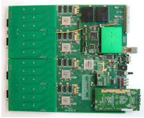

LHCb 2005-xxx
August 19, 2005

LHCb VELO and ST clusterization on TELL1

Version 2.0

Guido Haefeli[1] Alex Gong[2]

Abstract

The LHCb Vertex Locator (VELO) and Silicon Tracker (ST) are two silicon strip detectors with similar occupancies and cluster size distribution. This leads to a proposal to use an identical clusterization algorithm. To increase the modularity for the FPGA design a cluster format and cluster linking processing is proposed that can be used for the L1T data stream at 1.11 MHz and HLT data stream at 40 KHz but also at 1.11 MHz.

[1]E-mail: `Guido.Haefeli@epfl.ch`
[2]E-mail: `gonggh99@mails.tsinghua.edu.cn`

Contents

1 Revision **1**

2 Introduction **1**

3 Cluster format **2**

3.1 Only one cluster format on TELL1 2
3.2 Position encoding 2
3.3 ADC values 2
3.4 Byte aligned and bit ordering 2
3.5 Velo specific 3
3.6 Proposed format for Velo 3

4 ST specific **4**

4.1 Neighboring strip ADC values "Spill over information" 4
4.2 Proposed format for ST 4
4.3 L1T and HLT information separated from ADC values 5

5 Cluster algorithm for Velo **5**

5.1 Parameter definition and concepts 5
5.2 Velo clusterization summary 8

6 Cluster algorithm for ST **8**

6.1 ST clusterization summary 9

7 Cluster linking **10**

7.1 Clusterization derandomizer 10

1 Revision

Version 2.0, 19.8.2005
The L1T and HLT does not require the ADC values in case the total cluster position (14-bit Velo, 15-bit ST) can be made available. It seem advantageous for the use in the L1T (1MHz decoding) to send first the L1T information (16-bit) followed by all ADC values of the TELL1 board in a appended section. These comments come from Hans Dijkstra. **Version 1.7, 12.7.2005**
Implement comments given by Matthew, add sum threshold for L1 trigger algorithm, add example thresholds in the summery, use Beetle wide thresholds for qualification and sum threshold. Describe more precise the use of the inter strip position and the strip position.
Version 1.6, 5.7.2005
Implement comments given by Matthew, separate ST and Velo algorithm, don't merge split clusters over Beetle boundaries.
Inserted the remarks on first threshold to be strip individual, sum threshold fixed PP-FPGA wide and low threshold as well PP-FPGA wide. **Version 1.5, 22.6.2005**
Use minimal cluster format for Velo with 3-bit inter cluster position. After comment from Doris.
Version 1.4, 22.6.2005
Mark the inter strip position being "0000" for the one strip clusters.
Version 1.3, 22.6.2005
Removed the ST minimal format after discussion with Matt, add bit ordering remark.
Version 1.2, 22.6.2005
Removed sum threshold bit from the proposed ST cluster format since it is at the present not foreseen. There is one unused bit now. Made some references for things that were written twice. The cluster inter strip cluster position is now a signed value and understood as a correction to the cluster strip position.
Version 1.1, 21.6.2005
Changed the low threshold hit inclusion. Only the left and right neighbor to a seeding hit is included in the cluster. This avoids to give any asymmetries during the clusterization, 3 seeding strips do not have any low hits included (otherwise we need to do an asymmetric decision in case there is one to both sides).
Version 1.0, 20.6.2005
First draft of the cluster format, clusterization and cluster linking processing stage.

2 Introduction

TELL1 [1] is the data acquisition board used in LHCb to receive data from the electronics situated in the radiation area over long optical or copper links. The digitized data is sent to the so called "PreProcessor-FPGA" (PP-FPGA), where the zero suppression for the L1 trigger and the data storage during L1 latency is performed. The L1 accepted data is read out of the L1 buffer and sent for further processing to the "SyncLink-FPGA". In this note the clusterization used to zero suppress the data for the L1T and the HLT are discussed. This part was not discussed in very detail in [2].

3 Cluster format

The cluster format for the L1T and the HLT data streams have been defined in the past according to the requirements given by the their event rate. With an event rate of 40 KHz for the HLT compared to the 1.11 MHz for the L1T, a rather large data format for the HLT clusters was adopted where a sparse format of only 16-bit per cluster for the L1T was defined.
During the discussion of the feasibility of a combined readout at 1.11 MHz it became apparent that some optimization of the cluster format could lead to a large simplification of the processing on the TELL1.

3.1 Only one cluster format on TELL1

The idea is to define a unique cluster format for L1T and HLT that could be used also for the 1.11 MHz combined readout. The concept behind is to use one unique cluster format during the processing on TELL1 and strip off the additional information on the last processing stage for the L1T data stream where no ADC values are required.

3.2 Position encoding

The current cluster format does not encode the cluster position in a way that is useful for the processor in the track finding algorithm. The present format encodes the position of a cluster by sending position of the first strip in a cluster. Even the resolution for the track finding does not require inter strip position it needs to calculate a meaningful cluster position. For a new cluster format we propose to calculate the cluster position with a weighted average and allow to encode the inter strip position with 3-bit precision for both ST and Velo. A higher precision is not needed for the L1T and HLT algorithm and if needed off-line the ADC values must be used to recalculate this with some eta distribution or what ever. The total cluster position is now in the MSB part the cluster strip position (Velo=11-bit, ST=12-bit) and the 3 LSB the inter strip cluster position.

3.3 ADC values

By definition of the strips included in a cluster have positive values. This allows to remove the sign bit and leads to the 7-bit ADC information. Remark: The reduction of the 8-bit ADC value sampled and transmitted of the optical links for the ST are reduced to 8-bit signed values by subtracting the pedestal values. This leads to the signed representation (-128 to 127). The MSB of the ADC values which is the signed bit will be replaced by the start of cluster bit, indicating in the ADC value section how many ADC values belong to a cluster.

3.4 Byte aligned and bit ordering

To get a more compact format with less padding data, the cluster format is new byte aligned. The current HLT was 32-bit and the L1T 16-bit aligned. Remark: There is

no reason to have 2-byte aligned data since a general purpose CPU does work as well with byte as with two byte aligned formats.
All numbers given in the clusters are MSB on the right, LSB on the left which corresponds to the big endian concept.

3.5 Velo specific

3.6 Proposed format for Velo

The data format for the Velo does include all information already sent with the present L1T cluster format. New is the complete 14-bit center of cluster position. The resolution is therefore 1/8 of a strip pitch. To match ADC values with strip number the position calculation with a weighted average can be repeated using the ADC values of the cluster.

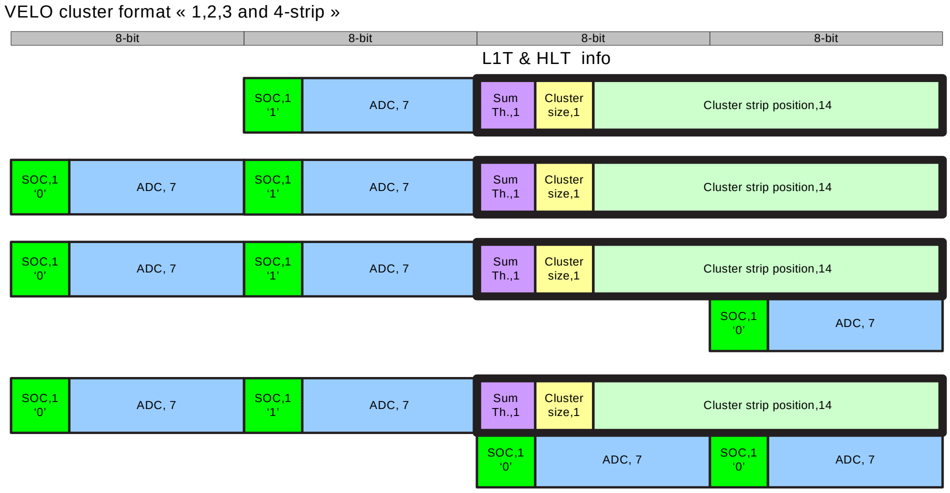

Figure 1: *Velo cluster format for 1,2,3 and 4 strip clusters*

Cluster strip position,14 For the Velo a 11-bit number is required to encode the strip position of a cluster. This at least is true for the R sensors (these 11 bit are the MSBs). The inter strip position is calculated with a weighted average with a precision of 3-bit. This makes up the 3 LSBs.

Cluster size,1 This bit indicates if the cluster is a one or two hit cluster or 3 and 4 hit cluster. The maximum size of cluster generated is 4. Clusters larger then 4 are split into multiple clusters. The clusters split by the boundary of analog readout links and Beetles are merged by using the neighboring strip information. This merging is restricted to the PP-FPGA boundary. Lower thresholds on the PP-FPGA boundary can compensate not to loose too much information. The encoding of the cluster size is.

Sum Threshold,1 This bit indicates if the sum of the ADC values of the strips in a cluster is above a sum threshold. This allows to qualify the cluster and if wanted allows to have a cut on the total charge in a cluster. For the Velo this is not an excluding condition for the cluster, in case of the ST it is. This is used by the L1 trigger algorithm.

ADC value,Nx7 For each strip in the cluster a 7-bit ADC value is attached to the cluster information. They are sent in order of increasing strip number (see also 3.3).

ADC SOC,1 The ADC start of cluster bit (SOC) is set to 1 for the first ADC value in the cluster and 0 for all following. (see also 3.3).

4 ST specific

4.1 Neighboring strip ADC values "Spill over information"

For the ST HLT cluster format it is foreseen to add the information about the strips next to the cluster. This information can be used to reduce the number of spill-over clusters. It has been seen that the spill-over clusters do have negative ADC values on the neighboring strips. This information can be sent out in a more compact way. We propose to calculate the sum of the ADC values of the neighboring strips. A 5-bit value can be used to indicate a sum in the range of (-16 to 15).

4.2 Proposed format for ST

The cluster for the ST have the same definition except that the strip position is increased by one bit, replacing the size of cluster information in the Velo format and in addition a 5-bit neighboring strip sum is calculated and sent among the ADC values.

Cluster strip position,15 For the ST a 12-bit number (MSB) is required to encode the strip position and a 3-bit inter strip position makes up the 3 LSB.

Sum Threshold,1 This bit indicates if the sum of the ADC values of the strips in a cluster is above a sum threshold. This allows to qualify the cluster and if wanted allows to have a cut on the total charge in a cluster. This is used by the L1 trigger algorithm.

Neighbor strip sum,5 The adc value of the right and left neighbor strip for each cluster is summed and encoded in a 5-bit signed value. This is a saturated sum:

$$Neighbor\ strip\ sum = \begin{cases} 15 & \text{if } x \geq 15 \\ x & \text{if } -16 < x < 15 \\ -16 & \text{if } x \leq -16 \end{cases} \tag{1}$$

ADC value,7 see Velo 3.6

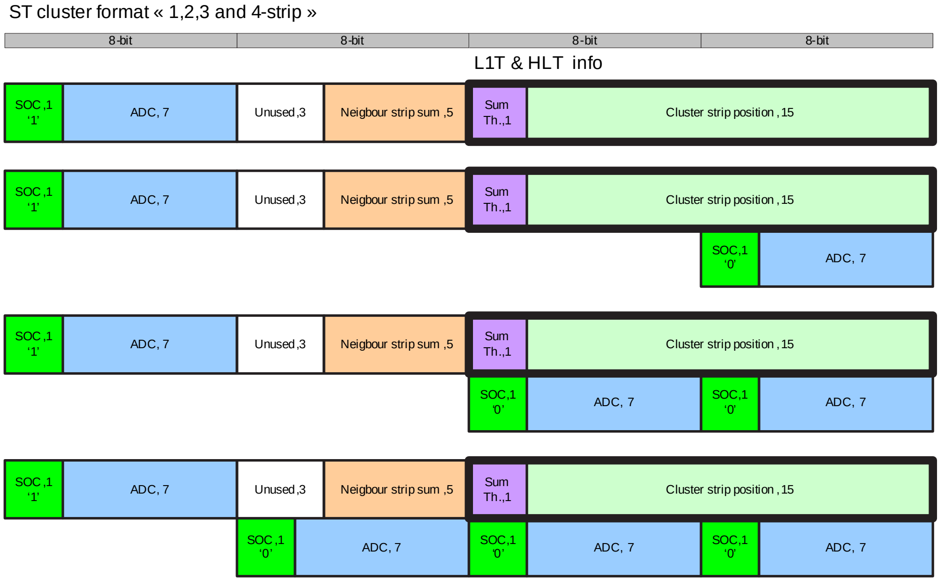

Figure 2: *ST cluster format for 1,2,3 and 4 strip clusters*

4.3 L1T and HLT information separated from ADC values

To allow a very efficient decoding of the L1T information for the 1.11-MHz event rate the L1T necessary information is collected in the first section of the data block followed by all off-line information as spill over and ADC information. The SOC bit allows for a very easy indication which ADC values belong to which cluster.

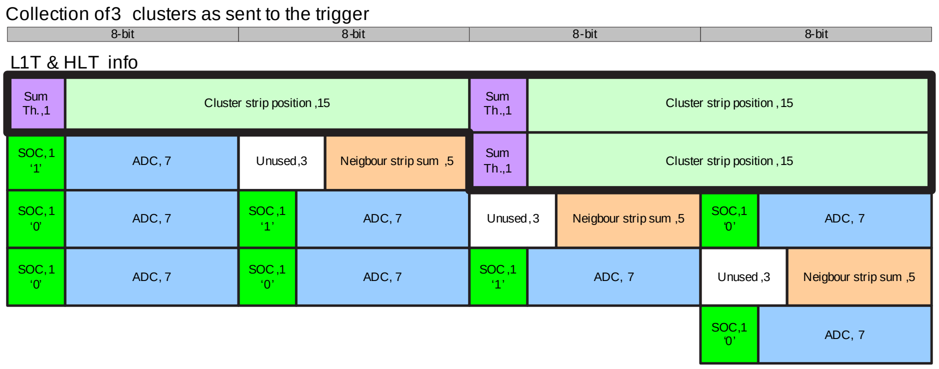

Figure 3: *In this example 3 clusters for the ST are shown. L1T and HLT relevant info is sent first with fixed size per cluster followed by the ADC values*

5 Cluster algorithm for Velo

5.1 Parameter definition and concepts

To perform the clustering a few parameters have to be fixed in order to reduce the amount of resources used for this task.

Threshold calculation The thresholds used to perform S/N cuts can not be calculated locally on TELL1 in the present system. The strict recalculation of the L1T clusters in the HLT can only be achieved if the thresholds are fixed. Therefore all thresholds are calculated off line!
Even in case the 1.11 MHz readout is adopted, the download of the thresholds is the preferable solution to reduce the amount of complexity in the TELL1.

Seeding strip To identify a cluster a **first** strip individual **threshold** is applied. The seeding strip comparison result is stored in a shift register for each strip. This allows to handle multiple seeding strips to become one cluster. In order to merge the clusters over the analog link boundary (32 strip) and Beetle boundary (128 strip) some left and right neighbors have to be made accessible for each cluster processor. With a maximal cluster size of 4, we required to add the 4 left neighbor and one right neighbor for each unit of 64 strips in order to have a perfect overlap of the boundaries. The first

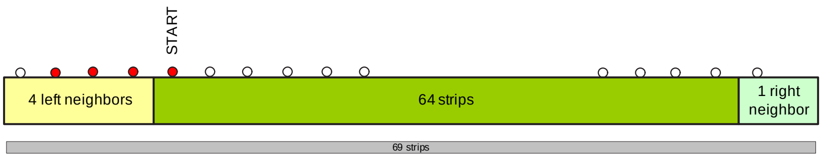

Figure 4: *For the cluster search 4 left and 1 right neighbor strips are added leading to a total of 69 strips.*

threshold is a strip individual value!

Low threshold With a comparison to a **low threshold** one left and one right neighbor to a seeding strip(s) can also be included in a cluster (except for 3 and 4 seeding strip clusters). This allows to find a small signal in the neighborhood of a large ADC value of a seeding strip. The sum threshold comparison result is also stored in a shift register and can be accessed and used for combinatorial logic anytime during the clusterization.
In [3] a sum threshold is calculated from the seeding strip value eg. 20% of the signal in the seeding strip. This kind of condition is rather difficult to handle in the cluster processor on the FPGA since this requires a recalculation of the of the threshold and the comparison. A lot of additional pipelining results from this.
The low threshold is a fixed value per Beetle!

Split clusters for R-Sensors All clusters that lay within one sector of the R-sensor (512-detector channels) are merged together.

Split clusters for Phi-Sensors Only clusters that lay within one PP-FPGA (512-detector channels) can be merged together. Inner and outer region per PP-FPGA make up two regions (≈ 171 inner and ≈ 341 outer) strips. These can have merged clusters.

Cluster sum A sum threshold bit is generated by comparing the total cluster charge with a **sum threshold**. The total cluster charge is calculated by summing the ADC values of all strips in a cluster. The ADC values of the left and right neighbors are stored in a shift register during the clusterization. This allows to output the ADC values of a cluster without any further memory access and the sum can be calculated pipelined.
The sum cluster threshold is a fixed value per Beetle!

Weighted average The weighted average of the maximal 4 strip cluster is calculated. The necessary division is resource hungry. The division was seen much to slow with the Altera LPM module foreseen for integer division. It is implemented in two steps. First the inversion of the sum with a lookup table and second the multiplication of the inverted sum with the weighted sum. The number of bits calculated has to be minimized in order to reduce the resource usage. A local cluster position x needs to be calculated.

$$x = \frac{y(0) * 0 + y(1) * 1 + y(2) * 2 + y(3) * 3}{y(0) + y(1) + y(2) + y(3)} = \frac{y(1) + y(2) * 2 + y(3) * 3}{y(0) + y(1) + y(2) + y(3)} \tag{2}$$

In order to make an inter strip position calculation with integer operation possible, the equation has to be scaled. Assuming that a 4-bit resolution is sufficient, the nominator has to be multiplied by 16 (shift 4bit to the left).

$$pos = \frac{16 * (y(1) + y(2) * 2 + y(3) * 3)}{y(0) + y(1) + y(2) + y(3)} = \frac{16 * wsum}{sum} \tag{3}$$

The required precision of the division operation has been evaluated with a c-routine:

$$wsum = y(1) + y(2) * 2 + y(3) * 3 \; [10bit\ unsigned] \tag{4}$$

$$sum = y(0) + y(1) + y(2) + y(3) \; [9bit\ unsigned] \tag{5}$$

$$sum_{inversed} = (1 << 16)/sum) \; [16bit\ unsigned] \tag{6}$$

$$pos_{4bit} = (wsum * sum_{inversed}) >> 12 \; [6bit\ unsigned] \tag{7}$$

C-code

```
sum  = y0 + y1 + y2 + y3;
wsum = y1 + 2*y2 + 3*y3;
inversed_sum = (int)((1<<16) /(sum&0x1FF));
result = (wsum&0x3FF) * (inversed_sum&0xFFFF);
result = (result >> 12);
```

The precision of the inter strip position has been verified to be better than 4bits.

Splitting clusters The maximal number of strips in a cluster is 4. For clusters larger than this a splitting rule has to be defined. The clusterization is performed from the low strip numbers to the high. This leads to some special cases illustrated in figure 5. Once a cluster is build, all strips included in the previous cluster are regarded as non seeding and not over low threshold anymore. The corresponding bits in the shift register are reset with this we can avoid sending the ADC value of one strip twice.

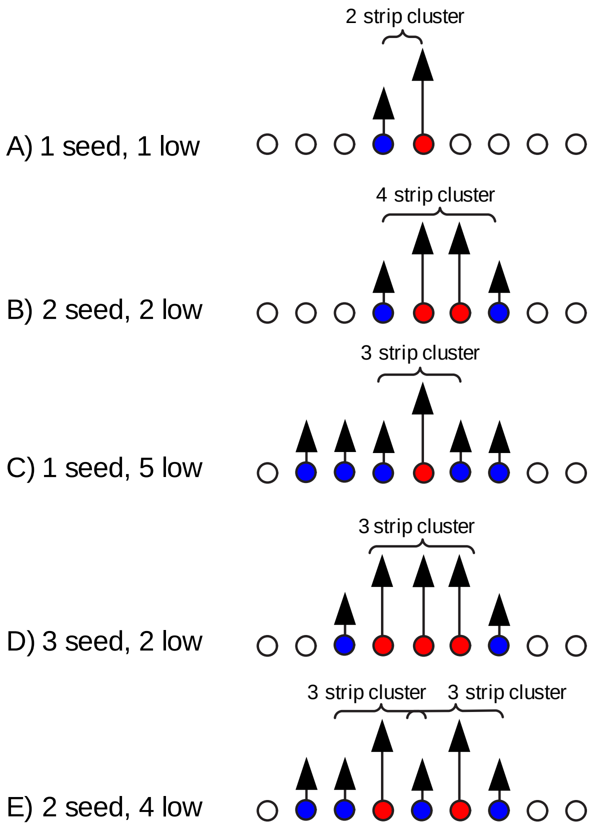

Figure 5: *This figure shows how the clusterization is done using a seeding and a low threshold (low threshold is only used by the Velo): A) Cluster with one seeding and one strip over the low threshold resulting in a two strip cluster. B) Two seeding strips in one 4 strip cluster. C) One seeding strip and too many low threshold strips, only one left and one right neighbor is used for one 3 strip cluster. D) The 3 seeding hit cluster does not get any low strip hits in the cluster. E) Two clusters are build, one strip ADC value is sent out double.*

5.2 Velo clusterization summary

- A seeding strip found with a **first threshold**.
- A **low threshold** is applied to add strips with less signal to the cluster.
- **The sum of the total deposited charge** of maximal 4 strips (seeding or low threshold past) is calculated. This information is used to set a qualifier bit called **sum threshold** bit.
- The **inter strip cluster position** is calculated with the weighted average and sent as a signed 3-bit correction value to the cluster strip position.

6 Cluster algorithm for ST

First threshold With a first threshold the strips in a cluster are detected. As for the Velo a maximum of 4 strips per cluster are accepted. A fifth strip would result in a 4 strip plus one strip cluster. The first threshold is a strip individual value!

Confirmation threshold After a cluster has been detected with a seeding strip, a confirmation threshold is applied to the sum of ADC values of the strips in the cluster. The cluster is not sent out if this confirmation threshold is not achieved.

Sum Threshold For the L1 trigger an additional qualifier bit is calculated using the sum threshold.

Weighted average See the same paragraph for Velo. Remark that the inter strip cluster position can be recalculated with some advanced correction in the CPU farm. This position can be used for the triggers and is supposed to be sufficient for that purpose.

Neighbor strip values For the ST a sum of neighboring strips is calculated for each cluster. This requires one more left and right neighbor to be included in the cluster data preparation (see figure 4). This replaces the concept of low threshold hits in case of the Velo. For the ST the concept of a low threshold was not used in the test beam analysis [4]. The total charge in the cluster is compared with a sum threshold. Clusters are only accepted (sent to the DAQ) if the sum threshold qualification has passed.
The sum cluster threshold is a fixed value per Beetle!

Split clusters over Beetle boundaries Only clusters that lay within one Beetle boundary (128-detector channels) are merged. Clusters that lay over Beetle boundaries are sent out separate.

6.1 ST clusterization summary

The clusterization described in [4] can be performed. This process uses the following elements:

- A seeding strip is detected with a **first threshold** that is eg. ($S/N > 3$).
- **The sum of the total deposited charge** of maximal 4 strips is calculated. Clusters below the **confirmation threshold** eg. ($S/N > 3.6$) are discarded!
- A **sum threshold** bit is set if the sum of the cluster charge is exceeding the sum threshold ($S/N > 6$).
- The **inter strip cluster position** is calculated with the weighted average and sent as a signed 4-bit correction value to the cluster strip position.
- **The sum of the neighboring strips** is calculated as a 5-bit signed value.

7 Cluster linking

With each clustering processor responsible for 64 detector channel, a maximum (for ST) of 12 processors are needed. Each processor can perform its task in less than 900 ns with a processor running at 100 MHz. The theoretical maximum number of clusters resulting from 64 detector channel is 32 3-strip cluster leading to 32 x 40-bit word of data. In a worst case event each cluster processor extracts 32x40-bit of data in 900 ns (1.3 Gbit/s). To perform the linking of the 12 data sources of such a worst case event in 900 ns a data bandwidth of 12 x 1.3 Gbit/s would be required.
The above calculation is only an estimate of the worst case bandwidth, in a real implementation a cut to restrict the maximum number of clusters per PP-FPGA is foreseen. This cut is high enough that linking of a worst case event remains a difficult task.

7.1 Clusterization derandomizer

The worst case event discussed in the above paragraph is an unlikely situation. With an expected average output of 4 clusters per PP-FPGA per event, a linking stage which is capable of linking 128 clusters each event is a waste of a large amount of logic.
The here after proposed linking scheme does make use of the two on chip so called M memory blocks (1 Mbit, 128-bit wide dual port). In a first stage (see figure 6) the common mode corrected data is stored in one of the M-RAM. This buffer acts as a derandomizer for the linking processor. The event size at this stage is 1 byte per detector channel (768 byte). This allows to store 128 events in the buffer. To comply to the L0-throttle requirement, one quarter of the buffer needs to be reserved to prevent the buffer from overflowing by events still arriving after a L0-throttle is flagged (the number of events in the L0-derandomizer plus the events in the common mode suppression processors).

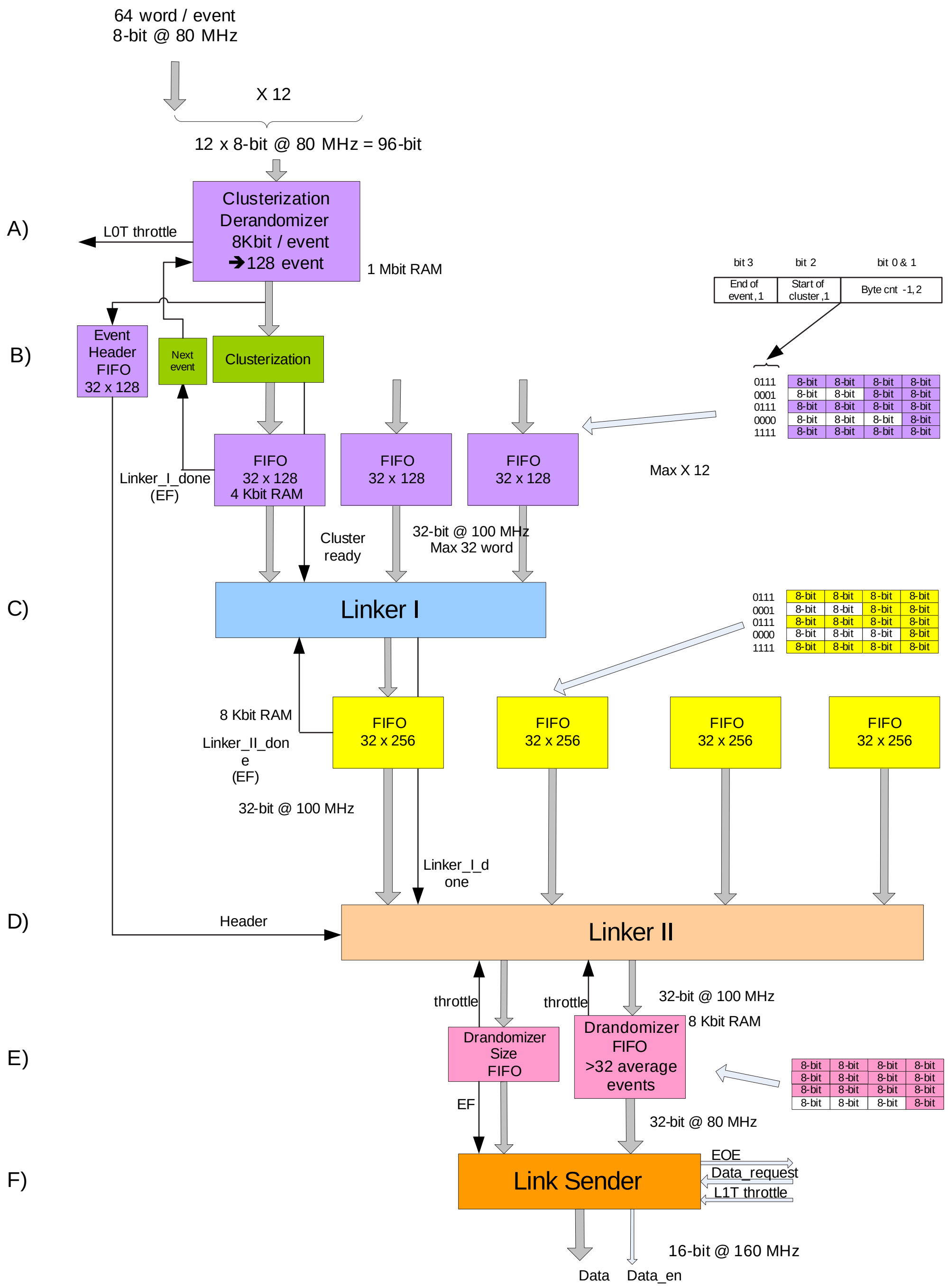

Figure 6: *A) Is the clusterization derandomizer buffer implemented in two 512-Kbit RAM, B) is the cluster processor performing all cluster generation, unused bytes are decoded in the parity bit section as well as start of cluster and end of event C) the cluster linker on level I serves to merge the clusters of 3 cluster FIFOs together, D) second cluster linker which serves 4 inputs, at this stage a cut on the maximum number of clusters can be applied but is not necessary, this stage also manages to write the clusters without the unused bytes, E) link derandomizer with corresponding size of event FIFO, F) link interface, two to one multiplexer.*

References

[1] Guido.Haefeli *et al.* "TELL1 Specification for a common read out board for LHCb", IPHE Note **2003-02**, LHCb Note **2003-007**.

[2] Guido.Haefeli "Contribution to the development of the acquisition electronics for the LHCb experiment", CERN-THESIS-2004-036.

[3] Niels.Tuning "L1-type Clustering in the VeLo on Test-beam Data and Simulation", LHCb Note **2003-073**.

[4] Matthew.Needham "Silicon Tracker simulation performance", LHCb Note **2003-015**.