SDE Upgrade PLD Firmware Specifications

David Nitz, Michigan Tech

October 30, 2020

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.

11-Mar-2017 Add mode to SB_TRIG where SSD trigger is "anded" with WCD trigger; tune parameters of baseline tracking & integral computations; correct error in MUON_BUF_STATUS register bit offsets; added SHWR_EVT_CTR to SHWR_BUF_STATUS register.

20-Mar-2018 Update documentation to include some information about the trigger interrupt module; add documentation of the ADC_PWD output; add documentation of the test control module; update documentation of the interface module.

27-Apr-2018 Update documentation of the test control module to include additional fake signal features.

29-Apr-2018 Update documentation to describe split of shower and muon interrupts and DMA.

16-May-2018 Update documentation to include compatibility time-over-threshold trigger.

25-May-2018 Update fake data description to include 2 additional random timing modes and a sawtooth ramp mode; clarify 40 MHz clock use for compatibility triggers; reduce number of muon triggers from 4 to 2 to aid in meeting timing constraints.

15-Jun-2018 Add more options to the fake data module.

16-Aug-2018 Update documentation to include compatibility time-over-threshold-deconvolved (ToTd) trigger.

20-Sep-2018 Update fake data module documentation.

24-Sep-2018 Expand interrupt module documentation.

23-Feb-2019 Add documentation of RD interface; update Test Control documentation; add documentation of 1pps interrupt module.

26-Feb-2019 DMA removed to free up resources for routing.

03-Mar-2019 Add RD interface ID register description.

01-Apr-2019 Add digital_interface module to facilitate factory test

29-Apr-2019 Implement scalers.

29-May-2019 Merge in Sjoerd's updates of RD data format, correct typo in RD interface status register; add trigger output disable.

11-Jul-2019 Update RD registers to include software reset functionality.

26-Aug-2019 Add ability to manually set trigger output high for the factory test.

29-Aug-2019 Modify to have digital interface module power up in factory test mode with all LVDS drivers set to input mode.

03-Sep-2019 Correct error in documentation for digital interface. The order of EXT0 and EXT1 in the registers was previously incorrectly documented.

25-Oct-2019 Enable documentation for the compatibility mode MOPS trigger in preparation for inclusion of this trigger.

08-Nov-2019 Add new random trigger documentation.

09-Dec-2019 Add register to report time from the trigger interrupt in order to measure the DAQ/Linux latency in servicing the interrupt; add $5\mu s$ time after transfer of RD data before a new trigger can be sent to the RD; add flag if RD transfer times out.

27-Sep-2020 Updated RD module documentation to clarify readout; updated buffer memories addresses.

30-Oct-2020 Added documentation of where the down sampling phase is stored and how to use it.

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. Test Control
4. Radio Detector Interface
5. AMIGA interface

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 Interfaces

The trigger circuit needs to send a trigger signal to the digital interface circuit and in some cases buffer control hand shaking. Note the trigger signal is not the same signal as that sent to the time tagging circuit. The signal to the digital interfaces are prompt, while that to the time tagging is tied to a fixed bin location within the tank PMT trace. Currently 2 interface modules are implemented: 1) AMIGA interface and 2) Radio Detector interface.

5 Trigger Module

5.1 Register Usage

The register address space of the trigger circuitry occupies 256 control registers and 10 interrupt control registers. The base address of the trigger module is SDE_TRIGGER_BASE = 0x43c2 0000, while that of the shower trigger interrupt module is SDE_SHWR_TRIGGER_INTR_BASE = 0x43c1 0000 and the muon trigger interrupt module is at SDE_MUON_TRIGGER_INTR_BASE = 0x43c5 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. The more complex of these triggers (eg. all but the compatibility single bin trigger) 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 (see Sec. 6.2 for where the sampling phase is stored) 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 Compatibility Time-over-threshold trigger

The compatibility time-over-threshold trigger first applies trigger conditions of the type used for the compatibility 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. This trigger operates on the 40 MHz down-sampled traces.
A block of 7 registers controls the compatibility time-over-threshold trigger instance.

Name	R/W	Width	Description
COMPATIBILITY_TOT_TRIG_THR0_ADDR	R/W	12	Threshold for PMT0
COMPATIBILITY_TOT_TRIG_THR1_ADDR	R/W	12	Threshold for PMT1
COMPATIBILITY_TOT_TRIG_THR2_ADDR	R/W	12	Threshold for PMT2
COMPATIBILITY_TOT_TRIG_ENABLE_ADDR	R/W	10	Control bits for the trigger
COMPATIBILITY_TOT_TRIG_OCC_ADDR	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. The width of the sliding window is fixed at $3\mu s$, 120 (40 MHz) bins.

5.4 Time-over-threshold trigger (deconvolved)

The compatibility time-over-threshold-deconvolved 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 3μs sliding window which meet pulse height window conditions. Finally, if the occupancy exceeds a specified value in the required multiplicity of deconvolved traces, a time-over-threshold-deconvolved shower trigger is generated. Please note that the order of operations is not the same as for the compatibility time-over-threshold trigger above.
A block of 11 registers controls the time-over-threshold-deconvoluted trigger instance.

Name	R/W	Width	Description
COMPATIBILITY_TOTD_TRIG_THR0_ADDR	R/W	12	Lower threshold for PMT0
COMPATIBILITY_TOTD_TRIG_THR1_ADDR	R/W	12	Lower threshold for PMT1
COMPATIBILITY_TOTD_TRIG_THR2_ADDR	R/W	12	Lower threshold for PMT2
COMPATIBILITY_TOTD_TRIG_UP0_ADDR	R/W	12	Upper threshold for PMT0
COMPATIBILITY_TOTD_TRIG_UP1_ADDR	R/W	12	Upper threshold for PMT1
COMPATIBILITY_TOTD_TRIG_UP2_ADDR	R/W	12	Upper threshold for PMT2
COMPATIBILITY_TOTD_TRIG_ENABLE_ADDR	R/W	8	Control bits for the trigger
COMPATIBILITY_TOTD_TRIG_OCC_ADDR	R/W	7	Occupancy required for trigger
COMPATIBILITY_TOTD_TRIG_FD_ADDR	R/W	6	Decay constant - binary fraction
COMPATIBILITY_TOTD_TRIG_FN_ADDR	R/W	6	Normalizer - binary fixed point
COMPATIBILITY_TOTD_TRIG_INT_ADDR	R/W	24	Integral of the signal for each PMT

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 4095 to an UPx register disables that constraint. Note that the integral is the sum of 25 ns (not 8.3 ns!) bins, referenced to the signal 6μs previous. In addition, the integral is decayed to 0 with a $\approx 50\mu$s time constant to ensure recovery from any overflow. Similarly, the OCC refers to the count of 25 ns (not 8.3 ns!) bins.

With respect to the pre-upgrade electronics, the OCC, FD, FN register values remain the same, while the value loaded into the INT register should be increased by a factor of 4.
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.

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

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

5.5 Compatibility Multiplicity of Positive Steps

The compatibility 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 $3\mu s$ 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 $\approx 6\mu s$ (250 25 ns bins) earlier. In addition, the integral is decayed to 0 with a $\approx 50\mu$s time constant (2048 25 ns bins) to ensure recovery from any overflow. 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.
A block of 10 registers controls the multiplicity-of-positive-steps trigger instance.

Name	R/W	Width	Description
COMPATIBILITY_MOPS_MIN0_ADDR	R/W	12	Lower threshold for PMT0
COMPATIBILITY_MOPS_MIN1_ADDR	R/W	12	Lower threshold for PMT1
COMPATIBILITY_MOPS_MIN2_ADDR	R/W	12	Lower threshold for PMT2
COMPATIBILITY_MOPS_MAX0_ADDR	R/W	12	Upper threshold for PMT0
COMPATIBILITY_MOPS_MAX1_ADDR	R/W	12	Upper threshold for PMT1
COMPATIBILITY_MOPS_MAX2_ADDR	R/W	12	Upper threshold for PMT2
COMPATIBILITY_MOPS_ENABLE_ADDR	R/W	8	Control bits for the trigger
COMPATIBILITY_MOPS_INT_ADDR	R/W	24	Minimum integrated signal
COMPATIBILITY_MOPS_OCC_ADDR	R/W	7	Occupancy required for trigger
COMPATIBILITY_MOPS_OFS_ADDR	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 ($\leq$) the register value. Thus writing 4095 to a MAXx register disables that constraint.

The veto is active for $max(0, int(log_2(ph)) - 1 - ofs)$ (25, not 8.3 ns!) bins following a aggregate step of size ph. The veto can be disabled by setting OFS to 12 (decimal). Also note that a veto length of ≤ 1

has no effect since at least one negative bin to bin step is required to finish an aggregate step. With respect to the pre-upgrade electronics, the OCC and OFS register values remain the same, while the values loaded into the INT register and the MINx and MAXx registers should nominally be increased by a factor of 4 to preserve the trigger behavior. The following table lists the veto duration for some example OFS and step sizes.

OFS	Aggregate step	Veto duration
0	8	2
0	12	2
0	36	4
1	8	1
1	20	2
1	32	3
2	32	2
4	32	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

Th initial porting of this module from the UB to the UUB was performed by Fabio Convenga.

5.6 Random Trigger

The functionality of the old UB random trigger is now essentially available in the LED module. The new random trigger module for the UUB, which is not compatible with the old module, is described here. It can be useful for recording background traces or testing DAQ programs with various trigger rates. There is just one register controlling this module at address RANDOM_TRIG_MODE_ADDR. Once started the random trigger continues until 0 is written to the mode register. When changing the mode register one should first write a 0 to the register to avoid a spurious immediate random trigger. The contents of this register are described below:

Value	Fake Data
0	Reset and stop random trigger
1	Fixed rate: 10 ms period
2	Fixed rate: 100 ms period
3	Fixed rate: 1 s period
4	Fixed rate: 10 s period
5	Fixed rate: 100 s period
6	Fixed rate: 200 s period
7	Fixed rate: 400 s period
11	Random timing: $0 - 171$ μs spacing
15	Random timing: $0 - 2.73$ ms spacing
18	Random timing: $0 - 22$ ms spacing
21	Random timing: $0 - 175$ ms spacing
22	Random timing: $0 - 350$ ms spacing
23	Random timing: $0 - 700$ ms spacing
25	Random timing: $0 - 2.8$ s spacing
28	Random timing: $0 - 22.4$ s spacing
31	Random timing: $0 - 179$ s spacing
Anything else ≤ 31	Fixed rate: 800 s period

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 trigger 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
16	SB_TRIG_SSD_AND_SHIFT	AND SSD & WCD triggers if set

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. If the SB_TRIG_SSD_AND bit is not set (default), the SSD is included in the multiplicity logic just like the 3 WCD PMTs. On the other hand if the SB_TRIG_SSD_AND bit is set, the multiplicity logic is applied only to the WCD PMTs, and the result of that logic is "anded" with the SSD above threshold condition.

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
SHWR_BUF_LATENCY_ADDR	R		Time in μs since trigger interrupt asserted for this buffer

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 deconvolved trigger
3	COMPATIBILITY_SHWR_BUF_TRIG_MOPS	Multiplicity-of-positive-steps trigger
4	COMPATIBILITY_SHWR_BUF_TRIG_EXT	External trigger
5	SHWR_BUF_TRIG_RNDM	Random trigger
6		Pre-scale compatibility single bin trigger by 256 if set
7		Pre-scale time-over-threshold trigger if set
8		Pre-scale time-over-threshold deconvolved 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
25		Pre-scale single bin trigger by 256 if set

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
17	SB_TRIG	Initial full bandwidth SB trigger
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
31:16	SHWR_BUF_EVT_ID_MASK(_SHIFT)	Shower event ID

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 or 2, 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
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
7:4	MUON_BUF_FULL_MASK (_SHIFT)	Bit map of full buffers
8	MUON_INTR_PEND_MASK (_SHIFT)	Interrupt pending if set
11:9	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 DAQ program. 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 when the signal meets specified threshold conditions. These are similar to the those of the Compatibility Single Bin trigger. Each scaler block has the following format, where "x" in the address is either "A", "B", or "C":

Address	R/W	Width	Description
COMPATIBILITY_SCALER_x_THR0_ ADDR	R/W	12	Threshold for PMT0
COMPATIBILITY_SCALER_x_THR1_ ADDR	R/W	12	Threshold for PMT1
COMPATIBILITY_SCALER_x_THR2_ ADDR	R/W	12	Threshold for PMT2
COMPATIBILITY_SCALER_x_ENAB_ADDR	R/W	10	Control bits (See section 5.2)
COMPATIBILITY_SCALER_x_COUNT_ADDR	R/W	32	R: Scaler count, W: Reset count

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. Writing anything to the COMPATIBILITY_SCALER_x_COUNT register will reset the count.

5.15 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

5.16 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-firmeware 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
29:28	ENABLE40 = Down sampling clock phase
30	Spare
31	Triggered bin flag

The address map for the shower memories is shown below:

Description	Base Address
TRIGGER_MEMORY_SHWR0	0x8000 8000
TRIGGER_MEMORY_SHWR1	0x8001 0000
TRIGGER_MEMORY_SHWR2	0x8001 8000
TRIGGER_MEMORY_SHWR3	0x8002 0000
TRIGGER_MEMORY_SHWR4	0x8002 8000

When trying to reproduce the filtering and down sampling, one should select the 2-ENABLE40'th bin, the 3+2-ENABLE40'th bin, the 6+2-ENABLE40'th bin, etc. of the filtered trace.

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	0x8004 0000
TRIGGER_MEMORY_MUON1	0x8006 0000

6.4 Reading Shower & Muon Memories

The shower and muon memories can be read 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. Symbolic register addresses are defined in the "time_tagging_defs.h" header file. 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 TIME_TAGGING_BASE = 0x43c3 0000.

8 Radio Detector Interface

The Radio Detector Interface module manages communication between the FPGA and the cosmic ray radio wave detector. The following registers are defined:

Register	Description
RD_IFC_CONTROL	RD interface control register
RD_IFC_STATUS	RD interface status register
RD_IFC_ID	ID register
RD_RESET	Reset register

The bit usage for the RD_IFC_STATUS register is as follows:

Bit(s)	Bit Mask (or Shift)	Description
1:0	RD_BUF_RNUM	Buffer number to read
3:2	RD_BUF_WNUM	Buffer number currently being written
7:4	RD_BUF_BUSY	Bit mask of busy buffers
11:8	RD_BUF_FULL	Bit mask of full buffers
15:12	RD_PARITY0	Bit mask of buffers with parity errors on serial stream 0
19:16	RD_PARITY1	Bit mask of buffers with parity errors on serial stream 1
23:20	RD_BUF_TIMEOUT	Bit mask of buffers timing-out during transfer

The bit usage for the RD_IFC_CONTROL register is as follows:

Bit(s)	Bit Mask (or Shift)	Description
1:0	RD_BUF_RESET	Buffer number to reset

The ID register contains the date of last tagged changes to the module in the same format as the trigger ID register. The RD interface may be reset by writing a 1 to the RD_RESET register.

1. Receive trigger from trigger module
   (a) If not in middle of transfer from the RD
      i. Save current write buffer number
      ii. Pass trigger on to RD
      iii. Mark buffer as busy
      iv. Upon receipt of 60 Mhz clock from the RD, transfer serial data from RD to UUB FPGA memory (data continuously checked for correct parity).
      v. Wait 5 μs after end of clock from the RD for RD buffers to flush
      vi. Mark buffer as full and not busy
      vii. Go back to step 1) wait for next trigger
   (b) Otherwise, go back to step 1) to wait for next trigger

After an event trigger is seen by the processor, the following steps are be taken by the DAQ code:

1. Transfer WCD & SSD data buffer to processor
2. Reset corresponding WCD/SSD buffer full status after latching RD read buffer number.
   (a) This allows the WCD to accept new WCD and SSD traces
   (b) Note that this will invalidate RD_BUF_RNUM (which is just a copy of SHWR_BUF_RNUM), so this must be saved before performing the reset above.
3. Check if same buffer number is full for RD
   (a) If full, go to step 4
   (b) If not full then
      i. If busy, wait a short while & check again
         A. The RD interface incorporates a watchdog timer that will time-out in $\sim 550\ \mu s$ if the readout does not complete for some reason. If so, skip to step 4.
      ii. If not busy, mark RD buffer as empty and skip to step 5
4. Transfer RD data to processor
   (a) Note if parity errors flagged in transfer
5. Reset corresponding RD buffer full status

The base address of the RD interface module is RD_BASE = 0x43c6 0000. Symbolic register offsets and bit masks are defined in “rd_interface_defs.h”.

8.1 RD memory organization

The organization of the data in the RD memory was chosen to simplify unpacking. One memory block, containing 4 2048 word long buffers is used to store the traces. The format of each data word in RD_EVENT memory is shown below. The RD ADC data is (Modified May, 2019 for RD field tests).

Bits	Description
0	RD ADC0 parity
12:1	RD ADC0
15:13	Unused
16	RD ADC1 parity
28:17	RD ADC1
31:29	Unused

The RD event memory is starts at RD_EVENT_BASE=0x8000 0000. The RD event memory can be read via PDT.

Please see "rd_test.c", included in the sde_pld_hhddmmyy.tar file for an example of the readout sequence for the RD event memory.

9 AMIGA Interface

The AMIGA interface does not have any control or status registers readable by the CPU.

10 Other Modules of Interest

10.1 Interface module

Some miscellaneous and test functions are grouped together in an interface module with base address INTERFACE_UUB_BASE = 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
	Bit 1 - ADC Power Down
2	Bit 0 - Test point P65
3	Bit 0 - WATCHDOG output value
	Bit 1 - Enable WATCHDOG output

10.2 Test Control Module

The test control module allows enabling of some test features. It is at base address TEST_CONTROL_BASE = 0x43c4 0000. Symbolic register offsets and bit masks are defined in "test_control_defs.h". Starting with FPGA version 11270418 register 1 contains mode bits to customize somewhat the behavior of the fake data. The 5 low order mode bits apply only to the fake shower data. The fake muon data is generated with alternating separations of 20 ms and 10 ms, irrespective of the mode bits. The 10 higher order mode bits allow differential delays (in 8.33 ns steps) between the WCD ADCs to be specified, while another 8 allow specification of the pulse width. Note that now the fake data generator must be reset before using it.

It is also possible to play back a triggered event by loading a 2048 bin event trace can be loaded into memory. The WCD PMT0 and PMT1 signals are loaded into the memory starting at FAKE_EVENT0_BASE = 0x4E00 0000, with the PMT0 signal in the bits 11:0, and the PMT1 signal in bits 27:16. The WCD PMT2 and SSD signals are loaded into the memory starting at FAKE_EVENT1_BASE = 0x5200 0000, with the

PMT2 signal in bits 11:0 and the SSD signal in bits 27:16. Only the high gain channels are loaded. The low gain channels are derived from the high gain channel signals. The small PMT channel is not supported in this mode. This feature is activated by specifying fake data mode bits [4:0] = 7.

Register	Bit	Description
0 = USE_FAKE	USE_FAKE_PPS_BIT	Generate fake GPS PPS if 1
	USE_FAKE_SHWR_BIT	Generate fake shower data if 1
	USE_FAKE_MUON_BIT	Generate fake muon data if 1
	USE_FAKE_RD_BIT	Generate fake RD data if 1
	USE_FAKE_RDCLK_BIT	Generate fake RD CLK if 1
	DISABLE_TRIG_OUT_BIT	Disable front panel trigger output if set
	GENERATE_TRIG_OUT_BIT	Set front panel trigger output high if 1 DISABLE_TRIG_OUT overrides this
1 = FAKE_MODE		Bits 4:0 - Fake data mode bits
		Bits 6:5 - WCD PMT0 delay
		Bits 8:7 - WCD PMT1 delay
		Bits 10:9 - WCD PMT2 delay
		Bits 18-11 - Pulse width in 8.3ns bins
		Bit 19 - Exponential shape if set
		Bits 31-20 Pulse height (incl. pedestal)

The mode bits are described in the following table.

Value	Fake Data
0	Reset fake data generator
1	Fixed rate: 10 μs period
2	Fixed rate: 1 ms period
3	Fixed rate: 10 ms period
4	Fixed rate: 100 ms period
5	Fixed rate: 1 s period
6	Sawtooth ramp
7	Read fake event from memory, 2 s period
11	Random timing: $0 - 17.1$ μs spacing
15	Random timing: $0 - 273$ μs spacing
18	Random timing: $0 - 2.2$ ms spacing
21	Random timing: $0 - 17.5$ ms spacing
22	Random timing: $0 - 35$ ms spacing
23	Random timing: $0 - 70$ ms spacing
25	Random timing: $0 - 280$ ms spacing
28	Random timing $0 - 2.24$ s spacing
31	Random timing: $0 - 17.9$ s spacing
Anything else ≤ 31	Fixed rate: 5 s period

10.3 Digital Interface Module

The digital interface module is located at base address DIG_IFC_BASE = 0x43c8_0000. This module is positioned between the connections to digital interfaces and the functional modules described above. It controls whether the digital interfaces are configured for the factory test or for normal operation of the RD and AMIGA interface modules (and associated SPI/UART connections). By default, normal operation is enabled. Factory test mode must be selected by setting a bit in the control register. The following registers are defined:

Register	Description
0 = DIG_IFC_CONTROL	Control register
1= DIG_IFC_INPUT	Inputs from digital interface connectors
2 = DIG_IFC_OUTPUT	Outputs to digital interface connectors
3 = DIG_IFC_ID	Module ID in hhddmmyy format

The following bits are defined in the Control Register:

Bits	Description
16	Factory test mode if set to 0, normal operation otherwise
15:8	Interface 0 bits 7:0 set to output if 1 in factory test mode
7:0	Interface 1 bits 7:0 set to output if 1 in factory test mode

Bits 15:0, which control the direction of corresponding digital interface lines, are only used if bit 16 is set to 0. Otherwise the directions are set according to the requirements of the RD and AMIGA interfaces. That is, before using for RD and/or AMIGA bit 16 must be set to 1! The order of the bits in the DIG_IFC_INPUT and DIG_IFC_OUTPUT registers is the same as the low order 16 bits of the DIG_IFC_CONTROL register.

The Input Register returns a 1 in the corresponding bit if 1) the factory test mode is enabled, 2) the interface line direction is set to input, and 3) a high level is applied to the input. Otherwise 0 is returned. When the 1st 2 conditions are met, but the input is left floating, the result is indeterminate. The Output Register puts a high level on the corresponding digital interface line if 1) the factory test mode is enabled, 2) the interface line direction is set to output, and 3) the corresponding bit is set. If the first 2 conditions are met and the bit is not set, a low level is output. The DIG_IFC_CONTROL and DIG_IFC_OUTPUT registers will read back what has been written to them. The DIG_IFC_INPUT and DIG_IFC_ID registers are read only. If bit 16 of the control register is not set, the registers have no affect on the digital connectors.

10.4 Shower Trigger Interrupt Module

The shower trigger interrupt module is located at base address SDE_SHWR_TRIGGER_INTR_BASE = 0x43c1 0000. By convention the register names all have _ADDR appended. The following registers are defined:

Register	Description
0 = INTR_GLOBAL_EN	Bit 0 - Global trigger interrupt enable if set
1 = INTR_EN	Bit 0 - Enable shower trigger interrupt if set
2 = INTR_STATUS	Bit 0 - Shower trigger interrupt detected
3 = INTR_ACK	Bit 0 - Shower trigger interrupt acknowledge
4 = INTR_PENDING	Bit 0 - Shower trigger interrupt pending

The interrupt module will assert bit 0 of register 2 whenever an interrupt request from the trigger module is present. If the trigger interrupt is enabled in register 1, this will cause bit 0 of register 4 to be asserted. If, furthermore, the Global interrupt is enabled, an IRQ request to the processor will be generated. The interrupt request remains asserted (register 2) until the buffer is reset via the SHWR_BUF_CONTROL register. The IRQ remains asserted (register 4) until the interrupt is acknowledged by writing 1 to register 3. If another shower memory buffer is full at that time, the interrupt detected will be re-asserted.

10.5 Muon Trigger Interrupt Module

The muon trigger interrupt module is located at base address SDE_MUON_TRIGGER_INTR_BASE = 0x43c5 0000. By convention the register names all have _ADDR appended. The following registers are defined:

Register	Description
0 = INTR_GLOBAL_EN	Bit 0 - Global trigger interrupt enable if set
1 = INTR_EN	Bit 0 - Enable muon trigger interrupt if set
2 = INTR_STATUS	Bit 0 - Muon trigger interrupt detected
3 = INTR_ACK	Bit 0 - Muon trigger interrupt acknowledge
4 = INTR_PENDING	Bit 0 - Muon trigger interrupt pending

Similar comments as those above for the Shower Trigger Interrupt Module apply here also.

10.6 Time Tagging Interrupt Module

The time tagging interrupt module is located at base address TIME_TAGGING_INTR_BASE = 0x43c7 0000. By convention the register names all have _ADDR appended. The following registers are defined:

Register	Description
0 = INTR_GLOBAL_EN	Bit 0 - Global 1 PPS interrupt enable if set
1 = INTR_EN	Bit 0 - Enable 1 PPS interrupt if set
2 = INTR_STATUS	Bit 0 - 1 PPS interrupt detected
3 = INTR_ACK	Bit 0 - 1 PPS interrupt acknowledge
4 = INTR_PENDING	Bit 0 - 1 PPS interrupt pending

The time tagging interrupt module will assert bit 0 of register 2 whenever the 1 PPS signal from the GPS transitions from 0 to 1. Note that unlike the shower and muon trigger interrupts, the time tagging 1 PPS interrupt is edge triggered. If the 1 PPS interrupt is enabled in register 1, this will cause bit 0 of register 4 to be asserted. If, furthermore, the Global interrupt is enabled, an IRQ request to the processor will be generated. The IRQ remains asserted until the interrupt is acknowledged by writing 1 to register 3. This will also reset the interrupted detected signal.