# SIS3320
# 200 MHz 12-bit
# VME Digitizer

## 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: 1.03 as of 23.06.06 sis3320-M-0-001v103

# Revision Table:

| Revision | Date | Modification |
|---|---|---|
| 0.01 | 02.05.05 | Generation |
| 0.12 | 05.10.05 | Prerelease |
| 1.00 | 18.10.05 | First official release |
| 1.01 | 21.02.06 | Firmware 0x104 (summation feature)<br>Bug fixes |
| 1.02 | 13.04.06 | Explanation of DAC load mechanism |
| 1.03 | 23.06.06 | Firmware 0x106 (random clock, external clock x 5, key reset effects ADC FPGAs also) |

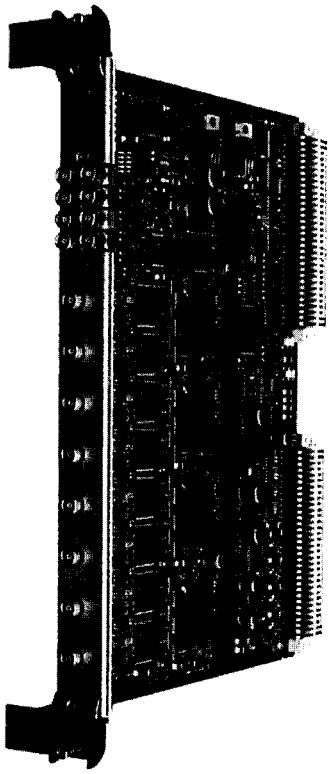
# - Table of contents

# 1 Introduction

The SIS3320 is an eight channel ADC/digitizer board with a sampling rate of up to 210 MHz (for the individual channel) and a resolution of 12-bit. The board is a single width 6U VME card, which has no special (i.e. non standard VME) voltage requirements.
The flexible combination of DDR2 memory technology data storage in combination with FPGA based data handling/movement allows for a generic design which covers a variety of applications.

Applications comprise but are not limited to:

- digitization of "slow" detectors like calorimeters
- accelerator/machine controls



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.

## *1.1 Related documents*

A list of available firmware designs can be retrieved from
http://www.struck.de/sis3320firm.htm

# 2 Technical Properties/Features

## 2.1 Key functionality

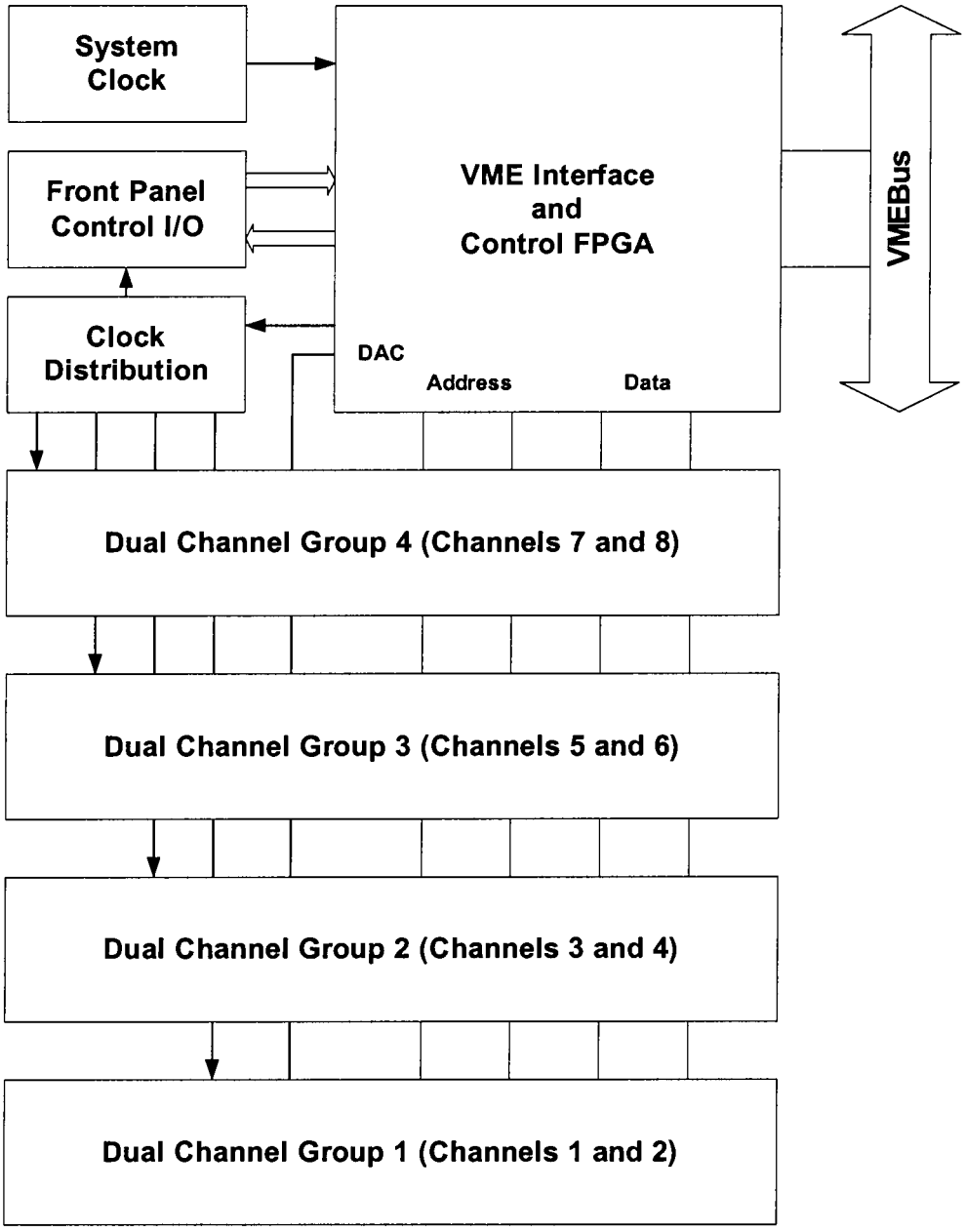
Find below a list of key features of the SIS3320 digitizer.

- 8 channels
- 12-bit resolution
- 32 MSamples/channel memory
- special clock modes (clock prescaling, external "arbitrary" clock)
- 2 range settings
- offset DACs
- external/internal clock
- external random clock
- multi event mode
- read on the fly (actual sample value)
- pre/post trigger option
- readout in parallel to acquisition
- trigger generation (FIR trigger)
- 4 NIM control inputs/4 NIM control outputs
- A32 D32/BLT32/MBLT64/2eVME
- Hot swap (in conjunction with VME64x backplane)
- VME64x Connectors
- VME64x Front panel
- VME64x extractor handles (on request)
- F1002 compatible P2 row A/C assignment
- +5 V, +12V and –12 V VME standard voltages

**Note:** The SIS3320 shall not be operated on P2 row A/C extensions, like VSB e.g. due to the compatibility to the F1001 FADC modules clock and start/stop distribution scheme.

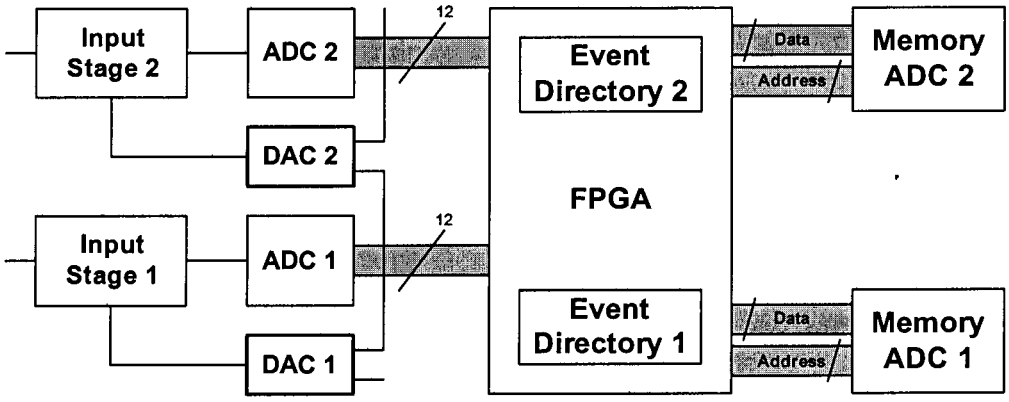
## 2.2 Module design

The SIS3320 consists of four identical groups of 2 ADC channels each and a control section as shown in the simplified block diagram below.



### 2.2.1 Dual channel group

Two ADC channels form a group, which memory is handled by one Field Programmable Gate Array (FPGA).




## *2.3 Modes of Operation*

The SIS3320 was developed with maximum flexibility in mind. The FPGA based design of the card allows to meet the requirements of many readout applications with dedicated firmware designs in the future. The initial firmware is supposed to furnish you with an easy to use yet powerful high speed high resolution Flash Analog to Digital Converter (FADC) implementation, that covers many everyday analog to digital applications.

## *2.4 Memory management*

The memory can be used either as one contiguous memory or as a subdivided multi event memory. In addition memory depth can be limited in single event operation to match the requirements of the given application.

### 2.4.1 Single Event Mode

The full memory of 32 MSamples of the SIS3320 is used as one big circular buffer or as single shot memory in single event mode, unless memory size is limited by the event configuration register.

### 2.4.2 Multi Event Mode

The memory can be divided in pages or events to make the acquisition of shorter signals more efficient. Up to 512 stop pointers for the individual page can be retrieved from the event directory. In auto start mode the ADC advances to the next page and starts sampling automatically.

## 2.5 Clock sources

The SIS3320 features the two basic clock modes

- Internal clock
- External symmetric clock

### 2.5.1 Internal clock

The internal clock is generated from an on board 50 MHz quartz. It is either doubled or multiplied by 4 to generate 100 MHz and 200 MHz respectively.

| Internal clock speeds |
|---|
| 200 MHz |
| 100 MHz |
| 50 MHz |

### 2.5.2 External clock

A symmetric external clock (NIM level, ratio between 45:55 and 55:45) can be fed to the module through a LEMO00 connector. An ECL clock over rows A/C of the J2 VME backplane can be used as an alternative. For optimum performance the clock frequency should be within the specified range for the given ADC chip.

| Module | Min. sym. clock | Max sym. clock |
|---|---|---|
| SIS3320-210 | 40 MHz | 210 MHz |

## 2.6 Trigger control (pre/post, start/stop and gate mode)

The SIS3320 features pre/post trigger capability as well as start/stop mode acquisition and a gate mode (in which start and stop are derived from the leading and trailing edge of a single control input signal).
The trigger behaviour is defined by the acquisition control register.

## 2.7 Internal Trigger generation

The trigger output of the SIS3320 can be either used to interact with external trigger logic or to base start/stop on a threshold (i.e. one individual threshold per ADC channel) of the digitized data. Trigger generation can be activated with two conditions:

- module armed (i.e. sample clock active, trigger can be used to start acquisition)
- module armed and started (trigger can be used to stop acquisition)

The user can select between triggering on the conditions above and below threshold

### 2.8 VME Interrupts

Two registers, the Interrupt configuration and the Interrupt control register, are implemented for interrupt setup and control.
Two interrupt sources are implemented:

- End of event
- End of last event in multievent mode

# 3 VME Addressing

As the SIS3320 VME FADC features memory options with up to 8 times 32 MSamples, A32 addressing was implemented as the only option. The module occupies an address space of 0x7FFFFFF Bytes, i.e. 128 MBytes are used by the module.
The base address is defined by the selected addressing mode, which is selected by jumper array JP80 and SW1 and SW2 (in non geographical mode).

The table below summarises the possible base address settings.

| J1 Setting | | | Bits | | | | | |
|---|---|---|---|---|---|---|---|---|
| A32 | GEO | VIPA | 31 | 30 | 29 | 28 | 27 | |
| x | | | SW1 | | | | | SW2=0...7<br>Bit 27=0 |
| x | | | SW1 | | | | | SW2=8...F<br>Bit 27=1 |
| x | x | | Not implemented in this design | | | | | |
| | | x | Not implemented in this design | | | | | |

| Shorthand | Explanation |
|---|---|
| SW1/SW2 | Setting of rotary switch SW1 or SW2 respective |

**Notes:**

- This concept allows the use of the SIS3320 in standard VME as well as in VME64x environments, i.e. the user does not need to use a VME64x backplane.
- The factory default setting is EN_A32 closed, SW1=3, SW2=0 (i.e. the module will react to A32 addressing under address 0x30000000). With more than one unit shipped in one batch a set of addresses (like 0x10000000, 0x20000000, 0x30000000,...) may be used also.

### 3.1 Address Map

The SIS3320 resources and their locations are listed in the table below.
**Note:** Write access to a key address (KA)with arbitrary data invokes the respective action

| Offset | Size in Bytes | BLT | Access | Function |
|---|---|---|---|---|
| 0x00000000 | 4 | - | W/R | Control/Status Register (J-K register) |
| 0x00000004 | 4 | - | R | Module Id. and Firmware Revision register |
| 0x00000008 | 4 | - | R/W | Interrupt configuration register |
| 0x0000000C | 4 | - | R/W | Interrupt control register |
| | | | | |
| 0x00000010 | 4 | - | R/W | Acquisition control/status register (J-K register) |
| 0x00000014 | 4 | - | R/W | Extern Start Delay register |
| 0x00000018 | 4 | - | R/W | Extern Stop Delay register |
| | | | | |
| 0x00000020 | | | R/W | Max. Nof Event Register |
| 0x00000024 | | | R | Actual Event Counter |
| | | | | |
| | | | | |
| 0x00000030 | 4 | - | R/W | CBLT/Broadcast Setup register |
| 0x00000034 | 4 | - | R/W | ADC Memory Page register |
| | | | | |
| 0x00000050 | 4 | - | R/W | DAC Control Status register |
| 0x00000054 | 4 | - | R/W | DAC Data register |
| 0x00000058 | 4 | - | R/W | ADC Gain register |
| | | | | |
| 0x00000060 | | | R/W | XILINX JTAG_TEST/JTAG_DATA_IN |
| 0x00000064 | | | W | XILINX JTAG_CONTROL |
| | | | | |
| | | | | |
| 0x00000400 | 4 | - | KA W | General Reset |
| | | | | |
| 0x00000410 | 4 | - | KA W | Arm Sampling Logic |
| 0x00000414 | 4 | - | KA W | Disarm Sampling Logic |
| 0x00000418 | 4 | - | KA W | VME Start sampling |
| 0x0000041C | 4 | - | KA W | VME Stop sampling |
| | | | | |
| 0x00000428 | 4 | - | KA W | Reset DDR2 Memory Logic |

| Event information all ADC groups | | | | |
|---|---|---|---|---|
| 0x01000000 | 4 | - | W only | Event configuration register (all ADCs) |
| 0x01000004 | 4 | | W only | Sample Length register (all ADCs) |
| 0x01000008 | 4 | - | W only | Sample Start address register (all ADCs) |
| 0x0100000C | 4 | - | W only | ADC input mode register (all ADCs) |
| | | | | |

| Event information ADC group 1 | | | | |
|---|---|---|---|---|
| 0x02000000 | 4 | - | R/W | Event configuration register (ADC1, ADC2) |
| 0x02000004 | 4 | | R/W | Sample Length register (ADC1, ADC2) |
| 0x02000008 | 4 | - | R/W | Sample Start address register (ADC1, ADC2) |
| 0x0200000C | 4 | - | R/W | ADC input mode register (ADC1, ADC2) |
| | | | | |
| 0x02000010 | 4 | - | R | Next Sample address register ADC1 |
| 0x02000014 | 4 | - | R | Next Sample address register ADC2 |
| | | | | |
| 0x02000020 | 4 | - | R | Actual Sample Value (ADC1, ADC2) |
| 0x02000024 | 4 | - | R | internal Test register |
| 0x02000028 | 4 | - | R | DDR2 Memory Logic Veryfication register (ADC1, ADC2) |
| | | | | |
| 0x0200002C | 4 | | R/W | Trigger Flag Clear Counter register (ADC1, ADC2) |
| | | | | |
| 0x02000030 | 4 | | R/W | ADC1 Trigger setup register |
| 0x02000034 | 4 | | R/W | ADC1 Trigger Threshold register |
| 0x02000038 | 4 | | R/W | ADC2 Trigger setup register |
| 0x0200003C | 4 | | R/W | ADC2 Trigger Threshold register |
| | | | | |
| 0x02010000 | 0x800 | X | R | Event directory ADC1 |
| 0x02018000 | 0x800 | X | R | Event directory ADC2 |

| Event information ADC group 2 | | | | |
|---|---|---|---|---|
| 0x02800000 | 4 | - | R/W | Event configuration register (ADC3, ADC4) |
| And so on (as for ADC group 1) | | | | |

| Event information ADC group 3 | | | | |
|---|---|---|---|---|
| 0x03000000 | 4 | - | R/W | Event configuration register (ADC5, ADC6) |
| And so on (as for ADC group 1) | | | | |

| Event information ADC group 4 | | | | |
|---|---|---|---|---|
| 0x03800000 | 4 | - | R/W | Event configuration register (ADC7, ADC8) |
| And so on (as for ADC group 1) | | | | |

| ADC memory pages | | | | |
|---|---|---|---|---|
| 0x04000000 | 8 MByte | X | R | ADC 1 memory page |
| 0x04800000 | 8 MByte | X | R | ADC 2 memory page |
| 0x05000000 | 8 MByte | X | R | ADC 3 memory page |
| 0x05800000 | 8 MByte | X | R | ADC 4 memory page |
| 0x06000000 | 8 MByte | X | R | ADC 5 memory page |
| 0x06800000 | 8 MByte | X | R | ADC 6 memory page |
| 0x07000000 | 8 MByte | X | R | ADC 7 memory page |
| 0x07800000 | 8 MByte | X | R | ADC 8 memory page |

**Note 2:** MBLT64 read access is supported from memory (i.e. not from register space) only.

# 4 Register Description

The function of the individual registers is described in detail in this section.
The first line after the subsection header (in Courier font) like:

```
#define SIS3320_CONTROL_STATUS      0x0     /* read/write; D32 */
```

refers to the sis3320.h header file.

## 4.1 Control/Status Register(0x, write/read)

```
#define SIS3320_CONTROL_STATUS      0x0     /* read/write; D32 */
```

The control register is implemented as 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. The only function at this point in time is user LED on/off.
On read access the same register represents the status register.

| Bit | write Function | read Function |
|---|---|---|
| 31 | Clear reserved 15 (*) | 0 |
| 30 | Clear reserved 14 (*) | 0 |
| 29 | Clear reserved 13 (*) | 0 |
| 28 | Clear reserved 12 (*) | 0 |
| 27 | Clear reserved 11 (*) | 0 |
| 26 | Clear reserved 10 (*) | 0 |
| 25 | Clear reserved 9 (*) | 0 |
| 24 | Clear reserved 8 (*) | 0 |
| 23 | Clear reserved 7 (*) | 0 |
| 22 | Clear reserved 6 (*) | 0 |
| 21 | Clear reserved 5 (*) | 0 |
| 20 | Clear reserved 4 (*) | 0 |
| 19 | Clear reserved 3 (*) | 0 |
| 18 | Clear reserved 2 (*) | 0 |
| 17 | Clear reserved 1 (*) | 0 |
| 16 | Switch off user LED (*) | 0 |
| 15 | Set reserved 15 | Status reserved 15 |
| 14 | Set reserved 14 | Status reserved 14 |
| 13 | Set reserved 13 | Status reserved 13 |
| 12 | Set reserved 12 | Status reserved 12 |
| 11 | Set reserved 11 | Status reserved 11 |
| 10 | Set reserved 10 | Status reserved 10 |
| 9 | Set reserved 9 | Status reserved 9 |
| 8 | Set reserved 8 | Status reserved 8 |
| 7 | Set reserved 7 | Status reserved 7 |
| 6 | Set reserved 7 | Status reserved 6 |
| 5 | Set reserved 7 | Status reserved 4 |
| 4 | Set reserved 7 | Status reserved 4 |
| 3 | Set reserved 3 | Status reserved 3 |
| 2 | Set reserved 2 | Status reserved 2 |
| 1 | Set reserved 1 | Status reserved 1 |
| 0 | Switch on user LED | Status User LED (1=LED on, 0=LED off) |

(*) denotes power up default setting

### *4.2 Module Id. and Firmware Revision Register (0x4, read)*

```
#define SIS3320_MODID              0x4       /* read only; D32 */
```

This register reflects the module identification of the SIS3320 and its minor and major firmware revision levels. The major revision level will be used to distinguish between substantial design differences and experiment specific designs, while the minor revision level will be used to mark user specific adaptations.

| Bit | Function | Reading |
|---|---|---|
| 31 | Module Id. Bit 15 | 3 |
| 30 | Module Id. Bit 14 | |
| 29 | Module Id. Bit 13 | |
| 28 | Module Id. Bit 12 | |
| 27 | Module Id. Bit 11 | 3 |
| 26 | Module Id. Bit 10 | |
| 25 | Module Id. Bit 9 | |
| 24 | Module Id. Bit 8 | |
| 23 | Module Id. Bit 7 | 2 |
| 22 | Module Id. Bit 6 | |
| 21 | Module Id. Bit 5 | |
| 20 | Module Id. Bit 4 | |
| 19 | Module Id. Bit 3 | 0 |
| 18 | Module Id. Bit 2 | |
| 17 | Module Id. Bit 1 | |
| 16 | Module Id. Bit 0 | |
| 15 | Major Revision Bit 7 | |
| 14 | Major Revision Bit 6 | |
| 13 | Major Revision Bit 5 | |
| 12 | Major Revision Bit 4 | |
| 11 | Major Revision Bit 3 | |
| 10 | Major Revision Bit 2 | |
| 9 | Major Revision Bit 1 | |
| 8 | Major Revision Bit 0 | |
| 7 | Minor Revision Bit 7 | |
| 6 | Minor Revision Bit 6 | |
| 5 | Minor Revision Bit 5 | |
| 4 | Minor Revision Bit 4 | |
| 3 | Minor Revision Bit 3 | |
| 2 | Minor Revision Bit 2 | |
| 1 | Minor Revision Bit 1 | |
| 0 | Minor Revision Bit 0 | |

#### 4.2.1 Major revision numbers

Find below a table with major revision numbers used to date

| Major revision number | Application/user |
|---|---|
| 0x01 | Generic designs |

### *4.3 Interrupt configuration register (0x8)*

```
#define SIS3320_IRQ_CONFIG      0x8      /* read/write; D32 */
```

This read/write register controls the VME interrupt behaviour of the SIS3320 ADC. Four interrupt sources are foreseen, for the time being three of them are associated with an interrupt condition, the fourth condition is reserved for future use.
The interrupter type is DO8 .

#### 4.3.1 IRQ mode

In RORA (release on register access) mode the interrupt will be pending until the IRQ source is cleared by specific access to the corresponding disable VME IRQ source bit. After the interrupt is serviced the source has to be activated with the enable VME IRQ source bit again.
In ROAK (release on acknowledge) mode , the interrupt condition will be cleared (and the IRQ source disabled) as soon as the interrupt is acknowledged by the CPU. After the interrupt is serviced the source has to be activated with the enable VME IRQ source bit again.

| Bit | Function | Default |
|---|---|---|
| 31 | | 0 |
| ... | | 0 |
| 16 | | 0 |
| 15 | | 0 |
| 14 | | 0 |
| 13 | | 0 |
| 12 | RORA/ROAK Mode (0: RORA; 1: ROAK) | 0 |
| 11 | VME IRQ Enable (0=IRQ disabled, 1=IRQ enabled) | 0 |
| 10 | VME IRQ Level Bit 2 | 0 |
| 9 | VME IRQ Level Bit 1 | 0 |
| 8 | VME IRQ Level Bit 0 (0 always) | 0 |
| 7 | IRQ Vector Bit 7; placed on D7 during VME IRQ ACK cycle | 0 |
| 6 | IRQ Vector Bit 6; placed on D6 during VME IRQ ACK cycle | 0 |
| 5 | IRQ Vector Bit 5; placed on D5 during VME IRQ ACK cycle | 0 |
| 4 | IRQ Vector Bit 4; placed on D4 during VME IRQ ACK cycle | 0 |
| 3 | IRQ Vector Bit 3; placed on D3 during VME IRQ ACK cycle | 0 |
| 2 | IRQ Vector Bit 2; placed on D2 during VME IRQ ACK cycle | 0 |
| 1 | IRQ Vector Bit 1; placed on D1 during VME IRQ ACK cycle | 0 |
| 0 | IRQ Vector Bit 0; placed on D0 during VME IRQ ACK cycle | 0 |

The power up default value reads 0x 00000000

**Note: IRQ levels 2, 4 and 6 are supported only**

### 4.4 Interrupt control register (0xC)

```
#define SIS3320_IRQ_CONTROL          0xC       /* read/write; D32 */
```

This register controls the VME interrupt behaviour of the SIS3320 ADC. Four interrupt sources are foreseen, for the time being three of them are associated with an interrupt condition, the fourth condition is reserved for future use.

| Bit | Function (w) | (r) | Default |
|---|---|---|---|
| 31 | unused | Status IRQ source 3 (reserved) | 0 |
| 30 | unused | Status IRQ source 2 (reserved) | 0 |
| 29 | unused | Status IRQ source 1 (end of last event, disarm) | 0 |
| 28 | unused | Status IRQ source 0 (end of event) | 0 |
| 27 | unused | Status VME IRQ | 0 |
| 26 | unused | Status internal IRQ | 0 |
| 25 | unused | 0 | 0 |
| 24 | unused | 0 | 0 |
| 23 | Clear IRQ source 3 | Status flag source 3 | 0 |
| 22 | Clear IRQ source 2 | Status flag source 2 | 0 |
| 21 | Clear IRQ source 1 | Status flag source 1 | 0 |
| 20 | Clear IRQ source 0 | Status flag source 0 | 0 |
| 19 | Disable IRQ source 3 | 0 | 0 |
| 18 | Disable IRQ source 2 | 0 | 0 |
| 17 | Disable IRQ source 1 | 0 | 0 |
| 16 | Disable IRQ source 0 | 0 | 0 |
| 15 | unused | 0 | 0 |
| 14 | unused | 0 | 0 |
| 13 | unused | 0 | 0 |
| 12 | unused | 0 | 0 |
| 11 | unused | 0 | 0 |
| ... | ... | ... | 0 |
| 4 | unused | 0 | 0 |
| 3 | Enable IRQ source 3 | Status enable source 3 (read as 1 if enabled, 0 if disabled) | 0 |
| 2 | Enable IRQ source 2 | Status enable source 2 (read as 1 if enabled, 0 if disabled) | 0 |
| 1 | Enable IRQ source 1 | Status enable source 1 (read as 1 if enabled, 0 if disabled) | 0 |
| 0 | Enable IRQ source 0 | Status enable source 0 (read as 1 if enabled, 0 if disabled) | 0 |

The power up default value reads 0x 00000000

IRQ source 3: reserved
IRQ source 2: reserved
IRQ source 1: end of last event (disarm)
IRQ source 0: end of event

## 4.5 *Acquisition control register (0x10, read/write)*

```
#define SIS3320_ACQUISTION_CONTROL    0x10      /* read/write; D32 */
```

The acquisition control register is in charge of most of the settings related to the actual configuration of the digitization process.
Like the control register it is implemented in a J/K fashion.

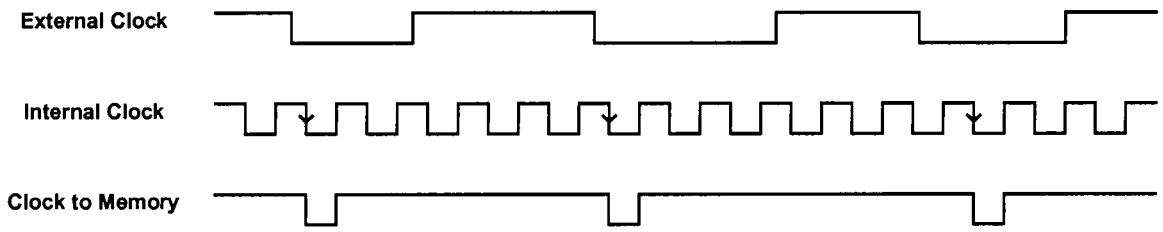
| Bit | Write Function | Read |
|---|---|---|
| 31 | Clear reserved 15 (*) | 0 |
| 30 | Clear Clock Source Bit2 | 0 |
| 29 | Clear Clock Source Bit1 | 0 |
| 28 | Clear Clock Source Bit0 | 0 |
| 27 | Clear reserved 11 (*) | 0 |
| 26 | Clear reserved 10 (*) | 0 |
| 25 | Clear reserved 9 (*) | 0 |
| 24 | Disable front panel LEMO start/stop logic | 0 |
| 23 | Clear reserved 7 (*) | 0 |
| 22 | Disable internal trigger as stop (*) | 0 |
| 21 | Disable Multi Event mode (*) | 0 |
| 20 | Disable Autostart mode (*) | 0 |
| 19 | Clear reserved 3 (*) | 0 |
| 18 | Clear reserved 2 (*) | 0 |
| 17 | Clear reserved 1 (*) | ADC Sampling Busy |
| 16 | Clear reserved 0 (*) | ADC Sampling Logic Armed |
| 15 | Set reserved 15 | Status reserved 15 |
| 14 | Set clock source Bit 2 | Status clock source Bit 2 |
| 13 | Set clock source Bit 1 | Status clock source Bit 1 |
| 12 | Set clock source Bit 0 | Status clock source Bit 0 |
| 11 | Set reserved 11 | Status reserved 11 |
| 10 | Set reserved 10 | Status reserved 10 |
| 9 | Set reserved 9 | Status reserved 9 |
| 8 | Enable front panel Lemo Start/Stop logic | Status front panel start/stop logic |
| 7 | Set reserved 7 | Status reserved 7 |
| 6 | Enable internal trigger as stop | Status internal trigger as stop |
| 5 | Enable Multi Event mode | Status Multi Event mode |
| 4 | Enable Autostart mode | Status Autostart mode |
| 3 | Set reserved 3 | Status reserved 3 |
| 2 | Set reserved 2 | Status reserved 2 |
| 1 | Set reserved 1 | Status reserved 1 |
| 0 | Set reserved 0 | Status reserved 0 |

The power up default value reads 0x

Clock source bit setting table:

| Clock Source Bit2 | Clock Source Bit1 | Clock Source Bit0 | Clock Source |
|---|---|---|---|
| 0 | 0 | 0 | internal 200 MHz |
| 0 | 0 | 1 | internal 100 MHz |
| 0 | 1 | 0 | internal 50 MHz |
| 0 | 1 | 1 | external clock x 5 |
| 1 | 0 | 0 | external clock with internal doubling (the external clock has to be in the range 24 MHz-100 MHz) |
| 1 | 0 | 1 | Random Clock |
| 1 | 1 | 0 | external clock (LEMO front panel) ; min. 40 MHz |
| 1 | 1 | 1 | P2-Clock (not yet implemented) |

**External random clock mode** :
This mode allows for sampling at arbitrary low and non symmetric external clock. The digitizer is set up for an internal clock of 200 MHz and will strobe one datum to memory with the leading edge of the internal clock cycle that follows the leading edge of an external clock pulse as illustrated below. Pipelining between the actual analog input signal and the value stored to memory has to be taken into account.




Multi Event mode :

0 : Sampling Logic Armed state will be cleared at end of event

1 : Sampling Logic Armed state will be cleared at end of last event (defined with Max_NOF_Event register)

Autostart mode:

0 : The Sampling will start when the Sampling Logic is armed and a START issue (External LEMO or VME KEY command).

1 : The Sampling will start with the Sampling Logic Arm command (VME KEY command) and if Multi Event Mode is enabled with end of Event to start sampling in the next page

### 4.6 Start Delay register (0x14, read/write)

```
#define SIS3320_START_DELAY      0x14      /* read/write; D32 */
```

Pretrigger operation can be implemented via the start delay register in conjunction with front panel start/stop mode operation. The external Start Signal and the Autostart Signal will be delayed by the value of the register .

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 24 | unused, read as 0 |
| 23 | START_DELAY_BIT23 |
| .. | |
| .. | |
| 0 | START_DELAY_BIT0 |

The power up default value is 0

### 4.7 Stop Delay register (0x18, read/write)

```
#define SIS3320_STOP_DELAY       0x18      /* read/write; D32 */
```

Posttrigger operation can be implemented via the stop delay register in conjunction with front panel start/stop mode operation. The external stop signal and the internal trigger (if enabled) will be delayed by the value of the register .

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 24 | unused, read as 0 |
| 23 | STOP_DELAY_BIT23 |
| .. | |
| .. | |
| 0 | STOP_DELAY_BIT0 |

The power up default value is 0

### 4.8 Max_Nof_Events_Register (0x20, read/write)

```
#define SIS3320_MAX_NOF_EVENT                0x20        /* read/write; D32 */
```

The Sampling Logic will be disarmed In Multi Event mode as soon as the Event counter reaches the value of the Max_Nof_Events register.

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 20 | unused, read as 0 |
| 19 | MAX NOF Events Bit19 |
| .. | |
| .. | |
| 0 | MAX NOF Events Bit 0 |

The power up default value is 0

### 4.9 Actual_Event_Counter (0x24, read)

```
#define SIS3320_ACTUAL_EVENT_COUNTER         0x24        /* read; D32 */
```

This register holds the actual number of events in multi event mode.
The Event Counter is cleared when the Sampling Logic is armed and it is incremented with every start sampling.

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 20 | unused, read as 0 |
| 19 | Actual event counter Bit 19 |
| .. | |
| .. | |
| 0 | Actual event counter Bit 0 |

The power up default value is 0

### 4.10 CBLT/Broadcast setup register

```
#define SIS3320_CBLT_BROADCAST_SETUP      0x30      /* read/write; D32 */
```

The CBLT feature is not implemented yet (as of firmware 01 02).

This read/write register defines, whether the SIS3320 will participate in a Broadcast. The configuration of this register and the registers of other participating modules is essential for proper Broadcast behaviour.

| Bit | Function |
|---|---|
| 31 | CBLT/Broadcast address bit 31 |
| 30 | CBLT/Broadcast address bit 30 |
| 29 | CBLT/Broadcast address bit 29 |
| 28 | CBLT/Broadcast address bit 28 |
| 27 | CBLT/Broadcast address bit 27 |
| 26 | CBLT/Broadcast address bit 26 |
| 25 | CBLT/Broadcast address bit 25 |
| 24 | CBLT/Broadcast address bit 24 |
| 23 | reserved |
| 22 | reserved |
| 21 | reserved |
| 20 | reserved |
| 19 | reserved |
| 18 | reserved |
| 17 | reserved |
| 16 | reserved |
| 15 | reserved |
| 14 | reserved |
| 13 | reserved |
| 12 | reserved |
| 11 | reserved |
| 10 | 0 |
| 9 | 0 |
| 8 | 0 |
| 7 | 0 |
| 6 | 0 |
| 5 | Enable Broadcast Master |
| 4 | Enable Broadcast |
| 3 | 0 |
| 2 | reserved |
| 1 | reserved |
| 0 | reserved |

### *4.11 ADC Memory Page register*

```
#define SIS3320_ADC_MEMORY_PAGE_REGISTER 0x34      /* read/write; D32 */
```

The SIS3320 default memory size per channel is 64 MByte (i.e. 32 MSample).

The VME address space window per ADC is limited to 8 MByte (4 MSample) however. The read/write ADC memory page register is used to select one of the 8 memory subdivisions (pages).

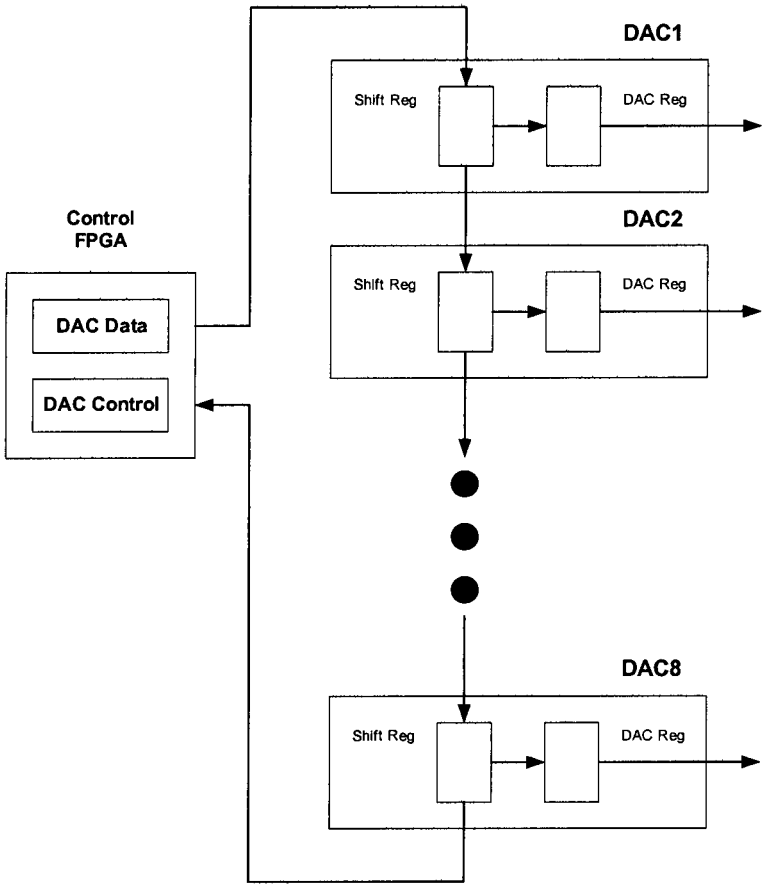
| Bit | Function |
|---|---|
| 31 | reserved |
| .. | |
| .. | |
| 4 | reserved |
| 3 | Page register bit 3 (reserved) |
| 2 | Page register bit 2 |
| 1 | Page register bit 1 |
| 0 | Page register bit 0 |

Example: readout routine for 32MSample readout
(see sis3320_single_event_sample_wrap_test1.c)

```
int sis3320_Read_ADC_Channel(    unsigned int module_address,
                                 unsigned int vme_read_mode,
                                 unsigned int adc_channel /* 0 to 7 */,
                                 unsigned int event_sample_start_addr,
                                 unsigned int event_sample_length,
                                 unsigned int* uint_adc_buffer)
```

### *4.12 DAC Control Registers*

This set of registers is used to program the 16-bit offset DACs for the 8 ADC channels. Refer to the documentation of the AD5570 DAC chip for details also and have a look to the configuration example in sis3320_adc_test1.c (CVI directory)
All 8 DACs are in a daisy chain .




Example routine:

```
int sis3320_write_dac_offset( unsigned int module_addr,
                              unsigned int *offset_value_array)
```

**Note:** You will have to keep in mind, that the DAC settings of all 8 channels will change temporarily in the process of changing a setting for a particular channel.

### 4.12.1 DAC Control/Status register (read/write)

```
#define SIS3320_DAC_CONTROL_STATUS 0x50        /* read/write; D32 */
```

| Bit | Write Function | Read Function |
|---|---|---|
| 31 | None | 0 |
| .. | .. | .. |
| .. | .. | .. |
| 16 | None | 0 |
| 15 | None | DAC Read/Write/Clear Cycle BUSY |
| 14 | None | 0 |
| .. | .. | .. |
| 4 | .. | .. |
| 3 | .. | .. |
| 2 | none | 0 |
| 1 | DAC Command Bit 1 | DAC Command Bit 1 Status |
| 0 | DAC Command Bit 0 | DAC Command Bit 0 Status |

DAC Command Bit

| Bit 1 | Bit 0 | Function |
|---|---|---|
| 0 | 0 | No function |
| 0 | 1 | Load shift register |
| 1 | 0 | Load DACs |
| 1 | 1 | Clear DACs |

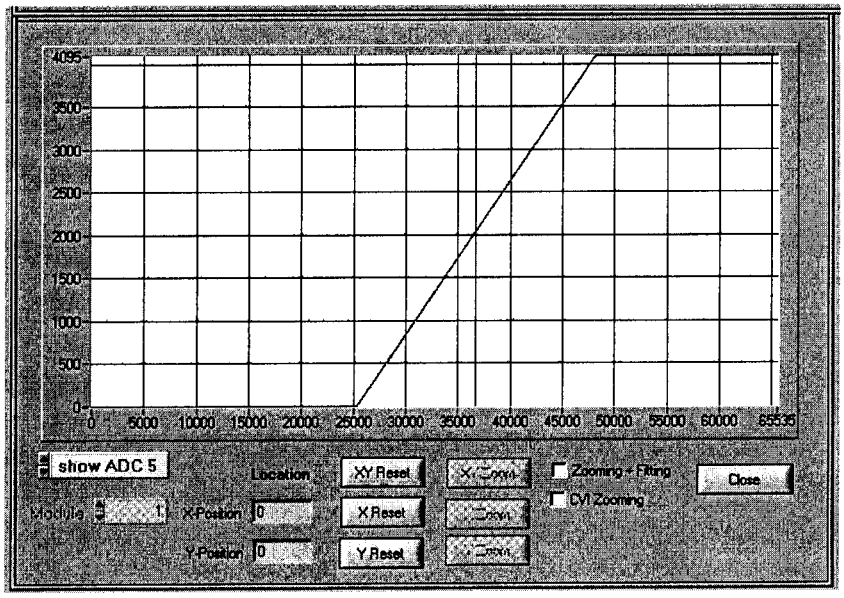
A "Clear DAC" command sets the value of the DAC to 32768 (0x8000).

### 4.12.2 DAC Data register (read/write)

```
#define SIS3320_DAC_DATA            0x54        /* read/write; D32 */
```

| Bit | Write Function | Read Function |
|---|---|---|
| 31 | none | DAC Input Register Bit 15 (from DAC) |
| .. | .. | .. |
| .. | .. | .. |
| 16 | none | DAC Input Register Bit 0 |
| 15 | DAC Output Register Bit 15 | DAC Output Register Bit 15 |
| .. | .. | 0 |
| .. | .. | 0 |
| 0 | DAC Output Register Bit 0 | DAC Output Register Bit 0 |

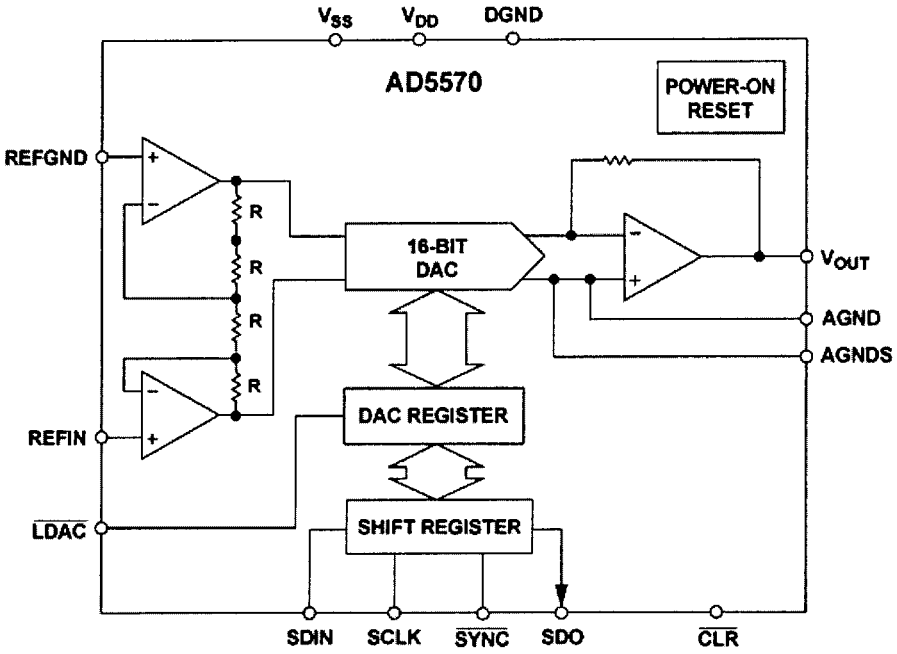
Find below a DAC scan that was acquired with the DAC test function of the SIS3320 ADC Labwindows application. The SIS3320 under test was configured for an input span of some 2 $V_{pkpk}$. It can be seen, that a DAC offset of some 37000 counts is required to accomplish an input range of -1...+1V on this particular channel.




**Note:** The actual sample value registers can be used to monitor the influence of the DAC settings.

### 4.12.3 DAC load procedure

A block diagram of the AD5570 DAC is shown below.



The actual sample registers (offset 0x02000020 for ADC1/2 e.g.) furnish a good way to check the offset while the DACs are programmed/ramped. As an initial step you may want to execute a clear DAC what brings all 8 DACs to midrange (0x8000).

The 8 DACs are in a chain. Data (from register 0x54) are shifted through with a write with datum=1 to the DAC control register (0x50) and loaded with a write with datum=2 to the DAC control register.
As the DAC (and serial DAC chain) is fairly slow you will have to make sure, that the DAC busy flag is back to 0 before executing the next command.

To set all channels to the same offset you will write the value 0x9500 e.g. to the DAC data register and execute 8 times the sequence shift/load (with waiting for not busy).

The full DAC chain of the SIS3320 is loaded as shown below (assuming, that the DAC value for channel 1 is stored in dacdatum[0])

```
for(i=0;i<8;i++)
    dacdata=dacdatum[8-i]
    daccontrol=1 (shift)
        read dacstatus
    until busy==0
    daccontrol=2 (load)
        read dacstatus
    until busy==0
```

**Note 1:** It is not obvious from the block diagram of the AD5570 DAC, but the load command is required after the shift command for the shift operation to take effect

**Note 2:** With a 9th shift/load you should see the datum coming back to the data registers upper 16-bit. But this step is not needed, its rather a test feature.

### *4.13 ADC Gain register (read/write)*

```
#define SIS3320_ADC_GAIN_CONTROL    0x58      /* read/write; D32 */
```

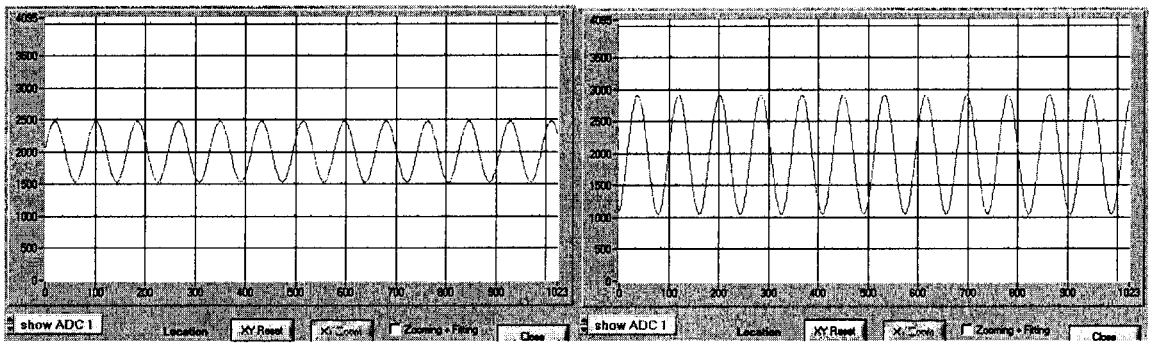
The ADC chip supports two input ranges. The input of the ADC has a range of 1,5 $V_{pkpk}$ in full scale and a range of 0,75 $V_{pkpk}$ in half scale mode. I.e. the scale bit allows you to change the input sensitivity by a factor of 2. The actual scale will depend on the configuration of the input stage of your unit.
The ADC scale register is a read/write register which lowest 8 bits are used to select the scale of the 8 ADC channels.

| Bit | Write Function | Read Function |
|---|---|---|
| 31 | none | 0 |
| .. | .. | .. |
| .. | .. | .. |
| 8 | none | 0 |
| 7 | Gain Bit ADC 8 | Gain Bit ADC 8 |
| 6 | Gain Bit ADC 7 | Gain Bit ADC 7 |
| 5 | Gain Bit ADC 6 | Gain Bit ADC 6 |
| 4 | Gain Bit ADC 5 | Gain Bit ADC 5 |
| 3 | Gain Bit ADC 4 | Gain Bit ADC 4 |
| 2 | Gain Bit ADC 3 | Gain Bit ADC 3 |
| 1 | Gain Bit ADC 2 | Gain Bit ADC 2 |
| 0 | Gain Bit ADC 1 | Gain Bit ADC 1 |

The power up default value is 0 (all channels at full scale)

| Scale bit setting | Scale |
|---|---|
| 0 | Full (default at power up) |
| 1 | Half |

Find below screen shots with gain 0 (left hand side) and gain 1 setting with a 400 $mV_{pkpk}$ sine on a unit with 2 $V_{pkpk}$ range.



### 4.14 Key address general reset

```
#define SIS3320_KEY_RESET                    0x400    /* write only; D32 */
```

A write with arbitrary data to this register (key address) resets the SIS3320 to it's power up state.

### 4.15 Key address VME arm sampling logic

```
#define SIS3320_KEY_ARM                      0x410    /* write only; D32 */
```

A write with arbitrary data to this register (key address) will arm the sampling logic.
If Autostart mode is enabled then the sampling will also start.

### 4.16 Key address VME disarm sampling logic

```
#define SIS3320_KEY_DISARM                   0x414    /* write only; D32 */
```

A write with arbitrary data to this register (key address) will disarm the sampling logic.

### 4.17 Key address VME start sampling

```
#define SIS3320_KEY_START                    0x418    /* write only; D32 */
```

A write with arbitrary data to this register (key address) will initiate if the sampling logic is armed for sampling.

### 4.18 Key address VME stop sampling

```
#define SIS3320_KEY_STOP                     0x41C    /* write only; D32 */
```

A write with arbitrary data to this register (key address) will halt sampling on the active page. The Sample Logic will be disarmed also in Single Event Mode or if the Actual_Event_Counter reaches the value of the Max_Nof_Events register.

### 4.19 Key address Reset DDR2 Memory Logic

```
#define SIS3320_KEY_RESET_DDR2_LOGIC         0x428    /* write only; D32 */
```

A write with arbitrary data to this register (key address) will reset the DDR2 Memory Logic..
A possible memory synchronization problem can be detected with the 4 DDR2 test registers.

### 4.20 Event configuration registers

```
#define SIS3320_EVENT_CONFIG_ALL_ADC      0x01000000  /* write only;D32 */

#define SIS3320_EVENT_CONFIG_ADC12        0x02000000  /* read/write;D32 */
#define SIS3320_EVENT_CONFIG_ADC34        0x02800000  /* read/write;D32 */
#define SIS3320_EVENT_CONFIG_ADC56        0x03000000  /* read/write;D32 */
#define SIS3320_EVENT_CONFIG_ADC78        0x03800000  /* read/write;D32 */
```

This register is implemented for each channel group and it has to be written with the same value, the best way is to make use of the address `SIS3320_EVENT_CONFIG_ALL_ADC` to write to the registers of all channel groups simultaneously.

| Bit | Function |
|---|---|
| 31 | unused; read 0 |
| ... | ... |
| 16 | unused; read 0 |
| 15 | unused; read 0 |
| 14 | unused; read 0 |
| 13 | unused; read 0 |
| 12 | Enable User In (Control Input 1) to data stream |
| 11 | unused; read 0 |
| 10 | Enable User In (Control Input 1) as accumulator gate |
| 9 | Shift down accumulator result by 4 bits |
| 8 | Enable accumulator mode |
| 7 | unused; read 0 |
| 6 | unused; read 0 |
| 5 | Enable Sample Length stop |
| 4 | Enable Page Wrap around mode<br>0 : Wrap around full memory<br>1 : Wrap around page until STOP (External or KEY or Length Stop) |
| 3 | Wrap Page size Bit 3 |
| 2 | Wrap Page size Bit 2 |
| 1 | Wrap Page size Bit 1 |
| 0 | Wrap Page size Bit 0 |

Enable Sample Length stop:

0: no stop from sample Length (count) logic

1: sampling will stop after counts defined in register "Event Length register".

Enable Page Wrap around mode:

0 : Wrap around full memory (32 MSample)

1 : Wrap around in pages defined by pages size bits

The pages size is defined by this register in wrap event mode. The lowest four bits define the number of memory divisions as listed in the table below.

### 4.20.1 Wrap Page size

The page/event size is defined by the 4 page size bits as follows:

| Page size Bit 3 | Page size Bit 2 | Page size Bit 1 | Page size Bit 0 | Page size |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 16 M Samples |
| 0 | 0 | 0 | 1 | 4 M Samples |
| 0 | 0 | 1 | 0 | 1 M Samples |
| 0 | 0 | 1 | 1 | 256 K Samples |
| 0 | 1 | 0 | 0 | 64K Samples |
| 0 | 1 | 0 | 1 | 16 K Samples |
| 0 | 1 | 1 | 0 | 4 K Samples |
| 0 | 1 | 1 | 1 | 1 K Samples |
| 1 | 0 | 0 | 0 | 512 Samples |
| 1 | 0 | 0 | 1 | 256 Samples |
| 1 | 0 | 1 | 0 | 128 Samples |
| 1 | 0 | 1 | 1 | 64 Samples |
| 1 | 1 | 0 | 0 | reserved |
| 1 | 1 | 0 | 1 | reserved |
| 1 | 1 | 1 | 0 | reserved |
| 1 | 1 | 1 | 1 | reserved |

### 4.20.2 Accumulator mode

Accumulator mode is considered as an experimental feature (firmware version 0x104). It can be used to sum up the digital values for the duration of an external gate. The accumulator sum is limited to 36-bits and implies operation in non wrapping mode (as the event directory is used for the sum instead for the stop pointer).
Accumulator mode is inactive with Bit 8=0 and active with Bit 8=1.
The 36-bit deep accumulator sum is cleared at the beginning of an event. The digitized ADC values are added during the sampling process. Samples with active User Input LEMO signal only are added with bit 10 set (Enable User In as accumulator gate).
32-bits of the accumulated sum are stored to the event directory at the end of sampling according to the table below

| Bit 9 | Store to event directory |
|---|---|
| 0 | Bits 31:0 of accumulated sum |
| 1 | Bits 35:4 of accumulated sum |

### 4.21 Sample Length register

```
#define SIS3320_SAMPLE_LENGTH_ALL_ADC          0x01000004

#define SIS3320_SAMPLE_LENGTH_ADC12            0x02000004
#define SIS3320_SAMPLE_LENGTH_ADC34            0x02800004
#define SIS3320_SAMPLE_LENGTH_ADC56            0x03000004
#define SIS3320_SAMPLE_LENGTH_ADC78            0x03800004
```

This register defines the number of samples if the "Enable Max. Sample count stop" is enabled.

The register can be used to signal, that a page was completely filled in Stop Mode (sampling wraps in memory) with "Enable Max. Sample count stop" disabled.
Example:
Wrap in 1 K page, the wrap bit will be set in the event directory as soon as the value of the Length register is reached.

Value of Sample Length register = (Sample Length - 4) & 0xfffffC

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 25 | unused, read as 0 |
| 24 | Sample Length Register BIT24 |
| .. | |
| 2 | Sample Length Register BIT2 |
| 1 | Unused |
| 0 | Unused |

The power up default value is 0

**Example:** Desired sample length 0x100 -> set the register to 0xFC

### *4.22 Sample Start address register*

```
#define SIS3320_SAMPLE_START_ADDRESS_ALL_ADC          0x01000008

#define SIS3320_SAMPLE_START_ADDRESS_ADC12            0x02000008
#define SIS3320_SAMPLE_START_ADDRESS_ADC34            0x02800008
#define SIS3320_SAMPLE_START_ADDRESS_ADC56            0x03000008
#define SIS3320_SAMPLE_START_ADDRESS_ADC78            0x03800008
```

These registers define the memory start address.
The value is given in samples (i.e. number of 16-bit words)

Only Sample Start addresses on a 4 16-bit (sample) boundary (i.e. 8 bytes) are valid.

| Bit | |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 25 | unused, read as 0 |
| 24 | Sample Start Address Register Bit 24 |
| .. | |
| 2 | Sample Start Address Register Bit 2 |
| 1 | unused |
| 0 | unused |

The power up default value is 0

#### 4.22.1 Explanation (sample start address)

The contents of the start sample register is assigned as memory data storage address with the arm command (key address arm sampling). The user has to distinguish between two the two wrap modes of the SIS3320.

**No Page wrap around mode (event configuration register bit 4 =1):**
Bits 24:2 of the sample start register are used as memory start address upon the arm condition. The lowest 2 bits are 0 (i.e. the start address is on a 4 sample or 8 byte boundary). The internal address is incremented by 1 with every sample clock.
The stop will be on a 4 sample boundary.
The next (adjacent) address will be used as the starting point for the next event in multi event mode.

**Page wrap around mode (event configuration register bit 4 =0):**
The upper part of the sample start register (page address) is used as memory start address upon the arm command. The lower bits (number of bits defined by the page size, bits 9:0 for a page size of 1K e.g.). The lower portion is incremented by 1 with every sample clock. The stop will be on a 4 sample boundary also. The exact stop location can be obtained from the lowest 2 bits of the next sample register or the event directory.
The upper portion of the address (page address) is incremented by 1 with the start of the next event and the lower portion is cleared again.

### 4.23 ADC Input mode register

```
#define SIS3320_ADC_INPUT_MODE_ALL_ADC          0x0100000C

#define SIS3320_ADC_INPUT_MODE_ALL_ADC12        0x0200000C
#define SIS3320_ADC_INPUT_MODE_ALL_ADC34        0x0280000C
#define SIS3320_ADC_INPUT_MODE_ALL_ADC56        0x0300000C
#define SIS3320_ADC_INPUT_MODE_ALL_ADC78        0x0380000C
```

This register set is used to generate memory test data.

| Bit | Function |
|---|---|
| 31 | Unused; read 0 |
| ... | ... |
| 20 | Unused; read 0 |
| 19 | |
| 18 | |
| 17 | Test Data 32-bit Mode : 0 = 16-bit mode ; 1 = 32-bit mode |
| 16 | Enable Simulate ADC Test Data Mode |
| 15 | ADC Test Start Data Bit 15 |
| .. | |
| 0 | ADC Test Start Data Bit 0 |

Test Start Data has to be different from 0xYYFE and 0xYYFF.

An increment pattern is recorded in test mode (Bit 16 = 1) instead of real ADC data. Data are incremented by 2 in 32-bit mode.

### *4.24 ADC1-8 Next Sample address register*

```
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC1          0x02000010
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC2          0x02000014
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC3          0x02800010
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC4          0x02800014
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC5          0x03000010
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC6          0x03000014
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC7          0x03800010
#define SIS3320_ACTUAL_SAMPLE_ADDRESS_ADC8          0x03800014
```

These 8 read only registers hold the next sampling address for the given channel.

| Bit | Function |
|---|---|
| 31 | unused, read as 0 |
| ... | |
| 25 | unused, read as 0 |
| 24 | Sample Address Bit 24 |
| .. | |
| 2 | Sample Address Bit 2 |
| 1 | Sample Address Bit 1* |
| 0 | Sample Address Bit 0* |

The power up default value is 0

* Sample address bits 1 and 0 have a dedicated meaning.

Explanation:

Data are stored to memory in packets of 4 consecutive samples by the sample logic. This implies, that either one additional sample is stored in stop mode (wrap mode respective ) or that the last 1-2 samples before the stop are not stored.
Sample address bits 1:0 can be used to obtain the position of the stop signal within the 4 sample packet.
The stop delay can be used to make sure, that all samples up to the external stop condition are written to memory.

| Sample Address Bits 1:0 | Correction |
|---|---|
| 3 | - 1 |
| 0 | 0 |
| 1 | + 1 |
| 2 | + 2 |

### 4.25 Actual Sample Value registers

```
#define SIS3320_ACTUAL_SAMPLE_VALUE_ADC12          0x02000020
#define SIS3320_ACTUAL_SAMPLE_VALUE_ADC34          0x02800020
#define SIS3320_ACTUAL_SAMPLE_VALUE_ADC56          0x03000020
#define SIS3320_ACTUAL_SAMPLE_VALUE_ADC78          0x03800020
```

Read "on the fly" of the actual converted ADC values.
The read only registers are updated with every ADC clock, unless a concurrent VME read access is pending.
The register contents is refreshed and can be read any time (i.e. they are updated independent of the unarmed, armed, sampling state) as long as a sampling clock is distributed on the ADC board (internal clock or active/clocking external clock)

| ADC 1 / 3 / 5 / 7 | | ADC 2 / 4 / 6 / 8 | |
|---|---|---|---|
| D31:28 | D27:16 | D15:12 | D11:0 |
| 0 | 12-bit data | 0 | 12-bit data |

### 4.26 DDR2 Memory Logic Verification registers

```
#define SIS3320_DDR2_TEST_REGISTER_ADC12           0x02000028
#define SIS3320_DDR2_TEST_REGISTER_ADC34           0x02800028
#define SIS3320_DDR2_TEST_REGISTER_ADC56           0x03000028
#define SIS3320_DDR2_TEST_REGISTER_ADC78           0x03800028
```

| ADC 2 / 4 / 6 / 8 | | | |
|---|---|---|---|
| D31:28 | D27:24 | D23:20 | D19:16 |
| Fifo_11_wr_addr | Fifo_10_wr_addr | Fifo_01_wr_addr | Fifo_00_wr_addr |

| ADC 1 / 3 / 5 / 7 | | | |
|---|---|---|---|
| D15:12 | D11:8 | D7:4 | D3:0 |
| Fifo_11_wr_addr | Fifo_10_wr_addr | Fifo_01_wr_addr | Fifo_00_wr_addr |

This register set can be used to verify proper operation of the DDR2 memory read logic. The contents of all registers has to be identical. So far no deviation/error condition with firmware V1.02 has been detected. Please report problems to Struck Innovative Systeme. The error can be salvaged by issuing a key reset DDR2 memory logic command.

### *4.27 Trigger setup register registers*

```
#define SIS3320_TRIGGER_SETUP_ADC1          0x02000030
#define SIS3320_TRIGGER_SETUP_ADC2          0x02000038
#define SIS3320_TRIGGER_SETUP_ADC3          0x02800030
#define SIS3320_TRIGGER_SETUP_ADC4          0x02800038
#define SIS3320_TRIGGER_SETUP_ADC5          0x03000030
#define SIS3320_TRIGGER_SETUP_ADC6          0x03000038
#define SIS3320_TRIGGER_SETUP_ADC7          0x03800030
#define SIS3320_TRIGGER_SETUP_ADC8          0x03800038
```

These read/write registers hold the Peaking and Gap Time of the trapezoidal FIR filter.
(Gap Time = SumG Time – Peaking Time)

| Bit | Function | |
|---|---|---|
| 31 | reserved; read 0 | |
| .. | .. | |
| 24 | reserved; read 0 | |
| 23 | Puls Length bit 7 | Trigger Pulse Length |
| 22 | Puls Length bit 6 | |
| 21 | Puls Length bit 5 | |
| 20 | Puls Length bit 4 | |
| 19 | Puls Length bit 3 | |
| 18 | Puls Length bit 2 | |
| 17 | Puls Length bit 1 | |
| 16 | Puls Length bit 0 | |
| 15 | reserved | SumG time (time between both sums) |
| 14 | reserved | |
| 13 | reserved | |
| 12 | SumG bit 4 | |
| 11 | SumG bit 3 | |
| 10 | SumG bit 2 | |
| 9 | SumG bit 1 | |
| 8 | SumG bit 0 | |
| 7 | reserved | Peaking time P $\sum_{i=x}^{x+P} Si$ |
| 6 | reserved | |
| 5 | reserved | |
| 4 | P bit 4 | |
| 3 | P bit 3 | |
| 2 | P bit 2 | |
| 1 | P bit 1 | |
| 0 | P bit 0 | |

The power up default value reads 0x 00000000

Si: Sum of ADC input sample stream from x to x+P
P: Peaking time (number of values to sum)
SumG: SumGap time (distance in clock ticks of the two running sums)

The maximum SumG time: 16 (clocks)
The minimun SumG time: 1 (clocks)
Values > 16 will be set to 16
Value = 0 will be set to 1

The maximum Peaking time: 16 (clocks)
The minimun Peaking time: 1 (clocks)
Values > 16 will be set to 16
Value = 0 will be set to 1

### *4.28 Threshold registers*

```
#define SIS3320_TRIGGER_THRESHOLD_ADC1          0x02000034
#define SIS3320_TRIGGER_THRESHOLD_ADC2          0x0200003C
#define SIS3320_TRIGGER_THRESHOLD_ADC3          0x02800034
#define SIS3320_TRIGGER_THRESHOLD_ADC4          0x0280003C
#define SIS3320_TRIGGER_THRESHOLD_ADC5          0x03000034
#define SIS3320_TRIGGER_THRESHOLD_ADC6          0x0300003C
#define SIS3320_TRIGGER_THRESHOLD_ADC7          0x03800034
#define SIS3320_TRIGGER_THRESHOLD_ADC8          0x0380003C
```

These read/write registers hold the threshold values for the ADC channels.

| Bit | 31-26 | 25 | 24 | 16-0 |
|---|---|---|---|---|
| Function | none | Trigger Mode GT | Trigger Mode LT | **Trapezoidal** threshold value |

default after Reset:

The value of the Sum (trapezoidal value) depends on the peaking time P. Therefore the selection of the value of the Trapezoidal threshold depends on P also.
See Trigger Example.

Trapezoidal value calculation:

Trapezoidal value = (SUM2 – SUM1) + 0x10000

Where

$$SUM1 = \sum_{i=x}^{x+P} Si$$

$$SUM2 = \sum_{j=x+sumG}^{x+P+sumG} Sj$$

The FIR Filter logic adds 0x10000 to the result of the subtraction of the two running sums. This implies, that the internal value of the trapezoid is on average 0x10000.

A Trigger Output pulse is generated on two conditions:

1. LT is set: the Trigger Out Pulse will be issued if the actual trapezoidal value **goes** below the programmable trapezoidal threshold value.

2. GT is set: the Trigger Out Pulse will be issued if the actual trapezoidal value **goes** above the programmable trapezoidal threshold value

### *4.29 Trigger Flag Clear Counter register*

```
#define SIS3320_TRIGGER_FLAG_CLR_CNT_ADC12          0x0200002C
#define SIS3320_TRIGGER_FLAG_CLR_CNT_ADC34          0x0280002C
#define SIS3320_TRIGGER_FLAG_CLR_CNT_ADC56          0x0300002C
#define SIS3320_TRIGGER_FLAG_CLR_CNT_ADC78          0x0380002C
```

The Trigger Flag bit is set as soon as an ADC channel meets the trigger criterion. This flag remains latched until the next event start, i.e. it will not be cleared as new ADC data which do not meet the trigger criterion come in with Wrap mode active.

The Trigger Flag Clear Counter register allows you to define a number of 100MHz clocks (NOT sample clock !!) after which the Trigger Flag bit will be cleared unless a new trigger occurred. A counter (for the given ADC channel) is preloaded with the value of the Trigger Flag Clear counter register when the trigger criterion for this channel is met.
The 100 MHz clock will decrement the counter and the Trigger Flag bit will be cleared as soon as the counter reaches 1. If a new trigger occurs before the counter has reached 1, it will be reloaded with the value from the register (retrigger).

**Note:** typically the user may want to set the value of the Trigger Flag Clear counter register to the memory page size , but this is not mandatory.

The Trigger Flag Clear Logic is disabled if the counter is loaded with 0 (power up default).

| Bit | 31-0 |
|---|---|
| Function | Trigger Flag Clear counter register |

The power up default value is 0

### 4.30 Event directories

```
#define SIS3320_EVENT_DIRECTORY_ADC1          0x02010000
#define SIS3320_EVENT_DIRECTORY_ADC2          0x02018000
#define SIS3320_EVENT_DIRECTORY_ADC3          0x02810000
#define SIS3320_EVENT_DIRECTORY_ADC4          0x02818000
#define SIS3320_EVENT_DIRECTORY_ADC5          0x03010000
#define SIS3320_EVENT_DIRECTORY_ADC6          0x03018000
#define SIS3320_EVENT_DIRECTORY_ADC7          0x03810000
#define SIS3320_EVENT_DIRECTORY_ADC8          0x03818000
```

The event directories hold the stop pointer(s) (i.e. sample end address+1) of each channel.
The directories are 512 words deep and 32 bits wide. A wrap around bit (i.e. bit 28) will be set if the page was filled at least once (i.e. if the memory pointer has reached the end )
The event directory holds the accumulator sum instead of the next sample address in accumulator mode .

| Vme offset address | (D31:30) | (D29) | (D28) | (D27:D25) | Event Next Sample Address (D24:D0) |
|---|---|---|---|---|---|
| 0x0 | 00 | T | W | 000 | (Next Sample Address) of Event 1 |
| .. | ... | ... | ... | | ... |
| 0x7fc | 00 | T | W | 000 | (Next Sample Address) of Event 512 |

W: wrap around bit
T: trigger bit

Sample End Address = Next Sample Address - 1

Wrap around bit:
This bit is cleared at start of sampling and it is set when the number of samples reached the value of the Sample Length register.
See also "Sample Length register".

Trigger bit:
The trigger bit is set with the internal trigger condition is valid.
See also "Trigger Flag Clear Counter register".

See "Next Sample Address Register" for more Information on the Next Sample Address

### *4.31 ADC memory*

```
#define SIS3320_ADC1_OFFSET                    0x04000000
#define SIS3320_ADC2_OFFSET                    0x04800000
#define SIS3320_ADC3_OFFSET                    0x05000000
#define SIS3320_ADC4_OFFSET                    0x05800000
#define SIS3320_ADC5_OFFSET                    0x06000000
#define SIS3320_ADC6_OFFSET                    0x06800000
#define SIS3320_ADC7_OFFSET                    0x07000000
#define SIS3320_ADC8_OFFSET                    0x07800000
```

The 64 MByte ADC memory per channel can be address in pages of 8 MByte. The page is selected with the ADC Memory page register. One 32-bit word holds 2 ADC samples as shown in the table below.

## Data format:

| VME offset address | D31 | D30:28 | D27:16 | D15 | D14:12 | D11:0 |
|---|---|---|---|---|---|---|
| X + 0x0 | U | 0 | 12-bit data sample N+1 | U | 0 | 12-bit data sample N |
| X + 0x4 | U | 0 | 12-bit data sample N+3 | U | 0 | 12-bit data sample N+2 |
| X + 0x8 | U | 0 | 12-bit data sample N+5 | U | 0 | 12-bit data sample N+4 |

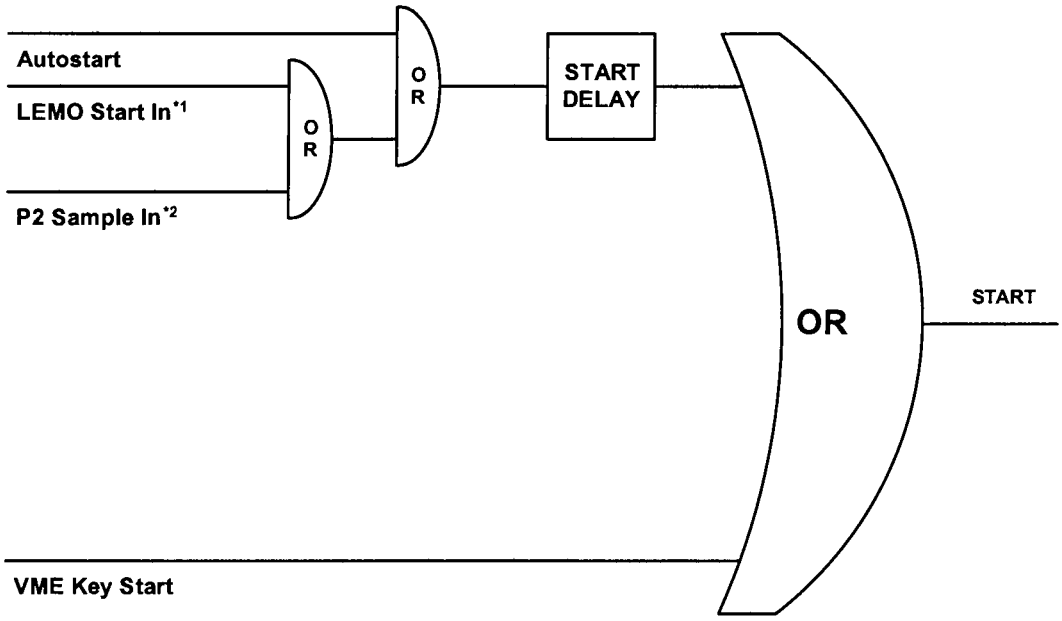
The content U of bits D15 and D31 depends on the status of bit 12 of the event configuration register and possibly the logic level of the user input.

| Bit 12 of Evt. Config Reg. | User input logic level | D31/D15 |
|---|---|---|
| 0 | 0 | U=0 |
| 0 | 1 | U=0 |
| 1 | 0 | U=0 |
| 1 | 1 | U=1 |

# 5 Description of Start/Stop modes of operation

## 5.1 Start logic summary

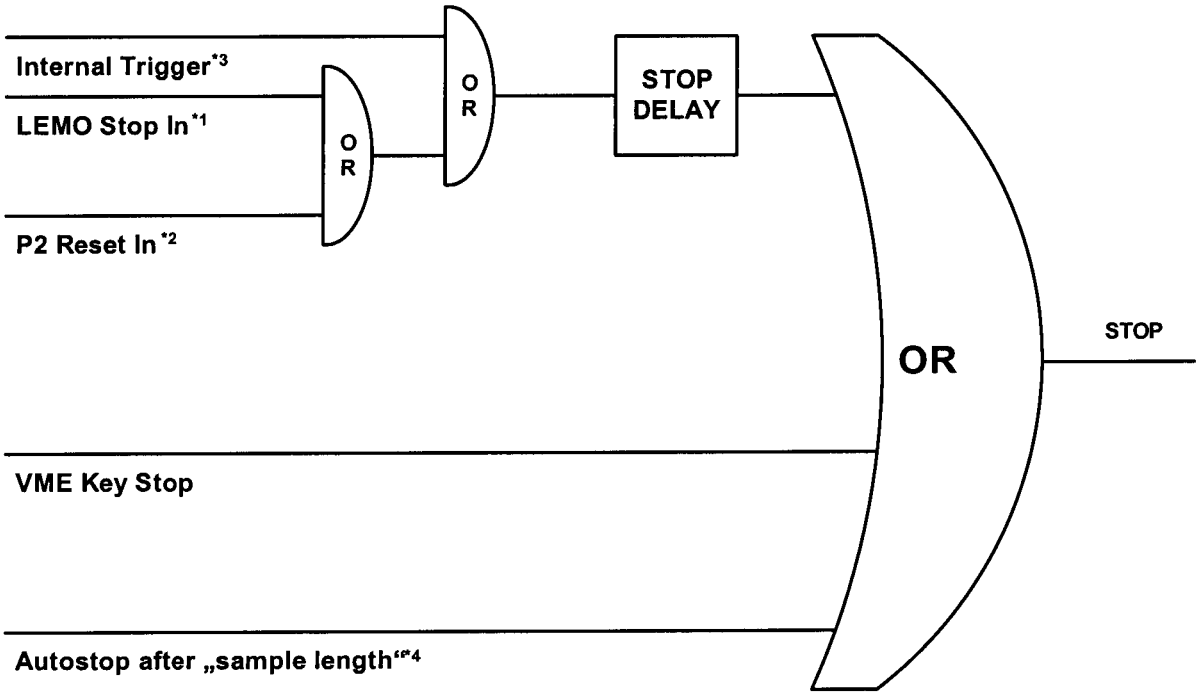
The diagram below illustrates the implemented start conditions of the SIS3320,




| Note | Condition | Register | Comment |
|---|---|---|---|
| *1 | `Bit 8 = 1` | Acquisition Control | Enable front panel start/stop logic |
| *2 | `Bit 9 = 1` | Acquisition Control | not implemented yet (Enable P2 start/stop logic ) |

## 5.2 Stop logic summary

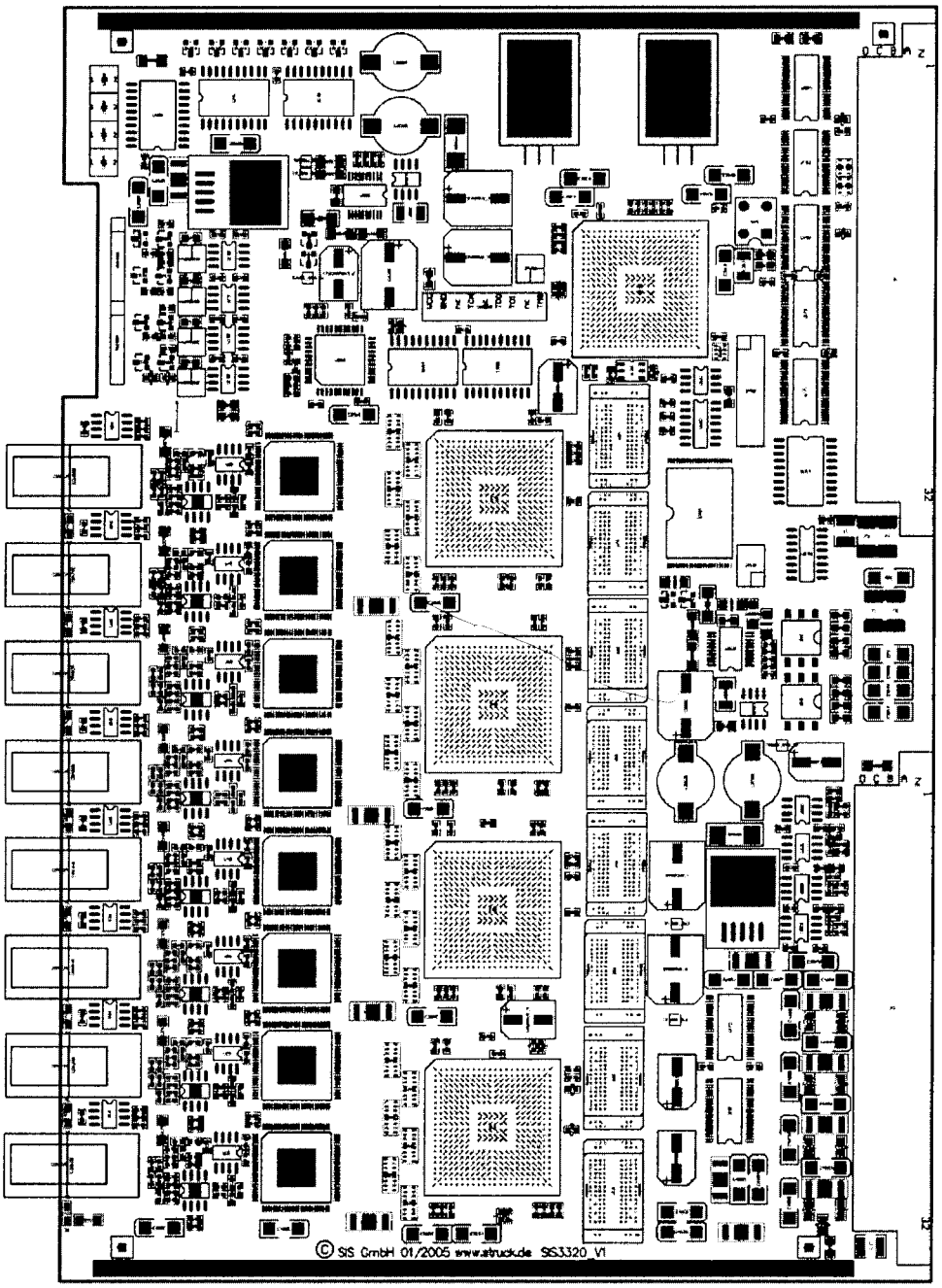
The diagram below illustrates the implemented stop conditions of the SIS3320.




| Note | Condition | Register | Comment |
|---|---|---|---|
| *1 | `Bit 8 = 1` | Acquisition Control | Enable front panel start/stop logic |
| *2 | `Bit 9 = 1` | Acquisition Control | not implemented yet (Enable P2 start/stop logic) |
| *3 | `Bit 6 = 1` | Acquisition Control | internal trigger as stop |
| *3 | `Bit 5 = 1` | Event Configuration | Enable Max_Sample_count_stop |

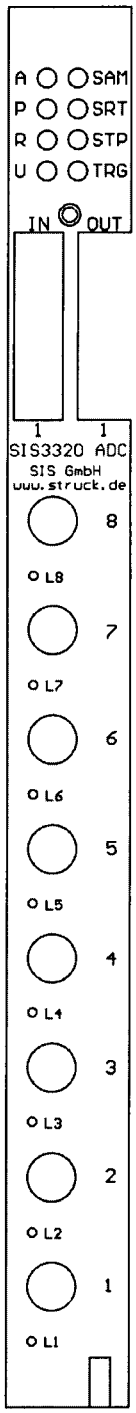
# 6 Board layout

A printout of the silk screen of the component side of the PCB is shown below.



# 7 Front panel

The SIS3320 is a single width (4TE) 6U VME module. A sketch of the SIS3320 front panel (without handles) is shown below. The IN/OUT breakouts hold 4 LEMO connectors each.



## 7.1 Control In/Outputs

The control I/O section features 8 LEMO00 connectors with NIM levels.

| Designation | Inputs | Outputs | Designation |
|---|---|---|---|
| 4 | Clock In | Clock Out | 4 |
| 3 | Start | Ready for start (ADC sample logic armed) | 3 |
| 2 | Stop | Ready for stop (ADC sampling busy) | 2 |
| 1 | User in | Trigger output | 1 |

The ready for start and ready for stop outputs can be used to interfere with external deadtime logic. Ready for start will become active as soon as the sample clock for one of the banks is active. Ready for stop will go active as soon as the start signal was seen by the module.

The external clock must be a symmetric signal unless the module is operated in external random clock mode
The width of an external start/stop pulse must be greater or equal two sampling clock periods.

### 7.1.1 User input

User input functionality was implemented to allow for synchronous recording of one external status bit (like chopper on/off e.g.) with the ADC data stream. The user bin information is recorded with the ADC data (see section 4.31). The current status of the logic level is represented by Bit 16 of the status register.

not implemented in 0102

### 7.2 LED's

The SIS3320 has 8 front panel LEDs to visualise part of the modules status. The user (and access) LED are a good way to check first time communication/addressing with the module.

| Color | Designator | Function |
|---|---|---|
| Red | A | Access to SIS3320 VME slave port |
| Yellow | P | Power |
| Green | R | Ready, on board logic configured |
| Green | U | User, to be set/cleared under program control |
| Red | SAM | Sampling busy |
| Yellow | SRT | Start, lit with start input (or leading edge in gate mode) |
| Green | STP | Stop, lit with stop input (or trailing edge in gate mode) |
| Green | TRG | Trigger, lit if one or more channels are above threshold |

The on duration of the access, sampling, start, stop and trigger LEDs is stretched to guarantee visibility even under low rate conditions.

### 7.3 Channel LED's

The 8 card edge surface mounted LEDs L1, ..., L8 can be seen through the corresponding holes in the front panel. They visualize the trigger status of the corresponding channel. The on duration is stretched for better visibility of short pulses.

### 7.4 PCB LEDs

The 8 surface mounted red LEDs D141A to D141G on the top left corner of the component side of the SIS3320 are routed to the control FPGA, their use may depend on the firmware design.

# 8 Jumpers/Configuration

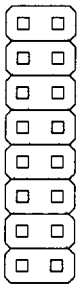
## 8.1 CON100 JTAG

The SIS3320 on board logic can load its firmware from a serial PROMs , via the JTAG port on connector CON100 or over VME. A list of firmware designs can be found under http://www.struck.de/sis3320firm.htm.
Hardware like the XILINX HW-JTAG-PC in connection with the appropriate software will be required for in field JTAG firmware upgrades. The JTAG chain configuration is selected with jumper JP101,jumper JP102 is used to chose VME or CON100 as JTAG source.
The JTAG connector is a 9 pin single row 1/10 inch header, the pin assignment on the connector can be found in the table below.

| Pin | Short hand | Description |
|---|---|---|
| 1 | VCC | Supply voltage |
| 2 | GND | Ground |
| 3 | nc | not connected, cut to avoid polarity mismatch |
| 4 | TCK | test clock |
| 5 | nc | not connected |
| 6 | TDO | test data out |
| 7 | TDI | test data in |
| 8 | nc | not connected |
| 9 | TMS | test modus |

## 8.2 JP80 VME addressing mode/reset behaviour

This 8 position jumper array is used to select the addressing mode and the reset behaviour of the SIS3320.




| Pos | Function | Factory default |
|---|---|---|
| 1 | A32 | closed |
| 2 | A16 (not supported) | open |
| 3 | GEO (not supported) | open |
| 4 | VIPA (not supported) | open |
| 5 | connect VME SYSRESET IN to FPGA reset | closed |
| 6 | connect watchdog to VME SYSRESET OUT | open |
| 7 | connect FPGA reset VME SYSRESET OUT | open |
| 8 | connect VME SYSRESET to board reset | closed |

The enable watchdog jumper has to be removed during (initial) JTAG firmware load.

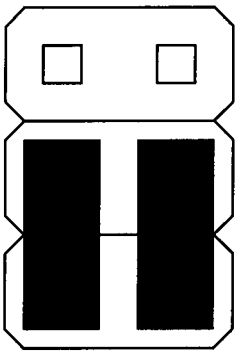
**NOTE:** avoid a power up deadlock situation by not setting Pos. 5 and 7 at the same time

### *8.3 JP101 JTAG chain*

The JTAG chain on the SIS3320 can be configured to comprise the serial PROM only (short JTAG chain) or to comprise the serial PROM and the 5 Spartan III FPGAs (long chain). The configuration is selected with the 6-pin array JP101 as sketched below:

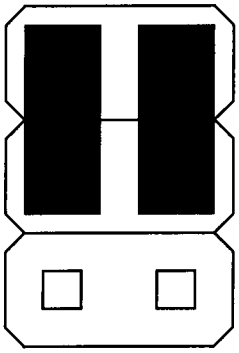
Long Chain (1-3 and 2-4 closed):

**JP101**



Short Chain (3-5 and 4-6 closed, factory default):

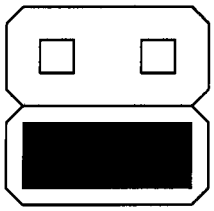
**JP101**



### *8.4 JP102 JTAG source*

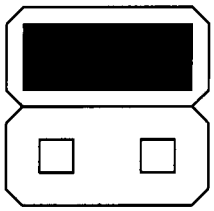
The JTAG chain can be connected to VME or to the JTAG connector CON...via the 4 pin jumper array JP102 as sketched below:

JTAG connected to VME (1-2 closed)



**JP102**

JTAG connected to connector CON100 (3-4 closed, factory default)

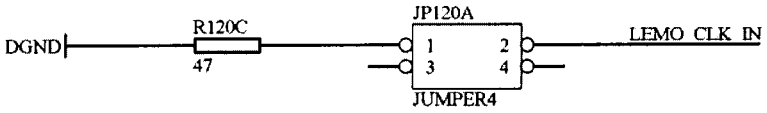


**JP102**

### 8.5 JP120A-JP120D control input termination

The contact pair 1-2 of these 4 jumper arrays is used to connect the termination resistor to the 4 control inputs as illustrated with the schematic for JP120A below.

| Jumper | Control Input | Factory Default |
|---|---|---|
| JP120A | Clock In | Closed |
| JP120B | Start In | Closed |
| JP120C | Stop In | Closed |
| JP120D | User In | Closed |




### 8.6 SW1 and SW2, VME base address

These 2 rotary switches are used to define 2 nibbles of the VME base address in non geographical addressing (refer to section base address also).

| Switch | Function |
|---|---|
| SW1 | ADR_LO |
| SW2 | ADR_UP |

# 9 Getting started

The directory sis3320\software\win of the Struck Innovative Systeme CDROM holds example code for VisualC++ and National Instruments Labwindows CVI. The source code can be used as a base for ports to other environments.
The sis3320.h header file can be found in the directory sis3320\software.
The routine `ConfigurationSetupAdc(void)` in the file `sis3320_adc_test1.c` (CVI directory) can be used as starting point for a setup routine for the SIS3320.

# 10 Appendix

## *10.1 Power consumption*

The SIS3320 uses standard VME voltages only.

| Voltage | Current |
|---|---|
| + 5V | 6A |
| +12 V | 100 mA |
| - 12 V | 470 mA |

## *10.2 Operating conditions*

### 10.2.1 Cooling

Although the SIS3320 is mainly a 2.5 and 3.3 V low power design, substantial power is consumed by the Analog to Digital converter chips and linear regulators however. Hence forced air flow is required for the operation of the board. The board may be operated in a non condensing environment at an ambient temperature between 10° and 25° Celsius. A power up warm up time of some 10 minutes is recommended to ensure equilibrium on board temperature conditions.

### 10.2.2 Hot swap/live insertion

Please note, that the VME standard does not support hot swap by default. The SIS3320 is configured for hot swap in conjunction with a VME64x backplane. In non VME64x backplane environments the crate has to be powered down for module insertion and removal.

### 10.3 Connector types

The VME connectors and the different types of front panel connectors used on the SIS3320 are:

| Connector | Purpose | Part Number |
|---|---|---|
| 160 pin zabcd | VME P1/P2 | Harting 02 01 160 2101 |
| LEMO PCB | Coax. control connector | LEMO EPB.00.250.NTN |
| 90° PCB LEMO | Analog input connector | LEMO EPL.00.250.NTN |
| 90° PCB LEMO | Analog input connector (3320 differential input version) | LEMO EPL.0S.302.HLN |

### 10.4 P2 row A/C pin assignments

The P2 connector of the SIS3320 has several connections on rows A and C for the F1002 compatible use at the DESY H1 FNC subdetector. This implies, that the module can not be operated in a VME slot with a special A/C backplane, like VSB e.g.. The pin assignments of P2 rows A/C of the SIS3320 is shown below:

| P2A | Function | P2C | Function |
|---|---|---|---|
| 1 | -5.2 V | 1 | -5.2 V |
| 2 | -5.2 V | 2 | -5.2 V |
| 3 | -5.2 V | 3 | -5.2 V |
| 4 | not connected | 4 | not connected |
| 5 | not connected | 5 | not connected |
| 6 | DGND | 6 | DGND |
| 7 | P2_CLOCK_H | 7 | P2_CLOCK_L |
| 8 | DGND | 8 | DGND |
| 9 | P2_START_H | 9 | P2_START_L |
| 10 | P2_STOP_H | 10 | P2_STOP_L |
| 11 | P2_TEST_H | 11 | P2_TEST_L |
| 12 | DGND | 12 | DGND |
| 13 | DGND | 13 | DGND |
| 14 | DGND | 14 | DGND |
| 15 | DGND | 15 | DGND |
| 16 | not connected | 16 | not connected |
| ... | ... | 17 | ... |
| 31 | not connected | 18 | not connected |

**Note:** The P2 ECL signals are bussed and terminated on the backplane of F1002 crates. The user has to insure proper termination if a cable backplane or add on backplane is used.

### *10.5 Row d and z Pin Assignments*

The SIS3320 is prepared for the use with VME64x and VME64xP backplanes. Foreseen features include geographical addressing (PCB revisions V2 and higher) and live insertion (hot swap). The prepared pins on the d and z rows of the P1 and P2 connectors are listed below.

| Position | P1/J1 | | P2/J2 | |
|---|---|---|---|---|
| | Row z | Row d | Row z | Row d |
| 1 | | VPC (1) | | |
| 2 | GND | GND (1) | GND | |
| 3 | | | | |
| 4 | GND | | GND | |
| 5 | | | | |
| 6 | GND | | GND | |
| 7 | | | | |
| 8 | GND | | GND | |
| 9 | | GAP* | | |
| 10 | GND | GA0* | GND | |
| 11 | RESP* | GA1* | | |
| 12 | GND | | GND | |
| 13 | | GA2* | | |
| 14 | GND | | GND | |
| 15 | | GA3* | | |
| 16 | GND | | GND | |
| 17 | | GA4* | | |
| 18 | GND | | GND | |
| 19 | | | | |
| 20 | GND | | GND | |
| 21 | | | | |
| 22 | GND | | GND | |
| 23 | | | | |
| 24 | GND | | GND | |
| 25 | | | | |
| 26 | GND | | GND | |
| 27 | | | | |
| 28 | GND | | GND | |
| 29 | | | | |
| 30 | GND | | GND | |
| 31 | | GND (1) | | GND (1) |
| 32 | GND | VPC (1) | GND | VPC (1) |

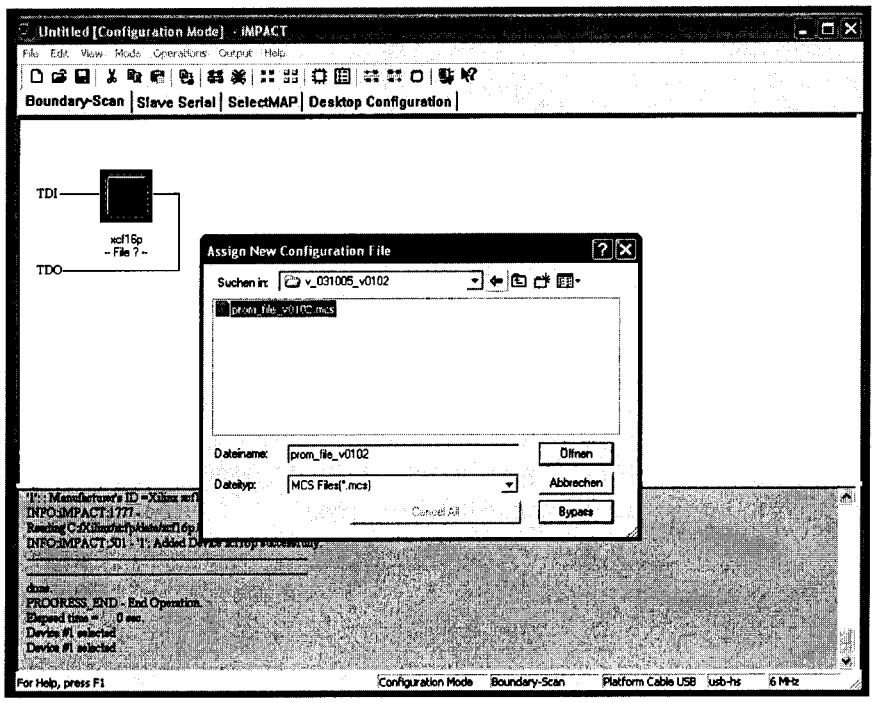
**Note: Pins designated with (1) are so called MFBL (mate first-break last) pins on the installed 160 pin connectors, VPC(1) pins are connected via inductors.**

### *10.6 Firmware upgrade*

The firmware of the SIS3320 can be upgraded over JTAG. The upgrade options are VME (on units that have intact firmware) and the JTAG connector CON100. The VME upgrade option is not tested for the current 01 02 firmware release yet.

#### 10.6.1 Upgrade over CON100

The firmware can be upgraded with the Xilinx Impact software, which is part of the Webpack that can be downloaded from the Xilinx web page for free. A version of the Webpack software (which may not be up to date and not compatible with your JTAG hardware) can be found in the xilinx_webpack directory of the Struck Innovative Systeme CDROM also. A Xilinx JTAG parallel cable or USB (Xilinx part number HW-USB) cable can be used to roll in the firmware.
Configure the SIS3320 for short JTAG chain (refer to section 8.3 JP101) and set the unit to JTAG over CON100 (refer to section 8.4 JP102 JTAG source).
With your hard- and software properly set up you should see a screen as illustrated below after executing the initialize chain command.




Load the mcs file to the serial PROM (shown as xcf16p).

#### 10.6.2 Upgrade over VME

not tested with SIS3320 firmware 01 02 yet

# 11 Index