An Asynchronous Level-1 Tracking Trigger for Future LHC Detector Upgrades

A. Madorsky, D. Acosta

University of Florida/Physics, POB 118440, Gainesville, FL, USA, 32611
madorsky@phys.ufl.edu, acosta@phys.ufl.edu

Abstract

We propose consideration of an asynchronous trigger system for future LHC upgrades of the Level-1 trigger. As the beam-crossing frequency increases in a future LHC upgrade (SLHC), and more data is brought into the Level-1 trigger system, the synchronization of the traditional synchronous trigger system composed of hundreds of boards will become even more difficult. To demonstrate the advantages of asynchronous trigger designs, we are developing an upgrade of the front-end trigger electronics of three spare cathode-strip chambers (CSCs) of the CMS Endcap Muon system to perform pattern recognition and bunch-crossing assignment from the anode data at 80 MHz frequency.

Trigger primitives will be transmitted to a newly designed asynchronous track-finding processor that receives data from up to 3 chambers.

I. INTRODUCTION

Current modern collider experiments, such as the upgraded Tevatron experiments and the LHC experiments currently under construction, operate (or will operate) with a pipelined, synchronous Level-1 trigger system. The principal advantage of such a system is that this first level of filtering operates in a straight-forward manner and is essentially dead-time free: all trigger data move in lockstep with the beam-crossing clock through the trigger decision chain, and after a fixed amount of latency, the global trigger decision arrives at the front-end boards and the detector data, which are held in storage pipelines, are either digitized or discarded. No time markers are required for the data, as the data is synchronized to the beam-crossing clock. The disadvantage of such a system is that this synchronicity must be maintained throughout the entire trigger system — with hundreds of boards — such that complicated synchronization procedures involving software and hardware must be provided between the boards in the trigger chain [1]. Some synchronization procedures even require human intervention to work properly.

As the beam-crossing frequency increases in a future LHC upgrade, and as more data is brought into the Level-1 trigger system in order to provide additional rate reduction, these procedures will become ever more complex and difficult to maintain.

We are developing a prototype system to test the fundamentals of a completely asynchronous electronic trigger system appropriate for future LHC upgrades of the Level-1 trigger systems. An asynchronous solution has been standard for the higher trigger levels in collider experiments, which are based on software filters running on conventional CPUs, but not in the first level of triggering. This does not imply that no synchronization is done, because an accurate time marker synchronized to the beam-crossing is still needed at the detector front-end to label data. But this synchronization need not propagate further downstream in the electronics. The modelling, firmware and hardware development for a prototype asynchronous trigger are conducted concretely within the context of the cathode-strip chamber muon system of the CMS experiment, but the ideas of an asynchronous trigger apply more generally to most, if not all, detector subsystems.

There are many reasons for a fully asynchronous trigger system. First of all, in order to achieve higher luminosity in an LHC collider upgrade (the so-called SLHC, with a luminosity of $L=10^{35}$ cm^{-2}s^{-1}), it has been proposed [2] to shorten the bunch spacing between collisions (to 12.5 ns, for example) or even to go to nearly continuous collisions with extremely long bunches (>100 m). The experimental challenges of handling such scenarios are daunting, since improvements in the time resolution of detector technologies may not keep pace with the increased collision rate. Thus, the Level-1 trigger system will be less able to clearly identify from which bunch-crossing a particle originated, and time-windows of several bunch-crossings will need to be employed by the trigger logic to catch efficiently the detector data. That being the case, a trigger architecture designed around a beam-crossing clock is less compelling than one designed around the arrival of the data itself.

In fact, time windows of several bunch crossings already are employed in the current Level-1 trigger design for the CMS experiment. The cathode-strip chamber trigger system expects to use a 50 ns window to efficiently identify muon track segments, and the drift-tube muon system will likely require even larger time windows [3]. Shorter bunch spacings will enlarge the number of bunch-crossings the trigger system must consider for the same detector technology.

Secondly, an asynchronous system could respond more robustly to bursts of data, if the amount of time used for transmission is allowed to vary. In a synchronous system, the bandwidth on the data links could be flooded for unusually busy events (which could be indicators of new physics!), whereas an asynchronous system could respond by taking more time to transmit all of the data. If the data buffers in the system are large enough, the effect of such bursts can be kept minimal.

It should be noted that the Level-1 trigger decision still needs to be reached within a certain amount of latency (although not necessarily fixed) since the size of the buffers holding detector data will always be finite. Every effort should be made to reduce the trigger processing time, of course.

A third consideration in the design of a future trigger system is the operating frequency of the data links used to transmit data. As the luminosity increases with an LHC upgrade, so does the amount of data. Most likely the experiments will make use of optical or copper links operating at 10 Gbit/s or even higher (whereas currently links operate at no more than a few Gbit/s). The serializers and de-serializers (serdes) associated with such links require very stable clocks for transmission and reception.

On each board in a traditional synchronous trigger system, all data-transmitting and processing components must also work synchronously with the system clock. While it is generally not a problem for components like FPGAs or memories, it becomes a major issue for other components, such as multi-gigabit optical links.

Multi-gigabit optical links require reference clock with very low jitter, typically around 40 ps [4]. Input clock frequency for such links depends on target bandwidth, but it always must be a multiple of system clock frequency.

Because of such strict jitter requirements, any conventional clock frequency multipliers, such as Digital Clock Managers (DCM) in Xilinx FPGAs, cannot be used, since they introduce much larger jitter into the output clock [5]. The only way to provide the reference clock with sufficient quality is to use Phase-Lock-Loop device (PLL) based on voltage-controlled crystal oscillator with ultra-low jitter. Such oscillators must be custom-made for each particular system, since the clock frequency required for a particular trigger system practically never matches the row of standard frequencies used in industry.

The maximum bandwidth of the optical link typically cannot be reached, since the frequency of reference clock of the optical link must be a multiple of the system clock frequency. Even if the link is able to run faster, one has to slow it down to match system clock.

In an asynchronous system, the optical links can be run from inexpensive and easily available crystal oscillators generating one of the industry-standard frequencies that allows for maximum bandwidth at which the link was designed to perform.

Moreover, in a typical trigger system, only a small fraction of the optical link's bandwidth is used to transmit the actual trigger data; most of the time the link transmits zeroes (or another "no data" placeholder). On the other hand, there are typically separate DAQ, fast monitoring and slow control data cables, which are also used only for fraction of the time.

To allow for better bandwidth use and to reduce the number of cables in the system, it seems to be logical to send all data communication streams (trigger, DAQ, etc.) via the same data link. Since some data are more time-critical than the other, a priority system must be established. One of the possible priority level examples is shown below:

From front-end to trigger board, in order of priority:

1. Track stub data for trigger decisions, with time markers.
2. Critical status information (buffer overflow, etc)
3. DAQ information (raw hit data)
4. Slow control data

From trigger board to front-end, in order of priority:

1. Level-1 accept, with time marker
2. Slow control commands and data

To send the higher-priority data, the transmission of lower-priority data must be interrupted immediately (if it was going on at that time). For example, according to the priority system shown above, track stub data for trigger decisions will not wait for anything and will be transmitted to trigger boards immediately. Similarly, Level-1 decision, when made, will be transmitted immediately to front-end boards.

A fourth consideration is that de-coupling the beam-crossing frequency from the trigger logic frequency allows the electronics designers to choose a clock frequency that is appropriate for a given technology and for a given trigger algorithm. In a synchronous system, there is always some overhead at the end of a clock cycle, so the frequency of the logic matches the system clock. This leads to a decreased performance of logic devices, such as FPGAs. In an asynchronous system, the logic core of each trigger board can work from its own clock generator, at the frequency that provides optimal performance for this particular board. This will also make any future upgrades much easier.

Finally, an asynchronous design opens up the possibilities of using more traditional CPUs in the Level-1 trigger system, such as DSPs, or hybrid designs such as the Xilinx Virtex-2 Pro [5], which combines FPGA programmable logic with (multiple) PowerPC CPUs. Such software-based designs have been standard in the Level-2 and Level-3 trigger systems of modern collider experiments, but have typically been deemed too slow in the past for the tight latency and synchronization constraints of the first level of triggering. But with CPUs operating in the GHz regime, and with memories becoming deeper and more inexpensive—thus allowing a longer Level-1 latency—we feel that such CPU-based solutions deserve re-evaluation for Level-1 triggering. In particular, more sophisticated jet or track reconstruction algorithms may be possible with CPU designs.

An asynchronous trigger system, however, does not alleviate the need for an accurate time marker synchronized to the beam structure; but this timing signal need only be distributed to the very front-end electronics that latch the detector data, and it is not used to drive data synchronously through the trigger system. Once the data are latched, a time marker is also stored (the bunch-crossing number, for

example) to denote when the data occurred. All further processing need only refer to this time marker. For example, in the software filters that implement high-level triggers, all data are referred to by their "Level-1 Accept" number. The same can also be done at Level-1 using the bunch-crossing marker. All other electronic boards in the trigger chain downstream of the front-end electronics need not keep their own bunch-crossing counters to monitor their synchronization, since the time marker is already sent with the data. Synchronization procedures in this case are necessary only to synchronize the front-end boards with each other, so the time markers assigned to data are accurate.

The "price" of an asynchronous trigger system is that the processing logic associated with the exchange of data between boards is slightly more complex. A board receiving data from multiple sources must decode the time markers, and using a predefined time window, determine which data should be considered as having come from the same collision. But this logic is small, and the size of available programmable logic is continually increasing.

The front-end boards in such system will contain some amount of memory to store the DAQ data, which may be required later, if Level-1 decision is made. The size of this memory cannot be infinite, so Level-1 decision must be made within certain time from the event. However, this time does not have to fixed as in synchronous systems. The only requirement is to have a Level-1 decision made no later than N bunch-crossings after the event. Level-1 decision signal transmitted to the front-end boards will have time marker attached, so the front-end boards can identify the portion of DAQ information to transmit.

If a Level-1 decision is still required with a fixed latency, (for example, if some of the front-end boards were designed for a synchronous system, but are used in an asynchronous system), it is always possible to re-align the "Level-1 Accept" signal at the last possible stage before delivering the result back to such older front-end boards. This wraps the asynchronous design into a synchronous "black box". But we expect that such a constraint is not needed unless to maintain compatibility with existing LHC electronics.

II. ASYNCHRONOUS TRIGGER SYSTEM PROTOTYPE PROJECT

A. Structure

The Asynchronous Trigger System Prototype will be constructed by upgrading the front-end trigger electronics of the cathode-strip chambers (CSCs) of the CMS Endcap Muon system to perform pattern recognition and bunch-crossing identification from the anode data at a twice higher frequency of 80 MHz, and to transmit these results asynchronously using high-speed (10 Gb/s) optical links to a newly-designed track-finding processor that receives data from up to 3 chambers. These prototypes will directly demonstrate the capability of

the Level-1 trigger of the CMS Endcap Muon system to operate at the higher bunch-crossing frequency anticipated for the SLHC; but more importantly, they will demonstrate the possibility that a system of Level-1 trigger electronics (muon or otherwise, and for any collider experiment) can operate essentially asynchronously in the pattern recognition and data transmission (although clearly the bunch-crossing assignment must be synchronous to the SLHC machine frequency at the very front-end). We also plan to test tracking triggers based on data input from multiple CSCs. Each CSC has 6 layers of (digital) wire-group data (about 100 wire-groups per layer), so two or more CSCs allows the possibility of studying track-finding algorithms in the Level-1 trigger based on data collection from 10 or more layers, which could be appropriate for implementing a Level-1 trigger for the silicon tracking systems of the LHC experiments.

This particular implementation of an asynchronous trigger design has been chosen to make full use of the existing resources at the University of Florida and our previous experience with the CMS Level-1 Muon trigger system.

The electronic project is broken up into two board types: a new mezzanine board that upgrades the ALCT trigger logic on the CSC and has a bi-directional optical link for trigger and DAQ data, and a track-processing board that receives data from up to 3 mezzanine boards, performs track-finding algorithms, and requests and receives CSC data from the mezzanine boards when a trigger is generated.

We will be able to test our completed system with real muon chambers taking cosmic-ray data. The UF high-energy physics group maintains a cosmic-ray test stand, shown in Fig. 1, which can accommodate up to three CSC chambers with associated gas, high-voltage, and readout electronics for tests of the trigger and data acquisition systems and future R&D. This will allow us to demonstrate that the asynchronous system can trigger the readout of several muon detectors. Moreover, we can make use of existing software to download firmware and control registers in VME boards.

Figure 1: UF Cosmic ray test facility for CSC and electronics

B. System-Level model

Before the hardware construction, the system-level model will be written and studied. This model will be written using VPP - a Verilog HDL simulation and generation library for C++ [6]. This library, which was developed in our group, allows for writing C++ programs that simulate the exact behaviour of programmable logic. Such model can be compiled by any C++ compiler, and later incorporated into larger model if necessary.

Once we are satisfied with the behaviour of the logic, the VPP-based model can be recompiled to generate a valid Verilog HDL code for all programmable devices in the system.

C. Anode Mezzanine Board

Since we are making use of the current CSC hardware, we only need to upgrade the mezzanine board on the ALCT front-end board, which contains the pattern-recognition FPGA, in order to prototype an asynchronous 80 MHz system. The existing anode electronics (preamplifier and discriminator) should work fine.

We plan to build 4 new mezzanine boards containing a Xilinx Virtex II Pro X and a 10 Gbit/s optical link. The ALCT equipped with this board will be able to:

- Input raw hit data from the cathode chamber

- Process these data in order to find best track segments

- Assign an exact time marker for each track segment found

- Report these track segments asynchronously via the serial link to the track processing board

- Store raw hit information in the circular memory buffer

- Retrieve the raw hit information upon a Level-1 decision and send it to the track processing board via the same serial link. Transmission of raw hit data will not disrupt trigger functionality because trigger data will be given higher priority for transmission.

In order to be able to assign the time marker to the track segments, this mezzanine board will accept a precise timing signal as input. This precise timing signal corresponds to the bunch-crossing time in a real trigger system.

D. Track Processor Board

The track-processing board will receive data asynchronously from the ALCT mezzanine boards and perform the track-finding algorithm that identifies muons traversing the CSC chambers. Our current intention is to design and construct a 9U VME board that has 3 optical link connections for the trigger/DAQ data from up to 3 chambers (a final design could have many more connections). A fourth optical link connection will be added to send the collected DAQ data directly to a computer (via gigabit Ethernet, for example). This will let us avoid reading out the data through the VMEbus, which may be slow (although such an option will be possible, especially for early debugging).

Two processors will be built. This will allow us to test inter-processor communication, such as might be required for tracks crossing processor boundaries or for calorimeter clustering algorithms. The processor prototypes can be arranged such that each receives data from two mezzanine boards, and the third trigger/DAQ link on each processor can be used for inter-communication.

The firmware for the processor consists of a de-serialization stage, a data alignment stage, a track-finding stage, a trigger decision stage, a data retrieval stage, and a DAQ output stage. Additionally, there will be slow and fast monitoring controls, such as link error detection and buffer overflow.

After de-serialization, the data alignment logic must correlate the data received from multiple mezzanine boards in order to determine which track segments belong to the same track by checking for agreement in the assigned bunch-crossing numbers within a pre-programmed tolerance.

In order to be detected, a track must contain a certain number of track segments. If all those segments are received from ALCT mezzanine boards, the track-processing board will:

- Check that these segments have time markers matching within certain limits

- Log the complete track (instead of reporting to a higher level of triggering, which is not implemented in this prototype). A computer can later read out the information about this track.

- Generate a Level-1 acceptance decision and send it to the ALCT mezzanine boards, along with a time marker of the track found

- Receive the raw hit data for this particular track from the ALCT mezzanine boards, and send it to the DAQ computer along with the data for the track which was found

The track segments that did not result into the complete track will be discarded.

The track-finding logic, initially, will be designed to find straight tracks in the anode view of the CSCs. This logic will analyse the aligned data and search for at least two track segments that lie within a road along the wire-group view with a programmable tolerance.

Each track found by a track processing board will be assigned the bunch-crossing number, corresponding to the bunch-crossing number of the track segments composing this track.

If a track is identified, the trigger decision logic generates a "Level-1 Accept." This means that the bunch-crossing number corresponding to the found track is reported back to the front-end boards (the mezzanine boards) through the same bi-directional link. The front-end boards will retrieve data

from buffers that lie within a pre-set time window around this bunch-crossing number, and send the data back through the trigger/DAQ link to the processing board.

The prototype system described above will let us test one more conceptual approach to the construction of a trigger system. As the programmable logic devices (FPGAs) become increasingly powerful, it becomes possible to cluster more logic into one device, instead of separating it into several devices, thus reducing the number of logical stages and latency of a trigger system. For this particular prototype, we plan to prepare a special version of the firmware for the track processing board that will receive all raw hit data directly from the ALCT mezzanine boards. The track processor can then directly reconstruct tracks from all raw hits instead of first creating track segments in the ALCT and then linking them into tracks in the track processor. Preliminary calculations show that this will let us significantly increase the quality and speed of track reconstruction, because we can analyse all chamber layers simultaneously.

III.   REFERENCES

1.   L.Uvarov. "MPC to SP Synchronization Procedure" http://www.phys.ufl.edu/~acosta/cms/LU-MPC_SP_Synch_Procedure_2d0.pdf

2.   O.Bruning et al., "LHC Luminosity and Energy Upgrade: A Feasibility Study", LHC Project Report 626.

3.   CMS Collaboration, "The Trigger and Data Acquisition Project, Volume 1", CERN/LHCC 2000-038 (2000).

4.   Texas Instruments. "TLK2501 data sheet." http://www-s.ti.com/sc/ds/tlk2501.pdf

5.   Xilinx, "Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet", http://direct.xilinx.com/bvdocs/publications/ds083.pdf

6.   "VPP – a Verilog HDL simulation and generation library", http://www.phys.ufl.edu/~madorsky/vpp