**MEC3 — A PIPELINED ZERO SUPPRESSION AND TRIGGER
MATCHING CHIP**

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

J. Christiansen
CERN, Geneva, Switzerland.

Abstract

The MEC3 chip is a demonstrator of the general purpose MEC architecture. This architecture is intended for the digital front-end of detector channels where the detector signal is sampled at a constant rate. In addition to simple storage during the first-level trigger latency, data of interest are extracted by zero suppression and trigger matching. An event synchronized read-out interface takes care of merging event data from several channels. The three main ports (1: sampled data in, 2: trigger and 3: read-out), can run completely asynchronously. The synchronization of the three ports inside the chip is performed at the event level by the use of time tags and FIFOs.

Zero suppression is performed by adaptive thresholding that takes baseline variations into account. In addition a programmable FIR filter is available to process the signal before the pulse detection thresholding. Trigger matching is done by a comparison between time tags of the extracted pulses and the trigger decision.

All functions are implemented with a high level of programmability to accommodate different signal characteristics. Special handling of channel pile-up and clustering has also been included. Extensive simulations at behavioural level have been performed to optimize the architecture and an ASIC has finally been implemented in standard cells in a 1.0 μ m CMOS process.

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

1 INTRODUCTION

In future High Energy Physics experiments, data from millions of detector channels must be sampled at high rates (40 MHz for the LHC — Large Hadron Collider). The large quantity of data generated must be filtered in order to reduce it to a manageable amount that can be stored for off-line analysis. This data reduction should be performed as early as possible to reduce buffering requirements, power consumption and transmission bandwidth in the front-end electronics and the data acquisition system [1]. Filtering can be done at the global level using a trigger decision to reject events not of interest, and on the channel level by suppression of samples of no interest (zero suppression). After trigger filtering and zero suppression the data rate is so low that channel multiplexing and data processing can be performed on-line [2].

A pipelined integrated circuit performing such data extraction and first-level trigger buffering functions for calorimetry has been designed as a collaboration between CERN and LIP. The architecture has been extensively simulated and finally mapped into an ASIC.

This chip, which is a part of the MEC architecture [3], is intended to be used after the detector signal has been amplified, shaped and finally sampled by an ADC. At the output of the chip, data are presented in an event-synchronized manner, such that several MEC3s can perform local event-building into a shared second-level buffer.

2 ARCHITECTURE

Signal pulses originating from particles passing a detector element are identified and extracted by a zero suppression unit. For each pulse a set of consecutive samples is transferred into the first level event buffer. A set of flags is associated with each pulse at this level to tag double pulse and pileup conditions. A time tag is written into the buffer together with the pulse data to keep track of its time position in the original continuous stream of samples from the ADC.

Fig. 1 MEC architecture.

Extracted pulses are stored in the event buffer until the corresponding trigger information is available. Only non-zero data are stored in the buffer, enabling a small buffer size to contain the required data during the first level-trigger latency.

Trigger decisions are received as time tags of positive triggers. The trigger time tags are stored in a small buffer, if the trigger matching unit is still busy processing previous triggers.

The selection of pulses in the event buffer belonging to a trigger is based on their time tags. Pulses belonging to a trigger are written into a read-out buffer as an event fragment together with an event number. In case the read-out buffer is full the trigger matching process stops and the event buffer and trigger buffer must keep their information for an extended period.

Data are taken from the read-out buffer when an external read out controller requests them. Data are read out at the event level by an event-synchronized protocol enabling event fragments from several MEC3s to be built into a local event data structure.

The three main ports of the chip (input, trigger and output) can run complete asynchronously. The synchronization between them is achieved through the use of time tags and FIFOs.

Pipelines are used extensively in order to achieve the high throughput needed in the zero suppression unit running at full sampling speed [4].

1.2 Zero suppression

The basic pulse detection scheme is simple thresholding.

Two complementary schemes have been included to improve the pulse detection efficiency: adaptive thresholding continuously derives the signal baseline and adjusts the threshold accordingly, and a simplified FIR filter enhances the signal characteristics before performing thresholding.

Fig. 2 Zero suppression.

The signal baseline is extracted with a moving average of 2^N samples (where N can be chosen according to the dynamics of the baseline shifts), excluding periods where a pulse is detected. Extensive simulations have shown this simple algorithm to be very efficient for applications where the baseline variations are of relatively low frequency (Fig. 3).

Dual level thresholding is available to perform an efficient detection of bipolar- and tripolar-shaped detector signals.

Fig. 3 Adaptive thresholding using extracted baseline.

The FIR filter spans the complete pulse duration (set to eight samples in the MEC3 implementation). Since the filter is not in the normal signal path of data stored in the event buffer, it has been simplified to only five bit coefficients without a noticeable degradation of the pulse detection efficiency. The filter coefficients are programmable so they can be optimized to a specific application. A deconvolution can possibly be performed to identify individual pulses (Fig. 4) in high rate experiments where pileup may be a serious constraint [5].

Fig. 4 Deconvolution shows two pulses where simple thresholding would see only one.

Special circuitry can discard pulses that are shorter than a specified number of samples (noise spikes).

A programmable number of pre-samples before the detection point of the pulse is supported. This enables the pulse amplitude to be extracted at a later stage with high precision taking baseline variations into account.

A channel with below-threshold data may be forced to store its current samples if one of its neighbouring channels has detected a pulse above threshold. This enables the halo of a cluster to be retained so that the total energy in the cluster can then be extracted with high precision. The halo detection is accomplished with the minimum possible connectivity between channels (Fig. 5). Different skews on the path of the halo signal between the neighbouring channels can be compensated for internally in the chip. A channel forced by the halo signal to accept its data tags it with a ‘halo flag’. In detectors with high event rates, pileup between successive pulses occurs frequently. In such a situation, the knowledge of the detectors' pulse shape can be used to compensate for this effect. Both the pulse that suffers from pileup and that which causes it should be stored, bound together by a set of flags.

Fig. 5 Halo connectivity.

The zero suppression circuitry may tag pileup data in two different ways:

If a pulse is detected within a programmable time window from the previous, then the pulse is stored with a ‘pileup flag’ appended to it (Fig. 6).

Fig. 6 Pileup detection with pileup window.

If two pulses are so close together that they are superimposed, the circuitry will store them as two consecutive pulses, adding a ‘double pulse flag’ to the last one. This signals that the complete description of the pulses is spread throughout these two groups of eight samples.

A pulse identified by the zero suppression is stored as a single very wide word in an elastic event buffer together with the time tag, the extracted baseline and the corresponding flags (Fig. 2). Data is queued here until a first-level trigger decision is received.

2.2 Trigger matching

After zero suppression, pulses are treated as a single wide data word at 1/8 of the sampling frequency, resulting in a reduced power consumption.

Trigger information is received serially as a trigger time tag. A derandomizing trigger buffer is used to accommodate the variable rate of the trigger. The trigger time tag is compared to the time tags of the pulses in the event buffer (Fig. 7) in order to select the interesting pulses to be passed to the read-out buffer. A rejected pulse is temporarily stored in the ‘previous pulse register’.

Fig. 7 Trigger matching.

After the comparison of time tags, the trigger-matching circuitry will check for pulses that have been marked as pileup or double pulses in the zero suppression unit. If the pulse being matched has a ‘double pulse flag’ or a ‘pileup flag’, then the previous pulse is passed together with it. The next pulse is also checked for ‘double pulse flag’; if found, this pulse is also passed to the read-out buffer. All these pulses will have the same trigger time tag attached to them.

If a ‘pileup flag’ is found on the matched event, the trigger matching will store a ‘pileup data flag’ together with the pulse that caused the pileup. The present pulse which in this case has suffered from pileup will be stored with a ‘pileup effect flag’ attached to it.

The trigger matching in this way accepts and passes all data necessary to fully characterize the pulse that was matched to a trigger.

Some uncertainty in the detection time of the pulse (caused by variable delays in the path from the detector to MEC3 or by the different amplitudes of the pulses) is accommodated by the introduction of a programmable trigger window. This allows for the acceptance of pulses with a time tag within the matching window.

Finally, an event number needed by the event-synchronized read-out is appended to the pulses. The event number is generated by a simple counter incremented with the processing of

every new trigger. The event number tag is the same for all pulses that have been accepted by the same trigger.

A pulse accepted by the trigger matching is stored in the read-out buffer with the following data: data samples, time tag, baseline level, pileup effect flag, pileup data flag, double pulse flag, halo flag, trigger time tag and event number.

When there are no trigger tags available in the trigger buffer, data older than a defined maximum latency of the first level trigger decision can be automatically discarded. This prevents the event buffer from running full.

2.3 Event synchronized read-out

After trigger matching the data rate is so low that data from several channels (MEC3 outputs) can be multiplexed into a single data stream. This is done in an event synchronized order, such that data belonging to the same trigger are read out as one block of data (local event building). A channel ID is also read out to be able to identify the channel to which the event data belongs.

The event synchronization is done via a local read-out event number counter that is incremented by the external read-out controller (thus having the same value for every channel connected to it). If any channel has data with the same event number tag, it will signal the read-out controller of its existence, asking to be read out. The read-out controller scans all channels for data belonging to the same event, and sequentially reads every one that has data.

If many channels are multiplexed into the same bus this arbitration scheme becomes non-trivial. Thus an easy to use token ring read-out protocol that does not require complicated external arbitration logic is supported (Fig. 8). In this scheme, a token is passed around by the read out controller. If a particular MEC3 has data for the present event, it will hold the token while being read out. After all data has been sent out, the token is passed to the next channel in the chain.

Fig. 8 Event synchronized read-out.

Channels without information for this event just pass the token to the next, without further action. Only when it reaches the read-out controller has the complete event been read out and can a new event cycle begin.

If some of the control words introduced by the MEC3 are not desired, their read-out can be disabled individually.

2.4 Channel calibration

In order to fully characterize and monitor each detector channel some testing modes have been introduced. In the flash mode the chip will, from a pre-set time, capture 128 consecutive samples, thus allowing for the study of the channel background. In the single-shot mode the MEC3 will store 128 consecutive samples when a pulse has been detected, thereby showing the complete response of the detector/shaper/ADC chain.

3 DEMONSTRATOR CHIP

The implementation of the MEC architecture described in this paper is intended as a demonstrator of the concepts used in the architecture. In a real experiment parts of its functionality may not be needed, or additional specialized circuitry may need to be added. All functions described can optionally be disabled in the demonstrator chip, thus giving a very high level of flexibility in operation.

The demonstrator chip features a wide data path of 12 bits and has some redundant functions that are introduced for the completeness of the demonstration. This has resulted in a fairly large silicon usage, but the 40 MHz clock rate is still attainable.

The event buffer, in this implementation, was designed to store up to eight pulses, which is enough buffering for the first-level trigger latency in a typical experiment.

The demonstrator chip has been fully simulated in Verilog HDL and has been completely mapped into schematics using a 1.0 μm CMOS standard cell library from ES2. Placement and routing has also been performed, resulting in a silicon area of 55 mm^2 . Prototypes are expected by the end of 1994.

4 FUTURE DEVELOPMENTS

After complete testing of the prototypes, we aim to test the MEC3 chip against real data from a calorimeter, in order to prove the functionality introduced by this architecture.

In the future, with a better knowledge of the needs of a particular experiment, a new chip can be designed. This can be specifically tailored for the application taking into account issues such as power consumption and radiation hardness.

Several channels can be merged into a single chip, depending on the functionality required, further reducing the off-chip communication.

Higher clock speeds can be obtained simply by the redesign of some critical blocks and by introducing more pipelining into the signal paths of the zero suppression.

REFERENCES

- [1] F. Anghinolfi, Front-End Electronics for Future High-Energy Physics Experiments, CERN/ECP 91–11.
- [2] P. Gomes et al., NANA – An integrated signal processor and record builder for the level-2 readout of asynchronous event-filtering digital pipelines, presented at Nuclear Science Symposium 1994, Norfolk, Virginia, also available as CERN–ECP/94–21.
- [3] J. Christiansen et al., A Micro-Pipelined Zero Suppression, Trigger Matching and Recalibration Integrated Circuit, in Nuclear Science Symposium, Orlando, Florida, October 1992, pp. 477–479.
- [4] J. Christiansen, The MEC3 demonstrator chip, NEBULAS note 92/11, CERN/ECP.
- [5] S. Gadomski et al., The deconvolution method of fast pulse shaping at hadron colliders, CERN–PPE/92–24, 31/1/1992.