LHCb 2000-046
CALO
6 June 2000

The readout of the LHCb Calorimeters

Christophe Beigbeder, Dominique Breton, Olivier Callot, Jacques Lefrançois

Laboratoire de l'Accélérateur Linéaire - Orsay

ABSTRACT

This document describes the readout of the LHCb calorimeters. The front-end electronics has been described in a note [1], which is mainly devoted to the shaping, integration and digitisation at 40 MHz of the PM signals. Once digitised, the signal is sent to the trigger part of the front-end board, described in [2], and also to a pipeline, waiting for the trigger decision. This note describes how the digitised values are handled, up to the interface with the DAQ system. The readout of the PreShower and SPD information will follow a very similar path. We in fact intend to use the same design for this part of the front-end board, and the same Calorimeter Read Out Card (CROC) to format the data for the DAQ. While the design of the front-end board is advanced, and a prototype already available, the CROC, which are needed in a smaller number (26), are still at the conceptual design stage.

1. PIPELINES ON THE FRONT-END BOARD

The block diagram of the calorimeter front-end board is displayed in Figure 1. This note concentrate on the bottom right part, which contains the RAM for data coding, the L0 latency pipeline and de-randomiser buffer, and the L1 latency buffer.

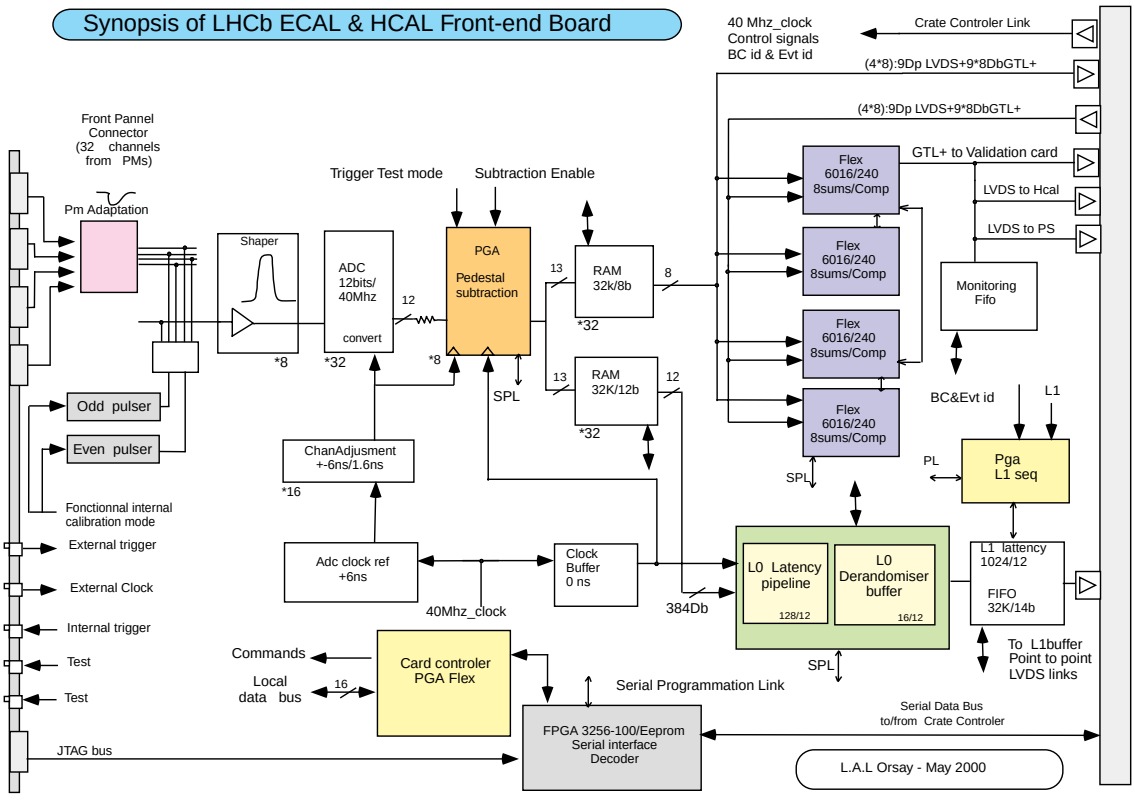

Figure 1: Block diagram of the calorimeter front-end Board

1.1. Data processing

Starting from the output of the ADC, the data is processed in several steps. The lower of the previous two samplings is subtracted from the raw ADC digitisation for low frequency noise removal and pedestal subtraction. This 13-bit value addresses a 12-bit RAM, which contains the coded measurement for this cell. We intend to use a non-linear coding as output of this RAM, keeping a good precision for low values

of the signal. This RAM allows to correct for possible non-linearity, to apply a first gain correction, and also to remove the remaining pedestal. As can be easily understood, subtracting the lower of the previous two samples underestimates the average pedestal, producing an average offset of about 40% of the width of the pedestal. This pedestal offset is expected to be around 0.5 counts.

With a requirement to keep the measurement accurate to 2 per mil for medium or large values, one can devise a coding scheme such that we can code a 16-bit number on 12 bits, a 3-bit exponent and a 9-bit mantissa, with provision for small negative values. A possible scheme is described in Table 1. With such a scheme, the mean pedestal can be corrected to 1/8 of an ADC channel.

Original value	Exponent	Rule for the Mantissa
-32768 to –512	0	0
-511 to –1	0	Original + 512
0 to 511	1	Original
512-1023	2	Original – 512
1024-2047	3	Original/2 – 512
2048-4095	4	Original/4 – 512
4096-8191	5	Original/8 – 512
8192-16535	6	Original/16 – 512
16536-32767	7	Original/32 – 512

Table 1: Possible coding on 12 bits of a 16 bit signed number

1.2. Timing and trigger processing

The L0 and L1 decisions are received by the TTCrx chip, which is in the CROC, and distributed over the backplane, together with the ECS signals used for configuring the cards. The clock is distributed on the backplane by a point to point isochronous connection and the other TTC signals (reset and trigger decisions) are bussed.

Data is pipelined during the fixed L0 trigger latency (4us). Upon reception of L0 yes, data is sent to the de-randomiser buffer, a 16 word FIFO [3] for each channel, together with the 12 bit BCID of the current event on another FIFO.

A 34-word buffer is then produced, by creating a header including the BCID information, followed by the 32 data words, and a trailer including error detection flags. We use now 2 extra bits to tag the header and trailer in the block, to allow a safer detection of the event boundary. This block is stored in a 1024 [3] block (events) FIFO (1k x 34 x 14 bits), waiting for the L1 yes or no decision, which always fires its readout. Only data corresponding to L1 yes is sent over the backplane towards the CROC. For the PreShower card, the channel information is 8 bit of ADC encoding, plus the two bits (PreShower and SPD) used for the trigger.

2. TRANSMISSION TO THE CROC

A crate will host up to 16 front-end cards. Each calorimeter card contains 32 channels with 12 bits of information. Each PreShower card contains 64 channels with 10 bits each. The total amount of information to process in the Readout Card is then 512 or 1024 words of 12 bits, with an average input rate of 40 kHz, the L1 accept rate. However, two consecutive L1 accept may be separated by no less than 900 ns. This implies that one should be able either to process completely each event in less than 900 ns, or to buffer the following event while processing the previous event if this processing takes longer.

We intend to extract the event from the L1 latency buffer in less than 900 ns, and to have a de-randomiser function in the Calorimeter Readout Card. This allows the processing in the CROC to take longer than 900 ns, the limit being to stay comfortably below 25 μs on average, to handle the 40 kHz L1 rate.

2.1. LVDS links

Readout data will be output on the backplane through LVDS serial links. This new link standard allows serialising up to 21 bits at 40MHz over only three differential data pairs plus one clock pair. Every front-end board will then be connected individually to the readout board and send its data at the same time. The multiplexing will occur only on the CROC.

3. THE CALORIMETER READOUT CARD (CROC)

The first purpose of the Readout Card is to collect the data, the second purpose is to reduce the amount of data, by applying a zero suppression algorithm. Next, this card should do the final formatting of the data. The last function is to send the data to the DAQ system. The physical implementation of this card may be in fact split into two physical objects: A first card in the crate on top of the calorimeter, handling the collection of the data with simple electronics which can be radiation tolerant. A second card in the electronics barracks, where processors can be used without risks. The bandwidth needed between these two cards is easily obtained with usual optical links.

3.1. DPRAM to receive the data

The data received in the Readout Card is written in a Dual Port Random Access Memory (DPRAM). This allows filling it at any time with a new event, while processing previous events. We will use one memory for each link, this is for each front-end card, holding 34 14-bit values per event for the ECAL/HCAL, 68 12-bit values for the PreShower. Note that these values have two bits to identify header and trailer, as described in Section 1.2, so the effective data with is 12 bits for the calorimeters and 10 bits for the PreShower information. As DPRAMs have a larger address space, one can use the high order bits of the address to code the event number, implementing at almost no cost a de-randomisation buffer. It only requires implementing globally on the Readout Card a 'received event number' and a 'processed event number'.

3.2. Zero suppression

Clearly, some flexibility is needed in the way the data are processed. A first requirement is to be able to output all data, without zero suppression, as this may be needed at the beginning of LHCb to understand the detector, and also when performing calibration activities. We also intend to implement two modes of zero suppression.

3.2.1. Simple zero-suppression

A first and simple option for zero suppression is to apply a threshold to the value, and output only those channels over threshold. A label must identify each channel. The simplest output data format is then a sequence of label-data-label-data. This is the preferred scheme for the PreShower.

3.2.2. 2-D zero-suppression

For accurate energy measurement, it is better to measure the tails of the shower, even if the cell energy is close to the noise. In the simple zero-suppression scheme, the threshold is pushed as low as possible to get this information. The idea of the 2D zero-suppression is to read all the neighbouring cells of the cells over threshold, called seeds. We can increase substantially the threshold for the seed, as we will always read the neighbours. The total amount of data can then be similar in both methods, with a better control of the tails of the shower, which could be a serious advantage when reconstructing π^0.

The proposed method is to build a bitmap of the channels to be readout, first by applying a threshold to find the seeds, then by adding the neighbours into the bitmap, and to read back this map to build the output buffer. As the output buffer will contain, by construction, many consecutive cells, an economical output format is to pack the label of the first cell and the number of consecutive cells in a single word, followed by the list of content of those cells. Note that the same format can easily be implemented with the simple zero suppression and with no zero suppression. All algorithms should of course implement the same output format, for obvious simplicity reason!

One difficulty is here again to obtain the neighbours of a cell. The various regions of the calorimeter are handled independently, as it is very difficult to cable the neighbourhood of cells of different sizes, with a size ratio of either 1.5 or 2. The neighbours we are considering are only cells of the same size, whose address is plus or minus one in the row and column number. The small loss of the shower tails will then occur when a shower is near a boundary between two calorimeter regions, but this is almost purely geometrical, and can be simulated.

As long as the neighbouring cells are in the same crate, they can be easily accessed. But a mechanism is needed when the cells are on another crate. However, one doesn't need to transport the data, only the bit pattern of seeds: The decision to output a cell has to be taken on the readout card the cell is being readout. As front-end cards are 8 by 4 cells, and the neighbourhood in the same crate is on the long side of the card, we have to send and receive only 16x4 = 64 bits per crate. The building of the neighbours can proceed only when all these bits are received, which implies a mechanism for validating the received pattern, based probably on the event number. This synchronisation between crate should not add to the latency, as all crates should anyway process usually the same event at the same time.

A possible layout of the crates is shown on Figure 2 where the various crates are shown with various grey levels. The left part shows the repartition in the left part of ECAL, with 3 crates for the external region and two crates for each of the middle and inner region, with a symmetric layout for the right side of the calorimeter. The right part shows the HCAL situation, one crate for the external and one crate for the internal regions. It is clear that in many cases part of the neighbours are in fact in the same crate. However the connection granularity should support the various cases.

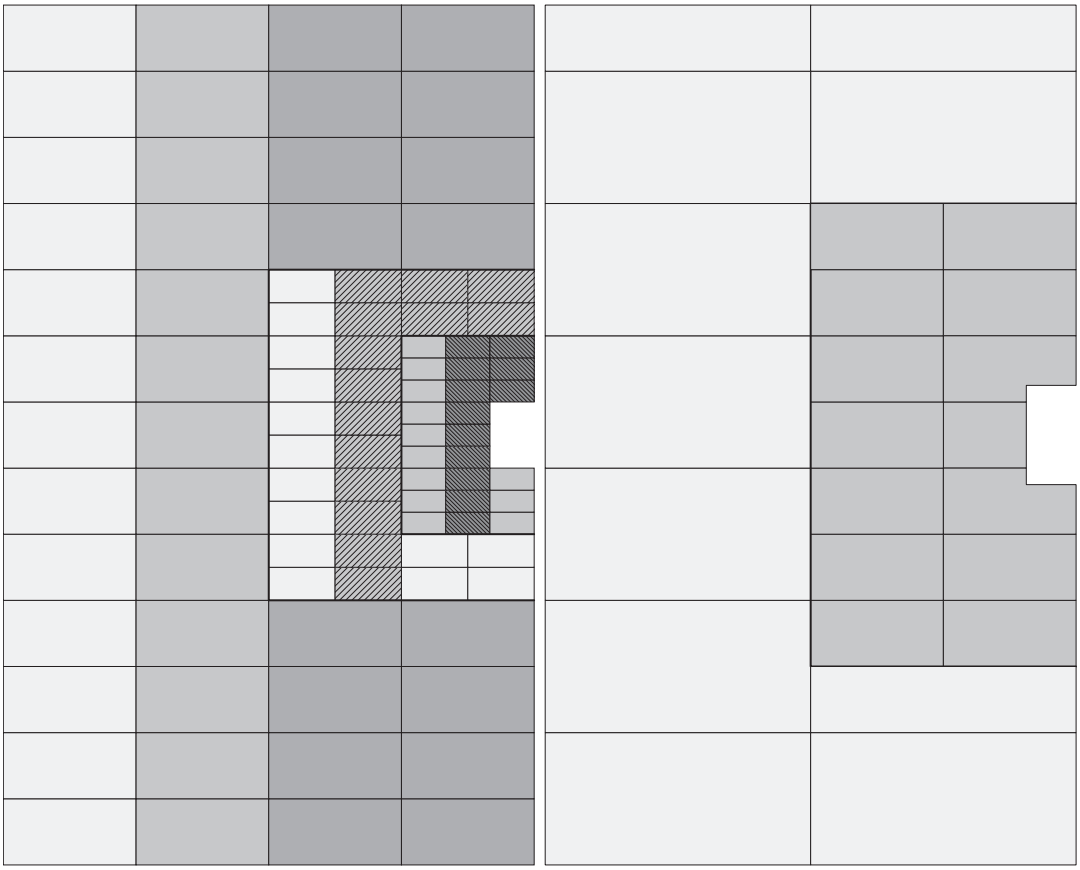

Figure 2: Crate layout, left for ECAL and right for HCAL, left half of the calorimeters.

3.3. Data formatting

The received data is coded on 12 bits. For later processing, it is easier and more efficient to code the content of a calorimeter cell as a floating-point number in physics energy scale, MeV or GeV. A look-up table can convert the 12 bits coding of the ADC content to floating point number. A multiplication by a channel-dependent calibration constant may also be needed to obtain the 'final' energy value. Non linearity is already corrected by the LUT situated after the ADC, as explained in Section 1.

A label must identify each channel. The proposed format, with the bit range, is the following :

	Region	Row	Column
ECAL	13 - 12	11 - 6	5 - 0
HCAL	10 - 10	9 - 5	4 - 0

Our current view is to use 14 bits for the ECAL and PreShower labels, and 11 bits for the HCAL ones. Both formats of labels are built in a similar way: The upper bit(s) identify the region of the calorimeter, one bit for HCAL and two bits for ECAL and PreShower. The next bits are the row and column numbers, which are on 6 bits for ECAL and PreShower but only on 5 bits for HCAL.

As indicated earlier, it is advisable to output only one label for a group of consecutive cells, packing into the same word the label of the first cell and the number of consecutive cells. The format of the output buffer could be as shown on Figure 3, surrounded by the headers and trailers required to describe the type of data and its length.

31-16	15-0
Label of the first cell	Cluster size
IEEE floating data, GeV	
IEEE floating data, GeV	
IEEE floating data, GeV	
……..	
Label of the first cell	Cluster size
IEEE floating data, GeV	
……..	

Figure 3: Proposed output format

3.4. Connection to the DAQ

The resulting data block is then formatted according to the FE multiplexing unit requirement, this means that a header and a trailer in the appropriate format are added. The physical connection is not yet specified, it may be via an optical link or via LVDS multiplexed lines. We will follow the agreed transport interface. The readout protocol is a push-through protocol, this implies no buffering and essentially no control of the output port: Available data is just send out on the link, the next stage is guaranteed to have always-available buffer to store the data.

4. SUMMARY

The readout of the LHCb calorimeters will be performed by dedicated electronics, from the Level 0 pipeline to the final formatted output blocks. A smart zero suppression scheme is foreseen, to get the best physics performance out of the detector. More work is clearly needed to fully specify the CROC, its hardware implementation and the connections between CROCs. The option to split the CROC in two boards, one in the cavern and one in the barracks, has also to be fully investigated.

REFERENCES

[1] *C.Beigbeder et al.*, "The front-end Electronics for LHCb Calorimeters", **LHCb 99-053**, 31 December 1999

[2] *C.Beigbeder et al.*, "The front-end Electronics for the Calorimeter Triggers", **LHCb 2000-010**, 27 March 2000

[3] *J.Christiansen*, "Requirements to the L0 front-end electronics", **LHCb 99-029**, 22 November 1999