

An Efficient Hardware Design for Rejecting Common Mode in a Group of Adjacent Channels of Silicon Microstrip Sensors Used in High Energy Physics Experiments

Nikolaos Manthos, Georgios Sidiropoulos, and Paschalis Vichoudis

Abstract—Algorithms have been studied using Monte Carlo techniques and implemented in a fast Xilinx Virtex II pro field programmable gate array (FPGA), in order to calculate and remove, after pedestal subtraction, the common mode of a group of adjacent channels. The implementation of the algorithms has been optimized both for speed and minimal FPGA resources, so as to be used in multi-channel applications. The aim of this work is to define the optimum algorithm for common mode calculation to be implemented for common mode rejection in the CMS Preshower detector.

Index Terms—Algorithm, common mode, fast sorting, field programmable gate array.

I. INTRODUCTION

THE readout chain of the detectors based on microstrip sensors includes front-end (FE) electronics for amplification and shaping of the signal induced on the strips by the passage of a charged particle through the sensor or by the absorption of a photon. The signals usually are then digitized and sent out of the detector for additional processing or storage.

The common mode in this work is defined as the time dependent mean base line shift of the channel pedestals, i.e., of the signal level of the channel without particle charge. This shift is eventually common for a number of adjacent channels, due to internal or external EM sources (ground bounce, external strip-cable lines acting as RF antennas etc). Each channel is considered to include a sensor strip, an amplifier-shaper, and an analog memory in the FE chip, a multiplexing stage in the FE output and an ADC. The input of the algorithm calculating the common mode is considered to be the digitized signal values of the channels in a group of adjacent channels after pedestal subtraction. The channel pedestals (mean base line level) are usually measured using a number of events without particle charge in predefined time intervals short enough to take into account

any eventual significant variation. Although the algorithm and its implementation presented here can be applied to remove either the common mode in any electronic detector or the time dependent background of an image, it has been developed in order to reject the common mode in the readout chain of the CMS silicon Preshower detector [1].

The CMS silicon Preshower is a fine grain detector placed in front of the endcap calorimeter ECAL. Its primary function is to detect photons with good spatial resolution in order to perform π^0 rejection required in the search for Higgs bosons.

Each silicon sensor has a total active area of $61 \times 61 \text{ mm}^2$ and is divided into 32 strips of 1.9 mm pitch, with strip capacitance in the region of 50 pF. The PACE3 chip [2], a large dynamic range, two-gain FE ASIC, is used for amplification, shaping, and temporary storage of the analogue signals. The 32 strip signals are multiplexed on demand by the CMS first level trigger and sent out to a 12-bit ADC, AD41240 [3]. The digitized data from a group of up to 4 sensors are multiplexed [4] and sent to the CMS off-detector electronics through an optical link. In the off-detector electronics the data reduction algorithms are applied in order to propagate the useful part of data to the CMS DAQ event builder for further online analysis and storage.

II. METHOD OF COMMON MODE CALCULATION

Various methods have been used for common mode estimation of a group of adjacent channels in electronic detectors used in High Energy Physics [5]–[13]. Some of them are based on the differences between the pulse heights of each channel and the corresponding mean value of all channels in the group. Other methods estimate the common mode using the median pulse height. The main difficulty in common mode estimation is the distinction between the channels having particle induced charge and the channels having common mode only, particularly in cases where individual channel noise is high.

In this work the common mode has been calculated for groups of 16 channels and therefore the digitized signals of each Preshower sensor have been divided into two groups of 16 channels each. In an earlier study a method, using a cut in the difference between the ADC value of each strip and the mean value of the group of strips has been studied and implemented in an FPGA, showing a limitation especially for low ADC values of the signal [13]. In the present work a more efficient method has been studied and implemented in an FPGA. According to this method:

Manuscript received June 17, 2005; revised March 14, 2006. This work was supported in part by the program “Heraklitos” of the Operational Program for Education and Initial Vocational Training of the Hellenic Ministry of Education under the 3rd Community Support Framework and the European Social Fund.

N. Manthos and G. Sidiropoulos are with the Physics Department, University of Ioannina, Ioannina GR 45110, Greece (e-mail: nmanthos@cc.uoi.gr; me00569@cc.uoi.gr).

P. Vichoudis is with the Physics Department, University of Ioannina, Ioannina GR 45110, Greece, and also with the CERN/PH Geneva 23, CH1211, Switzerland (e-mail: paschalis.vichoudis@cern.ch).

Digital Object Identifier 10.1109/TNS.2006.874040

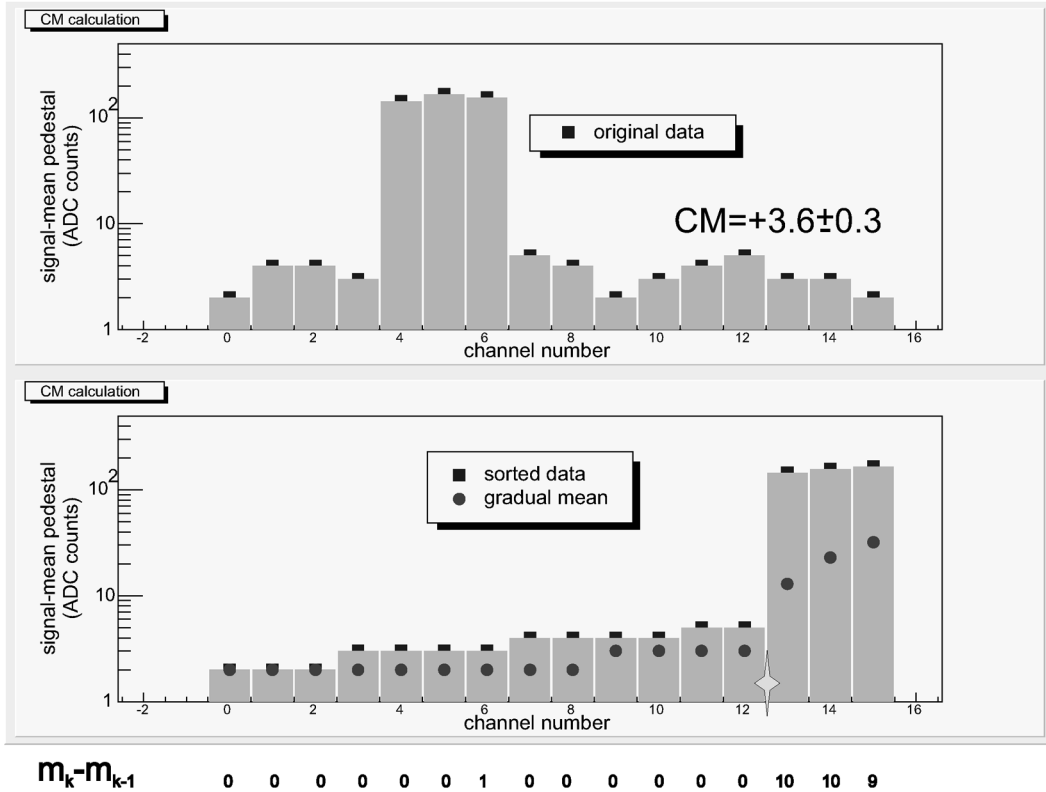


Fig. 1. Demonstration of the common mode calculation method. See details in the text. Note that the vertical scale is logarithmic.

- Fast ascending sorting of the 16 ADC values (v_i) after the mean pedestal subtraction is performed. A sorting algorithm which will be explained later in this work was used to define the first part of the sorting list, which eventually does not include particle charge signals. This part is used to calculate the common mode. The length of the first part of the list is calculated in the following steps.
- Calculation of the gradual mean $m_k = (v_1 + \dots + v_k)/k$ of the sorted values v_i , where $k = 1$ to 16.
- If $m_k - m_{k-1} < c_1$ AND $m_{k+1} - m_k < c_2$, where c_1, c_2 are constants, then the common mode is m_k .

If the mentioned criterion is fulfilled for the first time starting from $k = 1$, it ensures that the strips used to calculate the common mode do not include signals from particles. The constants $c_1 \approx c_2$ depend on the channel noise (pedestal rms) as well as on the number of non-hit strips. They are used for finding the position of the channel with the lowest particle signal in the sorted list (position $k + 1$ in the list). Gradual mean values have been used in order to smooth out any fluctuation in the common mode among the adjacent channels.

Fig. 1 demonstrates the above method. The first plot shows the ADC values v_i of the 16 channels where 3 of them ($i = 5$ to 7) have particle signal (with common mode) and the rest of them have only common mode. The second plot shows the previous values sorted, together with the gradual mean. At the bottom of the plot the $m_k - m_{k-1}$ values are shown. In this case, if $c_1 = c_2 = 2$ is used, the length of the first part of the sorted list can be determined and the common mode can be calculated (3 ADC counts, due to the fact that quotient has been truncated).

It is worthwhile to mention that the ADC values after pedestal subtraction are considered to be integers. Although this is an approximation and the division used to calculate the gradual mean has been approximated using multiplication and right bit shifting to speed up the common mode calculation procedure, the precision in the calculation remains acceptable.

This method has been simulated using estimated values for the pedestal and common mode variation. The performance of the method, as shown in Fig. 2, is satisfactory. Fig. 2 depicts the difference between the calculated and input common mode. For the left plot, as input to the simulation, normally-distributed pedestals have been used with channel noise 7 ADC counts (estimated for high gain in the PACE3 without any extra shielding of the on-detector electronics) as well as a normally-distributed common mode with mean 5 ADC counts and rms 10 ADC counts. In addition a common mode variation, $\sim 25\%$ of the mean common mode, has been added to the 16 strips. Signals from the decay of Higgs ($300 \text{ GeV} \rightarrow Z(Z^*) \rightarrow 4 \text{ leptons}$) have been used, mixed with the appropriate minimum bias events for LHC luminosity $2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$. $c_1 = c_2 = 3$ was used in the method to calculate the common mode.

The rms of the distribution in Fig. 2 (left) is ~ 1.8 ADC counts, which is quite low if one takes into consideration that a single minimum ionizing particle (MIP) produces a signal of ~ 50 ADC counts for PACE3 high gain.

For the right plot in Fig. 2, normally-distributed pedestals have been used with rms 3 ADC counts (estimated for low gain in the PACE3) as input to the simulation. The input common mode is similar to that used for the left plot. The rms of the dis-

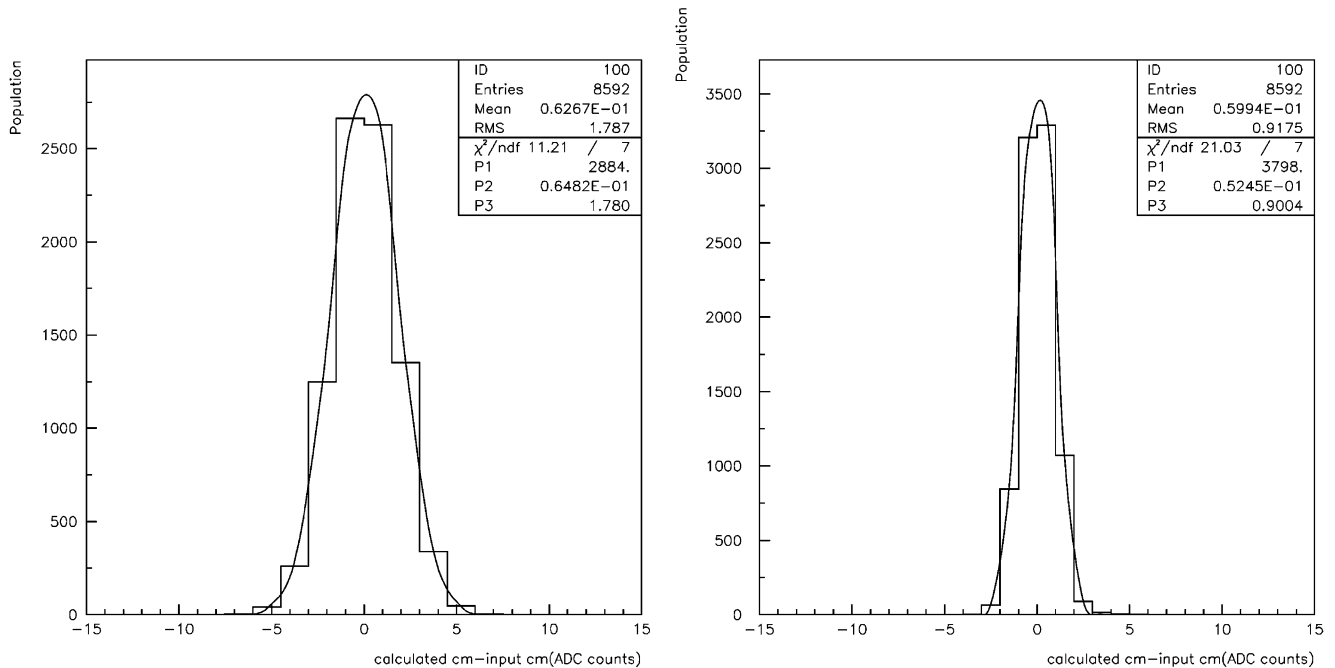


Fig. 2. Performance of the common mode calculation method: Difference between the calculated and input common mode using simulated Preshower data (Details in the text). Left: for PACE3 high gain. Right: for PACE3 low gain.

tribution is ~ 0.9 ADC counts, compared to ~ 8 ADC counts corresponding to a MIP for PACE3 low gain.

III. IMPLEMENTATION

The algorithms have been developed using VHDL and implemented in a XC2VP7 Virtex-II Pro Xilinx FPGA. The implementation of the algorithm is under a trade-off between processing time and resources occupancy since it will be used for multi-channel application. The 2VP7 FPGA, includes 8 RocketIO transceivers, 792 Kb of block and 154 Kb distributed RAM, 44 18×18 multipliers and 1 PowerPC processor. The Xilinx ISE Foundation tool has been used for the implementation together with the Synplicity Synplify Pro synthesizer. ModelSIM simulator by Mentor Graphics has been used for the verification of the algorithm.

The most time consuming part of the algorithm is the sorting procedure. Different sorting methods have been tested. It seems that there is a trade-off between fast sorting time and minimum logic requirements. The method we concluded to employ is the selection sorting method in conjunction with Gray encoding/decoding. This method has the minimum logic requirements and a satisfactory sorting time.

The sorting is performed in a bit-sequential mode using dual RAM banks [14]. An offset has been added at the start of the procedure to the ADC values in order to have positive integer numbers only. The positive integers are transformed to Gray codes using the $G_k = \text{XOR}(b_k, b_{k-1})$, where b_k is the k -th bit of the number and G_k the corresponding Gray code bit. At the end of the procedure the Gray codes are transformed back to binary using $b_k = \text{XOR}(b_{k-1}, G_k)$ and the added offset is subtracted.

The improved sorting method used is the following.

- The elements enter the 1st of the two banks (pages).
- A state machine sequentially reads all values from the 1st bank, and values with LSB equal to 0 go into the 2nd bank beginning from top to the bottom, while values with LSB equal to 1 go into the 2nd bank beginning from bottom to top. Therefore, after this step, the 2nd bank includes values with LSB equal to 0 followed by values with LSB equal to 1 (i.e., all values are sorted in respect to the LSB).
- The state machine sequentially reads all values from the 2nd bank. Values with $\text{LSB}+1$ equal to 0 go into the 1st bank from top to the bottom. While values with $\text{LSB}+1$ equal to 1 go into 1st bank from bottom to top.
- Steps b and c are repeated checking the rest of the bits ($\text{LSB} + j$), $j = 2$ to MSB. If the width of the values is even (12-bit for the Preshower) the 1st bank comprises the final sorted array.

This method is very fast compared with the bubble sort method and processing time is always the same, $n \cdot m$, where n is the number of the elements to be sorted (16 for the Preshower) and m is the number of bits in each element (12 bits for the Preshower). Therefore for the CMS Preshower case the processing time is 192 clock cycles. This implementation can run with a clock up to 160 MHz and therefore its sorting time is $1.2 \mu\text{s}$ for 16 numbers. If the maximum actual length of the numbers is less than 12 bits (in the case that no particle signal is present or the induced charge is low) the procedure is executed faster, using a skipping circuit to determine the maximum actual length of the numbers. The resources occupied in the FPGA are 39 logic slices i.e., 2% of the XC2VP7 FPGA. It is worth mentioning that the method described in [14] requires twice the amount of time to perform the sorting in comparison to the time required by the current method. A demonstration of

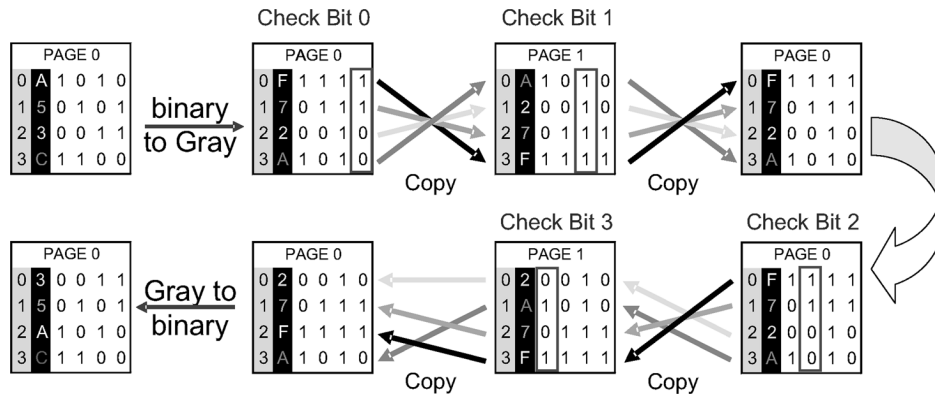


Fig. 3. Demonstration of the selection sorting method using Gray encoding/decoding. See details in the text.

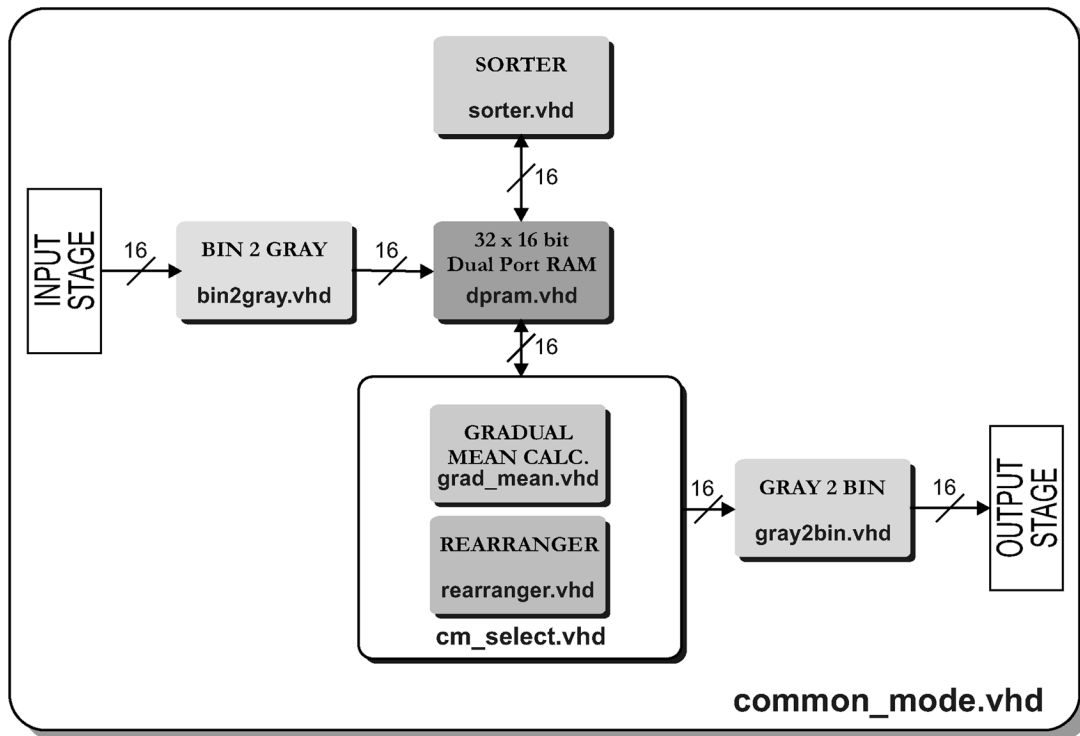


Fig. 4. Block diagram of the common mode calculation procedure.

the sorting method of four 4-bit numbers is shown in Fig. 3. The hexadecimal numbers to be sorted are A, 5, 3, C. When they are converted to Gray codes become F, 7, 2, A, and after sorting 2, 7, F, A. The Gray codes are then converted to binary and the resulting list is 3, 5, A, C.

A multiple-register sorting method has also been tested to be used for sorting [14]. This method was rejected for the current work due to the extremely high amount of logic resources needed, even though it requires only 16 clocks for 16 numbers.

The block diagram of the implemented full procedure calculating the common mode is shown in Fig. 4. The procedure occupies 368 slices of the FPGA resources (i.e., the 7.5%) and it is executed in 1.45 μ s.

In order to minimize the execution time no direct division is used for finding the gradual mean, but a multiplication using a lookup table (LUT) followed by a bitwise right shifting. In particular, for a division by a power of 2, right shifting is per-

TABLE I
LUT FOR HARDWARE DIVISION

| Divisor | 3 | 5 | 6 | 7 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|--------------------|----|----|----|----|----|----|----|----|----|----|----|
| Multiplying factor | 85 | 51 | 43 | 37 | 28 | 26 | 23 | 21 | 20 | 18 | 17 |

formed to the bits of the dividend by the corresponding number of bits. For a division by a number between 3 and 15 the division has been replaced by a multiplication with the corresponding number in a LUT table shown in Table I followed by right bit shifting of 8 bits (division by 256). This approximation (error less than 1%) dramatically increases the speed and also keeps the FPGA occupancy and required execution time low having no compromise to the accuracy.

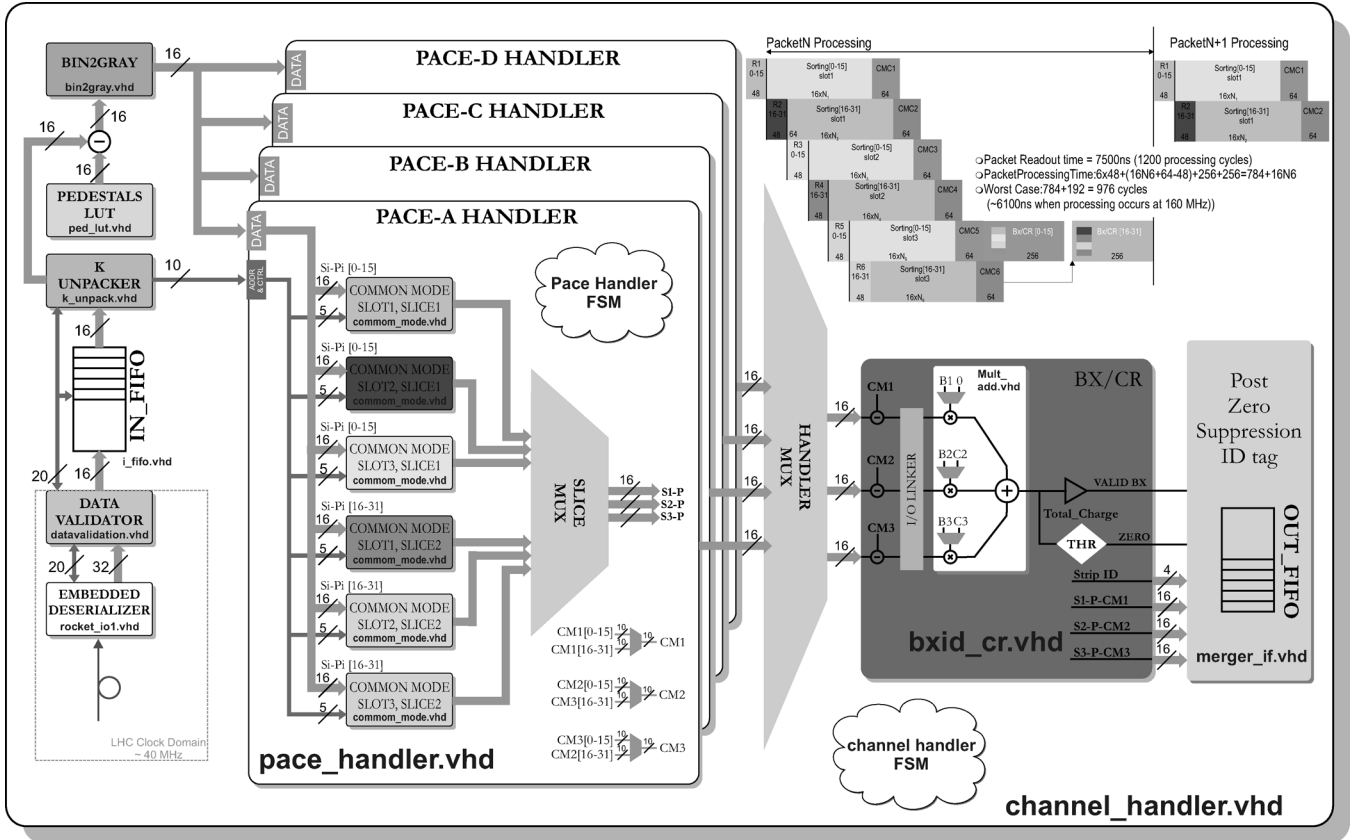


Fig. 5. Block diagram of the implemented off-detector electronics functionality of the Preshower.

In addition to the common mode of the adjacent channels (mean common mode) the rms of the common mode is calculated in order to eventually use it for a cut after the common mode subtraction to assign the channels having particle signals. The rms is approximated as $m_k - m_1 / (k/4)$

As shown in Fig. 4, the words in the memory used for common mode calculation are 16-bits long. The address (4-bits) of each strip is saved together with the digitized channel signal amplitude (12-bit) in the same memory location after pedestal subtraction. After the sorting, the data of strips in the 1st bank are rearranged to recover their original position in the list in order to be used in the following stages of the off-detector electronics. This has been done to avoid using extra memory for temporary storage of the original data. The rearrangement process occurs concurrently with the calculation of the common mode after sorting and, therefore, no extra time is needed. At the end of the common mode calculation process, the common mode is subtracted.

IV. USE OF THE COMMON MODE REJECTION METHOD FOR THE CMS PRESHOWER READOUT

As mentioned in the introduction, the data received by the off-detector electronics are multiplexed. Each data frame includes 3 subsequent, with 25 ns time difference, digitized samples (SLOTS) of the signals from the strips of up to 4 sensors in order to reconstruct the pulse produced by the preamplifier-shaper. The originally-induced charge is calculated from

the reconstructed pulse shape using the deconvolution technique [15]–[17]. The off-detector electronics functionality for the Preshower is shown in Fig. 5. The data frame after integrity checks (de-serialization, CRC, packet synchronization between on and off detector electronics) is being unpacked in order to construct the lists of the 16 adjacent channels values (SLICES). The pedestals are subtracted online during the unpacking.

After the common mode calculation and subtraction, a bunch-crossing-identification procedure is performed in order to reject samples including residual signal from previous events. Finally, the particle-induced charge is reconstructed and a threshold depending on the pedestal-common mode variation is applied. Only data from strips with particle-induced charge are transmitted to the CMS DAQ system together with their addresses and a frame header. Due to the CMS Preshower low occupancy, less than 5% of the original data would be sent to the CMS DAQ event builder.

As shown in the upper right part of Fig. 5, the execution time of the common mode calculation procedure for 4 PACE chips, in a Virtex-II Pro FPGA XC2VP7 running with a clock of 160 MHz, is 6.1 μ s in the longest case. This is below the frame readout time of 7.5 μ s (readout clock 40 MHz) and the occupancy is $6 \cdot 7.5\% = 45\%$ of the FPGA logic resources.

V. DISCUSSION

The method of calculating the common mode described in this paper as stated in the introduction considers the common mode as the time dependent mean base line shift of the channel

pedestals in a group of adjacent channels. With this consideration the performance of the method is good even in cases where the pedestal and common mode rms are high.

Unfortunately, there are cases where the common mode, which is mainly external electromagnetic noise, is not common for all the channels in the group, but increases or decreases linearly or almost linearly with the channel number, having one or two slopes. Therefore, the method must be extended to calculate the actual common mode contribution to the signal of each channel in these events using the slope of the mentioned dependency.

This can be done without any major change in the design described here since the strip addresses are available as mentioned earlier. Successive channel addresses in the sorted list used to calculate the common mode indicate that the common mode depends on the channel number. Furthermore, in this case the slope of its dependency can be calculated, and, therefore, the common mode of each channel can be calculated and subtracted.

REFERENCES

- [1] P. Wertelaers, ECAL Preshower Engineering Design Review [Online]. Available: <http://edms.cern.ch/document/115565/1> CMS ECAL EDR-4
- [2] P. Aspell, D. Barney, W. Bialas, J. Crooks, M. Dupanloup, A. Go, K. Kloukinas, D. Moraes, Q. Morrissey, and S. Reynaud, "PACE3: A large dynamic range analogue memory ASIC assembly designed for the readout of silicon sensors in the LHC CMS Preshower," in *Proc. 10th Workshop Electronics for LHC Experiments*, Boston, MA, 2004, pp. 137–141.
- [3] G. Minderico, C. Fachada, I.-L. Chan, K.-M. Chan, H.-M. Cheong, A. Lopes, P. Cardoso, J. Vital, K. Kloukinas, and A. Marchioro, "A CMOS low power, quad channel, 12 bit, 40MS/s pipelined ADC for applications in particle physics calorimetry," in *Proc. 9th Workshop on Electronics for LHC Experiments*, Amsterdam, The Netherlands, 2003, pp. 88–91.
- [4] K. Kloukinas, P. Aspell, D. Barney, S. Bonacini, and S. Reynaud, "Kchip: A Radiation Tolerant Digital Data Concentrator chip for the CMS Preshower Detector," in *Proc. 9th Workshop on Electronics for LHC experiments*, Amsterdam, The Netherlands, 2003, pp. 66–70.
- [5] I. R. Tomalin, "On calibration, zero suppression algorithms and data format for the silicon tracker FEDs," *CMS-IN 2001/025* [Online]. Available: http://cmsdoc.cern.ch/documents/01/in01_025.pdf
- [6] M. De Fez-Laso, "Beam test performance of the APV5 chip," *CMS-IN 1996/051*. [Online]. Available: http://cmsdoc.cern.ch/documents/96/t96_051.pdf
- [7] S. J. Inkinen and Y. Neuvo, "Base line normalization of high energy physics detector signals using sparse median operations," in *Proc. IEEE Winter Workshop on Nonlinear Digital Signal Processing Tampere, Finland*, 1993, pp. 4.1_3.1–4.1_3.6.
- [8] Y. Chang, A. Chen, S. Hou, and W. Lin, "Study of the charge cluster characteristics and spatial resolution of a silicon microstrip detector," *NIMA*, vol. 363, pp. 538–544, 1995.
- [9] E. Banas, "Halny: A digital signal processor based module for the readout of silicon strip detectors," *NIMA*, vol. 469, pp. 364–372, 2001.
- [10] A. Bay, G. Haefeli, and P. Koppenburg, "LHCb VeLo off detector electronics preprocessor and interface to the level 1 trigger," LHCb VeLo 2001-043 [Online]. Available: <http://doc.cern.ch/archive/electronic/cern/others/LHB/public/lhcb-2001-043.pdf>
- [11] N. Tuning, L1-type Clustering in the VELO on Test-beam Data and Simulation, LHCb 2003-073 TRIG [Online]. Available: <http://doc.cern.ch/archive/electronic/cern/others/LHB/public/lhcb-2003-073.pdf>
- [12] H. Ikeda *et al.*, "Study of common-mode noise of the SMA2SH-64A preamplifier array," *NIMA*, vol. 389, pp. 454–462, 1997.
- [13] N. Manthos and O. Mitropoulos, "A first attempt at an algorithm for on-line pedestal and common mode subtraction of data from the CMS preshower delta stream chip, and its implementation in an FPGA," *CMS IN 2002/047* [Online]. Available: http://cmsdoc.cern.ch/documents/02/in02_047.pdf
- [14] C. Peichel, "Integers out of sorts? program an FPGA to put them in order," *EDN Mag.*, Aug. 15, 1997.
- [15] S. Gadomski, "The deconvolution method of fast pulse shaping at hadron colliders," *NIMA*, vol. 320, pp. 217–227, 1992.
- [16] N. Bingefors, "A novel technique for fast pulse shaping using a slow amplifier at LHC," *NIMA*, vol. 326, pp. 112–119, 1993.
- [17] P. Bloch and E. Tournefier, "BC assignment and charge reconstruction with voltage sampling Preshower electronics," *Preshower Internal Document*, 1999 [Online]. Available: <http://cmsdoc.cern.ch/cms/ECAL/preshower/Documents/preshower/vsam.pdf>