SIS3302
12xx Firmware - Gamma

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: sis3302-M-1201-1-V101-gamma as of 12.02.08

Revision Table:

Revision	Date	Modification
0.01	27.11.07	Generation from SIS9300 Gamma
1.00	21.01.08	First official release
1.01	12.02.08	Add VME Addressing

1 Table of contents

2 Introduction

This firmware manual/addendum describes the functionality and implementation of the SIS3302 firmware major revision 0x12. Besides finite response filter (FIR) based triggering this version supports asynchronous readout of a programmable set of features of raw (i.e. digitized wave form) and/or computed (like signal height/energy e.g.) digitizer information.
No hardware modification to the SIS3302 is required for installation of this firmware.
This firmware implementation should be of particular interest for detector studies with Gamma ray tracking and strip detectors.

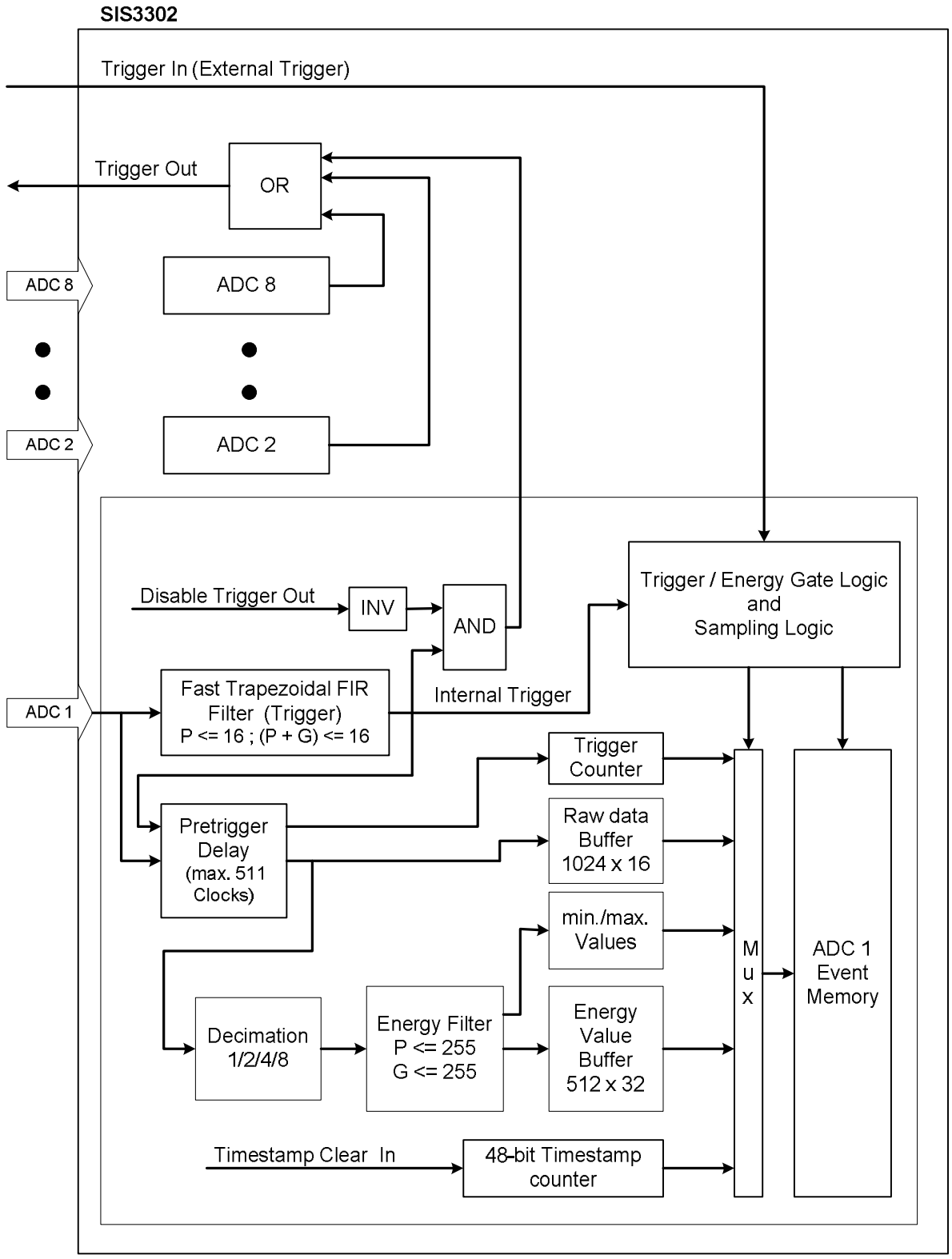
2.1 Functionality

The main functions of the firmware are listed below and illustrated in the Gamma logic block diagram.

- 8 channel asynchronous and synchronous operation
- Decimation
- Trapezoidal Energy filter
- Trigger FIR filter
- Trigger or output
- Timestamp
- Flexible event storage

2.2 Gamma Logic Implementation

2.2.1 General block diagram of one ADC channel and the full module



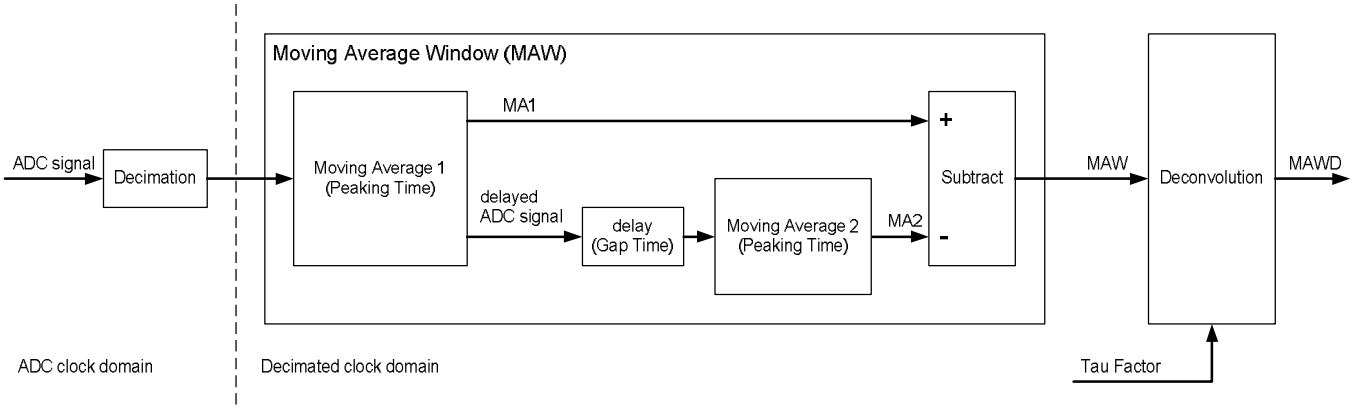
2.2.2 Trapezoidal Energy Filter

The trapezoidal MAWD algorithm is based on firmware V_040x (V_0403) which was implemented originally for the SIS3150/9300 combination.

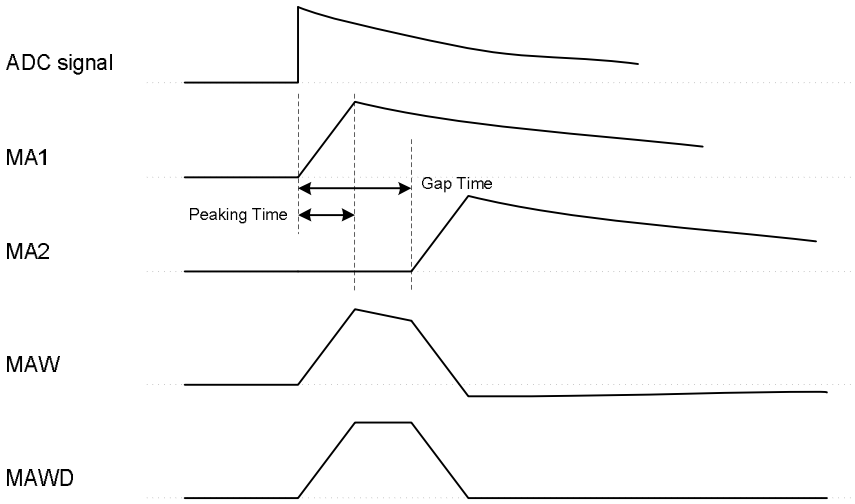
The Trapezoidal Energy Filter parameters are:

- MAWD: moving average window deconvolution (Tau correction)
- MAW: moving average window
- MA: moving average
- Decimation: decreasing the sample rate
- Peaking Time: the length of the MA for moving average unit
- Gap Time: the differentiation time of the mowing window average unit

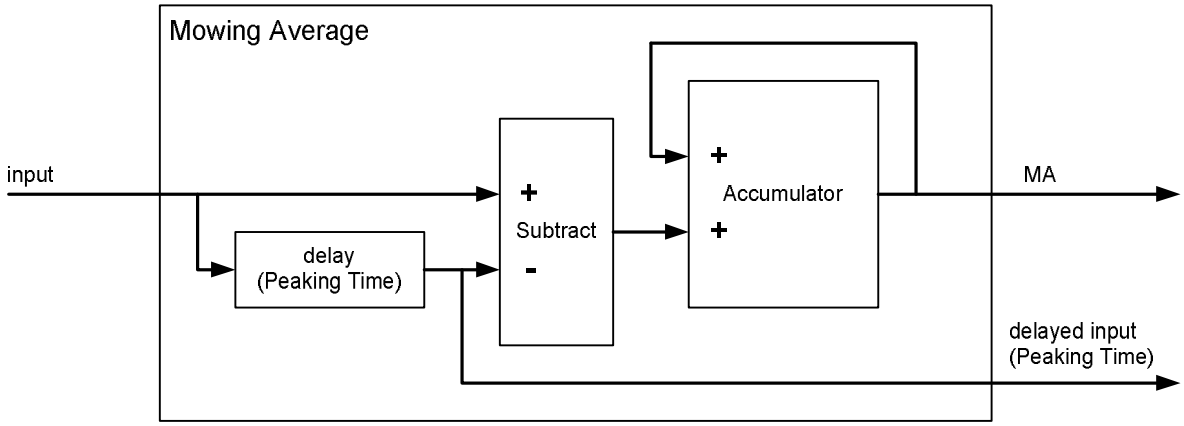
2.2.2.1 Block diagram of the MAWD unit



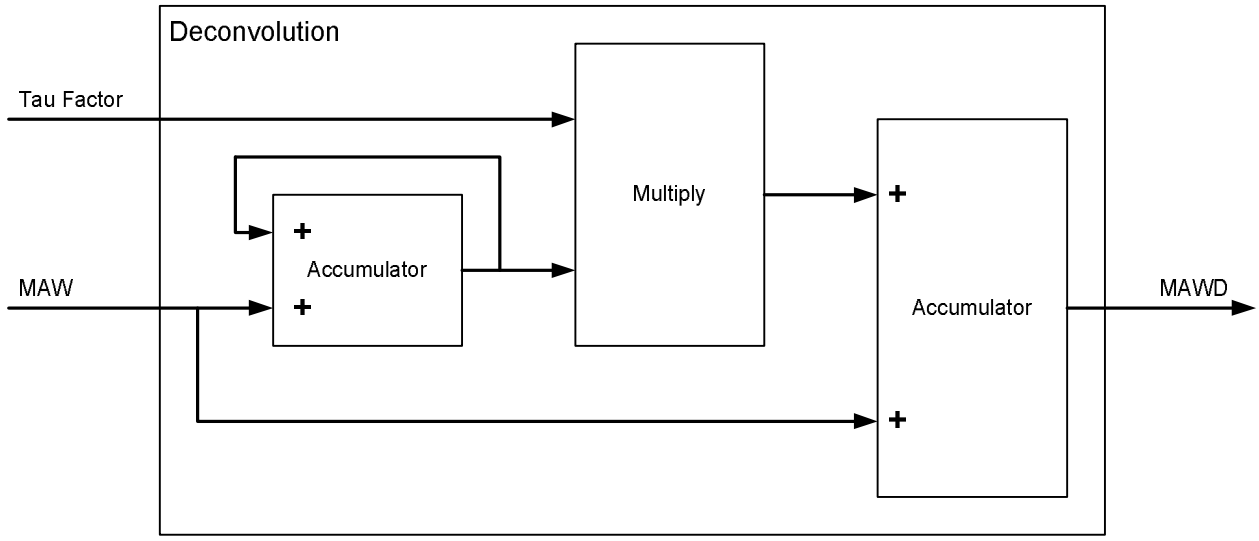
2.2.2.2 Signal diagram of the MAWD unit



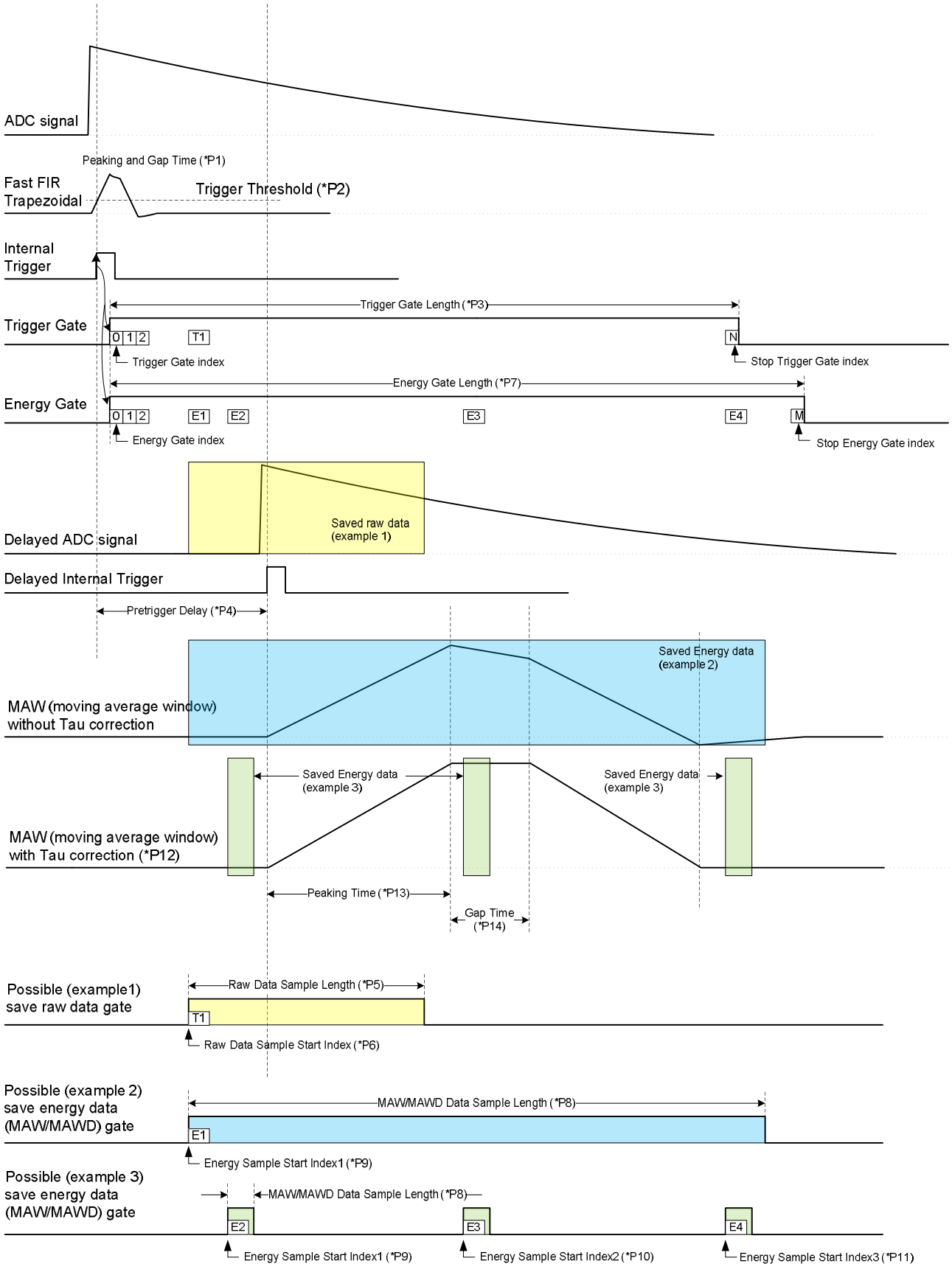
2.2.2.3 Block diagram of the moving average (MA) unit



2.2.2.4 Block diagram of the Deconvolution (Tau correction) unit



2.2.3 Gamma Logic Parameter ilustration



The Gamma Logic parameters are:

- *P1: Fast FIR Filter (Trigger) parameters: see Trigger Setup ADCx registers
- *P2: Fast FIR Filter (Trigger) Threshold: see Trigger Threshold ADCx registers

- *P3: Trigger Gate Length: see Pretrigger Delay and Trigger Gate Length registers
- *P4: Pretrigger Delay: see Pretrigger Delay and Trigger Gate Length registers

- *P5: Raw Data Sample Length: see Raw Data Buffer Configuration registers
- *P6: Raw Data Sample Start Index: see Raw Data Buffer Configuration registers

- *P7: Energy Gate Length: see Energy Gate Length registers

- *P8: Energy Sample Length: see Energy Sample Length registers
- *P9: Energy Sample Start Index1: see Energy Sample Start Index1 registers
- *P10: Energy Sample Start Index2: see Energy Sample Start Index2 registers
- *P11: Energy Sample Start Index3: see Energy Sample Start Index3 registers

- *P12: Energy Tau correction Factor: see Tau Factor registers

- *P13: Energy Filter Peaking Time: see Energy Setup GP registers
- *P14: Energy Filter GapTime: see Energy Setup GP registers

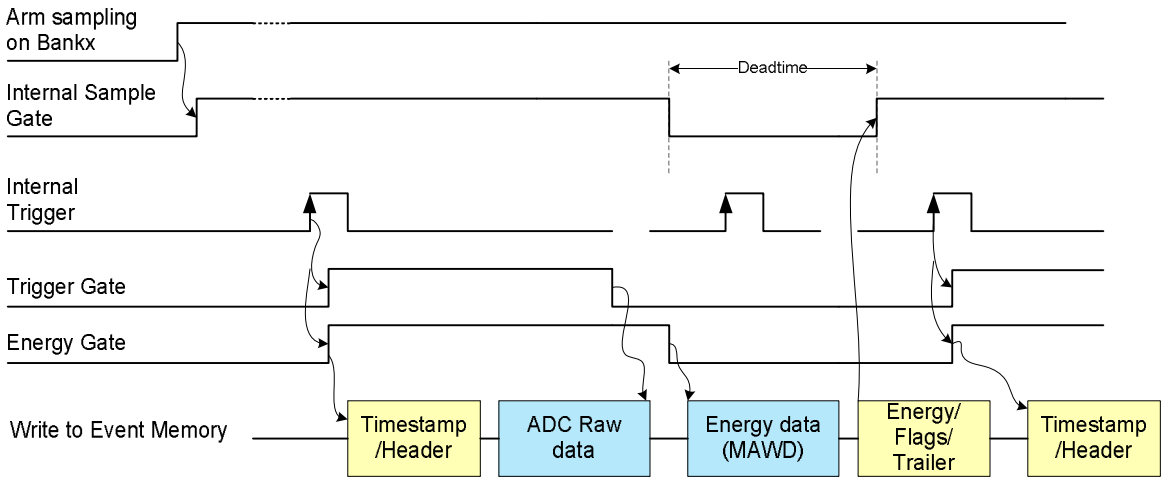
2.2.4 Sample Logic

The sample logic starts with a trigger signal and executes following steps:

1. starts the Trigger and Energy Gate and stores the 48-bit timestamp.
2. writes the 16-bit programmable ADC Header and the stored 48-bit Timestamp into the Event Memory.
3. writes a programmable number of ADC Raw Values into the Event Memory.
4. writes at the end of the Energy Gate a programmable number of Slow FIR Values (MAWD) into the Event Memory.
5. writes the maximum and minimum (first value of Energy Gate) MAWD value into the Event Memory.
6. writes the Fast Filter Information register into the Event Memory.
7. writes a Trailer into the Event Memory.

2.2.5 Deadtime

Event storage induces Deadtime.



Deadtime table

Raw Data Sample Length	Energy Sample Length	Deadtime (approximate)
0	0	150 ns
100	0	560 ns
500	0	2 us
1000	0	3.9 us
0	12	1 us
100	12	1.2 us
500	12	3 us
1000	12	7 us
0	256	5.4 us
100	256	5.6 us
500	256	8.0 us
1000	256	10.6 us
0	512	9.6 ns
100	512	9.8 us
500	512	10.8 us
1000	512	14 us

2.2.6 Fast FIR Filter Trigger Generation

A trapezoidal FIR filter is implemented for each ADC Channel to generate a trigger signal. This Trigger Signal can be used to trigger the sample logic immediately (asynchronous sample mode) or it can be routed to SIS3302 Lemo Output.
This Trigger Signal will be also used to store the "Fast Filter trigger" information.

Features for each ADC channel:
- Programmable Peaking Time (max. 16 Clocks)
- Programmable Gap Time (max. 16 Clocks)
- Programmable Trigger pulse out length (max. 255 Clocks)
- Programmable Trigger Threshold
- Programmable Trigger Mode (GT,Disable)
- Programmable Trigger OUT (Enable,Disable)

see Trigger Setup ADCx registers and Trigger Threshold ADCx registers

2.2.7 Slow FIR Filter Energy sample logic

A trapezoidal FIR filter is implemented for each ADC Channel to generate a "moving window average" stream (MWA) .
A decimation logic (average) is also implemented.

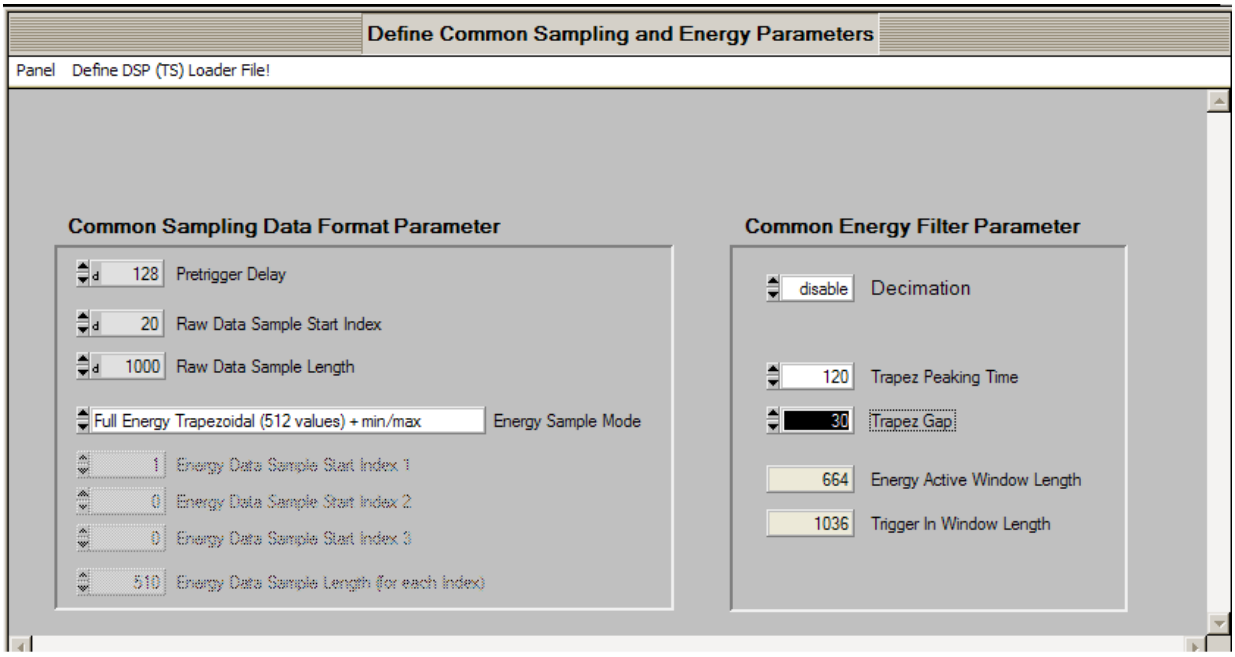
Features of the Slow FIR Filter:
- Programmable decimation (1 / 2 / 4 / 8 Clocks)
- Programmable Peaking Time (max. 255 Clocks)
- Programmable Gap Time (max. 255 Clocks)

Five registers are implemented to control the sampling of the MWD.
- Energy Gate Length
- Energy Sample Length register
- Energy Sample Start Index1 register
- Energy Sample Start Index2 register
- Energy Sample Start Index3 register

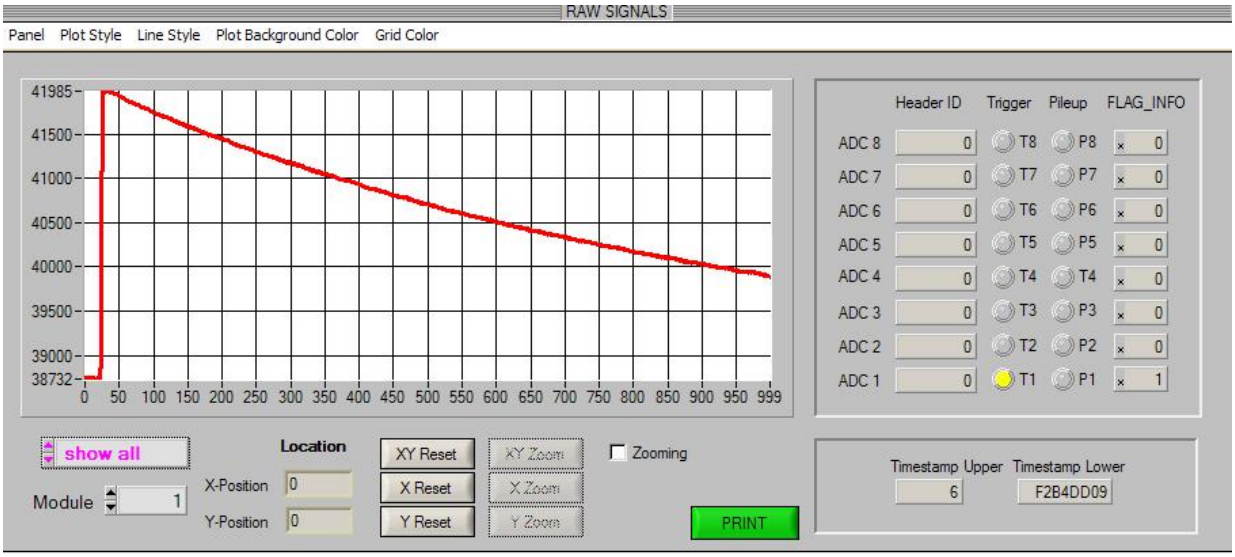
The "Slow FIR Filter Energy sample" logic starts with the Energy Gate and executes following steps:
1. Clears an internal Energy Index counter and starts this Index counter.
   The logic is busy until the Index counter reaches the value of the Energy Gate Length register (busy with the Energy Gate).
2. Compares the Index counter with the Energy Sample Start Index x registers.
   If the result is equal the logic writes N (Energy Sample Length register) values into the Energy Buffer.

2.2.8 CVI plots

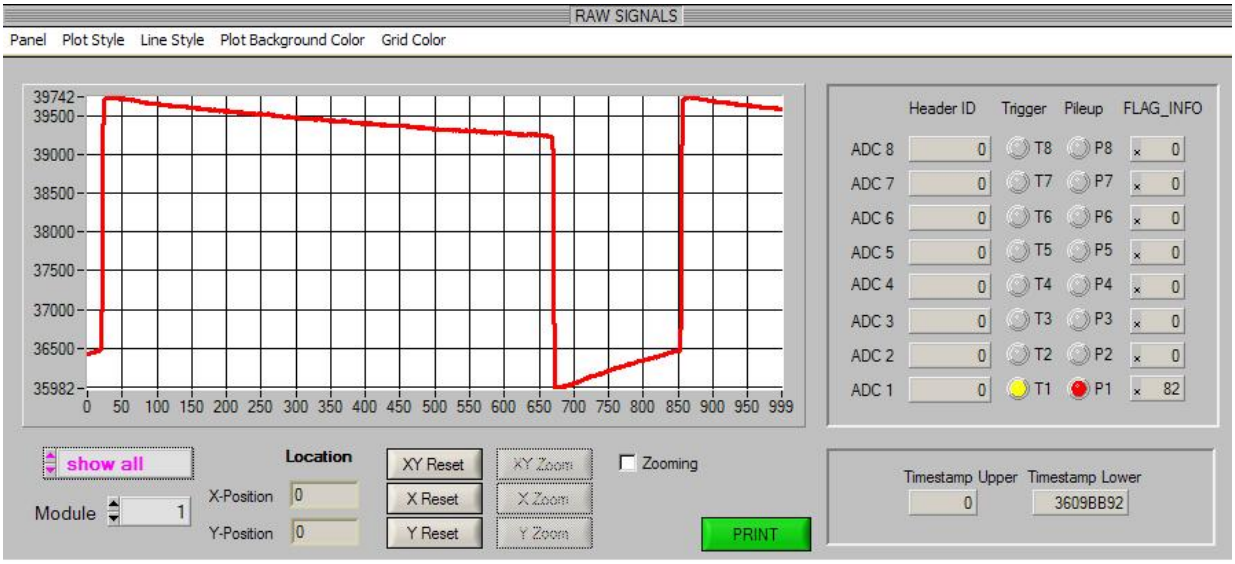
2.2.8.1 Sampling and Energy Filter Parameter screenshot



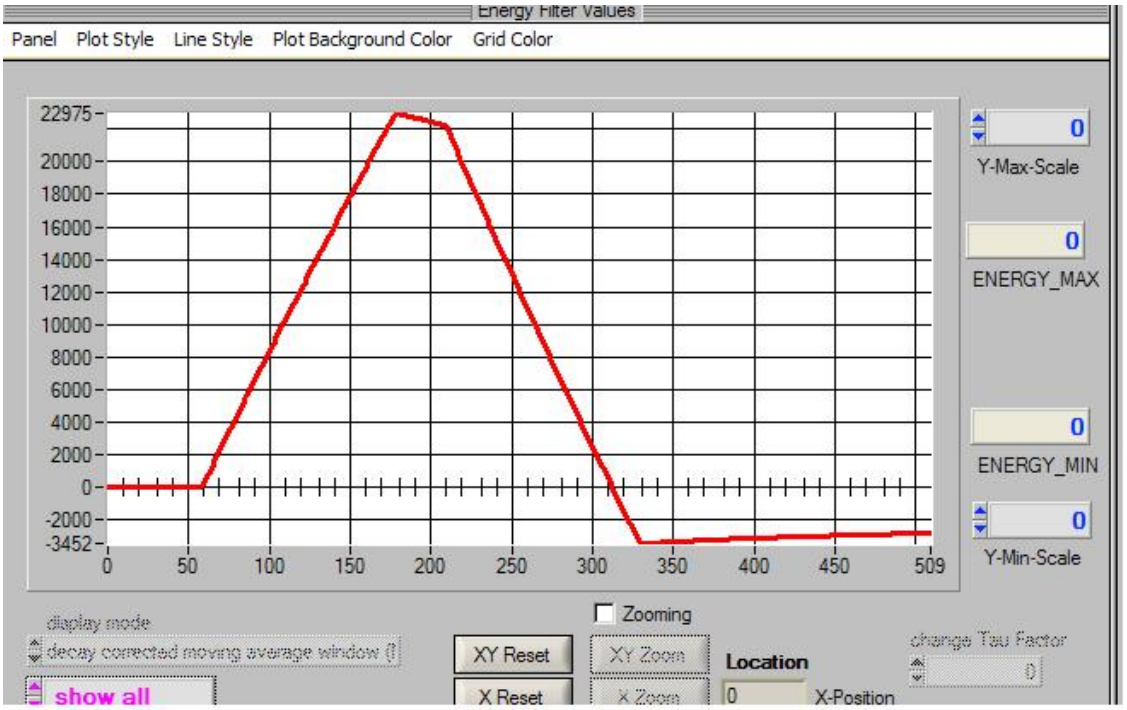
2.2.8.2 Raw signal screenshot



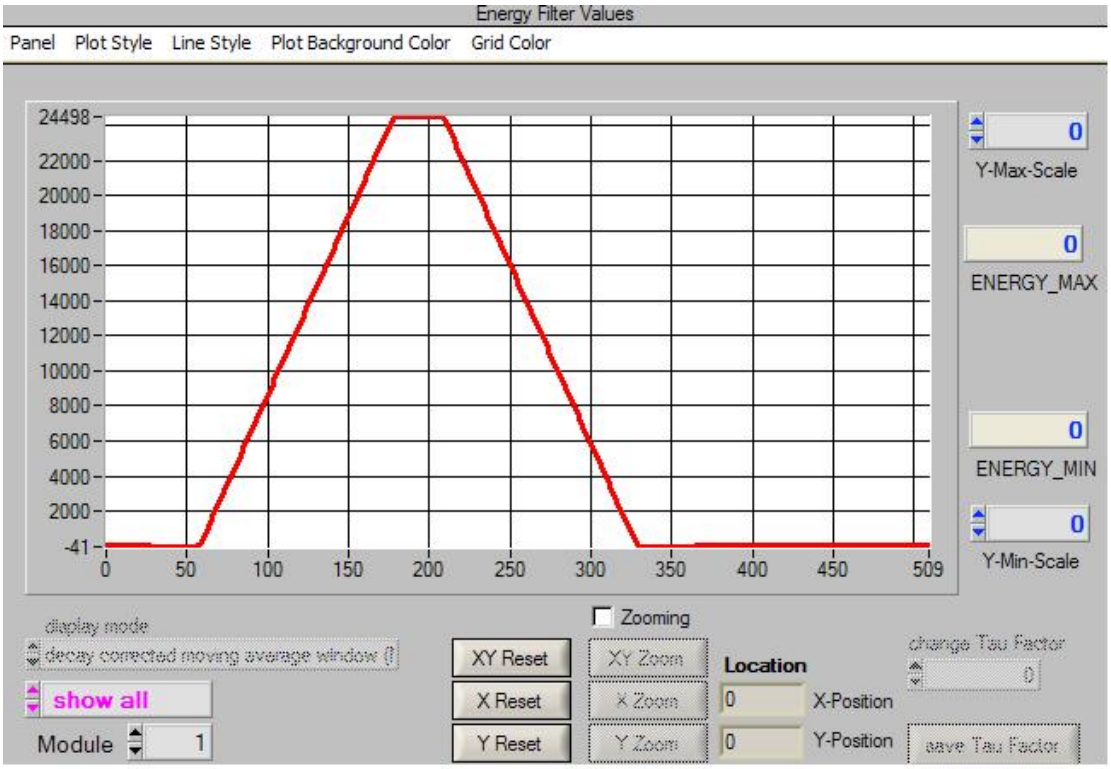
2.2.8.3 Raw signal with Pileup screenshot



2.2.8.4 Energy Filter data without Tau correction (MAW) screenshot



2.2.8.5 Energy Filter data with Tau correction (MAWD) screenshot



3 VME Addressing

As the SIS3302 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 Bit 27=0	
x			SW1				SW2=8...F 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 SIS3302 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 0x120x firmware specific SIS3302 resources and their locations are listed with function in the table below. The header file (sis3302_v1201.h or equivalent) provides the define statements.

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)
0x00000030	4	-	R/W	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
0x00000060			R/W	XILINX JTAG_TEST/JTAG_DATA_IN
0x00000064			W	XILINX JTAG_CONTROL
0x00000400	4	-	KA W	Key address Reset
0x00000410	4	-	KA W	Key address Sample Logic Reset
0x00000414	4	-	KA W	Key address Disarm Sample Logic
0x00000418	4	-	KA W	Key address Trigger
0x0000041C	4	-	KA W	Key address Timestamp Clear
0x00000420	4		KA W	Key address Disarm Sample Logic and Arm sampling on Bank 1
0x00000424	4		KA W	Key address Disarm Sample Logic and Arm sampling on Bank 2
0x00000428	4	-	KA W	Key address Reset DDR2 Memory Logic

Event information all ADC groups				
0x01000000	4	-	W only	Event configuration (all ADCs)
0x01000004	4		W only	End Address Threshold (all ADCs)
0x01000008	4	-	W only	Pretrigger Delay and Trigger Gate Length (all ADCs)
0x0100000C	4	-	W only	Raw Data Buffer Configuration (all ADCs)
0x01000040	4		W only	Energy Setup GP (all ADCs)
0x01000044	4		W only	Energy Gate Length (all ADCs)
0x01000048	4		W only	Energy Sample Length (all ADCs)
0x0100004C	4		W only	Energy Sample Start Index1 (all ADCs)
0x01000050	4		W only	Energy Sample Start Index2 (all ADCs)
0x01000054	4		W only	Energy Sample Start Index3 (all ADCs)
0x01000058	4		W only	Energy Tau Factor ADC1/3/5/7
0x0100005C	4		W only	Energy Tau Factor ADC2/4/6/8

Event information ADC group 1				
0x02000000	4	-	R/W	Event configuration (ADC1, ADC2)
0x02000004	4		R/W	End Address Threshold (ADC1, ADC2)
0x02000008	4	-	R/W	Pretrigger Delay and Trigger Gate Length (ADC1, ADC2)
0x0200000C	4	-	R/W	Raw Data Buffer Configuration (ADC1, ADC2)
0x02000010	4	-	R	Actual Sample address ADC1
0x02000014	4	-	R	Actual Next Sample address ADC2
0x02000018	4		R	Previous Bank Sample address ADC1
0x0200001C	4		R	Previous Bank Sample address ADC2
0x02000020	4	-	R	Actual Sample Value (ADC1, ADC2)
0x02000024	4	-	R	internal Test
0x02000028	4	-	R	DDR2 Memory Logic Verification (ADC1, ADC2)
0x02000030	4		R/W	Trigger Setup ADC1
0x02000034	4		R/W	Trigger Threshold ADC1
0x02000038	4		R/W	Trigger Setup ADC2
0x0200003C	4		R/W	Trigger Threshold ADC2
0x02000040	4		R/W	Energy Setup GP (ADC1, ADC2)
0x02000044	4		R/W	Energy Gate Length (ADC1, ADC2)
0x02000048	4		R/W	Energy Sample Length (ADC1, ADC2)
0x0200004C	4		R/W	Energy Sample Start Index1 (ADC1, ADC2)
0x02000050	4		R/W	Energy Sample Start Index2 (ADC1, ADC2)
0x02000054	4		R/W	Energy Sample Start Index3 (ADC1, ADC2)
0x02000058	4		R/W	Energy Tau Factor ADC1
0x0200005C	4		R/W	Energy Tau Factor ADC2

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

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

Event information ADC group 4				
0x03800000	4	-	R/W	Event configuration (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

4 Register/Resource Description

The 0x12xx firmware related registers are described in this section. The define statements were taken from the sis3302_v1201.h header file. Examples refer to the C code which is underlying the sis3150_3302_gamma CVI project.

4.1 Control/Status register

```
#define SIS3302_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

```
#define SIS3302_MODID              0x4      /* read only; D32 */
```

This register reflects the module identification of the SIS3302 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	0
22	Module Id. Bit 6	
21	Module Id. Bit 5	
20	Module Id. Bit 4	
19	Module Id. Bit 3	2
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
0x12	Gamma

4.3 Interrupt configuration register

```
#define SIS3302_IRQ_CONFIG      0x8      /* read/write; D32 */
```

This read/write register controls the VME interrupt behaviour of the SIS3302 ADC. Four interrupt sources are foreseen, for the time being two of them are associated with an interrupt condition, two condition are 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

4.4 Interrupt control register

```
#define SIS3302_IRQ_CONTROL          0xC      /* read/write; D32 */
```

This register controls the VME interrupt behaviour of the SIS3302 ADC. Four interrupt sources are foreseen, for the time being two of them are associated with an interrupt condition, two condition are reserved for future use.

Bit	Function (w)	(r)	Default
31	Update IRQ Pulse	Status IRQ source 7 (reserved)	0
30	Unused	Status IRQ source 6 (reserved)	0
29	Unused	Status IRQ source 5 (reserved)	0
28	Unused	Status IRQ source 4 (reserved)	0
27	Unused	Status IRQ source 3 (reserved)	0
26	Unused	Status IRQ source 2 (reserved)	0
25	Unused	Status IRQ source 1 (End Address Threshold Flag; level sensitive)	0
24	Unused	Status IRQ source 0 (End Address Threshold Flag; edge sensitive)	0
23	Disable/Clear IRQ source 7	Status flag source 7	0
22	Disable/Clear IRQ source 6	Status flag source 6	0
21	Disable/Clear IRQ source 5	Status flag source 5	0
20	Disable/Clear IRQ source 4	Status flag source 4	0
19	Disable/Clear IRQ source 3	Status flag source 3	0
18	Disable/Clear IRQ source 2	Status flag source 2	0
17	Disable/Clear IRQ source 1	Status flag source 1	0
16	Disable/Clear IRQ source 0	Status flag source 0	0
15	Unused	Status VME IRQ	0
14	Unused	Status internal IRQ	0
13	Unused	0	0
12	Unused	0	0
11	Unused	0	0
10	Unused	0	0
9	Unused	0	0
8	Unused	0	0
7	Enable IRQ source 7	Status enable source 7 (read as 1 if enabled, 0 if disabled)	0
6	Enable IRQ source 6	Status enable source 6 (read as 1 if enabled, 0 if disabled)	0
5	Enable IRQ source 5	Status enable source 5 (read as 1 if enabled, 0 if disabled)	0
4	Enable IRQ source 4	Status enable source 4 (read as 1 if enabled, 0 if disabled)	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: reached Address Threshold (level sensitive)
IRQ source 0: reached Address Threshold (edge sensitive)

4.5 *Acquisition control register*

```
#define SIS3302_ACQUISTION_CONTROL    0x10      /* read/write; D32 */
#define SIS3302_ACQUISITION_CONTROL   0x10      /* read/write; D32 */
```

Note: missing I in ACQUISITION in older header files

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	Disable front panel LEMO Timestamp Clear (*)	0
24	Disable front panel LEMO Start (external Trigger) (*)	0
23	Clear reserved 7 (*)	0
22	Disable internal trigger (*)	0
21	Clear reserved 5 (*)	0
20	Clear reserved 4 (*)	0
19	Clear reserved 3 (*)	Status of End Address Threshold Flag
18	Clear reserved 2 (*)	Status of ADC-Sample-Logic Busy (Armed)
17	Clear reserved 1 (*)	Status of ADC-Sample-Logic Armed Bank2
16	Clear reserved 0 (*)	Status of ADC-Sample-Logic Armed Bank1
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	Enable front panel LEMO Timestamp Clear	Status LEMO Timestamp Clear
8	Enable front panel LEMO Start (external Trigger)	Status LEMO Start (external Trigger)
7	Set reserved 7	Status reserved 7
6	Enable internal trigger	Status internal trigger
5	Set reserved 5	Status reserved 5
4	Set reserved 4	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	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 100 MHz
0	0	1	internal 50 MHz
0	1	0	internal 25 MHz
0	1	1	internal 10 MHz
1	0	0	internal 1 MHz
1	0	1	internal 100 MHz
1	1	0	external clock (LEMO front panel) ; min. 1 MHz
1	1	1	P2-Clock (not yet implemented)

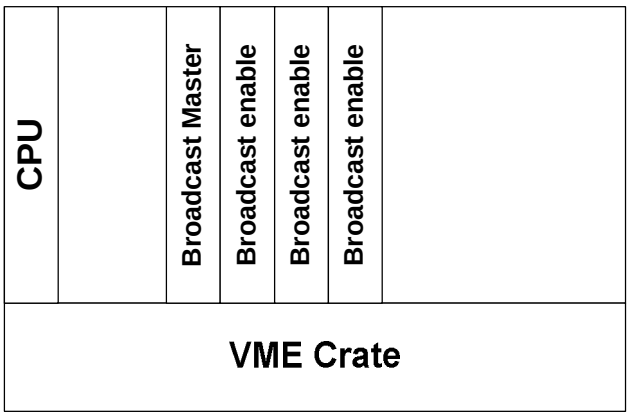
4.6 Broadcast setup register

```
#define SIS3302_CBLT_BROADCAST_SETUP      0x30      /* read/write; D32 */
```

This read/write register defines, whether the SIS3302 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	Broadcast address bit 31
30	Broadcast address bit 30
29	Broadcast address bit 29
28	Broadcast address bit 28
27	Broadcast address bit 27
26	Broadcast address bit 26
25	Broadcast address bit 25
24	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

Broadcast functionality is implemented for all Key address cycles.
Modules which are supposed to participate in a broadcast have to get the same broadcast address. The broadcast address is defined by the upper 8 bits of the broadcast setup register. One module has to be configured as broadcast master, the enable broadcast bit has to be set for the others as illustrated below.



Broadcast setup example (broadcast address 0x34000000):

Module	Broadcast Setup Register	Comment
1	0x34000020	Broadcast Master
2	0x34000010	Broadcast enable
3	0x34000010	Broadcast enable
4	0x34000010	Broadcast enable

All 4 modules will participate in a key reset (A32/D32 write) to address 0x34000400.

Note: Do not use a broadcast address that is an existing VME address of a VME card in the crate.

4.7 ADC Memory Page register

```
#define SIS3302_ADC_MEMORY_PAGE_REGISTER 0x34      /* read/write; D32 */
```

The SIS3302 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: dual 1/2 memory buffer acquisition scheme
(see sis3150_gamma_running.c)

```
data = 0x0 ; //Bank2 is armed and Bank1 (page 0) has to be readout
if (bank1_armed_flag == 1) { // Bank1 is armed and Bank2 (page 4) has to be readout
    data = 0x4 ;
}
addr = gl_uint_ModAddrRun[module_index] + SIS3302_ADC_MEMORY_PAGE_REGISTER ;
if ((error = sub_vme_A32D32_write(addr,data )) != 0) {
    sisVME_ErrorHandling(error, gl_uint_ModAddrRun[module_index],
                         "sub_vme_A32D32_write");
    return -1;
}
```

4.8 DAC Control Registers

```
#define SIS3302_DAC_CONTROL_STATUS        0x50      /* read/write; D32 */
```

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 sis3302_adc_test1.c (CVI directory)

Example routine:

```
int sis3302_write_dac_offset( unsigned int module_addr,
                              unsigned int *offset_value_array)
```

4.8.1 DAC Control/Status register (read/write)

```
#define SIS3302_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
...		...
8	None	0
7	None	0
6	DAC selection Bit 2	status of DAC selection Bit 2
5	DAC selection Bit 1	status of DAC selection Bit 1
4	DAC selection Bit 0	status of DAC selection Bit 0
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 of selected DAC
1	0	Load selected DAC
1	1	Clear all DACs

A “Clear DAC” command sets the value of all DACs to analog ground

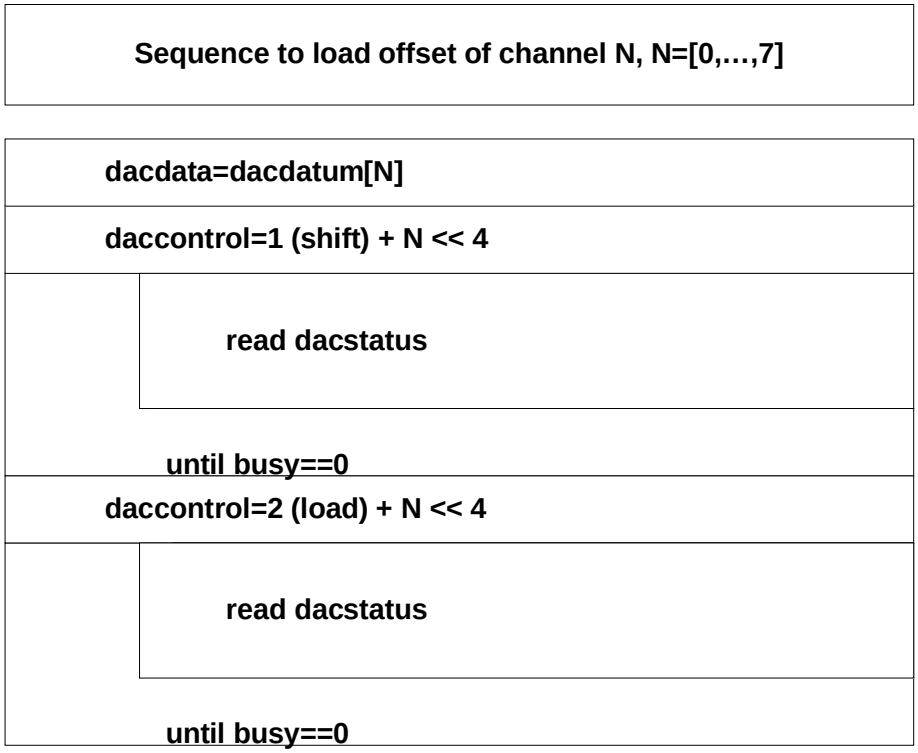
4.8.2 DAC Data register

```
#define SIS3302_DAC_DATA            0x54      /* read/write; D32 */
```

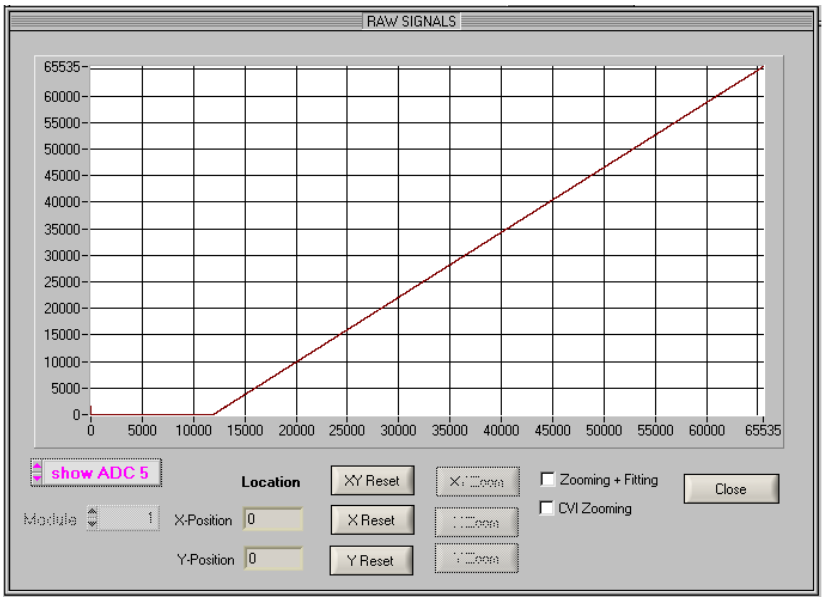
Bit	Write Function	Read Function
31	none	DAC Input Register Bit 15 (from DAC)
..	..	..
..	..	..
16	none	DAC Input Register Bit 0
15	DAC Output Register Bit 15	DAC Output Register Bit 15
..	..	0
..	..	0
0	DAC Output Register Bit 0	DAC Output Register Bit 0

4.8.3 DAC load sequence

The sequence to load the DAC of a single channel is shown below.
The example routine `sis3302_write_dac_offset` loads the 8 DACs of a SIS3302 module at base address module_ offset in a loop.



Find below a DAC scan that was acquired with the DAC test function of the SIS3302 ADC Labwindows application. The SIS3302 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 on the ADC values.

4.9 Key address general reset (0x400)

```
#define SIS3302_KEY_RESET                0x400   /* write only; D32 */
```

A write with arbitrary data to this register (key address) resets the SIS3302 to it's power up state.

4.10 Key address Sample Logic reset (0x410)

```
#define SIS3302_KEY_SAMPLE_LOGIC_RESET   0x410   /* write only; D32 */
```

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

4.11 Key address VME Disarm Sample logic (0x414)

```
#define SIS3302_KEY_DISARM               0x414   /* write only; D32 */
```

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

4.12 Key address VME Trigger (0x418)

```
#define SIS3302_KEY_Trigger              0x418   /* write only; D32 */
```

A write with arbitrary data to this register (key address) will trigger the sampling logic if "external trigger" is enabled (see Event configuration registers).

4.13 Key address Timestamp Clear (0x41C)

```
#define SIS3302_KEY_TIMESTAMP_CLEAR      0x41C  /* write only; D32 */
```

A write with arbitrary data to this register (key address) clears the 48-bit Timestamp counter.

4.14 Key address Disarm Sample Logic and Arm sampling on Bank 1 (0x420)

```
#define SIS3302_KEY_DISARM_AND_ARM_BANK1      0x420  /* write only; D32 */
```

Disarms sampling on current bank and arms bank 1. Used in dual bank acquisition as illustrated in the example in subsection 4.15 below.

4.15 Key address Disarm Sample Logic and Arm sampling on Bank 2 (0x424)

```
#define SIS3302_KEY_DISARM_AND_ARM_BANK2        0x424  /* write only; D32 */
```

Disarms sampling on current bank and arms bank 2. Used in dual bank acquisition as illustrated in the example below.

Example (from sis3150_gamma_running.c):

```
if (bank1_armed_flag == 1) {
    addr = gl_uint_ModAddrRun[0] + SIS3302_KEY_DISARM_AND_ARM_BANK2 ;
    bank1_armed_flag = 0; // bank 2 is armed
}
else {
    addr = gl_uint_ModAddrRun[0] + SIS3302_KEY_DISARM_AND_ARM_BANK1 ;
    bank1_armed_flag = 1; // bank 2 is armed
}
if ((error = sub_vme_A32D32_write(addr,0x0 )) != 0) {
    sisVME_ErrorHandling (error, addr, "sub_vme_A32D32_write")   ;
}
```

4.16 Event configuration registers

```
#define SIS3302_EVENT_CONFIG_ALL_ADC     0x01000000  /* write only;D32 */

#define SIS3302_EVENT_CONFIG_ADC12       0x02000000  /* read/write;D32 */
#define SIS3302_EVENT_CONFIG_ADC34       0x02800000  /* read/write;D32 */
#define SIS3302_EVENT_CONFIG_ADC56       0x03000000  /* read/write;D32 */
#define SIS3302_EVENT_CONFIG_ADC78       0x03800000  /* read/write;D32 */
```

This register is implemented for each channel group. The `SIS3302_EVENT_CONFIG_ALL_ADC` register can be used to write the same setting to the registers of all channel groups simultaneously.

Bit	Function
31	HEADER_ADC_ID bit 15
30	..
..	..
21	..
20	HEADER_ADC_ID bit 4
19	HEADER_ADC_ID bit 3
18	Channel Group ID Bit 1 (read only), used as HEADER_ADC_ID bit 2
17	Channel Group ID Bit 0 (read only), used as HEADER_ADC_ID bit 1
16	unused
15	reserved
14	reserved
13	reserved
12	reserved
11	ADC 2 external trigger enable (synchronous mode)
10	ADC 2 internal trigger enable (asynchronous mode)
9	reserved
8	ADC 2 input invert bit
7	reserved
6	reserved
5	reserved
4	reserved
3	ADC 1 external trigger enable (synchronous mode)
2	ADC 1 internal trigger enable (asynchronous mode)
1	unused
0	ADC 1 input invert bit

ADCx input invert bit = 0: use for positive signals
ADCx input invert bit = 1: use for negative signals

ADCx Trigger enable bits

external trigger enable	internal trigger enable	Function
0	0	No triggering
0	1	internal channel based trigger (asynchronous mode)
1	0	external trigger (synchronous mode)
1	1	Or of internal channel based trigger and external trigger

4.17 End Address Threshold registers

```
#define SIS3302_END_ADDRESS_THRESHOLD_ALL_ADC        0x01000004

#define SIS3302_END_ADDRESS_THRESHOLD_ADC12          0x02000004
#define SIS3302_END_ADDRESS_THRESHOLD_ADC34          0x02800004
#define SIS3302_END_ADDRESS_THRESHOLD_ADC56          0x03000004
#define SIS3302_END_ADDRESS_THRESHOLD_ADC78          0x03800004
```

These registers define the “End Address Threshold” values for the ADC channel groups.

The value of the Actual Next Sample address counter will be compared with value of the End Address Threshold register.
The value is given in samples (i.e. number of 16-bit words)

Bit	
31	unused, read as 0
...	
24	unused, read as 0
23	Sample Start Address Register Bit 23
..	
2	Sample Start Address Register Bit 2
1	unused, read as 0
0	unused, read as 0

The power up default value is 0

4.18 Pretrigger Delay and Trigger Gate Length registers

```
#define SIS3302_PRETRIGGER_DELAY_TRIGGERGATE_LENGTH_ALL_ADC       0x01000008

#define SIS3302_PRETRIGGER_DELAY_TRIGGERGATE_LENGTH_ADC12         0x02000008
#define SIS3302_PRETRIGGER_DELAY_TRIGGERGATE_LENGTH_ADC34         0x02800008
#define SIS3302_PRETRIGGER_DELAY_TRIGGERGATE_LENGTH_ADC56         0x03000008
#define SIS3302_PRETRIGGER_DELAY_TRIGGERGATE_LENGTH_ADC78         0x03800008
```

This register defines the length of the Trigger Gate (1 to 1024) and the Pretrigger Delay (0 to 1023) .

D31:26	D25:16	D15:12	D11:0
0	Pretrigger Delay	0	Trigger Gate Length (-1)

The power up default value is 0

Example:
Desired Trigger Gate Length of 1024 clocks and a Pretrigger Delay of 256 clocks (samples) -> set the register to 0x010003FF

4.19 Raw Data Buffer Configuration registers

```
#define SIS3302_RAW_DATA_BUFFER_CONFIG_ALL_ADC          0x0100000C

#define SIS3302_RAW_DATA_BUFFER_CONFIG_ADC12            0x0200000C
#define SIS3302_RAW_DATA_BUFFER_CONFIG_ADC34            0x0280000C
#define SIS3302_RAW_DATA_BUFFER_CONFIG_ADC56            0x0300000C
#define SIS3302_RAW_DATA_BUFFER_CONFIG_ADC78            0x0380000C
```

This register is used to configure the number of samples of raw data to be acquired and to define the number of pre trigger samples (combination of “Raw Data Sample Start Index” and “Pretrigger Delay”).

While the Trigger Gate is active the logic compares the value of the Trigger Gate Index (an internal counter, which is cleared at the beginning of the event and incremented by each clock)
with the value of the “Raw Data Sample Start Index” register. If the result of the compare is equal then the logic writes N values (“Raw Data Sample Length”) into the Raw Data Buffer.

Both values are 12-bit deep. The number of samples has to be quad sample aligned and the number of pre trigger samples has to b even.

The number of “Raw Data Sample Length” is limited to 1024 samples!

Note: The register is implemented on the FPGA group level also.

Bit	Function
31	Unused
…	
28	Unused
27	Bit 11 of Raw Data Sample Length
…	…
18	Bit 2 of Raw Data Sample Length
17	0 (quad sample aligned values only)
16	0 (quad sample aligned values only)
15	Unused
…	…
12	Unused
11	Bit 11 of Raw Data Sample Start Index
…	…
1	Bit 1 of Raw Data Sample Start Index
0	0 (even values only)

Example:

```
data =     (gl_uint_RawPreSampleStart_index & 0xffe)
              + ((gl_uint_RawSampleLength & 0xffc) << 16) ;
addr = module_addr + SIS3302_RAW_DATA_BUFFER_CONFIG_ALL_ADC ;
if ((error = sub_vme_A32D32_write(addr,data )) != 0) {
       sisVME_ErrorHandling (error, addr, "sub_vme_A32D32_write")   ;
}
```

4.20 ADC1-8 Next Sample address register

```
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC1          0x02000010
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC2          0x02000014
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC3          0x02800010
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC4          0x02800014
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC5          0x03000010
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC6          0x03000014
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC7          0x03800010
#define SIS3302_ACTUAL_NEXT_SAMPLE_ADDRESS_ADC8          0x03800014
```

These 8 read only registers hold the current next sampling address for the given channel.
Note: the Next Sample Address points to 16-bit words (samples).

Bit	Function
31	unused, read as 0
...	
25	unused, read as 0
24	Next Sample Address Bit 24 (Bank flag)
..	
2	Next Sample Address Bit 2
1	Next Sample Address Bit 1*
0	Next Sample Address Bit 0*

* Sample address bits 1 and 0 are always “0”.
Data are stored to memory in packets of 4 consecutive samples by the sample logic.

4.21 ADC1-8 Previous Bank Sample address register

```
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC1     0x02000018
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC2     0x0200001C
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC3     0x02800018
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC4     0x0280001C
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC5     0x03000018
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC6     0x0300001C
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC7     0x03800018
#define SIS3302_PREVIOUS_BANK_SAMPLE_ADDRESS_ADC8     0x0380001C
```

These 8 read only registers hold the stored next sampling address of the previous bank. It is the stop address + 1;
Note: the Next Sample Address points to 16-bit words (samples).

Bit	Function
31	unused, read as 0
...	
25	unused, read as 0
24	Next Sample Address Bit 24 (Bank flag)
..	
2	Next Sample Address Bit 2
1	Next Sample Address Bit 1*
0	Next Sample Address Bit 0*

4.22 Actual Sample registers

```
#define SIS3302_ACTUAL_SAMPLE_VALUE_ADC12          0x02000020
#define SIS3302_ACTUAL_SAMPLE_VALUE_ADC34          0x02800020
#define SIS3302_ACTUAL_SAMPLE_VALUE_ADC56          0x03000020
#define SIS3302_ACTUAL_SAMPLE_VALUE_ADC56          0x03000020
```

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
D31:16	D15:0
16-bit data	16-bit data

4.23 Trigger Setup registers

```
#define SIS3302_TRIGGER_SETUP_ADC1          0x02000030
#define SIS3302_TRIGGER_SETUP_ADC2          0x02000038
#define SIS3302_TRIGGER_SETUP_ADC3          0x02800030
#define SIS3302_TRIGGER_SETUP_ADC4          0x02800038
#define SIS3302_TRIGGER_SETUP_ADC5          0x03000030
#define SIS3302_TRIGGER_SETUP_ADC6          0x03000038
#define SIS3302_TRIGGER_SETUP_ADC7          0x03800030
#define SIS3302_TRIGGER_SETUP_ADC8          0x03800038
```

These read/write registers hold the Peaking and Gap Time of the trapezoidal FIR filter.
(Flat Time = Gap 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	P : Peaking time $\sum_{i=x}^{x+P} Si$
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

Note: ADC raw data are shifted 4 bits to the right for trigger formation

4.24 Trigger Threshold registers

```
#define SIS3302_TRIGGER_THRESHOLD_ADC1          0x02000034
#define SIS3302_TRIGGER_THRESHOLD_ADC2          0x0200003C
#define SIS3302_TRIGGER_THRESHOLD_ADC3          0x02800034
#define SIS3302_TRIGGER_THRESHOLD_ADC4          0x0280003C
#define SIS3302_TRIGGER_THRESHOLD_ADC5          0x03000034
#define SIS3302_TRIGGER_THRESHOLD_ADC6          0x0300003C
#define SIS3302_TRIGGER_THRESHOLD_ADC7          0x03800034
#define SIS3302_TRIGGER_THRESHOLD_ADC8          0x0380003C
```

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

Bit	31-27	26	25	24	23-17	16-0
Function	none	Disable Trigger Out	Trigger Mode GT	None	None	**Trapezoidal** threshold value

default after Reset: 0x0

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:

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

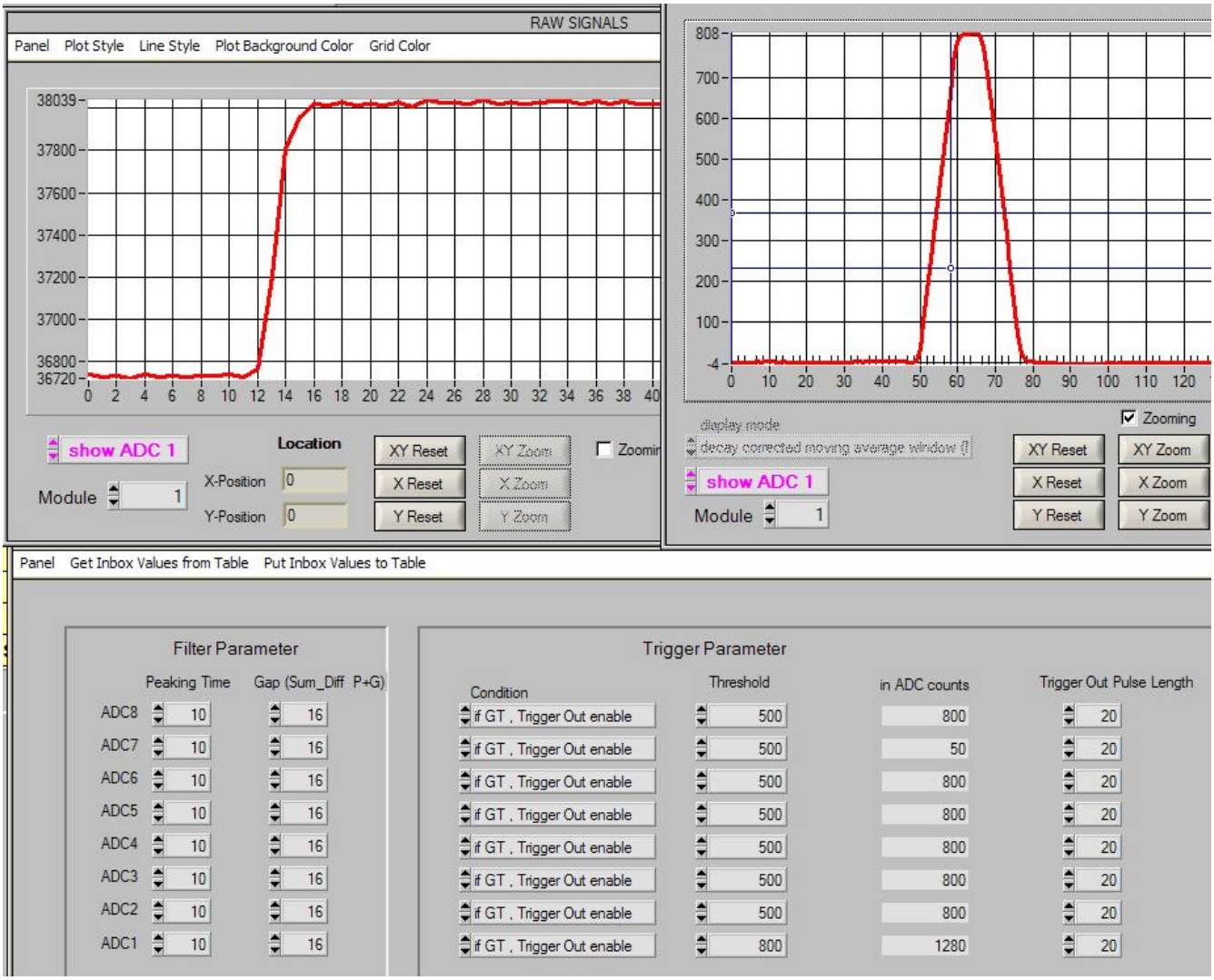
Note: use “ADCx input invert bit” for negative signals (see Event configuration registers)

4.24.1 Trigger example

Screen shot 1 below shows a triggered signal on ADC channel 1 and the resulting FIR. The Peaking Time P is set to 10 and sumG Time is set to 16 (Gap Time = 6).
The trigger condition is set to GT, the trapezoidal trigger threshold is set to 800 (trigger threshold reg = 0x10000 + 800), what results in a decimal threshold of 1280 ADC counts.

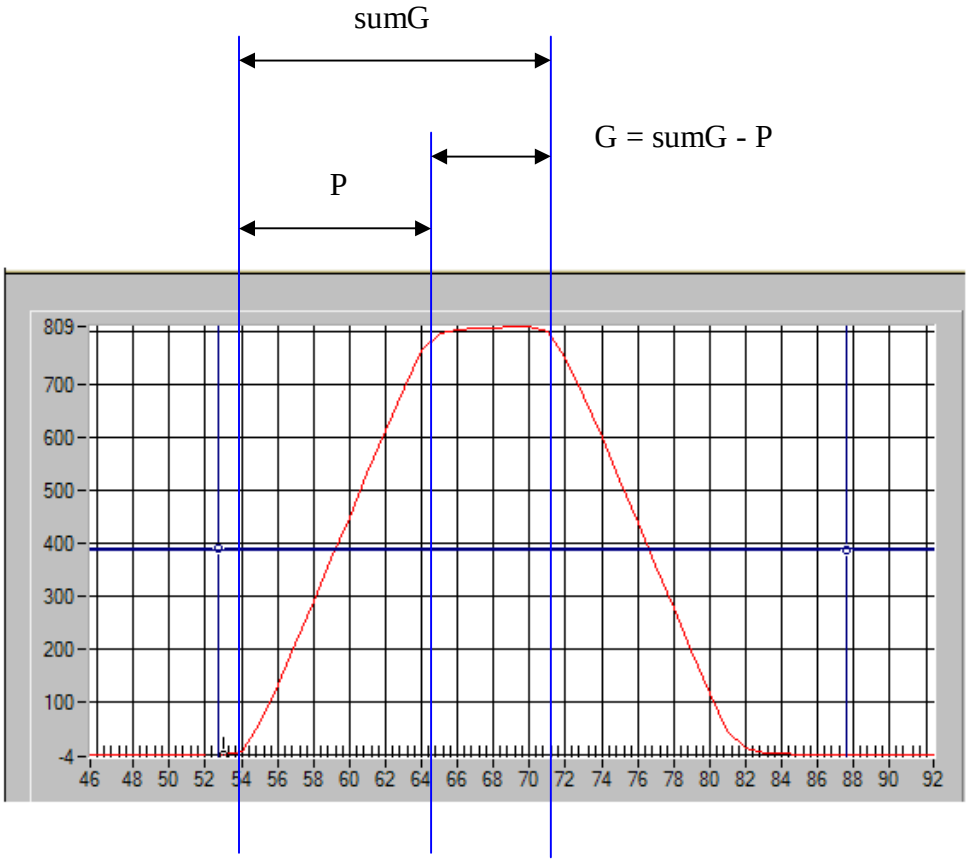
Trigger Threshold = 800 -> ((800 * 16) / 10) = 1280 adc counts
- * 16: because data are shifted right by 4
- / 10: because P = 10 (sum over 10 counts)

ADC Raw data: (38039 – 36270 = 1319 corresponds to 39mV) .



Screen shot 1

Peaking Time = 10
SumG Time = 16
Gap Time = 6



Screen shot 2

4.25 Energy Setup GP registers

```
#define SIS3302_ENERGY_SETUP_GP_ALL_ADC                0x01000040

#define SIS3302_ENERGY_SETUP_GP_ADC12                  0x02000040
#define SIS3302_ENERGY_SETUP_GP_ADC34                  0x02800040
#define SIS3302_ENERGY_SETUP_GP_ADC56                  0x03000040
#define SIS3302_ENERGY_SETUP_GP_ADC78                  0x03800040
```

This read/write register holds the Decimation, Peaking and Gap Time of the trapezoidal FIR Energy filter.

Bit		
31	reserved	
30	reserved	
29	Decimation mode bit 1	
28	Decimation mode bit 0	
27	reserved	
26	reserved	
25	reserved	
24	reserved	
23	reserved	reserved
..	..	
..	..	
16	reserved	
15	**G**ap Time bit 7	Gap time (time between both sums)
..	..	
..	..	
8	**G**ap Time bit 0	
7	**P**eaking Time bit 7	Peaking time P x+P $\sum_{i=x} Si$
..	..	
..	..	
0	**P**eaking Time bit 0	

The power up default value reads 0x 00000000

Decimation Mode bit setting table:

Bit1	Bit0	Decimation (in clocks)
0	0	1 (no decimation)
0	1	2 clocks
1	0	4 clocks
1	1	8 clocks

4.26 Energy Gate Length registers

```
#define SIS3302_ENERGY_GATE_LENGTH_ALL_ADC                0x01000044

#define SIS3302_ENERGY_GATE_LENGTH_ADC12                  0x02000044
#define SIS3302_ENERGY_GATE_LENGTH_ADC34                  0x02800044
#define SIS3302_ENERGY_GATE_LENGTH_ADC56                  0x03000044
#define SIS3302_ENERGY_GATE_LENGTH_ADC78                  0x03800044
```

This 14-bit register (bits 11:0) defines the length of the energy gate and defines test modes (bits 13:12) of the Energy Data . The Energy Gate starts with begin of sampling and stops after
("value" * decimation factor) clocks.

Bits	31-14	13-12	11 :0
Function	0	Test Mode bits[1 :0]	Energy Gate Length

Test Mode bit setting table:

Bit1	Bit0	meaning of ADC Energy Data
0	0	MWD-MA, Trapez (no test mode)
0	1	MW-MA, Trapez
1	0	reserved
1	1	reserved

4.27 Energy Sample registers

This register set (`ENERGY_SAMPLE_LENGTH`, `ENERGY_SAMPLE_START_INDEX1`, `ENERGY_SAMPLE_START_INDEX2` and `ENERGY_SAMPLE_START_INDEX3`) controls the storage of the energy filter values.
While the Energy Gate is active the logic compares the value of the Energy Gate Index (an internal counter, which is cleared at the beginning of the event and incremented by each decimated clock) with the values of the "`ENERGY_SAMPLE_START_INDEX`" registers. If the result of the compare is equal then the logic writes N values ("`ENERGY_SAMPLE_LENGTH`") into the Energy Buffer.
The number of values is limited to 512 values in total !

4.27.1 Energy Sample Length registers

```
#define SIS3302_ENERGY_SAMPLE_LENGTH_ALL_ADC          0x01000048
#define SIS3302_ENERGY_SAMPLE_LENGTH_ADC12            0x02000048
#define SIS3302_ENERGY_SAMPLE_LENGTH_ADC34            0x02800048
#define SIS3302_ENERGY_SAMPLE_LENGTH_ADC56            0x03000048
#define SIS3302_ENERGY_SAMPLE_LENGTH_ADC78            0x03800048
```

Bits	31-11	10 :0
Function	0	Energy Sample Length

4.27.2 Energy Sample Start Index1 registers

```
#define SIS3302_ENERGY_SAMPLE_START_INDEX1_ALL_ADC    0x0100004C
#define SIS3302_ENERGY_SAMPLE_START_INDEX1_ADC12      0x0200004C
#define SIS3302_ENERGY_SAMPLE_START_INDEX1_ADC34      0x0280004C
#define SIS3302_ENERGY_SAMPLE_START_INDEX1_ADC56      0x0300004C
#define SIS3302_ENERGY_SAMPLE_START_INDEX1_ADC78      0x0380004C
```

Bits	31-11	10 :0
Function	0	Energy Sample Start Index1

Energy Sample Start Index1 = 0 : disable Start

4.27.3 Energy Sample Start Index2 registers

```
#define SIS3302_ENERGY_SAMPLE_START_INDEX2_ALL_ADC    0x01000050
#define SIS3302_ENERGY_SAMPLE_START_INDEX2_ADC12      0x02000050
#define SIS3302_ENERGY_SAMPLE_START_INDEX2_ADC34      0x02800050
#define SIS3302_ENERGY_SAMPLE_START_INDEX2_ADC56      0x03000050
#define SIS3302_ENERGY_SAMPLE_START_INDEX2_ADC78      0x03800050
```

Bits	31-11	10 :0
Function	0	Energy Sample Start Index2

Energy Sample Start Index2 = 0 : disable Start

4.27.4 Energy Sample Start Index3 registers

```
#define SIS3302_ENERGY_SAMPLE_START_INDEX3_ALL_ADC    0x01000054
#define SIS3302_ENERGY_SAMPLE_START_INDEX3_ADC12      0x02000054
#define SIS3302_ENERGY_SAMPLE_START_INDEX3_ADC34      0x02800054
#define SIS3302_ENERGY_SAMPLE_START_INDEX3_ADC56      0x03000054
#define SIS3302_ENERGY_SAMPLE_START_INDEX3_ADC78      0x03800054
```

Bits	31-11	10 :0
Function	0	Energy Sample Start Index3

Energy Sample Start Index3 = 0 : disable Start

4.28 Energy Tau Factor registers

```
#define SIS3302_ENERGY_TAU_FACTOR_ADC1357       0x01000058
#define SIS3302_ENERGY_TAU_FACTOR_ADC2468       0x0100005C

#define SIS3302_ENERGY_TAU_FACTOR_ADC1          0x02000058
#define SIS3302_ENERGY_TAU_FACTOR_ADC2          0x0200005C
#define SIS3302_ENERGY_TAU_FACTOR_ADC3          0x02800058
#define SIS3302_ENERGY_TAU_FACTOR_ADC4          0x0280005C

#define SIS3302_ENERGY_TAU_FACTOR_ADC5          0x03000058
#define SIS3302_ENERGY_TAU_FACTOR_ADC6          0x0300005C
#define SIS3302_ENERGY_TAU_FACTOR_ADC7          0x03800058
#define SIS3302_ENERGY_TAU_FACTOR_ADC8          0x0380005C
```

Those registers hold the 7-bit wide Tau factor for the corresponding ADC .
They are implemented on the FPGA group level also what allows you to run all 8 channels on the board with individual Tau settings.

Bit	Function
31	unused
…	…
7	unused
6	Bit 6 of Tau factor
…	…
0	Bit 0 of Tau factor

The decay time depends on the Tau factor, on the decimation mode and on the sample clock :

```
switch (gl_uint_SIS3302ClockModeConf) {
      case  0: // intern 100 MHz
            *sample_clock = 100000 ;         //
            break;
      case  1: // intern 50 MHz
            *sample_clock = 50000 ;    //
            break;
      .....
} // end switch (gl_uint_SIS3302ClockModeConf)

switch (gl_uint_FirDecimationMode)  {
      case  0: // no decimation
            float_sampling_time_us = 1000.0/sample_clock  ;
            break;
      case  1: //
            float_sampling_time_us = 2000.0/sample_clock  ;
            break;
      case  2: //
            float_sampling_time_us = 4000.0/sample_clock  ;
            break;
      case  3: //
            float_sampling_time_us = 8000.0/sample_clock  ;
            break;
} // end switch (gl_uint_FirDecimationMode)
```

```
float_decay_factor  = (Tau factor / 32768.0 ) ;

float_decay_time_us = -(float_sampling_time_us / (log(1.0-float_decay_factor)));
```

Example: 100 MHz; Decimation = 4

```
Tau Factor = 1     decay_time_us = 1310.69999990 us
Tau Factor = 2     decay_time_us =  655.33999980 us
Tau Factor = 3     decay_time_us =  436.88666636 us
Tau Factor = 4     decay_time_us =  327.65999959 us
Tau Factor = 5     decay_time_us =  262.12399949 us
Tau Factor = 6     decay_time_us =  218.43333272 us
Tau Factor = 7     decay_time_us =  187.22571357 us
Tau Factor = 8     decay_time_us =  163.81999919 us
Tau Factor = 9     decay_time_us =  145.61555464 us
Tau Factor = 10    decay_time_us =  131.05199898 us
Tau Factor = 11    decay_time_us =  119.13636252 us
Tau Factor = 12    decay_time_us =  109.20666545 us
Tau Factor = 13    decay_time_us =  100.80461406 us
Tau Factor = 14    decay_time_us =   93.60285572 us
Tau Factor = 15    decay_time_us =   87.36133181 us
Tau Factor = 16    decay_time_us =   81.89999837 us
Tau Factor = 17    decay_time_us =   77.08117474 us
Tau Factor = 18    decay_time_us =   72.79777595 us
Tau Factor = 19    decay_time_us =   68.96526122 us
Tau Factor = 20    decay_time_us =   65.51599796 us
Tau Factor = 21    decay_time_us =   62.39523596 us
Tau Factor = 22    decay_time_us =   59.55817958 us
Tau Factor = 23    decay_time_us =   56.96782375 us
Tau Factor = 24    decay_time_us =   54.59333089 us
Tau Factor = 25    decay_time_us =   52.40879746 us
Tau Factor = 26    decay_time_us =   50.39230505 us
Tau Factor = 27    decay_time_us =   48.52518244 us
Tau Factor = 28    decay_time_us =   46.79142572 us
Tau Factor = 29    decay_time_us =   45.17723843 us
Tau Factor = 30    decay_time_us =   43.67066361 us
Tau Factor = 31    decay_time_us =   42.26128717 us
Tau Factor = 32    decay_time_us =   40.93999674 us
Tau Factor = 33    decay_time_us =   39.69878452 us
Tau Factor = 34    decay_time_us =   38.53058477 us
Tau Factor = 35    decay_time_us =   37.42913929 us
Tau Factor = 36    decay_time_us =   36.38888522 us
Tau Factor = 37    decay_time_us =   35.40486110 us
Tau Factor = 38    decay_time_us =   34.47262771 us
Tau Factor = 39    decay_time_us =   33.58820116 us
Tau Factor = 40    decay_time_us =   32.74799593 us
Tau Factor = 41    decay_time_us =   31.94877631 us
Tau Factor = 42    decay_time_us =   31.18761477 us
Tau Factor = 43    decay_time_us =   30.46185609 us
Tau Factor = 44    decay_time_us =   29.76908643 us
Tau Factor = 45    decay_time_us =   29.10710653 us
Tau Factor = 46    decay_time_us =   28.47390836 us
Tau Factor = 47    decay_time_us =   27.86765479 us
Tau Factor = 48    decay_time_us =   27.28666178 us
Tau Factor = 49    decay_time_us =   26.72938277 us
Tau Factor = 50    decay_time_us =   26.19439491 us
Tau Factor = 51    decay_time_us =   25.68038696 us
Tau Factor = 52    decay_time_us =   25.18614855 us
```

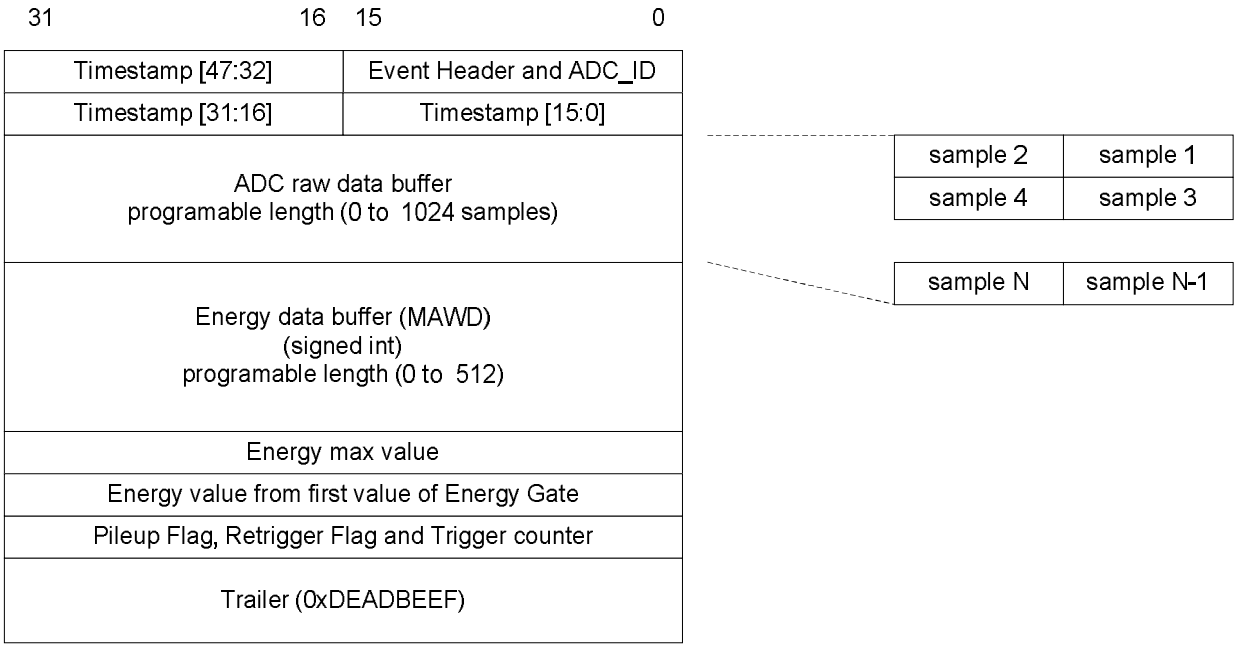
```
Tau Factor = 53    decay_time_us =   24.71056064 us
Tau Factor = 54    decay_time_us =   24.25258709 us
Tau Factor = 55    decay_time_us =   23.81126713 us
Tau Factor = 56    decay_time_us =   23.38570858 us
Tau Factor = 57    decay_time_us =   22.97508192 us
Tau Factor = 58    decay_time_us =   22.57861478 us
Tau Factor = 59    decay_time_us =   22.19558721 us
Tau Factor = 60    decay_time_us =   21.82532722 us
Tau Factor = 61    decay_time_us =   21.46720690 us
Tau Factor = 62    decay_time_us =   21.12063885 us
Tau Factor = 63    decay_time_us =   20.78507295 us
```

4.29 ADC memory

```
#define SIS3302_ADC1_OFFSET                    0x04000000
#define SIS3302_ADC2_OFFSET                    0x04800000
#define SIS3302_ADC3_OFFSET                    0x05000000
#define SIS3302_ADC4_OFFSET                    0x05800000
#define SIS3302_ADC5_OFFSET                    0x06000000
#define SIS3302_ADC6_OFFSET                    0x06800000
#define SIS3302_ADC7_OFFSET                    0x07000000
#define SIS3302_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.

4.29.1 Event Buffer Data Format



Event Header: see Event configuration registers

Pileup Flag and Trigger Counter:

Bits	31	30	30 :28	27 :24	D23:0
function	Pileup Flag	Retrigger Flag	0	Fast Trigger Counter	0

While the Trigger Gate is active the Trigger Counter is incremented with each delayed Trigger (up to 0xf).
- Pileup Flag is set if Fast Trigger Counter > 1
- Retrigger Flag is set if an earlier Fast Trigger was ((P+G)*Decimation) Clocks before the actual Fast Trigger. In this case the Energy Filter contains also the “Energy” of the earlier Fast Trigger.

5 Modes of acquisition of current software

The current software illustrates two data acquisition modes. Single event and multi event double buffer acquisition. They can be found in sis3150_gamma_running.c as routines:

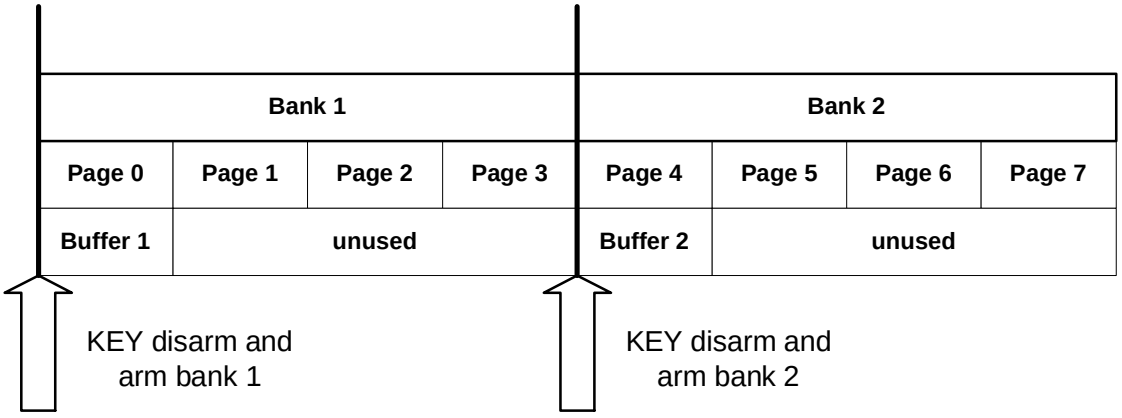
```
RunPC_SingleEventAquisition();
RunPC_SIS3302ANL_MultiEvent_DoubleBuffer_Aquisition();
```

5.1 Single event acquisition

Events are read out one by one in single event acquisition. This mode is most straightforward to implement but not suited for throughput.

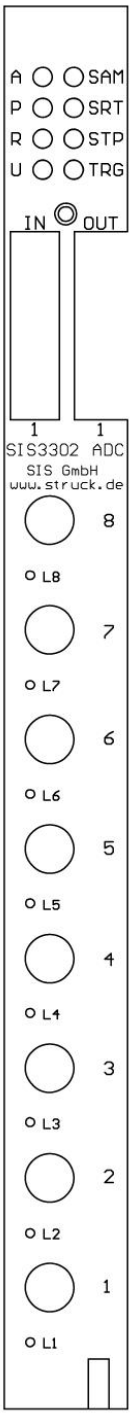
5.2 Multi Event Double Buffer acquisition

Dual bank acquisition with two memory sections of 8 MBytes each is implemented as illustrated below.
Typically events are acquired into one buffer until the end address threshold is reached. At this point in time acquisition is passed to the alternate bank/buffer with the key disarm and arm bank N command and data are readout from the inactive bank.
The key disarm and arm bank N registers position the memory pointer to the beginning of the corresponding buffer.



6 Front panel

The SIS3302 is a single width (4TE) 6U VME module. A sketch of the SIS3302 front panel (single ended LEMO00 version without handles) is shown below. The IN/OUT breakouts hold 4 LEMO connectors each.



6.1 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 (External Trigger)	ADC sample logic armed	3
2	Timestamp Clear	ADC sampling busy	2
1	Not used	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 .
The width of an external start/Timestamp Clear pulse must be greater or equal two sampling clock periods.

6.2 LED's

The SIS3302 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 SIS3302 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 on Bank 1
Yellow	SRT	Sampling busy on Bank 2
Green	STP	Lit if the lower Timestamp counter bits 27 to 0 are equal 0x0FFFFFFF (or of all four timestamp counters) . (100MHz -> lit every 2.6 sec)
Green	TRG	Trigger, lit if one or more channels are triggered

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

6.3 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.

6.4 PCB LEDs

The 8 surface mounted red LEDs D141A to D141G on the top left corner of the component side of the SIS3302 are routed to the control FPGA, their use may depend on the firmware design.

7 Jumpers/Configuration

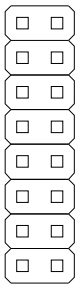
7.1 CON100 JTAG

The SIS3302 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/sis3302firm.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

7.2 JP80 VME addressing mode/reset behaviour

This 8 position jumper array is used to select the addressing mode and the reset behaviour of the SIS3302.


JP80

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.

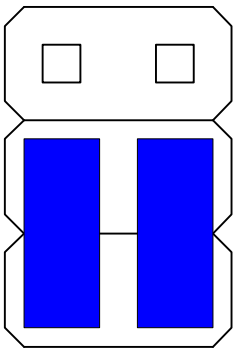
NOTE: avoid a power up deadlock situation by not setting Pos. 5 and 7 at the same time

7.3 JP101 JTAG chain

The JTAG chain on the SIS3302 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:

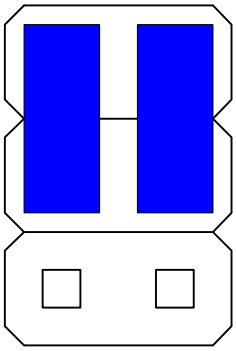
Long Chain (1-3 and 2-4 closed):

JP101



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

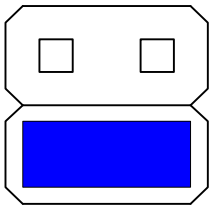
JP101



7.4 JP102 JTAG source

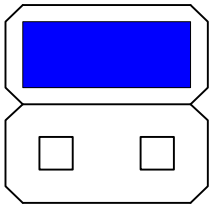
The JTAG chain can be connected to VME or to the JTAG connector CON...via the 4 pin jumper array JP102 as sketched below:

JTAG connected to VME (1-2 closed)



JP102

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

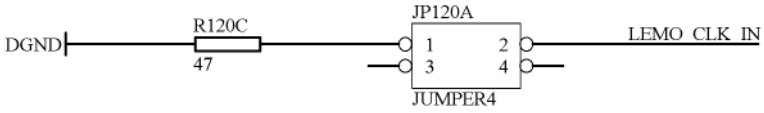


JP102

7.5 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



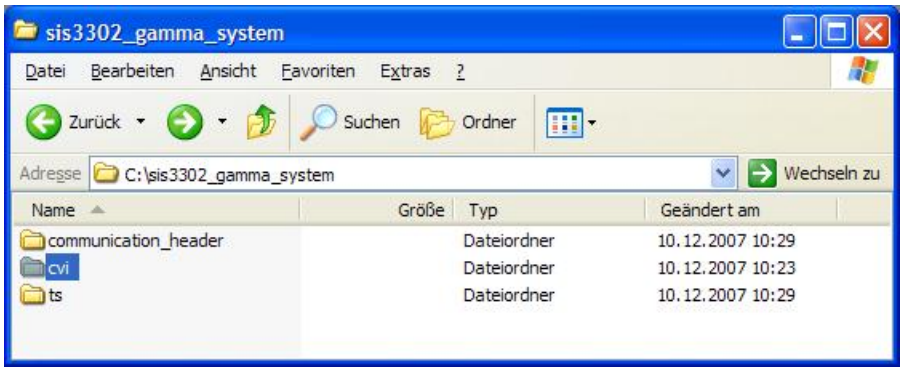
7.6 SW1 and SW2, VME base address

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

8 Software/getting started

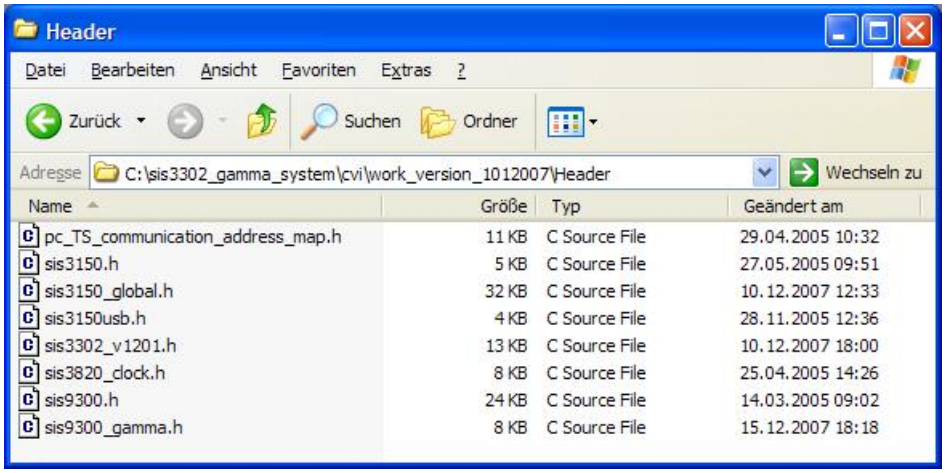
The original Gamma implementation was developed for a setup consisting of several SIS3150 CMC carrier boards with SIS9300 digitizer CMCs, a SIS3820 clock distributor and a SIS PCI or USB to VME interface. It was adopted for SIS3302 boards in combination with a SIS1100/3100 PCI to VME or SIS3150 USB to VME interface later on. The graphical user interface (GUI) is based on National Instruments CVI. As CVI applications are based on underlying C code you can use the code as basis or examples for adaptations to your environment and application.
The required files are kept in three directories as shown below.



8.1 Header files

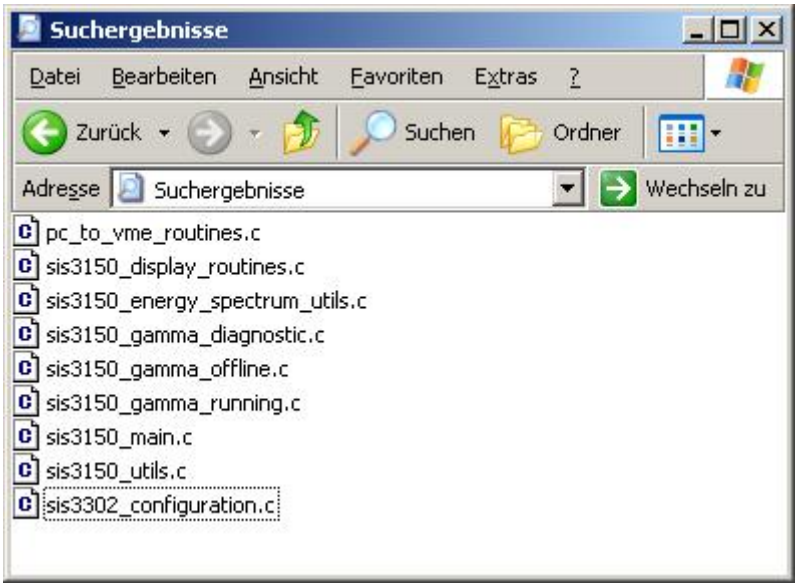
The registers of the SIS3302 gamma firmware can be found in:
`sis3302_v1201.h`

If you want to use the routines unchanged you may have to include header files for the SIS3820 clock distributor, for the sis3150 card and for the SIS9300 ADC board.
An overview of all header files of the project can be seen in the screen dump below.



8.2 C code

Following C files are part of the software:



Following files are most important to get started with your own project:
sis3302_configuration.c holds the configuration for the digitizer.
sis3150_gamma_running.c holds the readout code

9 Appendix

9.1 Power consumption

The SIS3302 uses standard VME voltages only.

Voltage	Current
+ 5V	8A
+12 V	115 mA
- 12 V	340 mA

9.2 Operating conditions

9.2.1 Cooling

Although the SIS3302 is mainly a 2.5 and 3.3 V 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. An air capacity in excess of 160 $m^3/_h$ is required. Unoccupied adjacent slots of the VME crate have to be equipped with filler modules to ensure proper air flow. 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.

9.2.2 Hot swap/live insertion

Please note, that the VME standard does not support hot swap by default. The SIS3302 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.

9.3 Connector types

The VME connectors and the different types of front panel connectors used on the SIS3302 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^o PCB LEMO	Analog input connector	LEMO EPL.00.250.NTN
90^o PCB	Analog input connector (SMA option)	SMA
90^o PCB LEMO	Analog input connector (3302 differential input version)	LEMO EPL.0S.302.HLN

9.4 P2 row A/C pin assignments

The P2 connector of the SIS3302 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 SIS3302 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.

9.5 Row d and z Pin Assignments

The SIS3302 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)

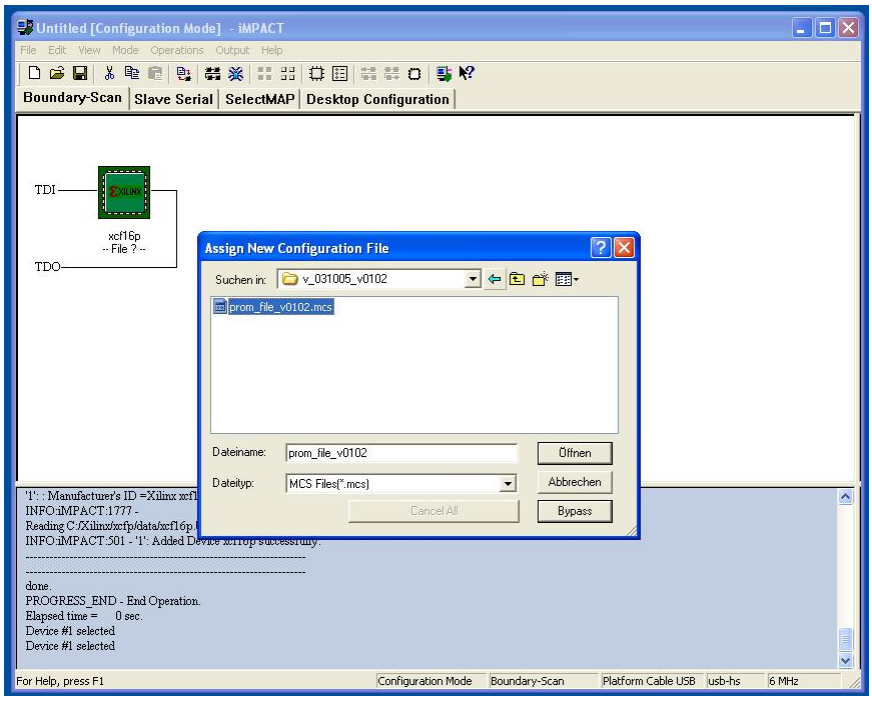
Note: Pins designated with (1) are so called MFBL (mate first-break last) pins on the installed 160 pin connectors, VPC(1) pins are connected via inductors.

9.6 Firmware upgrade

The firmware of the SIS3302 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.

9.6.1 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 SIS3302 for short JTAG chain (refer to section 7.3 JP101) and set the unit to JTAG over CON100 (refer to section 7.4 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).

9.6.2 Upgrade over VME

not tested with SIS3302 firmware 12 02 yet

10 Index