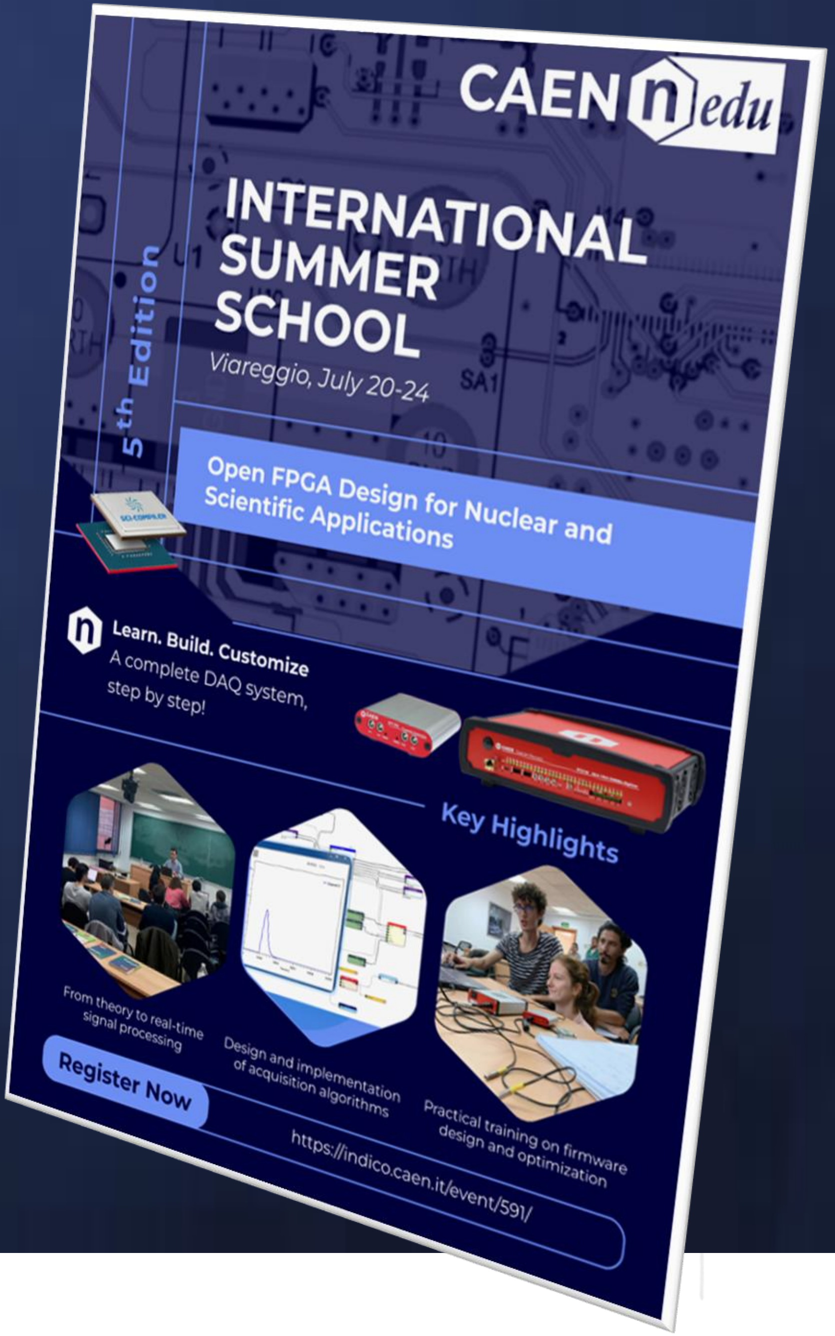
CAEN edu

Digital Pulse Processing Algorithm

CAEN Summer School 2026

Viareggio, Italy, July 20th – 24th, 2026

Alberto Potenza



CAEN Tools for Discovery

Summary

01 From Particle to Digital Domain

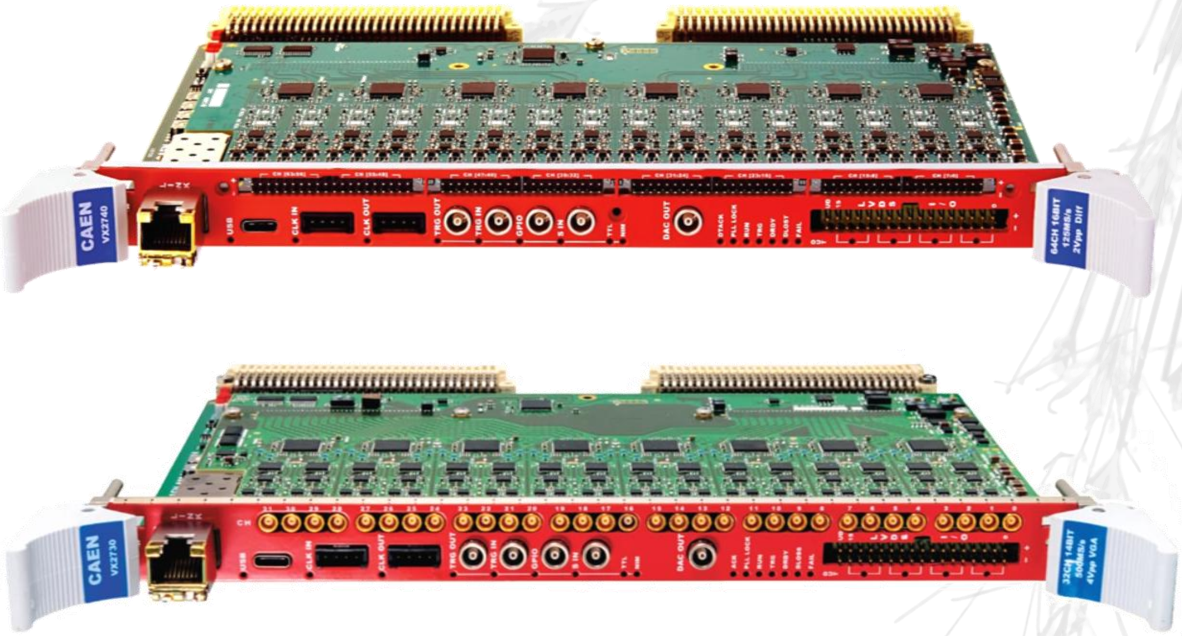
› Physics of Nuclear Pulse
› Analog Front-End
› ADC Parameters

02 Real-Time Event Triggering

› Leading-Edge
› CFD
› Sub-clock Interpolation

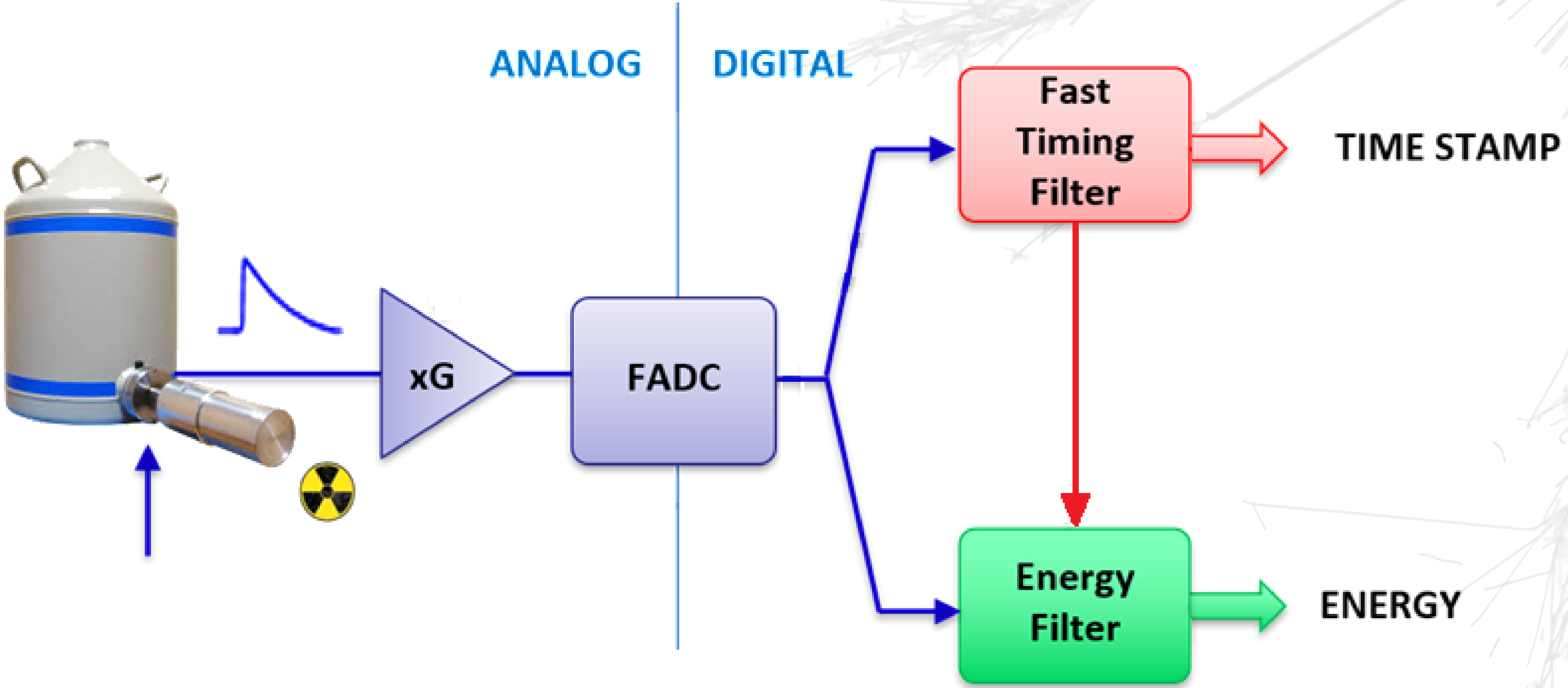
03 Digital Energy Spectroscopy

› Charge Integration
› Trapezoidal Filter
› Pulse Shape Discrimination

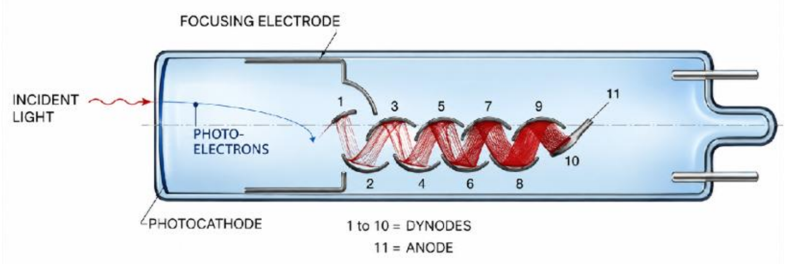


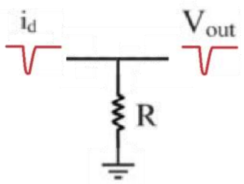
From Particles to Digital Domain

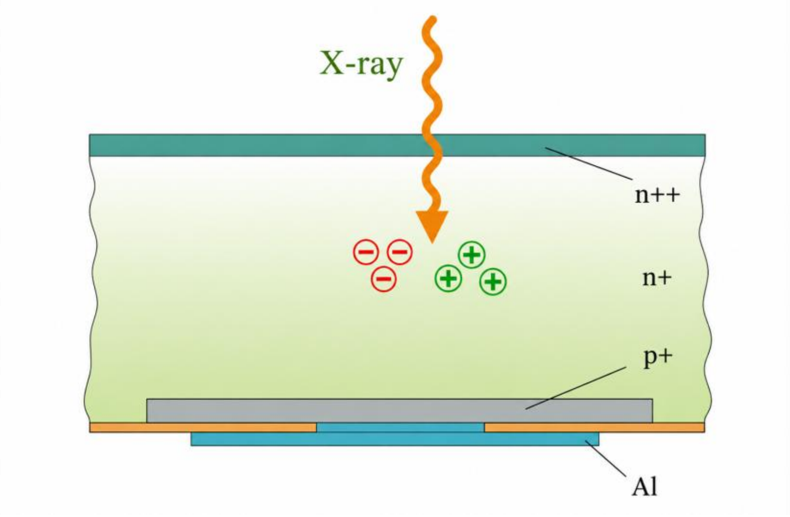
From Particle to Digital Domain

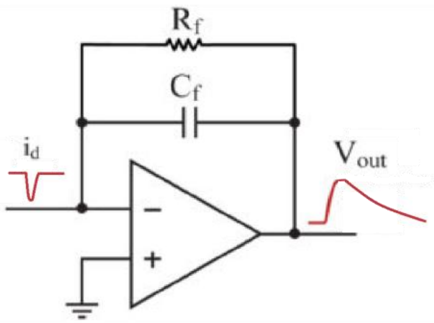


Physics of Nuclear Pulse









Moving Charges

Current Signal

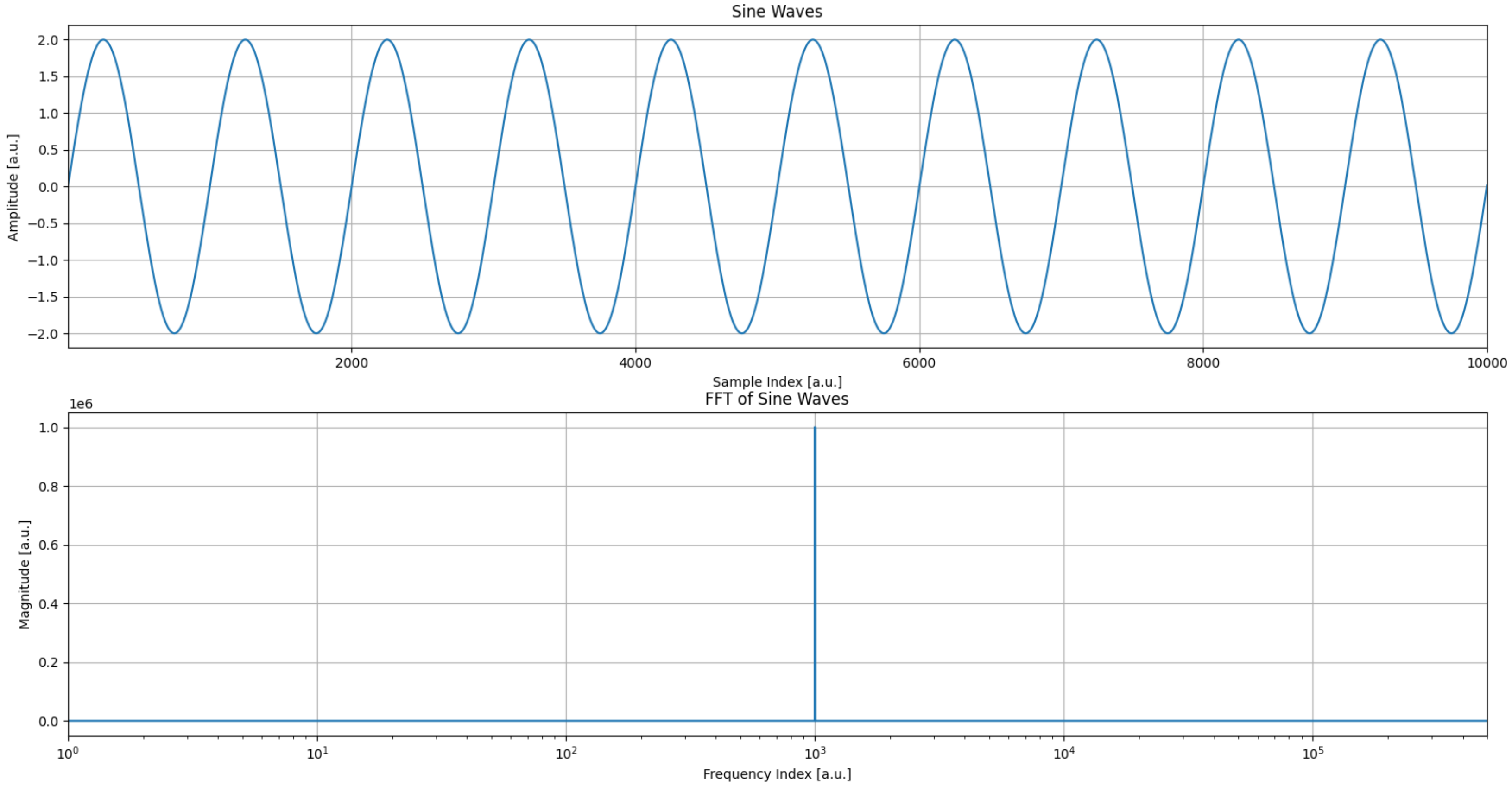
Ground Resistor

Charge Amplifier

$$V(t) = V_0\,(e^{-t/\tau_f} - e^{-t/\tau_r})$$

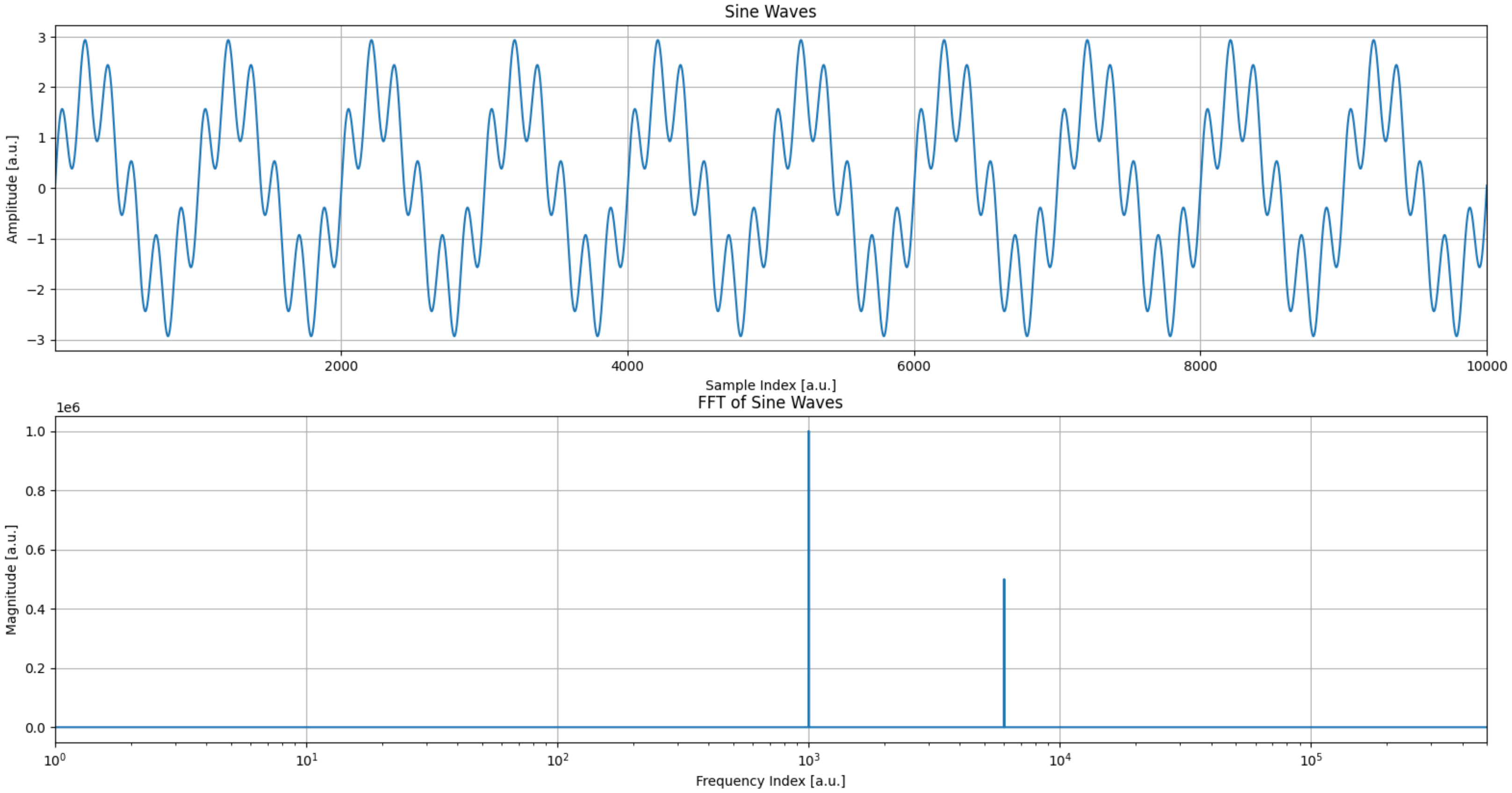
Physics of Nuclear Pulse

$$V(t) = V_0 \ \sin(\omega_0 t)$$



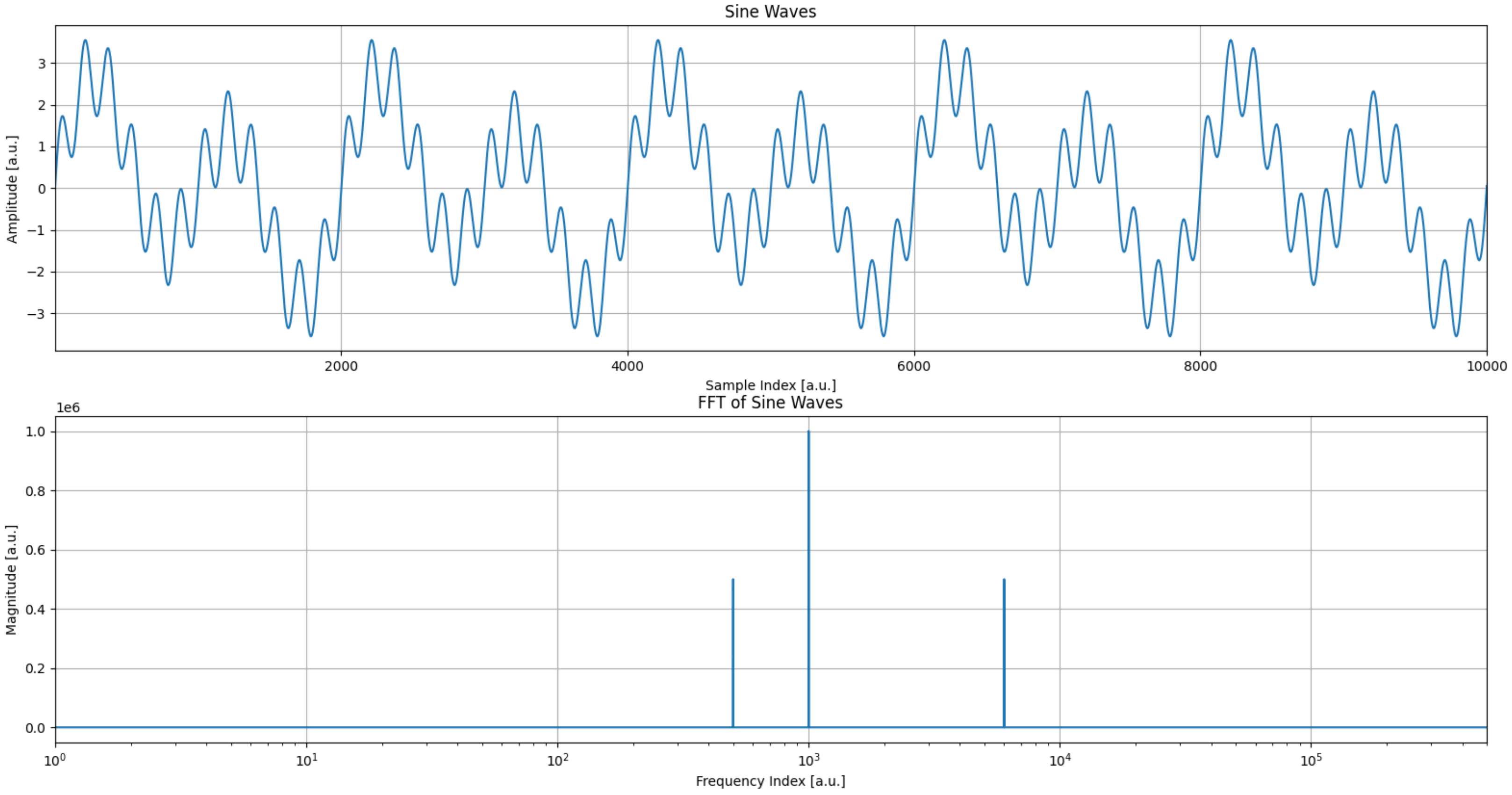
Physics of Nuclear Pulse

$$V(t) = V_0\ \sin(\omega_0 t) + V_1\ \sin(\omega_1 t)$$



Physics of Nuclear Pulse

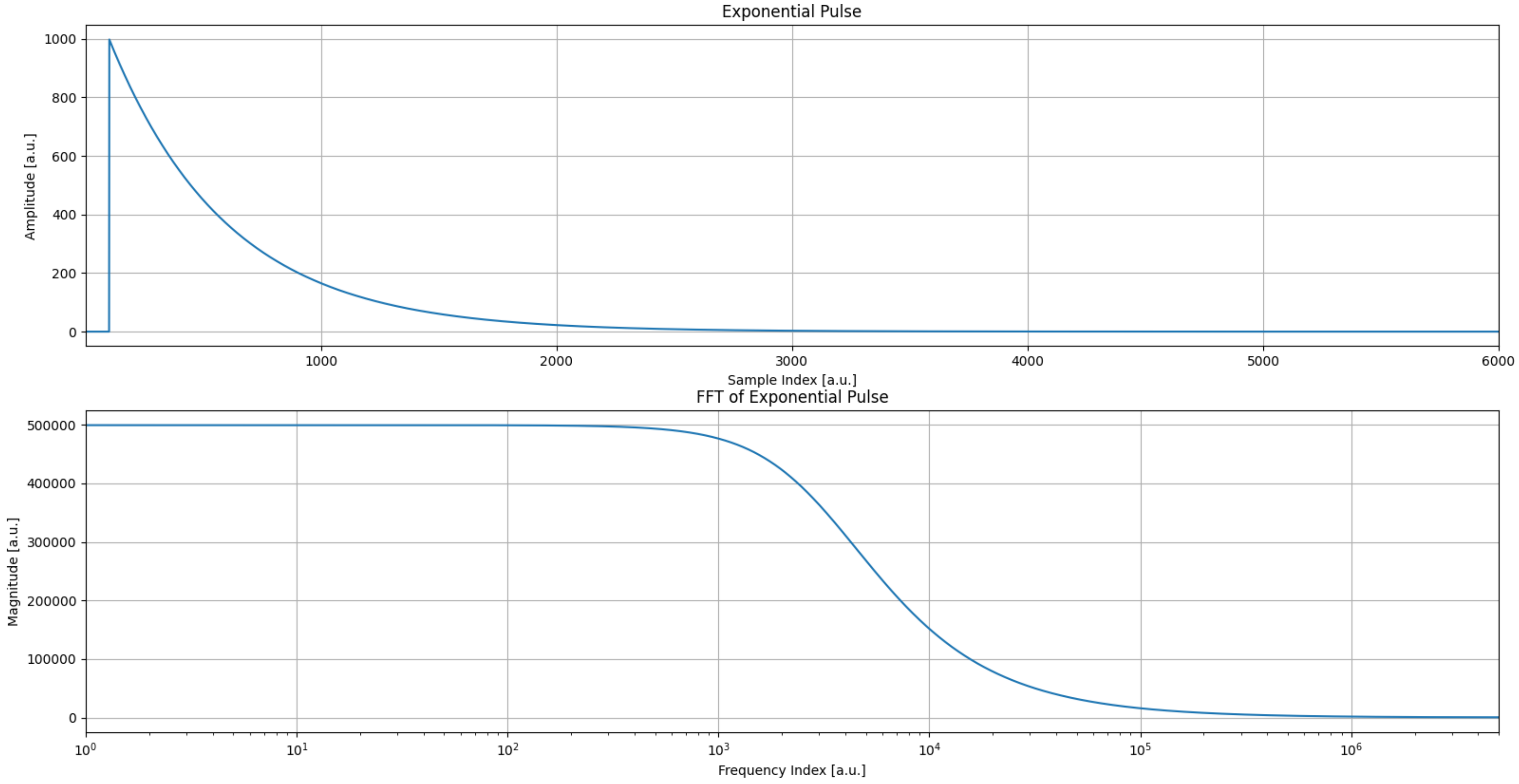
$$V(t) = V_0\ \sin(\omega_0 t) + V_1\ \sin(\omega_1 t) + V_2\ \sin(\omega_2 t)$$



CAEN
Tools for Discovery

Physics of Nuclear Pulse

$$V(t) = V_0\,\left(e^{-t/\tau_f} - e^{-t/\tau_r}\right)$$



CAEN Tools for Discovery

From Particle to Digital Domain

$$V(t) = V_0\,(e^{-t/\tau_f} - e^{-t/\tau_r})$$

$$\xrightarrow[\tau_r \to RC]{t \ll \tau_f}$$

$$V(t) = V_0\,(1 - e^{-t/RC})$$

For a simple one-stage low-pass RC network, the 10% to 90% rise time is proportional to the time constant $\tau = RC$

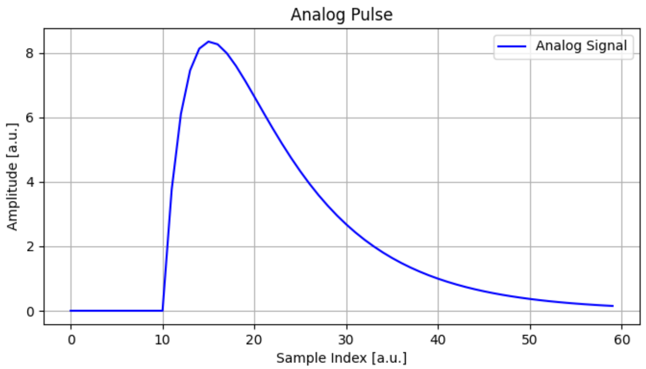
$$V(t_1) = 0.1\,V_0$$
$$V(t_2) = 0.9\,V_0$$

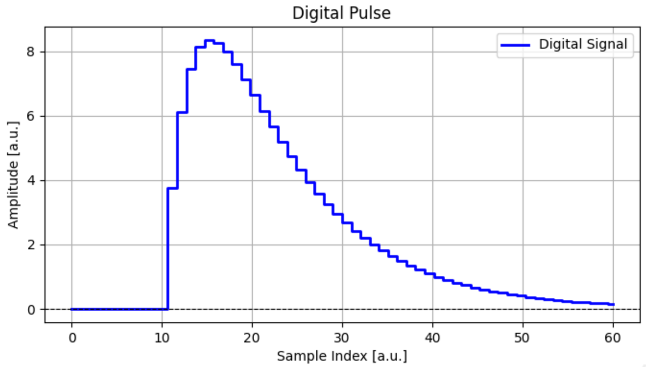
$$t_{rise} = RC\,ln(9) \longleftrightarrow RC = \frac{1}{2\pi\,BW}$$

$$\boxed{BW \approx \frac{0.35}{t_{rise}}}$$

CAEN Tools for Discovery

From Particle to Digital Domain





$t_{rise} \approx 100\ ns$

$BW \approx 3.5\ MHz$

$f_{sampleNyq} \approx 7\ MHz$

$Why\ fsamp_{ling} \gg fs_{ampleNyq}$?

CAEN
Tools for Discovery

From Particle to Digital Domain

Beyond Nyquist Limits

- Nyquist guarantees information preservation, but exact reconstruction requires ideal sinc interpolation.
- Ideal sinc interpolation is computationally expensive and never performed in real-time.

Real ADC Performance

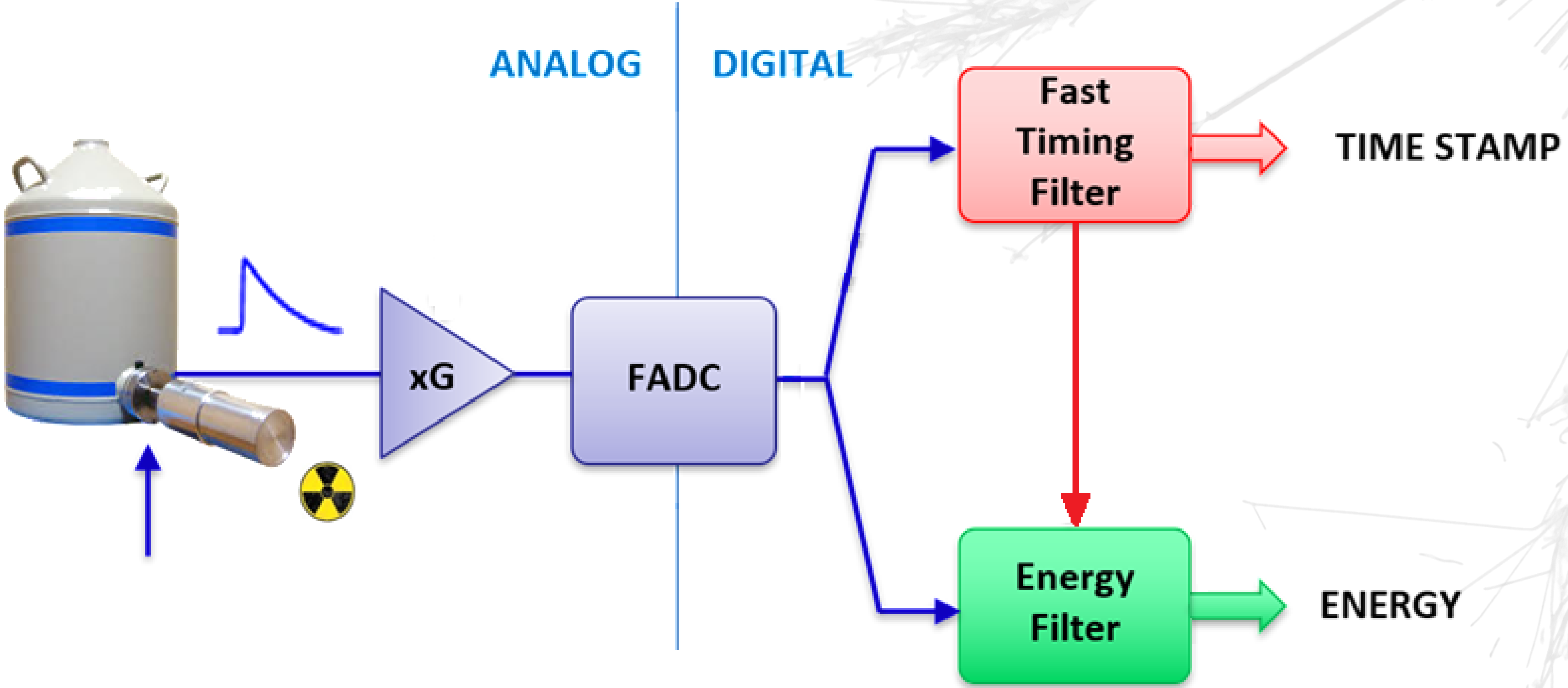
- Resolution is degraded by ENOB, clock noise, INL/DNL non-linearity.
- High-rate oversampling directly helps to average out and mitigate these effects.

Filter Performance Margin

- High sample density reduces interpolation error at the timing threshold and pulse peaks.
- Digital filters perform better with more samples.

CAEN Tools for Discovery

From Particle to Digital Domain



Trigger Filters

Trigger Filters

Primary Objective: Precise Event Timestamping

Target: Determine the exact arrival time with maximum temporal precision.

Application: Time-of-Flight (ToF), coincidence and event timestamping.

Filter Optimization: Minimizing Temporal Jitter

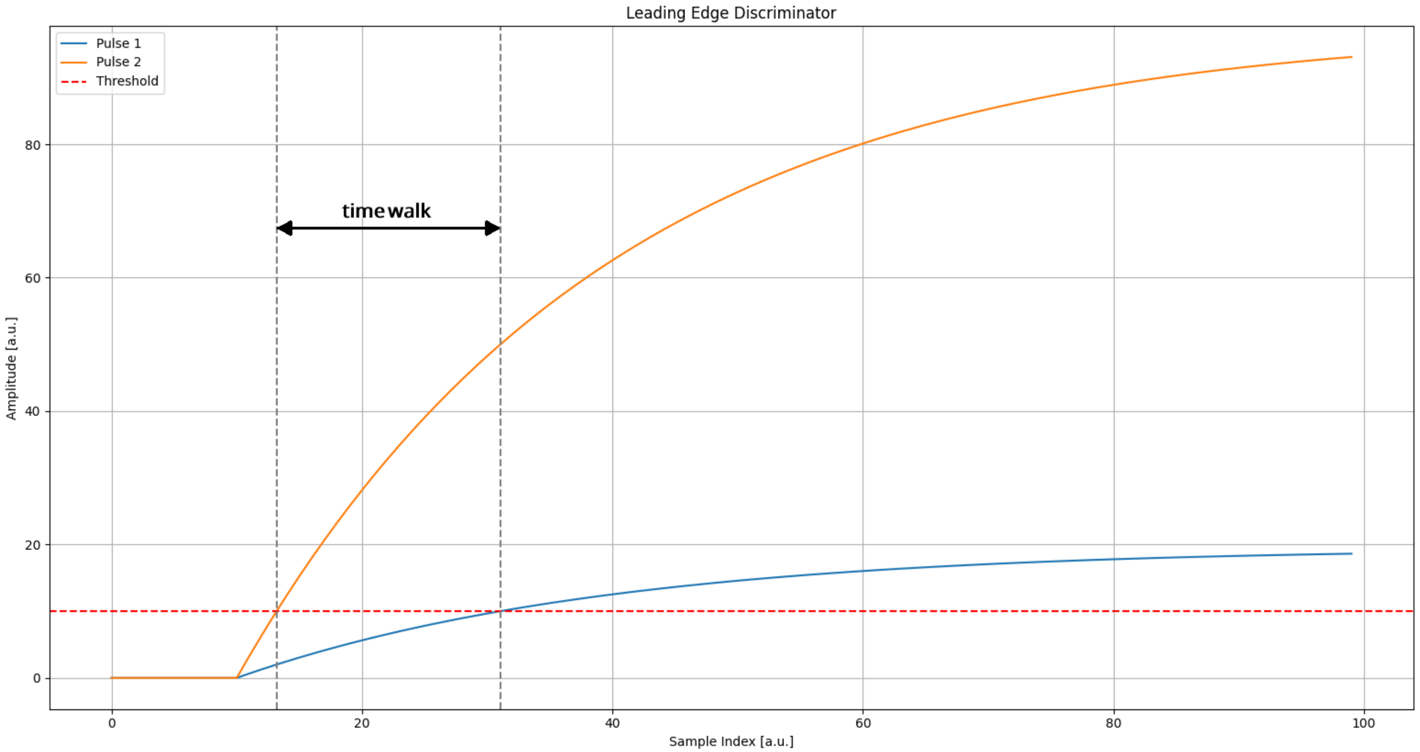
- Jitter Equation: Temporal jitter depends strictly on noise and the signal slope at the threshold crossing point:

$$\sigma_t = \frac{\sigma_V}{\frac{dV}{dt}}$$

- Bandwidth Requirement: Must preserve high frequencies to maintain a steep rise time (dV/dt)

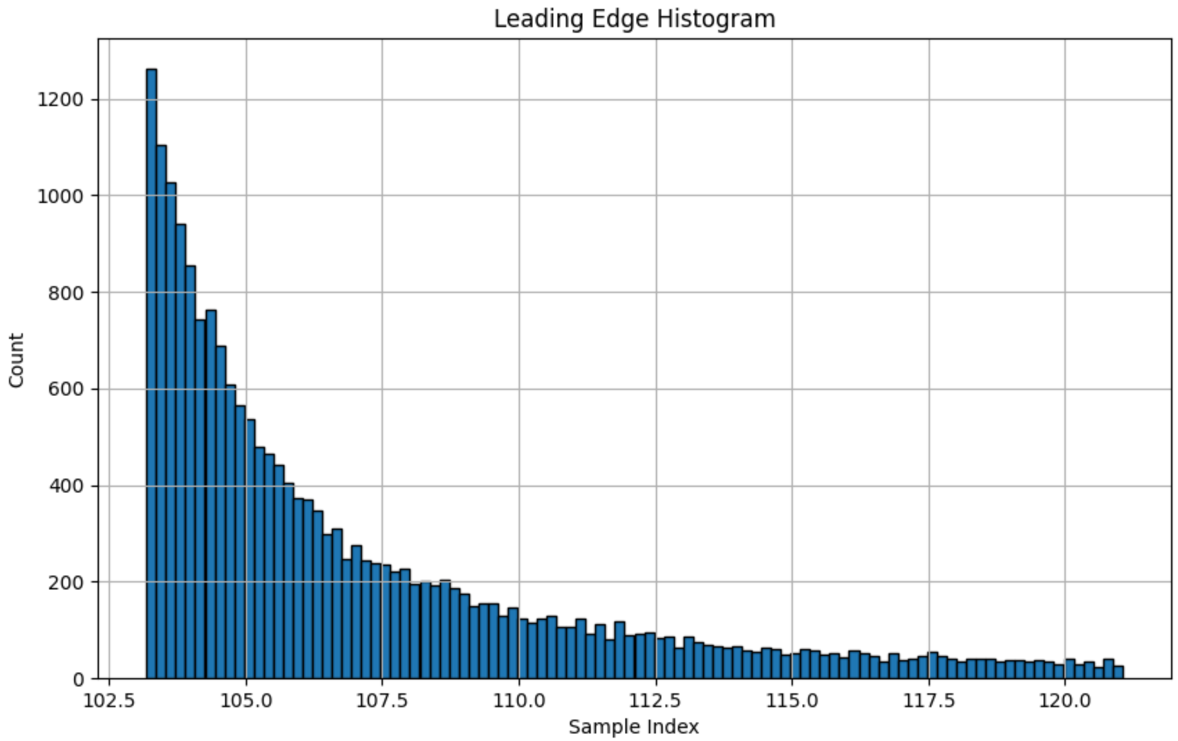
Leading Edge Discriminator

$$t_{trig} = min\{t : V(t) \geq V_{th}\}$$

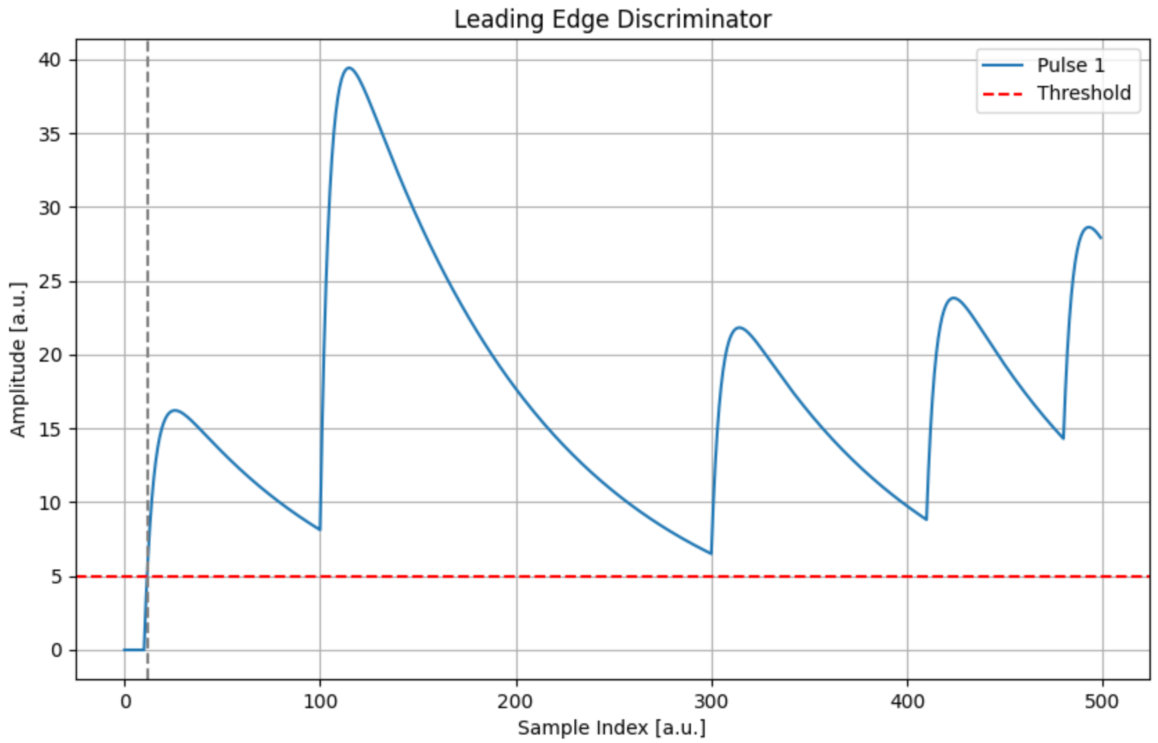


Leading Edge Discriminator

$$Time\ Walk\ Problem$$



Pile-up Problem

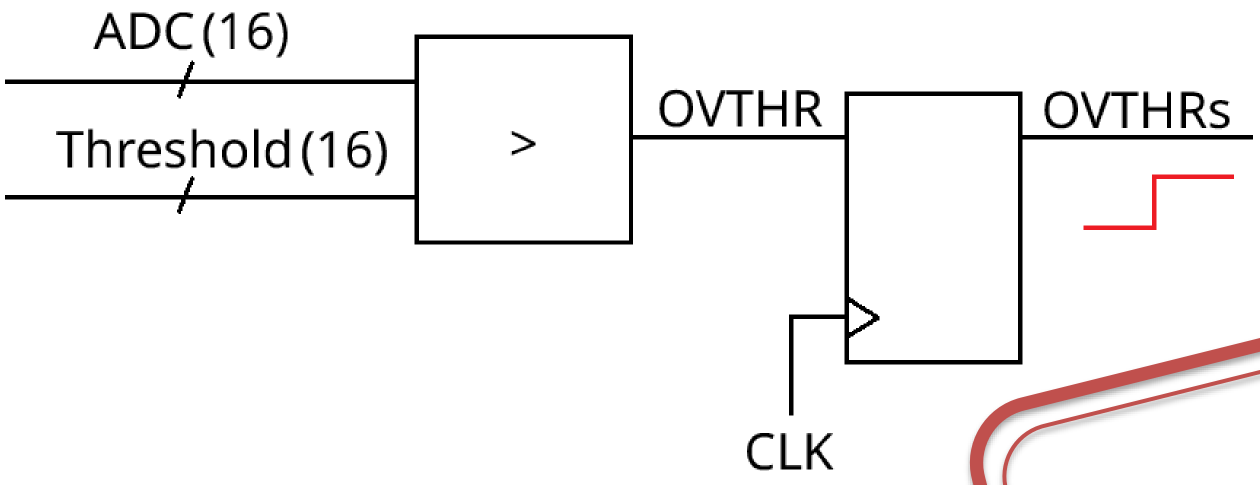


Leading Edge Discriminator

$$Why\ a\ Leading\ Edge\ Discriminator\ ?$$

Leading Edge Discriminator

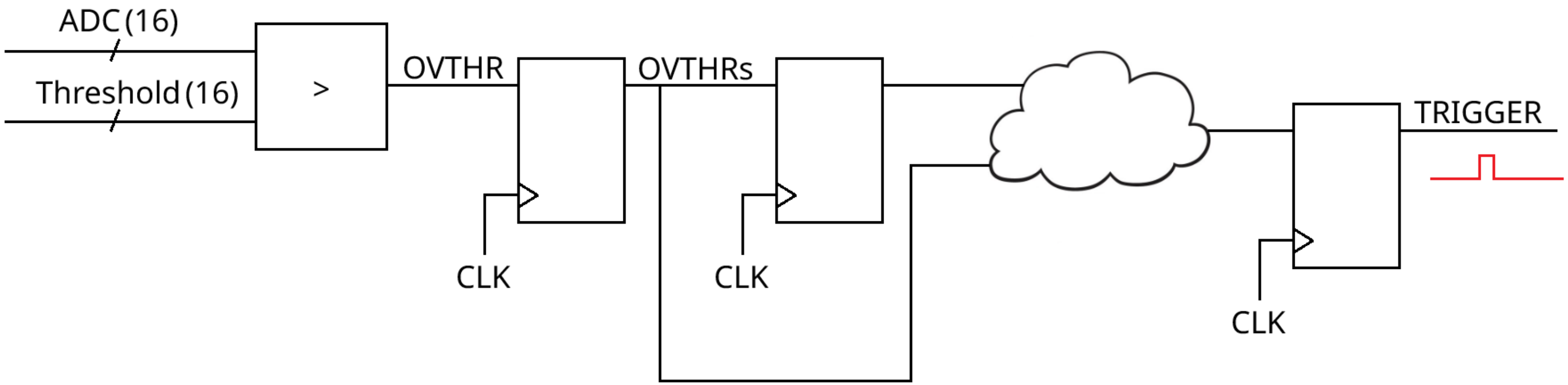
Why a Leading Edge Discriminator ?



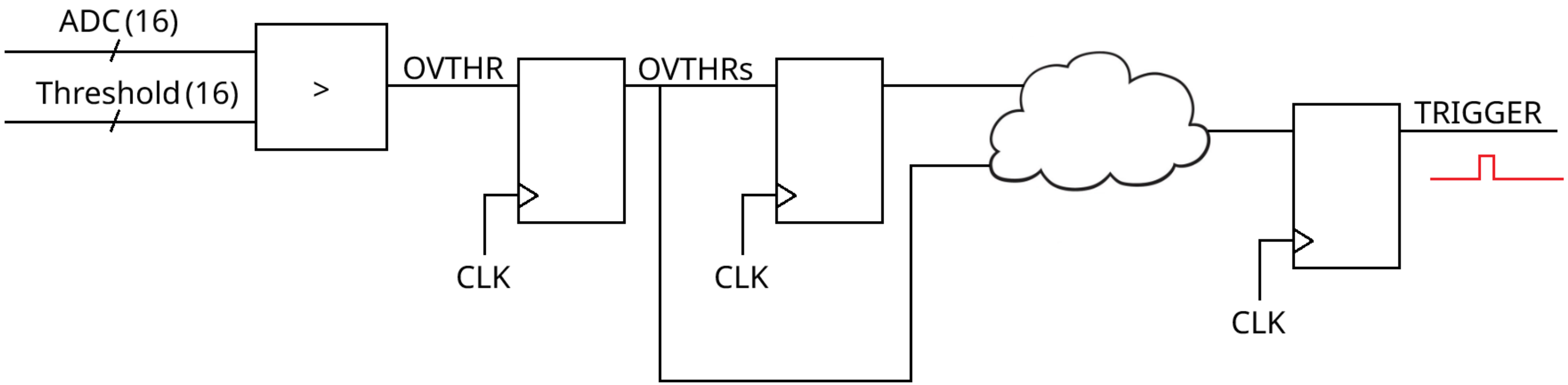
SIMPLE !!!

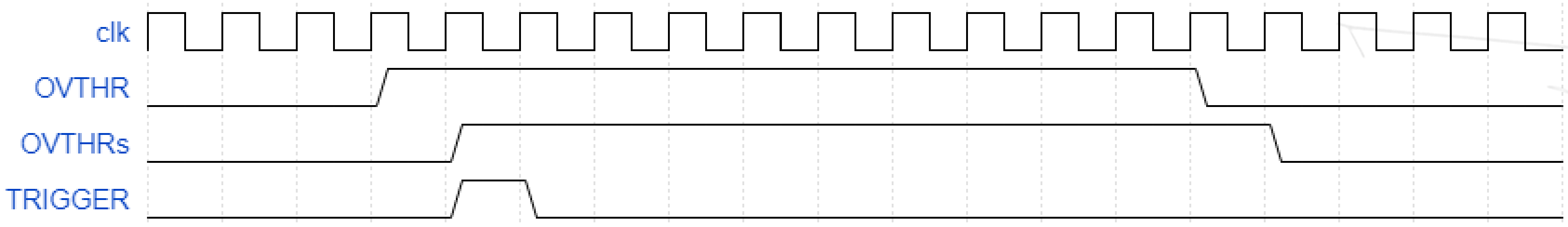
Leading Edge Discriminator

$$Why\ a\ Leading\ Edge\ Discriminator\ ?$$



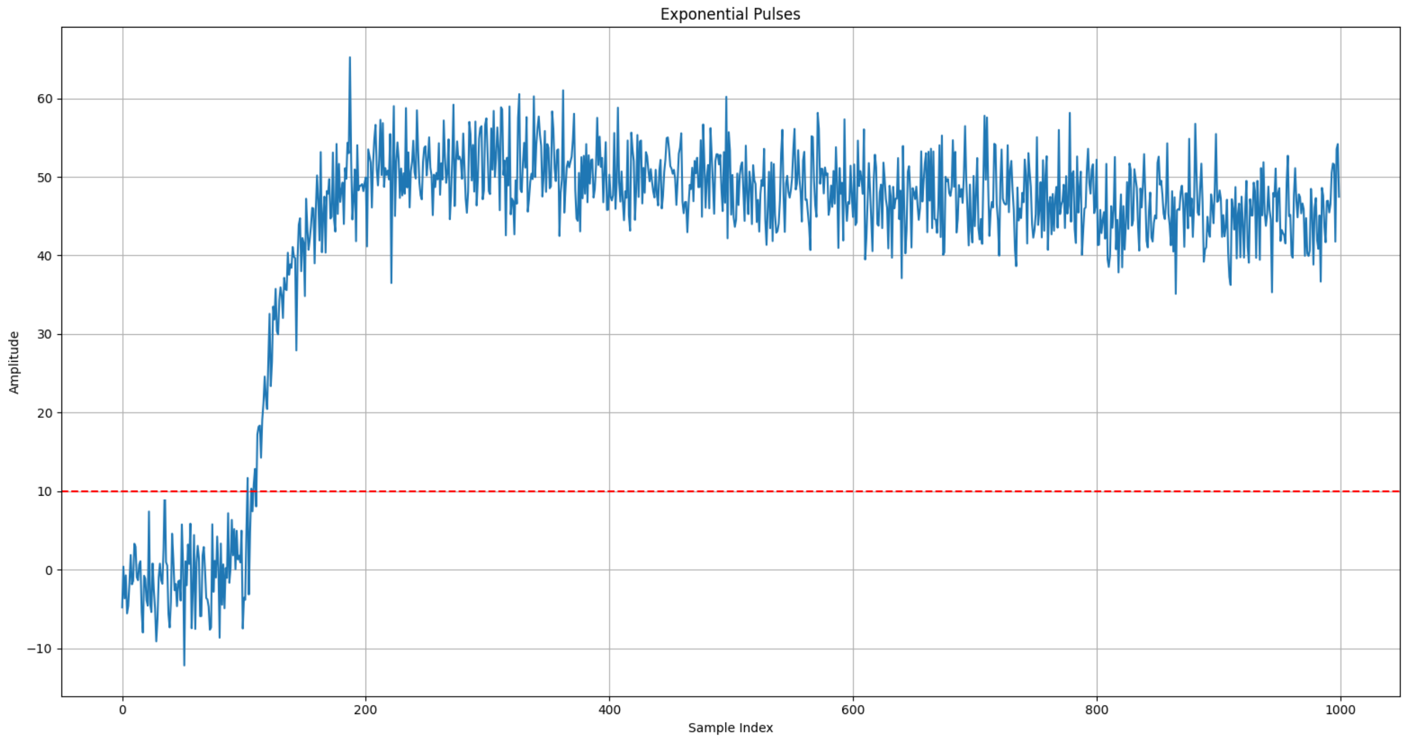
Leading Edge Discriminator





Leading Edge Discriminator

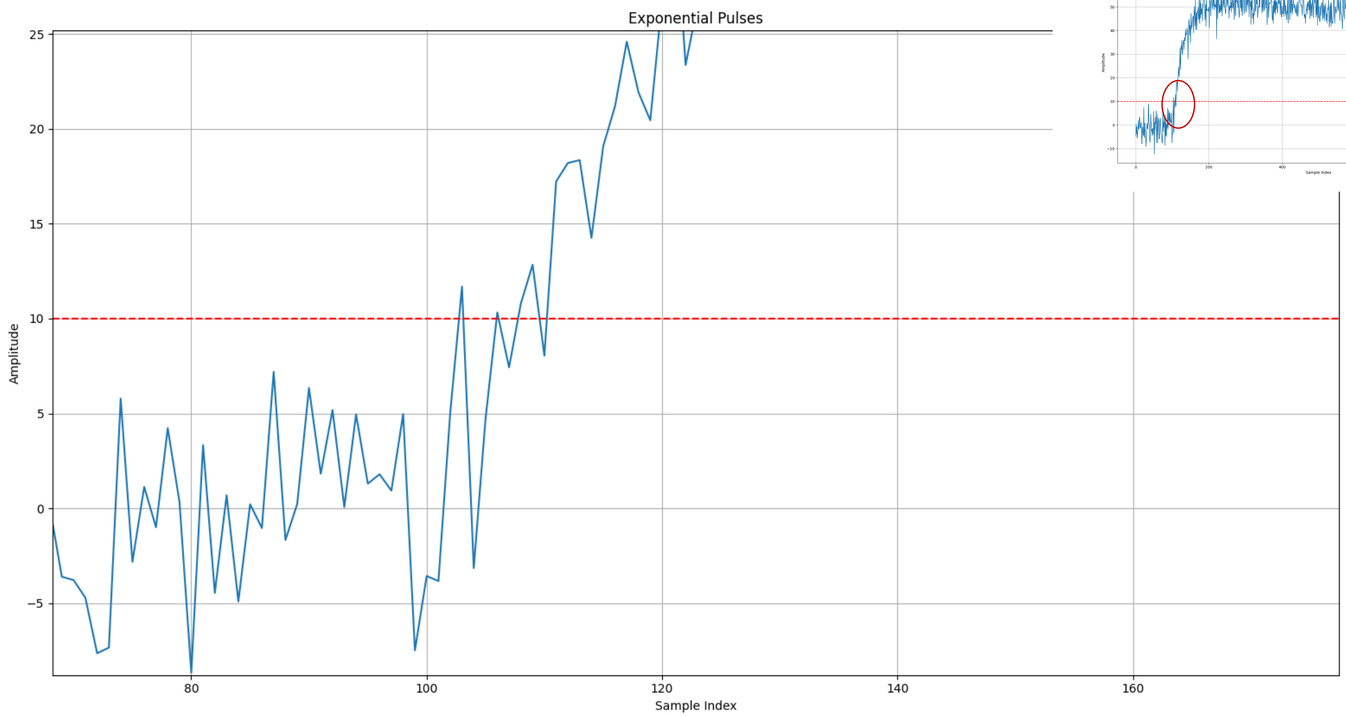
$Noise\ Effects$

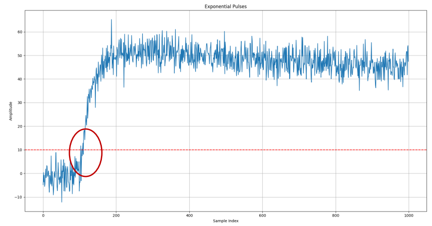


CAEN Tools for Discovery

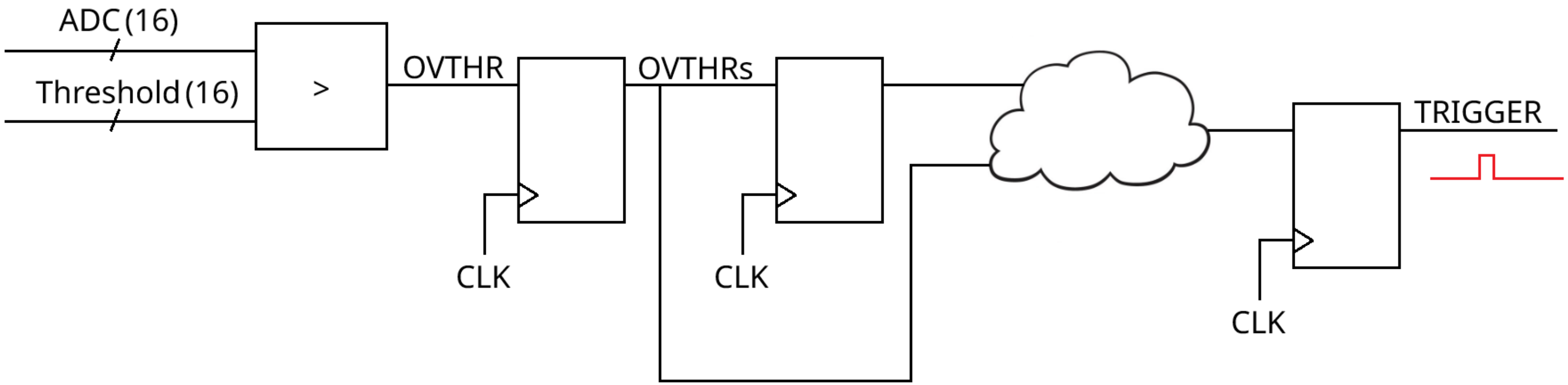
Leading Edge Discriminator

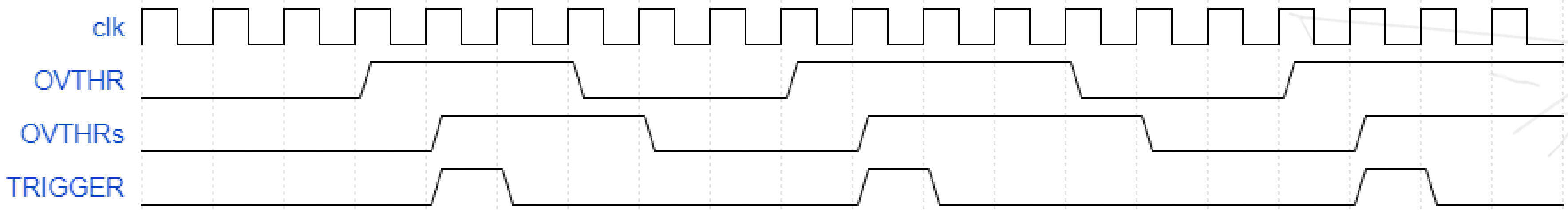
$Noise\ Effects$



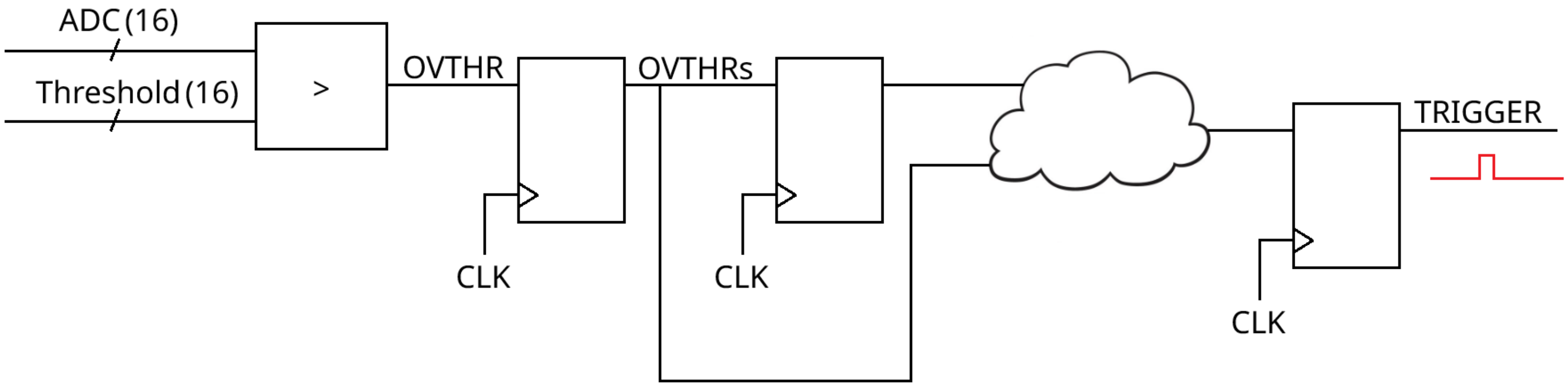


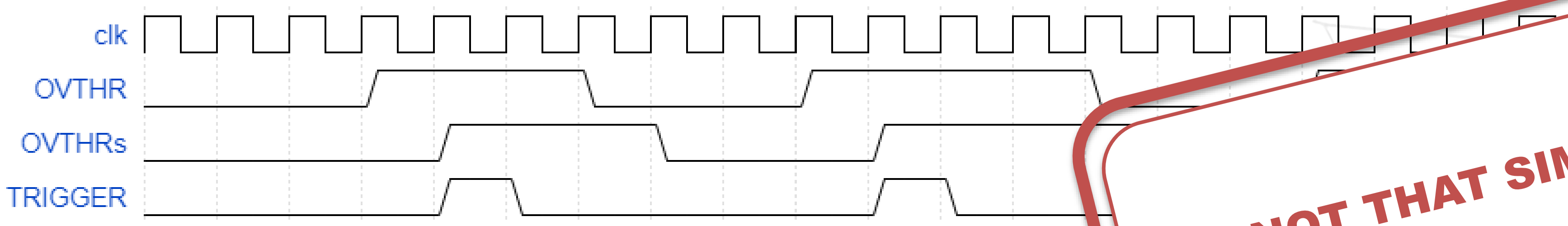
Leading Edge Discriminator





Leading Edge Discriminator

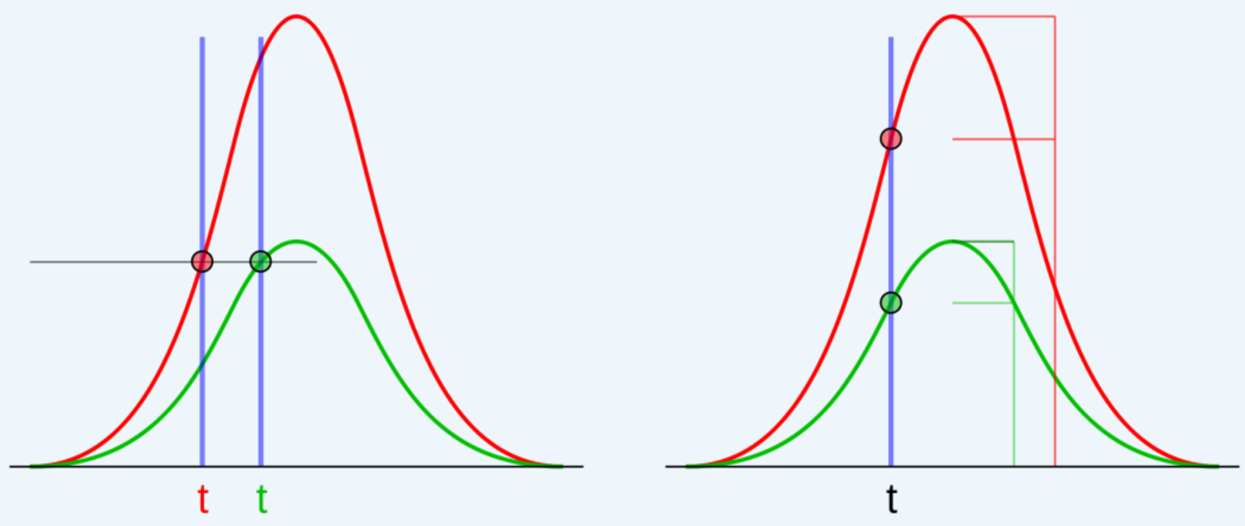




NOT THAT SIMPLE !!!

Constant Fraction Discriminator

Shifting the paradigm:
from an absolute voltage threshold to an amplitude-independent fraction of the pulse.



Constant Fraction Discriminator

$$V_{cfd}(t) = kV_{in}(t) - V_{in}(t - t_d)$$

$$0 = kV_{in}(t) - V_{in}(t - t_d)$$

$$\Downarrow$$

$$kV_0\left(e^{-t/\tau_f} - e^{-t/\tau_r}\right) = V_0\left(e^{-\frac{t-t_d}{\tau_f}} - e^{-\frac{t-t_d}{\tau_r}}\right)$$

Constant Fraction Discriminator

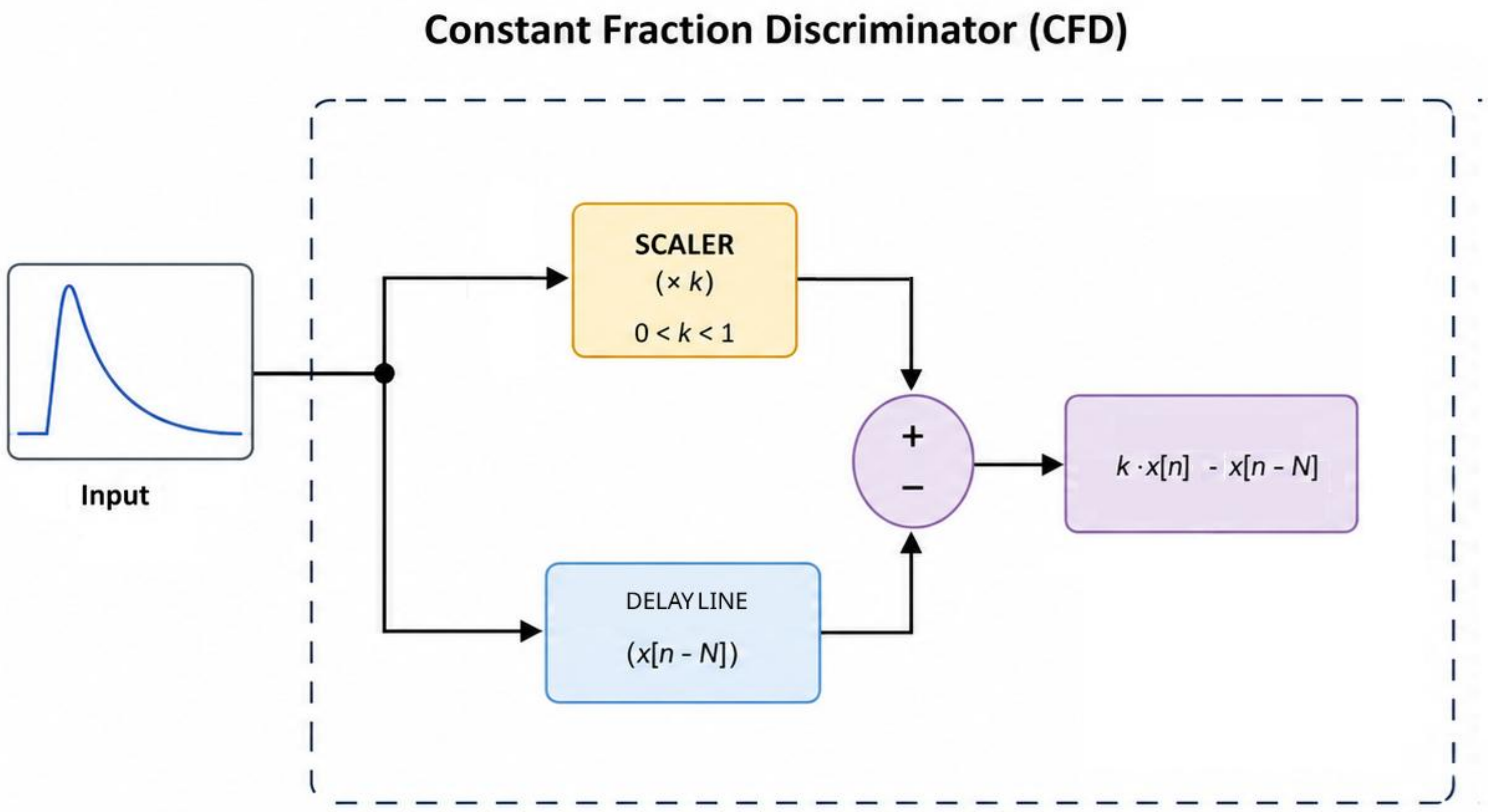
$$V_{cfd}(t) = kV_{in}(t) - V_{in}(t - t_d)$$

$$0 = kV_{in}(t) - V_{in}(t - t_d)$$

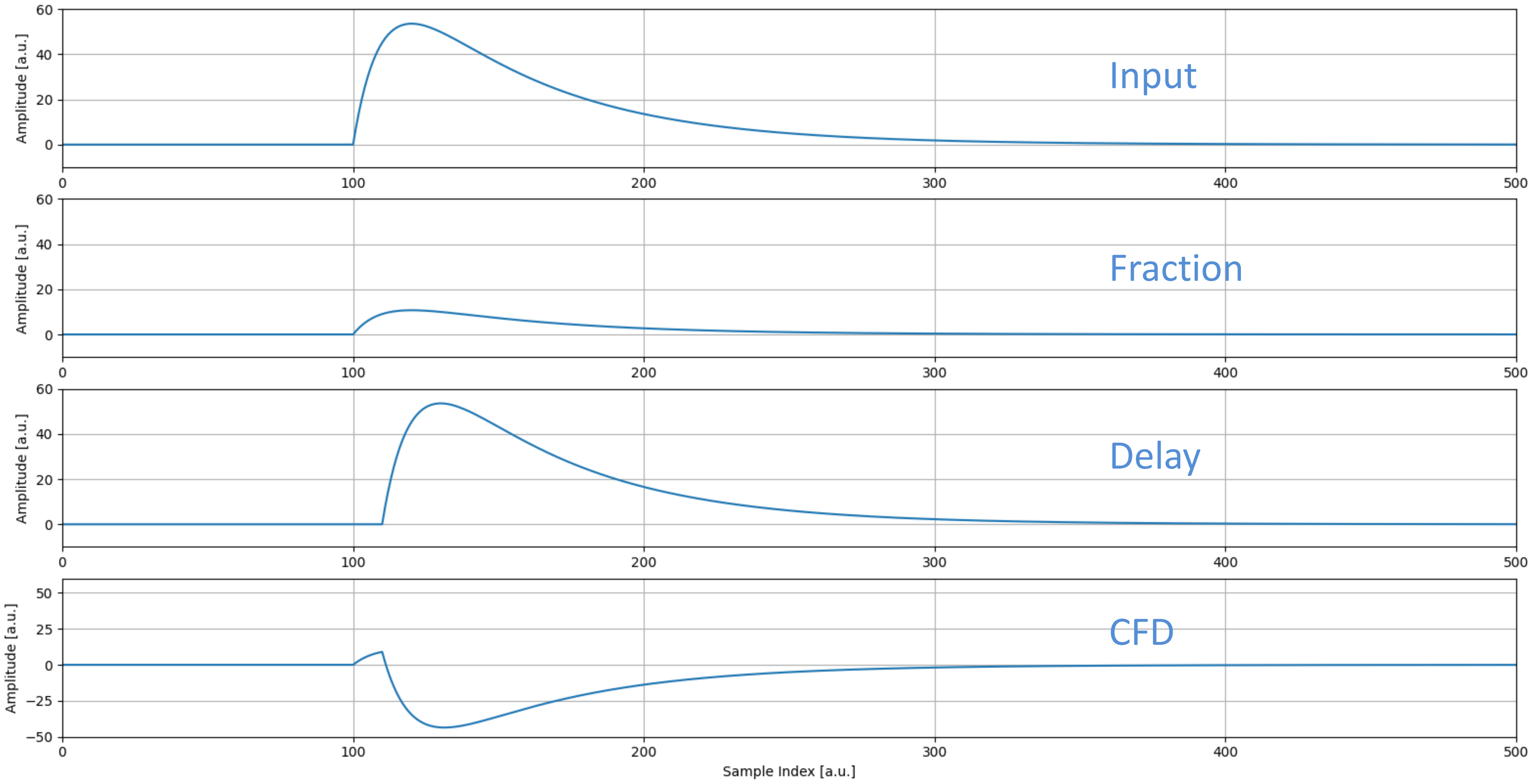
$$k\cancel{V_0}\left(e^{-t/\tau_f} - e^{-t/\tau_r}\right) = \cancel{V_0}\,(e^{-t-t_d/\tau_f} - e^{-t-t_d/\tau_r})$$

Constant Fraction Discriminator

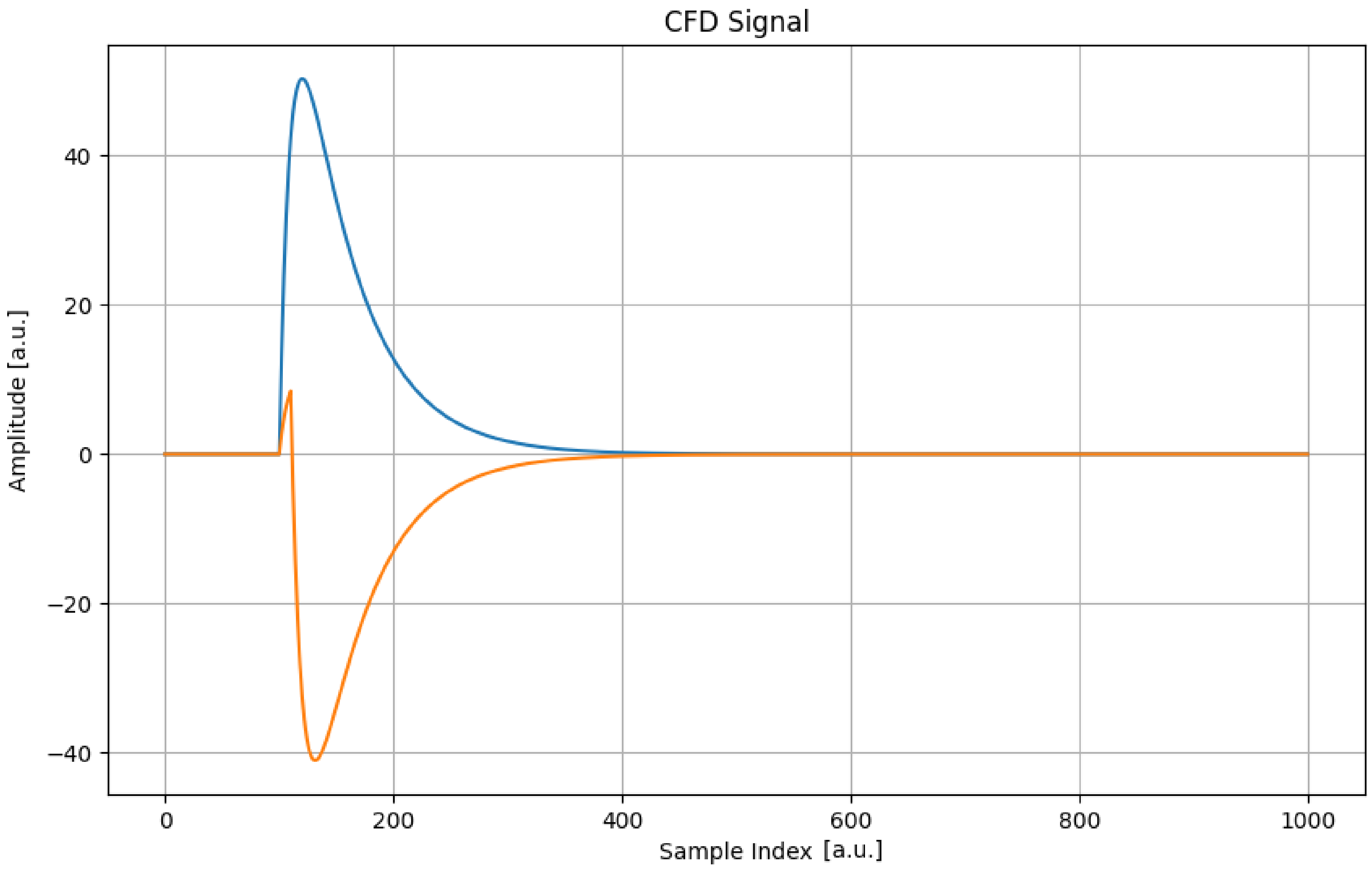
$$y[n] = k*x[n] - x[n-N]$$



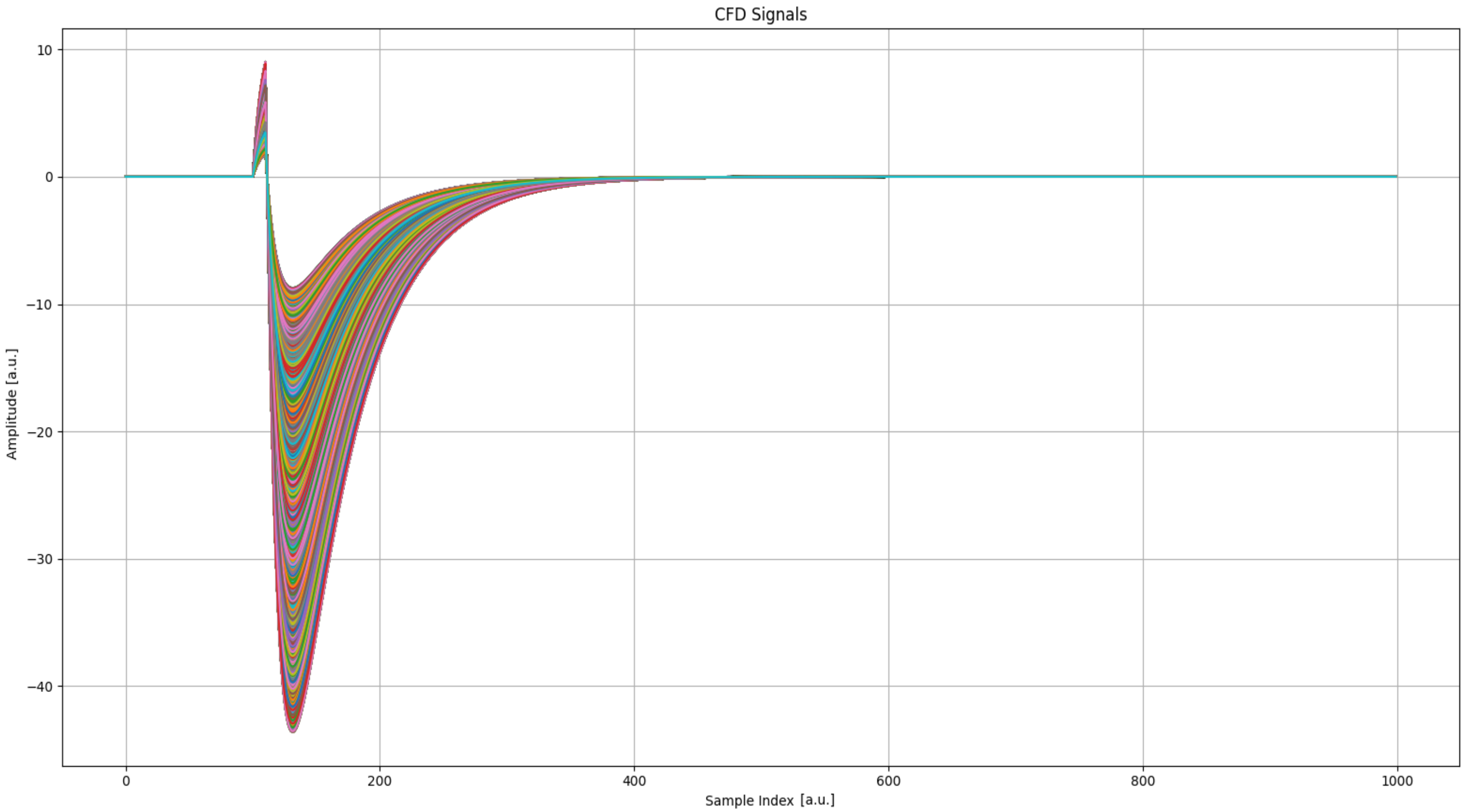
Constant Fraction Discriminator



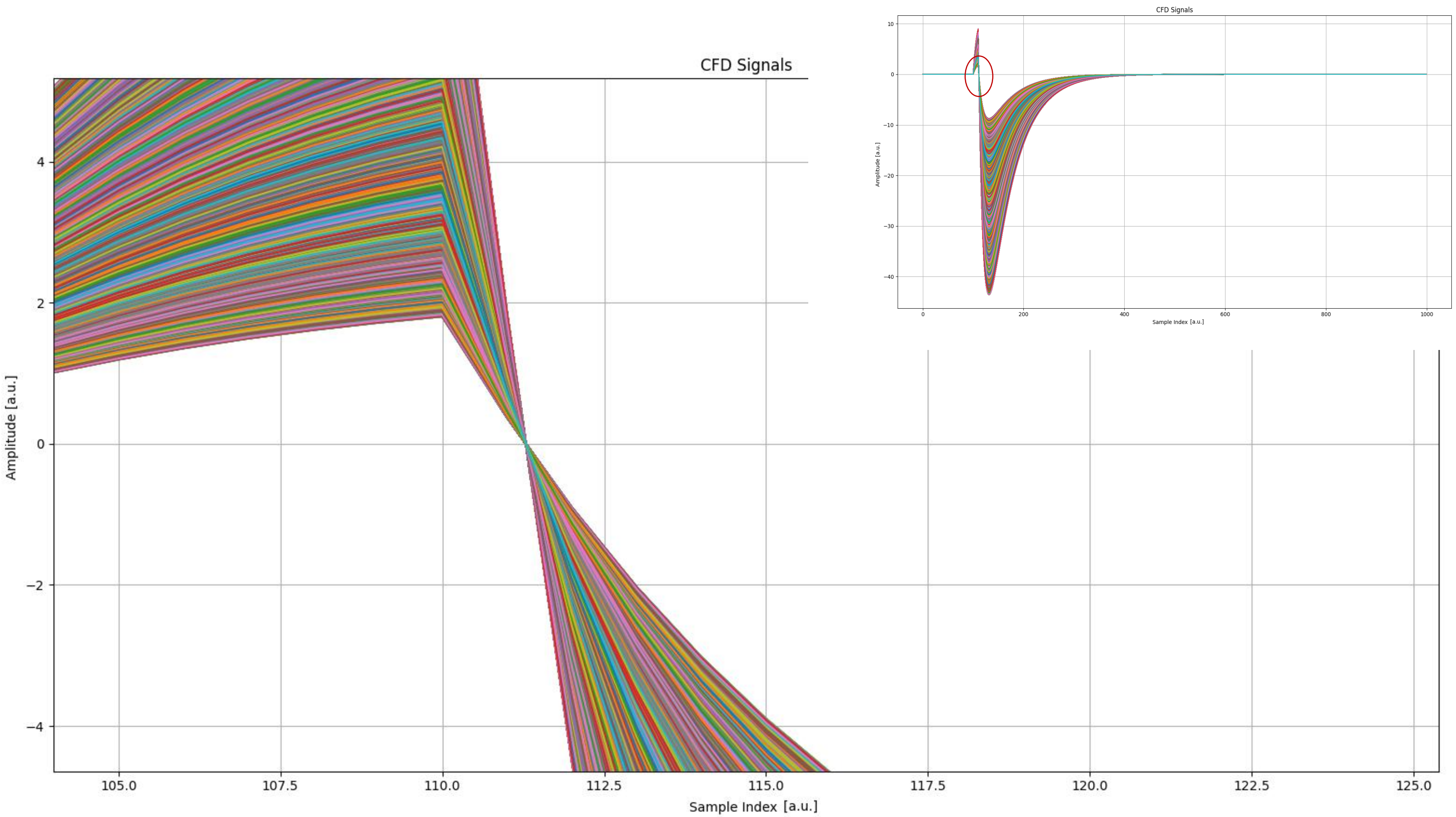
Constant Fraction Discriminator



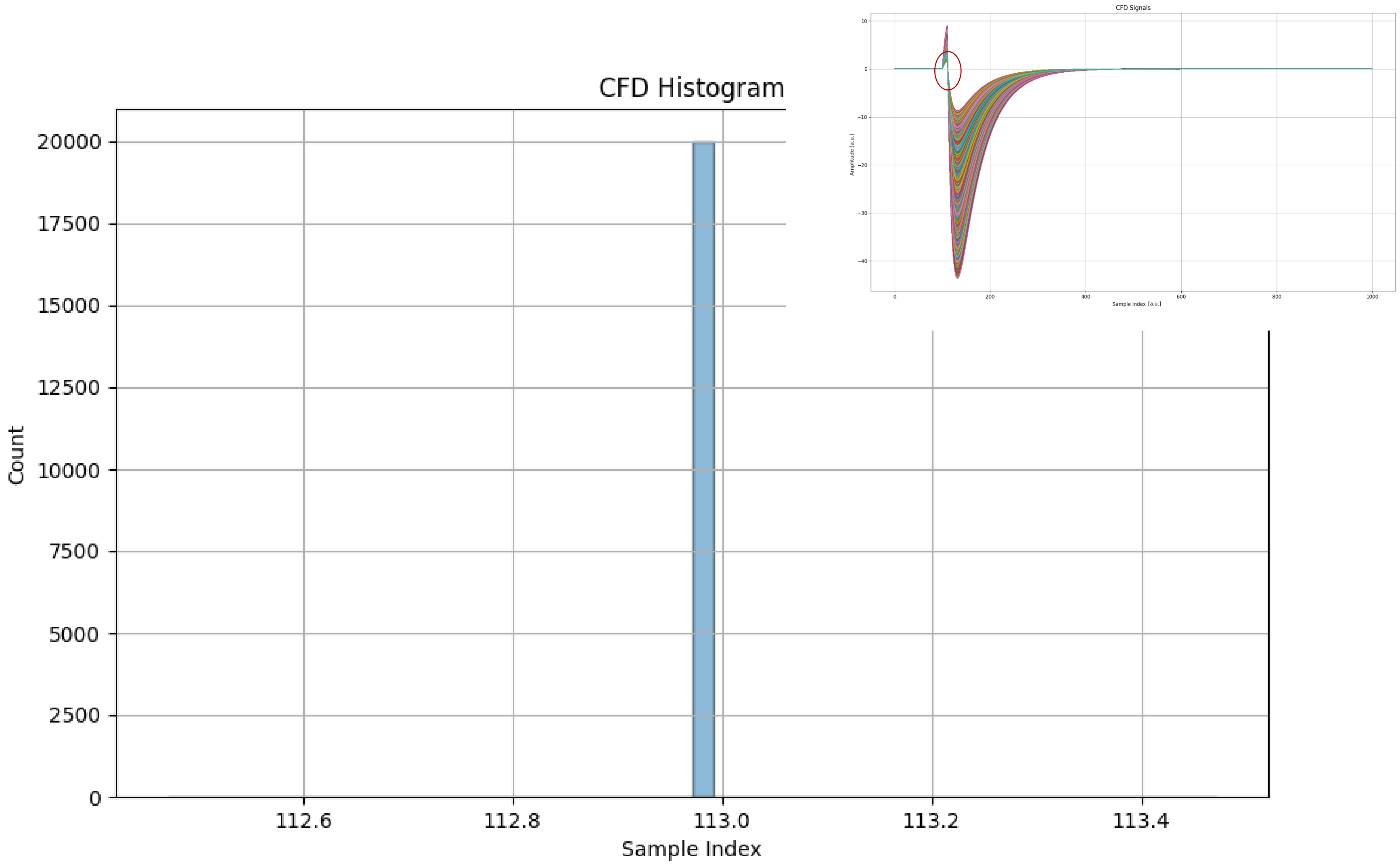
Constant Fraction Discriminator



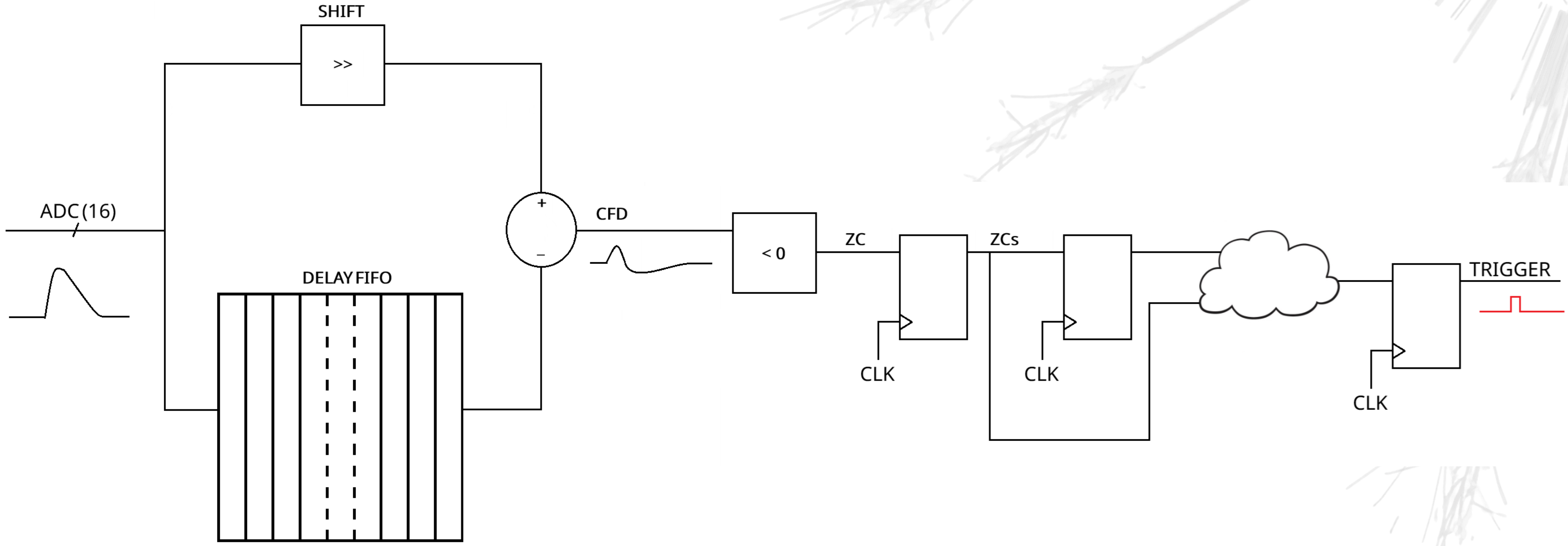
Constant Fraction Discriminator



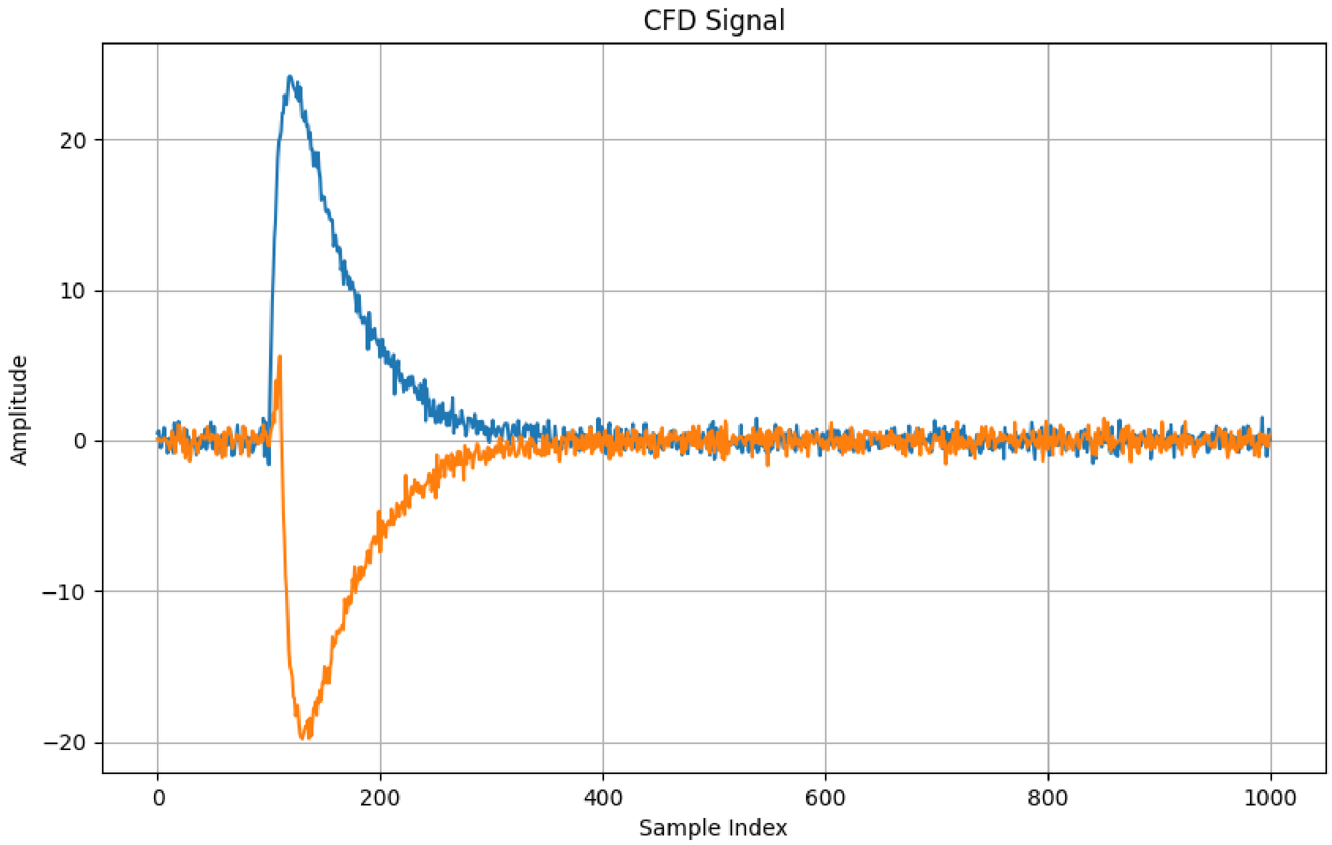
Constant Fraction Discriminator



Constant Fraction Discriminator



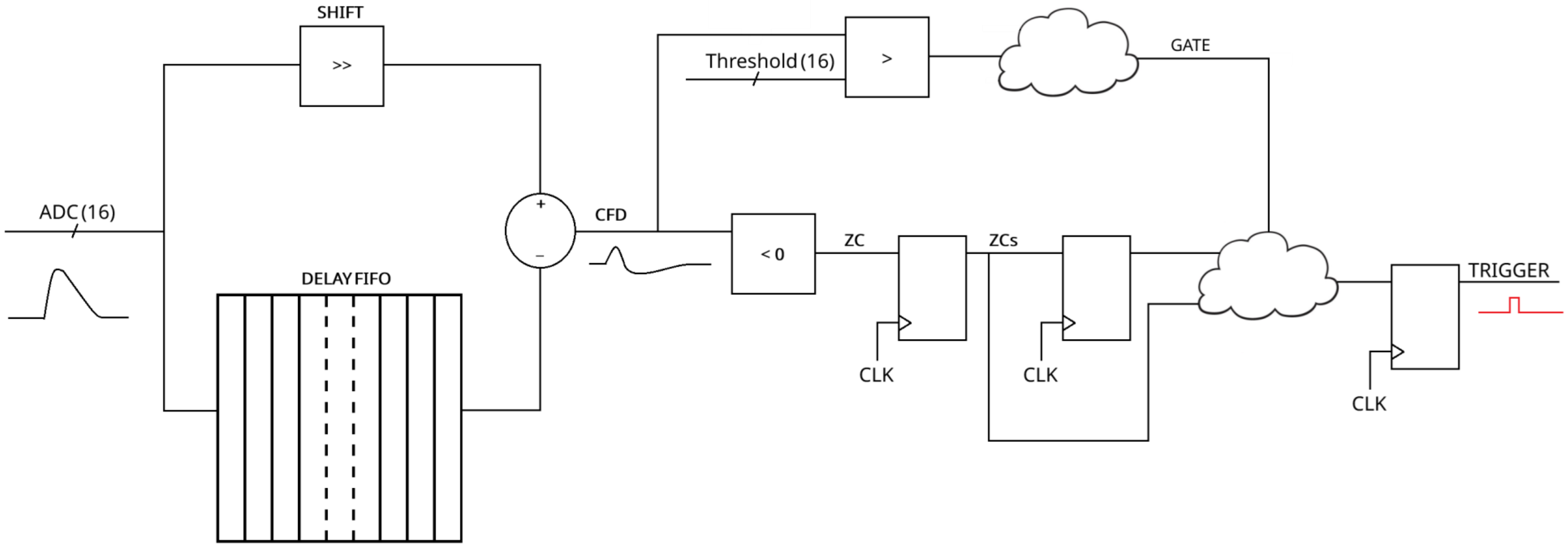
Noise Effects



Noise Effects

Constant Fraction Discriminator

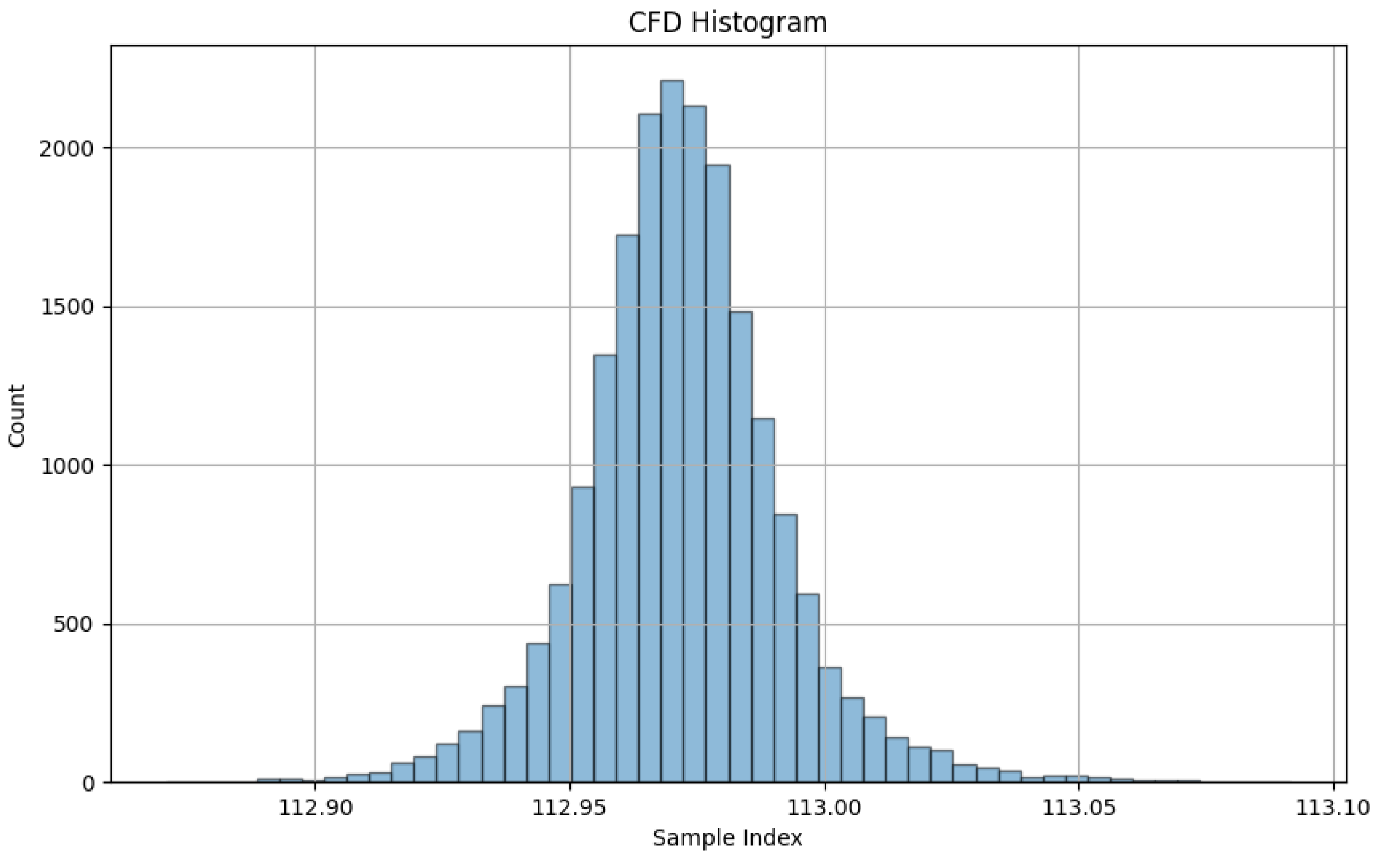
Noise Effects



CAEN
Tools for Discovery

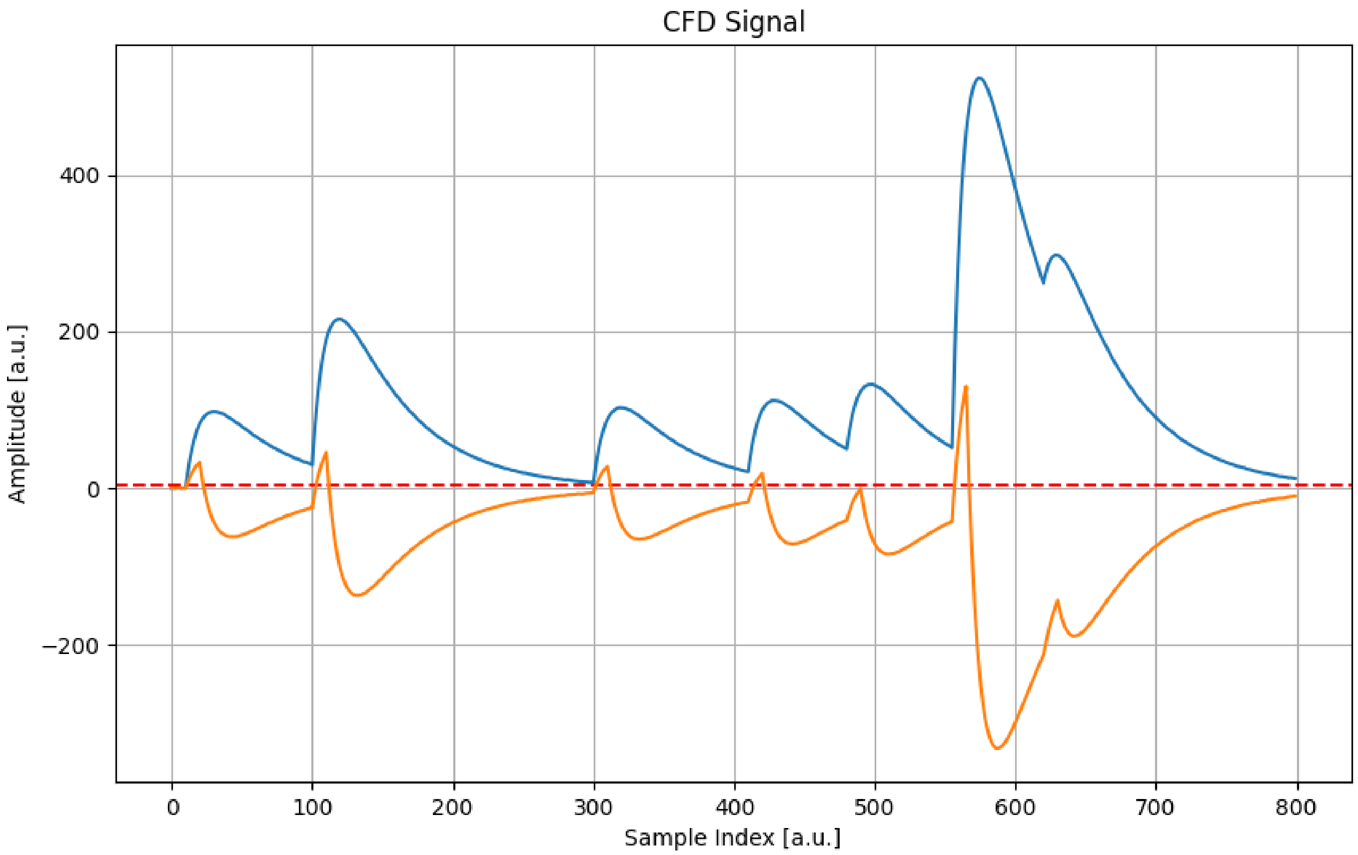
Constant Fraction Discriminator

Noise Effects



Constant Fraction Discriminator

Pile-up Problem

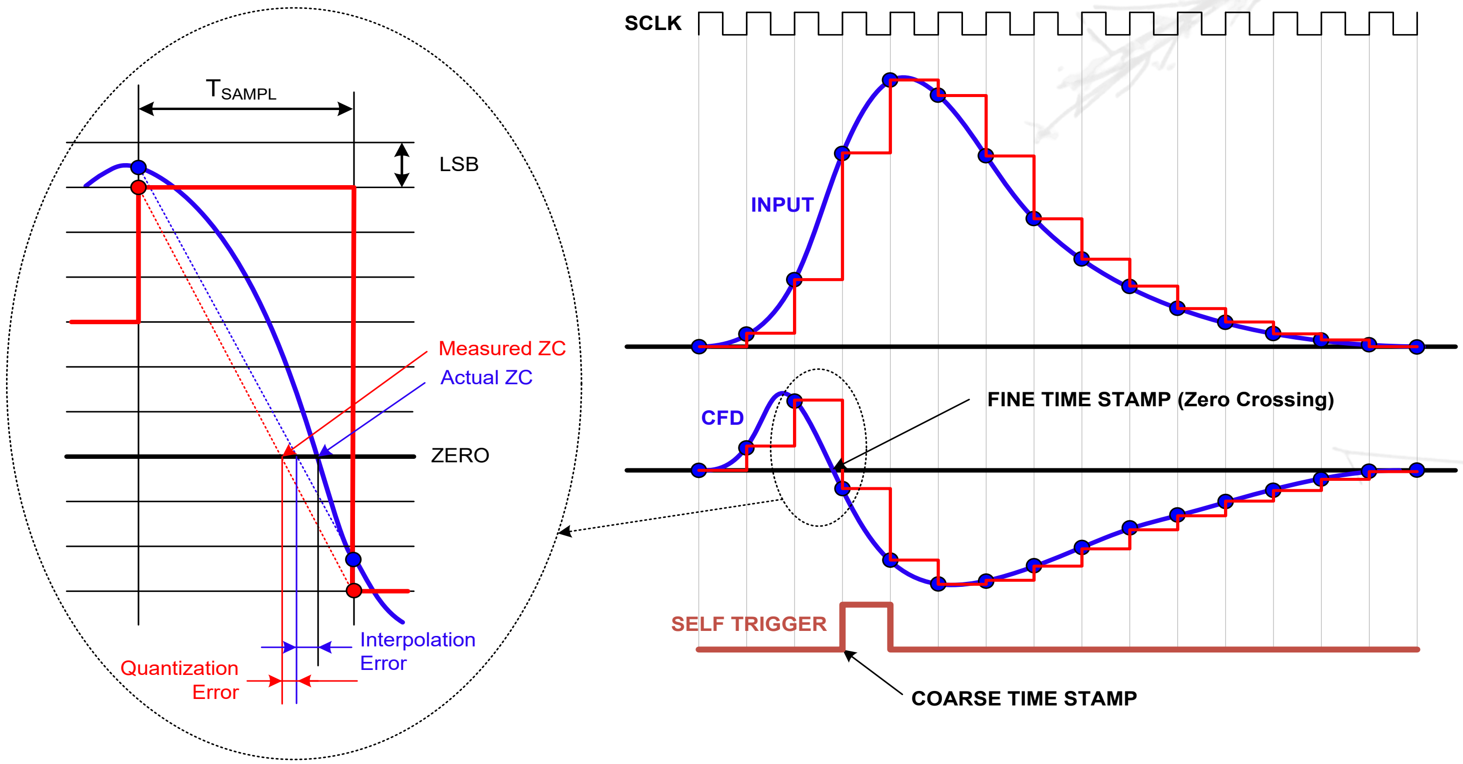


Trigger Filters

	Leading Edge Discriminator	Constant Fraction Discriminator
Trigger Criterion	Absolute Threshold	Fixed Fraction
Time Walk	Significant	Eliminated
Complexity	Minimal	Medium
Rate Sensitivity	High Rate Pile Up	High Rate Pile Up
Application	First Level Trigger, Coarse Timing	ToF, Coincidence, Fine Timing

Sub-Clock Interpolation

What about a Coarse Time Stamp improvement?

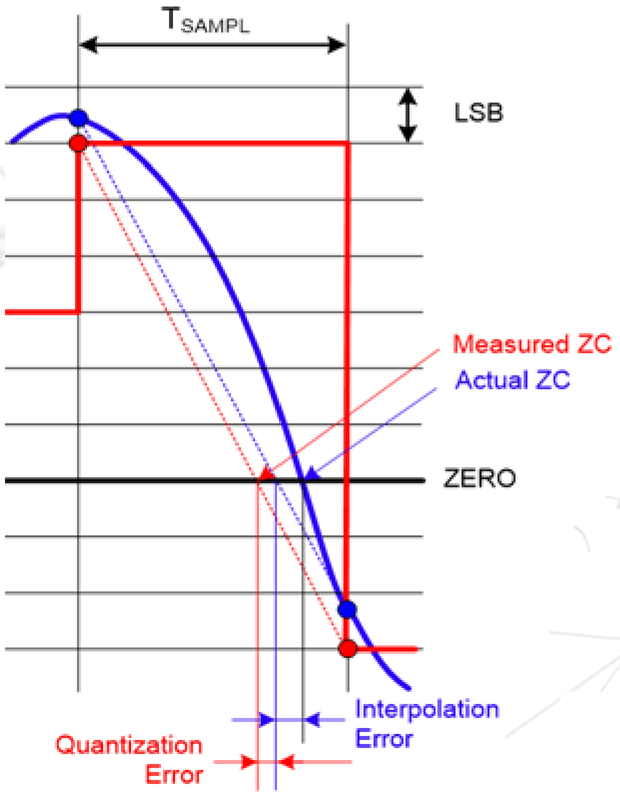


CAEN Tools for Discovery

- The ADC gives you integers at fixed intervals
- The zero-crossing lives somewhere in between two integers
- Linear interpolation gives the fractional offset δ:

$$\delta = \frac{y[n]}{y[n] - y[n+1]}$$

- Timing resolution then becomes $\delta \cdot T_s$, well below the sampling period
- **On an FPGA: no floating-point division**



Sub-Clock Interpolation

Idea: Binary search between two samples — no divider, no float, deterministic latency.

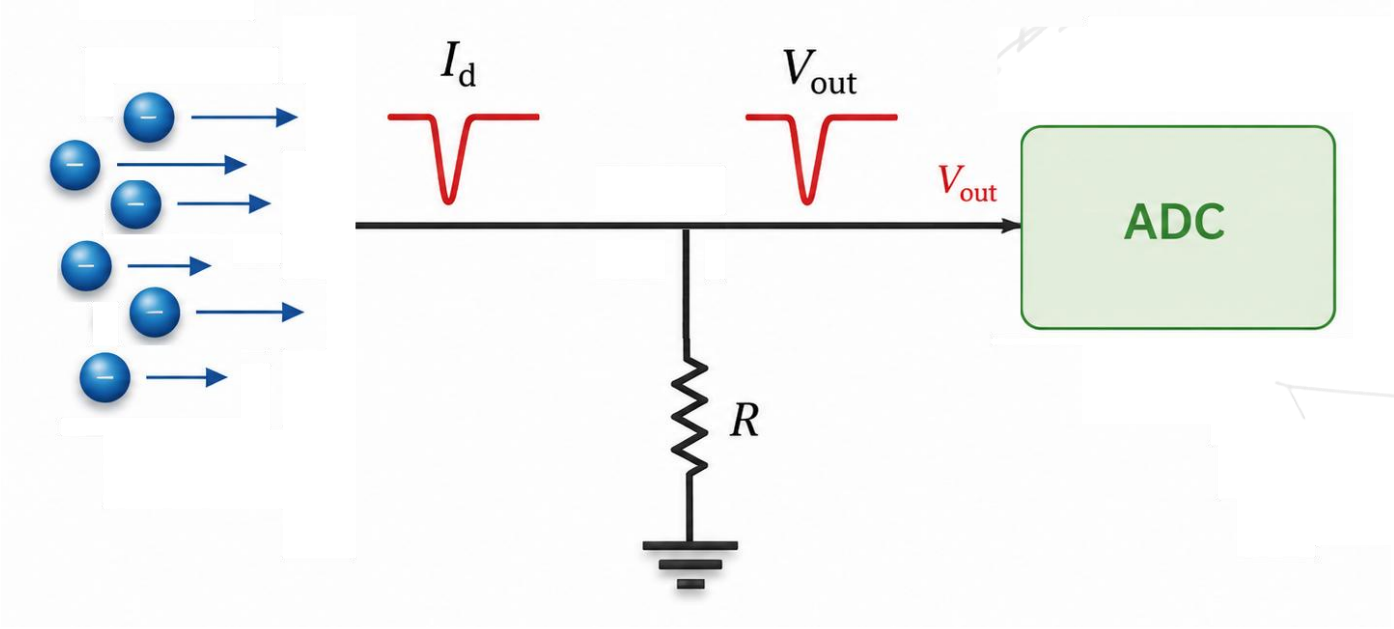
1. Evaluate the midpoint m (sum + shift)
2. If m > 0, zero crossing is in the second half
3. If m < 0, zero crossing is in the first half
4. Repeat for k-iterations

SUM and SHIFT are easy on FPGA

Energy Filters

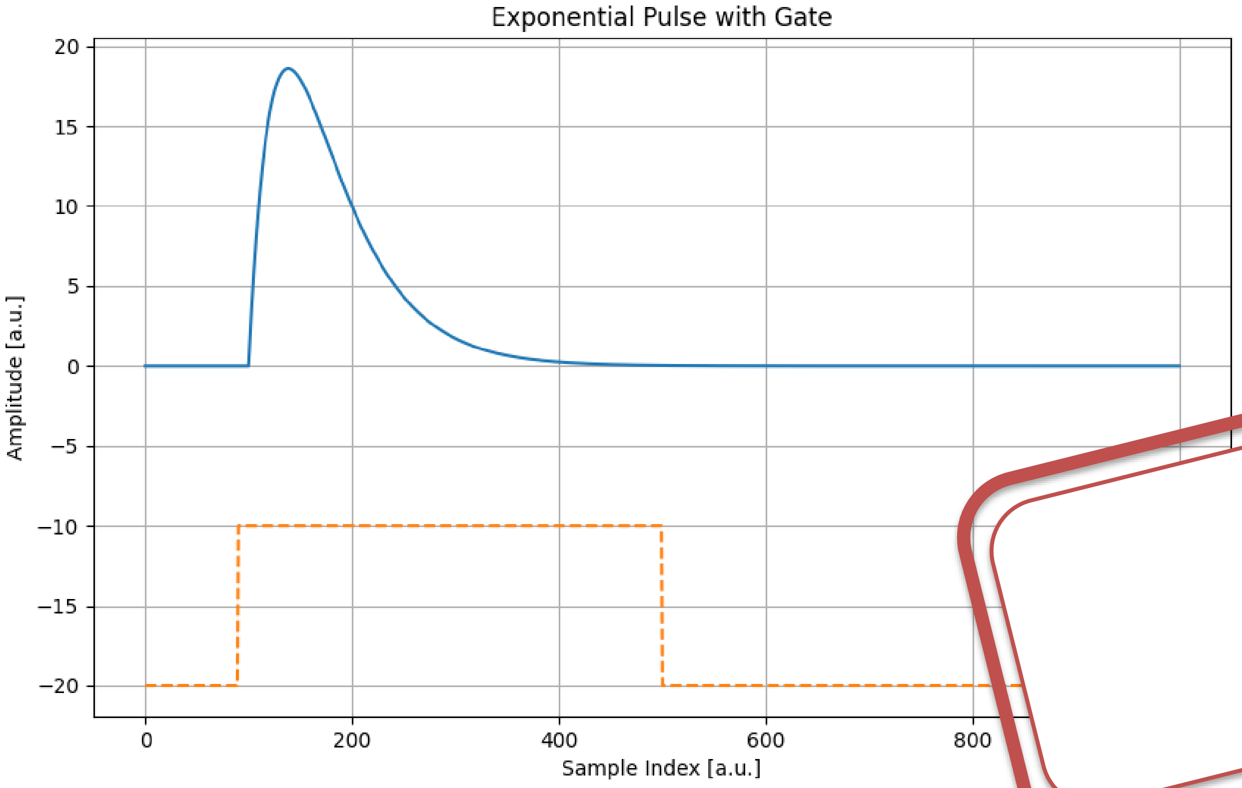
Charge Integration

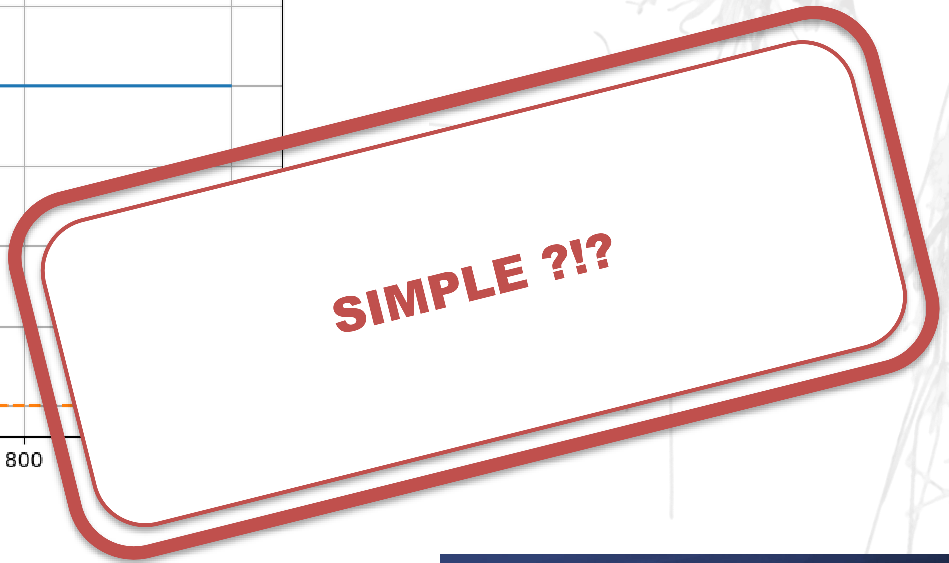
$$Q = \sum_{n=a}^{a+N} x[n]$$



Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$

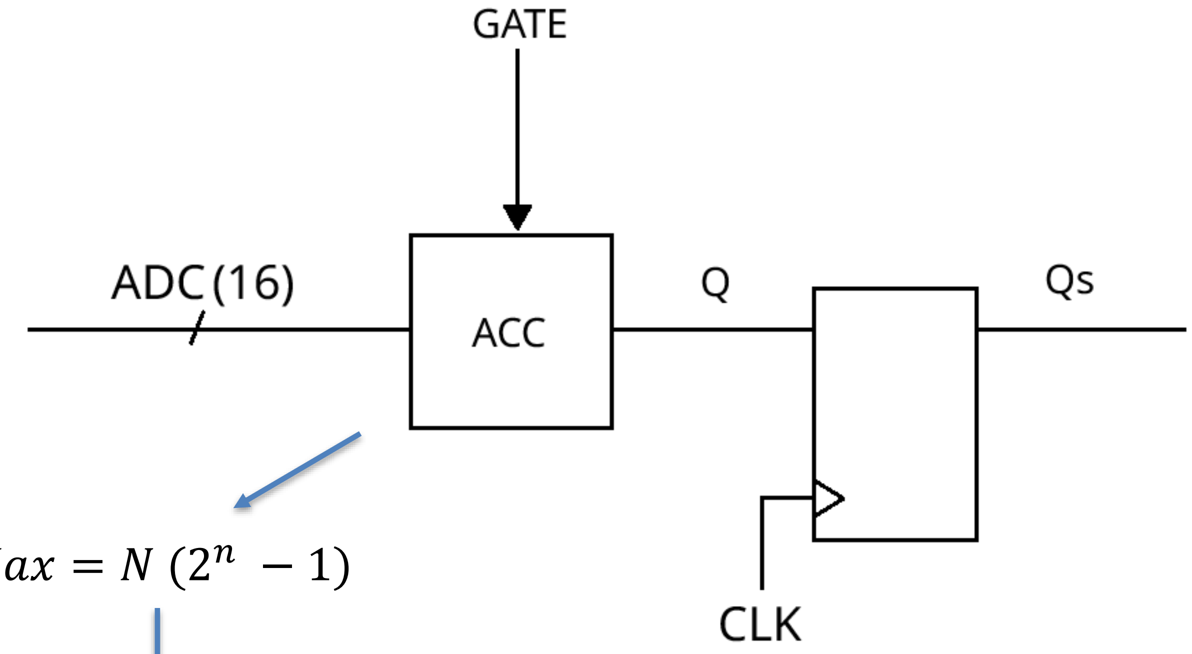




CAEN Tools for Discovery

Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$

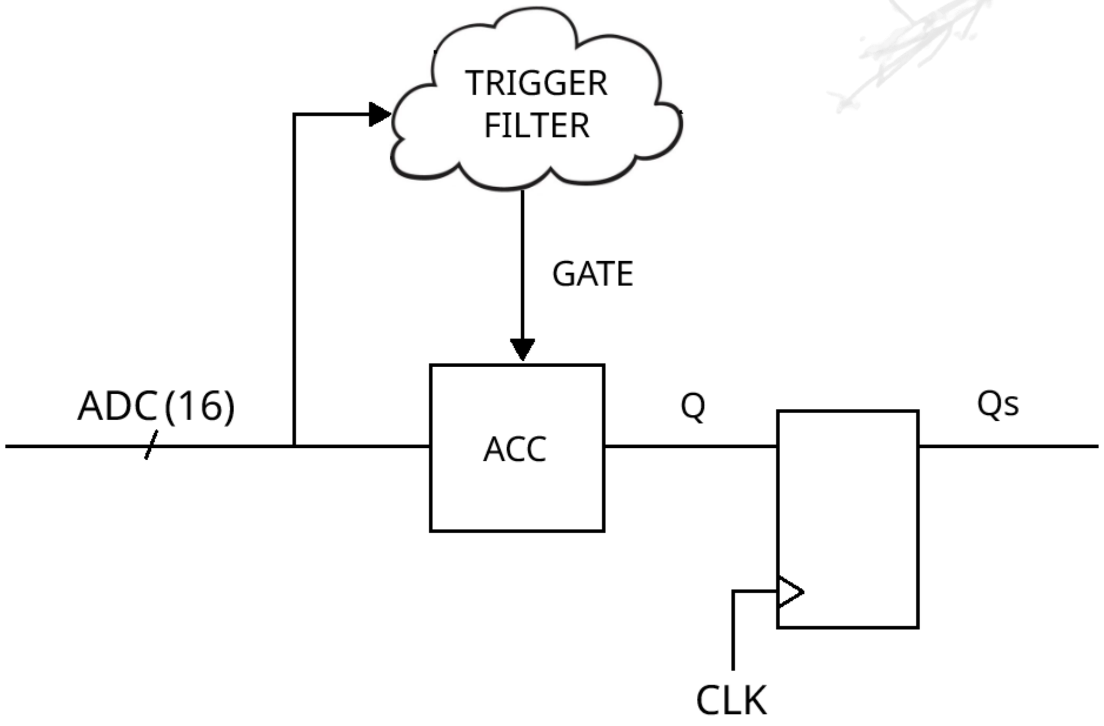


$$Max = N\,(2^{n} - 1)$$

$$Bits = n + log_2(N)$$

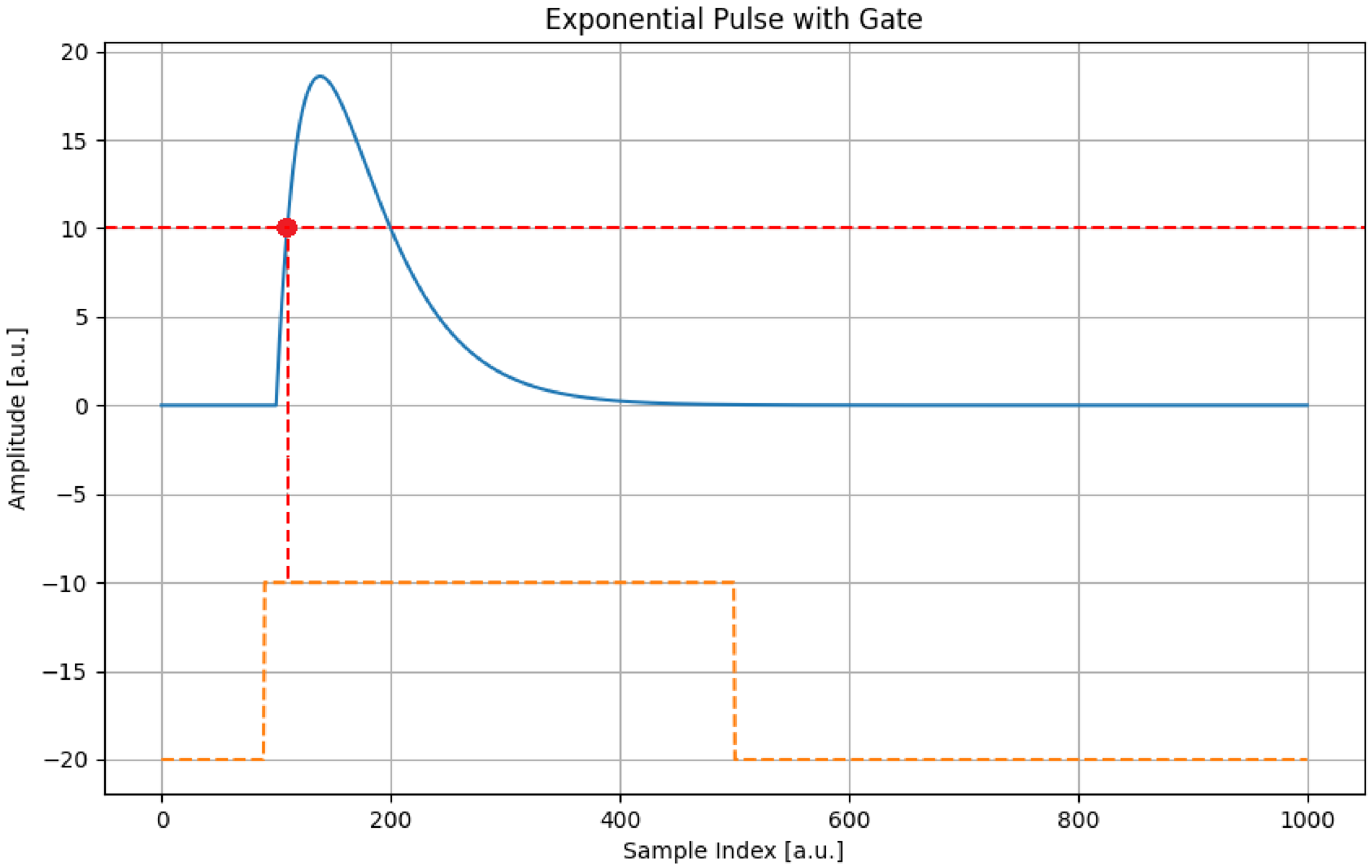
Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$



Charge Integration

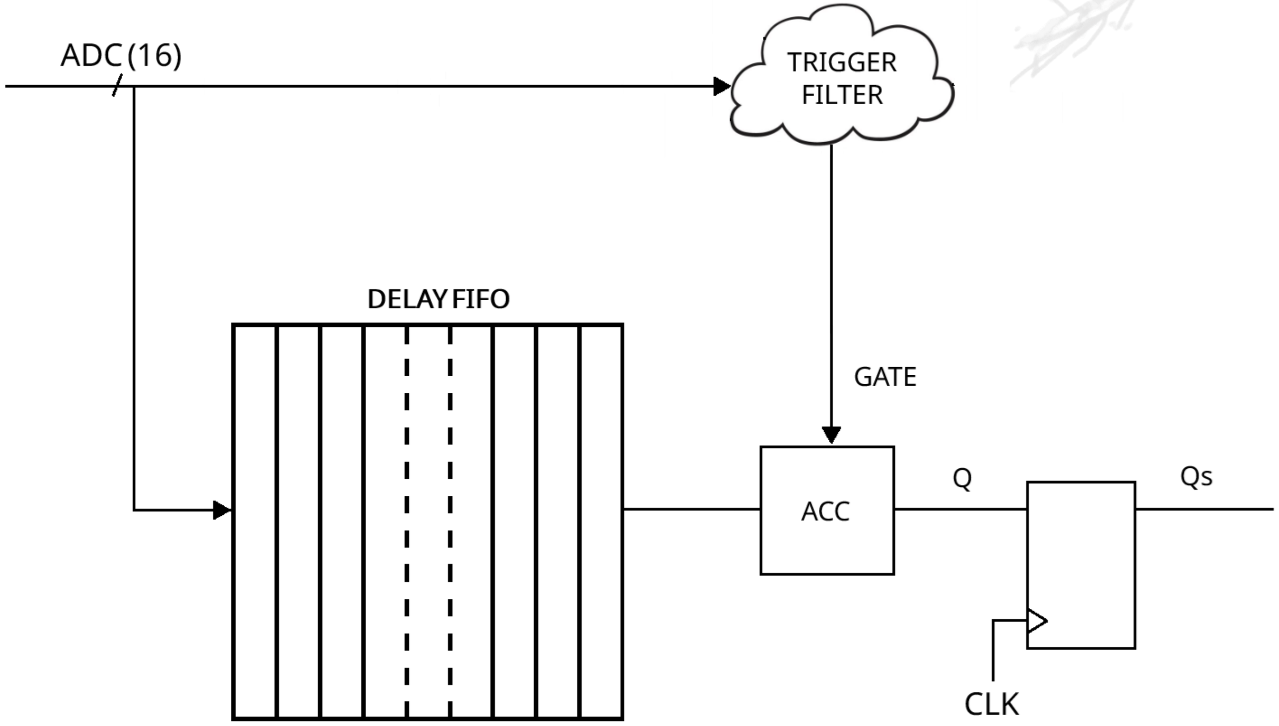
$$Q = \sum_{n=a}^{a+N} x[n]$$



CAEN
Tools for Discovery

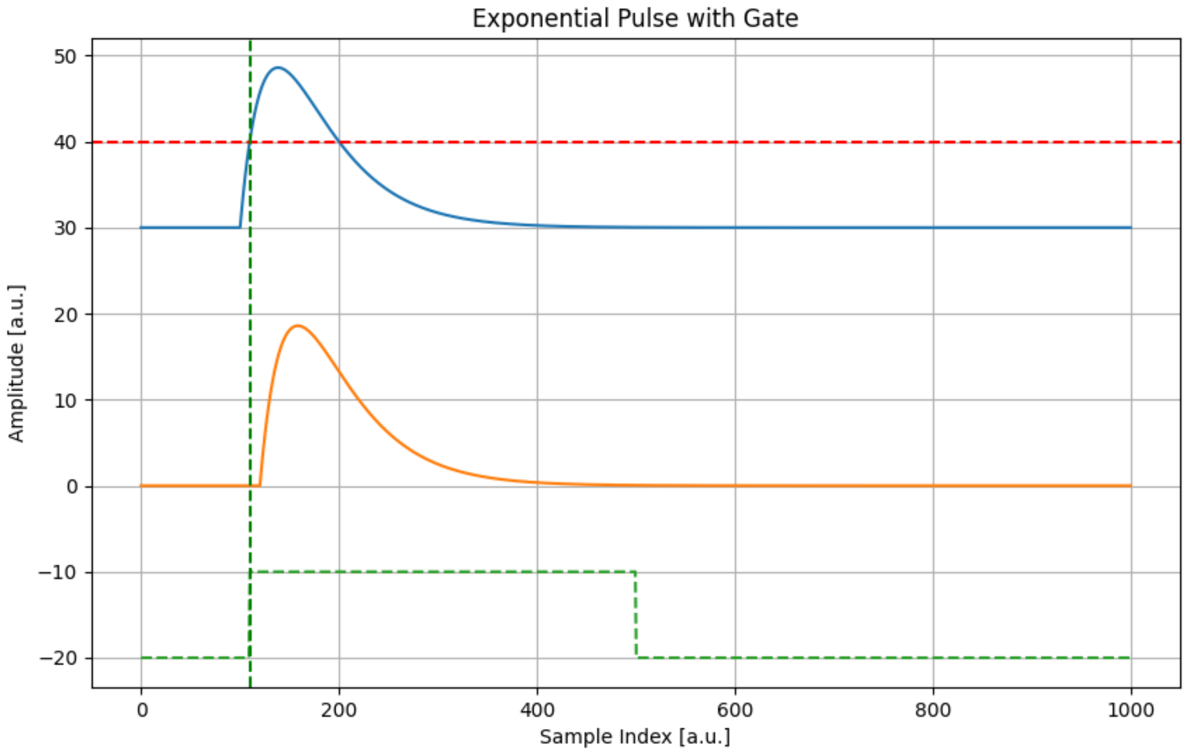
Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$



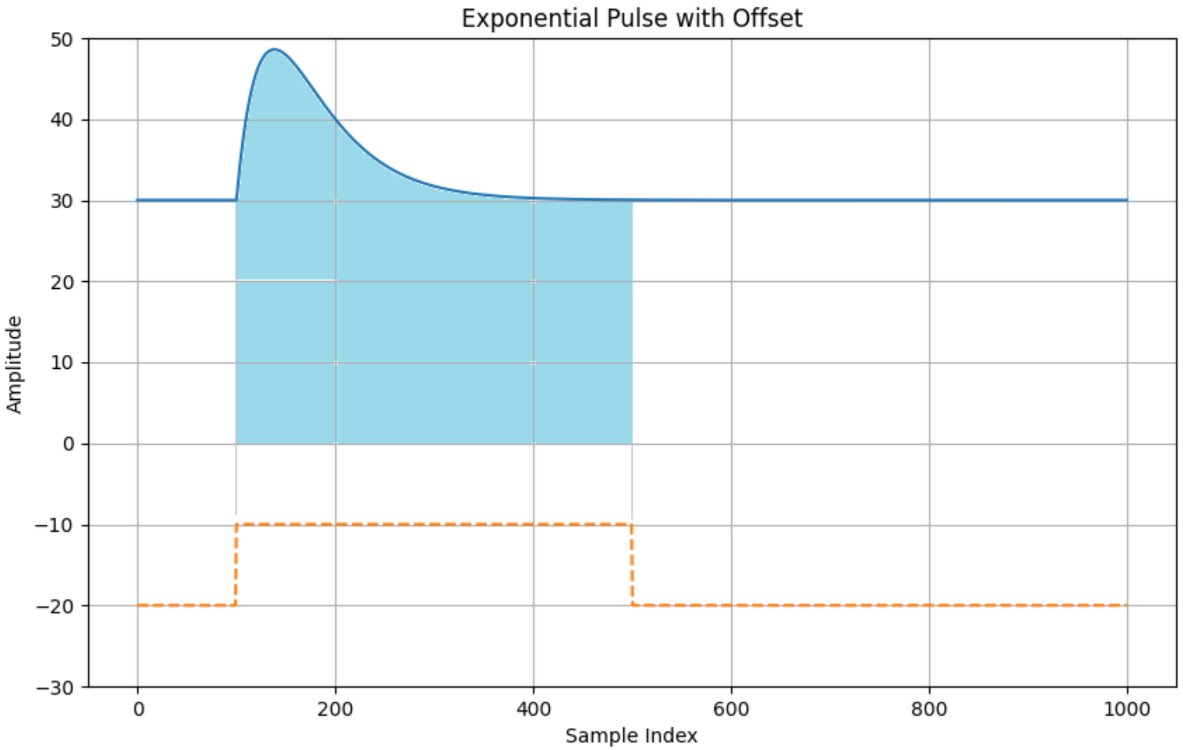
Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$



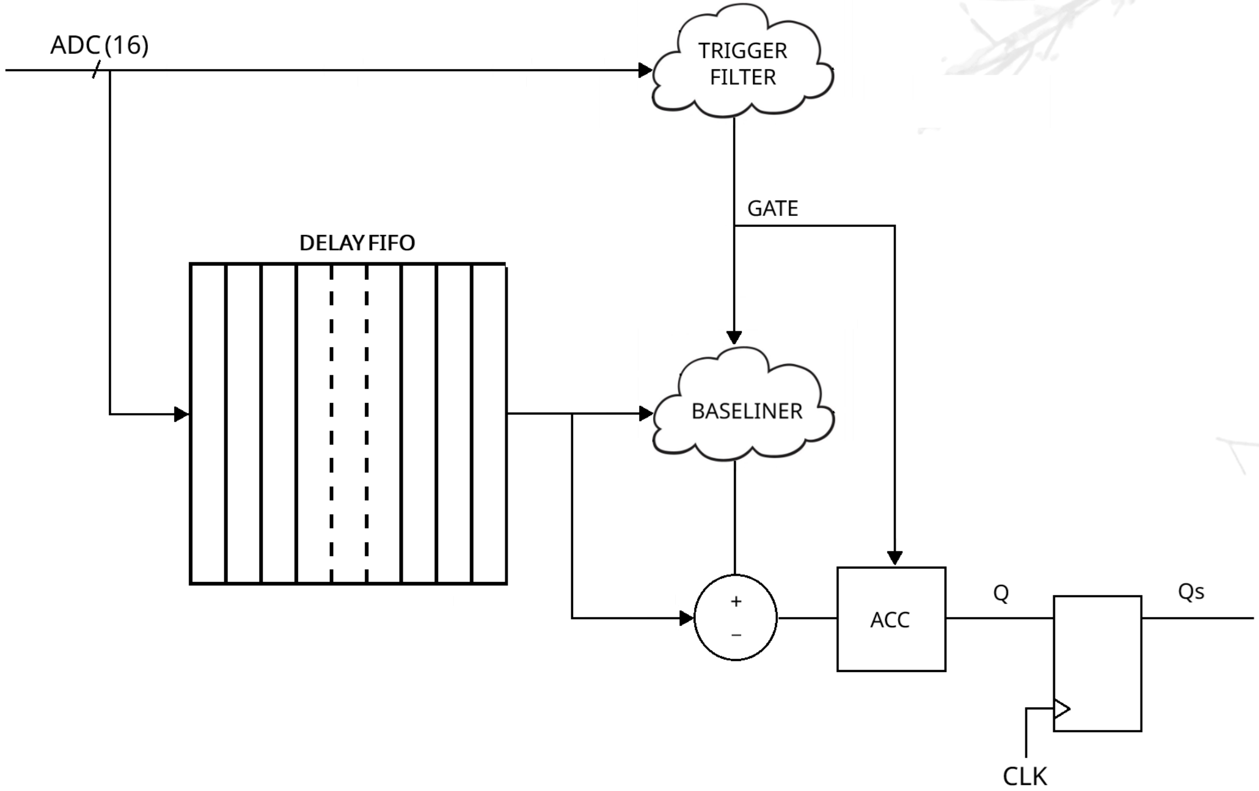
Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$



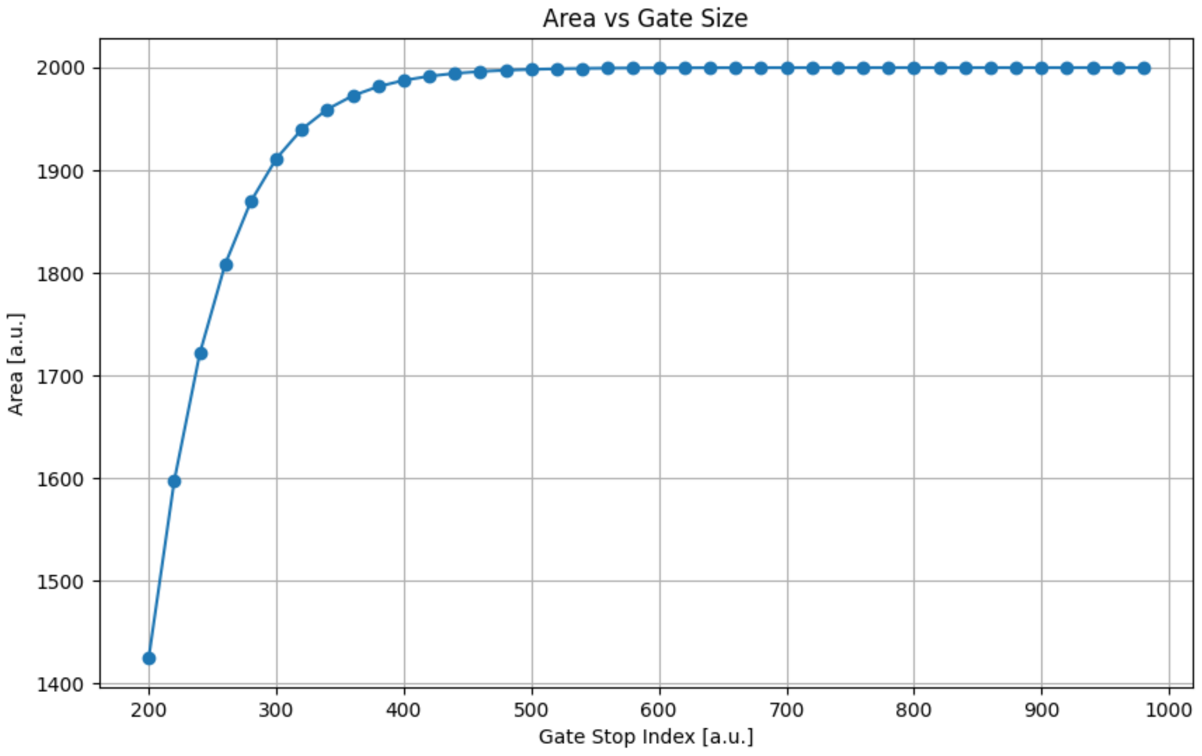
Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$



Charge Integration

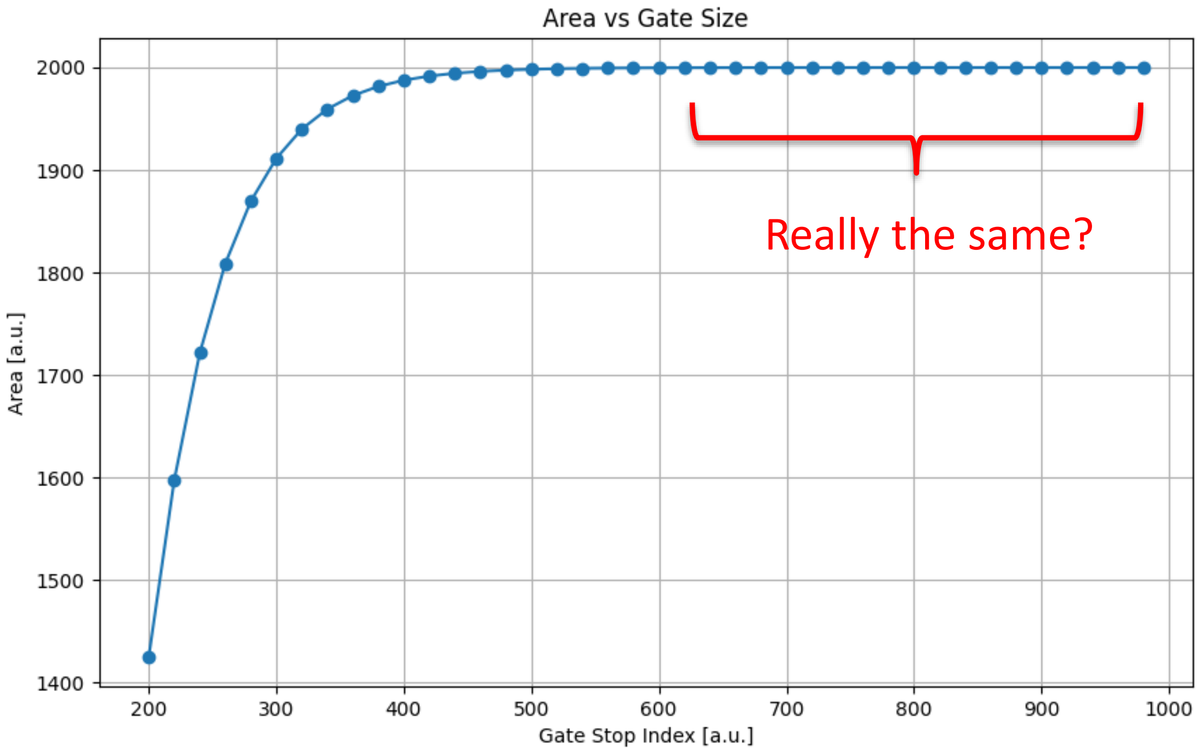
$$Q = \sum_{n=a}^{a+N} x[n]$$



CAEN
Tools for Discovery

Charge Integration

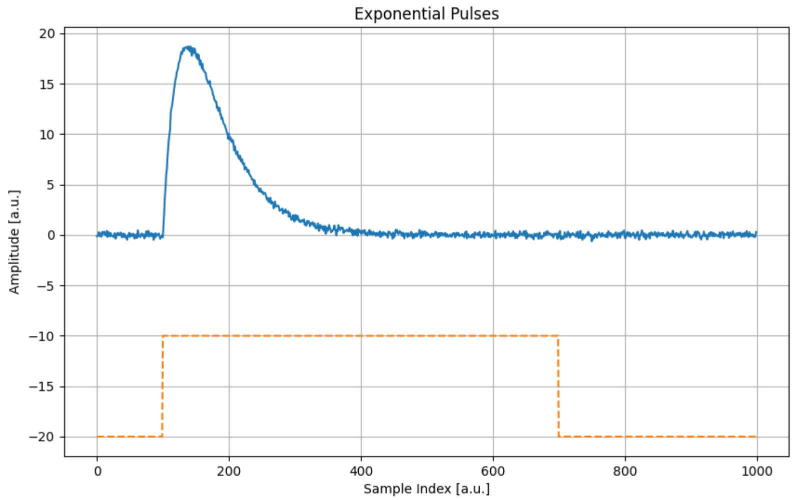
$$Q = \sum_{n=a}^{a+N} x[n]$$

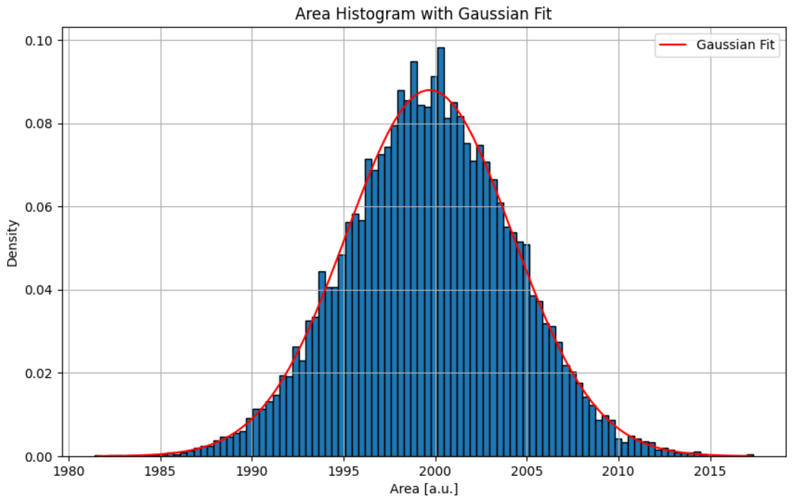


Charge Integration

$$Q = \sum_{n=a}^{a+N} x[n]$$

$$\sigma_{tot}^2 = \sum_{n=a}^{a+N} \sigma^2 = N\sigma^2$$

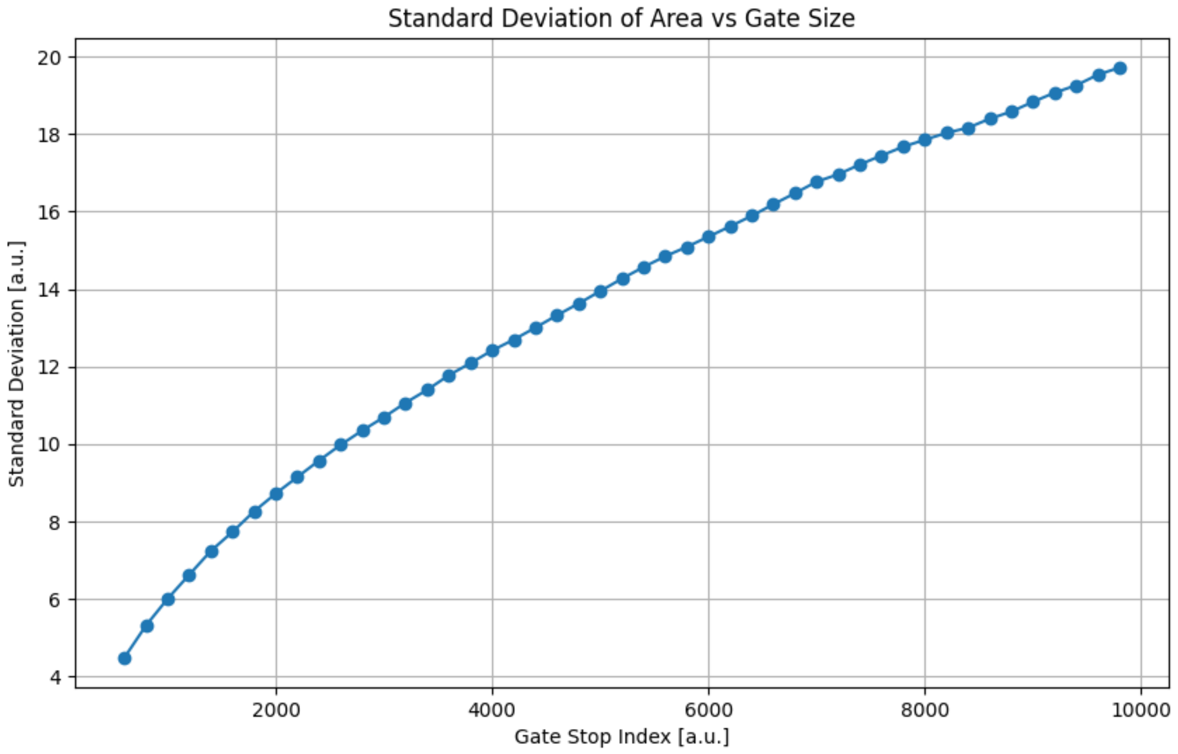




CAEN
Tools for Discovery

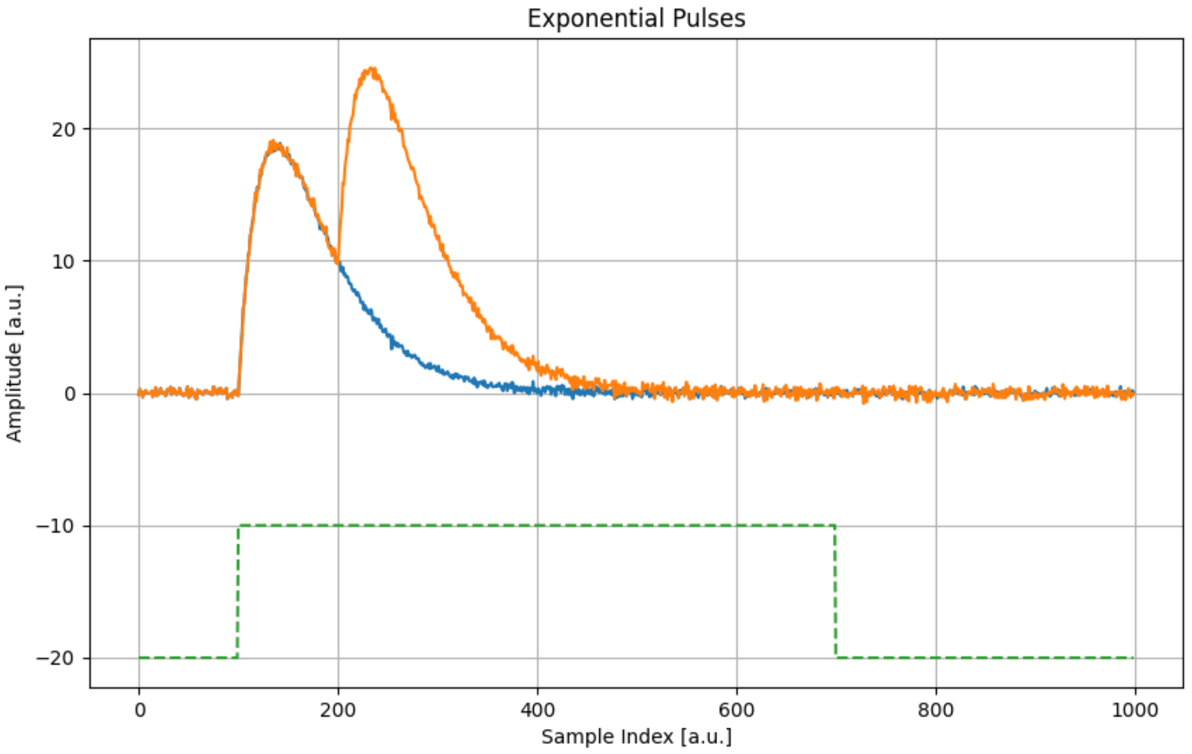
Charge Integration

$$\sigma_{tot} = \sigma\sqrt{N}$$



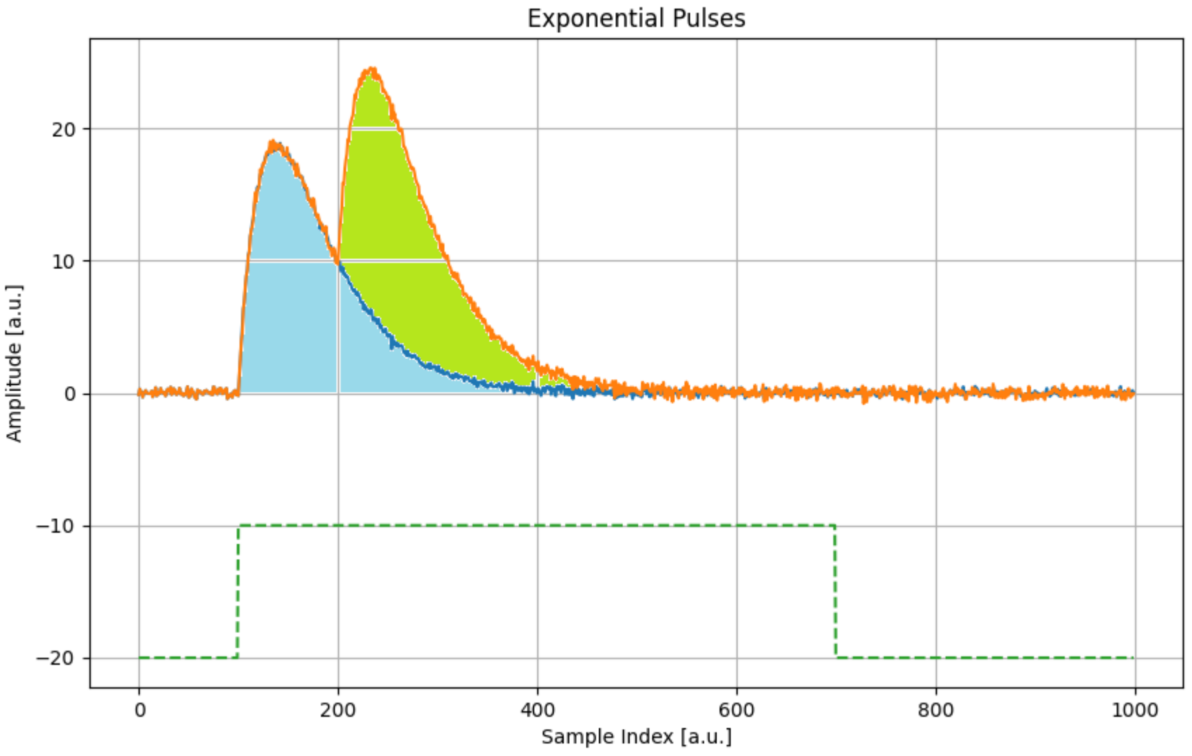
Charge Integration

Pile-up Limit



Charge Integration

Pile-up Limit



Charge Integration

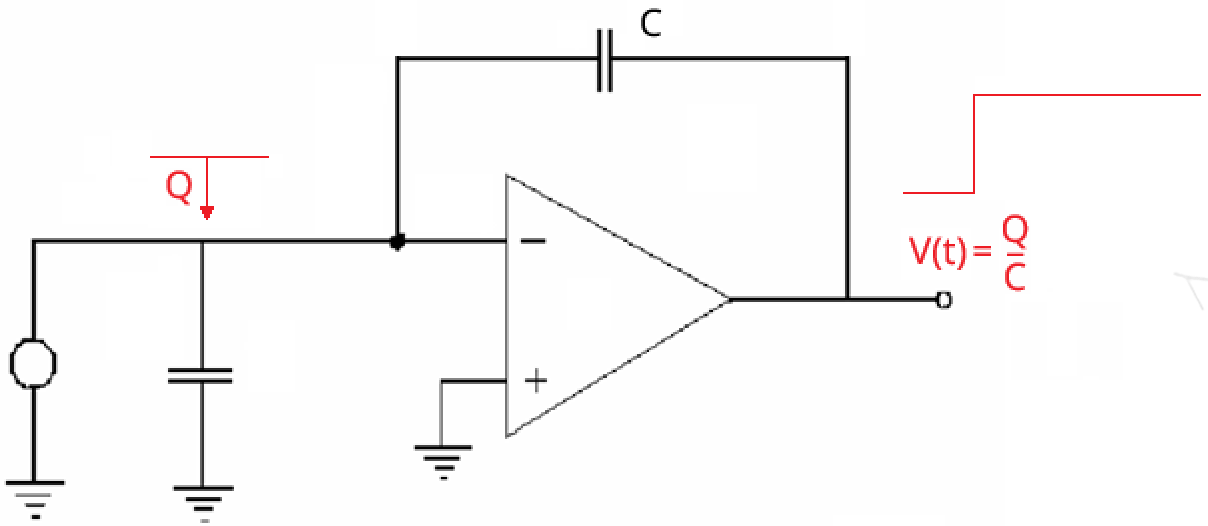
PROs	CONs
• Hardware Efficiency	• No bandpass filter
• Minimal Parametric Option	• Noise Penalty
• Shape Independence	• Pile-up Vulnerability

CAEN
Tools for Discovery

… but what if the charge integration is done by a charge preamplifier ?!?

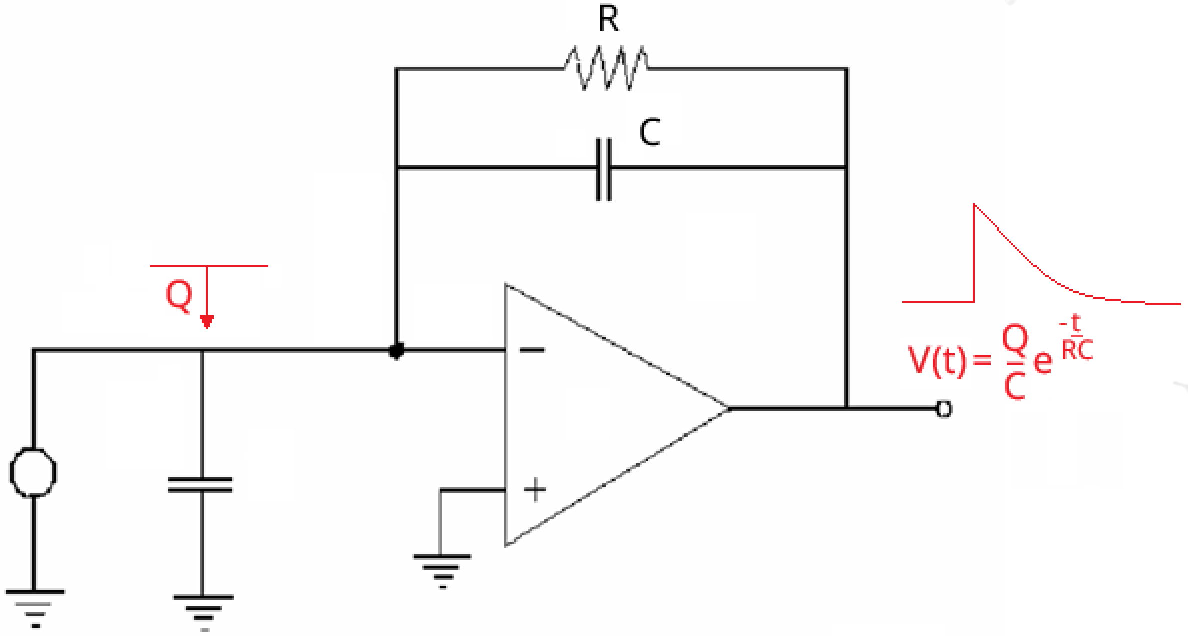
Charge Sensitive Preamplifier

$$V(t) = \begin{cases} 0, & t < 0 \\ \frac{Q}{C}, & t \geq 0 \end{cases}$$



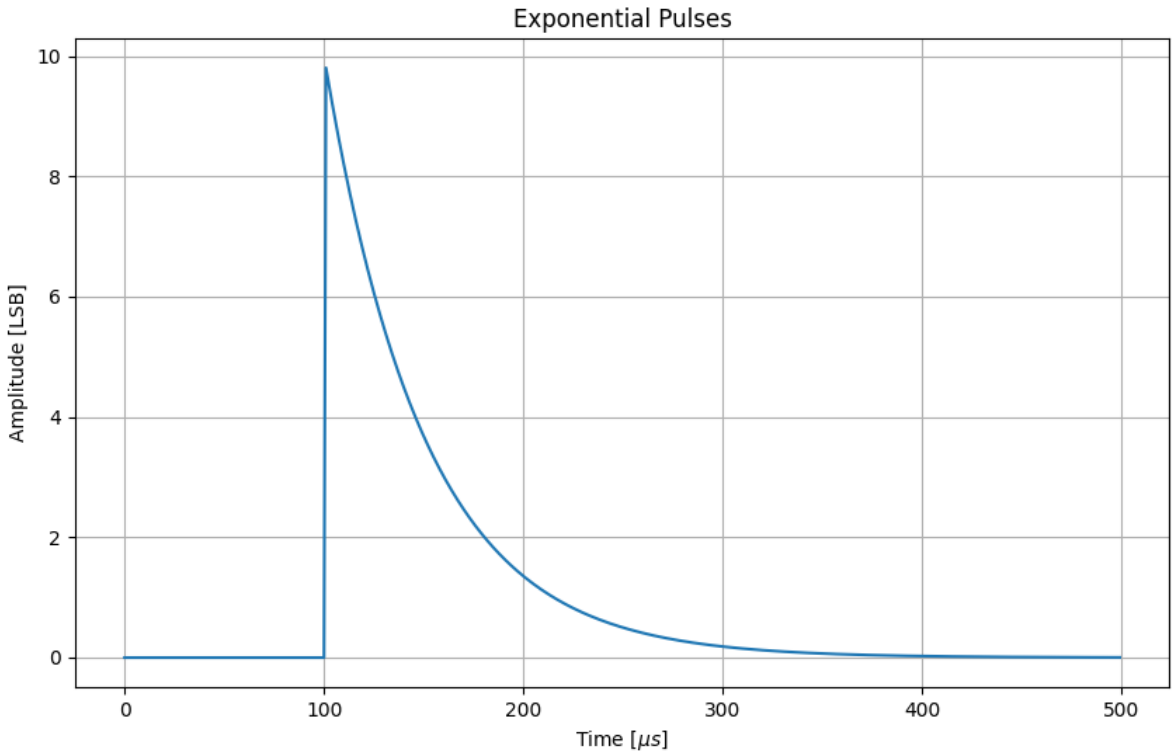
Charge Sensitive Preamplifier

$$V(t) = \begin{cases} 0, & t < 0 \\ \frac{Q}{C} e^{-\frac{t}{RC}}, & t \geq 0 \end{cases}$$



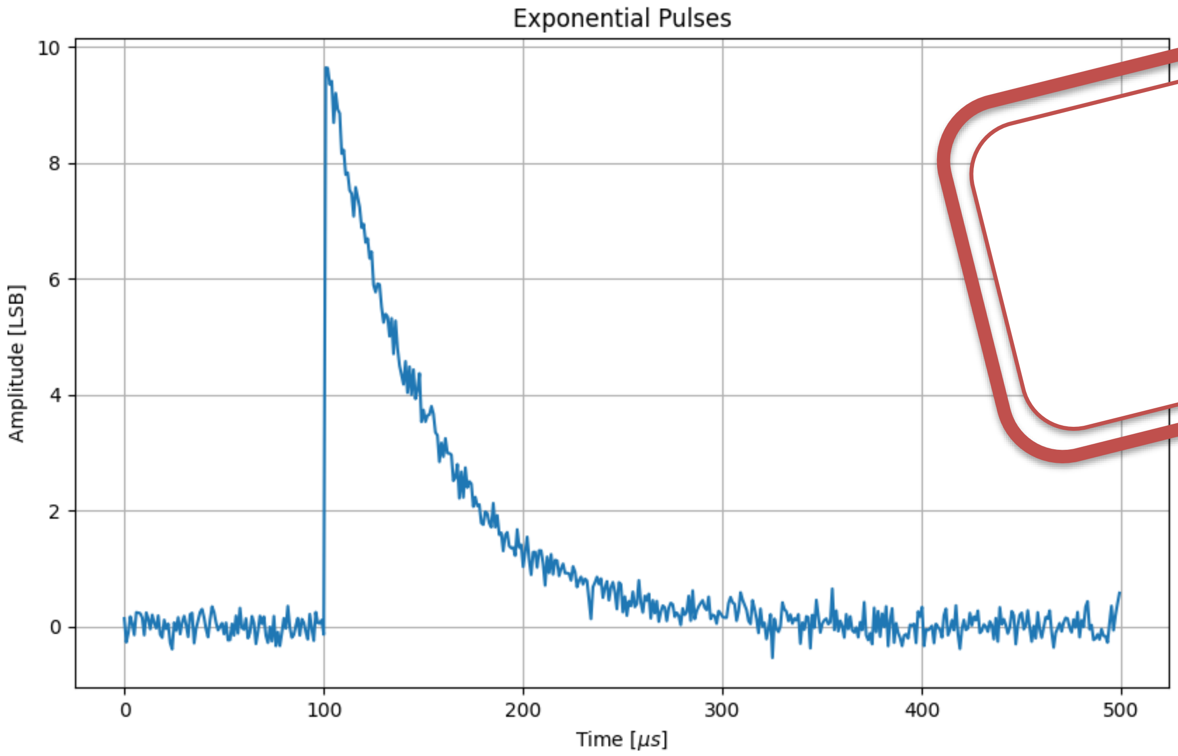
Charge Sensitive Preamplifier

$$Q \propto Vma_x$$



Charge Sensitive Preamplifier

$$Q \propto Vmax$$

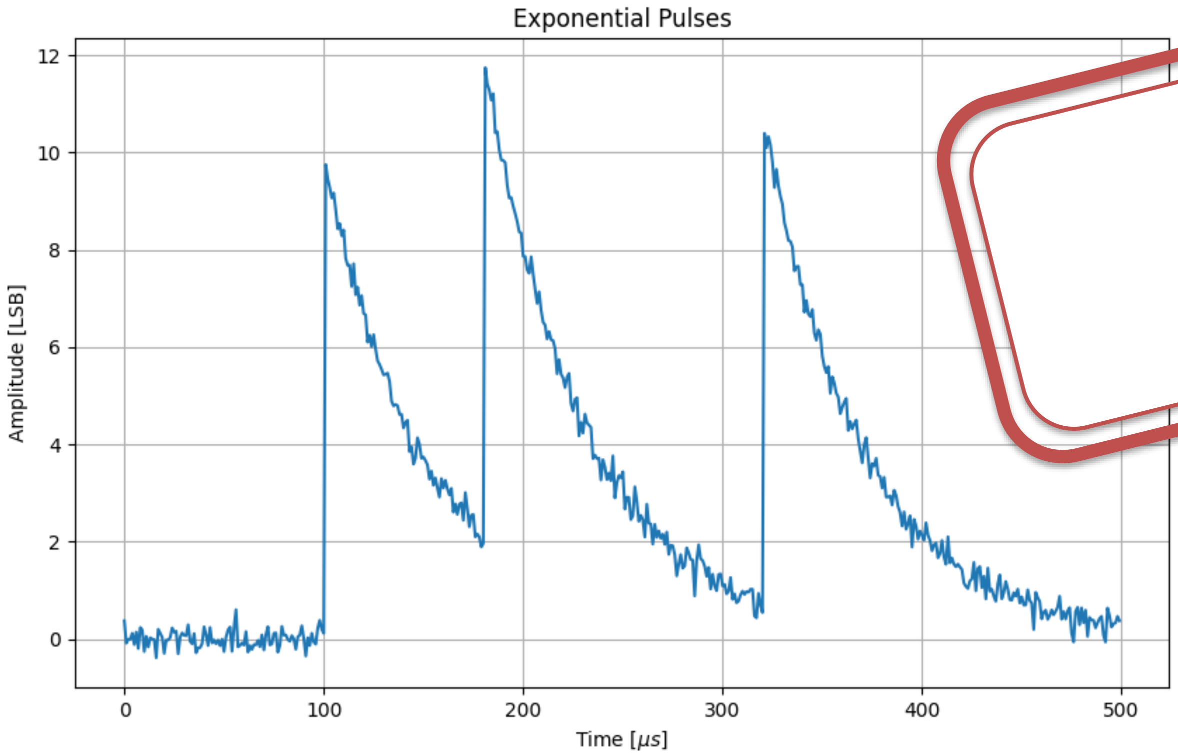


NOISE

CAEN
Tools for Discovery

Charge Sensitive Preamplifier

$$Q \propto Vmax$$

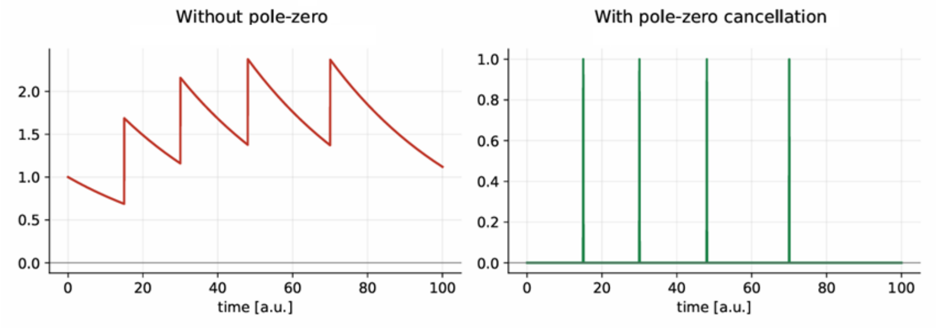


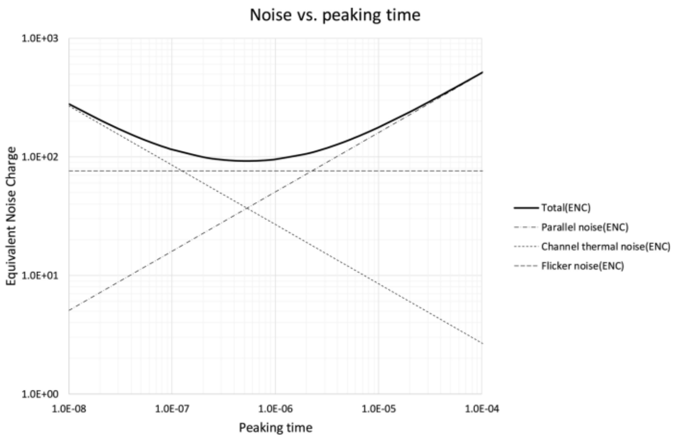
PILE-UP

Digital Signal Processing

Pole-Zero Compensation

Digital Filters





Trapezoidal Filter

Jordanov's Trapezoidal Filter (1994)

- Combines discrete-time Pole-Zero cancellation and band-pass shaping into a single, real-time recursive algorithm
- Provides efficient filtering for white series, white parallel and flicker noise
- Implements a lightweight recursive equation
- Usually adopted as the de-facto baseline for modern digital gamma-ray spectroscopy

N·H ELSEVIER
Nuclear Instruments and Methods in Physics Research A 353 (1994) 261–264
NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

Digital techniques for real-time pulse shaping in radiation measurements

Valentin T. Jordanov [a,*], Glenn F. Knoll [b], Alan C. Huber [a], John A. Pantazis [a]
[a] Amptek Inc., 6 De Angelo Dr., Bedford, MA 01730, USA
[b] The University of Michigan, Department of Nuclear Engineering, Ann Arbor, MI 48109, USA

Abstract

Recursive algorithms for real-time digital pulse shaping in pulse height measurements have been developed. The differentiated signal from the preamplifier (exponential pulse) is amplified and then digitized. Digital data are deconvolved so that the response of the high-pass network is eliminated. The deconvolved pulse is processed by a time-invariant digital filter which allows trapezoidal/triangular or cusp-like shapes to be synthesized. A prototype of a digital trapezoidal processor was built which is capable of sampling and processing digital data in real time at clock rates up to 50 MHz.

1. Introduction

In our previous work [1] we described recursive algorithms for real time pulse processing in high resolution spectroscopy. In that paper we also proposed a hardware configuration for a trapezoidal/triangular pulse shaper. Although we presented some initial results obtained using a quasi-real time system, our further work has now resulted in the assembly and testing of a prototype that operates in true real time.

In the discussion that follows, it is assumed that an exponential pulse is digitized. This signal can be obtained by CR differentiation of the signal from a reset type charge sensitive preamplifier or by differentiation with a pole-zero cancellation network of the signal from a resistive feedback preamplifier. This approach allows elimination of the dc offset of the preamplifier signal and sufficient amplification of the short exponential pulses so that maximum utilization of the resolution of the sampling ADC can be

second algorithm transforms an exponential or step pulse to a symmetrical pulse shape with the leading edge proportional to $t^2 + t$. We call this shape "cusp-like."

2. Trapezoidal digital pulse-shaper

The recursive algorithm [1] that converts a digitized exponential pulse $v(n)$ into a symmetrical trapezoidal pulse $s(n)$ is given as

$$d^{k,l}(n) = v(n) - v(n-k) - v(n-l) + v(n-k-l), \tag{1}$$

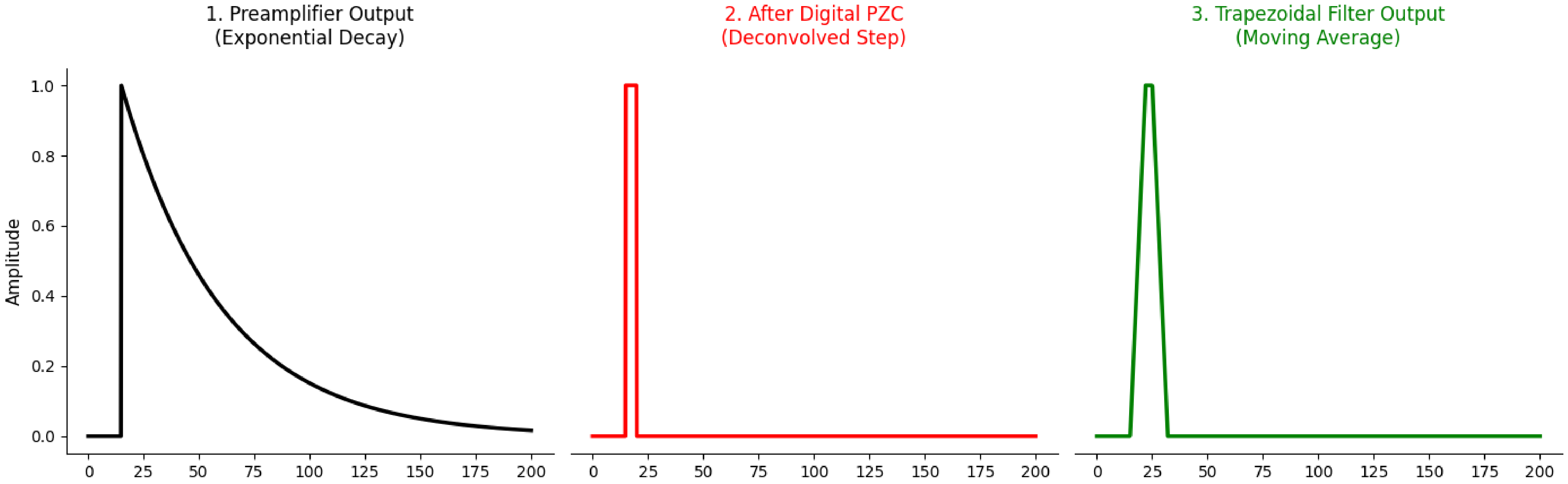
$$p(n) = p(n-1) + d^{k,l}(n), \quad n \geq 0, \tag{2}$$

$$r(n) = p(n) + Md^{k,l}(n), \tag{3}$$

$$s(n) = s(n-1) + r(n), \quad n \geq 0, \tag{4}$$

where $v(n)$, $p(n)$, and $s(n)$ are equal to zero for $n < 0$.

Algorithm Steps



Pole Zero Compensation

$$\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt} = -\tau\,(e^{-\frac{t}{\tau}})_0^T = \tau - \tau e^{-\frac{T}{\tau}}$$

$$(\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt}) + \tau e^{-\frac{T}{\tau}} = \tau - \tau e^{-\frac{T}{\tau}} + \tau e^{-\frac{T}{\tau}} = \tau$$

Riemann Sum

$$\sum_{i=0}^{N} e^{-\frac{i\,\Delta t}{\tau}}\,\Delta t + \tau e^{-\frac{N}{\tau}} = \tau$$

Pole Zero Compensation

$$\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt} = -\tau\,(e^{-\frac{t}{\tau}})_0^T = \tau - \tau e^{-\frac{T}{\tau}}$$

$$(\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt}) + \tau e^{-\frac{T}{\tau}} = \tau - \tau e^{-\frac{T}{\tau}} + \tau e^{-\frac{T}{\tau}} = \tau$$

Riemann Sum

$$\sum_{i=0}^{N} e^{-\frac{i\,\cancel{\Delta t}}{\tau}}\,\cancel{\Delta t} + \tau e^{-\frac{N}{\tau}} = \tau$$

Δ t = clock cycle = 1

Pole Zero Compensation

$$\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt} = -\tau\,(e^{-\frac{t}{\tau}})_0^T = \tau - \tau e^{-\frac{T}{\tau}}$$

$$\left(\int_0^T e^{-\frac{t}{\tau}}\mathrm{dt}\right) + \tau e^{-\frac{T}{\tau}} = \tau - \tau e^{-\frac{T}{\tau}} + \tau e^{-\frac{T}{\tau}} = \tau$$

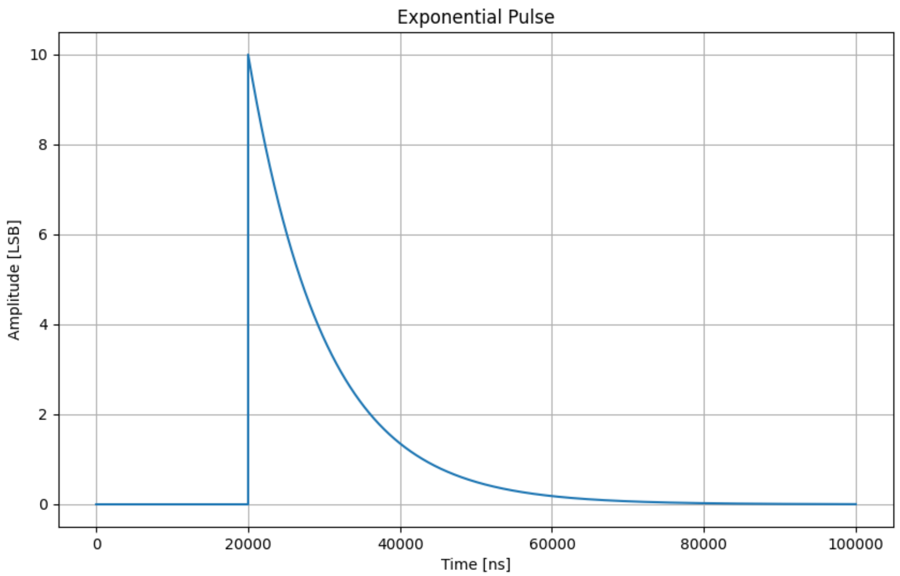
Riemann Sum

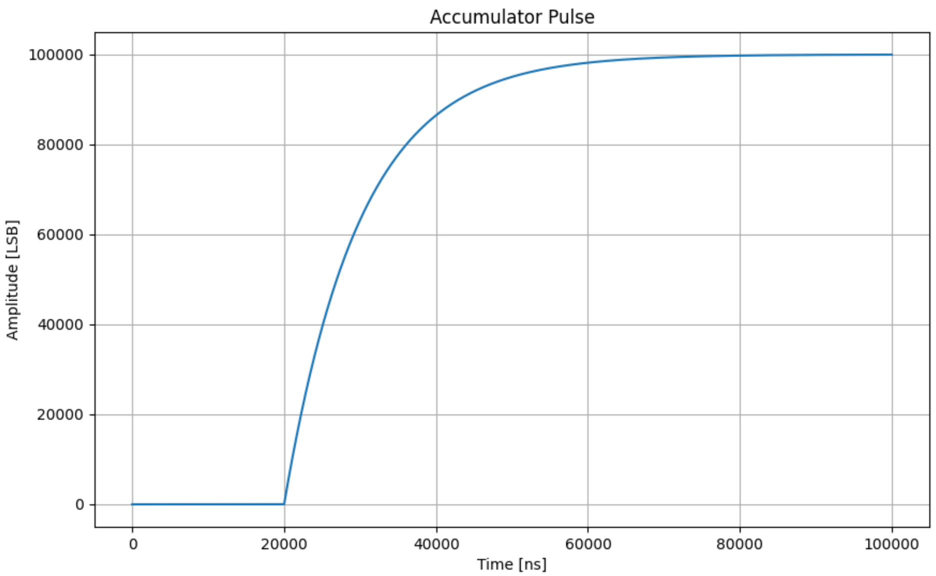
$$\sum_{i=0}^{N} e^{-\frac{i}{\tau}} + \tau e^{-\frac{N}{\tau}} = \tau$$

CAEN Tools for Discovery

Pole Zero Compensation

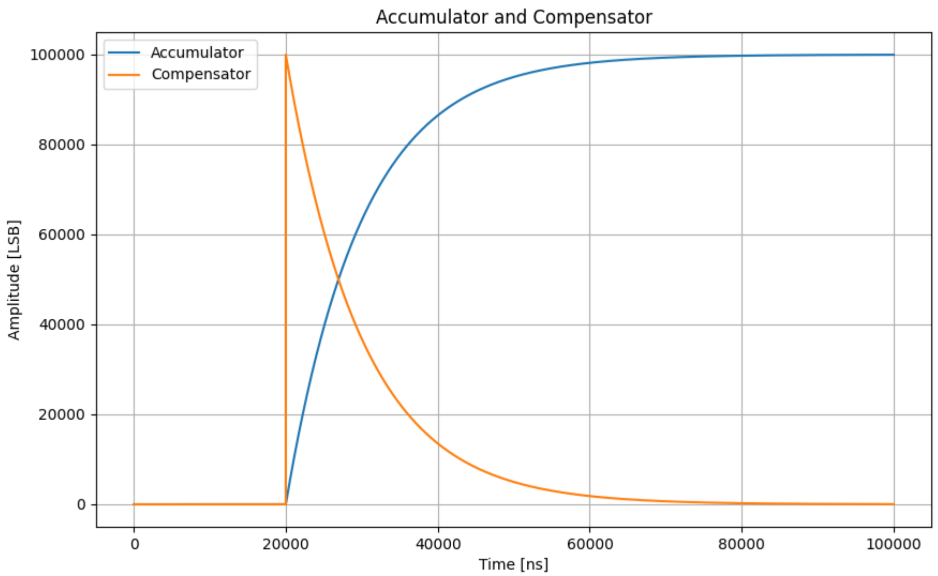
$$\sum_{i=0}^{N} e^{-\frac{i}{\tau}} + \tau e^{-\frac{N}{\tau}} = \tau$$





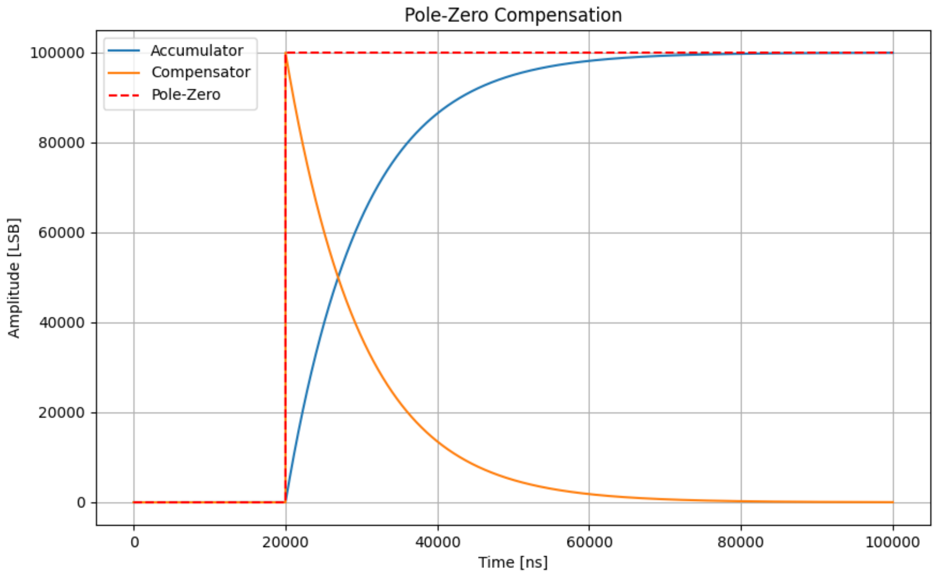
Pole Zero Compensation

$$\sum_{i=0}^{N} e^{-\frac{i}{\tau}} + \tau e^{-\frac{N}{\tau}} = \tau$$



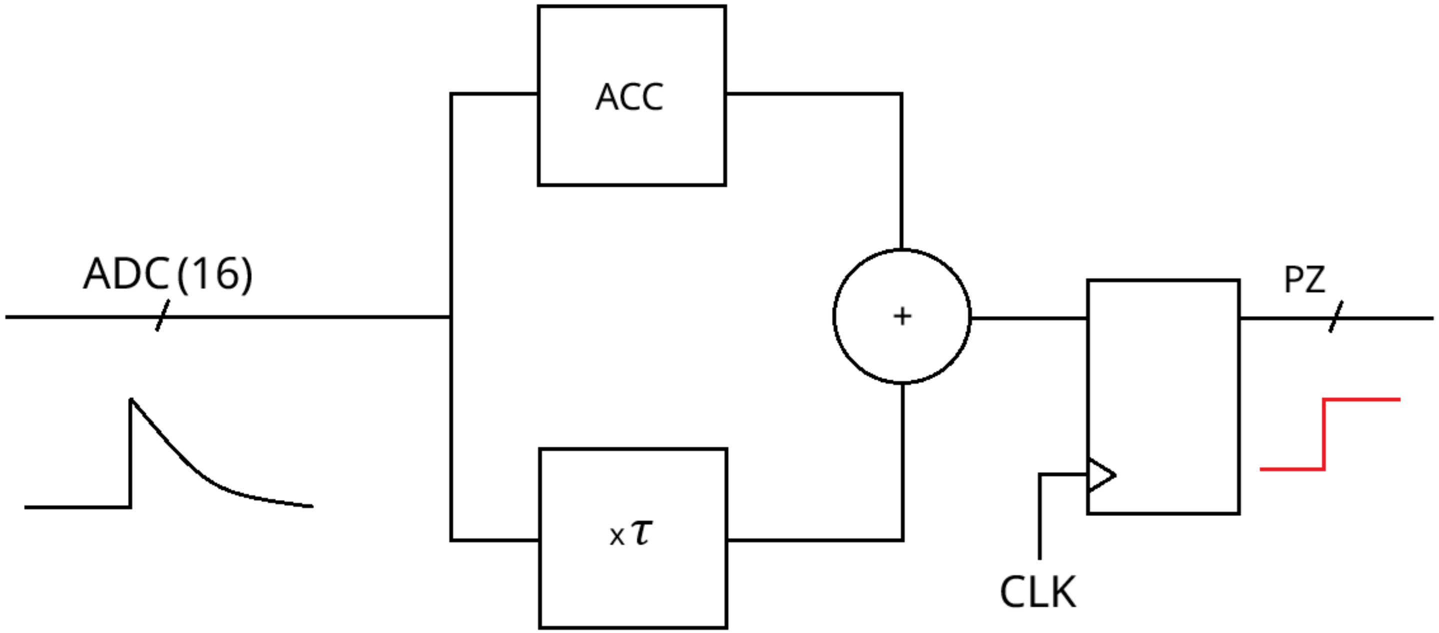
Pole Zero Compensation

$$\sum_{i=0}^{N} e^{-\frac{i}{\tau}} + \tau e^{-\frac{N}{\tau}} = \tau$$



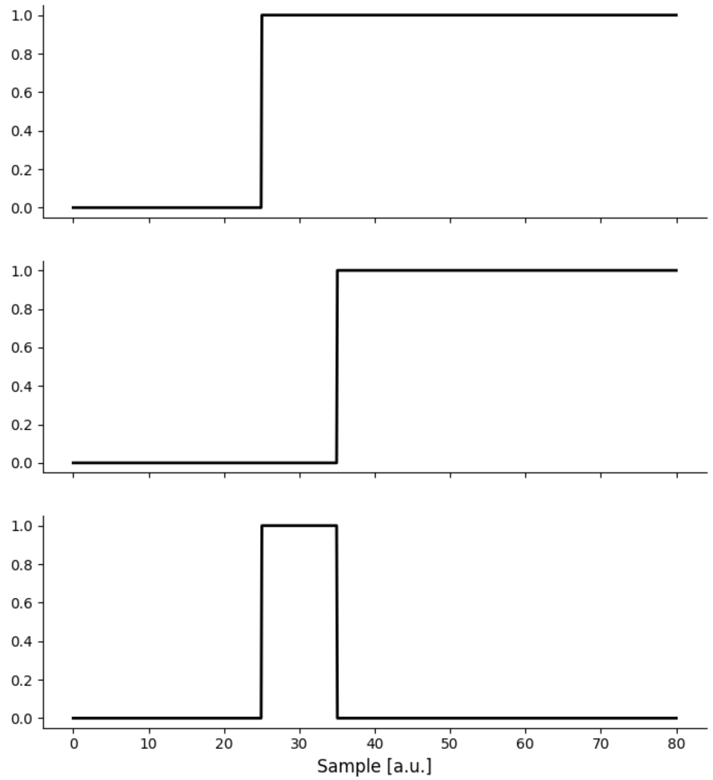
Pole Zero Compensation in FPGA

$$\sum_{i=0}^{N} e^{-\frac{i}{\tau}} + \tau e^{-\frac{N}{\tau}} = \tau$$



First Difference Window

$$d[n] = x[n] - x[n\text{-}r]$$



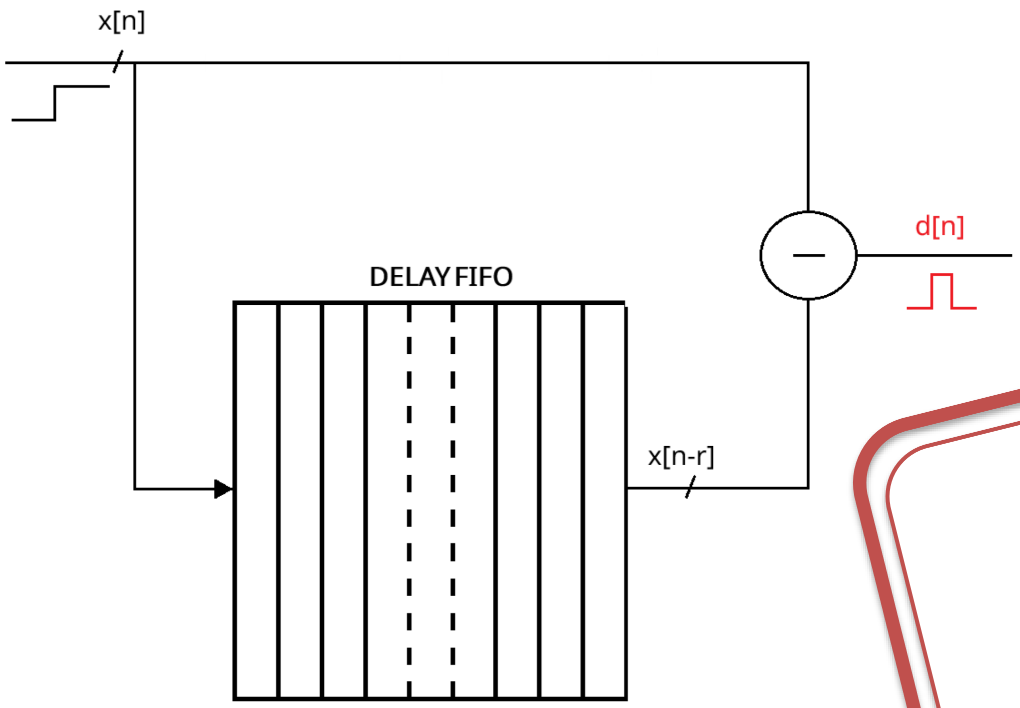
Step Function

Delayed Step Function

Rectangular Wave

First Difference Window

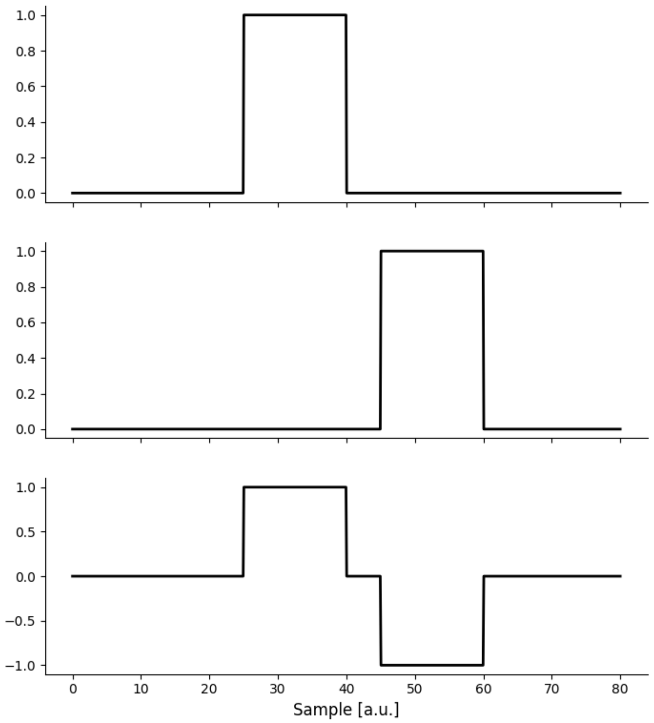
$$d[n] = x[n] - x[n-r]$$



High Pass Filter

Second Difference Window

$$p[n] = d[n] - d[n\text{-}m]$$



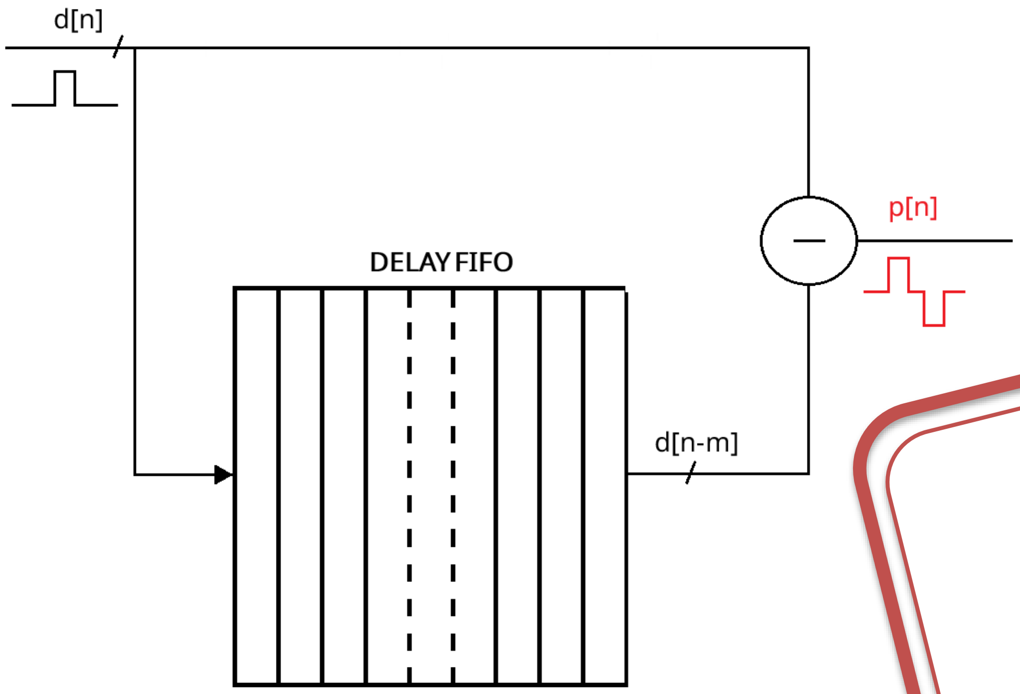
Rectangular Pulse

Delayed Rectangular Pulse

Bipolar Rectangular Pulse

First Difference Window

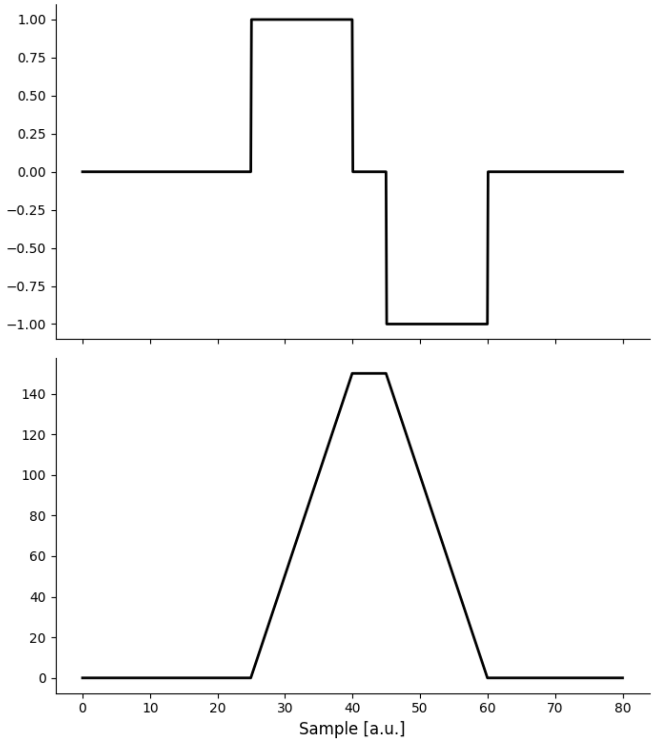
$$p[n] = d[n] - d[n\text{-}m]$$



High Pass Filter

Trapezoidal Accumulator

$$g[n] = g[n-1] - d[n]$$

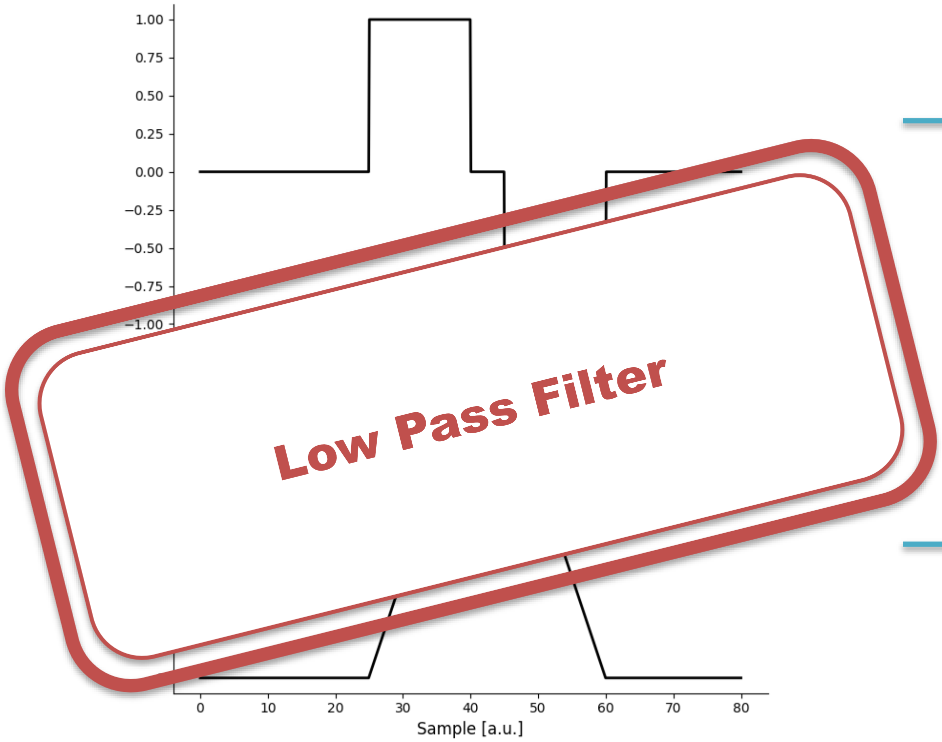


Bipolar Rectangular Pulse

Trapezoid

Trapezoidal Accumulator

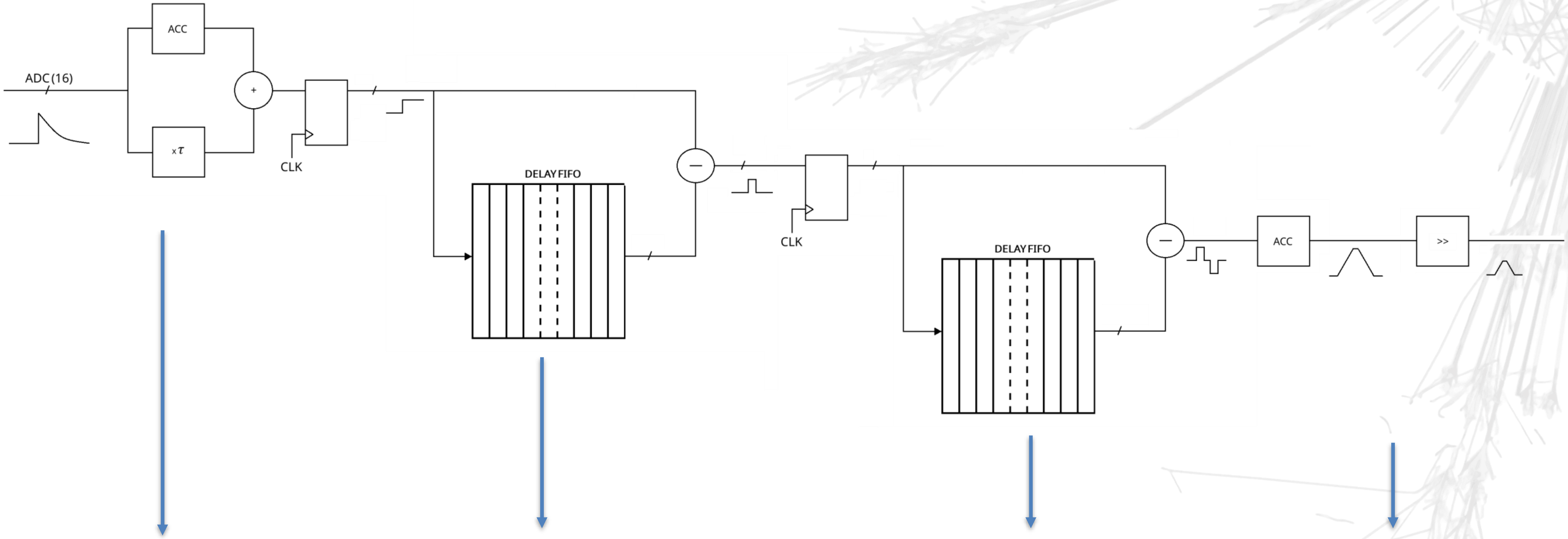
$$g[n] = g[n - 1] - d[n]$$



Bipolar Rectangular Pulse

Trapezoid + Shift

Trapezoid Chain



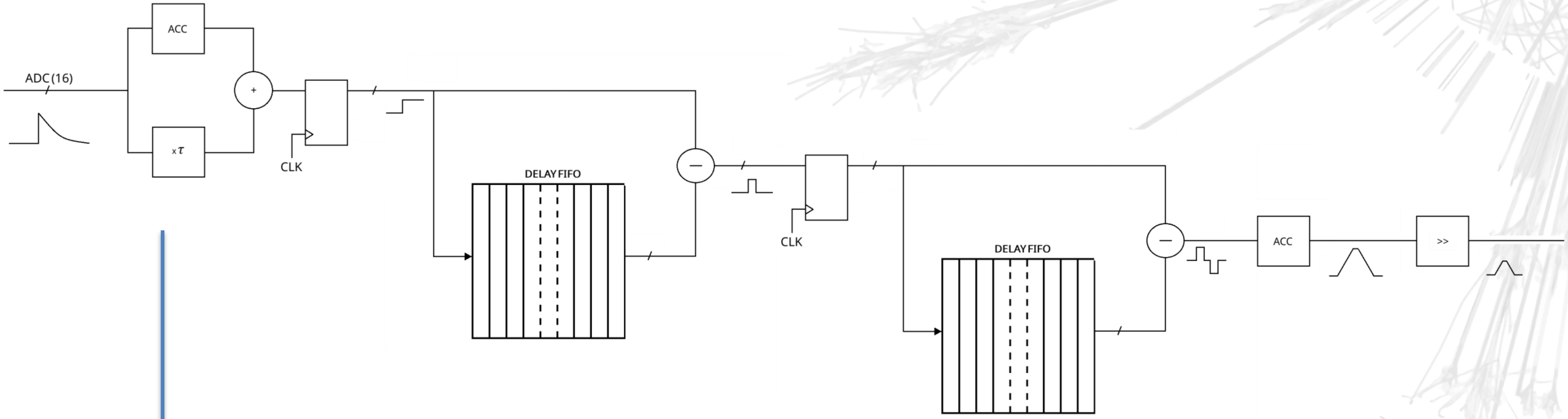
Pole-Zero

HP- Filter

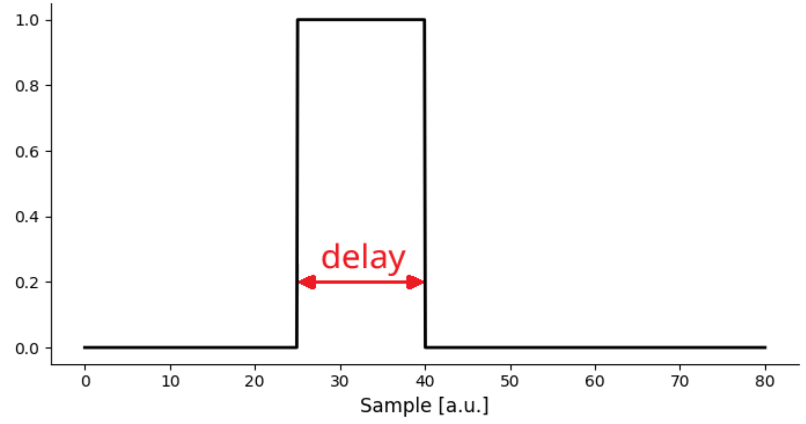
HP- Filter

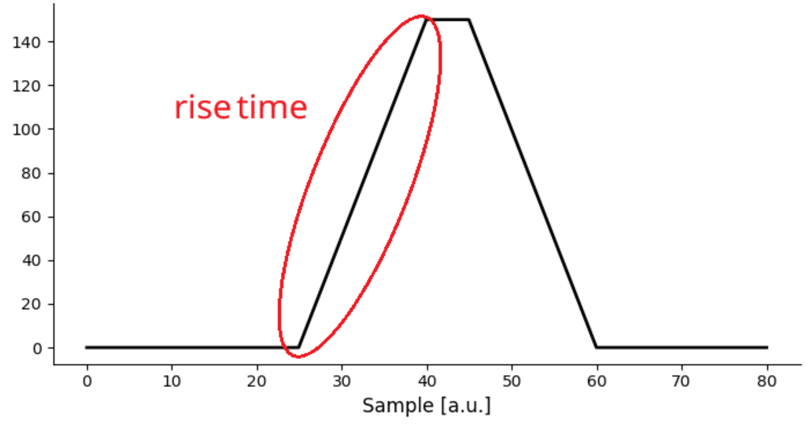
LP- Filter

Trapezoid Chain

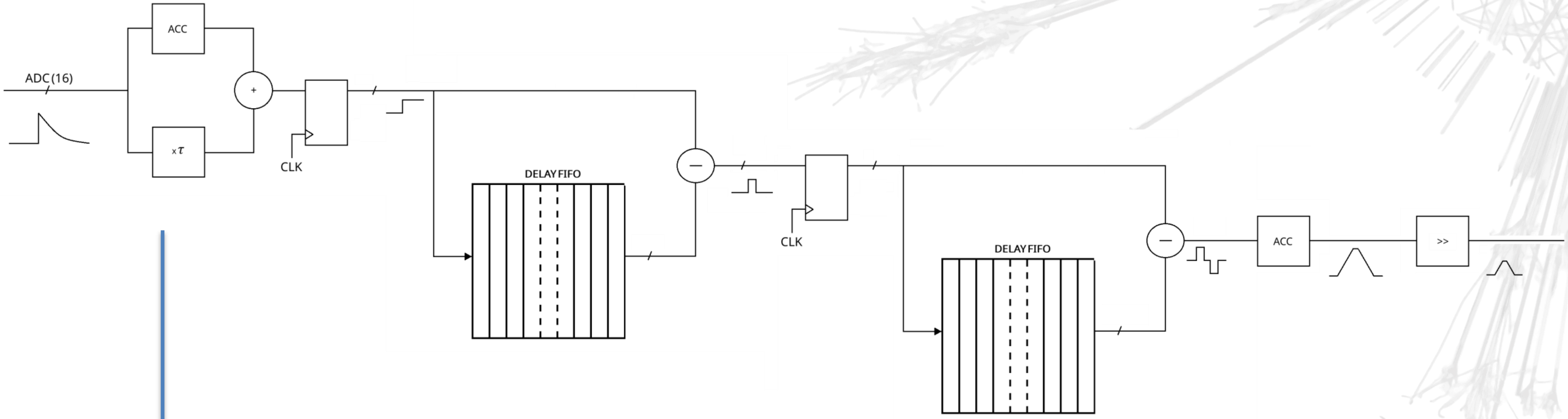


Pole-Zero

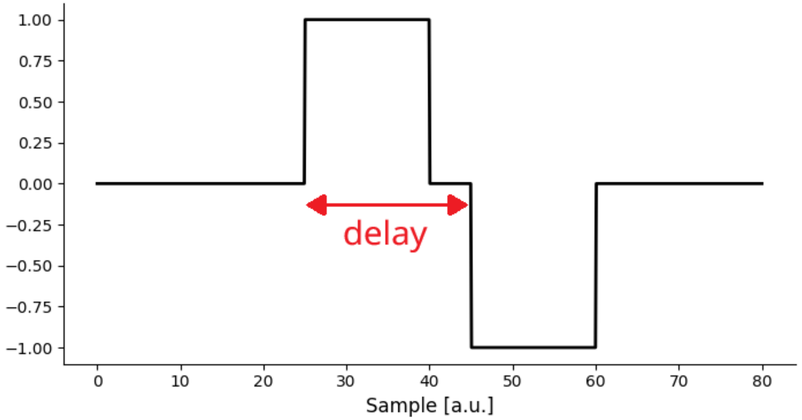


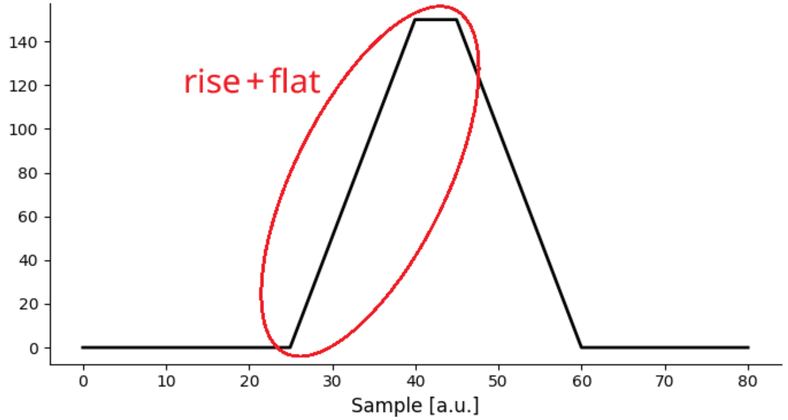


Trapezoid Chain

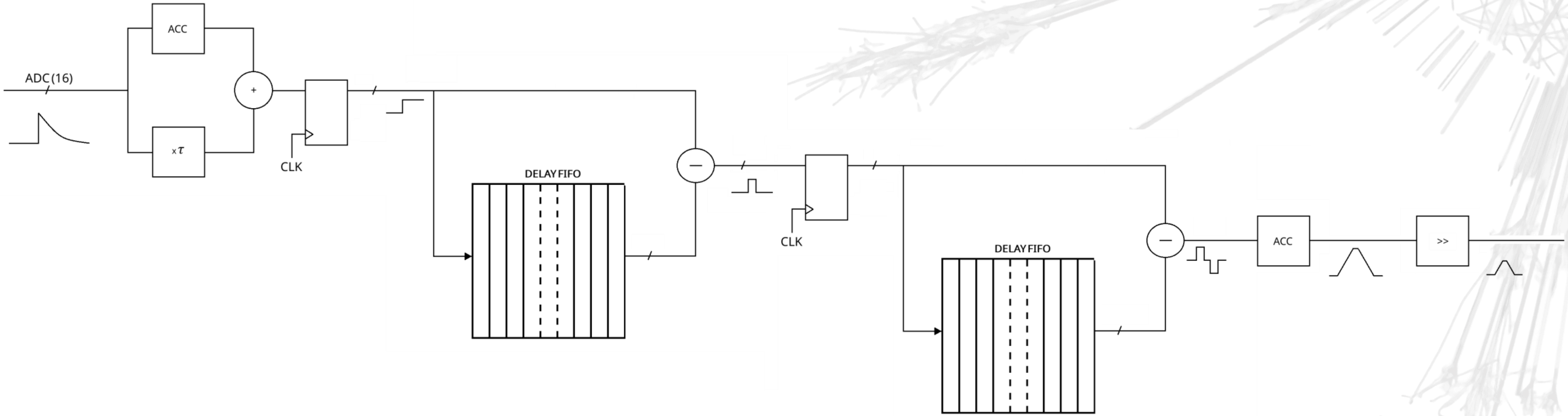


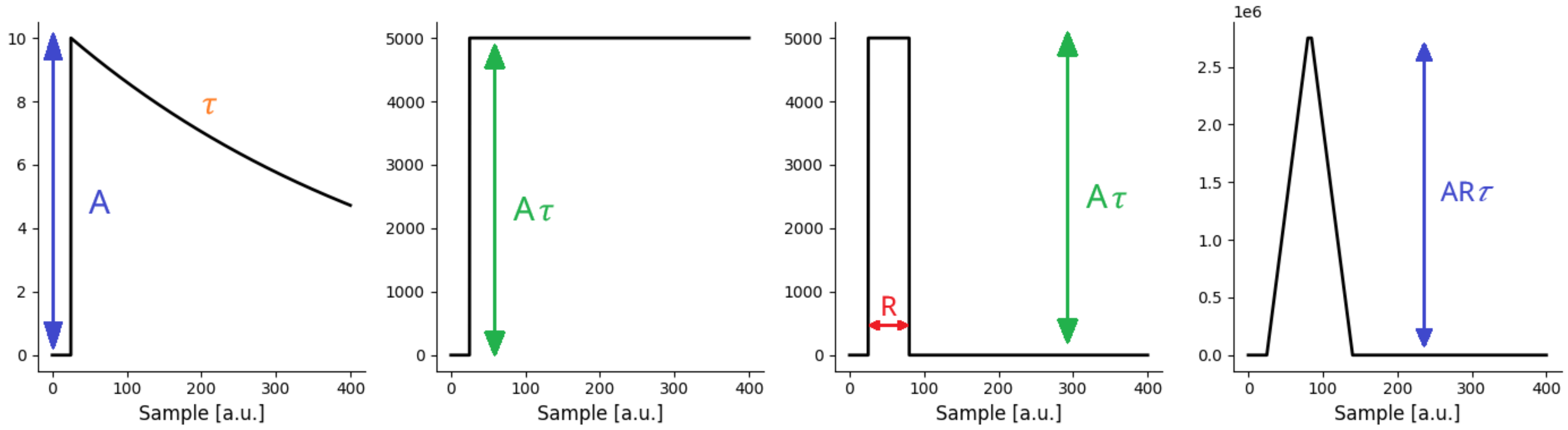
Pole-Zero





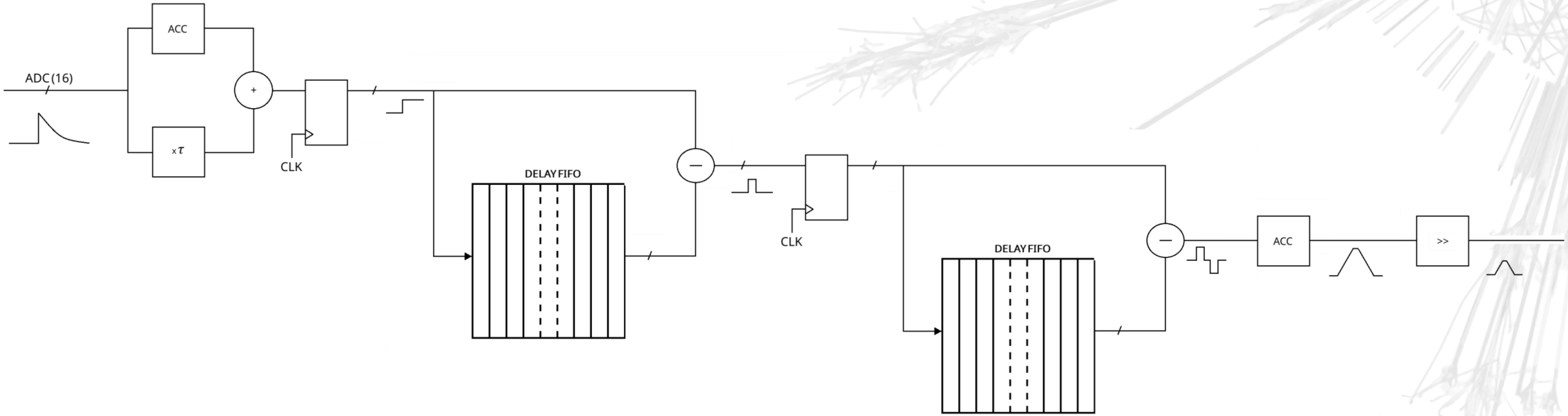
Trapezoid Chain

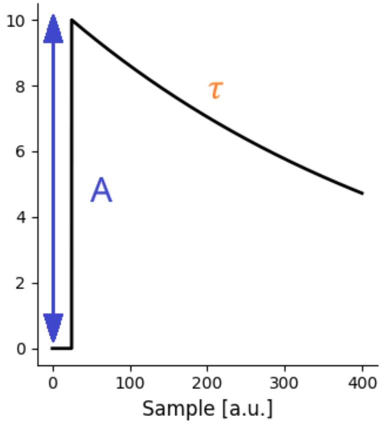


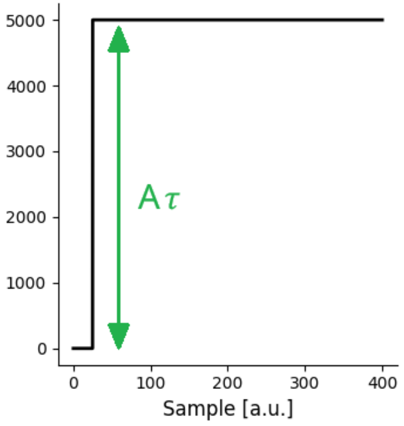


CAEN Tools for Discovery

Chain Optimization





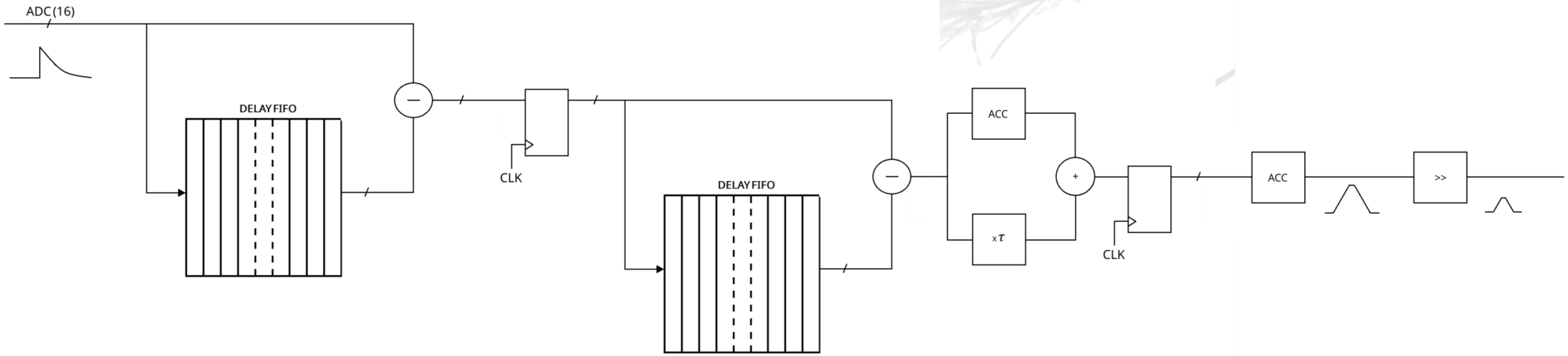


ADC = 16 bit

τ = 13 bit

A x τ = 29 bit

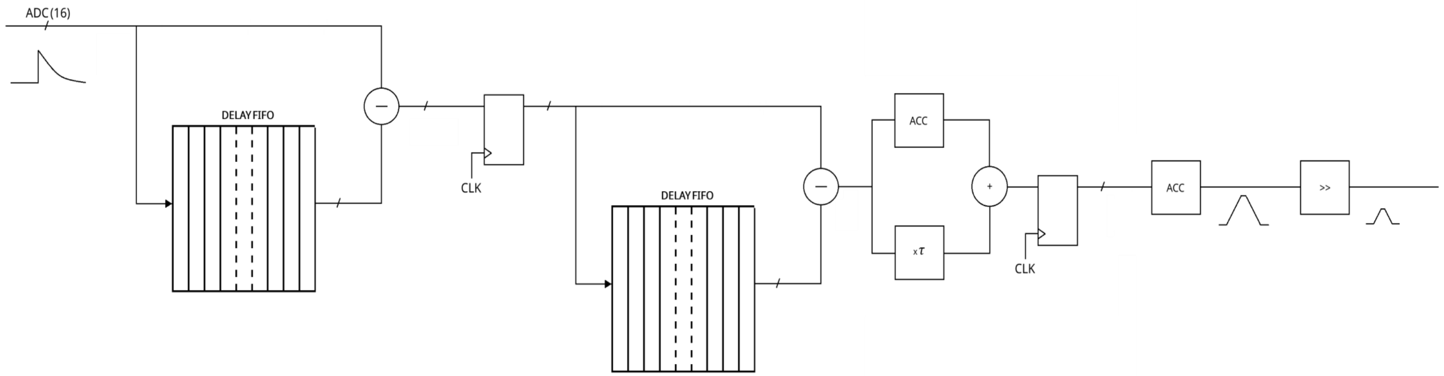
Final Chain



Pole-zero cancellation and double-differencing are both LTI operations :
cascaded LTI systems commute, so their order is interchangeable without affecting the output.

Trapezoid Advantages

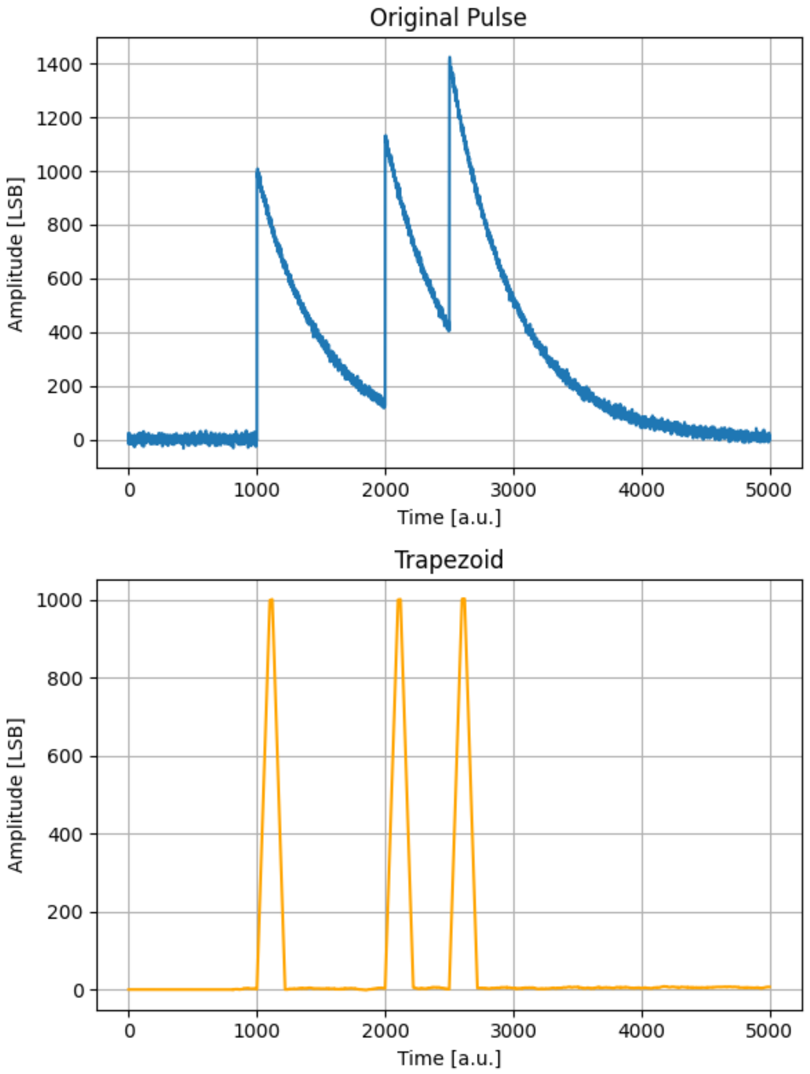
- Real Time Filter



Trapezoid Advantages

- Real Time Filter

- Improved Pile-up management due to the Pole-Zero Compensation

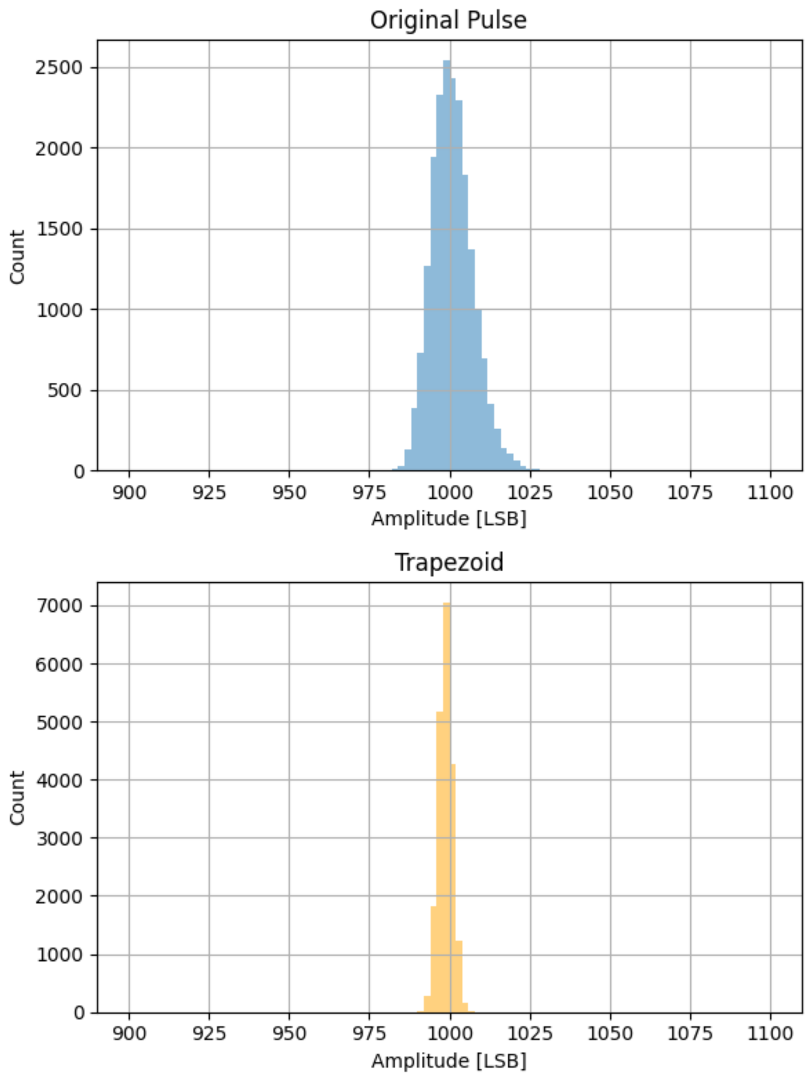


Trapezoid Advantages

- Real Time Filter

- Improved Pile-up management due to the Pole-Zero Compensation

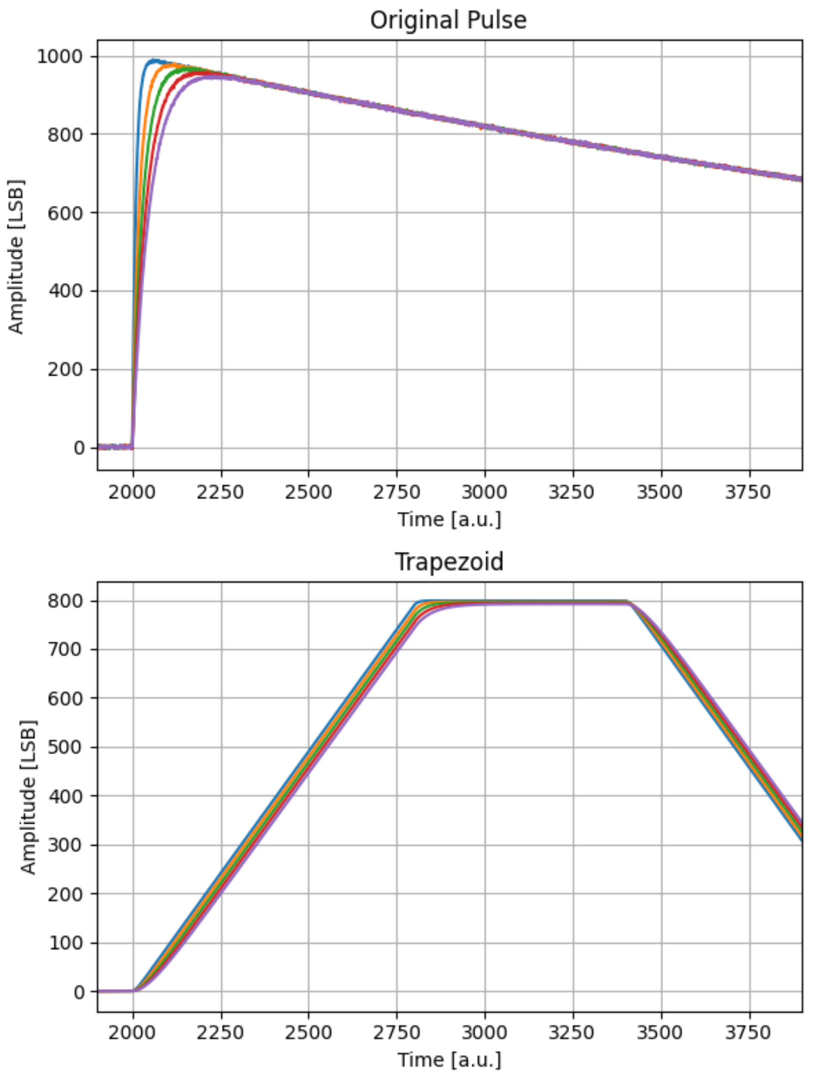
- Better Noise Performance due to the Pass-Band Filter



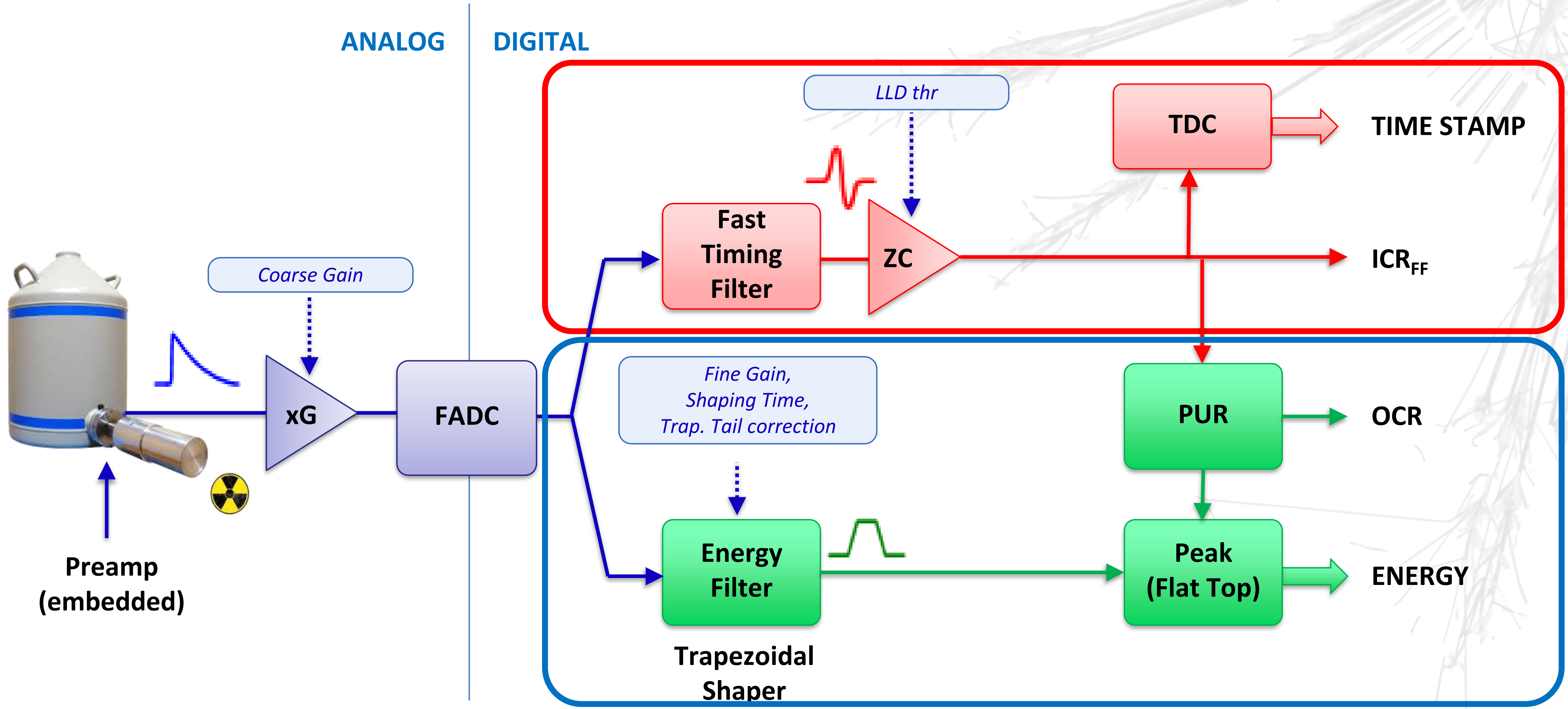
CAEN
Tools for Discovery

Trapezoid Advantages

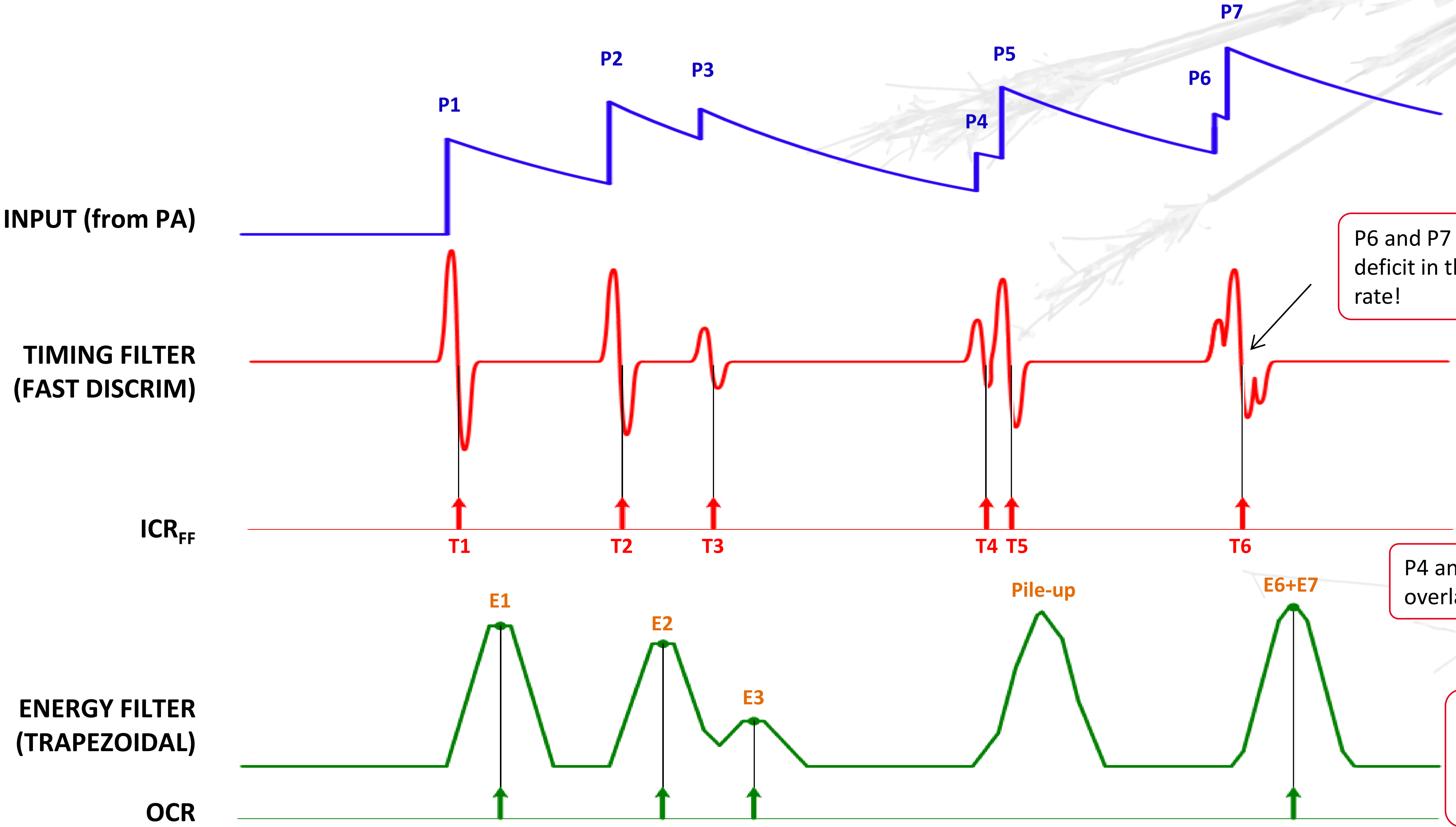
- Real Time Filter
- Improved Pile-up management due to the Pole-Zero Compensation
- Better Noise Performance due to the Pass-Band Filter
- Ballistic Deficit Mitigation due to the Flat-Top



The DPP-PHA Algorithms and Block Diagram



The DPP-PHA Signals



P6 and P7 not resolved: deficit in the input count rate!

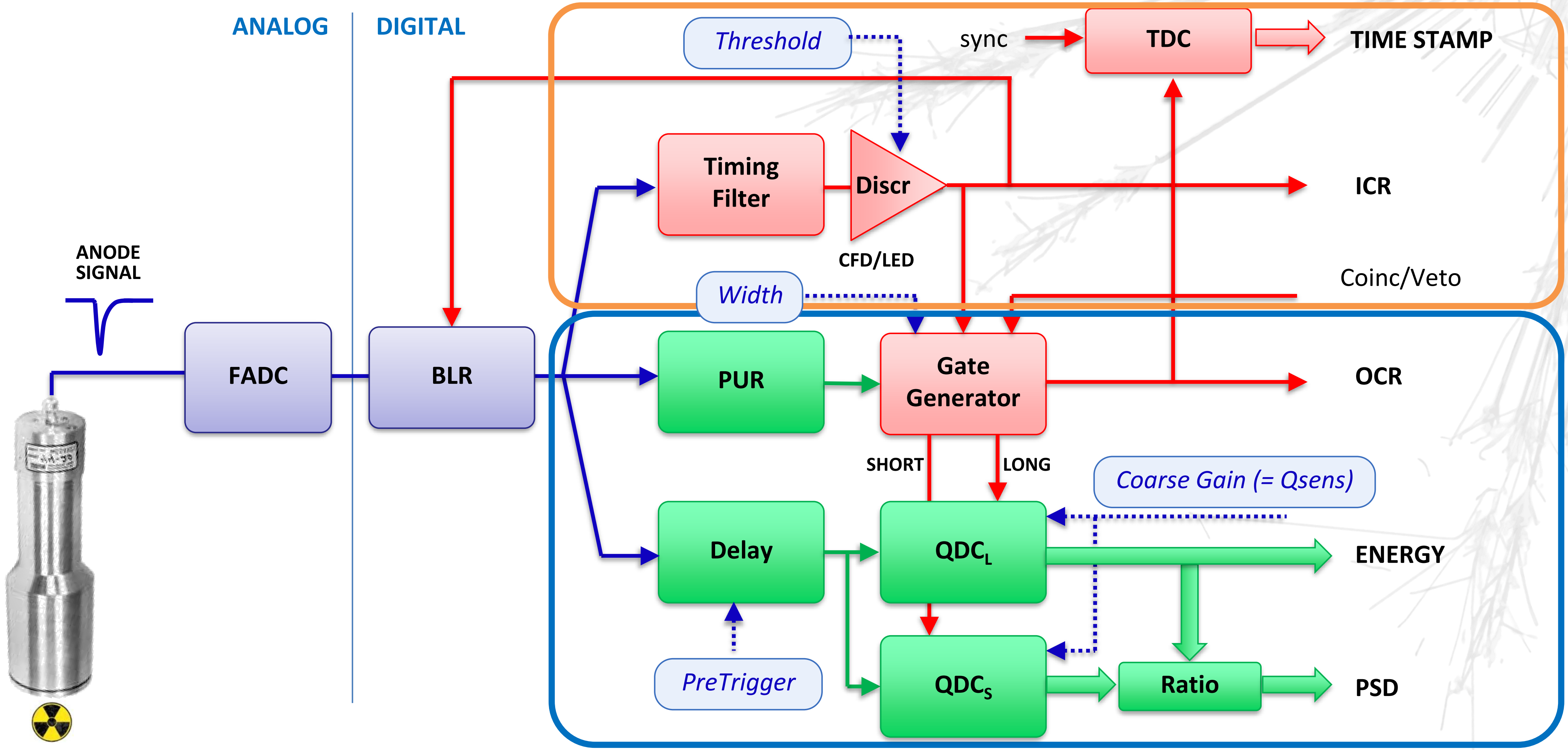
P4 and P5 trapezoids overlap: Pile-up!

P6 and P7 generate only 1 trapezoid: sum peaks!

PSD

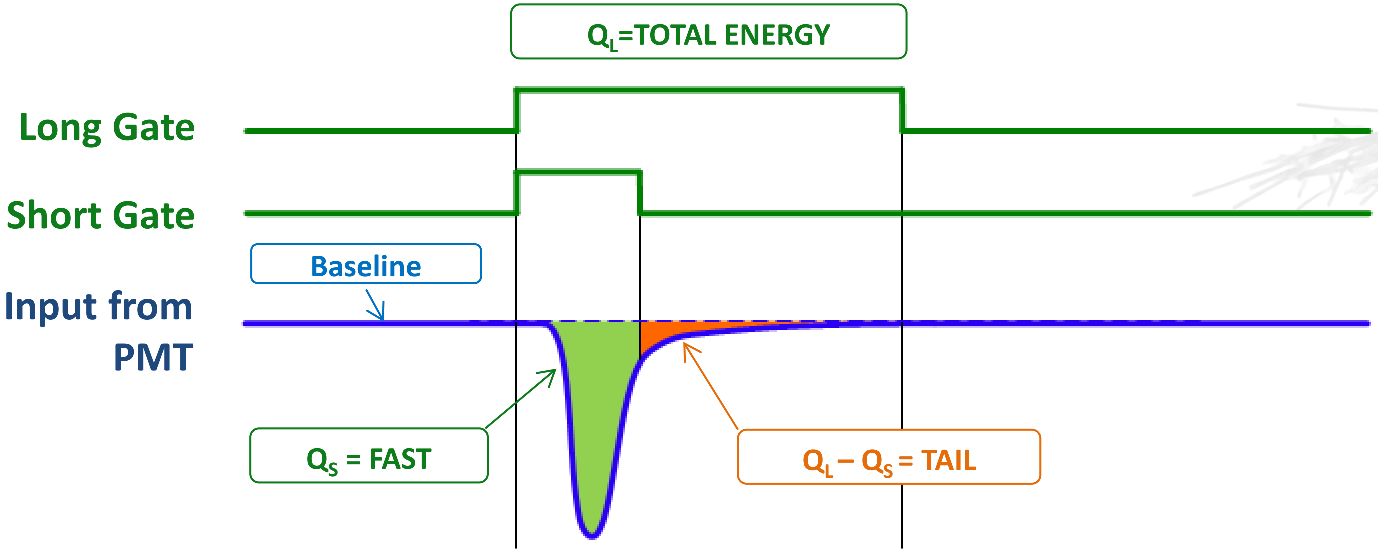
Pulse Shape Discrimination

The DPP-PSD/QDC Algorithm and Block Diagram

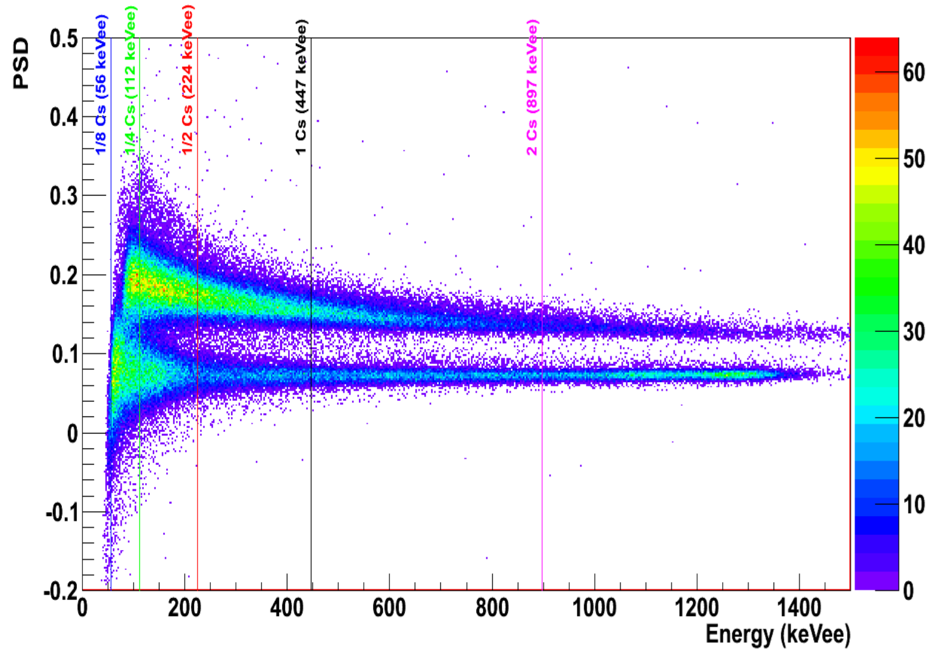


CAEN
Tools for Discovery

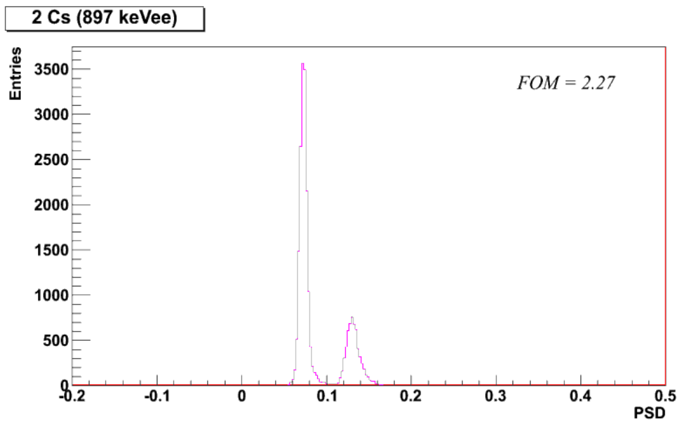
The DPP-PSD/QDC Signals

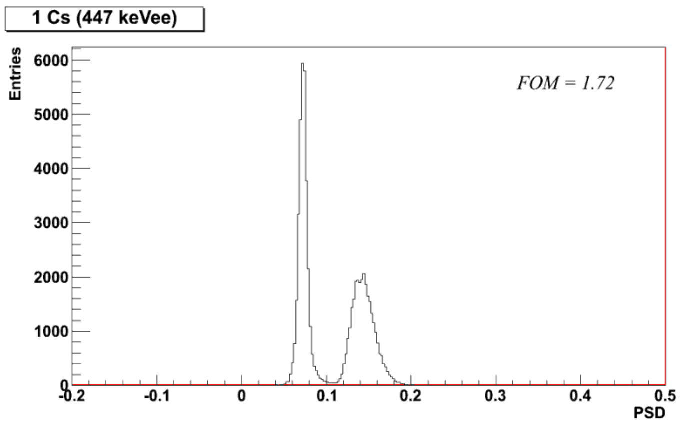


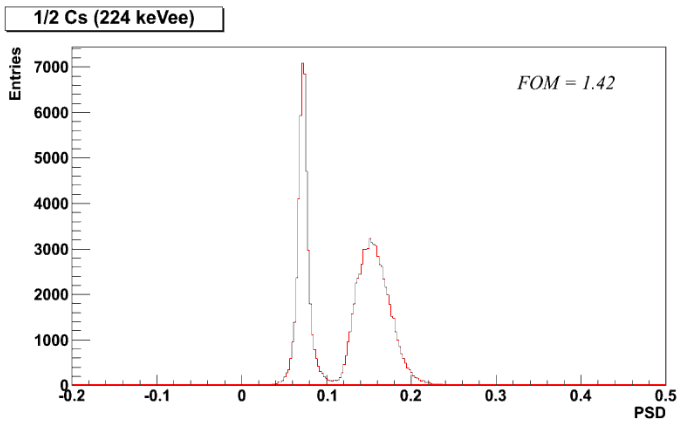
$$PSD = \frac{TAIL}{TOTAL}$$

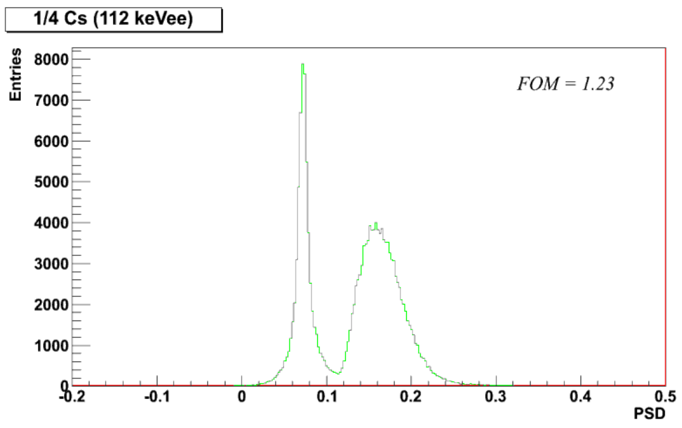


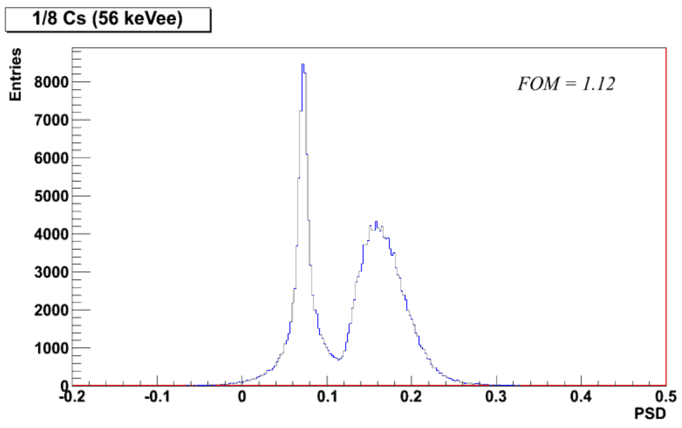
The DPP-PSD/QDC Results











$$FOM = \frac{\Delta PEAK}{FWHM_{\gamma} + FWHM_{n}}$$

CAEN
Tools for Discovery

Thank you for your attention!

Alberto Potenza

a.potenza@caen.it

Reach us at educational@caen.it
Visit www.edu.caen.it
www.caen.it

Backup

Why Real Time Digital Filters

DAQ → Waveform → PC → Offline Analysis

Data Throughput Continuous Acquisition (e.g.)

- ADC = 16 bit
- Sampling Rate = 100 Msps
- N. Channels = 64

Data Throughput Continuous Acquisition
2[B] * 100 [Msps] * 64 [ch] =
= 12 [GB/s]

Data Throughput Triggerd Acquisition (e.g.)

- ADC = 16 bit
- Sampling Rate = 100 Msps
- Trigger Rate = 5 kevents/s
- Waveform Stored = 100 µs = 10000 samples
- N. Channels = 64

Data Throughput Triggered Acquisition
2[B] * 5000 [events/s] * 10000 [sample/event] * 64 [ch] =
= 6.4 [GB/s]

CAEN
Tools for Discovery

Why Real Time Digital Filters

DAQ → Online Analysis → PC

Data Throughput Online Analysis (e.g.)

- ADC = 16 bit
- Sampling Rate = 100 Msps
- N. Channels = 64
- Trigger Rate = 5 kevents/s
- Timestamp Word = 6B
- Energy Word = 2B
- Extra Word = 4B

Data Throughput Continuous Acquisition

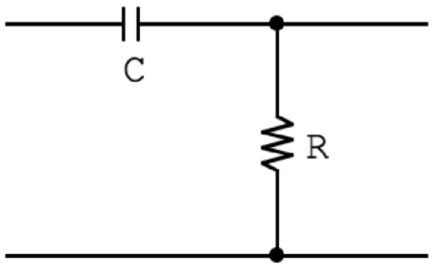
(6[B] + 2[B] + 4[B])* 5000 [events/s] * 64 [ch] =

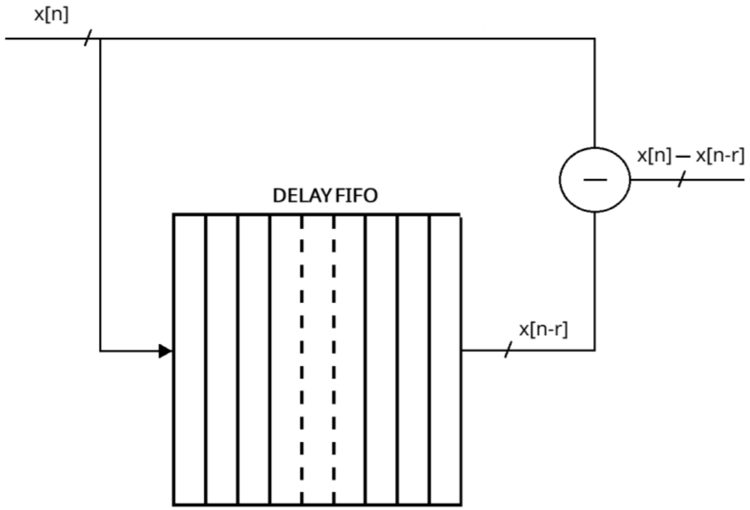
= 3.84 [MB/s]

Huge Data Reduction!!

CAEN
Tools for Discovery

Simple High Pass Filter



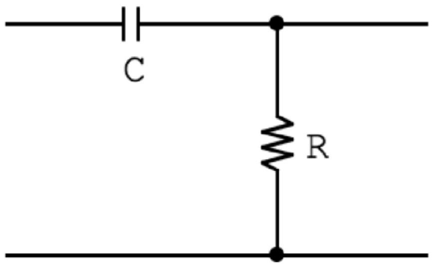


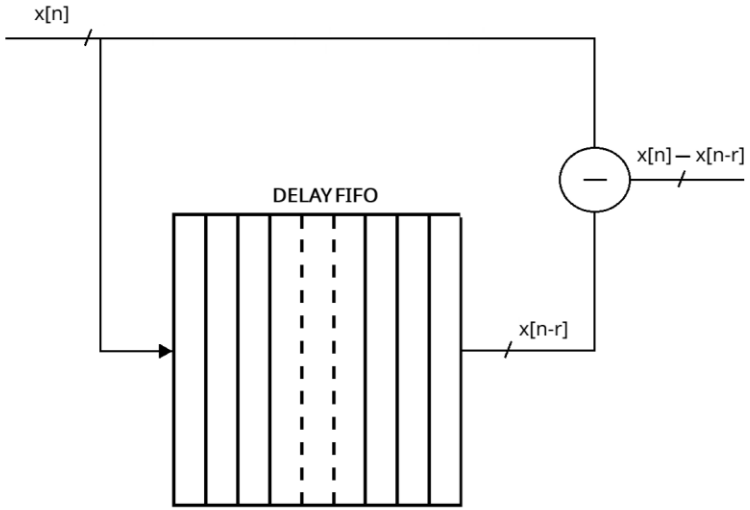
x[n]   = low frequency component [n]   + high frequency component [n]
x[n - r] = low frequency component [n - r] + high frequency component [n - r]

low frequency component [n] ≈ low frequency component [n - r]

x[n] - x[n - r] = high frequency component [n] - high frequency component [n - r]

Simple High Pass Filter





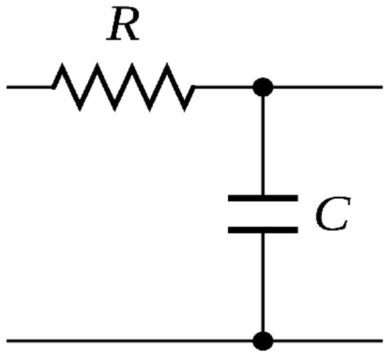
x[n]   = low frequency component [n]   + high frequency component [n]
x[n - r] = low frequency component [n - r] + high frequency component [n - r]

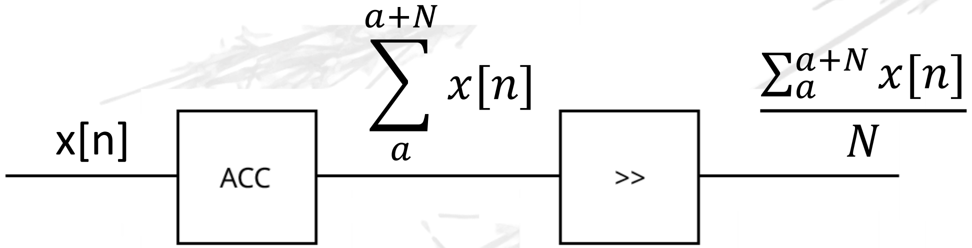
low frequency component [n] ≈ low frequency component [n - r]

x[n] - x[n - r] = high fre

Low frequency component does not change between sample n and sample n – r   !!!!!

Simple Low Pass Filter





$x[n] \rightarrow \sigma_{x[n]}$ → $\sum_{a}^{a+N} x[n] \rightarrow \sigma_{x[n]}\sqrt{N}$ → $\frac{\sum_{a}^{a+N} x[n]}{N} \rightarrow \frac{\sigma_{x[n]}}{\sqrt{N}}$

High Frequency Noise Reduction $\frac{1}{\sqrt{N}}$