A SOFTWARE SUITE FOR TESTING THE PERFORMANCE OF THE OPTICAL TRIGGER MOTHERBOARD ELECTRONICS SYSTEM FOR THE CMS EXPERIMENT AT THE LHC

An Undergraduate Research Scholars Thesis

by

AUSTIN WILLIAM SCHNEIDER

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Alexei Safonov

May 2014

Major: Physics
Computer Science

TABLE OF CONTENTS

ABSTRACT

A Software Suite for Testing the Performance
of the Optical Trigger Motherboard
Electronics System for the CMS Experiment at
the LHC. (May 2014)

Austin William Schneider
Department of Physics
Department of Computer Science
Texas A&M University

Research Advisor: Dr. Alexei Safonov
Department of Physics

The Optical Trigger Mother Board (OTMB) and mezzanine card is part of the upgrade of the muon Level-1 trigger system of the Compact Muon Solenoid (CMS) experiment at the Large Hadron Collider (LHC). 72 of the new boards are to be installed and commissioned in 2014 to ensure high efficiency of data collection following the upgrade of the LHC beam energy and intensity. The comprehensive testing of electronics is crucial to operation and efficiency of the CMS muon system, as electronics can become inaccessible for years at a time in the high radiation environment of the detector cavern. A complete test-stand has been designed and built at Texas A&M to exhaustively test new electronics components and long term stability in an environment emulating real operating conditions. As part of the test-stand a special software suite has been developed to operate the test-stand, evaluate new boards, and preserve certification data. The software system was designed with the consideration that it will become a part of the standard testing procedure used in normal operations.

ACKNOWLEDGMENTS

I would like to acknowledge Nick Amin for his work on testing.

I would like to acknowledge Will Flanagan for his work on testing and development of hardware procedures.

I would like acknowledge Jason Gilmore for his development of OTMB firmware and help in development of tests.

I would like to acknowledge Heshani Jayatissa for her work on testing.

I would like to acknowledge Vadim Khotilovich for his advise and expertise in software design.

I would like to acknowledge Alexei Safonov for his mentorship and physics expertise.

I would like to acknowledge Indara Suarez for her work at CMS and reproduction of results at CMS.

I would like to acknowledge Aysen Tatarinov for his work on testing.

NOMENCLATURE

CERN	European Organization for Nuclear Research (Organisation europenne pour la recherche nuclaire)
LHC	Large Hadron Collider
CMS	Compact Muon Solenoid, also refers to the CMS detector system
Muon	An elementary particle similar to the electron, with unitary negative electric charge of -1 but with higher mass (105.7 MeV/c^2). Muons can penetrate through meters of matter without decaying.
Luminosity	A measure of the intensity and rate of collisions in the experiment
CSC	Cathode Strip Chamber, primary muon detection system in the encap regions of CMS
OTMB	Optical Trigger Mother Board, part of the muon Level-1 trigger system for CSCs, generates trigger objects that are used in the global muon trigger
ODMB	Optical Data Mother Board, they read out all of the CSC data to the CMS data acquisition system
FPGA	Field Programmable Gate Array, a programmable logic chip that is used to make all of the trigger decisions
Mezzanine card	PCB with FPGA that connects via mezzanine connection to the OTMB
Crate	Peripheral Crate, a custom electronics crate that houses the muon system triggering electronics

CHAPTER I

INTRODUCTION

The Large Hadron Collider

The Large Hadron Collider (LHC) [1] located in Geneva Switzerland and built at the European Organization of Nuclear Research (CERN), performs proton-proton collisions at high energy. Proton bunches are accelerated to near the speed of light through a series of booster rings until they reach the 27 km (circumference) LHC ring where protons are accelerated to 8 TeV center of mass energy and collided at four locations around the accelerator ring corresponding to the four particle detectors (ATLAS, CMS, ALICE, LHCb) (Fig. I.1). The high energy proton-proton collisions give rise to new particles which in turn decay into other lighter particles that can be identified as they pass through the surrounding detector.

The two largest experiments at the LHC, CMS [2] and ATLAS [3], announced the discovery of the Higgs boson in July of 2012. This discovery has far reaching implications in physics, explaining the mechanism of particles acquiring their mass. However further study promises developments in verifying the standard model via precision measurements, further understanding of the symmetry breaking within the electroweak interaction, and insight into the origin and evolution of the universe. To continue studying the properties of the Higgs after its discovery requires much more data than the LHC can provide at its current beam intensity. In the coming years the LHC will undergo a series of upgrades to reach 14 TeV center of mass collision energy and an order of magnitude greater beam intensity; this will allow precision measurements of the Higgs properties to be made, and will open the door to new physics discoveries including alternate theories such as supersymmetry. LHC beam upgrades however will create a new operating regime for the experiments requiring matching detector upgrades.

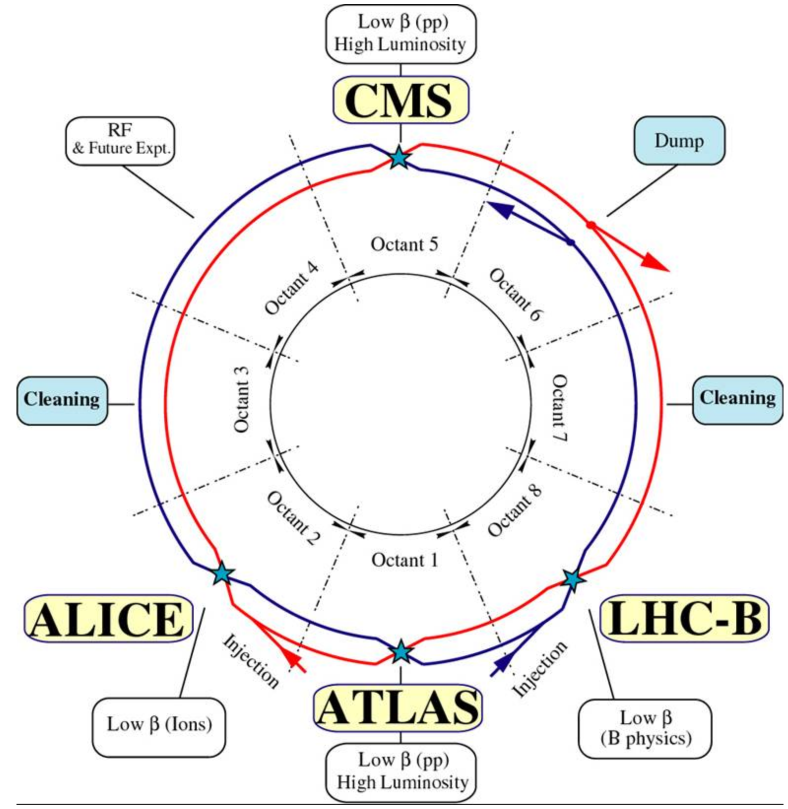


Fig. I.1. Schematic layout of the LHC and experiments (ATLAS, CMS, ALICE, LHCb).

The Compact Muon Solenoid Experiment

The Compact Muon Solenoid (CMS) experiment [2] is one of two flagship detector systems at the LHC. The CMS experiment is a cylindrical detector system 25 m long and 15 m in diameter that consists of multiple layers of sub-detectors systems. When particles pass through the detector they generate signals which are recorded and processed to reconstruct what was produced in the collision. However, bunch crossings occur every 25 ns generating 320 Tbps of data, which cannot be feasibly readout or stored. As a result, data must undergo fast selection at an early stage to identify the most interesting events; this process is known as online triggering [4] and is performed by an intricate system of high-speed digital and analog electronics. The muon system [5] of the CMS experiment is critical for triggering and is a key sub-detector system. It identifies muons that pass through the detector which are part of the decay channels the experiment focuses on; triggering cannot be effective without this valuable information.

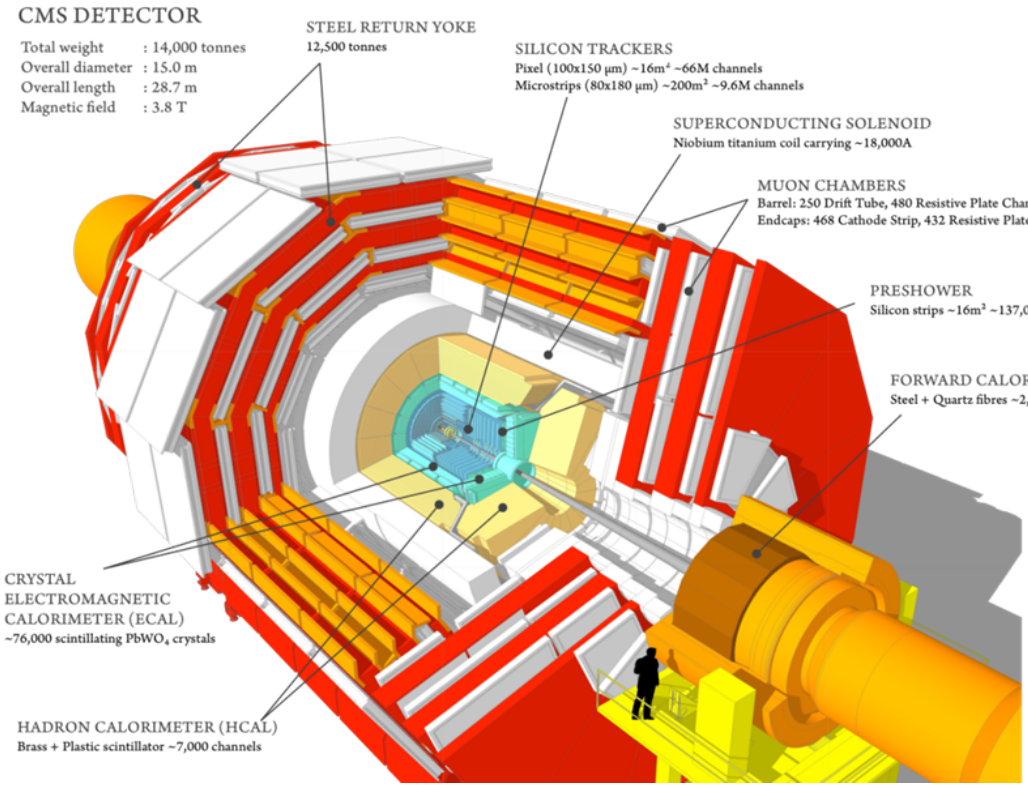


Fig. I.2. Cut out of CMS detector displaying layers of sub-detectors.

The increased LHC proton beam intensity presents a challenge for the CMS detector systems, data acquisition (DAQ), trigger system, and offline computing systems due to a large increase in particle rates, occupancies, and radiation levels. The current electronics of the CMS experiment cannot handle the increased particle rates and will require upgrades as the beam energy and intensity is increased. The forward region of the muon system endcap is exposed to much higher particle flux than other regions of the detector so an upgrade to this region is very important if functionality is to be maintained. The DAQ and triggering system is of particular importance as its performance governs effectiveness of data collection. CMS sub-detector systems will need to undergo upgrades in order to maintain continuous performance in the new operating regime.

Upgrade of the muon system trigger electronics

The muon system is a key detector sub-system of CMS; it detects muons, a product of one of the main decay channels of the Higgs boson and critical for triggering at CMS. Monte-Carlo simulations of the CMS experiment show that after upgrade in luminosity, trigger efficiency of the current system will decrease to levels that will not support future physics analysis (Fig. I.3). However the planned upgrades to the trigger algorithms, trigger electronics, and detector restore trigger efficiency at high luminosity (Fig. I.4).

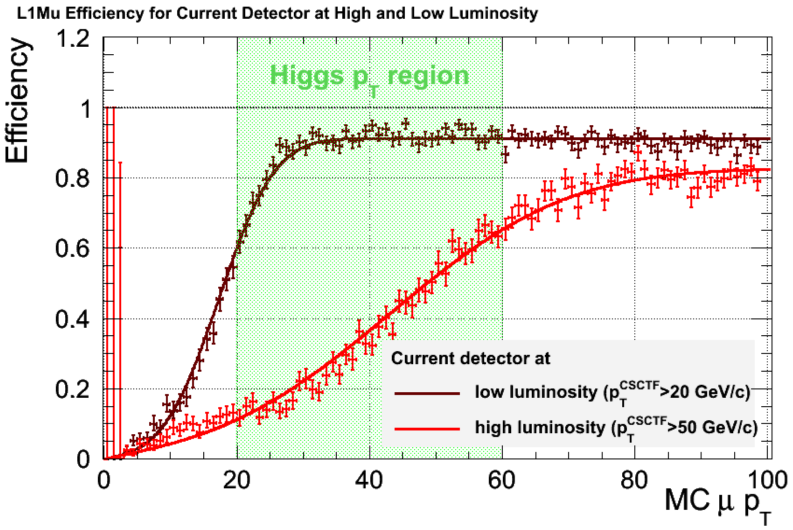


Fig. I.3. Level-1 Muon System with current detector and electronics will yield unacceptably low trigger efficiency at high luminosities.

The muon trigger system is a complex multi-level system of analog and digital components working together. The tasks of the muon trigger system are divided amongst different electronics boards; some of the major boards in the muon trigger system are the Digital Cathode Front End Boards (DCFEBs), Trigger Mother Board (TMB), Data acquisition Mother Board (DMB), Resistive Plate Chamber (RPC), Anode Local Charged Track board (ALCT), Muon Port Card (MPC), and Clock and Control Board (CCB). Many of the trigger electronics boards are receiving upgrades, however the focus of this document is the replacement of the TMB with the Optical Trigger Mother Board (OTMB).

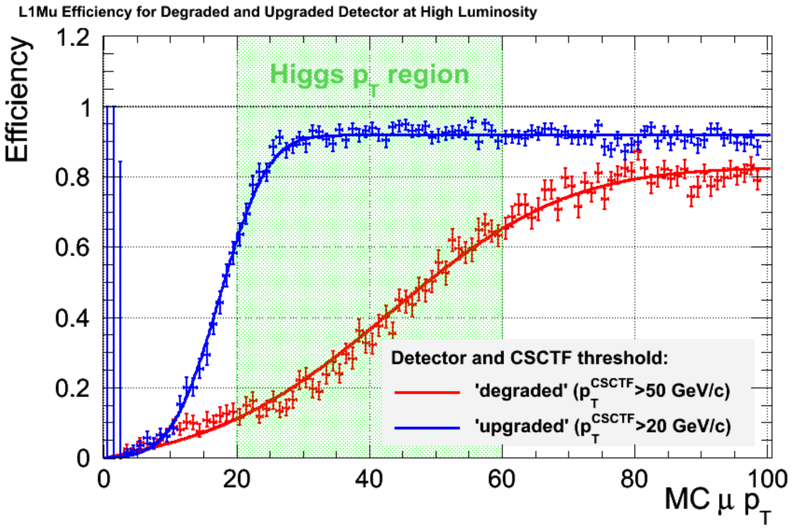


Fig. I.4. Level-1 Muon System with upgraded detector and electronics allows the recovery of performance in the high luminosity regime.

The OTMB and its accompanying mezzanine card reconstruct tracklets in CSC muon chambers, which are then used by higher level components to reconstruct muon candidates (from up to 4 tracklets), and make trigger decisions for the muon system (the outermost subdetector system) and directly affect which events are saved or discarded. The critical role they play and the high speeds at which they operate means that they must function at top efficiency, they also must have long term stability as they cannot be accessed for years at a time when underground at CMS. It is therefore necessary to thoroughly test every board for any possible defects or issues in functionality.

CHAPTER II

METHODS

Electronics

The OTMB has been designed as a drop-in replacement for the TMB; it consists of two components, the OTMB baseboard and OTMB mezzanine card. Both the OTMB baseboard and mezzanine were designed as part of the CMS muon system upgrade. The OTMB handles several functions for the CSC muon sub-detector system at CMS. It uses cathode strip data to find cathode muon stubs for the CMS L1 muon trigger, sends fast cathode pre-trigger signals to the DMB so that the CFEBs can preemptively digitize data for possible readout, and receives L1 muon trigger stubs from the on-chamber anode board, ALCT. Anode stubs are matched to the cathode stubs by timing and quality by the TMB or OTMB. 3D stubs are then built from up to two 2D stubs and are sent via the MPC to the Track Finder crate for matching between muon stations and pt assignment to muon tracks.

The OTMB uses a Virtex-6 FPGA as opposed to the Virtex-2 of the TMB, this provides the OTMB with ×6 logic capacity and ×2 the speed of the TMB, making many improvements to the trigger algorithm possible. Additionally the OTMB provides fiber optic communication with the DCFEBs.

Test stand setup

All the muon Level-1 triggering hardware resides in peripheral crates when operating underground at CMS; these custom crates provide power and cooling to the boards as well as communication pathways between them. The CSC electronics system is a highly interconnected network of electronics and as a result the simplest way to perform testing is to mimic the operational setup. The TAMU test stand consists of a peripheral crate identical to those used underground at CMS. Within the crate, a crate controller, CCB, and MPC are used to help perform tests. The crate controller is required to control the crate and communicate

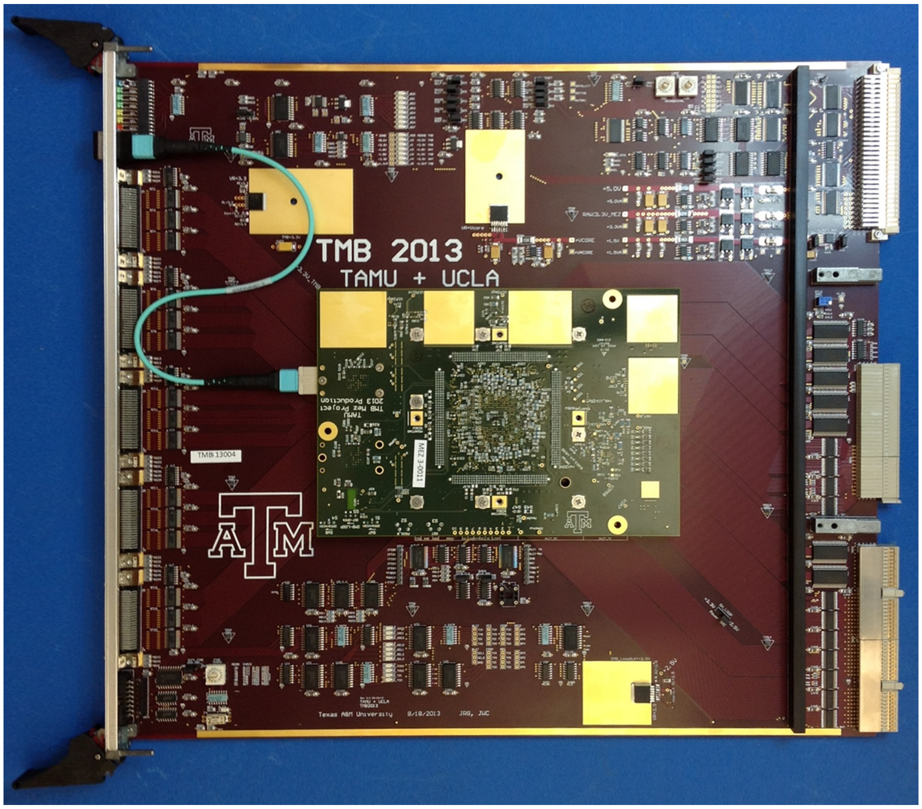


Fig. II.1. Optical Trigger Mother Board (OTMB) and attached mezzanine.

with the boards it contains, it receives commands via fiber link from a computer. The CCB provides some control features for the other boards and is necessary for proper clocking. The MPC is the consumer of the TMB information, and is required for the tests performed on the boards. The OTMB is also housed in the crate alongside the other electronics.

The use of the crate and other electronics for testing avoids the need for electronics to be created specifically for testing. One consequence of this is that some signal pathways become inaccessible in that they can either not be controlled or read by the mezzanine and baseboard. To compensate for this, two simple PCBs were created to modify some signal pathways in the crate. The mezzanine also communicates with other boards, the DCFEBs, via fiber link. Rather than communicating with multiple DCFEBs to test this communication pathway, another mezzanine fitted with a fiber transmitter communicates with the mezzanine and baseboard being tested, a CFEB emulator board (previously developed for testing) is also used to route signals from the backplane to the fiber transmitter board. This gives greater control over the fiber testing procedure and removes the need for custom DCFEB firmware.

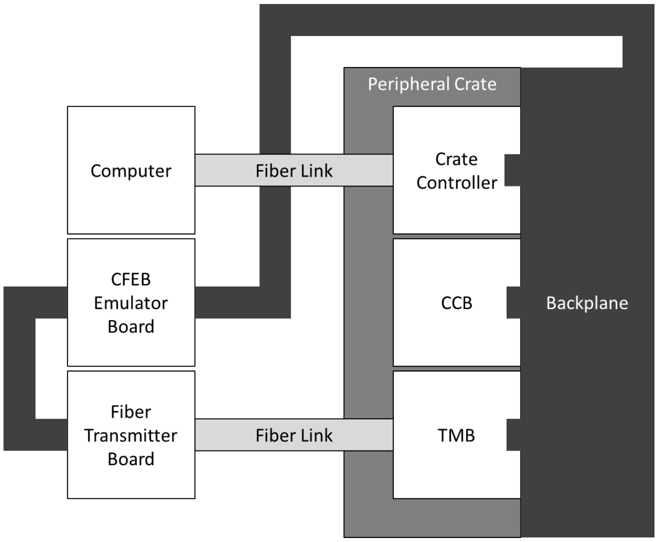


Fig. II.2. TAMU test stand electronics setup.

Firmware and software

Reliable testing of the boards requires functionality that is not available with production firmware. Production firmware written in verilog was modified to create testing firmware which provides testing specific functionality. To test individual pathways, chips, and registers the firmware needs the ability to read and write registers explicitly with provided values. To test the fiber communication, the firmware must synchronize a random pattern with the fiber transmitter board as well as perform matching between the pattern received over fiber link and the pattern generated by the OTMB itself. Although this testing functionality exists in the firmware, testing routines cannot be easily written into the firmware.

A software testing suite was developed for the OTMB that is used to verify the connectivity of all pathways in the board and inter-board communication. This approach has been used for other types of electronics, and has the benefit of minimizing the possibility for human error. The testing suite provides detailed logging information about all tests performed and offers details test failures that help to diagnose any issues that arise. The testing software communicates with the peripheral crate via fiber-optic link (standard for these electronics), and works in concert with board firmware to perform tests. The firmware executes commands received from the software which can include reading data from registers, writing data to registers, and performing randomized sequence fiber testing. The interconnected nature of the software and firmware required them to be developed in parallel. Communication tools from an existing code base (EMU Lib) were used to interface with the crate. Web interface tools in XDAQ (a data acquisition framework used in CMS) were used to create the graphical user interface. The graphical user interface provides a means for non experts to execute groups of tests and interpret the results without in-depth knowledge of the OTMB.

Testing procedure

When the OTMB baseboard and mezzanine are first received from production they require testing to verify their functionality and reliability. A 7-8 person team of non-experts used the testing software to test the boards as they were received from production. Other preliminary work besides testing had to be performed so a testing manual was created to detail the process and provide a reference for future testing.

Mezzanine assembly

The mezzanine cards required some assembly before testing could begin. A stiffener bar had to be installed on the board, some pins needed to be trimmed, and the fiber receiver needed to be installed.

Baseboard assembly

The OTMB baseboard requires installation of VME connectors and Z-Pack connectors that is done with alignment guides and an arbor press. A stiffener bar is also installed, much like on the mezzanine card. Shunt jumpers that govern the settings for the board must then be installed, and rotary switches set to the correct positions. The faceplate and LEDs of the board are installed last.

Board testing

The resistance between key points on each board is then checked and recorded to predict power usage and check for electrical shorts. The mezzanine can then be mounted on the OTMB baseboard and plugged into the crate with proper fiber and programming cable connections. The mezzanine is then flashed with custom TAMUTestStand firmware and the testing sequence begun. After completion of the testing sequence, the mezzanine is then flashed with production firmware (the firmware used during operation at CMS) and a series standard tests from EMU-Lib are run. At this point testing for the boards is complete.

CHAPTER III

DESIGN OF THE TESTING SOFTWARE FRAMEWORK

Software influences

The organization of the software is motivated by several factors, most notably: the functionality and pathways built into the hardware, existing firmware, EMU-Lib, and XDAQ. Functionality of the hardware is important because some of the signal pathways in the boards (mezzanine and baseboard) can only be utilized by performing specific actions, often many pathways are used simultaneously and can be checked in the same way; this naturally lead to grouping of similar pathways into different tests. The existing firmware has a fully functional set of features for the boards and is designed to be used during normal operation, the simplest course of action was to modify the firmware for testing purposes and allow it to be controlled by the software. EMU-Lib (Endcap Muon Library) is standard in CMS projects and provides tools for communicating with the crate, allowing software to communicate with the boards via Versa Module Europa (VME) bus. EMU-Lib is closely associated with XDAQ, a software package that is designed for data-acquisition but provides tools for creating a web based GUI. XDAQ is also standard for CMS applications, and is easy to work with when using C++.

Software design

In software design the question is often asked whether an application should be designed from the top down or the bottom up. In the case of this project it makes sense to use a meet in the middle approach because of requirements on both ends. The upper level requirements come from the choice of XDAQ and features of the user interface. XDAQ requires a main page class for the application and uses subsequent method calls from that page to generate content, this implies that objects for each page should be created and interface with the individual tests. This leads to an association between the tests and the page they are accessed from. The lower level requirements come from testing and logging needs. Tests need to report results (Pass/Fail) as well as detailed information about any failures and some messages directly to the user outside of logs. The logs must contain information about all tests performed on boards, and in each instance the results of testing with detailed failure information. Most interaction with the logs should be automatic, such as the recording and naming, however they should also be human readable to an extent.

Implementation

Testing

These constraints on the software resulted in many of the main design features. Tests are a defined function type that are present in subclasses of the class TestWorkerBase.

```
typedef boost::function<int ()> TestProcedure;
```

TestWorkerBase requires that tests be registered by name for callback from outside of the class, results of the most recent testing cycle are also stored in subclasses of TestWorkerBase. Each instance of TestWorkerBase is also associated with a TestLogger instance. The TestLogger class handles logging for each TestWorkerBase object; entries to the log are created when boards are registered for testing, when a test begins, when an error occurs, when a test ends, and when logging for a board ends. Methods of the of the TestLogger are called when actions are performed through TestWorkerBase and can be called by individual tests to report errors with custom messages. The TestLogger also responds to global actions that apply to all instances of TestWorkerBase, these calls are made by an instance of TMBTestManager which manages instances of TestWorkerBase and TestLogger. TMBTestManager serves as the interface between the pages and tests, making it convenient to perform global actions. Each functional page of the application has two main components, a ”module” that handles the generation of the page, and a ”worker” that provides an interface for performing tasks related to testing or communicating with hardware. In the case of the testing pages an instance of TestWorkerBase fills the role of the ”worker”. The only page that does not follow this design is the CCBBackplaneUtils page which was a preliminary proof of concept of the GUI and ability to communicate with the hardware, written much like the standard portions of EMU-Lib.

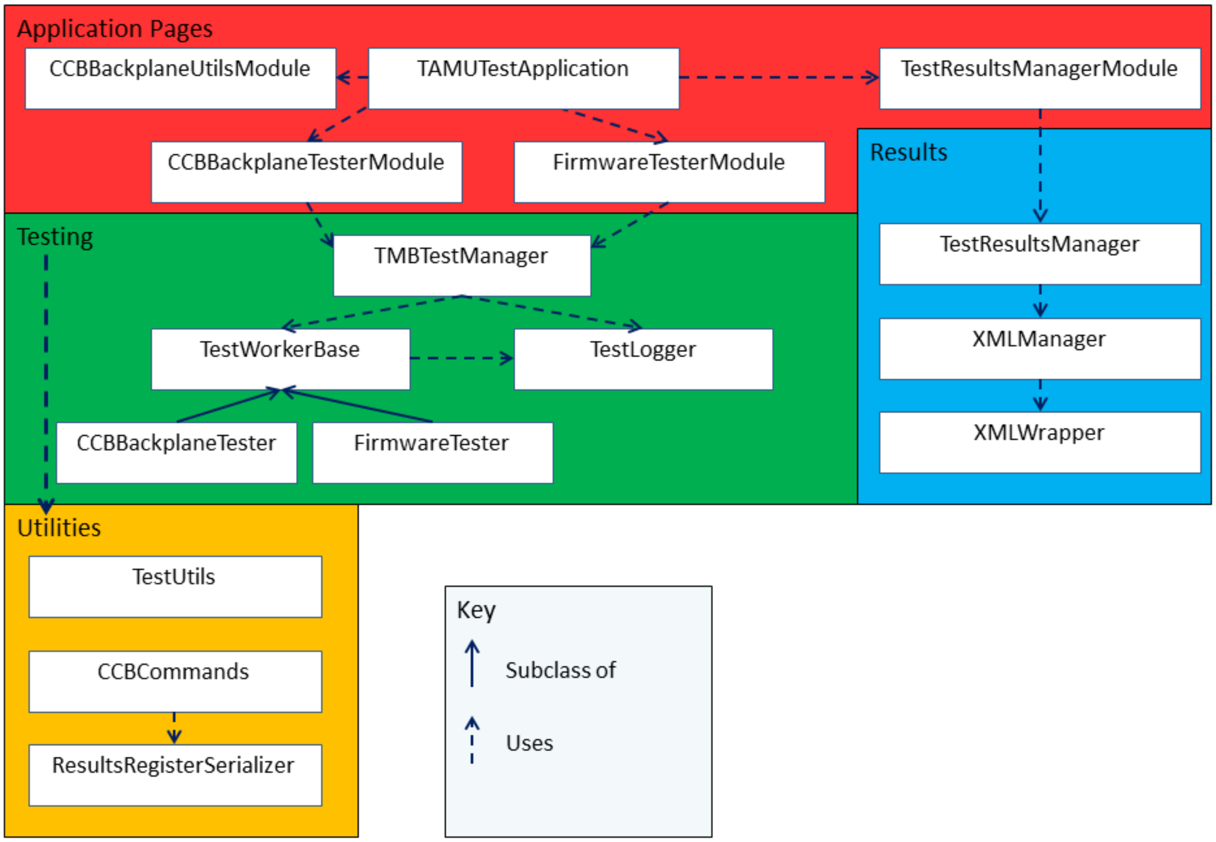


Fig. III.1. Class hierarchy of TAMUTestStand application.

Utilities

There are several collections of utilities that are used in all the testing classes. It was useful to collect these outside of the main class structure of the application so that they could be easily accessed by a number of classes. TestUtils is not a class itself but contains some commonly used utilities for comparisons, user messaging, and string/integer conversions. CCBBackplaneCommands is not a class either; it contains many constants that are taken as input for hardware communication functions, and data structure definitions for processing the data that is returned from hardware communications. CCBBackplaneUtilsModule provides utilities that can be used to manually send commands to the boards, read back information from the VME address space, and utilize serial read-back from the board. These low-level utilities are useful for advanced debugging of the hardware and firmware.

One of the first issues encountered was how to retrieve information from the board while minimizing dependency on communication pathways. To resolve this problem, two bits not used for any essential functions were designated as communication bits, one handshake bit, and one data bit. In combination these could be used for serial communication. Whenever a command is sent to the board the board stores the last command received and additional data if applicable for the command. A read command can then be then sent from the software, upon receiving this command the board freezes the value of the register until another command other than the read command is seen. Successive read commands read each bit back one at a time, the handshake bit then signals the end of the data. The ResultRegisterSerializer provides a convenient interface to this method of communication, and specialized structures in the CCBBackplaneCommands help the processing of data from different commands.

Results

To automatically handle interaction with the logs and make the information contained in the logs more accessible to non-experts the TestResultsManager class was created. TestResultsManager keeps track of all the logs and provides an interface for viewing them. Functionality was also added to track changes to the logs while tests are being performed. The logs are stored in xml format so an indexing and querying system was created to access the information contained in the logs. The XMLManager class keeps track of individual xml files, checking for changes and updating each XMLWrapper object if necessary, performing any required file or file-system operations, and providing a querying interface for the collection of xml files. The XMLWrapper class is a wrapper for a single xml file, the class handles operations pertaining to that file and its corresponding document object stored in memory. XMLWrapper also provides an querying interface for the document object associated with its xml file.

Testing sequence

There are many signal pathways in the OTMB baseboard and mezzanine that need to be tested, some of these are tested by writing data to certain areas and reading it back, but others are tested indirectly by using functionalities of the boards and their firmware.

CCBBackplaneTester

The CCBBackplaneTester is one of two TestWorkerBase subclasses, it contains functionality tests and signal pathway tests that are routed through the crate backplane to the CCB. The first tests to run check the reset capability of the OTMB; two types of reset are used during normal operation, L1Reset and HardReset, which reset the state of the boards in different ways and are used during all other tests. The CommandBus is then checked to ensure that commands can be sent to the board and are received properly. These are all the tests that are performed from the CCBBackplaneTester:

L1Reset : Writes information to several registers, send L1Reset, check if register is cleared.

TMBHardReset : Writes information to several registers, send HardReset, check if register is cleared.

CommandBus : Walks over range of possible commands and checks to see if the correct command code was received.

PulseCounters : Sends several 25ns pulses of different kinds and checks that the appropriate count and flags are shown.

CCBReserved : Tests CCBReserved bits.

TMBReservedOut : Tests TMBReservedOut bits.

DMBReservedOut : Tests DMBReservedOut bits.

DataBus : Tests Data Bus functionality by walking over a range of values, writing them to the Data Bus and reading them back.

CCBClock40 : Checks the functionality of the 40Hz clock.

DMBReservedInLoopback : Tests DMBReservedInLoopback bits.

DMBL1AReleaseLoopback : Tests DMBL1AReleaseLoopback bits.

TestLEDFrontPanel : Tests outgoing signals on the front panel controlled by the TMB. LEDs are used to display the state of these signals and must be human read.

FirmwareTester

The FirmwareTester class handles tests that are firmware controlled, these include all of the tests where communication with another board is involved. The main idea behind firmware controlled tests is that the firmware can operate at speeds much faster than the software is capable of, and therefore more rigorous testing can be performed. The software simple enables, disables, and checks these tests. When a firmware test is performed the firmware on the boards generates a pseudo-random sequence of data and checks the sequence it generated with the sequence it receives. There are three types of tests: fiber, DMB, and RPC. The fiber tests require a fiber transmitter board, and tabletop modified mezzanine card that transmits a sequence of data over fiber to the OTMB baseboard and mezzanine being tested. The same pseudo-random sequence is generated on each board after synchronization and the two are compared on the board being tested. The DMB and RPC tests work in a different manner, they utilize loopback boards, a special PCB that feeds board output back into the board via another signal pathway. The OTMB generates pseudo-random data, sends it out to the loopback boards, and checks that the data coming back in matches. This process simulates communication with the DMB and RPC boards.

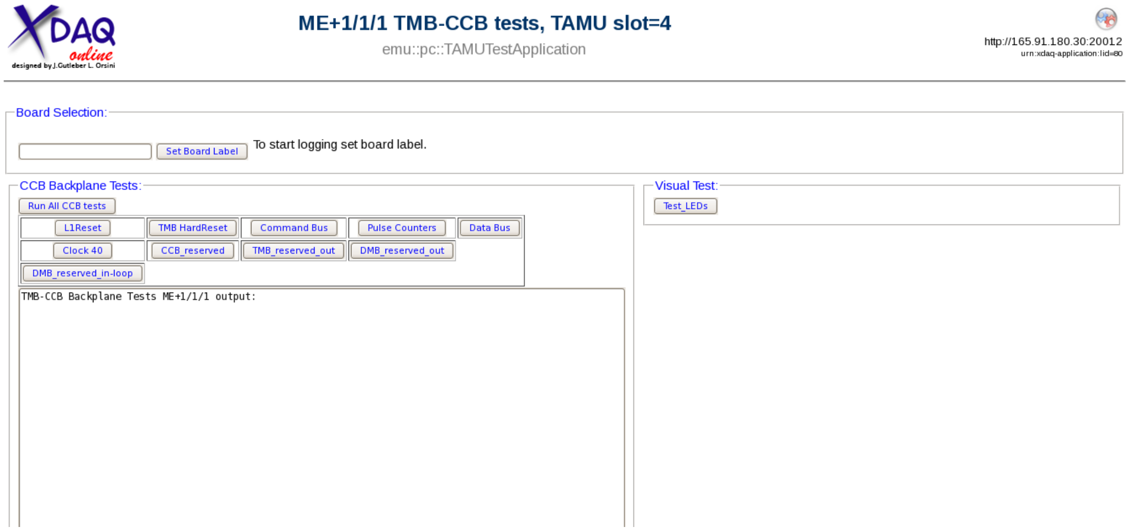
CHAPTER IV

RESULTS

Software

The completed software testing framework encompasses all the tests needed for verifying the functionality of the boards and connectivity of the signal pathways. It is entirely controlled by the graphical user interface which is split into multiple pages by functionality.

The CCB Backplane Tests page is where board tracking begins with the entry of the board label. It also performs tests of the basic functionality and all backplane signals related to the CCB, users need only to click the ”Run All CCB Tests” which will run all of the tests on the page, the only other user intervention required is the visual test which has the user look at lights on the front panel of the board and enter information about the patterns displayed.



ME+1/1/1 TMB-CCB tests, TAMU slot=4

emu::pc::TAMUTestApplication

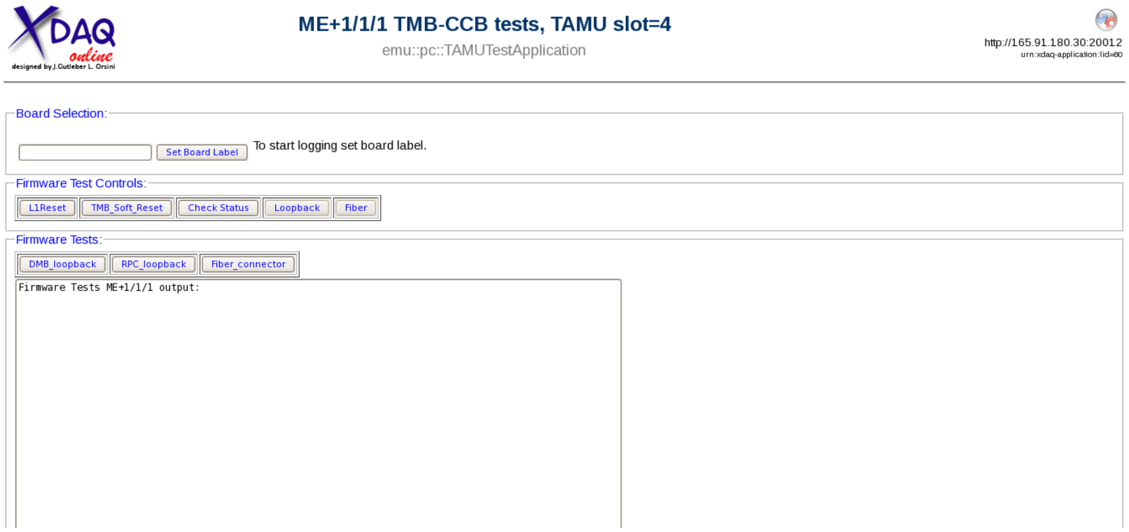
http://165.91.180.30:20012
urn:xdaq-application:lid=80

To start logging set board label.

```
TMB-CCB Backplane Tests ME+1/1/1 output:
```

Fig. IV.1. CCB Backplane Tests page.

The Firmware Tests page is where fiber and loop-back tests are controlled, the user can start and stop these tests from here, there are also fixed duration tests and extra commands for clearing registers that are used only for diagnostics.

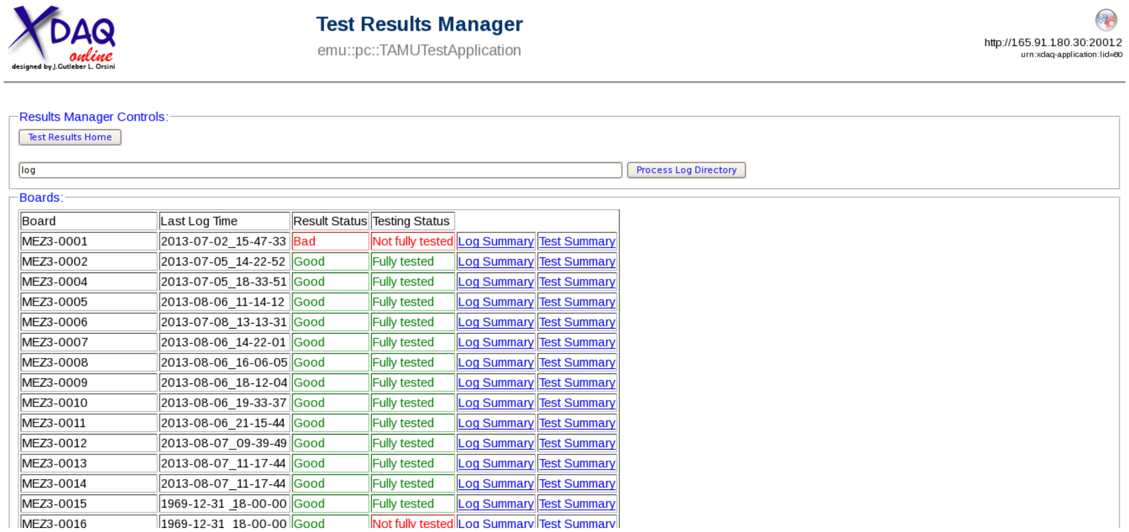


To start logging set board label.

```
Firmware Tests ME+1/1/1 output:
```

Fig. IV.2. Firmware Tests page.

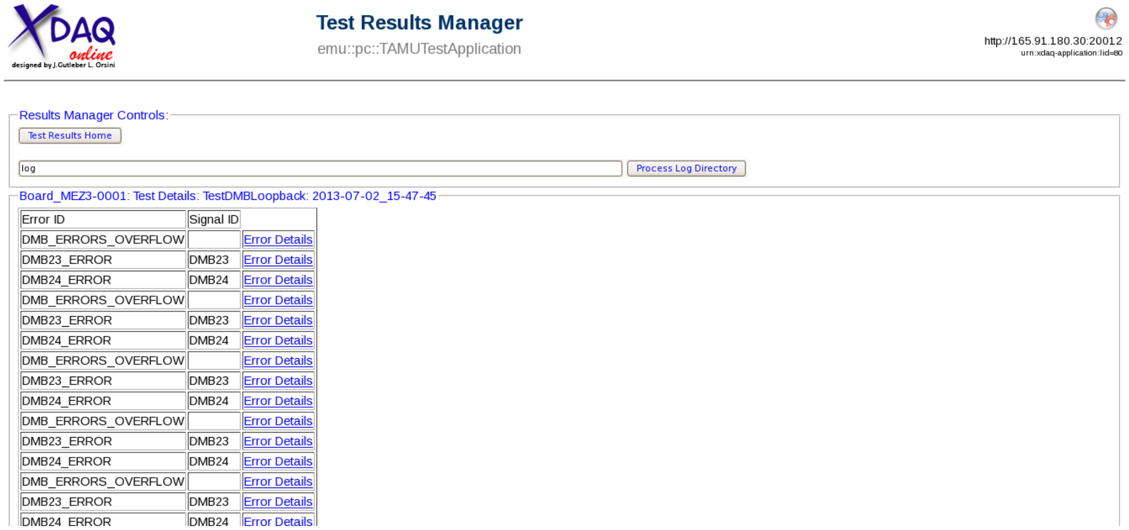
The Tests Results page shows a summary of test results from all boards and if they have been fully tested or not. Fig. IV.3 shows an example of the a display one might see after some testing.



Board	Last Log Time	Result Status	Testing Status		
MEZ3-0001	2013-07-02_15-47-33	Bad	Not fully tested	Log Summary	Test Summary
MEZ3-0002	2013-07-05_14-22-52	Good	Fully tested	Log Summary	Test Summary
MEZ3-0004	2013-07-05_18-33-51	Good	Fully tested	Log Summary	Test Summary
MEZ3-0005	2013-08-06_11-14-12	Good	Fully tested	Log Summary	Test Summary
MEZ3-0006	2013-07-08_13-13-31	Good	Fully tested	Log Summary	Test Summary
MEZ3-0007	2013-08-06_14-22-01	Good	Fully tested	Log Summary	Test Summary
MEZ3-0008	2013-08-06_16-06-05	Good	Fully tested	Log Summary	Test Summary
MEZ3-0009	2013-08-06_18-12-04	Good	Fully tested	Log Summary	Test Summary
MEZ3-0010	2013-08-06_19-33-37	Good	Fully tested	Log Summary	Test Summary
MEZ3-0011	2013-08-06_21-15-44	Good	Fully tested	Log Summary	Test Summary
MEZ3-0012	2013-08-07_09-39-49	Good	Fully tested	Log Summary	Test Summary
MEZ3-0013	2013-08-07_11-17-44	Good	Fully tested	Log Summary	Test Summary
MEZ3-0014	2013-08-07_11-17-44	Good	Fully tested	Log Summary	Test Summary
MEZ3-0015	1969-12-31 _18-00-00	Good	Fully tested	Log Summary	Test Summary
MEZ3-0016	1969-12-31 _18-00-00	Good	Not fully tested	Log Summary	Test Summary

Fig. IV.3. Test Results page.

Within the test results page is the error viewer which allows the errors reported for an individual test to be viewed. Fig. IV.4 shows an example of errors one might see.



Test Results Manager

emu::pc::TAMUTestApplication

http://165.91.180.30:20012
urn:xdaq-application:lid=80

Results Manager Controls:

log

Board_MEZ3-0001: Test Details: TestDMBLoopback: 2013-07-02_15-47-45

Error ID	Signal ID	
DMB_ERRORS_OVERFLOW		Error Details
DMB23_ERROR	DMB23	Error Details
DMB24_ERROR	DMB24	Error Details
DMB_ERRORS_OVERFLOW		Error Details
DMB23_ERROR	DMB23	Error Details
DMB24_ERROR	DMB24	Error Details
DMB_ERRORS_OVERFLOW		Error Details
DMB23_ERROR	DMB23	Error Details
DMB24_ERROR	DMB24	Error Details
DMB_ERRORS_OVERFLOW		Error Details
DMB23_ERROR	DMB23	Error Details
DMB24_ERROR	DMB24	Error Details
DMB_ERRORS_OVERFLOW		Error Details
DMB23_ERROR	DMB23	Error Details
DMB24_ERROR	DMB24	Error Details

Fig. IV.4. Test Results page.

The graphical user interface of the software testing framework was successfully used by non-experts to test and diagnose issues with the boards. The software is modular enough to be reused for future testing projects.

Testing Results

87 mezzanine cards were produced, assembled, and tested. Testing of the boards went according to plan and required only minor modification of the procedure to check for bent pins on the Snap12 fiber receivers. Of the 87 mezzanine cards produced 83 passed all tests initially. Two of the boards failed all software tests pointing to a serious issue, upon further investigation with an oscilloscope it was determined to be an issue with the Quartz Crystal Phase-Locked Loop (QPLL) chip. The QPLL chips were replaced and retesting repairs showed no issue with the boards. The other two failures turned out to be false positives, it was determined to be a procedural error that resulted in the mezzanine having a bad connection to the baseboard. Retesting of the false positives showed no issues.

During the assembly of the mezzanine cards after production it was noticed that 10 of the Snap12 fiber receivers had a single bent pin. Testing with the bent pins showed no issue, however all 10 Snap12 receivers were either repaired or replaced to avoid any possible issues in the future. After repairs all 87 of the produced boards were shown to be in working order. 87 OTMB baseboards were also produced, these were paired with the already tested mezzanine cards and tested again together. All of the 87 boards passed without issue and no modifications to the procedure were needed.

Board status

After the completion of testing and repairs 78 of the 87 boards were sent to CMS and 72 are now installed underground, ready for operation. 5-6 of the boards will stay in the United States at several institutions for future testing and development that will be needed as the bean intensity of the LHC is further increased. The rest of the boards are to be kept as spares or given to other institutions.

The installation of these boards in the forward region of the CMS Muon Endcap will provide significant triggering improvements and allow the CMS experiment to handle the higher luminosity experienced after the LHC upgrade.

REFERENCES

[1] Lyndon Evans and Philip Bryant. LHC Machine. *Journal of Instrumentation*, 3(08):S08001, 2008.

[2] The CMS Collaboration. The CMS experiment at the CERN LHC. *Journal of Instrumentation*, 3(08):S08004, 2008.

[3] The ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. *Journal of Instrumentation*, 3(08):S08003, 2008.

[4] D. Acosta et al. Efficiency of finding muon track trigger primitives in {CMS} cathode strip chambers. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 592(12):26 – 37, 2008.

[5] The CMS Collaboration. The performance of the CMS muon detector in proton-proton collisions at $s = 7$ TeV at the LHC. *Journal of Instrumentation*, 8(11):P11002, 2013.