SIS1100/3100
Standard design
Firmware Version 5/6
User Manual

SIS GmbH
Harksheider Str. 102A
22399 Hamburg
Germany

Phone: ++49 (0) 40 60 87 305 0
Fax: ++49 (0) 40 60 87 305 20

email: info@struck.de
http://www.struck.de

Version: sis3100-M-1-1-v135-hardware as of 27.03.08

Revision Table:

Revision	Date	Modification
0.0	19.01.01	Generation
0.1	13.07.01	mki touch up
1.0	26.10.01	Initial release
1.1	05.11.01	- OPT-IN/OUT register - OPT-IN register - OPT-VME-Interrupt register - Doorbell register
1.11	26.11.01	- VME SLAVE register description
1.20	29.01.01	SIS9200 SHARC option registers added
1.21	04.06.02	Modifications for firmware V_310502
1.30	17.08.02	Firmware version 5 VME interrupt generation added
1.31	23.04.03	Multi master section added
1.32	16.12.03	3U cPCI front panel added
1.33	24.02.04	bug fix in texts of IRQ generation register add auto copy address range to SHARC address map
1.34	18.12.05	changed photograph, ESD note, BERR timeout bug fix
1.35	28.03.08	Add error code table from LINUX driver Typo fix in register 0x104

1 Table of contents

1 Table of contents ... 3
2 Introduction ... 5
3 Overview ... 6
3.1 Design Concept of VME side ... 7
3.1.1 I/O Option ... 8
3.1.2 DSP option ... 8
3.1.3 SDRAM option ... 8
3.2 Design concept of PCI side (SIS1100) ... 9
3.2.1 I/O Option ... 9
4 Gigabit Hardware ... 10
5 Gigabit-Link Transfer Protocol ... 10
5.1 Protocol Integrity ... 10
5.2 General Transfer Protocol ... 11
5.3 Protocol Header, special word ... 12
5.4 VME Access ... 14
5.4.1 VME D08 and BLT8 Access ... 14
5.4.2 VME D16 and BLT16 Access ... 14
5.4.3 VME D32 and BLT32 Access ... 15
5.4.4 VME BLT64 Access ... 15
5.5 Remote Register Transfer Protocol ... 17
5.5.1 Write remote register ... 17
5.5.2 Read remote register ... 17
5.6 Direct VME Bus Access Transfer Protocol ... 18
5.6.1 Single Word Write Direct VME Bus Access ... 18
5.6.2 Single Word Read Direct VME Bus Access ... 19
5.6.3 Block transfer Write Direct VME Bus Access ... 20
5.6.4 Block transfer Read Direct VME Bus Access ... 20
5.7 Mapped VME Bus Access Transfer Protocol ... 21
5.7.1 Single Word Write mapped VME Bus Access ... 21
5.7.2 Single Word Read mapped VME Bus Access ... 21
5.7.3 Block transfer Write mapped VME Bus Access ... 22
5.7.4 Blocktransfer Read mapped VME Bus Access ... 22
5.8 SDRAM Transfer Protocol ... 23
5.8.1 Single Word Write SDRAM ... 23
5.8.2 Single Word Read SDRAM ... 23
5.8.3 Blocktransfer Write SDRAM ... 24
5.8.4 Blocktransfer Read SDRAM ... 24
6 SIS3100 Access through the Optical Interface ... 25
6.1 Control register space ... 25
6.1.1 Type-Identifier/Version register(0x0, read) ... 26
6.1.2 Optical status register (0x4, r/w) ... 27
6.1.3 Optical control register (0x8, r/w) ... 28
6.1.4 OPT-IN/OUT Register (0x80,read /write) ... 29
6.1.5 OPT-IN-LATCH_IRQ Register (0x84,read /write) ... 30
6.1.6 OPT-VME-Master Status/Control register (0x100,read/write) ... 31
6.1.7 OPT-VME Interrupt Status/Control register (0x104,read /write) ... 33
6.1.8 VME interrupt register (0x110, r/w) ... 36
6.1.9 OPT-VME-Slave Status/Control register (0x200,read /write) ... 37
6.1.10 OPT-VME-Master DMA_WRITE_BYTE_counter register (0x204,read) ... 38
6.1.11 OPT-DSP Status/Control Register (0x300, r/w) ... 39
6.1.12 OPT-VME-Address MAP register 0..255 (0x400..0x7FC,read/write) ... 40
6.2 OPT-Sharc space ... 41
6.3 OPT-SDRAM space ... 42
6.3.1 OPT_SDRAM_SPD register ... 43
7 VME slave access ... 44
7.1 VME Slave Address Map ... 44
7.2 Type-Identifier/Version register(0x0, read) ... 45

7.3 VS-DSP Status/Control Register (0x10, r/w) 46
7.4 SDRAM Page Register (0x14, r/w) 47
8 VME side LEDs 48
8.1 Front panel LEDs 48
8.1.1 Explanation of front panel LEDs 48
8.1.2 Special case: LED Link up off 49
8.2 PCB LEDs 49
9 SIS3100 Jumpers 50
9.1 J10 50
9.2 J90 51
9.2.1 JP_DSP 51
10 VME master/system controller 52
10.1 Multi master operation 52
10.1.1 System Controller 52
10.1.2 Bus grant/bus mastership 52
11 SHARC resources 53
11.1 SHARC address map 53
11.1.1 SHARC Local Bus interface registers 54
11.1.2 Address map of SHARC registers 58
11.1.3 Address map of SHARC-Optical Dual Ported Ram 58
11.2 SHARC flags 66
11.3 SHARC DSP interrupts 67
11.3.1 DSP Interrupt sources 67
11.3.2 DSP Interrupt configuration 67
12 SIS1100 Hardware Description 68
12.1 PCI Front panel 68
12.2 3U cPCI front panel 69
12.3 SIS1100 LEDs 69
12.3.1 Front panel LEDs 69
12.3.2 SIS1100-OPT PCB LEDs 70
12.3.3 SIS1100-CMC PCB LEDs 70
12.4 SIS1100 Input termination 71
13 Appendix 72
13.1 Error codes 72
13.2 Power consumption 73
13.3 I/O option Jumper description 74
13.3.1 JP710 74
13.3.2 JP770 74
13.3.3 Schematic of flat I/O connector 75
13.3.4 Schematic of LEMO I/O section 76
13.4 Boot mechanisms 77
13.4.1 ISP PROM 77
13.4.2 JTAG 77
13.5 Connector types 78
14 Index 79

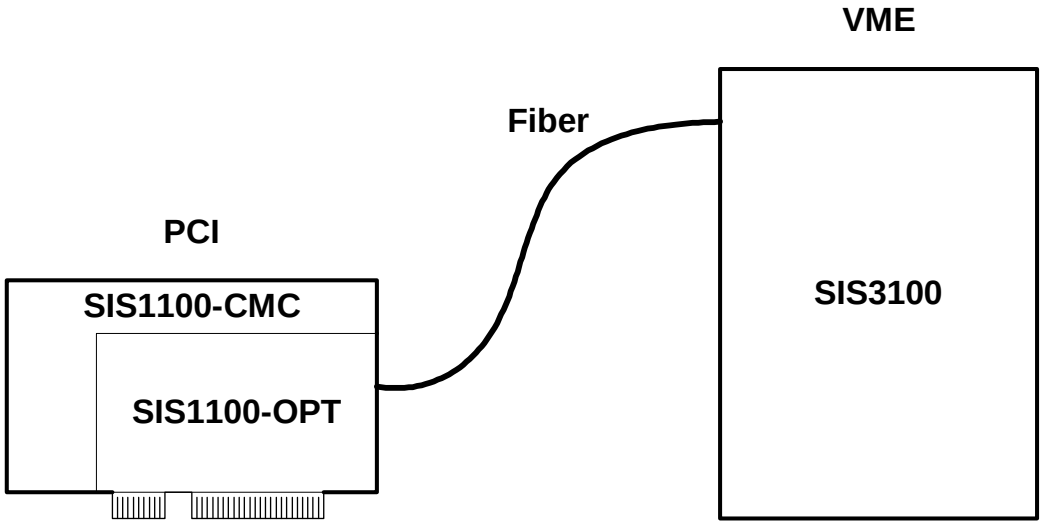
2 Introduction

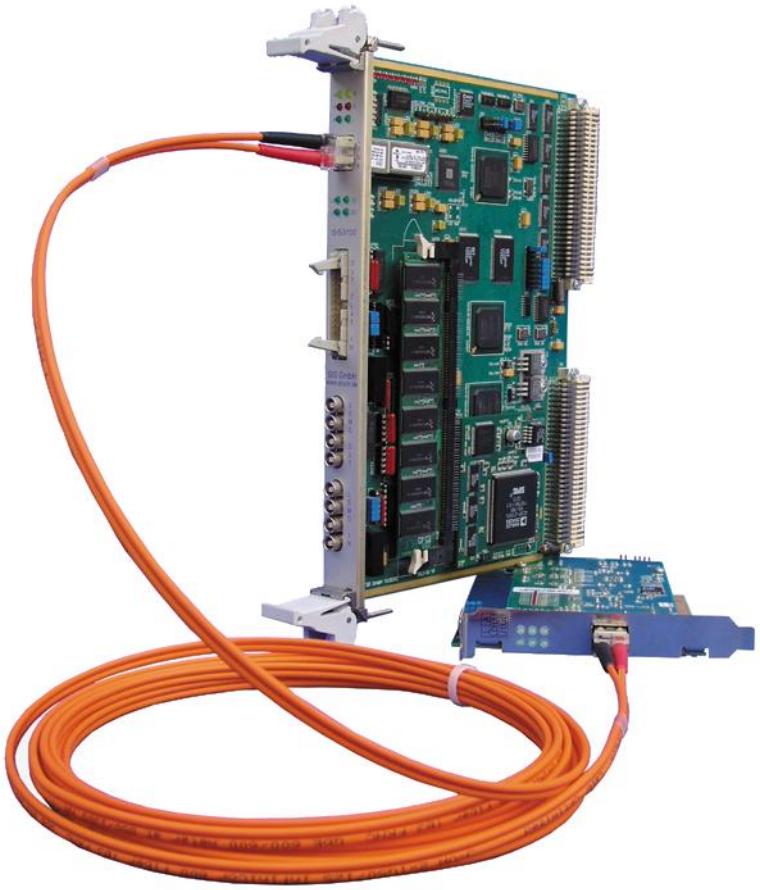
The goal of the project was the development of a high performance VME list sequencer to PCI interface, which was tailored to match the requirements of Particle Physics experiments, related applications and other demanding data acquisition systems. The maximum anticipated data rate on the VME side, required medium to long link distances in large scale setups and the wish for electrical decoupling resulted in the selection of a fibre optic Gigabit solution as the interconnecting technology.
As we are aware, that no manual is perfect, we appreciate your feedback and will try to incorporate proposed changes and corrections as quickly as possible. The most recent version of this manual can be obtained by email from info@struck.de, the revision dates are online under http://www.struck.de/manuals.htm. A list of available firmware designs can be retrieved from http://www.struck.de/sis3100firm.htm

Note: The SIS1100/3100 PCI to VME interface was developed in a collaborative effort between the ZEL department of the Research Center Jülich and SIS GmbH.

3 Overview

The SIS PCI to VME interface consists of the SIS1100 PCI card and the SIS3100 VME list sequencer and an interconnecting link fibre. The SIS1100 card is divided into the SIS1100-CMC PCI CMC (common mezzanine card) and the SIS1100-OPT gigabit link CMC card.





Photograph of SIS1100 and SIS3100 with I/O, DSP and SDRAM option

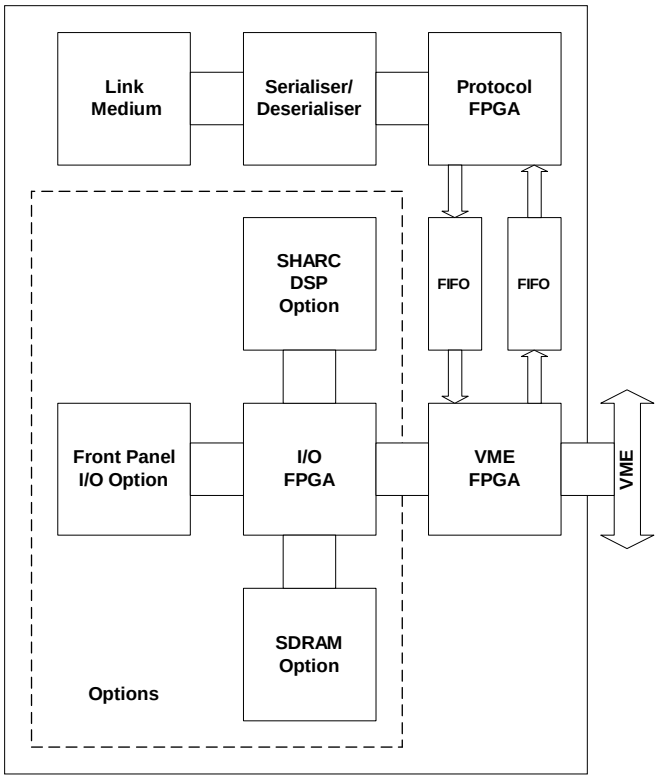
3.1 Design Concept of VME side

The VME side of the PCI to VME interface, the SIS3100, is a modular design, that can be configured for the given application.

Find below a list of key features of the SIS3100.

- VME List sequencer
- Mapping table with 256 entries
- VME Master: A16/A24/A32/A40 D8/D16/D32/BLT32/MBLT64/2eMBLT64
- VME Slave: A32/D32/BLT32/MBLT64
- Block transfer address auto increment on/off (for FIFO reads)
- System controller function (can be disabled by jumper)
- Hot swap (in conjunction with VME64x backplane)
- VME64x Connectors
- VME64x Side Shielding
- VME64x Front panel
- VME64x extractor handles (on request)
- 10 front panel and 8 PCB LEDs
- single supply (+5 V)

A block diagram of the SIS3100 is shown below.



SIS3100 block diagram

3.1.1 I/O Option

The I/O option features extended data handling and input/output functionality. It comprises the I/O FPGA, connectors for the optional SIS9200 SHARC DSP , the DIMM socket and the front panel input/output hardware, which features:

- 4 flat cable inputs (ECL or TTL)
- 4 flat cable output (ECL or TTL)
- 3 LEMO inputs (NIM or TTL)
- 3 LEMO outputs (NIM or TTL)
- 1 LEMO reset input (NIM or TTL)
- 1 LEMO reset output (NIM or TTL)

The inputs can be used for conditional VME sequencer control and the outputs can be set/cleared under sequencer control for interaction with external dead time or other logic. This results in a substantial increase in performance compared to the use of an external VME I/O register, as no VME cycle (leave alone interrupt cycle) is involved.

3.1.2 DSP option

The DSP option comprises a SIS9200 SHARC DSP piggy back board with SHARC links (i.e. ADSP21062L chip) for histogramming or higher level trigger applications.

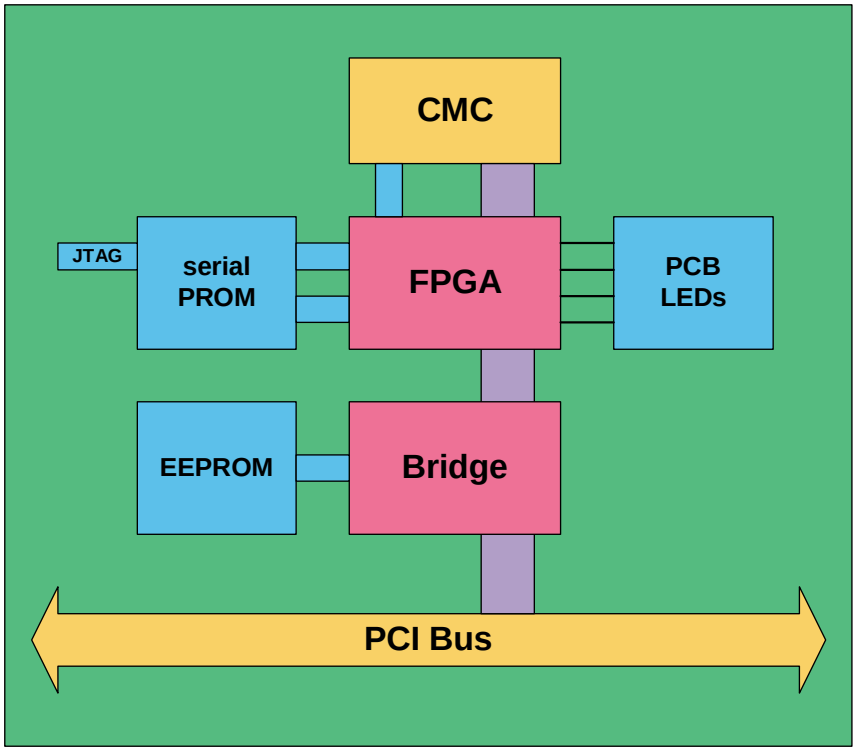
3.1.3 SDRAM option

The SDRAM option consists of the DRAM controller firmware and a SDRAM memory strip, which is available in 64, 128, 256 and 512 MByte

3.2 Design concept of PCI side (SIS1100)

The SIS1100 PCI Gigabit link card was developed to act as PCI target and initiator to allow for the use of PCI block transfer cycles. A PLX9054 PCI to local bus interface, which is PCI 2.1 and 2.2 compliant is used as the interfacing hardware. The SIS1100 is subdivided into the SIS1100-CMC, which is a CMC carrier board, and the SIS1100-OPT, which is a CMC Gigabit link board..

- PCI 2.1 and 2.2 compliant
- PLX9054 PCI master bridge chip
- Protocol FPGA
- serial configuration PROMs for FPGA and PLX bridge
- JTAG port to FPGA PROM and FPGA
- CMC (IEEE P1386 Draft 2.3) single size carrier
- all CMC data lines routed to FPGA
- 4 SMD LEDs routed to FPGA



Block diagram of SIS1100-CMC

3.2.1 I/O Option

The I/O option for the SIS1100-CMC, which can be used on the SIS1100 board was introduced to allow for direct interaction with the readout PC. The input/output option features:

- 2 LEMO inputs (TTL)
- 2 LEMO outputs (TTL)

4 Gigabit Hardware

Small form factor (SFF) Gigabit link media were chosen as the physical layer of the link of the SIS PCI to VME interface. Media with LC connectors are used, patch fibres to other standards like ST are readily available for large scale connections through 19" patch fields. The link is clocked at 125 MHz (i.e. a 62.5 MHz clock is doubled by a delay locked loop in the protocol XILINX FPGA), what results in a payload of 125 Mbytes/s. With the standard multimode link media distances of up to 450 m can be covered, single mode media and fibres extend the range up to 80 km. Due to pipelining single cycle and high speed block transfer capabilities link latency will not play a significant role for most applications even at very long distances.

5 Gigabit-Link Transfer Protocol

32-bit words are transmitted over the Gigabit-Link. The link hardware is in charge of proper structure. Loss of synchronisation or errors are reported to the corresponding host. Any transmission starts with a special word, in the case of a block transfer it can end with a special character also if the transfer length is undefined or smaller than the requested length.

One bit in the FIFOs is used to flag a special word, with Byte 0 being 0x1C (SC_PROT K28.0).
Byte loss can be detected as all 4 Bytes of a word are transmitted without interruption. All characters up to the next special character are ignored if the data link layer detects an error.

5.1 Protocol Integrity

A protocol sequence has to be transmitted without interruption, i.e. a mixture of request and confirmation protocols is not allowed.

5.2 General Transfer Protocol

The protocol structure depends on the protocol header (special word).
In general the transfer protocol structure is as described in the table below.

	Bit[31:24] phys. Byte 0	**Bit[23:16] Byte 1**	**Bit[15:8] Byte 2**	**Bit[7:0] Byte 3**	
special word	SC_PROT Special Char	ctl: control	sp: space	be: byte enable	
AM	Address Modifier Bit 15:0				direct VME access only
ADDR_H	Address A63-32				64 Bit only
ADDR_L	Address A31-0				
DATA_H	Data D63-32 (register contents swapped)				64 Bit only
DATA_L/BC	Data D31-0/Byte count with BT Read				
DATA	consecutive data word(s)				Blockt ransfer Write only
...					
special word	SC_PROT	ctl: **END**			

The individual 32-bit words are transmitted beginning with Byte 0, a protocol sequence starts with special character SC_PROT always. Hence this Byte is transmitted as the first Byte of the *special words*. As mentioned above the value is 0x1C (SC_PROT, K28.0).

5.3 Protocol Header, special word

The special word is formed as illustrated in the table below

Bit	Byte	Bit	Comment
31-24 FF000000		SC_PROT 0x1C always	
23 00800000	**CTL** (*Control*)	**EOT** DMA end	To be set in protocol **END** only
22 00400000		**FIFO** no address increment	With **BT** (block transfer) only
21 00200000		**BT** block transfer	The address is followed by the byte count (1 word) in a read request.
20 00100000		**A64** 64/bit address	with request protocol only
19 00080000		**AM** address modifier contained	An AM can be present in a request protocol only
18 00040000		**WR** write request	0: read 1: write
17-16 00030000		00 **REQ** 01 **END** 10 **CON** 11 **ECON**	Request, protocol start End of block transfer Confirmation, positive confirmation Error confirmation
15-14 0000C000	**SP** Space, to be returned unchanged	0: normal transfer 1: Buffer pipe 2: DMA0 pipe 3: reserved	Local space, to be used for pipelined read **Not used with SIS3100**
13-8 00003F00		0: register 1: direct VME access 2-4: not used 6: SDRAM, DSP 7-13: not used	remote space: interpreted by SIS3100
7-0 000000FF	**BE** (request) **EC** (confirm)	01 Byte 0 02 Byte 1 04 Byte 2 08 Byte 3 F0 Byte 7-4	Byte enable is ignored during a register transfer, as a 32-bit word is transmitted the value should be 0x0F however. Two data words (64-bit) are expected if enable of Byte 7-4 is different from 0 This byte holds the error information on error confirmation (**ECON**)

Byte Enable Bits Summary

Byte Enable	Combination	Transfer length
00	invalid	
01	valid	byte
02	valid	byte
03	valid	double byte
04	valid	byte
05	invalid	
06	invalid	
07	invalid	
08	valid	byte
09	invalid	
0A	invalid	
0B	invalid	
0C	valid	double byte
0D	invalid	
0E	invalid	
0F	valid	quad byte
10	valid	eight byte transfer
..	valid	eight byte transfer
FF	valid	eight byte transfer

NOTE: The VME access width (D8/D16/D32/D64) is defined by the Byte enable bits

5.4 VME Access

5.4.1 VME D08 and BLT8 Access

During VME single byte transfers (D08 und BLT8) one valid Byte only is transferred over the optical link per 32-bit word.

Assignment of Byte Enable Bits

Adress End Bits (a1 a0)	Byte Enable BE	PCI data bits Little Endian valid bits	Optical data bits Big Endian valid bits	VME data bus valid bits	VME DS1*	VME DS0*	VME A1	VME A2	VME Lword *
..00	01	[7:0]	[31:24]	[15:8]	low	high	0 (a1)	a2	high
..01	02	[15:8]	[23:16]	[7:0]	high	low	0 (a1)	a2	high
..10	04	[23:16]	[15:8]	[15:8]	low	high	1 (a1)	a2	high
..11	08	[31:24]	[7:0]	[7:0]	high	low	1 (a1)	a2	high

* low active

During a block transfer (BT) 8-bit per 32-bit word are transferred also, the start address a1, a0 defines which data bits are valid during the first data word. During consecutive Bytes the data bits will become valid in following order: : .... [31:24], [23:16], [15:8], [7:0] , [31:24], ...

5.4.2 VME D16 and BLT16 Access

During a VME double byte transfer (D16 and BLT16) two valid Bytes are transferred over the optical link per 32-bit word.

Assignment Byte Enable Bits

Address End Bits (a1 a0)	Byte Enable	Little Endian	Big Endian (Optical data bits)	VME data bus	VME DS1*	VME DS1*	VME A1	VME A2	VME Lword*
..00	03	[7:0], [15:8]	[31:24], [23:16]	[15:0]	low	low	0 (a1)	a2	high
..10	0C	[23:16], [31:24]	[15:8], [7:0]	[15:0]	low	low	1 (a1)	a2	high

Rule: a0 must be 0 if Byte Enable = 03 or 0C

The start address a1 defines which data bits of the first data word are valid. During consecutive double Bytes the data bits will become valid in following order: [31:16], [15:0], [31:16], [15:0], ...

5.4.3 VME D32 and BLT32 Access

All 32-bits on the optical data path are valid during a VME quad byte transfer (D32 and BLT32).

Assignment of Byte Enable Bits

Address End Bits (a1 a0)	Byte Enable	Little Endian	Big Endian (Optical data bits)	VME data bus	VME DS1*	VME DS1*	VME A1	VME A2	VME Lword*
..00	0F	[7:0], [15:8], [23:16], [31:24]	[31:24], [23:16], [15:8], [7:0]	D[31:0]	low	low	0 (a1)	a2	low

Rule: a1, a0 must be 00 if Byte Enable = 0x0F

5.4.4 VME BLT64 Access

Two 32-bit data words are transmitted over the optical link during a multiplexed VME eight byte block transfer (MBLT64).

Assignment of Byte Enable Bits

Address End Bits (a2, a1 a0)	Byte Enable	Little Endian	Big Endian (Optical data bits)	VME data bus	VME DS1*	VME DS1*
..000	10..ff	[7:0], [15:8], [23:16], [31:24]	1. datum [63:56], [55:48], [47:40], [39:32]	A[31:1], LOWRD*	low	low
		[39:32], [47:40], [55:48], [63:56]	2. datum [31:24], [23:16], [15:8], [7:0]	D[31:0]		

Rule: a2, a1, a0 must be 000 if Byte Enable = BLT64

AM Protocol word

Function of AM Protocol Bits on VME Cycle

AM Protocol	15	14	11	10	9	8	7	[7:6]	[5:0]
VME		IRQ ACK Cycle							AM [5:0]

* low active

5.5 Remote Register Transfer Protocol

SC_PROT: *0x1C (immer)*
CTL: *xxx*

EOT: 0 (not used here)
FIFO: 0 (not used here)
BT: 0 (not used here)
A64: 0 (not used here)
AM: 0 (not used here)
WR: 0 -> Read; 1 -> Write

REQ: 00 (Bit17,16)
CON: 10 (Bit17,16; postive Confirmation)
ECON: 11 (Bit17,16; negative Confirmation)

SP: *0x00: Register*
BE: *0x0F: 32-bit transfer (register access 32-bit always)*

5.5.1 Write remote register

CTL: *0x04 (REQ and WR)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x04	SP: 00	BE: 0F
Address A31-0			
data			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x6	SP: 00	---

or in case of error

SC_PROT	CTL: 0x7	SP: 00	EC

5.5.2 Read remote register

CTL: *0x00 (REQ and RD)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x00	SP: 00	BE: 0F
Address A31-0			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x2	SP: 00	---
data			

or in case of error

SC_PROT	CTL: 0x3	SP: 00	EC

EC to be defined

5.6 Direct VME Bus Access Transfer Protocol

SC_PROT: *0x1C (always)*
CTL: *xxx*

EOT:
FIFO: 0 -> no address increment; 1 -> address increment (relevant with BT only)
BT: 0 -> single cycle; 1 -> block transfer
A64: 0 -> addresse A31-0 only part of request protokol
1 -> address A63-32 and address A31-0 part of request protocol
AM: 0 -> AM Code not part of request protocol (default:AM=0x09 ;A32 non priv. data)
1 -> AM Code part of request protocol
WR: 0 -> Read; 1 -> Write

REQ: 00 (Bit17,16)
CON: 10 (Bit17,16; positive confirmation)
ECON: 11 (Bit17,16; negative confirmation)

SP: *0x01: direct VME bus access*
BE: *xx*

5.6.1 Single Word Write Direct VME Bus Access

CTL: *0x04 (REQ and WR)*
CTL: *0x0C (REQ and WR and AM)*
CTL: *0x1C (REQ and WR and AM and A64)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 01	BE: 0F
address modifier (with CTL:AM =1 only)			
Address A63-32 (with CTL:A64 =1 only)			
address A31-0			
data			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x6 ?	SP: 01	---

or in error case

SC_PROT	CTL: 0x7 ?	SP: 01	EC

5.6.2 Single Word Read Direct VME Bus Access

CTL: *0x00 (REQ and RD)*
CTL: *0x08 (REQ and RD and AM)*
CTL: *0x18 (REQ and RD and AM and A64)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 01	BE: 0F
address modifier (with CTL:AM =1 only)			
address A63-32 (with CTL:A64 =1 only)			
address A31-0			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x2 ?	SP: 01	---
data			

or in error case

SC_PROT	CTL: 0x3 ?	SP: 01	EC

5.6.3 Block transfer Write Direct VME Bus Access

CTL: *0x24 (REQ , WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 01	BE: 0F
address nodifier (with CTL:AM =1 only)			
address A63-32 (with CTL:A64 =1 only)			
(Start) address A31-0 (4 Byte aligned)			
datum 1			
datum 2			
.....			
datum n			
SC_PROT	CTL : 0x81	SP: 01	BE: 0F

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x26 ?	SP: 01	---

or in error case

SC_PROT	CTL: 0x27 ?	SP: 01	EC

5.6.4 Block transfer Read Direct VME Bus Access

CTL: *0x20 (REQ and BT)*
CTL: *0x22 (CONF and BT; arbitrary: WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x20	SP: 01	BE: 0F
address modifier (with CTL:AM =1 only)			
address A63-32 (with CTL:A64 =1 only)			
(Start) address A31-0 (4 Byte aligned)			
BC (byte count; 4-er steps: 4,8, ...)			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x22	SP: 01	---
datum 1			
datum 2			
..			
datum n			
SC_PROT	CTL: 0x81 ?	SP: 01	---

or in error case

SC_PROT	CTL: 0x23 ?	SP: 01	EC

5.7 Mapped VME Bus Access Transfer Protocol

SC_PROT: *0x1C (immer)*
CTL: *xxx*

EOT:
FIFO: 0 -> no address increment; 1 -> address increment (relevant with BT only)
BT: 0 -> single cycle access; 1 -> block transfer
A64: 0 (not relevant here)
AM: 0 (not relevant here)
WR: 0 -> Read; 1 -> Write

REQ: 00 (Bit17,16)
CON: 10 (Bit17,16; positive confirmation)
ECON: 11 (Bit17,16; negative confirmation)

SP: *0x04: mapped VME bus access*
BE: *xx*

5.7.1 Single Word Write mapped VME Bus Access

CTL: *0x04 (REQ and WR)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 04	BE: 0F
address A31-0 A31-24: not used A23-16: address (and AM) map pointer A15-0: direct to VME			
datum			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x6 ?	SP: 04	---

or in error case

SC_PROT	CTL: 0x7 ?	SP: 04	EC

5.7.2 Single Word Read mapped VME Bus Access

CTL: *0x00 (REQ and RD)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 04	BE: 0F
address A31-0 A31-24: not used A23-16: address (and AM) map pointer A15-0: direct to VME			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x2 ?	SP: 04	---
datum			

or in error case

SC_PROT	CTL: 0x3 ?	SP: 04	EC

5.7.3 Block transfer Write mapped VME Bus Access

CTL: *0x24 (REQ , WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL	SP: 01	BE: 0F
address A31-0 A31-24: not used A23-16: address (and AM) map pointer A15-0: direct to VME			
datum 1			
datum 2			
.....			
datum n			
SC_PROT	CTL : 0x81	SP: 04	BE: 0F

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x26 ?	SP: 04	---

or in error case

SC_PROT	CTL: 0x27 ?	SP: 04	EC

5.7.4 Blocktransfer Read mapped VME Bus Access

CTL: *0x20 (REQ and BT)*
CTL: *0x22 (CONF and BT; arbitrary: WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x20	SP: 04	BE: 0F
address A31-0 A31-24: not used A23-16: address (and AM) map pointer A15-0: direct to VME			
BC (byte count; in steps of 4: 4,8, ...)			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x22	SP: 04	---
datum 1			
datum 2			
..			
datum n			
SC_PROT	CTL: 0x81 ?	SP: 04	---

or in error case

SC_PROT	CTL: 0x23 ?	SP: 04	EC

5.8 SDRAM Transfer Protocol

SC_PROT: *0x1C (always)*
CTL: *xxx*

EOT:
FIFO: 0 (not relevant)
BT: 0 -> Einzelwortzugriff; 1 -> Block transfer
A64: 0 (not relevant)
AM: 0 (not relevant)
WR: 0 -> Read; 1 -> Write

REQ: 00 (Bit17,16)
CON: 10 (Bit17,16; positive confirmation)
CON: 11 (Bit17,16; negative confirmation)

SP: *0x06: SDRAM*
BE: *0x0F: 32-bit transfer (not used, register access is 32-bit wide by default)*

5.8.1 Single Word Write SDRAM

CTL: *0x04 (REQ and WR)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x04	SP: 06	BE: 0F
address A31-0			
datum			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x6	SP: 06	---

or in error case

SC_PROT	CTL: 0x7	SP: 06	EC

5.8.2 Single Word Read SDRAM

CTL: *0x00 (REQ and RD)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x00	SP: 06	BE: 0F
address A31-0			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x2	SP: 06	---
datum			

or in error case

SC_PROT	CTL: 0x3	SP: 06	EC

5.8.3 Blocktransfer Write SDRAM

CTL: *0x24 (REQ , WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x24	SP: 06	BE: 0F
(Start) address A31-0 (4 Byte aligned)			
datum 1			
datum 2			
.....			
datum n			
SC_PROT	CTL : 0x81	SP: 06	BE: 0F

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x26	SP: 06	---

or in error case

SC_PROT	CTL: 0x27	SP: 06	EC

5.8.4 Blocktransfer Read SDRAM

CTL: *0x20 (REQ and BT)*
CTL: *0x22 (CONF and BT; arbitrary: WR and BT)*
CTL: *0x81 (END and EOT; arbitrary: WR and BT)*

Request (from PCI)

Bit 31 Bit 0

SC_PROT	CTL : 0x20	SP: 06	BE: 0F
(Start) Address A31-0 (4 Byte aligned)			
BC (byte count; 4-er steps: 4,8, ...)			

Confirmation from SIS3100

Bit 31 Bit 0

SC_PROT	CTL: 0x22	SP: 06	---
data 1			
data 2			
..			
data n			
BC (n x 4)			
SC_PROT	CTL: 0x81	SP: 06	---

or in error case

SC_PROT	CTL: 0x23	SP: 06	EC

6 SIS3100 Access through the Optical Interface

6.1 Control register space

Offset	Access	Function
0x000	R	Type-Identifier/Version register
0x004	R/W	Optical Status register
0x008	R/W	Optical Control register (reserved functions)
0x080	R/W	OPT-IN/OUT Register (FLAT/LEMO I/O)
0x084	R/W	OPT-IN-LATCH_IRQ Register
0x100	R/W	OPT-VME-Master Status/Control register
0x104	R/W	OPT-VME-Master Interrupt Status/Control register
0x110	R/W	OPT-VME Interrupt generation register (added with firmware version 5)
0x200	R/W	OPT-VME-Slave Status/Control register
0x204	R	OPT-VME-Master DMA_WRITE_BYTE_COUNTER
0x300	R/W	OPT-DSP Status/Control Register

Control register space can be accessed with the routines:

```
int s3100_control_read(int p, int offset, u_int32_t* data)
int s3100_control_write(int p, int offset, u_int32_t data)
```

Note: long word access, the offset has to be long word aligned (0x0, 0x4, 0x8 ...)

6.1.1 Type-Identifier/Version register(0x0, read)

This read only register holds the SIS3100 board type to allow for a distinction between different interface types. The board type of the SIS3100 VME side is 2.
Find below a table of board types (for the time being PCI and VME are the only implemented boards).

BIT	access	Name	Function
31-24 FF000000	RO	Firmware Version	1..255
23-16 00FF0000	RO	Firmware Id.	1 = universal other Ids. for dedicated firmware
15-8 0000FF00	RO	Hardware Version	1..255
7-0 000000FF	RO	Identifier 0x02	1 = PCI/PLX Interface (SIS1100) 2 = VME Controller (SIS3100) 3 = CAMAC/FERA Controller (SIS5100) 4 = Readout system with LVD SCSI

Example: The current version reads 0x 05 01 01 02

6.1.2 Optical status register (0x4, r/w)

BIT	Name	access	Function
31-16 FFFF0000	reserved	RO	0x0000
15 00008000	BAND_ERROR	WR: sel clr	VSC: Out-of-Band Error (not reseted after powerup and Link reset)
14 00004000	DISPAR_ERROR	WR: sel clr	VSC: Disparity Error (not reseted after powerup and Link reset)
13 00002000	UORUN_ERROR	WR: sel clr	VSC: Under/Overrun error (not reseted after powerup and Link reset)
12 00001000	TBERR_ERROR	WR: sel clr	VSC: Transmit Buffer Error(not reseted after powerup and Link reset)
11 00000800			0
10 00000400	LWORD_ERROR	WR: sel clr	Lword aligned error on optical interface
9 00000200			0
8 00000100			0
7 00000080	REC_VIOLATION		0 (reserved)
6 00000040	SEMA_CHG		0 (reserved)
5 00000020	INH_CHG	WR: sel clr	INHIBIT signal has changed (to inhibit)
4 00000010	SYNCH_CHG	WR: sel clr	RX/TX_SYNCH has changed
3 00000008	CONFIGURED	RO	allows remote side to detect RESET or power up (1 after reset or power up)
2 00000004	INHIBIT	RO	Transfer to remote side locked (TRANS_WAIT_FLAG_L) remote has send xoff or TRANSMIT_LINK_WAIT is active
1 00000002	TX_SYNCH	RO	Optical remote receiver is synchronised (TRANSMIT_LINK_OK)
0 00000001	RX_SYNCH	RO	Optical receiver is synchronised (RECEIVE_LINK_OK)

6.1.3 Optical control register (0x8, r/w)

This register is implemented as a selective J/K register. The user has to avoid to write a 1 to the clear and set bit of the same output at the same time, as an undefined toggle state may result.
The functions are reserved. They will be used in applications like VME to VME coupling.

Bit	Write Function	Read Function
31:16	Clear reserved bit [15:0]	0x0000
15:0	Set reserved bit [15:0]	Status reserved bit [15:0]

6.1.4 OPT-IN/OUT Register (0x80,read /write)

This register is under control of the Input/Output option (as far as installed on the given SIS3100).
On read access the status of the outputs and the current level on the inputs can be obtained, on write access the level of the outputs can be set. The register is implemented as a selective J/K register, the specific function (set/clear output) is executed by writing a 1 to the corresponding bit location, writes with a 0 have no effect. The user has to avoid to write a 1 to the clear and set bit of the same output at the same time, as an undefined toggle state may result.

Bit	Write Function	Read Function
31	no function	0
30	Generate pulse LEMO_OUT3	0
29	Generate pulse LEMO_OUT2	0
28	Generate pulse LEMO_OUT1	0
27	Generate pulse FLAT_OUT4	0
26	Generate pulse FLAT_OUT3	0
25	Generate pulse FLAT_OUT2	0
24	Generate pulse FLAT_OUT1	0
23	no function	0
22	Clear LEMO_OUT3	Status LEMO_IN3
21	Clear LEMO_OUT2	Status LEMO_ IN2
20	Clear LEMO_OUT1	Status LEMO_ IN1
19	Clear FLAT_OUT4	Status FLAT_ IN4
18	Clear FLAT_OUT3	Status FLAT_ IN3
17	Clear FLAT_OUT2	Status FLAT_ IN2
16	Clear FLAT_OUT1	Status FLAT_ IN1
15	no function	0
14	no function	0
13	no function	0
12	no function	0
11	no function	0
10	no function	0
9	no function	0
8	no function	0
7	no function	0
6	Set LEMO_OUT3	Status LEMO_OUT3
5	Set LEMO_OUT2	Status LEMO_OUT2
4	Set LEMO_OUT1	Status LEMO_OUT1
3	Set FLAT_OUT4	Status FLAT_OUT4
2	Set FLAT_OUT3	Status FLAT_OUT3
1	Set FLAT_OUT2	Status FLAT_OUT2
0	Set FLAT_OUT1	Status FLAT_OUT1

pulse length : 12.5ns
pulse polarity: if SET_OUTx is set then the polarity is inverted

6.1.5 OPT-IN-LATCH_IRQ Register (0x84,read /write)

This register is under control of the Input/Output option (as far as installed on the given SIS3100).
On read access the status of the outputs and the current level on the inputs can be obtained, on write access the level of the outputs can be set. The register is implemented as a selective J/K register, the specific function (set/clear output) is executed by writing a 1 to the corresponding bit location, writes with a 0 have no effect. The user has to avoid to write a 1 to the clear and set bit of the same output at the same time, as an undefined toggle state may result.

Bit	Write Function	Read Function	
31	Clear DSP_IRQ_LATCH bit	Status DSP_IRQ_LATCH bit	Doorbell
30	Clear LEMO_IN3_LATCH bit	Status LEMO_ IN3_LATCH bit	
29	Clear LEMO_IN2_LATCH bit	Status LEMO_ IN2_LATCH bit	
28	Clear LEMO_IN1_LATCH bit	Status LEMO_ IN1_LATCH bit	
27	Clear FLAT_ IN4_LATCH bit	Status FLAT_ IN4_LATCH bit	
26	Clear FLAT_ IN3_LATCH bit	Status FLAT_ IN3_LATCH bit	
25	Clear FLAT_ IN2_LATCH bit	Status FLAT_ IN2_LATCH bit	
24	Clear FLAT_ IN1_LATCH bit	Status FLAT_ IN1_LATCH bit	
23	Clear DSP_IRQ_Enable bit	Status DSP_IRQ_Enable	
22	Clear LEMO_IN3_IRQ Enable bit	Status LEMO_IN3	
21	Clear LEMO_IN2_IRQ Enable bit	Status LEMO_IN2	
20	Clear LEMO_IN1_IRQ Enable bit	Status LEMO_IN1	
19	Clear FLAT_ IN4_IRQ Enable bit	Status FLAT_IN4	
18	Clear FLAT_ IN3_IRQ Enable bit	Status FLAT_IN3	
17	Clear FLAT_ IN2_IRQ Enable bit	Status FLAT_IN2	
16	Clear FLAT_ IN1_IRQ Enable bit	Status FLAT_IN1	
15	1 Shot: IRQ_UPDATE	Status DSP_IRQ bit	
14	no function	Status LEMO_ IN3_IRQ bit	
13	no function	Status LEMO_ IN2_IRQ bit	
12	no function	Status LEMO_ IN1_IRQ bit	
11	no function	Status FLAT_ IN4_IRQ bit	
10	no function	Status FLAT_ IN3_IRQ bit	
9	no function	Status FLAT_ IN2_IRQ bit	
8	no function	Status FLAT_ IN1_IRQ bit	
7	SET DSP_IRQ Enable bit	Status DSP_IRQ Enable bit	
6	Set LEMO_IN3_IRQ Enable bit	Status LEMO_ IN3_IRQ Enable bit	
5	Set LEMO_IN2_IRQ Enable bit	Status LEMO_ IN2_IRQ Enable bit	
4	Set LEMO_IN1_IRQ Enable bit	Status LEMO_ IN1_IRQ Enable bit	
3	Set FLAT_ IN4_IRQ Enable bit	Status FLAT_ IN4_IRQ Enable bit	
2	Set FLAT_ IN3_IRQ Enable bit	Status FLAT_ IN3_IRQ Enable bit	
1	Set FLAT_ IN2_IRQ Enable bit	Status FLAT_ IN2_IRQ Enable bit	
0	Set FLAT_ IN1_IRQ Enable bit	Status FLAT_ IN1_IRQ Enable bit	

for PCI-Doorbell IRQ generation see 6.1.7 OPT-VME-Interrupt Status/Control register (0x104,read /write)

6.1.6 OPT-VME-Master Status/Control register (0x100,read/write)

The control register is in charge of the control of most of the basic properties of the SIS3100 board in write access. It is implemented via a selective J/K register, a specific function is enabled by writing a 1 into the set/enable bit, the function is disabled by writing a 1 into the clear/disable bit (which location is 16-bit higher in the register). An undefined toggle status will result from setting both the enable and disable bits for a specific function at the same time.

Bit	Write Function	Read Function
31	Clear SYSTEM VME BERR TIMER BIT1	0
30	Clear SYSTEM VME BERR TIMER BIT0	0
29	Clear LONG TIMER BIT1	0
28	Clear LONG TIMER BIT0	0
27	no function	0
26	no function	0
25	no function	0
24	Clear enable VME retry	0
23	Switch off user LED	0
22	Clear VME REQUESTER TYPE BIT	0
21	Clear VME_REQ_LEVEL BIT1	0
20	Clear VME_REQ_LEVEL BIT0	0
19	Clear POWER_ON_RESET bit	0
18	Clear LEMO_OUT_RESET bit	0
17	Clear VME_SYSRESET bit	0
16	Clear VME System Controller Enable bit (*2)	Status VME System Controller (*3)
15	Set SYSTEM VME BERR TIMER BIT1	Status SYSTEM VME BERR TIMER BIT1
14	Set SYSTEM VME BERR TIMER BIT0	Status SYSTEM VME BERR TIMER BIT0
13	Set LONG TIMER BIT1	Status LONG TIMER BIT1
12	Set LONG TIMER BIT0	Status LONG TIMER BIT0
11	no function	0
10	no function	0
9	no function	0
8	Enable VME retry(*7)	0
7	Switch on user LED	Status user LED
6	Set VME REQUESTER TYPE BIT	Status VME REQUESTER TYPE BIT
5	Set VME_REQ_LEVEL BIT1	Status VME_REQ_LEVEL BIT1
4	Set VME_REQ_LEVEL BIT0	Status VME_REQ_LEVEL BIT0
3	Set POWER_ON_RESET bit (*6)	Status POWER_ON_RESET
2	Set LEMO_OUT_RESET bit (*5)	Status LEMO_OUT_RESET bit
1	Set VME_SYSRESET bit (*4)	Status VME_SYSRESET bit
0	Set VME System Controller Enable bit (*2)	Status VME System Controller Enable bit

The power up value is 0x0000C000 (may depend on SIS3100 firmware version)

Notes:

(*2) is ored with the Jumper J10/1-2 ; Caution: if the jumper is not installed and the VME system controller functionality is enabled by software, the 16 MHz clock is not active during power up. This may result in problems with peculiar VME slave designs that use the VME clock to initialise on board logic.
(*3) is set if Jumper J10/1-2 is inserted or if VME System Controller Enable bit is set
(*4) if Jumper J90/11-12 is inserted and VME_SYSRESET bit is set then VME_SYSRESET is issued at power up

(*5) if Jumper J90/3-4 is inserted and the LEMO_OUT_RESET bit is set then LEMO_OUT_RESET is set (ored upon POWER_ON_RESET if Jumper J90/5-6 is inserted
(*6) if Jumper J90/9-10 is inserted and the POWER_ON_RESET bit is set, the SIS3100 generates a power up Reset)
(*7) A retry error (error code 0x212) may be caused by older VME backplanes, which do not properly terminate this previously reserved (pin B3 on connector P2) if retry is enabled (via bit 8).

Explanation/function of bit combinations:

SYSTEM VME BERR TIMER BIT1	SYSTEM VME BERR TIMER BIT0	VME Bus Error after
0	0	1,25 µs
0	1	6,25 µs
1	0	12,5 µs
1	1	100 µs (default)

Note: The default value of 100 µs will be fine with all VME slaves on the market, you may want to use a shorter value for your system however. The bus error code is 0x211.

LONG TIMER BIT1	LONG TIMER BIT0	LONG Timeout after
0	0	1 ms (default)
0	1	10 ms
1	0	50 ms
1	1	100 ms

LONG Timeout: arbitration timeout, no reply from current VME master or VME bus mastership not granted
The arbitration timeout error code is ox214.

VME_REQ_LEVEL BIT1	VME_REQ_LEVEL BIT0	VME Bus Request Level
0	0	BR3 (highest Level, default)
0	1	BR2
1	0	BR1
1	1	BR0

VME REQUESTER TYPE BIT	VME Bus Requester Type
0	Release on Request (default)
1	Release when Done

6.1.7 OPT-VME Interrupt Status/Control register (0x104,read /write)

The VME interrupts are enabled with their corresponding bit in this register. In addition the user can check on the status of the interrupt sources.

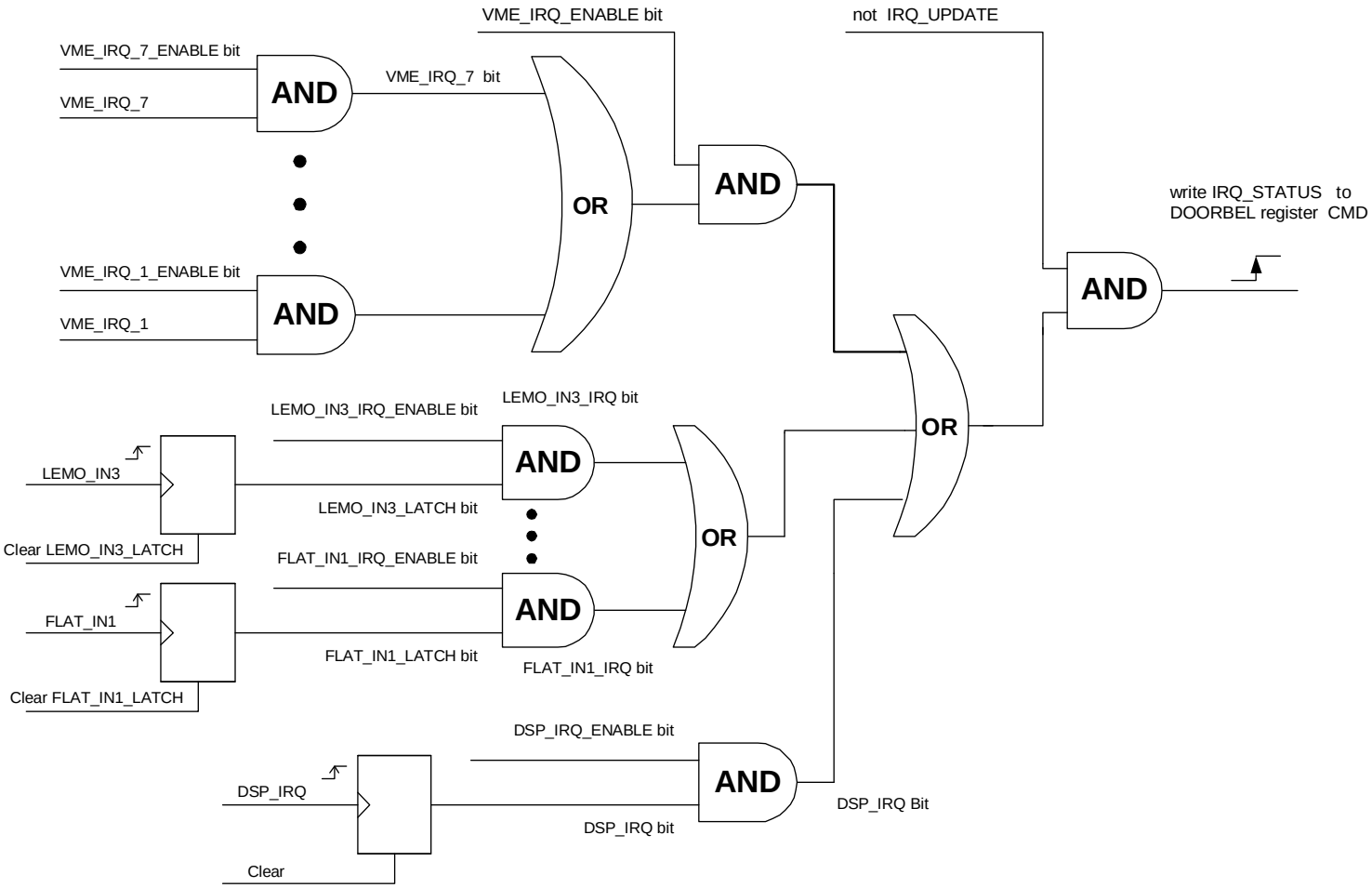
Bit	Write Function	Read Function
31		Status VME IRQ 7 on VME BUS
30		Status VME IRQ 6 on VME BUS
29		Status VME IRQ 5 on VME BUS
28		Status VME IRQ 4 on VME BUS
27		Status VME IRQ 3 on VME BUS
26		Status VME IRQ 2 on VME BUS
25		Status VME IRQ 1 on VME BUS
24		
23	Clear VME IRQ 7 Enable Bit	Status VME IRQ 7 bit
22	Clear VME IRQ 6 Enable Bit	Status VME IRQ 6 bit
21	Clear VME IRQ 5 Enable Bit	Status VME IRQ 5 bit
20	Clear VME IRQ 4 Enable Bit	Status VME IRQ 4 bit
19	Clear VME IRQ 3 Enable Bit	Status VME IRQ 3 bit
18	Clear VME IRQ 2 Enable Bit	Status VME IRQ 2 bit
17	Clear VME IRQ 1 Enable Bit	Status VME IRQ 1 bit
16	Clear VME IRQ Enable Bit	0
15	1 Shot: IRQ_UPDATE	0
14		0
13		0
12		0
11		0
10		0
9		0
8		0
7	Set VME IRQ 7 Enable Bit	Status VME IRQ 7 Enable Bit
6	Set VME IRQ 6 Enable Bit	Status VME IRQ 6 Enable Bit
5	Set VME IRQ 5 Enable Bit	Status VME IRQ 5 Enable Bit
4	Set VME IRQ 4 Enable Bit	Status VME IRQ 4 Enable Bit
3	Set VME IRQ 3 Enable Bit	Status VME IRQ 3 Enable Bit
2	Set VME IRQ 2 Enable Bit	Status VME IRQ 2 Enable Bit
1	Set VME IRQ 1 Enable Bit	Status VME IRQ 1 Enable Bit
0	Set VME IRQ Enable Bit	Status VME IRQ Enable Bit

The power up default value reads 0x 00000000

Status internal VME IRQ 1 = Status VME IRQ 1 Enable Bit and Status VME IRQ 1 on VME BUS
Status VME IRQ 1 = Status internal VME IRQ 1

PCI-Doorbell:
The leading edge of an IRQ issues an optical request, which writes the IRQ status in the doorbell register of the PLX PCI bridge chip.
With the command "IRQ_UPDATE " (write 0x8000 to `OPT-IN-LATCH_IRQ Register` or to `OPT-VME-Interrupt Status/Control register)` a pending IRQ is disabled shortly (if more than one IRQs are pending) what results in another leading edge generated by the other pending IRQ with consecutive doorbell register update.

Doorbell register bit	Function
31:16	0 (reserved)
15	Status DSP_IRQ
14	Status LEMO_ IN3_IRQ latch bit
13	Status LEMO_ IN2_IRQ latch bit
12	Status LEMO_ IN1_IRQ latch bit
11	Status FLAT_ IN4_IRQ latch bit
10	Status FLAT_ IN3_IRQ latch bit
9	Status FLAT_ IN2_IRQ latch bit
8	Status FLAT_ IN1_IRQ latch bit
7	Status VME IRQ 7 bit
6	Status VME IRQ 6 bit
5	Status VME IRQ 5 bit
4	Status VME IRQ 4 bit
3	Status VME IRQ 3 bit
2	Status VME IRQ 2 bit
1	Status VME IRQ 1 bit
0	0 (reserved)



SIS3100 PCI Doorbell IRQ blockdiagram

6.1.8 VME interrupt register (0x110, r/w)

This register was implemented with firmware version 5 to allow for VME interrupt generation with the SIS1100/3100. This functionality is useful to implement data transfers/message exchange between multiple masters. A VME IRQ with the given level and vector is generated if bit 11 is set. The interrupt vector will be passed to the bus during the (other masters) IRQ acknowledge cycle. Bit 11 will be cleared with the termination of the IRQ acknowledge cycle.. The interrupter type is DO8.

Bit	Write Function	Read Function
31	unused	0
...	...	...
12	unused	0
11	IRQ_Bit	1: IRQ not serviced yet 0: no IRQ generated or IRQ serviced
10	IRQ level bit 2	IRQ level bit 2
9	IRQ level bit 1	IRQ level bit 1
8	IRQ level bit 0	IRQ level bit 0
7	IRQ vector bit 7	IRQ vector bit 7
6	IRQ vector bit 6	IRQ vector bit 6
5	IRQ vector bit 5	IRQ vector bit 5
4	IRQ vector bit 4	IRQ vector bit 4
3	IRQ vector bit 3	IRQ vector bit 3
2	IRQ vector bit 2	IRQ vector bit 2
1	IRQ vector bit 1	IRQ vector bit 1
0	IRQ vector bit 0	IRQ vector bit 0

Note: The same IRQ level shall not be enabled on the VME to PCI side

6.1.9 OPT-VME-Slave Status/Control register (0x200,read /write)

This register controls the VME slave address of the SIS3100. The SIS3100 can either use geographical addressing (in conjunction with a VME64x backplane), use an emulated geographical address (via jumper array J10) or the base address defined by this control register. Note that the register is implemented in J/K style. .
The VME slave is disabled by default.

Bit	Write Function	Read Function
31	Clear Address Offset Bit A31	Status of GA4
30	Clear Address Offset Bit A30	Status of GA3
29	Clear Address Offset Bit A29	Status of GA2
28	Clear Address Offset Bit A28	Status of GA1
27	Clear Address Offset Bit A27	Status of GA0
26	Clear Enable VME Slave_OPT bit	Status of GAP
25	Clear Disable VME Slave_GA bit	0
24	no function	0
27:16	no function (reserved)	0
15	Set Address Offset Bit A31	Status Address Offset Bit A31
14	Set Address Offset Bit A30	Status Address Offset Bit A30
13	Set Address Offset Bit A29	Status Address Offset Bit A29
12	Set Address Offset Bit A28	Status Address Offset Bit A28
11	Set Address Offset Bit A27	Status Address Offset Bit A27
10	Set Enable VME Slave_OPT bit	Status Enable VME Slave_OPT bit
9	Set Disable VME Slave_GA bit	Status Disable VME Slave_GA bit
8	no function	0
7:0	no function (reserved)	0

Summary on VME slave address setting :

The VME slave base address is defined by the offset bits A[31:27] if the „Enable VME Slave_OPT bit" is set to 1

The VME slave address is defined by the GA lines (jumper J10 repsectively) if the VME Slave_OPT bit" is set to 0. GA4 is compared to A31, . GA0 to A27.

The slave is disabled if all GA lines (GAP, GA[4:0]) are 0 (corresponding jumpers on J10 open respectively). This is the factory default.

The VME slave is disabled while the SIS3100 is VME master.

6.1.10 OPT-VME-Master DMA_WRITE_BYTE_counter register (0x204,read)

This register holds the counter of written Bytes upon a bus error terminated block transfer write cycle. . It is cleared with the start of a block transfer write cycle.

Bit	Function
31	0
30	0
29	Byte counter bit 29
...	
0	Byte counter bit 0

6.1.11 OPT-DSP Status/Control Register (0x300, r/w)

The DSP control register is in charge of the control of several properties of the SIS9200 DSP add on option (if installed, can be checked with bit-24 of opt-dsp status/control register). It is implemented via a selective J/K register, a specific function is enabled by writing a 1 into the set/enable bit, the function is disabled by writing a 1 into the clear/disable bit (which location is 16-bit higher in the register). An undefined toggle status will result from setting both the enable and disable bits for a specific function at the same time.

Bit	Write Function	Read Function
31	none	DSP FLAG 3 Status
30	none	DSP FLAG 2 Status
29	none	DSP FLAG 1 Status
28	none	DSP FLAG 0 Status
27	Clear OPT_DSP_BOOT_CTRL_ENABLE *	0
26	Clear Control 10 *	0
25	Clear OPT_DSP BOOT EPROM *	0
24	Clear OPT_DSP RUN *	DSP available
23	none	0
22	none	0
21	none	0
20	none	0
19	none	0
18	none	0
17	none	0
16	none	0
15	none	0
14	none	0
13	none	0
12	none	0
11	Set OPT_DSP_BOOT_CTRL_ENABLE	Status OPT_DSP_BOOT_CTRL_ENABLE
10	Set Control 10 (reserved)	Status Control 10
9	Set OPT_DSP_BOOT_EPROM	Status OPT_DSP_EPROM
8	Set OPT_DSP_RUN	Status OPT_DSP_RUN
7	Set Control 7 (reserved)	0
6	Set Control 6 (reserved)	0
5	Set Control 5 (reserved)	0
4	Set Control 4 (reserved)	0
3	Set Control 3 (reserved)	0
2	Set Control 2 (reserved)	0
1	Set Control 1 (reserved)	0
0	Generate DSP IRQ 2 pulse	0

Summary of DSP control bits:

OPT_DSP_BOOT_CTRL_ENABLE:
0 : DSP_BOOT_EPROM and DSP_RUN are controlled from VME Slave
1 : DSP_EPROM and DSP_RUN are controlled from Optical interface
(OPT_DSP BOOT EPROM , OPT_DSP RUN)

OPT_DSP BOOT EPROM :
0 : DSP boots from external SRAM
1 : DSP boots from Flasheprom

OPT_DSP RUN :
0 : DSP is in Reset state
1 : DSP is in Run state
set from 0 to 1: DSP will boot

6.1.12 OPT-VME-Address MAP register 0..255 (0x400..0x7FC,read/write)

Bit	Function
31	VME A31
..	..
16	VME A16
15	0
..	
8	0
7	0
6	0
5	VME AM5
4	VME AM4
3	VME AM3
2	VME AM2
1	VME AM1
0	VME AM0

During a mapped VME transfer the protocol addresses [31:24] are ignored. They are used to address the VME address map.

6.2 OPT-Sharc space

This address space (through the optical interface) is occupied by the SIS9200 SHARC DSP (where installed).

Offset (byte_adr)	Access	Function
0x4000 4000 to 0x4000 03FC	R/W	SHARC dual ported RAM
0x4000 0400	R/W	OPT_SDRAM_SPD EEPROM register
0x8100 0000 to 0x811F FFFC	R/W	(Boot) FLASH PROM ; 4Mbit (512K Byte) - only one Byte is valid: data D7:D0 ←→ FLASH Prom D7:D0 - Offset_A20 : Offset_A2 ←→ FLASH Prom A18:A0
0x8120 0000 to 0x812F FFFC	R/W	Extern DSP SRAM ; 256 K x 48bit - data D31:D0 ←→ SHARC D47:D16
0x8130 0000 (to 0x813F FFFC)	R/W	D48 Register - data D15:D0 ←→ SHARC D15:D0

This address space can be accessed with the routines:

```
int s3100_sharc_write(int p_sharc, u_int32_t byte_adr, u_int32_t* ptr_data,
u_int32_t num_of_lwords)
int s3100_sharc_read(int p_sharc, u_int32_t byte_adr, u_int32_t* ptr_data,
u_int32_t num_of_lwords)
```

with p_sharc ist being the descriptor for SIS3100sharc !!

Note: The timeout is set to 8 ms. If a resource is blocked for the timeout period, the corresponding access will be terminated.

6.3 OPT-SDRAM space

This address space exists through the optical interface if the SDRAM option is installed.

Offset (byte_adr)	Access	Function
0x0000 0000 to 0x03ff fffc	R/W	Start address of optional SDRAM End address of 64 Mbyte SDRAM
or 0x07ff fffc	R/W	 End address of 128 Mbyte SDRAM
or 0x0fff fffc	R/W	 End address of 256 Mbyte SDRAM
or 0x1fff fffc	R/W	End address of 512 Mbyte SDRAM

Note: The OPT_SDRAM_SPD register is supposed to be accessed by the driver to detect the SDRAM memory size and to configure the register accordingly. As the SDRAM device will limit to the memory addresses at this stage, the SHARC device has to be used.

SDRAM can be accessed with the routines:

```
int s3100_sdram_write(int p_sdram, u_int32_t byte_adr, u_int32_t* ptr_data,
u_int32_t num_of_lwords)
int s3100_ sdram _read(int p_sdram, u_int32_t byte_adr, u_int32_t*
ptr_data, u_int32_t num_of_lwords)
```

with p_sdram being the descriptor for the SIS3100sdram

Note: The SDRAM timeout is 8 ms. If the VME slave or the SHARC DSP blocks optical link SDRAM access for a longer period, the cycle will be terminated.

6.3.1 OPT_SDRAM_SPD register

This read/write register is used to define the size of the SDRAM option.

Bit	Write Function	Read Function
31	none	0
30	none	0
29	none	0
28	none	0
27	none	0
26	none	0
25	Clear SDRAM Size Bit 1	0
24	Clear SDRAM Size Bit 0	0
23	none	0
22	none	0
21	none	0
20	none	0
19	none	0
18	none	0
17	Set SDRAM Size Bit 1	Status SDRAM Size Bit 1
16	Set SDRAM Size Bit 0	Status SDRAM Size Bit 0
15	none	0
14	none	0
13	none	0
12	none	0
11	none	0
10	none	0
9	none	SDA_IN
8	none	SDA_IN_LATCH (latch with SCL)
7	none	0
6	none	0
5	none	0
4	none	0
3	none	0
2	SDA_OUT_EN (1: enable SDA)	0
1	SDA_OUT (serial data out)	0
0	SCL (serial clock)	0

Size Bit Settings		
Bit 1	Bit 0	SDRAM Size
0	0	64/128 Mbyte
0	1	256/512 Mbyte
1	0	reserved
1	1	reserved

Supported SDRAM strip				
Size in Mbytes	Row	Col	Banks	Size 0 Bit*
64	12	9	1	no
128	12	9	2	no
256	13	10	1	yes
512	13	10	2	yes

Size 0 Bit*: in register OPT_SDRAM_SPD_EEPROM

7 VME slave access

The VME slave base address is controlled through theOPT-VME-Slave Status/Control register (refer to section 6.1.6).

7.1 VME Slave Address Map

The SIS3100 resources and their locations are listed in the table below.

Offset (VME addr)	R/W	Access	Function
0x0000 0000	R	D32	Type-Identifier/Version register
0x0000 0010	R/W	D32	VS-DSP Control/Status register
0x0000 0014	R/W	D32	SDRAM page register
0x0100 0000 to 0x011F FFFC	R/W	D32	(Boot) FLASH PROM ; 4Mbit (512K x8) - only one Byte is valid - data D7:D0 ←→ FLASH Prom D7:D0 - Offset_A20 : Offset_A2 ←→ FLASH Prom A18:A0
0x0120 0000 to 0x012F FFFC	R/W	D32 BLT32 MBLT64	Extern DSP SRAM ; 256 K x 48bit - data D31:D0 ←→ SHARC D47:D16
0x0130 0000 (to 0x013F FFFC)	R/W	D32 (BLT32) (MBLT64)	D48 Register - VME D15:D0 ←→ SHARC D15:D0
0x0200 0000	R/W	D32	Optical space (future use VME-VME coupling)
0x0300 0000 to 0x0300 03FC	R/W	D32 BLT32 MBLT64	Dual ported RAM (with SIS9200 DSP installed)
0x0300 4000	R/W	D32	SPD EEPROM of SDRAM
0x0400 0000 to 0x07FF FFFC	R/W	D32 BLT32 MBLT64	Start addreess of optional SDRAM End address of 64 Mbyte SDRAM

An address space of 64 MBytes is reserved for the SDRAM option. To access larger memories a page offset register is yet to be implemented.

7.2 Type-Identifier/Version register(0x0, read)

This read only register holds the SIS3100 board type to allow for a distinction between different interface types. The board type of the SIS3100 VME side is 2.
Find below a table of board types (for the time being PCI and VME are the only implemented boards).

BIT	**access**	**Name**	Function
31-24 FF000000	RO	Firmware Version	1..255
23-16 00FF0000	RO	Firmware Id.	1 = universal other Ids. for dedicated firmware
15-8 0000FF00	RO	Hardware Version	1..255
7-0 000000FF	RO	Identifier 0x02	1 = PCI/PLX Interface (SIS1100) 2 = VME Controller (SIS3100) 3 = CAMAC/FERA Controller (SIS5100) 4 = Readout system with LVD SCSI

Example: The current version reads 0x 03 01 01 02

7.3 VS-DSP Status/Control Register (0x10, r/w)

The DSP control register is in charge of the control of several properties of the SIS9200 DSP add on option if installed. It is implemented via a selective J/K register, a specific function is enabled by writing a 1 into the set/enable bit, the function is disabled by writing a 1 into the clear/disable bit (which location is 16-bit higher in the register). An undefined toggle status will result from setting both the enable and disable bits for a specific function at the same time.

Bit	Write Function	Read Function
31	reserved)	DSP FLAG 3 Status
30	reserved	DSP FLAG 2 Status
29	reserved	DSP FLAG 1 Status
28	reserved	DSP FLAG 0 Status
27	reserved	0 (Reserve)
26	reserved	0 (Reserve)
25	Clear VS_DSP BOOT EPROM *	0 (Reserve)
24	Clear VS _DSP RUN *	DSP available
23	Clear Control 7 *	0 (Reserve)
22	Clear Control 6 *	0 (Reserve)
21	Clear Control 5 *	0 (Reserve)
20	Clear Control 4 *	0 (Reserve)
19	Clear Control 3 *	0 (Reserve)
18	Clear Control 2 *	0 (Reserve)
17	Clear Control 1 *	0 (Reserve)
16	Clear Control 0 *	0 (Reserve)
15	reserved	0 (Reserve)
14	reserved	0 (Reserve)
13	reserved	0 (Reserve)
12	reserved	0 (Reserve)
11	reserved	Status OPT_DSP_BOOT_CTRL_ENABLE
10	reserved	0 (Reserve)
9	Set VS _DSP_BOOT_EPROM	Status VS _DSP_EPROM
8	Set VS _DSP_RUN	Status VS _DSP_RUN
7	Set Control 7 (reserved)	Status Control 7
6	Set Control 6 (reserved)	Status Control 6
5	Set Control 5 (reserved)	Status Control 5
4	Set Control 4 (reserved)	Status Control 4
3	Set Control 3 (reserved)	Status Control 3
2	Set Control 2 (reserved)	Status Control 2
1	Set Control 1 (reserved)	Status Control 1
0	Set Control 0 (reserved)	Status Control 0

OPT_DSP_BOOT_CTRL_ENABLE: (setable only from Optical Interface)

0 : DSP_BOOT_EPROM and DSP_RUN are controlled from VME Slave
(VS_DSP BOOT EPROM , VS_DSP RUN)

1 : DSP_EPROM and DSP_RUN are controlled from Optical interface

VS_DSP BOOT EPROM :

0 : DSP boots from external SRAM

1 : DSP boots from Flasheprom

VS_DSP RUN :

0 : DSP is in Reset state

1 : DSP is in Run state

set from 0 to 1: DSP will boot

7.4 SDRAM Page Register (0x14, r/w)

The SDRAM page register was implemented to allow for access to SDRAM memory strips beyond 64 MByte. The 64 MByte window to be addressed is selected viat Bits [2:0] of the register as shown on the table below.

Bit 3	Bit 2	Bit 1	Bit 0	Address window
0	0	0	0	0-64
0	0	0	1	64-128
0	0	1	0	128-192
0	0	1	1	192-256
0	1	0	0	256-320
0	1	0	1	320-384
0	1	1	0	384-448
0	1	1	1	448-512
1	x	x	x	Reserved for 1 GB

8 VME side LEDs

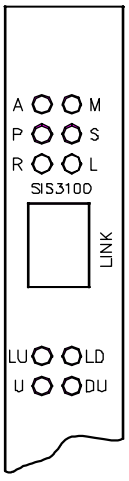
The SIS3100 has 10 front panel and 8 surface mounted printed circuit board (PCB) LEDs to visualise part of the units status. While the front panel LEDs allow the user to monitor part of the boards activities, the PCB LEDs were implemented for hardware and firmware debugging purposes mainly.

8.1 Front panel LEDs

The SIS3100 has 10 front panel and 8 surface mounted printed circuit board (PCB) LEDs to visualise part of

LED	Color	Function
A	yellow	Access (to VME slave port)
M		Master
P	red	Power
S		Sequencer activity
R	green	Ready (logic configured)
L		Link up
LU	green	Link data up (PCI to VME)
LD		Link data down (VME to PCI)
U	green	User
DU		DSP user

The arrangement of the front panel LEDs on the upper part of the front panel is shown in the sketch below.



8.1.1 Explanation of front panel LEDs

LED	Description
A	VME access to VME slave port of SIS3100
M	VME master, lit whenever the SIS3100 accesses the VME bus
P	Power, signals presence of +5 V supply voltage
S	Signals activity of the SIS3100 sequencer
R	Ready, lit when on board logic is configured (off during power up LED self test)

L	Link up, lit when connection to PCI side (or loopback connection) is established
LU	Link data up, lit when data are send (and LED link up lit), special case as described below when LED link up is off
LD	Link data down, lit when link data are received (and LED link up lit), special case as described below when LED link up is off
U	User LED, to be set and cleared under user program control
DU	DSP user LED, to be set and cleared under optional DSPs user program

8.1.2 Special case: LED Link up off

In standard operation (i.e. VME and PCI side powered an connected with optical fiber) the LED Link up off condition signals a problem on the Gigabit link connection. The LEDs Link data up and Link data down are used to signal the problem cause under this condition. Link data up is lit in case of a problem on the transmitter side, link data down is lit in case of a problem on the receiver side. A short loopback cable (with proven reliability) is useful to track down the problem source (fibre, VME side or PCI side)

8.2 PCB LEDs

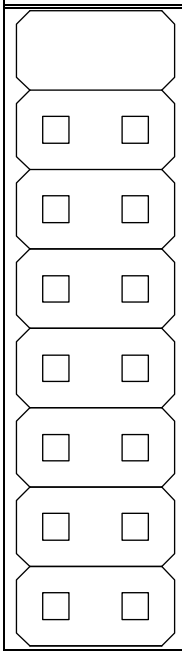
The 8 red PCB LEDs D651-D658 are mounted close to the front panel on the upper edge of the SIS3100. The reflect the status of the Vitesse serializer/deserializer (SERDES) chip.

LED	Function
D651	valid data
D652	valid KChar
D653	idle detect
D654	resync
D655	lossync
D656	norun error
D657	band error
D658	dispar error

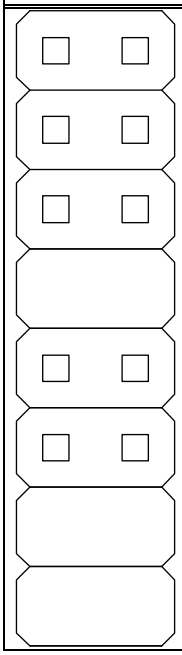
9 SIS3100 Jumpers

The SIS3100 card has two jumper arrays with 8 jumpers each. Array J10 controls VME slave port access and VME system controller functionality, J90 handles reset conditions mainly. A more detailed description of the two arrays and their factory default settings is given in the tables below.

9.1 J10

	Function	Factory default setting
	VME system controller	closed
	unused	open
	GAP	open
	GA0	open
	GA1	open
	GA2	open
	GA3	open
	GA4	open

9.2 J90

	Function	Factory default setting
	unused	open
	connect FPGA reset to LEMO reset output	open
	connect power on reset to LEMO reset output	open
	connect NIM reset input to execution of SIS3100 power on reset (see Note 3)	closed
	power on reset	open
	VME SYSRESET initiates power on reset of SIS3100	open
	FPGA reset results in VME SYSRESET	closed
	power on reset of SIS3100 results in VME SYSRESET	closed

Notes:
1.) some jumper combinations may result in a power up reset deadlock
2.) Typical Master/slave SYSRESET setting
While it is typical for a VME master to issue SYSRESET upon power up (jumper 8 of J90 closed) it is more suited for a VME slave to execute a power on reset as soon as the VME SYSRESET condition is detected (jumper 6 of J90 closed).
3.) The width of the reset signal has to be greater than 20 ms

9.2.1 JP_DSP

If the jumper JP_DSP is opened, the JTAG lines TDI and TDO of the installed SIS9200 DSP piggy will be connected to the main board and the programmable components on the card will become part of the SIS3100 JTAG chain. The default setting is jumper closed (i.e. closed TDI, TDO chain on the SIS3100 board).

10 VME master/system controller

10.1 Multi master operation

VME is a multi master system, what allows you to use several SIS3100 modules or a mixture of SIS3100 VME sequencers and other VME master hardware in one crate. The sections below have to be taken into account for successful multi master operation.

10.1.1 System Controller

The SIS3100 can act as VME system controller. The 16 MHz VME system clock is generated by the SMD oscillator U10. and enabled by jumper 1 of jumper array J10.
Make sure not to have more than one system controller on the VME backplane. The system controller has to be the leftmost master in the crate (typically it will reside in slot 1). In the case of the SIS3100 the system controller is enabled/disabled with jumper pair 1-2 of jumper array J10 . SIS3100 system controller functionality can also be enabled/disabled via the OP-VME control register. The system controller on/off status is an OR of the control register setting and the jumper and can be read back from the status register. The factory default is system controller enabled (as most SIS3100 cards are used in a single master environment.

Note: A VME diagnosis module like the VDIS or a measurement with a VME bus extender can be used to check, whether a particular CPU or interface generates system clock (with all other interfaces/CPUs unplugged from the VME backplane. Some VME slave modules may use the system clock to initialize on board resources, this mechanism may fail if the system clock is generated by more than one board in the crate. The system clock can also be activated by software if the jumper is not in place. In this case the user has to be aware, that no SYSCLOCK will be generated during the power up phase of the crate. A SYSRESET may be required by certain VME slaves for proper initialization of on board circuitry after SIS3100 SYSCLOCK generation was enabled.

10.1.2 Bus grant/bus mastership

- make sure to set the jumpers on the bus grant (BG) daisy chain properly unless your crate has an automatic daisy chain backplane (refer to the VME specification).
- Make sure, that no VME master locks bus mastership. It may be a good idea to use release when done instead of release on request where possible. It may be necessary to use a higher arbitration timeout than the standard value of 1 ms (selected via the OPT-VME control register).
- use different bus request (BR) levels as needed. The bus request level of the SIS3100 is programmed with the OPT-VME control register. The BR level of the SIS3100 defaults to 3 (highest level)

11 SHARC resources

The optional SIS9200 SHARC DSP add on will help you to implement many demanding data acquisition solutions.

11.1 SHARC address map

This section lists the resources that are accessible on/through the DSP.

Internal Address start	Internal Address end	SHARC Address SELECT lines	Data width	Resource
0x0000 0000	0x0000 0022		16 bit	Sharc Communication registers
0x0040 0000	0x0043 FFFF	MS0	48 bit	SRAM (256K x 48)
0x0080 0000	0x0087 FFFF	MS1 and BMS	8 bit	512K x 8 Flash Memory (SHARC: Data [23:16])
0x00C0 0000	0x00FF FFFF	MS2	16 bit	Communication interface space (SHARC: Data[31:16] are used)
0x0100 0000	0x013F FFFF	MS3	16 bit	Local interface space (SHARC: Data[31:16] are used)
0x0140 0000	0x1FFF FFFF		32 bit	not used (dead !)
0x2000 0000	0x3FFF FFFF		32 bit	Internal space (register , dual ported memory)
0x4000 0000	0x7FFF FFFF		32 bit	VME Sequencer space
0x8000 0000	0x8FFF FFFF		32 bit	SDRAM (Block Access)
0x9000 0000	0x9FFF FFFF		32 bit	SDRAM (VME read FIFO access)
0xC000 0000	0xDFFF FFFF		32 bit	SDRAM (Random Access)
0xE000 0000	0xFFFF FFFF		32 bit	SDRAM (Read – Add – Write access)

Notes on SDRAM block access:

The DSP will pass the address and arbitrate for the SDRAM with the first block access cycle. The DSP will hold mastership over the SDRAM until the next random access (note that SDRAM access via the optical interface or the VME slave may result in a timeout or BERR respective). The first random access following block access will use the last incremented block access address and will release SDRAM mastership upon completion (i.e. the first random access will be a dummy cycle in most applications).

- Read block mode can be terminated with a random read cycle only
- Write block mode can be terminated by a random write cycle or a dummy random read cycle

SDRAM VME read FIFO access:

This address range allows for direct copy of data from the VME read FIFO to SDRAM in a single instruction. The write datum is a dummy value.

Example:

```
M3=1;                           /* preset postincrement to 1          */
R3=0x90000000;                  /* SDRAM VME read FIFO access         */
I3=R3;                          /* load register with address         */

LCNTR=32, DO(PC,1) UNTIL LCE ; /* copy 32 values from FIFO to SDRAM  */
DM(I3,M3) = R0 ;                /* copy from VME FIFO to SDRAM        */
                                /* R0 is dummy                        */
```

Read/Add/Write access:

Read/add/write access was implemented to facilitate histogramming applications. If A29 is set during SDRAM write access to a memory location, the contents of the memory location will be added to the writes datum and the summed value will be stored back to the memory location.

11.1.1 SHARC Local Bus interface registers

The local bus interface register set allows for interaction with the input output option of the SIS3100 and to retrieve information on the sequencer status and the VME read FIFO fill level

Address	Mnemonic	DSP RW	Function
0x0100 0000	LOC_ SEQ _STATUS	R	
0x0100 0004	LOC_ IO _IN_REG	R/W	Read/clear latched input information
0x0100 0005	LOC_ IO _OUT_REG	R /W	FLAT [1..4] OUT Reg
0x0100 0006	LOC_ IO _OUT_PULSE	W	FLAT [1..4] OUT Pulse

Note: you will have to set access to two wait states prior to usage as illustrated in the Assembly language example below (the example is extracted from the file 060_link.asm)

```
#define WAIT   0x02

       R6=0x2004EAA5;
       DM(WAIT)=R6 ;          /* Modify WAIT      */
```

11.1.1.1 SHARC LOC_SEQ_STATUS register (0x0100 0000; read only)

The read only local sequencer status register holds busy/not busy information on the sequencer status and the fill level of the VME read FIFO

Bit	Read Function
31	undefined
..	
12	undefined
11	VME_RD_FIFO_FULL
10	VME_RD_FIFO_MORE_EQ_480
9	VME_RD_FIFO_MORE_EQ_256
8	VME_RD_FIFO_MORE_EQ_128
7	VME_RD_FIFO_MORE_EQ_64
6	VME_RD_FIFO_MORE_EQ_32
5	VME_RD_FIFO_MORE_EQ_2
4	VME_RD_FIFO_EMPTY
3	"0"
2	"0"
1	Status VME_SEQUENCER_ERROR_FLAG
0	Status VME_SEQUENCER_BUSY

Example: Subroutine wait for completion of VME transaction

```
#define LOC_SEQ_STATUS 0x01000000

Wait_On_VME_Busy:
      R15 = DM (LOC_SEQ_STATUS) ;
      BTST R15 by 0 ;
      IF NOT SZ JUMP (PC,Wait_On_VME_Busy) ;
      RTS ;
```

11.1.1.2 SHARC LOC_IO_IN_REG register **(0x0100 0004 ; read/write)**

This read/write register gives access to the latched input information of the front panel input option. The latched bit is set if a leading edge is detected on the corresponding input.

Bit	Read Function	Write Function
31	0	none
...	...	...
16	0	none
15	Status LATCHED_BIT_VME_IRQ	if '1': clear LATCHED_BIT_VME_IRQ
14	Status LATCHED_BIT_LEMO_IN_3	if '1': clear LATCHED_BIT_ LEMO_IN_3
13	Status LATCHED_BIT_LEMO_IN_2	if '1': clear LATCHED_BIT_ LEMO_IN_2
12	Status LATCHED_BIT_LEMO_IN_1	if '1': clear LATCHED_BIT_ LEMO_IN_1
11	Status LATCHED_BIT_FLAT_IN_4	if '1': clear LATCHED_BIT_ FLAT_IN_4
10	Status LATCHED_BIT_FLAT_IN_3	if '1': clear LATCHED_BIT_ FLAT_IN_3
9	Status LATCHED_BIT_ FLAT_IN_2	if '1': clear LATCHED_BIT_ FLAT_IN_2
8	Status LATCHED_BIT_FLAT_IN_1	if '1': clear LATCHED_BIT_ FLAT_IN_1
7	Status VME_IRQ	none
6	Status LEMO_IN_3	none
5	Status LEMO_IN_2	none
4	Status LEMO_IN_1	none
3	Status FLAT_IN_4	none
2	Status FLAT_IN_3	none
1	Status FLAT_IN_2	none
0	Status FLAT_IN_1	none

Example: Test/clear all latched bits, check on NIM1 input

```
#define LOC_IO_IN_REG 0x01000004
#define INTRAM 0x00400000                        // SIS9200 SRAM
#define INT0ST INTRAM+7

     R1=DM(LOC_IO_IN_REG);     // read latched IO bits
     DM(LOC_IO_IN_REG)=R1;     // clear all set IO bits
     BTST R1 by 12;                           // NIM1
     IF SZ JUMP (PC,READIRQ6) ;
     R2=DM(INT0ST);
     R2=R2+1;
     DM(INT0ST)=R2;
READIRQ1:
```

11.1.1.3 SHARC LOC_ IO _OUT_REG register **(0x0100 0005 ; read/write)**

This register allows you to set the level on the four flat cable outputs of the output option of the SIS3100.

Bit	Read Function	Write Function
31	'0'	none
..		
4	'0'	none
3	Status FLAT_OUT_4	FLAT_OUT_4
2	Status FLAT_OUT_3	FLAT_OUT_3
1	Status FLAT_ OUT_2	FLAT_ OUT_2
0	Status FLAT_ OUT_1	FLAT_ OUT_1

11.1.1.4 SHARC LOC_ IO _OUT_PULSE register **(0x0100 0006 ; write)**

This register provides efficient means to generate short output pulses on the flat cable outputs (basically in a single DSP instruction). The pulse width is 50 ns.

Bit	Write Function
31	none
..	
4	none
3	if '1' : generate FLAT_OUT_4 pulse
2	if '1' : generate FLAT_OUT_3 pulse
1	if '1' : generate FLAT_OUT_2 pulse
0	if '1' : generate FLAT_OUT_1 pulse

Example: Subroutine to generate output pulse on flat cable output 1

```
#define LOC_IO_OUT_PULSE 0x01000006

      // pulse SIS3100 ECL1/TTL1
      PULSE4:
      R15 = 1;
      DM(LOC_IO_OUT_PULSE) = R15;
      RTS ;
```

Note: Pulses on the LEMO outputs can be generated through the DSP SHARC Flags

11.1.2 Address map of SHARC registers

A list of the implemented SHARC communication registers is given in the table below.

Sharc address	R/W	Communication register
0x2000 0000 rw	R/W	VME Sequencer Control/STATUS register
0x2000 0001 w	W	VME Sequencer Data Write register
0x2000 0002 w	W	VME Sequencer word count register
0x2000 0003 r	R	VME Sequencer DMA actual word counter register
0x2000 0004 r	R	VME Sequencer Read FIFO flags and counter
0x2000 0005		
0x2000 0008 r	R	VME Sequencer Data Read Register
0x2000 0009 r	R	Internal Sequencer Data Read Register

11.1.3 Address map of SHARC-Optical Dual Ported Ram

Sharc address	Optical offset address	
0x2000 0100	0x4000 0000	Start address of Dual Ported Ram (with Sharc)
0x2000 0101	0x4000 0004	
0x2000 0102	0x4000 0008	
..	..	
0x2000 01FF	0x4000 03FC	End address of Dual Ported Ram

11.1.3.1 VME Sequencer Control/STATUS register (0x2000 0000; read/write)

The read/write VME sequencer control status holds status information (busy) and information on errors during the last transaction(s). . The error flag is cleared by setting bit 1 of the register and will go to 1 as soon as an error condition is encountered.

Bit	Write Function	Read Function
31	Set Control 31 (reserved)	0
...	...	...
...	...	...
4	Set Control 4 (reserved)	0
3	Set Control 3 (reserved)	0
2	Set Control 2 (reserved)	0
1	clear VME_SEQUENCER_ERROR_FLAG	Status VME_SEQUENCER_ERROR_FLAG
0	no function	Status VME_SEQUENCER_BUSY

11.1.3.2 VME Sequencer DMA Word count register (0x2000 0002; write only)

The write only VME sequencer DMA word counter register is used to load the length of VME block read/write cycles. The contents of the register is not changed during execution, i.e. you can reuse the preloaded value if several consecutive block reads have the same length.

Bit	Read Function
31	"0"
..	
14	"0"
23	Wordcount Bit 23
..	
0	Wordcount Bit 0

Example: preset the DMA (block transfer) word counter to 24 words

```
#define VME_DMA_MAX_WORDCOUNT 0x20000002

// dma read 24 words
    R0 = 24 ;
    DM(VME_DMA_MAX_WORDCOUNT) = R0 ;
```

11.1.3.3 VME Sequencer DMA transfer word count register (0x2000 0003; read only)

This read only register holds the actual number of transferred data words. It is cleared with the start of the block transfer.

Bit	Read Function
31	"0"
..	
14	"0"
23	Wordcount Bit 23
..	
0	Wordcount Bit 0

11.1.3.4 VME Sequencer Read FIFO flags and counter (0x2000 0004; read only)

Besides the FIFO read flags, which can be read more efficient from the SHARC LOC_SEQ_STATUS register, this read only register holds the actual word counter of the VME read FIFO.

Bit	Read Function
31	"0"
..	
24	"0"
23	VME_RD_FIFO_FULL
22	VME_RD_FIFO_MORE_EQ_480
21	VME_RD_FIFO_MORE_EQ_256
20	VME_RD_FIFO_MORE_EQ_128
19	VME_RD_FIFO_MORE_EQ_64
18	VME_RD_FIFO_MORE_EQ_32
17	VME_RD_FIFO_MORE_EQ_2
16	VME_RD_FIFO_EMPTY
15	"0"
..	"0"
9	"0"
8	FIFO word count Bit 8
7	FIFO word count Bit 7
..	
0	FIFO word count Bit 0

11.1.3.5 VME Sequencer Space **(0x4000 0000 - 0x7fff ffff, write)**

Sequencer action is controlled by the accessed address of the VME sequencer address space and the data that are written to the location

.

The function of the individual bits is given in the table below.

Bit	Write Function			
27	reserved			
26	reserved			
25	reserved			
24	reserved			
23	READ_FIFO_DISABLE		0: VME read data will be pushed into FIFO 1: VME read data will be stored in register only	
22	READ_FIFO_CLR_DISABLE		0: VME read FIFO will be cleared at the beginning 1: VME read FIFO will not be cleared at the beginning	
21	WORD_COUNT_CLR_DISABLE		0: clear word count on each sequencer read 1: leave word count uncleared	
20	VME ADDR_INC_DISABLE (DMA)			
19	HOLD_VME_MASTER			
18	DMA_CYCLE			
17	SINGLE_CYCLE			
16	VME_CYCLE (not INTERNAL_CYCLE)			
		for VME_CYCLE	**for Internal_CYCLE**	
15			INT_D7	Data bits of internal registers
14			INT_D6	
13	reserved		INT_D5	
12	reserved		INT_D4	
11	reserved		INT_D3	
10	Force VME AS Single cycle during DMA with constant Vme Addresses		INT_D2	
9	VME DS1 Veto		INT_D1	
8	VME DS0 Veto		INT_D0	
		for VME_CYCLE	**for Internal_CYCLE**	
7	SEQ_PROT_IO_7	VME_IACK		
6	SEQ_PROT_IO_6	VME_WRITE	INT_ WRITE	
5	SEQ_PROT_IO_5	VME_AM5	INT_A5	Address bits for internal sequencer registers
4	SEQ_PROT_IO_4	VME_AM4	INT_A4	
3		VME_AM3	INT_A3	
2		VME_AM2	INT_A2	
1		VME_AM1	INT_A1	
0	SEQ_PROT_IO_0	VME_AM0	INT_A0	

More detailed description of bits [19:16]

ADDR_19 HOLD_VME_ MASTER	ADDR_18 DMA_CYCLE	ADDR_17 SINGLE_CYCLE	ADDR_16 INTERN CYCLE	
x	x	x	1	internal Cycle
0	0	0	0	release VME MASTER_SHIP
1	0	0	0	VME Arbitration only and hold VME MASTER_SHIP
0	0	1	0	VME Single Cycle with Arbitration and release VME MASTER_SHIP
1	0	1	0	VME Single Cycle with Arbitration and hold VME MASTER_SHIP
0	1	x	0	VME DMA Cycle with Arbitration and release VME MASTER_SHIP
1	1	x	0	VME DMA Cycle with Arbitration and hold VME MASTER_SHIP

Example:

```
// Sharc VME Master defines
#define VME_DATA_FIFO_READ                  0x40000000
#define VME_ONLY_ARBITRATION_CYCLE          0x40080000
#define VME_ONLY_RELEASE_MASTER_CYCLE       0x40000000
#define VME_DMA_CYCLE                       0x40040000
#define VME_SINGLE_CYCLE                    0x40020000

#define READ_FIFO_DISABLE                   0x00800000
#define READ_FIFO_CLR_DISABLE               0x00400000
#define WORD_COUNT_CLR_DISABLE              0x00200000
#define VME ADDR_INC_DISABLE                0x00100000

#define VME_HOLD_MASTER                     0x00080000
#define VME_WRITE                                 0x40
#define VME_AM_F                                  0x0F
#define VME_AM_9                                  0x09

// define SIS3600 latch address&registers
# define SIS3600_MODUL1                     0x20000000
# define SIS3600_CTRL_REG                          0x0
# define SIS3600_READ_FIFO                       0x100
# define SIS3600_CTRL_PULSERON                    0x10

// write 0x10 to VME address 0x20000000
R0 = SIS3600_CTRL_PULSERON ;
DM(VME_WRITE_DATA_REG)=R0 ;
R0 = SIS3600_MODUL1 + SIS3600_CTRL_REG ; // address
DM (VME_SINGLE_CYCLE + VME_HOLD_MASTER + VME_WRITE + VME_AM_9) = R0 ;
CALL Wait_On_VME_Busy ; /* R15 will be used */
```

11.1.3.5.1 Address map (addr5..0) of Internal Sequencer registers

Up to 64 internal registers can be addressed with the internal register address bits INT_A0 through INT_A5 of the VME sequencer space. For the time being two are implemented. The actual register contents is defined by the bits INT_D0 through INT_D7 of the sequencer space.

address bits 5..0	register
0x0 rw	Internal control register (4 bits)
0x1 rw	Internal VME IRQ enable register
0x2 rw	Internal Input IRQ enable register
0x3 r	VME IRQ register
0x4 r	enabled VME IRQ register

11.1.3.5.1.1 Internal Control register (0x0)

The internal control register gives access to resources on the so called VME FPGA. The activation of the front panel DSP user LED under sequencer control is the only implemented function for the time being.

Bit	Write Function	Read Function (readable from Internal Sequencer Data Read Register)
d7 (addr 15)	clear reserved	0
d6	clear reserved	0
d5	clear reserved	0
d4	1: clear LED DU (DSP User)	0
d3	set reserved	status reserved
d2	set reserved	status reserved
d1	set reserved	status reserved
d0 (addr 8)	1: set LED DU (DSP User)	status LED DU (DSP User)

11.1.3.5.1.2 Internal VME IRQ Enable register (0x1)

The 7 VME interrupts can be enabled to generate DSP interrupts under sequencer control via the internal VME IRQ enable register.

Bit	Write Function	Read Function (readable from Internal Sequencer Data Read Register)
d7 (addr 15)	`VME IRQ7 Enable Bit`	
d6	`VME IRQ6 Enable Bit`	
d5	`VME IRQ5 Enable Bit`	
d4	`VME IRQ4 Enable Bit`	
d3	`VME IRQ3 Enable Bit`	
d2	`VME IRQ2 Enable Bit`	
d1	`VME IRQ1 Enable Bit`	
d0 (addr 8)		0

11.1.3.5.1.3 Internal Input IRQ Enable register (0x2)

The front panel inputs can be used to generate DSP interrupts, this functionality is controlled by the internal input IRQ enable register.

Bit	Write Function	Read Function (readable from Internal Sequencer Data Read Register)
d7 (addr 15)		0
d6	`LEMO Input3 IRQ Enable Bit`	
d5	`LEMO Input2 IRQ Enable Bit`	
d4	`LEMO Input1 IRQ Enable Bit`	
d3	`FLAT Cable Input4 IRQ Enable Bit`	
d2	`FLAT Cable Input3 IRQ Enable Bit`	
d1	`FLAT Cable Input2 IRQ Enable Bit`	
d0 (addr 8)	`FLAT Cable Input1 IRQ Enable Bit`	

11.1.3.5.1.4 Internal VME IRQ register (0x3)

This read only register holds the interrupt status of the 7 VME interrupts. The corresponding bit will be set to 1 if the given interrupt line is set.

Bit	Read Function
d7 (addr 15)	VME IRQ7 Status
d6	VME IRQ6 Status
d5	VME IRQ5 Status
d4	VME IRQ4 Status
d3	VME IRQ3 Status
d2	VME IRQ2 Status
d1	VME IRQ1 Status
d0 (addr 8)	0

11.1.3.5.1.5 Internal Enabled VME IRQ register (0x4)

The information of the internal VME IRQ register is anded bit wise with the enabled VME interrupts to form this read only register.

Bit	Read Function
d7 (addr 15)	VME IRQ7 Status anded with VME IRQ enable bit7
d6	VME IRQ6 Status anded with VME IRQ enable bit6
d5	VME IRQ5 Status anded with VME IRQ enable bit5
d4	VME IRQ4 Status anded with VME IRQ enable bit4
d3	VME IRQ3 Status anded with VME IRQ enable bit3
d2	VME IRQ2 Status anded with VME IRQ enable bit2
d1	VME IRQ1 Status anded with VME IRQ enable bit1
d0 (addr 8)	0

11.2 SHARC flags

LEMO outputs are treated in a different way than flat cable outputs for SIS3100 architecture reasons. The SHARC flags can be used to set/clear the 3 LEMO outputs of the front panel output option and to generate a PCI doorbell interrupt.

SHARC Flags	**VME FPGA**	**VME FPGA**
FLAG_0	LEMO OUT 1 (inverted)	in
FLAG_1	LEMO OUT 2 (inverted)	in
FLAG_2	LEMO OUT 3 (inverted)	in
FLAG_3	PCI Doorbell IRQ (inverted)	in

Note: In most applications you will want to clear the NIM level by setting the corresponding bit in the ASTAT register before further use.

Example: Subroutine to generate a 25ns wide pulse on LEMO output 1

```
// pulse SIS3100 LEMO1
PULSE1:
BIT CLR ASTAT 0x80000;   /set level
BIT SET ASTAT 0x80000;   /clear level
RTS ;
```

11.3 SHARC DSP interrupts

11.3.1 DSP Interrupt sources

The three interrupts IRQ2, IRQ and IRQ0 can be used for SIS1100/3100 specific interrupt generation. IRQ2 has the highest priority. The three interrupt sources are listed below.

Interrupt	Source	Description
IRQ2	PCI	Generated by setting bit0 of the OPT-DSP status/control register of the SIS1100. This interrupt has to be configured to edge sensitive (refer to section 11.3.2)
IRQ1	Input	IRQ generation with input(s) of the front panel I/O option of the SIS3100. Activation is controlled with the internal input IRQ enable register (refer to section 11.1.3.5.1.3)
IRQ0	VME	IRQ generation by VME interrupts. Activation for the 7 IRQ levels with the internal VME IRQ enable register (refer to section 11.1.3.5.1.2)

11.3.2 DSP Interrupt configuration

Interrupt behaviour is configured through the mode1, mode2 and imask registers of the DSP. The table below lists the involved bits.

Bit	Bit name	Function
mode 1 register bit 12	IRPTEN	1: enable IRQ; 0 IRQ disabled
mode 2 register bit 2	IRQ2E	1: edge sensitive; 0: level sensitive
mode 2 register bit 1	IRQ1E	1: edge sensitive; 0: level sensitive
mode 2 register bit 0	IRQ0E	1: edge sensitive; 0: level sensitive
mode 1 register bit	IRQ2I	1: enable IRQ0; 0: IRQ2 disabled
mode 1 register bit	IRQ1I	1: enable IRQ0; 0: IRQ1 disabled
mode 1 register bit	IRQ0I	1: enable IRQ0; 0: IRQ0 disabled

Example: Set IRQ0, IRQ1 and IRQ2 to be edge sensitive and enable all three IRQs

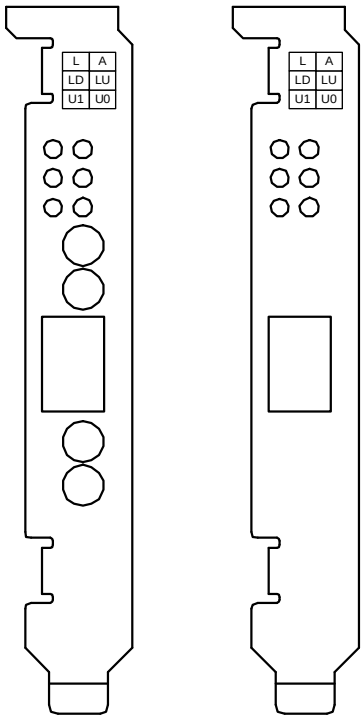
```
bit set mode2 IRQ0E | IRQ1E | IRQ2E;  // set edge sensitive
bit set imask IRQ0I | IRQ1I | IRQ2I;  // enable
bit set mode1 IRPTEN;                 // global IRQ enable
```

Note: The bit definitions for the mode and imask registers are contained in file def21060.h of the ADSP-21000 family development software. The system registers of the core processor (like mode1 e.g.) are made known to the compiler/assembler without the necessity for an include file.

12 SIS1100 Hardware Description

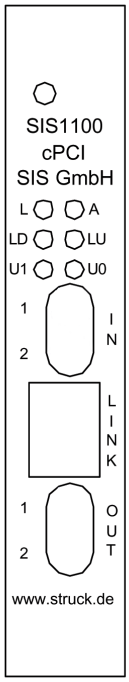
12.1 PCI Front panel

The SIS1100 uses a standard PCI front panel. The front panel as seen from the rear of the PC is shown in the graph below. The front panel to the left is the version with I/O option, to the right hand side shows the standard version.



12.2 3U cPCI front panel

The 3U cPCI (compact PCI) version of the SIS1100 is available with I/O option only.



12.3 SIS1100 LEDs

The two boards that form the SIS1100 have several LEDs to assist the user in case of problems. The front panel LEDs can be seen from the rear of the (closed) PC when the SIS1100 is installed, the PCB LEDs of the SIS1100-OPT and the SIS1100-CMC carrier can be seen if the PC is open.

12.3.1 Front panel LEDs

The green front panel LEDs of the SIS are grouped in 3 rows of 2 LEDs each. Find below a table of the LEDs as seen from the rear of the module. The LEDs are actually part of the SIS1100-OPT card.

Left	Right
Link up	Access
Link data down	Link data up
User 1	User 0

The function of the LEDs is explained in a little more detail in the table below.

LED	Function
Access	Lit with access to the SIS1100-OPT carrier board
Link up	Signals link connection to SIS3100 or other link partner
Link data up	Link data are being transmitted by SIS1100-OPT special case as described in section 8.1.2 when LED link up is off
Link data down	Link data are being received by SIS1100-OPT special case as described in section 8.1.2 when LED link up is off
User 1	To be set and cleared under user program control
User 0	To be set and cleared under user program control

12.3.2 SIS1100-OPT PCB LEDs

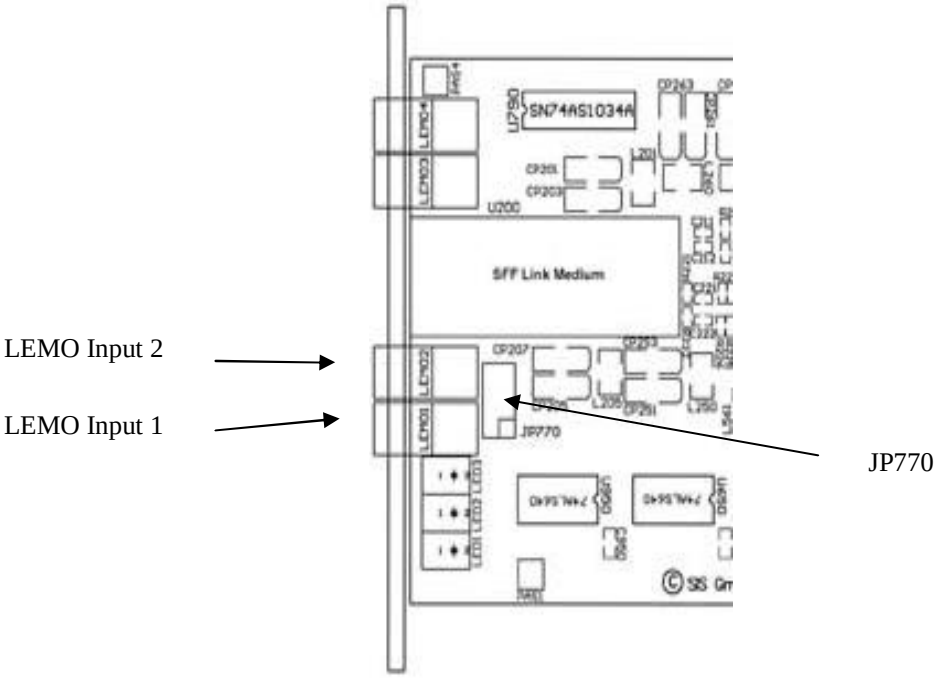
The SIS1100-OPT carrier board has 8 SMD LEDs. The have the same function and names as the corresponding LEDs on the SIS3100. Refer to section 8.2 for a detailed description.

12.3.3 SIS1100-CMC PCB LEDs

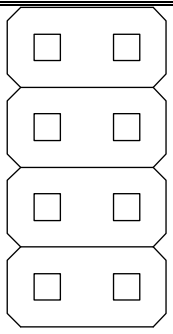
The SIS1100-CMC carrier board has 4 SMD LEDs.

12.4 SIS1100 Input termination

The input termination of the two LEMO inputs can be configured for 50 Ω and 1 KΩ with the four jumpers of jumper array JP770. The jumper array is located on the component side of the SIS1100-OPT, i.e. the board will have to be removed from the CMC carrier board for reconfiguration. The partial placement plan below shows the front panel section of the SIS3100-OPT (component side facing you, connectors to the right hand side. The jumper array and the function of its 4 positions are illustrated below. The factory default is both



inputs configured for 50 Ω termination, i.e. lowest jumper (next to text JP770 and third jumper set).

Function	Jumper array
Input 2 terminated to 1 KΩ if closed	
Input 2 terminated to 50 Ω if closed	
Input 1 terminated to 1 KΩ if closed	
Input 1 terminated to 50 Ω if closed	JP770

13 Appendix

13.1 Error codes

Some of the error codes may be driver specific, a general overview is given below.
Codes starting with 0x100 are generated on the PCI side (i.e. SIS1100), codes starting with 0x200 are generated on the VME side and error codes starting with 0x1000 are coming from the PCI to local bus bridge chip). Please refer to the documentation of the corresponding PLXtech PCI (Express) to local bus bridge chip for the later class (typically you are not likely to encounter those).
A table of the relevant codes is given below.

Name	Code	Condition
LE_SYNCH	0x101	Transmission error, protocol could not be send due to lack of synchronization (see status TX_SYNCH).
LE_TO	0x107	Protocol timeout, request not answered. The error is flagged until **p_balance** reaches zero (or is reset).
RE_NRDY	0x202	remote station not ready (status of control register bit READY)
RE_PROT	0x206	protocol error (illegal request e.g.)
RE_TO	0x207	protocol timeout
RE_BERR	0x211	VME bus error
RE_RETRY	0x212	VME retry
RE_ARB	0x214	Arbitration timeout, SIS3100 could not get bus mastership

Note 1: The VME retry line a previously reserved line (connector P2 pin B3). It is not properly terminated on some older VME crates. VME retry functionality is active by default on the SIS3100 up to firmware revision major 1 minor 5, what will result in retry errors in such a crate. SIS3100 firmware revisions major 1 minor 6 and higher have the retry inactive as power up default to avoid this problem source. Users that want to make use of retry functionality will have to activate the feature in the control register of the SIS3100.

Note 2: A more extensive error code list can be found in the gigalink documentation

13.2 Power consumption

The SIS3100 is a +5 V single supply design. On board voltages other than +5V are generated by linear regulators or DC/DC converters. A list with the used components can be found below.

Component designator	Voltage	Component	Powered components
U2	2.5 V	LM1084IT	FPGAs
U3	3.3 V	LM1084IT	link medium/SERDES/drivers
U5	-5 V	TMH0505S	flat cable in/outputs (ECL)
U6	-5 V	TMH0505S	LEMO in/outputs (NIM)

Note: U5 and U6 will be stuffed when required by the given I/O configuration only

The power consumption will depend on installed options and board activity. The figures below are worst case estimates/measurements.

U in V	Current in A	Configuration
+5 V		Base configuration
+5 V	2,1	Base configuration with front panel I/Os
+5 V	2,2	Base configuration with front panel I/Os and 64 MB
+5 V	2,4	Base configuration with front panel I/Os, 64 MB and DSP

13.3 I/O option Jumper description

A description of the jumpers can be found in the following subsections. Please note, that some of the jumpers may not be used with the actual hardware configuration of your board.

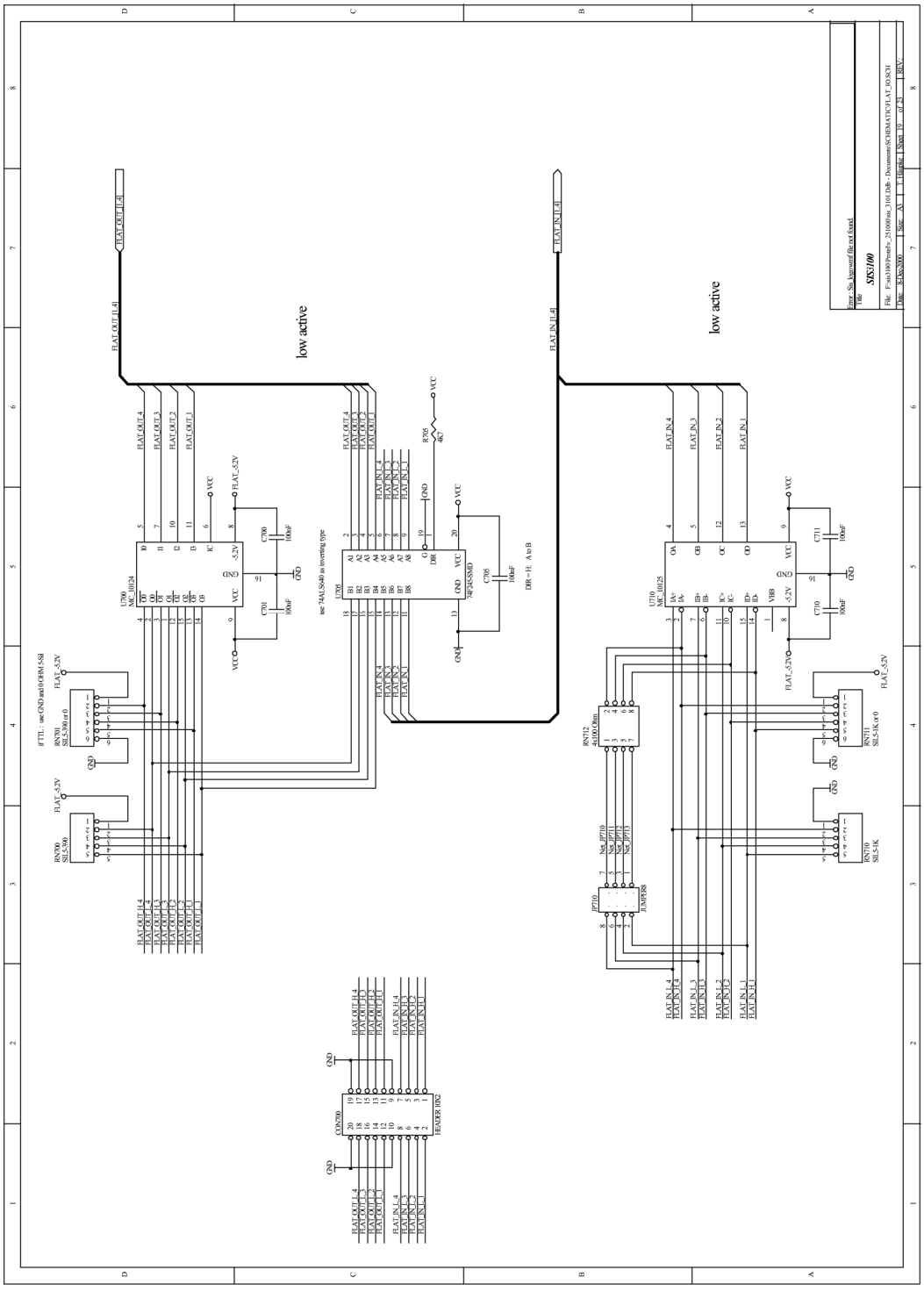
13.3.1 JP710

Termination of flat cable inputs (ECL or high impedance TTL).
Refer to the schematic for the flat cable I/O section for an overview on the complete configuration options (see section 13.3.3).

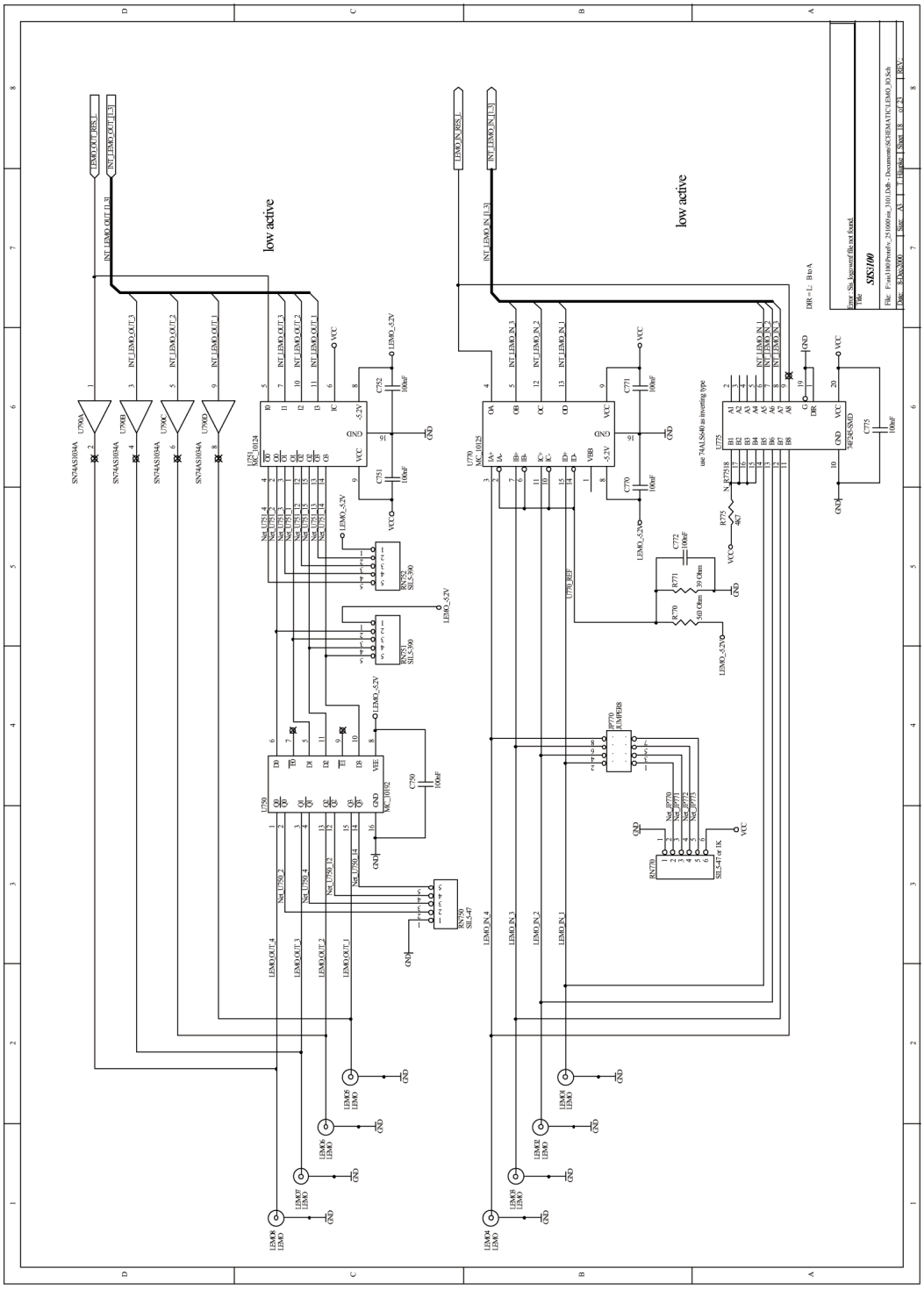
13.3.2 JP770

Termination of LEMO inputs (NIM or 50 Ω TTL). Refer to the schematic for LEMO I/O (see section 13.3.4).

13.3.3 Schematic of flat I/O connector



13.3.4 Schematic of LEMO I/O section



13.4 Boot mechanisms

The firmware of the SIS3100 can be loaded to the boards FPGAs by two different mechanisms. Normally the user will use the factory installed firmware, which will be loaded at power up by default, in some cases it may be of interest however to load special designs or to upgrade the firmware to use extended functionality with the card. The boot options are listed in the table below.

Mechanism	Connector/Chip designator	Hardware
ISP PROM	U501	XC18V04VQ44
JTAG	CON500	9-pin header

13.4.1 ISP PROM

A XILINX XC18V04 ISP (in system programmable) PROM is installed as default firmware load source of the SIS3100. The contents of the serial PROM can be altered via the JTAG port.

13.4.2 JTAG

The XILINX_JTAG connector (CON500) is designed for the use with standard JTAG (Joint Test Action Group) programming tools like the XILINX HW-JTAG_PC can be either used to program the on board EEPROM, or to load firmware to the FPGAs directly for test purposes. Find the pin assignment of the JTAG connector below.

Pin designator	Description
JCC	
GND	Ground
nc	not connected
TCK	Test clock
nc	not connected
TDO	Test data out
TDI	Test data in
nc	not connected
TMS	Test mode select

13.5 Connector types

Find below a list of the used connector types of the SIS3100.

Designation	Function	Manufacturer	Part Number
U200	Optical Link	IBM	42F10SNNAA20 or 30
CON700	Flat cable user I/O	AMP	2-828581-0
LEMO1-8	LEMO user I/O	LEMO	EPL.00.250.NTN
STD-168DIMM	DIMM socket	Berg	61327-31872
CON_D1	SHARC socket long	Samtec	TFM-150-02-S-D-A
CON_D2	SHARC socket short	Samtec	TFM-145-02-S-D-A
P1/P2	VME connector	Harting	02011602101.00

14 Index

+2.5 V 73
+3.3 V 73
+5 V 73
16 MHz 52
3U 69
-5 V 73
A 48
Address Map 25, 44
addressing
  geographical 37
AM 12, 40
AMP 78
arbitration
  timeout 32
arbitration timeout 72
backplane 7
bank 43
Berg 78
BERR 38, 72
  timeout 32
BG 52
BLT16 14
BLT32 15
BLT8 14
boot mechanisms 77
BR 52
bus error 32, 72
bus grant 52
bus mastership 72
bus request 52
bus request level 32
byte 13
byte enable 12, 13
byte enable bit 13
CMC 6, 9
column 43
compact PCI 69
CON 12
CON_D1 78
CON_D2 78
CON500 77
CON700 78
connector 7
  LC 10
  ST 10
connector types 78
Control and Status register 25, 44
control register space 25
cPCI 69
CPU 52
D16 13, 14
D32 13, 15
D64 13
D8 13, 14
DC/DC 73
design 7
DIMM 8, 78
DO8 36
double byte 13
DRAM 8
DSP 8, 51, 53
  available 39, 46
  interrupt 67
  run 39, 46
DU 48
ECL 8, 73, 74
ECON 12
eight byte 13
electrostatic 5
EOT 12
error code 72
ESD 5
FIFO 12
  read 60
  VME read 54, 55
firmware 77
flat cable 8
flat I/O 75
FPGA 77
front panel 7
  cPCI 69
GA 37
GAP 37
gigabit
  hardware 10
  transfer protocol 10
gigalink 72
Harting 78
IBM 78
imask 67
input
  flat cable 29, 30
  LEMO 29, 30
input option 54
input termination 71
INT_A0 63
INT_A5 63
INT_D0 63
INT_D7 63
interrupt
  DSP 67
interrupt generation 36
interrupt sources 67
introduction 5
IRPTEN 67
IRQ 34
  acknowledge cycle 36
  level 36
  vector 36
IRQ0E 67
IRQ1E 67
IRQ2E 67
ISP 77
ISP PROM 77

J/K 37
J10 50, 52
J90 51
JCC 77
JP770 71
JTAG 51, 77
jumper 71
  description 74
  JP_DSP 51
  JP710 74
  JP770 74
  system controller 52
L 48
LC 10
LC connector 10
LD 48
LED 48, 50
  access 48, 70
  data down 70
  data up 70
  DSP user 48, 63
  front panel 7, 48
  link data down 48
  link data up 48
  link up 48, 49, 70
  master 48
  PCB 7, 49
  power 48
  ready 48
  sequencer activity 48
  surface mounted 48
  user 31, 48
  user 0 70
  user 1 70
LEMO 8, 9, 74, 78
  output 66
LEMO I/O 76
linear regulator 73
Link
  data down 49
  data up 49
local bus 72
LU 48
M 48
mastership 32, 52, 72
MBLT64 15
MByte 8
mode1 67
mode2 67
multi master 52
multimode 10
nc 77
NIM *8*, 73, 74
oscillator 52
output
  flat cable 29
  lemo 29
output option 54
P 48
P1 78
P2 78
PCB 48
PCB LED
  SIS1100-CMC 70
  SIS1100-OPT 70
PCI 6
PLX 34, 72
power consumption 73
PROM 77
protocol
  direct VME bus access 18
  general transfer 11
  integrity 10
  mapped VME bus access 21
  remote register transfer 17
  SDRAM transfer 23
  XILINX FPGA 10
protocol error 72
protocol header 12
protocol timeout 72
quad byte 13
R 48
register
  board type 26, 45
  control 39, 46
  DMA_WRITE_BYTE_counter 38
  doorbell 34
  enabled VME IRQ 63
  in/out 25, 29, 30, 34
  input IRQ enable register 63
  internal control 63
  internal Enabled VME IRQ 65
  internal input IRQ enable 67
  internal VME IRQ 65
  internal VME IRQ enable 67
  interrupt control 30, 33
  interrupt generation 25
  LOC_IO_IN_REG 56
  LOC_SEQ_STATUS 55
  LOQ_SEQ_STATUS 60
  OPT_SDRAM SPD 43
  OPT_SDRAM_SPD_EEPROM 43
  OPT-DSP status/control 67
  optical control 28
  optical status 27
  OPT-VME 52
  OPT-VME status 44
  sdram page 47
  SHARC communication 58
  SHARC LOC_IO_OUT_PULSE 57
  SHARC LOC_IO_OUT_REG 57
  slave control 37
  slave status 37
  status 25
  VME adress map 40
  VME IRQ enable register 63
  VME IRQ register 63
  VME sequencer control/status 58
  VME sequencer DMA word count 59
  VME sequencer read FIFO flags and counter 60

release
on request 32
when done 32
request level 32
requester 32
retry 32, 72
return code 72
row 43
S 48
Samtec 78
SDRAM 8, 42, 43
sequencer status 55
SERDES 49
Set Register 39, 46
SFF 10
SHARC 8, 53, 78
address map 53
local bus register 54
SHARC address space 41
SHARC flags 66
side cover 7
single mode 10
SIS1100 68
SIS1100-CMC 69
SIS1100-OPT 69, 71
SIS9200 8, 51, 53
ST 10
ST connector 10
status
latched input 56
sequencer 55
STD-168DIMM 78
SYSRESET 51, 52
system controller 52
TCK 77
TDI 77
TDO 77
termination 74
input 71
timeout
arbitration 32
BERR 32
TMS 77
TTL 8, 9, 74
U 48
U10 52
U200 78
VDIS 52
Vitesse 49
VME 6, 78
address modifier 40
bus extender 52
master 7, 52
retry 72
slave 7, 44
system controller 52
VME transfer
mapped 40
VME64x 7, 37
VME-VME 28
XC18V04 77
XILINX 77
XILINX_JTAG 77