

A Processor for Calorimetry at the Large Hadron Collider in the FERMI Project

Luigi Dadda*, Sami Inkinen**, Vincenzo Piuri*

* Dept. of Electronics and Information, Politecnico di Milano, I-20133 Milano, Italy

**CERN, CH-1211 Geneva 23, Switzerland

Abstract

A dedicated digital signal processor has been designed as part of a fully digital front-end for calorimetric detectors developed for experiments in high-energy particle physics to be carried out at CERN with the Large Hadron Collider. Its function is to evaluate the collision energy by analyzing the 16-bits samples (at 67 Msamples per second) contained in a time window through convolutions with 10-bits coefficients. In order to improve the time and amplitude accuracy of the evaluated pulse maximum, we adopted a scheme composed of three inner-product units followed by a sorting circuit which identifies the maximum, the minimum, and the median value among the three inner-product outputs. Due to the massively radioactive environment in which the calorimeter operates, the fault tolerance issues have been particularly taken into account to provide both concurrent error detection through residue arithmetic and graceful degradation.

1: Introduction

The electronics for signal detection and analysis in experiments at future proton supercolliders is one of the most complex systems designed for high-energy physics. In particular, detection and identification of physical events require the use of advanced calorimeters capable of dealing with an impressive amount of samples in a very hard real time. To solve this problem, a new-generation of digital front-end and readout electronics for calorimeters is under design for the future LHC (Large Hadron Collider) accelerator at CERN (Geneva, Switzerland), within the FERMI project. A detailed description of FERMI project can be found in [11].

Events' detectors produce analog signals which are sampled and digitalized at the rate of 67 Msamples/sec: due to the high number of channels (about 100,000), the digital outputs of the detecting section in the calorimeter generate a huge amount of numeric data (in the order of hundreds of TBytes/sec). A complete analysis of all these samples is therefore not feasible in real time by using traditional or general-purpose architectures to provide the necessary information for system control and for storing only the relevant data concerning physical events. Samples must be progressively reduced to extract only the physically-relevant data, through subsequent compression steps (called trigger levels). In the first-level trigger, low-resolution samples are selected with a reduction ranging from $10^3:1$ to $10^4:1$. Then, a second-level trigger operates on high-resolution data and achieves a further reduction of about $10^2:1$ (thus, only 1

This work has been done in the frame of the FERMI Collaboration Group, including CERN (Geneva, CH), University of Stockholm (S), University of Pavia (I), University of Linköping (S), Universities of Paris VI-VII (F), K. Tekniska Högskola (S), Politecnico di Milano (I), SiCon AB (S), ABB-Häfa AB (S), CAEN (I). The scope of this collaboration is the design and realization of a digital readout system for the calorimeters to be used in the Large Hadron Collider at CERN.

sample out of 10^5 or 10^6 is delivered by the second trigger's output). A third triggering level is concerned with the detailed analysis of the relevant events.

The large amount of data and the real-time constraints on operation led to design a dedicated architecture to perform any data analysis related to the first and the second triggering levels. In particular, this paper presents the processor for continuous measurement of the pulse amplitude at high accuracy in the second-triggering level, operating on all samples generated by the first-level trigger in a time window and related to the pulse due to particles' collision. Since the sample timing jitter and various noise sources tend to limit the accuracy of the results generated by such a processor, specific design approaches must be adopted by exploiting the physical characteristics of the measured events.

In Section 2, we introduce the theoretical basis of the measurement operations that must be performed to derive the energy values for the particles' collisions. Specific solutions are adopted to guarantee the result accuracy. In the literature, the best results were achieved by using Hybrid FIR Order Statistic filters [5] for similar data analyses on continuous sequences of samples; however, this solution cannot be directly applied to our application since we deal with disjoint set of samples that are presented sequentially to the processor for collision energy evaluation (i.e., we are concerned with sequences of inner products instead of traditional digital filtering). Architectural solutions proposed for median hybrid filters [6, 7] cannot be adapted since we need a high flexibility in output generation and in system configuration. In fact, the uncompleted knowledge of the operating environment characterization does not allow for a fixed definition both of the structure and of characteristic parameters of the processor; these characteristics must be chosen dynamically to provide graceful degradation.

In Section 3, the architecture of the dedicated processor is presented and discussed. It is organized in an array structure at two abstraction levels: the lowest one is concerned with inner-product generation, while the highest one is directed to sort the separately-convolved data and to extract the actual energy of the collision events. Section 4 analyzes the fault tolerance issues to provide features which are mandatory in the hard radiation calorimeter environment. In particular, concurrent error detection is considered to certify continuously the correctness of results; graceful degradation is also taken into account to allow system survival (even if with reduced capabilities) after fault occurrence.

2: Hybrid F.I.R. order statistic filters

In the first trigger stages the position and the amplitude of the pulses caused by particle hits are computed using 1-D time-domain data. The analog signal obtained from a single detector channel is converted into digital domain using a fast A/D converter with an analog amplifier. Sampling is synchronized with the collider bunch crossing. However, due to the non-zero length of the particle bunches and the geometry of a detector system the pulse timing relative to the sampling position is slightly different for each pulse. Here this deviation is assumed to be Gaussian-distributed with parameters $G(0, \sigma_s)$, where σ_s is the *sample timing jitter*.

The discrete-time signal also contains noise, divided into electronics and pile-up noise (due to small pulses). Since the noise is injected into the system prior to the analog amplifier, it is correlated in the time domain. The electronic noise, having Gaussian-like distribution, can be effectively attenuated using a linear FIR filter; discussion of optimal linear filtering can be found in [10]. The pile-up noise component has a distribution close to the Laplacian probability density function. It has been shown that of all the N-point operators the *maximum likelihood estimator* (MLE) for the Laplacian distribution is the *standard median* (SM) filter

$$y(i) = \text{MED} [x(i-n), x(i-n+1), \dots, x(i), \dots, x(i+n)]$$

where $N = 2n + 1$; it is obtained by sorting the samples $x(t)$ in an ascending order and selecting the $(n+1)$ -th largest value as output. For this reason, we assume that the optimal filter for the signal corrupted by the two noise components (electronics and pile-up noise) is a non-linear operator.

Filters composed of linear and non-linear operators (*hybrid filters*) are difficult to design using an analytical approach. A training algorithm for FIR *order statistic* (FIR-OS) filters has been presented in [12]: this method is based on the use of neural techniques to configure an hybrid order statistic filter, by adapting the algorithm presented in [8]. Application of this approach has been adopted for the prototype implementation of the digital calorimeter in the FERMI project.

The FIR subfilters form a bank of filters whose outputs are the inputs of the order statistic operator; such operator selects one subfilters' outputs for each input sequence centered on the sample $x(i)$ [14, 5]. The filter is given by

$$y(i) = \text{OS}[\phi_1(i), \phi_2(i), \dots, \phi_m(i)]_r$$

where r is any rank between 1 and m , and $\phi_k(\cdot)$'s are the FIR subfilters given by

$$\phi_k(i) = w^k_1 x(i-n_a) + w^k_2 x(i-n_a+1) + \dots + w^k_N x(i+n_b)$$

where the tap coefficient vector is $\mathbf{W}_k = \{w^k_1, w^k_2, \dots, w^k_N\}$, and $N = n_a + n_b + 1$. The filter structure can be optimized using a training procedure to find the rank r and the FIR filter tap coefficient vectors $\mathbf{W}_k$: the optimum solution corresponds to the minimum total error between the filter output and the expected output. The cost function of the optimization problem is defined by using the mean squared error (MSE) norm:

$$E = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(y(i) - \hat{y}(i) \right)^2$$

where $\hat{y}(i)$'s are the expected filter outputs, and N_q is the number of samples in the training set Q . By minimizing the cost function E , the optimal filter for set Q is obtained in the MSE norm sense. The training algorithm, based on the use of *soft order statistics* and conjugate gradient optimization, is explained in detail in [12].

The performance of the trained Simple-FIR and FIR-OS filters was evaluated using a set of simulated pulses with a bipolar shaping function as transfer function of the A/D converters, and 15 ns sampling interval. The signal is corrupted by electronics and pile-up noise: we assumed $\sigma_s = 2$ ns sample timing jitter. Overlapping pulses were added to the sequences, using a logarithmic energy distribution function

$$e = \begin{cases} e_s \ln[P_m / P] & \text{if } P < P_m \\ 0 & \text{otherwise} \end{cases}$$

where $e_s = 1$ GeV is an energy scaling term, and $P_m = 0.03$ is the maximum probability of having any significant energy deposit in a single event. The probability value P is uniformly distributed between 0 and 1. The normalized amplitude error values $\text{MSE}[y]/y$ are shown in Fig. 1. Clearly the FIR-OS filters outperform the FIR filter, providing more degrees of freedom to suppress all noise components (good amplitude precision) and the effect of overlapping pulses (high time resolution). On the other hand, the use of a high number m of subfilters does not appear particularly attractive since the performance improvement in the filtering capability is

not relevant. For the signal processing purposes in the FERMI project, simulations have shown that the best compromise between performance and hardware complexity is m equal to 3; even if $m = 2$ appears sufficient to provide a good filtering performance, both the possible presence of unexpected distortions or misaligned pulses and definition of graceful degradation policies for fault tolerance suggest to adopt $m = 3$. When no fault is present, the best filtering capabilities are achieved. Fault occurrence in a FIR filter can be overcome by excluding the faulty unit and by adapting both the filtering coefficients and the rank of the hybrid filter: filtering performances are lightly reduced. After a second fault in a FIR filter, the system is still able to survive by adapting the filter configuration; the filtering characteristics are reduced with respect to the nominal behavior, but they may be still accepted instead of declaring the whole system faulty. Graceful degradation implies therefore the capability of dynamic definition both of the filtering coefficients and of the rank.

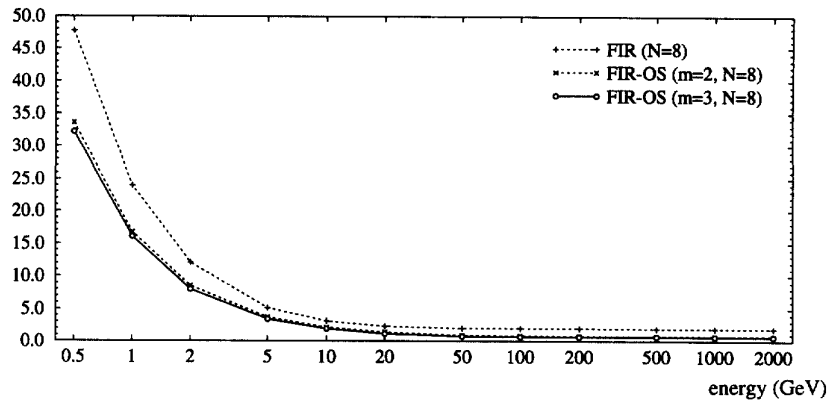


Fig. 1 - Normalized amplitude error in pulse amplitude extraction

3: The processor architecture

The processor architecture for the implementation of a FIR-OS filter (with $m=3$) is shown in Fig. 2. The core of this dedicated system is composed of three *inner multipliers* [2, 9] which compute three inner products to provide three different views of the same set of sampled data through three different sets of coefficients (see Fig. 3). Each of these units is an inner multiplier since it produces the weighted summation of the input samples in a time window with respect to a given set of weights. These weights are defined according to the requirements of the physical analysis that must be carried out on collision data and to the theoretic frame discussed in Section 2. To guarantee enough flexibility to the calorimeter structure and to the second triggering level, the inner-product weights can be loaded into memories on board of the processor before starting its operation.

The outputs of the inner multipliers (A, B, and C) are connected to a sorting circuit which selects the one that must be delivered as processor output. To achieve this goal, the sorting circuit identifies first the maximum (M), the minimum (μ), and the median (m) values among the three inner-multipliers' outputs; then, it propagates the value having the desired rank r of the FIR-OS filter to the final processor output.

The sorting circuit is composed of three comparators, three r -rank inner-product selectors, one control selector, and one output selector. The comparators produce the sign of the differences (A-B), (B-C), and (C-A), respectively. In the given time window, these signals are used by the inner-product selector associated to the rank r to identify the inner product having such a rank among the actual values generated by the inner multipliers. Each inner-product selector is specialized to identify the maximum ($\mathbf{M}$, for $r=3$), the minimum (μ , for $r=1$), or the median ($\mathbf{m}$, for $r=2$) value, respectively; the output of each inner-product selector is the identifier of the inner multiplier which produced the value at the rank associated with the inner-product selector itself. The control selector allows to choose the rank of the FIR-OS filter by means of external control signals; the output of this selector is the identifier of the inner multiplier which produced the value at the externally-specified rank.

Finally, the output selector allows to deliver the inner product having the desired rank to the processor output; its operation is driven by the output of the control selector.

Each inner multiplier is implemented by cascading one multiplier and one adder-accumulator. The multiplier generates the product between one sample and the corresponding weight, while the adder produces the partial summation by adding the actual product available at multiplier's output and the previous value of the partial summation stored in the accumulator register. Then, the accumulator stores the new partial summation. This sequence of operations is performed until all samples have been treated. To maximize the throughput, a pipelined structure has been adopted [13].

The resulting architecture of the pipelined multiplier is given in Fig. 4. Samples (in the application envisioned, they are represented as unsigned 16-bits numbers) are input in parallel as multiplicand, at the system speed rate of 67 Msamples/sec. The inner-product coefficients w^k_j (two's-complement 10-bits numbers) are input as multipliers in the 4-bits-parallel/skewed format. Coefficients' bits are grouped four at a time (namely, $4 + 4 + 2$ in the specific application); bits within the individual group are presented in parallel, while groups are input in the skewed form. This mixed operands' format is due to the internal structure of the pipelined multiplier, designed according to the well-know Braun's scheme and collapsing four multiplier-array rows in the single pipeline step.

The ten least-significant bits of the product are output from the right side of the array in the 4-bits-parallel/skewed form (as for the multiplier factors w^k_j). The 16 most-significant bits are implicitly represented by Sum and Carry bits generated from the row of Full Adders at the bottom of the multiplier array; they can be obtained in an explicit form by adding the Sum and Carry bits and by propagating the carries properly. This operation could be completely performed in parallel within a single clock period by using a fast adder (e.g., a carry-look-ahead adder). However, since the final inner product will be stored in the Accumulator and in the modulo-3 Accumulator and will be then processed in the Comparators, it seems convenient to adopt the 4-bits-parallel/skewed form also for the most significant part of the result. This choice is in fact useful to simplify the design both of the adder-accumulator and of comparators. On the

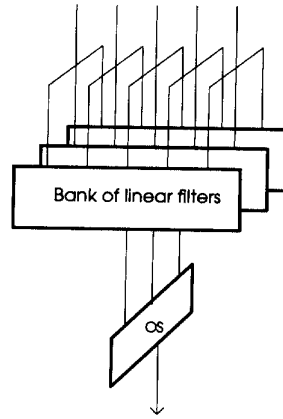


Fig. 2 - The general scheme of the processor

other hand, the higher latency is not so relevant in the application envisioned while the processor throughput must be equal to the detectors' output rate of 67 Msamples/sec.

Each stage of the pipeline operates on four product array rows since, in the implementation technology adopted, the cycle time imposed by throughput is long enough to accommodate addition of four operands in carry-save form.

Since the number $n_w = 10$ of the multiplier's bits is not divisible by 4, the third pipeline stage contains only two rows of the multiplier array. This implies that the second half of each cycle time should be wasted in the third pipeline stage. To avoid this loss of computational capability, two stages of the pipelined adder array are collapsed in the third inner-product pipeline stage together with the two multiplier's rows.

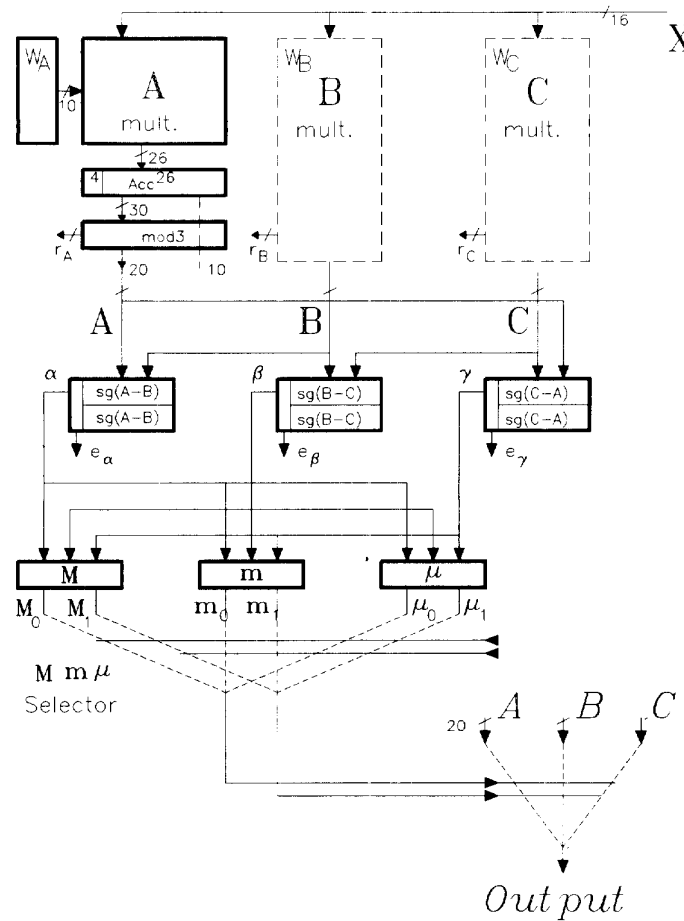


Fig. 3 - The scheme of the processor with the three inner multipliers

The bottom part of the pipelined multiplier in Fig. 4 (generating the 16 most-significant bits of the product) has been designed to minimize the number of adders and latches: in particular, it requires several adders less than the traditional scheme usually suggested. This new scheme has been used in previous designs [9] and is illustrated by the authors in [13].

The new value of the partial inner product, computed by adding the current sample-weight product to the previous value of the partial inner product, is then stored into the accumulator shown in Fig. 5. To simplify the following circuits in the FERMI system, while preserving the required accuracy, the final summation (i.e., the inner product of the considered time window) is truncated: only the 20 most-significant bits are delivered as processor's outputs.

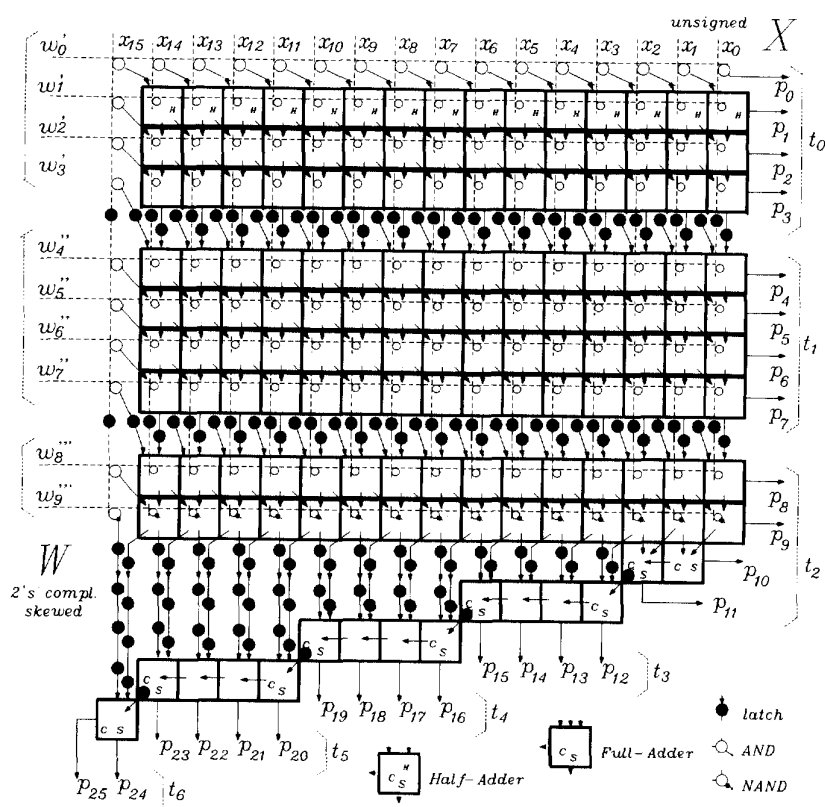


Fig. 4 - The scheme of the pipelined 10x16-bits multiplier

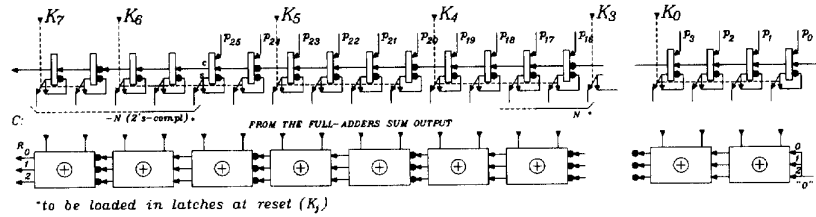


Fig. 5 - The scheme of the accumulator, including the modulo-3 accumulator

The sorting circuit orders the values stored in the three accumulators at the end of the inner-product computation. To achieve this goal, first it compares the three pairs of inner products (A,B), (B,C), and (C,A). The result of each comparison is the sign of the difference between corresponding inner products; these signs are obtained by the circuits labeled $sg(A-B)$, $sg(B-C)$, $sg(C-A)$, whose internal structure is shown in Fig. 6. To minimize the circuit complexity, the sign of each difference is not obtained by performing the complete subtraction of the nominal operands but by using a dedicated circuit for bit comparison of the two operands; this approach is feasible since A, B and C represent energies and, thus, are necessarily positive numbers. The outputs of the above comparators (α , β , and γ , respectively) are used to determine the maximum, the minimum and the median values.

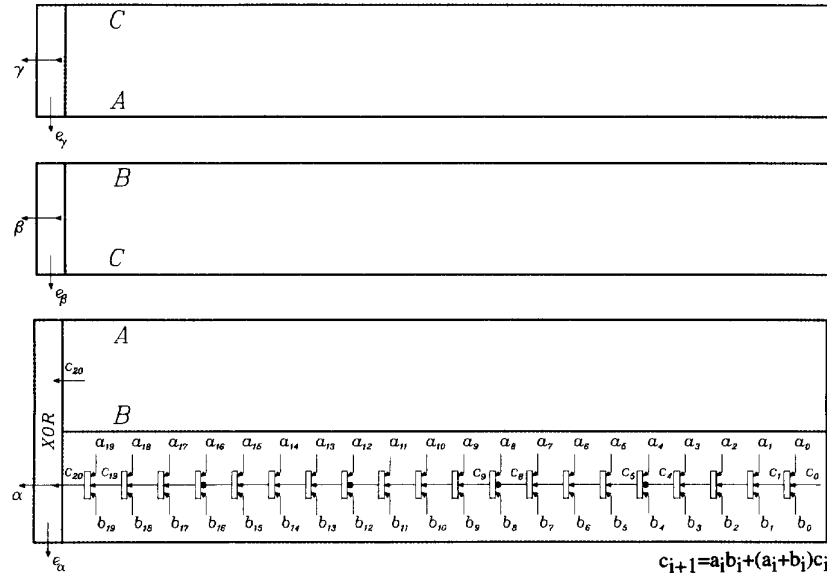


Fig. 6 - The scheme of the inner-products comparators

As already said, we note that input samples x_i are unsigned numbers since they are generated by the collision detectors, while coefficients w_j^k are two's-complement integer numbers. The inner product stored in the Accumulator at the end of the whole inner-product computation is always positive since it represents the collision energy: however, some partial inner products (i.e., the values stored in the Accumulator during each intermediate step) may be negative. The method adopted for the two's-complement multiplier factors needs that the last partial-product row is generated by using NAND gates (instead of the AND gates required for the other rows) and that the constant $C = -2^{n \times n_w - 1} + 2^{n \times n_l}$ is added to each product. This can be obtained by pre-loading the constant NC (N being the number of terms added in the inner product) into the Accumulator at reset time. By applying this method, the output of the multiplication array is always a positive number.

4: The fault tolerance issues

The core of the calorimeter will be installed near the collision chamber in which two proton beams of particles collide; since circuits described in this paper are placed on the back of the detectors, they will operate permanently in a strongly radioactive environment. Massively fault tolerant approaches need therefore to be considered in order to provide a high degree of reliability for the measurements, a high availability of the whole calorimeter, and possible survival even in the presence of a limited number of faults. In fact, even if the large quantity of data to be processed by the FERMI system contains implicitly a high degree of redundancy, both the *completeness* (no loss of data) and the *credibility* (no corruption, masking or aliasing) of data need absolutely be granted.

The project was started in the absence of definite knowledge about the behavior of microelectronic digital devices in the hard radioactive environment. This has prevented the possibility of formulating realistic, complete fault and error models. Preliminary testing results have shown that both transient and permanent faults (in general multiple) are to be expected for a set of VLSI technologies, including CMOS and Silicon on Insulator (SOI). Therefore, we refer to a functional fault model [4] in order to eliminate a definite dependency on the physical nature of faults.

Several policies can be envisioned to grant fault tolerance: it has been decided to design the FERMI system giving priority to *self diagnosis* capabilities for data correctness. Due to the large number of FERMI chips and to the amount of data to be processed, self diagnosis is performed in a concurrent and distributed way. *Self correction* has not been included due to the overhead in time and circuit complexity. Finally, *graceful degradation* has been taken into account to provide system survival even if with reduced functional performances. Remote control is necessary for managing the reconfiguration process since it may be too complex to be performed locally.

To provide *concurrent error detection* in all arithmetic units of the inner-product subsystems, residue codes has been adopted [1, 11]. Each datum x is represented by the codeword $\mathcal{C} = [x \parallel \mathcal{R}_b(x)]$; $\mathcal{C}$ is obtained by cascading the information bits x and the redundancy bits given by the residue $\mathcal{R}_b(x)$ of x with respect to the given base b . Since the residue code is a *separate* code, the redundancy bits of the result of any arithmetic operation can be directly generated from the redundancy bits of the operands.

Error checking is performed by comparing the residue of the nominal result with the expected residue generated independently from the residues of the operands: an error is detected (in the processor or in the checking circuits) whenever the actual and the expected residues are different. This approach is useful to detect all single/multiple errors whose residue (in the given

base b) are not null. In the envisioned application, a reasonable balancing between the hardware overhead and the detection capabilities is achieved by using the base 3, when a careful design has been adopted for multipliers and adders to guarantee that single faults appear as single errors [3]; the base 3 is in fact capable to detect any single error in summation and in multiplication [1].

Operations defined in Section 3 are performed also on the residue modulo-3 of the nominal operands, in parallel with the nominal computation (see Fig. 7). Then, the modulo-3 residue of the value stored in the Accumulator is compared to the corresponding result generated by the modulo-3 operations to detect possible errors. The circuit computing the modulo-3 residue of the Accumulator is shown in Fig. 5.

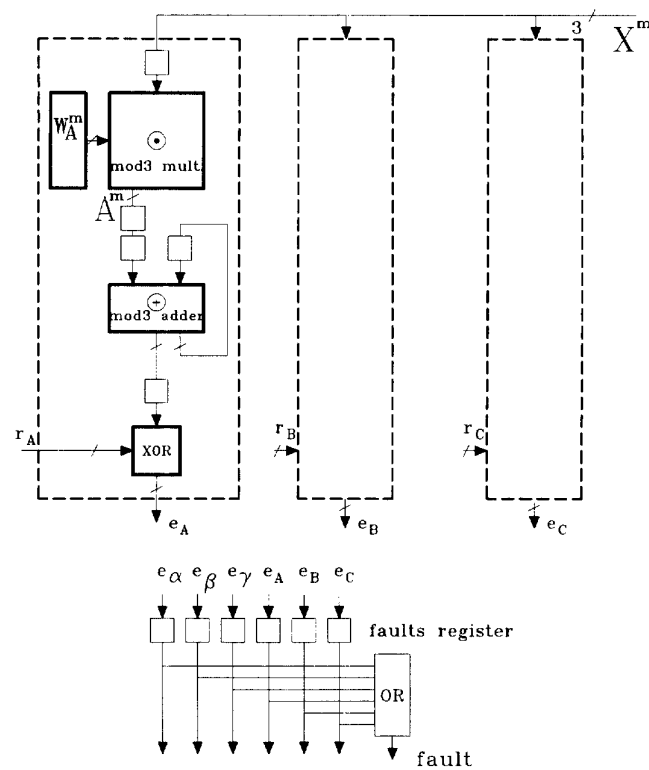


Fig. 7 - The scheme of the modulo-3 processor, with error signals

The inputs to the residue accumulator in Fig. 5 are the outputs of the product accumulator, so that the R output is the modulo-3 residue of each partial sum; therefore, each product is checked indirectly. The residue of the truncated output does not coincide with R : it can be obtained by subtracting (or by adding the two's-complement of) the residue of the discarded part (generated by the corresponding section of the residue-accumulator) from R .

Since faults in the Accumulators appear usually as single errors, residue codes are useful to protect also the memory elements.

Note that iterative use of the same faulty devices in the inner multipliers could induce either multiple errors in the inner products or error masking; both these cases may not be detected by the checking circuits with a critical reduction in the system credibility. However, since checking in our processor is performed on the result generated by *each* iteration of the inner product algorithm, this case does not occur: in fact, the concurrent checking procedure verifies each single use of any device in the inner multiplier. Error masking may occur only for contemporaneous multiple faults inducing multiple errors with null residue; in the case of sequential multiple faults (i.e., sequences of single faults occurring one at a time) inducing sequential multiple errors, detection of the first error allows for invalidation of all subsequent results generated by the faulty unit.

Consider now the sorting circuit: residue codes are not suitable since sorting is not an arithmetic operation. However, error detection must be guaranteed. In fact, in some cases, the use of erroneous comparators' outputs may lead to an erroneous selection of the inner product that is delivered to the external outputs; however, sometimes, comparators' errors can be detected by carefully analyzing the consistency among the three comparators' outputs. In other cases, comparators' faults modify the correct ordering among inner products, but they may be not relevant with respect to the values delivered to the processor output (i.e., the relative ordering between the other inner products is changed but the delivered value coincides with the expected one); also in some of these cases comparators' errors can be detected by a careful consistency analysis. A cost effective approach is based on modular redundancy (see Fig. 6); each comparator is duplicated and outputs of the modules in the pair are compared: an error is detected whenever these outputs are different.

Three basic alternatives are available for selectors: no protection, modular redundancy (i.e., duplication with comparison), or two-rail logic design. In the first case, selectors constitute the hard-core of the whole system; this approach may be justified by the limited circuit complexity and by the possible use of robust design techniques. In the case of modular redundancy, two copies of the selecting part (circuits at the bottom of Fig. 3) is duplicated and the signals generated by the two output selectors are compared. In the case of two-rail logic, the complete design of all selectors is reviewed to introduce some intrinsic redundancy in the signal representation; each signal is generated with its complemented value by using separate devices and interconnecting paths: the nominal signal delivered by the output selector and its complemented value must therefore be complementary, otherwise an error can be detected.

To guarantee *system survival* even in the presence of faults, no reconfiguration policy has been adopted by using spares units and redundant interconnections (possibly, switched busses) due to the massive circuit complexity that it should have been required. However, some reconfiguration features with *graceful degradation* of the nominal behavior have been included to increase the overall system availability. In particular, the presence of three inner multipliers - instead of two as required minimally by the FIR-OS filtering approach - allows for continuing the nominal filtering operation even if one inner multiplier is declared faulty. In this case, the remote host, controlling the dedicated processor, suspends the nominal operations: then, it loads a new set of coefficients for the two fault-free inner multipliers by assuming that the FIR-OS filter has $m=2$, while the control signals for rank selection are configured to take into account both the location of the faulty inner multiplier and the actual rank associated to the coefficient set. Finally, the remote host restarts the operation of the filtering processor at the degraded performances (even if degradation is not so relevant).

The same approach can be used also for one comparator or one r -rank inner-product selector; unfortunately, the corresponding inner multiplier is completely lost (as it was faulty). More

complex solutions could be adopted by means of modular redundancy and reconfiguration, but the additional circuit complexity should be too high and could induce a massive increase in the fault probability of these components.

The control selector and the output selector constitute the hard-core of the system with respect to the system survival, i.e., any fault in such components is fatal for the whole processor. Also in this case, the use of any redundancy technique would induce a high circuit complexity for redundancy and a high increase in the fault probability.

5: Concluding remarks

The architecture of a dedicated processor has been presented for energy evaluation in particles' collisions at LHC, by computing the maximum value of collision signal pulse represented by a sequence of samples in a time window. The theoretic foundation of the algorithm have been described to provide the physical motivations to the processor architecture. Computational constraints on cycle time, throughput, and circuit complexity have been derived from the characteristics of the application.

Fault tolerance features have been introduced to validate the system outputs by means of error detection capability and to guarantee a reasonable survival capability even in the presence of errors, possibly with degraded performances and characteristics.

This processor will be integrated in the next future by using a Silicon On Insulator technology due to the high resistance to fault occurrence in hard radiation environments. Prototype implementation will use a 0.8 μ production process. Experiments will be carried out to verify the efficiency of the fault tolerance policies with respect to the actual operating environment (which is partially unknown at the moment) and to define an optimum balancing of the hardware/time overhead for error detection and for system survival.

6: References

- [1] T.R.N. Rao, *Error Coding for Arithmetic Processors*, Academic Press, NY, 1974
- [2] E.E. Swartzlander, B.K. Gilbert, I.S. Reed, "Inner Product Computers", *IEEE Trans on Computers*, vol C-27, n.1, pp. 21-31, Jan.1978
- [3] R. Stefanelli, M. Annaratone, "A Multiplier with Multiple-error Correction Capability", *Proc. Int'l Conference on Computer Arithmetic ARITH6*, 1983
- [4] J.A. Abraham, W.K. Fuchs, "Fault and Error Models for VLSI", *Proc. IEEE*, May 1986
- [5] P. Heinonen, Y. Neuvo, "FIR-Median Hybrid Filters", *IEEE Trans. on Acoustic, Speech, and Signal Processing*, vol. 35, no. 6, pp. 852-858, June 1987
- [6] O. Vainio, P. Heinonen, Y. Neuvo, "Integrated FIR Median Hybrid Filter", *INTEGRATION - The VLSI Journal*, no. 5, pp. 171-177, 1987
- [7] O. Vainio, H. Tenhunen, T. Korpiharju, J. Tomberg, Y. Neuvo, "A Video Frequency Bit-parallel CMOS FIR Median Filter", *Proc. 1988 Symp. on VLSI Circuits*, Tokyo, Japan, August 1987
- [8] F. Palmieri, "A Backpropagation Algorithm for Multilayer Hybrid Order Statistic Filters", *Proc. 1989 IEEE Int. Conference on Acoustics, Speech and Signal Processing*, Glasgow, UK, May 1989
- [9] L. Dadda, "A Fast Pipelined Inner-product Unit", Dip. di Elettronica ed Informazione, Politecnico di Milano, Milano, Italy, Report No. 92-025, April 1992
- [10] W. E. Cleland, E. G. Stern, "Signal Processing for Liquid Ionization Calorimeters", *Proc. III International Conference on Calorimetry*, Corpus Christi, TX, Sept. 1992
- [11] The FERMI collaboration, "A Digital Front-End and Readout Microsystem for Calorimetry at LHC: the FERMI Project", *IEEE Trans. on Nuclear Science*, vol. 40, n. 4, pp. 516-531, Aug. 1993
- [12] S. J. Inkinen, Y. Neuvo, "Trainable Hybrid Filter Structures in Particle Detector Readout Systems", *Proc. VII European Signal Processing Conference*, Edinburgh, UK, Sept. 1994
- [13] L. Dadda, V. Piuri, "Pipelined Adders", Dip. di Elettronica ed Informazione, Politecnico di Milano, Milano, Italy, Report No. 94-038, March 1994 (submitted to the IEEE Trans. on Computers)
- [14] The FERMI collaboration, "The Digital Filter Section of the FERMI Readout Chip", FERMI Technical Report, April 1994 (submitted to the IEEE Trans. on Nuclear Science)