Available on CMS information server **CMS IN -2011/022**

CMS Internal Note

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

21 September 2011

A Grand Unified Trigger for CMS

A. Rose[1)]

Blackett Laboratory, Imperial College, London SW7 2BW, UK

Abstract

Over the next decade the LHC upgrades will increase the luminosity by a factor 5 and the pile up by a factor 5 to 10 depending on the bunch crossing spacing (i.e. 25 ns v 50 ns). This paper describes an alternative approach to triggering that will, it is hoped, help identify interesting physics events from the increasingly larger background.

Current trigger systems tend to operate by searching over a small local area for the signature of an interesting physics object. These are forwarded to the next stage where they are sorted in order of importance. The approach presented here runs many processing nodes in parallel, so that the entire calorimeter or muon detector data for a given event can be processed in one, or at most two, large FPGAs, providing a lot of flexibility in terms of algorithm design.

A design for both the calorimeter and muon trigger is presented that uses the significant improvements in the performance of FPGAs, coupled with their programmable nature, to minimise the number of card types required. The concept is extended to illustrate how a tracking trigger may be built, based around a hypothetical tracker design, requiring only a relatively modest amount of hardware, even using currently available technology. A staged installation plan is presented to illustrate how a combined calorimeter, muon and tracking trigger could be accomplished.

[1)] awr01@imperial.ac.uk

Contents

1. Introduction

The CMS trigger is expected to operate well up to the LHC design specifications, namely a luminosity of 10^{34}cm^{-2}s^{-1} and bunch spacing of 25 ns. It is now expected, however, that the LHC will exceed its design specification, with the luminosity expected to reach 2×10^{34}cm^{-2}s^{-1}. At such a luminosity, the Level-1 trigger systems will begin to experience degraded performance of the algorithms, in particular, the increase in occupancy will cause the electron- and τ-isolation algorithms to have reduced rejection at fixed efficiency and the muon trigger to have increased background rates from random coincidences.

The proposed upgrades of the CMS detector and off-detector electronics necessarily coincide with the long shutdowns of the LHC. The LHC program of upgrades may be considered to be three phases: In phase-0, expected to be in 2013/2014, the accelerator is prepared for 14TeV operation. In phase-1, expected to be in 2017, the accelerator will be prepared for higher luminosity running. In phase-2, expected to be in 2021, the accelerator will be modified for very high luminosity running.

The upgrades proposed, planned for, or affecting, the CMS trigger during each of these phases are [1]:

In phase-0, the Hybrid Photodiodes (HPDs) of the Outer Hadronic calorimeter (HO) will be replaced with Silicon Photomultipliers (SiPMs). This will provide improved capturing of jet tails and additional layers for identifying muons at trigger level, when included in the Resistive-Plate Chamber (RPC) trigger system. Some fraction of the copper serial links between the calorimeter trigger primitive generators and the regional calorimeter trigger will be replaced with optical links.

In phase-1, the photo-detectors and readout electronics in the Barrel, End-cap and Forward Hadronic calorimeters (HB, HE & HF) will be replaced, fixing the noise problems in the calorimeters. Additional, longitudinal segmentation of each calorimeter tower and the higher quantum efficiency of the new photo-detectors will provide improved energy resolution. The muon end-caps will also be upgraded in phase-1, including replacement of the ME1/1 Cathode-Strip Chamber (CSC) Electronics and installation of the ME4/2 CSC and RPC Chambers. A high-η muon system may also be added. In this phase, upgrades are planned for the Calorimeter Trigger, in order to improve the τ-trigger algorithm and provide higher position-resolution of all trigger objects; the Muon Track-Finder Trigger, in order to include the new ME1/1 and ME4/2 signals; and the Global Trigger, in order that it might exploit the new information from the calorimeter and muon triggers, and provide more complex triggering criteria than are available currently.

In phase-2, the Silicon Tracker will be replaced and its replacement will be able to provide some form of data that can be used by the Level-1 Trigger.

The schedule for the upgrades has already changed on several occasions and is likely to continue to evolve, particularly in response to new physics results, potential improvements in detector performance and changes in the LHC machine schedule. The demands on the trigger are also likely to evolve depending on the, as yet unknown, results from the existing experiment. Furthermore, there is currently no clear or comprehensive picture for what the replacement tracker or the tracking trigger would look like, what information it might provide for the other triggers or how it would be integrated with them. Following such an unclear and dynamic target demands flexibility in a way that the existing trigger does not, and in a certain sense cannot, provide. Even without considering a tracking trigger, rather than simply modifying the existing trigger to handle the additional calorimeter and muon data, the replacement trigger should aim to improve the "physics" performance over that of the current system.

Whilst always important, it is particularly important in the current economic climate to minimize the cost of the upgrade project and to minimize it across the whole system, rather than on a system-by-system basis. To this end, as far as is possible, work done in phase-0 should not need to be replaced in phase-1 and, similarly, work done in phase-1 should not need to be replaced in phase-2. Increasing the homogeneity of the trigger system by moving to fewer board designs and sharing common hardware between subsystems would, not only reduce costs by limiting the number of pre-production/small-batch runs and allow for bulk ordering, but would also improve the maintainability of the upgraded trigger over the current system.

2. Bandwidth constraints and triggering

The volume of data that may be passed to a board for a given event may be described by Equation 1.

$$N_{bits} = N_{links} \times W_{link} \times R \times \varepsilon_{encoding} \times \tau_{transmission} \qquad \text{Equation 1}$$

Where N_{bits} is the number of bits of data, N_{links} is the number of input links, W_{link} is the number of channels per link (i.e. the width of a parallel bus; unity for a single serial link), R is the line-speed of the links, $\varepsilon_{encoding}$ is the efficiency of the encoding scheme and $\tau_{transmission}$ is the transmission period. The middle three terms, W_{link}, R and $\varepsilon_{encoding}$, are determined by the choice of link technology used and, whilst the choices may be somewhat optimised, the three terms are neither entirely independent, nor entirely free. Furthermore, the number of links, N_{links}, is limited by the physical space available on the circuit board for connectors and the number of I/O resources available on the chosen processing element. Since the trigger must be free of dead-time and render a trigger-decision on each bunch-crossing, the current level-1 trigger uses a pipelined approach. This requires that, for every bunch-crossing, each processing step completes its task and is free to receive new data on the next clock; because of this, the transmission period, $\tau_{transmission}$, is restricted to a single bunch-crossing, or 25ns.

By choosing a fast, efficient and dense standard with small connectors, the volume of data which may be received onto a board may be maximized, which is important since it minimizes the amount of data which must be shared between boards to cover the boundaries between boards. The links from the detector to the trigger are constrained by the existing infrastructure and, in most cases, the data density is low, requiring many boards at the front end of the trigger to receive the data, with each board handling only a small region. In order to consider objects on the scale of interest, data from a large number of small regions must be brought to a small number of boards, each covering a large physical space. Whilst increasing the line-speed and encoding efficiency of the links can be used to increase the data density somewhat, the increase is insufficient to bring all data to a single point and, instead, data must be thrown away at each processing stage. This is done by creating coarser "higher-level" objects, sorting these objects according to some criteria and only sending a select few candidates from each region to the next processing step, the remaining candidates being ignored, Figure 1.

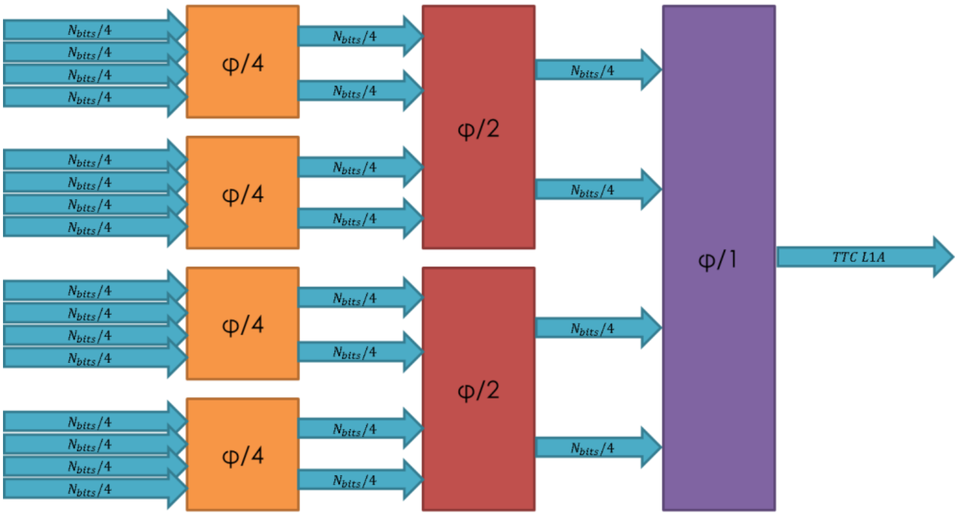

Figure 1 : A conventional trigger architecture, like the existing level-1 trigger. Each “processing node” can receive N_{bits} split across four links every bunch-crossing. Each sequential node handles a region twice the size of the previous node, but in order to be able to do so, half of the received information must be thrown away at each processing step. Not shown in this diagram is the data sharing required between neighbouring boards within each processing step that is needed to handle boundaries.

From Equation 1, it is clear that if we could increase the period over which the data is received by a board, $\tau_{transmission}$, by, say, a factor of ten, then we could similarly increase the total volume of data received. Since each board would take (for example) ten bunch crossings to receive all its data, then in order that no data is lost, it is clear that we require at least ten processing nodes operating in parallel to receive all the data. Because the data from the detector is not sent like this, a pre-processing step is required: pre-processor cards each receive the data arriving from a particular region of the detector on each and every bunch-crossing, buffer it and retransmit it over ten bunch crossings. Each pre-processor must, then, send at least one physical link to each of the parallel processing nodes, Figure 2. The data which arrived at the pre-processor on bunch-crossing one is transmitted on link one to the first processing node over ten bunch crossings, the data which arrived at the pre-processor on bunch-crossing two is transmitted on link two to the second node over ten bunch crossings, the data which arrived at the pre-processor on bunch-crossing three is transmitted on link three to the third node over ten bunch crossings and so on until the data which arrived at the pre-processor on bunch-crossing ten is transmitted on link ten to the tenth node over ten bunch crossings. On the eleventh bunch-crossing, data arriving at the pre-processor can be transmitted on link one to the first node again since, by this point, all data from bunch crossing one has been sent and received.

Since each pre-processor need only send a single link to each processing node, each processing node may receive links from many pre-processors, Figure 3, and since each pre-processor is sending all the data from a particular region of the detector, each processing node can, in this way, receive the full, raw data. With the full, raw data in a single processing node, the distinction between regional and the global stages becomes redundant and the output from each processing node is the globally-sorted list of trigger objects for a particular bunch-crossing.

The pre-processor furthermore offers the ideal location to translate the existing trigger links to a more modern, higher data-density standard, reducing the number of processing nodes relative to the number of pre-processors.

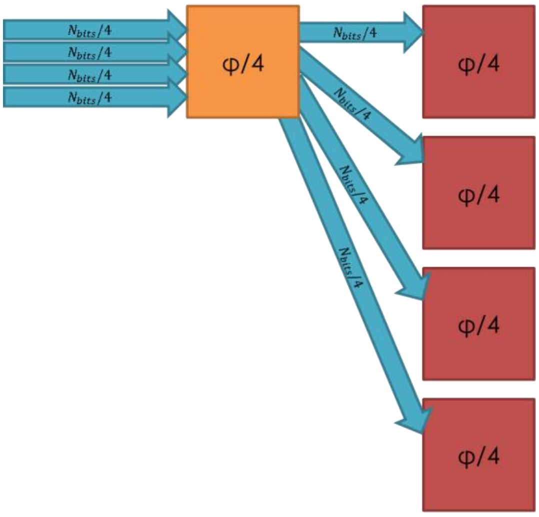

Figure 2 : A preprocessor (orange) receives N_{bits} split across four links (for clarity, could equally well be ten or any other value) every bunch-crossing and retransmits the N_{bits} to four processing nodes (red), spread over four bunch crossings. Each processing node receives $N_{bits}/4$ on its input link on each bunch-crossing so that after four bunch-crossings the processing node has all the data from the pre-processor. The input bandwidth and the output bandwidth are identical; no data is lost, rejected or thrown away by the pre-processor.

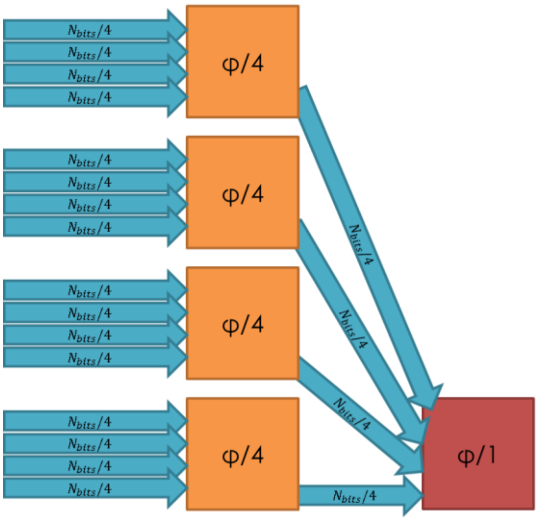

Figure 3 : Since each input link of the processing node (red) contains all the data that the pre-processor (orange) received spread over four (for clarity, could equally well be ten or any other value) bunch-crossings, by receiving a link from each pre-processor, the processing node can receive the full, raw input data.

Certain constraints exist with such an architecture. The number of pre-processor cards is determined by two factors: the total number of detector links which must be received and the number of input links on each pre-processor card. Furthermore, each processing node is required to receive a link from each pre-processor card. One does not, therefore, have total freedom over the number of links between the pre-processor and processing stages. It may be that the number of links which must be received by each processing node exceeds the number of links which may be received by a single board and each processing node must, instead, be comprised of several processing cards. Whilst this may appear to negate the original advantage of eliminating the boundaries between processing boards, the number of boards and boundaries in each node is smaller than for the equivalent conventional design by a factor at least equivalent to the time-multiplexing period. To handle the boundary region, data could be shared between the cards constituting each node, although doing so incurs an additional latency penalty. Alternatively, the relevant data may be duplicated at the pre-processor stage and be transmitted on two links, one link going to the processing card handling one side of the boundary and the other link to the processing card handling the other side, avoiding any additional latency.

Because each node is handling a different bunch-crossing, in order to send the trigger signal back to the detector, the final-processing stage must receive data from all the processing nodes, perform any remaining algorithms (such as associating candidates with those from other trigger subsystems) and make the final trigger decision, Figure 4. Since each processing node has processed the full data for its particular bunch-crossing, it may produce a globally-sorted list of candidates and need only send those of the highest-quality. As such, the bandwidth leaving the processing nodes is significantly lower than that entering the processing nodes as, in-fact, it must be in order to be able to bring candidates from every bunch-crossing to a single point.

Although the processing capacity of programmable logic devices is rising rapidly, it is still conceivable that one may wish to run more algorithms than can fit into a single chip. In such a scenario, and should the latency budget allow, processing nodes may be daisy-chained one-to-one (Figure 5) using all the available bandwidth to pass data between processing stages. These links are internal to the node and, apart from a latency cost, their presence is essentially unobservable to the outside system.

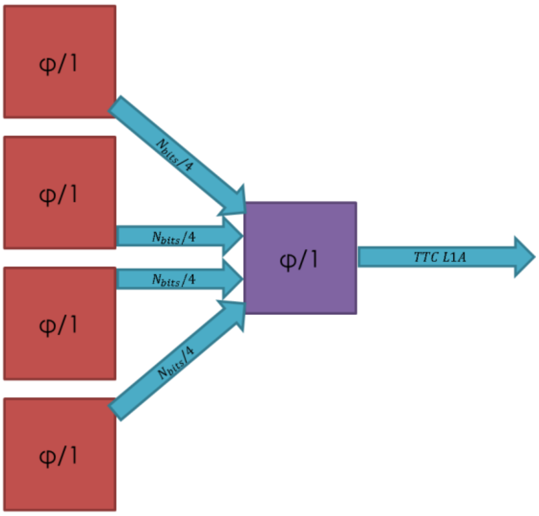

Figure 4 : Each processing node (red), having access to the raw data, can produce a sorted list of high-quality candidates for an individual bunch-crossing. A final-processing stage (purple) is required to interpret the candidates from each bunch-crossing and produce a trigger decision.

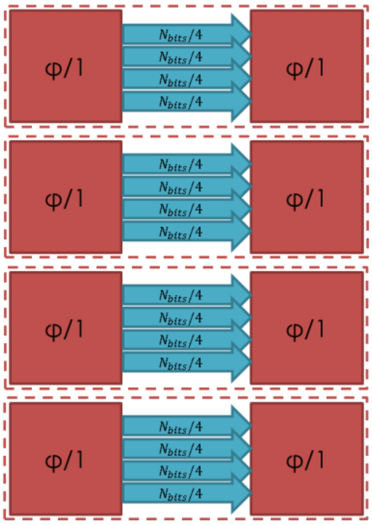

Figure 5 : If the desired algorithms exceed the capacity of an individual programmable logic device, multiple processing nodes may be daisy-chained together using all available bandwidth. As indicated by the dashed lines, each daisy-chain is self-contained and is, to the outside system, indistinguishable from an individual node with the exception of the latency cost.

When time-multiplexing over, say, 10 bunch-crossings, it may initially appear that the main-processor must wait for 10 bunch-crossings (to receive all the data) before it may proceed: this is not the case. Whilst it would be possible to do this, it is the least efficient processing model. Instead, by arranging the data in the pre-processor so that it is delivered in the order that it is used by the algorithms, each algorithm can start to run as soon as the bare minimum amount of data has been received. For triggering, the first stage of the algorithms is generally to combine data “regionally” into higher-level objects and, as such, it is most sensible for the data to be sent to the main-processor in a geometrical ordering. Each incoming trigger-primitive or piece of raw detector-data is associated, either implicitly or explicitly, with three spatial coordinates: Geometric ordering means to bin the data in three dimensions and, sequentially working along one of the dimensions, to send, in parallel, all of the bins corresponding to the other two dimensions. For example, binning the data in coordinates r, φ and η, one possible arrangement would be to send, in the first data frame, the strip at φ=0 covering all radii and the entire pseudo-rapidity range. In the second data frame, to send the strip at φ=1; in the third, the strip at φ=2, and so on. Another possible arrangement would be to send all data from the r-φ rings corresponding to |η|=0 in the first data frame, the r-φ rings corresponding to |η|=1 in the second data frame and so on.

Processing data in this fashion is referred to by some as “spatial-pipelining”.

3. Advantages and disadvantages of a Time-multiplexed Trigger

Much of the variability in the existing trigger may be traced back to the problem of handling boundaries between processing regions. With careful planning and good fortune it is possible, in some cases, to duplicate only the bare minimal subset of the links; in the majority of cases, however, the data to be duplicated is mixed with data that need not be duplicated. Boundary sharing is particularly problematic for data transmitted on serial links, since there is no passive means of separating the data within each link and there is a relatively large latency penalty for each Serialization/Deserialization step. Furthermore, as link speeds increase, the number of trigger objects per link also increases and the problem becomes worse.

The process of time-multiplexing has been shown above to eliminate, or at-least vastly reduce, the number of boundaries and, as such, virtually eliminates the need for data duplication or sharing. For a given trigger-object resolution, therefore, a time-multiplexed trigger will require fewer processing boards than its conventional equivalent.

The elimination of the boundary handling problem also eliminates the need for customizing processing boards to handle specific use-cases: all specialization may be contained purely within the firmware. In the ultimate scenario, a single design of processing board would be used reducing the overall cost by minimizing the number of different production runs and by allowing for bulk purchasing of components. Furthermore, risk would also be reduced, since only a single type of board would need to be validated and, so, may be validated more thoroughly. Homogeneity of processing hardware also allows for homogeneity of associated infrastructure and, together, these factors improve the maintainability of the system, since more users may become "experts".

The elimination of boundaries also improves the scalability and flexibility of the trigger system. Since each input on a processing node is, in essence, identical, it is trivial to add new data sources. Furthermore, because all specialization is within firmware, the algorithms can not only be easily modified but could, were it ever desirable, be rewritten in a completely different structure.

Since the purpose of the trigger is, essentially, the identification and separation of "good", "bad" and "fake" particle candidates, it is reasonable to suppose that the more information that is available to distinguish between them, the better the quality of the selection process. Although the latency requirements and finite logic availability within an FPGA constrain what may be done with the raw data, this constraint is far less severe than the bandwidth constraint between FPGAs. A time-multiplexed trigger should, therefore, be able to offer higher quality candidates for selection, although whether this offers a significant improvement in the trigger purity or efficiency has not yet been demonstrated.

In a conventional trigger, the loss of a processing node results in the loss of a region of the detector, affecting every bunch-crossing. In a time-multiplexed design, each processing node is handling a different bunch-crossing, and so loss of a node results in the loss of every "N'th" bunch-crossing, with all other crossings being unaffected. If there are sufficient output channels on the pre-processor cards, it would be possible to include additional links to several "spare" processing nodes. These nodes would normally be dormant but, in the event of a node failure, could be awakened and the data on the pre-processors rerouted from the link to the dead node onto the link to the "spare" node. This process could, then, be performed remotely during a run without the need for physical intervention. Since the spare nodes are powered, they are referred to here as "online-spares".

An alternative use for an online-spare node is as a demonstrator platform for new trigger algorithms. The spare node may receive a bitwise copy of the data from (say) the first bunch-crossing and be used to validate new firmware on real data without interrupting the operation of the trigger. Once thoroughly tested, a firmware may be safely ported to the other processing-nodes and the online-spare used for further development.

For conventional algorithms, where all the data must be processed in the same clock-cycle, many copies of the same algorithm must be run in parallel and, frequently, this requires sharing of data between multiple copies of the algorithms. This is, again, a boundary sharing problem, and is particularly problematic for the firmware synthesis tools, specifically, in making the algorithms run sufficiently quickly to meet the LHC timing requirements[2][3]. By receiving far more of the data sequentially, a time-multiplexed system is ideally suited for making use of pipelined algorithms – the favoured paradigm for firmware design. Benchmark studies[2] based on the Xilinx Virtex-5 indicate that pipelined designs use significantly fewer logic whilst also running considerably faster than the combinatorial equivalent. Furthermore, since no distinction is made between regional and global algorithms in the time-multiplexed scheme, no serialization/deserialization stage is needed between the two and an additional latency saving is made.

As stated previously, by arranging the data in the pre-processor so that it is delivered in the order that it is used by the pipelined algorithms, each algorithm can start to run as soon as the bare minimum amount of data has been received. To optimize the pipelining requires careful consideration of the detector geometry, the packing of data within data-frames and the nature of the algorithms themselves, all resulting in a not inconsiderable amount of design effort. On the other hand, conventional trigger architectures require a significant amount of design effort to handle boundaries, which the time-multiplexed design avoids.

In order to create a globally sorted list of candidates or to perform a global topological association of candidates, all candidates must be available: This requires the regional algorithms which produce the candidates to have been run on all data and it is at this point only that the time-multiplexing period introduces a latency "penalty". The ability to run the algorithms faster and the eliminated serialization stage, however, means that the overall latency is very similar to a conventional architecture.

If data from multiple time-multiplexed systems are to be merged, care must be taken that the data are time-multiplexed in the same orientation (that is, along the η-direction or around the φ-direction) to avoid introducing any additional latency penalty. Such a decision may not be simple when considering combining data from two systems with different preferred orientations.

In a time-multiplexed system, no two links join the same pre-processor to the same processing node and no data is thrown away by the pre-processor cards. As such, the bandwidth between these two stages must, at least, equal the raw, incoming bandwidth and, in order to map the data between the two stages, an elaborate optical patch-panel is required. In certain cases, it may be possible to perform algorithms on the data as it passes through the pre-processor to reduce the bandwidth which is needed to pass the data to the main processor. Such a process may significantly reduce the number of optical links and, potentially, the number of main-processor boards, although, in doing so, one is sacrificing some of the flexibility and "elegance" of a system which processes the raw data.

Finally, a time-multiplexed architecture is a new architecture for the CMS level-1 trigger and, as with all novel designs, presents some risk. This risk is not as significant as it may initially appear, since the core technologies have all been previously demonstrated in the existing trigger and are common to all trigger upgrade proposals. Any remaining risk may be reduced by the construction of a prototype system. Furthermore, the concept of processing each event in a separate node is, in-fact, already widespread in particle-physics trigger systems: a modern example is the CMS Higher-Level trigger[4][5] which processes each event in a separate node on a farm of commercial PCs and routes data through a commercial Myrinet switch. An older example is the third level trigger at Zeus, a transputer-based system which routed data through a custom-made crossbar switch[6]. The time-multiplexed architecture presented above has an important difference to both of these examples: both the volume of data being sent to each node and the latency of the algorithms within each node are fixed and, as such, the "location" of data within the system is deterministic, as it is in the current Level-1 trigger. Furthermore, because the timing is deterministic, the routing from the inputs to the processing-nodes is static, eliminating the need for active switching as required in the two examples.

4. A Time-multiplexed Calorimeter Trigger

The details of a time-multiplexed calorimeter trigger are explored in great detail in an existing CMS internal note[7], but, for completeness, the key features are included here. Discussed here is the "baseline" design featured in [7]; recent technological advances suggest it may be possible to build a far more compact Time-multiplexed trigger, but these advances are not considered here.

One possible scheme would be for data to arrive from ECAL & HCAL on serial links running at 4.8Gb/s. For the barrel and end-cap calorimeters each pre-processor would receive a ring in φ (72 towers) that was 2 towers wide in η. The pre-processor would accept data on 18 fibres from the ECAL and HCAL trigger primitive generators, with each fibre spanning 4 towers in φ and 2 towers in η, matching the current granularity of ECAL/HCAL. The data would be time-multiplexed sequentially around the ring and re-transmitted at double the rate (9.6Gb/s using 8B/10B encoding) on 10 fibres over 9 bunch crossings. 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 end-cap require 2×28 pre-processor cards (ECAL + HCAL) whereas the HF requires only 4 pre-processor cards.

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 processor node is split over two cards with each card then handling half the detector in η, with a large, 8 tower, overlap available to handle the boundary region.

Ten processing nodes would operate in a round robin scheduling manner, each only receiving data from every tenth bunch crossing. Each processing card in a processing node would receive a single link from each pre-processor 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 hardware requirements for this scheme are summarized in Table 1, and it is these numbers which will be referred to in the rest of this document.

Although the inherent elegance of the above 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, and that 16 bits per tower may be sufficient. By reducing the data in the pre-processor cards, 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. 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|, simplifying both the pre-processor and main-processing firmware design, since there is no wrap-around in the η-dimension and all pre-processor cards would operate on topologically-identical sectors of the detector. Such a scheme requires a slightly longer time-multiplexing period; 13 bunch-crossings, rather than 10.

Subsystem		Pre-processors						Processing nodes	
		Board Count	Input		Output			Node Count	Input
			Count	Rate (Gbit/s)	Count	Rate (Gbit/s)	Usage		Count
Hcal	$\eta=0$	8	18	4.8	20	9.6	90%	10	16
	$\eta\neq0$	20	18	4.8	10	9.6	90%	10	20
	forward	4	18	4.8	10	9.6	90%	10	4
Ecal	$\eta=0$	8	18	4.8	20	9.6	90%	10	16
	$\eta\neq0$	20	18	4.8	10	9.6	90%	10	20
Total		60	-	4.8	-	9.6	90%	10	76

Table 1 : Hardware summary for a time-multiplexed calorimeter trigger requiring no data reduction during pre-processing. These figures exclude any online-spare processing nodes.

5. A Time-multiplexed Muon Trigger

Since the final design for the upgraded muon system and on-detector electronics is far from clear, a time-multiplexing trigger scheme is presented here based on certain stated assumptions.

CSC Trigger

It has been proposed [8] that the Muon Port Card of the CSC track-finder will be modified to forward all 18 muon candidates to the upgraded muon trigger, rather than sending only 2 or 3. This will be done using 12-channel optics at 2.4Gbit/s. Each muon candidate is a 32-bit entity and, thus, 12 channels are required per Muon Port Card to transmit 18 candidates every bunch-crossing. There are 60 Muon Port Cards, resulting in a total of 720 channels operating at 2.4Gbit/s.

We may consider using 20 pre-processors, each with 36 input channels, to receive all the CSC muon candidates. If the data is retransmitted at 9.6Gbit/s, then clearly we require at-least 9 optical channels to provide sufficient bandwidth to retransmit all the incoming data. For time-multiplexed triggering, the incoming data from a single bunch-crossing would be retransmitted on a single output channel over a period of 9 bunch-crossings. To allow bandwidth for bit alignment commas, CRC checks etc, it is assumed that the experimental data would be transmitted sequentially across 10 output channels, rather than 9.

To process the data, 10 processing nodes would be required, each with 20 input channels, one from each pre-processor card.

DT Trigger

With the muon trigger upgrade, the Sector Collector of the DT track-finder is to be replaced, although the exact form of the replacement is still under discussion. What is known is that all data sent to the current Sector Collector is to be transmitted from the detector to the underground sorting room. Each Sector Collector receives 10 links at 480Mbit/s and, for the sake of this study, we shall assume that the upgraded Sector Collector simply retransmits the data on either a single 4.8Gbit/s link or on a pair of 2.4Gbit/s links. There are 5 Sector Collectors per sector and 12 sectors, resulting in a total of 60 optical channels operating at 4.8Gbit/s or 120 optical channels operating at 2.4Gbit/s.

If 2.4Gbit/s links are used between the detector and the underground counting room, we may consider using 4 pre-processors, each with 30 input channels, to receive all the DT muon candidates. If the data is retransmitted at 9.6Gbit/s, then we require at-least 7½ optical channels to provide sufficient bandwidth to retransmit all the incoming data. For time-multiplexed triggering, the incoming data from a single bunch-crossing would be retransmitted on a single output channel over a period of 7½ bunch-crossings.

If 4.8Gbit/s links are used between the detector and the underground counting room, we may either use 4 pre-processors, each with 15 inputs, or we may use 2 pre-processors, each with 30 inputs channels. In the former scenario, if the data is retransmitted at 9.6Gbit/s, then clearly we require at-least 7½ optical channels to provide sufficient bandwidth to retransmit all the incoming data, whilst the latter requires at least 15 output channels operating at 9.6Gbit/s to retransmit the data. In the latter case, rather than time-multiplexing over 15 bunch-crossings, the incoming data from a single bunch-crossing could, instead, be retransmitted on a pair of output channels, again over a period of 7½ bunch-crossings.

In either scenario, each processing node receives only 4 input channels (1 from each of four pre-processors or 2 from each of 2 pre-processors) and, as such, dedicating a processing card for each processing node is a particularly inefficient use of resources. We may, instead, consider sharing the processing nodes with the CSC trigger. This has the added advantage over having separate CSC and DT systems in that no external links are required to perform the data sharing required by the algorithms.

To be compatible with the CSC trigger, the pre-processors must transmit the data sequentially across 10 output channels or pairs of output channels, depending on the scenario. Each of the 10 processing nodes would, then, receive 20 input channels carrying CSC data and 4 input channels carrying DT data.

RPC Trigger

The RPC trigger system currently sends trigger data via 300 optical links from the barrel region and 72 optical links from each end-cap, with all links operating at 1.6Gbit/s. The installation of the fourth end-cap stations will require an additional 12 links per endcap and a high-η upgrade would require a further 96 links per endcap[9].

For the barrel region, we may consider using 12 pre-processors, each with 25 input channels, to receive all the RPC data. If the data is retransmitted at 9.6Gbit/s, then we require at-least 4.17 optical channels to provide sufficient bandwidth to retransmit all the incoming data. For time-multiplexed triggering, the incoming data from a single bunch-crossing would be retransmitted on a single output channel over a period of 4.17 bunch-crossings.

For the end-caps, the arrangement of the pre-processors is complicated by the staged nature of the upgrades and whether all the upgrades will occur. We can consider two scenarios: either, pre-processors are added as necessary to handle each new type of muon station, or, the initial pre-processors are installed with a sufficient number of spare inputs so that any future stations may be added. The former option is cheaper in the eventuality that the foreseen stations are not added, whilst the latter is more homogenous, has a lower latency and offers improved algorithm performance by optimizing the data-stream for pipelined algorithms.

In the first scenario, we may consider using 4 pre-processors, each with 36 input channels, to receive all the RPC data from the existing chambers. Two pre-processors with 12 input channels may then be added to handle the fourth end-cap stations[1] and 6 pre-processors, each with 32 input channels may then be added to handle a high-η upgrade. If the data is retransmitted at 9.6Gbit/s, then the three types of pre-processor require at-least 6, 2 and 5⅓ optical channels, respectively, to provide sufficient bandwidth to retransmit all the incoming data. For time-multiplexed triggering, the incoming data from a single bunch-crossing would be retransmitted on a single output channel over a period of up to 6 bunch-crossings.

In the second scenario, we may consider using a fixed number of pre-processors – 6 – for each end-cap regardless of whether the fourth end cap stations or high-η upgrade are installed. Using the current geometry, each card would accept 12 input channels. To handle the fourth end-cap stations, two further input channels are connected per pre-processor card, and to handle a high-η upgrade, 16 further input channels are connected per pre-processor card. If the data is retransmitted at 9.6Gbit/s, then the pre-processor would eventually require up to 5 optical channels to provide sufficient bandwidth to retransmit all the incoming data. For time-multiplexed triggering, the incoming data from a single bunch-crossing would be retransmitted on a single output channel over a period of up to 5 bunch-crossings.

As such, however the end-caps are installed, a total of 24 pre-processor cards are required to handle both barrel and endcap, with the data being transmitted over a period of up to 6 bunch-crossings. To process the data, each processing node must receive 24 input channels, one from each pre-processor. It is expected that the main-processor board for a time-multiplexed trigger would provide 48 available inputs and, if logic resources allowed, therefore, we could consider using the same processing nodes for handling RPC data (24 links) as for CSC and DT data (20 + 4 links).

Having the raw data from all three muon systems available within each processing node offers several advantages over having three separate systems. Primarily, it offers the possibility of forming cross-system candidates from the raw data, rather than sending up to four candidates from each system to the Global Muon Trigger and combining them there. It furthermore eliminates the need for the "cross-over" connections between the CSC and DT triggers which, not only, makes the link structure simpler than having two interconnected systems, but also eliminates ghost candidates and the need for deghosting at the Global Muon Trigger.

Global Muon Trigger

Since, in such a system, all cross-system muon candidates for a particular bunch-crossing would be contained within each processing nodes, much of the functionality of the Global Muon Trigger would also be moved into the processing nodes.

The various possible architectures for a time-multiplexed muon trigger are summarized in Table 2.

[1] Although a single card with 24 input channels may be considered to handle both end-caps, it is more likely that one would want to keep the two end-caps entirely separate.

Subsystem		Pre-processors						Processing nodes	
		Board Count	Input		Output			Node Count	Input
			Count	Rate (Gbit/s)	Count	Rate (Gbit/s)	Usage		Count
CSC		20	36	2.4	10	9.6	90%	10	20
DT	Option 1	4	30	2.4	10	9.6	75%	10	4
	Option 2	4	15	4.8	10	9.6	75%	10	4
	Option 3	2	30	4.8	20	9.6	75%	10	4
RPC	Barrel	12	25	1.6	10	9.6	41.7%	10	12
	End-cap	12	30	1.6	10	9.6	50%	10	12
Total		48 or 46	-	-	-	9.6	-	10	48

Table 2 : Hardware summary of various possible architectures for a time-multiplexed muon trigger. These figures exclude any online-spare processing nodes.

6. Integration of Time-multiplexed Calorimeter and Muon Triggers

Since each processing node is handling a different bunch-crossing, it is clear that the global trigger must receive at-least one input channel from each processing node, associate the data from each node with a particular bunch-crossing, produce a level-1 trigger decision and broadcast that decision via the TTC system.

The data from each processing node need not be transmitted across a single bunch-crossing period, but may be transmitted across the entire time-multiplexing period and received into parallel processing pipelines on the global trigger board. If 9.6Gbit/s links are used and data is transmitted over a period of 9 bunch-crossings, this corresponds to 1728 bits per link per node. This may be compared to the current trigger where, every bunch-crossing, the Calorimeter Trigger sends 448 bits to the Global Trigger and 768 bits to the Muon Trigger, and the Muon Trigger sends 104 bits to the Global Trigger.

The time-multiplexed systems described above each use 10 processing nodes; the Calorimeter trigger with two cards per node and the muon trigger with one card per node. To avoid the need for communication between processing cards, it seems reasonable to assume that each processing card in the Calorimeter Trigger could send a single link to the Global Trigger, so that 1728 bits are available for the transmission of calorimeter candidates from each η-half of the detector. It is reasonable, then, to assume that each processing card in the Muon Trigger may send up to two links to the Global Trigger, for example, one link for barrel candidates and another for end-cap candidates. As such the global trigger would need to be capable of receiving 40 optical links. If online-spare processing nodes are included, then the number of links required will also increase: if we assume that the Global Trigger could accept a maximum of 48 input links, then, using the above scheme, there are sufficient number of inputs to allow two online-spare processing nodes in both the calorimeter and the muon trigger.

The unified trigger architecture, including provision for online-spares, is shown in Figure 6. An estimated latency budget for a unified calorimeter and muon trigger is shown in Figure 7.

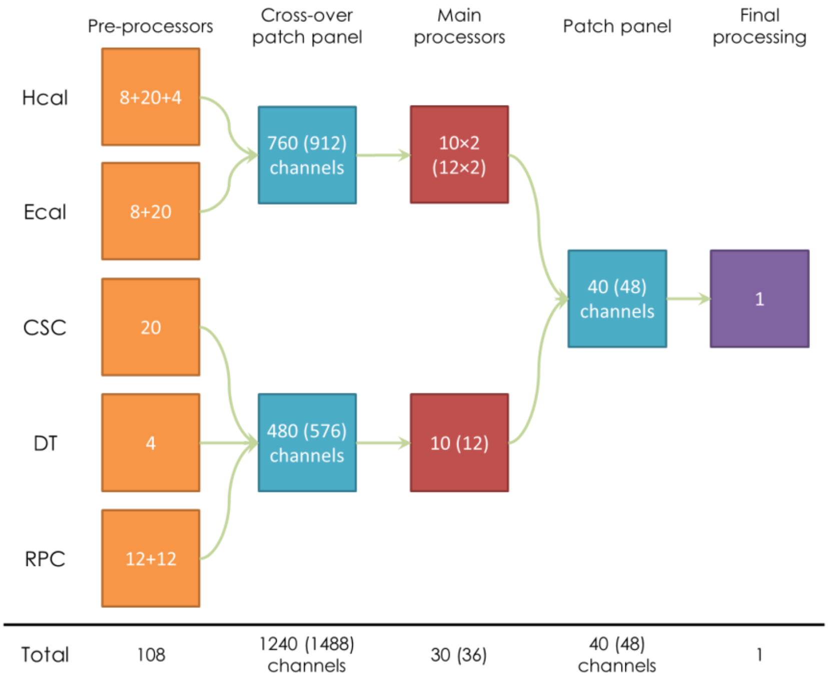

Figure 6 : Board and link counts for a unified trigger. The numbers in brackets indicate the board/link count including two online-spare processing nodes for both the calorimeter and the muon trigger. A total of 108 pre-processor cards and 30 main-processor cards are required. If online-spares are included, an additional 6 main-processor cards are required.

It should be noted that, in sections 4 and 5, no reference has been made to any particular board form-factor, link technology (other than the data-rate) or even processing technology. Since boundaries are (largely) eliminated and any which do exist may be handled by duplication at the pre-processor stage, the topology of each trigger system becomes essentially identical and may be handled with identical hardware, the differences between each system being encapsulated completely within the firmware. As such, both the Calorimeter and the Muon triggers may be constructed from two types of board; a pre-processor board and a main-processor board. The only difference between the two boards is the size of the FPGA, the pre-processor using a smaller, lower specification FPGA to reduce cost. Since both the calorimeter and muon systems use identical hardware and the number of different boards is very small, overall cost is reduced by minimizing the number of different production runs and by allowing for bulk purchasing of components, etc. It is also worth reiterating that such a move reduces risk, since fewer types of boards need be validated and so each may be validated more thoroughly.

The end result is a trigger system in which the physicist has all the calorimeter and muon trigger data in a single FPGA – or, if not a single FPGA, then in two FPGAs with a substantial overlap – coupled with the latest state-of-the-art digital signal processing capabilities. The slew of new data emerging from the existing experiment, has led to very few studies on the performance required from upgrades to CMS; and whether all this extra available processing power yields a significant physics benefit. This document has, however, demonstrated in detail how such processing capability could be brought to bear, if required.

Such a trigger design clearly presents a new way of exploiting the resources available to the trigger community. Design challenges are faced once, rather than on a per-subsystem basis. Software and firmware infrastructures are common to both triggers, reducing code repetition and improving maintainability. By eliminating the need for each subsystem to “reinvent the wheel”, effort is made available for the study, implementation and optimization of the data-processing (“physics”) algorithms. The process of designing a trigger changes from being primarily a hardware task to, instead, being a firmware task.

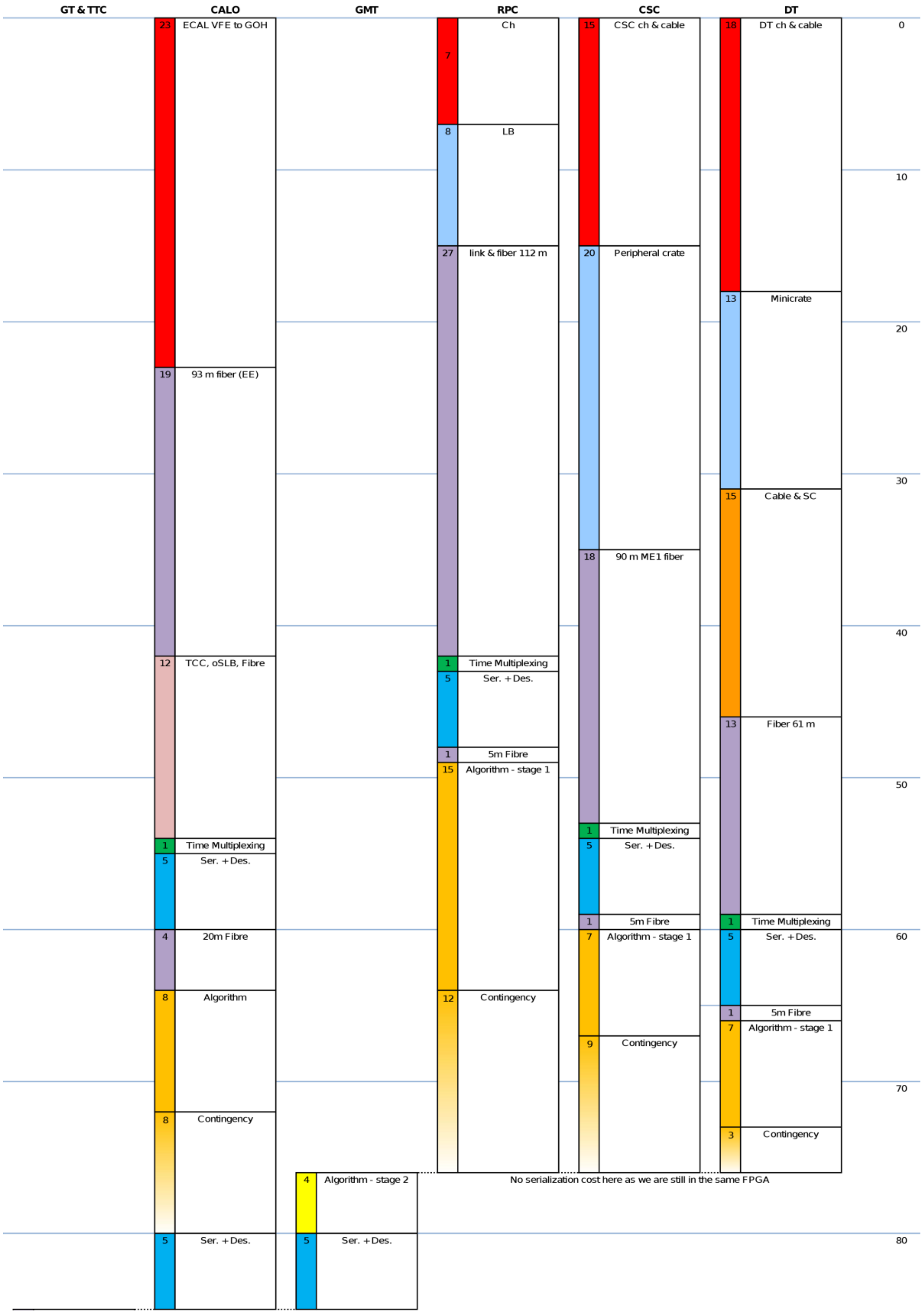

Figure 7a : Estimated latency budget of a unified calorimeter and muon trigger. Latencies for the front-end electronics are assumed unchanged from the current trigger. A worst case serialization/deserialization latency of 5 bunch-crossings is assumed. The estimated time for algorithms is based on the existing algorithms and the fact that pipelined algorithms can typically run up to five times faster than their combinatorial equivalent. A ten bunch crossing “Time-multiplexing period” is included once, in the global trigger budget, although it may equally have been drawn at the “Time-multiplexing” step. For clarity, an overlap in time is included with the next subfigure.

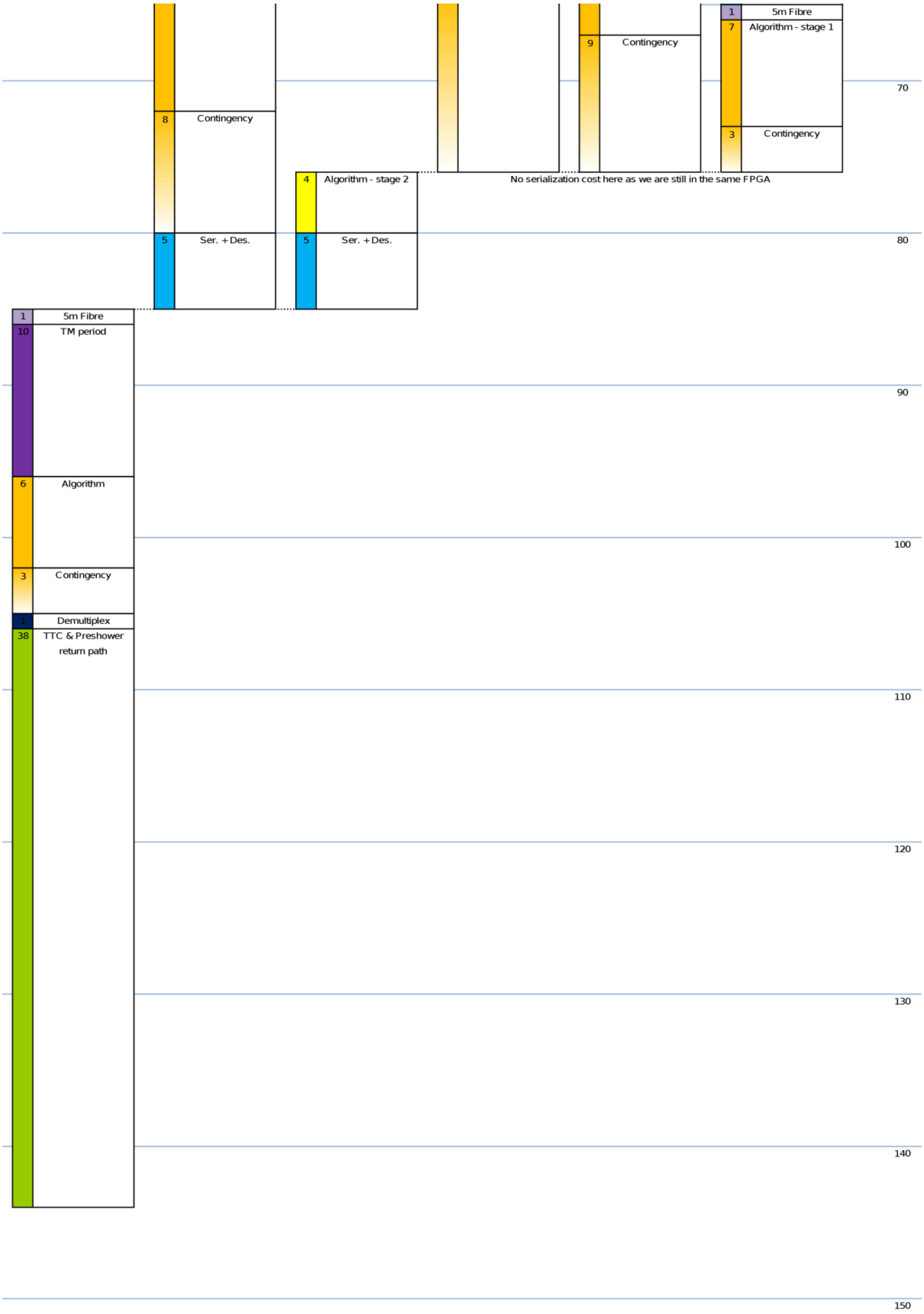

Figure 7b : Estimated latency budget of a unified calorimeter and muon trigger. Latencies for the front-end electronics are assumed unchanged from the current trigger. A worst case serialization/deserialization latency of 5 bunch-crossings is assumed. The estimated time for algorithms is based on the existing algorithms and the fact that pipelined algorithms can typically run up to five times faster than their combinatorial equivalent. A ten bunch crossing “Time-multiplexing period” is included once, in the global trigger budget, although it may equally have been drawn at the “Time-multiplexing” step. For clarity, an overlap in time is included with the previous subfigure.

7. A Tracking Trigger case-study – the "worst case scenario"

The long-term aim of the trigger is the eventual inclusion of tracker data and given that any tracking trigger will be considerably more complex than either the calorimeter or muon trigger, it is surely prudent, when considering trigger upgrades, to consider firstly, how tracker data may be received and processed, and secondly, how it may be combined with calorimeter and muon data. Because there is currently no concrete concept for the form that an upgraded tracker would take, we consider here a conceptual model representing the "worst case scenario" for the trigger, that is, one producing the largest volume of trigger data.

For most readers, the details in this section are largely unimportant and no important results are missed by skipping directly to section 8. Before doing so, it should simply be noted that, to build a tracking-trigger capable of handling raw tracker data, requires 324 pre-processor cards and 108 main-processor cards of the types proposed for the calorimeter and muon trigger, making it approximately 3 times larger than a combined calorimeter + muon system. The system requires a time-multiplexing period of 12 bunch-crossings, with 9 main-processor cards per processing node.

The assumptions made here are designed to be both realistic and conservative and, where possible, parameters have been chosen to agree with those being considered by the tracker-upgrade community. It should be noted, however, that this design has neither been presented to, nor endorsed by, the tracker community. Furthermore, no assertion is made as to the algorithmic feasibility of the tracking trigger, merely the assertion that it is possible to bring all the necessary data for track-finding into a single logic device.

The fundamental assumption for this study is the use of tracker modules of dimension 10cm in z by 5cm in rφ with the prerequisite of no more than one gigabit link per module. In rφ, it has been assumed that the strips are 98μm wide, giving 511 strips per module and thereby requiring 9-bit address precision. In z, it has been assumed that the strips are longer than 6.25mm, giving up to 16 strips per module, requiring 4-bit address precision. It is assumed that on-detector clustering will be used in the rφ-direction and that, for each cluster, the centre of the cluster will be returned, with a resolution of half a strip in rφ. It is assumed that the width of the cluster would also be returned up to a maximum value, which is an "overflow" value. For this study, it is assumed that clusters of 9 or more strips in rφ are represented by the overflow pattern. Noting that the half-pixel bit in the rφ-address is necessarily the same as the least significant bit of the cluster width, one bit may be saved by sharing this between the two. No on-detector clustering is assumed in z. As stated previously, it is assumed that there will be no more than one gigabit link per tracker module. It has been assumed that the gigabit links will provide at least 3.2 Gbit/s of useable bandwidth, corresponding to 64 useable bits per bunch crossing, or up to four 16-bit candidates per bunch-crossing per module. A tracker design with 2.4m long barrel layers, similar to the current tracker, is considered. No consideration of the end-cap region is currently included. The module counts used are based on simple tessellation, and include little overlap in rφ and no overlap in z. The number of clusters per layer used here are based on the arithmetic mean hit density per layer of the high material-budget Strawman-B tracker design at a luminosity of 10^{35}cm^{-2}s^{-1}. This assumption should represent an over estimate of the density for a light-weight tracker using on-detector clustering.

Based on these assumptions, for a luminosity of 10^{35}cm^{-2}s^{-1}, all modules outside a radius of 75cm are potentially readable and for our "worst case scenario" we shall consider a tracker with five layers at 75, 85, 95, 105 and 115cm. The choice of five layers is arbitrary and is made to maximally stress the trigger design; it is not driven by any requirement from the track trigger. For larger radii, the average number of hits per module is much lower than the four possible candidates. Whilst the simplest solution for module readout would be to have a single link per module, for the trigger and DAQ systems, the dominant cost is for FPGAs with on-board SerDes and the cost of these FPGAs increases disproportionately with the number of links. Instead we consider here a scheme of sharing links between modules for the outer layers. For simplicity, sharing is only considered between 2, 4, 8, 16, etc. modules. Taking into account the additional minimum number of bits required to address the source module, the potential for sharing and the required number of links are shown in Table 3.

radius (cm)	Potential for link sharing	Unshared readout links			Shared readout links		
		Number of links	Bits per cluster	Total bandwidth (Gbit/s)	Number of links	Bits per cluster	Total bandwidth (Gbit/s)
75	1	2304	16	7372.8	2304	16	7372.8
85	1	2592	16	8294.4	2592	16	8294.4
95	2	2880	16	9216	1440	17	4608
105	2	3168	16	10137.6	1584	17	5068.8
115	4	3504	16	11212.8	876	18	2803.2
Total		14448		46233.6	8796		28147.2

Table 3: Number of detector readout links and total bandwidths assuming the use of unshared and shared links

Over the development timescale of the tracker upgrade, FPGA technology is likely to develop considerably. The figures used in this document are based on current mid-range to high-end FPGAs on the assumption that such a technology will be similar to the low-end to mid-range FPGAs of the future. If improved technology is available and affordable in the future, then the baseline systems quoted here would be reconsidered and could be reduced in size.

We consider here that a pre-processor card receives up to 36 links running at 3.2Gbit/s and retransmits the data on up to 24 links running at 9.6Gbit/s. A main-processor card, receiving up to 48 fibres at 9.6Gbit/s and provides the trigger logic.

Within each Time-multiplexed trigger node, it has been assumed that the tracker is split into 9 wedges, each covering $\frac{1}{9}$ of the barrel in φ, the entire barrel in z and all trigger layers. A two full-module overlap (arbitrary choice) is considered on either side of each sector and is handled by data duplication in the pre-processor stage. The non-overlapping/unduplicated regions and overlapping/duplicated regions are considered separately.

For the non-overlapping/unduplicated regions, it has been assumed that each pre-processor receives a maximum of 33 detector readout links and time-multiplexes the data over 11 bunch crossings. In this scenario, the number of fibres received by the main-processor card to cover a region is $\frac{1}{33}$ of the number of detector readout links required to cover the same region. The overlapping/duplicated regions are Time-multiplexed in exactly the same way as the non-overlapping/unduplicated regions, except that each output channel is duplicated onto two fibres, one fibre going to each processing card handling either side of the boundary. The number of modules, detector links and trigger links required to process one bunch crossing of data for one non-overlapping/unduplicated region is shown in Table 4 and for one overlapping/duplicated region in Table 5.

radius (cm)	Number of modules in the rφ-direction	Number of modules	Number of detector links	Number of trigger links
75	8	192	192	6
85	10	240	240	8
95	11	264	132	4
105	12	288	144	5
115	14	336	84	3

Table 4 : Number of modules, detector links and trigger links required to process one bunch crossing of data for one non-overlapping/unduplicated region

radius (cm)	Number of modules in the rφ-direction	Number of modules	Number of detector links	Number of trigger links per main-processor
75	3	72	72	3
85	2	48	48	2
95	3	72	36	2
105	3	72	36	2
115	3	72	18	1

Table 5 : Number of modules, detector links and trigger links required to process one bunch crossing of data for one overlapping/duplicated region

Each main-processor card receives one non-overlapping/unduplicated region and two overlapping/duplicated regions, and so the total number of links received by each sector processor card is as shown in Table 6.

radius (cm)	Link count per main-processor card
75	12
85	12
95	8
105	9
115	5
Total number of links per MP card	46
Total number of links per MP node	414
Total number of PP-MP links	4968

Table 6 : The link count per main-processor card, per main-processor node and the total number of trigger links required for the illustrative tracking trigger architecture

The number of main-processor cards is 108, independent of the tracker layout, being fully constrained by the requirement of one card per sector, nine sectors per time-multiplexed node and a total of twelve time-multiplexed nodes (excluding online-spares).

The number of pre-processor cards is dependent on the number of detector links and thus on the tracker architecture, Table 7.

radius (cm)	Number of pre-processor cards (non-overlap)	Number of pre-processor cards (overlap)	Number of pre-processor cards (total)
75	54	27	81
85	72	18	90
95	36	18	54
105	45	18	63
115	27	9	36
Total number of pre-processor cards			324

Table 7 : The number of preprocessor cards required for the illustrative tracking trigger architecture

8. A Grand Unified Trigger for CMS

Since the output from the tracking trigger must consist of at least one link from each main-processor card, the number of output links will be a multiple of 108. Rather than trying to bring all of these links to a single final-processing board for combination with the calorimeter and muon data, we will use the widely held assumption that the processing time will double or possibly quadruple by the time a tracking trigger is built, to introduce an intermediate processing stage where the data are merged. The nature of the intermediate processing stage is determined by whether the tracking trigger can construct track-candidates without seeding from the calorimeter and muon systems.

An unseeded tracking trigger

If the tracking trigger can construct track candidates without external seeding, then an elegant solution is to consider introducing twelve "global-processing" nodes, each consisting of a single main-processor card. Each global-processing node would receive data from only a single tracking trigger node and so would always be processing the data from the same bunch-crossing as that tracking trigger node. Because, each global processing node is connected to only a single trigger node, the data may be transmitted between the two over the full time-multiplexing period of 11 bunch-crossings[1]. Furthermore, because there is no connection between different processing nodes, the arrangement of the optical links is trivial and no elaborate optical patch-panel is required.

We can consider each global-processing card receiving four links from each of the nine cards within its corresponding tracking trigger node. The data may be received over 11 bunch-crossings corresponding to 8,448 bits per ninth of the detector or 76,032 bits of track-candidate data for each bunch-crossing.If we then consider a time-multiplexed calorimeter and muon trigger as described previously, but with each consisting of 12 nodes, rather than 10, it is clear that each calorimeter trigger node and each muon trigger node may also be uniquely associated with a single global-processing card. We may then consider sending 6 links from each calorimeter trigger node and from each muon trigger node to their associated global-processing card, corresponding to 12,672 bits of calorimeter-candidate data and 12,672 bits of muon-candidate data for each bunch-crossing.

Since each global-processing node has all the candidates from all three systems from a single bunch-crossing, it has everything it needs to make the final trigger decision. Each node need, therefore, send only a single link to the final-processing stage, where the trigger decisions are associated with the correct bunch-crossing before being sent to the TTC system. The final processing stage in this design may use the same hardware as for the unified calorimeter + muon trigger and all changes encapsulated in firmware. This scheme is shown in Figure 8.

[1] It should be noted, here, that there is no latency penalty associated with transmitting across 11 bunch-crossings, since the candidates may be transmitted as soon as they are produced.

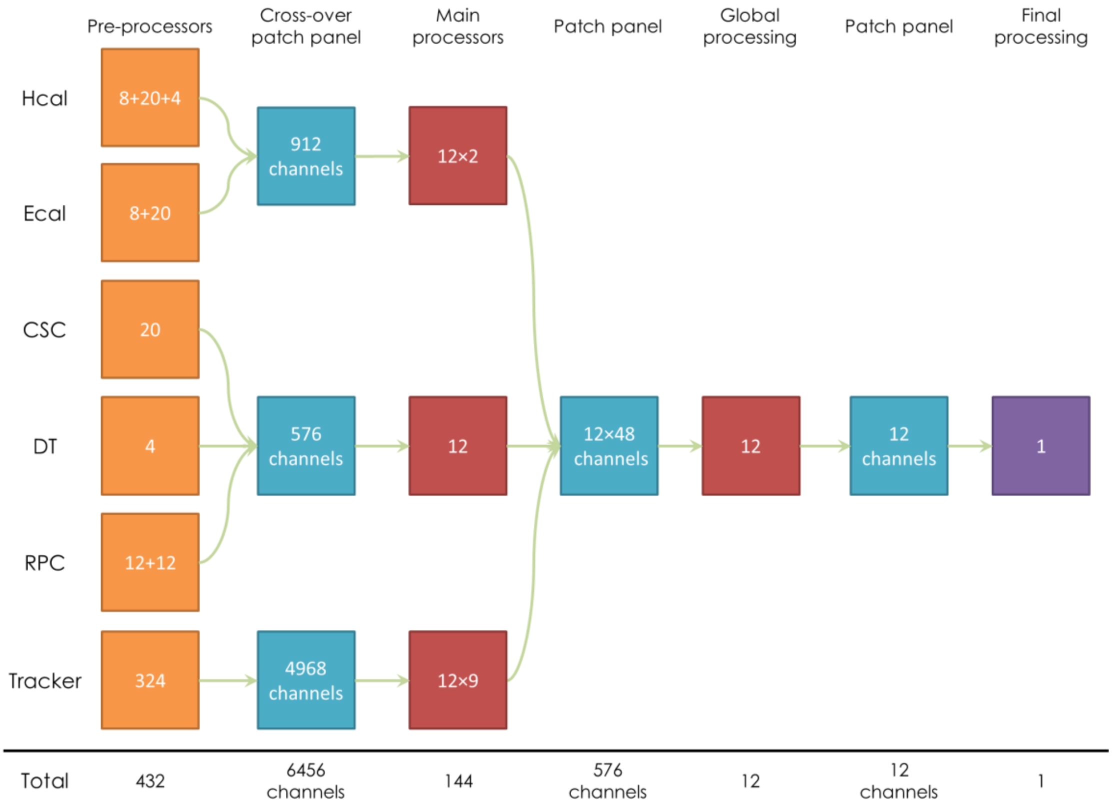

Figure 8 : Board and link counts for a grand unified trigger where the tracking trigger can construct track candidates without external seeding. No online-spares are included in this diagram.

A seeded tracking trigger

If the tracking trigger cannot construct track candidates without external seeding, then the calorimeter and muon candidates must be used as inputs to the tracking trigger. Each calorimeter and muon trigger node must send the candidates from a particular region to the associated tracking trigger processing card, requiring an additional four inputs per tracking trigger processing card, corresponding to 4,224 bits of calorimeter-candidate data and 4,224 bits of muon-candidate data per ninth of the detector.

Since each tracking trigger processing card only handles a ninth of the detector, it cannot make the final trigger decision. Each processing card must, therefore, send a list of candidates to the final-processing stage, where they must be globally sorted before any trigger decision is made, associated with the correct bunch-crossing and sent to the TTC system. Such a final-processing stage would require replacing the final-processing stage used in the unified calorimeter + muon trigger. This scheme is shown in Figure 9.

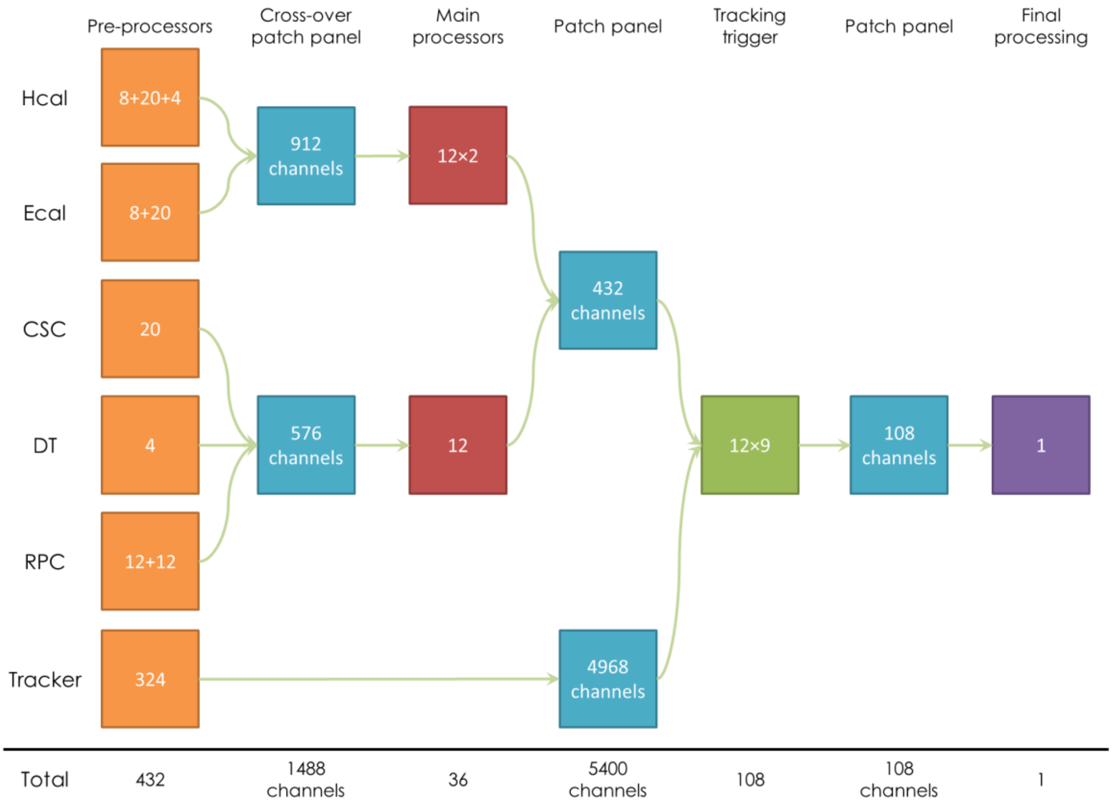

Figure 9 : Board and link counts for a grand unified trigger where the tracking trigger requires seeding from the calorimeter and muon triggers. The tracking trigger processor boards are coloured green to indicate that they require more input channels than the calorimeter or muon processor boards. No online-spares are included in this diagram.

9. Staged installation plan

Based on the schedule presented in section 1 and upon the asserted aim that work done in phase-0 should not need to be replaced in phase-1 and, similarly, work done in phase-1 should not need to be replaced in phase-2, we may consider how a time-multiplexed trigger may be installed.

In phase-0, some fraction of the copper serial links between the calorimeter trigger primitive generators and the regional calorimeter trigger are replaced with optical links. Duplication of these links, whether active or by passive splitting, allows the installation of a "test-slice". Running in parallel with the existing trigger, allows such a slice to, not only, demonstrate the technologies and infrastructure needed for a trigger upgrade, but also to demonstrate that the performance may firstly be replicated, and then bettered. For a time-multiplexed system, such a slice clearly requires sufficient pre-processors to receive all the optical links, but the number of main-processing nodes is flexible. For instance, if the installation of only one or two nodes is feasible, then the trigger principle may still be demonstrated for one or two bunch-crossings in every ten. If it is possible to install all ten nodes, then the trigger may be demonstrated without dead-time.

In phase-1, the test slice is expanded to replace the entire calorimeter trigger, requiring the replacement of the remaining copper serial links from the ECAL trigger-concentrator card. The HCAL trigger primitive generators will be replaced and will provide a direct optical output. Additional pre-processors would be installed in parallel with those already installed in the test slice. Since there is no interaction between pre-processor cards, installation of additional cards does not necessitate modifying or changing those already installed. If only a subset of the main-processing nodes have been installed, then the additional nodes must also be installed and, since there is no interaction between main-processing nodes, installation of additional nodes, again, does not necessitate modifying or changing those already installed. All connections between existing and newly-added cards are made at the patch-panel interfacing the pre-processor cards and the main-processing nodes.

For the muon trigger, an outright replacement is expected in phase-1, to coincide with the installation of new detector elements and upgrades of the front-end electronics. By using the same hardware as has already been validated in the calorimeter test-slice, the risk inherent in an outright replacement may be reduced.

To integrate the new calorimeter and muon triggers with the global trigger requires that the global trigger receives 48 optical links. Rather than replacing the entire global trigger and central trigger control system, it may be sufficient to upgrade only the Final Decision Logic board in the existing system.

In phase-2, the tracking-trigger may be installed, again, without requiring change to those processing boards already installed. The 48 optical links from the calorimeter and muon triggers to the global trigger must be removed and the outputs from the calorimeter and muon triggers routed to either the global-processing nodes or tracking-trigger nodes, depending on the chosen architecture. The outputs from the global-processing nodes or tracking-trigger nodes must then be routed to the global trigger. A scheme based on global-processing nodes offers a more elegant upgrade than a seeded tracking-trigger scheme, since the existing phase-1 trigger algorithms may be simply be ported from the global trigger card to the global-processing nodes, before being augmented to use the tracker data. The global trigger then need simply receive the trigger decision from each node, associate them with the correct bunch-crossing and send them to the TTC system. Furthermore, the use of global-processing nodes (rather than a seeded tracking-trigger) does not necessitate a second replacement of the Final Decision Logic board, although it is possible, or even probable, that the TTC system will also be replaced, and so require a new interface irrespective.

Recent rapid progress in increasing the LHC luminosity has led to the suggestion that the trigger upgrades should be moved forward from phase-1 to phase-0. In such a scenario, the use of a common hardware platform is even more desirable since, under such a schedule, time and resources are even more highly constrained. By providing a common trigger platform, each subsystem is freed to focus on upgrading the detector and the front-end electronics. As stated previously, out-right replacement of the trigger represents an inherent risk, especially without the reassurance of a previously validated test-slice, and a parallel installation with a period of validation is favourable.

Since the copper serial links between the calorimeter trigger primitive generators and the regional calorimeter trigger are to be replaced with optical links anyway, this presents a reasonable opportunity to duplicate the data, as was planned for the test slice. The CSC track-finder requires a replacement of the Muon Port Card for any upgrade and, by running the new links in parallel to the existing links, could allow the validation of the new CSC track-finder without affecting the existing trigger infrastructure. Similarly, it has been stated that the Sector Collector of the DT track-finder must be replaced. This, again, presents an opportunity to operate both old and new triggers in parallel. The existing RPC trigger already runs its serial links to active splitter boards which duplicate the data and distribute it to the various trigger boards. If there is sufficient spare capability in these boards it may be possible to feed a copy of each incoming link to the new trigger. If there is not sufficient spare capability in the splitter boards then an alternative solution must be found.

To run the existing trigger and the new trigger in parallel requires that the Final Decision Logic of the global trigger be capable of accepting both the existing 128 links from the Global Trigger Logic and the additional 48 optical links from the new calorimeter and muon trigger nodes. Alternatively, if the latency is acceptable, a single main-processor card may be used to generate the trigger decision and the result passed to the existing Final Decision Logic as a technical trigger, or such like.

Estimates for the amount of hardware required for a like-for-like replacement of the existing trigger and for a time-multiplexed trigger requiring no data reduction in the calorimeter pre-processors are made in Table 8.

System	Like-for-like replacement		Time-multiplexed trigger	
	Board	Count	Board	Count
RCT/GCT	Optical SLB	Dependant on upgrade path	Optical SLB	Dependant on upgrade path
	Cal. Trig Processor	24	Pre-processor Card	60
	Sharing/IO card	16	Main-processor Card	20
	Summary Crate Cards	unspecified		
CSC	Muon Port Card Replacement	60	Muon Port Card Replacement	60
	Receiver Mezzanine	7 × 20	Pre-processor Card	20
	Sector Processor FPGA Mezzanine	2 × 20		
DT	Sector Collector Replacement	60	Sector Collector Replacement	60
	Track-finder Card	24	Pre-processor Card	4
	Sorter Card	unspecified		
RPC	PAC Mezzanine (No detector mods.)	5 × 7 × 12	Pre-processor Card	24
	PAC Mezzanine (Extra stations)	5 × 2 × 12		
	Trigger Board (Extra stations)	2 × 12		
	High-η upgrade (currently unclear)			
GMT/GT	RJ45 Conversion Card	6	Main-processor Card	10
	Hirose Conversion Card	2	Final Decision Logic Board	1
	GMT Conversion Cards	unspecified		
	μTCS, μGTLFDL, μGMT	3 (commercial boards)		
Total Boards	more than 939		259 (excl. Optical SLBs)	
Total Types	more than 16		5	

Table 8 : Best estimates for the amount of hardware required for a like-for-like replacement of the existing trigger and for a time-multiplexed trigger requiring no pre-processing of calorimeter data. If data reduction is performed in the calorimeter trigger pre-processor stage, the number of Pre-processor Cards in the RCT/GCT replacement drops from 60 to 36 and the number of Main Processor Cards drops from 20 to 13. Recent technological advances suggest it may be possible to build an even more compact Time-multiplexed trigger whilst maintaining full data-resolution: these advances are not considered here.

10.Conclusions

The factors affecting the concentration of data from distributed sources have been explored in an abstract context and from this the concept of time-multiplexed triggering derived.

The advantages and disadvantages of time-multiplexed triggering have been discussed, and it has been demonstrated that the reduction or elimination of boundaries within the trigger improves the homogeneity of the trigger hardware and associated infrastructure, thereby improving the maintainability of the system. Furthermore, by reducing the number of types of hardware to two cards, total cost and risk are reduced. The elimination of boundaries also improves the scalability and flexibility of the system by promoting equality of data. A time-multiplexed system also makes natural provision for online-spare processing nodes which may be remotely enabled in the case of a node failure at run-time. Even if no online-spare nodes are included, loss of a processing node results in dead-time of the trigger, not complete loss of the trigger. It has been shown that by pipelining the data spatially, the time-multiplexing period introduces a latency "penalty" only once, when performing "global" algorithms such as global sorting or global topological triggers. The ability to run the algorithms in a pipelined fashion and the eliminated serialization stage, however, means that the overall latency is very similar to a conventional architecture. To optimize performance of a time-multiplexed trigger, the pipelining of the data requires careful consideration of the detector geometry, the packing of data within data-frames and the nature of the algorithms themselves, all resulting in a not inconsiderable amount of design effort. Furthermore, if data from multiple time-multiplexed systems are to be merged, care must be taken that the data are time-multiplexed in the same orientation to avoid introducing any additional latency penalty. To route the data from the pre-processor boards to the processing nodes requires a large and elaborate optical patch-panel although, in certain cases, the number of links and the size of the patch-panel may be reduced by performing data-reduction on the pre-processor, at the expense of some of the flexibility and "elegance" of a system which processes the raw data. Finally, although time-multiplexed-like "event-builder" architectures have been used on other experiments, it is a new architecture for the CMS level-1 trigger and, as with all novel designs, presents some risk. This risk is not as significant as it may initially appear, since the core technologies have all been previously demonstrated in the existing trigger and are common to all trigger upgrade proposals. Any remaining risk may be reduced by the construction of a prototype system. This is currently in progress.

The theory of time-multiplexing has been applied to the calorimeter trigger and to the muon trigger and it has been shown that the bandwidth constraints of both systems are amenable to a time-multiplexed approach. If no data reduction is performed in the pre-processor cards, both the calorimeter and muon trigger may be built using the same processing boards, requiring a total of 108 pre-processor boards and 30 main-processor boards. If data reduction is performed, these numbers may be reduced.

A realistic, "worst-case scenario" trigger-data providing tracker has been "designed" to stress the time-multiplexed trigger concept and it has been shown that even using currently available technology, a time-multiplexed trigger to handle such a design requires only a relatively modest amount of hardware; being approximately three times the size of a unified calorimeter + muon trigger.

Two concepts for merging the tracking data from the conceptual tracking trigger with the calorimeter and muon data have been presented, indicating the flexibility of a time-multiplexed design.

An outline has been given for how a time-multiplexed trigger might be installed under two different upgrade schedules and a comparison made of the hardware requirement for a like-for-like replacement of the existing trigger and for a time-multiplexed trigger.

References

[1] S. Dasu, *Upgrade Physics*, https://twiki.cern.ch/twiki/bin/viewauth/CMS/UpgradePhysics

[2] J. Jones, *Results from Calorimeter Trigger Algorithms on Xilinx V5*, Phase I Triggers and Platforms, 19th November 2008, https://indico.cern.ch/conferenceOtherViews.py?view=cms&confId=45544

[3] A. Rose, *Calorimeter Trigger Algorithms in Firmware*, Trigger Upgrade for Phase I/Phase II, 28th April 2010, https://indico.cern.ch/sessionDisplay.py?sessionId=14&confId=74957#20100428

[4] S. Cittolin et al., *CMS trigger and data-acquisition project: Technical Design Report, volume 2: Data-acquisition and High-Level trigger*, 2002, ISBN 9290831114, CERN-LHCC-2002-026 ; CMS-TDR-006-add-2

[5] T. Virdee et al., *CMS High Level Trigger*, 27th June 2007, LHCC-G-134. CERN-LHCC-2007-021

[6] U. Behrens, *The event builder of the ZEUS experiment*, Conference on Computing in High-Energy Physics, Annecy, France, 21st – 25th Sep 1992, pp.162-166, DESY-92-150

[7] G. Iles, A. Rose et al., *A Time-multiplexed Calorimeter Trigger for CMS with Addendum*, CMS IN-2011/008

[8] M. Matveev and P. Padley, *Upgrade of the CSC Endcap Muon Port Card at CMS*, Journal of Instrumentation, Volume 5, November 2010, doi:10.1088/1748-0221/5/11/C11013

[9] K. Buńkowski, Personal Communication

Acknowledgements

The author would like to thank Dr K. Buńkowski for his communications regarding the proposed upgrades of the RPC trigger and to thank the following colleagues for their comments and feedback during the writing of this note: (alphabetically) Mr C. Hunt, Dr G. Iles, Dr J. Jones, Dr J. Marrouche, Dr D. Newbold and Dr A. Tapper.