16-Channel Discriminator/Scaler VME Module

Revision D

Jefferson Lab

Fast Electronics Group

Feb 11th, 2011

**Overview**

The 16-Channel Discriminator/Scaler Board contains 16 non-updating dual-threshold discriminators, programmable digital delays, and two 32-bit scalers per discriminator and threshold. The discriminator pulses are output as differential ECL logic levels through two front-panel headers. One group of outputs will connect to a TDC and the other group can be used as input to trigger logic. Both TDC and trigger output channels can individually be enabled/disabled with outputs widths and delays being user programmable. All programming is done through VME registers.

All discriminators and logic reside on a 6U VME64x mainboard. Each channel contains two analog receiver fast comparators (discriminator), and pulsers. Each discriminator channel has 2 programmable thresholds which can be programmed from VME. The output pulse width is also programmable from VME, but is common to the TDC and trigger discriminator channels separately. The TDC output is driven from the discriminator channel and not routed through the FPGA to minimize jitter and delays. The trigger (TRG) output is the second threshold per discriminator and is routed through the FPGA. The TRG output can be individually delayed in 4ns steps up to 1020ns and the pulse reshaped in the FPGA to provide a 4 to 1024ns pulse width. A TRG output delay setting of zero bypasses delay and pulse reshaping logic. The TRG output of the discriminator can also select which discriminator threshold source to use in the case a single threshold is desired for the TDC and TRG outputs.

Each discriminator output pulse is recorded by two 32 bit counters (scaler). For each channel an external gate (NIM) is applied to one scaler while the other scaler is free running. Scalers can be latched, read, and cleared through VME. There is a “OR” (NIM level) ouput that is the logical OR of all the unmasked discriminator outputs.

Discriminator outputs are provided as dECL levels on the front panel for interfacing with TDCs and trigger logic.

The VME64x interface is A32/A24/D32/D64/BLT32/BLT64/2eVME/2eSST with support for interrupts.

Table of Contents

**Overview1**

**Discriminator/Scaler Block Diagram3**

**Specifications4-5**

**PCB Overview6**

**Front Panel7**

**VME Accessible Registers8**

**Module Performance27**

Signal Jitter27

Slew Rate Dispersion27

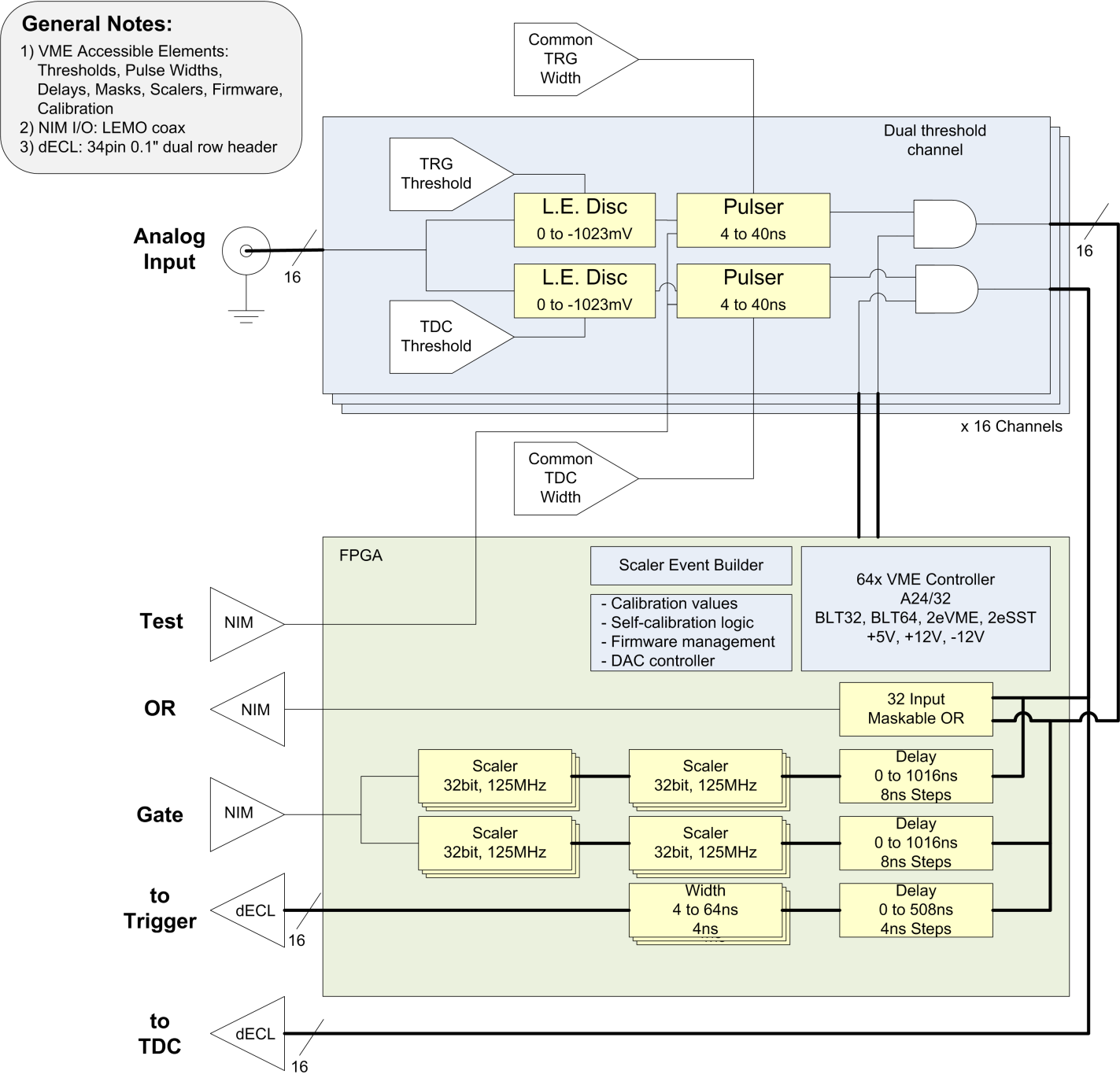
Channel Isolation28

Maximum Rate28

**Scaler Event Readout29**

**Firmware Revision History33**

1. **Discriminator/Scaler Block Diagram**

****

**Delay**

**0 to 1020ns**

**4ns step**

**Delay**

**0 to 1020ns**

**4ns step**

**Delay**

**0 to 1020ns**

**4ns step**

**Width**

**4 to 1024ns**

**4ns step**

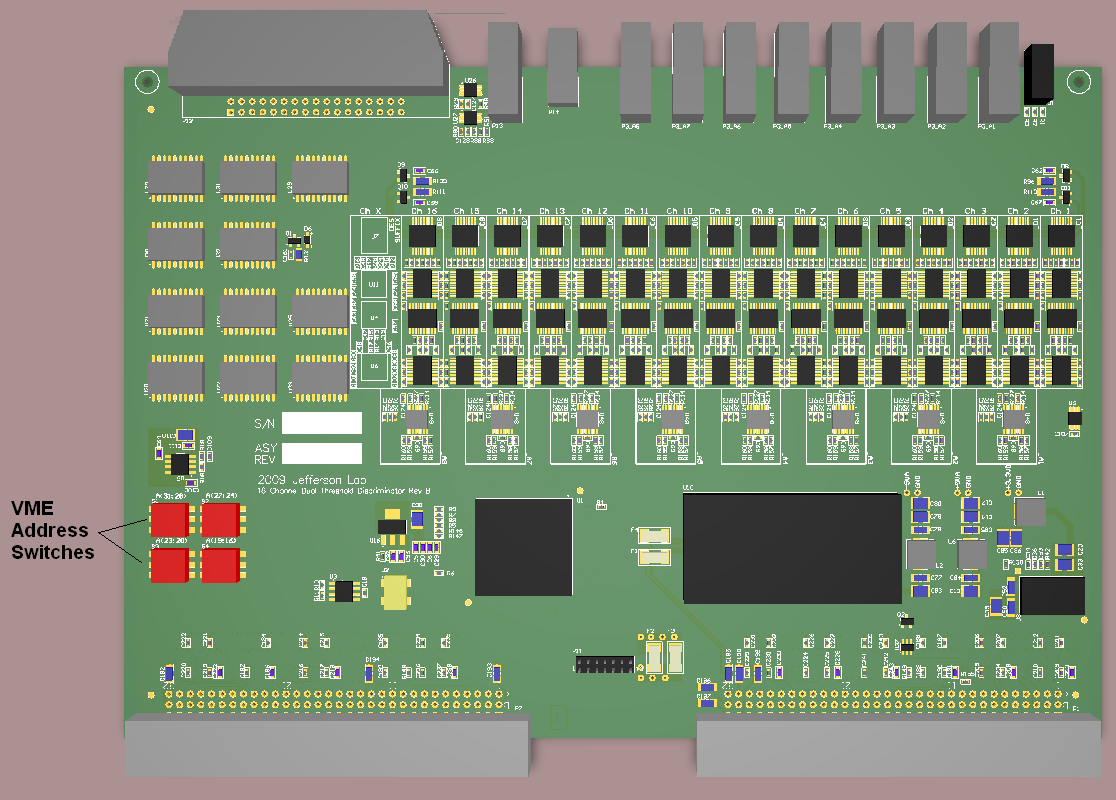
Note: due to FPGA firmware updates, the above block diagram may not fully reflect all of the available functionality of the module. See the register description section below for updated details.

1. **Specifications**

|  |  |  |
| --- | --- | --- |
| General | Spec | |
| Power consumption | +/-12v, 500mA; +5v, 5.0A (30W typ.) | |
| Fuses | +/-12v, 1.0A; +5v, 10.0A | |
| Dimensions | 6U VME, Single-wide; 160mm card depth | |
| Front Panel I/O | Input Signals: 16 LEMO  Gate Input: 1 LEMO  Test Input: 1 LEMO  Dual dECL Output: 2x34 Header  OR Output: 1 LEMO | |
| Onboard connectors | JTAG: 2x7pin 2mm Xilinx | |
| LEDs | Green: VME Activity/Power  Yellow: TDC Discriminator OR  Red: TRG Discriminator OR | |
|  |  | |
| Analog Inputs | From PMT or coaxial detector signals | |
| Channels | 16 | |
| Signal Level | +/-1.5v, DC-coupled, clamped | |
| Termination | 50ohm | |
|  |  | |
| Gate Input | Gates scalers (NIM, 50ohm termination jumper selectable) | |
| Test Input | Pulses Discriminator Outputs (NIM, 50ohm termination jumper selectable) | |
| OR Output | 1 (NIM) | |
|  |  | |
| Discriminator Channels |  | |
| Dual threshold control | 0 to -1023mV Threshold  (for each TDC and TRG output) | |
| Pulser | Non-updating | |
| Pulser Width control | 4ns to 40ns width +/-1ns accuracy | |
| Pulser dead-time | ~4ns w/8ns Pulse Width, ~10ns w/40ns Pulse Width | |
| Maximum rate | 80MHz w/8ns pulse setting | |
| Channel-Channel Crosstalk | >65dB Isolation | |
| Input Hysteresis | ~5mV | |
| Input Noise band | <2mV RMS, 1.3mV RMS typ. | |
| Offset Error | <3mV max, <1mV typ. | |
|  |  | |
| dECL Outputs |  | |
| Channels | Dual 16 channel output | |
| Connector | 34pin header in LeCroy ECL format | |
| 1st group of 16 (TDC output) | Fast discriminator output | |
|  | Common pulser width: 4 to 40ns | |
|  | Programmable mask register | |
| 2nd group of 16 (TRG output) | Common pulser width 4 to 40ns  Individual digital reshaped pulse width 4ns to 1us  Individual digital delay: 4ns to 1us | |
|  | Programmable mask register | |
| Channel Threshold Control | 10bit 1mV step (0 to -1023mV, +2048mV to -2047mV with firmware update) | |
|  |  | |
| Digital Delays | Trigger Out | Scaler/Gate |
| Delay step size | 4ns | 8ns |
| Delay range | 0 to 1020ns | 0 to 4086ns |
| Uncertainty | 4ns | 8ns |
| Input/Gate timing alignment | With 20ns | |
|  |  | |
| Scalers |  | |
| Quantity | 2 per threshold | |
| Width | 32bit | |
| Input source | Digital delay | |
| Gating | External, internal, & free run scalers | |
| Maximum Count rate | 125MHz | |
| Readout dead-time | None | |
| Control | VME latch, read, clear, overflow, event build | |
|  |  | |
| VME Interface |  | |
| Protocols | A32/A24,D32/D64/BLT32/BLT64/2eVME/2eSST | |
| Address space | 64kbyte | |
|  |  | |
| Misc |  | |
| EEPROM | 2Mbyte | |
| Firmware Upgradable | Using VME | |
|  |  | |
| Delays |  | |
| Input -> TDC Output | <6ns, <4.5ns typ. | |
| Input-> TRG Output | 15ns typ. | |
|  |  | |

1. **PCB Overview**

FR406 substrate, 1/16” Thickness, 8 Layers (4 signal, 4 plane)



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Front Panel** | **dECL Output Connector J1**  (To TDC)   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Pin | **Function** |  | **Pin** | **Function** | |  |  |  |  |  | | 1 | Ch 1 + |  | 2 | Ch 1 - | | 3 | Ch 2 + |  | 4 | Ch 2 - | | 5 | Ch 3 + |  | 6 | Ch 3 - | | 7 | Ch 4 + |  | 80 | Ch 4 - | | 9 | Ch 5 + |  | 10 | Ch 5 - | | 11 | Ch 6 + |  | 12 | Ch 6 - | | 13 | Ch 7 + |  | 14 | Ch 7 - | | 15 | Ch 8 + |  | 16 | Ch 8 - | | 17 | Ch 9 + |  | 18 | Ch 9 - | | 19 | Ch 10 + |  | 20 | Ch 10 - | | 21 | Ch 11 + |  | 22 | Ch 11 - | | 23 | Ch 12 + |  | 24 | Ch 12 - | | 25 | Ch 13 + |  | 26 | Ch 13 - | | 27 | Ch 14 + |  | 28 | Ch 14 - | | 29 | Ch 15 + |  | 30 | Ch 15 - | | 31 | Ch 16 + |  | 32 | Ch 16 - | | 33 | No Connect |  | 34 | No Connect |   **dECL Output Connector J2**  (To Trigger Input)   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Pin | **Function** |  | **Pin** | **Function** | |  |  |  |  |  | | 1 | Ch 1 + |  | 2 | Ch 1 - | | 3 | Ch 2 + |  | 4 | Ch 2 - | | 5 | Ch 3 + |  | 6 | Ch 3 - | | 7 | Ch 4 + |  | 80 | Ch 4 - | | 9 | Ch 5 + |  | 10 | Ch 5 - | | 11 | Ch 6 + |  | 12 | Ch 6 - | | 13 | Ch 7 + |  | 14 | Ch 7 - | | 15 | Ch 8 + |  | 16 | Ch 8 - | | 17 | Ch 9 + |  | 18 | Ch 9 - | | 19 | Ch 10 + |  | 20 | Ch 10 - | | 21 | Ch 11 + |  | 22 | Ch 11 - | | 23 | Ch 12 + |  | 24 | Ch 12 - | | 25 | Ch 13 + |  | 26 | Ch 13 - | | 27 | Ch 14 + |  | 28 | Ch 14 - | | 29 | Ch 15 + |  | 30 | Ch 15 - | | 31 | Ch 16 + |  | 32 | Ch 16 - | | 33 | No Connect |  | 34 | No Connect | |

1. **VME Accessible Registers**

All discriminator board registers can be accessed through the VME bus in the following modes:

* 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:

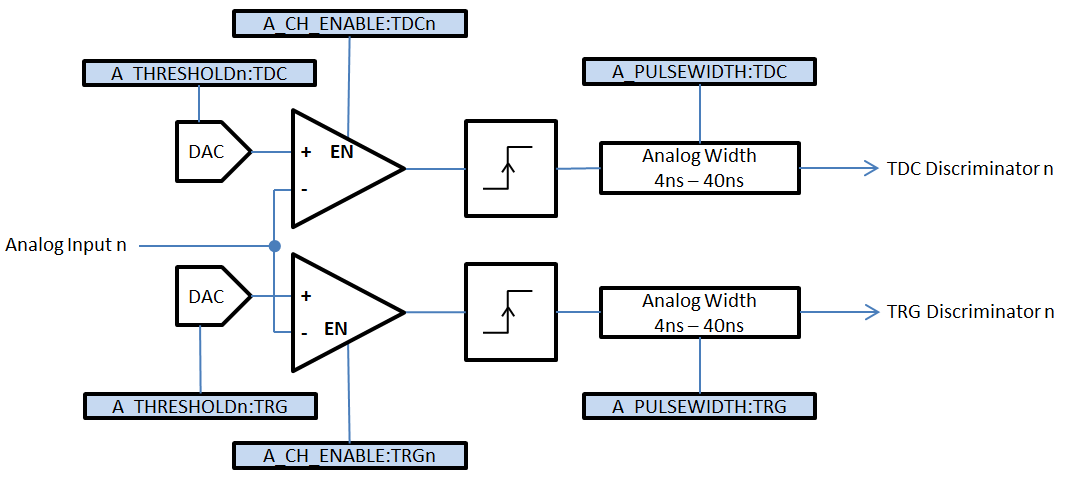
* A32: single cycle, BLT, MBLT, 2eVME, 2eSST
* Note: transfer rate for 2eSST is 200MB/s

**Register Summary:**

|  |  |  |
| --- | --- | --- |
| **Register** | **Description:** | **Address Offset-Range:** |
| **A\_THRESHOLD\_CH0** | Threshold Control Ch0 | 0x0000 |
| **…** | Threshold Control ChX | … |
| **A\_THRESHOLD\_CH15** | Threshold Control Ch15 | 0x003C |
| **A\_TRGOUT\_CH0** | Trigger Out Control Ch0 | 0x0040 |
| **…** | Trigger Out Control ChX | … |
| **A\_TRGOUT\_CH15** | Trigger Out Control Ch15 | 0x007C |
| **A\_PULSEWIDTH** | Pulse Width Control | 0x0080 |
| **A\_CH\_ENABLE** | Channel Control | 0x0088 |
| **A\_OR\_MASK** | OR Output Control | 0x008C |
| **A\_DELAY** | Input/Output Delays | 0x0090 |
| **A\_TEST** | Test Input Control | 0x0094 |
| **A\_SCALER\_LATCH\_GRP1** | Group 1 Scaler Latch | 0x009C |
| **A\_SCALER\_LATCH\_GRP2** | Group 2 Scaler Latch | 0x0098 |
| **A\_SCALER\_GATE\_GRP1** | Group 1 Scaler Gate Control | 0x00BC |
| **A\_SCALER\_GATE\_GRP2** | Group 2 Scaler Gate Control | 0x00B8 |
| **A\_ADR32** | A32 VME Base Address | 0x00A4 |
| **A\_INTERRUPT** | VME Interrupt Control | 0x00A8 |
| **A\_INTERRUPT\_ACK** | VME Interrupt Ack | 0x00AC |
| **A\_GEO** | VME Geographic Address | 0x00B0 |
| **A\_IO** | Input/Output Control/Status | 0x00B4 |
| **A\_PULSER\_PERIOD** | Pulser Period Ticks | 0x00C0 |
| **A\_PULSER\_LOW** | Pulser Low Ticks | 0x00C4 |
| **A\_PULSER\_NPULSES** | Pulser Pulse Count | 0x00C8 |
| **A\_PULSER\_START** | Pulser Start | 0x00CC |
| **A\_PULSER\_STATUS** | Pulser Status | 0x00D0 |
| **A\_SERIAL\_NUM** | Assembly Serial Number | 0x0408 |
| **A\_SERIAL\_MFG** | Assembly Manufacturer | 0x040C |
| **A\_TRG\_SCALER\_GRP1\_CH0** | Group 1 Trigger Scaler Ch0 | 0x0100 |
| **…** | Group 1 Trigger Scaler ChX | … |
| **A\_TRG\_SCALER\_ GRP1\_CH15** | Group 1 Trigger Scaler Ch15 | 0x013C |
| **A\_TDC\_SCALER\_GRP1\_CH0** | Group 1 TDC Scaler Ch0 | 0x0140 |
| **…** | Group 1 TDC Scaler ChX | … |
| **A\_TDC\_SCALER\_GRP1\_CH15** | Group 1 TDC Scaler Ch15 | 0x017C |
| **A\_TRG\_SCALER\_GRP2\_CH0** | Group 2 Trigger Scaler Ch0 | 0x0180 |
| **…** | Group 2 Trigger Scaler ChX | … |
| **A\_TRG\_SCALER\_GRP2\_CH15** | Group 2 Trigger Scaler Ch15 | 0x01BC |
| **A\_TDC\_SCALER\_GRP2\_CH0** | Group 2 TDC Scaler Ch0 | 0x01C0 |
| **…** | Group 2 TDC Scaler ChX | … |
| **A\_TDC\_SCALER\_GRP2\_CH15** | Group 2 TDC Scaler Ch15 | 0x01FC |
| **A\_REF\_SCALER\_GRP1** | Group 1 Ref Scaler | 0x0204 |
| **A\_REF\_SCALER\_GRP2** | Group 2 Ref Scaler | 0x0200 |
| **A\_FIRMWARE\_REV** | Firmware Revision | 0x0400 |
| **A\_BOARDID** | Board Identifier | 0x0404 |
| **A\_READOUT\_CLEAR** | Clear Event Builder FIFO | 0x0500 |
| **A\_READOUT\_START** | Trigger Event Builder | 0x0504 |
| **A\_READOUT\_CFG** | Event Builder Config | 0x0508 |
| **A\_READOUT\_PULSER** | Event Builder Pulser | 0x050C |
| **A\_MEM\_ARRAY** | Embedded CPU Shared Memory | 0x8000-0x87FF |
| **A\_MEM\_EXECUTE** | Notify Embedded CPU | 0x9000 |

**Analog Discriminator Control Registers**

The 16 analog front panel inputs are fed into individual discriminator/pulser circuits as shown in the following figure. The registers are shown that control the features of these channels. Note that the TDC pulse width and TRG pulse width is common to all channels of the discriminator board, while all thresholds and enable masks are individually controllable for each channel.



**Register: A\_THREHOLD\_CH0 -> A\_THRESHOLD\_CH15**

Address Offset: 0x0000, 0x0004, …0x003C

Size: 32bits

Reset State: 0x03FF000A

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

**TDC Threshold (R/W):**

TDC CHx Threshold (in -1mV units)

**TRG Threshold (R/W):**

TRG CHx Threshold (in -1mV units)

**Notes:**

1. TRG threshold should be >25mV above TDC threshold (for same channel) to avoid introducing jitter onto timing sensitive TDC comparator. If same threshold for both outputs are desired, see A\_TRGOUT\_CHx registers to route TDC output to TRGOUT signals and set TRG thresholds out of range so they do not fire.

**Register: A\_PULSEWIDTH**

Address Offset: 0x0080

Size: 32bits

Reset State: 0x00280028

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

**TDC Pulser Width (R/W):**

Controls pulser width (in units ns) for all TDC channels. Will be calibrated from 4ns to 40ns (~1ns accuracy). Values outside this range are not guaranteed to be calibrated.

**TRG Pulser Width (R/W):**

Controls pulser width (in units ns) for all TRG channels. Will be calibrated from 4ns to 40ns (~1ns accuracy). Values outside this range are not guaranteed to work.

**Register: A\_CH\_ENABLE**

Address Offset: 0x0088

Size: 32bits

Reset State: 0xFFFFFFFF

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| TRG15EN | TRG14EN | TRG13EN | TRG12EN | TRG11EN | TRG10EN | TRG9EN | TRG8EN |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRG7EN | TRG6EN | TRG5EN | TRG4EN | TRG3EN | TRG2EN | TRG1EN | TRG0EN |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TDC15EN | TDC14EN | TDC13EN | TDC12EN | TDC11EN | TDC10EN | TDC9EN | TDC8EN |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TDC7EN | TDC6EN | TDC5EN | TDC4EN | TDC3EN | TDC2EN | TDC1EN | TDC0EN |

**TDCENx (R/W):**

‘0’ – TDC channel x disabled

‘1’ – TDC channel x enabled

**TRGENx (R/W):**

‘0’ – TRG channel x disabled

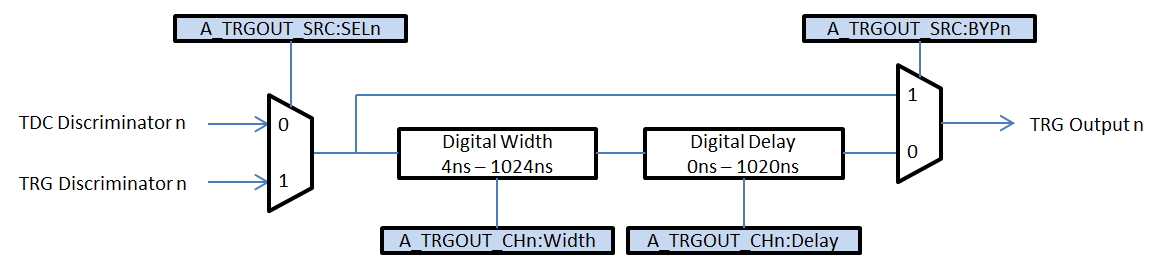
‘1’ – TRG channel x enabled

**Notes:**

1. A disabled TDC or TRG channel will prevent discriminator comparator and pulser from firing by using out of range thresholds.

**Trigger Output Control Registers**

There are 32 ECL outputs from the discriminator. The first 16 are grouped into 1 connector and are directly fed from the TDC threshold discriminators. These 16 TDC outputs are only programmable at the discriminator channel level (threshold, enable mask, analog formed pulse width). The second group of 16 channels come from the FPGA and has several programmable features as outlined in the following figure. Each bit of the 16 channel output corresponds to a specific discriminator channel (TRG Output bit ‘n’ comes from Discriminator channel ‘n’). The TRG output can select which discriminator threshold to use: TRG or TDC. The selected source can then be fed directly to the TRG output to minimize jitter and delay. Alternatively the selected source can pass through a digital pulse width block and digital delay block. By passing through the digital width/delay sections a ~4ns jitter and additional propagation delay are added to the TRG output. Each of the 16bits are separately controllable.

****

**Register: A\_TRGOUT\_CH0 -> A\_ TRGOUT \_CH15**

Address Offset: 0x0040, 0x0044, …0x007C

Size: 32bits

Reset State: 0x03FF000A

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

**TRGOUT Width (R/W):**

Pulse width in 4ns steps (0-255): width = 4ns \* (TRGOUTWidth+1)

**TRGOUT Delay (R/W):**

Pulse delay in 4ns steps (0-255): delay = 4ns \* (TRGOUTDelay)

**Notes:**

1. A\_TRGOUT\_SRC register allows option to bypass the width reshaping & delay elements. In this case, the pulse width will be defined by A\_PULSEWIDTH register and the delay will be as fast as possible (<15ns)

**Register: A\_TRGOUT\_SRC**

Address Offset: 0x00A0

Size: 32bits

Reset State: 0x0000FFFF

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| BYP15 | BYP14 | BYP13 | BYP12 | BYP11 | BYP10 | BYP9 | BYP8 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| BYP7 | BYP6 | BYP5 | BYP4 | BYP3 | BYP2 | BYP1 | BYP0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SEL15 | SEL14 | SEL13 | SEL12 | SEL11 | SEL10 | SEL9 | SEL8 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEL7 | SEL6 | SEL5 | SEL4 | SEL3 | SEL2 | SEL1 | SEL0 |

**BYPx (R/W):**

‘0’ – Delay and pulse width reshaper for TRGOUT channel x defined by A\_TRGOUT\_CHx

‘1’ – Bypasses delay and pulse width reshaper for TRGOUT channel x

**SELx (R/W):**

‘0’ – Select TDC threshold for TRGOUT source

‘1’ – Select TRG threshold for TRGOUT source

**Register: A\_OR\_MASK**

Address Offset: 0x008C

Size: 32bits

Reset State: 0xFFFFFFFF

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| TRG15EN | TRG14EN | TRG13EN | TRG12EN | TRG11EN | TRG10EN | TRG9EN | TRG8EN |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| TRG7EN | TRG6EN | TRG5EN | TRG4EN | TRG3EN | TRG2EN | TRG1EN | TRG0EN |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| TDC15EN | TDC14EN | TDC13EN | TDC12EN | TDC11EN | TDC10EN | TDC9EN | TDC8EN |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TDC7EN | TDC6EN | TDC5EN | TDC4EN | TDC3EN | TDC2EN | TDC1EN | TDC0EN |

**TDCxEN (R/W):**

‘0’ – TDC channel x not used in front-panel OR output

‘1’ – TDC channel x used in front-panel OR output

**TRGxEN (R/W):**

‘0’ – TRG channel x not used in front-panel OR output

‘1’ – TRG channel x used in front-panel OR output

**Notes:**

1. All TDC channels enabled in the above MASK are used to display the TDC front-panel LED
2. All TRG channels enabled in the above MASK are used to display the TRG front-panel LED

**Register: A\_DELAY**

Address Offset: 0x0090

Size: 32bits

Reset State: 0x00080008

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| - | - | - | - | - | - | ScalerDelayGrp2 | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| ScalerDelayGrp2 | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| - | - | - | - | - | - | ScalerDelayGrp1 | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ScalerDelayGrp1 | | | | | | | |

**ScalerDelayGrp1 (R/W):**

Discriminator input delays for scaler group 1. 0-1023 count (in 8ns ticks)

**ScalerDelayGrp2 (R/W):**

Discriminator input delays for scaler group 2. 0-1023 count (in 8ns ticks)

**Register: A\_TEST**

Address Offset: 0x0094

Size: 32bits

Reset State: 0x00000001

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

**TEST (WO):**

‘0’ – Does nothing

‘1’ – Software test pulse is sent

**TestSrc (R/W):**

bit 0: ‘1’ – routes front-panel IN1 to test input of discriminator channels

bit 1: ‘1’ – routes front-panel IN2 to test input of discriminator channels

**Notes:**

1. When front-panel IN1 or IN2 is enabled, a NIM logic level ‘1’ must be supplied to front-panel test input signal to test fire the discriminator channels.

**Register: A\_SCALER\_GRP1\_LATCH**

Address Offset: 0x009C

Size: 32bits

Reset State: 0xXXXXXXXX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| SCALER\_GRP1\_LATCH | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SCALER\_GRP1\_LATCH | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SCALER\_GRP1\_LATCH | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCALER\_GRP1\_LATCH | | | | | | | |

**SCALER\_GRP1\_LATCH (WO):**

Write any value to latch scaler group 1.

**Notes:**

1. After latching scalers for readout, hardware scalers will be reset.

**Register: A\_SCALER\_GRP2\_LATCH**

Address Offset: 0x0098

Size: 32bits

Reset State: 0xXXXXXXXX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| SCALER\_GRP2\_LATCH | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SCALER\_GRP2\_LATCH | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SCALER\_GRP2\_LATCH | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCALER\_GRP2\_LATCH | | | | | | | |

**SCALER\_GRP2\_LATCH (WO):**

Write any value to latch scaler group 2.

**Notes:**

1. After latching scalers for readout, hardware scalers will be reset.

**Register: A\_SCALER\_GATE\_GRP1**

Address Offset: 0x00BC

Size: 32bits

Reset State: 0x00000004

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| - | - | - | - | GateSrc | | | |

**GateSrc (R/W):**

bit 0: ‘1’ – routes front-panel IN1 to gate for scaler group 1

bit 1: ‘1’ – routes front-panel IN2 to gate for scaler group 1

bit 2: ‘1’ – routes constant ‘1’ to gate for scaler group 1

bit 3: ‘1’ – routes pulser output to gate for scaler group 1

**Register: A\_SCALER\_GATE\_GRP2**

Address Offset: 0x00B8

Size: 32bits

Reset State: 0x00000002

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| - | - | - | - | GateSrc | | | |

**GateSrc (R/W):**

bit 0: ‘1’ – routes front-panel IN1 to gate for scaler group 2

bit 1: ‘1’ – routes front-panel IN2 to gate for scaler group 2

bit 2: ‘1’ – routes constant ‘1’ to gate for scaler group 2

bit 3: ‘1’ – routes pulser output to gate for scaler group 2

**Register: A\_READOUT\_CLEAR**

Address Offset: 0x0500

Size: 32bits

Reset State: 0xXXXXXXXX

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

**READOUT\_CLEAR (WO):**

Write any value to clear event building FIFO.

**Register: A\_READOUT\_START**

Address Offset: 0x0504

Size: 32bits

Reset State: 0xXXX0XX00

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

**BUILDER\_FLAG (R/W):**

bit 7: ‘1’ - latch scaler group 1 for event build. ‘0’ - do nothing.

bit 6: ‘1’ - latch scaler group 2 for event build. ‘0’ - do nothing.

bit 5: ‘1’ - write reference scaler group 2 for event build. ‘0’ - do nothing.

bit 4: ‘1’ - write reference scaler group 1 for event build. ‘0’ - do nothing.

bit 3: ‘1’ - write TDC scaler group 2 for event build. ‘0’ - do nothing.

bit 2: ‘1’ - write TRG scaler group 2 for event build. ‘0’ - do nothing.

bit 1: ‘1’ - write TDC scaler group 1 for event build. ‘0’ - do nothing.

bit 0: ‘1’ - write TRG scaler group 1 for event build. ‘0’ - do nothing.

**BUILDER\_TRG\_SRC (R/W):**

bit 3: ‘1’ - enables internal pulser roll-over as a trigger source

bit 2: ‘1’ - enables SWTRG VME as a trigger source

bit 1: ‘1’ - enables front-panel IN2 as a trigger source

bit 0: ‘1’ - enables front-panel IN1 as a trigger source

**SWTRG (WO):**

‘1’ – generates software trigger

‘0’ – do nothing

**Notes:**

1. When trigger edge occurs, the scaler event builder executes and fills the readout FIFO with a scaler event as defined by the BUILDER\_FLAG field

**Register: A\_PULSER\_PERIOD**

Address Offset: 0x00C0

Size: 32bits

Reset State: 0x 00000000

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

**PULSER\_PERIOD (R/W):**

Range: 0 – 4294967295 cycles.

This pulser increments every 20 ns.

**Register: A\_PULSER\_HIGH**

Address Offset: 0x00C4

Size: 32bits

Reset State: 0x 00000000

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

**PULSER\_HIGH (R/W):**

Range: 0 – 4294967295 cycles.

While pulser counter is <= PULSER\_HIGH the output of the pulser is logic ‘1’, else ‘0’

**Register: A\_PULSER\_NPULSES**

Address Offset: 0x00C8

Size: 32bits

Reset State: 0x 00000000

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

**PULSER\_NPULSES (R/W):**

0: Disables pulser

1-4294967294: Pulser will fire for this number of pulses after A\_PULSER\_START is written

4294967295: Pulser enable for continuous operation (NPULSES will be infinite)

**Register: A\_PULSER\_START**

Address Offset: 0x00CC

Size: 32bits

Reset State: 0x 00000000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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):**

Writing any value will start pulser operation if A\_PULSER\_NPULSES is operating in finite pulse count mode.

**Register: A\_PULSER\_STATUS**

Address Offset: 0x00D0

Size: 32bits

Reset State: 0x 00000000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| - | - | - | - | - | - | - | STATUS |

**STATUS (RO):**

‘1’ indicates pulser is not active

‘0’ indicates pulser is active. This can be used to check when NPULSES have been delivered after START has been issued running in finite pulse mode.

**Register: A\_READOUT\_CFG**

Address Offset: 0x0508

Size: 32bits

Reset State: 0x0001FFFE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| EVT\_NUM | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| EVT\_NUM\_INT\_LEVEL | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| EVT\_WORD\_INT\_LEVEL | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EVT\_WORD\_INT\_LEVEL | | | | | | | BERREN |

**EVT\_NUM (RO):**

Indicates the number of built events residing in the FIFO ready for readout.

**EVT\_WORD\_INT\_LEVEL (R/W):**

Range: 0 to 16383. 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 255. 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\_ADR32**

Address Offset: 0x00A4

Size: 32bits

Reset State: 0xXXXXXXX0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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)

This field is initialized to the dip switch bits 31:24 on VME SYSRESET assertion.

**A32\_EN (R/W):**

‘0’ – disables VME A32 addressing mode

‘1’ – enabled VME A32 addressing mode

**Register: A\_INTERRUPT**

Address Offset: 0x00A8

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: 0x00AC

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 |
| - | - | - | - | - | - | - | - |

**Notes:**

1. 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: 0x00B0

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 number. On parity error the value returned is 0x1E.

**Notes:**

1. Geographical addressing is only support when module is used on aVME64X compatibly crate.

**Register: A\_IO**

Address Offset: 0x00B4

Size: 32bits

Reset State: 0xXXXXXXX0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| - | - | - | - | - | IN2 | IN1 | OR\_OUT |

**OR\_OUT (R/W):**

‘0’ – software controlled ‘0’ or’d on OR front panel output

‘1’ - software controlled ‘1’ or’d on OR front panel output

**IN1 (RO):**

‘0’ – Front panel IN1 is NIM logic low

‘1’ – Front panel IN1 is NIM logic high

**IN2 (RO):**

‘0’ – Front panel IN2 is NIM logic low

‘1’ – Front panel IN2 is NIM logic high

**Register: A\_TRG\_SCALER\_GRP1\_CH0 -> A\_ TRG\_SCALER\_GRP1 \_CH15**

Address Offset: 0x0100, 0x0104, …0x013C

Size: 32bits

Reset State: 0xXXXXXXXX

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

**TRG SCALER(RO):**

Trigger threshold scaler for CHx.

Belongs to scaler group 1, which uses scaler gate 1 and latch 1 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP1\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_TDC\_SCALER\_GRP1\_CH0 -> A\_ TDC\_SCALER \_GRP1\_CH15**

Address Offset: 0x0140, 0x0144, …0x017C

Size: 32bits

Reset State: 0xXXXXXXXX

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

**TDC SCALER(RO):**

Trigger threshold scaler for CHx.

Belongs to scaler group 1, which uses scaler gate 1 and latch 1 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP1\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_TRG\_SCALER\_GRP2\_CH0 -> A\_ TRG\_ SCALER \_GRP2\_CH15**

Address Offset: 0x0180, 0x0184, …0x01BC

Size: 32bits

Reset State: 0xXXXXXXXX

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

**TRG SCALER(RO):**

Trigger threshold scaler for CHx.

Belongs to scaler group 2, which uses scaler gate 2 and latch 2 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP2\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_TDC\_SCALER\_GRP2\_CH0 -> A\_ TDC\_SCALER \_GRP2\_CH15**

Address Offset: 0x01C0, 0x01C4, …0x01FC

Size: 32bits

Reset State: 0xXXXXXXXX

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

**TDC SCALER(RO):**

Trigger threshold scaler for CHx.

Belongs to scaler group 1, which uses scaler gate 2 and latch 2 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP2\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_REF\_SCALER\_GRP1**

Address Offset: 0x0204

Size: 32bits

Reset State: 0xXXXXXXXX

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

**REF SCALER(RO):**

Reference scaler, increments at 125MHz while gate source is high.

Belongs to scaler group 1, which uses scaler gate 1 and latch 1 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP1\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_REF\_SCALER\_GRP2**

Address Offset: 0x0200

Size: 32bits

Reset State: 0xXXXXXXXX

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

**REF SCALER(RO):**

Reference scaler, increments at 125MHz while gate source is high.

Belongs to scaler group 2, which uses scaler gate 2 and latch 2 as control sources.

32bit value indicates an overflow when count is saturated (at 0xFFFFFFFF).

**Notes:**

1. A scaler latch must be performed (by writing to register A\_SCALER\_GRP2\_LATCH, or using event builder) to update these registers with current scaler counts

**Register: A\_FIRMWARE\_REV**

Address Offset: 0x0400

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

**FIRMWARE\_REV\_MINOR(RO):**

Minor firmware revision

**Register: A\_BOARDID**

Address Offset: 0x0404

Size: 32bits

Reset State: 0x44534332

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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):**

0x44534332 = “DSC2” in ASCII

**Register: A\_SERIAL\_NUM**

Address Offset: 0x0408

Size: 32bits

Reset State: 0x44534332

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

**SERIAL\_NUM(RO):**

32bit serial number

**Register: A\_SERIAL\_MFG**

Address Offset: 0x040C

Size: 32bits

Reset State: 0x44534332

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

**SERIAL\_MFG (RO):**

4 digit ASCII manufacturer ID

**Register: A\_MEM\_ARRAY**

Address Offset: 0x8000-0x87FF

Size: 32bits

**Notes:**

1. This memory is reserved for testing, calibration, and firmware upgrade use.

**Register: A\_MEM\_EXECUTE**

Address Offset: 0x9000

Size: 32bits

**Notes:**

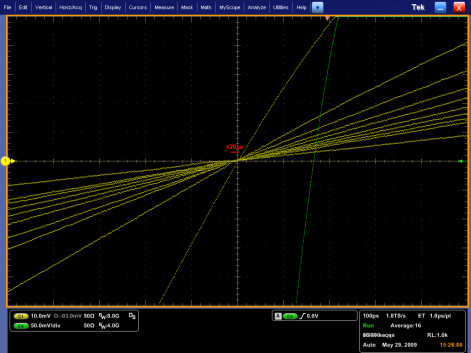
1. This register is reserved for testing, calibration, and firmware upgrade use.
2. **Module Performance (Typical)**

**Measured signal jitter:**

Input: 1Vpp, 1MHz square wave with an overdrive from 5mV to 500mV, and slew rate from 10mV/ns to 1000mV/ns. The following plot indicates measured performance.

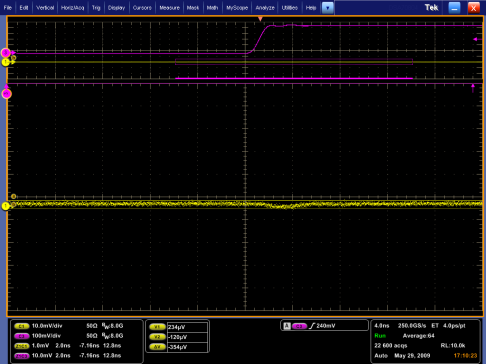
**Measured slew rate dispersion:**

356mV, 30ns width negative pulse. Overdrive set to 100mV, edge rate varied from 250mV/ns to 20mV/ns, pulse rate 100Hz. Roughly a 20ps dispersion with respect to slew rate measured.



**Measured channel isolation:**

A 1V edge with slew rate 1V/ns injected into discriminator channel. The pickup measured on adjacent channels was less than 350μV (>69dB channel-channel isolation). Yellow trace below is measured induced voltage on victim channel.



**Measured maximum rates:**

|  |  |
| --- | --- |
| 110MHz Rate w/4ns Output Pulse Width | 80MHz Rate w/8ns Output Pulse Width |
|  |  |

1. **Scaler Event Readout**

Scaler event readout is performed by writing to the **A\_READOUT\_START** register with the appropriate flags set according to the user preference and then by generating a trigger from an appropriate source (either software, internal pulser, or external input). Each time a trigger is generated the internal event builder will write to the readout FIFO with the scaler event determined by the flags set. Events can be generated until the readout FIFO not longer contains enough space for a full event to be written. This guarantees that only full events are written and never partial events. The output buffer can store multiple events. Readout is performed by accessing the modules A32 VME address using any of the support VME protocols of this board.

**Event Format:**

The event format is compatible with the Jlab DAQ group’s proposed format for Jlab DAQ modules. There are a number of redundant bit fields in the readout format for the discriminator that may be eliminated in software after readout or can also be removed in the discriminator firmware based on request.

**Data Word Categories**

Data words from a 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 raw data. Any number of Data Type Continuation words

may follow a Data Type Defining word.

EXCEPTION: In order to allow full 32-bit User payload data for specific modules, the User may

create Data Type Defining Words (for the Data Types 4 – 13) that include the specific

number of 32 bit Continuation words that follow. In this way the full 32 bits for each

Continuation word may be used (and bit 31 is not required to be 0).

**Data Type List**

0 Block Header

1 Block Trailer

2 Event Header

3 Reserved

4 Scaler Header

5 Reserved

6 Reserved

7 Reserved

8 Reserved

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 | | MODULEID | | | | | BLOCK\_NUMBER | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | | 8 |
| BLOCK\_NUMBER | | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | | 0 |
| EVENT\_COUNT | | | | | | | | |

**NUM\_EVENTS:**

Number of events in block

**BLOCK\_NUMBER:**

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

**MODULEID:**

Discriminator Module ID (0x8)

**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 | SLOTID | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| SLOTID | | 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

**SLOTID:**

Slot ID (set by VME64x backplane)

**Data Type: Scaler Header**

Type: 0x8

Size: 1+N words

Description: Scaler header. A field inside this word indicates how many 32bit scaler words are to follow.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | | 24 |
| 1 | 0 | 1 | 0 | 0 | 0 | | 0 | 0 |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | | IN2 | IN1 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | | 8 |
| BUILDER\_FLAGS | | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | | 0 |
| SCALER\_LEN | | | | | | | | |

**IN2:**

Front panel input IN2 value at time of trigger.

**IN1:**

Front panel input IN1 value at time of trigger.

**BUILDER\_FLAGS:**

These flags indicate the scaler data order that follows. Data types are prioritized in terms on the bit index: the lowest bit index in the BUILDER\_FLAGS field has the higher priority which will be reported first. If the bit is ‘0’ then scalers corresponding to that bit flag are skipped by the event builder.

Bit 0: ‘1’ - Trigger threshold scaler group 1 are reported in ascending channel order (16 scalers)

Bit 1: ‘1’ – TDC threshold scaler group 1 are reported in ascending channel order (16 scalers)

Bit 2: ‘1’ - Trigger threshold scaler group 2 are reported in ascending channel order (16 scalers)

Bit 3: ‘1’ – TDC threshold scaler group 2 are reported in ascending channel order (16 scalers)

Bit 4: ‘1’ – Scaler group 1 125MHz reference (1 scaler)

Bit 5: ‘1’ – Scaler group 2 125MHz reference (1 scaler)

Bit 6: not defined

Bit 7: not defined

**SCALER\_LEN:**

0-255: number of 32bit scaler data words to follow directly after this header. This data type is an exception to the normal “data continuation” bit 31 indicator so that full 32bit scaler data can be accommodated.

**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 | | | | | | | |

1. **Firmware Revision History**

**V1.C 7/2/2013:**

1. Initial tracked release.