

**ATLAS Level-1 Calorimeter Trigger
Cluster Processor Module**

Project Specification (Post PRR)

Version: 2.03

Date: 18 October 2006

Stephen Hillier

Gilles Mahout

Richard Staley

Alan Watson

University of Birmingham

Table of Contents

Table of Contents	2
Document History.....	2
1 Introduction	3
1.1 Overview	3
1.2 Organisation.....	5
1.3 Related projects	8
2 Functional requirements	8
2.1 Serial link inputs	8
2.2 Serialiser and fan-in/-out.....	9
2.3 CP chip and cluster processing	13
2.4 Result merging.....	14
2.5 Read-out	15
2.6 Timing	19
2.7 Configuration and control	22
2.8 Grounding.....	24
3 Implementation	25
3.1 Serial link inputs	26
3.2 Serialiser and fan-in/out.....	27
3.3 CP chip and cluster processing	28
3.4 Result-merging.....	29
3.5 Read-out	30
3.6 Timing	33
3.7 Configuration and Control	34
3.8 Interfaces	38
3.9 Module layout.....	43
3.10 Programming Model	44
4 Project Management.....	54
4.1 Overview and deliverables.....	54
4.2 Personnel	54
4.3 Design and verification.....	54
4.4 Manufacturing.....	54
4.5 Test	55
4.6 Costs	56
References	56
Glossary.....	57
Appendix A: Backplane connector layout	58
Appendix B: Summary of BC-mux logic.....	62

Document History

The history of this document is as follows:

- June 2000: Version 0.8 released for PDR (followed closely by version 0.9 contained some material missing from 0.8)
- July 2000: PDR held at RAL
- October 2000: Version 1.03 released, post-PDR incorporating referees' comments
- March 2005. Version 2.00 released for FDR.
- April 2005. Version 2.01 post FDR corrections.
- June 2005. Version 2.02 submitted for PRR.

1 Introduction

This document describes the specification for the production Cluster Processor Module (CPM) of the ATLAS Level-1 Calorimeter Trigger .

The document is structured as follows: Section 1 gives an outline of the key algorithmic tasks of the Cluster Processor, and describes its organization and context within the Level-1 Calorimeter Trigger; Section 2 describes the functional requirements of the Cluster Processor Module; Section 3 describes the corresponding implementations of these functions, together with some additional technical aspects of the module; Section 4 describes aspects of the management of the project; Section 5 provides a brief summary of the specification and status of the module.

1.1 Overview

The Cluster Processor Module is the main module of the Cluster Processor (CP) of the ATLAS first-level (LVL1) calorimeter trigger. The main tasks of the Cluster Processor are:

- to identify possible isolated electrons, photons and semi-hadronic τ decays using information from the ATLAS electromagnetic (em) and hadronic (had) calorimeters.
- to calculate the multiplicities of e/γ and τ candidates passing different threshold conditions on transverse energy (E_T).
- to transmit these multiplicities as inputs to the LVL1 trigger decision; the e/γ and τ candidate multiplicities are summed from individual CPMs by the Common Merger Module (CMM), which sends the total multiplicities on to the LVL1 Central Trigger Processor (CTP).
- to transmit the co-ordinates and classifications of candidates found in accepted events to the second-level trigger system (LVL2) as Regions of Interest (RoIs) to guide LVL2 processing.
- to transmit DAQ data from Serialisers and Hit sums.

The Cluster Processor is one of three sub-systems of the Calorimeter Trigger. The other sub-systems are:

- the Pre-processor (PPr), which takes signals from the ATLAS calorimeters as shaped analogue pulses, digitises and synchronises them, identifies the bunch-crossing from which each pulse originated (known as Bunch-Crossing Identification or BCID), performs the final E_T calibrations, and prepares the digital signals for serial transmission.
- the Jet/Energy-sum (JEP) processor, responsible for the Jet, missing- E_T and total- E_T triggers.

The Pre-processor provides the input data used by both the CP and JEP systems. The two processors operate independently of each other, and communicate their results separately to the CTP.

The CP system looks for e/γ and τ candidates within the coverage of the ATLAS tracking system ($|\eta| < 2.5$). Its input signals are formed by analogue summation of calorimeter cells to form “trigger towers” of granularity $\Delta\phi \times \Delta\eta \sim 0.1 \times 0.1$, separately in the ATLAS electromagnetic and hadronic calorimeters in this η range. The PPr sub-system receives 64×50 ($\phi \times \eta$) trigger tower signals from the LAr electromagnetic calorimeters, and a matching array of projective trigger towers of the same granularity from the hadronic calorimeters (TileCal and HEC). The PPr outputs to the CP are digital trigger tower signals which are 8-bit calibrated E_T values associated in time with a single bunch-crossing. Since the bunch-crossing identification algorithms suppress off-peak samples from the broad calorimeter pulses, many of the tower signals from any given bunch-crossing will carry zeroes. All trigger tower signals from every bunch-crossing are supplied by the Pre-processor synchronously to the CP at the 40.08 MHz LHC bunch-crossing rate. For the remainder of the document, 40 MHz will imply the LHC frequency of 40.08 MHz unless written as a decimal. Similarly 160MHz will imply 160.32 MHz. (Each 25 ns cycle of the 40 MHz bunch-crossing frequency is colloquially known as a *tick*).

The algorithms used in the CP system to identify e/γ and τ candidates are illustrated in Figure 1 below. The cluster-finding algorithms [1] are performed within *windows* of 4×4 trigger towers ($\phi \times \eta$), each window having two *layers* of matching projective trigger towers, from the electromagnetic and hadronic

calorimeters. The window slides by one tower in both the η and ϕ directions¹ so as to completely cover the calorimeters within the acceptance of the algorithms ($|\eta| < 2.5$). Each window has two parts on each layer, the central 2×2 *core* and the surrounding 12-tower *isolation ring*.

The algorithms utilise the following sums of tower E_T values within an algorithm window:

- a set of four *trigger clusters*, used to define the E_T of the candidate. For the e/γ algorithm these consist of 1×2 or 2×1 pairs of electromagnetic towers within the central 2×2 core of the window. For the τ algorithm each electromagnetic pair is added to the central 2×2 sum of hadronic trigger towers.
- an electromagnetic *isolation sum* (the sum of the E_T in the ring of 12 towers surrounding the central 2×2 region)
- a hadronic isolation sum (corresponding 12-tower ring region in the hadronic calorimeters)
- for the e/γ trigger, the central 2×2 sum of hadronic towers is an additional isolation region.
- an *RoI cluster*, which is the sum of E_T in the central 2×2 towers of both calorimeters.

Convention depicts the algorithm window with η horizontal ($-\eta$ to the left and $+\eta$ to the right) and ϕ vertical ($-\phi$ at the bottom and $+\phi$ at the top). This orientation will be reflected — at least for η — in the layout of the CP crates as viewed from the front. The lower left ($-\phi, -\eta$) tower of the core is referred to as the *reference tower*, and is used to identify the position of the window, although i) all 4 core towers are effectively equivalent, since a window may contain a cluster dominated by any of the four towers, ii) the RoI co-ordinate of any cluster is actually at the centre of the core of 2×2 towers

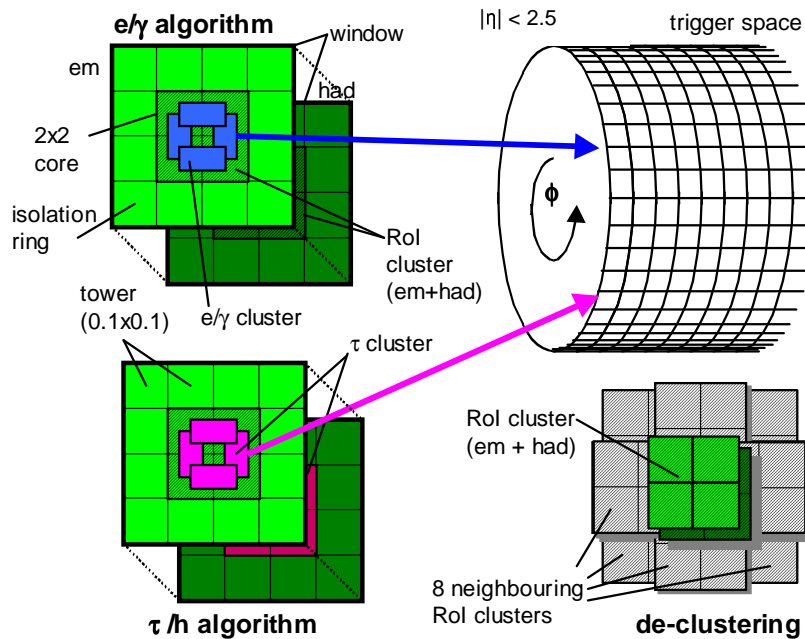


Figure 1: Cluster Processor e/γ and τ algorithms.

For a window to be accepted as containing a candidate trigger object, the following requirements must be met:

- at least one of the trigger clusters must be above a *trigger threshold*.
- all of the isolation sums must be below their corresponding *isolation thresholds*.
- the central RoI cluster must be a *local E_T maximum*, i.e. must be more energetic than the eight other 2×2 clusters that can be formed within the window.

This last requirement is known as *de-clustering* since it eliminates the double-counting of clusters which are jointly contained by adjacent windows. This makes use of the fact that for any window, the RoI clusters of the eight neighbouring windows are contained within the original central window, and so each window can make a comparison of itself to its neighbours, even if the neighbouring windows are

¹ This is a “logical” description — in the implementation all windows will be evaluated in parallel.

actually processed in a different hardware device. To avoid inefficiency due to finite resolution of digital E_T sums, the requirement is loosened to requiring that the central cluster be more energetic than its (overlapping) neighbours at the $+\phi$ and $+\eta$ edges of the window, and at least as energetic as its neighbours along the opposite edges.

The CP system will provide 16 sets of trigger thresholds. Eight of these will be used only for the e/γ algorithm, while the other eight may be used for either e/γ or τ algorithms. Within each threshold set comprising cluster and isolation tests, each threshold can be chosen independently of any of the others.

1.2 Organisation

This section presents details of the organisation of the cluster-finding task within the Cluster Processor Module. The trigger tower inputs to the Cluster Processor naturally form a map of the calorimeters over the two dimensions of pseudo-rapidity (η) and azimuth (ϕ , continuous). This is the *trigger space* within which the cluster-finding algorithms operate; sub-sets of the trigger space applicable to a particular trigger component are referred to as *sub-spaces* or *regions* and are described by the numbers of towers in each dimension as $N_\phi \times N_\eta \times N_{\text{layer}}$ (latter only where appropriate to considering both em and hadronic layers). The Cluster Processor divides its trigger space into four *quadrants* in ϕ , each of which is processed by a single Cluster Processor crate containing 14 CPMs. A CPM processes 64 windows, arranged logically in a 16×4 ($\phi \times \eta$) array. A crate of 14 CPMs is therefore sufficient to process 56 trigger windows in η , while data will only be available from the 50 trigger towers spanning the pseudo-rapidity range of $|\eta| < 2.5$. The CPMs at both ends of each crate will have a number of unused windows and those that are used will have only some of their input towers populated². The quadrant architecture is illustrated in Figure 2.

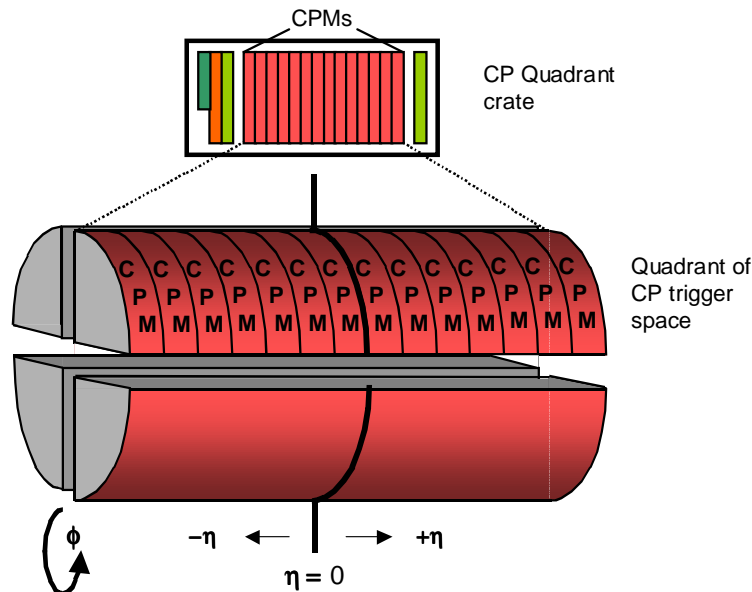


Figure 2: Quadrant crate architecture of CP sub-system.

In this architecture, the reference towers of each CPM are provided by a separate Pre-processor Module (PPM) for the electromagnetic and hadronic layers (one PPM for each layer mapping onto one CPM). In order to form the overlapping windows required by the cluster-finding algorithms, a CPM must share trigger tower data with other CPMs which process regions of the trigger space adjacent in η (within the same crate) or ϕ (in a separate CP quadrant crate). The sharing of data between CPMs is a major function of the Cluster Processor system and is handled in two ways:

- Trigger tower data required by 2 CP crates are duplicated in the Pre-processor and received directly by CPMs in both crates.

² While it would be more efficient to use only 13 CPMs per crate, this would require the central module to span the $\eta=0$ boundary and would not match well with the modularity of the Pre-processor, which is constrained by the modularity of the cables carrying analogue data from the calorimeters.

- Trigger tower data required by 2 neighbouring CPMs in the same crate are received directly from the Pre-processor by one module and copied to its neighbour via the crate backplane.

A feature of this crate/module architecture is that no trigger tower data are required by more than two crates (so only two-fold duplication is required in the Pre-processor), and that within a crate data are required by a CPM and (at most) one of its immediate neighbours (Figure 2). This provides a very simple pattern of data sharing within the system.

Each trigger window on a CPM is displaced by one tower in ϕ and/or η from its immediate neighbours, both on the same module and on modules processing adjacent regions in the same, or another, crate. The set of 64 reference towers of the 64 windows on a CPM are sometimes referred to as the *core* or *fully processed* towers, and are all received directly from the Pre-processor on *serial links*. The remaining towers required to complete these 64 windows are referred to as the *environment* of the CPM. Environment towers which are adjacent in ϕ to the reference towers, are also supplied to the CPM by serial link; the remainder are supplied by *fan-in* from the fast quasi-serial backplane. Each CPM shares 3/4 of its link data with its neighbours by *fan-out* on the same backplane, and receives its environment towers from the core towers of its neighbours.

In order to complete a 4x4 window around a single reference tower, three additional towers in both ϕ and η directions are required. The complete region for a device processing $M \times N$ windows is $(M+3) \times (N+3)$ towers. The same is true of the environment around the 64 windows of a CPM, so that the 16x4 reference towers of a CPM therefore require a region of 19 towers in ϕ by 7 towers in η (core plus environment). The environment towers for a CPM are supplied by fan-in from either duplication at the Pre-processor module or backplane fan-out by a neighbouring CPM in the same crate. The algorithm definition causes fan-in and fan-out at opposite edges of the environment to appear asymmetric in ϕ/η , in the following way:

Fan-in: 2 rows/columns *from* $+\phi / +\eta$, 1 row/column *from* $-\phi / -\eta$
Fan-out: 1 rows/columns *to* $+\phi / +\eta$, 2 rows/columns *to* $-\phi / -\eta$

Trigger towers are combined into pairs in ϕ by a bunch-crossing multiplexing (BC-mux) scheme, in order to make better use of the available bandwidth from the Pre-processor and within the CP sub-system. This scheme employs the fact that the bunch-crossing-identification algorithms force any non-zero trigger tower E_T value to be followed by a zero. This zero-cycle of any channel carrying trigger tower data can therefore be used to carry information about a neighbouring tower from the same or subsequent bunch-crossing, with the addition of a flag bit used to encode which input tower appears in which cycle of the multiplexing algorithm. The result is that any CP module or chip must handle tower data with a dimension in ϕ that is an even number of towers. The CPM therefore receives data from 280 towers, as a region of 20x7 towers on each of the electromagnetic and hadronic layers. Of these, 160 towers are received via serial links directly from the Pre-processor, and the remaining 120 towers as fan-in from the backplane. The “bottom row” (closest to $-\phi$) of towers in this region is unused by the CPM.

The CP algorithms are carried out within a CP chip, which is realised in a large programmable logic device (see section 2.3). Each CP chip processes 8 overlapping algorithm windows, arranged as an array of two in ϕ by four in η , requiring towers from a 5x7-tower region (the BC-mux scheme limits the inputs to pairs of towers in ϕ , so that 6x7 towers are actually read in by each CP chip). A CP chip therefore spans the full width of a CPM in η and the 8 CP chips hosted by the CPM are logically arranged in a 1-dimensional array on the CPM, with their core towers adjacent in ϕ and their environments overlapping. The regions which comprise an algorithm window, a CP chip and a CP module are illustrated in Figure 3.

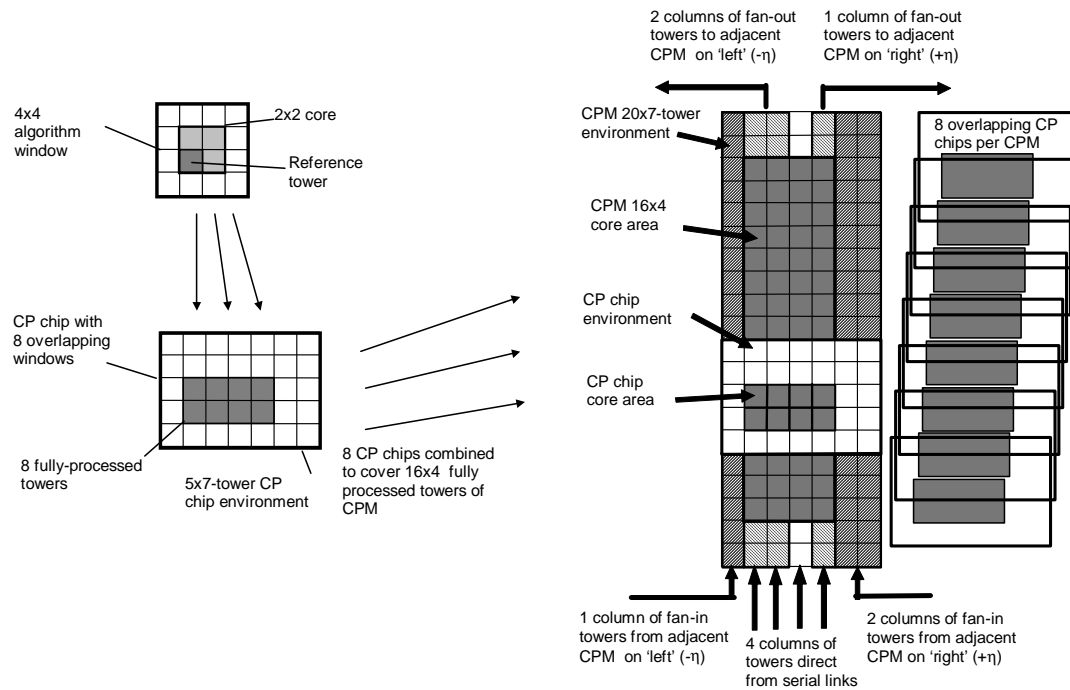


Figure 3: Organisation of trigger towers on the CPM.

The de-clustering algorithm allows only one window in each 2x2 half of a CP chip to produce non-zero results from any set of input data — if there were two clusters within adjacent windows, one or the other would be suppressed by the de-clustering algorithm. The two halves of the CP chip can therefore each produce an independent cluster candidate described by 16 bits, a single bit indicating whether the cluster passed one of 16 threshold sets (a cluster at a particular threshold is referred to as a *hit*). These two 16-bit results are output in parallel by the CP chip on every cycle to be combined with the results from the other CP chips on a module. The overall result from the CPM is the combination of the CP chip results to form three-bit multiplicity counts of trigger objects (saturating at 7 maximum) at each of the 16 threshold sets, known as *hit counts*.

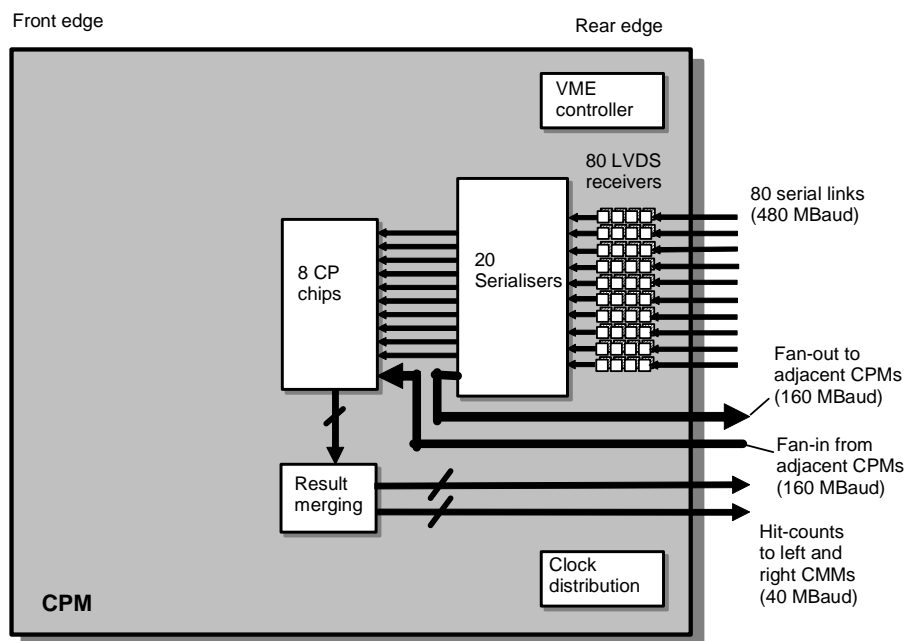


Figure 4: Real-time data paths.

The *real-time data path* of the CPM is defined by the data that contribute to the Level-1 trigger decision, and is illustrated in Figure 4. BC-multiplexed pairs of trigger tower data are transmitted from the Pre-processor as 10-bit words at 400 Mbit/s over 80 serial links to each CPM. The serial link signals are *de-serialised* to parallel data, re-timed and then *re-serialised* to 160 Mbit/s by a Serialiser FPGA for distribution both on the CPM and across the backplane as fan-out to neighbouring modules. Each 160 Mbit/s *stream* carries a 4-bit nibble from the 10-bit trigger tower word, and so two-and-a-half serial streams are required for each tower pair. Data enter the CP chips as 160 Mbit/s streams, to be de-serialised internally and used in the CP algorithms at a pipeline clocking frequency of 40 MHz.

Hit results are output by each CP chip on every 40 MHz tick, from where they enter hit-merging logic to form the module hit counts. Each module transmits its hit counts across the crate backplane as parallel data at 40 MHz to the crate-summing logic on two Common Merger Modules (CMMs), each handling the hit counts from all modules in the crate for eight of the 16 CP threshold sets. The results from all four crates of CPMs are then combined in the system-summing logic of the two CMMs in one of the four quadrant crates. These CMMs transmit the overall results to the Central Trigger Processor, where they form a significant fraction of the inputs to the Level-1 trigger decision for each bunch-crossing. The various steps in the real-time data path, from serial links, through fan-out, cluster processing and result merging are discussed in more detail in Section 2.

1.3 Related projects

The CPM fits within the context of other components of the Level-1 Calorimeter Trigger, which are being developed in parallel. This document embodies a number of requirements specified by the components which are either sited on the CPM (Serialiser and CP chip FPGAs) or with which it communicates (Pre-Processor ASIC and MCM upstream and ROD downstream). All of these components have undergone a Preliminary Design Review of their own, and have or will soon be undergoing their Final Design Review. There is extensive existing documentation to describe these components, accompanied by the Technical Design Report (TDR) [2] for the whole of the ATLAS First-Level Trigger (produced in June 1998). This and some of the documents most relevant to the CPM are listed below (see Glossary for definition of component acronyms):

Level-1 TDR	http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/tdr.html
Algorithm Descriptions	ATLAS note ATL-DAQ-2000-046
PPr ASIC and MCM	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Components.html
CP Chip	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Components.html
Serialiser FPGA	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Components.html
ROD	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Modules.html
TTC	http://www.cern.ch/TTC/intro.html
TTCDec	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Modules.html
TCM	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Modules.html
CMM	http://hepwww.rl.ac.uk/Atlas-L1/Modules/Modules.html

2 Functional requirements

This section introduces the requirements on the functions of the Cluster Processor Module. Details of how these functions are realised are presented in Section 3.

2.1 Serial link inputs

The serial links from the Pre-processor to the Cluster Processor will use Low-Voltage Differential Signalling (LVDS) over shielded (non-twisted) pair cables. A BC-multiplexed³ pair of trigger towers are described by a 10-bit word for every 25 ns bunch-crossing cycle; 8 bits give the calibrated E_T value for one of the towers, one bit gives a BC-mux flag which indicates which tower of the pair it is and on which of the two possible bunch-crossings the tower signal occurred, and 1 bit gives odd parity encoded over the nine other bits. Each link will carry one 10-bit word per bunch-crossing cycle, with an effective bandwidth of 400 Mbit/s per link, carrying data from a (BC-multiplexed) pair of trigger towers every 25

³ See Appendix B for a summary of the BC-mux scheme.

ns. The inclusion of two LVDS synchronisation bits, providing a guaranteed zero-one transition in every 25 ns cycle, increases the link baud rate to 480 MBaud.

A CPM hosts 80 serial links from the Pre-processor, with electromagnetic and hadronic tower data being transmitted from separate Pre-processor modules over 40 links of each kind. A region of 20×4 towers on the electromagnetic and hadronic layers is therefore received directly from the Pre-processor by a CPM, the remaining environment towers being received from neighbouring CPMs via backplane fan-in (see Section 2.2). The serial links are combined in cable assemblies of 4 links, with a single compact two-row connector at each end. The CPM receives 8 cables (32 links) from each of two main PPMs (one electromagnetic, one hadronic) covering the same quadrant. In order to supply the duplicated data shared between two crates at the ϕ edge of a CP quadrant, each module also receives a single cable from four further PPMs, in the two adjacent quadrants (one electromagnetic and one hadronic from each quadrant in $+\!-\phi$).

Each serial link signal goes directly from the incoming cable to a discrete LVDS receiver, which de-serialises the data to a 10-bit parallel word. This parallel word is routed to a Serialiser FPGA, which prepares the data for fan-out to destinations on the same CPM, and to an adjacent CPM across the crate backplane (see Section 2.2 below). The format of the LVDS link, with reference to the parallel I/O is given in figure 5.

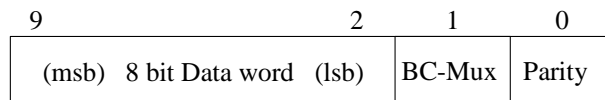


Figure 5: Format of LVDS link data D9..D0

It is important that the connection of cable plant from the pre-processor to the CPM respects both mechanical compactness and electrical integrity of the signals it carries. In order to minimise the disconnection/re-connection of cables when exchanging faulty CPMs and thus to enhance cable reliability, all incoming trigger tower data will be connected via the rear-edge of the CPM, through the backplane. Since these data are required at a number of destinations on the CPM, the cable plant must be distributed over the length of the module's rear edge — this will assist the routing of a large volume of data and help keep signal paths uniform in length. Cable positions along the rear edge of the module will be allocated alternately to cables carrying electromagnetic and hadronic data.

Each LVDS receiver must be supplied with an external clock of around 40 MHz, from which it can lock into the data transmitted at a multiple of the LHC clock frequency (see Section 2.6 for more details of clocking). The Serialiser FPGA receives a “link-locked” output from the LVDS receiver, so that the condition of the link can be monitored and recorded in the data-stream written to DAQ.

The CPM's main functions with regard to the serial links are:

- to route the incoming link signals as differential pairs through the rear edge of the CPM;
- to distribute the links' entry points along the length of the CPM rear edge;
- to provide electrical termination of the serial links;
- to host the serial link receivers;
- to route the parallel data outputs from the serial link receivers to the Serialisers;
- to route the link-locked signals from the serial link receivers to the Serialisers;
- to provide a 40 MHz clock source to the serial link receivers;
- to provide a conditioned power supply to the serial link receivers.

2.2 Serialiser and fan-in/-out

The distribution of trigger tower data within a CP quadrant crate — both on a single CPM and between neighbouring CPMs — is performed by the Serialiser FPGA. Each of the four serial links in a cable is converted by an LVDS receiver into a parallel 10-bit word. The four 10-bit words from a single cable enter a Serialiser FPGA as two 20-bit fields (two words each). Within a 20-bit field, nibbles of 4 bits

are multiplexed up to 160 MBaud streams, with each stream being duplicated four times at the Serialiser output, resulting in a total of 40 output streams. Of the four copies of each stream, up to three go to CP chips on the same CPM as the Serialiser, and the fourth goes via the CP backplane to one of the neighbouring CPMs, where it is sent to up to three CP chips on that module.

Each Serialiser covers a region of 2×4 towers in $\phi \times \eta$ from one cable, which also corresponds to the 8 reference towers of a single CP chip (see section 2.3). There are 20 Serialisers per CPM handling the 80 serial links from the Pre-processor and distributing data to 3 CPMs (the CPM on which they sit and its neighbours on both sides). Each Serialiser handles either purely electromagnetic or purely hadronic tower data, and so the 20 Serialisers can be viewed as two identical groups of 10; for every 'electromagnetic' Serialiser there will be a corresponding 'hadronic' Serialiser covering the same $\phi - \eta$ region of trigger space and routing its outputs to an identical set of destination CP chips. A group of 10 (em or hadronic) Serialisers are labelled (from $-\phi$ to $+\phi$) V, A-H and W, where the central block of 8 Serialisers (A-H) correspond to the reference towers of the 8 CP chips on the module, which are also labelled A-H. The mapping between Serialisers and the CP chips on the same module is shown in Figure 6. The Serialisers V and W handle the towers that are actually in an adjacent quadrant of the trigger space (and duplicated by the Pre-processor) and are used purely as environment towers on the CPM in question (they are reference towers for CPMs in the CP crates allocated to adjacent quadrants). For the V Serialiser only one row in ϕ of the trigger tower data is actually required by the CPM, while for the W Serialiser, both rows in are required.

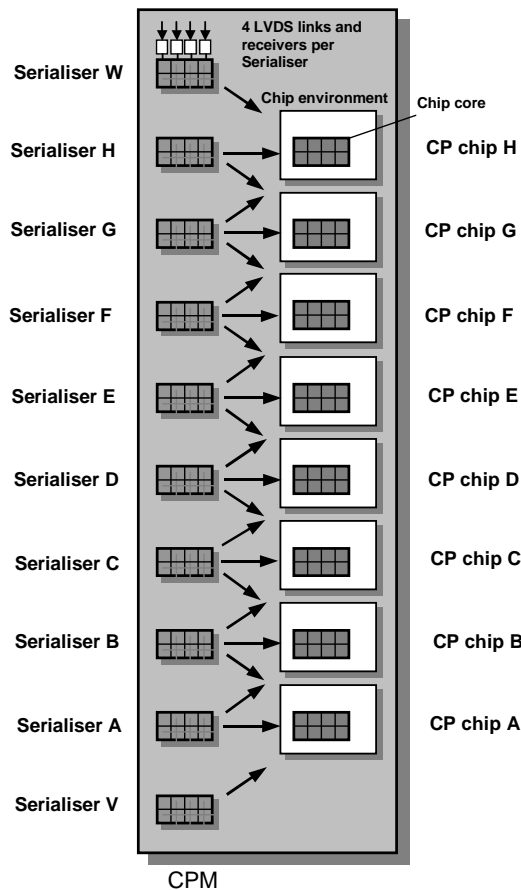


Figure 6: Mapping between Serialisers and CP chips on a single CPM; the CP chip environments are actually overlapping, so that the dark-shaded core regions are adjacent.

The trigger tower data handled by one Serialiser are needed by up to three CP chips on the same CPM as the Serialiser; the Serialiser supplies the reference towers of the CP chip with the same label and the environment towers of CP chips to $-\phi$ and $+\phi$. The 4 Serialisers processing towers that are almost at the edge of the CPM's environment in ϕ (A and H, em and hadronic) have only 2 CP chips on the module

to which they send their data, and the V and W Serialisers have no corresponding CP chips, and so supply environment towers only (to CP chips A and H respectively). In addition, three of the four tower pairs in a Serialiser are required by up to three CP chips on neighbouring CPMs, to which they are sent by backplane fan-out.

Each half of a Serialiser can effectively be viewed separately to the other half, since one half will be fanning out data to $-\eta$, and the other half will be fanning out data to $+\eta$. The 20-bit parallel data inputs of the Serialiser halves are labelled DIN_X and DIN_Y respectively. The CPM will allocate the X inputs to the $-\eta$ half of the Serialiser, and the Y inputs to the $+\eta$ half. A half-Serialiser takes a pair of 10-bit parallel words and re-serialises them into a “bus” of five 160 MBaud streams, shown as AQ[4:0] in figure 7, for example. Each bus is duplicated 4 times, and labelled as AQ—DQ for the X inputs, and AR—DR for the Y inputs. Up to 3 of the bus copies feed CP chips on the same CPM, and one copy (always the D copy) will be fanned out over the backplane to an adjacent CPM (over a short point-to-point backplane link). In the special case of fanout to a neighbouring CPM in $+\eta$ only 3 of the 5 outputs, DR[2..4], are needed.

To aid capture of the data by the receiving CP chip, the Serialiser will delay output of data sent over the onboard path, in order to compensate for the longer transmission time seen by the offboard/backplane path.

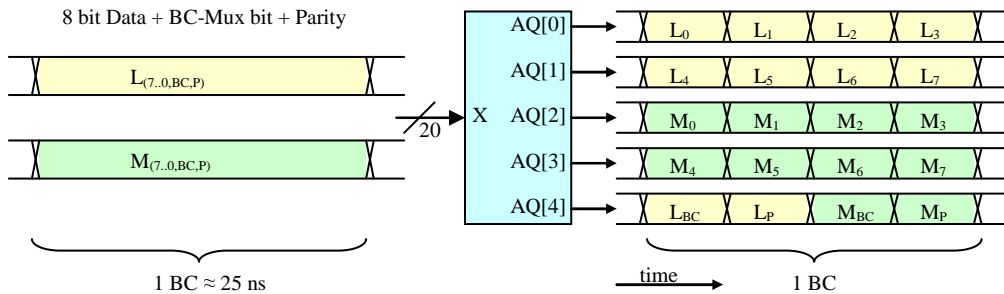


Figure 7: Format of the serial 160 Mbaud streams

Each 10-bit tower pair occupies two entire Serialiser output streams (jointly carrying eight E_T bits) and half of a third which carries the BC-mux flag and parity bits for each tower pair (it is the AQ[4] and AR[4] streams which are shared between tower pairs). The tower data remain BC-multiplexed and parity-encoded in the output streams, being decoded for real-time usage by each destination CP chip. However, the Serialiser also decodes the BC multiplexing and pipelines the individual tower data for later readout to DAQ, if and when a Level-1 Accept (L1A) signal is received (see sections 2.5 and 3.5). The allocation of towers to a Serialiser and the labelling of its inputs and outputs are shown in Figure 8.

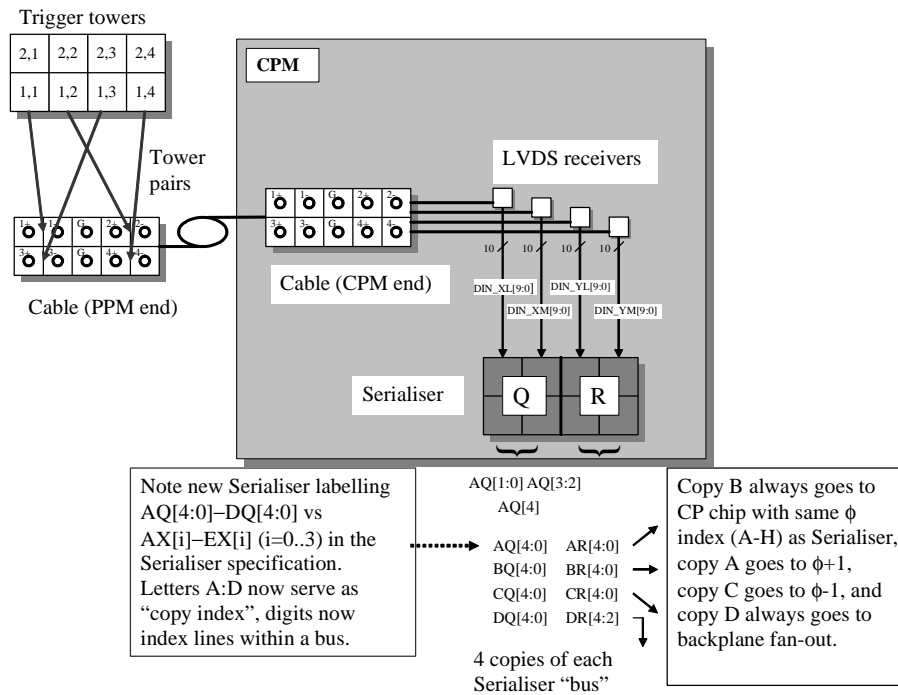


Figure 8: Assignment and labelling of Serialiser I/O.

For fan-out in the $-\eta$ direction, both tower pairs from the X half-Serialiser (i.e. one copy of each of the 5 S_Data_X streams) are sent via the backplane. For fan-out in the $+\eta$ direction, only one of the tower pairs from the Y half-Serialiser is required, meaning that only 3 of the 5 160 MBaud streams from that Serialiser half are required to be sent to the backplane. Note that the asymmetry mentioned in section 1.2 is realised here; the Serialiser must not only select the correct half of the Serialiser outputs to be fanned out in the $\pm\eta$ directions but also select the correct lines for the streams specifying the fan-out data required. All Serialisers send one copy of 8 out of 10 serial output streams to the backplane (the remaining two output streams carry tower data that is not fanned out to neighbouring CPMs). The distribution of tower data is shown schematically in Figure 9.

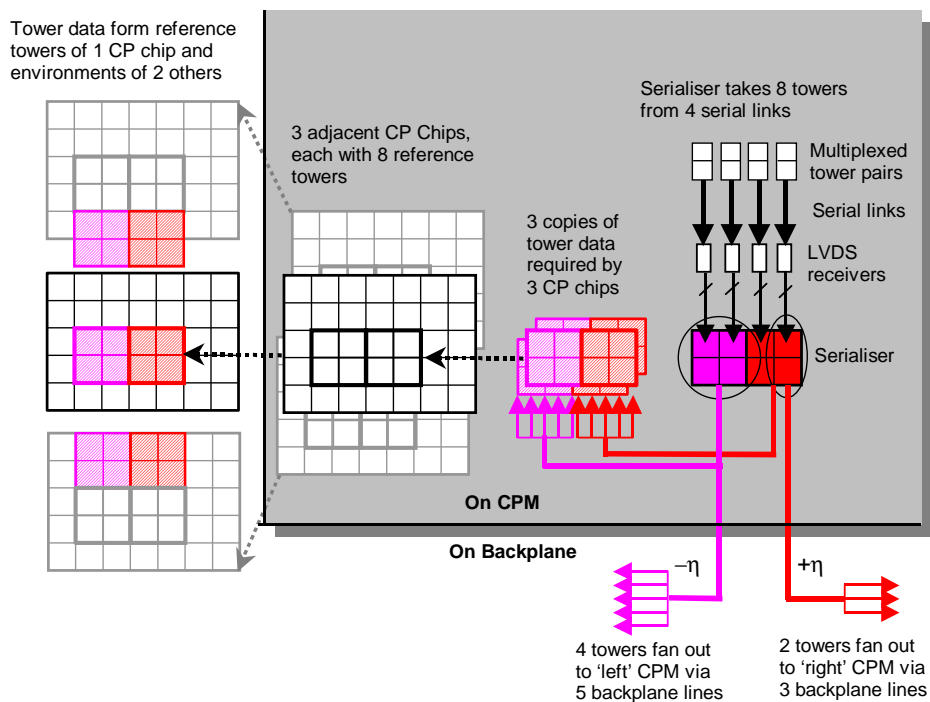


Figure 9: Dataflow through the Serialiser.

In total, a CPM will fan-out 120 of the 160 trigger towers it receives on its 80 serial links, requiring 160 backplane fan-out lines to do so (5 lines for each of 10×2 — $\phi \times$ layers — Serialiser halves required for $-\eta$ fan-out, and 3 lines for each of the 10×2 tower pairs required for $+\eta$ fan-out). As well as fan-out of data to neighbouring CPMs, each CPM receives an equivalent amount of fan-in data from its neighbours via the backplane, on another 160 backplane fan-in lines. The 320 backplane fan-in lines, plus associated grounding, are a major fraction of the module rear-edge connectivity discussed in section 3.8. Each of the incoming fan-in lines must be routed to up to 3 CP chips, for which it supplies environment data from the columns of towers in η adjacent to the fully processed towers. This must be done with a short routing to keep propagation delays comparable with on-board Serialiser data, and with appropriate termination.

The Serialiser provides an internal error counter for each of its four channels. A Parity-Error output from each Serialiser is latched high when any of the four error counters is non-zero. The state of each line is made VME-accessible by the CPM as a single bit in one of two Parity Error registers (separately in groups of ten for “electromagnetic” and “hadronic” Serialisers), described in Section 3.10. Similarly, the Serialiser also has a Link-loss output line which is the OR of the link loss signals for each of its input LVDS links. This signal from each Serialiser is made VME-accessible as a single bit of the Link Loss register (again, two registers describing groups of ten Serialisers), and is also made visible as an LED on the CPM front-panel (see Section 3.8).

The input tower data to the CPM is de-multiplexed and pipelined by the Serialiser for read-out to DAQ in response to a Level-1 Accept signal. This function is described in Section 2.5. The Serialiser also has two additional modes — Playback and Synchronisation — which must be enabled by the CPM’s VME controller. The Playback mode requires that the Pipeline memory is loaded with data via VME, which is then sent to the Serialiser outputs when the En-Playback line is asserted. The Synchronisation mode allows the Serialiser to synchronise its four LVDS input channels to one of four phases with respect to the Serialiser’s 40 MHz clock (the selected phase for each channel can also be loaded into the Serialiser directly). The input INIT_SYNC starts the synchronisation process, and its completion is indicated by the output SYNC_DONE; these signals must be sourced and monitored respectively by the CPM VME controller.

One final requirement for the prototype CPM is that some of the additional unused pins of each Serialiser FPGA should be routed to the module controllers (VME controller, ROC and result-merging logic) to provide flexibility if anything has been overlooked in the module design.

The CPM’s main functions with regard to the Serialisers and the fan-out of serial data are:

- to distribute copies of the ten 160 MBaud output streams to up to 3 on-CPM CP chips;
- to route one copy of eight out of ten 160 MBaud output streams from each Serialiser to a backplane fan-out link;
- to provide VME access to the Serialisers;
- to provide rapid configuration of Serialiser FPGAs from VME-accessible memories.
- to provide 40 MHz clock signals from the TTC system (see section 2.6);
- to provide a conditioned power supply;
- to provide Parity Error and Link Loss registers for monitoring serial data errors;
- to send the Link Loss signals to a front panel indicator (see section 3.8).

2.3 CP chip and cluster processing

The Cluster Processing algorithms described in section 1 are implemented in the eight CP chips on the CPM. The CP chip is implemented as a Field Programmable Gate Array (FPGA) which is downloaded *in situ* with configuration data that defines the logic to implement the algorithms. The complexity of the algorithms means that only recent generations of FPGA have been both large enough and fast enough to be suitable for cluster processing. The FPGA choice yields considerable benefits in flexibility, particularly at the development stage.

The input pin-count to the CP chip is minimised by using the 160 MBaud streams from the Serialiser, and by decoding the BC-multiplexed input data only within the CP chip. The CP chip has 108 input pins $((3 \times 3 \times 5 + 3 \times 1 \times 3) \times 2)$ for 160 MBaud tower data, through which data for 42 $(3 \times 7 \times 2)$ tower pairs are sent for each 25 ns tick. In parallel to the de-multiplexing of BC-multiplexed tower data within the CP chip, the parity of the data is also checked, and if found to be inconsistent with the received parity bit, the data for a multiplexed tower pair is zeroed (the BC-multiplexing flag may also have been corrupted, so errors cannot be associated to a single tower within a pair). The CP chip maintains an error map register with one bit for each of 42 BC-multiplexed tower pairs (each spread across 3 input lines) which is set when a parity error is detected. The CP chip also maintains an internal counter which is incremented for each bunch-crossing in which a parity error is detected for any of the 42 tower pairs, providing an indication of the rate of errors in serial data reaching the chip. Both the error map and the error counter are VME accessible and will be reset after being read. The internal error map bits are ORed by the CP chip and available as a latched output line (Error). This signal is accessible as a single bit of a CPM CP-error register (see section 3.10) and as a front-panel indication for errors in the serial data of each chip (see section 3.8).

The CP chip is clocked by the CPM at 40 MHz; any faster clocks required for the 160 MBaud input streams or internal pipelines are provided by DLLs within the CP chip. The measured latency of the CP chip is 6.4 ticks, including the input data de-multiplexing and error-checking. On each tick the CP chip outputs 2 hit bits for each of the 16 threshold sets, indicating if a cluster passing that threshold set was found in each half of the CP chip. Due to the algorithm latency, the set of hit bits result from tower data which were clocked into the CP chips several bunch-crossings earlier. The hit bits form the inputs to the result-merging logic described in Sections 2.4 and 3.4, where the results from the CP chips on a CPM are combined.

For bunch-crossings for which an L1A is received, the CPM also sends information on its trigger objects to the Level-2 trigger as Regions of Interest (RoIs). The RoI data from each bunch-crossing are pipelined on the CP chip until they are read out under the control of dedicated read-out logic on the CPM. The CP chip sends a word for each possible RoI (i.e. one for each half CP chip) to a read-out link, regardless of whether any of the threshold sets were passed for that RoI. Read-out from the CP chip is described in Sections 2.5 and 3.5.

The CP chips will be implemented as FPGAs, and like the Serialisers, they will require configuration data to be supplied to each device (see scheme described in Section 2.7). The CP chip has a number of registers which are VME-accessible through the CPM, specifying parameters of the algorithms and providing control and monitoring functions for the CP chip. Configuration and monitoring functions provided by the module are described in Section 3.10. One final requirement for the prototype CPM is that some of the additional unused pins of each CP chip FPGA should be routed to the module controllers (VME controller, ROC and result-merging logic) to provide flexibility if anything has been overlooked in the module design.

The CPM's main functions for the CP chips are:

- to route input tower data from the Serialisers;
- to route CP chip outputs to the result merging logic;
- to provide the read-out control signals which manage RoI read-out;
- to route RoI output data from the CP chips to the RoI read-out controller;
- to monitor the error line from each CP chip, as a bit in a VME register and front-panel indicator;
- to provide VME access to the CP chips;
- to provide rapid configuration of CP FPGAs from VME-accessible memories.
- to provide 40 MHz clock signals from the TTC system;
- to provide a conditioned power supply with appropriate levels for I/O and core logic.

2.4 Result merging

Each CP chip will output two 16-bit hit words, describing which thresholds have been passed for the possible RoI in each half of the CP chip (in the vast majority of cycles, none of the 16 threshold sets will be passed). Each bit of a given hit word forms a one-bit input into a sum across all 16 CP RoIs,

made independently for each threshold set. The resulting sum saturates at 7 (binary 111), so that the module multiplicity result can be expressed as one 3-bit count per threshold set⁴. Thus for 16 threshold sets there are a total of 48 result bits from each CPM.

The results for each module are then sent across the CP crate backplane to two Common Merger Modules (CMM), each of which combines the module hit counts for eight thresholds independently, for all modules in a crate. The CMMs in one of the four crates combine the results for all four crates before transmitting the total CP multiplicities at each threshold to the Central Trigger Processor. Each CMM deals with only eight of the 16 CP threshold sets, requiring 24 bits from each CPM. The 24 bits from each CPM are also combined into a checksum from which a 25th odd parity bit is derived (the parity bit is generated slightly later than the hit counts). The hit counts are transmitted in parallel at 40 Mbaud across the backplane to CMMs at either end of the backplane. The hit count outputs will be latched, prior to transmission onto the backplane. The lower threshold sets (0 – 7) representing electron hits are sent, over the backplane, to the CMM in the JMM slot on the right, and the upper threshold sets (8 – 15) nominally representing Tau hits are sent to the CMM in the SMM slot on the left.

The module hit counts at each threshold must also be sent to the read-out control (ROC) logic, where they are pipelined before being appended to the Serialiser data read out as part of the DAQ data-stream. The individual 16-bit words describing the hit results from CP chip-halves are also the basis of the RoI information transmitted to the Level-2 trigger after a L1A (see section 2.5 for both DAQ and RoI read-out). The implementation of result-merging logic is discussed in section 3.4.

The CPM's main functions for the result-merging logic are:

- to route each hit bit to the correct threshold bit-field of a hit sum in the result-merging logic;
- to route the results to the backplane for transmission to the crate CMMs;
- to route the module hit-counts at each threshold to the read-out control logic;
- to provide 40 MHz clock signal from the TTC system, with programmable-delay strobe;
- to provide a conditioned power supply.

2.5 Read-out

For the Level-1 Calorimeter Trigger, which is primarily a synchronous real-time system, the term *read-out* means the export of data in response to a Level-1 Accept (L1A) signal arising from a Level-1 trigger decision, as distinct from part of the process leading to the decision. The read-out may proceed with variable delay after the L1A signal is received, and is therefore not part of the synchronous real-time data path. The read-out involves a somewhat complex interaction between logic blocks on the CPM, including the two kinds of large FPGA which it hosts – the Serialisers and CP chips – which source much of the read-out data. This read-out process will be fully described here, in order to make clear the requirements for CPM module logic to comply with the parts of the read-out function implemented by the large FPGAs. In addition, the scheme has evolved considerably since the PDRs of those FPGAs, and so this later document should embody the latest version of the scheme for future reference⁵.

Data are read out from the CPM for two purposes. Firstly, the input tower values from the Serialiser FPGAs and the corresponding hit outputs from the CP chip FPGAs are read out to the ATLAS DAQ system as for any detector sub-system. These data can be used for off-line monitoring in order to verify the function of the CP system and to understand the performance of the calorimeter trigger. Secondly, RoI data describing the trigger features identified by the CPM must be sent to the Level-2 trigger, providing the starting point for its analysis. Both DAQ and RoI data are read out from the CPM when an L1A signal is received from the TTC system, indicating that a particular bunch-crossing has been triggered. The two kinds of data are read out in two entirely independent streams, through separate, but similar, hardware channels, some time after the real-time processing of the Calorimeter Trigger is completed.

⁴ Ditto results on a CMM from a single crate and from the entire CP sub-system — a three-bit count per threshold set is allocated at all stages, with any count in excess of 7 saturating all three bits.

⁵ Although the Serialiser and CP chip specifications have, of course, been brought up to date, and in fact contain more detailed information about the exact timing of the read-out signals.

CPM read-out is a two-step process:

1. On receipt of an L1A signal from the TTC system, pipelined read-out data is immediately secured in a FIFO buffer;
2. When the read-out link is available, data is moved from the FIFO buffer to a shift register and scrolled out to the read-out link which carries data from the CPM.

These two steps can be carried out essentially asynchronously, buffered by the capacity of the FIFO to absorb short-term rate differences between the two steps. The L1A signal arrives back at the CPM with fixed and known latency after the data from which the L1A was generated had entered the pipeline, and the address of the data to be read-out can be derived by converting this latency to a write-read offset, held in a programmable register. While the FIFO is not empty, the shift register is continually shifting out data to the read-out link, receiving a new input word from the FIFO as it completes each previous word.

The data-flow diagram for the read-out from the CPM is shown in Figure 10. The majority of the read-out data is generated and pipelined within the Serialisers (DAQ data) and CP chips (RoI data), before being exported from the CPM over separate DAQ and RoI high-bandwidth serial read-out links to Read-Out Driver (ROD) modules. The read-out functions on the CPM are managed by the read-out controller logic (ROC), which interacts with the read-out sequencer logic in the Serialisers and CP chips. There is an independent read-out controller for each of the DAQ and RoI read-out functions on the CPM, which will be similar in implementation, although the RoI ROC has reduced functionality and slightly different data formatting. The read-out controllers' two main tasks are to initiate the read-out on receipt of an L1A signal and to control the transmission of the subsequently buffered data from the module via shift register and serial read-out link. In addition, the controllers provide additional data to be integrated with the data streams read from the Serialiser and CP chip pipelines. The ROCs will be implemented as two separate FPGAs, and controlled by means of VME-accessible registers.

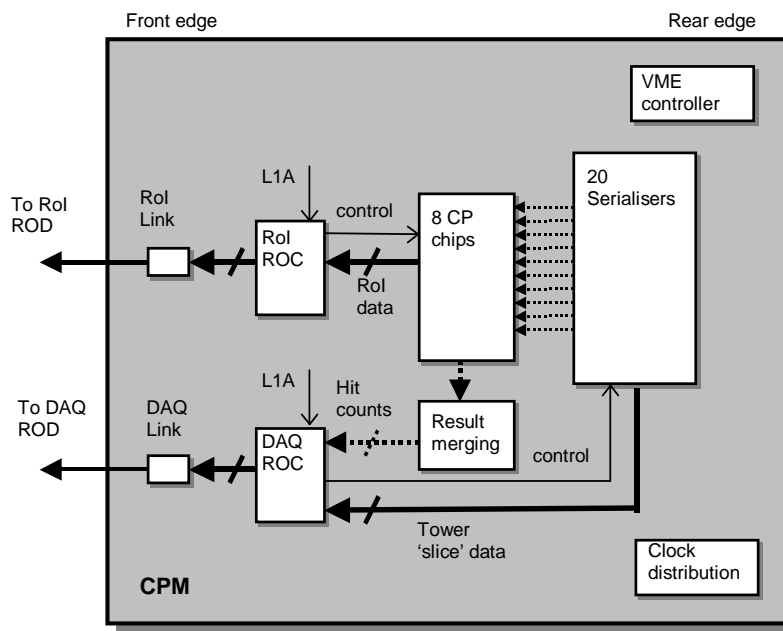


Figure 10: Read-out data-flow diagram (real-time data shown as dashed lines).

For both streams, data describing the entire module (inputs and outputs respectively for the DAQ and ROI streams) are pipelined for every bunch-crossing in Dual-Port RAMs (DP-RAMs) within the Serialisers and CP chips. These pipelines have local write address counters which increment with every bunch-crossing. They also have read address counters which are offset from the write addresses by a number of pipeline locations corresponding to the latency between data entering the pipeline and the L1A signal being received by the CPM. Local read address counters within the FPGAs reduce the number of FPGA input pins required for read-out compared to provision by the ROC of a single pipeline read address to all devices. In order to keep all devices synchronised, the ROC must provide

regular resets to all address counters (after a number of ticks which is an integer multiple of the length of the pipelines). When reset, the write address returns to zero and the read address returns to the offset relative to zero. The pipelines are *not* addressed by any derivative of the bunch-crossing counter, since this is reset at a non-integer multiple of the pipeline length (every 3564 ticks), which would result in some data being secured for less than the full pipeline length.

The data for a single bunch-crossing associated with a trigger are referred to as a *slice* of data; for DAQ read-out only, multiple slices of data may be retrieved from the pipelines for a single received L1A. In order to make use of a single link for the (DAQ or RoI) data from an entire CPM, the slice of data from a single device (Serialiser or CP chip) is presented bit-wise to a single bit field of the parallel data input of the read-out link. It therefore takes a significant number of ticks to read out a single slice of data (depending on the read-out slice length, which differs for the DAQ and RoI channels) and this slice read-out time greatly exceeds the minimum interval between two L1A signals, which may occur with a minimum separation of only 5 ticks. In order to accommodate a sequence of L1As that arrive at a rate above the long-term average, data from the DP-RAMs are buffered in a FIFO before being loaded into a shift register and presented to the read-out link. Like the DP-RAMs, the FIFOs and shift registers are integral to the Serialisers and CP chips and are controlled by signals from the ROCs. In order to decouple the FPGA shift registers from the read-out links and allow the ROC to re-time bit-streams from multiple sources, the shift registers send their slice data to the read-out link via the ROC. This also simplifies the provision by the ROC of additional data in each bit-stream (see below).

The ROC controls the number of slices that are read from the pipelines for each L1A, in the ranges 1 to 5 for DAQ read-out, and 1 slice only for RoI read-out. The ROC must also know the length of the shift registers, which determines the length of time it takes one slice to be read out, and then as each slice is completed the ROC will synchronise the transfer of each subsequent slice from the FIFOs to the shift registers. Finally, for the duration of slice data which comprise a single trigger, the ROC must provide a framing signal which can be carried with the data on the read-out link to indicate to the receiving ROD that a sequence of bits are all part of a single triggered event.

In addition to the data held within the Serialiser and CP chip pipelines, the ROC provides further data to be appended to the Serialiser and CP chip slice data as part of the DAQ and RoI streams. Firstly, the hit results from the result-merging logic must be pipelined and buffered within the ROC to be read-out as part of the DAQ stream at the same time as the input slices from which the results were calculated (note that the result data enter their pipeline with different latency to the corresponding input tower data and so must be read-out with a different write-read offset). Secondly, the bunch-crossing number (BCN) for the data being read out must also be appended to that data, for both DAQ and RoI streams. Finally, a parity bit must be calculated for each read-out bit-stream over the slice data and appended hit counts or BCN bits. In order to manage the appended data, each ROC replicates some of the functions of the read-out logic within the Serialisers and CP chips, by having its own synchronous pipelines, FIFOs and shift registers, and managing them in the same way as those external to the ROC.

One evolutionary step in the read-out scheme has been the decision to route all read-out data through the ROC, so that data from the external and internal pipelines can be easily integrated. This has the advantage of bringing all serial streams to a single point before sending them on to the read-out link, which facilitates routing. This would also simplify any change in the link implementation, since it is only the ROCs which deal directly with the links rather than the complex larger FPGAs. When a slice of data from the external pipelines has been scrolled out of the shift register, the ROC can switch to the internal FIFOs to provide the data to be appended. The ROC synchronises the external and internal pipelines, FIFOs and shift registers so that coherent data relating to the same bunch-crossing are read out in each slice. The possibility of streams getting out of step is controlled by monitoring the state of the external FIFOs via their FIFO-EF (empty flag) signals; if the ROC finds any of the external FIFOs in a different state to its internal FIFOs it generates an error signal.

Serial transmission of the read-out links will use the Agilent G-Link format, transmitted over Multimode fibres. Data is exported over the read-out serial links in uncompressed form and the read-out driver modules (ROD) compress and format the data for storage and later use.

One ROD per CP crate will be required for each read-out type. A common ROD module of a single design will be used, but with data-processing appropriate to one or other application implemented in programmable logic. The compressed data are then sent on to a Read-Out Buffer (ROB) in the case of DAQ data, or to a Level-2 RoI Builder (RoIB) in the case of RoI data. The bandwidth of the read-out

links is more than sufficient for the data-flow resulting from the average rate of Level-1 Accepts, which is limited by the CTP to at most 100 kHz.

One further requirement is that of VME access to the entire pipeline in order to obtain a long-duration picture of the state of the module's inputs and outputs. This can be done by setting the number of slices to be read from the pipeline into the FIFO to equal the FIFO length⁶, initiating the transfer from the pipeline with an "artificial" L1A and suspending all "real" L1A's until the FIFO has been read by VME (FPGA FIFOs are directly accessible to VME; the same is required of the ROC FIFOs). Note that this suspends trigger operation and normal read-out until as many FIFOs as are required in a crate have been read, and is thus to be used only in diagnostic modes.

In summary, the CPM ROC performs the following functions:

- Initiation of read-out on receipt of L1A signal from the TTCrx;
- Provision of read/write counter resets to external and internal pipeline address counters;
- Loading of data into external shift registers from FIFOs when previous slice is complete;
- Signalling on the read-out link that valid event data is being scrolled from the shift registers;
- Re-timing of external shift-register inputs before output to the read-out link;
- Provision of internal pipelines, FIFOs and shift registers for appended hit-count/BCN data;
- Provision of an internal bunch-crossing counter, reset by a TTCrx signal;
- Calculating and appending parity for each individual bit-stream;
- Multiplexing of externally and internally pipelined data;
- Monitoring all FIFO empty/full states, generating an error when different to internal FIFO.
- VME access to pipeline contents by transfer to the FIFO.

The implementation of these functions is described in more detail in section 3.5.

2.5.1 DAQ read-out

For the DAQ read-out, each of the 20 Serialisers provides data through its shift register output to a ROC input, which passes on that data (after re-timing and the appending of hit bits and parity) to a single bit-field of the 20-bit-wide input to the DAQ read-out link. All 20 Serialisers are thus read out in parallel through a single link.

The Serialiser data comprises eight tower E_T values from a single bunch-crossing, after BC-multiplexing of the Serialiser's four input streams is de-multiplexed, converting 4 tower pairs into 8 tower values. The DAQ data word contains some additional information on the serial link status for each of the eight towers handled by the Serialiser in a given bunch-crossing, plus either 3 result bits for one of the 16 threshold sets (on link fields D0..D15), or a group of three bits from the 12-bit bunch-crossing number (on link fields D16..D19). The first 80 bits (the tower data) are supplied from the shift register of each Serialiser to one input of the DAQ ROC. The hit data are received in real-time from the local CPM result-merging logic and managed in pipelines, FIFOs and shift registers by the DAQ ROC, in a similar way to the slice read-out from the Serialisers. The bunch-crossing number for each L1A is generated locally by a 12-bit bunch-crossing counter in the ROC. The entire DAQ word is protected by an appended parity bit generated by the ROC, which is added when the preceding 83 bits of the DAQ slice have been clocked through by the ROC. The format of the DAQ read-out data is shown in Figure 11.

Up to 5 slices may be read out in association with a single L1A, and the chaining of slices is managed by the ROC. Since the event header produced by the ROD from this data will take the event bunch-crossing number from the first slice received, all slices in a multi-slice read-out should carry identical bunch-crossing number information, which indicates the true BC number for the received L1A. All 128 slices in the pipeline may be transferred to the FIFO to be read via VME (when ordinary L1A's are suspended).

⁶ Pipelines and FIFOs both have 128 locations, allowing the FIFO to hold the entire pipeline contents.

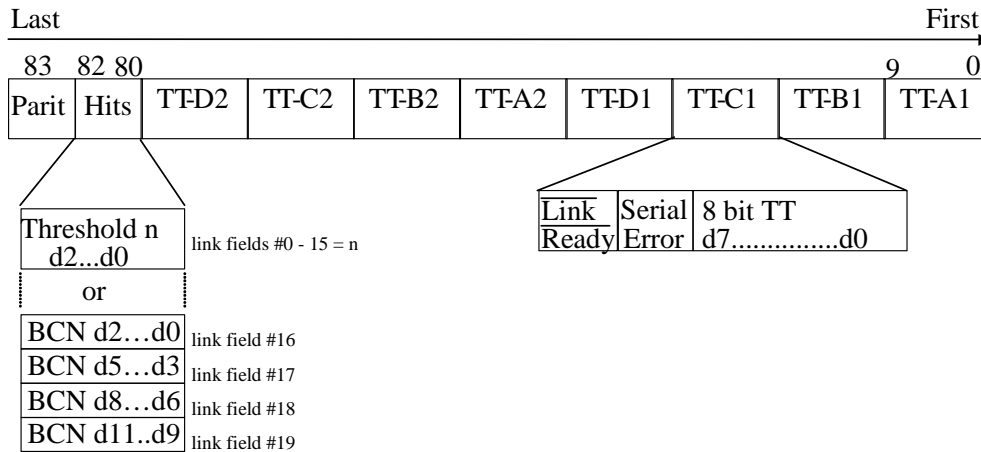


Figure 11: Format of DAQ data slice on read-out links D0..D19.

2.5.2 RoI read-out

The RoI read-out uses only 16 of the available bit-fields, each field being dedicated to an RoI from half of a CP chip. The RoI data consist of a mask indicating which of the 16 threshold sets an RoI candidate has passed, some bits indicating whether this RoI contained a saturated tower or a detected error, and 2 bits to indicate which one of the 4 possible RoI positions in half of a CP chip was occupied. The CP chips supply their hit RoI data from internal shift registers to 16 of the 20 ROC inputs. The RoI ROC does not need an internal pipeline, since there is no real-time data equivalent of the hit outputs to be read out to the RoI stream. It does store the BCN in a FIFO for each received L1A, and appends one bit to the 20-bit RoI bit-stream on link fields D0..D11. Each RoI slice is parity-protected, by a parity bit generated and appended by the RoI ROC. The format of the RoI slice data is shown in Figure 12. Only one RoI slice will be read out after an L1A is received, although all 128 slices in a pipeline can be transferred to the CP chip FIFOs to be read via VME.

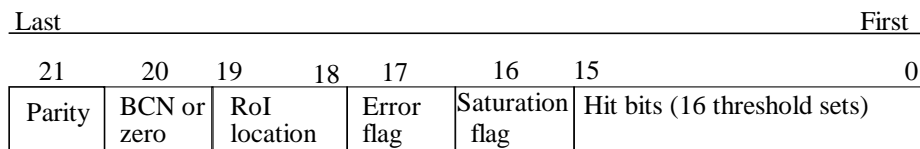


Figure 12: Format of RoI data slice on read-out link D0..D15.

2.6 Timing

The CPM will obtain its clock from the ATLAS Timing Trigger and Control (TTC) system, which also provides global control signals, such as Level-1 Accept to initiate detector read-out. The TTC network is divided into partitions (one per detector sub-system) and sub-partitions, within which all TTCrx devices will receive the same signal. The TTC signals are distributed through a sub-partition optically, with both clock and data encoded in a single 160.32 MBaud channel. Within each CP crate, the optical signal is converted to an electrical signal by a Timing and Control Module (TCM) and then distributed differentially as point-to-point channels on the CP backplane. Each CPM hosts a single TTCrx chip mounted on a small TTC decoder card (TTCdec) from which the LHC clock(s) and a number of control signals will be derived locally by decoding the encoded digital signal.

Data transmission to and within the CP system is all based on the 40.08 MHz LHC clock distributed by the TTC system. Where transmission is at a multiple of this frequency, for example over the serial links from the Pre-processor or over the CP backplane, these clocks are generated internally by the transmitting device, based on a supplied 40 MHz clock. The distribution of a coherent high-quality 40 MHz clock to each Serialiser (20) and CP chip (8) as well as to the Result Merging and Read-Out

Control logic and the serial and read-out links is therefore the most vital timing function of the CPM. Distribution and clock stability will be improved by using high-quality PLLs on the CPM.

The TTCdec provides three 40 MHz clocks, two of which, Clk40Des1 and Clk40Des2, are phase-adjustable by each TTCrx chip. The CPM will supply Clk40Des1 for the Serialisers, ROC and result-merging logic and Clk40Des2 for the CP chips. The Serialisers and CP chips are clocked with different phases to allow timing scans to be performed and the timing margins of the FIO signals between the Serialisers and the CP chips to be measured.

The CP chips will see FIO signals arriving with slightly different times to one another, depending upon the paths taken, whether across the board or via the backplane. The CP chip must capture the 160 MBaud signals at 160 MHz with sufficient timing margins well away from any level transitions, using a TTC derived clock. Fine tuning of the clock phase feeding each individual input is not possible, so instead the CP chip will internally generate four phases of a 160 MHz clock, with one phase selected for each input to give the best timing margin. To allow low cost FPGAs to be used, i.e. those that can only produce 180° or inverted phases at 160 MHz, the CP chips are supplied with two 40 MHz clocks separated by 1.56 ns. Both of these clocks when multiplied in frequency, and having 0° and 180° phases each, will provide the 4 equally spaced clocks at 160 MHz. (At 160 MBaud a quarter of a bit period is 1.56ns).

The LVDS receivers need a clock of approximately the same frequency as the transmit clock in order to synchronise to the incoming 480 MBaud signal, from which the transmit clock is recovered. Tests have shown that the receivers function well with a clock with identical frequency to the LVDS transmitter, and so the LVDS receivers will be supplied by the fixed phase Clk40Des1 clock.

The readout links' Glink Transmitter will be driven from an onboard 40.000 MHz crystal oscillator, using a FIFO within each Readout Controller to retime the data from the TTC rate of 40.08 MHz. The TTC derived clock is not recommended for driving high-speed links as its timing jitter may transfer to the serial data, and cause errors at the receiver. The readout links can be asynchronous, so a crystal oscillator is an ideal source for a low jitter clock.

The principal timing signals used by the CPM are:

- Clk40Des1 40 MHz LHC clock skew 1 (adjustable phase)
- Clk40Des2 40 MHz LHC clock skew 2 (adjustable phase)
- L1A Level-1 Accept (initiates CPM read-out of DAQ and RoI data)
- BCntRes Bunch-Counter Reset (resets local bunch-counters).
- RstLdCnt Counter Reset of Pipeline Readout (every 128 ticks)

The first two signals are distributed as clock networks, as shown in Figure 13. Clk40Des2 is further split into three phases and distributed to just the CP FPGAs, for capturing of the FIO data. BCntRes and RstLdCnt go directly to the read-out controller(s) on the CPM. RstLdCnt is also distributed to the Serialiser and CP FPGAs

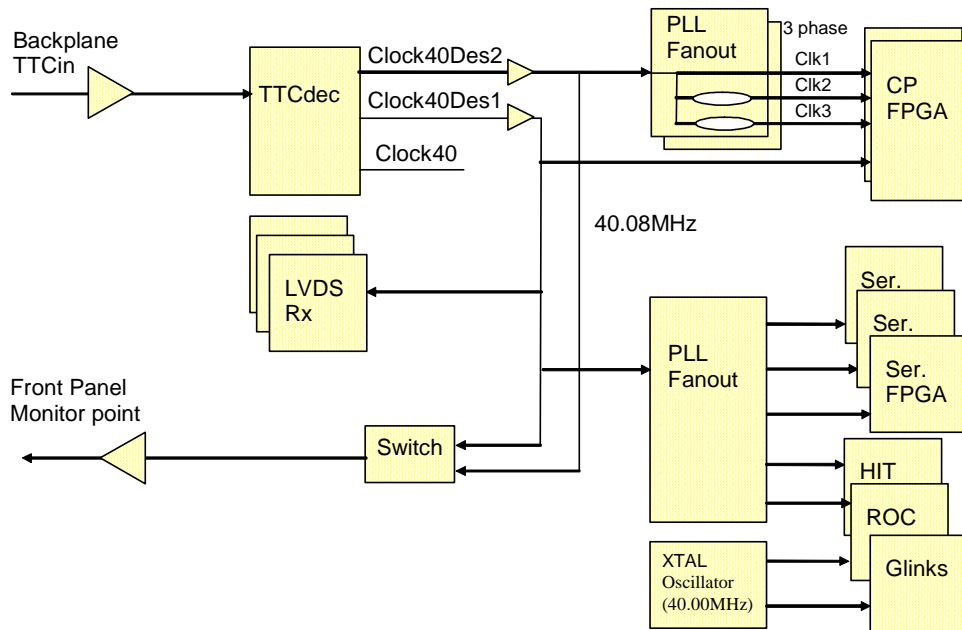


Figure 13: TTC clock signal distribution on the CPM.

The Data and Broadcast buses from the TTCrx chip are required for module control functions. The intended use of the Broadcast Commands is to start synchronous playback of Serialiser pipeline memories in test mode. These buses will go to the CPM VME controller where the issuing of synchronous commands can be integrated with the VME-based control and monitoring paths (see Section 3.6 for more details). Any commands issued to the CPM via the TTC system can then access logic on the CPM via the normal control lines to each device, but a number of devices and modules can be controlled in a synchronous fashion.

2.6.1 Latency

The whole Level-1 trigger can be viewed as a pipeline, into which are sent analogue signals, which are then digitised and processed, and from which a single signal ultimately emerges indicating that a particular source bunch-crossing contained physics data deserving further attention, i.e. a trigger decision, or just “a trigger”, has been made. There is obviously a finite and significant period of time between when an analogue signal is generated by an ATLAS detector and when the trigger decision reaches its destination. This time period is the trigger latency. The ATLAS detectors must preserve all data from a bunch-crossing in local front-end memory during this latency, so that the data can be saved for further processing if the trigger decision was positive. The known latency between when data entered the pipeline and when the decision emerged is required to retrieve the detector data to be kept from the front-end memory.

The Calorimeter Trigger forms only a section of this pipeline. It is at the boundary between the analogue and digital parts of the pipeline, and its outputs receive further processing before the trigger decision is produced by the Central Trigger Processor and distributed by the Timing Trigger and Control system. The delay of analogue signals before they reach the Calorimeter Trigger is not uniform amongst trigger towers, due to propagation times in cables of varying lengths, and hence one of the first steps within the Pre-processor after digitisation is to bring all trigger towers into synchronisation (with the analogue signal with the greatest latency) by variable length FIFO buffers. The Calorimeter Trigger then performs a complex sequence of steps through a succession of hardware devices before delivering to the CTP a set of parallel bits describing the trigger features which it identified.

A very important constraint on the Calorimeter Trigger is the latency (elapsed time) between when real-time input data is supplied to the trigger and when its output bits are made available to the CTP. The processing time of the CPM makes up a sizeable part of this latency, owing to the complexity of the algorithms that it performs. In large part, the CPM latency is made up of the processing delays within

the logic devices that the module hosts, the Serialiser and CP chip FPGAs, but there are additional time delays introduced by propagation (which are small, since the signals only travel short distances with the CP sub-system), re-synchronisation and additional logic functions performed by the module outside of the two main FPGAs, primarily result merging.

A list of the measured latency steps on the CPM is shown below in Table 1. The total CPM latency does not include the latency for result-merging by the CMM, inter-crate merging by two of the eight CMMs, or the transmission of results to the CTP. These steps were each estimated at a single tick in the TDR, which may be a slight underestimate in both cases.

Stage	Latency measurement
Cables from Pre-processor to CPM (12 m)	60 ns
LVDS de-serialisation	50 ns
Serialiser synchronisation and re-serialisation	50 ns
Transmission to CP chip (via backplane)	2 ns
CP chip synchronisation and processing	165 ns
Result-merging	12 ns
Propagation across backplane to CMM (max)	4 ns
Total	343 ns (=13.7 BC)

Table 1: CPM latency times.

2.7 Configuration and control

2.7.1 VME control

The CPM's principal means of configuration and control is a reduced functionality D16-A24 VME bus (named VME--) accessed from the CP crate backplane through the top section of the rear edge of the CPM. In order to reduce space on the backplane connection of the CPM this implements a very simple subset of the VME control signals – just $DS0^*$, $Write^*$, $DTACK^*$ and $SYSRESET$ – which obviously reduces the bus functionality to single-cycle transfers with a single bus master. The VME-- bus in each crate is controlled by a VME master with local CPU, hard disk and network connection. Each module will occupy a unique portion of the VME address space, determined by a geographical crate address, within which each device or register will be accessed via a unique range of addresses. The provisional memory map for the CPM is outlined in Section 3.10.

The bus signals pass to a VME controller logic block through which control signals and/or memory access is provided to the following devices on the CPM:

- 20 Serialisers
- 8 CP chips
- FPGA configuration memories and logic
- Result-merging logic
- Read-out controller logic
- TTCrx chip
- CAN controller
- Serial link status registers
- DAQ/RoI read-out link status registers

The VME controller must provide the following functions:

- Module and device identification.
- Module and device reset.
- $DTACK^*$ response for completed VME cycles.
- Control registers for selecting operational modes of the module and devices on the module.
- Pulsed control signals to initiate actions on the module.

- Selection masks to select a sub-set of devices of one kind for a particular operation or mode.
- Status registers summarising the condition of the module, e.g. error flags, busy signals.
- Access to registers and RAM within on-board devices, such as FPGAs and the TTCrx ASIC
- TTC command interface and TTC/VME arbitration.
- VME interface for read/write of configuration memories.

The last two functions are discussed in more detail in sections 3.6 and 3.7.2 respectively. The VME bus and VME controller must operate at speeds of ~10 MHz.

2.7.2 FPGA configuration

The CP chips, Serialisers, result-merging and read-out logic are implemented as FPGAs. The use of in-system programmable logic will allow debug and improvements to the implementation of these blocks, and even alteration of their functional specification (for algorithmic upgrade) without modification of the module hardware. It is expected that the flexibility provided by an FPGA to alter the algorithms that it implements will be utilised with changing experimental conditions; for example, a low-luminosity version of the CP chip algorithms could have slightly different functionality to that described in Section 1.

The use of FPGAs means that the hardware description of each device must be loaded after any cold restart of the module. Acceptable set-up times for the module should be of the order of a few seconds. For both of these large FPGAs there are multiple devices on each module that must be configured identically, and repeated loading of configuration data to individual devices would take some time. The default power-up process will be first to verify the configuration data, and then to simultaneously load the operational configuration for each type of device. In addition, there is a need to load alternate configurations, so that as well as performing their standard operational algorithms, the devices can be tested and the serial data path can be monitored. A quick changeover between operational and test configurations is essential, with a target specification of 1 second to re-load all configurations in parallel into FPGAs throughout the system.

It is therefore a requirement that configuration data for FPGAs be held locally on the CPM in VME-accessible non-volatile memories, from which multiple FPGAs can be programmed in parallel. The configuration data for the Serialisers and CP chips need 1 Mbit and 8 Mbit storage respectively.

The Serialiser and CP chip FPGAs are provided with more than one configuration store, each pre-loaded from VME with a different configuration, so that the FPGAs can be quickly swapped from one functional mode to another, by re-loading from the alternate memory. Although all devices of the same type (Serialisers or CP chips) will normally be configured identically, a mask register is provided to specify which FPGAs are to be loaded in parallel with a particular configuration. A second configuration pass with an orthogonal mask setting can then load the alternate set of FPGAs with a different configuration (either from the alternate memory, or from the same memory after it has itself been re-loaded). Transfer of data from configuration memories to FPGAs is a single operation initiated by the VME controller.

The time taken to reprogram the configuration memories is less than a minute. Reprogramming will be used only rarely and is not as time-critical as the configuration of the FPGAs themselves. It is a requirement that the VME interface to these memories provide FIFO buffers of at least 1 kbyte in order that fast VME data transfers will not be slowed down by slow cycle times for the transfer of each byte to configuration memory. The VME controller provides the necessary interface such that the online software can fill the configuration memories simply by writing each byte to the FIFO buffer, which indicates when its contents have been successfully transferred to a configuration memory. Separate VME interfaces are provided for the Serialiser and CP chip memories, so that the two can be configured in parallel. The configuration memories can be downloaded while a run is in progress, although the configuration data transfer via VME will have to be interleaved with the ordinary VME activity that is in progress during a run, and secondly, that configuration data can be loaded in to an FPGA while the alternate memory for that device is being written with its configuration. The configuration data is readable from both the configuration memories and the configured FPGA devices. Finally, direct configuration from VME is also possible, writing in parallel to all devices selected by their configuration mask.

The other FPGAs on the module — read-out controllers and result-merging logic — also need non-volatile storage for their configuration data, although each of these devices will only need a single stable configuration. These memories must be programmable from VME so that the configuration can be easily changed if necessary, e.g. for de-bugging. The configuration scheme is discussed further in Section 3.7.2.

Control infrastructure, such as the VME interface and FPGA configuration, must be available at power-up. The CPM has a number of CPLDs to do this, with devices distributed around the module. Programming the CPLDs of a CPM in-situ will require a PC or laptop connected via the front-panel JTAG interface.

2.7.3 DCS monitoring

The physical state of the CP system is monitored by the ATLAS Detector Control System (DCS). This will poll devices throughout ATLAS to obtain small amounts of status information in order to detect conditions such as temperature, supply voltage and current. The CPM provides sensors to measure the required conditions, and interfaces these devices to a suitable controller, which will communicate with the DCS via CANbus. The CANbus interface will use the CANbus Microcontroller circuit that has been adopted for the modules of the Level1 Calorimeter Trigger processor crates.

The CPMs and other modules within each crate are connected to a local CANbus. The crate TCM has two CANbus interfaces that allow it to act as a bridge between this local CANbus and the external DCS CANbus connection.

2.7.4 Boundary scan and verification

The connectivity between FPGAs and CPLDs on the module will be tested using boundary-scans over JTAG interfaces. The devices to be scanned will be connected in a number of JTAG chains, with the chains accessed by connections either on the front-panel of the CPM, or within the module. The chains are scanned using dedicated test equipment, attached to each CPM in turn.

2.7.5 Front panel indicators

As a basic diagnostic tool the CPM will also provide some visible (LED) indications of its state, covering power, clock, VME activity, serial link lock and module result status. Changes of state in the front-panel indicators must be “stretched” (latched for a short fixed period) so that intermittent error conditions can be observed. A front panel LEMO connector will be provided to monitor the TTCrx generated clocks used onboard, selectable by software.

2.8 Grounding

2.8.1 ESD protection

During insertion into the crate, the module must dissipate any acquired static charge before making connection to the backplane. This is to prevent static charge from damaging any device, especially the large FPGAs, that are directly wired to the backplane connector. Conductive ESD strips will be etched along the upper and lower edges of the PCB which will connect to grounded wipers on the PCB guide rails. $1\text{M}\Omega$ resistors between the ESD strips and the PCB ground plane provides a safe discharge path.

IEEE 1101.10 standard front panels will be used to provide an additional discharge path before connection. The module front panel metalwork becomes chassis ground well before the module is fully inserted.

2.8.2 Signal and Power Ground

The module ground plane is connected to the backplane signal ground using a large number of pins to provide a low impedance, short return path for the FIO signals and other board I/O. The ground plane is also the logic power supply connection via the 0V pin of the backplane power connector. The backplane power pins are wired to bus-bars. The crate 0V bus-bar is strapped to the crate metalwork to prevent the electronics from floating, and is the only low impedance connection between the module ground plane and chassis ground.

The incoming LVDS cable has its screen connected to chassis ground at the backplane. There is no direct connection between the module ground and chassis ground.

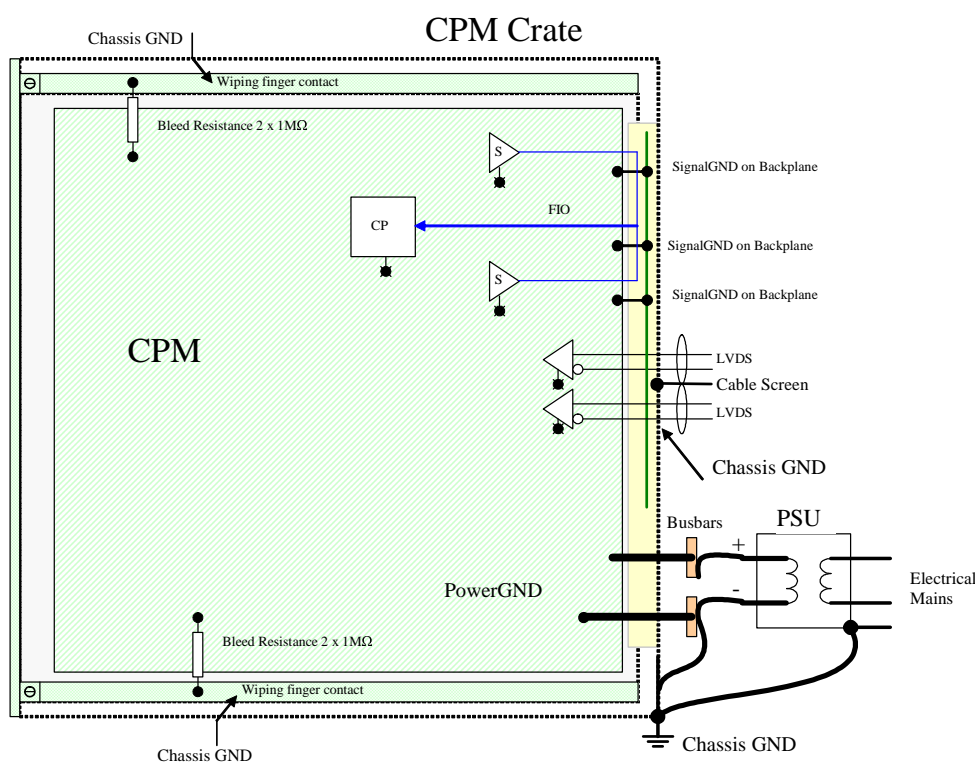


Figure 14: Grounding of the CPM.

3 Implementation

This section presents implementation details of the CPM, which show how the above functional requirements have been met. The CPM is a complex module, processing nearly 7 Gbyte/s of incoming data at a pipeline clocking frequency of 40 MHz. It handles multiple data streams at speeds of 40, 160 and 480 MBaud. Most of the processing of the module is carried out in large programmable logic devices; the CPM will host 20 Serialiser FPGAs and 8 CP chip FPGAs, plus Result-Merging and Read-out Control logic. There are also 80 serial link de-serialisers and 2 read-out links on each module.

Most of the I/O of the module is through its rear edge where the module connects to the multi-function custom CP crate backplane (which is identical to the JEP crate backplane). The input serial links are routed through the rear edge of the CPM for easy module inter-changeability (the two read-out links are routed through the front panel). This must be interleaved with the fan-in/out (FIO) data which are shared between CPMs on the backplane at 160 MBaud and the 40 MBaud result-merging transmission to the CMMs within each crate. The high-speed data and size of the module require care with clock distribution, track impedances, termination, grounding, routing and path length uniformity.

As well as handling the real-time data-flows, the module also provides access through the backplane to a VME control bus, a DCS monitoring bus (CAN) and the TTC signals distributed by the TCM. A large number of backplane ground pins are used to ensure the quality of these various speeds of signal, and to isolate signals of differing speeds from one another. Therefore a very high degree of connectivity is required for the CPM backplane, a total of 820 pins (plus power connections). This is described in more detail in Section 3.8.

The CPM is implemented as a 9U (366 mm) height multi-layer Printed Circuit Board mounted in a standard 9U crate, supported by guide rails at top and bottom, but connecting with a high-density custom backplane at the rear edge. The large insertion forces required to introduce such a high connectivity module will require leverage from the front panel. Suitable Insertion/extraction handles must be fitted to the module.

Board rigidizers / bracing bars will be used to reduce bowing of the PCB after assembly. One vertical bar near the backplane connector, and two horizontal bars near the top and bottom edges will be fitted. The crate will have 2.0mm wide guide slots, and so the edges of PCB will be profiled accordingly .

Live insertion of the CPM, i.e. with crate power on, will not be supported

The Cluster Processor system will have a total of 56 Cluster Processor Modules in four CP crates each covering one quadrant in azimuth . In addition to the modules required for the operational system, a subset of the system must be available as a test platform. There will also be a need for a number of spare modules, which must be sufficient to maintain an operational Cluster Processor for the foreseen 10-year lifetime of the ATLAS experiment. The 'spares' policy is to build 10% extra modules and purchase 25% more components of those that cannot easily be reworked, such as those in BGA packages.

The four crates will be grouped close together in the ATLAS trigger cavern, and must also be close to the 8 Pre-processor crates from which their input data will be supplied, in order to keep cable lengths, and therefore latency, to a minimum. The CTP, which receives real-time results from the calorimeter trigger, and the RODs, which receive the read-out data, will also be sited close to the Cluster Processor crates.

Nine modules, covering two minor design iterations, have been made to test backplane communications between CPMs, as well as to fully test the individual module and its upstream/ downstream interfaces. The CPM is the result of an extensive programme of concept and technology demonstration.. This has included digital simulations of FPGAs, analogue simulations of backplane transmission, and hardware demonstrators for backplane driving and serial link transmission. All of the major custom components on the CPM (Serialisers and CP chip FPGAs) or with which it communicates (Pre-processor ASIC and MCM, ROD) have been reviewed at the functional level (PDR) . A long period of test and development with other modules has evolved the design into its present state. .

3.1 Serial link inputs

The 80 serial links between the Pre-processor and the Cluster Processor and Jet/Energy-sum Processor use Low-Voltage Differential Signalling (LVDS). Each of the links to the Cluster Processor carries a 10-bit trigger tower word in every 40 MHz clock cycle, to which are added two synchronisation and framing bits, making a 480 MBaud serial stream. The incoming data are received and de-serialised back into a 10-bit parallel by an LVDS de-serialiser. The part selected is a National Semiconductor DS92LV1224. The received 10-bit word is sent on to a Serialiser FPGA for phase adjustment and distribution (see Section 3.3). The data are clocked into the Serialiser by the 40 MHz clock recovered from the 480 MBaud bit-stream (each 10-bit Serialiser input has a separate strobe). The link lock indication of each LVDS de-serialiser is sent to the Serialiser FPGAs, where the integrity of each serial link may be monitored. The link lock signals are OR-combined by the CPM in groups of 4 (by cable) and made available as 20 front panel indicators.

The CP module receives the LVDS links through the backplane feed-through pins housed in a rear-mounted guidance shroud accepting a compact connector cable of type 1370754-1 made by

AMP/TYCO. The cables are halogen-free, but connectors use PVC material. The cables are not field-repairable. The cable has a characteristic impedance of 100Ω and will be terminated by a resistor of this value adjacent to the LVDS de-serialiser input. These cables are typically 12m in length (exact length to be determined) and require equalisation at the driving end. The LVDS de-serialiser inputs will tolerate the presence of LVDS signals with the receivers powered-down.

The signals are brought from the CPM backplane connector by impedance-matched differential transmission lines to the LVDS de-serialisers, which are placed close to the rear edge of the module in order to minimise the path length for these high-speed signals. The four LVDS de-serialisers for each cable will be clustered with the corresponding Serialiser FPGA. The electrical characteristics of each LVDS input are as follows:

Termination impedance	100 Ω
Receiver Threshold	50mV peak differential
Recommended signal level	200mV peak differential
Input voltages while unpowered.	-0.3V to +3.6V

There is a requirement on the Pre-Processor to deliver LVDS signals of sufficient quality to ensure reliable links over the cable used. It has been agreed with the Pre-Processor group that the LVDS signals arriving at the CPM will meet the following criterion:

LVDS signal quality	Eye-opening 900ps by +/-100mV.
---------------------	--------------------------------

The LVDS receivers are powered from a clean 3.3 V supply, separate from the CPM's main logic supply.

3.2 Serialiser and fan-in/out

The distribution of incoming trigger tower data by the Serialiser is shown in Figure 15. The backplane links over which fan-out data is driven are about 2 cm in length between adjacent slots in the CP crate. On the destination CPM, these links are continued to connect a maximum of 3 CP chips before being terminated. This scheme ensures a direct connection between source and destination FPGAs to allow for changes in the I/O standard. Serialisers and CP chips are both positioned close to the rear edge of the CPM to minimise total path length for the 160 MBaud streams, and the backplane provides at least one grounded pin adjacent to every backplane fan-in/-out line (see Appendix A).

The Serialiser FPGA multiplexes/serialises 4 LVDS links for distribution within a CP crate. A Xilinx VirtexE FPGA, the XCV100-E will provide this function, housed in a 256-pin Fineline BGA. The prototype design used LVCMOS2 signalling between Serialiser and CP chip FPGAs, but the production module will use the SSTL2 standard, another 2.5V signalling standard. SSTL2 outputs drive lower currents and the receivers have improved noise margin due to their use of a quiet reference voltage.

The Serialiser is clocked at 40 MHz from the module clock supplied by the TTCdec card as Clk40Des1 (see Section 3.6). The Serialisers internally generate the 160MHz clock for the output 160Mbaud data. The outputs that drive onboard data include an extra delay of 3.12 ns so that the signals travelling over this path arrive at the CP Chips in time with many of those travelling further via the backplane. This scheme reduces the clock resources needed within the CP Chip.

Effort has been taken to route data lines in a prescribed manner to maximise the timing margins of the received data. Each CP chip will see similar path delays for a particular group of its input signals.

The configuration of the Serialiser FPGAs is discussed in Section 3.7.2. The Serialiser itself has a number of registers, plus the pipeline and FIFO memories, which will be VME-accessible (see Section 3.10). The CPM provides some external error counter and error-map registers to monitor the state of the serial data received by the Serialisers; these are in addition to the internal Serialiser error counters and error maps. Should a particular LVDS link be suffering from errors, its received data can be masked-out inside the Serialiser using VME accessible registers.

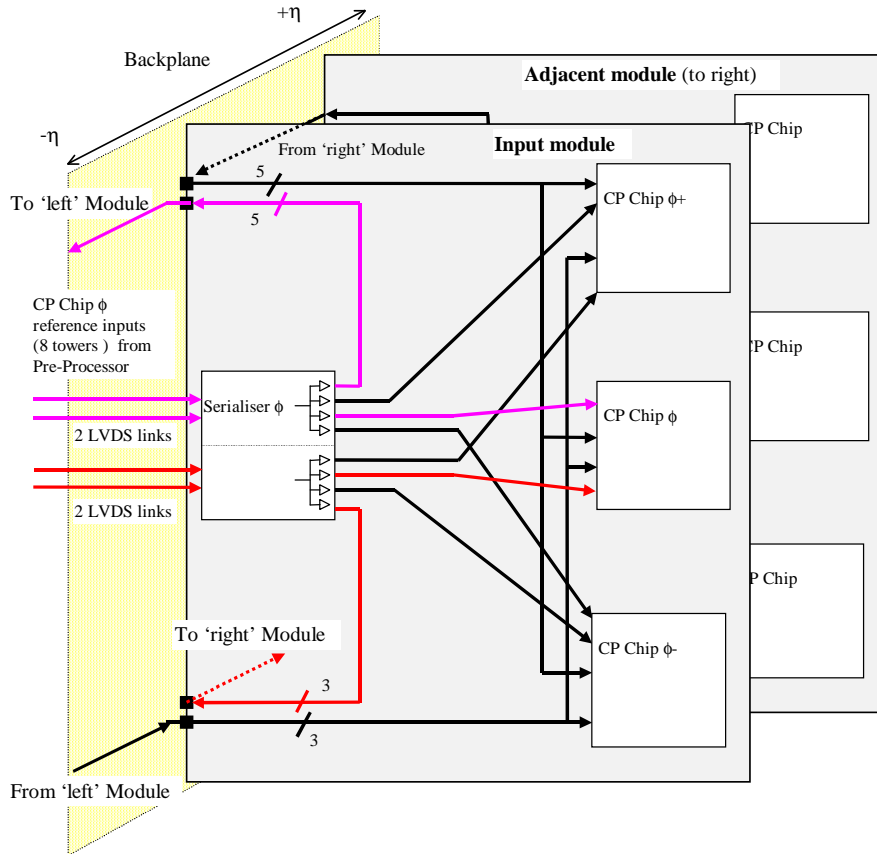


Figure 15: Distribution of 160 MBaud fan-out data on-CPM and across backplane.

3.3 CP chip and cluster processing

The principal task of the Cluster Processor Module, that of cluster-finding using the algorithms described in Section 1, is almost entirely contained within the CP chips which the CPM hosts. The CPM’s role in cluster processing is therefore to supply the CP chips with the necessary tower data (108 160 MBaud lines coming from the Serialisers on the same and adjacent CPMs — see Section 3.2) and to merge the outputs from all 8 CP chips to give module-wide hit multiplicities (see Section 3.4). The read-out of RoIs is described in Sections 2.5 and 3.5. Configuration of CP chips is described in Section 3.7.2. The CPM provides an external error-map register to monitor the state of the serial data received by the CP chips, in addition to the internal error counters and error maps of individual CP chips.

Tower data signals are labelled by the source Serialiser in a way which indicates which part of the trigger sub-space each signal corresponds to, and go to pins on a destination CP chip, which themselves are labelled so as to clearly map out the region carried by each CP chip. Figure 16 shows labelling of signals and pins for one CP chip and those of the Serialisers which supply it. CP chip pins are named by groups of 5 pins which are the destination of a 5-line bus from a Serialiser, carrying tower data describing a 2x2-tower region. There are 12 such groups of pins, labelled L, M and N from bottom to top ($-\phi$ to $+\phi$), and P, Q, R and S from left to right ($-\eta$ to $+\eta$). Data fanned in from the $-\eta$ edge of the CPM is organised as 2x1-tower pairs, carried on 3 lines rather than 5, but these pins and signals are labelled consistently.

The CPM VME controller asserts various control signals which select the CP chip mode and sequence the read-out of RoI data. The CPM provides VME access to the CP chip in order to load algorithm parameters, such as threshold values and input masks, and to read monitoring information, such as error masks and counters (described in section 2.4). The CP chip is clocked at 40 MHz by a system clock, supplied by the TTCdec card as Clk40Des2 (see Section 3.6).

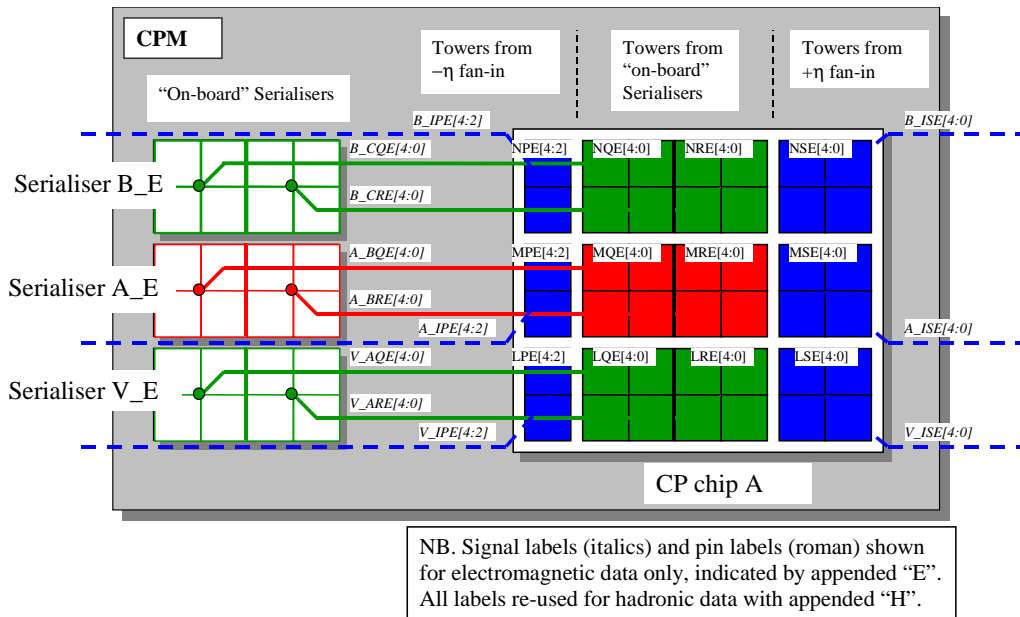


Figure 16: Labelling of CPM signals and CP chip input pins.

There are two main transmission paths between Serialiser and CP chip; a direct connection from onboard Serialisers, and a connection via the backplane from Serialisers on adjacent modules. The path difference is approximately 2 - 3ns. The CP chip is provided with two 40MHz clocks of different phases, that are multiplied up to 160MHz and used to capture the incoming data. These 'input' clocks are derived from the TTCrx Deskew2 clock, which is adjusted in time to give optimum capture of the input data. A third clock derived from the TTCrxDeskew1 clock is to be used for retiming the output data before sending to the Hit count logic and the readout dedicated to the ROI.

This design requires all data to arrive at the CP Chips reasonably close together, and with good timing margins with respect to either of the two internal 160MHz clocks. Each input pin is sampled by both clocks in parallel, and the sample with the best timing margin is selected by software from a previous calibration run. The timing of the data is mainly defined by the routing of the tracks across the board, so although the clock setting may be different for each CP chip, the calibration will be constant across all modules.

A Xilinx XCV1000E is used for the CP chip design. The PCB was designed to accept larger gate count FPGAs such as the XCV1600E or XCV2000E devices, but use of these parts is now unlikely.

The CP chip has its Vref inputs connected to receive SSTL2 levels. The number of Vref connections varies with FPGA size. Some of the pins that use Vref on the larger devices mentioned above become general I/O pins on the XCV1000E device. The firmware will configure these pins to high impedance so they do not load the Vref source. Six LP2996 devices provide a localised Vref source for the CP chips, and also a localised termination voltage Vtt for parallel terminating the FIO backplane signals. Onboard FIO signals, being series terminated at the Serialiser, do not connected to Vtt.

3.4 Result-merging

The 16 1-bit hit results (from 8 CP chips) for each threshold will be summed by two separate FPGAs, one device summing results for eight thresholds. These devices will be referred to as the hit-count FPGAs, and are implemented within a XILINX XCV100E FPGA. A VME interface to aid testing is available, but as there are no spare pins available without going for a larger package, the data bus is shared with the LED outputs. This scheme will not affect the real-time operation of the hit-counting. Calculation of the saturated 3-bit sum for each threshold will take a maximum of 12 ns, including parity bit. However, an extra 2 - 3 ns in the latency could be saved since the parity bit is not needed until the parity checksum has been calculated from the hit bits received by the CMM, and can therefore be

transmitted slightly later than the hit bits. The result-merging logic is shown in Figure 17. The Hit-count FPGA summing the lower 8 thresholds is located at the bottom of the board and feeds the CMM in slot 20, the other Hit-count FPGA summing the upper 8 thresholds is located at the top of the board and feeds into the CMM in slot3.

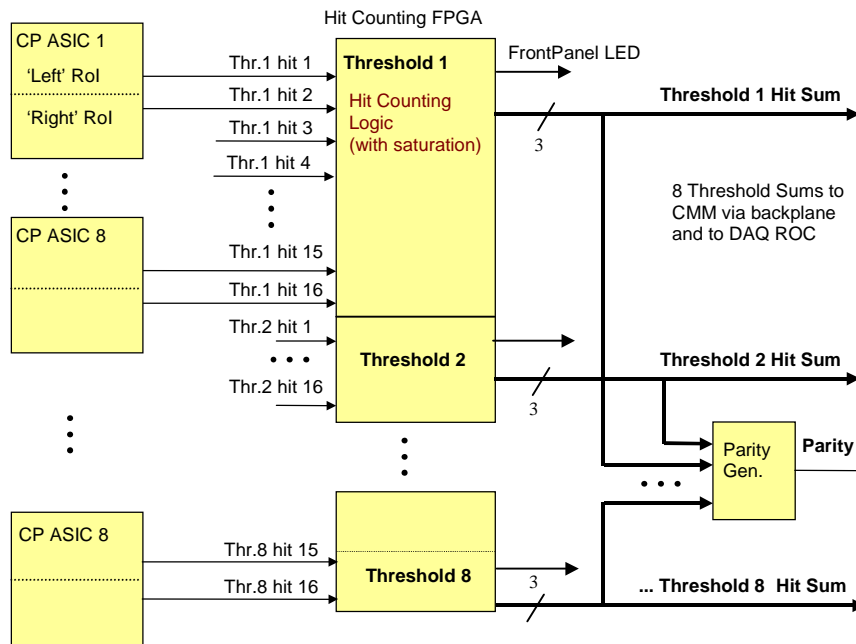


Figure 17: Schematic of (half of) result-merging (hit-counting) logic.

The 3-bit sums plus parity will be driven at 40 MBaud, in parallel off the module and over the backplane, to the CMM as series-terminated 2.5 V CMOS signals. These signals have a long path across the backplane, over 400 mm for some signals, and are buffered on exit from the CPM. The module supplies the hit-count FPGAs with a Clk40Des1 40 MHz clock.

3.5 Read-out

There are two independent but functionally similar read-out controllers on the CPM, one for DAQ read-out (predominantly from the Serialiser FPGAs, but also including the hit sums) and one for RoI read-out (from the CP chips). As stated in section 2.5, each ROC must control external pipelines and FIFOs and shift registers in the Serialiser or CP chips, whilst also replicating the pipeline, FIFO and shift register functions for additional hit and bunch-crossing number data to be appended to the data streams read out from the FPGAs under its control. The read-out controller (ROC) functions will themselves be implemented on a read-out controller FPGA. Since a part of the ROC's functions will correspond directly to the implementation within the Serialisers / CP chips, common implementation code will be used where possible, so the same Xilinx FPGA platform for the ROC as for the Serialiser FPGA and the CP chip have been used.

The read-out control logic is shown in Figure 18. Each dual-port memory pipeline within a Serialiser or CP chip has an address counter associated with the write port which is continually updating the memory with real-time data at the 40 MHz LHC BC rate. Each pipeline also has a read address counter which differs from the write counter by an offset obtained from a VME-programmable register. Both counters are reset simultaneously when the ROC sends an AddrReset signal; the read counter resets by loading in the offset relative to the write counter at x00. The pipelines are 128 locations deep and will be reset every 128 ticks. Data are moved from the pipeline to the FIFO by asserting the EnReadout input to the read-out sequencer of each Serialiser or CP chip, which then on the next clock tick writes out data from the pipeline DP-RAM address indicated by the read address counter. Multiple slices are transferred by holding the EnReadout signal in an asserted state; the read address will increment and the next slice will be transferred to the FIFO

The ROC monitors the state of the FIFOs and when it finds that they have data available (indicated by a FIFO-EmptyFlag) it uses the signal LoadShift to cause all shift registers under its control to read the next slice from the FIFO. In fact, the ROC uses the state of its own FIFO-EF line to indicate when to load all shift registers, and generates an external error flag if any of the external FIFO-EF lines are not in the same state as its own. The shift registers are continually scrolling and so begin to move out their data as soon as they are loaded; once these data have left the shift register, the shift register outputs a stream of zeroes. The FIFOs are (at least) 64 locations deep. The ROC can flush the external FIFOs by repeatedly asserting the LoadShift signal when its own FIFO is empty, inhibiting these data from being scrolled to the read-out link.

The bunch-crossing number is formed locally in each ROC by a 12-bit bunch-crossing counter, which is reset by the TTCrx signal BcntRes which is asserted once per LHC turn, i.e. once in every 3564 cycles of the 40 MHz LHC clock. This signal is issued synchronously with L1A in such a way that the local counters always read the BCN which corresponds to the L1A just received. This allows the ROD to check each CPM's BCN against the BCN from the TTC system. If there is a mis-match, it indicates that the CPM may be missing some clock cycles between BcntRes signals. The BCN within the ROC does not need to be pipelined, but goes into a FIFO on each L1A. The ROC must latch the BCN value when an L1A is received, and place it into a 12-bit BCN FIFO for the number of slices to be read out i.e. on each tick for which EnReadout is being held asserted. Note that the 12-bit BCN FIFO within each ROC is read out in different formats; it is broken into either three-bit chunks (DAQ) or single bits (RoI) and appended to four or 12 read-out streams (DAQ and RoI respectively). The ROC therefore needs multiple shift registers for each FIFO, accessing different sub-ranges of the FIFO's contents.

All external shift registers are buffered through the ROC to the read-out link. When the slice data from the external shift registers have been completely scrolled out, as indicated by an internal ROC counter set to the length of the external shift registers (see sections 2.5.1 and 2.5.2 for details of the slice formats), the ROC switches each stream to the additional hit-count or BCN data in its own shift registers. Finally, when its own data have been scrolled out, the ROC appends a parity bit calculated from all of the bits which have passed through the ROC for this slice.

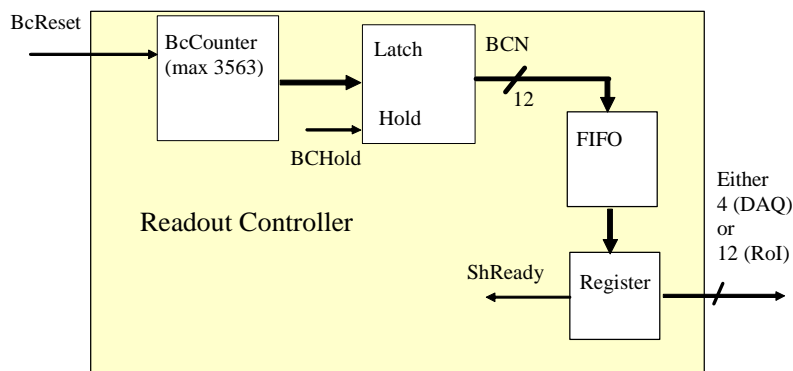


Figure 18: Bunch-crossing counter logic.

3.5.1 DAQ read-out

The DAQ ROC will initiate and control the transfer of DAQ data from all 20 Serialisers (plus pipelined hit results and bunch-crossing number) to the ROD on receipt of a TTC level-1 accept (L1A) signal — see Figure 19. A single slice of DAQ data is 84 bits in length (80 bits from the Serialiser pipelines and 4 bits appended by the ROC — see Figure 10) and up to 5 slices may be read out for each L1A, under the control of the ROC and determined by a VME-accessible register.

The DAQ ROC must also maintain an internal DP-RAM pipeline, with its own write and read counters, for pipelining hit results, to be read out at the same time as the Serialiser pipelines. These hit results are 48 bits wide and must be written to the pipeline on each bunch-crossing. On a pipeline read, the data will be transferred from the pipeline to a 48-bit wide FIFO, and then loaded into 16 3-bit shift registers.

The hit bits are derived with different latency to the input tower data for the same bunch-crossing, and so require a different read address offset to their pipeline. The 12-bit BCN FIFO transfers its contents to four 3-bit shift registers, as for the hit count data. The pipeline memory is VME accessible and is used as a spy memory on the hit output. It also provides a playback memory for testing the Readout Path.

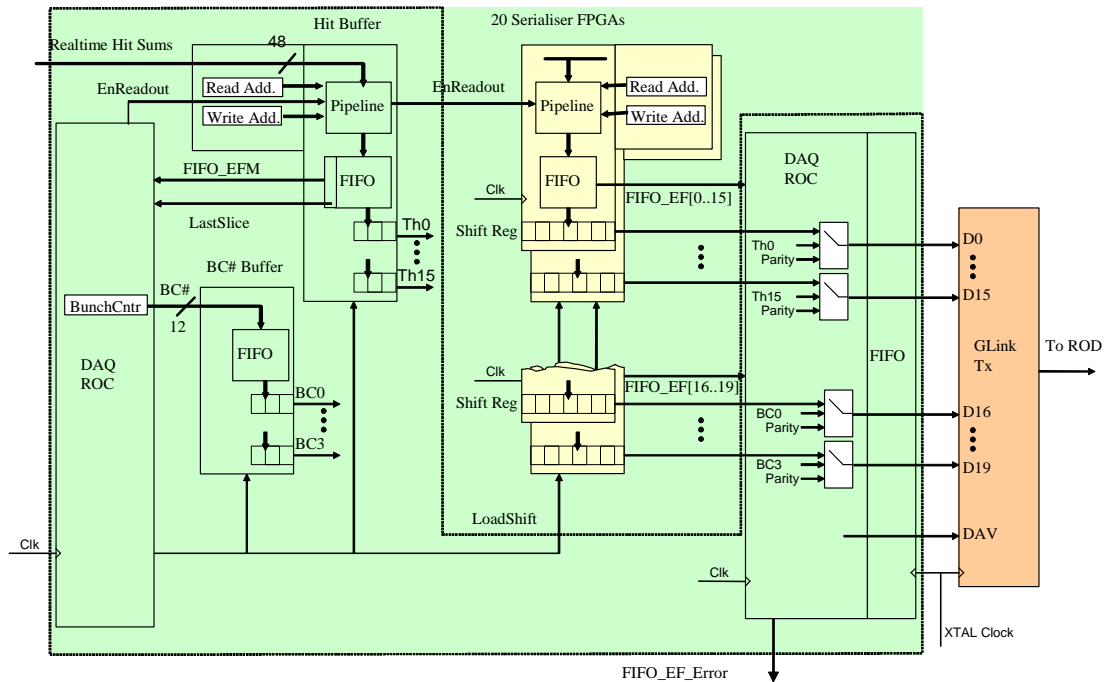


Figure 19: Schematic of Serialiser data read-out to DAQ.

The read address offsets (separate offsets for external Serialiser pipelines and internal hit count pipeline) and read-out slice count are downloaded into VME-accessible registers by the trigger online software. The registers required for the DAQ ROC are:

- Hit count read-out offset (read address counter pre-load)⁷
- Number of slices per L1A (1 minimum, 5 maximum)

3.5.2 RoI read-out

The RoI read-out is similar to the DAQ read-out, minus the pipelining of hit results. The CP chip shift register outputs are routed to 16 ROC inputs, which relays them to G-link inputs D0..D15. The RoI ROC does not contain a pipeline memory, and only requires a 12-bit bunch-crossing counter with FIFO. The shift register is a single parallel output register, since all 12 bits of the BCN are presented in parallel to inputs D0..D11 of the RoI read-out G-link. The RoI read-out uses only 16 of the 20 parallel input lines of the G-link transmitter. The remaining four fields of the RoI read-out link are not used, so the corresponding input pins (D16..D19) are grounded. The RoI read-out for each “normal” L1A consists of a single slice of data from the CP chip pipelines (plus the BCN).

3.5.3 Read-out G-Links

The read-out links are implemented using the 20-bit Agilent G-link chipset (transmitter device HDMP-1022) serialising 20-bit parallel words⁸ at 40 MHz and putting out an encoded bit-stream at 960 MBaud. These links will run continuously and carry an additional DAV* signal which is used to notify the transmitter of valid data at its inputs and recovered at the link receiver on the ROD module. The CPM must use the DAV* signal to frame valid slice data for the ROD; data received while DAV* is unasserted will be ignored by the ROD. For multi-slice read-out DAV* is maintained in the asserted

⁷ Serialiser slice read-out offset is loaded into a VME-accessible register within each Serialiser.

⁸ The ROI G-link only needs 16 bits of the 20-bit input field, but will operate in 20-bit mode so that the receiving ROD modules need only deal with one width of data from the CPM over the read-out links.

state for all slices read out, while it must be unasserted between consecutive slices from different LIAs. A minimum dead time of 3 ticks, controllable via VME, has to be kept between consecutive DAV*s to allow the ROD to recover.

The Glink transmitter is driven by an onboard crystal oscillator to provide a better quality clock than that available from the TTC system. A FIFO at the ROC output retimes the data from the 40.08MHz TTC rate, to the Glink rate of 40.00 MHz. The G-link transmitter/receiver pair are used in a uni-directional or simplex connection with the far-end G-Link receiver configured to run 'Simplex Method III'⁹ for link start-up and re-synchronisation.

Each G-link is directly connected to a fibre-optic transceiver. Standard SFP (Small Form Pluggable) transceivers are used to allow easy replacement of the laser should this fail during use. Separate transceivers are used, these being mass produced and so cheaper than a single Dual Transmitter. The receiver sections are unconnected and without power.

The G-links are powered from the CPM's 5V supply via a filter. A low profile heat-sink will be fitted to each G-Link device as each part consumes about 2 W of power.

3.6 Timing

The TTCrx decoding ASIC is contained on the TTCdec daughter card mounted on each CPM. The CPMs will use the latest version of TTCdec card with Samtec connectors and extra PLLs. The PLLs are used with external feedback so that any downstream fan-out buffering of the TTCdec 40 MHz clocks may be included within the zero-delay control loop.

The incoming differential TTC signal from the backplane is terminated with 100 Ω and then AC coupled into a 3.3V PECL buffer, before being sent on to the TTCdec card. This buffer is located close to the backplane connector. PECL voltage level swings of 700mV are expected on the input.

Clocks will be distributed on the CPM with equal delays using 'serpentine' tracks to equalise the transmission paths, and using PLL clock buffers to provide zero phase delay buffering on all clock signals, since there are at least 30 possible destinations. This is especially important for the high-speed backplane communication between Serialiser FPGA and CP Chip as the 160 MBaud receive clock is derived from the central module clock and not extracted from received data. PLL techniques prevent buffer propagation delays varying with changes in supply voltage and temperature.

Serialisers and CP Chips will be clocked from CY7B9950 PLLs. The module layout provides the CP chip with 3 phases of the TTC deskew2 clock for timing in the FIO signals (see section 3.3) plus a System Clock derived from the TTC deskew1 clock. The default phase offsets for the three CP chip deskew2 clocks are 0, +1.56ns and -0.78ns, but this may be adjusted by altering etched links on the PCB.

The PLLs chosen have been selected for low jitter from measurements of different PLL devices mounted on test boards.

⁹ This is one of the many schemes for controlling link start-up and synchronisation that are listed in the Agilent HDMP-1022/23 data sheet.

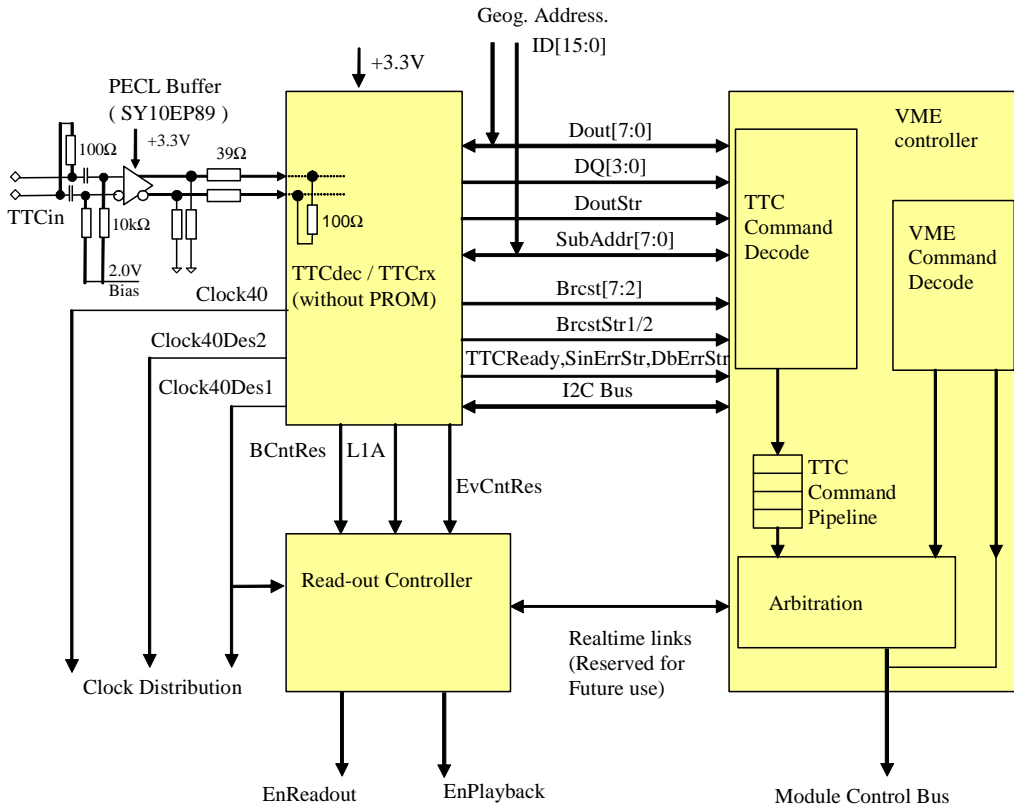


Figure 20: Schematic TTC decoder card and interfaces to CPM logic blocks.

The TTCdec card and its interfaces are shown in Figure 20. The real-time signals (L1A, BcntRes, EvCntRes) go directly to the read-out controller. The remaining TTCrx signals go to the VME controller, where they are decoded and placed in a pipeline. Any VME access that could clash with a TTC Command is arbitrated against the TTC command pipeline; certain VME operations, such as reading the Status and ID registers, do not need arbitration, and so bypass the arbitration block. If the TTC pipeline is empty, the module activity initiated by the VME access will proceed and complete within a certain time, say 50-100ns, as defined by the pipeline delay. If the TTC pipeline is not empty when the VME request arrives, then the VME access will be held-off until the TTC pipeline becomes empty. The VME interface will then access the module and acknowledge completion using VME signal DTACK.

3.7 Configuration and Control

3.7.1 VME control

The CPM VME-- control interface is a reduced D16-A24 subset of the VME64 standard, transmitted via (part of) a compact backplane connector (see Section 3.8.1). The VME-- interface signals are shown in Table 2. The VME signalling level for the production system will be +3.3V. to minimise the potential for interference with real-time signals. The VME bus signals are handled on the CPM by VME controller logic contained within a PLD and each CP crate has a networked local CPU as VME master. Each module of any particular type has a unique 6-bit geographical address within the 4-crate CP system which will define the VME base address of the module. An outline provisional memory map of the CPM is described in Section 3.10.

SYSRESET	1
A[23..1]	23
D[15..0]	16
DS*	1
Write*	1
DTACK*	1
Total Number	43

Table 2: VME signal description.

The DS* signal is edge-sensitive and requires a high-hysteresis input buffer to prevent false triggering from reflections on the backplane. The buffer is placed as close to the connector as possible to minimise the capacitive loading on the backplane.

3.7.2 FPGA Configuration

As described in Section 2.8, a rapid parallel download of configuration data to the large Serialiser and CP chip FPGAs is required. Any Serialiser FPGA can be configured from either of two areas of memory, with one area containing the operational configuration and another available for test (scratch) configurations. The CP chip FPGA configuration operates in a similar manner, with two configuration memories available. One will be the operational cluster processing configuration, and the other a serial data diagnostic configuration. The memories fitted have enough capacity to allow the use of four CP configurations should these be needed during the running of the experiment.

FLASH memory is used for holding the configuration data. This will be loaded in parallel from a sequencer within the VME controller. Each byte written to the memory is converted to a sequence of commands on the address and data buses of the memory; this sequence of commands then initiates a cycle of up to 300 μ s to store the byte. Before loading the data each memory must be cleared, taking ~20 seconds, and loading could take up to 7 seconds (both figures based on an 8 Mbit Flash RAM from STM). The Serialiser Configuration Controller will also program the configuration data for the ROC and Hit-counting FPGAs. Each configuration area may be individually erased and reprogrammed. The XILINX FPGAs will not be configured on power-up, as the intention is to first verify the contents of the memory. These FPGAs will not reject a configuration file intended for another Xilinx FPGA.

The basis of the FPGA configuration scheme is shown in Figure 21. By using local FIFO buffers of 64 kbytes in size, the transfer of data from VME to different configuration memories on one module, and on separate modules, can proceed in parallel, so that the total download time for all modules in a crate can be something like 40 seconds. However, this is only required when the configuration data for either the operational or alternate configuration of an FPGA has changed, which should be a very rare event. In ordinary circumstances, the operational configuration of either FPGA is present in memory on the module, and the FPGAs can be configured within 1 second. With either memory implementation, the data will be transferred from the memories to the FPGAs in parallel via an 8-bit data bus, with the devices to be programmed in a given pass determined by a configuration mask. A configuration source register will determine which one of the configuration memories is enabled. Configuration of a set of CP chip FPGAs will take approximately 100 ms. The configuration logic will be clocked from an on-board crystal oscillator at 12 MHz, to completely decouple the configuration process from the 40 MHz system clocks.

The DONE and INIT configuration pins on each Serialiser and CP chip have their own individual pull-up resistor as these are configured in parallel. The DONE pin is bidirectional, used for control as well as status, and a common pull-up would allow an unconfigured FPGA to prevent the others from becoming active. Only one FPGA per type, Serialiser VE and CP Chip A, have these pins monitored

during the configuration process. The remaining types , i.e. Readout Controllers and Hit Counting FPGAs are configured in series, do have their INIT pins tied together, but do not have external pull-ups on DONE or INIT.

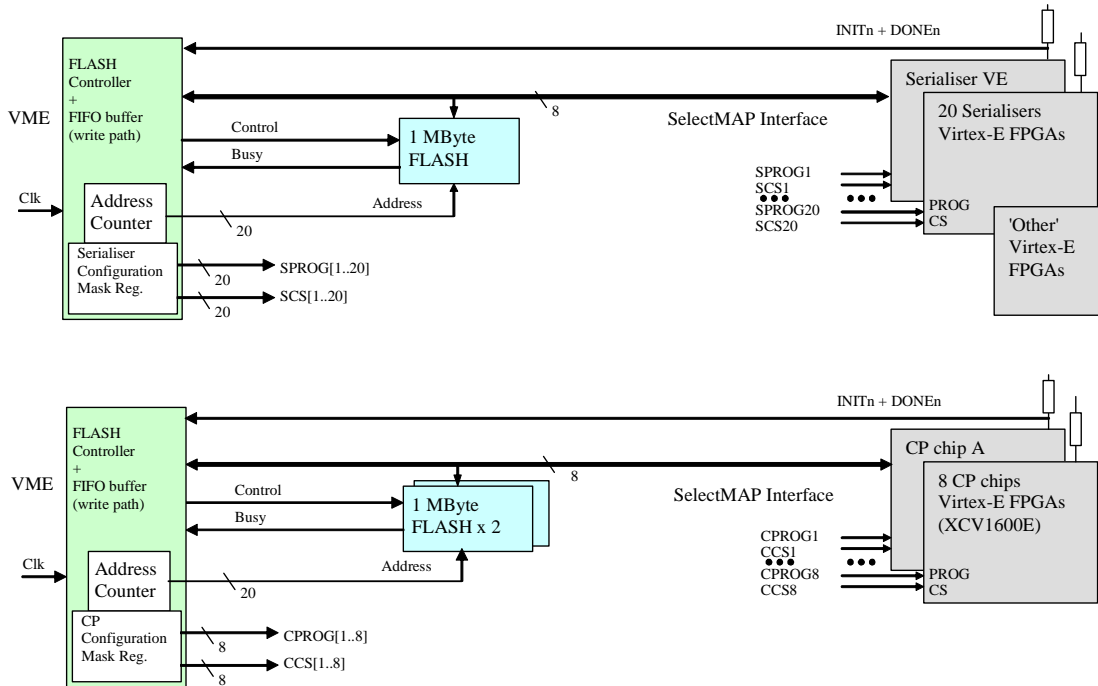


Figure 21: FPGA configuration via flash memories and VME.

The ALTERA FPGA containing the TTCrx Interface is configured from a socketed EEPROM. Connectivity is available to reprogram the EEPROM serially via VME, but firmware and software has to be written to allow this.

3.7.3 DCS monitoring

The module will provide temperature, supply voltage and current information via a local CAN bus on the backplane, to the crate CAN controller on the TCM. The onboard CAN interface will use the Fujitsu MB90F594A, which is a stand-alone micro-controller with CAN functionality. A CAN bus transceiver located next to the backplane connector connects the micro-controller to the backplane CAN bus. The Fujitsu micro-controller supports CAN bus specifications V2.0 Parts A and B.

Various link options allow the CAN controller to be reset from a VME or Module reset.

Internal FLASH-ROM memory is used for program storage. A Front Panel mounted 9 Way D-type connector is available to reprogram the internal FLASH-ROM memory using a standard RS232 cable from a PC. The programming mode is selected either by adding a link or by using a VME register. The VME option avoids the module having to be removed from the crate.

Eight analogue inputs are available to feed an internal 8-bit ADC; these analogue inputs will be used for GLink heatsink temperature sensors and supply voltage and current monitoring. The CP Chip temperature sensing diodes are monitored via the MAX1668 converter which is attached to the digital inputs of the CANuC.

Intended only for diagnostics, the CANuC is connected to the VME interface CPLD.

The use of the CAN controller to implement DCS monitoring on the CPM is illustrated in Figure 22.

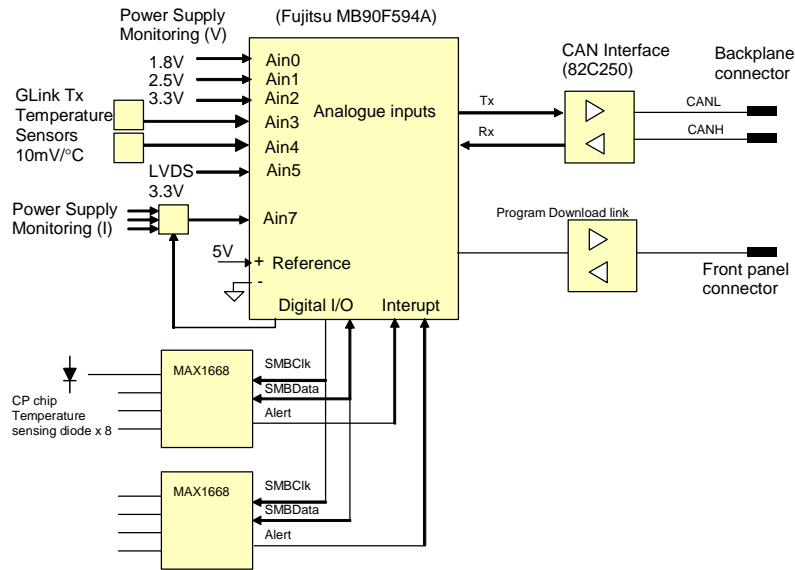


Figure 22: Schematic of DCS on CPM

3.7.4 Boundary scan and verification

The FPGAs on the CPM will be chained together via JTAG interfaces for device and module testing (see Figure 23). It is proposed to group the devices in three separate chains; the first has the ROCs, result-merging and TTCrx interface; the second has the Serialisers; the third chain has the CP FPGAs. Each of the three JTAG chains have header connectors on the module compatible with XILINX X-Checker cables to be used for JTAG testing.

Any or all of the three JTAG chains can be connected through the front panel by setting link options. Further options allow only either ALTERA or XILINX devices to be connected.

This front panel connector is compatible with the ALTERA Byte-Blaster cable for reprogramming of the ALTERA CPLDs. A XILINX ChipScope connection can also be attached via this connector to probe a running system while the module is in situ.

Non boundary-scan devices, such as the backplane connector with 320 FIO signals, and the LVDS receivers driving 800 signals, all of which terminate on BGA packages, will have to be tested using loop-back connections and LVDS test signal generators, respectively.

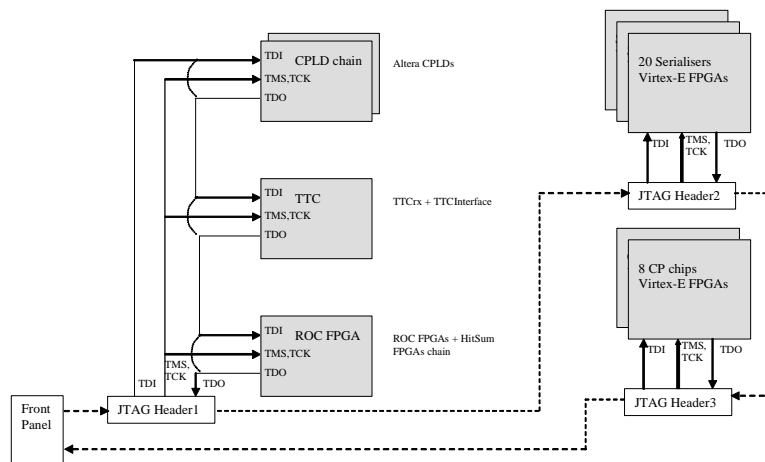


Figure 23: JTAG boundary scan chains.

3.8 Interfaces

3.8.1 Backplane

A common backplane will be used for the CP and JEP sub-systems, which have very similar requirements in terms of serial link and fan-in/-out connectivity. The backplane will be constructed as a multi-layer PCB spanning the entire width of the crate, with positions for 16 processor modules (required by the JEP, only 14 positions will be used by the CP) and additional modules (CMM, TCM and CPU) in each crate. Each processor module position must be equipped with high-density connectors covering almost the entire length of the module’s rear edge. The high pin count requires a 2 mm pitch grid-type connector. The Tyco Z-Pack connector range (IEC 1076-4-101) will be used, since a compact Halogen-free cable with high-density connector is also available. This connector style provides connections on a 2mm pitch, 5 columns wide and with a choice of connector heights. There are also integral ground shield planes built into the edges of the connectors. Both ‘upper’ and ‘lower’ shields will be fitted to the module connector.

The backplane will provide the male for all connections, the modules will mate with a female connection from the front of the backplane, and the serial link cables will mate with a female connection from the rear. Technical specifications for the Tyco connectors are shown in Table 3.

Manufacturer’s data	Tyco Z-Pack (IEC 1076-4-101)
Second Source	FCI, ERNI, ...
Pin Insertion force	0.7N (max)
Block insertion force	0.45N per pin (measured by Tyco)
Mating levels	3
Mating life / Cycles	250
Enhanced Grounding	Outside column (Z & F) connect to external shield.

Table 3: Technical specifications of the Tyco Z-pack connectors.

Backplane connectors with different mating levels will be used to reduce the maximum insertion force by staggering the engagement sequence. The Tyco connector family can provide successive engagement of different signals and power if required. The 19 row B19 connectors will be custom made, specified with signal ground pins having the longest length and the cable signal pins having the shortest length. The preferred mating sequence as shown in Table 4. Cross-sections of the CP backplane connectors are shown in Figure 24:

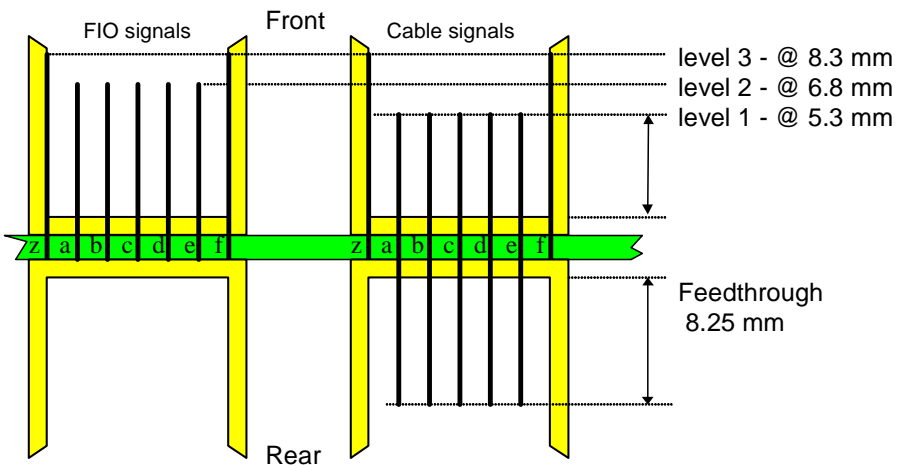


Figure 24: Illustration of AMP backplane connector with three mating levels.

Make first	Backplane Power Ground Return
	Backplane Signal Ground Return
	Backplane Power Supplies
	Backplane Signals
Make last	Cable Signals + Cable Ground Return

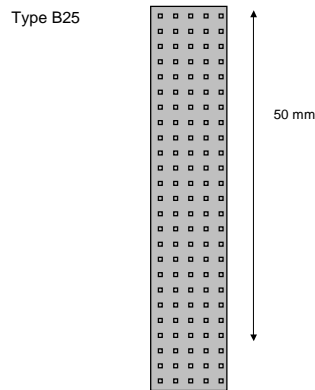
Table 4: Mating sequence of variable length backplane connections.

Guide peg



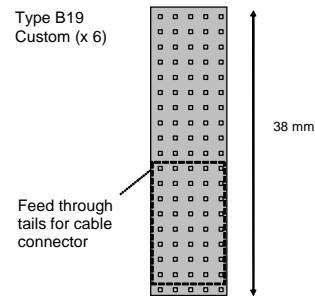
Connector 1

VME, CMM out, CPM-CPM FIO links



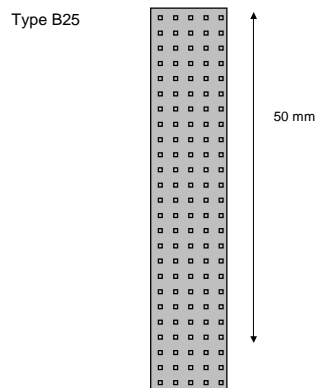
Connectors 2 – 7

LVDS input, CPM-CPM FIO links (Custom)



Connector 8

CMM out, CPM-CPM FIO links, TTC, DCS



Connector 9

Power

Total length = 361 mm

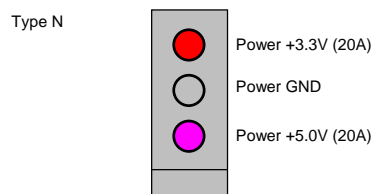


Figure 25: Layout of CPM back-plane edge connectors.

The high-density backplane layout requires two connector types, a B25 25-row connector at the top and bottom and a custom B19 19-row connector with long feed-through pins on the rear for rows 2–9 for rear-mounting 4 cables from the Pre-processor. There is a guide peg in the top position, and the power connections are made at the bottom, below the lower B25 connector. The arrangement of connector types is shown in Figure 25.

3.8.2 Front panel

3.8.2.1 Indicators

The status of the module will be indicated on the front panel by pairs of LEDs on a 0.1” pitch. Transient signals indicating VME activity, hit outputs and triggered bunch-crossing (L1A) will be “stretched” to be active for a significant fraction of a second, in order that they be visible. Indications of normal board state will be green, those of VME activity will be yellow and error conditions will be red.

Function	Colour	Quantity
Power supplies, 5V, 3.3V, 2.5V, 1.8V , LVDS & FibreTx	Green	6
VME access (stretched)	Yellow	1
TTCdec Lock & Status	Green	2
L1A (stretched)	Yellow	1
LVDS Rx Out-of-Lock indication (1 per link cable)	Red	20
Hit output (1 LED per threshold set, stretched)	Yellow	16
CP chip serial data error (1 per CP chip, stretched)	Red	8
FPGA configuration: Serialiser, CP chip and TTCinterface	Green	3
DLL Fail: Serialiser and CP chip	Red	2
GLink Tx PLL fail: DAQ and ROI	Red	2
Serialiser Synched	Green	1
CAN uC Tx and Rx	Yellow	2

Table 5: Front panel indicator LEDs.

3.8.2.2 Connectors

On the front panel there will be optical connectors for the two read-out links and LEMO monitor points for the two TTC clocks.

Function	Number	Type
Clock Monitor, 40 MHz	1	LEMO 00
RS232 for CAN uC Updates	1	9 pin D-type Socket
JTAG Access for CPLD programming and ChipScope test	1	10pin IDC
Breakout for internal signals	1	16pin IDC

Table 6: Front panel signal connections.

3.8.2.3 Handles / Injectors

Due to the very large force needed to insert a module into the backplane connector, a more robust alternative to the standard IEEE 1101.10 handle is fitted to the front panel. TrippleEase ‘Unbreakable’ style handles are used.

3.8.3 Power supplies

All of the Xilinx Virtex-E family of devices require a 1.8V supply for the core logic. From a conservative estimate of the power requirements of the Serialisers and CP chips this leads to rather high currents at this low voltage. The FPGA I/O drivers and backplane terminations will be powered from 2.5V and so require a large amount of power. The LVDS receivers will be supplied separately from a dedicated “quiet” supply at 3.3V to avoid noise problems leading to serial data errors, which have been experienced in tests. Some of the on-board logic also require 3.3V. The VME interface for the prototype is powered from 3.3V using 5V tolerant parts. The G-links transmitters are also supplied at 5V and are quite power-hungry, but since there are only two per CPM, this is not a great contribution to the overall module power consumption.

The detailed power requirements of the principal devices on the CPM are listed below in Table 7:

Device	Number per CPM	1.8 V per device	2.5 V per device	3.3 V per device	5 V per device	3.3 V (Quiet)
Serialiser	20	0.29 W	0.14 W			
CP chip	8	4.34 W		0.05 W		
LVDS receivers	80					0.19 W
G-links	2				2.50 W	
TTCdec	1			0.50 W		
CAN	1				0.50 W	
ROC	1	0.60 W	0.25 W			
Result-merging	1		0.25 W			
VME etc.	1			2.5 W		
FIO termination	160		0.02 W			
Total		41.12 W	6.50 W	3.4 W	5.50 W	15.20 W
Current		22.85 A	2.60 A	1.03 A	1.10 A	4.60 A

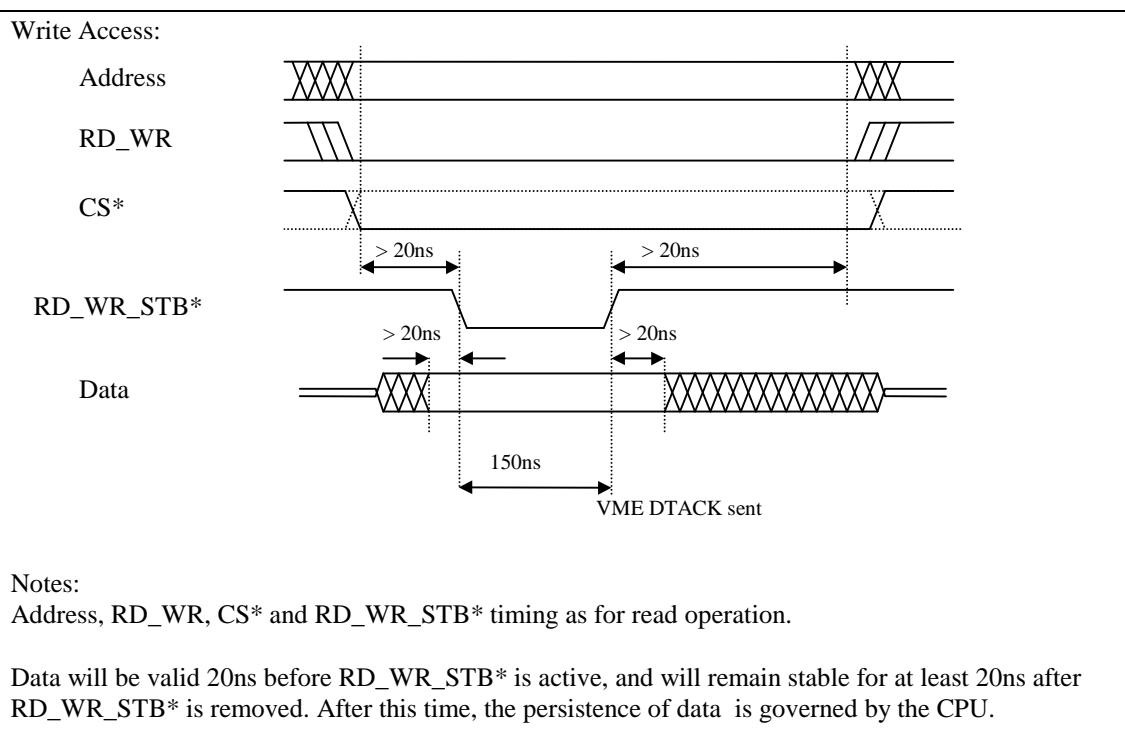
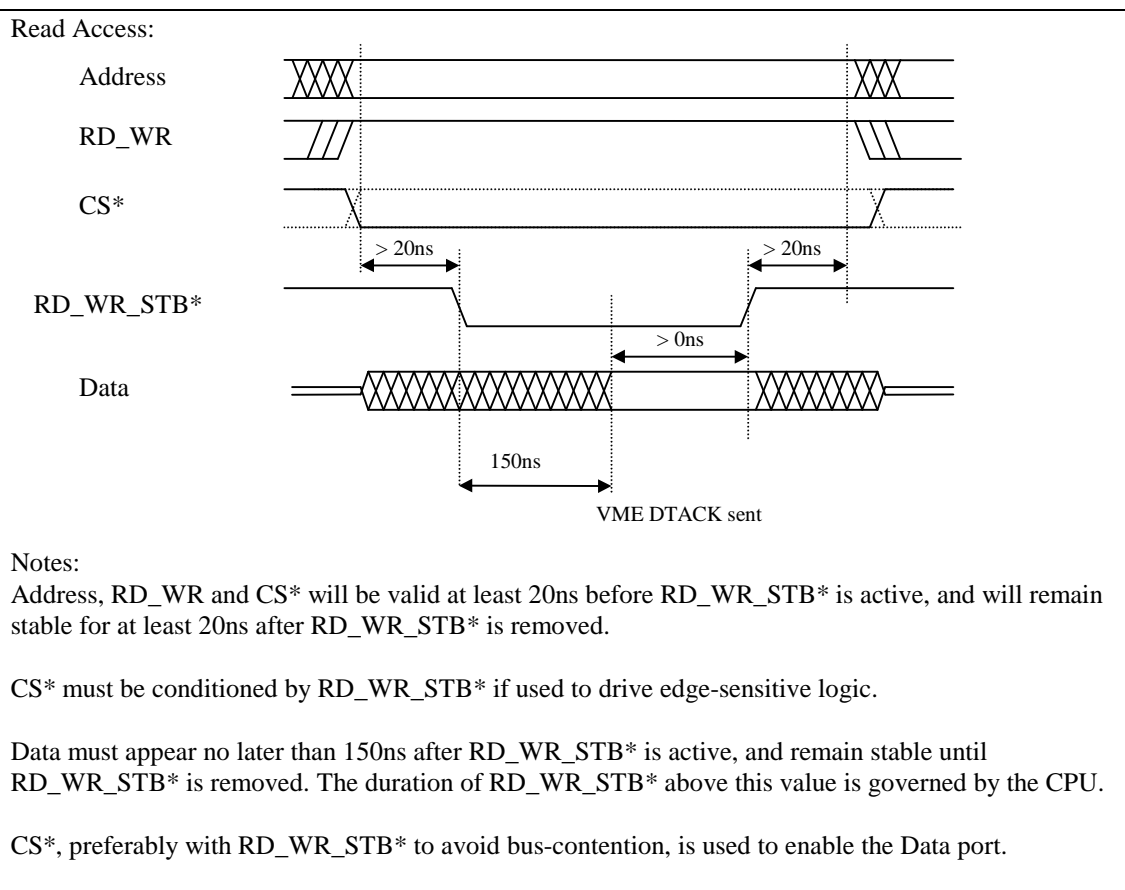
Table 7: CPM Power and current requirements.

There are only three high-current dedicated power pins available from the backplane. One of these is GND return, one will carry the 5V ‘mains’, and one will be the dedicated quiet 3.3V supply for the LVDS receivers. The AMP power connector is rated at 40A per pin, but, somewhat strangely, the right-angle connection onto the module is only rated to 20A. Three converters are used, for 1.8V, 2.5V and 3.3V. Filter chokes and Re-settable fuses will be used at the supply inputs to the module. The inputs to the onboard supply voltage converters will have additional filtering to prevent noise being generated back onto the 5V crate supply. Care has been taken to avoid over-stressing the Power Modules from excessive ringing spikes on the input supply.

The total (non-LVDS) power required from the 5V supply is ~56 W, which means a current of 11 A if assuming perfect efficiency in the conversion process, or ~70 W if assuming a conservative conversion efficiency of 80%. This is within the power connector maximum current rating, although the GND return will be approaching this limit when the current contribution from the LVDS supply and the conversion inefficiencies are taken into account. Power connections are placed at the bottom edge of the module in order not to interrupt the module hit-count signals, which run diagonally across the backplane.

3.8.4 VME interface for FPGA and CPLDs

The timing and polarity of the onboard VME signals for the Xilinx FPGAs (in particular the Serialiser and CP chip designs which were created at RAL) have been ‘formalised’ and are specified below. An ‘*’ indicates ‘active-low operation.



The VME interface to the ALTERA FPGA and CPLDs uses a similar scheme, except that CS* has been conditioned externally by RD_WR_STB*

3.9 Module layout

The CPM boards are 9U high (366 mm) and 400 mm deep. Figure 26 below shows the footprints of all major components in their approximate locations on the prototype CPM. The layout of the CPM is complex, given the number of components and the high level of I/O from the module, but by following a sensible topological layout of the major components the complications of routing and the effects of path length and module noise on the quality of the signals have been minimised.

In order to minimise and equalise path lengths for high-speed signals, incoming LVDS signals (at 480 MBaud) and backplane FIO signals (at 160 MBaud) are both distributed along most of the height of the back edge of the CPM. Lateral positioning across the depth (front-to-back) of the module will also be considered with regard to airflow for effective cooling. This results in the “chequered” pattern for the groups of Serialisers and CP chips shown in Figure 26. The LVDS receivers and Serialiser FPGAs are both sited near to the backplane connections, with the LVDS receivers mounted in groups of 4 adjacent to the corresponding Serialiser FPGA. The CP chips are located near to their associated Serialisers (each CPM receives data from 6 Serialisers on the same CPM, plus backplane fan-in), in order to keep signal traces short. Thus the majority of the real-time components handling trigger tower will be laid out in a roughly topological map of the region of trigger space processed by the module, with top to bottom of the module running from $+\phi$ to $-\phi$ and input to output devices placed from rear edge to front edge (although real-time module outputs return to the rear edge). Note that serial links and Serialisers are dedicated to either electromagnetic or hadronic tower data, and for maximum uniformity of signal path lengths, devices handling these two kinds of data should be interleaved on the module.

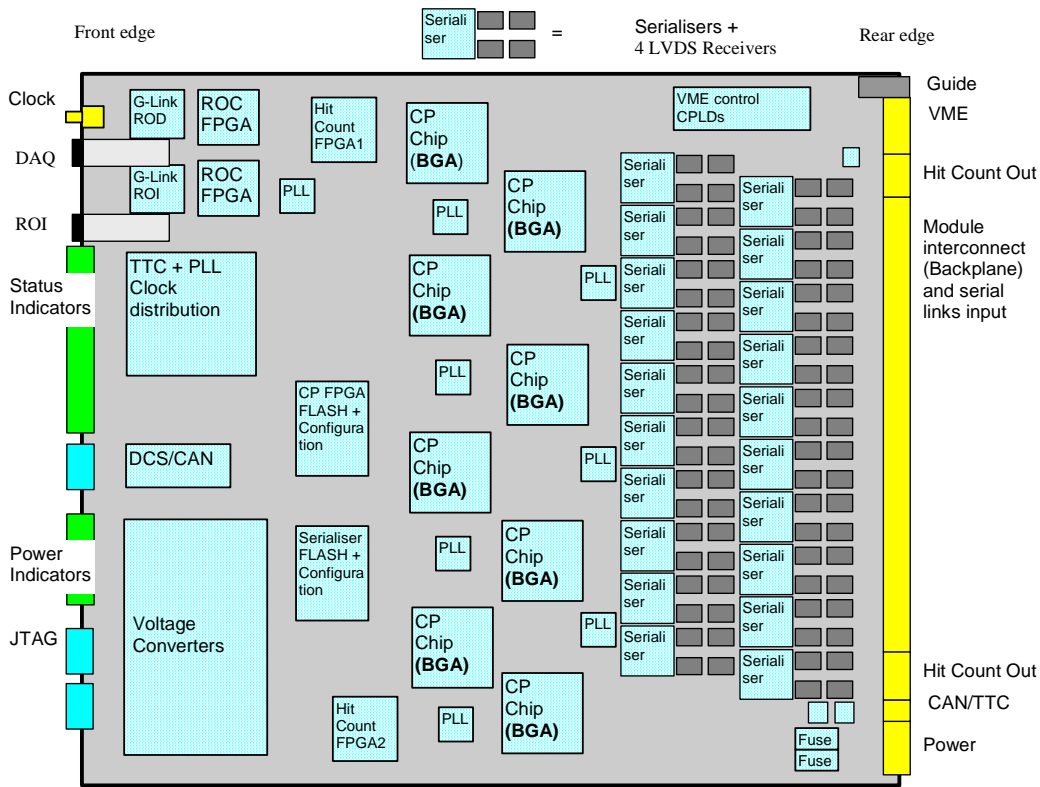


Figure 26: Approximate CPM layout and component footprint.

The hit-count FPGAs are placed at the top and bottom edges of the module, where the summed signals (two groups of 25 bits) are buffered onto the long backplane traces to the CMMs at either end of the backplane. The read-out control logic is located in the upper front corner of the module, where the two read-out links will exit the module. The VME interface is located at the top of the module, close to where the signals enter the module from the backplane. The DCS components and the TTCdec card is located close to the front edge of the module. A significant amount of space will be required for the voltage conversion from the 5 V supply, which enters at the bottom of the backplane.

3.10 Programming Model

The following outlines the programming model for VME access to registers and memories of the CPM and its components. Each CP crate will have a local CPU directly accessing the VME part of the crate backplane and thus an entirely self-contained VME address space. The details of memory locations and organization are *provisional* and very likely to change. The description set out here is indicative only and not intended to serve as a reference for the actual implementation of the CPM — suitable documentation of the programming model will be provided to accompany the prototype and final modules.

3.10.1 Guidelines

These are to aid the development of software control for the module.

- All registers can be read by the crate CPU via VME — there are no ‘write-only’ registers, with the exception of memory locations that serve as “broadcast”, i.e. for the simultaneous writing to multiple individual register locations, which may then be read back (or over-written) individually.
- The register bits generally have the same meaning for reads as for writes.
 - All Status Registers shall be read-only registers.
 - All Control Registers shall be Read/Write registers.
 - Reading back a register will generally return the last value written.
- Attempts to write to read-only registers or undefined portions of registers will result in the non-modifiable fields being left unchanged.
- It is illegal for the crate CPU to write a value which the CPM itself is able to modify at the same time.
- When the address space occupied by the CPM is accessed, it will always respond with a handshake (DTACK*) to avoid a bus error.
- The power-up condition of all registers will be all zeros, unless otherwise stated.

3.10.2 Notation

- In this model a byte is always an 8-bit field, a word is always 16 bits and a long-word is always 32 bits.
- Setting a bit-field means writing a 1 to it, clearing it means writing a 0.
- RO means that the computer can only read the value of this register; writing has no effect either to the value or the state of the module.
- RW means that the computer can affect the state of the module by writing to this register.
- WO means that the computer can write to a single memory location, which is then copied to multiple locations, which can be read back individually.

3.10.3 Memory Map

The CP module is addressed using a D16/A24 subset of the VME address bus. Each module is allocated a 512 kbyte block of contiguous address space, within the upper 8Mbyte address used exclusively by the CP modules. The module base address is defined by comparing the VME address bus A19...A22 with the four least-significant geographical address inputs. The CPM memory map is shown below, and occupies only 1/4 the allocated space. Within the CPM, functions provided with each CPLD are grouped together and placed within a 1K block. Module access is always as 16bit words from the VME bus.

3.10.4 Register Descriptions

The following provides some description and location of the registers within the CPM memory map.

Type	Function name	Size (16 bit Words)	VME Address Offset from Base (Hex)
RO	Module ID A	1	00000
RO	Module ID B	1	00002
RO	Status	1	00004
RW	Control	1	00006
RW	Pulse	1	00008
RO	Serialiser parity error register '*E'	1	0000C
RO	Serialiser parity error register '*H'	1	0000E
RO	DCS CAN uC status	1	00010
RW	DCS CAN uC control	1	00012
RO	Serialiser link loss register '*E'	1	00020
RO	Serialiser link loss register '*H'	1	00022
RO	Serialiser DLOCK status '*E'	1	00024
RO	Serialiser DLOCK status '*H'	1	00026
RO	CP chip parity error map	1	00028
RO	CP chip DLOCK status	1	0002A
RO	Glink status	1	0002C
RO	Display CPLD revision	1	0002E
RO	HitCount FPGA0 revision(thr. 0-7)	1	00030
RO	HitCount FPGA1 revision(thr. 8-15)	1	00032
RO	Serialiser Sync_Done status '*E'	1	00040
RO	Serialiser Sync_Done status '*H'	1	00042
RW	HitCount FPGA0 (thr. 0-7)	8	00080
RW	HitCount FPGA1 (thr. 8-15)	8	00090
RW	Serialiser FPGA Configuration	8	01000-
RW	CP Chip FPGA Configuration	8	01800-
RW	TTCrx Control / Status / DumpFIFO	3	02000-
RW	TTCrx I2C controller+(Reserved)	2 + (30)	02040-
RW	DAQ ROC FPGA	384 + 384 + 5	03000-
RW	RoI ROC FPGA	3	03800-
	Unused		04000 – 067FE
WO	CP chip broadcast	1K	06800-
RW	CP chip 'A'	1K	07000-
RW	CP chip 'B'-'H'	7 x 1K	07800 – 0AFFE
WO	Serialiser broadcast	2K	0B000-
RW	Serialiser 'VE'	2K	0C000-
RW	Serialiser 'VH', 'AE', 'AH' ... 'WH'	19 x 2K	0D000 – 1FFFE
	No DTACK response		20000 – 7FFFE

3.10.4.1 Module ID Register A

A 16-bit register conforming to the LVL1 Calorimeter trigger module ID convention:

Bits 0-15: Module Type, CPM = 2418

These bits are set within a PLD.

3.10.4.2 Module ID Register B

A 16-bit register conforming to the LVL1 Calorimeter trigger module ID convention:

Bits 0-7: Serial Number in the range 0-255.

Bits 8-11: PCB Module Revision No

Bits 12-15: Firmware revision No (for Register CPLD).

3.10.4.3 Status Register

A 16-bit read-only register reserved for module status information.

Bit 0: DAQ Transceiver Removed

Bit 1: DAQ Transmitter Fault

Bit 2: ROI Transceiver Removed

Bit 3: ROI Transmitter Fault.

Bits 4-15 Unused - read as Zero.

3.10.4.4 Control Register

A 16-bit read-write register containing static (non-pulsed) module controls.

Bit 0: LVDS REN – LVDS Receiver Enable. Power-up Reset to '1'

Bit 1: LVDS PWDNn – LVDS Receiver On . Power-up Reset to '1'

Bit 2: INIT_SYNC – Serialiser input synchronisation. Power-up Reset to '0'

Bit 3: FP_Monitor. Power-up to '0', selecting Deskew1 clock. '1' selects Deskew2 clock.

Bit 4: TTCDEC_TTC / XTAL. Power-up to '1', selecting TTC as clock source.

Bit 5: Glink_TTC / XTAL. Power-up to '0', selecting XTAL. A '1' selects TTCdesk1.

Bit 6: LASER Disable. Power-up to '0', enabling ROI and DAQ OpticTxs.

Bits 7-15 Unused - read as Zero.

3.10.4.5 Pulse Register

A 16-bit read-write register containing pulsed actions. Writing a '0' to individual bits has no effect.

Readback as '0'.

Bit 0: Module_Reset. FPGA Configuration , Display and TTCRX controllers

Bit 1: Reset CP_DLL.

Bit 2: Reset Global (Serialiser).

Bit 3: Reset CP_Reset.

Bit 4: Reset RST_DLL(Serialiser)..

Bit 5: Reset ROI Readout controller.

Bit 6: Reset DAQ Readout controller.

Bit 7: Reset TTCRx.

3.10.4.6 Serialiser Error Registers E & H

Two 10-bit read-only register with one bit from each Serialiser. The line is latched to 1 by the Serialiser when a parity error is detected in the corresponding device. All Error Map bits are cleared by writing the Error map clear bit in the Serialiser control register in the Serialiser broadcast space, which clears the latched input lines to the Error Map.

Address	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
E	WE	HE	GE	FE	EE	DE	CE	BE	AE	VE
H	WH	HH	GH	FH	EH	DH	CH	BH	AH	VH

3.10.4.7 DCS CAN uC Status

An 8-bit read-only register, bits 0-7, used to read data output by the on-board CAN controller.

3.10.4.8 DCS CAN uC Control

A 1-bit R/W register used in conjunction with the on-board CAN controller.

Bit 0: When set, causes the CAN uC to perform a reset.

Bit 1: When set, places the CAN uC into Program mode.

Bits 2-15 Unused - read as Zero.

3.10.4.9 Serialiser LVDS Link-Loss Status

Two 10-bit read-only registers, with one bit from each Serialiser. Each Serialiser presents an 'OR' of the link-loss count from each of its four associated LVDS receivers.

	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
E	WE	HE	GE	FE	EE	DE	CE	BE	AE	VE
H	WH	HH	GH	FH	EH	DH	CH	BH	AH	VH

3.10.4.10 Serialiser DLL Lock Status

Two 10-bit read-only registers, with one bit from each Serialiser.

	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
E	WE	HE	GE	FE	EE	DE	CE	BE	AE	VE
H	WH	HH	GH	FH	EH	DH	CH	BH	AH	VH

3.10.4.11 CP Chip Parity Error Register

An 8-bit read-only register with 1 bit for each CP Chip. The bit is set when a parity error is indicated by the Error line from the corresponding CP Chip. Details of which link(s) caused the error may be obtained by reading the error registers and counters of the CP Chip concerned. All Error Map bits are cleared by writing the Error map clear bit in the CP chip control register in the CP chip broadcast space, which clears the latched input lines to the Error Map.

			bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
			H	G	F	E	D	C	B	A

3.10.4.12 CP Chip DLL Lock Status

A read-only register, with one bit from each CP Chip.

			bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
			H	G	F	E	D	C	B	A

3.10.4.13 GLink Tx Status - PLL Lock Error.

									bit1	bit0
									ROI	DAQ

3.10.4.14 Display/Status CPLD Revision No.

A 10bit register.

3.10.4.15 HitCount FPGA revision No.

Two 5-bit registers .

These will be moved into the new HitCount area, as shown below

				bit4	bit3	bit2	bit1	bit0
Threshold 0 - 7 FPGA				Revision number				
Threshold 8 - 15 FPGA				Revision number				

3.10.4.16 Serialiser SYNC-DONE Status

Two 10-bit read-only registers, with one bit from each Serialiser.

	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
E	WE	HE	GE	FE	EE	DE	CE	BE	AE	VE
H	WH	HH	GH	FH	EH	DH	CH	BH	AH	VH

3.10.4.17 HitCount FPGAs (reserved)

A byte wide area using bits0..7. This addition will provide the HitCount FPGAs with a register area for diagnostics, such as sending test data to the CMMs. 8 locations are available.

(0) RO, Firmware Revision

8 bit revision number.

(1-7) R/W, Unspecified

3.10.4.18 Serialiser, ROC and HitSum FPGA Configuration Controller

Originally designed for just the Serialiser, the controller and associated FLASH now also provide the default configuration scheme for the ROI ROC, DAQ ROC, SMM HitSum and JMM HitSum FPGAs. Unless specified, bits <15:0> readback as '0'.

(0) RO, Firmware Version number

Bits <9:0> starting from 1. (currently 4 as of 13/6/02)

(1)(2) RW, Serialiser Reconfiguration Masks (Electromagnetic and Hadronic)

	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
(1) Elec.	WE	HE	GE	FE	EE	DE	CE	BE	AE	VE
(2)Hadr.	WH	HH	GH	FH	EH	DH	CH	BH	AH	VH

(3) RW, Control register

Bits<6:4>	Bits <3:0>	Command code.
.	0000	IDLE - Resets FLASH address pointer to zero
n	0100	BLOCK ERASE. Behaves similar to FULL ERASE.
.	0101	FULL ERASE. Clears FIFO and Resets FLASH. FLASH READY set when done
n	0110	PROGRAM. Data transferred from FIFO into FLASH memory ;
n	0111	FLASH VERIFY. Reads FLASH memory
.	1000	CONFIGURE all FPGAs from FLASH
.	1001	CONFIGURE Serialiser from FLASH BLOCK#1.
.	1100	SELECT MAP Direct Access to FPGA interface (not yet implemented)
.	1111	FIFO read / write.

Where 'n' = FLASH Block number 0 .. 5. '.' = don't care.

(4) RO, Controller Status

Bits<9:6>	Bits<5:2>	bit1	bit0
Configuration State	.	Command Busy	Command Done

(5) RO, FPGA Status

- Bit <6> ROC and HITSUM INIT signal.
- Bit <5> JMM HITSUM DONE signal.
- Bit <4> SMM HITSUM DONE signal.
- Bit <3> DAQ ROC DONE signal.
- Bit <2> ROI ROC DONE signal.
- Bit <1> Serialiser INIT signal.
- Bit <0> Serialiser DONE signal.

(6) RO, FIFO & FLASH Status

- Bit <2> FLASH READY.
- Bit <1> FIFO FULL.
- Bit <0> FIFO EMPTY.

(7) RW, Configuration Data.

Bits <7:0>: Access to FIFO, FLASH and Serialiser-FPGA SelectMap bus. Target is selected using Command Mode in Control Register. Data is programmed into FLASH via the FIFO. FIFO capacity is 64k Bytes.

Automatic configuration of FPGAs will NOT be performed on power-up. Configuration is started by writing the CONFIGURE FPGA command to the CONTROL register. This allows the software to check the contents of the flash configuration store beforehand.

The controller will then proceed to configure all associated FPGAs, in the following order; Serialiser, ROI ROC, DAQ ROC, SMM HITSUM and then JMM HITSUM. The last four devices have their INIT signals wired together and appears as one signal, the Serialisers INITs are similarly wired together.

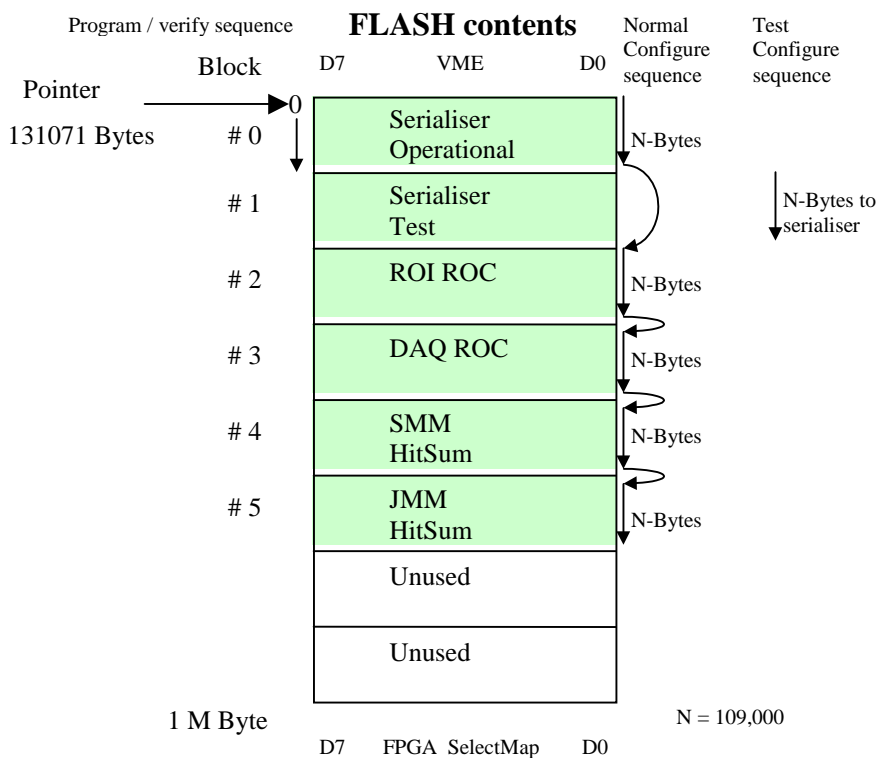
All Devices, can be reconfigured from the FLASH memory at any time by writing a new command into the control register. The Serialiser configuration data can be selected from one of two locations within the FLASH memory.

Transition between Controller Commands must go via the IDLE command .

Completion of (re)configuration will be indicated when the Done bit becomes active in the Controller Status register, and also by the Front panel LED. Configuration will be prematurely terminated if a new command is written into the control register before DONE becomes active. The VME has overriding control to avoid a 'lock-up' condition should the configuration freeze for reasons such as wrong or corrupt data in the FLASH memory.

The FLASH memory is accessed using an address counter (pointer) internal to the controller. Individual configurations start on 1Mbit boundaries. Bytes are written to the FIFO in the same order as presented to the FPGAs. Bit significance matches that of the FPGA Select-Map port, i.e. D(0) -> Select-Map(D0)

During configuration, the FLASH memory is unloaded sequentially into the FPGA being configured. Once this FPGA signals DONE of a given number of bytes have been transferred, the memory pointer is advanced to the beginning of the next block for the next configuration . Shown below is the configuration sequence following power-up / module reset.



3.10.4.19 CP FPGA Configuration Controller

On power-up, the Mask bits are set and the Default Configuration is selected
The controller is connected to only the 10 LSBs of the data bus, the unused 6 MSBs are externally pulled to zero. Unless specified, bits <15:0> readback as '0'.

(0) RO, Firmware Version number

Bits <9:0> starting from 1. (currently 3 as of 28/5/02)

(1) RW, CP Chip Reconfiguration Mask

Used for selectively reconfiguring CP FPGAs

	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
	.	.	H	G	F	E	D	C	B	A

(2) RO, Unused

(3) RW, Control register

Bits<6:4>	Bits <3:0>	Command code.
.	0000	IDLE - Resets FLASH address pointer to zero
.	0100	BLOCK ERASE. Not used for CP Chip.
n	0101	FULL ERASE. Clears FIFO and Resets FLASH. FLASH READY set when done
n	0110	PROGRAM. Data transferred from FIFO into FLASH memory ;
n	0111	FLASH VERIFY. Reads FLASH memory
.	1000	CONFIGURE FPGAs from FLASH #0
.	1001	CONFIGURE FPGAs from FLASH #1.
.	1100	SELECT MAP Direct Access to FPGA interface (not yet implemented)
.	1111	FIFO read / write.

Where 'n' = FLASH Device number 0 .. 1. '.' = don't care.

(4) RO, Controller Status

Bits<9:6>	Bits<5:2>	bit1	bit0
Configuration State	.	Command Busy	Command Done

(5) RO, FPGA Status

Bit <1> CP FPGA INIT signal.

Bit <0> CP FPGA DONE signal.

(6) RO, FIFO & FLASH Status

Bit <2> FLASH READY.

Bit <1> FIFO FULL.

Bit <0> FIFO EMPTY.

(7) RW, Configuration Data.

Bits <7:0>: Access to FIFO, FLASH and CP-FPGA SelectMap bus. Target is selected using Command Mode in Control Register. Data is programmed into FLASH via the FIFO.

FIFO capacity is 64k Bytes.

Two FLASH devices are used, each providing 1M Byte of storage for a different configuration file. Configuration for Normal running is stored in FLASH #0, with a second Configuration for test purposes stored in FLASH#1. Operation is similar to the Serialiser circuit described above.

3.10.4.20 TTCrx Control, status, dump FIFO

Provides control of the TTCrx chip. (Incomplete)

(0) RO, Firmware Revision

16 bit revision number.

(1) R/W, Control

VME control of TTCRx Command signals.

Bit <5> Reset Command
 Bit <4> Calibrate CP FPGA
 Bit <3> Calibrate Serialiser inputs
 Bit <2> Playback Serialiser
 Bit <1> Enable SCAN Register
 Bit <0> Enable TTC Commands

(2) WO, Reset Pulse

Bit <0> TTCRx JTAG reset

(3) RO, Status

Bit <2> FIFO full
 Bit <1> FIFO empty
 Bit <0> TTCrx Ready

(4) RO, Bunch_Counter

Bits <11:0> Local counter clocked from TTCRx Clock40Deskew1 signal

(5) RO, Event_Counter

Bits <15:0> Local counter clocked from TTCRx L1A signal.

(8) WO, Pulse L1A

Bit <0> 1 = pulse L1A pin high for 1 LHC clock period of 25ns. Otherwise pin is Hi-Z.

(16) RO, FIFO

Bits <7:0> TTCrx DOUT Data from 'TTCrx Dump' command
 Bits <11:8> TTCrx DQ Data from 'TTCrx Dump' command.

3.10.4.21 TTCrx I2C Controller

Provides access to the 20 user-accessible internal registers of the TTCrx chip via the I2C interface. Refer to the TTCrx Reference Manual version 3 page 13 for programming details.

(32) R/W, Control

Bit <15> Reset Controller
 Bit <13> Write / Read operation
 Bits <12:8> TTCrx Register index
 Bits <7:0> Data for TTCrx Register

(33) RO, Status

Bit <14> I2C Error
 Bit <13> I2C Busy
 Bits <7:0> Data from TTCrx Register

3.10.4.22 DAQ ROC

Provides direct mapped access to all internal registers and FIFOs of the DAQ readout controller. The pipeline is directly accessible, as well as the contents of the FIFO. The DAQ ROC FIFO is 128 locations deep by 48 bits wide, requiring 128 addressable word locations for each of bits 0-15, 16-31 and 32-47. In addition, 4 VME accessible parameter registers and 1 Control register are available:

DROC NSLICES

Number of slices to read out in response to a Level-1 accept. Normally set in the range 1 to 5, with a minimum permitted value of 1 (default). In order to read-out the Serialiser pipelines, the number of slices must be temporarily set to 128, and then re-set to the previous value.

DROC HITOFFSET

Delay in 25 ns ticks (= pipeline memory locations) from the write address to the read address for the hit slice read-out, in the range 0 to 127.

DROC MinDAVLength

Normally set to 3, but could be changed to match a new ROD design, in the range 0 to 127.

DROC BCNOFFSET

Normally set to 0, but could be changed to accommodate local test rigs, in the range 0 to 127.

DROC ControlRegister

2 bits wide:

- 01 Default mode, enable receiving data
- 10 Enable playback data

3.10.4.23 RoI ROC

Provides direct mapped access to all internal registers and memories of the RoI readout controller. The RoI ROC has no pipeline and only a 12-bit FIFO 128 locations deep, requiring 128 addressable words. In ordinary read-out mode, only a single slice of RoI data is ever required, but a register describing the number of slices is required to accommodate the read-out of external CP chip pipelines via the FIFOs. No pipeline offset register is required in the ROI ROC, since it does not have a pipeline of its own and the external CP chip pipelines have their own read address offset registers.

RROC NSLICES

Number of slices to read out in response to a Level-1 accept. Normally set to 1 (default), but in order to read-out the CP chip pipeline via the FIFO, it must be temporarily set to 128, and then re-set to the previous value.

RROC MinDAVLength

Normally set to 3, but could be changed to match a new ROD design, in the range 0 to 127.

RROC BCNOFFSET

Normally set to 0, but could be changed to accommodate local test rigs, in the range 0 to 127.

3.10.4.24 CP Chip/FPGA #1-8

Provides direct mapped access to all internal registers and memories of individual CP chips, with 1 kbyte addresses allocated per chip. These include all threshold and isolation values, timing, control and error-count registers, plus access to memories and FIFOs. Refer to the CP Chip specification for details. A ninth 1 kbyte address space is allocated as a CP Chip broadcast space. Writing to a CP chip register location within this space will broadcast the written data to all CP chips on the module.

3.10.4.25 Serialiser FPGA #1 - 20

Provides direct mapped access to all internal registers and memories of individual Serialiser FPGAs, with 2K byte addresses allocated per device. These include timing and control values, link error counters, and access to memories and FIFOs. Refer to the Serialiser specification for details. A twenty-first 2-kbyte address space is allocated as a Serialiser broadcast space. Writing to a Serialiser register location within this space will broadcast the written data to all Serialisers on the module.

4 Project Management

4.1 Overview and deliverables

The CPM is a complex multi-functional module, although much of the functionality is provided by the two types of FPGAs which it hosts — the Serialiser and the CP chip. The CPM is part of the CP sub-system and naturally must be compatible with the other components of both this sub-system and the Pre-processor sub-system from which it receives its input data. Several of these components have already been specified (and reviewed), and so this specification has tried to embody the requirements made of the CPM by those components. Changes to those specifications will naturally result in changes to the specification of the CPM.

The deliverable products of this phase of the project are:

- Specification (this document).
- 65 production CPMs (see Section 4.4).
- Configuration Software for programmable logic (including Serialisers and CP chips).
- CPM User Reference Guide.
- Design documentation (schematics, layout information, component data-sheets etc.)
- Configuration, control and monitoring software for ATLAS running..

4.2 Personnel

The CPM is the main hardware responsibility of the Birmingham group and has been designed there, although closely coupled with design activities at RAL, and to a lesser extent Heidelberg, Mainz and Stockholm. The main personnel for the project are named below, supported by several other members of the Level-1 Calorimeter Trigger Collaboration:

- Project Manager: Tony Gillman (RAL)
- Project Engineer: Richard Staley (Birmingham)
- Layout: Darren Ballard (RAL Drawing Office)

4.3 Design and verification

The design of the CPM will rest with Birmingham, although the services of other institutions will be used for specific stages, particularly the layout and manufacture of the module, which will be managed by the RAL Drawing Office. The layout of the CPM will be based on schematic design information provided by Birmingham. The electronics CAD suite Cadence will be used for the schematic design of the module at Birmingham, remotely accessing the central RAL installation of this program. Local installations of FPGA design tools will be used for the CPM logic; tools for both the popular Altera and Xilinx FPGA platforms are available. These tools include design verification and simulation capabilities.

4.4 Manufacturing

The PCB layout and routing has been done by the RAL Drawing Office, which also supervise the manufacture and assembly (population) of the module. The PCB contains 18 layers, with profiled board edges to fit standard 2mm guide rails. The size of the module and its degree of complexity make some

demanding requirements of the manufacturer. As a number of components will be large Ball Grid Arrays (BGA), an external company specialising in placement of these packages will be needed; some rework of part- or fully-assembled modules may also be required. For production and pre-production modules, companies will be sought to provide a “1-stop shop” for the module assembly, in order to avoid any conflict between the PCB manufacturer and PCB assembly company. RAL now has a ‘framework’ agreement with a number of companies who deal with the whole process, from component supply through to providing assembled modules. The agreement provides that any module failing our acceptance tests will be returned to the manufacturer and corrected at their expense.

An initial assembly run of 12 modules plus 2 pre-production V1.9 modules have been successfully tested together in a fully populated crate. A full production run will provide the remaining 53 modules.

4.5 Test

The assembled modules from the manufacture will be delivered to RAL, who will perform the acceptance tests on the modules. These tests at RAL include visual checks, module power-up and JTAG boundary scan. Any failing module will be returned to the manufacturer for re-work. JTAG testing is a very important part of the assembly procedure, checking the connectivity of complex devices particularly BGAs, and gives an immediate indication of build quality. However there are still a large number of connections and components that are unable to be tested by this method.

Good CPMs will be sent to Birmingham for configuration and more detailed checks. The modules will be hosted within a 9U crate with power supplies and backplane. Testing/commissioning of individual modules will be carried out at Birmingham to verify the most basic functions (power, connectivity, clocking and control) before more sophisticated tests can proceed. TTCvi sources will be required at this stage. Standard test equipment, such as high-speed oscilloscopes, will be required and some simple re-work capability (minor board defects, missing SM components etc.) will be provided.

Once the module is “up” and the VME control has been verified, tests of the module interfaces and functions will require one or more DSS and LSM modules to replicate the up-stream and down-stream components with which the CPM communicates. Inter-CPM communication across the backplane will need to be verified before full CPM real-time functionality can be tested.

Software will be required to drive the CPM for testing purposes. Initially, standalone testing of the CPM will be performed with diagnostic software within the HDMC framework, but more complex testing will use functions from the CP on-line control software. Both of these programs will use the same software description of the module and its configuration.

The serial link cables used for module and system testing will be identical to that being installed at CERN in USA15.

Testing of modules at Birmingham will require several people, for both hardware and software support. A detailed test specification will be required for the final production of 65 CPMs which will require a more formalised test procedure, capable of being followed by a number of test technicians.

4.6 Costs

Expected costs are given below, taken from quotes from recent tendering of the CPM manufacture, based on a production run of the remaining 53 modules.

	NRE costs	Cost each in UKP
PCB	900 (tooling + CAM/Eng)	468
Assembly	2181 (Machine set-up)	324
Components		5300

Components such as Agilent G-links which are not so new to market, or on a long lead-time, have already been purchased. We are not aware of any 'obsolete' notices being issued on any of the components used on the CPM design.

References

1. A.T. Watson, Updates to the Level-1 e/γ and τ /hadron algorithms, ATL-DAQ-2000-046.
2. ATLAS First-level trigger technical design report (TDR), CERN/LHCC/98-14.
3. <http://www.xilinx.com/apps/xapp.htm>
4. R.J.Staley, P.Bright-Thomas, A.T.Watson & N.Gee, CPM Preliminary Design Review document

Glossary

ASIC	Application Specific Integrated Circuit
Backplane	Multi-purpose high-speed backplane within CP crate
BC	One LHC bunch-crossing, occurring every 25 ns
BC-mux	BC-multiplexing scheme
BCN	Bunch-crossing number, 12-bit label in the range 0–3563
BP	Backplane
CAN	Controller Area Network, standard for communication of DCS information
CMM	Common Merger Module
Core	Central part of a region, which may contain an RoI
CP	Cluster Processor sub-system of the Calorimeter Trigger
CP chip	Cluster Processor chip, implements cluster-finding algorithms
CP crate	Electronics crate processing a quadrant of trigger space with 14 CPMs
CPM	Cluster Processor Module: module specified in this document
CTP	Central Trigger Processor
DCS	Detector Control System, monitoring of physical condition of ATLAS
DLL	Delay Locked Loop
Environment	Part of a region surrounding core towers, may not contain an RoI
FIO	Backplane fan-in/fan-out
FPGA	Field Programmable Gate Array (large programmable logic device)
G-link	Agilent Gigabit serial link
Hit	Candidate passing the criteria of a particular threshold set
Hit count	Three bit multiplicity of trigger candidates for a particular threshold set
JEM	Jet/Energy-sum Module; major module of JEP
JEP	Jet/Energy-sum Processor sub-system of the Calorimeter Trigger
LAr	Liquid Argon calorimeters (often primarily electromagnetic barrel/endcap)
Link	Single data channel between modules
LVDS	Low-Voltage Differential Signalling

MCM	Multi-chip Module
PLD	Programmable Logic Device
PLL	Phase-Locked Loop
PPM	Pre-processor module
PPr	Pre-processor sub-system of the Calorimeter Trigger
Quadrant	One quarter of trigger space in azimuth, covering entire pseudo-rapidity range
Region	An assembly of trigger towers
ROB	Read-out Buffer
ROC	Read-out Controller logic on the CPM
ROD	Read-out Driver module
RoI	Region of Interest; candidate with set of 16 hit bits and local co-ordinates
Serialiser	Device which re-serialises data from the serial links for distribution.
Slice	Data associated with a single tick
Stream	Single multiplexed 160 MBaud signal
TCM	Timing and Control Module, distributes TTC signals within crate
Threshold set	Set of cluster and isolation threshold E_T values (one of 16)
Tick	One cycle of a 40.08 MHz clock, a 24.95 ns period
TileCal	Scintillating tile (hadronic barrel) calorimeter
Trigger space	Entire set of trigger towers: $64 (\phi) \times 50 (\eta) \times 2$ (em/had)
Trigger tower	Data from an analogue sum of calorimeter cells sent to Level-1
TTC	Trigger Timing and Control
VME--	Reduced functionality VME bus, D16-A24 transfers with reduced control

Appendix A: Backplane connector layout

The CP backplane will be of a design common with the JEP, and so must take into account the modularity of both systems (the JEM has in fact slightly more LVDS cable connections and FIO pin requirements). The layout of the backplane connector position is motivated by a number of considerations:

- LVDS cable positions and FIO pins are distributed over the length of the module rear-edge, consistent with short, equal signal paths on the CPM/JEM.
- The CMM outputs are at opposite edges horizontally and opposite ends vertically of each connector position in order to have simply laid-out diagonal signal paths for the long backplane merger traces.
- The FIO blocks include a high proportion of grounds; every FIO signal pin is adjacent to at least one ground pin. There are no more than 4 FIO pins in any one row, to keep to only 4 FIO layers on the backplane.
- The LVDS signals are screened from the FIO signals by a complete row of ground pins.
- The LVDS cable screen grounds have a separate 'quiet' ground.
- The VME block is located at the top of the module for simplicity, and the DCS and TTC near the bottom in order to be able to route timing signals around the module's edge.
- Power is placed at the bottom, and a mechanical guide pin was placed at the top.

The connector ordering from top to bottom of the backplane is:

- Connector 0: Guide peg
- Connectors 1 to 8: signals
- Connector 9: power.

Each of the connector types is shown in the following map of connections as seen by the CPM. The merger outputs in connectors 1 and 8 are labelled SMM and JMM as an artefact of the requirements for the JEM, which will also use this backplane; SMM refers to the CMM used as a Sum Merger Module (with a custom configuration of the merging FPGAs on the CMM), and JMM refers to a CMM used as a Jet Merger Module (where the merging FPGAs will again be configured differently from their usage in the CP system).

Connectors 0, 1, 8 and 9 will be off-the-shelf components, but connectors 2–7 will contain a custom arrangement of pin lengths within a standard B19 shroud, in order to accommodate backplane through connections from the rear-mounted serial link cables and backplane fan-in/-out within the same connector position. The use of custom connectors is motivated by the need to distribute both LVDS signals and fan-in/-out along the length of the module's edge, rather than concentrate them in one zone of dedicated connectors. Connectors 2 and 7 differ very slightly from connectors 3–6: connector 2 contains geographical addressing pins in place of some grounds in rows 1 and 10, and connector 7 has half of its LVDS pins converted to FIO — this will require that the long through pins in this region are cropped to avoid signals pick-up from the attached LVDS cable.

Guide Pin (0-8mm) (AMP parts 223956-1, 223957-1, or equivalent)

Connector 1 (8-58mm) Type B-25 connector (short through-pins)

Pos.	A	B	C	D	E
1	SMM0	<G>	VMED00	VMED08	VMED09
2	SMM1	VMED01	VMED02	VMED10	VMED11
3	SMM2	<G>	VMED03	VMED12	VMED13
4	SMM3	VMED04	VMED05	VMED14	VMED15
5	SMM4	<G>	VMED06	VMEA23	VMEA22
6	SMM5	VMED07	<G>	VMEA21	VMEA20
7	SMM6	<G>	VMEDS0*	<G>	<G>
8	SMM7	VMEWRITE*	<G>	VMEA18	VMEA19
9	SMM8	<G>	VMEDTACK*	VMEA16	VMEA17
10	SMM9	VMEA07	VMEA06	VMEA14	VMEA15
11	SMM10	<G>	VMEA05	VMEA12	VMEA13
12	SMM11	VMEA04	VMEA03	VMEA10	VMEA11
13	SMM12	<G>	VMEA02	VMEA08	VMEA09
14	SMM13	VMERESSET*	VMEA01	<G>	<G>
15	SMM14	<G>	<G>	FL0	FR0
16	SMM15	FL1	FL2	<G>	FR1
17	SMM16	<G>	FL3	FR2	FR3
18	SMM17	FL4	FL5	<G>	FR4
19	SMM18	<G>	FL6	FR5	FR6
20	SMM19	FL7	FL8	<G>	FR7
21	SMM20	<G>	FL9	FR8	FR9
22	SMM21	FL10	FL11	<G>	FR10
23	SMM22	<G>	FL12	FR11	FR12
24	SMM23	FL13	FL14	<G>	FR13
25	SMM24	<G>	FL15	FR14	FR15

Connector 2 (58-96mm) Custom B-19 connector (mixed short/long through pins)

1	FL16	FL17	FR16	<G>	FR17
2	FL18	<G>	FL19	FR18	FR19
3	FL20	FL21	FR20	<G>	FR21
4	FL22	<G>	FL23	FR22	FR23
5	FL24	FL25	FR24	<G>	FR25
6	FL26	<G>	FL27	FR26	FR27
7	FL28	FL29	FR28	<G>	FR29
8	FL30	<G>	FL31	FR30	FR31
9	FL32	FL33	FR32	<G>	FR33
10	GEOADD5	<G>	GEOADD4	<G>	GEOADD3
11	1+	1-	<SG>	2+	2-
12	3+	3-	<SG>	4+	4-
13	1+	1-	<SG>	2+	2-
14	3+	3-	<SG>	4+	4-
15	1+	1-	<SG>	2+	2-
16	3+	3-	<SG>	4+	4-
17	1+	1-	<SG>	2+	2-
18	3+	3-	<SG>	4+	4-
19	GEOADD2	<G>	GEOADD1	<G>	GEOADD0

Connector 3 (96-134mm) Custom B-19 connector (mixed short/long through pins)

1	FL34	FL35	FR34	<G>	FR35
2	FL36	<G>	FL37	FR36	FR37
3	FL38	FL39	FR38	<G>	FR39
4	FL40	<G>	FL41	FR40	FR41
5	FL42	FL43	FR42	<G>	FR43
6	FL44	<G>	FL45	FR44	FR45
7	FL46	FL47	FR46	<G>	FR47
8	FL48	<G>	FL49	FR48	FR49
9	FL50	FL51	FR50	<G>	FR51
10	<G>	<G>	<G>	<G>	<G>
11	1+	1-	<SG>	2+	2-
12	3+	3-	<SG>	4+	4-
13	1+	1-	<SG>	2+	2-
14	3+	3-	<SG>	4+	4-
15	1+	1-	<SG>	2+	2-
16	3+	3-	<SG>	4+	4-
17	1+	1-	<SG>	2+	2-
18	3+	3-	<SG>	4+	4-
19	<G>	<G>	<G>	<G>	<G>

Connector 4 (134-172mm) Custom B-19 connector (short through pins)

1	FL52	FL53	FR52	<G>	FR53
2	FL54	<G>	FL55	FR54	FR55
3	FL56	FL57	FR56	<G>	FR57
4	FL58	<G>	FL59	FR58	FR59
5	FL60	FL61	FR60	<G>	FR61
6	FL62	<G>	FL63	FR62	FR63
7	FL64	FL65	FR64	<G>	FR65
8	FL66	<G>	FL67	FR66	FR67
9	FL68	FL69	FR68	<G>	FR69
10	<G>	<G>	<G>	<G>	<G>
11	1+	1-	<SG>	2+	2-
12	3+	3-	<SG>	4+	4-
13	1+	1-	<SG>	2+	2-
14	3+	3-	<SG>	4+	4-
15	1+	1-	<SG>	2+	2-
16	3+	3-	<SG>	4+	4-
17	1+	1-	<SG>	2+	2-
18	3+	3-	<SG>	4+	4-
19	<G>	<G>	<G>	<G>	<G>

Connector 5 (172-210mm) Custom B-19 connector (mixed short/long through pins)

1	FL70	FL71	FR70	<G>	FR71
2	FL72	<G>	FL73	FR72	FR73
3	FL74	FL75	FR74	<G>	FR75
4	FL76	<G>	FL77	FR76	FR77
5	FL78	LF79	FR78	<G>	FR79
6	FL80	<G>	FL81	FR80	FR81
7	FL82	FL83	FR82	<G>	FR83
8	FL84	<G>	FL85	FR84	FR85
9	FL86	FL87	FR86	<G>	FR87
10	<G>	<G>	<G>	<G>	<G>
11	1+	1-	<SG>	2+	2-
12	3+	3-	<SG>	4+	4-
13	1+	1-	<SG>	2+	2-
14	3+	3-	<SG>	4+	4-
15	1+	1-	<SG>	2+	2-
16	3+	3-	<SG>	4+	4-
17	1+	1-	<SG>	2+	2-
18	3+	3-	<SG>	4+	4-
19	<G>	<G>	<G>	<G>	<G>

Connector 6 (210-248mm) Custom B-19 connector (mixed short/long through pins)

1	FL88	FL89	FR88	<G>	FR89
2	FL90	<G>	FL91	FR90	FR91
3	FL92	FL93	FR92	<G>	FR93
4	FL94	<G>	FL95	FR94	FR95
5	FL96	FL97	FR96	<G>	FR97
6	FL98	<G>	FL99	FR98	FR99
7	FL100	FL101	FR100	<G>	FR100
8	FL102	<G>	FL103	FR102	FR103
9	FL104	FL105	FR104	<G>	FR105
10	<G>	<G>	<G>	<G>	<G>
11	1+	1-	<SG>	2+	2-
12	3+	3-	<SG>	4+	4-
13	1+	1-	<SG>	2+	2-
14	3+	3-	<SG>	4+	4-
15	1+	1-	<SG>	2+	2-
16	3+	3-	<SG>	4+	4-
17	1+	1-	<SG>	2+	2-
18	3+	3-	<SG>	4+	4-
19	<G>	<G>	<G>	<G>	<G>

Connector 7 (248-286mm) Custom B-19 connector (mixed short/long through pins)

1	FL06	FL107	FR106	<G>	FR107
2	FL108	<G>	FL109	FR108	FR109
3	FL110	FL111	FR110	<G>	FR111
4	FL112	<G>	FL113	FR112	FR113
5	FL114	FL115	FR114	<G>	FR115
6	FL116	<G>	FL117	FR116	FR117
7	FL118	FL119	FR118	<G>	FR119
8	FL120	<G>	FL121	FR120	FR121
9	FL122	FL123	FR122	<G>	FR123
10	<G>	<G>	<G>	FL124	FR124
11	1+	1-	<SG>	<G>	FR125
12	3+	3-	<SG>	FL125	FL126
13	1+	1-	<SG>	<G>	FR126
14	3+	3-	<SG>	FL127	FR127
15	1+	1-	<SG>	<G>	FR128
16	3+	3-	<SG>	FL128	FL129
17	1+	1-	<SG>	<G>	FR129
18	3+	3-	<SG>	FL130	FR130
19	<G>	<G>	<G>	FL131	FR131

Connector 8 (286-336mm) Type B-25 connector (short through pins)

1	FL132	FR132	FR133	<G>	JMM0
2	FL133	<G>	FL134	FR134	JMM1
3	FL135	FR135	FR136	<G>	JMM2
4	FL136	<G>	FL137	FR137	JMM3
5	FL138	FR138	FR139	<G>	JMM4
6	FL139	<G>	FL140	FR140	JMM5
7	FL141	FR141	FR142	<G>	JMM6
8	FL142	<G>	FL143	FR143	JMM7
9	FL144	FR144	FR145	<G>	JMM8
10	FL145	<G>	FL146	FR146	JMM9
11	FL147	FR147	FR148	<G>	JMM10
12	FL148	<G>	FL149	FR149	JMM11
13	FL150	FR150	FR151	<G>	JMM12
14	FL151	<G>	FL152	FR152	JMM13
15	FL153	FR153	FR154	<G>	JMM14
16	FL154	<G>	FL155	FR155	JMM15
17	FL156	FR156	FR157	<G>	JMM16
18	FL157	<G>	FL158	FR158	JMM17
19	FL159	FR159	FR160	<G>	JMM18
20	FL160	<G>	FL161	FR161	JMM19
21	FL162	FR162	FR163	<G>	JMM20
22	FL163	<G>	FL164	FR164	JMM21
23	<G>	<G>	<G>	<G>	JMM22
24	CAN+	<G>	TTC+	<G>	JMM23
25	CAN-	<G>	TTC-	<G>	JMM24

Connector 9 (336-361mm) Type D (N) connector

2	+3.3V
6	Power GND
10	+5.0V

Appendix B: Summary of BC-mux logic

For the CPM system¹⁰, the pre-processor combines two trigger towers in a scheme known as BC-muxing. This is a form of compression which relies on the fact that for each trigger-tower, both adjacent timeslices will be zero. The BC-mux logic takes 2 consecutive pairs of tower data bytes and packs them, with parity protection, into two consecutive 10 bit words. The BC-mux scheme for encoding two towers, A and B, is described as follows:

The first non-zero tower out of the pair is sent first, with a flag bit indicating which tower. If both are non-zero then tower A is sent first. On the following BC, the other tower is sent, with the flag bit indicating whether it was from the same BC as the first, or from the BC following. Therefore the meaning of the BC-mux bit changes between consecutive non-zero BCs.

The table below (derived from Table 6-1 of the TDR) summarises the use of the BC-mux bit:

Case	Inputs				Outputs	
	A(i)	A(i+1)	B(i)	B(i+1)	V(i)	V(i+1)
1	0	0	0	0	0,0	0,0
2	X	0	0	0	X,0	0,0
3	0	0	Y	0	Y,1	0,0
6	X	0	Y	0	X,0	Y,0
7	X	0	0	Y	X,0	Y,1
8	0	X	Y	0	Y,1	X,1

The input columns show the data within timeslices (i) and (i+1) for trigger towers A and B. The output columns show when the data is output and the value of the BC-mux bit. This scheme is self-synchronising in that the consecutive pairs can be from any BC, odd or even, and any subsequent transmitted zero resets the sequence. The above table is written assuming that both A and B have been zero on the previous BC, or that either A or B have had non-zero data for an even number of BCs.

¹⁰ The JEP system receives data from the Pre-processor using a different coding scheme.