Specification of the MCEN Modules

Emanuel T. Machado, John M. Butler

Version 0.2.16, 11/18/96 5:22 PM

Overview of the MCEN

The Muon Centroid (MCEN) cards form a set of VME cards used to form mini-track segments ("centroids") from raw MDT hits as part of the Level 1 (L1) muon trigger decision for Run II of D0. A block diagram of the Muon Centroid (MC) VME crate is shown in Figure 1. There are **four** such crates in the system with a total of 60 MCEN modules and they are physically located on the detector platform. Much of the functionality of the MCEN cards is the same as the MTC05 and MTC10 cards [1].

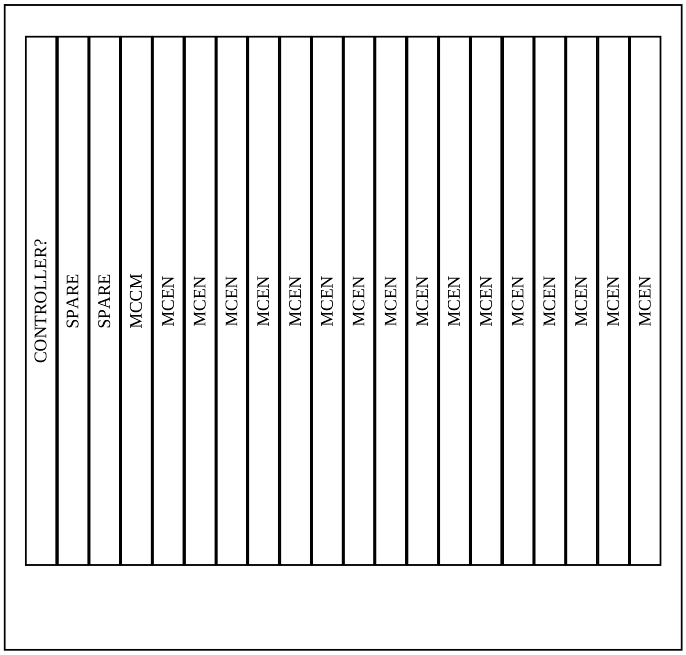

Figure 1 – The Centroid Crate

Inputs to the MCEN cards come from the Mini Drift Tube (MDT) front end electronics. Each MCEN card forms centroids from MDT hits within one octant of a given layer. The centroids are then transmitted to MTC10 cards located in a Muon Trigger crate on the platform.

There are two types of input signals received by the MCEN card from the outside world: **hit** information from MDT front end electronics and **timing** trigger signals from the MCCM. Each MCEN card can accept hit information from **12** high speed serial links. This data is assumed to be transmitted over RG-58 cables. The timing and trigger signals originate at the Trigger Framework (TF). They are distributed to geographic sectors (defined as a VME crate containing a VME buffer driver, VBD) over high speed Serial Command Links (SCLs).

The particular geographic sectors for the muon front end and L1 trigger system are located in the 3rd floor moveable counting house (MCH) and contain a muon fanout card (MFC) and muon readout cards (MRCs) in addition to the VBD. The MFC contains an SCL receiver which decodes the timing and trigger signals and distributes them to the MRC cards over the VME backplane. These timing and trigger signals are sent over copper (coax cable for timing signals and 50 pin twist and flat ribbon cable for trigger information) to the MCCM. The MCCM subsequently distributes them to the MCEN cards over the VME backplane.

The output of each MCEN card is up to 4 SCLs of centroid information passed on to one MTC10.

The relationship between the MCEN and the MCCM can be seen in Figure 2:

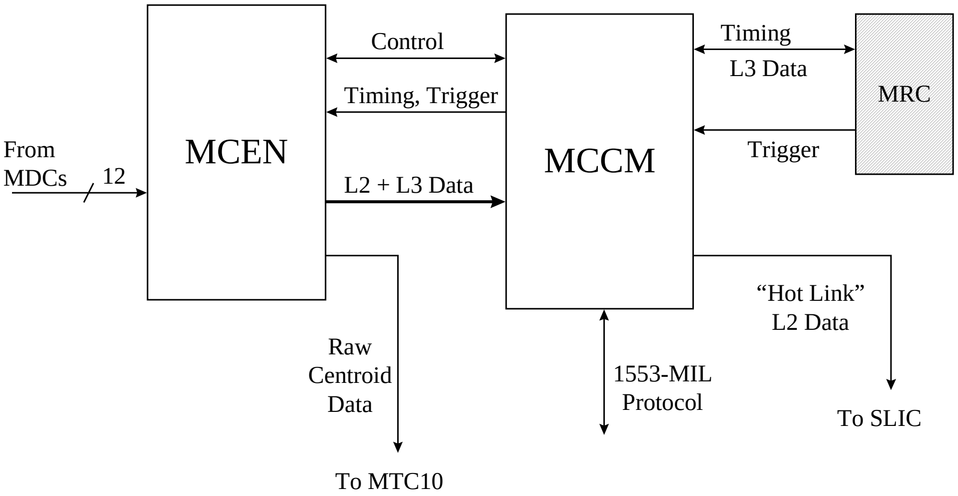

Figure 2 – Inputs/Outputs to MCEN/MCCM Cards

The MCEN is constructed as a 9U x 400mm VME card and provides the following functions:

◊ Receives hit information from the muon front end electronics on up to 12 coax cables. This information is transmitted over high speed serial links at 1060 Mbps.

◊ Synchronizes this hit information so that hit and track information from a given bunch crossing (BC) are processed together.

◊ Processes this hit information on FPGAs to form centroid data which is made available to the MTC10.

◊ Receives timing and trigger information from the MCCM over the VME backplane.

◊ Buffers all input hit and track information, supplemental trigger decision data, card errors and status, and BC number pending an L1 decision. The supplemental trigger decision data is buffered in DPM. All other data is buffered in FIFOs.

◊ Upon receipt of a L1 ACCEPT signal the MCEN buffers all input data, card errors and status, and BC number pending an L2 decision. The buffering of the **input** data is for 1/*n* L1 ACCEPTs where *n* is user defined at the MCCM level. On a L1 ACCEPT, the supplemental trigger information (the result of the centroid FPGAs) is read by the MCCM from each MCEN for inclusion in the data sent to the L2 muon preprocessor. The data is physically buffered in DPM pending an L2 decision.

◊ Upon receipt of a L2 ACCEPT signal, all input data, card errors and status, and BC number are read from each MCEN by the MCCM for inclusion in the data sent to L3 via the MRC and VBD.

◊ The MCEN monitors itself for error conditions. A word containing these errors is asynchronously sent to the MCCM card. These errors include:

- Synchronization error on a serial input line from the muon front end electronics, or MCEN cards.
- Mismatch between internal crossing number and the crossing number received from the trigger framework via the MRC.
- Data not being ready at proper time (including, not coming ready at all)
- VME bus errors.
- Parity problems with input data (if to be included in logic)

◊ The MCEN monitors itself for BUSY conditions which include near fill conditions of either the FIFOs or DPMs. Any BUSY condition is sent asynchronously to the MCCM card.

◊ The MCEN checks the BC number on each RESET received, generates an error and resets the BC counter if it is at the wrong number. At the first RESET signal received after an INITIALIZE, the MCEN also resets the Turn Number, clears all FIFOs and clears all DPM pointer Buffers.

Serial Inputs

There are two types of input signals received by the MCEN card from the outside world: **hit** information from MDT front end electronics and **timing** trigger signals from the MCCM.

Hit information comes from the MDCs in 12 serial links. Each link handles 96 bits, and they are coded into 16-bit words as follows (*note the order of the bits)*:

	D0	**D1**	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**	**D11**	**D12**	**D13**	**D14**	**D15**
Word 1	1	7	13	19	25	31	37	43	49	55	61	67	73	79	85	91
Word 2	2	8	14	20	26	32	38	44	50	56	62	68	74	80	86	92
Word 3	3	9	15	21	27	33	39	45	51	57	63	69	75	81	87	93
Word 4	4	10	16	22	28	34	40	46	52	58	64	70	76	82	88	94
Word 5	5	11	17	23	29	35	41	47	53	59	65	71	77	83	89	95
Word 6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96

Table 1 – Serial Link Input Data Format

The 12 serial links contain data from three separate hitmaps, each one with 344 bits. These hitmaps are sent in four links each to the centroid FPGAs.

Data comes serially to the high speed receiver daughter boards, which use the AMCC's S2043 fiber-channel receiver chips. The output of the boards is the data organized as in Table 1. The data sent on the serial link is as follows:

Word #	**Description**
1	Data (MDT hits)
2	Data (MDT hits)
3	Data (MDT hits)

4	Data (MDT hits)
5	Data (MDT hits)
6	Data (MDT hits)
7	Longitudinal Parity

Synchronization at this point uses the S2043 chip recovered clock. As data exits the daughter-board, data is clocked into FIFOs by this recovered clock. Synchronization can be achieved by recovering the data from these FIFOs using internal clock and timing available in the MCEN boards, originating in the MCCM boards.

As valid data arrives from the Daughter Board, the DAV signal is set. This will enable the data to be written on the FIFO. Data will only be read from the FIFO as soon as all the inputs have received some data. In this way all the data at the exit of this first FIFO will be synchronized, regardless of the size of the cables. The only thing to pay attention here is that the difference between cable sizes shouldn't be more than (in time) 4.0 μs (which should be a very reasonable margin).

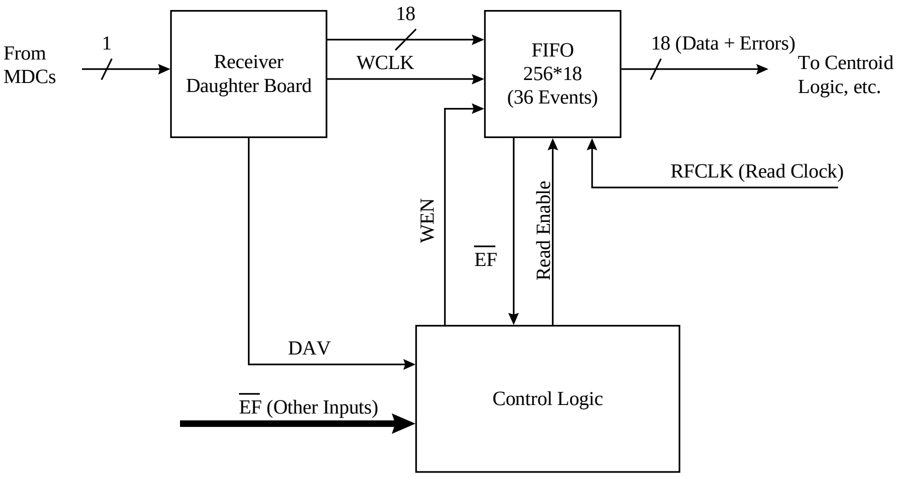

Figure 3– Input Synchronization

The serial connection to the daughter boards is very delicate, as the speed is very high. To avoid problems in impedance matching, it is recommended that the signal comes to each daughter board through a dedicated coaxial cable.

Input channels can be enabled or disabled. This can be important in cases where the geographic placement requires that. There is, then, a configuration register that can be written to define what is the state of the channel.

Timing and Trigger Inputs

These inputs come from the MCCM through the backplane (J2). The consist of the following signals:

The Event Buffering

According to the D0 specification, any front end or trigger system for D0 must provide that

- All data must be buffered while L1 decisions are pending
- Upon receipt of an L1 ACCEPT, all data must be buffered while L2 decisions are pending
- Upon receipt of an L2 ACCEPT, all data must be buffered while transfer to L3 takes place.

We can, then, define several buffers, according to their purpose and situation in the timing landscape:

- **The Input Synchronization** — This is a FIFO that receives all input data, written with timing signals recovered from the serial receivers. This FIFO doesn't need to be very deep. Its depth depends only on the timing difference between the MDCs and the MCEN. It is read with local timing. Its output data is fed to the Centroid FPGAs, where the centroid calculation takes place.

- **The L1 Waiting List** — After the FPGAs do their work, resulting data (centroid data) is immediately sent to the MTC10 modules to help on the L1 decision. It is also stored here, where data is kept while the L1 decision is in process. Input data is also stored here, along with error information related to the present event. Data is taken out of here whenever a L1 ACCEPT or L1 REJECT (absence, in due time, of L1 ACCEPT) is received. With the input data, event counters are also included in the List.

- **The L2 Waiting List** — When a L1 ACCEPT is received, zero-suppressed data and event information (counters) is moved to this buffer, pending the transfer to L2 decision. If this list was empty prior to receiving an event, the data is immediately sent to L2. Otherwise, it will be buffered here while the current transfer is being done.

- **The L3 Waiting List** — When a L1 ACCEPT is received, data is moved to this buffer, where it stays pending a L2 ACCEPT or L2 REJECT signal. Once a L2 ACCEPT is received, the event is marked for sending to L3 on the basis of 1/*n* events. At this point, all data should be here: input data, centroid data, zero-suppressed data, errors, version information, counters, etc.

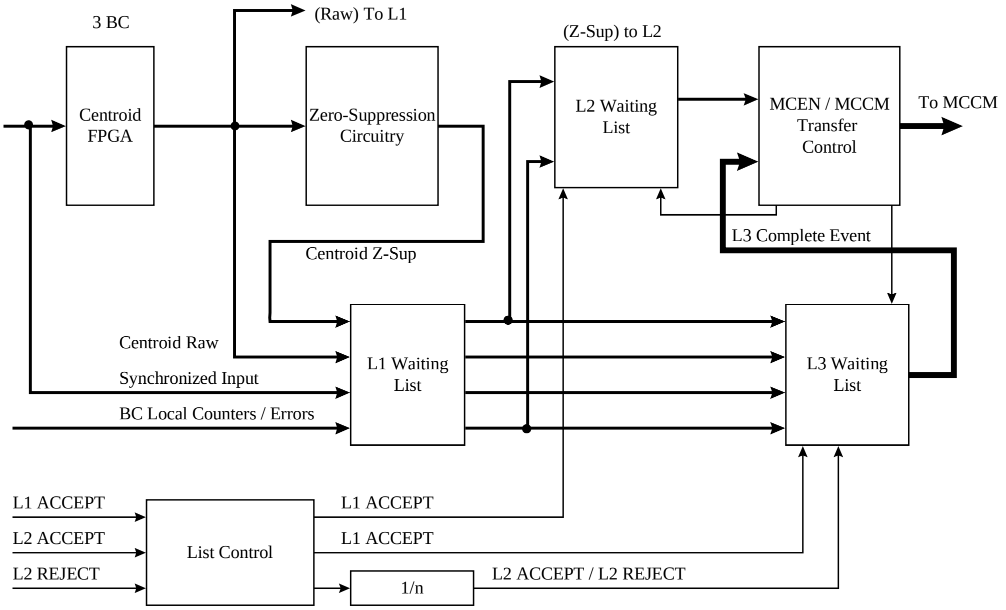

Figure 4 – The MCEN Waiting Lists

Note that, for design purposes, zero-suppressed data may vary in size, i.e. savings can be achieved or (if too much noise in the system) lost (see below).

The L1 Waiting List

The L1 Waiting List is a fairly complex device, because of the types of data, depth variation, and speed that it must accommodate.

Parts of it may be clocked with BC, other parts with the 53 MHz clock.

There is some depth variation within the L1 Waiting List, due to the fact that some of the inputs actually enter the list a few clock cycles later than other inputs. E.g. the actual inputs are stored here, along with the results of the centroid calculation, coming a bit later, and along with the zero-suppressed data, slightly later, waiting for the L1 ACCEPT signal.

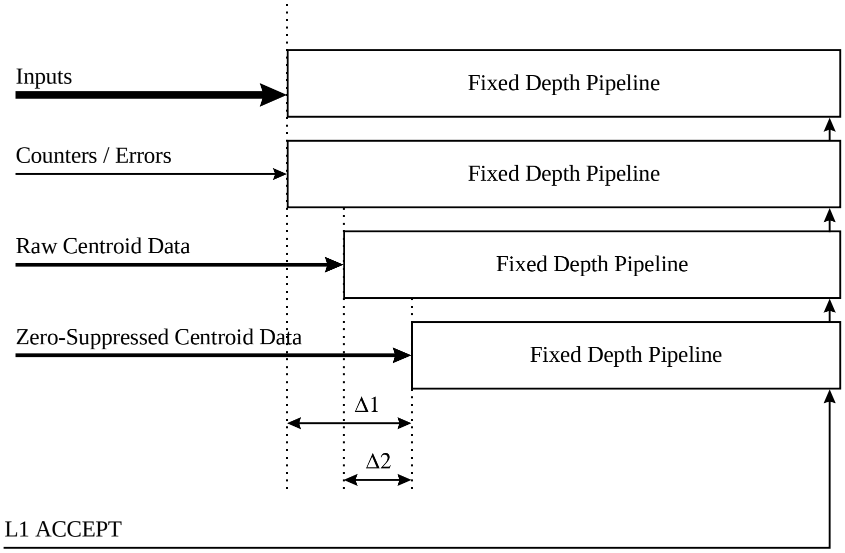

Figure 5 — The L1 Waiting List

The exact depths and frequencies will be defined later, when we have a more precise design and simulation of the centroid calculation and zero-suppression.

The important thing to keep in mind here is that the L1 waiting list is a fixed depth pipeline, not your typical FIFO. It is critical, then, to know how much time exactly (in terms, perhaps, of number of beam crossings) does the decision take.

One way of being (slightly) more relaxed about this is the following (refer to Figure 5):

- We predefine the depths of the Input Buffer and the Counter /Errors Buffer to Δ1. This depth is design constrained, and does not change.
- We predefine the depth of the Raw Centroid Data Buffer to Δ2. This depth is also design constrained, and does not change.
- A mask + delay register (the DEPTH register), programmable, will define a certain amount of waiting time till the end of the very first L1 decision. During this wait period, no read operations are allowed to the L1 Waiting List, so the FIFO is being full of events waiting for the decision. This waiting time is

defined so that, at the end of it, a L1 ACCEPT can appear, validating the first event. If it doesn't, the event is discarded.

- After this first "depth tune-up", the FIFO is continuously read synchronous to the RF or BC clocks.

The L2 Waiting List

The L2 Waiting List is basically a FIFO. It includes the zero-suppressed data from the centroid calculation, and the BC counters for event synchronization. If the FIFO is found empty, data follows, without further delay, to the MCEN / MCCM transfer control unit. Otherwise, it will stay in the FIFO waiting for the previous data to be transferred.

Its depth depends on the transfer rate to the MCCM and on the estimated number of events it will have to hold.

The L3 Waiting List

The L3 Waiting List implements, in a Dual Port Memory, the standard D0 data buffering scheme, which includes a L1 buffer and a L2 buffer. The logic version of the list is shown in Figure 6. In reality, is made as a Dual Port Memory (DPM) where accepts are just marked for readout.

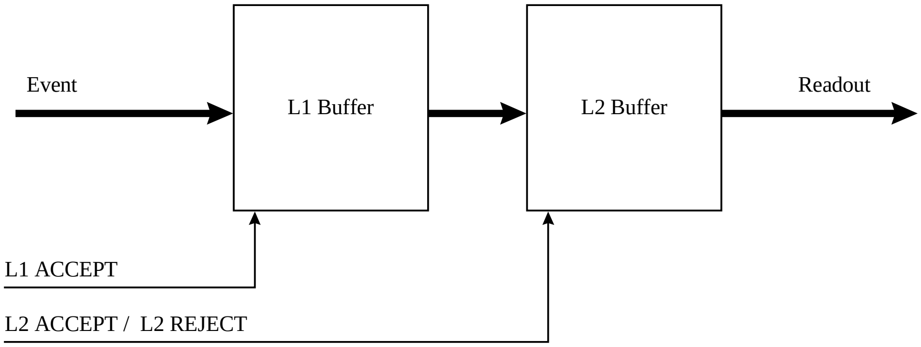

Figure 6 — The two buffers in the L3 Waiting List

In order to save memory space, the event written to the memory with L1 ACCEPT, marked for reading with L2 ACCEPT, marked for deleting with L2 REJECT, and read as soon as possible.

Centroid Calculation

Centroid Calculation is taken care by Altera FPGAs (10K family, typically 10K70). As inputs, these FPGAs receive the paralleled input hitmaps, and produce, as output, a hitmap with the centroid positions.

The input buses arrive in the format specified in Table 1. The seventh word is reserved for parity checking, which is dealt with by the fast receivers / transmitters. The organization of the buses is given in Figure 7. During the course of the design procedure we will try to fit the structure into fewer chips, if possible, cheaper and safe (timing wise).

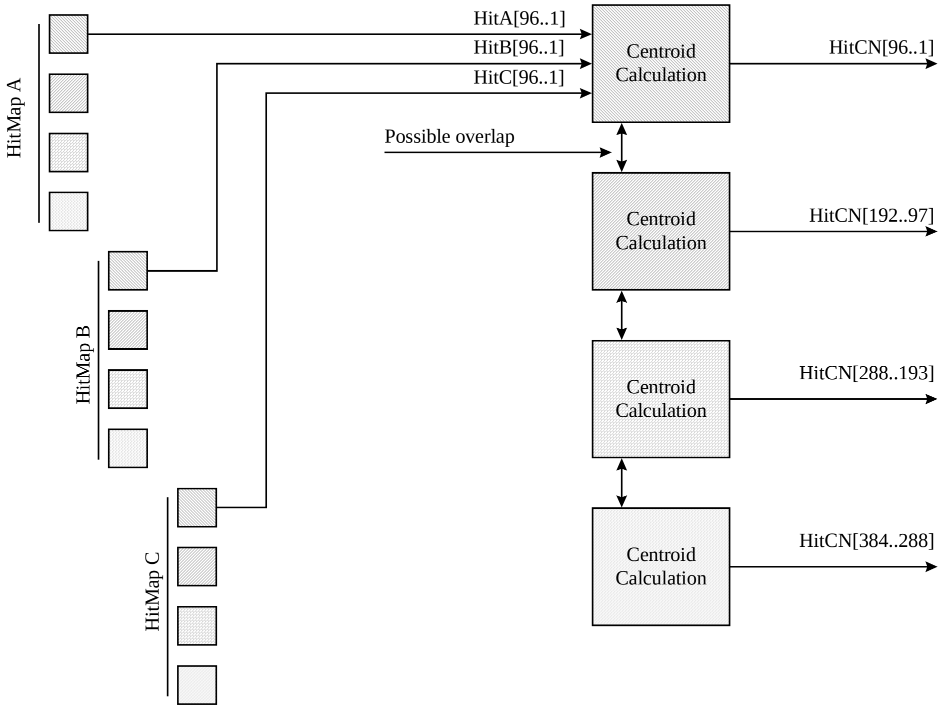

Figure 7 – Organization of the Hitmaps into FPGAs

The first job, prior to actual centroid calculation, is the alignment of the 6 data words into one clock. I call this “de-bussing”. When this is done, actual hitmaps can be put together in order to perform proper centroid calculation.

The calculation itself should take one to two 53 MHz clock cycles.

The resulting hitmap is again compacted into one or two 16-bit buses (“Bussing”). The output is then sent to the L1 trigger.

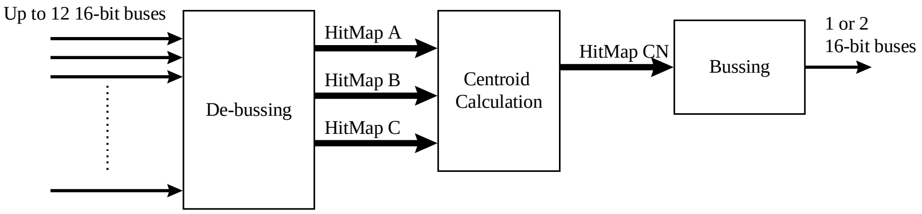

Figure 8 – The Centroid FPGAs

The work for the centroid calculation is actually divided into 4 Altera chips. Each one can be programmed slightly differently according to the particular geographic needs of the area.

Parity Checking

Parity is checked in the Serial Receivers, if they are configured with a bit, **parity_enable**, set to true. The parity word is also added in the output, as data goes to the output serial links (to L1).

The parity error can be sent, with other errors, to L3. The users will have, however, the possibility of not sending this error at all. Two bits, then, are necessary to control the situation:

Bit	***Type***	***Meaning***
parity_enable	Boolean	The 7th word contains parity data (input or output) or data.
parity_report	Boolean	The error will be reported (or ignored) to L3

Centroid Zero Suppression

The hitmap produced by the centroid FPGA should look like a long string of 0's with very few '1'. In order to compact data, we produce a zero suppressed list of addresses, i.e. a list of addresses in which actual centroids were observed.

An idea of implementation is given below (Figure 9). In this suggestion, the hitmap is divided into segments of 8 bits, which are scanned to see whether they contain a hit or not. If hits are not found, whole segments or words are ignored. If hits are found, however, a word is written into an output FIFO, containing the 8-bit pattern and the address of the segment. This will allow fast reconstruction of the segment if necessary, while keeping the "cost" (number of words transmitted) down to a maximum given by the maximum number of hitmap wires.

In this implementation, data comes out of the Altera FPGAs in buses of width 16, and each enters a module such as below. If possible, these modules may be implemented in the FPGAs themselves (small FIFOs can also be implemented there). Each 16-bit word shows as 2 8-bit segments. As a segment and module address is being appended to each segment, each one is being independently scanned for hits. If there is a hit, the composite word formed by the segment address and hitmap is written to a FIFO. The two FIFOs are necessary in order to process the two segment information so that it can be kept in sync with the data flow. As a L1 ACCEPT or L1 REJECT is received, data is immediately sent to L2 via the MCCM. The order of the reading out of the FIFOs will be specified later.

The great advantage of this scheme is that there is no busy time, and that data is written into the FIFO within **one** BC.

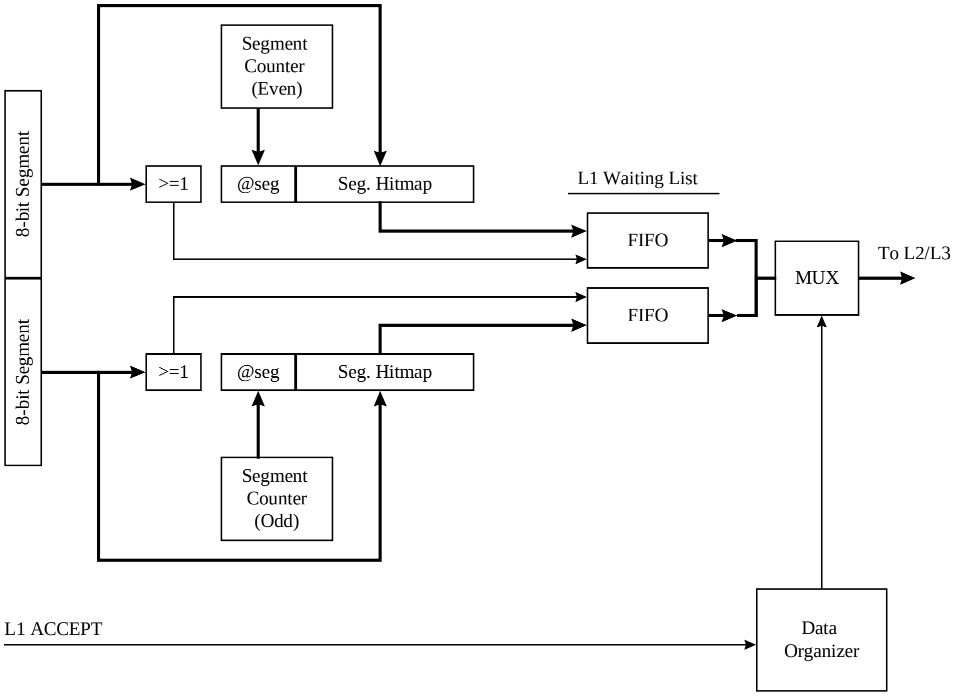

Figure 9 – Implementation of the Segmented Zero Suppression

The four outputs of the centroid FPGAs are, then, directly connected to the input of this module (there should be 4 such modules, then). Outputs can be read later as one single FIFO, so only the readout time varies. The word format with information on each hit of the zero-suppressed data sent to the MCCM card is given below (Figure 10):

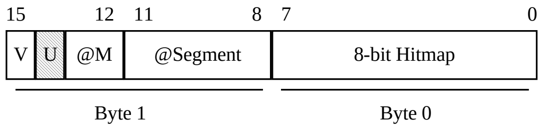

Figure 10 – Zero Suppression data format at MCEN/MCCM communication

Here, the “Module” address tells you which 96-bit hitmap (of the 4 existing per card) we are taking, and the “Segment” address tells you which 8-bit segment of each 96-bit hitmap are we considering. The 8-bit hitmap is the actual data where the hits are.

The “V” bit is a “valid” bit that needs to be sent from the MCEN to the MCCM, but may not need to be propagated to the SLICs afterwards. As each 32-bit word includes typically up to two zero compressed words, it is important that the MCCM knows which of the two words are valid, or whether one of them is just bogus.

Outputs

The main outputs of this module are the following:

- Up to 4 serial links 2043-based containing centroid raw data to L1
- A bus based link (VMEbus or special backplane connection to MCCM) for L2 and L3 data readout
- Control signals for the dialog with the MCCM

Why 4 serial links to L1? Data, coming from the centroid FPGA, consists of a 344-bit hitmap. This fits into 4 96-bit hitmaps, using essentially the same format as input (6 16-bit words, followed by a parity word or, optionally, a 7th data word). Note that the hitmap doesn't fit into three links even if using the 7th word as data. The four links allow us to use the maximum data rate (1.06 Gbps) in the accepted serial link.

Backplane Connectors

On the MCEN board, there will be three main connectors, all in the back of the board: J1, J2 and J3.

- J1 and J2 are used as a A24/A32 D32 VMEbus connectors.
- The user defined pins in J2 are used to receive timing and trigger information from the MCCM. Some pins are also used as special control pins to improve transfer rate of L2 and L3 data to the MCCM.
- J3 is used to receive and send out fast serial data: 12 input serial links from the MDTs, and 4 output serial links to L1 logic.

Connections to MCCM Board (J2 Connector)

The user defined pins of the J2 connector are used to communicate between the MCEN cards and MCCM.

The following is a list of some of the signals that exist on the J2 connector. For each signal, we have indicated the signal name, the type of signal (TTL, PECL, etc.), and the source and destination of the signal.

⇒ M_CLOCK - (Differential PECL) - (from MCCM to MCENs) 4 differential signals that are distributed one for each pair of MCENs in the crate, this is the 53.1 MHz RF clock that is received from the MRC.

⇒ BC_Trig[0:7] - (TTL, terminated on backplane) - (from MCCM to MCENs) - This bus contains the BC Number to associate with the data when the L1-ACCEPT signal is active. Note that the BC Number is only 8 bits, the other 7 bits are not defined as of this revision.

⇒ STORE_ALL - (TTL - terminated on backplane) - (from MCCM to MCENs) - This signal, which can only be active when L1 ACCEPT is active, indicates to the MCCMs that the data for the accepted BC needs to have all of its input data stored and made accessible to the VME bus for possible transmission to Level 3.

⇒ L1 ACCEPT - (TTL, terminated on the backplane) - (from MCCM to MCENs) - Signal to indicate that the crate has received a L1 ACCEPT for the last trigger candidate sent to the Trigger Framework and all data should be saved. This signal comes a set number of crossings after the data is sent to the trigger framework, any candidate that does not receive an L1_ACCEPT is assumed to be rejected.

⇒ SYNCH GAP - (TTL, terminated on the backplane) - (from MCCM to MCENs) - This signal indicates that the beam is in the synch Gap, which means that the input receiving FIFOs should be going empty with (??) clock signals. All of the input FIFOs must be emptied within the Gap, which will result in the INPUT_READY signal going high.

⇒ INPUT_READY - (Open Collector TTL, pulled up on the MCCM) - (from MCENs to MCCM) - This signal is a “wire-or” that contains a high level signal when all of the MCENs in the crate have empty conditions in all of the input FIFOs that have not been masked off. If this signal does not go low during the synch Gap, this is an indication of an error condition to the MCCM.

⇒ L2 ACCEPT - (TTL, terminated on the backplane) - (from MCCM to MCENs) - Signal to indicate that the Level 2 Trigger Framework has accepted a trigger candidate and that the address of the data should be made available in the Buffer Pointer FIFO.

⇒ L2 REJECT - (TTL, terminated on the backplane) - (from MCCM to MCENs) - Signal to indicate that the Level 2 Trigger Framework has rejected the trigger candidate and that the buffer should be freed for use.

⇒ DATA_READY - (Open Collector TTL, pulled up on the MCCM) - (from MCENs to MCCM) - This signal is a “wire-or” that contains a high level signal when all of the MCENs in the crate have L2 data ready to transfer to the MCCM. Each MCEN will release this signal when all of it’s active serial inputs, that have not been masked off on initialization, have received at least one byte of data and the card has sent its Card Trigger Data to the MCCM module.

⇒ BC_CLOCK - (TTL, terminated on the backplane) - (From MCCM to MCENs) BC Clock created by the MCCM. This clock indicates a crate BC whenever the DATA_READY signal indicates that all MCENs have received valid data.

⇒ LEVEL_1_BUSY* - (Open Collector TTL, pulled up on the MCCM) - “Wire-or” signal, each MCEN can cause this line to go low if there is a memory or FIFO full condition that would make it impossible for the board to process a L1 ACCEPT signal.

⇒ LEVEL_2_BUSY* - (Open Collector TTL, pulled up on the MCCM) - “Wire-or” signal, each MCEN can cause this line to go low if there is a condition that make it impossible for the board to process an L2 ACCEPT signal.

⇒ LEVEL_1_ERROR* - (Open Collector TTL, pulled up on the MCCM) - “Wire-or” signal, each MCEN can cause this line to go low if there is an error condition that needs to be reported to the MRC. This error could be caused by a mismatch between the expected and received BC Number for a L1 ACCEPT, L2 ACCEPT or L2 REJECT. This error could also be caused by an inoperative serial receiver on a MCEN Board.

⇒ RESET_COUNTERS - (TTL, terminated on the backplane) - (from MCCM to MCENs) - Signal that causes the MCENs to reset their internal BC counters

⇒ MASTER_RESET - (TTL, terminated on backplane) - (from MCCM to MCENs) - Master reset generated from the initialize command that causes all internal counters to be reset to initial condition.

⇒ L2_MCEN_[15:0] – (TTL, terminated on backplane) – (from MCENs to MCCM) – Each of the modules will indicate, with one of these signals, if it has L2 data to be transferred to the MCCM. A brief sample of the bus in the beginning can indicate if there is at all any data. The bus will be priority encoded so that access is made to the modules that have useful data to be transferred.

⇒ X_CLOCK – (TTL, terminated on backplane) – (from MCCM to MCENs) – This clock has a frequency of typically RF/2 (26.5 MHz) to drive the fast data transmission bus. The exact frequency can be defined later, after performance tests. Another possibility is 32 or 33 MHz, typically also available due to the VMEbus interface chips.

Other signals may be used. This will be defined shortly, when we settle on which type of bus to transfer the data.

VME Interface

The MCEN has a built-in slave-only VME interface, that is mainly used for control and configuration purposes, and not for data acquisition. This allows it to be a fairly cheap implementation. The proposed implementation uses a single-chip VME interface such as Tundra's Trooper II. It can be used as a A32/A24 D32/D16/D08 interface, with full compatibility with the VMEbus standard.

The following devices can be accessed by the VMEbus:

- The VME Controller Configuration Register Area
- A MCEN 32-bit Configuration Register
- A 16 words MCEN 16-bit Status/Error Records
- A L1 Configuration Area (to be defined)
- A 32-bit Counter Register (including BC and number of turns)
- The L2 Waiting List
- The L3 Waiting List

This is possible to be done because, to achieve very fast performance, the address and data lines are shared by the VMEbus and the faster, trigger data transfer bus.

L1 Event Size and Transfer Time

The only data that is sent to L1 is the centroid raw data. As this is sent through 4 serial links per module, each L1 transfer is only 7 16-bit words, clocked with the 53 MHz clock, and takes only **131.6 ns**. The total transfer time (16 cards) is the same, as the operation is made in parallel. The operation, from the input to the output, is also very fast as it takes less than 4 BC.

L2 Event Size and Transfer Time

Data to L2, that is to be transferred as soon as possible, is the zero suppressed centroid data. Normally, at the receipt of an L1 ACCEPT, data is already ready to be immediately transmitted to L2. The quantity of data and duration of transfer varies with the expected occupancy value.

Data, variable in size, is transferred from the MCEN to the MCCM at a rate of 106.2 MB/s, using the RF/2 clock. This transfer rate is enough to determine that it is not the bottleneck in the total operation.

The data is then combined in the MCCMs and sent to the L2 SLICs through **two** serial links, at a rate of 32 MB/s (both).

The absolute minimum is **1** segment hit per crate, and the absolute maximum is **768** segments with hits per crate. These come down to an absolute minimum of **1** 32-bit word, and an absolute maximum (real worst case) of **384** 32-bit words, giving total (MCEN + MCCM) transfer times from **660 ns** through **62.5 μs**.

If one considers a typical occupancy factor of roughly 0.5 %, we can estimate a case where we have two hits per module on the 16 modules. In this case, we would have **6.4 μs** transfer time.

Other occupancy values:

Occupancy (%)	***Number of Words***	***Total Transfer Time***
0.52	32	5.7 µs
1	62	10.6 µs
2	123	20.5 µs
4	246	40.0 µs
6.25	384	**62.5 µs**

Table 2 — Some Occupancy Values and Their Consequences (L2 Transfer)

The worst case is at 6.25 %, as after that all the 32-bit words transmitted have one or more segments with at least one hit. Note the interesting relationship between occupancy and total transfer times, which allows us to easily predict other values. This is true for high occupancy values.

L3 Event Size and Transfer Time

We have, **per event and per MCEN card**, *at least* the following data components:

- The Input Data: 12*16*7/8 = 168 bytes (**42** 32-bit words)
- The Centroid Raw Data: 48 bytes (12*4) , doesn't include the outgoing parity word (**12** 32-bit words)
- The Zero Suppressed Centroid Data: variable, 2 to 96 bytes (**1** to **24**, worst case, 32-bit words)
- Counters: 4 bytes (**1** 32-bit word)
- Errors: 4 bytes (**1** 32-bit word)
- Status: 4 bytes (**1** 32-bit word)

Total: Minimum of 58 words, maximum of 81 words.

We can repeat the same exercise as before for L3 data. What varies is the number of L2 zero-suppressed words. Considering that there will be only one HotLink to L3, at 16 MB/s, the bottleneck for speed is clearly on this side, as the transfer from the MCEN to MCCM will still be done at 106.5 MB/s. Some estimated values are indicated below for different occupancy numbers.

Occupancy (%)	***Number of L2 Words***	***Total Transfer Time***
0.26	16	268.5 µs
0.52	32	273.1 µs
1	62	281.8 µs
2	123	299.3 µs
4	246	334.7 µs
6.25	384	**374.4 µs**

Table 3 — Some Occupancy Values and Their Consequences (L3 Transfer)

NOTE: For other transfer Rates studies, see "*MCEN Transfer Rate Studies"*

References

[1] — Johns, Steinberg, *Muon Trigger Cards (MTC05 and MTC10)*, 07/08/1996

Appendices

1. The J1 Connector

J1 Connections					
Pin	Label	Pin	Label	Pin	Label
A1	DATA00	B1	BBSY*	C1	DATA08
A2	DATA01	B2	BCLR*	C2	DATA09
A3	DATA02	B3	ACFAIL*	C3	DATA10
A4	DATA03	B4	BG0IN*	C4	DATA11
A5	DATA04	B5	BG0OUT*	C5	DATA12
A6	DATA05	B6	BG1IN*	C6	DATA13
A7	DATA06	B7	BG1OUT*	C7	DATA14
A8	DATA07	B8	SPARE02	C8	DATA15
A9	GND	B9	SPAR03	C9	GND
A10	+3.3V	B10	+3.3V	C10	SYSFAIL*
A11	GND	B11	+3.3V	C11	BERR*
A12	DS1*	B12	BR0*	C12	SYSRESET*
A13	DS0*	B13	BR1*	C13	LWORD*
A14	WRITE*	B14	+5V	C14	AM5
A15	GND	B15	+5V	C15	ADDR23
A16	DTACK*	B16	AM0	C16	ADDR22
A17	SPARE00	B17	AM1	C17	ADDR21
A18	AS*	B18	AM2	C18	ADDR20
A19	SPARE01	B19	AM3	C19	ADDR19
A20	IACK*	B20	GND	C20	ADDR18
A21	IACKIN*	B21	+3.3V	C21	ADDR17
A22	IACKOUT*	B22	+3.3V	C22	ADDR16
A23	AM4	B23	GND	C23	ADDR15
A24	ADDR07	B24	IRQ7*	C24	ADDR14
A25	ADDR06	B25	IRQ6*	C25	ADDR13
A26	ADDR05	B26	IRQ5*	C26	ADDR12
A27	ADDR04	B27	IRQ4*	C27	ADDR11
A28	ADDR03	B28	IRQ3*	C28	ADDR10
A29	ADDR02	B29	IRQ2*	C29	ADDR09
A30	ADDR01	B30	IRQ1*	C30	ADDR08
A31	GND	B31	GND	C31	GND
A32	+5V	B32	+5V	C32	+5V

2. The J2 Connector

J2 Connections					
Pin	Label	Pin	Label	Pin	Label
A1	M_CLOCK+0	B1	+5V	C1	M_CLOCK-0
A2	M_CLOCK+1	B2	GND	C2	M_CLOCK-1
A3	M_CLOCK+2	B3	RESERVED	C3	M_CLOCK-2
A4	M_CLOCK+3	B4	A24	C4	M_CLOCK-3
A5	MASTER_RESET	B5	A25	C5	RESET_COUNTERS
A6	STORE_ALL	B6	A26	C6	LEVEL_1_BUSY*
A7	SYNCH_GAP	B7	A27	C7	LEVEL_2_BUSY*
A8	DATA_READY	B8	A28	C8	LEVEL_1_ERROR*
A9	GND	B9	A29	C9	GND
A10	BC_Trig0	B10	A30	C0	BC_Trig1
A11	BC_Trig2	B11	A31	C1	BC_Trig3
A12	BC_Trig4	B12	GND	C2	BC_Trig5
A13	BC_Trig6	B13	+5V	C3	BC_Trig7
A14	GND	B14	D16	C4	GND
A15	+5V	B15	D17	C5	+5V
A16	L1_ACCEPT	B16	D18	C16	SPARE_0
A17	L2_ACCEPT	B17	D19	C17	TDI
A18	L2_REJECT	B18	D20	C18	TDO
A19	BC_CLOCK	B19	D21	C19	TMS
A20	L2_DATA	B20	D22	C20	TCLK
A21	X_CLOCK	B21	D23	C21	FAS*
A22	GND	B22	GND	C22	GND
A23	L2_MCEN_8	B23	D24	C23	L2_MCEN_0
A24	L2_MCEN_9	B24	D25	C24	L2_MCEN_1
A25	L2_MCEN_10	B25	D26	C25	L2_MCEN_2
A26	L2_MCEN_11	B26	D27	C26	L2_MCEN_3
A27	L2_MCEN_12	B27	D28	C27	L2_MCEN_4
A28	L2_MCEN_13	B28	D29	C28	L2_MCEN_5
A29	L2_MCEN_14	B29	D30	C29	L2_MCEN_6
A30	L2_MCEN_15	B30	D31	C30	L2_MCEN_7
A31	GND	B31	GND	C31	GND
A32	+5V	B32	+5V	C32	+5V

3. The J3 Connector

J3 Connections			
Pin	Label	Pin	Label
A1	L1_OUT_0	B1	L1_OUT_2
A2	L1_OUT_1	B2	L1_OUT_3
A3	S_INPUT_0	B3	S_INPUT_6
A4	S_INPUT_1	B4	S_INPUT_7
A5	S_INPUT_2	B5	S_INPUT_8
A6	S_INPUT_3	B6	S_INPUT_9
A7	S_INPUT_4	B7	S_INPUT_10
A8	S_INPUT_5	B8	S_INPUT_11