# THE DATA ACQUISITION AND CALIBRATION OF LUMINOSITY AT THE COMPACT MUON SOLENOID EXPERIMENT

By

Jessica H. Tang

Honors Thesis
Submitted to the Faculty of the
Vanderbilt University Department of Physics and Astronomy
in partial fulfillment of the requirements
for

DEPARTMENTAL HONORS
in

PHYSICS

Nashville, Tennessee

Thesis Committee:

Professor Alfredo Gurrola, Ph.D.

Professor Thomas Kephart, Ph.D.

Professor Paul Sheldon, Ph.D.

To my family, friends, and colleagues.

# ACKNOWLEDGMENTS

Thanks to all who have been here for the last four years. In particular, thanks to Dr. Will Johns, Dr. Francesco Romeo, Dr. Brenda Fabela Enriquez, and Dr. Daniel Li for all the guidance, wisdom, patience and kindness. Special thanks to the CMS BRIL collaboration, everyone involved in online calibration, the PLT working group, and CMS TriDAS for their help in this project, as well as USCMS for the funding of this work. Thanks to all my professors at Vanderbilt, the Department of Computer Science, the Department of Mathematics, and most of all, thanks to the Department of Physics and Astronomy and the Vanderbilt CMS group. Finally, I would like to thank all my friends here at Vanderbilt for being my second family and to my loved ones for their support and encouragement.

# TABLE OF CONTENTS

Page

# LIST OF FIGURES

# CHAPTER 1

# Introduction

The Pixel Luminosity Telescope (PLT) is a dedicated silicon pixel luminosity monitor (luminometer) at the CMS Experiment at the CERN LHC. It was installed at the beginning of Run 2 in 2015 and will be in operation until the end of Run 3 when the Long Shutdown 3 (LS3) begins. The determination of luminosity is used for optimization of accelerator performance at the LHC and is used as a parameter to any analysis of physics processes. This work discusses various methods to reduce statistical and systematic uncertainties in online and offline luminosity measurements. Reducing these uncertainties will improve our ability to constrain or discover new physics beyond the Standard Model (BSM) and make precision Standard Model measurements at the CMS experiment.

Chapter 1 will discuss the LHC and the CMS experiment. Chapter 2 will discuss the luminosity measurement at CMS and the van der Meer (VdM) scan methology and software used for the analysis of these scans. Chapter 3 will discuss the PLT and the upgrade of data acquisition system (DAQ) as well as the implementation of this system.

## 1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is an accelerator complex operated by the European Organization for Nuclear Research (CERN) whose primary purpose is to collide beams of protons or heavy ions at a center-of-mass energy of up to $\sqrt{s} = 14$ TeV. These beams are made to collide at four points around the LHC ring, where physicists study the interactions between elementary particles. At these four points sit four detectors: the two general-purpose experiments ATLAS and CMS, and two specialized experiments LHCb and ALICE for studying B physics and heavy ions respectively. Beams at the LHC are arranged into bunches of around $10^{11}$ protons with 3564 bunch crossings (BXs) per orbit. An orbit is defined as the time it takes for one bunch to circulate the entire LHC ring. The revolution frequency at the LHC $f_{rev}$ is 11.245 kHz, which corresponds to a bunch crossing frequency of 40 MHz [5].

Some of the physics goals at the LHC include a better understanding of the mechanism behind electroweak symmetry breaking by either the discovery or the exclusion of the Higgs Boson, the discovery of BSM physics, precise determination of properties of the Standard Model, an investigation of the parameters of CP violation, and the exploration of the quark-gluon plasma [16]. In 2012, the Standard Model Higgs was discovered jointly by the CMS and ATLAS experiments at a mass of 125 GeV.

## 1.2 The Compact Muon Solenoid Experiment

The Compact Muon Solenoid (CMS) Experiment sits at LHC Point 5 and is located near the Swiss border in Cessy, France. The underground experimental cavern houses the detector approximately 100 meters below the surface, while all the back-end electronics sit in the underground service cavern. The CMS detector is built in concentric layers around the interaction point (IP) at which the beams meet, with each layer helping to measure the energy and momentum of particles that are produced as a result of the collisions and further decays. The inner part of the detector consists of a 3.8 T superconducting solenoid 6 meters in diameter surrounded by a muon system embedded in the steel return yoke. Within the solenoid volume are a fully silicon tracker made up of a pixel detector surrounded by a strip tracker, an electromagnetic calorimeter (ECAL) composed of lead tungstate crystals, and a brass and scintillator hadronic calorimeter (HCAL) [6].

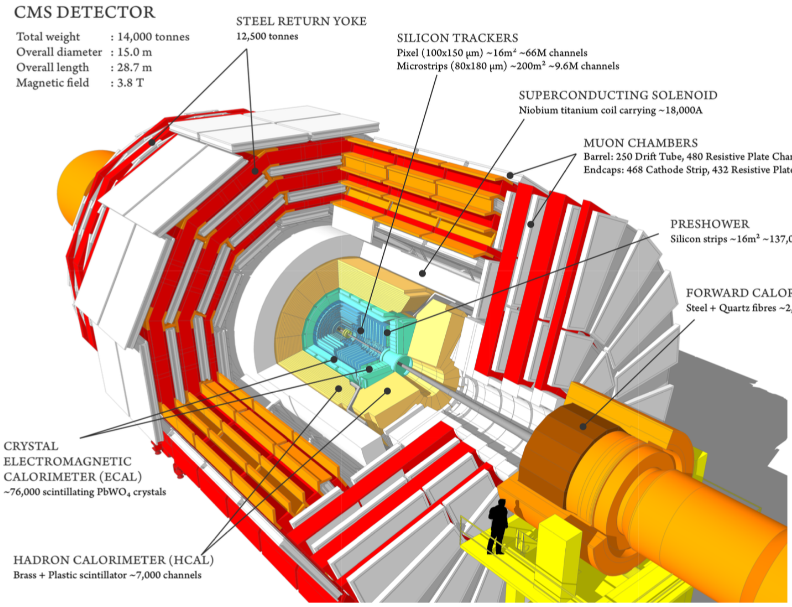


Figure 1.1: The CMS Detector and its subsystems

### 1.2.1 The CMS Data Acquisition System

The CMS subsystems are designed to capture signals that indicate the presence of particles produced as a result of the collisions of the beams, however these signals need to be reconstructed and assembled into events that can be analyzed offline. The hardware-based Level 1 Trigger(L1T) captures the most interesting fraction of these events, reducing event rate from the full bunch crossing frequency of 40 MHz to around 100 kHz. The CMS Data Acquisition (DAQ) System then reads out and assembles the events, and sends events to the software-based High-Level Trigger (HLT) for further filtering of events. The readout of the front end electronics is done using a VME module called the Front-End Driver (FED) which digitizes and builds event fragments and send them to the DAQ system via a dedicated readout link. In the Run 1 DAQ system (DAQ-1), the FEDs were read out using the SLINK-64 standard [13] based on copper links at a rate of up to 400 MB/s and forwarded to the Myrinet network. During Run 2, CMS was expected to run in higher pile-up scenarios with larger event sizes, so DAQ-2 (Figure 1.2) features a new optical link based SLINK-express standard that supports the new event size and allows for re-transmission. From there, the events are sent to the Front-End Readout Optical Link (FEROL), where TCP/IP protocol is implemented for speeds up to 10 Gb/s. A more detailed description of the CMS Event Builder network can be found in Ref. [3].

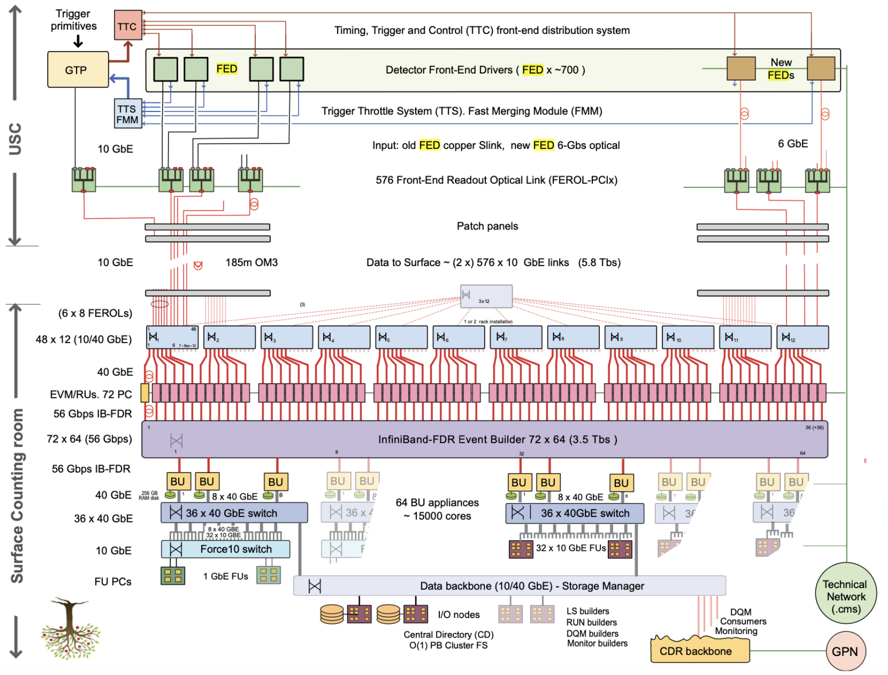


Figure 1.2: The DAQ-2 Readout System

# CHAPTER 2

## Luminosity

At each interaction point, the beams are concentrated into as small of an area as possible before they collide in order to ensure as many particles are colliding as possible at any given moment. This is particularly important for the physics goals of the LHC as more collisions increases the chances of observing a rare process. Luminosity is a measure of the number of effective collisions occurring within a machine at a given moment. For a given process with a cross section $\sigma_p$, we have the following relationship:

$$\frac{dR}{dt} = \mathscr{L} \cdot \sigma_p \tag{2.1}$$

where $\frac{dR}{dt}$ is the number of interactions per second and $\mathscr{L}$ is the luminosity. The value of instantaneous luminosity is reported to the LHC to measure the machine's performance and to optimize data-taking conditions for CMS.

Luminosity is also integrated over a time period of interest to determine the interactions occurring during that time. Integrated luminosity is thus the proportionality constant that is multiplied by the cross section of a process to give the interactions of interest occurring during the time integrated over [9]. In many physics analyses occurring at the LHC, this means that integrated luminosity is a necessary parameter for comparing the observed number of interactions with theoretical predictions and simulations. In many cases, the luminosity is one of the largest contributors to the uncertainty in the channels being studied, which is why the precision determination of luminosity is so important.

### 2.1 Luminosity Measurement at CMS

At CMS, one of the primary responsibilities of the Beam Radiation, Instrumentation and Luminosity (BRIL) Group is the determination of luminosity. BRIL operates several dedicated luminometers and uses several CMS subsystems to deliver online and offline luminosity values. These include the Pixel Luminosity Telescope (PLT), the Fast Beam Conditions Monitor (BCM1F), the Hadronic Forward Calorimeter (HF) with a method based on occupancy counting (HFOC) and a method based on the transerse energy sum (HFET), and Pixel Cluster Counting (PCC) with the pixel detector. Many of these subsystems are designed to remain in operation regardless of the status of CMS and the central Data Acquisition System (DAQ) ans such that it is not subject to the constraints of the trigger bandwidth of the central system.

Each luminometer determines some rate by counting tracks, energy, hits or some other quantity measured in the detectors. This rate is related to the instantaneous luminosity at any given time via the relationship:

$$R_{luminometer} = \mathscr{L}_{inst}\sigma_{vis} \tag{2.2}$$

where $\sigma_{vis}$, also known as the visible cross section, is a calibration constant determined for each detector, $\mathscr{L}_{inst}$ is the instantaneous luminosity and $R_{luminometer}$ is the luminometer rate [7]. This calibration constant is determined using the van der Meer (VdM) scan method.

## 2.2 The VdM Scan Method

To determine $\sigma_{vis}$, the following relationship is used:

$$\sigma_{vis} = \frac{2\pi\Sigma_x\Sigma_y R(x_0, y_0)}{N_1 N_2 f_{rev}} \tag{2.3}$$

where $\Sigma_x = \frac{1}{2\pi}\frac{\int R_x(\Delta)d\Delta}{R_x(0)}$ is defined to be the beam overlap width where $R_x(\Delta)$ is the rate at a beam separation of $\Delta$ in the x direction. $N_1$ and $N_2$ are the number of protons or ions in the two colliding bunches. Experimentally, this value is determined by sweeping two beams across each other and measuring the rate at each separation step. This scan is done in both the x and y directions and $\Sigma_x$ and $\Sigma_y$ are determined by fitting the scan curves [8]. During VdM fills at the LHC, most of the BXs are empty

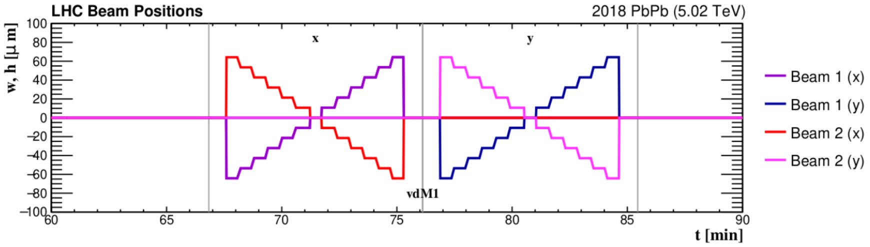


Figure 2.1: Horizontal (x) and vertical (y) beam displacement measurements provided by LHC magnet currents during the vdM PbPb scan program in 2018 at CMS.

More details on the determination of this relationship and the procedure can be found in Ref. [7].

## 2.3 The VdM Framework

The primary software tool used to analyze the van der Meer scans by BRIL is called the VdM Framework. The framework is implemented in Python 3 and takes in data from BRILDAQ stored as Hierarchical Data Format (HDF5/H5) files. Files contain the luminometer rates, beam currents $N_1$ and $N_2$, and beam separations. Each file contains the data for one fill, and the framework is capable of analyzing one fill at a time. The framework takes in a file name, correction flags, fitting functions, and flags indicating the luminometers for which results should be produced. The files are first read in, then the framework produces intermediate files for scan info, beam currents, rates and corrections. The rates are then corrected, a function is fitted to the rates (usually a Gaussian or some form of a Gaussian is used), and finally $\sigma_{vis}$ is computed[14].

The framework is designed to be used in online and offline analysis of the VdM scans, the latter case of which frequently is done for reprocessing of data and requires the framework be run with many different corrections on a large number of files. Many improvements have been made to the VdM Framework over the course of Run 3 in order to speed up the analysis process for a large number of scan files. The framework can take anywhere from a couple of minutes to hours to run for all of the desired fills. In the cases where multiple luminometers, corrections, or fitting functions are used in one run, the framework is capable of running in multiple threads on parallel processors to speed up the analysis. Since analysis is usually done on one machine only, the design of the framework requires that the user run one analysis for one fill with all of the required flags at a time, wait for that analysis to finish, and then run the framework again for the next file. For a large number of files, this is time consuming.

### 2.3.1 HT Condor

HT Condor is a High Throughput Computing (HTC) batch management system that allows users to utilize many idle computing resources at once in parallel. The user creates the jobs, requests a certain number of resources (namely CPUs and computing time), and submits them to Condor. From there, Condor handles the allocation of resources from the computing cluster and matches jobs to machines on the cluster.

A new tool was added to the VdMFramework that allowed many instances of it to run on the bigbird and thunderbird clusters at CERN. This tool was implemented in Python 3 and allows users to submit multiple jobs at once, specifying the flags, fills as a range or list, output directory, data directory, and version of the framework to be run through a YAML configuration file. The tool then automatically generates the jobs to be submitted. This tool also ensures that the analysis, particularly those that need to be run in parallel, are run on the computing clusters at CERN with available resources.

### 2.3.2 Performance

By further parallelizing the framework over multiple fills, the runtime gains a factor of N, where N is the number of jobs submitted. For some directories, the framework may need to be run for hundreds of different fills.

For example, the framework takes around 5 minutes to run for PLT rates only. With 130 files to analyze, this could take up to 10 hours to run, but using Condor this takes around 10 minutes. In another particular case, the framework needed to be run on only a couple of files with many flags and corrections and for all

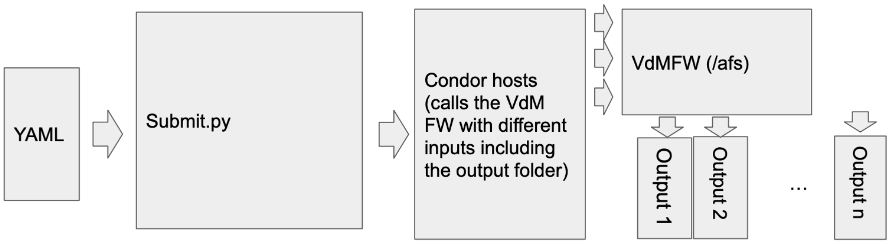
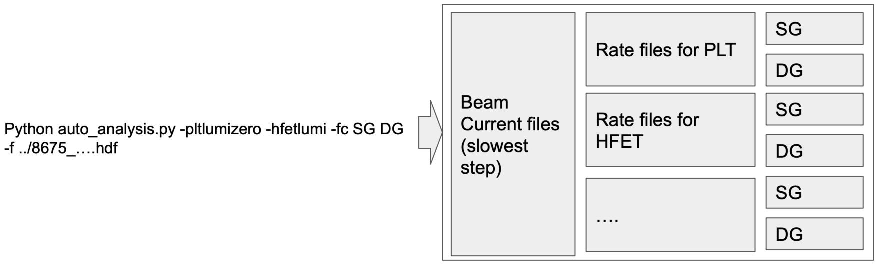


Figure 2.2: The above diagram shows how the framework is run on HT Condor in the special case of FOM Analysis. For other analyses, there is only one output folder.

luminometers. Here, the runtime was reduced from 2 days to a couple of hours.

In the particular case of PLT nonlinearity studies, the framework runs with different corrections, also known as Figure of Merit (FOM). These outputs need to be specified in different output folders, so the framework was additionally modified to take the output folders as an input argument. In this case, the framework is not being parallelized over fills, but rather over a range of corrections. HT Condor should be able to be used to run virtually any parallelizable task relating to the framework.

In practice, further parallelization is not necessarily more effective. For example, the structure of the framework and the analysis steps makes it difficult to effectively run the framework in parallel for different luminometers.



```
Python auto_analysis.py -pltlumizero -hfetlumi -fc SG DG
-f ../8675_....hdf
```

Figure 2.3: The framework is run with various luminometers for a single and double gaussian fit.

In this scenario, the beam files are general to all luminometers and take the longest. After this, rate files for specific luminometers and fits can be done in multi-threaded mode. Each Condor host would have to generate its own beam current, and the quicker steps that are already done in parallel in different threads would run on separate machines.

# CHAPTER 3

## The Pixel Luminosity Telescope (PLT)

### 3.1 Overview

The Pixel Luminosity Telescope (PLT) consists of 48 pixel sensors arranged into 16 telescopes pointing at the CMS IP. Each telescope consists of three sensors measuring 8x8 mm$^2$ that are arranged perpendicular to the beam line 172.5 cm from the interaction point outside of the pixel endcaps. Eight telescopes are placed on each side of the interaction point, which are referred to as the +z and -z sides.

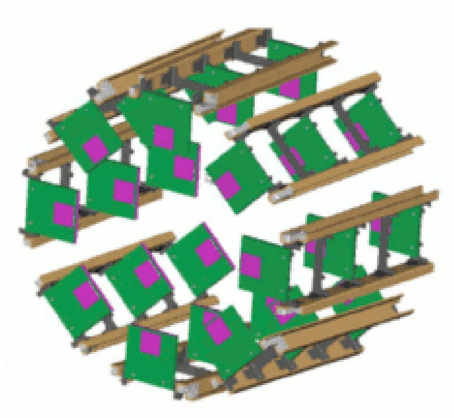


Figure 3.1: The layout of one half of the PLT detector on one side of the interaction point with 8 telescopes.

The PLT itself is based on the Phase-0 Pixel detector that was the original detector installed in 2009 with the same sensors [1] and PSI46v2 readout chips (ROCs) bump bonded to the sensors [10]. Each telescope has 3 ROCs managed by a Token Bit Manager (TBM) chip that distributes clock and trigger signals and coordinates the readout. More details on the front-end hardware can be found in [8].

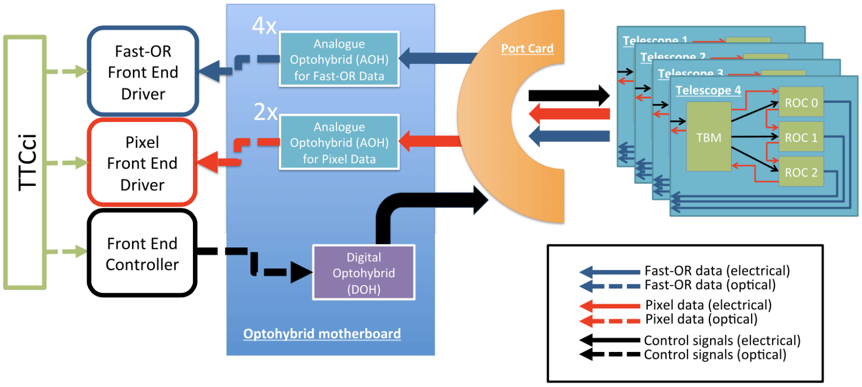


Figure 3.2: The control and readout systems of the PLT.

### 3.1.1 Readout Modes and Luminosity Measurement

The PLT has two readout modes: the fast-OR readout and the full pixel data readout. In the fast-or readout, the number of ”triple-coincidences”, where three hits during a BX on all three planes of a telescope are registered, are counted. This data does not contain any information on the location or pulse height of the hits, but are read out at a full bunch crossing frequency of 40 MHz and provide the main online luminosity measurement. The full pixel readout, also sometimes referred to as the PLT Slink readout, provides the more detailed information on each hit but is read out when a trigger is received at a much slower rate.

The rate measured by the PLT is the number of mean triple coincidences originating from the IP. This rate is estimated using the zero-counting method where we assume that triple coincidences will follow a Poisson distribution with a mean $\mu$. The fraction of hits where no triple coincidence is registered, $f_0$, can be used to determine the number of triple coincidences using the relationship

$$\mu = -ln(f_0) \tag{3.1}$$

This rate used to determine luminosity along with the calibration constant $\sigma_{vis}$ in the procedure discussed in Chapter 2. With the fast-or readout, a source of nonlinearity in this relationship comes from the registering of a triple coincidence produced by a track or multiple tracks that do not originate from the interaction point. This is often corrected for using the full track information that comes from the pixel readout [8].

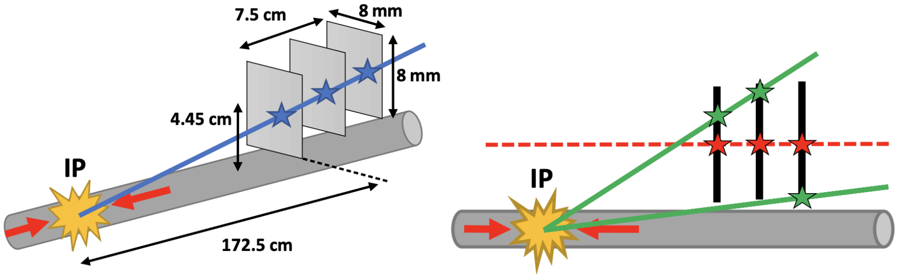


Figure 3.3: Left: A triple coincidence originating from the IP measured by the PLT. Right: Accidentals that might be recorded as a triple coincidence. In green, two independent tracks generate a triple coincidence. In red, a triple coincidence is produced from a source other than the IP.

The readout of the data itself is done using the same Front End Drivers (FEDs) as the pixel detector. Two of the FEDs are used for the readout of the fast-or data, and this data is then published to BRILDAQ via DIP protocol to report the online luminosity value.

Another FED, known as the TTC FED, receives information from the CMS Timing Control Distribution System (TCDS). These signals include the central clock and orbit signals, current fill number, run number, luminosity section (lumi-section) number, luminosity nibble (lumi-nibble) number. The lumi-sections and lumi-nibbles are further divisions of the runs, with each lumi-section corresponding to around 23 seconds of running and the lumi-section corresponding to the time it takes for the LHC to complete 4096 orbits. These signals are received via fiber connection from the central system by the TTC FED and then distributed and used by the PLT system. The last FED is referred to as the Pixel FED.

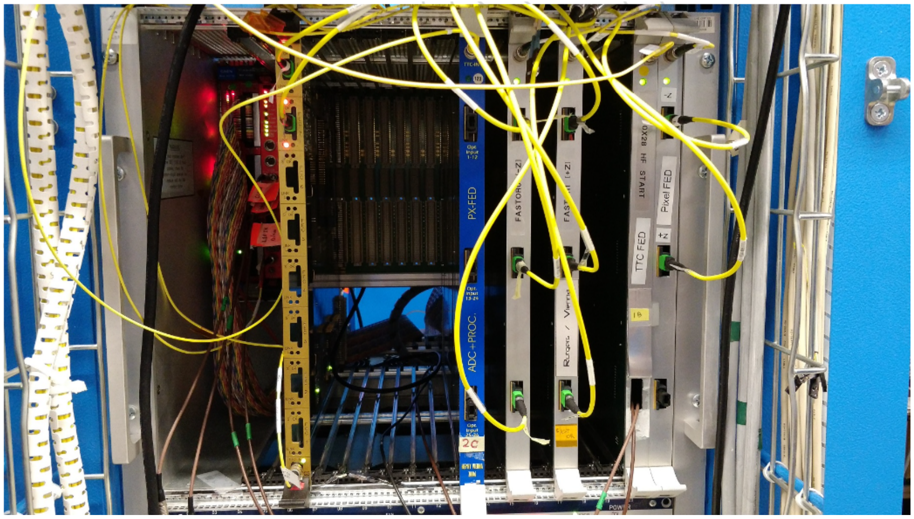


Figure 3.4: The PLT VME Crate: The four boards on the right are the two fast-or FEDs, the TTC FED, and the Pixel FED.

### 3.2 Full Pixel Readout

The other FED is used to decode and digitize the full pixel data from the ROCs and this data is sent over an SLINK-64 data link (discussed in Section 1.2.1) to a PC and saved to disk. Some quantities, such as the rate of accidentals are also published via BRILDAQ. This readout link is the same one from the Phase-0 and is much older and slower than most of the readouts from the Run 2 and 3. As a result, the trigger rate is limited to a couple of kHz in order to prevent data overflow.

This data can also be used to provide a luminosity measurement, however the limited trigger rate means this method cannot be used as an online measurement because of its poor statistical precision. However, this data sees a smaller contribution from accidentals since the pixel data provides location of the hits and can be corrected for. Since the PLT sits in the very forward regions of the detector, tracks registered are nearly parallel to the magnetic field and generally experience very little bending due to the field, so we can assume the tracks are straight lines. This method constructs track data from the pixel data, then the tracks are fitted to a slope and the x and y residual values are recorded. These values are then aggregated and form a Gaussian distribution with a mean of $\mu$ and standard deviation of $\sigma$. Candidates are then rejected as accidentals using a selection of $2\sigma$ for either the slope values or residual values [8].

Since most bunches are empty during a VdM scan, a higher-rate trigger is used during these fills to trigger primarily on the colliding bunches. The track data for these fills can similarly be analyzed for $\sigma_{vis}$ with a selection of $5\sigma$ instead.

### 3.2.1 Motivation for Upgrade

The statistical precision of the full pixel data is largely limited by the back end hardware, specifically the latency of the data readout link. During physics runs, the PLT SLink data link and PC save approximately 1 GB of data per hour. The readout is a legacy system that has not been used in the rest of CMS for decade, with the PC being from 2005. This poses a particular challenge to maintain for the BRIL project as many of the services on this machine are far outdated and no longer supported by central CMS IT services.

Attempts have been repeatedly made to upgrade this readout and the PC throughout the PLT's operational history. The BRIL project attempted unsuccessfully to implement the SLink interface with a newer XPCI interface. The throttling trigger discussed in Section 3.2.1 was implemented as a means to mitigate and prevent data overflow. However, the continued limiting of the trigger rate was preferred to this implementation since it required further maintenance and development.

In 2022 to 2023, steps to implement a USB 3.1 interface began at Vanderbilt with the hope of being able to achieve speeds of up to 5 Gbps. This implementation was done on Cyclone V FPGA and was purchased as an out of box solution, but had little documentation or support from the vendor. Additionally, since it was purchased from an outside source, this solution also needed to be modified to interface with the SLink. In the end, this solution was scrapped.

During the Summer of 2023, funding was received from USCMS to upgrade the readout to use the SLink-express and Front End Readout Optical Link (FEROL) used in DAQ-2 and discussed in Section 1.2.1. This new system was obtained through discussion and collaboration with the CMS DAQ group. Work to set up a test stand for this new system began at CERN in the Tracker Integration Facility (TIF) at the Meyrin site. Decommissioned Phase-0 Pixel FEDs and SLink mezzanine cards were located in the CERN Prevessin site and brought back to the TIF to be used in the new setup. These cards are identical to the ones used for the PLT currently in operation.

The FEDs, SLink boards and the FEROL were shipped back to Vanderbilt to continue implementation in an identical setup. Considerable time was spent understanding and examining the software package typically used for interfacing with and controlling the hardware. Issues with the firmware version were also identified and resolved with help from the DAQ group. The software was eventually modified to better suit the needs of the PLT system and streamlined for running. After this, the setup was preliminarily tested for performance and throughput by sending test data through the system. The details of the implementation and a description are discussed in depth in Section 3.2.2.

This system has potential to increase the bandwidth of the system by an order of 1000, thus allowing for an increased trigger rate and an improved statistical precision on the luminosity measurement from the full pixel data. Additionally, the FEROL and SLink-express are currently in commission and have the benefit of being supported by the CMS DAQ group. The PC must also necessarily be upgraded from the old PC in use currently in order to even run the software needed to start up and interface with the hardware.

### 3.2.2 Trigger Throttling

The Trigger Throttling System (TTS) [12] is implemented for the SLink interface to specify four signals: ready, busy, out-of-sync, and warn to indicate the status of the data link. Each signal is asserted respectively when the SLink is ready to receive more data, busy and should not receive more data, out of sync, or almost full. These signals are used to prevent the trigger from overwhelming the back-end electronics responsible for receiving, decoding and saving the detector signals as data and can be used to detect and mitigate a variety of issues. In the CMS DAQ system, the TTS is implemented as a part of the Timing, Trigger and Control System (TTC) [15].

In the PLT system, a simple throttling trigger was implemented using custom electronics and an Altera MAX II FPGA. The custom board takes in orbit and clock signals as NIM logic signals and the TTS signals as LVDS signals and converts them to Low Voltage TTL signals that can be used as pin inputs to the FPGA. The FPGA implements a simple logic that scales the LHC clock to trigger once per orbit on incrementing BXs such that on the first orbit, the firmware will trigger on the first bunch crossing, and then the next orbit it will trigger on the second, and so on. This trigger output is converted back to NIM and is used as an input signal to the TTCci, which is the VME interface module in the TTC. The firmware also vetoes the trigger whenever a signal indicating that the SLink is busy is asserted using the TTS and resumes incrementing through the BXs once the SLink is able to take more data.

An alternate implementation takes in a list of bunch crossings as well as the orbit signal, and cycles only through these instead of all 3564 BXs. This implementation is particularly useful for the VdM scan fills discussed in Chapter 2 that only have a couple of colliding bunches out of the 3564. This implementation will also be more effective since the bandwidth of the new data link in theory should be able to handle more than one trigger per orbit.

### 3.2.3 Description of the New System

The FEROL supports two 6 Gbit/s optical links and two 10 Gbit/s optical links via Small Form-factor Pluggable transceivers. One of the 10 Gbit/s links implements a unidirectional TCP/IP protocol used for sending the data to the dedicated PC, where the data is received and saved to disk. The other three links implement the protocol are used to receive data from the SLink-express. The FEROL is implemented on an Altera Arria II GX 125 FPGA core. Implementation and the structural design of the firmware can be found in Ref. [2]. The TCP implementation ensures lossless data transfer and should also implement flow control in the case of data overflow. However, in order to utilize the full potential of this system, there must be a way for the trigger to slow such that the system will recover once the system asserts that overflow has occurred or is imminent, namely through the TTS.

The SLINK64 sits on an SLink mezzanine card and is replaced by the SLink-express card. An optical link connects the new SLink to the FEROL through one of the 6 Gbps SFPs. The FEROL sends the data received to a PC using the 10 Gbps optical link, after which it is saved and passed further down stream to BRILDAQ and used for offline analysis.

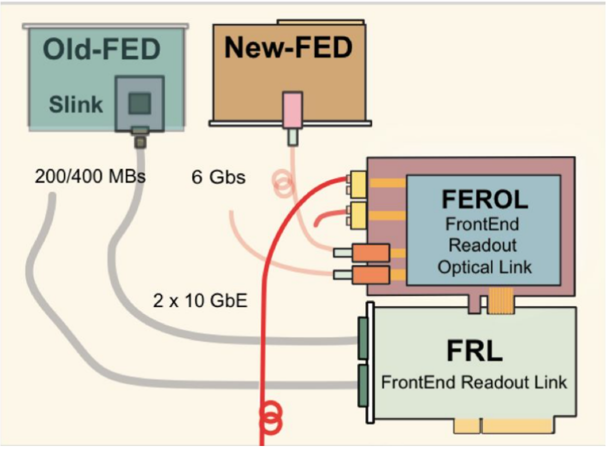


Figure 3.5: The diagram above illustrates the flow of data from the FED in the old system and new system. The FEROL sends the data to a dedicated PC using the red optical link.

In the central DAQ system, the reconstruction of events is done using the XDAQ software package Event Builder (EvB), which relies on the status of CMS. In addition, the FEROL must also be configured, enabled, or halted using the FerolController XDAQ application [4]. The software package supplied for the purposes of testing the FEROL utilizes these two XDAQ applications. The PC sends messages using the FerolController to first configure the FEROL and then to enable it to receive data, the latter of which will fail if the PC has not also opened up a socket using the receiver that is listening on the destination IP address. Once the connection is to be closed, the PC sends another message to the FerolController to halt. The applications in charge of receiving, reading out, managing, verifying and building events from multiple source FEDs are all a part of the Event Builder. In the case of the testing of one stream intended to be integrated into the central DAQ system, the test stream is managed and read out using these applications exactly as they would in the real system.

Since the PLT is designed to operate at all times and independently of CMS and its DAQ system, event reconstruction cannot be done using the CMS Event Builder. This application was stripped out of the software package and was replaced with a simpler application used by the DAQ group to debug the FEROL. In particular, a receiver based on Linux sockets is implemented in C and allows for the verification and reconstruction of event packets. The receiver implements a simple server interface where the FEROL acts as the client and sends data packets to the PC. The program was modified to handle the saving of raw data files to disk using the same SLink data format used in the old system. The data packets are written from a 1 MB buffer, where the data is verified by checking the CRC and event size.

### 3.2.4 Performance

The performance of the new system will not yet be known until it has been installed in data-taking conditions. The size of the data packets per trigger depends largely on the number of hits that are received per ROC. This number is expected to be relatively low per BX, but since the events follow a Poisson distribution, the probability of a large event size due to many hits is non-zero at any given time.

Regardless, there are a number of ways to simulate data-taking and test the upper limits on the maximum bandwidth of this system. Using the throttling trigger that implements one trigger per orbit, the FED can be simulated to register multiple hits per ROC, with approximately 2 kB sent per hit. The performance of this system scales linearly in the number of hits, with approximately 0.2 Gb or 25 MB of data sent per second per hit. Additionally, the trigger frequency can be increased to test the throughput for different hit rates. The trigger rate of the current system is limited to one trigger for every couple of orbits. With one hit per ROC and a trigger rate of around 200 kHz, the throughput reaches around 2 Gb/s, which is less than half of the capacity of the 6 Gb/s links. Increasing the trigger rate beyond this begins to violate many of the CMS trigger rules. Similarly, two hits per ROC and a trigger rate of 80 kHz reaches around 3 Gb/s. As discussed in Section 1.2.1, the central trigger rate of CMS that the detector operates with in the L1 Trigger is around 100 kHz.

### 3.2.5 Outlook

In Summer 2024, the further development and integration of the new SLink and FEROL into the PLT system are expected to occur with the goal of testing it within the PLT framework.

Data sent by the Pixel FED follows a 64-bit format, with 64-bit header and trailer words and 32-bit data words. To ensure an even number of 32-bit words such that it fills the 64-bit format, filler words are inserted in and must be stripped out. Data words are formatted with bits 31 to 26 indiciating the channel number, bits 25 to 21 indicating the ROC number, bits 20 to 16 indicating the double column number, bits 15 to 8 indicating the pixel number within the double column, and bits 7 to 0 indicating the pulse height[11]. The filler words are inserted in with ROC number 27. The current PLT SLink machine strips out the filler words and partially rewrites the SLink trailer word as a time stamp before these are dumped to a file as binary values. Once this file reaches a certain size, it is closed and a new one is opened for dumping. In addition to the raw data used for offline analysis, the pixel data is sent to BRILDAQ using a ZeroMQ publisher and analyzed for quantities of interest in the PLT SLink Processor.

The simplest way to integrate the new system into the PLT framework is to keep the data format identical and to add functionalities into the receiver to strip out filler words as they are read from the buffer and dumped to file and to publish this data to BRILDAQ. Another consideration is that the increased throughput and trigger rate means that the new system fills up the same 1 GB of disk space that would usually take an hour in the old system in a matter of seconds, even with a fraction of the throughput capacity. Even if everything else is kept the same, it is likely that handling this amount of data itself will present a variety of challenges.

Increasing the amount of data means that this new system may have the capability to provide improved corrections to the fast-OR data and an improvement in the uncertainties on the offline luminosity value reported using track data. The hope is that the increased statistics on this method may be enough that the fast-OR results can be reproduced and cross checked for further redundancy. For the offline analysis and luminosity value, the data will no longer need to be aggregated over the 5 minute period to achieve the level of precision needed.

# References

[1] Y. Allkofer, Claude Amsler, D. Bortoletto, V. Chiochia, L. Cremaldi, S. Cucciarelli, A. Dorokhov, C. Hoermann, R. Horisberger, D. Kim, M. Konecki, D. Kotlinski, Kirill Prokofiev, Christian Regenfus, T. Rohe, D.A. Sanders, S. Son, M. Swartz, and T. Speer. Design and performance of the silicon sensors for the CMS barrel pixel detector. COMPACT MUON SOLENOID. *Nucl. Instrum. Methods Phys. Res., A*, 584:25–41, 2008.

[2] G Bauer, T Bawej, U Behrens, J Branson, O Chaze, S Cittolin, J A Coarasa, G L Darlea, C Deldicque, M Dobson, A Dupont, S Erhan, D Gigi, F Glege, G Gomez-Ceballos, R Gomez-Reino, C Hartl, J Hegeman, A Holzner, L Masetti, F Meijers, E Meschi, R K Mommsen, S Morovic, C Nunez-Barranco-Fernandez, V O'Dell, L Orsini, W Ozga, C Paus, A Petrucci, M Pieri, A Racz, O Raginel, H Sakulin, M Sani, C Schwick, A C Spataru, B Stieger, K Sumorok, J Veverka, C C Wakefield, and P Zejdl. 10 Gbps TCP/IP streams from the FPGA for the CMS DAQ eventbuilder network. *Journal of Instrumentation*, 8(12):C12039, dec 2013.

[3] Tomasz Adrian Bawej, Ulf Behrens, James Branson, Olivier Chaze, Sergio Cittolin, Georgiana Lavinia Darlea, Christian Deldicque, Marc Dobson, Aymeric Dupont, Samim Erhan, Andrew Kevin Forrest, Dominique Gigi, Frank Glege, Guillelmo Gomez Ceballos, Robert Gomez-Reino Garrido, Jeroen Guido Hegeman, Andre Georg Holzner, Lorenzo Masetti, Franciscus Meijers, Emilio Meschi, Remigius Mommsen, Srecko Morovic, Vivian O'Dell, Luciano Orsini, Christoph Maria Ernst Paus, Andrea Petrucci, Marco Pieri, Attila Racz, Hannes Sakulin, Christoph Schwick, Benjamin Bastian Stieger, Konstanty Sumorok, Jan Veverka, and Petr Zejdl. The New CMS DAQ System for Run 2 of the LHC. Technical report, CERN, Geneva, 2015.

[4] V. Brigljevic et al. Using XDAQ in application scenarios of the CMS experiment. *eConf*, C0303241:MOGT008, 2003.

[5] Oliver Sim Brüning, Paul Collier, P Lebrun, Stephen Myers, Ranko Ostojic, John Poole, and Paul Proudlock. *LHC Design Report*. CERN Yellow Reports: Monographs. CERN, Geneva, 2004.

[6] The CMS Collaboration. The CMS experiment at the CERN LHC. The Compact Muon Solenoid experiment. *JINST*, 3:S08004, 2008.

[7] The CMS Collaboration. CMS luminosity measurement for the 2015 data-taking period. Technical report, CERN, Geneva, 2017.

[8] The CMS BRIL Collaboration. The Pixel Luminosity Telescope: A detector for luminosity measurement at CMS using silicon pixel sensors. *Eur. Phys. J. C*, 83(7):673, 2023.

[9] Werner Herr and B Muratori. Concept of luminosity. 2006.

[10] D. Kotliński, E. Bartz, W. Erdmann, K. Gabathuler, R. Horisberger, Ch. Hörmann, H-Ch. Kästli, B. Meier, and S. Schnetzer. The control and readout systems of the CMS pixel barrel detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 565(1):73–78, 2006. Proceedings of the International Workshop on Semiconductor Pixel Detectors for Particles and Imaging.

[11] M. Pernicka and W. Johns. "PixelFED version 5 manual". 2007.

[12] A Rácz. Trigger throttling system for CMS DAQ. 2000.

[13] A Rácz, R McLaren, and E Van der Bij. The S-LINK 64 bit extension specification: S-LINK 64. Technical report, CERN, Geneva, 2003.

[14] Santeri Saariokari. Calibration of luminometers at Compact Muon Solenoid experiment. Luminometrien kalibrointi Compact Muon Solenoid -kokeessa, 2022. Presented 2022.

[15] J Troska, E Corrin, Y Kojevnikov, T Rohlev, and J Varela. Implementation of the timing, trigger and control system of the CMS experiment. *IEEE Trans. Nucl. Sci.*, 53:834–837, 2006.

[16] J Wormersley. The LHC Physics Programme. pages 61–76, 1997.