FERMILAB-CONF-12-274-CD

An Application Using MicroTCA for Real-Time Event Assembly

Ryan A. Rivera, Alan Prosser, Lorenzo Uplegger, Jeff Andresen, Simon Kwan

Abstract – **The Electronic Systems Engineering Department of the Computing Sector at the Fermi National Accelerator Laboratory has undertaken the effort of designing an AMC that meets the specifications within the MicroTCA framework. The application chosen to demonstrate the hardware is the real-time event assembly of data taken by a particle tracking pixel telescope. In the past, the telescope would push all of its data to a PC where the data was stored to disk. Then event assembly, geometry inference, and particle tracking were all done at a later time. This approach made it difficult to efficiently assess the quality of the data as it was being taken – at times, resulting in wasted test beam time. Now, we can insert in the data path, between the telescope and the PC, a commercial MicroTCA crate housing our AMC. The AMC receives, buffers, and processes the data from the tracking telescope and transmits complete, assembled events to the PC in real-time. In this paper, we report on the design approach and the results achieved when the MicroTCA hardware was employed for the first time during a test beam run at the Fermi Test Beam Facility in 2012.**

I. INTRODUCTION

In recent years there has been a growing desire within the high energy physics (HEP) community to advance the predominant electronics standard beyond Versa Module European (VME) to a more modern standard. As this new standard will likely become the de facto standard for impending data acquisition upgrades for the Large Hadron Collider and future experiments, it must be characterized by high availability, high reliability, and scalability. One of the candidates for such a replacement of VME is the Telecommunications Computing Architecture (TCA).

The Electronic Systems Engineering Department of the Computing Sector at the Fermi National Accelerator Laboratory (FNAL) provides innovative solutions to the data acquisition and trigger system challenges faced by HEP. Emerging commercial standards like TCA, which is driven by developments in the telecommunications industry, offer the promise of modular solutions that meet the requirements of high data volume HEP experiments. The effort described in this paper represents an exploration of the use of components designed to these standards for applications in HEP, including future data acquisition and trigger systems.

We selected real-time event assembly for our TCA application because the department had already developed to maturity a tracking telescope system [1] that included event assembly. In the past, the event assembly and further analysis were completed off-line after storing the raw tracking data to disk on a PC. With the addition of real-time event assembly, we can shrink the data footprint on disk and streamline event processing.

The Micro Telecommunications Computing Architecture (MicroTCA or µTCA) [2] was chosen for this application as the more affordable and appropriately-scaled implementation of TCA, versus the Advanced Telecommunications Computing Architecture (ATCA) [3].

II. SYSTEM OVERVIEW

The pixel tracking telescope is located along the particle beam line within the beam enclosure at the Fermi Test Beam Facility at FNAL as shown in Fig. 1. The active area of the telescope is made up of ten pixel planes split among three data acquisition stations that generate data tagged with a trigger number each time a particle passes through the pixel planes and a scintillator downstream of the telescope. The tagged data from the ten pixel planes are accumulated at the three stations.

The data acquisition station is a Compact And Programmable daTa Acquisition Node (CAPTAN) [4, 5, 6]. Each CAPTAN has a gigabit Ethernet link. There is a gigabit Ethernet switch that negotiates the network traffic between the three CAPTAN stations, the MicroTCA crate, and the PC in the control room.

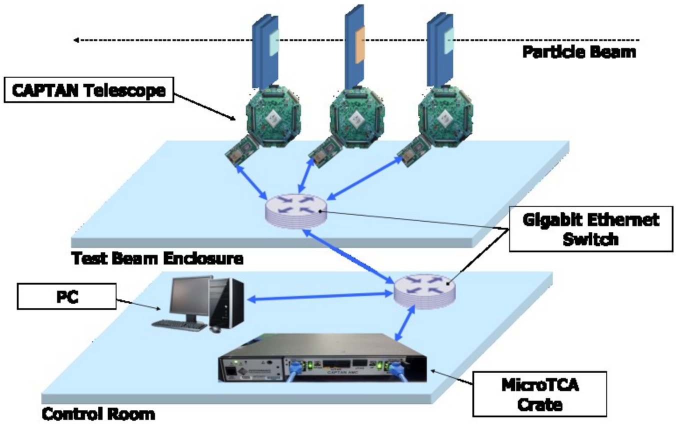

Fig. 1. Above is the layout of the pixel telescope system. The MicroTCA-based event assembler is shown in a gigabit Ethernet network along with the CAPTAN telescope and the data storage PC.

Prior to the MicroTCA crate conducting the real-time event assembly, the raw data at the CAPTAN stations were pushed to the PC - where the three individual data streams were stored to disk in separate directories. Then an off-line application

---

Manuscript received June 11, 2012. This work was supported by Fermi National Accelerator Laboratory operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

R. A. Rivera is with Fermi National Accelerator Laboratory, Batavia, IL 60510 USA (e-mail: rrivera@fnal.gov).

scanned the files in the separate directories, merged the station data into events based on matching trigger number, and saved the resulting merged data file to disk.

With the addition of the MicroTCA crate, the event building can occur in real-time thus bypassing the need for the intermediate storage of the raw data. The PC receives only completed events for storage, and the analysis of the runs can begin with geometry inference, followed by track reconstruction.

III. REAL-TIME EVENT ASSEMBLY

The MicroTCA card at the heart of the event assembly is designed by the Electronic Systems Engineering Department at FNAL. It meets the physical specifications for a full-size, double width Advanced Mezzanine Card (AMC) [7] and is shown in Fig. 2. It relies on two daughter cards to provide most of the functionality: one daughter card to conduct the Module Management Controller (MMC) operations that are required under the MicroTCA standard, and one to conduct the event assembly.

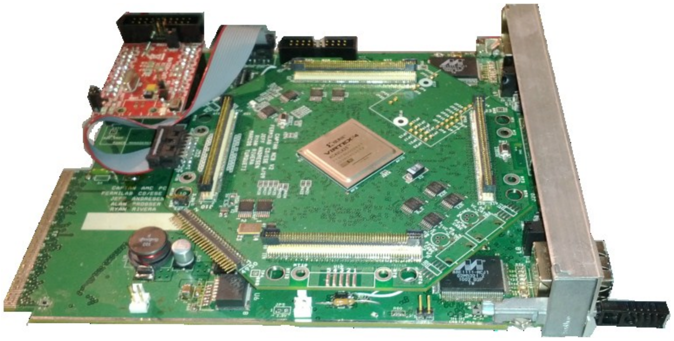

Fig. 2. The AMC designed by the Electronic Systems Engineering Department at FNAL is pictured above. The MMC daughter card is at the top-left corner of the board. The event assembly daughter card is the octagonal board with an FPGA at its center.

The MMC daughter card is the LPC-H2148 from Olimex. It utilizes the LPC2148 microcontroller from NXP Semiconductors (founded by Phillips) and is programmed with C code. The MMC orchestrates communication with the shelf manager of the MicroTCA crate to request power, respond to status requests, manage on-board resources, and access crate resources.

The event assembly daughter card is an additional CAPTAN – identical to hardware used in the tracking telescope. The CAPTAN has an LX25 Virtex-4 FPGA from Xilinx at its center with four bus connectors around its perimeter for vertical stacking. We already were in possession of CAPTAN boards, so compatible connectors were designed into the AMC to leverage the hardware already on hand. The CAPTAN on-board FPGA affords the processing power and memory for the event assembly algorithm.

IV. RESULTS

Due to the 2012 accelerator complex shutdown at Fermilab, the Fermi Test Beam Facility beam time was in high demand when this project was ready to acquire data. However, we were able to take several good runs while running parasitically with another test beam experiment. These runs lent confidence to our real-time event assembly algorithm.

In order to assess the event assembly algorithm, the data from the telescope was both forwarded to the PC, as had been done in the past, and to the AMC in the MicroTCA crate. Both the AMC and PC received exact replicas of the telescope data. The merged file from the PC software application was then crosschecked with the file of received events from the AMC.

Results from two runs, where complete agreement was found, are presented here. In Fig. 3, the assembled events from *Run 1112* are shown. In Fig. 4, the assembled events from *Run 1117* are shown. *Run 1112* had 1,587 recorded events. *Run 1117* had 3,378 recorded events.

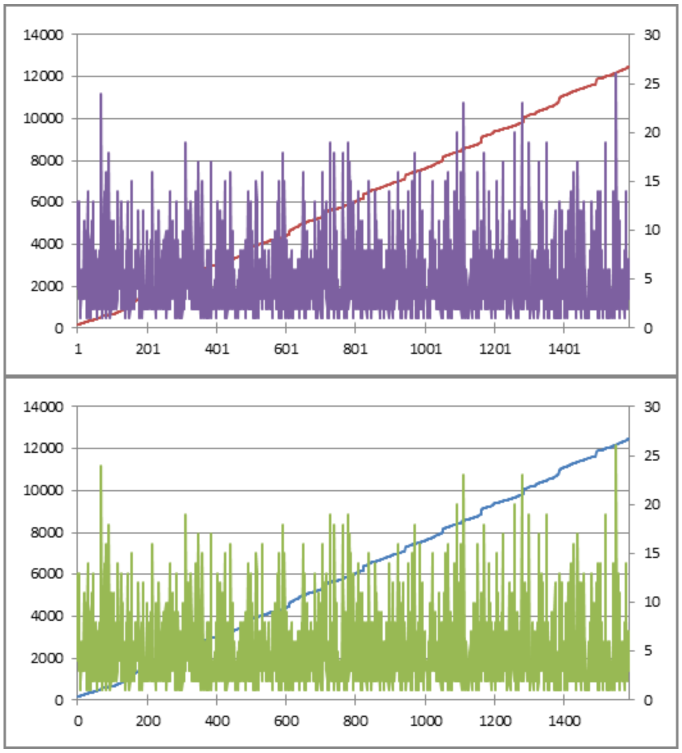

Fig. 3. On both charts above there are two data series from *Run 1112*. The monotonically increasing data series is the scintillator trigger counter (left Y-axis) and the other data series is the number of pixel hits (right Y-axis) for an event. The X-axis is the unique event number assigned to a specific scintillator trigger and its associated pixel hits. The upper chart is for the AMC hardware, and the lower chart is for the off-line PC software. 1,587 events were in complete agreement for this run.

In both Fig. 3 and Fig. 4, the upper chart contains the AMC hardware results and the lower chart contains the off-line PC software results. Complete agreement is achieved if the data of the two charts exactly match. Each chart is itself comprised of two data series. Both data series have an entry for each unique, integer event number on the X-axis.

The monotonically increasing data series is the scintillator trigger counter with values mapping to the left Y-axis. The scintillator counter increases each time the scintillator fires, which occurs due to noise or a particle depositing energy as the particle passes through the scintillating material.

The other data series is the combined hit count from all ten of the telescope pixel detectors for each assembled event, and has values mapping to the right Y-axis.

The telescope had ten pixel planes of varying size and orientation. Under ideal conditions, one would expect that each time the scintillator fires, the corresponding particle traces out a track by leaving charge in a pixel or group of pixels in each of the ten planes. However, for these runs, the telescope was situated off-axis with respect to the beam center while running parasitically with the primary test beam experiment. So the scintillator is impinged by only a small percentage of the total beam – essentially the stray particles. And these stray particles are not likely to pass nicely through all ten pixel planes and the scintillator. This explains why the average hit value is less than the idealized ten hits.

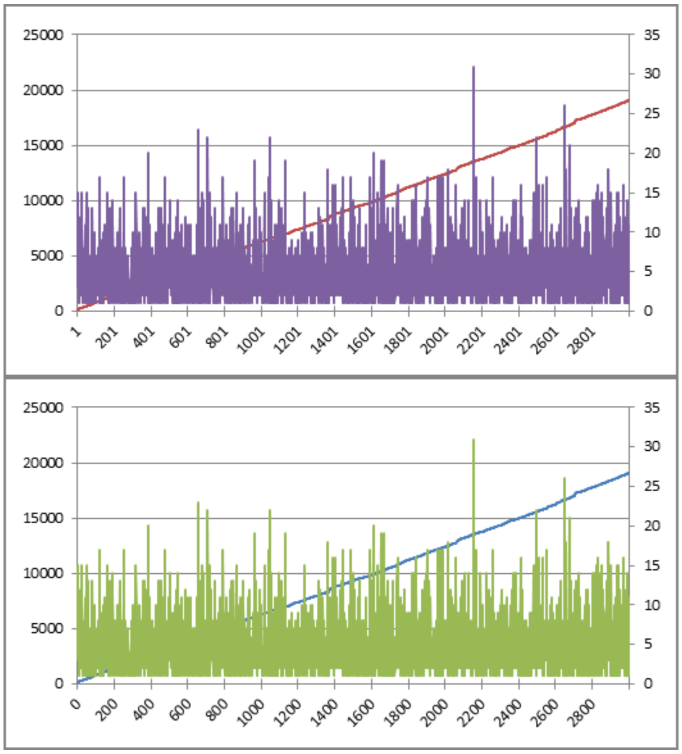

Fig. 4. On both charts above there are two data series from *Run 1117*. The monotonically increasing data series is the scintillator trigger counter (left Y-axis) and the other data series is the number of pixel hits (right Y-axis) for an event. The X-axis is the unique event number assigned to a specific scintillator trigger and its associated pixel hits. The upper chart is for the AMC hardware, and the lower chart is for the off-line PC software. 3,378 events were in complete agreement for this run.

The scintillator used to generate the trigger was also less than ideal. Even when no beam was present it still had a substantial firing rate. Under normal running conditions, two or three scintillators would be used with a trigger generated by their coincidence. But because of the off-axis running conditions, single coincidence was used for wider acceptance. Due to the scintillator noise and time structure of the beam delivery, which was beam for 4 seconds followed by no beam for 60 seconds, small vertical discontinuities can be seen in the scintillator trigger counter data series.

In Fig. 5 the number of telescope pixel planes present is shown for each assembled event to give an idea of plane occupancy during the runs. Although rare, there are events in which all 10 planes report a hit. On the same chart, the number of overflow errors is shown coinciding with each event. These errors would be due to an internal memory overflow detected in the FPGA configware. During *Run 1117* there were no overflow errors detected for the duration of the 3,378 assembled events.

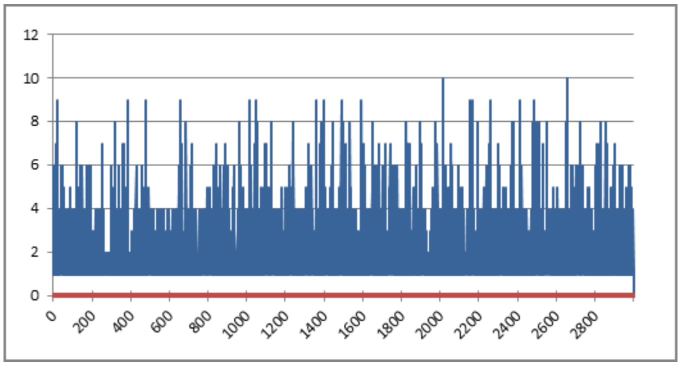

Fig. 5. The chart above has two data series from *Run 1117*. The unique event numbers on the X-axis are identical to those found in Fig. 4. The data series that is 0 for every event is the number of memory overflow errors in the configware for each event. The other data series is the number of planes that had data for each event.

One final item of interest from the test beam results is the data footprint comparison for the original off-line event assembly method versus the real-time MicroTCA method.

For *Run 1112*, the off-line assembly software total data footprint was 88 KB + (3 • 1,954 KB) = 5,950 KB – 88 KB for the final merged, assembled events file and 3 • 1,954 KB for the three separate raw data files. The three raw data files are separate because the telescope data is transmitted to the PC from three separate gigabit Ethernet interfaces. Each interface is controlled by its own software process which fills the file while receiving data. For disk throughput considerations and because the amount of data arriving to these software processes is unknown a priori, these files are saved in fixed-size chunks of 2 MB. So this 2 MB defines the minimum raw data file size.

For the same run, the data footprint for data received at the PC from the MicroTCA hardware was 63 KB. Therefore the AMC implementation of real-time event assembly reduced the on-disk data footprint by a factor of 94. Note that the MicroTCA assembled events file (63 KB) is smaller than its off-line counterpart (88 KB) because the off-line file maintains a PC generated 8 byte timestamp for each event. This extra information is not present in the MicroTCA version.

For *Run 1117*, the off-line software data footprint was 174 KB + (3 • 1,954 KB) = 6,036 KB versus the MicroTCA data footprint of 122 KB. Thus, for *Run 1117* the footprint reduction factor was 49.

V. CONCLUSION

With the results attained during this test beam run, it can be concluded that conducting the real-time event assembly of pixel tracking data is possible using a single VIRTEX-4 FPGA in the MicroTCA form factor.

An emerging standard like MicroTCA is well positioned to become the backbone for future HEP experiments due to its growing interest in HEP and the role the telecommunication industry plays pushing TCA toward high availability, high reliability, and scalability. Conducting real-time event reconstruction in the MicroTCA form factor will give the Electronic Systems Engineering Department firm footing to move forward towards the next project milestone, which is real-time particle track reconstruction for the pixel telescope system. This project has yielded valuable experience for our engineers that can be leveraged for the future.

REFERENCES

[1] R. A. Rivera, M. Turqueti, L. Uplegger, "A Telescope Using CMS PSI46 Pixels and the CAPTAN for Acquisition and Control over Gigabit Ethernet," *2009 IEEE Proc. Nuclear Science Symposium*.

[2] PICMG MicroTCA.0 Micro Telecom Computing Architecture Base Specification (and associated subsidiary specifications), Revision 1.0, July 6, 2006. PICMG (www.picmg.org).

[3] PICMG 3.0 Advanced Telecom Computing Architecture Base Specification (and associated subsidiary specifications), Revision 3.0, March 24, 2008. PICMG (www.picmg.org).

[4] M. Turqueti, R. A. Rivera, A. Prosser, J. Andresen, J. Chramowicz, S. Kwan, "CAPTAN: A Hardware Architecture for Integrated Data Acquisition, Control, and Analysis for Detector Development," *2008 IEEE Proc. Nuclear Science Symposium*.

[5] R. A. Rivera, M. Turqueti, A. Prosser, S. Kwan, "A Software Solution for the Control, Acquisition, and Storage of CAPTAN Network Topologies," *2008 IEEE Proc. Nuclear Science Symposium*.

[6] M. Turqueti, R. A. Rivera, A. Prosser, S. Kwan, "A Generic Readout Environment for Prototype Pixel Detectors," *2009 Technology and Instrumentation in Particle Physics*.

[7] PICMG AMC.0 Advanced Mezzanine Card Base Specification (and associated subsidiary specifications), Revision 2.0, November 16, 2006. PICMG (www.picmg.org).