# CMX Base Function FPGA Firmware functionality

Wojciech Tadeusz Fedorko
University of British Columbia (CA)
Pawel Piotr Plucinski
Stockholm University (SE)
Michigan State University (USA)

*version 1.1*

*CERN 7 September 2016*

# Contents

# 1 General Description

The CMX was designed (see Figure 1) to fulfill the following criteria [3]:

- perform all tasks previously handled by any CMM and to conduct these tasks at higher input and output line rates
- provides more computing power to support additional algorithms
- adds new functionality to send out processed copies of its inputs over optical fibres
- provides optical functionality to perform Topological Processing on CMX data

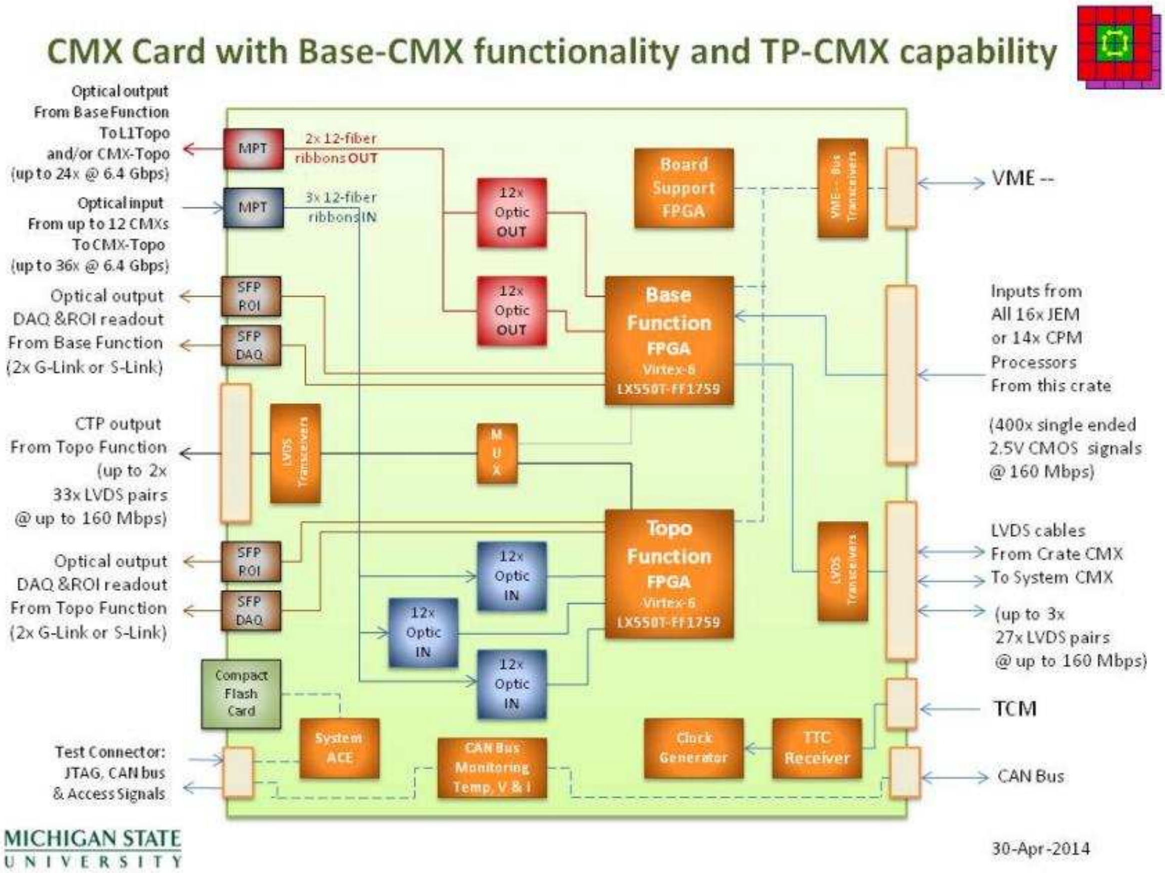

CMX Card with Base-CMX functionality and TP-CMX capability
Optical output From Base Function To L1Topo and/or CMX-Topo (up to 24x @ 6.4 Gbps)
MPT
2x 12-fiber ribbons OUT
Optical input From up to 12 CMXs To CMX-Topo (up to 36x @ 6.4 Gbps)
MPT
3x 12-fiber ribbons IN
12x Optic OUT
12x Optic OUT
Board Support FPGA
VME -- Bus Transceivers
VME --
Optical output DAQ &ROI readout From Base Function (2x G-Link or S-Link)
SFP ROI
SFP DAQ
Base Function FPGA Virtex-6 LX550T-FF1759
Inputs from All 16x JEM or 14x CPM Processors From this crate
(400x single ended 2.5V CMOS signals @ 160 Mbps)
CTP output From Topo Function (up to 2x 33x LVDS pairs @ up to 160 Mbps)
LVDS Transceivers
M U X
Topo Function FPGA Virtex-6 LX550T-FF1759
LVDS Transceivers
LVDS cables From Crate CMX To System CMX
(up to 3x 27x LVDS pairs @ up to 160 Mbps)
Optical output DAQ &ROI readout From Topo Function (2x G-Link or S-Link)
SFP ROI
SFP DAQ
12x Optic IN
12x Optic IN
12x Optic IN
Compact Flash Card
TCM
System ACE
CAN Bus Monitoring Temp, V & I
Clock Generator
TTC Receiver
CAN Bus
Test Connector: JTAG, CAN bus & Access Signals
MICHIGAN STATE UNIVERSITY
30-Apr-2014

Figure 1: CMX block diagram with TP FPGA installed.

## 1.1 CMM emulation

In order to become a replacement for the CMM card, the CMX module is able to operate in the CMM slots of the L1calo crates. It obeys the same backplane pinout for all its backplane VME power, signal, control and monitoring pins. Signal names were carried over from the CMM to the CMX to avoid confusion. The CMX also provides the same LVDS connectors as CMM for sending its results to the CTP over the existing cable plant. The CMX is able to provide the same backplane Merger Cable IO capabilities as the CMM and is able to operate as a Crate CMX or System CMX. The set of CMX cards installed in L1calo is thus able to use the existing RTM modules and Crate CMX to System CMX cable plant. Similarly to the CMM the CMX provides G-link ports for optical DAQ and ROI outputs to the existing RODs over the existing fibers. The CMX uses System ACE for configuring the Virtex-6 FPGA firmware. In addition, the CMX supports CAN Bus monitoring. Temperatures from several locations on the board and voltages from the on-board power supplies are being monitored on the CMX as on the CMM. In addition to the power supply voltages the CMX monitors their currents. Because of its additional features, the CMX is not able to present the exact same set of VME control registers as the CMM, and the online software is modified and extended to control and monitor the CMX cards.

## 1.2 Increased Bandwidth

The CMX is able to run at the speed of the CMM card, if desired, but also supports a new mode of operation with higher line rates for:

- backplane signals it receives from the JEM or CPM modules
- cable IO connecting a Crate CMX to its corresponding System CMX
- CTP ouput

### 1.2.1 Inputs from JEM or CPM modules

Over the backplane the CMX receives inputs from up to 16 JEM or 14 CPM processor modules present in the same crate. 25 signals are received from each processor module for a total of 400 (16 x 25) backplane input signals. The CMM mode of operation is based on receiving one bit of information on each backplane input line for each beam crossing, i.e.one bit every 25 ns or 40 Mbps per line. In this mode one bit out of the 25 bits received from each source processor module dedicated to parity. The CMX is able to receive 4 bits of information on every backplane line for every beam crossing, i.e. one bit every 6.25 ns or 160 Mbps. The bit that was previously used for parity is now dedicated to carry a clock signal, alternating between low and high every 6.25 ns, which is the 80 MHz forwarded clock sent by the processor module. The characteristic impedance of the 400 processor input lines is 60 Ohms. The CMX tries to maintain a 60 Ohm characteristic impedance over the whole path from the backplane pins to the FPGA input pins.

### 1.2.2 Crate CMX to System CMX Cable IO

The CMX card is able to send or receive parallel LVDS data to or from other CMX cards. The direction of data flow depends on whether the CMX card is used as a Crate CMX or a System CMX. Up to three LVDS cables can be connected to a CMX card via a Rear Transition Module (RTM) plugged in the back of the crate that route the cable LVDS signals to the backplane pins. On the CMM card three sets of 27 LVDS signals are operated together as inputs or as outputs. The CMM operation is based on sending or receiving one bit of information on each Cable IO Common Merger eXtended (CMX) 17 LVDS line for each beam crossing, i.e. one bit every 25 ns or 40 Mbps, with one bit out of the 27 bits from each cable being dedicated to parity and one bit being reserved. The CMX is able to send or receive up to 4 bits of information on every LVDS line for every beam crossing, i.e. one bit every 6.2 5 ns or 160 Mbps. A bit from each cable which is currently reserved could be used as a clock signal if necessary. The existing RTMs and cable plant are able to support operation at 80Mbps. New RTM modules might be required if operation at 160Mpbs becomes desirable. The direction of the LVDS transceivers used for each IO Cable is controllable from the Base Function FPGA, and each cable can be controlled independently to operate as input or output.

### 1.2.3 Output to CTP

The CMX card is able to send parallel LVDS data to the CTP system. Only the System CMX cards send trigger information to the CTP. Up to two CTP LVDS cables can be connected to a CMX card via its two front panel connectors. On the CMM card two sets of 33 LVDS signals are used as outputs. The CMM operation is based on sending one bit of information on each CTP Output LVDS line for each beam crossing, i.e. one bit every 25 ns or 40 Mbps, with one bit out of the 33 bits in each cable being dedicated to parity. The parity bit (odd) is calculated based on trigger information to the CTP (see format of cable data from specific CMX type to the CTP).

The CMX is able to send up to 4 bits of information on every LVDS line for every beam crossing, i.e. one bit every 6.25 ns or 160 Mbps. The current usage plan is to operate the CTP Outputs at 40 Mbps. The direction of the LVDS transceivers used for each CTP Output connector is controllable from the BF FPGA (or the TP FPGA when present), and each cable can be controlled independently to operate as input or output. The CMM provides boundary scan for this port, but CMX test firmware and loopback cables can be used to achieve a similar test feature.

## 1.3 Increased Processing Power

One motivation for redesigning the CMM module using newer FPGA technology is that more logic blocks and thus more processing power is available on the CMX than on CMM by a factor of between 1 and 2 orders of magnitude, depending on what is being considered.

## 1.4 Added Functionality

In addition to reproducing and extending the functionality existing on the CMM card, the CMX card provides new functionality. The main motivation for replacing the CMM cards was to send the backplane input information that is received on the CMX card to an external L1 Topological Processor system (L1topo). A given CMX card only sees information local to its crate and only information of one type (electron, tau, or jets objects, or energy). The key characteristic of a L1topo system is in concentrating the information from all 12 CMX cards (plus other sources as they become available). Using more complex algorithms L1topo is able to compare and combine information including the full geographic coverage (e.g. for invariant masses or angle of separation) and combining multiple types of information (e.g. electrons and jets). Originally requested as a backup plan and in case a dedicated L1topo system would not be built or would not become available on time, some Topological Processing ability was proposed for the CMX card (CMX-Topo). Topological Processing on the CMX platform is no longer likely to be used because a dedicated L1topo has been designed and built. The functionality required for supporting a reduced L1topo system on the CMX platform has still been included for future undefined optional usage. This feature provides some attractive flexibility for interconnecting CMX cards and concentrating in one place information from geographically separate sources. It could for example be viewed as an alternate and higher bandwidth method of connecting Crate CMX cards to their System CMX card.

## 1.5 Real-time Data Path

There are two separate usages for the CMX cards in the L1calo system. Some CMX cards act as Crate CMX cards and others as System CMX cards. Consequently there are two separate patterns for the real-time data paths possible in the operation of the Base Function tasks performed by the CMX. There is a total of twelve CMX cards in the full L1calo system interconnected as 4 groups, with each group handling the trigger information concerning a particular object type: electron, tau, Jet or Energy.

### 1.5.1 Crate CMX

A CMX acting as a Crate CMX receives the 400 backplane inputs coming from the 16 processor modules slots in that crate and computes local counts of objects before sending that information out through the backplane over one or two LVDS cables to a System CMX (see Figure 2). A Crate CMX also sends this local trigger information out optically to L1topo. Information going to the L1topo is serialized by the Base Function FPGA which drives two Avago MiniPOD optical transmitters.

### 1.5.2 System CMX

A CMX acting as a System CMX receives the 400 backplane inputs coming from the processor modules in the crate and sends this local trigger information out optically to L1topo exactly like a Crate CMX does (see Figure 2). A System CMX computes its counts of local objects as well but does not send that information out. Rather it merges its own count information with the count information it receives through the backplane over 2 or 3 LVDS cables coming from all the Crate CMXs handling the same type of information. A System CMX forms the final trigger information for the object type it handles and sends it to the CTP over 1 or 2 LVDS cables attached to the front panel. All twelve CMX cards in the full L1calo system send optical information to the external L1Topo.

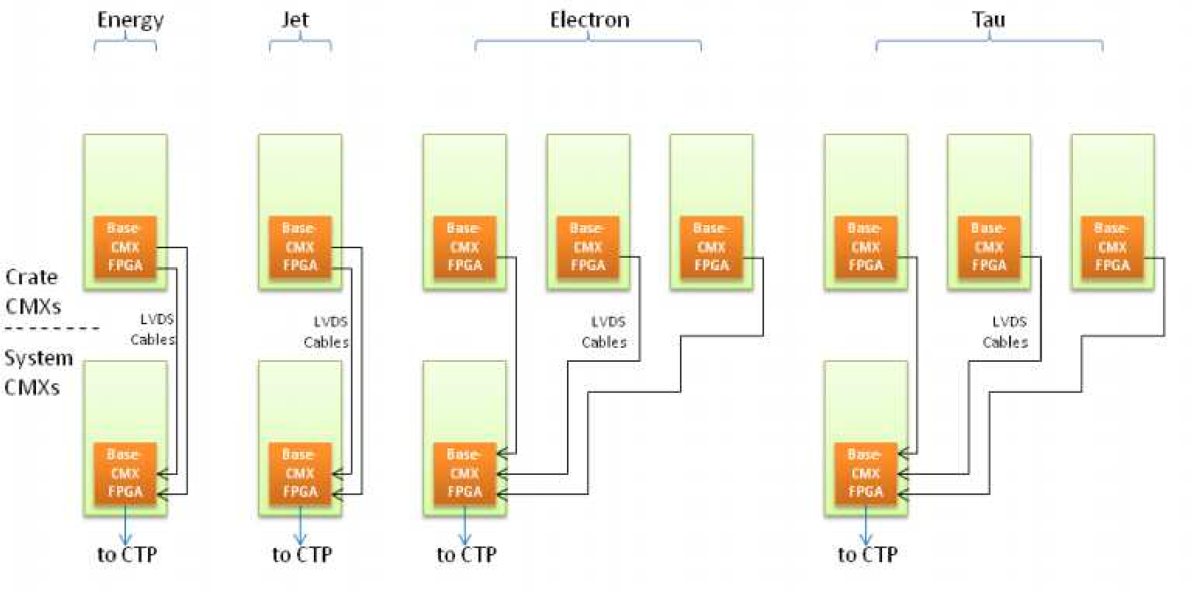
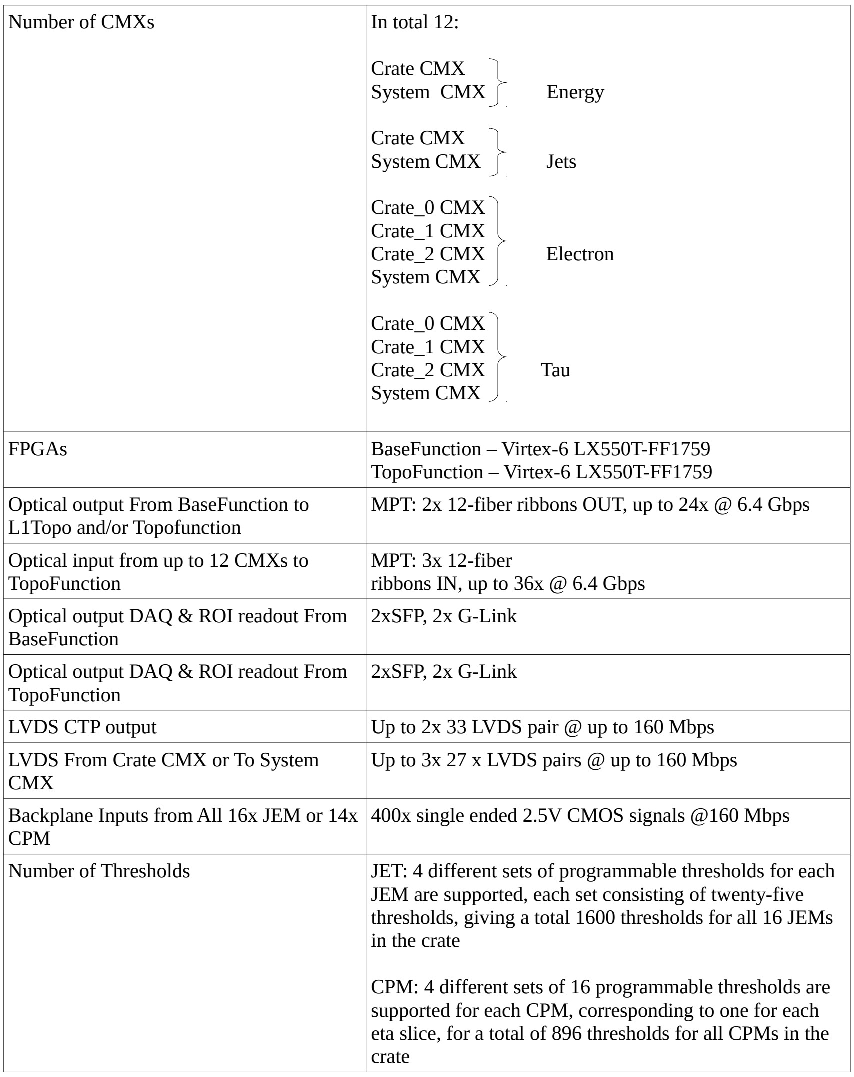


Figure 2: Crate CMX and System CMX arrangement in L1Calo.

## 1.6 CMX in numbers

The set of numbers which characterizes the CMX are presented in the table 1.



| | |
|---|---|
| Number of CMXs | In total 12:<br>Crate CMX<br>System CMX Energy<br>Crate CMX<br>System CMX Jets<br>Crate_0 CMX<br>Crate_1 CMX<br>Crate_2 CMX Electron<br>System CMX<br>Crate_0 CMX<br>Crate_1 CMX<br>Crate_2 CMX Tau<br>System CMX |
| FPGAs | BaseFunction – Virtex-6 LX550T-FF1759<br>TopoFunction – Virtex-6 LX550T-FF1759 |
| Optical output From BaseFunction to L1Topo and/or Topofunction | MPT: 2x 12-fiber ribbons OUT, up to 24x @ 6.4 Gbps |
| Optical input from up to 12 CMXs to TopoFunction | MPT: 3x 12-fiber ribbons IN, up to 36x @ 6.4 Gbps |
| Optical output DAQ & ROI readout From BaseFunction | 2xSFP, 2x G-Link |
| Optical output DAQ & ROI readout From TopoFunction | 2xSFP, 2x G-Link |
| LVDS CTP output | Up to 2x 33 LVDS pair @ up to 160 Mbps |
| LVDS From Crate CMX or To System CMX | Up to 3x 27 x LVDS pairs @ up to 160 Mbps |
| Backplane Inputs from All 16x JEM or 14x CPM | 400x single ended 2.5V CMOS signals @160 Mbps |
| Number of Thresholds | JET: 4 different sets of programmable thresholds for each JEM are supported, each set consisting of twenty-five thresholds, giving a total 1600 thresholds for all 16 JEMs in the crate<br>CPM: 4 different sets of 16 programmable thresholds are supported for each CPM, corresponding to one for each eta slice, for a total of 896 thresholds for all CPMs in the crate |

Table 1: The CMX in numbers.

# 2 Overall layout of the base FPGA firmware

A conceptual diagram of the functional blocks is shown in Figure 3. The functionality of backplane data capture and synchronization, data transmission over MGTs to L1Topo as well as readout, and to the CTP and crate/system CMX over LVDS links as well as VME communication will be common to all types of the CMX. Type-specific firmware modules conform to the common interfaces with the common modules.

# 3 Backplane data capture (Input Module)

VHDL component name: `CMX_INPUT_MODULE`

The function of the input module is to capture the backplane data, time-demultiplex it and

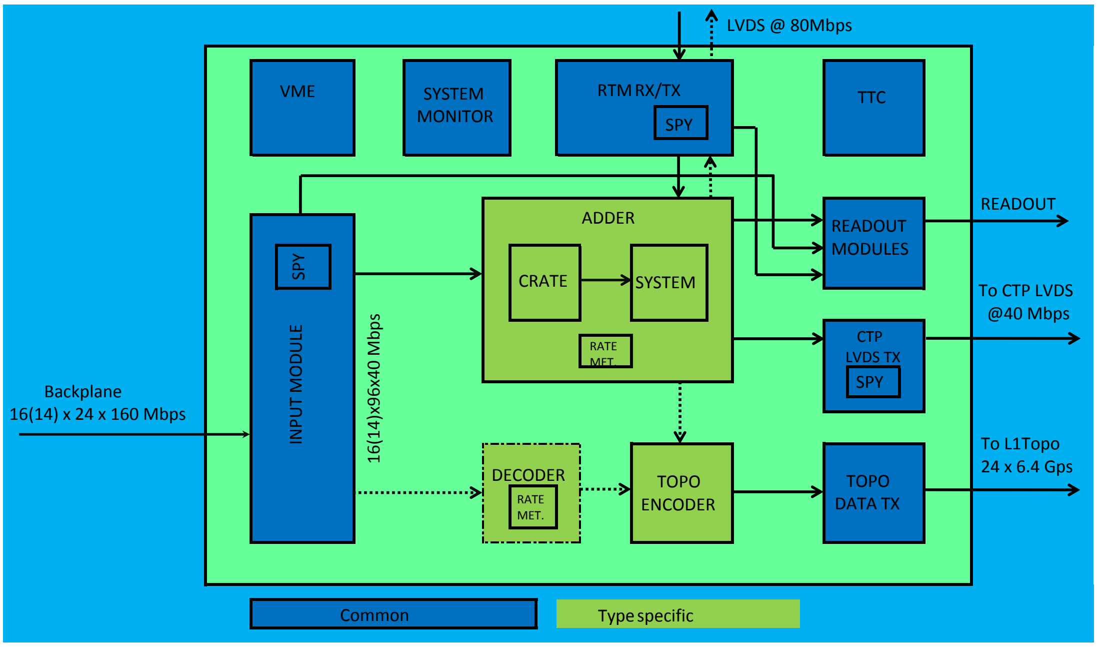


Figure 3: Conceptual diagram of the firmware modules and data flow between them.

bring it to the system time domain, as well as detect parity errors. The inputs of the module are the FPGA IOBs connected to the backplane transmission lines. Each processor input provides 24 data bits at 160 Mbps and one clock line at 80 MHz with edges centred in the data windows.

The data and clock signals from each processor input are piped through an IODELAY module, which provides the capability to delay the signals by up to 2.4 ns (up to 31 taps of 78 ps). The delays are controlled by the register block REG_RW_IDELAY_BACKPLANE. Writing to the last register in the block triggers reconfiguration of the IODELAY circuits for all of the backplane signals. The input module has 12 IDELAYCTRL circuits providing delay calibration. Calibration procedure of the IDELAYCTRL circuits is triggered on the assertion of the lock signal from the main system MMCM. The Mixed-Mode Clock Manager (MMCM) is used to generate multiple clocks with defined phase and frequency relationships to a given input clock [7]. Readiness of each calibration circuit is signalled by corresponding bis in the register REG_RO_IDELAYCTRL_RDY. The reset hold (for instance during main MMCM reset) and completion of the reset pulse is signalled by the corresponding bits in REG_RO_IDELAYCTRL_RST and REG_RO_IDELAYCTRL_WAS_RST respectively.

Data are captured and time de-multiplexed to 80 Mbps using the IDDR circuits built into each IOB using the forwarded clock. They are then de-multiplexed further to 96 bits x 40 Mbps using the forwarded clock, and then captured into a system clock domain register. The firmware distinguishes first pair of words from the second pair of words in an event based on the phase of the forwarded clock with respect to the system clock (derived from deskew 1 input). The startup procedure is initiated by global MMCM lock assertion or de-assertion of the input module reset from the register REG_RW_INPUT_MOD_RESET bit 0. Startup procedure completion for each input channel is signalled by assertion of the corresponding bit in the register REG_RO_INPUT_MOD_COUNTER_ENABLE.

It is expected that the relationship between the system clock and the backplane forwarded clocks will not vary. The deskew 1 clock must be adjusted so there is at least 2 ns advance between each of the clocks forwarded on the backplane and the deskew 1 clock).

The Input module provides 32-bit parity error counters (one for each module input) in the register block REG_RO_PARITY_ERROR_COUNTER, as well as a 32-bit global event counter at REG_RO_EV_COUNTER. The counters can be reset synchroneously by a write to the register REG_RW_COUNTER_RESET. The parity error is determined in the forwarded clock domain to aid IDELAY setting selection procedure. Hardware-based verification, capture and playback capabilities are also provided by the spy memories as described in section 10.2.

Any given channel can be masked by asserting a corresponding bit in the register REG_RW_BACKPLANE_INPUT_CHANNEL_MASK. If a mask is applied, the data from the channel will be substituted with an empty event data. Parity errors from masked channels are not counted, and do not contribute to error handling procedures in the downstream processing.

# 4 Rear Transition Module cable interface (RTM)

VHDL component name: CMX_SYSTEM_CABLE_INPUT_MODULE

Data between "crate" and "system" CMXs are transferred via the Rear Transition Module (RTM) cable connection. The direction of the connection is controlled by the BSPT FPGA as requested by the BF FPGA. The request lines are driven appropriately to the type (crate or system) of the CMX. Data are transferred at 80 Mbps / pair, together with the forwarded data clock. The RTM output module is equipped with spy memories, providing capture, verification and playback functionalities as described in section 10.2.

On the system type CMXs, the data are captured and de-multiplexed to 40 Mbps using the forwarded clock, registered using the deskew-2 derived clock. From there they are output to the adder modules where, after initial processing, the data are registered with the deskew-1 derived system clock. To ensure proper timing, the deskew-2 clock must be set at least 1.0 ns after the forwarded clock, and the deskew-1 clock must be set at least 2.0 ns after the deskew-2 clock. The two-stage RTM data capture ensures that on open window of data validity exists, as the deskew-1 system clock phase setting is primarily based on phases of the backplane forwarded clock. Setting the deskew-1 clock based on the backplane clocks may not be valid for capturing the cable input data directly.

Parity error detection is performed in the forwarded clock domain. Parity error counter for each of the cable inputs is provided in the register block REG_RO_RTM_PARITY_ERROR_COUNTER and can be synchroneously reset by a write to REG_RW_RTM_INPUT_COUNTER_RESET. Spy memories are provided with data capture, verification and playback functionality in each of the three clock domains (forwarded, deskew-2 and system clocks), supporting the timing procedure for determining and verifying the deskew-2 and deskew-1 settings as described in the software documentation.

A mask may be applied on any given cable by asserting a corresponding bit in the register REG_RW_RTM_INPUT_CHANNEL_MASK. If a mask is applied the data from the channel will be substituted with an empty event data. Parity errors from this cable will not be counted and will not contribute to error handling procedures in the downstream processing.

# 5 Adder

The adder is a system-specific block, with different versions used depending on the needed functionality and location in the system. In the CP, the adder performs multiplicity-based algorithms on EM or hadronic clusters for various applied threshold sets. In the JEP, one pair of CMXs performs jet multiplicity algorithms for jets with different size/ET thresholds, while the other pair performs system-wide energy sum algorithms.

The crate-level results of each adder block are merged over the RTM cables to one CMX that performs system-wide results to be sent to the CTP. The following three subsections describe each of the adder types in detail.

## 5.1 Cluster Adder

VHDL component name: `ADDER_TOP_VS_CP`

The cluster adder receives data from 14 CPMs, with four 25-bit words from each module formatted according to Figure 4. The four consecutive words provide the coordinates, 8-bits cluster energy and 5-bit encoded isolation information for up to five clusters.

The cluster adder produces 16 cluster multiplicities, each corresponding to the number of clusters that satisfy a certain threshold set based on a combination of cluster energy and isolation parameter. Four different sets of 16 programmable thresholds are supported for each CPM, corresponding to one for each eta slice, for a total of 896 thresholds for all CPMs in the crate. Each threshold is programmed via a 13-bit VME threshold register, corresponding to 5 bits for isolation and an 8-bit transverse energy threshold. (see VME register map 2, at the register block `ADDR_REG_RW_JET_THRESHOLD_BLOCK`). To apply the correct thresholds to a given cluster, the eta coordinate of each cluster is calculated from its respective presence bits and local coordinates. The cluster energy and isolation parameters are then compared against the relevant set of 16 thresholds, producing a group of 16 cluster multiplicities ("cluster hit bits"). Separate counts of cluster hit bits for each of the sixteen thresholds from all CPMs are produced, with an arithmetical saturation that limits the maximum count to 7.

In the crate CMX, odd parity and overflow bits are generated and sent with the intermediate hit sums over over cable links via the rear transmition module to the system CMX. An overflow bit is set if at least one CPM finds more than 5 clusters. The intermediate hit sums are saturated when overflow bit is set to 1 or if there is a parity error with the force flag set to 1. The force flag is controlled by the register `ADDR_REG_RW_QUIET_FORCE`.
The system CMX receives the intermediate sums by cable link, formatted as in Figure 5. These are added to the intermediate, "local" sums to form the final, system-wide cluster hit counts. The intermediate and "local sums are synchronized with the use of programmable pipeline controlled by the register `REG_RW_DELAY_INPUT_DATA_ADDER`. The hit sums are sent over parallel links to the CTP, formatted as in Figure 6.

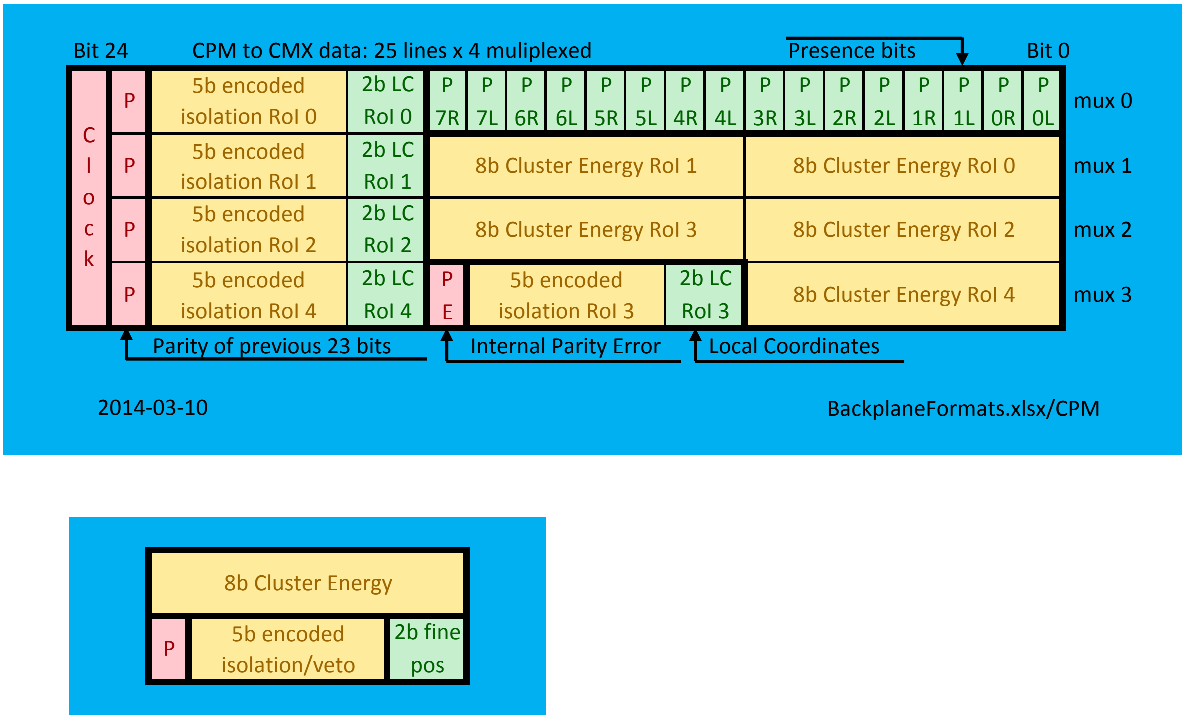


Figure 4: Format of Backplane CPM - CMX CP data.

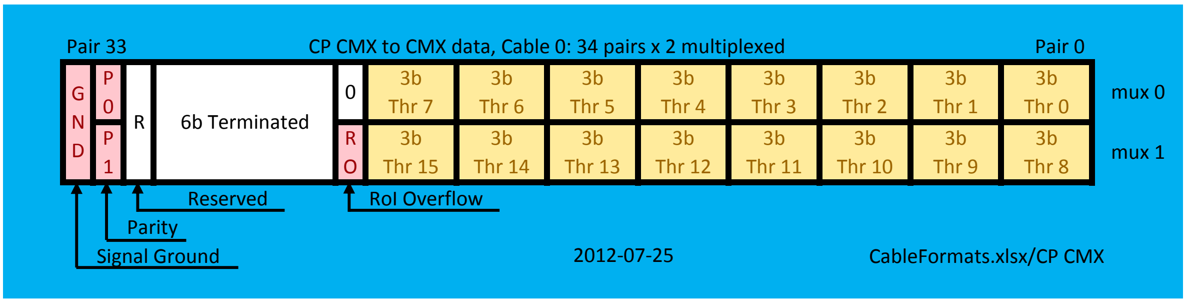


Figure 5: Format of cable input data from remote CP CMX.

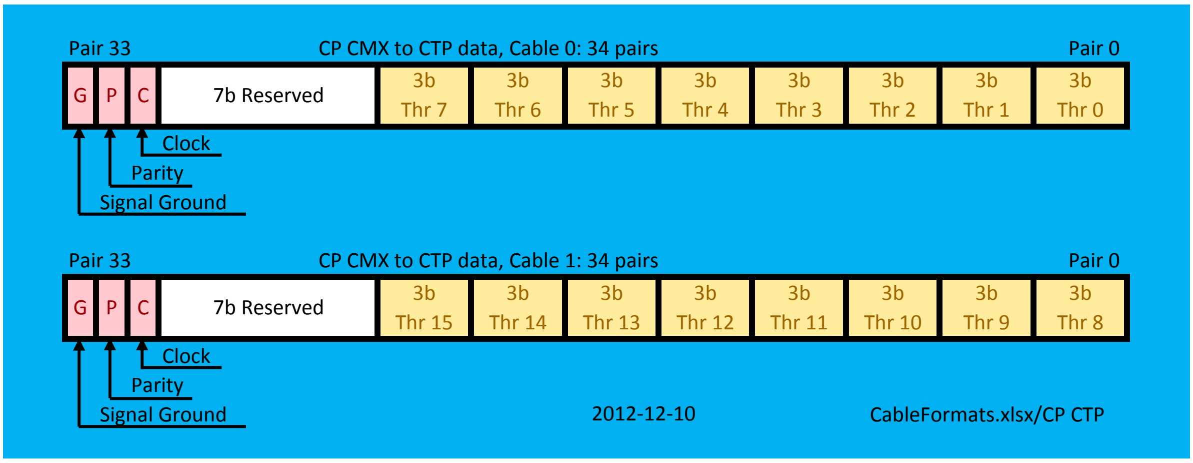


Figure 6: Format of cable data from CP CMX to CTP.

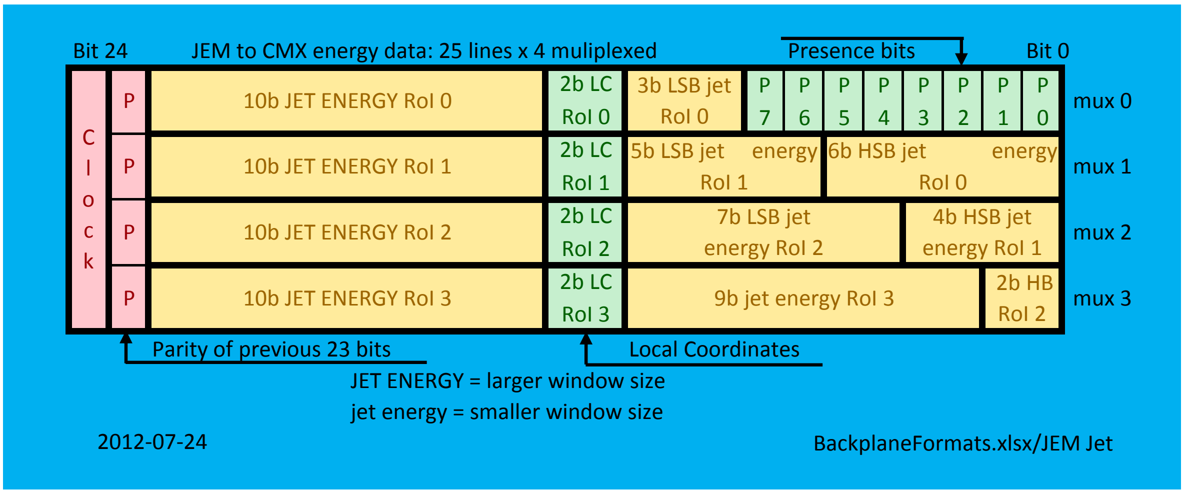


Figure 7: Format of Backplane JEM - CMX jet data.

## 5.2 Jet Adder

VHDL component: `ADDER_TOP_VS`

The jet counting process is similar to the cluster counting process. There is no requirement to treat forward and central jets separately, so all jets are included in each sum. The received data from each JEM consists of four data words, each word containing 25-bits as shown in Figure 7. The consecutive words in the backplane package provide the coordinates and 9- and 10-bit energy sums for up to four jets.

To perform the jet counting algorithm, the energies of each jet are compared against a set of relevant thresholds. Four different sets of programmable thresholds for each JEM are supported (one for each eta slice), each set consisting of twenty-five thresholds, giving a total 1600 thresholds for all 16 JEMs (see table 10). Each VME threshold register contains 11-bits: 1-bit corresponds to the cluster size while 10-bit the threshold value (see VME register map 2, the register block `ADDR_REG_RW_JET_THRESHOLD_BLOCK`). The cluster size bit chooses which energy sum (9- or 10-bit) is compared against the threshold. The cluster size one corresponds to the 9-bit energy, while the cluster size zero to the 10-bit energy.

Similarly to the cluster adder, the eta coordinate of a jet calculated from the presence bits and local coordinates is used to select the correct threshold set. The relevant size and minimum energy criteria are applied, producing a set of twenty-five "jet hit" bits per jet. The multiplicities of corresponding jet hit bits from all JEMS are counted seperately for each of the twenty-five thresholds, with arithmetic saturation to seven or three bits, depending on the threshold number. These 7- and 3-bit saturated sums are encoded in 3- and 2-bit bins, respectively.

In the crate CMX, odd parity and overflow bit are generated and sent with the intermediate hit sums over cable links to the system CMX. An overflow bit is set when at least one JEM finds more than 4 jets. The intermediate hit sums are saturated when overflow bit is set to 1 or if there is a parity error with the force flag set to 1. The force flag is controlled by the register `ADDR_REG_RW_QUIET_FORCE`.

The system CMX receives intermediate sums by cable link, formatted as in Figure 8. These are added to the intermediate, "local" sums to form the final, system-wide jet hit counts. The intermediate and "local sums are synchronized with the use of programmable pipeline controlled by the register `REG_RW_DELAY_INPUT_DATA_ADDER`. The hit sums are sent over parallel links to the CTP, formatted as in Figure 9.

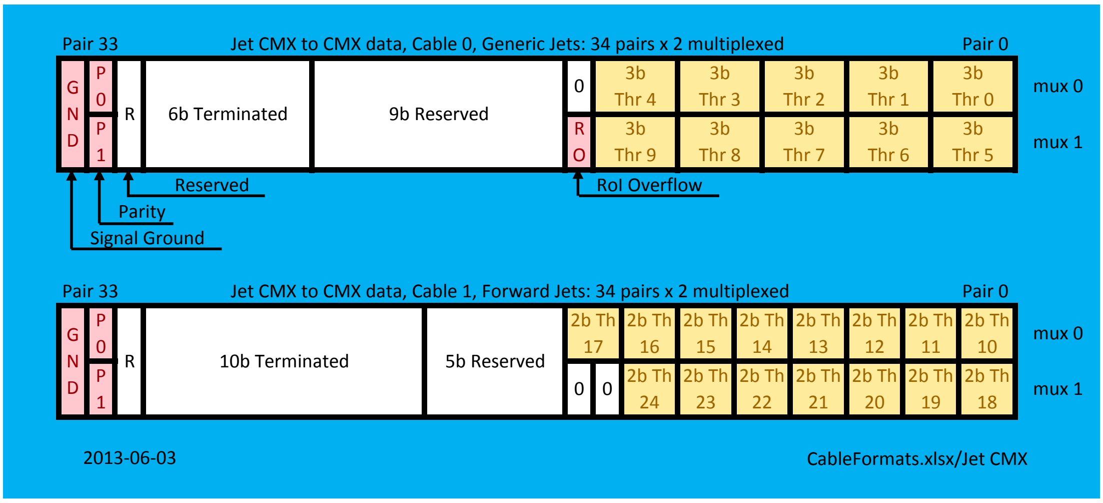


Figure 8: Format of cable input data from remote Jet CMX.

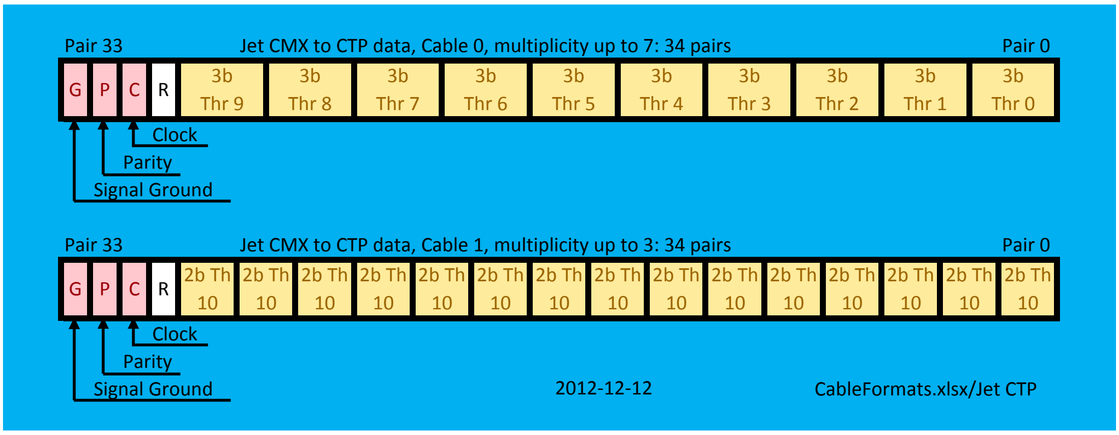
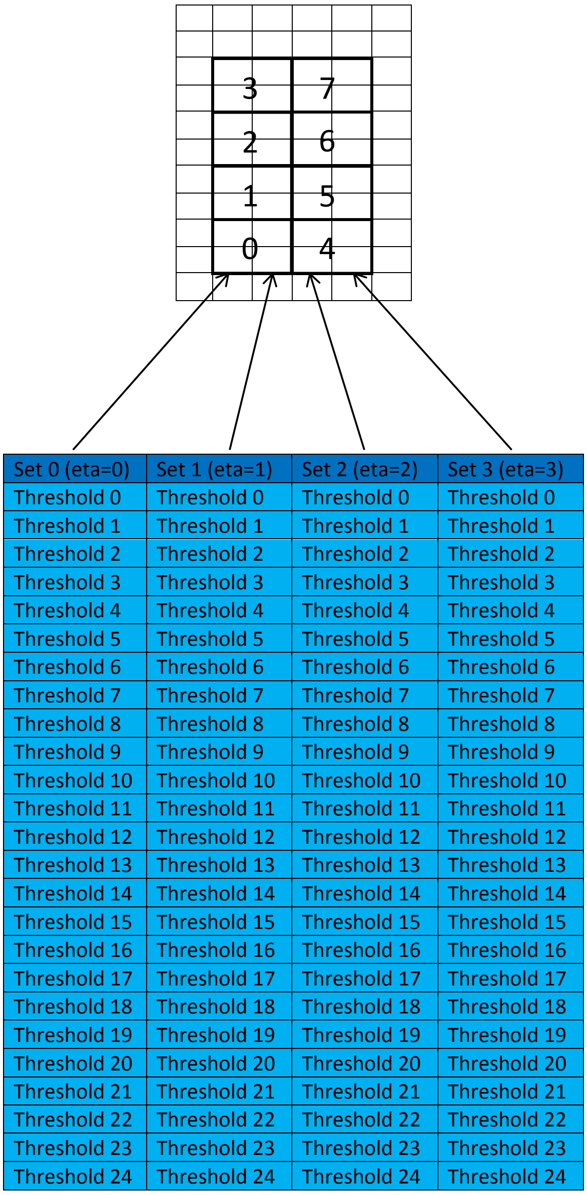


Figure 9: Format of cable data from Jet CMX to CTP.



| Set 0 (eta=0) | Set 1 (eta=1) | Set 2 (eta=2) | Set 3 (eta=3) |
|---|---|---|---|
| Threshold 0 | Threshold 0 | Threshold 0 | Threshold 0 |
| Threshold 1 | Threshold 1 | Threshold 1 | Threshold 1 |
| Threshold 2 | Threshold 2 | Threshold 2 | Threshold 2 |
| Threshold 3 | Threshold 3 | Threshold 3 | Threshold 3 |
| Threshold 4 | Threshold 4 | Threshold 4 | Threshold 4 |
| Threshold 5 | Threshold 5 | Threshold 5 | Threshold 5 |
| Threshold 6 | Threshold 6 | Threshold 6 | Threshold 6 |
| Threshold 7 | Threshold 7 | Threshold 7 | Threshold 7 |
| Threshold 8 | Threshold 8 | Threshold 8 | Threshold 8 |
| Threshold 9 | Threshold 9 | Threshold 9 | Threshold 9 |
| Threshold 10 | Threshold 10 | Threshold 10 | Threshold 10 |
| Threshold 11 | Threshold 11 | Threshold 11 | Threshold 11 |
| Threshold 12 | Threshold 12 | Threshold 12 | Threshold 12 |
| Threshold 13 | Threshold 13 | Threshold 13 | Threshold 13 |
| Threshold 14 | Threshold 14 | Threshold 14 | Threshold 14 |
| Threshold 15 | Threshold 15 | Threshold 15 | Threshold 15 |
| Threshold 16 | Threshold 16 | Threshold 16 | Threshold 16 |
| Threshold 17 | Threshold 17 | Threshold 17 | Threshold 17 |
| Threshold 18 | Threshold 18 | Threshold 18 | Threshold 18 |
| Threshold 19 | Threshold 19 | Threshold 19 | Threshold 19 |
| Threshold 20 | Threshold 20 | Threshold 20 | Threshold 20 |
| Threshold 21 | Threshold 21 | Threshold 21 | Threshold 21 |
| Threshold 22 | Threshold 22 | Threshold 22 | Threshold 22 |
| Threshold 23 | Threshold 23 | Threshold 23 | Threshold 23 |
| Threshold 24 | Threshold 24 | Threshold 24 | Threshold 24 |

Figure 10: Four different sets of programmable thresholds for each JEM are supported (one set for each eta slice).

## 5.3 Energy summation and threshold application

### 5.3.1 Calculation of the local sums

VHDL component name: `CMX_SUM_Et`

Two CMXs located in the JEP0 and JEP1 crate receive the energy sums from JEMs in the format shown in Fig. 11. The CMX then performs local summing of the total transverse energies (TE) and $E_x$, $E_y$ taking into account the location of the source modules to determine the sign of the summation for $E_x$ and $E_y$. The local sums are shortened to 15-bit signed values for $E_x$, $E_y$ and 14-bit unsigned value for TE. Overflow bits are calculated if the result overflows the 15 or 14 bit value however the value itself is not in any way corrected. Similarly energies forwarded by the JEMs indicated in the restriction masks are summed to form restricted TE and restricted $E_x$ and $E_y$ and the three restricted overflow bits are calculated. Two separate restriction masks are available: one for TE at the register `REG_RW_SUMET_MASK` and one for $E_x$ and $E_y$ at the register `REG_RW_MISSET_MASK`. The local sums as well as the six overflow bits are sent on serial links to topological processor as described in section 6. These quantities are also sent from crate to system CMX over the RTM connection in the format shown in Fig. 13. Within the system CMX the quantities are transferred to the system adder via a programmable pipeline controlled by the register `REG_RW_DELAY_INPUT_DATA_ADDER`, as in the CP and Jet system CMXs.

### 5.3.2 Threshold application

The system CMX sums the locally computed energy values and energy values received from the other CMX to form 15-bit unsigned TE and signed $E_x$ and $E_y$ sums and their restricted equivalents. Six global overflow bits are computed for each quantity. These quantities are used to set total energy (TE), missing energy (XE), restricted TE, restricted XE and missing energy significance bits to CTP according to the diagram in Fig. 13. Calculation proceeds by computing set of criteria and then combining them into a final result.

#### 5.3.2.1 TE thresholding

For the TE threshold bit i to be set we require that the computed TE is strictly greater than the threshold in the corresponding register block at `REG_RW_SUM_ET_THR_BLOCK` or that a TE overflow has occured. Similarly for the restricted TE threshold bit i to be set we require that the computed restricted TE is strictly greater than the corresponding threshold in the corresponding register bock at `REG_RW_SUM_ET_RES_THR_BLOCK` or that a restricted TE overflow has occured.

#### 5.3.2.2 XE thresholding

Criterion i for missing energy is: Is missing energy > threshold or has an overflow occurred in local or global $E_x$ or $E_y$ calculation?

In firmware we can perform an equivalent comparison:

$$XE^2 > T^2_{MISS,i} \tag{1}$$

where $E_x$ and $E_y$ are global sums computed with 15 bit (signed) precision + overflow carry $XE^2$ is Missing Energy squared: $XE^2 = E_x^2 + E_y^2$ - since $E_x$ and $E_y$ are 15 bit $XE^2$ is 31 bit $T^2_{MISS,i}$ is a 31 bit parameter from VME register block `REG_RW_MISS_E_THR_BLOCK`- the ith missing energy threshold. Similarly for the restricted XE threshold i the threshold ith threshold from `REG_RW_MISS_E_RES_THR_BLOCK` is used together with the restricted XE overflow.

#### 5.3.2.3 XS thresholding

Several criteria are computed and combined in the final step.

**Criterion** $c0_i$**:** Is Missing Energy < minimum Missing Energy ?
In firmware we can perform an equivalent comparison:

$$XE^2 < T^2_{MISS,MIN,i} \tag{2}$$

where $T_{MISS,MIN,i}$ is a 31 bit parameter from the register block - the minimum Missing Energy squared for threshold i from the register block `REG_RW_T_MISS_E_MIN_PARAM_BLOCK`.

**Criterion** $c1_i$**:** Is Missing Energy >= maximum Missing Energy, or has overflow occurred in calculating local or global Ex and Ey beyond (signed) 15 bits?
In firmware we can perform an equivalent comparison:

$$XE^2 >= T^2_{MISS,MAX,i} \tag{3}$$

where $T^2_{MISS,MAX,i}$ is a 31 bit parameter from the register block `REG_RW_T_MISS_E_MAX_PARAM_BLOCK`- the maximum Missing Energy squared for threshold i.

**Criterion** $c2_i$**:** Is$\sqrt(SumEnergy) < \sqrt(minimumSumEnergy)$?
In firmware we can perform an equivalent comparison:

$$TE < T_{SUM,MIN_i} \tag{4}$$

where $T_{SUM,MIN,i}$ is: a 15 bit parameter from the register block `REG_RW_T_SUM_E_MIN_PARAM_BLOCK` - the minimum Sum Energy for threshold i.

**Criterion** $c3_i$**:** Is$\sqrt(SumEnergy) >= \sqrt(maximumSumEnergy)$ or has TE local or remote summation overflown?
In firmware we can perform an equivalent comparison:

$$TE >= T_{SUM,MAX,i} \tag{5}$$

where $T_{SUM,MAX,i}$ - is: a 15 bit parameter from the register block `REG_RW_T_SUM_E_MAX_PARAM_BLOCK`- the maximum Sum Energy for threshold i.

**Criterion** $c4_i$**:** Is missing energy significance XS > threshold $T_i$?
missing energy significance is:

$$XS = \frac{XE}{a(\sqrt{TE} - b)} \tag{6}$$

where $a$ is a scale and $b$ is an offset parameter. In order to avoid the truncation in square root and division, in FW we perform two comparisons:

$c4A_i$:

$$(T_i^2 a^2 TE + T_i^2 a^2 b^2 - XE^2) < 0 \tag{7}$$

$c4B_i$:

$$4(T_i^2 a^2)^2 b^2 TE > (T_i^2 a^2 TE + T_i^2 a^2 b^2 - XE^2)^2 \tag{8}$$

Following bit-ranges are used for the configuration parameters:

- $T_i^2 a^2$ - a single parameter from the register block `REG_RW_XS_T2_A2_THR_BLOCK` (there is no need to separate ”a” and ”T”): 31 bits to match the dynamic range of $XE^2$.
- $b_i^2$ - 15 bit parameter from the register block `REG_RW_XS_B2_PARAM_BLOCK` to match the dynamic range of TE, for threshold i. For the comparison to be equivalent the inequality 6 $b_i^2$ must be 1 or greater.

Bit ranges of the compared quantities are 95 bits for the LHS and 98 for RHS

The criteria are combined according to the following pseudocode for combining the criteria:

```
if c0_i
  XS hit i false
else

  if c1_i
    XS hit i true
  else
    if c2_i or c3_i
      XS hit i false
    else
      if c4A_i
        XS bit i set
      else
        if c4B_i
          XS bit i set
else
          XS bit i not set
        end if
      end if
    end if
  end if

end if
```

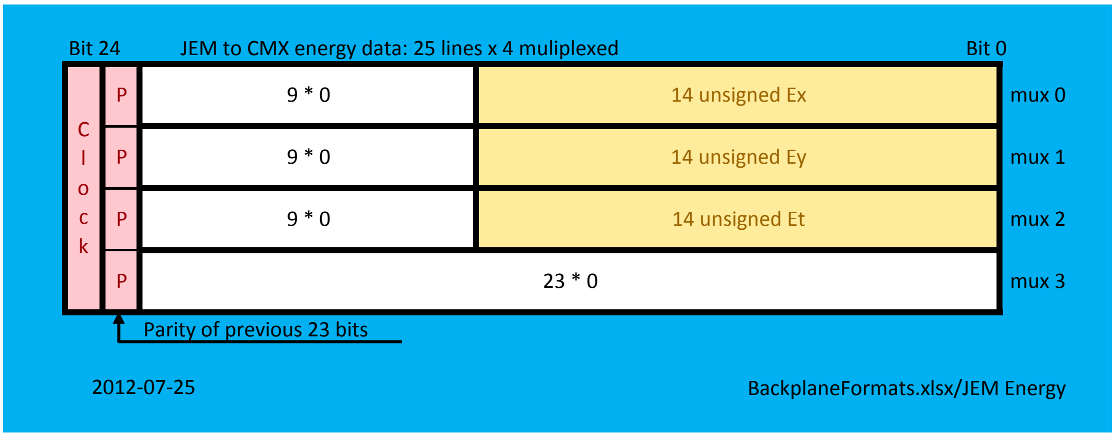


Figure 11: Format of Backplane JEM - CMX Energy data.

# 6 Topo stream

## 6.1 Topological data protocol and format

VHDL component names:

`DECODER`
`TOPO_DATA_TX`
`CMX_JET_TOPO_ENCODER`

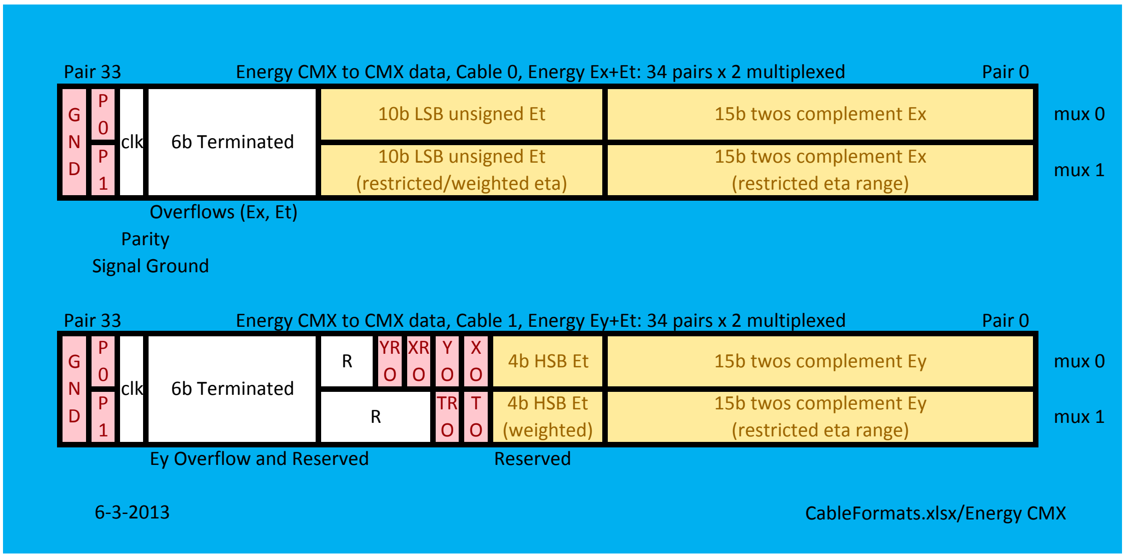

Pair 33
Energy CMX to CMX data, Cable 0, Energy Ex+Et: 34 pairs x 2 multiplexed
Pair 0
GND
P0
P1
clk
6b Terminated
10b LSB unsigned Et
15b twos complement Ex
mux 0
10b LSB unsigned Et (restricted/weighted eta)
15b twos complement Ex (restricted eta range)
mux 1
Overflows (Ex, Et)
Parity
Signal Ground
Pair 33
Energy CMX to CMX data, Cable 1, Energy Ey+Et: 34 pairs x 2 multiplexed
Pair 0
GND
P0
P1
clk
6b Terminated
R
YRO
XRO
YO
XO
4b HSB Et
15b twos complement Ey
mux 0
R
TRO
TO
4b HSB Et (weighted)
15b twos complement Ey (restricted eta range)
mux 1
Ey Overflow and Reserved
Reserved
6-3-2013
CableFormats.xlsx/Energy CMX

Figure 12: Format of cable input data from remote Energy CMX.

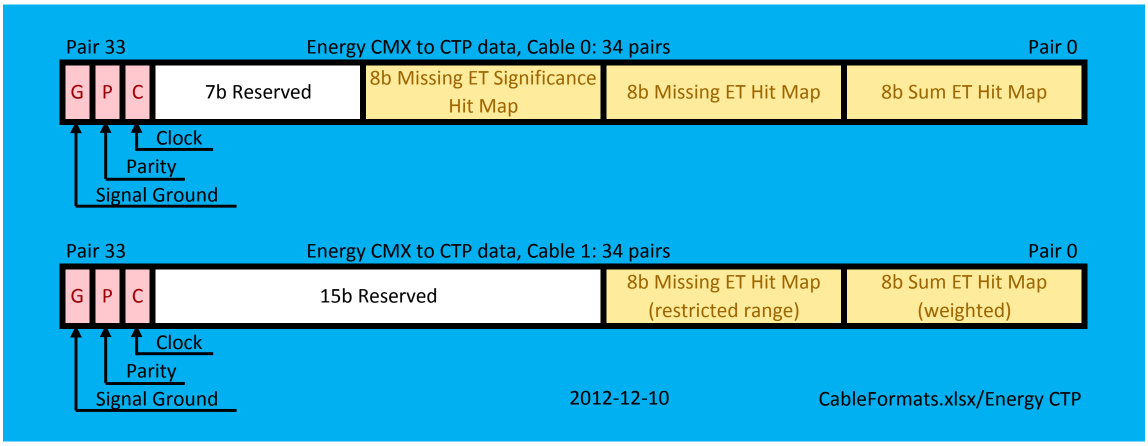

Pair 33
Energy CMX to CTP data, Cable 0: 34 pairs
Pair 0
G
P
C
7b Reserved
8b Missing ET Significance Hit Map
8b Missing ET Hit Map
8b Sum ET Hit Map
Clock
Parity
Signal Ground
Pair 33
Energy CMX to CTP data, Cable 1: 34 pairs
Pair 0
G
P
C
15b Reserved
8b Missing ET Hit Map (restricted range)
8b Sum ET Hit Map (weighted)
Clock
Parity
Signal Ground
2012-12-10
CableFormats.xlsx/Energy CTP

Figure 13: Format of cable data from Energy CMX to CTP.

`CMX_CP_TOPO_ENCODER`

TOB data are formatted in a type-specific manner and transmitted from CMX to the L1Topo processor. In this section we describe the formats, as well as line encoding, positioning of the TOB data within the transmission, the data fields within a TOB, and provisions for data alignment and error checking.

Each CMX provides up to 24 fiber outputs to L1Topo. Each EM/Tau CMX provides four duplicate 6-fiber outputs, while each Jet CMX provides three duplicate 8 fiber outputs. The energy sum CMX output to L1Topo can be packed on a single fiber, and four duplicate of SumET outputs are provided. Within this limit up to 32 Jet TOBs and up to 30 EM/Tau TOBs per event can be sent to each duplicate output.

The outputs of each CP and Jet CMX contain a sorted list of TOBs. Sorting is performed in the decoder (c.f. 3) module using the Batcher odd-even mergesort [2, 6] method. In the Jet CMX, the TOBs are sorted according to their 10-bit Et value, while CP TOBs have are sorted by their 8-bit Et values. If the 33rd Jet TOB (31st CP TOB) has non-zero Et, an overflow is signalled on all output channels. In case of backplane overflow the same overflow bit is asserted.

In the SumET CMX the computation of local energy sums is performed in the first stages of the adder module, as the local sums sent to L1Topo are identical to those sent to the "system" summing module.

The line rate on the CMX → Topo serial links is 6.4 Gbps. The serial data links use 8b/10b encoding, providing a sufficient number of transitions for PLL lock on the RX side as well as DC balancing. Virtex 6 and 7 FPGAs provide hardware support for 8b/10b encoding, making the implementation especially convenient. Given the line rate and encoding overhead, the data rate on a single serial link is 5.12 Gbps, or 128 bits of information per fiber per event. In the rest of this document this unencoded (or “parallel”) data will be discussed, as this is the data available for processing in the FPGA logic.

The maximum number of EM/Tau trigger objects that can be sent on a single serial link is five per event, for 115 bits of 128 available. A maximum of four jet TOBs can be sent per fiber and event, using 112 of the 128 bits. A single bit signaling overflow of backplane or zero-suppressed data is also be sent on all channels, leaving 12 bits on any given channel for error checksum. Based on the available bandwidth for error checking, a Cyclic Redundancy Check (CRC) of length 12 was chosen for signaling transmission errors. The error-detection properties and the computation of the 12 bit CRC checksum are discussed in the next section. The formats of the Jet and EM CMX to L1 Topo data transmission are shown in figures 14 and 15. The information carried by each TOB is shown in figures 16 and 17.

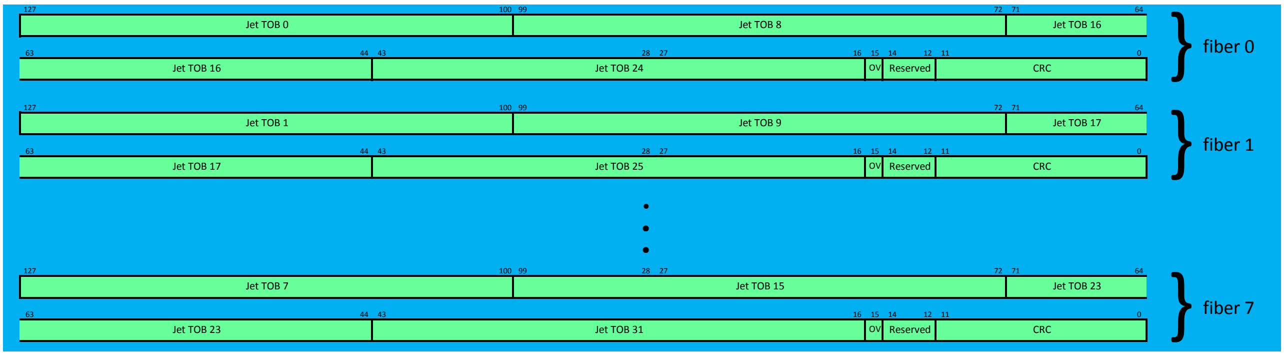


Figure 14: Jet TOB packing on fibers. The word carrying the highest significant bits of the extended message (127 down to 112) are sent first, then successively the words carrying lower significant bits.

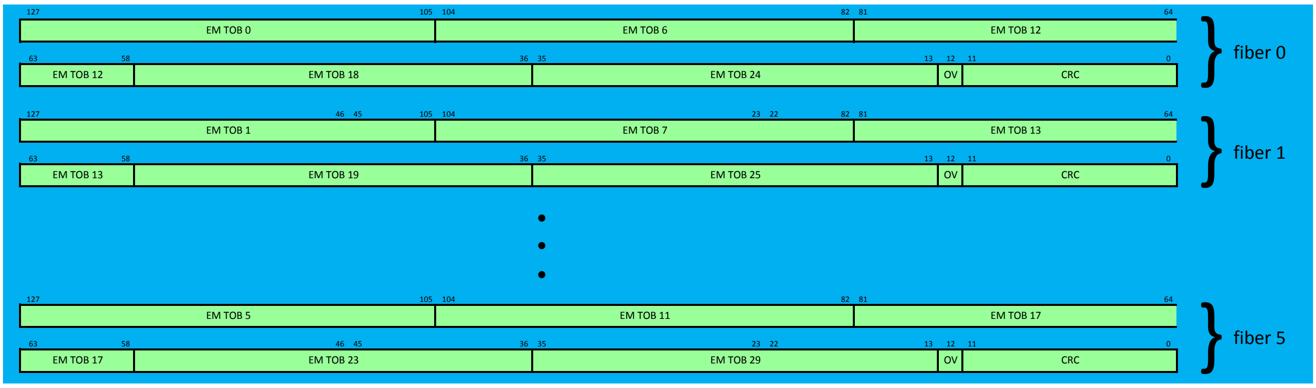


Figure 15: CP TOB packing on fibers. The word carrying the highest significant bits of the extended message (127 down to 112) are sent first, then successively the words carrying lower significant bits.

The TOBs are front-loaded onto the serial links in descending order of $E_T$ (for the Jet TOB the 10-bit $E_T$ value is used). The highest energy TOB in the event is sent as the first TOB of the first link, the second-highest energy TOB in the event is sent as the first TOB on the second link, and so on. If there are more than 8 (6) TOBs to be sent by the Jet (CP) CMX, the second TOB slots on

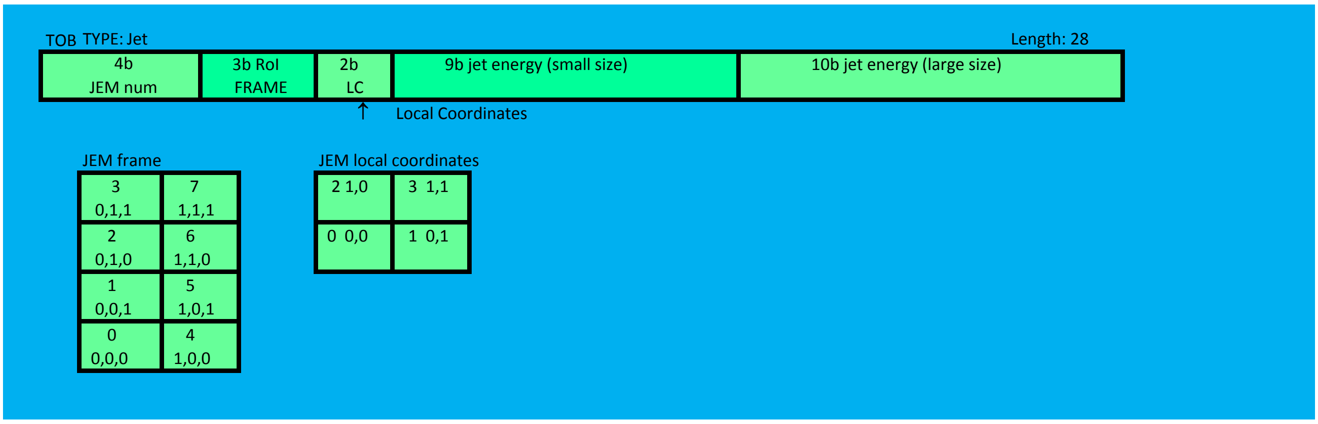


Figure 16: Jet TOB format.

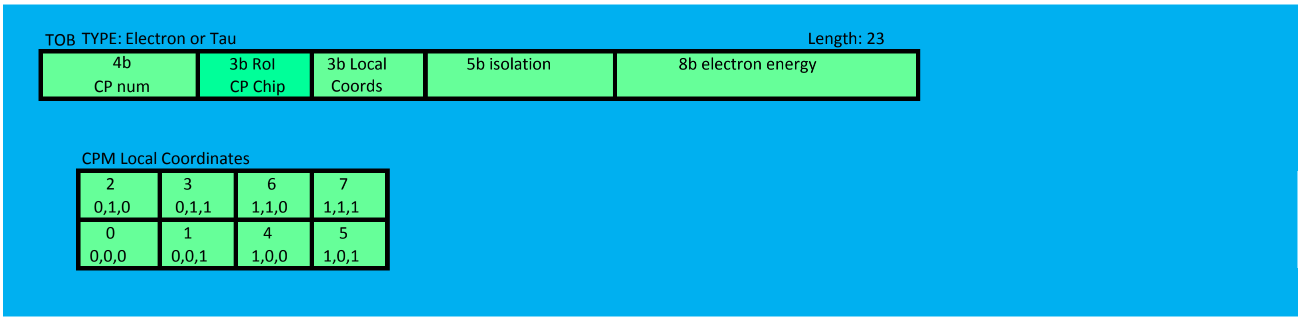


Figure 17: CP TOB format.

each link will start to be filled in the same ordering: The 9th (7th) highest $E_T$ Jet (CP) TOB will be sent on 2nd slot on 1st channel and so on.

For the Jet (CP) CMX, if the 3rd and 4th (3rd, 4th and 5th) TOBs positions have 0 energy, bits 71 to 52 are replaced by a 20-bit alignment word containing a K28.5 character and 12 bit BCID. The K character is used for RX byte alignment and monitoring. The fixed position of the K character (always on byte 8) (if it appears) gives the receiver necessary information for initial alignment and subsequent monitoring of the event boundaries within the data stream. The receiver shall keep positive and negative comma alignment enabled, and monitor for realignment events (indicating a data transmission problem). The inclusion of the BCID counter allows for aligning the information on different fibers to the same event and subsequent monitoring. The alignment word is shown in Fig. 18.

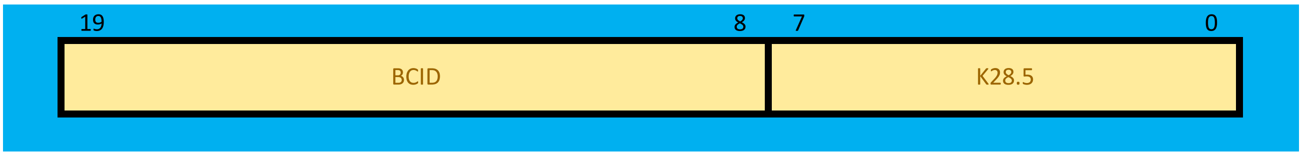


Figure 18: Alignment word.

The energy sum information uses 94 bits including the 6 overflow bits. The 128 bit per event bandwidth allows for the 94 bits of physics information, together with the alignment word and the CRC sum, to be sent on every event. However the constant position of the alignment word demands partitioning of the energy sum TOB as shown on Figure 19. Four serial channels carry copies of the SumET information. Twenty remaining channels do not carry physics information, but the carrier information (alignment word and the CRC) are sent on these channels.

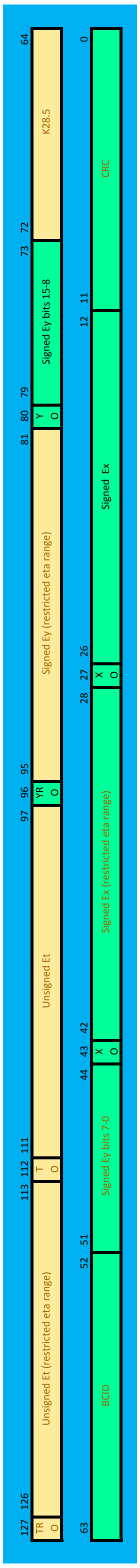


Figure 19: Sum $E_T$ packing on fibers. The word carrying the highest significant bits of the extended message (127 down to 112) are sent first, then successively the words carrying lower significant bits.

Sorting of the TOBs for Jet and CP CMXs is performed in the decoder module, and a dedicated decoder flavour is supplied for these two CMX types. The SumET CMX flavour does not have a decoder module; local sum information is computed in the adder module and also used in setting of the threshold hits on the CTP path.

Each CMX type has a simple encoder module, which arranges the "physics" information to be sent on each serial channel according to physical CMX board layout and the CMX-to-L1Topo fiber map. In case of the SumET CMX the encoder also provides four copies of the physics payload, while for Jet and CP CMX the copies are provided by the decoder. Neither decoder nor encoder modules have programmable parameters.

The Topo TX module encapsulates GTX transceiver and their clocking infrastructure and handles their startup procedures. This module receives 40 Mbps data from the decoder module and serializes it to 320 Mbps, while also splicing the alignment word, overflow bit and the CRC. A minimalistic fixed-latency FIFO is implemented for each channel to transfer the 320 Mbps data in the system clock domain to the user clock domain of each transceiver.

The serialisation module is re-started when the main system MMCM becomes locked. The GTX transceivers can be reset by asserting bit 1 of the `REG_RW_TOPOTR_GTX_RESET`. Upon de-asserion of the reset bit the synchronisation procedure of each GTX is performed followed by the synchronisation of the system clock to user clock FIFOs. The logical OR of the status of all GTX transceivers is published on bit 1 of the register `REG_RO_TOPOTR_GTX_STATUS`. The (hardware level) polarity flip of each GTX is controlled by corresponding bit of the register `REG_RW_TX_POLARITY`.

## 6.2 CRC calculation and performance

The cyclic redundancy check provides varying levels of transmission error detection depending on the length of the transmitted data ("message") and the length of the checksum and the choice of so called "generator polynomial". The CRC is especially well suited for detecting errors common to communication channels. In particular, provided that the generator polynomial is well chosen the CRC will detect any transmission error where odd number of bits are flipped and any burst [1] error of length up to the length of the checksum. Protection against arbitrary bit flips scattered through the transmission is highly dependent on the generator polynomial choice. Note that since the data is 8b/10b encoded protection against burst errors is highly desired. If a bit flip occurs in the serialized (10b) data the resulting ten bit pattern may not be in the encoding table, or it is in the table and therefore is interpreted by the receiver as a different eight bit payload. The megabit transceiver hardware checks for the first possibility (user logic must monitor the appropriate port (RXNOTINTABLE) of the receiver). Error resulting in a valid 10-bit character will be detected by the CRC performed on the decoded data. A failure resulting in a dark fiber will also be caught by RXNOTINTABLE as within the 8b/10b encoding long strings of 0's or 1's are not permitted. Alternatively the receiver PLL will lose synchronization. The CRC does not need to protect against this failure mode, however the receiver logic must instrument apropriate ports on the receiver.

Since the maximum length of the "message" (physics+alignment data) present in the system is 116 bits and the transmission length is 128 we are left with 12 bits for the CRC check. Following recommendations in [5] a generating polynomial 0x80f has been chosen (in notation with implied leading 1). Performance of the error check provided by this choice of CRC has been tested using a software model.

It may be desirable to test against possible single bit flips giving valid 10-bit characters resulting in more than one length 8 burst errors scattered through the transmission, however even without this check the error detection capability seems sufficient.

The calculation of the CRC in hardware is relatively simple and incurs a modest latency and logic cost. In the implementation on the CMX the calculation proceeds by computing partial results on 16-bit words at 320 MHz. The results are successively combined as the extended message [2] is fed

[1] 1. Burst error of length n is a series random of bit flips where the first and last bit flips are confined to a bit string of length n. If the offset of the first bit flipped is k then the offset of the last bit flipped is k+n-1.

[2] Extended message for the CRC length q calculation is the is the message appended with q 0's. In our case

through the computation unit and at the last word the CRC is appended to the message (this is why the CRC needs to appear at the end of the transmission). As compared to not computing the CRC the data appears on the output one tick of the 320 MHz clock later. Internally the CRC computation module uses two 8 x 12 ROMs explicitly implemented in the distributed RAM resources to improve timing performance. The resources cannot be shared between the channels and need to be replicated.

# 7 Readout

VHDL component name:
`GLINK_ENCODER`
`DAQ_GLINK`
`GLINK_INTERFACE`
`DAQ_COLLECTOR`

Upon an L1A, the CMX reads out event data to the data acquisition system (DAQ) as well as to the RoI Builder for use by the Level-2 Trigger. The CMX DAQ system comprises several components, including pipeline memories, derandomiser FIFOs, and shift registers that feed readout data to an emulated 20-bit G-link transmitter.

During normal running, backplane input data, intermediate sums, final hit counts and some control data are stored in the pipeline memories for every bunch crossing. When an L1A is received, the CMX copies data for the relevant bunch crossings from the pipeline memories to the derandomiser FIFOs. The FIFO outputs are written in parallel to 20 shift registers, one for each of the 20 emulated G-Link input data pins. From there the data are shifted serially into the emulated G-Link. The serial stream to each G-link input is followed by an odd parity bit for error detection. The G-Link encoded data are transmitted by the low-speed optical transmitters to the L1Calo RODs, where they are reformatted into S-Link ROD fragments, and sent on to the DAQ or RoI Builder.

## 7.1 G-link protocol

Readout to the DAQ and RoI RODs is carried out by a pair of emulated G-link transmitters with 20-bit data fields as shown in Figure 20, using GTX transmitters clocked at 960 Mbits/s. The G-Link protocol was successfully implemented and tested in Virtex-6 [1]. In eye-diagram tests of the optical output performed on an evaluation board, it was seen that the FPGA logic and GTX can produce an optical output exceeding the specifications of the original G-link hardware, with rise and fall times measured at less than 240 ps as shown in Figure 21.

The CMX G-Link protocol encodes 20 bits of user data along with a 4-bit CIMT header, and supports both the "data available" (DAV) signal and an internally-generated FLAG bit for additional link error detection in the ROD. When data is available in the shift registers, the DAV signal is asserted and the G-link encoder receives and processes the 20 serial lanes in parallel at 40 Mb/s. The 24-bit output of the Encoder is then multiplexed to 8 bits at 120 MHz, which are then passed to the input of a GTX transmitter running at 960 Mbit/s. The GTX output is finally brought to the Low Speed Optical components for transmission to the RODs. After an event has been read out, the CMX logic de-asserts the DAV signal. The emulated G-Link returns to its quiescent state, sending fill frames for a minimum of three bunch crossings before a new event can be read out.

## 7.2 Handling Multiple Time Slices

The CMX firmware includes a feature which allow to copy multiple data slices from the pipeline memory into the FIFO. The multiple data slice parameter is controlled by the register:

the 116 bits of physics and alignment data are appended with 12 zeros so that the last 16-bit word fed into the CRC calculation module consists of the last four bits of the physics message (on the four most significant bits) and 12 0's on the least significant bits.

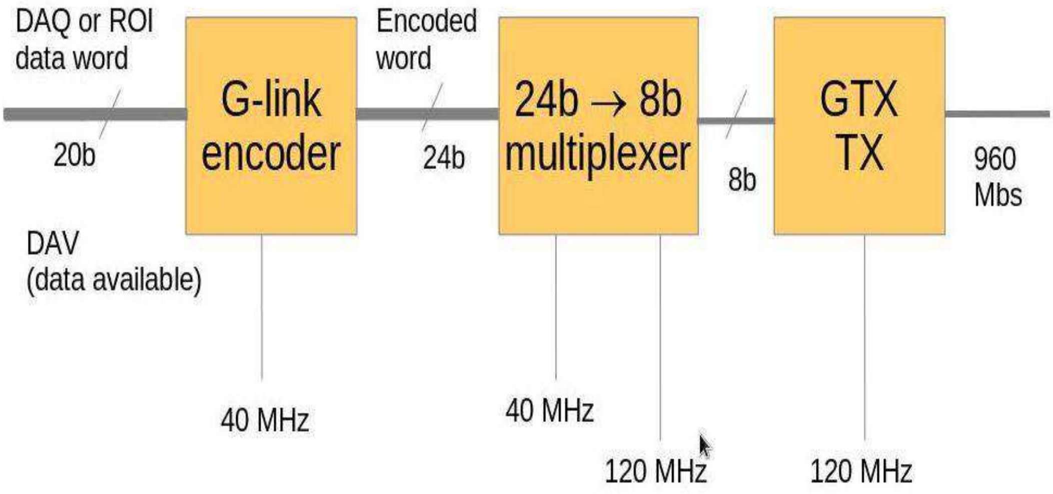


Figure 20: Overview of the emulated G-link protocol in Virtex-6.

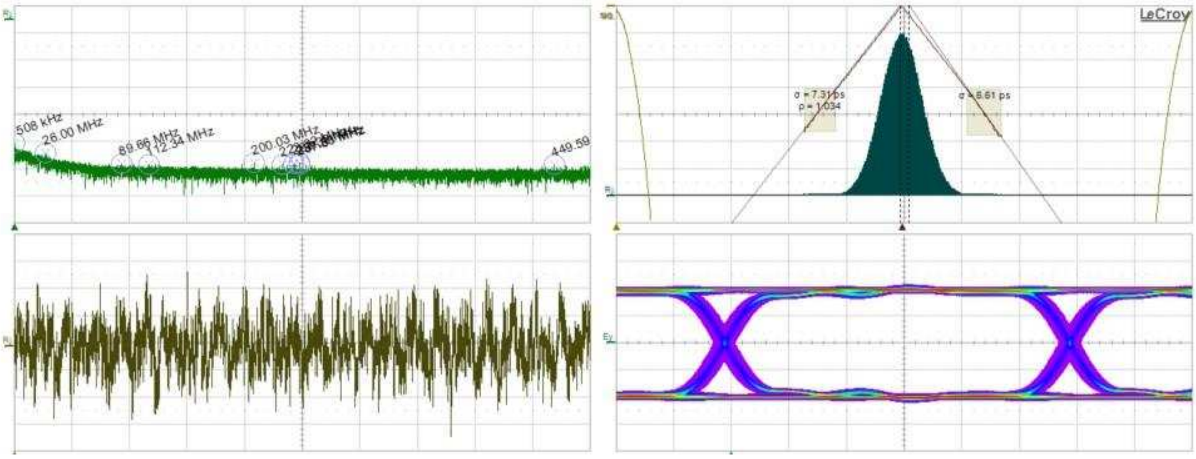


Figure 21: The scope tests of the optical output (an eye diagram) is presented. The rise and fall times measured at less than 240 ps.

`ADDR_REG_RW_DAQ_SLICE`. When multiple time slices are to be sent, the CMX loads the data from each consecutive time slice into the shift register, sending the data to the G-Link emulated chain without pause while keeping the DAV signal asserted. Each successive time slice is followed by its respective odd parity bits on each G-Link data input.

The CMX firmware includes a bunch-crossing number (BCN) counter that is synchronized by the BC reset signal from the TTC. When an L1A is received, the corresponding BCN is written to one of the readout FIFOs. The readout logic writes the BCN data to the shift register, but in a multi-slice event readout the same BCN is used for every slice.

When the transmission of all timeslices is completed, the readout logic de-asserts the G-Link DAV signal. This creates a so-called "DAV gap" which the ROD recognises as the end of the data for a given event. According to the protocol, the DAV is de-asserted for three clock cycles to allow the ROD to complete the current event processing and prepare for the next event.

## 7.3 Slice Readout for e/gamma and tau-counting CMX

The e/gamma and tau CMX use a 97-bit format as shown in Figure 22. This format includes some empty bit fields to maintain compability with the other CMX flavors. Bits 0 - 13 carry copies of the incoming backplane data from all CPMs, including parity errors. Bits 14 - 16 carry sub-sums from the three remote CMXs, with parity errors and overflow bits. G-link bits 17 and 18 carry the local sums and the total hit counts, and finally bit 19 carries the BC number, checksum, and CPM occupancy data. The relative timing between the G-link consecutive bits is defined by the register block: `ADDR_REG_RW_DAQ_RAM_RELATIVE_OFFSET`.

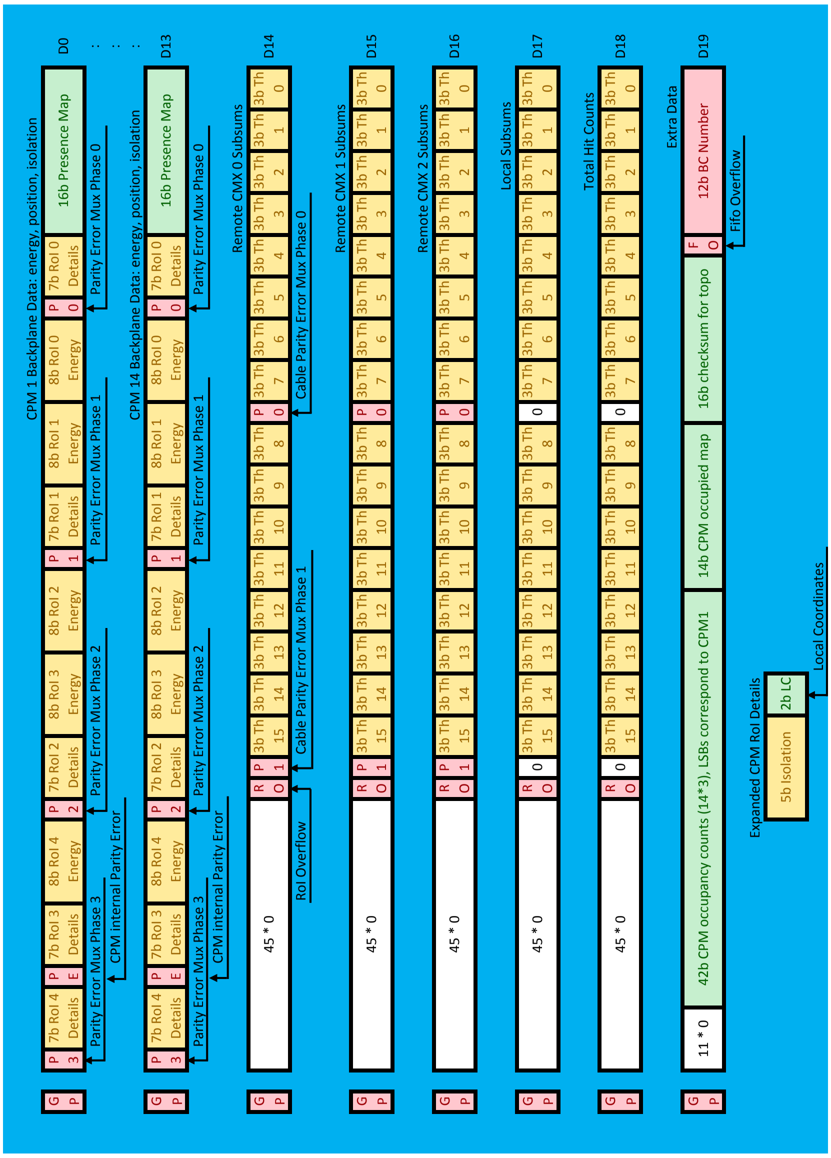


Figure 22: Serial readout format from Cluster-merging CMX.

## 7.4 Slice Readout for Jet-merging CMX

The 97-bit format used to read out the jet CMXs formatted is shown in Figure 23. G-Link bits 0-15 carry copies of the backplane data from all JEMs 0-15, with the parity errors. Bits 16 and 17 carry the remote jet count and local jet sums with the parity error, respectively. Bit 18 carries total main jets counts, with the parity. And bit 19 carries the BC number, checksum, and jet occupancy mask. All G-link bits include an odd-parity flag after the last data bit of each slice. The relative timing between the G-link consecutive bits is defined by the register block: `ADDR_REG_RW_DAQ_RAM_RELATIVE_OFFSET`.

## 7.5 Slice Readout for Energy-merging CMX

The format used for energy-merging CMX readout is ilustrated in Figure 24. G-Link bits 0-15 carry copies of the backplane data from JEMs 0-15, including parity error bits. G-Link bits 16 and 17 carry the remote and local energy sums, respectively, again with parity errors. G-Link bit 18 carries the total energy sums with the parity error, and bit 19 carries the BC number, overflow and the hit masks. All G-Link bits carry an odd-parity indicator after the last data bit of each slice. The relative timing between the G-link consecutive bits is defined by the register block: `ADDR_REG_RW_DAQ_RAM_RELATIVE_OFFSET`.

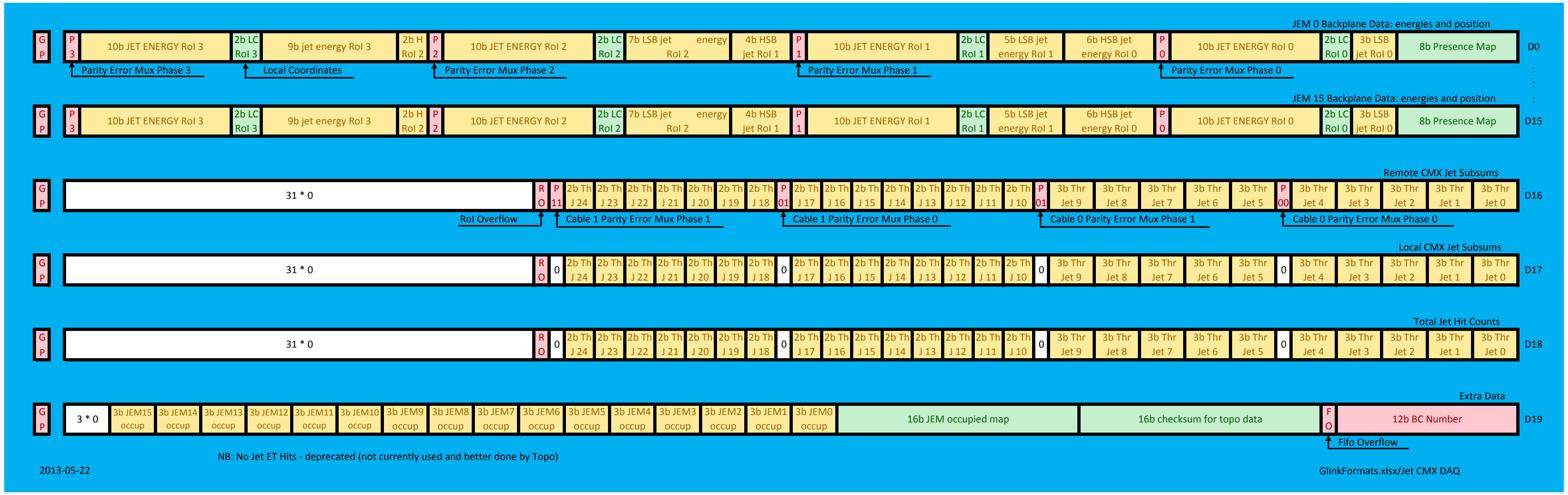

JEM 0 Backplane Data: energies and position
G P | P 3 | 10b JET ENERGY RoI 3 | 2b LC RoI 3 | 9b jet energy RoI 3 | 2b H RoI 2 | P 2 | 10b JET ENERGY RoI 2 | 2b LC RoI 2 | 7b LSB jet energy RoI 2 | 4b HSB jet RoI 1 | P 1 | 10b JET ENERGY RoI 1 | 2b LC RoI 1 | 5b LSB jet energy RoI 1 | 6b HSB jet energy RoI 0 | P 0 | 10b JET ENERGY RoI 0 | 2b LC RoI 0 | 3b LSB jet RoI 0 | 8b Presence Map | D0
Parity Error Mux Phase 3
Local Coordinates
Parity Error Mux Phase 2
Parity Error Mux Phase 1
Parity Error Mux Phase 0
:
:
:
JEM 15 Backplane Data: energies and position
G P | P 3 | 10b JET ENERGY RoI 3 | 2b LC RoI 3 | 9b jet energy RoI 3 | 2b H RoI 2 | P 2 | 10b JET ENERGY RoI 2 | 2b LC RoI 2 | 7b LSB jet energy RoI 2 | 4b HSB jet RoI 1 | P 1 | 10b JET ENERGY RoI 1 | 2b LC RoI 1 | 5b LSB jet energy RoI 1 | 6b HSB jet energy RoI 0 | P 0 | 10b JET ENERGY RoI 0 | 2b LC RoI 0 | 3b LSB jet RoI 0 | 8b Presence Map | D15
Remote CMX Jet Subsums
G P | 31 * 0 | R O | P 11 | 2b Th J 24 | 2b Th J 23 | 2b Th J 22 | 2b Th J 21 | 2b Th J 20 | 2b Th J 19 | 2b Th J 18 | P 01 | 2b Th J 17 | 2b Th J 16 | 2b Th J 15 | 2b Th J 14 | 2b Th J 13 | 2b Th J 12 | 2b Th J 11 | 2b Th J 10 | P 01 | 3b Thr Jet 9 | 3b Thr Jet 8 | 3b Thr Jet 7 | 3b Thr Jet 6 | 3b Thr Jet 5 | P 00 | 3b Thr Jet 4 | 3b Thr Jet 3 | 3b Thr Jet 2 | 3b Thr Jet 1 | 3b Thr Jet 0 | D16
RoI Overflow
Cable 1 Parity Error Mux Phase 1
Cable 1 Parity Error Mux Phase 0
Cable 0 Parity Error Mux Phase 1
Cable 0 Parity Error Mux Phase 0
Local CMX Jet Subsums
G P | 31 * 0 | R O | 0 | 2b Th J 24 | 2b Th J 23 | 2b Th J 22 | 2b Th J 21 | 2b Th J 20 | 2b Th J 19 | 2b Th J 18 | 0 | 2b Th J 17 | 2b Th J 16 | 2b Th J 15 | 2b Th J 14 | 2b Th J 13 | 2b Th J 12 | 2b Th J 11 | 2b Th J 10 | 0 | 3b Thr Jet 9 | 3b Thr Jet 8 | 3b Thr Jet 7 | 3b Thr Jet 6 | 3b Thr Jet 5 | 0 | 3b Thr Jet 4 | 3b Thr Jet 3 | 3b Thr Jet 2 | 3b Thr Jet 1 | 3b Thr Jet 0 | D17
Total Jet Hit Counts
G P | 31 * 0 | R O | 0 | 2b Th J 24 | 2b Th J 23 | 2b Th J 22 | 2b Th J 21 | 2b Th J 20 | 2b Th J 19 | 2b Th J 18 | 0 | 2b Th J 17 | 2b Th J 16 | 2b Th J 15 | 2b Th J 14 | 2b Th J 13 | 2b Th J 12 | 2b Th J 11 | 2b Th J 10 | 0 | 3b Thr Jet 9 | 3b Thr Jet 8 | 3b Thr Jet 7 | 3b Thr Jet 6 | 3b Thr Jet 5 | 0 | 3b Thr Jet 4 | 3b Thr Jet 3 | 3b Thr Jet 2 | 3b Thr Jet 1 | 3b Thr Jet 0 | D18
Extra Data
G P | 3 * 0 | 3b JEM15 occup | 3b JEM14 occup | 3b JEM13 occup | 3b JEM12 occup | 3b JEM11 occup | 3b JEM10 occup | 3b JEM9 occup | 3b JEM8 occup | 3b JEM7 occup | 3b JEM6 occup | 3b JEM5 occup | 3b JEM4 occup | 3b JEM3 occup | 3b JEM2 occup | 3b JEM1 occup | 3b JEM0 occup | 16b JEM occupied map | 16b checksum for topo data | F O | 12b BC Number | D19
Fifo Overflow
NB: No Jet ET Hits - deprecated (not currently used and better done by Topo)
2013-05-22
GlinkFormats.xlsx/Jet CMX DAQ

Figure 23: Serial readout format from Jet-merging CMX.

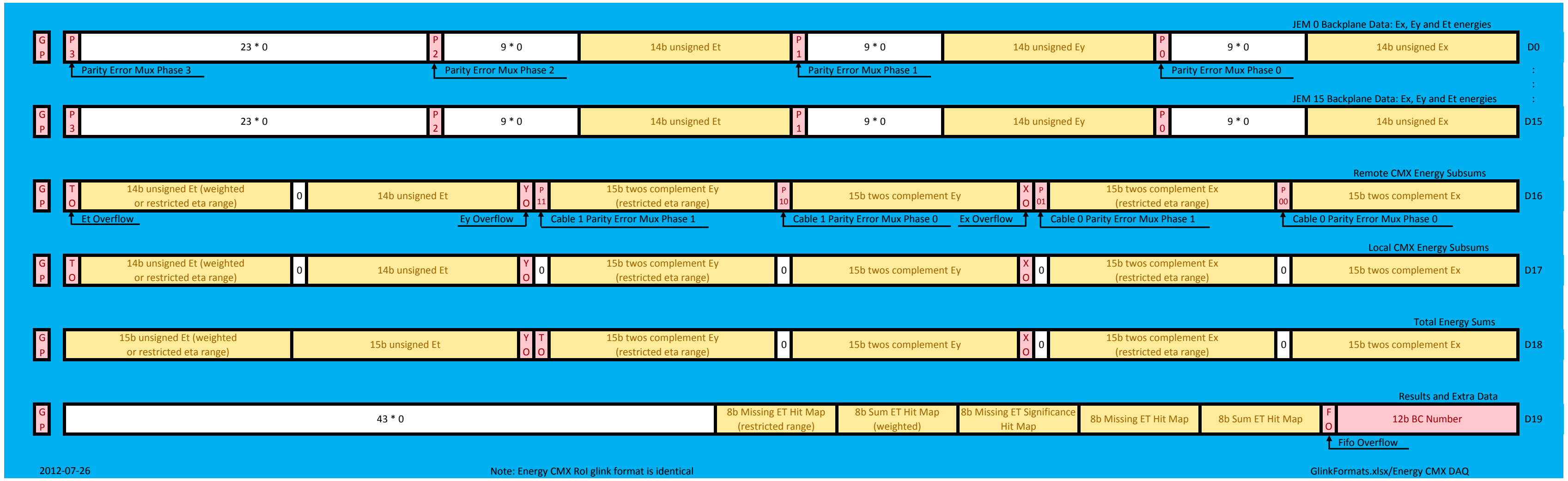

JEM 0 Backplane Data: Ex, Ey and Et energies
G P
P 3
23 * 0
P 2
9 * 0
14b unsigned Et
P 1
9 * 0
14b unsigned Ey
P 0
9 * 0
14b unsigned Ex
D0
Parity Error Mux Phase 3
Parity Error Mux Phase 2
Parity Error Mux Phase 1
Parity Error Mux Phase 0
JEM 15 Backplane Data: Ex, Ey and Et energies
D15
Remote CMX Energy Subsums
T O
14b unsigned Et (weighted or restricted eta range)
0
14b unsigned Et
Y O
P 11
15b twos complement Ey (restricted eta range)
P 10
15b twos complement Ey
X O
P 01
15b twos complement Ex (restricted eta range)
P 00
15b twos complement Ex
D16
Et Overflow
Ey Overflow
Cable 1 Parity Error Mux Phase 1
Cable 1 Parity Error Mux Phase 0
Ex Overflow
Cable 0 Parity Error Mux Phase 1
Cable 0 Parity Error Mux Phase 0
Local CMX Energy Subsums
D17
Total Energy Sums
15b unsigned Et (weighted or restricted eta range)
15b unsigned Et
D18
Results and Extra Data
43 * 0
8b Missing ET Hit Map (restricted range)
8b Sum ET Hit Map (weighted)
8b Missing ET Significance Hit Map
8b Missing ET Hit Map
8b Sum ET Hit Map
F O
12b BC Number
D19
Fifo Overflow
2012-07-26
Note: Energy CMX RoI glink format is identical
GlinkFormats.xlsx/Energy CMX DAQ

Figure 24: Serial readout format from Energy-merging CMX.

## 7.6 Buffer lengths and deadtime requirements

The buffers lengths and deadtime requirements are summarized below:

- Buffer length is 512 events
- Simple deadtime is equal to number of slices (1,3,5)
- Complex deadtime: 512/(96*nslices + 3)
- Max rate:
  - nslices = 1: 404 kHz
  - nslices = 2: 137 kHz
  - nslices = 3: 82 kHz

# 8 Timing domain

The CMX timing domains are summarized in the figure 25.

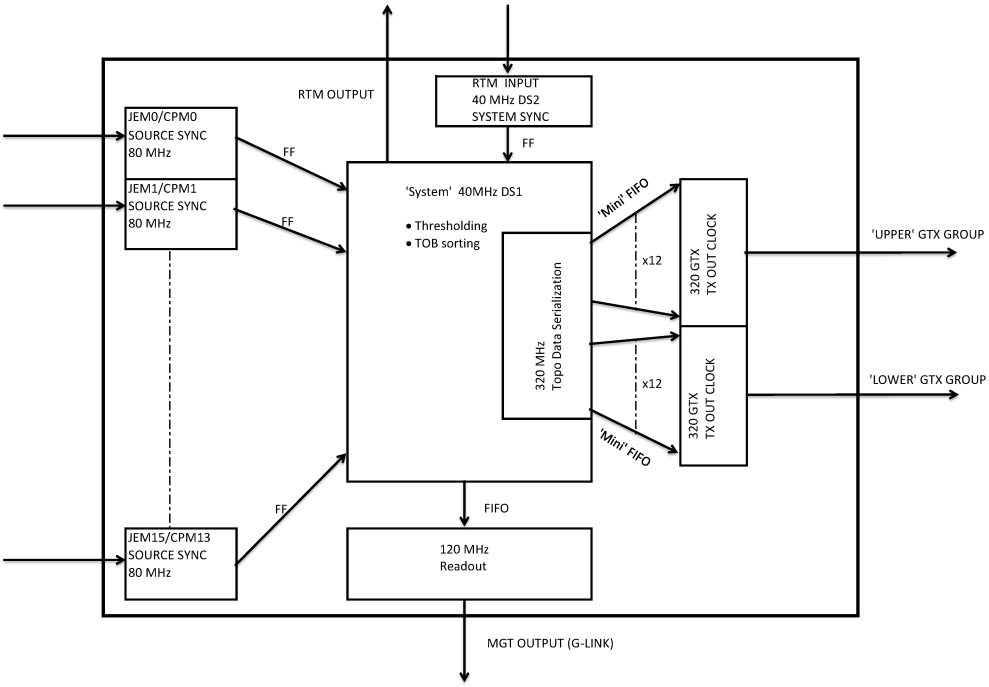


Figure 25: The CMX timing domains.

# 9 VME Interface

Similarly to the CMM, the CMX uses a subset of the VME standard, referred to as “VME–” and described in [4]. The subset provides only the signal lines necessary to execute A24, D24 cycles from a single crate master but without interrupts or arbitration. All cycles are A24, D16. VME interfacing, the address decoding and control register are implemented in FPGA. The CMX VME registers are listed and described in table 2.

# 10 Spy memory Capability

## 10.1 Spy memory functionality overview

VHDL component names:
`CMX_GENERIC_MODULE_SPY_MEM_CONTROL_FSM`
`CMX_INPUT_MODULE_SPY_MEM_CONTROL_FSM`
`CMX_MEMORY_SPY_INHIBIT`
`CMX_INPUT_MODULE_SPY_MEM_CONTROL_FSM`

The CMX modules provide several spy memories for capturing data at different points in the system. For backplane data, two sets are provided: one set capturing the data with the forwarded clocks and one set capturing the data in the system clock domain. Data sent out on the RTM and the CTP outputs are captured in respective, separate memory sets. RTM input data are captured in three spy memory sets: one in the forwarded clock domain, one in the deskew 2 clock domain and a third in the deskew 1 clock domain.

Each memory set provides the capability to capture data in real time, verify the data against a pre-loaded pattern, and play back pre-loaded data, with the exception of the forwarded-clock memories (for the the backplane and RTM input) which do not provide playback functionality. The backplane input memories clocked with the forwarded clocks have 512 addresses, with two addresses needed to capture the full information for an event. All other memories have 256 addresses, using one address per event. Data storage, playback or verification proceed in a cyclical fashion using port B of the dual-port memories, while Port A is used for the VME programming or readout.

Each memory has a corresponding `START_ADDRESS` register that defines the starting address of the playback cycle. The B address is reset to the value in this register when the “start playback” TTC command is received. Writing to all memories is synchronously inhibited upon assertion of bit 0 of the register `REG_RW_SPY_MEM_WRITE_INHIBIT`. This functionality is provided so that backplane, RTM and CTP interfaces can be correlated.

Writing to the spy memories is inhibited when a read or write operation from the VME interface is being performed.

## 10.2 Controlling operation mode, programming and reading the spy memories

From the user/software perspective, all spy memories are controlled identically. All memory controllers have `WORD`, `CONTROL` and `STATUS` ports mapped onto VME registers. The `CONTROL` register controls the operation mode and read/write requests to the memory. The three operation modes are as follows:

- SPY: a normal mode where data flows through the usual processing path and is also cyclically written to the memory so that it can be read out for monitoring or debugging purposes.
- PLAYBACK: data from upstream is ignored and instead data read cyclically from the memory is supplied to downstream processing
- VERIFY: data from upstream is compared to data read from memory and apropriate counters and error latches are set if required. Data from upstream is passed for downstream processing.

The mode of operation is controlled by bits 3 down to 0 of the control register. Possible values are `CONST_DPR_CONTROL_SPY`, `CONST_DPR_CONTROL_PLAYBACK`, `CONST_DPR_CONTROL_VERIFY`. Multiplexers controlling the data flow are set immediately upon a write to the control register. Bit 4 of the control register is the read/write request bit, and bit 5 specifies whether the request is for a read (0) or a write (1) operation. Bits 6 and higher of the control register are reserved.

The status register consists of two portions. The lower 12 bits are the address of the next read or write operation, with the lower 8 bits specifying the event number and bits 11 down to 8 specifying the channel number. The upper four bits give the status of the memory controller Finite State Machine. The status can be:

- `CONST_DPR_STATUS_NORMAL` : the operation requested by the bottom four control register bits s ongoing.
- `CONST_DPR_STATUS_WAIT` : read or write operation to memory is underway
- `CONST_DPR_STATUS_READ` : the memory has been read and the value is being presented on the VME interface, awaiting a read command from the word registers.
- `CONST_DPR_STATUS_WRITE` : the memory is ready for writing, awaiting a write command from the word registers.

To read out the source or system spy memory, the user procedure is as follows:

1. Verify that status is NORMAL
2. Write to the control register, setting bits 5 and 4 to “01”
   (a) At this point any ongoing reads and writes to the memory (for purpose of monitoring, playback or verification) will be suspended. Note that this may have undesirable consequences. For instance, if the control is set to play back the value that happened to be played back at the moment of read, initiation will be repeated. If the control was set to SPY normal data processing will not be interrupted, however acquisition of new events into memory will be stopped.
   (b) The status will change to WAIT
   (c) The data at channel 0, location 0 will be read and presented to VME WORD registers.
   (d) The status word will change to `CONST_DPR_STATUS_READ & 0x000`, signifying that data is ready to be read from VME, and that it corresponds to channel 0, event 0.
3. Read in order from the lowest to highest WORD registers.
   (a) Upon read from the highest WORD register the status will change to WAIT
   (b) The next memory location will be read and data presented to VME
   (c) The status will return to READ, and the lower 12 address bits will be incremented by one.
4. Verify that the status is READ and address is as expected.
   (a) If the state is WAIT sleep and check again with some reasonable timeout.
   (b) Unexpected states or address locations must be reported.
5. Repeat the last two steps until the whole memory has been read.
   (a) Upon reading the highest WORD register when the status register is READ and the lower bits of the register point to the last address, the status will return to NORMAL and operation prescribed by the control register will resume.
6. Verify that the status is NORMAL

To write to the memory a similar sequence is performed:

1. Verify that status is NORMAL
2. Write to the control register setting bits 5 and 4 to “11”
   (a) As with read operation, at this point any ongoing reads and writes to the memory will be suspended.

   (b) The status will change to `CONST_DPR_STATUS_WRITE & 0x000`, signifying that the controller is waiting for user to write data to the WORD registers and that data will be written to memory of channel 0, event 0.

3. Write in order to the WORD registers, from lowest to highest
   (a) Upon write to the highest WORD register the status will change to WAIT
   (b) Data written to VME will be written to memory location.
   (c) Status will return to WRITE and the lower 12 address bits will be incremented by one.

4. Verify that the status is WRITE and address is as expected.
   (a) If the state is WAIT sleep and check again with some reasonable timeout.
   (b) Unexpected states or address locations must be reported.

5. Repeat the last two steps till the whole memory has been written.
   (a) Upon writing the highest WORD address when the status register is WRITE with the address pointing to the highest location in the memory the status will change to WAIT; last event will be written and the status will return to NORMAL and operation prescribed by the control register will resume.

6. Verify that the status is NORMAL

# 11 Error Handling

VHDL component name:
`CMX_INPUT_MODULE`
`CMX_SYSTEM_CABLE_INPUT_MODULE`

In order to control the data transmission integrity, a parity bit is generated and sent with the data. At the receiving end, another parity bit is generated based on the received data and compared with the parity bit sent by the transmitter. If the parity bit does not match, an error has occured during transmission and such data are labelled with the parity error flag set to 1. Two modules are obliged to perform the data transmission integrity check: Input Module and Rear Transition Module. According to the protocol, for a each event, the Input Module checks the parity bit sent with the data transmitted by all the JEMs or CPMs, and if the mismatch is found for at least single channel, the parity error bit is set to '1'. The Rear Transition Module performs the same checks based on the data sent by the CMX "crate" module. Two different parity error flags are introduced to distinguish the transmission error, one corresponds to the Input Module, while the second to Rear Transition Module. The adder real-time sums are saturated if there is parity error with the force flag set to 1. The force flag is controled by the register: `ADDR_REG_RW_QUIET_FORCE`. In addition to that, the CMX is equipped with a group of 32-bit parity error counters, these counters are accessible at the register: `ADDR_REG_PARITY_ERROR_COUNTER` and `ADDR_REG_RTM_PARITY_ERROR_COUNTER`.

# 12 Rate Metering Capability

## 12.1 Jet Rate Metering

VHDL component name: `ADDER_TOP_VS`

The Jet CMX is equipped with type-dependent jet rate metering modules. The crate-level hit metering component includes a set of twenty-five 32-bit counters, one for each of the available jet thresholds. The counters are in the same order as the hit bins, beginning with the ten 3-bit bins followed by the fifteen 2-bit bins. These counters are accessible at the register block: ADDR_REG_RO_MULT_LOCAL_COUNTER.

The System hit metering component contains three sets of twenty-five 32-bit counters. These sets correspond to the intermediate hit sums, local hit sums and final system-wide hit sums, respectively. These counters are accessible at the following register blocks:
ADDR_REG_RO_MULT_LOCAL_COUNTER,
ADDR_REG_RO_MULT_REMOTE_COUNTER,
ADDR_REG_RO_MULT_TOTAL_COUNTER.

The counters are controlled by two flags: Inhibit and Reset. These flags are defined by the following registers: ADDR_REG_RW_RATE_COUNTER_INHIBIT and ADDR_REG_RW_RATE_COUNTER_RESET. The Inhibit flag stops the counting process, while the Reset zeros the counters. Inputs from a given JEM are not counted if that input is disabled in the Backplane Disable Register.

## 12.2 Cluster Rate Metering

VHDL component name: ADDER_TOP_VS_CP

The CP CMX monitors the rate of cluster hits for all thresholds. The cluster rate metering modules are divided into crate and system groups. The Crate hit metering component contains sixteen 32-bit counters. each corresponding to the sixteen respective cluster thresholds. These counters are accessible at the register block: ADDR_REG_RO_MULT_LOCAL_COUNTER.

The System hit metering component contains three sets of sixteen 32-bit counters. These sets correspond to the intermediate cluster hit sums, local cluster hit sums and the system-wide cluster hits, respectively. These counters are accessible at the following register blocks:
ADDR_REG_RO_MULT_LOCAL_COUNTER,
ADDR_REG_RO_MULT_REMOTE_COUNTER,
ADDR_REG_RO_MULT_TOTAL_COUNTER.

As for the jet metering, Inhibit and Reset flags are used to control the counters, and are set by the following registers:
ADDR_REG_RW_RATE_COUNTER_INHIBIT and ADDR_REG_RW_RATE_COUNTER_RESET. The Inhibit flag stops the counting process, while the Reset zeros the counters. In both cases, the input from a particular CPM module will not be counted if it is disabled in the Backplane Disable Register.

## 12.3 Energy Rate Metering

VHDL component name: CMX_SumEt

The Energy CMX owns five groups of eight 32-bit counters. The five groups represent, respectively, the 8-bit Missing ET siginificance hit map, 8-bit Missing ET hit map, 8-bit sum ET hit map, 8-bit Missing ET hit map (restricted range), and the 8-bit sum ET hit map from each JEM. Each group consist of eight 32-bit counters to count each hit map separately. These counters are accessible at the following register block:
ADDR_REG_RO_SUM_ET_COUNTER,
ADDR_REG_RO_MISSING_ET_COUNTER,
ADDR_REG_RO_MISSING_ET_SIGN_COUNTER,
ADDR_REG_RO_SUM_ET_WEIGHTED_COUNTER,
ADDR_REG_RO_MISSING_ET_RES_COUNTER.

There is also a pair of Inhibit and Reset flags in the system to control the counters, which are defined by the following registers:

ADDR_REG_RW_RATE_COUNTER_INHIBIT and ADDR_REG_RW_RATE_COUNTER_RESET.
The Inhibit flag stops the counting process, while the Reset zeros the counters. In both cases, the input from a particular CPM module will not be counted count if it is disabled in the Backplane Disable Register.

## 12.4 TOBs Rate Metering

VHDL component name: DECODER

The CP and Jet CMX have the capability to monitor the number of Trigger Objects (TOBs) reported by each JEM or CPM, based on the number of presence bits set by each module. The number of TOB bits for each module is summed added to a 32-bits counter indexed at the register block ADDR_REG_RO_TOB_COUNTER.

The CP CMX is equipped with a group of 14 counters, while the JET CMX has 16. As with other rate monitors, Inhibit and Reset flags are also used to control the counters.

# 13 System Monitor

VHDL component name: SYS_MONITOR

During normal operation it's important to monitor several FPGA operating parameters. The Virtex-6 FPGA is equipped with a single System Monitor which is based on a 10-bit, 200kSPS (kilo-samples per second) Analog-to-Digital Converter (ADC) and on-chip sensors [8]. The main purpose is to measure FPGA operating parameters, for example the on-chip power supply and die temperatures. Additionally, access to external voltages is also provided. This is done through a analog-input pair (Vp/Vn) and 16 user-selectable analog inputs, known as auxiliary inputs (Vaux[15:0],Vaux[15:0]). The external analog input allow to monitor the physical environment of the board.

The CMX logic allows to access the following operating parameters like on-chip temperature T, average on-chip VccINT and VccAUX. These parameters are accessible at the register block: ADDR_REG_RO_SYSMON_DATA_BLOCK.

# 14 Structure of the repository

The CMX repository has the following structure. The common IP cores, modules (.vhd) and package for all the CMX types (CP, Jet, SumEt) and variants (Crate and System) are located in the directories: Common/trunk/ipcore_dir and Common/trunk/sources respectively. The IP cores and modules (.vhd) specific to the CMX type are placed in a seperate svn directories: CP, Jet and SumEt. The svn directory for each CMX type has subdirectories to archive the project files seperately for the crate and system variants. The common IP cores and modules (.vhd) to the CMX type are archived in the directories: Type_common/trunk/ipcore_dir and Type_common/trunk/sources respectively.

- Common
  - trunk
    - ipcore_dir ← Common cores
    - sources ← Common modules (.vhd)
- Type
  - Type_common
    - trunk
      - ipcore_dir ← Cores common to the type

- sources ← Modules (.vhd) common to the type
- Type_crate
  - trunk
    - sources ← Modules (.vhd) specific to variant, Top modules and project files
- Type_system
  - trunk
    - sources ← Modules (.vhd) specific to variant, Top modules and project files

The CMX repository can be accessed here:
svn+ssh://svn.cern.ch/reps/atl1calo/firmware/CMX/CMX_project/Base

# References

[1] Raffaele Giordano Alberto Aloisio Francesco Cevenini and Vincento Izzo. *Emulating the GLink Chip-Set with FPGA Serial Transceivers in the ATLAS Level-1 Muon Trigger.*

[2] Kenneth E. Batcher. "Sorting Networks and their Applications". In: *Spring Joint Computer Conference, AFIPS Proc.* 32 (1968), pp. 307–314.

[3] Yuri Ermoline Chip Brock Dan Edmunds Philippe Laurens and Wojtek Fedorko. "Atlas Level 1 Calorimeter Trigger Common Merger eXtended module (CMX), Hardware Description, Document prepared for the CMX Production Readiness Review, Michigan State University, USA". In: (2014).

[4] C.N.P Gee. "VME– specification, version 1.2, ATLAS Level-1 Calorimeter Trigger". In: (2006).

[5] P. Koopman and T. Chakravarty. "Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks". In: *Institute for Software Research. Paper 682.* (2004).

[6] R. Sedgewick. *Algorithms in Java, Parts 1-4 (3rd Edition) (Pts.1-4)*. Chap. 11.2.

[7] XILINX. "Virtex-6 FPGA Clocking Resources, User Guide". In: (2014).

[8] XILINX. "Virtex-6 FPGA System Monitor, User Guide". In: (2014).

| Register Name | Address (Hex) | Description |
|---|---|---|
| ADDR_REG_RO_TEST | 00100 | Test RO register |
| ADDR_REG_RW_TEST | 00102 | Test RW register |
| ADDR_REG_RO_BACKPLANE_FORWARD | 00104 | 32 address for forwarding the backplane data, for test purpose only |
| ADDR_REG_RW_IDELAY_BACKPLANE | 00144 | 400 delay values |
| ADDR_REG_RO_EV_COUNTER | 00464 | 32-bit event counter for the input module maxes out but doesn't overturn |
| ADDR_REG_RO_PARITY_ERROR_COUNTER | 00468 | 16 32-bit parity error counters for the input module, these counters max out but don't overturn |
| ADDR_REG_RW_COUNTER_RESET | 004A8 | Counter reset; resets counters of events and errors |
| ADDR_REG_RW_INPUT_SPY_MEM_SOURCE_WORD | 004AA | Six consecutive addresses |
| ADDR_REG_RW_INPUT_SPY_MEM_SOURCE_CONTROL | 004B6 | Input SPY memory control and status words with states defined below:<br>• CONST_DPR_CONTROL_SPY(0001)<br>• CONST_DPR_CONTROL_PLAYBACK(0010)<br>• CONST_DPR_CONTROL_VERIFY(0011)<br>• CONST_DPR_STATUS_NORMAL(001)<br>• CONST_DPR_STATUS_WAIT_INHIBIT(010)<br>• CONST_DPR_STATUS_WAIT_READ(011)<br>• CONST_DPR_STATUS_WAIT_WRITE(100)<br>• CONST_DPR_STATUS_WRITE(101)<br>• CONST_DPR_STATUS_READ(110) |
| ADDR_REG_RW_INPUT_SPY_MEM_SOURCE_START_ADDRESS | 004BA | 16 9-bit words (upper 7 bits unused), ending address 4D8 |
| ADDR_REG_RO_INPUT_SPY_MEM_SOURCE_CHECK_ERROR | 004DA | 16 24-bit error latches |
| ADDR_REG_RO_INPUT_SPY_MEM_SOURCE_NOERROR_COUNTER | 0051A | 16 32-bit counters that stop when a first bit error on any bit is encountered |
| ADDR_REG_RW_CLOCK_MANAGER_RESET | 0055A | System MMCM reset register; writing '1' to LS bit causes MMCM to be held in reset state (so write a '0' right after) |
| ADDR_REG_RO_IDELAYCTRL_RDY | 0055C | Status of the io delay control circuits |
| ADDR_REG_RO_IDELAYCTRL_RST | 0055E | Current IDELAYCTRL reset status |
| ADDR_REG_RO_IDELAYCTRL_WAS_RST | 00560 | States if IDELAYCTRL has been reset |
| ADDR_REG_RW_INPUT_MOD_RESET | 00562 | async register for controlling the reset of the input module |
| ADDR_REG_RO_INPUT_MOD_COUNTER_ENABLE | 00564 | counter enable values for the input module |
| ADDR_REG_RW_CTP_TESTER_DATA_SELECT | 00566 | CTP tester data source select LSB=0 means predefined pattern, LSB=1 pseudo-random |
| ADDR_REG_RW_TOPOTR_GTX_RESET | 00568 | TOPO TR GTX RX (bit 0) TX (bit 1) reset |
| ADDR_REG_RW_RX_POLARITY | 0056A | 3 consecutive addresses for the RX polarity (RX is not used on Basefunction but the functionality in FW is provided |
| ADDR_REG_RW_TX_POLARITY | 00570 | 3 consecutive addresses for the TX polarity |
| ADDR_REG_RW_JET_THRESHOLD_BLOCK | 00576 | CMX Jet threshold definition: 25 THR x 16 JEMs x 2 eta local positions x 2 fine positions = 1600 registers |
| ADDR_REG_RW_DAQ_SLICE | 011F6 | Define how many "slices" to read out with the event causing the L1A being the middle "slice" only two LSB are used 00: one slice; 01: three slices; 10: five slices |
| ADDR_REG_RW_DAQ_RAM_OFFSET | 011F8 | How far back in the buffer memory we need to look to find the event that has caused the L1A we are currently getting |
| ADDR_REG_RW_BCID_RESET_VAL | 011FA | What is the value of the BCID we are get when the BC reset is received |
| ADDR_REG_RW_INPUT_SPY_MEM_SYSTEM_WORD | 011FE | Six consecutive SPY memory addresses ending at 1208 |
| ADDR_REG_RW_INPUT_SPY_MEM_SYSTEM_CONTROL | 0120A | Control and status words |
| ADDR_REG_RO_INPUT_SPY_MEM_SYSTEM_STATUS | 0120C | Control and status words |
| ADDR_REG_RW_INPUT_SPY_MEM_SYSTEM_START_ADDRESS | 0120E | One 8-bit words (upper 8 bits unused) |
| ADDR_REG_RO_INPUT_SPY_MEM_SYSTEM_CHECK_ERROR | 01210 | 16 24-bit error latches. Last address is 124E |
| ADDR_REG_RO_INPUT_SPY_MEM_SYSTEM_NOERROR_COUNTER | 01250 | 16 32-bit counters that stop when a first bit error on any bit on any channel is encountered. Last address is 128E |
| ADDR_REG_RW_SPY_MEM_WRITE_INHIBIT | 01290 | Writing 1 to LSB triggers a synchroneous lock on writing to the spy memories. This is to for reading out different memories as a snapshot of activity from entire CMX |
| ADDR_REG_RW_CTP_OUTPUT_COUNTER_RESET | 01292 | Access to this address will reset the error counters and latches in the CTP output module |
| ADDR_REG_RW_CTP_SPY_MEM_WORD | 01294 | CTP Spy memory data access four addresses starting at this one |
| ADDR_REG_RW_CTP_SPY_MEM_CONTROL | 0129C | Control word word for the CTP spy memory |
| ADDR_REG_RO_CTP_SPY_MEM_STATUS | 0129E | Status word for the CTP spy memory |
| ADDR_REG_RW_CTP_SPY_MEM_START_ADDRESS | 012A0 | Start address for the CTP Spy memory |
| ADDR_REG_RO_CTP_SPY_MEM_CHECK_ERROR | 012A2 | Error latches for the CTP Spy memory there are 4 addresses |
| ADDR_REG_RO_CTP_SPY_MEM_NOERROR_COUNTER | 012AA | 32-bit no error counter for the CTP spy |
| ADDR_REG_RW_RTM_INPUT_COUNTER_RESET | 012AE | reset for the RTM input module counters |
| ADDR_REG_RW_RTM_SPY_SOURCE_MEM_WORD | 012B0 | data words for the RTM 'SOURCE' SPY memory 4 words |
| ADDR_REG_RW_RTM_SPY_SOURCE_MEM_CONTROL | 012B8 | Control register |
| ADDR_REG_RO_RTM_SPY_SOURCE_MEM_STATUS | 012BA | Status register |
| ADDR_REG_RW_RTM_SPY_SYSTEMDS2_MEM_WORD | 012BC | Control register |
| ADDR_REG_RW_RTM_SPY_SYSTEMDS2_MEM_CONTROL | 012C4 | Control register |
| ADDR_REG_RO_RTM_SPY_SYSTEMDS2_MEM_STATUS | 012C6 | Control register |
| ADDR_REG_RW_RTM_SPY_SYSTEM_MEM_WORD | 012C8 | Control register |
| ADDR_REG_RW_RTM_SPY_SYSTEM_MEM_CONTROL | 012D0 | Control register |
| ADDR_REG_RO_RTM_SPY_SYSTEM_MEM_STATUS | 012D2 | Control register |

| Register Name | Address (Hex) | Description |
|---|---|---|
| ADDR_REG_RW_RTM_SPY_SOURCE_MEM_START_ADDRESS | 012D4 | Max three addresses for the start pointers (since there are max 3 input rtm cables) for the source memory |
| ADDR_REG_RW_RTM_SPY_SYSTEM_MEM_START_ADDRESS | 012DA | One register each for the start adresses for the system and ds2 rams |
| ADDR_REG_RW_RTM_SPY_SYSTEMDS2_MEM_START_ADDRESS | 012E0 | Control register |
| ADDR_REG_RO_RTM_PARITY_ERROR_COUNTER | 012E6 | Parity error counters; each counter is 32-bits (so 2 addresses) and there are max three counters, (there may be 3 rtm cables coming in) so reserve six addresses total |
| ADDR_REG_RO_RTM_SPY_SOURCE_MEM_CHECK_ERROR | 012F2 | Bit error latches for RTM source 2 bytes per latch and possibly 3 cables so six addresses total |
| ADDR_REG_RO_RTM_SPY_SYSTEMDS2_MEM_CHECK_ERROR | 012FE | Bit error latches for RTM source |
| ADDR_REG_RO_RTM_SPY_SYSTEM_MEM_CHECK_ERROR | 0130A | Bit error latches for RTM source |
| ADDR_REG_RO_CLOCK_MANAGER_STATUS | 01316 | PLL status (bit 0 is DS1 MMCM lock and bit 1 is for DS2 MMCM) |
| ADDR_REG_RW_DELAY_INPUT_DATA_ADDER | 01318 | Delay for the input adder data |
| ADDR_REG_RW_BACKPLANE_INPUT_CHANNEL_MASK | 0131A | Channel mask bit q set to '1' will zero out data from input q parity will be set to 1 in the data presented to downstream logic |
| ADDR_REG_RW_RTM_OUTPUT_COUNTER_RESET | 0131C | Error counter reset register for the RTM output module |
| ADDR_REG_RW_RTM_OUTPUT_SPY_SYSTEM_MEM_WORD | 0131E | 4 addresses for the RTM output spy words |
| ADDR_REG_RW_RTM_OUTPUT_SPY_SYSTEM_MEM_CONTROL | 01326 | Output RTM spy control and status |
| ADDR_REG_RO_RTM_OUTPUT_SPY_SYSTEM_MEM_STATUS | 01328 | Output RTM spy control and status |
| ADDR_REG_RO_RTM_OUTPUT_SPY_SYSTEM_MEM_CHECK_ERROR | 0132A | Bit error latch; six addresses (two per cable x max 3 cables) |
| ADDR_REG_RW_RTM_OUTPUT_SPY_SYSTEM_MEM_START_ADDRESS | 01336 | Start address for the RTM spy memory |
| ADDR_REG_RW_DAQ_RAM_RELATIVE_OFFSET | 01338 | Beginning adresses for the relative DAQ offsets (19 addresses) |
| ADDR_REG_RW_SUMET_MASK | 0135E | Mask for the restricted range sum ET |
| ADDR_REG_RW_MISSET_MASK | 01360 | Mask for the restricted range missing ET |
| ADDR_REG_RO_CLOCK_DETECT_COUNTER | 01362 | Clock detect counters |
| ADDR_REG_RW_QUIET_FORCE | 01382 | control for quiet/force accept behavior bit 0 set will cause a force accept in case of a parity error on the backplane or on an RTM cable bit 1 set will cause data from problematic input module/cable to be zeroed |
| ADDR_REG_RW_RTM_INPUT_CHANNEL_MASK | 01384 | mask for the RTM inputs |
| ADDR_REG_RO_DAQ_ROI_STATUS | 01386 | Status of the GTX's driving the DAQ and ROI SFPs, meaning of bits:<br>• 0: local_pll_lock_out_SFP_DAQ<br>• 1: GTX_TX_READY_OUT_TX_SFP_DAQ<br>• 2: GTX_RX_READY_OUT_TX_SFP_DAQ<br>• 3: PLLLKDET_diag_TX_SFP_DAQ<br>• 4: local_pll_lock_out_SFP_ROI<br>• 5: GTX_TX_READY_OUT_TX_SFP_ROI<br>• 6: GTX_RX_READY_OUT_TX_SFP_ROI<br>• 7: PLLLKDET_diag_TX_SFP_ROI<br>• 8: readout_rst_out |
| ADDR_REG_RW_DAQ_ROI_GTX_RESET | 01388 | Bit 0 reset DAQ, bit 1 resets ROI GTX (should be done at the same time) |
| ADDR_REG_RO_TOPOTR_GTX_STATUS | 0138A | Bit 0 gives status of RX; bit 1 gives status of TX |
| ADDR_REG_RW_RATE_COUNTER_INHIBIT | 0138C | Bit 0 set stops the rate counters synchroneously |
| ADDR_REG_RW_RATE_COUNTER_RESET | 0138E | Bit 0 set resets (and holds reset) the rate counters |
| ADDR_REG_RO_RATE_NORMALISATION_COUNTER | 01390 | 32 bit rate normalisation counter (2 addresses) |
| ADDR_REG_RO_MULT_LOCAL_COUNTER | 01394 | 25 32-bit local rate counters (50 addresses), last one is 13F6 |
| ADDR_REG_RO_MULT_REMOTE_COUNTER | 013F8 | 25 32-bit remote rate counters (96 addresses), last one is 14B8, 96 addresses because CP has 16 thresholds x3 cables x2 words/counter |
| ADDR_REG_RO_MULT_TOTAL_COUNTER | 014BA | 25 32-bit global rate counters (50 addresses), last one is 151C |
| ADDR_REG_RO_TOB_COUNTER | 0151E | 16 32-bit tob counters (32 adresses), last one is 155C |
| ADDR_REG_RO_LOCAL_BACKPLANE_OVERFLOW_COUNTER | 0155E | 16 32-bit local backplane overflow counters, last one is 159C |
| ADDR_REG_RO_GLOBAL_BACKPLANE_OVERFLOW_COUNTER | 0159E | 32-bit global backplane overflow counter, 2 addresses |
| ADDR_REG_RO_TOTAL_OVERFLOW_COUNTER | 015A2 | 32-bit total (backplane OR number of TOBs) overflow counter, 2 adresses |
| ADDR_REG_RO_BC_RESET_ERROR_COUNTER | 015A6 | 16-bit counter of BCID errors, counter is incremented when upon the reception of bc reset the counter is not at the expected value |
| ADDR_REG_RW_BC_RESET_ERROR_COUNTER_RESET | 015A8 | Hold bit 0 high to reset the BCRESET error counter |
| ADDR_REG_RO_SUM_ET_COUNTER | 015AA | 8 32-bit sum ET threshold counters, 16 adresses, last one 15C8 |
| ADDR_REG_RO_MISSING_ET_COUNTER | 015CA | 8 32-bit missing ET threshold counters, 16 adresses, last one 15E8 |
| ADDR_REG_RO_MISSING_ET_SIGN_COUNTER | 0160A | 8 32-bit missing ET significance threshold counters, 16 adresses, last one 1628 |
| ADDR_REG_RO_SUM_ET_WEIGHTED_COUNTER | 0162A | 8 32-bit weighted/restricted sum ET threshold counters, 16 adresses, last one 1648 |
| ADDR_REG_RO_MISSING_ET_RES_COUNTER | 0164A | 8 32-bit weighted/restricted missing ET threshold counters, 16 adresses, last one 1668 |
| ADDR_REG_RO_PRESENCE_COUNTER | 0166A | 224 32-bit counters, 448 addresses, last one is 19EA |
| ADDR_REG_RW_DISABLE_OVERFLOW_MASK | 019EC | Mask for overflow behavior, set any of bits (1-14(CP),0-15(JET)) to disable trigger force on overflow discovery from corresponding source module |
| ADDR_REG_RO_SYSMON_DATA_BLOCK | 019EE | Reserve 16 addresses for the sysmon, in first implementation 15 are used |
| ADDR_REG_RW_MISS_E_THR_BLOCK | 01A0E | 8 31-bit MISS_E THRESHOLDS, 16 address, last one 1A2C |

| Register Name | Address (Hex) | Description |
|---|---|---|
| ADDR_REG_RW_MISS_E_RES_THR_BLOCK | 01A2E | 8 31-bit MISS_E_RES THRESHOLDS, 16 address, last one 1A4C |
| ADDR_REG_RW_SUM_ET_THR_BLOCK | 01A4E | 8 15-bit SUM_ET THRESHOLDS, 8 address, last one 1A5C |
| ADDR_REG_RW_SUM_ET_RES_THR_BLOCK | 01A5E | 8 15-bit SUM_ET_RES THRESHOLDS, 8 address, last one 1A6C |
| ADDR_REG_RW_XS_T2_A2_THR_BLOCK | 01A6E | 8 31-bit XS_T2_A2 THRESHOLDS, 16 address, last one 1A8C |
| ADDR_REG_RW_T_MISS_E_MIN_PARAM_BLOCK | 01A8E | 8 31-bit T_MISS_E_MIN PARAM, 16 address, last one 1AAC |
| ADDR_REG_RW_T_MISS_E_MAX_PARAM_BLOCK | 01AAE | 8 31-bit T_MISS_E_MAX PARAM, 16 address, last one 1ACC |
| ADDR_REG_RW_T_SUM_E_MIN_PARAM_BLOCK | 01ACE | 8 15-bit T_SUM_E_MIN PARAM, 8 address, last one 1ADC |
| ADDR_REG_RW_T_SUM_E_MAX_PARAM_BLOCK | 01ADE | 8 15-bit T_SUM_E_MAX PARAM, 8 address, last one 1AEC |
| ADDR_REG_RW_XS_B2_PARAM_BLOCK | 01AEE | 8 15-bit XS_B PARAM, 8 address last one 1AFC |
| ADDR_REG_RO_CLOCK_DIFF_DETECT_COUNTER | 01AFE | 32 bit clock difference counters for each backplane input the counter counts the difference between the system clock ticks and forwarded clock ticks the clock reset (REG_RW_COUNTER_RESET) must be used periodically, otherwise the counters will top-out and the diference will be 0 irrespective of the clock behavior, last one is 1B3C |
| ADDR_REG_RO_CMX_FLAVOR | 04998 | Cmx flavor number value of the register corresponds to the 'revision' number on the CF (using v5.4 of the BSPT FW): 2=EM/TAU Crate; 3=EM/TAU System; 4=SumET Crate; 5=SumET System; 6=Jet Crate; 7=Jet System; 0,1 are 'special |
| ADDR_REG_RO_VERSION_COMMON | 05000 | Version register |
| ADDR_REG_RO_VERSION_FLAVOR_COMMON | 05004 | Version register |
| ADDR_REG_RO_VERSION_FLAVOR_LOCAL | 05008 | Version register |

Table 2: VME register map