Nuclear Physics Division
Fast Electronics Group& Hall B DAQ

Description and Requirements for the CLAS12 Drift Chamber Readout Board (DCRB)

Sergey Boyarinov
Chris Cuevas
Benjamin Raydo

Table of Contents

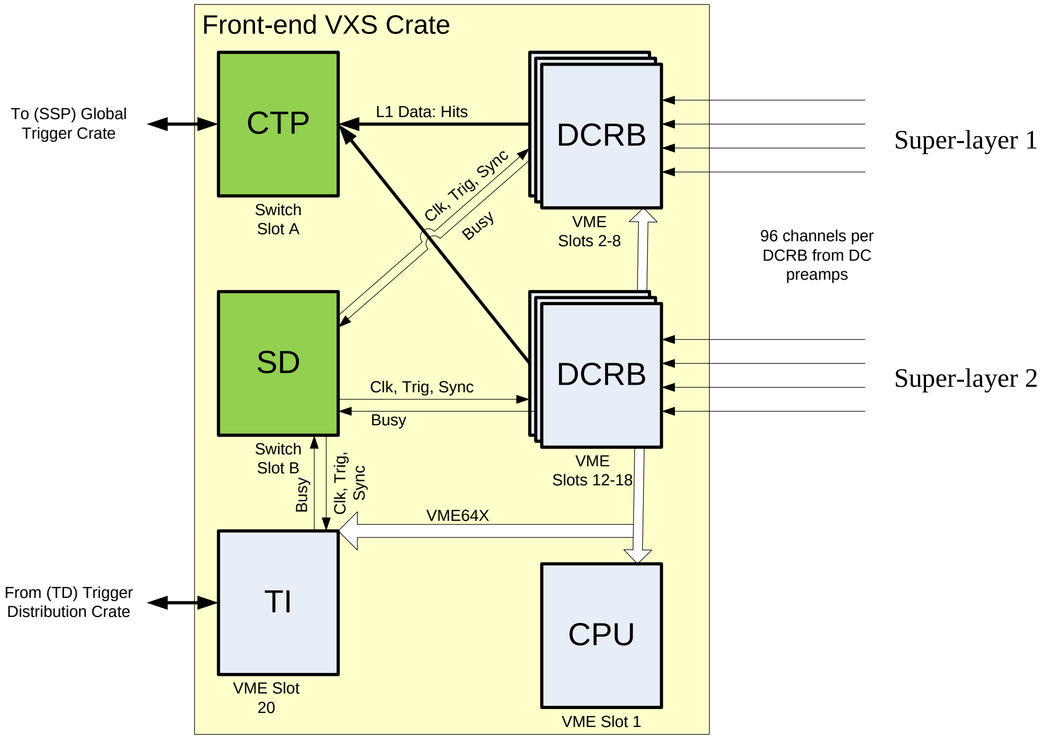
Section	Title	Page
1	Introduction	3
2	Specifications	6
3	Function Description	7-11
4	L1 Trigger Description	12-19
5	DCRB Event Data Format	20-23
6	DCRB VME Registers	24-54

1 Introduction

The Drift Chamber Readout Board (DCRB) is a 96 channel TDC with trigger output path that has amplifier/discriminator for each of its inputs. It has been designed to upgrade the Hall B DAQ electronics used by the drift chamber. The DCRB provides the functionality of the existing Amplifier-Discriminator-Board (ADB), MUX board, and FastBus TDC. By replacing the old equipment the DAQ readout performance is significantly improved (reducing dead-time), significant power is savings are realized, and a high speed trigger path allows for an advanced drift chamber tracking trigger to be added to the level 1 CLAS12 trigger.

The DCRB is a 6U, 160mm depth, VXS form factor/crate. The VXS board/crate was chosen because this is the primary DAQ/trigger platform used for CLAS12. Figure 1a shows the front-end crate configuration for the DCRB electronics. A VXS crate contains a CPU board, Trigger Interface (TI), Signal-Distribution (SD), Crate-Trigger-Processor (CTP), and 14 DCRBs. Signals within the crate communicate over the backplane and use VME or high speed point-to-point serial links. Signals to/from other crates use fiber optics to carry the information.

Figure 1a: DCRB in the Front-End Crate



CPU

The CPU is responsible for configuration of all devices in the VXS crate. It communicates using the VME bus to all modules. The CTP and SD modules are switch cards and are not directly connected to the VME bus, but rather to the TI with separate I^{2}C busses. The CPU communicates to the CTP and SD through the TI and since the SD and CTP do not generate event data the relatively slow I^{2}C bus connections do not present any bottlenecks.

During data taking, the CPU is responsible for reading out the event data from the TI and DCRB units using the VME bus, which currently has a 200MB/sec readout limit.

TI (Trigger Interface)

The TI is responsible for providing several critical signals to the front-end crates using fiber optics that are distributed from a single global Trigger Distribution Crate.

1. **L1 trigger accept signal:** signals all front end crates to synchronously capture an event for readout. This has a fixed latency.
2. **Global clock:** all crates run from a common global clock. This is distributed as 250MHz, but crates can convert this to 250MHz, 125MHz, 31.25MHz, or other values that are all synchronous to the 250MHz clock. Frequencies less than 250MHz start with a common phase offset across all crates through the use of synchronization signals which is handled by the TI.
3. **Synchronization signal:**
    - **a.** Aligns clock phases with frequencies less than 250MHz across all crates.
    - **b.** Resets the board timer that timestamp event data synchronously across all crates.
    - **c.** Synchronously starts the trigger logic across all crates.
4. **Busy signal:** crates can provide a busy feedback to the Trigger Distribution Crate which will inhibit triggers. This is used to create dead-time that allows the readout/network system to catch up and empty the front-end board buffers when lots/excessive triggers/data are received in a short period.

SD (Signal Distribution)

The SD is a fan-out board that provides the critical TI signals (CLK, trigger, sync, and busy) to all front-end boards using the VXS backplane. This eliminates the need for cables to carry these common signals to all the front-end boards.

CTP (Crate Trigger Processor)

The CTP is where trigger logic at the crate level is collected and processed. It communicates to each front-end board using the VXS backplane with up to four full-duplex high speed serial links. The current CTP implementation receives 2 links running at 2.5Gbps each from each DCRB. When overhead is taken into account, each DCRB can stream data for trigger processing at 4Gbps to the CTP. The CTP then processes all DCRB data streams in parallel and has a 10Gbps fiber link connection to the Global Trigger Crate where all other CTPs send their data.

DCRB (Drift Chamber Readout Board)

The DCRB is a front-end board that receives 96 pre-amplified drift chamber wire differential signals. Each input is separately amplified and discriminated with a time-over-threshold voltage discriminator that uses a programmable threshold common to all channels. The discriminated signals go to an FPGA where the leading-edge times are measured with a ~1ns resolution TDC. The TDC information is held in a pipeline (32μs maximum) where a trigger can cause the data to be captured and form an event for readout. The leading-edge of each discriminator output is also pulse width stretched by a programmable parameter in firmware and serialized to go the CTP for trigger processing. The CLAS12 drift chamber and DCRB are designed to only identify wire hits and does not perform dE/dX measurements. This significantly simplifies the complexity and cost of the input signal stage.

A block diagram of the FPGA firmware is shown in Figure 1b. Firmware controls many features of the design, which provides a high degree of flexibility for future optimizations or algorithm changes. The TDC, event building, and trigger pulse processing are the main function implemented in the FPGA firmware. These will be discussed in further detail in a following section.

Figure 1b: DCRB Firmware Block Diagram

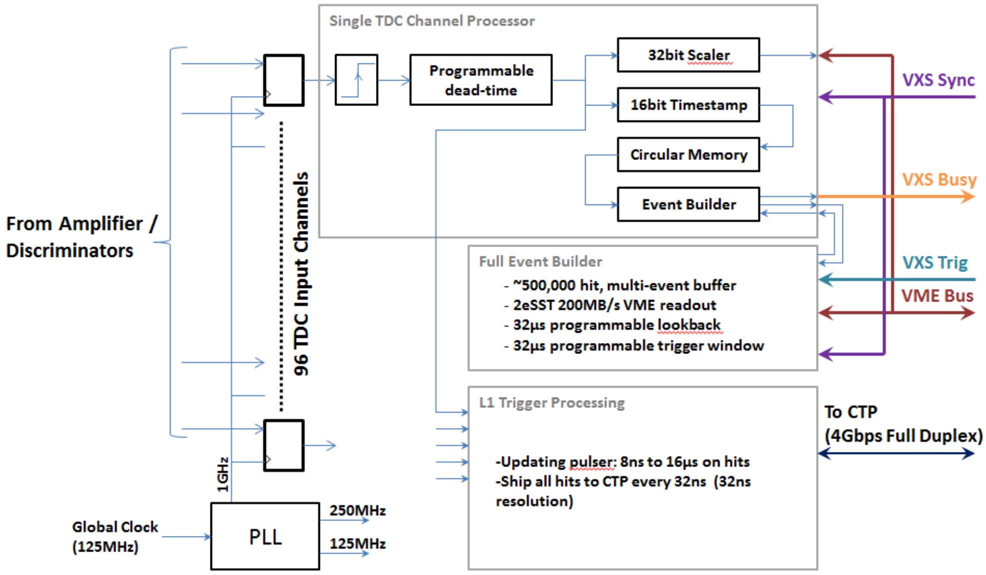
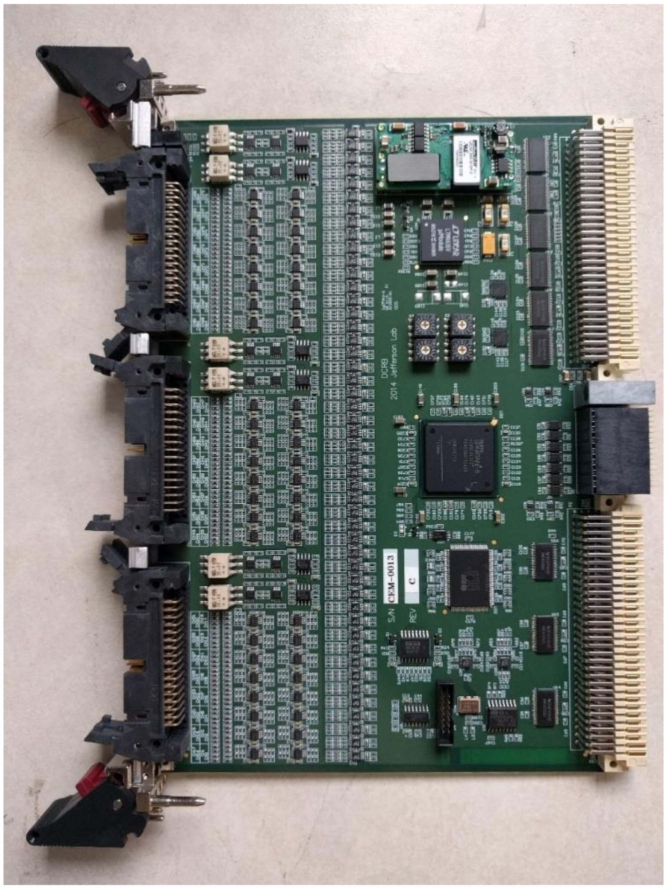

Single TDC Channel Processor
Programmable dead-time
32bit Scaler
16bit Timestamp
Circular Memory
Event Builder
VXS Sync
VXS Busy
VXS Trig
VME Bus
From Amplifier / Discriminators
96 TDC Input Channels
Full Event Builder
- ~500,000 hit, multi-event buffer
- 2eSST 200MB/s VME readout
- 32µs programmable lookback
- 32µs programmable trigger window
L1 Trigger Processing
-Updating pulser: 8ns to 16µs on hits
-Ship all hits to CTP every 32ns (32ns resolution)
To CTP
(4Gbps Full Duplex)
1GHz
Global Clock (125MHz)
PLL
250MHz
125MHz

Figure 1c: DCRB Assembly



2 Specifications

MECHANICAL

- Single width 9U VXS Payload Module

HIGH SPEED SERIAL P0 INPUTS/OUTPUTS:

- 125MHz LVPECL Clock
- Trig 1, Trig 2, Sync Inputs
- 4x 2.5Gbps Lane L1 trigger streams to CTP

Front Panel INPUTS/OUTPUTS:

- 96x TDC Inputs
  - Differential, 110Ω
  - Analog gain of 30, ~7ns rise time
  - 0-175mV discriminator
  - ~1ns TDC resolution
- 1x LEMO discriminator vref threshold output
- 1x LEMO calibration pulser input

INDICATORS: (Front Panel)

- Power OK – Green LED
- VME DTACK – Red LED

PROGRAMMING:

- On board JTAG Port or through VME

POWER REQUIREMENTS:

- ~30W total typical
- +5.0v @ 3.5A typical
- -12.0V @ 500mA typical
- +12.0V @ 500mA typical

ENVIRONMENT:

- Forced air cooling: T.B.D. CFM
- Commercial grade components (70 Celsius)

DAQ:

- Event buffer size: 500,000 hits
- VME transfer rate: 200MB/sec w/2eSST
- TDC Resolution: ~500ps RMS
- TDC channel dead-time: 32ns
- Maximum trigger rate: >1MHz
- Dead-time (100kHz, 10% occupancy): <<1%
- 32bit scaler per channel

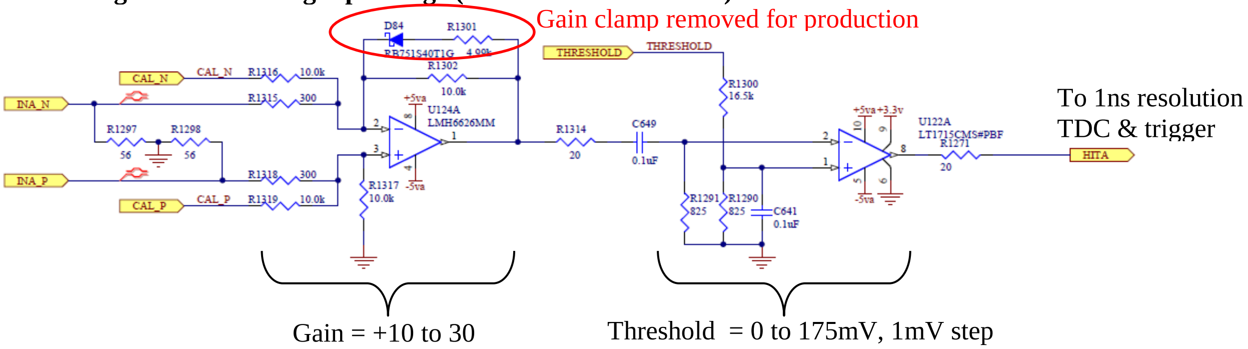
3 Functional Description

The DCRB can be split into several parts: analog signal path, time-to-digital converter (TDC), event builder, and trigger processing.

3.1 Analog signal path

Each of the 96 preamp input channels has a dedicated amplifier and comparator stage as shown in schematic of Figure 3.1a.

Figure 3.1a: Analog input stage (1 of 96 channels shown)



3.1.1 Input signal from chamber wire

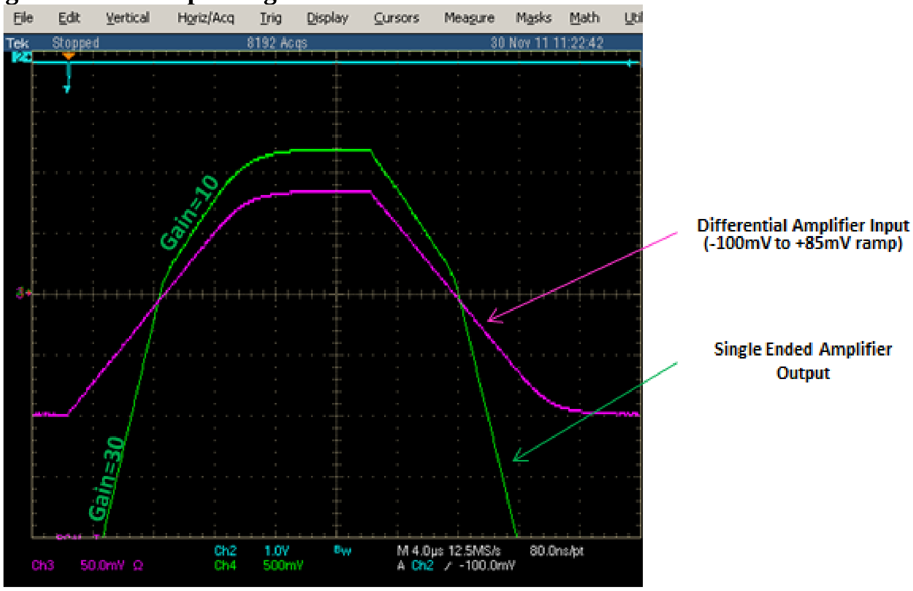
The signals from the drift chamber are pre-amplified and made differential to drive a long twisted pair cable bundle to reach the DCRB. The pre-amplifiers at the chamber provide a signal gain of about 2mV/µA where roughly half of the signal is lost over the twisted pair cable run.

3.1.2 Input amplifier

The amplifier provides a small signal gain of 30 so the signal at the comparator is about 30mV for a 1µA signal seen on the chamber wire. A gain of 30 is achieved for 0-1mV signals at the differential input. As the differential input voltage increases the gain lowers and saturates to about 10 as figure 3.1.2.a shows. This increases the dynamic range of the amplifier and reduces the risk of saturating the output of the amplifier.

.

Figure 3.1.2a: Amplifier gain transfer function

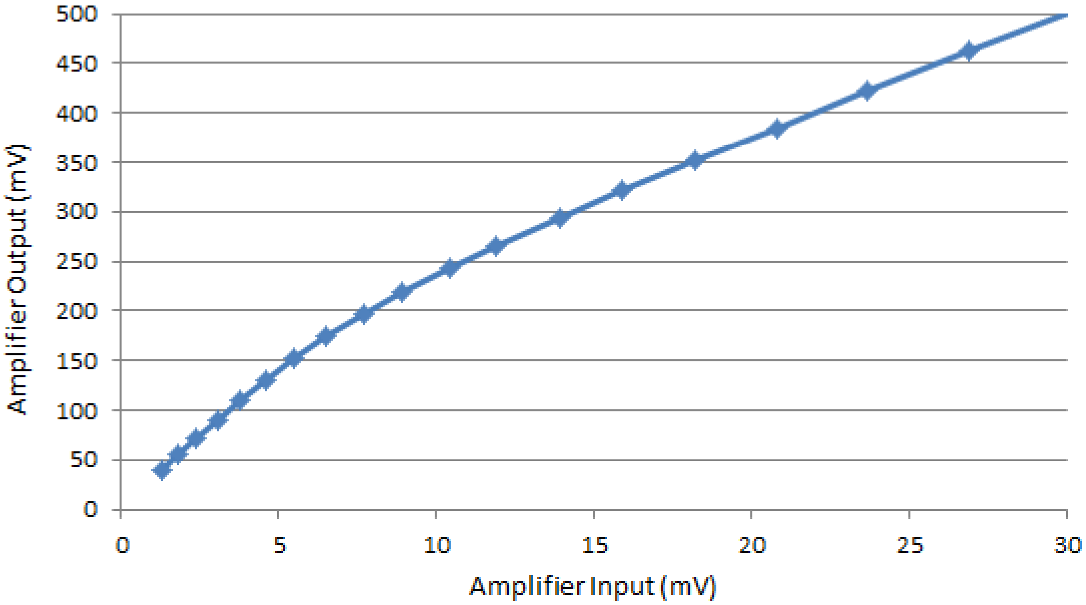


The input is bandwidth limited to ~60MHz and can accept rise-times down to 7ns before reaching the limit of the amplifier. Additionally, a local calibration pulse can be injected to each amplifier input which provides a simple way to test the gain/functionality of the local DCRB hardware.

3.1.3 Discriminator

Each amplified channel is voltage discriminated by a programmable threshold that is common to all channels of each DCRB. The discriminator threshold is programmable from 0 to 175mV with a 1mV resolution. The outputs of the 96 amplifier/discriminator channels are time-over-threshold TTL pulses that feed the FPGA. Figure 3.1.3a plots the differential signal pulse heights to what the comparator sees which may be helpful in determining the desired threshold to use.

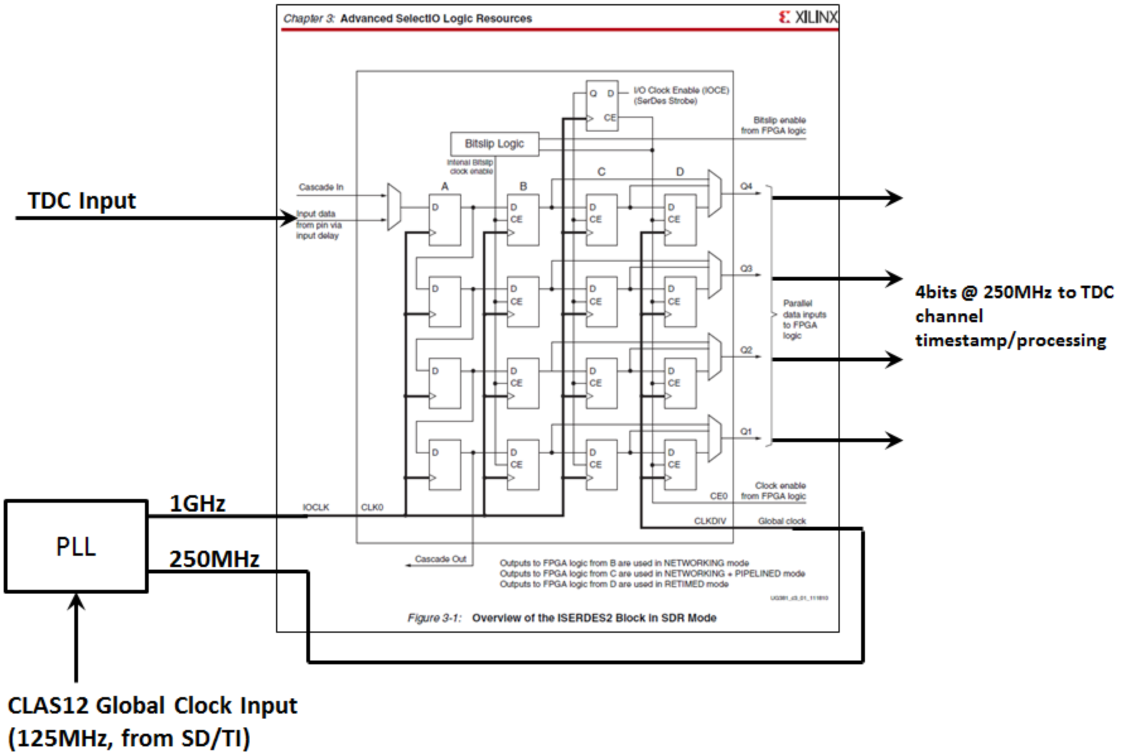
Figure 3.1.3a: Differential input to comparator input pulse height



3.2 TDC

The time-to-digital (TDC) function is performed inside the FPGA using a very simple, stable approach. The desired timing resolution of the TDC is ~1ns which is easily achievable with an FPGA (much higher resolution has been achieved and can be achieved in the DCRB by using non-conventional FPGA techniques that exploit the high speed carry chains inside the FPGA which is an option in firmware that can be implemented in the future of this board). The Xilinx Spartan 6 series FPGA have an ISERDES logic on each of the FPGA inputs which can run using a 1GHz clock input. Figure 3.2a shows how this FPGA component is used on each of the discriminated digital inputs to the FPGA. A programmable dead-time can be set on the TDC inputs to prevent a single chamber wire hit, which may generate multiple comparator crossings, from reporting multiple leading-edge times (only the first leading-edge is TDC then the remaining edges are ignored during the dead-time).

Figure 3.2a: TDC ISERDES (1 of 96 channels shown)



Chapter 3: Advanced SelectIO Logic Resources
XILINX
TDC Input
I/O Clock Enable (IOCE) (SerDes Strobe)
Bitslip enable from FPGA logic
Bitslip Logic
Internal Bitslip clock enable
Cascade In
Input data from pin via input delay
A
B
C
D
Q4
Q3
Q2
Q1
Parallel data inputs to FPGA logic
4bits @ 250MHz to TDC channel timestamp/processing
Clock enable from FPGA logic
CE0
CLKDIV
Global clock
IOCLK
CLK0
1GHz
PLL
250MHz
Cascade Out
Outputs to FPGA logic from B are used in NETWORKING mode
Outputs to FPGA logic from C are used in NETWORKING + PIPELINED mode
Outputs to FPGA logic from D are used in RETIMED mode
Figure 3-1: Overview of the ISERDES2 Block in SDR Mode
CLAS12 Global Clock Input (125MHz, from SD/TI)

The DCRB modules use the 125MHz signal from the TI (which is synchronized to the global 250MHz clock and phase aligned to the 125MHz clock in other DCRB crates). An internal PLL of the FPGA multiplies the 125MHz to 1GHz clock which feeds the ISERDES components as a single data rate (SDR) clock. The output of the ISERDES is a 4bit parallel word operating at 250MHz which contains 4 samples from the 1GHz input sampler. The 250MHz 4bit words are then searched for rising edge pattern, which identifies the drift chamber leading edges with a ~1ns timing resolution.

The TDC is has excellent INL and DNL characteristics thanks to the very stable 1GHz PLL clock source and because only the rising edge of the clock is used. Figure 3.2b plots the measure pulse delay vs the fed pulse delay and includes a linear fit.

Figure 3.2b: DCRB TDC Linearity

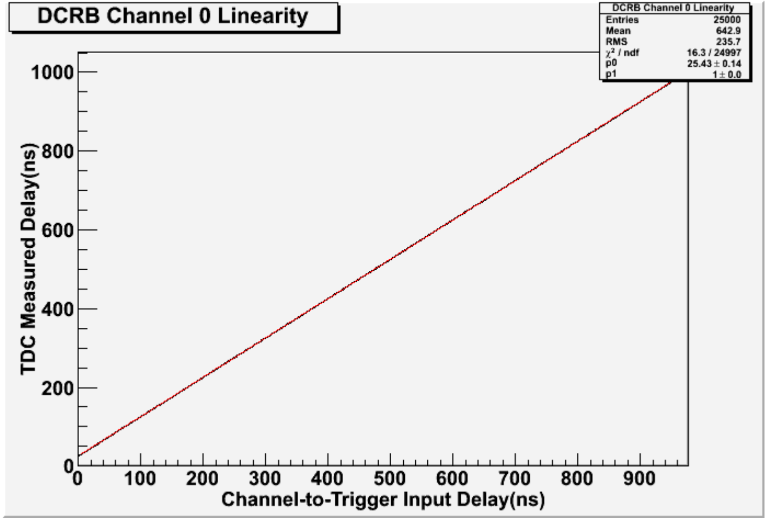
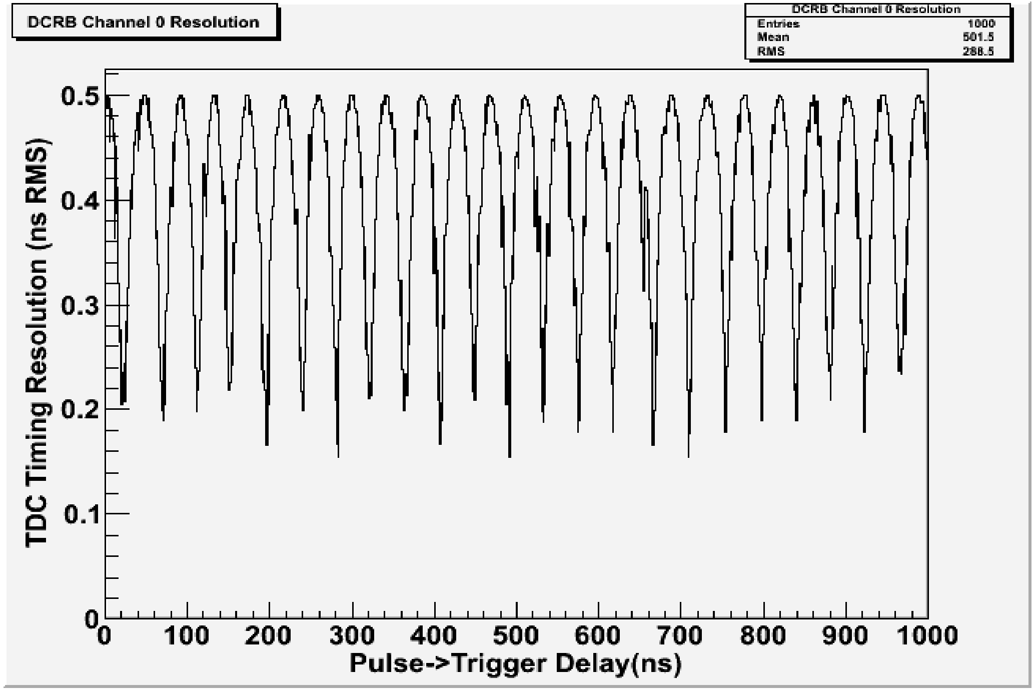


Figure 3.2c shows a plot for the measured TDC RMS resolution over a range of 0 to 1µs. For the test a delay was incremented by slightly less than 1ns which causes a periodic response which peaks where the 1GHz sampler performs the worst due to the 1ns jitter, but in the troughs the resolution is limited by the FPGA clock jitter and analog signal path jitter (i.e. 1ns delay can be measured very reliably and is limited by analog/clock jitter, 1.5ns delay measurement is worse because half the time 1ns is measured and the other half a 2ns time is measured).

Figure 3.2c: DCRB TDC Resolution



3.3 Event Builder

The DCRB TDC event builder implemented in the FPGA firmware and is broken into two parts: channel event builder and full event builder.

Channel Event Builder

4 TDC inputs are handled simultaneously by the channel event builder. As hits arrive at the DCRB, they have their leading-edge times measured by the TDC function and put into a buffer than holds the hits for up 32µs. The history buffer length also sets a limit on the average rate per channel to ensure the initial buffer does not fill up and cause data loss. The following table shows the average channel rates that can be handled without losing data as a function of the history length:

Table 3.3a: DCRB average rate for no data loss vs. buffer time length:

Avg. Rate per Channel	History Buffer Length
4MHz	32µs
8MHz	16µs
16MHz	8µs

For drift chamber signals these rate limitations present no problems because their rates are extremely lower than any of these limits. For CLAS12, an 8µs history buffer is used which is the expected system trigger latency.

When a trigger is received by the DCRB from the TI, a look back parameter and a window width parameter (both are specified in ns units from 0-32µs) determine how far back in time from the trigger point to capture TDC hits and how wide of a time window to capture. Hits that fall in this time window are captured. In the event data the TDC times for the hits are measured from the received trigger time. A small event FIFO temporarily stores the captured hits for each group of 4 TDC channels. This FIFO is large enough to hold 128 events at 100% occupancy for all channels before becoming full. As this FIFO nears the full state, a busy signal can be asserted to inhibit triggers which ensure no data is lost for triggered events (but this creates dead-time). There are 24 of these small event FIFOs to handle all 96 channels and they are constantly being emptied by the full event builder (at a rate of 50M TDC Words/sec).

Full Event Builder

The full event builder creates the shell of an event (headers, trailers, blocking/trigger information) and reads out the channel event builder FIFOs to fill the event with the DCRB specific TDC information. The 24 FIFOs from the channel event builders are readout (in a tree-link funneling structure) at up to 50M Words/sec. The full event builder uses an external 2MByte SRAM to buffer up to 500,000 event words, which is extremely large in comparison to what the running conditions of CLAS12. The full event builder can read and write at a total of 500MB/s to this external SRAM, which exceeds the maximum VME transfer rate the DCRB can handle.

Events are readout through the VME interface using the 2eSST transfer mode at a 200MB/s rate. The VME limitation is a non-issue for CLAS12 trigger rates and drift chamber occupancies (200MB/s allows 14 DCRB to operate at 100% occupancy at > 30kHz trigger rates), so the DCRB DAQ system should present no dead-time under normal operating conditions.

3.4 Trigger Processing

Trigger processing on locally on the DCRB is very minimal since a single DCRB has only has 96 wires visible. Instead of processing the local hits for the trigger, each DCRB ships the status of all 96 wire hits to the CTP every 32ns over the VXS backplane using 2 high speed serial lanes than each run at 2.5Gbps (the DCRB data only uses 3Gbps without any compression). This also means the timing resolution of chamber hits are 32ns at the trigger level. These hits have a programmable pulse width so that a leading-edge chamber pulse at the DCRB will report as being hit to the CTP for anywhere from 32ns to 16µs. This programmable pulse width is used to form the coincidence time so that the long drift times (<1µs) of the chamber are accounted for when performing hit based segment recognition at the CTP.

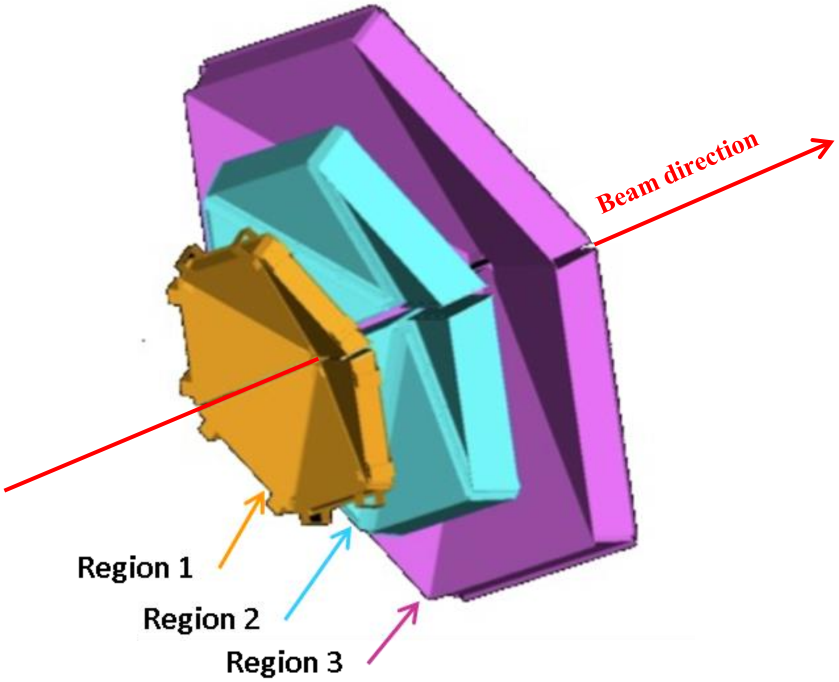
This generic implementation allows the DCRB firmware to work with essentially any trigger algorithm that would run at the CTP and global trigger level. How the DCRB fits into the CLAS12 level 1 trigger will be discussed next.

4 L1 Trigger

The primary motivation for the DCRB design was so that the CLAS12 drift chambers could be incorporated into the L1 trigger decision. By doing so, the efficiency of events where an electron is required increases significantly. Also many new trigger types can be developed that do not require an electron in trigger.

Figure 4a shows the CLAS12 drift chambers arranged in their operating positions. CLAS12 has a six-fold symmetry where it is broken into six identical sectors of detectors. Within each sector there are 3 regions of drift chambers, where region 1 is the smallest and closest to the target. There are a total of 18 drift chambers, but only 3 unique geometries according to the region. Each chamber encloses 2 stereo (+/-6°) super-layers. Each super-layer is an array of 112x6 sense wires arranged hexagonally.

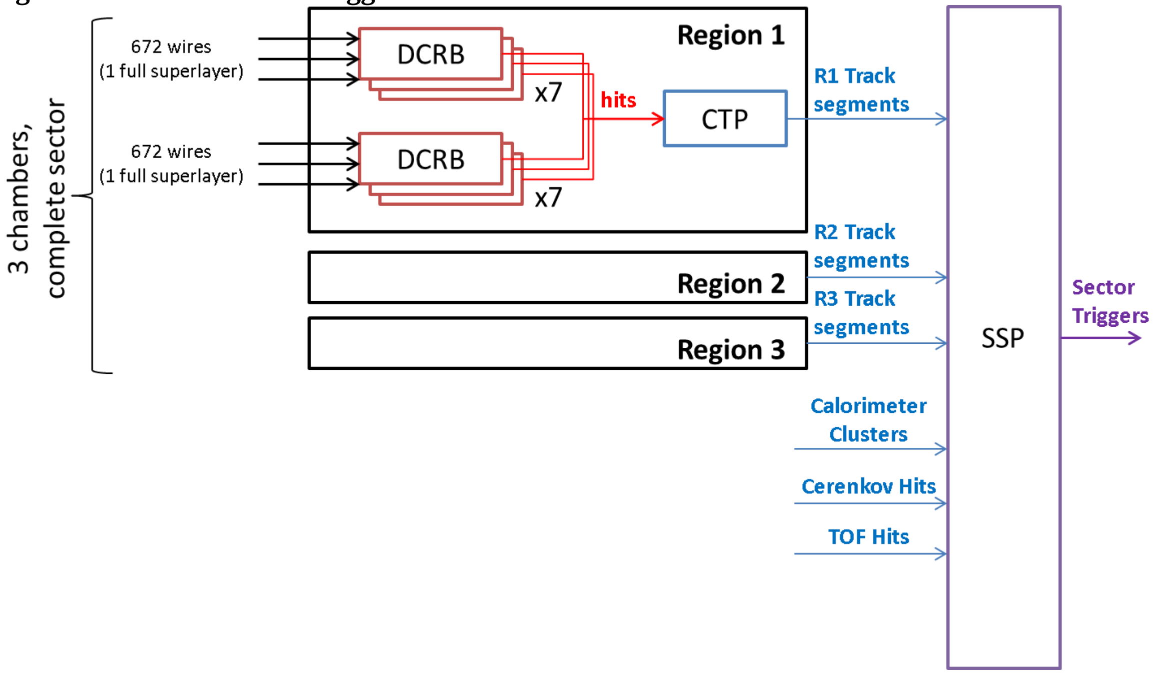
Figure 4a: CLAS12 Drift Chambers



The segmentation of the drift chambers isolates all 18 chambers from each other so that allows track segments can be found on the chambers independently. The entire drift chamber is composed of 24,192 sense wires, but segments can be recognized by viewing a single super-layer of 672 sense wires. This fits the layout of the DAQ segmentation so that 1 crate of DCRBs contains 1 sector, 1 region, and both super-layers of a drift chamber as shown in Figure 1a in the introduction; therefore, a total of 18 crates of DCRB electronics are needed (one for each chamber).

The drift chamber trigger is broken down into a few stages that work on different triggering hardware: hit processing, segment finding, and road finding/detector matching. Figure 4b shows the data flow across the trigger modules for a single sector of the drift chamber.

Figure 4b: Drift chamber trigger modules



The CLAS12 level 1 trigger is a fixed latency trigger system made up mostly of the Jefferson Lab designed 250Msps Flash Analog-to-Digital modules (FADC250), Crate-Trigger-Processor (CTP), SubSystem Processor (SSP), and Global Trigger Processor (GTP). The FADC250 modules are responsible for measuring detector pulse energy and time. The FADC250 pulse information is readout when triggered, but also the pulse information is sent to the CTP to build the level 1 trigger decision. The CTP uses the FADC250 pulse information depending on the detector: for calorimeters clusters are found and their positions and energies are reported, for other detectors the locations of the hits are reported. The CTP send the trigger data from each crate to the SSP where the data from the different detector types within a single sector meet.

Drift Chamber Hit Processing

The DCRB samples the 96 inputs at 1GHz where the leading edge is searched for. When a leading edge is found it is reported to the trigger processing firmware and the TDC event builder. A programmable hold off, or dead-time, is applied to the channel after the rising edge is detected to eliminate registering multiple hits due to multiple threshold crossings typically associated with a drift chamber hit.

The trigger processing firmware collects the hits from all 96 channels and creates a programmable mono-stable pulse on each channel that can be programmed to generate a pulse width up to 16µs long. The mono-stable pulser operates in updating mode, so if another hit arrives while the mono-stable is already active the timer will reset at that point which keeps the pulser active a full period longer from that point. The pulser outputs are sampled every 32ns and sent serially to the CTP over the VXS backplane. This requires a bandwidth of 3Gbps which is easy handled by the high speed serial transceivers and backplane of the DCRB, VXS backplane, and CTP.

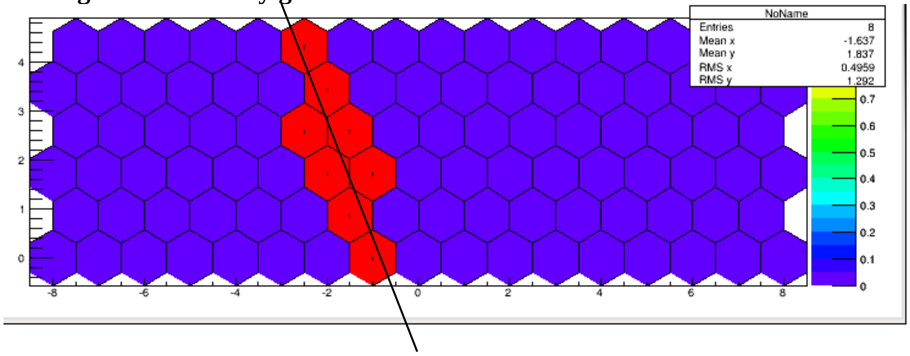
This implementation allows the CTP to see all chamber hits received by all DCRBs in the crate with a timing resolution of 32ns.

Drift Chamber Segment Finding

Segment finding is performed by the CTP, which resides in the VXS switch slot A. The CTP collects the hits from all DCRBs (a full chamber, which is 2 super-layers) in the crate with a 32ns timing resolution as described in the previous section. The CTP performs segment finding for the 2 super-layers independently and reports the found segments to the SSP using fiber optics.

The segment finding algorithm is a simple hit-based and essentially ignores drift times. The region 1 chambers have a maximum drift time of about 250ns and region 2 & 3 chambers have a maximum drift time of about 500ns. The drift times are compensated for in the algorithm by programming the DCRB trigger pulse widths to be a little larger than the drift times to ensure hits from the same segments are eventually seen in timing coincidence by the segment finder logic on the CTP. The acceptable segments are generated by a program written in ROOT, dctrackbin.C, which sweeps a line from -50° to +50° and -2 to +2 horizontal cell offset in fine increments. Segments are found based on the hexagonal cells that the line touches (Figure 4c shows an example of a segment defined by the sweeping line).

Figure 4c: Segment dictionary generator



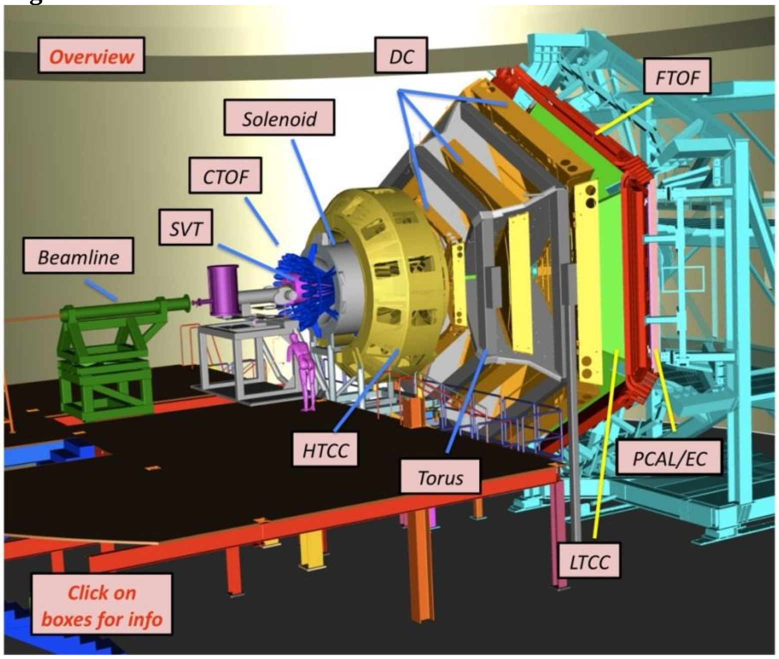
The possible segments are the same for any starting wire number, so the ROOT program only finds unique segments around a single wire number. The CTP reports found segments by constantly updating a 112x16 bit array. The indexes into this array are the segment entry wire number, 1 to 112, and the segment exit wire number delta, -8 to +7, which references the entry wire number. The ROOT program generates a VHDL source code containing the equations that map segments to the 112x16 bit array. The VHDL generated code includes a threshold that accounts for efficiency losses by defining a threshold for the number of layers (0-6) that must match segments. As the threshold is lowered, the resolution of the trigger degrades. Known bad channels of the chamber can be disabled for readout and set to report a constant hit for the trigger. In this way only efficiency degrades for that region rather than the whole chamber.

The 112x16 bit array is reported to the SSP for both super-layers every 256ns using a 16Gbps fiber optic link. The CTP has a programmable mono-stable updating pulser for each bit in the array that can extend the reporting time for up to 4μs to guarantee a coincidence can be formed with other detectors.

SSP Sector Trigger

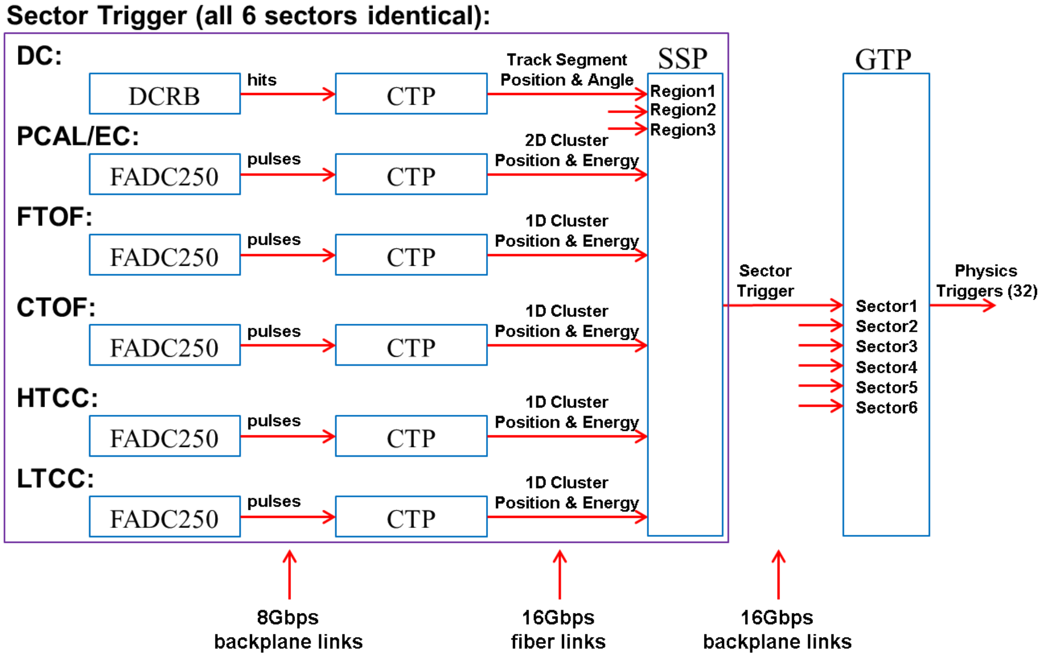
Figure 4d shows the assembled CLAS12 detector. Most detectors provide information to the trigger system (FTOF, PCAL/EC, LTCC, DC, HTCC, and CTOF). Many are sector based and do not need to share information with adjacent sectors to form triggers (FTOF, PCAL/EC, LTCC, and DC). The CTOF and HTCC detectors have signals relevant to multiple sectors at boundaries and therefore their trigger information is copied and made available to all sectors.

Figure 4d: CLAS12 detector



The SubSystem Processor in CLAS12 collects trigger information from up to 8 CTPs using 16Gbps fiber optics. The SSP computes trigger decisions for each individual sector (1 SSP per sector) by finding coincidences in space and time depending on the defined physics trigger. Figure 4e provides a block diagram of the trigger system. Front-end crates that participate in the trigger use either the FADC250 (a 250MHz FADC readout/trigger module – for fast/PMT detectors) or the DCRB (for the drift chamber). Trigger information flows from the FADC250 (digitized pulses information contain energy, time, and channel id for each pulse received) or from the DCRB (digitized chamber hits containing time and channel id) that flow to the CTP. The CTP performs a subsystem specific trigger algorithm on the data (in the case of the DCRB this would be segment finding as discussed previously) and send the processed trigger information to the SSP. The SSP now has triggering information from all subsystems for a single sector of CLAS12 and this information is used to make a sector trigger decision.

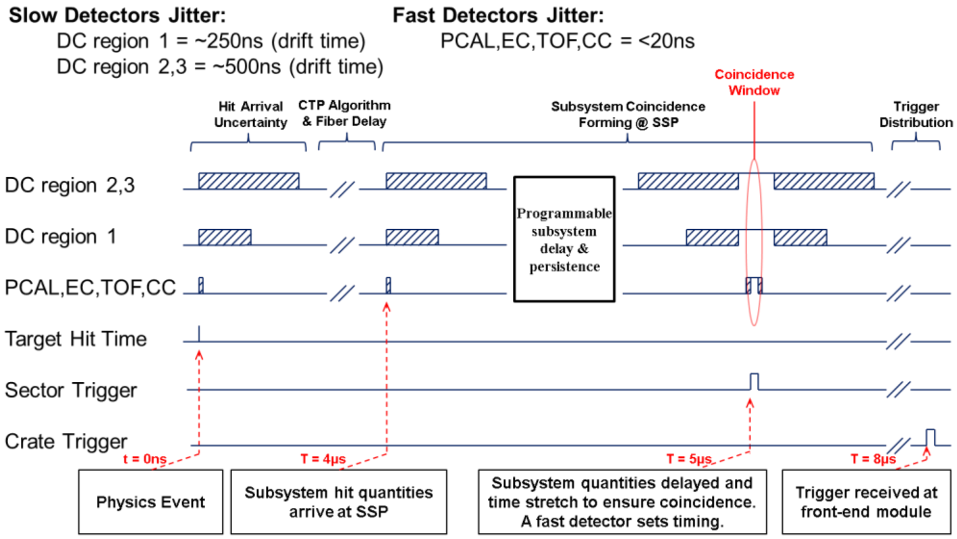
Figure 4e: CLAS12 Trigger System Layout



The trigger information from the CTP contains spatial information about hits that allow matching the drift chamber segments directly to the adjacent detectors. Initially we plan to not attempt reconstructing the full drift chamber track across all regions. Instead we plan to use the region 3 drift chamber segments to point at the calorimeters to check for a cluster in that area for a spatial/temporal coincidence. Similarly we can use the region 1 drift chamber segments to point at the HTCC and CTOF detectors to looks for a spatial/temporal coincidence. This conservative approach simplifies drift chamber trigger significantly and is a vast improvement over the previous L2 drift chamber trigger system. There are significant resources available in the SSP to explore full drift chamber track fitting in the trigger in the future if needed, but it may be found that additional resources are needed to perform this task and, if so, only the small quantity of SSP module would need to be upgraded.

The SSP must perform a timing coincidence between slows detectors (with large uncertainties in hit arrival time) with fast detectors (with small uncertainties in hit arrival time). To do this, the SSP has a programmable delay and coincidence time that is specific to each subsystem. Since the drift chambers have drift times (~250ns for region 1, ~500ns for region 2 and 3) found track segments will persist for a much longer time as compared with fast detectors. The final trigger signal will have a precise timing with respect to the physics event by using a fast detector signal as the final qualifier, which eliminates the large drift chamber segment timing uncertainties when the trigger accepted. In Figure 4f the timing uncertainties of the fast and slow detectors are shown along with the expected latencies through the trigger system. The SSP contains the programmable delay and persistence logic that is applied to individual subsystems which ensure a reliable coincidence window can be formed across all detectors 5μs after the target impact. The final trigger shows up at the front-end crates at 8μs, which is the maximum trigger latency allow (due to buffer limitations in the FADC250). The triggering system is pipelined so that even though the latencies are large, decisions are be made at a much fast time. The drift chamber segment tables are updated every 256ns at the SSP. The FADC250 detectors are much faster and typically run trigger computations at 250MHz.

Figure 4f: CLAS12 Trigger Timing Diagram

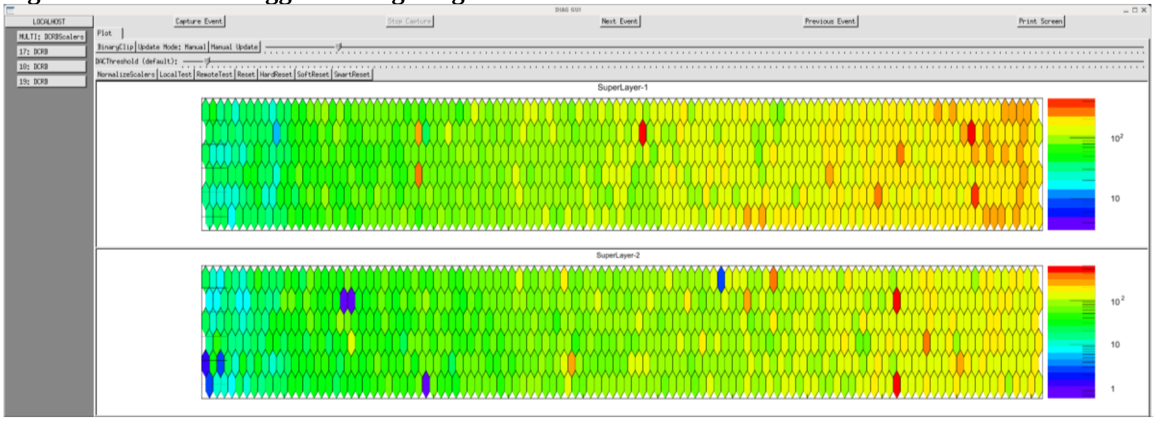

Slow Detectors Jitter:
DC region 1 = ~250ns (drift time)
DC region 2,3 = ~500ns (drift time)
Fast Detectors Jitter:
PCAL,EC,TOF,CC = <20ns
Coincidence Window
Hit Arrival Uncertainty
CTP Algorithm & Fiber Delay
Subsystem Coincidence Forming @ SSP
Trigger Distribution
DC region 2,3
DC region 1
PCAL,EC,TOF,CC
Target Hit Time
Sector Trigger
Crate Trigger
Programmable subsystem delay & persistence
t = 0ns
T = 4μs
T = 5μs
T = 8μs
Physics Event
Subsystem hit quantities arrive at SSP
Subsystem quantities delayed and time stretch to ensure coincidence. A fast detector sets timing.
Trigger received at front-end module

Drift Chamber Trigger Testing

In 2013, Hall B procured a full crate of DCRBs and we completed development of the DCRB firmware (triggering and readout). In we implemented the CTP firmware for segment finding on a Hall D trigger board (the GTP, but it will continue to be referred to as the CTP in this document). In this test we instrumented a full drift chamber region (2 super-layers) with the DCRB system and tested the trigger logic using cosmic rays.

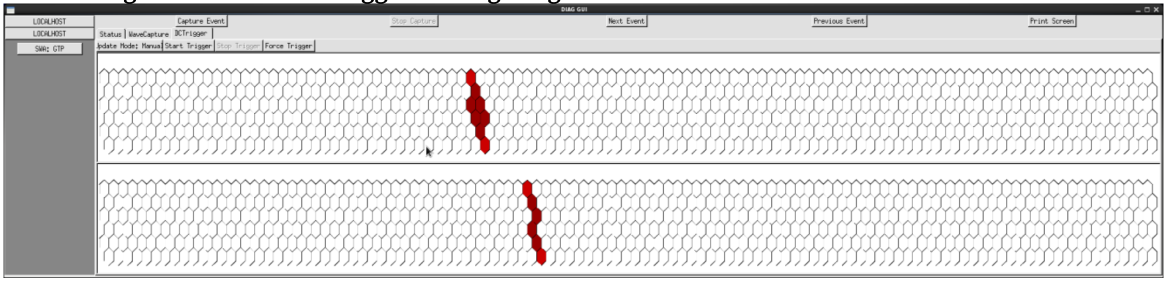
A display of the chamber scalers for each wire of the chamber region is shown in Figure 4g. This shows the cosmic (and noise) hits on the chamber wires, but notice the rate decreases from right to left, which is consistent with the shorter wires on the left side. The scalers have been very helpful for identifying dead channels, hot channels, and low/high voltage wiring issues.

Figure 4f: CLAS12 Trigger Timing Diagram



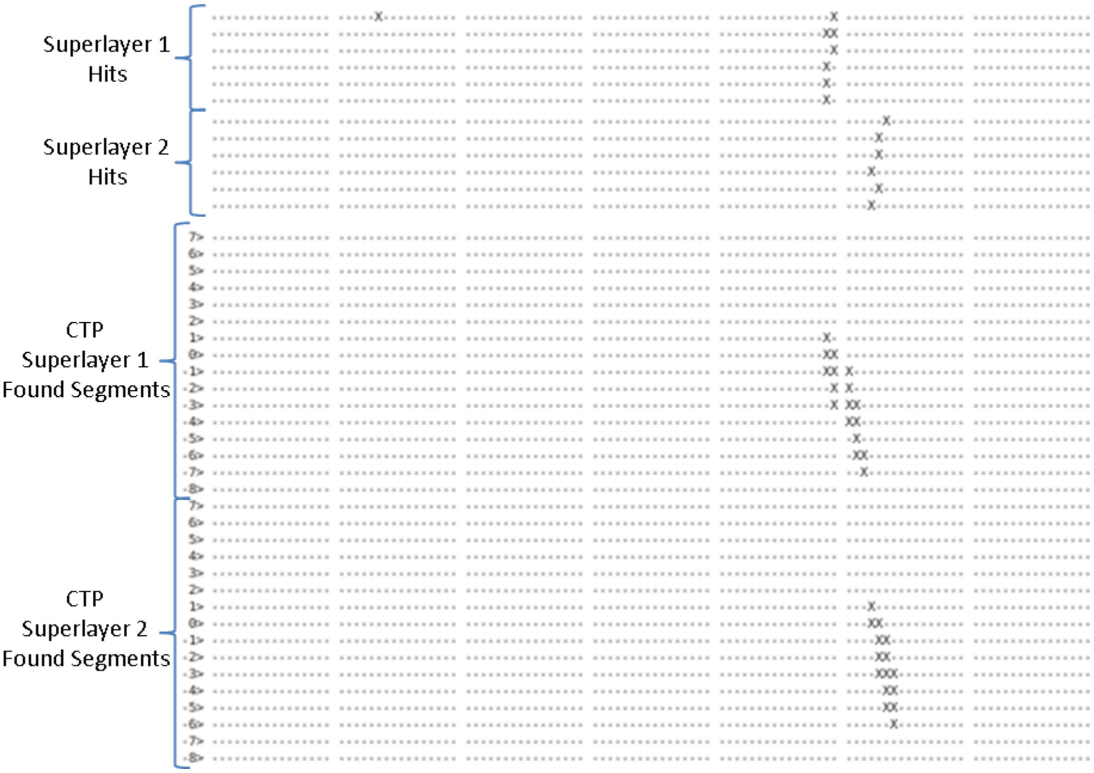
The CTP was then setup to generate a trigger signal if it found a single segment with 5 out of 6 layers hit. A useful debugging feature allows the CTP to capture the chamber hits +/-1µs around the segment finding trigger for visual inspection. An example view of cosmic track segments found by the CTP are shown in Figure 4g. The 2 segments shown are for both super layers of the same region which show very reasonable segment shapes.

Figure 4f: CLAS12 Trigger Timing Diagram



Once the segment finding logic seemed to be working the trigger efficiency was explored by finding track segments in the DCRB TDC hit event data offline and comparing it with the CTP trigger decision. To do this, we used a pair of scintillator paddles around the drift chamber and used the coincidence of these as the trigger signal, which captured the DCRB TDC hits and the CTP trigger decision. The CTP trigger decision was captured as an event and readout with the DCRB TDC data. The CTP event contained a 112x16 bit table for both super-layers that show the found track segments at their respecting starting wire position and exit wire position (as described in the segment finding section about). The CTP event was then compared with the offline reconstructed version of this event from the DCRBs to look for disagreements.

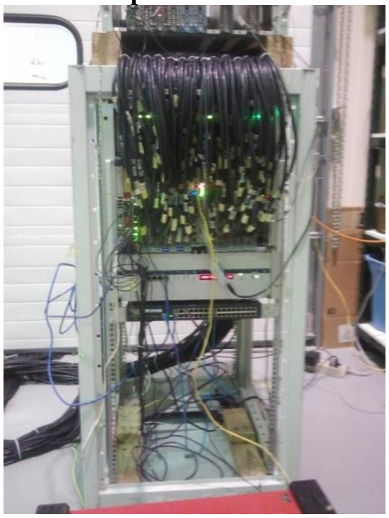
Sergey Boyarinov implemented nice ASCII display program that displays the DCRB TDC hits in their respective positions. Directly under the tracks is the CTP 112x16 for both super-layers which indicate the segments definitions that met the efficiency threshold. In this case the efficiency threshold was set to 4 out of 6 layers matching, which obviously widens the angular resolution of the track segments. The offline analysis produced exactly the same table as the CTP.


Superlayer 1 Hits
Superlayer 2 Hits
CTP Superlayer 1 Found Segments
CTP Superlayer 2 Found Segments

We had very good agreement between offline and trigger decisions, but occasionally we had a disagreement which indicates a bug in either the FPGA or offline analysis which needs to be resolved. It is expected that we can use an efficiency of 5 out of 6 layers matching for segment definitions which improves the angular resolution of the track segments. Even so, it is clear that the 112x16bit table is rather large and can be reduced without much of an impact on resolution.

The test setup is shown, which ran with a full crate of DCRBs, a CTP, and a region 3 drift chamber. There are 84 cables that each carry 17 differential pairs (16 chamber signals, 1 preamp pulser test signal). A standard VME based Intel Core I7 CPU board was used to readout the events.

Figure 4g: Full DCRB crate test setup with CLAS12 drift chamber



5. DCRB Readout Data Format

The DCRB readout data format utilizes the same encoding scheme defined for the JLAB FADC250. The word length for the readout data is 32bits. The event length is variable and depends on several factors (detector occupancy, headers, trailers, filler words).

Data Word Categories

Data words from the module are divided into two categories: Data Type Defining (bit 31 = 1) and Data Type Continuation (bit 31 = 0). Data Type Defining words contain a 4-bit data type tag (bits 30 - 27) along with a type dependent data payload (bits 26 - 0). Data Type Continuation words provide additional data payload (bits 30 – 0) for the *last defined data type*. Continuation words permit data payloads to span multiple words and allow for efficient packing of various data types spanning multiple data words. Any number of Data Type Continuation words may follow a Data Type Defining word.

Data Type List

0	Block Header
1	Block Trailer
2	Event Header
3	Trigger Time
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	DCRB TDC Hit
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Data Not Valid (empty module)
15	Filler Word (non-data)

Data Type: Block Header

Type:	0x0
Size:	1 word
Description:	Indicates the beginning of a block of events. (High-speed readout of a board or a set of boards is done in blocks of events)

31	30	29	28	27	26	25	24
1	0	0	0	0	SLOTID		
23	22	21	20	19	18	17	16
SLOTID		NUM_EVENTS					
15	14	13	12	11	10	9	8
NUM_EVENTS					BLOCK_NUMBER		
7	6	5	4	3	2	1	0
BLOCK_NUMBER							

BLOCK_NUMBER:

Event block number (used to align blocks when building events)

NUM_EVENTS:

Number of events in block

SLOTID:

Slot ID (set by VME64x backplane)

Data Type: Block Trailer

Type: 0x1

Size: 1 word

Description: Indicates the end of a block of events. The data words in a block are bracketed by the block header and trailer.

31	30	29	28	27	26	25	24
1	0	0	0	1	SLOTID		
23	22	21	20	19	18	17	16
SLOTID		NUM_WORDS					
15	14	13	12	11	10	9	8
NUM_WORDS							
7	6	5	4	3	2	1	0
NUM_WORDS							

NUM_WORDS:

Total number of words in block of events

SLOTID:

Slot ID (set by VME64x backplane)

Data Type: Event Header

Type: 0x2

Size: 1 word

Description: Indicates the start of an event. The included trigger number is useful to ensure proper alignment of event fragments when building events. The 27bit trigger number (134M count) is not a limitation, as it will be used to distinguish events within event blocks, or among events that are concurrently being built or transported.

31	30	29	28	27	26	25	24
1	0	0	1	0	TRIGGER_NUMBER		
23	22	21	20	19	18	17	16
TRIGGER_NUMBER							
15	14	13	12	11	10	9	8
TRIGGER_NUMBER							
7	6	5	4	3	2	1	0
TRIGGER_NUMBER							

TRIGGER_NUMBER:

Accepted event/trigger number

Data Type: Trigger Time

Type:	0x3
Size:	2 words
Description:	Time of trigger occurrence relative to the most recent global reset. The time is measured by a 48bit counter that is clocked from the 125MHz system clock. The assertion of the global reset clears the counter. The de-assertion of global reset enables counter and thus sets t=0 for the module. The trigger time is necessary to ensure system synchronization and is useful in aligning event fragments when building events.

Word 1:

31	30	29	28	27	26	25	24
1	0	0	1	1	0	0	0
23	22	21	20	19	18	17	16
TRIGGER_TIME_H							
15	14	13	12	11	10	9	8
TRIGGER_TIME_H							
7	6	5	4	3	2	1	0
TRIGGER_TIME_H							

TRIGGER_TIME_H:

This is the upper 24bits of the trigger time

Word 2:

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
TRIGGER_TIME_L							
15	14	13	12	11	10	9	8
TRIGGER_TIME_ L							
7	6	5	4	3	2	1	0
TRIGGER_TIME_ L							

TRIGGER_TIME_L:

This is the lower 24bits of the trigger time

Data Type: DCRB TDC Hit

Type:	0x8
Size:	1 word
Description:	FSSR2 strip hit words. A separate word is used for each strip hit that is inside the trigger window.

31	30	29	28	27	26	25	24
1	1	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	CHANNEL						
15	14	13	12	11	10	9	8
TDC							
7	6	5	4	3	2	1	0
TDC							

CHANNEL:

7bit channel number for TDC hit (0-95)

TDC:

Hit time in 1ns ticks with respect to trigger time

Data Type: Data Not Valid

Type: 0x14

Size: 1 word

Description: Module has no data available for readout. This can if the module is being read out too quickly after receiving (event building is in process and no data words have been put into the buffer yet) a trigger or if the module doesn't have any events to report.

31	30	29	28	27	26	25	24
1	1	1	1	0	UNDEFINED		
23	22	21	20	19	18	17	16
UNDEFINED							
15	14	13	12	11	10	9	8
UNDEFINED							
7	6	5	4	3	2	1	0
UNDEFINED							

Data Type: Filler Word

Type: 0x15

Size: 1 word

Description: Non-data word appended to the block of events. This is used to force the total number of 32-bit words read out of a module to be a multiple of 2 or 4 when

31	30	29	28	27	26	25	24
1	1	1	1	1	UNDEFINED		
23	22	21	20	19	18	17	16
UNDEFINED							
15	14	13	12	11	10	9	8
UNDEFINED							
7	6	5	4	3	2	1	0
UNDEFINED							

6. DCRB VME Registers

All DCRB board registers can be accessed through the VME bus in the following mode:

- A24: single cycle accesses, 32bit aligned read or write access (register specific)

Event readout can be access through the VME bus in the following modes:

- A32: single cycle, BLT, MBLT, 2eVME, 2eSST
- Note: transfer rate for 2eSST is 200MB/s

Register Summary:

Register Name	Description	Address Offset
Board Information		
A_FIRMWARE_REV	Firmware revision	0x0000
A_BOARDID	Board identification	0x0004

Debug/upgrade Configuration		
A_ICAP	FPGA configuration interface	0x0018
A_SPI_FLASH	Non-volatile flash interface	0x0014
A_SRAM_DBG_ADR	Sram test address	0x0060
A_SRAM_DBG_DATA	Sram test data	0x0064

Readout Configuration		
A_GRP_BUSY_FIFO	Event builder channel fifo full	0x0008
A_GRP_BUSY_TRIG	Event build channel trigger busy	0x000C
A_GRP_ERROR_FIFO	Event builder channel overrun	0x0010
A_ADR32M	Multi-board readout address	0x001C
A_LOOKBACK	Trigger lookback time	0x0020
A_WINDOW_WIDTH	Trigger window width	0x0024
A_BLOCK_CFG	Event blocking setup	0x0028
A_TDC_CFG	TDC deadtime	0x002C
A_CLOCK_CFG	Clock setup	0x0030
A_TESTPULSE_CFG	Amplifier calibration pulse setup	0x0034
A_DAC_CFG	Discriminator threshold	0x0038
A_TRIG_BUSY_THR	Busy threshold	0x003C
A_AD32	Readout address	0x0044
A_INTERRUPT	Interrupt setup	0x0048
A_INTERRUPT_ACK	Interrupt acknowledge	0x004C
A_GEO	Geographical address	0x0050
A_FIFO_WORD_CNT	Event FIFO word count	0x0054
A_FIFO_EVENT_CNT	Event FIFO event count	0x0058
A_FIFO_BLOCK_CNT	Event FIFO block count	0x0080
A_READOUT_CFG	Readout interrupt setup	0x005C
A_RESET	Soft reset	0x0068
A_CH_ENABLE_N0	TDC Channel disable 0-31	0x006C
A_CH_ENABLE_N1	TDC Channel disable 32-63	0x0070
A_CH_ENABLE_N2	TDC Channel disable 64-95	0x0074
A_INT_WORDCNT	Interrupt word count level	0x0084
A_INT_EVENTCNT	Interrupt event count level	0x0088
A_INT_BLOCKCNT	Interrupt block count level	0x008C
A_BUSY_WORDCNT	Busy word count level	0x0090
A_BUSY_EVENTCNT	Busy event count level	0x0094
A_BUSY_BLOCKCNT	Busy block count level	0x0098

I/O Configuration		
A_TRIG_SRC	Trigger setup	0x0040
A_SWAGPIO	VXS-SWA GPIO setup	0x0108
A_SWBGPIO	VXS-SWB GPIO setup	0x010C
A_TOKENIN_CFG	Readout token input setup	0x0018
A_TOKENOUT_CFG	Readout token output setup	0x001C
A_SDLINK_CFG	VXS-SDLINK setup	0x0020
A_TRIGOUT_CFG	VXS-TRIGOUT setup	0x0024

Pulser Configuration		
A_PULSER_PERIOD	Pulser period	0x0120
A_PULSER_LOW	Pulser high cycle time	0x0124
A_PULSER_NPULSES	Pulser pulse count	0x0128
A_PULSER_START	Pulser start	0x012C
A_PULSER_STATUS	Pulser status	0x0130

Trigger Configuration		
A_TRIGGER_CTRL	Trigger Pulse Width	0x009C
A_GTP_CTRL	GTP control	0x00A0
A_GTP_CTRL_TILE0	GTP tile control	0x00A4
A_GTP_DRP_CTRL	GTP DRP port	0x00A8
A_GTP_STATUS	GTP status	0x00AC
A_SOFTERR_CNT	GTP soft error counters	0x00B0
A_PRBSERR_CNT	GTP prbs error counters	0x00B4

Scalers		
A_SCALER_LATCH	Latch scalers	0x0078
A_SCALER_BUSY	Busy scaler	0x0FE8
A_SCALER_BUSY_CYCLES	Busy time scaler	0x0FEC
A_SCALER_VMECLK	VME clock scaler	0x0FF0
A_SCALER_SYNC	VXS-SYNC input scaler	0x0FF4
A_SCALER_TRIG1	VXS-TRIG1 input scaler	0x0FF8
A_SCALER_TRIG2	VXS-TRIG2 input scaler	0x0FFC
A_SCALER_TDC_0	TDC input 0 scaler	0x1000
…		
A_SCALER_TDC_95	TDC input 95 scaler	0x117C

6.1 Board Information Registers Section

Basic board information registers can be used to verify that this board is the DCRB and check for the software revision, which should be checked for compatibility.

Register: A_FIRMWARE_REV

Address Offset: 0x0000
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
FIRMWARE_REV_MAJOR							
7	6	5	4	3	2	1	0
FIRMWARE_REV_MINOR							

FIRMWARE_REV_MAJOR (RO):
Major firmware revision number
FIRMWARE_REV_MINOR (RO):
Minor firmware revision number

Register: A_BOARDID

Address Offset: 0x0004
Size: 32bits
Reset State: 0x5653434D

31	30	29	28	27	26	25	24
BOARD_ID							
23	22	21	20	19	18	17	16
BOARD_ID							
15	14	13	12	11	10	9	8
BOARD_ID							
7	6	5	4	3	2	1	0
BOARD_ID							

BOARD_ID (RO):
0x68675242 = “DCRB” in ASCII

6.2 Debug/upgrade Configuration Registers Section

The registers in this section are not intended to be used by a typical user of this board. These provide access to the configuration flash memory and the direct FPGA configuration. The SRAM registers provide a simple interface to test the external memory, which bypasses the FIFO logic the event builder uses when accessing this memory.

Register: A_ICAP

Address Offset: 0x0018
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	BUSY	CE	CLK	WRITE
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

DATA (R/W):
SPARTAN6_ICAP data port
WRITE (WO):
SPARTAN6_ICAP write port
CLK (WO):
SPARTAN6_ICAP clk port
CE (WO):
SPARTAN6_ICAP ce port
BUSY (RO):
SPARTAN6_ICAP busy port
Notes:

1) This interface provides direct access to the FPGA configuration interface. Only intended use is for a VME based FPGA reload after new firmware has been programmed into flash memory.

Register: A_SPI_FLASH

Address Offset: 0x0014
Size: 32bits
Reset State: 0x00000007

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	MISO	NCS	CLK	MOSI

MOSI, CLK, NCS (WO):
SPI interface outputs to flash memory
MISO (RO):
SPI interface input from flash memory
Notes:

1) This interface is used for firmware updates and general non-volatile parameter storage.

Register: A_SRAM_DBG_ADR

Address Offset: 0x0060
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	ADDR		
15	14	13	12	11	10	9	8
ADDR							
7	6	5	4	3	2	1	0
ADDR							

ADDR (WO):

Address to use for read/write when A_SRAM_DBG_DATA is accessed

Notes:

1) This interface is used for testing read/write accesses to the external SRAM, which is normally controlled by the event builder FIFO.

Register: A_SRAM_DBG_DATA

Address Offset: 0x0064
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
DATA							
23	22	21	20	19	18	17	16
DATA							
15	14	13	12	11	10	9	8
DATA							
7	6	5	4	3	2	1	0
DATA							

DATA (R/W):

When read (written), this reads (writes) data from (to) the external SRAM at the address specified in A_SRAM_DBG_ADR

Notes:

1) This interface is used for testing read/write accesses to the external SRAM, which is normally controlled by the event builder FIFO.

6.3 Readout Configuration Registers Section

Register: A_GRP_BUSY_FIFO

Address Offset:	0x0008
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
BUSY							
BUSY							
BUSY							

BUSY(x) (RO):

‘0’ – channels 4*x to 4*x+3 event FIFO is less than half full
‘1’ – channels 4*x to 4*x+3 event FIFO is half full or greater

Notes:

1) The OR of these busy bits can be sent to the TS to inhibit further triggers to ensure no data loss at the DCBR. The FIFO is 512 elements deep and feeds the full event builder large buffer.

Register: A_GRP_BUSY_TRIG

Address Offset:	0x000C
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
BUSY							
BUSY							
BUSY							

BUSY(x) (RO):

‘0’ – channels 4*x to 4*x+3 event trigger processing is not busy
‘1’ – channels 4*x to 4*x+3 event trigger processing is busy

Register: A_GRP_ERROR_FIFO

Address Offset: 0x0010
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
ERROR							
ERROR							
ERROR							

ERROR(x) (RO):

‘0’ – no buffer overflow/data loss has occurred on channels 4*x to 4*x+3

‘1’ – buffer overflow/data loss has occurred on channels 4*x to 4*x+3. DCRB event builder must be reset to clear this flag – this should never happen if busy is used!

Register: A_ADR32M

Address Offset: 0x001C
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
STATUS	-	-	TAKE	LAST	FIRST	EN	ADR_MAX
23	22	21	20	19	18	17	16
ADR_MAX							
15	14	13	12	11	10	9	8
							ADR_MIN
7	6	5	4	3	2	1	0
ADR_MIN							

ADR_MIN (R/W):

Low A32 address (bits 31:23) used for multi-board, token-passing readout

ADR_MAX (R/W):

High A32 address (bits 31:23) used for multi-board, token-passing readout

EN (R/W):

‘0’ – disables VME A32 multi-board addressing mode

‘1’ – enabled VME A32 multi-board addressing mode

FIRST (R/W):

‘0’ – not first board in token passing scheme

‘1’ – first board in token passing scheme

LAST (R/W):

‘0’ – not last board in token passing scheme

‘1’ – last board in token passing scheme

TAKE (R/W):

‘0’ – does nothing

‘1’ – gives board token

STATUS (R/W):

‘0’ – board does not have token

‘1’ – board does have token

Register: A_LOOKBACK

Address Offset: 0x001C
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
LOOKBACK							
7	6	5	4	3	2	1	0
LOOKBACK							

LOOKBACK (R/W):

0 to 65535 - Lookback time (in ns units) to go back from trigger point and look for TDC hits to capture for event building.

Register: A_WINDOW_WIDTH

Address Offset: 0x0024
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
WINDOW_WIDTH							
7	6	5	4	3	2	1	0
WINDOW_WIDTH							

WINDOW_WIDTH (R/W):

0 to 65535 – Window width time (in ns units) to capture TDC hits for event building after the look back time.

Register: A_BLOCK_CFG

Address Offset: 0x0028
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	BLOCK_SIZE		
7	6	5	4	3	2	1	0
BLOCK_SIZE							

BLOCK_SIZE (R/W):

Number of events per block to be built by the event builder.

Register: A_TDC_CFG

Address Offset: 0x002C
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-			
7	6	5	4	3	2	1	0
DEADTIME							

DEADTIME (R/W):

TDC channel deadtime to incur after each hit received in 8ns ticks. Must be a minimum of 32ns (DEADTIME=4). Typically used to prevent multiple TDC hits from being reported by a single signal that has opportunity to cross threshold multiple times.

Register: A_CLOCK_CFG

Address Offset: 0x0030
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	CLKRST	CLKSEL

CLKSEL (R/W):

‘0’ – selects onboard local 125MHz oscillator source for triggering/FSSR reference
‘1’ – selects VXS-SWB 125MHz oscillator source for triggering/FSSR reference

CLKRST (R/W):

‘0’ – clock system reset cleared
‘1’ – clock system reset set

Notes:

1) When changing CLKSEL, the CLKRST must be set and cleared to ensure.
2) A soft reset must be applied after changing clock sources. See A_RESET register.

Register: A_TESTPULSE_CFG

Address Offset: 0x0034
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	DCE5	DCE4	DCE3	DC2
7	6	5	4	3	2	1	0
DCE1	DCE0	FCE5	FCE4	FCE3	FCE2	FCE1	FCE0

FCEx (R/W):

‘0’ – disables front panel pulser for local amplifier on channels 16*x to 16*x+15
‘1’ – enables front panel pulser for local amplifier on channels 16*x to 16*x+15

DCEx (R/W):

‘0’ – disables front panel pulser for remote amplifier on channels 16*x to 16*x+15
‘1’ – enables front panel pulser for remote amplifier on channels 16*x to 16*x+15

Register: A_DAC_CFG

Address Offset: 0x0038
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	THRESHOLD					
7	6	5	4	3	2	1	0
THRESHOLD							

THRESHOLD (R/W):

0 to 16383 – sets discrimination threshold in 10.7µV units (range is 0 to 175mV). This threshold is set for all channels.

Register: A_TRIG_BUSY_THR

Address Offset: 0x003C
Size: 32bits
Reset State: 0x00000080

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
BUSY_THRESHOLD							

BUSY_THRESHOLD (R/W):

Defines the number of outstanding triggers the event builder has to process before a BUSY status is asserted. This can be used to inhibit triggers at the global trigger distribution.

Register: A_AD32

Address Offset: 0x0044
Size: 32bits
Reset State: 0xXXXX8000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
A32_BASE							
7	6	5	4	3	2	1	0
A32_BASE	-	-	-	-	-	-	A32_EN

A32_BASE (R/W):

A32 base address (bits 31:23)

A32_EN (R/W):

‘0’ – disables VME A32 addressing mode
‘1’ – enabled VME A32 addressing mode

Register: A_INTERRUPT

Address Offset: 0x0048
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
INT_EN	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
					INT_LEVEL		
7	6	5	4	3	2	1	0
INT_ID							

INT_ID (R/W):

VME bus interrupt ID

INT_LEVEL (R/W):

VME bus interrupt level

INT_EN (R/W):

VME bus interrupt enable

Register: A_INTERRUPT_ACK

Address Offset: 0x004C
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
INTERRUPT_ACK							
23	22	21	20	19	18	17	16
INTERRUPT_ACK							
15	14	13	12	11	10	9	8
INTERRUPT_ACK							
7	6	5	4	3	2	1	0
INTERRUPT_ACK							

INTERRUPT_ACK (WO):

Writing to this register will acknowledge any outstanding interrupt. This will allow further interrupt from this module to interrupt on the VME bus if any interrupting condition persists or occurs in the future.

Register: A_GEO

Address Offset: 0x0050
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
VME_ADDR							
23	22	21	20	19	18	17	16
VME_ADDR							
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	SLOTID				

VME_ADDR (RO):

VME address switch settings. The lower 8 bits form the A24 base address. The upper 8 bits are not used in the firmware, but are available to the user for any purpose desired (for example, the user could read this and set the A32_BASE to this value to use dip switch controlled A32 VME addressing).

SLOTID (RO):

VME geographical addressing slot. Slot 30 will be reported on parity error.

Notes:

1) Geographical addressing is only support when module is used on aVME64X compatibly crate. A parity error will be generated on non-VME64X compatible crates.

Register: A_FIFO_WORD_CNT

Address Offset: 0x0054
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
WORD_CNT							
23	22	21	20	19	18	17	16
WORD_CNT							
15	14	13	12	11	10	9	8
WORD_CNT							
7	6	5	4	3	2	1	0
WORD_CNT							

WORD_CNT (RO):

The number of 32bit event builder words currently residing in readout FIFO.

Register: A_FIFO_EVENT_CNT

Address Offset: 0x0058
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
EVENT_CNT							
23	22	21	20	19	18	17	16
EVENT_CNT							
15	14	13	12	11	10	9	8
EVENT_CNT							
7	6	5	4	3	2	1	0
EVENT_CNT							

EVENT_CNT (RO):

The number of event builder events currently residing in readout FIFO.

Register: A_FIFO_BLOCK_CNT

Address Offset: 0x0080
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
BLOCK_CNT							
23	22	21	20	19	18	17	16
BLOCK_CNT							
15	14	13	12	11	10	9	8
BLOCK_CNT							
7	6	5	4	3	2	1	0
BLOCK_CNT							

BLOCK_CNT (RO):

The number of event builder blocks currently residing in readout FIFO.

Register: A_READOUT_CFG

Address Offset: 0x005C
Size: 32bits
Reset State: 0x00000000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">EVT_WORD_INT_LEVEL</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">EVT_WORD_INT_LEVEL</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">EVT_NUM_INT_LEVEL</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="7">EVT_NUM_INT_LEVEL</td><td>BERREN</td></tr>
</table>

EVT_WORD_INT_LEVEL (R/W):

Range: 0 to 65535. Sets the 32bit word interrupt threshold for the event builder. If the number of 32bit event words inside the event builder FIFO is greater-than or equal to this value an interrupt will be generated if enabled by the A_INTERRUPT register.

EVT_NUM_INT_LEVEL (R/W):

Range: 0 to 32767. Sets the event count interrupt threshold for the event builder. If the number of events inside the event builder FIFO is greater-than or equal to this value an interrupt will be generated if enabled by the A_INTERRUPT register.

BERREN (R/W):

‘0’ – disable VME bus error assertion for end-of-event signaling (user must know event size or parse readout contents to ensure event synchronization/alignment)
‘1’ – enables VME bus error assertion for end-of-event signaling

Register: A_RESET

Address Offset: 0x0068
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	RESET

RESET (WO):

‘0’ – clear “soft” reset
‘1’ – set “soft” reset. This will clear all event builder FIFOs. To ensure reliable reset, no triggers should be delivered during the reset period.

Register: A_CH_ENABLE_N0

Address Offset:	0x006C
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
ENABLE_N							
23	22	21	20	19	18	17	16
ENABLE_N							
15	14	13	12	11	10	9	8
ENABLE_N							
7	6	5	4	3	2	1	0
ENABLE_N							

ENABLE_Nx (R/W):

‘0’ – channel x is enabled for TDC and trigger
‘1’ – channel x is disabled for TDC and trigger

Register: A_CH_ENABLE_N1

Address Offset:	0x0070
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
ENABLE_N							
23	22	21	20	19	18	17	16
ENABLE_N							
15	14	13	12	11	10	9	8
ENABLE_N							
7	6	5	4	3	2	1	0
ENABLE_N							

ENABLE_Nx (R/W):

‘0’ – channel 32+x is enabled for TDC and trigger
‘1’ – channel 32+x is disabled for TDC and trigger

Register: A_CH_ENABLE_N2

Address Offset:	0x0074
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
ENABLE_N							
23	22	21	20	19	18	17	16
ENABLE_N							
15	14	13	12	11	10	9	8
ENABLE_N							
7	6	5	4	3	2	1	0
ENABLE_N							

ENABLE_Nx (R/W):

‘0’ – channel 64+x is enabled for TDC and trigger
‘1’ – channel 64+x is disabled for TDC and trigger

Register: A_INT_WORDCNT

Address Offset: 0x0084
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
WORDCNT							
23	22	21	20	19	18	17	16
WORDCNT							
15	14	13	12	11	10	9	8
WORDCNT							
7	6	5	4	3	2	1	0
WORDCNT							

WORDCNT (R/W):

Word count in event FIFO that will generate an interrupt on VME bus.

Register: A_INT_EVENTCNT

Address Offset: 0x0088
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
EVENTCNT							
23	22	21	20	19	18	17	16
EVENTCNT							
15	14	13	12	11	10	9	8
EVENTCNT							
7	6	5	4	3	2	1	0
EVENTCNT							

EVENTCNT (R/W):

Event count in event FIFO that will generate an interrupt on VME bus.

Register: A_INT_BLOCKCNT

Address Offset: 0x008C
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
BLOCKCNT							
23	22	21	20	19	18	17	16
BLOCKCNT							
15	14	13	12	11	10	9	8
BLOCKCNT							
7	6	5	4	3	2	1	0
BLOCKCNT							

BLOCKCNT (R/W):

Block count in event FIFO that will generate an interrupt on VME bus.

Register: A_BUSY_WORDCNT

Address Offset:	0x0090
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
WORDCNT							
23	22	21	20	19	18	17	16
WORDCNT							
15	14	13	12	11	10	9	8
WORDCNT							
7	6	5	4	3	2	1	0
WORDCNT							

WORDCNT (R/W):

Word count in event FIFO that will generate a busy signal.

Register: A_BUSY_EVENTCNT

Address Offset:	0x0094
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
EVENTCNT							
23	22	21	20	19	18	17	16
EVENTCNT							
15	14	13	12	11	10	9	8
EVENTCNT							
7	6	5	4	3	2	1	0
EVENTCNT							

EVENTCNT (R/W):

Event count in event FIFO that will generate a busy signal.

Register: A_INT_BLOCKCNT

Address Offset:	0x0098
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
BLOCKCNT							
23	22	21	20	19	18	17	16
BLOCKCNT							
15	14	13	12	11	10	9	8
BLOCKCNT							
7	6	5	4	3	2	1	0
BLOCKCNT							

BLOCKCNT (R/W):

Block count in event FIFO that will generate a busy signal.

6.4 I/O Configuration Registers Section

This section covers the registers control how I/O signals interact with the firmware. Most of the signals support multiplexing to provide flexibility to aid in various configurations or debug. The multiplexor inputs are all the same for all signal that have support for multiplexed inputs and is defined as follows:

Multiplexor Input	Signal	Description
0	0	Constant ‘0’
1	1	Constant ‘1’
2	PULSER_OUTPUT	Onboard pulser output
3	FP_INPUT0	Reserved
4	FP_INPUT1	Reserved
5	FP_INPUT2	Reserved
6	FP_INPUT3	Reserved
7	VXS_SWB_SYNC	Reserved
8	VXS_SWB_TRIG1	Reserved
9	VXS_SWB_TRIG2	Reserved
10	VXS_SWA_GPIO0	VXS Switch A GPIO Input 0
11	VXS_SWA_GPIO1	VXS Switch A GPIO Input 1
12	VXS_SWB_GPIO0	VXS Switch B GPIO Input 0
13	VXS_SWB_GPIO1	VXS Switch B GPIO Input 1
14	BUSY	Internal busy
15	FSSR_GOTHIT	Reserved
16	DAC_TRIGGER	Reserved
17	DAC_TRIGGER_DELAYED	Reserved
18	BCOCLK	Reserved
19	TOKENOUT	VME Token Out
20	TOKENIN	VXS Switch B Token input
31-21	Reserved	Reserved

Register: A_TRIG_SRC

Address Offset: 0x0040
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	SWSYNC	SSYNC	T2SYNC	T1SYNC
7	6	5	4	3	2	1	0
-	-	-	HIRES	SWTRIG	STRIG	T2TRIG	T1TRIG

T1TRIG (R/W):
‘0’ - VXS Trig1 is disabled as a trigger source
‘1’ - VXS Trig1 is enabled as a trigger source
T2TRIG (R/W):
‘0’ - VXS Trig2 is disabled as a trigger source
‘1’ - VXS Trig2 is enabled as a trigger source
STRIG (R/W):
‘0’ - VXS Sync is disabled as a trigger source
‘1’ - VXS Sync is enabled as a trigger source
SWTRIG (WO):
‘1’ – Send software trigger (automatically cleared bit)
HIRES (R/W):
‘1’ – capture Trig1 or Trig2 trigger source with 1ns resolution
‘0’ – capture Trig1 or Trig2 trigger source synchronous to 125MHz selected clock
T1SYNC (R/W):
‘0’ - VXS Trig1 is disabled as a sync source
‘1’ - VXS Trig1 is enabled as a sync source
T2SYNC (R/W):
‘0’ - VXS Trig2 is disabled as a sync source
‘1’ - VXS Trig2 is enabled as a sync source
SSYNC (R/W):
‘0’ - VXS Sync is disabled as a sync source
‘1’ - VXS Sync is enabled as a sync source
SW SYNC (WO):
‘1’ – Send software

Register: A_SWAGPIO

Address Offset:	0x0108
Size:	32bits
Reset State:	0x01000100

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	DIR1
23	22	21	20	19	18	17	16
-	-	-	MUX_SRC1				
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	DIR0
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC0				

MUX_SRCx (R/W):

Selects VXS-Switch-A-SEx driver source: see MUX_SRC description at the beginning of this section for source signal description.

DIRx (R/W):

‘0’ – Set as output (drives VXS-Switch-A-SEx line)
‘1’ – Set as input (weak pull-up on VXS-Switch-A-SEx line)

Notes:

1) Default values should be left unchanged.

Register: A_SWBGPIO

Address Offset:	0x010C
Size:	32bits
Reset State:	0x01000100

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	DIR1
23	22	21	20	19	18	17	16
-	-	-	MUX_SRC1				
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	DIR0
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC0				

MUX_SRCx (R/W):

Selects VXS-Switch-B-SEx driver source: see MUX_SRC description at the beginning of this section for source signal description.

DIRx (R/W):

‘0’ – Set as output (drives VXS-Switch-B-SEx line)
‘1’ – Set as input (weak pull-up on VXS-Switch-B-SEx line)

Notes:

1) Default values should be left unchanged.

Register: A_TOKENIN_CFG

Address Offset: 0x0018
Size: 32bits
Reset State: 0x00000014

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC				

MUX_SRCx (R/W):

Selects readout token input source: see MUX_SRC description at the beginning of this section for source signal description.

Notes:

1) Default values should be left unchanged.

Register: A_TOKENOUT_CFG

Address Offset: 0x001C
Size: 32bits
Reset State: 0x00000013

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC				

MUX_SRCx (R/W):

Selects readout token input source: see MUX_SRC description at the beginning of this section for source signal description.

Notes:

1) Default values should be left unchanged.

Register: A_SDLINK_CFG

Address Offset: 0x0020
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC				

MUX_SRCx (R/W):

Selects readout token input source: see MUX_SRC description at the beginning of this section for source signal description.

Notes:

1) Default values should be left unchanged.

Register: A_TRIGOUT_CFG

Address Offset: 0x0024
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	MUX_SRC				

MUX_SRCx (R/W):

Selects readout token input source: see MUX_SRC description at the beginning of this section for source signal description.

Notes:

1) Default values should be left unchanged.

6.5 Pulser Configuration Registers Section

Register: A_PULSER_PERIOD

Address Offset: 0x0120
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
PERIOD							
23	22	21	20	19	18	17	16
PERIOD							
15	14	13	12	11	10	9	8
PERIOD							
7	6	5	4	3	2	1	0
PERIOD							

PERIOD (R/W):

Defines number of 8ns ticks for the pulser period

Register: A_PULSER_LOW

Address Offset: 0x0124
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
LOW_CYCLES							
23	22	21	20	19	18	17	16
LOW_CYCLES							
15	14	13	12	11	10	9	8
LOW_CYCLES							
7	6	5	4	3	2	1	0
LOW_CYCLES							

LOW_CYCLES (R/W):

Defines number of 8ns ticks of the pulser period the output stays low.

Register: A_PULSER_NPULSES

Address Offset: 0x0128
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
COUNT							
23	22	21	20	19	18	17	16
COUNT							
15	14	13	12	11	10	9	8
COUNT							
7	6	5	4	3	2	1	0
COUNT							

COUNT (R/W):

0x00000000: disable pulser output
0x00000001 to 0xFFFFFFFE: number of periods to deliver pulser output
0xFFFFFFFF: infinite cycle count for pulser output

Notes:

1) When using fixed count of pulses the pulser must be trigger to start by writing to the A_PULSER_START register

Register: A_PULSER_START

Address Offset: 0x012C
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
PULSER_START							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PULSER_START							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PULSER_START							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PULSER_START							

PULSER_START (WO):

Write any value to start pulser operation. The pulse number counter is cleared.

Register: A_PULSER_STATUS

Address Offset: 0x0130
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	DONE

DONE (RO):

‘0’ – pulser is still delivering pulses as defined in A_PULSER_NPULSES
‘1’ – pulser is is not active (either disabled or has finished fixed pulse count)

6.6 Trigger Registers Section

Register: A_TRIGGER_CTRL

Address Offset:	0x009C
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	TRIGGER_WIDTH			
7	6	5	4	3	2	1	0
TRIGGER_WIDTH							

TRIGGER_WIDTH (R/W):

0 to 4095 – trigger pulse width to send to CTP (in 8ns ticks) whenever TDC channel is hit..

Register: A_GTP_CTRL

Address Offset:	0x00A0
Size:	32bits
Reset State:	0x00000203

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	ERR_EN	ERR_RST	RESET	TXENPRBS
7	6	5	4	3	2	1	0
TXENPRBS	RXENPRBS		LOOPBACK			GTRESET	PWRDN

PWRDN (R/W):

‘1’ disable power of the transceiver
‘0’ enable power of the transceiver

GTRESET (R/W):

‘1’ reset transceiver
‘0’ release reset of transceiver

LOOPBACK (R/W):

“000” – keep at this value

RXENPRBS (R/W):

“00” – keep at this value

TXENPRBS (R/W):

“00” – keep at this value

RESET (R/W):

‘1’ reset link
‘0’ release reset of link

ERR_RST (R/W):

‘1’ reset error counts
‘0’ release reset error counts

ERR_EN (R/W):

‘1’ enable error counts
‘0’ disable error counts (error counters should be disabled when read)

Register: A_GTP_CTRL_TILE0

Address Offset:	0x00A4
Size:	32bits
Reset State:	0x0F402F40

31	30	29	28	27	26	25	24
-	-	TXPOL1	TXDIFFCTRL1				TXBUFDIFFCTRL1
23	22	21	20	19	18	17	16
TXBUFDIFFCTRL1		TXPREEMPHASIS1				RXEQMIX1	
15	14	13	12	11	10	9	8
-	-	TXPOL0	TXDIFFCTRL0				TXBUFDIFFCTRL0
7	6	5	4	3	2	1	0
TXBUFDIFFCTRL0		TXPREEMPHASIS0				RXEQMIX0	

Do not modify values on this register.

Register: A_DRP_CTRL

Address Offset:	0x00A8
Size:	32bits
Reset State:	0x00000000

31	30	29	28	27	26	25	24
-	-	-	-		DEN_TILE1	DEN_TILE0	DWE
23	22	21	20	19	18	17	16
-	DRP_ADDR						
15	14	13	12	11	10	9	8
DRP_DIN							
7	6	5	4	3	2	1	0
DRP_DIN							

Do not modify values on this register.

Register: A_GTP_STATUS

Address Offset: 0x00AC
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-							
15	14	13	12	11	10	9	8
	SRCRDYN	TXLOCK	CHUP	RXPOL3	RXPOL2	RXPOL1	RXPOL0
7	6	5	4	3	2	1	0
LANEUP3	LANEUP2	LANEUP1	LANEUP0	HARDERR3	HARDERR2	HARDERR1	HARDERR0

HARDERRx (RO):
‘0’ – no hard error on lane x
‘1’ – hard error occurred on lane x. Reset is required.

LANEUPx (RO):
‘0’ – lane x is down.
‘1’ – lane x is up

RXPOLx (RO):
‘0’ – rx polarity is normal
‘1’ – rx polarity is inverted

CHUP (RO):
‘0’ – channel is down
‘1’ – channel is up

TXLOCK (RO):
‘0’ – TX PLL not locked (likely due to missing 250MHz global clock)
‘1’ – TX PLL is locked

SRCRDYN (RO):
‘0’ – Data is current being received over channel
‘1’ – Data is not being received over channel

Register: A_SOFTERR_CNT

Address Offset: 0x00B0
Size: 32bits
Reset State: 0x00000000

31	30	29	28	27	26	25	24
ERRORS1							
23	22	21	20	19	18	17	16
ERRORS1							
15	14	13	12	11	10	9	8
ERRORS0							
7	6	5	4	3	2	1	0
ERRORS0							

ERRORSx (RO):
soft error count seen on lane x

Register: A_PRBSERR_CNT

Address Offset:	0x00B4
Size:	32bits
Reset State:	0x00000000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">ERRORS1</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">ERRORS1</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">ERRORS0</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">ERRORS0</td></tr>
</table>

ERRORSx (RO):

prbs error count seen on lane x

6.7 Scalers Registers Section

Register: A_SCALER_LATCH

Address Offset: 0x0078
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_LATCH							
23	22	21	20	19	18	17	16
SCALER_LATCH							
15	14	13	12	11	10	9	8
SCALER_LATCH							
7	6	5	4	3	2	1	0
SCALER_LATCH							

SCALER_LATCH (WO):

Write any value to this register to latch all board scalers.

Notes:

1) Scalers are all buffered and auto-cleared when latched for readout.

Register: A_SCALER_BUSY

Address Offset: 0x0FE8
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_BUSY							
23	22	21	20	19	18	17	16
SCALER_BUSY							
15	14	13	12	11	10	9	8
SCALER_BUSY							
7	6	5	4	3	2	1	0
SCALER_BUSY							

SCALER_BUSY (RO):

Number of rising edges seen on internal BUSY since last scaler latch

Register: A_SCALER_BUSY_CYCLES

Address Offset: 0x0FEC
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_BUSY_CYCLES							
23	22	21	20	19	18	17	16
SCALER_BUSY_CYCLES							
15	14	13	12	11	10	9	8
SCALER_BUSY_CYCLES							
7	6	5	4	3	2	1	0
SCALER_BUSY_CYCLES							

SCALER_BUSY_CYCLES (RO):

Number of clock cycles (125MHz) internal BUSY was seen high since last scaler latch.

Register: A_SCALER_VMECLK

Address Offset: 0x0FF0
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_VMECLK							
23	22	21	20	19	18	17	16
SCALER_VMECLK							
15	14	13	12	11	10	9	8
SCALER_VMECLK							
7	6	5	4	3	2	1	0
SCALER_VMECLK							

SCALER_VMECLK (RO):

Number of VMECLK clock edges (50MHz) seen since last scaler latch.

Notes:

1) This scaler can be used to normalize other scalers latched by the GLOBAL.

Register: A_SCALER_SYNC

Address Offset: 0x0FF4
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_SYNC							
23	22	21	20	19	18	17	16
SCALER_SYNC							
15	14	13	12	11	10	9	8
SCALER_SYNC							
7	6	5	4	3	2	1	0
SCALER_SYNC							

SCALER_SYNC (RO):

Number of rising edges seen on the VXS-SWB-SYNC input since last scaler latch

Register: A_SCALER_TRIG1

Address Offset: 0x0FF8
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_TRIG1							
23	22	21	20	19	18	17	16
SCALER_TRIG1							
15	14	13	12	11	10	9	8
SCALER_TRIG1							
7	6	5	4	3	2	1	0
SCALER_TRIG1							

SCALER_TRIG1 (RO):

Number of rising edges seen on the VXS-SWB-TRIG1 input since last scaler latch

Register: A_SCALER_TRIG2

Address Offset: 0x0FFC
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_TRIG2							
23	22	21	20	19	18	17	16
SCALER_TRIG2							
15	14	13	12	11	10	9	8
SCALER_TRIG2							
7	6	5	4	3	2	1	0
SCALER_TRIG2							

SCALER_TRIG2 (RO):

Number of rising edges seen on the VXS-SWB-TRIG2 input since last scaler latch

Register: A_SCALER_TDC0, …, A_SCALER_TDC95

Address Offset: 0x1000, …, 0x117C
Size: 32bits
Reset State: 0xXXXXXXXX

31	30	29	28	27	26	25	24
SCALER_TDC							
23	22	21	20	19	18	17	16
SCALER_TDC							
15	14	13	12	11	10	9	8
SCALER_TDC							
7	6	5	4	3	2	1	0
SCALER_TDC							

SCALER_TDC (RO):

Number of hits seen for the TDC channel.