SDE Upgrade PLD Firmware Specifications

David Nitz, Michigan Tech

January 24, 2017

Abstract

This document describes the high level specifications for the PLD firmware for the SDE Upgrade Electronics. In particular, it describes the interface between the custom firmware and the embedded ARM processor as well as the necessary interconnections between the various subsections. This document is a work in progress. The most current version is kept in the "docs" subdirectory of the "auger-prime-sde/uub-firmware" github repository.

1 History

19-Sep-2016 Added description of full bandwidth single bin shower trigger; corrected errors in time tagging register description.

26-Sep-2016 Added base addresses of trigger module, time tagging module, interface module, and memories; added bit definitions for interface module.

09-Nov-2016 Changed maximum SSD or WCD delay in full bandwidth single bin & muon triggers from 3 to 7.

22-Nov-2016 Removed WCD delay in full bandwidth single bin & muon triggers. Increased maximum overlap & consecutive bins allowed. Added MUON_BUF_SIPM_CAL bit to MUON_BUF_TRIG_MASK. If this bit is set, each the 4th ADC value will alternate between the SSD high gain channel & the SIPM calibration channel. Added computation of area, peak, and baseline for showers.

04-Dec-2016 Changed WATCHDOG from an input to an output; update description in the Interface Module section.

12-Dec-2016 Added output enable control for WATCHDOG output.

24-Jan-2017 Updated documentation to consistently number PMTs starting at 0.

2 Modules

The development of the PLD firmware is split into several logically distinct, loosely connected, modules to define a high level structure and enable partitioning of the development effort among various institutions.

The major high level modules are

1. Trigger (including event buffers)
2. Time tagging
3. LED controller
4. Digital interface to external detectors

3 Tools

The Xilinx Vivado development tools are used. The highest level which includes the various interfaces the processor via the AXI bus is built using the Vivado block diagram tools. Lower levels of custom firmware are coded directly in Verilog. This structure allows the relatively clean mating to the git repository.

For storage in git a directory structure recommended by Xilinx is used. The shell script "tag_for_git", located in the base directory scans through a project directory and tag those files that need to be stored in git. Some hand addition or removal of files from git may be required. This helps to avoid temporary files created by Vivado from being stored in git. Please look at the source code for "tag_for_git" to more information about the directory structure it expects. For custom IP in the "ip_repo" directory, the entire tree of the IP should be added to git.

Each WP with firmware has its own master directory in the git repository. For example, front end development work is in "/wp1", the trigger code development work is in "wp2" and time tagging is in "wp3". Archives of snapshots of the integrated Vivado projects are kept in "wp5/sde_pl_hhddmmyy.tar", where hhddmmyy corresponds to the ID of the trigger module. In addition to the Vivado project archive, a copy of this file, the sde_trigger header files, and some example programs are included in the tar file.

4 Interconnections

4.1 Modules to Processor

In previous incarnations of the SDE electronics for Auger and Auger North the processor and the PLD firmware were on separate chips. This meant that the PLD interfaced to the processor bus. In fact the PLD looked just like an external static RAM to the processor. This meant that the PLD needed to decode addresses, control signals and strobes.

This is not necessary anymore with the Zynq chip and the Vivado tools. The Xilinx provided bus interface IP can take care of all of those details.

1. The control register interfaces are implemented via a Vivado AXI register peripheral IP.

2. CPU interrupt signals, if required, are implemented via the same IP.

3. Event storage (& associated information) is implemented as double buffered dual ported memory. This will be discussed further in a later section.

4.2 Trigger and Time Tagging Interconnection

In the original Auger (South) design, the trigger and time-tagging circuits are on separate chips. A somewhat elaborate scheme of bus signals was devised to ensure that the correct time-tag was associated with the correct event. This is simplified in the new design.

In order to correctly associate the event time with the event data, the trigger module send a signal to the time tagging circuit telling it to latch the current time. In both previous designs this is simply the number of clock ticks since the last GPS 1PPS signal. But this is not sufficient. The trigger circuit must also tell the time tagging which of the event buffers the time should be associated with. This implies the time tagging and the trigger need to use the same number of buffers. We have chosen to use 4 buffers.

In the original Auger South design we also encode the trigger signal to be able to measure the dead-time of the circuit. That is, the trigger time (actually the end of trace time) is indicated when an active-low signal (called /EVTCLKF in the currently operating design) is asserted. If another buffer is available at that time, the signal returns to its non-active high state after 100 ns. However, if there is no buffer available, the signal remains low until a buffer becomes available. The time-tagging circuit records both edges. In the upgrade dead-time will be indicated by an active high signal "DEAD" from the trigger to the time-tagging modules. The time-tagging module will keep track of the dead-time fraction.

It is important to note that in addition to the shower event data, single muons also recorded for calibration. When a muon calibration buffer becomes full (containing many muons) a muon trigger interrupt is asserted. This requires each of the signals described above to be duplicated for that trigger.

If additional specialized calibration triggers are developed that require their own buffers, then additional copies of the interconnection signals would follow. (For example, we don't know how to implement an efficient muon decay trigger at present. It is possible this may require something beyond the normal muon buffers.)

The trigger circuit will present to the processor the information detailing which buffer should be read for each event. As long at the processor reads the same buffer from each interface the time tags will remain synchronized with the events.

A corollary to this scheme is that all of the circuits that provide portions of the event data utilize the same "done" signal from the CPU to free up the buffer for a new event. All must use the same trigger signal to close one buffer and move to the next, and that signal should not be set by the processor until the data for an event has been read from all of the interfaces. Presumably the full event readout is done in a single interrupt service routine, so this restriction is not difficult to adhere to.

Signal	Direction	Description
DEAD	Trigger→Time-tagging	Active high during time trigger not possible
SHWR_TRIGGER	Trigger→Time-tagging	Active high pulse at end of triggered trace
SHWR_BUF_NUM[1:0]	Trigger→Time-tagging	Buffer number triggered trace is stored in
MUON_TRIGGER	Trigger→Time-tagging	Active high pulse at end of muon buffer
MUON_BUF_NUM[1:0]	Trigger→Time-tagging	Buffer number muon data stored in

4.2.1 Trigger and LED interface

The trigger circuit needs to be informed when the trigger is initiated by a LED pulse. Operationally we wish to be able to initiate a LED pulse of a given amplitude at a given time. The trigger circuit does not need to know anything about the requested amplitude, but it does need to know that there was a LED flash so that it can add a flag to the trigger bits for the event. This is implemented in the sde_trigger module, based upon the VHDL code provided by Roberto Assiro.

4.3 Trigger and Digital Interface

The trigger circuit needs to send a trigger signal to the digital interface circuit. Note that this is not the same signal as that sent to the time tagging circuit. The signal to the digital interface should presumably be prompt, while that to the time tagging is tied to a fixed bin location within the tank PMT trace. Details are to be defined if/when interfaced detector is specified.

5 Trigger Module

5.1 Register Usage

The register address space of the trigger circuitry occupies 256 control registers and 5 interrupt control registers. The base address of the trigger module is 0x43c2 0000, while that of the trigger interrupt module is 0x43c1 0000. Compatibility with the previous register structure has been maintained where practical. Where this is the case the word "Compatibility" will appear in the title, and the register addresses and bit structure will be the same as in the pre-upgrade electronics. Not (yet) implemented features are indicated by cyan text. The symbolic names and addresses of all the registers are defined in the file "sde_trigger_defs.h", which should be included in any C program that references the registers. (sde_trigger_defs.h is located in the ip_repo/sde_trigger tree.) In addition to register names and addresses, various bit masks and shifts associated with the registers are defined in this header file. Some of the symbolic names and addresses are shown in the included tables. Additional ones can be found in sde_trigger_defs.h. Many of the tables in the following sections include a "Width" field. While all reads and writes transfer a 32 bit word, the "Width" field, if provided, indicates the highest bit of the word read or written that useful. Higher order bits than that return 0 on read, and are ignored on write.

The compatibility mode triggers operate on a signal that has been filtered using a FIR Nyquist filter with a 20 MHz cutoff to approximate the frequency response of the previous electronics. These triggers sample every 3rd bin of the filtered signal. The filtered signals are saved in the shower buffers, in otherwise unused bits, to allow access for debugging purposes. They may be stripped out before sending the data from the local station to the campus. However, the filtered waveform (except for the sampling phase) can be generated by applying a finite impulse response filter to the full bandwidth waveform, as indicated in the code snippet below:

```
int fir[21] = {5,0,12,22,0,-61,-96,0,256,551,681,551,256,0,-96,-61,0,22,12,0,5};
int filt[2048], unfilt[2048];

for (i=21; i<2048; i++)
```

```
{
  filt[i] = 0;
  for (j=0; j<21; j++)
    filt[i] = filt[i] + unfilt[i-21+j]*fir[j];
}
```

The code snippet below indicates how to use the bit masks and shifts defined in sde_trigger_defs.h.

```
int wrt_word, rd_word, value1, value2;

// Prepare information to load multi-field register
wrt_word = (value1 & VALUE1_MASK) << VALUE1_SHIFT;

// Extract information from word read from multi-field register
value1 = (rd_word & VALUE1_MASK) >> VALUE1_SHIFT;
value2 = (rd_word & VALUE2_MASK) >> VALUE2_SHIFT;
```

5.2 Compatibility single bin trigger

Compatibility single bin triggers operate on a single ADC time bin, after down-sampling from 120 to 40 MHz. This trigger is controlled by the registers shown below. The new ADCs have 2 more bits, added at the low end, so the thresholds (above baseline) must be approximately 4 times larger (in ADC counts) than with the previous electronics to replicate the behavior of the original electronics. Note that the signal must be strictly greater than the specified threshold for the trigger to fire. The multiplicity must be greater than or equal to the specified multiplicity.

Address	R/W	Width	Description
COMPATIBILITY_SB_TRIG_THR0_ADDR	R/W	12	Threshold for PMT0
COMPATIBILITY_SB_TRIG_THR1_ADDR	R/W	12	Threshold for PMT1
COMPATIBILITY_SB_TRIG_THR2_ADDR	R/W	12	Threshold for PMT2
COMPATIBILITY_SB_TRIG_ENAB_ADDR	R/W	10	Control bits for the trigger

The bit usage for the COMPATIBILITY_SB_TRIG_ENAB register is as follows:

Bit(s)	Bit Mask (or << Shift)	Description
3	COMPATIBILITY_SB_TRIG_INCL_PMT0	Include PMT0 in multiplicity logic if set
4	COMPATIBILITY_SB_TRIG_INCL_PMT1	Include PMT1 in multiplicity logic if set
5	COMPATIBILITY_SB_TRIG_INCL_PMT2	Include PMT2 in multiplicity logic if set
7:6	<<COMPATIBILITY_SB_TRIG_COINC_LVL_SHIFT	Coincidence level for multiplicity logic sub-trigger
9		Require 2 consecutive bins above threshold if set

5.3 Time-over-threshold trigger

The time-over-threshold trigger first applies trigger conditions of the type used for the single bin trigger to individual bins, then counts the number of bins within a sliding window which meet those conditions. If the occupancy exceeds a specified value, a time-over-threshold shower trigger is generated.
A block of 7 registers controls each time-over-threshold trigger instance.

Name	Address	R/W	Width	Description
THR0	base+0	R/W	10	Threshold for PMT0
THR1	base+1	R/W	10	Threshold for PMT1
THR2	base+2	R/W	10	Threshold for PMT2
	base+3	R/W	12	Not longer used
ENABLE	base+4	R/W	10	Control bits for the trigger
WID	base+5	R/W	7	Width in bins of sliding window (see below!)
OCC	base+6	R/W	7	Occupancy required for trigger

The bit usage for the ENABLE register is the same as for the single bin trigger above.
The sub-trigger that feeds the time-over-threshold occupancy logic is the logical OR of the enabled multiplicity and pulse height sum sub-triggers.

Note: Previously implemented trigger instances B and C are not present in code versions x260210y and later.

Note: The maximum WID and OCC is reduced to 127 starting with code versions x270711y.

Note: WID is fixed at 120 bins starting with code versions x121211y.

5.4 Time-over-threshold trigger (deconvoluted)

The time-over-threshold-deconvoluted trigger first deconvolutes the typical exponential fall from the FADC trace, secondly applies a pulse height window to each deconvoluted trace, and thirdly counts the number of bins within a sliding window which meet pulse height window conditions. Finally, if the occupancy exceeds a specified value in the required multiplicity of deconvoluted traces, a time-over-threshold-deconvoluted shower trigger is generated. Please note that the order of operations is not the same as for the time-over-threshold trigger above.
Also note that as of version x161012y the signal integral condition in the MOPS trigger registers is also applied to the ToTd trigger, and that the same threshold for the integral applies.
A block of 11 registers controls the time-over-threshold-deconvoluted trigger instance.

Name	Address	R/W	Width	Description
THR0	base+0	R/W	10	Lower threshold for PMT0
THR1	base+1	R/W	10	Lower threshold for PMT1
THR2	base+2	R/W	10	Lower threshold for PMT2
UP0	base+3	R/W	10	Upper threshold for PMT0
UP1	base+4	R/W	10	Upper threshold for PMT1
UP2	base+5	R/W	10	Upper threshold for PMT2
ENABLE	base+6	R/W	8	Control bits for the trigger
WID	base+7	R/W	7	Width in bins of sliding window (see below!)
OCC	base+8	R/W	7	Occupancy required for trigger
FD	base+9	R/W	6	Decay constant - binary fraction
FN	base+a	R/W	6	Normalizer - binary fixed point

Note that the THRx and OCC conditions are satisfied when the signal is strictly greater than (>) the register value, while UPx condition is satisfied when the signal is less than or equal to ($\leq$) the register value. Thus writing 1023 to an UPx register disables that constraint.
The bit usage for the ENABLE register is shown below.

Bit(s)	Description
0	Not used
1	Not used
2	Not used
3	Include PMT0 in multiplicity logic if set
4	Include PMT1 in multiplicity logic if set
5	Include PMT2 in multiplicity logic if set
7:6	Coincidence level for multiplicity logic sub-trigger

FD is obtained from $FD = e^{-\Delta t/\tau}$ where $\Delta t = 25ns$, and τ is the decay time of light in the tank. FD is 6 binary digits constrained to be less than 0.75. The value of FN to load is determined by FD from the equation $FN = 1/(1 - FD)$. It is loaded as a fixed point number xx.yyyy, and must be less than 4. Values of FD and FN are within the allowed range for $\tau \leq 80$ ns.

The following logic can be used to obtain the values of FD and FN to load in the registers.

$$FD = (int)\,(64. * e^{-\Delta t/\tau} + 0.5)$$

$$FN = (int)\,(1024./(64. - FD) + 0.5)$$

This trigger is present in code versions x040410y and later.

Note: WID is fixed at 120 bins starting with code versions x121211y.

5.5 Multiplicity of Positive Steps

The multiplicity-of-positive-steps (MOPS) trigger counts positive steps in each FADC trace, accumulating bin to bin steps until the first bin which is smaller than the previous bin is encountered. The aggregate positive step is then subjected to a minimum & maximum height constraint. If it passes this condition the step is added to the count of such steps within a fixed length 120 bin long sliding window. A variable length veto may be applied to ignore aggregate steps following another such step. Finally, if the occupancy exceeds a specified value in the required multiplicity of traces, and the integrated signal exceeds a specified value in those same traces, a multiplicity-of-positive-steps trigger is formed.

The signal integration operates by keeping a running sum of the pulse height in the current ADC time bin minus that in the bin 250 time slots earlier. However, there is a period after a trigger, when no triggers are possible due to finishing the ADC trace or if no buffers are available, in which the earlier time bin is not available in the PLD. While this does not affect any triggered events, it could cause the integrated signal to diverge from zero. In order to mitigate this effect, the integrated signal is exponentially decayed with a time constant of 2048 time bins. The actual integral decays faster than that due to the undershoot following a signal. Note that in any case, the integral depends upon the history of signals not only in the trace, but before the trace, so an exact offline reconstruction of the trigger is not possible. The integral remains valid for ≈250 time bins after the primary trigger.
A block of 10 registers controls the multiplicity-of-positive-steps trigger instance.

Name	Address	R/W	Width	Description
MIN0	base+0	R/W	10	Lower threshold for PMT0
MIN1	base+1	R/W	10	Lower threshold for PMT1
MIN2	base+2	R/W	10	Lower threshold for PMT2
MAX0	base+3	R/W	10	Upper threshold for PMT0
MAX1	base+4	R/W	10	Upper threshold for PMT1
MAX2	base+5	R/W	10	Upper threshold for PMT2
ENABLE	base+6	R/W	8	Control bits for the trigger
INT	base+7	R/W	10	Minimum integrated signal
OCC	base+8	R/W	7	Occupancy required for trigger
OFS	base+9	R/W	4	Veto

Note that the MINx, OCC, and INT conditions are satisfied when the signal is strictly greater than (>) the register value, while MAXx condition is satisfied when the signal is less than or equal to (≤) the register value. Thus writing 1023 to a MAXx register disables that constraint.

The veto is active for $max(0, int(log_2(ph)) + 1 - ofs)$ bins following a aggregate step of size ph. The veto can be disabled by setting OFS to 10 (decimal). Also note that a veto length of 1 has no effect since at least on negative bin to bin step is required to finish an aggregate step. The following table lists some example veto durations.

OFS	Aggregate step	Veto duration
0	2	2
0	3	2
0	9	4
1	2	1
1	5	2
1	8	3
2	8	2
4	8	0

The bit usage for the ENABLE register is shown below.

Bit(s)	Description
0	Not used
1	Not used
2	Not used
3	Include PMT0 in multiplicity logic if set
4	Include PMT1 in multiplicity logic if set
5	Include PMT2 in multiplicity logic if set
7:6	Coincidence level for multiplicity logic sub-trigger

This trigger is present in code versions x270711y and later. The minimum integrated signal requirement is present in code versions x081211y and later. Note this uses the register previously used to specify the window length.

Also note that as of version x161012y the minimum integrated signal requirement set for the MOPS trigger is also applied to the ToTd trigger.

5.6 Random Trigger

The random trigger provides a means for the PPC403 to generate a trigger after a specified delay, irrespective of the ADC inputs. It is controlled by a bank of 3 registers.

Name	Address	R/W	Width	Description
DELAY A	base+0	R/W	12	Low order 12 bits of delay (25ns increments)
DELAY B	base+1	R/W	12	High order 12 bits of delay (25ns increments)
START	base+2	W	–	Writing anything to this register starts the random trigger timer

Only 1 random trigger is generated per write to the START register. Also note that the trigger inhibit will block random triggers with less than 256 clock cycle delay.

5.7 Full bandwidth single bin trigger

The full bandwidth single bin trigger operates on a single bin or small number of consecutive bins. The single bin trigger registers are organized as follows:

Address	R/W	Width	Description
SB_TRIG_THR0_ADDR	R/W	12	Threshold for PMT0
SB_TRIG_THR1_ADDR	R/W	12	Threshold for PMT1
SB_TRIG_THR2_ADDR	R/W	12	Threshold for PMT2
SB_TRIG_SSD_ADDR	R/W	12	Threshold for SSD
SB_TRIG_ENAB_ADDR	R/W	15	Control bits for the trigger

The bit usage for the SB_TRIG_ENAB register is as follows:

Bit(s)	Bit Mask (or Shift)	Description
0	SB_TRIG_INCL_PMT0	Include PMT0 in multiplicity logic if set
1	SB_TRIG_INCL_PMT1	Include PMT1 in multiplicity logic if set
2	SB_TRIG_INCL_PMT2	Include PMT2 in multiplicity logic if set
3	SB_TRIG_INCL_SSD	Include SSD in multiplicity logic if set
6:4	SB_TRIG_COINC_LVL_MASK (_SHIFT)	Coincidence level for multiplicity logic sub-trigger
9:7	SB_TRIG_SSD_DELAY_SHIFT	Delay of SSD signals prior to coincidence
12:10	SB_TRIG_COINC_OVLP_SHIFT	Coincidence overlap width increase (bins)
15:13	SB_TRIG_CONSEC_BINS_SHIFT	# of consecutive bins required above threshold

A signal amplitude must be strictly greater than its respective threshold for the trigger to fire. The multiplicity must be greater than or equal to the specified multiplicity. The window for the multiplicity logic coincidence can be set from 1 time bin (8.33 ns) up to 4 time bins. Loading 0 into the SB_TRIG_COINC_OVLP bits sets the overlap to 1 bin, loading 1, sets it to 2 bins, etc. Either the WCD PMT or the SSD signal may be digitally delayed prior to forming a coincidence. Setting the SB_TRIG_WCD_DELAY (or SB_TRIG_SSD_DELAY) bits to 0 adds no delay, setting the bits to 1 adds 1 time bin delay, etc. Additionally, a requirement for more than one consecutive bin for each of the PMT, SSD signals to be above threshold can be added. Setting SB_TRIG_CONSEC_BINS bits to 0 requires just 1 bin to be above threshold, setting it to 1, requires 2 bins to be above threshold, etc.

5.8 Shower memory

This bank of registers controls aspects of the shower memories and provides status information. Currently there are no special compatibility shower memory buffers. The buffers described here are the full bandwidth buffers.

Address	R/W	Width	Description
SHWR_BUF_TRIG_MASK_ADDR	R/W	17	Mask of allowed triggers
SHWR_BUF_TRIG_ID_ADDR	R		Indicates the trigger for the buffer being read
SHWR_BUF_CONTROL_ADDR	W	2	Resets buffer #N when written
SHWR_BUF_STATUS_ADDR	R		Reports shower memory buffer status
SHWR_BUF_START_ADDR	R		Word offset to start of trace in buffer being read

The following bits are defined in the SHWR_BUF_TRIG_MASK register. Note that the external trigger is enabled by default after a reset. If SHWR_BUF_TRIG_LED is set in the SHWR_BUF_TRIG_MASK register, then a trigger is generated on a LED flash, even if other trigger conditions are not met. Whether or not this bit is set in the SHWR_BUF_TRIG_MASK register, the presence of a LED flash is indicated in the SHWR_BUF_TRIG_ID register.

Bit(s)	Bit Mask	Description
0	COMPATIBILITY_SHWR_BUF_TRIG_SB	Compatibility single bin trigger
1	COMPATIBILITY_SHWR_BUF_TRIG_TOT	Time-over-threshold trigger
2	COMPATIBILITY_SHWR_BUF_TRIG_TOTD	Time-over-threshold deconvoluted trigger
3	COMPATIBILITY_SHWR_BUF_TRIG_MOPS	Multiplicity-of-positive-steps trigger
4	COMPATIBILITY_SHWR_BUF_TRIG_EXT	External trigger
5	COMPATIBILITY_SHWR_BUF_TRIG_RNDM	Random trigger
6		Pre-scale compatibility single bin trigger by 256 if set
7		Pre-scale time-over-threshold A trigger if set
8		Pre-scale time-over-threshold deconvoluted trigger if set
9		Pre-scale MoPS trigger if set
10		Pre-scale external trigger by 256 if set
11		Pre-scale random trigger if set
16	SHWR_BUF_TRIG_LED	Trigger on LED flash
17	SB_TRIG	Full bandwidth single bin trigger

The following bits are defined in the SHWR_BUF_TRIG_ID register. The initial trigger bits indicate which trigger generated the T1 trigger from the logic. The additional trigger bits flag any trigger conditions that are satisfied by the FADC traces after an initial trigger.

Bit(s)	Bit Mask	Description
0	COMPATIBILITY_SHWR_BUF_TRIG_SB	Initial single bin trigger
1	COMPATIBILITY_SHWR_BUF_TRIG_TOT	Initial time-over-threshold A trigger
2	COMPATIBILITY_SHWR_BUF_TRIG_TOTD	Initial time-over-threshold deconvoluted trigger
3	COMPATIBILITY_SHWR_BUF_TRIG_MOPS	Multiplicity-of-positive-steps trigger
4	COMPATIBILITY_SHWR_BUF_TRIG_EXT	Initial external trigger
5	COMPATIBILITY_SHWR_BUF_TRIG_RNDM	Initial random trigger
8	COMPATIBILITY_SHWR_BUF_TRIG_SB_DLYD	Additional single bin trigger
9	COMPATIBILITY_SHWR_BUF_TRIG_TOT_DLYD	Additional time-over-threshold A trigger
10	COMPATIBILITY_SHWR_BUF_TRIG_TOTD_DLYD	Additional time-over-threshold deconvoluted trigger
11	COMPATIBILITY_SHWR_BUF_TRIG_MOPS_DLYD	Additional multiplicity-of-positive-steps trigger
12	COMPATIBILITY_SHWR_BUF_TRIG_EXT_DLYD	Additional external trigger
13	COMPATIBILITY_SHWR_BUF_TRIG_RNDM_DLYD	Additional random trigger
16	SHWR_BUF_TRIG_LED	LED flasher fired
24	SHWR_BUF_TRIG_LED_DLYD	LED flasher fired after trigger
25	SB_TRIG_DLYD	Additional single bin trigger

The following bits are defined in the SHWR_BUF_CONTROL register.

Bit(s)	Description
1-0	Reset buffer full status of specified buffer

The following bits are defined in the SHWR_BUF_STATUS register.

Bit(s)	Bit Mask (or Shift)	Description
1:0	SHWR_BUF_RNUM_MASK (_SHIFT)	Number of the shower buffer to be read
3:2	SHWR_BUF_WNUM_MASK (_SHIFT)	Number of buffer currently being written
7:4	SHWR_BUF_FULL_MASK (_SHIFT)	Bit map of full buffers
8	SHWR_INTR_PEND_MASK (_SHIFT)	Interrupt pending if set
10:9	SHWR_BUF_NFULL_MASK (_SHIFT)	Number of full buffers

5.9 Shower Features

The shower features block contains the area, peak, and baseline computed by the FPGA for each shower. There are 10 SHWR_PEAK_AREAx (x ranges from 0 to 9) registers and 5 SHWR_BASELINEy (y ranges from 0 to 4) registers. The area is computed for SHWR_AREA_BINS after a trigger. The same region is used to compute the peak and check for saturated bins. The contents of each SHWR_PEAK_AREAx register is shown below.

Bit(s)	Bit Mask (or Shift)	Description
18:0	SHWR_AREA_MASK	Shower area above the baseline
30:19	SHWR_PEAK_MASK (_SHIFT)	Peak signal above the baseline
31	SHWR_SATURATED	Set if any bin is saturated within SHWR_AREA_BINS of trigger

The baseline is calculated in the region before the trigger. This is the value reported in the SHWR_BASELINEy registers. For the computation of the peak and integral, however, the RC undershoot is accounted for. The parameters of the undershoot calculation will need to be updated if the value of the series input capacitor on the UUB is changed. The contents of each SHWR_BASELINEy register is shown below.

Bit(s)	Description
3:0	Fractional bits low gain channel baseline
15:4	Baseline for low gain channel
19:16	Fractional bits high gain channel baseline
31:20	Baseline for high gain channel

SHWR_BASELINE0 through SHWR_BASELINE2 contain the baselines for the 3 WCD PMTs, SHWR_BASELINE3 contains the baselines for the small PMT (low gain part) and SIPM calibration (high gain part). SHWR_BASELINE4 contains the baselines for the SSD sensor.

5.10 Trigger Rates

The trigger rates block counts reports the rates of several of the triggers accumulated over the last 26.8 seconds. The rate counters are double buffered, so there is always a static value representing the number of triggers during a 26.8 second interval that ended some time between 0 and 26.8 seconds ago. The trigger rate registers are organized as follows:

Name	Address	R/W	Width	Description
TRIGGER RATES	base+0	R	32	Rates for single bin, ToT, ToTd, and MoPS triggers
DELAYED RATES	base+1	R	32	Rates for single bin, ToT, ToTd, and MoPS delayed triggers

The information is packed in each of these two registers as follows:

Bit(s)	Description
7:0	# single bin triggers in last 26.8 seconds / 32
15:8	# ToT triggers in last 26.8 seconds
23:16	# ToTd triggers in last 26.8 seconds
31:24	# MoPS triggers in last 26.8 seconds

The counts in each field are limited to a maximum value of 252 (fc_{16}). Any rate higher than 252 per 26.8 second interval will read as 252. For example, any ToTd rate higher than about 9.38 Hz will read 252.

5.11 Muon triggers

Multiple muon triggers are implemented to allow various combinations of triggers for WCD & SSD calibration. The muon trigger registers are organized as follows:

Address	R/W	Width	Description
MUON_TRIGx_THR0_ADDR	R/W	12	Threshold for PMT0
MUON_TRIGx_THR1_ADDR	R/W	12	Threshold for PMT1
MUON_TRIGx_THR2_ADDR	R/W	12	Threshold for PMT2
MUON_TRIGx_SSD_ADDR	R/W	12	Threshold for SSD
MUON_TRIGx_ENAB_ADDR	R/W	15	Control bits for the trigger

The bit usage for the MUON_TRIGx_ENAB register is as follows:

Bit(s)	Bit Mask (or Shift)	Description
0	MUON_TRIG_INCL_PMT0	Include PMT0 in multiplicity logic if set
1	MUON_TRIG_INCL_PMT1	Include PMT1 in multiplicity logic if set
2	MUON_TRIG_INCL_PMT2	Include PMT2 in multiplicity logic if set
3	MUON_TRIG_INCL_SSD	Include SSD in multiplicity logic if set
6:4	MUON_TRIG_COINC_LVL_MASK (_SHIFT)	Coincidence level for multiplicity logic sub-trigger
7-9	MUON_TRIG_SSD_DELAY_SHIFT	Delay of SSD signals prior to coincidence
12:10	MUON_TRIG_COINC_OVLP_SHIFT	Coincidence overlap width increase (bins)
15:13	MUON_TRIG_CONSEC_BINS_SHIFT	# of consecutive bins required above threshold

Here "x" is either 1, 2, 3, or 4, corresponding to which of the 4 simple triggers is desired. A signal amplitude must be strictly greater than its respective threshold for the trigger to fire. The multiplicity must be greater than or equal to the specified multiplicity. The window for the multiplicity logic coincidence can be set from 1 time bin (8.33 ns) up to 4 time bins. Loading 0 into the MUON_TRIG_COINC_OVLP bits sets the overlap to 1 bin, loading 1, sets it to 2 bins, etc. Either the WCD PMT or the SSD signal may be digitally delayed prior to forming a coincidence. Setting the MUON_TRIG_WCD_DELAY (or MUON_TRIG_SSD_DELAY) bits to 0 adds no delay, setting the bits to 1 adds 1 time bin delay, etc. Additionally, a requirement for more than one consecutive bin for each of the PMT, SSD signals to be above threshold can be added. Setting MUON_TRIG_CONSEC_BINS bits to 0 requires just 1 bin to be above threshold, setting it to 1, requires 2 bins to be above threshold, etc.

5.12 Muon memory buffers

This bank of registers controls aspects of the muon memory buffers and provides status information.

Address	R/W	Width	Description
MUON_BUF_TIME_TAG_A_ADDR	R	30	Time tag at the beginning of the buffer
MUON_BUF_TIME_TAG_B_ADDR	R	30	Time tag at the end of the buffer
MUON_BUF_TRIG_MASK_ADDR	R/W	1	Enable mask of allowed triggers
MUON_BUF_CONTROL_ADDR	W	2	Resets buffer #N when written
MUON_BUF_STATUS_ADDR	R	11	Reports muon memory buffer status
MUON_BUF_WORD_COUNT_ADDR	R	13	Reports number of words in buffer

The following bits are defined in the MUON_BUF_TRIG_MASK register.

Bit	Bit Mask (or Shift)	Description
0	MUON_BUF_TRIG_SB1	Enable muon simple threshold 1 trigger if set
1	MUON_BUF_TRIG_SB2	Enable muon simple threshold 2 trigger if set
2	MUON_BUF_TRIG_SB3	Enable muon simple threshold 3 trigger if set
3	MUON_BUF_TRIG_SB4	Enable muon simple threshold 4 trigger if set
4	MUON_BUF_TRIG_EXT	Enable muon external trigger if set
5	MUON_BUF_SIPM_CAL	Enable mode where SIPM cal. & high gain alternate

The following bits are defined in the CONTROL register.

Bit(s)	Description
1:0	Reset buffer full status of specified buffer

The following bits are defined in the MUON_BUF_STATUS register.

Bit(s)	Bit Mask or Shift	Description
1:0	MUON_BUF_RNUM_MASK (_SHIFT)	Number of the muon buffer to be read
3:2	MUON_BUF_WNUM_MASK (_SHIFT)	Number of buffer currently being written
6:3	MUON_BUF_FULL_MASK (_SHIFT)	Bit map of full buffers
7	MUON_INTR_PEND_MASK (_SHIFT)	Interrupt pending if set
10:8	MUON_BUF_NFULL_MASK (_SHIFT)	Number of full buffers

5.13 LED pulses

LED pulses may be generated under program control through the trigger module. (Ported original code developed by Roberto Assiro, Lecce.) Timing of the pulses is controlled by the trigger module. Amplitude of the pulses is controlled by the slow control micro-controller. There is one LED_CONTROL register at address LED_CONTROL_ADDR. The bit usage for this register is as follows:

Bit(s)	Bit Mask or Shift	Description
0	LED_NOW	Do LED pulse now
1	LED_ENAPPS	Enable LED pulse after PPS if set
18:2	LED_DELAY_MASK (_SHIFT)	# time bins delay after PPS
29:19	LED_PULSWID_MASK (_SHIFT)	Width of LED pulse in time bins

Note that if LED_ENAPPS is set, that LED pulses will continue to be generated at the specified time after the PPS. This will continue until the LED_ENAPPS bit is cleared. If the timing of the LED pulse is not critical, LED_NOW can be used to generate a single pulse. LED_NOW must be cleared before another single pulse can be generated.

5.14 Scalers

There are 3 32-bit scaler instances. Each scaler instance accumulates counts of s single bin trigger instance. The count can be reset by writing (anything) to the scaler register. Unlike code versions prior to x010105y, these scalers are no longer tied to muon buffer intervals and they are not double buffered. Scaler incrementing is inhibited while scaler is being read. Each scaler block has the following format.

Name	Address	R/W	Width	Description
Trigger Block	base+0:base+4	R/W	Varies	Block defining dedicated trigger
SOURCE MASK	base+5	R/W	7	Mask of sources for this scaler instance - no longer used
HOLDOFF	base+6	R/W	10	Mininimum number of bins between triggers - no longer used
SCALER	base+7	R/W	32	Scaler value

The trigger block has the same format as the single bin trigger block described above. The scalers count on the leading edge of a trigger satisfied condition. After counting, the scalers will not count again until after the input data has fallen below the trigger threshold.

5.15 AMIGA trigger

This trigger block produces the interface signals for the AMIGA trigger. Only one register is used.

Name	Address	R/W	Width	Description
AMIGA GPS STAMP	15	W	24	GPS timestamp at trigger

5.16 Other

The following registers don't fit into any of the groupings above.

Address	R/W	Width	Description
ID_REG_ADDR	R	32	ID register
GLOBAL_CONTROL_ADDR	W	1	Some global control operations

The ID registers contain the following bits.

Bits	Description
31:0	8 Hex digits: "xhhddmmyy", where: "hhddmmyy" represents the compile date (hour/day/month/year) of the code version release.

The GLOBAL CONTROL register contains the following bits.

Bit(s)	Description
0	Generate a reset of the trigger logic

5.17 Address Map

Both the address map and field definitions for the trigger module are defined in the file "sde_trigger_defs.h", the master copy of which is kept in the "sde_trigger" subdirectory of the "ip_repo" directory in the auger-prime-sde/uub-firmware git repository. A version corresponding to the integrated project snapshot is included in the wp5/sde_pl_hhddmmyy.tar file.

6 Shower Triggers & Memory

6.1 Register write order

To avoid spurious triggers, normally all registers should be loaded with their desired values before enabling any triggers in the SHWR_BUF_TRIG_MASK or MUON_BUF_TRIG_MASK registers.

6.2 Shower memory organization

After some consideration and testing, the organization of the ADC data in the shower memory buffers was chosen to simplify unpacking. Five memory blocks, containing 4 buffers each are used to store the traces. The format of each data word in TRIGGER_MEMORY_SHWR0 is shown below.

Bits	Description
11:0	WCD PMT0 low gain ADC
15:12	Low order 4 bits of filtered PMT0
27:16	WCD PMT0 high gain ADC
31:28	Middle 4 bits of filtered PMT0

The format of each data word in TRIGGER_MEMORY_SHWR1 is shown below.

Bits	Description
11:0	WCD PMT1 low gain ADC
15:12	High order 4 bits of filtered PMT0
27:16	WCD PMT1 high gain ADC
31:28	Low order 4 bits of filtered PMT1

The format of each data word in TRIGGER_MEMORY_SHWR2 is shown below.

Bits	Description
11:0	WCD PMT2 low gain ADC
15:12	Middle 4 bits of filtered PMT1
27:16	WCD PMT2 high gain ADC
31:28	High order 4 bits of filtered PMT1

The format of each data word in TRIGGER_MEMORY_SHWR3 is shown below.

Bits	Description
11:0	Low gain PMT ADC
15:12	Low order 4 bits of filtered PMT2
27:16	Spare (or SIPM calibration) ADC
31:28	Middle 4 bits of filtered PMT2

The format of each data word in TRIGGER_MEMORY_SHWR4 is shown below.

Bits	Description
11:0	SSD low gain ADC
15:12	High order 4 bits of filtered PMT2
27:16	SSD high gain ADC
31:28	Flags; used for debugging

The address map for the shower memories is shown below:

Description	Base Address
TRIGGER_MEMORY_SHWR0	0x4400 0000
TRIGGER_MEMORY_SHWR1	0x4600 0000
TRIGGER_MEMORY_SHWR2	0x4800 0000
TRIGGER_MEMORY_SHWR3	0x4a00 0000
TRIGGER_MEMORY_SHWR4	0x4c00 0000

6.3 Muon memory organization

The muon memories contain time stamped short blocks of data, intended mainly for single particle calibration data. The format of the first entry of TRIGGER_MEMORY_MUON0 is shown below.

Bits	Description
30:0	Time Tag for beginning of burst
31	1, Indicating this is a time tag

The format of the first entry of TRIGGER_MEMORY_MUON1 is shown below.

Bits	Description
5:0	Muon triggers and SIPM calibration flag for this burst
31	1, Indicating this is a trigger tag

The format of subsequent words in each block in TRIGGER_MEMORY_MUON0 is shown below.

Bits	Description
11:0	PMT0 high gain ADC
15:12	Flags; current used for testing
27:16	PMT1 high gain ADC
30:28	Flags; currently used for testing
31	0, Indicating this is ADC data

Similarly, the format of subsequent words in TRIGGER_MEMORY_MUON1 is shown below.

Bits	Description
11:0	PMT2 high gain ADC
15:12	Low order 4 bits of burst counter
27:16	SSD high gain ADC
30:28	High order 3 bits of burst counter
31	0, Indicating this is ADC data

The address map for the muon memories is shown below:

Description	Base Address
TRIGGER_MEMORY_MUON0	0x4000 0000
TRIGGER_MEMORY_MUON1	0x4200 0000

6.4 Reading Shower & Muon Memories

The shower and muon memories can be read via simple DMA, scatter-gather DMA, or via PDT.

Please see "trigger_test.c", included in the sde_pld_hhddmmyy.tar file for examples of the readout sequence for the shower and muon memories.

7 Time Tagging Module

The file "Time_Tagging_Module_Specification_SDE_Upgrade_Version_3.pdf" is the primary documentation for the time tagging module. However, symbolic register addresses are also defined in the "sde_trigger_defs.h" header file for the convenience of writing code that references both the trigger module and the time tagging module. As for the trigger module registers, the symbolic names of the addresses all end in "_ADDR". The following register names are defined:

Register	Description
TTAG_SHWR_TICS	120 Mhz counter at shower trigger
TTAG_SHWR_SECONDS	Seconds counter at shower trigger
TTAG_SHWR_PPS_TICS	120 Mhz counter at last PPS before shower
TTAG_SHWR_PPS_CAL	120 Mhz calibration counter at last PPS before shower
TTAG_MUON_TICS	120 Mhz counter at muon buffer trigger
TTAG_MUON_SECONDS	Seconds counter at muon buffer trigger
TTAG_MUON_PPS_TICS	120 Mhz counter at last PPS before muon buf.
TTAG_MUON_PPS_CAL	120 Mhz calibration counter at last PPS before muon buf.
TTAG_PPS_SECONDS	Seconds counter at last PPS
TTAG_PPS_TICS	120 Mhz counter at last PPS
TTAG_PPS_CAL	120 MHz calibration counter at last PPS
TTAG_PPS_DEAD_CTR	Dead time counter at last PPS
TTAG_STATUS	Status register
TTAG_CTRL	Control register
TTAG_ID	ID register, returns "ttag"

The *TICS registers contain the 120 Mhz counter value in the low order 27 bits (TTAG_TICS_MASK). Bit 27 is always 0. In the TTAG_SHWR_TICS and TTAG_MUON_TICS, bits 31:28 contain the value of the event counter for the shower and muon buffer triggers respectively (TTAG_EVTCTR_SHIFT & TTAG_EVTCTR_MASK). The *SECONDS registers contain the seconds counter in the low order 28 bits (TTAG_SECONDS_MASK). The high order 4 bits are zero.

The TTAG_STATUS register contains bits useful for testing the time tagging module. Bits which get set in this register in response to stimuli can be reset via the TTAG_CONTROL register. The bit assignments in the TTAG_STATUS register are shown below.

Bit	Bit	Description
0	TTAG_MUON_TRIG	Muon buffer trigger occured
1	TTAG_PPS	PPS occurred
2	TTAG_SHWR_TRIG	Shower trigger occurred
3	TTAG_DEAD_TIME	Dead time occurred

The bit assignments in the TTAG_CONTROL register are shown below.

Bit	Bit Mask	Description
0	TTAG_RESET	Reset the time tagging module when 1
1	TTAG_CLR_PPS	Clear the PPS occurred flag
2	TTAG_CLR_SHWR_TRIG	Clear the shower trigger occurred flag
3	TTAG_CLR_MUON_TRIG	Clear the muon buffer trigger occurred flag
4	TTAG_CLR_DEAD_TIME	Clear the dead time occurred flag

The base address of the time tagging module is 0x43c3 0000.

8 Digital Trigger

Left blank for now. Open for future expansion.

9 Other Modules of Interest

9.1 Interface module

Some miscellaneous and test functions are grouped together in an interface module with base address 0x43c0 0000. This module has 4 registers:

Register	Description
0	Bits 7:0 - board switches
	Bit 8 - USB_IFAULT
	Bit 9 - not used
	Bit 10 - Radio reset in
1	Bit 0 - Radio reset out
2	Bit 0 - Generate fake GPS PPS if 1
	Bit 1 - Generate fake data if 1
	Bit 2 - Generate fake muon data if 1
3	Bit 0 - WATCHDOG output value
	Bit 1 - Enable WATCHDOG output

9.2 AXI DMA controller

The AXI DMA controller for the shower & muon memories is located at base address 0x7e20 0000.