SIS4100
VME to FASTBUS
Interface

User Manual

SIS GmbH
Moorhof 2d
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: 1.04 as of 20.08.99

Revision Table:

Revision	Date	Modification
0.1	01.02.99	Generation
1.0	26.07.99	First official release
1.01	27.07.99	some additions
1.02	29.07.99	default address setting
1.03	09.08.99	bug fixes in VME address map
1.04	20.08.99	Pedestal subtracted event length determination

Table of contents

1 Introduction

The FASTBUS standard was mainly introduced to cover the needs of high energy physics experiments around 1983. The main difference to the VME standard, which came up at about the same time is the bigger form factor and the presence of more supply voltages as well as extended addressing capabilities and a definition of a way to implement multi crate systems. As the FASTBUS standard did never play a role in industry applications, it suffered from a lack of standard readout controllers (or masters). As more and more institutes and labs decided to migrate towards VME, first VME to FASTBUS interfaces became available. The most straightforward implementation of such an interface is to build a FASTBUS master interfacing to a small VME card cage. The master is programmed through a COTS (Commercial Of The Shelf) VME CPU in these implementations, interconnects of bigger systems are typically established through the network capabilities of the CPU, like fast Ethernet , instead of the cluster or segment interconnects, which tended to be a potential problem source in bigger FASTBUS setups.
The SIS4100 NGF (Next Generation FASTBUS) is a new interface, combining the very successful VME to FASTBUS interfacing concept with the special capabilities of earlier master implementations. The main characteristics are:

- VME to FASTBUS interface
- internal VME crate (with 3 VME slots)
- on the fly pedestal subtraction (sparsification) and channel remapping
- up to two SHARC DSP piggy packs for higher level trigger processing

The SIS4100 is designed with experiments in mind, which want to save the investment in the frontend modules and crates and upgrade to higher trigger/readout rates at the same time. It is perfectly suited to run a single crate lab system as well as a large scale experiment with a combination of VME and FASTBUS (and possibly other) data acquisition hardware.
To meet the demands of experiments at the GSI, BNL, JLAB, TRIUMF and SPring8, a firmware mode with downward compatibility to the STR340 SFI FASTBUS master was developed.
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.

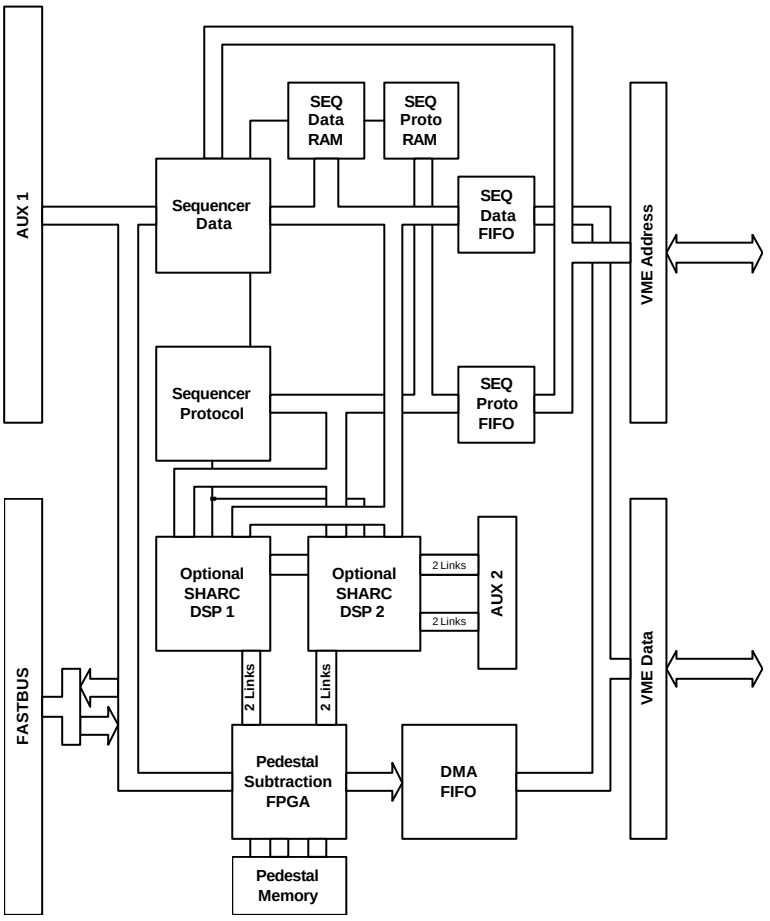
2 NGF Working Principle

FASTBUS transactions are handled by a sequencer on board of the SIS4100. The sequencer commands are passed to the sequencer either from a sequencer RAM or directly. In both cases the NGF acts as a VME slave during this configuration portion of the transaction. The VME slave addressing mode is A24/D32. or A32/D32The readout FB data are passed to a VME slave in the NGFs card cage, in the minimum setup the VME slave memory resides on the VME CPU. During this portion of the transfer the NGF acts as a VME master, the addressing mode of the transfer is A32/D32.

As an alternative the readout data can be shipped off through an ECL bus over the FASTBUS AUX backplane, in this approach a system with more predictable readout behaviour then in the Ethernet case can be achieved.

If one or two of the optional SHARC DSPs are installed, data can also be passed over the SHARC link ports. This may be of interest in the case of the need for fast information exchange between different frontend systems, one may think of tasks like matching drift chamber tracking with calorimeter energy information.

A simplified block diagram of the NGF is shown below.



3 Design

3.1 Features

The key functionality of the NGF comprises:

- FASTBUS list sequencer
- sequencer RAM and FIFO
- pedestal subtraction unit with pedestal and remap memory
- one or two optional higher level trigger/filter SHARC DSP(s)
- 4 ECL in/4 ECL outputs
- 4 NIM/LEMO in/ 4 NIM LEMO outputs
- 4 TTL outputs
- 1 NIM reset input
- 16 LEDs
- VME card cage

The NIM, TLL and ECL outputs can be set under sequencer control, and in return the NIM and ECL inputs can be used to control list execution of the sequencer, i.e. a sequencer list can start upon an external signal with minimum overhead as no operating system latency is involved.

3.2 Mechanical concept

The SIS4100 consists of 4 PCBs, the VME card cage and the mechanics.
The PCBs are:

- mainboard
- VME backplane
- front panel connector card
- aux SHARC link and power card

3.3 VME properties

The NGF is basically a VME crate, which is embedded in a FASTBUS module.
The basic properties are:

- 6U form factor (double Eurocard)
- 3 VME slots (12 TE)
- auto daisy chain backplane
- +5 V, -12 V and +12 V power supply

Due to mechanical restrictions access to the P2 connector of slots 2 and three of the card cage is not possible. Signals of the P2 connector of slot 1 are routed to the first AUX slot (with connector on the mainboard) of the NGF and can be used to interface via the VDB bus e.g..

4 VME slave port

4.1 VME Address space

The NGF is an A32/D32 or A24/D32 VME slave, as bits 23-20 can be set by means of a rotary switch on the printed circuit board, the unit occupies an address space of 0x100000 bytes.

4.2 VME Address map

Only part of the 0x100000 bytes of the NGFs address space is actually occupied by system resources. Find below a table with the implemented addresses.

Address	Read Function	Write Function
0x01x00	Internal FB I/O Bus	VME Out Signal Register
0x01x04	FB Last Primary Address Register	KA Clear VME Out Signal Register
0x01x10		Internal AUX Port Register
0x01x14		KA Generate AUX B40 Pulse
0x02x00	FB Timeout Register	FB Timeout Register
0x02x04	FB Arbitration Level Register	FB Arbitration Level Register
0x02x08	FB protocol register	-
0x02x0C	Sequencer FIFO and ECL/NIM input register	-
0x02x10	VME IRQ and Level Register	VME IRQ and Level Register
0x02x14	VME IRQ Source and Mask Register	VME IRQ Source and Mask Register
0x02x18	Next Sequencer RAM Address Register	Next Sequencer RAM Address Register
0x02x1C	Last Sequencer Protocol Register	KA Reset Register Group LCA2
0x02x20	Sequencer Status Register	KA Sequencer Enable
0x02x24	FB Status Register 1 (Arbitration/Primary Status)	KA Sequencer Disable
0x02x28	FB Status Register 2 (Data Cycle/DMA Status)	KA Sequencer RAM Load Enable
0x02x2C		KA Sequencer RAM Load Disable
0x02x30		KA Sequencer Reset
0x02x38		KA Clear Sequencer CMD Flag
0x02x3C		KA Enable Pedestal Subtraction
0x02x40		KA Disable Pedestal Subtraction
0x04xxx	SEQ2VME FIFO	
0x1xxxx		VME2SEQ FIFO
0x2x000	Pedestal Memory Pointer Register	Pedestal Memory Pointer Register
0x2x004	Pedestal and Remap Register	Pedestal and Remap Register
	SHARC 1	SHARC 1
	SHARC 2	SHARC 2

Note: The short hand KA designates a key address.

Address nibbles filled with x can have an arbitrary address value in this location to minimise possible VME master cache problems and to increase flexibility on the user side.
Example: The FB Timeout Register (0x2x00) can be read from 0x2000, 0x2100, 0x2200, ..., 0x2900

4.3 VME Register description

4.3.1 Internal FASTBUS I/O Bus (0x1000, Read)

Data can be read from a user AUX card through the trigger interface via this register.
A description of the trigger interface can be found in section 9.1

Bit	Function
31	Internal FB IO bus bit 31
...	...
0	Internal FB IO bus bit 0

4.3.2 VME out register (0x1000, Write)

All front panel outputs and the user LEDs as well as three signals on the auxiliary connector can be set and cleared through this register. The individual signals are implemented as J/K style register.

Note: the data of the VME out register are ored with the data of the sequencer out register

Bit	Function
31	reserved
30	clear auxiliary connector output A45
29	clear auxiliary connector output A28
28	clear auxiliary connector output A10
27	clear NIM output 4
26	clear NIM output 3
25	clear NIM output 2
24	clear NIM output 1
23	clear ECL output 1
22	clear ECL output 2
21	clear ECL output 3
20	clear ECL output 4
19	clear TTL output 4 and user LED 4
18	clear TTL output 3 and user LED 3
17	clear TTL output 2 and user LED 2
16	clear TTL output 1 and user LED 1
15	reserved
14	set auxiliary connector output A45
13	set auxiliary connector output A28
12	set auxiliary connector output A10
11	set NIM output 4
10	set NIM output 3
9	set NIM output 2
8	set NIM output 1
7	set ECL output 1
6	set ECL output 2
5	set ECL output 3
4	set ECL output 4
3	set TTL output 4 and user LED 4
2	set TTL output 3 and user LED 3
1	set TTL output 2 and user LED 2
0	set TTL output 1 and user LED 1

4.3.3 FASTBUS last primary address register (0x1x04, Read)

This read only register is updated with the FASTBUS primary address during the primary address cycle . It can be used to track down the faulty FASTBUS slave if list command execution fails or other debugging purpose. The 32-bits of the register hold the 32 AD lines, the power up reset value is 0x0.

4.3.4 Internal AUX-port register (0x1x10, Write)

Data can be written to a user AUX card through the trigger interface via this register.
A description of the trigger interface can be found in section 9.1

Bit	Function
31	AUX port bit 31
...	...
0	AUX port bit 0

4.3.5 FASTBUS timeout register (0x2x00, READ/Write)

The FASTBUS standard distinguishes between the so called short timeout and long timeout. The later was basically implemented for addressing beyond segment interconnects, a crate interconnect method which is no longer needed in the presence of fast ethernet and other interconnections.

The short timeout is used during primary and address data cycles as long as WT is not asserted. The long timeout is used during arbitration and during primary address and data cycles if WT is asserted.

The FASTBUS timeout register is of read/write type, the bit assignments and the corresponding timeout values are given in the two tables below.

Bit	R/W	Function
31	R	none, read as 1
...	...	...
8	R	none, read as 1
7	R/W	disable long timeout
6	R/W	long timeout bit 2
5	R/W	long timeout bit 1
4	R/W	long timeout bit 0
3	R/W	disable short timeout
2	R/W	reserved, read as 0
1	R/W	short timeout bit 1
0	R/W	short timeout bit 0

Bit 2	Bit 1	Bit 0	short timeout	long timeout
0	0	0	2 µs	64 µs
0	0	1	4 µs	256 µs
0	1	0	8 µs	2 ms
0	1	1	16 µs	16 ms
1	0	0		128 ms
1	0	1		1 s
1	1	0		8 s
1	1	1		32 s

4.3.6 FASTBUS Arbitration level register (0x2x04, read/write)

The NGFs arbitration level and arbitration behaviour can be programmed through this read/write register. The bit assignments are given in the table below.

Bit	R/W	Function
31	R	none, read as 1
...	...	...
8	R	none, read as 1
7	R/W	Assured Access (FAIR)
6	R/W	reserved, read as 0
5	R/W	arbitration level bit 5
4	R/W	arbitration level bit 4
3	R/W	arbitration level bit 3
2	R/W	arbitration level bit 2 (set as reset default)
1	R/W	arbitration level bit 1
0	R/W	arbitration level bit 0

The reset value is 0xFFFFFF04

4.3.7 FASTBUS protocol register (0x2x08, read)

This read only register reflects the status of a number of FASTBUS signals. the bit assignment is given in the table below.

Bit	FASTBUS Signal
31	none, read as 1
...	...
16	none, read as 1
15	Reset Bus (RB)
14	Bus Halt
13	Service Request (SR)
12	Arbitration Logic
11	Grant Acknowledge (GK)
10	Arbitration Grant (AG)
9	Arbitration Request Inhibit (AI)
8	Arbitration Request (AR)
7	Enable Geographic (EG)
6	none, read as 1
5	Address Acknowledge (AK)
4	Data Acknowledge (DK)
3	Wait (WT)
2	slave status bit 2 SS2
1	slave status bit 1 SS1
0	slave status bit 0 SS0

4.3.8 Sequencer FIFO flag and ECL/NIM input register (0x2x0C, read)

This read only register holds the status of the levels of the NIM and ECL inputs as well as the level of the AUX_B42 input. In addition the lowest 8 bits reflect the status of the VME to sequencer and the sequencer to VME FIFOs.

Bit	Function
31	none, read as 1
...	...
16	none, read as 1
15	input level of AUX_B42
14	input level of NIM3
13	input level of NIM2
12	input level of NIM1
11	input level of ECL4
10	input level of ECL3
9	input level of ECL2
8	input level of ECL1
7	sequencer to VME FIFO full (synchronous)
6	sequencer to VME FIFO half full (asynchronous)
5	sequencer to VME FIFO almost empty (asynchronous)
4	sequencer to VME FIFO empty (synchronous)
3	VME to sequencer FIFO full (synchronous)
2	VME to sequencer FIFO half full (asynchronous)
1	VME to sequencer FIFO almost empty (asynchronous)
0	VME to sequencer FIFO empty (synchronous)

(asynchronous) explanation
(synchronous)

The used FIFO type is IDT72225 with a size of 1K or IDT72245 with a size of 4k , the almost empty flag is set for: $0 \leq$ number of words in FIFO < 128

4.3.9 VME IRQ level and vector register (0x2x10, Read/Write)

The VME interrupt behaviour of the NGF is controlled through this register. Via the internal VME IRQ bit it is possible to check the presence of an enabled interrupt source without actually using the VME interrupt mechanism at all. This maybe useful for debugging reasons or if interrupt handling is to be avoided at all. If the VME interrupt generation is enabled the lower 8 bits of the register define the interrupt vector, the level of the interrupt is defined by bits 8 through 10. The type of the interrupter of the SIS4100 is D08(O).

Bit	Name	Function
31-16	reserved	read as 1
15	VME_IRQ	read as 1 if VME IRQ is set, otherwise 0
14	I_VME_IRQ	read as 1 if internal VME IRQ is set, otherwise 0
13	reserved	read as 0
12	reserved	read as 0
11	VME_IRQ_ENABLE	generation of VME IRQ disabled (0)/enabled(1)
10	VME_IRQ_LEV2	VME IRQ level bit 2
9	VME_IRQ_LEV1	VME IRQ level bit 1
8	VME_IRQ_LEV0	VME IRQ level bit 0
7	IRQ_VECTOR7	VME IRQ vector bit 7
6	IRQ_VECTOR6	VME IRQ vector bit 6
5	IRQ_VECTOR5	VME IRQ vector bit 5
4	IRQ_VECTOR4	VME IRQ vector bit 4
3	IRQ_VECTOR3	VME IRQ vector bit 3
2	IRQ_VECTOR2	VME IRQ vector bit 2
1	IRQ_VECTOR1	VME IRQ vector bit 1
0	IRQ_VECTOR0	VME IRQ vector bit 0

4.3.10 VME IRQ source and mask register (0x2x14, Read/Write)

A variety of VME interrupt sources is implemented on the NGF. They can be enabled and disabled/cleared via this register, which is implemented as J/K register. The enable and IRQ status of the sources can be read back.

Bit	Read	Write
31-16	1	not implemented
15	SEQ_DISABLE_IRQ_FLAG	disable/clear sequencer disable source
14	SEQ_CMD_FLAG	disable/clear sequencer cmd flag source
13	SR_IRQ_FLAG	disable/clear SR (FB service request) source
12	AUX_B42_IRQ_FLAG	disable/clear AUX_B42 source
11	ECL1_IRQ_FLAG	disable/clear ECL1 IRQ source
10	NIM3_IRQ_FLAG	disable/clear NIM3 IRQ source
9	NIM2_IRQ_FLAG	disable/clear NIM2 IRQ source
8	NIM1_IRQ_FLAG	disable/clear NIM1 IRQ source
7	SEQ_DISABLE_ENABLE	enable sequencer disable source
6	SEQ_CMD_FLAG_ENABLE	enable sequencer cmd flag source
5	SR_ENABLE	enable SR (FB service request) source
4	AUX_B42_ENABLED	enable AUX_B42 source
3	ECL1_IRQ_ENABLED	enable ECL1 IRQ source
2	NIM3_IRQ_ENABLED	enable NIM3 IRQ source
1	NIM2_IRQ_ENABLED	enable NIM2 IRQ source
0	NIM1_IRQ_ENABLED	enable NIM1 IRQ source

4.3.11 Next sequencer RAM address register (0x2x18, Read/Write)

The next sequencer RAM address register is used to set the memory pointer for RAM list loading, after completion of a RAM list (or loading of a list) it contains the last address of the list. The lowest 8 bits of the start address are always equal 0, i.e. RAM lists always start at 0x100 boundaries (0x000, 0x100, 0x200, ...). As the size of the sequencer RAM is 32 K x 48-bit (the sequencer command consists of 16-bit to define the action and of 32-bits of data), the maximum number of lists which can be stored in sequencer RAM is limited to 0x80.
The access to the registers data is limited:

- The sequencer has to be disabled and not in RAM load mode if a new address is written to the next sequencer RAM address register.
- The address pointer can be read, if the sequencer is not in RAM mode and not in RAM load mode.

Find below a table with the bit assignments of the register.

Bit	Function
31-16	read as 1
15	sequencer RAM address bit 15
14	sequencer RAM address bit 14
13	sequencer RAM address bit 13
12	sequencer RAM address bit 12
11	sequencer RAM address bit 11
10	sequencer RAM address bit 10
9	sequencer RAM address bit 9
8	sequencer RAM address bit 8
7	sequencer RAM address bit 7 (loaded from the sequencer as 0 at list start)
6	sequencer RAM address bit 6 (loaded from the sequencer as 0 at list start)
5	sequencer RAM address bit 5 (loaded from the sequencer as 0 at list start)
4	sequencer RAM address bit 4 (loaded from the sequencer as 0 at list start)
3	sequencer RAM address bit 3 (loaded from the sequencer as 0 at list start)
2	sequencer RAM address bit 2 (loaded from the sequencer as 0 at list start)
1	sequencer RAM address bit 1 (loaded from the sequencer as 0 at list start)
0	sequencer RAM address bit 0 (loaded from the sequencer as 0 at list start)

4.3.12 Last sequencer protocol register (0x2x1C Read)

This register can be used for debugging purposes after the occurrence of a sequencer command error. It is a read only register and stores the last sequencer key address. The register is updated with every command read from the VME to sequencer protocol FIFO.

Bit	Function
31	none, read back as 1
...	...
16	none, read back as 1
15	last sequencer key address bit A15
14	last sequencer key address bit A14
13	last sequencer key address bit A13
12	last sequencer key address bit A12
11	last sequencer key address bit A11
10	last sequencer key address bit A10
9	last sequencer key address bit A9
8	last sequencer key address bit A8
7	last sequencer key address bit A7
6	last sequencer key address bit A6
5	last sequencer key address bit A5
4	last sequencer key address bit A4
3	last sequencer key address bit A3
2	last sequencer key address bit A2
1	none, read back as 1
0	none, read back as 1

The reset value of the register is 0xFFFF0003

4.3.13 Sequencer Status Register (0x2x20 Read)

The sequencer status register is a read only register, which provides information about the current status of the sequencer, like sequencer enabled e.g..

Bit	Function
31	none, read back as 1
...	...
16	none, read back as 1
15	sequencer done (sequencer is in idle loop or disabled
14	sequencer busy (valid if sequencer s enabled)
13	sequencer idle (sequencer is enabled, but got no command)
12	none, read back as 0
11	sequencer executes a control command
10	sequencer executes an arbitration or primary address command
9	sequencer executes a FB data cycle
8	sequencer executes a FB DMA block transfer
7	error during FB DMA block transfer
6	error during FB data cycle
5	error during arbitration or primary address cycle
4	invalid sequencer command (undefined key address)
3	wait for external or internal event
2	sequencer is in RAM load mode
1	sequencer enabled and in RAM mode
0	sequencer enabled (in FIFO or RAM mode)

The reset value of the register is 0xFFFF0000

4.3.14 FASTBUS primary status register (0x2x24 Read)

The register is read only and be used for debugging purposes in case of a sequencer primary address cycle error .

Bit	Function
31	none, read back as 1
...	...
12	none, read back as 1
11	none, read back as 0
10	PA cycle (AK) timeout with active WT (SI far side timeout)
9	PA cycle (AK) timeout
8	none, read back as 0
7	last PA cycle stopped witn non zero SS response
6	bit SS2 of last PA cycle
5	bit SS1 of last PA cycle
4	bit SS0 of last PA cycle
3	AS timeout , WT detected on setting AS
2	pending master with other master still active
1	Arbitration timeout
0	AS/AK lock detected on setting AS

The reset value of the register is 0xFFFFF000

4.3.15 FASTBUS data status register (0x2x28 Read)

This register is the data cycle counter part to the FASTBUS primary status register for the data cycle phase of the FB transaction. It gives information on the error condition if the sequencer data or DMA cycle error bit in the sequencer status register is set . The FB data status register is read only.

Bit	Function
31	none, read back as 1
...	...
16	none, read back as 1
15	DMA done
14	DMA busy
13	VME timeout during last DMA
12	FB timeout during last DMA
11	DMA stopped by limit counter
10	bit SS2 of last DMA cycle
9	bit SS1 of last DMA cycle
8	bit SS0 of last DMA cycle
7	cycle stopped with non 0 SS-response
6	bit SS2 of last data cycle
5	bit SS1 of last data cycle
4	bit SS0 of last data cycle
3	DK timeout
2	DS timeout (WT asserted)
1	DK set
0	no AS/AK lock

The reset value of the register is 0xFFFF0000

4.3.16 Pedestal Pointer Register (0x2x000, read/write)

The NGFs on the fly pedestal subtraction and remapping mechanism is handled by a 64K x 32-bit memory and one of the on board FPGAs . The lower 16-bits of the pedestal memory are used to store the pedestal for the given address, the upper 16-bit can hold a 16-bit address. The pedestal memory is accessed through the pedestal pointer register and the pedestal and remap register, i.e. it is mapped into the NGFs VME slave address space by two addresses. The pedestal pointer (0x – 0xFFFF) defines the address within the memory, which is written to/read from through the pedestal and remap register.

Bit	Function
31	don't care
...	...
16	don't care
15	Pedestal pointer Bit 15
...	...
0	Pedestal pointer Bit 0

4.3.17 Pedestal and Remap Register (0x2x004, read/write)

Pedestal values and remap addresses are written to memory through this register. The actual memory location in the pedestal RAM is defined by the address the pedestal pointer register points to. The lower 16-bits of the register hold the pedestal value, the upper 16-bits are used to store a remap address, which is used if remapping is enabled.

Bit	Function
31	Remap address Bit 15
...	...
16	Remap address Bit 0
15	Pedestal Bit 15
...	...
0	Pedestal Bit 0

4.4 Minimum operation test

A minimum operation/NGF resource access test consists in switching on the four user LEDs by setting the respective bits (0-3) of the VME out register. No additional commands except a A32/D32 (or A24/D32) write to the address 0xA0B01000 (0xB01000 respective, with A being the selected rotary switch setting for address bits 31-28 and B being the setting of the rotary switch defining bits 23-20 of the NGF VME address space) with data word 0xF is required (the default address setting is A24 and A32 addressing enabled and 0xE00000 as 24-bit and 0x80E00000 as 32-bit address). This write has to result in successful completion (i.e. no VME bus error) and the four user LEDs have to light up.

If this test fails, the reason may be a default D16 data width in conjunction with A24 addressing of your CPU. This can be verified by monitoring the AS and DS0 VME lines with a VME extender and an oscilloscope. If your CPU tries to access the VME slave with two A24/D16, you will see two AS and DS0 signals on the scope, if one proper A24/D32 cycle is executed you will see one AS and DS0 only.

In the first case you will have to check your CPUs hard- and software documentation for the correct setup of the transfer, in the later case you will have to check with a VME diagnosis module (like the VDIS) or an extender, what address your CPU is actually trying to access.

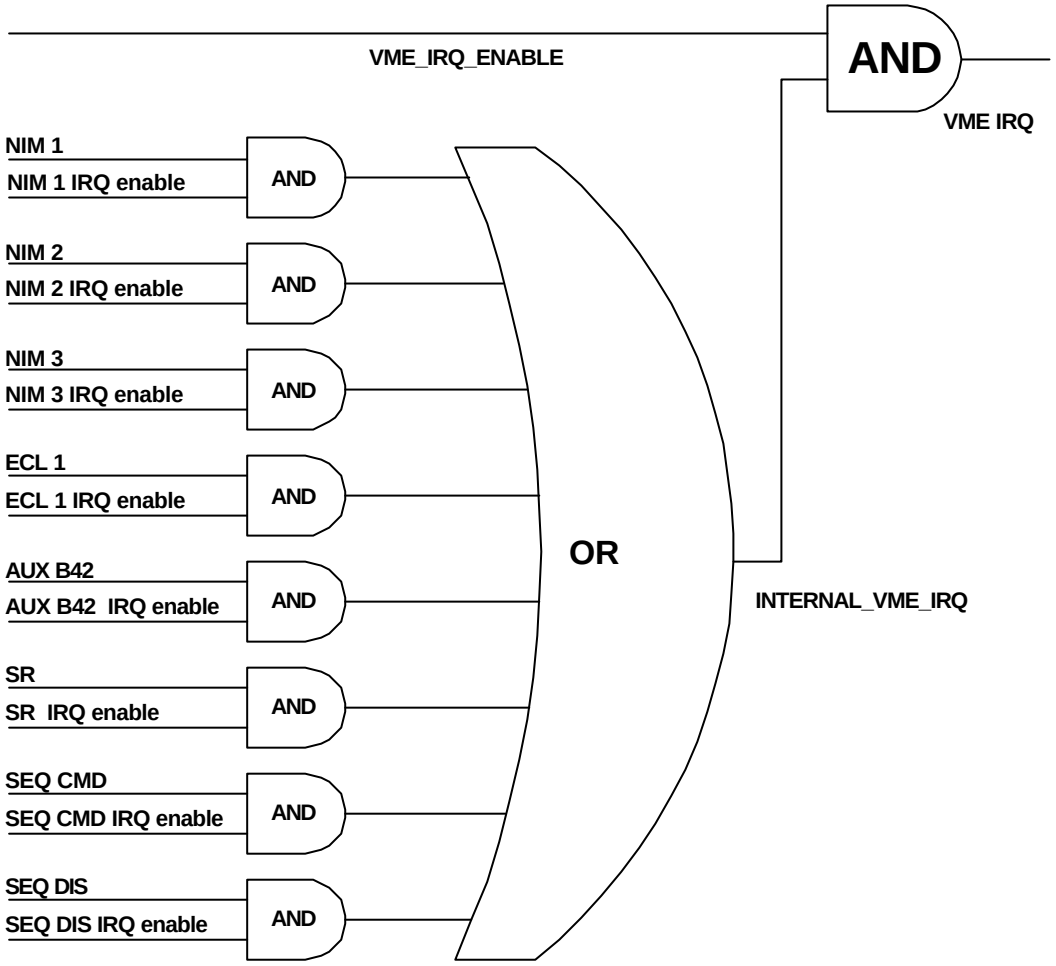
5 VME interrupt sources

Eight VME interrupt sources are implemented on the NGF, they can be activated individually by means of the interrupt source and mask register .. The VME bus interrupt level (0-7) and the interrupt vector is defined through the VME IRQ level and vector register..
Find below a table with the eight IRQ sources.

Source	IRQ condition/clearing of IRQ condition
NIM 1 input	The NIM 1 IRQ flag is set with the leading edge of the NIM 1 input. The flag is cleared by disabling the NIM 1 input source.
NIM 2 input	The NIM 2 IRQ flag is set with the leading edge of the NIM 2 input. The flag is cleared by disabling the NIM 2 input source.
NIM 3 input	The NIM 3 IRQ flag is set with the leading edge of the NIM 3 input. The flag is cleared by disabling the NIM 3 input source.
ECL 1 input	The ECL 1 IRQ flag is set with the leading edge of the ECL 1 input. The flag is cleared by disabling the ECL 1 input source.
AUX_B42 input	The AUX_B42 IRQ flag is set with the high to low transition (negative TTL edge) of the AUX_B42 input. The flag is cleared by disabling the AUX_B42 input source.
SR	The SR IRQ flag is set with a FASTBUS Service Request (SR). The flag is cleared by clearing the SR or by disabling the SR IRQ source.
SEQ_CMD_FLAG	The sequencer command IRQ flag is set if the “set sequencer command flag” sequencer control action is executed (see section 6.4). The flag is cleared with a write to the key address “clear sequencer command flag”.
SEQ_DISABLE	The sequencer disable IRQ flag is set when the sequencer is disabled. The flag is cleared by disabling the source.

An internal interrupt is generated if one or more of the interrupt sources are active, a VME interrupt is generated if the VME IRQ level is not zero and VME interrupt generation is enabled (level and generation are set through the VME IRQ level and vector register , see section 4.3.9 also).

The interrupt source can be seen in the VME IRQ source and mask register . The information is updated upon the occurrence of an internal interrupt (i.e. independent whether the VME interrupt generation is enabled or disabled).
Find below a sketch of the VME interrupt mechanism.



6 Sequencer

The sequencer is the NGFs key component to decouple VME and FASTBUS transactions. Data are passed to the sequencer via the VME2SEQ FIFO, the resulting data can be read either through the SEQ2VME FIFO (single cycle) or are pushed directly inte a VME slave memory (DMA block transfer). The VME target (slave) memory does not necessarily have to reside on the same board as the VME master.

6.1 Sequencer enable/arm

The sequencer is disabled and all on board FIFOs of the NGF are empty after a power reset. The sequencer is enabled with write access to the key address sequencer enable. As soon as the sequencer is enabled it scans the VME2SEQ FIFO continuously for new data. The sequencer logic reads the sequencer protocol and the data input FIFOs and tries to interpret the command as soon as the FIFO not empty condition is detected. The sequencer is automatically disabled by occurrence of an error condition, the FASBTUS lines will be cleared according to the definition of the FASTBUS protocol.
A table with the four error condition is given below.

Error flag	Error condition
SEQ_CMD_ERR	An invalid sequencer command was written to the VME2SEQ FIFO. The last sequencer protocol register can be used to read back the command for debugging purposes.
SEQ_PRIM_ERR	An error occurred during the FASTBUS primary address cycle . More detailed information can be retrieved from the FASTBUS primary status register .
SEQ_DATA_ERR	An error occurred during the FASTBUS data cycle . More detailed information can be retrieved from the FASTBUS data status register .
SEQ_DMA_ERR	An error occurred during a FASTBUS DMA cycle . More detailed information can be retrieved from the FASTBUS data status register

6.2 Sequencer command structure

The sequencer is loaded through the VME to sequencer FIFO, which resides on VME addresses 0x10000 through 0x1FFFF. The lower 16 bits of this address are interpreted as the sequencer command, the 32-bit datum of the A24/D32 VME cycle is interpreted as the parameter of the command (if the command involves a parameter at all). The sequencer command has the structure:

Address Bit	Name	Explanation
15	x	Function specific
14	x	
13	x	
12	x	
11	x	
10	x	
9	x	
8	x	
7	F3	Function code
6	F2	
5	F1	
4	F0	
3	SEQ_CTR	0 for FB transaction, 1 for sequencer control action
2	FB_EN	1 for FB transaction, 0 for sequencer control action
1	0	always 0
0	0	always 0

6.3 FASTBUS action sequencer command structure

Address Bit	Name	Explanation
15	x	unused
14	x	unused
13	x	unused
12	x	EG (enable geographic)
11	x	RD (read)
10	x	MS2
9	x	MS1
8	x	MS0
7	F3	F3
6	F2	F2
5	F1	F1
4	F0	F0
3	SEQ_CTR	0
2	FB_EN	1
1	0	0
0	0	0

The function codes for FB actions are listed in the table below:

F	F3	F2	F1	F0	Function	Datum
0	0	0	0	0	primary address cycle (including arbitration)	FB Primary Address
1	0	0	0	1	primary address cycle (including arbitration, holds mastership)	FB Primary Address
2	0	0	1	0	disconnect AS-AL lock	-
3	0	0	1	1	disconnect AS-AK lock and release mastership	-
4	0	1	0	0	transfer data cycle	FB write data
5	0	1	0	1	transfer data cycle with disconnect	FB write data
6	0	1	1	0	reserved	-
7	0	1	1	1	reserved	-
8	1	0	0	0	reserved	-
9	1	0	0	1	load block transfer read/write VME address pointer	VME slave address
A	1	0	1	0	load limit counter, clear word counter and start autonomous block transfer	Mode/Limit counter
B	1	0	1	1	load limit counter and start autonomous block transfer	Mode/Limit counter
C	1	1	0	0	reserved	-
D	1	1	0	1	store next VME address pointer in SEQ2VME FIFO	-
E	1	1	1	0	store DMA status and word counter in SEQ2VME FIFO	-
F	1	1	1	1	store word counter in SEQ2VME FIFO	-

6.3.1 VME mode and limit counter

As described earlier, the D32 VME data word during a write to the VME2SEQ FIFO defines the VME mode and the number of words for the limit counter for functions 0xA (load limit counter, clear word counter and start block transfer) and 0xB (load limit counter and start block transfer).
The function of the individual bits is described in the two tables below.

Bit	Function
31	remap upper 16 data bits (1)/don't remap(0)
30	subtract pedestal (1)/don't subtract (0)
29	store subtracted (1)/store raw value(0)
28	enable direct mode
27	enable VME mode
26	VME mode bit 2
25	VME mode bit 1
24	VME mode bit 0
23	limit counter bit 23
...	...
0	limit counter bit 0

A FASTBUS block transfer stops if limit counter+1 words have been transferred or if another stop condition (SS2 response e.g.) occurs during the transfer. The actual number of transferred words can be obtained from the SEQ2VME FIFO as DMA status and word counter (see below).
If both direct mode and VME mode are enabled, timing will be dominated by the slower of the two data paths (i.e. the FIFO almost full condition or wait from the AUX port).

The VME mode bits define the VME transfer type. No address increment takes place, if bit2 is set. This allows to write data to VME slaves with FIFO architecture. Be aware, that a FIFO full condition will not be detected by the NGF.

VME mode				
bit2	bit 1	bit 0	transfer type	AM
0	0	0	MBLT (64-bit block transfer) with address increment	0x8
0	0	1	D32 with address increment	0x9
0	1	0	BLT32 (32-bit block transfer) with address increment	0xB
0	1	1	reserved	
1	0	0	MLT64 (64-bit block transfer) without address increment	0x8
1	0	1	D32 without address increment	0x9
1	1	0	BLT32 (32-bit block transfer) without address increment	0xB
1	1	1	reserved	

6.3.2 Block transfer restrictions

6.3.2.1 MBLT

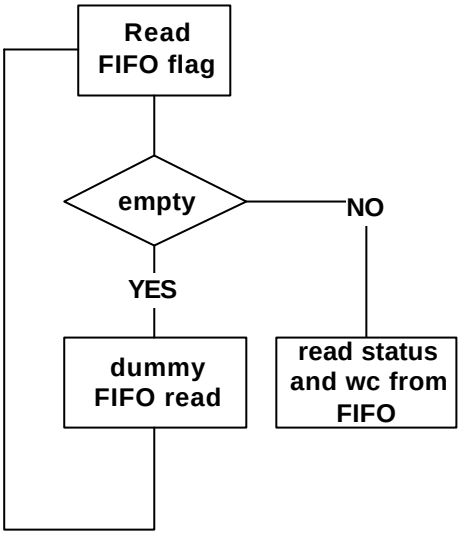
The VME slave address for a MBLT has to be aligned to an 8-byte boundary. According to the VME specification a new block transfer is started autonomously be the NGF every 2048-bytes to allow for access/arbitration by other masters.
The last data word of a FASTBUS block transfer with odd number of words is stored twice. Once at the 8-byte aligned address and a second time at the aligned address+4 bytes. The VME address pointer does not point to the last aligned address however and has to be initialised to an 8-byte aligned address before the next MBLT cycle is started.

6.3.2.2 BLT

The VME slave address for a BLT has to be aligned to an 4-byte boundary. According to the VME specification a new block transfer is started autonomously be the NGF every 256-bytes to allow for access/arbitration by other masters.

6.3.3 DMA status and word counter

The DMA status and word counter is stored in the SEQ2VME FIFO by the sequencer after a block transfer has terminated. The status can be read from the FIFO if the FIFO empty flag is cleared. For compatibility reasons the user has to update the FIFO flag by a dummy read to the FIFO, as indicated in the flow chart below.



The bit assignment of the data word is given in the table below.

Bit	Function
31	reserved, read as 0
30	reserved, read as 0
29	VME timeout during DMA
28	FASTBUS timeout during DMA
27	DMA stopped by word counter
26	SS2 response of DMA
25	SS1 response of DMA
24	SS0 response of DMA
23	word counter bit 23
...	...
0	word counter bit 0

6.4 Sequencer control action command structure

Address Bit	Explanation
15	unused
14	RAM Address 14
13	RAM Address 13
12	RAM Address 12
11	RAM Address 11
10	RAM Address 10
9	RAM Address 9
8	RAM Address 8
7	F3
6	F2
5	F1
4	F0
3	1
2	0
1	0
0	0

The function codes for sequencer control actions are listed in the table below:

F	F3	F2	F1	F0	Function	Datum
0	0	0	0	0	set sequencer out register	register data
1	0	0	0	1	disable sequencer	-
2	0	0	1	0	enable RAM sequencer mode, start RAM list at defined RAM address	-
3	0	0	1	1	disable RAM sequencer mode	-
4	0	1	0	0	wait for SEQ_GO_FLAG	
5	0	1	0	1	clear SEQ_GO_FLAG	
6	0	1	1	0	set sequencer SEQ_CMD_FLAG (generates VME IRQ if enabled)	-
7	0	1	1	1	reserved	-
8	1	0	0	0	reserved	
9	1	0	0	1	reserved	
A	1	0	1	0	reserved	
B	1	0	1	1	reserved	
C	1	1	0	0	reserved	
D	1	1	0	1	reserved	
E	1	1	1	0	reserved	
F	1	1	1	1	reserved	

Note: The SEQ_GO_FLAG is set with the leading edge of input ECL 1

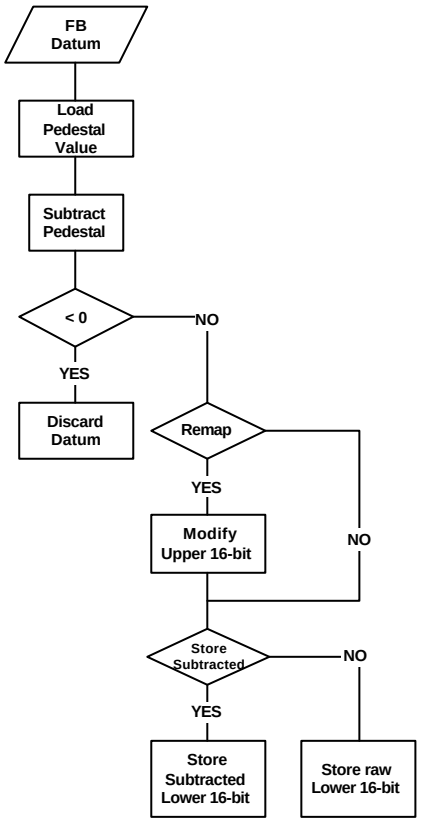
6.4.1 Sequencer out register

By means of the sequencer out register, a sequencer RAM list has the possibility to interfere with the front panel signals and user LEDs. With the exception of the AUX lines the sequencer out register has access to the same signals as the VME out register, the status of the two register setting is ored. The sequencer out register provides a good way to control an experiments deadtime logic, by setting a readout not busy FLIPFLOP e.g.. The sequencer out register is implemented as a J/K register.

Bit	Function
31	reserved
30	reserved
29	reserved
28	reserved
27	clear NIM output 4
26	clear NIM output 3
25	clear NIM output 2
24	clear NIM output 1
23	clear ECL output 4
22	clear ECL output 3
21	clear ECL output 2
20	clear ECL output 1
19	clear TTL output 4 and user LED 4
18	clear TTL output 3 and user LED 3
17	clear TTL output 2 and user LED 2
16	clear TTL output 1 and user LED 1
15	reserved
14	reserved
13	reserved
12	reserved
11	set NIM output 4
10	set NIM output 3
9	set NIM output 2
8	set NIM output 1
7	set ECL output 4
6	set ECL output 3
5	set ECL output 2
4	set ECL output 1
3	set TTL output 4 and user LED 4
2	set TTL output 3 and user LED 3
1	set TTL output 2 and user LED 2
0	set TTL output 1 and user LED 1

7 Pedestal Subtraction Unit (PSU)

The on the fly pedestal subtraction and remapping mechanism of the NGF is implemented by two 16-bit wide and 64 K deep memory sections and a XILINX Spartan FPGA from the hardware point of view. Pedestal subtraction and remapping have to be enabled, as the default power up state of the NGF is non sparsifying and non remapping. The speed of the pedestal subtraction matches the typical block transfer readout cycle performance of FASTBUS, i.e. the duration of the data processing for one channel is less than 50 ns (referring to 40 MB/s data rate). Due to the pipelined architecture this overhead will not be seen by the user. The upper 16-bit of the incoming read data are used as pedestal memory address. Two sparsifying modes can be used. In the first mode the pedestal stored in the pedestal memory location of a given channel is subtracted from the 16 bit data portion of the data word, channels with negative result are suppressed, the modified contents is passed onto the data stream if the result is non negative. In the second mode the data portion is compared to the pedestal memory location contents, the datum is suppressed if the data value is smaller than the pedestal or passed onto the data stream unchanged, if the data value exceeds the threshold. For non suppressed data words remapping of the data can take place in parallel.
A flow chart of the pedestal subtraction mechanism can be found below.



7.1 Configuration of pedestal subtraction

The pedestal subtraction mechanism is configured by the VME mode and limit counter (see section 6.3.1) portion during setup of a block transfer (i.e. the user can decide for the individual block transfer whether to read sparsified or not) and the data written to the pedestal memory. The pedestal memory contents is undefined after power up, it is mapped into the NGFs VME slave space via the pedestal pointer register (see 4.3.16) and the pedestal and remap register (see 4.3.17). Depending on the data format of the FASTBUS slave the user may have to store the pedestal for a given channel in several pedestal memory locations. This is illustrated by the following example.

Note: The sequencer has to be disabled during pedestal memory configuration

7.2 Example LRS 1885F ADC

The LRS 1885F ADC has the following data format.

Bit	Function
31	geographical address bit 4
30	geographical address bit 3
29	geographical address bit 2
28	geographical address bit 1
27	geographical address bit 0
26	event bit 2
25	event bit 1
24	event bit 0
23	range
22	channel bit 6
21	channel bit 5
20	channel bit 4
19	channel bit 3
18	channel bit 2
17	channel bit 1
16	channel bit 0
15	unused
14	unused
13	unused
12	
11	data bit 11
...	
0	data bit 0

The pedestal memory address is computed from the upper 16-bits of the data word. The pedestal for a given channel has to be stored in 16 different locations if auto ranging is enabled (the high and low range may well have a different pedestal in addition) and the 8 ring buffers are used.

Example: The pedestals of channel 5 of an 1885F in slot 3 of a FASTBUS crate will have to be stored under the following addresses.

low range:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0x1805
0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	1	0x1905
0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	1	0x1A05
0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	1	0x1B05
0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	1	0x1C05
0	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	0x1D05
0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	1	0x1D05
0	0	0	1	1	1	1	1	0	0	0	0	0	1	0	1	0x1E05

high range:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Address
0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	1	0x1885
0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	1	0x1985
0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	1	0x1A85
0	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	0x1B85
0	0	0	1	1	1	0	0	1	0	0	0	0	1	0	1	0x1C85
0	0	0	1	1	1	0	1	1	0	0	0	0	1	0	1	0x1D85
0	0	0	1	1	1	1	0	1	0	0	0	0	1	0	1	0x1D85
0	0	0	1	1	1	1	1	1	0	0	0	0	1	0	1	0x1E85

7.3 Event length determination

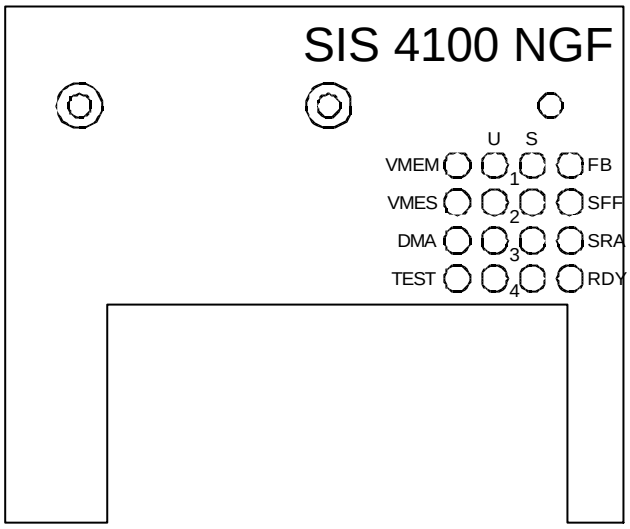
The event length in sparsified mode can be determined by subtracting the VME start address from the Next VME address pointer register .

event length [in Bytes] = Next VME address pointer – VME start address - 4

8 Front panel elements

8.1 LEDs

16 front panel LEDs are implemented on the top section of the NGF. Due to space limitations the front panel designation may not be clear enough in some cases.
Find below a drawing of the LED front panel portion, a description of the function is given in the table underneath.



Designation	Color	Description
VMEM	yellow	NGF is VME master
VMES	red	NGF is VME slave
DMA	yellow	NGF to VME DMA in progress
TEST	red	test mode active
FB	yellow	FASTBUS transaction in progress
SFF	green	sequencer FIFO mode active
SRA	yellow	sequencer RAM mode active
RDY	green	logic configured
S1	red	SHARC 1 (to be set and cleared under DSP program control)
S2	red	SHARC 2 (to be set and cleared under DSP program control)
S3	red	SHARC 3 (to be set and cleared under DSP program control)
S4	red	SHARC 4 (to be set and cleared under DSP program control)
U1	yellow	User 1 (to be set and cleared under user program control)
U2	yellow	User 2 (to be set and cleared under user program control)
U3	yellow	User 3 (to be set and cleared under user program control)
U4	yellow	User 4 (to be set and cleared under user program control)

8.1.1 LED self test

A power up LED self test is implemented. All LEDs except the green RDY LED are on during the logic configuration phase, as soon as the configuration is completed, all LEDs all LEDs are switched off and the green RDY LED is switched on.

8.2 Connectors

13 LEMO sockets and one 20 pin header connector are available on the NGF front panel. The main purpose of the connectors to facilitate the integration of a standard VME CPU into a physics data acquisition setup by providing several input and output connections. Find below the lower NGF front panel section with the inputs and outputs.

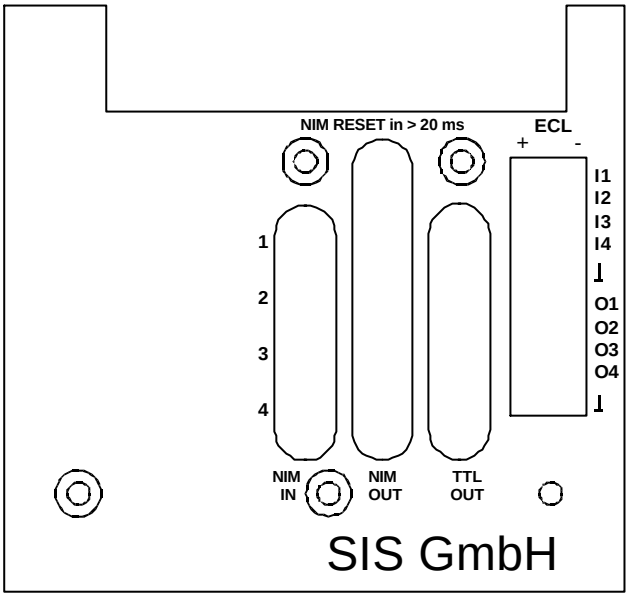


Table of LEMO connectors:

Designation	Level	Description
NIM In 1	NIM	NIM Input 1
NIM In 2	NIM	NIM Input 2
NIM In 3	NIM	NIM Input 3
NIM In 4	NIM	NIM Input 4
NIM Out 1	NIM	NIM Output 1
NIM Out 2	NIM	NIM Output 2
NIM Out 3	NIM	NIM Output 3
NIM Out 4	NIM	NIM Output 4
RESET	NIM	Reset (the reset signal has to be active for > 20 ms)
TTL Out 1	TTL	TTL Output 1 (to be set and cleared under user program control)
TTL Out 2	TTL	TTL Output 2 (to be set and cleared under user program control)
TTL Out 3	TTL	TTL Output 3 (to be set and cleared under user program control)
TTL Out 4	TTL	TTL Output 4 (to be set and cleared under user program control)

Pin Assignment on 20 Pin Header:

Pin	Description	Description	Pin
1	Input 1+	Input 1-	2
3	Input 2+	Input 2-	4
5	Input 3+	Input 3-	6
7	Input 4+	Input 4-	8
9	Ground	Ground	10
11	Output 1+	Output 1-	12
13	Output 2+	Output 2-	14
15	Output 3+	Output 3-	16
17	Output 4+	Output 4-	18
19	Ground	Ground	20

9 AUX interface

The AUX interface to the NGFs first auxiliary connector has two main tasks. The first is to establish a connection between row A and C of the first VME slots P2 connector, the second is to allow the user to connect a trigger interface to the NGF. In principle it is also possible to connect one or several NGFs to an event building system via an ECL bus, which gets FASTBUS data in direct mode.
The P2 connection is of interest for two purposes again. The first is the connection of transition cards to the VME CPU to fan out SCSI, Ethernet and terminal ports, the second is the connection to the VSB bus to establish VDB crate interconnects in existing DAQ setups based upon this concept.
The pin assignment of the AUX interface connector can be found in section 14.2.

9.1 Trigger interface

The user can interfere with the NGF via the trigger interface in several ways. Besides the possibility of VME interrupt generation input and output lines for data path control are implemented. In addition you can write to and read from the trigger interface, what allows for the implementation of multi crate setup event number distribution e.g.
Find below a table with the trigger interface AUX signals.

AUX pin	Signal/Function
A10	TTL output (power up reset condition low) to be cleared and set through the VME signal out register
A28	TTL output, function as A10
A38	TTL output, LCA ready High level indicates, that the NGF is in reset condition, low level flags, that the NGFs on board logic is configured
A45	TTL output, function as A10 In addition setting A45 enables the data path to write data to the trigger interface (see 9.2.1.2) **Note: the sequencer has to be disabled before A45 is set**
B40	TTL output pulse (low active) Generated by VME access to the key address generate AUX_B40 pulse
B42	TTL input The level of this input can be read through the sequencer FIFO flag and ECL/NIM input register . An open input is seen as high, hence the user should foresee the use as low active signal. A TTL high to low transition sets the interrupt flag if AUX_B42 is enabled as interrupt source.
B43	TTL input (pullup resistor) Enable (low active) data path from the auxiliary connector to the internal FB_IO_AD bus **Note: the sequencer has to be disabled**
B44	TTL input (pullup resistor) put AUX2FB data buffer in latch mode (high) put AUX2FB data buffer in transparent mode (low)
B45	TTL input (pullup resistor) Enable (low active) internal FB_IO_AD bus data path to the auxiliary connector
B46	TTL input (pullup resistor) put FB2AUX data buffer in latch mode (high) put FB2AUX data buffer in transparent mode (low)

9.2 Direct mode interface

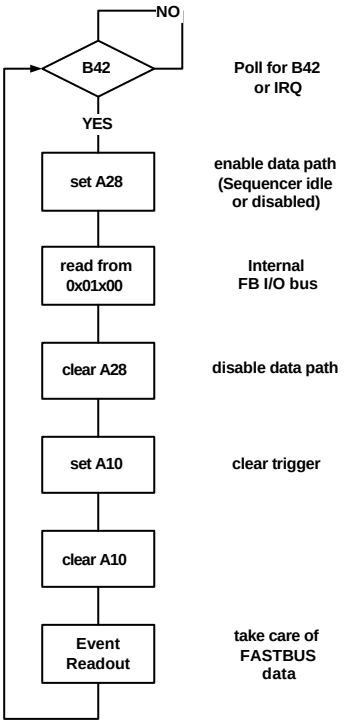
In addition to the signals described in the previous section (9.1) three more bits are implemented to control data flow to a user auxiliary card in direct mode.

AUX pin	Signal/Function
A40	TTL output (power up reset condition low) FB2AUX write enable This signal is active low during a block transfer if direct mode is enabled.
A41	TTL output FB2AUX write clock The data on the BUF_AD bus become valid with the low to high transition of this clock (setup and hold time 15 ns)
B37	TTL input (pullup resistor) FB2AUX DMA wait A FASTBUS block transfer can be delayed by asserting a low level **Note: the user logic has to take into account, that one more word may be written to the AUX port after asserting wait**

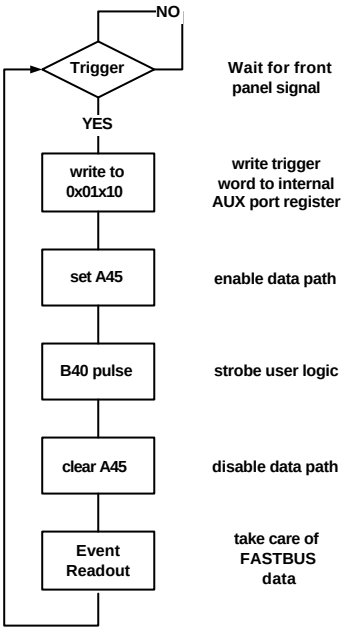
9.2.1 Trigger interface examples

The two flow charts below are intended to demonstrate the possibilities of the AUX trigger interface.

9.2.1.1 Read word from trigger interface



9.2.1.2 Write word to trigger interface



10 Jumper and rotary switch locations and functions

10.1 Input Termination

The termination of the individual 4 NIM and ECL inputs can be removed to facilitate cabling in multi crate setups. The jumpers JP870-JP873 and JP810-JP813 are located right behind the lemo input/output PCB.

10.1.1 NIM Inputs

The input termination of the 4 NIM inputs through the 47 Ω resistor network RN870 can be removed by opening jumpers 870 through 873. The NIM reset input is terminated through jumper JP880. The factory default is inputs terminated (jumpers installed). Find below a list of the jumper/input assignment.

Jumper	Input
JP870	NIM 1
JP871	NIM 2
JP872	NIM 3
JP873	NIM 4
JP880	NIM Reset

10.1.2 ECL Inputs

The input termination of the 4 ECL inputs can be removed by opening jumpers 810 through 813. The factory default is inputs terminated (jumpers installed). Find below a list of the jumper/input assignment.

Jumper	Input
JP810	ECL 1
JP811	ECL 2
JP812	ECL 3
JP813	ECL 4

10.2 A24/A32 Slave address width and VME slave base address

Jumpers EN_A32 and EN_A24 define the VME slave address width (A32 or A24) of the NGF. The jumpers are located next to the two rotary switches for the base address selection.

EN_A32	EN_A24	Width	Base Adress
closed	closed	A32 A24	0xA0B00000 0xB00000
closed	open	A32	0xA0B00000
open	closed	A24	0xB00000
open	open	-	undefined

Note: A denotes the setting of the A32 and B the setting of the A24 rotary switch

The default address setting is A24 and A32 addressing enabled and 0xE00000 as 24-bit and 0x80E00000 as 32-bit address).

10.3 JP_SH1 and JP_SH2

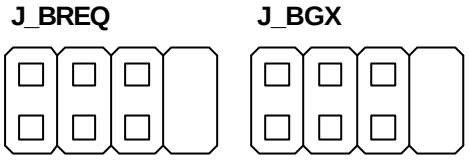
Jumpers JP_SH1 and JP_SH2 can be used to integrate the XILINX EPLDs of the SIS9200 SHARC DSP Piggy pack into the JTAG chain of NGFs the on board XILINX EPLDs. . This functionality is not designed for standard end user access.

10.4 Shark1 and Shark2

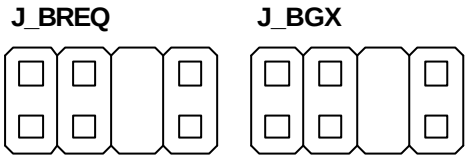
These two jumpers are located on the power connector card, they are used to enable/disable the 3.3 V TTL bus switches for the SHARC links of SHARC 1 and SHARC 2 respective.

10.5 J_BREQ and J_BGX

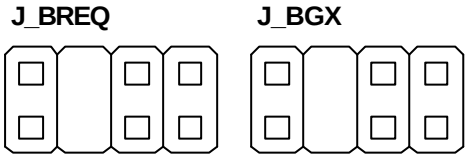
The VME master bus request and bus grant level of the NGF is defined by the settings of the jumper arrays J_BREQ and J_BGX. The jumper position set in the J_BREQ jumper field has to be set in the J_BGX field also. Find below a drawing with the four possible jumper settings.
The factory default setting is level 3.



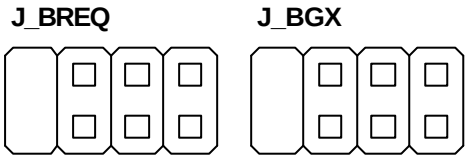
Requester and Bus Grant Level 3 (Factory default setting)



Requester and Bus Grant Level 2



Requester and Bus Grant Level 1



Requester and Bus Grant Level 0

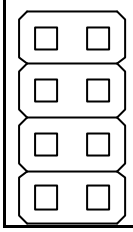
10.6 VME arbiter type JP219

The NGF can be set to release when done (RWD) or release on request (ROR) arbiter mode via JP219. The factory default is RWD.

JP219	VME arbiter type
closed	release when done (RWD, factory default)
open	release on request (ROR)

10.7 Boot File Selection and Sysreset behaviour J960

One jumper of jumper array 960 defines the VME sysreset behaviour of the NGF, the others are to select the boot mode and the boot address.
This jumper array is covered by the power connector card, the drawing of the connector



Jumper	Function
Sysreset	closed sysreset by VME, open generates Sysreset with NGF reset
boot source	closed FLASHPROM, open JTAG (the JTAG mode is reserved to maintenance)
A17	boot file bit 0 (closed = 0/open =1)
A18	boot file bit 1 (closed = 0/open =1)

The default setting is all jumpers closed (boot from FLASHPROM file 0. , NGF reset with VME sysreset)

11 JTAG

Two JTAG chains are implemented on the NGF. One to program the XILINX EPLDs and one to program the XILINX FPGAs. The two connectors are designated JTAGEPLD and JTAGFPGA. They are not meant for end user access.

12 SHARC support

Up to two SHARC DSPs can be used for data processing on the NGF. They are connected via two SHARC link ports to the PSU and the AUX backplane of the second FASTBUS slot of the NGF.
In the first firmware revision of the SIS4100 SHARC support is not implemented yet.

13 Power consumption and limits

It has to be taken into account, that the power consumed by the NGF and the modules in its VME card cage have to be provided by the FASTBUS backplane in a save manner. The NGF draws +5 V from two FASTBUS slots via the maincard and the power connector and SHARC link card, hence 36 A are available for the VME slaves and the interface itself (according to the FB standard +5 V power is limited to 18 A per slot).
VME ±12 V is generated by linear voltage regulators out of the ±15 V FB supplies.

NGF power consumption:

Voltage	Current
+5 V	4 A
-5.2 V	1,5 A
-2 V	-
+15 V	depends on VME +12 V consumption
- 15V	depends on VME -12 V consumption

Limit for total VME module consumption:

Voltage	max. Current
+5 V	33 A
+12 V	1 A
-12 V	1 A

14 Appendix

14.1 Connector Types

In case you should experience problems with worn or broken connectors, we list the used types in this section of the manual. In most cases it may be most straightforward to contact us for a replacement, but maybe your workshop got hold of the required part also.

14.1.1 VME Backplane

The SIS4100 is equipped with a custom design mechanical auto daisy chain VME backplane. The backplane is available with two or three VME slots. The user does not have to take care of bus grant and interrupt acknowledge line jumpering due to the auto daisy chain functionality. Rows A and C of the P2 are routed to the FASTBUS auxiliary connector. The VME backplane connects to the SIS4100 board via two high density connectors, i.e. the unit can be retrofitted for other VME slot counts in conjunction with a mechanical upgrade kit. Find below a list of the used connector types.

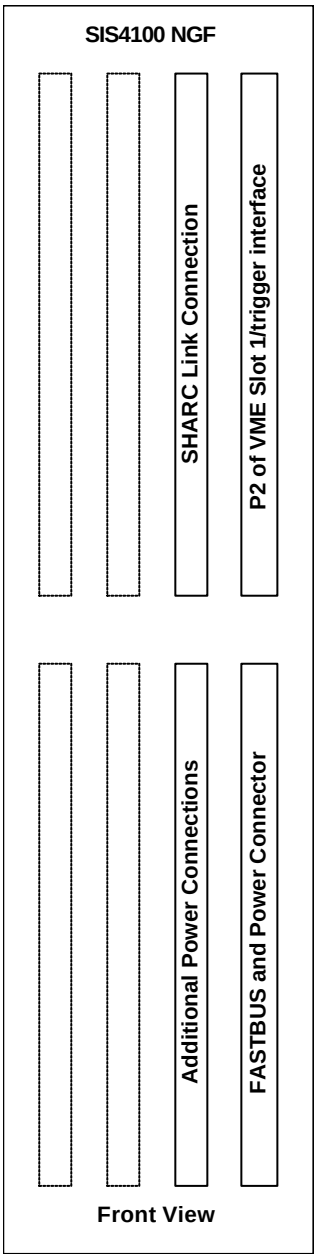
Connector	Manufacturer	Part Number
P1	EPT	104-60464-3
P2	EPT	104-60064-03
backplane to SIS4100	SAMTEC	TFM-150-01-S-D-RA
SIS4100 to backplane	SAMTEC	SFM-150-01-S-D
power terminal	EPT	911-32046

14.1.2 FASTBUS

Standard two row FASTBUS connectors are used as backplane connectors, they are manufactured by AMP, Dupont and BERG, find below the manufacturer and his part number for the first batch of NGF modules.

Connector	Manufacturer	Part Number
FASTBUS	BERG	66527-565
Auxiliary Connector	BERG	66527-565

Find below a drawing with the position of the NGFs four FASTBUS connectors (as seen from the front of the module).



14.1.3 Front panel

Two connector types are used on the front panel, standard LEMO connectorsfor the NIM signals and pin headers for ECL and TTL signals. The part numbers are listed in the table below:

Connector	Manufacturer	Part Number
LEMO	LEMO	EPA.00.250.NTN
Flat cable (20-pin)	3M	2520-5002

14.2 Auxiliary connector pin assignment

AUX SHARC link connector				AUX VME P2/trigger connector			
A1	GND	GND	B1	A1	GND	GND	B1
A2	SH1 EXT BUS 12	SH2 EXT BUS 12	B2	A2	P2A 32	P2C 32	B2
A3	SH1 EXT BUS 13	SH2 EXT BUS 13	B3	A3	P2A 31	P2C 31	B3
A4	not connected	not connected	B4	A4	P2A 30	P2C 30	B4
A5	SH1 EXT BUS 14	SH2 EXT BUS 14	B5	A5	P2A 29	P2C 29	B5
A6	SH1 EXT BUS 15	SH2 EXT BUS 15	B6	A6	P2A 28	P2C 28	B6
A7	SH1 EXT BUS 16	SH2 EXT BUS 16	B7	A7	P2A 27	P2C 27	B7
A8	SH1 EXT BUS 17	SH2 EXT BUS 17	B8	A8	P2A 26	P2C 26	B8
A9	not connected	not connected	B9	A9	P2A 25	P2C 25	B9
A10	not connected	VCC	B10	A10	IO160	VCC	B10
A11	SH1 EXT BUS 18	SH2 EXT BUS 18	B11	A11	P2A 24	P2C 24	B11
A12	SH1 EXT BUS 19	SH2 EXT BUS 19	B12	A12	P2A 23	P2C 23	B12
A13	not connected	not connected	B13	A13	P2A 22	P2C 22	B13
A14	SH1 EXT BUS 20	SH2 EXT BUS 20	B14	A14	P2A 21	P2C 21	B14
A15	SH1 EXT BUS 21	SH2 EXT BUS 21	B15	A15	P2A 20	P2C 20	B15
A16	SH1 EXT BUS 22	SH2 EXT BUS 22	B16	A16	P2A 19	P2C 19	B16
A17	SH1 EXT BUS 23	SH2 EXT BUS 23	B17	A17	P2A 18	P2C 18	B17
A18	not connected	not connected	B18	A18	P2A 17	P2C 17	B18
A19	GND	-5.2 V	B19	A19	GND	-5.2 V	B19
A20	not connected	not connected	B20	A20	P2A 16	P2C 16	B20
A21	not connected	not connected	B21	A21	P2A 15	P2C 15	B21
A22	not connected	not connected	B22	A22	P2A 14	P2C 14	B22
A23	not connected	not connected	B23	A23	P2A 13	P2C 13	B23
A24	not connected	not connected	B24	A24	P2A 12	P2C 12	B24
A25	not connected	not connected	B25	A25	P2A 11	P2C 11	B25
A26	not connected	not connected	B26	A26	P2A 10	P2C 10	B26
A27	not connected	not connected	B27	A27	P2A 9	P2C 9	B27
A28	not connected	GND	B28	A28	IO161	GND	B28
A29	not connected	not connected	B29	A29	P2A 8	P2C 8	B29
A30	not connected	not connected	B30	A30	P2A 7	P2C 7	B30
A31	not connected	not connected	B31	A31	P2A 6	P2C 6	B31
A32	not connected	not connected	B32	A32	P2A 5	P2C 5	B32
A33	not connected	not connected	B33	A33	P2A 4	P2C 4	B33
A34	not connected	not connected	B34	A34	P2A 3	P2C 3	B34
A35	not connected	not connected	B35	A35	P2A 2	P2C 2	B35
A36	not connected	not connected	B36	A36	P2A 1	P2C 1	B36
A37	GND	not connected	B37	A37	GND	2AXDMAWT	B37
A38	not connected	not connected	B38	A38	LCA_RDY_L	2FBDATVALID	B38
A39	not connected	VCC	B39	A39	SFT_RESET_L	VCC	B39
A40	not connected	not connected	B40	A40	2AXDMAWENL	AXDMARES	B40
A41	not connected	-5.2 V	B41	A41	2AXDMAWCLR	-5.2 V	B41
A42	not connected	not connected	B42	A42	2FBDMARENL	AX LCA 1	B42
A43	not connected	not connected	B43	A43	2FBDMARCLR	2FBOEL	B43
A44	not connected	not connected	B44	A44	2FBDMAOEL	2FBIEL	B44
A45	not connected	not connected	B45	A45	AX LCA 2	2AXOEL	B45
A46	GND	not connected	B46	A46	GND	2AXIEL	B46
A47	not connected	not connected	B47	A47	AD0	AD8	B47
A48	not connected	not connected	B48	A48	AD1	AD9	B48
A49	not connected	not connected	B49	A49	AD2	AD10	B49
A50	not connected	not connected	B50	A50	AD3	AD11	B50
A51	not connected	not connected	B51	A51	AD4	AD12	B51
A52	not connected	not connected	B52	A52	AD5	AD13	B52
A53	not connected	not connected	B53	A53	AD6	AD14	B53
A54	not connected	not connected	B54	A54	AD7	AD15	B54
A55	-2 V	-5.2 V	B55	A55	-2 V	-5.2 V	B55
A56	not connected	not connected	B56	A56	AD16	AD24	B56
A57	not connected	not connected	B57	A57	AD17	AD25	B57
A58	not connected	not connected	B58	A58	AD18	AD26	B58
A59	not connected	not connected	B59	A59	AD19	AD27	B59
A60	not connected	VCC	B60	A60	VCC	VCC	B60
A61	not connected	not connected	B61	A61	AD20	AD28	B61
A62	not connected	not connected	B62	A62	AD21	AD29	B62
A63	not connected	not connected	B63	A63	AD22	AD30	B63
A64	not connected	not connected	B64	A64	AD23	AD31	B64
A65	GND	GND	B65	A65	GND	GND	B65

14.3 Nomenclature

Find below a description of expressions, which are used throughout this manual.

14.3.1 J/K register

Several NGF registers are implemented as so called J/K registers. The functions of a J/K register are switched on by writing a 1 into the on position and switched off by writing a 1 into the off position. A 0 written has no effect, an undefined toggle state results from setting the on and off bit at the same time

14.3.2 Key Address

A write cycle with arbitrary data to the given key address executes the specified function.

14.4 Software Support

The NGFs are tested with a VME PC with Tundra Universe II PCI bridge under Windows NT and a graphical user interface, which is build with National Instruments CVI. While this overall setup and the full software is only of limited usability for the use of the SIS4100 in a particle physics setup, the subroutines are perfectly suited as examples and to build the base for dedicated user readout code. The source code of our test environment is included on a floppy disk with the first NGF shipped to a group/detector/institute.

15 Index