Available on CMS information server CMS IN -2011/008

CMS Internal Note

The content of this note is intended for CMS internal use and distribution only

22 March 2011 (v2, 29 March 2011)

A Time-Multiplexed Calorimeter Trigger for CMS with Addendum

G. Hall, G. Iles[i], J. Marrouche, A. Rose[ii] and A. Tapper
Blackett Laboratory, Imperial College, London SW7 2BW, UK

R. Frazier and D. Newbold
H.H. Wills Physics Laboratory, Tyndall Avenue, Bristol BS8 1TL, UK

J. Jones

Abstract

A time-multiplexed approach to the CMS experiment's level-1 calorimeter trigger is explored. The concept may also be useful for other parts of CMS under development, such as a tracking trigger. It is particularly useful in applications that benefit from concentrating several terabits in a single FPGA or that require more processing capability than is available in a single chip at any one time. The underlying technology, which is common to a conventional trigger, is also evaluated, including hardware prototypes and software infrastructure.

Addendum

The consequences for the design of a time-multiplexed calorimeter trigger are examined in the scenario that the main processing nodes receive 16 bits of calorimeter information per tower, rather than the raw data of between 18 and 24 bits per tower. An ECAL-only energy cloud sum is also considered.

[i] g.iles@imperial.ac.uk

[ii] awr01@imperial.ac.uk

Contents

1. Introduction

The level-1 trigger system of the CMS experiment selects interesting physics events at a rate of 100kHz from an input rate of 40MHz. It is designed to operate up to a luminosity of 10^{34} cm^{-2} s^{-1}. With the planned LHC phase-I upgrade in 2015, the luminosity will increase to 2×10^{34} cm^{-2} s^{-1} and will reach 5×10^{34} cm^{-2} s^{-1} with the LHC phase-II upgrade in 2020. The level-1 trigger system will operate well up to the nominal luminosity but, beyond that, the performance will be degraded due to the increased number of pile-up events which makes distinguishing physics objects from background more challenging.

To counter any loss of performance, it is planned to upgrade the calorimeter clustering algorithms and improve the resolution at which these operate so that they take full advantage of the $0.087\eta \times 0.087\varphi$ granularity of the trigger primitives generated by ECAL (Electromagnetic Calorimeter) and HCAL (Hadronic Calorimeter). The upgrade should also leave open the potential to include trigger information from the Tracker at phase-II.

The calorimeter trigger is one of the most challenging aspects of CMS, not only because the volume of data exceeds several Tb/s, but also because (a) the data must be shared or duplicated between processing nodes to satisfy boundary constraints, (b) the resulting physics objects need to be sorted in order of significance and (c) all processing must be achieved within a latency budget of ~1μs. The data sharing is a particularly significant constraint which has, in the past, required complex system architectures and backplanes to either share or duplicate data between processing nodes.

The extra algorithmic complexity required for an upgrade is only now becoming feasible because of the continuing advances in the performance of digital signal processing in reconfigurable programmable logic (FPGAs). This technology should allow CMS to build a much more powerful, yet simpler and easier to maintain trigger than exists currently, but which has characteristics that are significantly different from the technologies used in the past.

Firstly, the inclusion of embedded serializers has meant that high-speed serial links have emerged as an ideal means of bringing large volumes of data into FPGAs. They do, however, have a high latency (typically 50-200ns) and it is, therefore, essential that the number of serialisation stages be kept to a minimum. Consequently, all new designs are based on just 3 or 4 serialisation stages, including both the serialisation stages from the ECAL and HCAL to the calorimeter trigger and those from the calorimeter trigger to the GT (Global Trigger).

Secondly, due to the reduction in semiconductor feature size (e.g. 28nm for Xilinx-7 series), FPGAs now operate with a clock of several hundred MHz. Where the native clock is much faster than the raw data rate, this lends itself to a pipelined architecture rather than one that is simply clocked. For example, imagine data words being clocked from a deserializer at 240MHz (i.e. six times the LHC bunch crossing clock). A conventional architecture will typically wait until all the data are present and then process it in parallel, first doing task A, then B, then C, etc. until the next bunch crossing (i.e. it consists of six stages that are only active for one sixth the time). In a true pipelined design all tasks are running concurrently with new data fed into task A on each 240MHz clock cycle. This approach is only really applicable when you start to have many clock cycles of data to process. It may be possible to optimise a conventional design so that it does not have to wait until all the data is present, but at the expense of algorithm complexity.

These changes in the available technology require us to re-evaluate the calorimeter trigger architecture; from the current architecture which follows a conventional design of parallel nodes that process the data from small parts of the detector for every bunch crossing to one based on a time-multiplexed architecture that processes large areas of the detector over many bunch crossings. This concept was first proposed by Jones et al.[1][2].

2. Basic Concept

These two designs are fundamentally different: to understand these differences, it is useful to look at some very simple examples (Figure 1).

In a conventional trigger (such as the current CMS calorimeter trigger) the experimental data are processed at increasingly coarser resolutions: at each processing stage the trigger performs clustering operations to build physics objects, which are then sorted in terms of importance. Depending on the number of parallel processing nodes, the result may need to be passed to a subsequent sorting stage to reduce the data rate into the next processing stage. If the CMS calorimeter trigger is upgraded using a conventional architecture, there would be two processing stages; a Regional Calorimeter Trigger which clusters and sorts electron/tau candidates at tower resolution and a Global Calorimeter Trigger which clusters jets at region resolution (4×4 tower) or possibly ½-region resolution (2×2 tower).

In a time-multiplexed trigger (Figure 1), multiple data from a single bunch crossing (bx) are concatenated and delivered to a single processing system over multiple bx. This approach requires several processing systems operating in a "round-robin" fashion with processing system 1 takes bx = n, processing system 2 takes bx = n+1 and so on. This mode of operation is identical to the existing Higher Level Trigger (HLT), which uses ~1000 PC cores operating in a time-multiplexed manner.

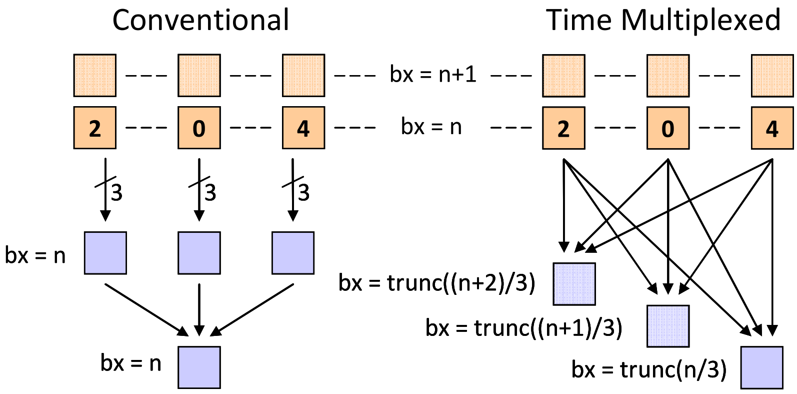

Figure 1: A simple comparison of a conventional versus time-multiplexed trigger.
The orange boxes at the top of the diagram represent the experimental data arriving from three locations on two successive bunch-crossings (bx). On each bunch-crossing, the data from each location is a number between 0 and 7 and, therefore, to fully describe the data, 3 bits per bx per source are required. The blue boxes represent hypothetical processing nodes, each with 3 inputs and 3 outputs, where each input is capable of receiving 1 bit per bx and each output capable of transmitting 1 bit per bx. In a conventional trigger architecture, each processing node uses all 3 inputs to receive all the information from a particular location within a single bunch crossing. The next processing stage, however, must encompass a broader area, and so, in order to receive data from all three previous nodes, the data from each must be reduced to a single bit per bunch-crossing, giving a far coarser resolution.
A time-multiplexed architecture has each input of the processing node connected to a different location. At the first bx, the first node can receive a single bit from each location, say, the most-significant-bit. We cannot send the other two bits to the other nodes, this would make no sense, so instead we "store" them in the source location and send nothing to the other nodes. At the second bx, we send the stored next-most-significant-bit to the first processing node, the current most-significant-bit to the second node (again storing the other two bits) and nothing to the third node. On the third bx, the remaining least-significant-bit is sent to the first node, the stored next-most-significant-bit to the second node and the current most-significant-bit to the third node. The first node has now received all the bits from all location for the first bx and can, therefore, perform both local and coarse processing. As the first node needs no more input for the first bx, it is also ready to receive the most-significant-bit from the next (4th) bunch-crossing.

The obvious advantage of a time-multiplexed trigger is that the boundaries between processing nodes that exist in a conventional trigger are removed and all the data is processed in one location, making the system very flexible. In the simple time-multiplexed example (Figure 1) each processing node has all the input data from the bunch crossing it is processing. In particular, for algorithms that require large overlaps between neighbouring processing nodes in a conventional trigger architecture, a time-multiplexed system becomes much more efficient because the ratio of the area processed to the boundary area is substantially increased. This results in fewer cards and fewer interconnections, which also makes the subsequent sorting of trigger objects simpler, faster and, more importantly, more accurate since no partial sorting is required.

The system has other advantages: for example, it is possible to prototype the entire trigger system with just a fraction of the hardware and use it to develop and test new algorithms throughout the lifetime of the system. It also offers redundancy; if one of the processing systems were to fail the data could easily be redirected to a backup processing node. Furthermore, the system does not require complex active or passive backplanes and can be built with a single card design, although a two card design may be a more cost effective solution.

The obvious drawback with this approach is that there is an initial latency cost due to the time delay introduced by the multiplexor; this is offset, however, by the ability to build a much more compact trigger, requiring fewer serialisation stages and no sharing of data across boundaries. A further saving is made in the lower processing latency of a time-multiplexed design introduced by the substantially higher speed at which time-multiplexed processing pipelines are clocked. A further disadvantage of a time-multiplexed design is that it may also require a substantial cabling arrangement between time-multiplexer and processing node; this issue can be solved with a large optical patch panel, although the use of compact, ribbonised optics will allow this component to be far smaller than past equivalents using copper serial links.

In practice, there is often an extra stage in both the conventional and time-multiplexed triggers. A conventional trigger needs to share data between boundary regions and thus the experimental data needs to be repackaged so that the data is duplicated efficiently across processing nodes. A time-multiplexed trigger requires a stage to time-multiplex the incoming data.

3. The Trigger Geometry of CMS

A diagram of the CMS calorimeter trigger geometry is shown in Figure 2. The region of CMS in which both ECAL and HCAL trigger input data are present spans a range of ±3 η and all of φ. It is segmented into 56 towers in η and 72 towers in φ with a granularity of 0.087η × 0.087φ up to ±1.74 η. The HF (Forward Hadronic Calorimeter) extends η coverage up to ±5 η, albeit at a coarser resolution. On each side the HF is segmented into 8 units in η and 36 units in φ, covering an area equivalent to 16 barrel towers in η and 72 in φ.

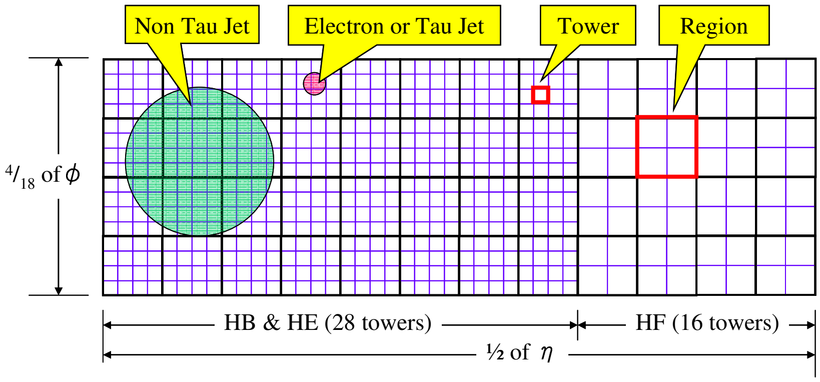

Figure 2: A graphical view of the CMS trigger geometry for the HCAL. The geometry of the barrel and endcap ECAL is identical. There is no forward ECAL.

4. A Time-Multiplexed System for CMS

There are many ways to time-multiplex the incoming calorimeter trigger information; one choice with elegant features, however, is to have rings of constant η on each optical fibre and to time-multiplex the data in the φ-direction, Figure 3. One possible scheme for implementing this design is discussed here and shown in Figure 4.

Data would arrive from ECAL & HCAL on serial links running at 4.8Gb/s and the time-multiplexing would be performed in the Pre-Processor (PP) cards. For the barrel and endcap calorimeters each PP would receive a ring in φ (72 towers) that was 2 towers wide in η. The data would be delivered on 18 fibres that each spanned 4 towers in φ and 2 towers in η (i.e. matching current granularity of ECAL/HCAL). It would be time-multiplexed sequentially around the ring and re-transmitted at double the rate (9.6Gb/s using 8B/10B encoding or 7.92Gb/s using 64B/66B encoding) on 10 fibres over 9 bunch crossings. It is necessary to have 10, rather than 9, fibres to provide space for the necessary packet header, trailer, checksum and byte alignment commas. The lack of ECAL data in the forward region and the lower resolution of the HF enable these rings to be 8 towers wide in η. Therefore, the barrel and endcap require 2×28 PP cards (ECAL + HCAL) whereas the HF requires only 4 PP cards.

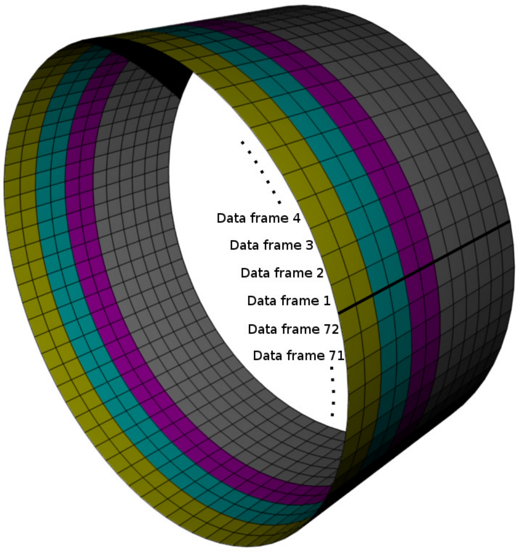

Figure 3: Schematic indicating one possible method of time-multiplexing data for a calorimeter trigger. Each fibre, indicated by colour, carries a ring of constant η and data arrives sequentially in the φ-direction.

Although it is appealing to place all the time-multiplexed data into a single FPGA, this presents practical problems unless the time-multiplex period is lengthened or the data rate is increased. Instead, it is proposed that the Main-Processor node is split over two cards (MP+ and MP-), with each card then handling half the detector in η, with a large, 8 (or 12) tower, overlap available to handle the boundary region. More advanced FPGAs, such as one of the largest Xilinx Virtex-7 parts (e.g. the XC7VX690T or XC7VX865T) with 72 transceivers[3], would make a single-board solution possible and allow for either more complex algorithms or a reduction in latency. Such a design would also require a shift to MicroPODs[4] or similar ultra-small form factor optics to provide sufficient front panel space for at least 64 optical receivers and 24 transmitters in a double-width µTCA card.

In such a scheme, ten main processing nodes would operate in a round robin scheduling manner, each only receiving data for every tenth bunch crossing. Each MP card would receive a single link from each Pre-Processor (PP) card in their respective η half. They would also receive 8 links (4 ECAL + 4 HCAL) containing data from the 8 adjacent towers in the opposite η half so that they have sufficient boundary information to build physics objects at the boundary between the two processing nodes.

The main processing cards would be based around a Xilinx Virtex-7 XC7VX485T in either a FFG1929 or FFG1158 package[3], these having 56 GTX links operating up to 10.3Gb/s. 48 Rx and 24 Tx links would be routed to 6 PPODs[5] (4 receivers and 2 transmitters) on the front panel and the remaining 12 links would be used for AMC ports: 0 for Ethernet, 1 for DAQ, 2 for SATA, 4-7 for Fat-Pipe, 8-11 for Extended Fat-Pipe and 1 spare.

The benefit of using the GTX transceivers is that they are very similar to those used on the Virtex-5/6, with which we already have significant experience. Furthermore, in 8B/10B mode, the internal bus width remains at 20 bits, implying that the internal SerDes clock is running at 500MHz for 10Gb/s operation and, thus, the latency ought to be low (see section: Latency).

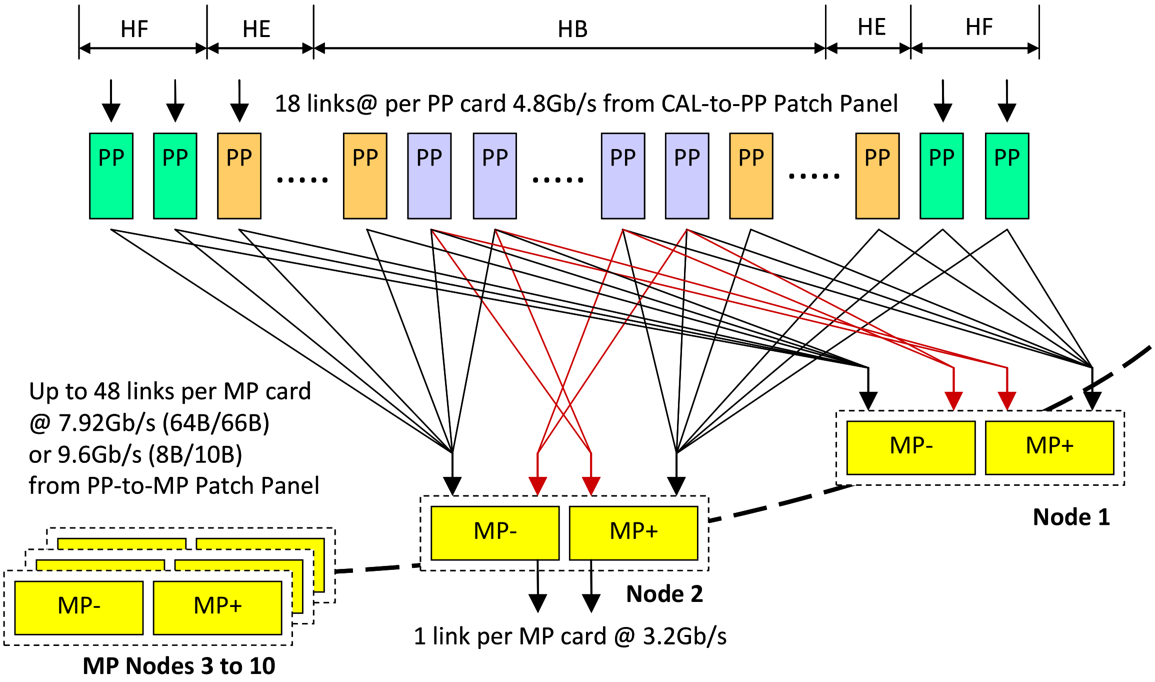
HF
HE
HB
HE
HF
18 links@ per PP card 4.8Gb/s from CAL-to-PP Patch Panel
PP
.....
Up to 48 links per MP card
@ 7.92Gb/s (64B/66B)
or 9.6Gb/s (8B/10B)
from PP-to-MP Patch Panel
MP-
MP+
Node 1
Node 2
1 link per MP card @ 3.2Gb/s
MP Nodes 3 to 10

Figure 4: A time-multiplexed system for the CMS. Only HCAL is shown, but ECAL is similar, albeit without the 4×HF PP cards (green). The PP (Pre-Processor) cards each receive data from rings in φ. There are 28 PP cards (orange & blue) that span HE and HB with 8 of these cards situated around η = 0 (blue only) that duplicate data to provide an 8 tower overlap in the MP cards at η = 0.

It is expected that the BRAM (Block RAM) of 37Mb in the proposed part would be sufficient for buffering data until receipt of a level-1 accept, albeit with little margin. External DDR2/3 or QDRII memory could provide considerably more memory, albeit with significantly lower bandwidth, than the on-chip BRAM, whilst another option would be to select a larger FPGA, for example, the XC7VX690T with 52Mb or XC7VX865T with 63Mb. Another possibility is to include a card dedicated to data acquisition within each processing node. A further consideration is that Xilinx are investigating stacked silicon interconnect technology for the Virtex-7 series, of which one application may be additional memory; whether this technology would be available in the medium term or would provide additional memory is, however, not clear. The requirements for memory is, technically-speaking, probably the most important issue to understand before building a final system.

A time-multiplexed design based on the Virtex-6 and using 2.4Gb/s data from ECAL & HCAL was also considered for two reasons; the parts are available now and using 2.4Gb/s links would allow the new OSLB (Optical Serial Link Board) mezzanine card, required for sending ECAL data to the trigger, to use low cost Spartan 6 FPGAs. It was found that such a scheme does not necessarily save either money or latency and potentially offers significantly less flexibility. A data rate of 2.4Gb/s, rather than 4.8Gb/s, doubles the number of links required and, since each link is run straight into a high performance FPGA, the number of these expensive devices is similarly increased and the cost is simply moved from one system to another. Furthermore, running the GTX receiver at 2.4Gb/s rather than 4.8Gb/s adds 2.5bx to the latency. To run the old and new systems in parallel would require fibre splitting rather than data duplication within the FPGA and thus offers far less flexibility.

The Virtex-6 design would have required 7 MP cards per MP node, a custom backplane because of the large overlap to processing area and a third processing stage because of the limited overlap between processing MP cards.

5. Patch Panels and Rack Layout

Optical patch panels will accommodate the physical re-arrangement of fibres from ECAL/HCAL to the PP stage and from the PP to MP stage (Figure 5). At 4.8Gb/s there would be 504 fibres from ECAL and 576 from HCAL+HF to the PP stage.

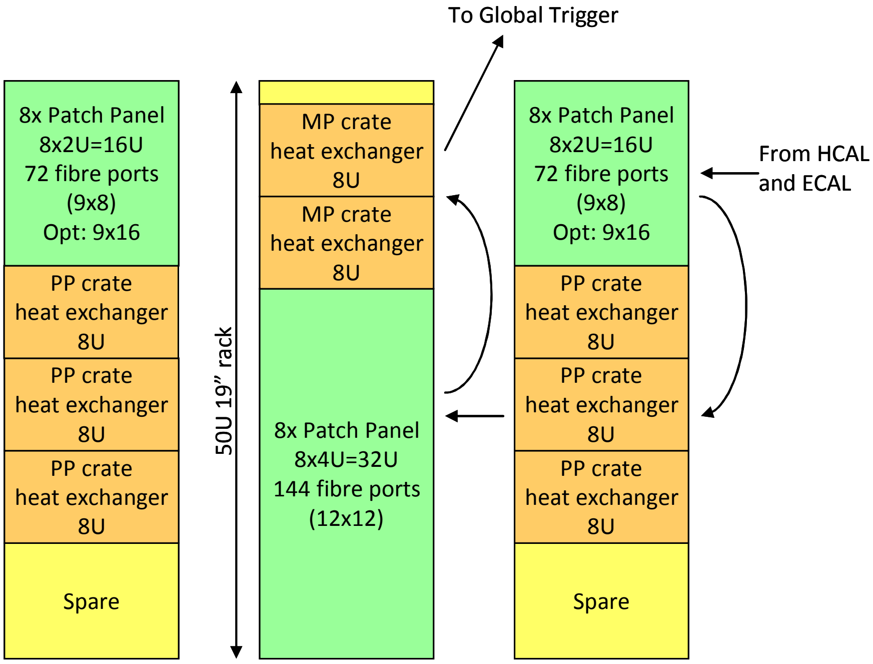

Figure 5: An illustration showing the rack space required by both patch panels and crates and the data flow between them. It would probably be feasible to reduce the total number of racks to 2 by placing the patch panels in the rear of the racks.

The current RCT is segmented into 18 VME crates, which each spanning ½ of η and 8 towers in φ and with the Receiver Cards aligned along 4-tower boundaries in η and φ. Only a minimal upgrade to ECAL planned and thus the physical layout of the crates are not expected to change. The cabling must accommodate this and, consequently, it is foreseen that ½ of η and 4 towers in φ (14 fibres for ECAL, 14 for HCAL and 2 for HF) would be mapped onto 2×12-way ribbon cables, with at least 4 spare fibres per ribbon. These would connect to 4×2U patch panels (for example, FW-RMS02-1410G, 8×9 LC fibre ports)[6] that would span ½ of η and all of φ. A higher density version of the patch panel (FW-RMS02-41410G) would allow the spare fibres to be connected so that a damaged fibre would not require a change to the back side of the patch panel (i.e. make maintenance simpler). The 8 patch panels needed to span ½ of η for both ECAL and HCAL would require 16U of rack space.

The 30 PP cards (14 ECAL, 14 HCAL, 2 HF) to span ½ of η would be accommodated in 3 double-width µTCA crates which, including heat exchangers, would require 24U (3×8U) of rack space. Two 50U CMS racks, one for each ½ of η, would, therefore, be required for both the pre-processor patch panels and pre-processor crates, leaving 10U spare in each.

The PP to MP patch panel would be designed for maximum flexibility (e.g. addition of a tracking trigger input, or extra overlap between MP cards) rather than minimum space by providing a port on the patch panel for all 48 MP card optical inputs, rather than just the 38 required for the baseline system. It would use 8×4U (FW-RMS04-1410G, 12×12 LC fibre ports) patch panels[7]. A double-density version of the patch panel (FW-RMS04-41410G) exists that would reduce rack space requirements, but physically connecting all ports would be very awkward.

Two double-width, 12-slot, µTCA crates would provide space for the 20 MP cards, a redundant MP node and a test node. They would require 16U of rack space, including heat exchangers. The MP patch panel and cards would, therefore, require 48U of rack space, which is just feasible in a 50U CMS rack.

6. Latency

The current latency of the calorimeter trigger path, from the input of the SerDes blocks on the Synchronization & Link Boards that are mounted on the ECAL and HCAL trigger cards through to the output of the SerDes blocks within the GT Pipelined Synchronising Buffer, is approximately 47bx (bunch crossings), Table 1.

	Time (bx)
CAL SLB SerDes Tx	**2**
SLB cable	**2**
RCT	**21**
GCT	**19**
GCT to GT link	**3**
Total	**47**

Table 1: The latency contributions from the current trigger system (e/γ path). The SLB SerDes Tx component is an estimate because the literature does not split the contributions of the ECAL/HCAL trigger processing cards and the SerDes blocks. All latencies are given at the entry or exit to the SerDes block because measuring the latency of just the Tx or Rx component is technically difficult and sometimes not feasible (e.g. asynchronous links).

The latency of a time-multiplexed trigger is estimated to be 34.5bx (Table 2), and thus well within the current latency budget. This estimate assumes that the final jet clustering and sort (not yet fully implemented) will take an additional 4bx beyond the 4bx used for the electron clustering. The processing contribution to the latency currently assumes a 120MHz clock, which has been used to prototype the electron clustering inside on the MINI-T5 prototype card. The SerDes delay of 2.5bx at 7.92Gb/s (2bx at 9.6Gb/s) for a Virtex-7 GTX transceiver is extrapolated from measurements and theoretical understanding of how the Virtex-5/6 GTX transceivers operate, in particular the impact of the internal parallel bus width and the number of internal clock cycles. The table does not include any contribution for realigning the different data serial links or for operation with an asynchronous link clock as used in the current GCT. This could, potentially, add up to 2bx per SerDes stage.

The full SerDes chain of a Xilinx Virtex-5 TXT GTX transceiver takes 24 clock cycles if it is used in a typical configuration (i.e. transmit FIFO, byte alignment and 8B/10B not bypassed, elastic buffer set to minimum and 32-bit wide fabric interface). The maximum speed of the internal SerDes clock is 406.25MHz (speed grade 2), which would potentially offer a latency of 2.4bx but, in practice, this is not achieved because of other constraints. The maximum speed of a TXT part is 5Gb/s and the internal bus, with 8B/10B enabled, is 20 bits wide, limiting the internal bus to 250MHz. At CMS, however, the 40MHz LHC clock makes 4.8Gb/s with a 240MHz internal SerDes clock more appropriate. This translates into a 4.0bx delay (24 clock cycles at 240MHz).

	Time (bx)
CAL/PP SerDes (Tx+Rx) at 4.8Gb/s	**4**
CAL/PP cable (10m)	**2**
PP processing + time-multiplex	**10**
PP/MP SerDes (Tx+Rx) at 7.92Gb/s	**2.5**
PP/MP cable (20m)	**4**
MP processing	**8**
MP/GT SerDes (Tx+Rx) at 7.92Gb/s	**2.5**
MP/GT cable	**0.5**
GT de-multiplex	**1.0**
Total	**34.5**

Table 2: The latency contributions from a time-multiplexed trigger system.

It is assumed that the GTX transceiver in a Virtex-7 part will remain relatively unchanged from the Virtex-5/6 part but that the internal bus will operate at up to 515MHz so as to reach the rated 10.3125Gb/s. Note that faster Virtex-7 transceivers (13.1Gb/s GTH and 28.05Gb/s GTZ may not necessarily have a shorter latency because the internal bus width may be larger to accommodate the higher maximum line rate and, in this case, the internal bus would have to be clocked at a lower rate to reach 7.92Gb/s or 9.6Gb/s.

As stated previously, one option that has been suggested is to drive the PP cards with 2.4Gb/s links rather than 4.8Gb/s links so that a low cost FPGA could be used on the ECAL OSLB cards, replacing the current 4×1.2Gb/s copper links from the SLB with 2×2.4Gb/s optical links. Decreasing the speed of the GTX transceiver to 2.4Gb/s would increase the latency to 8.0bx. In practice the latency would be closer to 6.5bx because the low cost Spartan 6 FPGA has a GTP, rather than GTX, transceiver with a 10-bit wide internal bus. This example illustrates the point that a more expensive, higher line rate transceiver does not necessarily have a lower latency. As a general rule the transceiver should be operated at close to its maximum line rate to have the lowest latency.

Note that in a conventional trigger system the algorithm normally waits until the entire bunch crossing data are present and, thus, it is necessary to add another bunch crossing of latency. In a time-multiplexed trigger, data passes directly into the algorithm and so the only constraint is that the data from each serial link must be aligned to the same clock cycle. A custom synchronization block has been built and it has been shown that the slowest path through it (i.e. the SerDes link with the largest latency contribution) only passes through a single LUT, making the additional latency required to align the data very small.

7. Technology Demonstrator: The MINI-T5

To evaluate the feasibility of different trigger architectures, gain experience in the latest technologies (e.g. μTCA) and develop the core firmware and software blocks that are common to both a conventional and time-multiplexed design, we have developed a double-width, full-height AMC card, called the MINI-T5, to prototype new trigger designs (Figure 6).

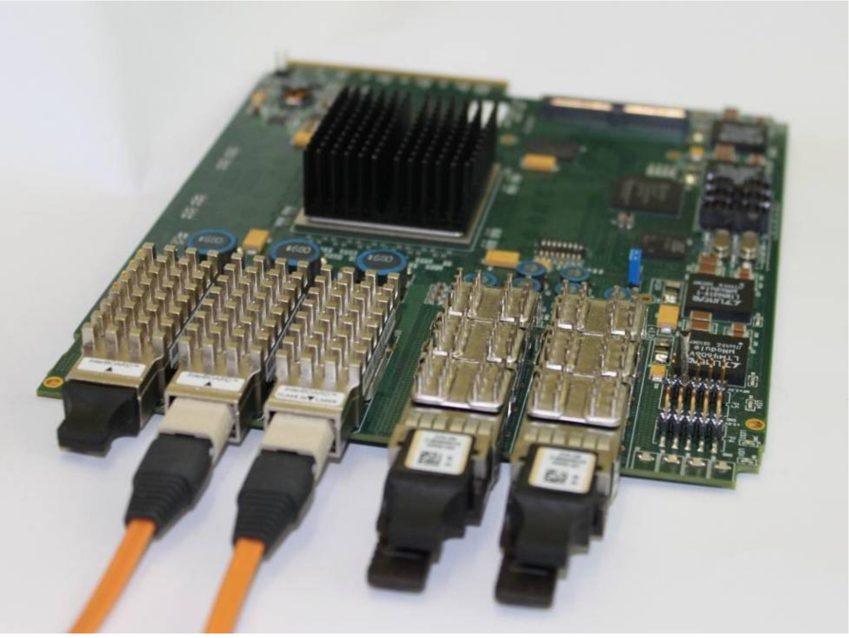

Figure 6: The MINI-T5 prototype generic trigger card.

The card is compatible with either a Xilinx XC5VTX150T or XC5VTX240T FPGA and offers 32 input and 20 output optical links running at 5Gb/s, providing 160Gb/s (input) and 100Gb/s (output) of optical IO capability. A mixture of unidirectional and bidirectional optics is used: two QSFPs, each providing 4 bidirectional links, and three (2 input, 1 output) SNAP12s (board revision 0) or PPODs (board revision 1), providing 24 input links and 12 output links. The change from SNAP12s to PPODs between board revisions 0 and 1 was driven simply by a lack of availability of SNAP12 devices; these are, however, available once more.

Because of the asymmetric input/output, the 32 optical links (SNAP12/QSFP) were tested in two stages; in the first stage, 20 channels were connected through external fibre loopback and the rest through internal transceiver loopback and in the second stage, the channels were swapped so that all had been tested using external fibre loopback. The links were operated in this way for 12 hours, corresponding to ~7×10^{15} bits, without error. The PPOD optics on board revision 1 will be similarly tested shortly.

The majority of the remaining high-speed serial transceivers are connected so that the card is compatible with the services available from a standard μTCA telecom crate with the MCH in the primary slot, namely ports 0 for Ethernet, 2 for SATA/SAS and 4-7 for Fat Pipe (e.g. SRIO, 10GbE, PCIe). The absence of a dedicated PCIe clock in a telecom crate does require that any PCIe devices support the PCIe independent clock option. The last two high-speed serial transceivers are connected to AMC ports 1 and 8 so that they can utilise the DAQ functionality provided by the CMS service card[8]. The remaining AMC ports are connected to LVDS.

In addition to high-speed serial connectivity, the MINI-T5 also has dual 40×800Mb/s LVDS IO via a 40 way differential Samtec connector on either side of the card. These can be joined together via an off-the-shelf Kapton cable from Samtec. It provides a 2×32Gb/s low-latency connection. An Atmel AVR32 microprocessor provides IPMI control and a USB2 interface.

8. Firmware Algorithms & Laboratory System

The laboratory demonstrator system is simplified to focus on the resource-hungry components of the trigger algorithms. It assigns 8 bits per tower for both calorimeters, which is used solely for the energy deposition. In the existing system there are 9 bits per tower, with the extra bit used for calorimeter-specific information. The demonstrator system also ignores the HF because the lower resolution (double tower) and missing ECAL in this region reduces the data rate by a factor of 8.

The demonstrator system is, therefore, a good approximation to the challenges posed by a real system. In the full lab system there would be 28 links (although, at present, we have limited the system to 12 input links because, with a single card, we cannot drive all 28 input links) running at 4.8Gb/s, each loading 2 ECAL and 2 HCAL energy depositions per 120MHz clock. A single clock cycle therefore loads an entire row of constant φ (i.e. 56 towers in η) and, hence, it takes 72 clocks (24bx) to loop over the full φ span of 72 towers. Whilst this is most likely too long for the final system, it is perfectly acceptable for testing algorithms in the laboratory. Within the smaller FPGA of the Mini-T5 board (XC5VTX150T), the test system required 22% of registers, 29% of LUTs but almost all the BRAMs. The cause of this is now known to have been the inefficient packing of 18kb BRAMs into the larger 36kb BRAMs, which should be relatively simple to fix. Although external RAM will be available in a final design, it is unsuitable for use in the L1 pipelines, which require a very large bandwidth.

The test system currently implements only the 2×2 electron-finding algorithm[9], albeit implemented for a time-multiplexed trigger, which has been verified using a C++ test bench in conjunction with ModelSim's Foreign Language Interface. Work on the jet finding and subsequent sort has been delayed so that a robust software structure for hardware access, similar to the CMS HAL for VME access, can be put in place.

A direct comparison of conventional and time-multiplexed algorithms has previously been performed on the GCT Matrix card using exactly equivalent implementations of both the current electron-finder algorithm and current jet algorithm; the output of both methods being shown to be identical[2].

9. Firmware Infrastructure

The core firmware architecture is relatively simple (Figure 7). It comprises 5Gb/s GTX transceiver elements configured to have the minimum possible latency without bypassing elements such as the transmit FIFO or receive elastic buffer. These can be bypassed to reduce latency at the expense of added complexity. A low latency design in the GCT project suffered from data corruption depending on the firmware build, a problem eventually traced to subtle clock routing issues[10]. Although these issues were eventually solved, it shows that considerable care must be taken when using the SerDes blocks in a non-standard configuration.

Many validation features are included in the firmware. Both the input to the GTX transmitter path and the input to the algorithms can be driven by a pattern derived from a bunch crossing counter or from a pattern injection RAM. The RAM can also be used to capture incoming data. The firmware controlling the links was validated by driving data out of the GTX transceivers onto optical fibres of differing lengths and receiving it with different GTX transceivers, thereby ensuring that the low-latency synchronisation blocks which align the incoming data are fully tested.

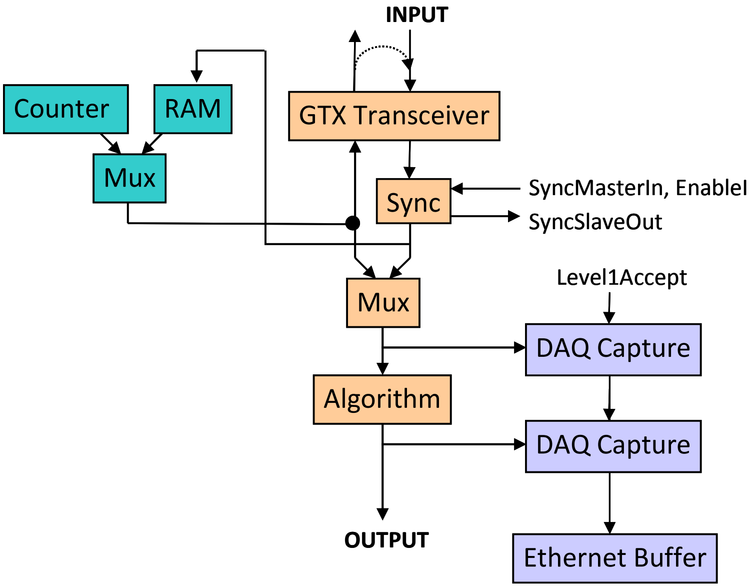

Figure 7: Diagram of core firmware used in the MINI-T5 processing card for the laboratory demonstrator system.

The firmware interface for passing data into the algorithm is simply an array of 32-bit wide words connecting directly to the output of the GTX transceivers; this makes replacing the existing algorithm with new or modified algorithms trivial.

DAQ capture units are firmware modules which can capture an array of 32-bit wide data of arbitrary length in a pipeline and, upon receipt of a level-1 trigger, transfer it to a DAQ buffer. A DAQ capture unit is placed both before and after the algorithm module, enabling direct verification of the algorithm in software.

10.Software and Control Infrastructure

Although the software infrastructure is essentially independent of the choice of a conventional or time-multiplexed trigger, the choice of µTCA technology and the increased homogeneity of the system introduced by the use of generic, rather than the existing highly-specialized, processing cards offers some interesting possibilities for the design of the software, as well as offering some obvious direction.

The AMC.2 standard provides the specification for using Gigabit Ethernet (GigE) as the µTCA control bus and, with the wide availability and low cost of networking hardware, this was an obvious choice for the control fabric. With the choice of GigE as the link layer, a transport layer is also required; this is currently UDP/IP because of the simplicity and low resource usage of implementing a UDP core in firmware. If the use of UDP is shown to be too unreliable or if a "better" transport layer is found, however, this may change in future.

Within the FPGA, a simple bus, "IPbus"[11], developed by HCAL is used. The bus uses a flat architecture with 32-bit address space and 32-bit data bus, although, to allow parallel development of firmware modules, a scheme for the logical division of address space is used so as to emulate a hierarchical bus[12]. An Ethernet Hardware Access Layer (HAL) has been implemented called IPbusClient, which provides a simple software interface for each of the IPbus instructions and, since it is very inefficient to send small Ethernet packets, concatenates the instructions until a dispatch command is called[12].

The central principle of the trigger upgrade software and firmware project is code reuse; with a common hardware platform, the core ("systems") firmware should be the same for all boards and should not need to be implemented by each subsystem that uses it. The same principle applied to software is, therefore, not only good practice but, in fact, simply a logical extension.

To accommodate the hierarchically divided bus, a recursive bus description file is required and it is, therefore, natural to use an XML file format. In accordance with the modular design principle, an XML file my include a reference to another XML file, just as an HDL file may use modules from other HDL files. The recursive structure of firmware (and resultant hierarchical bus structure) has no analogue in standard object-oriented software and, as such, a truly recursive structure, called Redwood, was created[12]. The end-points of the recursive structure are referred to as leaves and, not only have a one-to-one correspondence to registers on the IPbus, but also provide direct access to that register through the IPbusClient. In this way a register in the firmware is accessed in software in an object-oriented fashion. A factory method converts the XML description into the recursive object.

Each "branch" of the Redwood tree corresponds to a firmware module but, without specialization, is simply a container for end-points. Each branch may be specialized by the addition of "OperationObjects", which specify the behaviour of the branch when it receives a CONFIGURE, ENABLE, SUSPEND, RESUME or STOP signal. If no specialization is required for a particular signal, no OperationObject is required and the signal is simply forwarded to each child of current the branch. OperationObjects are added by means of plug-in modules loaded through dynamically linked libraries and, as such, when a plug-in is modified or new plug-in is created, only that plug-in must be recompiled. Furthermore, because each plug-in describes the behaviour of one function, the plug-ins are usually small, typically a few tens of lines, making them easy to maintain. The OperationObjects can naturally be added to the tree through the XML file and, if a new OperationObject is written for a particular module, the user chooses between them in the file.

By separating the code in this way, it is intended that the "expert" who writes a firmware module also writes any OperationObject plug-in modules that are required and an XML file containing the description of the registers within their module and which OperationObjects to use. A user can then simply drop the firmware module into their design, include a reference to the relevant XML file in their own XML description and the software will be correctly constructed for them.

In such a scenario, it could be envisaged that the user need never write an executable, merely the XML description, and, instead, a single executable handles all of the infrastructure tasks such as framework integration, network access, database access, etc. Whether this is practical or even possible is still to be determined.

All of this software functionality has been written to make the task of the end user as simple as possible, not to prevent manual access to the cards; for "non-standard" operation of hardware, such as debugging, bench top tests, etc., the user may well choose to use all, some or none of the listed functionality. For example, the MINI-T5 card currently uses an XML file to construct the tree but is still configured by explicit C++ code; the card has previously been configured using python scripts. Some, but not all, of the OperationObjects for automatic configuration of the MINI-T5 card have been implemented and the rest are currently underway.

As stated above, all of the features described are essentially independent of the choice between a conventional and time-multiplexed trigger.

11.Cost

The material cost is dominated by the choice and number of FPGAs, rather than the number of cards. At present, it is not possible to cost the proposal accurately since pricing for Xilinx Virtex-7 parts is not yet available. It is, however, possible to state that the proposed system would require 20 Main Processor cards, each with at least 52 links exceeding 10Gb/s (e.g. XC7VX485T) and 60 Pre-Processor cards with 28 exceeding 10Gb/s (e.g. XC7V285T or XC7VX585T). The final choice may also depend on the quantity of internal BRAM available, since the data rate for the level-1 accept pipelines precludes the use of conventional RAM. External RAM may still be useful for DAQ buffering, thereby freeing the DAQ BRAMs for use in the L1 pipelines.

Whilst the system has, so far, been described in terms of two cards (the PP card and the MP card), it is possible for the described MP card to be used as 2 PP cards, with the exception of the region around $\eta = 0$ which requires data duplication. The system could, therefore, be built using a single card design, with 40 MP cards replacing the 60 PP cards, although the increased number of larger FPGAs makes such a design potentially more expensive.

The total cost may be compared to a conventional trigger: both architectures typically have two stages, the first to pre-process and remap the data either spatially for the conventional trigger or temporally for the time-multiplexed trigger and the second to construct "physics" objects and sort them in order of importance.

A recent upgrade proposal[8] using a conventional trigger design, but currently neglecting the HF, requires 42 cards, each with two, large, 48-link FPGAs mounted on them, whereas the time-multiplexed design described here uses 20 MP cards, each with a single, large, 52-link FPGA and 60 PP cards, each with a single, more modest, 28-link FPGA.

12.Schedule

The current 10-year plan foresees a shutdown in 2012 to upgrade the LHC for running at 14TeV centre-of-mass energy, although, it now looks likely that this down-time will slip to 2013. During this shutdown, it is planned to replace approximately 5% of the copper links which run from the ECAL/HCAL to the RCT with an optical equivalent running at a higher line-rate, by replacing the existing SLBs with the new OSLBs. It is proposed that, either, the OSLB would have dual optical outputs, or the optical fibres would incorporate a passive splitter, a second optical output allowing a vertical slice of a new trigger system to be fully tested and verified before the second LHC shutdown, in 2016. At this time, two options would be available; to install the new trigger system and test it until the LHC restarts in the spring of 2017, or, to install the new trigger in parallel with the old trigger by either duplicating or splitting all of the ECAL/HCAL to RCT fibres. There is obviously a final and manpower cost-penalty in creating a duplicate system.

The MINI-T5 prototyping card already provides almost all the functionality needed for a final system and, thus, much of the hardware testing and firmware and software development can proceed whilst the trigger architecture is being decided and the final hardware is designed, manufactured and tested.

13.Conclusions

A time-multiplexed trigger design has been presented and shown to be feasible. The primary advantage of the design is that it is intrinsically very flexible, with all calorimeter data (12 bits per ECAL tower, 12 bits per HCAL tower) for a particular bunch crossing effectively being processed in a single location.

The design is very conservative with regards to latency; 12.5bx are reserved as contingency, if, for example, the SerDes paths take longer than expected. If the latency can be controlled as expected, then the time multiplexing period may be extended from 9bx to 18bx and the main processing node reduced to a single FPGA; an even more flexible and elegant solution than that presented. If the multiplexing period cannot be extended, the main processing node could still be reduced from two FPGAs to one by either, combining ECAL and HCAL tower energy depositions in the pre-processor cards or, reducing the energy resolution of the ECAL and HCAL towers, and sending only 12 bits of combined information per tower. This would, however, mean that the main processing nodes would not have access to the raw data. The inherent flexibility of a time-multiplexed system allows all these permutations to be implemented using the same hardware, and so, a final decision need not be made until 2015, when data and experience from several years of running at 14TeV will be available to guide the decision.

The inherent flexibility and completely generic processor nodes furthermore offer a direct path for the development and implementation of future triggers, such as tracking triggers, with all the required hardware development having already been performed. It is also not inconceivable, therefore, to consider a combined tracker+calorimeter trigger, with tracking information being processed in the same nodes as calorimeter information, or even a full tracker+calorimeter+muon trigger, should the muon trigger also be upgraded.

Progress towards a final trigger upgrade, whether time-multiplexed or conventional, has been demonstrated by the production of working hardware prototypes, core firmware/software development, algorithm implementation, etc. Work will continue to combine these into the realistic prototype system needed to develop the basic infrastructure and test new algorithms.

14.References

[1] J. Jones et al., *The GCT Matrix Card and its Applications*, TWEPP-09: Topical Workshop on Electronics for Particle Physics, Paris, France, 21 - 25 Sep 2009, pp.259-264

[2] J. Jones, *CMS: A Future Trigger Architecture*, CMS Upgrade Workshop, Fermi National Lab, USA, 28 – 30 Oct 2009

[3] http://www.xilinx.com/publications/prod_mktg/Virtex7-Product-Table.pdf

[4] http://www.avagotech.com/docs/sm-avago-supercomputing.pdf

[5] http://www.avagotech.com/docs/AV00-0169EN

[6] http://www.fonetworks.com/Fiber-Optic-RMS02_Series--SubSubCatItems.aspx

[7] http://www.fonetworks.com/Fiber-Optic-RMS04_Series--SubSubCatItems.aspx

[8] E. Hazen et al., *Development of a MicroTCA Carrier Hub for CMS at SLHC*, These proceedings.

[9] P. Klabbers et al., *CMS Regional Calorimeter Trigger Upgrade*: Hardware and Firmware Proposals and Development, 2010 CMS Internal Note (awaiting reference number).

[10] G. Iles et al., *Trigger R&D for CMS at SLHC*, TWEPP-09: Topical Workshop on Electronics for Particle Physics, Paris, France, 21 - 25 Sep 2009, pp.249-253

[11] J. Mans et al., *Simple IP-based μTCA Control System*, Version 1.2, 14 Feb 2010, http://projects.hepforge.org/cactus/trac/export/149/trunk/doc/related/Simple_IP_uTCA_Protocol_r1_2.pdf

[12] A. Rose et al., *Redwood & co.*, 13 Dec 2010, http://projects.hepforge.org/cactus/trac/export/148/trunk/doc/user_manual/Redwood.pdf

Addendum

1. A Time-Multiplexed Trigger with pre-processor data reduction

Although the inherent elegance of the time-multiplexed scheme is that no data need be discarded prior to the main processing node, it has been suggested that calorimetric clustering operations do not require the 18 to 24 bits per tower that ECAL and HCAL could potentially provide, with some considering 10 bits per tower sufficient[1]. In such a scenario, the processing card would receive an energy sum of ECAL+HCAL (9 bits) and a ratio of HCAL/ECAL (1 bit). To facilitate a direct comparison between a time-multiplexed scheme and a conventional scheme using data pre-processing, we consider here an alternative implementation of a time-multiplexed trigger which includes data reduction in the pre-processor (PP) cards. For the time-multiplexed trigger it is, in fact, simpler to send 16 bits per tower, rather than 10, and the use of 16 bits per tower is assumed here.

In sending 16 bits per tower, rather than 24, the reduction in data volume sent to the main processor (MP) node allows a substantial simplification to the time-multiplexed trigger already proposed. It is no longer necessary to split the main processing node across two FPGAs, with a single FPGA being sufficient. This, then, also removes the requirement for a large overlap region, further reducing the number of input fibres into the FPGA, making it possible to build the system with 36-link FPGAs rather than those based on 48 or more.

The direction of the time multiplexing can also been switched, so that the data is time-multiplexed in the η- rather than φ-direction, that is, data arrives in order of increasing η or increasing |η|. This allows the time-multiplexed trigger design to map onto the existing RCT crate system and, as such, the potentially large patch panel between the ECAL & HCAL and the PP cards can be avoided. Such a scheme also simplifies the PP and MP firmware design, since there is no wrap-around in the η-dimension and all PP cards would operate on topologically-identical sectors of the detector.

The option to time-multiplex in the φ-dimension remains possible because the η-φ space is essentially square and the hardware requirements for both scenarios are similar.

2. Cost

To implement such a scheme would require 36 PP cards and 12 MP cards. For now, we shall assume 14 MP cards, with at least 2 of the MP cards being redundant spares or test systems. The entire system could, therefore, be built with 50 FPGAs in 5 uTCA crates (8U) and 4 patch panels (4U). The total rack space required is just over 1 rack (56U).

The choice of FPGA for such a system is not entirely obvious. Although the simplest option is to have one large FPGA, in addition to the 36 data-links we also need a DAQ link and a Gigabit-Ethernet link, and so would require moving from the Xilinx XC7V series to the XC7VX series. This series offer either substantially more links (e.g. 56) or higher speed transceivers (e.g. 24 GTX at 10Gb/s plus 24 GTH at 13Gb/s), but are likely to incur a cost premium disproportionate to the number of extra links required. An alternative solution may be to split the design into two parts, with a relatively low-cost Kintix-series part providing the services (e.g. DAQ, Gigabit-Ethernet, embedded CPU, etc.) and a Virtex-series part for data processing. The advantage of this is that there are many 36-link parts which should be pin compatible (i.e. XC7V585T, XC7V855T, XC7V1500T and XC7V2000T) in an FFG1761 package. Of these, the XC7V855T is probably the most attractive, since it contains a relatively large amount of block ram.

3. Latency

The latency of this design, Table 3, is increased by 6 bunch crossings (bx) relative to the original design because the time-multiplex period increases from 9+1 to 12+1 bx and because it is envisaged that such a scheme would reuse the OGTI interface to the Global Trigger; the latter increase because the OGTI links are limited to a maximum of 3.2Gb/s and the GTX transceivers must, therefore, be run relatively slowly, incurring a latency penalty. Despite this, the design remains well within the latency envelope. If the resolution was lowered to less than 16 bits per tower, the latency could potentially be improved, although, packing the data efficiently would be made more difficult.

	Time (bx)
CAL/PP SerDes (Tx+Rx) at 4.8Gb/s	4
CAL/PP cable (10m)	2
PP processing + time-multiplex	13
PP/MP SerDes (Tx+Rx) at 7.92Gb/s	2.5
PP/MP cable (20m)	4
MP processing	8
MP/GT SerDes (Tx+Rx) at 2.4Gb/s	5.5
MP/GT cable	0.5
GT de-multiplex	1.0
Total	40.5

Table 3 : The latency contributions from a time-multiplexed trigger system that transmits 16 bits per tower to the Main Processor cards, rather than 24 bits per tower. The maximum latency permitted is 47bx.

4. Electron Cloud Algorithm

The electron cloud algorithm (sum of ECAL energy) is easy to incorporate into the design. After transmitting the tower data, the PP card transmits a single 32-bit word containing the sum of all ECAL energies that it received. The MP cards must then simply sum all these values together. The extra latency is less than 5ns and no additional hardware is required.

5. Conclusions

It has been shown that if it were either desirable or necessary, it would be possible to sacrifice some of the flexibility and elegance of the previously proposed time-multiplexed calorimeter trigger, to build a system which is both more compact than a conventional trigger design and still maintains a better energy resolution. Such a design suffers from additional latency compared to the previous time-multiplexed design, but is still within the prescribed latency envelope.

6. References

[1] P. Klabbers et al., *CMS Regional Calorimeter Trigger Upgrade*: Hardware and Firmware Proposals and Development, 2010 CMS Internal Note (awaiting reference number).