or PIO data registers will be terminated with Target-Abort.

PRD Table Address – IDE1

Address Offset: 0x0C
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Table Address – IDE1																														Reserved	

This register defines the PRD Table Address register for IDE Channel #1 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:02]**: PRD Table Address (R/W) – Physical Region Descriptor Table Address. This bit field defines the Descriptor Table base address.
- **Bit [01:00]**: Reserved (R). This bit field is reserved and returns zeros on a read.

PCI Bus Master2 – IDE0

Address Offset: 0x10
Access Type: Read/Write
Reset Value: 0x0808_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE1 PBM Simplex	IDE1 PBM DMA Cap 0	IDE1 PBM DMA Cap 1	IDE1 Watchdog	IDE1 Buffer Empty	IDE1 DMA Comp	IDE1 PBM Error	IDE1 PBM Active	IDE0 PBM Simplex	IDE0 PBM DMA Cap 1	IDE0 PBM DMA Cap 0	IDE0 Watchdog	IDE0 Buffer Empty	IDE0 DMA Comp	IDE0 PBM Error	IDE0 PBM Active	IDE Watchdog	IDE1 DMA Comp	Software						Reserved	SATAINT1	Reserved	SATAINT0	PBM Rd-Wr	Reserved		PBM Enable

This register defines the second PCI bus master register for IDE Channel #0 in the SiI3512 controller. The system must access these register bits through this address to enable the Large Block Transfer Mode.

The register bits are defined below.

- **Bit [31:29]**: (R) – These bits are copy of PCI Bus Master IDE1 bits [23:21].
- **Bit [28]**: IDE1 Watchdog (R): This bit is a copy of bit 12 in IDE1 Task File Configuration + Status register.
- **Bit [27]**: IDE1 Buffer empty (R). This bit set indicates the IDE1 FIFO is empty.
- **Bit [26:24]**: (R) – These bits are copy of PCI Bus Master IDE1 bits [18:16].
- **Bit [23]**: PBM Simplex (R) – PCI Bus Master Simplex Only. This read-only bit field is hardwired to zero to indicate that both IDE channels can operate as PCI bus master at any time.
- **Bit [22]**: PBM DMA Cap 1 (R/W) – PCI Bus Master DMA Capable – Device 1. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [21]**: PBM DMA Cap 0 (R/W) – PCI Bus Master DMA Capable – Device 0. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [20]**: IDE0 Watchdog (R): This bit is a copy of bit 12 in IDE0 Task File Configuration + Status register.

- **Bit [19]**: IDE0 Buffer empty (R). This bit set indicates the IDE0 FIFO is empty.
- **Bit [18]**: IDE0 DMA Comp (R/W1C) – IDE0 DMA Completion Interrupt. During write DMA operation, this bit set indicates that the IDE0 interrupt has been asserted and all data has been written to system memory. During Read DMA, This bit set indicates that the IDE0 interrupt has been asserted.
  This bit must be W1C by software when set during DMA operation (bit 0 is set). During normal operation, this bit reflects IDE0 interrupt line.
- **Bit [17]**: PBM Error (R/W1C) – PCI Bus Master Error – IDE0. This bit set indicates that a PCI bus error occurred while the SiI3512 device was bus master. Additional information is available in the PCI Status register in PCI Configuration space.
- **Bit [16]**: PBM Active (R) – PCI Bus Master Active – IDE0. This bit set indicates that the SiI3512 controller is currently active in a data transfer as PCI bus master. This bit is cleared by the hardware when all data transfers have completed or PBM Enable bit is not set.
- **Bit[15]**: IDE Watchdog Timer Status (R) – This bit is an ORed result of bit 12 in IDE1 Task File Timing + Configuration + Status and bit 12 of IDE0 Task File Timing + Configuration + Status registers. When set indicates that either IDE0 or IDE1 Watchdog timer has expired.
- **Bit[14]**: IDE1 DMA Completion Interrupt Status (R) – This bit is a copy of Bit[18] IDE1 DMA Completion Interrupt in PCI Bus Master – IDE1.
- **Bit [13:08]**: Software Data (R/W) – System Software Data Storage. This bit field is used for read/write data storage by the system. The properties of this bit field are detailed below.

Table 24. Software Data Byte, Base Address 5, Offset 0x10

Bit Location	Default	Description
[13:12]	0bXX	Not cleared by any reset
[11:10]	0b00	Cleared by PCI reset
[09:08]	0bXX	Cleared only by a D0-D3 power state change

- **Bit [07]**: Reserved (R) – This bit is reserved and returns zeros on a read.
- **Bit [06]**: SATAINT1 – This bit is the logical OR of all Serial ATA interrupt sources for Channel 1.
- **Bit [05]**: Reserved (R) – This bit is reserved and returns zeros on a read.
- **Bit [04]**: SATAINT0 – This bit is the logical OR of all Serial ATA interrupt sources for Channel 0.
- **Bit [03]**: PBM Rd-Wr (R/W) – PCI Bus Master Read-Write Control. This bit is set to specify a DMA write operation from IDE0 to system memory. This bit is cleared to specify a DMA read operation from system memory to an IDE0 device.
- **Bit [02:01]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [00]**: PBM Enable (R/W) – PCI Bus Master Enable – IDE0. This bit is set to enable PCI bus master operations for IDE Channel #0. PCI bus master operations can be halted by clearing this bit, but will erase all state information in the control logic. If this bit is cleared while the PCI bus master is active, the operation will be aborted and the data discarded. While this bit is set, accessing IDE0 Task File or PIO data registers will be terminated with Target-Abort.

PCI Bus Master2 – IDE1

Address Offset: 0x18
Access Type: Read/Write
Reset Value: 0x0008_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								PBM Simplex	PBM DMA Cap 1	PBM DMA Cap 0	IDE1 Watchdog	IDE1 Buffer Empty	IDE1 DMA Comp	PBM Error	PBM Active	Reserved								Reserved			SATAINT1	PBM Rd-Wr	Reserved		PBM Enable

This register defines the second PCI bus master register for IDE Channel #1 in the SiI3512 controller. The system must access these register bits through this address to enable the Large Block Transfer Mode.

- **Bit [31:24]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

- **Bit [23]**: PBM Simplex (R) – PCI Bus Master Simplex Only. This read-only bit field is hardwired to zero to indicate that both IDE channels can operate as PCI bus master at any time.
- **Bit [22]**: PBM DMA Cap 1 (R/W) – PCI Bus Master DMA Capable – Device 1. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [21]**: PBM DMA Cap 0 (R/W) – PCI Bus Master DMA Capable – Device 0. This bit field has no effect. The device is always capable of DMA as a PCI bus master.
- **Bit [20]**: IDE1 Watchdog (R). This bit is a copy of bit 12 in IDE1 Task File Timing + Configuration + Status register.
- **Bit [19]**: IDE1 Buffer empty (R) – This bit set indicates IDE1 FIFO is empty.
- **Bit [18]**: IDE1 DMA Comp (R/W1C) – IDE1 DMA Completion Interrupt. During write DMA operation, this bit set indicates that the IDE1 interrupt has been asserted and all data has been written to system memory. During Read DMA, this bit set indicates that the IDE1 interrupt has been asserted. This bit must be W1C by software when set during DMA operation (bit 0 is set). During normal operation, this bit reflects IDE1 interrupt line.
- **Bit [17]**: PBM Error (R/W1C) – PCI Bus Master Error – IDE1. This bit set indicates that a PCI bus error occurred while the SiI3512 controller was bus master. Additional information is available in the PCI Status register in PCI Configuration space.
- **Bit [16]**: PBM Active (R) – PCI Bus Master Active – IDE1. This bit set indicates that the SiI3512 controller is currently active in a data transfer as PCI bus master. This bit is cleared by the hardware when all data transfers have completed or PBM Enable bit is not set.
- **Bit [15:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07:05]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [04]**: SATAINT1 – This bit is the logical OR of all Serial ATA interrupt sources for Channel 1.
- **Bit [03]**: PBM Rd-Wr (R/W) – PCI Bus Master Read-Write Control. This bit is set to specify a DMA write operation from IDE1 to system memory. This bit is cleared to specify a DMA read operation from system memory to an IDE1 device.
- **Bit [02:01]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [00]**: PBM Enable (R/W) – PCI Bus Master Enable – IDE1. This bit is set to enable PCI bus master operations for IDE Channel #1. PCI bus master operations can be halted by clearing this bit, but will erase all state information in the control logic. If this bit is cleared while the PCI bus master is active, the operation will be aborted and the data discarded. While this bit is set, accessing IDE1 Task File or PIO data registers will be terminated with Target-Abort.

PRD Address – IDE0

Address Offset: 0x20
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Address																															

This register reflects the current DMA address and uses for diagnostic purposes only.

- **Bit [31:00]**: PRD Address (R) – This field is the current DMA0 Address.

PCI Bus Master Byte Count – IDE0

Address Offset: 0x24
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
End of Table	Byte Count High															Byte Count Low															

This register defines the byte count register in the PCI bus master logic for IDE Channel #0 in the SiI3512. The register bits are defined below.

- **Bit [31]**: End of Table (R) – This bit set indicates that this is the last entry in the PRD table.
- **Bit [30:16]** Byte Count High (R) – This bit field is the PRD entry byte count extension for Large Block Transfer Mode. Under generic mode, this bit field is reserved and returns zeros on a read.
- **Bit [15:00]** Byte Count Low (R) – This bit field reflects the current DMA0 byte count value.

PRD Address – IDE1

Address Offset: 0x28
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PRD Address																															

This register reflects the current DMA1 Address and uses for diagnostic purposes only.

- **Bit [31:00]**: PRD Address (R) – This field is the current DMA1 Address.

PCI Bus Master Byte Count – IDE1

Address Offset: 0x2C
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
End of Table	Byte Count High															Byte Count Low															

This register defines the byte count register in the PCI bus master logic for IDE Channel #1 in the SiI3512 controller. The register bits are defined below.

- **Bit [31]**: End of Table (R) – This bit set indicates that this is the last entry in the PRD table.
- **Bit [30:16]**: Byte Count High (R) – This bit field is the PRD entry byte count extension for Large Block Transfer Mode. Under generic mode, this bit field is reserved and returns zeros on a read.
- **Bit [15:00]**: Byte Count Low (R) – This bit field reflects the current DMA1 byte count value.

FIFO Valid Byte Count and Control – IDE0

Address Offset: 0x40
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved							FIFO Valid Byte Count – IDE0									Reserved					FIFO Wr Req Ctrl – IDE0			Reserved						FIFO Rd Req Ctrl – IDE0		

This register defines the FIFO valid byte count register and PCI bus request control for IDE Channel #0 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:25]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [24:16]**: FIFO Valid Byte Count – IDE0 (R) – This bit field provides the valid byte count for the data FIFO for IDE Channel #0. A value of 0x000 indicates empty, while a value of 0x100 indicates a full FIFO with 256 bytes.
- **Bit [15:11]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [10:08]**: FIFO Wr Req Ctrl – IDE0 (R/W) – FIFO Write Request Control. This bit field defines the FIFO threshold to assign DMA0 priority when requesting a PCI bus write operation. A value of 0x00 indicates that DMA0 write request priority is set whenever the FIFO contains greater than 32 bytes, while a value of 0x07 indicates that DMA0 write request priority is set whenever the FIFO contains greater than 7x32 bytes (=224 bytes). This bit field is useful when two DMA channels are competing for accessing PCI bus.

  When the two DMA channels request the PCI bus at the same time, the one with the higher priority will have the bus when it's granted to the SiI3512 controller. If the two DMA channels have the same priority, the channel that had the bus last will have the bus when it's granted to the SiI3512 controller.

  When one DMA channel is controlling the PCI bus and the other channel requests the PCI bus, if the channel currently controlling the PCI bus has the same or higher priority, it remains controlling the bus. However, if the channel requesting the PCI bus has higher priority, the lower priority channel terminates the PCI transaction, yielding the bus to the channel with the higher priority.

- **Bit [07:03]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [02:00]**: FIFO Rd Req Ctrl – IDE0 (R/W) – FIFO Read Request Control. This bit field defines the FIFO threshold to assign DMA0 priority when requesting a PCI bus read operation. A value of 0x00 indicates that DMA0 read request priority is set whenever the FIFO has greater than 32 bytes available space, while a value of 0x07 indicates that DMA0 read request priority is set whenever the FIFO has greater than 7x32 bytes (=224 bytes) available space. This bit field is useful when two DMA channels are competing for accessing the PCI bus.

  When the two DMA channels request the PCI bus at the same time, the channel that had the bus last will have the bus when it's granted to the SiI3512 controller.

FIFO Valid Byte Count and Control – IDE1

Address Offset: 0x44
Access Type: Read /Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved							FIFO Valid Byte Count – IDE1									Reserved					FIFO Wr Req Ctrl – IDE1			Reserved						FIFO Rd Req Ctrl – IDE1		

This register defines the FIFO valid byte count register and PCI bus request control for IDE Channel #1 in the SiI3512. The register bits are defined below.

- **Bit [31:25]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

- **Bit [24:16]**: FIFO Valid Byte Count – IDE1 (R) – This bit field provides the valid byte count for the data FIFO for IDE Channel #1. A value of 0x000 indicates empty, while a value of 0x100 indicates a full FIFO with 256 bytes.
- **Bit [15:11]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [10:08]**: FIFO Wr Req Ctrl – IDE1 (R/W) – FIFO Write Request Control. This bit field defines the FIFO threshold to assign DMA1 priority when requesting a PCI bus write operation. A value of 0x00 indicates that DMA1 write request priority is set whenever the FIFO contains greater than 32 bytes, while a value of 0x07 indicates that DMA1 write request priority is set whenever the FIFO contains greater than 7x32 bytes (=224 bytes). This bit field is useful when two DMA channels are competing for accessing PCI bus.

  When the two DMA channels request the PCI bus at the same time, the one with the higher priority will have the bus when it's granted to the SiI3512 controller. If the two DMA channels have the same priority, the channel that had the bus last will have the bus when it's granted to the SiI3512 controller.

  When one DMA channel is controlling the PCI bus and the other channel requests the PCI bus, if the channel currently controlling the PCI bus has the same or higher priority, it remains controlling the bus. However, if the channel requesting the PCI bus has higher priority, the lower priority channel terminates the PCI transaction, yielding the bus to the channel with the higher priority.

- **Bit [07:03]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [02:00]**: FIFO Rd Req Ctrl – IDE1 (R/W) – FIFO Read Request Control. This bit field defines the FIFO threshold to assign DMA1 priority when requesting a PCI bus read operation. A value of 0x00 indicates that DMA1 read request priority is set whenever the FIFO has greater than 32 bytes available space, while a value of 0x07 indicates that DMA1 read request priority is set whenever the FIFO has greater than 7x32 bytes (=224 bytes) available space. This bit field is useful when two DMA channels are competing for accessing the PCI bus.

  When the two DMA channels request the PCI bus at the same time, the channel that had the bus last will have the bus when it's granted to the SiI3512 device.

System Configuration Status – Command

Address Offset: 0x48
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								IDE1 Int Block	IDE0 Int Block	Reserved					M66EN	Reserved								IDE0 Module Rst	IDE1 Module Rst	FF0 Module Rst	FF1 Module Rst	Reserved		ARB Module Rst	PBM Module Rst

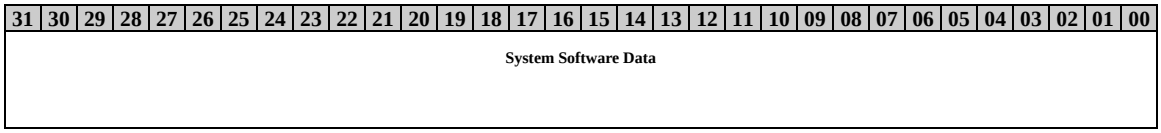
This register defines the system configuration status and command register for the SiI3512 controller. The register bits are defined below.

- **Bit [31:24]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [23]**: IDE1 Int Block (R/W) – IDE1 Interrupt Block. This bit is set to block interrupts from the IDE Channel #1 to the PCI bus.
- **Bit [22]**: IDE0 Int Block (R/W) – IDE0 Interrupt Block. This bit is set to block interrupts from the IDE Channel #0 to the PCI bus.
- **Bit [21:17]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [16]**: M66EN (R) – PCI 66-MHz Enable. This bit reflects input pin M66EN.
- **Bit [15:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07]**: IDE0 Module Rst (R/W) – IDE0 Module Reset. This bit is set to reset the interface logic for the IDE Channel #0.
- **Bit [06]**: IDE1 Module Rst (R/W) – IDE1 Module Reset. This bit is set to reset the interface logic for the IDE Channel #1.
- **Bit [05]**: FF0 Module Rst (R/W) – FF0 Module Reset. This bit is set to reset the logic in the FIFO for IDE Channel #0.

- **Bit [04]**: FF1 Module Rst (R/W) – FF1 Module Reset. This bit is set to reset the logic in the FIFO for IDE Channel #1.
- **Bit [03:02]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [01]**: ARB Module Rst (R/W) – ARB Module Reset. This bit is set to reset the internal logic for the PCI-IDE arbiter.
- **Bit [00]**: PBM Module Rst (R/W) – PBM Module Reset. This bit is set to reset the internal logic for the PCI bus master state machine.

System Software Data Register

Address Offset: 0x4C
Access Type: Read/Write
Reset Value: Undefined

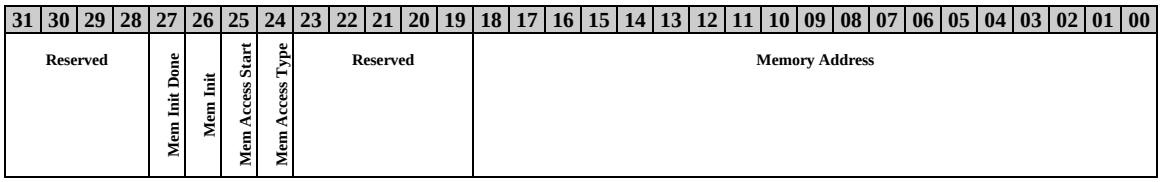


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
System Software Data																															

This register is used by the software for non-resettable data storage. The contents are unknown on power-up and are never cleared by any type of reset.

Flash Memory Address – Command + Status

Address Offset: 0x50
Access Type: Read/Write
Reset Value: 0x0800_0000



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved				Mem Init Done	Mem Init	Mem Access Start	Mem Access Type	Reserved					Memory Address																		

This register defines the address and command/status register for flash memory interface in the SiI3512 controller. The register bits are defined below.

- **Bit [31:28]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [27]**: Memory Init Done (R) – This bit set indicates that the memory initialization sequence is done. The memory sequence is activated upon the release of reset.
- **Bit [26]**: Mem Init (R) – Memory Initialized. This bit set indicates that the memory was initialized properly (a correct data sequence was read from the flash.)
- **Bit [25]**: Mem Access Start (R/W) – Memory Access Start. This bit is set to initiate an operation to flash memory. This bit is cleared by the chip when the operation is complete.
- **Bit [24]**: Mem Access Type (R/W) – Memory Access Type. This bit is set to define a read operation from flash memory. This bit is cleared to define a write operation to flash memory.
- **Bit [23:19]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [18:00]**: Memory Address (R/W). This bit field is programmed with the address for a flash memory read or write access.

Flash Memory Data

Address Offset: 0x54
Access Type: Read/Write
Reset Value: 0x0000_FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																Reserved								Memory Data							

This register defines the data register for the flash memory in the SiI3512 controller. The system writes to this register for a write operation to flash memory, and reads from this register on a read operation from flash memory.

- **Bit [31:16]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [15:08]**: Reserved (R)
- **Bit [07:00]**: Memory Data (R/W) – Flash Memory Data. This bit field is used for flash write data on a write operation, and returns the flash read data on a read operation.

EEPROM Memory Address – Command + Status

Address Offset: 0x58
Access Type: Read/Write
Reset Value: 0x1800_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved			Mem Error	Mem Init Done	Mem Init	Mem Access Start	Mem Access Type	Reserved																Mem Address							

This register defines the address and command/status register for EEPROM memory interface in the SiI3512 controller. The register bits are defined below.

- **Bit [31:29]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [28]**: Mem Error (R/W1C) – Memory Access Error. This bit set indicates that the EEPROM interface logic detects three NAKs from the memory device (EEPROM most likely not present.)
- **Bit [27]**: Mem Init Done (R) – Memory Initialization Done. This bit set indicates that the memory initialization sequence is done. The memory initialization sequence is activated upon the release of reset.
- **Bit [26]**: Mem Init (R) – Memory Initialized. This bit set indicates that the memory was initialized properly (a correct data sequence was read from the EEPROM.)
- **Bit [25]**: Mem Access Start (R/W) – Memory Access Start. This bit is set to initiate an operation to EEPROM memory. This bit is cleared by the chip when the operation is complete.
- **Bit [24]**: Mem Access Type (R/W) – Memory Access Type. This bit is set to define a read operation from EEPROM memory. This bit is cleared to define a write operation to EEPROM memory.
- **Bit [23:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07:00]**: Memory Address (R/W). This bit field is programmed with the address for an EEPROM read or write access.

EEPROM Memory Data

Address Offset: 0x5C
Access Type: Read/Write
Reset Value: 0x0000_00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Memory Data							

This register defines the data register for EEPROM memory interface in the SiI3512 controller. The system writes to this register for a write operation to EEPROM memory, and reads from this register on a read operation from EEPROM memory. The register bits are defined below.

- **Bit [31:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07:00]**: Memory Data (R/W) – EEPROM Memory Data. This bit field is used for EEPROM write data on a write operation, and returns the EEPROM read data on a read operation.

FIFO Port – IDE0

Address Offset: 0x60
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Port – IDE0																															

This register defines the direct access register for the FIFO port of IDE Channel #0 in the SiI3512 controller. This register is used for hardware debugging purposes only. The system can read from or write to this register for direct access to the data FIFO between the PCI bus and IDE Channel #0. While DMA0 is active, reading this register will be terminated with Target-Abort.

FIFO Pointers1– IDE0

Address Offset: 0x68
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Byte 1 Wr Pointer – IDE0								FIFO Byte 1 Rd Pointer – IDE0								FIFO Byte 0 Wr Pointer – IDE0								FIFO Byte 0 Rd Pointer – IDE0							

This register provides visibility into the data FIFO for IDE Channel #0 in the SiI3512 controller. The data FIFO is organized as a four byte-wide x 64 deep memory array. There are separate write and read pointer for each of the byte slices. This register is used for hardware debugging purposes only. The register bits are defined below.

- **Bit [31:24]**: FIFO Byte 1 Wr Pointer – IDE0 (R) FIFO Byte 1 Write Pointer. This bit field provides the status on the write pointer for Byte 1.
- **Bit [23:16]**: FIFO Byte 1 Rd Pointer – IDE0 (R) FIFO Byte 1 Read Pointer. This bit field provides the status on the read pointer for Byte 1.
- **Bit [15:08]**: FIFO Byte 0 Wr Pointer – IDE0 (R) FIFO Byte 0 Write Pointer. This bit field provides the status on the write pointer for Byte 0.
- **Bit [07:00]**: FIFO Byte 0 Rd Pointer – IDE0 (R) FIFO Byte 0 Read Pointer. This bit field provides the status on the read pointer for Byte 0.

FIFO Pointers2– IDE0

Address Offset: 0x6C
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Byte 3 Wr Pointer – IDE0								FIFO Byte 3 Rd Pointer – IDE0								FIFO Byte 2 Wr Pointer – IDE0								FIFO Byte 2 Rd Pointer – IDE0							

This register provides visibility into the data FIFO for IDE Channel #0 in the SiI3512 controller. The data FIFO is organized as a four byte-wide x 64 deep memory array. There are separate write and read pointer for each of the byte slices. This register is used for hardware debugging purposes only. The register bits are defined below.

- **Bit [31:24]**: FIFO Byte 3 Wr Pointer – IDE0 (R) FIFO Byte 3 Write Pointer. This bit field provides the status on the write pointer for Byte 3.
- **Bit [23:16]**: FIFO Byte 3 Rd Pointer – IDE0 (R) FIFO Byte 3 Read Pointer. This bit field provides the status on the read pointer for Byte 3.
- **Bit [15:08]**: FIFO Byte 2 Wr Pointer – IDE0 (R) FIFO Byte 2 Write Pointer. This bit field provides the status on the write pointer for Byte 2.
- **Bit [07:00]**: FIFO Byte 2 Rd Pointer – IDE0 (R) FIFO Byte 2 Read Pointer. This bit field provides the status on the read pointer for Byte 2.

FIFO Port – IDE1

Address Offset: 0x70
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Port – IDE1																															

This register defines the direct access register for the FIFO port of IDE Channel #1 in the SiI3512 controller. This register is used for hardware debugging purposes only. The system can read from or write to this register for direct access to the data FIFO between the PCI bus and IDE Channel #1. While DMA1 is active, reading this register will be terminated with Target-Abort.

FIFO Pointers1– IDE1

Address Offset: 0x78
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Byte 1 Wr Pointer – IDE1								FIFO Byte 1 Rd Pointer – IDE1								FIFO Byte 0 Wr Pointer – IDE1								FIFO Byte 0 Rd Pointer – IDE1							

This register provides visibility into the data FIFO for IDE Channel #1 in the SiI3512 controller. The data FIFO is organized as a four byte-wide x 64 deep memory array. There are separate write and read pointer for each of the byte slices. This register is used for hardware debugging purposes only. The register bits are defined below.

- **Bit [31:24]**: FIFO Byte 1 Wr Pointer – IDE1 (R) FIFO Byte 1 Write Pointer. This bit field provides the status on the write pointer for Byte 1.
- **Bit [23:16]**: FIFO Byte 1 Rd Pointer – IDE1 (R) FIFO Byte 1 Read Pointer. This bit field provides the status on the read pointer for Byte 1.

- **Bit [15:08]**: FIFO Byte 0 Wr Pointer – IDE1 (R) FIFO Byte 0 Write Pointer. This bit field provides the status on the write pointer for Byte 0.
- **Bit [07:00]**: FIFO Byte 0 Rd Pointer – IDE1 (R) FIFO Byte 0 Read Pointer. This bit field provides the status on the read pointer for Byte 0.

FIFO Pointers2– IDE1

Address Offset: 0x7C
Access Type: Read Only
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIFO Byte 3 Wr Pointer – IDE1								FIFO Byte 3 Rd Pointer – IDE1								FIFO Byte 2 Wr Pointer – IDE1								FIFO Byte 2 Rd Pointer – IDE1							

This register provides visibility into the data FIFO for IDE Channel #1 in the SiI3512 controller. The data FIFO is organized as a four byte-wide x 64 deep memory array. There are separate write and read pointer for each of the byte slices. This register is used for hardware debugging purposes only. The register bits are defined below.

- **Bit [31:24]**: FIFO Byte 3 Wr Pointer – IDE1 (R) FIFO Byte 3 Write Pointer. This bit field provides the status on the write pointer for Byte 3.
- **Bit [23:16]**: FIFO Byte 3 Rd Pointer – IDE1 (R) FIFO Byte 3 Read Pointer. This bit field provides the status on the read pointer for Byte 3.
- **Bit [15:08]**: FIFO Byte 2 Wr Pointer – IDE1 (R) FIFO Byte 2 Write Pointer. This bit field provides the status on the write pointer for Byte 2.
- **Bit [07:00]**: FIFO Byte 2 Rd Pointer – IDE1 (R) FIFO Byte 2 Read Pointer. This bit field provides the status on the read pointer for Byte 2.

IDE0 Task File Register 0

Address Offset: 0x80
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Task File Starting Sector Number								IDE0 Task File Sector Count								IDE0 Task File Features (W) IDE0 Task File Error (R)															
																								IDE0 Data (byte access)							
																IDE0 Data (word access)															
IDE0 Data (dword access)																															

This register contains some of the IDE Channel #0 Task File registers and provides access to the IDE data bus. Access to this register is determined by the PCI bus Byte Enables at the time of the read or write operation, i.e., what is accessed is determined by the address and by the size of the access. The register bits are defined below.

- **Bit [31:00]**: IDE0 Data (R/W). This bit field provides access to the IDE0 Data. This register can be accessed as an 8-bit, 16-bit, or 32-bit word.
- **Bit [31:24]**: IDE0 Task File Starting Sector Number (R/W). This bit field defines the IDE0 Task File Starting Sector Number register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [23:16]**: IDE0 Task File Sector Count (R/W). This bit field defines the IDE0 Task File Sector Count register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [15:08]**: IDE0 Task File Features (W). This write-only bit field defines the IDE0 Task File Features register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [15:08]**: IDE0 Task File Error (R) – This read-only bit field defines the IDE0 Task File Error register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.

IDE0 Task File Register 1

Address Offset: 0x84
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Task File Command + Status								IDE0 Task File Device+Head								IDE0 Task File Cylinder High								IDE0 Task File Cylinder Low							

This register defines one of the IDE Channel #0 Task File registers in the SiI3512 controller. Access to these bit fields is permitted if the PCI bus Byte Enable is active for one byte only.

The register bits are defined below.

- **Bit [31:24]**: IDE0 Task File Command (W). This write-only bit field defines the IDE0 Task File Command register.
- **Bit [31:24]**: IDE0 Task File Status (R) – This read-only bit field defines the IDE0 Task File Status register.
- **Bit [23:16]**: IDE0 Task File Device+Head (R/W). This bit field defines the IDE0 Task File Device and Head register.
- **Bit [15:08]**: IDE0 Task File Cylinder High (R/W). This bit field defines the IDE0 Task File Cylinder High register.
- **Bit [07:00]**: IDE0 Task File Cylinder Low (R/W). This bit field defines the IDE0 Task File Cylinder Low register.

IDE0 Task File Register 2

Address Offset: 0x88
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								IDE0 Task File Device Control IDE0 Task File Auxiliary Status								Reserved								Reserved							

This register defines one of the IDE Channel #0 Task File registers in the SiI3512 controller. Access to these bit fields is permitted if the PCI bus Byte Enable is active for one byte only.

The register bits are defined below.

- **Bit [31:24]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [23:16]**: IDE0 Task File Device Control (W). This bit field defines the IDE0 Task File Device Control register.
- **Bit [23:16]**: IDE0 Task File Auxiliary Status (R) – This bit field defines the IDE0 Task File Auxiliary Status register.
- **Bit [15:00]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

IDE0 Read Ahead Data

Address Offset: 0x8C
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Read Ahead Data																															

This register defines the read ahead data port for PIO transfers on IDE Channel #0 in the SiI3512 controller. This register can be accessed as an 8-bit, 16-bit, or 32-bit word, depending upon the PCI bus Byte Enables. The data written to this register must be zero-aligned.

IDE0 Task File Register 0 – Command Buffering

Address Offset: 0x90
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Task File Starting Sector Number								IDE0 Task File Sector Count								IDE0 Task File Features								Unused							

This register defines one of the IDE Channel #0 Task File registers used for Command Buffered accesses in the SiI3512 controller. The register bits are defined below.

- **Bit [31:24]**: IDE0 Task File Starting Sector Number (R/W). This bit field defines the IDE0 Task File Starting Sector Number register.
- **Bit [23:16]**: IDE0 Task File Sector Count (R/W). This bit field defines the IDE0 Task File Sector Count register. Access to this bit field is permitted only if the PCI bus Byte Enable for byte 2 is active.
- **Bit [15:08]**: IDE0 Task File Features (W). This write-only bit field defines the IDE0 Task File Features register. Access to this bit field is permitted only if the PCI bus Byte Enable for byte 1 is active.

IDE0 Task File Register 1 – Command Buffering

Address Offset: 0x94
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Task File Command								IDE0 Task File Device+Head								IDE0 Task File Cylinder High								IDE0 Task File Cylinder Low							

This register defines one of the IDE Channel #0 Task File registers used for Command Buffered accesses in the SiI3512 controller. The register bits are defined below.

- **Bit [31:24]**: IDE0 Task File Command (W). This write-only bit field defines the IDE0 Task File Command register.
- **Bit [23:16]**: IDE0 Task File Device+Head (R/W). This bit field defines the IDE0 Task File Device and Head register.
- **Bit [15:08]**: IDE0 Task File Cylinder High (R/W). This bit field defines the IDE0 Task File Cylinder High register.
- **Bit [07:00]**: IDE0 Task File Cylinder Low (R/W). This bit field defines the IDE0 Task File Cylinder Low register.

IDE0 Extended Task File Register – Command Buffering

Address Offset: 0x98
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Task File Cylinder High Ext								IDE0 Task File Cylinder Low Ext								IDE0 Task File Start Sector Ext								IDE0 Task File Sector Count Ext							

This register defines one of the IDE Channel #0 Task File registers used for Command Buffered accesses in the SiI3512 controller. The register bits are defined below. If this register is written, the IDE0 Task File Device+Head byte of the IDE0 Task File Register 1 – Command Buffering register must not be written.

- **Bit [31:24]**: IDE0 Task File Cylinder High Ext(R/W). This write-only bit field defines the IDE0 Task File Extended Cylinder High register.
- **Bit [23:16]**: IDE0 Task File Cylinder Low Ext (R/W). This bit field defines the IDE0 Task File Extended Cylinder Low register.
- **Bit [15:08]**: IDE0 Task File Start Sector Ext (R/W). This bit field defines the IDE0 Task File Extended Start Sector register.
- **Bit [07:00]**: IDE0 Task File Sector Count Ext (R/W). This bit field defines the IDE0 Task File Extended Sector Count register.

IDE0 Virtual DMA/PIO Read Ahead Byte Count

Address Offset: 0x9C
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE0 Virtual DMA/PIO Read Ahead Byte Count																															Not Used

This register defines the read ahead byte count register for Virtual DMA and PIO Read Ahead transfers on IDE Channel #0 in the SiI3512 controller. In Virtual DMA mode (PCI bus master DMA with PIO transfers on the IDE), all 32 bits are used as the word-aligned byte count. In PIO Read Ahead mode, only the lower 16 bits are used as the word-aligned byte count.

IDE0 Task File Configuration + Status

Address Offset: 0xA0
Access Type: Read/Write
Reset Value: 0x6515_0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																Reserved	Watchdog Int Ena	Watchdog Ena	Watchdog Timeout	Interrupt Status	Virtual DMA Int	Reserved							Channel Rst	Buffered Cmd	Reserved

This register defines the task file configuration and status register for IDE Channel #0 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:16]**: Reserved (R) – This bit field is reserved and defaults to 0x6515.
- **Bit [15]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

- **Bit [14]:** Watchdog Int Ena (R/W) – IDE0 Watchdog Interrupt Enable. This bit is set to enable Interrupt when Watchdog timer expired.
- **Bit [13]**: Watchdog Ena (R/W) – IDE0 Watchdog Timer Enable. This bit is set to enable the watchdog timer for IDE0. This bit is cleared to disable the watchdog timer.
- **Bit [12]**: Watchdog Timeout (R/W1C) – IDE0 Watchdog Timer Timeout. This bit set indicates that the watchdog timer for IDE0 timed out. When enabled, and IORDY monitoring bit is also enabled, during IDE0 PIO operation, the watchdog counter starts counting when IORDY signal is de-asserted. If after 256 PCI clocks, the IORDY signal is still de-asserted, the Watchdog Timer is expires, and this bit is set and the SiI3512 controller continue its operation and stop monitoring IORDY signal. Software writes one to clear this bit. Once this bit is cleared, the SiI3512 controller starts monitoring IORDY on Channel 0 again.
- **Bit [11]**: Interrupt Status (R) – IDE0 Interrupt Status. This bit set indicates that an interrupt is pending on IDE0. This bit provides real-time status of the IDE0 interrupt pin.
- **Bit [10]**: Virtual DMA Int (R) – IDE0 Virtual DMA Completion Interrupt. This bit set indicates that the Virtual DMA data transfer has completed. This bit is cleared when bit[0] PBM enable in PCI Bus Master – IDE0 is cleared.
- **Bit [09:03]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [02]**: Channel Rst (R/W) – IDE0 Channel Reset. When this bit is set, IDE Channel #0 RST signal is asserted.
- **Bit [01]**: Buffered Cmd (R) – IDE0 Buffered Command Active. This bit set indicates that a Buffered Command is currently active. This bit is set when the first command byte is written to the command buffer. This bit is cleared when all of the task file bytes, including the command byte, have been written to the device.
- **Bit [00]**: Reserved (R) – This bit is reserved and returns one on a read.

Data Transfer Mode – IDE0

Address Offset: 0xB4
Access Type: Read/Write
Reset Value: 0x0000_0022

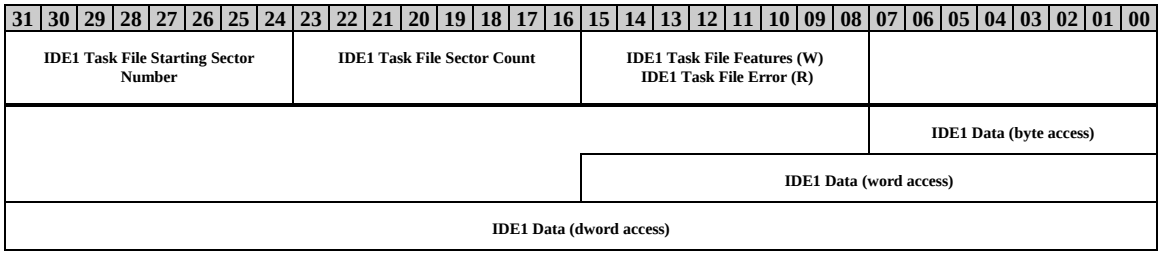
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Reserved		IDE0 Device 1 Transfer Mode		Reserved		IDE0 Device 0 Transfer Mode	

This register defines the transfer mode register for IDE Channel #0 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07:06]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [05:04]**: Device 1 Transfer Mode (R/W) – IDE0 Device 1 Data Transfer Mode. This bit field is used to set the data transfer mode on IDE side during PCI DMA transfer: 0b00 or 0b01 = PIO transfer; 0b10 or 0b11 = DMA transfer.
- **Bit [03:02]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [01:00]**: Device 0 Transfer Mode (R/W) – IDE0 Device 0 Data Transfer Mode. This bit field is used to set the data transfer mode on IDE side during PCI DMA transfer: 0b00 or 0b01 = PIO transfer; 0b10 or 0b11 = DMA transfer.

IDE1 Task File Register 0

Address Offset: 0xC0
Access Type: Read/Write
Reset Value: 0x0000_0000

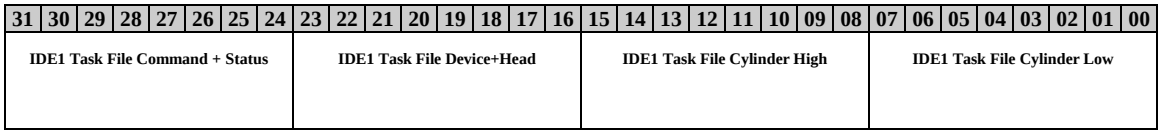


This register contains some of the IDE Channel #1 Task File registers and provides access to the IDE data bus. Access to this register is determined by the PCI bus Byte Enables at the time of the read or write operation, i.e., what is accessed is determined by the address and by the size of the access. The register bits are defined below.

- **Bit [31:00]**: IDE1 Data (R/W). This bit field provides access to the IDE1 Data. This register can be accessed as an 8-bit, 16-bit, or 32-bit word.
- **Bit [31:24]**: IDE1 Task File Starting Sector Number (R/W). This bit field defines the IDE1 Task File Starting Sector Number register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [23:16]**: IDE1 Task File Sector Count (R/W). This bit field defines the IDE1 Task File Sector Count register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [15:08]**: IDE1 Task File Features (W). This write-only bit field defines the IDE1 Task File Features register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.
- **Bit [15:08]**: IDE1 Task File Error (R) – This read-only bit field defines the IDE1 Task File Error register. Access to this bit field is permitted if the PCI bus Byte Enable is active for this byte only.

IDE1 Task File Register 1

Address Offset: 0xC4
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines one of the IDE Channel #1 Task File registers in the SiI3512 controller. Access to these bit fields is permitted if the PCI bus Byte Enable is active for one byte only. The register bits are defined below.

- **Bit [31:24]**: IDE1 Task File Command (W). This write-only bit field defines the IDE1 Task File Command register.
- **Bit [31:24]**: IDE1 Task File Status (R) – This read-only bit field defines the IDE1 Task File Status register.
- **Bit [23:16]**: IDE1 Task File Device+Head (R/W). This bit field defines the IDE1 Task File Device and Head register.
- **Bit [15:08]**: IDE1 Task File Cylinder High (R/W). This bit field defines the IDE1 Task File Cylinder High register.
- **Bit [07:00]**: IDE1 Task File Cylinder Low (R/W). This bit field defines the IDE1 Task File Cylinder Low register.

IDE1 Task File Register 2

Address Offset: 0xC8
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved								IDE1 Task File Device Control IDE1 Task File Auxiliary Status								Reserved								Reserved							

This register defines one of the IDE Channel #1 Task File registers in the SiI3512 controller. Access to these bit fields is permitted if the PCI bus Byte Enable is active for one byte only. The register bits are defined below.

- **Bit [31:24]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [23:16]**: IDE1 Task File Device Control (W). This bit field defines the IDE1 Task File Device Control register.
- **Bit [23:16]**: IDE1 Task File Auxiliary Status (R) – This bit field defines the IDE1 Task File Auxiliary Status register.
- **Bit [15:00]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

IDE1 Read/Write Ahead Data

Address Offset: 0xCC
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE1 Read Ahead Data																															

This register defines the read ahead data port for PIO transfers on IDE Channel #1 in the SiI3512 controller.

This register can be accessed as an 8-bit, 16-bit, or 32-bit word, depending upon the PCI bus byte enables. The data written to this register must be zero-aligned.

IDE1 Task File Register 0 – Command Buffering

Address Offset: 0xD0
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE1 Task File Starting Sector Number								IDE1 Task File Sector Count								IDE1 Task File Features								Unused							

This register defines one of the IDE Channel #1 Task File registers used for Command Buffered accesses in the SiI3512 controller. The register bits are defined below.

- **Bit [31:24]**: IDE1 Task File Starting Sector Number (R/W). This bit field defines the IDE1 Task File Starting Sector Number register. Access to this bit field is permitted only if the PCI bus Byte Enable for byte 3 is active.
- **Bit [23:16]**: IDE1 Task File Sector Count (R/W). This bit field defines the IDE1 Task File Sector Count register. Access to this bit field is permitted only if the PCI bus Byte Enable for byte 2 is active.
- **Bit [15:08]**: IDE1 Task File Features (W). This write-only bit field defines the IDE1 Task File Features register. Access to this bit field is permitted only if the PCI bus Byte Enable for byte 1 is active.

IDE1 Task File Register 1 – Command Buffering

Address Offset: 0xD4
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE1 Task File Command								IDE1 Task File Device+Head								IDE1 Task File Cylinder High								IDE1 Task File Cylinder Low							

This register defines one of the IDE Channel #1 Task File registers used for Command Buffered accesses in the SiI3512 device. The register bits are defined below.

- **Bit [31:24]**: IDE1 Task File Command (W). This write-only bit field defines the IDE1 Task File Command register.
- **Bit [23:16]**: IDE1 Task File Device+Head (R/W). This bit field defines the IDE1 Task File Device and Head register.
- **Bit [15:08]**: IDE1 Task File Cylinder High (R/W). This bit field defines the IDE1 Task File Cylinder High register.
- **Bit [07:00]**: IDE1 Task File Cylinder Low (R/W). This bit field defines the IDE1 Task File Cylinder Low register.

IDE1 Extended Task File Register – Command Buffering

Address Offset: 0xD8
Access Type: Read/Write
Reset Value: 0x0000_0000

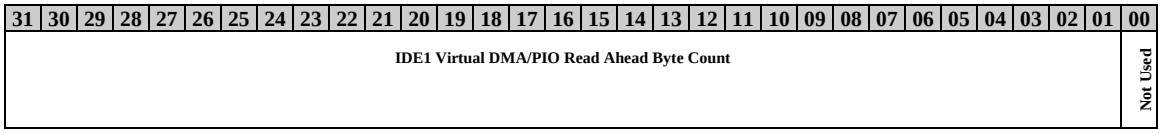
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IDE1 Task File Cylinder High Ext								IDE1 Task File Cylinder Low Ext								IDE1 Task File Start Sector Ext								IDE1 Task File Sector Count Ext							

This register defines one of the IDE Channel #1 Task File registers used for Command Buffered accesses in the SiI3512 device. The register bits are defined below. If this register is written, the IDE1 Task File Device+Head byte of the IDE1 Task File Register 1 – Command Buffering register must not be written.

- **Bit [31:24]**: IDE1 Task File Cylinder High Ext(R/W). This write-only bit field defines the IDE1 Task File Extended Cylinder High register.
- **Bit [23:16]**: IDE1 Task File Cylinder Low Ext (R/W). This bit field defines the IDE1 Task File Extended Cylinder Low register.
- **Bit [15:08]**: IDE1 Task File Start Sector Ext (R/W). This bit field defines the IDE1 Task File Extended Start Sector register.
- **Bit [07:00]**: IDE1 Task File Sector Count Ext (R/W). This bit field defines the IDE1 Task File Extended Sector Count register.

IDE1 Virtual DMA/PIO Read Ahead Byte Count

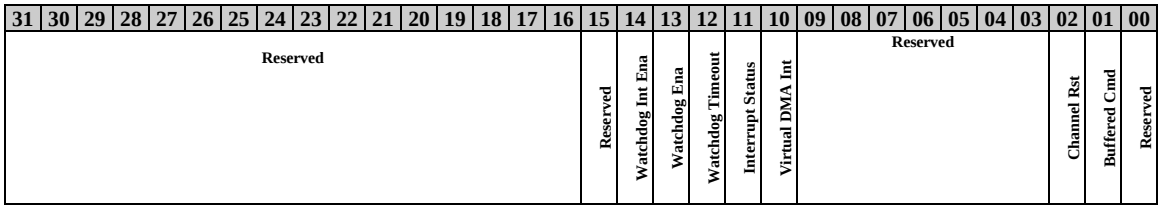
Address Offset: 0xDC
Access Type: Read/Write
Reset Value: 0x0000_0000



This register defines the read ahead byte count register for Virtual DMA and PIO Read Ahead transfers on IDE Channel #1 in the SiI3512 controller. In Virtual DMA mode (PCI bus master DMA with PIO transfers on the IDE), all 32 bits are used as the word-aligned byte count. In PIO Read Ahead mode, only the lower 16 bits are used as the word-aligned byte count.

IDE1 Task File Configuration + Status

Address Offset: 0xE0
Access Type: Read/Write
Reset Value: 0x6515_0101



This register defines the task file configuration and status register for IDE Channel #1 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:16]**: Reserved (R) – This bit field is reserved and defaults to 0x6515.
- **Bit [15]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [14]**: Watchdog Int Ena (R/W) – IDE1 Watchdog Interrupt Enable. This bit is set to enable Interrupt when Watchdog timer expired.
- **Bit [13]**: Watchdog Ena (R/W) – IDE1 Watchdog Timer Enable. This bit is set to enable the watchdog timer for IDE1. This bit is cleared to disable the watchdog timer.
- **Bit [12]**: Watchdog Timeout (R) – IDE1 Watchdog Timer Timeout. This bit set indicates that the watchdog timer for IDE1 timed out. When enabled, and IORDY monitoring bit is also enabled, during IDE0 PIO operation, the watchdog counter starts counting when IORDY signal is de-asserted. If after 256 PCI clocks cycles, the IORDY signal is still de-asserted, the Watchdog Timer expires, and this bit is set and the SiI3512 controller continues its operation and stops monitoring the IORDY signal. Software writes one to clear this bit. Once this bit is cleared, the SiI3512 controller starts monitoring IORDY on Channel 1 again.
- **Bit [11]**: Interrupt Status (R) – IDE1 Interrupt Status. This bit set indicates that an interrupt is pending on IDE1. This bit provides real-time status of the IDE1 interrupt pin.
- **Bit [10]**: Virtual DMA Int (R) – IDE1 Virtual DMA Completion Interrupt. This bit set indicates that the Virtual DMA data transfer has completed. This bit is cleared when bit[0] PBM enable in PCI Bus Master – IDE1 is cleared.
- **Bit [09:03]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [02]**: Channel Rst (R/W) – IDE1 Channel Reset. When this bit is set, IDE Channel # 1 RST signal is asserted.
- **Bit [01]**: Buffered Cmd (R) – IDE1 Buffered Command Active. This bit set indicates that a Buffered Command is currently active. This bit is set when the first command byte is written to the command buffer. This bit is cleared when all of the task file bytes, including the command byte, have been written to the device.
- **Bit [00]**: Reserved (R) – This bit field is reserved and returns one on a read.

Data Transfer Mode – IDE1

Address Offset: 0xF4
Access Type: Read/Write
Reset Value: 0x0000_0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																								Reserved		IDE1 Device 1 Transfer Mode		Reserved		IDE1 Device 0 Transfer Mode	

This register defines the transfer mode register for IDE Channel #1 in the SiI3512 controller. The register bits are defined below.

- **Bit [31:08]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [07:06]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [05:04]**: Device 1 Transfer Mode (R/W) – IDE1 Device 1 Data Transfer Mode. This bit field is used to set the data transfer mode on IDE side during PCI DMA transfer: 0b00 or 0b01 = PIO transfer; 0b10 or 0b11 = DMA transfer.
- **Bit [03:02]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [01:00]**: Device 0 Transfer Mode (R/W) – IDE0 Device 0 Data Transfer Mode. This bit field is used to set the data transfer mode on IDE side during PCI DMA transfer: 0b00 or 0b01 = PIO transfer; 0b10 or 0b11 = DMA transfer.

Serial ATA SControl

Address Offset: 0x100 (Channel 0) / 0x180 (Channel 1)
Access Type: Read/Write
Reset Value: 0x0000_0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved												PMP				Reserved				IPM				SPD				DET			

This register is the SControl register as defined by the Serial ATA specification.

- **Bit [31:20]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [19:16]**: PMP - This field is the 4-bit value to be placed in the Port Multiplier Port field of all transmitted FISes.
- **Bit [15:12]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [11:08]**: IPM – This field identifies the interface power management states that may be invoked via the Serial ATA interface power management capabilities.

Value	Definition
0000	No interface power management restrictions (Partial and Slumber modes enabled)
0001	Transitions to the Partial power management state are disabled
0010	Transitions to the Slumber power management state are disabled
0011	Transitions to both the Partial and Slumber power management states are disabled
others	Reserved

- **Bit [07:04]**: SPD – This field identifies the highest allowed communication speed the interface is allowed to negotiate.

Value	Definition
0000	No restrictions
0001	Limit to Generation 1 (1.5 Gb/s) (default value)
others	Reserved

- **Bit [03:00]**: DET – This field controls host adapter device detection and interface initialization.

Value	Action
0000	No action
0001	ATA Reset is generated until another value is written to the field
0100	No action
others	Reserved, no action

Serial ATA SStatus

Address Offset: 0x104 (Channel 0) / 0x184 (Channel 1)
Access Type: Read
Reset Value: 0x0000_00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved																				IPM				SPD				DET			

This register is the SStatus register as defined by the Serial ATA specification.

- **Bit [31:12]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [11:08]**: IPM – This field identifies the current interface power management state.

Value	Definition
0000	Device not present or communication not established
0001	Interface in active state
0010	Interface in Partial power management state
0110	Interface in Slumber power management state
others	Reserved

- **Bit [07:04]**: SPD – This field identifies the negotiated interface communication speed.

Value	Definition
0000	No negotiated speed
0001	Generation 1 communication rate (1.5 Gb/s)
others	Reserved

- **Bit [03:00]**: DET – This field indicates the interface device detection and PHY state.

Value	Action
0000	No device detected and PHY communication not established
0001	Device presence detected but PHY communication not established
0011	Device presence detected and PHY communication established
0100	PHY in offline mode as a result of the interface being disabled or running in a BIST loopback mode
others	Reserved, no action

Until a device is detected (IPM and DET fields become nonzero), the SiI3512 controller issues a COMRESET every 100 milliseconds.

Serial ATA SError

Address Offset: 0x108 (Channel 0) / 0x188 (Channel 1)
Access Type: Read/Clear
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
R	R	R	R	R	R	**F**	**T**	**S**	**H**	**C**	**D**	**B**	**W**	**I**	**N**	R	R	R	R	**E**	**P**	**C**	**T**	R	R	R	R	R	R	**M**	**I**
DIAG																**ERR**															

This register is the SError register as defined by the Serial ATA specification.

- **Bit [31:16]**: DIAG – This field contains bits defined as shown in the following table. Writing a 1 to the register bit clears the B, C, F, N, H, and W bits.

Table 25. SError Register Bits (DIAG Field)

Bit	Definition	Description
B	10b to 8b decode error	Latched decode error or disparity error from the Serial ATA PHY
C	CRC error	Latched CRC error from the Serial ATA PHY
D	Disparity error	N/A, always 0; this error condition is combined with the decode error and reported as B error
F	Unrecognized FIS type	Latched Unrecognized FIS error from the Serial ATA Link
I	PHY Internal error	N/A, always 0
N	PHYRDY change	Indicates a change in the status of the Serial ATA PHY
H	Handshake error	Latched Handshake error from the Serial ATA PHY
R	Reserved	Always 0
S	Link Sequence error	N/A, always 0
T	Transport state transition error	N/A, always 0
W	ComWake	Latched ComWake status from the Serial ATA PHY

- **Bit [15:00]**: ERR – This field contains bits defined as shown in the following table. The ERR Field is not implemented; all bits are always 0.

Table 26. SError Register Bits (ERR Field)

Bit	Definition	Description
C	Non-recovered persistent Communication error or data integrity error	N/A, always 0
E	Internal Error	N/A, always 0
I	Recovered data Integrity error	N/A, always 0
M	Recovered communications error	N/A, always 0
P	Protocol error	N/A, always 0
R	Reserved	Always 0
T	Non-recovered Transient data integrity error	N/A, always 0

SERIAL ATA SActive

Address Offset: 0x10C (Channel 0) / 0x18C (Channel 1)
Access Type: Read/Write 1/Clear
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SActive bits																															

The bits of this register may be written with a 1, but are cleared if the corresponding bits of the second dword of a FIS are set when the SDevice Bits FIS is received. All 32 bits may be cleared by writing 0x0000_0000 to the register; individual bits may not be cleared except by the hardware.

SMisc

Address Offset: 0x140 (Channel 0) / 0x1C0 (Channel 1)
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIS_Done	Transmit_FIS	Transmit_OK	IFIS_OK	IntrlckFIS	Reject_IF	Accept_IF	Rx_IFIS	SDB	pterr	Scr_dis	Cont_dis	VS_Lock_Abort	fpdmawr	dmainen	dmaouten	Transmit_BIST	devdrvn	nienfis_dis	Reserved	ComWake	pm_fiscfg		pm_locken	reffismode	PMCHG	PMMODE		Reserved		PMREQ	

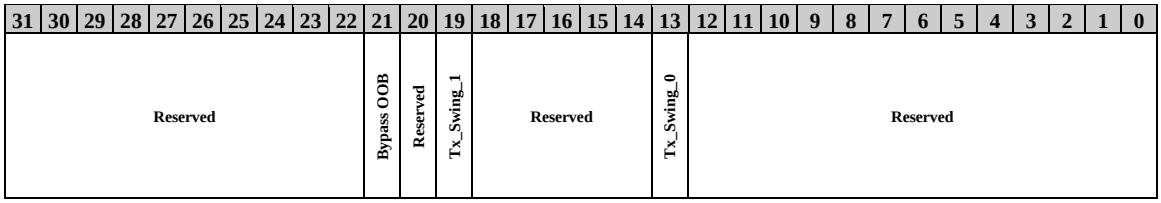
This register contains bits for controlling Serial ATA power management, ComWake, loopback modes, and FIS transfers.

- **Bit [31]**: FIS_Done (R/W) – This bit is used to indicate to the link logic that all the data for the Transparent FIS has been transferred and that the link can proceed to close out the FIS. This is used in Transparent FIS transmission. Please refer to the FIS Support section on page 89 for more details.
- **Bit [30]**: Transmit_FIS (W) – This bit is used to signal the link logic to start the process of transmitting a Transparent FIS. Please refer to the FIS Support section on page 89 for more details.
- **Bit [29]**: Transmit_OK (R) – This bit is used in Transparent FIS transmission. It is used by the link to signal to the host that the current Transparent FIS has been successfully transferred to the device, and that R_OK has been received.
- **Bit [28]**: IFIS_OK (R) – This bit is used in the reception of Interlocked FISes. This bit is set by the link logic to inform the host that the current Interlocked FIS has been successfully received with no errors.
- **Bit [27]**: IntrlckFIS (R) – This bit is set to indicate to the host driver that the link has detected an the arrival of an interlocked FIS and that the host should set up the DMA engine to start transfer of data
- **Bit [26]**: Reject_IFIS (W) – This bit is set by the host driver to indicate to the link that the current Interlocked FIS should be rejected. The link logic will respond to the device with an R_ERR when the complete FIS has been received.
- **Bit [25]**: Accept_IFIS (W) – This bit is set by the host driver to indicate to the link that the current interlocked FIS should be accepted. The link logic will respond to the device with R_OK
- **Bit [24]**: Rx_IFIS (W) – This bit is set by the host driver to inform the link/transport logic that the host has set up the DMA engine to transfer the incoming Interlocked FIS and that the DMA cycles can begin
- **Bit [23]**: SDB (R) – This bit indicates that a Set Device Bits FIS has been received
- **Bit [22]**: pterr (R) – This bit indicates that a Protocol Error has occurred. An interrupt will be generated if bit 20 of SIEN is set.
- **Bit [21]**: Scr_dis (R/W) – This bit disables the scrambling of data on the serial ATA bus. This is used only for debugging purposes and should not be changed by the user
- **Bit [20]**: Cont_dis (R/W) – Setting this bit disables the CONT primitive, i.e., the SiI3512 controller will always send the actual primitive instead of a CONT followed by random data.

- **Bit [19]**: VS_Lock_Abort (R/W) – This bit controls the changes to the entries in the Command Protocol Table upon receiving a VS_Lock command. If this bit is set, all Command Protocol Table will be cleared. If this bit is not set, the Command Protocol Table will not be cleared in the VS_Lock state.
- **Bit [18]**: fpdmawr (W)– Setting this bit initiates a DMA write transfer
- **Bit [17]**: dmainen(R/W) – This bit enables Read DMA operations for First Party DMA or transparent FIS operation.
- **Bit [16]**: dmaouten (R/W) – This bit enables Write DMA operations for First Party DMA or transparent FIS operation.
- **Bit [15]**: Reserved (R/W). This bit is reserved and returns zero on a read. Always write 0 to these bits.
- **Bit [14]**: devdrvn (R/W) – This bit enables the protocol to be solely determined by FISes from the device.
- **Bit [13]**: nienfis_dis (R/W) – If this bit is set, a Control Register FIS will not be sent in response to a change in nIEN.
- **Bit [12]**: Reserved (W). Always write 0 to these bits.
- **Bit [11]**: ComWake/Clear_BSY (R/W) – When the Serial ATA interface is in PARTIAL or SLUMBER mode, setting this bit (to 1) asserts ComWake on the Serial ATA bus. When the Serial ATA interface is ON and an interlocked FIS is received, setting this bit (to 1) clears BSY in the ATA Status.
- **Bit [10:09]**: pm_fiscfg[1:0] (R/W) – Configuration for interpreting FISes with a different Port Multiplier port number from that specified in SControl.
- **Bit [08]**: pm_locken (R/W) – If set, no SYNC is sent after a DMA Activate FIS, a PIO Setup FIS for PIO Out, or an interlocked FIS when dmaouten (bit 16) is set.
- **Bit [07]**: regfismode (R/W) – If set, received Register FIS will not be used to update task file if BSY = DRQ = 0.
- **Bit [06]**: PMCHG (R/W1C) – This bit reports a change in the Power Management mode. This bit corresponds to the interrupt enabled by bit 26 of SIEN. This bit is cleared by writing a 1.
- **Bit [05:04]**: PMMODE (R) – These bits report the power management mode status: bit 5 corresponds to Slumber mode; bit 4 to Partial mode. A transition on either of these bits causes a Power Management mode change interrupt.
- **Bit [03:02]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [01:00]**: PMREQ (W) – These bits initiate power management requests: setting bit 1 will send a Slumber mode request to the device; setting bit 0 will send a Partial mode request to the device.

Serial ATA PHY Configuration

Address Offset: 0x144
Access Type: Read/Write
Reset Value: 0x8000_80B1



The PHY Configuration register is auto-initialized from external Flash or EEPROM. The bit definitions are as follows:

- **Bit[31:22]**: Reserved. The values of these bits should not be changed from their defaults otherwise erratic operation may result
- **Bit[21]**: Bypass OOB sequence. If the bit set to 1, all channel Tx outputs random pattern data.
- **Bit[20]**: Reserved. The value of this bits should not be changed from their defaults otherwise erratic operation may result
- **Bit[19]**: Tx_Swing_1: This bit, together with Tx_Swing_0, sets the nominal output amplitude for the Transmitter
- **Bit[18:14]**: Reserved. The values of these bits should not be changed from their defaults otherwise erratic operation may result
- **Bit[13]**: Tx_Swing_0: This bit, together with Tx_Swing_1, sets the nominal output swing for the Transmitter. The available combinations are as follows:

Tx_Swing_1	Tx_Swing_0	Nominal Output Swing
0	0	500 mV
0	1	600 mV
1	0	700 mV
1	1	800 mV

- **Bit[12:0]**: Reserved. The values of these bits should not be changed from their defaults otherwise erratic operation may result.

SIEN

Address Offset: 0x148 (Channel 0) / 0x1C8 (Channel 1)
Access Type: Read/Write
Reset Value: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved		Transmit_OK	IFIS_OK	IntrlckFIS	PMCHG	F	Reserved	SDB	H	C	Reserved	B	W	Reserved	N	Reserved															

This register contains bits for enabling interrupts.

- **Bit [31:30]**: Reserved (R) – This bit field is reserved and returns zeros on a read.
- **Bit [29]**: Transmit_OK – This bit enables an interrupt upon the assertion of the Transmit_OK bit in the SMisc register.
- **Bit [28]**: IFIS_OK – This bit enables an interrupt upon the assertion of the IFIS_OK bit in the SMisc register.
- **Bit [27]**: IntrlckFIS – This bit enables an interrupt upon the assertion of the IntrlckFIS bit in the SMisc register.
- **Bit [26]**: PMCHG – This bit enables an interrupt upon a Power Management Mode change. The interrupt is reported in bit 6 of SMisc.
- **Bit [25]**: F – This bit enables an interrupt upon the assertion of the F bit in the DIAG field of the SError register.
- **Bit [24]**: Reserved (R) – This bit is reserved and returns zero on a read.
- **Bit [23]**: SDB – This bit enables an interrupt upon the assertion of the SDB bit in the SMisc register.
- **Bit [22]**: H – This bit enables an interrupt upon the assertion of the H bit in the DIAG field of the SError register.
- **Bit [21]**: C – This bit enables an interrupt upon the assertion of the C bit in the DIAG field of the SError register.
- **Bit [20]**: Reserved (R) – This bit is reserved and returns zero on a read.
- **Bit [19]**: B – This bit enables an interrupt upon the assertion of the B bit in the DIAG field of the SError register.
- **Bit [18]**: W – This bit enables an interrupt upon the assertion of the W bit in the DIAG field of the SError register.
- **Bit [17]**: Reserved (R) – This bit is reserved and returns zeros on a read.
- **Bit [16]**: N – This bit enables an interrupt upon the assertion of the N bit in the DIAG field of the SError register.
- **Bit [15:00]**: Reserved (R) – This bit field is reserved and returns zeros on a read.

SFISCfg

Address Offset: 0x14C (Channel 0) / 0x1CC (Channel 1)
Access Type: Read/Write
Reset Value: 0x1040_1555

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Reserved		FIS27cfg		FIS34cfg		FIS39cfg		FIS41cfg		FIS46cfg		FIS58cfg		FIS5Fcfg		FISA1cfg		FISA6cfg		FISB8cfg		FISBFcfg		FISC7cfg		FISD4cfg		FISD9cfg		FISOcfg	

This register contains bits for controlling Serial ATA FIS reception. See the FIS Support section on page 89 for an explanation of the configuration bits.

To work around a bug that would cause the 3512 to reject a DMA activate FIS under certain circumstances, the SFISCfg register should be set to a value of 0x1040_1554.

RxFIS0-RxFIS6

Address Offset: 0x160–0x178 (Channel 0) / 0x1E0–0x1F8 (Channel 1)
Access Type: Read
Reset Value: 0x????_????

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FIS Dword																															

These registers contain 7 dwords from a Serial ATA FIS reception.

Programming Sequences

Recommended Initialization Sequence for the SiI3512

The recommended initialization sequence for the SiI3512 device is detailed below.

Initialize PCI Configuration Space registers:

- Initialize Base Address Register 0 with the address of an 8-byte range in I/O space.
- Initialize Base Address Register 1 with the address of a 4-byte range in I/O space.
- Initialize Base Address Register 2 with the address of an 8-byte range in I/O space.
- Initialize Base Address Register 3 with the address of a 4-byte range in I/O space.
- Initialize Base Address Register 4 with the address of a 16-byte range in I/O space.
- Initialize Base Address Register 5 with the address of a 512-byte range in memory space.
- To enable the bios expansion ROM, initialize the Expansion ROM Base Address register with the address of a 512-kbyte range in memory space.
- Enable I/O space access, memory space access, and bus master operation by setting bits [2:0] of the PCI Command register.

NOTE: The preceding configuration space register initialization is normally done by the motherboard BIOS in PC type systems.

NOTE: To work around a bug that would cause the 3512 controller to reject a DMA activate FIS under certain circumstances, the default value of SFISCfg register need to be modified. The bits [01:00] in the SFISCfg register, Base Address Register 5 offset 0x14C for Channel 0 and offset 0x1CC for Channel 1, need to be set to 0 during initialization.

If the PCI-IDE arbiter's default FIFO read/write request thresholds are not suitable for the application they may be changed via the FIFO Valid Byte Count and Control IDEx register. The read threshold is defined by bits [02:00], and the

write threshold is defined by bits [10:08] in the FIFO Valid Byte Count and Control – IDEx register. In most environments, setting these bit fields to zero results in the best utilization of the PCI bus by the SiI3512 controller.

If interrupt driven operation is **not** desired, set bits [23:22] of the System Configuration Status and Command register to block IDE interrupts from reaching the PCI bus.

Serial ATA Device Initialization

This section provides a general overview of the steps necessary to initialize a Serial ATA device before it can be used for read/write operations.

Select the Serial ATA device. The device is selected by programming bits [23:16] in the IDEx Task File Register 1 register.

If interrupt driven operation is desired, ensure that IDE interrupts are enabled by writing 0 to bits [23:16] of the IDEx Task File Register 2 register.

For ATA devices only:

Issue the Initialize Device Parameters command by

- Programming bits [23:16] in the IDEx Task File 0 register with the number of logical sectors per logical track.
- Programming bits [23:16] in the IDEx Task File 1 register with the maximum head number.
- Programming bits [31:24] in the IDEx Task File Register 1 register with the value = 0x91.
- Wait for the command to complete. This can be accomplished by waiting for an interrupt if interrupts have been enabled at both the controller and the device. If interrupts are not enabled, command completion can be detected by polling bits [31:24] of the IDEx Task File Register 1 register until the BUSY bit is no longer asserted.

If device supports read/write multiple commands, issue the Set Multiple Mode command by:

- Programming bits [23:16] in the IDEx Task File 0 register with the number of sectors per block to use on the following Read/Write Multiple commands.
- Programming bits [31:24] in the IDEx Task File Register 1 register with the value = 0xC6.
- Wait for the command to complete (see above).

For both ATA and ATAPI devices:

Set device transfer mode by:

- Programming bits [15:08] in the IDEx Task File 0 register with the value 0x03 to “Set the transfer mode based on value in Sector Count Register”.
- Programming bits [23:16] in the IDEx Task File 0 register to the desired transfer mode. The settings are defined below:
  - 0x08 = PIO Mode 0
  - 0x09 = PIO Mode 1
  - 0x0A = PIO Mode 2
  - 0x0B = PIO Mode 3
  - 0x0C = PIO Mode 4
  - 0x20 = Multiword DMA Mode 0
  - 0x21 = Multiword DMA Mode 1
  - 0x22 = Multiword DMA Mode 2
  - 0x40 = Ultra DMA Mode 0
  - 0x41 = Ultra DMA Mode 1
  - 0x42 = Ultra DMA Mode 2
  - 0x43 = Ultra DMA Mode 3
  - 0x44 = Ultra DMA Mode 4
  - 0x45 = Ultra DMA Mode 5
  - 0x46 = Ultra DMA Mode 6
- Programming bits [31:24] in the IDEx Task File Register 1 register with the value = 0xEF.
- Wait for the command to complete (see above).

In order to use the controller's DMA capability to perform the data transfer for an ATA/ATAPI command, the controller needs to be configured for the transfer mode to use when transferring data to or from the ATA bus. The data transfer mode is set by programming bits [1:0] of the IDEx Data Transfer Mode register. The transfer mode select values are listed below:

0b00 = PIO/Virtual DMA Mode
0b10 = DMA Mode

Issue ATA Command

The following describes the sequence to issue a read/write type command to an ATA device.

Select the IDE device. The IDE device is selected by programming bits [23:16] in the IDEx Task File Register 1 register.

Set the number of sectors to be transferred by programming bits [23:16] of the IDEx Task File Register 0 register.

Set the location of data to be transferred. The location is defined by programming the following.

Bits [31:24] in the IDEx Task File Register 0 register define the Starting Sector.
Bits [23:16] in the IDEx Task File Register 1 register define the Device and Head value.
Bits [15:08] in the IDEx Task File Register 1 register define the Cylinder High value.
Bits [07:00] in the IDEx Task File Register 1 register define the Cylinder Low value.

Issue the Read/Write PIO/DMA command by programming bits [31:24] in the IDEx Task File Register 1 register with the command desired.

IDE PIO Mode Read/Write Operation

Once the SiI3512 controller is initialized via the initialization sequence described above, the ATA device has been initialized for PIO mode data transfer per the guidelines in Serial ATA Device Initialization section on page 79, and the controller channel has been initialized for PIO mode data transfer, PIO read/write operations may be performed by following the programming sequence described below.

Issue a PIO Read/Write command to device following the steps in the Issue ATA Command section on page 80.

Read Operation

Wait until an IDE channel interrupt (bit 11 in the IDEx Task File Timing + Configuration + Status register is set).

Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

If no error, continue to read IDE data via the IDEx Task File Register 0 register, until the expected number of sectors of data per interrupt are read.

Repeat the above three steps until all data for the read command has been transferred or an error has been detected.

Write Operation

Wait until bit 27(DRQ) in the IDEx Task File Register 1 register is set.

Continue to write IDE data via the IDEx Task File Register 0 register until the expected number of sectors of data per interrupt are written.

Wait until an IDE channel interrupt (bit 11 in the IDEx Task File Timing + Configuration + Status register is set).

Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

If no error, repeat the previous four steps until all data for the write command has been transferred or an error has been detected.

Watchdog Timer Operation

The purpose of the watchdog timer is to prevent the host system from hanging because a device operating in PIO mode stopped responding to task file accesses. If, during a task file access by the host, the device negates IORDY and then stops responding, the host will hang waiting for the access to complete. It is this type of hang, that the watchdog timer is designed to protect against.

The watchdog timer monitors the length of time the IORDY signal is negated. If the watchdog timer detects that the IORDY signal has remained negated longer than the watchdog timeout period (approximately 1000 PCI clocks), the watchdog timer will force the task file access cycle to complete, and set the watchdog timeout bit in the IDEx Task File Timing + Configuration + Status register. The data associated with a timed out access should be considered invalid. Additionally, the watchdog timer can be configured to generate an interrupt when a timeout is detected by setting bit 14 of the IDEx Task File Timing + Configuration + Status register.

The watchdog timer feature is disabled by default.

In addition to the controller channel initialization specified previously, add the following two steps to enable the watchdog timer:

1. Enable the watchdog timer by setting bit 13 of the IDEx Task File Timing + Config + Status register.
2. If an interrupt is desired whenever the watchdog times out, enable the watchdog interrupt by setting bit 14 of the IDEx Task File Timing + Config + Status register.

The following programming sequence is needed for each PIO Mode Read/Write Operation with the watchdog timer enabled:

- Issue a Read/Write PIO Command to the ATA drive following the steps listed in the Issue ATA Command section on page 80.

  Read Operation

  Wait for an IDE channel interrupt.

  If controller interrupts are disabled, poll for the IDE interrupt by reading the IDEx Task File Timing + Configuration + Status register. If bit 12 is set, a watchdog timeout has occurred. If bit 11 is set, the ATA device is interrupting.

  If the watchdog timeout bit is set,

  Write ‘1’ to bit 12 in the IDEx Task File Timing + Configuration + Status register to clear watchdog timeout status.

  The watchdog timeout represents a fatal error as far as the current ATA command is concerned. A course of action that might be appropriate at this point might be to reset and reinitialize the ATA channel and then retrying the command that failed.

  If the ATA device interrupt bit is set,

  Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

  Write ‘1’ to bit 18 of the PCI Bus Master – IDEx Register to clear the ATA interrupt.

  If the ATA device is not reporting an error, continue to read IDE data via the IDEx Task File Register 0 register, until the expected number of sectors of data per interrupt are read.

  Repeat the read operation steps until all data for the read command has been transferred or an error has been detected.

  Write Operation

  Wait until bit 27(DRQ) in the IDEx Task File Register 1 register is set.

  Continue to write IDE data via the IDEx Task File Register 0 register until the expected number of sectors of data per interrupt are written.

  Wait for an IDE channel interrupt.

If controller interrupts are disabled, poll for the IDE interrupt by reading the IDEx Task File Timing + Configuration + Status register. If bit 12 is set, a watchdog timeout has occurred. If bit 11 is set, the ATA device is interrupting.

If the watchdog timeout bit is set,

Write '1' to bit 12 in the IDEx Task File Timing + Configuration + Status register to clear watchdog timeout status.

The watchdog timeout represents a fatal error as far as the current ATA command is concerned. A course of action that might be appropriate at this point might be to reset and reinitialize the ATA channel and then retrying the command that failed.

If the ATA device interrupt bit is set,

Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

Write '1' to bit 18 of the PCI Bus Master – IDEx Register to clear the ATA interrupt.

If no error, repeat the write operation steps until all data for the write command has been transferred or an error has been detected.

IDE PIO Mode Read Ahead Operation

The Read ahead operation allows the controller to "pre-fetch" data from the IDE bus and store it in the controller's channel FIFO, where it will later be retrieved by the host. This mode of operation has the potential to speed-up PIO data transfers by not forcing the host to wait the programmed PIO cycle time for every access to the task file data register. The amount of any speed increase will depend on the PIO mode in use, the characteristics of the host PCI bus, as well as the speed of the host processor.

To use the controller's PIO read ahead capability, make the following changes to the "Read Operation" portion of the previous sections.

Just prior to retrieving the read data, set the read ahead byte count by programming bits [15:00] in the IDEx Virtual DMA/PIO Read Ahead Byte Count register with the exact number of bytes to be read for the interrupt.

Instead of reading the IDEx Task File Register 0 register to retrieve the data, read the IDEx Read Ahead Data register instead.

IDE MDMA/UDMA Read/Write Operation

After the SiI3512 controller is initialized with the initialization sequence described earlier and the ATA device has been initialized for MDMA/UDMA mode data transfer, DMA read/write operations may be performed by following the programming sequence described below.

Issue a DMA read/write command to the device.

Program Bus Master Registers

Clear bit 17 in the PCI Bus Master – IDEx register. This bit is set if an error occurred during the previous DMA access.

Clear bit 18 in the PCI Bus Master – IDEx register. This bit is set if an IDE interrupt occurred during the previous DMA access.

Create a Physical Region Descriptor (PRD) Table

A PRD table is an array where each entry describes the location and size of a physical memory buffer that will be used during the DMA operation. Each PRD table entry is 64-bits in length, formatted as follows; bits [31:0] contain the 32-bit starting address of the memory buffer, bits [47:32] contain the 16-bit size of the memory buffer, bits [62:48] are normally unused, bit 63 flags the end of the PRD table and therefore should only be set in the last entry of the PRD table. The PRD table itself must be constructed in a memory region that can be directly accessed by the SiI3512 controller. Once the PRD table is built, the controller must be informed of its location. This is accomplished by writing the 32-bit address of the PRD table to the PRD Table Address – IDEx register.

Enable DMA Transfer

DMA is enabled by writing bits [7:0] of the PCI Bus Master – IDEx register. Bit 3 of this register controls the direction of the DMA transfer; 1 = write to memory, 0 = read from memory. Setting bit 0 of the register enables the controller to perform DMA operations.

Note: Task file registers are inaccessible as long as bit 0 is set.

Wait for a PCI Interrupt

When a PCI interrupt occurs, read the PCI Master – IDEx status register and check the DMA status bits. The possible combinations of the status bits [18:16] are defined below.

0b000 = If the IDE device does not report an error, then the PRD table specified a size that is smaller than the IDE transfer size.
0b001 = DMA transfer in progress.
0b010 = The controller had a problem transferring data to/from memory.
0b100 = Normal completion.
0b101 = If the IDE device does not report an error, then the PRD specified a size that is larger than the IDE transfer size.

Make sure PCI bus master operation of the SiI3512 device is stopped by clearing bit 0 of the PCI Bus Master – IDEx register.

Note: The task file registers are not accessible as long as bit 0 is set. Clearing bit 0 causes bit 16 to be cleared as well.

Read the device status at bits [13:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was error.

Write '1' to bit 18 (write-one-to-clear) in the PCI Bus Master – IDEx register to clear the PCI Interrupt.

IDE Virtual DMA Read/Write Operation

In virtual DMA operation the controller uses a PIO data transfer mode to move data between an ATA/ATAPI device and the controller, and uses DMA to move that same data between the controller and the host memory. For ATA/ATAPI devices that cannot operate in a "true" DMA mode, virtual DMA provides two benefits; first, using DMA to move data reduces the demand on the host CPU, and second, systems that use virtual memory often require that data buffers that will be accessed directly by low level device drivers be "mapped" into the operating system's address space, in virtual DMA mode the CPU does not access the data buffer directly, so the overhead of obtaining the mapping to operating system address space is eliminated.

Using Virtual DMA with Non-DMA Capable Devices

After the SiI3512 controller is initialized with the initialization sequence described earlier and the ATA device has been initialized for PIO mode data transfer, virtual DMA read/write operations can be performed by following the programming sequence described below.

NOTE: The watchdog timer feature is compatible with virtual DMA operation.

Issue a PIO read/write command to the device.

Read Operation

Wait for a PCI interrupt.

Read the DMA status bits [18:16] of the PCI Bus Master – IDEx register, and check that bit 18 is set to make sure the interrupt was generated by the expected channel.

If expected channel interrupted, read bits [11:10] of the channel's IDEx Task File Timing + Configuration + Status register to determine the cause of the interrupt. Bit 11 is set if the ATA/ATAPI device has an interrupt pending, bit 10 is set if a virtual DMA operation completed.

If a virtual DMA operation completed:

Write '0x00' to bits [7:0] of the PCI Bus Master – IDEx register to disable DMA operation.

Write '1' to bits [18:17] of the PCI Bus Master –IDEx register to reset the DMA status and virtual DMA interrupt bits, and the PCI interrupt.

Check the previously read DMA status bits to ensure the DMA completed successfully.

Because ATA/ATAPI commands that transfer data using PIO can generate several interrupts during the data transfer phase of the command, a race condition is created between the interrupt indicating the completion of a virtual DMA operation, and the interrupt from the ATA/ATAPI device indicating it is ready to perform the next part of the data transfer. To prevent missing an ATA/ATAPI device interrupt due to this race condition, it is necessary to re-read the channel's IDEx Task File Timing + Configuration + Status register after disabling DMA operation and examining bit 11. If bit 11 is set, the ATA/ATAPI device is interrupting and should be serviced by following the steps below (assuming that the virtual DMA operation completed successfully).

If the ATA/ATAPI device has interrupted:

Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

Write ‘1’ to bit 18 of the PCI Bus Master – IDEx register to clear the DMA Complete bit (NOTE: The DMA Complete bit acts as a latched copy of the ATA interrupt line when the channel is not performing a DMA operation).

If the ATA/ATAPI device is not reporting an error, and DRQ is asserted (bit 27 of IDEx Task File Register 1), then the device is interrupting to transfer data to the host. To transfer the data, the DMA registers are setup to only perform that part of the data transfer expected for this interrupt. The DMA is setup similarly to the way it is when performing a normal read DMA command, but with one additional step. Before the DMA is enabled, the IDEx Virtual DMA/PIO Read Ahead Byte Count register must be written with the 32-bit count of the number of bytes to be transferred for this interrupt.

Repeat the above steps until all data for the read command has been transferred or an error has been detected.

Write Operation

Poll the IDEx Task File Register 1 bits [31:24] until either bit 27 (DRQ) is set indicating the device is ready for write data transfer, or bit 24 (ERR) is set indicating the device has detected an error with the write command.

If no error, and DRQ is asserted (bit 27 of IDEx Task File Register 1), then the device is waiting for write data transfer. To transfer the data, the DMA registers are setup to only perform that part of the data transfer expected at this time. For example, a Write Sectors command would expect to transfer 1 sector (512 bytes), while a Write Multiple command would expect to transfer the lesser of the number of sectors set by the Set Multiple Mode command or the total number of sectors specified by the Write Multiple command. The DMA is setup similarly to the way it is when performing a normal write DMA command, but with one additional step. Before the DMA is enabled, the IDEx Virtual DMA/PIO Read Ahead Byte Count register must be written with the 32-bit count of the number of bytes to be transferred.

Wait for a PCI interrupt.

Read the DMA status bits [18:16] of the PCI Bus Master – IDEx register, and check that bit 18 is set to make sure the interrupt was generated by the expected channel.

If expected channel interrupted, read bits [11:10] of the channel’s IDEx Task File Timing + Configuration + Status register to determine the cause of the interrupt. Bit 11 is set if the ATA/ATAPI device has an interrupt pending, bit 10 is set if a virtual DMA operation completed.

If a virtual DMA operation completed:

Write 0x00 to bits [7:0] of the PCI Bus Master – IDEx register to disable DMA operation.

Write ‘1’ to bits [18:17] of the PCI Bus Master –IDEx register to reset the DMA status and virtual DMA interrupt bits, and PCI interrupt.

Check the previously read DMA status bits to ensure the DMA completed successfully.

Because ATA/ATAPI commands that transfer data using PIO can generate several interrupts during the data transfer phase of the command, a race condition is created between the interrupt indicating the completion of a virtual DMA operation, and the interrupt from the ATA/ATAPI device indicating it is ready to perform the next part of the data transfer. To prevent missing an ATA/ATAPI device interrupt due to this race condition, it is necessary to re-read the channel’s IDEx Task File Timing + Configuration + Status register after disabling DMA operation and examining bit 11. If bit 11 is set, the ATA/ATAPI device is interrupting and should be serviced by following the steps below (assuming that the virtual DMA operation completed successfully).

If the ATA/ATAPI device has interrupted:

Read the device status at bits [31:24] in the IDEx Task File Register 1 register to clear the device interrupt and determine if there was an error.

Write ‘1’ to bit 18 of the PCI Bus Master – IDEx register to clear the DMA Complete bit (NOTE: The DMA Complete bit acts as a latched copy of the ATA interrupt line when the channel is not performing a DMA operation).

If the ATA/ATAPI device is not reporting an error, and DRQ is asserted (bit 27 of IDEx Task File Register 1), then the device is interrupting to transfer data to the device. To transfer the data, the DMA

registers are setup to only perform that part of the data transfer expected for this interrupt. The DMA is setup similarly to the way it is when performing a normal write DMA command, but with one additional step. Before the DMA is enabled, the IDEx Virtual DMA/PIO Read Ahead Byte Count register must be written with the 32-bit count of the number of bytes to be transferred for this interrupt.

Repeat the above steps starting at "Wait for PCI interrupt" until all data for the write command has been transferred or an error has been detected.

Using Virtual DMA with DMA Capable Devices

Even though a device may be DMA capable, there are ATA/ATAPI commands that require that a PIO mode be used to transfer data. For these commands, virtual DMA can be used to perform the data transfer. Using virtual DMA with an ATA/ATAPI device that has already been configured to use DMA for normal read/write operation is performed very much like the sequence described above for PIO mode only devices, but with the following additional consideration:

- The Data Transfer Mode – IDEx register associated with the ATA/ATAPI device needs to be programmed for a PIO type transfer mode **before** DMA operation is enabled, and must be re-programmed with the DMA/UDMA transfer type used during normal DMA operation once the virtual DMA operation is complete.

Second PCI Bus Master Registers Usage

In order to provide backward compatibility with existing drivers, the Physical Region Descriptor (PRD) tables used by the SiI3512 controller when performing DMA transfers suffer the following limitations; a PRD table entry cannot represent a memory area greater than 64k, nor can a PRD table entry represent a memory area that spans a 64k address boundary. Whenever DMA is initiated via the PCI Bus Master – IDEx registers, the foregoing limitations are enforced by the SiI3512 controller.

A feature known as Large Block Transfer in the SiI3512 controller allows drivers to get around the 64k size and address limits of PRD table entries expected by existing drivers. Large Block Transfer simplifies the creation of PRD tables by reducing the number of table entries that need to be created and eliminating the need to make sure a memory region does not cross a 64k boundary. Large Block Transfer mode is enabled whenever DMA is initiated by writing to the PCI Bus Master 2 – IDEx registers (base address 5, offset 0x10 or 0x18). When performing DMA in Large Block Transfer mode, the SiI3512 controller interprets the fields of a PRD table entry differently. In all other respects, DMA interrupt generation, DMA status bit interpretation, etc., Large Block Transfer mode behaves identically to a non-Large Block Transfer mode DMA operation. The following table describes the format of a PRD table entry:

Table 27. Physical Region Descriptor (PRD) Format

Bits	Description
Bits 31:0	32-bit starting address of the memory region.
Bits 47:32	When not operating in Large Block Transfer mode, this field specifies the size of the memory region. If the size of the memory region is greater than 64k, or crosses a 64k address boundary, then two or more PRD table entries will need to be created to describe it. If operating in Large Block Transfer mode, this field contains the least significant 16-bits of the size of the memory region.
Bits 62:48	If not operating in Large Block Transfer mode, this field is unused. If operating in Large Block Transfer mode, this field contains the most significant 15-bits of the size of the memory region.
Bit 63	When set, this bit indicates that this is the last entry in the PRD table.

Power Management

Power Management in the SiI3512 device is controlled by the following register bits.

Table 28. Power Management Register Bits

Register	Bits	Description
SMisc	PMCHG Bit 6	This bit reports a change in the Power Management mode. It corresponds to the interrupt enabled by bit 26 of SIEN.
SMisc	PMMODE Bits 5,4	These bits report the power management mode status: bit 5 corresponds to Slumber mode; bit 4 to Partial mode. A transition on either of these bits causes a Power Management mode change interrupt.
SError	W Bit 18	ComWake received from the Serial ATA bus
SMisc	ComWake Bit 11	Generates a ComWake condition on the Serial ATA bus
SMisc	PMREQ Bits 1,0	Generates a request from the Host for the Device to go to a Power Management state; bit 1 corresponds to Slumber mode; bit 0 corresponds to Partial mode. These bits are effective regardless of the state of the HPMDS bit.
SControl	IPM Bits 11-8	This bit field disables transitions to Partial or Slumber power management states; bit 9 corresponds to Slumber mode; bit 8 corresponds to Partial mode.
SStatus	IPM Bits 11-8	This bit field reports the power management state; ‘0110’ corresponds to Slumber mode; ‘0010’ corresponds to Partial mode.

Power Management Summary

There are two power management modes: Partial and Slumber. These power management modes may be software initiated through the SMisc register or device initiated from the Serial ATA device.

Transitions to and from either power management mode generate an interrupt, the Power Management Mode Change Interrupt, which may be masked in the SIEN register (bit 26).

Partial Power Management Mode

Partial mode may be initiated by software through the SMisc register (bit 0). By setting the bit, the software causes PMREQ_P primitives (Power Management Request – Partial) to be sent to the Serial ATA device, which will respond with either a PMACK or PMNAK. If a PMACK is received the Partial mode is entered; A PMNAK is ignored; the request remains asserted.

The Serial ATA device may initiate partial mode. This is indicated by the reception of PMREQ_P primitives from the device. Software enables the acknowledgement of this request by setting the IPM value in the SControl register to ‘00x1’. If enabled, a PMACK will be sent to the device; if not enabled, a PMNAK will be sent. When the request is received and its acknowledgement is enabled, Partial mode is entered.

Partial mode status is reported in both the SStatus register (‘0010’ in the IPM field) and the SMisc register (bit 4). Partial mode is cleared by setting the ComWake bit in the Smisc register. This will send a COMWAKE signal to the device through the Serial ATA link to initiate a Partial to On sequence. Partial mode can also be cleared through receipt of OOB signals from the device.

Slumber Power Management Mode

Slumber mode may be initiated by software through the SMisc register (bit 1). By setting the bit, software causes PMREQ_S primitives to be sent to the Serial ATA device, which will respond with either a PMACK or PMNAK. If a PMACK is received the Slumber mode is entered. A PMNAK is ignored; the request remains asserted.

The Serial ATA device may initiate slumber mode. This is indicated by the reception of PMREQ_S primitives. Software enables the acknowledgement of this request by setting the IPM value in the SControl register to ‘001x’. If enabled, a

PMACK will be sent to the device; if not enabled, a PMNAK will be sent. When the request is received and its acknowledgement is enabled, Slumber mode is entered.

Slumber mode status is reported in both the SStatus register (‘0110’ in the IPM field) and the SMisc register (bit 5).

Slumber mode is cleared by setting the ComWake bit in the Smisc register. This will send a COMWAKE signal to the device through the Serial ATA link to initiate a Slumber to On sequence. Slumber mode can also be cleared through receipt of OOB signals from the device.

Hot Plug Support

The state diagram below illustrates the logic to support Hot Plugging.

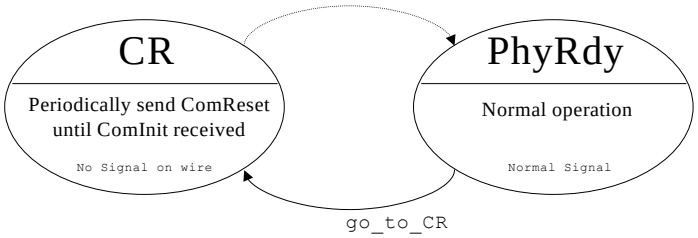


Figure 14. Hot Plug Support Logic

The go_to_CR signal is generated by a timer if the internal logic fails to detect valid signals from the Serial ATA wire for 200 ns. Logic behavior is as follows:

1. Initial power-up – A ComReset is generated during initial power up. If a device is present and operational, the PhyRdy state will be entered. If a device is not present or not responding, the CR state will be entered and ComReset will be generated every 100 ms.
2. Device is unplugged – The internal logic detects that no more signal is present on the Serial ATA wire. The timer will expire after 200 ns and go_to_CR will be asserted; the CR state will be entered and ComReset will be generated every 100 ms. The internal PHYRDY signal will go false causing an interrupt to the host driver (PHYRDY change interrupt, bit 16 of SError register; enabled by bit 16 of SIEN register).
3. Device is plugged in – The device will respond to the ComReset with a ComInit. Normal operation will commence and the internal logic will detect a PHYRDY signal going true causing an interrupt to the host driver (PHYRDY change interrupt, bit 16 of SError register; enabled by bit 16 of SIEN register).

FIS Support

This section describes the implementation of Serial ATA FIS support.

FIS Summary

The following table summarizes the implementation of FIS Support. Note that 14 FIS codes meet the criteria of FIS code selection in Serial ATA, and 8 out of the 14 are already defined.

Table 29. FIS Summary

FIS Code	FIS Name	Host to Device	Device to Host	Comment
0x27	Register (Host to Device)	√		Support Expanded Registers. HOB not sent to device (device bridge ignores HOB received). Can be individually controlled via PCI registers - default to reject
0x34	Register (Device to Host)		√	Support Expanded Registers. Host to Device transmission is possible as Transparent. Can be individually controlled via PCI registers - default to accept
0x39	DMA Activate		√	Supported per Serial ATA specification. Host to Device transmission is possible as Transparent. Can be individually controlled via PCI registers - default to accept
0x41	DMA Setup	√	√	On reception, the first 7 dwords of any FIS can be read directly by the PCI. Transmission: As transparent FIS Can be individually controlled via PCI registers - default to reject
0x46	Data	√	√	Supported per Serial ATA specification. Can be individually controlled via PCI registers - default to accept
0x58	BIST Activate	√	√	Support for reception of Far-End Retimed Loopback. No transmission supported. Can be individually controlled via PCI registers - default to accept for Far-End Retimed Loopback; default to reject for all other BIST types
0x5F	PIO Setup		√	Supported per Serial ATA specification. Host to Device transmission is possible as Transparent. Can be individually controlled via PCI registers - default to accept
0xA1	Set Device Bits		√	Supported per Serial ATA specification. Host to Device transmission is possible as Transparent. Can be individually controlled via PCI registers - default to accept
0xA6	reserved	TBD	TBD	Supported as one group of unrecognized FIS, together with other unsupported FISes, such as "Others" below, and FIS Code 0x27 in the reception direction. Can be individually controlled via PCI registers - default to reject
0xB8	reserved	TBD	TBD	
0xBF	reserved	TBD	TBD	
0xC7	reserved	TBD	TBD	
0xD4	reserved	TBD	TBD	
0xD9	reserved	TBD	TBD	
Others	reserved	TBD	TBD	Supported as one group of unrecognized FIS, together with other unsupported FISes (FIS Code 0x27, 0xA6, 0xB8, 0xBF, 0xC7, 0xD4, 0xD9) in the reception direction. All "Others" are controlled as a group via PCI registers - default to reject

FIS Transmission

There are two ways in which a FIS transmission is initiated:

- Protocol-initiated FIS transmission, e.g., when an ATA command is written to the SiI3512 controller it will send a Command Register FIS and expects some FISes (e.g., PIO Setup, Register, DMA Activate, Data, Set Device bits).
- Transparent FIS transmission.

FISes that are not protocol initiated can also be transmitted as transparent FISes, under the control of the host driver.

Sequence

Host sets the Transmit_FIS bit in the Smisc register (bit 30). This tells the Transport/Link logic that a transparent FIS needs to be transmitted.

The Transport/Link logic responds by setting itself up to transfer data from the host through UMDA cycles.

The host writes the data through the PCI interface. Note that the FIS header (Dword 0 that contains the FIS type) must also be written. The Transport/Link logic sends the FIS to the device. Note that:

There is no size limit on a transparent FIS. Data written to the SiI3512 device from setting of Transmit_FIS to setting of FIS_Done (see below) will be transmitted in a FIS.

There must be an even number of words.

As in Data FIS, upon a transmission error, no retries can be supported. The PCI block must restart the transparent FIS transmission from the beginning.

Serial ATA CRC is calculated by the Transport/Link logic. The host will NOT append the CRC at the end.

After the last write, the host sets the FIS_Done bit in the Smisc register (bit 31). This indicates to the link that all data for this transaction has been transferred. The Transport/Link logic will then close out the FIS by appending CRC and EOF and wait for termination. If R_OK is received from the downstream device, the Transmit_OK bit will be set to indicate to the host that the FIS has been successfully transferred to the device. If there is an error in the transmission process (e.g., the FIS not recognized by the downstream device) resulting in the device acknowledging the FIS with an R_ERR, the F bit of the Serror Register will be set (Bit 25).

The values of the status registers are latched and will not be cleared automatically. Before the next Transparent FIS is being sent, the host must clear the status bits by performing a write to the particular status registers.

FIS Reception

The SiI3512 controller is capable of receiving Unrecognized FIS types through an Interlocked FIS scheme. This capability is over and above the regular protocol related FISes as defined in the Serial ATA specifications.

In general, an internal table determines the behavior when receiving all possible FIS types. This table is defined in the register SFISCfg. The configuration codes in the SFISCfg register are defined as follows:

Table 30. Configuration Bits for FIS Reception

FISxxCFG[1:0]	Comments
0b00	Accept FIS without interlock. If there is no error detected for the entire FIS, R_OK will be sent after EOF is received. If any error is received, R_ERR will be sent after EOF
0b01	Reject FIS without interlock. R_ERR will be sent
0b10	Interlock. This allows the host to examine the first dwords of the FIS to determine whether to accept or reject the FIS
0b11	reserved.

The following table shows the default configurations of all Serial ATA FIS types:

Table 31. Default FIS Configurations

FIS Code	FIS Name	Configuration Bits		Comments
		Register Bits	Default Value	
0x27	Register (Host to Device)	FIS27cfg[1:0]	0b01	Default to reject FIS without interlock.
0x34	Register (Device to Host)	FIS34cfg[1:0]	0b00	Default to accept FIS without interlock.
0x39	DMA Activate	FIS39cfg[1:0]	0b00	Default to accept FIS without interlock.
0x41	DMA Setup	FIS41cfg[1:0]	0b01	Default to reject.
0x46	Data	FIS46cfg[1:0]	0b00	Default to accept FIS without interlock.
0x58	BIST Activate	FIS58cfg[1:0]	0b00	Default to accept for far-end retimed loopback, reject for any other.
0x5F	PIO Setup	FIS5Fcfg[1:0]	0b00	Default to accept FIS without interlock.

0xA1	Set Device Bits	FISa1cfg[1:0]	0b00	Default to accept FIS without interlock.
0xA6	reserved	FISa6cfg[1:0]	0b01	Default to reject FIS without interlock.
0xB8	reserved	FISb8cfg[1:0]	0b01	Default to reject FIS without interlock.
0xBF	reserved	FISbFcfg[1:0]	0b01	Default to reject FIS without interlock.
0xC7	reserved	FISc7cfg[1:0]	0b01	Default to reject FIS without interlock.
0xD4	reserved	FISd4cfg[1:0]	0b01	Default to reject FIS without interlock.
0xD9	reserved	FISd9cfg[1:0]	0b01	Default to reject FIS without interlock.
Others	reserved	FISocfg[1:0]	0b01	Default to reject FIS without interlock.

RxFIS[0-6] – First seven dwords received from device. RxFIS[0] is the first dword that contains the FIS header. RxFIS[6] is the last of the seven dwords received. It is enough to support DMA Setup FIS.

Note that:

FIS data can also be read out directly from RxFIS (first seven dwords).

All data to be transferred must be sent within one UDMA burst. Burst termination will not be allowed and may produce unpredictable result.

There is no limit on received frame size.

In a Data FIS, the receive FIFO will automatically advance one dword to skip the header. Upon an interlocked FIS, the FIFO read pointer will rewind to the beginning so that the first dword read is the header.

The following list summarizes the behavior:

- On power up, the default configurations are as follows:

  All defined FISes, except BIST Activate and DMA Setup, default to be supported (FISxxcfg[1:0] = '00').

  BIST Activate is default to be accepted ONLY for Far-end Retimed Loopback and to be rejected for any other BIST types.

  DMA Setup defaults to be rejected.

  All undefined FISes default to be rejected (FISxxcfg[1:0] = '01').

- Sequences:

  Upon reception of an unsupported FIS (FISxxcfg[1:0] = '01'), the Link/Transport Logic responds with R_ERR to the downstream device. The host will not be notified.

  Upon reception of a supported FIS (FISxxcfg[1:0] = '00'), the Link/Transport Logic responds with R_OK at WTRM (if no error is detected) or R_ERR (if an error is detected) to the downstream device. The host will be notified only as required by the protocol.

  Upon reception of an interlocked FIS (FISxxcfg[1:0] = '10'), the Link/Transport Logic sets the IntrlckFIS bit in the Smisc register. The following describes the possible sequence of events:

  Sequence 1:

  The Link Logic will continue to receive data while its buffer is being filled up. IntrlckFIS will cause an interrupt to the host.

  The first 7 dwords of the FIS are available to the host in the RxFIS0 to RxFIS6 registers. The driver will check the FIS type, clean up the PCI section, arm the DMA controller, and then assert the Rx_IFIS bit in the Smisc register.

  The Link/Transport Logic transfers the received FIS, including the header, through the PCI interface to the host.

  When all the data is received with no errors, the Link/Transport Logic will assert the IFIS_OK bit in the Smisc register. Otherwise one of the error bits will be set in the Serror register.

  The host will set the Accept_IFIS bit to accept or Reject_IFIS to reject the FIS.

  If no error is detected inside the frame and the Accept_IFIS bit is asserted, the Link/Transport Logic will send R_OK to the downstream device. If Reject_IFIS is asserted or any error is detected, the

Link/Transport Logic will respond with R_ERR. Note that there is an interlock - if the frame is good, it will always wait for the Accept_IFIS or Reject_IFIS (if not asserted already) before responding.

Sequence 2:

Link/Transport Logic will continue to receive data while its buffer is being filled up. IntrlckFIS will cause an interrupt to the host.

Host reads the header; the driver will check the FIS type in RxFIS register and knows that the entire FIS is not larger than the size of RxFIS0 to 6 register.

Host waits for IFIS_OK (if any error detected – the error signals).

If IFIS_OK is received, host reads all data directly via PCI registers and then issues a Accept_IFIS (Link/Transport Logic to send R_OK) or a Reject_IFIS (Link/Transport Logic to send R_ERR).

If any error is detected, host can ignore, the Link will respond with R_ERR anyway.

FIS Types Not Affiliated with Current ATA/ATAPI Operations

BIST Support

Far-End Retimed Loopback is supported in reception mode only. All other BIST codes will be rejected via R_ERR. It defaults to be interlocked supported (for Far-End Retimed Loopback only).

The SiI3512 controller does not support any BIST in transmission mode. There is no provision to send the test patterns and compare against loopback data.

BIST Signals

When the SiI3512 controller enters the BIST operation, the “PHY offline” mode will be set in the DET bits of the Sstatus register. This condition will remain asserted until the host generates an ATA reset (hreset_b asserted) or a COMINIT is received from the device.

DMA Setup

DMA Setup FIS can only be sent as a transparent FIS. On Power-up, DMA Setup FIS defaults to be rejected.

First Party DMA Read of Host memory by Device

Sequence (FIS41cfg[1:0] = '10', i.e., interlocked):

- Device sends DMA Setup FIS to host. The "D" field in the FIS is '0'.
- The IntrlckFIS bit is set and causes an interrupt to the host.
- The host driver checks the FIS type (RxFIS), sets up, and arms the DMA controller.
- The host sets the DMAOutEn in the Serial ATA SMisc register.
- The host sets the FPDMAWr in the Serial ATA SMisc register.
- The host sets the Accept_FIS bit to accept the FIS.
- The host sends one or more Data FISes. Note that no DMA Activate FIS is required for first party DMA.
- There is no need to report transfer status.
- The host clears the DMAOutEn when the transfer count is exhausted.

First Party DMA Write of Host Memory by Device

Sequence (FIS41cfg[1:0] = '10', i.e., interlocked):

- Device sends DMA Setup FIS to host. The "D" field in the FIS is '1'.
- The IntrlckFIS bit is set and causes an interrupt to the host.
- The host driver checks the FIS type (RxFIS), sets up, and arms the DMA controller.
- The host sets the DMAInEn in the Serial ATA SMisc register.
- The host sets the Accept_FIS bit to accept the FIS.
- The device sends one or more Data FISes.
- There is no need to report transfer status.
- The host clears the DMAInEn when the transfer count is exhausted.

ATA Commands Supported

Data Modes

The SiI3512 device has an internal data path interface between the PCI block and the Serial ATA controller block. The data modes (Register mode, PIO mode and DMA mode) are of no significance inside the SiI3512 controller.

ATA Commands

The SiI3512 controller decodes ATA commands in hardware. The commands supported include ATA/ATAPI-5 and ATA/ATAPI-6 commands, including the 48-bit LBA extended commands. Certain obsolesced commands are also supported. The supported commands are listed below:

Table 32. Supported ATA Commands

Command	Command/ Features Codes	Comment
CFA Erase Sectors	0xC0	
CFA Request Extended Error Code	0x03	
CFA Translate Sector	0x87	
CFA Write Multiple without Erase	0xCD	
CFA Write Sectors without Erase	0x38	
Check Media Card Type	0xD1	
Check Power Mode	0xE5	
Configure Stream	0x51	
Device Configuration Freeze Lock	0xB1/0xC1	
Device Configuration Identify	0xB1/0xC2	
Device Configuration Restore	0xB1/0xC0	
Device Configuration Set	0xB1/0xC3	
Device Reset	080x	
Download Microcode	0x92	
Execute Device Diagnostics	0x90	The two Serial ATA ports for the SiI3512 device are both "single masters".
Flush Cache	0xE7	
Flush Cache Ext	0xEA	48-bit LBA Command
Format Track	0x50	Obsolesced vendor-specific command, needs to be programmed as vendor-specific commands
Get Media Status	0xDA	
Identify Device	0xEC	
Identify Packet Device	0xA1	
Idle	0xE3	
Idle Immediate	0xE1	
Initialize Device Parameters	0x91	Obsolesced in ATA/ATAPI-6.
Media Eject	0xED	
Media Lock	0xDE	
Media Unlock	0xDF	
Nop	0x00	
Packet	0xA0	
Read Buffer	0xE4	
Read DMA	0xC8	
	0xC9	Obsolesced Command code supported, decoded as Command Code 0xC8
Read DMA Ext	0x25	48-bit LBA Command
Read DMA Queued	0xC7	
Read DMA Queued Ext	0x26	48-bit LBA Command

Command	Command/ Features Codes	Comment
Read Log Ext	0x2F	
Read Long	0x22	Obsolesced command
	0x23	
Read Multiple	0xC4	
Read Multiple Ext	0x29	48-bit LBA Command
Read Native Max Address	0xF8	
Read Native Max Address Ext	0x27	48-bit LBA Command
Read Sector(s)	0x20	
	0x21	Obsolesced Command code supported, decoded as Command Code 0x20
Read Sector(s) Ext	0x24	48-bit LBA Command
Read Stream DMA	0x2A	
Read Stream PIO	0x2B	
Read Verify Sector(s)	0x40	
	0x41	Obsolesced Command code supported, decoded as Command Code 0x40
Read Verify Sector(s) Ext	0x42	48-bit LBA Command
ReadFPDMAQueued	0x2C	
Recalibrate	0x10	Obsolesced command supported.
Security Disable Password	0xF6	
Security Erase Prepare	0xF3	
Security Erase Unit	0xF4	
Security Freeze Lock	0xF5	
Security Set Password	0xF1	
Security Unlock	0xF2	
Seek	0x70	
Service	0xA2	
Set Features	0xEF	
Set Max Address	0xF9/0x00	
Set Max Address Ext	0x37	48-bit LBA Command
Set Max Freeze Lock	0xF9/0x04	
Set Max Lock	0xF9/0x02	
Set Max Unlock	0xF9/0x03	Obsolesced command supported.
Set Max Set Password	0xF9/0x01	
Set Multiple Mode	0xC6	The SiI3512 controller intercepts the command to set up the number of sectors for a DRQ block upon this command.
Sleep	0xE6	
Smart Disable Operations	0xB0/0xD9	
Smart Enable Operations	0xB0/0xD8	
Smart Enable/Disable Attributes Autosave	0xB0/0xD2	
Smart Execute Off-Line Immediate	0xB0/0xD4	
Smart Read Attribute Thresholds	0xB0/0xD1	Obsolesced command supported.
Smart Read Data	0xB0/0xD0	
Smart Read Log	0xB0/0xD5	
Smart Return Status	0xB0/0xDA	
Smart Save Attribute Values	0xB0/0xD3	Obsolesced command supported.
Smart Write Log	0xB0/0xD6	
Standby	0xE2	
Standby Immediate	0xE0	
Write Buffer	0xE8	

Command	Command/ Features Codes	Comment
Write DMA	0xCA	
	0xCB	Obsolesced Command code supported, decoded as Command Code 0xCA
Write DMA Ext	0x35	48-bit LBA Command
Write DMA Queued	0xCC	
Write DMA Queued Ext	0x36	48-bit LBA Command
Write Log Ext	0x3F	
Write Long	0x32	Obsolesced command supported
	0x33	
Write Multiple	0xC5	
Write Multiple Ext	0x39	48-bit LBA Command
Write Sector(s)	0x30	
	0x31	Obsolesced Command code supported, decoded as Command Code 0x30
Write Sector(s) Ext	0x34	48-bit LBA Command
Write Stream DMA	0x3A	
Write Stream PIO	0x3B	
WriteFPDMAQueued	0x3C	

Obsolesced Commands

Certain obsolesced commands are supported as shown in the table above. Commands Read Long and Write Long are to be treated differently (see the following section)

Read/Write Long

Read Long and Write Long commands are implemented in accordance with the ATA/ATAPI-3. The PIO Mode used (Mode 0) is of no significance in the SiI3512 controller, as the data path interface between the PCI and the Link/Transport logic is internal. The number of vendor-specific bytes is provided by the Serial ATA PIO Setup FIS from the downstream device as follows:

$$n = ((XC - 512) + 1) \div 2$$ (i.e., XC – 512 divided by 2 with round up)

where:

- n is the number of vendor-specific bytes.
- XC is the transfer count.

The total number of data dwords in the Data FIS is given by:

$$m = (XC + 3) \div 4$$ (i.e., XC divided by 4 with round up)

where:

- m is the number of data dwords in the Data FIS, excluding the FIS header (and CRC).
- XC is the transfer count.

In this command, the Data FIS must use the following format:

Table 33. Data FIS

Dword	Byte 3	Byte 2	Byte 1	Byte 0
0	Data FIS Header			
1	Sector Data Byte 3	Sector Data Byte 2	Sector Data Byte 1	Sector Data Byte 0
2	Sector Data Byte 7	Sector Data Byte 6	Sector Data Byte 5	Sector Data Byte 4
3... 126				
127	Sector Data Byte 507	Sector Data Byte 506	Sector Data Byte 505	Sector Data Byte 504
128	Sector Data Byte 511	Sector Data Byte 510	Sector Data Byte 509	Sector Data Byte 508
129	Don't care	Vendor-Specific Byte 1	Don't care	Vendor-Specific Byte 0
130	Don't care	Vendor-Specific Byte 3	Don't care	Vendor-Specific Byte 2
...				
Last (n is even)	Don't care	Vendor-Specific Byte n-1	Don't care	Vendor-Specific Byte n-2
Last (n is odd)	Don't care	Don't care	Don't care	Vendor-Specific Byte n-1

Vendor-Specific Command Support

The SiI3512 controller supports most vendor-specific commands that utilize existing protocols.

Silicon Image's Vendor-Specific Commands

Silicon Image defines several vendor-specific commands (all of which use Expanded Features in 48-bit LBA addressing) to support vendor-specific and reserved commands:

- VS Unlock Vendor Specific: Unlock the host or device to support vendor-specific commands.
- VS Unlock Reserved: Unlock the host or device to support reserved commands.
- VS Unlock Individual: Unlock the host or device to support individual vendor-specific and reserved commands.
- VS Lock: Lock the host or device to abort all vendor-specific and reserved commands.
- VS Set General Protocol: Determine the General Protocol Code to be used for all subsequent vendor-specific commands (if unlocked via a VS Unlock Vendor Specific command) and reserved commands (if unlocked via a VS Unlock Reserved command).
- VS Set Command Protocol: Select protocols for individual vendor-specific and reserved commands (if unlocked via a VS Unlock Individual command). A Command Protocol Table will be maintained. Commands set up via this will follow the protocol set in this command instead of the original command protocol. Hence, commands can be "overloaded" using this method.

Potential Conflicts with Other Vendor-Specific Commands

The commands chosen use Subcommand (Features) code 0xF1 under the SMART command (0xB0). While this code is not expected to be used by device manufacturers, there is always the possibility that it is used. If such conflict happens, the device manufacturers must reassign a new code to the conflicting command in order to use this scheme.

Other Expanded Features Codes

The commands above use some of the l Expanded Features Codes. All Expanded Features Codes under Command Code 0xB0 and Subcommand (Features) Code 0xF1 are reserved as Silicon Image vendor-specific commands.

Vendor-Specific, Reserved, Retired and Obsolesced Commands

These types of commands are treated differently:

- Vendor-specific commands: Expect for those commands whose protocols are individually set (via the VS Unlock Individual and VS Set Command Protocol commands), the host or device must be unlocked via the VS Unlock

Vendor Specific command before such commands can be issued. Otherwise, vendor-specific commands are aborted.

- Reserved commands: Expect for those commands whose protocols are individually set (via the VS Set Unlock Individual and VS Set Command Protocol commands), the host or device must be unlocked via the VS Unlock Reserved command before such commands can be issued. Otherwise, reserved commands are aborted.
- Obsolesced and Retired commands: Implementation of such commands is optional.

Definitions

Command — Unless otherwise stated, this is the value written to the ATA Command register.
Command Code — This is the code corresponding to the ATA command. It is also a field in the Command Protocol Table.
Command Protocol Table — The table that contains the individual vendor-specific and reserved commands supported.
Features — Unless otherwise stated, this is the value written to the ATA Features register.
Features Code — This is the code corresponding to the ATA Features register. It is also a field in the Command Protocol Table.
Features Mask — This is a field in the Command Protocol Table that allows several Features Codes to be used for the same command.
General Protocol Code — On a VS Set General Protocol command after a VS Unlock Vendor Specific or VS Unlock Reserved command, the General Protocol Code will be set as the protocol for all undefined vendor-specific (if unlocked) and/or undefined reserved (if unlocked) commands. An undefined vendor-specific/reserved command is one that does not have an entry in the Command Protocol Table.
Protocol Code — This code determines the protocol associated with a command. It is also a field in the Command Protocol Table.
Subcommand Code — Same as Features Code.
VS Features Set — The commands needed to support this scheme (see SiI3512 Vendor-Specific Commands section on page 100).
VS State Machine — The state machine that determines what vendor-specific and reserved commands are to be supported (see State Transitions section on page 111).

Scheme

Reset

Upon any hardware reset or the Serial ATA COMRESET, or COMINIT, the VS State Machine in the SiI3512 controller will be initialized to the locked state (the "default" state), which will abort all vendor-specific and reserved commands.

Soft Reset (via Device Control register bit 2) does NOT affect the VS State Machine.

Operation

The following summarizes how the vendor-specific/reserved commands are supported. Detailed operations are described in later sections.

1. The default state is locked. All vendor-specific commands will be aborted.
2. Unlock:

   To unlock the SiI3512 controller and/or device to support vendor-specific commands: Issue a VS Unlock Vendor Specific command. The SiI3512 device will also send this command to the Serial ATA device. If the downstream Serial ATA device is a bridge, the device bridge may optionally issue this command to the attached parallel ATA device. Note that the unlock will take effect in the SiI3512 controller even if an ABORT status is reported.

   To unlock the SiI3512 controller and/or device to support reserved commands: Issue a VS Unlock Reserved command. The SiI3512 controller will also send this command to the Serial ATA device. If the downstream Serial ATA device is a bridge, the device bridge may optionally issue this command to the attached parallel ATA device. Note that the unlock will take effect in the SiI3512 device even if an ABORT status is reported.

   To support individual vendor-specific or reserved command: Issue a VS Unlock Individual command.

Combinations of the above can be supported by simply issuing the appropriate combinations of VS Unlock Vendor Specific, VS Unlock Reserved and VS Unlock Individual commands.

3. Set protocol. There are two ways to set up protocol(s):

   Issue a VS Set Command Protocol command to set up a protocol for a specific command. The information is logged in a Command Protocol Table inside the SiI3512 controller. This protocol will remain valid until overwritten by a VS Set Command Protocol command that overwrites the Command Protocol Table entry, the VS Lock command, hardware reset, COMRESET, or COMINIT. The SiI3512 controller will also send this command to the Serial ATA device. If the downstream Serial ATA device is a bridge, the device bridge may optionally issue this command to the attached parallel ATA device. Note that the protocol will be set in the SiI3512 controller even if an ABORT status is reported. If more than one command protocol has to be set up, a VS Set Command protocol will have to be issued for each command.

   Issue a VS Set General Protocol command to set the General Protocol Code for the next vendor-specific command. This protocol will remain valid until the next VS Set General Protocol command, VS Lock command, hardware reset, COMRESET, or COMINIT. The SiI3512 controller will also send this command to the Serial ATA device. If the downstream Serial ATA device is a bridge, the device bridge may optionally issue this command to the attached parallel ATA device. Note that the protocol will be set in the SiI3512 controller even if an ABORT status is reported. Commands already set up via the VS Set Command Protocol will follow the protocol set in the VS Set Command Protocol command instead of the one set in this command.

4. Issue any command:

   Any vendor-specific commands (if unlocked for vendor-specific commands) or reserved commands (if unlocked for reserved commands) that has an associated protocol set via the VS Set Command Protocol command will be executed using that protocol.

   Any vendor-specific commands (if unlocked for vendor-specific commands) or reserved commands (if unlocked for reserved commands) that does not have an associated protocol, i.e. not set up by the VS Set Command Protocol command, will be executed using the protocol loaded from the latest VS Set General Protocol command.

   Other supported commands will follow the predefined protocols.

   Other unsupported commands will be aborted.

5. To change the protocol for vendor-specific commands, simply reissue the VS Set General Protocol or the VS Set Command Protocol command with the new protocol.

6. When done, issue the VS Lock command to return to the default VS state. The SiI3512 controller will also send the VS Lock command to the Serial ATA device. If the downstream Serial ATA device is a bridge, the device bridge may optionally issue this command to the attached parallel ATA device. Note that the lock will take effect in the SiI3512 controller even if an ABORT status is reported.

SiI3512 Vendor-Specific Commands

Feature Set/Command Summary

Table 34. Vendor-Specific Command Summary

Command	Command Code	Features Code	Expanded Features Code	Description
VS Lock	0xB0	0xF1	0xD5	Return VS state machine to VS_LOCKED.
VS Unlock Vendor Specific	0xB0	0xF1	0x12	Unlock VS state machine to support vendor-specific commands.
VS Unlock Reserved	0xB0	0xF1	0x22	Unlock VS state machine to support reserved commands.
VS Unlock Individual	0xB0	0xF1	0x32	Unlock VS state machine to support reserved commands.
VS Set General Protocol	0xB0	0xF1	0xF0	Set the General Protocol Code for all vendor-specific commands and reserved commands, if the corresponding command types are unlocked. The vendor-specific and reserved commands that are individually set via VS Set Command Protocol commands will not follow the protocol set by this command.
VS Set Command Protocol	0xB0	0xF1	0x87	Set protocol for an individual vendor-specific or reserved command. The information is logged in a Command Protocol Table entry.
	0xB0	0xF1	Other than above	Reserved.

Compared with other features sets, The VS Features Set ignores the bit 0 (ERR) in the Status register together with the Error register. All commands are considered completed once BSY = 0 and DRDY = 1 in the Status register.

VS Lock

Command/Subcommand/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0xD5

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0xD5							
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		na	na	na	na	na	na	na	na
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

This command locks the SiI3512 controller from supporting vendor-specific commands. All vendor-specific and reserved commands issued afterwards will be aborted.

Non-data (ext) protocol will be used with this command. The SiI3512 device will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.
- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512 controller.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512.

  However, both the SiI3512 controller and the device bridge will ignore the abort status and will consider the VS block locked.
- Case 3: The Serial ATA device is a native device and responds with an abort. The SiI3512 controller will ignore the abort status and will consider the VS block locked.

In other words, regardless of the status reported (aborted or complete), the SiI3512 controller and device that support this scheme will be locked.

VS Unlock Vendor Specific

Command/Subcommand/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0x12

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0x12							
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		**na**	**na**	**na**	**na**	**na**	**na**	**na**	**na**
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

This command unlocks the SiI3512 to support vendor-specific commands. Once this command is executed, the SiI3512 will remain unlocked until:

- A VS Lock command that returns the VS state to the default locked state, or;
- A hardware reset, or COMINIT or COMRESET.

Note that the VS Unlock Individual command, the VS Unlock Reserved command and Soft Reset have no effect on the VS state.

- If a VS Unlock Individual command is issued afterwards, the SiI3512 controller will be unlocked for both individual vendor-specific/reserved commands and other vendor-specific commands.
- If a VS Unlock Reserved command is issued afterwards, the SiI3512 controller will be unlocked for both vendor-specific and reserved commands.
- If both VS Unlock Individual and VS Unlock Reserved are issued afterwards, the SiI3512 controller will be unlocked for individual vendor-specific/reserved commands, as well as other vendor-specific and reserved commands.

The SiI3512 will use the non-data (ext) protocol with this command. The SiI3512 will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.

- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512.

  However, the SiI3512 will ignore the abort status and will consider the unlock event successful.

- The Serial ATA device is a native device and responds with an abort. The SiI3512 will ignore the abort status and will consider the unlock event successful.

In other words, regardless of the status reported (aborted or complete), the SiI3512 will be unlocked to support vendor-specific commands.

VS Unlock Reserved

Command/Subcommand/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0x22

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0x22							
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		na	na	na	na	na	na	na	na
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

This command unlocks the SiI3512 to support reserved commands. Once this command is executed, the SiI3512 will remain unlocked until:

- A VS Lock command that returns the VS state to the default locked state, or;
- A hardware reset, or COMINIT or COMRESET.

Note that the VS Unlock Vendor Specific command, the VS Unlock Individual command and Soft Reset have no effect on the VS state.

- If a VS Unlock Vendor Specific command is issued afterwards, the SiI3512 will be unlocked for both reserved and vendor-specific commands.
- If a VS Unlock Individual command is issued afterwards, the SiI3512 will be unlocked for both individual vendor-specific/reserved command protocols and other reserved commands.
- If both VS Unlock Vendor Specific and VS Unlock Individual are issued afterwards, the SiI3512 be unlocked for individual vendor-specific/reserved command protocols, as well as other vendor-specific and reserved commands.

The SiI3512 will use the non-data (ext) protocol with this command. The SiI3512 will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.
- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512.

  However, the SiI3512 will ignore the abort status and will consider the unlock event successful.
- Case 3: The Serial ATA device is a native device and responds with an abort. The SiI3512 will ignore the abort status and will consider the unlock event successful.

In other words, regardless of the status reported (aborted or complete), the SiI3512 will be unlocked to support reserved commands.

VS Unlock Individual

Command/Subcommand/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0x32

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0x32							
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		na	na	na	na	na	na	na	na
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

This command unlocks the SiI3512 to support individual vendor-specific and reserved commands. Once this command is executed, the SiI3512 will remain unlocked until:

- A VS Lock command that returns the VS state to the default locked state, or;
- A hardware reset, or COMINIT or COMRESET.

Note that the VS Unlock Vendor Specific command, the VS Unlock Reserved command and Soft Reset have no effect on the VS state.

- If a VS Unlock Vendor Specific command is issued afterwards, the SiI3512 will be unlocked for both individual command protocols and other vendor-specific commands.
- If a VS Unlock Reserved command is issued afterwards, the SiI3512 will be unlocked for both individual vendor-specific/reserved command and other reserved commands.
- If both VS Unlock Vendor Specific and VS Unlock Reserved are issued afterwards, SiI3512 will be unlocked for individual vendor-specific/reserved command, as well as other vendor-specific and reserved commands.

The SiI3512 will use the non-data (ext) protocol with this command. The SiI3512 will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.
- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512.

  However, the SiI3512 will ignore the abort status and will consider the unlock event successful.
- Case 3: The Serial ATA device is a native device and responds with an abort. The SiI3512 will ignore the abort status and will consider the unlock event successful.

In other words, regardless of the status reported (aborted or complete), the SiI3512 will be unlocked to support individual vendor-specific/reserved commands.

VS Set General Protocol

Command/Subcommand Code/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0xF0

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0xF0							
Sector Count	Current	na							
	Previous (Expanded)	Protocol Code							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		na	na	na	na	na	na	na	na
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

If the VS state is unlocked for vendor specific or for reserved, this command will set the General Protocol Code for the next vendor-specific/reserved command(s), except for those individually set via the VS Set Command Protocol commands. The protocol will return to, Abort (Protocol Code = 0x00) upon a lock event, such as:

- A VS Lock command to return the VS state to the default locked state, or;
- A hardware reset, or COMINIT or COMRESET.

The General Protocol is passed to the SiI3512 and device via the Expanded Sector Count register.

The SiI3512 will use the non-data (ext) protocol with this command. The SiI3512 will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.
- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512.

  However, the SiI3512 will ignore the abort status and will consider the unlock event successful.
- Case 3: The Serial ATA device is a native device and responds with an abort. The SiI3512 will ignore the abort status and will consider the protocol set.

In other words, regardless of the status reported (aborted or complete), the SiI3512 will accept the protocol as valid.

VS Set Command Protocol

Command/Subcommand/Expanded Features Code

Command Code: 0xB0
Subcommand (Features) Code: 0xF1
Expanded Features Code: 0x87

Protocol

Non-data (Ext)

Inputs

Register		7	6	5	4	3	2	1	0
Features	Current	0xF1							
	Previous (Expanded)	0x87							
Sector Count	Current	0	0	0	0	Code Tag			
	Previous (Expanded)	Protocol Code							
LBA Low	Current	Command Code							
	Previous (Expanded)	na							
LBA Mid	Current	Features Code							
	Previous (Expanded)	Features Mask							
LBA High	Current	0x00							
	Previous (Expanded)	0x00							
Device		obs	na	obs	DEV[1]	na	na	na	na
Command		0xB0							

1. The DEV bit usage in the Serial ATA specification must be followed.

Outputs

Register		7	6	5	4	3	2	1	0
Error		na	na	na	na	na	na	na	na
Sector Count	Current	na							
	Previous (Expanded)	na							
LBA Low	Current	na							
	Previous (Expanded)	na							
LBA Mid	Current	na							
	Previous (Expanded)	na							
LBA High	Current	na							
	Previous (Expanded)	na							
Device		obs	na	obs	DEV[1]	na	na	na	na
Status		BSY	DRDY	na	na	na	na	na	na[2]

1. The DEV bit usage in the Serial ATA specification must be followed.
2. Error bit will be ignored. Completion is determined by BSY = 0 and DRDY = 1 only.

Description

If the VS state is unlocked for individual vendor-specific/reserved commands, this command will set the protocol for the specific commands. Up to 16 individual vendor-specific/reserved commands are supported via a Command Protocol Table. The 16 entries are organized as follows:

Table 35. 16-Entry Command Protocol Table

Code Tag (Entry #)	Command Code	Features Code	Features Mask	Protocol Code
0x0				
0x1				
...				
0xE				
0xF				

When a command is issued, its Command and Features registers will be compared against all of the above entries. If the following conditions are all met, the protocol for that entry will be used:

- Command = Command Code, and;
- (Features ⊕ Features Code) & Features Mask = 0x00.

Note that:

- If a vendor-specific or reserved command is mapped to more than one entry the result is indeterminate.
- Commands set up using this command will follow the new protocol instead of the original protocol defined for the command, i.e., commands can be “overloaded” using this mechanism.

Upon a lock event, all Command Codes will be initialized to NOP (0x00) and all Protocol Codes will be initialized to Abort (0x00). The following conditions are considered lock events:

- A VS Lock command to return the VS state to the default locked state, or;
- A hardware reset, or COMINIT or COMRESET.

The following registers are used when issuing the command (but have no meaning for outputs):

Register		Bit(s)	Field	Description
Sector Count	Current	7-4	0x0	Must be 0x0. Reserved for expansion if more than 16 individual vendor-specific/reserved commands are supported.
		3-0	Code Tag	Up to 16 individual vendor-specific/reserved commands are supported. This code tag is to select which of the 16 entries the code is to be written to. Earlier content in that entry will be replaced with the new information.
	Previous (Expanded)	7-0	Protocol Code	
LBA Low	Current	7-0	Command Code	The Command register value for the individual vendor-specific/reserved command.
	Previous (Expanded)	7-0	na	Not used.
LBA Mid	Current	7-0	Features Code	The Features register value for the individual vendor-specific/reserved command.
	Previous (Expanded)	7-0	Features Mask	One single protocol can be assigned to a group of commands with the same Command Code but different Features Codes. If a Features Mask bit is '0', the corresponding Features Code bit will be ignored for comparison.
LBA High	Current	7-0	0x00	Reserved for Expanded Features Code.
	Previous (Expanded)	7-0	0x00	Reserved for Expanded Features Mask.

The SiI3512 will use the non-data (ext) protocol with this command. The SiI3512 will send this command to the Serial ATA device. The following situations may happen:

- Case 1: The Serial ATA device (native or bridge) responds with a completed status. Both sides are set up to support this scheme.

- Case 2: The Serial ATA device bridge supports this scheme. It may optionally pass this command to a parallel ATA device:

  If passed to a parallel ATA device, the parallel ATA device responds with an abort status, which may be reported back to the SiI3512.

  If not passed to a parallel ATA device, the device bridge should still respond with a device-to-host Register FIS to terminate BSY in the SiI3512

  However, the SiI3512 will ignore the abort status and will consider the unlock event successful.
- Case 3: The Serial ATA device is a native device and responds with an abort. The SiI3512 will ignore the abort status and will consider the protocol set.

In other words, regardless of the status reported (aborted or complete), the SiI3512 will accept the protocol as valid.

State Transitions

Table 36. Default State — VS_LOCKED

VS_LOCKED	Vendor-specific/Reserved commands not supported. All vendor-specific and reserved commands will result in an ABORT status. General Protocol Code will be 0x00. Command Protocol Table initialized with all Command Codes = 0x00 and all Protocol Codes = 0x00.		
1	Received VS Unlock Vendor Specific command	→	VS_VS
2	Received VS Unlock Reserved command	→	VS_RSV
3	Received VS Unlock Individual command	→	VS_IND
4	Otherwise	→	VS_LOCKED

Table 37. VS_VS State Transition

VS_VS	On VS Set General Protocol command, set General Protocol Code. Commands other than vendor-specific or reserved commands will be executed according to the predefined protocol. All vendor-specific commands will be executed according to the General Protocol Code. All reserved commands will result in an ABORT status.		
1	Received VS Unlock Reserved command	→	VS_VS_RSV
2	Received VS Unlock Individual command	→	VS_VS_IND
3	Received VS Lock command	→	VS_LOCKED
4	Otherwise	→	VS_VS

Table 38. VS_RSV State Transition

VS_RSV	On VS Set General Protocol command, set General Protocol Code. Commands other than vendor-specific or reserved commands will be executed according to the predefined protocol. All reserved commands will be executed according to the General Protocol Code. All vendor-specific commands will result in an ABORT status.		
1	Received VS Unlock Vendor Specific command	→	VS_VS_RSV
2	Received VS Unlock Individual command	→	VS_RSV_IND
3	Received VS Lock command	→	VS_LOCKED
4	Otherwise	→	VS_RSV

Table 39. VS_IND State Transition

VS_IND	On VS Set Command Protocol command, update the corresponding Command Protocol Table entry. All commands with entries in the Command Protocol Table will be executed according to the Protocol Code in the corresponding Command Protocol entry. All other non-vendor-specific and non-reserved commands will be executed according to the predefined protocol. All other commands will result in an ABORT status.		
1	Received VS Unlock Reserved command	→	VS_VS_RSV
2	Received VS Unlock Vendor Specific command	→	VS_VS_IND
3	Received VS Lock command	→	VS_LOCKED
4	Otherwise	→	VS_IND

Table 40. VS_VS_RSV State Transition

VS_VS_RSV	On VS Set General Protocol command, set General Protocol Code. Commands other than vendor-specific or reserved commands will be executed according to the predefined protocol. All vendor-specific/reserved commands will be executed according to the General Protocol Code.		
1	Received VS Unlock Individual command	→	VS_VS_RSV_IND
2	Received VS Lock command	→	VS_LOCKED
3	Otherwise	→	VS_VS_RSV

Table 41. VS_VS_IND State Transition

VS_VS_IND	On VS Set General Protocol command, set General Protocol Code. On VS Set Command Protocol command, update the corresponding Command Protocol Table entry. All commands with entries in the Command Protocol Table will be executed according to the Protocol Code in the corresponding Command Protocol entry. All other vendor-specific commands will be executed according to the General Protocol Code. All other non-reserved commands will be executed according to the predefined protocol. All other commands will result in an ABORT status.		
1	Received VS Unlock Reserved command	→	VS_VS_RSV_IND
2	Received VS Lock command	→	VS_LOCKED
3	Otherwise	→	VS_VS_IND

Table 42. VS_RSV_IND State Transition

VS_RSV_IND	On VS Set General Protocol command, set General Protocol Code. On VS Set Command Protocol command, update the corresponding Command Protocol Table entry. All commands with entries in the Command Protocol Table will be executed according to the Protocol Code in the corresponding Command Protocol entry. All other reserved commands will be executed according to the General Protocol Code. All other non-vendor-specific commands will be executed according to the predefined protocol. All other commands will result in an ABORT status.		
1	Received VS Unlock Vendor Specific command	→	VS_VS_RSV_IND
2	Received VS Lock command	→	VS_LOCKED
3	Otherwise	→	VS_RSV_IND

Table 43. VS_VS_RSV_IND State Transition

VS_VS_RSV_IND	On VS Set General Protocol command, set General Protocol Code. On VS Set Command Protocol command, update the corresponding Command Protocol Table entry. All commands with entries in the Command Protocol Table will be executed according to the Protocol Code in the corresponding Command Protocol entry. All other vendor-specific/reserved commands will be executed according to the General Protocol Code. All other commands will be executed according to the predefined protocol.		
1	Received VS Lock command	→	VS_LOCKED
2	Otherwise	→	VS_VS_RSV_IND

Protocols Summary

Table 44. Protocol Code Encoding Scheme

Protocol Code	Protocol	Codes Defined	Bit Assignment
0x00	Abort	0x00	
0x01–0x3F 0xA2–0xAF 0xB3–0xBF 0xE0–0xEF 0xF1–0xFF			Reserved
0x40–0x4F			Vendor Specific
0x80–0x8F 0xC0–0xCF (1x00xxxxb)	PIO Data in/Out	0x80, 0x81, 0x82, 0x87, 0x88, 0x89, 0x8A, 0x8B, 0x8F, 0xC0, 0xC2, 0xC8, 0xCA	Bit 6: 0 — legacy addressing 1 — 48-bit LBA addressing Bit 3: 0 — data in (read) 1 — data out (write) Bits 2-0: 0b000 — sector count is given by the Sector Count register. 0b001 — only one sector, Sector Count is ignored. 0b010 — blocks of multiple sectors, e.g., Read/Write Multiple. 0b011 — sector count is given by Sector Number and Sector Count registers, e.g. Download Microcode. 0b100–0b110 — reserved 0b111 — 512 plus vendor-specific bytes, e.g. Read/Write Long.
0x90–0x9F 0xD0–0xDF (1x01xxxxb)	DMA	0x90, 0x91, 0x98, 0x99, 0xD0, 0xD1, 0xD8, 0xD9	Bit 6: 0 — legacy addressing 1 — 48-bit LBA addressing Bit 3: 0 — data in (read) 1 — data out (write) Bits 2-1: 0b00 — currently defined 0b01–0b11 — reserved. Bit 0: 0 — not queued. 1 — queued.
0xA0	Packet	0xA0	
0xA1	Service	0xA1	
0xB0,0xF0 (1x110000b)	Non-Data	0xB0, 0xF0	Bit 6: 0 — legacy addressing 1 — 48-bit LBA addressing
0xB1	Execute Device Diagnostic	0xB1	
0xB2	Device Reset	0xB2	

Table 45. Vendor-Specific Protocol Code (in Alphabetical Order)

Protocol	Protocol Code	Description
Abort	0x00	Abort command. Status = 0x51 and Error = 0x04. Command will not be passed to downstream device(s).
Device Reset	0xB2	Device Reset protocol.
Execute Device Diagnostic	0xB1	Execute Device protocol (for host bridges arranged in master-slave configuration, both will respond regardless of the DEV bit in the Device register.
Non-Data	0xB0	Non-Data protocol.
Non-Data (Ext)	0xF0	Non-Data (Ext) protocol.
Packet	0xA0	Packet protocol.
PIO Data In (Read Multiple)	0x82	PIO Data In protocol for reading blocks of multiple sectors, e.g., Read Multiple.
PIO Data In (Read Multiple, Ext)	0xC2	PIO Data In protocol for reading blocks of multiple sectors for 48-bit LBA commands, e.g., Read Multiple Ext.
PIO Data In (Sectors)	0x80	PIO Data In protocol, sector count is given by the Sector Count register.
PIO Data In (Sectors, Ext)	0xC0	PIO Data In protocol for 48-bit LBA commands, sector count is given by the Sector Count register.
PIO Data In (Single Sector)	0x81	PIO Data In protocol, only one sector, Sector Count is ignored.
PIO Data Out (Download Microcode)	0x8B	PIO Data Out protocol, sector count is given by Sector Number and Sector Count registers.
PIO Data Out (Sectors)	0x88	PIO Data Out protocol, sector count is given by the Sector Count register.
PIO Data Out (Sectors, Ext)	0xC8	PIO Data Out protocol for 48-bit LBA commands, sector count is given by the Sector Count register.
PIO Data Out (Single Sector)	0x89	PIO Data Out protocol, only one sector, Sector Count is ignored.
PIO Data Out (Write Multiple)	0x8A	PIO Data Out protocol for writing blocks of multiple sectors, e.g., Write Multiple.
PIO Data Out (Write Multiple, Ext)	0xCA	PIO Data Out protocol for writing blocks of multiple sectors for 48-bit LBA commands, e.g., Write Multiple Ext
Read DMA	0x90	Read DMA protocol.
Read DMA (Ext)	0xD0	Read DMA protocol for 48-bit LBA commands.
Read DMA Queued	0x91	Read DMA Queued protocol.
Read DMA Queued (Ext)	0xD1	Read DMA Queued for 48-bit LBA commands.
Read Long	0x87	PIO Data In protocol, 512 plus vendor-specific bytes, e.g., Read Long.
Service	0xA1	Service protocol.
Write DMA	0x98	Write DMA protocol.
Write DMA (Ext)	0xD8	Write DMA protocol for 48-bit LBA commands.
Write DMA queued	0x99	Write DMA queued protocol.
Write DMA queued (Ext)	0xD9	Write DMA queued for 48-bit LBA commands.
Write Long	0x8F	PIO Data Out protocol, 512 plus vendor-specific bytes, e.g., Write Long

Table 46. Vendor-Specific Protocol Code (by Protocol Code)

Protocol Code	Protocol	Description
0x00	Abort	Abort command. Status = 0x51 and Error = 0x04. Command will not be passed to downstream device(s).
0x80	PIO Data In (Sectors)	PIO Data In protocol, sector count is given by the Sector Count register.
0x81	PIO Data In (Single Sector)	PIO Data In protocol, only one sector, Sector Count is ignored.
0x82	PIO Data In (Read Multiple)	PIO Data In protocol for reading blocks of multiple sectors, e.g., Read Multiple.
0x87	Read Long	PIO Data In protocol, 512 plus vendor-specific bytes, e.g., Read Long.
0x88	PIO Data Out (Sectors)	PIO Data Out protocol, sector count is given by the Sector Count register.
0x89	PIO Data Out (Single Sector)	PIO Data Out protocol, only one sector, Sector Count is ignored.
0x8A	PIO Data Out (Write Multiple)	PIO Data Out protocol for writing blocks of multiple sectors, e.g., Write Multiple.
0x8B	PIO Data Out (Download Microcode)	PIO Data Out protocol, sector count is given by Sector Number and Sector Count registers.
0x8F	Write Long	PIO Data Out protocol, 512 plus vendor-specific bytes, e.g., Write Long
0x90	Read DMA	Read DMA protocol.
0x91	Read DMA Queued	Read DMA Queued protocol.
0x98	Write DMA	Write DMA protocol.
0x99	Write DMA queued	Write DMA queued protocol.
0xA0	Packet	Packet protocol.
0xA1	Service	Service protocol.
0xB0	Non-Data	Non-Data protocol.
0xB1	Execute Device Diagnostic	Execute Device protocol (for host bridges arranged in master-slave configuration, both will respond regardless of the DEV bit in the Device register.
0xB2	Device Reset	Device Reset protocol.
0xC0	PIO Data In (Sectors, Ext)	PIO Data In protocol for 48-bit LBA commands, sector count is given by the Sector Count register.
0xC2	PIO Data In (Read Multiple, Ext)	PIO Data In protocol for reading blocks of multiple sectors for 48-bit LBA commands, e.g., Read Multiple Ext.
0xC8	PIO Data Out (Sectors, Ext)	PIO Data Out protocol for 48-bit LBA commands, sector count is given by the Sector Count register.
0xCA	PIO Data Out (Write Multiple, Ext)	PIO Data Out protocol for writing blocks of multiple sectors for 48-bit LBA commands, e.g., Write Multiple Ext
0xD0	Read DMA (Ext)	Read DMA protocol for 48-bit LBA commands.
0xD1	Read DMA Queued (Ext)	Read DMA Queued for 48-bit LBA commands.
0xD8	Write DMA (Ext)	Write DMA protocol for 48-bit LBA commands.
0xD9	Write DMA queued (Ext)	Write DMA queued for 48-bit LBA commands.
0xF0	Non-Data (Ext)	Non-Data (Ext) protocol.

Reading and Writing of Task File and Device Control Registers

48-bit LBA Addressing

The SiI3512 supports 48-bit LBA. The SiI3512 does not differentiate a non-extended command (one that does not use 48-bit LBA address) from an extended command (one that uses the 48-bit LBA address). The "expanded" registers can be read with the HOB bit of the Device Control register se to '1'.

Device Control Register and Soft Reset

When the Device Control register is written, a Register FIS for Control will be sent downstream upon one of the following conditions:

- There is a change in the SRST bit, or;
- With SRST bit being '0', there is a change in the NIEN bit.

Note that:

- When the SRST is '1', the NIEN bit in the Register FIS sent is insignificant.
- Any change in the HOB bit will not initiate any Register FIS to be sent. In fact, HOB bit is always '0' in the Register FIS sent.
- If the Serial ATA channel is in PARTIAL or SLUMBER state, a COMWAKE will be automatically initiated to wake up the channel before the Register FIS is sent. However, the channel will stay at the ON state at the end of the operation, even if no soft reset occurs.

A soft reset will do the following:

- Wake up the downstream Serial ATA device from ATA IDLE, STANDBY or SLEEP.

LED Support

The SiI3512 supports two activity LEDs via two 12-mA open-drain drivers, LED[0..1]. LED0 indicates activity in IDE Channel 0 and LED1 indicates activity in IDE Channel 1.

When there is activity for a non-ATAPI device, as indicated by BSY in the ATA Status being set, or when any bit in the Serial ATA SActive register is set, the corresponding LED driver outputs will be driven low.

There is no activity LED support for ATAPI device. If the downstream device is an ATAPI device, the corresponding LED output will not be driven low.

Flash and EEPROM Programming Sequences

Flash Memory Access

The SiI3512 supports an external flash memory device of up to 4-Mbit capacity. Access to the flash memory is available through two means: PCI Direct Access and Register Access.

PCI Direct Access

Access to the Expansion ROM is enabled by setting bit 0 in the Expansion ROM Base Address register at offset 0x30 of the PCI Configuration Space. When this bit is set, bits [31:19] of the same register are programmable by the system to set the base address for all flash memory accesses. Read and write operations with the flash memory are initiated by Memory Read and Memory Write commands on the PCI bus. Accesses may be as bytes, words, or dwords.

Register Access

This type of flash memory access is carried out through a sequence of internal register read and write operations. The proper programming sequences are detailed below.

Flash Write Operation

Verify that bit 25 is cleared in the register at Offset 0x50 of Base Address 5. The bit reads one when a memory access is currently in progress.

It reads zero when the memory access is complete and ready for another operation.

Program the write address for the flash memory access. The address field is defined by bits [18:00] in the Flash Memory Address – Command + Status register.

Program the write data for the flash memory access. The data field is defined by bits [07:00] in the Flash Memory Data register at Offset 54 of Base Address 5.

Program the memory access type. The memory access type is defined by bit 24 in the Flash Memory Address – Command + Status register. The bit must be cleared for a memory write access.

Initiate the flash memory access by setting bit 25 in the Flash Memory Address – Command + Status register.

Flash Read Operation

Verify that bit 25 is cleared in the Flash Memory Address – Command + Status register at Offset 0x50 of Base Address 5. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete and ready for another operation.

Program the read address for the flash memory access. The address field is defined by bits [18:00] in the Flash Memory Address – Command + Status register.

Program the memory access type. The memory access type is defined by bit 24 in the Flash Memory Address – Command + Status register. The bit must be set for a memory read access.

Initiate the flash memory access by setting bit 25 in the Flash Memory Address – Command + Status register.

Verify that bit 25 is cleared in the Flash Memory Address – Command + Status register. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete.

Read the data from the flash memory access. The data field is defined by bits [07:00] in the Flash Memory Data register at Offset 0x54 of Base Address 5.

EEPROM Memory Access

The SiI3512 supports an external 256-byte EEPROM memory device. Access to the EEPROM memory is available through internal register operations in the SiI3512.

EEPROM Write Operation

Verify that bit 25 is cleared in the EEPROM Memory Address – Command + Status register at Offset 0x58 of Base Address 5. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete and ready for another operation.

Write '1' to clear bit 28 in the EEPROM Memory Address – Command + Status register. The bit is set if an error occurred during a previous memory access.

Program the write address for the EEPROM memory access. The address field is defined by bits [07:00] in the EEPROM Memory Address – Command + Status register. Program bits [15:08] to zero.

Program the write data for the EEPROM memory access. The data field is defined by bits [07:00] in the EEPROM Memory Data register at Offset 0x5C of Base Address 5.

Program the memory access type. The memory access type is defined by bit 24 in the EEPROM Memory Address – Command + Status register. The bit must be cleared for a memory write access.

Initiate the EEPROM memory access by setting bit 25 in the EEPROM Memory Address – Command + Status register.

Poll bit 25 in the EEPROM Memory Address – Command + Status register. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete.

Check bit 28 in the EEPROM Memory Address – Command + Status register. The bit is set if an error occurred during a previous memory access.

EEPROM Read Operation

Verify that bit 25 is cleared in the EEPROM Memory Address – Command + Status register at offset 0x58 of Base Address 5. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete and ready for another operation.

Write '1' to clear bit 28 in the EEPROM Memory Address – Command + Status register. The bit is set if an error occurred during a previous memory access.

Program the read address for the EEPROM memory access. The address field is defined by bits [07:00] in the EEPROM Memory Address – Command + Status register. Program bits [15:08] to zero.

Program the memory access type. The memory access type is defined by bit 24 in the EEPROM Memory Address – Command + Status register. The bit must be set for a memory read access.

Initiate the EEPROM memory access by setting bit 25 in the EEPROM Memory Address – Command + Status register.

Poll bit 25 in the EEPROM Memory Address – Command + Status register. The bit reads one when a memory access is currently in progress. It reads zero when the memory access is complete.

Check bit 28 in the EEPROM Memory Address – Command + Status register. The bit is set if an error occurred during a previous memory access.

Read the data from the EEPROM memory access. The data field is defined by bits [07:00] in the EEPROM Memory Data register at Offset 0x5C of Base Address 5.

Appendix: SiI3512 NAND Tree

A NAND tree is available in the SiI3512 for functional testing of input pins. The NAND tree test mode can be invoked by setting the following signals in the SiI3512:

Table 47. Signal Settings to Invoke NAND Tree Test Mode

Pin Name	Pin Number	Logic State
TEST_MODE	57	1
FL_ADDR[18]	56	0
FL_ADDR[17]	55	0
FL_ADDR[16]	54	1
FL_ADDR[15]	53	0

The NAND tree is shown in the following diagram:

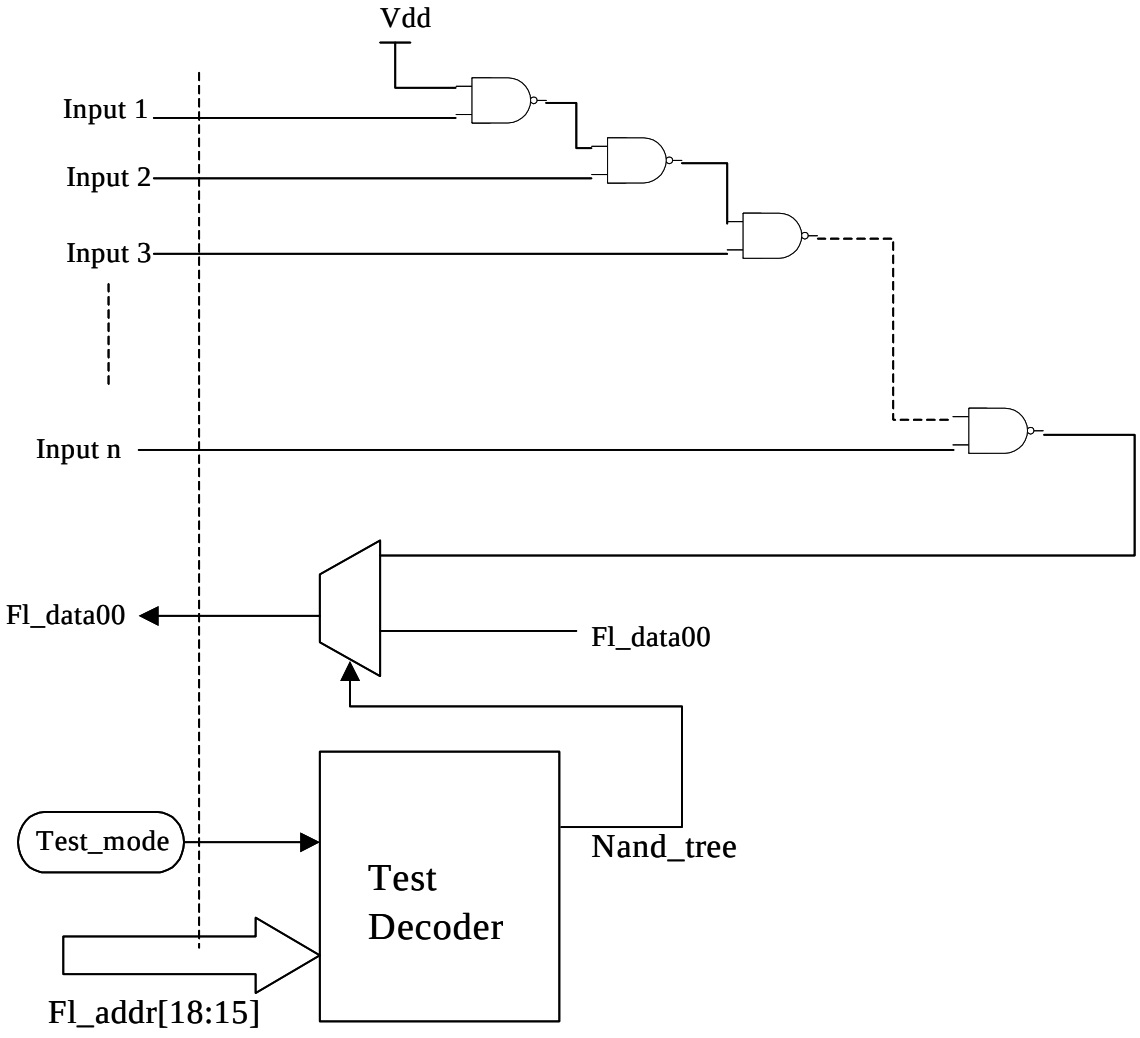


Figure 15. Input Pin Testing NAND Tree

The test decoder generates the NAND tree signal. This signal performs the following functions:

1) Drives the enable pins of all the bi-directional I/Os and sets them in input mode. The only exception is the FL_DATA00 pin, which is used as the output for the NAND tree.

2) Drives the output multiplexer to direct the output of the NAND tree to the FL_DATA00 pin.

All the input cells and bi-directional pins (except FL_DATA00) are connected to the NAND tree. The following table lists the order of inputs by pin name and number.

Table 48. Order of Inputs to NAND Tree

Order	Pin Number	Pin Name
1	30	MEM_CS_N
2	1	PCI_AD00
3	128	PCI_AD01
4	127	PCI_AD02
5	126	PCI_AD03
6	125	PCI_AD04
7	124	PCI_AD05
8	121	PCI_AD06
9	120	PCI_AD07
10	119	PCI_CBE0
11	118	PCI_AD08
12	117	PCI_AD09
13	116	PCI_M66EN
14	115	PCI_AD10
15	114	PCI_AD11
16	111	PCI_AD12
17	110	PCI_AD13
18	109	PCI_AD14
19	108	PCI_AD15
20	105	PCI_CBE1
21	104	PCI_PAR
22	102	PCI_TRDY_N
23	101	PCI_DEVSEL_N
24	100	PCI_STOP_N
25	99	PCI_PERR_N
26	98	PCI_IRDY_N
27	97	PCI_FRAME_N
28	96	PCI_CBE2
29	95	PCI_AD16
30	94	PCI_AD17
31	93	PCI_AD18
32	90	PCI_AD19
33	89	PCI_AD20
34	88	PCI_AD21
35	87	PCI_AD22
36	86	PCI_AD23
37	85	PCI_IDSEL
38	84	PCI_CBE3
39	83	PCI_AD24
40	82	PCI_AD25
41	79	PCI_AD26
42	78	PCI_AD27
43	77	PCI_AD28
44	74	PCI_AD29
45	73	PCI_AD30
46	72	PCI_AD31
47	70	PCI_GNT_N

Order	Pin Number	Pin Name
48	69	PCI_CLK
49	68	PCI_RST_N
50	65	FL_DATA[07]
51	64	FL_DATA[06]
52	63	FL_DATA[05]
53	62	FL_DATA[04]
54	61	FL_DATA[03]
55	60	FL_DATA[02]
56	59	FL_DATA[01]
57	52	FL_ADDR[14]
58	50	FL_ADDR[13]
59	49	FL_ADDR[12]
60	46	FL_ADDR[11]
61	45	FL_ADDR[10]
62	44	FL_ADDR[09]
63	43	FL_ADDR[08]
64	42	FL_ADDR[07]
65	39	FL_ADDR[06]
66	38	FL_ADDR[05]
67	37	FL_ADDR[04]
68	36	FL_ADDR[03]
69	35	FL_RD_N
70	34	FL_WR_N
71	33	FL_ADDR[02]
72	32	FL_ADDR[01] / BA5_EN
73	31	FL_ADDR[00] / IDE_CFG
74	3	EEPROM_SCLK
75	2	EEPROM_SDAT

Disclaimers

These materials are provided on an "AS IS" basis. Silicon Image, Inc. and its affiliates disclaim all representations and warranties (express, implied, statutory or otherwise), including but not limited to: (i) all implied warranties of merchantability, fitness for a particular purpose, and/or non-infringement of third party rights; (ii) all warranties arising out of course-of-dealing, usage, and/or trade; and (iii) all warranties that the information or results provided in, or that may be obtained from use of, the materials are accurate, reliable, complete, up-to-date, or produce specific outcomes. Silicon Image, Inc. and its affiliates assume no liability or responsibility for any errors or omissions in these materials, makes no commitment or warranty to correct any such errors or omissions or update or keep current the information contained in these materials, and expressly disclaims all direct, indirect, special, incidental, consequential, reliance and punitive damages, including WITHOUT LIMITATION any loss of profits arising out of your access to, use or interpretation of, or actions taken or not taken based on the content of these materials.

Silicon Image, Inc. and its affiliates reserve the right, without notice, to periodically modify the information in these materials, and to add to, delete, and/or change any of this information.

Notwithstanding the foregoing, these materials shall not, in the absence of authorization under U.S. and local law and regulations, as required, be used by or exported or re-exported to (i) any U.S. sanctioned or embargoed country, or to nationals or residents of such countries; or (ii) any person, entity, organization or other party identified on the U.S. Department of Commerce's Denied Persons or Entity List, the U.S. Department of Treasury's Specially Designated Nationals or Blocked Persons List, or the Department of State's Debarred Parties List, as published and revised from time to time; (iii) any party engaged in nuclear, chemical/biological weapons or missile proliferation activities; or (iv) any party for use in the design, development, or production of rocket systems or unmanned air vehicles.

Products and Services

The products and services described in these materials, and any other information, services, designs, know-how and/or products provided by Silicon Image, Inc. and/or its affiliates are provided on as "AS IS" basis, except to the extent that Silicon Image, Inc. and/or its affiliates provides an applicable written limited warranty in its standard form license agreements, standard Terms and Conditions of Sale and Service or its other applicable standard form agreements, in which case such limited warranty shall apply and shall govern in lieu of all other warranties (express, statutory, or implied). EXCEPT FOR SUCH LIMITED WARRANTY, SILICON IMAGE, INC. AND ITS AFFILIATES DISCLAIM ALL REPRESENTATIONS AND WARRANTIES (EXPRESS, IMPLIED, STATUTORY OR OTHERWISE), REGARDING THE INFORMATION, SERVICES, DESIGNS, KNOW-HOW AND PRODUCTS PROVIDED BY SILICON IMAGE, INC. AND/OR ITS AFFILIATES, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND/OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS. YOU ACKNOWLEDGE AND AGREE THAT SUCH INFORMATION, SERVICES, DESIGNS, KNOW-HOW AND PRODUCTS HAVE NOT BEEN DESIGNED, TESTED, OR MANUFACTURED FOR USE OR RESALE IN SYSTEMS WHERE THE FAILURE, MALFUNCTION, OR ANY INACCURACY OF THESE ITEMS CARRIES A RISK OF DEATH OR SERIOUS BODILY INJURY, INCLUDING, BUT NOT LIMITED TO, USE IN NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION, EMERGENCY SYSTEMS, OR OTHER SYSTEMS WITH A SIMILAR DEGREE OF POTENTIAL HAZARD. NO PERSON IS AUTHORIZED TO MAKE ANY OTHER WARRANTY OR REPRESENTATION CONCERNING THE PERFORMANCE OF THE INFORMATION, PRODUCTS, KNOW-HOW, DESIGNS OR SERVICES OTHER THAN AS PROVIDED IN THESE TERMS AND CONDITIONS.

1060 E. Arques Avenue
Sunnyvale, CA 94085
T 408.616.4000 F 408.830.9530
www.siliconimage.com

1060 E. Arques Avenue
Sunnyvale, CA 94085
T 408.616.4000 F 408.830.9530
www.siliconimage.com