Master's Thesis
Physics

Digital Signal Processing for Particle Detectors in Front-End Electronics

Tiina Naaranoja
(2014)

Supervisor: Prof. Risto Orava
Ph.D. Paul Aspell
Reviewers: Prof. Risto Orava
Univ. Lect. Kenneth Österberg

University of Helsinki
Department of Physics

PL 64 (Gustaf Hällströmin katu 2)
00014 Helsingin yliopisto Finland

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty Faculty of Science	Laitos — Institution — Department Department of Physics

Tekijä — Författare — Author

Tiina Naaranoja

Työn nimi — Arbetets titel — Title

Digital Signal Processing for Particle Detectors in Front-End Electronics

Oppiaine — Läroämne — Subject

Physics

Työn laji — Arbetets art — Level Master's Thesis	Aika — Datum — Month and year September 2014	Sivumäärä — Sidoantal — Number of pages 69

Tiivistelmä — Referat — Abstract

The Large Hadron Collider (LHC) at CERN is currently being started up after a long shutdown. Another similar maintenance and upgrade period is due to take place in a few years. The luminosity and maximum beam energy will be increased after the shutdowns. Many upgrade projects stem from the increased demands from the changed environment and the opportunity of installation work during the shutdowns. The CMS GEM collaboration proposes to upgrade the muon system in CMS experiment by adding Gaseous Electron Multiplier (GEM) chambers.

The new GEM-detectors need new Front-End electronics. There are two parallel development branches for mixed-signal ASICs; one comes with analog signal processing (VFAT3-chip) and another with analog and digital signal processing (GdSP-chip). This Thesis covers the development of the digital signal processing for the GdSP-chip. The design is described on algorithm level and with block diagrams.

The signal originating in the triple GEM-detector sets special challenges on the signal processing. The time constant in the analog shaper is programmable due to irregularities in the GEM-signal. This in turn poses challenges for the digital signal processing. The pulse peaking time and signal bandwidth depend on the choice made for the time constant.

The basic signal processing techniques and needs are common for many detectors. Most of the digital signal processing has shared requirements with an existing, well-tested Front-End chip. Time pick-off and trigger production was not included in these shared tasks. Several time pick-off methods were considered and compared with simulations. The simulations were performed first using Simulink running on Matlab and then on Cadence tools using Verilog hardware description language.

Time resolution is an important attribute determined jointly by the detector and the signal processing. It is related to the probability to associate the measured pulse with the correct event. The effect of the different time pick-off methods on time resolution was compared with simulations. Only the most promising designs were developed further. Constant Fraction Discriminator and Pulse Recognition, the two most promising algorithms, were compared against analog Constant Fraction Discriminator and Time over Threshold time pick-off methods. The time resolutions obtained with noiseless signal were found to be comparable. At least in gas detector applications digital signal processing should not be ruled out of fear for deteriorated time resolution.

The proposed digital signal processing chain for GdSP includes Baseline Correction, Digital Shaper, Integrator, Zero Suppression and Bunch Crossing Identification. The Baseline Correction includes options for using fixed baseline removal and moving average filter. In addition it contains a small memory, which can be used as test signal input or as look-up-table et cetera. Pole-zero cancellation is proposed for digital shaping. The integrator filters high frequency noise. The Constant Fraction Discriminator was found optimal for Bunch Crossing Identification.

Avainsanat — Nyckelord — Keywords

Digital Signal Processing, DSP, Electronics, ASIC, Particle Detectors, CERN, LHC, CMS, GEM

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1. Introduction

1.1. GEMs for CMS

The Compact Muon Solenoid (CMS) is one of the four large experiments at the Large Hadron Collider (LHC) at CERN (Conseil Européen pour la Recherche Nucléaire). It is a general-purpose detector designed to cover as wide part of the physics at the LHC as possible. Its achievements so far include the participation in the discovery of the Higgs boson. The future holds studies on the discovered Higgs boson, searches for supersymmetry and extra dimensions. The detector consists of onion-like layers of different detector types: innermost the inner tracker consisting of a pixel detector and a silicon tracker, the electromagnetic calorimeter, the hadronic calorimeter, the superconducting solenoid yielding a 4 Tesla magnetic field and outermost the muon system. The layout is shown in figure 1.1. Geometrically the detector is divided into the cylindrical barrel, which makes up the bulk, and the planar endcaps at the ends of the cylinder. [1]

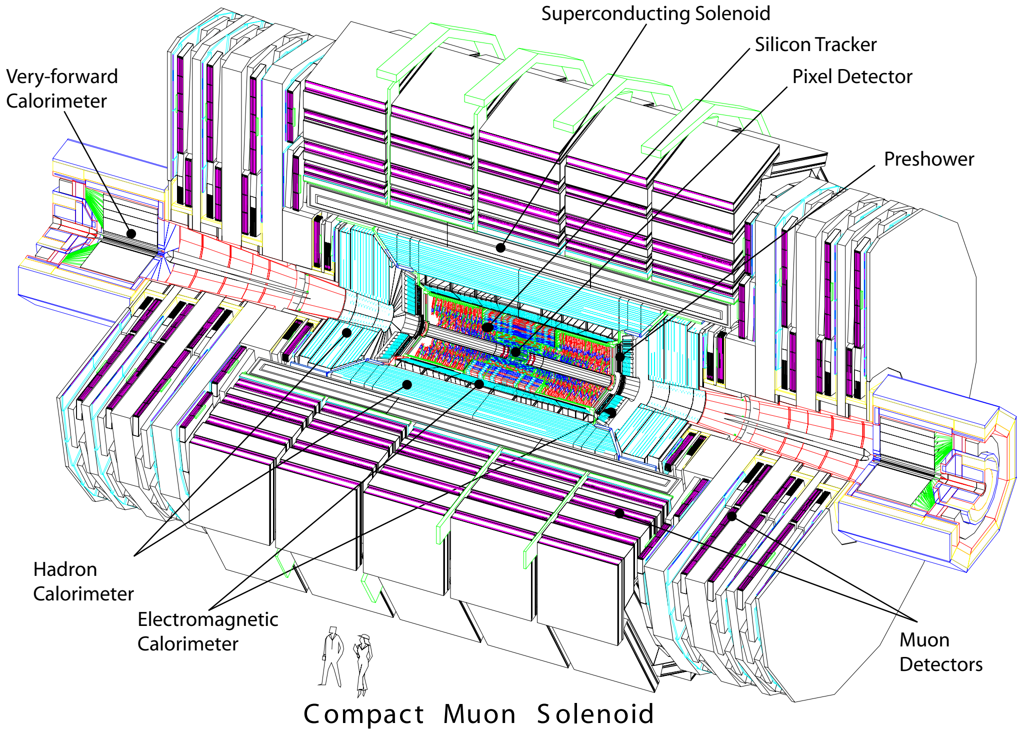


Figure 1.1.: A perspective view of the CMS detector (Image source [1])

The muon system is a tracker that is used for muon identification, momentum measurement and triggering. It includes Cathode Strip Chambers (CSCs), Drift Tubes (DTs) and Resistive Plate Chambers (RPCs). The DTs are used in the barrel and CSCs in the endcaps. They both have good position resolution and good background rejection. The RPCs form a independent triggering system. They have the good temporal resolution that is needed for correct bunch crossing time assignment, but coarser position resolution. [1]

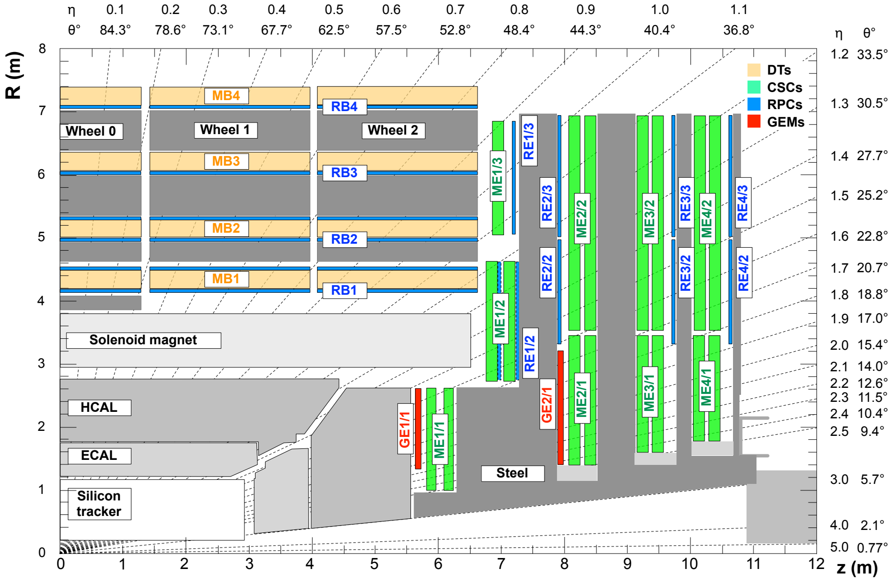


Figure 1.2.: Transverse section of the CMS detector. The proposed GEM detectors are indicated with red boxes on the endcap.(Image source [2])

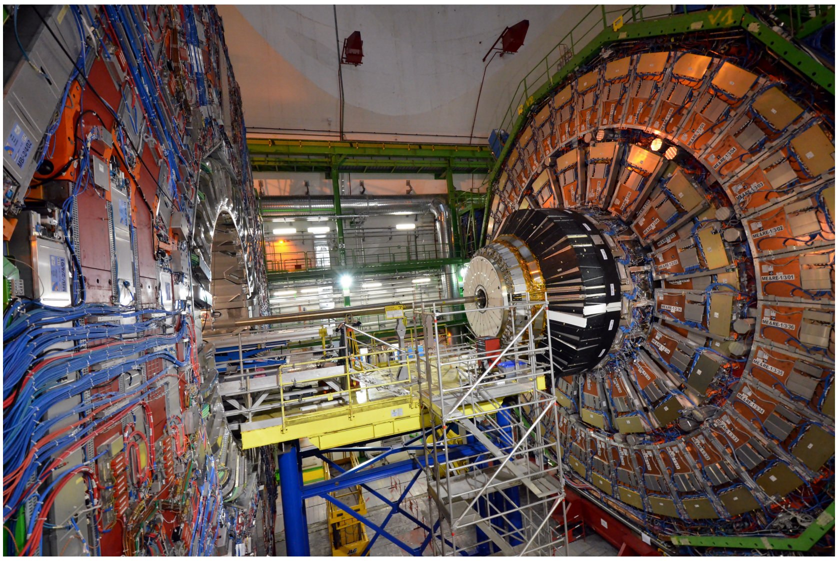


Figure 1.3.: CMS detector when opened for maintenance. Barrel is on left and endcap on right.

The CMS GEM Collaboration proposes to install Gas Electron Multiplier (GEM) chambers in the forward region of the CMS experiment endcaps (see figure 1.2). They would extend the CMS muon system that now has lowered redundancy in the forward region (pseudorapidity $1.6 < |\eta| < 2.4$). Currently the region has only CSCs installed. GEMs are an attractive option, because they have large gain (exceeding 10^4), good spatial resolution, good time resolution and they can cope with the anticipated high rates. First GEMs would be installed to the region labeled GE1/1 during second LHC Long Shutdown in 2017-2018. The region labeled GE2/1 would be equipped during third LHC Long Shutdown scheduled beyond 2020. [2]

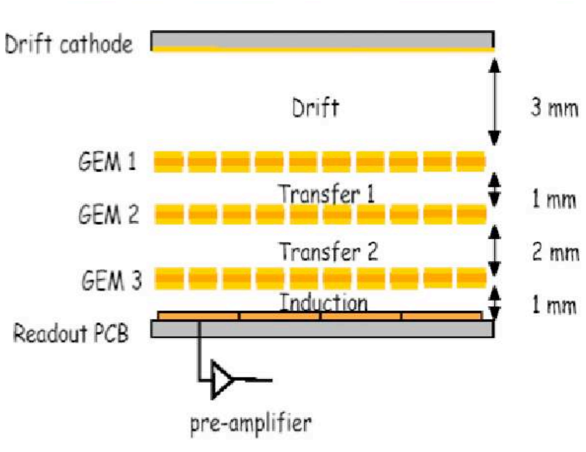


Figure 1.4.: Triple GEM with CMS gap configuration. (Image source [3])

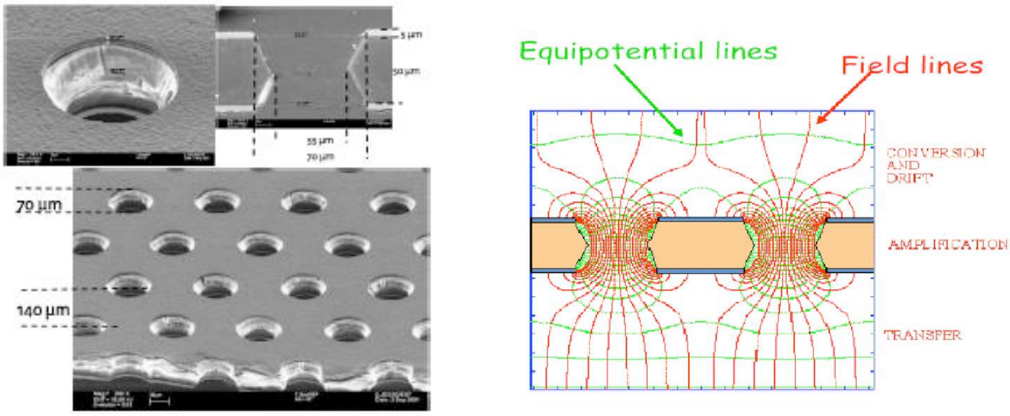


Figure 1.5.: GEM foil. Electron microscope image of GEM foil (left, image source [5]) and field inside the holes (right, image source [3]) .

GEMs are gaseous micro-pattern detectors first introduced in 1996 [4]. The primary ionization occurs in drift region, which is effectively a proportional counter. What sets the detector apart from normal proportional counter is the GEM foil shown in Fig. 1.5. The surfaces of the foil are of a conducting material with insulating layer between them. Into the foil are etched micrometer scale holes. There is large difference in the electric potential between the surfaces of the foil . This results in a high field inside the holes as seen in Fig. 1.5. When electrons from the primary ionization enter the holes they are accelerated enough to cause secondary ionization. Triple-GEMs have three layers of GEM foils. After the electrons are multiplied in GEM foils they are collected on electrodes (strips) and the signal is amplified and processed in the front-end electronics. (Eg. [6])

At present there are triple-GEM chambers installed in Common Muon and Proton Apparatus for Structure and Spectroscopy (COMPASS) [7], Total Cross Section, Elastic Scattering and Diffraction Dissociation (TOTEM) [8] and Large Hadron Collider beauty (LHCb) [9, 10] experiments. COMPASS has 10 years of experience with the GEMs. COMPASS and TOTEM GEMs use Ar/CO_2 gas mixture with proportions 70:30 and LHCb GEMs

$Ar/CO_2/CF_4$ mixture. Tests have confirmed that the faster gas mixture $Ar/CO_2/CF_4$ with proportions 45:15:40 has also better time resolution[11].

Triple-GEM prototypes both small (10×10 cm^2) for initial testing and two full scale superchamber have been built and tested in CERN test beam facilities. Both analog (APV25-chip) and digital (VFAT2) front-ends were used. With analog readout a spatial resolution below 110 µm and a time resolution of 4 ns were achieved. The digital readout increased the spatial resolution to 270 µm. [2]

1.2. GEM-electronics

A detector such as GEM needs a lot of electronics for signal read-out and triggering. For some parts existing electronics can be used, some need to be custom designed and for some there are ongoing generic development projects. Mostly the envisaged electronics falls to the two latter categories.

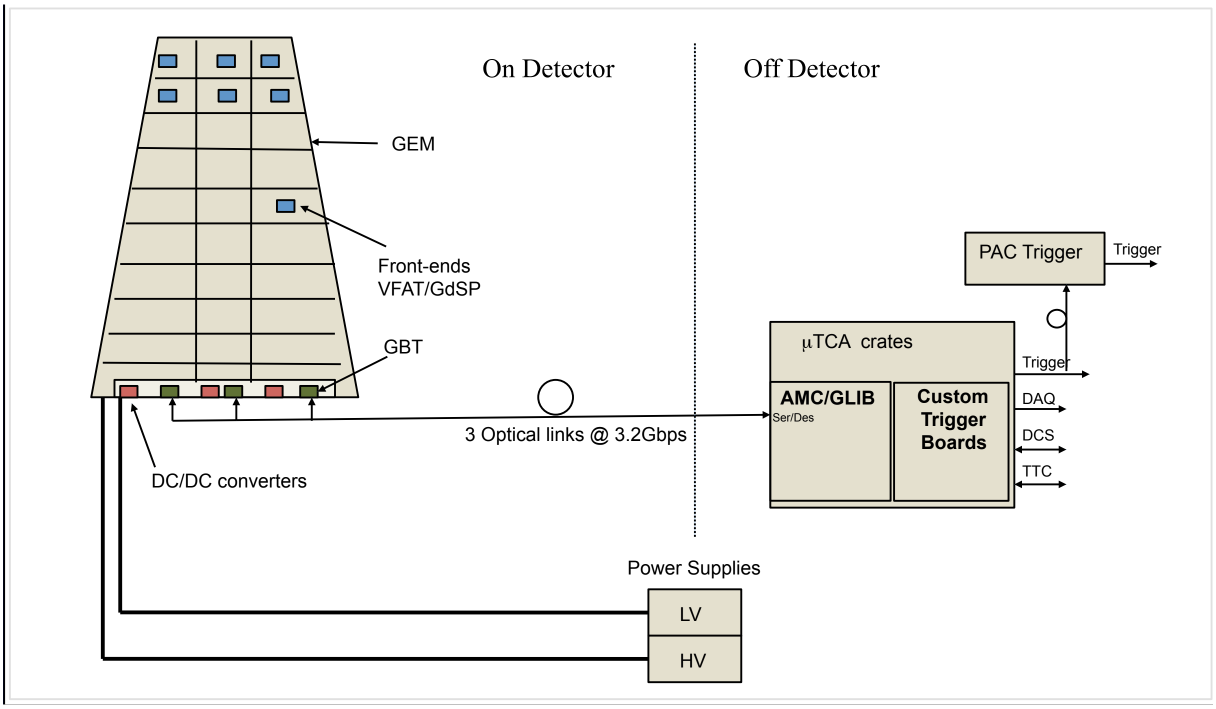


Figure 1.6.: Diagram of proposed GEM electronics.(Image source [27])

The overall design of the GEM electronics is shown in Fig. 1.6. The GEM superchamber is divided into segments that each work as an independent detector. Each segment has 128 strips and one front-end (FE) Application Specific Integrated Circuit (ASIC). One surface of the GEM superchamber is a Printed Circuit Board (PCB). It acts at the same time as electronics board and closes the gas volume. On the detector side surface the innermost layer is dedicated to the electrode strips. The outer layers are dedicated to Low-Voltage Differential Signaling (LVDS) lines for communication with the FE ASICs, clock, ground, high and low voltage supply [12].

The electronic control and readout system has components both on detector and off detector in the counting room. The GigaBit Transceiver (GBT) [13] ASIC is a generic development project that is foreseen to control and readout FE ASICs on detector. The off detector system provides interface to the CMS Data Acquisition (DAQ), TTC and trigger systems. The

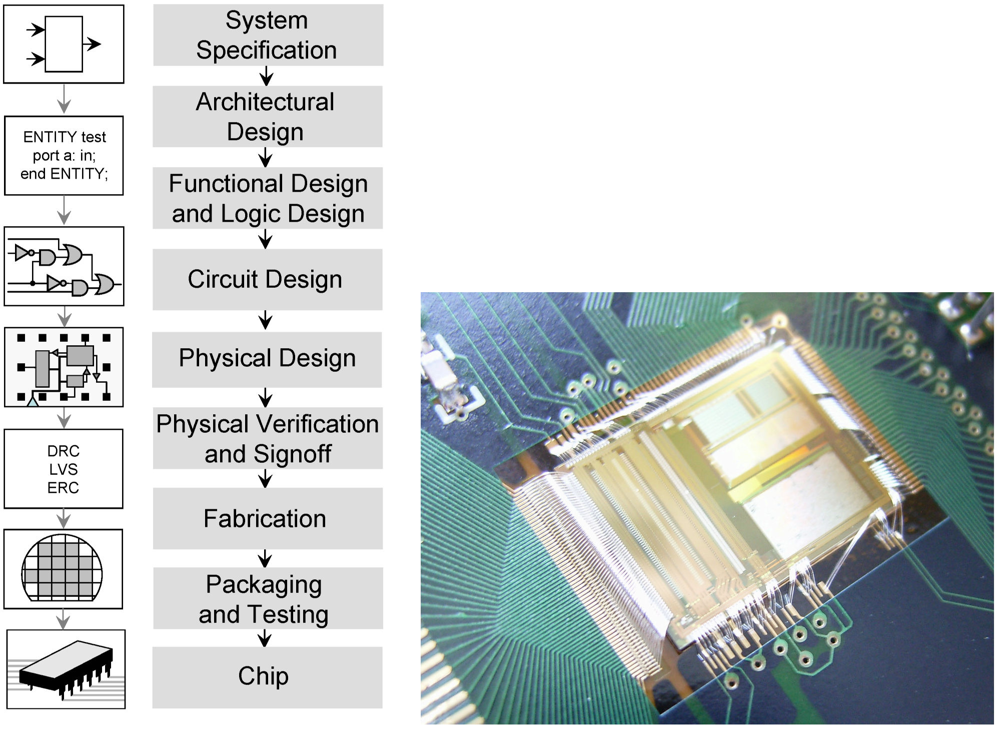


Figure 1.7.: Major steps of integrated circuit design flow (left, image source [15]) and an example of end product VFAT2 front-end ASIC (right, image source [17]).

design will be based on Advanced Mezzanine Cards (AMC) used with Micro Telecommunications Computing Architecture (µTCA) standard. It will contain Gigabit Link Interface Boards (GLIB) [14] from generic development project and custom Field Programmable Arrays (FPGA).

The FE ASICs will be custom designed. The design process usually takes some years, in this case it is foreseen to take over two years [11]. The main steps of the design flow are shown in Fig. 1.7. The design flow begins with system definition and requirements. Simultaneous to particle detector front-ends ASIC design the development of the detector itself is often ongoing. This leads often to changing requirements in the middle of the design process. The behavior and logic of the system is designed and tested with simulations. In the case of digital design the design is written in Hardware Description Language (HDL) such as Verilog. In HDL there are three different levels of abstraction: Behavioural level, Register Transfer Level (RTL) and Gate level. At behavioral level the model describes the algorithm, but has no regard for the structural realization. A RTL model is synthesizable. Being syntehsizable means that the logic can be expressed as a circuit consisting of discrete primitive operations. The RTL model uses outside clock and has well defined timing bounds. The gate level code is usually generated with synthesis tools from the RTL model. It has only logical values (0, 1, z, x) and uses only primitive operations (AND, OR etc) defined by the library that is being used. After more testing comes placing and routing. In the case of digital design it is done with automated tools. The gates are expressed as transistors on the semiconductor substrate. The routing via conducting lines is planned. Based on the created model masks can be created. Using the masks the chips are fabricated. First only few prototype chips are fabricated and tested. The behavior of the real chip might differ from the expected and in this case some level of redesigning is needed.

The COMPASS GEMs use APV25 (Analogue Pipeline Voltage mode) [16] front-end ASIC

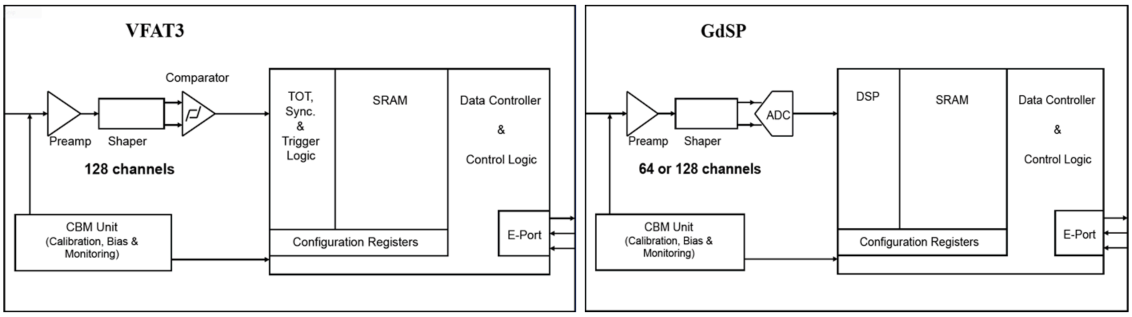

VFAT3
Comparator
Preamp
Shaper
128 channels
TOT, Sync. & Trigger Logic
SRAM
Data Controller & Control Logic
CBM Unit (Calibration, Bias & Monitoring)
Configuration Registers
E-Port
GdSP
ADC
Preamp
Shaper
64 or 128 channels
DSP
SRAM
Data Controller & Control Logic
CBM Unit (Calibration, Bias & Monitoring)
Configuration Registers
E-Port

Figure 1.8.: Block diagrams of VFAT3 and GdSP front-end ASICs (Credits Paul Aspell)

originally designed for the CMS tracker. It has 128 analog channels that are sampled and multiplexed for analog readout. In TOTEM the readout is based on VFAT2 (Very Forward Atlas and Totem) [17] front-end chip designed in CERN microelectronics group primarily for TOTEM. It has 128 analog input channels. The output is binary information from comparators. The LCHb experiment uses Carioca-GEM[19], which is an adaptation of CARIOCA (CERN and Rio Current-mode Amplifier) ASIC. It has 8 analog input channels and like VFAT2 binary outputs from a discriminator. Common task for all of the front-end ASICs is that they all have at least amplifier and shaper for all channels. From these the VFAT2 chip is closest to the CMS GEM requirements.

The CMS GEM front-end ASIC will be based on VFAT2. The requirements for trigger and tracking are similar to those of TOTEM, but there are some differences. The new chip will need to be compatible with different readout system using a different interface. The method for trigger production will be different. The trigger signal will be processed instead of direct readout of comparator signal. The charge collection time needs to be longer. Since the optimal collection time is unknown the pulse shaping time of the shaper will be programmable. There are two alternative options that are being investigated VFAT3 and GdSP (Gas detector digital Signal Processing). The block diagrams of both designs are shown in Fig. 1.8. The main difference between the VFAT3 and GdSP is that the latter will have Analog to Digital Converters (ADC) and additional digital signal processing (DSP). For optimal time resolution in VFAT3 digital Time over Threshold (ToT) technique will be used after the comparator for precise trigger timing. Also an alternative using analog Constant Fraction Discriminator (CFD) technique is being investigated [36]. The GdSP will use digital time pick-of method that is discussed in sections 3.4 and 5.5. Many parts of the DSP on GdSP are based on the DSP on S-Altro (Super-ALICE TPC Read Out) [20] chip, which in turn is based on ALTRO (ALICE TPC Read Out) [21] chip currently in use in A Large Ion Collider Experiments (ALICE) Time Projection Chamber (TPC) detector. The S-Altro was used as role model, source of inspiration and for some parts as cautionary example in the design process. Three of the five filters in GdSP DSP are based on S-Altro. The baseline correction filter was largely changed, the pole-zero filter was scaled down and zero suppression was used unchanged. The advantage of S-Altro blocks is that they have been tested in use. Many of the filters are unchanged from ALTRO chip, which is in use in ALICE. Valuable feedback on how they are working in practice was received. The digital shaper that was changed since ALTRO has been tested with 16 channel S-Altro demonstrator chip [20].

This thesis work concerns the design of the digital signal processing part for GdSP chip. The focus is on the algorithm level and conceptual design. New designs and ideas were initially tested using Simulink [24] that builds upon Matlab [23]. The benefit was fast design process and easy testing. Based on the Simulink designs models were written in Verilog language [25]. The Simulink translator was not used since automatically generated code tends to be non-optimized and difficult to read and modify. The Verilog models were designed and tested using Cadence tools [38].

1.3. Motivations for Digital Signal Processing

None of the preceding GEM front-end ASICs have ADCs and DSP. The main reason for it has been the power consumption of ADCs. If there is to be an ADC on all 128 channels the power consumption of them needs to be very low. Only recently the development of low power ADCs has reached the limit where it is feasible to consider including them. The Successive Approximation Register (SAR) ADCs [22] currently being developed in AGH University are envisioned for GdSP. They have very low $1\,mW$ power consumption per channel at 40 MHz sampling rate. The development work is still ongoing, but promising.

The benefits of DSP include additional processing which leads to cleaner trigger. More of the background artefacts can be distinguished from the signal. An important aspect is reduction of read out data. With the luminosity increase in LHC upgrade, overflowing buffers in front-end ASICs are threatening to become a very real problem. Important attributes, such as timing or center of charge cluster, can be extracted from the signal. The data is reduced when only the necessary attributes are read out. The digital processing is also more flexible than analog and the filters are relatively easy to make programmable.

As discussed the position resolution was seen to be worse with digital readout in GEM tests. This could be remedied with cluster processing. The center of gravity of the clusters could be either evaluated on chip for coarser position resolution or off-chip for finer resolution. If the clusters are to be processed off chip the pulse amplitude is needed and needs to be read out. Alternatively the data could be reduced by only reading out the central channel of a cluster. Unfortunately one single algorithm cannot satisfy all of these desires simultaneously. In GdSP the possibility where the user could choose between pulse amplitude readout and binary readout was pursued. There is no certainty that cluster processing will be needed. If the GEMs are to work in the same fashion as RPCs in the muon system, very fine position resolution is not needed.

2. Review on Detector Signal Processing

The amount of literature on signal processing for particle detectors is very limited. Some radiation detection textbooks such as Knoll [6] contain a chapter or two on signal processing. The abundant literature on digital signal processing covers only part of the signal processing relevant for particle detectors. The whole picture is obtained by combining the information from signal processing specific and radiation detection specific literature.

The signal processing chain for particle detectors begins in the detector and ends with formatted data. The main objective of the signal processing is to remove irrelevant information (noise, baseline fluctuations etc.) from the signal and extract the wanted information from it. Each step in the chain depends on the previous step and often also on the following steps. The signal processing is dependent on the detector type and what is foreseen to be done with it. The properties of the detector, such as charge collection time, determine what kind of time constant is used in the pre-amplifier. This in turn has large implications on the following signal processing. The usage, for which the detector is built, determines what kind of attributes of the signal are needed. In trackers the knowledge that there was a pulse is all that is needed in the end. For other applications (calorimeters for example) the pulse height is very important attribute. The importance of timing varies with application as well.

2.1. Signal generation and amplification

The signal begins in the detector. In gas detectors the particle being detected causes ionization in the detector. The electrodes start collecting charge already when the electrons are still on their way and have not reached the electrode yet. The moving charge induces charge on the electrode. The full charge is collected when the electrons reach the electrode.

The Shockley-Ramo theorem from the late 30's predicts the form for the induced signal. It uses the concept of weighting field $\vec{E_w}$, which is the electric field inside the detector when one electrode is connected to unity potential (ie. 1 Volt). It is a calculated quantity and not a real field. The induced current and charge are given by

$$i = -q\vec{E_w} \cdot \vec{v} \tag{2.1}$$

and

$$Q = q \int \vec{E_w} dx \tag{2.2}$$

respectively, where $\vec{v}$ is the velocity of the moving charge.[26]

In GEMs the moving charge is screened from the electrodes by the GEM foils everywhere but in the collection region shown in Fig. 2.1. When a charged particle enters the GEM it ionizes the gas in the drift zone. The electrons and ions are distributed along the track of the particle. The ions drift towards the drift-cathode and are not used in measurement. The primary electrons drift towards the first GEM foil. When they reach the foil they are distributed in time by the drift time across the drift region. In the GEM foil holes, the local strong electric field accelerates the electrons, which acquire enough energy to ionize the gas. The formed secondary electrons join the primary electrons. The electrons are multiplied by

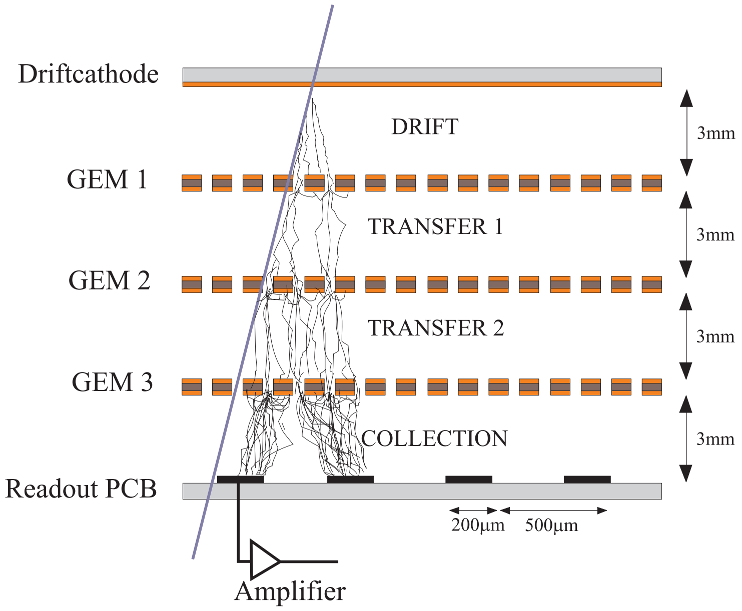


Figure 2.1.: Schematic cross-section of triple GEM detector. The dimensions do not correspond to CMS GEMs, but the concept is same. (Image source [10])

factor around 20 [27]. This is repeated at each GEM foil. The total charge collected is the sum of all these electrons.

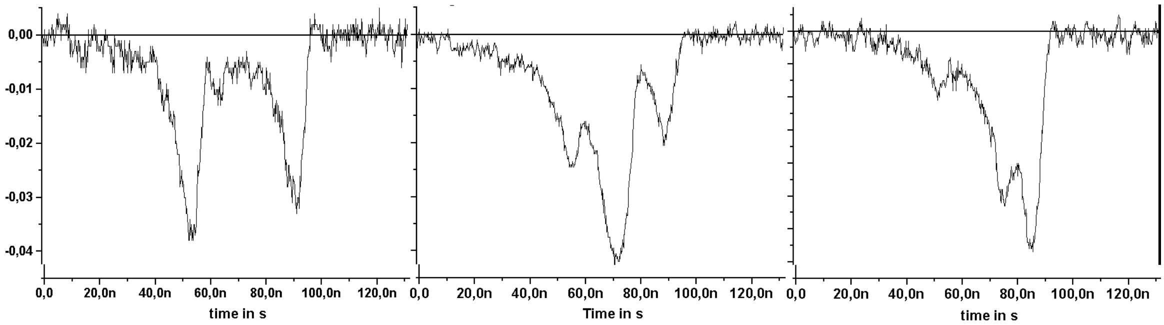


Figure 2.2.: Triple GEM signal with 2 ns shaping time. On y-axis the signal amplitude is given in arbitrary units. (Image source [10])

Three signals measured from MIPs in triple GEM by Ziegler et al. is shown in Fig. 2.2. As seen from the figure the signal from triple GEM is very irregular. The total duration of the signal is consistent with charge collection time that is obtained by summing the drift times in drift and collection regions [10]. The irregularities in the signal are thought to originate from clusters created during the primary electron formation [10]. The irregularities of the signal need to be addressed in the amplifier.

The amplification is usually performed in stages shown in Fig. 2.3 a). In some applications the first stage, preamplifier is the only necessary stage. The last stage, shaper (e.g. pole-zero cancellation) is optional. Often an RC-circuit is used to discharge controllably the accumulated charge from the amplifier as illustrated in Fig. 2.3 b). Without it the output would be a step function with height corresponding to the total charge collected. The

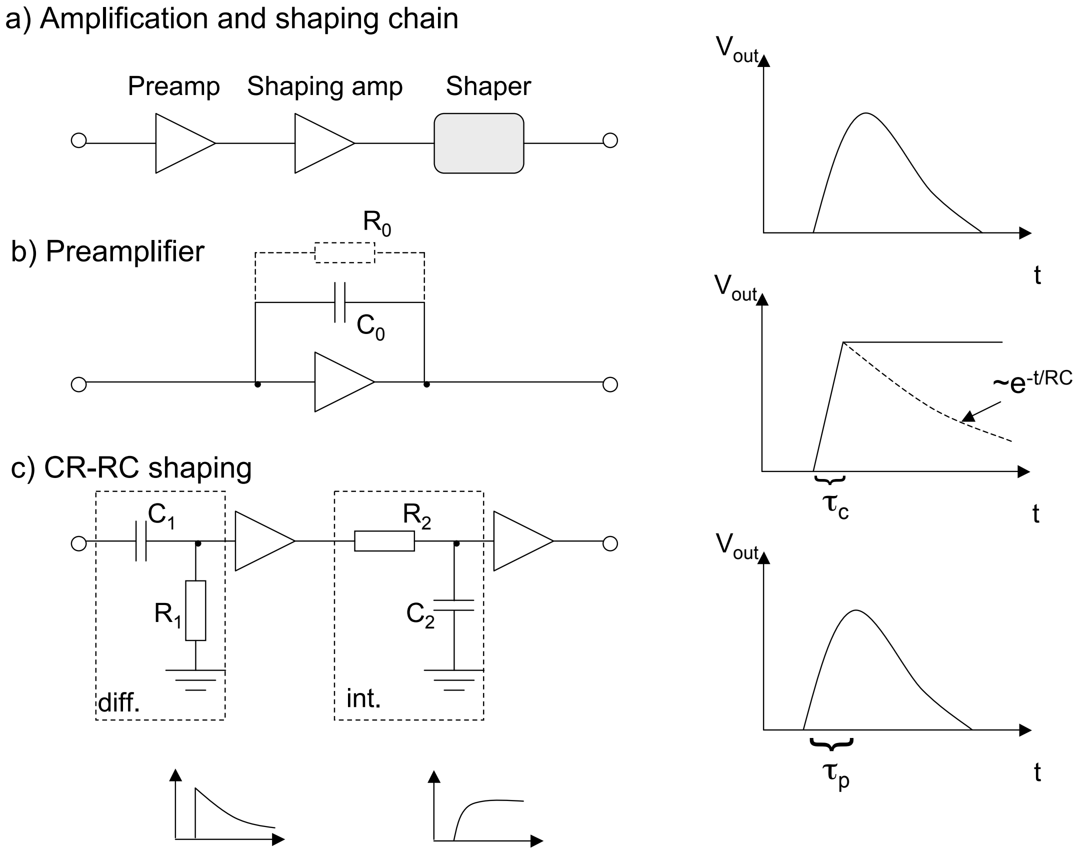


Figure 2.3.: Simplified diagrams of the signal amplification and shaping with their output.

preamplifier is usually followed by shaping amplifier. Fig. 2.3 c) shows an example of shaping amplifier; CR-RC shaper with amplification. Additional shaping, for example pole-zero cancellation, is sometimes needed.

After the preamplifier the pulse has a rise time that equals the charge collection time in the detector τ_c. If the preamplifier is accompanied by the resistor R_0, the charge is discharged with decay constant $\tau_0 = R_0C_0$. The decay constant is independent of the detector capacitance C_i, if the amplification is

$$A \gg \frac{(C_i + C_0)}{C_0}.$$

In the shaping amplifier the differentiator modifies the pulse decay. Differentiators impulse response is of the form

$$V \sim e^{-\frac{t}{\tau_1}},$$

where the decay constant is $\tau_1 = R_1C_1$. The integrator modifies the rise time of the pulse. It has the time constant $\tau_2 = R_2C_2$. The impulse response of the integrator is of the form

$$V \sim (1 - e^{-\frac{t}{\tau_2}}).$$

In the special case where $\tau_1 = \tau_2 = \tau$ the combined impulse response of the differentiator and integrator is given by

$$V = V_0 \frac{t}{\tau} e^{-\frac{t}{\tau}} \tag{2.3}$$

where V_0 is input pulse height. The pulse has a peak when $t = \tau$. In this case the peaking time τ_p and shaping time τ are equal. Later in this text they are used as synonyms. Equation 2.3 applies to the preamplifier output, if the collection time $\tau_c < \tau$. If this condition is not met, the pulse suffers from ballistic deficit, when all charge is not included in the integration. [6]

In GEMs the collection time depends on the fill gas and detector geometry. It is expected to be around 40 ns [27] for CMS GEMs. In addition to the ballistic deficit too short shaping time can lead to deformed pulse shape, because of the irregularities in the signal [28]. For these reasons the shaping time needed is foreseen to be at least 50 ns. This is twice the sampling period in the ADCs. The consequences of it are discussed in the following chapters.

2.2. Time resolution

In the context of detectors the time-resolution might not mean what one first thinks. It does not refer to the sampling frequency of the electronics. It's not necessarily even the precision of the measurement with respect to time. One way of thinking about the time resolution is to think of it as the variance in the peaking time in the signal. Another is to think it as the inherent uncertainty in the time measurement

The time t_0, when the particle enters the detector[1], is deduced from the time the pulse is detected. When these latencies are plotted, they are spread over time. The time resolution is defined as the standard deviation[2] of the latency. For continuous signal the time spectrum could take the form shown in picture 2.4.

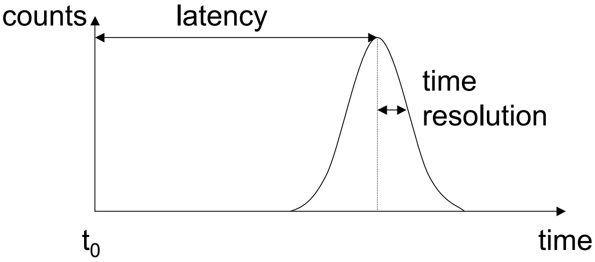


Figure 2.4.: Time resolution of continuous signal.

When the signal is discrete, the time resolution might not be calculated straightforward. Fig. 2.5 shows a histogram of the latencies, when the time resolution is much smaller than the sampling period. One clearly cannot simply take the standard deviation here. The Gaussian fit presented in the figure is obtained using error function. It is assumed that the that the temporal distribution can be approximated with normal distribution

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.4}$$

[1] In simulation t_0 is known. In measurements for example a faster detector can be used as reference or coincidence setup can be used.

[2] Sometimes full width at half maximum is used [6].

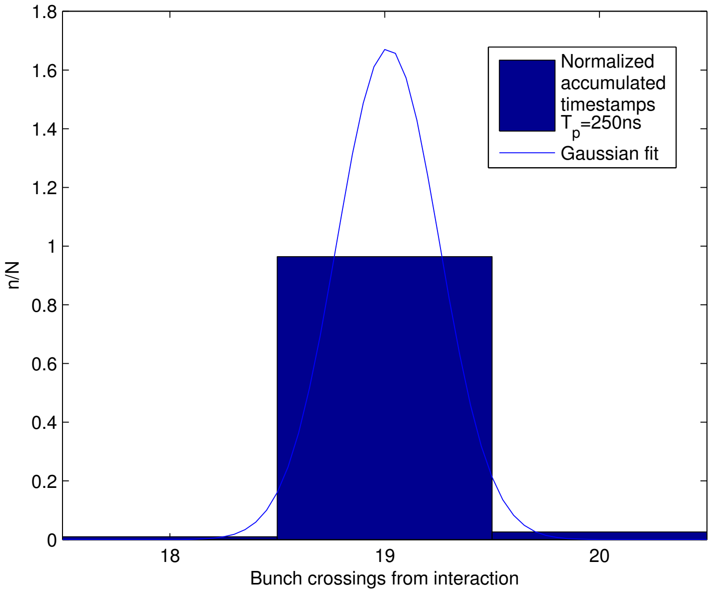


Figure 2.5.: Time resolution of discrete signal

where μ is the mean and σ the standard deviation. The error function

$$erf(x) = \frac{2}{\pi}\int_0^x e^{-t^2}dt \tag{2.5}$$

is closely related to the integral of normal distribution. The percentage of counts inside interval from $\mu - n\sigma$ to $\mu + n\sigma$ is given by $erf\left(n/\sqrt{2}\right)$. The timing efficiency η_t is defined as the percentage of counts in the right time bin. It can be tied to error function

$$\eta_t = erf\left(\frac{n}{\sqrt{2}}\right), \tag{2.6}$$

where the width of the time bin is $\Delta t = 2n\sigma$. Combining the equations and solving for standard deviation gives the time resolution

$$\sigma_t = \frac{1}{2}\frac{\Delta t}{\sqrt{2}\, erf^{-1}(\eta_t)} \tag{2.7}$$

2.3. Sampling

The required sampling frequency is usually given by Nyquist–Shannon theorem. If the signal frequency has a low limit $|f| > f_N$, the signal can be uniquely determined by the samples taken with frequency $f_{sampling} > 2f_N$ [29]. In particle detectors the signal is not periodic, but it is possible to take the Fourier transformation of it. For example signal shaped with CR-RC filter with single shaping time has a peak, when $t = \tau$. The dominant frequency of the rising edge $f_{signal} = 1/4\tau$ can be estimated with the dominant frequency in the Fourier transform of the signal. This is the frequency of a sine wave that is closest in form to the signal. The sampling is usually synchronous with the LHC-clock, which has a 40MHz frequency. This means, that the shaping time should be $\tau > 12.5$ ns. In GdSP the shaping time is programmable ranging from 25 ns to 200 ns and the sampling period is 25 ns.

Digitization process results in quantization noise. The digitized value is rounded and often slightly different from the real value. The error e range is

$$-{}^{\Delta}\!/_2 < e \leq {}^{\Delta}\!/_2$$

where Δ is the step size of the quantization. The noise created is random, white noise with no correlation with the input signal. The probability distribution is uniform over the quantization error range. The relation between the step Δ and signal can have great effect on overall signal to noise ratio. [29]

In real ADCs there is always some noise (mostly thermal "kTC-noise") and non-linearities (harmonic distortions). Effective number of bits (ENOB) is a measure of how close the ADC is to the ideal. ENOB is given by

$$ENOB = \frac{SINAD - 1.76dB}{6.02}, \tag{2.8}$$

where SINAD is SIgnal to Noise And Distortion ratio [30]. In simpler words ENOB is the number of bits in an ideal ADC reduced by the noise and distortions in the ADC.

2.4. Filters, Transfer Functions and z-transform

The calculations in signal processing can be carried out equivalently in time domain and frequency domain. With detector signal processing it is often more convenient to operate in time domain. For some filters like pole-zero cancellation filter frequency domain comes more naturally.

In time domain, a filters operation is characterized by impulse response $h\,[n]$. It gives the output $y\,[n]$ for a given input $x\,[n]$. The impulse response is obtained from the output, when the input is a delta function:

$$y\,[n] = h\,[n]\,x\,[n] = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} h_1 & 0 & 0 & 0 & \cdots \\ h_2 & h_1 & 0 & 0 & \cdots \\ h_3 & h_2 & h_1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \tag{2.9}$$

Transfer function is a generalization of the impulse response. It maps the filter input to the output in a specified domain for a linear time-invariant (LTI) system[3]. A discrete-time sequence (input, output, transfer function) can be represented in discrete frequency domain via z-transform. The variable z is defined as $z = e^{i\omega}$, where ω is the angular frequency and i is the imaginary unit. [30]

For example for discrete-time first order pole-zero filter discussed in the chapters 3.3.1 and 5.4 the output $y[n]$ is given by $y\,[n] = y\,[n-1] \cdot K - x\,[n-1] \cdot L + x\,[n]$, where $x[n]$ are the samples of the input signal and K and L constants having value between zero and one. The z-transformed transfer function is easy to calculate after the whole function is transformed using:

$$\begin{cases} x\,[n] \to X(z) \\ y\,[n] \to Y(z) \\ x\,[n-1] \to z^{-1}X(z) \\ y\,[n-1] \to z^{-1}Y(z) \end{cases}$$

[3] Filters in detector signal processing are in general linear time-invariant systems or at least aspire to be.

The z-transformed output is

$$Y(z) = z^{-1}Y(z) \cdot K - z^{-1}X(z) \cdot L + X(z).$$

After short rearranging the transfer function is found:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - L \cdot z^{-1}}{1 - K \cdot z^{-1}} = \frac{z - L}{z - K} \tag{2.10}$$

where $L \cdot z^{-1}$ is the zero and $K \cdot z^{-1}$ is the pole.

The frequency response of the filter has maximum at the pole $z = K$ and goes to zero at $z = L$. This is easy to see from equation 2.10. If $z = K$, the denominator goes to zero and the transfer function tends towards infinity. When in turn $z = L$, the numerator goes to zero as does the transfer function.

In general the step response of first order filter in discrete time domain with real pole K and real zero L has the form

$$h[n] = e^{-K \cdot n}\left(1 - \frac{L}{K}\right) + \frac{L}{K}. \tag{2.11}$$

This can be split into four cases:

$$h[n] = \begin{cases} 1 & L = K \\ e^{-K \cdot n} & L = 0 \\ a - b \cdot e^{-K \cdot n} & K > L, \text{ where 0<a<1 and b >0} \\ a + b \cdot e^{-K \cdot n} & K < L, \text{ where a>1 and 0<b<1} \end{cases} \tag{2.12}$$

, where $a = \frac{L}{K}$ and $b = \left|1 - \frac{L}{K}\right|$ are constants. When the poles and zeros equal, the filter passes all signals unchanged. In single pole case (zero in origo) and the filter “shaves off” an exponential from the signal tail. When zero is not in origo and pole is larger than the zero, in addition to canceling the exponential the signal is added with a constant value after pulse. This is useful for correction of undershoots. In the case, when the zero is larger than the pole, the leading edge of the pulse is softened and the peaking time increases. The filter is most effective in tail cancellation when the zero is at origo $L = 0$ and pole is as large as possible $K = 1$.

When several filters are in cascade each first order filter operates individually. This is the case in the exemplary filter. The total transfer function is a product of the first order filter transfer functions:

$$H(z) = \frac{1 - L_1 \cdot z^{-1}}{1 - K_1 \cdot z^{-1}} \cdot \frac{1 - L_2 \cdot z^{-1}}{1 - K_2 \cdot z^{-1}} \cdot \frac{1 - L_3 \cdot z^{-1}}{1 - K_3 \cdot z^{-1}} \cdot \frac{1 - L_4 \cdot z^{-1}}{1 - K_4 \cdot z^{-1}} \tag{2.13}$$

Similarly the total impulse response is simply product of first order impulse responses $h[n] = h_1[n] \cdot h_2[n] \cdot h_3[n] \cdot h_4[n]$.

It is worth noting that there are usually more than one possible designs with the same transfer function. The transfer function is not unique for certain design.

2.5. Noise Sources

The noise accompanying detector signal can originate in the detector itself, parasitic capacities between channels, preamplifier, analog shapers and digitization process. The digital signal processing contributes to the noise as well. An often used measure of noise is the signal to noise ratio (SNR). For systems, where absolute level of noise is known or calculated, but signal amplitude is unknown, the noise is often expressed as equivalent noise charge (ENC). It gives the equivalent noise to an ideal system at the system input. The unit is rms electrons. The SNR can be easily calculated from ENC when the detector response is known.

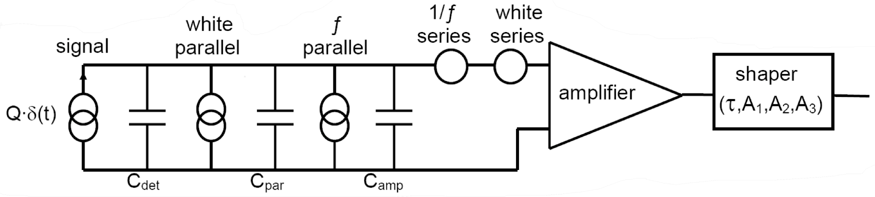


Figure 2.6.: Noise sources in charge amplifiers (Image source [26]).

For the noise sources from detector to analog shaper the concept of weighting function $w(t)$ is used in noise calculations. It is the mirror image of the impulse response of the amplifying and shaping circuit. The noise is expressed as ENC. The amplifier noise can be modeled using equivalent circuit shown in Fig. 2.6. The detector is modeled as signal generator and capacitance in parallel. The noise is modeled with voltage sources in series and current sources in parallel with the detector and perfect amplifier. ENC is the noise charge at the input to the amplifier in this model. It is also a measure of how well the amplifier and shaper filter noise.

The noise is a sum of the series white noise, 1/f-noise and parallel white noise [26]:

$$\begin{aligned} ENC^2 &= ENC^2_{series} + ENC^2_f + ENC^2_{parallel} \\ = \frac{1}{2}e_n^2C_{in}^2 \int\limits_{-\infty}^{\infty} \left[w'(t)\right]^2 dt &+ \pi C_{in}^2 A_f \int\limits_{-\infty}^{\infty} \left[w^{(1/2)}(t)\right]^2 dt + \frac{1}{2}i_n^2 \int\limits_{-\infty}^{\infty} \left[w(t)\right]^2 dt \\ &= \frac{1}{2}e_n^2C_{in}^2\frac{A_1}{\tau_p} + \pi C_{in}^2 A_f A_2 + \frac{1}{2}i_n^2 A_3 \tau_p \end{aligned} \tag{2.14}$$

The average series noise voltage spectral density is $\overline{e_n^2} = 4kTR_s$, where R_s is equivalent series noise resistance. The 1/f-noise spectral density is given by A_f/f, where A_f is a constant. Parallel noise current spectral density is $i_n^2 = 2qI_0$ for current trough the detector and $i_n^2 = 4kT/R_p$ for a resistance in parallel with the detector. The constants A_1, A_2 and A_3 depend on the weighting function shape. τ_p is the peaking time of the impulse response. [26]

Since the series noise is inversely proportional to peaking time and parallel noise is directly proportional, there exist an optimal shaping time, that minimizes the noise. The 1/f-noise is independent of the shaping time, but is affected by the form of the weighting function. The 1/f-noise has fractal nature. It looks the same independent of the time scale. It can be minimized by adjusting the weighting function shape. The 1/f-noise is proportional to constant A_2. For example for triangular weighting function $A_2 = 0.88$, for Gaussian $A_2 = 1.0$ and CR-RC $A_2 = 1.18$. Hence the best choice for minimizing 1/f-noise would be triangular weighting function. [26]

Most of the noise is integrated over in the shaper. Noise that averages on zero and has frequency outside passband is filtered out. Noise that consists of unipolar pulses is integrated to a constant baseline. The series and parallel white noise are both temperature dependent. This results in a baseline, that shifts with temperature.

As discussed previously the input to ADC needs to be band-limited. Frequencies above the Nyquist frequency result in aliasing. The losses in signal to noise ratio due to aliasing cannot be recovered by any digital signal processing means. In practice this means that at least some analog low-pass filtering is always needed before sampling [26]. The ADCs contribute to the signal to noise ratio as discussed previously. When the number of bits is high, the quantization noise is low.

How well the bandwidth of the system is coupled with the signal frequency, affects the SNR in digital signal processing as well as in analog shaping. In addition to this any arithmetics in digital system may suffer from rounding errors. To minimize this error the whole digital signal processing can be performed with increased precision and rounded only after the full processing chain. The benefits are usually well worth the increase in surface area.

3. Digital Signal Processing Algorithms

3.1. General requirements and similarities with S-Altro chip

There are five general tasks that the signal processing unit should be able to perform:

1. Baseline removal.
2. Noise filtering.
3. Pulse shaping and/or tail cancellation.
4. Pulse time pick-off.
5. Trigger production.
6. Zero suppression.

Noise originating from the detector is mostly filtered in the analog front-end, but slow temperature drifts in the baseline and the noise originating in the front-end chip itself are propagated. Low frequency noise, such as the temperature drift, can be filtered with the baseline, when using adaptive method for calculating the baseline in stead of fixed baseline. Some time pick-off methods are highly sensitive to high frequency noise.

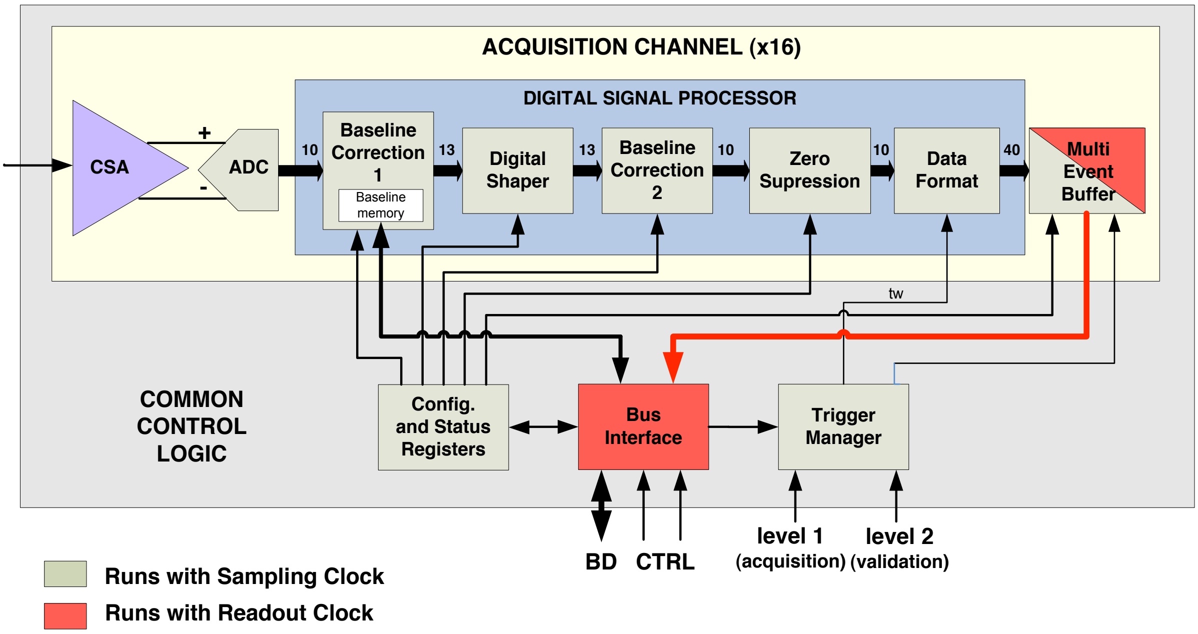


Figure 3.1.: S-Altro prototype block diagram (Image source [20])

Pile-up[1] might become a problem with the relatively high hit rates in the forward region and long shaping times necessary for gas detectors. The pulse shape can be modified to reduce pulse pile-up. Either the whole pulse is shortened or only a long tail is removed.

Much of the requirements for GdSP is similar to S-Altro chip. The block diagram of the S-Altro prototype is shown in Fig. 3.1. The Verilog code of the digital signal processing part of the prototype was available as a starting point for the GdSP development. As seen from the figure baseline removal, digital shaping and zero suppression are common tasks for the chips. The signals from TPC and GEM both benefit from long shaping times. Differences arise in the requirement for timing information and trigger production for GEMs. The GEM signal might benefit also from high frequency noise filtering, which could be needed to obtain good resolution for the timing extraction algorithms.

3.2. Baseline Correction

After digitization the signal has an arbitrary baseline value, that is usually optimized so that as much of the dynamic range as possible can be taken advantage of and in the same time allow some fluctuations in the baseline. If the signal is scaled too strictly to match the ADC range, even small baseline fluctuations take the signal outside the dynamic range. In addition to the added constant baseline the baseline fluctuates slowly with temperature. As seen in section 2.5 on page 14 the thermal noise current, that is after analog filtering seen mostly as baseline level, depends on the temperature. The restoration of the baseline to zero is important, because many techniques in digital signal processing, such as the pole-zero filter, are based on the assumption, that the baseline is removed.

There are several possible approaches to Baseline Correction (BC). Easiest is to subtract only a constant baseline. This is adequate only if there is no remarkable fluctuations in the baseline. For elimination of the slow fluctuations a form of high-pass filter can be used. Usually it is based on subtracting averaged signal. The moving average of the signal is calculated continuously and subtracted from the signal. In applications for fast detectors, when the pulse rise times are short, the moving average of few samples can be simply subtracted from the signal [37]. This creates a undershoot after a pulse that is relative to pulse height and duration as illustrated in Fig. 3.2. For short pulses and moderate rates the dead time created after a pulse can be tolerated. The pulses in gas detector applications are as a rule too long and the rates in LHC too high. Such a dead time after the pulse would lower the detection efficiency considerably. In S-Altro the problem of the undershoot is solved by freezing the baseline calculation for the duration of the pulse as illustrated in Fig. 3.3. The signal is compared two low and high threshold. When the signal is outside the thresholds, the calculation is frozen. The drawback of this solution is the risk that baseline calculation gets frozen unintentionally and remains at erratic value until reset.

In the S-Altro the baseline correction is performed in two stages as seen in the prototype block diagram figure 3.1. The first Baseline Correction block (BC1) offers several options for more coarse baseline correction. The second Baseline Correction block (BC2) uses the moving average subtraction with pulse exclusion for finer corrections of baseline drift. The BC1 offers options for constant pedestal subtraction, subtraction of pre-recorded baseline from SRAM, conversion mode, where the SRAM is used as a look-up-table, test mode, where the input signal is read from the SRAM, self calibrated variable pedestal (Vpd) subtraction and several combinations of these. The ALICE experiment takes data in bursts. Data acquisition windows alternate with time when only background is present. The Vpd value is calibrated outside the acquisition window and kept constant during the acquisition. The

[1] When sequential pulses are close enough to overlap significantly, the latter pulse will have increased height.

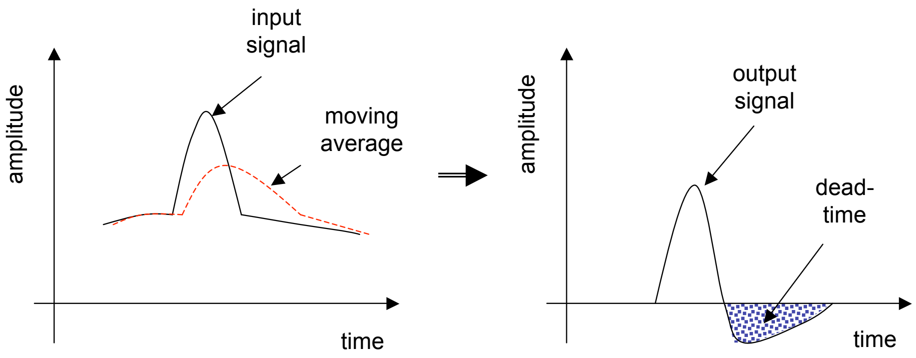


Figure 3.2.: Moving average subtraction.

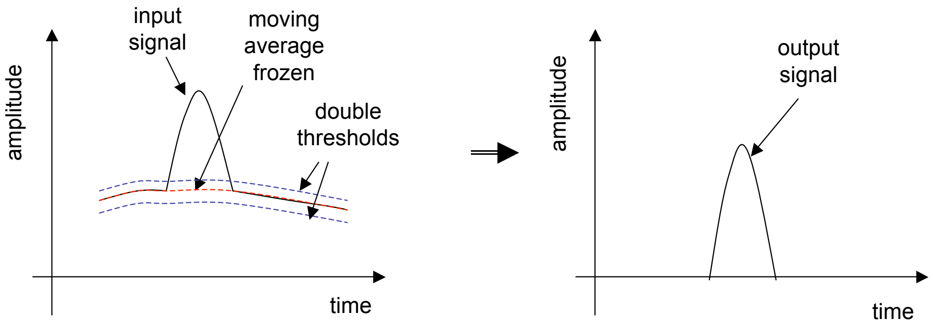


Figure 3.3.: Moving average subtraction with pulse exclusion.

CMS takes data continuously, which means that this approach is not applicable; there are no acquisition windows and hence no time between them. The idea behind pre-recorded baseline is to remove systematic perturbations during an acquisition window. This could be used also in continuous mode, if there are systematic perturbations occurring with known exact frequency. The justification for having SRAM for each channel in GdSP would not come from baseline correction, but rather from the conversion mode and test mode. The presence of moving average subtraction does not exclude the alternative baseline correction methods. The idea to offer the user a few options for baseline correction is seen as wise. It is for example difficult to foresee whether the stability of simple constant pedestal subtraction is valued over the more precise moving average subtraction. Although extra caution is taken to make the moving average calculation as stable as possible, the long term stability is ultimately revealed only in use.

The moving average calculation can be implemented as Infinite Impulse Response (IIR) or Finite Impulse Response (FIR) filter. Fig. 3.4 shows an example of IIR implementation of baseline calculation. The double threshold scheme is also included. This filter is used in S-Altro for Vpd calculation. For every sample the difference between the signal and calculated baseline is divided by an exponent of two and added to the baseline. When the baseline is below the signal, it is increased. When the baseline is above the signal, it is decreased. As the baseline approaches the correct value, the corrections get smaller. The effects of a given

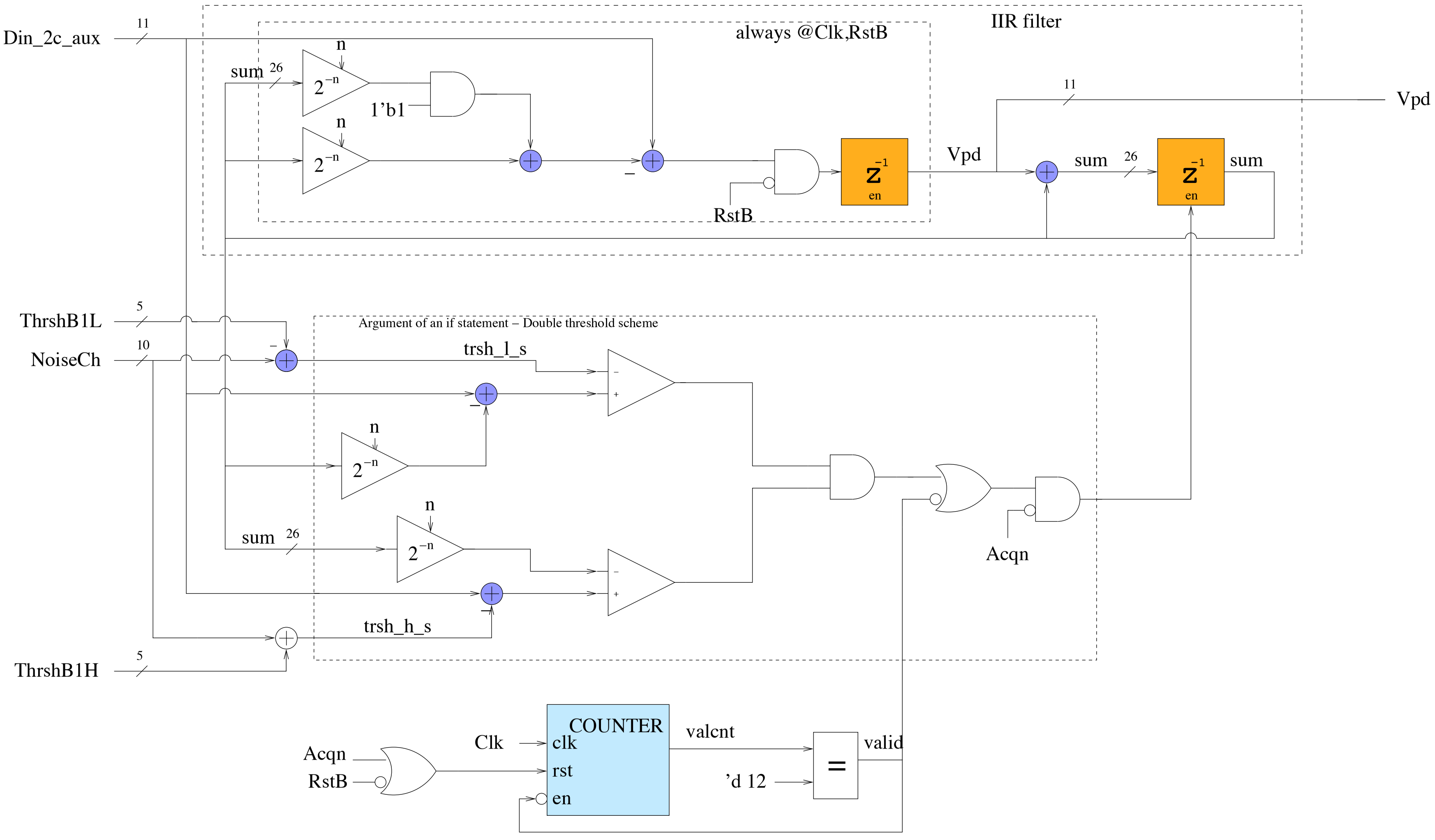


Figure 3.4.: IIR implementation of moving average filter with double threshold scheme [20].

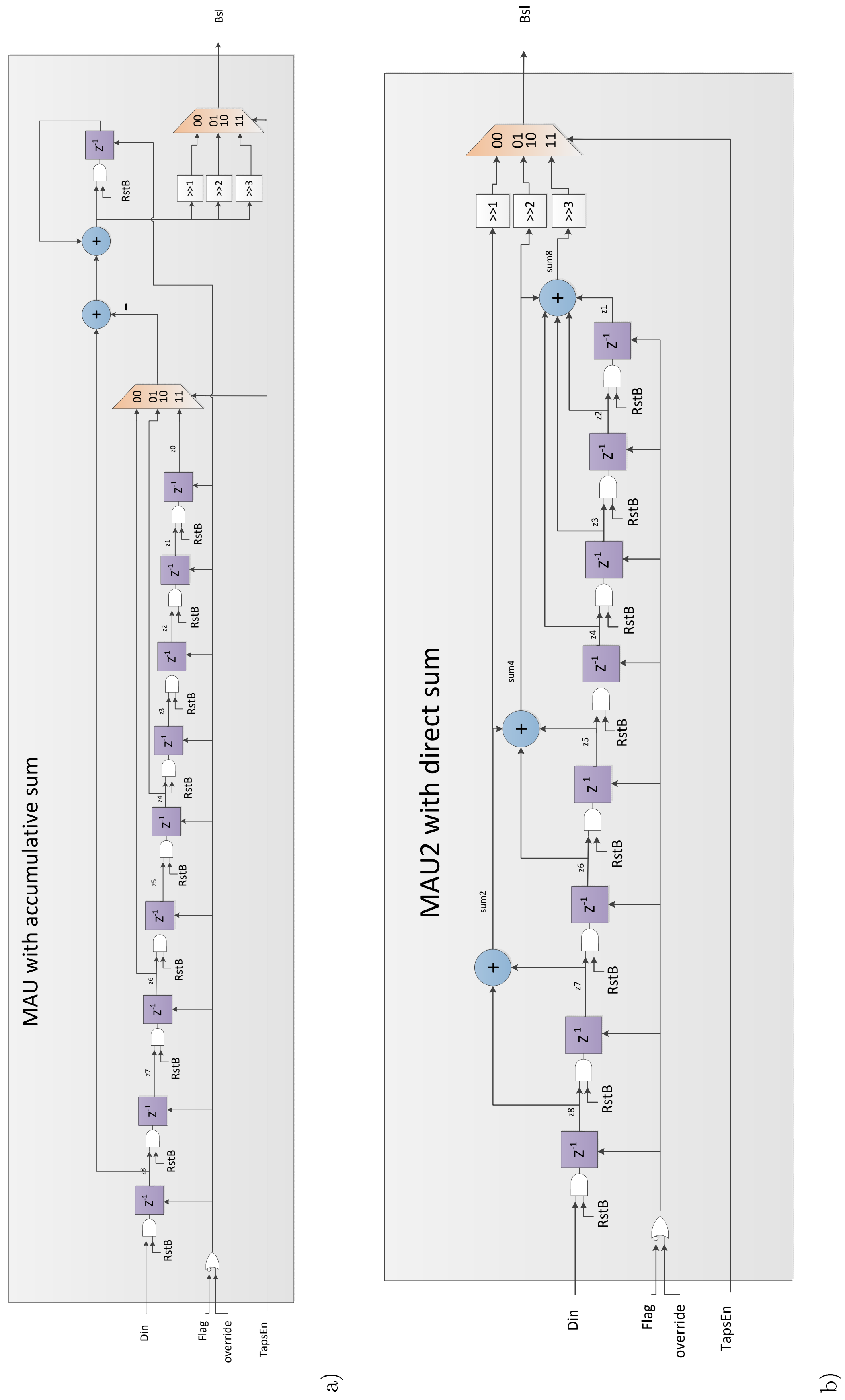


Figure 3.5.: Moving average unit using a) accumulative sum b) direct sum FIR filter.

sample are diluted with time, but they cease to affect the output completely only at reset.

Two examples of FIR filters are shown in Fig. 3.5. The filter a) is used in S-Altro for moving average calculation in BC2. It uses accumulative sum. The filter b) uses direct sum. In both cases the samples are read into a pipeline. With direct sum, the sum over two, four and eight samples is taken and divided with two, four and eight respectively. The used average is selected from these options.

With accumulative sum the same averages over two four or eight samples can be calculated. Lets assume that the average over eight samples is calculated. Always when the pipeline moves the newest sample $z8$ is added to the sum and the oldest sample $z0$ is subtracted:

$$sum[n] = sum[n-1] + z8[n] - z0[n], \tag{3.1}$$

where n denotes the current time. When there are nine samples in the pipeline, this gives the sum over the first eight samples. It is easy to see by inserting the previous sum

$$sum[n-1] = z8[n-1]+z7[n-1]+z6[n-1]+z5[n-1]+z4[n-1]+z3[n-1]+z2[n-1]+z1[n-1]$$

to the accumulative sum (eq. 3.1) and remembering how the pipeline shifts, ie. $z0[n] = z1[n-1]$ and so forth.

The calculated sum is divided by the selected exponent of two (two, four or eight) by shifting bits. The number of samples in the sum is altered by taking different sample in the pipeline as the new sample.

In principle the direct and accumulative sums give the same results, but there is a suspicion that accumulative sum might have stability issues. The suspicion is based on experiences by users of ALTRO chip, which has accumulative sum implementation in baseline correction. Instability with the accumulative sum was not experienced in simulations (section4).

IIR filters are in general more efficient in implementation. This efficiency can be measured by counting how many adders there are in the design. In contrast to the two adders in the IIR implementation of baseline calculation, the FIR implementation requires five adders and even then the range of programmability is more restricted. The downside to IIR filters is that they can be unstable [29, 30]. In baseline calculation instability would manifest itself as radically oscillating value for the calculated baseline. The stability needs to be checked through analysis of the region of convergence i.e. the poles of the filter need to be inside the unit circle. FIR filters are always stable, because they always have all poles in origo.

3.3. Digital Shaping

The main aim of digital shaping is to shorten the signal and consequently reduce the effect of pulse pile-up. In gas detectors ion tails are common causes for the need of digital shaping. One of the benefits of GEM detectors is that they do not suffer from ion tails, because the signal is formed from electrons alone. The need for shaping would come from the need for long shaping times that arises from the long charge collection time. The anticipated muon trigger rate at CMS high η region is 1 MHz. This corresponds pulses that arrive on average every 40 sampling clocks. With the longest shaping time in the analog filter, the pulse itself is almost 40 clocks long. In this environment pulse pile-up is likely to happen, but it is unsure if it will be enough to cause problems if left untreated. The time pick-off techniques are in general designed to tolerate moderate pulse pile-up.

3.3.1. Pole-Zero Cancellation

Pole-zero cancellation is often used method in detector signal processing both in analog and digital form. It is usually used either to eliminate undershoot after pulse [6] or long ion tail

[20].

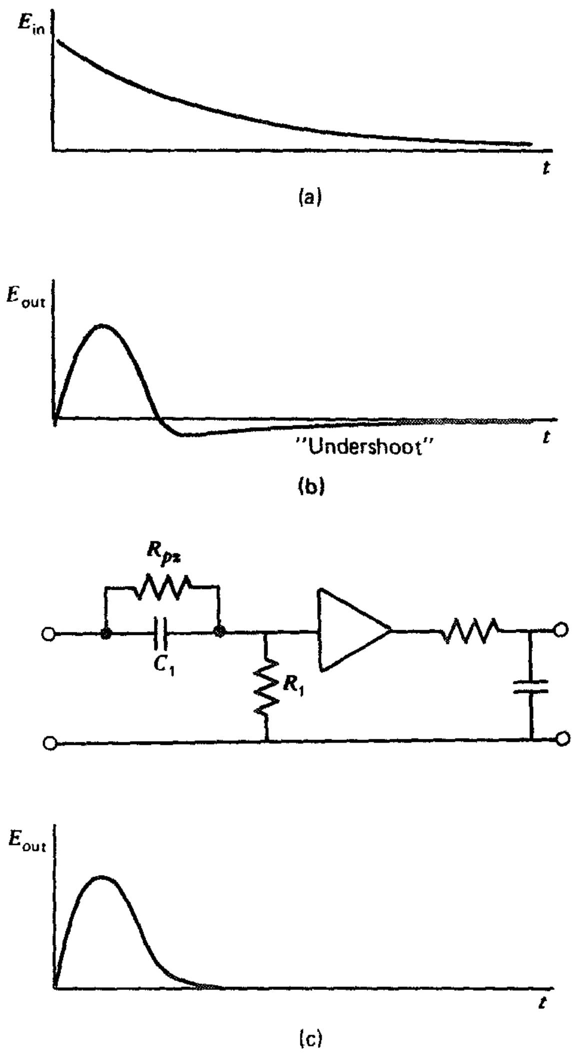


Figure 3.6.: Pole-zero cancellation with analog electronics. a) Signal from preamplifier. b) Signal from CR-RC shaper. c) CR-RC shaper with pole-zero cancellation. d) Signal with pole-zero cancellation. (Image source [6])

The best demonstration to pole-zero cancellation is to use analog filter and oscilloscope with particle detector. The pole-zero filter has only one control, that adjusts the resistor $\mathrm{R_{pz}}$ in Fig. 3.6. Turning the control one way will make the pulse tail longer. Adjusting the other way will shorten the tail and eventually result in undershoot.

A digital pole-zero filter has similar behavior, but the poles K and zeros L are given directly as variables. In section 2.4 the transfer function of pole-zero filter was calculated and impulse response was investigated.

Pole-Zero cancellation is the filter of choice in S-Altro. Its block diagram can be found in appendix B. Its Digital Shaper[2] is a 4^{th} order pole-zero filter. It is used primarily to cancel ion tails in the signal. It consists of four first order pole-zero filters in cascade. The poles and zeros are constrained to be real, positive and inside the unit circle. These constraints ensure that the filter is stable. [20]

[2] It also goes by name Tail Cancellation Filter.

The pole-zero filter has one obvious limitation. It cannot make the pulse rise time any shorter. It can shorten the pulse tail, correct undershoots and increase pulse rise time.

3.3.2. Single Delay Line Shaping

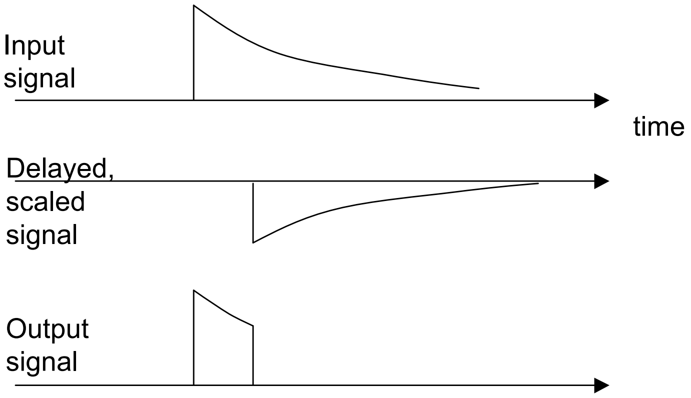


Figure 3.7.: Delay line shaping. The output signal is the sum of the original signal and the delayed, inverted and scaled signal.

Single Delay Line (SDL) shaping illustrated in Fig. 3.7 has its origins in coaxial cables that are shorted at the receiving end. The signal is reflected from the shorted end and the reflection cancels the signal tail [6]. Same effect can be achieved with digital filter that delays and scales down the signal and subtracts it from the original signal. The resulting pulse has the duration equal to the delay. To avoid ballistic deficit in the pulse height the delay needs to be at least as long as the pulse peaking time.

This is most effective technique when the signal has either step function shape or long exponential tail. With GdSP SDL was considered for use on preamplifier signal allowing to skip the analog shaping entirely.

3.3.3. Peak Sharpening

Peak sharpening or resolution enhancement is more common procedure in context with distributions (e.g. spectroscopy) and image processing. It improves the apparent peak resolution making the peak artificially more narrow. It makes superimposed peaks easier to isolate. It is related to deconvolution, where the response function broadening the peak or pulse is known and can be countered. Deconvolution is discussed more in the chapter 3.4.3, where it is used in the context of the time pick-off algorithm. Unlike with deconvolution the convoluting function does not need to be known for peak sharpening. There are at least two approaches for peak sharpening. One is based on subtraction of second derivative of the signal [31]. Alternatively FIR filter calculating weighted sum of few samples could be used [35]. Limited time for the design process and prioritization of time-pick off over digital shaping allowed to design and test only one peak sharpening filter. The second derivative technique was chosen as it was simpler and more interesting. It has simple and eloquent mathematical foundation, whereas the FIR filter is a collection of fitted parameters. The second derivative technique is effective but very sensitive to noise. From SNR considerations point of view a FIR filter might have been a better option.

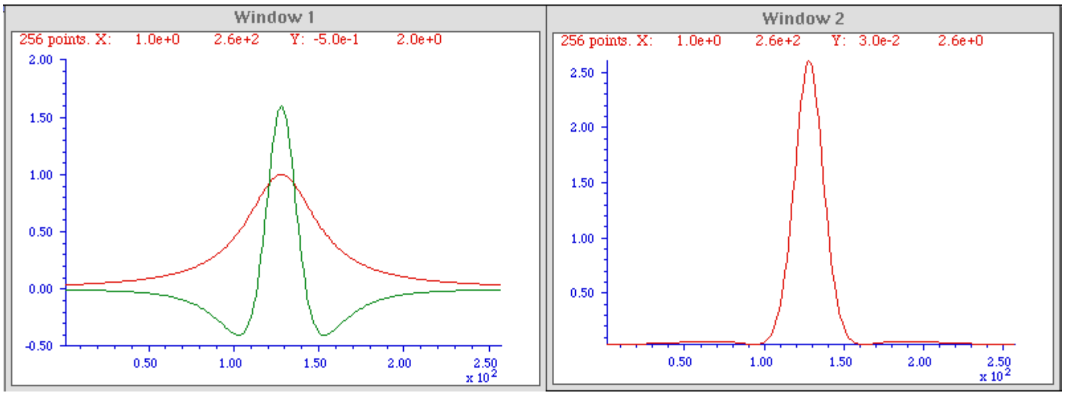


Figure 3.8.: Peak sharpening using second derivative. On left a Lorentzian peak in red is shown together with negative of its second derivative in green. On right is output peak. (Image source [31])

Peak Sharpening using second derivative subtracts the second derivative x'' of the signal x. The output signal y is given by

$$y = x - k \cdot x'', \tag{3.2}$$

where k is adjustable parameter. For discrete time filter realization the equation 3.2 gets form

$$y[n] = x[n] + k \cdot (\{x[n+1] - x[n]\} - \{x[n] - x[n-1]\})\,. \tag{3.3}$$

Fig. 3.8 shows a Lorentzian peak with its second derivative and the resulting enhanced peak. The inverted second derivative will approximately cancel the signal at the sides and add to the signal at the peak. In the output peak, the baseline on each side of the peak is not exactly zero. The wiggles are minimized by adjusting k appropriately. The total area under the second derivative is zero. This means that the area under the peak is conserved. Because the technique is linear, the proportionality of pulse height to collected charge is conserved and normal calibration techniques are applicable.

3.4. Bunch Crossing ID, Amplitude and trigger

In general a trigger is binary logic pulse marking a pulse of significant height. In particle detectors the first level trigger tells that a particle has been detected.

The trigger and time-tag from the filter need to have fixed time delay in respect to the beginning of the pulse. If trigger is given when the signal crosses a threshold, the trigger may have time walk for two different reasons (see Fig. 3.9). The time walk might originate in varying pulse rise time. The rise time is usually assumed constant. With short shaping times, the GEM signal rise time does have variance. The rise time walk cannot usually be eliminated. Another source of time walk is the varying pulse height. Different timing is given to pulses with different amplitudes. The amplitude time walk can be eliminated using two different strategies. One can use threshold comparison and use pulse amplitude to correct the timing. Second strategy is to pick a characteristic of the pulse, for example peak, that has fixed delay from the pulse beginning and use it for trigger production.

In the applications foreseen for the GdSP-chip timing information is more crucial than amplitude information. The needed accuracy for measured amplitude is not known. It's not even clear if amplitude information is needed outside the chip. There are two basic

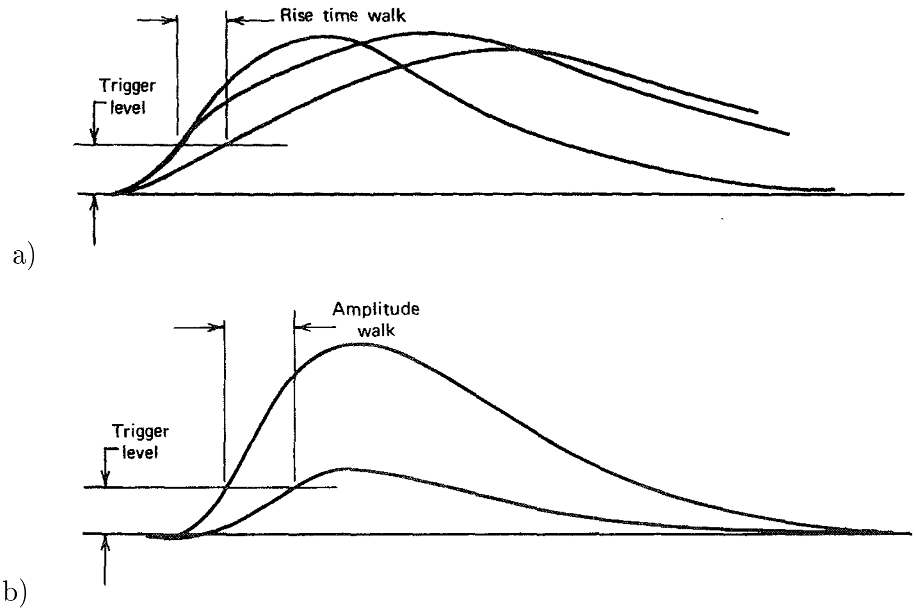


Figure 3.9.: Time walk originating from a) variations in rise time and b) variations in pulse height (Image source [6])

techniques for extracting amplitude information. In simplest case, we simply take the value of the largest sample in a pulse. For greater accuracy weighted sum of samples around the peak of the pulse can be used. The Piece-wise linear fitting algorithm uses the weighted sum method and uses it in the calculation for timing information. The other bunch crossing identification (BXID) algorithms can be used with either one of the amplitude extraction methods.

Some assumptions were made in designing the BXID:

1. Pulses are synchronous with the sampling clock.
2. Phase difference between the clock and pulses can be adjusted i.e. signal can be delayed with precision of few nanoseconds in the analog front-end.
3. Timing information is needed only for bunch crossing assignment (time-bins are 25 ns wide).

The first assumption is met in LHC experiments in general, but not necessarily in other possible applications. In GdSP analog front end the signal delay can be adjusted. If either one of these assumptions should not be met, the best time resolution that can be delivered is half sampling period (12.5 ns). If better resolution is required, there are other possible approaches: To use BXID algorithm that delivers natively sub-clock precision time-stamp (for example piece-wise linear fitting), use faster ADC, interpolate the signal and use existing block (for example Constant Fraction Discriminator (CFD)) with faster clock or interpolate inside the BXID block (for example Zero-Crossing Identification (ZCI) and CFD are well suited for this).

The S-Altro does not produce trigger and extract pulse timing information. This is however a common task in detector electronics and there is no need to invent a new filter. To get the best performance several known methods were compared. Multiple simulations were designed to help in the designing of the filters and to choose the best from them. The initially considered filters were Piece-wise linear fitting[32], Time over Threshold, Deconvolution method[33, 34], CFD, Peak Finder and ZCI. One previous study including Peak Finder, Deconvolution filter, ZCI and CFD was found [35]. It was conducted with quite different setup in mind (calorimeter as detector and constant, short shaping time) and the results could not be reliably generalized for GdSP.

3.4.1. Piece-wise linear fitting

Piece-wise linear fitting (PWLF) was developed by Buzuloiu for time and amplitude pick-off in HEP applications[32]. The amplitude is calculated with a weighted sum of three samples. The amplitude time walk is calculated using a Look-Up Table (LUT) with another weighted sum as input. The parameters for the weighted sums are calculated by fitting a line to the pulses in the sample space (Fig. 3.10).

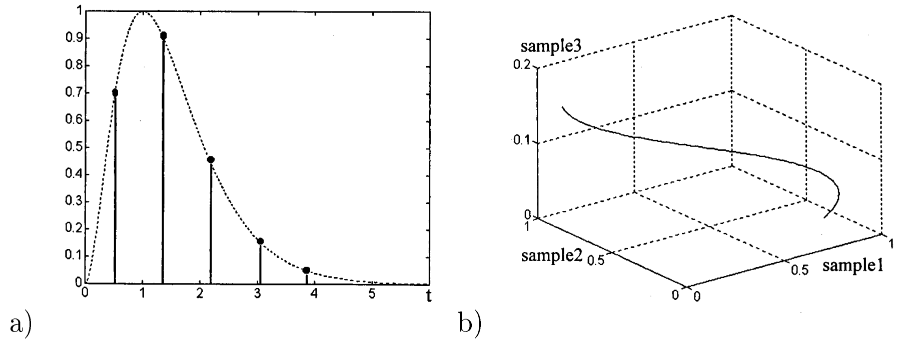


Figure 3.10.: a) Sampled pulse b) Pulse representation in sample space (Image source [32])

Unlike the other time pick-off methods described in this chapter, this method is designed to compute the time walk with time steps smaller than the sampling period. This means it is more precise than a mere bunch crossing algorithm.

For the GEM application only bunch crossing identification was needed and PWLF was considered too complex and needy on chip resources.

3.4.2. Time over Threshold

Time over Threshold (ToT) is based on trigger production at threshold crossing. The timing is corrected afterwards. At threshold crossing a counter is started. It is stopped when the signal returns below threshold. The reading on the counter is proportional to the pulse amplitude. The pulse amplitude and the time walk can be read from LUT (look-up table) using the time over threshold as input. The operation principle of ToT is illustrated in Fig. 3.11.

ToT is often the only available method offline, when the only information on the signal is binary values from a comparator. It is also a valid method when used on analog signal (e.g. see simulation results using analog ToT in fig. 4.12).

In digital designs ToT has certain challenges. As a part of digital signal processor, it would be easiest to use the sampling clock for the counter. The pulse length varies with the shaping time from 4 to 32 sampling clocks. Taking the average pulse length of 14 clocks and quantization error of half sampling clock, the uncertainty of ToT value becomes 3.6 %. To get more accurate value one needs a faster counter. In this case the signal needs to be interpolated. Then the accuracy of the threshold crossing highly depends on the accuracy of the samples on either side of the crossing. This leaves the method highly sensitive to noise.

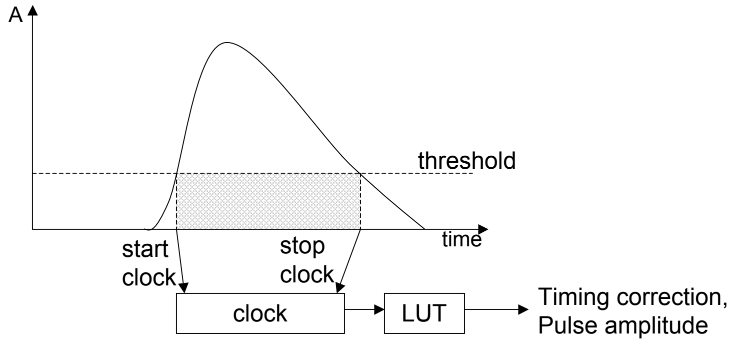


Figure 3.11.: ToT operation principle.

The need to choose between relatively high uncertainty and more complex algorithm that is sensitive to noise led to the decision to exclude ToT from the simulations discussed in Section 4.

3.4.3. Deconvolution method and pulse recognition

The deconvolution method aims to restore the signal to the form prior to the convolution in preamplifier and filters[34]. It has been successfully used in Preshower electronics[33]. In Preshower application it is assumed that the signal from the detector can be approximated by a delta peak. In practice it means that the original signal pulse is shorter than 25 ns.

The deconvolution method is based on matrix inversion of the impulse response of the amplifier (and filters) $h(\Delta t) = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots \end{bmatrix}$, where Δt is the sampling period. When signal S with samples s_i is convoluted with the matrix H with elements h_{ij} from impulse response, then the convoluted signal V with samples v_j is obtained:

$$\begin{bmatrix} h_1 & 0 & 0 & 0 & \cdots \\ h_2 & h_1 & 0 & 0 & \cdots \\ h_3 & h_2 & h_1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \end{bmatrix} \tag{3.4}$$

The original signal s can be found by using matrix inversion: $S = H^{-1}HS = WV$, where $W = H^{-1}$. [34]

The deconvolution filter used in Preshower is a variation of the theme. It uses three samples. Three quantities are calculated

$$\alpha = w_1 v_1 \tag{3.5}$$

$$\beta = w_1 v_2 + w_2 v_1 \tag{3.6}$$

$$\gamma = w_1 v_3 + w_2 v_2 + w_3 v_1 \tag{3.7}$$

and a bunch crossing is assigned when β is the greatest quantity: $\alpha < \beta > \gamma$ [33]. If deconvolution method is used on non-ideal pulse, the result is a short pulse instead of one clock long delta peak as shown in Fig. 3.12. The quantities α, β and γ are needed for unambiguous bunch crossing assignment i.e. the first sample on a pulse is found. The method as it is used in Preshower works well only when the peaking time is shorter or equal

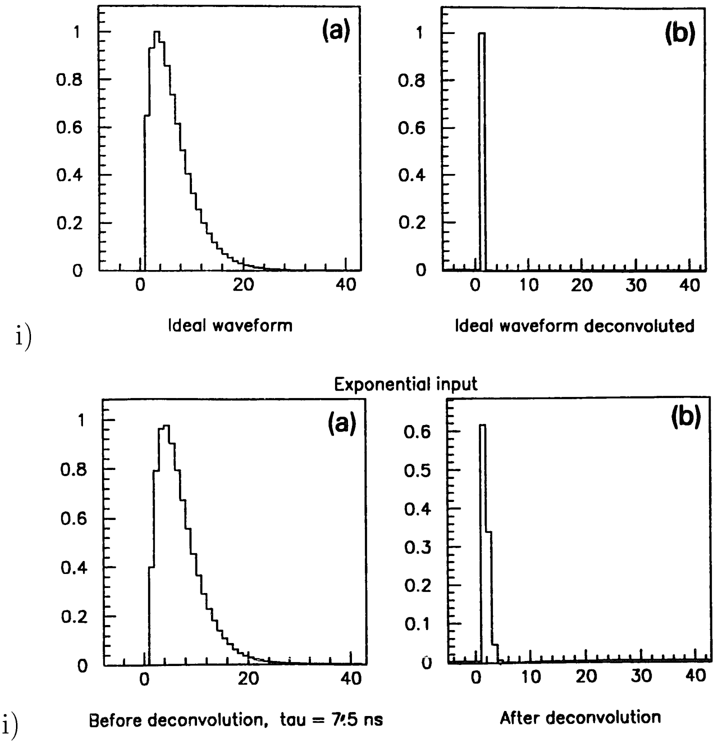


Figure 3.12.: Simulated deconvolution on i) ideal pulse and ii) non-ideal pulse [34]

to the sampling period. In the case where it was longer the calculation of suitable weights proved impossible in simulations (see section 4).

One of the strengths of deconvolution method is the extremely short dead time. It can tell apart two pulses at just two samples distance from each other[34]. With triple GEMs this might become disadvantage. It would be possible to have multiple triggers on one pulse, when using short shaping times, because the clusters in the GEM signal are not integrated fully.

For these reasons an alternative scheme was devised that acquired the name Pulse Recognition (PuR). It has roots in the deconvolution method used in Preshower. The quantities α, β and γ are calculated as described above. The difference is that the temporal distance n between samples v_2 and v_3 can be adjusted and the weights w_1, w_2 and w_3 are not calculated using deconvolution method. The temporal distance between samples v_1 and v_2 is constant 3 sampling periods. The idea is to iteratively find weights for which the inequality $\alpha < \beta > \gamma$ is true only when the sample v_2 is at the top of a pulse. In other words it recognizes the pulse.

For finding the weights the following method was used. Sampled average pulse and temporal distance n were assumed known. First the samples in the averaged pulse that are wanted as v_1, v_2 and v_3 at the time of trigger were identified as $signal(t_1)$, $signal(t_2)$ at the peak and $signal(t_3)$. Weight w_1 is set to 1. To calculate weight w_2 from the requirements that the inequality $\alpha < \beta > \gamma$ holds for the identified samples, but not when the time is shifted. By assuming that $signal(t_1) \ll signal(t_2)$ and shifting either $signal(t_1)$, $signal(t_2)$ or $signal(t_3)$ or all of them by one clock one gets inequalities

$$w_2 \leq w_1 \left(1 - \frac{signal(t_3 - \Delta t)}{signal(t_2)}\right) \tag{3.8}$$

$$w_2 \leq w_1 \left(1 - \frac{signal(t_2)}{signal(t_1 + \Delta t)}\right) \tag{3.9}$$

from which the smaller is picked as the value for w_2. The last remaining weight can be then be calculated using

$$w_3 = w_2 \left(1 - \frac{signal(t_2 + \Delta t)}{signal(t_1 + \Delta t)}\right) + w_1 \left(1 - \frac{signal(t_3 - \Delta t) - signal(t_2 + \Delta t)}{signal(t_1 + \Delta t)}\right). \tag{3.10}$$

The best value for n was found iteratively. Simulations were run with several different n and the one yielding best time resolution was chosen. The optimal number was usually when the $signal(t_1)$ was found among the first samples on the pulse.

An approximate total charge Q at trigger ($\beta = Q$) can be obtained when the weights are calibrated using the same scaling factor for all of them. For the GEMs a total charge measurement is not necessary and scaling was not performed. For very accurate charge measurement the charge should be calculated using 3 additional weights as is done in Preshower.

3.4.4. Constant fraction discriminator

Constant fraction discriminator (CFD) produces a trigger, when constant fraction k of the pulse peak amplitude is reached. Because the rise time is constant, the timing for the pulses does not depend on the pulse amplitude. The trigger is given at fraction f of the peaking time t_p. This method is at its best, when the pulse does not have a sharp peak, but the pulse shape is constant. Most commonly $k = f = 1/2$.

Traditionally in both analog and digital filters the signal is divided to two components. One signal (i) is delayed by time t_d and the other (ii) is multiplied with $-k$. The two signals are summed (iii) and the triggering time t is found when the signal (iii) crosses zero.

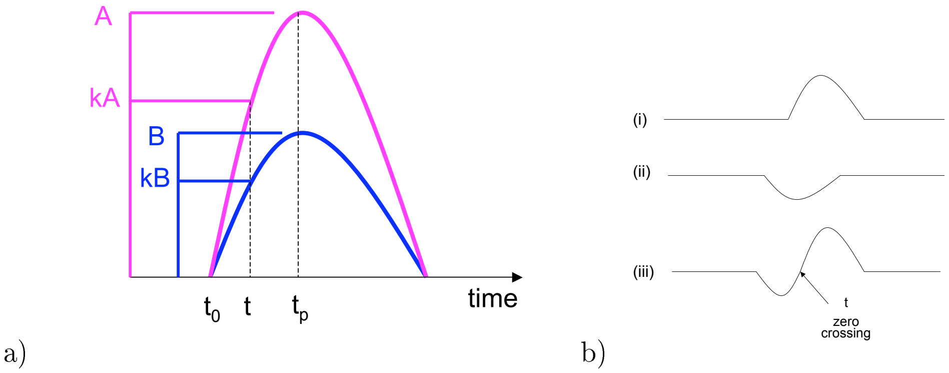


Figure 3.13.: Constant fraction discriminator. a) Timing is independent of pulse height. b) Waveforms used in CFD

Another common attribute to the pulses, besides the same fraction of the pulse peak height at the time of trigger, is that they have the same gradient at this point. For pulses that do not have constant derivative this can be exploited. Triggering on pulse derivative would not give unambiguous trigger. The correct derivative would be found before and after the maximum derivative. First of them would often, but not always, be below the threshold separating signal from noise. This would lead to unpredictable timing. In stead of derivative the relation of two sequential samples

$$a = \frac{sample(t + \Delta t)}{sample(t)}$$

was used. It has the same form and values for different pulse heights assuming that the baseline is properly removed. It has maximum value at the foot of the pulse. It falls from it's initial value reaching 1 at peak amplitude and nears zero asymptotically. The maximum value for a depends on the shaping time. For shorter shaping times the pulse rises with steeper slope. This leads to higher values for a. Picking a value for a that corresponds to the maximum derivative was found most functioning. The filter obtained with this approach corresponds to CFD with fixed time delay of one sampling period ($t_d = \Delta t$) and adjustable $k(a)$. The function mapping $a \to k$ depends on the form of the pulse. Most often it is more practical to determine a from waveform than analytical calculations.

3.4.5. Peak finder

Peak finder is a simple, straightforward algorithm. It compares three consecutive samples, s_{n-1}, s_n and s_{n+1}. If the middle sample is the greatest i.e. $s_{n-1} < s_n \geq s_{n+1}$, the peak is found. It's one of the four time pick-off methods listed in the paper on BXID for calorimeter [35]. It is used at least in ALICE EmCal trigger [37]. As it is simple it takes very little chip resources. For peak finder for one channel only two comparators are needed.

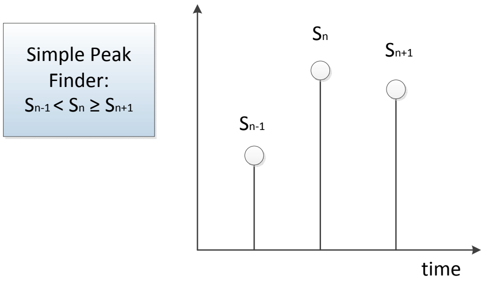


Figure 3.14.: Peak finder principle.

The limitation of the peak finder is that it works best for sharp peaks. This limits it's effective use to shorter shaping times. The amplitude information is given by the value of the peak sample.

3.4.6. Zero-crossing identification

Zero-crossing identification (ZCI) is a variation of peak finder. The simple peak finder described above only works for discrete samples. ZCI is implemented as readily for continuous as for discrete signals. It is based on the fact that the derivative of the signal at the peak is zero. The signal is first differentiated. The zero-crossing in the differentiated signal coincides with the peak in the original signal as shown in Fig. 3.15. Amplitude information is given by the amplitude of the original signal at the time of zero-crossing.

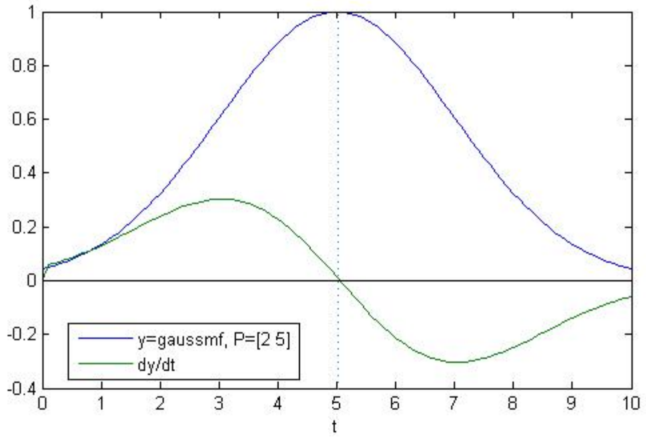


Figure 3.15.: Zero-crossing of differentiated signal. The derivative (green) crosses zero when the Gaussian pulse (blue) reaches its peak. Time and amplitude are in arbitrary units.

The algorithm is same for discrete signal. One has only to remember that the signal is delayed by 1/2 sampling period when differentiated. ZCI is one of the four BXID:s listed in the calorimeter study [35]. The digital algorithm is as simple as for peak finder and needs as much resources. It has the same limitations as well; It only works well for well defined peak, which is obtained when pulse rise time is not much longer than the sampling period.

3.5. Zero Suppression

In the average detector most of the time a given channel has nothing going on. The “zero” signal does not give any information and including it in data read out is a waste of bandwidth. In reality the signal is rarely strictly zero, because even in the absence of a pulse there is still noise. The task of Zero Suppression (ZS) is to discern the meaningful signal pulse from the meaningless background noise and tag the meaningful signal for readout. In its classic form ZS compares the signal to a fixed threshold, which is set above the noise level and below signal pulse height. Everything below the threshold is interpreted as zero and everything above as meaningful signal. The meaningful signal can be either flagged for data packet formatting or processed instantly. The method applies, when whole pulses are read out. Alternatively in some applications only pulse height data or binary data (hit or no hit) might be needed. The extraction of pulse height or binary data is the task of BXID algorithms described previously.

The ZS filter in S-Altro flags data for separate data formatting block. The ZS consists of five parts: comparison to absolute threshold, glitch filter, adding flag to pre-samples, adding flag to post samples and merging two flagged regions close to each other. The ZS block flags data for readout and storage. Fig. 3.16 illustrates how ZS selects the data for readout. Initially data is flagged when the sample is above an absolute threshold. Glitch filter removes flag from events that are shorter than is expected for a signal pulse. In order to save complete pulse, samples can be flagged as pre-samples before exceeding threshold. Similarly samples can be flagged as post samples after returning below threshold. If two flagged pulses are close together, the samples between them are flagged merging the flagged regions. The filter is discussed with more details in section 5.6.

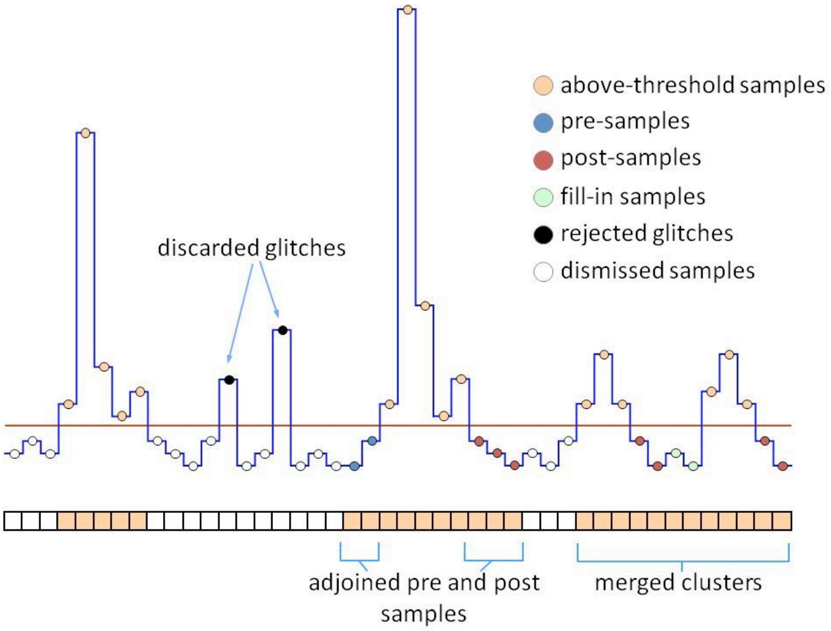


Figure 3.16.: Zero Suppression in S-Altro. The chain of boxes below waveform represent flag bits. (Image source [20])

4. Simulations

4.1. Migrating filters from S-Altro

The Verilog code from S-Altro was available as starting point for the DSP development. The overall architecture of the two chips is very different, but the requirements for the basic building blocks, the filters, were very similar. For the initial assessment of suitability for GdSP purposes the basic DSP blocks were extracted from the S-Altro design and a test channel was constructed, where filters could be easily examined individually and together. Each filter had a bypass option. The filters behavior was simulated using Cadence products NCLaunch and SimVision [38]. After the initial assessment the blocks were edited.

The Digital shaper was observed to have less effect on the signal length than expected. When using shaping times above 100 ns, the effects were barely noticeable and no visible undershoot was observed even with the maximum settings $L = 0$ and $K = 1$. The filter worked most effectively with 25 ns shaping time.

Two separate Baseline Correction (BC) blocks were thought to be excess and they were merged. The reasoning behind separate BC blocks was that the Digital Shaper (DS) needs at least rough baseline correction to operate normally. The Moving Average Unit (MAU) in BC2 can react unpredictably to pulse pile-up. The DS was thought to reduce the pulse pile-up. In the case of GEM detectors the DS was not seen to reduce pulse pile-up enough to justify separate BC blocks.

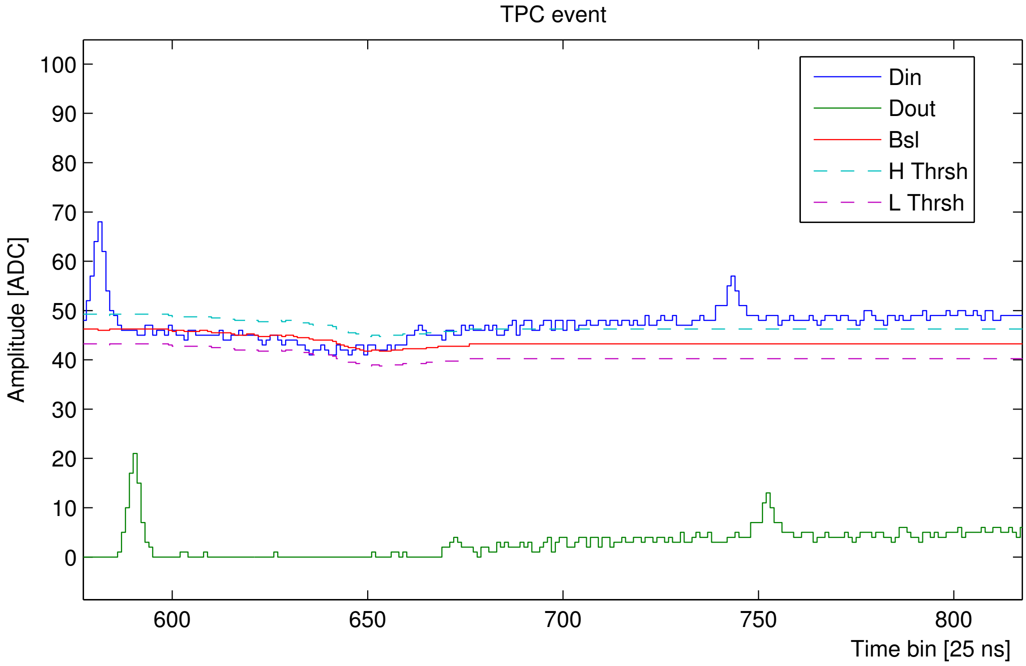


Figure 4.1.: Shifted baseline in BC2 with TPC data

Based on the feedback from Christian Lippmann, the BC2 was investigated with data from two TPC events that had caused problems with ALTRO. Both cases had a saturated pulse followed by shifted baseline. The behavior was reproduced with S-Altro filters when using tight relative thresholds. It was found that the baseline had little to do with the saturated pulse. It was instead caused by noise crossing the threshold and freezing the baseline calculation. A zoomed view of the event is shown in Fig. 4.1 where Din is the input signal to BC2, Dout output signal from BC2, Bsl the calculated baseline and H Thrsh and L Thrsh the double thresholds. When the input signal is outside the threshold limits, the baseline calculation is frozen. The baseline shifted outside the thresholds while the calculation was frozen and thus remained frozen. To prevent this the duration for which the calculation remains frozen after a pulse should have some dependence on the pulse duration. This was solved by disabling it for short, one to two clock long pulses. The latency after reset before threshold scheme is implemented was made programmable. A soft reset only for MAU was implemented. The adjustments increase the stability of the filter and make recovery from errors easier, but the filter is never completely stable while the double threshold scheme is used. The modifications include possibility to completely bypass the double threshold scheme. Same result might be achieved by setting the thresholds to maximum, but bypassing the thresholds is surer.

Some of the main changes to the BC blocks came from the desire to enable resetting while taking data. In S-Altro the reset could always be made outside the acquisition window and no attention had been given to artefacts arising from reset. In continuous data taking it would be advantageous to be able to reset without producing artefacts that might be interpreted as signal. Also programmability and feedback from the chip were taken into special consideration. Troubleshooting is easier when more data is available.

4.2. Preliminary Comparison of Different Time pick-off Methods

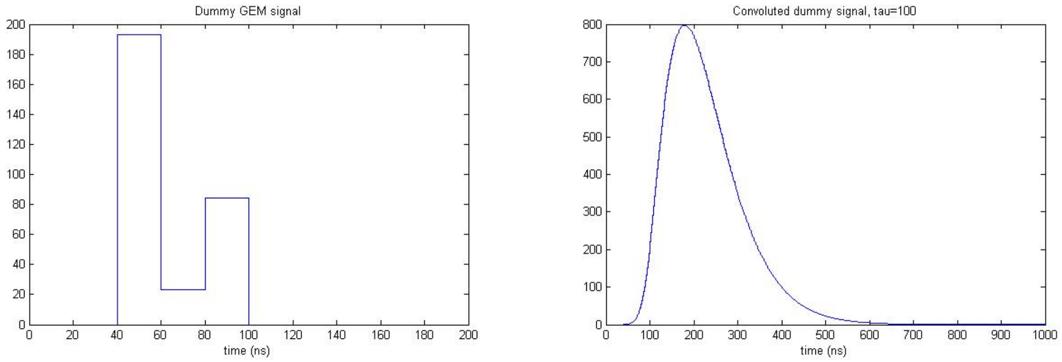
There are many different existing algorithms for extracting timing information from a pulse. The purpose of these simulations was to resolve which of the algorithms seemed most promising. Only the most promising ones would be further examined and developed.

There has been a previous study comparing Peak Finder, ZCI, CFD and a combination on Peak-Sharpening algorithm and Peak Finder [35]. The study served as a good starting point and source of inspiration, but a further study was needed since the methods were compared using only one, relatively short shaping time, whereas in this application the shaping time is programmable.

In many ways these simulations are only tentative and are only valid for comparison within the simulations. While adequate for the objectives of the simulations this does not fully describe a realistic system. The signal in these simulations is a sampled but not digitized. In other words it is a set of double precision floating point numbers. This was convenient for finding suitable values for the algorithm variables. The noise used in the simulations is based on Gaussian random number generator. Besides noise, signal and design, the time resolution is affected by the phase of the signal relative to the sampling clock. In these simulations the phase was not optimized and could add substantially to time resolution. In later simulations, that were compared to analog methods, the phase was optimized.

4.2.1. Approximation of GEM Signal

For the input signal a dummy GEM signal was generated. At the time of the simulations no large amount (over 100 pulses) of data from either detector measurements or simulations



(a) (b)

Figure 4.2.: Creation of dummy GEM signal for simulations. a) Signal from GEMs is approximated by three boxes with random height. b) The approximated GEM signal is convoluted with the analog front-end transfer function.

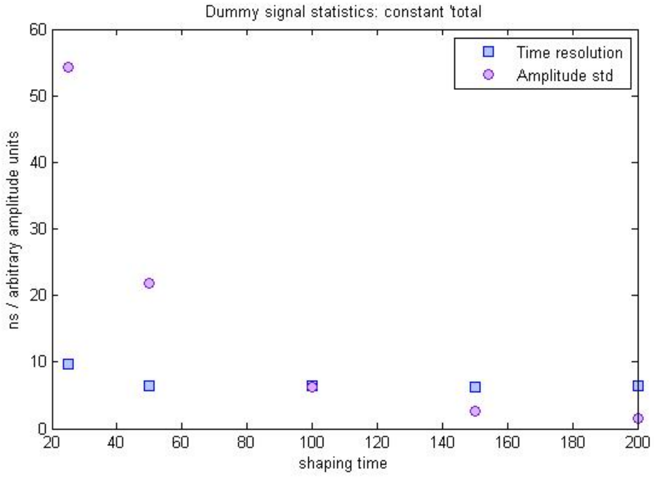


Figure 4.3.: Pulse height resolution and peaking time resolutions of the dummy signal (using peak finder with 1 ns sampling).

were available. For the simulation purposes a simple approximation of the signal was drafted. The signal from GEMs appears to consist on average of three separate pulses close together that have random heights (see Fig. 2.2 on page 9). This signal was approximated with three boxes with random heights demonstrated in Fig. 4.2a. The signal was normalized so that the sum of the box heights was one. The signal was then convoluted with the transfer function of the analog front-end:

$$h(t) = \left(\frac{t}{\tau}\right)^2 e^{-\frac{2t}{\tau}} \tag{4.1}$$

where τ is the shaping time. The analog shaping time is programmable with five options 25, 50, 100, 150 and 200 ns, that were also used in the simulations. The resulting shaped signal is shown in Fig. 4.2b. The signal statistics, pulse amplitude resolution and peaking time resolution, shown in Fig. 4.3 are close to GEM signal, when using Ar/CO$_2$ gas mixture with proportions 70:30.

4.2.2. Simulation Methods

The simulations were conducted using Matlab Simulink. Initially crude models of the different methods were designed. The considered algorithms listed in the order from least promising to most promising were

- Time over Threshold (ToT)
- Piece-Wise Linear Fitting (PWLF)
- Deconvolution method
- Peak Finder
- Zero-Crossing Identification (ZCI)
- Pulse Recognition (PuR)
- Constant Fraction Discriminator (CFD)

where ToT was excluded already prior to simulations.

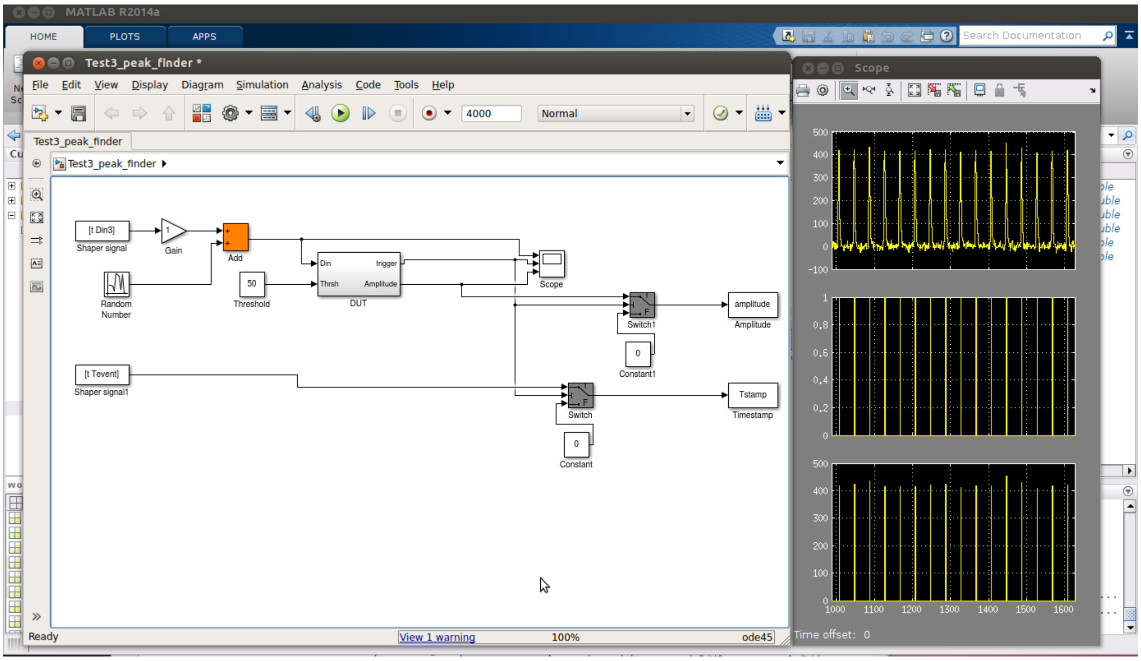


Figure 4.4.: Screen capture of Simulink simulation. The stimulus to the Design Under Test (DUT) comes from Matlab workspace variables and noise from random number generator. The amplitude and Timestamps from the trigger are saved as workspace variables. The input data, produced trigger and corresponding amplitude are monitored with Scope. The zoomed Scope waveforms are shown on right.

The simulations were conducted in rounds of test cases. There were altogether six test cases:

1. Design verification. The designs were tested with single test pulse. The criteria for verification was that one trigger was observed for the pulse and that the output amplitude scaled with the input pulse amplitude.

2. Effect of filtering noise and peak sharpening on time resolution.
3. Performance over shaping time. Reliable and even performance with the different pulse peaking times was desired. The effects of the detector were excluded by using same pulse 100 times. Only the random noise superimposed on signal stimulated the deviations from perfect performance. The time resolution, amplitude resolution and excess triggers caused by noise were measured. Based on the results some of the designs were dropped from further development.
4. Time resolution with realistic detector. The used dummy signal had similar time and amplitude resolution as GEM-detectors. Random noise was superimposed on the signal.
5. Sensitivity to pulse pile-up. The minimum temporal distance between two equal peaks was determined.

After or during each test case the models would be either discarded, modified or kept as they were. One of the models, PuR, started as repeated modification on Deconvolution filter. The final block diagrams of the Simulink models can be found in Appendix C.

A simulation test bench is shown in Fig. 4.4. The signal is read from Matlab workspace and the noise is generated in Gaussian random number generator. The input “Tevent” from Matlab workspace counts the bunch crossings from the incidence of particle to detector and pulse beginning. It simulates a counter that is started when the detector is stimulated. The signal may be amplified before it enters the Design Under Test (DUT). The amplitude output and the counter were sampled using the produced trigger and read into Matlab variables. In most simulations the different models were tested simultaneously by setting them in parallel in the test bench.

4.2.3. Simulation results

The most important results from the simulations were the time resolutions given by the different methods. When interpreting the plots one should give little attention to the exact values of the time resolution. It will change greatly with calibration. More important is the shape of the curve in respect to peaking time. For example the peak finder has relatively good time resolution with short 25 ns shaping time. With longer shaping times the time resolution deteriorates.

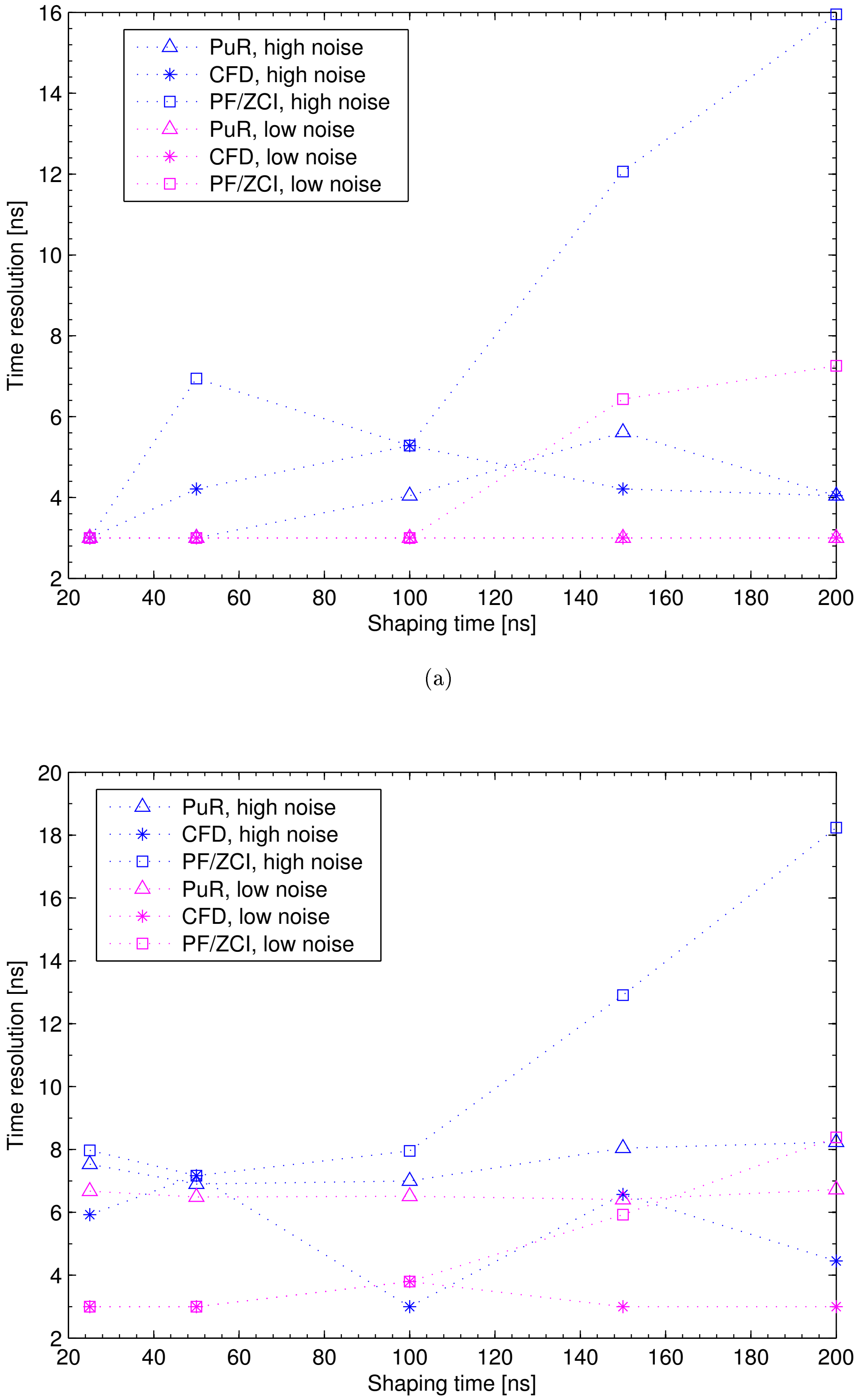
Fig. 4.5 shows the results for time resolution when

(a) an ideal detector was assumed; only random noise gives rise to the time resolution

(b) a realistic detector was assumed; in addition to random noise the detector signal has inherent time resolution.

The signal to noise ratio in the plots is $S/N = 16$ for high noise and $S/N = 50$ for low noise. The expected ratio is between them around $S/N = 20...30$. In all cases it can be seen that the peak finder algorithms give increasing time resolution for increasing shaping time. In the detector independent study CFD and PuR gave similar time resolution. With low level of noise the time resolution is beneath the detection limit. With GEM-like signal and low level noise CFD seems to give better time resolution than PuR. When the noise level is increased, the difference diminishes.

The Deconvolution method and Piece-Wise Linear Fitting (PWLF) are not included in the graphs. The Deconvolution method yielded consistently circa 19 ns time resolution with



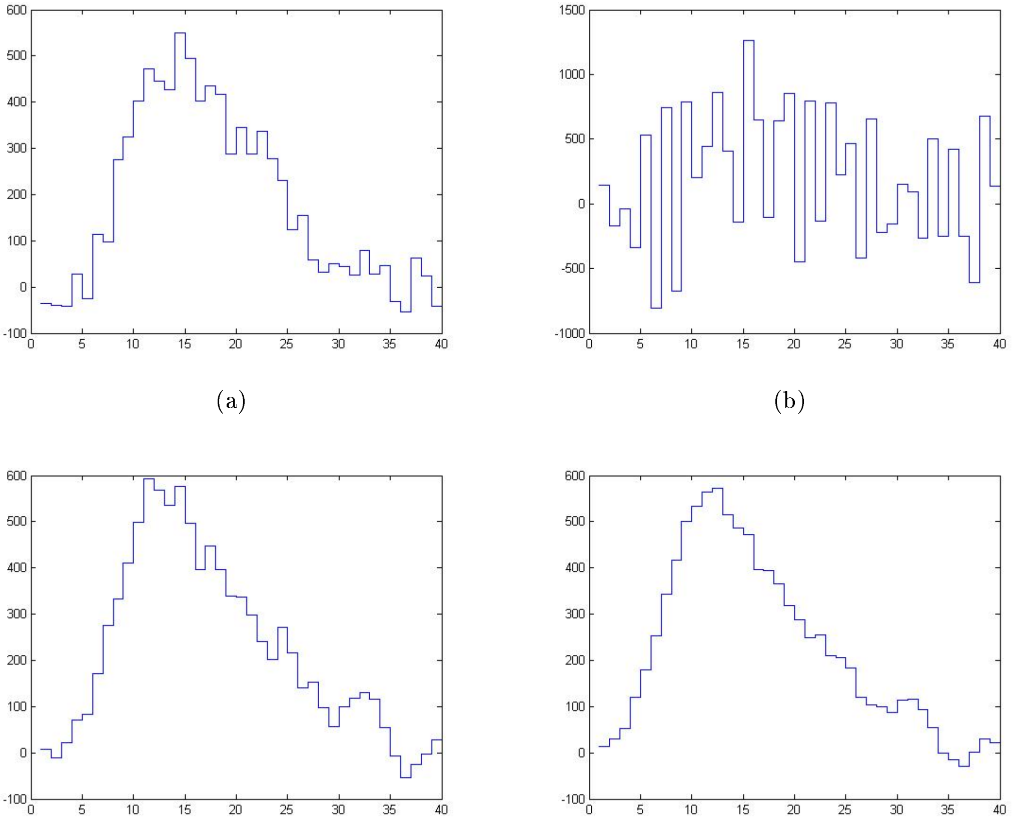
(a)

(b)

Figure 4.5.: Time resolution of different time pick-off algorithms with (a) random noise and (b) realistic detector and random noise. Lowest measurable time resolution for the setup is 3 ns.

all shaping times. This corresponds to the case where half of the events are assigned with correct timing. When one thinks on the principle of deconvolution, the reason for this time resolution becomes apparent. The deconvolution aims to restore the original signal. The original signal is however around 60 ns long and consists of multiple peaks with random relative heights. The pulse from a deconvolution filter would hence be at least two clocks long and have random form. In a way the deconvolution method sees the original signal too clearly and is more susceptible to irregularities in the signal.

The PWLF was only tested with one shaping time $\tau = 50$ ns and assuming ideal detector. It gave competitive results with the other methods. For low noise the time resolution was $\sigma_t = 2.4$ ns and for higher noise $\sigma_t = 9.6$ ns. The time resolution presumably could have been even better, if filter complexity was increased. The problem with PWLF was not in the time resolution. It had the potential to give the best time resolution from all the filters. The problem was complexity. Even the first order was so heavy that it would eventually limit the number of channels on the chip.



(a)

(b)

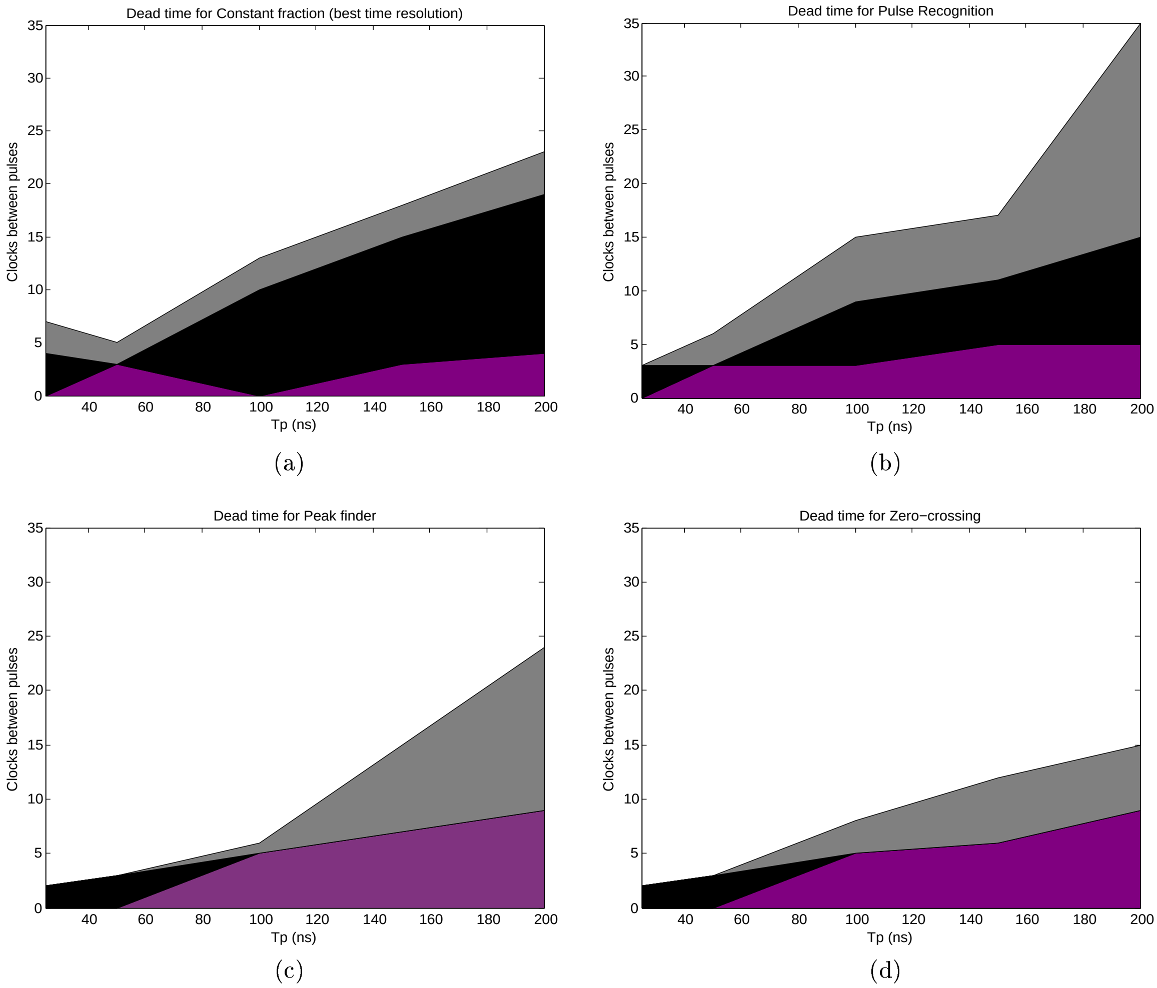
(c)

(d)

Figure 4.6.: Peak Sharpener, Integrator and noise. (a) Input signal. (b) Output from Peak Sharpener. (c) and (d) Output from Integrator. On x-axis is time and on y-axis ADC counts.

The peak sharpener in question was found to amplify noise too much to be useful as illustrated in Fig. 4.6. The ATLAS study [35] had found FIR peak sharpening slightly useful. For best filtering performance, the FIR-filter should be long enough to accommodate whole pulse with the longest shaping time. This would give at least 20 clock latency for the filter. The latency would be problematic, because it is added to the trigger latency.

The signal smoothing Integrator was found helpful with high level, high frequency noise. In Fig. 4.6 (c) and (d) the output signal from integrator is shown after first and second integration. CFD, PF and ZCI benefited from integrator in most cases. With PuR the benefits weren't as consistent and clear. In as many cases the integrator interfered with its operation as helped it.



(a)

(b)

(c)

(d)

Figure 4.7.: Time resolution and pulse pile-up. The distance between two equal pulses before timing is effected For (a) CFD, (b) PuR, (c) PF and (d) ZCI.

Fig. 4.7 illustrates the required delay between pulses. The gray color indicates region where the triggering efficiency is not reduced, but the timing is altered. Black color indicates dead time. The trigger efficiency is reduced. Only one pulse of the two is seen, but the timing is correct. Purple color marks a region where both triggering and timing efficiencies are reduced. Only one pulse is seen and the timing for it is altered. The minimum time between pulses for no effects from pulse pile-up was found to be shortest for ZCI and longest for PuR. The dead time was shortest for peak finder and longest for CFD. The effects seem to be more dependent on the shaping time than the BXID algorithm.

4.3. Comparison between digital and analog BXID methods

Two most promising methods in previous simulations CFD and Pulse Recognition were developed into Verilog models. The time resolution of the models was compared to analog methods using the same signal. The analog methods were investigated in Université Libre De Bruxelles.

4.3.1. Simulated GEM signal

The signal used in the simulations was obtained from Garfield simulations of the GEM detector conducted in Université Libre De Bruxelles (ULB) [36]. The simulation is for gas mixture Ar:CO_2:CF_4 (45:15:40) and contains 500 separate events for a MIP entering the GEM. The same signal was used in simulations using digital and analog time pick-off methods.

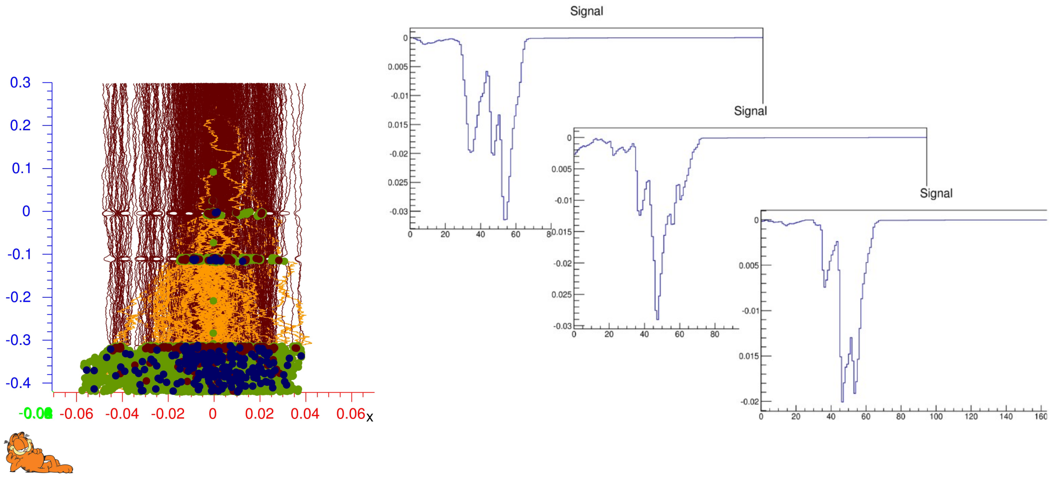


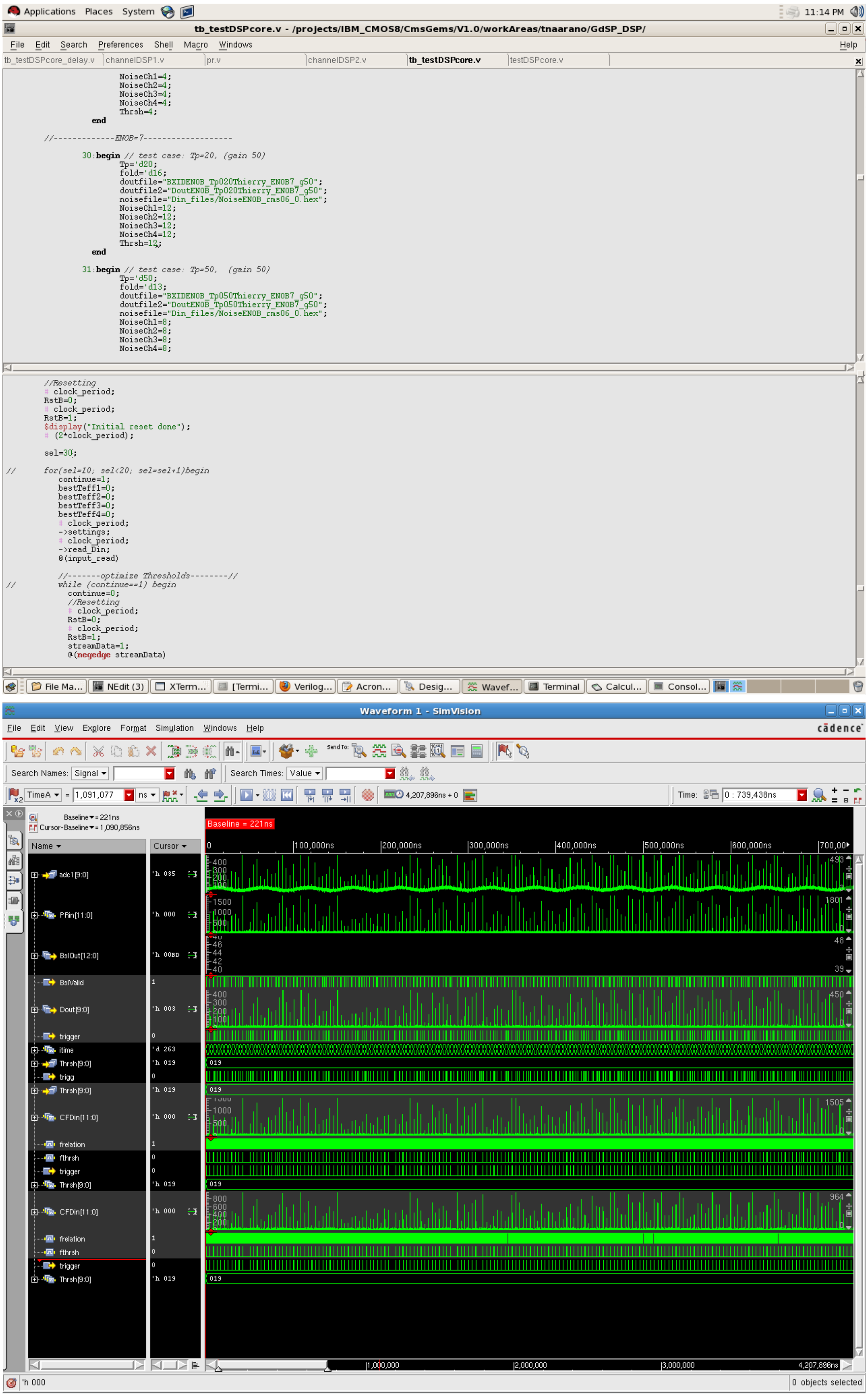
Figure 4.8.: Garfield simulations on GEM signal (Image source [36])

The ULB group shared the convoluted, unscaled signal. The Garfield simulation gives results in arbitrary units. The average signal height was scaled to correspond to a expected pulse height from a MIP. All events were multiplied with the same factor so that the deviations in the pulse height were maintained.

4.3.2. Simulation methods

Table 4.1.: Settings for best time resolution

Shaping time	Constant Fraction discriminator		Pulse Recognition			
[ns]	a	order of integration	n	v_1	v_2	v_3
25	4.43	1^{st}	2	1	0	-0.4
50	3.98	1^{st}	3	1	-0.4	-0.3
100	2.86	2^{nd}	5	1	-2.2	4.2
250	1.60	2^{nd}	9	1	-1.7	2.9
500	1.27	2^{nd}	9	1	-0.2	0



```
                NoiseCh1=4;
                NoiseCh2=4;
                NoiseCh3=4;
                NoiseCh4=4;
                Thrsh=4;
            end

    //------------ENOB=7------------------

            30:begin // test case: Tp=20, (gain 50)
                Tp='d20;
                fold='d16;
                doutfile="BXIDENOB_Tp020Thierry_ENOB7_g50";
                doutfile2="DoutENOB_Tp020Thierry_ENOB7_g50";
                noisefile="Din_files/NoiseENOB_rms06_0.hex";
                NoiseCh1=12;
                NoiseCh2=12;
                NoiseCh3=12;
                NoiseCh4=12;
                Thrsh=12;
            end

            31:begin // test case: Tp=50,  (gain 50)
                Tp='d50;
                fold='d13;
                doutfile="BXIDENOB_Tp050Thierry_ENOB7_g50";
                doutfile2="DoutENOB_Tp050Thierry_ENOB7_g50";
                noisefile="Din_files/NoiseENOB_rms06_0.hex";
                NoiseCh1=8;
                NoiseCh2=8;
                NoiseCh3=8;
                NoiseCh4=8;
```

```
    //Resetting
    # clock_period;
    RstB=0;
    # clock_period;
    RstB=1;
    $display("Initial reset done");
    # (2*clock_period);

    sel=30;

//  for(sel=10; sel<20; sel=sel+1)begin
      continue=1;
      bestTeff1=0;
      bestTeff2=0;
      bestTeff3=0;
      bestTeff4=0;
      # clock_period;
      ->settings;
      # clock_period;
      ->read_Din;
      @(input_read)

      //-------optimize Thresholds--------//
//    while (continue==1) begin
        continue=0;
        //Resetting
        # clock_period;
        RstB=0;
        # clock_period;
        RstB=1;
        streamData=1;
        @(negedge streamData)
```

Figure 4.9.: Screen capture of simulation using NCLaunch and SimVision.

The simulations were conducted using Cadence products NCLaunch and SimVision [38]. Verilog Hardware Definition Language (HDL) was used. SimVision was launched either from within NCLaunch or directly from terminal depending on the design complexity. Fig. 4.9 shows a screen capture of a simulation with NCLaunch and SimVision. Snippets of the test bench code in text editor are shown above and waveform data in SimVision window below. The uppermost waveform is the input signal to the DSP core and the first below it is the baseline corrected input signal to the BXID blocks.

The greatest difference between the Simulink models and Verilog models is that in the latter all variables, inputs and outputs have less precision and tighter range. In Simulink, when the required range of the variables was yet unknown, the variables were all of type double. Based on these previous simulations the ranges and precision for the variables needed to represent all the required values could be drafted. The values for variables used in the simulation are presented in Table 4.1. The hard coded ranges and precisions tie the filters to the designed range of shaping times (25 ns to 200 ns) resulting on poorer time resolution when using shaping times outside the range. Finding appropriate settings for PuR on 500 ns shaping time proved impossible. The variable range for n was too small. With shaping times 25 ns and 500 ns PuR is effectively used as simple peak finder.

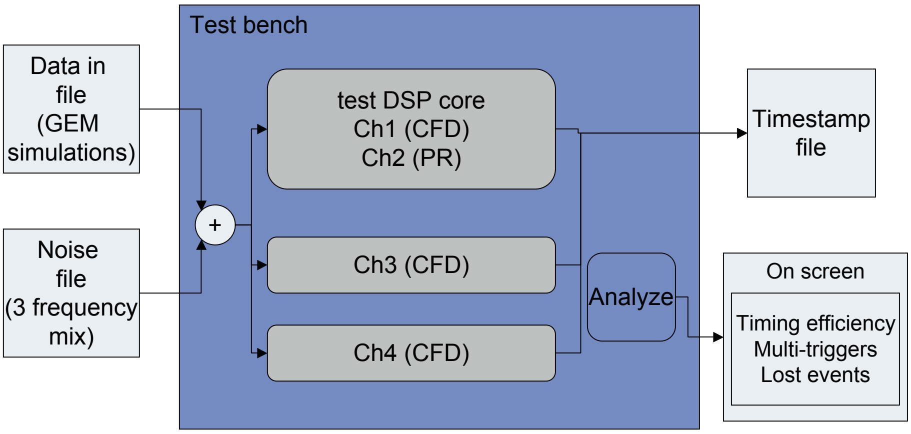
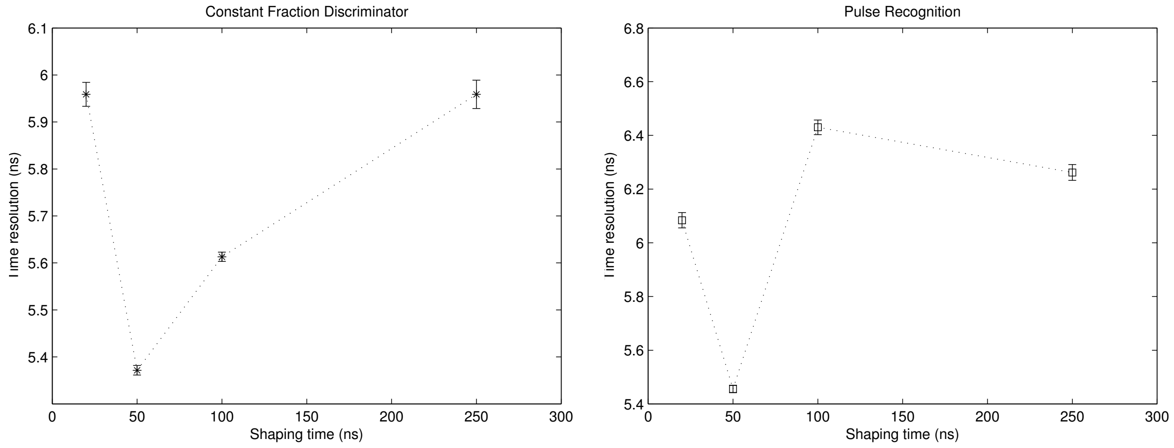


Figure 4.10.: Block diagram of the simulation test bench.

A block diagram of the simulation is shown in Fig. 4.10. For more realistic outcome the filters were used with the full DSP chain. A truncated test version of a DSP-core with two channels was created. The channels within the core used same settings. Additional two channels were used with different settings. The main difference in settings was the use of integrator. The input signals were read from input file. Optional noise could be read from separate noise file. At this point the noise was not used. Some approximated key results were displayed on screen at the end of the simulation. They were mainly used in the calibration (eg. fine-tuning the signal phase). The time resolution was calculated with Matlab using the output file for timestamps.

The simulations for the analog methods were conducted in ULB by Thierry Maerschalk under supervision of Gilles De Lentdecker. [36] They were based on models written in Python programming language. The simulations were conducted at this point without noise.

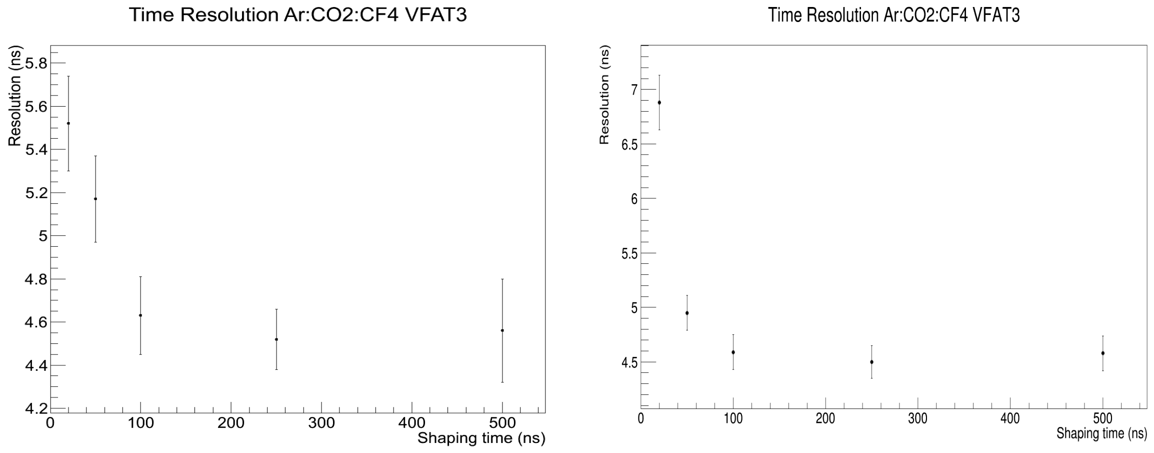
4.3.3. Simulation results



(a)

(b)

Figure 4.11.: Time resolution using digital (a) Constant Fraction Discriminator and (b) Pulse Recognition using Front-End gain 12,5 mV/fC without noise.



(a)

(b)

Figure 4.12.: Time resolution using analog methods a) Time over Threshold and b) Constant Fraction Discriminator without noise (Image from [36])

Fig. 4.11 shows the time resolution for digital methods and Fig. 4.12 for analog methods. With short shaping times the time resolution is larger, because the GEM signal is not fully integrated in the analog front-end. This results in distorted and irregular pulse shape. As the shaping time increases the pulses become more uniform. The time resolution from the analog methods follow this principle. With the digital methods this effect is seen with the shorter shaping times. With longer shaping times the calculations need to be increasingly accurate as the relative differences between sample values become smaller. The digital system also suffers from quantization noise. It may interfere with the finer calculations and increases the time resolution for longer shaping times. As a result the digital methods have optimal shaping time around 50 ns. For this shaping time the time resolution for digital and analog methods is almost the same. For digital methods it is around 5.4 ns and for analog methods around 5.2 ns.

4.4. Required Effective Number of Bits

The Effective Number of Bits (ENOB) required for good time resolution is an important specification for the ADCs. The ENOB is also discussed in section 2.3 on page 12. It is assumed that the ADC input and output ranges are perfectly matched and the ENOB can be expressed as random noise. The effect of ENOB was assessed in three different ways: by analytical calculation, using differential pulse height spectrum and comparing the time resolution with different noise levels.

The primary, central channels closest to the incident particle collect the most charge. A much smaller signal can be seen on the neighboring channels. The signal on neighboring channels can be used to get better spatial resolution. The effects of the ENOB are assessed also for the neighboring channels.

4.4.1. Analytical Estimation

The noise resulting from ENOB N_{ADC} is added to the noise in analog front end N_{FE} giving the total noise

$$N_{tot} = \sqrt{N_{FE}^2 + N_{ADC}^2}$$

. The expected equivalent noise charge for the front-end is $N_{FE} = 1100\,e$. The chip has two options for amplifier gain: $g = 12.5^{mV}/_{fC}$ and $g = 50^{mV}/_{fC}$. Analytically one can calculate a value for ENOB for which the two noise sources are balanced: $N_{FE} = N_{ADC} \Rightarrow ENOB = 9.8$ for $g = 12.5\,^{mV}/_{fC}$ and $ENOB = 7.8$ for $g = 50\,^{mV}/_{fC}$. With these values for ENOB neither noise source dominates and in principle the ADC noise should not significantly increase the time resolution.

4.4.2. Differential Pulse Height Spectrum

A traditional way to quantify tolerable noise level is to take a differential pulse height spectrum from the signal. When a threshold can be set between the peaks representing the noise and signal, the noise is tolerable.

The simulated GEM signal has approximately 1 to 5% overlap with the front-end noise alone. A pulse height region without any pulses between the signal and noise cannot be found. The separation between the signal and noise does depend on ENOB as illustrated in Fig. 4.13 and 4.14. The applied threshold T in ADC counts is on the x-axis. On y-axis is the number of pulses N per a threshold interval $dT = 10$. This gives the number of pulses that have amplitude in the range from $T - dT$ to T.

The collected charge was estimated to be 14 fC on the primary channel and 6 fC on neighbors [27]. The pulse amplitude on primary channel would be then 358.4 ADC counts with $g = 50\,^{mV}/_{fC}$ and 153.6 ADC counts with $g = 12.5\,^{mV}/_{fC}$. The total RMS noise in ADC counts using $g = 50\,^{mV}/_{fC}$ was estimated to be 4.6 for ENOB=9, 6.0 for ENOB=7 and 16.6 for ENOB=5. Using $g = 12.5\,^{mV}/_{fC}$ it was estimated to be 1.5 for ENOB=9, 4.2 for ENOB=7 and 16.0 for ENOB=5.

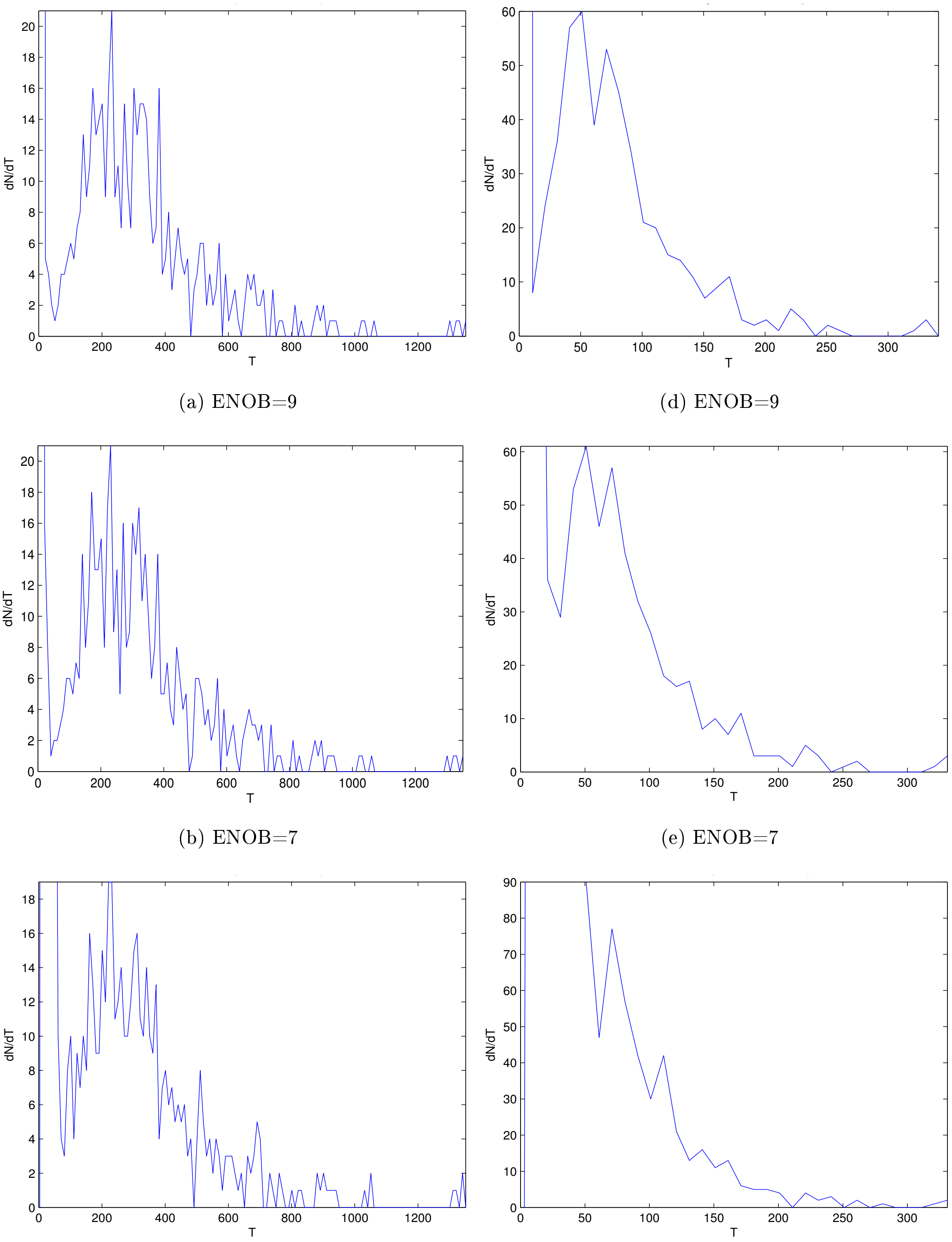
Figures with the differential pulse height spectra are found on pages 48 and 49. The minimum value for ENOB depends on the amplifier gain and the collected charge on the channel. The best separation between signal and noise is achieved on the primary channel with gain $g = 50\,^{mV}/_{fC}$. In this case they are separated even with 5 ENOB as can be seen in Fig.4.13 (c). The worst case is the neighboring channel with gain $g = 12.5\,{}^{mV}/{}_{fC}$. Here the signal is barely distinguishable from the noise even with 9 ENOB as seen in Fig. 4.14 (d).

4.4.3. ENOB and time resolution

The same, Verilog based test bench was used as when comparing the digital designs to analog. The test bench is shown in Fig. 4.10 on page 44. Pulse amplitude corresponding to primary channel was used. The noise was a mixture of three different frequencies. Modeling the noise with random numbers is likely to result in overestimation in the noise and in the time resolution. It is fully possible that time resolution measured in laboratory tests would be better.

Fig. 4.15 on page 50 shows the time resolutions dependency on ENOB for CFD and PuR for the primary channel. Fig. 4.15 (b) has exactly the expected form of approximate rectangular hyperbola. The time resolution approaches its optimal value when ENOB tends to infinity and begins to rapidly increase below 8 ENOB. With the greater gain (see Fig. 4.15 (a)) the time resolution begins to increase correspondingly below 7 ENOB. Unexpectedly the best time resolution for the different methods differs with over 1 ns. With the greater gain some clipping of the pulses was seen with the larger amplitudes. This occurrence was not estimated frequent enough to cause significant deterioration in the time resolution. When compared to the time resolution with lower gain, the time resolution with CFD is around the same value and with PuR it's improved. This gives a hint that PuR benefits from the increased accuracy in the calculations as the pulse is magnified.

For the neighbor channels it is expected that the base level of time resolution is higher resulting from the smaller S/N. Higher required ENOB is also expected.



(a) ENOB=9

(d) ENOB=9

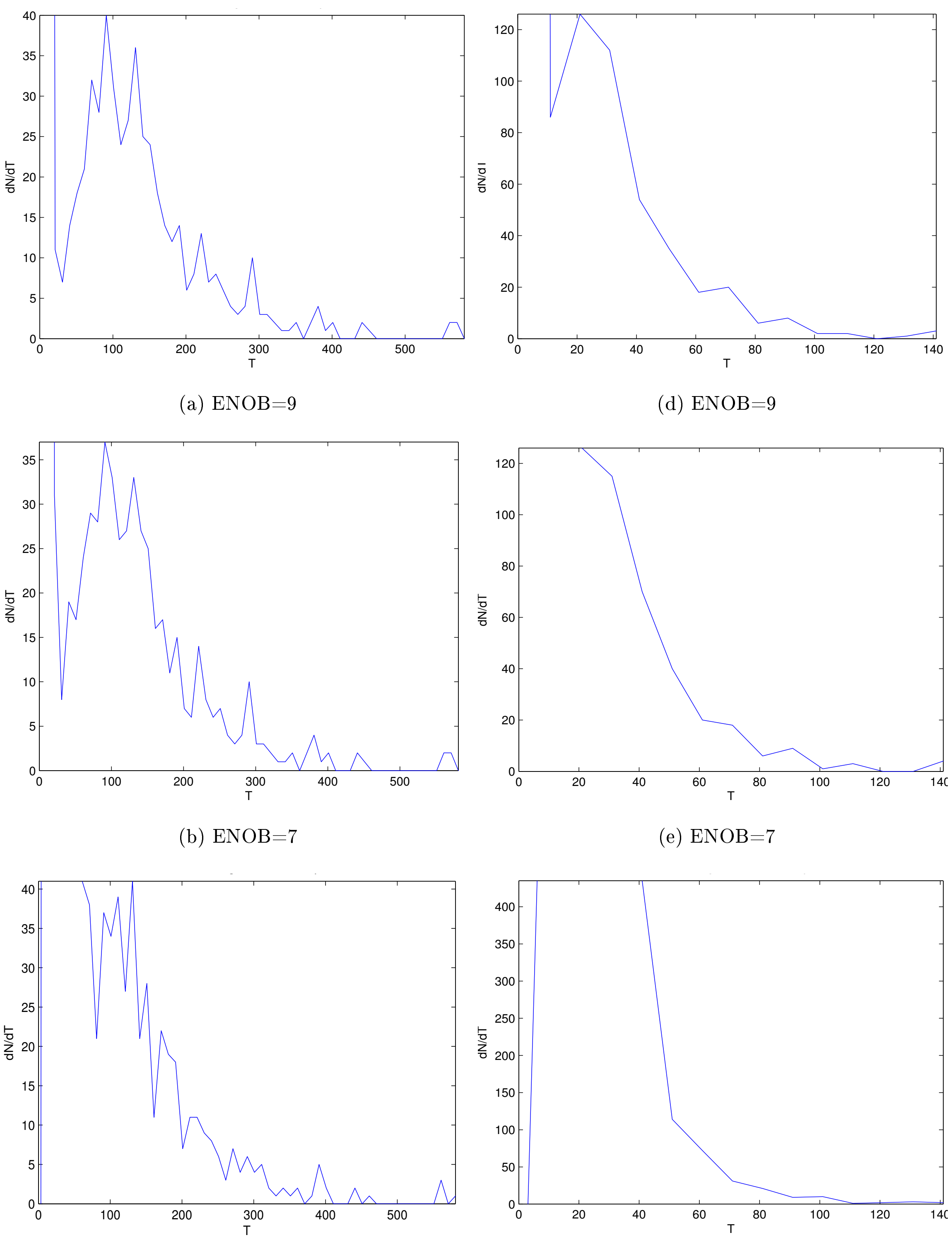
(b) ENOB=7

(e) ENOB=7

(c) ENOB=5

(f) ENOB=5

Figure 4.13.: Differential pulse height spectrum for primary channel for shaping time 100 ns and amplifier gain (a)-(c) $g = 50\, {}^{mV}/{}_{fC}$ and (d)-(f) $g = 12.5\, {}^{mV}/{}_{fC}$



(a) ENOB=9

(d) ENOB=9

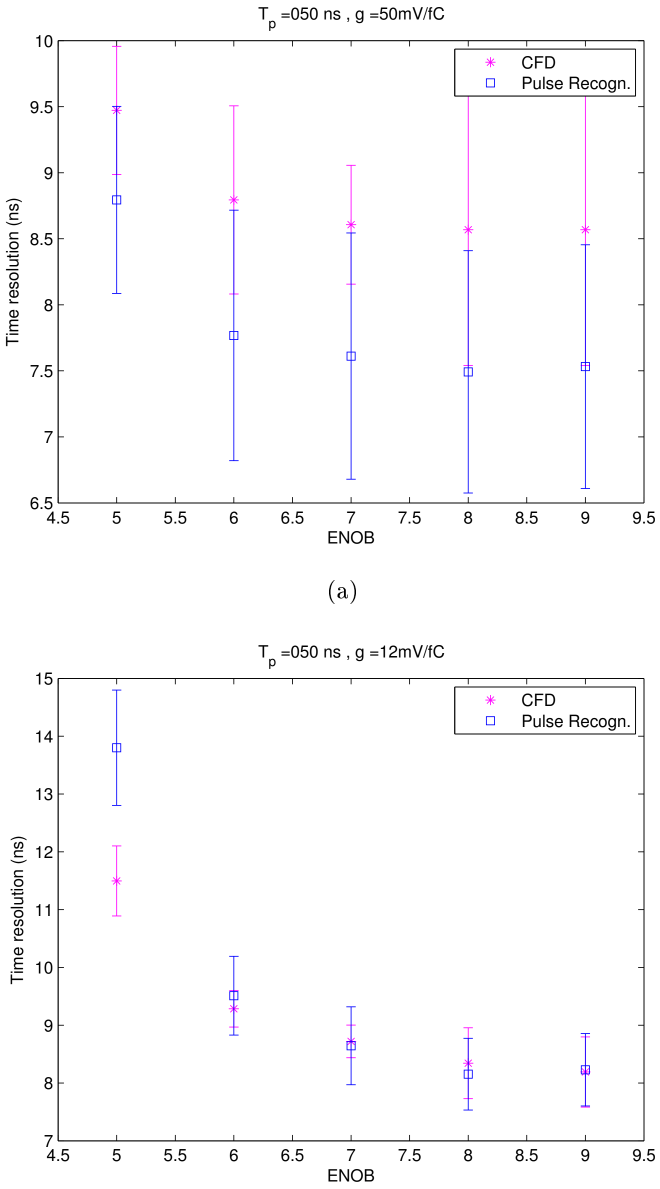
(b) ENOB=7

(e) ENOB=7

(c) ENOB=5

(f) ENOB=5

Figure 4.14.: Differential pulse height spectrum for neighboring channel for shaping time 100 ns and amplifier gain (a)-(c) $g = 50\,{}^{mV}\!/_{fC}$ and (d)-(f) $g = 12.5\,{}^{mV}\!/_{fC}$



(a)

(b)

Figure 4.15.: Time resolution with different ENOBs for amplifier shaping time $\tau = 50ns$ and gains (a)$g = 50\,{}^{mV}/_{fC}$ and (b) $g = 12.5\,{}^{mV}/_{fC}$.

5. Proposal for the GdSP Signal Processing Chain

5.1. Digital Signal Processing Core and Chain for one channel

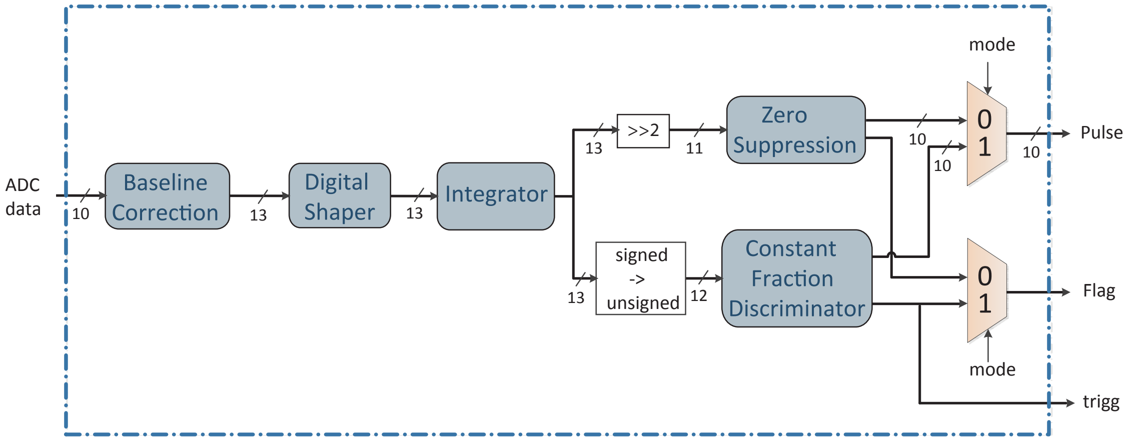


Figure 5.1.: Data path on the Digital Signal Processing chain.

The DSP core contains either 128 or 64 channels and a 10 bit counter. The counter is used to read the SRAM in BC time-wise. The DSP chain for one channel is illustrated in Fig. 5.1. For simplicity only the data path is shown.

The data input to DSP chain comes directly from an ADC. The data outputs `Pulse` and `Flag` go to SRAM, where it the data is delayed for the duration of CMS trigger latency. The data is formatted after the SRAM. The formed trigger signal `trigg` has direct route to the E-Port for readout.

The DSP has two modes of operation; Tracker and Waveform. Baseline Correction (BC), Digital Shaper and Integrator are used in same fashion with both modes. In the tracker mode the Zero Suppression (ZS) block is completely unused. The outputs come from Constant Fraction Discriminator (CFD). `Pulse` contains extracted pulse amplitude information. `Flag` is the trigger signal. The choice, whether amplitude information or binary data is read out, is made in data formatting block.

In waveform mode complete pulses are read out. In this mode `Pulse` contains the processed waveform data. The `Flag` marks meaningful data for readout. The CFD can be simultaneously used to produce trigger.

Many of the DSP blocks make use of a threshold value to distinguish signal from noise. These blocks are BC2, ZS and CFD. All three filters have independently adjustable threshold values. These values are common to all channels. The thresholds are added with component, that is adjusted separately for each channel. This NoiseCh variable indicates the noise level on the channel.

It is common practice to register the output from a filter. This prevents the bit rise time from causing instability. These register levels add to the latency of the filter. If shorter trigger latency would be needed, it could be investigated, if stable filters could be achieved without the registers.

The reset scheme for the DSP-chain includes two resets. One is active-low common reset “RstB” for all blocks. It clears all registers and calculated values to zero. The second reset `ma_rstB` is distributed trough the configuration registers. It resets exclusively the Moving Average Unit (MAU) in BC2. Both common reset and MAU reset are designed to be usable while the detector is on-line and taking data. The Table 5.1 lists the system inputs that are common to all DSP filters in the core.

Table 5.1.: System inputs common to all blocks in the DSP core.

System inputs			
I/O	Name	Width	Description
Input	Clk	1	40 MHz clock signal
Input	RstB	1	Active low reset

5.2. Baseline Correction

The Baseline Correction (BC) is divided into two blocks BC1 and BC2. These two blocks are both optional and can be bypassed to reduce latency (see Fig. 5.2). All tough they can be both bypassed it is recommended that at least one of them is used. All the input and output signal are summarized in table 5.2.

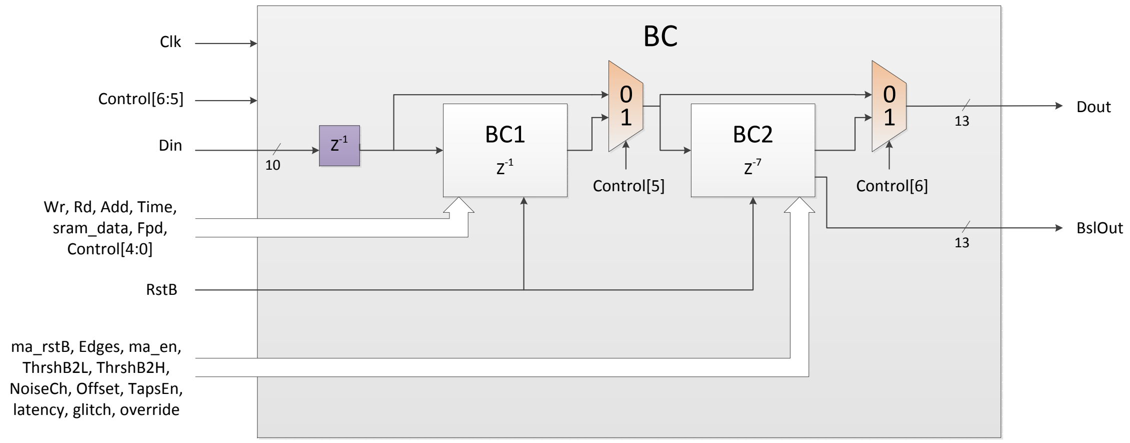

BC
Clk
Control[6:5]
Din
10
z^{-1}
BC1
z^{-1}
0
1
Control[5]
BC2
z^{-7}
0
1
Control[6]
13
Dout
13
BslOut
Wr, Rd, Add, Time, sram_data, Fpd, Control[4:0]
RstB
ma_rstB, Edges, ma_en, ThrshB2L, ThrshB2H, NoiseCh, Offset, TapsEn, latency, glitch, override

Figure 5.2.: Block diagram of Baseline Correction

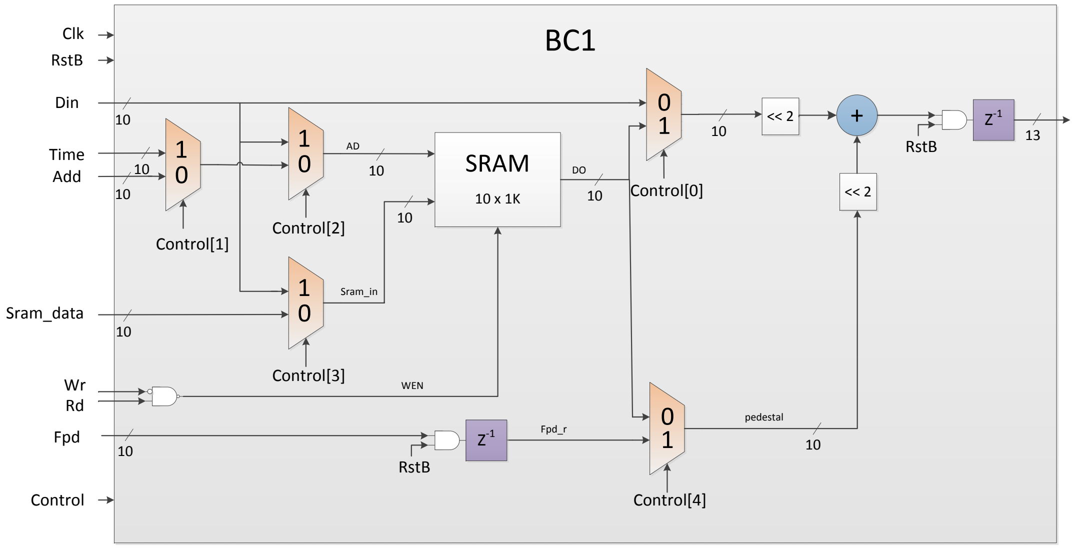


Figure 5.3.: Block diagram of Baseline Correction 1

The block diagram of BC1 is shown in Fig. 5.3. It is based on the BC1 in S-Altro ASIC. Features that were not compatible with continuous data taking were discarded in the migration process. The main features of BC1 are fixed pedestal subtraction and small 10x1024 SRAM. The filter offers multiple usage options. The options in BC1 are chosen through chip registers using the first five bits of input variable named `Control`. All possible permutations for `Control` might not make sense. The most useful options (and binary values for `Control`) are

- Fixed pedestal subtraction from input signal (`Control`=1xxx0)
- Test mode: Signal is read from SRAM and fixed pedestal is subtracted (`Control`=1x011)
- Conversion mode: SRAM is used as LUT for input and fixed pedestal is subtracted (`Control`=1x1x1)
- Subtraction of periodic disturbances written to SRAM (`Control`=0x010)
- Subtracting converted signal: using SRAM as LUT in signal→baseline conversion (`Control`=0x1x0)
- Recording input signal to SRAM (`Control`=x101x)
- Writing data to SRAM trough control register interface (`Control`=x000x)

The BC1 can be a powerful tool in case of systematic periodic disturbances or other systematic distortions. The fixed pedestal subtraction offers a foolproof baseline correction, when low frequency baseline shifts are not a significant problem.

The BC2 filter subtracts a self-calibrating baseline. The filter is based on S-Altro BC2, which in turn is unchanged version from ALTRO ASIC. In adaptation to GdSP the filter was updated significantly based on feedback. The block diagram of the filter can be found in Fig. 5.4 on the next page. As seen from the figure, the double threshold scheme and moving average calculations control logic are a considerable part of the filter.

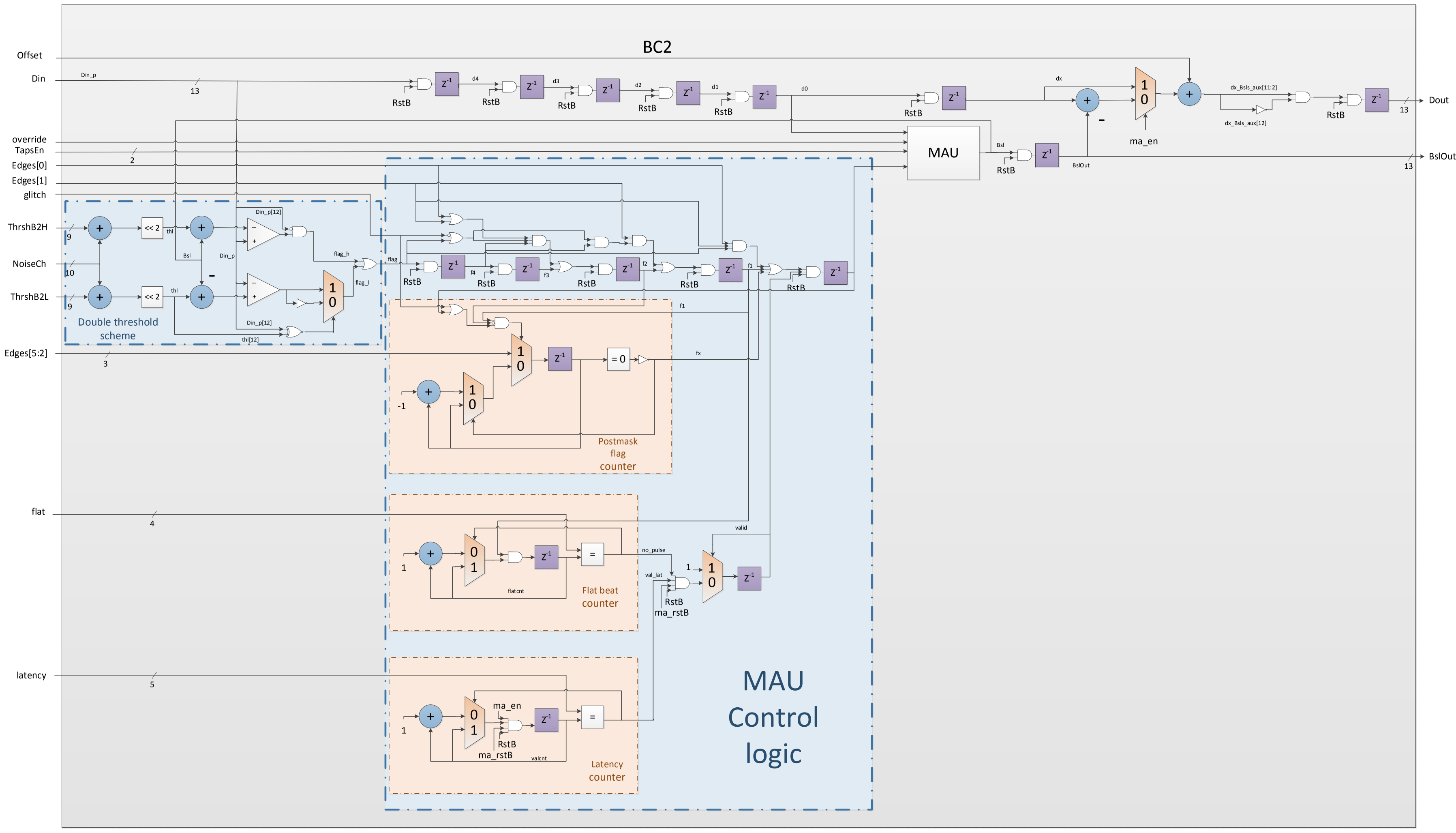


Figure 5.4.: Block diagram of Baseline Correction 2

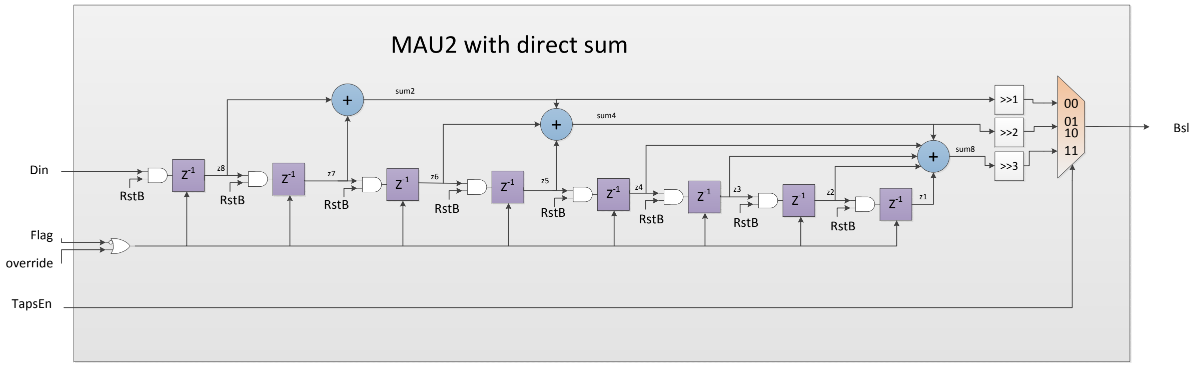


Figure 5.5.: Block diagram of Moving Average Unit.

The double threshold scheme excludes pulses from the baseline calculation. The input signal is compared to high and low thresholds relative to calculated baseline. If the signal is outside the threshold limits, the baseline calculation is suspended. The calculation resumes when the signal returns inside the threshold limits. Unlike in S-Altro the usage of the threshold scheme is optional. The risks of the double threshold scheme were discussed in the context of S-Altro BC and simulation results (see sections 3.2 on page 18 and 4.1 on page 34). The downside of threshold-less calculation is that the pulses will be followed with undershoots resulting in dead time as discussed in section 3.2. The bright side is that the calculation is extremely predictable and stable as experienced in use in ALICE EmCal trigger boards [37].

The control logic contains latency counter, postmask counter and flat beat counter. The latency counter lets the registers in MAU fill after reset before implementing the double threshold scheme. The postmask counter keeps the baseline calculation suspended for the specified time even after the signal has returned inside the threshold limits. The flat beat counter makes certain that there are no pulses present in the signal at the moment when the double threshold scheme is implemented after reset. It keeps counting up as long as the signal is within threshold limits. If the signal is not within the limits, the counter is reset. When the counter reaches a programmable value, it raises a flag signifying that no pulses are present in the signal.

The baseline calculation is based on Moving Average Unit (MAU) that is illustrated in Fig. 5.5. It is the Finite Impulse Response filter with direct sum described in section 3.2. The MAU calculates the moving average over two, four or eight samples. The baseline is subtracted from the signal, if the filter is enabled.

Table 5.2.: Baseline Correction inputs, outputs and variables from configuration registers.

Data in and out			
I/O	Name	Width	Description
Input	Din	10	Data input from ADC. Unsigned.
Output	Dout	13	Data output to next block. Signed two's complement.

Register variables			
I/O	Name	Width	Description
Input	Control	7	Selects the data path in the Baseline Correction.
Input	Add	10	Address for the SRAM in BC1.
Input	Fpd	10	Fixed pedestal value.
Input	Sram_data	10	Data to be written to the SRAM in BC1.
Input	Rd	1	Read enable for the SRAM in BC1.
Input	Wr	1	Write enable for the SRAM in BC1.
Input	Edges	6	Sets the time before and after a pulse, when the baseline is frozen.
Input	flat	4	Length of time period after reset during which there are no pulses present before threshold scheme is turned on.
Input	glitch	1	Prevents short glitches from freezing baseline calculation.
Input	latency	5	Latency after reset before threshold scheme is turned on.
Input	NoiseCh	10	Noise per channel is added to the thresholds.
Input	override	1	Turns the threshold scheme off.
Input	TapsEn	2	Selects the number of samples in the moving average calculation.
Input	ThrshB2H	9	High threshold. When signal is above this threshold, baseline calculation is frozen.
Input	ThrshB2L	9	Low threshold. When signal is below this threshold, baseline calculation is frozen.
Output	BslOut	13	Baseline that is being removed from the signal.

Other			
I/O	Name	Width	Description
Input	Time	10	Output from a counter on DSP-core level. Used for reading or writing the SRAM in BC1 time wise.

5.3. Integrator

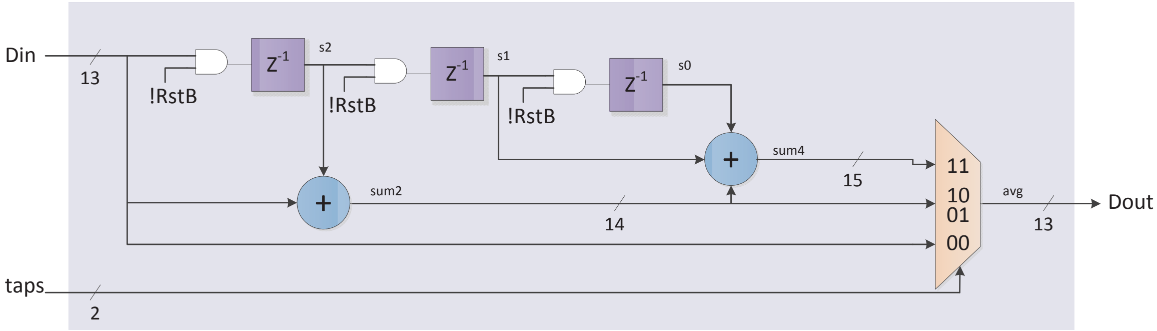


Figure 5.6.: Block diagram of Integrator.

The integrator is used to filter high frequency noise that is close to sampling frequency. It takes the average of two or four sequential samples. These correspond to the numerical integral and second order numerical integral over time covering one sampling period.

The block diagram of the integrator is shown in Fig. 5.6. The input Din is saved into short pipeline containing registered samples s2, s1 and s0. The input taps is used to select the signal for output Dout. The signal can be directly routed from the input, when there is no wish to use the integrator. Other options are the sum of input and first sample in the pipeline and a sum where the last two samples in the pipeline are added to the sum. The division by two or four is performed by omitting LSB(s) in the selection.

The integrator is needed for obtaining good time resolution in the bunch crossing assignment, when there is noise present. The importance of it grows with the shaping time. The process shifts the signal frequency spectrum into a lower frequency and reduces pulse height as a side effect. The severity of these side effects depends on the shaping time in the analog shaper. With long shaping times, the effects are barely noticeable. Without the noise treatment the Constant Fraction Discriminator will give poorer time resolution and is likely to produce false triggers on noise even with moderate level of noise. In the other hand with very low level of noise integration will have adverse effect on time resolution. The integrator has been made programmable so that the optimal level of integration may be chosen.

Table 5.3.: Integrator inputs, outputs and variables from configuration registers.

Data in and out			
I/O	Name	Width	Description
Input	Din	13	Data input from previous block. Signed two's complement.
Output	Dout	13	Data output to next block. Signed two's complement.

Register variables			
I/O	Name	Width	Description
Input	taps	2	Selects whether integration is off, first order or second order. Corresponding values for taps are 00, (01 or 10) and 11.

5.4. Digital Shaper

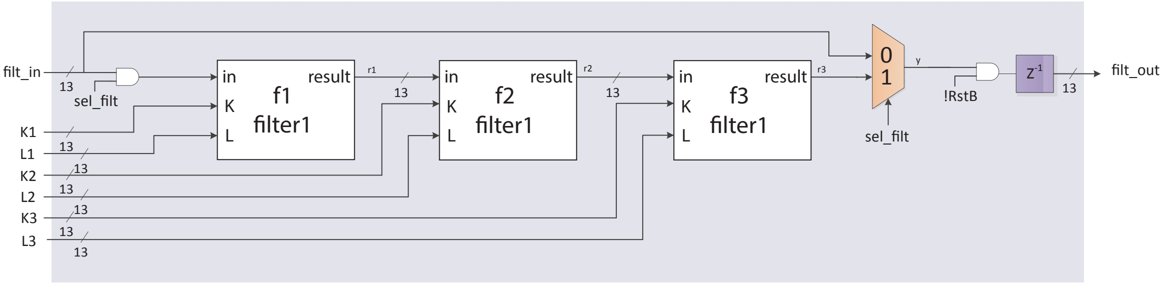


Figure 5.7.: Digital Shaper block diagram

The Digital Shaper (DS) is a 3rd order pole-zero filter (see Fig. 5.7). It is adapted from S-Altro with only one change. It was downscaled from 4th order cascade to 3rd order. The transfer function of the filter was derived in chapter 2.4 on page 13:

$$H(z) = \frac{1 - L_1 \cdot z^{-1}}{1 - K_1 \cdot z^{-1}} \cdot \frac{1 - L_2 \cdot z^{-1}}{1 - K_2 \cdot z^{-1}} \cdot \frac{1 - L_3 \cdot z^{-1}}{1 - K_3 \cdot z^{-1}} \tag{5.1}$$

The three zeros $\mathrm{L_i}$ and poles $\mathrm{K_i}$ adjust the passband of the filter. As a rule of the thumb, when the value for poles is increased, the signal tail is shortened and when the value for zeros is increased the peaking time is increased.

The pole-zero filter in Fig. 5.8 is a type of transposed direct form filter. The optimal form of the filter has been carefully studied for S-Altro and the probability of overflow has been reduced compared to ALTRO [20].

The values for zeros $\mathrm{L_i}$ and poles $\mathrm{K_i}$ can be within the range [0,1[. As the Verilog language does not support floating point arithmetics, the variables need to be expressed as integers.

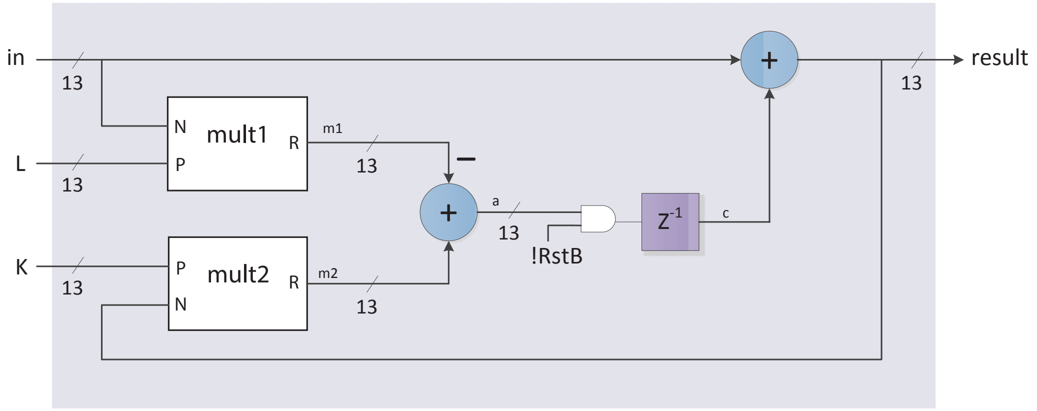


Figure 5.8.: Block diagram of Pole-zero filter in DS.

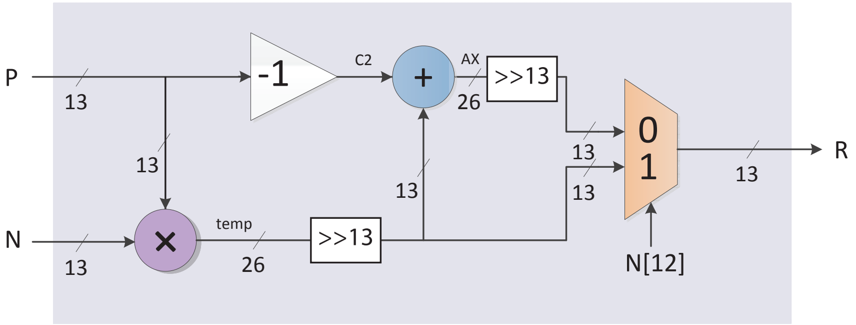


Figure 5.9.: Multiplication in DS.

The trick is to use integer variable in the arithmetic operation. The result is then divided using bit shifting operation. In the multiplication block (Fig. 5.9) the signal and variable (K or L) are first multiplied and then the 13 LSBs of the result are shifted away. As the variables K and L are 13 bits wide, the result is the same as multiplying with floating point number that has a positive value below one. Strictly speaking the variables $K_{i,var}$ in the filter are not the same as the poles $K_{i,eq}$ in the equation 5.1. They follow the relation $K_{i,eq}=K_{i,var}/2^{13}$. The same applies to the zeros.

It is debatable whether digital shaper is even needed in GdSP. The GEM detector does not suffer from long ion tails as the TPC. Signal that has peaking and decay times of the same order of magnitude has been proven challenging for digital shaping. One of the ideas for DS has been to use it to shape signal directly from the preamplifier. In any case the signal would benefit from shortened pulses.

Table 5.4.: Digital Shaper inputs, outputs and variables from configuration registers.

Data in and out			
I/O	Name	Width	Description
Input	filt_in	13	Data input from previous block. Signed two's complement.
Output	filt_out	13	Data output to next block. Signed two's complement.

Register variables			
I/O	Name	Width	Description
Input	sel_filt	1	Enable filter
Input	L1	13	First order zero
Input	L2	13	Second order zero
Input	L3	13	Third order zero
Input	K1	13	First order pole
Input	K2	13	Second order pole
Input	K3	13	Third order pole

5.5. Constant Fraction Discriminator

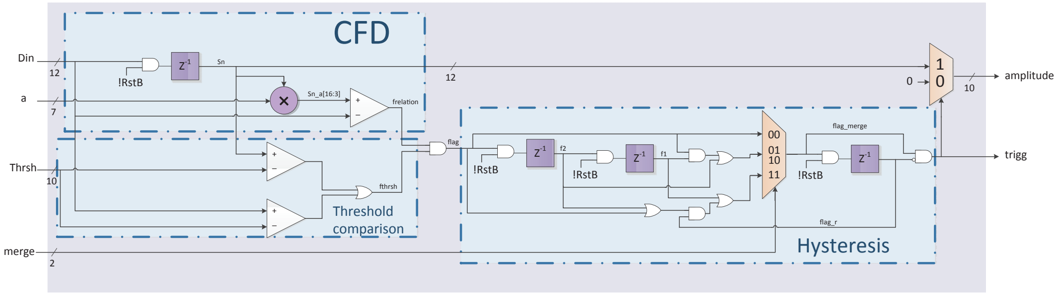


Figure 5.10.: Constant Fraction Discriminator block diagram. The filter consists of the actual CFD logic, threshold comparison that separates the signal from noise and flag merging that adds hysteresis to the comparators.

The Constant Fraction Discriminator (CFD) in Fig. 5.10 contains relational comparison between two sequential samples, threshold comparison and a hysteresis block. The relational operation is the core of the CFD. The comparison is true, at a certain fraction of the pulse height. The fraction can be adjusted with variable a. The idea behind the filter was discussed in depth in chapter 3.4.4.

The multiplication with variable a follows the same technique for introducing decimal points that was described for the DS multy block on page 59. The variable has effectively maximum value 15.875 and minimum step 0.125. In other words it has approximately range from zero to 16 with one decimal digit precision. This was the minimum requirement for longer shaping times. The digital noise would be reduced, if more decimals were added. The decimal digits are needed with the longer shaping times. In this region the filter is very sensitive to noise on the signal. The benefits of reduced digital noise would most likely be overshadowed by the sensitivity to other noise sources.

The threshold comparison discerns signal from noise. Requiring that both the two samples have the right relations and the signal is above threshold ensures that trigger is given only on meaningful signal.

On a threshold crossing there tends to be jitter when enough noise is present. Using hysteresis to eliminate the jitter is common practice in analog electronics. This kind of hysteresis is emulated by merging small gaps in the flag. The flag given by the comparison of two samples stays up from the point when the condition for slope is satisfied until the end of the pulse (when the effects of noise are eliminated). The threshold comparison flag is up always when the signal is above the threshold. Combining these flags with AND gives a flag that goes up, when the condition for pulse slope is satisfied, and returns down, when the signal goes below threshold. Any gaps in the flag are caused by noise. These gaps will cause false triggers if they are not treated. Merging short gaps in the flag was perceived as an effective way to eliminate the noise induced triggers. After the merging the resulting flag is clipped to one clock long trigger signal.

The amplitude given by the CFD is only directional. Amplitude information was not foreseen to be used for the GEM detectors and amplitude extraction is only an uncalibrated byproduct of the filter. The amplitude is proportional to the pulse height and should be adequate for center of gravity calculation for better spatial resolution. If precise amplitude information was desired[1] an additional amplitude extraction filter should be designed. For example weighted sum of three samples following the trigger might give the desired result.

In the simulations in section 4 CFD and Pulse Recognition (PuR) were almost identical in delivered time resolution and required resources on chip. By itself CFD is much lighter algorithm, but to operate reliably when noise is present it needs to be accompanied by an integrator. The only remaining justifications are the space required in configuration registers and the ease of use. CFD has only one parameter that needs to be calibrated. In contrast PuR has four of them. Even when the one variable needed for integrator is added, the number of parameters is still halved compared to PuR.

[1] Precise amplitude might be needed for applications such as calorimetry or energy loss measurements.

Table 5.5.: Constant Fraction Discriminator inputs, outputs and variables from configuration registers.

Data in and out			
I/O	Name	Width	Description
Input	Din	13	Data input from previous block. Signed two's complement.
Output	amplitude	10	Pulse amplitude information. Gives pulse amplitude simultaneously to trigger, otherwise zero.
Output	trigg	1	Trigger signal.

Register variables			
I/O	Name	Width	Description
Input	a	7	Relation between two sequential samples at the moment the trigger is given.
Input	Thrsh	10	Threshold between noise and meaningful signal.
Input	merge	2	Level of hysteresis applied to threshold crossing.

5.6. Zero-Suppression

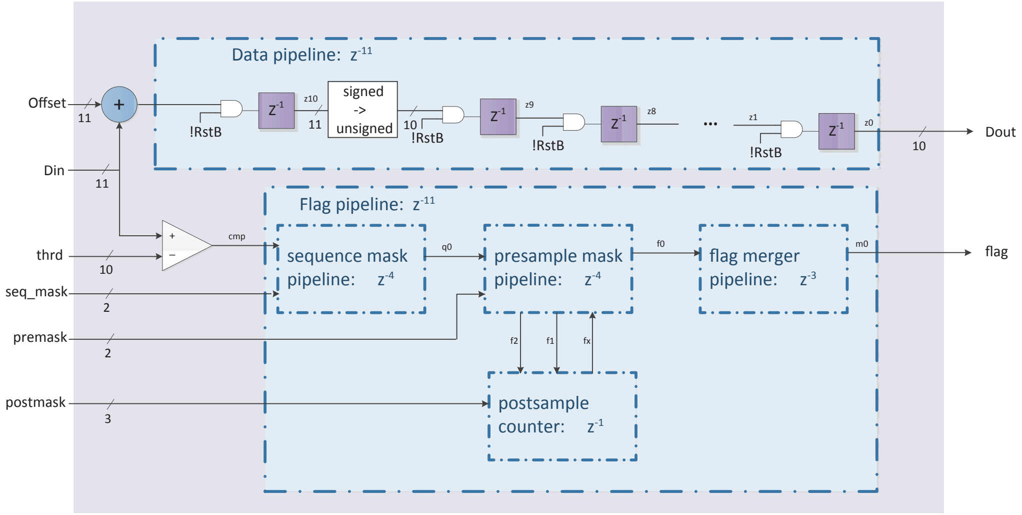


Figure 5.11.: Zero Suppression simplified block diagram.

The Zero Suppression (ZS) filter block diagrams can be found in Fig. 5.11-5.15. The filter has been extracted from S-Altro completely unchanged and is the same version as used in

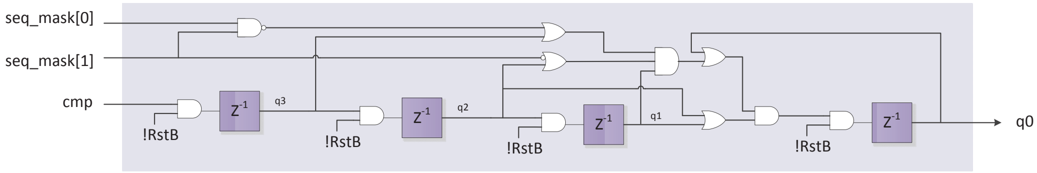


Figure 5.12.: Block diagram of sequence mask pipeline in ZS.

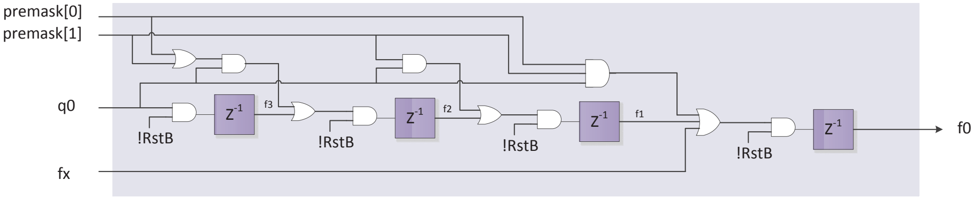


Figure 5.13.: Block diagram of pre-sample mask pipeline in ZS.

ALTRO. The reasoning behind zero suppression and the basic properties of the ZS were discussed briefly in chapter 3.5.

Two alternatives were considered for the filter outputs. The chosen option was delayed, but otherwise completely unaltered, signal data together with a flag marking the meaningful data. This methods gives more possibilities for the data formatter after the SRAM as all of the data is available. The other alternative was to have only one output that contained zero suppressed data. Outside the meaningful data the signal would be forced to zero. This method would need slightly smaller SRAM.

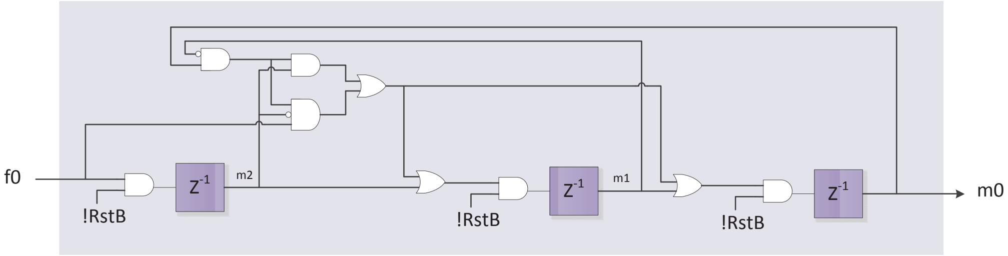


Figure 5.14.: Block diagram of Flag merger pipeline in ZS.

The ZS contains options for glitch filtering (sequence mask pipeline), use of pre- and post-samples and merging of two clusters. The principles of the flag formation were illustrated in Fig. 3.16 on page 33. The glitch filter (see Fig. 5.12) removes flag from pulses that have only 1, 2 or 3 samples above threshold depending on the value assigned to variable `seq_mask`. The pre-sample mask (see Fig. 5.13) adds samples to the flag before threshold crossing. Including pre-samples allows to read out values starting from the baseline value. The number of pre-samples is given by variable premask. The maximum number of pre-samples is three. The post-sample counter (see Fig. 5.15) adds maximum seven post-samples after the signal return below threshold. In addition to reading out the entire pulse the use of post-samples allows to catch possible undershoot after pulse. The flag merger (see Fig. 5.14) merges two flag

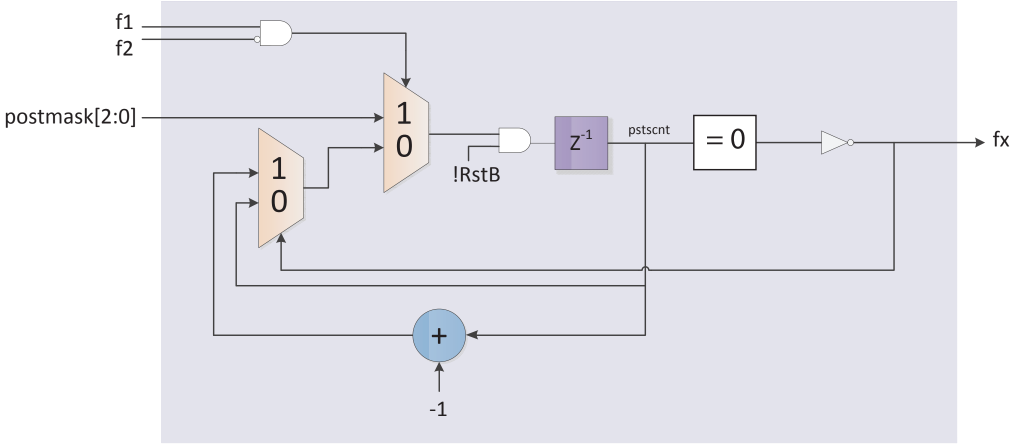


Figure 5.15.: Block diagram of Post-sample counter in ZS.

clusters if there are only one or two samples between them. In data formatting time tag and channel label are added to the data samples. The added labels are usually both 10 bit words. In stead of sending the extra labels, the readout of extra data samples is preferred when the number of communicated bits is the same.

Table 5.6.: Constant Fraction Discriminator inputs, outputs and variables from configuration registers.

Data in and out			
I/O	Name	Width	Description
Input	Din	11	Data input from previous block. Signed two's complement.
Output	Dout	10	Data out. Delayed unsigned data.
Output	flag	1	Flag marking meaningful signal.

Register variables			
I/O	Name	Width	Description
Input	Offset	10	Offset added to the signal before converting to unsigned.
Input	thrd	10	Threshold between noise and meaningful signal.
Input	seq_mask	2	Minimum number of samples above threshold for glitch filter.
Input	postmask	3	Number of post-samples flagged after returning below threshold.
Input	premask	2	Number of pre-samples flagged before threshold crossing.

6. Conclusions

As no front-end chip exists, which would exactly meet the CMS GEM requirements, a novel chip needed to be designed. Two approaches were considered: VFAT3 with analog signal processing and GdSP with both analog and digital signal processing (DSP).

The design of the DSP in the GdSP was greatly effected by the properties of the GEM signal. The irregularities in the signal shape lead to long shaping times in analog shaping. This in turn has to be taken into consideration in the DSP. Some of the requirements originate in the CMS experiment set-up. The experiment has to operate for long periods of time and it takes data continuously. This puts extra significance on delivering as sparse data as possible. It also increases the probability that the chip has to be reset while taking data. Resetting in these conditions should be part of normal operation. It should not cause, for instance, readout of disturbances in the signal, which were caused by the reset.

The DSP was designed to have two modes of operation. In tracker mode, only binary timing information and optional pulse amplitude are read out. In waveform mode, whole pulse is read out. The DSP required methods for baseline collection, digital shaping, noise filtering, zero suppression and time pick-off. For three of these methods, existing filter models could be used. These filters were migrated from S-Altro chip. Two of them were changed with varying degree in the adaptation process. Two filters needed to be designed from scratch. An integrator was used for high frequency noise filtering. For time pick-off, several different methods were considered. Constant Fraction Discriminator (CFD) proved to be the best alternative.

The migrated blocks were verified with simulations. Different simulation approaches were used to find the best time pick-off algorithm. The design of DSP-chain with all the filters was verified with simulations.

From time pick-off algorithms, CFD and Pulse Recognition (PuR) gave similar time resolution. When CFD is used with noise filtering and PuR without, CFD and PuR are of same size. The only difference is in number of variables, which is considerably higher for PuR. Piece-Wise Linear Fitting would be a suitable option, when high accuracy is needed and lower number of channels on chip is required.

The DSP sets high requirements for the ADC. The chip has minimum 64 channels and one ADC per channel. Due to the high number of channels the power consumption per channel needs to be low. In addition to the requirement for low power, the ADC needs to have high ENOB. Time resolution comparable to analog methods would demand ENOB above 9. An ADC with the combination of low power and high precision at the relatively high 40 MHz sampling do not presently exist for CMOS 130 nm process. Novel SAR-ADC meeting the criteria is being developed. At a checkpoint in 2013 the SAR-ADC had not reached sufficient ENOB with 40 MHz sampling frequency. The GdSP development was put on hold and the development was focused on VFAT3.

Acknowledgements

In addition to my supervisors and thesis reviewers several people have helped on the way.

I would like to thank the Technical Student programme and especially Laura Saulnier at CERN for making the internship possible. As I've understood CMS experiment is the correct party to thank for the funding.

The whole PH-ESE-ME section at CERN was helpful and eager to answer my questions. Special thanks go to Massimiliano De Gaspari for discussing the peculiarities of the S-Altro chip. Eduardo García, who designed the S-Altro DSP, was kind enough to visit CERN and answer my questions even though he no longer worked there.

Christian Lippmann was kind enough to share experiences with using ALTRO on the ALICE TPC. The feedback had a great impact on the design and hopefully made it much better.

I've asked for a wish list for front-end DSP from people I came into contact with during the design development mainly in CMS GEM collaboration and GEM users in University of Helsinki. Thank you for sharing your ideas and opinions.

My family and friends have been great at cheering throughout the writing process. Thanks for my fellow officemate at CERN Marko for teaching me the importance of cat memes and videos. And last but not least thanks to Henri Riihimäki, who has stood by me throughout the process.

Bibliography

[1] The CMS Collaboration (2008): “The CMS experiment at the CERN LHC,” JINST 3, S08004.

[2] M. Tytgat *et al.* (2013): ”Status of the Triple-GEM Project for the Upgrade of the CMS Muon System”, Proc. of MPGD2013, Zaragoza, Spain, July 2013

[3] A. Sharma *et al.* (2011): “An overview of the design, construction and performance of large area triple-GEM prototypes for future upgrades of the CMS forward muon system”, Proc. of MPGD2011, Kobe, Japan, Aug. 2011

[4] F. Sauli (1997): “GEM: A new concept for electron amplification in gas detectors”, Nucl. Instrum. Methods A386, 531-534

[5] A. Sharma (2012): “A GEM Detector System Upgrade of the High-η Muon Endcap Stations GE1/1 + ME1”, IV$^{\mathrm{th}}$CMS GEM Workshop, CERN, Nov. 2012

[6] G.F. Knoll (2010): *Radiation Detection and Measurement*, John Wiley & Sons

[7] B. Ketzer et al. (2001): "GEM detectors for COMPASS". IEEE Trans. Nucl. Sc. 48, 1065

[8] The TOTEM Collaboration (2008): ”The TOTEM Experiment at the CERN Large Hadron Collider “, JINST 3, S08007

[9] The LHCb Collaboration (2008):”The LHCb Detector at the LHC”, JINST 3, S08005

[10] M. Ziegler, P. Cwetanski and U. Straumann (June 1999): “A triple GEM detector for LHCb”, LHCb internal note TRAC 99-024

[11] CMS GEMs Collaboration (2012): “A GEM Detector System for an Upgrade of the CMS Muon Endcaps”, Technical Proposal, CMS IN 2012/001, CERN

[12] A. Marinov (2012): ”GEM PCB Development, GEM Tests & 904 Test Facility”, IV$^{\mathrm{th}}$CMS GEM Workshop, CERN, Nov. 2012

[13] P. Moreira et al., (2010): “The GBT SerDes ASIC prototype”, Published in J. Instrum. 5 C11016, presented at: Topical Workshop on Electronics for Particle Physics 2010. Aachen, Germany, Sep. 2010

[14] P. Vichoudis et al. (2010): “The Gigabit Link Interface Board (GLIB), a flexible system for the evaluation and use of GBT-based optical links”, Published in J. Instrum. 5 C110167 presented at TWEPP2010. Aachen, Germany, Sep. 2010

[15] Image source: http://en.wikipedia.org/wiki/File:Integrated_circuit_design.png, retrieved 13. march 2014

[16] R. Turchetta et al. (2001) “Design and results from the APV25, a deep sub-micron CMOS front-end chip for the CMS tracker”, Nucl. Instrum. Methods A 466, 359–365

[17] P. Aspell et al. (2007) ”VFAT2 : A front-end system on chip providing fast trigger information and digitized data storage for the charge sensitive readout of multi-channel silicon and gas particle detectors”, TWEPP2007, Prague, Czech Republic, Sep. 2007

[18] M. Alfonsi et al. (2007) ”Production and performance of LHCb triple-GEM detectors equipped with the dedicated CARDIAC-GEM front-end electronics”, Nucl. Instrum. Methods A572, 12-13

[19] W. Bonivento et al. (2005) ”Design and performance of the front-end electronics of the LHCb Muon Detector “, Presented at LECC, Heidelberg, Germany, Sep. 2005

[20] E. García (2012) “Novel Front-end Electronics for Time Projection Chamber Detectors”, Ph.D. Thesis, Universidad Politécnica de Valencia,

[21] B. Mota (2003) "Time-Domain Signal Processing Algorithms and their Implementation in the ALTRO chip for the ALICE TPC", Ph.D. Thesis, Ecole Polytechnique Fédérale de Lausanne

[22] J. Moroń (2013): ”Development of variable sampling rate low power 10-bit SAR ADC in 130 nm IBM technology”, TWEPP2013, Perugia, Italy, Sep. 2013

[23] ”Matlab, The Language of Technical Computing”, http://www.mathworks.se/products/matlab/, visited 23.8.2014

[24] ”Simulink, Simulation and Model-Based Design” http://www.mathworks.se/products/simulink/ , visited 23.8.2014

[25] D. K. Tala: ”Verilog tutorial”, http://www.asic-world.com/verilog/veritut.html, visited 1.6.2012 - 23.8.2014

[26] V. Radeka (2011): “Signal Processing for Particle Detectors”, in Schopper, H. and Fabjan, C. (ed.): *Elementary Particles, Subvolume B: Detectors for Particles and Radiation*, Springer

[27] P. Aspell (2012): “GEMs for CMS, From an electronics perspective”, Seminar, CERN, May 2012

[28] F. Guilloux (2012): “GdSP/VFAT3 ASIC, CFE analogue prototype”, presentation at GEMs for CMS Electronics meeting, CERN, Oct. 2012

[29] A.V. Oppenheim, R.W. Schafer (1989): *Discrete-Time Signal Processing*, Prentice Hall

[30] W. Kester (2003): *Mixed-signal and DSP Design Techniques*, Elsevier Science

[31] T. O’Haver: ”Resolution enhancement (Peak Sharpening)”, http://terpconnect.umd.edu/~toh/spectrum/ResolutionEnhancement.html, visited 25.4.2014

[32] V. Buzuloiu (1992): “A fast and precise peak finder for the pulses generated by future HEP detectors”, Proc. CHEP’92 2, 827-831, Sep. 1992

[33] P. Bloch and E. Tournefier (1999): “BC assignment and charge reconstruction with voltage sampling Preshower electronics”, Preshower Internal document, CERN, March 1999

[34] S. Gadomski et al (1992): “The deconvolution method of fast pulse shaping at hadron colliders”, Nucl. Instrum. Methods Phys. Res. A320, 217-227

[35] I. Brawn et al (1995): “Bunch-Crossing Identification for the ATLAS First-Level Calorimeter Trigger”, 293-296. 18. CERN-LHCC-95-56, Oct 1995

[36] Th. Maerschalk, G. De Lentdecker, G. Mullier (2013): ”Timing Resolution Techniques - TOT and CFD - and Fast Simulation”, VI$^{\text{th}}$CMS GEM Workshop, CERN, May 2013

[37] J. Kral (2012):”L0 trigger for the EMCal detector of the ALICE experiment”, Nucl. Instrum. Methods A693, 261-267

[38] ”CADENCE NCLAUNCH TUTORIAL”, http://www.ee.virginia.edu/~mrs8n/cadence/nclaunchtu
visited 18.8.2014

Appendices

A. Acronym Glossary

ADC Analog to Digital Converter.

ALICE A Large Ion Collider Experiment.

ALTRO ALICE TPC Read Out. Signal processing and read out front end chip developed for ALICE's TPC-detector. Used extensively throughout ALICE.

AMC Advanced Mezzanine Cards

APV Analogue Pipeline Voltage mode.

ASIC Application Specific Integrated Circuit

ATLAS A Toroidal LHC ApparatuS experiment.

BC Baseline Correction. [1]

BX Bunch Crossing.[1]

BXID Bunch Crossing IDentification.

CARIOCA CERN and Rio Current-mode Amplifier.

CFD Constant Fraction Discriminator. A pulse time pick-off method with both analog and digital implementations.

CMS Compact Muon Solenoid.

COMPASS Common Muon and Proton Apparatus for Structure and Spectroscopy

CSC Cathode Strip Chamber.

CERN Conseil Européen pour la Recherche Nucléaire = European Laboratory for Particle Physics

DAQ Data Acquisition.

DSP Digital Signal Processing.

DT Drift Tube.

ENC Equivalent Noise Charge

ENOB Effective Number Of Bits.

FE Front-end.

FIR Finite Impulse Response.

[1] In literature sometimes BC refers to bunch crossing. Here for clarity BX is used for Bunch Crossing and BC for Baseline Correction.

FPGA Field Programmable Gate Array.

GBT GigaBitTransceiver.

GdSP Gas detector/digital Signal Processing.

GEM Gas Electron Multiplier.

GLIB Gigabit Link Interface Board.

HEP High Energy Physics.

HDL Hardware Description Language.

IIR Infinite Impulse Response.

LHC Large Hadron Collider.

LHCb Large Hadron Collider beauty.

LUT Look-Up Table.

LSB Least Significant Bit.

LTI Linear Time-Invariant.

LVDS Low-Voltage Differential Signaling.

MA Moving Average

MAF Moving Average Filter.

MAU Moving Average Unit.

MIP Minimum Ionizing Particle.

MSB Most Significant Bit.

PCB Printed Circuit Board.

PWLF Piece-Wise Linear Fitting.

VFAT Very Forward Atlas Totem. Microelectronics front-end chip for tracking and triggering. Developed for and used mainly in Totem.

rms Root mean square.

RPC Resistive Plate Chamber.

RTL Register Transfer Level

S-Altro Super-ALTRO. Microelectronics front end chip for amplifying, digitizing, processing and reading out detector data. Designed for gas detectors in mind. Updated version of Altro chip used in ALICE.

SAR Successive Approximation Register.

SINAD SIgnal to Noise And Distortion ratio.

SNR Signal to Noise Ratio.

S/N Signal to Noise Ratio.

SRAM Static Random Access Memory.

ToT Time over Threshold

TOTEM Total Cross Section, Elastic Scattering and Diffraction Dissociation.

TPC Time Projection Chamber.

TTC Timing, Trigger and Control.

uTCA Micro Telecommunications Computing Architecture. MicroTCA Electronics system for telecommunication. Micro refers to the smaller size of the system compared to its predecessors.

ZCI Zero-Crossing Identification

B. Block diagrams of S-Altro DSP filters

The block diagrams correspond the S-Altro prototypes Verilog code that was available. They differ in some parts from the block diagrams in the thesis of Eduardo Garcia [20].

B.1. First Baseline Correction

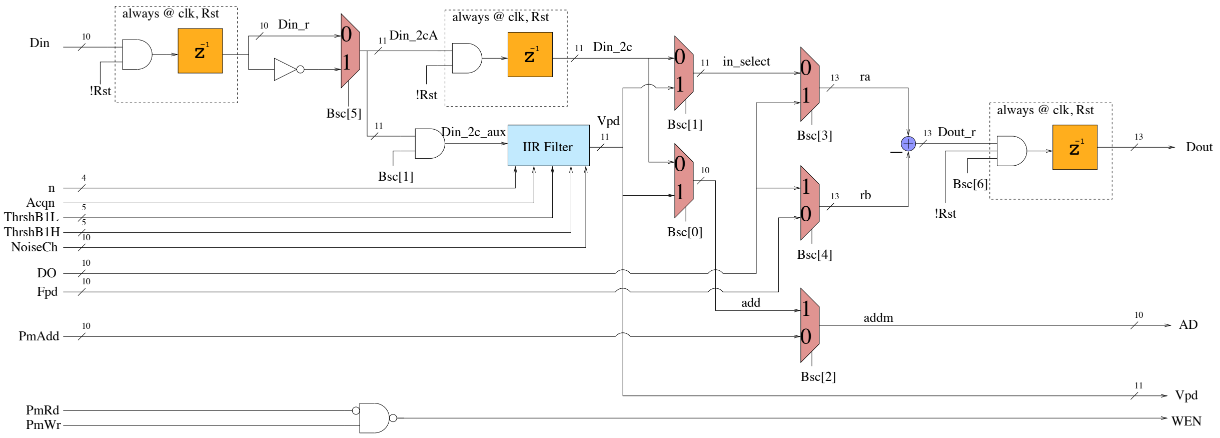


Figure B.1.: First Baseline Correction

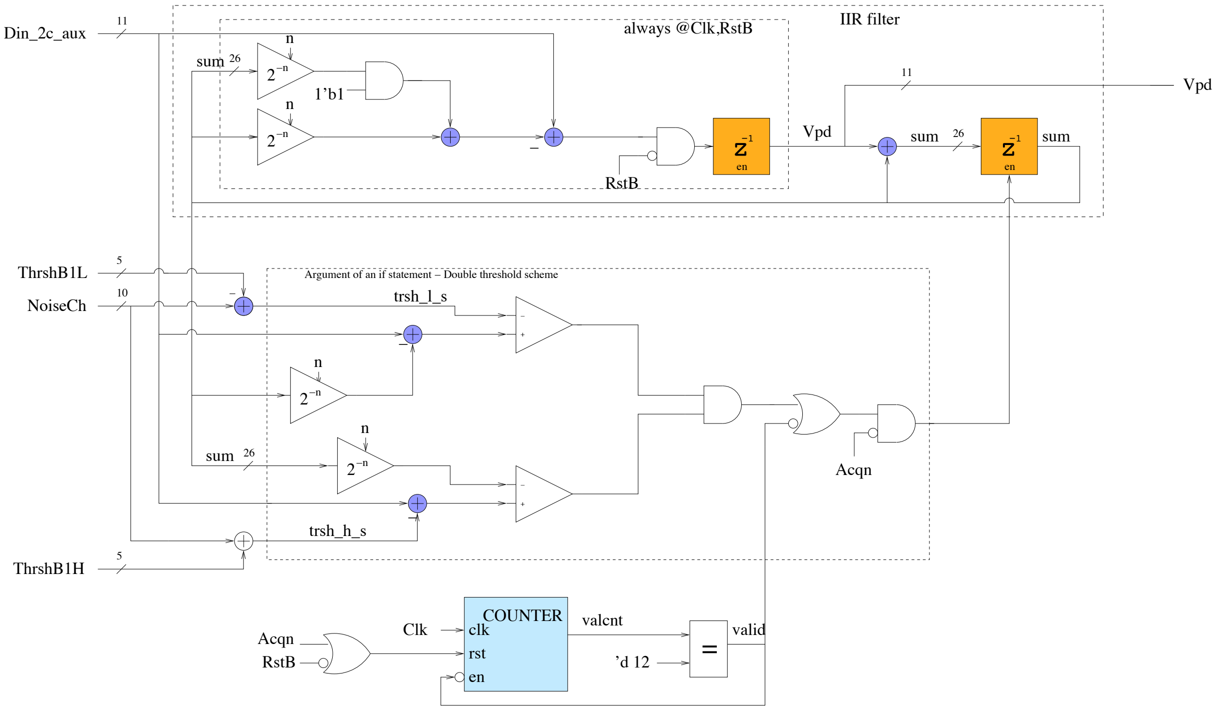


Figure B.2.: IIR filter

B.2. Digital Shaper / Tail Cancellation Filter

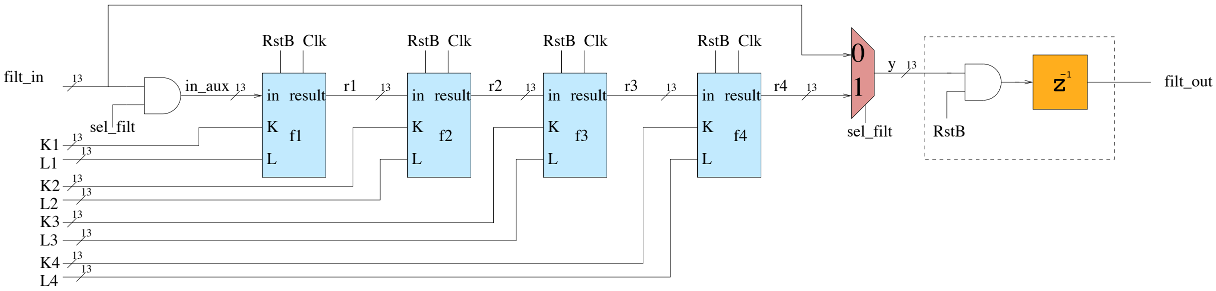


Figure B.3.: Digital Shaper. Cascade of four first order pole-zero filters.

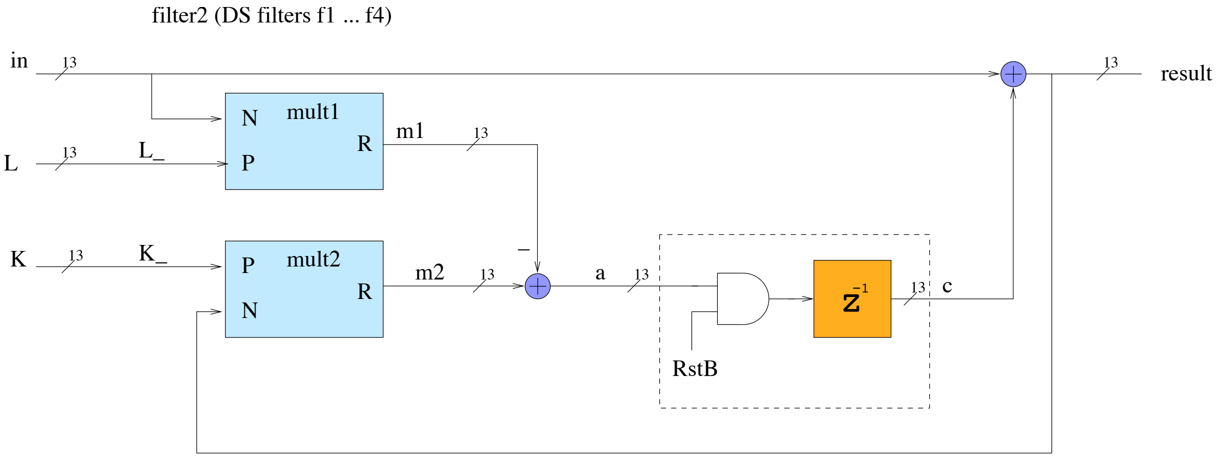


Figure B.4.: First order pole-zero filter used in digital shaper.

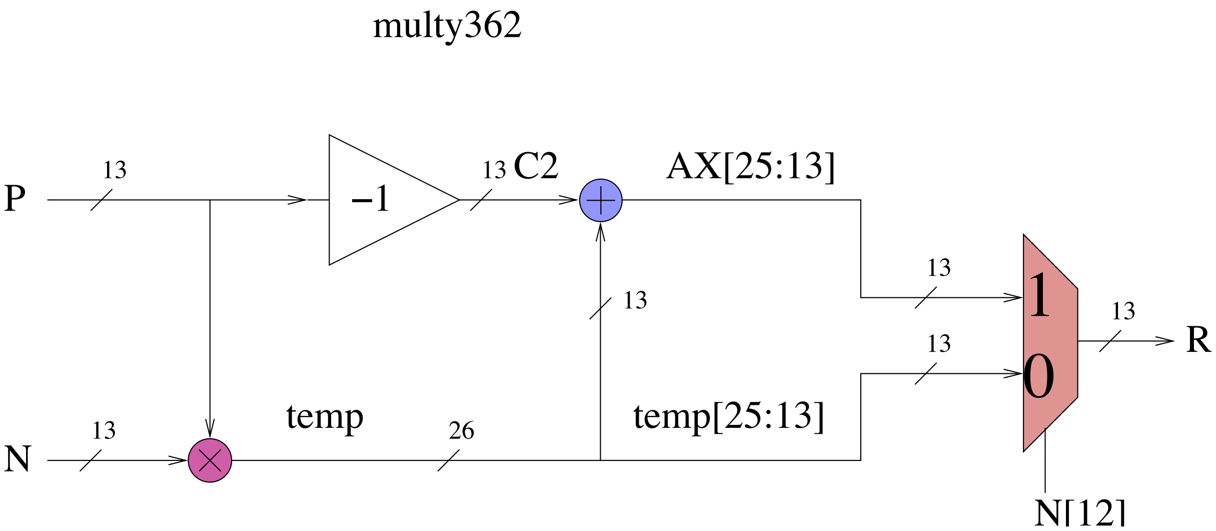


Figure B.5.: Custom multiplication operation used in Digital shaper. Input P is assumed to be always positive, whereas N can be positive or negative.

B.3. Second Baseline Correction

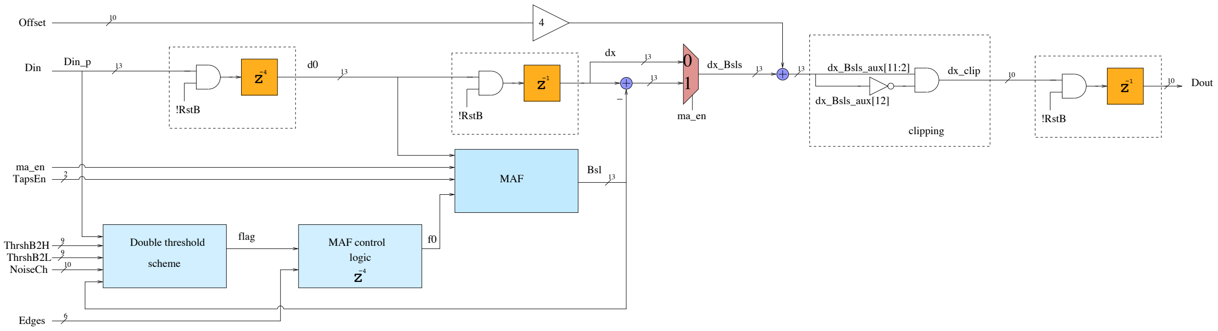


Figure B.6.: Second Baseline Correction abstracted block diagram.

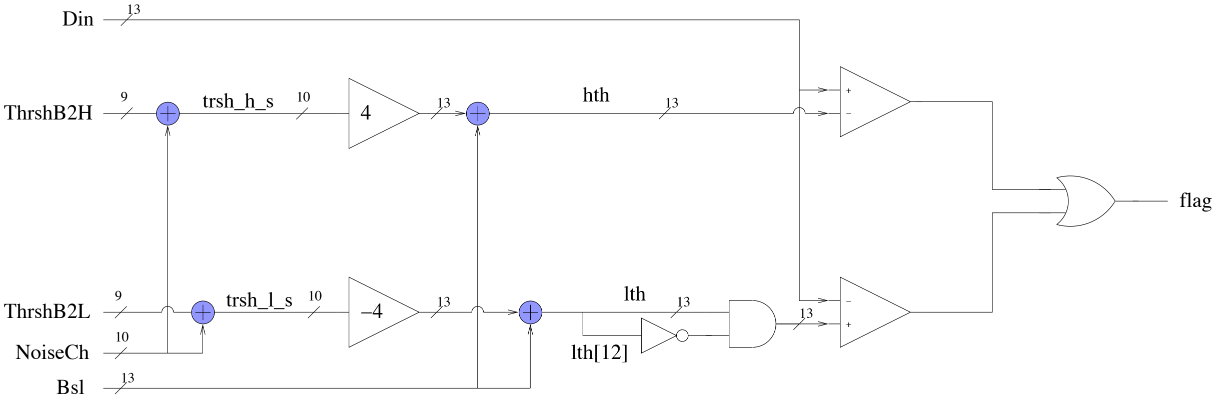


Figure B.7.: Double threshold scheme.

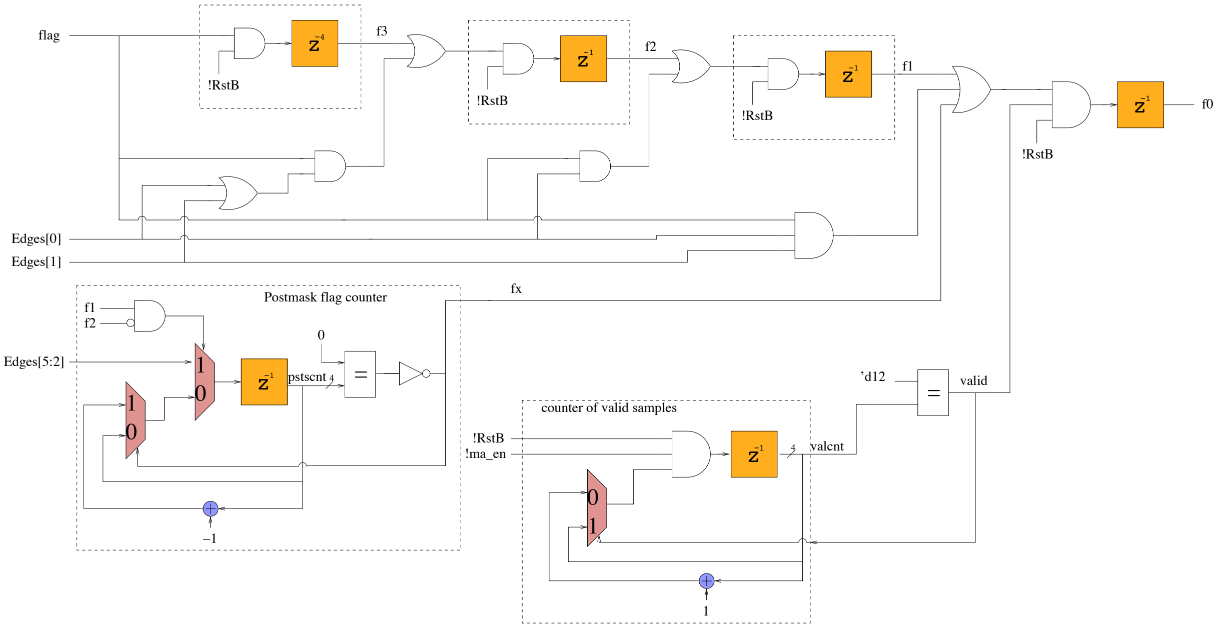


Figure B.8.: Moving Average Filter control logic.

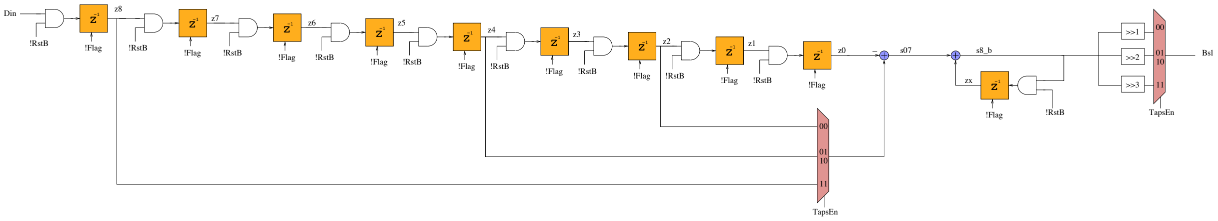


Figure B.9.: Moving Average Filter.

B.4. Zero Suppression

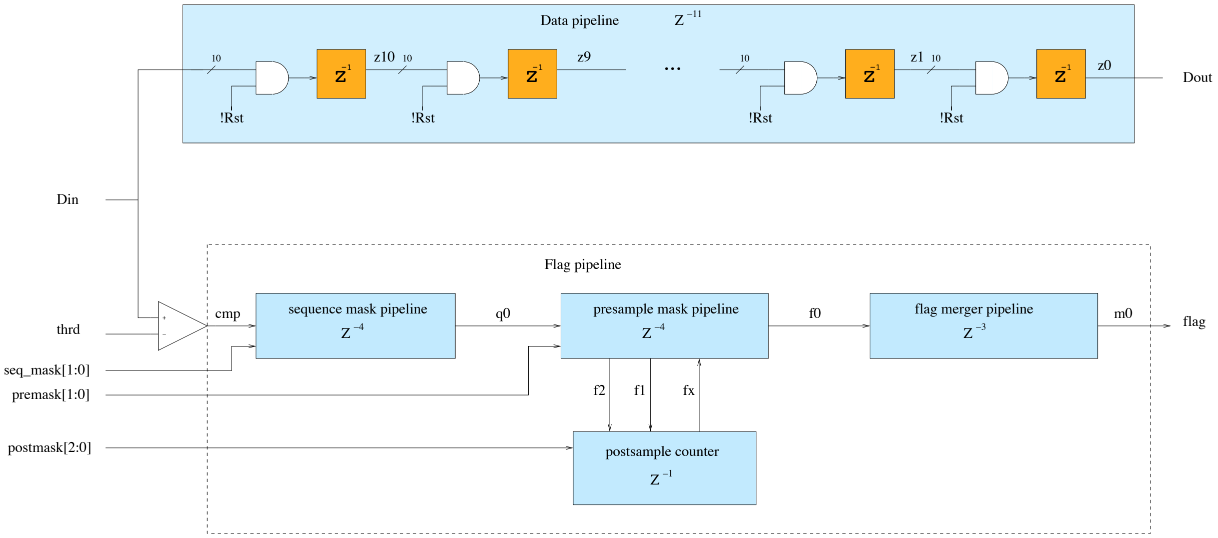


Figure B.10.: Zero suppression abstracted block diagram.

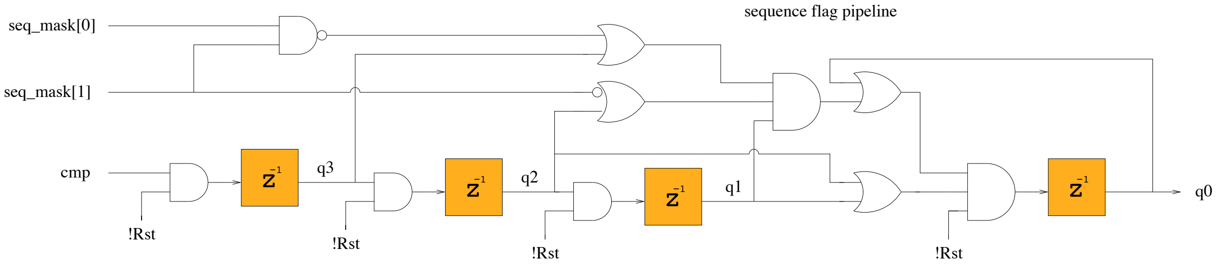


Figure B.11.: Sequence mask pipeline.

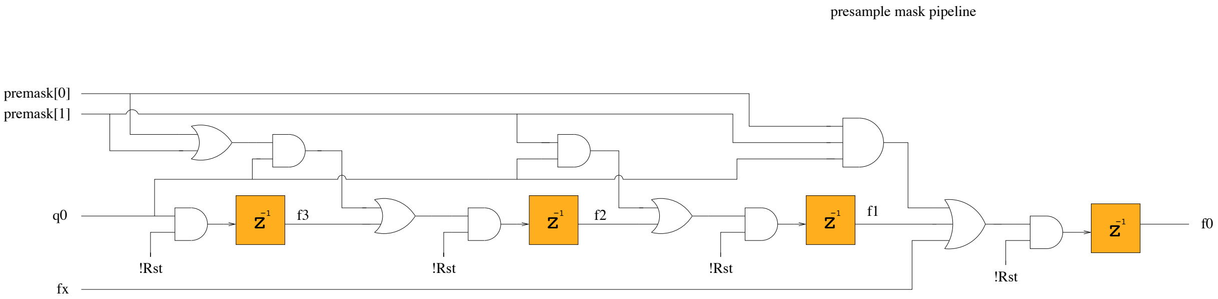


Figure B.12.: Presample mask pipeline.

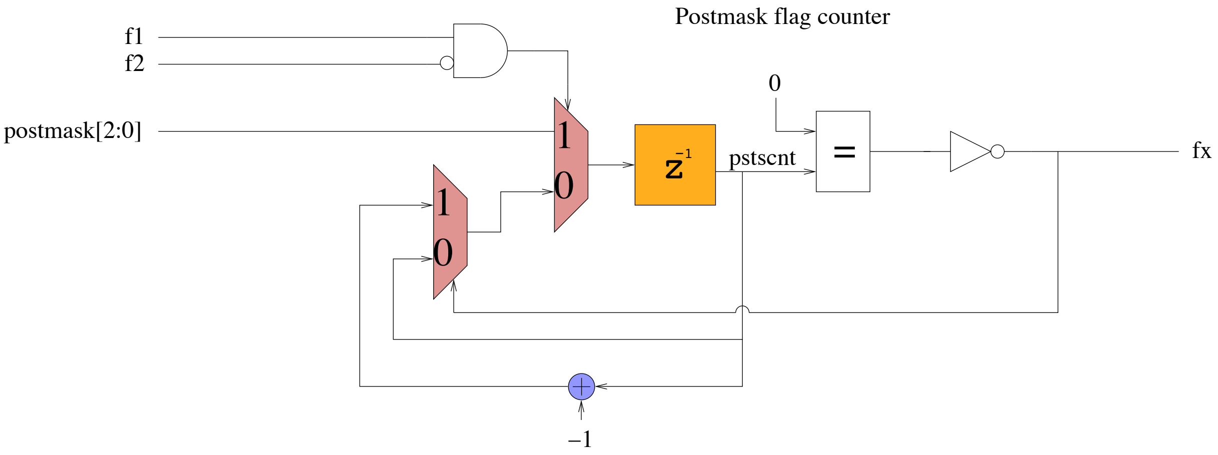


Figure B.13.: Post sample counter.

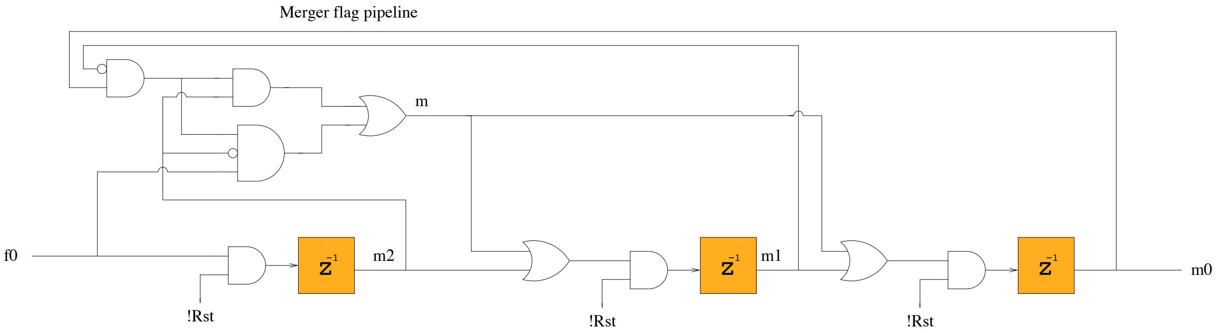


Figure B.14.: Flag merger pipeline.

C. Block diagrams of Simulink models

Simulink models of time pick-off methods, integrator and peak sharpener were used in simulations described in section 4.2. The block diagrams represent the models at the end of the simulations, when they were either excluded as options or they were converted into Verilog models.

C.1. Piece-wise Linear Fitting

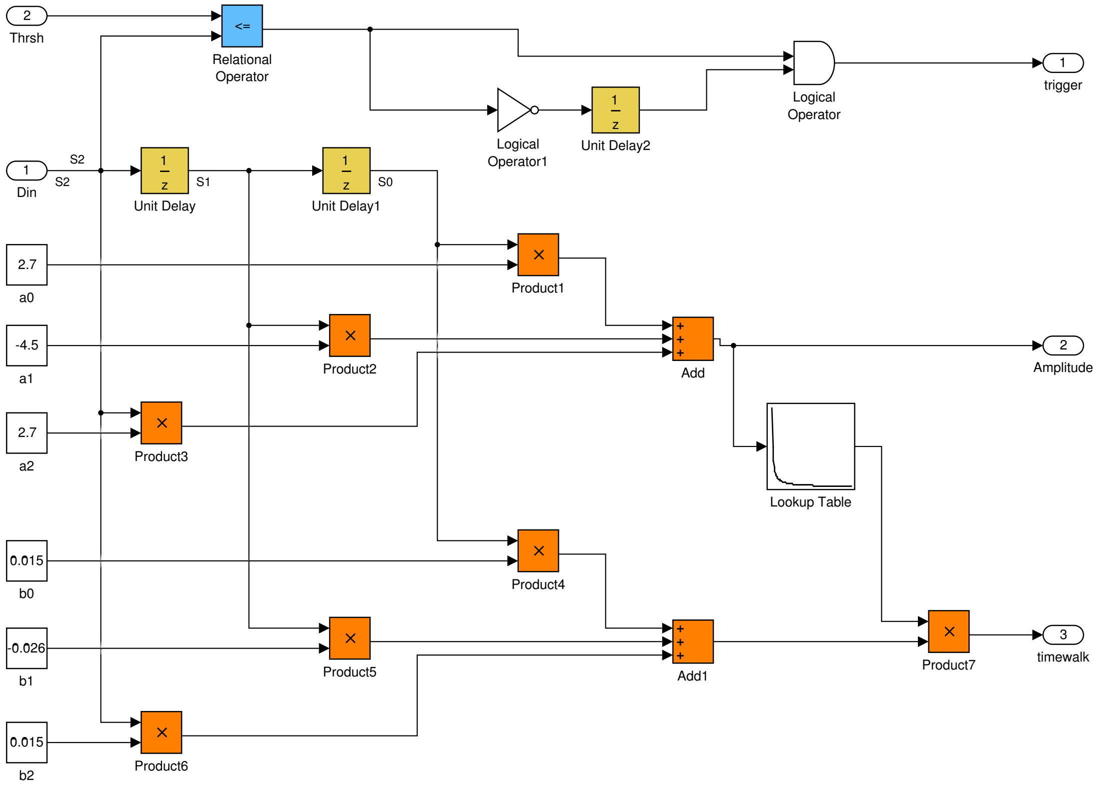


Figure C.1.: Piece-Wise Linear Fitting block diagram

C.2. Deconvolution method

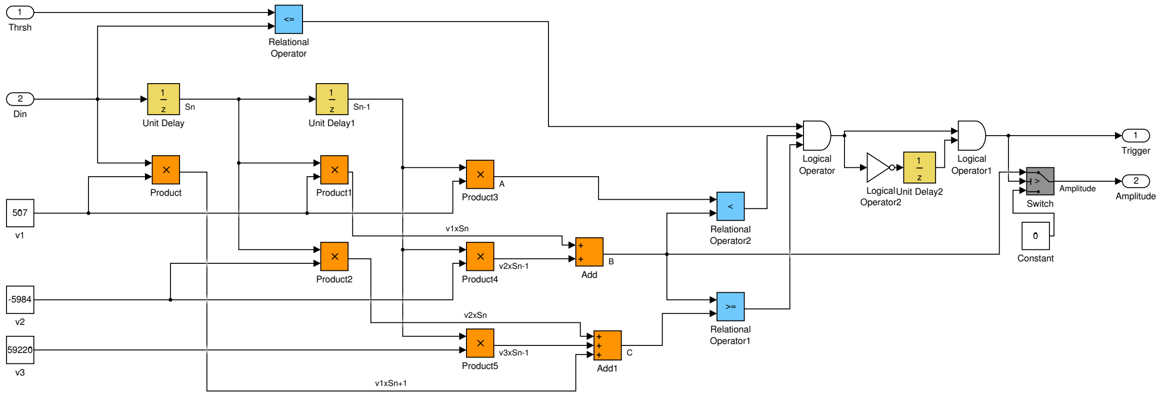


Figure C.2.: Deconvolution filter block diagram

C.3. Peak Finder

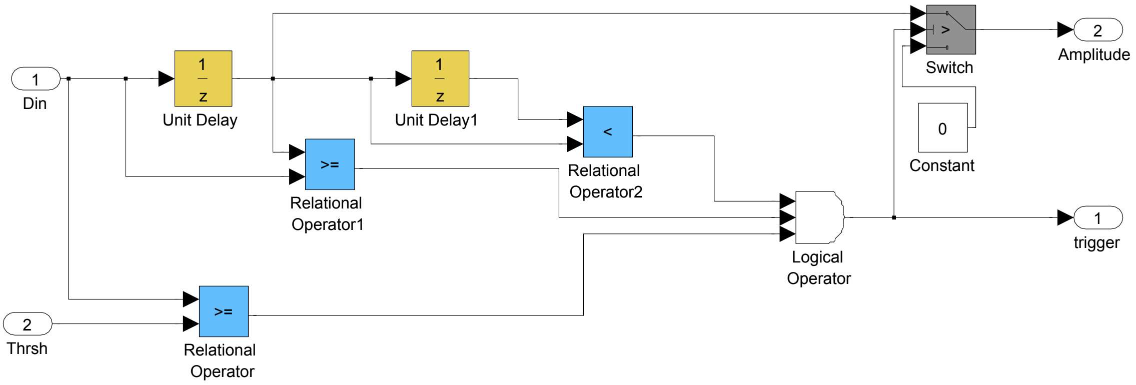


Figure C.3.: Peak Finder block diagram

C.4. Zero-Crossing Identification

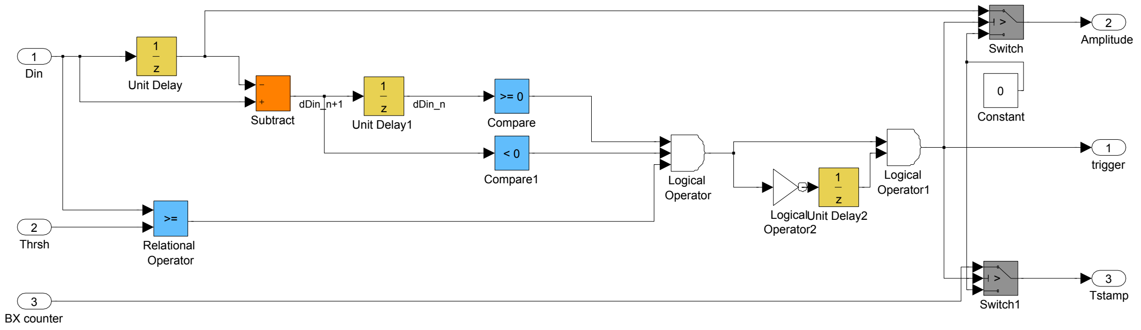


Figure C.4.: Zero-Crossing Identification version A block diagram

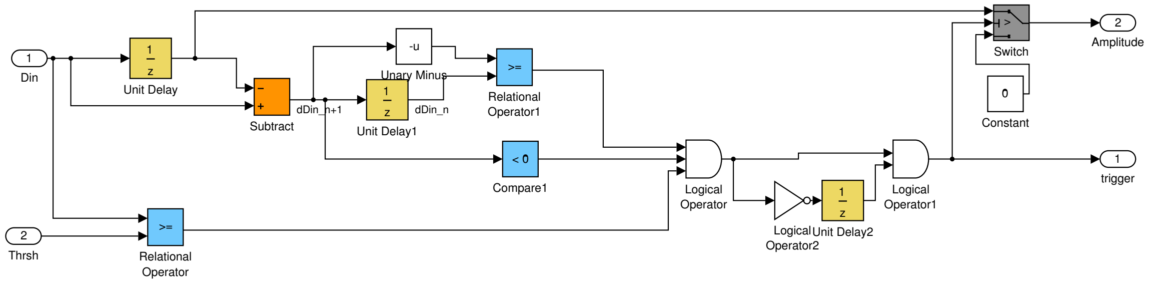


Figure C.5.: Zero-Crossing Identification version B block diagram

C.5. Pulse Recognition

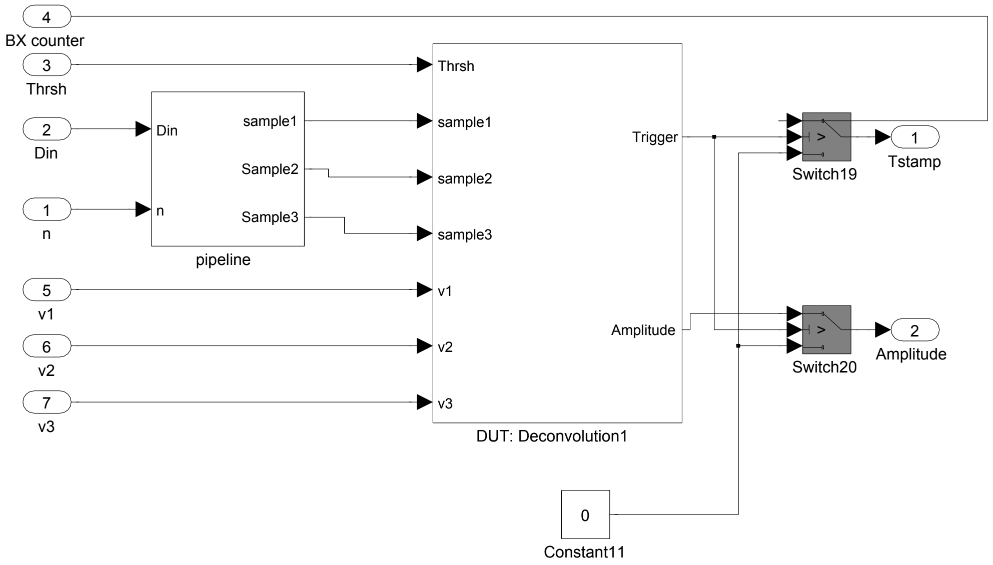


Figure C.6.: Pulse Recognition (PuR) block diagram

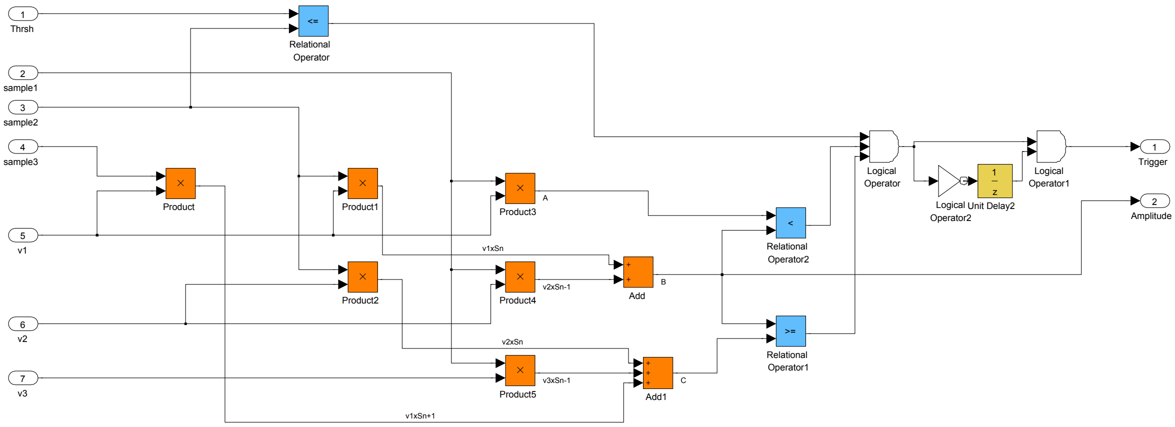


Figure C.7.: Deconvolution1 in PuR

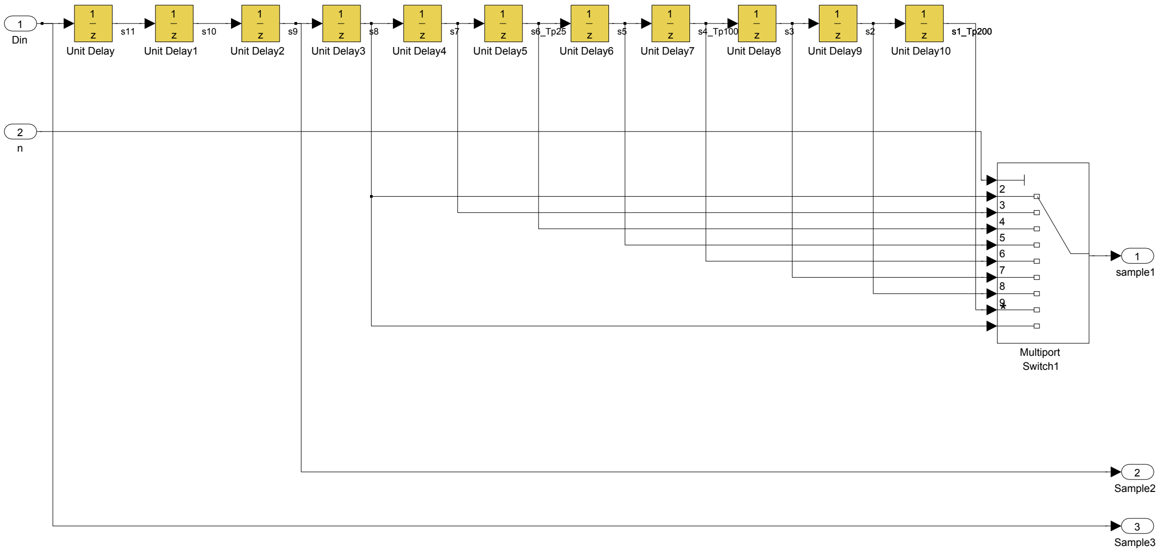


Figure C.8.: Pipeline in PuR

C.6. Constant Fraction Discriminator

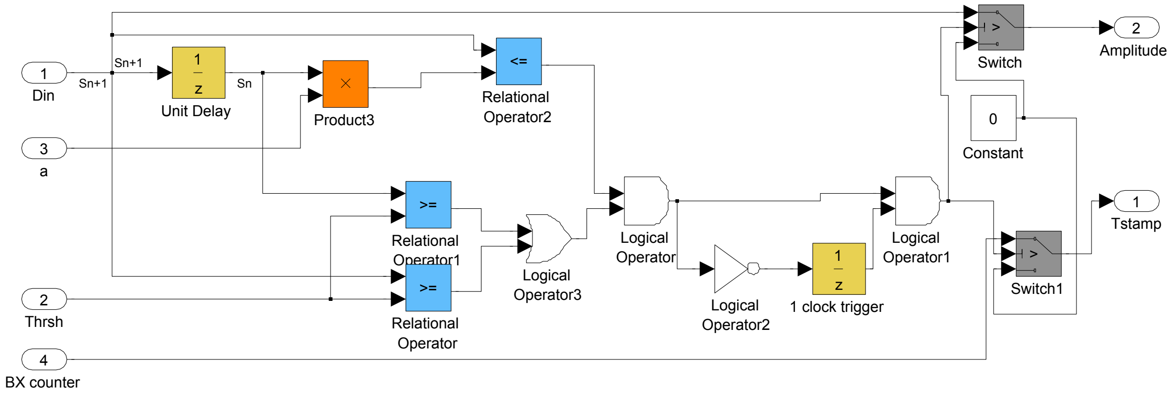


Figure C.9.: Constant fraction discriminator block diagram

C.7. Peak sharpening

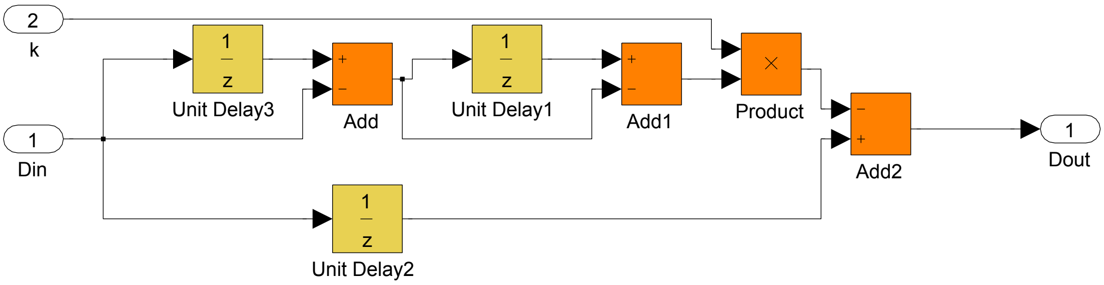


Figure C.10.: Peak sharpener block diagram

C.8. Integrator

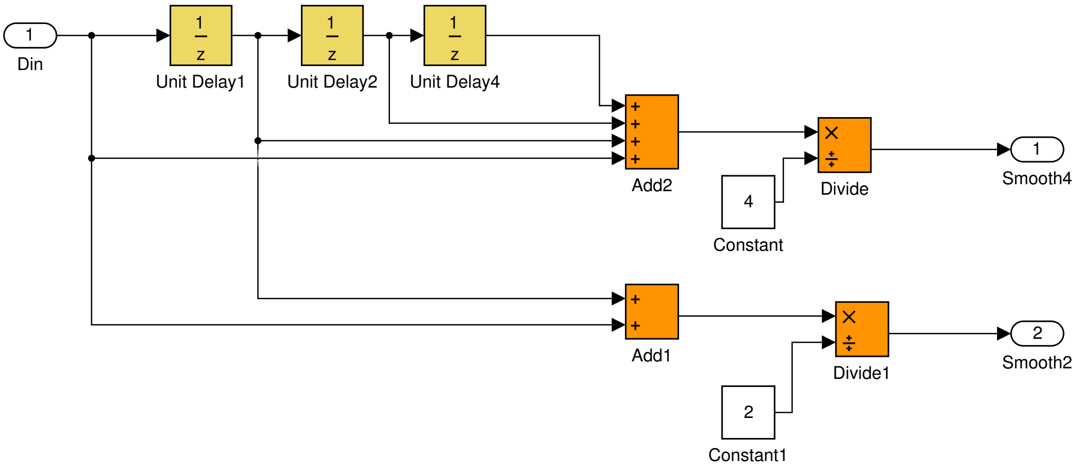


Figure C.11.: Integrator block diagram

D. Block Diagram of Pulse Recognition (Verilog Model)

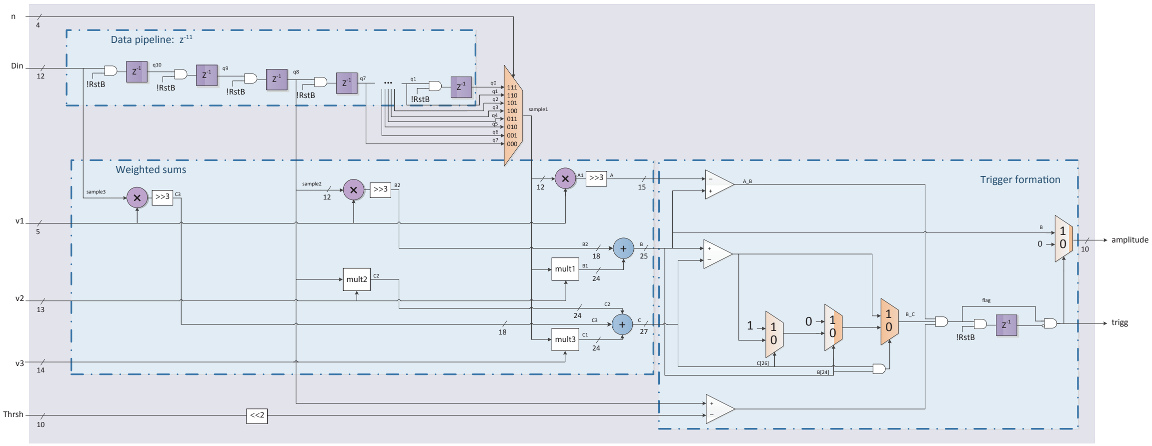


Figure D.1.: Block diagram of Pulse Recognition.

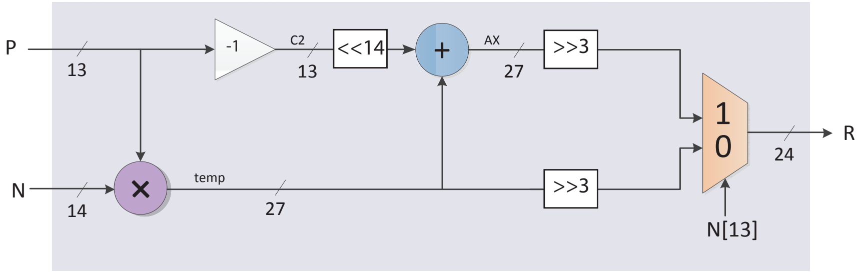


Figure D.2.: Multiplication used in PuR.