Optimized digital feature extraction in the FERMI microsystem

FERMI Collaboration

H. Alexanian e,j, G. Appelquist i, P. Bailly c, R. Benetta a, S. Berglund i, J. Bezamat e,j, F. Blouzon e, C. Bohm i, L. Breveglieri d, S. Brigati f, P.W. Cattaneo g, L. Dadda d, J. David e, M. Engström i, J.F. Genat e, M. Givoletti f, V.G. Goggi g, S. Gong e, G.M. Grieco l, M. Hansen a, H. Hentzell e, T. Holmberg m, I. Höglund k, S.J. Inkinen a, A. Kerek h, C. Landi f, O. LeDorze e, M. Lippi e, B. Lofstedt a, B. Lund-Jensen h, F. Maloberti f, S. Mutz e,d, P. Nayman e, V. Piuri d, G. Polesello e, M. Sami d, A. Savoy-Navarro e, P. Schwemling e, R. Stefanelli d, R. Sundblad m, C. Svensson b, G. Torelli f, J.P. Vanuxem a, N. Yamdagni i, J. Yuan b, A. Ödmark m

a CERN, Geneva, Switzerland
b Department of Physics and Measurement Technology, University of Linköping, Sweden
c Center for Industrial Microelectronics and Materials Technology, University of Linköping, Sweden
d Dipartimento di Elettronica, Politecnico di Milano, Italy, Sezione INFN, Pavia, Italy
e LPNHE, Université Pierre et Marie Curie, Paris VI-VII, Paris, France
f Dipartimento di Elettronica dell' Università e Sezione INFN, Pavia, Italy
g Dipartimento di Fisica Nucleare e Teorica dell' Università, Sezione INFN, Pavia, Italy
h Physics Department, Royal Institute of Technology, Stockholm, Sweden
i Fysikum, University of Stockholm, Sweden
j Also from ESIEE, Paris, France
k ABB-Habo AB, Järnsätra, Sweden
l C.A.E.N. S.p.A., Viareggio, Italy
m SiCom AB, Linköping, Sweden
Optimized digital feature extraction in the FERMI microsystem

FERMI Collaboration


1. Introduction

The general architecture and implementation of the FERMI readout microsystem has been presented in Refs. [1,2]. In this paper we concentrate on the digital signal processing operations performed by two separate digital filter units. Large data rate reduction factors in the trigger and acquisition chains require high-precision digital signal processing algorithms. Due to circuit area limitations the filter structures are a compromise between optimum feature extraction performance and hardware complexity. We describe the architectures of the digital filters used in the FERMI microsystem. We also present the performance of linear and nonlinear discrete-time operators and introduce a training algorithm to optimize the filters against the different artefacts present in the acquired signal sequences. The paper is organized as follows: we begin by discussing the optimal filtering problem in signal processing for particle detectors. We then introduce the filter algorithms and the procedure for obtaining the filter parameters. The hardware implementation, showing both VHDL modelling and subsystem optimization, is preceded by fault tolerance considerations. The last part contains the system performance evaluation and concluding remarks.
1.1. Digital signal processing and optimal filtering

Let the pulse waveform at the output of the analog preamplifier be \( g(t) \). The discrete-time signal sequence associated with a particle passing through the detector at time \( t = 0 \) is thus given by (assuming that the effect of the dynamic range compression is perfectly compensated in the renormalization lookup table)

\[
x(i) = e^{g[t_i - \tau]},
\]

where \( e \) is the energy deposit of the particle, \( t_i \) is the sampling interval, \( \tau \) is the time deviation from the bunch crossing and \( g(t) = 0 \) for all \( t < 0 \). Signal sampling in individual channels is synchronized with the bunch crossing so that the expectation value of \( \tau \) is a multiple of the sampling interval, i.e., \( E[\tau] = n_t \). However, due to the non-zero length of the particle bunches, the geometry of the detector system and the intrinsic signal jitter, the deviation is slightly different for each pulse. Here \( \tau \) is assumed to be Gaussian-distributed with parameters \( G(n_t, \sigma) \), where \( \sigma \) is called the sample timing jitter. In addition to the sample timing jitter and the inherent quantization noise, the feature extraction performance is affected by electronics and pile-up noise components and the time-domain superposition of several pulses.

The electronics noise is of thermal origin and it is modelled using additive Gaussian white noise. The pile-up noise, originating from minimum bias events, has a distribution somewhat similar to the Laplacian (double exponential) distribution. Both noise components pass through the analog shaper and thus become correlated in the time domain. The appearance of the sample timing jitter in individual samples is more complicated. The error in timing is converted to an error in the sample value, with the amplitude deviation proportional to the gradient of \( g[\cdot] \) at the sample position. As the sample timing jitter is a signal-dependent noise component, we can assume that it is the dominant error source for high-energy pulses.

Optimal filtering depends on the noise distribution [3]. For amplitude extraction of a pulse contaminated by Gaussian-distributed noise the optimal filter is a matched finite impulse response (FIR) filter, referred to as the optimal filter in experimental physics literature [4-6]. Similarly, the optimal filter for amplitude extraction in Laplacian noise environment is the matched median filter [7]. We thus assume that the optimal filter for the signal sequence corrupted by the different noise components and the superposition of high-energy pulses is a nonlinear operator. The truly optimal filter can be found using the maximum likelihood estimation theory [8]. However, in the presence of a large number of independent artefacts the analytical solution for the optimal filtering problem is considered unfeasible. For this reason we adopt the procedure of predefining the filter structure and using an optimization algorithm to compute its parameters. Several structures combining linear and nonlinear operators into a hybrid filter have been presented in the signal processing literature: LJ-filters [9,10], hybrid order statistic filters [11], FIR-order statistic hybrid (FIR-OS) filters [12,13], FIR-weighted order statistic hybrid (FIR-WOS) filters [14,15] and neural filters [16]. The structural complexity limits the practical usage of some of these filters in detector readout systems. However, by comparing the feature extraction performance of the different operators we can find a compromise to fulfil the filtering accuracy, minimum circuit area and fault tolerance requirements. The FIR-OS structure has been selected for evaluation in the first FERMI prototype.

2. Filter structures and algorithms

The filter functions of the FERMI microsystem are implemented as two separate digital filter blocks. This division is related to the typical trigger structure of high energy physics experiments. Filter F1 operates at the sampling rate and extracts information for the first-level trigger process. The high-precision output of filter F2, achieved using more complicated signal processing algorithms, is directed to higher-level trigger processes and to the data readout system. Filter F1 performs the following operations: identification of the event time, measurement of the pulse amplitude, and identification and flagging of overlapping pulses. Filter F2 performs the following operations: high-precision measurement of amplitude for individual pulses and reduced-precision measurement of amplitude for overlapping pulses.

Let the signal sequence acquired from a single detector channel \( c \) be given by

\[
x_c = \{x_c(i-1), x_c(i), x_c(i+1), \ldots\},
\]

where the samples \( x_c(i) \in \mathbb{Z}^+ \) by definition. The input of filter F1 is obtained by summing all the C channels in a FERMI, giving the sequence \( x_t(i) = \sum_{c=1}^{C} x_c(i) \).

2.1. Pulse finding stage: filter F1

Filter F1 is a parallel structure of two independent FIR filters (Fig. 1). Filter F1e is optimized for pulse amplitude measurement and filter F1t reshapes the pulses to obtain precise time information. The output \( y_t(i) \) is given by

\[
y_t(i) = \begin{cases} y_{tu}(i) & \text{if } y_{tu}(i) > e_T \text{ and } y_{tu}(i) = 1, \\ 0 & \text{otherwise}, \end{cases}
\]

where \( e_T \) is an energy threshold, and the output \( y_{tu}(i) \) of the amplitude extraction filter and the output \( y_{tu}(i) \) of the time extraction filter are given by

\[
y_{tu}(i) = h_{e,1}x_t(i-n_{1a}) + h_{e,2}x_t(i-n_{1a} + 1) + \ldots + h_{e,3l}x_t(i+n_{1b}), \]

\[
y_t(i) = \begin{cases} 1 & \text{if } z(i) > z(i-1) \text{ and } z(i) \geq z(i+1), \\ 0 & \text{otherwise}, \end{cases}
\]

where \( z(i) \) is the sample value, with the amplitude deviation proportional to the gradient of \( g[\cdot] \) at the sample position. As the sample timing jitter is a signal-dependent noise component, we can assume that it is the dominant error source for high-energy pulses.

Optimal filtering depends on the noise distribution [3]. For amplitude extraction of a pulse contaminated by Gaussian-distributed noise the optimal filter is a matched finite impulse response (FIR) filter, referred to as the optimal filter in experimental physics literature [4-6]. Similarly, the optimal filter for amplitude extraction in Laplacian noise environment is the matched median filter [7]. We thus assume that the optimal filter for the signal sequence corrupted by the different noise components and the superposition of high-energy pulses is a nonlinear operator. The truly optimal filter can be found using the maximum likelihood estimation theory [8]. However, in the presence of a large number of independent artefacts the analytical solution for the optimal filtering problem is considered unfeasible. For this reason we adopt the procedure of predefining the filter structure and using an optimization algorithm to compute its parameters. Several structures combining linear and nonlinear operators into a hybrid filter have been presented in the signal processing literature: LJ-filters [9,10], hybrid order statistic filters [11], FIR-order statistic hybrid (FIR-OS) filters [12,13], FIR-weighted order statistic hybrid (FIR-WOS) filters [14,15] and neural filters [16]. The structural complexity limits the practical usage of some of these filters in detector readout systems. However, by comparing the feature extraction performance of the different operators we can find a compromise to fulfil the filtering accuracy, minimum circuit area and fault tolerance requirements. The FIR-OS structure has been selected for evaluation in the first FERMI prototype.

2. Filter structures and algorithms

The filter functions of the FERMI microsystem are implemented as two separate digital filter blocks. This division is related to the typical trigger structure of high energy physics experiments. Filter F1 operates at the sampling rate and extracts information for the first-level trigger process. The high-precision output of filter F2, achieved using more complicated signal processing algorithms, is directed to higher-level trigger processes and to the data readout system.

Filter F1 performs the following operations: identification of the event time, measurement of the pulse amplitude, and identification and flagging of overlapping pulses. Filter F2 performs the following operations: high-precision measurement of amplitude for individual pulses and reduced-precision measurement of amplitude for overlapping pulses.

Let the signal sequence acquired from a single detector channel \( c \) be given by

\[
x_c = \{x_c(i-1), x_c(i), x_c(i+1), \ldots\},
\]

where the samples \( x_c(i) \in \mathbb{Z}^+ \) by definition. The input of filter F1 is obtained by summing all the C channels in a FERMI, giving the sequence \( x_t(i) = \sum_{c=1}^{C} x_c(i) \).

2.1. Pulse finding stage: filter F1

Filter F1 is a parallel structure of two independent FIR filters (Fig. 1). Filter F1e is optimized for pulse amplitude measurement and filter F1t reshapes the pulses to obtain precise time information. The output \( y_t(i) \) is given by

\[
y_t(i) = \begin{cases} y_{tu}(i) & \text{if } y_{tu}(i) > e_T \text{ and } y_{tu}(i) = 1, \\ 0 & \text{otherwise}, \end{cases}
\]

where \( e_T \) is an energy threshold, and the output \( y_{tu}(i) \) of the amplitude extraction filter and the output \( y_{tu}(i) \) of the time extraction filter are given by

\[
y_{tu}(i) = h_{e,1}x_t(i-n_{1a}) + h_{e,2}x_t(i-n_{1a} + 1) + \ldots + h_{e,3l}x_t(i+n_{1b}), \]

\[
y_t(i) = \begin{cases} 1 & \text{if } z(i) > z(i-1) \text{ and } z(i) \geq z(i+1), \\ 0 & \text{otherwise}, \end{cases}
\]

where \( z(i) \) is the sample value, with the amplitude deviation proportional to the gradient of \( g[\cdot] \) at the sample position. As the sample timing jitter is a signal-dependent noise component, we can assume that it is the dominant error source for high-energy pulses.

Optimal filtering depends on the noise distribution [3]. For amplitude extraction of a pulse contaminated by Gaussian-distributed noise the optimal filter is a matched finite impulse response (FIR) filter, referred to as the optimal filter in experimental physics literature [4-6]. Similarly, the optimal filter for amplitude extraction in Laplacian noise environment is the matched median filter [7]. We thus assume that the optimal filter for the signal sequence corrupted by the different noise components and the superposition of high-energy pulses is a nonlinear operator. The truly optimal filter can be found using the maximum likelihood estimation theory [8]. However, in the presence of a large number of independent artefacts the analytical solution for the optimal filtering problem is considered unfeasible. For this reason we adopt the procedure of predefining the filter structure and using an optimization algorithm to compute its parameters. Several structures combining linear and nonlinear operators into a hybrid filter have been presented in the signal processing literature: LJ-filters [9,10], hybrid order statistic filters [11], FIR-order statistic hybrid (FIR-OS) filters [12,13], FIR-weighted order statistic hybrid (FIR-WOS) filters [14,15] and neural filters [16]. The structural complexity limits the practical usage of some of these filters in detector readout systems. However, by comparing the feature extraction performance of the different operators we can find a compromise to fulfil the filtering accuracy, minimum circuit area and fault tolerance requirements. The FIR-OS structure has been selected for evaluation in the first FERMI prototype.
The FIR tap coefficients of the two filters with length \( N_1 = n_{la} + n_{lb} + 1 \) are \( h_{r,k} \) and \( h_{l,k} \), respectively. To enable coefficient switching for filter F2 an overlap flag \( f(i) \) is associated with the time frame information. For the current pulse it is given by

\[
f(i) = \begin{cases} 1 & \text{if } c < d_T, \\ 0 & \text{otherwise,} \end{cases}
\]

where \( d_T \) defines the minimum number of bunch crossing intervals between two successive pulses so that the pulses are not considered to overlap. Counter \( c \) is updated after each bunch crossing according to the following rule:

\[
c = \begin{cases} c + 1 & \text{if } y_1(i) = 0, \\ 0 & \text{otherwise,} \end{cases}
\]

i.e., if a pulse is found in location \( i \) the counter \( c \) is reset, otherwise its contents are incremented at the sample rate.

The architecture with two separate FIR filters was selected for evaluation in the first FERMI prototype. By programming identical coefficients to both filters the use of a single FIR filter for both amplitude and time extraction can be simulated. A modified architecture, based on an FIR-OS filter with two subfilters [17], is also included in the performance comparison.

### 2.2. Feature extraction stage: filter F2

Filter F2 is an FIR-OS hybrid filter structure [12,13]. The input is a time frame containing \( N_2 \) samples \( x_c(i) = \{ x_c(i-n_{2a}), x_c(i-n_{2a}+1), \ldots, x_c(i+n_{2b}) \} \) of the original sequence. Using these samples the filter yields as output the measured pulse amplitude. The FIR subfilters form a bank of filters from which the order statistic operator selects one in each input sequence point \( x_c(i) \). We use the notation \( OS[\cdot] \), for an operator which sorts the input values in an ascending order and selects the \( r \)th largest value as output. The FIR-OS filter is given by

\[
y_2(i) = OS[\phi_1(i), \phi_2(i), \ldots, \phi_m(i)],
\]

where \( r \) is any rank between 1 and \( m \) and \( \phi_i \)'s are the FIR subfilters, with the tap coefficient vector \( \mathbf{h}^i = \{ h^i_1, h^i_2, \ldots, h^i_{N_2} \} \), given by

\[
\phi_i(i) = h^i_1 x_c(i-n_{2a}) + h^i_2 x_c(i-n_{2a}+1) + \ldots + h^i_{N_2} x_c(i+n_{2b}),
\]

1. where the filter length \( N_2 = n_{2a} + n_{2b} + 1 \). An architecture using sample-serial arithmetic in the subfilters is illustrated in Fig. 2. The prototype includes a memory with four sets of FIR coefficients which can be selected individually for each time frame. This enables e.g. switching between a filter optimized for single pulses and a filter optimized for overlapping pulses. The selection is controlled by the overlap flag \( f(i) \) associated with the time frame data.

### 2.3. Optimizing the filter coefficients

The method selected for obtaining the parameters of the predefined filter structure is the mean squared (MS) error minimization. If the filter is a linear one this corresponds to the linear mean squared (LMS) optimization. The procedure of finding the optimal FIR filter weights according to the MS minimization criterion is referred to as the Wiener filtering problem [18]. Here we do not require the operator to be causal, i.e., we assume that the observations to the future of the current sample \( x(i) \) are also available. The set containing the samples of the original sequence and the corresponding operator outputs is called the training set, adopting the terminology used for artificial neural networks. Adaptive filtering [19,20] has a common optimization strategy but the adaptation is performed during operation whereas the filter training is performed prior to operation. Thus adaptive filters can adapt to changing conditions in real-time but they have the drawback of being computationally intensive. The total cost function to be minimized is obtained from the difference between the filter output and the desired output in each input sequence point using the MSE norm:

\[
E = \frac{1}{N_2} \sum_{i=1}^{N_2} (y(i) - \hat{y}(i))^2,
\]
where \( \hat{y}(i) \)'s are the desired filter outputs and \( N_q \) is the number of training pairs in the training set \( Q \). In the case of an amplitude extraction filter the training is performed using pulses extracted from the detector signal sequence. If the time frame containing pulse \( p \) consists of samples \( \{x_p(1), x_p(2), \ldots, x_p(N)\} \) and the corresponding pulse amplitude is \( \hat{y}_p \), the training set has the form

\[
X_q = \begin{bmatrix}
    x_1(1) & x_1(2) & \cdots & x_1(N) \\
    x_2(1) & x_2(2) & \cdots & x_2(N) \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{N_q}(1) & x_{N_q}(2) & \cdots & x_{N_q}(N)
\end{bmatrix},
\]

(12)

\[
\hat{y}_q = \begin{bmatrix}
    \hat{y}_1 \\
    \hat{y}_2 \\
    \vdots \\
    \hat{y}_{N_q}
\end{bmatrix}.
\]

(13)

Several optimization algorithms for nonlinear hybrid filters have been presented in the signal processing literature. The hybrid order statistic adaptation procedure is based on error backpropagation [11]. The FIR-OS optimization has been performed by using adaptive subfilters [21]. Several WOS and FIR-WOS filter adaptation algorithms minimizing the mean absolute error norm have been presented in Refs. [14,15]. Here we use an alternative method to optimize the complete FIR-OS filter structure under the MSE criterion in order to obtain the trainable FIR-OS filter class [17,22,23]. The algorithm is based on conjugate gradient minimization and at the beginning of the procedure the filter tap coefficients are initialized to small random values.

The selection of the training energy (or set of energies) and the optional use of a weighting function affects the filter properties. If the amplitude extraction filter \( F1 \) is trained using low-energy pulses the response corresponds to the matched FIR filter presented in Refs. [4,5]. If, however, high-energy pulses are used in the training process the filter has a better performance in the region where the sample timing jitter is the dominant source of error. By carefully selecting the training set and by using amplitude weighting the filter response can be optimized over a wide energy interval to match the required detector resolution.

3. Filter implementations

In order to complete the design of the digital filter section, the arithmetic accuracy of the implementation has also to be taken into account. Three sources of quantization error exist in the digital filtering approach: the quantization of the input samples, the finite representation of the FIR tap coefficients, and the accumulation of rounding errors in the arithmetic operations. The number of bits in the input data samples is limited by the resolution of the FERMI ADC. The data obtained from the renormalization lookup table (LUT) contains 16 bits, representing the dynamic range of particle energies from e.g. 30 MeV to 2 TeV. The whole dynamic range is utilized by filter \( F2 \) whereas the input of filter \( F1 \) is truncated to 14 bits after the channel summation. The number of bits at the filter \( F1 \) output is defined by the requirements of the first-level trigger process. In order to maintain generality we have adopted an architecture with 11 data bits, representing a dynamic range from 250 MeV to 500 GeV. The above-mentioned figures are given as examples and should be scaled for each application.

If the input sample is represented using \( n_t \) bits (unsigned format) the width of quantization \( q_t \) is given by \( q_t = 2^{-n_t} \). Simulations have shown [24] that the error sequence can be closely approximated by a white-noise sequence which is uncorrelated with the input signal and uniformly distributed over \( [-q_t/2, q_t/2] \). Therefore, the standard deviation of the quantization error is given by [24]

\[
\sigma_q = \frac{q_t}{\sqrt{12}} = 2^{-n_t/2} \frac{2}{\sqrt{12}}.
\]

(14)

The rounding errors are accumulated in the filters as the intermediate products are represented using less than \( n_t + n_c - 1 \) bits where \( n_c \) is the number of data bits at the multiplier input and \( n_c = 1 \) is the number of coefficient bits (2's complement format). If the products are rounded to \( n_t \) bits prior to each summation the accumulated error at the filter output is given by

\[
\sigma_f = \sqrt{\frac{N}{12}} \times 2^{-n_t},
\]

(15)

where \( N \) is the filter length. Substituting \( n_t = 14 \) for filter \( F1 \) and \( n_t = 16 \) for filter \( F2 \) we obtain the errors \( \sigma_f = 3.94 \times 10^{-3} \) and \( \sigma_f = 1.25 \times 10^{-3} \), respectively. The effect of finite coefficient wordlength on the filter response can be qualitatively determined by computing the filter output for different word lengths. Again, the error in individual coefficients is uniformly distributed over \( [-q_c/2, q_c/2] \) where \( q_c = 2^{-n_c-1} \). To obtain the required performance we select \( n_c = 8 \) for filter \( F1 \) and \( n_c = 10 \) for filter \( F2 \).

Two concurrent design methodologies have been used when implementing the digital filter section. The whole digital part of FERMI is modelled using VHDL. The critical parts of the system have also been optimized to obtain maximum performance with minimum circuit area. We start by presenting fault-tolerance architecture considerations and then continue by introducing the VHDL implementation and the hardware optimization of the basic building blocks for filters \( F1 \) and \( F2 \).

3.1. Fault tolerance considerations

Due to the lack of definite knowledge on the behaviour of microelectronics devices in hard radioactive environment, we refer to a functional fault model in this consideration [25]. As described in Ref. [2] fault tolerance techniques have been adopted in order to guarantee the completeness
Fig. 3. 16 × 10 bit pipelined multiplier unit for filter F2.

(no losses) and the creditability (no corruption, masking or aliasing) of data. To provide concurrent error detection in all arithmetic units, e.g. in the inner-product multiplier subsystems, residue codes have been adopted [26]. Each code word \( x_r(i) \) is obtained by cascading the data word \( x(i) \) and the residue bits

\[
x_r(i) = \{ x(i) | R_b[x(i)] \}
\]

where \( R_b[ \cdot ] \) denotes the residue operator with respect to 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 residue bits. An error is detected whenever the actual and expected residues differ. This approach is useful to detect all single and multiple errors which produce a non-zero residue in the given base \( b \). In the FERMI application a reasonable balance between the hardware overhead and the detection capabilities is achieved using base \( b = 3 \). The implementation of multipliers and adders is selected to guarantee that single faults result in single errors. Since faults in the accumulators typically appear as single errors, the residue code is also useful to protect the memory elements.

3.2. Filter Fl implementation

Filter Fl includes two FIR filter modules and four sign comparator units to implement the peak finding and energy thresholding operations. The input data are truncated to 14 bits (unsigned) after the summing network. The FIR tap coefficients are represented using 8 bits (2's complement) and encoded with quasi-Booth method [27] to reduce the number of multiplier rows by a factor of two. The applied coding system transforms the coefficients \( h_{k0} \) and \( h_{k1} \) into a form which allows the four-row product array to be reduced to a two-row equivalent array through carry-save adders. The convolution result is obtained in the redundant two-row form with a latency of one clock cycle. A further clock cycle is then required to output the result in the standard binary non-redundant form using a fast carry look-ahead adder. The combination of the multiplier and an adder forms a basic functional module which can be bypassed in case of a fault. Although five taps are sufficient for filtering, a total of six taps are provided to maintain a high performance under graceful degradation.
3.3. Filter F2 implementation

The core of filter F2 consists of three parallel inner-product multipliers [28,29], each corresponding to one FIR subfilter of the FIR-OS structure. The inner-product multipliers are implemented by cascading a multiplier unit (Fig. 3) and an adder-accumulator unit (Fig. 4). The partial products $h_kx(i)$ are recursively added to the contents of the accumulator to obtain the final subfilter output. In order to fulfill the data throughput requirements a pipelined structure has been used [28,29]. Each pipeline step includes the addition of four operands in the carry-save form.

The input data samples $x(i)$ are by definition positive numbers and they are fetched as 16-bit unsigned values in a bit-parallel format. The subfilter tap coefficients $h_k$, represented using 10-bit two's complement values, are input in a parallel/skewed format. The 10 bits are grouped into three skewed groups with $2+4+4$ parallel bits in each. This mixed operand format is due to the internal structure of the pipelined multiplier, designed according to the Braun’s scheme where four multiplier array rows are collapsed into a single pipeline step. The 10 least significant bits of the product are output in the same parallel/skewed format. The 16 most significant bits are implicitly represented by the sum and carry bits generated from the row of full adders at the bottom of the multiplier array. As the final product will be stored in the accumulator and then forwarded to the comparators we have adopted the skewed format also for the most significant part of the result.

As the number of bits in the coefficients is not divisible by four, the third pipeline stage contains only two multiplier array rows. In order to avoid the inherent capacity loss two rows of the pipelined adder are included in the same pipeline stage. The part of the multiplier generating the 16 most-significant bits has been designed to minimize the number of adders and latches [28,30]. In the adder-accumulator unit the output $h_kx(i)$ of the multiplier is added to the current contents of the accumulator unit. To simplify the hardware implementation the final subfilter output is truncated to 20 bits. The method adopted for the multiplier coefficients requires that the last partial product row is generated using NAND gates, instead of AND gates used for the other rows, and that the constant

$$C_2 = -2^n + n_x + 1 + 2^{n_x-1},$$

where $n_x$ is the number of bits in the data samples and $n_x$ is the number of bits in the coefficients, is added to each intermediate product. This can be achieved by preloading the constant $N_2 C_2$ where $N_2$ corresponds to the number of subfilter tap coefficients into the accumulator.

The computation of the expected modulo-3 residue values is performed in parallel with the computation of the subfilter output. The expected residues are compared with the obtained residues to detect possible faults in the inner-product multiplier unit. The iterative use of the same faulty device in the multiplier unit can induce either multiple errors or error masking; these may not be detected, causing a critical reduction of system credibility. However, since the checking is performed on the result of each iteration of the inner product algorithm this cannot happen. Error masking only occurs when multiple transient faults induce multiple errors with a null residue. In the case of sequential multiple errors the detection of the first error forces the invalidation of the subsequent results. The graceful degradation is performed by disconnecting the faulty subfilter unit and by reprogramming the remaining FIR subfilters with a new set of tap coefficients.

The outputs of the three inner-product multiplier units are connected to the order statistic operator. This operator identifies the minimum, the median and the maximum of the FIR subfilter output values and directs the value corresponding to the programmed rank $r$ to the filter output. In order to minimize the hardware complexity the sign of each difference is not obtained by performing the complete subtraction but by using a dedicated circuit for bit comparison of the two unsigned operands. The error detection in the order statistic operator is realized by duplicating the comparator units and by checking the consistency of the outputs.
Table 1

<table>
<thead>
<tr>
<th>Property</th>
<th>Filter F1</th>
<th>Filter F2</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of gates</td>
<td>20000</td>
<td>25000</td>
<td>45000</td>
</tr>
<tr>
<td>Silicon area [mm²]</td>
<td>29.5</td>
<td>29.5</td>
<td>59</td>
</tr>
<tr>
<td>Output sample rate [MHz]</td>
<td>&gt; 40</td>
<td>&gt; 4.4/N₂</td>
<td></td>
</tr>
<tr>
<td>Latency (clock cycles)</td>
<td>5.5</td>
<td>N₂ + 4.5</td>
<td></td>
</tr>
<tr>
<td>Number of data bits</td>
<td>14</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Number of coefficient bits</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

3.4. Hardware description and implementation

In the prototyping phase, VHDL simulations provide a fast method for evaluating different architectural solutions. The final implementation was selected after a comparison of several architectures to optimize speed, latency and circuit area of the filter units. The VHDL model will also provide a high-level system description for further subsystem optimization.

The hardware was synthesized from the register transfer level (RTL) VHDL description using the ES2 0.7 µm CMOS technology. The final filter chip including the two filter stages, the register file for filter F2 coefficients and error detection and reconfiguration system is a 45 000 equivalent gate module with a silicon area of 59 mm². The maximum clock frequency under worst-case conditions, equal to the processing rate of filter F1, is estimated to be superior to 40 MHz. A detailed list of the filter hardware characteristics is shown in Table 1. Due to the sample-serial architecture the output sample rate and the latency of filter F2 are dependent on the length of the time frame, denoted by N₂. The latency values represent the total filter latencies, including the error correction code handling in filter F2. The layout of the prototope filter chip is shown in Fig. 5. The left-hand side of the layout contains the channel summation block and the FIR filters F1e and F1t with five taps each. Below these are the peak finding and energy thresholding blocks. The right-hand side contains the three inner-product multipliers of filter F2 and the order statistic operator.

4. Performance evaluation

In order to evaluate the amplitude and time extraction performance of the FERMI filters two sets of simulated detector signals were created. The sets used for filter optimization, denoted by A' and B', are uncorrelated with the corresponding evaluation sets, denoted by A and B. Set A' contains single pulses with a bipolar shaping function g[. The pulses are corrupted by sample timing jitter with σₙ = 2 ns, electronics noise with σₑ = 50 MeV and pile-up noise with σᵤ = 60 MeV. In addition to these artefacts, set B' includes overlapping pulses [17]. Amplitude weighting was applied to modify the height and thus also the cost function of individual training pulses.

The amplitude extraction performance of different filter structures was compared using the normalized root mean squared error RMSE[x]/σ criterion, taking the nominal pulse amplitude 0.5 as the reference value. The normalized RMSE value for energy E₀ is obtained by dividing the root mean squared error of the filter output by the nominal pulse amplitude 0.5. The evaluation sets contain pulses within the energy range 0.5–2000 GeV, 2000 pulses at each energy. The time extraction performance is obtained by dividing the number of incorrectly located pulses by the total number of pulses. In this case the size of the evaluation sets A and B is 5000 pulses per energy. The amplitude extraction performance of filters F1 and F2 was also evaluated using signals acquired from a liquid argon calorimeter prototype. The full LHC energy scale was emulated by adjusting the gain in the external analog stages.

4.1. Amplitude extraction with filter F1

Five-tap FIR filters and an FIR-OS filter with two sub-filters and five taps in each were trained using sets A' and B'. Pulses of 2, 20 and 200 GeV were used in the training, 500 pulses per energy. The amplitude weights were 0.5, 0.1 and 0.02, respectively, meaning that e.g. the samples of the 2 GeV pulse and the correct pulse amplitude were multiplied by 0.5 prior to the training process. The normalized RMSE values are shown in Figs. 6 and 7. The FIR A filter performs well for the energy region above 20 GeV, where the sample timing jitter is the dominant source of error. The FIR B filter is slightly better for the low-energy region. The FIR-OS filter is an efficient combination of a deconvolution FIR filter and the FIR A filter, giving a precise amplitude.
estimate over the whole energy range. It also achieves a similar performance for set B whereas the performance of the FIR filters starts to degrade. The pulse amplitude gain of the FIR-OS subfilter 2, obtained by filtering single pulses without electronics and pile-up noise, was found to be 3% below unity. As the FIR-OS filter rank $r = 2$ the subfilter giving the larger output value is selected in each sample position. For this reason subfilter 2 is used only when subfilter 1 extracts an energy estimate which is too small, e.g. when the superposition of several pulses takes place.

For visual comparison a complete signal sequence containing 5 GeV pulses corrupted by all the artefacts was also applied to the trained filters. Part of the signal and the filter outputs are shown in Fig. 9. The FIR A filter has a relatively wide response on the pulses: its output it less affected by electronics noise and sample timing jitter than the output of the FIR B filter. However, it is unable to produce a correct output for overlapping pulses. It is clear that an FIR filter which has a high amplitude precision has a relatively low time-domain resolution, and vice versa [18, pp. 296-298]. The FIR-OS filter uses different subfilters for single and overlapping pulses and achieves simultaneously a good time-domain resolution and a high amplitude precision.

4.2. Time extraction with filter $F_1$

The event time identification is considered correct if the three-point maximum finder following filter $F_1t$ is able to locate the maximum at the expected position $i$. The superposition of several pulses together with the two noise com-
Fig. 8. Time identification error of filter F1.

Fig. 9. Original and filtered signal sequences.
components and the sample timing jitter modifies the ideal pulse shape so that the maximum is shifted along the time axis. The percentage of incorrectly identified pulses is shown in Fig. 8. The operator shown is a three-point deconvolution filter; an equal performance was obtained with the FIR-OS subfilter 2. Several other optimized filters were also applied to the signal shaping prior to the maximum finder. For set A the differences between filters are rather small, except for the FIR A filter; in this case the low time-domain resolution causes errors when pile-up pulses are within a few sampling intervals from the pulse to be measured. Also the maximum finder without filter F1t achieves a good performance. For set B the filters with a high time-domain resolution clearly outperform the maximum finder and filters with a wide time-domain response.

4.3. Amplitude extraction with filter F2

An eight-tap FIR filter and FIR-OS filters with two and three eight-tap subfilters were trained using set B' and the extracted single and overlapping pulses, 200 pulses per energy. The amplitude weights to match the filter performance to the detector performance in the training procedure are given by 0.2/√E where E is the pulse energy in GeV. The normalized RMSE values are shown in Figs. 10 and 11. The effect of the overlap flag f(i) on the FIR-OS filter performance is rather marginal. However, the performance of a single FIR filter can be improved using the pulse overlap information. In graceful degradation the feature extraction performance is reduced stepwise by disconnecting faulty subfilter units. Comparing Figs. 6 and 10 we note that the performance of
filter F2 is almost identical to the performance of filter F1 in the low-energy region. In the energy region where sample timing jitter is the dominant source of error [22] the extra subfilter taps and the finer sample quantization increase the amplitude extraction precision. The amplitude error in this region is close to 0.2 % for the sample timing jitter with \( \sigma_t = 2 \text{ ns} \).

5. Conclusions

The architecture and implementation of the FERMI digital filter section, consisting of optimized FIR and FIR-OS filters, were presented. The filter optimization was performed using a training algorithm based on the total error minimization by conjugate gradient search. The simulation results show that while an FIR filter has to compromise between the amplitude precision and the time-domain resolution the FIR-OS filter can simultaneously obtain a high performance with respect to both criteria. This means that the FIR-OS filter produces a precise amplitude estimate both in the low-energy region (pile-up noise and overlapping pulses as main artefacts) and in the high-energy region (sample timing jitter being the main source of error). For the high-energy region, the amplitude error obtained under severe jitter conditions is close to 0.2 %. As the FERMI system will be installed in a high radiation environment, fault tolerance, as a combination of distributed error diagnostics and graceful degradation schemes, was introduced. The layout of the prototype system has been designed using the VHDL hardware descriptive language. The VHDL description is considered an efficient tool for system simulations, architecture comparisons and prototype implementations during the design phase. The basic building blocks of the two filter stages will be further optimized to minimize the circuit area and to increase the level of redundancy for higher performance under graceful degradation.

Acknowledgements

We thank the ATLAS and CMS Collaborations for many fruitful discussions. We thank S. Cittolin for his continuing support of the project.

Financial support from CERN to the CERN group, from INFN, Italy to the Milano and Pavia groups, from IN2P3, France to the Paris group, from NFR, Sweden to the Fysikum and KTH groups and from NUTEK, Sweden to the Fysikum and Linköping groups, as well as their joint support to our industrial partners, is gratefully acknowledged.

References