Development and Commissioning of the CMS Luminosity Readout System

J. Jones [a]

[a] Princeton University, Princeton, NJ, USA

john.jones@cern.ch

Abstract

We discuss the development and commissioning of a luminosity monitor. It is based on hardware that provides real-time histograms of data from the forward hadronic calorimeters in CMS. Measuring the total energy deposition and occupancy in these detectors allows us to calculate the relative instantaneous luminosity of the collider on a bunch by bunch basis, which is useful for machine diagnostics. Once calibrated with measurements from the LHC we will be able to make the first proton-proton inelastic cross-section measurement. Further information on the calculations themselves can be found in [1]. This paper discusses the readout hardware design and implementation.

I. INTRODUCTION

Luminosity monitoring is a critical component of any particle physics experiment, allowing one to compute the cross-section for the physical processes occurring in the detector. The luminosity measurement in CMS will be used to monitor the LHC beam and provide overall normalisation for physics analyses. For offline analyses in CMS, the design goal is a systematic accuracy of less than 10% for a range of beam luminosities from $10^{28}cm^{-2}s^{-1}$ to $10^{34}cm^{-2}s^{-1}$. The CMS luminosity monitor is a system based on a mezzanine card called the HLX, mounted on the HCAL Trigger and Readout (HTR) board. It operates by histogramming data from the HTR boards for the Hadronic Forward (HF) calorimeters in CMS. It computes energy sums, and counts the number of physical towers below, between and above a pair of preset energy thresholds for this region. Both of these measurement techniques can be used to infer the relative luminosity. Combined with a careful offline study, both the delivered and recorded luminosity can be measured. Using smaller sets of data at approximately one second intervals, the real-time behaviour of the system can be monitored online.

A. The HF Calorimeter

The HF calorimeter comprises φ-wedges (detector co-ordinate system) of iron with quartz fibres embedded into them. These emit Cerenkov light which is collected by an array of PhotoMultiplier Tubes (PMTs). There are 36 in the whole HF detector, each of which corresponds to one HTR board off-detector (see figure 1). QIE ASICs are used to digitise the signals from the HF PMTs on a per-bunch basis, which are then routed to the 36 HTR boards. The HF has a physical coverage over the range $3 < |\eta| < 5$ in both end caps of the detector.

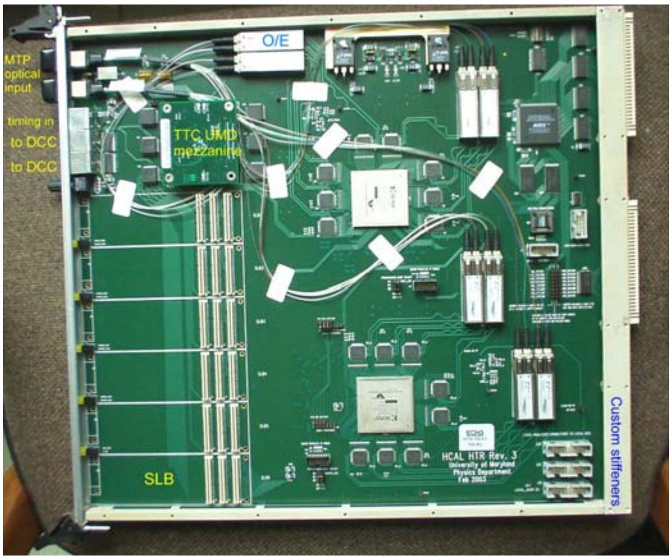

Figure 1: Photo of the HTR board. There are two main processing FPGAs which receive optical data from the front-end electronics. The six mezzanine card sites can be seen on the bottom-left of the board.

A wedge can also be divided into a set of physical towers, as shown in figure 2.

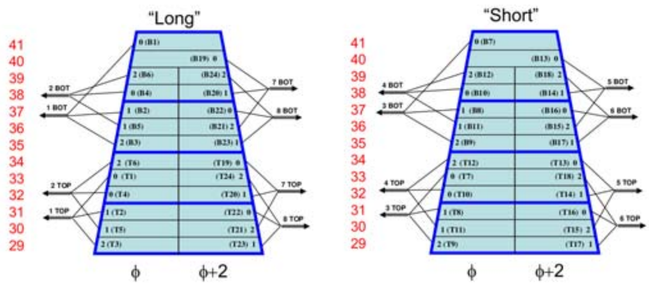

Figure 2: Figure showing the long and short fibre mappings for the physical towers in each HF wedge. The fibres are mapped to either the top of bottom FPGA on each HTR board, divided between the inner and outer half of the wedge.

The energy sums for the towers in each wedge are sent to a HTR board where the data are mapped and processed by two Field Programmable Gate Arrays (FPGAs). These data are then forwarded to the HLX mezzanine card to be processed into luminosity histograms.

II. HARDWARE IMPLEMENTATION

The HLX is a HTR mezzanine card which processes the data from the HF calorimeter. In normal use, it occupies SLB site 5 on an HF HTR card (there are six SLB sites in total, but only two are used for the HF readout). It comprises a small

PCB with a 2Mgate FPGA (Xilinx Virtex-II Pro) and a large CPLD (Xilinx 9500 series). The CPLD is used in combination with a FLASH PROM to configure the FPGA automatically at power-up or on-demand via VME. The firmware can also be loaded via VME, so the use of a programming cable is unnecessary once the CPLD has been configured. Pictures of the HLX are shown in figures 3 and 4. The FPGA performs the calculations associated with the luminosity measurement and provides the functionality necessary to ship the data to the luminosity readout server.

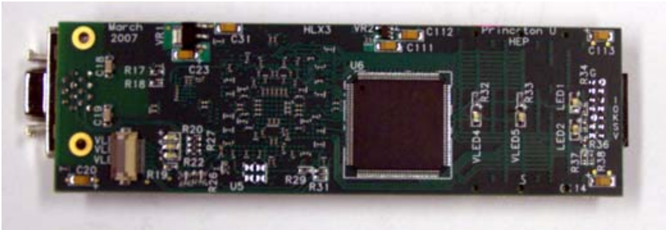

Figure 3: Top view of the final version of the HLX board.

Figure 4: Bottom view of the final version of the HLX board.

In addition to logic resources, the board also incorporates a 10/100Mb Ethernet link (used for shipping the data to the server), and power regulation circuits. A triple PMC connector is used to attach the HLX to the HTR, providing both power and data connectivity between the two boards. In this way data processed by the HTR is forwarded to the HLX before being sent to the luminosity readout server over Ethernet. There are 36 HLXs in total, corresponding to 18 for each of the HF+ and HF- detectors. Each HLX produces an Ethernet data stream with a line rate of a few Mb/s per card. The signals from these boards are sent to a passive patch panel above each HF HTR crate. At this point the links are converted to standard Ethernet cables. The 36 cables are then routed to a 40-port HP switch in rack S2F17, where the data are merged before being sent to the luminosity readout PC, as shown in figure 5. The total bandwidth required for this system is approximately 50Mb/s, which is of the same order as the maximum bandwidth of the network interface. Therefore care must be taken to avoid overloading both the switch and the readout PC due to an excessively high instantaneous data rate from the HLXs. This is avoided in the HLX readout system by configuring each HLX with a different readout delay. Hence the data from each board arrives slightly later than the preceding one, minimising the load on the switch and the PC. This approach has been tested at a rate of 72Mb/s for an extended period of time without any errors.

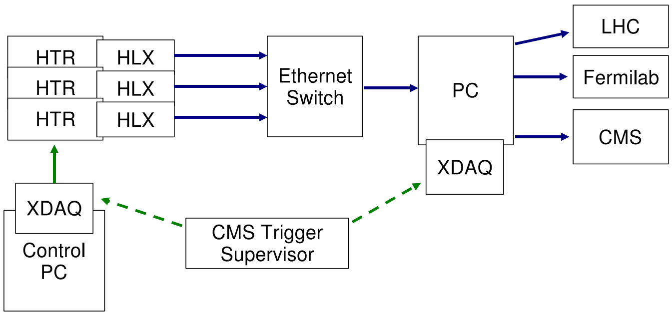

Figure 5: Readout and control pathways for the HF luminosity readout system. 36 HLXs produce data which is routed to an Ethernet switch, where the data are merged and streamed into the readout PC. This PC processes the data before distribution to the various LHC/CMS clients.

III. Firmware Implementation

The firmware for the luminosity measurement can be sub-divided into two parts: the first stage of processing is carried out in the HTR board. This involves taking the raw data from the HF calorimeter and performing linearization or pedestal subtraction depending on whether the data are used for an E_T sum or occupancy-based luminosity measurement. As the HTR is also used for detector readout and triggering, care must be taken to ensure that the configuration of the luminosity monitor has no effect on the HF trigger data. The pathways for trigger and data readout are therefore kept separate from the luminosity pathway, ensuring that changes in the settings for one do not affect the other. The second stage of firmware processing involves applying a per-physical-tower mask to the data to prevent `hot' or `warm' towers from creating measurement bias. These settings are configurable over VME and therefore from the CMS Trigger Supervisor. It is possible to individually mask both the long and short fibres in each HTR tower for all the towers used in the E_T and LHC histograms. For the occupancy histograms, only the middle eight towers can be masked (for both long and short fibres). These settings will be stored in the HCAL configuration database, to be loaded during the configuration stage of a data-taking run. The final stage of pre-processing involves the calculation of the quantities to be histogrammed by the HLX on a per-bunch basis. The partially-processed data are then forwarded to the HLX to be histogrammed and subsequently shipped to the luminosity server over the Ethernet link.

A. *HTR Firmware*

The first stage of processing carried out in the HTR involves masking physical towers that do not operate as expected (e.g. `stuck towers'), and would therefore reduce the accuracy of the luminosity measurement. The data are then processed according to the type of histogram being produced, resulting in an E_T measurement and seven occupancy-style measurements. They are then forwarded to the HLX for further processing. It should be noted that the HTR splits the processing between two FPGAs, corresponding to HF rings 29-34 and 35-41. Therefore two copies of these calculations are sent to the HLX, which are then combined in the HLX firmware to produce the final result. Although all HF channels can be read by the HLX, Monte Carlo studies indicate that the optimal result for occupancy histograms is obtained using just two η rings. Each of the HTR FPGAs sends 12 bits of occupancy data to the HLX. There are three occupancy histograms dedicated to each of the following possible states for each tower: below-lower-threshold, between-thresholds and above-upper-threshold. In addition, a fifteen-bit E_T sum value is sent to the HLX, and a further occupancy-type histogram based on all thirteen HF η rings for use by the LHC.

B. HLX Firmware

The HLX firmware comprises two key components: the first is a histogrammer module that takes 40MHz data from the HTR FPGAs and combines them to produce the eight histograms. The second component is the Ethernet core that packages the data and ships it to the luminosity server using UDP. The firmware in the HLX takes the partially processed data from the HTR and histograms it over a complete orbit, thereby providing a per-bunch representation of luminosity, including empty bunches. The method used to achieve this efficiently in the FPGA involves taking advantage of the dual-port block RAMs in Xilinx Virtex devices. The output of the second port of the block RAM is fed back into the input of the first port of the block RAM. By setting the address of the output port to one greater than the input port, one can compensate for the register latency on the output and simply feed data in for multiple orbits and accumulate with minimal logic overhead. One of the requirements of the luminosity monitor is that it performs dead-time-free readout with no delay in the accumulation of data between each set of histograms. In order to achieve this there are two copies of each histogrammer, one of which is read via Ethernet while the other one is accumulating. The multiplexing of the two cores is managed by a small Finite State Machine (FSM). Partial bunch readout is also possible, allowing diagnostic testing or fast readout of a partial orbit, as opposed to the slower readout of a full orbit's worth of data. After accumulating for a programmable number of orbits, the data are forwarded to the Ethernet core. It is based on the Opencores [3] 10/100Mb Ethernet MAC operating in full-duplex mode. Controlling this are two FSMs that implement a full UDP/IP stack and Address Resolution Protocol (ARP). The latter is not always necessary when transmitting directly to a PC, however many devices (including switches) expect to be able to query the MAC address associated with a corresponding IP. Failure to implement at least these two protocols can prevent the network interface from operating correctly. The Ethernet core has a streaming First In First Out (FIFO) interface and automatically appends an 8-bit CRC at the end of each packet to permit error detection on the readout PC. As the data payload in a given histogram is generally greater than the maximum size of an Ethernet packet (approximately 1500 bytes), it is necessary to transmit each histogram in several packets. A header is appended to the data payload providing information about the type and part of the histogram being transmitted. These data are received by the luminosity readout PC and recombined to form the full set of eight histograms. They are transmitted from each HLX once approximately every 0.37s, which is safely within the 1.45s worst-case histogram overflow time. The eight sets of histograms comprise about 70KB of data, which is transmitted at a rate of approximately 1.6Mb/s per HLX to the Ethernet switch that aggregates the data from all the boards on the luminosity readout PC.

IV. ONLINE SOFTWARE

Each packet of luminosity data are self-contained, including markers for what histogram the payload data corresponds to. Processing of HLX data is split into several classes as shown in figure 6.

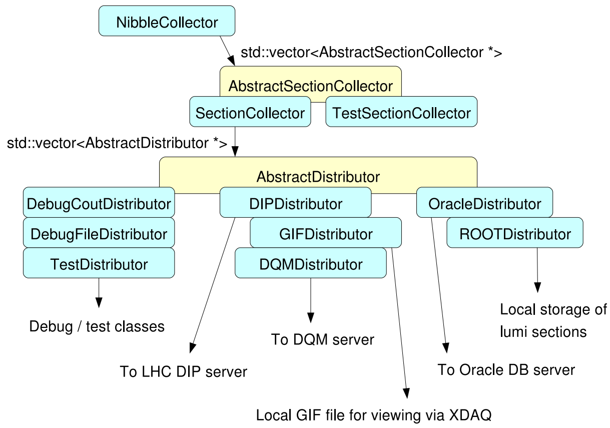

Figure 6: Software class hierarchy in the HLX readout framework.

The NibbleCollector class is used for basic processing and packet identification. Once this has been achieved the data are forwarded to the SectionCollector class, where they are aggregated into ‘luminosity sections’, or multiples of luminosity nibbles. Once the desired number of nibbles has been collected, the data are forwarded to the distributors, which output the data to various monitoring systems, including CMSSW DQM, Oracle DB, ROOT, GIF, LHC DIP, std::cout and flat file. There are also various test classes which can be used to verify both the readout software and the data received from an HLX when operating in test pattern mode. Every distributor possesses its own thread and readout buffer, making the readout system immune to fluctuations in the processing time taken by each distributor.

A. The Nibble Collector

This class creates a UDP listener connection on port 21306 and waits to receive packets from the HLXs. The data are received using an independent real-time worker thread which copies the data into a circular buffer. This increases the load on the CPU but isolates the reception of data from fluctuations in processor usage further up the processing chain. It also allows one to monitor the buffer to see if it overflows, indicating packet loss and an overloaded readout PC. The first stage of processing involves validating the HLX data by computing a CRC-8 checksum over the received data, and comparing it with one embedded in the data. This is carried out using a pre-computed CRC look-up table, which is necessary in order to minimise the load on the PC for checksum calculation. If the computed CRC does not match the one sent in the data payload, the packet is discarded and an error counter is incremented. Upon receiving a packet with the correct CRC, the packet header is analysed to determine the histogram type, segment and the HLX that it originated from. The readout software realigns fragmented data by using the 'StartOrbit' identifier in the header, which should match for all corresponding HLX packets. The 'StartBunch' identifier is used to rebuild fragmented histogram data. Upon completion of a luminosity nibble, the data for all the histograms from a given HLX is forwarded to the SectionCollector class. In order to produce a normalised result, the occupancy nibbles are only forwarded if all six occupancy histograms are received. It should be noted that the NibbleCollector may forward data to multiple SectionCollector instances, in particular if one wishes to produce luminosity sections with different accumulation periods.

B. The Section Collector

The section collector accumulates a programmable number of nibbles before forwarding them to a std::vector of distributors. It is also responsible for some further checks on the data to ensure it is complete. The section collector is updated using a ProcessNibble() function called by the NibbleCollector, during which it is presented with the next nibble available for accumulation. The section collectors also incorporate a worker thread which manages forwarding of the data to the distributors. In this way the readout system is protected from stalling due to a single failed distributor.

C. The Distributors

Various distributors exist depending on the application to which the system is being used. Every distributor must implement a ProcessSection() function, called by the SectionCollector class every time a new luminosity section is available. The distributor then acts to reformat the data and forward it to the various readout systems. Three test classes are available for use as debugging tools. These are:

- **DebugCoutDistributor** – outputs luminosity section header information to std::cout whenever a new luminosity section is submitted by the section collector.
- **DebugFileDistributor** – outputs both the header information and the data themselves to a file in ASCII format.
- **Test Distributor** – can be used to validate the NibbleCollector and SectionCollector when the HLXs are operating in test mode.

There are three distributor classes for use with short section collectors (of the order of a few thousand LHC orbits). These are:

- **DIPDistributor** – publishes the LHC histogram via the DIP [4] system. Updates occur at a rate of approximately one Hertz.
- **GIFDistributor** – creates a temporary GIF file from a ROOT TH1D histogram, which is imported into the XDAQ [5] readout web page to assist in debugging the system.
- **DQMDistributor** – this is in fact based on two separate modules: one of them is a TCPDistributor that forwards the section collector data using a network loopback to a CMSSW DQM application. This avoids cross-compilation issues between CMSSW, XDAQ, Oracle, DIP and the other libraries used in the readout system.

Finally, there are two distributors for use with long section collectors (several hundred thousand LHC orbits). These are:

- **OracleDistributor** – submits the luminosity section data to and Oracle database.
- **ROOTDistributor** – saves the luminosity section data in a time-stamped ROOT file on the luminosity readout server.

D. Standard Implementation

In normal use, the luminosity readout has one 3-nibble SectionCollector instance and one 256-nibble instance, with different distributors attached to each instance. The short section distributors are the DIP, GIF and DQM distributors, while the ROOTDistributor and OracleDistributor are attached to the long section collector. Detailed luminosity information is collected online and propagated to the offline Tier 0 data centre for insertion into the event stream. A summary for each luminosity section, as well as the bunch-by-bunch luminosity data, are recorded in an online luminosity database by the OracleDistributor. Offline, the Tier 0 work-flow manager makes a request for each luminosity section through a server which resides on the edge of the online network, but that is also visible to the offline systems. Accessing the luminosity data in this manner ensures that there is minimal latency between the appearance of detector data offline and the availability of the corresponding luminosity information. The Tier 0 data flow includes as its first step a re-packing procedure that maps the online events streamed by the storage manager into the offline files. This is the ideal moment to insert the luminosity data into the event. This task is performed using the CMS offline software framework. The luminosity block includes the instantaneous luminosity summed over all bunch crossings, the bunch-by-bunch luminosity data, and estimates of the efficiency and quality of the measurements. Each luminosity block object in the data file is common to all data events taken within the time period of the associated luminosity section (approximately 93 seconds). The entire chain from online database to Tier 0 re-packing was built and tested in early August 2007, and was proven to operate correctly. Simulated luminosity data was loaded into the online database. The offline work-flow was then initiated and the requests for the luminosity data were made to the luminosity gateway server. The delivered data was formatted and read by the CMS re-packer job, and inserted into the data stream. The output files were examined and the luminosity information was verified to be the same as that originally available in the online database. It is anticipated that the luminosity information will also be available offline in a form outside of the event itself. In particular, the raw luminosity data will be recorded online into ROOT files, which will be transferred at the end of each run to an offline repository. These files will be studied and summarised into an offline luminosity database that will be used in conjunction with the offline data book-keeping system. CMS physicists will be able to use this information to associate integrated luminosities with the data sets they are analysing.

V. System Tests

Initial testing of the HLX system focussed on validation of the readout chain in the CMS counting room. The first full test of the system occurred in August 2007, during a 'global run'. These runs involve the integration of the various DAQ and trigger systems, as well as the luminosity monitor, and allowed real data to be recorded from the HF+ detector. An example of this is shown in figure 7, which shows a histogram of the below-threshold occupancy for an individual HF+ wedge. As the detector was simply powered but received no real input signal, the data represents the noise in the readout system. The peaks observed show that the readout chain is functioning correctly. Furthermore the E_T sum measurements showed the correct approximate value for the noise (a few ADC counts per channel). The whole HF+ system (18 HLXs) has also been tested at full rate using the RCMS control system, showing that both the configuration and readout are functioning properly. The link error rate from the HLXs to the readout PC is estimated to be less than one part in 10^{12} at a 95% confidence level.

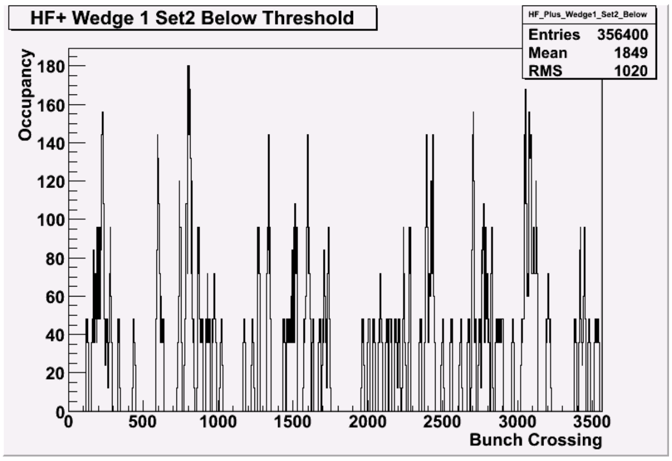

Figure 7: An example occupancy histogram for a single HF+ wedge. The peaks represent noise in the HF+ readout system.

VI. Future Plans

In addition to loading the luminosity data into the event file, we also plan to load a summary of the luminosity data and L1 scalers into the event stream at a rate of approximately one Hertz [2]. We plan to use a GIII PCI card, mounted in a dedicated PC, which will provide an S-LINK data output via a PMC mezzanine card. Loading the luminosity value and the L1 scalers will assist in the debugging of the detector data and beam performance studies, which will be crucial during the commissioning phase.

An alternative luminosity monitoring method is also being developed in collaboration with Princeton, Rutgers and U.C. Davis. The Pixel Luminosity Telescope (PLT) operates in the high η region by counting forward tracks. As with the HLX system, is a measure of relative luminosity; however it should prove more immune to beam background effects, and is anticipated to measure the luminosity with less than 1% error.

VII. Acknowledgements

The work described in this paper was carried out in collaboration with: Princeton University, The University of Maryland, U.C. Davis, Rutgers University and The Fermi National Accelerator Laboratory.

VIII. References

[1] N. Adam, V. Halyo, J. Jones, D. Marlow, S. Schnetzer, Y. Guo, L. Lueking, K. Maeshima, W. Badgett, L. Sexton-Kennedy, I. M. De Abril, M. M. De Abril, The CMS Luminosity System, CMS IN-2007/030

[2] W. Badgett, V. Halyo, K. Maeshima, D. Marlow, Putting Luminosity and Trigger Scaler Information into the Event Stream, CMS IN-2007/025

[3] Opencores Organisation, http://www.opencores.org/

[4] ITCOBE DIP Software Page,

http://itcobe.web.cern.ch/itcobe/Projects/Framework/Download/Components/DIP/

[5] J. Gutleber, L. Orsini, The XDAQ Framework,

http://xdaq.web.cern.ch/xdaq/