PoS PROCEEDINGS OF SCIENCE

FED Firmware Interface Testing with Pixel Phase 1 Emulator

Matthew Kilpatrick for the CMS Collaboration[*][†]
Rice University
E-mail: matthew.kilpatrick@cern.ch

A hardware emulation of the CMS pixel detector phase 1 upgrade front-end electronics has been developed to test and validate the architecture of the back-end electronics (FED) firmware. The emulation is implemented on a Virtex 6 FPGA on the CERN GLIB uTCA platform, utilizing an 8-way SFP FPGA Mezzanine Card to drive compatible optical transmitters to the back-end electronics at 400 Mbps. The firmware emulates the complex functions of the phase 1 pixel readout chips (PSI46digv2 and PROC600) and token bit manager ASICs and allows for possible abnormalities that can occur in the output data stream. The emulation implements both fixed data patterns that are used as test vectors and realistic simulated data to drive the readout of the FED at the expected data and trigger rates. Testing software was developed to control the emulator and verify correct transmission of data and exception handling in the FED. An installation has been integrated into the pixel DAQ test system at CMS to be used for fast validation of FED firmware upgrades.

Topical Workshop on Electronics for Particle Physics
11 - 14 September 2017
Santa Cruz, California

[*] Speaker.
[†] A thanks to the CMS collaboration

1. Phase 1 Pixel Emulation and Upgrade

In February 2017, the Phase 1 upgrade of the CMS pixel detector was installed. This new system increased the pixel layers in the barrel (BPIX) from 3 to 4 and the number of endcap disks (FPIX) from 2 to 3. It went from 768 modules to 1184 in the BPIX and still has 672 modules in the FPIX, but they are now using the same module as the BPIX. The total number of pixels increased from 48 million (18 million) to 79 million (45 million) in the BPIX (FPIX). A detector module is made out of two rows of readout chips (ROCs), which are bump bonded to the silicon pixel sensors. The physical size of a pixel is $100 \times 150\ \mu\text{m}^2$. This upgrade will lead to higher efficiencies for track reconstruction, lower dead-time, lower fake tracks reconstructed, and will help with the better identification of particles in interactions [1,2]. The upgrade will improve the vertexing resolution which will help to resolve multiple vertices in large pileup scenarios, where pileup is the number of particle interactions in a single beam crossing. The LHC is expected to further increase the instantaneous luminosity and thus also the pileup in the years leading to long shutdown 3.

The new structure of the pixel detector requires upgraded data aquisition (DAQ). A new Front-end Driver (FED) hardware has been developed which requires new firmware to handle the many different event scenarios that can occur. Since the detector was not ready at the time of developement an emulation of the Pixel Phase 1 Upgrade of the CMS detector was completed to supply the correct signals.

The emulation consists of a clock and trigger signal that is sent to three Gigabit Link Interface Boards (GLIBs) that are controlled with customized firmware and software. These triggers can be customized to simulate any trigger rate that is allowed by the CMS detector. Output signals from the GLIBs are used to test the FED firmware, to confirm it can correctly identify events that may contain errors and can be used to create highly improbable scenarios, such as every event missing a ROC. The internal state machine of the emulated firmware includes the signals coming from the Token Bit Manager (TBM) and the ROCs. We will take a deep look into the components of the pixel emulation and understand how it emulates multiple pixel modules in Sec. 1.1, 1.2, and 1.3. Then, how it can customize the events to generate expected errors in the FED Sec. 2 and the ability to emulate real data distributions, Sec. 2.1. Finally, a full test of the framework to test the maximum data throughput of the FED, Sec. 3.

1.1 Token Bit Manager

The TBM is responsible for the readout of a collection of eight ROCs and transmission to the Front End Driver (FED). It distributes the clock, trigger, and reset signals while also transmitting the pixel address from the ROCs. There are three different variations of the TBM that each handle different data rates that are expected in the various layers of the pixel detector. The TBM08 is for the FPIX as well as layer 3, and 4 of the BPIX. The TBM09/10 is for layer 2 and layer 1 [1,3].

The TBM09 is the equivalent of two TBM08s. The output of each of these is a 400 MHz encoded optical data stream. The layers that experience a higher flux of particles will require multiple fibers for this output. This upgraded system will be able to read out the higher data rates that are expected in Run 2 of the LHC [4]. The TBM is only one part of the upgrade, the ROC also needs to be upgraded, which controls the pixel readout.

1.2 Readout Chip

The pixel detector that has been in use since 2007 uses a PSI46V2 chip and has a 2.5% data loss rate in layer 1 of the pixel detector at the design luminosity of the LHC and designed trigger rate of the CMS detector. The ROC from the pixel phase 0 detector is designed to withstand a pixel hit rate of 115 MHz/cm^2 [4]. The increased luminosity of the LHC causes rates up to 600 MHz/cm^2. This data rate would cause the data loss of the detector to increase, so a new design has been implemented. A new chip was installed, the PSI46dig. Layer 1 will have a different chip, PROC600, that can readout faster than the PSI46dig chip, but the internal structure of the event will be the same. A ROC is used to store and readout the hit information of pixels that exceeded the charge threshold. A hit will contain the address (column and row), pulse height, and time stamp information [1,4]. We ensure a reduced data loss by increasing the readout per unit time and the storage size for events waiting to be read out.

1.3 Encoding/Decoding

Once an L1A is received, the TBM will send the event information at 160 Mbps. Since there are two channels in the TBM08, we want to combine them. These two signals are then interleaved together at a bitwise level. This new signal is sent at 320 Mbps, but it requires some further encoding. A special dictionary is used to do a 8-bit to 10-bit encoding to achieve a 400 Mbps signal. After doing this conversion, the final step is to have a non-return to zero inverted encoding. This will take the input signal and negate the output only when it is '1'. This is the final signal that is then sent to the FED to thus be decoded and have the event information extracted.

The importance of this encoding scheme is to have an equal number of 0s and 1s in the data stream. A nice result of this is seen when the TBM is outputing the idle signal, the encoded output is '11111 00000'. Since this is being transmitted at 400 Mbps it will mimick the 40 MHz clock. It is then easy to monitor and extract when an event is being sent.

2. Customization and Firmware Testing

The phase 1 emulator can emulate the TBM08, TBM09, TBM10, PSI46dig, and PROC600. The customization is ideal to allow for consistent tests of the FED firmware. The bitwise signal of the events can be altered to cause errors in the FED. For example, the headers of any event can be changed such that the FEDs decoder will miss the event and cause timeouts or event number errors. The transmission of the events can also be delayed by an integer number of clock cycles, to emulate the travel time for signals from the pixel detector to the FEDs. Event readout can be done in a fixed data size, where every event is the same, or in SRAM mode, where one can program via software the event size and pixel location. The whole phase 1 emulator is an independent test stand. Multiple tests stands can be set up and used to test the FED firmware without the need for clock input from CMS or beam collisions.

The phase 1 emulator recieves commands from an advanced mezzanine card and responds in agreement with the real pixel detector. Once a reset is seen, the emulation will initiate the correct protocol. The signal is then sent to the FED which is decoded and compared to what was originally sent. The FED is able to decode any event that is sent. Events that have resets or possible errors are still decoded but marked in an error FIFO. All of these errors can then be easily monitored. The FED has multiple error counters that can count independently for each channel. These are all probed in the phase 1 emulator framework to confirm that the count for each error is accurate. An example of a special error, known as a Single Event Upset (SEU), which can cause a channel to stop transmitting a signal all together. If this happens the FED will decode a signal that is nonsensical. The phase 1 emulator is able to emulate this and confirm that it is handled correctly. The flexability of the system allows for specific and consistent errors to be sent and decoded by the FED. This allows a more robust firmware to be developed and integrated into the DAQ.

2.1 SRAM Mode

The SRAM of the GLIBs are a software loadable memory that can be accessed by the event readout framework. There are two separate memory locations that each hold approximately 8.4 MB of data. The first SRAM is designated to hold the hits/ROC distributions, see Fig. 1. The second is to hold the emulated pixel locations for each hit. The SRAM read is driven by a 160 MHz clock. Since each GLIB can emulate 16 independent channels, there are 32 different operations which must occur to emulate an event. The first SRAM only needs to be read once per event and only takes 100 ns to complete. The second SRAM needs to be read any time a pixel hit needs to be sent, one hit per channel it will also be 100 ns. The smallest amount of time for an event readout is 1.150 μs, therefore SRAM readout can be done easily at 100 kHz.

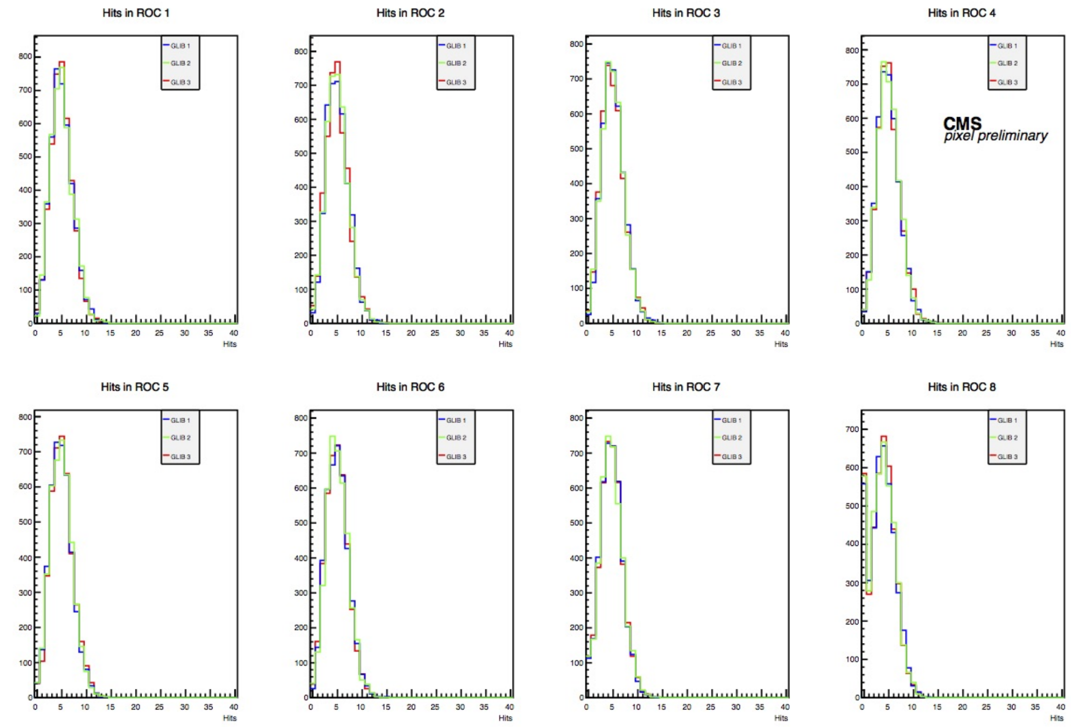

Figure 1: A Poisson hit distribution of events loaded onto the SRAM of the phase 1 emulator. The distributions can be independent for each board and readout at a sufficient speed. Number of ROCs in channels can also be changed depending on the emulated layer.

3. Data Throughput through FEROL Link

Using the phase 1 emulator it is possible to check the data throughput at the frontend readout optical link (FEROL) link of the FED. Since the size of events in the emulator are easily controlled it is simple to find the maximum possible data throughput of the FED. If the FED recieves too much data and the internal FIFO structure reaches capacity the FED will move to a busy state and cause triggers to stop. Once it has caught up the triggers can start again. Because of this, once the event sizes are large enough the trigger throttling causes a maximum data throughput. Using a 10 G FEROL link, we can acheive a maximum throughput of approximately 4.575 Gbps. We are able to test the throughput using fixed or SRAM data to allow for more realistic conditions.

4. Conclusion with Outlook

The phase 1 emulator is a standalone project that can be used to test the CMS Pixel phase 1 FED firmware's readout structure. The software tests of the FED are completely automated. As soon as a new firmware version is released the testing can be completed within a few hours. If a new feature is added a test can be made and included in the framework quickly. This allows for minimum down time between a new version of the firmware and deployment into the production system. The phase 1 emulator was able to give a preliminary data rate of 4.575 Gbps before the installation of CMS. With the framework already included, ability to generate custom errors, readout realistic data, and probe the maximum data rate the phase 1 emulator will be able to operate with the pixel detector for the foreseeable future.

5. Acknowledgements

This contribution was supported by the U. S. Department of Energy grant DE-SC0010103.

References

[1] CMS Collaboration, *CMS Technical Design Report for the Pixel Detector Upgrade*, Tech. Rep. CERN-LHCC-2012-016. CMS-TDR-11 (CERN,Geneva, 2012). https://cds.cern.ch/record/1481838

[2] CMS Collaboration, *The CMS experiment at the CERN LHC*, J. Instrum. 3, S08004 (2008).

[3] R. Stringer. A digital readout system for the CMS Phase I Pixel Upgrade. Journal of Instrumentation, 10(04):C04037, 2015.

[4] B Meier. CMS pixel detector with new digital readout architecture. Journal of Instrumentation, 6(01):C01011, 2011.