In presenting this thesis in partial fulfilment of the requirements for an advance degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of Graduate Studies, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature ______________________

Date __________________________

# Region 1 Detector
# Jefferson Lab Qweak Experiment

by
Warren Parsons

A thesis
submitted in partial fulfillment
of the requirements for the degree of
Master of Science in the Department of Physics
Idaho State University
May 2010

To the Graduate Faculty:

The members of the committee appointed to examine the thesis of Warren Parsons find it satisfactory and recommend that it be accepted.

Major Advisor

Committee Member

Graduate Faculty Representative

## Acknowledgements

There are many gifted people involved with this thesis without whom the completion thereof could not have been effected as quickly and as successfully as it was. Without them this project would have come to a grinding halt long ago; I feel compelled to express my heart-felt appreciation for all the hard work, time, and effort that they each put into this project.

To Dr. Forest, I would like to express my gratitude for giving me the opportunity to participate in this challenging project; thank you for your patience, wisdom, and understanding during the span of this project.

Thank you Tamuna for taking the time to understand and operate many of the various electronic components of this project most of which you did not have the luxury of offering input into their design. Had you had the chance, many of the designed components might have made sense to a normal person.

Brian, thank you for agreeing to be a part of a project for which there was no direct recognition or benefit other than the experience gained and the joy of creation. You are an inspiration to me going forward.

Thanks must also be given to all those who had a hand in this project, upon who's "shoulders" I built my portion of this detector. Indeed, I do not believe it is possible to accomplish something of this magnitude without the direct influence of many people.

For supporting me through a difficult phase of my life, I thank my family, including my wife, children, parents, brothers, and sisters. Your overall perseverance – which flies in the face of mainstream society with its quick fixes and numerous unsubstantiated mores – made this excursion possible.

Lastly, although not an official "thank you" to an individual, I would like to dedicate this thesis to science, critical thinking, and truth; the accelerated pursuit of these during this time period has left an indelible impression upon me that I hope to use as an impetus to effect positive change in the world.

# Contents

# List of Figures

# List of Tables

# List of UML Diagrams

# Abstract

The Qweak experiment at Jefferson Lab will make a precision measurement of the weak mixing angle using parity-violating electron scattering from the proton. The Qweak apparatus will use a tracking system to measure the electron elastic scattering profile using a set of detectors that are separate from the main detector systems.

The Region 1 detector of the Qweak experiment is the first detector encountered by scattered particles in this tracking system. The R1 detector uses Gas Electron Multiplication (GEM) to amplify the ionization signal generated by scattered electrons traveling through the detector. The resulting cloud of charge generated by these preamplifiers is directed, by an external electric field, towards a charge collector which allows a determination of the location of the deposited charge.

A signal processing integrated circuit, design at CERN for the TOTEM experiment, amplifies and discriminates the analog charge deposited on the charge collector converting it into a digital yes/no hit signal. This serialized data is then transferred to permanent data files. Once in this form, these files can then later be parsed to determine position of particles passing through the detector.

The focus of this thesis is on the process of querying the VFAT ICs on the GEM detectors, collecting all of the simultaneous data packets from the VFATs into one data structure using an FPGA (Field Programmable Gate Array), and then transferring this large data file to a Linux-based computer for future parsing.

# Chapter 1

# Introduction

## 1.1 Purpose and Scope

This thesis describes a system capable of digitally recording the analog charge deposited on the Region 1 tracking system detectors used by the Qweak experiment at Jefferson Lab. Charged particle position information is made available by using a process known as Gas Electron Multiplication (GEM) whereby the original particle generates an avalanche of charged particles large enough to be detected by large-scale electronics. The final stage of this GEM detector contains analog voltage signals on individual copper strips within the charge collector. The analog signals on these individual strips are processed by a custom-made integrated circuit (IC), called VFAT2 (or VFAT), containing preamplifiers and pulse-shaping networks. [1][2][3]

For the purpose of recording the content of multiple VFAT cards simultaneously, it was necessary to use the V1495 Field Programmable Gate Array module built by CAEN which operates on a VME backplane.[4] The block diagram for the Region 1 detector is shown in Figure 1-1 on page 2.

Where necessary this thesis will reference and attempt to explain other portions from various manuals; all of these pertinent manuals are included on a CD with the original copy of this thesis. For the user firmware and the ROC (Read Out Controller) portions of the detector, UML diagrams have been included to simplify the process of familiarizing the reader with this detector and its respective algorithms for data acquisition from the VFATs. Unfortunately, due to the finite nature of this thesis, there are many assumptions that are made about the level of technical proficiency of the reader; hopefully, these are minimal, but certainly they are unavoidable.

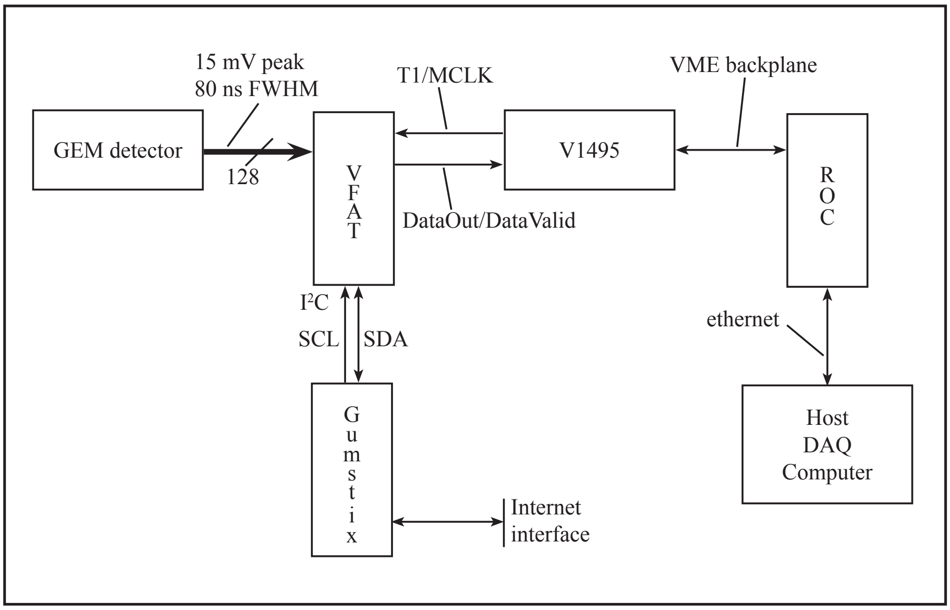


**Figure 1-1: Block diagram for Region 1 detector electronics**

The main portion of this thesis begins where the analog voltage signals exit the GEM detectors and ends where the data is transferred to the Linux DAQ computer. The most important topics, for which discussion is necessary to understand the full scope of this project, will be briefly touched upon in this introduction.

## 1.2 Region 1 Detector Description

The Region 1 Detector is a particle position detector for the Qweak experiment at Jefferson Lab. A graphical depiction of the Qweak experiment, along with the relative location of the Region 1 detector within this experiment, can be seen in Figure 1-2 on page 3. Very precise cylindrical position can be obtained from a particle by a process known as Gas Electron Multiplication (GEM). This process is a well-established technology that uses two different processes to obtain gains of ~$10^6$ charged particles.

The first process the GEM detector uses is ionization of an otherwise inert gas via inelastic scattering. These liberated charged particles are accelerated by a static electrical field of ~3 kV/ cm. Because either process can be responsible for starting the ionization process, either high-energy charged particles or photons can be detected.

After the initial ionization process, ionized particles are accelerated towards insulating Kapton layers sandwiched between layers of copper. This copper-Kapton-copper assembly is then placed within a series of other similar assemblies and electronically biased at a very high static voltage. As electrons come into close proximity with the very high electric fields of the perforated holes in the Kapton assembly, even more electrons are released due to the acceleration the particles undergo thereby causing even more ionization events. Figure 1-3 on page 4 depicts these perforated holes along with the field lines that they generate. Because a vast majority of these ionization events occur at the high-field regions near the holes of the copper-Kapton-copper layers, detectors of this type are useful for having both large preamplification and higher spatial resolution than many other detectors.

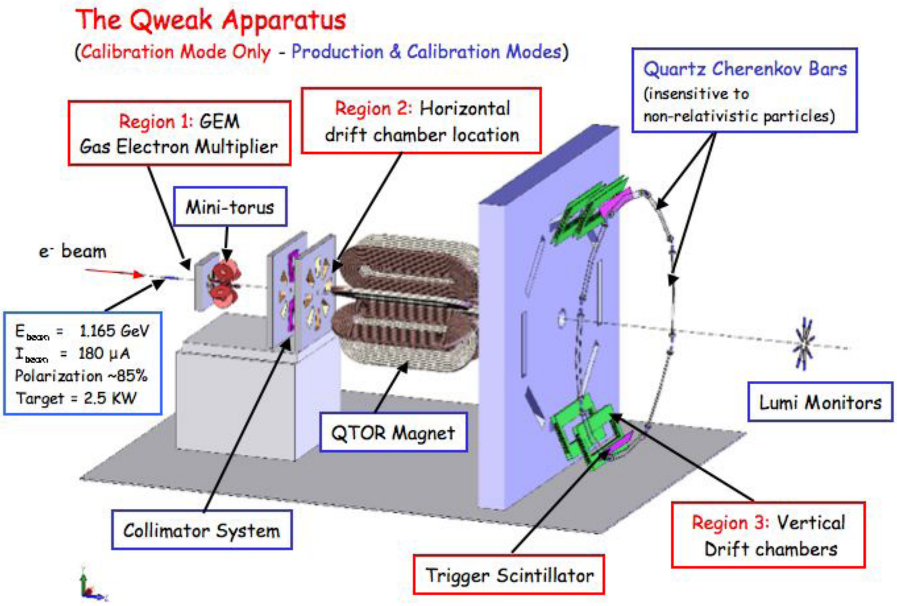

The Qweak Apparatus
(Calibration Mode Only - Production & Calibration Modes)
Region 1: GEM Gas Electron Multiplier
Region 2: Horizontal drift chamber location
Quartz Cherenkov Bars (insensitive to non-relativistic particles)
Mini-torus
e- beam
$E_{beam}$ = 1.165 GeV
$I_{beam}$ = 180 μA
Polarization ~85%
Target = 2.5 KW
Lumi Monitors
QTOR Magnet
Collimator System
Region 3: Vertical Drift chambers
Trigger Scintillator

**Figure 1-2: Graphical Depiction of Qweak experiment including the Region 1: GEM detector**

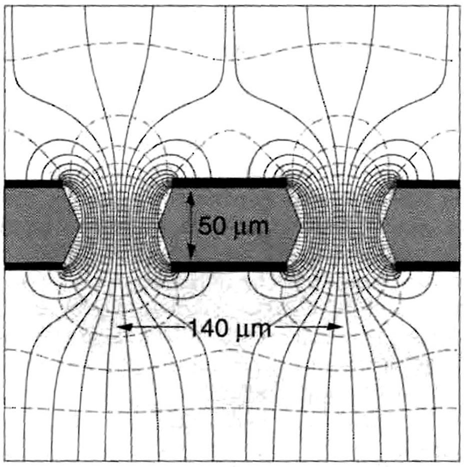


**Figure 1-3: Perforated hole structure of copper-Kapton-copper layer[12]**

## 1.3 VFAT2 Readout Card

A digital hit-or-miss comparison is based on a programmable threshold voltage after having passed through the preamplifiers and pulse-shaping networks. The VFAT cards are fast-acting integrated circuits (ICs) developed by CERN. Each one possesses the ability to process 128 channels simultaneously. The VFATs' discriminators used to identify charge levels are programmable via the VFATs' registers. All of the programmable registers of the VFAT cards are accessed using its I$^2$C interface.

### 1.3.1 Preamp/Filter/Comparator

The front-end of the VFAT consists of a transimpedance amplifier, an integrator, an adjustable differential threshold detector, a differential-mode comparator, and ends with a common-mode comparator for digital output. These different sections are shown in Figure 1-4 for demonstration purposes. Unfortunately, as will be discussed later in this section, this schematic is only for demonstration purposes and cannot possibly strictly adhere to the actual circuitry in the VFAT front-end because there is little correlation to the adjustable I$^2$C registers.

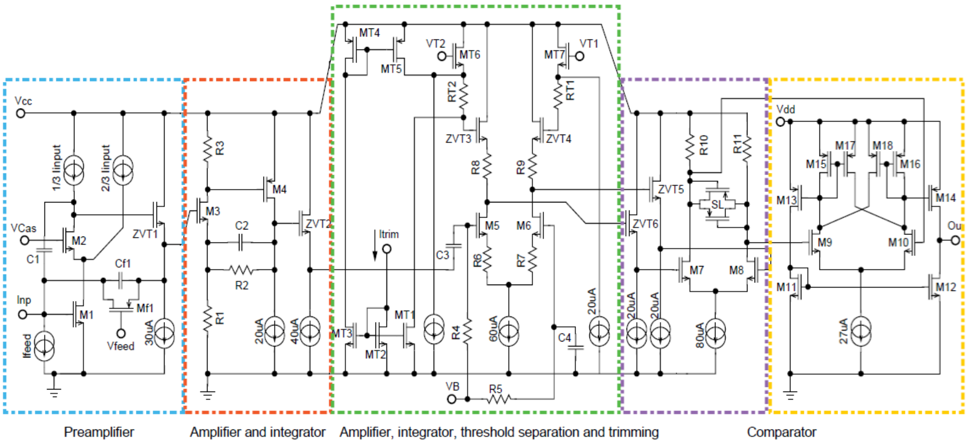


**Figure 1-4: The VFAT2 front-end with different sections outlined for clarity**

The transimpedance amplifier (shown in Figure 1-4 as surrounded by blue) converts a current (or charge) signal into a voltage signal. Adhering to the naming convention of these various amplifiers, one takes the amplifier gain as the output signal over the input signal. Therefore, taking the output signal (voltage) and dividing it by the input signal (current) provide us with the units of impedance and hence the name of the amplifier.

Per the literature on the VFAT, we know that the pre-amp is comprised of a cascode-configuration amplifier with an NMOS input transistor operating close to weak inversion. It can be surmised that operating this transistor near the weak inversion layer is necessary because this is precisely where the equivalent gate capacitance of the input transistor is at (or nearly at) a minimum – particularly at higher frequencies, i.e. 100Mhz. This is because the NMOS transistor equivalent gate capacitance is dominated by the series capacitance of the depletion layer at this point.[13][14]

Operating a transimpedance amplifier at a lower capacitance means that a greater voltage output gain can be achieved given the same amount of input charge. This follows if the input capacitance is the main (or one of the main) transimpedance gain factor. Also, the cascode configuration gives the amplifier very high gain with the only penalty being that the output voltage range decreases by the overhead of one overdrive voltage, e.g. $V_{gs}$ - $V_{th}$.

This preamplifier stage also contains a tunable active-feedback capacitance. This provides a 13ns time constant for a 0.8 μA feedback current. Using this active-feedback one can theoretically set the lower level of accepted frequencies into the pulse-shaping network.

The second distinct stage of the analog front-end is another amplifier with an integrator (shown in Figure 1-4 as red). Using this stage, the upper end of the accepted frequencies for the overall amplifier can theoretically be set.

After the integrator, the signal passes through the adjustable threshold discriminator. DC signals are blocked coming into this stage by the input capacitors. This stage appears as a differential amplifier with adjustable loads placed in the differential current paths. This clever configuration allows for adjustment of the input voltage necessary to flip the respective voltage outputs taken from the drains of the input NMOS transistors shown in Figure 1-4 as M5 and M6. Setting the proper current for this differential amplifier and then shifting the input voltage will steer the current from one side of the differential amplifier to the other thereby switching the voltage at the outputs drastically. Changing $V_{th2}$ and $V_{th1}$ causes there to be more or less voltage necessary to shift the voltage outputs. During this project we have operated under the instructions that $V_{th2} - V_{th1} > 0$ is the proper setting for detecting positive input pulses and $V_{th2} - V_{th1} < 0$ is the proper setting for detecting negative pulses. Since we are detecting pulses of accumulated electrons, we have our threshold settings set to the latter.

The final two sections of the front-end detectors provide a differential-mode comparison (shown in purple) of the output voltages from the differential amplifier followed by a single-ended comparison (shown in orange).

One difficult aspect of this project is that the schematic in Figure 1-4 can only be used to indirectly determine the values of the adjustable I$^2$C registers, found as primary registers 0x02 - 0x07. For instance, it would appear that the integrator portion of the schematic is adjustable via the IShaper or IShaperFeed registers, but one cannot made a direct correlation between the two. Not knowing exactly how to adjust these registers to tune in or tune out particular aspects of our pulses forced us to treat this aspect of the Region 1 Detector as a black box. Fortunately, we did receive special instruction on how to set these values for our experiment from Paul Aspell, one of the designers for the VFAT. Note that these actual values varied from the default settings found in the VFAT2 Operators Manual. These values are listed in Table 1-1 below.

| Register | Value |
|---|---|
| IPreampIn | 142d = 0x8E |
| IPreampFeed | 70 d = 0x46 |
| IPreampOut | 130 d = 0x82 |
| IShaper | 127 d = 0x7F |
| IShaperFeed | 85 d = 0x55 |
| IComp | 100 d = 0x64 |

**Table 1-1: Recommended I$^2$C Preamp Settings**

## 1.3.2 T1 Command Interpreter

There are four basic commands, called T1 commands. Each one of these is comprised of an LVDS, 3-bit pattern sent in sync with the IC's external clock pulse (MCLK), used to direct the action of the VFAT.[1] It must be noted that no two T1 commands can arrive closer than 3 clock periods due to their encoded nature.

| Pattern | Command Name | Function |
|---|---|---|
| 0b100 | LV1A (Level 1 Accept) | Trigger |
| 0b111 | CalPulse | Timing of calibration pulse |
| 0b110 | ReSync | Resynchronisation of all state machines |
| 0b101 | BC0 | Bunch crossing zero identifier |

**Table 1-2: T1 Commands**

The LV1A signal is the most commonly used signal; this signal tells the VFAT to search into its recorded memory for hit data and package this data into a serial packet to be read out on the LVDS DataOut line synchronously with the MCLK signal. The VFAT has two SRAM registers for "hit" information. The write address to the first, SRAM1, is incremented every clock cycle of MCLK. When a LV1A signal is sent to the memory module, the hit record contained in the current register minus the latency value, I$^2$C extended register 0x00, is transferred from SRAM1 to SRAM2. This hit record is then serialized, packaged, and sent out via the DataOut line.

For reference, the structure of the serial data reported per each LV1A signal is shown below.

<table>
<tr><td>1010</td><td colspan="2">BC<11:0></td></tr>
<tr><td>1100</td><td>EC<7:0></td><td>Flags<3:0></td></tr>
<tr><td>1110</td><td colspan="2">ChipID<11:0></td></tr>
<tr><td colspan="3">Channel Data <127:0></td></tr>
<tr><td colspan="3">CRC 16 checksum <15:0></td></tr>
</table>

**Table 1-3: Structure of DataOut Serial Packet**

The CalPulse signal is used to send a calibration pulse onto the signal lines of the VFAT. We were successfully able to verify that a pulse could be detected on each individual signal line of the VFAT ICs via this method. One can see the results in section "4.6 CalPulse" on page 65. Beyond this, the operation details are clearly described in the VFAT Operation Manual.[1]

At times it may be necessary to reset the values of the state machines. This happens automatically at the beginning every time the power is turned on to the VFAT. It is also possible to do this during normal operation by sending a ReSync pulse on the T1 line to the VFATs. This can be particularly useful between the time when the VFATs are first powered on and when actual hits are to be recorded. As described in “Readout Controller Library“, this signal is automatically sent to the VFATs when the function “v1495Reset()” is called on the Readout Controller.

Finally, rather than completely resetting all of the state machines contained in the VFAT, the BC0 signal can be sent to only reset the Bunch Crossing Number (BCN). The BCN is a 12-bit counter that increments once for every clock period. It might be useful, for example, to use this number to determine if there are missing LV1A requests in the data.

### 1.3.3 I$^2$C Communication

The physical mode of communicating with the configuration registers of the VFAT chip is performed via an I$^2$C protocol.

I$^2$C is a synchronous, full-duplex communication scheme albeit not a simultaneous full-duplex signal. The clock signal is always sent from the master microcontroller via the SCL line and the data is sent either by the master (in the event of a write or request) or by the slave (in the case of a read) via the SDA line. Both lines are sent via an open-drain configuration. This means that while the I$^2$C is being used by one device, another device can look at the I$^2$C lines to see if they are pulled low before proceeding to send data.

Each VFAT has a unique 7-bit I$^2$C address for which the first three MSBs indicate the specific VFAT and the four LSBs indicate which primary address is to either be written to or read. A description of how to set the I$^2$C address on the VFAT Breakout Board can be found in section “4.2.2 I$^2$C Address and Scan Enable Jumpers” on page 42. If one of the extended registers is to be written to or read from, one must first fill the “ExtRegPointer” primary register 0x0D with the address of the respective extended register. The data can then either be written to or read from the “ExtRegData” primary register 0x0E This allows the VFAT to contain an extra 135 registers.

There are many references on utilizing the I$^2$C interface. For more information on this common digital communication scheme, the reader is encourage to review the content of other references on the subject.

### 1.3.4 Gumstix Microcontroller

In order to communicate to the VFAT registers via I$^2$C and to know their respective contents for our experiments, it was determined that this could most efficiently be implemented by using a Gumstix Microcontroller. The name for Gumstix aptly describes these powerful microcontrollers as their size roughly resembles that of a stick of gum. These Computer-on-Module (COM) microcontrollers include many extensible modules including an I$^2$C communication board and an ethernet board.

The actual setup and implementation of these microcontrollers was performed by Brian Oborn. The Gumstix microcontroller actually uses a form of C-Shell from Linux and is supported by the OpenEmbedded Project. Thus, we were able to use the I$^2$C and CGI capabilities of this distribution of Linux to not only read and write to the I$^2$C registers on the VFATs but also to display the contents of these registers via a website hosted on the Gumstix controller.

A picture of one of our Gumstix computers is shown below in Figure 1-5.

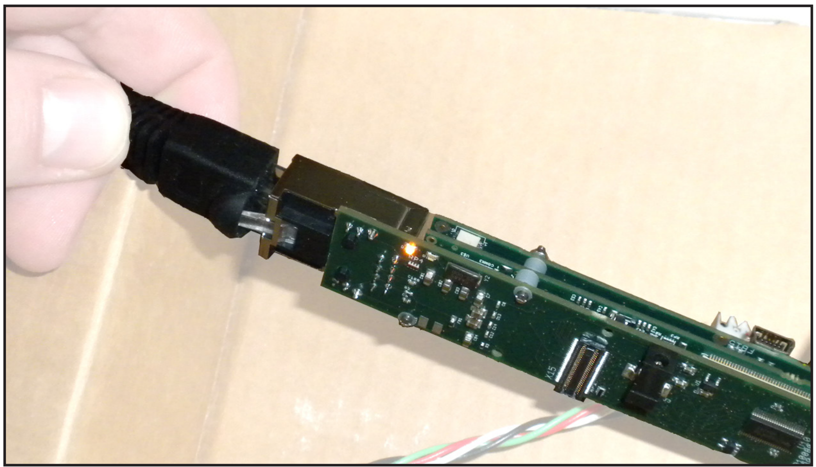


**Figure 1-5: Gumstix microcontroller with I$^2$C module and ethernet module attached**

## 1.4 V1495

The V1495 module contains two Field-Programmable Gate Arrays (FPGAs). The first FPGA is programmable by the user and is used as the main interface to the external I/Os on the faceplate of the V1495. The second FPGA controls the interfacing of the User FPGA and the VME backplane. Both FPGAs are members of the Cyclone device family.[5] The complete operation and underpinnings of the V1495 by itself are worthy of a separate thesis; the Cyclone Device Handbook, Volume 1, contains 385 pages. Of this, an understanding of perhaps 10 percent is sufficient to comprehend, operate, and effectively alter the properties of the Cyclone FPGA for the Region 1 detector.

Programming and utilizing the V1495 is one of the main portions of this thesis and therefore has an entire chapter dedicated to the user-defined code that was implemented to query the VFAT chips and store that information until it is requested by the Readout Controller.

### 1.4.1 Readout Controller (ROC)

Part of using the VME protocol is that one rarely directly accesses one of the VME modules on the VME plane; this project is no exception. One actually communicates to the individual modules, i.e. the V1495, via a Single-board Computer (SBC) known as the MVME6100. For our project we called this the Readout Controller.[6],[7]

The MVME6100 is a very powerful SBC with a high-performance MPC7457 PowerPC processor capable of dictating instructions to all of the modules contained within a single VME crate via the VME backplane at 320MB/s. It also has the capability of communicating to other computers outside of the VME crate via its ethernet ports. Programs for the V1495 are actually cross-compiled on a separate Linux-based host computer and then downloaded to the MVME6100 via this ethernet connection. The target OS that we utilize on the MVME6100 is actually an embedded version of VxWorks which is also used in many scientific as well as commercial application, i.e. Linksys Routers.

As with many of the modules discussed within the introduction, the reader is referred to other resources for specifics on how to utilize this module.

### 1.4.2 CODA Readout

Data from our cosmic particle experiments as well as HRRL experiments were taken and processed by a software package known as CODA. This software package was developed for use at Jefferson Lab. It contains many libraries that are useful for capturing and storing data that is recorded using the VME system.

For more information, visit the CODA resource website.[15]

Figure 1-6 below shows the typical setup of our cosmic particle detection runs. On the left side we can see the Region 1 detector laying flat on a stand. In the lower right side of the picture we can also see the VME crate containing both the V1495 as well as the Readout Controller.

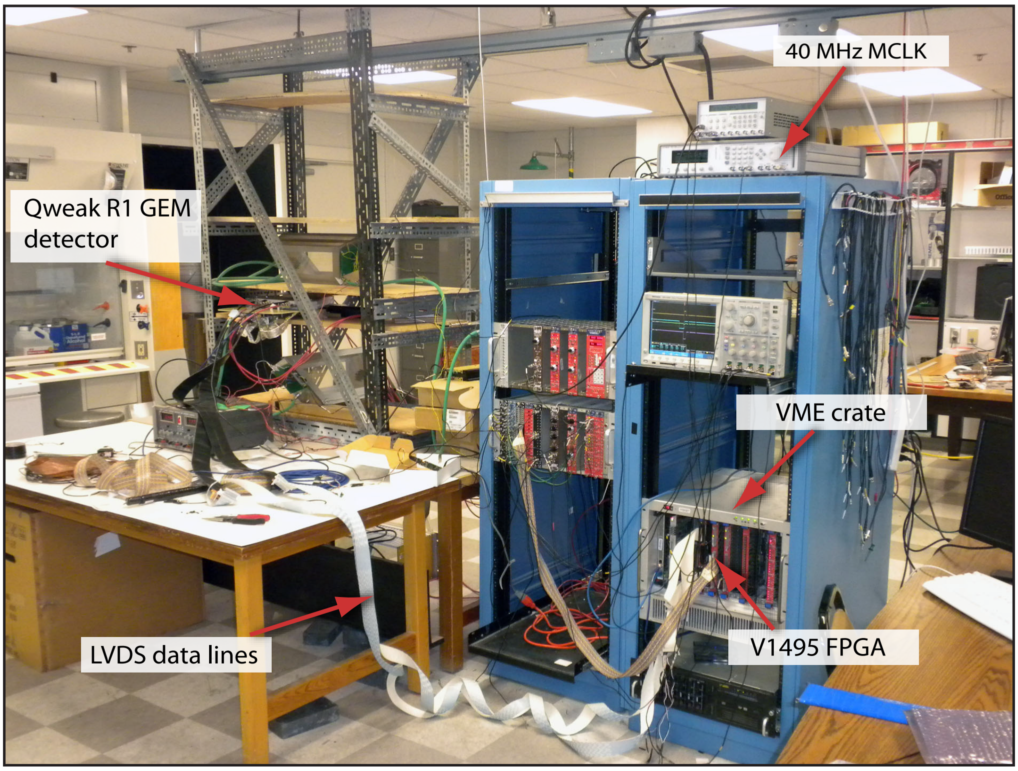


**Figure 1-6: A typical setup of the Region 1 detector including VME crate**

# Chapter 2

# V1495 Module by CAEN

## 2.1 Overview of V1495 Module

From the V1495 User Manual[4]:

> The Mod. V1495 is a VME 6U board, 1U wide, suitable for various digital Gate/Trigger/Translate/Buffer/Test applications, which can be directly customised by the User, and whose management is handled by two FPGA's: FPGA "Bridge", which is used for the VME interface and for the connection between the VME interface and the 2nd FPGA (FPGA "User") through a proprietary local bus. FPGA "Bridge" manages also the programming via VME of the FPGA "User". FPGA "User", which manages the front panel I/O channels. FPGA "User" is provided with a basic firmware which allows to perform coincidence matrix, I/O register and asynchronous timers functions.
>
> FPGA "User" can be also free reprogrammed by the user with own custom logic function (see § 5.1). It is connected as slave to the FPGA "Bridge" via CAEN Local Bus, whose protocol shall be used in order to communicate with the FPGA "Bridge" and thus with the VME bus.
>
> The I/O channel digital interface is composed by four

sections (A, B, C, G) placed on the motherboard (see § 1.2). The channel interface can be expanded in the D, E, F sections by using up to 3 mezzanine boards (see § 2.6 and § 2.7), which can be added, choosing between the four types developed in order to cover the I/O functions and the ECL, PECL, LVDS, NIM, TTL electrical standard (see § 1.2). The maximum number of channels can be expanded up to 194.

The FPGA "User" can be programmed "on the fly" directly via VME, without external hardware tools, without disconnecting the board from the set up, without resetting it or turning the crate off, allowing quick debug operations by the developer with his own firmware. A flash memory on the board can store the different programming file, which can be loaded to the FPGA "User" at any moment.

Four independent digital programmable asynchronous timers are available for Gate/Trigger applications. It is possible to chain them for generating complex Gate/Trigger pulse.

## 2.2 MCLK/LCLK/PLLCLK/PLLCLK_90

There are two digital clocks that synchronize all of the digital state machines in the V1495 as well as the VFATs for the GEMReadout project. Only the local clock is used in the v1495usr_firmare project.

The local clock, LCLK, for the V1495 has a fixed frequency of 40Mhz and is the clock to which all other internal digital states other than those governed by the MCLK are synchronized. This includes the important read/write/reset operations involving the VME bus. This clock is necessary for all V1495 projects.

The MCLK is a variable clock for the GEMReadout project that is injected by an external pulse generator (using the input labeled "G0") that controls the read/write cycles of the VFAT as well as the data that is received from the VFATs by the V1495. Usually when referring to the signal as seen by the VFATs, it is referred to as MCLK per the VFAT2 literature. Internal to the V1495 all data sent and received is synchronized to the rising edge of MCLK, but here it is referred to as PLLCLK. Depending on the frequency of MCLK this may be controlled by a PLL. See the section "PLLBlock.vhd" for more information. PLLCLK_90 is the improperly named, inverted version of PLLCLK. This is the signal sent to the VFATs since their read/write cycle is 180 degrees out of phase with the V1495.

Having two different clocks internal to the V1495 has posed some particularly difficult problems. Care has been taken in this project when one signal is changed in sync with one of the clocks and then analyzed in sync with the other clock. An example of this is the CalPulse and Reset functionality in the V1495. When dealing with HDL one has to be careful to include conditional statements on two separate lines when the changes that occur are due to the two different clocks; otherwise the HDL translator takes the liberty of changing one of the signals on the rising edge of the incorrect clock. This is not particularly easy to diagnose as it does not behave as one might expect from an intelligently designed compiler.

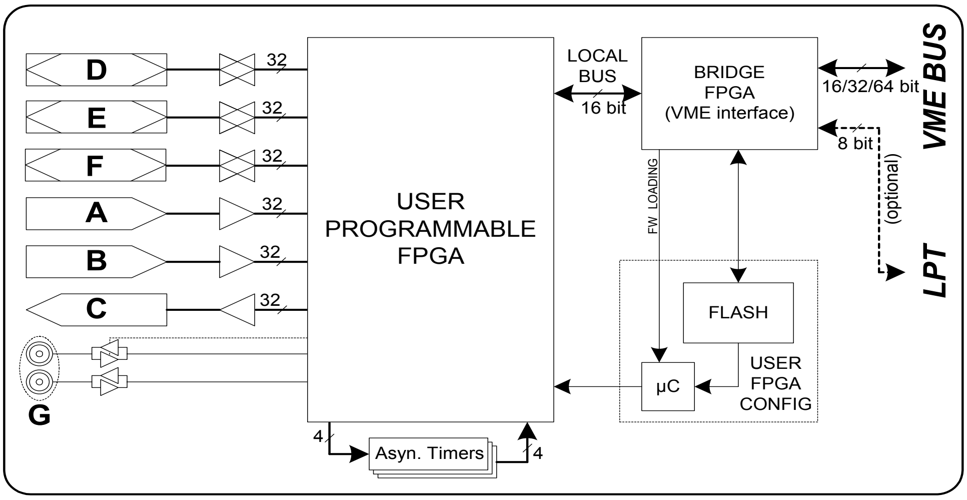


**Figure 2-1: Mod. V1495 Block Diagram**

## 2.3 Inputs and Outputs

As one can see from the block diagram in Figure 2-1, and as also found in the V1495 User Manual, the v1495 has two LVDS, 32-channel input ports (e.g. A and B), one LVDS, 32-channel output port ( C ), and two NIM/TTL I/Os (G0, G1) that come standard. There are three other mezzanine expansion ports that can be purchased with addition 32-channel LVDS/ECL/PECL, 8-channel NIM/TTL, or even 8-channel 16-bit resolution analogue ports. For more information on this refer to the CAEN website. A diagram of the physical layout of these ports from the perspective of the front panel is also included in Figure 2-2 on page 16.

For the purposes of this project, ports A, C, G0, and G1 have been utilized with port G configured as TTL logic inputs and 50-Ohm termination. The termination switches are physically located on the board near the G port, and it is not specifically mentioned in the V1495 User Manual how to configure this. Furthermore, the inner workings of the V1495 (i.e. board layout) are not readily available even if the user wishes to have these for configuration purposes. The easiest way to determine how to program and utilize the V1495 is to familiarize oneself with the concepts of an FPGA and how the Cyclone FPGA for this board is connected to the external world via the V1495_USR_FIRMWARE that is provided for free. A basic overview of this firmware is provided in section "2.7 V1495_USR_firmware" on page 20.

All signals received by the V1495 are determined by the rising edge of their respective clocks. There are no other dexterous schemes implemented for noise rejection or signal degradation.

There are also two LEDs on the front panel of the V1495 for the user interface. The DTACK LED blinks green whenever a VME read/write operation is performed on the board. This can be used to determine whether or not a

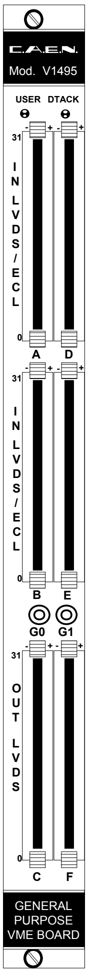


**Figure 2-2: Model V1495 front panel (with A395A/B/C piggy back boards)**

firmware update is working. It is not utilized during normal operation since most read/write operations to the V1495 happen faster than the LED can blink let alone how fast the human eye can perceive a pulse of this light. The USER LED has three colors: green/orange/red (really this is just green and red with the orange being the combination of green and red). The green part of this LED blinks in sync with the system clock, LCLK, at a rate of 0.60Hz and a 50% duty cycle. This comes from bit 25 of the signal "HEART_BEAT_CNT" which is incremented at the rising edge of LCLK. The red LED is on whenever the system reset is active. This happens when either the asynchronous system reset, nLBRES, is active low or the PLL is not locked. For the version of the GEMReadout firmware where there is no PLL (e.g. at MCLK frequencies lower than 15Mhz), this LED can only be active when the asynchronous reset is low.

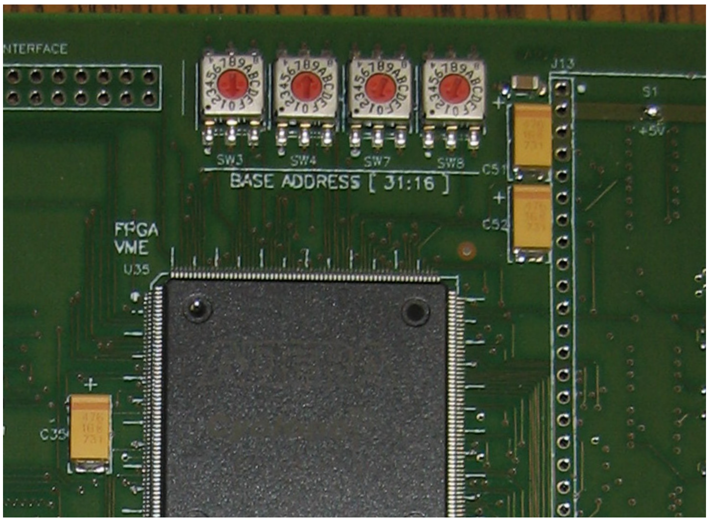


**Figure 2-3: HEX Address Switches on V1495**

## 2.4 Addressing the V1495

It is important to note that the VME bus address is different than the memory-mapped logical address for the board as referred to by the PowerPC controller. For most of the functions written for the Read Out Controller (ROC), these will reference the PowerPC logical address rather than the VME bus address. The one major exception for this is the v1495Init() function which uses the VME bus address to check for the presence of the V1495 and then returns the logical memory address. More on this function is described in the chapter "Readout Controller Library" on page 33.

For this project we have set the address lines of the V1495 to the hex values "0811" respectively. As can be seen in the silk screen print on the board in

Figure 2-3, this is the base address for the 32 most significant bits in the VME bus address (BASE ADDRESS [31:16]). This, of course, signifies that the V1495 uses a 32-bit addressing scheme. According to the V1495 User Manual 24-bit addressing could also be used, but we have not implemented this.

Determining the exact relationship between the VME bus address and the ROC memory address are related by a simple offset that is ROC model dependent. For the MVME6100 this offset is 0x7800_0000. With the VME bus address of 0x0811_0000 this gives a memory address of 0x8011_0000.

All register addresses in the V1495 are located using this offset plus their respective addresses. These registers can essentially be divided into two different classes: Register addresses 0x0000 – 0x7FFC are the USER FPGA Access registers and register addresses 0x8000 – 0x801FE contain the V1495 CAEN interface registers.

The USER FPGA registers are determined in the v1495usr_pkg.vhd module. Table 2-4 shows these registers with their "struct" equivalents in the ROC software. These registers contain most of the pertinent information being

| USER FPGA Registers | V1495ReadoutCtrl struct | Register Numbers | Description |
|---|---|---|---|
| A_BOARDIDS | MezzanineIDs | 0x0000 - 0x0001 | Mezzanine IDs for D, E, and F |
| A_REVISION | Revision | 0x0002 - 0x0003 | Firmware Revision (hardcoded in firmware) |
| A_RESET | Reset | 0x0004 - 0x0005 | Used to Reset V1495 |
|  | Reserved1[5] | 0x0006 - 0x000F |  |
| A_GEM_TX_START | TxStart | 0x0010 - 0x0011 | Sends GEM_TX_START to TxChannel |
| A_GEM_SOFT_TRIG | SoftTrigger | 0x0012 - 0x0013 | Sends SOFT_TRIGGER signal |
| A_GEM_TRIG_WORD | TriggerWord | 0x0014 - 0x0015 | Contains SOFT and HARD trigger words |
| A_GEM_TX_WORD_(0 - B) | TxWord[12] | 0x0016 - 0x002D | By default, this is V1495 VME Input register |
|  | Reserved2[1] | 0x002E - 0x002F |  |
| A_GEMA(0-5)_FIFOSIZE | FIFOLength[12] | 0x0030 - 0x0047 | Contains total number of words currently in dataFIFO as well as the sizeFIFO |
| A_GEMA(0-5)_EVENTSIZE<br>A_GEMB(0-5)_EVENTSIZE | EventSize[12] | 0x0048 - 0x005F | Contains the value for the number of words stored in dataFIFO from most resent event |
|  | Reserved3[16] | 0x0060 - 0x007F |  |
| A_GEMA(0-5)_EVENTS_SENT_H<br>A_GEMB(0-5)_EVENTS_SENT_H | EventsSentH[12] | 0x0080 - 0x0097 | Contains bits 32 - 16 of the value of the number of events sent |
| A_GEMA(0-5)_EVENTS_SENT_L<br>A_GEMB(0-5)_EVENTS_SENT_L | EventsSentL[12] | 0x00A0 - 0x00B6 | Contains bits 15 - 0 of the value of the number of events sent |
| A_GEMA(0-5)_EVENTDATA<br>A_GEMB(0-5)_EVENTDATA | EventData[12][128] | 0x4000 - 0x4BFF | Contains the actual DataOut data as captured in 16-bit segments from the VFATs |

| V1495ReadoutStatus struct | Register Numbers |
|---|---|
| data[16384] | 0x0000 - 0x7FFF |
| control | 0x8000 |
| status | 0x8002 |
| intLevel | 0x8004 |
| intVector | 0x8006 |
| geoAddr | 0x8008 |
| moduleReset | 0x800A |
| firmwareRev | 0x800C |
| selflashVME | 0x800E |
| flashVME | 0x8010 |
| selflashUSER | 0x8012 |
| flashUSER | 0x8014 |
| configUSER | 0x8016 |
| scratch16 | 0x8018 |
| res1[3] | 0x801A |
| scratch32 | 0x8020 |
| res2[110] | 0x8024 |
| configROM[127] | 0x8100 |

**Table 2-4: USER FPGA Registers and "struct" Equivalents**

communicated between the V1495 and the ROC: for example, A_GEMA0_EVENTDATA, address 0x4000, contains the information from the top of the DataFIFO for the first VFAT detector when requested.

The V1495 CAEN interface registers contain information on the V1495 board itself as well as controlling how to reflash and reset the VME and USER FPGAs. For further details the reader is referred to the V1495 User Manual pg. 17-21.[4]

## 2.5 Brief Overview of FPGAs

FPGAs or "Field Programmable Gate Arrays" are electronic ICs -- typically digital -- that have the ability to be reprogrammed for different purposes. Quite often these circuits are used in the initial design phases of an IC when a hardwired circuit would be much more costly to produce for each revision of the product. Once the IC has been proven to work correctly, the design can then become hardwired onto an IC that can be mass produced.

In our experiment, we also use FPGAs for their ability to be reprogrammed, but we do not intend on mass-producing a hardwired version of our IC. We simply use its reprogrammable abilities to perform our highly specialized functionality. In this instance, we control the LVDS T1 signal going to the VFATs, extract the LVDS packets from the VFAT, store the data, and then pass it on to the ROC.

When referring to electronics, HDL stands for "Hardware Description Language". It is the term used for the code that describes hardware layout. For this reason it is also commonly referred to as "firmware" as it is neither completely software nor hardware. There are many different HDL languages, but the only two utilized for this project are VHDL (Very High Speed HDL) and Verilog. Most of the modules that the user will even want to change are those in VHDL; it is therefore recommended that if one wants to alter the user firmware for the V1495 that one spends most of one's time learning VHDL. There is only one module in Verilog, V1495usr_hal.vqm, the purpose of which is described in section "2.8.2 Verilog Modules" on page 31.

HDL's utility lies in its ability to describe hardware using a behavioral, a dataflow, or a structural model to describe the circuit one wants to build.

A behavioral model is one in which the user uses more traditional software flow-control statements to control the logical properties of the circuit; at this level in VHDL, these include "Process" statements, "If" statements, "Case" statements, Loop statements (e.g. "Do", ”While"), "Next"/”Exit" statements, as well as “NULL” statements. It is at this level that the classic Mealy and Moore state machines are implemented.

Dataflow modeling is similar to behavioral modeling except instead of using flow-control similar to other traditional programs, one uses concurrent assignment operators to control the logical flow of data with operators such as “OR”, “AND”, “XOR”, etc. It has some conditional control like behavioral modeling, but it remains more low-level. It is conceivable that we could have used this form of modeling for the register-access modules, but these modules are in behavioral form of modeling.

Structural modeling is virtually a literal translation of traditional digital logic symbols (e.g. AND-gates, OR-gates, etc) into a software format. In structural modeling one uses previously-defined components to create the necessary logical operations by connecting their I/O’s properly. This form of modeling is very similar to Object Oriented Programming; the components can be thought of as the classes and the logical flow can be thought of as using the specific instantiations of the classes/components. For this thesis, we do not utilize this form of modeling either.

## 2.6 Quartus II

Instructions for Downloading and running Quartus II

For this project we used the free Quartus II Web Edition Software. At the time of publication for this thesis, it is available at the following web address: http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html

Fortunately, for the purposes of this project, we only need to use it for its ability to handle and compile modules for the Cyclone family of FPGAs. It is, however, also useful for viewing the “Help” files on a number of modules in this project as well as viewing the pinouts for the FPGA (Assignments->Pins menu in Quartus). Although this level of detail is not necessary for this project, it does help one understand which signals are internal to the FPGA and therefore capable of being altered by the user firmware on the v1495, and which signals are external to the FPGA and not capable of being altered by the user firmware.

## 2.7 V1495_USR_firmware

The V1495_USR_firmware is provided by CAEN for the purpose of demonstrating how to use the v1495 module. This firmware is provided as well on the CD included with this thesis but can also be found on CAEN's website. A number of functions are provided to introduce the V1495 user to the utilities of the V1495.

## 2.8 GEMReadout Firmware Modules

### 2.8.1 VHDL Modules

#### v1495usr.vhd

This is the lowest hierarchical level for V1495 user firmware. It includes all of the I/Os as can be seen on the Pinout for the project. The Pinout for any project in Quartus II can be found in the Assignments->Pins menu.

#### tristate_if_rtl.vhd

This module simply contains the logic necessary to control the direction of the FPGA data bits found in v1495usr.vhd and the logic to control the direction of the bidirectional I/Os associated with the VME data bus. Although there is probably little reason to change this section of code a brief description is provided below.

By default the FPGA data bits are set at High-Z (as can be seen by following the signal into v1495usr_hal.vqm). Since these bits are not used in this project this is of little consequence.

More importantly, the direction of the "LAD" bits (Local Data Bus) is controlled by "LAD_OE". If "LAD_OE" is high then the "LAD" bits will be configured as outputs. As reason would predict, if "LAD_OE" is low then the "LAD" bits will be inputs. Ultimately the value of "LAD_OE" is controlled by convoluted code found in v1495usr_hal.vqm. It should be noted that in GEMReadout.vhd the signals that determines whether a read or write operation is to be performed are "REG_RDEN" and "REG_WREN" respectively. Again, these signals are generated in the convoluted logic found in v1495usr_hal.vqm.

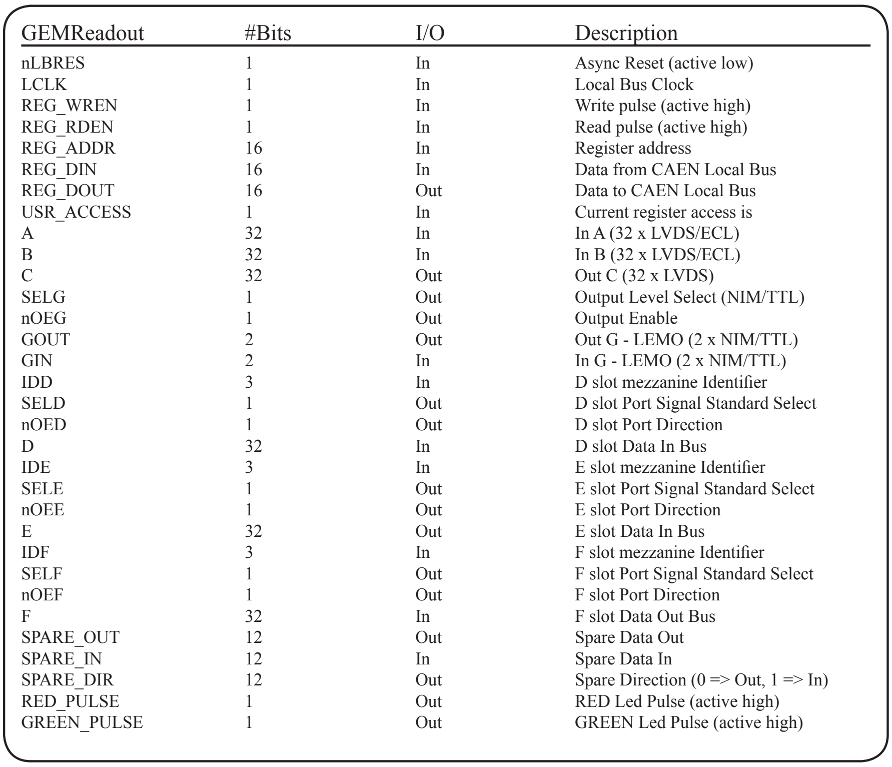
## spare_if_rtl.vhd

This module simply contains the logic necessary to control the direction of the spare bidirectional I/Os included with the Cyclone FPGA. These are not utilized with the current version of the GEMReadout program but could potentially be adapted for future utility by CAEN for the V1495. It would be difficult to adapt their usage at this level of design because we are not privy to the inner schematics of the V1495 board to see where these signals go.

## GEMReadout.vhd

This module handles all of the receiving/transmitting functionality to/from the VFAT ICs, the FIFO storage, the PLL timing (if used), control of the LEDs on the V1495 and the ROC interface capabilities via the VME bus.

As previously stated, it is not the intent of this thesis to clearly describe how VHDL works which is needed if one wants to be able to read the code and precisely determine how this module works. A UML has been provided to help elucidate the operation of this module. See section “2.3 Inputs and Outputs” on page 15 for a description of the I/Os for the V1495 including the LEDs.

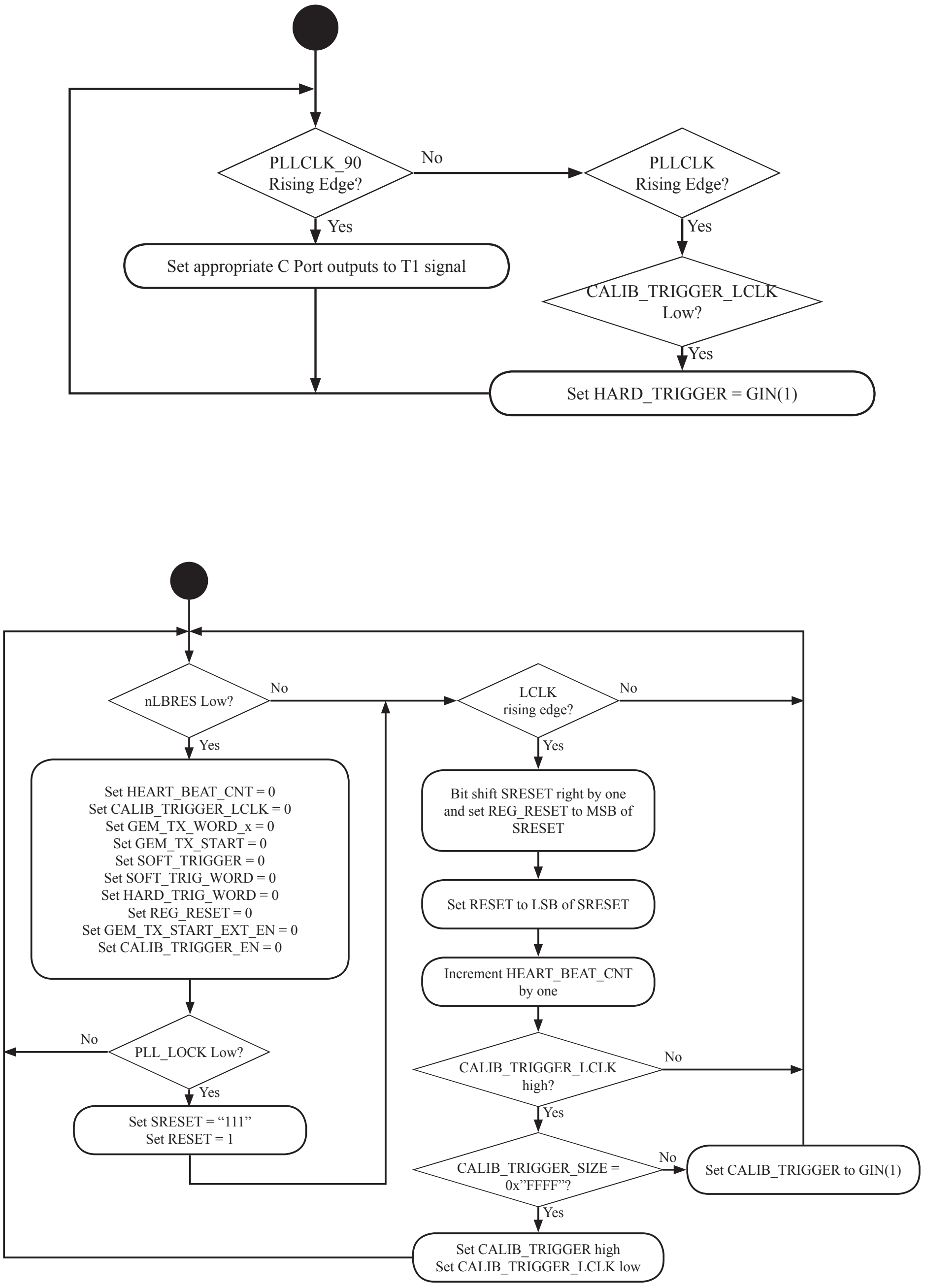


| GEMReadout | #Bits | I/O | Description |
|---|---|---|---|
| nLBRES | 1 | In | Async Reset (active low) |
| LCLK | 1 | In | Local Bus Clock |
| REG_WREN | 1 | In | Write pulse (active high) |
| REG_RDEN | 1 | In | Read pulse (active high) |
| REG_ADDR | 16 | In | Register address |
| REG_DIN | 16 | In | Data from CAEN Local Bus |
| REG_DOUT | 16 | Out | Data to CAEN Local Bus |
| USR_ACCESS | 1 | In | Current register access is |
| A | 32 | In | In A (32 x LVDS/ECL) |
| B | 32 | In | In B (32 x LVDS/ECL) |
| C | 32 | Out | Out C (32 x LVDS) |
| SELG | 1 | Out | Output Level Select (NIM/TTL) |
| nOEG | 1 | Out | Output Enable |
| GOUT | 2 | Out | Out G - LEMO (2 x NIM/TTL) |
| GIN | 2 | In | In G - LEMO (2 x NIM/TTL) |
| IDD | 3 | In | D slot mezzanine Identifier |
| SELD | 1 | Out | D slot Port Signal Standard Select |
| nOED | 1 | Out | D slot Port Direction |
| D | 32 | In | D slot Data In Bus |
| IDE | 3 | In | E slot mezzanine Identifier |
| SELE | 1 | Out | E slot Port Signal Standard Select |
| nOEE | 1 | Out | E slot Port Direction |
| E | 32 | Out | E slot Data In Bus |
| IDF | 3 | In | F slot mezzanine Identifier |
| SELF | 1 | Out | F slot Port Signal Standard Select |
| nOEF | 1 | Out | F slot Port Direction |
| F | 32 | In | F slot Data Out Bus |
| SPARE_OUT | 12 | Out | Spare Data Out |
| SPARE_IN | 12 | In | Spare Data In |
| SPARE_DIR | 12 | Out | Spare Direction (0 => Out, 1 => In) |
| RED_PULSE | 1 | Out | RED Led Pulse (active high) |
| GREEN_PULSE | 1 | Out | GREEN Led Pulse (active high) |

**UML Diagram 2-1: GEMReadout.vhd**

## GEMTrigger.vhd

The GEMTrigger module handles the conversion of a single-pulse trigger to a 3-bit (or 8-bit in the case of the Calibration Pulse signal) trigger word which is sent to the T1 line of the VFATs which is in sync with the MCLK. The single-pulse trigger can be generated as a hard trigger from the G0 port, as a soft trigger



**UML Diagram 2-2: GEMReadout.vhd**

internally, as a reset trigger internally, or even as a calibration trigger which can be generated either internally or externally depending on the value written to the A_GEM_CALIB_START register. Each of these triggers is more thoroughly described below.

A hard trigger from the G0 input port generates whatever signal is located in HARD_TRIG_WORD. By default this is the LV1A signal, “0b100”, as this is probably used the most often. From the VFAT2 Manual we know that this is a request to the VFAT for a data packet of all of the hits on the lines. For a more thorough explanation of this process see the section in the VFAT2 Manual V2.0 on page 4 covering an LV1A request. HARD_TRIG_WORD is programmable by writing to bits 5-3 in register A_GEM_TRIG_WORD (offset + 0x0014). See “Table 2-4” on page 18 for a list of the available registers on the v1495.

A soft trigger is similar to a hard trigger except that it is requested by a write operation to the A_GEM_SOFT_TRIG register. The SOFT_TRIGGER_WORD is located in bits 2-0 in register A_GEM_TRIG_WORD (offset + 0x0014). The default value for this word is “0b000”.

The calibration trigger is the most complicated of all the triggers. Its main purpose is for quickly producing the S-curves necessary to adjust the threshold values for the channels on the VFATs high enough to reject systemic noise but still low enough to detect sensitive, real signals. A single sequence of the calibration trigger is comprised of a CalPulse signal, “0b110”, followed by a single blank clock cycle which is then followed by an LV1A signal, “0b100” to the T1 line.

There are two different ways of using the calibration trigger pulse. A write to the A_GEM_CALIB_START (offset + 0x000E) register with words 0x0000 – 0xFFFE (anything other than 0xFFFF) will cause a single burst of the calibration pulse sequence immediately following the write request. Writing 0xFFFF to this register causes the calibration pulse sequence to only pulse when a single external trigger pulse is applied to the G0 input. This can be done indefinitely and is changed by writing a value other than 0xFFFF to the aforementioned register. Originally it was intended to have the number being written to this register control how many times the calibration pulse sequence was executed, but this functionality awaits future development.

### v1495usr_pkg.vhd

This module simply contains the offset address for all of the user registers of the v1495. Table 2-4 listed in “Addressing the V1495” gives a detailed description of the names and addresses of these registers as listed in this module as well as their equivalent names in v1495Lib.o data structures.

## PLLBlock.vhd

A PLL, Phase Lock Loop, is a timing stabilization mechanism. It is available to the users of the Cyclone for clock frequencies between 15Mhz and 1Ghz. At these extreme frequencies clock phase synchronization can become problematic for the internal hardware of the FPGA. By using at least one of the two available PLLs on the Cyclone FPGA, one can mitigate the clock phase synchronization error and thus achieve a much higher running frequency. For frequencies below 15Mhz the PLLBlock module is completely bypassed in the firmware. Usually these versions of the firmware include the suffix "NoPLL" in their names to indicate the difference.

For the GEMReadout user firmware this module takes care to replicate the original MCLK signal coming in on G0 and outputting it to the PLLCLK signal (when referring to the VFAT, this clock is called MCLK). It also creates a clock signal that is an exact inverse of this signal. This signal has the misnomer PLLCLK_90. Since the VFATs change their output bits on a rising edge of MCLK and evaluate incoming bits on the falling edge of MCLK, the inputs and outputs to the v1495 are evaluated and changed on the rising edge of a clock 180 degrees out of phase with this (i.e. PLLCLK_90).

The main portion of this module uses the "altpll" component from the Quartus II library. For a complete description of all of the attributes of this module see the help files in Quartus II on "altpll". There are, however, a number of attributes that the user may need to change in order to run MCLK for the v1495 at a specific frequency between 15Mhz and 1Ghz. A listing and a brief description of each of these is provided below.

inclk0_input_frequency: This is the value given for the frequency of the primary clock. The value is the period given in pico seconds and is not enclosed in quotations. This is not listed in the help-file literature. Example: A value of 25000 would give a primary running frequency of 40Mhz since 1/25000*10^-12 = 40Mhz

clk1_phase_shift: This is the value given for the amount of phase shift on the secondary clock output relative to the primary clock output. This, too, is measured in picoseconds, but this attribute's value does require quotations. Example: For the previous examples of 40Mhz, if the user requires a phase shift of 180 degrees the value of "12500" must be entered (i.e. half of the frequency period).

## GEMRxEventDataFIFO.vhd/ GEMRxEventSizeFIFO.vhd

These modules both contain the specific layout of the DataFIFO and the SizeFIFO and both are extremely similar in layout.

The dcfifo, of which the DataFIFO and SizeFIFO are comprised, is a true dual-port FIFO (one clock for reading and one for writing as long as it is not the same node) using M4K technology from the Cyclone family of devices. For reference to this device structure see the Quartus II help files on "dcfifo Megafunction" where a more detailed description of the parameters for specific parameters can be found.

The dcfifo megafunction works precisely as one would expect a FIFO to work. When a write request is received by the FIFO, the word currently on "data[]" will be written into the FIFO (as long as the FIFO is not full). For the DataFIFO this is the current word taken from the DataOut stream. If the FIFO is full, the FIFO will ignore the write request. Therefore, care must be taken to ensure that the FIFO does not reach capacity or else the V1495 will start dropping packets received from the VFATs. As long as the FIFO is not empty, when a read request is received the FIFO will place the oldest word in the FIFO on the "q[]" signal. In the case of the DataFIFO this is "EVENT_DATA.". In the case of the SizeFIFO this is "EVENT_SIZE." A detailed description of how the DataFIFO and SizeFIFO are used for our project can be found in the section "GEMRxChannel.vhd" on page 28.

For our project some of the more important parameters for the DataFIFO are as follows:

- Number of words the DataFIFO can store = 1024
- Bit width of the data being read and written to the FIFO = 16
  - N.B. This matches the width of the data bus on the VME.
- Bit width of the number of words that are currently in the FIFO = 10
  - N.B. This matches the number of bits needed to report the maximum number of words that can be stored in the FIFO.

The parameters for the SizeFIFO are as follows:

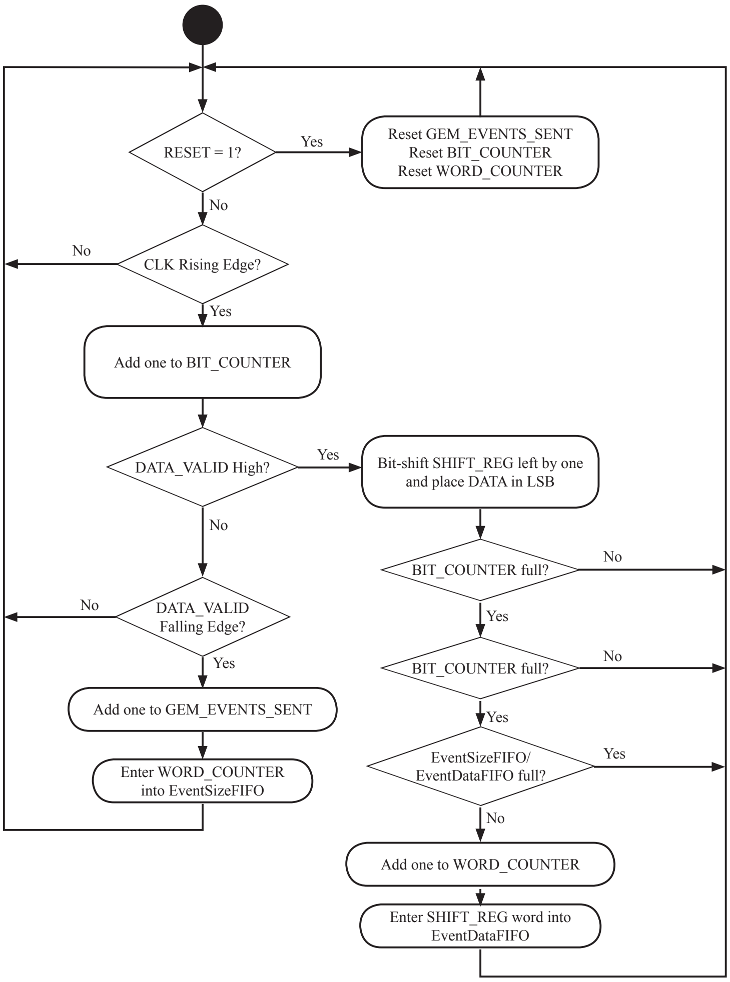
- Number of words the DataFIFO can store = 64
- Bit width of the data being read and written to the FIFO = 4

  - N.B. This matches the width of the data bus on the VME.

- Bit width of the number of words that are currently in the FIFO = 6
  - N.B. This matches the number of bits needed to report the maximum number of words that can be stored in the FIFO.

| GEMRxChannel | | GEMReadout | #Bits | I/O |
|---|---|---|---|---|
| LCLK | => | LCLK | 1 | In |
| RD_EVENT_SIZE | => | GEMx_RD_EVENT_SIZE | 1 | In |
| EVENT_SIZE | => | GEMx_EVENT_SIZE | 4 | Out |
| EVENT_COUNT | => | GEMx_EVENT_COUNT | 6 | Out |
| RD_EVENT_DATA | => | GEMx_RD_EVENT_DATA | 1 | In |
| EVENT_DATA | => | GEMx_EVENT_DATA | 16 | Out |
| EVENT_DATA_SIZE | => | GEMx_EVENT_DATA_SIZE | 10 | Out |
| GEM_EVENTS_SENT | => | GEMx_EVENTS_SENT | 32 | Out |
| CLK | => | PLLCLK | 1 | In |
| RESET | => | RESET | 1 | In |
| DATA | => | GEMx_DATA | 1 | In |
| DATA_VALID | => | GEMx_DATA_VALID | 1 | In |

where ‘x’ is for the respective VFAT



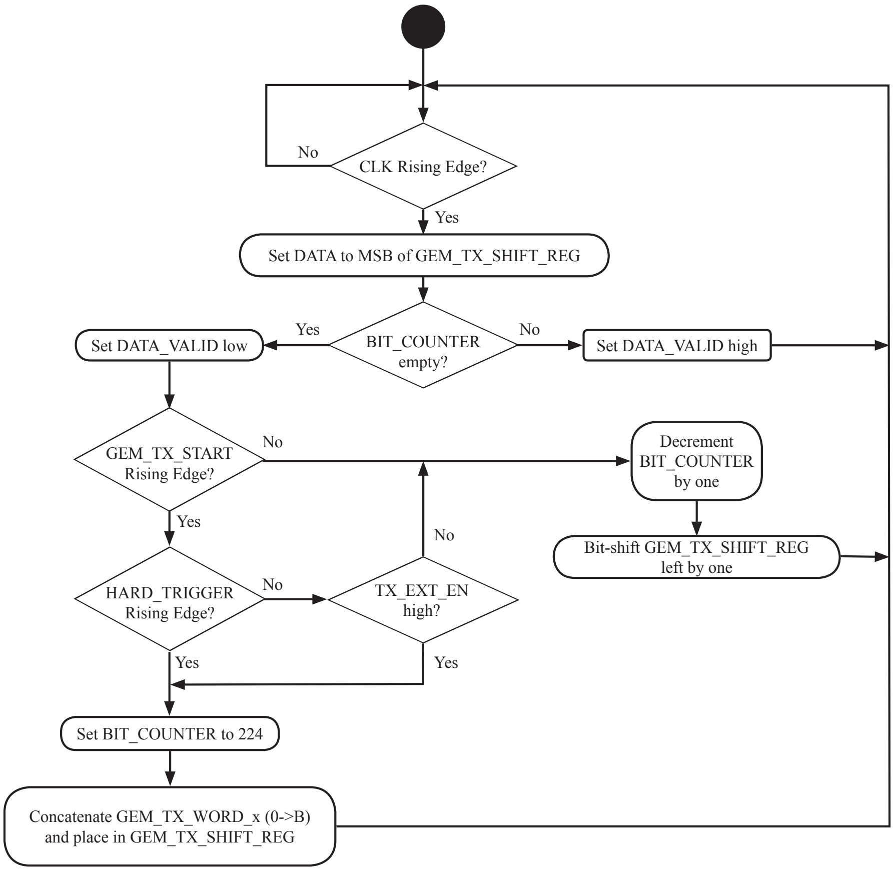
**UML Diagram 2-3:GEMRxChannel.vhd**

## GEMRxChannel.vhd

The GEMRxChannel.vhd module governs the behavior of the data received from the VFATs on their respective DataOut and DataValid channels. This is shown below in “UML Diagram 2-3” on page 27 for this module. For a more detailed description of the parameters settings for these FIFOs see section “GEMRxEventDataFIFO.vhd/ GEMRxEventSizeFIFO.vhd.”

| GEMTxChannel | | GEMReadout | #Bits | I/O |
|---|---|---|---|---|
| LCLK | => | LCLK | 1 | In |
| GEM_TX_WORD_x | => | GEM_TX_WORD_x | 16 | In |
| GEM_TX_START | => | GEM_TX_START | 1 | In |
| CLK | => | PLLCLK | 1 | In |
| TX_EXT_EN | => | GEM_TX_START_EXT_EN | 1 | In |
| HARD_TRIGGER | => | HARD_TRIGGER | 1 | In |
| DATA | => | GEM_TX_DATA | 1 | Out |
| DATA_VALID | => | GEM_TX_DATA_VALID | 1 | Out |

where ‘x’ is for the respective TX_WORD (0-B)


No
CLK Rising Edge?
Yes
Set DATA to MSB of GEM_TX_SHIFT_REG
Set DATA_VALID low
Yes
BIT_COUNTER empty?
No
Set DATA_VALID high
GEM_TX_START Rising Edge?
No
Yes
Decrement BIT_COUNTER by one
No
Bit-shift GEM_TX_SHIFT_REG left by one
HARD_TRIGGER Rising Edge?
No
TX_EXT_EN high?
Yes
Yes
Set BIT_COUNTER to 224
Concatenate GEM_TX_WORD_x (0->B) and place in GEM_TX_SHIFT_REG

**UML Diagram 2-4:GEMTxChannel.vhd**

## GEMTxChannel.vhd

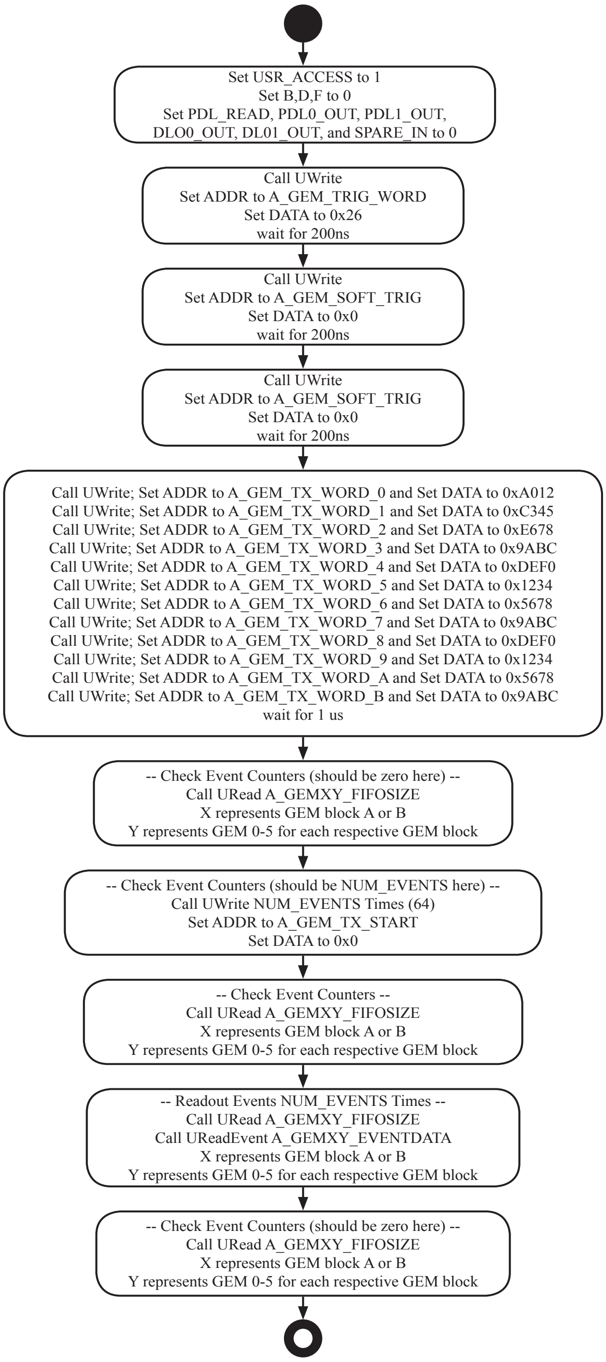
This section was not utilized during the final stages of this project. It was used (along with the GEMReadout_tb) during the initial stages to provide debugging methods for the basic functionality of the I/Os on the V1495. It structure is simply as shown in “UML Diagram 2-4” on page 28.

## GEMReadout_tb.vhd

Although it was not used in the final stages of this project (that which was conducted at ISU), this module provides a useful test bench with which to test the basic I/O functionality of the V1495. This is particularly useful when one

| GEMReadout_tb | #Bits | I/O | Description |
|---|---|---|---|
| nLBRES | 1 | In | Async Reset (active low) |
| LCLK | 1 | In | Local Bus Clock |
| REG_WREN | 1 | In | Write pulse (active high) |
| REG_RDEN | 1 | In | Read pulse (active high) |
| REG_ADDR | 16 | In | Register address |
| REG_DIN | 16 | In | Data from CAEN Local Bus |
| REG_DOUT | 16 | Out | Data to CAEN Local Bus |
| USR_ACCESS | 1 | In | Current register access is |
| A | 32 | In | In A (32 x LVDS/ECL) |
| B | 32 | In | In B (32 x LVDS/ECL) |
| C | 32 | Out | Out C (32 x LVDS) |
| SELG | 1 | Out | Output Level Select (NIM/TTL) |
| nOEG | 1 | Out | Output Enable |
| GOUT | 2 | Out | Out G - LEMO (2 x NIM/TTL) |
| GIN | 2 | In | In G - LEMO (2 x NIM/TTL) |
| IDD | 3 | In | D slot mezzanine Identifier |
| SELD | 1 | Out | D slot Port Signal Standard Select |
| nOED | 1 | Out | D slot Port Direction |
| D | 32 | In | D slot Data In Bus |
| IDE | 3 | In | E slot mezzanine Identifier |
| SELE | 1 | Out | E slot Port Signal Standard Select |
| nOEE | 1 | Out | E slot Port Direction |
| E | 32 | Out | E slot Data In Bus |
| IDF | 3 | In | F slot mezzanine Identifier |
| SELF | 1 | Out | F slot Port Signal Standard Select |
| nOEF | 1 | Out | F slot Port Direction |
| F | 32 | In | F slot Data Out Bus |
| PDL_WR | 1 | Out | Write Enable |
| PDL_SEL | 1 | Out | PDL Selection (0=>PDL0, 1=>PDL1) |
| PDL_READ | 8 | In | Read Data |
| PDL_WRITE | 8 | Out | Write Data |
| PDL_DIR | 1 | Out | Direction (0=>Write, 1=>Read) |
| PDL0_OUT | 1 | In | Signal from PDL0 Output |
| PDL1_OUT | 1 | In | Signal from PDL1 Output |
| DLO0_OUT | 1 | In | Signal from DLO0 Output |
| DLO1_OUT | 1 | In | Signal from DLO1 Output |
| PDL0_IN | 1 | Out | Signal TO PDL0 Input |
| PDL1_IN | 1 | Out | Signal TO PDL1 Input |
| DLO0_GATE | 1 | Out | DLO0 Gate (active high) |
| DLO1_GATE | 1 | Out | DLO1 Gate (active high) |
| SPARE_OUT | 12 | Out | SPARE Data Out |
| SPARE_IN | 12 | In | SPARE Data In |
| SPARE_DIR | 12 | Out | SPARE Direction (0 => OUT, 1 => IN) |
| RED_PULSE | 1 | Out | RED Led Pulse (active high) |
| GREEN_PULSE | 1 | Out | GREEN Led Pulse (active high) |

**UML Diagram 2-5: GEMReadout_tb.vhd**


Set USR_ACCESS to 1
Set B,D,F to 0
Set PDL_READ, PDL0_OUT, PDL1_OUT,
DLO0_OUT, DL01_OUT, and SPARE_IN to 0
Call UWrite
Set ADDR to A_GEM_TRIG_WORD
Set DATA to 0x26
wait for 200ns
Call UWrite
Set ADDR to A_GEM_SOFT_TRIG
Set DATA to 0x0
wait for 200ns
Call UWrite
Set ADDR to A_GEM_SOFT_TRIG
Set DATA to 0x0
wait for 200ns
Call UWrite; Set ADDR to A_GEM_TX_WORD_0 and Set DATA to 0xA012
Call UWrite; Set ADDR to A_GEM_TX_WORD_1 and Set DATA to 0xC345
Call UWrite; Set ADDR to A_GEM_TX_WORD_2 and Set DATA to 0xE678
Call UWrite; Set ADDR to A_GEM_TX_WORD_3 and Set DATA to 0x9ABC
Call UWrite; Set ADDR to A_GEM_TX_WORD_4 and Set DATA to 0xDEF0
Call UWrite; Set ADDR to A_GEM_TX_WORD_5 and Set DATA to 0x1234
Call UWrite; Set ADDR to A_GEM_TX_WORD_6 and Set DATA to 0x5678
Call UWrite; Set ADDR to A_GEM_TX_WORD_7 and Set DATA to 0x9ABC
Call UWrite; Set ADDR to A_GEM_TX_WORD_8 and Set DATA to 0xDEF0
Call UWrite; Set ADDR to A_GEM_TX_WORD_9 and Set DATA to 0x1234
Call UWrite; Set ADDR to A_GEM_TX_WORD_A and Set DATA to 0x5678
Call UWrite; Set ADDR to A_GEM_TX_WORD_B and Set DATA to 0x9ABC
wait for 1 us
-- Check Event Counters (should be zero here) --
Call URead A_GEMXY_FIFOSIZE
X represents GEM block A or B
Y represents GEM 0-5 for each respective GEM block
-- Check Event Counters (should be NUM_EVENTS here) --
Call UWrite NUM_EVENTS Times (64)
Set ADDR to A_GEM_TX_START
Set DATA to 0x0
-- Check Event Counters --
Call URead A_GEMXY_FIFOSIZE
X represents GEM block A or B
Y represents GEM 0-5 for each respective GEM block
-- Readout Events NUM_EVENTS Times --
Call URead A_GEMXY_FIFOSIZE
Call UReadEvent A_GEMXY_EVENTDATA
X represents GEM block A or B
Y represents GEM 0-5 for each respective GEM block
-- Check Event Counters (should be zero here) --
Call URead A_GEMXY_FIFOSIZE
X represents GEM block A or B
Y represents GEM 0-5 for each respective GEM block

**UML Diagram 2-6: GEMReadout_tb.vhd**

does not have access to VFAT chips for generating and capturing data packets. See “UML Diagram 2-5” on page 29 and UML Diagram 2-6.

## 2.8.2 Verilog Modules

### v1495usr_hal.vqm

This module contains many of the logical functions that serve as a building foundation for the v1495 module. These have been automatically generated using the Synopsys Synplify design entry/synthesis tool. Synopsys Synplify is used to create, synthesize, and optimize a project and then generate a Verilog Quartus Mapping file (.vqm) for compilation in the Quartus II software. Reference here to the Altera website:

http://www.altera.com/support/software/nativelink/synthesis/synplicity/eda_view_using_synplty.html

# Chapter 3

# Readout Controller Library

## 3.1 Overview

The Readout Controller (formally known as the MVME6100) is a VMEbus single-board computer (SBC) with a powerful MPC7457 PowerPC processor running at 1.267 GHz. Combined with being the first series of SBCs to run 2eSST (two edge source synchronous transfers), the MVME6100 series delivers unprecedented data acquisition power for embedded system applications.

For our particular application, we have implemented VxWorks by Wind River as the operating system for the ROC (Readout Controller). The VxWorks OS can be found in numerous applications including Linksys WRT54G routers (version 5 and later), Boeing 787 and 747-8 aircraft, as well as many spacecraft including Spirit and Opportunity, the Mars Exploration Rovers.

Unlike other self-contained operating systems, VxWorks must have its programs cross-compiled on a separate operating system such as Linux. A description of the algorithms used for the Region 1 detector as well as how to compile these programs are contained within this chapter.

## 3.2 V1495 User Programs For The ROC

### 3.2.1 v1495Init

This function establishes a pointer value for the V1495 VME card within the MVME6100 memory map; the VxWorks documentation refers to this as a “memory anchor.” Without this, all other v1495 programs on the ROC will not know how to function properly since the memory location of the V1495 card will not be known to the ROC. To establish a connect between the ROC and the V1495, one needs to provide the 32-bit VME address of the V1495 as an argument to the function. The first 16 bits of the 32-bit VME address are physically set on the V1495 card using the hex switches as shown in Figure 3-1 below. Here we have chosen the VME address as 0x08110000.

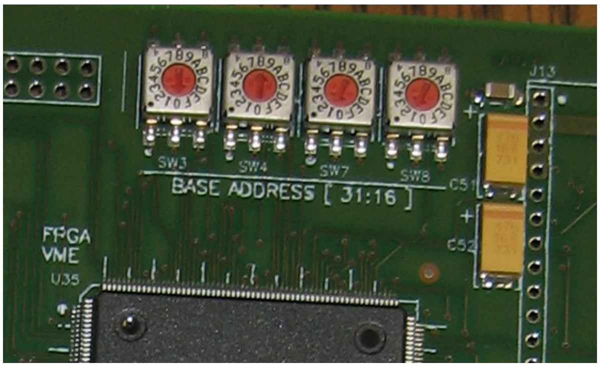


**Figure 3-1: Hexidecimal Address Switches on V1495**

Before the memory map address on the ROC is actually determined, however, this function checks to make sure that 32-bit addressing is supported on the ROC by making sure that the VxWorks version does not correspond to VXWORKS68K51: otherwise an error is thrown and a link will not be established.

Then, in order to actually establish a memory location in the ROC, a call to the VxWorks function “sysBustoLocalAddr” is made.

For all v1495 functions, with the exception of v1495Init, verification is made to make sure that a link has been established between the ROC and the V1495 card before performing any other functionality. This check will not be mentioned in any of the other functions’ respective sections but the reader should be aware of the presence of this check.

### 3.2.2 v1495ReadEvent

The v1495ReadEvent function queries the VFATs and then stores their data into the “V1495ReadoutStatus” struct before getting parsed up and sent to the DAQ.

The following values are stored into the “V1495Readout Status struct”:

- TotalBytes (Contains the total number of bytes recorded from all of the VFATs)
- TotalFrames (Contains the number of events/frames contained in all of the VFATs)
- DeltaBytes (In its current form, this performs exactly as “TotalBytes”)
- DeltaFrames (In its current form, this performs exactly the same as TotalFrames)
- LastSize (This records the value for the most recent number of words in the DataFIFO for the last event.)
- LengthDataFIFO (Contains the most recent number of words in the DataFIFO)
- LengthSizeFIFO (Contains the most recent number of words in the SizeFIFO)
- Sent (Contains the total number of events sent per VFAT)
- LastCapture[12] (Contains the actual DataOut data from the VFATs)

### 3.2.3 v1495FillData

This function simply takes a pointer to a 32-bit integer and fills it with the data from the structs of type “V1495ReadoutStatus” with 32-bit “chunks.” Here “chunks” refers to the fact that the data elements in the “V1495ReadoutStatus” structs are themselves not 32-bit elements; thus the sizes of the data elements are lost in this transition. The position and size of the data elements are just simply known on the receiving end of the data in the CODA data acquisition system.

This array of 32-bit pointers is filled out by first calling “v1495ReadEvent” to get the number of words in the event data   if there exists at least one event recorded on the V1495   and then subsequently filling out the “V1495ReadoutStatus” structs.

A more detailed description of the v1495ReadEvent function can be found in section “3.2.2 v1495ReadEvent” on page 34.

### 3.2.4 v1495Sprint

This function is a watered-down printout of some of the most useful registers recorded by the V1495 after a query of the VFAT data.

While not normally used in regular operation, this function can provide extremely useful information regarding VFAT operation; especially during debugging.

The registers printed off by this function include one of the following for each VFAT:

- Number of words in dataFIFO (dataFIFO is the FIFO containing the actual data from the VFAT)
- Number of words in sizeFIFO (sizeFIFO is the FIFO containing the number of words contained in the dataFIFO)
- EVENTSSENTH (upper 16 bits from the number of events currently recorded in the V1495 with an “event” being the number of times the DataValid line on the VFAT goes from high to low)
- EVENTSSENTL (lower 16 bits from the number of events currently recorded in the V1495)
- HARD_TRIGGER_WORD (the 4-bit hexadecimal value sent out on the T1 line whenever a HARD TRIGGER pulse is sent to the G1 port on the V1495; by default this is 0x4)

### 3.2.5 v1495StatusPrint

As with v1495Sprint, this function is not used during regular operation, but its output includes many useful values for debugging among which are the following for each VFAT:

- Number of TotalBytes
- Number of TotalFrames
- Number of Sent Bytes
- VFAT serial data in hexadecimal, 32-bit segments

If there is serial data present in the V1495 then the EventID as well as the ChipID for each VFAT will also be printed.

### 3.2.6 v1495HextoBin

As the name implies, this function simply changes the hexadecimal values reported in the v1495StatusPrint function to their binary equivalent. This can make it easier to determine which lines on the VFAT recorded a hit.

It must be noted, however, that this is a computationally intensive function and it has been found to crash the ROC while the CODA system is running, so this basically reduces it functionality to debugging.

### 3.2.7 v1495test

This function prints off several key registers found in the V1495. Each of these is useful on a VME card-by-card basis and could quite possibly be used to identify a particular V1495 card if more than one is used. The registers that are printed out are the following:

- Control Register
- Firmware Revision
- SelfflashVME
- Flash VME
- Self Flash User
- flash USER

- Configuration ROM

### 3.2.8 v1495reload

The v1495reload function is called from within v1495firmware and simply reboots the User FPGA by writing a '1' to the USER FPGA Configuration Register. This is a crucial step as the USER FPGA will maintain its original configuration unless this is done.

### 3.2.9 v1495firmware

While the purpose of the V1495 User Firmware is discussed in the "V1495 Module by CAEN" chapter, there are still several ways to actually accomplish the downloading process for the firmware.

One can use the precompiled software for Windows from CAEN as described in the V1495 User Manual, or one can use the VME backplane along with the V1495 Bridge FPGA to flash the User FPGA.

Using the latter method, the "v1495firmware" function was actually developed by a third party, namely C. Tintori of CAEN, but its overarching functionality is simple enough. **To not use it carefully can have adverse effects such as accidentally overwriting the FPGA VME firmware when it was intended to overwrite the FPGA User firmware.** The UML diagram for the firmware downloading process is found on page 45 of the V1495 User Manual.

The first parameter passed to this function is the ROC memory address for the V1495, e.g. 0x80110000 for our particular V1495. The second parameter which must be enclosed in quotes is the actual firmware file generated using Quartus II and should have the extension ".rbf". The third parameter tells the function whether to write to the "standard" image or the "backup" image. To accomplish this a '0' needs to be passed for the "standard" image and a '1' for the "backup" image. It must be noted that one cannot write a "backup" image to the User FGPA. The fourth and last parameter to this function tells the function whether to download to the User FPGA or to the VME FPGA; either a '0' or '1' respectively.

### 3.2.10 v1495release

This function simply sets the v1495 pointer to a NULL value thereby releasing the reference of the “V1495ReadoutCtrlRegs“ struct to the ROC memory map.

### 3.2.11 v1495DataReady

This function uses the FIFOLength register to determine which VFAT has the greatest number of entries. The greatest number of entries is then returned as an argument by the function.

### 3.2.12 v1495CalPulse

This function was originally intended to take the parameter passed to it to instruct the V1495 to send that number of CalPulses to the VFATs.

In its current form, this function can perform one of two tasks: Sending a number other than 0xFFFF will instruct the V1495 to send a CalPulse on the next MCLK signal; if 0xFFFF is sent then every time an external Hard Trigger signal is send, rather than sending a LV1A signal the V1495 sends a CalPulse signal. To have the V1495 return to normal operation one only needs to send a number other than 0xFFFF with this function.

### 3.2.13 v1495Reset

This function simply resets the V1495. It accomplishes this by writing a ‘0’ to the reset register in the V1495. For more information on how this is accomplished see the chapter “V1495 Module by CAEN”.

### 3.2.14 Other VxWorks Debugging Programs

m <32-bit address>,<Number of Bytes>

The ‘m’ command – which stands for “modify” – is a particularly useful, native VxWorks command that helps one to write to a given register address. The syntax for this function is given above. After the function enter the beginning address to be written to followed by how many bytes of data you wish to change. Hit Enter if you don’t want to change a value. Enter a ‘.’ and hit enter to exit the function.

d <32-bit address>,<Number of Bytes>

The ‘d’ command – which stands for “dump” – displays the values for the requested number of bytes. Enter a ‘.’ to exit the function. Note that this function displays the value of the bytes on the screen relative to their hexadecimal address: therefore, addresses ending in 0x0 display to the far left while addresses ending in 0xF to the far right with all others displaying in their respective, relative position.

Also, I was not able to find in any literature I had found what the characters next to the function are that appear between the two asterisks. I suspect that it may have to do with error correction, but this is of course just a guess.

## 3.3 Compiling programs for usage on the ROC

As previously discussed, the ROC uses the VxWorks operating system which cannot compile its own programs: it is therefore necessary to cross-compile its programs using another operating system such as Linux. As with most real-world applications, the software packages necessary to cross-compile our programs come pre-made and we have no exception here. The “makefile” used to cross-compile our programs is included in the appendix.

# Chapter 4

# VFAT Breakout Board

## 4.1 Introduction

As with any other real world implementation of an experiment comes the ubiquitous noise which completely undermines nearly everything one originally plans in the design stages; our experiment was no exception. It could, in fact, be stated that noise, which by definition is any unwanted signal in one's electronics, was the number one issue plaguing this experiment from our original designs through completion.

By its very nature, this project specifically contains several key issues that defy common noise-cancellation techniques. What these issues are along with a brief synopsis of the common noise-cancellation techniques will be discussed.

The VFAT Breakout Board employes several noise cancellation techniques which were utilized in this experiment and because of their profound consequences warrant a separate chapter. The first of these is the VFAT Breakout Board itself which employs several noise cancellation techniques in its design and layout, many of which seem to defy common sense. Second is the build-in radiation protection of the VFAT, charge-discriminating ICs, namely the Single Event Upset (SEU) triplicated logic and the Scan Chain ability for detecting erroneous digital gates. Also built into the VFAT controllers are several differential-type signals that extend beyond the VFAT Breakout Board that the ISU Laboratory for Detector Science, e.g. LDS, team designed. We also used many shielding and grounding techniques throughout to the experiment. Their respective caveats and explanations are discussed in this chapter. Lastly are the design and implementation of S-Curves for setting the appropriate threshold levels of the VFATs to mitigate spurious signals even after all of the aforementioned noise-cancellation techniques are utilized.

## 4.2 VFAT Breakout Board Design

For this project we went through several designs for the VFAT Breakout Board. Each design added functionality and noise suppression. The number one problem with circuit board design and Electromagnetic Compliance (EMC) in general is the nasty habit of electrons to not read schematics.[9] In this section, we will discuss the various features of the VFAT Breakout Board.

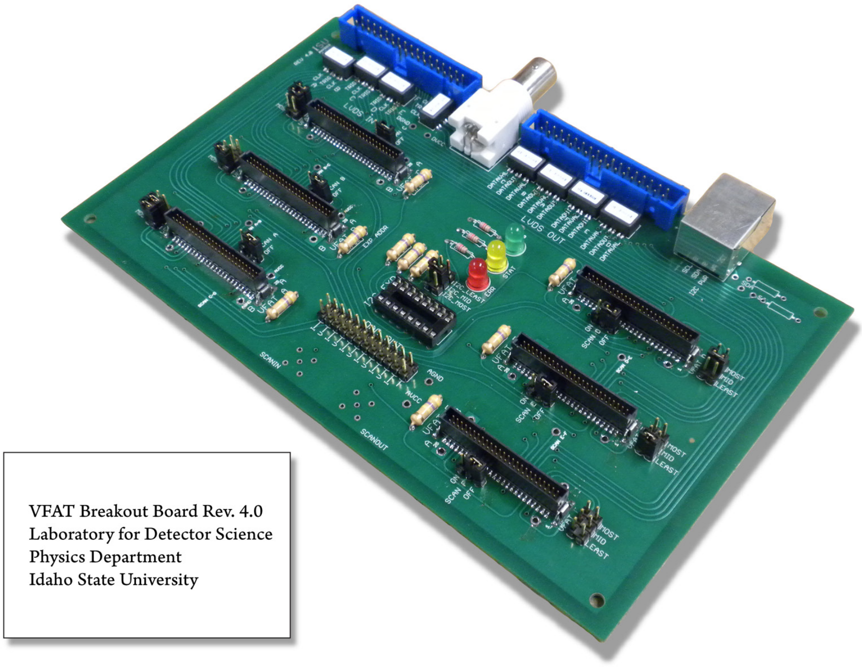
### 4.2.1 Status Indicator LEDs

There are three differently colored LEDs on the VFAT Breakout Board that indicate different statuses of the board. The green LED is simply designed to indicated that there is sufficient power to the breakout board. If the voltage supply drops below 2 V, the green LED will only glow very dimly. Any lower than this and the LED will turn off. Care must be taken not to increase the voltage on the board beyond 3 V. Besides being much higher than the 2.75 V rating of the VFATs, this voltage is likely to damage the green LED as it will begin to draw too much current.

The yellow and red status LEDs indicate a Status problem or Error. These LEDs can only be set by the I$^2$C Expander chip and are software controlled by the Gumstix controller. The yellow or red LED is turned on by writing a 0 to bit 7 or bit 6 (0x7F or 0xCF) to the I$^2$C Expander respectively. The bit lines run from 0 - 7, as is common for many electronics.

### 4.2.2 I$^2$C Address and Scan Enable Jumpers

Each of the binary I$^2$C addresses of the six ports on the VFAT Breakout Board are completely selectable via the jumpers physically next to that port. Each of these jumpers contain the top three MSBs of a seven-bit I$^2$C address and are labeled “MOST”, “MID”, and “LEAST”. One can therefore select any of six address from 0x16 - 0x112 by multiples of 16. By convension the port numbers are assigned in ascending order from 0x16 through 0x112 skipping over 0x32 because this is the default address of the on-board I²C expanded (located in the center of the board) if it is utilized. These are the addresses of the VFAT Ports A through F respectively. Jumping a pin will connect that respective address pin to ground thereby signifying a 0. If the jumper is not connected then internal pull-up resistors inside of the VFATs will pull the address line high thereby signifying a 1. For example to get the I²C address 0x16 one would jump the first two MSBs and leave the LSB floating. For the I²C address 0x112 one would leave all of the I²C address lines floating thereby leaving all three bits high.



VFAT Breakout Board Rev. 4.0
Laboratory for Detector Science
Physics Department
Idaho State University

**Figure 4-1**

The only exception to this are the address pins to the I²C Expander; these are set in the reverse fashion as the VFATs. Each address pin is jumped with it is to be set high, e.g. 1, and left floating for a low, e.g. 0.

The Scan Enable pins must be jumped either high or low. To enable the Scan Chain functionality of the VFATs one must jump the ScanEn pin high (as indicated by the silk screen on the board). Otherwise, one must jump this pin low. **Failure to do so will cause the VFAT chips to operate improperly.**

### 4.2.3 Soft Reset

The Soft Reset pins to the VFATs are all active low. These are software controlled via the I$^2$C expander. To reset a VFAT using a Soft Reset one must send a logic 0 to that respective VFAT by writing a 0 on bit line 0 - 5 for VFAT A - F respectively. Furthermore, since these Soft Reset lines are connected with pull-up resistors, these lines will all pull to VCC when an I$^2$C Expander chip is not present.

### 4.2.4 Differential-mode vs. Common-mode Signals

Basic lumped circuit theory relies on the assumption that the current carried in any transmission line is of equal and opposite polarity. This conforms to the “laws” of electromagnetics as well as making intuitive sense based on a closed-circuit model and Kirchhoff’s Current Law (or more fundamentally the Conservation of Charge equation). Even more misleading is that in basic microwave theory the “bottom” conductor is referred to as “ground” and all voltages are referred to the “upper” conductor. The basic deficiency in this model is the closed-circuit portion of it.

Two fixes are proposed in *Introduction to Electromagnetic Compatibility* by Clayton Paul.[9] In either case, the first thing one needs to do is allow current to flow outside of the transmission line via conductivity or radiation. This allows for unbalancing the two different currents of the two conducting paths of the transmission line. The currents that travel in the equal and opposite direction are referred to as the differential-mode currents. The currents that travel in the equal but same direction as referred to as the common-mode currents. Obviously, some other transmission medium must be present (although not necessarily accounted for in this model) or else the circuit could not form a closed path.

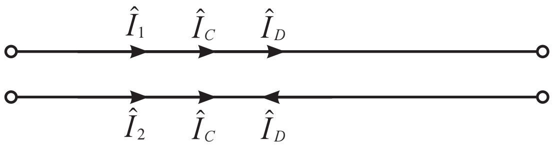


**Figure 4-2: Common graphical depiction of differential-mode and common-mode current on a two-wire transmission line.**

Figure 4-2 above shows graphically how the “upper” current and “lower” currents both contribute to the total current. Mathematically these can be calculated as the following:

$$\hat{I}_1 = \hat{I}_C + \hat{I}_D$$
$$\hat{I}_2 = \hat{I}_C - \hat{I}_D$$

A simple algebraic manipulation of these equations in terms of the common-mode current and the differential-mode current yields the following:

$$\hat{I}_D = \frac{1}{2}(\hat{I}_1 - \hat{I}_2)$$
$$\hat{I}_C = \frac{1}{2}(\hat{I}_1 + \hat{I}_2)$$

At this point there is nothing to suggest that this makes any substantial difference in how we analyze the circuit. However, as can be demonstrated, the radiative electric fields due to each differential-mode current are nearly equal and opposite thereby cancelling each other out while those from the common-mode currents actually add together for a significant net effect.

To see this we invoke the equations for the electromagnetic radiation from a Hertzian Dipole which consists of an infinitesimal current of length, *dl,* and a current, *I*, that has the same magnitude and phase throughout the whole segment.[9] This model is valid when the electrical length, $\lambda_0$, in terms of the wavelength of the signal on the wires is large when compared to the actual length, $\mathcal{L}$, e.g. $\lambda_0 > 10\mathcal{L}$. With a twisted-pair ribbon cable, $\lambda_0 = {\sim}7.5\,m$ for a 40 MHz signal. If we concentrate on the last section of cable closest to the detector, we can see that the current will have a constant magnitude and phase. Justification for neglecting the rest of the cable will follow shortly based on the particular layout of our detector and ribbon cable and the respective lengths from different sections of the cable to the detector.

A picture of this configuration is shown in Figure 4-3 on page 46 for demonstration purposes. The equations for the magnetic field intensity vector and the electric field intensity vectors are as follows:[9]

$$\hat{H}_r = 0$$
$$\hat{H}_\theta = 0$$
$$\hat{H}_\phi = \frac{\hat{I}\,dl}{4\pi}\beta_0^2 \cos\theta\left(j\frac{1}{\beta_0 r} + \frac{1}{\beta_0^2 r^2}\right)e^{-j\beta_0 r}$$

$$\hat{E}_r = 2\frac{\hat{I}\,dl}{4\pi}\eta_0\beta_0^2 \cos\theta\left(\frac{1}{\beta_0^2 r^2} - j\frac{1}{\beta_0^3 r^3}\right)e^{-j\beta_0 r}$$
$$\hat{E}_\theta = \frac{\hat{I}\,dl}{4\pi}\eta_0\beta_0^2 \sin\theta\left(j\frac{1}{\beta_0 r} + \frac{1}{\beta_0^2 r^2} - j\frac{1}{\beta_0^3 r^3}\right)e^{-j\beta_0 r}$$
$$\hat{E}_\phi = 0$$

where $\eta_0 = \sqrt{\mu_0/\varepsilon_0}$ is the intrinsic impedance of free space and $\beta_0 r = 2\pi r/\lambda_0$ which can be thought of as the electrical distance from the radiator.

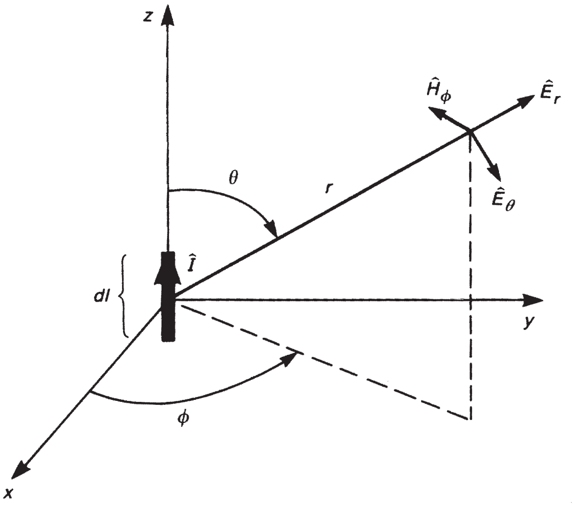


**Figure 4-3: Electric and Magnetic Fields from Hertzian Dipole[9]**

Of particular interest are the components that couple well to the GEM detector. This can be determined using the boundary conditions shown below for an ideal conductive plane since this is practically what we have inside of the detector.

$$\hat{n} \cdot \vec{\mathbf{D}} = \Sigma$$
$$\hat{n} \times \vec{\mathbf{H}} = \vec{\mathbf{K}}$$
$$\hat{n} \cdot \left(\vec{\mathbf{B}} - \vec{\mathbf{B}}_c\right) = 0$$
$$\hat{n} \times \left(\vec{\mathbf{E}} - \vec{\mathbf{E}}_c\right) = 0$$

Here the $\hat{n}$ unit vector is in the direction perpendicular to the plane and the "c" subscript indicates the fields inside of the conductor. From this we see that the perpendicular components of the electric field and the parallel components of the magnetic field will couple producing a surface-charge density, $\Sigma$, and a surface current, $\vec{\mathbf{K}}$, respectively. The last two equations demonstrate that the other remaining field components will be mitigated.

Putting everything together finally, there are two main practical dependencies that determine the whether or not we will experience coupling between the ribbon cable and the GEM detector. First, depending on the orientation of the cable, either the $\hat{E}_r$ or the $\hat{E}_\theta$ component of the electric field will couple better depending on which one is more perpendicular to the plane of the GEM detector. Second, the distance from the cable to the detector will govern whether we can make use of the near field or far fields portions of the aforementioned radiation equations.

For our experiment, the physical placement of the ribbon cable with respect to the GEM detector means that $\hat{E}_r$ is most likely to be the dominant coupling electric field component of the segment of cable physically closest to the detector – $\hat{E}_r$ being both perpendicular and within the near field regime $(\beta_0 r) \ll (\beta_0 r)^2, (\beta_0 r)^3$ – while $\hat{E}_\theta$ will dominate the rest of the cable coming from the V1495. At the site of the detector $\hat{H}_\phi$ will couple most when the cable is physically parallel.

For the segment of ribbon cable far away from the detector, the dominant $\hat{E}_\theta$ field will almost completely cancel out for the differential-mode currents. The contribution from each common-mode current will add together due to having the currents in the same direction. However, because this is in the far field region, the radiated power will be very small due to the relatively large electrical distance to the detector.

Considering the case of the differential currents on the segment of ribbon cable closest to the detector, if it weren't for the slight difference in distance to the measurement location these electric fields would cancel. Instead there is a slight net electric field. The electric fields due to the common-mode currents, however, actually add due to their being driven in the same direction. It is for this reason, despite their often being substantially smaller in magnitude than differential-mode currents, that radiated electric fields from common-mode currents are by far the predominant mechanism for producing radiated electric fields. In "4.5 Shielding and Grounding", we discuss how this effected our detector design and how we were able to resolve the problem.

### 4.2.5 Common-mode Chokes

Revision 4.0 of the VFAT Breakout Board includes common-mode chokes on all of the LVDS signals both entering and exiting the board. See Figure 4-4 on page 48. The purpose of these chokes is to mitigate the spurious common-mode signals on the ribbon cables between the VFAT Breakout Boards and the V1495. As explained in "4.2.4 Differential-mode vs. Common-mode Signals", these signals arise due to a difference in ground potential between the two different systems and cause radiative coupling to other components much more readily than differential signals.

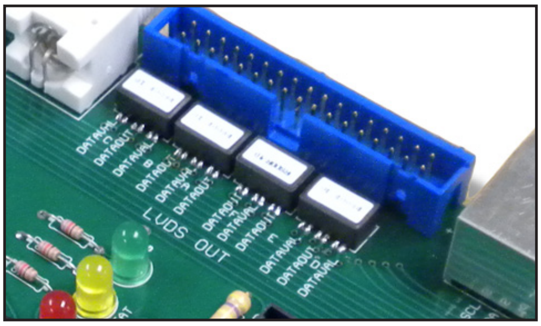


**Figure 4-4: Common-mode chokes on DataVal and DataOut**

These chokes are wire pairs wrapped in opposite directions about the same toroidal ferromagnetic core. Ideally this allows the differential signals to pass through completely unabated but any common-mode signals will find these coils to provide a very high impedance as the mutual inductances of the two different loops completely cancel each other out. Section "5.9 Common-mode Chokes" and "Chapter 9 Crosstalk" in *Introduction to Electromagnetic Compatibility*[9] give an excellent discussion on this subject. Figure 4-5, which can also be found in the aforementioned book, illustrates a basic sketch of this toroidal configuration used to block common-mode currents.

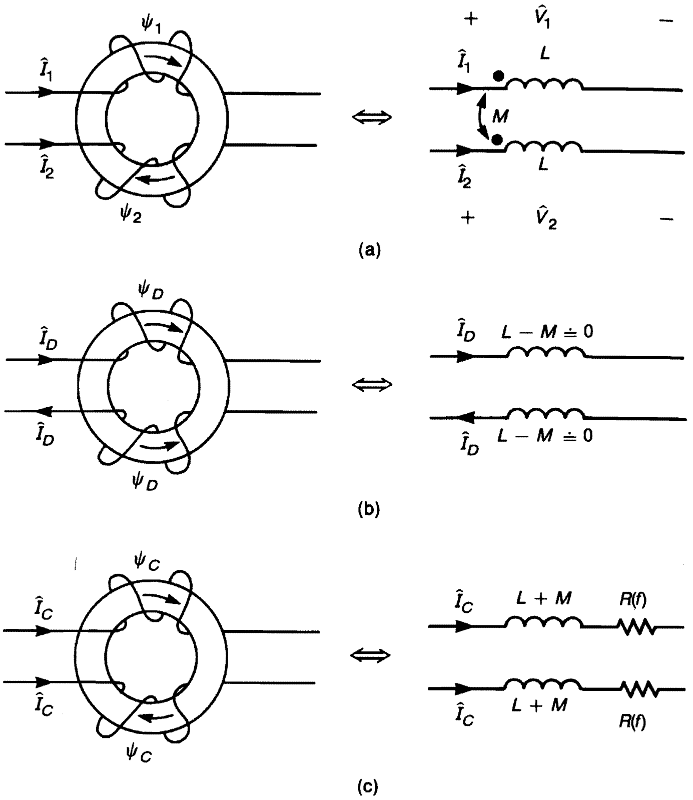


**Figure 4-5: (a) the currents of a two-wire line, (b) the differential-mode components, and (c) the common-mode components[9]**

### 4.2.6 Minimizing Current Loops

Undoubtably one of the most important improvements made on Rev 4.0 of the VFAT Breakout Board is the reduction of current loops. This is perhaps the only aspect of this circuit board that can explain why several of our ports failed to communicate via LVDS signals on previous versions of the board; no other single aspect can explain this since the electrical connections of the other boards are identical and we used the exact same cables as before.

It can be shown that as the loop area of a current path increases, the total impedance due to inductance increases as well. This comes as the difference between the inherent self inductance becomes much greater compared to the mutual inductances; the mutual inductances actually acts to decrease the voltage drop due to total inductance in a conductor.[9] In particular, it is extremely important to reduce the amount of voltage drop through the power and ground planes, otherwise the supply voltage ripple can filter through to both the analog and digital functionality of the components; even the ones that are not necessary responsible for creating the voltage supply ripple.

This also manifests itself in two other common ways:

The first is when there is a break in the power or ground plane supplying the current. This break forces the current to divert itself around the break thereby increasing the current loop and once again increasing the current loop and inductance. Every effort has been made to not break the ground or power planes in line with the regular currents needed by the VFATs or other components. There are very few instances where is was necessary to use a small portion of the power and/or ground planes for a regular signal, but these mainly towards the side farthest from the power supply.

Second, it is a commonly-held misconception that the digital and analog power planes should be electrically separated. While it is true that the two segments should physically be located in different regions of the board, it is not better to actually separate the two planes.[10][11] There are several reasons for this, the least of which is if currents travel from the digital power supply over to the analog ground or visa versa, the current loops from these can be tremendous.

### 4.2.7 Geographical Layout of Components

Besides using solid power and ground planes our latest version of the VFAT Breakout Board uses judicious geographical placement of the various components. The rules to obey for this are very basic: the highest frequency components should be as far away from the power supply of the board as possible and as the speed decreases and the precision of the voltage values increases (as in the case of the analog components), the components must be physically placed closer to the power supply.

There are two basic reasons for this layout: first, the longer the land distances between the high-speed components and the power supply, the more they will take advantage of the impedances already inherent to the board thereby lessening the effects of their higher harmonics; second, the longer the conductor distance that two currents from two different sources share together, the more that they will share common voltage drops due to common-impedance coupling. Thus, the longer one runs a sensitive signal along the same current path with a high-speed signal, the more likely it is for the sensitive signal to couple with the high speed signal and become distorted.[9] Figure 4-6 on page 51 shows the layout file of the VFAT Breakout Board Rev. 4.0 and the respective locations of the different types of signals with regards to their signals.

Lastly, one must as much as possible place the off-board connectors on the same side of the board. Failure to do so can set up oscillating signals across the power planes that can cause an inordinate amount of radiation.[9] So while it makes the layout much easier to place off-board connectors on different sides of the board, this must be avoided if at all possible. The only exception to this are the DAC outputs and the Scan Chain I/Os. In this case, these signals will only be utilized when most of the other LVDS signals are dormant.

## 4.3 Project-Specific Noise

### 4.3.1 Radiation Environment and Inability to Use Active-Network Noise Cancellation Techniques

The Region 1 detector is to be installed in a high-radiation environment; the normal usage of active-network noise cancellation techniques was therefore prohibited. Using active-networks, e.g. the usage of regular analog integrated circuits such as operational amplifiers, to effectively mitigate the noise was simply not possible because of their poor characteristics in a radiation rich environment with regards to both the added uncorrelated voltages and the permanent damage to their physical structure. This damage, even if it does not completely destroy the operation of the IC, can cause permanent shifts in operational amplifiers such as offset currents and voltages which may cause just as much noise in the circuit as one would hope to get rid of in the first place.

This, therefore, makes the design of the Region 1 detector all the more difficult. Designs such as Noise Shaping and Active Guard Drive were simply not feasible for this project.[8] Other techniques such as passive filtering, shielding, and special design techniques in the layout were employed by default and necessity.

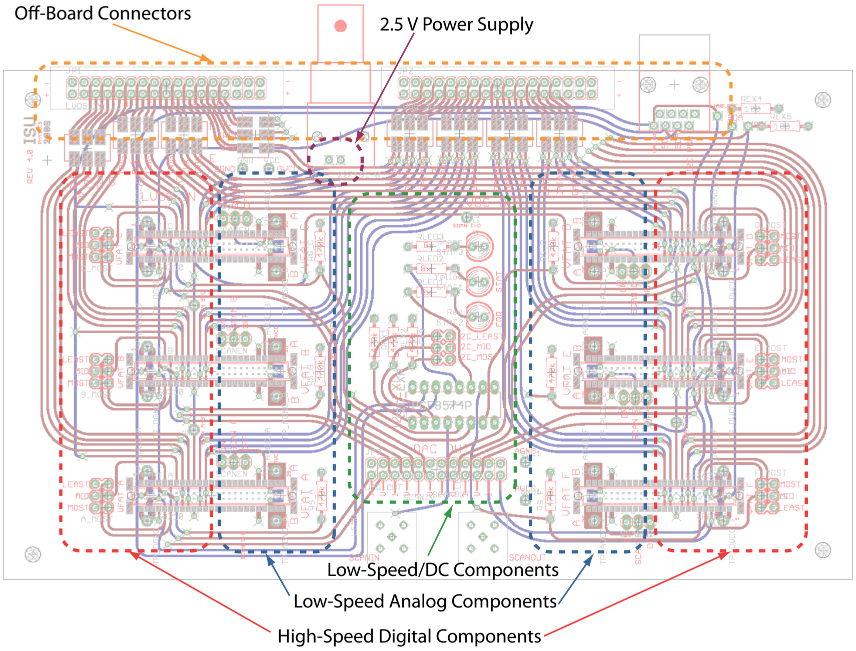


**Figure 4-6: VFAT Breakout Board layout with specific regions based on speed emphasized**

### 4.3.2 GEM Trigger Pulse

The trigger pulse from the last stage of the GEM HV distribution network (see Figure 4-7 on page 53) can be used to signify a hit on the detector. A hit on the detector releases a fairly consistent Gaussian-like pulse with a full-width-half-max (FWHM) of approximately 80ns from Trig Out. From a noise perspective it is useful to determine what the frequency content is of this pulse since one could then use this information to determine how much of the frequency energy content is from an actual hit and how much energy is from the noise and what frequencies the noise may contain.

It is known that the FWHM has the following relationship to the standard deviation, σ, of the Gaussian pulse:

$$\text{FWHM} = 2\sqrt{2\ln 2}\,\sigma$$

Furthermore, the frequency content for a Gaussian pulse can be found using the Fourier Transform. For a Gaussian pulse this has the following form:

$$\mathcal{F}\left\{e^{-t^2/2\sigma^2}\right\} = \sigma\sqrt{2\pi}\,e^{-\sigma^2\omega^2/2}$$

As expected, as the FWHM of the time-domain pulse decreases, the FWHM of its respective Fourier Transform increases thereby causing the pulse to contain high frequencies.

Using 100ns as the FWHM gives us: $\sigma = \frac{FWHM}{2\sqrt{2\ln 2}} = 42.466 \cdot 10^{-9}\,s$

The standard deviation in the frequency domain can then be found as the following:

$$\sigma_f = 1/\sqrt{4\pi^2\sigma^2} = 3.748\,MHz$$

Furthermore, since approximately 95% of the energy content is contained within two standard deviations of any Gaussian curve, any spectral content above double this frequency, namely 7.5 MHz, can be considered noise and is subject to removal via filtering.

### 4.3.3 Frequency Dependent Noise

Since we often had coupling of the 40 MHz MCLK lines to the detector, it would be feasible to build a passive filter at the Trig Out site to filter out the energy content from the MCLK. A similar filter might have been used at the V1495 site of the ribbon cables between the VFAT Breakout Board and the V1495. The following pictures, show the Trig Out pulses both before and after a built-in 20MHz filter was applied on the signal by the oscilloscope. Because the filter is at the oscilloscope input, the noise from the MCLK is in reality still there.

In Figure 4-2 above, the light-blue colored graph is the Trig Out signal with a very high level of noise. One can see from the red graph, the FFT of this signal, that there is an enormous amount of energy at 40 MHz and its higher level harmonics, i.e. 80 MHz, 120 MHz. A hit signal on this line would be completely hidden and undetectable at this point.

With exactly the same setup, Figure 4-3 shows that applying a 20 MHz filter to the output of Trig Out actually lowers the RMS noise from ~8.37 mV to ~1.37 mV. One can also see that the energy from the first harmonic of 40 MHz alone dropped 9dB and the other higher-order harmonics have all but disappeared at this point.

When a signal from Trig Out was captured on the oscilloscope as in Figure 4-4, the ~15 mV peak is easily discernible from the remaining noise. Also note that the extra energy from the pulse seems to be of a Gaussian shape and lie mostly below 10 MHz. This is in agreement to the energy calculations in the previous section if one were to adjust for this particular pulse having a FWHM of ~80ns rather than 100ns.

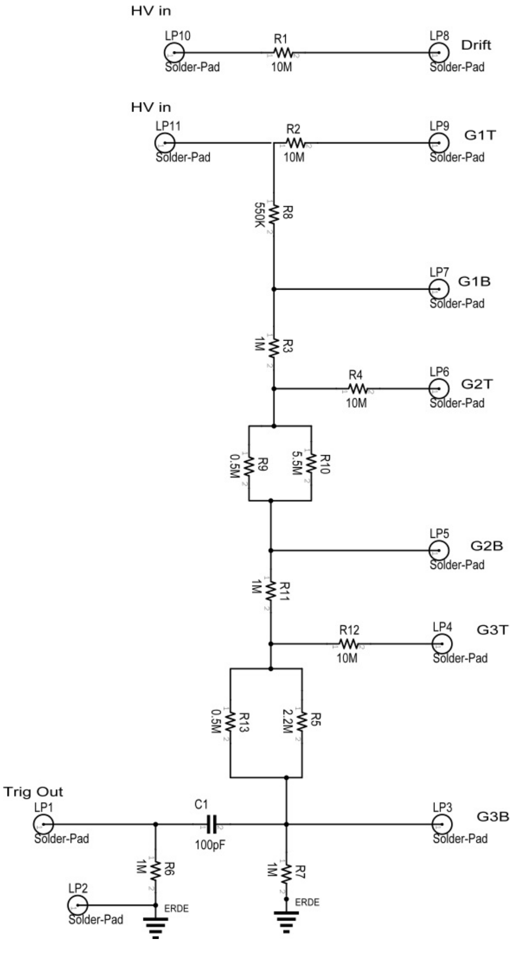
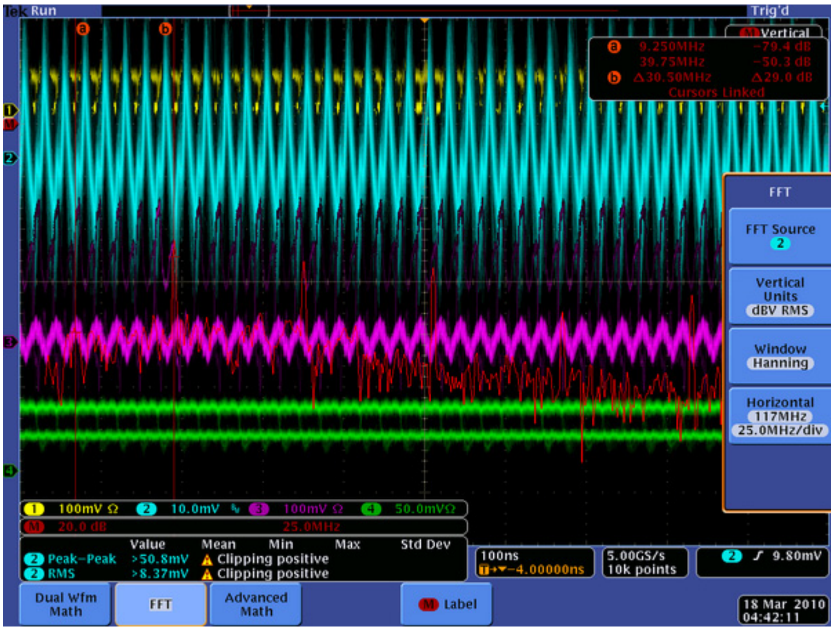


**Figure 4-7: GEM HV Distribution Network**

In the end, it was found that using ungrounded shielding on both the ribbon cables to the VFAT cards themselves and the twisted-pair cables between the VFAT Breakout Board and the V1495 was more than effect enough to pull the

RMS noise voltage on Trig Out to ~5 mV. (see “4.5 Shielding and Grounding”) At this level, the Trig Out pulse with an amplitude of ~10-20 mV should readily be detected.



```
a   9.250MHz      -79.4 dB
    39.75MHz      -50.3 dB
b   Δ30.50MHz     Δ29.0 dB
       Cursors Linked
```

FFT Source 2

Vertical Units dBV RMS

Window Hanning

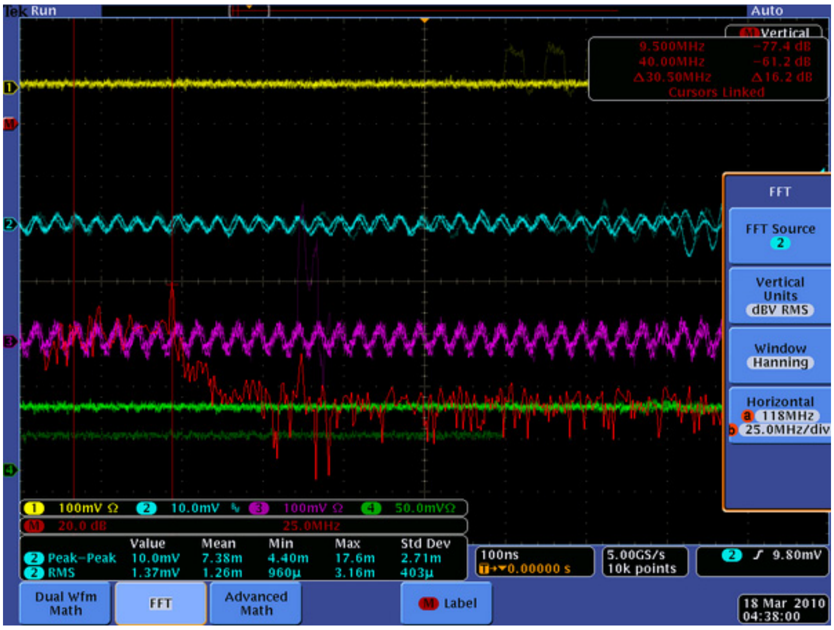
Horizontal 117MHz 25.0MHz/div

```
1  100mV Ω   2  10.0mV      3  100mV Ω   4  50.0mVΩ
M  20.0 dB              25.0MHz
               Value     Mean     Min     Max     Std Dev
2 Peak-Peak    >50.8mV   ⚠ Clipping positive
2 RMS          >8.37mV   ⚠ Clipping positive

100ns              5.00GS/s      2  ⨍ 9.80mV
T→▼-4.00000ns      10k points

18 Mar 2010
04:42:11
```

Figure 4-8: Unfiltered random signal on Trig Out with high level of cross-talk with MCLK



```
    9.500MHz      -77.4 dB
    40.00MHz      -61.2 dB
    Δ30.50MHz     Δ16.2 dB
       Cursors Linked
```

FFT Source 2

Vertical Units dBV RMS

Window Hanning

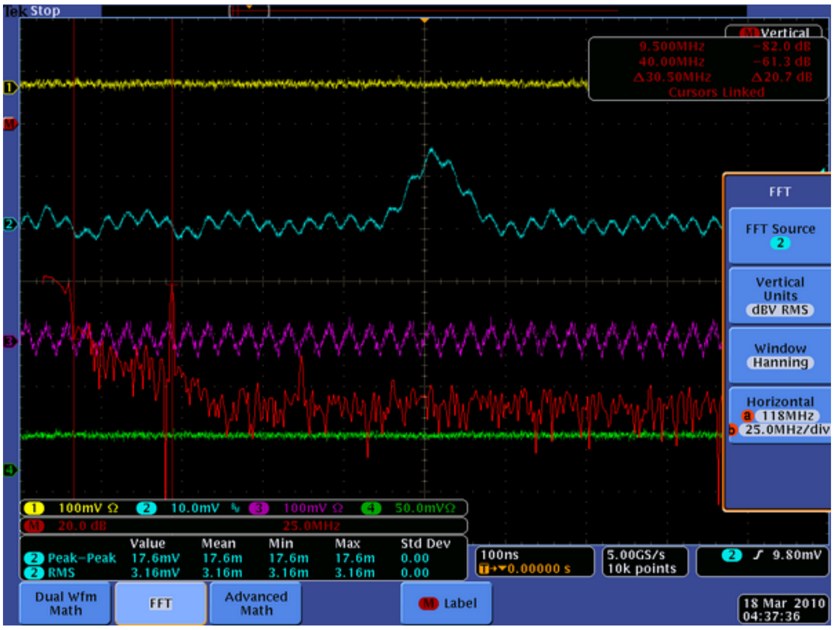
Horizontal a 118MHz b 25.0MHz/div

```
1  100mV Ω   2  10.0mV      3  100mV Ω   4  50.0mVΩ
M  20.0 dB              25.0MHz
               Value     Mean     Min     Max     Std Dev
2 Peak-Peak    10.0mV    7.38m    4.40m   17.6m   2.71m
2 RMS          1.37mV    1.26m    960µ    3.16m   403µ

100ns              5.00GS/s      2  ⨍ 9.80mV
T→▼0.00000 s       10k points

18 Mar 2010
04:38:00
```

Figure 4-9: 20 MHz filter applied to random signal on Trig Out



| | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 2 Peak–Peak | 17.6mV | 17.6m | 17.6m | 17.6m | 0.00 |
| 2 RMS | 3.16mV | 3.16m | 3.16m | 3.16m | 0.00 |

Figure 4-10: 20 MHz filter applied to actual Trig Out signal with MCLK running VFATs

### 4.3.4 Thermal Noise of the GEM High-voltage Distribution Network

Having familiarity with thermal noise and then observing a circuit built for DC high voltage with large resistors can cause a justified uneasiness. The easiest way to explain this is to first explain how thermal noise in a circuit is represented. Perhaps the most common way for representing quantifiable noise in general, $x_n(t)$ , is by its root-mean-square (rms) value.

$$X_n = \left( \frac{1}{T} \int_0^T x_n^2(t) \right)^{1/2}$$

This rms value can be interpreted as the amount of power that the signal would consume if placed across a 1-Ω resistor.

Furthermore, it is commonly known that given multiple sources of noise, e.g. $x_{n1}(t), x_{n2}(t), \ldots$ , that their summation can be determined as follows:

$$X_n^2 = \frac{1}{T} \int_0^T [x_{n1}(t) + x_{n2}(t)]^2 = X_{n1}^2 + X_{n2}^2 + \frac{2}{T} \int_0^T x_{n1}(t) x_{n2}(t)\, dt$$

And if the two signals are uncorrelated, the average of their product disappears, and the rms values add up as the square-root of the sum of the squares:

$$X_n = \sqrt{X_{n1}^2 + X_{n2}^2}$$

Thermal noise (a.k.a. Johnson-Nyquist Noise) is an inherent noise meaning that the noise is present simply due to the random fluctuations of the atoms in the composite material. In the case of resistors, the electrons themselves will move about in the material enough to cause measurable voltage/current fluctuation. These fluctuations average to zero (this is obvious or else there would be net positive power consumed or created) and occur even if the resistor is sitting on a laboratory workbench completely disconnected from anything.

The power spectral density (or noise per unit frequency) for thermal noise is given as follows:

$$e_R^2 = 4kTR$$

Here $e_R^2$ can be taken as the power spectral density of a voltage source in series with a noiseless resistor.

Returning to the schematic of the GEM HV distribution network shown again in Figure 4-1, we see that on the voltage division side of the network there are several very high valued resistors each of these generating its own thermal noise.

For this analysis we ignore the effects of the GEM foil capacitors, and combine all of these resistances into one equivalent resistor. The result is the following:

$$\begin{aligned} e_{R_{eq}}^2 &= e_{R_8}^2 + e_{R_3}^2 + e_{R_9 \| R_{10}}^2 + e_{R_{11}}^2 + e_{R_5 \| R_{13}}^2 + e_{R_7}^2 \\ &= 4kTR_{eq} = 1.65 \times 10^{-20}\, W/Hz \cdot 3.4157\, \mathrm{M\Omega} \\ &= 55.3 \times 10^{-14}\, V^2/Hz \end{aligned}$$

This leaves us with the following equivalent circuit:

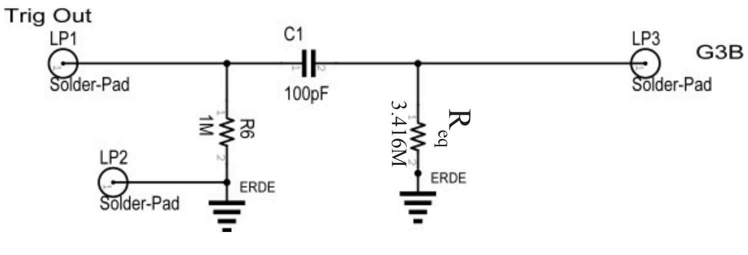


**Figure 4-11: GEM HV Equivalent Noise Circuit**

All we have remaining is to refer this noise to the Trig Out. It is also known that one can refer the noise spectrum, $S_{In}(f)$ , through a linear time-invariant system with transfer function H(s) as follows:

$$S_{Out}(f) = S_{In}(f)|H(f)|^2$$

On the last leg of the HV Distribution network we have a simple highpass RC filter. This gives us the following relationship for the power spectral density of the noise referred to the GEM Trig Out:

$$\begin{aligned} S_{Out}(f) &= S_{R_{eq}}(f)\left|\frac{V_{out}}{V_{R_{eq}}}(j\omega)\right|^2 \\ &= 4kTR_{eq}\frac{4\pi^2R_6^2C_1^2f^2}{4\pi^2R_6^2C_1^2f^2+1} \end{aligned}$$

And since for our circuit, $(4\pi^2R_6^2C_1^2)^{-1/2} = 1592\ Hz$, this gives us a very low cutoff frequency for the high-pass filter of < 2 kHz. Therefore, for all intents and purposes, this highpass filter will filter out relatively little of the thermal noise generated by these resistors.

Lastly, we must simply integrate the total noise contribution in the valid field of measurement. From our frequency dependent noise discussion we can see the valid frequency range to be ~100 MHz. This gives us a total RMS voltage noise of $\sqrt{55.3 \cdot 10^{-14}\ V^2/Hz} \cdot \sqrt{100\ MHz} \backsimeq 1mV$. Thus, we cannot hope to see the Trig Out signal if it drops below this value. Fortunately, we see that a typical Trig Out pulse has a peak value of ~15 mV.

## 4.4 VFAT2 Built-in Radiation Protection

The VFAT2 was created with the express purpose of operating in a radiation environment. In testing, the VFAT was able to withstand 10 Mrad of radiation with no observable effects[2]. Among other potentially built in features of the VFAT to ward against radiation effects, the VFAT offers two user accessible radiation protection features; the first of these is the Single Event Upset protection and the second is testing of digital gate through the Scan Path. With these features the VFAT is actually designed to withstand 100MRad of radiation.

### 4.4.1 Single Event Upset (SEU) Protection

The VFAT employs triplicated flip-flops in the Control Logic. If there is ever a discrepancy between the triplicated logic, a 2 to 1 “vote” is made by way of a logical “OR” of all of these outputs. The result is then used to increment an 8-bit synchronous counter. The result of this counter is then stored in an 8-bit register called the “UpsetReg”. As with all of the registers on the VFAT, the $I^2C$ can then read back the values of this register.[1]

## 4.4.2 Scan Path

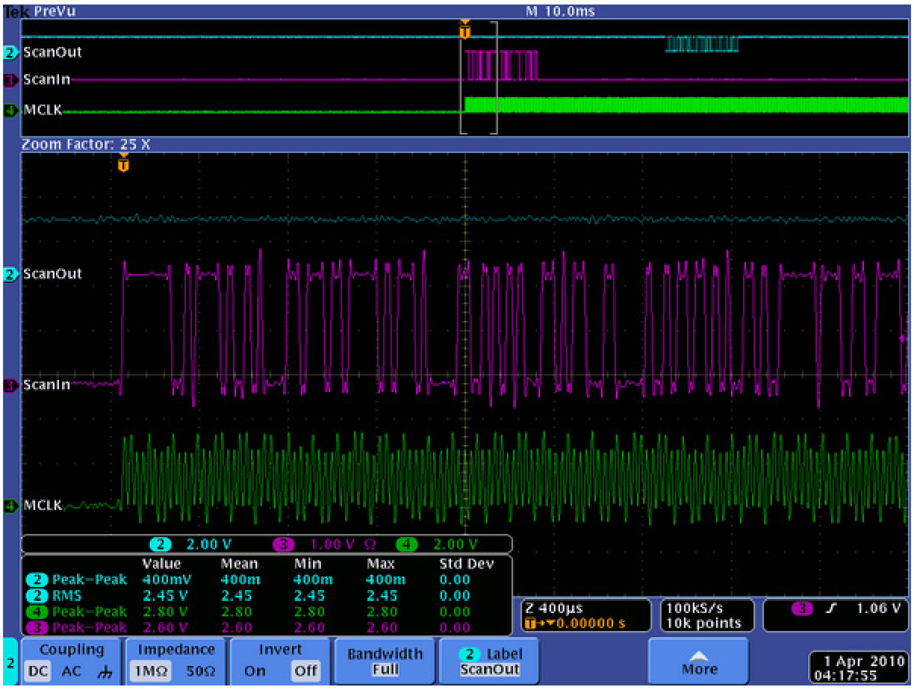
As can also be read in the VFAT Manual, the Scan Path (a.k.a Scan Chain) enables testing of all of the logic flip flops. This is done by cascading all of the outputs of the flip flops rather than having them connected in their normal configuration. When put into “Scan Enable” mode, a serial pattern can be clocked into the VFAT via the ScanIn pin. After having passed through all of the available, cascaded flip flops the digital signal will then pass out of the ScanOut pin. The ScanClk pin is the same as the LVDS MCLK pin.

If all of the flip flops are operational then the two patterns will match with the exception of the signal being inverted. If, for instance, one of the flip flops is stuck high or low, the Scan Chain will read completely high or completely low respectively. If one or more of the flip flops are stuck at a fault, the VFAT chip may still be operational, but this will effectively count the “UpsetReg” once per fault per clock cycle. If more than one flip flop is stuck high or low then one would only be able to detect it by counting the change in the UpSetReg and then dividing by the number of MCLK periods that have transpired.

### Implementation

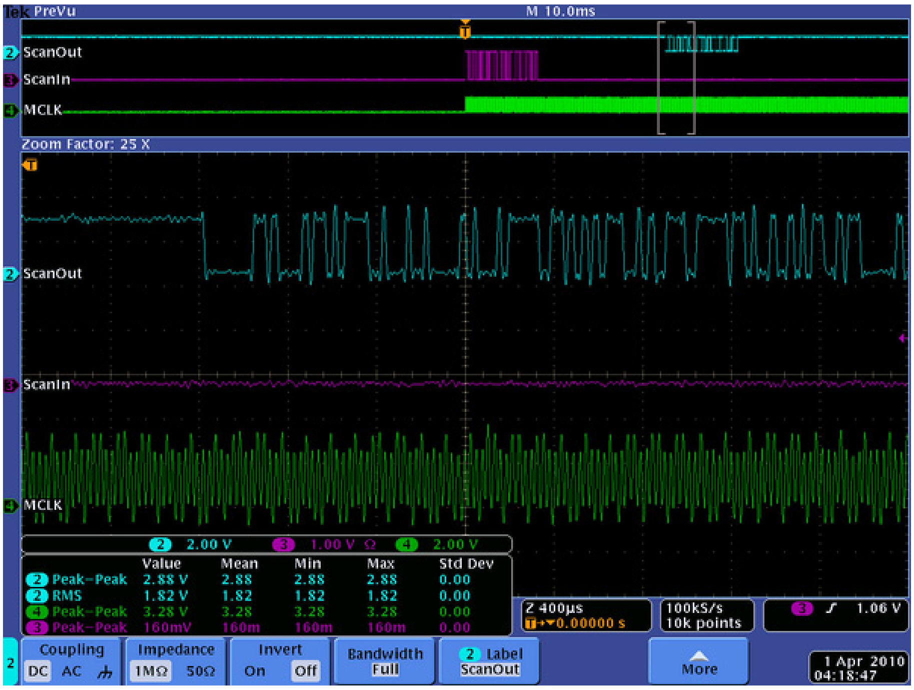
Our own implementations of the Scan Chain are shown in the figures below. In Figure 4-12 on page 59 a binary pseudo random bit sequence (PRBS) with base 8 is injected into the ScanIn of a single VFAT; the output can then be seen, Figure 4-13 on page 59, on the ScanOut pin of the same VFAT.

For some reason unknown to the author, the bit pattern is actually inverted when passed through a single VFAT. At first it was believed that this was a simple polarity mistake in the setup. However, as demonstrated by directly daisy-chaining two VFATs on our VFAT Breakout Board in Figure 4-14 on page 60, using the exact same setup (with the except of the daisy-chained VFATs), the signal again had the proper polarity as can be seen in Figure 4-15 on page 60.



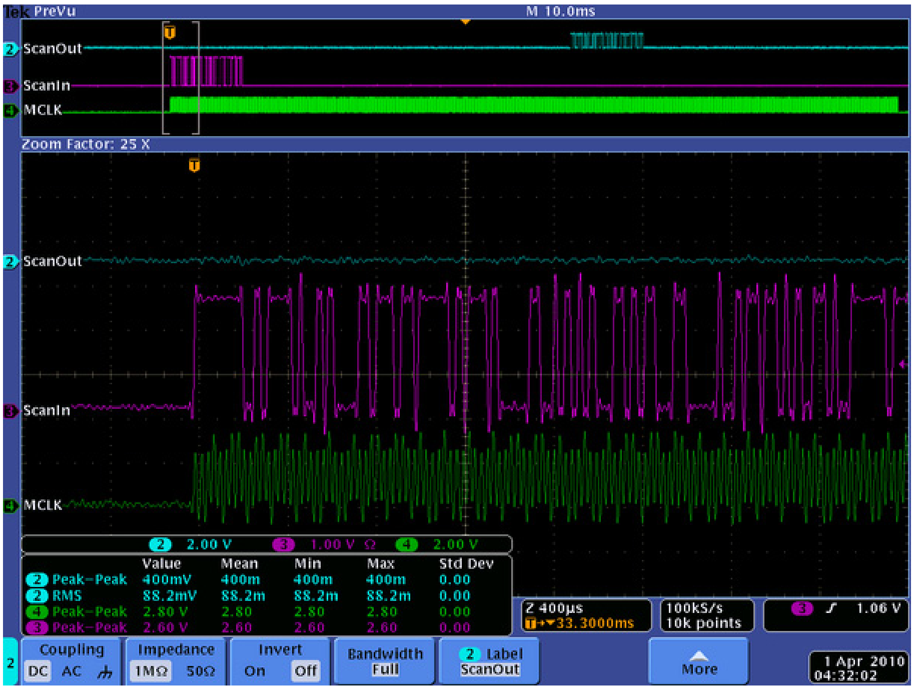
| | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 2 Peak−Peak | 400mV | 400m | 400m | 400m | 0.00 |
| 2 RMS | 2.45 V | 2.45 | 2.45 | 2.45 | 0.00 |
| 4 Peak−Peak | 2.80 V | 2.80 | 2.80 | 2.80 | 0.00 |
| 3 Peak−Peak | 2.60 V | 2.60 | 2.60 | 2.60 | 0.00 |

Figure 4-12: Single VFAT ScanIn PBRS with base 8 bit pattern



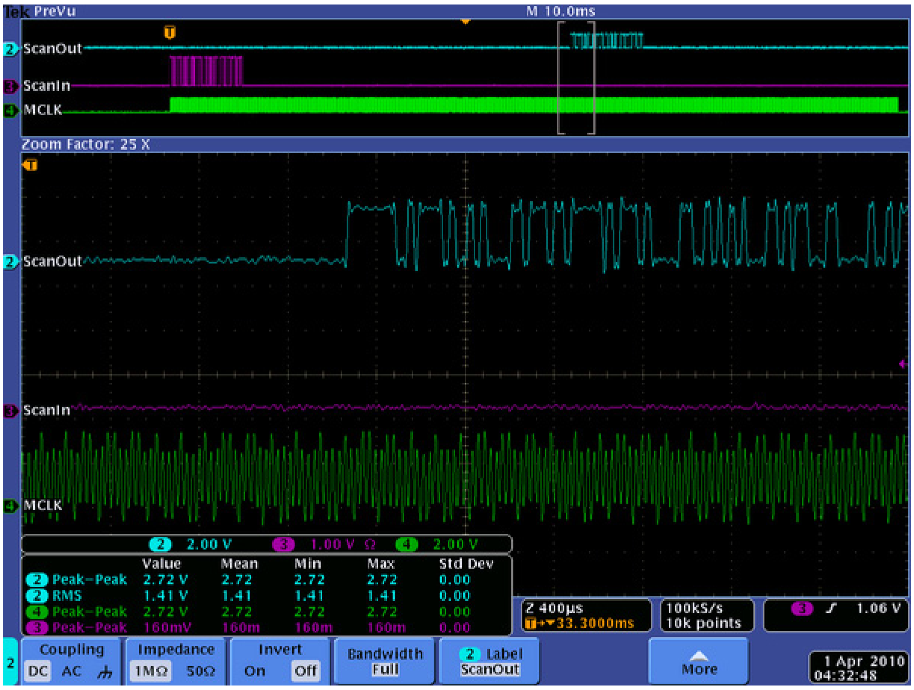
| | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 2 Peak−Peak | 2.88 V | 2.88 | 2.88 | 2.88 | 0.00 |
| 2 RMS | 1.82 V | 1.82 | 1.82 | 1.82 | 0.00 |
| 4 Peak−Peak | 3.28 V | 3.28 | 3.28 | 3.28 | 0.00 |
| 3 Peak−Peak | 160mV | 160m | 160m | 160m | 0.00 |

Figure 4-13: Single VFAT ScanOut



|  | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 2 Peak−Peak | 400mV | 400m | 400m | 400m | 0.00 |
| 2 RMS | 88.2mV | 88.2m | 88.2m | 88.2m | 0.00 |
| 4 Peak−Peak | 2.80 V | 2.80 | 2.80 | 2.80 | 0.00 |
| 3 Peak−Peak | 2.60 V | 2.60 | 2.60 | 2.60 | 0.00 |

Figure 4-14: Daisy-Chained VFATs ScanIn bit pattern



|  | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 2 Peak−Peak | 2.72 V | 2.72 | 2.72 | 2.72 | 0.00 |
| 2 RMS | 1.41 V | 1.41 | 1.41 | 1.41 | 0.00 |
| 4 Peak−Peak | 2.72 V | 2.72 | 2.72 | 2.72 | 0.00 |
| 3 Peak−Peak | 160mV | 160m | 160m | 160m | 0.00 |

Figure 4-15: Daisy-Chained VFATs ScanOut bit pattern

## 4.5 Shielding and Grounding

### 4.5.1 Implementation

As previously discussed, we ultimately found that the last remaining noise cancellation technique that was necessary to employ was shielding the VFAT ribbon cables as well as the LVDS ribbon cables with either tinfoil (as in the case of the VFAT ribbon cables) or ungrounded mesh cladding on the twisted pair cables. As a quick side note; it was found earlier that ungrounded shielded cable without twisted pairs caused two of our VFAT cards to quit transmitting properly.

For our latest HRRL and cosmic particle experiments we implemented this configuration, and it proved to be quite effective. In Figure 4-16 on page 62, we can see the setup for the HRRL experiment. Emphasis has been added to where the shielded cables are located for reference.

### 4.5.2 Explanation

Despite the fact that there are common-mode current chokes on the inputs and outputs of the VFAT Breakout Board, there can still be common-mode voltage on the lines. In other words, just because there is a high impedance to common-mode sources does not mean that the voltage goes away. In fact, it can be quite the opposite. Not having common-mode current can actually raise the voltage because there is no longer any voltage drop from the common-mode source impedance. By essentially attempting to eliminate the common-mode current we actually raised the common-mode voltage on the line. This is all the more pronounced because have low-resistance, 50-Ω cables. Having raised the common-mode voltage actually means that we will have a higher propensity for capacitive coupling to other parts of the system.

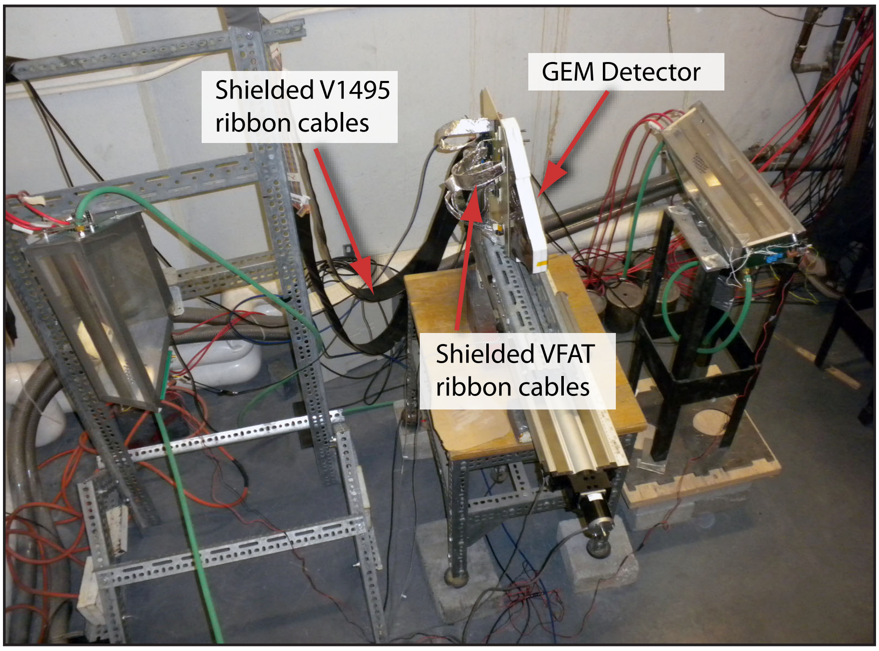


**Figure 4-16: HRRL Experiment Setup**

Both the shielding for the VFAT ribbon cables as well as the V1495 ribbon cables successfully employ the same basic technique to drastically reduce the coupled voltage to the GEM Detector-- surround the ribbon cable with an ungrounded metallic shield. There are a number of reasons why this would at first glace appear to defy common sense.

First, due to the speed of the signal and the proximity of the components, these fields are primarily static electric fields causes by a change in common-mode voltage on the respective cables. Gauss' Law states that the total charged contained within any Gaussian surface can be determined by the integration of the total electric flux found at that surface. Without any insulators present there is nothing to block the electric fields from leaving the shielding as if there were nothing there at all.

Looking at this more quantitatively we can find the answer. We can use the multiconductor transmission-line (MTL) model as proposed by Clayton Paul in *Introduction to Electromagnetic Compatibility* with a slight variation.[9] There in section 9.7.2 pg. 651 he proposes the following four-conductor model as shown in Figure 4-17.

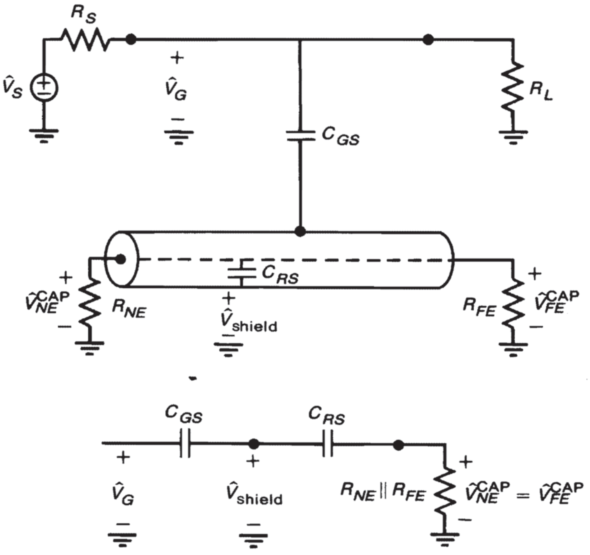


**Figure 4-17: Four-wire model to demonstrate effects of shielding[9]**

Here he has shown the signal from the unshielded wire capacitively coupling to the signal wire inside of the shield. In our experiment we are interested in shielding the wires containing the signals and thereby protecting the detector from coupling capacitively to the signals wires. We can simply exchange the source voltage and exchange all of the impedances as shown in Figure 4-18 on page 64. Note how the equivalent circuit does not change. This makes the math exactly the same for our purposes:

$$\hat{V}_{NE}^{CAP} = \hat{V}_{FE}^{CAP} = \frac{j\omega R C_{RS} \| C_{GS}}{1 + j\omega R C_{RS} \| C_{GS}} \hat{V}_G$$

where

$$R = \frac{R_{NE} R_{FE}}{R_{NE} + R_{FE}}$$

$$C_{RS} \| C_{GS} = \frac{C_{RS} C_{GS}}{C_{RS} + C_{GS}}$$

This coupling equation is nothing more than another high-pass filter. Frequencies below the cutoff frequency of $(2\pi R C_{RS} \| C_{GS})^{-1}$ will be attenuated but frequencies above this are passed through completely. Furthermore, decreasing the equivalent capacitance will push the cutoff frequency higher and attenuate the coupled signal more.

What can also be said for this model is that capacitors add in series just like resistors in parallel. That is to say that the lowest capacitor (particularly if it much lower than the other) will dominate the equivalent capacitance. Normally, $C_{RS} \gg C_{GS}$ which would imply that $C_{RS} \| C_{GS} \cong C_{GS} \cong C_{GR}$. This basically means that for a capacitive system where the line couples more strongly to the shield than to the external system it is as if the shielding is not even there. This seems to confirm what was intuitively discovered from the argument based on Gauss' Law. And yet the shielding works; what is going on here?

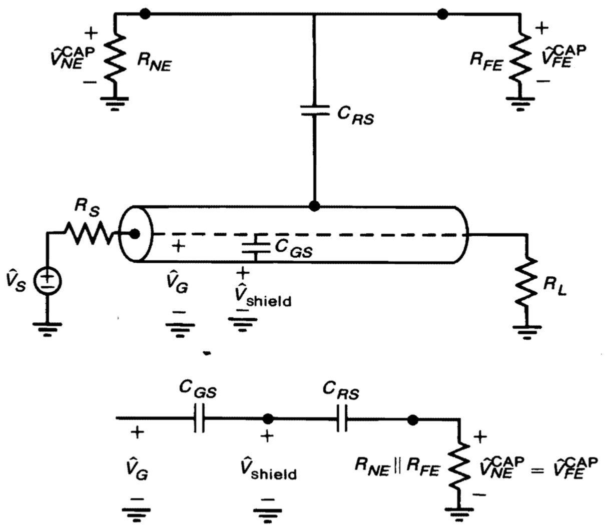


**Figure 4-18: Four-wire model modified for the Detector 1 shielding**

There can be only one answer; as long as there are no egregious errors in this model then it must be that the capacitance from the signal wires to the shield couples much *less* strongly than the shield to the GEM detector or even the signal wires to the GEM detector, e.g. $C_{RS} \ll C_{GS}$. This is a rare case, but it also makes sense. Despite the fact that the shield is a metal casing completely surrounding the signals wires at a very short distance (all the necessary components of a strong capacitor), the GEM detector itself is a series of large -- respectively speaking -- copper plates. Apparently this combination of copper plates on the GEM detector makes for a larger capacitor than the shielding wrapped around the signal wires.

Normally, the only way to get this kind of effect is to provide a solid ground connections at points $0.1\lambda$, e.g.10% of the electrical wavelength, to provide a grounding potential on the shielding cable. This assures that no voltage can couple to the shield thereby protecting the external circuits from the signal wire.

## 4.6 CalPulse

Fortunately, the VFAT contains the ability to self-test and calibrate each of its 128 channels individually without the need for an externally generated signal. The basic process for how known voltage pulses can be injected onto the input lines for the preamplifiers as follows:

Each channel contains a separate $C_{inj}$ capacitor that during normal operation is grounded on the side not connected to the channel. If the respective “CalOut” bit for a given channel is set high, that particular $C_{inj}$ capacitor is then disconnected from ground and connected to the CalOut channel. There are two highly capacitive nodes between which this CalOut signal will switch during the pulse-generating phase of operation; the first node is controlled by the VCal value as set in the $I^2C$ registers. The second value is a constant voltage reference. The first node is also referred to as Vlow, and the second node is referred to as Vhi. Depending on the value of the “CalPolarity” bit in primary register 0x00 (0 for a positive pulse and 1 for a negative pulse), whenever a “CalPulse” T1 signal is sent, the CalOut signal is either switched from Vhi to Vlow (as in the case of a negative pulse) or from Vlow to Vhi (as in the case of a positive pulse). The operation of the CalPulse is shown in the schematic of Figure 4-19 below.

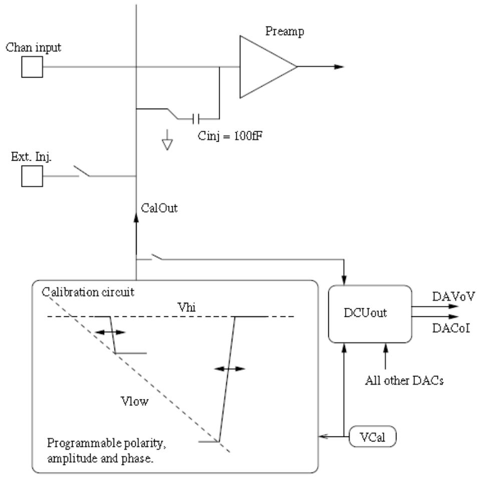
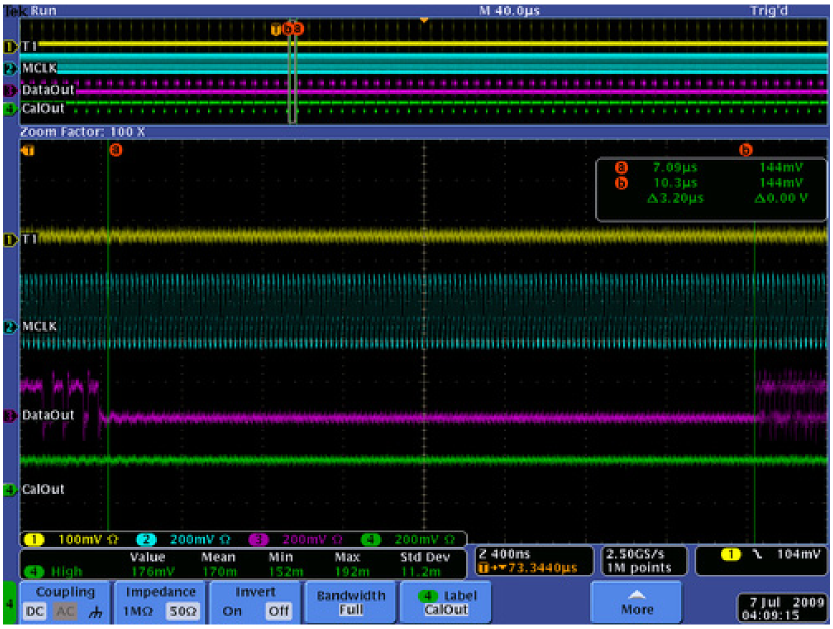


**Figure 4-19: Internal test pulse generator for calibration.**

The results of our own CalPulse implimentation are also shown. In Figure 4-20 on page 66, we can see that all of the channels are turned off and even though a CalPulse signal is sent on the T1 line none of the channels (which can be located between the cursors) actually record a hit. In Figure 4-21 on page 66, we were able to turn on just the odd channels. Finally in Figure 4-22 on page 67, we arbitrarily turned on channels 25 and 100 – just for demonstration purposes – to show that we could control individual channels.

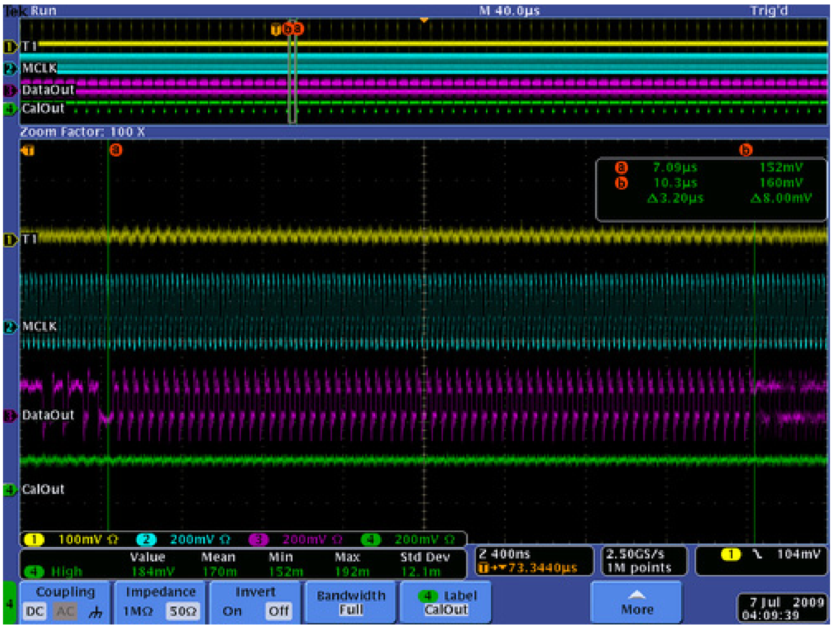


| | a | b | Δ |
|---|---|---|---|
| time | 7.09µs | 10.3µs | 3.20µs |
| voltage | 144mV | 144mV | 0.00 V |

| 4 High | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| | 176mV | 170m | 152m | 192m | 11.2m |

Z 400ns
T →▼73.3440µs
2.50GS/s
1M points
1 ↘ 104mV
7 Jul 2009
04:09:15

Figure 4-20: CalPulse applied with all channels turned off

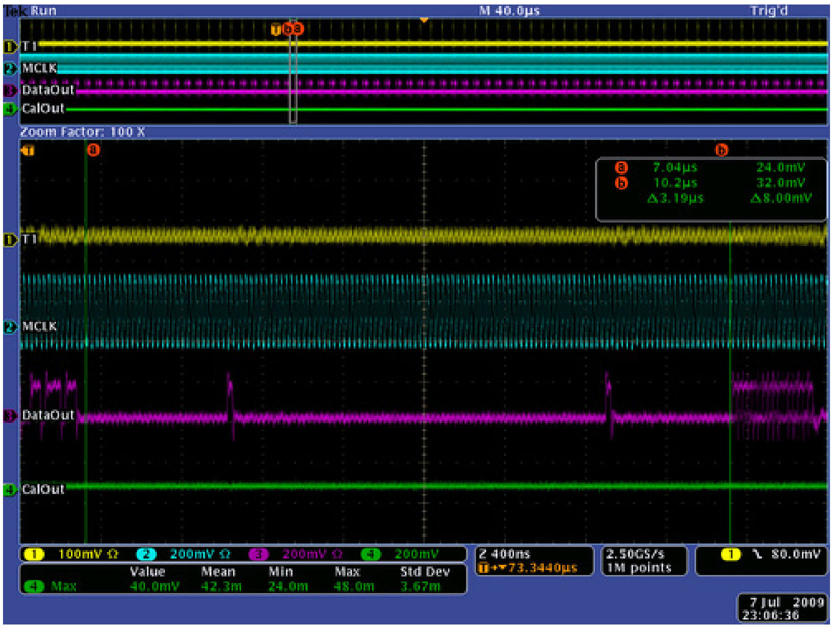


| | a | b | Δ |
|---|---|---|---|
| time | 7.09µs | 10.3µs | 3.20µs |
| voltage | 152mV | 160mV | 8.00mV |

| 4 High | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| | 184mV | 170m | 152m | 192m | 12.1m |

Z 400ns
T →▼73.3440µs
2.50GS/s
1M points
1 ↘ 104mV
7 Jul 2009
04:09:39

Figure 4-21: CalPulse applied with all odd channels turned on



| | Value | Mean | Min | Max | Std Dev |
|---|---|---|---|---|---|
| 4 Max | 40.0mV | 42.3m | 24.0m | 48.0m | 3.67m |

**Figure 4-22: CalPulse applied with all odd channels turned on**

# Chapter 5

# Results and Conclusions

In this thesis we discussed that the basic purpose of the Region 1 detector for the Qweak experiment at Jefferson Lab is to provide precise cylindrical positioning information of scattered particles. We have also discussed the basic theory behind its operation in that we have shown how the GEM detector uses ionizing radiation to initiate an avalanche of charged particles large enough to be detected by the VFAT IC. This occurs via a series of microscopic perforations in copper-Kapton-copper layers under a static potential difference. This amplifies the number of charged particles in a controlled manner with reference to the spacial resolution of the detector. We have then shown how this data is serialized and transmitted to a customized FPGA module. The data are then stored in a FIFO and subsequently transmitted to a Linux host computer via a VME controller.

There are two topics that of great importance that have yet to be discussed here. The first question is how well the overall detector is working and the second is how fast can data be transferred in real-time without error. We address the first question by way of using a statistical method known as S-Curves. The second question we address using a technique known as the deadtime measurement. Each of these techniques is discussed below.

## 5.1 S-Curves

To measure the noise level on the Region 1 detector, we used S-Curves. This measurement technique is performed by taking measurements while the voltage across the first stage of the GEM detector  is set at equilibrium. That is to say that there is no acceleration of the initial ionizing particles in the detector and therefore no amplification. Any noise that results in a hit or miss signal of any particular channel in a VFAT is therefore a result of the noise environment within the Region 1 detector set up for detecting cosmic particles. The shape of this noise should result in an "S"-like drop-off of the average number of at least one hits per event on each respective VFAT as the discriminator voltage, VThreshold1, is slowly increased. The setup for this experiment is demonstrated in the configuration shown in Figure 1-6 on page 11.

### 5.1.1 GEM Detector Disabled

The following graphs in Figure 5-1 - Figure 5-6 demonstrate the results of our experiments with the first amplification stage of the GEM detector disabled. Having no preliminary amplification means that any hits recorded by the VFATs are essentially noise. The triggering mechanism for this experiment was cosmic particles passing through two scintillators placed in coincidence and aligned vertically with the GEM detector. For all intents and purposes, the elaborate setup of having two scintillators in coincidence could have just as easily been replaced by a random trigger generator with a Poisson distribution as it is expected that the noise is mostly due to the environment of the laboratory in which these experiments were conducted and not the cosmic particles. Here "hits" are defined as whenever at least one channel on a given VFAT records a voltage greater than its respective threshold voltage. Therefore, it is only possible to have a hit value of '0' or '1' per coincidence. The average number of hits per coincidence of cosmic particles is the total number of hits divided by the total number of cosmic particle coincidences.

Due to time-consuming nature of taking these measurements, these values are single-run averages with an assumed error that this is Poisson distributed due to the assumed nature of the noise. These graphs all show a fairly good S-Curve shape indicating that we can indeed find the noise distribution of the individual VFAT chips to a fairly good approximation. There are, however, ostensible measurements that do not seem to fit this expectation. It is our prediction that taking further measurements and applying further error analysis would resolve these apparent discrepancies. Such an analysis is outside of the main scope of this thesis but is strongly recommend in the actual environment in which the Qweak experiment is to take place.

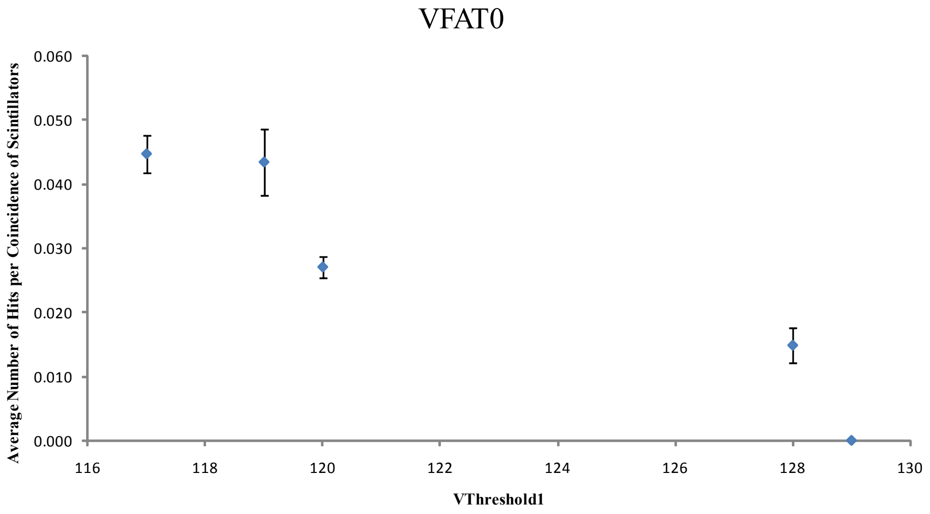


**Figure 5-1: VFAT0 with GEM Detector voltage disabled**

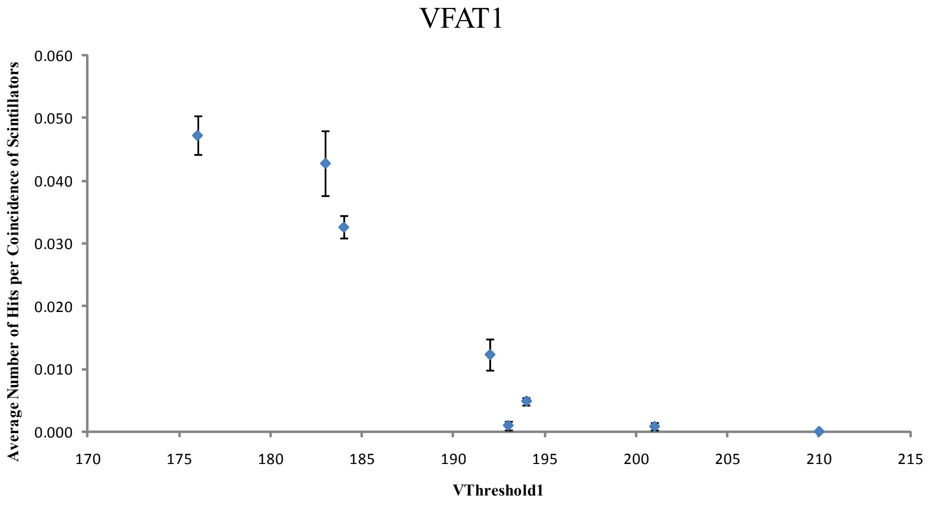


**Figure 5-2: VFAT1 with GEM Detector voltage disabled**

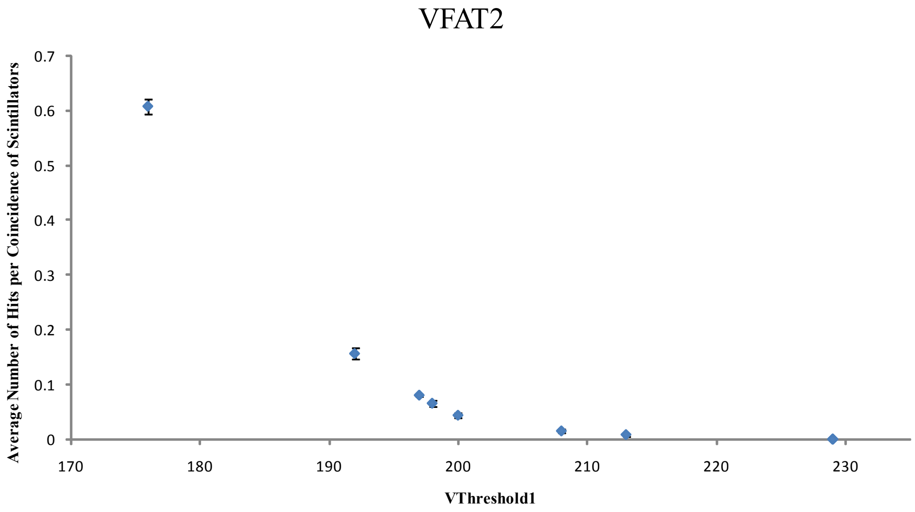


**Figure 5-3: VFAT2 with GEM Detector voltage disabled**

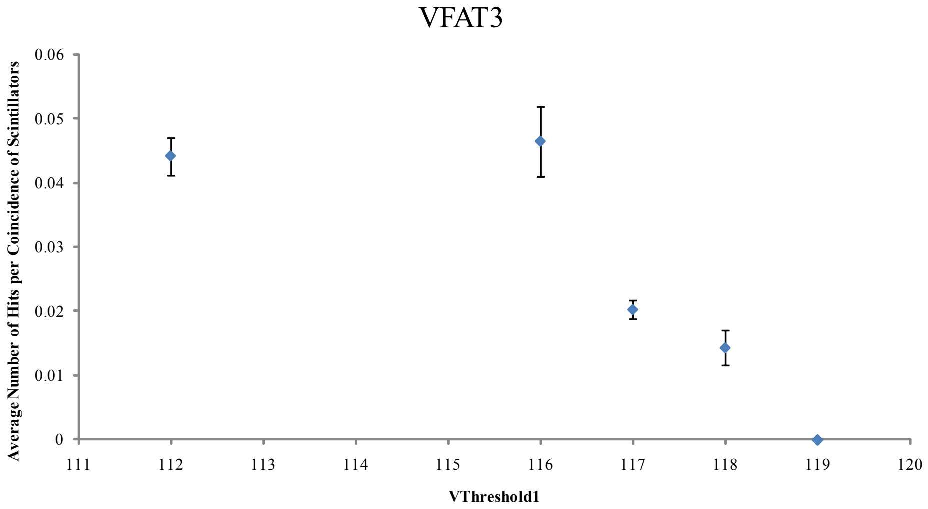


**Figure 5-4: VFAT3 with GEM Detector voltage disabled**

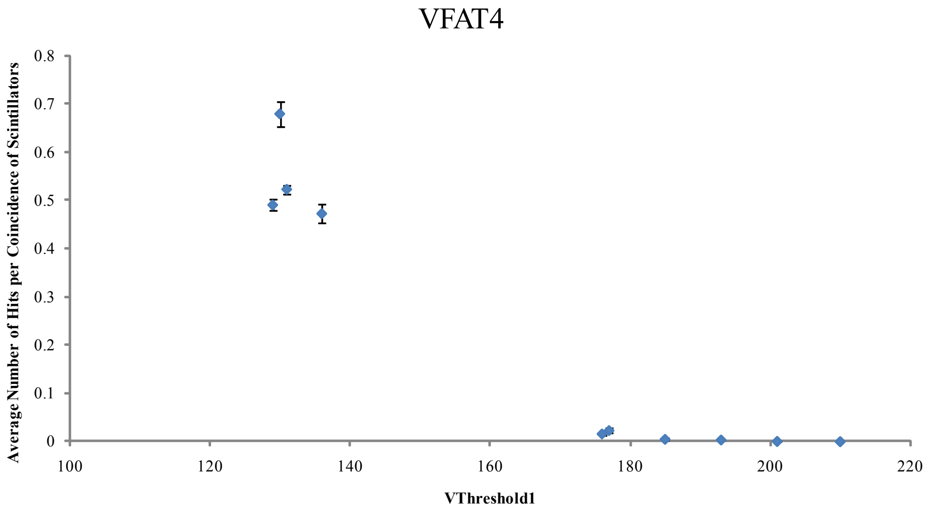


**Figure 5-5: VFAT4 with GEM Detector voltage disabled**

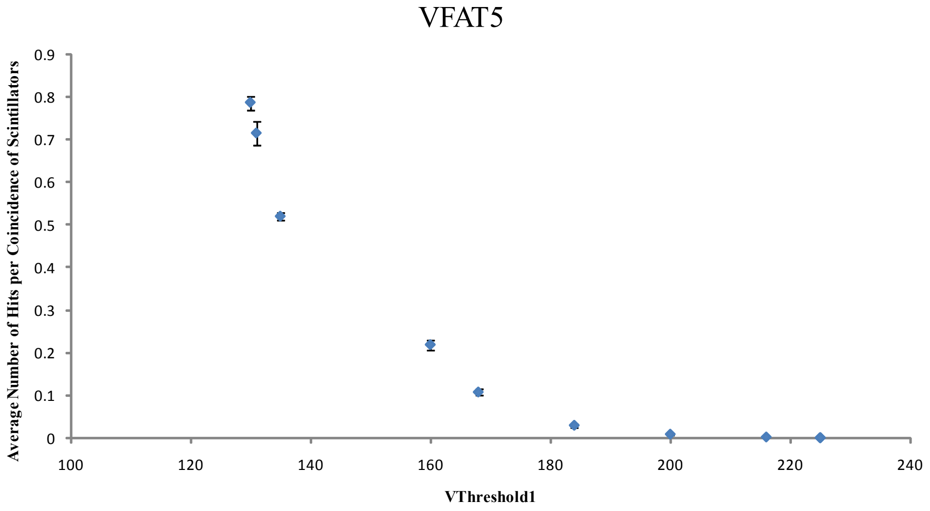


**Figure 5-6: VFAT5 with GEM Detector voltage disabled**

The recommended threshold values for VThreshold1 are set at the lowest value of VThreshold1 for which no hits were ever recorded for the entire run. These values are shown in Table 5-1. This is a fairly extreme value and may affect the ability of the VFAT detectors to detect true particles. If it is found that this value is too high to detect a sufficient number of true particles, this threshold may need to be decreased and a more statistical approach to finding true hits may need to be employed.

| VFAT Number | Recommend VTheshold1 Setting in Hexidecimal (Decimal) |
|---|---|
| VFAT0 | 0x81 (129) |
| VFAT1 | 0xD2 (210) |
| VFAT2 | 0xE5 (229) |
| VFAT3 | 0x77 (119) |
| VFAT4 | 0xD2 (210) |
| VFAT5 | 0xE1 (225) |

**Table 5-1: Recommend VThreshold1 Setting Per VFAT**

## 5.2 Maximum Data Throughput

The maximum rate at which the data may be transferred from the VFATs through the V1495 and recorded into a DAQ computer is dictated by numerous variables. However, we were able to measure the maximum rate by performing the following experiment:

With all of the VFAT ICs turned on and using a pulse generator, we produced dual Trigger pulses; the first is sent to the V1495 – generating an LV1A signal in the VFAT – and the second is a delayed copy that is sent to the SIS3600 module that generates an interrupt on the VME backplane thereby signaling the ROC to initiate a read of the V1495 data. Of course, the second Trigger pulse must be delayed enough for the V1495 to request the appropriate data packet, transmit this data, and store the data in the FIFOs on the V1495.

In order to determine if any LV1A requests were not fulfilled, we simply set the SIS3600 to signal a low value on one of its outputs when the system, e.g. the ROC, was busy and then return high again once the system was ready to receive another request. Due to jitter in the timing of the busy signal, the length of time that the system is busy is not always the same. There were two main parameters which we could affect in this experiment; the time delay between the two copies of the Trigger signal between the V1495 and the SIS3600 and the repetition rate of the requests for data transfer (LV1A).

It was found that we could not request transfers of data between the V1495 and the ROC any faster than 45ns as this was the time period found on the scope from when an LV1A request was sent to the V1495 and the busy signal from the SIS3600 would first go low. It is difficult to determine this number precisely as there are literally dozens of configurable systems and subsystems involved in the V1495 alone that dictate this time period.

We then set up a scalar (e.g. counter) that would count the number of requests for data transfers and the number of times that the SIS3600 would signify that the system was busy. At the end of a run using CODA, we could then just determine if the two values recorded by the scalar were indeed the same or not. If the number of LV1A requests was greater than the number of SIS3600 busy signal, then we knew that we were ignoring readout requests.

The following table, Table 5-2, below shows the number of times an LV1A request was sent and the subsequent number of busy signals from the SIS3600. The percentage of missed busy signals to LV1A requests – also known as the deadtime– is shown in the following tables. This experiment was repeated twice for verification purposes and a the statistical error is shown when available.

| Trigger Rate | Deadtime Percentage |
|---|---|
| 4 kHz | 0.00 |
| 5 kHz | 0.00 |
| 7 kHz | 2.05 ± 0.106 |
| 8 kHz | 5.61 ± 0.332 |
| 10 kHz | 50.00 |
| 12 kHz | 50.26 ± 0.367 |
| 14 kHz | 51.09 ± 0.049 |

**Table 5-2: Percentage Deadtime vs. Trigger Rate**

# Bibliography

[1] Aspell, Paul, *VFAT2 - Operating Manual*, July 2006

[2] Aspell, Paul, et al., *VFAT2: A front-end system on chip*, CERN,

[3] Aspell, Paul, et al., *The VFAT Production Test Platform for the TOTEM Experiment*, CERN

[4] Technical Information Manual, *MOD. V1495*, Feb. 27, 2009

[5] Cyclone Device Handbook, *Volume 1*

[6] DataSheet, *MVME6100 Series, VME Single-Board Computer*

[7] MVME6100 Single-Board Computer, *Programmer's Reference Guide*

[8] Franco, Sergio, *Design with Operational Amplifiers and Analog Integrated Circuits*, Third Edition, 2002

[9] Paul, Clayton, *Introduction to Electromagnetic Compatibility*, Second Edition, 2006

[10] H. W. Ott, Ground -- a path for current to flow, *Proc. 1979 IEEE Int. Symp. Electromagnetic Compatibility*, San Diego, CA, Oct. 1979

[11] H. W. Ott, Partitioning and layout of a mixed-signal PCB, *Printed Circuit Board Design*, 8-11, 2001

[12] Particle Data Group, *Particle Physics Booklet*, July 2008

[13] Streetman, Ben, Banerjee, Sanjay, *Solid State Electronic Devices*, Fifth Edition

[14] Behzad, Razavi, *Design of Analog CMOS Integrated Circuits*

[15] https://coda.jlab.org/wiki/index.php/Main_Page