Available on CMS information server **CMS NOTE 2006/146**

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

November 16, 2006

Implementation of on-line data reduction algorithms in the CMS Endcap Preshower Data Concentrator Card.

D. Barney[1], W. Bialas[1], P. Kokkas[2], N. Manthos[2], S. Reynaud[1],
G. Sidiropoulos[2*], P. Vichoudis[1*]

[1]*CERN, 1211 Geneva 23, Switzerland*
[2]*University of Ioannina, Ioannina, Greece*

ECAL Group

Abstract

The CMS Endcap Preshower (ES) sub-detector comprises 4288 silicon sensors, each containing 32 strips. The data are transferred from the detector to the counting room via 1208 optical fibres running at 800Mbps. Each fibre carries data from 2, 3 or 4 sensors. For the readout of the Preshower, a VME-based system - the Endcap Preshower Data Concentrator Card (ES-DCC) is currently under development. The main objective of each readout board is to acquire on-detector data from up to 36 optical links, perform on-line data reduction (zero suppression) and pass the concentrated data to the CMS event builder. This document presents the conceptual design of the Reduction Algorithms as well as their implementation into the ES-DCC FPGAs. The algorithms implemented into the ES-DCC resulted in a reduction factor of ~20.

Corresponding authors: Sidiropoulos Georgios, e-mail address: Georgios.Sidiropoulos@cern.ch
Vichoudis Paschalis, e-mail address: Paschalis.Vichoudis@cern.ch

1 Introduction.

The CMS Preshower [1] is a fine grain detector placed in front of the endcap Electromagnetic calorimeter. Its primary function is to detect photons with good spatial resolution in order to perform π^0 rejection. The detector comprises 4288 63mm x 63mm silicon sensors, each of which is divided into 32 strips. The micromodule [1], building unit of the CMS Preshower sub-detector, comprises a silicon sensor DC-coupled to a PCB hybrid containing the PACE3 [2] front-end electronics, all mounted on ceramic and aluminum support structures. The signals from each strip of the micromodule are amplified, shaped (peaking time around 25ns) and sampled continuously every ~25ns (40.08MHz) and temporarily stored in an analogue memory by the PACE3. On reception of a first-level trigger three consecutive time samples (on the baseline, near the peak and after the peak) are driven out of the micromodule and digitized by a 12-bit AD41240 ADC [3] on the Preshower 'system mother-board' (SMB). The digitized data from up to 4 micromodules are multiplexed and formatted* efficiently (in terms of bandwidth) by a K-chip [4] and transmitted through an optical link via the GOL [5] serializer ASIC to the ES-DCC in the Counting Room.

The data transport from the on-detector system is achieved by 1208 optical fibres running at 800Mbps. This enormous data flow necessitates a significant reduction in the data volume. The occupancy is relatively low in the Preshower - an average of about 2% at high luminosity ($2x10^{34}cm^{-2}s^{-1}$) and most "interesting" signals (from electron/photon showers) have large pulse heights - equivalent to tens or hundreds of MIPs (minimum ionizing particles). However, these signals can be spread over several strips, with the edge strips having quite low pulse-heights (few MIPs). A relatively low threshold (typically 3-sigma of the pedestal noise, equivalent to about 1 MIP in normal running) removes a majority of random (noise) hits whilst retaining virtually all real signals, including the signals in strips at the edges of clusters, which are important for photon-π^0 rejection. For the readout of the detector, a VME-based system - the Endcap Preshower Data Concentrator Card (ES-DCC) [6], is currently under development. The main objective of the ES-DCC is to acquire on-detector data from up to 36 optical links, perform on-line data reduction (zero suppression) and pass the concentrated data to the CMS event builder. The algorithms implemented into the ES-DCC resulted in a reduction factor of ~20, assuming high luminosity.

The readout chain of the Preshower is shown in Fig. 1.

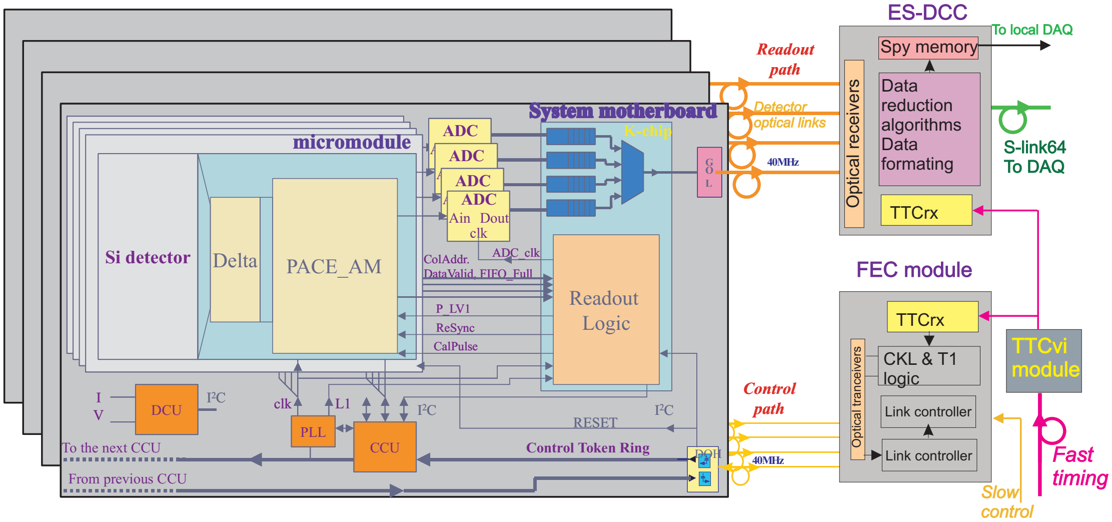

Figure 1: The Preshower Readout chain.

The major ES-DCC components are the following:

- Three FPGAs for the de-serialization of the input data streams (from the 36 optical links) and the reduction of the data volume. Twelve input data streams are treated by each FPGA. The FPGAs are compatible with the Gigabit Ethernet (8b/10b encoding) protocol supported by the GOL. These three FPGAs will be referred-to as "Reduction FPGAs".

* The K-chip organizes the 12-bit data from up to 4 micromodules in 16-bit word packets. The packet length is 299 words.

- One FPGA for merging the zero-suppressed data coming from the 3 FPGAs performing the data reduction as well as for building the ES-DCC event. This FPGA will be referred-to as “Merger FPGA”.
- An S-link [7] transmitter mezzanine card for transmitting the ES-DCC event to the global CMS DAQ system.
- Three FPGAs and sufficient memory chips for event monitoring through the VME bus. These three FPGAs will be referred-to as “Spy FPGAs”.
- A special circuitry, based on the TTCrx [8] ASIC for receiving the necessary timing, trigger and control signals. These signals will be referred-to as TTC signals.

The hardware architecture of the ES-DCC is shown in Fig. 2.

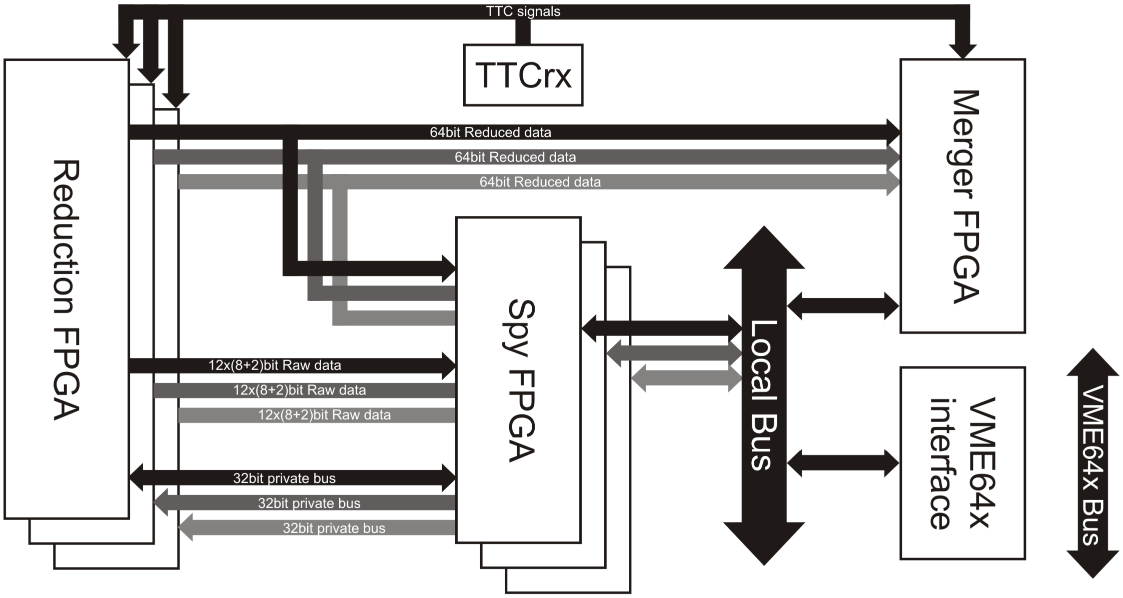

Figure 2: Simplified Block diagram of the ES-DCC hardware.

2 The data reduction stages.

The data reduction requirements for the Preshower and some preliminary algorithms studied the last 2-3 years (in the University of Ioannina) have shown that the resources required were more than available in an existing VME-based readout hardware, necessitating the development of the ES-DCC. These algorithms have been modified and refined and now include the following stages:

- De-serialization of the input data streams using embedded FPGA high speed de-serializers configured in Gigabit Ethernet mode [9].
- Integrity check of the incoming data based on the CRC information included in the packets [4]. The polynomial used is the $x^{16}+x^{12}+x^5+1$.
- De-multiplexing and de-formatting of the incoming data packets in order to extract the strip data and other information (time stamps, error flags etc.).
- Subtraction of the mean value of the digitized signal when there is no particle signal present (pedestal) for each channel. The pedestals are recalled from the CMS Conditions Database and downloaded to the ES-DCC lookup tables.
- Calibration of each channel based on a predefined coefficient is being performed since there is an inter-channel spread in the gain of the PACE3 pre-amplifiers. The calibration coefficients are recalled from the CMS Conditions Database and downloaded to the ES-DCC lookup tables.
- Calculation and removal of the mean baseline shift of the zero signal level due to electromagnetic interference which is common to the channels of each micromodule (referred to as common mode). Common mode is calculated on a sample-by-sample basis as the mean value of all “non-hit” strips. A “non-hit” strip is

defined as a strip with a value under a certain “non-hit” threshold. Although several definitions of the “non-hit” threshold were investigated [10], the least resource/time-consuming approach defines the “non-hit” threshold as the minimum strip value (per time sample) increased by a constant value. The constant is calculated off-line based on the pedestals sigma (typical value: 6 sigmas) and downloaded in the ES-DCC. The pedestal sigma is derived from the average of many thousand events during a calibration run. It is worth mentioning that this procedure is not influenced by the noisy strips, since they can be masked*. Fig. 3 illustrates graphically a common mode correction example.

- Bunch-crossing assignment in order to remove signals from bunches not corresponding to the trigger (from previous and next events). If the second time sample (of the three consecutive ones taken in steps of ~25ns per event) is not higher than the other two, all three time samples are discarded (see Fig. 4).
- Selection of useful data by applying a “zero-suppression” threshold (typical value: 3 sigmas) on the middle time sample. The “zero-suppression” threshold is calculated off-line based on the pedestal sigma and downloaded in the ES-DCC. If the middle time sample is higher than the “zero-suppression” threshold, all three time samples are forwarded to next stage with a 18-bit identification header. It is worth mentioning that there is no need for a different “zero-suppression” threshold per strip, since the pedestal sigma (noise) is uniform. Concerning the noisy strips, they are suppressed by masking (as mentioned previously).
- Collection of the sparsified data and local event formatting prior sending reduced packets to the CMS DAQ system. The collection of the data is performed in two parts. The first part is the creation of a packet from the reduced data inside each “Reduction FPGA” that belong to the same event. This part is very significant as it helps to avoid the possible de-synchronization of the system due to missing or spurious events. The second part deals with collating the 3 data packets in the “merger FPGA”, building this way the events locally.

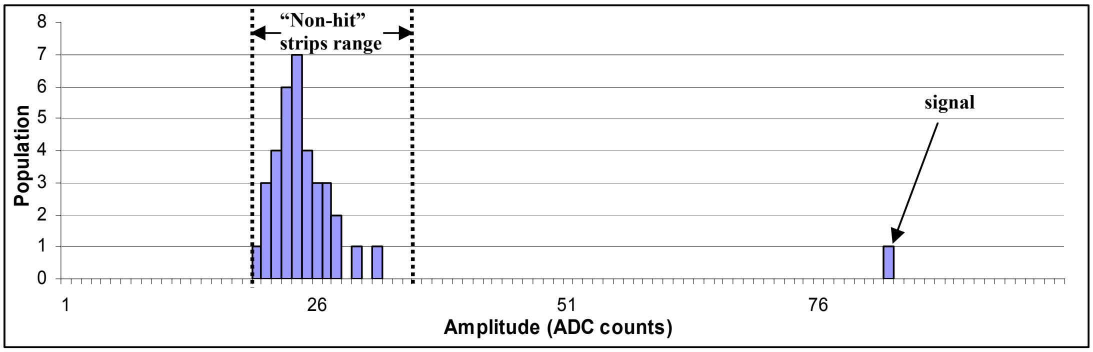

Figure 3: Common Mode Correction Example. The histogram shows the 32 strip values of a micromodule after the pedestal correction. The left dashed line in the graph indicates the minimum strip value (20 ADC counts) and the upper one the resulting “non-hit” threshold (35 ADC counts, assuming the pedestal sigma is 2.5 counts and a constant is defined as 6 sigmas). By averaging the strips below the “non-hit” threshold, the common mode results 24 ADC counts.

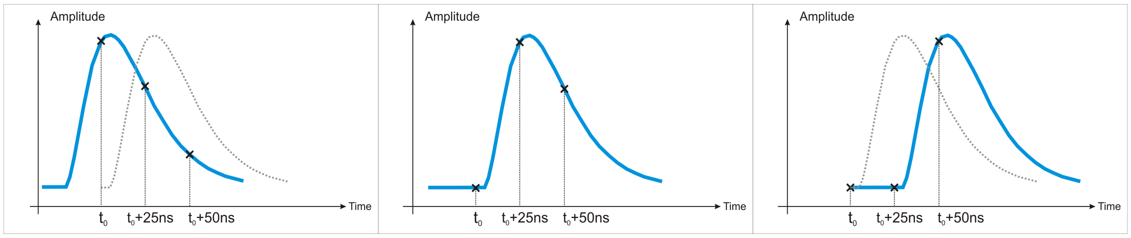

Figure 4: Bunch crossing assignment. The middle graph shows a signal (with realistic width) from the bunch corresponding to the trigger (2nd time sample>3rd time sample & 2nd time sample>1st time sample).The left graph shows a signal from the previous bunch (2nd time sample>3rd time sample & 2nd time sample<1st time sample). The right graph shows a signal from the next bunch (2nd time sample<3rd time sample & 2nd time sample=1st time sample). The three time samples from the left and right graphs are discarded.

* Off-line analysis will determine noisy strips and put this information into the configuration database. This information is downloaded to the ES-DCC.

3. Implementation of Data Reduction.

Most of the required functions for the ES-DCC are implemented in the so-called "Reduction FPGA" and for this reason this manuscript is focused on its architecture. Fig. 5 illustrates the internal architecture of the Reduction functions. The design incorporates twelve individual reduction chains running in parallel to process data from 12 optical links. The de-serialized data from each stream are fed to the associated reduction chain. At the same time, these data appear on the Raw Data Bus output pins for event monitoring purposes. After applying the reduction algorithms the reduced information is gathered and organized per event by the internal merger and forwarded to the "Merger FPGA" through a 64bit bus (Reduced Data Bus). All the necessary settings and working parameters are configured through the private bus interface.

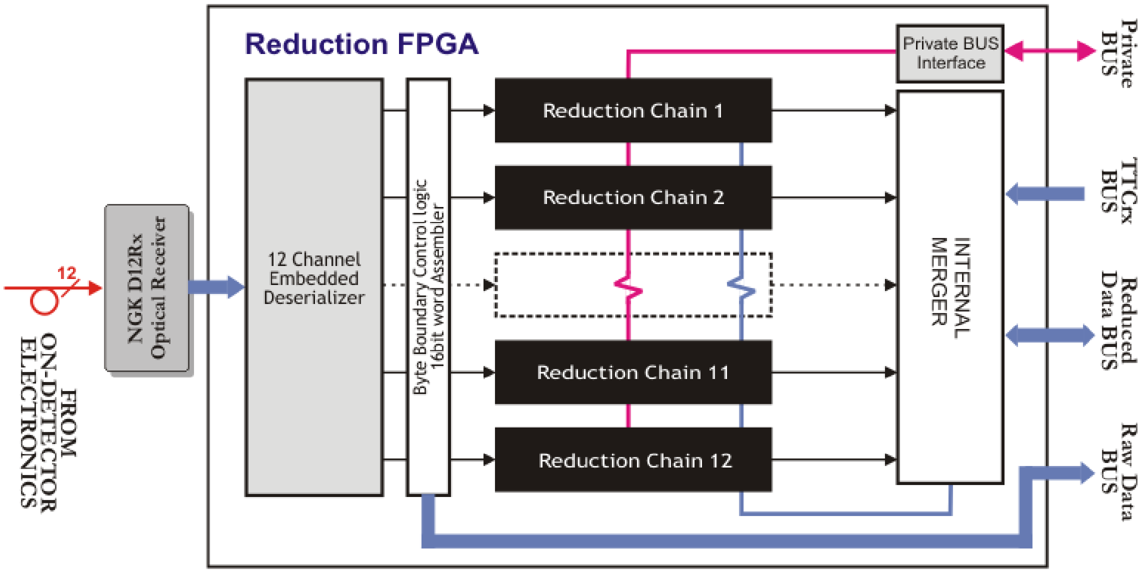

Figure 5: Internal architecture of the "Reduction FPGA".

Fig. 6 illustrates the data manipulation inside each reduction chain. The de-serialized data (after being re-assembled*) are fed to the packet inspection machine. The packet inspection machine evaluates the type, length and integrity of the incoming packets and generates the corresponding Error/Status flags. These flags, as well as the packet header, are stored in a FIFO memory (INFO FIFO) and recovered by the internal merger at a later stage. The strip data of the incoming packets are stored to another FIFO memory (DATA FIFO). Data exiting the DATA FIFO are re-assembled in 12bit words. The 12bit strip-data are then "corrected" by subtracting the pedestal and multiplied by a calibration coefficient. The individual values for the pedestals and the calibration coefficients are retrieved from a common dual-port Lookup Table.

After the signal correction, the strip data are fed to three individual dual-port memories, one for each of the three time samples taken. Another dual-port memory (SWAP dpRAM) works as a "scratch" storage area where data are exchanged with the Common Mode calculator machine. During the Common Mode calculation, the averaging of the strip values under the "non-hit" thresholds takes place for the 4 micromodules and the corresponding 4 results (common mode values) per time sample are stored to another dual-port memory (CM dpRAM). When all the 12 (4 x 3 time samples) common mode values are calculated, they are recalled from the CM dpRAM and subtracted from the data stored in the three time sample memories. After the common mode correction, the bunch crossing assignment and the "zero-suppression" threshold application is performed in order to select the useful data. The useful data together with the information stored previously in the INFO FIFO are saved to a FIFO (OUTPUT FIFO) waiting to be accessed later from the internal merger machine. As can be observed from Fig. 6 each reduction chain requires nine memory blocks (FIFOs and dpRAMs) of various sizes, resulting in a quite large number of block memories needed for the whole design. The total number of memory blocks was the decisive factor for choosing the ALTERA Stratix GX FPGA family [11].

* The serializer ASIC splits the 16-bit words (driven by the K-chip) in two bytes and encodes them according to the 8b/10b encoding scheme before transmitting the serial stream to the ES-DCC. Therefore, the recovered (de-serialized) bytes in the ES-DCC have to be re-assembled to 16-bit words in order to reproduce the 16bit-wide transmitted data from the K-chip.

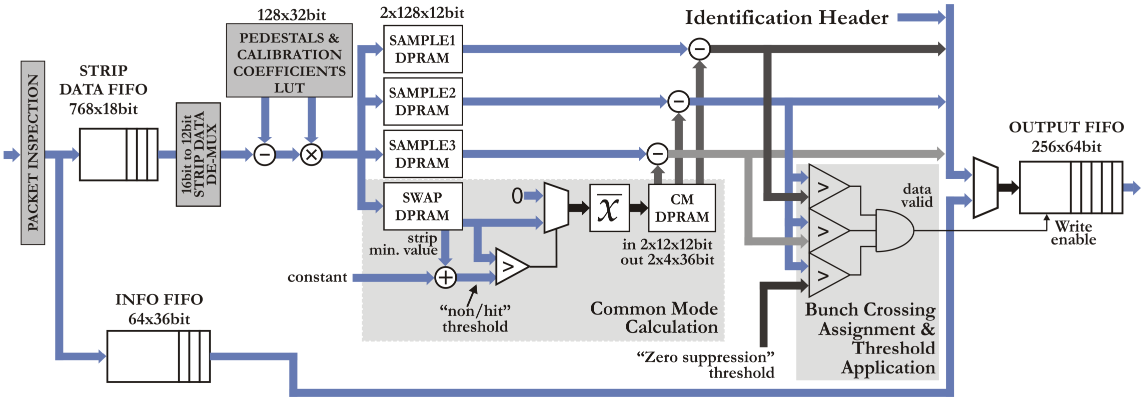

Figure 6: Block diagram of the architecture for one reduction chain.

Fig. 7 shows a diagram illustrating the timing relation between the procedures shown in Fig. 6. The rectangles with the "bricks" texture symbolize the 16bit-to-12bit de-multiplexing, pedestal subtraction, channel calibration and storage of one time sample (from the 4 micromodules). This procedure takes 128 cycles (32 strips x 4 micromodules x 1 cycle). The rectangles with the "basket-weave" texture symbolize the common mode calculation and common mode values storage for one time sample. This procedure takes 256 cycles (32 cycles x 4 micromodules to find the minimum value + 32 cycles x 4 micromodules to average the values below "non-hit" threshold). The rectangles with the "spiral" texture symbolize the common mode subtraction, bunch crossing assignment and "zero-suppression" threshold application. This procedure takes 128 cycles (32 strips x 4 micromodules x 1 cycle). Although the time for the processing of one packet seems to be 1024 cycles, the 256-cycle overlap between two consecutive packets results in an actual processing time of 768 cycles. The 256-cycle overlap is achieved by accessing the two pages of the dpRAMs in turn. It is worth mentioning that since the processing clock frequency is 120MHz, the total packet processing time of 6.4μs (768 cycles @ 120MHz) is significantly lower than the constant packet arrival time* of ~7.5μs (300 cycles @ 40MHz). Thus, no overflow can occur due to the data reduction processing.

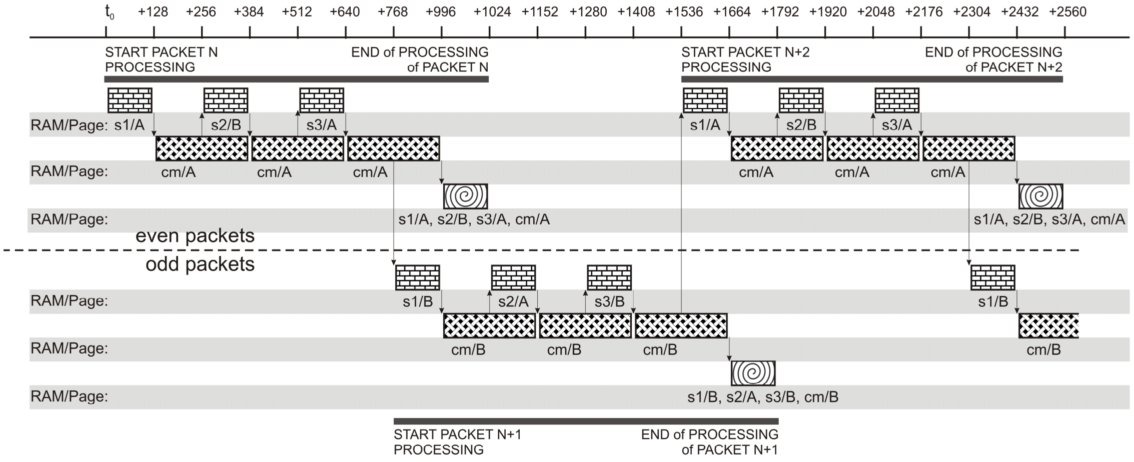

Figure 7: Timing diagram. The end of processing for packets N, N+1 & N+2 occur at t_0+1024cycles, t_0+1792 & t_0+2560, respectively.

The purpose of the internal merger machine in Fig. 5 is to collect the reduced packets belonging to the same event, from the twelve reduction chains in turn and store them to a single FIFO (MERGER FIFO). In order to achieve that, the merger compares the time packet time stamp (available in the OUTPUT FIFO) with the absolute time stamp coming from the onboard TTCrx [8] circuitry. In case of a match, the merger collects the packet and moves to the next chain. When the time stamp of the packet shows that it was recorded earlier than expected (compared to the absolute time stamp), the packet is considered as spurious. In this case, the contents

* 299 words per K-chip packet x 25ns = 7.475μs.

of the packet are discarded and the merger keeps searching for the corresponding packet in the same channel (until time-out occurs). On the contrary, when the time stamp shows that the packet was recorded later than expected, it is considered that the corresponding packet is lost and the packet under examination belongs to a later event. It is worth mentioning that the Merger FPGA in Fig. 2 uses the same technique to synchronize the data packets from the 3 reduction FPGAs.

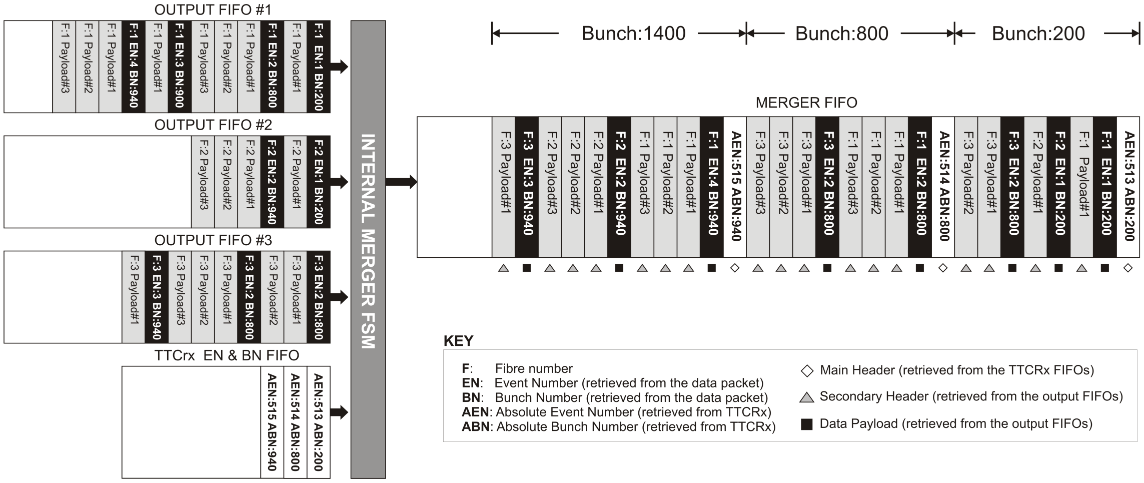

Figure 8: Output FIFOs vs Merger FIFO (example with three channels).

4. Conclusion.

A set of algorithms for online data reduction during the readout of the CMS Preshower detector has been developed and implemented using FPGAs. The set includes pedestal removal, channel calibration, common mode rejection, bunch crossing identification and zero suppression algorithms. Although the targeted FPGA technology for the Data Reduction has been already specified, the firmware has been written using the VHDL language in order to be portable among the various vendor technologies. The main guideline for this design is to reuse the resources as much as possible when time allows and to operate at high clock rates thus decreasing the total processing time. This design strategy allows all necessary data reduction functions to be carried out within the available time thus reducing to zero the possibility of overflow.

5. Acknowledgements.

The authors would like to thank Ian Tomalin, Egidio Longo and Roberto Tenchini for reviewing this manuscript.

References.

[1] P. Wertelaers et al, “ECAL Preshower Engineering Design Review”, CMS ECAL EDR-4, Vol.2, 2000-054 http://edms.cern.ch/document/115565/1

[2] P. Aspell et al, “PACE3 : A large dynamic range analogue memory ASIC assembly designed for the readout of silicon sensors in the LHC CMS Preshower”, 10th Workshop on electronics for LHC experiments (LECC2004) 137-141.

[3] G. Minderico, C. Fachada, I-L Chan, K-M Chan, H-M Cheong, A. Lopes, P. Cardoso, J. Vital, K. Kloukinas, A. Marchioro “A CMOS low power, quad channel, 12 bit, 40MS/s pipelined ADC for applications in particle physics calorimetry”, 9th Workshop on electronics for LHC experiments (LECC2003) 88-91.

[4] K.Kloukinas et al, “Kchip: A Radiation Tolerant Digital Data Concentrator chip for the CMS Preshower Detector”, 9th Workshop on electronics for LHC experiments (LECC2003) 66-70.

[5] P. Moreira et al, “G-Link and Gigabit Ethernet Compliant Serializer for LHC Data Transmission” presented at the IEEE Nuclear Science Symposium and Medical Imaging Conference, 2000.

[6] P. Vichoudis et al, “A VME-based readout system for the CMS Preshower sub-detector”, presented at the IEEE Nuclear Science Symposium and Medical Imaging Conference, 2006.

[7] H.C. van der Bij et al, “S-LINK, a Data Link Interface Specification for the LHC Era”, IEEE Trans. Nuclear Science, Vol. 44, Issue 3, Part 1, June 1997, pp. 398-402.

[8] J. Christiansen, A. Marchioro, P. Moreira and A. Sancho “Receiver ASIC for Timing, Trigger and Control Distribution in LHC Experiments”, IEEE Trans. Nuclear Science, Vol. 43, june 1996, pp. 1773-1777.

[9] IEEE standard 802.3, 1998 edition.

[10] N. Manthos, G. Sidiropoulos, P. Vichoudis, “An Efficient Hardware Design for Rejecting Common Mode in a Group of Adjacent Channels of Silicon Microstrip Sensors Used in High Energy Physics Experiments”, IEEE Trans. Nuclear Science, Vol.53, june 2006, pp.1045-1050.

[11] S. Reynaud, P. Vichoudis, “A multi-channel optical plug-in module for gigabit data reception”, presented at the 12th Workshop on electronics for LHC experiments, september 2006.