SIS3700
VME ECL FIFO

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: 1.21 as of 12.03.02

Revision Table:

Revision	Date	Modification
0.1	10.09.00	Generation
1.0	09.01.01	First official release
1.01	29.01.01	Jumper 242, jumper 241 closed jumper corresponds to 0
1.10	15.05.01	FERA Modifications (design V_150501) - new FERA Control outputs
1.20	11.06.01	FERA Modifications (design V_110601) - new FERA Control input : FERA Passin_Puls (edge sensitiv)
1.21	12.03.02	added example to VME data stream simulation

1 Table of contents

2 Introduction

The SIS3700 single width (4 TE) 6U (double euro form factor) card, which was designed to capture up to 32-bit wide ECL data streams. The main focus is on VME based acquisition of data generated by Le Croys PCOS and FERA frontend digitising systems.
As all SIS VME cards the board is equipped with the 5 row VME64x VME connectors, a side cover and EMC front panel, as well as the VIPA LED set. For users with VME64x or VME64xP subracks the corresponding extractor handles are available as option.

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/sis3700firm.htm

3 Technical Properties/Features

While the SIS3600 VME multi event latch /input register design focuses on the acquisition of single ECL, NIM or TTL data words or bit patterns like an event number, the SIS3700 design is specialised for the acquisition of event based ECL data streams with different end of event conditions (like timeout e.g.).

Find below a list of key features of the SIS3700.

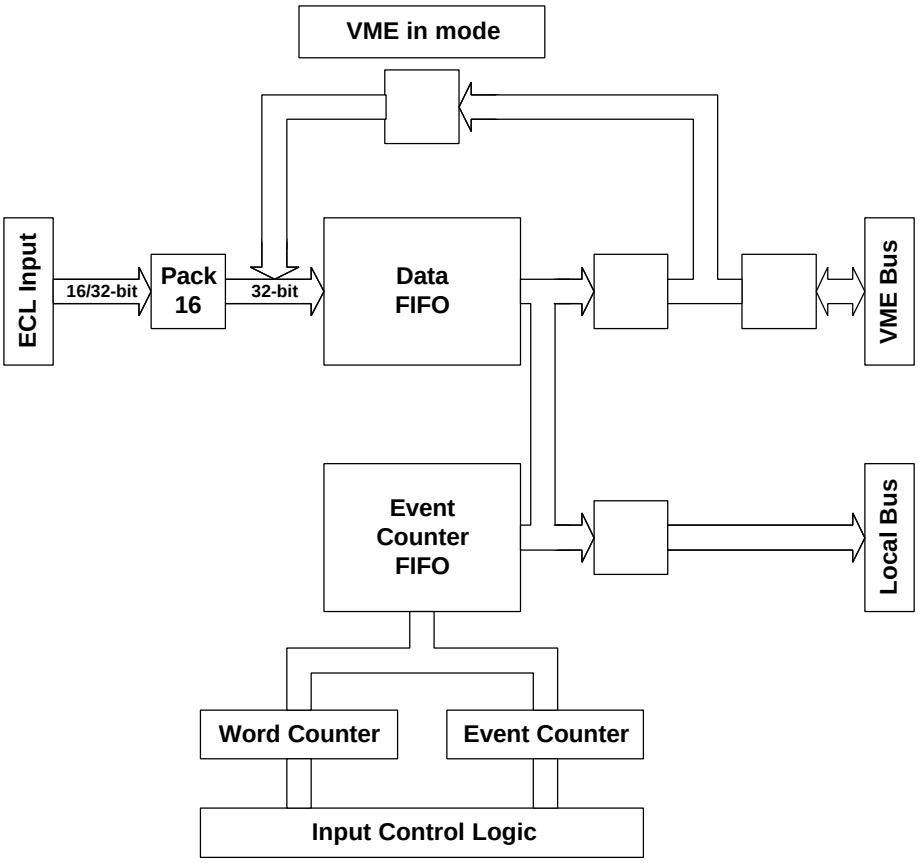
? 32-bit
? ECL signal levels
? Event structure
? 1K event counter and data FIFO (16K available on request)
? programmable end of event condition
? parallel acquisition and readout
? 40 MB/s (32-bit 100 ns/word) max. input data rate
? A24/A32 D32/BLT32
? Base address settable via 2 rotary switches
? 6 LEDs
? VME in (i.e. test) mode

For STR7090 compatibility reasons

? Support of dedicated P2 A/C readout/local bus
? Local bus address selectable via rotary switch

4 Board design and functionality

The functionality of the SIS3700 is illustrated with the simplified block diagram below.



Simplified SIS3700 block diagram

The 16 or 32-bit wide ECL input data stream is stored in the data FIFO after level conversion to TTL . 16-bit data streams are compressed to 32-bit in pack mode . The word counter is incremented with every data word, the event counter is incremented with a leading edge on the Gate input . The two counters are written to the event counter FIFO as soon as the end of event condition is detected. Available end of event conditions depend on the selected mode of operation (FERA e.g.
) . Both the event counter and the data FIFOs can be read out via the VME bus or the local bus . The local bus readout option was implemented to furnish VME and local bus decoupling in systems with one or several STR8090 digital signal processor (DSP) boards.
Data streams can be simulated in VME in mode for software and hardware tests, i.e. this mode can be used to verify the integrity of the data FIFO.

5 VME addressing

5.1 VME base address

The SIS3700 occupies 64KB of the VME address space. It can be configured to respond to standard (A24) and extended (A32) addressing by means of the jumpers enable A24 (EN_A24) and enable A32 (EN_A32). The base address is defined by the four rotary hexadecimal switches SW_A32U, SW_A32L, SW_A24U and SW_A24L. Which switches will be decoded depends on the selected addressing mode as illustrated in the table below.

Address Bits																	
A32	A24	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
x		SW_A32U				SW_A32L				SW_A24U				SW_A24L			
	x	not decoded				not decoded				SW_A24U				SW_A24L			

5.2 Address modifier overview

The SIS3700 supports A24/A32/D32/BLT32 supervisory and non privileged access. Please note, that block transfer (BLT32) is supported for FIFO read access only. . A24 and A32 addressing are enabled with the corresponding jumpers EN_A24 end EN_A32

AM code (hex.)	Description
0F	A32 supervisory block transfer (BLT)
0D	A32 supervisory data access
0B	A32 non privileged block transfer (BLT)
09	A32 non privileged data access
3F	A24 supervisory block transfer (BLT)
3D	A24 supervisory data access
3B	A24 non privileged block transfer (BLT)
39	A24 non privileged data access

6 Address map

The accessibility of part of the SIS3700 registers (i.e. the data and event counter FIFO) depends on the mode of operation. The complete address map and the condition under which a specific address is accessible is listed below-

Offset (hex.)	Access	Function	Access condition
0	read	Data FIFO	VME Output enabled
0	write	Data FIFO	VME Input enabled
4	read	Event/Word counter FIFO	VME Output enabled
8	read	Status register	always
8	write	Control register	always
C	write	Test function register	always

7 Register description

7.1 Data FIFO (0x0)

Incoming data are stored in the data FIFO. In conjunction with the event/word counter information full event information can be retrieved.
The data format (in pack mode) is shown in the table below (events/number of data words as in section 9.4.1). Please note, that a dummy 16-bit word is added to events with an odd number of data words.

Upper word (D31:D16)	Lower word (D15:D0)	Event
datum 0	datum 1	Event N 5 data words
datum 2	datum 3	
datum 4	dummy	
datum 0	datum 1	Event N+1 empty
datum 2	datum 3	Event N+2 8 data words
datum 4	datum 5	
datum 6	datum 7	

FIFO information

FIFO Type	CY7C4225	CY7C4245
FIFO Size	1K	4K

7.2 Event/Word counter FIFO (0x4)

Status , 13-bit wide word counter and 8-bit event number are stored in this FIFO on the event base. The register is 32-bit wide, but the lower 16-bits are used only. The bit assignments of the FIFO as word counter and event counter are listed in the table below.

Bit	Word counter	Event Counter
31	don't care	don't care
...	...	...
16	don't care	don't care
15	FIFO overflow bit	0
14	Timeout	0
13	0	0
12	Word counter bit 12	0
11	Word counter bit 11	0
10	Word counter bit 10	0
9	Word counter bit 9	0
8	Word counter bit 8	0
7	Word counter bit 7	Event counter bit 7
6	Word counter bit 6	Event counter bit 6
5	Word counter bit 5	Event counter bit 5
4	Word counter bit 4	Event counter bit 4
3	Word counter bit 3	Event counter bit 3
2	Word counter bit 2	Event counter bit 2
1	Word counter bit 1	Event counter bit 1
0	Word counter bit 0	Event counter bit 0

Status/word counter and event counter are stored in the event/word counter FIFO in an alternating fashion, as shown in the table below.

Low word (D15-0)
status and event counter event N
event counter of event N
status and event counter event N+1
event counter of event N+1
status and event counter event N+2
event counter of event N+2

7.3 Control register (0x8, write)

Except user LED on/off (which is handled in the test function register), the SIS3700 is configured by writes to the control register. The lowest 8 bits are implemented as 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 has a different location within the register). An undefined toggle status will result from setting both the enable and disable bits for a specific function at the same time.
On read access the control register location represents the status register.

Bit	Function
31	unused
...	unused
9	unused
8	clear data and event/word counter FIFOs, clear event counter
7	disable FIFO input for VME/enable FIFO input for ECL (*)
6	disable FIFO output for VME/enable output for local readout bus (*)
5	disable pack mode (*)
4	enable input timeout logic (*)
3	enable FIFO input for VME/disable FIFO input for ECL
2	enable FIFO output for VME/disable output for local readout bus
1	enable pack mode
0	disable input timeout logic

(*) denotes power up default state

7.4 Status register (0x8, read)

The status register allows to verify the settings made through the control register and to check on the FIFO and input logic status .

Bit	Function	0	1
31	unused		
...			
8	unused		
7	Data FIFO empty flag	not empty	empty
6	Data FIFO full flag	not full	full
5	Event/word counter FIFO empty flag	not empty	empty
4	Input logic busy flag	not busy	busy
3	FIFO input source	ECL input (*)	VME
2	FIFO output destination	Local bus (*)	VME
1	Pack mode	disabled (*)	enabled
0	Input logic timeout	enabled (*)	disabled

7.5 Test function register (0xC, write)

Like the control register this register is implemented in a J/K fashion as far as the activation of the user LED is concerned. The second implemented function is software driven gate generation for test applications. As the end of gate will originate from the timeout condition in this case, no software end of gate is implemented.

Bit	Function
31	unused
...	
8	unused
7	switch off user LED
6	unused
5	unused
4	unused
3	switch off user LED
2	unused
1	unused
0	generate input logic gate pulse (test gate)

8 Hardware description

8.1 Front panel/Input Connectors

The SIS3700 has two 34-pin header connectors and one 20-pin header connector to connect the module to the data stream and control signals. The pin assignments of the connectors are given below.

Control connector (20-pin header)

Pin	Signal	Signal	Pin		
20	Ground	Ground	19		
18	-FERA Clear	+FERA Clear	17		Output
16	-FERA Renout	+FERA Renout	15		Output
14	-FERA WAK	+FERA WAK	13		Output
12	-BUSY	+BUSY	11		Output
10	Ground	Ground	9		
8	-FERA Passin_puls	+FERA Passin_puls	7		Input
6	-FERA Passin	+FERA Passin	5		Input
4	-WST	+WST	3		Input
2	-GATE	+GATE	1		Input

Input connectors (34-pin headers)

Pin	Signal	Signal	Pin
34	Open	Open	33
32	- Data 16	+ Data 16	31
30	- Data 15	+ Data 15	29
28	- Data 14	+ Data 14	27
26	- Data 13	+ Data 13	25
24	- Data 12	+ Data 12	23
22	- Data 11	+ Data 11	21
20	- Data 10	+ Data 10	19
18	- Data 9	+ Data 9	17
16	- Data 8	+ Data 8	15
14	- Data 7	+ Data 7	13
12	- Data 6	+ Data 6	11
10	- Data 5	+ Data 5	9
8	- Data 4	+ Data 4	7
6	- Data 3	+ Data 3	5
4	- Data 2	+ Data 2	3
2	- Data 1	+ Data 1	1

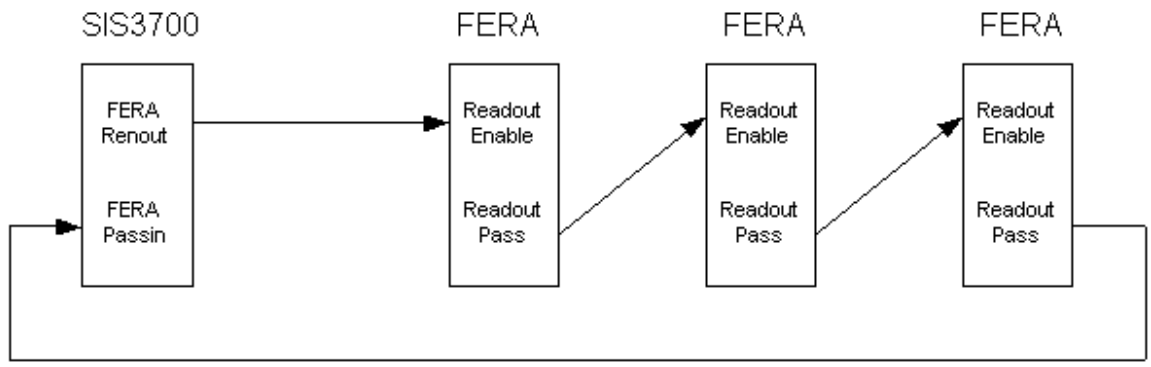
Note: The second data connector has the same pinout (i.e. input channel numbers are shifted by 16 on the input 17-32 connector.

8.2 Control Signal description

The SIS3700 has up to 4 control inputs and up to 4 control outputs.

Description of SIS3700 control signals (V_110601)	
Outputs	Function
FERA Clear	transmitted to the FERA
FERA Renout	transmitted to the FERA data sources in FERA mode (Readout Request)
FERA WAK	Write Acknowledge
BUSY	Busy output - set with leading edge of gate and cleared with end of event condition and also set with almost full condition of data and event counter FIFO - can be used to inhibit new triggers
Inputs	Function
FERA Passin_puls	signals end of event condition in FERA mode (edge sensitive)
FERA Passin	signals end of event condition in FERA mode (level sensitive)
WST	write strobe
Gate	arms input logic, clears word counter, increments event counter and sets busy 1 output

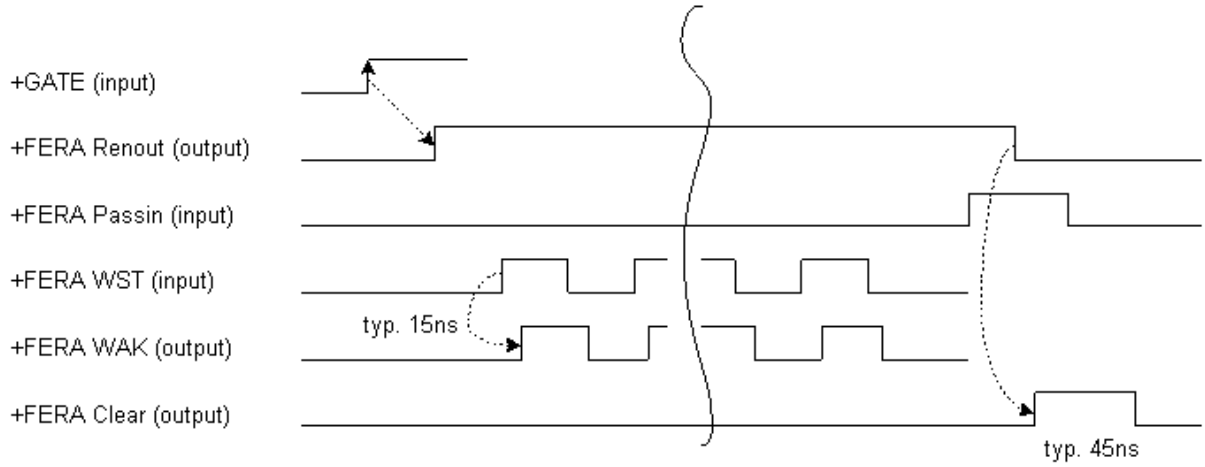
8.2.1 Configuration example with a chain of three FERA modules



FERA Readout Enable/Pass configuration

8.2.2 FERA control input/output timing diagram

The figure below illustrates the timing of the control signals during the FERA readout cycle.



FERA Control Input/Output timing

8.3 Input termination

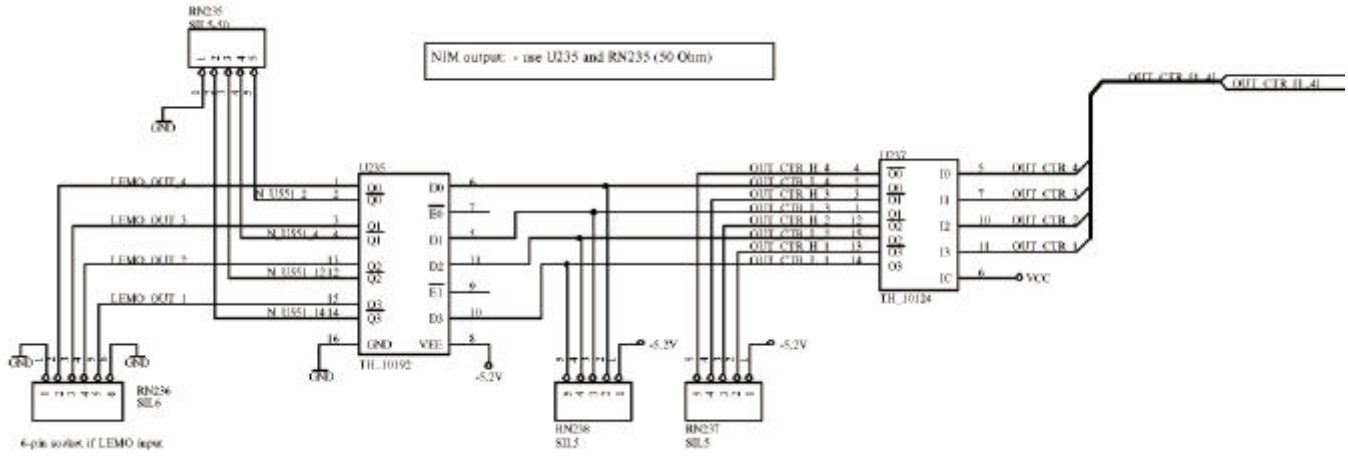
Input termination is provided by socketed single inline (SIL) resistor networks to allow for maximum flexibility.

8.3.1 Control signals

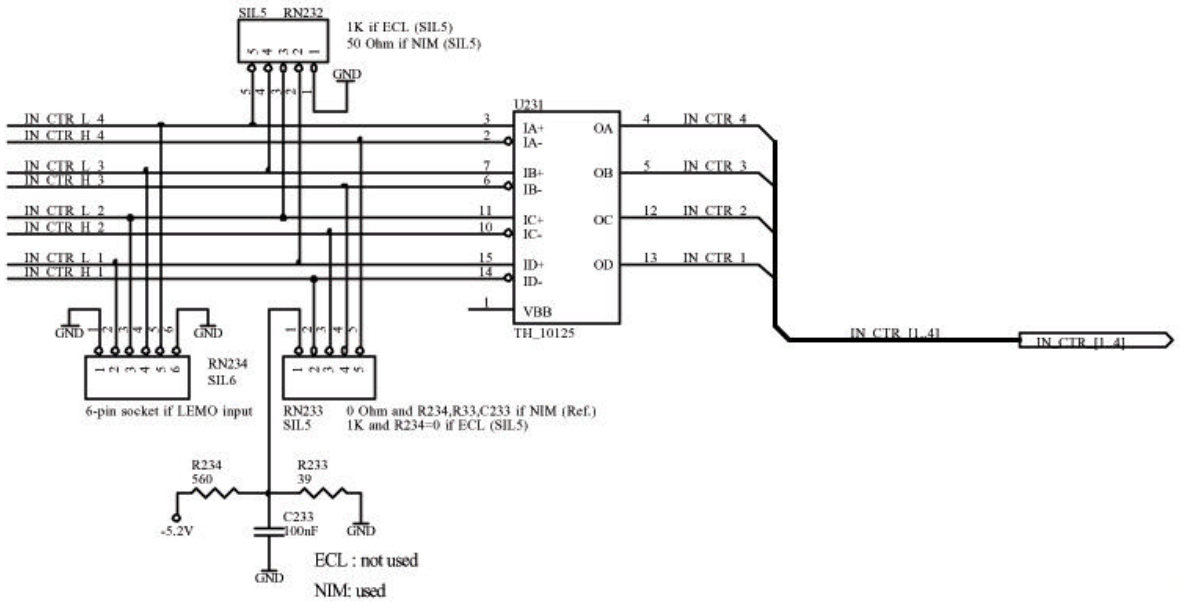
Control inputs and outputs are configured and terminated by resistor networks in sockets RN231 to RN238 as shown in the table below.

SIL network	Description
RN231	Input termination control input 1-4 (with ECL)
RN232	Input termination control input 1-4 (with NIM), GND pulldown with ECL
RN233	clamp to -5.2 V with ECL inputs
RN234	GND connection with LEMO inputs
RN235	Output termination with LEMO
RN236	GND connection with LEMO outputs
RN237	Clamp to –5.2 V Control output 1-4
RN238	Clamp to –5.2 V Control output 1-4

Schematic of control output section:



Schematic of control input section:



8.3.2 Data inputs

The inputs of the SIS3700 are terminated and clamped via 16 resistor networks as listed in the table below.

SIL network	Description
RN215	Input 1-8 (clamp to –5.2 V)
RN216	Input 1-8 (clamp to Ground)
RN217	Input 9-16 (clamp to –5.2 V)
RN218	Input 9-16 (clamp to Ground)
RN225	Input 17-24 (clamp to –5.2 V)
RN226	Input 25-32 (clamp to Ground)
RN227	Input 17-24 (clamp to –5.2 V)
RN228	Input 25-32 (clamp to Ground)
RN211	Termination input 1-4
RN212	Termination input 5-8
RN213	Termination input 9-12
RN214	Termination input 13-16
RN221	Termination input 17-20
RN222	Termination input 21-24
RN223	Termination input 25-28
RN224	Termination input 29-32

8.3.3 Non pack mode termination

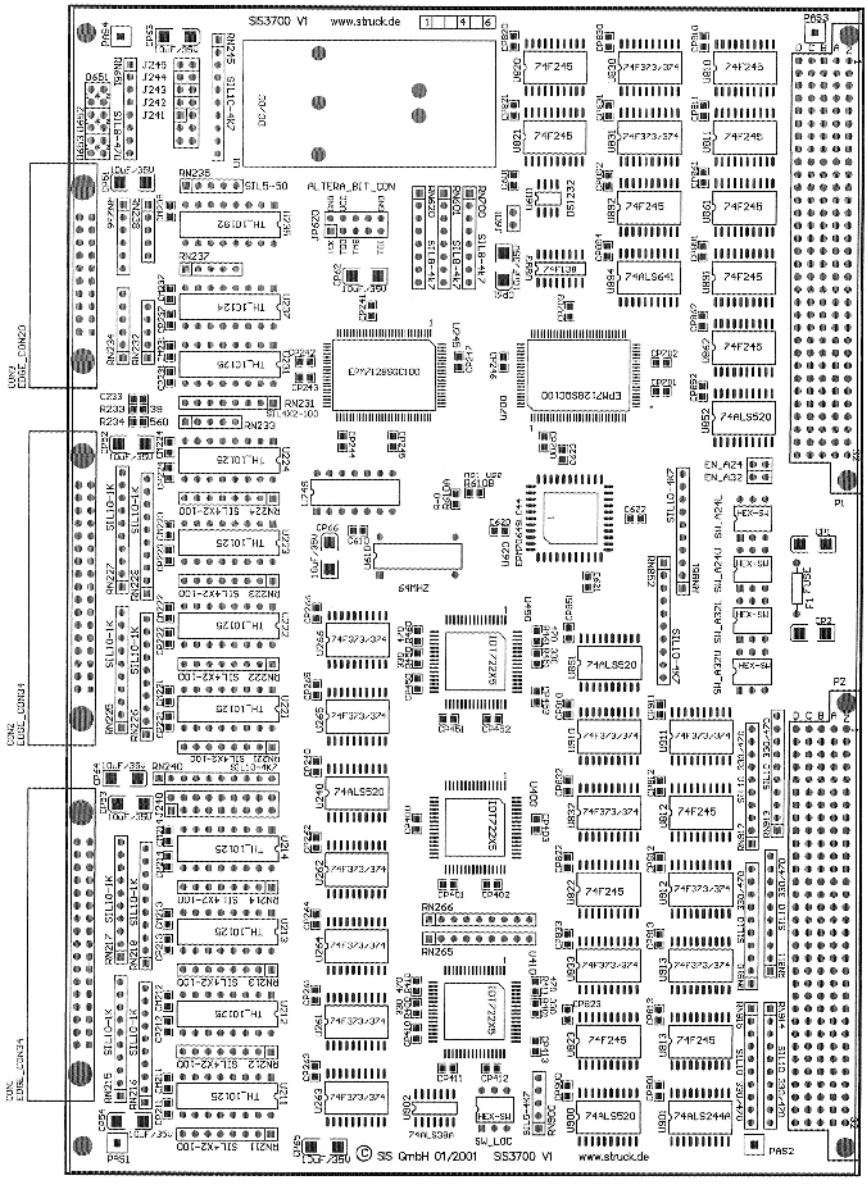
The upper 16 data bits are pulled down via resistor networks RN265 and RN266 in non pack 16-bit mode, i.e. the user will see zeroes in the upper 16 data bits.

8.4 LEDs

Six LEDs are implemented to visualise part of the SIS3700 status. The logic of the module will be initialised upon power up. All LEDs of the board except the USR LED will be on during this phase (about 1 to 2s) as a LED self test .

LED label	Color	Description
VME	yellow	Lit when the SIS3700 is accessed from VME
LOC	yellow	Lit when the SIS3700 is accessed from the local bus
EMP	green	lit when both FIFOs are empty
/EMP	green	lit when one or both FIFOs are not empty
BSY	red	on when a gate is present
USR	red	user LED, to be switched on/off via test function register

8.5 Board Layout



9 Local Bus

The local bus was implemented to allow for the parallel readout of frontend data into a event builder and readout of data from the event builders VME slave into a VME master. Originally this concept was implemented for the DSP based STR8090 in conjunction with an arbiter module. With readout controllers like the SIS3100 PCI to VME the SIS3700 FIFO can be read out via VME, as the data are shipped off to a PC or workstation via an optical link.

9.1 Local Bus Address Selection

Up to 16 SIS3700 can be operated on a local bus. The local bus address is defined by the hexadecimal rotary switch designated SW_LOC.. The local bus address can range from 0x0 to 0xF.

9.2 Local Bus Termination

The local bus must be terminated on both ends of the cable backplane . On one side of the cable the termination is implemented on the arbiter module , on the other side of the cable local bus termination must be established on the last SIS3700 module in the chain. Six 10-pin SIL (Single InLine) resistor networks of type 10x-1 with a value of yyy ? have to be installed in sockets (labelled RNxxx, ...) in the vicinity of the P2 connector. A rectangular solder pad marks the common pin of the resistor network.

9.3 Local Bus Pinout

The local bus is implemented on rows A and C of the P2 connector with a cable backplane . The cable backplane consists of a 64 wire flat cable and 96-pin female DIN connectors. and is installed on the rear side of the P2 backplane of the VME crate. The pin assignment of the local bus is shown in the table below.

Pin	Row A	Row C
1	GND	D0
2	D2	D1
3	D3	GND
4	GND	D4
5	D6	D5
6	D7	GND
7	GND	D8
8	D10	D9
9	D11	GND
10	GND	D12
11	D14	D13
12	D15	GND
13	GND	D16
14	D18	D17
15	D19	GND
16	GND	D20
17	D22	D21
18	D23	GND
19	GND	D24
20	D26	D25
21	D27	GND
22	GND	D28
23	D30	D29
24	D31	GND
25	GND	A0
26	A2	A1
27	A3	GND
28	GND	AS_L
29	SKIP	DS_L
30	CLEAR	OUT_RESERVE
31	GND	VALID_EV
32	END_EV	OVERFLOW_ERR

9.4 Local Bus Readout with STR8090

The SIS3700 is optimised to push data into the STR8090 module or similar hardware during the data phase of the local bus event readout. The data phase is preceded by the fetch word counter/event number phase. The local bus readout controller (i.e. STR8090) accesses the event counter FIFO of the first SIS3700 and waits for the word counter to become available (timeout?). The local bus interface switches to the data FIFO after the word counter has been transferred also. During the data phase the end of event is flagged by the SIS3700 and the STR8090 has the possibility to compare the word counter from the event counter FIFO to the number of data words that were actually transferred to the input FIFO of the STR8090..
The data phase transfer speed between a STR8090 and the SIS3700 was measured to be xx MB/s, the overhead to read word counter and event number from the module is in the order of xx ?s.
As the STR8090 module is obsolete, FIFO readout over VME with the SIS3100 PCI to VME interface has to be considered as an alternative. The anticipated block transfer speed is xx MB/s, the overhead is expected to be in the order of xx ?s.

9.4.1 Event Structure

Event counter FIFO data (i.e. word counter and event counter) and data FIFO contents from one SIS3700 will be stored in consecutive DSP input FIFO locations on the STR8090. As the event number is stored for dataless events with word counter 0, event synchronisation in multiple module setups is straightforward. Find below a table with the data structure as it would occur from the readout of a single SIS3700 in the input FIFO of the STR8090 (pack mode).

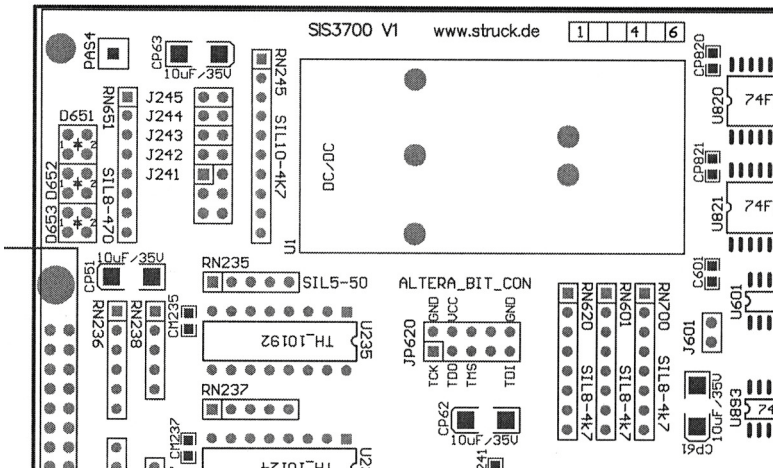
Upper (D31 ... D16)	Lower (D15 ... D0)	Comment
dummy	word counter event N	event N 5 data words
dummy	event counter event N	
datum 0	datum 1	
datum 2	datum 3	
datum 4	dummy	
dummy	word counter event N+1	event N+1 no data word
dummy	event counter event N+1	
dummy	word counter event N+2	event N+2 8 data words
dummy	event counter event N+2	
datum 0	datum 1	
datum 2	datum 3	
datum 4	datum 5	
datum 6	datum 7	

10 Jumper description

Some functions of the SIS3700 are defined by jumper settings. A summary of all jumpers is given in the table below.

Jumper Designation	Description
EN_A24	enable A24 addressing
EN_A32	enable A32 addressing
J240	Artificial PCOS end marker, APE
J241	Timeout selection
J243	PCOS/FERA mode selection
J244	PCOS special pack mode
J245	enable artificial PCOS end marker
J601	VME reset behaviour

Find below the vicinity of the SIS3700 with the DC/DC converter, jumper locations J241-J245 and J601 can be seen.



10.1 J240 (Artificial PCOS end marker, APE)

9 bits of the PCOS artificial end marker can be selected by the jumpers of jumper array J240. The incoming data word is compared to the APE setting in PCOS mode if the artificial PCOS end marker end of event condition is enabled (via jumper J245).
A closed jumper of array J240 is interpreted as 1

J240	
Bit	Setting
15	0
14	APE 9
13	APE 8
12	APE 7
11	APE 6
10	APE 5
9	APE 4
8	APE 3
7	APE 2
6	APE 1
5	0
4	0
3	0
2	0
1	0
0	x

Explanation:

0	bit has to equal 0
APE 1 – APE 9	compared to setting of bit on J240
x	not compared

Note: The position of APE 1 is designated with a rectangular mark on the PCB silk screen

10.2 J241 (Timeout selection, APE)-

Timeout and artificial PCOS end marker are the two possible end of event conditions in PCOS mode. The timeout value can be selected by three jumpers of jumper array J241 as listed in the table below.
A closed jumper corresponds to a logic 0.

J241			
Bit 2	Bit 1	Bit 0	Timeout value in ?s
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8 (factory default)
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Note: The position of Bit 0 is designated with a rectangular mark on the PCB silk screen

10.3 J242 16/32-bit wide input mode

The input width (16 or 32-bit) is defined with this jumper. A closed jumper selects the 16-bit wide input configuration.

10.4 J243 select PCOS/FERA mode

The user can choose among the PCOS and the FERA mode by J243. A closed jumper selects FERA mode, an open jumper selects PCOS mode. The possible end of event conditions depend on the selected mode of operation.

End of event conditions			
Mode	Timeout	APE	PASS_IN
PCOS	x	x	0
FERA	x	0	x

10.5 J244 enable PCOS special pack mode

The address of clusterised PCOS data will always show up in the second word in pack mode if PCOS special width mode is enabled by closing J244.

10.6 J245 enable APE

The artificial PCOS end marker end of event condition is defined by jumper array J240 and activated by J245. APE is enabled if the jumper is in closed.

10.7 J601 VME reset behaviour

The SIS3700 will reset on occurrence of a VME SYS_RESET if jumper J601 is closed.

11 VME data stream simulation

The SIS3700 allows the user to simulate data streams from arbitrary front ends in VME input test mode . This functionality is useful for several purposes. From the VME readout point of view it allows the user to check data integrity of the complete VME module chain (i.e. VME master, SIS3700 and VME crate), from the software point of view it allows for debugging of higher level trigger or other software with a set of well defined events.
The procedure for a data stream simulation with one SIS3700 consists of the following steps:

11.1 Sequence

1. Enable VME input test mode and FIFO to VME output
2. Clear FIFOs
3. Loop over:

? disable timeout
? set test gate
? write to data FIFO
? enable timeout
? wait until end of timeout (by polling event/word counter FIFO not empty status e.g.)

11.2 Shell Example

Find below a commented example session with a SIS1100/3100 PCI to VME interface and a SIS3700 module with base address 0 and A32 enabled.

```
/* enable VME input test mode and FIFO to VME output */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 8 0xc
vme_A32D32_write: return_code = 0x00000000
/* clear FIFOs */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 8 0x100
vme_A32D32_write: return_code = 0x00000000
/* disable timeout */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 8 0x1
vme_A32D32_write: return_code = 0x00000000
/* generate gate pulse */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 0xC 0x1
vme_A32D32_write: return_code = 0x00000000
/* write first datum */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 0x0 0x12345678
vme_A32D32_write: return_code = 0x00000000
/* write second datum */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 0x0 0x87654321
vme_A32D32_write: return_code = 0x00000000
/* enable timeout */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_write_d32 /tmp/sis1100 0x8 0x10
vme_A32D32_write: return_code = 0x00000000
/* read status /
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x8
vme_A32D32_read: return_code = 0x00000000
vme_A32D32_read: data = 0xffffff0c
/* read event counter FIFO event size*/
/root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x4
vme_A32D32_read: return_code = 0x00000000
vme_A32D32_read: data = 0xffff4002
/* read event counter FIFO event number */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x4
vme_A32D32_read: return_code = 0x00000000
vme_A32D32_read: data = 0xffff0001
/* read event counter FIFO now empty -> bus error */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x4
vme_A32D32_read: return_code = 0x00000211
vme_A32D32_read: data = 0x40138828
/* read first datum */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x0
vme_A32D32_read: return_code = 0x00000000
vme_A32D32_read: data = 0x12345678
/* read second datum */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x0
vme_A32D32_read: return_code = 0x00000000
vme_A32D32_read: data = 0x87654321
/* 3rd read from data FIFO now empt -> bus error */
root@mki:/home/mki/sis1100/vme_simple_routines > ./vme_read_d32 /tmp/sis1100 0x0
vme_A32D32_read: return_code = 0x00000211
vme_A32D32_read: data = 0x40138828
```

12 Index