Digitizer &

Trigger

Module

DTM Operation Manual

August 29, 2016

Rev. 1.1

DTM Operation Manual

Rev. 1.1

compiled by

Simon T. Norman-Hobbs

TRIUMF

Contents

List of Figures

Table of Symbols & Daqronyms

AARF Acrylic Reflector and Fiber Optics, a system of LED light injection used for detector calibration.

ASUM Analog Sum, an analog sum of the signals from 12 PMTs passed to the DTM from each of the SCBs.

ASUMSUM A digital sum of the 22 analog sums from the SCBs, used in the trigger logic.

CDM Cold Dark Matter.

DAQ Data Aquisition, the system of digitizers and backend computers recording and saving data to disk.

DEAP Dark Matter Experiment using Argon Pulse-shape discrimination.

DEAP-3600 Dark matter detector with a 3600 kg target mass of liquid Ar.

DTM Digitizer & Trigger Module, control board for the DAQ.

FIFO First in First Out buffer.

FMC FPGA Mezzanine Card, an ANSI standard which defines an expansion interface for a daughter card to an FPGA baseboard or other device with re-configurable I/O capability (See Samtec for data sheets and the standard description: https://www.samtec.com/standards/fmc).

FPGA Field Programmable Gate Array.

Fprompt Fraction of Prompt Light, the integrated charge in a short window divided by the charge from an entire event. Used in PSD to determine incident particle type.

JALISCO Java Lightweight System Console, JALISCO, is a Java GUI that controls hardware boards. Currently used in various projects including DEAP-3600, Jalisco is unique in that it automatically generates its GUI based on autodiscovery of the register files inside the firmware..

LED Light Emitting Diode, used in AARF calibration system.
LVDS Low-Voltage Differential Signalling.

MIDAS Maximum Integrated Data Acquisition System, MIDAS is a general-purpose software package for event-based data acquisition in small and medium scale Physics experiments [6].
MSPS Megasamples Per Second, denotes speed of the digitizers.

NIM Nuclear Instrument Module, standard size and interface for electronics hardware.

ODB Online Data Base, MIDAS DAQ central information hub containing all settings for the trigger. Refer to Section 3.6.

PLL Phase Lock Loop, clock frequency multiplier that hold the relative phase of the input output clocks fixed.
PMT Photomultiplier Tube.
PPG Pattern Pulse Generator, injects electronic pulses which cause the trigger to fire.
PSD Pulse Shape Discrimination.

SCB Signal Conditioning Board, takes in 12 channels of PMTs and produces a low gain channel for the V1740 digitizers, a high gain channel for th V1720s and an analog channel sum for the DTM.

ToT Time Over Threshold, requirement of the minimum bias trigger requiring a signal stay above the threshold for a set amount of time to reduce probability of triggering on noise.

V1720 CAEN V1720 12-bit, 8-channel waveform digitizing modules with a 250 MS/s sampling rate.
V1740 CAEN V1740 12-bit, 64-channel waveform digitizing modules with a 62.5 MS/s sampling rate.

VME Versa Module Europa like NIM, a standard for electronics hardware.

WIMP Weakly Interacting Massive Particle.

Chapter 1

Introduction

This user manual for the DEAP-3600 Digitizer and Trigger Module (DTM) is intended to gather all the necessary information to operate the trigger and to understand the firmware structure. Connections to the DTM and the parts necessary to explain the immediate connections are discussed, however one should refer to the appendix which gathers other collaboration sources that discuss the electronics and other components in more depth, links to the twiki are included along with various applicable links and papers that are useful to have in one place.

This manual is available on gitlab[1] if any inacurracies or out of date information are found please correct it.

[1] https://gitlab.phy.queensu.ca/shobbs/dtm-user-manual

Chapter 2

DTM Hardware

This chapter borrows heavily from the DEAP-3600 Electronics & DAQ Technical Design Report [3][1]

EDEV Project Location: `https://edev.triumf.ca/project/deap/edevel00013`

The DEAP-3600 Digitizer and Trigger Module (DTM) is a custom built 6U VME motherboard populated with three daughter mezzanine boards (shown in Fig. 2.1 and schematically in Fig. 2.2) (designed by the TRIUMF EDEV department). The main motherboard has three FPGA Mezzanine Card (FMC)[2] standard connectors which each interface with the daughter boards through a Xilinx Spartan-6 LX FPGA. The motherboard is equipped with an Altera Stratix IV GX FPGA[3] which is used as the primary processing and communication driver for the DTM. It is this primary FPGA that the triggering firmware is implemented on (see Chapter 3 for the trigger descriptions).

The DTM receives 22 analog sums (ASUMs) of 12 PMT signals from the signal conditioning boards (SCBs) which are digitized by the ADC mezzanine board (Section 2.2.1). The DTM is constantly integrating the ASUMs and uses Fprompt and charge to make a triggering decision. If the triggering requirements are met (as discussed in Chapter 3) the DTM will send a trigger command through the I/O board (Section 2.2.4) to the first in a

[1] `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/TechnicalDesignReport`

[2] See Samtec for data sheets and the standard description: `https://www.samtec.com/standards/fmc`

[3] `https://www.altera.com/products/fpga/stratix-series/stratix-iv/overview.html`

chain of digitizers which passes the trigger along to the rest in a daisy-chain arrangement (see Fig. 2.7).

Two mezzanine board configurations have been used and are discussed in Section 2.4. The four types of mezzanine boards used are discussed in this chapter, information pertaining to their associated firmware releases can be found in Chapter 4.

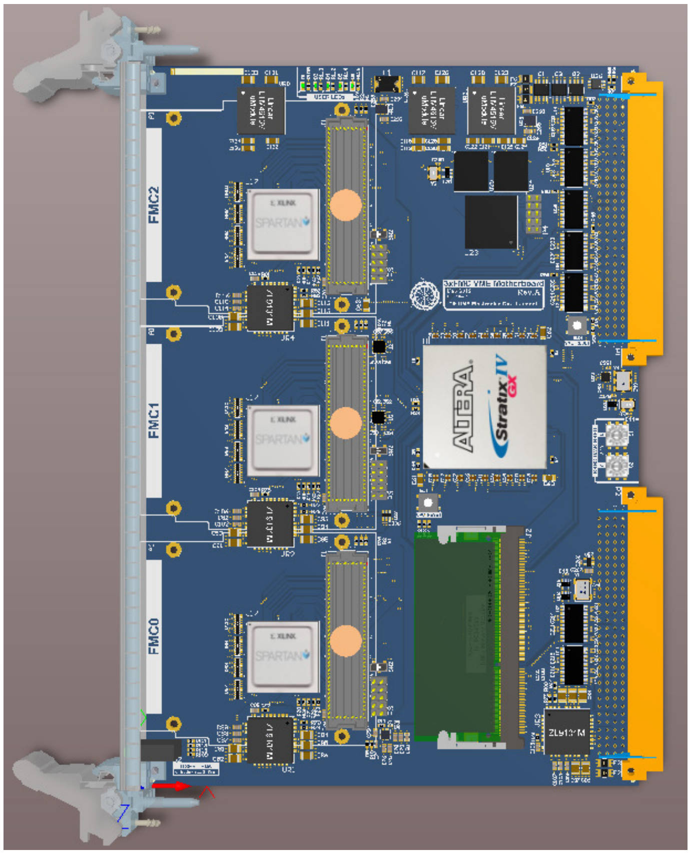


Figure 2.1: The digitizer and trigger module motherboard. The three FMC slots are labelled to the left of the board. Digital control is implemented on an Altera Stratix IV GX FPGA while interfacing with the FMCs are controlled by the three Spartan-6 FPGAs.

2.1 VME 3xFMC Carrier Motherboard

EDEV Project Location: `https://edev.triumf.ca/svn/edevel00052`

Most modern experiments have many common characteristics when it comes to data acquisition, processing, and triggering. These mixed signal systems typically use some com-

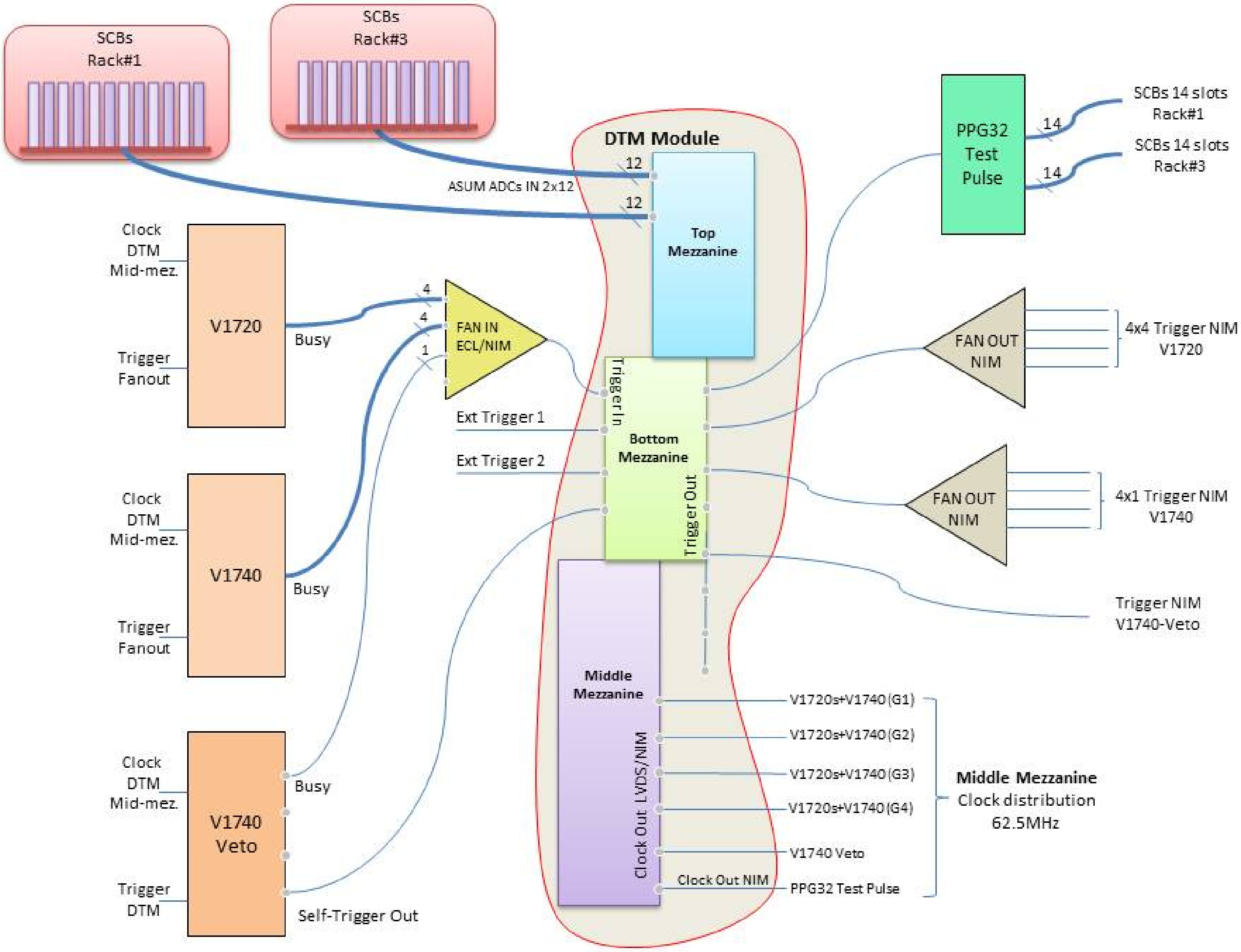


Figure 2.2: Diagram of the information and signal flow in the DTM. The clock generator mezzanine board is shown as being used although this is subject to change (see Section 2.4 for more on the configurations)

bination of fast ADCs and general purpose I/O to capture and digitize any relevant data which is then piped into an FPGA for processing. Various data suppression algorithms and triggering schemes are implemented before the final events of interest are displayed or written to disk. Large memory buffers are often necessary to prevent data loss and maximize the rate at which data can be acquired without overloading the processing capabilities of the FPGA or associated communication channels. The VME 3xFMC motherboard was conceived as a way to make better use of development resources. It was designed to meet these general needs while providing the flexibility to rapidly adapt to more specific project requirements.

An Altera Stratix IV GX FPGA (EP4SGX230KF40C2N)[4] serves as the main processing and communication hub for the motherboard module. The VME interface for the motherboard is provided by the Stratix IV GX FPGA. With the correct firmware, the motherboard fully supports all VME data and addressing modes and may be implemented as most types of VME module (Master, Slave, Arbiter, etc.). To support high speed ADCs and back-end communications (outside of VME), this FPGA has 28 dedicated gigabit serial links going to the FMC mezzanines (10, 10, and 8). Each link is capable of operating at data rates up to 8.5 Gbps.

Additional Xilinx Spartan-6 LX FPGAs (XC6SLX45-2FGG484I)[5] are used for each of the three FMCs providing support for bidirectional differential signalling (required by the FMC specification). The bidirectional differential signalling allows maximum compatibility for both custom and commercially available FMC modules. Each FMC interface on the carrier motherboard supports both single ended and differential I/O on banks LA and HA (except HA23), and single ended I/O and input only differential on bank HB (and HA23). This configuration gives a maximum per FMC of 80 differential pairs (23 input only), 160 single ended I/O, or any combination thereof. Although the Spartan-6 LX FPGAs are mainly used for I/O interfacing, they also provide resources for additional data buffering, delay matching, and additional processing power.

The carrier motherboard provides a very flexible clocking scheme. The VME interface also provides a 16 MHz system clock. On board oscillators provide fixed 16 MHz and 500 MHz clocks while a programmable oscillator (defaulting to 125 MHz) allows additional flexibility and a dedicated DDR3 memory interface reference clock. Each FMC module may provide up to two dedicated gigabit transceiver reference clocks and two dedicated clock inputs which allow the option for mezzanine mounted oscillators or external clock inputs of any frequency. All of these clock sources use dedicated PLL inputs on the Stratix IV GX and may be multiplied to any synthesizable frequency for driving internal logic or a wide variety of clock outputs.

[4] `https://www.altera.com/products/fpga/stratix-series/stratix-iv/overview.html`

[5] `http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf`

To meet the data buffering and general computing memory requirements, the motherboard includes a 204 pin DDR3 SO-DIMM interface (laptop memory). The memory controller on the Stratix can supports up to 2 GB with a memory clock up to 533 MHz. This translates into a peak transfer rate of approximately 8.5 GB/s. For configuration management, an Altera MaxV CPLD (5M2210ZF256C5N)[6] and 1Gb (2 x 512 Mb) of NOR flash memory is provided. The flash memory is directly accessible from the Stratix to allow for self programming and reconfiguration through VME or other communication interfaces. Uncompressed bitstreams for the Stratix IV GX and Spartan-6 FPGAs are 94,557,472 and 11,939,296 bits respectively so the provided flash allows support for several configurations for each with enough left over for some user data. While the CPLD handles the configuration of the Stratix, the configuration of the Spartan-6 FMC interface FPGAs is managed by the main Stratix IV GX FPGA. This scheme allows for FMC module identification and consequent reprogramming of the interface FPGAs.

Power management on the motherboard has also been implemented to allow maximum flexibility. The power design supports use in VME crates with or without the 3.3V supply provisioned in the VME64X spec[7]. While all supply voltages are provided regardless of the crate used, additional power is available for the individual FMC modules when the 3.3V supply is present. Each FMC slot is supplied with a 12V (<1A) and 3.3V (<3A) fixed supplies and a dedicated 0-3.3V adjustable supply, VADJ (<4A). In return, the FMC modules provide the carrier with I/O voltage for bank B and reference voltage for banks A and B if voltage referenced I/O standards are used. All of the power supplies are monitored and controlled from the Stratix IV GX FPGA on the carrier motherboard.

The VME 3xFMC Carrier Motherboard is a very flexible and capable card, however, most of the actual functionality comes from the FMC modules that are physically plugged in whether custom built or commercially available.

[6] `https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/max-v/max5_handbook.pdf`

[7] For the standard see: `http://file.wiener-d.com/documentation/General/WIENER_VME_VXI_VXS_introduction_1.0.pdf`

2.1.1 Motherboard Components

The major components of the DTM consist of the following:

(1) Altera Max V CPLD

On power-up configures the Stratix IV FPGA, and allows JTAG programming of both the Stratix IV and the Flash memory. Also performs watchdog on Stratix IV.

(1) Altera Stratix IV FPGA

Core of the DTM. Performs VME transfers, runs NIOSII embedded processors, trigger system, and configures the Spartan-6s. Only FPGA with direct access to DDR3 SDRAM, LEDs, switches, and VME. Shares bus to Flash memory with Max V.

(1) DDR3 SDRAM DIMM

Used for ADC data storage

(2) 512 Mb Flash Memory Chips

Contain configuration files for Stratix IV and Spartan-6 FPGAs.

(3) Xilinx Spartan-6 LX FPGAs

Used to decode ADC, initialize the clock module, and connect the FMCs I/O to the Stratix IV.

(3) FMC connectors

Where the mezzanine cards go.

(9) User-controlled LEDs

Diagnostics and status

(2) User-controlled 16 position switches

Set VME address of DTM

(1) VME64 compatible backplane

Communication link of DTM

2.1.2 Board Revisions

There have been two revisions of this motherboard to date, Rev.A and Rev.B[8]. The primary difference between these revisions is only the addition of a SD card on Rev.B. Although the hardware is essentially identical there were issues that arose which meant that the firmware release edevel00268 only worked on Rev.B boards although this issue has been fixed in the newer edevel00365 releases and the boards can now be used interchangeably (See Chapter 4 and 7).

2.1.3 Board Settings & Registers

2.1.3.1 VME Address

Twiki Location: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/Dtmboard`

The motherboard is equipped with two rotary switches for setting the VME base address. Default setting is 0x69 = sw2[7-4]: 0110, sw1[3-0]: 1001. **Bit 0**, the boot bit, **must always be on**. Bit[7-4][3-0] → 0110 1001 → 0xB(3,7-5) corresponding to the upper 4 bits of the A24: 0xB00000 (Rotary switch: 0x69), bit[7-4][3-0] → 1000 1001 → 0xC(3,7..5) corresponding to the upper 4 bits of the A24: 0xC00000 (Rotary switch: 0xC9).

The VME base address has to be adjusted correspondingly in the frontend or access code (deap/pro/FrontEnd/dtm/vme_register.h (DTM_BASE)). No two VME modules being used can use the same base address.

2.1.3.2 DTM Registers

Twiki Location: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/Dtmregisters`

In general, there exist two type of registers: ReadWrite (RW) and ReadOnly (RO). The first type is used to set parameters through ODB (see Section 3.6), the second one to read the status and data of the different DTM firmware modules. The last two bits of the address are not used, therefore the distance between registers is 0x4 in address space.

[8] See the errata for firmware compatibility issues (Chapter 7)

The exact register address is given by:

DTM_BASE + (RW/RO/USER)_OFFSET + REG_OFFSET

Reading of these registers can be done over JALISCO (see Section 5.1) or using VME command line commands using vme_read.exe -b 0xb00000 (which is the dtm base) -a (register address)[9].

Note: The registers in JALISCO are displayed in 32 bit words, so the JALISCO offset is the register table offset/4.

2.2 Mezzanine Boards & DTM Connections

The following describes the different FMC boards used and connections in the various DTM configurations. The different configurations themselves are discussed in Section 2.4 with their associated firmware in Chapter 4.

2.2.1 24 Channel ADC FMC

SVN Project Location: `https://edev.triumf.ca/svn/edevel00018`

SVN Firmware Location: `https://edev.triumf.ca/svn/edevel00230`

Hardware Connections: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/HWconnection#FMCADC`

The ADC (or Digitizer) board, seen in Fig. 2.3 is populated with three eight channel 12-bit 45 MSPS $\Sigma - \Delta$ ADCs (ADC12EU050)[10]. The DTM receives 22 analog sums (ASUMs) of 12 PMT signals from the SCBs which are digitized by the ADC mezzanine board[11]. These digitized ASUMs are used to make the trigger decision as discussed in Chapter 3.

The SCB ASUM output signals are received on differential lines by the ADC. The differential common mode is 0.6 V. To match the ADC ±1.05 V dynamic range the SCB

[9] See the twiki for how to run this:
`https://www.snolab.ca/deap/private/TWiki/bin/view/Main/UsingTriumfDtmTestSetup`

[10] Data sheet: `http://www.ti.com/lit/ds/symlink/adc12eu050.pdf`

[11] Have a look at the DEAP-3600 technical manual or the DEAP electronics paper

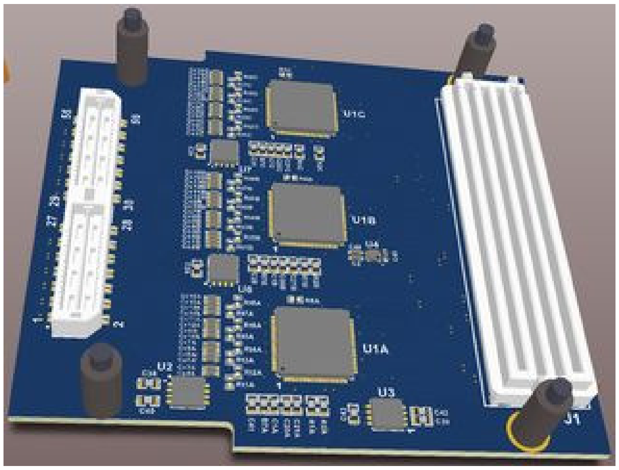


Figure 2.3: 24 Channel digitizer board which receives 22 analog sums of 12 PMTs each and passes the digitized result to the main Stratix IV FPGA for a trigger decision to be made.

applies a constant offset of 0.8 V to the ASUM signal. The ADC are supplied a 45.161290 MHz clock by a PLL on the DTM[12].

Note: The PLL is set to 45.25 MHz but the 45.161290 MHz is the fastest it can run, hence the odd rate.

To ensure a low jitter clock for sampling and data transfer off the ADC, this clock is first passed through a clean-up PLL integrated into the ADC. The 12-bit data sampled by the ADC at 45.161290 MHz is transferred to the Spartan-6 FPGA as serial DDR data.To enable the correct re-alignment of the 12-bit data word in the FPGA a word clock is also provided by the ADC. Control of the ADC is performed via a serial to parallel interface (SPI) command protocol. To test data alignment the ADC provides default test patterns and a user defined test pattern. The test patterns are enabled and in the case of the user test pattern defined over the SPI command structure. JALISCO includes tools for reading and debugging ADC's, refer to Section 5.1.

Configuration Notes: Unchanged board used in all configurations of the DTM and for firmware modules edevel00268 and edevel00365.

[12] Other documentation reports a 50 MHz clock to the ADCs, that is out of date. This is the maximum allowed speed of the ADCs.

2.2.2 DEAP Clock Generator Module

SVN Project Location: `https://edev.triumf.ca/svn/edevel00114`

SVN Firmware Location: `https://edev.triumf.ca/svn/edevel00230`

Hardware Connections: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/HWconnection#FMCCLOCK`

The clock generator board distributes an 62.5 MHz clock generated in the Stratix IV to the rest of the DAQ system via a daisy chain arrangement. These are distributed as: four clock signals to the four V1720 sectors; one clock signal to the V1740 sector; and one clock signal to the PPG module. The 62.5 MHz clock distribution to the four V1720 sectors, the V1740 sector and the PPG as discussed in Section 2.3.1. Within the V1720 and V1740 sectors the clock signal is daisy chained module to module (see Section 2.3).

The clock generator FMC has seven LVDS and three NIM I/O ports. Five LVDS are used to connect to each of the four clock groups and the VETO V1740s. One NIM port is used for the PPG clock. The remaining ports are unassigned general I/O[13].

Configuration Notes: Board used in all releases of firmware module edevel00268, replaced with the SFP and Mini-SAS board in mezzanine (See Section 2.2.3). Last stable release installed at SNOLAB using this board edevel00268/tags/t_a_release_206. See Section 2.4 for the different DTM configurations, and Section 2.3 for more on the DTM and DAQ clocking system.

2.2.3 SFP and Mini-SAS Interface

SVN Project Location: `https://edev.triumf.ca/svn/edevel00228`

SVN Firmware Location: `https://edev.triumf.ca/svn/edevel00230`

This module is equipped with:

1. (1) SFP connector allowing gigabit ethernet

[13] See `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/HWconnection`

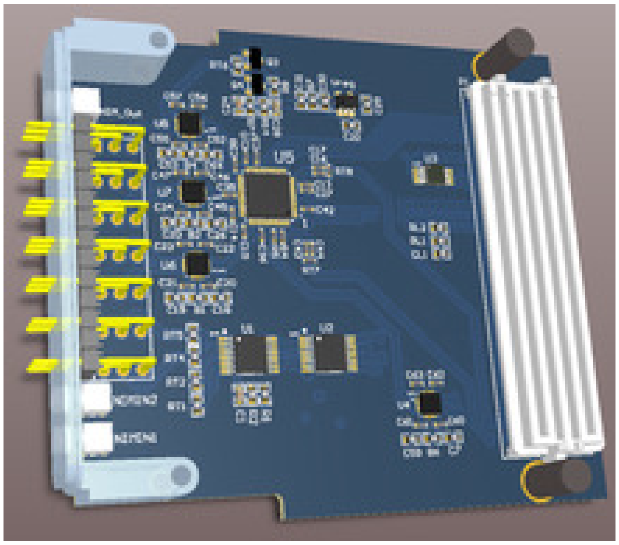


Figure 2.4: Seven LVDS and three NIM I/O ports configured to provide the master clock to the rest of the DAQ system in a daisy chain clocking configuration (see Section 2.3.1).

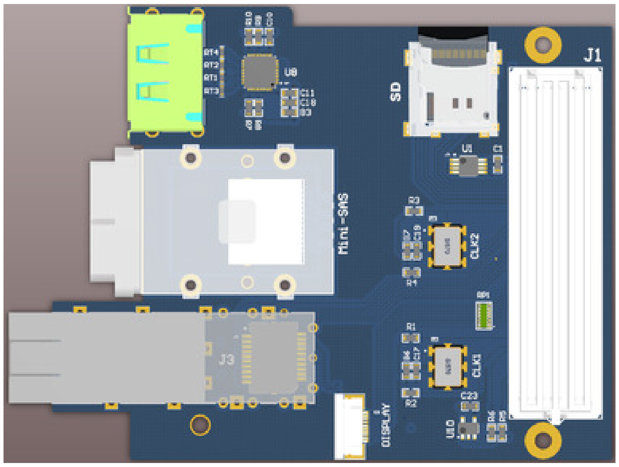


Figure 2.5: SFP ethernet and Mini-SAS interface board which adds FTP readout and flash programming interfacing with the DTM (see Chapter 5)

2. (1) Mini-SAS connector

3. (1) eSATA connector (can be configured as a clock in)

4. (1) Micro-SD card

The SFP and Mini-SAS Interface FMC has been designed for the DEAP-3600 experiment to add ethernet networking and data storage on the on-board micro-SD card, see Chapter 5 for interfacing and features.

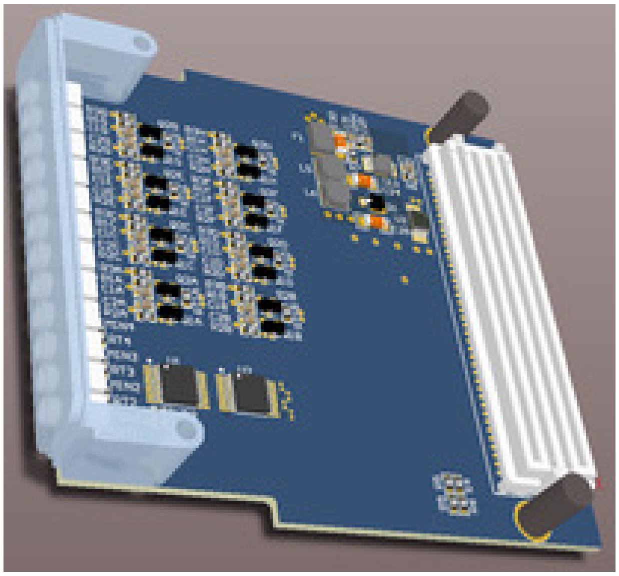


Figure 2.6: 12 Channel NIM I/O board used for controlling the rest of the DAQ system including the output of the trigger signal and busy inputs. In one DTM configuration this board also outputs the master clock (see Section 2.4).

2.2.4 DEAP NIM I/O Trigger Module

SVN Project Location: `https://edev.triumf.ca/svn/edevel00019`

SVN Firmware Location: `https://edev.triumf.ca/svn/edevel00279`

Hardware Connections: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/HWconnection#FMCIO`

The NIM I/O board is a twelve channel NIM board used for trigger output, DAQ control/status, calibration systems control, and in one configuration a master clock output[14].

The twelve channels are split between eight dedicated outputs and four dedicated inputs. The V1720 and V1740 can both accept NIM or TTL logic [1][2], for compatibility with the PPG[5] NIM logic is chosen. The V1720 and V1740 boards specify NIM lemo-00 50Ω trigger input connections, the PPG[5] specifies NIM lemo-00 inputs. Due to space limitations the I/O board is populated with mmcx connectors[15].

[14] See the configuration notes and Section 2.4

[15] These are fragile and very prone to breaking

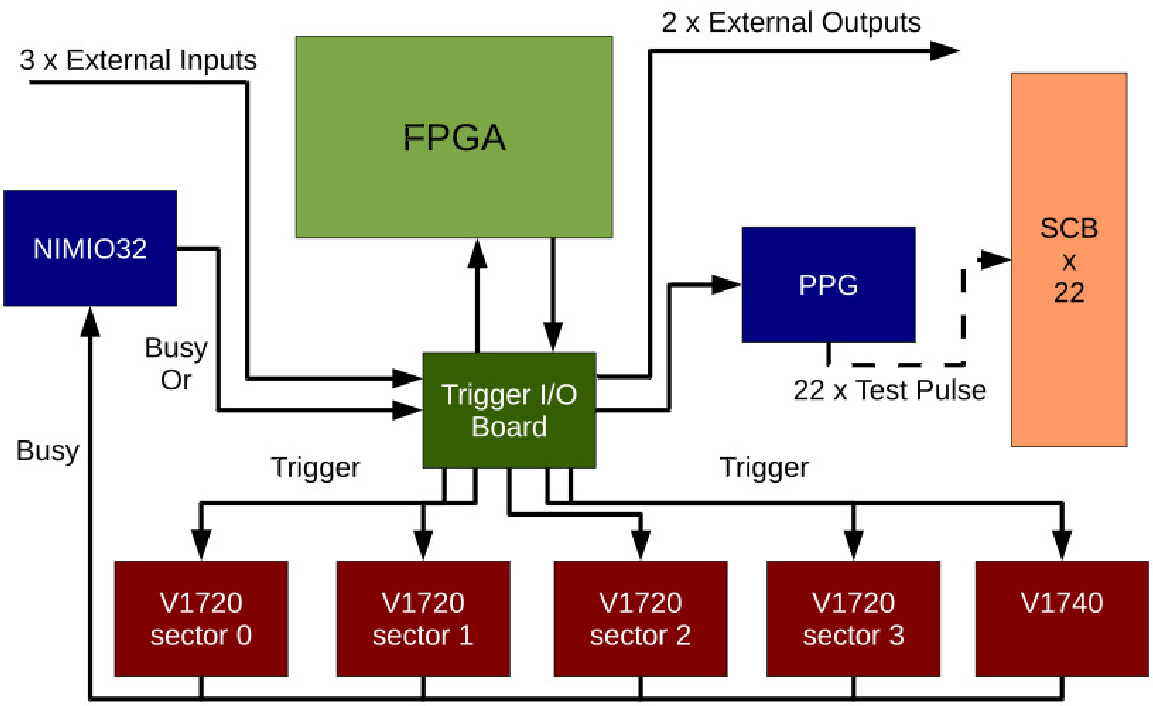


Figure 2.7: Diagram of the trigger signal distribution in the DAQ system.

2.2.4.1 Trigger Distribution

On the generation of a trigger signal by the DTM, or an external trigger input, a trigger signal is distributed to the DAQ system through the NIM board. The trigger signal is distributed on five dedicated trigger lines to the V1720 and V1740 sectors. For the purposes of clock and trigger distribution, the V1720 and V1740 digitizers are grouped into five logical sectors (Fig. 2.7). The 32 V1720s are grouped into four sectors where each logical sector contains eight physical modules. The V1740 modules form a single logical trigger and clock sector. Each logical trigger sector receives a single trigger signal distributed from the NIM board. The signal is received by the first board in each sector and then daisy chained, board to board, within each sector.

Note: For more on the busy signals, see the Twiki[16]

[16] https://www.snolab.ca/deap/private/TWiki/bin/view/Main/BusySystem

2.3 Clocking

Currently[17] the clock distribution and thereby the synchronization of events for the DTM and the V1720 and V1740 digitizers is done through a daisy chain arrangement.

The stability of this arrangement has been brought into question as some trigger signals were found to be arriving close to the clock boundary. This is an issue as the V1720 clock delays are only allowed to be certain values rather than arbitrary precision so the tuning of delays is limited. The issue has not arisen after the clock delays were readjusted. Further issues with the daisy chain clocking is that the cables are flaky. To address these potential problems a new clock distribution system has been developed. The daisy chained DTM generated master clock is to be replaced with a VME clock distribution system with the master clock being sourced from the DTM via the unused output on the NIM I/O board. The clock distribution modules[18], replace the need for a daisy chain arrangement and thereby removing a potential source of error. The clock generator FMC on the DTM is to be replaced with a SFP and Mini-SAS FMC.

2.3.1 Daisy Chain Clocking

The master clock for the DAQ system is generated from the motherboards 50 MHz source clock. A phase lock loop (PLL) on the DTM sets the frequency to 100 kHz for the periodic trigger (see Section 3.1), and 62.5 MHz clock for the PPG module and the digitizers. The dedicated clock generator mezzanine board passes the 62.5 MHz clock to the first digitizer in groups of at most six which is then passed along the chain. To match the expected trigger delay introduced by daisy chaining, a 3.57 ns delay is added to the clock daisy chain. The clock delay is configurable within the V1720 or V1740 modules as described by the manufacturer[1][2].

[17] Using edevel00268 July, 2016

[18] `https://edev.triumf.ca/project/edev/vme/edevel00163`

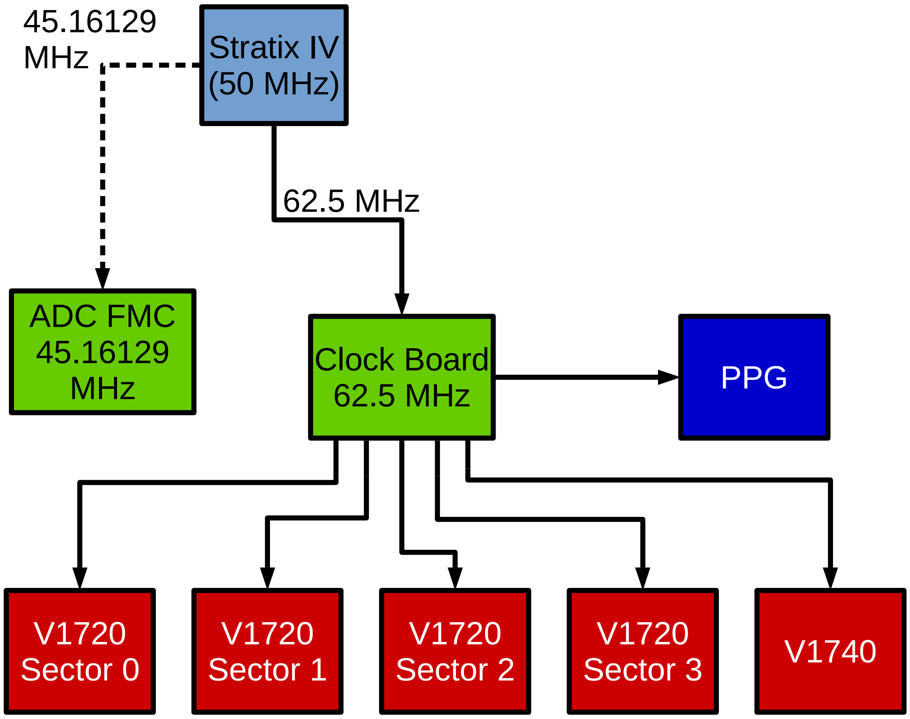


Figure 2.8: Clocking diagram of the data acquisition system under the daisy chain configuration

2.3.1.1 Distributed Master Clock

Replacing the daisy chain clocking is a distributed system removing the necessity of setting delays in the chain. The distributed system used the Clock Distribution Modules[19] (CDM) designed by the TRIUMF EDEV department for use on the GRIFFIN project. The 62.5 MHz DAQ master clock is generated in the DTM and fed through the NIM I/O FMC to the first of three the CDM. This first slave CDM cleans the clock and outputs two synced, phase locked clocks to two other CDMs which then fan out all of the DAQ clocks. As the same wire is used for all connections and the sources are synced and phase locked, there is no need for any programmed delays.

[19] https://edev.triumf.ca/project/edev/vme/edevel00163

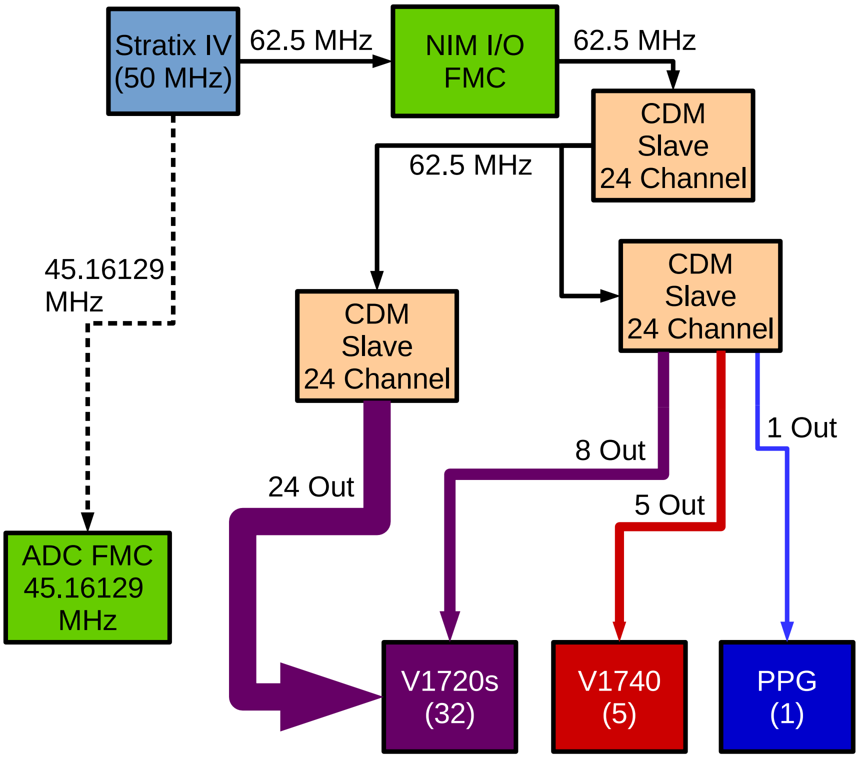


Figure 2.9: Diagram of the clock distribution system for the data acquisition system using a distributed master clock method, replacing the daisy chain configuration in Fig. 2.8.

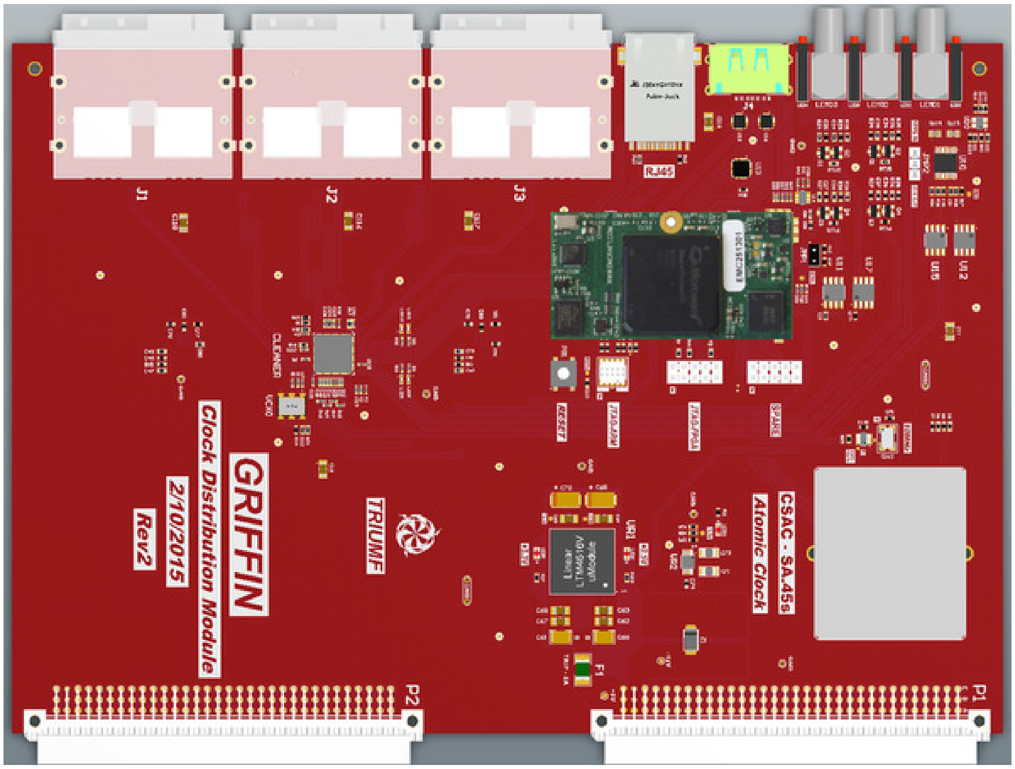


Figure 2.10: Clock distribution module to be used for the distributed clocking of the system, supplanting the current (July 2016) DAQ daisy chain method.

2.3.1.2 Clock Distribution Modules

EDEV Project Location: https://edev.triumf.ca/project/edev/vme/edevel00163

Summary: 24 channel low jitter LVDS Clock/Sync generation and fan out module

The clock distribution modules, shown in Fig. 2.10, are the essential part of the clock distribution system. Designed for the GRIFFIN experiment by the TRIUMF EDEV department, these boards output a stable and phase locked clock through up to 24 channels.

2.4 DTM Configurations

There are two FMC configurations used for the DTM, both use the same NIM I/O and 24-channel ADC FMCs but switch out the clock generator (Fig. 2.11) with the SFP

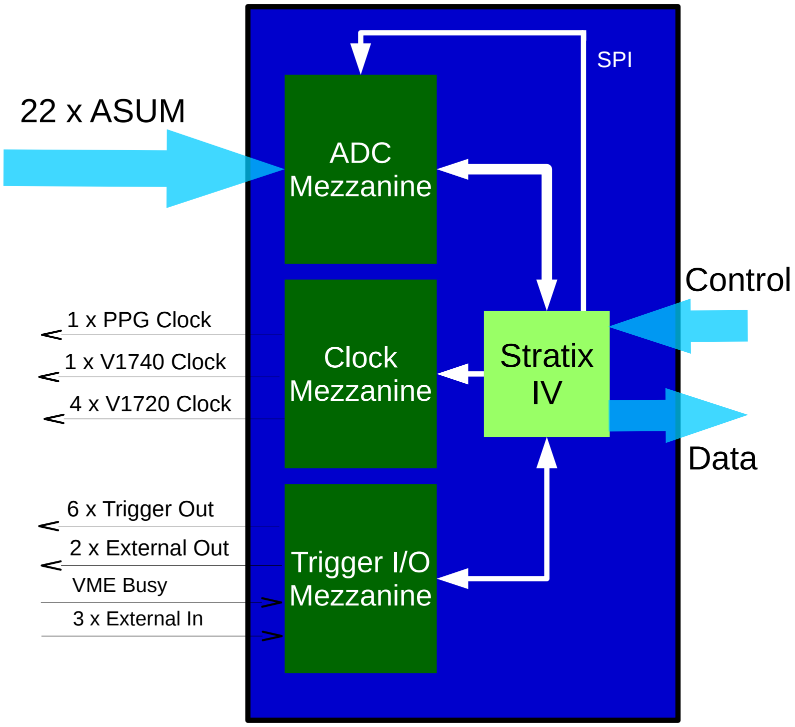


Figure 2.11: DTM configuration using the edevel00268 firmware project (see Section 4.5). The master clock is generated by the Stratix IV and fed out to the digitizers and externals through the clock generator module (Section 2.3.1). The master clock is fed out to first in a section of digitizers and is passed along in a daisy chain with delays programmed into the digitizers as described in section 2.3.1.

and Mini-SAS board (Fig. 2.12). This change of boards is due to an intended[20] change in the clock distribution method from a daisy-chain to a distributed phase-locked clock system (see Section 2.3). The addition of the SFP connection comes with added ethernet capabilities allowing ethernet data readout to replace the current VME readout system (see Chapter 5 for more on the FTP and ethernet data readout). The firmware project that uses the clock generator module is edevel00268 (Section 4.5) and the SFP and Mini-SAS project is edevel00365 (Section 4.6).

[20] Not yet installed at SNOLAB as of July 2016

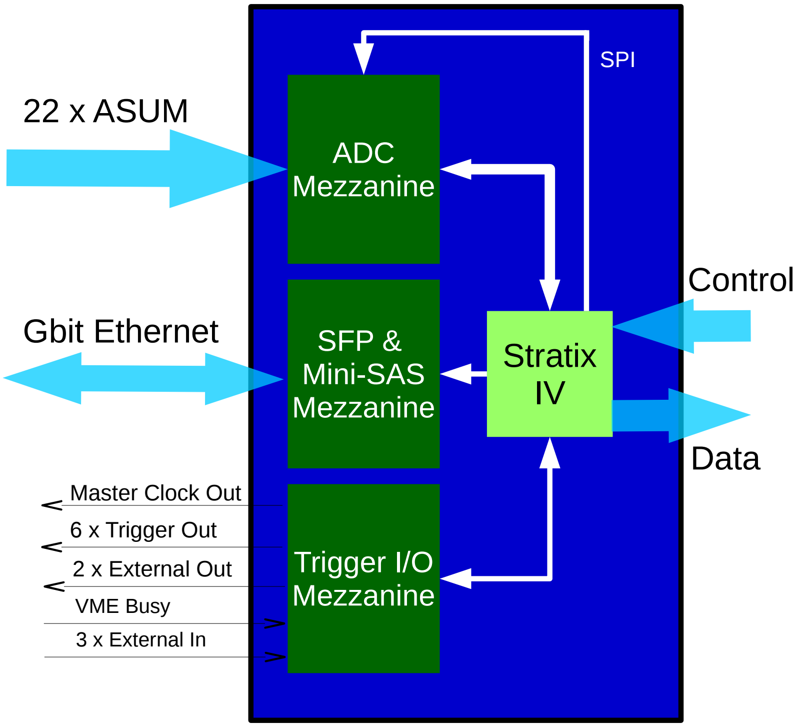


Figure 2.12: DTM configuration using the edevel00365 firmware project (see Section 4.6). The master clock is generated by the Stratix IV and fed out to the digitizers and externals through the NIM I/O module. The master clock is then sent to a clock distribution module in slave mode which feeds out two cleaned, phase locked, and synchronized clocks to two more clock distribution modules in slave mode (see Section 2.3.1.1). A SFP and Mini-SAS FMC replaces the clock generator board which adds FTP connectivity and ethernet readout of the registers.

Chapter 3

Trigger Types

There are five primary trigger types currently implemented in the DEAP-3600 firmware: periodic, exponential, external, minimum bias, and the ADC trigger. The trigger types, operation, and user settings are discussed below. More on the ODB can be found in Section 3.6 with the settings and variable descriptions in the twiki. This chapter borrows from the twiki, the link to which can be found here.

Note: Simulation information can be found on the twiki[1]

3.1 Periodic

SVN Location: https://edev.triumf.ca/svn/edevel00261/trunk/tsb/ip/rtl/source/ts_periodic.v

Module Name: TriggerSourcePeriodic

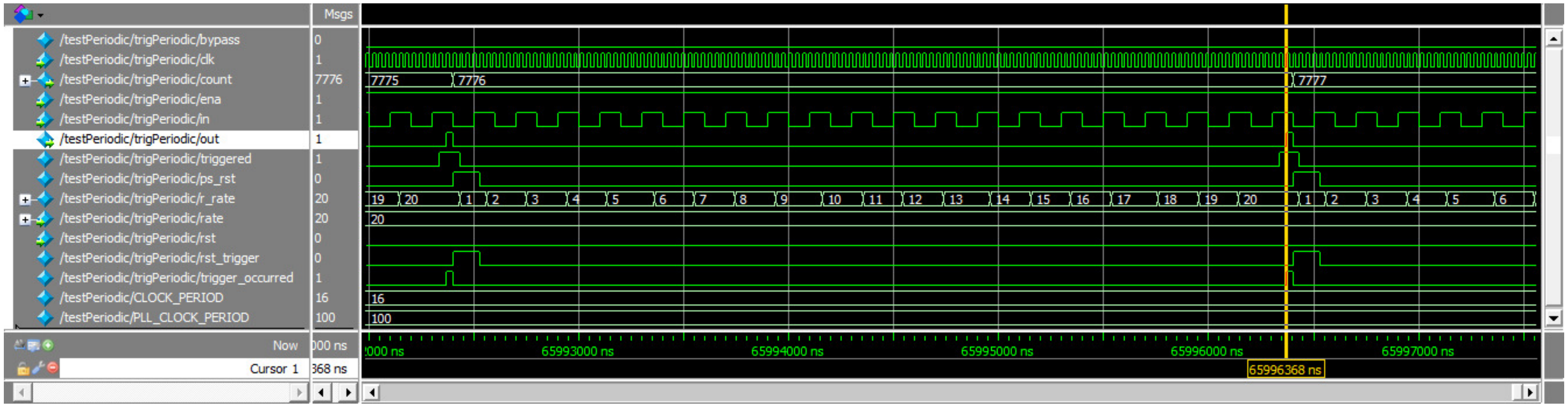
Frequency Range: 1 Hz → 100 kHz

Submodules: None

Summary: Used for testing the DAQ system with a periodic signal (fixed interval between triggers). Uses a 100 kHz PLL and a counter to divide the rate. This trigger does not have a separate module, but is included in the VME wrapper in ede-

[1] `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/FirmwareSimulation`

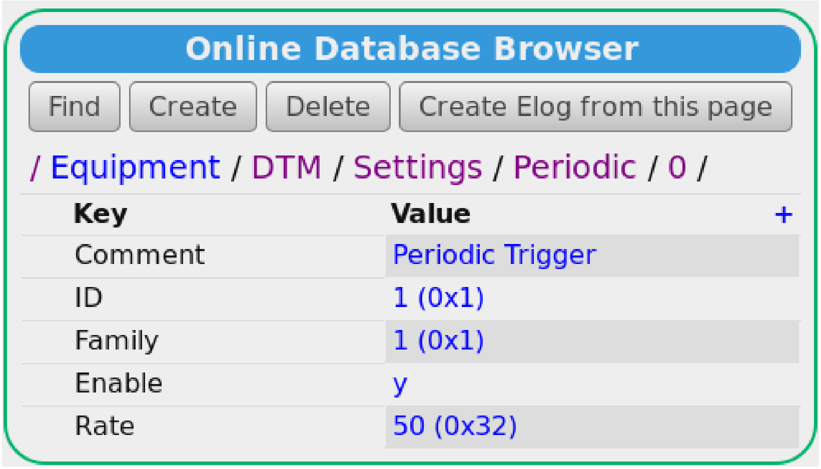
vel00270[1]. The periodic trigger has two identical channels used for the PPG pulse injection, AARF firing, status updates, etc.



```
                                                Msgs
/testPeriodic/trigPeriodic/bypass               0
/testPeriodic/trigPeriodic/clk                  1
/testPeriodic/trigPeriodic/count                7776
/testPeriodic/trigPeriodic/ena                  1
/testPeriodic/trigPeriodic/in                   1
/testPeriodic/trigPeriodic/out                  1
/testPeriodic/trigPeriodic/triggered            1
/testPeriodic/trigPeriodic/ps_rst               0
/testPeriodic/trigPeriodic/r_rate               20
/testPeriodic/trigPeriodic/rate                 20
/testPeriodic/trigPeriodic/rst                  0
/testPeriodic/trigPeriodic/rst_trigger          0
/testPeriodic/trigPeriodic/trigger_occurred     1
/testPeriodic/CLOCK_PERIOD                      16
/testPeriodic/PLL_CLOCK_PERIOD                  100
                                     Now        000 ns
                                     Cursor 1   368 ns
```

Figure 3.1: Timing diagram for the periodic trigger.

3.1.1 ODB Settings



/ Equipment / DTM / Settings / Periodic / 0 /

Key	Value
Comment	Periodic Trigger
ID	1 (0x1)
Family	1 (0x1)
Enable	y
Rate	50 (0x32)

Figure 3.2: Online data base settings for the periodic trigger. labelFig:periodicODB

Trigger ID: Two identical channels

- **Channel 0:** 0x1
- **Channel 1:** 0xFFFFFFFF

Family: 0x1 (Periodic)

Enable: Turns the trigger module on/off

Rate: Set rate of the trigger in Hz (1 → 10^5 Hz)

3.2 Exponential

SVN Location: https://edev.triumf.ca/svn/edevel00261/trunk/tsb/ip/rtl/source/ts_exponential.v

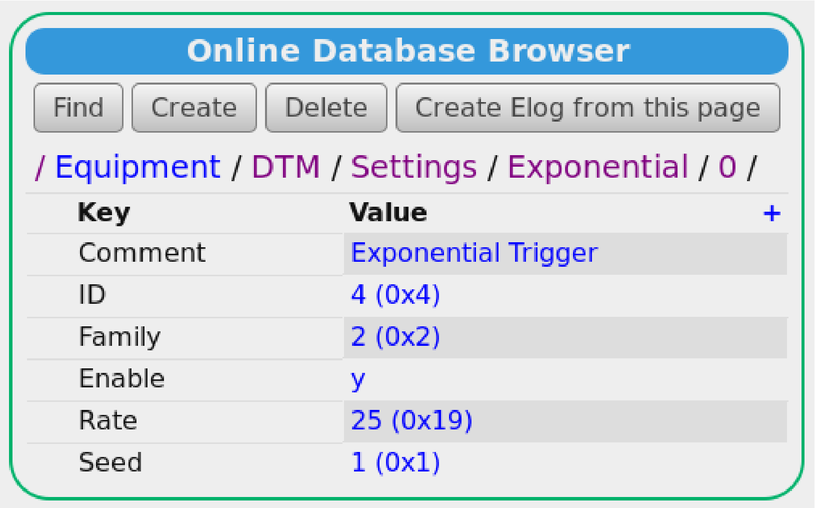
Module Name: TriggerSourceExponential

Frequency Range: 1 Hz → 62.5 MHz

Submodules: None

Summary: Used for testing the DAQ system with exponentially distributed random intervals between triggers. Can emulate LAr background decay trigger rate, PMT dark noise trigger rate, expected event trigger rate → Emulates real DAQ conditions. Uses a C++ program running on a NIOS core that generates the exponentially distributed random numbers for all 5 trigger sources, a FIFO to store those numbers and a counter that generates the trigger when the next random number is reached. The rate is more correctly the mean of the distribution from which delay times are taken, so the rate here is actually the time averaged rate.

3.2.1 ODB Settings



/ Equipment / DTM / Settings / Exponential / 0 /

Key	Value
Comment	Exponential Trigger
ID	4 (0x4)
Family	2 (0x2)
Enable	y
Rate	25 (0x19)
Seed	1 (0x1)

Figure 3.3: Online data base settings for the exponential trigger.

Trigger ID: Five different identical channels

Channel 0: 0x4

Channel 1: 0x8

Channel 2: 0x10

Channel 3: 0x20

Channel 4: 0x40

Family: 0x2 (Exponential)

Enable: Turns the trigger module on/off

Rate: Average rate of the trigger in Hz

Seed: Seed value for the random number generator

3.3 External

SVN Location: https://edev.triumf.ca/svn/edevel00261/trunk/tsb/ip/rtl/source/ts_external.v

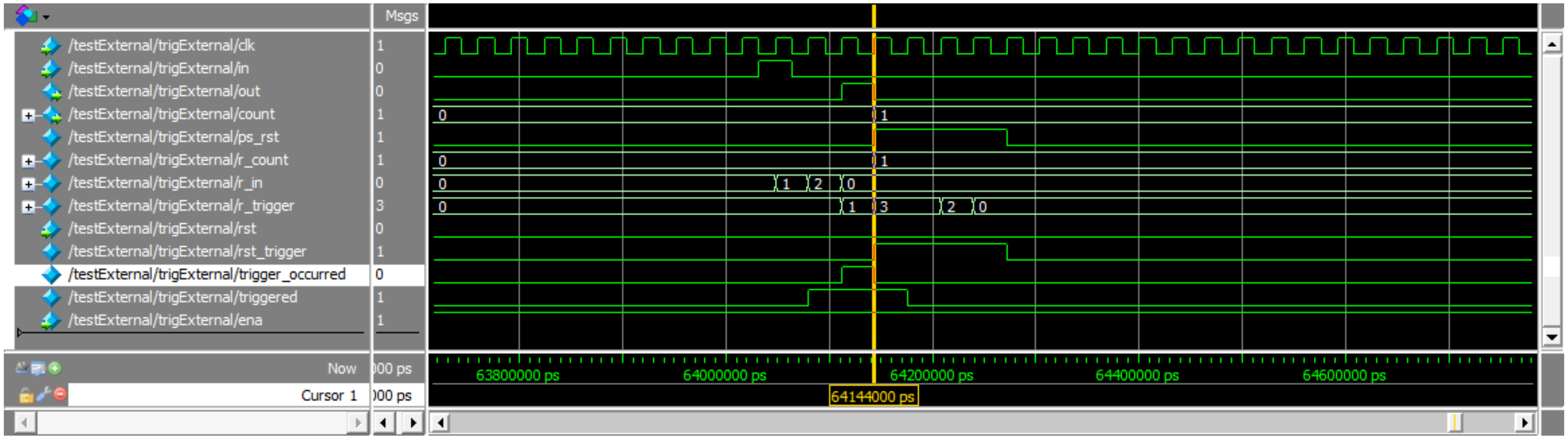
Module Name: TriggerSourceExternal

Frequency Range: 0 Hz → 62.5 MHz (Pulses must be at least 16 ns wide)

Submodules: None

Summary: External triggers are sourced from the veto PMTs and the calibration devices via the NIM I/O FMC (Section 2.2.4). These external triggers are primarily intended for the removal of muon events which can be tagged by the veto system as they create Cerenkov radiation in the water tank. There are three identical channels, one for the veto system and two of which are currently unassigned[2].

[2] July 2016



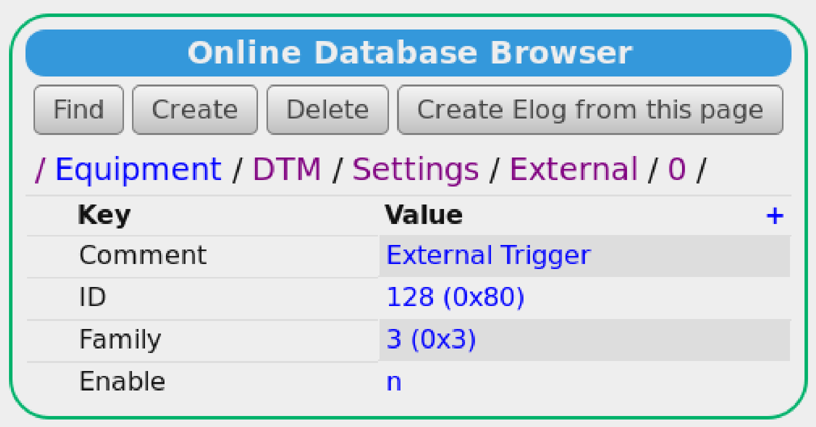
	Msgs
/testExternal/trigExternal/clk	1
/testExternal/trigExternal/in	0
/testExternal/trigExternal/out	0
/testExternal/trigExternal/count	1
/testExternal/trigExternal/ps_rst	1
/testExternal/trigExternal/r_count	1
/testExternal/trigExternal/r_in	0
/testExternal/trigExternal/r_trigger	3
/testExternal/trigExternal/rst	0
/testExternal/trigExternal/rst_trigger	1
/testExternal/trigExternal/trigger_occurred	0
/testExternal/trigExternal/triggered	1
/testExternal/trigExternal/ena	1

Figure 3.4: Timing diagram for the external trigger.

Table 3.1: Trigger ID and connections for the external trigger

Channel	Trigger ID	NIM Channel	Application
0	0x80	0	Veto
1	0x100	1	Calib
2	0x200	2	External Trigger 2

3.3.1 ODB Settings



/ Equipment / DTM / Settings / External / 0 /

Key	Value
Comment	External Trigger
ID	128 (0x80)
Family	3 (0x3)
Enable	n

Figure 3.5: Online data base settings for the external trigger.

Family: 0x3 (External)

Enable: Turns the trigger module on/off

3.4 Minimum Bias

SVN Location: https://edev.triumf.ca/svn/edevel00261/trunk/tsb/ip/rtl/source/ts_adc_min_bias.v

Module Name: TriggerSourceADCMinimumBias

Frequency Range: Settings Dependant, minimum time between triggering four clock delay (Reset and flip-flop setting) + dead time + ToT time (see Fig. 3.7 and 3.8)

Submodules: min_bias_sum (channel counting megafunction)

Summary: The minimum bias trigger compares the ASUM data to a threshold set in the online data base ODB. If the signal stays above this threshold for a certain defined time known as the time over threshold (ToT), then the channel is said to have latched (diagram in Fig. 3.6). The ToT requirement is intended to reduces the chance of triggering on noise.

There are two modes for the min. bias trigger: single channel or coincidence mode. In single channel mode, once one channel latches the trigger condition is met and a trigger signal is sent. In coincidence mode multiple channels need to latch within a definable amount of time known as the coincidence time window. In coincidence mode once the last channel has latched a trigger signal is sent.

Once a trigger condition has been met and a trigger signal output, the trigger goes into a dead state until the defined deadtime has been passed. The deadtime is intended to reduce the chance of retriggering on the same event.

Note: The minimum bias can be used with the 22 ASUM channels and the ASUM-SUM, which channels are enabled are set in the ODB.

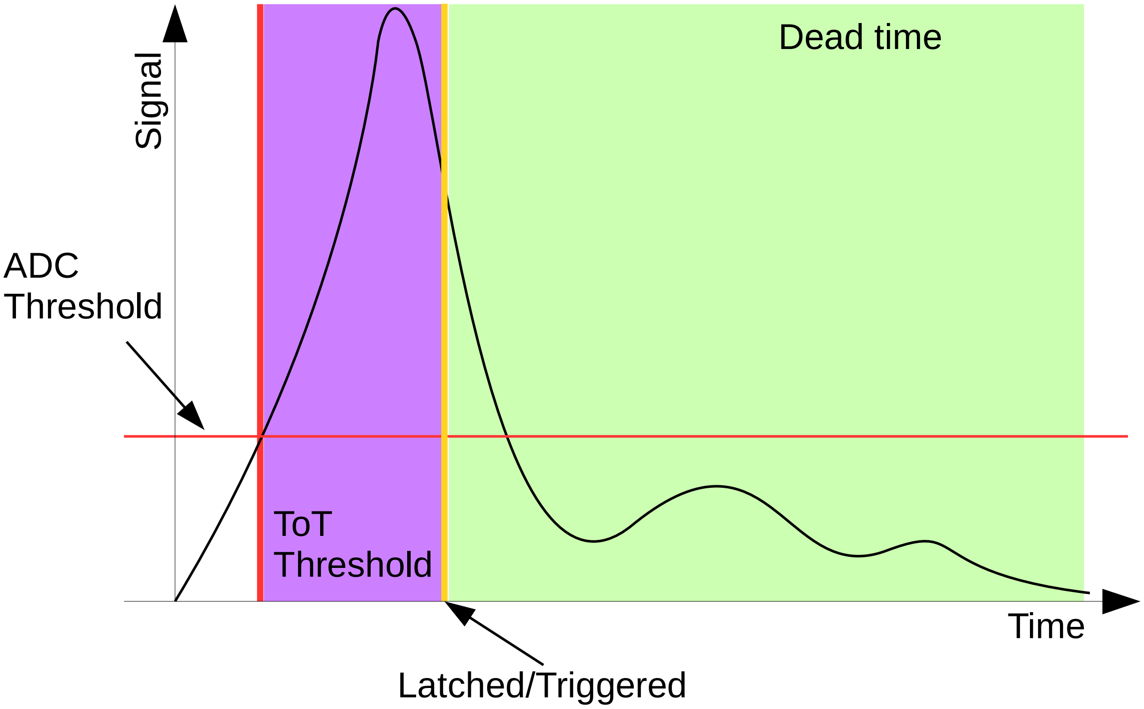
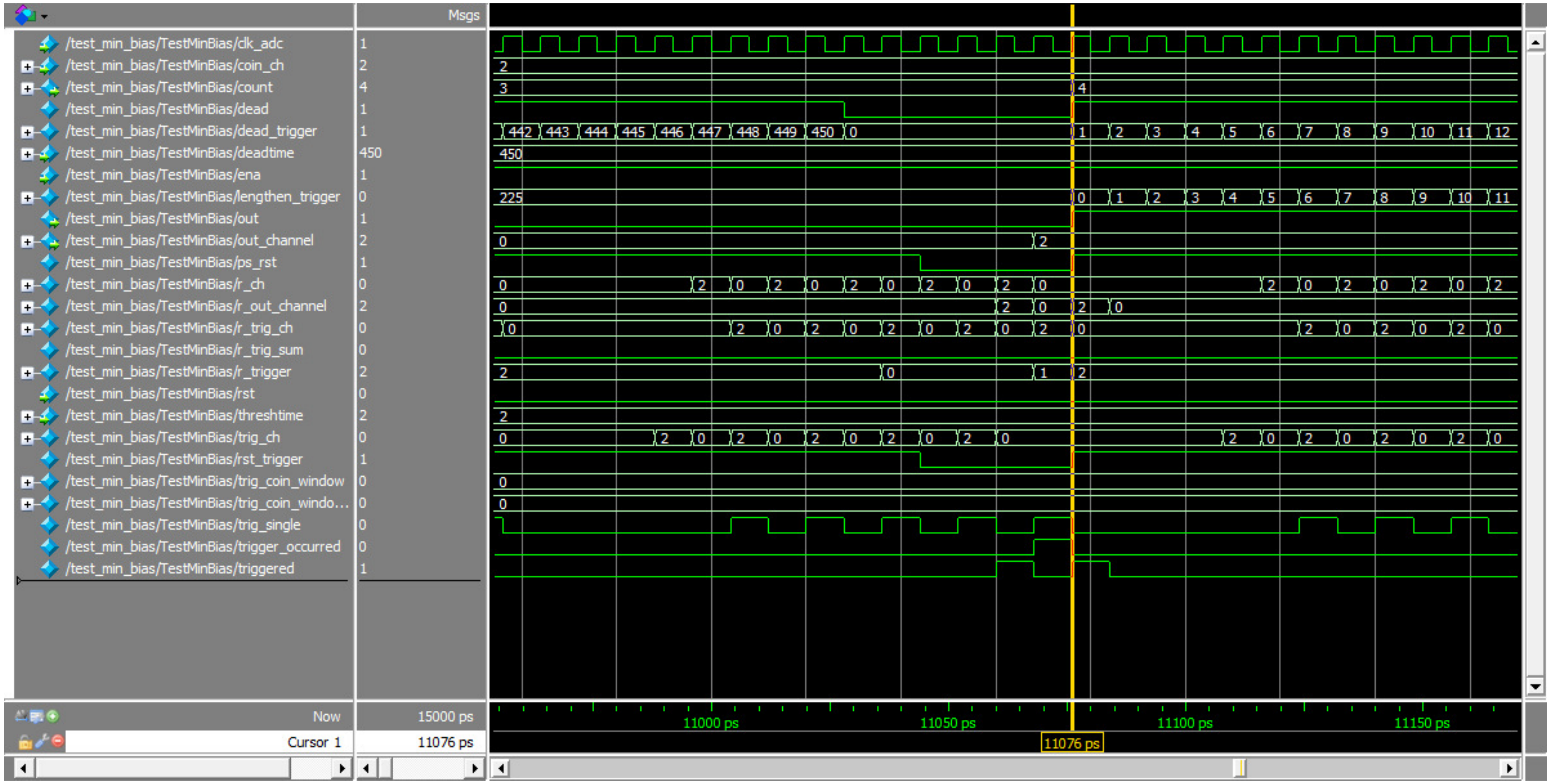
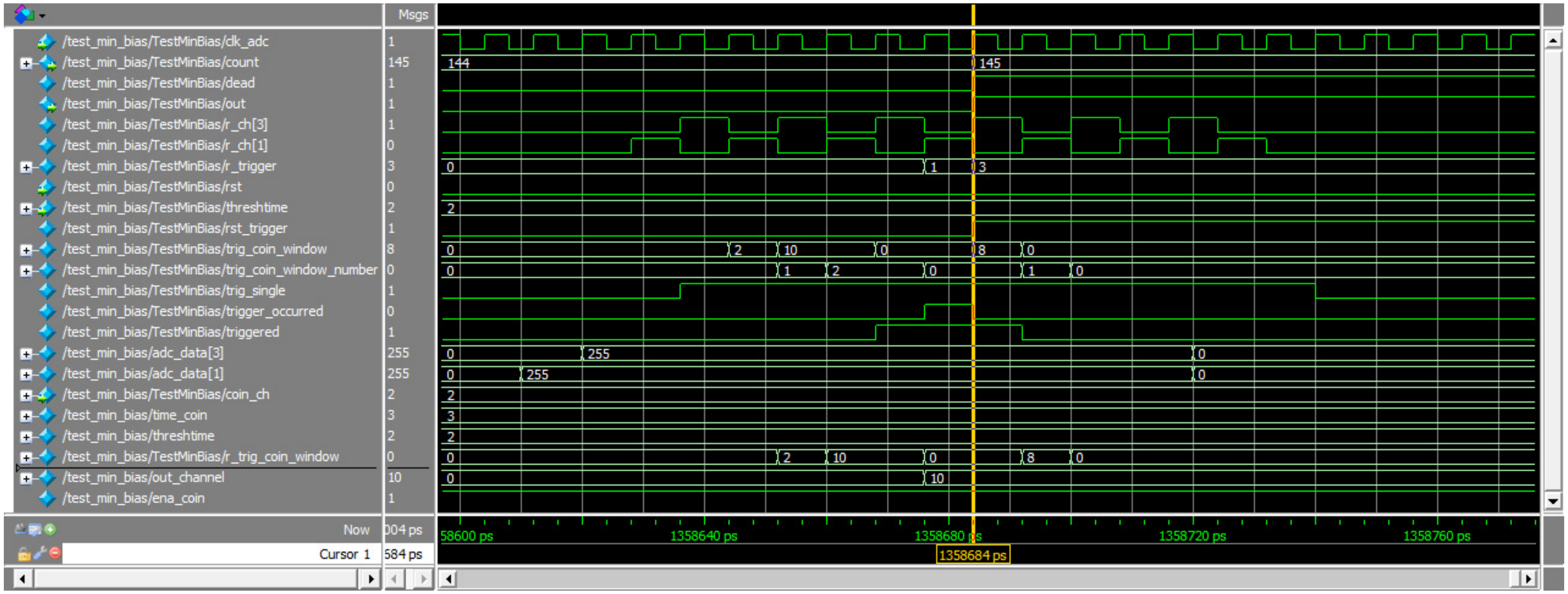


Figure 3.6: Cartoon depicting the operation of the minimum bias trigger. The trigger fires once the signal has remained above the adc threshold (horizontal line) for the time over threshold (vertical red line) at which point the trigger will fire immediately (vertical yellow line). The trigger may not re-fire until the deadtime has been exceeded.



	Msgs
/test_min_bias/TestMinBias/clk_adc	1
/test_min_bias/TestMinBias/coin_ch	2
/test_min_bias/TestMinBias/count	4
/test_min_bias/TestMinBias/dead	1
/test_min_bias/TestMinBias/dead_trigger	1
/test_min_bias/TestMinBias/deadtime	450
/test_min_bias/TestMinBias/ena	1
/test_min_bias/TestMinBias/lengthen_trigger	0
/test_min_bias/TestMinBias/out	1
/test_min_bias/TestMinBias/out_channel	2
/test_min_bias/TestMinBias/ps_rst	1
/test_min_bias/TestMinBias/r_ch	0
/test_min_bias/TestMinBias/r_out_channel	2
/test_min_bias/TestMinBias/r_trig_ch	0
/test_min_bias/TestMinBias/r_trig_sum	0
/test_min_bias/TestMinBias/r_trigger	2
/test_min_bias/TestMinBias/rst	0
/test_min_bias/TestMinBias/threshtime	2
/test_min_bias/TestMinBias/trig_ch	0
/test_min_bias/TestMinBias/rst_trigger	1
/test_min_bias/TestMinBias/trig_coin_window	0
/test_min_bias/TestMinBias/trig_coin_windo…	0
/test_min_bias/TestMinBias/trig_single	0
/test_min_bias/TestMinBias/trigger_occurred	0
/test_min_bias/TestMinBias/triggered	1
Now	15000 ps
Cursor 1	11076 ps

Figure 3.7: Timing diagram for the minimum bias trigger in single channel mode.



	Msgs
/test_min_bias/TestMinBias/clk_adc	1
/test_min_bias/TestMinBias/count	145
/test_min_bias/TestMinBias/dead	1
/test_min_bias/TestMinBias/out	1
/test_min_bias/TestMinBias/r_ch[3]	1
/test_min_bias/TestMinBias/r_ch[1]	0
/test_min_bias/TestMinBias/r_trigger	3
/test_min_bias/TestMinBias/rst	0
/test_min_bias/TestMinBias/threshtime	2
/test_min_bias/TestMinBias/rst_trigger	1
/test_min_bias/TestMinBias/trig_coin_window	8
/test_min_bias/TestMinBias/trig_coin_window_number	0
/test_min_bias/TestMinBias/trig_single	1
/test_min_bias/TestMinBias/trigger_occurred	0
/test_min_bias/TestMinBias/triggered	1
/test_min_bias/adc_data[3]	255
/test_min_bias/adc_data[1]	255
/test_min_bias/TestMinBias/coin_ch	2
/test_min_bias/time_coin	3
/test_min_bias/threshtime	2
/test_min_bias/TestMinBias/r_trig_coin_window	0
/test_min_bias/out_channel	10
/test_min_bias/ena_coin	1
Now	004 ps
Cursor 1	584 ps

Figure 3.8: Timing diagram for the minimum bias trigger in coincidence mode.

3.4.1 ODB Settings

Trigger ID: 0x8000

Family: 0x4 (Physics)

Enable: Turns the trigger module on/off

Enable Baseline Compensation: Will apply the baseline subtraction from the read ADC value. If enabled the thresholds will be used on the baseline subtracted reading.

Dead Time Window: 16-bit value which sets the number of clocks after a trigger event wherein no more triggers will be produced to avoid retrieving on the same event. (**Note:** This should be significantly larger than the long time window

Time over Threshold: 16-bit value which sets the minimum number of clocks after a threshold has been crossed that the signal must remain high for a trigger to be produced. **Note:** As this is used to only remove noise, the ToT is usually set at a only a few clocks

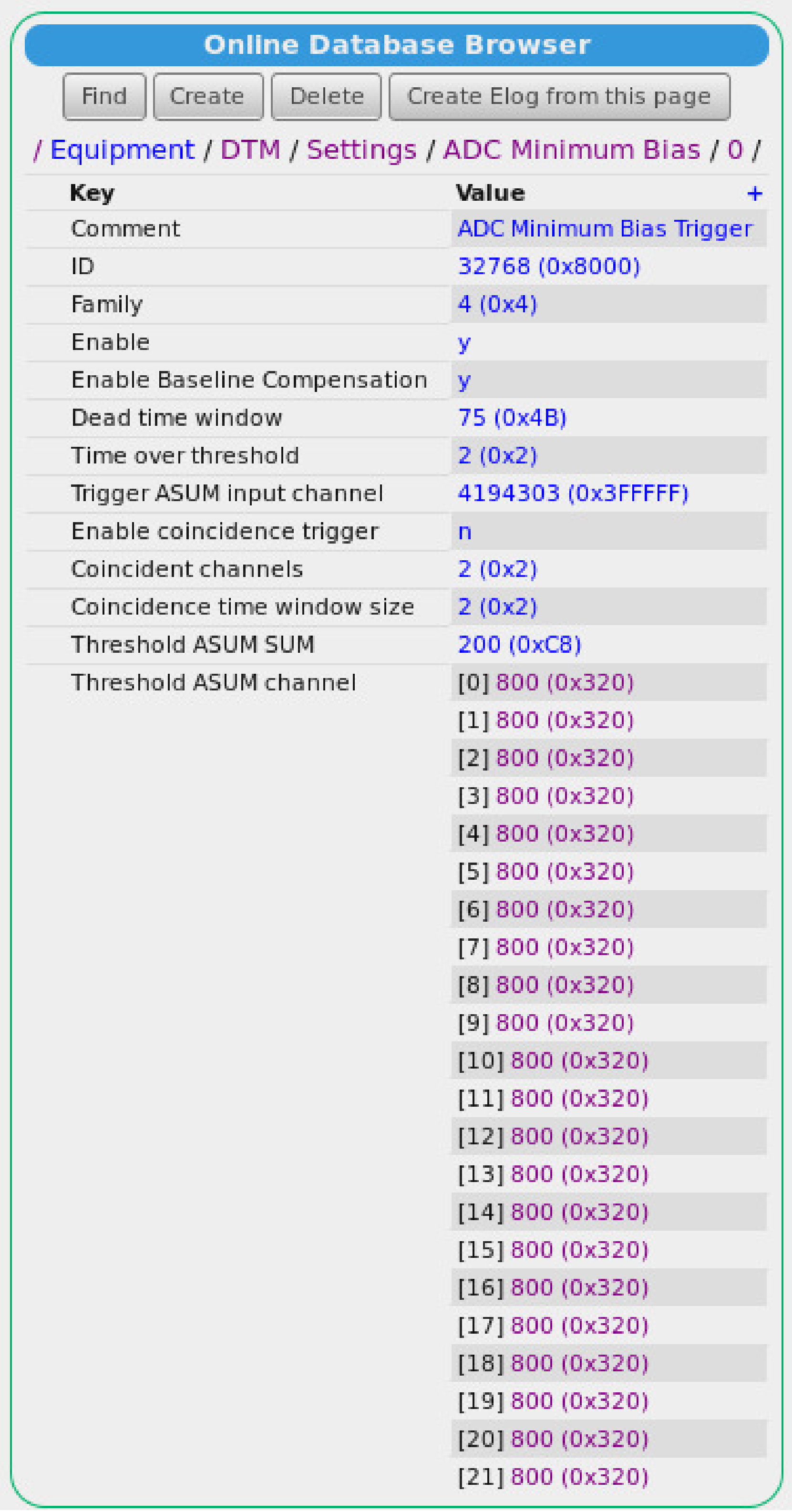
Trigger ASUM input channel: Bit mask for enabled channel inputs; a disabled channel has its corresponding bit set false. Bits 1-22 are the ASUM channels and bit 23 is the ASUMSUM. Any combination can be enabled.

Coincident Channels: Number of channels required to have fulfilled the triggering requirements within the coincident time window size if the coincidence trigger is enabled. For instance if this value is set to four, than within the coincidence time window size a minimum of 4 channels must have crossed their thresholds for a time greater than the ToT for a trigger to be produced.

Coincident time window size: 16-bit value which sets the number of clocks in the time window for the number of channels defined by the coincident channels setting to have each meant the single channel trigger requirements (only applies for coincidence trigger mode).

Threshold ASUMSUM: 17-bit value setting the minimum bias threshold for the ASUMSUM.

Threshold ASUM channel: 12-bit value setting the minimum bias threshold for each of the ASUM channels.



/ Equipment / DTM / Settings / ADC Minimum Bias / 0 /

Key	Value
Comment	ADC Minimum Bias Trigger
ID	32768 (0x8000)
Family	4 (0x4)
Enable	y
Enable Baseline Compensation	y
Dead time window	75 (0x4B)
Time over threshold	2 (0x2)
Trigger ASUM input channel	4194303 (0x3FFFFF)
Enable coincidence trigger	n
Coincident channels	2 (0x2)
Coincidence time window size	2 (0x2)
Threshold ASUM SUM	200 (0xC8)
Threshold ASUM channel	[0] 800 (0x320)
	[1] 800 (0x320)
	[2] 800 (0x320)
	[3] 800 (0x320)
	[4] 800 (0x320)
	[5] 800 (0x320)
	[6] 800 (0x320)
	[7] 800 (0x320)
	[8] 800 (0x320)
	[9] 800 (0x320)
	[10] 800 (0x320)
	[11] 800 (0x320)
	[12] 800 (0x320)
	[13] 800 (0x320)
	[14] 800 (0x320)
	[15] 800 (0x320)
	[16] 800 (0x320)
	[17] 800 (0x320)
	[18] 800 (0x320)
	[19] 800 (0x320)
	[20] 800 (0x320)
	[21] 800 (0x320)

Figure 3.9: Online data base settings for the min. bias trigger.

3.5 ADC Trigger

SVN Location: https://edev.triumf.ca/svn/edevel00261/trunk/tsb/ip/rtl/source/ts_adc.v

Module Name: TriggerSourceADC

Frequency Range: Settings Dependant, minimum time between triggering five clock delay (reset and flip-flop setting) + dead time + scan time[3] (See Fig. 3.13)

Submodules: trigger_self_sum_adc (see Section 4.2.2), TriggerSelfDecision (see Section 4.2.3)

Summary: The ADC trigger receives the baseline subtracted ASUMSUM and analyzes the short energy and Fprompt to make a triggering decision. The module trigger_self_sum (Section 4.2.2) performs a rolling integral over a short (∼8 clocks)[4] and a long time window (∼150 clocks) as shown in Fig. 3.12. The short integration window integral is delayed to be valid at the same time as the long window (see Section 4.2.2) so the decision is always made after the integration of both windows is finished.

The integrals are passed to the module TriggerSelfDecision (Section 4.2.3) which dynamically sorts 'events' into one of six groups as shown in Fig. 3.10. If the charge in the short window exceeds the low energy limit, then a variable 'trig_out' is set as shown in Fig. 3.11 (colours and bit positions are consistent with Fig. 3.10). If the value of trig_out increase (i.e. higher Fprompt or short charge) or stays the same with the energy increasing, the values from that time are saved and the comparison continues. If The value of trig_out decreases or stays the same with the short energy decreasing or staying the same, then an ODB defined timer referred to as the look ahead time [5] (∼15 clocks) begins counting. If a value of trig_out of the same or greater value is found within this look ahead time, then that the values (short/long window charge, trigger type, and trigger start time, etc.) are set to the new events

[3] Scan time in the ODB, look ahead time in the literature, the firmware variable in the timing diagram is 'waittime'

[4] This module uses the ADC clock of 45.161290 MHz

[5] this is called the scan time in the ODB, waittime in the firmware, or look ahead time in some documentation

and the timer is restarted[6]. Only once the timer reaches the look ahead time without finding a higher value of trig_out is a trigger produced.

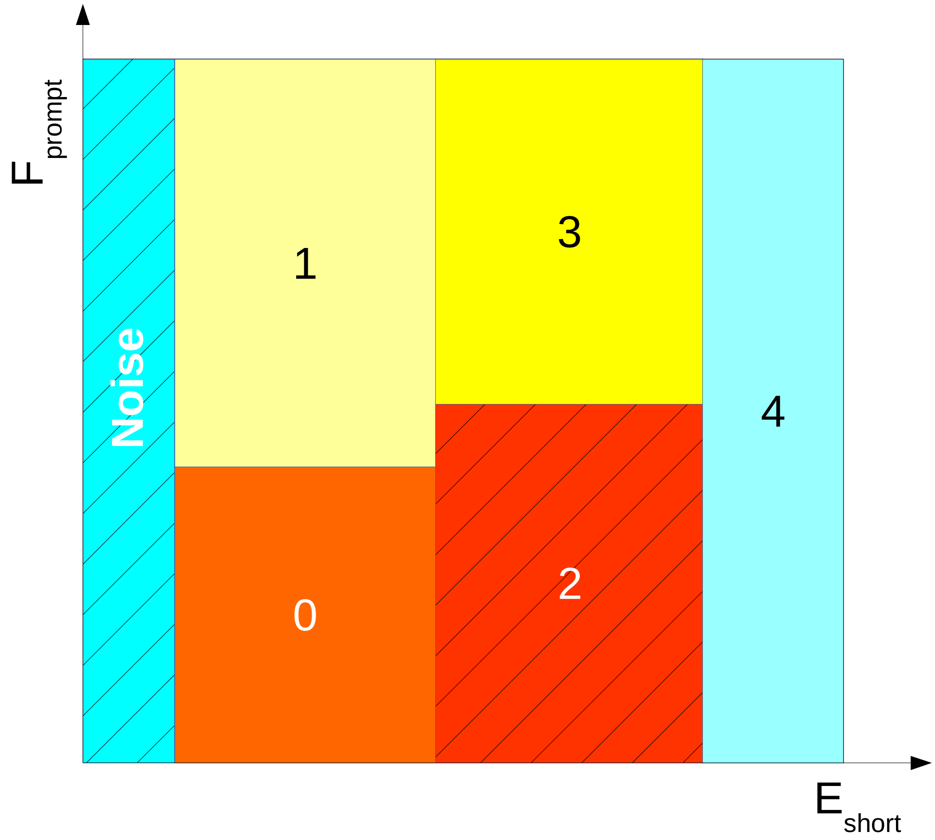


Figure 3.10: Trigger regions as used in the pulse shape discrimination background rejection method. Noise in the PMTs fall under very low energy and β events are grouped into high energy, low Fprompt. The WIMP region of interest is at high Fprompt and could fall into either region of energy.

Very High Energy Bit 4	High Energy High Fprompt Bit 3	High Energy Low Fprompt Bit 2	Low Energy High Fprompt Bit 1	Low Energy Low Fprompt Bit 0

Figure 3.11: trig_out variable, the higher the charge and Fprompt, the higher the value. Values and colours correspond to the regions shown in Fig. 3.10.

[6] If a second maxima is found in the look ahead time of equal or greater charge than the first, the scan time will start again from that point and the reported trigger will be of the last maxima.

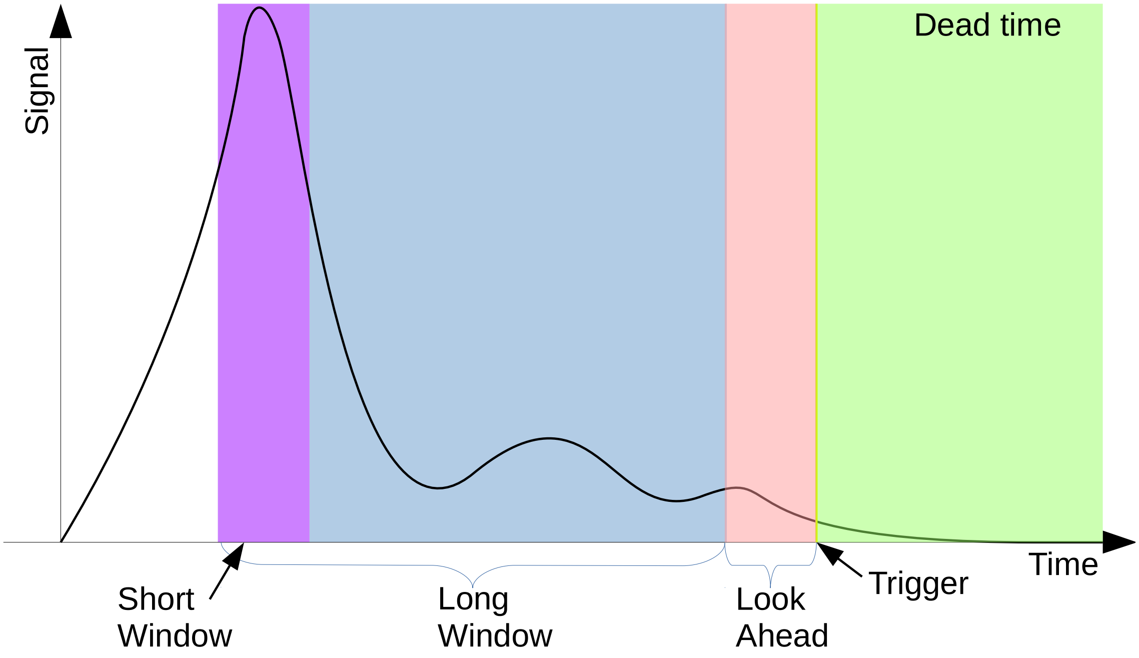


Figure 3.12: Short and long time integration windows for the determination of the fraction of prompt charge in an event (Fprompt).

The Fprompt is not explicitly calculated however it is used for event selection; for a defined Fprompt (for $1<\mathrm{F}_{\mathrm{thres}} < 256$) an event is recorded if:

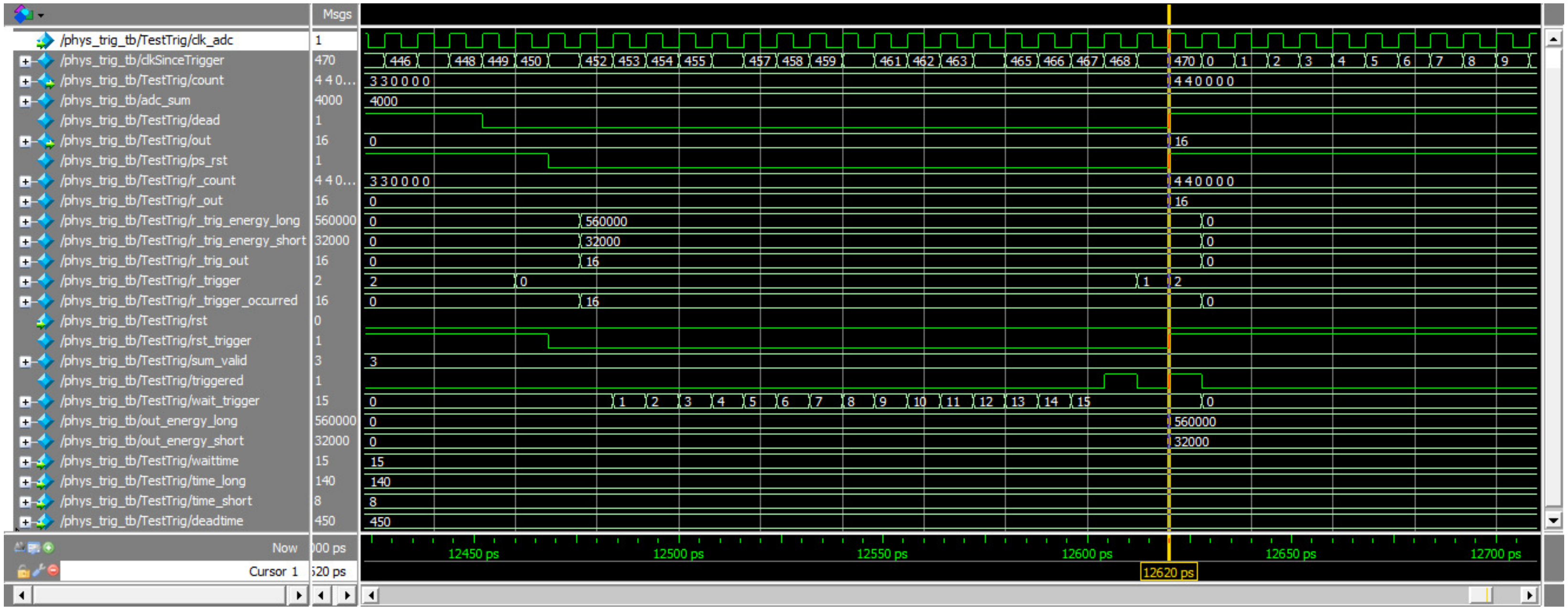
$$\mathrm{E}_{\mathrm{long}} > \mathrm{E}_{\mathrm{thres}} \ \& \ \mathrm{E}_{\mathrm{long}} \times \mathrm{F}_{\mathrm{thres}} > [\mathrm{E}_{\mathrm{short}} \times 256] \, . \tag{3.1}$$

Thus if a Fprompt of 0.5 is desired, the ODB value is set to $\mathrm{F}_{\mathrm{thres}} = (256 \times 0.5) = 128$. Events are sorted using the short energy and Fprompt into one of five regions as depicted in Fig. 3.10 (region 0 is defined as below threshold therefore nothing is recorded).

After a trigger signal has been generated the trigger will enter a dead state where no triggering is allowed. This dead time is implemented so as to avoid retriggering on the same event[7]. A diagram of the ADC trigger is given in Fig. 3.12.

Note: The look ahead time does occur after the long window finishes integrating, but as the short and long window times are valid at the same time due to a delay fifo, the look ahead time applies 'back in time' if you will with regards to Fig. 3.12. A full timing diagram of the ADC trigger while constantly triggering is shown in Fig. 3.13.

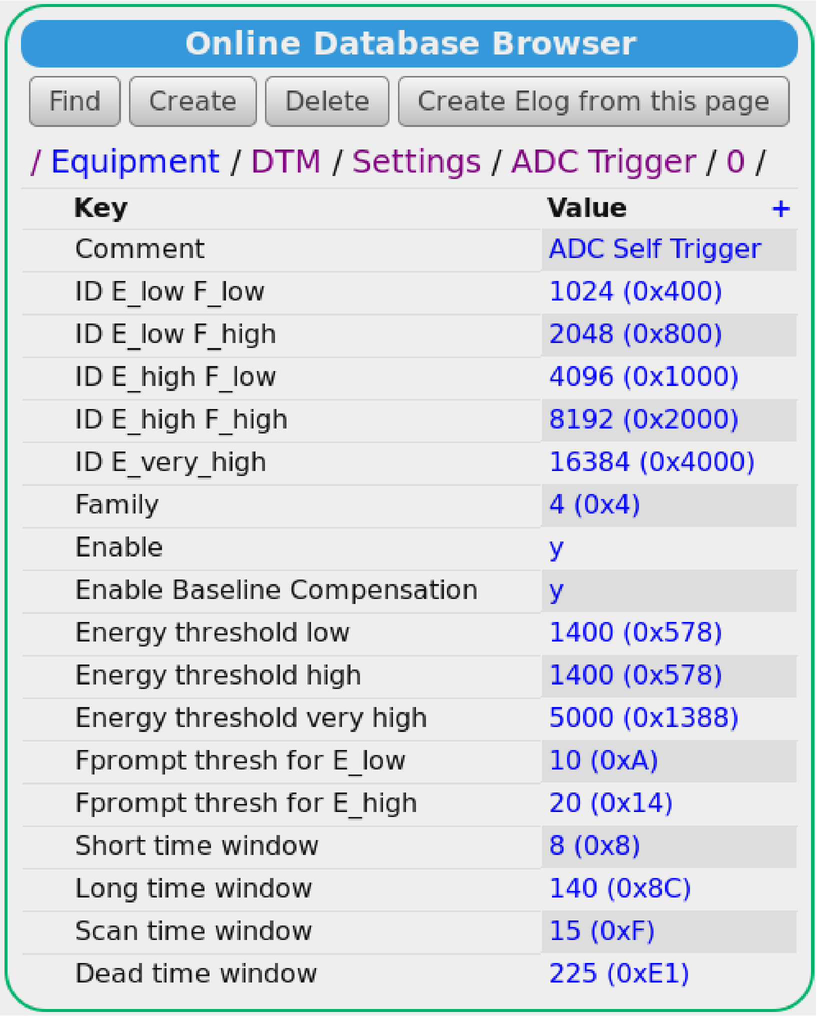
[7] Note that the scan time adds to the dead time for the minimum time between triggers



	Msgs
/phys_trig_tb/TestTrig/clk_adc	1
/phys_trig_tb/clkSinceTrigger	470
/phys_trig_tb/TestTrig/count	4 4 0…
/phys_trig_tb/adc_sum	4000
/phys_trig_tb/TestTrig/dead	1
/phys_trig_tb/TestTrig/out	16
/phys_trig_tb/TestTrig/ps_rst	1
/phys_trig_tb/TestTrig/r_count	4 4 0…
/phys_trig_tb/TestTrig/r_out	16
/phys_trig_tb/TestTrig/r_trig_energy_long	560000
/phys_trig_tb/TestTrig/r_trig_energy_short	32000
/phys_trig_tb/TestTrig/r_trig_out	16
/phys_trig_tb/TestTrig/r_trigger	2
/phys_trig_tb/TestTrig/r_trigger_occurred	16
/phys_trig_tb/TestTrig/rst	0
/phys_trig_tb/TestTrig/rst_trigger	1
/phys_trig_tb/TestTrig/sum_valid	3
/phys_trig_tb/TestTrig/triggered	1
/phys_trig_tb/TestTrig/wait_trigger	15
/phys_trig_tb/out_energy_long	560000
/phys_trig_tb/out_energy_short	32000
/phys_trig_tb/TestTrig/waittime	15
/phys_trig_tb/TestTrig/time_long	140
/phys_trig_tb/TestTrig/time_short	8
/phys_trig_tb/TestTrig/deadtime	450
Now	)00 ps
Cursor 1	;20 ps

Figure 3.13: Timing diagram for the ADC trigger at constant triggering.

3.5.1 ODB Settings


Online Database Browser
Find
Create
Delete
Create Elog from this page
/ Equipment / DTM / Settings / ADC Trigger / 0 /
+

Key	Value
Comment	ADC Self Trigger
ID E_low F_low	1024 (0x400)
ID E_low F_high	2048 (0x800)
ID E_high F_low	4096 (0x1000)
ID E_high F_high	8192 (0x2000)
ID E_very_high	16384 (0x4000)
Family	4 (0x4)
Enable	y
Enable Baseline Compensation	y
Energy threshold low	1400 (0x578)
Energy threshold high	1400 (0x578)
Energy threshold very high	5000 (0x1388)
Fprompt thresh for E_low	10 (0xA)
Fprompt thresh for E_high	20 (0x14)
Short time window	8 (0x8)
Long time window	140 (0x8C)
Scan time window	15 (0xF)
Dead time window	225 (0xE1)

Figure 3.14: Online data base settings for the ADC trigger.

Trigger ID: The ADC trigger has only a single channel but has 5 trigger IDs corresponding to the five zones depicted in Fig. 3.10. These trigger types are taken from trig_adc depicted in Fig. 3.11.

Zone 0: 0x400

Zone 1: 0x800

Zone 2: 0x1000

Zone 3: 0x2000

Zone 4: 0x4000

Family: 0x4 (Physics)

Enable: Turns the trigger module on/off

Enable Baseline Compensation: Will apply the baseline subtraction from the read ADC value. If disabled the thresholds will be used on the baseline subtracted reading. **Note:** The Fprompt is nonsensical with this disabled.

Energy Thresholds: 17-bit value (sum of 22 12-bit ASUMs). If the baseline subtraction is enabled then the thresholds are taken from the subtracted reading.

Low: Cutoff between noise and zones 0/1

High: Cutoff between zones 0/1 and zones 2/3

Very High: Cutoff between noise zones 2/3 and zone 4

Fprompt Thresholds: Cutoff between low and high Fprompt. Fprompt is given as an eight bit value, trigger decisions follow Eq. (3.1).

Low Energy: Fprompt separation between zones 0 and 1

High Energy: Fprompt separation between zones 2 and 3

Short Time Window: 1-50 value (1.1 μs) which sets the number of clocks the charge in the short integration window (see Fig. 3.12)

Long Time Window: 1-500 value (11 μs) which sets the number of clocks the charge in the long integration window (see Fig. 3.12)

Scan Time Window: 16-bit value which sets the number of clocks to look to see if the currently found maxima is local or general (within the scan time). The scan time is intended to ensure that the integrated value of the short window is maximized.

Dead Time Window: 16-bit value which sets the number of clocks after a trigger event wherein no more triggers will be produced to avoid retriggering on the same event. **Note:** This should be significantly larger than the long time window.

3.6 ODB

The online data base (ODB) contains all of the controls and settings for the triggers on a webpage. More information on the ODB can be found on the MIDAS site[8].

The DTM ODB settings can be found in their entirety on the twiki[9]

[8] https://midas.triumf.ca/MidasWiki/index.php/Online_Database

[9] https://www.snolab.ca/deap/private/TWiki/bin/view/Main/Dtmodb

Chapter 4

Firmware Modules & Settings

4.1 SVN Structure

The SVN structure used for the DEAP-3600 firmware depicted in Fig. 4.1 has two main logical parts, the physics payload (red) and the FMC drivers/software (blue) combined as SVN externals to the top project (green). The modules pertaining to the function of the trigger is considered the physics payload and is made up of edevel00261 (Section 4.2), edevel00260 (see Section 4.3), and parts of edevel00270 (Section 4.4). The rest of the drivers and control code is apart from the trigger itself and is beyond the scope of this manual. For more specifics on the driver firmware look at the repository[1], or contact Yair Linn[2] or Daryl Bishop[3]. The firmware pertaining to the trigger and general operation of the DTM follow.

[1] `https://edev.triumf.ca/project/deap/edevel00013`

[2] yairlinn@triumf.ca

[3] daryl@triumf.ca

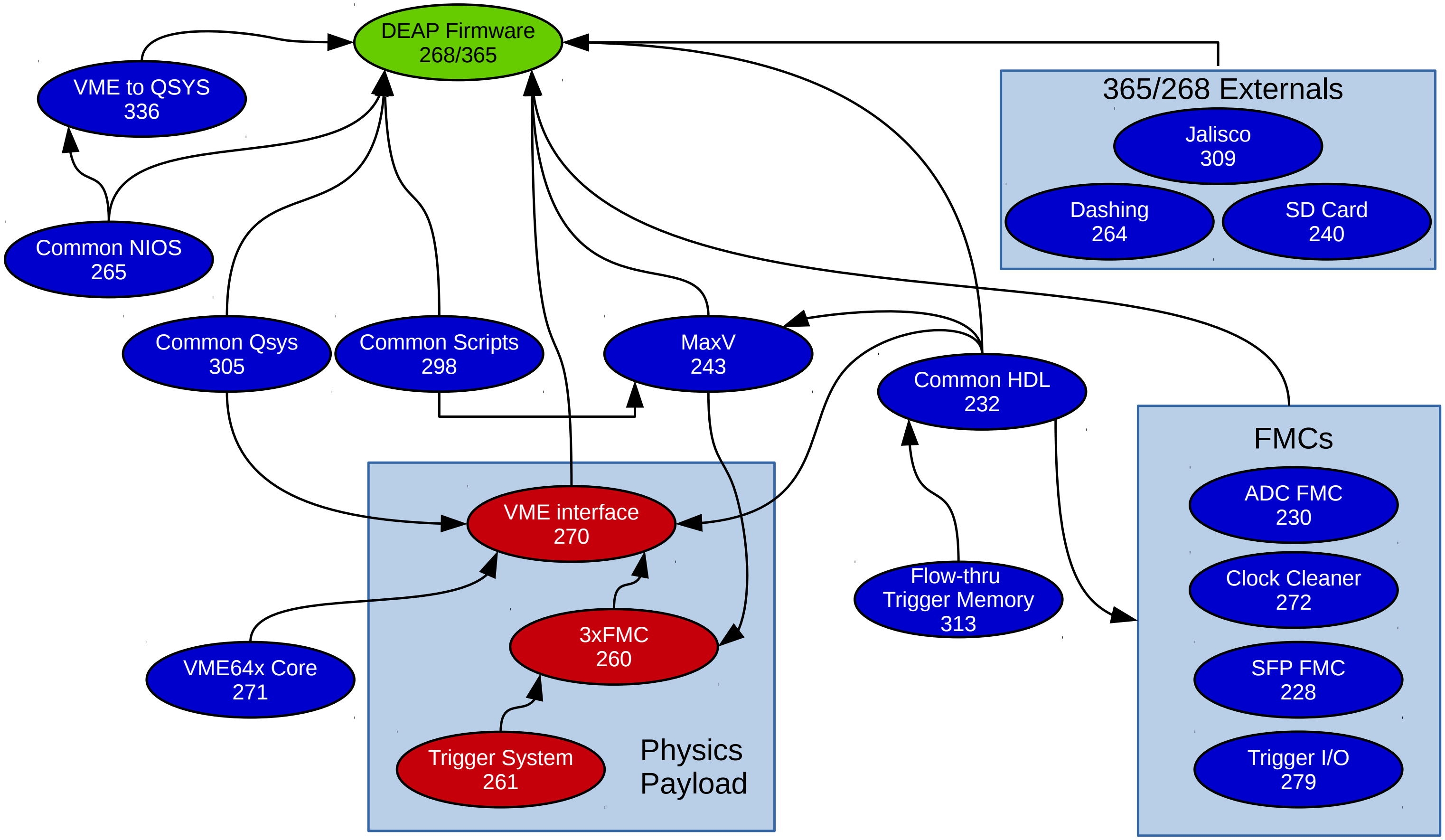


Figure 4.1: Flow chart of the firmware SVN structure. The top project is either edevel00268 or edevel00365 with the various other projects connected as externals. The modules in blue are maintained by the TRIUMF EDEV department whereas the modules in red (the physics payload) is maintained by DEAP-3600. The externals for 268 and 365 are identical as shown.

4.2 Trigger System (edevel00261)

SVN Location: https://edev.triumf.ca/svn/edevel00261

Last Installed Tag: tags/release_054 (in edevel00268)

Externals: None

Summary: The trigger system code contains all of the trigger modules and submodules which are called into the VME interface (Section 4.4). All the trigger modules described in Chapter 3 are located in tsb/ip/rtl/source. The other trigger system modules are introduced below.

4.2.1 Baseline Compensation

Module Name: trigger_baseline

SVN Location: `https://edev.triumf.ca/svn/edevel00261/trigger_baseline.v`

Modules used in: ts_adc & min bias, deap_vme_interface_wrapper.v

The ADC baseline compensation module is a module designed for the ADC trigger (Section 3.5) source to allow the calculation of Fprompt. The minimum bias trigger (Section 3.6) is also configured for it, but there is little motivation to use it as absolute ADC threshold values are equivalently useful.

The baseline calculation is preformed with an averaging function that sums ADC data (each ASUM channel has its own compensation module) for 2^n bins (the value of n is set in the ODB, a common value is 16 bins, i.e. n = 4). The sum is then right shifted by n bits giving the average value of the 2^n bins. A default value is set in the ODB and is used as the starting baseline. This value can remain static, an external baseline can be inputed, or a dynamic calculation can be enabled.

4.2.1.1 Baseline Calculation

If placed in baseline calculation mode as is typical to allow for some drift as is expected from the PMTs, the baseline will dynamically calculated and adjusted. Every 2^n clocks the baseline is re-averaged in the manner described above. This new value is compared to the previously set baseline and the difference is recorded. This repeats with the difference between the set and re-averaged baselines being summed each time. If the difference exceeds the tolerance set in the ODB then the set baseline is incremented by one in the direction of the trend. If the difference exceeds the tolerance the set baseline is incremented by one as the baseline is tending low. If the difference is less than the negative tolerance, the set baseline is decremented by one as the baseline is tending high.

The algorithm starts to run after the DTM is booted, so before the start of the run. However, it is reset at each start of run by the frontend. The baselines of all channels are not currently[4] saved into the data structure, however the long term stability can be seen on the front end.

Although previously the ASUMSUM baseline was calculated as the sum of the ASUM baselines (as stated in the twiki[5] location), this baseline is calculated separately from the sum of the signals.

Note: The change of the ASUMSUM baseline calculation is to remove the offset caused by truncation, with 22 channels the truncation error would average 11 ADC off.

4.2.1.2 Baseline Stability

ODB Baseline Plot: `https://deapdaqgw.snolab.ca/HS/Baseline/`

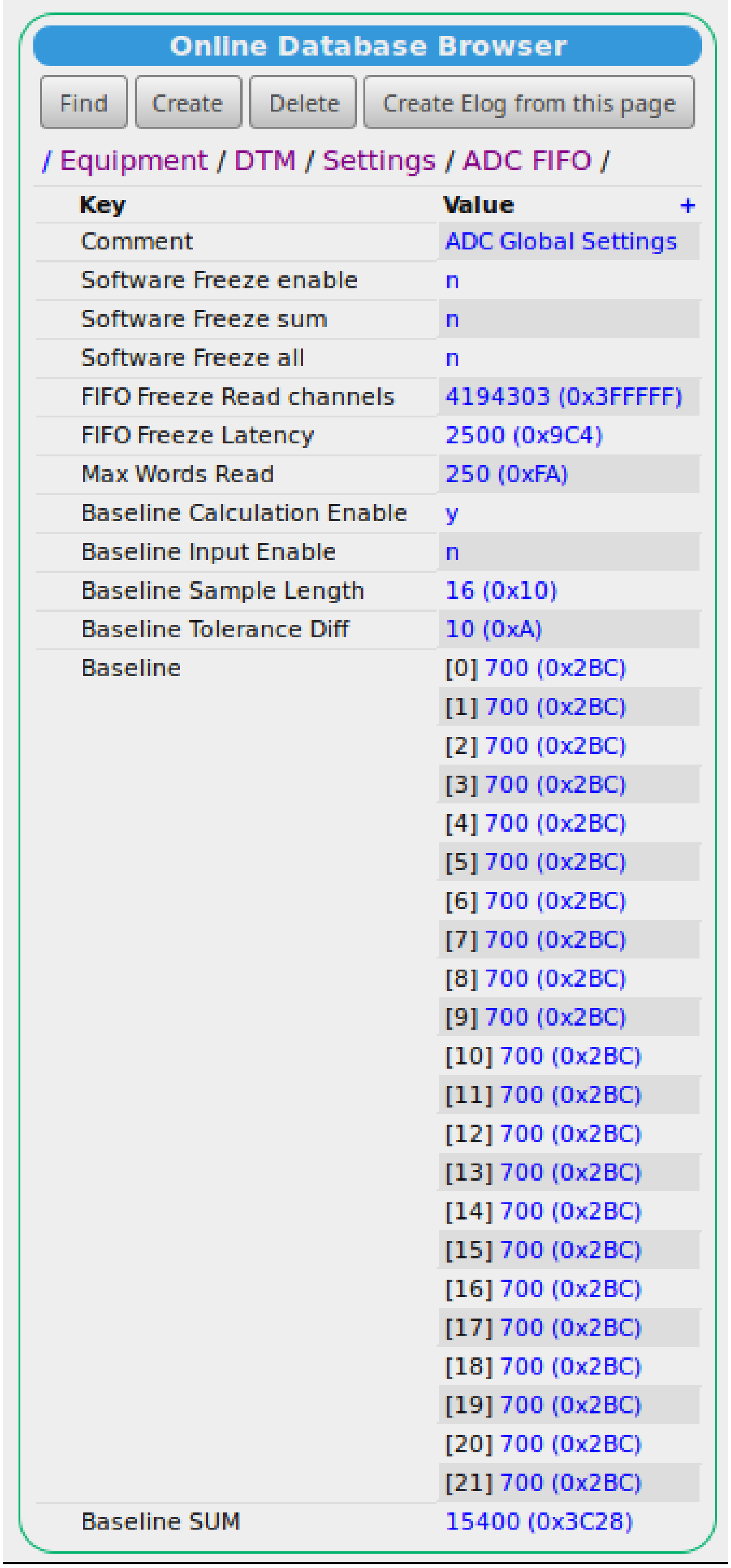
The baseline values are not currently[6] readout into the RAT data structure, however the stability can be monitered online at the location given above. Each of the ASUM baselines (Fig. 4.3 and Fig. 4.4) are monitered as well as the ASUMSUM (Fig. 4.5).

Note: The three day monitering is shown in Fig. 4.3 - 4.5 but this plot goes from 10 min to 7 days and of course can be configured to other times.

[4] July, 2016

[5] `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/DTMBaseline`

[6] July, 2016



/ Equipment / DTM / Settings / ADC FIFO /

Key	Value
Comment	ADC Global Settings
Software Freeze enable	n
Software Freeze sum	n
Software Freeze all	n
FIFO Freeze Read channels	4194303 (0x3FFFFF)
FIFO Freeze Latency	2500 (0x9C4)
Max Words Read	250 (0xFA)
Baseline Calculation Enable	y
Baseline Input Enable	n
Baseline Sample Length	16 (0x10)
Baseline Tolerance Diff	10 (0xA)
Baseline	[0] 700 (0x2BC)
	[1] 700 (0x2BC)
	[2] 700 (0x2BC)
	[3] 700 (0x2BC)
	[4] 700 (0x2BC)
	[5] 700 (0x2BC)
	[6] 700 (0x2BC)
	[7] 700 (0x2BC)
	[8] 700 (0x2BC)
	[9] 700 (0x2BC)
	[10] 700 (0x2BC)
	[11] 700 (0x2BC)
	[12] 700 (0x2BC)
	[13] 700 (0x2BC)
	[14] 700 (0x2BC)
	[15] 700 (0x2BC)
	[16] 700 (0x2BC)
	[17] 700 (0x2BC)
	[18] 700 (0x2BC)
	[19] 700 (0x2BC)
	[20] 700 (0x2BC)
	[21] 700 (0x2BC)
Baseline SUM	15400 (0x3C28)

Figure 4.2: Online data base settings for the ADC FIFO and baselines.

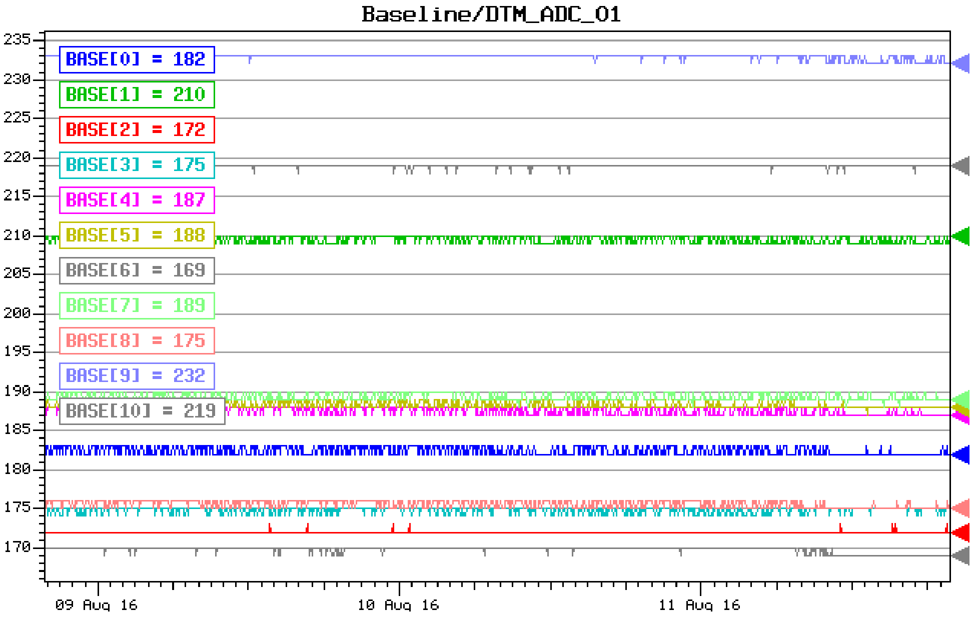


Figure 4.3: Long time plot of the first 11ASUM baselines over a 3 day period. Each baseline is continuously calculated and allowed to change.

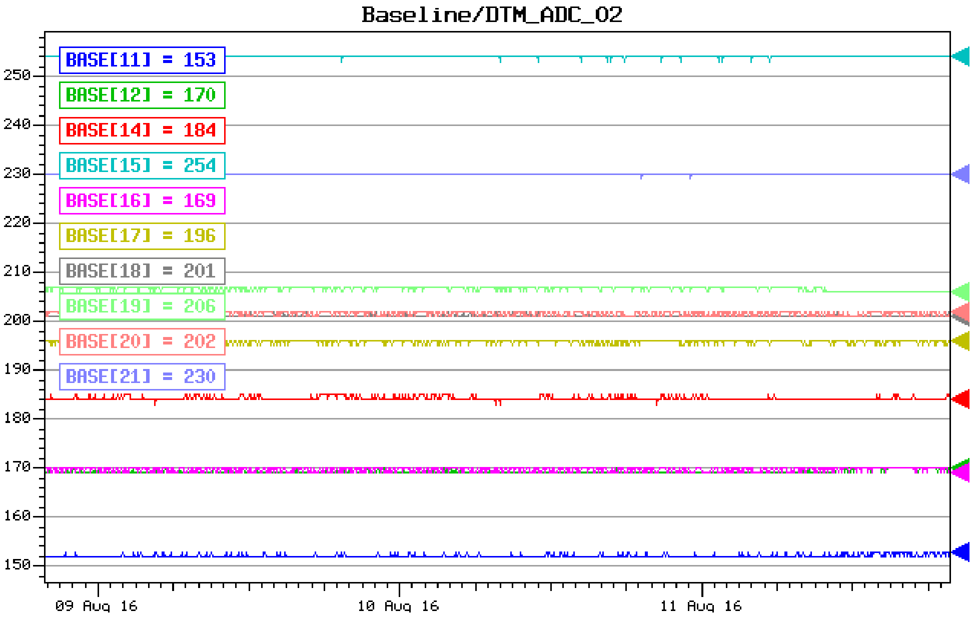


Figure 4.4: Long time plot of ASUMs 11-21 baselines over a 3 day period. Each baseline is continuously calculated and allowed to change.

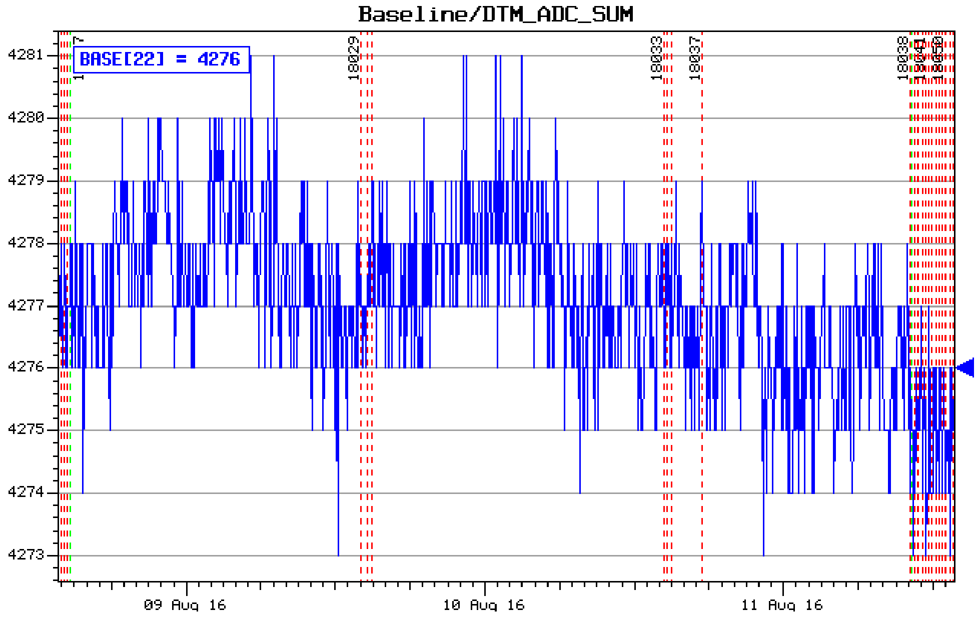


Figure 4.5: Long time plot of the ASUMSUM baseline over a 3 day period. The baseline is continuously calculated and allowed to change.

4.2.2 Self Sum Module

Module Name: trigger_self_sum_adc

SVN Location: tsb/ip/rtl/trigger_self_sum_adc.v

Modules used in: TriggerSourceADC

Summary: Returns the integrated charge within a defined short and long time window of the baseline subtracted (see above) ASUMSUM continuously. Both integrations start at the same time, so the short time window result is held in a FIFO with respect to the end of the long time window so that both results are present at the same time. The maximum values for windows are 50 DTM clocks for the short window (TICKS_SHORT = 50) and 500 DTM clocks for the long window (TICKS_LONG = 500). At 22.14 ns these max times are 1107 ns and 11070 ns respectively.

4.2.3 Trigger Self Decision Module

Module Name: TriggerSelfDecision

SVN Location: tsb/ip/rtl/trigger_self_decision.v

Modules used in: TriggerSourceADC

Summary: The self decision module uses the short and long integrated ASUMSUM windows passed by the trigger_self_sum_adc module (See above). As described in Section 3.5 and shown in Fig. 3.10 there are 4 energy regions: very low (or noise), low, high, and very high with the low and high regions being separated by their Fprompt. Energy selection is done by simply comparing the set threshold to the short energy from the self sum module. The energy is sorted into very high, then high, then low. The Fprompt selection is done by left shifting the short charge by eight bits and comparing that value to the Fprompts for each region (as set in the ODB as a value [1-256]) multiplied by the long window charge. The highest energy followed by highest Fprompt is set first.

4.2.4 Trigger Top Module

Module Name: DEAPTriggerTop

SVN Location: tsb/ip/rtl/trigger_top.v

Modules used in: vme wrapper

Summary: Trigger top level module. Calls TriggerNIMIO, TriggerEventArbiterWithFIFO, and TriggerPrescalerWithFIFO. This module looks after what happens after one of the enabled triggers puts out a signal, contains the NIM output control. Runs on the 62.5 MHz DAQ clock. Prescales events and takes care of collisions of events in TriggerEventArbiterWithFIFO. The event data written to the arbiter FIFO is passed to VME_MAPPED_REGFILE_STATUS which is then read out by MIDAS.

Note: It was being talked about adding a shutdown if events come too quickly, that could either be done here or in TriggerEventArbiterWithFIFO.

4.2.4.1 Trigger Event Arbiter With FIFO

Module Name: TriggerEventArbiterWithFIFO

SVN Location: tsb/ip/rtl/trigger_event.v

Modules used in: DEAPTriggerTop

Summary: Deals with what event from the prescalers gets sent through and written to a FIFO. Reads through events in prescalers by Round Robin Arbitration (One clock cycle per prescaler). Checks all the other prescalers if an event with the same timestamp is available if they are then they are merged if not, but another event with an earlier timestamp is available, buffer the first event and go to the earlier one. If no other event with the same or earlier timestamp is available then the event is written to the FIFO. This method ensures that although there are several different triggers, the ones written out to MIDAS are all is sequential order.

Note: It was being talked about adding a shutdown if events come too quickly, that could either be done here or in DEAPTriggerTop.

4.2.4.2 Trigger Prescaler With FIFO

Module Name: TriggerPrescalerWithFIFO

SVN Location: tsb/ip/rtl/trigger_prescaler_fifo.v

Modules used in: DEAPTriggerTop

Summary: Calls TriggerPrescaler and writes a FIFO with all of the trigger information for readout by TriggerEventArbiterWithFIFO. The trigger information is saved in a FIFO until called. Each trigger type has its own FIFO of nominally 16 words deep. If the module receives a trigger it will write the trigger information to the FIFO and the sequential events will be readout by the arbiter.

4.2.4.3 Trigger Prescaler

Module Name: TriggerPrescaler

SVN Location: tsb/ip/rtl/trigger_prescaler.v

Modules used in: TriggerPrescalerWithFIFO

Summary: Counts the number of triggers that come in and output one once the set prescale amount is met. Syncs the various event builder data so that the output of the trigger is valid with the values of the short/long ADC integrated charge or the number of coincidence channels from the Min. bias trigger.

4.2.4.4 Trigger NIM I/O

Module Name: TriggerNIMIO

SVN Location: tsb/ip/rtl/trigger_nim_io.v

Modules used in: DEAPTriggerTop

Summary: Trigger output module. This module receives a trigger signal in for one of its 8 output channels. The module checks to see if the suppression time since the last output on that channel has been exceeded yet, if it has and the busy signal is false the trigger count for that channel is incriminated and a trigger signal is sent out of the designated output. If the busy input on the NIM I/O FMC is high then no trigger output can occur[7].

4.3 3xFMC (edevel00260)

SVN Location: https://edev.triumf.ca/svn/edevel00260

Last Installed Tag: tags/t_a_release_086 (in edevel00268)

Externals: MaxV (edevel00243), TriggerSystem (edevel00261)

Summary: Contains the NIOS core for the exponential trigger, the software which generates the exponential random number and the global parameters used in the trigger system (parameters.v).

Note: Contains the global parameters for the trigger system in parameters.v

4.4 DEAP VME Interface (edevel00270)

SVN Location: https://edev.triumf.ca/svn/edevel00270

Last Installed Tag: tags/t_a_release_101 (in edevel00268)

Externals: Common QSYS (edevel00305), Common HDL (edevel00232), VME64x Core (edevel00271), 3xFMC (edevel00260)

Summary: The VME Interface is the top SVN project of the physics payload as shown in Fig. 4.1. Although not containing any of the actual FMC drivers themselves, this project encapsulates all of the data taking and processing as well as all of the triggers and event selection.

[7] For more on the busy signals see the twiki: `https://www.snolab.ca/deap/private/TWiki/bin/view/Main/InfoBusy`

4.4.1 VME Interface Wrapper

Module Name: deap_vme_interface_wrapper

SVN Location: tsb/ip/rtl/deap_vme_interface_wrapper.v

Modules used in: VME_3xFMC_Carrier (edevel00365 or edevel00268 in tsb/ip/rtl/VME_3xFMC_Carrier

Summary: This is the top level verilog file of this subproject. Instantiates the DEAP trigger sources, ADC VME FIFOs, DEAP trigger top module, exponential trigger NIOS, VME registers and their assignments. Anything related to the ADC Waveform FIFOs is to be done in this file. This module contains all of the physics level firmware.

4.5 DEAP Firmware Development Quartus 13.0sp1 (edevel00268)

SVN Location: https://edev.triumf.ca/svn/edevel00268

Last Installed Tag: tags/t_a_release_206

Externals: JALISCO (edevel00309), Dashing (edevel00264), SD Card (edevel00240), VME to QSYS (edevel00336), MaxV (edevel00243), Common NIOS (edevel00265), Common QSYS (edevel00305), Common Scripts (edevel00298), Common HDL (edevel00232), ADC FMC (edevel00230), Clock Cleaner (edevel00272), SFP FMC (edevel00228), Trigger I/O (edevel00279), VME Interface (edevel00270)

Summary: Currently edevel00268 is the top SVN project for DEAP-3600. Each of the preceding files and projects described above (excluding edevel00268) are included as externals. 268 uses the master clock distribution FMC board (see Section 2.2.2) with daisy chain clocking. 268 has experienced issues operating on the 3xFMC Rev. A board and therefore is only used with the Rev. B board (See the Errata 7). See Section 2.4 for the applicable hardware configuration. The programming scripts for the FPGAs are located in tsb/ip/scripts[8] with the binary files to be loaded held in tsb/ip/exe.

Note: It is crucial to make sure the correct FMCs are connected, **do not** load 268 with the SFP and Mini-Sas board attached and **do not** load 365 with the Clock distribution FMC.

4.5.1 VME 3xFMC Carrier

Module Name: VME_3xFMC_Carrier

SVN Location: tsb/ip/rtl/VME_3xFMC_Carrier.v

Modules used in: N/A (Top Project Module)

Summary: This is the very top module of the DEAP-3600 firmware, and instantiates the deap_vme_interface_wrapper module. This module is maintained exclusively by Yair Linn[9].

4.6 DEAP Firmware Quartus 14.1 (edevel00365)

SVN Location: https://edev.triumf.ca/svn/edevel00365

Last Installed Tag: tags/t_a_release_130

Externals: JALISCO (edevel00309), Dashing (edevel00264), SD Card (edevel00240), VME to QSYS (edevel00336), MaxV (edevel00243), Common NIOS (edevel00265), Common QSYS (edevel00305), Common Scripts (edevel00298), Common HDL (edevel00232), ADC FMC (edevel00230), Clock Cleaner (edevel00272), SFP FMC (edevel00228), Trigger I/O (edevel00279), VME Interface (edevel00270)

Summary: Like edevel00268, edevel00365 is the top SVN project for DEAP-3600, however the upgrade has not yet occurred[10]. Each of the preceding files and projects

[8] see Chapter 5 for programming

[9] yairlinn@triumf.ca

[10] Upgrade date as yet unset (as of July 2016)

described above (excluding edevel00268) are included as externals. 365 uses the SFP and Mini-Sas FMC board (see Section 2.2.3) adding FTP connectivity to the DTM (see Chapter 5). Additionally 365 has tighter timing constraints which has fixed the issues experienced with using 268 on the 3xFMC Rev. A board so may operate on either board (See the Errata 7). See Section 2.4 for the applicable hardware configuration. The programming scripts for the FPGAs are located in tsb/ip/scripts[11] with the binary files to be loaded held in tsb/ip/exe.

Note: It is crucial to make sure the correct FMCs are connected, **do not** load 268 with the STP and Mini-SAS board attached and **do not** load 365 with the clock distribution FMC.

4.6.1 VME 3xFMC Carrier

Module Name: VME_3xFMC_Carrier

SVN Location: tsb/ip/rtl/VME_3xFMC_Carrier.v

Modules used in: N/A (Top Project Module)

Summary: This is the same module as used in edevel00268 with several significant changes. This module is maintained exclusively by Yair Linn[12].

[11] see Chapter 5 for programming

[12] yairlinn@triumf.ca

Chapter 5

Installation and Operation

5.1 JALISCO

EDEV Project: `https://edev.triumf.ca/projects/edevel00309`

EDEV Wiki: `https://edev.triumf.ca/projects/edevel00309/wiki`

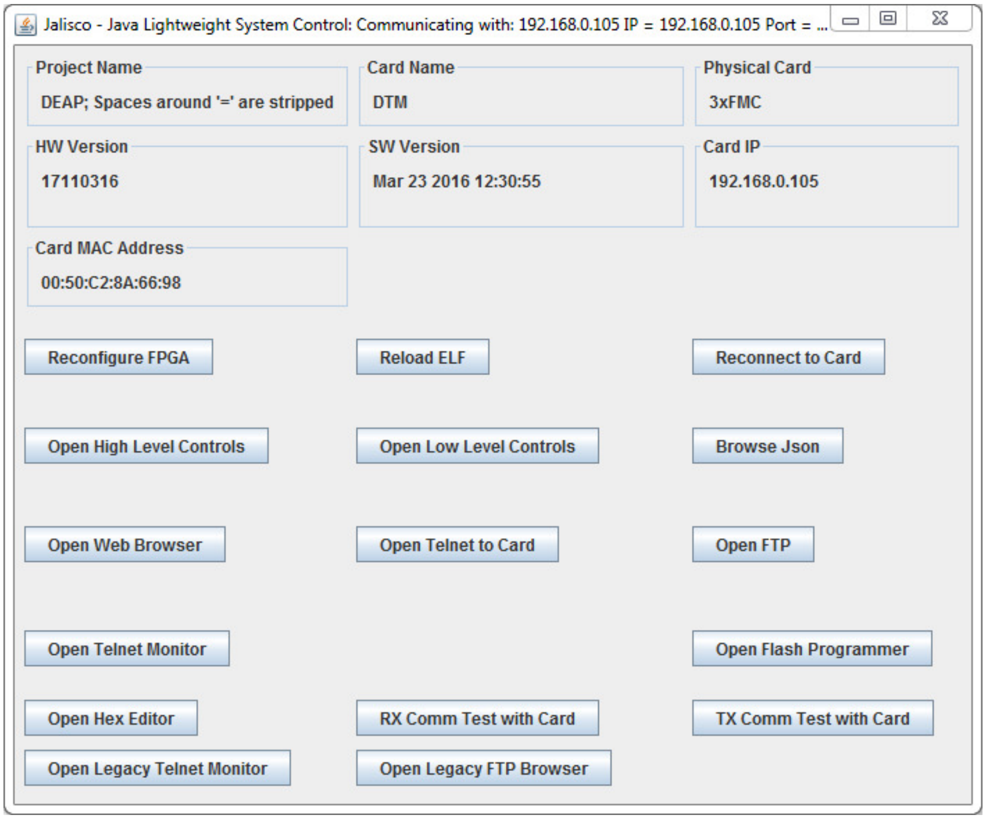
Tutorial Videos: `https://edev.triumf.ca/projects/edevel00365/wiki/Jalisco_Tutorial_Videos`

The Java Lightweight System Console, or JALISCO, is a Java based GUI that controls most aspects of the DTM hardware boards. Currently used in various projects such as DEAP-3600, JALISCO is unique in that it automatically generates its GUI based on the autodiscovery of the register files inside the firmware. So if the register exists in the firmware, it can be readout and altered by JALISCO.

JALISCO adds a GUI interface to the VME readout for all read/write and read-only DTM registers for the trigger and FMCs. The ADC's can be read out directly, allowing debugging of any hardware issues separate from the firmware. See the wiki and the tutorial videos for more information. JALISCO is maintained by Yair Linn[1].

The JALISCO startup screen is shown in Fig. 5.1. Once JALISCO is started and the appropriate host computer and port chosen, this GUI pops up.

[1] yairlinn@triumf.ca



Jalisco - Java Lightweight System Control: Communicating with: 192.168.0.105 IP = 192.168.0.105 Port = ...

Project Name: DEAP; Spaces around '=' are stripped

Card Name: DTM

Physical Card: 3xFMC

HW Version: 17110316

SW Version: Mar 23 2016 12:30:55

Card IP: 192.168.0.105

Card MAC Address: 00:50:C2:8A:66:98

Figure 5.1: Main menu of JALISCO following startup.

5.2 Loading Firmware

Note: See the EDEV page for more information.

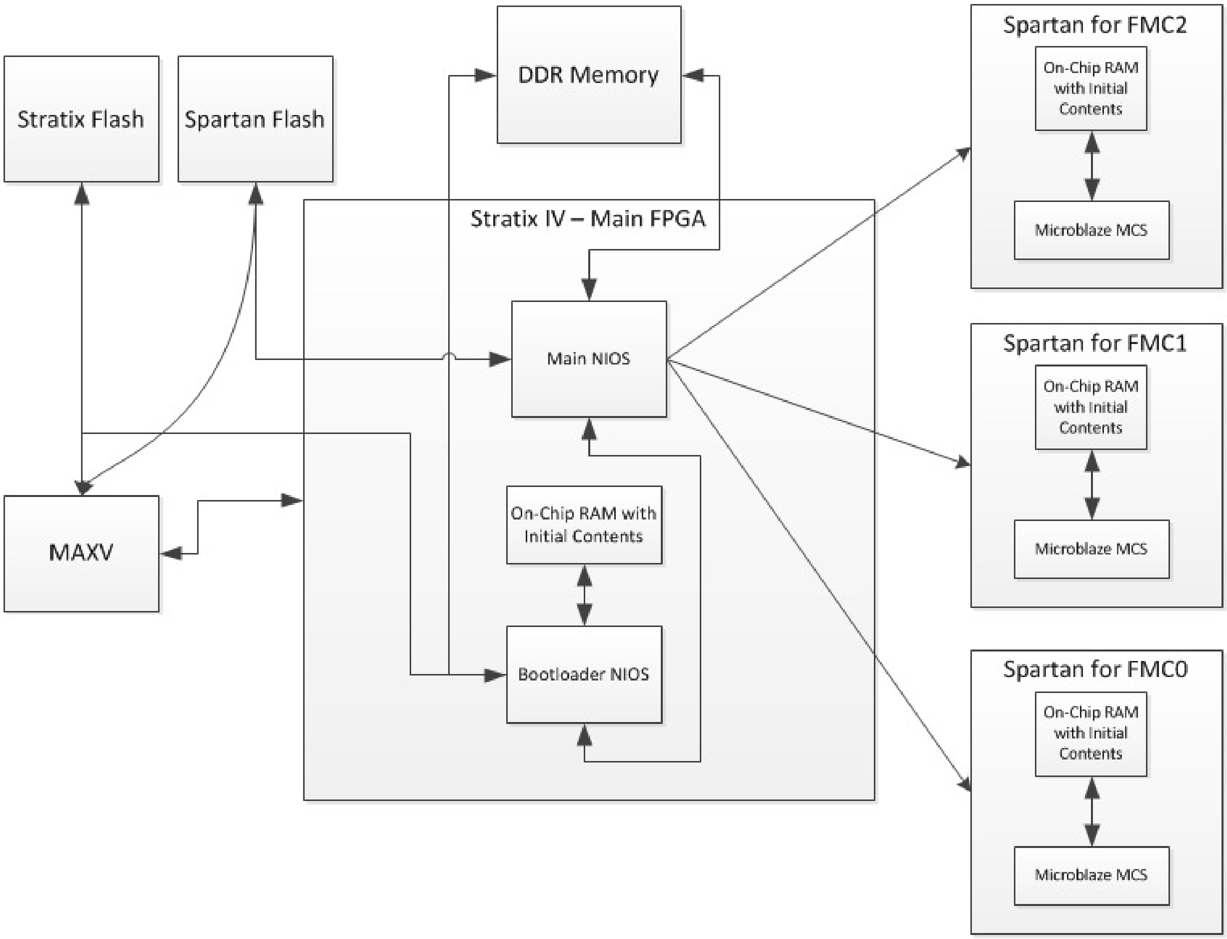


Figure 5.2: General layout of the FPGA, Embedded Processors, and Memory Devices on the DTM. Photo Credit EDEV Department (`https://edev.triumf.ca/projects/edevel00365/wiki/Boot_Sequence_and_Boot_Image_Selection`).

The installation of new firmware is done one of three different ways[2]: JTAG, FTP/Telnet, & via JALISCO. The safest method to update the firmware is over FTP. There are two versions of each flash image loaded as outlined in the following section (5.2.1). Once the various FPGA images are loaded into the respective flash locations, the power is cycled loading the new firmware.

[2] To date (July 2016) only JTAG loading is used

Note: Once a new image has been loaded check the JALISCO timestamp (see Section 5.1)

5.2.1 Dual Load Images

There are two images loaded onto the flash normally. The general concept is that there is a normal image or what should be used in normal operation and the image that is updated using JALISCO or FTP, and a fallback image that, for safety, cannot be modified via JALISCO nor FTP (only via JTAG) so it is far less susceptible to corruption. If an issue is detected with the primary image the secondary one will be used.

There are three parts to a complete image load and it is crucial that the correct pairing of files is done.

1. Hardware (.pof) image of the Stratix FPGA
2. Software Image for the Main Nios (.elf)
3. Software and Hardware images for Spartans on FMC 0, FMC1, and FMC2 (.hex)

Compiling of a new firmware release should be requested from the primary developer (Yair Linn[3]). The address map of the Stratix and the Spartan Flash images are shown in Fig. 5.3 and 5.4.

On bootup the DTM will pull these images from the flash and make a decision on which to use[4]

5.2.1.1 Updating and Verifying Images

EDEV Page: Link[5]Link

[3] yairlinn@triumf.ca

[4] For more on the startup sequence before this step see `https://edev.triumf.ca/projects/edevel00365/wiki/Boot_Sequence_and_Boot_Image_Selection#Advanced-behind-the-scenes-view-of-boot-sequence`

[5] `https://edev.triumf.ca/projects/edevel00365/wiki/Boot_Sequence_and_Boot_Image_Selection#Updating-and-Verifying-the-images`

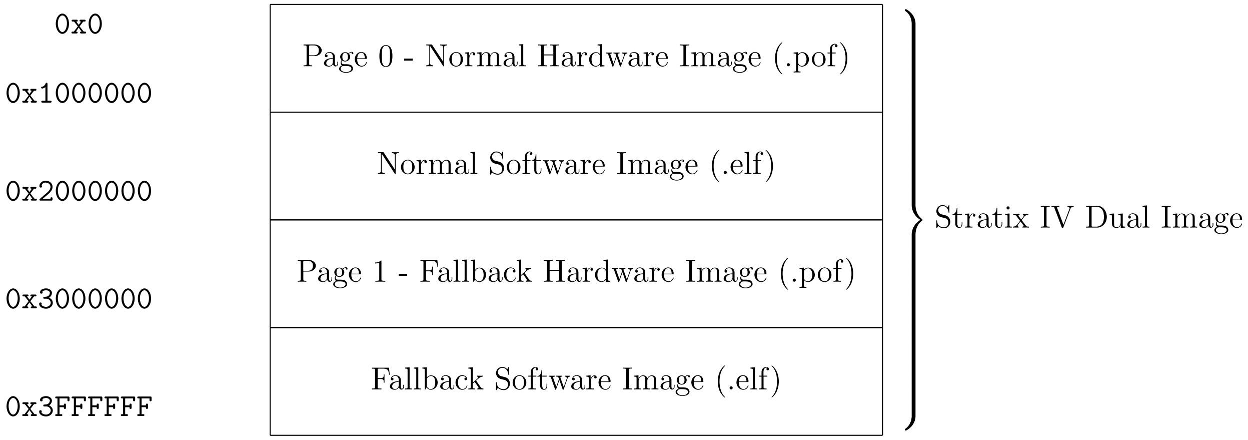

0x0
0x1000000
0x2000000
0x3000000
0x3FFFFFF
Page 0 - Normal Hardware Image (.pof)
Normal Software Image (.elf)
Page 1 - Fallback Hardware Image (.pof)
Fallback Software Image (.elf)
Stratix IV Dual Image

Figure 5.3: The address map of a Stratix IV flash file.

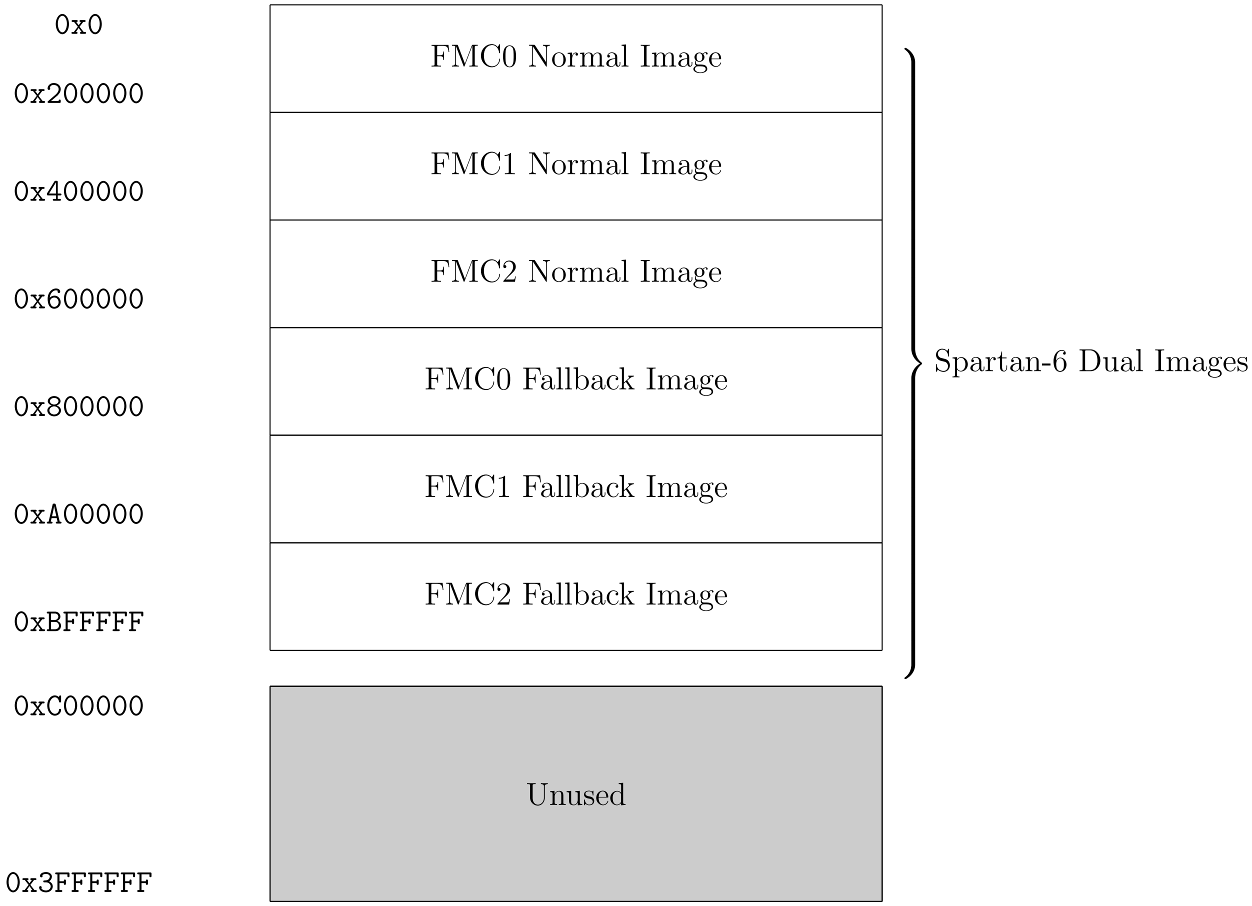


Figure 5.4: The address map of the flash file for the three FMC boards

In order to avoid accidental overwriting of the fallback image, the only way to modify the fallback image is with a complete programming of the flashes via JTAG via the do_program_stratix_flash.cmd and do_program_spartan_flash.cmd commands. These commands each program their entire respective images as shown in Fig. 5.3 & 5.4. Both the normal and fallback images are reflashed. It is intended that reprogramming via JTAG should be used infrequently, leaving the fallback image the same with the normal images being updated via JALISCO or FTP.

On bootup the loaded images are checked and verified, if the normal images are found to have issues then the fallback images are used instead to ensure bootability.

5.2.1.2 Choosing Flash Images

EDEV Page: Link[6]

Choosing which of the two images will be used is done by running a script from the Nios Command Shell (located in tsb/ip/scripts).

Load the normal image: source do_switch_to_normal_flash_image.cmd

Load fallback image: source do_switch_to_fallback_flash_image.cmd

These scripts program the MaxV on-chip User Flash Memory via the JTAG with the correct settings to select the appropriate image. The script needs to be run only once; bootups after will default to this selected image until the other script is run to select the other image. These scripts take about 10 seconds to run, resulting in fast, easy switching between images.

[6] https://edev.triumf.ca/projects/edevel00365/wiki/Boot_Sequence_and_Boot_Image_Selection

Chapter 6

Appendix

6.1 Resources & Documentation

6.1.1 Deap & TRIUMF Documents

Document	Date	Description	Location
DEAP-3600 Electronics & DAQ Technical Design Report	04/25/2012	Overview of the first DAQ and electronics structure, some changes have been made	Link

6.1.2 Manuals & Data Sheets

Document	Date	Description	Location
Altera Stratix IV Overview		Infomation of the Stratix IV family of FPGAs	Link
Samtec FMC Overview		FMC standard description	Link
Spartan-6 Family Overview		Data sheet for the Spartan-6 FPGAs used for the FMCs	Link
Altera MaxV Handbook		Operations handbook for the Altera MaxV FPGA	Link
ADC Data Sheet		24 Channel FMC ADC	Link

6.1.3 Links & Websites

Document	**Date**	**Description**	**Location**
Clock Distribution Module		EDEV project location of the VME Clock distribution module	Link
FMC HDL Submodules		EDEV project location of several of the ADC board, the SFP board, and the clock cleaner HDL files	Link

Chapter 7

Errata

7.1 JTAG Issues

7.1.1 USB Blaster Revision

USB Blaster User Guide: `https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_usb_blstr.pdf`

Issues have been noted using the JTAG USB blaster. It is unclear if the issue is due to the Linux software, however Rev.B of the JTAG USB Blaster wouldn't work at SNOLAB but switching to Rev.C fixed this issue. It is advised to continue using Rev.C.

7.1.2 NIOS Command Shell Version

USB Blaster User Guide: `https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_usb_blstr.pdf`

When loading the firmware it is necessary to use the correct nios2_command_shell.sh version. For edevl00268 the Version 13.0sp1, Build 232 works reliably. However for edevel00365 (compiled with Quartus 14.1), the JTAG programming seems to be more reliable using the Quartus Version 14.1, Build 190 NIOS2 Command Shell as opposed to the 13.0sp1 with every load succeeding instead of having to flash the card a couple of times for it to work.

Note: It has been previously stated that Altera Quartus Version 13.0sp1, Build 232 is used to compile hardware/software and for flash programming, while due to an Altera bug Version 12.1sp1 Build 243 is needed in order to run the system console script

7.2 VME Motherboard Version A

Edevel00268 has never worked on the Rev.A 3xFMC motherboard and only runs on the Rev.B board even though changes between the boards are minimal. Edevel00365 however has been confirmed to work interchangeably on the revisions.

7.3 Load Scripts Failing

The flash is not always programmed correctly with the JTAG (other loading methods have yet to be throughly tested), so it is essential to check the firmware timestamp (HW Version HHDDMMYY format) on JALISCO after a flash has been preformed. Additionally to ensuring that the correct install has been made the timestamp will hint as to which of the two images is being used. With a quick glance at the hardware timestamp (see Fig. 5.1) a lot of time and suffering can be averted.

Bibliography

[1] *CAEN Technical Information Manual, Module V1720.*

[2] *CAEN Technical Information Manual, Module V1740.*

[3] DEAP-3600 Collaboration. “DEAP-3600 Electronics & DAQ Technical Design Report”.

[4] DEAP-3600 Collaboration. “The DEAP-3600 Dark Matter Experiment”. Feb. 2016.

[5] TRIUMF DAQ Group. *VME-PPG32 - pulse pattern generator VME FPGA board.* URL: http://www.triumf.info/wiki/DAQwiki/index.php/VME-PPG32.

[6] *Midas documentation.* Jan. 26, 2016. URL: https://midas.triumf.ca/MidasWiki/index.php/Midas_documentation.

[7] F. Retiere et al. “DEAP trigger and readout electronics”. In: *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE.* Oct. 2012, pp. 1802–1805. DOI: 10.1109/NSSMIC.2012.6551421.