Recent Results on the Performance of the CMS Tracker Readout System

J. Fulcher [1], L. Mirabito [4], J. Coughlan [2], R. Bainbridge [1], I. Church [2], C. Foudas [1] G. Hall [1], M. Pesaresi [1], M. Ageron [4], G. Rogers [2], S. Taghavi [2], I.R. Tomalin [2], O. Zorba [1] , F Drouhin [6], L. Gross [3], D. Vintache [3], G Baulieu [4], A Giassi [7], R. Arcidiacono [5]

[1] Imperial College, London, UK. [2] CCLRC Rutherford Appleton Laboratory, Oxfordshire, UK, [3] Strasbourg University, France. [4] Institut de Physique Nucléaire, Lyon, France. [5] CERN, [6] Mulhouse, France, [6] Pisa University, Italy

Abstract

The CMS Silicon Tracker comprises a complicated set of hardware and software components that have been thoroughly tested at CERN before final integration of the Tracker. A vertical slice of the full readout chain has been operated under near-final conditions. In the absence of the tracker front-end modules, simulated events have been created within the FED (Front End Driver) and used to test the readout reliability and efficiency of the final DAQ (Data Acquisition). The data are sent over the S-Link 64 bit links to the FRL(Fast Readout Link) modules at rates in excess of 200 MBytes/s per FED depending on setup and conditions. The current tracker DAQ is fully based on the CMS communication and acquisition tool XDAQ. This paper discusses setup and results of a vertical slice of the full Tracker final readout system comprising 2 full crates of FEDs, 30 in total, read out through 1 full crate of final FRL modules. This test is to complement previous tests done at Imperial College[3] taking them to the next level in order to prove that a complete crate of FRLs using the final DAQ system, including all subcomponents of the final system both software and hardware with the exception of the detector modules themselves, is capable of sustained readout at the desired rates and occupancy of the CMS Tracker. Simulated data are created with varying hit occupancy (1-10%) and Poisson distributed trigger rates (<200KHz) and the resulting behaviour of the system is recorded. Data illustrating the performance of the system and data readout are presented.

I. INTRODUCTION

The CMS Silicon Tracker will produce large amounts of data, which must be read out by the Front End Drivers (FEDs) [3][4], which are 9U 400mm VME64x cards that process the raw data from a subset of 192 APV25[2] silicon readout ASICs, equivalent to 0.2% of the tracker. After multiplexing and streaming, the data from the tracker are routed via analogue optical links to the FEDs. 96 optical channels are then digitised to 10bit precision at 40MHz and processed in large FPGAs, before being collated into events and sent to the CMS DAQ via either VME or the SLINK-64 protocol.

Under final running conditions the SLINK-64 protocol must be used to enable a data throughput of up to, and in some cases where only one FED is connected per FRL exceeding, 200 Mbytes/s per FED. It is clear therefore that such a system must be well tested before final assembly at Point 5 on the LHC ring thus ensuring that the final operation of the system is certified. However, during the testing and assembly stages in 2006 there are not enough Tracker front-end modules available in order to drive a large enough number of FEDS to make a true vertical slice test. Nominally a vertical slice would consist of 1 whole crate of FRLs, which corresponds to two whole crates of FEDs ~ 30 in total. Therefore another method of testing the readout system without the need for the detector modules is required. This is achieved by creating Pseudo-Random fake events within the front-end FPGAs of the FED, which are injected directly into the FED exactly where the optical data samples would arrive from the final system.

In order to qualify the hardware readout system and operation of the DAQ, the system was set up in building 904 on the CERN Prevessin site in France. The system is capable of creating variable event occupancies in the front end of the FED without the need for the actual detector modules to create the event frames from the APV25 front-end readout chips. The "fake events" which are created in the FED have the capability to be set to operate at any occupancy between 0 and 100% and also include a pseudo random adjustment to the pedestals of the fake APV frames, in order to simulate the random nature of the hits in the front-end silicon detector modules.

The main purpose of this test is to qualify the functionality of the entire trigger and readout chain from the LTC, TTCci, TTCex, TTCoc, APVE, FMM, FRL, FED, Transition Card, Slink Sender Card and DAQ software. Operating the system under various conditions by varying the strip hit occupancy and therefore data throughput. In doing so one can investigate the limiting factors of the system to gain a better understanding of how the final tracker will perform.

II. TRACKER FED FAKE EVENTS

By exploiting some of the remaining block memory within the delay FPGAs on the front end of the FED, it has been possible to set aside enough space to program individually each optical channel with a unique multiplexed APV25 frame. The fake events that are produced within the FED are exact replicas of the data frames that would normally arrive over the optical analogue links. The frames are programmed into a block RAM within the delay FPGAs

using the same method over VME as is utilized to program all other configurable registers within the FED.

In order to accurately simulate more realistic final conditions, a pseudo random number generator has been implemented to augment the values set within the stored data frame. The magnitude of this random augmentation of frame samples is programmatically adjustable to have a range of between 0-1 up to 0-256 ADC counts. This factor can also be applied on a channel-by-channel basis. The value produced by the random number generator is then summed with the value stored in the block ram for the sample being input into the front end FPGA on that clock cycle. Between each sample the random number generator is cycled and a new number produced to be summed with the next sample in the APV data frame and so on, thus a sample by sample noise is introduced over the whole APV frame.
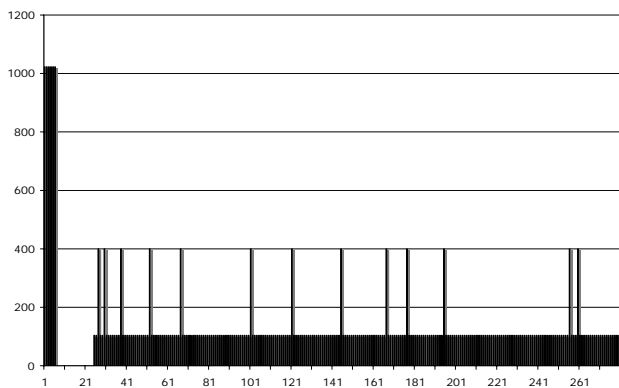

Figure 1: APV Fake Data Frame With 5% Occupancy

In this test the range of the noise was set to between 0 and 25 ADC counts. The strip hit values were set to 400 ADC counts. The hit threshold was set to 25 ADC counts. In order to fully test the functionality of the FED all frames were created with a base offset pedestal of 104. The same value was then set as the pedestal for subtraction in the FED and the FEDs were run in Zero Suppression mode. Figure 1 shows an example APV fake data frame with a strip-hit occupancy of 5% corresponding to 13 hits out of the 256 strips. It just so happens that an occupancy of 5% corresponds to a condition in the system where there is a fine balance between the buffer occupancy within the APV25 front end chips and that of the FEDs. At this occupancy with the trigger set to 100kHz one clearly sees backpressure being exerted by both buffers. This is merely a point of interest and not one for further discussion.

III. TEST SETUP

The test setup comprises a complete vertical slice of the full final CMS Tracker trigger and DAQ system excluding the detector silicon strip modules, analogue optical link and the back end readout link of the FRL for the final DAQ. For the purpose of this test the data were read out over the SLink, the CRC value was calculated for each event and

each FED, checked with the CRC stored within the data and the result was logged. Data were assumed to be ok given the corroboration of CRC checksums. Note that it is possible that limitations on the speed of this final section of the DAQ could reduce the achievable rates measured in this test.

Figure 2 is a schematic representation of the final test setup. For more detail on this system see reference [5]. The basic configuration of the system and interconnections can be seen in simplified form. Three VME crates are used in this system all of which are controlled by rack mounted PCs, running the Linux operating system, via the CAEN PCI to VME bus controller[7]. The software installed on all machines contains the full installation of XDAQ[1] and the Tracker specific software packages [6][8] for online Tracker commissioning and data analysis tasks. The DAQ PC controls a compact PCI crate and runs the specialised software developed to control and readout the FRLs[10]. The trigger is controlled by the final CMS system comprising LTC, TTCci, TTCex, TTCco and the Tracker Specific APVE.
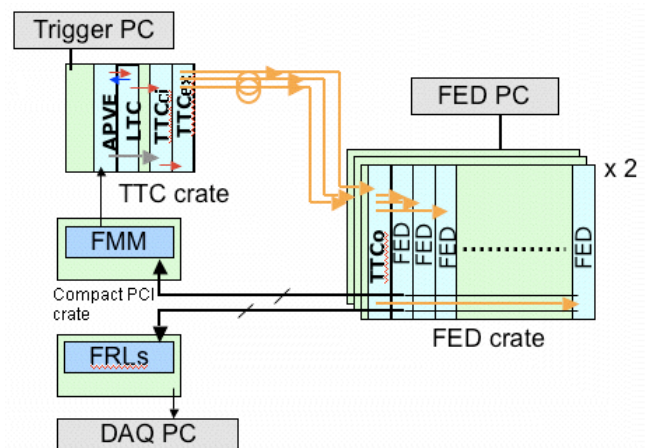

Figure 2: Test Setup

Back-pressure for this system is achieved using the final FFM feedback loop that is in place to prevent overflowing of buffers on the FED and also to communicate run time statuses of the FEDs in the system. The statuses can be logged by the FMM modules thus it is possible to record the history of state changes on the individual FMM channels. The complete set of statuses from all the FEDs are then combined in a logical OR to produce a single signal that is fed into the APVE module where this status is then combined in a further logical OR with the status of the APV emulator. The result of these merged signals is a global signal that is then fed to the LTC trigger card where upon receipt of a status other than READY, either WARN or BUSY, the trigger controller will then interrupt the flow of triggers in order to reduce the occupancy of buffers in the system to a safe level at which point the FEDs and/or APV emulator will then de-assert BUSY or WARN thus allowing triggers to resume.

All of the 30 FEDs are configured and controlled by the CMS Tracker DAQ online software with the use of the

Fed9USoftware[9] package developed by Imperial College London. Each FED is programmed with the same fake event of a specific representative detector occupancy and on receipt of a trigger the FEDs process the fake APV frame and output the Zero Suppressed data buffer containing the hit information for the APV frame.

IV. RESULTS

Results were taken over a full week of running and in total more than 30 billion events were read out from 30 FEDs. The system was run with pseudo random Poisson generated triggers with a mean rate of 140kHz. Note that this frequency drops when the backpressure system must assert BUSY and thus veto the trigger. Figure 3 shows an example distribution of triggers produced by the Poisson generator of the LTC when running with mean rate of 140 kHz. These data were taken directly from the trigger analyzer on the APVE card during one run with hit occupancy of 1%, thus resulting in no rate reduction due to backpressure.
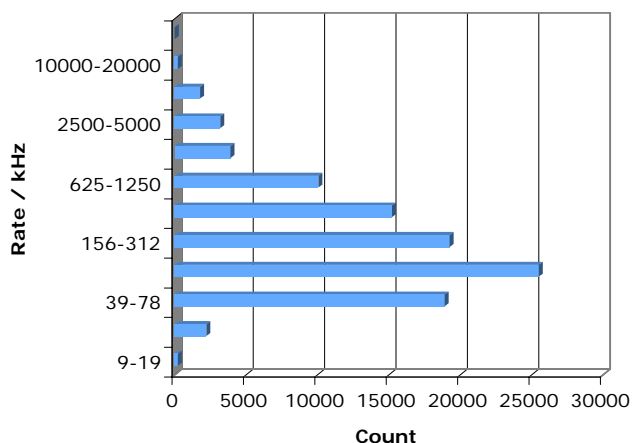

Figure 3: Trigger distribution for 140kHz mean rate

In an attempt to demonstrate what the results represent when considering the final CMS Tracker readout expectations the data are displayed in terms of percentage data loss calculated from the ratio of measured trigger rate to 100kHz, the published desired mean trigger rate of CMS. No data is actually lost, data from all triggers were received and no corruption or loss data was seen.

The FEDs were setup to run in Zero Suppression mode with pedestal subtraction and common mode subtraction all activated. The pedestals were set to 104 ADC counts as described in section III. Two sets of data were taken for two different modes of operation of the FED. Figure 4 shows the percentage data loss for full debug header mode, which is a verbose mode of operation giving more information about the status of each event.
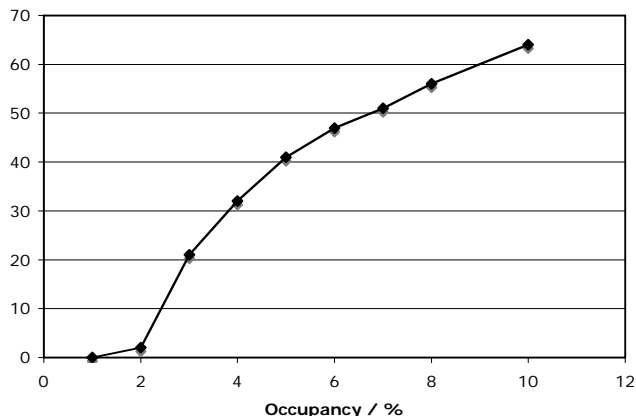

Figure 4: % Data Loss for Full Debug Data Frame at 100kHz

This plot shows that for low occupancy of 1-2% the system ran with practically no reduction in trigger rate. It is important to note that a large majority of the CMS Tracker will run under such conditions in the LHC. As we reach the limit of buffer occupancy in the FED and APV at 3% occupancy we find that the data loss has a sharp increase up to 20%. At a higher occupancy of 10% there was a 64% reduction in trigger rate therefore it is clear that this system is extremely sensitive to the hit occupancy within the silicon strips.
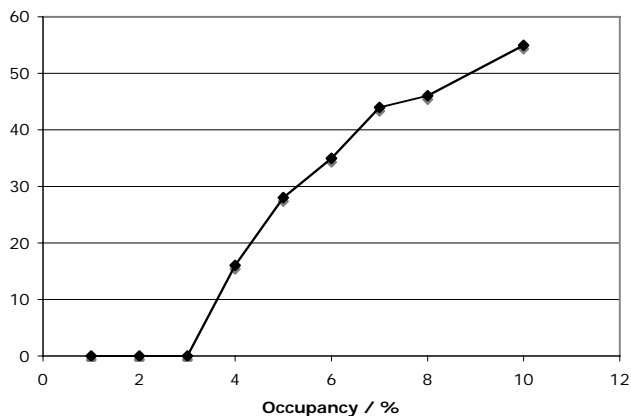

Figure 5: % Data Loss for Compressed Data Frame at 100kHz

For those innermost areas of the Tracker where the occupancy is expected to be the highest it is still encouraging to look at the results from Figure 5 where the data format is now set to compressed mode. With the reduction in header information and overall data packet size for this mode, one is able to readout data with hit occupancy of 3% with no reduction in trigger rate. Again there is a sharp rise in the data loss at 4% occupancy up to 17%. It is clear that this reduced data packet size will be essential when reading out the innermost layers of the Tracker in order to keep the sustainable trigger rates as high as possible.

V. CONCLUSIONS

The results of this test have shown that the system is stable for periods of time over many hours when running at 150KHz Poisson distributed triggers, the back pressure from the APVE and FED, via the FMM, works perfectly to throttle the trigger rate down to a sustained rate of 72Khz when operating the FED at 5% occupancy. At lower occupancies, 1-3%, more representative of nominal data taking conditions in the final Tracker, the system runs without interruption at trigger rates around 100 KHz sustained. Note that these measurements were conducted without final shipping of the data for processing and analysis, it is likely that some delay may be introduced into the readout of the data from FRLs due to slower final shipment of the data. However, this is Software and PC processor power dependent and therefore not without the possibility for future improvements. The results in this paper are the best possible achievable readout rates without further reduction due to data processing.

VI. REFERENCES

[1] J.Gutleber et al., "Architectural Software Support for Processing Clusters", IEEE International Conference on Cluster Computing (Cluster 2000), November 28 - December 2, 2000, Chemnitz, Germany, IEEE Conference Proceedings http://xdaq.web.cern.ch/xdaq/
[2] M.J.French et al. "Design and results from the APV25, a deep sb-micron fron-end chip for the CMS tracker", Ncl. Instr. And Meth. A466 (2001) 359-365
[3] G. Iles et al. "*Performance of the CMS Silicon Tracker Front-End Driver*" LECC Proceedings, Boston, 2004
[4] J. Coughlan et al. "*The Manufacture of the CMS Tracker Front-End Driver*" LECC Proceedings, Boston, 2004
[5] CMS Tracker Technical Design Report
[6] L.Mirabito et al., "Tracker data acquisition for beamtest and integration", CMS IN 2003/021
[7] CAEN VME Bus Adaptor Crate Controller
[8]https://uimon.cern.ch/twiki/bin/view/CMS/TrackerOnlineSoftware
[9] http://fed9u.web.cern.ch/fed9u/
[10]http://cms-frl.home.cern.ch/cms-frl/