Design and Implementation of the Global Calorimeter Trigger for CMS

J. J. Brooke, D. G. Cussans, R. Frazier, G. P. Heath, D. M. Newbold

Department of Physics, University of Bristol, BS8 1TL, UK
dave.newbold@cern.ch

S. Galagedera, A. Shah, S. Madani

CCLRC Rutherford Appleton Laboratory, Chilton, UK

Abstract

The functions of the CMS Global Calorimeter Trigger are to identify jet candidates, sort and count these and other physics objects, calculate missing and total transverse energy, and to provide on online estimate of LHC luminosity. The system uses a single design of configurable processor module, based around high-density FPGAs. Modules are interconnected by a high-speed serial backplane capable of moving over 1Tb/s. We present the design of the hardware, firmware and software components of the system, and review the lessons learnt during the design and prototyping phases of system development.

I. THE GLOBAL CALORIMETER TRIGGER

A. CMS Level-1 Trigger System

The purpose of the Level-1 trigger system [1] is to reduce the data rate from the CMS detector to the point where a software-based High Level Trigger (HLT) can perform a preliminary analysis of each accepted event [2]. The Level-1 trigger must accept all potentially interesting physics processes with high and well-understood efficiency, whilst providing a very large rejection factor against QCD background.

The data rates from CMS require that the Level-1 trigger be implemented solely in hardware. The system is designed to have negligible dead-time, and so is implemented in fully pipelined logic. The Level-1 latency must be as short as possible so as to minimize the required buffer memory on the detector. The system is designed to have a maximum latency of 128 LHC bunch-crossings (3.2μs).

The system architecture is shown in Figure 1. Only calorimeter and muon detector data are used at Level-1, since it is possible to find stand-alone candidate physics objects (e/γ, μ, τ high-E_t jets) in these systems, using local pattern recognition only. In each system, trigger primitive generators synchronize and coarsify the data. Local trigger processors identify trigger objects, and global trigger processors then further correlate and sort the candidates. A final decision is made in a single global trigger, based upon object E_t and event topology.

B. Global Calorimeter Trigger

The Global Calorimeter Trigger (GCT) performs those trigger algorithms that require data from the entire CMS calorimeter system. Its main functions are as follows:

- Sorting of the highest E_t e/γ e/γ object candidates found by the Regional Calorimeter Trigger (RCT).
- Identification of jets using a sliding window algorithm, and classification into forward, central and τ jets based upon location and local pattern recognition in the RCT.
- Jet counting for recognition of complex multi-jet events.
- Calculation of global energy flow quantities such as total vector and scalar E_t, and H_t.
- Real-time monitoring of the LHC luminosity on a crossing-by-crossing basis.

In addition, the GCT provides the CMS DAQ system with an event record representing the calorimeter trigger results for each triggered event. The GCT accepts input from the RCT and CMS TTC system [3], and provides input to the Global Trigger (GT), Global Muon Trigger and CMS DAQ.

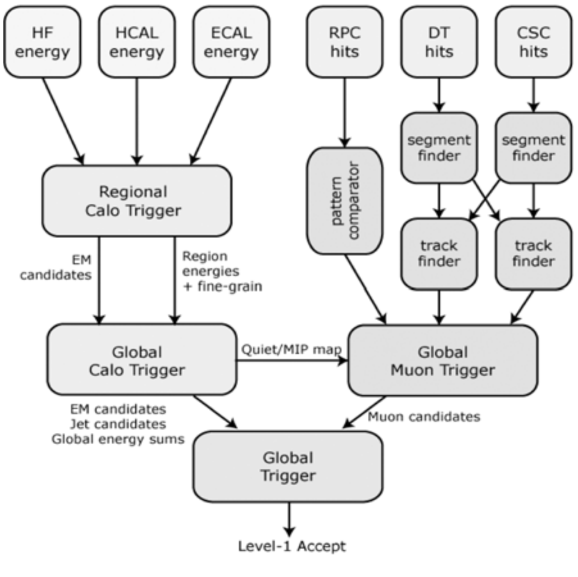

Figure 1: CMS Level-1 Trigger Architecture

C. Design Philosophy

The GCT is a small system on the scale of LHC electronics, but must perform a large variety of tasks. The main challenge for system design is the large quantity of data to be brought to a single central point in the system with minimal latency. Since the signal and background cross-sections at the LHC are not yet fully understood, the system must offer the maximum possible flexibility in its algorithms. The GCT represents a potential single point of failure for CMS, and so must be robust, reliable and maintainable for the lifetime of the experiment (up to 15 years). Finally, the system must be delivered within a relatively modest budget.

These design challenges have been met through use of state-of-the-art, but commercial, technologies. Some of the key principles followed in the design and implementation of the GCT system are as follows:

1) Reconfigurable logic

The recent availability of very high equivalent gate count FPGA devices [4] has been a key factor in the design of the GCT. All data processing in the system uses FPGAs, allowing a great deal of flexibility. FPGA performance still does not approach that of ASICs, but use of highly parallel, heavily pipelined algorithm implementations allows the required throughput to be attained at a reasonable clock speed.

2) High-speed serial communications

Multi-gigabit serial chipsets [5] are now commodity items, and, with a copper physical layer, are low enough in price to replace not only inter-crate parallel links but also the system backplane. The use of high-speed short haul links makes it possible to house the GCT within a single crate and to avoid the design of a custom backplane PCB.

3) Generic processing modules

The use of FPGAs and serial links enables another design feature of the GCT. The system is built from a single design of generic processor module, containing a heavily-interconnected set of FPGAs. The modules may be individually configured for any GCT processing task, or for additional functions not yet foreseen. Use of a single module type reduces design time, risk, complexity and spares cost.

4) Comprehensive self-test and monitoring

Use of highly integrated components, dense multi-layer PCBs, and gigabit signalling makes debugging with external test equipment impractical in many situations. The complexity of data flow in the GCT system makes it essential that the system be able to capture its own state at any time, and perform self-test at a component, board and system level. Essentially, the logic analyser is a part of the system.

5) Maximum functionality in software

The GCT requires a complex setup, control and monitoring procedure. A traditional strategy would suggest the implementation of a complex state-machine control system in firmware. The GCT design aims to minimize the total state in hardware, and implement as much complex functionality as possible in system software. The software and firmware design processes are closely coupled, but the availability of an accurate VHDL system model allows firmware and software co-design to proceed independently of hardware implementation. A disadvantage is that the system cannot operate without real-time software control. In future, it may be possible to shift much of the software functionality to a soft CPU within the system, regaining the traditional stand-alone functionality.

II. System Design and Implementation

A. Trigger Processor Module

The core of the GCT consists of a set of generic Trigger Processing Modules (TPMs). All trigger logic is implemented within these modules. Nine TPMs, housed within a single crate, are required for the baseline GCT system. The TPM is implemented as a single-width 9U x 400mm VME module (see Figure 2). The prototype version of the module is an 18-layer PCB with around 4000 components and 20000 traces.

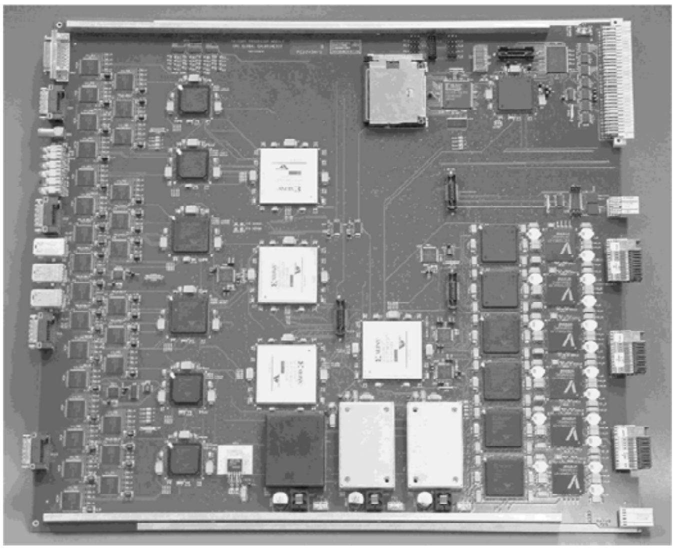

Figure 2: Prototype TPM Board

Algorithm logic is contained within four Xilinx FPGAs (XC2V3000) [4]. These are arranged in a 3-to-1 tree configuration, reflecting the data reduction function of the GCT (see Figure 3). Additional smaller FPGAs are used for synchronization and multiplexing of I/O data, and for control. All FPGAs are interconnected by wide data busses running at 160Mhz, using SSTL-2 with internal termination. Input data comes primarily from a set of National Semiconductor DS92LV16 serdes [5] behind the front panel. Similar devices are used for output links to the GT. Inter-board sharing of data is facilitated by six Vitesse VSC7226 transceiver devices, which provide four lanes of 3.2Gbit/s each [5].

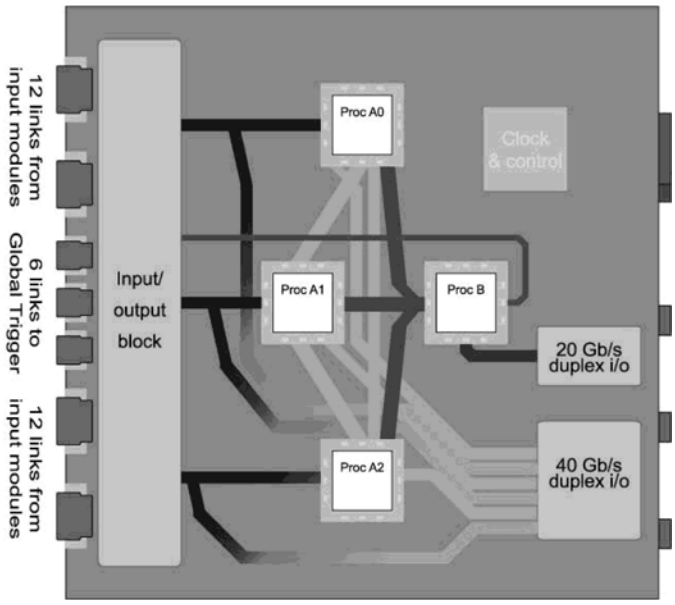

Figure 3: TPM Topology

The entire GCT system is run synchronously with the LHC bunch crossing clock, with a nominal 40MHz differential clock signal distributed to each FPGA via a zero-delay PLL buffer clock tree. The clock is phase-aligned and frequency-multiplied using DLLs within the FPGAs. The serial links are also run synchronously, and are provided with a very low jitter 80MHz clock.

Control and monitoring functions are provided via the crate VME bus. A multidrop system JTAG bus is used for both board-level test and configuration of the FPGAs at power-up. A Xilinx SystemAce-CF device [4] is used as configuration controller on each TPM. Power for the entire GCT system is provided by a single 3kW 48V communications supply from Valere [6]. The board voltages required for logic and I/O on the TPM are provided by several DC-DC converters with heavy filtering. Signal and shield grounds are kept separate throughout the system.

A TPM prototype was implemented in 2003, and has been extensive tested. Problems with PCB fabrication were initially experienced, but have been addressed. A major issue experienced during commissioning was with the JTAG bus itself. Efforts were made to avoid all fast-edged multidrop lines in the design, since these often present signal integrity problems. However, it is necessary to use such lines to implement the JTAG bus, and unreliability of this subsystem was observed. The problem is addressed in the final version of the TPM through use of additional buffering.

B. *Module Interconnect*

The design of a very high density parallel high-speed backplane has proved problematic in past systems. The use of high-speed serial signalling between TPMs requires a modest number of conductors, and so permits an alternative solution. Signals are carried between TPMs via twinax cables with foamed PTFE dielectric, terminated with 2mm HM five-row controlled impedance connectors [7]. Mechanical location and power supply to the TPMs is achieved through a PCB ‘midplane’ containing only ground / power planes and a simple system JTAG bus. The use of PTFE within the underground components of the CMS systems is strictly controlled, and the interconnect is required to be contained within a fireproof steel enclosure.

The use of a ‘cable backplane’ further enhances the flexibility of the system, since it may easily be modified or expanded at a later date. An example eye diagram from signal integrity testing is shown in Figure 4, and the bit error rate over a single serial link has been measured at $<10^{-13}$ without FEC, well within the system requirements.

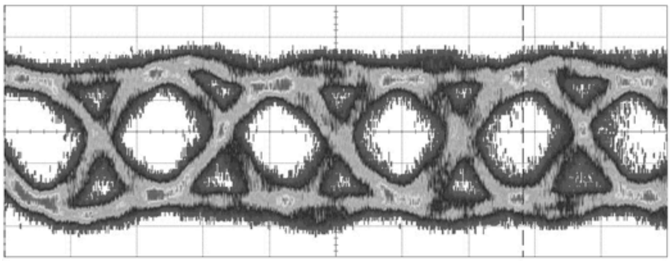

Figure 4: Backplane serial link eye diagram (200ps / div)

C. *Input Module*

The RCT is a large distributed system comprising 18 crates at up to 20m distance from the GCT. An early decision was taken to transmit the large amount of data between the two systems via parallel 80MHz twisted-pair copper links, since this technology has proved reliable in the past. However, it is impractical to terminate over 3500 copper conductors at the GCT crate. Instead, an auxiliary Input Module (IM) has been designed to receive six 34-pair cables, and send the data via copper serial links to one or more TPMs. To allow the use of relatively inexpensive twisted-pair cable without controlled skew, the IM can resynchronize each input bit independently through a 320MHz oversampling technique. The data phases may be measured automatically through transmission of known test patterns by the RCT.

The IM is implemented as a double-width 6U x 220mm module, with no external control bus. Eighteen IMs are required in the GCT, in crates above and below the processor crate (Figure 5). The module is relatively simple in design, consisting of wide common-mode range input buffers [8], a single Virtex-II FPGA for synchronization and test, and output serializers. Control and clock signals are obtained from a single serial uplink provided to each IM by a host TPM.

A three-channel prototype of the IM was successfully tested in conjunction with RCT and GT prototypes in 2003. A pre-production version of the module is now under test, and is shown in Figure 6.

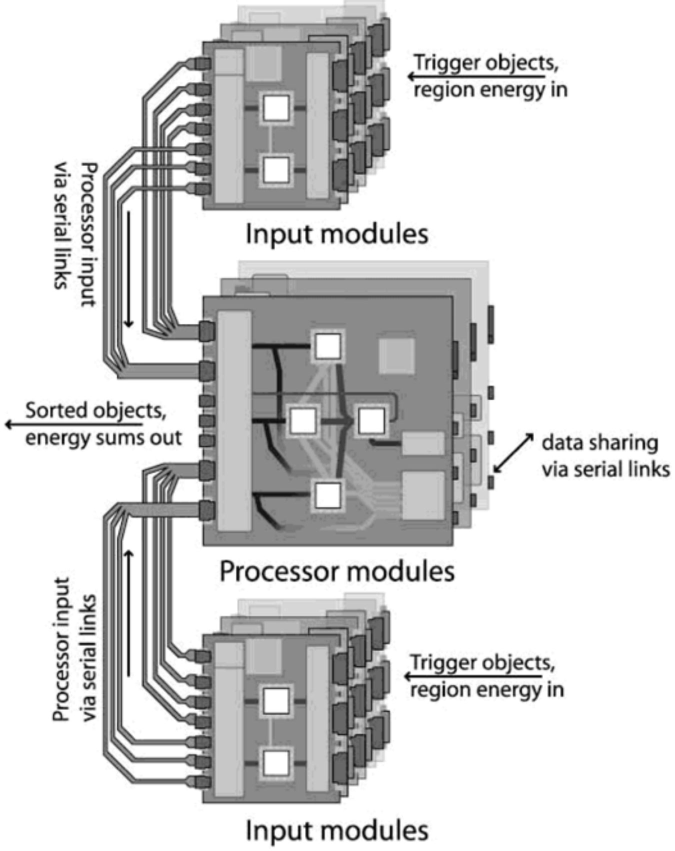
Trigger objects, region energy in
Processor input via serial links
Input modules
Sorted objects, energy sums out
data sharing via serial links
Processor modules
Processor input via serial links
Trigger objects, region energy in
Input modules

Figure 5: GCT module layout

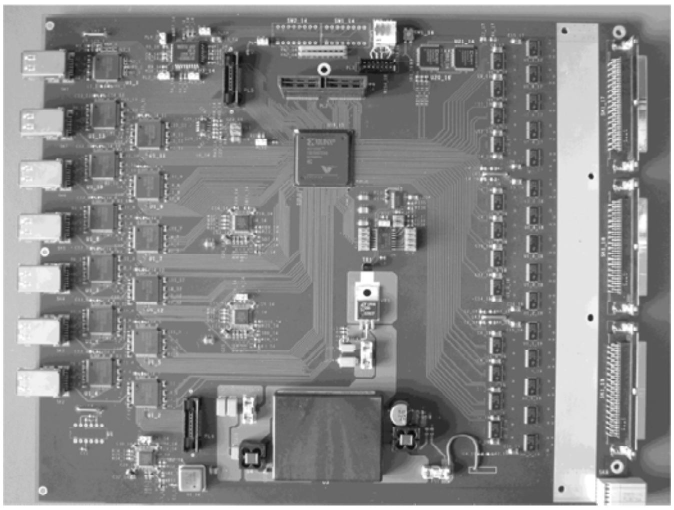

Figure 6: IM pre-production board

D. *Communications Module*

The GCT requires a number of interfaces to external systems that cannot be easily accommodated on a generic module. These include the TTC input, Slink-64 interface to the CMS DAQ, and VME64 interface for fast local DAQ purposes. A single Communications Module (CM), housed within the processor crate, accommodates these interfaces.

The CM receives the TTC clock and control signals via a TTCrq mezzanine board [3]. It distributes a 40MHz low-jitter clock to all TPMs via point-to-point cable links, and sends control information to a dedicated 'DAQ' TPM via a serial link. It receives up to 200MB/s DAQ output and status data from the TPM via four further serial links, and relays this to the DAQ. The CM firmware is responsible for encapsulating output data into a CMS standard DAQ packet.

A prototype CM board is currently under test, and seems suitable for direct inclusion into the final system.

III. FIRMWARE AND SOFTWARE

A. *Trigger Algorithm Firmware*

Algorithm firmware is implemented in RTL-style VHDL for synthesis into FPGA programs. All firmware modules necessary for the baseline GCT trigger functions have now been implemented and tested. Wherever possible, lookup table based decision logic is used, with tables accessible via the control bus. This offers the possibility of rapid adjustment of trigger parameters without recompilation of firmware. Data flowing between algorithm blocks in different FPGAs is accompanied by bunch-crossing identifier signals to allow rapid detection of any synchronization error during operation.

Amongst the issues encountered during firmware testing were problems with signals crossing domains between multiplied clocks generated from the FPGA DLLs; it is difficult to avoid race conditions due to small relative delays on the FPGA internal clock nets, without incurring a latency penalty due to double registering. In order to avoid this issue, all algorithm logic is designed to run from a single phase 160MHz clock, with the 40MHz phase information propagated as 180 degree shifted multiplexer select lines.

B. *Control and Monitoring Firmware*

In addition to algorithm logic, every FPGA in the system contains a latency buffer and derandomizer buffer to allow capture of its input and/or output data for DAQ or debugging purposes during normal trigger operation. In addition, the derandomizer buffer may be used as a combined spy / playback buffer for end-to-end offline tests.

A ring topology bus visits every FPGA on the TPMs, and acts as a shared control / DAQ data path. All bus operations are initiated by a control FPGA, which acts as a bridge to the system VME bus. A simple bus protocol is used, derived from the Wishbone open standard [9]. This allows the use of commercial and open-source cores for functions such as I2C interface, SRAM interface, etc. It also allows the use of a soft CPU core for control functions at a later time. During DAQ operations in response to a CMS Level-1 accept signal, the bus is acquired by a dedicated DMA controller to read out all necessary data from the FPGA derandomizer buffers. The data selected can be flexibly altered depending on the current operation mode and trigger type. All DAQ data is forwarded to the DAQ TPM, and thence to the CMS DAQ via the CM.

C. *Firmware Configuration System*

The GCT contains over 200 large FPGAs, each with a unique configuration. In order to simplify the firmware design, many VHDL modules are parameterized and reused throughout the system. Configuration control becomes an important issue with a system of this scale, and commercial development tools currently do not support this use case well.

In order to address this problem, a firmware configuration and build system ('Fbuild') has been implemented, based on techniques that are well known in software development. VHDL code is contained within a CVS repository to allow controlled simultaneous work by a number of developers. Given a CVS release and a database of configuration information, Fbuild is able to resolve dependencies between firmware modules, and target either synthesis or simulation of a TPM or the whole GCT system by passing appropriate scripts to the backend commercial tools. Fbuild also inserts simple cpp-style macros into VHDL source code to allow a simple form of code parameterisation, and can make simultaneous use of a cluster of CPUs to speed up the firmware build and debugging process.

D. *GCT Software*

The GCT software system, currently under development, consists of the following modules:

- GCT real-time executive, providing an interface to the hardware and setup and reconfiguration on demand.
- GCT configuration management, a component of the central CMS configuration database system.
- GCT offline simulation code, part of the overall CMS ORCA reconstruction package [10], and allowing GCT operation to be compared against simulation.
- GCT run control and monitor, providing an interface to the global CMS run control system and a private control / monitoring interface for testing.
- GCT debugger, encapsulating test, exercise and timing setup software.

Detailed design, implementation and testing of the GCT software system will be a major focus of the project during the integration period up to 2007.

IV. CONCLUSIONS

The GCT is an important part of the CMS Level-1 trigger system, implementing in custom hardware those trigger functions that require data from the entire calorimeter system. The compact but complex nature of the system requires the use of several novel design approaches to minimize cost, risk and design time. These include the use of modern FPGA and serial link technologies, use of a single generic processor module, and comprehensive self-test capabilities.

The complexity and flexibility of the GCT present challenges in firmware and software design, as well as hardware. These challenges are now being addressed during the period before LHC running, using techniques inspired by modern software development tools.

Successful implementation of the prototype GCT has demonstrated that the design strategy is workable and appropriate. The GCT is now entering the production and integration phase, in preparation for LHC start-up (Figure 7)

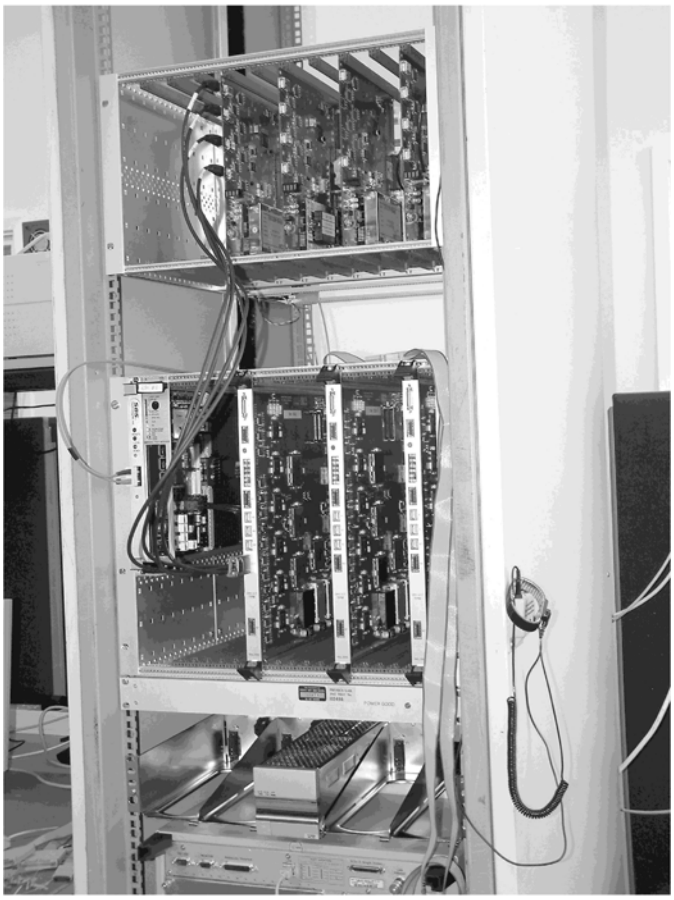

Figure 7: Integration tests of prototype GCT system

V. REFERENCES

[1] CMS Level-1 Trigger TDR, CERN/LHCC 2000-038

[2] CMS DAQ and HLT TDR, CMS/LHCC 2002-26

[3] B. G. Taylor, Proc. Of 8th Workshop on Electronics for LHC Experiments, Colmar

[4] http://www.xilinx.com

[5] http://www.national.com ; http://www.vitesse.com

[6] http://www.valerepower.com

[7] Eye-Opener 6 VHDM; http://www.goreelectronics.com

[8] SN65LVDS352; http://www.ti.com

[9] http://www.opencores.org

[10] http://cmsdoc.cern.ch/orca