

# SIS3316 16 Channel 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](mailto:info@struck.de)  
<http://www.struck.de>

Version: SIS3316-M-1-1-V124.doc as of 24.04.2020

## Revision Table:

Revision	Date	Modification
1.00	24.01.13	<p>First official release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2002 (0x33162002)</li> <li>- ADC FPGAs: V0250-0002 (0x02500002)</li> </ul> <p>Important Note, relating to the software, written for previous Firmware Versions:</p> <ul style="list-style-type: none"> <li>• Global functions like “external (global) trigger”, “external (global) timestamp clear” or “external (global) veto” have to be enabled, now. (see Acquisition register)</li> </ul>
1.01	28.01.2013	Comments on time resolution, and bank check
1.02	06.02.2013	JTAG programming, temperature/cooling
1.03	10.04.2013	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2003 (0x33162003)</li> <li>- ADC FPGAs: V0250-0003 (0x02500003)</li> </ul> <p>Bug fixes:</p> <ul style="list-style-type: none"> <li>- UDP read of ADC FPGA Status registers (0x1100 – 0x113C)</li> </ul> <p>Added features:</p> <ul style="list-style-type: none"> <li>- “FP-Bus Bank Swap Control” Logic</li> <li>- “Sample Bank Swap Control with NIM Input TI/UI” Logic</li> </ul> <p>Modifications:</p> <ul style="list-style-type: none"> <li>- the XILINX Soft Tri-Mode Ethernet MAC core is <b>full</b> licensed with no time restriction</li> <li>- extended LEMO Out “CO”, “TO” and “UO” Select registers with new sources</li> <li>- extended Acquisition Control register</li> <li>- added new Key address: Enable “Sample Bank Swap Control with NIM Input TI/UI” logic</li> </ul>

1.04	27.11.2013	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2004 (0x33162004)</li> <li>- ADC FPGAs: V0250-0004 (0x02500004)</li> <li>- ADC FPGAs: V0125-0004 (0x01250004)</li> </ul> <p>Bug fixes in firmware:</p> <ul style="list-style-type: none"> <li>- Trigger Filter with P=2</li> </ul> <p>Added features in firmware:</p> <ul style="list-style-type: none"> <li>- Energy Filter</li> <li>- Energy histogramming inside the FPGA</li> </ul> <p>Modifications:</p> <ul style="list-style-type: none"> <li>- speedup internal data transfer from 1.25 GHz to 2.5GHz</li> <li>- stretched trigger output:           <ul style="list-style-type: none"> <li>* selectable high energy stretched trigger output</li> <li>* 250MHz: trigger output synchronous to one sample clock</li> </ul> </li> <li>- adc iob tap delay logic:           <ul style="list-style-type: none"> <li>* add “add ½ Sample Clock periode delay bit”</li> </ul> </li> </ul>
1.05	10.12.2013	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2004 (0x33162004)</li> <li>- ADC FPGAs: V0250-0004 (0x02500004)</li> <li>- ADC FPGAs: V0125-0004 (0x01250004)</li> </ul> <p>Added features in firmware (only SIS3316-16bit, V0125-0004):</p> <ul style="list-style-type: none"> <li>- Average mode</li> <li>  Add Average Configuration register</li> </ul>
1.06	13.03.2014	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2004 (0x33162004)</li> <li>- ADC FPGAs: V0250-0005 (0x02500005)</li> <li>- ADC FPGAs: V0125-0005 (0x01250005)</li> </ul> <p>Added features in firmware: (only SIS3316-14bit, V0250-0005):</p> <ul style="list-style-type: none"> <li>- Peak/Charge mode with external Gate</li> <li>  Add Peak/Charge Configuration register</li> </ul>

1.07	05.05.2014	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2005 (0x33162005)</li> <li>- ADC FPGAs: V0250-0006 (0x02500006)</li> <li>- ADC FPGAs: V0125-0006 (0x01250006)</li> </ul> <p>Modifications (VME FPGA, ADC FPGA):</p> <ul style="list-style-type: none"> <li>- automatically internal data transfer speed setting: Modules with serial number 1-10: set to 1.25 GHz Modules with serial number 11 and higher: set to 2.50 GHz</li> </ul> <p>Documentation Modifications:</p> <ul style="list-style-type: none"> <li>- Analog Input stage</li> <li>- Energy Filter and internal Energy Histogramming <ul style="list-style-type: none"> <li>* four channel group blockdiagram</li> <li>* memory organisation blockdiagram</li> </ul> </li> </ul>
1.08	20.08.2014	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2005 (0x33162005)</li> <li>- ADC FPGAs: V0250-0007 (0x02500007)</li> <li>- ADC FPGAs: V0125-0007 (0x01250007)</li> </ul> <p>Modification in ADC FPGA firmware (V0xxx-0007):</p> <ul style="list-style-type: none"> <li>- Increase “Pre Trigger Delay” from 2K to 16K <ul style="list-style-type: none"> <li>* modifier “Pre Trigger Delay” registers</li> </ul> </li> <li>- Increase “Raw Data Sample Length” from 64K to 32M samples <ul style="list-style-type: none"> <li>* add “Extended Raw Data Buffer Configuration” registers see “Raw Data Buffer Configuration” registers, also</li> </ul> </li> </ul>

1.09	12.12.2014	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2006 (0x33162006)</li> <li>- ADC FPGAs: V0250-0008 (0x02500008)</li> <li>- ADC FPGAs: V0125-0008 (0x01250008)</li> </ul> <p>Modification in VME FPGA firmware (V3316-2006):</p> <ul style="list-style-type: none"> <li>- UDP (Ethernet access)           <ul style="list-style-type: none"> <li>* Jumbo Frame support</li> <li>* DHCP support</li> <li>* response on Ping-request</li> </ul> </li> <li>- add sixteen (16) 32-bit “Internal Trigger” counters</li> <li>- add PPS (pulse per second) input logic</li> <li>- add „External Trigger Disable with internal Busy“ logic</li> <li>- modified ADC FPGA SPI BUSY Status register</li> </ul> <p>Modification in ADC FPGA firmware (V0xxx-0008):</p> <ul style="list-style-type: none"> <li>- add PPS (pulse per second) latch logic</li> <li>- add “Suppress saving of more Hits/Events if Memory Address Threshold Flag is valid” Enable bit in End Address Threshold register</li> </ul> <p>Documentation:</p> <ul style="list-style-type: none"> <li>- add dead time description</li> <li>- add Energy Filter description</li> </ul>
1.10	06.03.2015	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2008 (0x33162008)</li> <li>- ADC FPGAs: V0250-0008 (0x02500008)</li> <li>- ADC FPGAs: V0125-0008 (0x01250008)</li> </ul> <p>Modification in VME FPGA firmware (V3316-2008):</p> <ul style="list-style-type: none"> <li>- UDP (Ethernet access)           <ul style="list-style-type: none"> <li>* modified DHCP protocol (“LINUX DHCP-Server” protocol)</li> <li>* changed UDP access protocol (Packet Identifier added)</li> <li>* “Read Last Packet again” command (Retry) added</li> </ul> </li> <li>- extended Acquisition control/status register</li> <li>- extended LEMO Out “TO” Select register</li> <li>- extended LEMO Out “UO” Select register</li> <li>- add a 32-bit prescaler for the sampling clock with UO output possibility</li> <li>- add Trigger Coincidence Lookup Table</li> <li>- add Feedback Selected Internal Trigger as External Trigger logic</li> </ul> <p>Documentation:</p> <ul style="list-style-type: none"> <li>- modified Tap-delay table (250MHz -&gt; from 0x08 to 0x02 !)</li> </ul>

1.11	04.06.2015	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2009 (0x33162009)</li> <li>- ADC FPGAs: V0250-0009 (0x02500009)</li> <li>- ADC FPGAs: V0125-0009 (0x01250009)</li> </ul> <p>Modifications in VME FPGA firmware (V3316-2009):</p> <ul style="list-style-type: none"> <li>- 512 Mbyte DDR3 Memory (4GBit) flag are saved in EEPROM see “Serial Number register”</li> </ul> <p>Modifications in ADC FPGA firmware (V0250-0009, V0125-0009):</p> <ul style="list-style-type: none"> <li>- support 512 Mbyte DDR3 Memory (4GBit) chips (only 256 Mbyte are supported, yet)</li> </ul>
1.12	01.07.2015	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200A (0x3316200A)</li> <li>- ADC FPGAs: V0250-000A (0x0250000A)</li> <li>- ADC FPGAs: V0125-000A (0x0125000A)</li> </ul> <p>Modifications in VME FPGA firmware (V3316-200A):</p> <ul style="list-style-type: none"> <li>- UDP (Ethernet access) <ul style="list-style-type: none"> <li>* supports optical Ethernet (optical SFP)</li> <li>* fix bug: read/write multiple registers with “SIS3316 Register Space Read Request command”</li> </ul> </li> </ul> <p>Modifications in ADC FPGA firmware (V0250-000A, V0125-000A):</p> <ul style="list-style-type: none"> <li>- Increase “MAW Test Sample Length” from 1K to 2K <ul style="list-style-type: none"> <li>* modified “MAW Test Buffer Configuration” registers</li> </ul> </li> <li>- Add “MAW Start Index and Energy Pickup Configuration” registers <ul style="list-style-type: none"> <li>* A MAW Start Index is used to delay the start of saving the Trigger or Energy trapezoidal (post-trigger) into the MAW Test buffer.</li> <li>* An Energy Pickup Index value can be used to save the Energy value from a defined position instead of saving the maximum value of the trapezoidal.</li> </ul> </li> <li>- Modified “High Energy Trigger Threshold” registers <ul style="list-style-type: none"> <li>* A Pileup detection pulse can be routed to the VME FPGA as “Internal trigger” or “High-Energy trigger”.</li> </ul> </li> <li>- Add “Extended Event Configuration” registers <ul style="list-style-type: none"> <li>* add “Ch x internal Pileup Trigger enable” bits</li> </ul> </li> </ul>

1.13	27.07.2015	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200A (0x3316200A)</li> <li>- ADC FPGAs: V0250-000A (0x0250000A)</li> <li>- ADC FPGAs: V0125-000A (0x0125000A)</li> </ul> <p>Documentation:</p> <ul style="list-style-type: none"> <li>- “Energy pickup index functionality”</li> <li>- “Extended Event Configuration” registers</li> <li>- “Pileup Trigger generation”</li> </ul>
1.14	06.08.2015	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200A (0x3316200A)</li> <li>- ADC FPGAs: V0250-000A (0x0250000A)</li> <li>- ADC FPGAs: V0125-000A (0x0125000A)</li> </ul> <p>Documentation:</p> <ul style="list-style-type: none"> <li>- modified “Hit/Event Data format” block diagrams</li> <li>    wrong order of “MAW value before Trigger” and “MAW value after Trigger”</li> <li>- support for optical Ethernet</li> </ul>
1.15	14.10.2015	<p>Documentation:</p> <ul style="list-style-type: none"> <li>- add LVDS Bus connector schematic</li> </ul>
1.16	22.03.2016	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200B (0x3316200B)</li> <li>- ADC FPGAs: V0250-000B (0x0250000B)</li> <li>- ADC FPGAs: V0125-000B (0x0125000B)</li> </ul> <p>Modifications in VME FPGA firmware (V3316-200B):</p> <ul style="list-style-type: none"> <li>- add an “External Veto” (Gate) Delay logic</li> <li>    * New External Veto/Gate Delay register (0x3C)</li> </ul> <p>Modifications in ADC FPGA firmware (V0250-000B, V0125-000B):</p> <ul style="list-style-type: none"> <li>- fix bug: Multievent Bank Swap logic stops in rare cases</li> <li>- “internal SUM-Trigger stretched pulse ch...” to VME FPGA</li> <li>    This stretched pulse will be only routed to the VME FPGA if the SUM-FIR-Trigger logic is enabled</li> </ul> <p>(SIS3316_ADC_CH..._SUM_FIR_TRIGGER_THRESHOLD_REG bit 31 = 1).</p> <p>If the SUM-FIR-Trigger logic is disabled then the “Internal Gate 1” will be routed to the VME FPGA , see</p> <p>SIS3316_ADC_CH..._INTERNAL_GATE_LENGTH_CONFIG_REG.</p>
1.17	02.06.2016	Power consumption table update

1.18	17.05.2018	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200C (0x3316200C)</li> <li>- ADC FPGAs: V0250-000C (0x0250000C)</li> <li>- ADC FPGAs: V0125-000C (0x0125000C)</li> </ul> <p>Modifications in VME FPGA firmware (V3316-200C):</p> <ul style="list-style-type: none"> <li>- modification in “Trigger Coincidence Lookup table” logic</li> </ul> <p>Modifications in ADC FPGA firmware (V0250-000C, V0125-000C):</p> <ul style="list-style-type: none"> <li>- modification in “Ch x High Energy Trigger stretched pulse generation” logic</li> </ul> <p>Documentation:</p> <ul style="list-style-type: none"> <li>- add “supported address modifier (AM)” description</li> </ul>
1.19	10.01.2019	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200F (0x3316200F)</li> <li>- ADC FPGAs: V0250-000E (0x0250000E)</li> <li>- ADC FPGAs: V0125-000C (0x0125000C)</li> </ul> <p>VME FPGA history:</p> <ul style="list-style-type: none"> <li>- <b>V3316-200D:</b> add UI-TDC logic (see sis3316-M-1-v100-sis3820-clock-addendum.pdf)</li> <li>- <b>V3316-200E:</b> VME IRQ vector bug fix</li> <li>- <b>V3316-200F:</b> <ul style="list-style-type: none"> <li>- add “User Counter logic”           <ul style="list-style-type: none"> <li>* insert 32-bit user counter in Header if enabled</li> <li>* use UI input as Clear</li> <li>* use TI input as Clock (increment)</li> </ul> </li> </ul> </li> </ul> <p>ADC FPGA history:</p> <ul style="list-style-type: none"> <li>- <b>V0250-000D:</b> max. MAW value bug fix       <ul style="list-style-type: none"> <li>* took wrong max. MAW value in rare cases</li> </ul> </li> <li>- <b>V0250-000E:</b> <ul style="list-style-type: none"> <li>- add “User Counter logic”</li> <li>- add “Decimation/Average” mode (4, 8, 16, ..., 512)</li> </ul> </li> </ul>
1.20	22.01.2019	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-200F (0x3316200F)</li> <li>- ADC FPGAs: V0250-000F (0x0250000F)</li> <li>- ADC FPGAs: V0125-000C (0x0125000C)</li> </ul> <p>ADC FPGA: V0250-000F</p> <ul style="list-style-type: none"> <li>- modified “Decimation/Average” logic (2, 4, 8, 16, ..., 512)       <ul style="list-style-type: none"> <li>- add decimation/average factor 2</li> </ul> </li> </ul>
1.21	23.05.2019	Fix in this table (V1.19 and 1.20 related)
1.22	18.06.2019	Add table with digitizer chips and 0.2/2V 250 MSPS version

1.23	28.10.2019	<p>release related to Firmware Versions</p> <ul style="list-style-type: none"> <li>- VME FPGA: V3316-2010 (0x33162010)</li> <li>- ADC FPGAs: V0250-0010 (0x02500010)</li> <li>- ADC FPGAs: V0125-0010 (0x01250010)</li> <li>(skipped Versions 0x0125000D/E/F)</li> </ul> <p>VME FPGA: V3316-2010</p> <ul style="list-style-type: none"> <li>- modified/expanded LEMO Out “TO” Select register</li> <li>- modified/expanded LEMO Out “UO” Select register</li> <li>- add “External Trigger function Deadtime” logic Can be used with “External Trigger” or “External Gate” applications to save the Events of all Channels synchronously           <ul style="list-style-type: none"> <li>* modified/expanded “External Veto/Gate Delay” register</li> <li>* modified/expanded “NIM Input Control/Status” register</li> </ul> </li> </ul> <p>ADC FPGA: V0125-0010 / V0250-0010</p> <ul style="list-style-type: none"> <li>- fix bug: Raw data of an Event are stored one Event later and doesn't correspond to the Timestamp and Gate values. This behaviour occurs after a Bank Switch in case that the raw data are saved with a Start-Index unequal zero. It is very rare and only in combination with high event rate.</li> <li>- modified/expanded Extended Event configuration registers           <ul style="list-style-type: none"> <li>* add “Save Raw data of first Event of Bankbuffer only” Enable bit</li> </ul> </li> <li>- add ADC Test Function registers           <ul style="list-style-type: none"> <li>* add “Save Internal Gate as Bit 0 of ADC data” Enable bit</li> </ul> </li> </ul>
1.24	24.04.2020	<p>Documentation:</p> <ul style="list-style-type: none"> <li>- fix in chapter “9.1.6 SW1/SW2 VME Base Address Rotary switches“</li> </ul>

## Table of contents

1	Introduction .....	14
2	Technical Properties/Features .....	15
2.1	SIS3316 VME board Overview .....	16
2.2	Block Diagram .....	17
2.3	Sample Clock distribution .....	18
2.3.1	Internal Programmable Sample Clock Oscillator .....	19
2.3.2	External NIM In Programmable Precision Clock Multiplier .....	19
2.3.3	Sample Clock configuration sequence .....	21
2.4	Analog Input stage .....	22
2.5	SFP cage .....	23
2.5.1	Ethernet SFP .....	23
2.5.1.1	Optical Ethernet SFP .....	23
2.5.2	Optical SFP .....	24
2.6	VXS .....	24
2.7	MGT Link Connections .....	25
2.8	Memory Handling .....	26
3	VME/CTRL FPGA Application Firmware .....	27
3.1	Sample Control overview .....	28
3.2	External (global) Signal overview .....	32
3.3	External Sample Control I/Os .....	33
3.3.1	External NIM .....	33
3.3.1.1	NIM Outputs .....	33
3.3.1.2	NIM Inputs .....	33
3.3.2	Front panel LVDS Bus .....	35
3.4	External (global) input signal distribution .....	37
3.5	Acquisition Control .....	38
3.6	Internally generated Trigger Features .....	39
3.6.1	Internal Trigger Counters .....	39
3.6.2	Distribute to LEMO outputs .....	39
3.6.3	Feedback Selected Internal Trigger as “External Trigger” to the ADC FPGAs .....	39
3.6.4	Trigger Coincidence Lookup table .....	40
3.7	PPS logic .....	42
3.8	Prescaler of the sampling clock with (possible) output to UO .....	42
4	ADC FPGA Application Firmware (Neutron/Gamma, Gamma) .....	43
4.1	Four Channel group blockdiagram .....	44
4.2	Trigger/Gate Logic .....	45
4.2.1	Coincidence Gate Logic inside a four channel group .....	47
4.3	Internal Trigger Generation .....	49
4.3.1	Block diagram of the Trigger MAW unit .....	49
4.3.2	Signal diagram of the Trigger MAW unit .....	50
4.3.3	CFD Trigger Feature .....	51
4.3.4	Higher Timestamp Resolution .....	52
4.3.4.1	Higher Timestamp Resolution Example 1 .....	53
4.3.4.2	Higher Timestamp Resolution Example 2 .....	55
4.3.5	High Energy Suppress Trigger Mode .....	57
4.3.6	Pileup Trigger Generation .....	58
4.4	Sample Logic .....	59
4.4.1	Neutron/Gamma application (accumulators, integration) .....	59
4.4.2	Gamma application (Trapezoidal FIR Energy Filter) .....	61
4.5	Pileup Detection Logic .....	66
4.5.1	Pileup Window .....	66
4.5.2	Re-Pileup (Pre-Pileup) Window .....	68
4.6	Hit/Event Data Format .....	69
4.7	Dead time .....	70
4.7.1	Sample Bank Swap logic dead time .....	70
4.7.2	Hit/Event storage dead time .....	71
4.8	Memory Organization .....	74
4.8.1	Double Bank mode .....	74
4.8.2	Single Bank mode .....	74

4.9	Trigger Statistic Counter .....	75
4.10	Peak/Charge ADC mode (Peak/Charge Sensing ADC) .....	77
4.10.1	SIS3316 Peak/Charge Data Format.....	77
4.11	Averaging mode (SIS3316-125MHz-16bit, only).....	78
4.11.1	SIS3316 Average Data Format (SIS3316-125MHz-16bit, only) .....	79
4.12	Average/Decimation mode (SIS3316-250MHz-14bit, only) .....	80
4.12.1	Average/Decimation Example .....	81
4.13	“User Counter” (SIS3316-250MHz-14bit, only) .....	82
5	VME Addressing.....	83
5.1	Address Map Overview.....	84
5.1.1	VME FPGA interface registers .....	85
5.1.2	VME FPGA registers .....	86
5.1.3	Key address registers.....	88
5.1.4	FPGA ADC group 1 registers (Ch1 to Ch4) .....	89
6	Register Description.....	92
6.1	Control/Status Register(0x0, write/read).....	92
6.2	Module Id. and Firmware Revision Register .....	94
6.2.1	Major revision numbers .....	94
6.3	Interrupt configuration register (0x8).....	95
6.3.1	IRQ mode .....	95
6.4	Interrupt control register (0xC) .....	96
6.5	Interface Access Arbitration Control Register .....	97
6.6	Broadcast setup register .....	98
6.7	Hardware Version Register .....	100
6.8	Temperature register .....	101
6.9	Onewire EEPROM Control register.....	102
6.10	Serial Number register .....	104
6.11	Internal Transfer Speed register .....	105
6.12	ADC FPGA Boot control register .....	106
6.13	SPI Flash Control/Status register .....	107
6.14	SPI Flash Data register.....	108
6.15	External Veto/Gate Delay register .....	109
6.16	Programmable Clock I2C registers .....	110
6.16.1	Programmable ADC Clock I2C register.....	110
6.16.2	Programmable MGT1 Clock I2C register .....	110
6.16.3	Programmable MGT2 Clock I2C register .....	110
6.16.4	Programmable DDR3 Clock I2C register.....	110
6.17	ADC Sample Clock distribution control register .....	111
6.17.1	ADC Sample Clock Multiplexer select bits: .....	111
6.18	External NIM Clock Multiplier SPI register .....	112
6.19	FP-Bus control register .....	113
6.20	NIM Input Control/Status register.....	114
6.21	Acquisition control/status register (0x60, read/write) .....	115
6.22	Trigger Coincidence Lookup Table Control register .....	117
6.23	Trigger Coincidence Lookup Table Address register.....	117
6.24	Trigger Coincidence Lookup Table Data register .....	118
6.25	LEMO Out “CO” Select register.....	119
6.26	LEMO Out “TO” Select register .....	120
6.27	LEMO Out “UO” Select register.....	121
6.28	Internal Trigger Feedback Select register.....	122
6.29	ADC FPGA Data Transfer Control registers.....	123
6.30	ADC FPGA Data Transfer Status registers .....	124
6.1	VME FPGA – ADC FPGA Data Link Status register.....	125
6.2	ADC FPGA SPI BUSY Status register .....	126
6.3	Prescaler Output Pulse Divider register .....	127
6.4	Prescaler Output Pulse Length register .....	127
6.5	Channel 1 to 16 Internal Trigger Counters.....	128
6.6	ADC Input tap delay registers .....	129
6.7	ADC Gain and Termination Control register .....	131
6.8	ADC Offset (DAC) Control registers.....	132

6.9	ADC Offset (DAC) Readback registers .....	134
6.10	ADC SPI Control register .....	135
6.11	ADC SPI Readback registers .....	136
6.12	Event configuration registers .....	137
6.13	Extended Event configuration registers .....	138
6.14	Channel Header ID registers .....	139
6.15	End Address Threshold register .....	140
6.16	Active Trigger Gate Window Length registers .....	141
6.17	Raw Data Buffer Configuration registers.....	142
6.18	Pileup Configuration registers.....	143
6.19	Pre Trigger Delay registers .....	144
6.20	Average Configuration registers .....	145
6.20.1	Average Configuration registers (SIS3316-125MHz-16bit) .....	145
6.20.2	Average/Decimation Configuration registers (SIS3316-250MHz-14bit).....	146
6.21	Data Format Configuration registers.....	148
6.22	MAW Test Buffer Configuration registers .....	149
6.23	Internal Trigger Delay Configuration registers .....	150
6.24	Internal Gate Length Configuration registers.....	151
6.25	FIR Trigger Setup registers .....	152
6.26	Trigger Threshold registers .....	153
6.27	High Energy Trigger Threshold registers.....	155
6.28	Trigger Statistic Counter Mode register.....	157
6.29	Peak/Charge Configuration registers.....	158
6.30	Extended Raw Data Buffer Configuration registers.....	159
6.31	Accumulator Gate X Configuration registers.....	160
6.32	FIR Energy Setup registers .....	161
6.33	Energy Histogram Configuration registers.....	163
6.33.1	Calculation of the Energy to the Histogram Index: .....	165
6.34	MAW Start Index and Energy Pickup Configuration registers .....	166
6.35	Test Function registers .....	167
6.36	ADC FPGA Firmware Version Register .....	168
6.37	ADC FPGA Status register .....	169
6.38	Actual Sample address registers.....	170
6.39	Previous Bank Sample address registers .....	171
6.40	Key addresses (0x400 – 0x43C write only) .....	172
6.40.1	Key address: Register Reset .....	172
6.40.2	Key address: User function logic .....	172
6.40.3	Key address: Arm sample logic.....	172
6.40.4	Key address: Disarm sample logic .....	172
6.40.5	Key address: Trigger .....	172
6.40.6	Key address: Timestamp Clear.....	173
6.40.7	Key address: Disarm Bankx and Arm Bank1 .....	173
6.40.8	Key address: Disarm Bankx and Arm Bank2 .....	173
6.40.9	Key address: Enable Sample Bank Swap Control with NIM Input TI/UI Logic .....	173
6.40.10	Key address: Disable Prescaler Output Pulse Divider logic.....	174
6.40.11	Key address: PPS_Latch_Bit_clear.....	174
6.40.12	Key address: Reset ADC-FPGA-Logic.....	174
6.40.13	Key address: ADC Clock DCM/PLL Reset .....	174
7	LEDs .....	175
7.1	Front Panel LEDs .....	175
7.2	SMD LEDs.....	175
8	Front panel .....	176
9	Board Layout .....	177
9.1	Configuration switches and jumpers .....	178
9.1.1	CON600B JTAG .....	178
9.1.2	J601 JTAG chain.....	179
9.1.2.1	Recommended JTAG Firmware Upgrade procedure: .....	180
9.1.3	JP900A-D LVDS termination .....	182
9.1.4	JP903A/B/C NIM input termination .....	183
9.1.5	JP971A/B Open collector termination.....	184

---

9.1.6	SW1/SW2 VME Base Address Rotary switches.....	185
9.1.7	SW80.....	185
10	Appendix.....	186
10.1	Power Consumption .....	186
10.2	FPGA Firmware Update over VME.....	186
10.3	Connector types.....	187
10.4	ADC/Digitizer Chip .....	187
10.5	Ordering options.....	187
10.6	Accessories.....	187
10.7	Software .....	188
10.7.1	Firmware Upgrade over VME or Ethernet.....	188
10.7.2	Ethernet operation.....	188
11	Index .....	189

## 1 Introduction

The SIS3316 is our first 16 channel digitizer card. It combines high channel density per card with the availability of greatly enhanced FPGA resources per channel and a flexible analog input stage followed by high resolution digitizer/ADC chips. The SIS3316 comes in two base configurations as listed in the table below.

Model	Sampling Speed	Resolution
SIS3316-125-16	125 MSPS	16-bit
SIS3316-250-14	250 MSPS	14-bit



SIS3316

As we are aware, that no manual is perfect, we appreciate your feedback and will 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](mailto:info@struck.de), the revision dates are online under <http://www.struck.de/manuals.html>. Feel free to apply for an account for our PDF documentation web page also.

The SIS3316 firmware page is at [www.struck.de/sis3316firm.html](http://www.struck.de/sis3316firm.html).

Information on SIS3316 applications, firmware news and other related issues will be posted on our DAQ blog at [www.struck.de/blog](http://www.struck.de/blog) also.



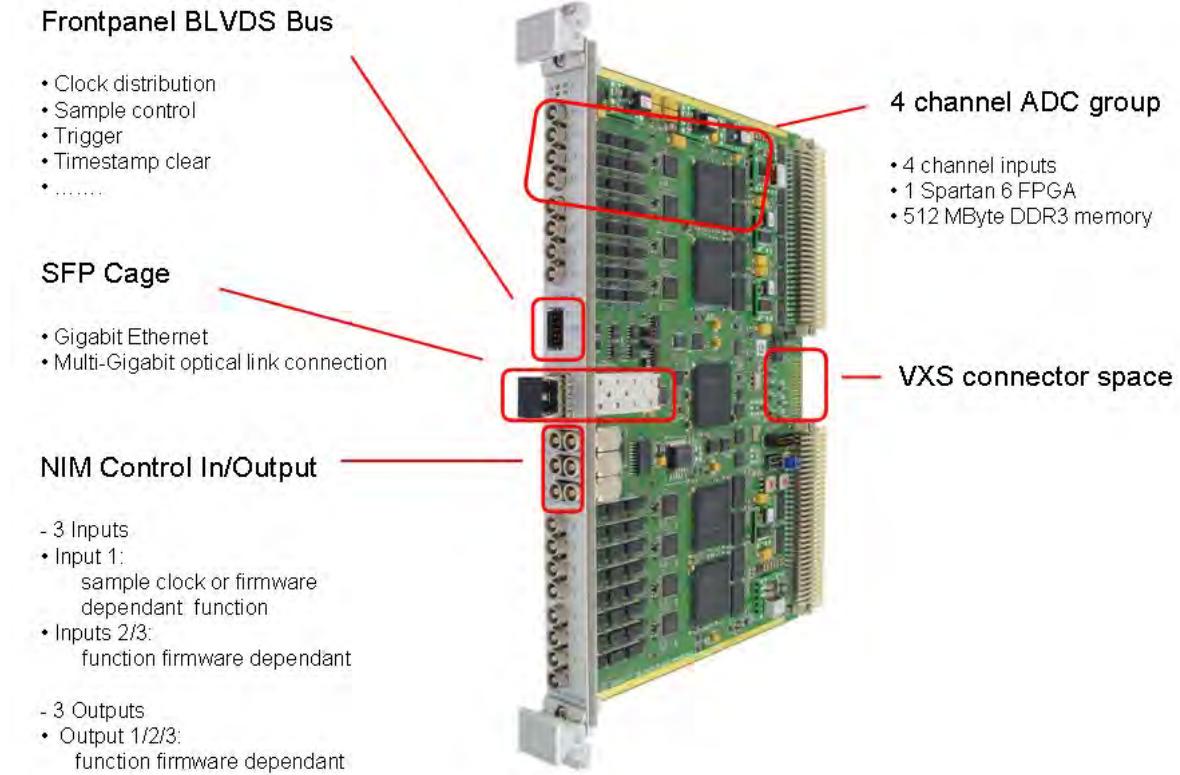
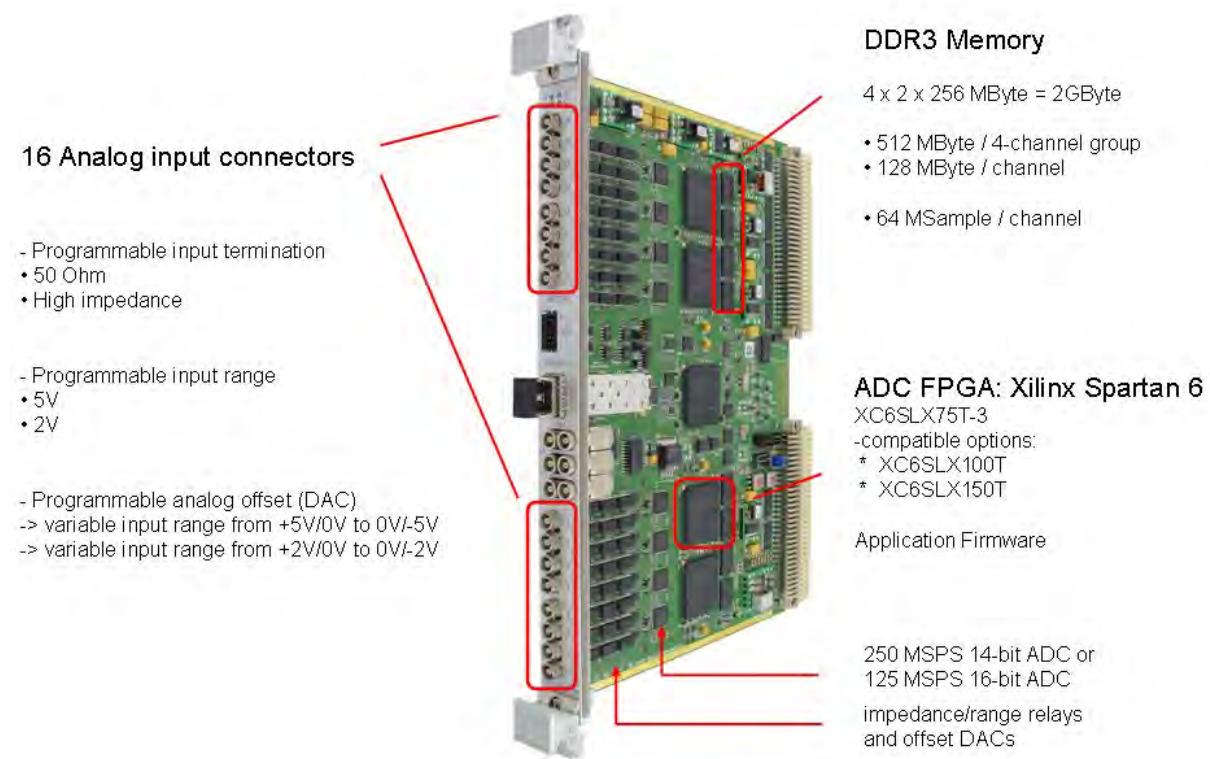
## 2 Technical Properties/Features

The SIS3316 board is a versatile 16 channel VME digitizer card.

Find below a list of key features of the SIS3316 digitizer.

- Single width 6U VME card
- 16 channels
- 125/250 MSPS per channel (simultaneous sampling)
- 62.5/125 MHz analog default bandwidth (others on request)
- 16-bit/14-bit resolution
- 64 MSamples memory/channel
- Two programmable input ranges
- 50 Ω or high impedance programmable
- Offset DACs
- Internal/External clock
- Multi event mode
- Double bank mode
- Readout in parallel to acquisition
- Pre/Post trigger capability
- Internal trigger generation
- Trigger OR output (16 individual thresholds)
- Front panel clock/trigger bus (FP-Bus)
- LEMO NIM Clock, Trigger and User (Veto) in
- LEMO NIM Clock, Trigger and User out
- SFP cage (Gigabit Ethernet or Multi-Gigabit optical link connection)
- A32 / D32 / BLT32 / MBLT64 / 2eVME / SSTVME
- 1-wire Id. serial PROM
- In field JTAG and VME firmware upgrade capability
- VXS implementation on request

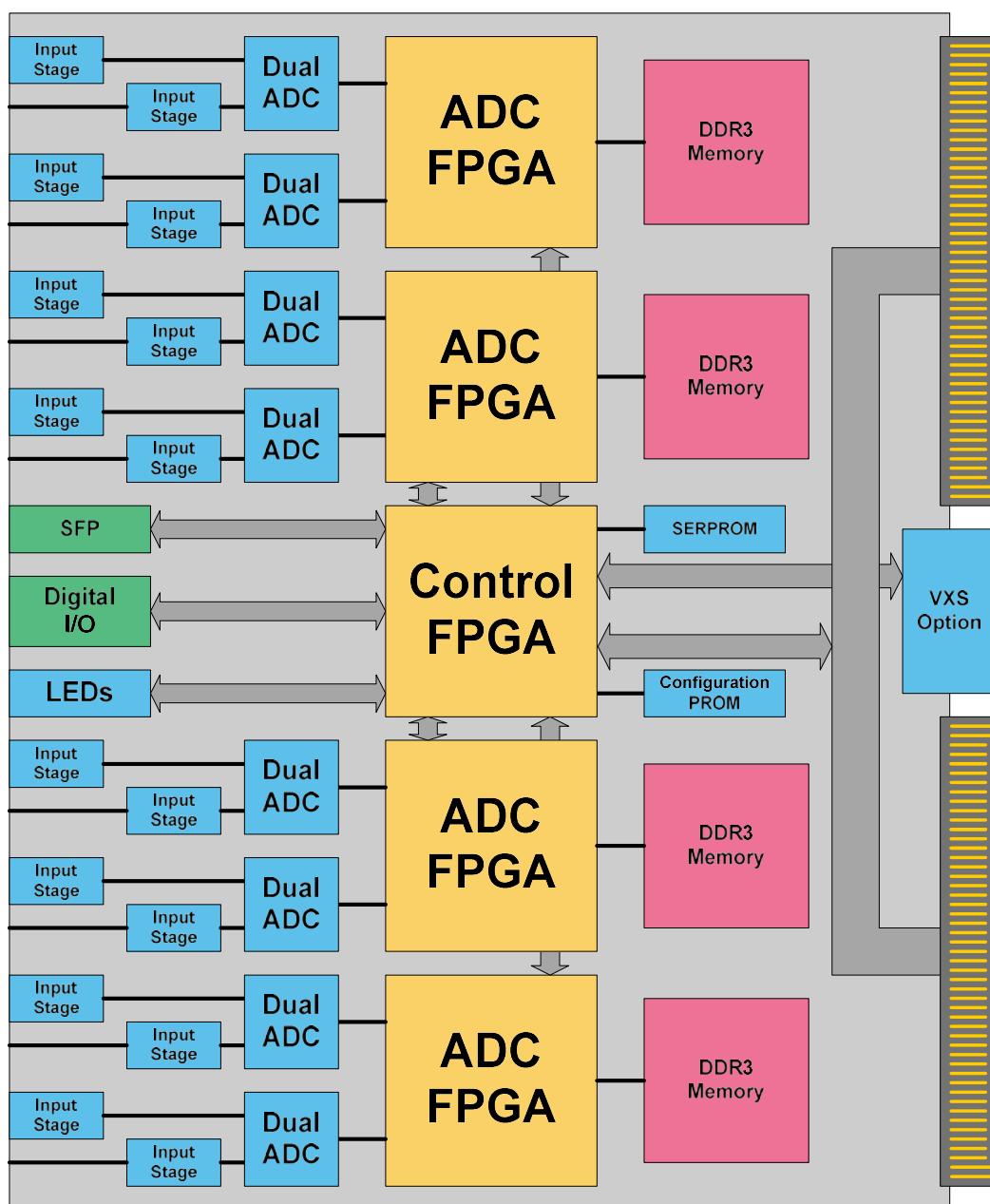
## 2.1 SIS3316 VME board Overview



## 2.2 Block Diagram

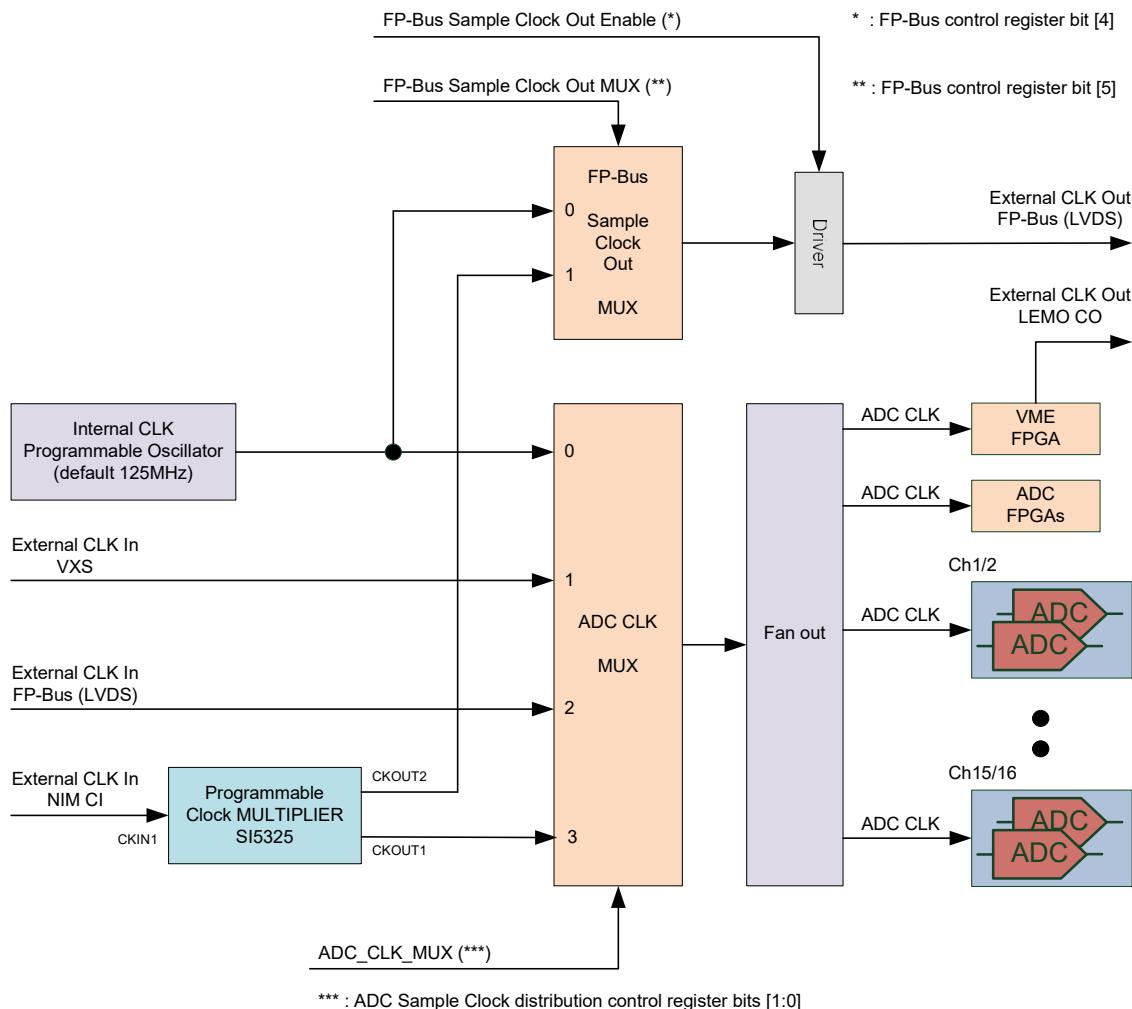
A simplified block diagram of the SIS3316 is shown below.

SIS3316 VME Digitizer



### 2.3 Sample Clock distribution

A simplified diagram of the clock distribution is shown below.



The SIS3316 features the following four clock modes:

- Internal programmable clock oscillator (default after power up is 125 MHz)
- External “Clock In” from VXS-Bus (option)
- External “Clock In” from FP-Bus (LVDS)
- External “Clock In” from NIM CI (routed through a programmable Clock Multiplier)

Sample clock requirements for the 250 MHz ADC 14-bit version (AD9643):

- frequency from 40 MHz to 250 MHz (limited by ADC chips and FPGAs)
- ratio between 45:55 and 55:45

Sample clock requirements for the 125 MHz 16-bit ADC version (AD9268):

- frequency from 20 MHz to 125 MHz (limited by ADC chips and FPGAs)
- ratio between 45:55 and 55:45

### 2.3.1 Internal Programmable Sample Clock Oscillator

The internal programmable clock oscillator (Silicon Labs SI570) is a low-jitter oscillator and can be programmed to generate any output clock in the range from 10 MHz to 1.4 GHz. The default power up output frequency is set to 125 MHz.

The oscillator can be programmed with the register “Programmable ADC Clock I2C” (I<sup>2</sup>C serial port).

See sis3316\_class.h and sis3316\_class.cpp:

```
int sis3316_adc::set_frequency(int osc, unsigned char *values)
where
    osc = 0 : ADC Clock I2C register
    osc = 1 : MGT1 Clock I2C register (reserved)
    osc = 2 : MGT2 Clock I2C register (reserved)
    osc = 3 : DDR3 Clock I2C register (reserved)

    *values = 0x20 0xC2 0xBC 0x33 0xE4 0xF2 : 250 MHz
    *values = 0x21 0xC2 0xBC 0x33 0xE4 0xF2 : 125 MHz
    ....
```

**Note:** after setting, the clock will be stable after 10ms

### 2.3.2 External NIM In Programmable Precision Clock Multiplier

The input frequency of the NIM input CI can either be fed directly to the ADC CLK Multiplexer and FP-Bus Out multiplexer by programming the programmable precision clock multiplier (SI5325) in Bypass mode or via the SI5325 PLL multiplier logic.

The input clock CKIN1 passes through the N3 input divider and is provided to the PLL. The input-to-output clock multiplication ratio is defined as follows:

$$f_{\text{OUT}} = f_{\text{IN}} \times N_2 / (N_1 \times N_3)$$

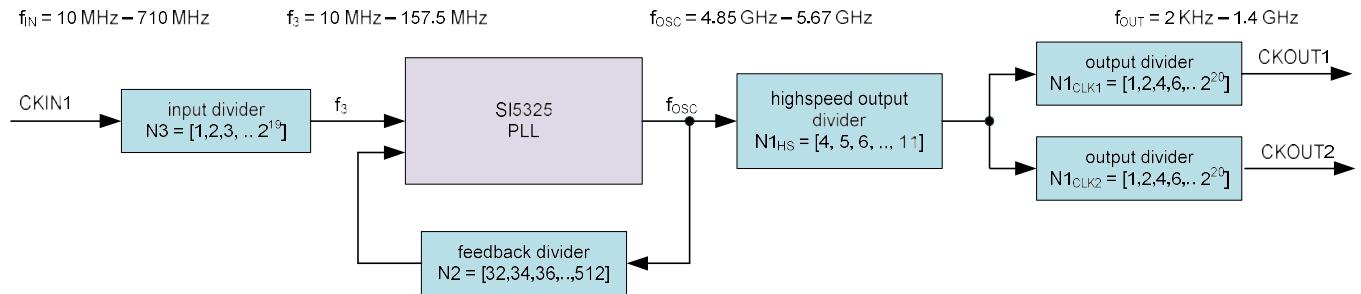
where:

N1 = output divider ( $N_{1\text{HS}} \times N_{1\text{CLKx}}$ )

N2 = feedback divider

N3 = input divider ( $N_{1\text{HS}} \times N_{1\text{CLKx}}$ )

Following parameters have to be met:



Examples:

CKIN1 in MHz	CKOUTx in MHz	Bw_sel	N1_HS	N1_CLKx	N2	N3	Input range in MHz	Output range in MHz
10	250	0	5	4	500	1	10,00 - 11,34	250 - 283
20	250	0 or 1	5	4	250	1	19,40 - 22,68	242 - 283
50	250	1 or 2	11	1	110	1	44,00 - 51,54	220 - 257
10	125	0	4	10	500	1	10,00 - 11,34	125 - 141
20	125	0 or 1	5	8	250	1	19,40 - 22,68	121 - 141
50	125	1 or 2	5	8	100	1	48,50 - 56,70	121 - 141

Example:

CKIN1 frequency: 10 MHz → met  $f_{IN}$   
 CHOOUT1 frequency: 250 MHz → met  $f_{OUT}$

$N3 = 1$  → met  $f_3$   
 $N2 = 500$  → met  $f_{OSC}$  (5 GHz)  
 $N1_{HS} = 5$   
 $N1_{CLKx} = 4$   
 $N1 = 20$  → met  $f_{OUT}$  (250 MHz)

See [www.silabs.com](http://www.silabs.com): SI5325 datasheet, SI53xxReferenceManual and use the PC-based software called DSPLLsim to compute proper parameter settings ([www.silabs.com/timing](http://www.silabs.com/timing)).

See sis3316\_class.h and sis3316\_class.cpp:

```

int sis3316_adc::get_status_external_clock_multiplier(unsigned int *status );
int sis3316_adc::bypass_external_clock_multiplier(void );

int sis3316_adc::set_external_clock_multiplier( unsigned int bw_sel,
                                                unsigned int n1_hs,
                                                unsigned int n1_clk1,
                                                unsigned int n1_clk2,
                                                unsigned int n2,
                                                unsigned int n3,
                                                unsigned int clkin1_mhz );
  
```

### 2.3.3 Sample Clock configuration sequence

For proper operation, the configuration of the Sample Clock demands a sequence of programming commands.

After Power Up or after any “Sample Clock” changes require the following sequence:

1. Issue a Key Register Reset command (disarms the sample logic, too)
2. Setup the ADC Sample Clock distribution logic via the “ADC Sample Clock distribution control” register
3. In case of use
 

Internal CLK:	program internal CLK oscillator and wait until the clock is stable (min. 10ms)
External VXS CLK In:	a valid stable clock must be present
External FP-Bus CLK In:	a valid stable clock must be present
External Lemo CLK In:	a valid stable clock must be present and program the Clock Multiplier (multiply factor or bypass) and wait until the clock is stable (max. 1200ms)
4. Issue a Key ADC Clock DCM/PLL Reset command.  
The DCM/PPL will be stable after max. 5ms.  
The internal status of the DCM/PLL Logic of the ADC FPGAs can be read via the “ADC FPGA Status” registers (bit 20).
5. Calibrate and configure the ADC FPGA input logic of the ADC data inputs via the “ADC Input tap delay” registers. The programmable tap delay depends on the sample frequency.  
See `sis3316_adc::configure_adc_fpga_iob_delays(unsigned int iob_delay)` in `sis3316_class.cpp`.

Example:

```

switch (clock_freq_choice) {
    case 0: iob_delay_value = 0x48 ; break; // 250000000.0
    case 1: iob_delay_value = 0x48 ; break; // 125000000.0
    case 2: iob_delay_value = 0x0 ; break; // 62500000.0
}

// Calibrate IOB-delay Logic Ch1 to Ch4
vme_A32D32_write (module_base_addr + SIS3316_ADC_CH1_4_INPUT_TAP_DELAY_REG, 0xf00 );
Sleep(1);

// set IOB-delay Ch1 to Ch4
vme_A32D32_write ( module_base_addr + SIS3316_ADC_CH1_4_INPUT_TAP_DELAY_REG,
                    0x300 + iob_delay_value );

```

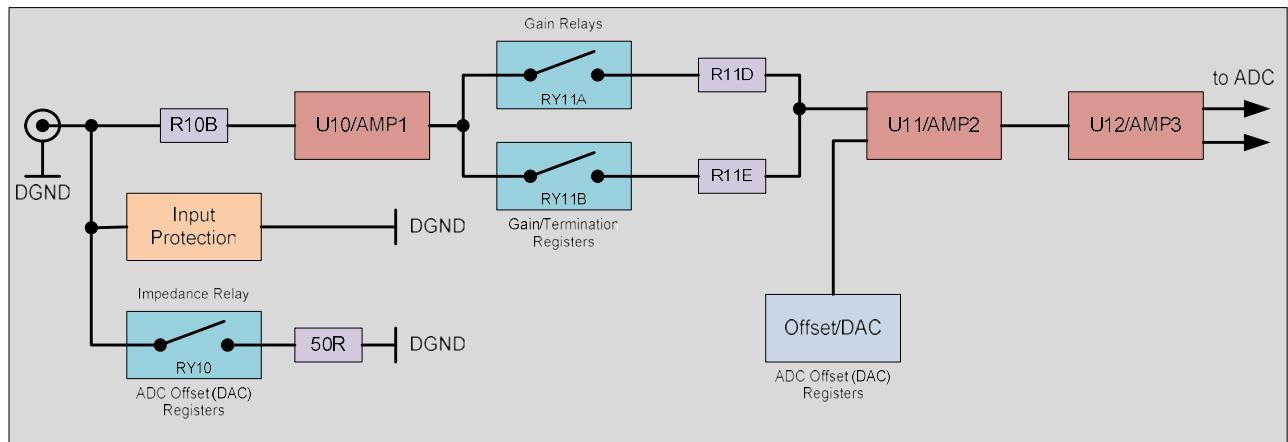
**Note:** `iob_delay_values` have changed for Firmware version with ADC-FPGA V0250-0004/V0125-0004 and higher.

6. Setup sample logic

## 2.4 Analog Input stage

The analog input stage allows you to select between two gains and  $50\ \Omega$  or high impedance ( $1K\ \Omega$ ) termination for each channel individually. Two input range settings are available and the selected range can be shifted full scale with an offset DAC for each channel individually, too.

A simplified diagram of the input stage is shown below (designators as of channel 1).



The gains and the termination of the channels are programmed via the “ADC Gain and Termination Control” registers. The Offset of the channels are programmed via the “ADC Offset (DAC) Control” register.

The Analog Input stage adapts the programmed Input Range to the “default” Input Range of the ADC chip. The Input Range of the ADC chip can be changed via the SPI Interface, also.

SIS3316-250MHz-14bit ADC chip “default” Input Range: 1.75 V

SIS3316-125MHz-16bit ADC chip “default” Input Range: 2.00 V

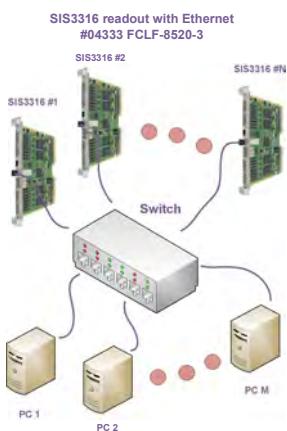
**Note:** the Input Range is 2% to 10% greater than specified

## 2.5 SFP cage

The Small Form factor Pluggable (SFP) media cage can be used to install a GBit Ethernet or an optical Fiber Channel link medium. Refer to section 10.6 (Accessories) for supported media types.

### 2.5.1 Ethernet SFP

The Ethernet SFP readout approach can be used both for standalone point to point readout (with a laptop e.g.) and for switch based readout of multiple units as illustrated below.



#### 2.5.1.1 Optical Ethernet SFP

Optical decoupling of data acquisition devices from the readout system is mandatory in some applications (like installation on a platform under high voltage). In case of the SIS3316(-DT) 125MSPS 16-bit and 250 MSPS 14-bit digitizers optical decoupling can be established with readout over an optical Ethernet connection.

Starting with VME FPGA firmware revision V3316-200A Switch **SW80-3** is used to activate/deactivate auto-negotiation for the Ethernet link. The switch has to be in on position for an optical connection and in off position for a copper connection or an optical connection over a media converter (like TP-LINK part number MC220L).

The tests were performed with a DELOCK PCI Express 1x SFP Slot Gigabit LAN card (DELOCK part number 89368), two SFP link media (TP-LINK part number TL-SM311LM or Finisar FTLF8524P2BNV, Struck part number 03145) and a standard LC-LC multimode duplex50/125 µm fiber.

### 2.5.2 Optical SFP

An optical point to point or ring type connection to a SIS1100e PCI Express optical link card can be used for non VME based readout as illustrated in the diagram below.  
This functionality requires yet to be developed custom firmware.



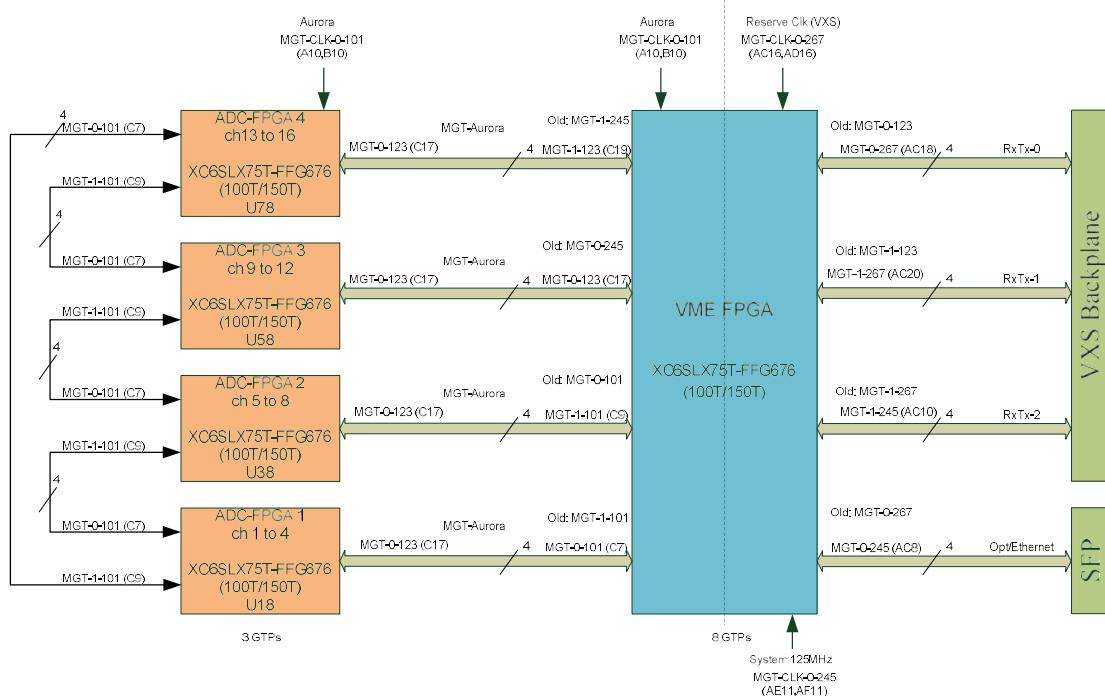
### 2.6 VXS

A VME extension for serial switching (VXS) implementation is prepared on the SIS3316. Actual use requires both a stuffing option (VXS connector, key and MGT clock) as well as a firmware implementation targeting a matching VXS switch module.  
The SIS3316 VXS preparation is supposed to be in accordance with the JLab-FADC250 card.

## 2.7 MGT Link Connections

The Multi Gigabit Transceivers (MGTs) are used for inter FPGA communication, to interface to the SFP and for the optional VXS implementation.

A simplified diagram of the MGT link connections is shown below.



## 2.8 Memory Handling

Each ADC FPGA group (4 channels) has physically two DDR3-memory-chips with 256 MByte each → 512 MByte for four channels → 128 MByte for the individual channel.

The AD FPGA Firmware stores ADC values in a defined format with additional information/features to these memories.

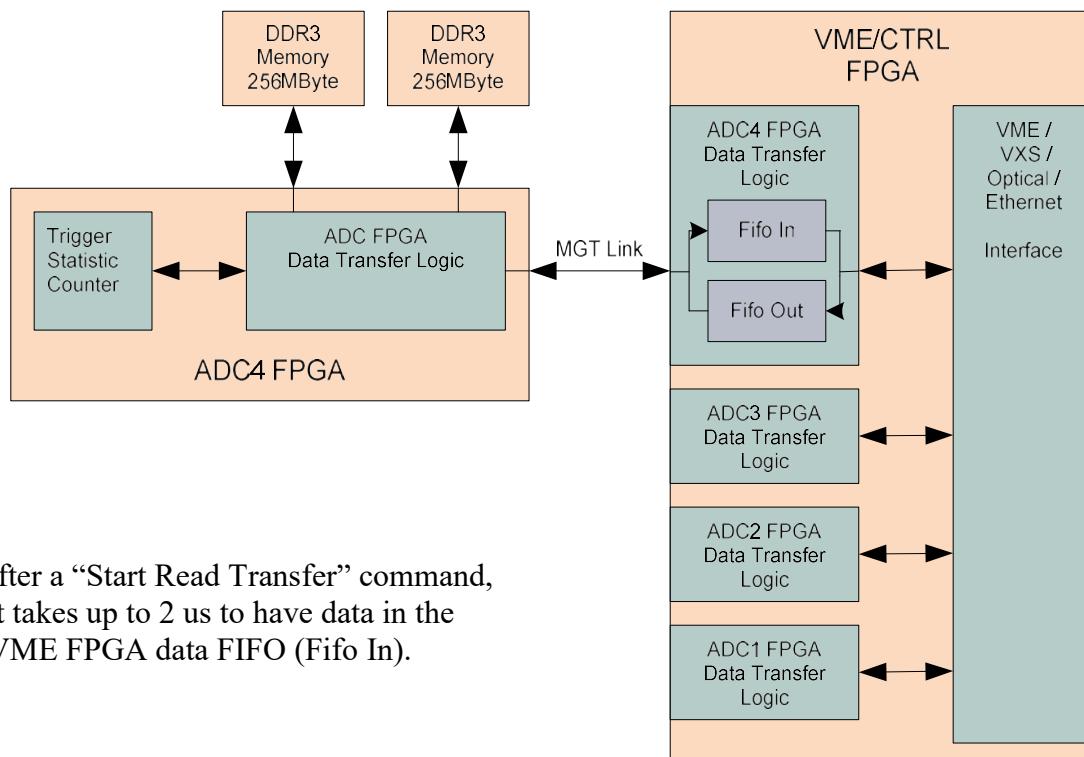
To read from and to write (test feature) to these memories a “Data Transfer” logic is implemented in the VME/CTRL FPGA to speed up the readout and to reduce the VME address space of the SIS3316.

The VME/CTRL FPGA has a “Data Transfer” logic for each ADC FPGA group. The logic consists of a data transfer FSM (State machine) and In- and Output FIFOs.

A read from the memory will be triggered by writing to the “ADC FPGA Data Transfer Control” register with a defined value. The logic will copy the data from the selected memory and addressed space via the MGT-Links to an internal FIFO in the VME/CTRL FPGA until it is ALMOST-Full. This internal FIFO can be read via VME with full speed. (option: read via Ethernet/Optical or VXS).. The FIFOs will be refilled automatically during the VME read.

A write to the memory will be triggered by writing a defined value to the “ADC FPGA Data Transfer Control” register. The logic will push the data from the internal FIFO (FIFO Out) to the selected memory in bursts of 64 32-bit words. This implies, that data have to be written in blocks of N x 64 32-bit words from the interface to the internal FIFO (FIFO Out).

See: “ADC FPGA Data Transfer Control” registers and Address Map Overview “ADC FPGA1 ch1-ch4 Memory Data FIFO”, ..... “ADC FPGA4 ch13-ch16 Memory Data FIFO”.

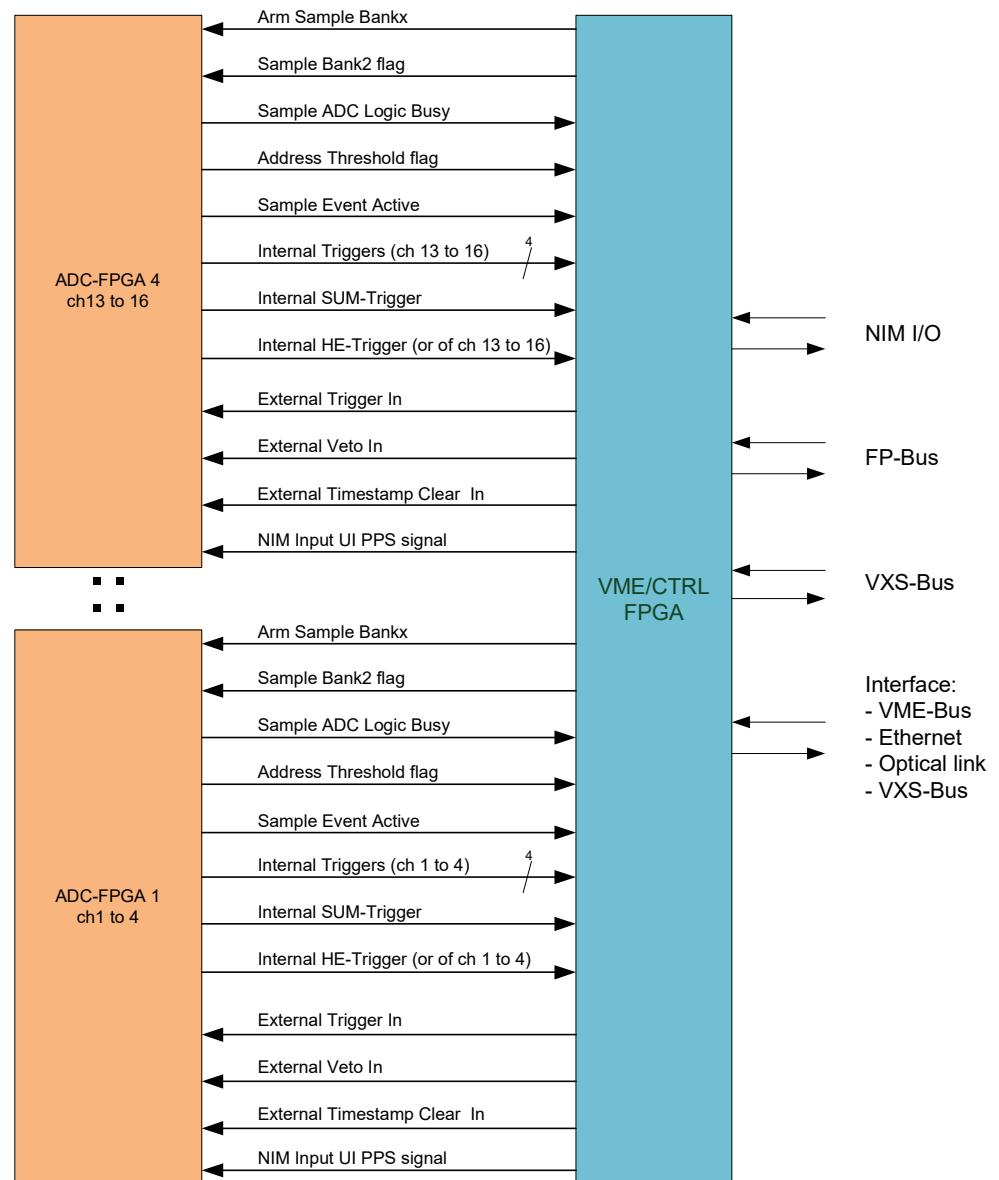


**Note:** after a “Start Read Transfer” command, it takes up to 2 us to have data in the VME FPGA data FIFO (Fifo In).

### 3 VME/CTRL FPGA Application Firmware

The implemented VME/CTRL FPGA firmware of the SIS3316 has the following features:

- double bank operation
- external (global) trigger generation/distribution
- external (global) veto generation/distribution
- external (global) timestamp-clear generation/distribution
- external (global) PPS pulse distribution (Pulse Per Second)



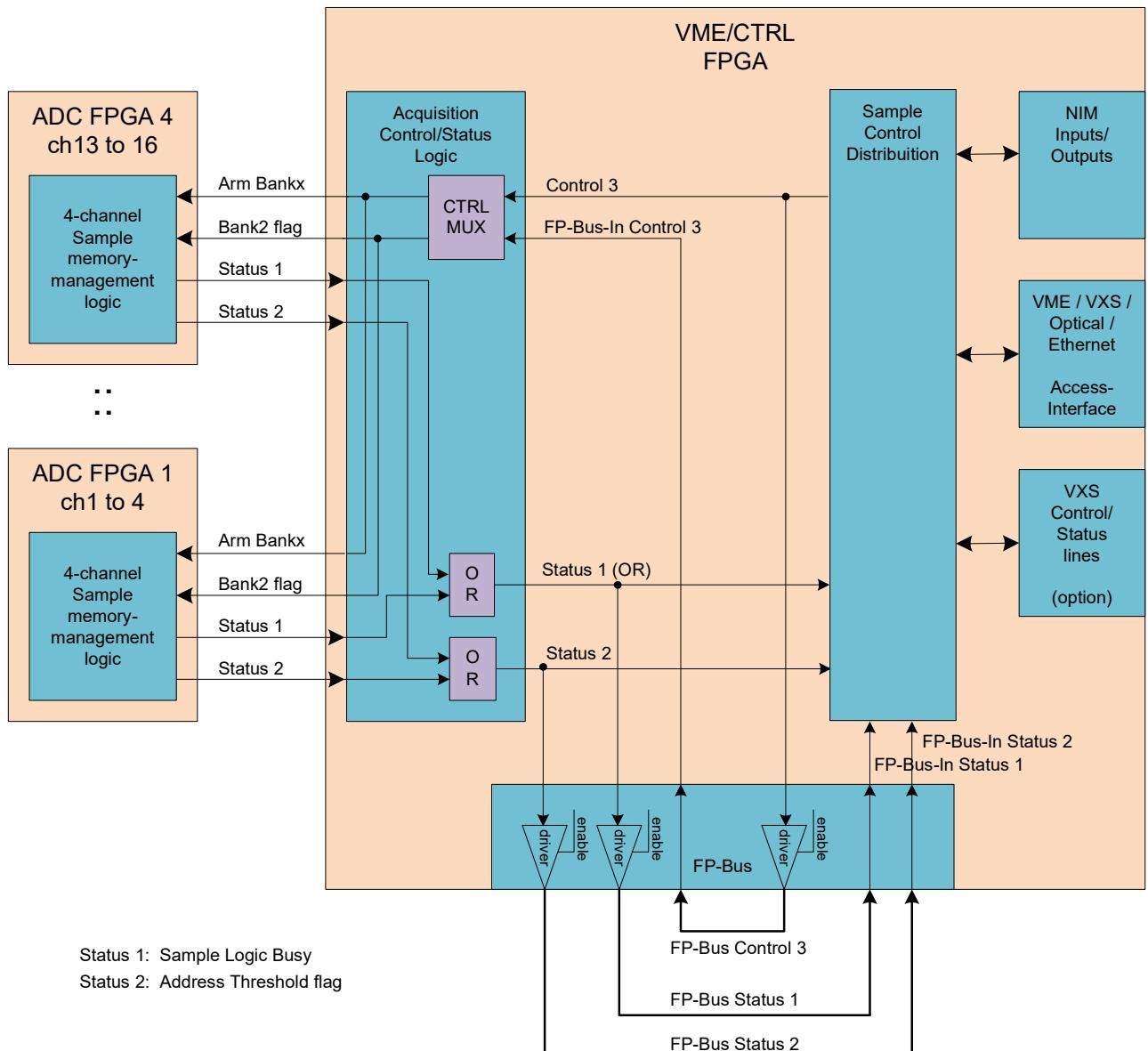
The following chapters describe the blocks:

- NIM Inputs/Outputs
- External (global) Signal Distribution
- Acquisition Control/Status
- Internal generated Trigger Features
- PPS logic

### 3.1 Sample Control overview

The block diagram illustrates the sampling control possibilities, respectively the handling of the memory management logic.

The ADC logic is ready to take (sample) hits/events if the ADC sample logic is armed. This will be done with “Arm Bank”. The “Bank2 flag” indicates to which memory address the hit/events will be saved (see chapter: Memory Organization). It is possible to set a “fill level mark” with the “address threshold” registers, As soon as a channel reaches the “fill level mark” it will set the “address threshold flag” (Status 1).



**How to control the Sample Control signals “Arm Bankx” and “Bank2 flag”:**

Note:

Interface write/read cycle means VME, UDP (Ethernet) or Optical Interface write/read cycle.

Multiple possibilities are supported:

- Interface write cycle to the SIS3316 base address + Key-address “Disarm Bankx and Arm Bank1/Bank2” (\*)
- VME write cycle to a defined broadcast address + Key-address “Disarm Bankx and Arm Bank1/Bank2” (\*\*)
- Interface write cycle to one “Master”-SIS3316 base address + Key-address “Disarm Bankx and Arm Bank1/Bank2” and using the FP-Bus (\*\*\*)
- External NIM signal feeds the SIS3316 (\*) if the “Sample Bank Swap Control with NIM Input TI/UI” Logic is enabled
- External NIM signal feeds one SIS3316 if the “Sample Bank Swap Control with NIM Input TI/UI” Logic is enabled and using the FP-Bus (\*\*\*)

**How to check the Sample Control signals “address threshold flag”:**

Multiple possibilities are supported:

- Interface read cycle from SIS3316 Acquisition register (\*)
- read VME IRQ Status or VME IRQ handling, provided IRQ registers are programmed (\*\*)
- Interface read cycle from one “Master”-SIS3316 Acquisition register and using the FP-Bus (\*\*\*)

(\*) use of one SIS3316

(\*\*) use of multiple SIS3316s in one VME Crate

(\*\*\*) use of multiple SIS3316s in one VME Crate use VME write cycle or Link Interface (Ethernet or Optical Link) write cycle

**Sample Control Flow controlled by Interface (read/write cycles):**

1. Disarm or Reset command
  2. Set address threshold registers
  3. Disarm active Bank and arm Bank2 command (start with the second bank)
- ```
do {
    4a. Poll on address threshold flag and wait until valid
        (also possible to wait for a defined time)
    5a. Disarm active Bank and arm Bank1 command
    6a. check if the active Bank is swapped (also possible to wait for defined time,
        for example the length of time of one event )
    7a. read "Previous Bank Sample address registers Ch1 to Ch16"
        (bit 24 will be cleared if the address corresponds to theBank1 and will be set
        if the address correspond to Bank2, means for checking whether the active
        bank was swapped already)
    8a. read sampled data from Ch1 to Ch16 (Memory Bank 2)

    4b. Poll on address threshold flag and wait until valid
        (also possible to wait for a defined time)
    5b. Disarm active Bank and arm Bank2 command
    6b. check if the active Bank is swapped
    7b. read "Previous Bank Sample address registers Ch1 to Ch16"
    8b. read sampled data from Ch1 to Ch16 (Memory Bank 1)
} (run == 1)

9. Disarm command
```

Respective to 7a/8a and 7b/8b refer to the routine below also:

```
int sis3316_adc::read_MBLT64_Channel_PreviousBankDataBuffer( . )
in ..\sis3316_class_library\sis3316_class.cpp
```

**Note 5ab / 6ab:**

With the command “Disarm active Bank and arm Bank1/2” the logic will disarm the active Bank to suppress a new start of capturing Hit/Events. The capturing of Hits/Events at this moment will be continued. The logic waits to arm the Bank1/2 (alternate) until “all channels are not busy”.

**Sample Control Flow controlled by NIM input signals:**

1. Disarm or Reset command
2. Enable “Sample Bank Swap Control with NIM Input TI/UI” Logic command.  
The Sample Bank Logic will be armed on Bank 1 with the next NIM input signal.  
The logic will toggle the active Bank with each following NIM input pulse.

```

do {
    3a. Poll on Sample Logic Armed and Bank 2 flags and wait until Armed on
        Bank 2 is valid (acquisition control register: bit 16 = 1 and bit 17 = 1)
    4a. read “Previous Bank Sample address registers Ch1 to Ch16”
        (bit 24 gives means for checking whether the active
        bank was swapped already. It will be cleared if the address corresponds to
        Bank1 and will be set if the address corresponds to Bank2,)
    5a. read sampled data from Ch1 to Ch16 (Memory Bank 1)

    3b. Poll on Sample Logic Armed and Bank 2 flags and wait until Armed on
        Bank 1 is valid (acquisition control register: bit 16 = 1 and bit 17 = 0)
    4b. read “Previous Bank Sample address registers Ch1 to Ch16”
        (bit 24 gives means for checking whether the active
        bank was swapped already. It will be cleared if the address corresponds to
        Bank1 and will be set if the address corresponds to Bank2,)
    5b. read sampled data from Ch1 to Ch16 (Memory Bank 2)

} (run == 1)

```

6. Disarm command

Respective to 4a/5a and 4b/5b refer to the routine below also:

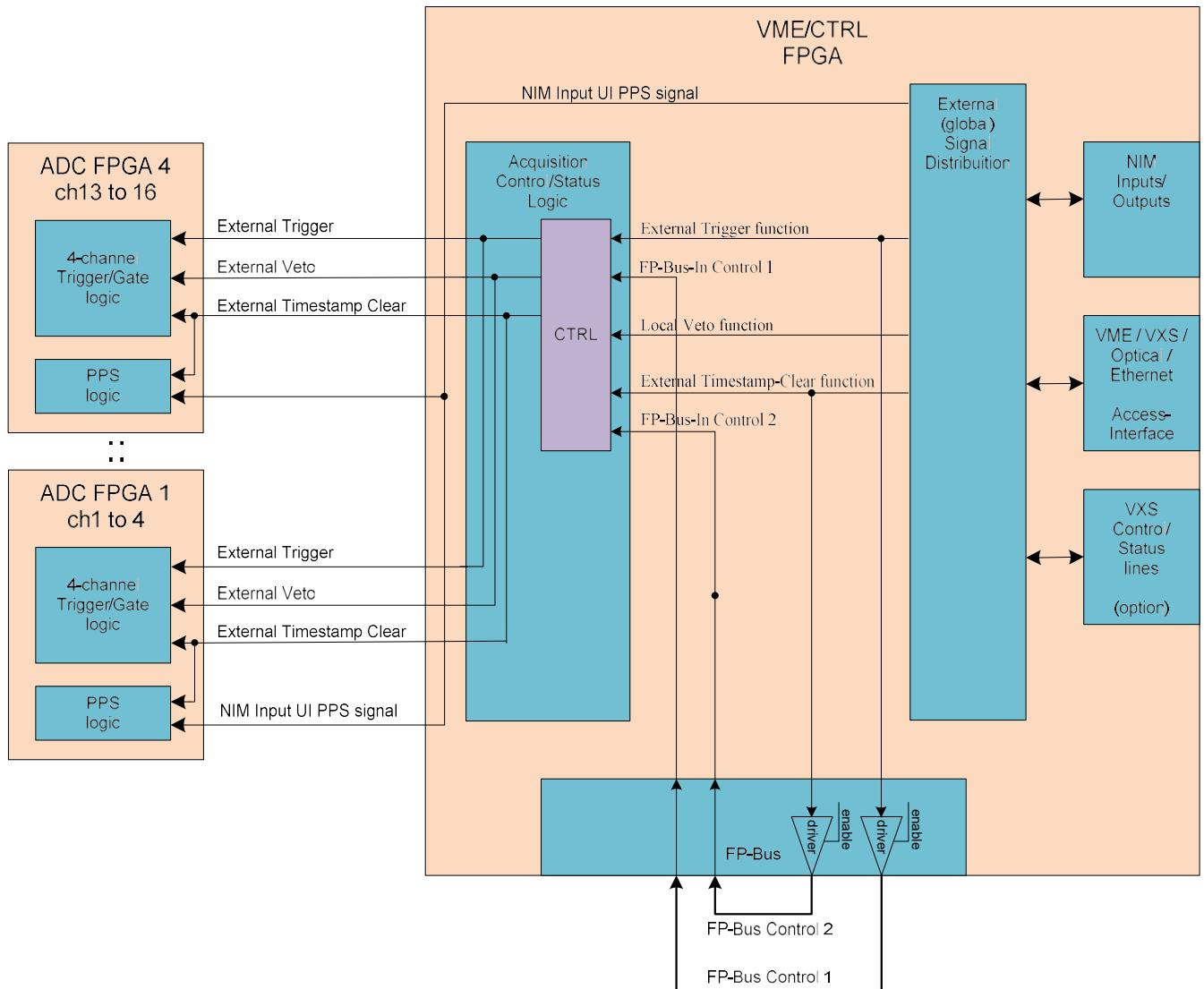
```
int sis3316_adc::read_MBLT64_Channel_PreviousBankDataBuffer( . )
in ..\sis3316_class_library\sis3316_class.cpp
```

**Note:**

The user has to care for proper timing between bank swapping (external NIM signal) and readout of the **not** active Bank.

### 3.2 External (global) Signal overview

The “External (global) signals” consist of the signals “External Trigger”, “External Veto”, “External Timestamp Clear” and “NIM Input UI as PPS signal” which routed the from the VME/CTRL FPGA to all ADC FPGAs. These signals are synchronized with the sample clock.



### 3.3 External Sample Control I/Os

The SIS3316 has several in- and outputs (NIM, front panel LVDS Bus and VXS Bus) which can be used for data sampling control. The following description depends to some extend on the given FPGA firmware implementation.

#### 3.3.1 External NIM

##### 3.3.1.1 NIM Outputs

Three NIM outputs are available to supply information/status like “internal generated triggers”, “sample busy”, “sample clock”, ..... .

With the registers **LEMO Out “CO” Select**, **LEMO Out “TO” Select** and **LEMO Out “UO” Select** it is possible to select/choose the output function of the three LEMO outputs.

**Note:** the clock CO output should be regarded more as a monitor feature to check the clock on a scope. Especially on 250 MSPS units it is recommended to use the front panel bus to distribute the sample clock.

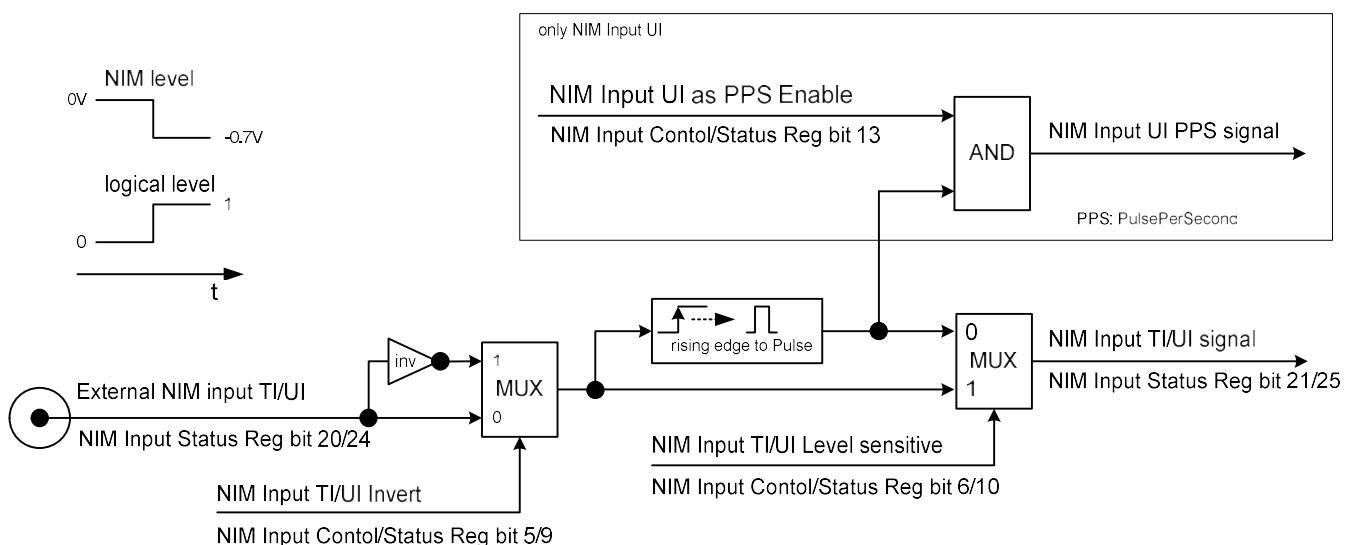
##### 3.3.1.2 NIM Inputs

Three NIM Inputs are available to feed external control signals to the SIS3316 card.

The NIM Input “CI” is designed to use an external sample clock but it is possible to use this input as a “control signal” also, provided the SIS3316 Hardware Version is V2 or higher. At the moment, no control function is assigned to it.

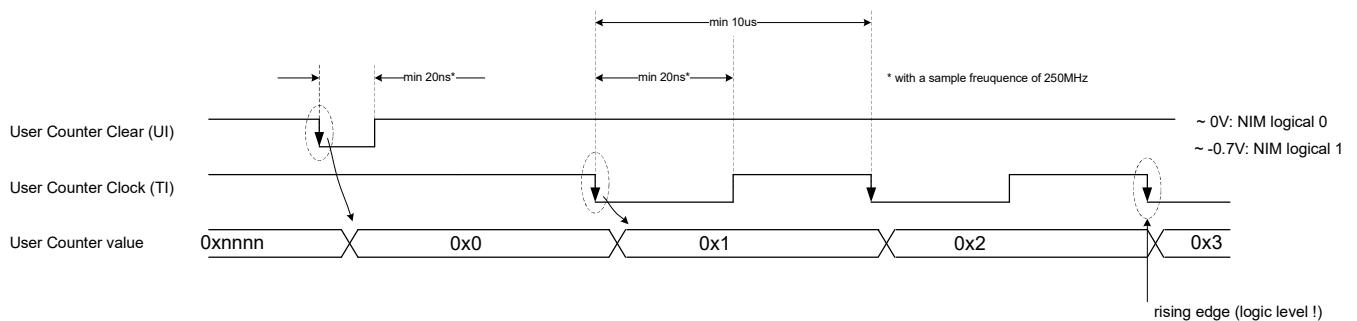
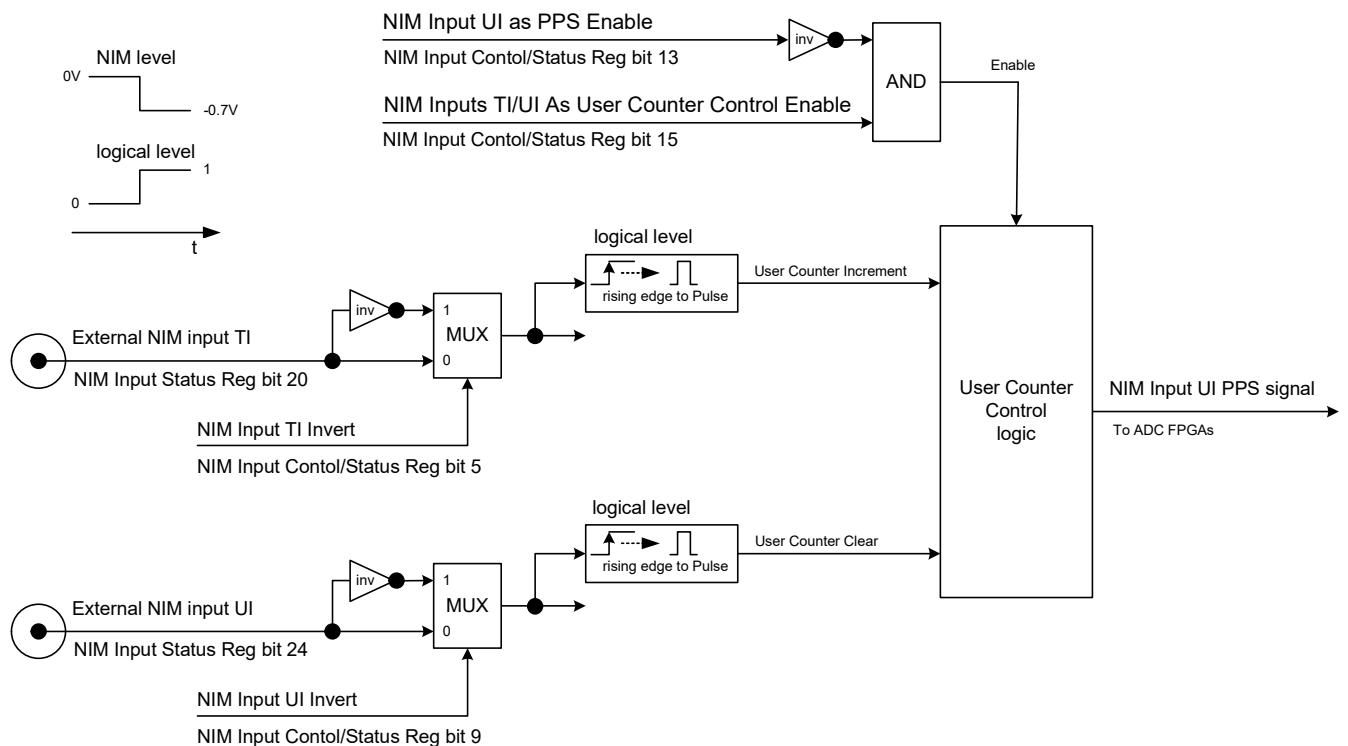
See: Sample Clock distribution.

The input logic behind the inputs “TI” and “UI” allows for signal inversion of the external NIM input signals as well as for the generation of “edge sensitive” behaviour.



### 3.3.1.2.1 Special use of NIM Inputs TI/UI for using the “User Counter” logic

The Inputs TI (increment) and UI (clear) can be used for the “User Counter” logic. To enable this feature the bit 15 of the “NIM Input Control” register has to be set and the bit 13 (PPS enable) has to be cleared.



### 3.3.2 Front panel LVDS Bus

The front panel LVDS Bus (FP-Bus) can be used for synchronized operation of multiple SIS3316s. In addition to the sample clock distribution (see Sample Clock distribution), the following signals are distributed on the FP-Bus, also:

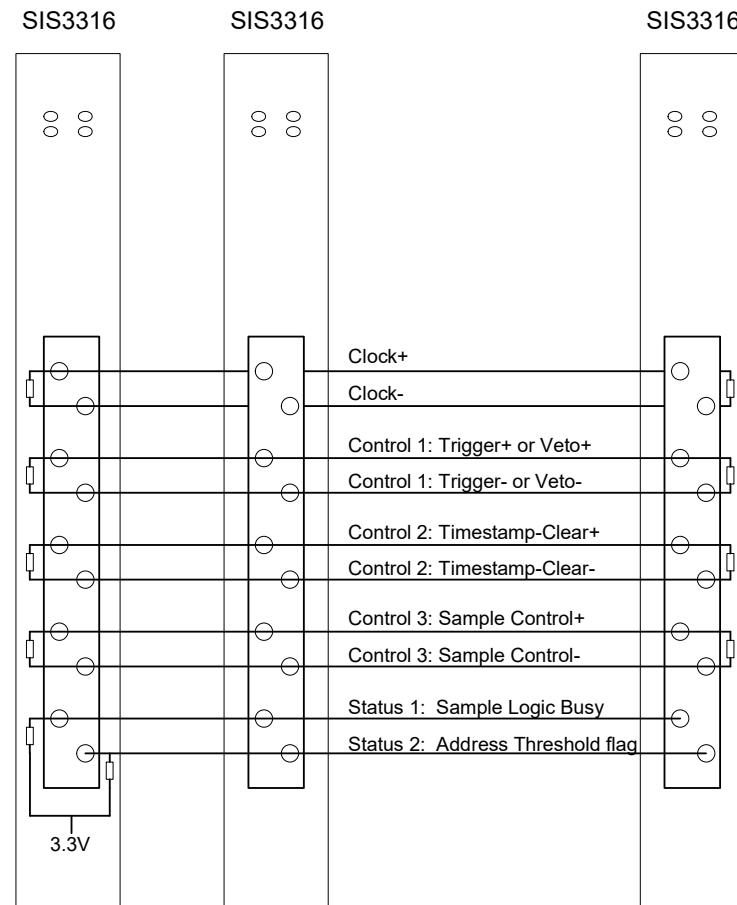
- Control 1: Global Trigger or Global Veto
- Control 2: Timestamp Clear
- Control 3: Sample Control (Disarm , Disarm Bankx and Arm Bank1 or 2)
- Status 1\*: Sample Logic busy
- Status 2: Address Threshold flag

\* TTL open-collector signals

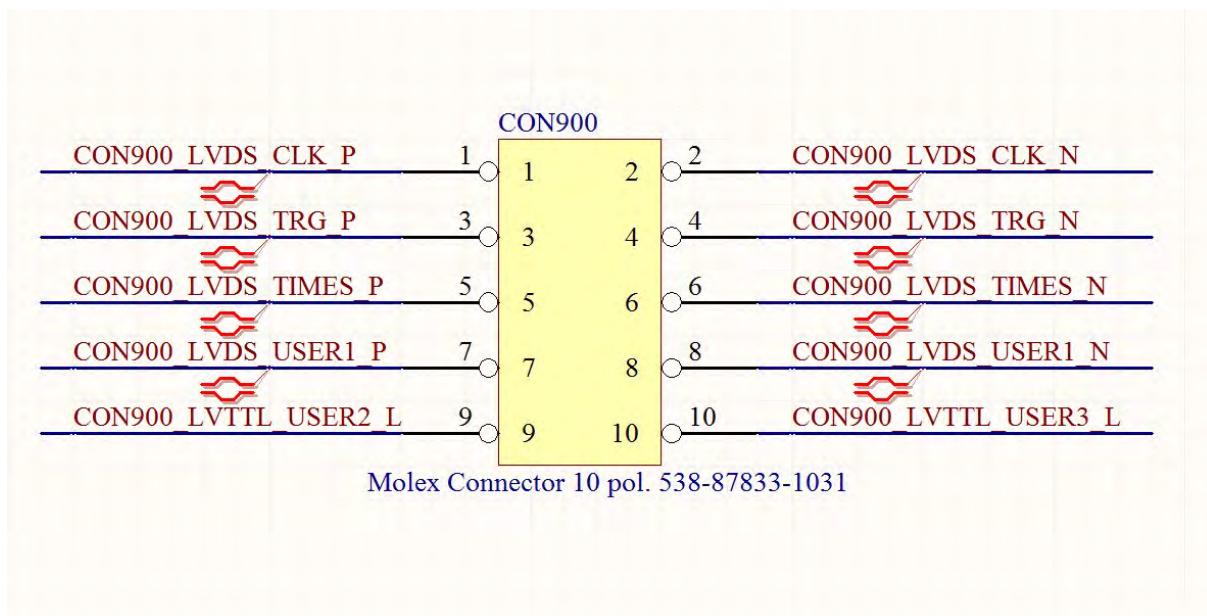
The control signals can be accepted by each SIS3316 and driven from each SIS3316 also. An “enable” bit has to be set in the FP-Bus control register to drive a control signal. One and only one SIS3316 has to drive the control signal(s).

The status signals can be accepted (Acquisition control/status register) and “set by condition” (as enabled) from each SIS3316. The status signals are “ored” on the FP-Bus.

Operation:    - enable Clock and Control only on one SIS3316 (left side, Master SIS3316)  
                   - enable Status on all SIS3316s

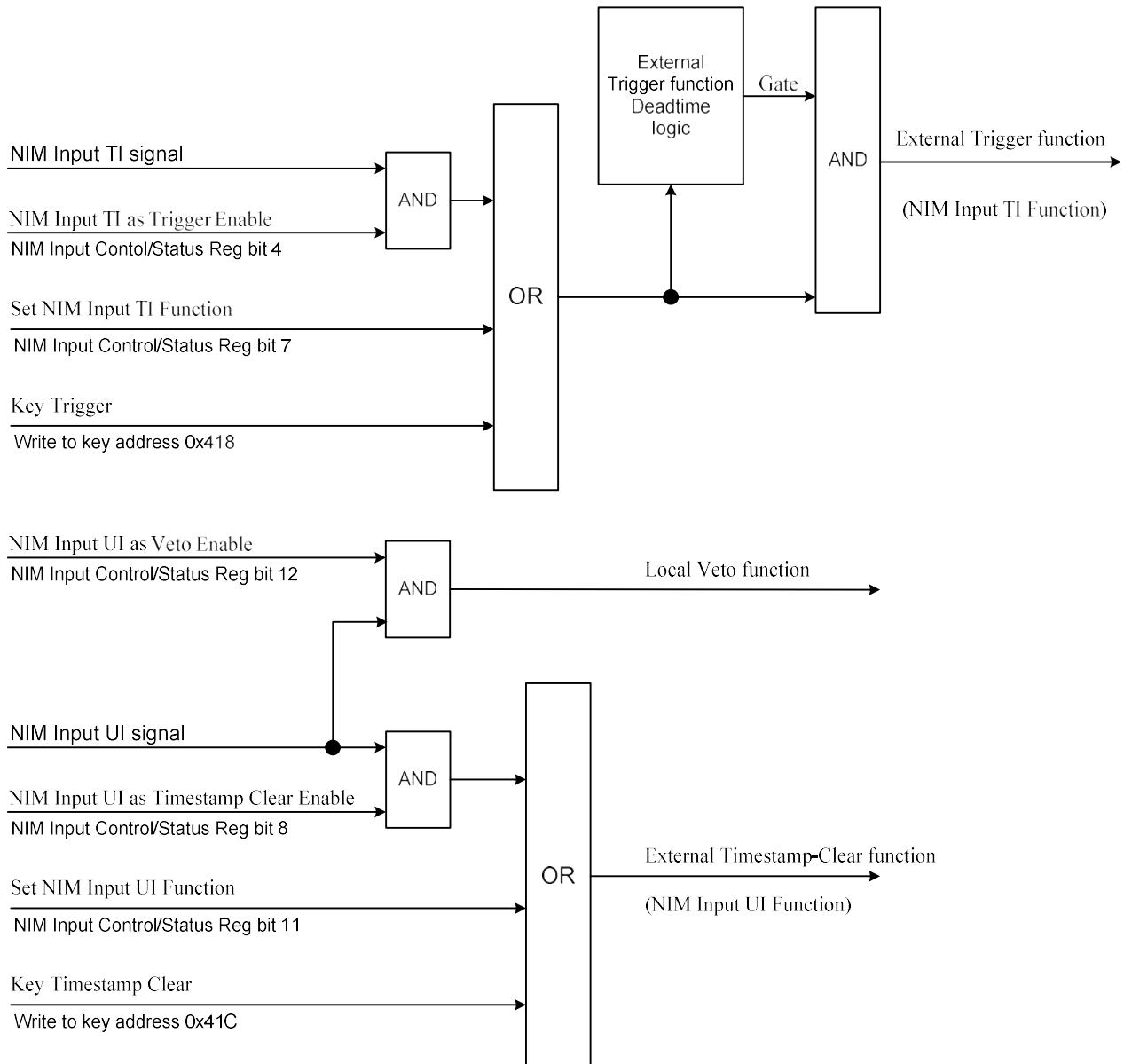


LVDS Bus connector:



### 3.4 External (global) input signal distribution

The SIS3316 has several In- and Outputs (NIM, front panel LVDS Bus and VXS Bus) which can be used to control the hit/event saving logic. The following description of the implementation depends on actual the FPGA Firmware.

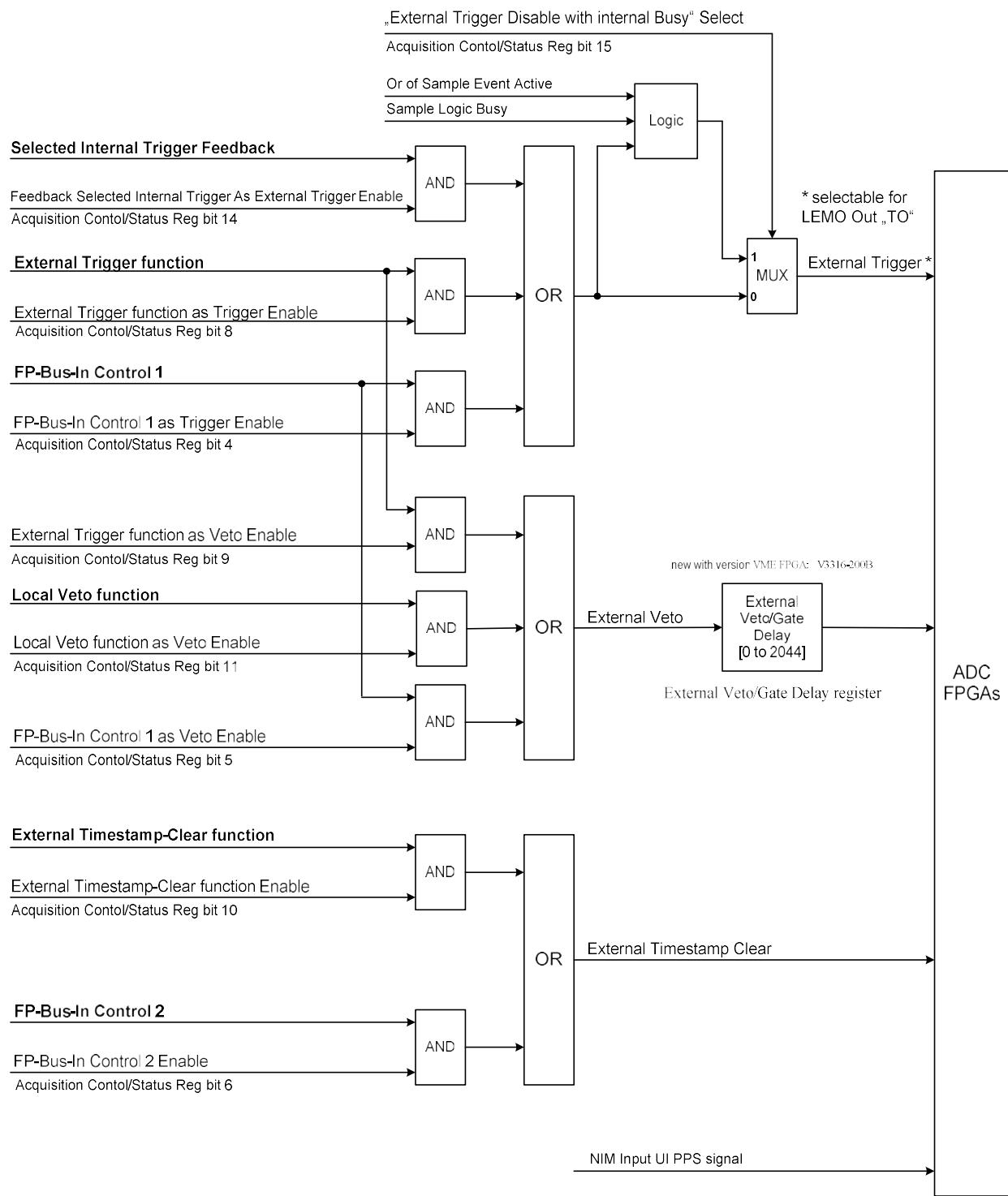


Note: External Trigger function Deadtime logic is new with VME FPGA Version V\_33160010

### 3.5 Acquisition Control

The block diagram below illustrates the possibilities of the generation of “External Trigger”, “External Veto” and “External Timestamp Clear”.

The „External Trigger Disable with internal Busy“ logic prevents External Triggers in case that not all 16 channels are ready to save Hits/Events.



### **3.6 Internally generated Trigger Features**

The 16 internally generated triggers from the ADC FPGAs (channels 1 to 16) are used for:

- Internal Trigger Counters
- Distribute to LEMO outputs
- Feedback Selected Internal Trigger as “External Trigger” to the ADC FPGAs
- Trigger Coincidence Lookup table

#### **3.6.1 Internal Trigger Counters**

16 Internal Trigger counters are available in the VME FPGA to count the internal (stretched) triggers of the individual channels. They can be cleared with a “Timestamp clear” command simultaneously. Each counter will stop incrementing upon reaching 0xFFFFFFFF.

#### **3.6.2 Distribute to LEMO outputs**

The internally generated triggers can be routed to the LEMO Outputs “TO” and “UO” by programming the two registers:

- LEMO Out “TO” Select register
- LEMO Out “UO” Select register

#### **3.6.3 Feedback Selected Internal Trigger as “External Trigger” to the ADC FPGAs**

The internally generated triggers and the Coincidence validation signal of table 1 can be fed back as “External Trigger” to the ADC FPGAs. This feature has to be enabled (Acquisition Control/Status register bit 14) and the source has to be selected with the “Internal Trigger Feedback Select” register.

### 3.6.4 Trigger Coincidence Lookup table

Two Lookup tables are implemented to generate Coincidence validation signals of each channel trigger combinations.

They have 65536 (0x10000) entries. Each table entry corresponds to one trigger combination. The Lookup table has to be initialized with the required/requested Coincidence truth table. The initialization is done by writing to the Lookup table registers:

- Trigger Coincidence Lookup table Control register
- Trigger Coincidence Lookup table Address register
- Trigger Coincidence Lookup table Data register

The Coincidence validation signal of table 1 can be routed (selected) to the LEMO output “TO” and to the “Feedback as External Trigger” feature for the ADC FPGAs.

The Coincidence validation signal of table 2 can be routed (selected) to the LEMO output “UO”.

**Example:** A Coincidence validation signal should be generated if the following channel trigger combination is valid inside a Coincidence window of 200ns:

```
Coincidence validation signal = 1
if ((ch 10 or ch 9) and
    (ch 8 or ch 7 or ch 6 or ch 5 or ch 4 or ch 3 or ch 2 or ch 1))
```

| channel |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Validation signal      | table address |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------|---------------|
| 16      | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  |    |                        |               |
| 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0x0000                 |               |
|         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                        |               |
|         |    |    |    | x  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0x0100                 |               |
|         |    |    |    | x  | 1  | x  | x  | x  | x  | x  | x  | x  | x  | x  | 1  | 1  | 0x0101<br>to<br>0x01FE |               |
|         |    |    |    | x  | 1  | x  | x  | x  | x  | x  | x  | x  | x  | 1  | x  | 1  |                        |               |
|         |    |    |    | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |                        |               |
|         |    |    |    | x  | 1  | 1  | x  | x  | x  | x  | x  | x  | x  | x  | x  | 1  |                        |               |
|         |    |    |    | x  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0x01FF                 |               |
|         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                        |               |
|         |    |    |    | 1  | x  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0x0200                 |               |
|         |    |    |    | 1  | x  | x  | x  | x  | x  | x  | x  | x  | x  | x  | 1  | 1  | 0x0201<br>to<br>0x02FE |               |
|         |    |    |    | 1  | x  | x  | x  | x  | x  | x  | x  | x  | x  | 1  | x  | 1  |                        |               |
|         |    |    |    | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |                        |               |
|         |    |    |    | 1  | x  | 1  | x  | x  | x  | x  | x  | x  | x  | x  | x  | 1  |                        |               |
|         |    |    |    | 1  | x  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0x02FF                 |               |
|         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                        |               |

| channel |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Validation signal | table address |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|---------------|
| 16      | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  |                   |               |
|         |    |    |    |    |    | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                 | 0x0300        |
|         |    |    |    |    |    | 1  | 1  | x  | x  | x  | x  | x  | x  | x  | 1  | 1                 | 0x0301        |
|         |    |    |    |    |    | 1  | 1  | x  | x  | x  | x  | x  | x  | 1  | x  | 1                 | to<br>0x03FE  |
|         |    |    |    |    |    | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | ..                |               |
|         |    |    |    |    |    | 1  | 1  | 1  | x  | x  | x  | x  | x  | x  | x  | 1                 | 0x03FF        |
|         |    |    |    |    |    | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1                 |               |
|         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                   |               |

1. Set Trigger Pulse Length to 200ns for all channels (FIR Trigger Setup registers)  
(configure FIR Trigger Setup)
2. Clear Lookup tables and set pulse length  

```
data = 0x80000000; // clear command ;
data = data + 0xf; // pulse length 16 sample clocks (0x10 - 1) ;
Trigger Coincidence Lookup table Control register = data;
```
3. Initialize Lookup tables:  

```
data = 1; // Lookup Table 1 Coincidence validation bit ;
//data = 2; // Lookup Table 2 Coincidence validation bit ;
//data = 3; // Lookup Table 1 and 2 Coincidence validation bits ;
address = 0x101 ;
Trigger Coincidence Lookup table Address register = address ;
for (i=0x001; i<0x0FF; i++) { // writes from address 0x101 to 0x1FF
    Trigger Coincidence Lookup table Data register = data ; // address will increment
}
address = 0x201 ;
Trigger Coincidence Lookup table Address register = address ;
for (i=0x001; i<0x0FF; i++) { // writes from address 0x201 to 0x2FF
    Trigger Coincidence Lookup table Data register = data ; // address will increment
}
address = 0x301 ;
Trigger Coincidence Lookup table Address register = address ;
for (i=0x001; i<0x0FF; i++) { // writes from address 0x301 to 0x3FF
    Trigger Coincidence Lookup table Data register = data ; // address will increment
}
```
4. Set Channel Trigger Mask:  

```
address = 0x03FF0000 ; // use only Trigger lines from channel 1 to 10
Trigger Coincidence Lookup table Address register = address ;
```

### 3.7 PPS logic

The PPS logic (Pulse Per Second) enables the feature to latch the timestamp in each ADC FPGA upon an external signal.

At the rising edge of the LEMO input UI, provided that the “NIM UI UI as PPS” is enabled (NIM Input control/status register bit 13), the 48-bit timestamps are latched in each ADC-FPGA and a “PPS latch bit” (Acquisition control/status register bit 23) is set to indicate that new 48-bit PPS-timestamp values are available in the ADC FPGAs.

The four latched PPS-timestamps have to be identical, provided a “Timestamp clear” command was issued at least once before. Therefore, it is only necessary to read one of them. The “PPS latch bit” can be cleared with a Key command (*PPS\_Latch\_Bit\_clear*).

### 3.8 Prescaler of the sampling clock with (possible) output to UO

A 32-bit prescaler logic for the sampling clock with possible output to UO is implemented.

The “prescaler output pulse divider” register allows you to prescale the sampling clock. In combination with the “prescaler output pulse length” and “LEMO Out UO Select” registers it is possible to generate a pulse at LEMO output UO. If the register value is unequal 0 (logic is not disabled) the prescaler logic will start with a Timestamp clear command (for example with a write to the Key address `SIS3316_KEY_TIMESTAMP_CLR` with bit 10 of the Acquisition register set).

## 4 ADC FPGA Application Firmware (Neutron/Gamma, Gamma)

The implemented FPGA firmware of the SIS3316 has the following features:

- Double Bank / Multi event operation
- 16 channel asynchronous (internal trigger generation) and synchronous (external trigger, global trigger) operation
- Internal Trigger generation for each channel (FIR filter with CFD feature, baseline independent)
- Internal “Quad Sum of four channels” Trigger generation (FIR Filter with CFD feature, baseline independent)
- Internal Gate generation for coincidence operations
- 8 Accumulators/Integrals (1 x 24-bit, 7 x 28-bit) for each channel
- Peak high finder
- 48-bit Timestamp
- 3 MAW (or MV: mowing window) values from FIR Trigger trapezoidal for higher timestamp resolution (up to 50ps)
- Flexible Hit/Event storage (additional raw data, FIR Trigger or FIR Energy trapezoidal MAW values)

New with Firmware ADC-FPGA V0250-0004/V0125-0004:

- Energy Filter (second FIR Filter)
- Internal Energy Histogram (up to 64K Histogram/each channel)

New with Firmware ADC-FPGA V0125-0004 (SIS3316-16bit, only)

- Average Mode

New with Firmware ADC-FPGA V0250-0005 (SIS3316-14bit, only)

- Peak/Charge Mode with external Gate

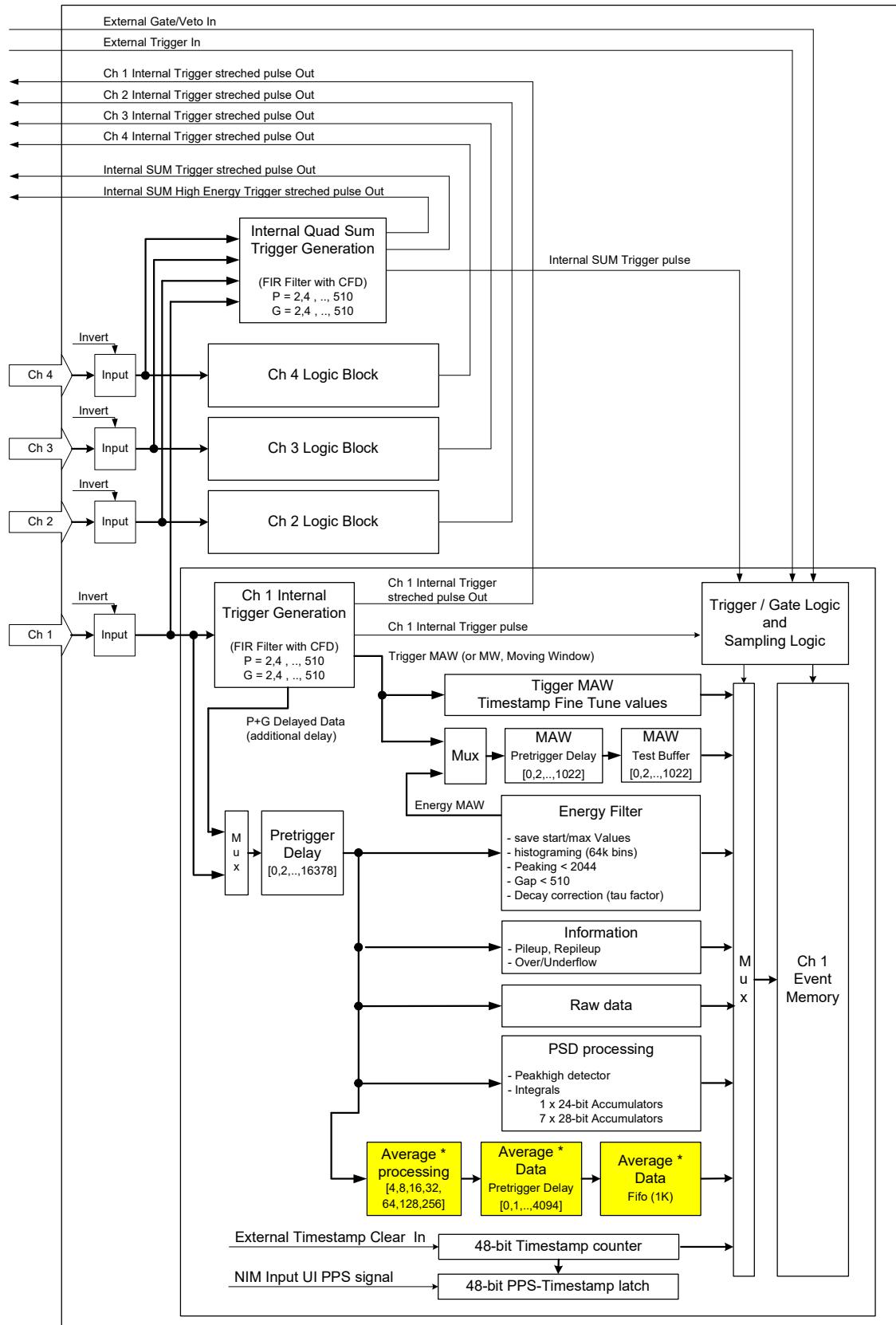
New with Firmware ADC-FPGA V0125-0007 and V0250-0007

- Increase “Raw Data Sample Length” from 64K to 32M samples
- Increase “Pre Trigger Delay” from 2K to 16K

New with Firmware ADC-FPGA V0125-0008 and V0250-0008

- PPS (Pulse Per Second) timestamp latch logic

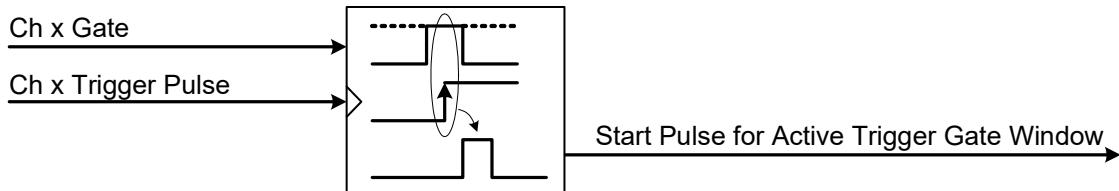
#### 4.1 Four Channel group blockdiagram



\* only SIS3316-125MHz-16bit

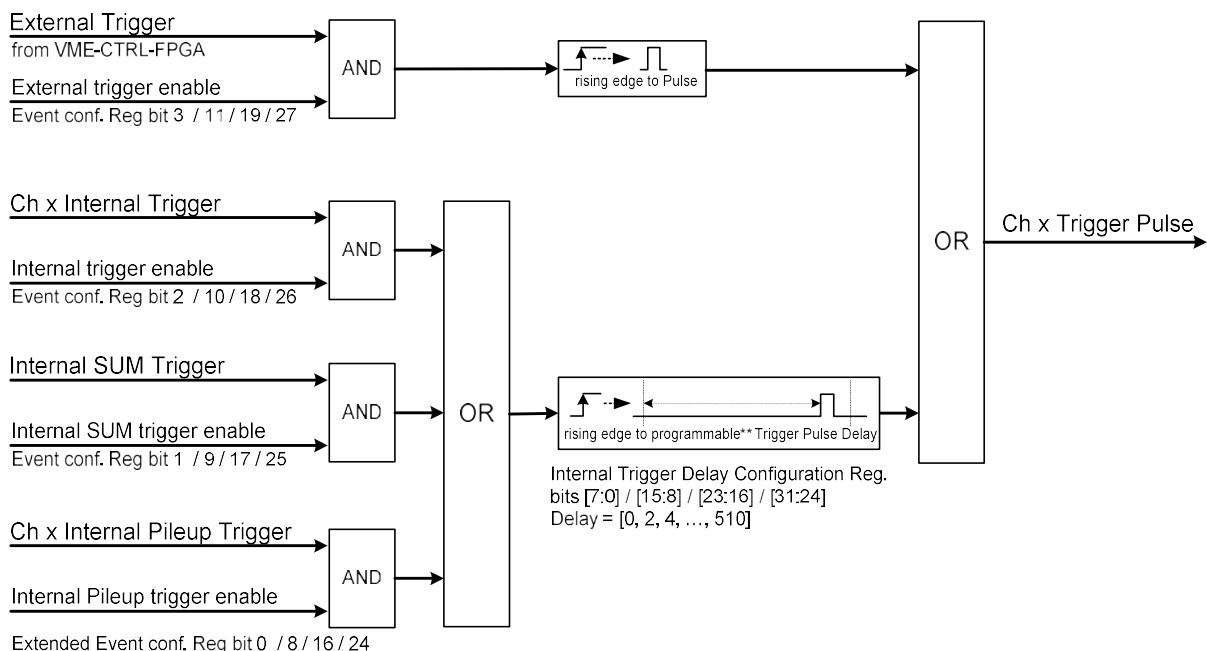
## 4.2 Trigger/Gate Logic

Trigger / Gate logic illustration:

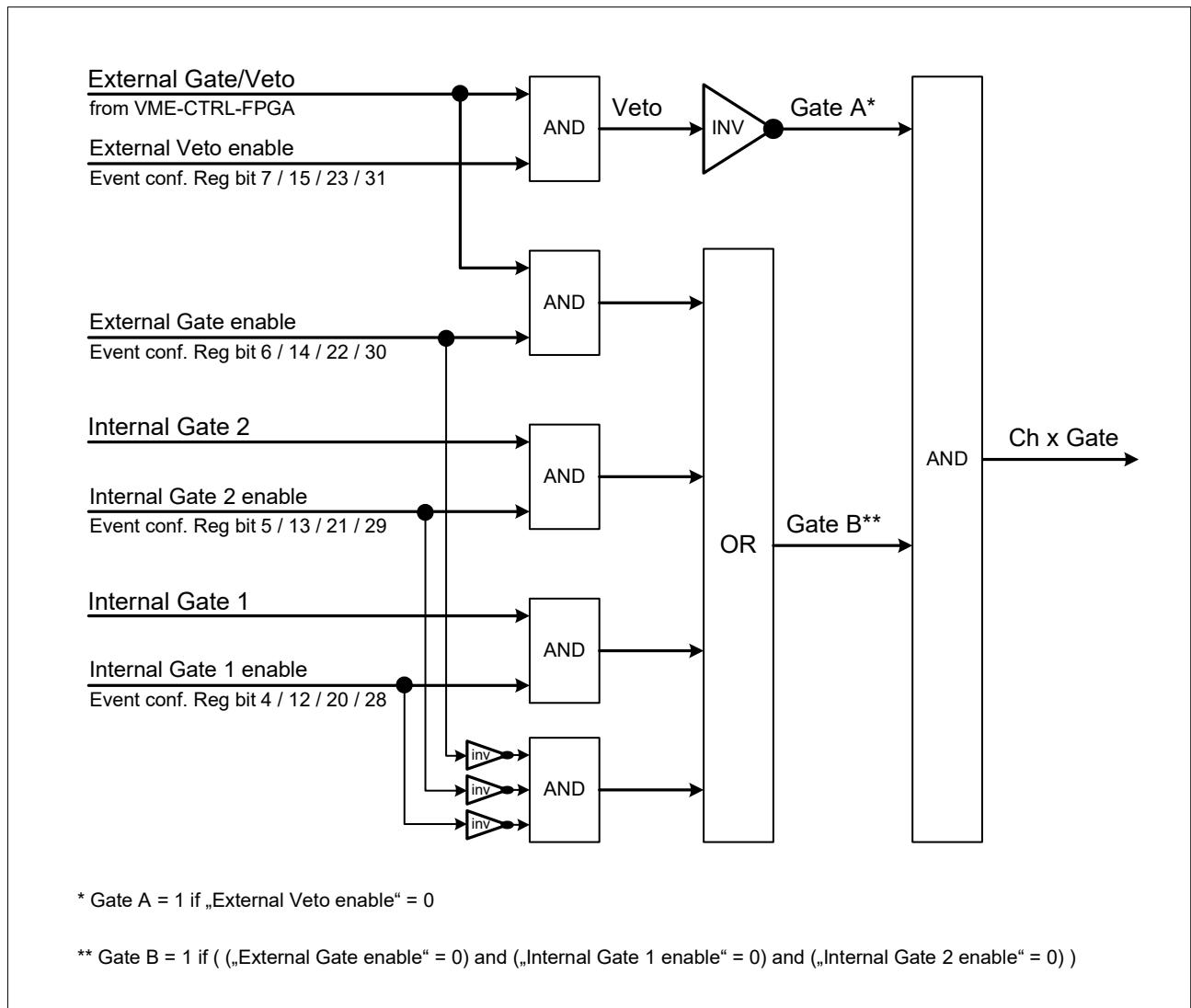


A "Ch x Trigger Pulse" with a valid "Ch x Gate" will generate a "trigger" pulse which starts the "Active Trigger Gate Window" or rather start the sampling of an Event/Hit.

Ch x Trigger logic illustration:

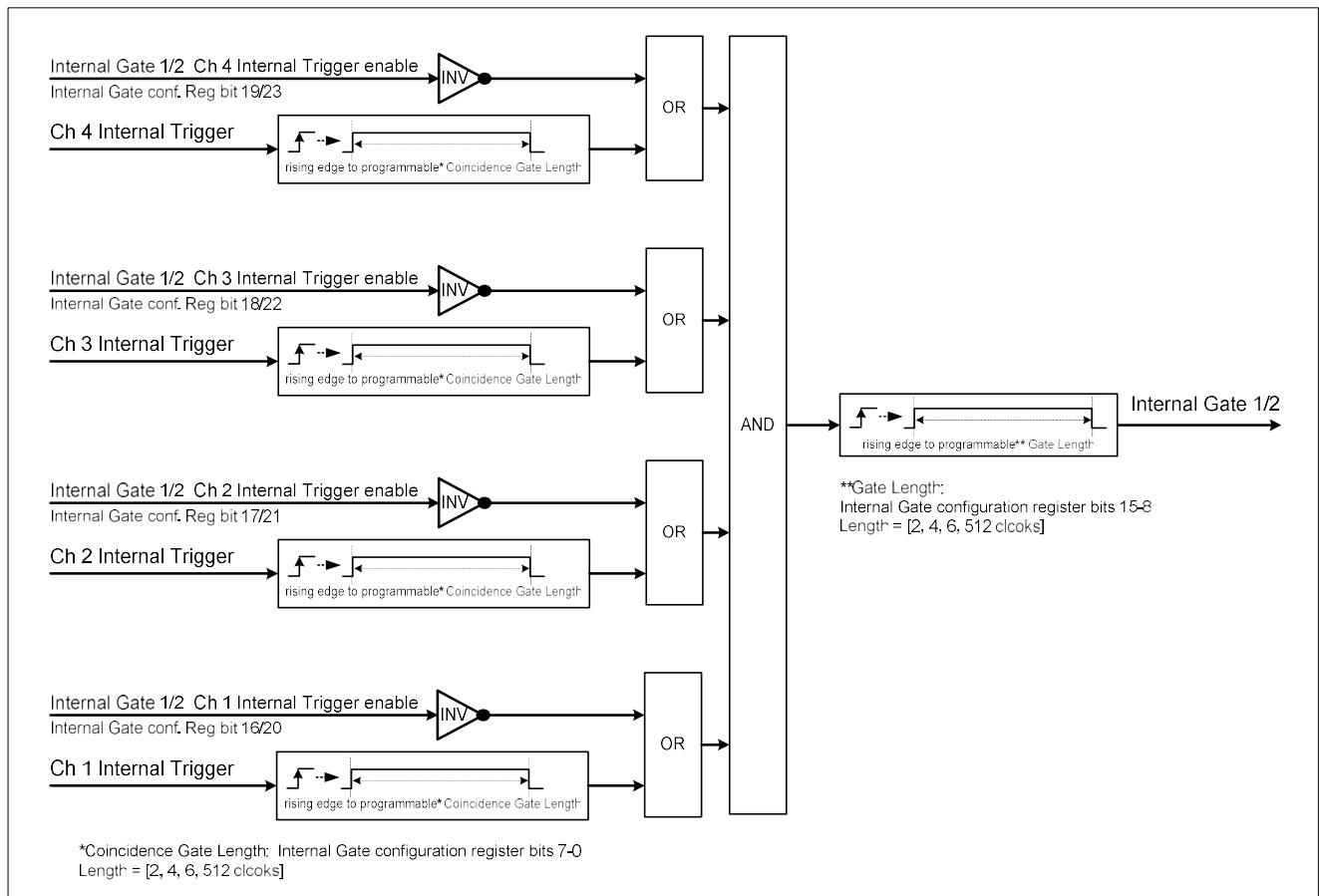


Ch x Gate logic illustration:



The internal Gates 1 and 2 can be derived from the internal triggers to build coincidence windows.

#### 4.2.1 Coincidence Gate Logic inside a four channel group

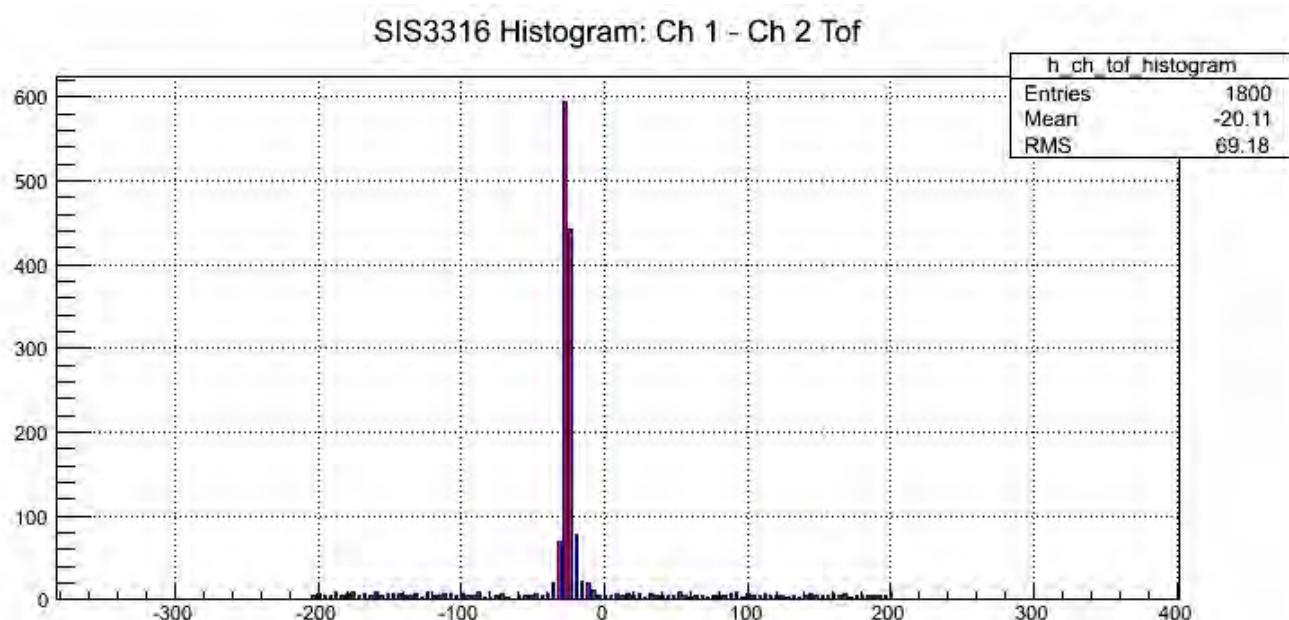


**Example:**

Coincidence between Ch 1 and Ch 2 → sample if both internal trigger within a time slot of 200ns.

1. set Coincidence Gate Length to 200ns →  $0x19 = 25 * (2 * \text{CLK-Period}) = 25 * 8\text{ns} = 200\text{ns}$
2. enable “Internal Gate 1 Ch 1 Internal Trigger” → set bit 16
3. enable “Internal Gate 1 Ch 2 Internal Trigger” → set bit 17
4. set Gate Length to  $200\text{ns} + 2 * X$ ; with  $X = 5 \rightarrow 35 * 8\text{ns} = 280\text{ns}$
5. set Ch 1 Internal Trigger Delay to  $200\text{ns} + X \rightarrow 30 * 8\text{ns} = 240\text{ns}$
6. set Ch 2 Internal Trigger Delay to  $200\text{ns} + X \rightarrow 30 * 8\text{ns} = 240\text{ns}$

The following histogram shows the time difference in ns between the 48-bit Timestamps of Ch1 and Ch 2.



See software project :

`sis3316_ch1_ch2_coincidence` and `sis3316_ch1_ch2_coincidence_offline`

### 4.3 Internal 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 gate/sample logic immediately (asynchronous sample mode) or it can be routed to SIS3316 LEMO Output.

This Trigger Signal will be also used to generate the “Pileup and Re-Pileup” information, to build coincidence windows and to feed the Statistic Counter.

Features for each ADC channel:

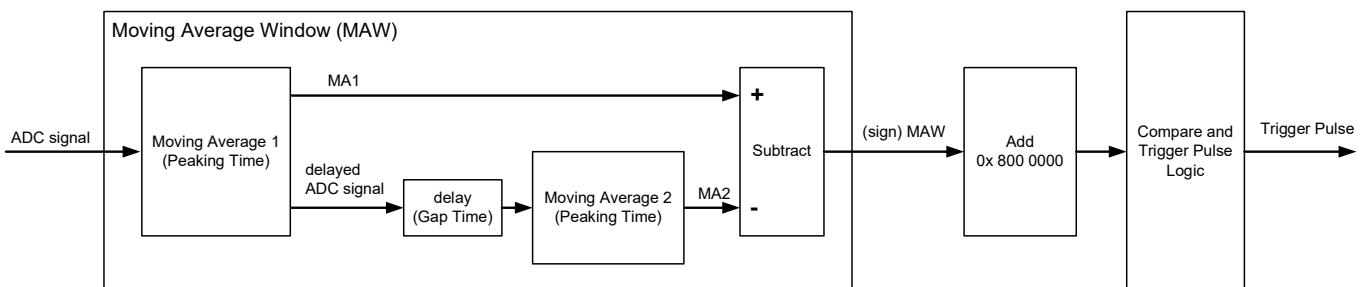
- Programmable Peaking Time (2, 4, 6, ....510)
- Programmable Gap Time (2, 4, 6, ....510)
- Programmable Trigger pulse out length (2, 4, 6, ....254)
- Programmable Trigger Threshold
- Programmable High Energy Trigger Threshold (Suppress trigger)
- Programmable Trigger Mode (CFD)
- Programmable Trigger OUT (Enable, Disable)

See “FIR Trigger Setup”, “Trigger Threshold” and “High Energy Trigger Threshold” registers.

#### 4.3.1 Block diagram of the Trigger MAW unit

Explanation:

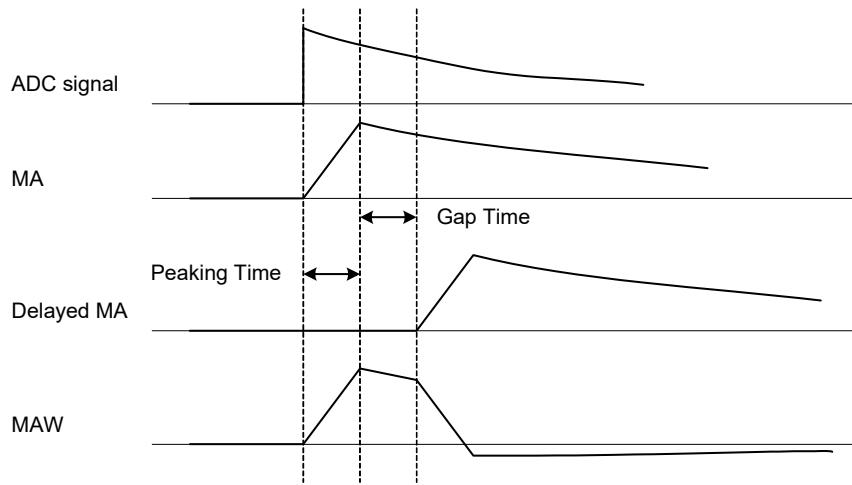
- **MAW or MW:** moving average window or moving window\*
- **MA:** moving average
- **Peaking Time:** the length of the MA for moving average unit
- **Gap Time:** the differentiation time of the moving window average unit



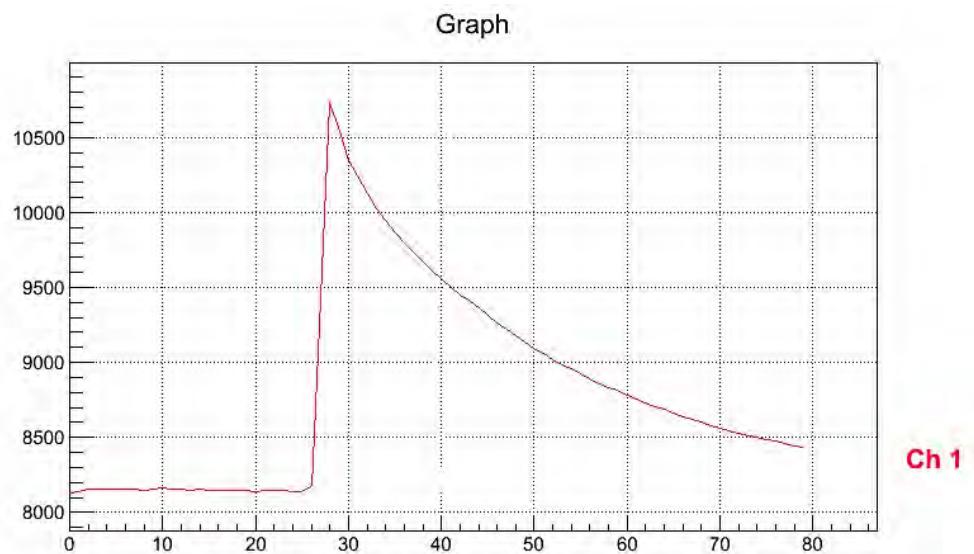
\* The FIR filters (Trigger and Energy) are building the complete sum of the adc data words over the Peaking Time. No “averaging” are performed!

The denotation of **MAW** is caused by history and it is to equate with **MW** in this document!

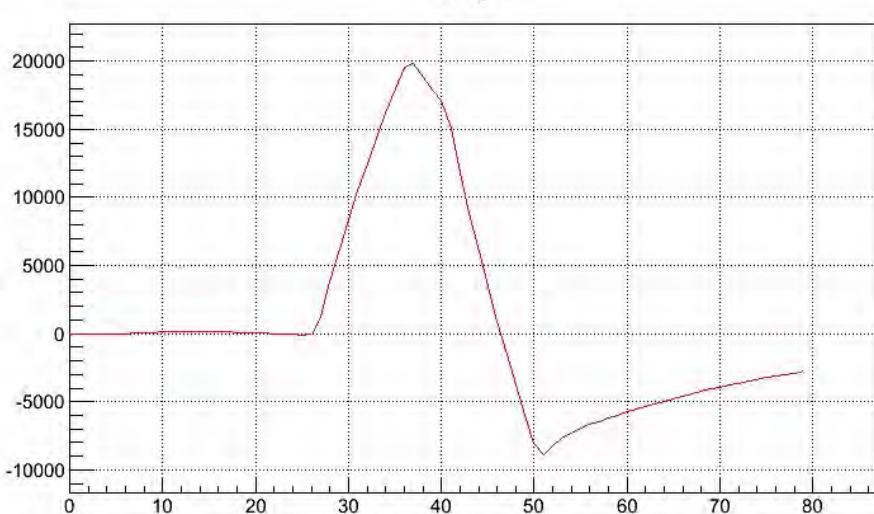
#### 4.3.2 Signal diagram of the Trigger MAW unit



ADC Signal:

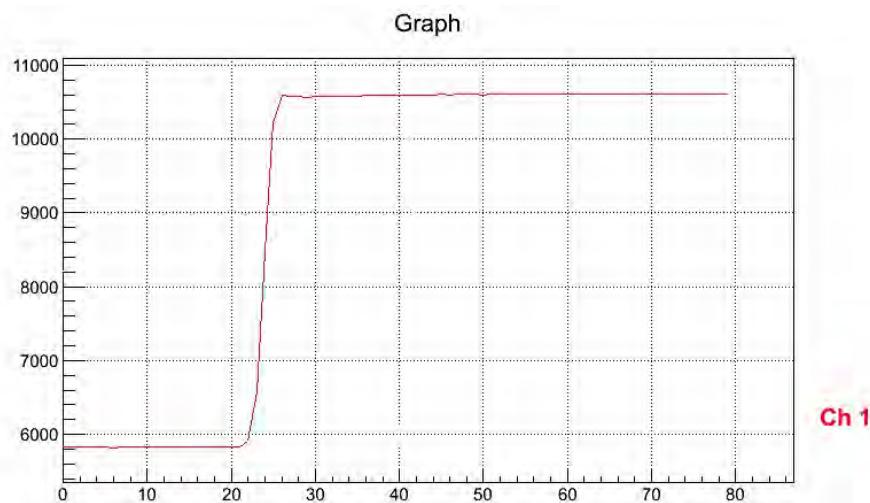


MAW Trapezoid:

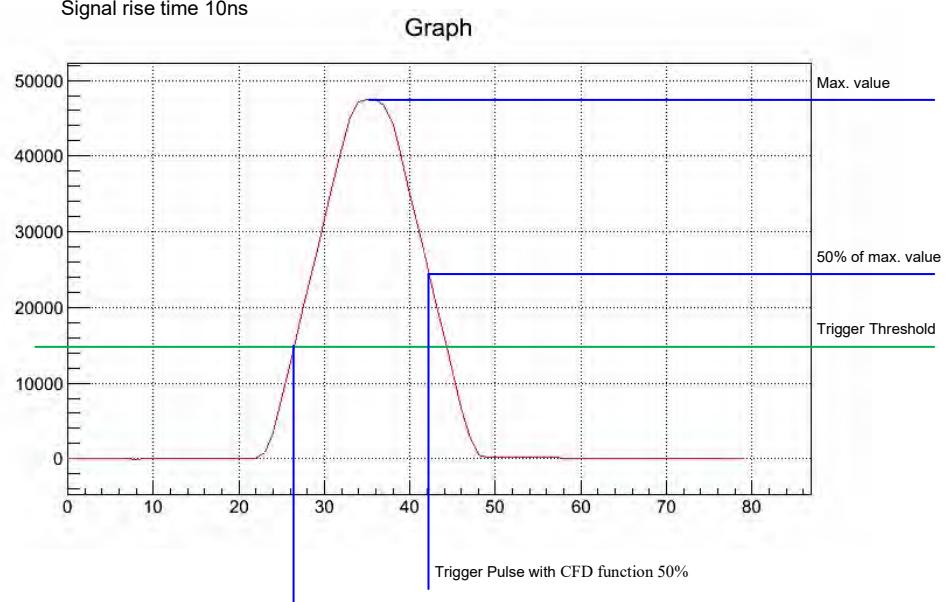
 $P = 10, G = 4$ 

### 4.3.3 CFD Trigger Feature

ADC Signal:



MAW:



CFD is disabled:

A trigger pulse will be issued if the actual trapezoidal value (MAW) goes above the programmable threshold value.

CFD is enabled:

The trigger logic will be armed if the actual trapezoidal value (MAW) goes above the programmable threshold value. The logic generates a trigger pulse if the trigger logic is armed and the actual trapezoidal value falls below the half of its maximum value (50%).

The 50% crossing of this digital CFD can be compared to the zero crossing of an analog CFD. The moment of the trigger doesn't depend on the pulse amplitude (provided that the rise times of the signals are identical).

The time resolution of the CFD trigger feature is limited by the sampling frequency.

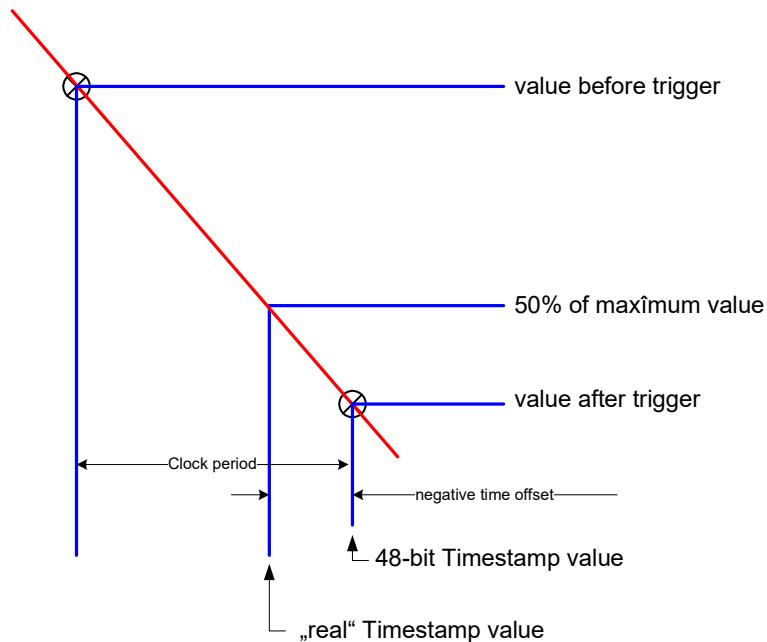
#### 4.3.4 Higher Timestamp Resolution

To get a Timestamp, which is better than the sampling period, the readout option to get three additional values of the MAW trapezoid is implemented:

- maximum value
- value after trigger
- value before trigger

With these three MAW values, it is possible to make a linear interpolation between the two values (before and after) to calculate a “negative time offset” to the latched 48-bit timestamp value to get the “real” timestamp.

**Note:** CFD feature must be enabled !



```

read_maw_max      = (gl_rblt_data[maw3_index]    & 0xffffffff) - 0x8000000 ;
read_maw_befor_trigger = (gl_rblt_data[maw3_index+1] & 0xffffffff) - 0x8000000 ;
read_maw_with_trigger = (gl_rblt_data[maw3_index+2] & 0xffffffff) - 0x8000000 ;

float_diff_N_to_MaxDiv2 = (read_maw_max/2) - (read_maw_with_trigger) ;
float_diff_Nm1_to_N     = (read_maw_befor_trigger) - (read_maw_with_trigger) ;

float_timestamp_correctur = float_diff_N_to_MaxDiv2 / float_diff_Nm1_to_N;

ch_double_timestamp_ns   = ( (4294967296.0 * (ch_Timestamp_upper >> 16))
                           +(ch_Timestamp_lower) ) * float_sample_periode_ns ;

// negative time offset
ch_correction_ns_value = float_timestamp_correctur * float_sample_periode_ns ;

// "real" Timestamp value
ch_double_timestamp_ns   = ch_double_timestamp_ns - ch_correction_ns_value ;

```

#### 4.3.4.1 Higher Timestamp Resolution Example 1

See software project: sis3316\_hi\_resolution\_timestamp\_test\_ch1

This program shows how to calculate the “negative time offset” factor (in sample clock period).

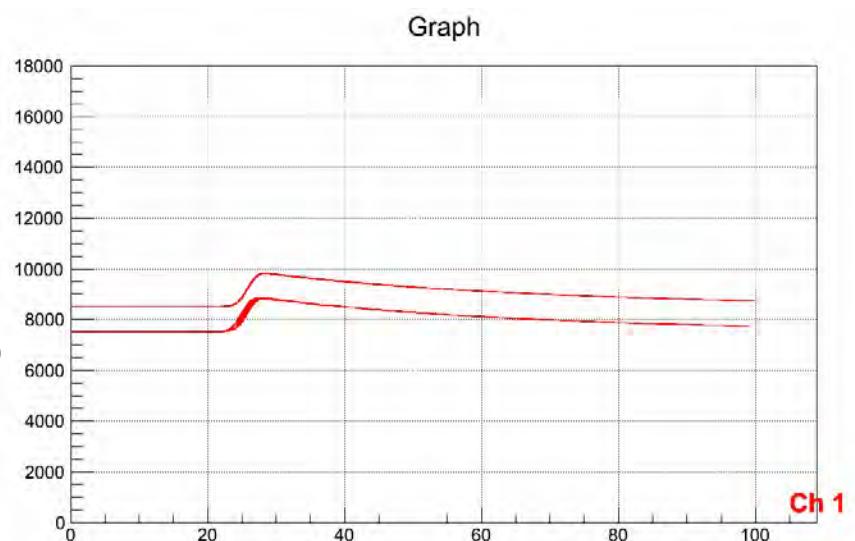
The following plot shows an exponential signal on channel 1 (amplitude 500mV, offset 0 mV),

which was sampled 30750 times.

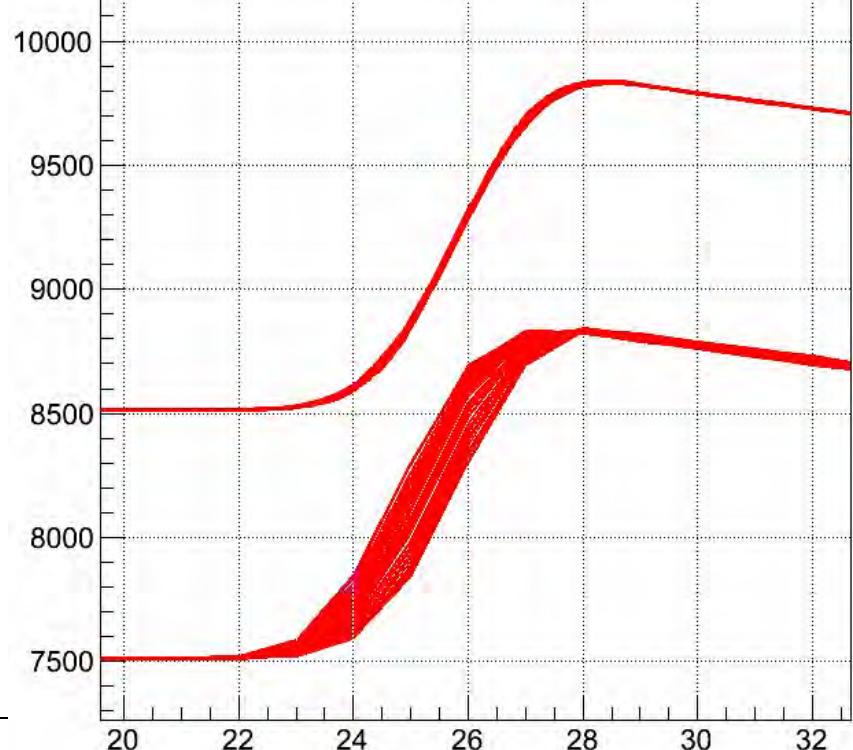
The **lower trace bunch** shows 30750 signals without the “Higher Timestamp Resolution” correction.

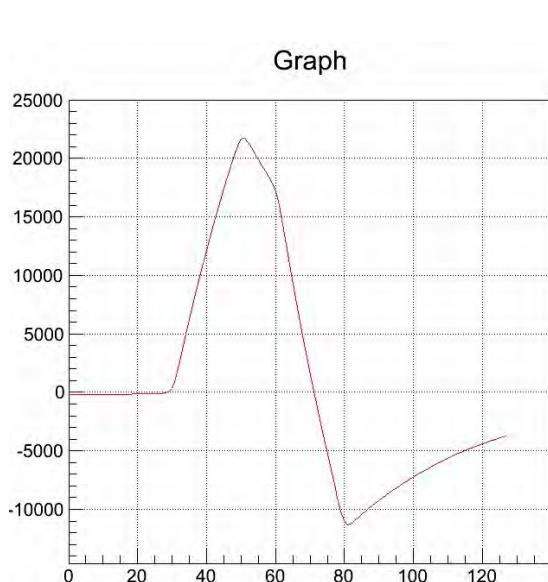
The plot shows a jitter of 1 sample clock period.

The **upper trace bunch** (add offset of 1000 to the y-axis) shows 30750 signals with the “Higher Timestamp Resolution” correction on the x-axis.  
(x-axis: sample clock period)

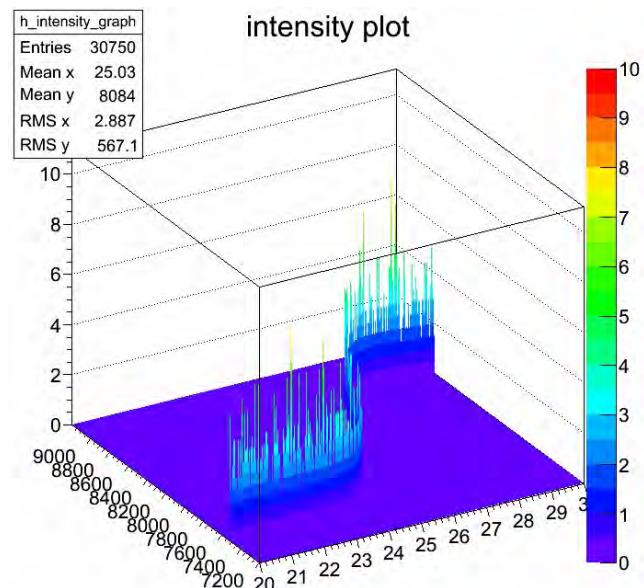


Zoomed:

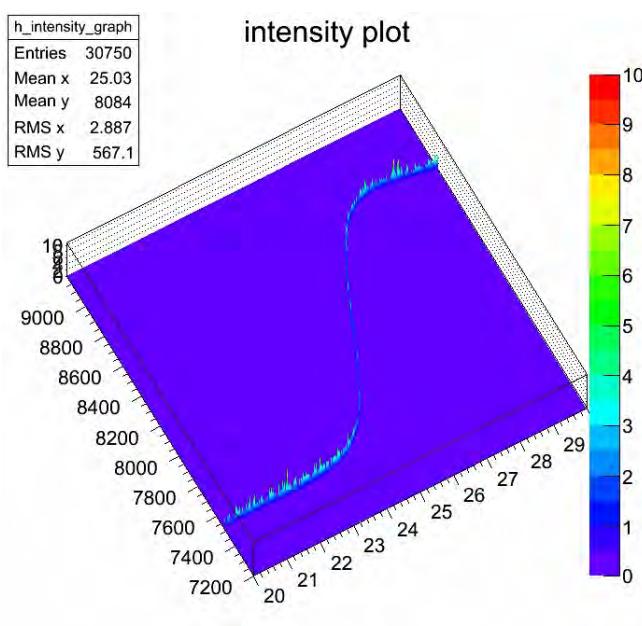




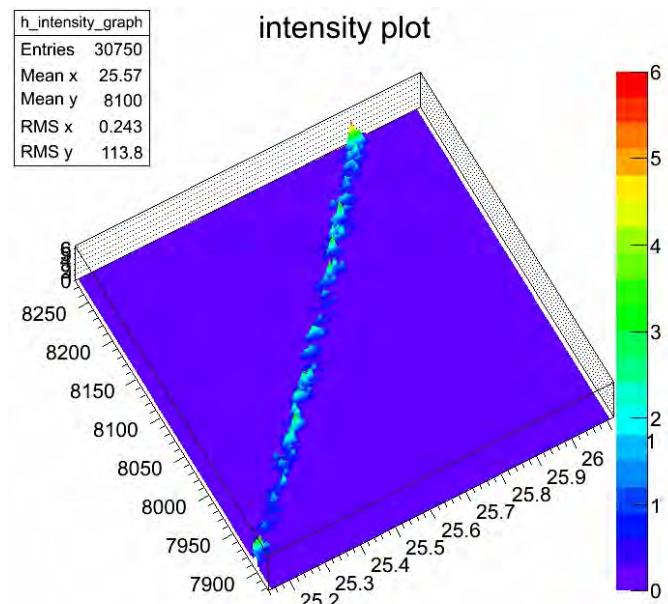
MAW Trapezoid



Intensity plot



Intensity plot



Zoomed intensity plot

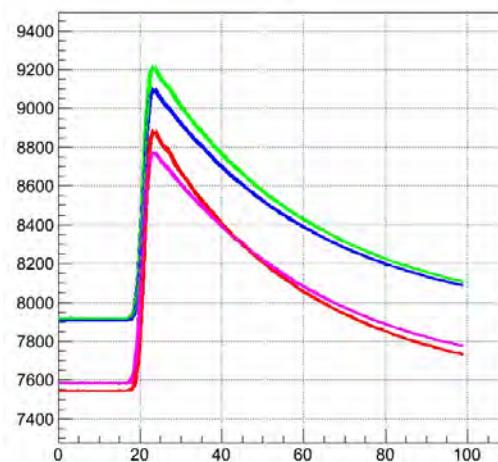
#### 4.3.4.2 Higher Timestamp Resolution Example 2

See software project: sis3316\_Delay\_measurement\_from\_ch1\_to\_2\_3\_4

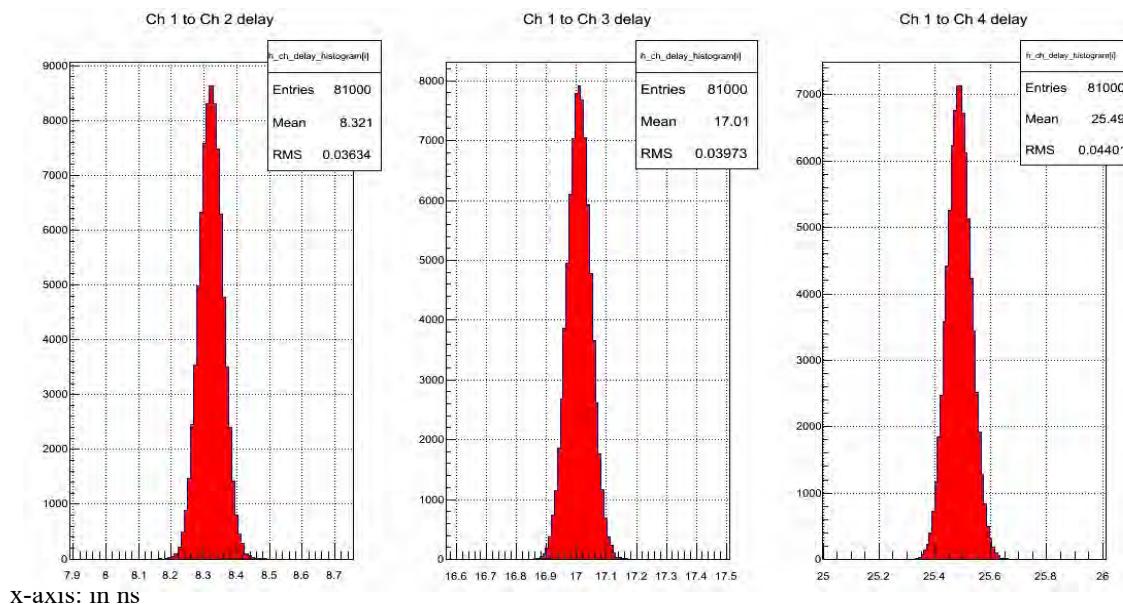
Test set-up: The channels 1, 2, 3, 4 are fed with the same signal but delayed (daisy chained) with three 1.5 meter LEMO cables.

Software: The program enables the 50 ohm termination for the last channel, only. It takes some 10000 events/hits of the four channels and calculates the “LEMO cable” delays between the first channel (ch1) and the other three channels (ch2, 3, 4) with the help of the “real” Timestamp equation:  
`ch_double_timestamp_ns = ch_timestamp_ns - ch_correction_ns_value ;`

Exponential input signal:



Time measurement histograms:

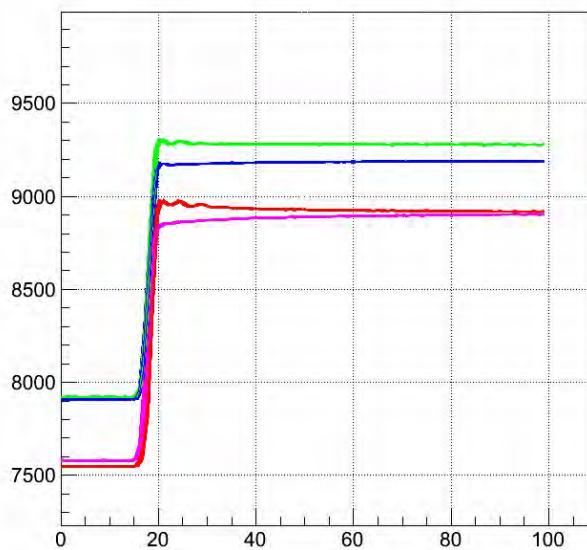


X-axis: in ns

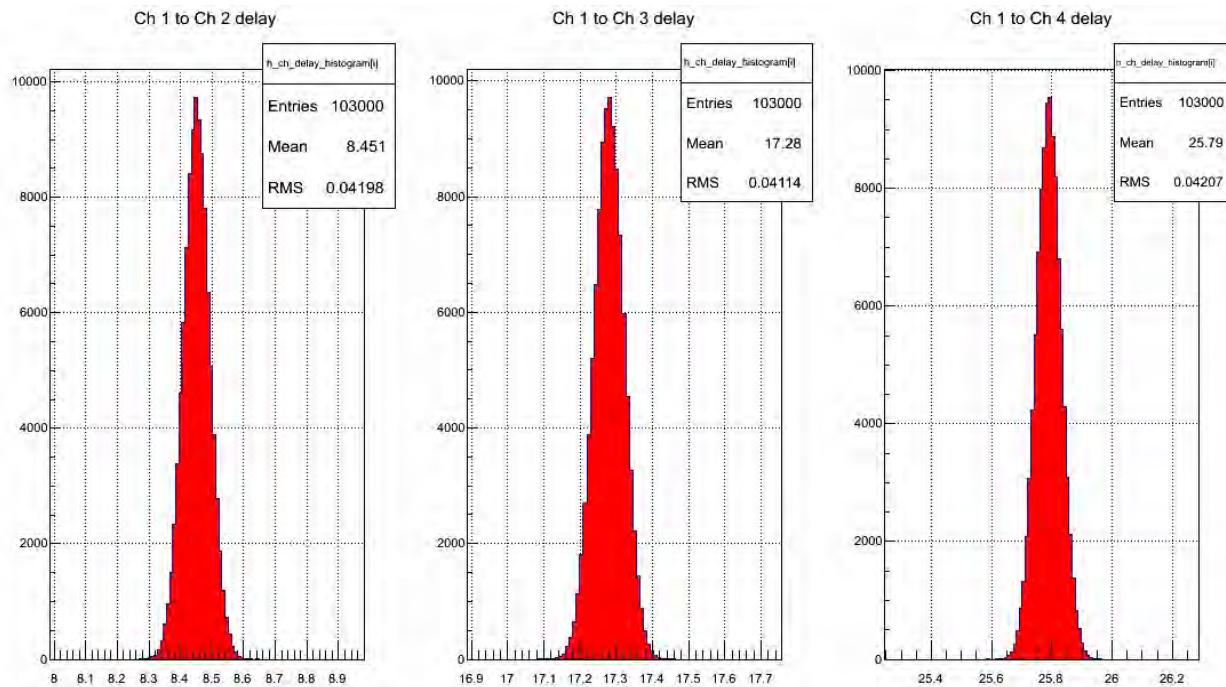
$$\text{Timing resolution} = \text{RMS} * 2.35 / 1.44 \rightarrow \sim 65\text{ps FWHM}$$

**Note:** This value can be seen as optimal intrinsic resolution of the digitizer for noise free signals with identical pulse shape/rise time. Results for real detector signals with noise and rise time fluctuations will vary.

input signal:



Time measurement histogram:



x-axis: in ns

Timing resolution =  $\text{RMS} * 2.35 / 1.44 \rightarrow \sim 68\text{ps FWHM}$  (see note on previous page)

See further software projects, also:

#### **sis3316\_Delay\_measurement\_from\_ch1\_to5\_9\_13:**

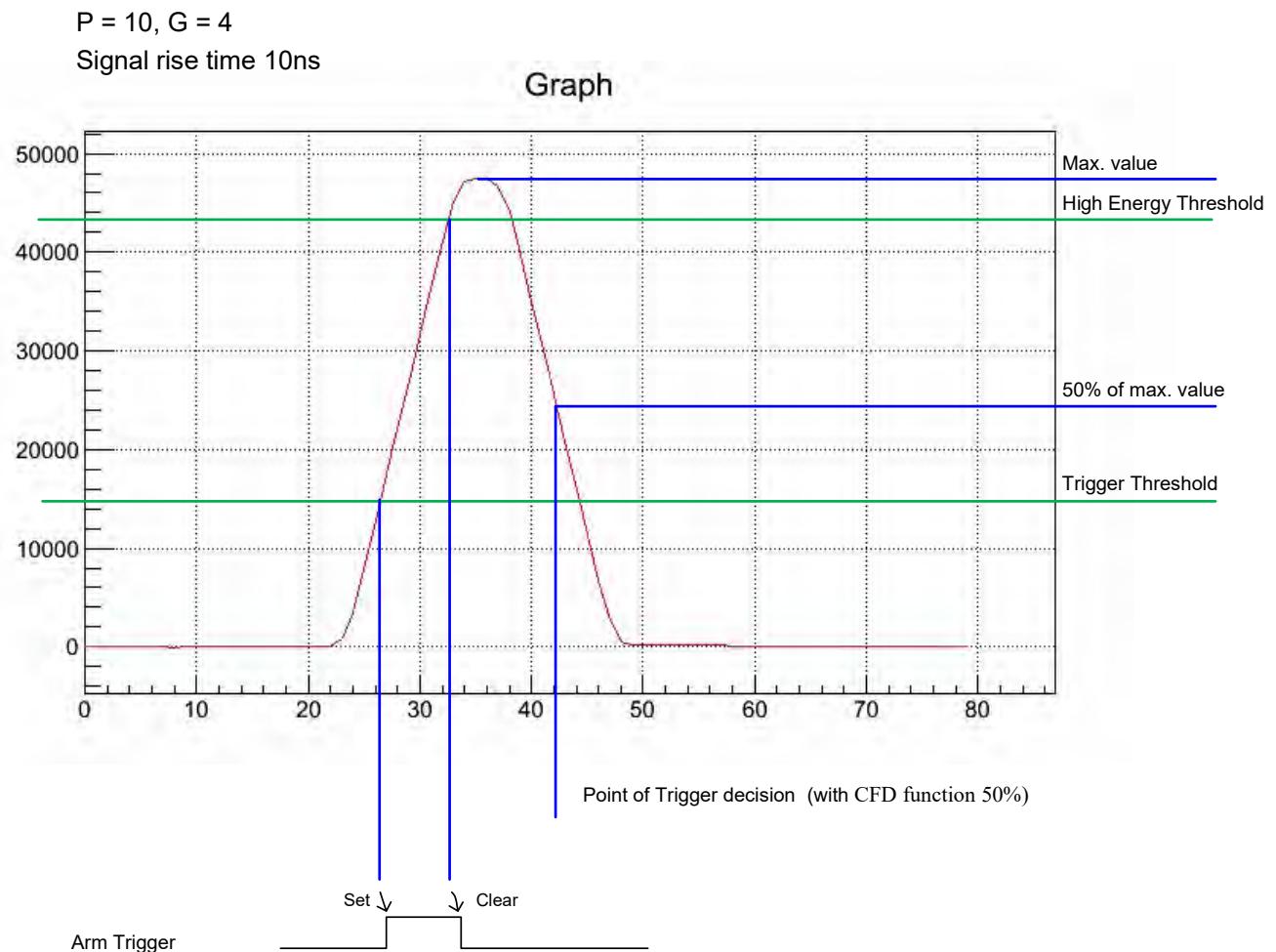
- measurement between channels of different ADC FPGA groups

#### **sis3316\_Delay\_mess\_from\_m1ch1\_to\_m2ch5\_9\_13:**

- measurement between channels of two different SIS3316 modules

#### 4.3.5 High Energy Suppress Trigger Mode

A High Energy Suppress Trigger Mode is implemented to suppress triggers. A trigger will be suppressed if the running sum of the trapezoidal filter goes above the value of the High Energy Threshold register.



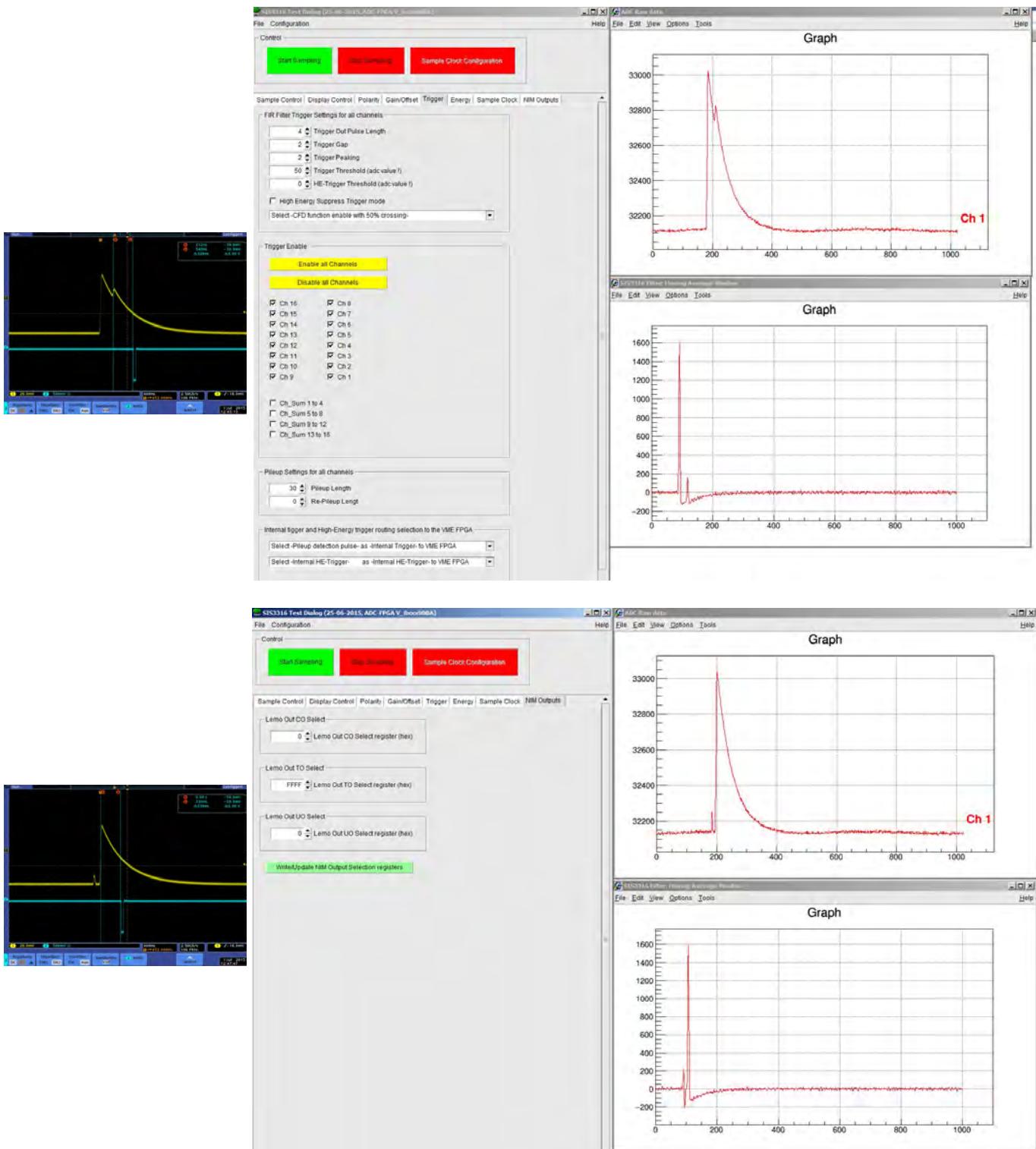
The Trigger Logic will be armed if the trapezoidal filter value (red) goes above the Trigger Threshold value and it will be disarmed if the trapezoidal filter value goes above the High Energy Threshold value. The trigger will be generated only if the “Arm Trigger” signal is valid at the point of trigger decision.

**Note:** This mode requires the CFD function to be enabled.

#### 4.3.6 Pileup Trigger Generation

In combination with the “Ch x internal Pileup Trigger enable” bits and the “Pile Window”, it is possible save Pileup-Events, only.

It is also possible to route the Pileup detection pulses to the LEMO outputs.



## 4.4 Sample Logic

The sample logic starts with a “Ch x trigger pulse” and executes the following steps provided the Sample Control logic is armed:

1. starts the “Active Trigger Gate Window” and latches the 48-bit timestamp.
2. writes the 16-bit programmable Channel Header ID and the latched 48-bit Timestamp to the Event Memory.
3. writes a programmable number of Ch x Raw Values to the Event Memory
4. writes at the end of the “Active Trigger Gate Window” the Peakhigh value and its Index, the Pileup information, the 8 accumulator values, three MAW values to the Event Memory respectively if it is enabled.
5. writes the buffered MAW data to the Event Memory if enabled.

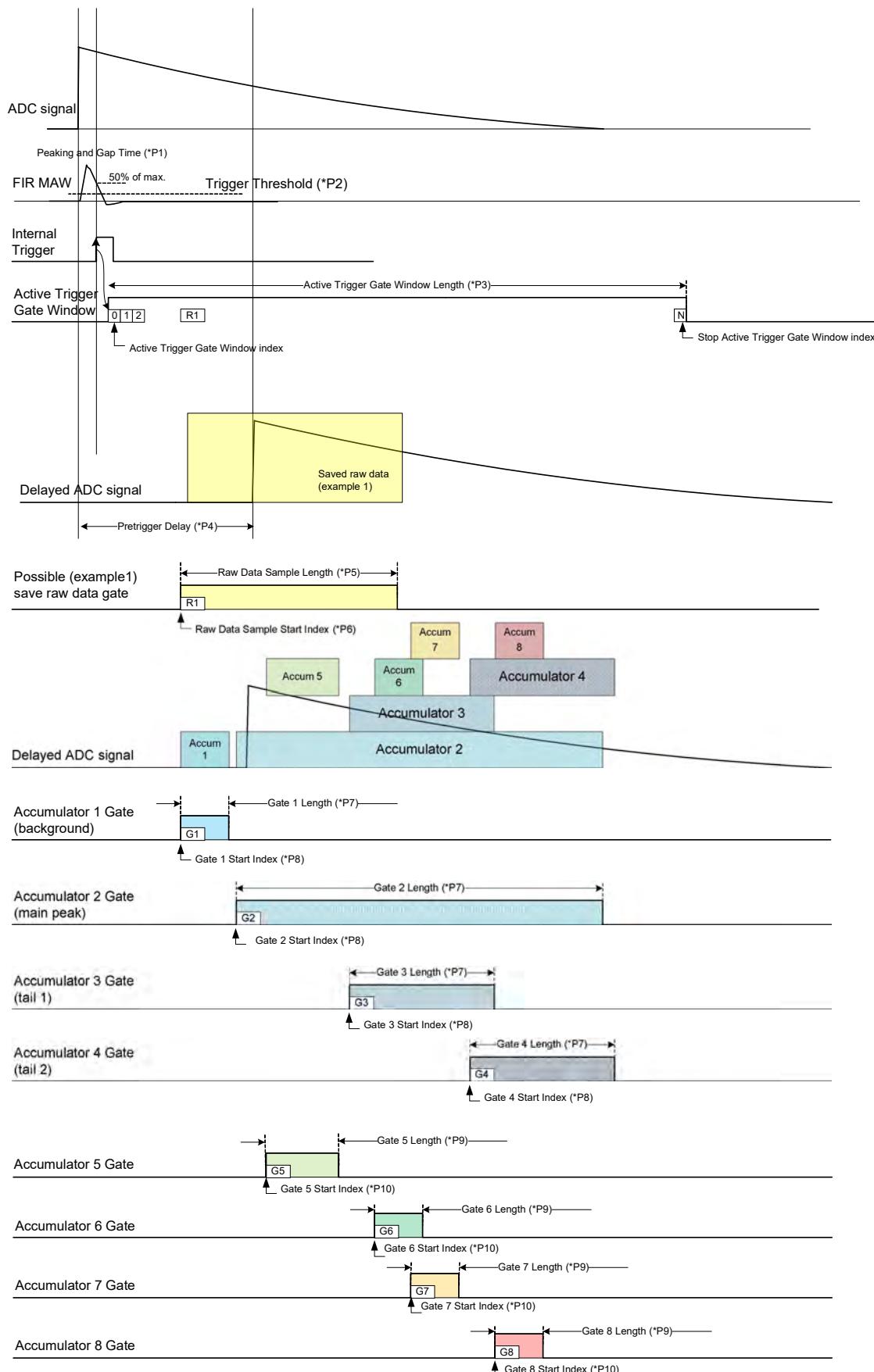
Hints: the “Active Trigger Gate Window” should cover the calculations of all used accumulators, the start of the sampling of the raw data, the length of the saved MAW test buffer and the maximum value of the Energy MAW (trapezoidal).

### 4.4.1 Neutron/Gamma application (accumulators, integration)

The Neutron/Gamma Logic parameters, related to the following illustration, are:

- \*P1: FIR Filter Trigger parameters: see FIR Trigger Setup registers
- \*P2: FIR Filter Trigger Threshold: see FIR Trigger Threshold registers
- \*P3: Active Trigger Gate Window Length: see Active Trigger Gate Window Length registers
- \*P4: Pre Trigger Delay: see Pre Trigger Delay registers
- \*P5: Raw Data Sample Length: see Raw Data Buffer Configuration registers and Extended Raw Data Buffer Configuration registers
- \*P6: Raw Data Sample Start Index: see Raw Data Buffer Configuration registers
- \*P7: Gate Length: see Accumulator Gate X Configuration registers
- \*P8: Gate Start Index: see Accumulator Gate X Configuration registers
- \*P9: Gate Length: see Accumulator Gate X Configuration registers
- \*P10: Gate Start Index: see Accumulator Gate X Configuration registers

## Neutron/Gamma Logic parameter illustration:



#### 4.4.2 Gamma application (Trapezoidal FIR Energy Filter)

A second trapezoidal FIR Energy filter, besides the trigger generation FIR filter, is implemented for each channel to generate a “moving window” stream.

Features of the Energy FIR Filter:

- Programmable Peaking Time (max. 2044 Clocks)
- Programmable Gap Time (max. 510 Clocks)
- Decay correction

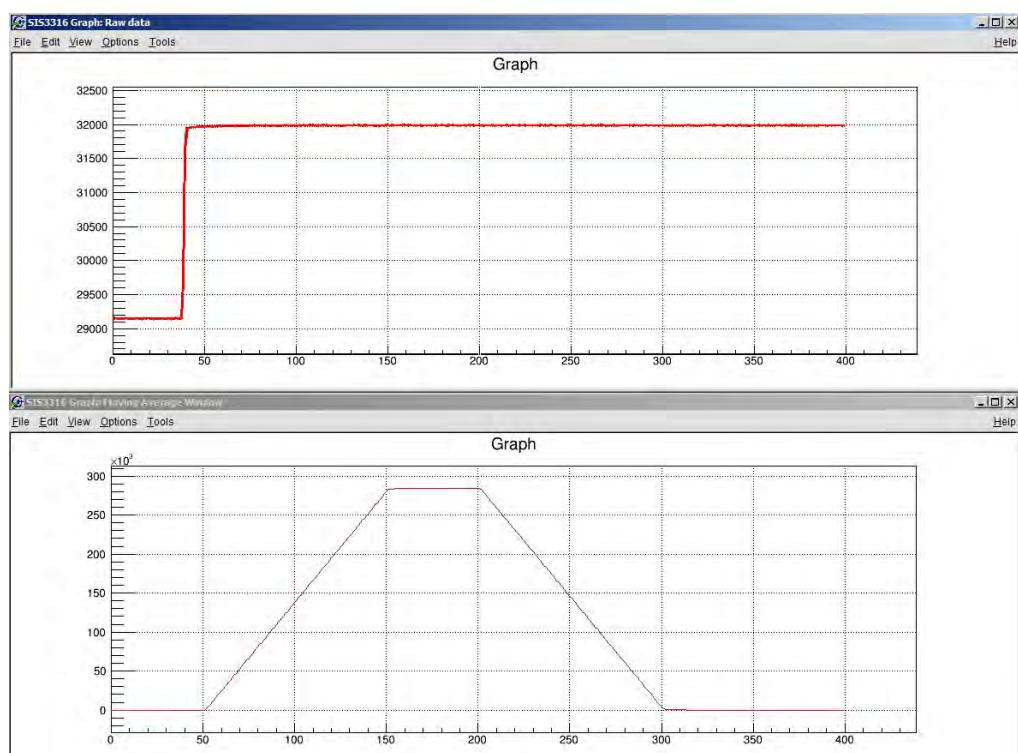
Storage Features for the Energy Filter value:

- Store up 1022 (2048\*) values of the “moving window”
- Save the start energy value at the beginning of the “Active Trigger Gate Window”
- Save the maximum value of the “moving window” (trapezoidal) or pickup an Energy value from a programmed index\*.
- Histogramming of the “saved” (maximum or pickup value) Energy value or inside the FPGA (64K bins x 32bits)

\* with ADC Version V-0250-000A and V-0125-000A and higher

The Energy Filter logic builds two sums over P (Peaking Time) ADC raw data values with a time difference of G (Gap Time). The result for an ideal step function signal is a trapezoidal as illustrated below.

ADC raw data  
100 mV step function

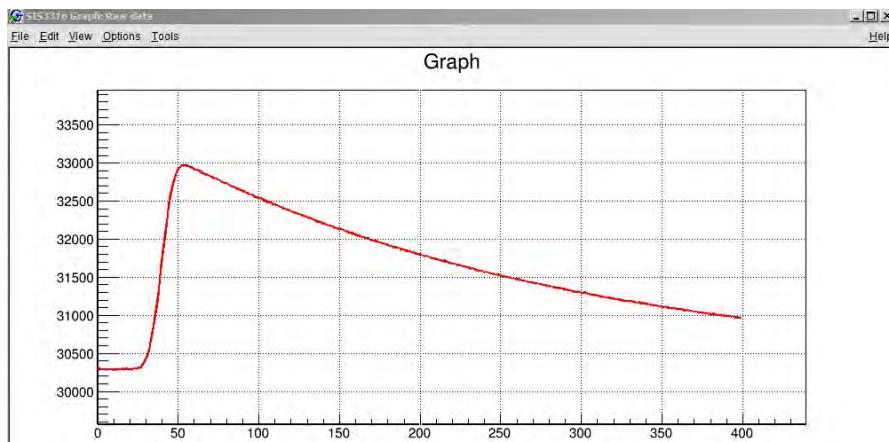


Moving window with  
- P = 100  
- G = 50

The amplitude of the input signal (step function) is 100mV. This corresponds to appr. 2850 ADC counts (SIS3316-16bit at 2V input range). The resulting maximum value of the “moving window” is appr. 285.000 (P \* amplitude counts).

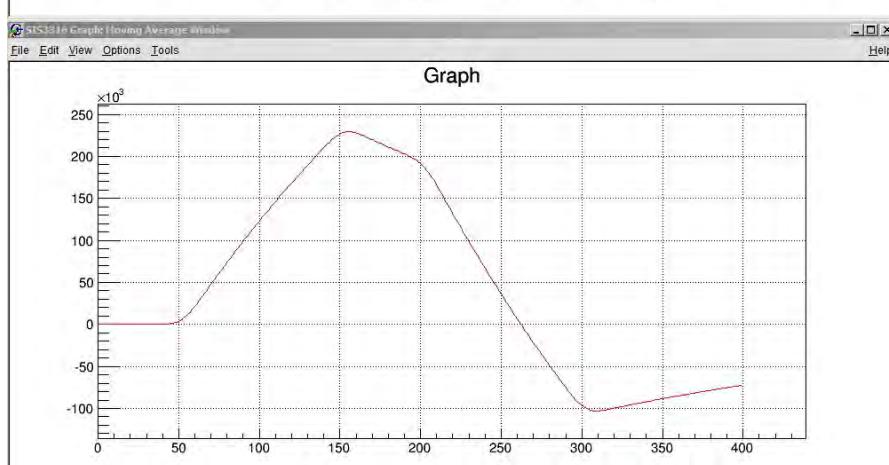
The following example shows the decay correction of a detector signal:

Detector signal



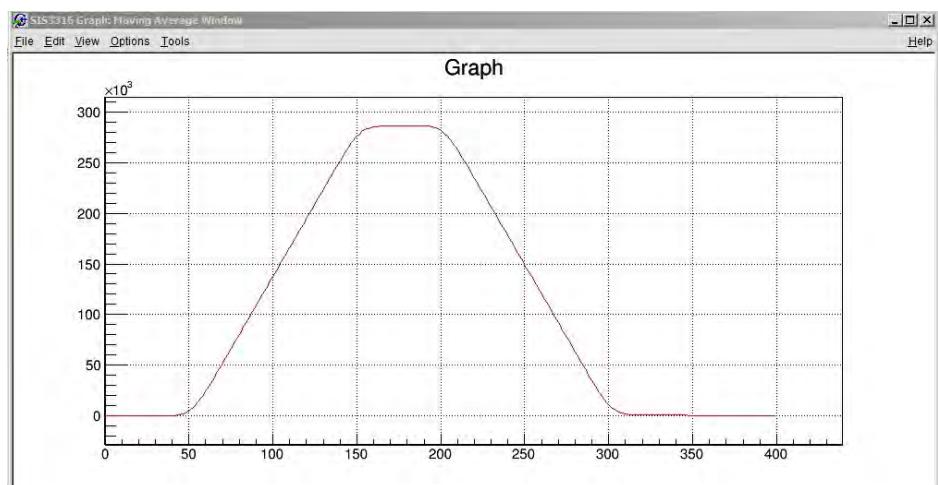
Moving window with

- P = 100
- G = 50
- no decay correction



Moving window with

- P = 100
- G = 50
- decay correction

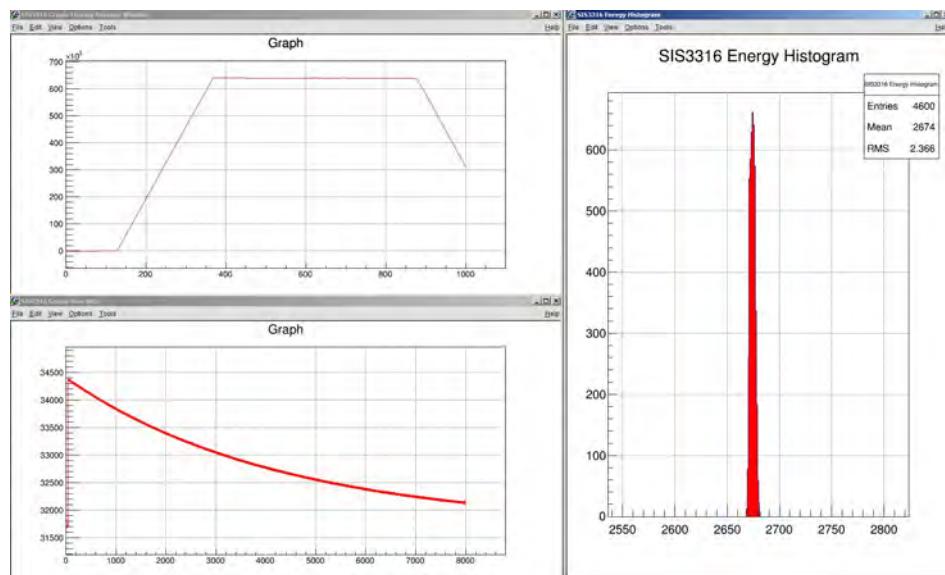


The decay correction value (tau table/factor) should be chosen in a fashion, that the moving window (trapezoidal) returns to the baseline.

The following example shows the histogram of the maximum value of the Energy filter:

Moving window with

- P = 240
- G = 510
- decay correction

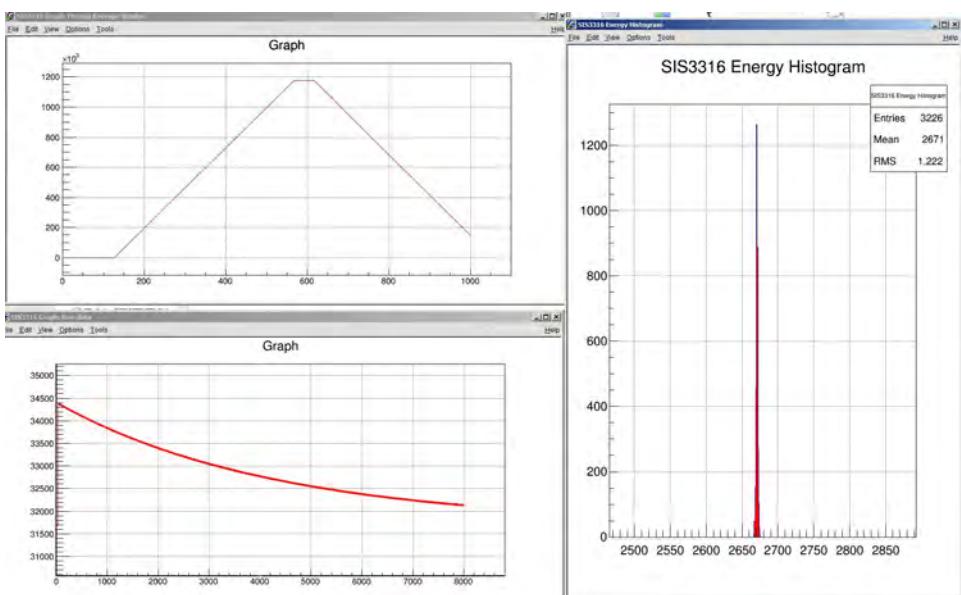


Detector signal

The following example shows the histogram of the maximum value of the Energy filter with a higher P and a smaller G value:

Moving window with

- P = 440
- G = 50
- decay correction



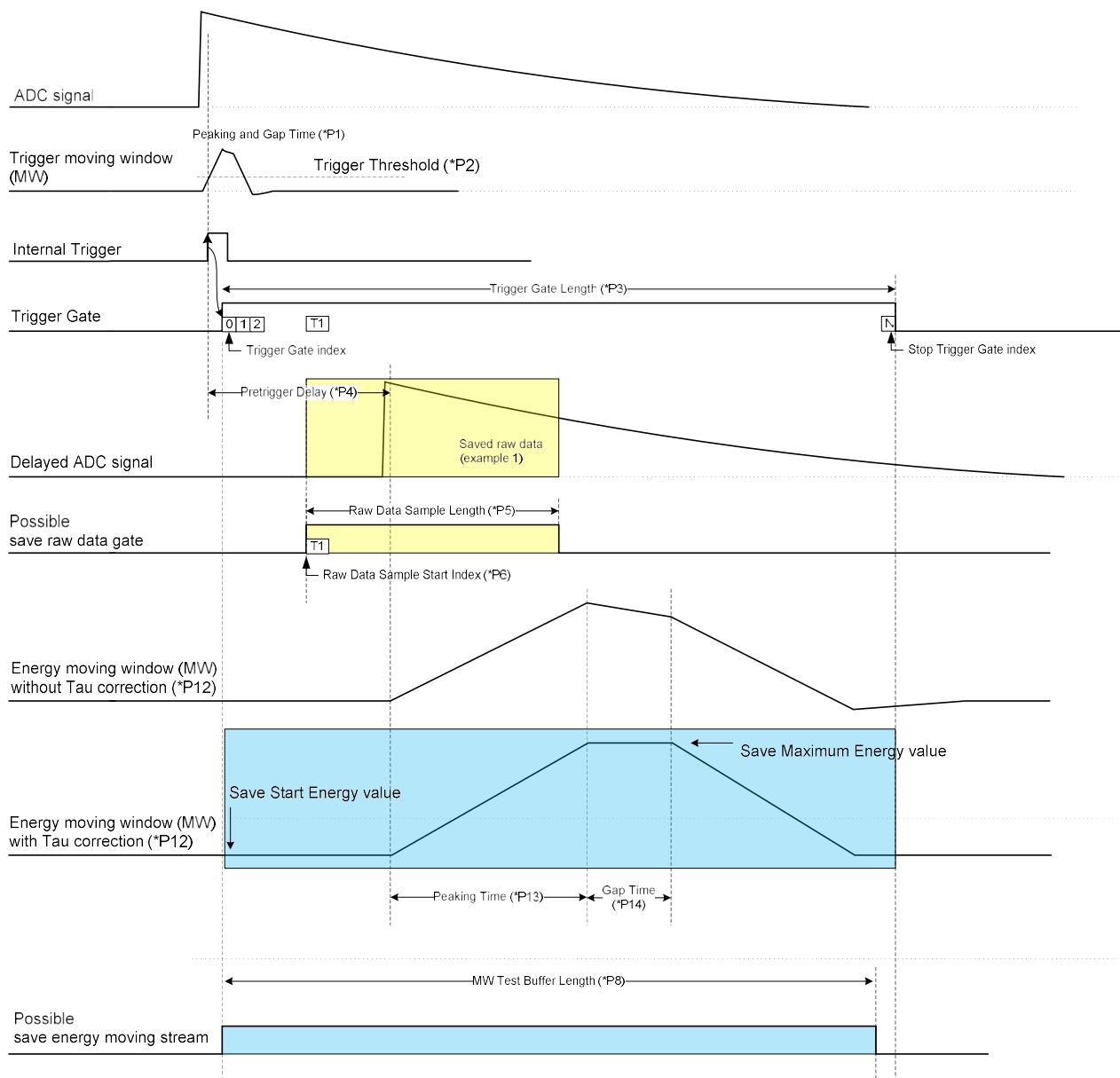
Detector signal

Energy resolution:

please, refer to the document SIS3316-M-1-1-Vxxx-InputRange-Resolution-addendum.pdf for more information on the resolution of the SIS3316.

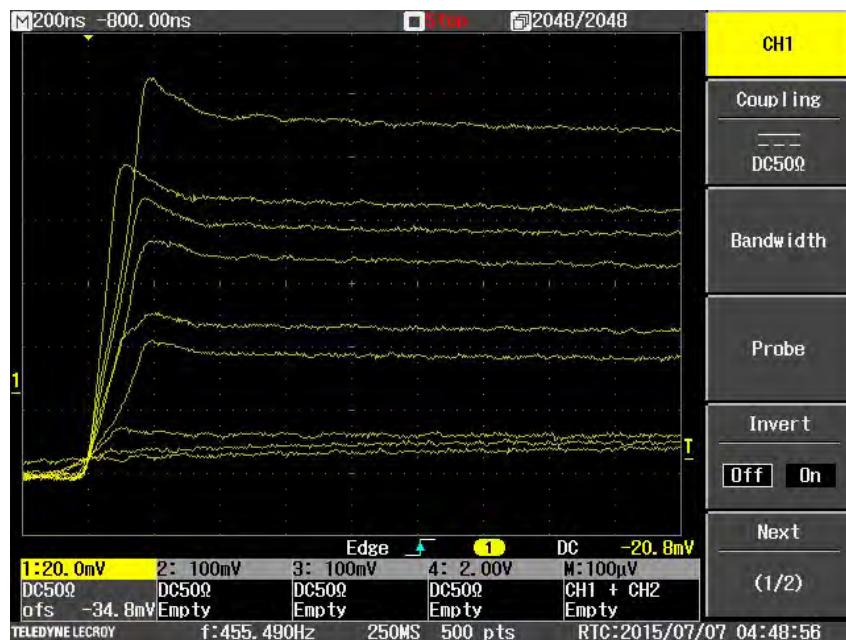
The Gamma Logic parameters, related to the following illustration, are:

- \*P1: FIR Filter Trigger parameters: see FIR Trigger Setup registers
- \*P2: FIR Filter Trigger Threshold: see FIR Trigger Threshold registers
- \*P3: Active Trigger Gate Window Length: see Active Trigger Gate Window Length registers
- \*P4: Pre Trigger Delay: see Pre Trigger Delay registers
- \*P5: Raw Data Sample Length: see Raw Data Buffer Configuration registers and Extended Raw Data Buffer Configuration registers
- \*P6: Raw Data Sample Start Index: see Raw Data Buffer Configuration registers
- \*P8: MW Test Buffer Length: MW Test Buffer Configuration register
- \*P12: Tau factor: see FIR Energy Setup registers
- \*P13: Peaking time: see FIR Energy Setup registers
- \*P14: Gap time: see FIR Energy Setup registers

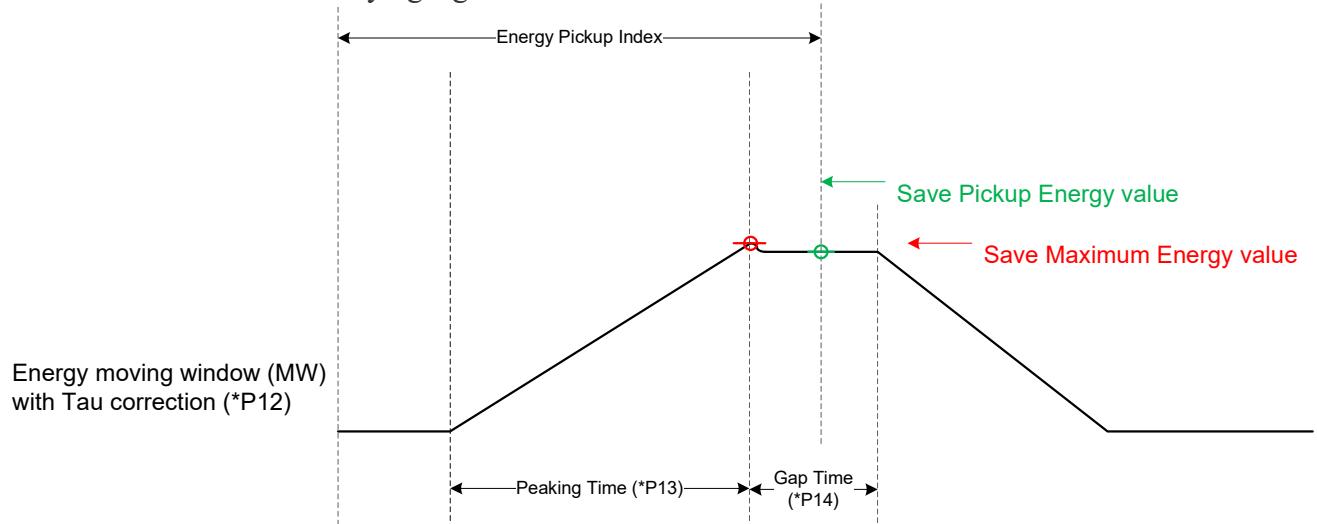


## SIS3316 new energy pickup index functionality:

Ge detector preamplifier output signals tend to produce an overshoot at the beginning of the signal as shown in the scope screenshot below. This overshoot may vary in amplitude even for a fixed energy. A variation of the extracted energy is the result whenever the maximum of the flat top is used, as the overshoot will also reflect in an overshoot in the energy moving window (MW).



The drawing below illustrates the difference between the use of the maximum energy value and the pickup energy index. The use of the pickup energy value will result in better energy resolution for cases with varying signal overshoot.



Note: The use of CFD triggering is recommended in combination with the energy pickup index functionality to minimize trigger jitter with induced jitter of the pickup point in time on the flat top of the energy MW.

## 4.5 Pileup Detection Logic

A pileup detection logic is implemented for each channel. This logic allows for the definition of two windows (time slots):

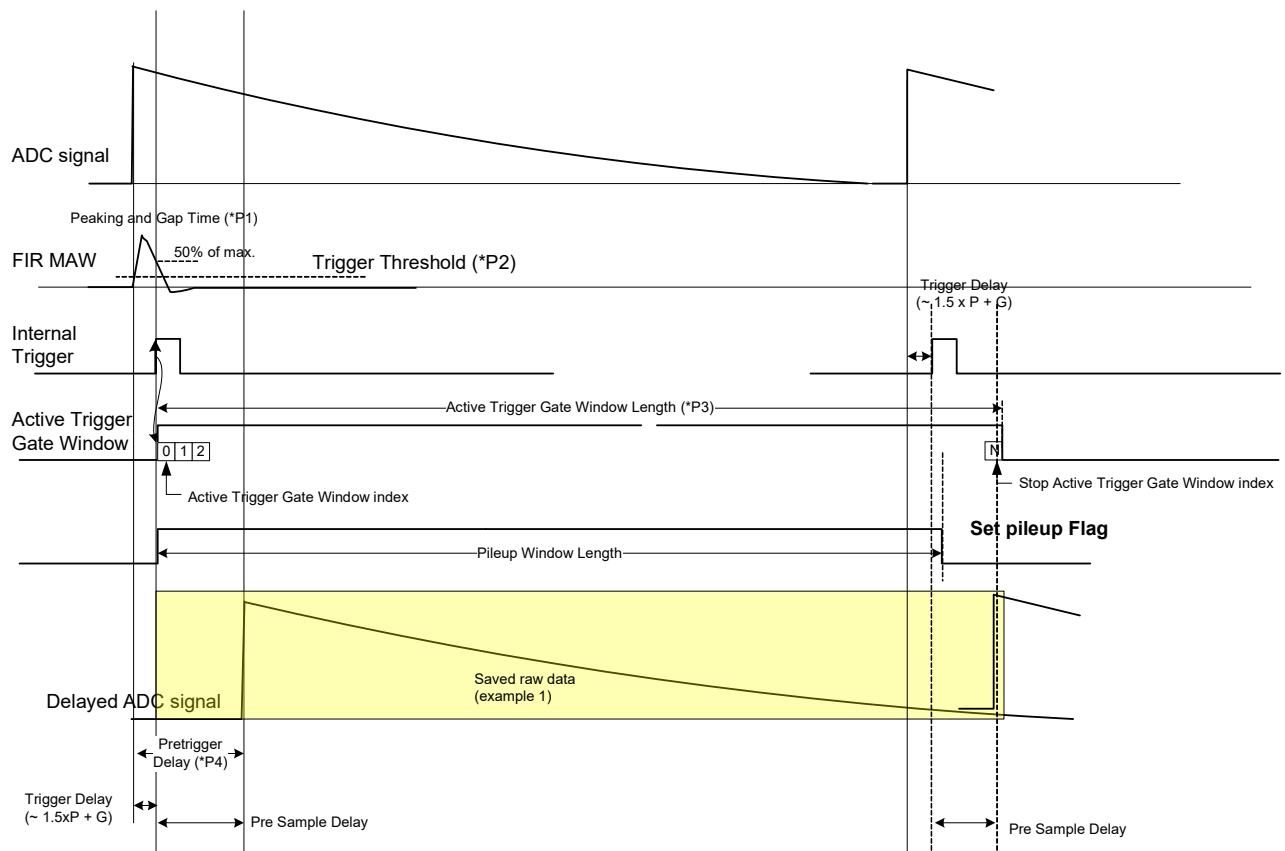
1. after the hit (trigger starts the “Active Trigger Gate Window” ) → detect Pileups
2. before the hit (trigger starts the “Active Trigger Gate Window” ) → detect Re-Pileups (or Pre-Pileup)

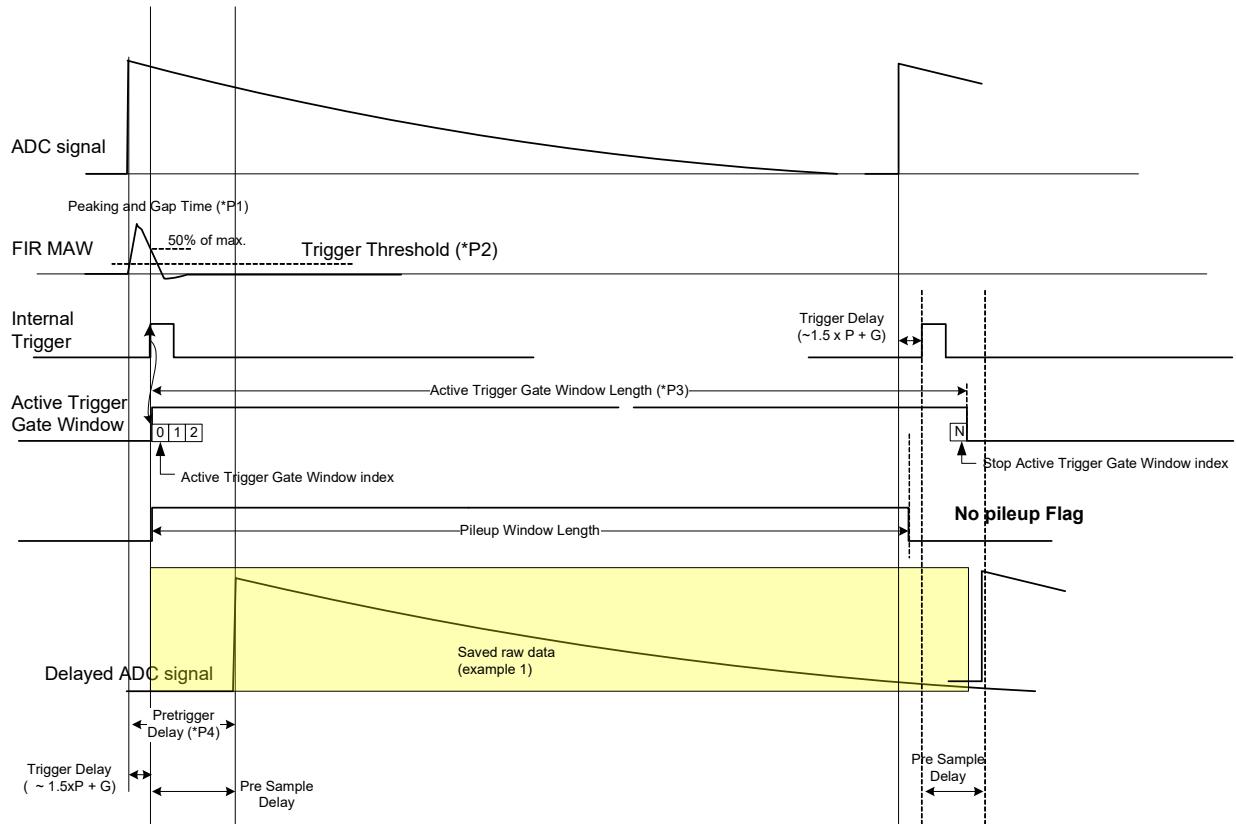
### 4.5.1 Pileup Window

The “Pileup Window” starts with the beginning of the “Active Trigger Gate Window”. The Length of the “Pileup Window” is programmed with the Pileup Configuration registers.

A Pileup Flag will be set if the “Pileup Window” is active and a further internal trigger pulse occurs (see below). It is also possible to use this “pileup window” logic to generate a trigger condition in case of pileups, only.

**Set Pileup Flag Illustration:**

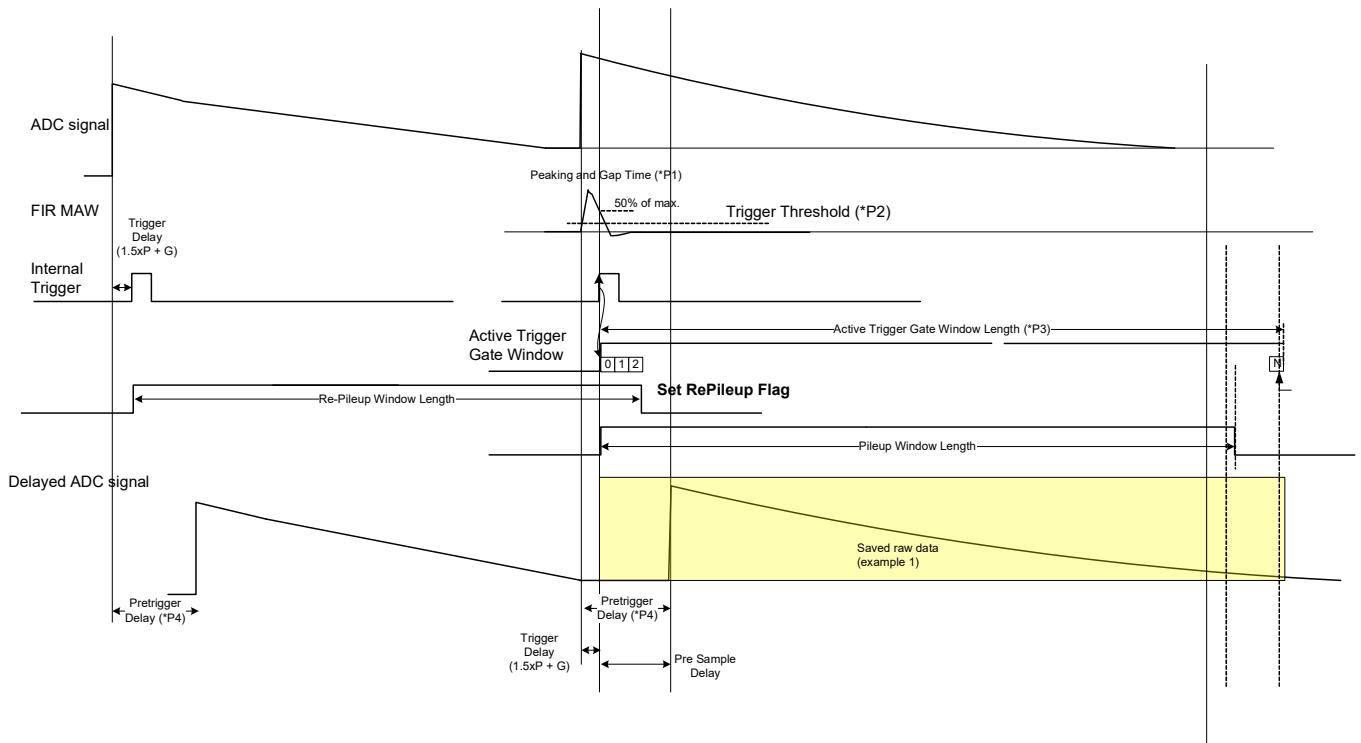


**Not set Pileup Flag Illustration:**

#### 4.5.2 Re-Pileup (Pre-Pileup) Window

The “Re-Pileup Window” starts (restarts) with each internal trigger pulse. The length of the “Re-Pileup Window” is programmed with the Pileup Configuration registers.

A Re-Pileup Flag will be set if the “Re-Pileup Window” is active and a further internal trigger pulse triggers the start of the “Active Trigger Gate Window” (see below).



## 4.6 Hit/Event Data Format

The Hits/Events are stored to Bankx memory with the following data format:

| 31                                                                  | 16                                  | 15                               | 4 | 3           | 0 |
|---------------------------------------------------------------------|-------------------------------------|----------------------------------|---|-------------|---|
| Timestamp [47:32]                                                   |                                     | Channel ID                       |   | Format bits |   |
| Timestamp [31:16]                                                   |                                     | Timestamp [15:0]                 |   |             |   |
| Index of Peakhigh value [15:0]                                      |                                     | Peakhigh value [15:0]            |   |             |   |
| Information [7:0]                                                   |                                     | Accumulator sum of Gate 1 [23:0] |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 2 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 3 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 4 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 5 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 6 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 7 [27:0]    |                                  |   |             |   |
| "0000"                                                              | Accumulator sum of Gate 8 [27:0]    |                                  |   |             |   |
| "0000"                                                              | MAW maximum value [27:0]            |                                  |   |             |   |
| "0000"                                                              | MAW value before Trigger [27:0]     |                                  |   |             |   |
| "0000"                                                              | MAW value after/with Trigger [27:0] |                                  |   |             |   |
| Start Energy value (Energy value from first value of Trigger Gate ) |                                     |                                  |   |             |   |
| Max. Energy value (during Trigger Gate active )                     |                                     |                                  |   |             |   |

If Format bit 0 = 1

If Format bit 1 = 1

If Format bit 2 = 1

If Format bit 3 = 1

| 31-28 | 27            | 26          | 25-0                                              |
|-------|---------------|-------------|---------------------------------------------------|
| 0xE   | MAW Test Flag | Status Flag | number of raw samples (x 2 samples, 32-bit words) |

|                                              |          |            |
|----------------------------------------------|----------|------------|
| ADC raw data if number of raw samples != 0x0 | sample 2 | sample 1   |
|                                              | sample 4 | sample 3   |
| sample N                                     |          | sample N-1 |

MAW Test data

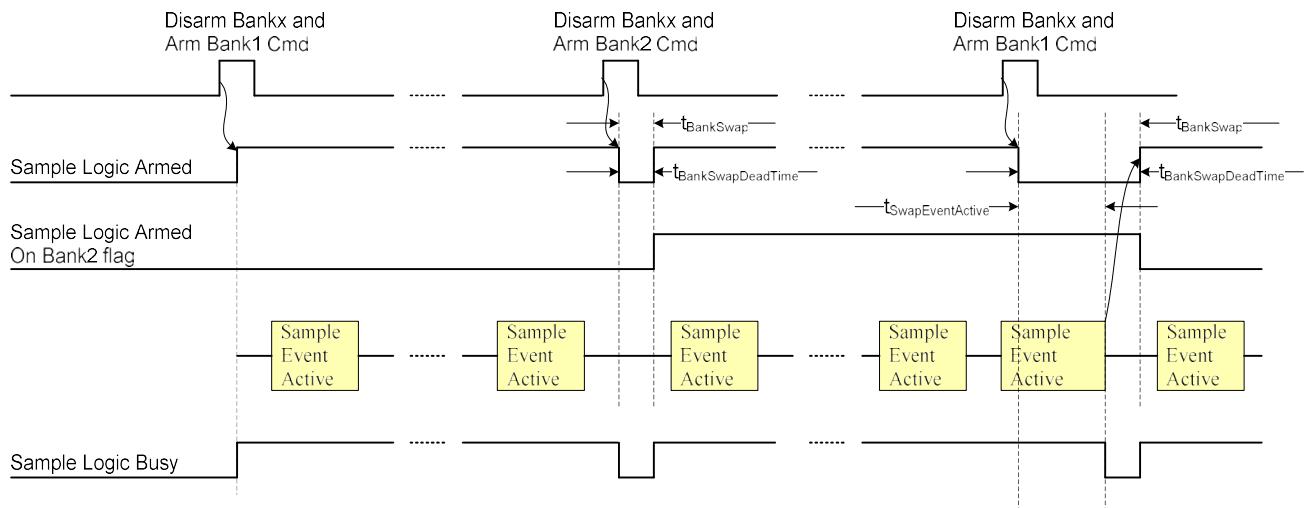
|                                                                                                                            |                                                              |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Information [7:0]                                                                                                          | Status Flag<br>until ADC FPGA Versions V0125-0009/V0250-0009 |
| - bit 7: Overflow flag<br>- bit 6: Underflow flag<br>- bit 5: RePileup flag<br>- bit 4: Pileup flag<br>- bit 5-0: reserved | Status Flag = RePileup or Pileup Flag                        |

| Status Flag<br>from ADC FPGA Versions V0125-000A/V0250-000A |                          |                        |                    |
|-------------------------------------------------------------|--------------------------|------------------------|--------------------|
| Internal Trigger enabled                                    | External Trigger enabled | Pileup Trigger enabled | Status Flag        |
| 1                                                           | x                        | x                      | RePileup or Pileup |
| 0                                                           | 1                        | x                      | Internal Trigger   |
| 0                                                           | 0                        | 1                      | 1                  |

## 4.7 Dead time

The overall dead time of the Sample logic consists of the dead time of the Sample Bank Swap logic plus the dead time of the Hit/Event storage logic.

### 4.7.1 Sample Bank Swap logic dead time



The Bank Swap dead time ( $t_{BankSwapDeadTime}$ ) is the sum of the Bank Swap time ( $t_{BankSwap}$ ) and a possible Swap Event Active time ( $t_{SwapEventActive}$ ).

The Bank Swap time ( $t_{BankSwap}$ ) is constant:  $t_{BankSwap} = 36$  sample clocks.

| Sample clock | Bank Swap time<br>$t_{BankSwap}$          |
|--------------|-------------------------------------------|
| 250 MHz      | $36 \times 4 \text{ ns} = 144 \text{ ns}$ |
| 125 MHz      | $36 \times 8 \text{ ns} = 288 \text{ ns}$ |

The Swap Event Active time ( $t_{swapEventActive}$ ) depends on the Event Active time and on the moment of the “Disarm Bankx and Arm Banky cmd” while at least one channel is busy to save an Event (Sample Event Active). The Swap Event Active time ( $t_{SwapEventActive}$ ) can vary between 0ns and the maximum of the Event Active ( $t_{EventActive}$ ).

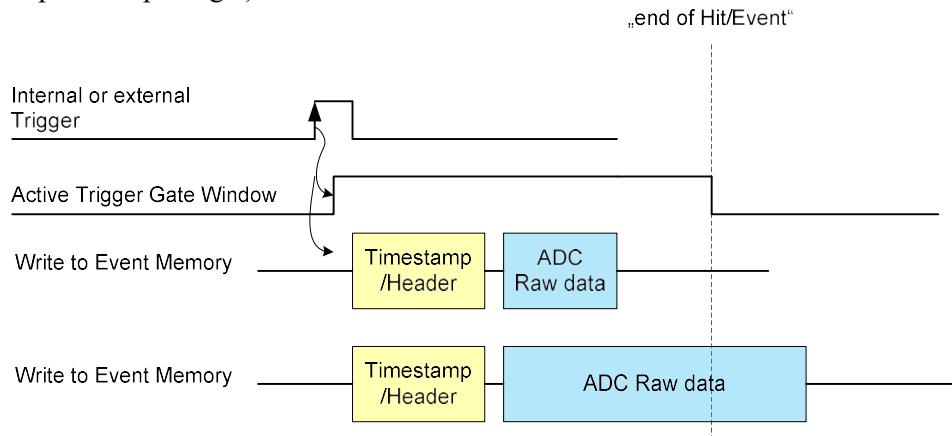
These timings can be checked with the NIM outputs CO, TO and UO with a corresponding setting of the registers “LEMO Out “CO” Select”, “LEMO Out “TO” Select” and “LEMO Out “UO” Select”.

#### 4.7.2 Hit/Event storage dead time

The Hit/Event storage dead time is defined from the moment of the “end of Hit/Event” until the logic is ready to save the next event.

The moment of the end of a “Hit/Event” is the end of the “Active Trigger Gate Window” independently of the “Raw data Sample length”.

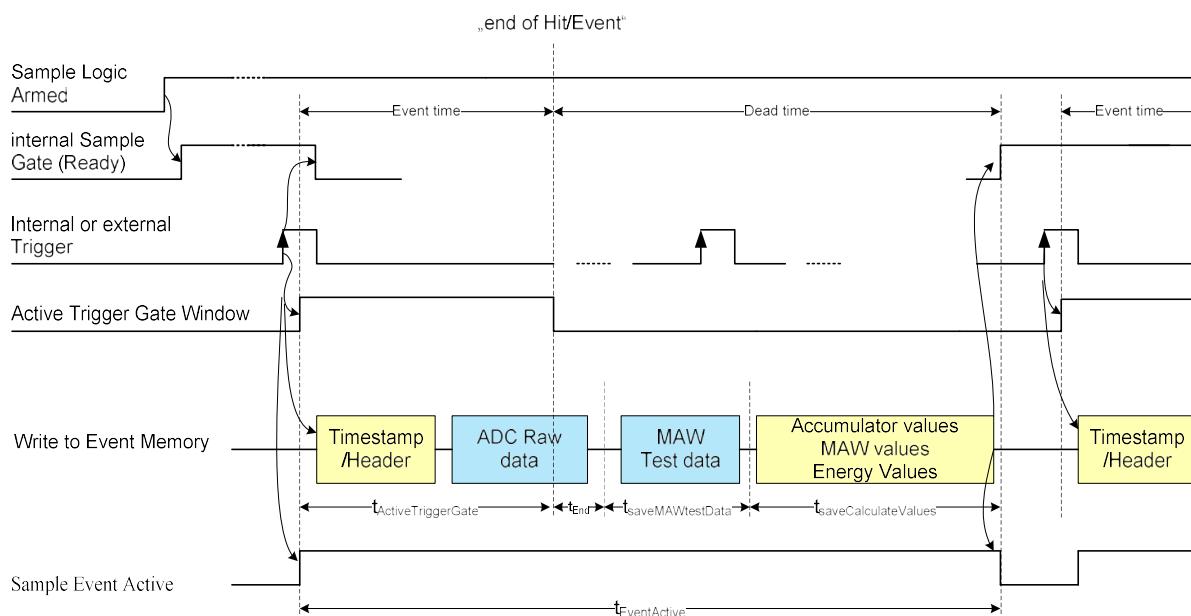
All calculations (Accumulators, Energy) are performed while the “Active Trigger Gate Window” is valid (see chapter Sample logic).



The point in time at which the Sample logic is ready (internal Sample Gate) to take a new hit/event depends on the requested Data Format and whether the option “Chx Save MAW Test Buffer” is enabled.

The Hit/Event storage dead time is the sum of  $t_{End}$ ,  $t_{saveMAVtestData}$  and  $t_{saveCalculateValues}$ .

The Event Active time is the sum of  $t_{ActiveTriggerGate}$  and the Hit/Event storage dead time.



The Sample Event Active time can be checked for example with the NIM output UO with a corresponding setting of the register “LEMO Out “UO” Select” (only bit 4 is set).

The time  $t_{End}$  depends on the sample length and the number of samples that are written to memory after the Active Trigger Gate window becomes invalid.

A table with  $t_{End}$  values for the 250 MHz case can be found below:

| Active Trigger<br>Gate window<br>length | Sample length<br>+<br>Start Index | $t_{End}$ (appr.)          |                                        |
|-----------------------------------------|-----------------------------------|----------------------------|----------------------------------------|
|                                         |                                   | odd or even channel only   | odd and even channel<br>simultaneously |
| 100                                     | 0                                 | 64 ns                      | 64 ns                                  |
| 100                                     | 50                                | 64 ns                      | 64 ns                                  |
| 100                                     | 70                                | 64 to 128 ns               | 64 to 256 ns                           |
| 100                                     | 100                               | 250 to 470 ns              | 456 to 700 ns                          |
|                                         |                                   |                            |                                        |
| 500                                     | 0                                 | 64 ns                      | 64 ns                                  |
| 500                                     | 400                               | 64 ns                      | 64 to 256 ns                           |
| 500                                     | 500                               | 250 to 470 ns              | 500 to 756 ns                          |
| 500                                     | 1000                              | 2.2 $\mu$ s to 2.4 $\mu$ s | 2.5 $\mu$ s to 2.75 $\mu$ s            |
|                                         |                                   |                            |                                        |
| 1000                                    | 0                                 | 64 ns                      | 64 ns                                  |
| 1000                                    | 800                               | 64 ns                      | 64 ns                                  |
| 1000                                    | 900                               | 64 ns                      | 64 to 256 ns                           |
| 1000                                    | 1000                              | 250 to 470 ns              | 500 to 756 ns                          |

A table with  $t_{saveCalculateValues}$  values (which are frequency independent) can be found below:

| Data Format             | $t_{saveCalculateValues}$ |
|-------------------------|---------------------------|
| bits 3 to 0 are cleared | 0 ns                      |
| bit 0 is set            | + 36 ns                   |
| bit 1 is set            | + 12 ns                   |
| bit 2 is set            | + 16 ns                   |
| bit 3 is set            | + 12 ns                   |
| bits 3 to 0 are set     | 76 ns                     |

A table with  $t_{saveMAVtestData}$  values (which are frequency independent) can be found below:

| MAW sample length | $t_{saveMAVtestData}$      |                                        |
|-------------------|----------------------------|----------------------------------------|
|                   | odd or even channel only   | odd and even channel<br>simultaneously |
| 0                 | 0 ns                       | 0 ns                                   |
| 100               | 460 to 660 ns              | 720 to 980 ns                          |
| 500               | 2.9 $\mu$ s to 3.3 $\mu$ s | 3.6 $\mu$ s to 4.2 $\mu$ s             |
| 1000              | 5.9 $\mu$ s to 6.3 $\mu$ s | 7.3 $\mu$ s to 8.0 $\mu$ s             |

Calculation:

Event time:  $t_{\text{ActiveTriggerGate}}$

Dead time:  $(t_{\text{End}} + t_{\text{saveMAWtestData}} + t_{\text{saveCalculateValues}})$

Event Active time = Event time + Dead time

or

$t_{\text{EventActive}} = t_{\text{ActiveTriggerGate}} + (t_{\text{End}} + t_{\text{saveMAWtestData}} + t_{\text{saveCalculateValues}})$

### Example 1:

Sample frequency: 250 MHz

Active Trigger Gate window length: 1000 (4  $\mu$ s)

sample raw data length: 400

sample raw data start index: 0

MAW sample length: 0

Data Format bits 3 to 0: 0x1 (bit 0 = 1)

$t_{\text{EventActive}} = t_{\text{ActiveTriggerGate}} + (t_{\text{End}} + t_{\text{saveMAWtestData}} + t_{\text{saveCalculateValues}})$

$t_{\text{EventActive}} = 4 \text{ us} + (64 \text{ ns} + 0 \text{ ns} + 36 \text{ ns}) = 4100 \text{ ns}$

Event time = 4000 ns

Dead time = 100 ns

Event Active time = 4100 ns

### Example 2: (even and odd channels simultaneously)

Sample frequency: 250 MHz

Active Trigger Gate window length: 1000 (4  $\mu$ s)

sample raw data length: 1000

sample raw data start index: 0

MAW sample length: 1000

Data Format bits 3 to 0: 0x7 (bit 0 = 1, bit 1 = 1, bit 2 = 1)

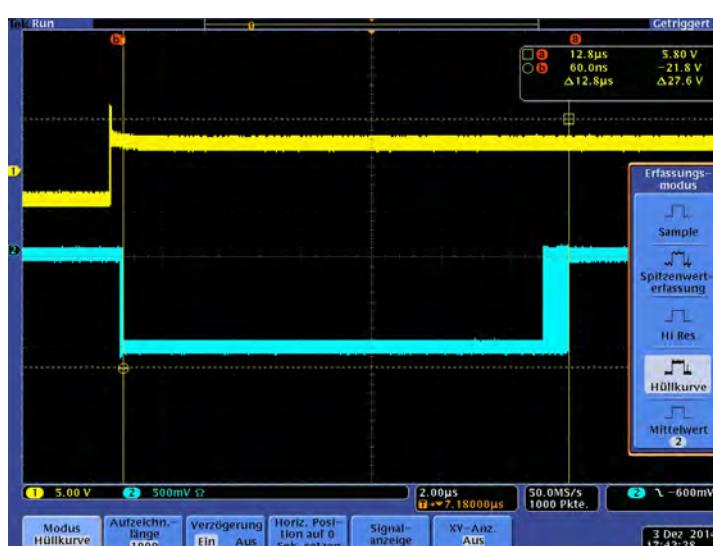
$t_{\text{EventActive}} = t_{\text{ActiveTriggerGate}} + (t_{\text{End}} + t_{\text{saveMAWtestData}} + t_{\text{saveCalculateValues}})$

$t_{\text{EventActive}} = 4 \text{ us} + ((500 \text{ to } 756 \text{ ns}) + (7.3 \text{ to } 8.0 \text{ us})) + 64 \text{ ns} = 11864 \text{ to } 12800 \text{ ns}$

Event time = 4000 ns

Dead time = 7864 to 8800ns

Event Active time = 11864 to 12800 ns



1 (yellow): trigger  
2 (blue): Event Active time

## 4.8 Memory Organization

Each ADC FPGA group (4 ADC channels) has physically two DDR3-memory-chips 256MByte each → 512 MByte for four channels → 128 MByte for each channel.

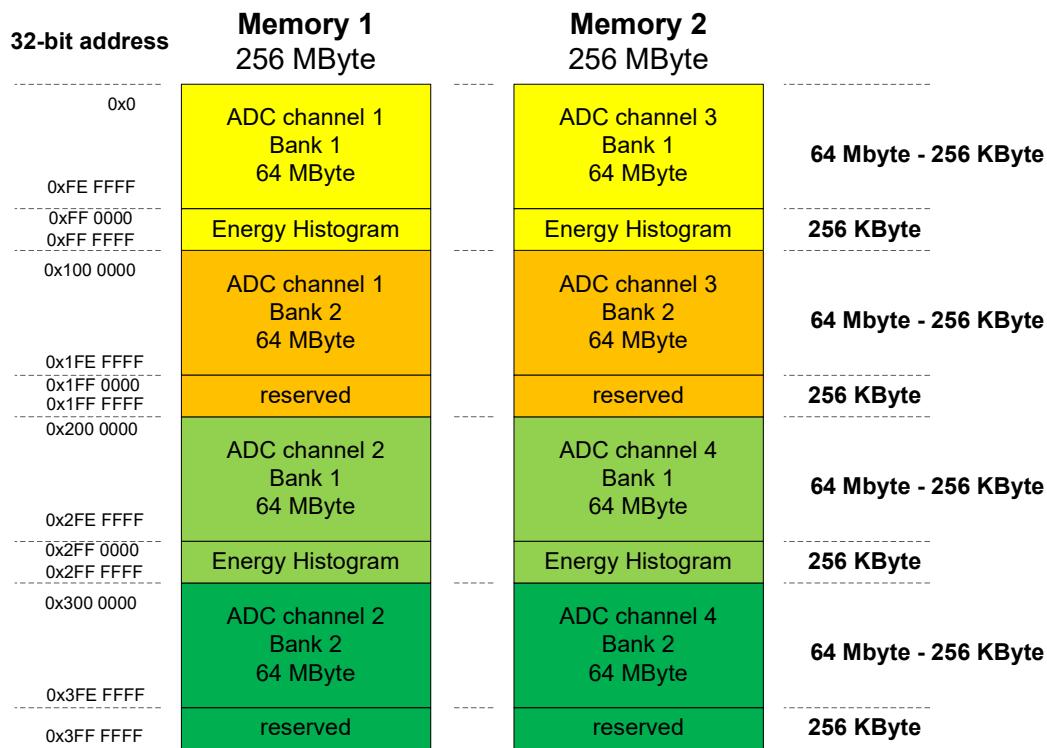
Two ADC data storage modes (hits/events) are foreseen:

- Double Bank mode
- Single Bank mode (reserved, not implemented yet)

The Energy Histogram Memory for each channel has a length of 64K bins and is implemented in the last 256Kbyte page of the Memory “ADC channel x Bank1”.

### 4.8.1 Double Bank mode

In Double Bank mode two banks of 64 MByte each are dedicated to each ADC channel.



The sample logic writes the Hits/Events with each trigger (internal or external) continuously to the “armed Bank”. The memory address logic generates a veto signal if the 64 MByte Bank is almost full (64 MByte – 512 KByte). In this case following triggers are ignored to prevent the overwriting of the following memory space (next Bank or channel Bank).

### 4.8.2 Single Bank mode

The Single Bank mode has to be defined !

#### 4.9 Trigger Statistic Counter

The following six 32-bit Statistic counters are implemented for each channel

- **Internal trigger counter:** counts all triggers, which are generated by the internal Trigger logic
- **Hit trigger counter:** counts hit triggers → internal triggers, which triggered a Hit/Event sampling
- **Dead time trigger counter:** counts triggers, in a time slot, in which the sample logic is not ready to save the Hit/Event (bank switching, memory bank almost full, write Test-Maw data, writing rest of ADC data in case of simultaneously readout)
- **Pileup trigger counter:** counts triggers during the sample logic is busy
- **Veto trigger counter:** counts triggers while a Veto signal is valid
- **He trigger counter:** counts suppressed High Energy triggers

All counters will be cleared with a “Timestamp-Clear” command/signal.

Two modes are implemented to define the moment of latching the counters.

See Trigger Statistic Counter Mode register bit 0 (Update Mode).

|                  |                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| Update Mode = 0: | Readout of the actual Trigger-Statistic-Counters                                                                        |
| Update Mode = 1: | Readout of the latched Trigger-Statistic-Counters                                                                       |
|                  | The Readout-Trigger-Statistic-Counter-Latches will be latched with each bank switching (at the end of a bank sampling). |

Statistic Counter Data Format:

|                                    |
|------------------------------------|
| Ch1 Internal trigger counter[31:0] |
| Ch1 Hit trigger counter[31:0]      |
| Ch1 Deadtime trigger counter[31:0] |
| Ch1 Pileup trigger counter[31:0]   |
| Ch1 Veto trigger counter[31:0]     |
| Ch1 He trigger counter[31:0]       |
| Ch2 Internal trigger counter[31:0] |
| Ch2 Hit trigger counter[31:0]      |

• •  
• •  
• •

|                                |
|--------------------------------|
| Ch4 Veto trigger counter[31:0] |
| Ch4 He trigger counter[31:0]   |

The “ADC FPGA Data Transfer” Logic has to be used to read/get the Trigger Statistic Counters.

Example: read Trigger Statistic Counter

```
// start readout FSM
i_adc=0; // only channel 1 to ch4
vme_crate->vme_A32D32_write ( module_base_addr +
    SIS3316_DATA_TRANSFER_CH1_4_CTRL_REG + (i_adc*4) ,
    0x80000000 + 0x30000000 ); // Space = Statistic counter

// read from FIFO
req_nof_32bit_words = 24 ;
vme_crate->vme_A32BLT32FIFO_read ( module_base_addr
    +SIS3316_FPGA_ADC1_MEM_BASE
    + (i_adc*SIS3316_FPGA_ADC_MEM_OFFSET), gl_rblt_data,
    req_nof_32bit_words, &got_nof_32bit_words); //

printf("\n");
printf("    All      Hits/Events  Deadtime   Pileup     Veto      High-Energy suppressed\n");
for (i_ch = 0; i_ch < 4;i_ch++) {
    printf("ch%2d    ", i_ch+1);
    for (i=0;i<6;i++) {
        printf("0x%08x   ", gl_rblt_data[(i_ch*6) + i]);
    }
    printf("\n");
}
}
```

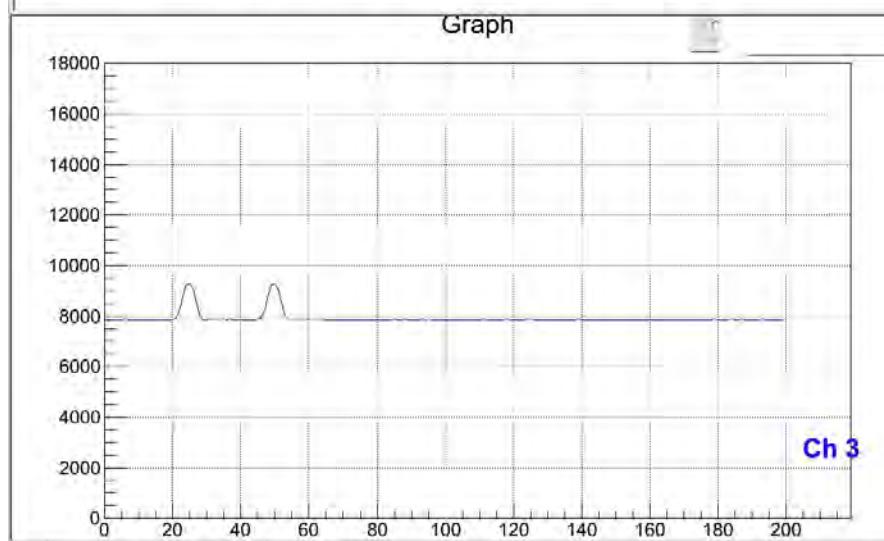
```
ROOT session
Bank buffer counter = 130
ch1 event counter = 0
ch2 event counter = 0
ch3 event counter = 40795
ch4 event counter = 0

    All      Hits/Events  Deadtime   Pileup     Veto      High-Energy suppressed
ch 1  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
ch 2  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
ch 3  0x00013fb  0x00000000  0x00000000  0x00000000  0x00000000
ch 4  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000

SIS3316 KEY_DISARM AND ARM_BMM2
ch1 last bank address = 0x00000000
ch2 last bank address = 0x02000000
ch3 last bank address = 0x01002942
ch4 last bank address = 0x03000000

Bank buffer counter = 131
ch1 event counter = 0
ch2 event counter = 0
ch3 event counter = 40996
ch4 event counter = 0

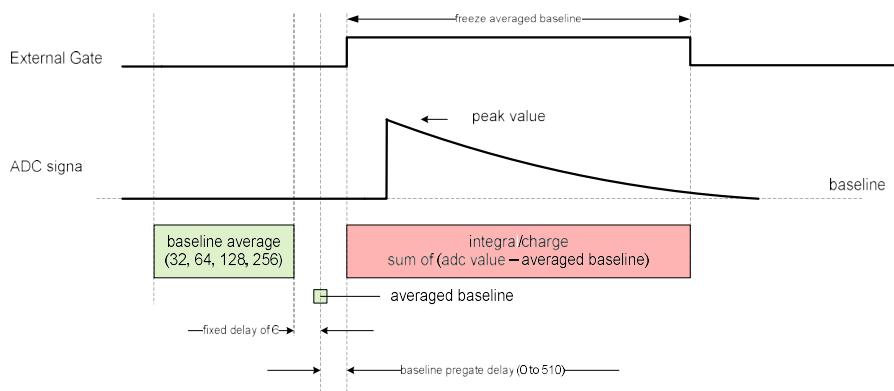
    All      Hits/Events  Deadtime   Pileup     Veto      High-Energy suppressed
ch 1  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
ch 2  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
ch 3  0x00013fb0  0x00000000  0x00000000  0x00000000  0x00000000
ch 4  0x00000000  0x00000000  0x00000000  0x00000000  0x00000000
```



## 4.10 Peak/Charge ADC mode (Peak/Charge Sensing ADC)

The peak and charge ADC mode of operation is available starting with ADC firmware version 0x02500005 and higher (SIS3316-250MHz-14bit, only).

An external gate (connected to front panel input TI) is used to define the area in which the signal maximum (peak height) and the charge are computed. The averaged baseline is subtracted in both cases. The baseline average can be 32, 64, 128 or 256 samples long (defined by the two baseline average mode bits) and the baseline pregate delay defines how many samples ahead of the leading edge of the gate the moving baseline average is frozen and stored to the event header. A fixed delay of 6 samples has to be taken into account as shown below.



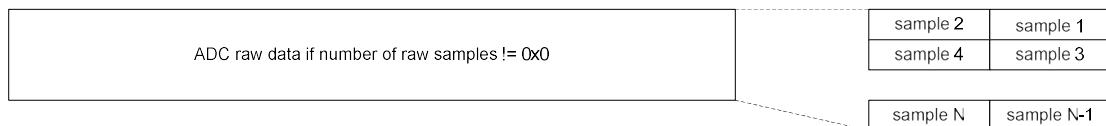
### 4.10.1 SIS3316 Peak/Charge Data Format

The data format in peak/charge mode of operation is illustrated below. If you choose a non zero value for the number of raw samples you have the option to get raw data besides the peak height and the charge integral. This may be of particular interest for signal processing of the leading edge of the signal or to compare the computed/stored baseline value to the real situation at the beginning of the gate. The timestamp information is stored at the leading edge of the gate.

| 31                    | 16 15                      | 4 3              | 0 |
|-----------------------|----------------------------|------------------|---|
| Timestamp [47:32]     | Channel ID                 | Format bits      |   |
| Timestamp [31:16]     |                            | Timestamp [15:0] |   |
| Baseline value [15:0] | Peak height value [15:0]   |                  |   |
|                       | Lower Charge value [31:0]  |                  |   |
|                       | Upper Charge value [63:32] |                  |   |

If Peak/ChargeMode = 1

| 31-28 | 27            | 26                      | 25-0                                              |
|-------|---------------|-------------------------|---------------------------------------------------|
| 0xE   | MAW Test Flag | RePileup or Pileup Flag | number of raw samples (x 2 samples, 32-bit words) |



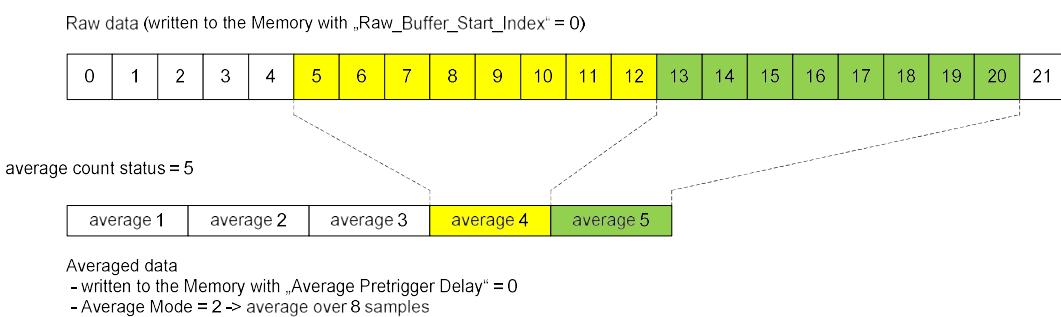
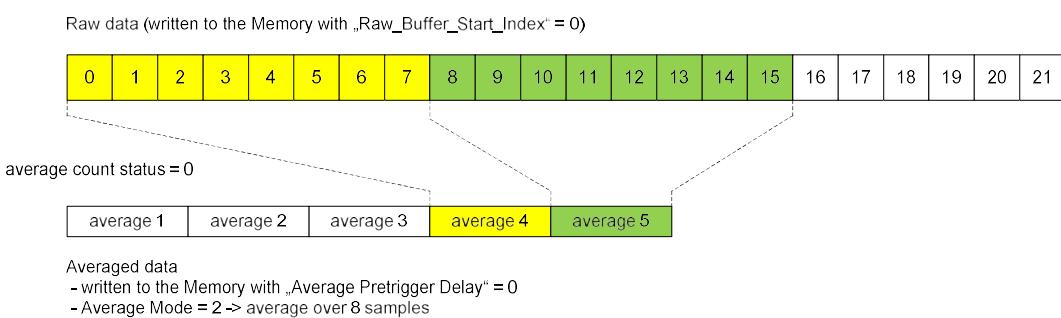
#### **4.11 Averaging mode (SIS3316-125MHz-16bit, only)**

Averaging mode is available in ADC firmware version 0x01250004 and higher (SIS3316-125MHz-16bit, only).

Originally this mode was implemented for the maXs (Micro-Calorimeter Arrays for High Resolution X-ray Spectroscopy) application. You can average 4, 8, 16, 32, 64, 128 or 256 samples (defined by the setting of the three average mode bits), program a 12-bit wide average pretrigger delay and define a 16-bit wide number of averaged samples (average sample length) to be acquired. The internal clock range from 10 MHz to 125 MHz corresponds to a sampling speed of some 40 KSPS to 500 KSPS in combination with 256 sample averaging. The average count status information in the data allows for timing of the first sample of the first averaged value relative to the trigger as illustrated below. The averaged values are shifted to 16-bit to enable storage of two averaged samples in one 32-bit data word.

Especially with a higher number of samples to be averaged, the effective number of bits (ENOB) is virtually equal to the nominal number of bits.

See “Average Configuration registers (SIS3316-125MHz-16bit)”.

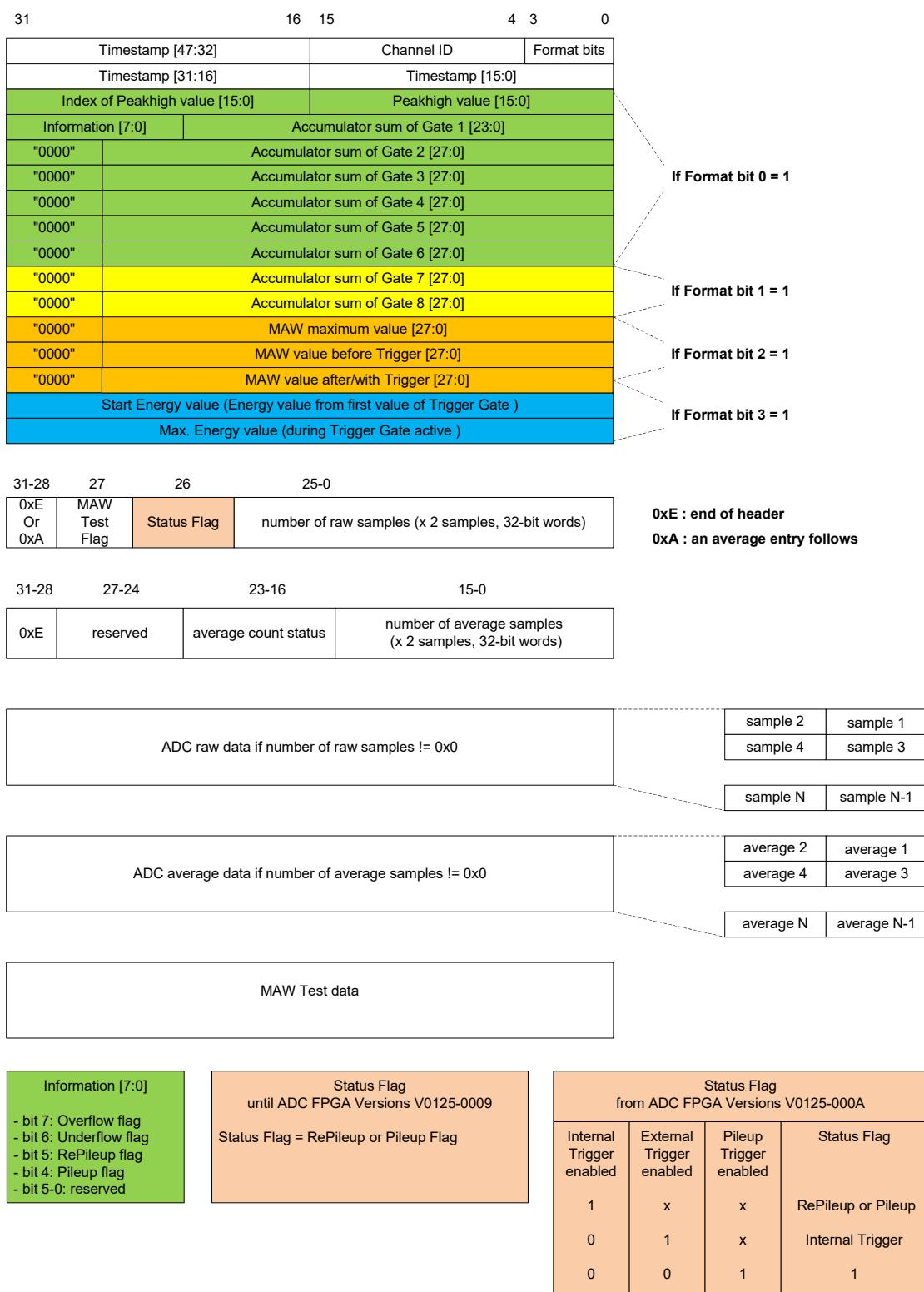


#### 4.11.1 SIS3316 Average Data Format (SIS3316-125MHz-16bit, only)

The data format with averaging mode is illustrated below.

You can store raw data in addition(\*) to averaged data to allow for higher resolution sampling at the leading edge of the signal e.g..

\* Note: the raw data sample length must be less than 1K x Average factor.

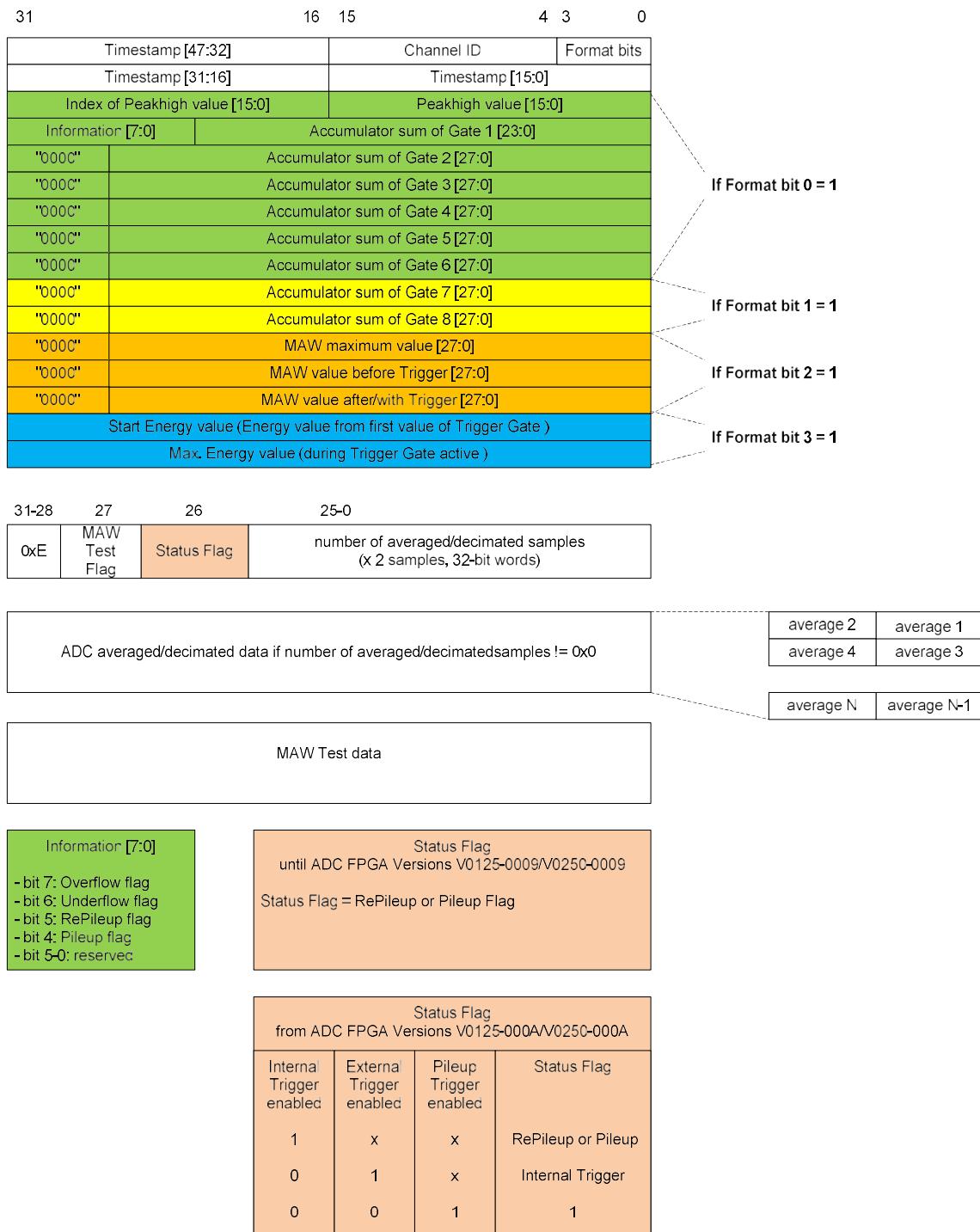


#### 4.12 Average/Decimation mode (SIS3316-250MHz-14bit, only)

Decimation/Average mode is available in ADC firmware version 0x0250000E and higher (SIS3316-250MHz-14bit, only). Average/Decimation mode is implemented to reduce the number of saved data and to improve the signal to noise ratio in lower speed digitization applications. N (2, 4, 8, 16, ..., 512) consecutive samples are averaged in the FPGAs.

The data format with Average/Decimation mode is illustrated below:

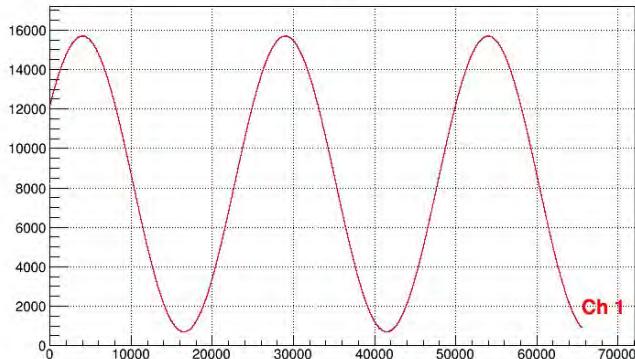
See “Average/Decimation Configuration registers (SIS3316-250MHz-14bit)”.



#### 4.12.1 Average/Decimation Example

Signal: 10 KHz Sine -> period = 100us -> 25.000 samples (sample rate: 250MHz)

Graph



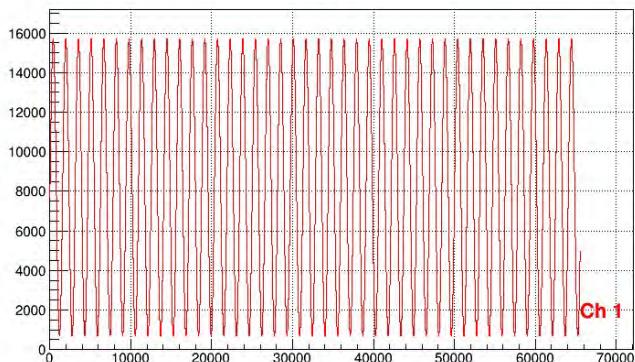
No Average/Decimation

Sine period = 100us -> 25.000 samples

`.. CH1_4_AVERAGE_CONFIGURATION_REG  
= 0xXXYYZZ00`

XX: Ch4  
YY: Ch3  
ZZ: Ch2

Graph

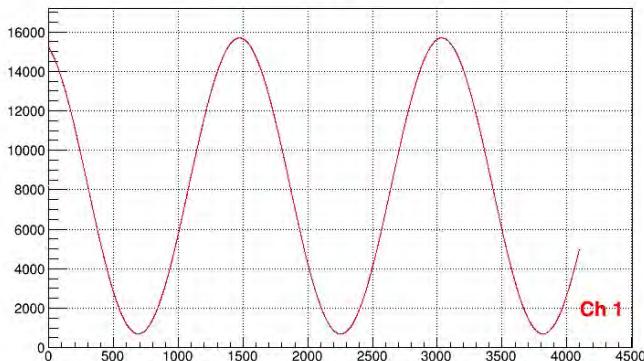


Average/Decimation = 16

Sine period = 100us -> 1562 samples

`.. CH1_4_AVERAGE_CONFIGURATION_REG  
= 0xXXYYZZ12`

Graph

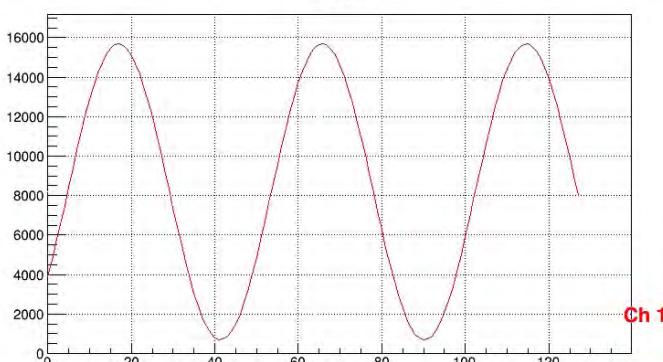


Average/Decimation = 16

Sine period = 100us -> 1562 samples

`.. CH1_4_AVERAGE_CONFIGURATION_REG  
= 0xXXYYZZ12`

Graph



Average/Decimation = 512

Sine period = 100us -> 48 samples

`.. CH1_4_AVERAGE_CONFIGURATION_REG  
= 0xXXYYZZ17`

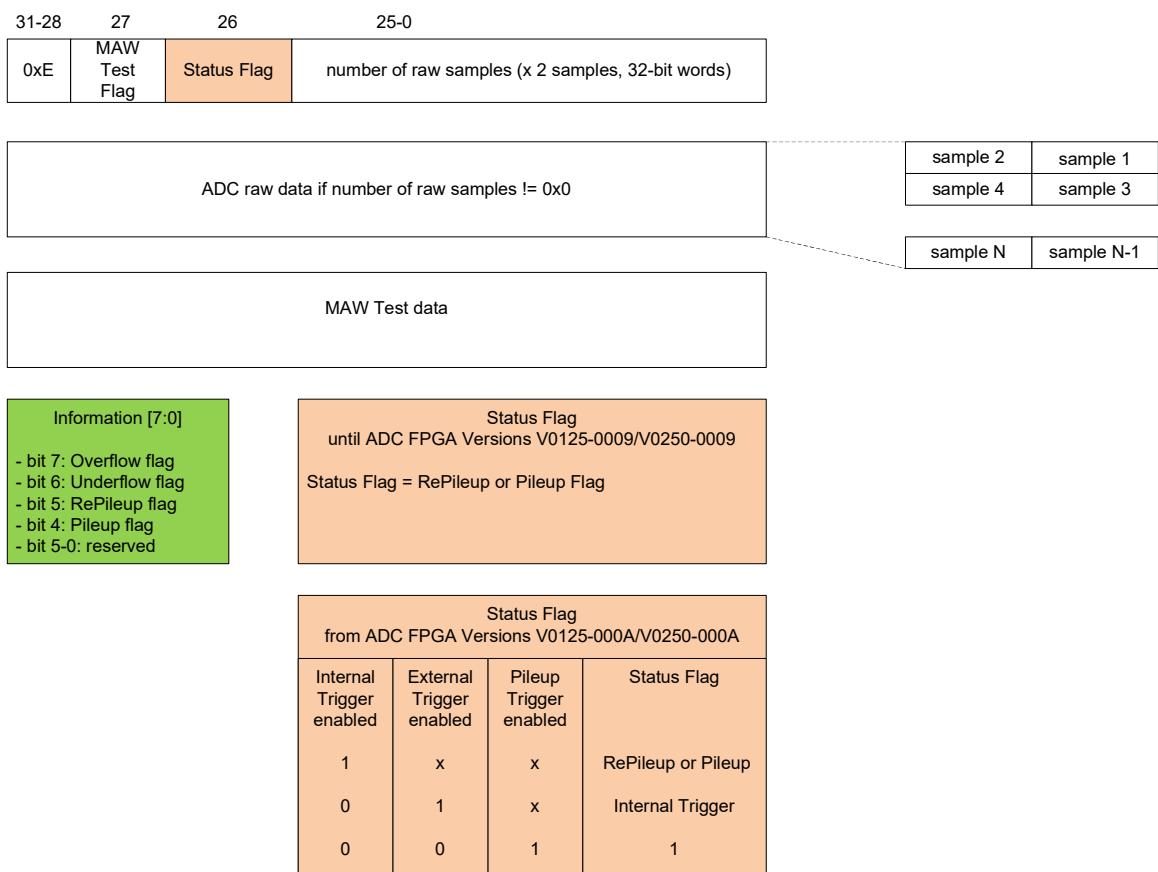
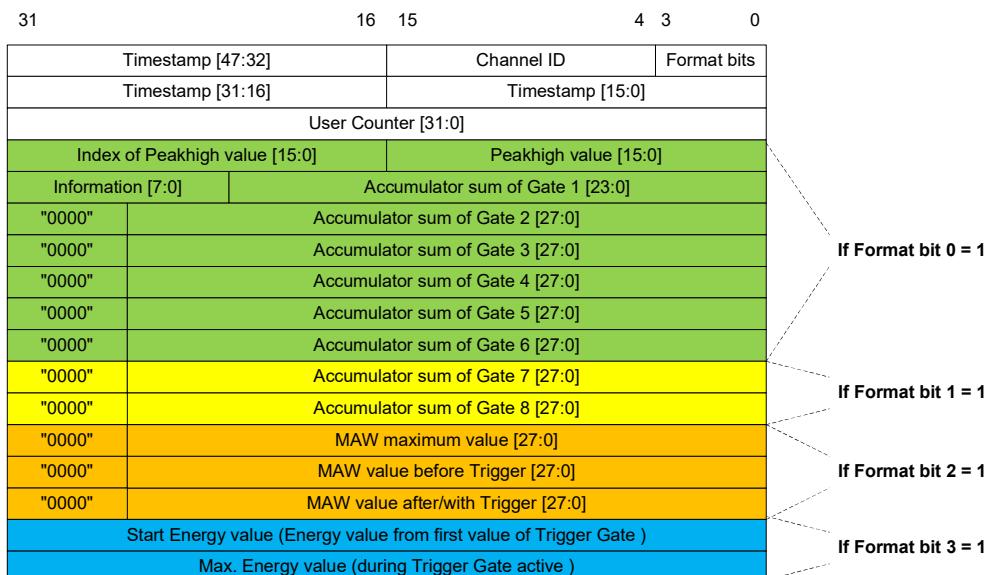
#### 4.13 “User Counter” (SIS3316-250MHz-14bit, only)

Ch4 Save “User Counter” value

An additional 32-bit Header word will be added if the “Save User Counter value” bit is set in the “Data Format Configuration” registers.

The “User Counter” will be cleared with the NIM Input UI and incremented with the Input TI if the “User Counter” logic is enabled in the “NIM Input Control” register.

The data format with “User Counter” logic is enabled is illustrated below.



## 5 VME Addressing

The base address is defined by the selected addressing mode, which is selected by jumper array SW80 and SW1 and SW2 (in non geographical mode).

The table below summarises the possible base address settings.

| SW80 Setting |        | Bits              |    |    |    |     |    |    |    |
|--------------|--------|-------------------|----|----|----|-----|----|----|----|
| EN_A32       | EN_GEO | 31                | 30 | 29 | 28 | 27  | 26 | 25 | 24 |
| x            |        | SW1               |    |    |    | SW2 |    |    |    |
|              | x      | To be implemented |    |    |    |     |    |    |    |

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

### Notes:

- This concept allows the use of the SIS3316 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=4, SW2=1 (i.e. the module will react to A32 addressing under address 0x41000000). With more than one unit shipped in one batch a set of addresses (like 0x41000000, 0x42000000, 0x43000000,...) may be used also.

## 5.1 Address Map Overview

The SIS3316 resources and their locations are listed in the tables below.

| Offset              | BLT | Access | AM | Function                                              |
|---------------------|-----|--------|----|-------------------------------------------------------|
| 0x000000 - 0x00001C | -   | W/R    | *  | VME FPGA interface registers                          |
|                     |     |        |    |                                                       |
| 0x000020 - 0x0000FC | -   | W/R    | *  | VME FPGA registers                                    |
| 0x000400 - 0x00043C | -   | W only | *  | VME FPGA key addresses (with Broadcast functionality) |
|                     |     |        |    |                                                       |
| 0x001000 - 0x001FFC | -   | R/W    | *  | ADC FPGA 1: ch1-ch4 registers                         |
| 0x002000 - 0x002FFC | -   | R/W    | *  | ADC FPGA 2: ch5-ch8 registers                         |
| 0x003000 - 0x003FFC | -   | R/W    | *  | ADC FPGA 3: ch9-ch12 registers                        |
| 0x004000 - 0x004FFC | -   | R/W    | *  | ADC FPGA 4: ch13-ch16 registers                       |
|                     |     |        |    |                                                       |
| 0x100000 - 0x1FFFFC | x   | R/W    | ** | ADC FPGA 1: ch1-ch4 Memory Data FIFO                  |
| 0x200000 - 0x2FFFFC | x   | R/W    | ** | ADC FPGA 2: ch5-ch8 Memory Data FIFO                  |
| 0x300000 - 0x3FFFFC | x   | R/W    | ** | ADC FPGA 3: ch9-ch12 Memory Data FIFO                 |
| 0x400000 - 0x4FFFFC | x   | R/W    | ** | ADC FPGA 4: ch13-ch16 Memory Data FIFO                |
|                     |     |        |    |                                                       |

Supported VME address modifier (AM)

- \* 0x09 A32 non privileged data access, D32 (32-bit only)
- \* 0x0D A32 supervisor data access, D32 (32-bit only)
- \*\* 0x0B A32 non privileged block transfer, D32 BLT (32-bit only)
- \*\* 0x0F A32 supervisor block transfer, D32 BLT (32-bit only)
- \*\* 0x08 A32 non privileged 64-bit block transfer, D64 MBLT
- \*\* 0x0C A32 supervisor 64-bit block transfer, D64 MBLT
- \*\* 0x20 A32 2eVME 6U D64 2eVME, 2eSST160, 2eSST267, 2eSST320

- Note 1:**
- Read/Write access to the VME FPGA interface register space is always possible.
  - Read access to the VME FPGA register space is always possible.
  - Write access to the VME FPGA register and “key address” space is only possible if the “Link” interface has no grant.
  - Read/Write access to the ADC FPGA register and memory space is only possible if the “Link” interface has no grant.

- Note 2:** Write access to a key address (KA) with arbitrary datum invokes the respective action

### 5.1.1 VME FPGA interface registers

| Offset     | Access | Function                                             |
|------------|--------|------------------------------------------------------|
| 0x00000000 | W/R    | Control/Status Register (J-K register)               |
| 0x00000004 | R only | Module Id. and Firmware Revision register            |
| 0x00000008 | R/W    | Interrupt configuration register                     |
| 0x0000000C | R/W    | Interrupt control register                           |
|            |        |                                                      |
| 0x00000010 | R/W    | Interface Access Arbitration control/status register |
| 0x00000014 | R/W    | CBLT/Broadcast Setup register                        |
| 0x00000018 | R/W    | Internal test                                        |
| 0x0000001C | R/W    | Hardware Version register                            |

### 5.1.2 VME FPGA registers

| Offset      | Access | Function                                                             |
|-------------|--------|----------------------------------------------------------------------|
| 0x000000020 | R      | Temperature register                                                 |
| 0x000000024 | R/W    | Onewire EEPROM control/status register (EEPROM DS2430)               |
| 0x000000028 | R      | Serial number register                                               |
| 0x00000002C | R/W    | Internal data transfer speed setting register                        |
| 0x000000030 | R/W    | ADC FPGAs Boot Controller                                            |
| 0x000000034 | R/W    | SPI Flash Control/Status                                             |
| 0x000000038 | R/W    | SPI Flash Data                                                       |
| 0x00000003C | R/W    | External Veto/Gate Delay register                                    |
| 0x000000040 | R/W    | Programmable ADC Clock: Oscillator I2C register                      |
| 0x000000044 | R/W    | Programmable MGT1 Clock: Oscillator I2C register                     |
| 0x000000048 | R/W    | Programmable MGT2 Clock: Oscillator I2C register (option, not used)  |
| 0x00000004C | R/W    | Programmable DDR3 Clock: Oscillator I2C register (option, not used)  |
| 0x000000050 | R/W    | ADC Sample Clock distribution control register                       |
| 0x000000054 | R/W    | External NIM Clock Multiplier (SI5325) SPI register                  |
| 0x000000058 | R/W    | FP-Bus control register                                              |
| 0x00000005C | R/W    | NIM-IN Control/Status register                                       |
| 0x000000060 | R/W    | Acquisition control/status register                                  |
| 0x000000064 | R/W    | Trigger Coincidence Lookup table Control register                    |
| 0x000000068 | R/W    | Trigger Coincidence Lookup table Address register                    |
| 0x00000006C | R/W    | Trigger Coincidence Lookup table Data register                       |
| 0x000000070 | R/W    | LEMO Out "CO" Select register                                        |
| 0x000000074 | R/W    | LEMO Out "TO" Select register                                        |
| 0x000000078 | R/W    | LEMO Out "UO" Select register                                        |
| 0x00000007C | R/W    | Internal Trigger Feedback Select register                            |
| 0x000000080 | R/W    | ADC FPGA 1: ch1-ch4 FPGA Data Transfer Control register              |
| 0x000000084 | R/W    | ADC FPGA 2: ch5-ch8 FPGA Data Transfer Control register              |
| 0x000000088 | R/W    | ADC FPGA 3: ch9-ch12 FPGA Data Transfer Control register             |
| 0x00000008C | R/W    | ADC FPGA 4: ch13-ch16 FPGA Data Transfer Control register            |
| 0x000000090 | R only | ADC FPGA 1: ch1-ch4 FPGA Data Transfer Status register               |
| 0x000000094 | R only | ADC FPGA 2: ch5-ch8 FPGA Data Transfer Status register               |
| 0x000000098 | R only | ADC FPGA 3: ch9-ch12 FPGA Data Transfer Status register              |
| 0x00000009C | R only | ADC FPGA 4: ch13-ch16 FPGA Data Transfer Status register             |
| 0x0000000A0 | R/W    | VME FPGA – ADC FPGAs Data Link Status register                       |
| 0x0000000A4 | R only | ADC FPGA SPI Busy Status register                                    |
| 0x0000000A8 | R/W    | reserved for SIS internal use: External and ADC Trigger Line IN Test |
| 0x0000000AC | R/W    | reserved for SIS internal use: VXS-Bus Line IN/OUT Test              |

|             |     |                                                                         |
|-------------|-----|-------------------------------------------------------------------------|
| 0x0000000B0 | R/W | reserved for SIS internal use: SFP I2C Interface Test                   |
| 0x0000000B4 | R/W | reserved for SIS internal use: SFP Interface Test                       |
| 0x0000000B8 | R/W | Prescaler output pulse divider register (possible use for output to UO) |
| 0x0000000BC | R/W | Prescaler output pulse length register                                  |
|             |     |                                                                         |
|             |     |                                                                         |
| 0x0000000C0 | R   | Channel 1 Internal Trigger Counter                                      |
| 0x0000000C4 | R   | Channel 2 Internal Trigger Counter                                      |
| 0x0000000C8 | R   | Channel 3 Internal Trigger Counter                                      |
| 0x0000000CC | R   | Channel 4 Internal Trigger Counter                                      |
|             |     |                                                                         |
| 0x0000000D0 | R   | Channel 5 Internal Trigger Counter                                      |
| 0x0000000D4 | R   | Channel 6 Internal Trigger Counter                                      |
| 0x0000000D8 | R   | Channel 7 Internal Trigger Counter                                      |
| 0x0000000DC | R   | Channel 8 Internal Trigger Counter                                      |
|             |     |                                                                         |
| 0x0000000E0 | R   | Channel 9 Internal Trigger Counter                                      |
| 0x0000000E4 | R   | Channel 10 Internal Trigger Counter                                     |
| 0x0000000E8 | R   | Channel 11 Internal Trigger Counter                                     |
| 0x0000000EC | R   | Channel 12 Internal Trigger Counter                                     |
|             |     |                                                                         |
| 0x0000000F0 | R   | Channel 13 Internal Trigger Counter                                     |
| 0x0000000F4 | R   | Channel 14 Internal Trigger Counter                                     |
| 0x0000000F8 | R   | Channel 15 Internal Trigger Counter                                     |
| 0x0000000FC | R   | Channel 16 Internal Trigger Counter                                     |

### 5.1.3 Key address registers

| Offset      | Size in Bytes | BLT | Access | Function                                                             |
|-------------|---------------|-----|--------|----------------------------------------------------------------------|
|             |               |     |        |                                                                      |
| 0x000000400 | 4             | -   | KA W   | VME Key Register Reset                                               |
| 0x000000404 | 4             | -   | KA W   | VME Key User function                                                |
| 0x000000408 | 4             | -   | KA W   | reserved                                                             |
| 0x00000040C | 4             | -   | KA W   | reserved                                                             |
|             |               |     |        |                                                                      |
| 0x000000410 | 4             | -   | KA W   | VME Key Arm Sample Logic (Single Bank Mode)                          |
| 0x000000414 | 4             | -   | KA W   | VME Key Disarm Sample Logic                                          |
| 0x000000418 | 4             | -   | KA W   | VME Key Trigger                                                      |
| 0x00000041C | 4             | -   | KA W   | VME Key Timestamp Clear                                              |
|             |               |     |        |                                                                      |
| 0x000000420 | 4             | -   | KA W   | VME Key Disarm Bankx and Arm Bank1                                   |
| 0x000000424 | 4             | -   | KA W   | VME Key Disarm Bankx and Arm Bank2                                   |
| 0x000000428 | 4             | -   | KA W   | VME Key Enable “Sample Bank Swap Control with NIM Input TI/UI” Logic |
| 0x00000042C | 4             | -   | KA W   | VME Key Disable Prescaler Output Pulse Divider logic                 |
|             |               |     |        |                                                                      |
| 0x000000430 | 4             | -   | KA W   | VME Key PPS latch bit clear                                          |
| 0x000000434 | 4             | -   | KA W   | VME Key Reset ADC-FPGA-Logic (DDR3-Memory, FPGA-Link Interface)      |
| 0x000000438 | 4             | -   | KA W   | VME Key ADC Clock DCM /PLL Reset                                     |
| 0x00000043C | 4             | -   | KA W   | reserved                                                             |

### 5.1.4 FPGA ADC group 1 registers (Ch1 to Ch4)

| Offset  | Size in Bytes | Access | Function                                                  |
|---------|---------------|--------|-----------------------------------------------------------|
| 0x01000 | 4             | R/W    | ADC Input Tap Delay register ch1-ch4                      |
| 0x01004 | 4             | R/W    | ADC Gain/Termination Control register ch1-ch4             |
| 0x01008 | 4             | R/W    | ADC Offset (DAC) Control register                         |
| 0x0100C | 4             | R/W    | ADC SPI Control register                                  |
|         |               |        |                                                           |
| 0x01010 | 4             | R/W    | Event Configuration register ch1-ch4                      |
| 0x01014 | 4             | R/W    | Channel Header ID register                                |
| 0x01018 | 4             | R/W    | End Address Threshold register ch1-ch4                    |
| 0x0101C | 4             | R/W    | Active Trigger Gate Window Length register                |
|         |               |        |                                                           |
| 0x01020 | 4             | R/W    | Raw Data Buffer Configuration register                    |
| 0x01024 | 4             | R/W    | Pileup Configuration register                             |
| 0x01028 | 4             | R/W    | Pre Trigger Delay register                                |
| 0x0102C | 4             | R/W    | Average Configuration register (SIS3316-16bit only)       |
|         |               |        |                                                           |
| 0x01030 | 4             | R/W    | Data Format Configuration register                        |
| 0x01034 | 4             | R/W    | MAW Test Buffer Configuration register                    |
| 0x01038 | 4             | R/W    | Internal Trigger Delay Configuration register             |
| 0x0103C | 4             | R/W    | Internal Gate Length Configuration register (Coincidence) |
|         |               |        |                                                           |
| 0x01040 | 4             | R/W    | FIR Trigger Setup Ch1                                     |
| 0x01044 | 4             | R/W    | Trigger Threshold Ch1                                     |
| 0x01048 | 4             | R/W    | High Energy Trigger Threshold Ch1                         |
| 0x0104C | 4             | R/W    | reserved                                                  |
|         |               |        |                                                           |
| 0x01050 | 4             | R/W    | FIR Trigger Setup Ch2                                     |
| 0x01054 | 4             | R/W    | Trigger Threshold Ch2                                     |
| 0x01058 | 4             | R/W    | High Energy Trigger Threshold Ch2                         |
| 0x0105C | 4             | R/W    | reserved                                                  |
|         |               |        |                                                           |
| 0x01060 | 4             | R/W    | FIR Trigger Setup Ch3                                     |
| 0x01064 | 4             | R/W    | Trigger Threshold Ch3                                     |
| 0x01068 | 4             | R/W    | High Energy Trigger Threshold Ch3                         |
| 0x0106C | 4             | R/W    | reserved                                                  |
|         |               |        |                                                           |
| 0x01070 | 4             | R/W    | FIR Trigger Setup Ch4                                     |
| 0x01074 | 4             | R/W    | Trigger Threshold Ch4                                     |
| 0x01078 | 4             | R/W    | High Energy Trigger Threshold Ch4                         |
| 0x0107C | 4             | R/W    | reserved                                                  |
|         |               |        |                                                           |

| Offset  | Size in Bytes | Access | Function                                                     |
|---------|---------------|--------|--------------------------------------------------------------|
| 0x01080 | 4             | R/W    | FIR Trigger Setup Sum[Ch1..Ch4]                              |
| 0x01084 | 4             | R/W    | Trigger Threshold Sum[Ch1..Ch4]                              |
| 0x01088 | 4             | R/W    | High Energy Trigger Threshold Sum[Ch1..Ch4]                  |
| 0x0108C | 4             | R/W    | reserved                                                     |
|         |               |        |                                                              |
| 0x01090 | 4             | R/W    | Trigger Statistic Counter Mode register [Ch1..Ch4]           |
| 0x01094 | 4             | R/W    | Peak/Charge Configuration register                           |
| 0x01098 | 4             | R/W    | Extended Raw Data Buffer Configuration register              |
| 0x0109C | 4             | R/W    | Extended Event Configuration register ch1-ch4                |
|         |               |        |                                                              |
| 0x010A0 | 4             | R/W    | Accumulator Gate 1 Configuration register                    |
| 0x010A4 | 4             | R/W    | Accumulator Gate 2 Configuration register                    |
| 0x010A8 | 4             | R/W    | Accumulator Gate 3 Configuration register                    |
| 0x010AC | 4             | R/W    | Accumulator Gate 4 Configuration register                    |
|         |               |        |                                                              |
| 0x010B0 | 4             | R/W    | Accumulator Gate 5 Configuration register                    |
| 0x010B4 | 4             | R/W    | Accumulator Gate 6 Configuration register                    |
| 0x010B8 | 4             | R/W    | Accumulator Gate 7 Configuration register                    |
| 0x010BC | 4             | R/W    | Accumulator Gate 8 Configuration register                    |
|         |               |        |                                                              |
| 0x010C0 | 4             | R/W    | FIR Energy Setup register Ch1                                |
| 0x010C4 | 4             | R/W    | FIR Energy Setup register Ch2                                |
| 0x010C8 | 4             | R/W    | FIR Energy Setup register Ch3                                |
| 0x010CC | 4             | R/W    | FIR Energy Setup register Ch4                                |
|         |               |        |                                                              |
| 0x010D0 | 4             | R/W    | Energy Histogram Configuration register Ch1                  |
| 0x010D4 | 4             | R/W    | Energy Histogram Configuration register Ch2                  |
| 0x010D8 | 4             | R/W    | Energy Histogram Configuration register Ch3                  |
| 0x010DC | 4             | R/W    | Energy Histogram Configuration register Ch4                  |
|         |               |        |                                                              |
| 0x010E0 | 4             | R/W    | MAW Start Index and Energy Pickup Configuration register Ch1 |
| 0x010E4 | 4             | R/W    | MAW Start Index and Energy Pickup Configuration register Ch2 |
| 0x010E8 | 4             | R/W    | MAW Start Index and Energy Pickup Configuration register Ch3 |
| 0x010EC | 4             | R/W    | MAW Start Index and Energy Pickup Configuration register Ch4 |
|         |               |        |                                                              |
| 0x010F0 | 4             | R/W    | Test register                                                |
|         |               |        |                                                              |
|         |               |        |                                                              |
|         |               |        |                                                              |

| Offset  | Size in Bytes | Access | Function                                                                     |
|---------|---------------|--------|------------------------------------------------------------------------------|
| 0x01100 | 4             | R      | ADC FPGA Version register                                                    |
| 0x01104 | 4             | R      | ADC FPGA Status register                                                     |
| 0x01108 | 4             | R      | ADC Offset (DAC) readback register                                           |
| 0x0110C | 4             | R      | ADC SPI readback register                                                    |
|         |               |        |                                                                              |
| 0x01110 | 4             | R      | Actual Sample address register Ch1                                           |
| 0x01114 | 4             | R      | Actual Sample address register Ch2                                           |
| 0x01118 | 4             | R      | Actual Sample address register Ch3                                           |
| 0x0111C | 4             | R      | Actual Sample address register Ch4                                           |
|         |               |        |                                                                              |
| 0x01120 | 4             | R      | Previous Bank Sample address register Ch1                                    |
| 0x01124 | 4             | R      | Previous Bank Sample address register Ch2                                    |
| 0x01128 | 4             | R      | Previous Bank Sample address register Ch3                                    |
| 0x0112C | 4             | R      | Previous Bank Sample address register Ch4                                    |
|         |               |        |                                                                              |
| 0x01130 | 4             | R      | PPS Timestamp register (bits 47 – 32)                                        |
| 0x01134 | 4             | R      | PPS Timestamp register (bits 31 – 0)                                         |
| 0x01138 | 4             | R      | Test: readback register 0x01018                                              |
| 0x0113C | 4             | R      | Test: readback register 0x0101C<br>(up to and including Version V_0xxx0006 ) |
| 0x0113C | 4             | R      | SIS Internal Test register<br>(from Version V_0xxx0007)                      |
|         |               |        |                                                                              |

## 6 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 SIS3316_CONTROL_STATUS          0x0      /* read/write; D32 */  
refers to the SIS3316.h header file
```

### 6.1 Control/Status Register(0x0, write/read)

```
#define SIS3316_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 same register represents the status register on read access.

| Bit | write Function                   | read Function                      |
|-----|----------------------------------|------------------------------------|
| 31  | Clear Reboot FPGAs (*)           | 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 Led 2 Application Mode (*) | 0                                  |
| 21  | Clear Led 1 Application Mode (*) | 0                                  |
| 20  | Clear Led U Application Mode (*) | 0                                  |
| 19  | Clear Led-Application Mode (*)   | 0                                  |
| 18  | Switch off LED 2 (*)             | 0                                  |
| 17  | Switch off LED 1 (*)             | 0                                  |
| 16  | Switch off LED U (*)             | 0                                  |
| 15  | Set Reboot FPGAs (**)            | Status Reboot FPGAs                |
| 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 Led 2 Application Mode       | Status Led 2 Application Mode      |
| 5   | Set Led 1 Application Mode       | Status Led 1 Application Mode      |
| 4   | Set Led U Application Mode       | Status Led U Application Mode      |
| 3   |                                  |                                    |
| 2   | Switch on LED 2 (***)            | Status LED 2                       |
| 1   | Switch on LED 1 (***)            | Status LED 1                       |
| 0   | Switch on LED U (***)            | Status LED U (1=LED on, 0=LED off) |

(\*) denotes power up default setting

(\*\*) provided that the switch SW80-7 is on (Watchdog enable) and that its own interface grant bit is set

(\*\*\*) provided that its own interface grant bit is set and the Led Application Mode is cleared

Reboot FPGAs = 1 will reboot all FPGAs provided that the switch SW80-7 is on (Watchdog enable) and that its own interface grant bit is set.

The meaning of the LEDs 2 are defined by the Control/Status register of the granted interface (VME or Ethernet interface) and in case of using the Ethernet interface the Link/DHCP status after power up (refer to SIS3316-M-0101-1-Vxxx-ethernet\_addendum.pdf).

The meaning of the LEDs U and 1 are defined by the Control/Status register of the granted interface (VME or Ethernet interface).

If the Link interface (Ethernet, Optical) has the permission, then the Link interface control/status register defines the meaning of the LEDs.

If the Link interface has not the permission or if the VME interface has the permission then this VME control/status register defines the meaning of the LEDs.

| LED U Application Mode | LED U is On if                                     |
|------------------------|----------------------------------------------------|
| 0                      | Status LED U = 1                                   |
| 1                      | Sample logic is sampling (write event into memory) |

| LED 1 Application Mode | LED 1 is On if                |
|------------------------|-------------------------------|
| 0                      | Status LED 1 = 1              |
| 1                      | Sample logic bankx is enabled |

| LED 2 Application Mode | LED 2 is On if             |
|------------------------|----------------------------|
| 0                      | Status LED 2 = 1           |
| 1                      | Sample logic bank 2 active |

## 6.2 Module Id. and Firmware Revision Register

```
#define SIS3316_MODID           0x4          /* read only; D32 */
```

This register reflects the module identification of the SIS3316 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     | 1       |
| 22  | Module Id. Bit 6     |         |
| 21  | Module Id. Bit 5     |         |
| 20  | Module Id. Bit 4     |         |
| 19  | Module Id. Bit 3     | 6       |
| 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 |         |

### 6.2.1 Major revision numbers

Find below a table with major revision numbers used to date

| Major revision number | Application/user |
|-----------------------|------------------|
| 0x20                  | n/Gamma          |

### 6.3 Interrupt configuration register (0x8)

```
#define SIS3316_IRQ_CONFIG           0x8          /* read/write; D32 */
```

This read/write register controls the VME interrupt behaviour of the SIS3316 ADC. Eight interrupt sources are foreseen, for the time being two of them are associated with an interrupt condition, the other condition is reserved for future use.

The interrupter type is DO8 .

#### 6.3.1 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       |
| 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

**6.4 Interrupt control register (0xC)**

```
#define SIS3316_IRQ_CONTROL           0xC          /* r/w; D32 */
```

This register controls the VME interrupt behaviour of the SIS3316 ADC. Eight interrupt sources are foreseen, for the time being two of them are associated with an interrupt condition, the others are reserved for future use.

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

The power up default value reads 0x 00000000

IRQ source 3:           reached Address Threshold (level sensitive)  
 IRQ source 2:           reached Address Threshold (edge sensitive)

IRQ source 1:           reserved  
 IRQ source 0:           reserved

## 6.5 Interface Access Arbitration Control Register

```
#define SIS3316_INTERFACE_ACCESS_ARBITRATION_CONTROL 0x10 /* r/w; D32 */
```

The Interface Access Arbitration Control register controls the permission to access to the internal SIS3316 space.

If no interface side has set the request bit, then VME has the permission to access the internal SIS3316 space.

| Bit | write Function                              | read Function                         |
|-----|---------------------------------------------|---------------------------------------|
| 31  | Kill of other interface request bit command | 0                                     |
| 30  |                                             | 0                                     |
| 29  |                                             | 0                                     |
| 28  |                                             | 0                                     |
| 27  |                                             | 0                                     |
| 26  |                                             | 0                                     |
| 25  |                                             | 0                                     |
| 24  |                                             | 0                                     |
| 23  |                                             | 0                                     |
| 22  |                                             | 0                                     |
| 21  |                                             | Status of other interface grant bit   |
| 20  |                                             | Status of own interface grant bit     |
| 19  |                                             | 0                                     |
| 18  |                                             | 0                                     |
| 17  |                                             | Status of other interface request bit |
| 16  |                                             | Status of own interface request bit   |
| 15  |                                             | 0                                     |
| 14  |                                             | 0                                     |
| 13  |                                             | 0                                     |
| 12  |                                             | 0                                     |
| 11  |                                             | 0                                     |
| 10  |                                             | 0                                     |
| 9   |                                             | 0                                     |
| 8   |                                             | 0                                     |
| 7   |                                             | 0                                     |
| 6   |                                             | 0                                     |
| 5   |                                             | 0                                     |
| 4   |                                             | 0                                     |
| 3   |                                             | 0                                     |
| 2   |                                             | 0                                     |
| 1   |                                             | 0                                     |
| 0   | Own interface request bit                   | Status of own interface request bit   |

**6.6 Broadcast setup register**

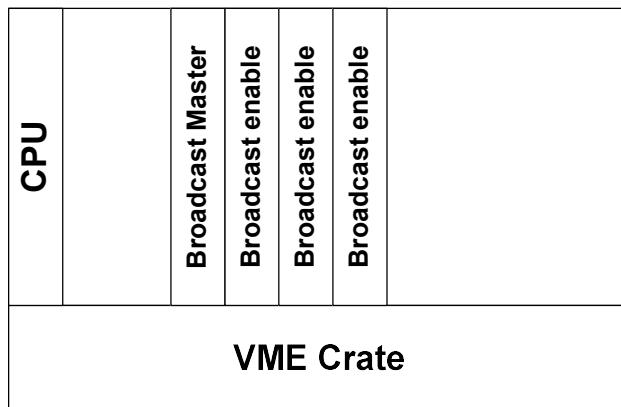
```
#define SIS3316_CBLT_BROADCAST      0x14      /* r/w; D32 */
```

This read/write register defines, whether the SIS3316 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 all modules as illustrated below.



Broadcast setup example (broadcast address 0x30000000):

| Module | Broadcast Setup Register | Comment                             |
|--------|--------------------------|-------------------------------------|
| 1      | 0x30000030               | Broadcast Master + Broadcast enable |
| 2      | 0x30000010               | Broadcast enable                    |
| 3      | 0x30000010               | Broadcast enable                    |
| 4      | 0x30000010               | Broadcast enable                    |

All 4 modules will participate in a key reset (A32/D32 write) to address 0x30000400.

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

## 6.7 Hardware Version Register

```
#define SIS3316_HARDWARE_VERSION          0x1C      /* read only; D32 */
```

This register reflects the hardware version of the SIS3316.

| Bit | Function               |
|-----|------------------------|
| 31  | 0                      |
| ..  | ..                     |
| ..  | ..                     |
| 4   | 0                      |
| 3   | Hardware Version Bit 3 |
| 2   | Hardware Version Bit 2 |
| 1   | Hardware Version Bit 1 |
| 0   | Hardware Version Bit 0 |

| Hardware Version | SIS3316 PCB version |
|------------------|---------------------|
| 1                | V1                  |
| 2                | V2 / 3              |

## 6.8 Temperature register

```
#define SIS3316_INTERNAL_TEMPERATURE_REG      0x20 /* r/w; D32 */
```

The SIS3316 is equipped with a serial 10-bit Analog Devices AD7314 temperature sensor. The temperature reading is stored in twos complement format.

Refer to the AD7314 data sheet for more detailed information.

| Bit | Read                         |
|-----|------------------------------|
| 31  | 0                            |
| ... | 0                            |
| 16  | 0                            |
| 15  | 0                            |
| ... |                              |
| 10  | 0                            |
| 9   | Temperature Data Bit 9 (MSB) |
| ..  |                              |
| 1   | Temperature Data Bit 1       |
| 0   | Temperature Data Bit 0 (LSB) |

The temperature range of the AD7314 covers -35 °C to +85 °C. The table below shows a couple of example readings.

| Temperature | Data Bit 9 . . . Bit 0 |
|-------------|------------------------|
| -25 °C      | 11 1001 1100           |
| -0.25 °C    | 11 1111 1111           |
| 0 °C        | 00 0000 0000           |
| +0.25 °C    | 00 0000 0001           |
| +10 °C      | 00 0010 1000           |
| +25 °C      | 00 0110 0100           |
| +50 °C      | 00 1100 1000           |
| +75 °C      | 01 0010 1100           |

**Note 1:** The Celsius temperature reading is obtained by casting the read datum to signed short and dividing the obtained value by 4.0 after float conversion.

**Note 2:** The unit shall be operated in an environment that results in a temperature reading of 50 °C or below. An airflow of 162 m<sup>3</sup> / h for a 21 slot crate is recommended (Schroff fan tray 10713-108 in SIS3000 crate e.g.).

**6.9 Onewire EEPROM Control register**

```
#define SIS3316_ONE_WIRE_CONTROL_REG      0x24 /* r/w; D32 */
```

Provides access to the 256-bit onboard EEPROM. The EEPROM is organized as 32 words \* 8bit.

| Bit | Write Function              | Read                     |
|-----|-----------------------------|--------------------------|
| 31  | reserved                    | EEPROM Busy              |
| 30  | reserved                    | 0                        |
| 29  | reserved                    | 0                        |
| 28  | reserved                    | 0                        |
| 27  | reserved                    | 0                        |
| 26  | reserved                    | 0                        |
| 25  | reserved                    | 0                        |
| 24  | reserved                    | Serial Number Valid Flag |
| 23  | reserved                    | Serial Number bit 15     |
| 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  | Onewire Command: RESET BUS  | ..                       |
| 9   | Onewire Command: WRITE BYTE | ..                       |
| 8   | Onewire Command: READ BYTE  | Serial Number bit 0      |
| 7   | EEPROM Write data bit 7     | EEPROM Read data bit 7   |
| 6   | EEPROM Write data bit 6     | EEPROM Read data bit 6   |
| 5   | EEPROM Write data bit 5     | EEPROM Read data bit 5   |
| 4   | EEPROM Write data bit 4     | EEPROM Read data bit 4   |
| 3   | EEPROM Write data bit 3     | EEPROM Read data bit 3   |
| 2   | EEPROM Write data bit 2     | EEPROM Read data bit 2   |
| 1   | EEPROM Write data bit 1     | EEPROM Read data bit 1   |
| 0   | EEPROM Write data bit 0     | EEPROM Read data bit 0   |

The power up default value reads 0x0

The Device Presence bit can be found in ‘EEPROM Read data bit 0’ after executing a ‘RESET BUS’ command.

A value of ‘0’ indicates that at least 1 device is present on the Onewire bus.

Programming: refer to sis3316\_class.cpp and sis3316\_class.h

```
private:  
    int owReset(int *presence);  
    int owRead(unsigned char *data);  
    int owWrite(unsigned char data);  
    int owEeReadPage(int page, unsigned char *data);  
    int owEeWritePage(int page, unsigned char *data);  
    int read_ee(int offset, int len, unsigned char *data);  
    int write_ee(int offset, int len, unsigned char *data);  
    int ow_id_ee(unsigned char *data);  
  
public:  
    int write_ow_dhcp_option(unsigned char *data);
```

Struck definition of the DS2430 contents:

| 8-bit address offset |                                   |
|----------------------|-----------------------------------|
| 0                    | Struck Serial Number (lower byte) |
| 1                    | Struck Serial Number (upper byte) |
| 2                    | Struck DHCP Option Value          |
| 3                    | Struck reserved Option Value      |
| 4                    | free                              |
| .                    | free                              |
| 31                   | free                              |

**6.10 Serial Number register**

```
#define SIS3316_SERIAL_NUMBER_REG           0x28      /* r; D32 */
```

This register holds the Serial Number of the module.

| BIT                              | access | Name                         | Function                        |
|----------------------------------|--------|------------------------------|---------------------------------|
| 31-24<br><small>FF000000</small> | RO     | DHCP Option Value            | Reserved for Ethernet interface |
| 23<br><small>00800000</small>    | RO     | 512 MByte Memory Flag        | 512 MByte Memory Flag           |
| 22-17<br><small>007E0000</small> | RO     | reserved                     |                                 |
| 16<br><small>00010000</small>    | RO     | Serial Number Not Valid Flag |                                 |
| 15-0<br><small>0000FFFF</small>  | RO     | Serial Number                | 1..65535                        |

Note: The Ethernet MAC address is 00-00-56-31-6n-nn (n-nn is the Serial Number)

512MByte Memory Flag = 0      -> 256 Mbyte DDR3-memory-chips

512MByte Memory Flag = 1      -> 512 Mbyte DDR3-memory-chips

**6.11 Internal Transfer Speed register**

```
#define SIS3316_INTERNAL_TRANSFER_SPEED_REG      0x2C      /* r/w; D32 */
```

This register can be used to change the internal data transfer speed (data link speed) between the FPGAs. Normally, it is not necessary to change it..

Depends on the hardware version the logic sets the internal data transfer speed automatically:

Hardware version = 1                           -> Data Link Speed flag = 0 -> 1.25GHz

Hardware version = 2 and higher   -> Data Link Speed flag = 1 -> 2.5GHz

| Bit | Write Function          | Read                        |
|-----|-------------------------|-----------------------------|
| 31  | reserved                | Data Link Speed flag        |
| 30  | reserved                | 0                           |
| ..  | ..                      | ..                          |
| 4   | reserved                | 0                           |
| 3   | reserved                | 0                           |
| 2   | reserved                | 0                           |
| 1   | Force to set to 2.5GHz  | Force to set to 2.5GHz bit  |
| 0   | Force to set to 1.25GHz | Force to set to 1.25GHz bit |

| hardware version | Force to set to 2.5GHz | Force to set to 1.25GHz | Internal data transfer speed                            |
|------------------|------------------------|-------------------------|---------------------------------------------------------|
| 1                | 0                      | x                       | 1.25 GHz (required for modules with serial number 1-10) |
| 1                | 1                      | x                       | 2.5 GHz (not recommend)                                 |
| 2 and higher     | 0                      | 0                       | 2.5 GHz                                                 |
| 2 and higher     | 1                      | x                       | 2.5 GHz                                                 |
| 2 and higher     | 0                      | 1                       | 1.25 GHz                                                |

**6.12 ADC FPGA Boot control register**

```
#define SIS3316_ADC_FPGA_BOOT_CSR           0x30          /* r/w; D32 */
```

| Bit | Write Function                  | Read                          |
|-----|---------------------------------|-------------------------------|
| 31  | reserved                        | 0                             |
| 30  | reserved                        | 0                             |
| 29  | reserved                        | 0                             |
| 28  | reserved                        | 0                             |
| 27  | reserved                        | 0                             |
| 26  | reserved                        | 0                             |
| 25  | reserved                        | 0                             |
| 24  | reserved                        | Boot Sequence finished        |
| 23  | reserved                        | ADC FPGA 4 done               |
| 22  | reserved                        | ADC FPGA 3 done               |
| 21  | reserved                        | ADC FPGA 2 done               |
| 20  | reserved                        | ADC FPGA 1 done               |
| 19  | reserved                        | ADC FPGA 4 error during boot  |
| 18  | reserved                        | ADC FPGA 4 error during boot  |
| 17  | reserved                        | ADC FPGA 4 error during boot  |
| 16  | reserved                        | ADC FPGA 4 error during boot  |
| 15  | reserved                        | ADC FPGA 4 boot logic running |
| 14  | reserved                        | ADC FPGA 3 boot logic running |
| 13  | reserved                        | ADC FPGA 2 boot logic running |
| 12  | reserved                        | ADC FPGA 1 boot logic running |
| 11  | reserved                        | 0                             |
| 10  | reserved                        | 0                             |
| 9   | reserved                        | 0                             |
| 8   | reserved                        | 0                             |
| 7   | reserved                        | 0                             |
| 6   | reserved                        | 0                             |
| 5   | reserved                        | 0                             |
| 4   | reserved                        | 0                             |
| 3   | reserved                        | 0                             |
| 2   | reserved                        | 0                             |
| 1   | Reset (Halt) Boot Logic command | 0                             |
| 0   | Reboot ADC FPGAs command        | 0                             |

### 6.13 SPI Flash Control/Status register

```
#define SIS3316_SPI_FLASH_CSR           0x34          /* r/w; D32 */
```

This register is used to program (update) the FPGA Firmware FlashPROM.

| Bit | Write Function           | Read                          |
|-----|--------------------------|-------------------------------|
| 31  | reserved                 | Flash logic busy              |
| 30  | reserved                 | 0                             |
| 29  | reserved                 | 0                             |
| 28  | reserved                 | 0                             |
| 27  | reserved                 | 0                             |
| 26  | reserved                 | 0                             |
| 25  | reserved                 | 0                             |
| 24  | reserved                 | 0                             |
| 23  | reserved                 | 0                             |
| 22  | reserved                 | 0                             |
| 21  | reserved                 | 0                             |
| 20  | reserved                 | 0                             |
| 19  | reserved                 | 0                             |
| 18  | reserved                 | 0                             |
| 17  | reserved                 | 0                             |
| 16  | reserved                 | 0                             |
| 15  | reserved                 | 0                             |
| 14  | reserved                 | 0                             |
| 13  | reserved                 | 0                             |
| 12  | reserved                 | 0                             |
| 11  | reserved                 | 0                             |
| 10  | reserved                 | 0                             |
| 9   | reserved                 | 0                             |
| 8   | reserved                 | 0                             |
| 7   | reserved                 | 0                             |
| 6   | reserved                 | 0                             |
| 5   | reserved                 | 0                             |
| 4   | reserved                 | 0                             |
| 3   | reserved                 | 0                             |
| 2   | SPI Flash 2 ChipSelect   | SPI Flash 2 ChipSelect status |
| 1   | SPI Flash 1 ChipSelect   | SPI Flash 1 ChipSelect status |
| 0   | Enable programming logic | Programming logic status      |

**6.14 SPI Flash Data register**

```
#define SIS3316_SPI_FLASH_DATA           0x38      /* r/w; D32 */
```

| Bit | Write Function         | Read                  |
|-----|------------------------|-----------------------|
| 31  | reserved               | 0                     |
| 30  | reserved               | 0                     |
| 29  | reserved               | 0                     |
| 28  | reserved               | 0                     |
| 27  | reserved               | 0                     |
| 26  | reserved               | 0                     |
| 25  | reserved               | 0                     |
| 24  | reserved               | 0                     |
| 23  | reserved               | 0                     |
| 22  | reserved               | 0                     |
| 21  | reserved               | 0                     |
| 20  | reserved               | 0                     |
| 19  | reserved               | 0                     |
| 18  | reserved               | 0                     |
| 17  | reserved               | 0                     |
| 16  | reserved               | 0                     |
| 15  | reserved               | 0                     |
| 14  | reserved               | 0                     |
| 13  | reserved               | 0                     |
| 12  | reserved               | 0                     |
| 11  | reserved               | 0                     |
| 10  | reserved               | 0                     |
| 9   | reserved               | 0                     |
| 8   | reserved               | 0                     |
| 7   | Flash write data bit 7 | Flash read data bit 7 |
| 6   | Flash write data bit 6 | Flash read data bit 6 |
| 5   | Flash write data bit 5 | Flash read data bit 5 |
| 4   | Flash write data bit 4 | Flash read data bit 4 |
| 3   | Flash write data bit 3 | Flash read data bit 3 |
| 2   | Flash write data bit 2 | Flash read data bit 2 |
| 1   | Flash write data bit 1 | Flash read data bit 1 |
| 0   | Flash write data bit 0 | Flash read data bit 0 |

### **6.15 External Veto/Gate Delay register**

```
#define SIS3316_EXTERNAL_VETO_GATE_DELAY_REG 0x3C
```

This register defines the “External Veto/Gate Delay” (0 to 2044) and the “External Trigger function Deadtime”.

| Bit | Function                                               |
|-----|--------------------------------------------------------|
| 31  | Enable “Deadtime with FP-Bus Address Threshold Flag“   |
| 30  | Enable “Deadtime with internal Address Threshold Flag“ |
| 29  | External Trigger function Deadtime Bit 13              |
| 28  | External Trigger function Deadtime Bit 12              |
| ... |                                                        |
| ..  |                                                        |
| 17  | External Trigger function Deadtime Bit 1               |
| 16  | External Trigger function Deadtime Bit 0               |
| 15  | External Veto/Gate Delay Bit 15                        |
| 14  | External Veto/Gate Delay Bit 14                        |
| ..  | ..                                                     |
| ..  | ..                                                     |
| 2   | External Veto/Gate Delay Bit 2                         |
| 1   | External Veto/Gate Delay Bit 1                         |
| 0   | External Veto/Gate Delay Bit 0                         |

The power up default value is 0

Valid values External Veto/Gate Delay : 0 to 2044.

| Value of written<br>External Veto/Gate Delay bits 15:0 | External Veto/Gate Delay |
|--------------------------------------------------------|--------------------------|
| 0                                                      | 0                        |
| 1                                                      | 1                        |
| ..                                                     | ..                       |
| 0x7FB                                                  | 2043                     |
| 0x7FC                                                  | 2044                     |
| 0x7FD                                                  | 2044                     |
| ..                                                     | ..                       |
| 0xFFFFE                                                | 2044                     |
| 0xFFFFF                                                | 2044                     |

“External Veto/Gate Delay” is new with VME FPGA firmware version V3316-200B.

“External Trigger function Deadtime” is new with VME FPGA firmware  
version V3316-2010.

## 6.16 Programmable Clock I2C registers

These read/write registers are used to set the Clock Frequencies via I2C.

| Bit | Write Function      | Read Function               |
|-----|---------------------|-----------------------------|
| 31  | None                | BUSY                        |
| 30  | None                | 0                           |
| ..  | ..                  | ..                          |
| ..  | ..                  | ..                          |
| 13  | READ BYTE, PUT ACK  | ..                          |
| 12  | WRITE BYTE, GET ACK | ..                          |
| 11  | STOP                | ..                          |
| 10  | REPEAT START        | ..                          |
| 9   | START               | ..                          |
| 8   | Ack on read cycle   | Received Ack on write cycle |
| 7   | Write data bit 7    | Read data bit 7             |
| ... | ...                 | ...                         |
| 1   | Write data bit 1    | Read data bit 1             |
| 0   | Write data bit 0    | Read data bit 0             |

See sis3316\_class.h and sis3316\_class.cpp:

```
int sis3316_adc::set_frequency(int osc, unsigned char *values)
where
    osc = 0 : ADC Clock I2C register
    osc = 1 : MGT1 Clock I2C register (reserved)
    osc = 2 : MGT2 Clock I2C register (reserved)
    osc = 3 : DDR3 Clock I2C register (reserved)

*values = 0x20 0xC2 0xBC 0x33 0xE4 0xF2 : 250 MHz
*values = 0x21 0xC2 0xBC 0x33 0xE4 0xF2 : 125 MHz
```

### 6.16.1 Programmable ADC Clock I2C register

```
#define SIS3316_ADC_CLK_OSC_I2C_REG      0x40          /* r/w; D32 */
```

This read/write register is used to set the Sample Clock Frequency via I2C.

### 6.16.2 Programmable MGT1 Clock I2C register

```
#define SIS3316_MGT1_OSC_I2C_REG      0x44          /* r/w; D32 */
```

This read/write register is used to set the Link1 Clock Frequency via I2C (not used).

### 6.16.3 Programmable MGT2 Clock I2C register

```
#define SIS3316_MGT2_OSC_I2C_REG      0x48          /* r/w; D32 */
```

This read/write register is used to set the Link2 Clock Frequency via I2C (not used).

### 6.16.4 Programmable DDR3 Clock I2C register

```
#define SIS3316_DD3_OSC_I2C_REG      0x4C          /* r/w; D32 */
```

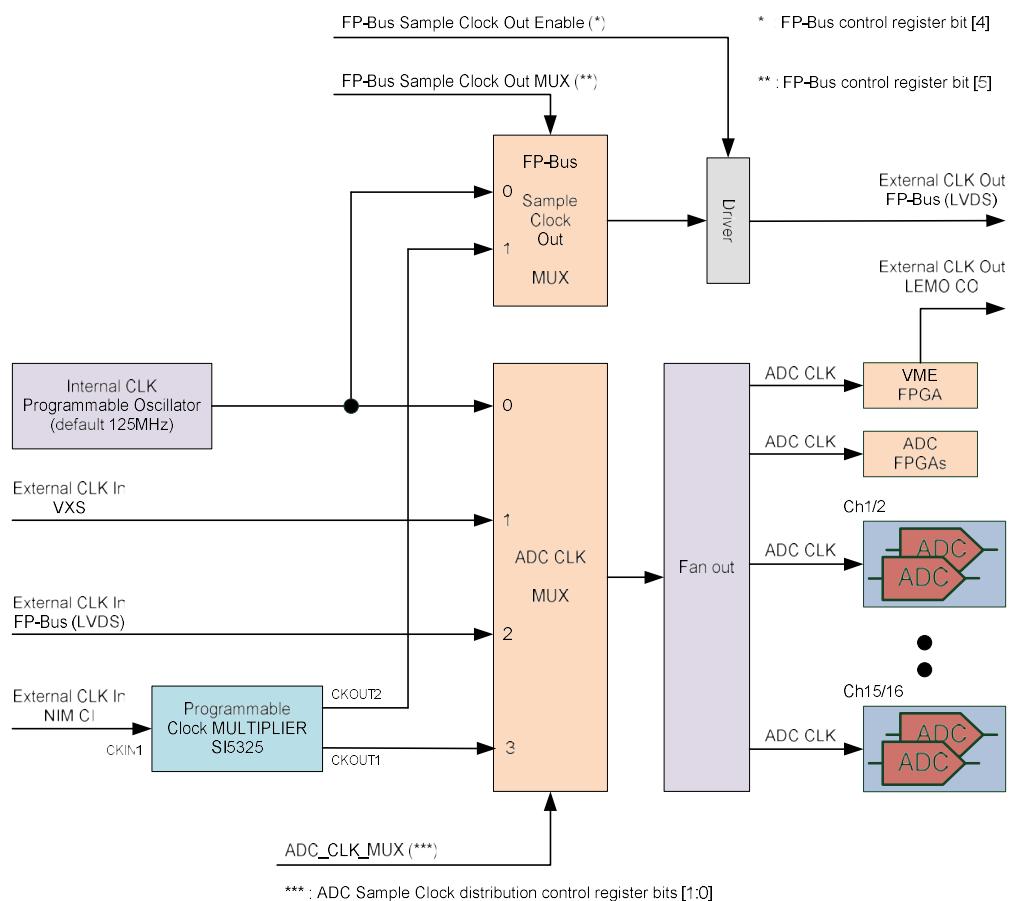
This read/write register is used to set the DDR3 Clock Frequency via I2C (not used).

**6.17 ADC Sample Clock distribution control register**

#define SIS3316\_SAMPLE\_CLOCK\_DISTRIBUTION\_CONTROL 0x50 /\* r/w; D32 \*/

The SIS3316

| BIT            | access | Name        | Function                                 |
|----------------|--------|-------------|------------------------------------------|
| 31-2<br>FFFFFC | R/W    | reserved    | no                                       |
| 1-0<br>0000003 | R/W    | ADC_CLK_MUX | ADC Sample Clock Multiplexer select bits |

**6.17.1 ADC Sample Clock Multiplexer select bits:**

| Sel1 | Sel0 | no | Clock source Description                                                           |
|------|------|----|------------------------------------------------------------------------------------|
| 0    | 0    | 0  | Onboard programmable Oscillator ( after Powerup: 125MHz)                           |
| 0    | 1    | 1  | VXS Clock from VXS Connector (Backplane)                                           |
| 1    | 0    | 2  | External Clock from FP-LVDS Bus Connector (Frontpanel)                             |
| 1    | 1    | 3  | External Clock from NIM Connector (Frontpanel) (via programmable Clock Multiplier) |

## 6.18 External NIM Clock Multiplier SPI register

```
#define SIS3316_NIM_CLK_MULTIPLIER_SPI_REG      0x54          /* r/w; D32 */
```

This read/write register is used to multiply the External NIM Clock (IN) via SPI.

Several parameters of the Clock Multiplier SI5325 chip can be configured with the SPI (serial Peripheral Interface).

Please refer to the documentation of the SI5325 chip for details.

| Bit | Write                   | read                     |
|-----|-------------------------|--------------------------|
| 31  | Cmd Bit 1               | Read/Write Cmd BUSY Flag |
| 30  | Cmd Bit 0               | Reset Cmd BUSY Flag      |
| 29  |                         |                          |
| ... |                         |                          |
| 17  |                         |                          |
| 16  |                         | Si53xx INT_C1B Status    |
| 15  | Instruction Byte Bit 7  |                          |
| ..  |                         |                          |
| ..  |                         |                          |
| 8   | Instruction Byte Bit 0  |                          |
| 7   | Address/Data Byte Bit 7 | Read Data Bit 7 (MSB)    |
| ..  |                         |                          |
| ..  | ..                      | Read Data Bit 1          |
| 0   | Address/Data Byte Bit 0 | Read Data Bit 0 (LSB)    |

The power up default value reads 0x0

| Cmd Bit 1 | Cmd Bit 0 | Command                    |
|-----------|-----------|----------------------------|
| 0         | 0         | Execute SPI Write/Read Cmd |
| 0         | 1         | Reset Cmd                  |

Reset Cmd: generates an 1us reset pulse

**6.19 FP-Bus control register**

#define SIS3316\_FP\_LVDS\_BUS\_CONTROL 0x58 /\* r/w; D32 \*/

| Bit | Name                               | Function                                                         |
|-----|------------------------------------|------------------------------------------------------------------|
| 31  | reserved                           |                                                                  |
| ... |                                    |                                                                  |
| 8   | reserved                           |                                                                  |
| 7   | reserved                           |                                                                  |
| 6   | reserved                           |                                                                  |
| 5   | FP-Bus Sample Clock Out MUX bit    | Selects the source of the FP-Bus Sample Clock                    |
| 4   | FP-Bus Sample Clock Out Enable     | Enables the Sample Clock to the FP- Bus<br>(only on one SIS3316) |
| 3   | reserved                           |                                                                  |
| 2   | reserved                           |                                                                  |
| 1   | FP-Bus Status Lines Output Enable  | Enables the Status Lines to the FP-Bus<br>(on all SIS3316)       |
| 0   | FP-Bus Control Lines Output Enable | Enables the Control Lines to the FP-Bus<br>(only on one SIS3316) |

The power up default value is 0x0

|                          |                                                                          |
|--------------------------|--------------------------------------------------------------------------|
| Sample Clock Out MUX bit | Clock source selection                                                   |
| 0                        | Onboard programmable oscillator ( after power up: 125MHz)                |
| 1                        | External clock from NIM connector CI (via programmable clock multiplier) |

## 6.20 NIM Input Control/Status register

```
#define SIS3316_NIM_INPUT_CONTROL_REG      0x5C          /* r/w; D32 */
```

| Bit | Write                                              | read                                                         |
|-----|----------------------------------------------------|--------------------------------------------------------------|
| 31  | reserved                                           | 0                                                            |
| 30  | reserved                                           | 0                                                            |
| 29  | reserved                                           | 0                                                            |
| 28  | reserved                                           | 0                                                            |
| 27  | reserved                                           | 0                                                            |
| 26  | reserved                                           | 0                                                            |
| 25  | reserved                                           | Status of NIM Input signal UI                                |
| 24  | reserved                                           | Status of External NIM Input UI                              |
| 23  | reserved                                           | 0                                                            |
| 22  | reserved                                           | 0                                                            |
| 21  | reserved                                           | Status of NIM Input signal TI                                |
| 20  | reserved                                           | Status of External NIM Input TI                              |
| 19  | reserved                                           | 0                                                            |
| 18  | reserved                                           | 0                                                            |
| 17  | reserved                                           | Status of NIM Input signal CI                                |
| 16  | reserved                                           | Status of External NIM Input CI                              |
| 15  | NIM Inputs TI and UI use for “User Counter” Enable | Status of NIM Inputs TI and UI use for “User Counter” Enable |
| 14  | “External Trigger function Deadtime logic“ Enable  | Status of “External Trigger function Deadtime logic“ Enable  |
| 13  | NIM Input UI as PPS Enable **                      | Status of NIM Input UI as PPS Enable                         |
| 12  | NIM Input UI as Veto Enable                        | Status of NIM Input UI as Veto Enable                        |
| 11  | Set NIM Input UI Function                          | Status of Set NIM Input UI Function                          |
| 10  | NIM Input UI Level sensitive                       | Status of NIM Input UI Level sensitive                       |
| 9   | NIM Input UI Invert                                | Status of NIM Input UI Invert                                |
| 8   | NIM Input UI as Timestamp Clear Enable             | Status of NIM Input UI as Timestamp Clear Enable             |
| 7   | Set NIM Input TI Function                          | Status of Set NIM Input TI Function                          |
| 6   | NIM Input TI Level sensitive                       | Status of NIM Input TI Level sensitive                       |
| 5   | NIM Input TI Invert                                | Status of NIM Input TI Invert                                |
| 4   | NIM Input TI as Trigger Enable                     | Status of NIM Input TI as Trigger Enable                     |
| 3   | Set NIM Input CI Function *                        | Status of Set NIM Input CI Function                          |
| 2   | NIM Input CI Level sensitive *                     | Status of NIM Input CI Level sensitive                       |
| 1   | NIM Input CI Invert *                              | Status of NIM Input CI Invert                                |
| 0   | NIM Input CI Enable *                              | Status of NIM Input CI Enable                                |

The power up default value is 0x0

\* Hardware version = 1 (Serial No 1-10):                           AC-coupled !  
 \* Hardware version > 1 (Serial No 11 and higher):               DC-coupled !

\*\* PPS : PulsePerSecond

The NIM Input “CI” was designed to accept an external sample clock but it is possible to use this input as a “control signal” also, provided the SIS3316 Hardware Version is V2 or higher. At the moment, no control function is assigned to it.

## 6.21 Acquisition control/status register (0x60, read/write)

```
#define SIS3316_ACQUISITION_CONTROL      0x60          /* read/write; D32 */
```

The acquisition control register is in charge of most of the settings related to the actual configuration of the digitization process.

| Bit | Write Function                                                       | Read                                                                           |
|-----|----------------------------------------------------------------------|--------------------------------------------------------------------------------|
| 31  |                                                                      | Status of Memory Address Threshold Flag Ch 13-16                               |
| 30  |                                                                      | Status of Sample Logic Busy Ch 13-16                                           |
| 29  |                                                                      | Status of Memory Address Threshold Flag Ch 9-12                                |
| 28  |                                                                      | Status of Sample Logic Busy Ch 9-12                                            |
| 27  |                                                                      | Status of Memory Address Threshold Flag Ch 5-8                                 |
| 26  |                                                                      | Status of Sample Logic Busy Ch 5-8                                             |
| 25  |                                                                      | Status of Memory Address Threshold Flag Ch 1-4                                 |
| 24  |                                                                      | Status of Sample Logic Busy Ch 1-4                                             |
| 23  |                                                                      | Status of “PPS latch bit”                                                      |
| 22  |                                                                      | Status of “Sample Bank Swap Control with NIM Input TI/UI” Logic enabled        |
| 21  |                                                                      | Status of FP-Bus-In Status 2: Address Threshold flag                           |
| 20  |                                                                      | Status of FP-Bus-In Status 1: Sample Logic busy                                |
| 19  |                                                                      | Status of Memory Address Threshold flag (OR)                                   |
| 18  |                                                                      | Status of Sample Logic Busy (OR)                                               |
| 17  |                                                                      | Status of ADC Sample Logic Armed On Bank2 flag                                 |
| 16  |                                                                      | Status of ADC Sample Logic Armed                                               |
| 15  | External Trigger Disable with internal Busy select                   | Status of External Trigger Disable with internal Busy select                   |
| 14  | Feedback Selected Internal Trigger as External Trigger Enable        | Status of Feedback Selected Internal Trigger as External Trigger Enable        |
| 13  | NIM Input UI as “disarm Bankx and arm alternate Bank” command Enable | Status of NIM Input UI as “disarm Bankx and arm alternate Bank” command Enable |
| 12  | NIM Input TI as “disarm Bankx and arm alternate Bank” command Enable | Status of NIM Input TI as “disarm Bankx and arm alternate Bank” command Enable |
| 11  | Local Veto function as Veto Enable                                   | Status of Local Veto function as Veto Enable                                   |
| 10  | External Timestamp-Clear function Enable                             | Status of External Timestamp-Clear function Enable                             |
| 9   | External Trigger function as Veto Enable                             | Status of External Trigger function as Veto Enable                             |
| 8   | External Trigger function as Trigger Enable                          | Status of External Trigger function as Trigger Enable                          |
| 7   | FP-Bus-In Sample Control Enable                                      | Status of FP-Bus-In Sample Control Enable                                      |
| 6   | FP-Bus-In Control 2 Enable                                           | Status of FP-Bus-In Control 2 Enable                                           |
| 5   | FP-Bus-In Control 1 as Veto Enable                                   | Status of FP-Bus-In Control 1 as Veto Enable                                   |
| 4   | FP-Bus-In Control 1 as Trigger Enable                                | Status of FP-Bus-In Control 1 as Trigger Enable                                |
| 3   | reserved                                                             |                                                                                |
| 2   | reserved                                                             |                                                                                |
| 1   | reserved                                                             |                                                                                |
| 0   | Set Single Bank Mode (reserved)                                      | Status of Single Bank Mode (reserved)                                          |

The power up default value:0x0

**Single Bank Mode bit:**

- 0 : Double Bank Mode is enable  
1 : Single Bank Mode is enable (reserved, not implemented yet)

**NIM Input UI as “disarm Bankx and arm alternate Bank” command Enable bit:**

This bit has only a function if the “Sample Bank Swap Control with NIM Input TI/UI” Logic is enabled.

- 0 : disables the toggling of the active Sample Bank with a signal on NIM Input UI  
1 : enables the toggling of the active Sample Bank with a signal on NIM Input UI

**NIM Input TI as “disarm Bankx and arm alternate Bank” command Enable bit:**

This bit has only a function if the “Sample Bank Swap Control with NIM Input TI/UI” Logic is enabled.

- 0 : disables the toggling of the active Sample Bank with a signal on NIM Input TI  
1 : enables the toggling of the active Sample Bank with a signal on NIM Input TI

**Status of “Sample Bank Swap Control with NIM Input TI/UI” Logic enabled:**

- 0 : “Sample Bank Swap Control with NIM Input TI/UI” Logic is not enabled  
1 : “Sample Bank Swap Control with NIM Input TI/UI” Logic is enabled

**6.22 Trigger Coincidence Lookup Table Control register**

```
#define SIS3316_LOOKUP_TABLE_CONTROL_REG 0x64      /* read/write; D32 */
```

| Bits     | 31                                                               | 30-16    | 15-8                                                 | 7-0                                                  |
|----------|------------------------------------------------------------------|----------|------------------------------------------------------|------------------------------------------------------|
| Function | Write: Lookup Table<br>Clear command<br>Read : Status Clear Busy | reserved | Lookup Table 2<br>Coincidence<br>output pulse length | Lookup Table 1<br>Coincidence<br>output pulse length |

default after reset: 0x0

This read/write register holds the pulse length values of the Lookup Table 1 and 2 Coincidence output pulses.

The (stretched) output pulse length is (pulse length value + 1) \* sample periode.

A write to this register with bit 31 = 1 clears the all 65536 entries in both tables.

The clearing process takes 65536 \* 8ns = 525us.

The “Status Clear Busy” bit is 1 during this process.

**6.23 Trigger Coincidence Lookup Table Address register**

```
#define SIS3316_LOOKUP_TABLE_ADDR_REG    0x68      /* read/write; D32 */
```

This read/write register holds the Write/Read Address of Lookup Tables 1 and 2 and a Channel Trigger Mask.

| Bits     | 31-16                                        | 15-0                                    |
|----------|----------------------------------------------|-----------------------------------------|
| Function | Lookup Table 1 and 2<br>Channel Trigger Mask | Lookup Table 1 and 2 Write/Read Address |

default after reset: 0x0

The 16-bit write/read address corresponds to the 16 channel trigger lines.

| Channel trigger line |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Table address /<br>Channel Trigger Mask |
|----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------------------------------|
| 16                   | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  |                                         |
| 0                    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0x0000                                  |
| 0                    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0x0001                                  |
| ..                   | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |                                         |
| 0                    | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0x0100                                  |
| ..                   | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |                                         |
| 1                    | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0x83FF                                  |
| ..                   | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |                                         |
| 1                    | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0xFFFF                                  |

The 16-bit channel mask enables the channel trigger line which will be used for the validation.

**6.24 Trigger Coincidence Lookup Table Data register**

```
#define SIS3316_LOOKUP_TABLE_DATA_REG      0x6C      /* read/write; D32 */
```

|          |                                                                   |          |                                                 |                                                 |
|----------|-------------------------------------------------------------------|----------|-------------------------------------------------|-------------------------------------------------|
| Bits     | 31-16                                                             | 15-2     | 1                                               | 0                                               |
| Function | Lookup Table 1 and 2<br>use falling edge of<br>Channel Trigger In | reserved | Lookup Table 2<br>Coincidence<br>validation bit | Lookup Table 1<br>Coincidence<br>validation bit |

default after reset: 0x0

A write to this register writes the Coincidence validation bits to the Lookup table at the address which is programmed in the Trigger Coincidence Lookup Table Address register. After a write to this register the contents of the Trigger Coincidence Lookup Table Address register is incremented automatically.

A read from this register holds the Coincidence validation bits of the Lookup at the address of the contents of the Trigger Coincidence Lookup Table Address register.

**6.25 LEMO Out “CO” Select register**

```
#define SIS3316_LEMO_OUT_CO_SELECT_REG 0x70 /* read/write; D32 */
```

If “Select Sample Clock” is set, then the Sample Clock is assigned to the LEMO Clock Out (“CO”) connector.

If “Select Sample Clock” is not set, then the other selected conditions are ored to the LEMO Clock Out (“CO”) connector.

| Bit | Write Function                                                       |
|-----|----------------------------------------------------------------------|
| 31  | reserved                                                             |
| 30  | Set                                                                  |
| 29  | reserved                                                             |
| 28  | reserved                                                             |
| 27  | reserved                                                             |
| 26  | reserved                                                             |
| 25  | reserved                                                             |
| 24  | reserved                                                             |
| 23  | reserved                                                             |
| 22  | Select Sample Logic Bank2 flag                                       |
| 21  | Select Sample Logic Bankx Armed                                      |
| 20  | Select “Sample Bank Swap Control with NIM Input TI/UI” Logic enabled |
| 19  | Select internal High Energy Trigger stretched pulse ch13-16          |
| 18  | Select internal High Energy Trigger stretched pulse ch9-12           |
| 17  | Select internal High Energy Trigger stretched pulse ch5-8            |
| 16  | Select internal High Energy Trigger stretched pulse ch1-4            |
| 15  | reserved                                                             |
| 14  | reserved                                                             |
| 13  | reserved                                                             |
| 12  | reserved                                                             |
| 11  | reserved                                                             |
| 10  | reserved                                                             |
| 9   | reserved                                                             |
| 8   | reserved                                                             |
| 7   | reserved                                                             |
| 6   | reserved                                                             |
| 5   | reserved                                                             |
| 4   | reserved                                                             |
| 3   | reserved                                                             |
| 2   | reserved                                                             |
| 1   | reserved                                                             |
| 0   | Select Sample Clock                                                  |

**6.26 LEMO Out “TO” Select register**

```
#define SIS3316_LEMO_OUT_TO_SELECT_REG    0x74      /* read/write; D32 */
```

The selected conditions are ored to the LEMO Trigger Out connector.

| Bit | Write Function                                                                   |
|-----|----------------------------------------------------------------------------------|
| 31  | Select and Generate Pulse                                                        |
| 30  | Set                                                                              |
| 29  | reserved                                                                         |
| 28  | Select “External Timestamp Clear“ to ADC FPGA **                                 |
| 27  | Select “External Veto/Gate“ to ADC FPGA **                                       |
| 26  | Select “External Trigger“ to ADC FPGA **                                         |
| 25  | Select “External Trigger“ to ADC FPGA (stretched output pulse, 16 sample clocks) |
| 24  | Select Lookup Table 1 Coincidence stretched output pulse                         |
| 23  | reserved                                                                         |
| 22  | Select Sample Logic Bank2 flag                                                   |
| 21  | Select Sample Logic Bankx Armed                                                  |
| 20  | Select “Sample Bank Swap Control with NIM Input TI/UI” Logic enabled             |
| 19  | Select internal SUM-Trigger stretched pulse ch13-16 *                            |
| 18  | Select internal SUM-Trigger stretched pulse ch9-12 *                             |
| 17  | Select internal SUM-Trigger stretched pulse ch5-8 *                              |
| 16  | Select internal SUM-Trigger stretched pulse ch1-4 *                              |
| 15  | Select internal Trigger stretched pulse ch16                                     |
| 14  | Select internal Trigger stretched pulse ch15                                     |
| 13  | Select internal Trigger stretched pulse ch14                                     |
| 12  | Select internal Trigger stretched pulse ch13                                     |
| 11  | Select internal Trigger stretched pulse ch12                                     |
| 10  | Select internal Trigger stretched pulse ch11                                     |
| 9   | Select internal Trigger stretched pulse ch10                                     |
| 8   | Select internal Trigger stretched pulse ch9                                      |
| 7   | Select internal Trigger stretched pulse ch8                                      |
| 6   | Select internal Trigger stretched pulse ch7                                      |
| 5   | Select internal Trigger stretched pulse ch6                                      |
| 4   | Select internal Trigger stretched pulse ch5                                      |
| 3   | Select internal Trigger stretched pulse ch4                                      |
| 2   | Select internal Trigger stretched pulse ch3                                      |
| 1   | Select internal Trigger stretched pulse ch2                                      |
| 0   | Select internal Trigger stretched pulse ch1                                      |

\* Note: Changed with versions V0250-000B, V0125-000B and higher.

The “internal SUM-Trigger stretched pulse” signals will be routed to the VME FPGA only if the SUM-FIR-Trigger logic is enabled. If the SUM-FIR-Trigger logic is not enabled (disabled) then the “Internal Gate 1” will be routed to the VME FPGA and could be selected as Lemo Output.

\*\* Note: New with VME FPGA Version V3316-0010

**6.27 LEMO Out “UO” Select register**

```
#define SIS3316_LEMO_OUT_UO_SELECT_REG 0x78 /* read/write; D32 */
```

The selected conditions are ored to the LEMO User Out (“UO”) connector.

| Bit | Write Function                                                       |
|-----|----------------------------------------------------------------------|
| 31  | Select and Generate Pulse                                            |
| 30  | Set                                                                  |
| 29  | reserved                                                             |
| 28  | Select “External Timestamp Clear“ to ADC FPGA **                     |
| 27  | Select “External Veto/Gate“ to ADC FPGA **                           |
| 26  | Select “External Trigger“ to ADC FPGA **                             |
| 25  | Prescaler Output Pulse                                               |
| 24  | Select Lookup Table 2 Coincidence stretched output pulse             |
| 23  | reserved                                                             |
| 22  | Select Sample Logic Bank2 flag                                       |
| 21  | Select Sample Logic Bankx Armed                                      |
| 20  | Select “Sample Bank Swap Control with NIM Input TI/UI” Logic enabled |
| 19  | Select Or of internal HE-Trigger stretched pulses ch13-16            |
| 18  | Select Or of internal HE-Trigger stretched pulse ch9-12              |
| 17  | Select Or of internal HE -Trigger stretched pulse ch5-8              |
| 16  | Select Or of internal HE-Trigger stretched pulse ch1-4               |
| 15  | reserved                                                             |
| 14  | reserved                                                             |
| 13  | reserved                                                             |
| 12  | reserved                                                             |
| 11  | reserved                                                             |
| 10  | reserved                                                             |
| 9   | Select Sample Logic Not Ready (Veto)                                 |
| 8   | Select Sample Logic Ready (Gate)                                     |
| 7   | Select External Timestamp Clear                                      |
| 6   | reserved                                                             |
| 5   | reserved                                                             |
| 4   | Select Sample Event Active Flag                                      |
| 3   | Select Address Threshold Flag                                        |
| 2   | Select Sample Logic Busy (Ready)                                     |
| 1   | Select Sample Logic Bankx Armed (Double Bank Mode)                   |
| 0   | Select Sample Logic Armed (Single Bank Mode; not implemented yet)    |

Sample Logic Not Ready (Veto): not ready to accept a trigger  
valid if Sample Logic Busy = 0 or Sample Event Active = 1

Sample Logic Ready (Gate): ready to accept a trigger  
valid if Sample Logic Busy = 1 and Sample Event Active = 0

\*\* Note: New with VME FPGA Version V3316-0010

**6.28 Internal Trigger Feedback Select register**

```
#define SIS3316_INTERNAL_TRIGGER_FEEDBACK_SELECT_REG 0x7C /* r/w D32 */
```

The selected conditions are ored to the Internal Trigger Feedback signal which can be used as External Trigger.

| Bit | Write Function                                           |
|-----|----------------------------------------------------------|
| 31  | reserved                                                 |
| 30  | reserved                                                 |
| 29  | reserved                                                 |
| 28  | reserved                                                 |
| 27  | reserved                                                 |
| 26  | reserved                                                 |
| 25  | reserved                                                 |
| 24  | Select Lookup Table 1 Coincidence stretched output pulse |
| 23  | reserved                                                 |
| 22  | reserved                                                 |
| 21  | reserved                                                 |
| 20  | reserved                                                 |
| 19  | Select internal SUM-Trigger stretched pulse ch13-16 *    |
| 18  | Select internal SUM-Trigger stretched pulse ch9-12 *     |
| 17  | Select internal SUM-Trigger stretched pulse ch5-8 *      |
| 16  | Select internal SUM-Trigger stretched pulse ch1-4 *      |
| 15  | Select internal Trigger stretched pulse ch16             |
| 14  | Select internal Trigger stretched pulse ch15             |
| 13  | Select internal Trigger stretched pulse ch14             |
| 12  | Select internal Trigger stretched pulse ch13             |
| 11  | Select internal Trigger stretched pulse ch12             |
| 10  | Select internal Trigger stretched pulse ch11             |
| 9   | Select internal Trigger stretched pulse ch10             |
| 8   | Select internal Trigger stretched pulse ch9              |
| 7   | Select internal Trigger stretched pulse ch8              |
| 6   | Select internal Trigger stretched pulse ch7              |
| 5   | Select internal Trigger stretched pulse ch6              |
| 4   | Select internal Trigger stretched pulse ch5              |
| 3   | Select internal Trigger stretched pulse ch4              |
| 2   | Select internal Trigger stretched pulse ch3              |
| 1   | Select internal Trigger stretched pulse ch2              |
| 0   | Select internal Trigger stretched pulse ch1              |

\* Note: Changed with versions V0250-000B, V0125-000B and higher.

The “internal SUM-Trigger stretched pulse” signals will be routed to the VME FPGA only if the SUM-FIR-Trigger logic is enabled. If the SUM-FIR-Trigger logic is not enabled (disabled) then the “Internal Gate 1” will be routed to the VME FPGA and could be selected as “External Trigger”.

## 6.29 ADC FPGA Data Transfer Control registers

```
#define SIS3316_DATA_TRANSFER_ADC1_4_CTRL_REG      0x80 /* r/w; D32 */
#define SIS3316_DATA_TRANSFER_ADC5_8_CTRL_REG      0x84 /* r/w; D32 */
#define SIS3316_DATA_TRANSFER_ADC9_12_CTRL_REG     0x88 /* r/w; D32 */
#define SIS3316_DATA_TRANSFER_ADC13_16_CTRL_REG    0x8C /* r/w; D32 */
```

With a write to this (these) register(s) the fast data transfer logic (2.5 GBit serial interface, between the ADC FPGA(s) and VME FPGA) will execute the written “Command”.

A “Reset Transfer FSM” command stops the fast data transfer logic and resets the FIFOs (ADC FPGA 1: ch1-ch4 Memory Data FIFO, ADC FPGA 2: ch5-ch8 Memory Data FIFO, ADC FPGA 3: ch9-ch12 Memory Data FIFO, ADC FPGA 4: ch12-ch16 Memory Data FIFO).

A “Start Read Transfer” command resets the FIFOs (ADC1 ch1-ch4 Memory Data FIFO, ADC2 ch5-ch8 Memory Data FIFO) and starts the fast data transfer “Read”. The logic transfers the memory data from the written Start Address to the VME FPGA data FIFO controlled by the FIFO Halffull flag.

A “Start Write Transfer” command resets the FIFOs (ADC1 ch1-ch4 Memory Data FIFO, ADC2 ch5-ch8 Memory Data FIFO) and starts the fast data transfer “Write”. The logic transfers the VME FPGA Data FIFO data to memory at the written Start Address controlled by the FIFO Empty flag.

**Note:** after a “Start Read Transfer” command, it takes up to 2 us to have data in the VME FPGA data FIFO.

| Bit      | 31-30 | 29-28             | 27-0                                                                             |
|----------|-------|-------------------|----------------------------------------------------------------------------------|
| Function | Cmd   | Space select bits | Memory 32-bit start address<br>(128 Meg x 16 : only address bits 25-0 are used ) |

default after Reset: 0x0

Command bit table:

| Command Bit1 | Command Bit0 | function             |
|--------------|--------------|----------------------|
| 0            | -            | Reset Transfer FSM   |
| 1            | 0            | Start Read Transfer  |
| 1            | 1            | Start Write Transfer |

Space select bit table:

| Space select Bit1 | Space select Bit0 | function                             |
|-------------------|-------------------|--------------------------------------|
| 0                 | 0                 | Memory 1 (Ch1 and Ch2)               |
| 0                 | 1                 | Memory 2 (Ch3 and Ch4)               |
| 1                 | 0                 | reserved                             |
| 1                 | 1                 | Statistic Counter (128 32-bit words) |

### 6.30 ADC FPGA Data Transfer Status registers

```
#define SIS3316_DATA_TRANSFER_ADC1_4_STATUS_REG      0x90 /* read; D32 */
#define SIS3316_DATA_TRANSFER_ADC5_8_STATUS_REG        0x94 /* read; D32 */
#define SIS3316_DATA_TRANSFER_ADC9_12_STATUS_REG       0x98 /* read; D32 */
#define SIS3316_DATA_TRANSFER_ADC13_16_STATUS_REG      0x9C /* read; D32 */
```

This set of four registers holds the status of the VME-ADC FPGA data transfer.

| Bit | Function                                                                           |
|-----|------------------------------------------------------------------------------------|
| 31  | Data Transfer Logic busy                                                           |
| 30  | Data Transfer Direction (Write-Flag; 0: Memory -> VME FPGA; 1: VME FPGA -> Memory) |
| 29  | 0                                                                                  |
| 28  | FIFO (read VME FIFO) Data AlmostFull Flag                                          |
| 27  | “max_no_pending_read_requests”                                                     |
| 26  | “no_pending_read_requests”                                                         |
| 25  | Data Transfer internal 32-bit Address counter bit 25                               |
| 24  | Data Transfer internal 32-bit Address counter bit 24                               |
| 23  | Data Transfer internal 32-bit Address counter bit 23                               |
|     | ..                                                                                 |
|     | ..                                                                                 |
| 0   | Data Transfer internal 32-bit Address counter bit 0                                |

## 6.1 VME FPGA – ADC FPGA Data Link Status register

```
#define SIS3316_VME_FPGA_LINK_ADC_PROT_STATUS 0xA0 /* r/w; D32 */
```

This register holds the VME FPGA Status of the data links between the VME FPGA and the ADC FPGAs .

| Bit | write                               | read                          |
|-----|-------------------------------------|-------------------------------|
| 31  | ADC FPGA 4: Clear Frame_error_latch | ADC FPGA 4: Frame_error_latch |
| 30  | ADC FPGA 4: Clear Soft_error_latch  | ADC FPGA 4: Soft_error_latch  |
| 29  | ADC FPGA 4: Clear Hard_error_latch  | ADC FPGA 4: Hard_error_latch  |
| 28  | no                                  | ADC FPGA 4: Lane_up_flag      |
| 27  | no                                  | ADC FPGA 4: Channel_up_flag   |
| 26  | no                                  | ADC FPGA 4: Frame_error_flag  |
| 25  | no                                  | ADC FPGA 4: Soft_error_flag   |
| 24  | no                                  | ADC FPGA 4: Hard_error_flag   |
| 23  | ADC FPGA 3: Clear Frame_error_latch | ADC FPGA 3: Frame_error_latch |
| 22  | ADC FPGA 3: Clear Soft_error_latch  | ADC FPGA 3: Soft_error_latch  |
| 21  | ADC FPGA 3: Clear Hard_error_latch  | ADC FPGA 3: Hard_error_latch  |
| 20  | no                                  | ADC FPGA 3: Lane_up_flag      |
| 19  | no                                  | ADC FPGA 3: Channel_up_flag   |
| 18  | no                                  | ADC FPGA 3: Frame_error_flag  |
| 17  | no                                  | ADC FPGA 3: Soft_error_flag   |
| 16  | no                                  | ADC FPGA 3: Hard_error_flag   |
| 15  | ADC FPGA 2: Clear Frame_error_latch | ADC FPGA 2: Frame_error_latch |
| 14  | ADC FPGA 2: Clear Soft_error_latch  | ADC FPGA 2: Soft_error_latch  |
| 13  | ADC FPGA 2: Clear Hard_error_latch  | ADC FPGA 2: Hard_error_latch  |
| 12  | no                                  | ADC FPGA 2: Lane_up_flag      |
| 11  | no                                  | ADC FPGA 2: Channel_up_flag   |
| 10  | no                                  | ADC FPGA 2: Frame_error_flag  |
| 9   | no                                  | ADC FPGA 2: Soft_error_flag   |
| 8   | no                                  | ADC FPGA 2: Hard_error_flag   |
| 7   | ADC FPGA 1: Clear Frame_error_latch | ADC FPGA 1: Frame_error_latch |
| 6   | ADC FPGA 1: Clear Soft_error_latch  | ADC FPGA 1: Soft_error_latch  |
| 5   | ADC FPGA 1: Clear Hard_error_latch  | ADC FPGA 1: Hard_error_latch  |
| 4   | no                                  | ADC FPGA 1: Lane_up_flag      |
| 3   | no                                  | ADC FPGA 1: Channel_up_flag   |
| 2   | no                                  | ADC FPGA 1: Frame_error_flag  |
| 1   | no                                  | ADC FPGA 1: Soft_error_flag   |
| 0   | no                                  | ADC FPGA 1: Hard_error_flag   |

## 6.2 ADC FPGA SPI BUSY Status register

```
#define SIS3316_ADC_FPGA_SPI_BUSY_STATUS_REG 0xA4 /* r; D32 */
```

This register holds the Status Flags of the ADC-SPI-Logic (ored) of the ADC-FPGAs.

| Bit | read                                                      |
|-----|-----------------------------------------------------------|
| 31  | ADC FPGAx: Busy flag (or of ADC FPGA 1 to 4 Busy flags) * |
| 30  | 0                                                         |
| 29  | 0                                                         |
| 28  | 0                                                         |
| 27  | 0                                                         |
| 26  | 0                                                         |
| 25  | 0                                                         |
| 24  | 0                                                         |
| 23  | 0                                                         |
| 22  | 0                                                         |
| 21  | 0                                                         |
| 20  | 0                                                         |
| 19  | 0                                                         |
| 18  | 0                                                         |
| 17  | 0                                                         |
| 16  | 0                                                         |
| 15  | 0                                                         |
| 14  | 0                                                         |
| 13  | 0                                                         |
| 12  | 0                                                         |
| 11  | 0                                                         |
| 10  | 0                                                         |
| 9   | 0                                                         |
| 8   | 0                                                         |
| 7   | 0                                                         |
| 6   | 0                                                         |
| 5   | 0                                                         |
| 4   | 0                                                         |
| 3   | ADC FPGA 4: Busy flag                                     |
| 2   | ADC FPGA 3: Busy flag                                     |
| 1   | ADC FPGA 2: Busy flag                                     |
| 0   | ADC FPGA 1: Busy flag                                     |

\* New with VME FPGA Version V\_3316\_2006

### 6.3 Prescaler Output Pulse Divider register

```
#define SIS3316_PRESCALER_OUTPUT_PULSE_DIVIDER_REG 0xB8
```

The “prescaler output pulse divider” register allows you to prescale the sampling clock. In combination with the “prescaler output pulse length” and “LEMO Out UO Select” registers it is possible to generate a pulse at LEMO output UO. If the register value is unequal 0 (logic is not disabled) the prescaler logic will start with a Timestamp clear command (for example with a write to the Key address `SIS3316_KEY_TIMESTAMP_CLR` and bit 10 of Acquisition register is set).

| Value of prescaler output pulse divider register | Divider value                  |
|--------------------------------------------------|--------------------------------|
| 0                                                | Logic is disabled              |
| 1                                                | 2                              |
| 2                                                | 3                              |
| ..                                               | ..                             |
| 0x FFFF FFFF                                     | 0x 1 0000 0000 (4.294.967.296) |

### 6.4 Prescaler Output Pulse Length register

```
#define SIS3316_PRESCALER_OUTPUT_PULSE_LENGTH_REG 0xBC
```

This register defines the Output pulse length in sample clocks.

| Value of prescaler output pulse length register | Pulse length in sample clocks  |
|-------------------------------------------------|--------------------------------|
| 0                                               | 1                              |
| 1                                               | 2                              |
| 2                                               | 3                              |
| ..                                              | ..                             |
| 0x FFFF FFFF                                    | 0x 1 0000 0000 (4.294.967.296) |

Examples with sampling clock of 250MHz:

| register value of prescaler output pulse divider | register value of prescaler output pulse length | Pulse frequency and length      |
|--------------------------------------------------|-------------------------------------------------|---------------------------------|
| 24 (0x18)                                        | 12 (0xC)                                        | 10 MHz (100 ns), 52ns (52/48%)  |
| 249 (0xF9)                                       | 24 (0x18)                                       | 1 MHz (1 us), 100 ns (90/10%)   |
| 24999 (0x61A7)                                   | 12499 (0x30D3)                                  | 10 KHz (100 us), 50 µs (50/50%) |
| 2499999 (0x26259F)                               | 24999 (0x61A7)                                  | 100 Hz (10 ms), 100 µs          |

## 6.5 Channel 1 to 16 Internal Trigger Counters

```
#define SIS3316_ADC1_INTERNAL_TRIGGER_COUNTER      0xC0 /* read; D32 */  
#define SIS3316_ADC2_INTERNAL_TRIGGER_COUNTER      0xC4 /* read; D32 */  
:  
#define SIS3316_ADC16_INTERNAL_TRIGGER_COUNTER     0xFC /* read; D32 */
```

Sixteen Internal Trigger counters are implemented in the VME FPGA to count the internal (stretched) triggers of each channel individually. These counters will be cleared with a “Timestamp clear” command simultaneously. Each counter will stop to increment if it reached 0xFFFFFFFF.

## 6.6 ADC Input tap delay registers

```
#define SIS3316_ADC_CH1_4_INPUT_TAP_DELAY_REG      0x1000
#define SIS3316_ADC_CH5_8_INPUT_TAP_DELAY_REG        0x2000
#define SIS3316_ADC_CH9_12_INPUT_TAP_DELAY_REG       0x3000
#define SIS3316_ADC_CH13_16_INPUT_TAP_DELAY_REG      0x4000
```

The input tap delay registers are used to adjust the ADC - FPGA data strobe timing.  
 After a change in the Sample Clock a “Calibration” command and a “Tap delay write” command are required.

| Bit      | 31-13 | 12                                       | 11          | 10                          | 9              | 8              | 7-0                                                   |
|----------|-------|------------------------------------------|-------------|-----------------------------|----------------|----------------|-------------------------------------------------------|
| Function | None  | Add ½ Sample Clock periode delay bit (*) | Calibration | Clear Link Error Latch bits | Ch3-Ch4 Select | Ch1-Ch2 Select | Tap delay value (* 40ps, max. ½ Sample Clock periode) |

\* new with ADC Version V-0250-0004 and V-0125-0004

Clear Link Error Latch bits : see ADC FPGA Status register

**Note 1:**

A Calibration takes 20 ADC sample clock cycles.

**Note 2:**

Due to the double data rate nature of the ADC chip a coarse tuning of the tap delay is needed for proper ADC channel 1/channel 2 sequence and a fine tuning to meet the eye.

Typical sample clock tap delay combinations are listed in the tables below.

## SIS3316-250MHz-14bit:

| ADC FPGA<br>Firmware Version | V-0250-0004                   |           | <= V-0250-0003       |
|------------------------------|-------------------------------|-----------|----------------------|
| Sample Clock in MHz          | Add ½ Sample<br>Clock periode | Tap delay | Tap delay            |
| 250.000                      | 1                             | 0x02      | 0x48                 |
| 227.273                      | 1                             | 0x1F      | not possible (0x1F*) |
| 208.333                      | 1                             | 0x35      | not possible (0x35*) |
| 178.571                      | 0                             | 0x12      | 0x12                 |
|                              |                               |           |                      |
| 166.667                      | 0                             | 0x20      | 0x20                 |
| 138.889                      | 0                             | 0x35      | 0x35                 |
| 125.000                      | 0                             | 0x50      | 0x50                 |
| 119.048                      | 0                             | 0x60      | 0x60                 |
|                              |                               |           |                      |
| 113.636                      | 1                             | 0x10      | not possible (0x10*) |
| 104.167                      | 1                             | 0x20      | not possible (0x20*) |
| 100.000                      | 1                             | 0x20      | not possible (0x20*) |
| 83.333                       | 1                             | 0x30      | not possible (0x30*) |
|                              |                               |           |                      |
| 71.429                       | 1                             | 0x60      | not possible (0x60*) |
| 62.500                       | 1                             | 0x60      | not possible (0x60*) |
| 50.000                       | 0                             | 0x20      | 0x20                 |
| 25.000                       | 0                             | 0x20      | 0x20                 |

\* odd and even channels are swapped

## SIS3316-125MHz-16bit:

| ADC FPGA<br>Firmware Version | V-0125-0004                   |           | <= V-0125-0003       |
|------------------------------|-------------------------------|-----------|----------------------|
| Sample Clock in MHz          | Add ½ Sample<br>Clock periode | Tap delay | Tap delay            |
| 125.000                      | 1                             | 0x20      | 0x7F                 |
| 119.048                      | 1                             | 0x20      | 0x7F                 |
|                              |                               |           |                      |
| 113.636                      | 1                             | 0x20      | 0x7F                 |
| 104.167                      | 1                             | 0x30      | not possible (0x30*) |
| 100.000                      | 1                             | 0x30      | not possible (0x30*) |
| 83.333                       | 1                             | 0x40      | not possible (0x40*) |
|                              |                               |           |                      |
| 71.429                       | 1                             | 0x60      | 0x0                  |
| 62.500                       | 0                             | 0x20      | 0x20                 |
| 50.000                       | 0                             | 0x30      | 0x30                 |
| 25.000                       | 0                             | 0x30      | 0x30                 |

\* odd and even channels are swapped

## 6.7 ADC Gain and Termination Control register

```
#define SIS3316_ADC_CH1_4_ANALOG_CTRL_REG      0x1004
#define SIS3316_ADC_CH5_8_ANALOG_CTRL_REG        0x2004
#define SIS3316_ADC_CH9_12_ANALOG_CTRL_REG       0x3004
#define SIS3316_ADC_CH13_16_ANALOG_CTRL_REG      0x4004
```

| Bit | function                           |
|-----|------------------------------------|
| 31  | reserved                           |
| 30  | reserved                           |
| ..  |                                    |
| 26  | Ch 4: Disable 50 Ohm Termination * |
| 25  | Ch 4: Gain Control bit 1           |
| 24  | Ch 4: Gain Control bit 0           |
| 23  | reserved                           |
| 22  | reserved                           |
| ..  |                                    |
| 18  | Ch 3: Disable 50 Ohm Termination * |
| 17  | Ch 3: Gain Control bit 1           |
| 16  | Ch 3: Gain Control bit 0           |
| 15  | reserved                           |
| 14  | reserved                           |
| ..  |                                    |
| 10  | Ch 2: Disable 50 Ohm Termination * |
| 9   | Ch 2: Gain Control bit 1           |
| 8   | Ch 2: Gain Control bit 0           |
| 7   | reserved                           |
| 6   | reserved                           |
| ..  |                                    |
| 2   | Ch 1: Disable 50 Ohm Termination * |
| 1   | Ch 1: Gain Control bit 1           |
| 0   | Ch 1: Gain Control bit 0           |

The power up default value is 0x0

\* Disable 50 Ohm termination will set the termination to 1K

Gain Control :

| Gain Control Bit 1 | Gain Control Bit 0 | Input Range |
|--------------------|--------------------|-------------|
| 0                  | 0                  | 5 V         |
| 0                  | 1                  | 2 V         |
| 1                  | 0                  | 1.9 V       |
| 1                  | 1                  | 1.9 V       |

## 6.8 ADC Offset (DAC) Control registers

|                                                 |        |
|-------------------------------------------------|--------|
| #define SIS3316_ADC_CH1_4_DAC_OFFSET_CTRL_REG   | 0x1008 |
| #define SIS3316_ADC_CH5_8_DAC_OFFSET_CTRL_REG   | 0x2008 |
| #define SIS3316_ADC_CH9_12_DAC_OFFSET_CTRL_REG  | 0x3008 |
| #define SIS3316_ADC_CH13_16_DAC_OFFSET_CTRL_REG | 0x4008 |

These registers are used to set the ADC Offset-DACs (AD5666) of the 4 ADC channels. The DAC is selected via the DAC Selection Bit in the ADC Offset (DAC) Control registers.

At first the Internal Reference has to be enabled with write cycles to these registers:

```
vme_A32D32_write (module_base_addr + SIS3316_ADC_CH1_4_DAC_OFFSET_REG, 0x88f00001);
```

|    | Write Function      | DAC DB[27:0] Function |
|----|---------------------|-----------------------|
| 31 | DAC Ctrl Mode bit 2 | none                  |
| 30 | DAC Ctrl Mode bit 1 | none                  |
| 31 | DAC Ctrl Mode bit 0 | none                  |
| 28 | reserved            | none                  |
| 27 | DAC DB27            | DAC Command Bit 3     |
| 26 | DAC DB26            | DAC Command Bit 2     |
| 25 | DAC DB25            | DAC Command Bit 1     |
| 24 | DAC DB24            | DAC Command Bit 0     |
| 23 | DAC DB23            | DAC Address Bit 3     |
| 22 | DAC DB22            | DAC Address Bit 2     |
| 21 | DAC DB21            | DAC Address Bit 1     |
| 20 | DAC DB20            | DAC Address Bit 0     |
| 19 | DAC DB18            | DAC Data Bit 15       |
| 18 | DAC DB18            | DAC Data Bit 14       |
| .. | ..                  | ..                    |
| 5  | DAC DB5             | DAC Data Bit 1        |
| 4  | DAC DB4             | DAC Data Bit 0        |
| 3  | DAC DB3             | x                     |
| 2  | DAC DB2             | x                     |
| 1  | DAC DB1             | x                     |
| 0  | DAC DB0             | x                     |

DAC Ctrl Mode bits

| Bits | Function                                                         |
|------|------------------------------------------------------------------|
| 0--  | No                                                               |
| 100  | Write Command/Address/Data to DAC                                |
| 101  | Write Command/Address/Data to DAC and generate “DAC LDAC” (load) |
| 110  | Generate “DAC LDAC” (load)                                       |
| 111  | Generate “DAC CLR” (clear)                                       |

### Note:

The logic needs approximately 23 usec to execute a command !

A “Logic-BUSY” flag is readable via the “ADC FPGA SPI Busy Status” register.

DAC Command bit table:

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description                                           |
|-------|-------|-------|-------|-------------------------------------------------------|
| 0     | 0     | 0     | 0     | Write to Input Register n                             |
| 0     | 0     | 0     | 1     | Update DAC Register n                                 |
| 0     | 0     | 1     | 0     | Write to Input Register n, update all (software LDAC) |
| 0     | 0     | 1     | 1     | Write to and update DAC Channel n                     |
| 0     | 1     | 0     | 0     | Power down/power up DAC                               |
| 0     | 1     | 0     | 1     | Load clear code register                              |
| 0     | 1     | 1     | 0     | Load LDAC register                                    |
| 0     | 1     | 1     | 1     | Reset (power-on reset)                                |
| 1     | 0     | 0     | 0     | Set up DCEN/REF register                              |
| 1     | 0     | 0     | 1     | No operation                                          |
| 1     | 0     | 1     | 0     | Reserved                                              |
| 1     | -     | -     | -     | Reserved                                              |
| 1     | 1     | 1     | 1     | Reserved                                              |

DAC Address bit table:

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Selected DAC Channel         |
|-------|-------|-------|-------|------------------------------|
| 0     | 0     | 0     | 0     | DAC A (Channel 1, 5, 9, 13)  |
| 0     | 0     | 0     | 1     | DAC B (Channel 2, 6, 10, 14) |
| 0     | 0     | 1     | 0     | DAC C (Channel 3, 7, 11, 15) |
| 0     | 0     | 1     | 1     | DAC D (Channel 4, 8, 12, 16) |
| 1     | 1     | 1     | 1     | All DAC's                    |

Example:

```
gl_vme_crate->vme_A32D32_write(module_base_addr + SIS3316_ADC_CH1_4_DAC_OFFSET_CTRL_REG,
    0x80000000 // DAC CTRL Mode: Write Command
    + 0x2000000 // DAC Command Mode: write to Input Register and update all
    + 0xf00000 // DAC Address bits: ALL DACs
    + ((dac_offset & 0xffff) << 4); // DAC offset
```

DAC offset setting table for 5 V input range (approximately)

| DAC offset value | Input Range      |
|------------------|------------------|
| 65535            | 0 V to 5 V       |
| 32768 (0x8000)   | -2.5 V to +2.5 V |
| 0                | -5 V to 0 V      |

DAC offset setting table for 2 V input range (approximately)

| DAC offset value | Input Range  |
|------------------|--------------|
| 52000            | 0 V to 2 V   |
| 32768 (0x8000)   | -1 V to +1 V |
| 13000            | -2 V to 0 V  |

## 6.9 ADC Offset (DAC) Readback registers

```
#define SIS3316_ADC_CH1_4_DAC_OFFSET_READBACK_REG      0x1108
#define SIS3316_ADC_CH5_8_DAC_OFFSET_READBACK_REG        0x2108
#define SIS3316_ADC_CH9_12_DAC_OFFSET_READBACK_REG       0x3108
#define SIS3316_ADC_CH13_16_DAC_OFFSET_READBACK_REG      0x4108
```

| Bit | Read Function        |
|-----|----------------------|
| 31  | 0                    |
| 30  | 0                    |
| ..  | .                    |
| 17  | 0                    |
| 16  | 0                    |
| 15  | DAC Read Data Bit 15 |
| 14  | DAC Read Data Bit 14 |
| ..  | ..                   |
| 1   | DAC Read Data Bit 1  |
| 0   | DAC Read Data Bit 0  |

## 6.10 ADC SPI Control register

```
#define SIS3316_ADC_CH1_4_SPI_CTRL_REG      0x100C
#define SIS3316_ADC_CH5_8_SPI_CTRL_REG       0x200C
#define SIS3316_ADC_CH9_12_SPI_CTRL_REG      0x300C
#define SIS3316_ADC_CH13_16_SPI_CTRL_REG     0x400C
```

Several parameters of the ADC AD9643 chip (like gain/offset, test modes ... e.g.) can be configured with the SPI (serial Peripheral Interface).

Please refer to the documentation of the ADC AD9643 chip for details.

A write to this register with Commands “Write Cmd” or “Read Cmd” initiates a SPI access to the ADC chip. Bit 22 and bits 20 to 0 are relevant for the SPI access.

| Bit | Write function          | read                    |
|-----|-------------------------|-------------------------|
| 31  | Command Bit 1           | Command Bit 1           |
| 30  | Command Bit 0           | Command Bit 0           |
| 29  | reserved                | reserved                |
| 28  | reserved                | reserved                |
| 27  | reserved                | reserved                |
| 26  | reserved                | reserved                |
| 25  | reserved                | reserved                |
| 24  | ADC Data Output Enable  | ADC Data Output Enable  |
| 23  | reserved                | reserved                |
| 22  | Select ADCx ch3/ch4 bit | Select ADCx ch3/ch4 bit |
| 21  | reserved                | reserved                |
| 20  | Address Bit 12          | Address Bit 12          |
| 19  | Address Bit 11          | Address Bit 11          |
| ..  | ..                      | ..                      |
| 10  | Address Bit 2           | Address Bit 2           |
| 9   | Address Bit 1           | Address Bit 1           |
| 8   | Address Bit 0           | Address Bit 0           |
| 7   | Write Data Bit 7 (MSB)  | Write Data Bit 7 (MSB)  |
| 6   | Write Data Bit 6        | Write Data Bit 6        |
| ..  | ..                      | ..                      |
| 1   | Write Data Bit 1        | Write Data Bit 1        |
| 0   | Write Data Bit 0 (LSB)  | Write Data Bit 0 (LSB)  |

The power up default value is 0x0

| Command | function                                  |
|---------|-------------------------------------------|
| 0       | no                                        |
| 1       | Reserved (ADC Synch Cmd)                  |
| 2       | Write Cmd (relevant bits: 22 and 20 to 0) |
| 3       | Read Cmd (relevant bits: 22 and 20 to 0)  |

**Note:**

Be aware, that the logic needs approximately 23 µs to execute a Write or Read command.  
A “Logic-BUSY” flag is readable via the “ADC FPGA SPI Busy Status” register.

Select ADCx ch3/ch4 bit = 0:      select ADC channel 1/2  
Select ADCx ch3/ch4 bit = 1:      select ADC channel 3/4

**ADC Data Output Enable:**

After power up, this bit is cleared and it disables the data output of the ADC chips.  
Once the ADC chip is configured for LVDS output, this bit has to be set to enable the data output.

**Note:**

With a “VME Key Register Reset” (write to offset address 0x400) or with a “VME Key Reset ADC FPGA-Logic” (write to offset address 0x434) Command the ADC Data Output will be disabled (ADC Data Output Enable bit = 0).

### **6.11 ADC SPI Readback registers**

|                                              |        |
|----------------------------------------------|--------|
| #define SIS3316_ADC_CH1_4_SPI_READBACK_REG   | 0x110C |
| #define SIS3316_ADC_CH5_8_SPI_READBACK_REG   | 0x210C |
| #define SIS3316_ADC_CH9_12_SPI_READBACK_REG  | 0x310C |
| #define SIS3316_ADC_CH13_16_SPI_READBACK_REG | 0x410C |

| Bit | Read Function   |
|-----|-----------------|
| 31  | 0               |
| 30  | 0               |
| ..  | .               |
| 9   | 0               |
| 8   | 0               |
| 15  | Read Data Bit 7 |
| 14  | Read Data Bit 6 |
| ..  | ..              |
| 1   | Read Data Bit 1 |
| 0   | Read Data Bit 0 |

## 6.12 Event configuration registers

```
#define SIS3316_ADC_CH1_4_EVENT_CONFIG_REG          0x1010
#define SIS3316_ADC_CH5_8_EVENT_CONFIG_REG           0x2010
#define SIS3316_ADC_CH9_12_EVENT_CONFIG_REG          0x3010
#define SIS3316_ADC_CH13_16_EVENT_CONFIG_REG         0x4010
```

This register is implemented for each four channel group.

| Bit | Function                            |
|-----|-------------------------------------|
| 31  | CH4 External Veto Enable bit        |
| 30  | CH4 External Gate Enable bit        |
| 29  | CH4 Internal Gate 2 Enable bit      |
| 28  | CH4 Internal Gate 1 Enable bit      |
| 27  | CH4 External Trigger Enable bit     |
| 26  | CH4 Internal Trigger Enable bit     |
| 25  | CH4 Internal SUM-Trigger Enable bit |
| 24  | CH4 Input Invert bit                |
| 23  | CH3 External Veto Enable bit        |
| 22  | CH3 External Gate Enable bit        |
| 21  | CH3 Internal Gate 2 Enable bit      |
| 20  | CH3 Internal Gate 1 Enable bit      |
| 19  | CH3 External Trigger Enable bit     |
| 18  | CH3 Internal Trigger Enable bit     |
| 17  | CH3 Internal SUM-Trigger Enable bit |
| 16  | CH3 Input Invert bit                |
| 15  | CH2 External Veto Enable bit        |
| 14  | CH2 External Gate Enable bit        |
| 13  | CH2 Internal Gate 2 Enable bit      |
| 12  | CH2 Internal Gate 1 Enable bit      |
| 11  | CH2 External Trigger Enable bit     |
| 10  | CH2 Internal Trigger Enable bit     |
| 9   | CH2 Internal SUM-Trigger Enable bit |
| 8   | CH2 Input Invert bit                |
| 7   | CH1 External Veto Enable bit        |
| 6   | CH1 External Gate Enable bit        |
| 5   | CH1 Internal Gate 2 Enable bit      |
| 4   | CH1 Internal Gate 1 Enable bit      |
| 3   | CH1 External Trigger Enable bit     |
| 2   | CH1 Internal Trigger Enable bit     |
| 1   | CH1 Internal SUM-Trigger Enable bit |
| 0   | CH1 Input Invert bit                |

CHx input invert bit = 0:    used for positive signals

CHx input invert bit = 1:    used for negative signals

ADCx Trigger enable bits (with CHx Internal Pileup Trigger Enable bit = 0)

| 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 |

**Note:** See Extended Event configuration register (Internal Pileup Trigger Enable bit)

### 6.13 Extended Event configuration registers

|                                                       |        |
|-------------------------------------------------------|--------|
| #define SIS3316_ADC_CH1_4_EXTENDED_EVENT_CONFIG_REG   | 0x109C |
| #define SIS3316_ADC_CH5_8_EXTENDED_EVENT_CONFIG_REG   | 0x209C |
| #define SIS3316_ADC_CH9_12_EXTENDED_EVENT_CONFIG_REG  | 0x309C |
| #define SIS3316_ADC_CH13_16_EXTENDED_EVENT_CONFIG_REG | 0x409C |

This register is implemented for each four channel group.

| Bit | Function                                                         |
|-----|------------------------------------------------------------------|
| 31  | reserved                                                         |
| 30  | reserved                                                         |
| 29  | Reserved                                                         |
| 28  | CH4 “Save Raw data of first Event of Bankbuffer only” Enable bit |
| 27  | reserved                                                         |
| 26  | reserved                                                         |
| 25  | reserved                                                         |
| 24  | CH4 Internal Pileup Trigger Enable bit                           |
| 23  | reserved                                                         |
| 22  | reserved                                                         |
| 21  | reserved                                                         |
| 20  | CH3 “Save Raw data of first Event of Bankbuffer only” Enable bit |
| 19  | reserved                                                         |
| 18  | reserved                                                         |
| 17  | reserved                                                         |
| 16  | CH3 Internal Pileup Trigger Enable bit                           |
| 15  | reserved                                                         |
| 14  | reserved                                                         |
| 13  | reserved                                                         |
| 12  | CH2 “Save Raw data of first Event of Bankbuffer only” Enable bit |
| 11  | reserved                                                         |
| 10  | reserved                                                         |
| 9   | reserved                                                         |
| 8   | CH2 Internal Pileup Trigger Enable bit                           |
| 7   | reserved                                                         |
| 6   | reserved                                                         |
| 5   | reserved                                                         |
| 4   | CH1 “Save Raw data of first Event of Bankbuffer only” Enable bit |
| 3   | reserved                                                         |
| 2   | reserved                                                         |
| 1   | reserved                                                         |
| 0   | CH1 Internal Pileup Trigger Enable bit                           |

An internal Pileup trigger will be generated with a second internal trigger while the “Pileup Window” is still active.

ADCx Trigger enable bits (with CHx Internal Trigger Enable bit = 0)

| external trigger enable | internal Pileup trigger enable | Function                                                   |
|-------------------------|--------------------------------|------------------------------------------------------------|
| 0                       | 0                              | No triggering                                              |
| 0                       | 1                              | internal channel pileup trigger (asynchronous mode)        |
| 1                       | 0                              | external trigger (synchronous mode)                        |
| 1                       | 1                              | Or of internal channel pileup trigger and external trigger |

## 6.14 Channel Header ID registers

|                                                |        |
|------------------------------------------------|--------|
| #define SIS3316_ADC_CH1_4_CHANNEL_HEADER_REG   | 0x1014 |
| #define SIS3316_ADC_CH5_8_CHANNEL_HEADER_REG   | 0x2014 |
| #define SIS3316_ADC_CH9_12_CHANNEL_HEADER_REG  | 0x3014 |
| #define SIS3316_ADC_CH13_16_CHANNEL_HEADER_REG | 0x4014 |

This register is implemented for each channel group. Its 12-bit content will be used in the event header. The four registers can be preset with a unique value to obtain channel and module (in a multi module setup) identification.

| Bit | Function                  |
|-----|---------------------------|
| 31  | Channel Header/ID bit 11  |
| ... | ...                       |
| 24  | Channel Header/ID bit 4   |
| 23  | Channel Header/ID bit 3** |
| 22  | Channel Header/ID bit 2** |
| 21  | Channel Header/ID bit 1*  |
| 20  | Channel Header/ID bit 0*  |
| 19  | reserved                  |
|     |                           |
| 1   | reserved                  |
| 0   | reserved                  |

Channel Header/ID bit 11:4 are writeable (for example with the VME Base address)

\*\*Channel Header/ID bit 3:2 has to be set with ADC FPGA group number -1

\*Channel Header/ID bits 1:0 (no write function, → set to channel number in FPGA )

## 6.15 End Address Threshold register

```
#define SIS3316_ADC_CH1_4_ADDRESS_THRESHOLD_REG      0x1018
#define SIS3316_ADC_CH5_8_ADDRESS_THRESHOLD_REG        0x2018
#define SIS3316_ADC_CH9_12_ADDRESS_THRESHOLD_REG       0x3018
#define SIS3316_ADC_CH13_16_ADDRESS_THRESHOLD_REG      0x4018
```

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

The value of the “Actual Sample address counter” (Bankx) will be compared with value of the “End Address Threshold” register.

If the value of the “Actual Sample address counter” goes above the value of the “End Address Threshold value” then a “Memory Address Threshold Flag Ch x” will set to indicate a “watermark”.

The value is given in 32-bit words !

| Bit |                                                                                         |
|-----|-----------------------------------------------------------------------------------------|
| 31  | “Suppress saving of more Hits/Events if Memory Address Threshold Flag is valid” Enable* |
| 30  | Reserved , read as 0                                                                    |
| ... | Reserved , read as 0                                                                    |
| 24  | Reserved , read as 0                                                                    |
| 23  | End Address Threshold value Bit 23                                                      |
| ..  |                                                                                         |
| 2   | End Address Threshold value Bit 2                                                       |
| 1   | End Address Threshold value Bit 1                                                       |
| 0   | End Address Threshold value Bit 0                                                       |

The power up default value is 0

\* New with ADC FPGA Version V\_0xxx\_0008

### “Suppress saving of more Hits/Events if Memory Address Threshold Flag is valid” Enable bit = 0:

If the value of the “Actual Sample address counter” goes above the value of the “End Address Threshold value” then the “Memory Address Threshold Flag Ch x” will set, only and more Hits/Events will be saved.

### “Suppress saving of more Hits/Events if Memory Address Threshold Flag is valid” Enable bit = 1:

If the value of the “Actual Sample address counter” goes above the value of the “End Address Threshold value” then the “Memory Address Threshold Flag Ch x” will set and no more Hits/Events will be saved.

## 6.16 Active Trigger Gate Window Length registers

```
#define SIS3316_ADC_CH1_4_TRIGGER_GATE_WINDOW_LENGTH_REG      0x101C
#define SIS3316_ADC_CH5_8_TRIGGER_GATE_WINDOW_LENGTH_REG        0x201C
#define SIS3316_ADC_CH9_12_TRIGGER_GATE_WINDOW_LENGTH_REG       0x301C
#define SIS3316_ADC_CH13_16_TRIGGER_GATE_WINDOW_LENGTH_REG      0x401C
```

These registers define the length of the Active Trigger Gate Window (2, 4, to 65536) .

| D31:16   | D15:0                                              |
|----------|----------------------------------------------------|
| reserved | Active Trigger Gate Window Length (bit 0 not used) |

The power up default value is 0

### Example:

Desired Active Trigger Gate Window of 1024 clocks -> set the register to 0x0000003FE

## 6.17 Raw Data Buffer Configuration registers

```
#define SIS3316_ADC_CH1_4_RAW_DATA_BUFFER_CONFIG_REG      0x1020
#define SIS3316_ADC_CH5_8_RAW_DATA_BUFFER_CONFIG_REG       0x2020
#define SIS3316_ADC_CH9_12_RAW_DATA_BUFFER_CONFIG_REG      0x3020
#define SIS3316_ADC_CH13_16_RAW_DATA_BUFFER_CONFIG_REG     0x4020
```

These registers define the start index of the raw data buffer and the number of samples which will be copy to the Memory.

| Bit | Function                         |
|-----|----------------------------------|
| 31  | Raw Buffer Sample Length Bit 15  |
| ... |                                  |
| ..  |                                  |
| 18  | Raw Buffer Sample Length Bit 2   |
| 17  | Raw Buffer Sample Length Bit 1   |
| 16  | Raw Buffer Sample Length Bit 0** |
| 15  | Raw Buffer Start Index Bit 15    |
| ... |                                  |
| ..  |                                  |
| 2   | Raw Buffer Start Index Bit 2     |
| 1   | Raw Buffer Start Index Bit 1     |
| 0   | Raw Buffer Start Index Bit 0*    |

The power up default value is 0

\* Raw Buffer\_Start\_Index bit 0 is always “0”.

\*\* Raw Buffer\_Sample\_Length bit 0 is always “0”.

Data are stored to memory in packets of 2 consecutive samples by the copy logic.

Note: The value of the “Raw Buffer Sample\_Length“ will be used if the value of the “**Extended** Raw Buffer Sample\_Length“ is zero !

See: **Extended** Raw Data Buffer Configuration registers

## 6.18 Pileup Configuration registers

```
#define SIS3316_ADC_CH1_4_PILEUP_CONFIG_REG          0x1024
#define SIS3316_ADC_CH5_8_PILEUP_CONFIG_REG           0x2024
#define SIS3316_ADC_CH9_12_PILEUP_CONFIG_REG          0x3024
#define SIS3316_ADC_CH13_16_PILEUP_CONFIG_REG         0x4024
```

These registers define the windows to recognize Pileups.

| Bit | Function                        |
|-----|---------------------------------|
| 31  | Re-Pileup Window Length 15      |
| ..  |                                 |
| 18  | Re-Pileup Window Length Bit 2   |
| 17  | Re-Pileup Window Length Bit 1   |
| 16  | Re-Pileup Window Length Bit 0 * |
| 15  | Pileup Window Length 15         |
| ..  |                                 |
| 2   | Pileup Window Length Bit 2      |
| 1   | Pileup Window Length Bit 1      |
| 0   | Pileup Window Length Bit 0 *    |

The power up default value is 0

\* Re-Pileup Window Length bit 0 is always “0”.

\* Pileup Window Length bit 0 is always “0”.

## 6.19 Pre Trigger Delay registers

```
#define SIS3316_ADC_CH1_4_PRE_TRIGGER_DELAY_REG          0x1028
#define SIS3316_ADC_CH5_8_PRE_TRIGGER_DELAY_REG           0x2028
#define SIS3316_ADC_CH9_12_PRE_TRIGGER_DELAY_REG          0x3028
#define SIS3316_ADC_CH13_16_PRE_TRIGGER_DELAY_REG         0x4028
```

These registers define the Pre Trigger Delay (0 to 2042/16.378).

The maximum Pre Trigger Delay value is 2042 for ADC\_FPGA firmware versions up to and including version V\_0xxx0006 and 16.378 for versions from version V\_0xxx0007.

| Bit | Function                                |
|-----|-----------------------------------------|
| 31  | reserved                                |
| ... |                                         |
| ..  |                                         |
| 16  | reserved                                |
| 15  | Additional Delay of Fir Trigger P+G Bit |
| 14  | reserved                                |
| 13  | Pretrigger Delay Bit 13                 |
| 12  | Pretrigger Delay Bit 12                 |
| 11  | Pretrigger Delay Bit 11                 |
| 10  | Pretrigger Delay Bit 10                 |
| ..  |                                         |
| 2   | Pretrigger Delay Bit 2                  |
| 1   | Pretrigger Delay Bit 1                  |
| 0   | Pretrigger Delay Bit 0 *                |

The power up default value is 0

\* Pretrigger Delay bit 0 is always “0”.

Valid values for Pre Trigger Delay : 0, 2, 4, 6 to 2042/16.378.

| Value of written<br>Pre Trigger Delay register | Pretrigger Delay<br>(<=V_0xxx0006) | Pretrigger Delay<br>(>=V_0xxx0007) |
|------------------------------------------------|------------------------------------|------------------------------------|
| 0 / 1                                          | 0                                  | 0                                  |
| 2 / 3                                          | 2                                  | 2                                  |
| ..                                             | ..                                 | ..                                 |
| 0x7FA / 0x7FB                                  | 2042                               | 2042                               |
| 0x7FC / 0x7FD                                  | 2042                               | 2044                               |
| 0x7FE / 0x7FF                                  | 2042                               | 2046                               |
| ..                                             | ..                                 | ..                                 |
| 0x2000 / 0x2001                                | 2042                               | 8192                               |
| ..                                             | ..                                 | ..                                 |
| 0x3FF8 / 0x3FF9                                | 2042                               | 16376                              |
| 0x3FFA / 0x3FFB                                | 2042                               | 16378                              |
| 0x3FFC / 0x3FFD                                | 2042                               | 16378                              |
| 0x3FFE / 0x3FFF                                | 2042                               | 16378                              |

## 6.20 Average Configuration registers

```
#define SIS3316_ADC_CH1_4_AVERAGE_CONFIGURATION_REG      0x102C
#define SIS3316_ADC_CH5_8_AVERAGE_CONFIGURATION_REG       0x202C
#define SIS3316_ADC_CH9_12_AVERAGE_CONFIGURATION_REG      0x302C
#define SIS3316_ADC_CH13_16_AVERAGE_CONFIGURATION_REG     0x402C
```

### 6.20.1 Average Configuration registers (SIS3316-125MHz-16bit)

These registers enable and define the Average mode. These registers are implemented in the Firmware “V0125-xxxx” (SIS3316-125MHz-16bit), only.

| Bit | Function                        |
|-----|---------------------------------|
| 31  | reserved                        |
| 30  | Average Mode Bit 2              |
| 29  | Average Mode Bit 1              |
| 28  | Average Mode Bit 0              |
| 27  | Average Pretrigger Delay Bit 11 |
| ..  | ..                              |
| 18  | Average Pretrigger Delay Bit 2  |
| 17  | Average Pretrigger Delay Bit 1  |
| 16  | Average Pretrigger Delay Bit 0  |
| 15  | Average Sample Length Bit 15    |
| ... | ..                              |
| ..  | ..                              |
| 2   | Average Sample Length Bit 2     |
| 1   | Average Sample Length Bit 1     |
| 0   | Average Sample Length Bit 0*    |

The power up default value is 0

\* Average Sample Length bit 0 is “0” always.

Valid values for Average Pretrigger Delay: 0, 1, 2 to 4094 .

Valid values for Average Sample Length: 0, 2, 4, 6 to 65534 .

Average Mode bits:

| Bit 2 | Bit 1 | Bit 0 | averaged samples         |
|-------|-------|-------|--------------------------|
| 0     | 0     | 0     | Average Mode is disabled |
| 0     | 0     | 1     | 4                        |
| 0     | 1     | 0     | 8                        |
| 0     | 1     | 1     | 16                       |
| 1     | 0     | 0     | 32                       |
| 1     | 0     | 1     | 64                       |
| 1     | 1     | 0     | 128                      |
| 1     | 1     | 1     | 256                      |

## 6.20.2 Average/Decimation Configuration registers (SIS3316-250MHz-14bit)

These registers enable and define the Average/Decimation mode. These registers are implemented in the Firmware “V0250-xxxx” (SIS3316-250MHz-14bit), only.

| Bit | Function                               |
|-----|----------------------------------------|
| 31  | reserved                               |
| 30  | reserved                               |
| 29  | reserved                               |
| 28  | Ch4 Average/Decimation Mode Enable Bit |
| 27  | Ch4 Average/Decimation Mode Bit 3      |
| 26  | Ch4 Average/Decimation Mode Bit 2      |
| 25  | Ch4 Average/Decimation Mode Bit 1      |
| 24  | Ch4 Average/Decimation Mode Bit 0      |
| 23  | reserved                               |
| 22  | reserved                               |
| 21  | reserved                               |
| 20  | Ch3 Average/Decimation Mode Enable Bit |
| 19  | Ch3 Average/Decimation Mode Bit 3      |
| 18  | Ch3 Average/Decimation Mode Bit 2      |
| 17  | Ch3 Average/Decimation Mode Bit 1      |
| 16  | Ch3 Average/Decimation Mode Bit 0      |
| 15  | reserved                               |
| 14  | reserved                               |
| 13  | reserved                               |
| 12  | Ch2 Average/Decimation Mode Enable Bit |
| 11  | Ch2 Average/Decimation Mode Bit 3      |
| 10  | Ch2 Average/Decimation Mode Bit 2      |
| 9   | Ch2 Average/Decimation Mode Bit 1      |
| 8   | Ch2 Average/Decimation Mode Bit 0      |
| 7   | reserved                               |
| 6   | reserved                               |
| 5   | reserved                               |
| 4   | Ch1 Average/Decimation Mode Enable Bit |
| 3   | Ch1 Average/Decimation Mode Bit 3      |
| 2   | Ch1 Average/Decimation Mode Bit 2      |
| 1   | Ch1 Average/Decimation Mode Bit 1      |
| 0   | Ch1 Average/Decimation Mode Bit 0      |

The power up default value is 0

## Chx Average/Decimation Mode bits:

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Averaged/Decimated samples |
|-------|-------|-------|-------|----------------------------|
| 0     | 0     | 0     | 0     | 4                          |
| 0     | 0     | 0     | 1     | 8                          |
| 0     | 0     | 1     | 0     | 16                         |
| 0     | 0     | 1     | 1     | 32                         |
| 0     | 1     | 0     | 0     | 64                         |
| 0     | 1     | 0     | 1     | 128                        |
| 0     | 1     | 1     | 0     | 256                        |
| 0     | 1     | 1     | 1     | 512                        |
| 1     | 0     | 0     | 0     | 2                          |
| 1     | 0     | 0     | 1     | 2                          |
| 1     | 0     | 1     | 0     | 2                          |
| 1     | 0     | 1     | 1     | 2                          |
| 1     | 1     | 0     | 0     | 2                          |
| 1     | 1     | 0     | 1     | 2                          |
| 1     | 1     | 1     | 0     | 2                          |
| 1     | 1     | 1     | 1     | 2                          |

## 6.21 Data Format Configuration registers

```
#define SIS3316_ADC_CH1_4_DATAFORMAT_CONFIG_REG      0x1030
#define SIS3316_ADC_CH5_8_DATAFORMAT_CONFIG_REG        0x2030
#define SIS3316_ADC_CH9_12_DATAFORMAT_CONFIG_REG       0x3030
#define SIS3316_ADC_CH13_16_DATAFORMAT_CONFIG_REG      0x4030
```

| Bit | Function                                                                                   |
|-----|--------------------------------------------------------------------------------------------|
| 31  |                                                                                            |
| 30  | Ch4 Save “User Counter” value                                                              |
| 29  | Ch4 Select Energy MAW Test Buffer bit                                                      |
| 28  | Ch4 Save MAW Test Buffer Enable bit                                                        |
| 27  | Ch4 Save Start Energy MAW value and Max. Energy MAW value                                  |
| 26  | Ch4 Save 3 x Fast Trigger MAW values (max value, value before Trigger, value with Trigger) |
| 25  | Ch4 Save 2 x Accumulator values (Gates 7,8)                                                |
| 24  | Ch4 Save Peak High values and 6 x Accumulator values (Gates 1,2, ..,6)                     |
| 23  |                                                                                            |
| 22  | Ch3 Save “User Counter” value                                                              |
| 21  | Ch3 Select Energy MAW Test Buffer bit                                                      |
| 20  | Ch3 Save MAW Test Buffer Enable bit                                                        |
| 19  | Ch3 Save Start Energy MAW value and Max. Energy MAW value                                  |
| 18  | Ch3 Save 3 x Fast Trigger MAW values (max value, value before Trigger, value with Trigger) |
| 17  | Ch3 Save 2 x Accumulator values (Gates 7,8)                                                |
| 16  | Ch3 Save Peak High values and 6 x Accumulator values (Gates 1,2, ..,6)                     |
| 15  |                                                                                            |
| 14  | Ch2 Save “User Counter” value                                                              |
| 13  | Ch2 Select Energy MAW Test Buffer bit                                                      |
| 12  | Ch2 Save MAW Test Buffer Enable bit                                                        |
| 11  | Ch2 Save Start Energy MAW value and Max. Energy MAW value                                  |
| 10  | Ch2 Save 3 x Fast Trigger MAW values (max value, value before Trigger, value with Trigger) |
| 9   | Ch2 Save 2 x Accumulator values (Gates 7,8)                                                |
| 8   | Ch2 Save Peak High values and 6 x Accumulator values (Gates 1,2, ..,6)                     |
| 7   |                                                                                            |
| 6   | Ch1 Save “User Counter” value                                                              |
| 5   | Ch1 Select Energy MAW Test Buffer bit                                                      |
| 4   | Ch1 Save MAW Test Buffer Enable bit                                                        |
| 3   | Ch1 Save Start Energy MAW value and Max. Energy MAW value                                  |
| 2   | Ch1 Save 3 x Fast Trigger MAW values (max value, value before Trigger, value with Trigger) |
| 1   | Ch1 Save 2 x Accumulator values (Gates 7,8)                                                |
| 0   | Ch1 Save Peak High values and 6 x Accumulator values (Gates 1,2, ..,6)                     |

Chx Select Energy MAW Test Buffer bit

| Select Energy bit | Function                 |
|-------------------|--------------------------|
| 0                 | FIR Trigger MAW selected |
| 1                 | FIR Energy MAW selected  |

## 6.22 MAW Test Buffer Configuration registers

```
#define SIS3316_ADC_CH1_4_MAWE_TEST_BUFFER_CONFIG_REG      0x1034
#define SIS3316_ADC_CH5_8_MAWE_TEST_BUFFER_CONFIG_REG      0x2034
#define SIS3316_ADC_CH9_12_MAWE_TEST_BUFFER_CONFIG_REG     0x3034
#define SIS3316_ADC_CH13_16_MAWE_TEST_BUFFER_CONFIG_REG    0x4034
```

These registers define the length and pretrigger delay of the saving of the FIR Trigger MAW or FIR Energy MAW (MW, Moving Window).

| Bit | Function                                |
|-----|-----------------------------------------|
| 31  | reserved                                |
| ... |                                         |
| 26  | reserved                                |
| 25  | MAW Test Buffer Pretrigger Delay Bit 9  |
| ..  |                                         |
| 18  | MAW Test Buffer Pretrigger Delay Bit 2  |
| 17  | MAW Test Buffer Pretrigger Delay Bit 1  |
| 16  | MAW Test Buffer Pretrigger Delay Bit 0* |
| 15  | MAW Test Buffer Length Bit 15 ***       |
| ... |                                         |
| 12  | MAW Test Buffer Length Bit 12 ***       |
| 11  | MAW Test Buffer Length Bit 11 ***       |
| 10  | MAW Test Buffer Length Bit 10           |
| 9   | MAW Test Buffer Length Bit 9            |
| ..  |                                         |
| ..  |                                         |
| 2   | MAW Test Buffer Length Bit 2            |
| 1   | MAW Test Buffer Length Bit 1            |
| 0   | MAW Test Buffer Length Bit 0**          |

The power up default value is 0

\* MAW Test Buffer Pretrigger Delay bit 0 is “0” always.

\*\* MAW Test Buffer Length bit 0 is “0” always.

\*\*\* new with ADC Version V-0250-000A and V-0125-000A

Valid values with ADC Version V-0250-0009 and V-0125-0009 and previous versions

MAW Test Buffer Pretrigger Delay :        2, 4, 6 to 1022

MAW Test Buffer Length :                  0, 2, 4, 6 to 1024

Valid values with ADC Version V-0250-000A and V-0125-000A

MAW Test Buffer Pretrigger Delay :        2, 4, 6 to 1022

MAW Test Buffer Length :                  0, 2, 4, 6 to 2048

The logic will set the internal MAW Test Buffer Length to 2048 in case the programmed number is greater than 2048.

### 6.23 Internal Trigger Delay Configuration registers

```
#define SIS3316_ADC_CH1_4_INTERNAL_TRIGGER_DELAY_CONFIG_REG      0x1038
#define SIS3316_ADC_CH5_8_INTERNAL_TRIGGER_DELAY_CONFIG_REG        0x2038
#define SIS3316_ADC_CH9_12_INTERNAL_TRIGGER_DELAY_CONFIG_REG       0x3038
#define SIS3316_ADC_CH13_16_INTERNAL_TRIGGER_DELAY_CONFIG_REG      0x4038
```

These registers define the Internal Trigger Delay. The value is to multiply by 2 Clocks.

| D31:24                         | D23:16                         | D15:8                          | D7:0                           |
|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| Ch 4<br>Internal Trigger Delay | Ch 3<br>Internal Trigger Delay | Ch 2<br>Internal Trigger Delay | Ch 1<br>Internal Trigger Delay |

The power up default value is 0

## 6.24 Internal Gate Length Configuration registers

```
#define SIS3316_ADC_CH1_4_INTERNAL_GATE_LENGTH_CONFIG_REG      0x103C
#define SIS3316_ADC_CH5_8_INTERNAL_GATE_LENGTH_CONFIG_REG        0x203C
#define SIS3316_ADC_CH9_12_INTERNAL_GATE_LENGTH_CONFIG_REG       0x303C
#define SIS3316_ADC_CH13_16_INTERNAL_GATE_LENGTH_CONFIG_REG      0x403C
```

These registers define the Internal Gate Length. The value is to be multiplied by 2 clocks.

| D31:24   | D23:20                           | D19:16                           | D15:8                   | D7:0                                   |
|----------|----------------------------------|----------------------------------|-------------------------|----------------------------------------|
| reserved | Gate 2 Enable<br>Ch4 Ch3 Ch2 Ch1 | Gate 1 Enable<br>Ch4 Ch3 Ch2 Ch1 | Internal Gate<br>Length | Internal<br>Coincidence<br>Gate Length |

The power up default value is 0

## 6.25 FIR Trigger Setup registers

```
#define SIS3316_ADC_CH1_FIR_TRIGGER_SETUP_REG          0x1040
#define SIS3316_ADC_CH2_FIR_TRIGGER_SETUP_REG          0x1050
#define SIS3316_ADC_CH3_FIR_TRIGGER_SETUP_REG          0x1060
#define SIS3316_ADC_CH4_FIR_TRIGGER_SETUP_REG          0x1070
#define SIS3316_ADC_CH1_4_SUM_FIR_TRIGGER_SETUP_REG    0x1080

#define SIS3316_ADC_CH5_FIR_TRIGGER_SETUP_REG          0x2040
#define SIS3316_ADC_CH6_FIR_TRIGGER_SETUP_REG          0x2050
#define SIS3316_ADC_CH7_FIR_TRIGGER_SETUP_REG          0x2060
#define SIS3316_ADC_CH8_FIR_TRIGGER_SETUP_REG          0x2070
#define SIS3316_ADC_CH5_8_SUM_FIR_TRIGGER_SETUP_REG    0x2080

#define SIS3316_ADC_CH9_FIR_TRIGGER_SETUP_REG          0x3040
#define SIS3316_ADC_CH10_FIR_TRIGGER_SETUP_REG         0x3050
#define SIS3316_ADC_CH11_FIR_TRIGGER_SETUP_REG         0x3060
#define SIS3316_ADC_CH12_FIR_TRIGGER_SETUP_REG         0x3070
#define SIS3316_ADC_CH9_12_SUM_FIR_TRIGGER_SETUP_REG   0x3080

#define SIS3316_ADC_CH13_FIR_TRIGGER_SETUP_REG         0x4040
#define SIS3316_ADC_CH14_FIR_TRIGGER_SETUP_REG         0x4050
#define SIS3316_ADC_CH15_FIR_TRIGGER_SETUP_REG         0x4060
#define SIS3316_ADC_CH16_FIR_TRIGGER_SETUP_REG         0x4070
#define SIS3316_ADC_CH13_16_SUM_FIR_TRIGGER_SETUP_REG 0x4080
```

These read/write registers hold the Peaking and Gap Time values of the trapezoidal FIR filter, Trigger Pulse Length value and Internal Gate Length value.

| Bit | Function                      |                                                                                         |
|-----|-------------------------------|-----------------------------------------------------------------------------------------|
| 31  | Pulse Length bit 7            |                                                                                         |
|     | ..                            |                                                                                         |
| 26  | ..                            |                                                                                         |
| 25  | Pulse Length bit 1            |                                                                                         |
| 24  | Pulse Length bit 0 (not used) | External NIM Out Trigger Pulse Length (stretched)<br>valid value = [2, 4, 6, ...., 256] |
| 23  | G bit 11                      |                                                                                         |
| 22  | G bit 10                      |                                                                                         |
| 21  | G bit 9                       |                                                                                         |
| 20  | G bit 8                       |                                                                                         |
| 19  | G bit 7                       | Gap time<br>(Flat Time)<br>valid value = [2, 4, 6, ...., 510]                           |
| 12  | G bit 0 (not used)            |                                                                                         |
| 11  | P bit 11                      |                                                                                         |
| 10  | P bit 10                      |                                                                                         |
| 9   | P bit 9                       |                                                                                         |
| 8   | P bit 8                       |                                                                                         |
| 7   | P bit 7                       |                                                                                         |
| ..  | ..                            |                                                                                         |
| .   | P bit 1                       |                                                                                         |
| 0   | P bit 0 (not used)            | P : Peaking time<br>$x+P$<br>$\sum_{i=x}^y S_i$<br>valid value = [2, 4, 6, ...., 510]   |

The power up default value reads 0x0

- Si: Sum of ADC input sample stream from x to x+P  
 P: Peaking time (number of values to sum)  
 G: Gap time, Flat Time

## 6.26 Trigger Threshold registers

```
#define SIS3316_ADC_CH1_FIR_TRIGGER_THRESHOLD_REG 0x1044
#define SIS3316_ADC_CH2_FIR_TRIGGER_THRESHOLD_REG 0x1054
#define SIS3316_ADC_CH3_FIR_TRIGGER_THRESHOLD_REG 0x1064
#define SIS3316_ADC_CH4_FIR_TRIGGER_THRESHOLD_REG 0x1074
#define SIS3316_ADC_CH1_4_SUM_FIR_TRIGGER_THRESHOLD_REG 0x1084

#define SIS3316_ADC_CH5_FIR_TRIGGER_THRESHOLD_REG 0x2044
#define SIS3316_ADC_CH6_FIR_TRIGGER_THRESHOLD_REG 0x2054
#define SIS3316_ADC_CH7_FIR_TRIGGER_THRESHOLD_REG 0x2064
#define SIS3316_ADC_CH8_FIR_TRIGGER_THRESHOLD_REG 0x2074
#define SIS3316_ADC_CH5_8_SUM_FIR_TRIGGER_THRESHOLD_REG 0x2084

#define SIS3316_ADC_CH9_FIR_TRIGGER_THRESHOLD_REG 0x3044
#define SIS3316_ADC_CH10_FIR_TRIGGER_THRESHOLD_REG 0x3054
#define SIS3316_ADC_CH11_FIR_TRIGGER_THRESHOLD_REG 0x3064
#define SIS3316_ADC_CH12_FIR_TRIGGER_THRESHOLD_REG 0x3074
#define SIS3316_ADC_CH9_12_SUM_FIR_TRIGGER_THRESHOLD_REG 0x3084

#define SIS3316_ADC_CH13_FIR_TRIGGER_THRESHOLD_REG 0x4044
#define SIS3316_ADC_CH14_FIR_TRIGGER_THRESHOLD_REG 0x4054
#define SIS3316_ADC_CH15_FIR_TRIGGER_THRESHOLD_REG 0x4064
#define SIS3316_ADC_CH16_FIR_TRIGGER_THRESHOLD_REG 0x4074
#define SIS3316_ADC_CH13_16_SUM_FIR_TRIGGER_THRESHOLD_REG 0x4084
```

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

| Bit      | 31             | 30                                | 29-28            | 27-0                    |
|----------|----------------|-----------------------------------|------------------|-------------------------|
| Function | Trigger Enable | High Energy Suppress Trigger Mode | CFD control bits | Trigger threshold value |

default after reset: 0x0

CFD control bit setting table:

| CFD Control Bit 1 | CFD Control Bit 0 | CFD function                            |
|-------------------|-------------------|-----------------------------------------|
| 0                 | 0                 | CFD function disabled                   |
| 0                 | 1                 | CFD function disabled                   |
| 1                 | 0                 | CFD function enabled with Zero crossing |
| 1                 | 1                 | CFD function enabled with 50%           |

### High Energy Suppress Trigger Mode:

A trigger will be suppressed if the running sum of the trapezoidal filter goes above the value of the High Energy Trigger Threshold register.

**This mode works only with CFD function enabled !**

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.  
The running sum is build with full accuracy.

The full 27-bit running sum + 0x800 0000 is compared to the threshold value of the Trigger Threshold register to generate the Trigger pulse.

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

## 6.27 High Energy Trigger Threshold registers

```
#define SIS3316_ADC_CH1_FIR_HIGH_ENERGY_THRESHOLD_REG 0x1048
#define SIS3316_ADC_CH2_FIR_HIGH_ENERGY_THRESHOLD_REG 0x1058
#define SIS3316_ADC_CH3_FIR_HIGH_ENERGY_THRESHOLD_REG 0x1068
#define SIS3316_ADC_CH4_FIR_HIGH_ENERGY_THRESHOLD_REG 0x1078
#define SIS3316_ADC_CH1_4_SUM_FIR_HIGH_ENERGY_THRESHOLD_REG 0x1088

#define SIS3316_ADC_CH5_FIR_HIGH_ENERGY_THRESHOLD_REG 0x2048
#define SIS3316_ADC_CH6_FIR_HIGH_ENERGY_THRESHOLD_REG 0x2058
#define SIS3316_ADC_CH7_FIR_HIGH_ENERGY_THRESHOLD_REG 0x2068
#define SIS3316_ADC_CH8_FIR_HIGH_ENERGY_THRESHOLD_REG 0x2078
#define SIS3316_ADC_CH5_8_SUM_FIR_HIGH_ENERGY_THRESHOLD_REG 0x2088

#define SIS3316_ADC_CH9_FIR_HIGH_ENERGY_THRESHOLD_REG 0x3048
#define SIS3316_ADC_CH10_FIR_HIGH_ENERGY_THRESHOLD_REG 0x3058
#define SIS3316_ADC_CH11_FIR_HIGH_ENERGY_THRESHOLD_REG 0x3068
#define SIS3316_ADC_CH12_FIR_HIGH_ENERGY_THRESHOLD_REG 0x3078
#define SIS3316_ADC_CH9_12_SUM_FIR_HIGH_ENERGY_THRESHOLD_REG 0x3088

#define SIS3316_ADC_CH13_FIR_HIGH_ENERGY_THRESHOLD_REG 0x4048
#define SIS3316_ADC_CH14_FIR_HIGH_ENERGY_THRESHOLD_REG 0x4058
#define SIS3316_ADC_CH15_FIR_HIGH_ENERGY_THRESHOLD_REG 0x4068
#define SIS3316_ADC_CH16_FIR_HIGH_ENERGY_THRESHOLD_REG 0x4078
#define SIS3316_ADC_CH13_16_SUM_FIR_HIGH_ENERGY_THRESHOLD_REG 0x4088
```

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

| Bit      | 31                               | 30                                                                             | 29-28                                                               | 27-0                                |
|----------|----------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------------|
| Function | Trigger on both edges enable bit | Internal High Energy Trigger Streched Output Pulse to VME FPGA multiplexer bit | Internal Trigger Streched Output Pulse to VME FPGA multiplexer bits | High Energy Trigger Threshold value |

default after reset: 0x0

The full 27-bit running sum + 0x800 0000 is compared to the High Energy threshold value.

### Trigger on both edges enable bit:

(this function is implemented in the Firmware “adc\_fpga\_V-0125-0004” and higher, only CFD mode has to be enabled to use this function !

|        |                                                                                                                                                                                                                                              |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bit 31 | Internal Trigger generation if                                                                                                                                                                                                               |
| 0      | the running sum of the trapezoidal filter goes <b>above</b> the value of the Trigger Threshold register                                                                                                                                      |
| 1      | the running sum of the trapezoidal filter goes <b>above</b> the value of the Trigger Threshold register<br>and if<br>the running sum of the trapezoidal filter goes <b>below</b> the <b>negative</b> value of the Trigger Threshold register |

### Internal Trigger Streched Output Pulse to VME FPGA multiplexer bits setting table:

|                                                                      |       |                                                                                                                       |
|----------------------------------------------------------------------|-------|-----------------------------------------------------------------------------------------------------------------------|
| Internal Trigger Stretched Output Pulse to VME FPGA multiplexer bits |       | Routed to VME FPGA as Internal Trigger (used for LEMO Out selection, global Trigger feedback or Trigger Lookup table) |
| Bit 1                                                                | Bit 0 |                                                                                                                       |
| 0                                                                    | 0     | Ch x Internal Trigger                                                                                                 |
| 0                                                                    | 1     | Ch x High Energy Trigger                                                                                              |
| 1                                                                    | 0     | Ch x Pileup Pulse                                                                                                     |
| 1                                                                    | 1     | reserved                                                                                                              |

**Ch x Internal Trigger:**

An internal Trigger stretched output pulse is routed to the VME FPGA as “Internal trigger”.

A trigger output pulse will be generated if the running sum of the trapezoidal filter goes above the value of the Trigger Threshold register.

**Ch x High Energy Trigger:**

An internal High Energy Trigger stretched output pulse is routed to the VME FPGA as “Internal trigger”.

A trigger output pulse will be generated if the running sum of the trapezoidal filter goes above the value of the High Energy Trigger Threshold register and the CFD mode is enabled.

**Ch x Pileup Pulse:**

A Pileup Trigger stretched output pulse is routed to the VME FPGA as “Internal trigger”.

A trigger output pulse will be generated if an internal trigger occurs while the Pileup window is valid.

A Pileup window starts (restarts) with each internal trigger.

Furthermore it is possible to use the High Energy Trigger Threshold register to generate a second trigger (HE-Trigger) which could be routed for example to the LEMO output UO.

With this feature, in combination with the “High Energy Suppress Trigger Mode“ is disabled, the HE-Trigger can be used as “main trigger” at LEMO output UO.

**Internal High Energy Trigger Stretched Output Pulse to VME FPGA multiplexer bit = 0:**

A HE-Trigger stretched output pulse is routed to the VME FPGA as “Internal HE-trigger”.

**Internal High Energy Trigger Stretched Output Pulse to VME FPGA multiplexer bit = 1:**

A Pileup stretched output pulse is routed to the VME FPGA as “Internal HE-trigger”.

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

## 6.28 Trigger Statistic Counter Mode register

```
#define SIS3316_ADC_CH1_4_TRIGGER_STATISTIC_COUNTER_MODE_REG 0x1090
#define SIS3316_ADC_CH5_8_TRIGGER_STATISTIC_COUNTER_MODE_REG 0x2090
#define SIS3316_ADC_CH9_12_TRIGGER_STATISTIC_COUNTER_MODE_REG 0x3090
#define SIS3316_ADC_CH13_16_TRIGGER_STATISTIC_COUNTER_MODE_REG 0x4090
```

These read/write register hold features of the Trigger Statistic Counters .

|          |          |             |
|----------|----------|-------------|
| Bit      | 31-1     | 0           |
| Function | reserved | Update Mode |

default after Reset: 0x0

Update Mode = 0: Readout of the actual Trigger-Statistic-Counters

Update Mode = 1: Readout of the latched (freezed) Trigger-Statistic-Counters  
The Readout-Trigger-Statistic-Counter-Latches will be latched  
with each bank switching (at the end of a bank sampling).

## 6.29 Peak/Charge Configuration registers

|                                                           |        |
|-----------------------------------------------------------|--------|
| #define SIS3316_ADC_CH1_4_PEAK_CHARGE_CONFIGURATION_REG   | 0x1094 |
| #define SIS3316_ADC_CH5_8_PEAK_CHARGE_CONFIGURATION_REG   | 0x2094 |
| #define SIS3316_ADC_CH9_12_PEAK_CHARGE_CONFIGURATION_REG  | 0x3094 |
| #define SIS3316_ADC_CH13_16_PEAK_CHARGE_CONFIGURATION_REG | 0x4094 |

These registers enable and define the Peak/Charge mode.

| Bit | Function                       |
|-----|--------------------------------|
| 31  | Enable Peak/Charge Mode        |
| 30  | reserved                       |
| 29  | Baseline Average Mode Bit 1    |
| 28  | Baseline Average Mode Bit 0    |
| 27  | Baseline Pregate Delay Bit 11  |
| ..  | ..                             |
| ..  | ..                             |
| 18  | Baseline Pregate Delay Bit 2   |
| 17  | Baseline Pregate Delay Bit 1   |
| 16  | Baseline Pregate Delay Bit 0 * |
| 15  | reserved                       |
| ... | ..                             |
| ..  | ..                             |
| 2   | reserved                       |
| 1   | reserved                       |
| 0   | reserved                       |

The power up default value is 0

\* Baseline Pregate Delay bit 0 is “0” always.

Valid values for Baseline Pregate Delay: 0, 2, 4, 6 to 510.

Baseline Average Mode bits:

| Bit 1 | Bit 0 | averaged samples |
|-------|-------|------------------|
| 0     | 0     | 32               |
| 0     | 1     | 64               |
| 1     | 0     | 128              |
| 1     | 1     | 256              |

### 6.30 Extended Raw Data Buffer Configuration registers

```
#define SIS3316_ADC_CH1_4_EXTENDED_RAW_DATA_BUFFER_CONFIG_REG 0x1098
#define SIS3316_ADC_CH5_8_EXTENDED_RAW_DATA_BUFFER_CONFIG_REG 0x2098
#define SIS3316_ADC_CH9_12_EXTENDED_RAW_DATA_BUFFER_CONFIG_REG 0x3098
#define SIS3316_ADC_CH13_16_EXTENDED_RAW_DATA_BUFFER_CONFIG_REG 0x4098
```

With these registers, it is possible to define a greater sample length than 64K, which can be defined with the Raw Data Buffer Configuration registers.

The value of the “Raw Buffer Sample\_Length“ will be used for the sample length if the value of the “**Extended** Raw Buffer Sample\_Length“ is zero !

If the value of the “**Extended** Raw Buffer Sample\_Length“ is not zero then this value will be used for the sample length !

The maximum “Raw Buffer Sample\_Length“ is 32M – 2 (33.554.430).

See: Raw Data Buffer Configuration register

| Bit | Function                                 |
|-----|------------------------------------------|
| 31  | reserved                                 |
| ... |                                          |
| ..  |                                          |
| 26  | reserved                                 |
| 25  | reserved                                 |
| 24  | Extended Raw Buffer Sample_Length Bit 24 |
| 23  | Extended Raw Buffer Sample_Length Bit 23 |
| ... |                                          |
| ..  |                                          |
| 2   | Extended Raw Buffer Sample_Length Bit 2  |
| 1   | Extended Raw Buffer Sample_Length Bit 1  |
| 0   | Extended Raw Buffer Sample_Length Bit 0* |

The power up default value is 0

\* Extended Raw Buffer Sample\_Length bit 0 is always “0”.

Data are stored to memory in packets of 2 consecutive samples by the copy logic.

**Note:** In case of use these “Extended Raw Data Buffer Configuration” registers to define the sample length, the internal “Address Threshold Flag” is used to suppress following triggers.

### 6.31 Accumulator Gate X Configuration registers

```

#define SIS3316_ADC_CH1_4_ACCUMULATOR_GATE1_CONFIG_REG      0x10A0
#define SIS3316_ADC_CH5_8_ACCUMULATOR_GATE1_CONFIG_REG      0x20A0
#define SIS3316_ADC_CH9_12_ACCUMULATOR_GATE1_CONFIG_REG     0x30A0
#define SIS3316_ADC_CH13_16_ACCUMULATOR_GATE1_CONFIG_REG    0x40A0

#define SIS3316_ADC_CH1_4_ACCUMULATOR_GATE2_CONFIG_REG      0x10A4
#define SIS3316_ADC_CH5_8_ACCUMULATOR_GATE2_CONFIG_REG      0x20A4
#define SIS3316_ADC_CH9_12_ACCUMULATOR_GATE2_CONFIG_REG     0x30A4
#define SIS3316_ADC_CH13_16_ACCUMULATOR_GATE2_CONFIG_REG    0x40A4

..
#define SIS3316_ADC_CH1_4_ACCUMULATOR_GATE8_CONFIG_REG      0x10BC
#define SIS3316_ADC_CH5_8_ACCUMULATOR_GATE8_CONFIG_REG      0x20BC
#define SIS3316_ADC_CH9_12_ACCUMULATOR_GATE8_CONFIG_REG     0x30BC
#define SIS3316_ADC_CH13_16_ACCUMULATOR_GATE8_CONFIG_REG    0x40BC

```

These registers define the length and the Start Index of the Accumulator Gates.

| D31:25 | D24:16      | D15:0                      |
|--------|-------------|----------------------------|
| 0      | Gate Length | Gate Start Index (Address) |

| Value of Gate Length register | Number of sample(s) |
|-------------------------------|---------------------|
| 0                             | 1                   |
| 1                             | 2                   |
| 2                             | 3                   |
| ..                            | ..                  |
| 0xFF                          | 0x200 (512)         |

### 6.32 FIR Energy Setup registers

|                                               |        |
|-----------------------------------------------|--------|
| #define SIS3316_ADC_CH1_FIR_ENERGY_SETUP_REG  | 0x10C0 |
| #define SIS3316_ADC_CH2_FIR_ENERGY_SETUP_REG  | 0x10C4 |
| #define SIS3316_ADC_CH3_FIR_ENERGY_SETUP_REG  | 0x10C8 |
| #define SIS3316_ADC_CH4_FIR_ENERGY_SETUP_REG  | 0x10CC |
| <br>                                          |        |
| #define SIS3316_ADC_CH5_FIR_ENERGY_SETUP_REG  | 0x20C0 |
| #define SIS3316_ADC_CH6_FIR_ENERGY_SETUP_REG  | 0x20C4 |
| #define SIS3316_ADC_CH7_FIR_ENERGY_SETUP_REG  | 0x20C8 |
| #define SIS3316_ADC_CH8_FIR_ENERGY_SETUP_REG  | 0x20CC |
| <br>                                          |        |
| #define SIS3316_ADC_CH9_FIR_ENERGY_SETUP_REG  | 0x30C0 |
| #define SIS3316_ADC_CH10_FIR_ENERGY_SETUP_REG | 0x30C4 |
| #define SIS3316_ADC_CH11_FIR_ENERGY_SETUP_REG | 0x30C8 |
| #define SIS3316_ADC_CH12_FIR_ENERGY_SETUP_REG | 0x30CC |
| <br>                                          |        |
| #define SIS3316_ADC_CH13_FIR_ENERGY_SETUP_REG | 0x40C0 |
| #define SIS3316_ADC_CH14_FIR_ENERGY_SETUP_REG | 0x40C4 |
| #define SIS3316_ADC_CH15_FIR_ENERGY_SETUP_REG | 0x40C8 |
| #define SIS3316_ADC_CH16_FIR_ENERGY_SETUP_REG | 0x40CC |

These read/write registers hold the Peaking Time, Gap Time and Tau Factor (decay time) values of the trapezoidal FIR Energy filter.

| Bit | Function           |                                                 |
|-----|--------------------|-------------------------------------------------|
| 31  | Tau table bit 1    | Select the Tau table                            |
| 30  | Tau table bit 0    |                                                 |
| 29  | Tau factor bit 5   |                                                 |
| ..  | ..                 |                                                 |
| 25  | Tau factor bit 1   | Tau factor<br>valid value = [0, 1, 2, ...., 63] |
| 24  | Tau factor bit 0   |                                                 |
| 23  | Extra filter bit 1 | Extra filter                                    |
| 22  | Extra filter bit 0 |                                                 |
| 21  | G bit 9            |                                                 |
| 20  | G bit 8            |                                                 |
| 19  | G bit 7            | Gap time<br>(Flat Time)                         |
| ..  | ..                 |                                                 |
| 12  | G bit 0 (not used) | valid value = [2, 4, 6, ...., 510]              |
| 11  | P bit 11           |                                                 |
| 10  | P bit 10           |                                                 |
| 9   | P bit 9            | P : Peaking time                                |
| 8   | P bit 8            | x+P                                             |
| 7   | P bit 7            | $\sum Si$                                       |
| ..  | ..                 | i = x                                           |
| .   | P bit 1            | valid value = [2, 4, 6, ...., 2044]             |
| 0   | P bit 0 (not used) |                                                 |

The power up default value reads 0x0

Si: Sum of ADC input sample stream from x to x+P

P: Peaking time (number of values to sum)

G: Gap time, Flat Time

Extra filter table:

| Bit 1 | Bit 0 | function        |
|-------|-------|-----------------|
| 0     | 0     | No extra filter |
| 0     | 1     | Average of 4    |
| 1     | 0     | Average of 8    |
| 1     | 1     | Average of 16   |

The decay time depends on the Tau table value, the Tau factor and on the sample clock.  
 Please have a look to the program “sis3316\_energy\_tau\_factor\_calculator.cpp” and to text files “sis3316-14Bit\_250MHz.txt”, “sis3316-14Bit\_125MHz.txt” and “sis3316-16Bit\_125MHz.txt”.

An extract of the text file sis3316-14Bit\_250MHz.txt” (usec):

```
SIS3316-14Bit: Tau factor - Decay Time table

sample_clock = 250.000000 MHz

i_table = 0  uint_tau_factor = 1      double_decay_time_us = 2097.150000
i_table = 0  uint_tau_factor = 2      double_decay_time_us = 1048.574000
i_table = 0  uint_tau_factor = 3      double_decay_time_us = 699.048667
..
i_table = 0  uint_tau_factor = 61     double_decay_time_us = 34.377541
i_table = 0  uint_tau_factor = 62     double_decay_time_us = 33.823032
i_table = 0  uint_tau_factor = 63     double_decay_time_us = 33.286127

i_table = 1  uint_tau_factor = 1      double_decay_time_us = 524.286000
i_table = 1  uint_tau_factor = 2      double_decay_time_us = 262.142000
i_table = 1  uint_tau_factor = 3      double_decay_time_us = 174.760667
..
i_table = 1  uint_tau_factor = 61     double_decay_time_us = 8.592885
i_table = 1  uint_tau_factor = 62     double_decay_time_us = 8.454258
i_table = 1  uint_tau_factor = 63     double_decay_time_us = 8.320032

i_table = 2  uint_tau_factor = 1      double_decay_time_us = 131.070000
i_table = 2  uint_tau_factor = 2      double_decay_time_us = 65.534000
i_table = 2  uint_tau_factor = 3      double_decay_time_us = 43.688667
..
i_table = 2  uint_tau_factor = 61     double_decay_time_us = 2.146721
i_table = 2  uint_tau_factor = 62     double_decay_time_us = 2.112064
i_table = 2  uint_tau_factor = 63     double_decay_time_us = 2.078507

i_table = 3  uint_tau_factor = 1      double_decay_time_us = 32.766000
i_table = 3  uint_tau_factor = 2      double_decay_time_us = 16.382000
i_table = 3  uint_tau_factor = 3      double_decay_time_us = 10.920667
..
i_table = 3  uint_tau_factor = 61     double_decay_time_us = 0.535178
i_table = 3  uint_tau_factor = 62     double_decay_time_us = 0.526514
i_table = 3  uint_tau_factor = 63     double_decay_time_us = 0.518124
```

### 6.33 Energy Histogram Configuration registers

```
#define SIS3316_ADC_CH1_ENERGY_HISTOGRAM_CONFIG_REG          0x10D0
#define SIS3316_ADC_CH2_ENERGY_HISTOGRAM_CONFIG_REG          0x10D4
#define SIS3316_ADC_CH3_ENERGY_HISTOGRAM_CONFIG_REG          0x10D8
#define SIS3316_ADC_CH4_ENERGY_HISTOGRAM_CONFIG_REG          0x10DC

#define SIS3316_ADC_CH5_ENERGY_HISTOGRAM_CONFIG_REG          0x20D0
#define SIS3316_ADC_CH6_ENERGY_HISTOGRAM_CONFIG_REG          0x20D4
#define SIS3316_ADC_CH7_ENERGY_HISTOGRAM_CONFIG_REG          0x20D8
#define SIS3316_ADC_CH8_ENERGY_HISTOGRAM_CONFIG_REG          0x20DC

#define SIS3316_ADC_CH9_ENERGY_HISTOGRAM_CONFIG_REG          0x30D0
#define SIS3316_ADC_CH10_ENERGY_HISTOGRAM_CONFIG_REG         0x30D4
#define SIS3316_ADC_CH11_ENERGY_HISTOGRAM_CONFIG_REG         0x30D8
#define SIS3316_ADC_CH12_ENERGY_HISTOGRAM_CONFIG_REG         0x30DC

#define SIS3316_ADC_CH13_ENERGY_HISTOGRAM_CONFIG_REG         0x40D0
#define SIS3316_ADC_CH14_ENERGY_HISTOGRAM_CONFIG_REG         0x40D4
#define SIS3316_ADC_CH15_ENERGY_HISTOGRAM_CONFIG_REG         0x40D8
#define SIS3316_ADC_CH16_ENERGY_HISTOGRAM_CONFIG_REG         0x40DC
```

These read/write registers configure the histogramming of the Energy.

| Bit | Function                                          |                                                |
|-----|---------------------------------------------------|------------------------------------------------|
| 31  | Writing Hits/Events into Event Memory Disable bit |                                                |
| 30  | Histogram clear with Timestamp-Clear Disable bit  |                                                |
| 29  | reserved                                          |                                                |
| 28  | reserved                                          |                                                |
| 27  | Divider bit 11                                    | Energy Divider<br><br>value = 0 is not allowed |
| 26  | Divider bit 10                                    |                                                |
| ..  |                                                   |                                                |
| 17  | Divider bit 1                                     |                                                |
| 16  | Divider bit 0                                     |                                                |
| 11  | Subtract Offset bit 7                             | Energy Subtract Offset                         |
| 10  | Subtract Offset bit 6                             |                                                |
| 10  | ..                                                |                                                |
| 9   | Subtract Offset bit 1                             |                                                |
| 8   | Subtract Offset bit 0                             |                                                |
| 7   | reserved                                          |                                                |
| 6   | reserved                                          |                                                |
| 5   | reserved                                          |                                                |
| 4   | reserved                                          |                                                |
| 3   | reserved                                          |                                                |
| 2   | reserved                                          |                                                |
| 1   | Pileup Enable bit                                 |                                                |
| 0   | Histogramming Enable bit                          |                                                |

The power up default value reads 0x0

**Histogramming Enable bit = 0:** the Histogram logic is disabled  
**Histogramming Enable bit = 1:** the Histogram logic is enabled

If the Sample logic is armed on Bank1 or Bank2 and if the Histogram logic is enabled then the Energy of each Hit/Event will be histogrammed in the histogram space of the Event Memory Ch x.

**Pileup Enable bit = 0:** don't increment histogram in case of pileup/repileup  
**Pileup Enable bit = 1:** increment histogram in case of pileup/repileup

**Histogram clear with Timestamp-Clear Disable bit = 0:**

The 64K bin Histogram is cleared with a Timestamp-Clear command.

**Histogram clear with Timestamp-Clear Disable bit = 1:**

The 64K bin Histogram is not cleared with a Timestamp-Clear command.

**Writing Hits/Events into Event Memory Disable bit = 0:**

If the Sample logic is armed on Bank1 or Bank2 and if the Histogram logic is enabled then the Sample logic writes Hits/Events to the Event Memory Ch x and the Histogram logic increments the Histogram simultaneously.

**Writing Hits/Events into Event Memory Disable bit = 1**

disables the writing of the Hits/Events into the Event Memory. The Hits/Events are histogrammed, only.

### 6.33.1 Calculation of the Energy to the Histogram Index:

The Histogram memory has a length of 64K bins and it is implemented in the last 256Kbyte page of the Event Memory ADC channel x Bank1.

- Event Memory 1/2 Channel 1/3/5/..15 32-bit address offset = 0 x 00FF 0000
- Event Memory 1/2 Channel 2/4/6/..16 32-bit address offset = 0 x 02FF 0000

$$\text{Histogram Index} = (\text{max. Energy value} / \text{Energy Divider}) - (\text{Energy Subtract Offset} * 0x100)$$

Example (SIS3316-250MHz-14bit):

Input Range: 5 V

Signal 100mV -> appr. 328 counts with Peaktime = 100 -> Energy value = appr. 32800

Energy Divider: 100

Energy Subtract Offset: 0

$$\text{Histogram Index} = (32800 / 100) - (0) = 328$$

Energy Divider: 25

Energy Subtract Offset: 2

$$\text{Histogram Index} = (32800 / 25) - (2 * 256) = 1312 - 512 = 800$$

### 6.34 MAW Start Index and Energy Pickup Configuration registers

|                                                                   |        |
|-------------------------------------------------------------------|--------|
| #define SIS3316_ADC_CH1_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x10E0 |
| #define SIS3316_ADC_CH2_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x10E4 |
| #define SIS3316_ADC_CH3_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x10E8 |
| #define SIS3316_ADC_CH4_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x10EC |
| <br>                                                              |        |
| #define SIS3316_ADC_CH5_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x20E0 |
| #define SIS3316_ADC_CH6_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x20E4 |
| #define SIS3316_ADC_CH7_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x20E8 |
| #define SIS3316_ADC_CH8_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x20EC |
| <br>                                                              |        |
| #define SIS3316_ADC_CH9_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG  | 0x30E0 |
| #define SIS3316_ADC_CH10_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x30E4 |
| #define SIS3316_ADC_CH11_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x30E8 |
| #define SIS3316_ADC_CH12_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x30EC |
| <br>                                                              |        |
| #define SIS3316_ADC_CH13_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x40E0 |
| #define SIS3316_ADC_CH14_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x40E4 |
| #define SIS3316_ADC_CH15_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x40E8 |
| #define SIS3316_ADC_CH16_MA_WSTART_INDEX_ENERGY_PICKUP_CONFIG_REG | 0x40EC |

These registers define the start index of the MAW Test data buffer and the Index at which the Energy will be taken from the Energy trapezoid (MAW) instead of the maximum value.

| D31:16              | D15:0                       |
|---------------------|-----------------------------|
| Energy Pickup Index | MAW Test Buffer Start Index |

The power up default value is 0

With the value of the “MAW Test Buffer Start Index“, it is possible to delay the start of saving the Trigger or Energy trapezoid into the MAW Test buffer. This can be helpful to save (see) the tail of the Energy trapezoid in case that the sum (Pretiggerdelay + 2\*Peaking time + Gap time) is greater than 2048.

You can save the Energy value from a defined position (index) of the Energy trapezoidal rather than the maximum value of the trapezoidal with a non zero Energy Pickup Index value. With a zero Energy Pickup Index value the logic will save the maximum value.

### 6.35 Test Function registers

```
#define SIS3316_ADC_CH1_4_TEST_FUNCTION_REG      0x10F0
#define SIS3316_ADC_CH5_8_TEST_FUNCTION_REG        0x20F0
#define SIS3316_ADC_CH9_12_TEST_FUNCTION_REG       0x30F0
#define SIS3316_ADC_CH13_16_TEST_FUNCTION_REG      0x40F0
```

| Bit | Function                                             |
|-----|------------------------------------------------------|
| 31  | reserved                                             |
| ... |                                                      |
| ..  |                                                      |
| 26  | reserved                                             |
| 25  | reserved                                             |
| 24  | reserved                                             |
| 23  | reserved                                             |
| ... |                                                      |
| ..  |                                                      |
| 2   | reserved                                             |
| 1   | reserved                                             |
| 0   | Save internal Gate as Bit 0 of ADC data (Ch1 to Ch4) |

### 6.36 ADC FPGA Firmware Version Register

```
#define SIS3316_ADC_CH1_4_FIRMWARE_REG      0x1100 /* rd only */
#define SIS3316_ADC_CH5_8_FIRMWARE_REG        0x2100 /* rd only */
#define SIS3316_ADC_CH9_12_FIRMWARE_REG       0x3100 /* rd only */
#define SIS3316_ADC_CH13_16_FIRMWARE_REG      0x4100 /* rd only */
```

This register reflects the ADC FPGA Firmware Version and Revision number. The Version level will be used to distinguish between substantial design differences and experiment specific designs, while the Revision level will be used to mark user specific adaptations.

| Bits | 31 - 16       | 15 - 8           | 7 - 0             |
|------|---------------|------------------|-------------------|
|      | Firmware Type | Firmware Version | Firmware Revision |

Example:

Firmware Type = 0x0125 : 125 MHz 16-bit ADC

Firmware Type = 0x0250 : 250 MHz 14-bit ADC

Firmware Version = 0x00: 1. official Version from 9.11.2012

Firmware Revision = 0x01: 1. official Revision from 9.11.2012

### 6.37 ADC FPGA Status register

```
#define SIS3316_ADC_CH1_4_STATUS_REG      0x1104 /* rd only */
#define SIS3316_ADC_CH5_8_STATUS_REG        0x2104 /* rd only */
#define SIS3316_ADC_CH9_12_STATUS_REG       0x3104 /* rd only */
#define SIS3316_ADC_CH13_16_STATUS_REG      0x4104 /* rd only */
```

This register holds the ADC FPGA Status of the Data Link between the ADC FPGA and the VME FPGA , the Status of the both Memory Controller and the Status of the ADC-Clock DCM.

| Bit | read                           |
|-----|--------------------------------|
| 31  | 0                              |
| ... | 0...                           |
| 24  | 0                              |
| 23  | 0                              |
| 22  | 0                              |
| 21  | ADC-Clock DCM RESET flag       |
| 20  | ADC-Clock DCM OK flag          |
| 19  | 0                              |
| 18  | 0                              |
| 17  | Memory 2 OK flag (ch3 and ch4) |
| 16  | Memory 1 OK flag (ch1 and ch2) |
| 15  | 0                              |
| 14  | 0                              |
| 13  | 0                              |
| 12  | 0                              |
| 11  | 0                              |
| 10  | 0                              |
| 9   | 0                              |
| 8   | Data Link Speed flag           |
| 7   | VME FPGA : Frame_error_latch   |
| 6   | VME FPGA : Soft_error_latch    |
| 5   | VME FPGA : Hard_error_latch    |
| 4   | VME FPGA : Lane_up_flag        |
| 3   | VME FPGA : Channel_up_flag     |
| 2   | VME FPGA : Frame_error_flag    |
| 1   | VME FPGA : Soft_error_flag     |
| 0   | VME FPGA : Hard_error_flag     |

The latched error bits may be set after power up. After a write with 0x400 (i.e. setting bit 10) to the corresponding ADC Input tap delay register the contents of the bits 7 to 0 should read 0x18.

|                      |                                                         |
|----------------------|---------------------------------------------------------|
| Data Link Speed flag | Internal data transfer speed                            |
| 0                    | 1.25 GHz (required for modules with serial number 1-10) |
| 1                    | 2.5 GHz                                                 |

### 6.38 Actual Sample address registers

```
#define SIS3316_ADC_CH1_ACTUAL_SAMPLE_ADDRESS_REG      0x1110 /* rd only */
#define SIS3316_ADC_CH2_ACTUAL_SAMPLE_ADDRESS_REG      0x1114 /* rd only */
#define SIS3316_ADC_CH3_ACTUAL_SAMPLE_ADDRESS_REG      0x1118 /* rd only */
#define SIS3316_ADC_CH4_ACTUAL_SAMPLE_ADDRESS_REG      0x111C /* rd only */

#define SIS3316_ADC_CH5_ACTUAL_SAMPLE_ADDRESS_REG      0x2110 /* rd only */
#define SIS3316_ADC_CH6_ACTUAL_SAMPLE_ADDRESS_REG      0x2114 /* rd only */
#define SIS3316_ADC_CH7_ACTUAL_SAMPLE_ADDRESS_REG      0x2118 /* rd only */
#define SIS3316_ADC_CH8_ACTUAL_SAMPLE_ADDRESS_REG      0x211C /* rd only */

#define SIS3316_ADC_CH9_ACTUAL_SAMPLE_ADDRESS_REG      0x3110 /* rd only */
#define SIS3316_ADC_CH10_ACTUAL_SAMPLE_ADDRESS_REG     0x3114 /* rd only */
#define SIS3316_ADC_CH11_ACTUAL_SAMPLE_ADDRESS_REG     0x3118 /* rd only */
#define SIS3316_ADC_CH12_ACTUAL_SAMPLE_ADDRESS_REG     0x311C /* rd only */

#define SIS3316_ADC_CH13_ACTUAL_SAMPLE_ADDRESS_REG     0x4110 /* rd only */
#define SIS3316_ADC_CH14_ACTUAL_SAMPLE_ADDRESS_REG     0x4114 /* rd only */
#define SIS3316_ADC_CH15_ACTUAL_SAMPLE_ADDRESS_REG     0x4118 /* rd only */
#define SIS3316_ADC_CH16_ACTUAL_SAMPLE_ADDRESS_REG     0x411C /* rd only */
```

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

**Note:** the Actual Sample Address points to 32-bit words.

| Bit |                                                                      |
|-----|----------------------------------------------------------------------|
| 31  | unused, read as 0                                                    |
| ... |                                                                      |
| 26  | unused, read as 0                                                    |
| 25  | Actual Sample Address Register Bit 25 : indicates the Channel Offset |
| 24  | Actual Sample Address Register Bit 24 : indicates the Bank           |
| 23  | Actual Sample Address Register Bit 23                                |
| ..  | ..                                                                   |
| 2   | Actual Sample Address Register Bit 2                                 |
| 1   | Actual Sample Address Register Bit 1                                 |
| 0   | Actual Sample Address Register Bit 0                                 |

The power up default value is 0

### 6.39 Previous Bank Sample address registers

```
#define SIS3316_ADC_CH1_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x1120
#define SIS3316_ADC_CH2_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x1124
#define SIS3316_ADC_CH3_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x1128
#define SIS3316_ADC_CH4_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x112C

#define SIS3316_ADC_CH5_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x2120
#define SIS3316_ADC_CH6_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x2124
#define SIS3316_ADC_CH7_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x2128
#define SIS3316_ADC_CH8_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x212C

#define SIS3316_ADC_CH9_PREVIOUS_BANK_SAMPLE_ADDRESS_REG      0x3120
#define SIS3316_ADC_CH10_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x3124
#define SIS3316_ADC_CH11_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x3128
#define SIS3316_ADC_CH12_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x312C

#define SIS3316_ADC_CH13_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x4120
#define SIS3316_ADC_CH14_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x4124
#define SIS3316_ADC_CH15_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x4128
#define SIS3316_ADC_CH16_PREVIOUS_BANK_SAMPLE_ADDRESS_REG     0x412C
```

These read only registers hold the stored next sampling address of the previous bank. It is the stop address + 1;

**Note:** the Previous Bank Sample Address points to 32-bit words.

| Bit |                                                                             |
|-----|-----------------------------------------------------------------------------|
| 31  | unused, read as 0                                                           |
| ... |                                                                             |
| 26  | unused, read as 0                                                           |
| 25  | Previous Bank Sample Address Register Bit 25 : indicates the Channel Offset |
| 24  | Previous Bank Sample Address Register Bit 24 : indicates the Bank           |
| 23  | Previous Bank Sample Address Register Bit 23                                |
| ..  |                                                                             |
| 2   | Previous Bank Sample Address Register Bit 2                                 |
| 1   | Previous Bank Sample Address Register Bit 1                                 |
| 0   | Previous Bank Sample Address Register Bit 0                                 |

The power up default value is 0

## 6.40 Key addresses (0x400 – 0x43C write only)

Write access (with Broadcast functionality) to a key address (KA) with arbitrary data invokes a respective action.

### 6.40.1 Key address: Register Reset

```
#define SIS3316_KEY_RESET           0x400 /* write only; D32 */
```

A write with arbitrary data to this register (key address) resets the SIS3316 registers to its power up state.

### 6.40.2 Key address: User function logic

```
#define SIS3316_KEY_USER_FUNCTION    0x404 /* write only; D32 */
```

A write with arbitrary data to this register (key address) will generate an “User function pulse”.

### 6.40.3 Key address: Arm sample logic

```
#define SIS3316_KEY_ARM             0x410 /* write only; D32 */
```

A write with arbitrary data to this register (key address) will arm the “Single Bank Mode” sample logic. (not implemented yet)

### 6.40.4 Key address: Disarm sample logic

```
#define SIS3316_KEY_DISARM          0x414 /* write only; D32 */
```

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

### 6.40.5 Key address: Trigger

```
#define SIS3316_KEY_TRIGGER         0x418 /* write only; D32 */
```

A write with arbitrary data to this register (key address) will generate a trigger.

#### 6.40.6 Key address: Timestamp Clear

```
#define SIS3316_KEY_TIMESTAMP_CLR      0x41C /* write only; D32 */
```

A write with arbitrary data to this register (key address) will clear the 48-bit Timestamp Counter. If the ENERGY Histogramming mode is enabled it will clear the Energy Histograms, also.

If the “Prescaler Output Pulse Divider” register is unequal 0 it will enable (start) the “Prescaler Output Pulse” Logic, also.

#### 6.40.7 Key address: Disarm Bankx and Arm Bank1

```
#define SIS3316_KEY_DISARM_AND_ARM_BANK1    0x420 /* write only; D32 */
```

A write with arbitrary data to this register (key address) will disarm the sampling logic of the active bank and will arm the sample logic of bank1.

#### 6.40.8 Key address: Disarm Bankx and Arm Bank2

```
#define SIS3316_KEY_DISARM_AND_ARM_BANK2    0x424 /* write only; D32 */
```

A write with arbitrary data to this register (key address) will disarm the sampling logic of the active bank and will arm the sample logic of bank2.

#### 6.40.9 Key address: Enable Sample Bank Swap Control with NIM Input TI/UI Logic

```
#define SIS3316_KEY_ENABLE_SAMPLE_BANK_SWAP_CONTROL_WITH_NIM_INPUT 0x428
```

A write with arbitrary data to this register (key address) will enable the “Sample Bank Swap Control with NIM Input TI/UI” Logic.

If the logic is enabled then the next pulse on NIM input TI or UI (one has to be enabled/selected) will arm the Sample Bank logic on Bank1. With each following pulse on the NIM input the logic will toggle the active Bank. With a Key “Disarm sample logic” the logic will be disabled.

**6.40.10 Key address: Disable Prescaler Output Pulse Divider logic**

```
#define SIS3316_KEY_DISABLE_PRESCALER_OUTPUT_PULSE_DIVIDER_LOGIC 0x42C
```

A write with arbitrary data to this register (key address) disables the Prescaler Output Pulse logic.

**6.40.11 Key address: PPS\_Latch\_Bit\_clear**

```
#define SIS3316_KEY_PPS_LATCH_BIT_CLEAR          0x430 /* write only; D32 */
```

A write with arbitrary data to this register (key address) clears the PPS Latch bit.

**6.40.12 Key address: Reset ADC-FPGA-Logic**

```
#define SIS3316_KEY_ADC_FPGA_RESET              0x434 /* write only; D32 */
```

A write with arbitrary data to this register (key address) resets the ADC-FPGA logic (including the DDR3-memory controller). Used for test purposes only.

**6.40.13 Key address: ADC Clock DCM/PLL Reset**

```
#define SIS3316_KEY_ADC_CLOCK_DCM_RESET         0x438 /* write only;  
D32 */
```

A write with arbitrary data to this register (key address) resets the DCM/PLL of all FPGAs. This command is necessary after a change in Sample Clock.

## 7 LEDs

### 7.1 Front Panel LEDs

The SIS3316 has 4 front panel LEDs, which are listed in the table below.

| LED designator | LED comment | Function                                     |
|----------------|-------------|----------------------------------------------|
| A              | Access      | Signals VME access or Ethernet access        |
| U              | User        | Set/cleared via bit 0/16 of control register |
| 1              | 1           | Firmware dependent                           |
| 2              | 2           | Firmware dependent                           |

While the VME/CTRL FPGA is booting all four LEDs (A, U, 1, 2) are on.

After the VME/CTRL FPGA has booted the logic boots the four ADC FPGAs one after the other. The Leds are flashing with 5 Hz for appr. 1 second for each ADC FPGA (1, 2, 3 and 4) in the following order: A, 2, U and 1.

After this booting process all Leds are off.

The meaning of the LEDs U, 1 and 2 are defined by the Control/Status register of the granted interface (VME or Ethernet interface).

### 7.2 SMD LEDs

8 red surface mount LEDs are on the SIS3316 to visualize part of the board status.

| LED designator | LED comment | Function                                                       |
|----------------|-------------|----------------------------------------------------------------|
| D200A          | TX FAULT    | Signals SFP transmit fault<br>(lit without SFP installed also) |
| D200B          | RX LOS      | Signals SFP link loss<br>(lit without SFP installed also)      |
| D850A          | D+1.2V      | Signals presence of 1.2V power                                 |
| D851A          | D+2.5V_AUX  | Signals presence of 2.5V power                                 |
| D852A          | D+2.5V_VCCO | Signals presence of 2.5V power                                 |
| D853A          | D+3.3V_VCCO | Signals presence of 3.3V power                                 |
| D854A          | D+1.5V      | Signals presence of 1.5V power                                 |
| D856A          | D+1.8V      | Signals presence of 1.8V power                                 |

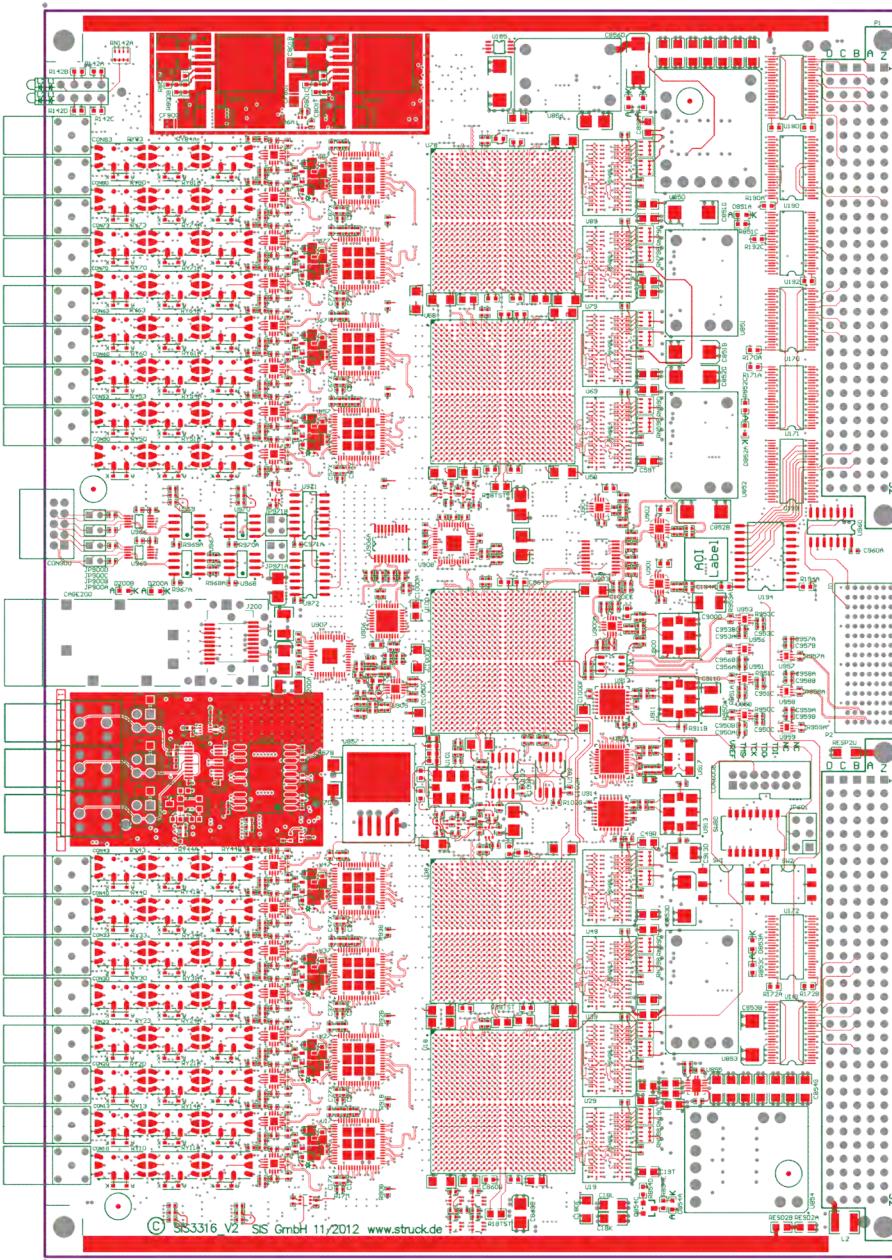
## 8 Front panel

A drawing of the SIS3316 front panel is shown below.



## 9 Board Layout

A print of the silk screen of the component side of the SIS3316 V2 PCB is shown below.



## 9.1 Configuration switches and jumpers

The configuration switches and jumpers of the SIS3316 card are described in the following subsections.

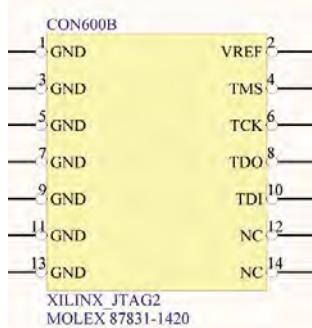
### 9.1.1 CON600B JTAG

The SIS3316 on board logic can load its firmware from a SPI PROM, via the JTAG port on connector CON600B or over VME. A list of firmware designs can be found under <http://www.struck.de/sis3316firm.htm>.

Hardware like the XILINX HW-USB-JTAG in connection with the appropriate software (typically Impact from the XILINX ISE environment) will be required for in field JTAG firmware upgrades.

CON600B is a 2mm (i.e. metric) 14 pin header that allows you to reprogram the firmware of the SIS3316 with a JTAG programmer. The pin out is shown in the schematic below. It is compatible with the cable that comes with the XILINX HW-USB platform cable.

The schematic for CON600B is shown below.

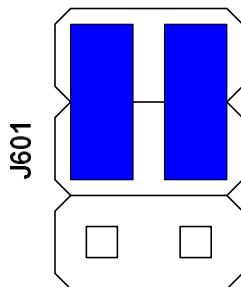


**Note:** The SIS3316 has to be powered for reprogramming over JTAG.

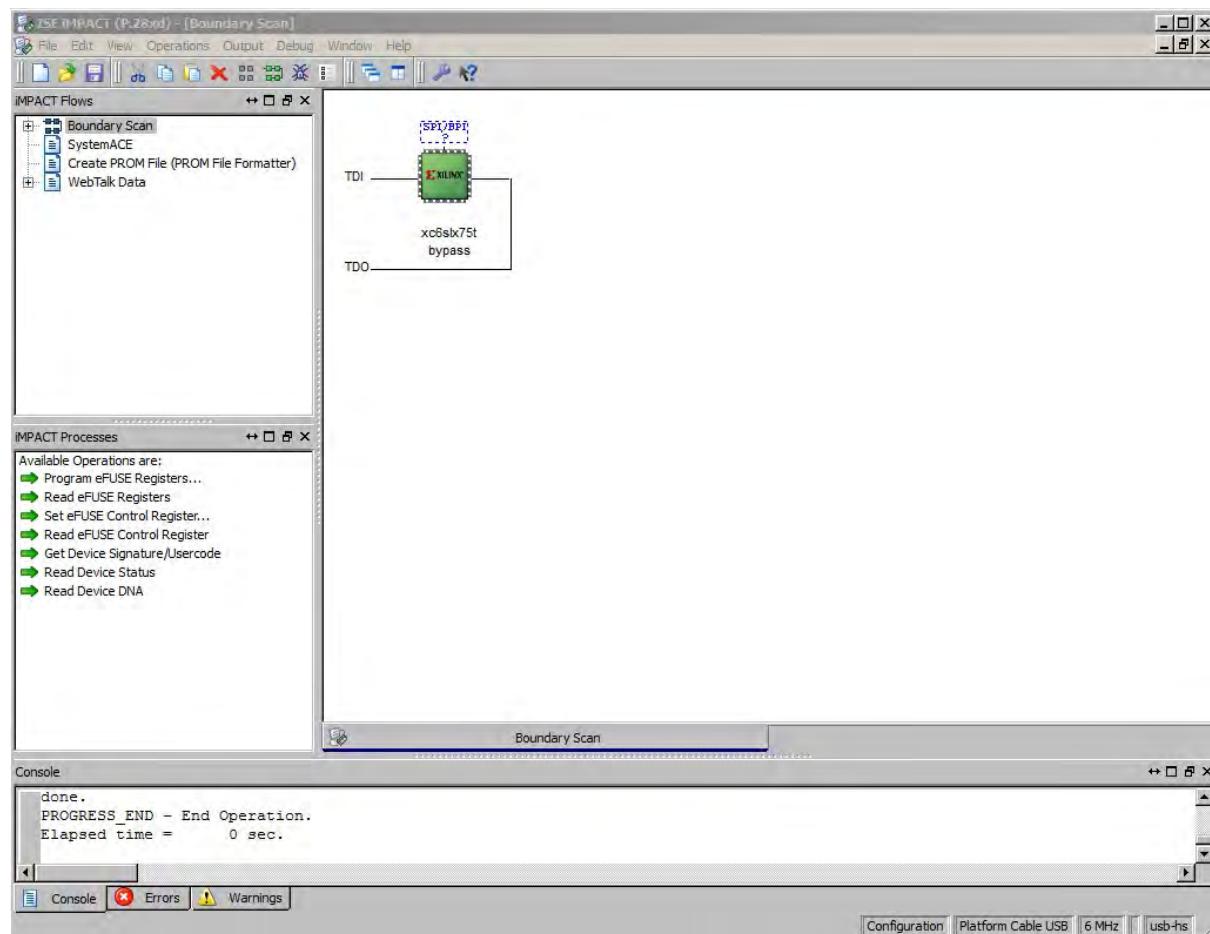
### 9.1.2 J601 JTAG chain

The JTAG chain on the SIS3316 can be configured to comprise the VME Xilinx only (short JTAG chain) or to comprise all 5 FPGAs (long chain). The configuration is selected with the 6-pin 2.54 mm jumper array J601 as sketched below:

**Short Chain** (3-5 and 4-6 closed, factory default, recommended for in field upgrades):

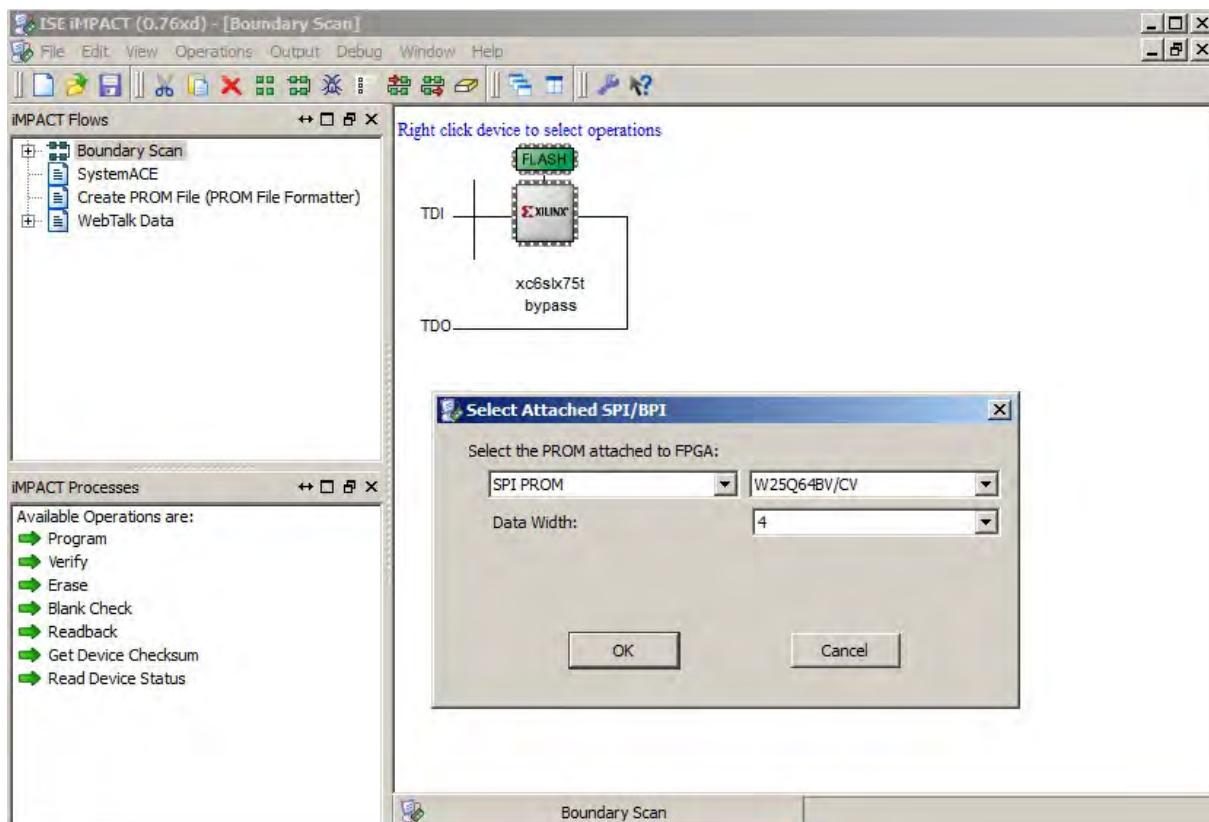


In the ISE Impact software you will see VME Spartan 6 Xilinx with the SPI PROMs attached.

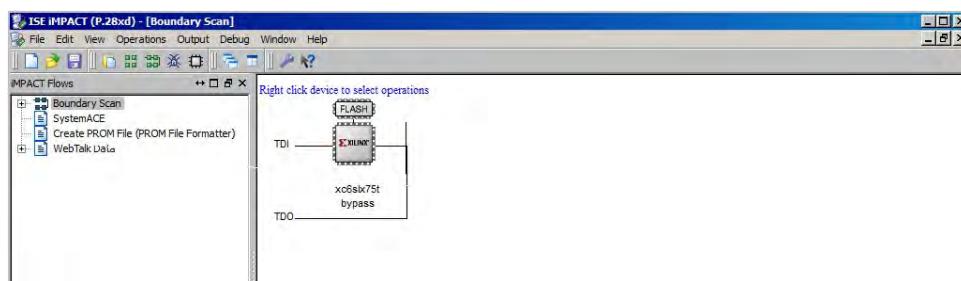


**Note: ISE Impact 13.3, 14.4, 14.6 and 14.7 are working properly  
ISE Impact 14.1, 14.2 and 14.3 are NOT working !**

By right clicking the SPI PROM and selecting a programming file and choosing W25Q64BV/CV as SPI PROM type and a data width of 4 -as shown below- you get access to the FLASH attached to the FPGA.



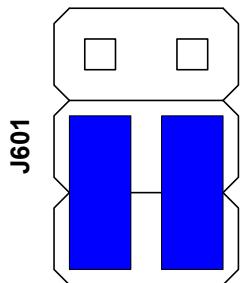
Your screen will look as shown below then.



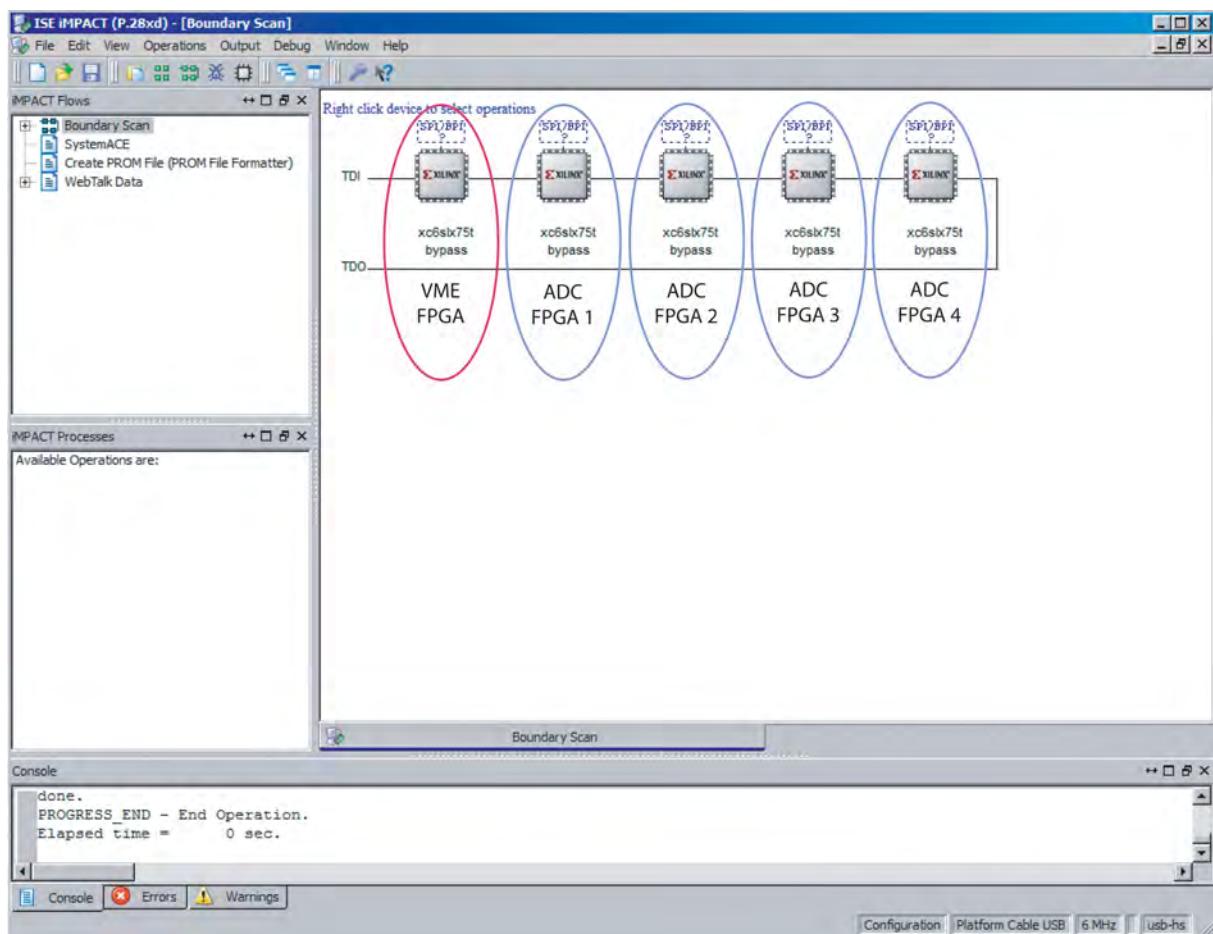
### 9.1.2.1 Recommended JTAG Firmware Upgrade procedure:

- 1.) Select short JTAG chain
- 2.) Disable watchdog (SW80-7 off)
- 3.) Connect JTAG cable and power up unit
- 4.) Use Impact 14.4
- 5.) Load MCS file (sis3316\_250MHz\_v2002a0002.mcs e.g.) into SPI PROM
- 6.) Power down unit
- 7.) Enable watchdog (SW80-7 on)
- 8.) Power up unit

**Long Chain (1-3 and 2-4 closed, experts only):**



In the ISE Impact software you will see all 5 Xilinx FPGAs as shown below:

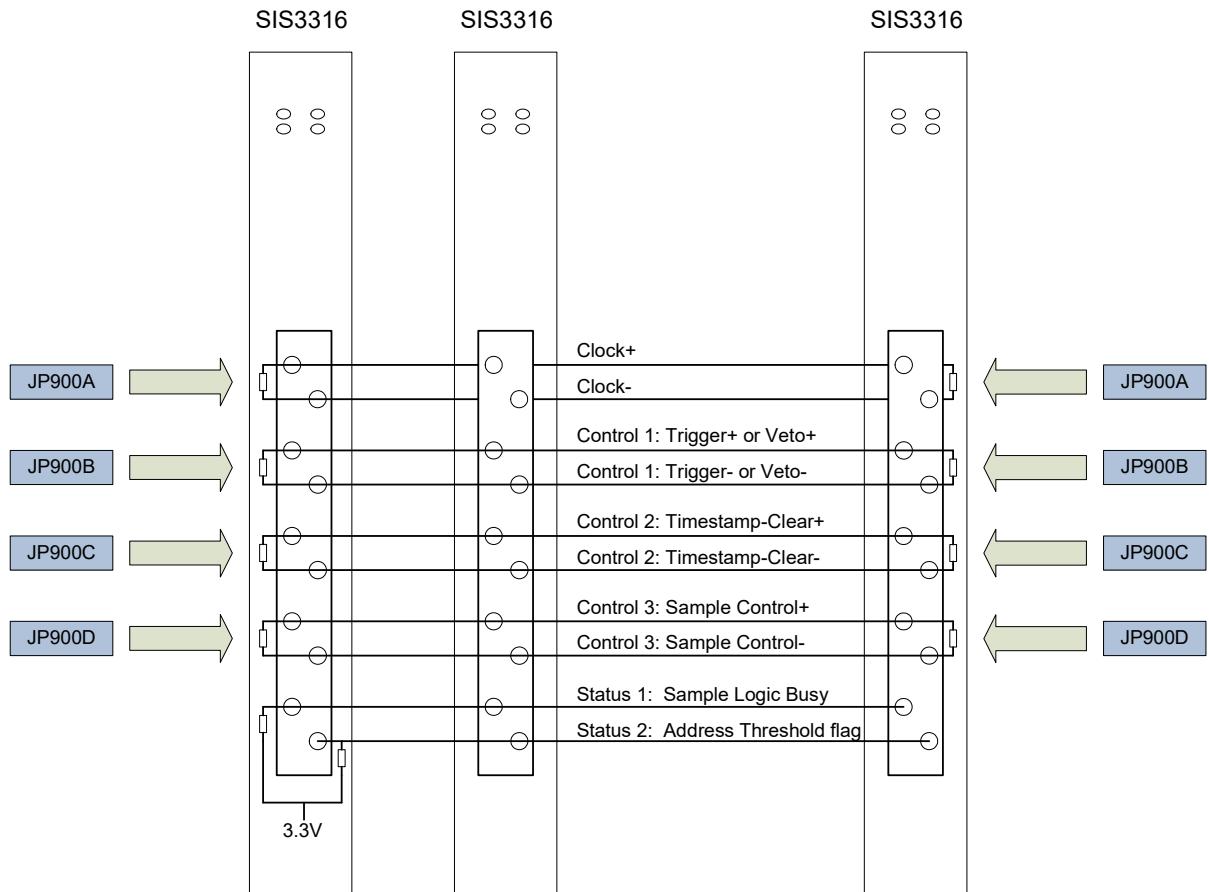


### 9.1.3 JP900A-D LVDS termination

This set of four 2mm jumpers is used to set the 100 Ohm termination on the first and the last unit in the LVDS chain.

| <b>Jumper</b> | <b>Signal</b>   | <b>terminated</b> | <b>unterminated</b> |
|---------------|-----------------|-------------------|---------------------|
| JP900A        | LVDS_Clock      | 1-2 closed        | 1-2 open (default)  |
| JP900B        | LVDS_TRG        | 1-2 closed        | 1-2 open (default)  |
| JP900C        | LVDS_Timestamp  | 1-2 closed        | 1-2 open (default)  |
| JP900D        | LVDS_USER1/Veto | 1-2 closed        | 1-2 open (default)  |

Schematic of a SIS3316 LVDS chain setup.



#### 9.1.4 JP903A/B/C NIM input termination

This set of two 2mm jumper arrays is used to set the termination on the NIM inputs.

| Jumper | Signal        | Terminated into 50 Ω | Unterminated |
|--------|---------------|----------------------|--------------|
| JP903A | NIM_CLK       | 1-2 closed (default) | 3-4 closed   |
| JP903B | NIM_TRG       | 1-2 closed (default) | 3-4 closed   |
| JP903C | NIM_Timestamp | 1-2 closed (default) | 3-4 closed   |

**Note:** Pin 1 has a square footprint

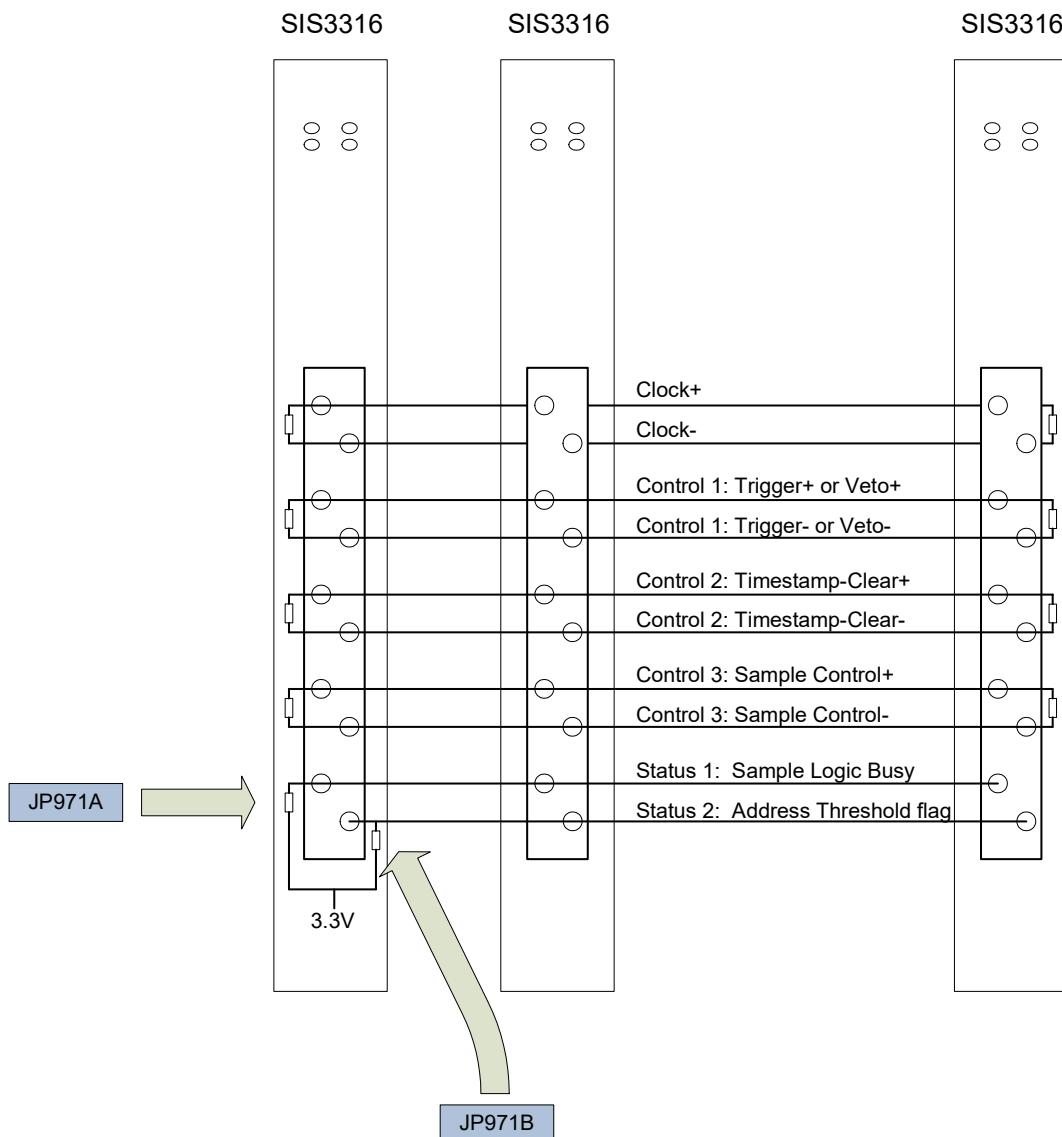
### 9.1.5 JP971A/B Open collector termination

This set of two 2mm jumper arrays is used to set the open collector termination of the two open collector lines on the flat cable front connector for the first unit in the chain.

**Note:** Pin 1 has a square footprint

| Jumper | Signal                      | Terminated | 4.7 KΩ to Unterminated         |
|--------|-----------------------------|------------|--------------------------------|
|        |                             | 3.3V       |                                |
| JP971A | User2/Busy                  | 1-2 closed | JP971A<br>3-4 closed (default) |
| JP971B | User3/<br>Address Threshold | 1-2 closed | JP971B<br>3-4 closed (default) |

Schematic of the open collector termination



### 9.1.6 SW1/SW2 VME Base Address Rotary switches

The two rotary switches of SW1 and SW2 are used to set the base address. The function may depend on switch setting of SW80 also (refer to the VME addressing section 4.4.2). SW1 defines bits 31 to 28 and SW2 bits 27 to 24.

### 9.1.7 SW80

SW80 is an 8 position dual inline switch. The functions are listed in the table below.

| SW80    | Off Function                               | On Function                           |
|---------|--------------------------------------------|---------------------------------------|
| O       | Disable A32 slave addressing               | Enable A32 slave addressing           |
| ↑       | Analog power off                           | Analog power on                       |
| ↑↑      | Deactivate auto-negotiation                | Activate auto-negotiation             |
| ↑↑↑     | DHCP disable                               | DHCP enable                           |
| ↑↑↑↑    | Not used                                   | Not used                              |
| ↑↑↑↑↑   | Not used                                   | Not used                              |
| ↑↑↑↑↑↑  | Watchdog disable                           | Watchdog enable                       |
| ↑↑↑↑↑↑↑ | Disconnect VME SYSRESET from<br>FPGA reset | Connect VME SYSRESET to<br>FPGA reset |

**Note:** The watchdog switch has to be in the disable position to install firmware over JTAG onto a unit that is no longer booting (initial installation or corrupted firmware)

## 10 Appendix

### 10.1 Power Consumption

The currents drawn by SIS3316 cards with 0x000B/0x200B firmware at different sampling speeds are listed in the table below.

| Module         | Sampling Clock | 5V    | 12V   | -12V  |
|----------------|----------------|-------|-------|-------|
| SIS3316-125-16 | 62.5 MHz       | 9,0 A | 1,1 A | 0,2 A |
| SIS3316-125-16 | 125 MHz        | 9,6 A | 1,1 A | 0,2 A |
| SIS3316-250-14 | 62.5 MHz       | 8,1 A | 1,1 A | 0,2 A |
| SIS3316-250-14 | 125 MHz        | 8,4 A | 1,1 A | 0,2 A |
| SIS3316-250-14 | 250 MHz        | 9,1 A | 1,1 A | 0,2 A |

These currents may vary substantially depending on the installed firmware implementation.

### 10.2 FPGA Firmware Update over VME

The firmware for the FPGAs is stored in a FlashPROM (configuration PROM). The firmware is loaded into the FPGAs from the FlashPROM at power up automatically.

The user has to write the contents of new FPGA firmware image files (VME FPGA firmware image file and ADC FPGA firmware image file) into the FlashPROM to update the firmware of the SIS3316.

Refer to the “sis3316\_root\_gui” software project and its documentation (SIS3316-M-Software-root\_gui-1-Vxxx..pdf) for details.

Within the project you will find the two files “sis3316\_adc.h” and “sis3316\_adc.cpp”.

See:

```
int sis3316_adc::update_firmware(char *path, int offset, void (*cb)(int percentage))
```

**Note:** Refer to section 9.1.2 in case of corrupted data in the FlashPROM for instructions on how to roll in firmware over the JTAG connector.

### 10.3 Connector types

The VME connectors and the different types of front panel connectors used on the SIS3316 are:

| Connector     | Purpose                  | Part Number            |
|---------------|--------------------------|------------------------|
| 160 pin zabcd | VME P1/P2                | Harting 02 01 160 2101 |
| Double LEMO   | NIM Control I/O          | LEMO EPY.00.250.NTN    |
| 90° PCB LEMO  | Analog input connector   | LEMO EPL.00.250.NTN    |
| LVDS          | LVDS Bus                 | Molex 87833-1031       |
| LVDS          | LVDS Bus cable connector | Molex 87568-1074       |
| SFP Housing   | SFP Card Cage            | TYCO 148779-1          |
| SFP Connector | SFP Connector            | TYCO 1367073-1         |

### 10.4 ADC/Digitizer Chip

Find below a table with the two types of digitizer chips, that are used on the two sampling speed/resolution variants of the SIS3316.

| Sampling Speed | ADC Chip                      |
|----------------|-------------------------------|
| 125 MSPS       | Analog Devices AD9268BCPZ-125 |
| 250 MSPS       | Analog Devices AD9643BCPZ-250 |

### 10.5 Ordering options

The SIS3316 comes in different base configurations. Feel free to inquire about application specific versions.

| Struck part number | Description                             |
|--------------------|-----------------------------------------|
| 04327              | SIS3316-250-14 2V and 5V input range    |
| 06391              | SIS3316-250-14 200mV and 2V input range |
| 04678              | SIS3316-125-16 2V and 5V input range    |

### 10.6 Accessories

Following SFP link media are available

| Struck part number | Coaxial Connectors  |
|--------------------|---------------------|
| 03145              | 4Gbit Short Wave FC |
| 04333              | 1000BASE-T Copper   |

Find below a table with other items related to the SIS3316 card.

| Struck part number | Component                                            |
|--------------------|------------------------------------------------------|
| 05410              | SIS3316 LVDS Bus Cable<br>2, 3, 4, 5 or 6 Connectors |
| Tbd.               | Desktop kit for SIS3316<br>Including 1000BASE-T SFP  |

## 10.7 Software

A ready to run example ROOT GUI (Graphical User Interface) comes with the SIS3316.

The CERN ROOT home page can be found at <http://root.cern.ch>

At this point in time code for the VC++10 Windows and Linux (Eclipse) versions are distributed with the product DVD.

### 10.7.1 Firmware Upgrade over VME or Ethernet

One section of the –to be expanded– SIS3316-M-Software-1-Vxxx.pdf software manual describes the firmware upgrade procedure.

### 10.7.2 Ethernet operation

Ethernet based operation of the SIS3316 is described in the Ethernet addendum to this manual. The SIS3316\_test\_gui\_etherent program is available as executable for Windows/Linux/MAC-OS and in source code.

## 11 Index

- 1000BASE-T 188
- 12 V 186
- 12 V 186
- 5 V 186
- A32** 185
- accessories 187
- Acquisition Control 38
- AD7314 101
- AD9268 18, 187
- AD9643 18, 187
- ADC 187
- ADC FPGA 186
- ADC FPGA Application Firmware 43
- Address Map 84
- address threshold 28
- address threshold flag 28
- Address Threshold flag 35
- airflow 101
- analog input stage 22
- Appendix 186
- arm 172
- Arm Banks 29
- auto-negotiation** 185
- Average 80
- Averaging 78
- Bank2 flag 28, 29
- base address 185
- Block diagram 17
- board layout 177
- broadcast
  - address 99
  - enable 98
  - master 98
- Broadcast 98
- calibration 135
- CFD control 153
- CFD Trigger Feature 51
- Charge Sensing ADC 77
- Chx input invert bit 154, 156
- CI 33, 114
- CO 70, 119
- Coincidence Gate Logic 47
- CON600B 178
- configuration 178
- connector types 187
- cooling 101
- D+1.2V 175
- D+1.5V 175
- D+1.8V 175
- D+2.5V\_AUX 175
- D+2.5V\_VCCO 175
- D+3.3V\_VCCO 175
- D200A 175
- D200B 175
- D850A 175
- D851A 175
- D852A 175
- D854A 175
- D856A 175
- data format 69
- Data Format
  - Average 79
  - Peak/Charge ADC 77
- data width 180
- dead time
  - Hit/Event storage 71
- dead time
  - Sample Bank Swap logic 70
- dead time 70
- decay correction 62
- Decimation 80
- desktop 188
- DHCP enable/disable** 185
- Digitizer 187
- DO8 95
- EEPROM 102
- energy filter 61
- energy pickup index functionality 65
- energy resolution 63
- Ethernet MAC 104
- Ethernet operation 189
- External (global) signal distribution 37
- External (global) Signal overview 32
- External NIM 33
- External Sample Control I/Os 33
- External Timestamp clear 32
- External Trigger 32
- External Veto 32
- Extra filter 162
- FADC250 24
- FIR filter 61
- firmware 178
- firmware update 186
- firmware upgrade 189
- Firmware upgrade 180
- FlashPROM 186
- Four Channel group blockdiagram 44
- FPGA
  - ADC 186
  - VME 186
- FPGA Firmware Update over VME 186
- front panel 176
- Front panel LVDS Bus 35
- G 61
- GA 83
- Gamma application 61
- Gamma parameters 64
- Gap Time 61
- Gate 1 Enable 151
- Gate 2 Enable 151
- Global Trigger 35
- Global Veto 35
- GND 178
- GUI 189
- Harting 187
- High Energy Suppress Trigger Mode 57
- Higher Timestamp Resolution 52
- Histogram 163
- Hit/Event Data Format 69
- HW-USB 178
- Impact 179
- input
  - TI 31
  - UI 31
- input range 131, 187
- Input tap delay

- Input tap delay 169
- input termination 183
- Internal Coincidence Gate Length 151
- Internal Gate Length 151
- Internal Programmable Sample Clock Oscillator 19
- internal trigger counter 128
- Internal Trigger Delay 150
- Internal Trigger Generation 49
- Internal Trigger to VME FPGA multiplexer bits setting table 156
- interrupter mode 95
- interrupter type 95
- introduction 14
- IRQ
  - End Address Threshold 96
  - IRQ mode 95
    - ROAK 95
    - RORA 95
- ISE 179
- J601 179
- JLab 24
- JP900A 182
- JP900B 182
- JP900D 182
- JP903A 183
- JP903B 183
- JP903C 183
- JP971A 184
- JP971B 184
- JTAG 178, 180, 186
- JTAG chain 179
- jumpers 178
- KA
  - ADC Clock DCM/PLL Reset 174
  - arm sample logic 172
  - Disable Prescaler Output Pulse Divider 174
  - Disarm Bankx and Arm Bank1 173
  - Disarm Bankx and Arm Bank2 173
  - disarm sample logic 172
  - Enable Sample Bank Swap Control with NIM Input TI/UI Logic 173
  - PPS\_Latch\_Bit\_clear 174
  - register reset 172
  - Reset ADC-FPGA-Logic 174
  - Timestamp Clear 173
  - Trigger 172
  - User function logic 172
- key address 84, 99, 172
- LC 188
- LED 175
  - Application Mode 93
  - U 92
- LED 1 92, 93
- LED 2 92, 93
- LED U 93
- LEDs
  - Front Panel 175
  - SMD 175
- LEMO 187
- LEMO Clock Out 119
- LEMO Trigger Out 120, 122
- LEMO User Out 122
- LVDS Bus connector 36
- LVDS Bus connector schematic 36
- M 152, 161
- MAC address 104
- MAW 149
- Memory Handling 26
- Memory Organization 74
- MGT 24, 25
- MGT link connections 25
- mode
  - double bank 74
  - single bank 74
- mode double bank 74
- MOLEX 187
- moving window 61
- negative signals 154, 156
- Neutron/Gamma application 59
- Neutron/Gamma parameters 59
- NIM 183
- Optical Ethernet** 185
- Option Value 104
- ordering options 187
- P 61, 152, 161
- Peak Sensing ADC 77
- Peaking Time 61
- pileup 68
- Pileup Logic 66
- pileup trigger
  - internal 138
- Pileup Trigger generation 58
- Pileup Window 66
- Power Consumption 186
- PPS 42, 114
  - PulsePerSecond 33, 42
- Prescaler of the sampling clock 42
- Programmable Precision Clock Multiplier 19
- PROM 178, 186
- range 187
- Reboot FPGAs 92
- register
  - Actual Sample address 170
  - Previous Bank Sample address 171
- register
  - Accumulator Gate Configuration 160
  - acquisition control/status 115
  - Active Trigger Gate Window Length 141
  - ADC FPGA Boot control 106
  - ADC FPGA firmware version 168
  - ADC FPGA SPI BUSY Status 126
  - ADC FPGA Status 129, 169
  - ADC Gain and Termination Control 131
  - ADC group 1 89
  - ADC IOB delay 129
  - ADC Offset (DAC) 132, 134
  - ADC SPI control 135
  - ADC SPI Readback) 136
  - Average Configuration 145
  - Average/Decimation Configuration 146
  - broadcast setup 98, 99
  - Channel header ID 139
  - control 168
  - Control/Status 92
  - Data Format Configuration 148
  - description 92
  - End Address Threshold 140
  - Energy Histogram Configuration 163
  - event configuration 137
  - extended event configuration 138
  - Extended Raw Data Buffer Configuration 159
  - External Veto/Gate Delay 109

fir Energy setup 161  
 fir trigger setup 152  
 firmware revision 94  
 FP-Bus control 113  
 FPGA Data Transfer Control 123  
 hardware version. 100  
 high energy trigger threshold 155  
 Interface Access Arbitration Control 97  
 Internal Gate LENGTH Configuration 151  
 Internal Transfer Speed 105  
 Internal Trigger Delay Configuration 150  
 Internal Trigger Feedback Select 122  
 interrupt configuration 95  
 interrupt control 96  
 key address 88  
 LEMO Out 119, 120, 121  
 MAW Start Index and Energy Pickup 166  
 MAW Test Buffer Configuration 149  
 module Id. 94  
 NIM Clock Multiplier 112  
 NIM Input Control/Status 114  
 OneWire EEPROM control 102  
 Peak/Charge Configuration 158  
 Pileup Configuration 143  
 Pre Trigger Delay 144  
 Prescaler Output Pulse Divider 127  
 Prescaler Output Pulse Length 127  
 Programmable Clock I2C 110  
 Raw Data Buffer Configuration 142  
 sample clock distribution control 111  
 Serial Number 104  
 SPI Flash Control/Status 107  
 SPI Flash Data 108  
 tap delay 129  
 temperature 101  
 Test registers 167  
 Trigger Coincidence Lookup Table Address 117  
 Trigger Coincidence Lookup Table Control 117  
 Trigger Coincidence Lookup Table Data 118  
 Trigger Statistic Counter Mode 157  
 trigger threshold 153  
 VME FPGA – ADC FPGA Data Link Status 125  
 Re-Pileup Window 68  
 resolution 63  
 resolution: 63  
 ROAK 95  
 ROOT 189  
 RORA 95  
 RX LOS 175  
 Sample Clock configuration sequence 21  
 Sample Clock distribution 18  
 Sample Control overview 28  
 Sample Event Active 115, 122  
 Sample Logic 59  
 Sample Logic busy 35  
 Sample Logic Busy 115, 122  
 Save Raw data of first Event of Bankbuffer only Enable bit 138  
 serial number 104  
 SFP 25, 188  
   Ethernet 23  
   Optical 24  
   Optical Ethernet 23  
 SFP cage 23  
 SI5325 19, 20, 86, 112  
 SI570 19  
 single bank mode 74  
 SIS1100e 24  
 SIS3316 VME board Overview 16  
 Software 189  
 SW1 83, 185  
 SW2 83, 185  
 SW80 83, 180, 185  
 switches 178  
**SYSRESET** 185  
 tap delay 129  
 tau factor 62  
 TCK 178  
 TDI 178  
 TDO 178  
 Technical Properties/Features 15  
 temperature 101  
 termination 131  
 TI 31, 33, 116  
 timestamp  
   clear 27  
   external clear 32  
 timestamp clear 35  
 TMS 178  
 TO 39, 40, 70, 120, 122  
 trigger 27  
   coincidence lookup table 39  
   counters 39  
   external 137  
   internal 137  
   internal to external 39  
   internally generated 39  
   to LEMO output 39  
 Trigger Coincidence Lookup table 40  
 Trigger Coincidence Lookup Table  
   truth table 40  
 Trigger Coincidence Lookup Table 118  
 Trigger MAW/MW 49  
 Trigger on both edges enable bit 155  
 Trigger Statistic Counter 75  
 Trigger/Gate Logic 45  
 truth table 40  
 TX FAULT 175  
 TYCO 187  
 UI 31, 116  
 UO 39, 40, 70, 71, 121, 156  
 USB 178  
 User Counter 114, 148  
 User Counter logic  
   Header 82  
   Inputs TI/UI 34  
 V2 33  
 VCC 178  
 veto 27  
 VME  
   connector 187  
 VME addressing 83  
 VME base address 185  
 VME FPGA 186  
 VME/CTRL FPGA Application Firmware 27  
 VXS 24, 25  
 W25Q64BV/CV 180  
 watchdog 180, 185  
 XILINX 178