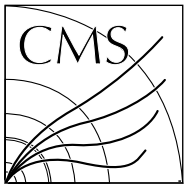
Available on CMS information server

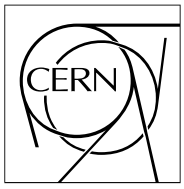
CMS NOTE 2007/018



The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland



July 18, 2007

The design of a flexible Global Calorimeter Trigger system for the Compact Muon Solenoid experiment

J.J. Brooke, D.G. Cussans, R.J.E. Frazier, G.P. Heath[a)], B.J. Huckvale, S.J. Nash, D.M. Newbold[b)]

H.H. Wills Physics Lab, University of Bristol, Tyndall Avenue, Bristol BS8 1TL, UK

S.B. Galagedera, A.A. Shah

Science and Technology Facilities Council, Rutherford Appleton Laboratory, Didcot OX11 0QX, UK

Abstract

We have developed a novel design of triggering system as part of the pipelined hardware Level-1 trigger logic for the CMS experiment at LHC. The Global Calorimeter Trigger is the last element in the processing of calorimeter data, and provides most of the input to the final Level-1 decision. We present the detailed functional requirements for this system. Our design meets the requirements using generic, configurable Trigger Processing Modules built from commercial programmable logic and high-speed serial data links. We describe the hardware, firmware and software components of this solution. CMS has chosen an alternative solution to build the final trigger system; we discuss the implications of our experiences for future development projects along similar lines.

[a)] Corresponding author, email: Greg.Heath@Bristol.ac.uk.

[b)] Also at STFC-RAL, Didcot.

1 Introduction

Recent advances in commercial microelectronics technology have had a significant impact on the design of hardware triggering systems for large high-energy physics detectors. Where complex logic previously required the design of custom integrated circuits, it is now possible to implement sophisticated pattern recognition algorithms meeting very tight timing constraints using programmable logic components. The increasing speed and density of data interconnect technology is equally important in allowing designers to build compact systems offering the high levels of performance required.

The triggering challenge is particularly acute for experiments at hadron colliders. In this paper we discuss a component of the central Level-1 trigger logic for the Compact Muon Solenoid (CMS) [1] detector at CERN's Large Hadron Collider (LHC) [2]. The LHC will deliver collisions between bunches of protons every 25 ns, with around twenty low transverse momentum interactions per bunch crossing at design luminosity. The trigger system must efficiently identify events of interest for a wide range of physics analysis channels, while limiting the rate of writing to permanent storage; the CMS computing system is now required to cope with event rates no higher than 300 Hz.

The trigger component that forms the subject of this paper is called the Global Calorimeter Trigger (GCT). We have developed a system design for the GCT based around the use of a single type of generic processing module, which may be reconfigured to perform a variety of trigger functions. The resulting system is well-integrated and compact; the required processing functions can be performed in a single crate of electronics; and the use of programmable hardware allows us to embed a comprehensive suite of system control, monitoring and test functionality alongside the trigger processing. This approach brings with it a number of design challenges. In the implementation of the trigger processing function, the main challenge is to achieve high enough communication bandwidth to make effective use of the processing power of modern programmable logic. This applies both to the input data and to the data exchanged between processing elements in our system. The design of the generic module is rather complex. The system is compact, with a high density of functionality; this imposes stringent requirements on aspects of electronics system design such as power distribution. The control, monitoring and test requirements are significant, and lead to large investment of effort in firmware and software engineering, on a scale rarely encountered either in previous HEP triggers or in commercial applications of these technologies.

This paper is structured as follows. We discuss the overall CMS Level-1 hardware trigger in this introductory Section, and briefly explain the GCT trigger function. We describe the calorimeter geometry from a trigger point of view. In Section 2 we look in detail at the processing requirements for the GCT, and describe the generic module concept developed to meet these requirements. In Section 3, we discuss the design of a system built from these modules, including the additional data paths and interface modules required. In Section 4, we turn to novel firmware and software developments. We describe our experiences with the implementation and testing of the system in Section 5, and present some brief conclusions in section 6.

1.1 The CMS trigger

CMS uses two levels of trigger to select events for permanent storage. The Level-1 trigger [3] is required to reduce the event rate to no more than 100 kHz using custom, pipelined hardware processors. The events accepted by Level-1 are then further processed in a farm of commercial PCs, which implement the High-Level Trigger [4] algorithms in software.

The Level-1 trigger uses a subset of the data from the muon detectors and calorimeters. The Level-1 latency is required to be shorter than 128 LHC bunch-crossings (3.2 μs) in order to limit the length of on-detector readout buffers. No significant deadtime is allowed; the system is therefore implemented using fully pipelined logic synchronised to the LHC bunch clock. The trigger algorithms are based upon the identification of "trigger objects", and on measurement of missing and total transverse energy. Trigger objects are candidates for later reconstruction as specific final state particles, including muons, electrons and photons (e/γ), and different classes of hadron jets. They are identified separately in the muon detector and calorimeter trigger subsystems; the Global Trigger combines information from the two subsystems, and makes a Level-1 trigger decision based upon object transverse energies and event topology. The Level-1 trigger architecture is shown in Figure 1.

This paper focuses on the calorimeter trigger, which delivers all of the Global Trigger input apart from muon candidates. The following processing stages comprise the calorimeter trigger: the Trigger Primitive Generator system processes the digitised calorimeter signals, to produce summary trigger input data with reduced precision in both transverse energy and position; the Regional Calorimeter Trigger implements e/γ identification algorithms, and sums the energy in larger regions; finally the Global Calorimeter Trigger finds jet candidates and energy sums,

and sorts the trigger objects, selecting a fixed number of each type for output to the Global Trigger.

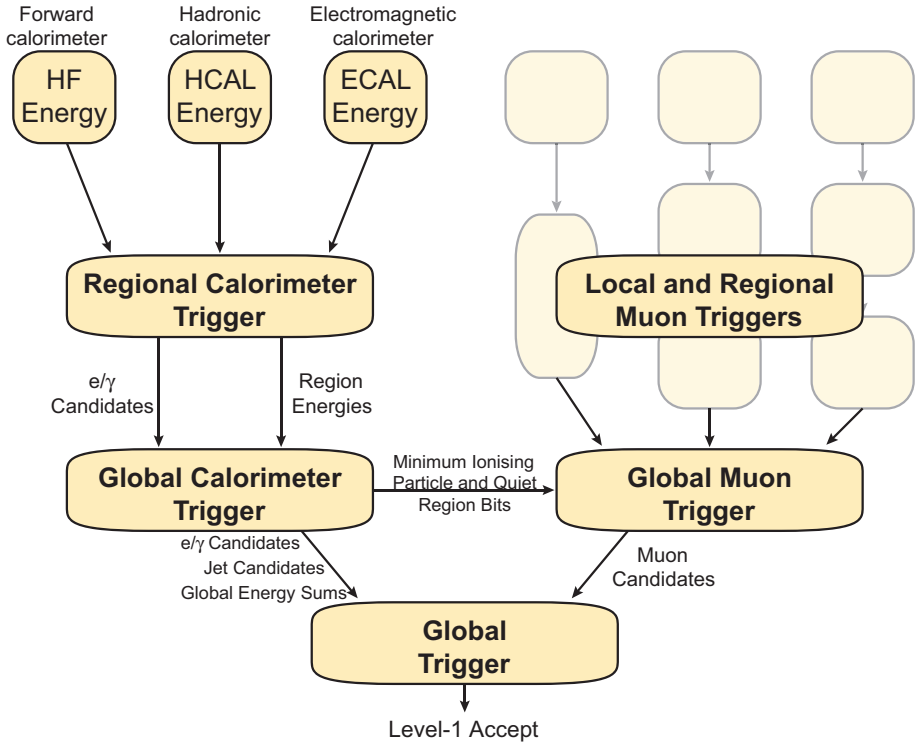


Figure 1: The CMS Level-1 Trigger. Details of the muon trackfinding have been suppressed for clarity.

1.2 The Global Calorimeter Trigger

The GCT is the final component in the Level-1 calorimeter trigger chain. Its purpose is to implement the stages of the trigger algorithms that require information from the entire CMS calorimeter system. The GCT receives a "map" of the energy distribution found in the calorimeters, on the basis of which it identifies and classifies jet trigger objects; it also receives lists of e/γ objects. It sorts the jet and e/γ objects, and forwards the most significant ones to the Global Trigger. The energy map is also used to generate several global quantities indicating the level of activity in the event, including the missing and total transverse energy sums.

The GCT also monitors activity rates in the calorimeter systems, providing an online estimate of relative LHC luminosity. Other requirements include the need to pass information on identified trigger objects for storage in the CMS event record. This facilitates monitoring of trigger performance, and may be used to seed higher level pattern recognition. The GCT must also operate reliably as a part of the CMS online environment; it must respond to control commands, and has to monitor and report its status to central hardware and software systems.

1.3 Calorimeter trigger geometry

CMS has high-resolution calorimeters covering the pseudorapidity[1)] range $|\eta| \leq 3$, and coarser-grained forward coverage up to $|\eta| = 5$. The calorimeter trigger examines the patterns of transverse energy deposition and finds candidate electrons or photons, jets and missing $\mathrm{E_T}$. Electron/photon recognition is implemented over the range $|\eta| \leq 2.5$, corresponding to the angular coverage of the CMS tracking system; jets are found over the full calorimeter acceptance.

All information input to the GCT is identified with a specific geometrical subdivision of the calorimeter, defined by intervals in azimuth angle ϕ and pseudorapidity η. We refer to these subdivisions as "trigger regions"; a region extends over a range $\Delta\phi = 20^\circ$ (0.35 radians) and a $\Delta\eta$ slice of 0.35–0.5. A total of 18×22 regions covers the

[1)] Pseudorapidity is defined: $\eta = -\ln\tan\frac{\theta}{2}$; where θ is the polar angle of particle tracks with respect to the LHC beam axis. $\eta = 3(5)$ corresponds to particles travelling at an angle of 5.7° (0.77°) to the axis.

whole calorimeter system, with 14 regions in η in the high resolution central portion of the calorimeters ($|\eta| \leq 3$), and four in each of the forward detectors.

2 Trigger processing in the GCT

In this section, we describe the various processing paths required to carry out the full GCT trigger function. We then discuss the implementation of the required functionality using commercial programmable logic devices and high speed data links. This leads on to the layout of the processing using generic processing modules. We focus on the trigger function, postponing the discussion of other aspects of the design of the generic module to Section 3.

2.1 Trigger processing functionality

2.1.1 Input data

The Regional Calorimeter Trigger (RCT) performs e/γ candidate object recognition, and sums the transverse energy in each trigger region for input to the jet and energy sum processing. Electron/photon candidates are identified as narrow energy deposits in the electromagnetic portion of the calorimeter, with little associated activity in the hadronic portion. Those with low activity in the surrounding trigger towers are flagged by the RCT as "isolated"; the GCT receives separate isolated and non-isolated e/γ streams. The region energy sums are formed by adding together the transverse energy in the electromagnetic and hadronic portions of the calorimeter contributing to each trigger region.

The RCT is organised into eighteen crates of electronics, with each crate covering 40° in ϕ and half the η range of the detector. The coverage of one crate corresponds to 2×11 trigger regions. Each crate sends to the GCT its four highest-ranked electrons, in each of the isolated and non-isolated streams. For each region the total transverse energy is calculated. The RCT also produces flags identifying regions with low calorimeter activity, to be used by the muon trigger logic.

In total, each RCT crate sends 348 bits of data to the GCT for every LHC bunch crossing (25 nanoseconds); the transverse energy sums in calorimeter regions account for nearly 70% of these bits. The aggregate input data rate to the GCT is 250 Gbit/s.

2.1.2 Trigger algorithms

The algorithm specifications for the GCT have been developed through a process of evaluating simultaneously the physics performance, via Monte Carlo simulation, and the feasibility of implementation in synchronous, pipelined logic, via firmware modelling. A number of new requirements have emerged from physics studies since the initial design of [3], and have been incorporated into the system. The implementation is made flexible and fast through the extensive use of lookup tables for calculation.

Jet finding The jet processing is based on a "sliding window" algorithm. This treats all trigger regions equally as potential jet seeds, at the centre of a 3×3-region window. A jet is found in a particular region if the transverse energy ($\mathrm{E_T}$) in that region is larger than in the eight surrounding regions. The $\mathrm{E_T}$ values are summed over the nine contributing regions and the total is the $\mathrm{E_T^{jet}}$ assigned to the jet. The algorithm treats consistently the rare case where two neighbouring regions have large and equal $\mathrm{E_T}$, and ensures that one and only one corresponding jet is found.

Jets found by the algorithm are classified into three streams. Jets found in the eight trigger regions between $3 < |\eta| < 5$ are "forward jets"; those in the central 14 regions in η are subdivided into "central jets" and "τ jets". This last classification uses pattern flags generated in the RCT to identify energy deposits confined to a small number of trigger towers in the centre of the 3×3-region window; a signature characteristic of the hadronic decay of τ leptons.

The use of a "sliding window" algorithm for jet identification is important to ensure uniform physics acceptance. It has a strong impact on the choice of processing architecture, requiring the sharing of input data among processing elements. This leads to an emphasis on a high processing density and fast links for data communication, as described in Section 2.2.

Sort processing Sorting is performed on five categories of object: three types of jet and two classes of e/γ candidate. The objects are specified by a "rank" and a (ϕ, η) trigger region; the rank is a non-linear compressed transverse energy scale, with the compression functions stored in lookup tables. For each category of object, the sort algorithm finds the four highest-ranked candidates from a total of 72 input.

Activity triggers The input to triggers based on overall event activity is generated, by summing a number of quantities over the whole calorimeter acceptance. The identification of events with significant missing transverse energy is particularly important, as this might indicate the production of undetected particles. We also wish to be able to trigger on events with very large total transverse energy, independent of any other signatures based on physics objects.

The jet finding logic is organised so as to enable the summation of region energies into strips of 20° in $\Delta\phi$. For the missing $\mathrm{E_T}$ calculation, the strip sums are transformed into a pair of orthogonal components, using the known position of the strip in ϕ; these are then summed over the whole calorimeter to form the components of the missing $\mathrm{E_T}$ vector. From the components, the magnitude of missing $\mathrm{E_T}$ is found along with a direction in ϕ. In parallel, the raw strip energies are also summed to form the total $\mathrm{E_T}$. An alternative measure of overall event activity, known as $\mathrm{H_T}$, is derived by summing $\mathrm{E_T^{jet}}$ for all jets found above a chosen threshold.

Multi-jet triggers To facilitate the design of triggers for specific multi-jet physics signatures, the GCT specification incorporates the requirement to provide counts of total numbers of jets above a range of thresholds. Other conditions such as cuts on particular η ranges can also be applied to the jet counts. Flexibility is required here to allow the cuts to be selected according to the physics requirements.

2.1.3 Output data

The GCT sends to the global trigger, for each bunch crossing, information on the four highest-ranked objects in each of five streams, as discussed above: isolated and non-isolated e/γ objects; forward, central and τ-flagged jets. It also sends the missing $\mathrm{E_T}$ magnitude and direction; the total $\mathrm{E_T}$ and $\mathrm{H_T}$; and a set of jet counts for use in multi-jet triggers. Additionally, the GCT is required to forward the muon activity flags generated in the RCT, for processing by the muon trigger logic. The total output is 910 bits for each LHC bunch crossing, at an aggregate rate of 36 Gbit/s.

2.2 Trigger processing implementation

The recent emergence of commercial VLSI devices offering multi-million-gate logic arrays, and multi-gigabit serial communication links, has made it possible to design trigger systems based on generic processing modules. For the GCT processing, the most demanding requirement comes from the jet finding algorithm. Once a processing architecture had been devised to implement the jet finding, the object sorting and other functions could be accommodated using the same hardware design. This allowed efficient use of limited design resources. The configurability of processing built in to this concept also facilitated some evolution of the requirements definition during the system development, without requiring changes to the underlying hardware.

The processing architecture was selected to allow the complete GCT function to be performed within one crate of modules. High bandwidth communication between modules was essential, both for the jet finding and for the final stages of algorithms requiring information from the complete calorimetry system. This is a common feature of trigger systems. We chose an innovative solution, which avoids the traditional challenges associated with the implementation and testing of one-off, monolithic PCB backplanes to route the high density of signals required. Following the design philosophy of configurable processing and data routing, a flexible configuration of point-to-point data links between modules was achieved by means of an arrangement known as the "cable backplane". High speed serial data are transmitted from one module, through a pair of connectors arranged back-to-back in the backplane, and onto a cable that is connected to a module in another slot. In Section 2.2.1 below, we discuss the processing and data link technologies selected. Subsequently in Section 2.2.2, we outline the architecture designed for the processor module and the GCT processing system.

2.2.1 Major technology choices

The design of the generic processor module was based around the selection of a small number of key technologies for processing and data communication. Two serial link transceiver devices were used: one for the input and

output data transmission; and a higher speed, shorter distance link for the communication between modules. In both cases, a significant factor in the choice of link was the latency for data transfers. The trigger processing, data routing and control of the link devices was carried out using Field Programmable Gate Arrays (FPGA).

Processing elements The processing elements in all the module designs were selected from the Xilinx Virtex-II family of FPGAs [5]. Device sizes of 1–1.5 million gates were chosen for data routing and control functions, with 3 million gate parts for the main triggering function. In most cases the selection of a particular Virtex-II part/package combination was driven by I/O considerations. The number of I/O pins per device was in the range of 200–400, with signalling speeds between FPGAs of 160 MHz, four times the LHC clock frequency. The Virtex-II family offered support for a much wider range of I/O signalling standards than earlier FPGAs of similar size. This allowed the selection of appropriate single-ended, terminated standards for high-speed parallel transfers between FPGAs, as well as interfacing directly to the serial transceiver chipsets.

Intercrate communication The transmission of the input and output data between the crate of processor modules and other parts of the trigger system used the DS92LV16 serialiser/deserialiser (serdes) chipset [6] from National Semiconductor. These allowed a data transmission rate of 1.28 Gbit/s per differential pair, with a physical bit rate of 1.44 Gbit/s. The latency for data transmitted between FPGAs on different boards, for cable lengths in the region of 1 metre, was within three LHC bunch crossings (75 ns). To transmit the serial data, standard Infiniband [7] connectors were chosen. Two variants were used, with the 1X connector allowing two differential pairs or the 4X connector accommodating eight pairs. Cables, including some custom 1X-4X assemblies, were supplied by CS electronics [8] and Leoni Special Cables [9].

Intracrate communication The serdes device selected for the communication between modules was the Vitesse VSC7226 [10]. These links were operated with a data transmission rate of 2.56 Gbit/s per differential pair, and a physical bit rate of 3.20 Gbit/s. Each transceiver contains four transmit and four receive circuits. The latency achieved using these devices, for transfer of data between FPGAs on different modules, was below 1.5 LHC bunch crossings (37.5 ns).

The cable backplane uses the VHDM-HSD connector family [11] from Teradyne, which allows a signal density of two wire pairs per 2 mm of card edge. Custom cable assemblies were obtained from Molex [12]. Each assembly carries two differential signal pairs, and consists of two 5-socket wafers connected by up to 50 cm of cable. The cable construction is twinax, with a foamed PTFE dielectric, and the assemblies are specified for signal propagation speeds up to 5 Gbit/s. The cable backplane components can be seen in Figure 4.

2.2.2 Processing architecture

Trigger Processor Module The generic Trigger Processor Module (TPM) contains a total of seventeen Virtex-II FPGAs. Figure 2 shows the arrangement of the major processing components, and data paths between them. A prototype module, of dimension $9\mathrm{U} \times 400$ mm, is shown in Figure 3. The processing is performed by four large (minimum 3 Mgate) FPGAs. A further twelve smaller FPGAs are used to control the link hardware, and to route the data between the processing FPGAs and the links; one FPGA is dedicated to overall control of the board.

The input data are received via four 4X Infiniband connectors on the front panel, using 24 serial link receivers. The available input bandwidth is 768 bits per LHC bunch crossing, or 30 Gbit/s. Three 1X connectors are provided, also on the front panel, for output on up to six serial links. For data exchange over the cable backplane, six VSC7226 quad serdes devices give an available bandwidth of 60 Gbit/s in each direction. Data arriving at the front panel can also be routed directly to the fast backplane links for low-latency sharing of data between modules.

Processing layout A total of eight TPMs is required to perform the GCT processing function. Two of these take care of the sorting of isolated and non-isolated e/γ candidates. A further six modules process the region energy data, including jet object recognition and the generation of jet and energy trigger data for the Global Trigger. The main trigger processing functions are carried out by the four large FPGAs on the eight TPMs. Three of the four FPGAs on each module receive the input data. These are referred to as "Proc A" devices. The fourth FPGA is the "Proc B" and calculates the final GCT output data. A ninth TPM is used as part of the control system, as described in Section 3.2.

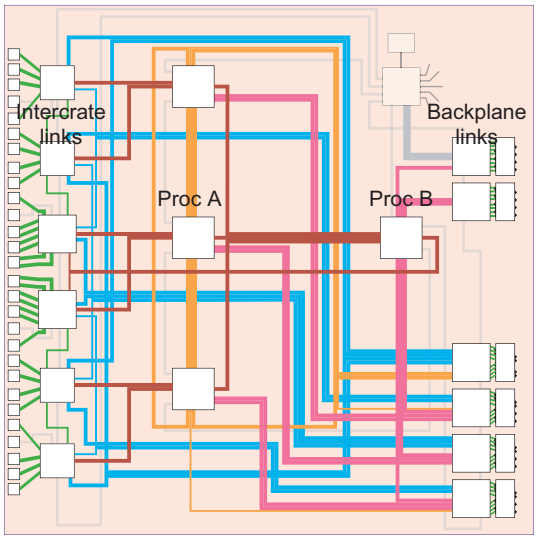


Figure 2: Schematic view of data paths on the Trigger Processor Module. The widths of the coloured lines are proportional to the data bus widths.

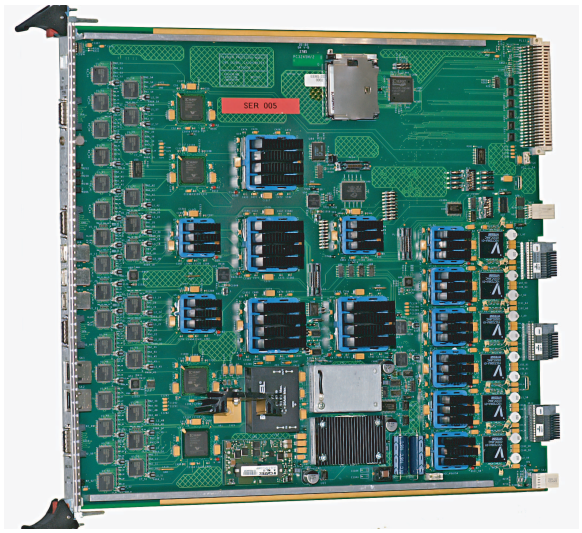


Figure 3: A prototype Trigger Processor Module.

The processing for each of the two categories of e/γ candidate takes place on a single module. There are 72 candidates input from the RCT. Each of the three "Proc A" devices receives 24 candidates, and performs a first stage of sorting to find the four with the highest rank. The "Proc B" finds the final four output objects.

The region energy data from the RCT are distributed over the remaining six TPMs. Three modules cover central η values, and three process the forward and backward regions, for a 120° range of ϕ in all cases. The "Proc A" devices perform the jet object recognition; to achieve this, around 3000 bits of data per 25 ns bunch crossing interval are exchanged over the cable backplane. The six "Proc B" FPGAs produce the sorted jets and activity trigger information, and a further 1275 data bits per 25 ns are exchanged here. A total of 39 links on the cable backplane are required for this processing. The backplane wiring is shown in Figure 5.

3 GCT system design

The previous Section focusses on the trigger processing aspects of the design of the TPM, without reference to important engineering issues such as power and timing distribution, and control, monitoring and test functionality. In order to address these issues, we now turn to the description of the GCT as a complete system operating within the CMS electronics and data acquisition environment.

Two additional module designs were required for the complete GCT implementation. The interfaces to CMS central timing, synchronisation, and data acquisition (DAQ) systems were provided through a Communications Module housed in the TPM crate. A set of Input Modules was required to receive the parallel data arriving on long cables from the RCT, and transmit them to the TPMs over fast serial links. Two additional crates were required to house the Input Modules, so that the full system was designed to occupy three crates in a single rack.

We begin the description of the system design of the GCT with a discussion of the additional requirements and external interfaces. We describe the Communications Module and the use of a ninth TPM to satisfy these requirements. We then move on to the power, timing and control aspects of the design of the TPM. Finally we describe the Input Module subsystem.

3.1 System requirements and external interfaces

In order to carry out its trigger processing function, the GCT must satisfy a range of additional design requirements. It must conform to standard interfaces for timing, fast control and status reporting; it has to provide information

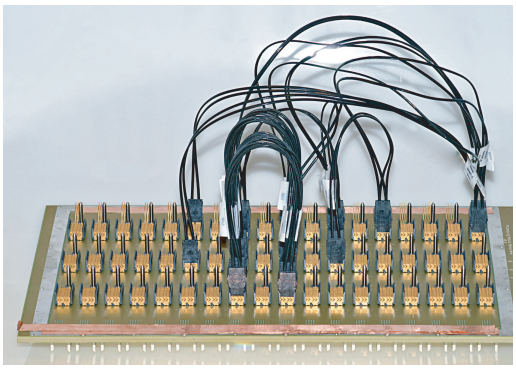


Figure 4: A cable backplane assembled for tests of prototype TPMs.

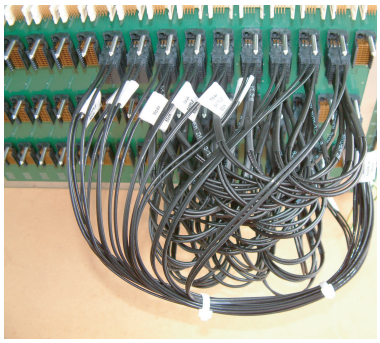


Figure 5: A mockup of the final cable backplane layout, including all connections required for the jet and energy processing.

for the offline record for each triggered event; and it is integrated with other trigger components in a common framework for control and testing purposes. It also provides monitoring information on the rates of activity seen by the calorimeter systems, as a measure of the LHC machine luminosity.

The standard interfaces are implemented in the Communications Module. A dedicated TPM collects and distributes control information to the other eight TPMs discussed above; this ninth module, the "DAQ-TPM", couples closely to the Communications Module. The TPMs and Communications Module implement a standard VMEbus [13] interface for software control and testing.

Most of the control functionality discussed in this section is implemented over common signal paths. A custom Control Bus protocol has been developed to enable reliable communication among the FPGAs on a single TPM, and this is then extended to the DAQ-TPM and Communications Module.

3.1.1 Timing and fast control

The LHC bunch crossing clock is distributed around the CMS front-end electronics, along with synchronous signals such as the Level-1 Accept decision, by the Trigger, Timing and Control (TTC) system [14] developed at CERN. The GCT incorporates a single TTC receiver device, located on the Communications Module. The clock is distributed to all TPMs via dedicated links on the cable backplane. Other synchronous signals are routed by the Control Bus mechanism.

CMS has a system for monitoring the status of all front-end components via the Fast Merging Module (FMM) [15]. Within the GCT, each FPGA continuously generates Ready/Busy status signals and has the capability to raise error flags which are then forwarded through to the FMM.

3.1.2 Trigger data capture

The GCT is required to provide information to the CMS DAQ system for each event passing the Level-1 trigger. This information is required for online performance monitoring and offline analysis of trigger efficiency. The DAQ and higher-level trigger systems may also make use of event summary data from the Level-1 trigger in order to identify regions of interest.

In normal running, all input and output data associated with the GCT are made available to the DAQ system for each triggered event. The system is sufficiently flexible to allow the capture of additional intermediate data, or information from beam crossings either side of the triggered one. Data captured for each triggered event are collected by the DAQ-TPM and sent to the Communications Module. The onward transmission to the CMS DAQ uses the S-Link64 [16] interface.

3.1.3 Control, monitoring and test

The interface to software control systems, such as the CMS Run Control, is provided through a CAEN V2718 PCI-VME bridge [17]. The TPM crate has a standard VME backplane in the top 3U, and all TPMs implement an A24/D16 VMEbus interface. During the development phase, this interface has been used to control testing of prototype hardware and the development of the large suite of firmware and software needed to operate the system. This will be discussed in more detail in Section 4. In the final system configuration, the VME interface allows the GCT to respond to commands from Run Control, and to report the results of continuous monitoring of link integrity and data quality built in to the link controller firmware.

3.1.4 Online luminosity measurement

Accurate knowledge of the LHC luminosity at the CMS interaction point is necessary in order to measure physics cross-sections. Absolute luminosity measurements will be performed using specialised detectors, with the resulting data evaluated offline. However, it is also important to perform continuous online monitoring of luminosity in order to provide rapid feedback to the CMS and LHC operators, and to monitor the luminosity for each LHC bunch-pair individually. Since the GCT receives data from the whole calorimeter system for every bunch crossing, it is possible to provide online luminosity monitoring on a bunch-by-bunch basis. The rates of various signatures can be monitored locally on the trigger processing TPMs. Studies showed that the precision required for an online measurement could be obtained monitoring jet rates in the forward calorimeter system, or at higher luminosities by a "counting zeroes" method as used at the Tevatron [18]. The rate summaries are collected by the DAQ-TPM and transferred to the central Run Control via VME.

3.2 Communications Module and DAQ-TPM

The Communications Module (CM) is required to accommodate two standard components providing external interfaces as discussed above: the TTC receiver ASIC for clock and trigger signals; and the S-Link64 transmitter daughterboard to send data to the DAQ system. Clock signals from the TTC system are fanned out across the custom backplane to all TPMs. The remaining synchronous control signals are sent to a Virtex-II FPGA and onwards to the GCT control system.

The DAQ-TPM uses identical hardware to the modules used for trigger processing, with dedicated firmware to enable it to carry out the required data acquisition and control functions. Its principal purpose is to generate the GCT data record for every Level-1 accepted event. It receives data blocks from the processing TPMs over links on the cable backplane; it buffers and formats the data, and sends them to the CM. The CM then places the data record in the S-Link64 output buffer for onward transmission to the DAQ system.

Other functions carried out by the DAQ-TPM include distribution of synchronous control signals to the processing TPMs; and fanning-in and recording of fast status signals, which are then used to generate the overall GCT status sent to the FMM. All communication between the CM and DAQ-TPM, for system synchronisation, fast control and DAQ, takes place over front panel links. These links are of the same type as those used for data input and output in the eight trigger TPMs.

3.3 TPM design

3.3.1 Timing and synchronisation

The precise, low-jitter, 40 MHz LHC clock is received from the CM via a dedicated link on the cable backplane. Zero-delay PLL-based fanout devices arranged in a tree structure are then used to provide clock signals to all FPGAs, serial link transceivers and other components on the TPM. The final jitter after all stages of fanout is below 80 ps rms, as required by the specification of the transmitter portions of the front panel links. Parts of the processing system are operated at multiples of the base 40 MHz frequency using the clock multiplication capabilities of the PLL fanouts, and the Digital Clock Manager facility inside the Virtex-II FPGAs.

The pipelined Level-1 trigger relies throughout on timing to identify data with a particular LHC bunch crossing; and so to associate correctly data from the same bunch crossing in different parts of the processing system. On the TPM, data received by every FPGA carries a synchronisation marker so that these timing relations can be monitored continuously. The synchronisation markers are derived from the LHC "Orbit" signal, generated once every 3564 bunch crossings (approximately $88\,\mu$s) and distributed via the TTC system. Those FPGAs that receive data from serial links are also required to monitor the status of the links, ensuring that they remain continuously

locked with a constant latency. Any errors detected are reported to the central status monitoring through the FMM interface.

Issues of data synchronisation are particularly important in the design of the Input Modules, discussed in Section 3.4 below.

3.3.2 Power distribution

Each TPM had a power consumption of between 100W and 150W, depending on the processing configuration loaded. Four different supply planes at voltages between 1.25V and 3.3V were required to power the FPGAs, with other components running off 2.5V or 3.3V. The power supply connectors can occupy only a small amount of the backplane, due to the space required for inter-module data communication. These requirements led to a decision to adopt a standard approach from the telecommunications industry, in distributing power at 48V to the modules and using on-board DC-DC converters to supply the high current, low voltage planes.

3.3.3 Board-level control

For control and Data Acquisition functions within a single TPM, a custom "Control Bus" has been implemented. For these purposes, the 16 FPGAs used for algorithm implementation and data routing are regarded as "slaves". The 17th FPGA acts as the single "master" on the Control Bus, and is known as the Control FPGA. The Control Bus is arranged as a unidirectional daisy chain, beginning and ending at the Control FPGA. It consists of 13 parallel bits, assigned as five control and eight data bits, and clocked at 160 MHz. To restrict the latency for operations on the Control Bus, the 16 "slave" FPGAs are divided into two groups of eight, operated independently.

The Control FPGA implements the A24/D16 VMEbus slave interface, and manages the communication with the DAQ-TPM. Control communications from the two sources share the Control Bus, with DAQ operations having arbitrated priority.

3.4 Input modules

3.4.1 Overview of the Input Module system

Constraints from the physical layout of the Level-1 trigger components dictate that the signals from the Regional Calorimeter Trigger are transmitted over cable lengths of $15\,\mathrm{m}$ to arrive at the GCT. To transform these signals to a high-density, compact format suitable for input to the TPMs, they are first received in the GCT by a set of input modules (IM). The IM, shown in Figures 6 and 7, is a double-width module assembled from two $6\mathrm{U} \times 220\,\mathrm{mm}$ PCBs. Each IM is capable of receiving the data from one RCT crate; thus eighteen modules are required in total. The physical format of the IM is determined by the cables used for transmission of the RCT output data. These are 34-twisted-pair cable assemblies with SCSI-2 [19] standard connectors. Each RCT crate sends its data on six cables, and three of the connectors fit onto a 6U high PCB.

The IMs are housed in custom crates with a capacity of nine modules per crate. The 54 input cables enter from the rear of the crate, where they are supported and connected to a PCB "mid-plane". The nine modules plug in to the front side of the "mid-plane". This arrangement provides good mechanical support and allows for modules to be exchanged as needed without requiring cables to be disconnected. Since the card edge space at the rear is completely taken up with the input connectors, the IM is not provided with any standard control interfaces via VME. Instead, control functionality is implemented via a link to the associated TPM.

3.4.2 Input data reception

The IM comprises a motherboard with a simple daughterboard. The motherboard contains most of the functionality. The daughterboard hosts three of the six input connectors, and associated line receivers, and transmits half of the input data onto the motherboard. The data from the RCT are transmitted at $80\,\mathrm{MHz}$ in differential ECL, over a distance of 15 metres. On the IM , they are received by line receiver chips [20] chosen for their wide common mode input voltage range. The practical advantage of these devices in our application is the capability to receive standard below-ground offset ECL voltage levels without providing a negative supply voltage.

3.4.3 Data synchronisation and output

The IM must synchronise the incoming data to the local GCT clock and compensate for any timing skew between different data bits. The control link from the associated TPM, discussed below, also provides the timing reference

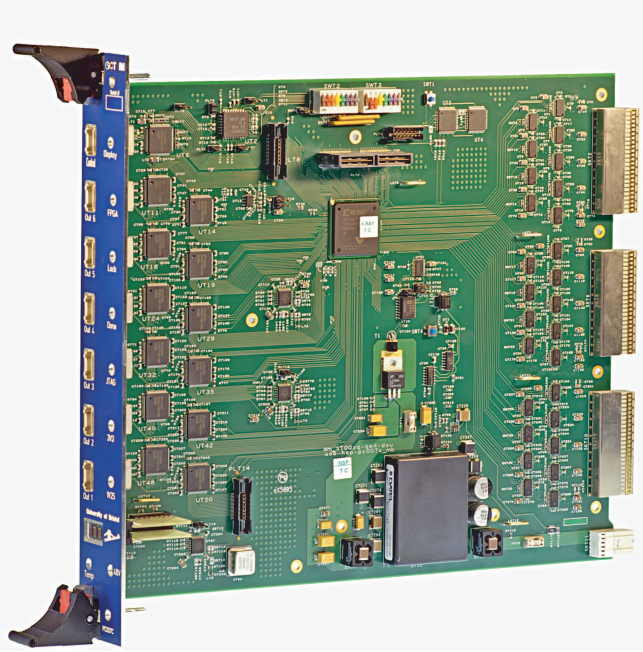


Figure 6: The Input Module motherboard.

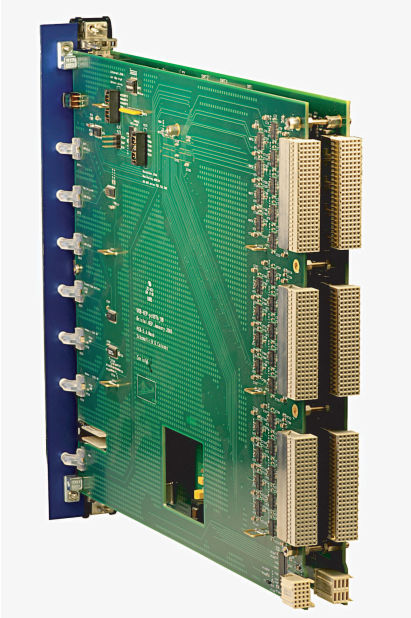


Figure 7: The fully assembled Input Module.

for the IM. The input data synchronisation is performed automatically, and monitored, using a single Virtex-II FPGA on the motherboard. This FPGA also incorporates the routing of data to twelve output serial links. The output data are sent through six Infiniband 1X connectors on the front panel. The routing of data to the output links takes account of the organisation of the processing algorithms on the TPM, as well as the fact that about 10% of the input data received are destined for onward transmission to the muon trigger, rather than the TPM system.

3.4.4 Input module data acquisition and control

The Input Modules have no direct communication path to the CMS DAQ, Run Control, or test and diagnostic software. A JTAG [21] interface is provided for simple testing of these modules, but the main control path in full system operation is via the attached TPM.

It is important to provide an interface to the DAQ from the Input Modules, in order to capture the trigger data as they arrive in the GCT. This allows the monitoring of the data integrity within the GCT itself. To provide this functionality, each Input Module has a control connection to a TPM via a duplex pair of serial links, through a seventh Infiniband 1X connector on the front panel. The FPGA responsible for these links at the TPM end is required to extend the Control Bus to include the FPGA on the Input Module.

4 GCT firmware and software

The complete GCT as described above contains over 170 FPGAs, of which only a small fraction have direct external connections. The FPGA firmware must support a range of system functionality: the capture of data at selected points in the trigger chain; the testing and monitoring of the local trigger path, including the reliability of data links; and the distribution of control messages. We have developed a system of firmware and software to distribute the required functionality throughout the GCT. New developments have also been required in the management of large and complex firmware configurations. In this section we describe the key novel features of these aspects of the system.

4.1 FPGA firmware design

All FPGAs involved in the trigger processing path are loaded with a common firmware structure, as shown in Figure 8. This structure incorporates the Control Bus slave node and a system of data buffers, along with the data routing and algorithm processing required for the trigger. When configured for normal running, the buffers are used to capture data at the input and output of the FPGA, delay it for the fixed Level-1 processing latency and make it available for readout on receipt of a Level-1 Accept trigger signal. A number of test configurations are also available to allow a variety of test patterns to be generated. Available tests, selected via commands on the Control Bus, include the generation of fixed patterns or random data, or the issuing of data stored in the buffers. This allows a wide range of tests to be carried out during system commissioning to verify the operation of all data links and algorithms in the system.

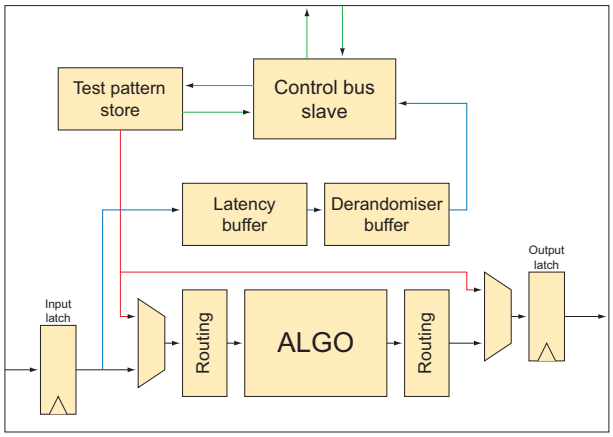


Figure 8: Structure of the firmware loaded into all trigger processing FPGAs. The trigger data path is shown at the bottom of the figure. Other blocks are required for data acquisition, control and test functions.

4.2 Firmware development environment

In order to simplify the firmware implementation and testing, many VHDL [22] modules for commonly used functionality are parameterised and reused throughout the system. Configuration control becomes an important issue with a system of this scale, and commercial firmware development tools currently do not support this use case well. In order to address this problem, a firmware configuration and build system has been implemented, based on techniques that are well known in software development. VHDL code is contained within a source repository [23] to allow controlled simultaneous work by a number of developers. Given a code version release and a database of configuration information, the firmware build tool is able to resolve dependencies between modules, and target either synthesis or simulation of a TPM or the whole GCT system by passing appropriate scripts to the backend commercial tools. The tool allows for code parameterisation and the insertion of standard test blocks, via a simple macro expansion facility similar to that implemented in compiler preprocessors. The firmware build and debugging process can be accelerated using a cluster of CPUs simultaneously.

4.3 GCT software

A suite of software was developed to test, control, operate and monitor the GCT hardware. This comprises the "GCT Driver", a library that implements the various functions; the "GCT Controller", which provides a remote interface to the Driver; and the "GCT GUI", which provides a graphical interface. Test operations, required during the development and integration period, are implemented through an easy-to-use scripting facility. Finally, a bit level emulation of the GCT algorithms was developed, that could be used to verify the hardware processing.

4.3.1 GCT Driver

The Driver is a model of the hardware, written in object-oriented C++. The relationships between hardware and firmware components are replicated in the object model, as illustrated in Figure 9, thus facilitating the implementation of additional functionality when required. We use the CMS Hardware Access Library (HAL) [24] to model and access the VMEbus and PCI-VME bridge used to control the GCT. A novel feature of our approach is to use

a software interface generation tool (SWIG [25]) to produce a Python [26] interface for the Driver. Python is an interpreted object-oriented language, providing an interactive environment in which the user may manipulate the object model of the GCT, and hence the hardware. This provides both a highly flexible system for debugging firmware, and a test-bed for the higher levels of functionality in the Driver itself. In real system development, this functionality is built up and debugged in parallel with firmware functionality, resulting in a well integrated hardware/software system. Python scripts may also be saved for future use as test programs.

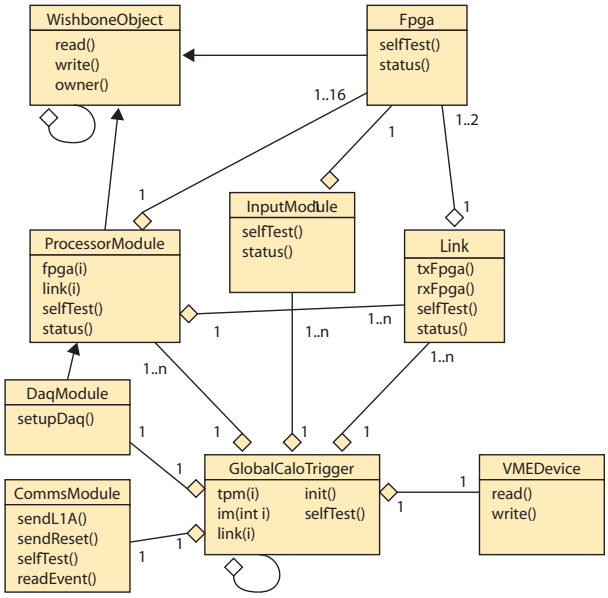


Figure 9: The object model of the hardware, as implemented in the GCT Driver.

4.3.2 GCT Controller and GUI

To integrate the GCT with the global CMS Run Control, the GCT Controller provides an interface to the Driver allowing the hardware to be controlled remotely from either the Run Control or the GCT GUI. The GCT Controller is a C++ application built using the XDAQ [27] platform, a software framework designed specifically for the development of distributed data acquisition systems. The GCT GUI is a Java application that allows the user to perform some common operations via a graphical interface, as well as providing the ability to send Python commands and scripts to the Controller (and hence Driver) if more flexible remote operation is required. It also displays the large amount of real-time monitoring information acquired from the hardware, including: status of all serial links; per-FPGA status of control-bus nodes and DAQ buffers; pipeline synchronisation status; and temperature and power supply data from each module. These monitoring data are captured at regular intervals by a separate monitor-collector thread implemented by the GCT Controller, which operates independently of the main control thread.

4.3.3 GCT emulator

The GCT emulator is based on a related object model of the GCT hardware and firmware. Rather than include methods to access the control system, this model has methods that model the trigger processing and data paths. Each FPGA object in the model stores pointers to the sources of its input data; when requested for its output, the FPGA first requests the output of these upstream components, processes the data received, then returns the output. This allows the complex data processing to be modelled in relatively simple pieces. The emulator has been used in Monte Carlo studies of trigger performance, and is intended to be used to verify the data processing for triggered events from the hardware readout record.

5 Prototyping and development

The GCT design as presented here represents a significant increase in functionality and complexity over that documented in [3] in 2000. The design requirements were extended to incorporate the pattern recognition for

jet identification, leading to an increased bandwidth specification for communication between TPMs on the cable backplane. This enhanced functionality was proposed and approved in 2002, following a successful series of tests on the data link technologies. The detailed design of the hardware modules, and of the firmware and software systems for test and control, proceeded in parallel.

5.1 Testing and system integration status

Tests using the prototype hardware have focussed initially on the reliability of data communication paths throughout the GCT system. These tests have included data paths both for control and for execution of trigger algorithms. The trigger data paths transfer data between pairs of embedded FPGAs. The modular firmware design approach allowed all paths to be tested using the same firmware and software; an extensive suite of pattern tests was developed for this purpose.

In order for tests to be carried out, it is imperative for the daisy chain Control Bus to operate with high reliability to allow communication with any FPGA in the system from the Driver software via the VMEBus. This has been achieved in systems with various hardware configurations incorporating several TPM and Input Module prototypes. A number of different "soak tests" were successfully carried out, running all on-board data paths between FPGAs and 96 links on the cable backplane reliably for several days. The aggregate number of error-free bit transfers in a continuous test was in excess of 10^{16}.

Algorithm firmware for all required trigger functionality has been developed. This includes the "sliding window" jet finding; sorting of trigger objects; energy summing and calculation of missing $\mathrm{E_T}$. The algorithms have been extensively tested and optimised using simulation tools, and shown to meet the requirements on latency and gate count. The sort algorithm was fully tested in Virtex-II hardware, giving results in agreement with simulations.

Integration activities have included the observation of data transfers between the GCT and other trigger subsystems, the Regional Calorimeter Trigger and the Global Trigger. The firmware for the Communications Module and DAQ-TPM has largely been completed and this aspect of the control system partially integrated. The Driver software approach has proved to be extremely useful and flexible during this testing phase.

5.2 Issues with the design

The prototype testing programme described above has covered all major components of the system design and has generally been successful in verifying their operation. No single major problem has been identified; however progress was slower than planned. In this section we describe two issues with the design as presented that have contributed to the slow development.

5.2.1 Data link reliability

The testing programme demonstrated successful operation of all data links in the system. In most cases this was achieved with few problems, both in demonstrator projects and with full scale module prototypes. In particular the cable backplane links, with the most demanding performance requirements, were shown to work extremely well in practice. The intercrate links, although successfully operated in a clean laboratory environment, proved quite susceptible to environmental and common mode noise. A source of common mode noise in our system was the use of high-power DC-to-DC converters, which initially generated levels of the order $1\,\mathrm{V}$ between ground planes of adjacent modules. A further hardware iteration was proposed to improve the performance of these links.

5.2.2 Firmware complexity

The TPM featured an unusually large number of FPGAs, to carry out the required functions of trigger processing and control of external data links. This led us to the design of a relatively complex firmware system, based on the multiplexing of all DAQ, control and test functionality over a single, narrow Control Bus. While offering powerful and general test functionality in principle, this approach had a number of disadvantages. Its implementation and debugging required considerable investment of effort. All other aspects of system development and integration relied on its trouble-free operation, and so were subject to delay in the case of any Control Bus problems. Perhaps most significantly, the assembly of multiple firmware configurations from modular components was carried out and tested in a custom environment using techniques from software engineering. This departure from the use of conventional commercial tools proved a significant barrier to attempts to speed up the firmware development by introducing additional effort into the project.

5.2.3 Summary

By early 2006, working prototypes of the hardware, firmware and software components of the system had been produced and a range of system integration activities was ongoing. Preparations for another hardware iteration, to address the issues identified, were well advanced. At this point concerns about the slow overall progress, and about the complexity and long-term reliability of the full-scale system, led to a decision by CMS to pursue an alternative design for the final GCT.

In the following paragraphs, we discuss prospects for future developments along similar lines. We highlight a number of issues that should be taken into account by system designers, and areas where recent commercial developments might facilitate the design of similar systems.

5.3 Lessons for future projects

For future trigger developments with similarly demanding requirements, it seems likely that the use of FPGAs for processing, and fast serial links for data communication, will become ever more widespread. The issues identified in our Introduction will need to be taken into account by designers of such future systems. Continuing advances in FPGA development mean that trigger processors can be extremely compact and powerful. This puts an emphasis on achieving a high density of information input to the processor, and efficient data sharing among processing elements. The choice of optical transmission of data between subsystems, even over relatively short distances of order $10\,\mathrm{m}$, is becoming increasingly favourable.

Since the design choices were made for this project, later developments in FPGA technology have led to devices with large numbers of integrated fast serial links. Such devices, if available at the time, might have been used to achieve a considerable reduction in the number of FPGAs needed for the TPM design, relaxing the requirements both for power distribution and firmware complexity. Alternatively, future designers might choose to optimise using similar numbers of FPGAs per module, with reductions in numbers of modules or further increased processing power.

Boards and crates containing multiple FPGAs will continue to have demanding power requirements. The need to deliver power at multiple DC voltages will remain, and system-level engineering solutions must be found. The approach of distributing power at a single, high DC voltage is used by modern sub-rack architectures [28]. The advantages are that backplane design is substantially eased, and excellent regulation of the power rails on each board can result. In our system, the issue of common mode noise was encountered; this is tractable, but requires careful design. An interesting development from industry, not available when the design choices for our project were made, is the recent introduction of point-of-load DC-to-DC converters delivering multiple voltages to a single device.

Future system designers will also need to address the issue of firmware engineering — managing the development and maintenance for systems containing large numbers of FPGAs. Concepts familiar from software engineering, such as modularity and code re-use, need to be integrated more closely with standard firmware development tools, and packaged in such a way that they can be used by hardware/firmware engineers. A related issue is the need to develop "system" firmware to exchange information among embedded FPGAs for the purposes of synchronisation, control and testing. Our experience suggests that this is likely to be substantially more complex than the trigger algorithm firmware, and its development needs to be considered carefully at an early stage of planning.

Our project was based around the choice of a single design of processing module to perform all trigger functions. Points made in the preceding discussion are applicable to any system where the requirements dictate the use of $\mathcal{O}(100)$ large FPGAs, communicating together to perform a variety of trigger functions. As remarked in Section 2.2, the motivation for choosing a generic module design was to allow some flexibility in the definition of the processing requirements, and to reduce the requirement for hardware design effort. It also simplifies the approach to "system" firmware. At all stages of development, testing and commissioning, the majority of resources can be focussed on a single object, the TPM in our case. On the other hand, the single object is necessarily quite complex. If the required functionality can be achieved with a relatively small number of simpler modules, the gains may not be apparent. The usefulness of the approach increases if it can be applied to larger systems, or to several trigger components operating together.

6 Summary and conclusions

We have developed the requirements specification for the Global Calorimeter Trigger, a part of the CMS hardware Level-1 trigger processing electronics. In this paper we have described the hardware, firmware and software aspects of a system designed to fulfil these requirements. We have reported on the status of prototype development and testing at the time that development was discontinued in favour of an alternative design, and presented a brief analysis of some factors leading to this decision.

The key aspect of the design emphasised here is the use of a generic hardware Trigger Processor Module, with configurable processing and data routing, for all processing functions. The TPM as described is rather complex; the integration of processing and serial data links available in later generations of FPGAs would enable simpler and more powerful versions to be implemented for future experiments.

We have pioneered a system-level approach to the construction of large-scale firmware implementations using modular components. Future applications following this line of development would need to pay attention to the ease of use of the environment by design engineers used to more traditional hardware and firmware systems. We have successfully introduced a novel approach to the development of test software, of rather general applicability.

Acknowledgements

The authors would like to thank the following for useful discussions on the project: Steve Quinton, Rob Halsall and John Maddox of Engineering and Instrumentation Division at Rutherford Appleton Lab; Magnus Hansen and Greg Iles of CERN PH/CME; the RCT group at the University of Wisconsin, Madison, particularly Sridhara Dasu and Pamela Klabbers; the Global Trigger group at the OEAW, Wien, particularly Claudia-Elisabeth Wulz and Toni Taurok; the CMS Level-1 trigger Project Manager, Wesley Smith; and CMS-UK management, Bob Brown and Geoff Hall. The TPM concept was originally proposed by Uli Schäfer. This work was funded by PPARC (now STFC).

References

[1] CMS — The Compact Muon Solenoid, CERN/LHCC/94-38, December 1994.

[2] The LHC Conceptual Design Report — The Yellow Book, CERN/AC/95-05 (LHC), 1995.

[3] CMS — The TriDAS Project Technical Design Report, Volume I: the Trigger Systems, CERN/LHCC/2000-38, December 2000.

[4] CMS — The TriDAS Project Technical Design Report, Volume II: Data Acquisition & High-Level Trigger, CERN/LHCC/2002-26, December 2002.

[5] Xilinx Inc, San Jose, CA, USA. Information on the Virtex-II family of FPGAs is available at: http://www.xilinx.com/products/silicon_solutions/fpgas/virtex/virtex_ii_platform_fpgas/index.htm

[6] National Semiconductor Corporation, Santa Clara, CA, USA. Information on the DS92LV16 serdes is available at: http://www.national.com/pf/DS/DS92LV16.html

[7] Information about the Infiniband standard is available from the website of the Infiniband Trade Association: http://www.infinibandta.org/home/.

[8] CS Electronics, Irvine, CA, USA. Website address http://www.cs-electronics.com/

[9] Leoni Special Cables, Friesoythe, Germany. Website address http://www.leoni-special-cables.com/eng/index_eng.php.

[10] Vitesse Semiconductor Corporation, Camarillo, CA, USA. Information on the VSC7226 serdes is available at: http://www.vitesse.com/products/product.php?number=VSC7226

[11] Now supplied by Amphenol-TCS, Nashua, NH, USA. Information available at: http://www.amphenol-tcs.com/products/connectors/backplane/vhdm_hsd/index.html

[12] Molex, Lisle, IL, USA. Website address http://www.molex.com

[13] IEEE Std 1014-1987, IEEE Standard for a Versatile Backplane Bus: VMEbus.

[14] The TTC project website. Available at: http://ttc.web.cern.ch/TTC/intro.html.

[15] The Fast Merging Module (FMM) for readout status processing in CMS DAQ, E. Cano et al, Proceedings of the ninth workshop on LHC electronics, Amsterdam, 2003, CERN/LHCC/2003-055, November 2003; The final prototype of the Fast Merging Module (FMM) for readout status processing in CMS DAQ, R. Arcidiacono et al, Proceedings of the tenth workshop on LHC electronics, Boston, CERN/LHCC/2004-030, November 2004.

[16] The S-Link project website. Available at: http://hsi.web.cern.ch/HSI/s-link/. The S-Link64 specification is at http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/docs/slink64.pdf.

[17] CAEN V2718 PCI-VME optical link bridge. Information available at: http://www.caen.it/nuclear/product.php?mod=V2718.

[18] D. Cronin-Hennessy, A. Beretvas and P.F. Derwent, Nucl. Instr. and Meth. A 443 (2000) 37.

[19] ANSI INCITS 131-1994 (R1999), Information Systems – Small Computer Systems Interface-2 (SCSI-2) (formerly ANSI X3.131-1994 (R1999)).

[20] Texas Instruments SN65LVDS352, quad LVDS receiver with -4 V to 5 V input range. Information available at http://focus.ti.com/docs/prod/folders/print/sn65lvds352.html.

[21] IEEE Std 1149.1-2001, IEEE Standard Test Access Port and Boundary Scan Architecture.

[22] IEEE Std 1076C-2007, IEEE Standard VHDL Language Reference Manual.

[23] The Subversion open-source version control system was used. The project website is at http://subversion.tigris.org/.

[24] The CMS HAL project website. Available at: http://cmsdoc.cern.ch/ cschwick/software/documentation/HAL/index.html.

[25] The Simplified Wrapper and Interface Generator, http://www.swig.org/.

[26] The Python programming language offical website is at http://www.python.org/.

[27] The XDAQ website is at http://xdaqwiki.cern.ch.

[28] See, for example, the PICMG 3.0 Advanced Telecommunications Computing Architecture (ATCA) specification, available from PCI Industrial Computers Manufacturers Group (PICMG); see http://www.picmg.org/index.htm.