EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–ECP/94–21
8 December 1994

NANA — AN INTEGRATED SIGNAL PROCESSOR AND RECORD BUILDER FOR THE LEVEL-2 READ-OUT OF ASYNCHRONOUS EVENT-FILTERING DIGITAL PIPELINES

P. Gomes, M. Mota
LIP, Lisbon, Portugal

J. Christiansen
CERN, Geneva, Switzerland

Abstract

NANA implements some fast and simple processing algorithms, for use in future High Energy Physics (HEP) experiments. After the first-level trigger, the data rate is sufficiently small to allow channel merging and some processing with dedicated hardware.
For detectors whose signal is companded prior to digitization, the data samples need to be expanded to the original dynamic range, as further physics analysis usually deals with linear values.
A 16-bit integer, piece-wise polynomial approximation of the expansion function recovers the original dynamic range, without introducing significant degradation in the detector's intrinsic resolution.
A 16-bit integer, multiplier and adder-accumulator is accurate enough for time information extraction and for determination of the energy deposited in one channel. The time resolution is more than sufficient to correctly identify the bunch crossing that corresponds to a particular pulse. The energy accuracy is limited only by the detector's resolution.
The extracted features carry all the information subsequently needed. The whole set of samples of a pulse can thus be replaced by a few data words.
Initially aimed at the calorimetry front-end, the algorithms may be applied to any similar sampled detector signal.

Presented at the 1994 IEEE Nuclear Science Symposium,
Oct. 30–Nov. 5, 1994, Norfolk VA, USA.

1 ARCHITECTURE

Given the continuous development of microelectronics, it is nowadays possible to integrate, within the front-end of data acquisition systems, procedures that were only feasible after high trigger levels or even off-line.

NANA is intended for use after zero suppression and level-1 trigger data reduction [1], where it is possible to merge several detector channels into a single chip. Additionally, some digital signal processing can still be performed.

The architecture includes (Fig. 1): channel multiplexing, data linearization, feature extraction, fine trigger matching, and pre-formatting for the global event builder. A high level of programmability is believed vital in order to cope with different requirements. Direct application in future HEP experiments was also considered, especially for calorimeters.

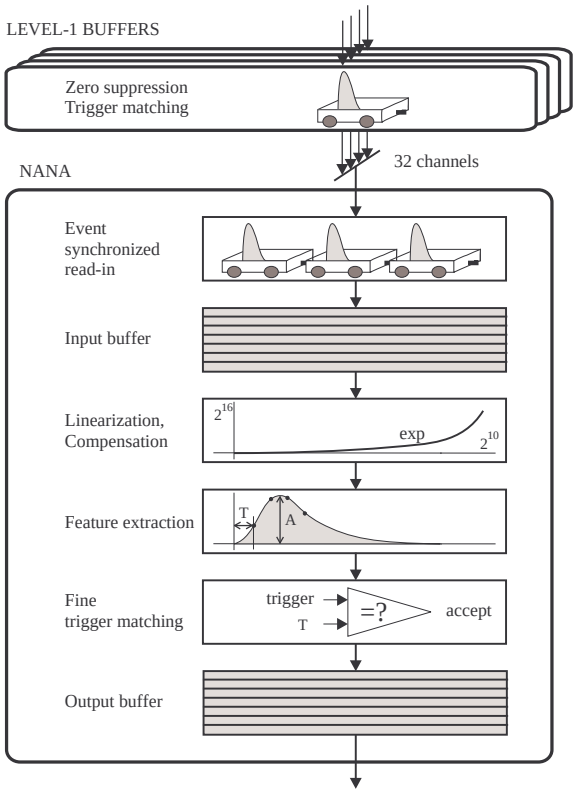

Fig. 1 NANA architecture.

1.1 Event synchronized read-in

Data is read-in from level-1 buffers, combining several channels in an event-synchronized way. A simple token ring protocol is used for this purpose (Fig. 2).

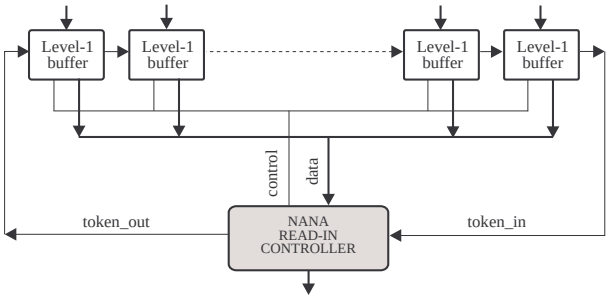

Fig. 2 Channel multiplexing by event synchronized read-in.

Each channel is sequentially checked for the availability of data corresponding to a given trigger selected event. If a particular channel has data, it grabs the token and, at the tick of a common synchronizing clock, sends all data and any extra information through a shared bus.

When it has finished writing, or if there were no data for that event, the token is passed to the next channel in the chain. When the token gets back to the read-in controller, all channels are simultaneously signaled that a new cycle begins for the next triggered event.

The protocol can be configured such that each channel sends additional information, such as its ID or any particular characteristics it found on the event.

1.2 Linearization

In the cases where the dynamic range of the signal produced by a detector is larger than that of available ADCs, analogue signal companding is needed before digitization. This must then be followed by a digital linearization.

Fast linearization can be achieved through look-up tables, needing relatively large amounts of memory for the dynamic ranges we are addressing. This becomes especially significant when different compensations — for example, for pedestal and gain — are needed for each of the merged channels.

Due to previous data reduction, the data rate into NANA is significantly lower than the initial sampling rate (40 MHz). Some time can thus be spent to implement the inverse of the companding function. However, its ‘precise’ computation would be unaffordably time-consuming.

We show that a piece-wise polynomial approximation of the linearization function is able to expand the dynamic range, without introducing noticeable degradation of the detector's resolution.

Several types of calorimeters, such as fibre, liquid ionization (namely Liquid Argon–LAr) and crystal were considered. Nonlinear ADC functions like hyperbolic, logarithmic and piece-wise linear were also taken into account.

One of the advantages of the proposed scheme is that one functional unit may be shared by many channels. Only a small set of (stored) constants are channel-dependent, in contrast to a large look-up table per channel.

1.3 Feature extraction

Usually, a detector's output signal needs to be shaped as a compromise between pile-up noise and electronic series noise. In Fig. 3, three types of pulse shapes are shown, together with the drift current I(t) of a liquid ionization calorimeter. The unipolar, bipolar and tripolar shapes result from one, two, or three differentiations on I(t), followed by a couple of integrations.

The pulse shape is normally well defined and fairly invariable with time. We show that the signal-characterizing parameters can be extracted from a linear combination of a restricted number of samples.

A reduction of raw data to a few meaningful parameters is then possible. This information can be directly used in further stages of event selection or analysis. Data transfer bandwidth and trigger processing power can then be more efficiently allocated.

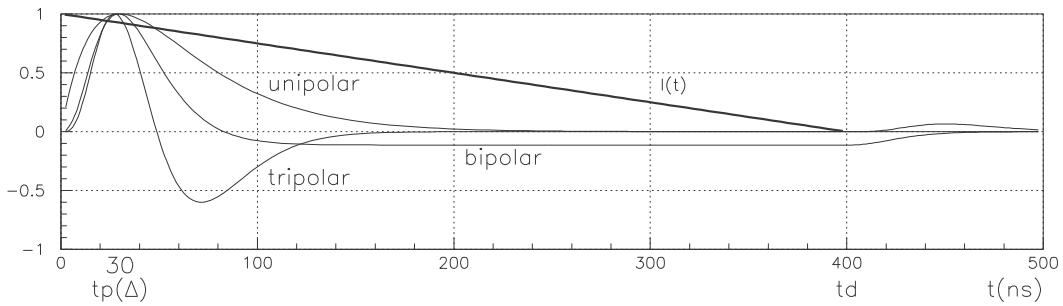

Fig. 3 Typical pulse shapes and the triangular current pulse of the LAr calorimeter.

1.4 Fine trigger matching

The zero suppression algorithms select a fixed number of consecutive samples, corresponding to a neat pulse above the baseline and noise fluctuations [1].

Each set of samples is identified by a 'time tag', corresponding to the first sample in the set. However, this first sample detected above threshold is not always at the same position within the pulse: a high pulse crosses the selection threshold before a small one of the same shape and width (Fig. 4).

Considering that the analog signal always starts at a fixed interval after the interaction, there is then an uncertainty in the position (t_1) of the first sample.

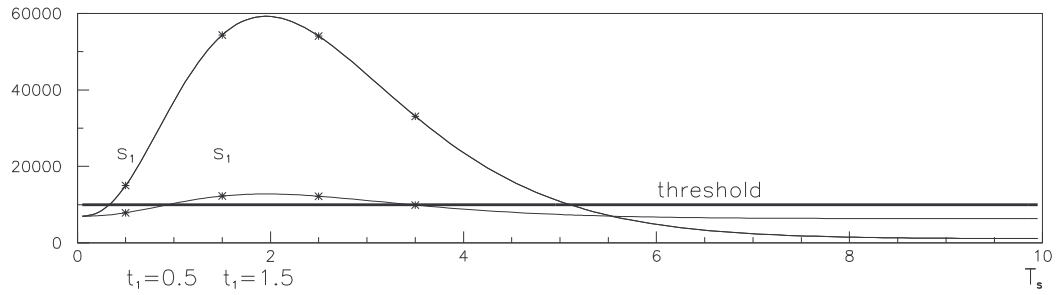

Fig. 4 The position of the first sample above threshold depends on the amplitude of the pulse.

The matching of the pulse with the level-1 trigger decision thus had to be initially performed within a coarse time window [1]. The extracted timing information can now be used to implement a finer trigger matching.

1.5 Read-out

The expected data rate supplied by one NANA is not sufficient to fully utilize a high-speed link, as required in a global event-building network — ATM, for example.

Event fragments collected from several NANAs can be merged together in a record structure appropriate for injection into the network. A wider level of event building is thus accomplished before the global building in the network.

Alternatively, processed data can be temporarily stored in external level-2 buffers, waiting for the level-2 trigger. A study of several possible read-out schemes and data structures can be found in [2].

2 LINEARIZATION

For the LHC (Large Hadron Collider, at CERN), the dynamic range currently assumed for calorimetry is about 16 bits. As ADCs with such resolution at LHC sampling rates (40 MHz) are not obvious at the moment, a smaller resolution — 10-bit, for example — nonlinear converter must be used.

A further digital linearization block should convert the samples back into a linear scale. Several companding functions can be used: hyperbolic [3], logarithmic [4], piece-wise log [4], piece-wise linear, or composites of these.

Figure 5 shows three examples of functions that compand the linear range 0.1 GeV, 2000 GeV into 10 bits. Their respective inverse functions will recover the 16-bit dynamic range.

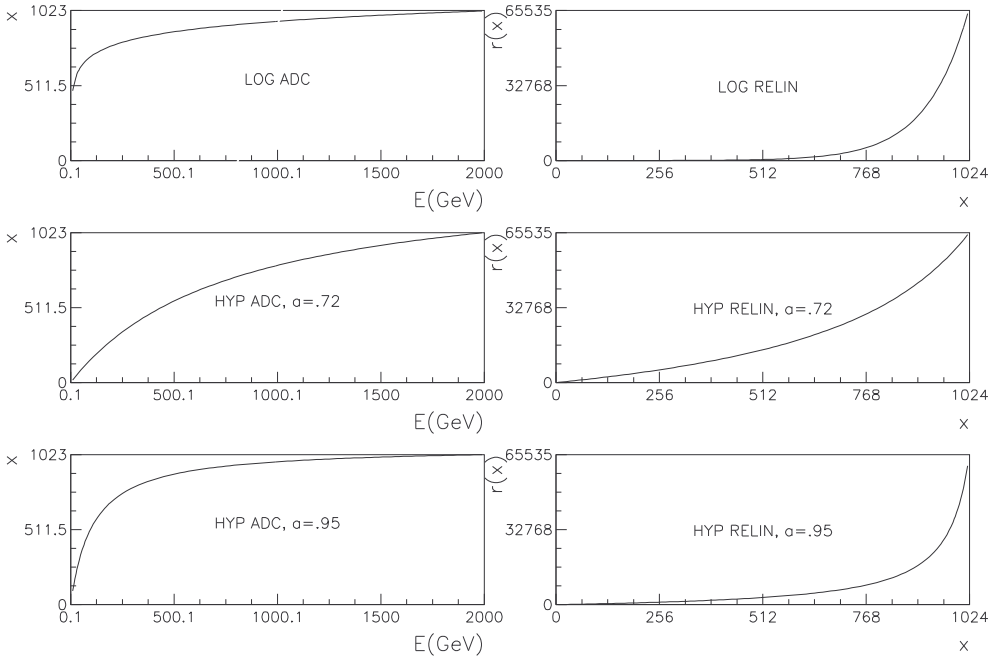

Fig. 5 Comparison of three companding functions and respective inverses.

The first function can be implemented with a logarithmic amplifier prior to the ADC. The hyperbolic functions are accomplished by driving the reference resistor chain of the ADC with a fraction (a) of the input signal.

Reference [4] demonstrates that a 10-bit log ADC can be used with fibre, LAr and crystal calorimeters, while keeping the resultution degradation below acceptable bounds (about 10%).

We propose to approximate the linearization function with a polynomial of limited degree, using integer arithmetic:

$$f(x) \approx p(x) = a_0 + a_1*x + a_2*x*x + a_3*x*x*x \; . \qquad (1)$$

It is assumed that only one integer multiplier and one integer accumulator are available, in order to minimize area and power consumption.

Each power of x is calculated from the previous one and then multiplied by the respective coefficient, to be next accumulated with the term calculated before (Table 1).

Therefore, a third-degree polynomial requires five multiplications and three add/accumulations for each incoming sample.

Table 1

Sequence of operations

MULTIPLIER	ACCUMULATOR
MUL = $x*a_1$	
MUL = $x*x$	ACC = $a_0 + (x*a_1)$
MUL = x^2*a_2	
MUL = $x*x^2$	ACC = ACC + (x^2*a_2)
MUL = x^3*a_3	
	ACC = ACC + (x^3*a_3)

A quite large degree polynomial would be needed to fit the function over all the input domain; this would imply too much time spent in calculations.

The partitioning of the input domain in several ranges allows the use of a lower degree polynomial for each range. Given an input value, the range it belongs to is determined by inspection of its most significant bits.

The polynomial coefficients have been determined by a chi-squared fit to the linearization function within each range, and then converted to integers.

The precision of the output depends on the number of bits used in each operation and on the sequence of the operations themselves. To use the multiplier and adder's full precision, some scalings between operations are performed.

The algorithm was tested with the three linearization functions mentioned above, converting sample values from 2^{10} to 2^{16}. The results were compared against the resolutions of fibre and LAr calorimeters.

Figure 6 compares the errors introduced by the linearization procedure (σ_L) with the LAr detector resolution (σ_d) and the quantization noise (σ_q) of the 10-bit log ADC.

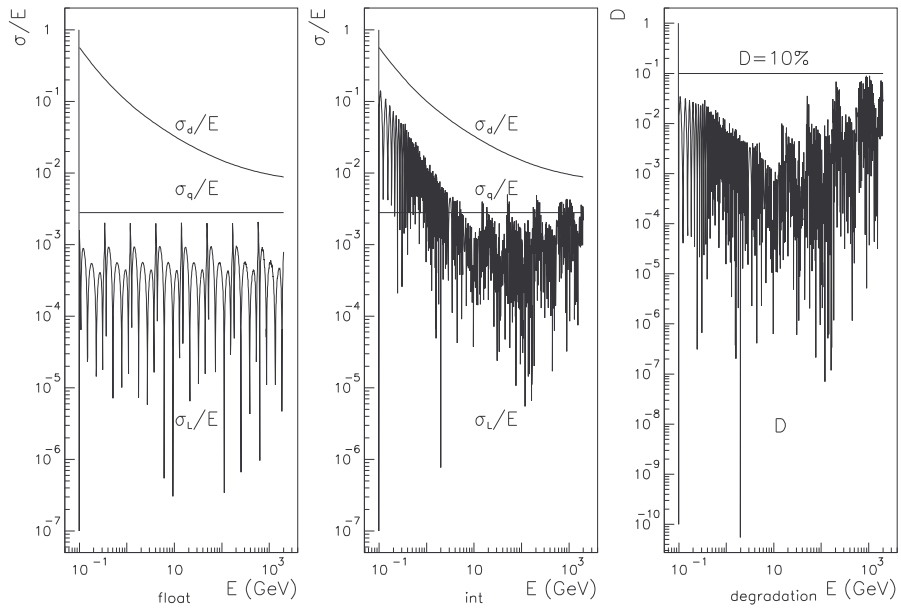

Fig. 6 Linearization performance.

The *first plot* shows the error σ_L in approximating the function by a polynomial, using float precision. A third degree is enough, if the input domain is divided in eight ranges. The error is always below the ADC's quantization noise (whose degradation on the detector resolution is negligible, as seen upwards). Hence, the effect of σ_L is also insignificant.

The *second plot* shows the effect of using limited accuracy (16-bit integer) arithmetic. We can see that the associated precision σ_L is always better than the detector's resolution.

The *third plot* confirms that the resolution degradation

$$D = (\sigma_d \oplus \sigma_L)/\sigma_d - 1 \,, \tag{2}$$

introduced by the linearization, is always below 10%, which is generally considered as an acceptable bound.

A third degree polynomial with eight ranges is therefore adequate for the LHC LAr calorimeters' resolution and dynamic ranges. As fibre calorimeters are less precise than LAr, these results also apply. Although crystal is a more accurate medium, we have seen that the algorithm is still applicable.

Pedestal correction and gain compensation for the ADC can easily be included in the polynomial constant and in the first term coefficient.

3 FEATURE EXTRACTION

Several algorithms [5, 6] have been investigated and used for signal feature extraction in HEP.

We have evaluated the possibility of extracting the bunch crossing information (T) and pulse amplitude (A) from a small number of samples. If T and A are extracted with enough accuracy, there is no further need to transmit the samples themselves. Furthermore, T can be used in fine trigger matching, as stated above.

We are concerned about local (channel-based) feature extraction only, without considering neighbouring channels. Piled-up pulses are not taken into account.

Simple arithmetic must be used for real-time implementation. A linear filter, with a minimum number of taps, arises as the basic choice.

In our algorithm, the FIR coefficients are interpreted as the weights of a linear neural network. However, any other appropriate set of coefficients can be loaded into the filter.

3.1 Time extraction

The error on T determination must be at least less than half the sampling period ($< 0.5\ T_s$), in order to correctly identify to which bunch crossing the pulse belongs. Moreover, there must be room for shape variation effects, baseline calculation error, etc., in order for the extraction to be precise enough to be usable in further physics selection and analysis.

Given a fixed pulse shape, its peak is always at the same distance from the pulse start and could be used as a measure of the bunch-crossing instant.

The largest sample of a pulse is also the one closest to the peak. As the samples are evenly spaced by T_s, the largest one can be as far apart from the peaking time as $\pm 1.0\ T_s$. Therefore, using the largest sample's position as an approximation of the peaking time gives a guaranteed precision of only $1.0\ T_s$. This is unacceptable.

Let us then define the time to be extracted (T), as the interval between the pulse start and the first sample (s_1) accepted above threshold. We intend to compute T as a simple linear combination of the values of a restricted set of samples.

$$T \equiv t_1 \approx c + \sum_{i=1}^{4} \left[a_i * (s_i - B) \right]. \qquad (3)$$

Note that this is not a fit of the pulse shape by a polynomial. Rather, we are defining T as a function of the sample values and approximating this function by a first degree polynomial on the (N-dimensional) vector $\underline{s}$.

Among the several ways to calculate the coefficients (c, a_i), we have chosen to interpret them as the ‘weights of a linear two-layer feed-forward neural network’. In this context, those coefficients can be computed by the ‘delta rule’ [7]. Of course this is no more than an N tap linear filter whose coefficients are calculated with a particular numerical procedure.

Other network types were tested, such as sigmoidal threshold multilayer feed-forward with back propagation learning [7], attaining interesting results. However, they imply the use of nonlinearities, which slows down the execution or increases the silicon area by the use of look-up tables.

The set of samples to be used must include the pulse region that has the main portion of the time information, which is normally the fast-rising edge. Our simulations show that, for bipolar shaped pulses, the first four consecutive samples are the most appropriate.

We also found that a fixed set of filter coefficients is not enough to achieve the required accuracy. The interval of possible values for t_1 had to be partitioned into four ranges. The range selection is done by comparison of the values of two given samples.

The signal baseline can vary considerably with time and the extracted T is sensitive to its value. The time extraction should thus be preceded by a baseline (B) subtraction. The value of B is supposed to have been calculated upstream [1], or can easily be extracted from pre-samples.

If a pulse of a given shape and width is sampled at fixed instants (as is the case), the sample values are proportional to the pulse amplitude (A). Extracted as a linear combination of those values, T will also be proportional to A.

As we want to simplify arithmetic, division by A is excluded. To limit the dependence of T with A, the samples can be scaled (after baseline removal) by powers of two, in order that the value of the largest sample falls into the upper half-range of possible values (2^{15}; 2^{16}).

This is not an exact normalization, but reduces the dependence with A, and has the advantage of being implemented by simple shifts.

In Fig. 7 we show the error in time extraction, as a function of the amplitude and the position (t_1) of the first sample (s_1), for four different implementations of the algorithm. The energies are in (4 GeV; 2000 GeV), and t_1 in (0.5 T_s; 1.5 T_s). We can see that the time extraction error is always far less than half the sampling period (±0.50 T_s).

In the *first plot*, no arithmetic is performed, and the extracted time is set as a constant, equal to the mid-point of each time range. The error is limited to about ±0.20 T_s.

In the *second plot*, time is extracted as a linear combination of the first 4 samples. The baseline is removed but no scaling is performed on the samples. The error is then limited to ±0.16 T_s.

In the *third plot*, the pulses are now scaled to the upper-half range of energies. Some improvement is achieved, with the error now limited to ±0.10 T_s.

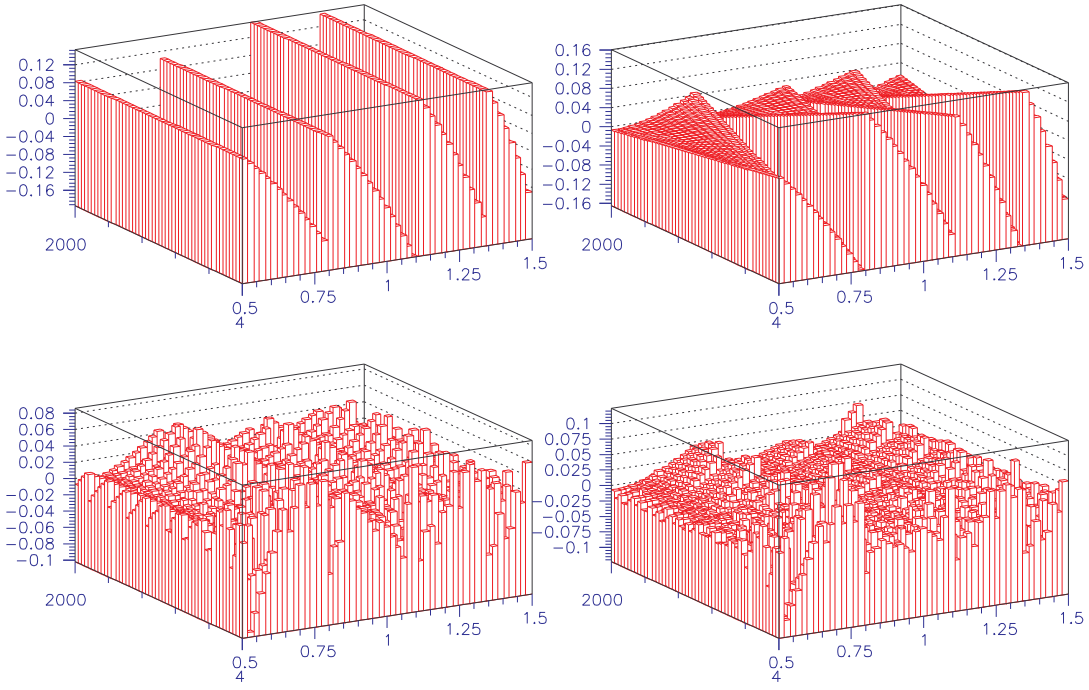

Fig. 7 Time extraction error for energies in (4 GeV; 2000 GeV)

The *fourth plot* shows the effect of a scaling to the upper-quarter range of energies. There is a general reduction in the error, except for low-energy pulses. This is useful in cases where T extraction needs to be more accurate for higher-energy pulses, at the expense of lower T resolution for small pulses.

3.2 Amplitude extraction

The samples that carry amplitude information are the ones around the pulse peak. For a bipolar pulse, they are again the first four samples.

For the value of a sample at a given time to be proportional to the pulse amplitude, the baseline (B) has to be subtracted.

Again, the coefficients are determined through the delta rule and four ranges are defined the same way. As we do indeed want to extract the amplitude dependency of the sample values, there is no sense in scaling them.
The algorithm then becomes:

$$A = \sum_{i=1}^{4} \left[b_i * (s_i - B) \right]. \tag{4}$$

In Fig. 8, the fractional standard deviation (σ_A/E) of the extracted amplitude (using 16-bit integer samples) is compared with the detector's fractional resolution (σ_d/E).

For reference, we show the error when both the samples and arithmetic have float precision ('all float'). LSB/E (16-bit) is the quantization error associated with the 16-bit integer precision. The limit for an acceptable degradation on the energy measurement, $\sigma/E(D = 10\%)$, is also shown.

In the *first plot*, the integer samples were submitted to 16-bit integer arithmetic, whereas in the *second plot* the calculations had float precision.

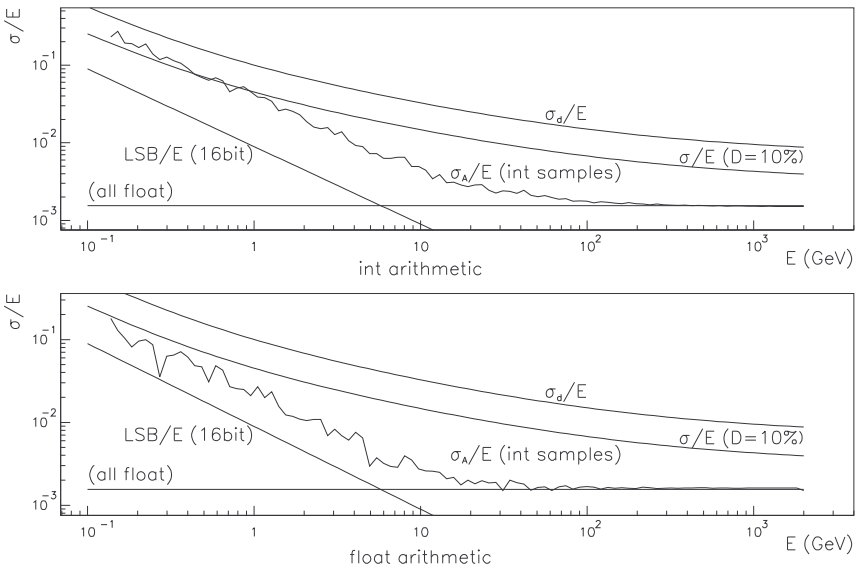

Fig. 8 Amplitude extraction error.

We can see that for energies above some 20 GeV the extraction error is independent of E and of the number representation. Below that point, it follows the slope of the quantization error. In this region, if the arithmetic has float precision, the result is slightly better than using integer arithmetic.

A degradation below 10% is guaranteed for energies above 1 GeV. Below 0.4 GeV, the degradation slightly surpasses 10%, although still remaining better than the detector's resolution.

This is mainly due to the quantization noise introduced by the conversion of real-valued samples into 16-bit integers. As expected, the effect is noticeable only for small amplitude pulses. If float precision is used for samples and arithmetic, the error ($\approx 0.15\%$) is absolutely independent of E, as expected.

4 CONCLUSION

The signal processing algorithms presented here have been defined and completely simulated using the C language. We have thus confirmed that they are adequate for LHC calorimetry. The channel multiplexing read-in and linearization blocks, those that are the most time-critical, have been described and simulated in Verilog HDL.

Considering a bunch-crossing rate of 40 MHz, a channel occupancy of the order of 5% and a level-1 trigger rate of 100 KHz, it is possible to merge 32 channels into a single chip using 1 μm CMOS standard cell technology.

This study was conducted with the goal of integrating signal processing together with buffering, local event building, and some control functionality, within a single ASIC. A separate DSP chip would have its capabilities largely wasted and need external circuitry to implement actions other than signal processing.

Additionally, general purpose DSPs, although becoming more powerful and faster, have a few drawbacks due to their complexity: high power consumption and heat dissipation, large area, and difficult implementation in RadHard technologies.

An interesting possibility might be the inclusion of an already-designed DSP core.

REFERENCES

[1] M. Mota et al., MEC3 — A pipelined zero suppression and trigger matching chip, presented at the IEEE 1994 Nuclear Science Symposium, also available as CERN–ECP/94–23.

[2] J. Christiansen, NANA buffer interface and control, CERN/ECP–MIC internal note, Aug. 1993.

[3] B. Hallgren et al., New developments in the time and pulse height digitizers, IEEE Trans. in Nucl. Science, NS–27, N°1, Feb. 1980.

[4] G. Goggi et al., A digital front-end electronics for calorimetry at LHC, CERN/DRDC/90–74, P19, Dec. 1990.

[5] J. Christiansen, Using digital signal processing to extract detector signal features, CERN/ECP–MIC internal note, Feb. 1993.

[6] V. Buzuloiu, Real time recovery of the amplitude and shift a pulse from its samples, CERN/LAA, Apr. 1992.

[7] Rumelhart et al., Parallel Distributed Processing, Vol.1, MIT Press, Cambridge MA, 1986.