Precise Pulse Timing based on Ultra-Fast Waveform Digitizers

Eric Delagnes

Many thanks to :
D. Breton, H. Frisch, H. Grabas, J.F. Genat, J. Maalmi, S. Ritt, G. Varner, J. Va'vra …

Lecture given at the IEEE NSS Symposium, Valencia (Spain) 2011

eric.delagnes@cea.fr

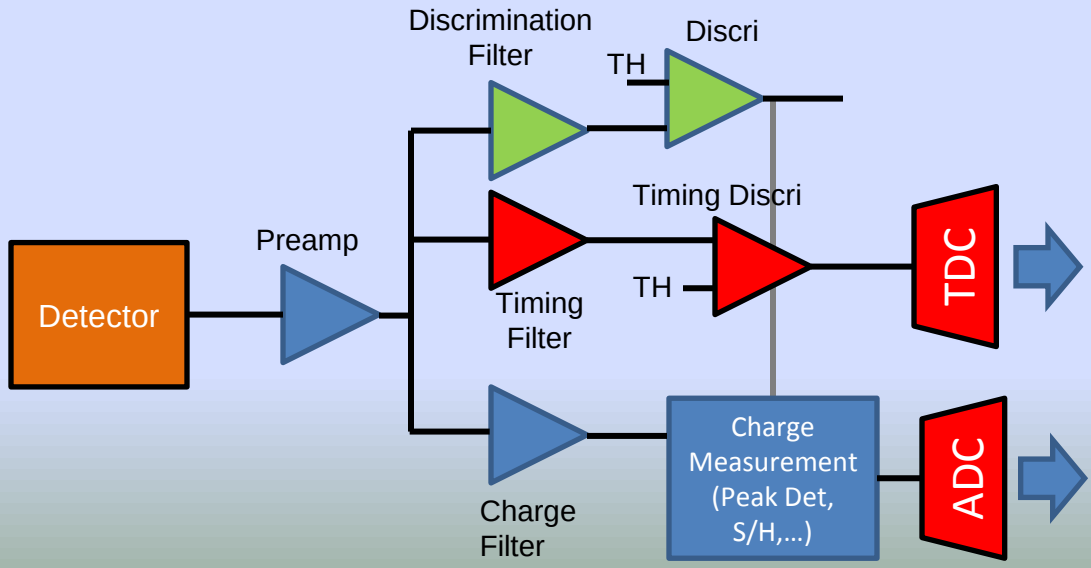
Outline

- Introduction: waveform sampling for time picking.
- Digitizers: State of the art.
- Digitizer parameters.
- Ultra Fast SCA designs for timing.
- Digital Time picking algorithms.

Introduction:
waveform sampling for time picking

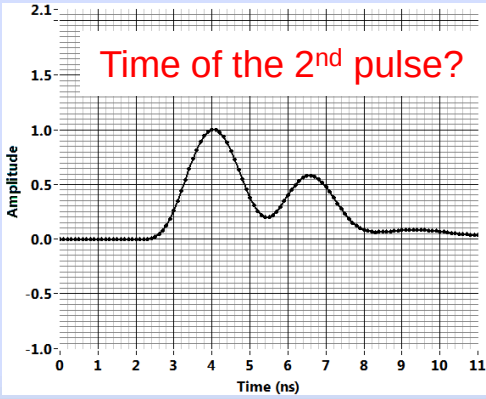
Why use waveform sampling for time picking ?

- Standard “analogue” timing systems for particle systems are:
  - Using separate chains for charge, timing (and discrimination)
  - Using discriminators + TDC (or TAC +ADC).
  - small DAQ effort required (low data throughput).
- They are very efficient, but:
  - Often very specialized for one use.
  - They use a « à priori » signal processing (fraction of CFD, delay of ZC-CFD)
  - Limited by one of their main components (TDC or discriminator).
  - Can be integrated in ASICS, but it is difficult to merge low threshold FE electronics and precise TDCs.
  - High timing resolution discriminators are difficult to design.
  - For very high performances (<20ps FWHM resolution): power hungry and expensive..



Limitations of “standard” timing chains

- Decision taken very early in the processing chain
  - Only few post processing possible (use of TOT or Q/A for time walk correction)
  - No possibility to remove coherent or predictable non stationary/pickup noise
- Chain designed once for all:
  - No possibility to change the chain
  - Sometimes even difficult to change its key parameters (delay of ZC-CFD..)
- Each block add its noise/jitter:
  - Discriminator (noise, residual time walk & non-linearities)
  - TDC noise, jitter.
  - Absolute limit of the TDC quantization step (LSB/ √12) => no possible interpolation
- Wrong timing for pile-up events
- Optimum system tuning depends on the signal shape & on the noise (can change with HV, type of particle,…)
- Detectors with 2 kinds of signal (phoswich [Semmaoui] , ….)



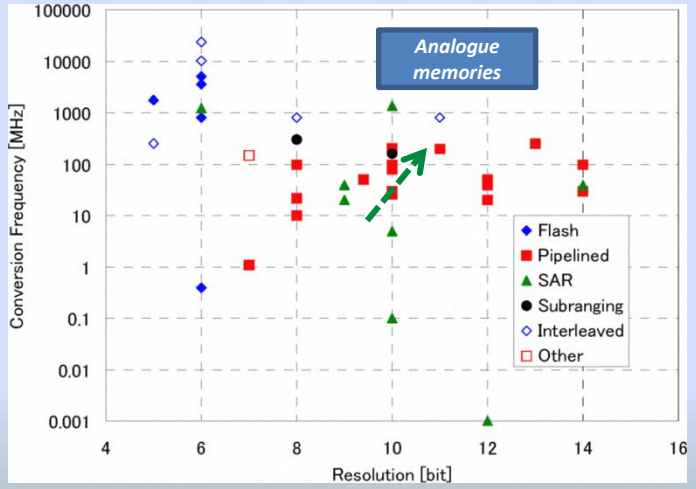
When can waveform sampling be useful ?

- Difficult environment :
  - Pile up
  - Coherent noise or “predictable” noise which can be digitally subtracted before processing => strong Electromagnetic Interference, (example of initial confinement fusion using laser experiment).
- When versatility is required:
  - Pulse shapes unknown before experiment or changing with varying parameters
  - CFD parameters difficult to tune.
  - Various class of event/pulse shape in the same experiment.
- Very high precision timing.
  - < 20ps rms resolution requires expensive analogue electronics.
  - Quite easy with fast waveform sampling.
- When digitized data are already required:
  - For pulse detection / Triggering
  - other pulse parameters are required (charge, pulse shape…):

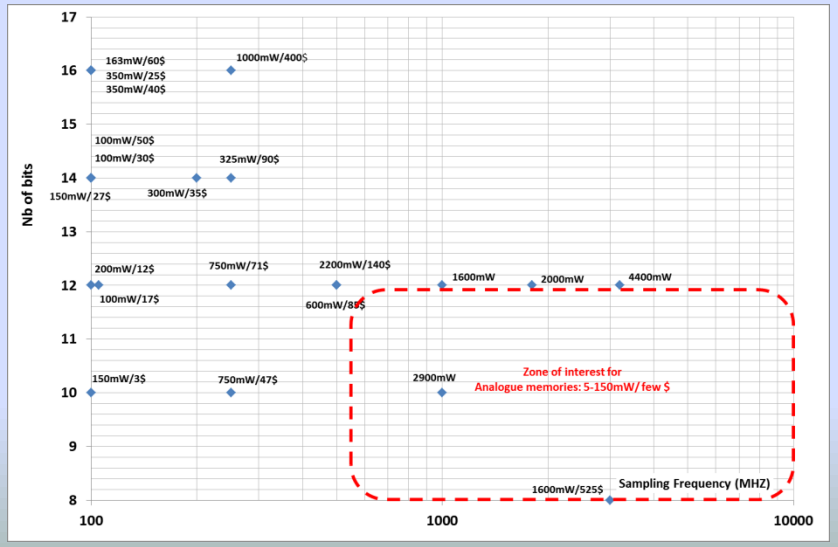
Fast Digitizers: State of the art

Digitizers: State of the art

- Huge progresses on high speed ADC during the last decades:
  - First due to BiCMOS technology.
  - Then to technology scaling in pure CMOS:
    - ⇒ Decrease of capacitances => higher speed, higher bandwidth, lower power consumption.
    - ⇒ reduced vdd => use of simpler architecture.
    - ⇒ Size reduction of digital cells => Rise of algorithmic structures, Generalization of on-chip digital corrections.
- Generalization of full differential structures and use of high speed serial output link:
  - ⇒ Make the integration of ADC easier in a system.
- Commercial availability of ultra-high speed ADCs (>500MSPS, >8 bits).
  - => But expensive (1-10kE/ channel) + need for very high-end FPGAs
- Development of the 2nd generation of ultra-high speed analogue memories in Physics Labs.

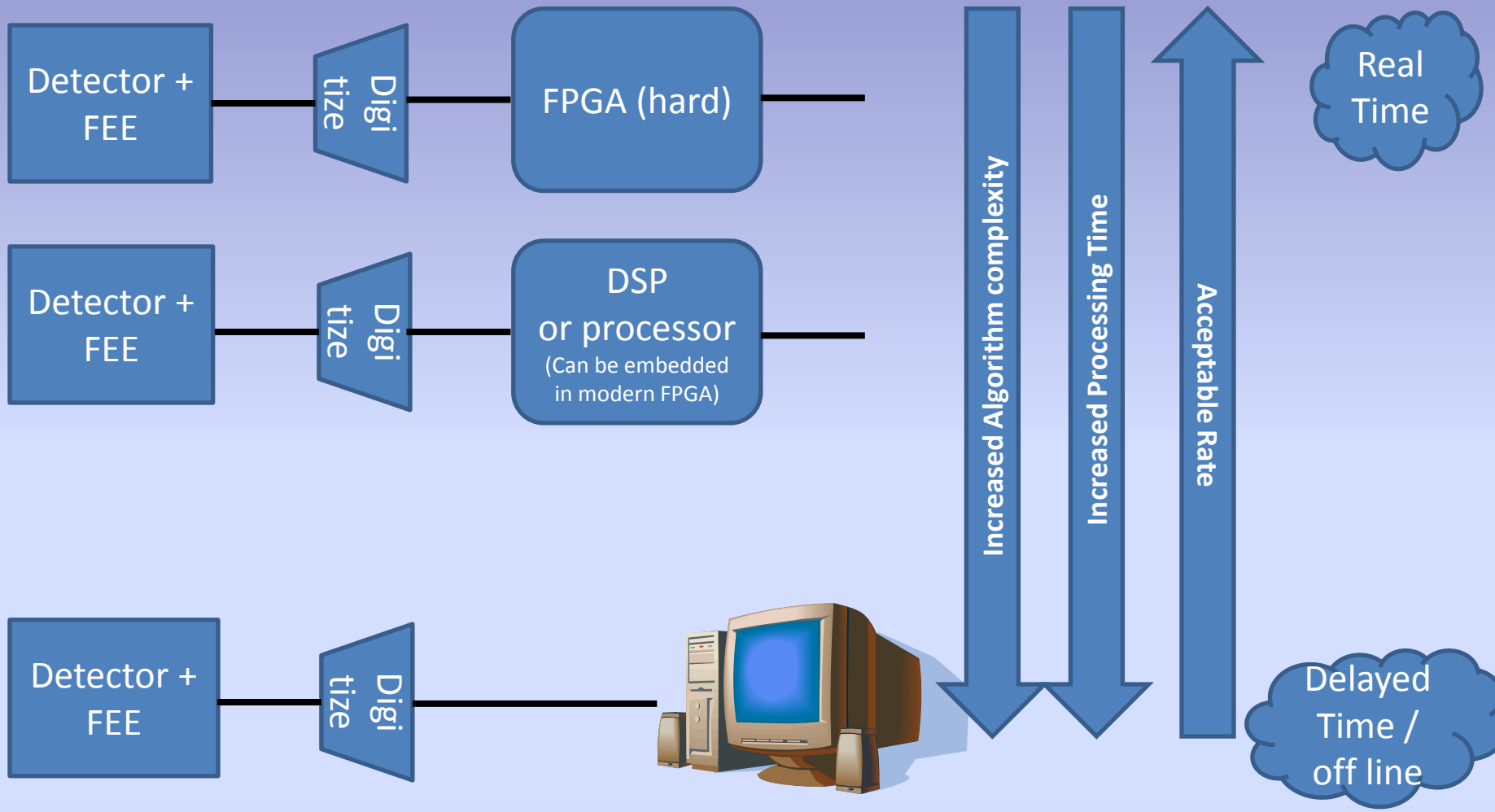


Modified from A. Matsuzawa, "High speed and low power ADC design with dynamic analog circuits", IEEE ASICON 2009, Changsha, China. (R&D products)



Commercial products (2011), Survey of the products from 5 providers.

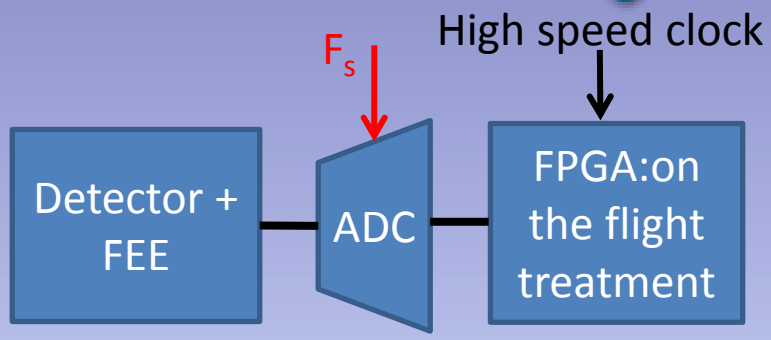
What kind of digital treatment?



Two possible philosophies

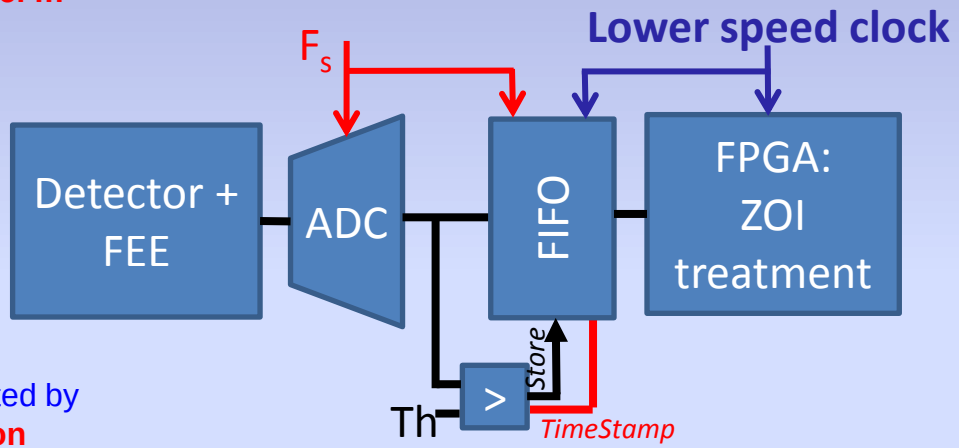
Continuous on the flight data treatment:

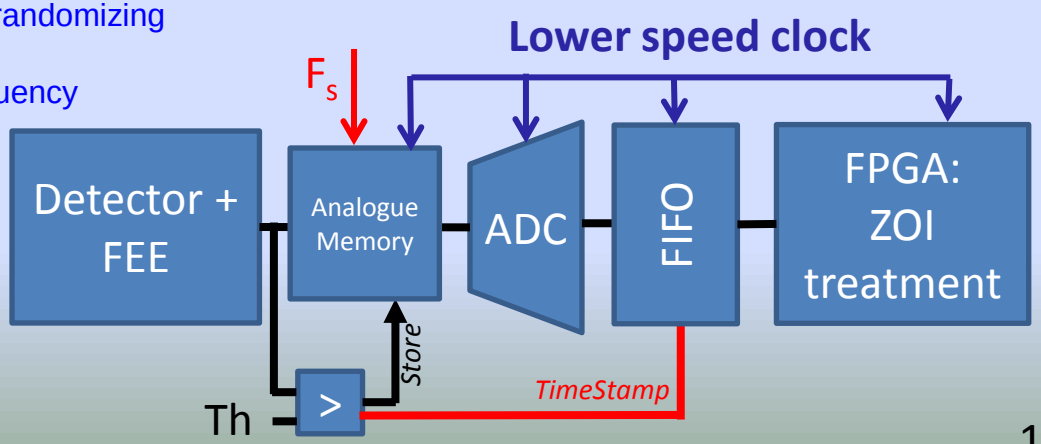
- All the digital data enters in the timing electronics at the sampling rate.
- The result is obtained after a fixed latency
- Only compatible with continuously sampling ADC
- **For a 12bit / 3GSPS => 36 Gbit/s stream to treat/channel !!!**



Zone of interest treatment:

- Pulses are first discriminated and time stamped.
- Compatible with ADC or with analogue memories
- may require an analogue discriminator
- Only the data within a zone of interest (= 1 event) are treated by the digital timing electronics **=> strong data flow reduction**
- Possibility to use intermediate digital FIFOs as derandomizing buffers.
- For the same operations requires lower clock frequency
- The timing electronics may have non fixed latency

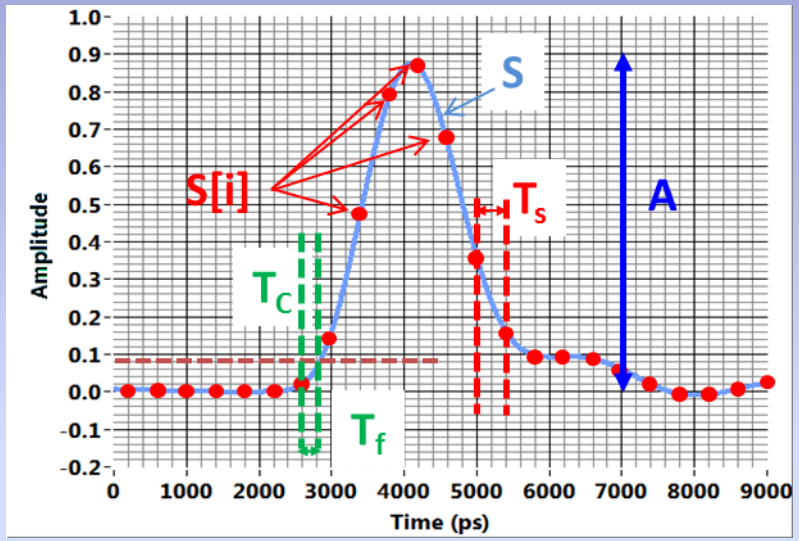




Digitizer parameters

Notations used in the next slides

- The input Signal S with an amplitude A is sampled at F_s frequency (T_s period).

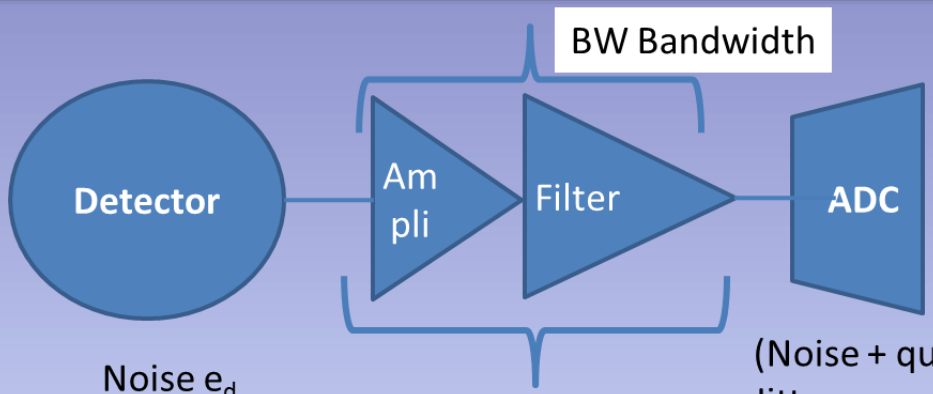


- The samples are $S_i = S[i]$

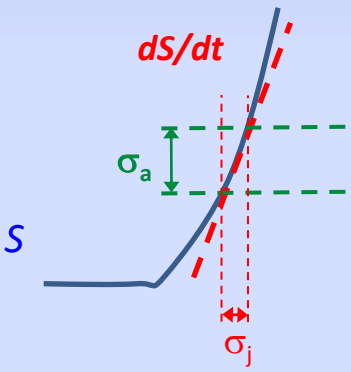
- A coarse time T_c with one sampling clock period quantization step is eventually determined, the residual fine time to find is T_f.

- For some of the methods described further, a normalized reference pulse Ref (continuous in time) is determined by calculation or averaging of measurements. Its sampled version is $Ref_i = Ref[i] = Ref(i.T_s + T_0)$

Limiting factors for the timing precision of **ONE** sample



$$\sigma_T^2 = \sigma_{TTS}^2 + \sigma_j^2 + (v_{fd}^2+v_{fA}^2+v_{ADC}^2) / [dS/dt]^2$$



Where v_{fA} and v_{fd} are the detector + amp noise **filtered.**

$dS/dt = K_1 . A/BW \quad K_1 \sim 0.35$

$v_{fd}^2+v_{fa}^2 = K_2.(v_d^2+v_A^2).BW$ (for a flat noise spectrum)

$$\Rightarrow \sigma_T^2 = \sigma_{TTS}^2 + K_1 . K_2.(v_d^2+v_A^2)/(\mathbf{BW . A^2.}) + K_1.v_{ADC}^2/ (\mathbf{BW^2 . A^2}) + \sigma_j^2$$

$$\sigma_T^2 = \sigma_{TTS}^2 + \frac{0.35\, K_2}{BW\, SN_{FE}^2} + \frac{0.35}{BW^2\, SN_{ADC}^2} + \sigma_j^2$$

Better Resolution for higher Bandwidth, specially true if ADC contribution is dominant !!!

And we considering several samples ?

With waveform sampling:

=> Several Samples

=> Several measurement of the time

=> Averaging of some errors (those which are uncorrelated from sample to sample):

- digitizer noise
- a part of the digitizer jitter
- usually a very small part of the FE noise part (strong correlation exists between samples after filter)

$$\sigma_{T,N}{}^2 = \sigma_{TTS}{}^2 + \alpha\left(\frac{1}{N}\right)\frac{0.35\,K_2}{BW\,SN_{FE}{}^2} + \frac{1}{N}\frac{0.35}{BW^2\,SN_{ADC}{}^2} + \beta\left(\frac{1}{N}\right)\sigma_j{}^2$$

The improvement with oversampling is mostly only on the digitizer contribution

Some Key parameters of digitizers

- Power Consumption
- Input analogue Bandwidth.
- Sampling/Conversion Rate.
- Nb of coding bits.
- Noise.
- Non linearities: integral & differential
- **Distortions.**
- **Aperture Jitter σ_J**

All these parameters are taken into account in the **ENOB (effective number of bits)** parameter:

≠Log(Max signal/noise)/ Log(2) as sometimes said but measured **with a sinewave input of Max amplitude:**

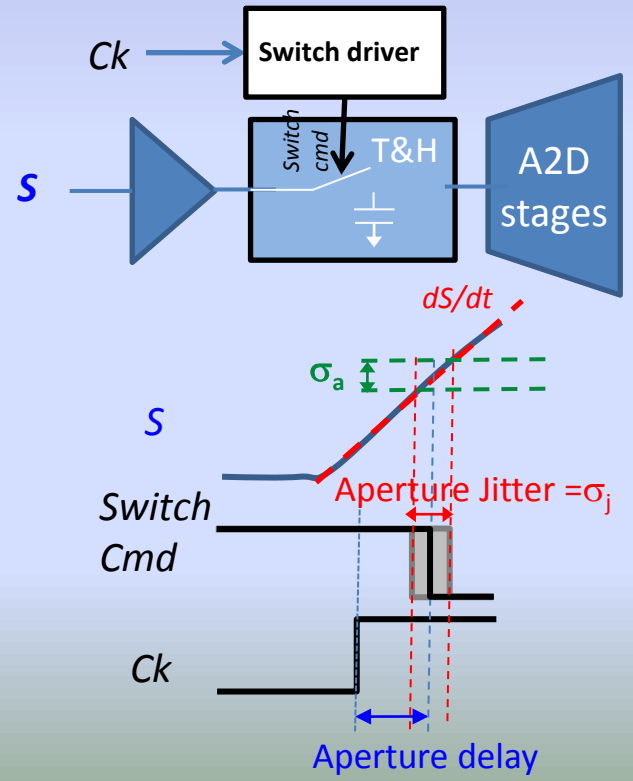
ENOB = [10 Log (P_S/P_R) -1.76]/6.02

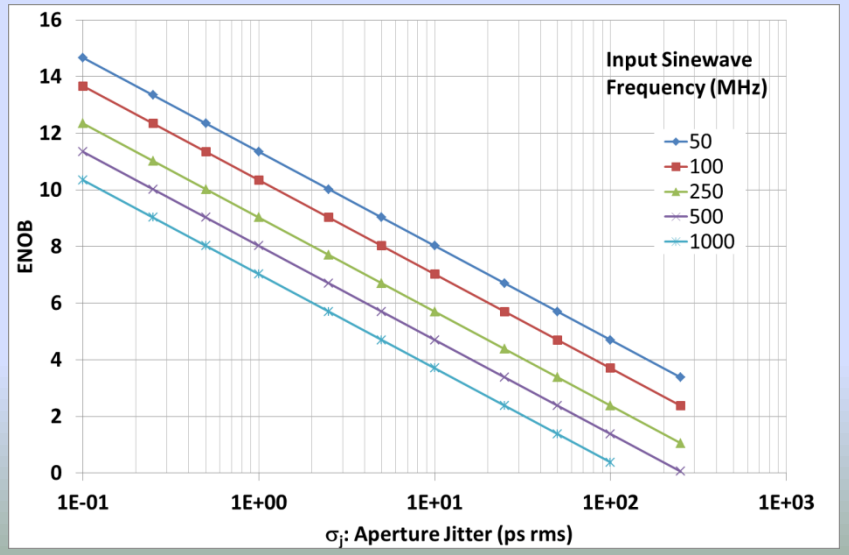
P_S is the power of the input sinewave, P_R is the power of the residues (when the sinewave is subtracted to data)

Highly depends on the sinewave Freq. =>

Aperture Jitter limits ENOB :

$ENOB_J$ = (-20 Log ($2.\pi .\sigma_J .F_{sine}$))-1.76)/6.02





Is ENOB the right parameter for jitter calculation?

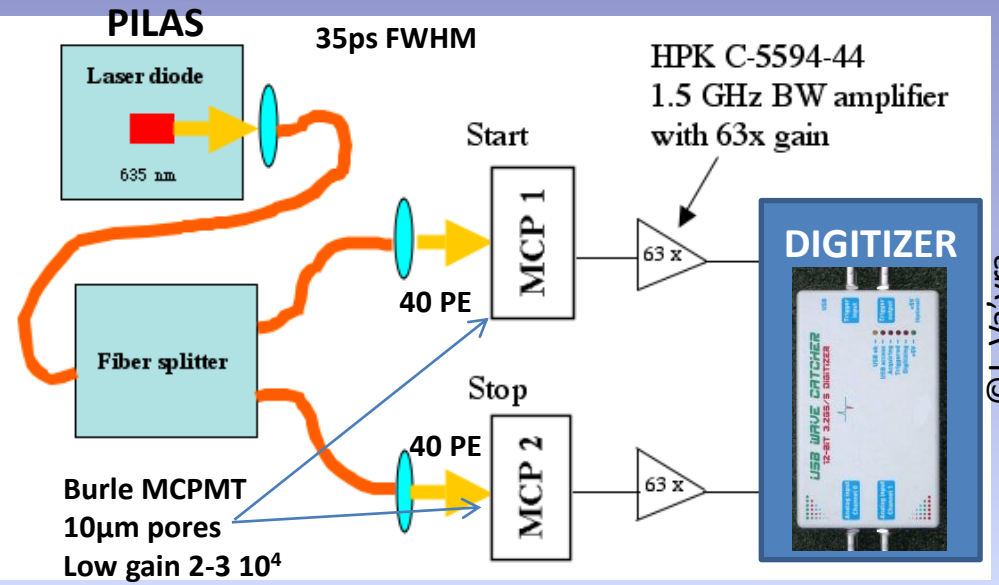
- For a sinewave with F_{sin} frequency the variance of the sample amplitude is:

$$\sigma_{ADC}{}^2 = \frac{1}{12 \cdot 4^{ENOB}}$$

- σ_{ADC} contribution is overestimated (underestimated) if the slope of the signal is smaller (larger) than the max slope of the sinewave.

- Practically, ENOB can be used for a very first estimation of the sampling jitter, if the signal slope is similar to the one of the sinewave used to specify ENOB. Otherwise we have to really know what are the separate contributions of aperture jitter and of ADC noise and distortion.

Illustrating example: will be used all along the presentation

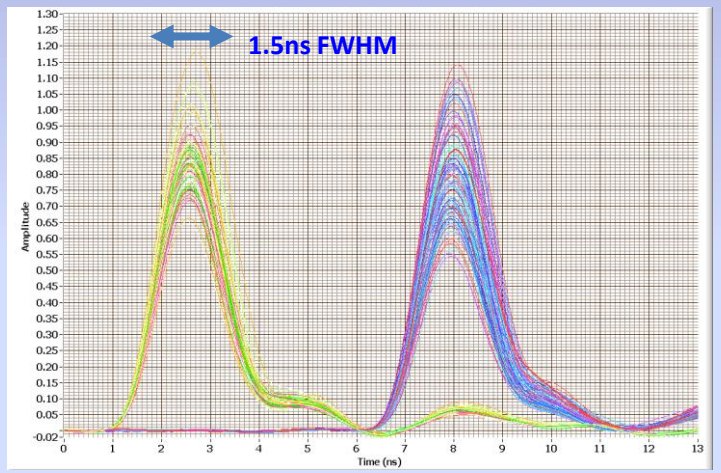


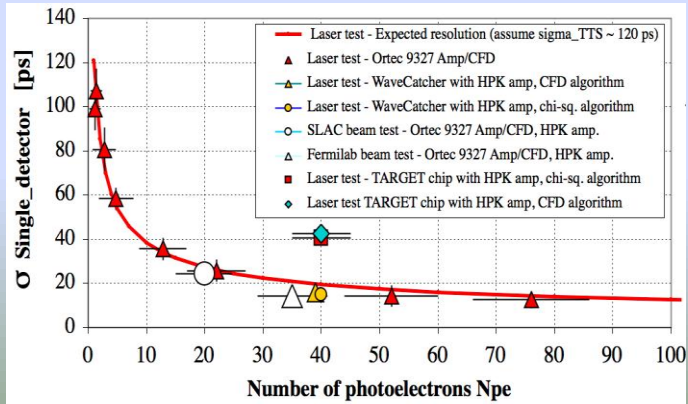
J. VaVra's test setup @ SLAC

- Setup and results described in detail in [Breton].
- A average value of 40 PE detected by each MCPMT .
- Spread of timing difference measured
  - => timing resolution $\sigma_{SINGLE} = \sigma_{DIF}/\sqrt{2}$
- Data digitized with the **Wavecatcher** module
- Use **SAM analogue memory** ASIC
- 2 channels 3.2GSPS/12bit/BW=450MHz. < 8ps rms resolution
- Factor 2 max amplitude fluctuation
- Very good signal/noise= 550 ! σ_{noise}= 1.5mV rms
- Signal widened by digitizer BW : FWHM => <800ps =>1.5ns

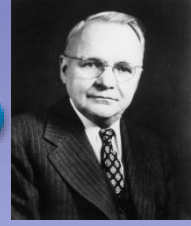
ANALOGUE REFERENCE: in the same conditions, using analogue CFD, TAC +ADC (resolution with pulser =3.4ps rms)

- σ_{SINGLE} = 17ps => 14ps with offline extra timewalk corrections. Low F= 20%.





How to choose Fs?

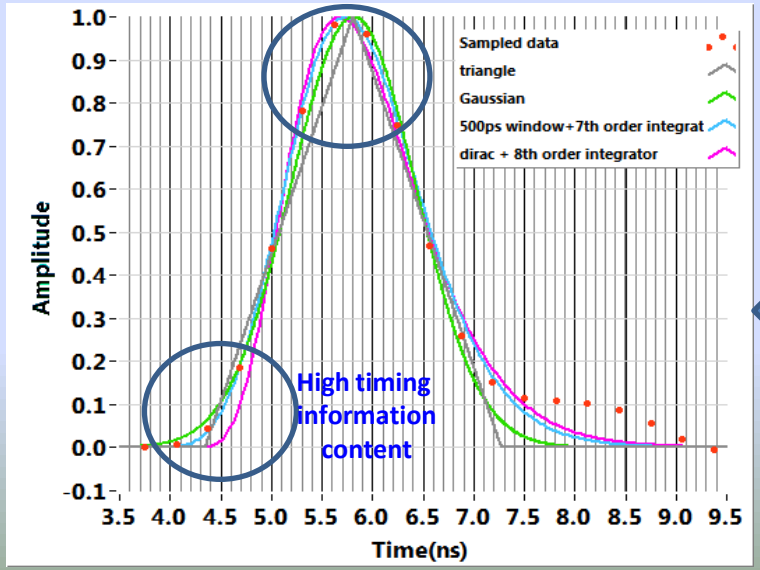


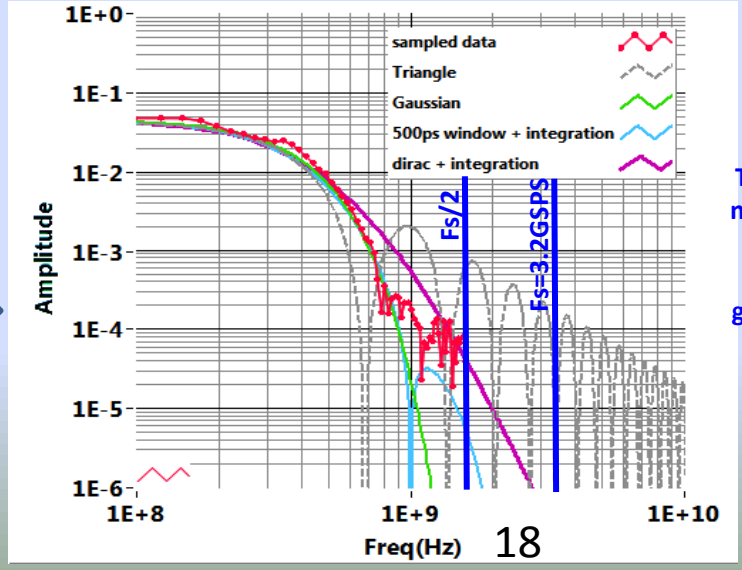
H. Nyquist

Ideally => the higher Fs is the better: but increases the cost and data throughput…

- **In the frequency domain:**
  - Nyquist-Shannon say : Fs must be > 2.$F_{max.}$ (Fmax is the largest frequency of the signal (and of the noise) spectrum).
  - If not: aliasing => a part of signal and noise is transformed in HF "noise", impossible to filter
  - Mandatory for digital filtering
  - Fmax is much larger than the -3dB BW ! Depends on the system filtering order
  - There is no obvious way to calculate easily Fmax from the pulse's basic parameters (tr,tf,FWHM):
    => find Fmax from a calculated or measured spectrum.
    => Set it using a known antialiasing filter.

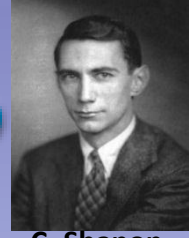
Various « models » emulating the MCP-PMT Pulse with same FWHM,tr,tf
=> Very Different high frequency behaviour





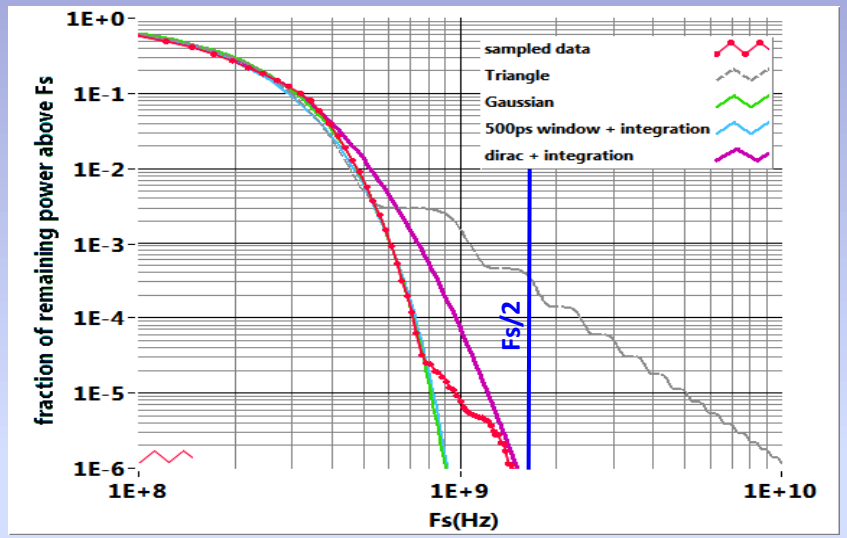
The analogue memory input stage is naturally a good antialias filter !

How to choose Fs?


C. Shanon

- **In the frequency domain :**

  Best criterion: Plot (fraction of the power remaining above Fs) vs Fs

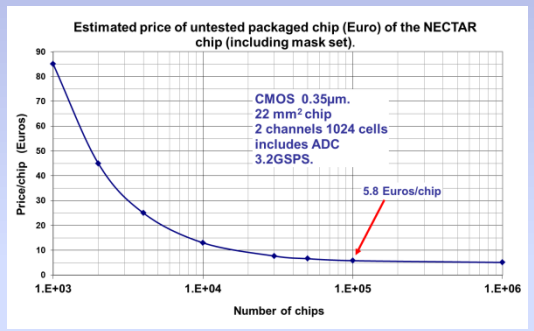


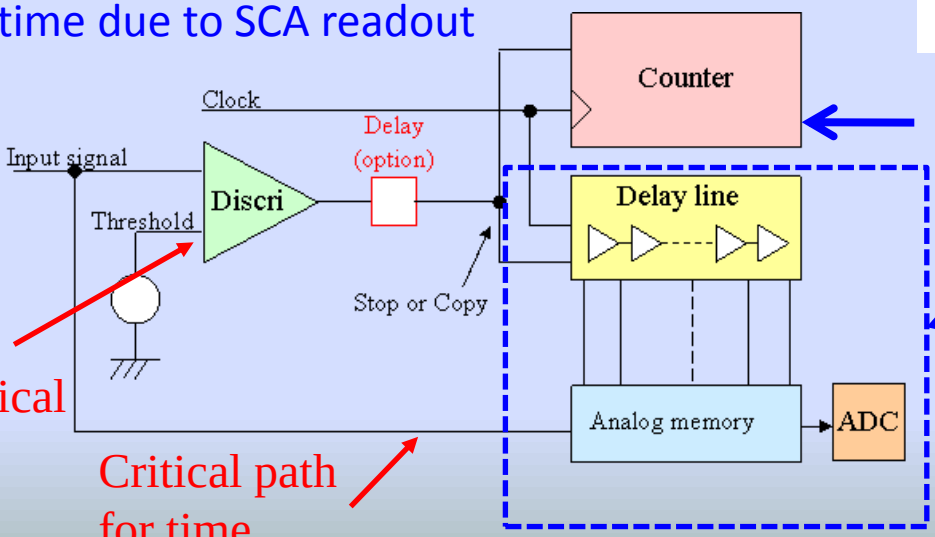
- **In the time domain:**
  - Oscilloscope manufacturers rule of thumb: > 5-10 times the BW.
  - To emulate “analogue-like” timing algorithms, a minimum of 3 samples is required in the trailing edge. More samples allow to use simpler algorithms (linearization).
  - If F_s is >> 2 . F_{max} , 2 consecutive samples will be highly correlated => there is redundancy between them => oversampling: can be used to decrease (by a factor √N) the noise/quantization contribution of the digitizer

Ultra Fast SCA designs for timing

Ultra fast SCAs for timing

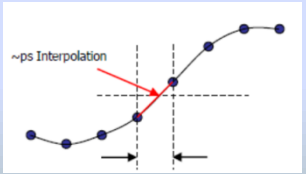
- High sampling rates help for timing
- Higher sampling frequencies => simpler algorithms .
- Continuous ADCs are the perfect digitizers but at least 99% of data are often going to the bin at owner's expense! (power, FPGA, …)
- Ultrafast analogue memories are a good alternative to ADC fro frequency above >500MHz.
- Fast, high dynamic, low data throughput
- Low power consumption. Low price
- High integration Level
- But deadtime due to SCA readout





Coarse timing

Waveform capture => Fine timing



No more critical For timing

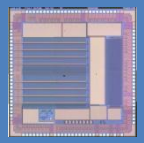
Critical path for time measurement

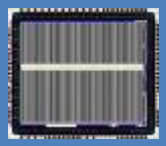
Ultra fast switched capacitor arrays in the world

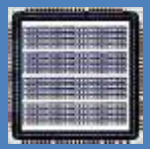
G. Varner Univ. Hawaii

Many chips for different projects
Buffered and unbuffered
Very deep arrays
ADC on chip.
Philosophy => pushing the limit of the SCA technology


Straw3


Labrador


Labrador3


Target


BLAB family

H. Frisch et al., Univ. Chicago


ps family

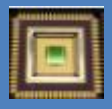
Goal: reach a 1ps precision !
Pioneering R&D work
130nm IBM
18 GSPS, 256 samples, 6ch
ADC on chip

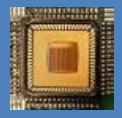
Initiator of a networking activity on SCAs and ps-timing

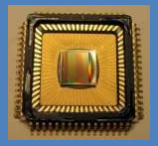
S. Ritt, R. Dinapoli PSI

Universal chip for many applications
8 + 1 channels 1024 cells
5GSPS, 950 MHz BW
Low power consumption
Short readout time
Several possible modes of operation

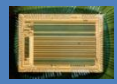

DRS1

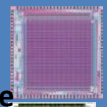

DRS2

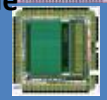

DRS3

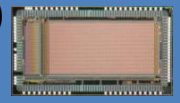

DRS4

D. Breton IN2P3/LAL
E. Delagnes CEA/Saclay


ARS


MATACQ

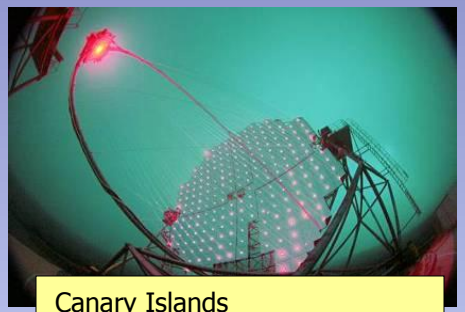

SAM family


Nectar

More than 120.000 SCAs operating worldwide
Buffered (f_{-3dB} 400-500MHz) 3.2GSPS
High dynamic range
Robust (minimum calibration or ext. control)
Conservative technologies
Moderate depth 256-1024 cells/ 2ch
On-chip ADC in the last chip

From an Orignal slide of S. Ritt

Ultra fast SCAs around the world: some applications

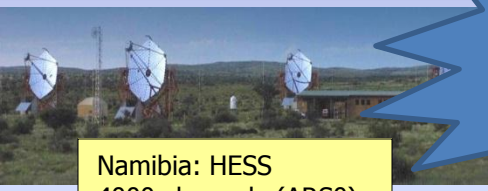


Canary Islands
Magic II: 400 channels DRS 2
Magic Upgrades: DRS4

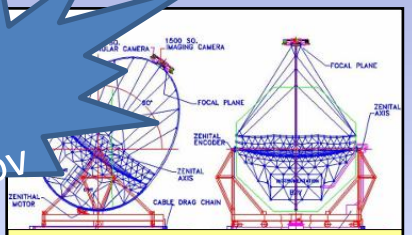
CTA ~100.000 channels (planned)

BPM for XFEL@PSI
1000 channels DRS4 (planned)

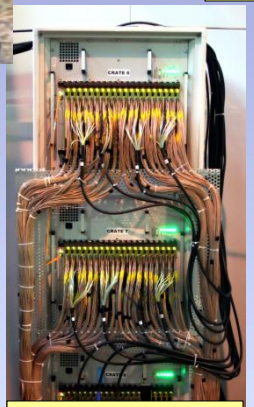
Gamma TeV Cherenkov



Namibia: HESS
4000 channels (ARS0)

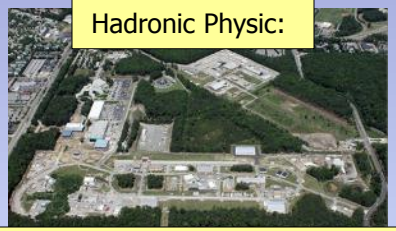


MACE (India) 400 channels
DRS4 (planned)



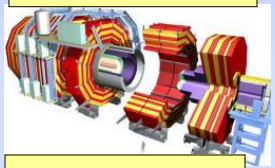
MEG @PSI
3000 channels
(DRS4)

Hadronic Physic:



JLAB(USA): calo DVCS (ARS0)

CMS Calo (CERN)



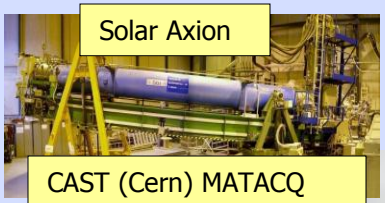
Monitoring
(MATACQ)

Namibia: HESS-2
2000 channels (SAM)

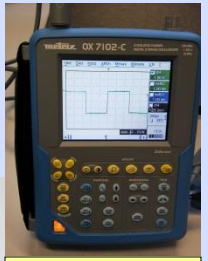
Neutrino –Cosmic Rays / Radio

ANITA (Ballon Antarctica)

Solar Axion



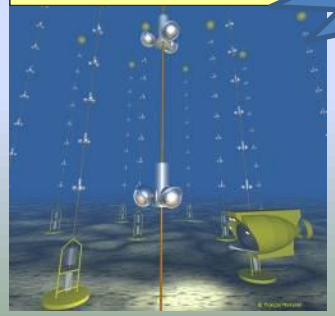
CAST (Cern) MATACQ



Oscilloscopes
(MATACQ)

Neutri nos

Antares
(Mediterranean sea)
: 1000 channels ARS1



AMANDA-IceCube
(Antarctica)

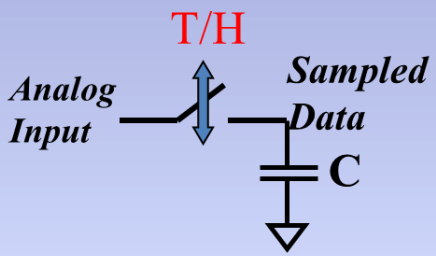
Cosmic Ray Radio

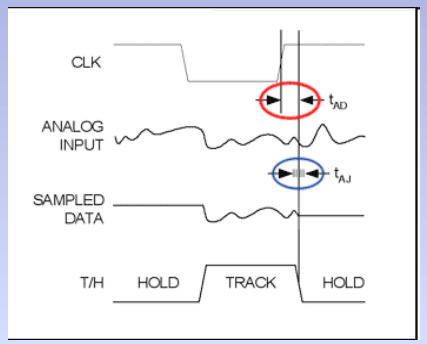
Codalema (France)
: 240 ch MATACQ

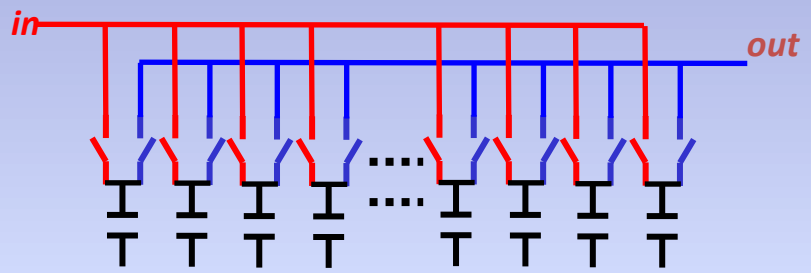
TOF PET Siemens

SCAs 1.0

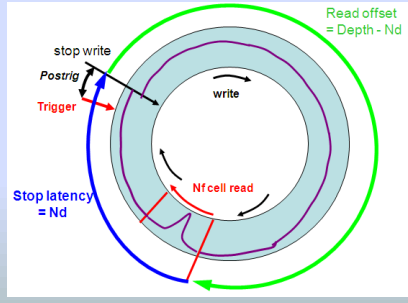
- Introduction of Analogue Memories for HEP experiments at the end of the 80's by S. Kleinfelder.
- Principle: Sample & Store an incoming signal in an array of capacitors, waiting for (selective) readout and digitization= bank of Track & Holds





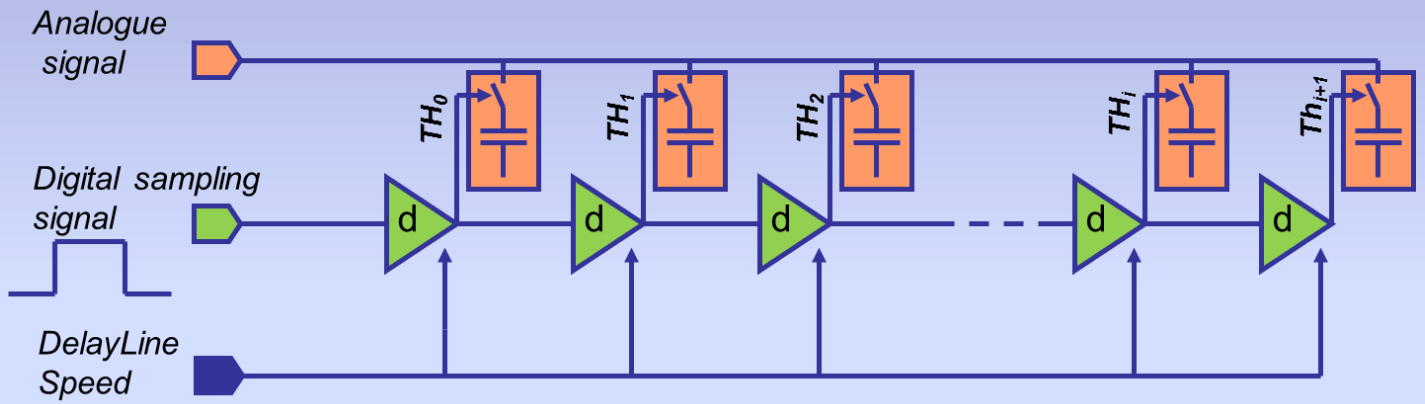


- A/D conversion can be:
  - delayed (waiting for an external decision) slower than sampling frequency.
  - Slower than Sampling Frequency.
  - Shared between channels => first level of data concentration
- More than 13 bit dynamic range. High integration: 12 to 128 channels, depth of few hundred cells. Low power.
- Sequential or simultaneous (double port FIFO-like) operations.
- Sample & Hold commands generated by Flip-Flops => Sampling frequency limitation.
- Widely used with sampling rates < 100 MHz in many experiments (ATLAS,CMS,STAR,T2K…) as Level 1 buffer. Region of interest readout.

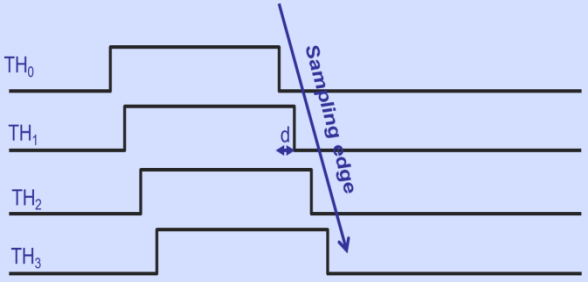


SCAs 1.0

- Introduced in 1990's again by **S. Kleinfelder** (ATWR, ATWD chips).
- The Sample & Hold commands are now generated using a pulse propagating through a delay line with $\mathbf{N_{TAP}}$: $F_s = 1/d$ => multiGSPS operation possible even in ~1µm technologies.

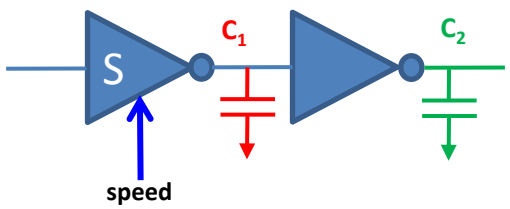


- F_s tunable through an analogue command
- In the early designs:
  - The digital sampling signal input was a single pulse = trigger => need for an analogue delay on the analogue signal path to generate the "Pretrig".
  - The width of the sampling pulse was defined by the width of the digital pulse.

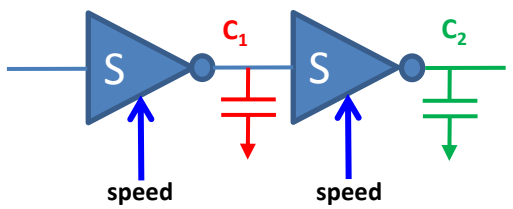


Delay elements zoology

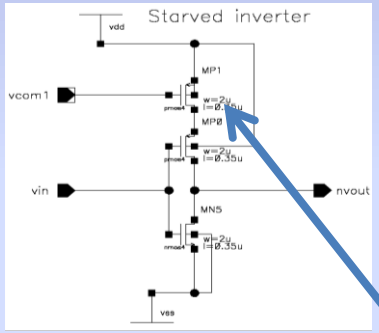
- Basically the same as those used in digital TDCs, made with 2 cascaded **inverting** cells :



Fast.
2nd inverter reshapes the signal => sampling edge always sharp

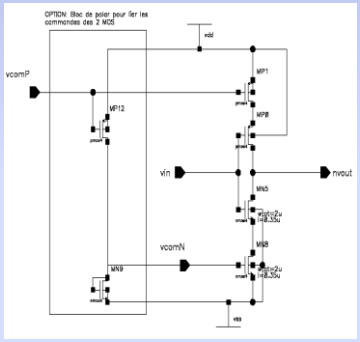


Slower
Symetrical egdes if C1=C2

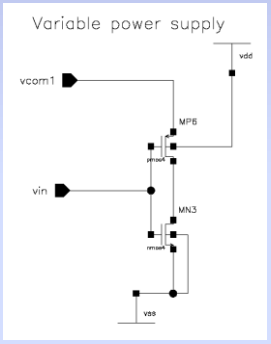


$$t_p \propto t_r = 2.C_l.Vdd.\frac{L_{MP1}}{W_{MP1}.K_{MP1}.(V_{GS\,MP1}-V_{TH})^2.}$$

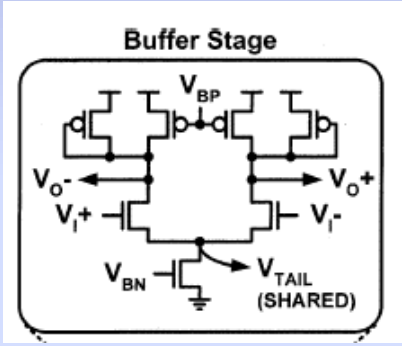
Only the rising edge is slowed down



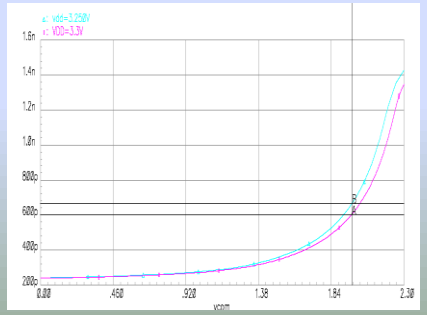
More symetric output



Highest speed, but requires low impedance command



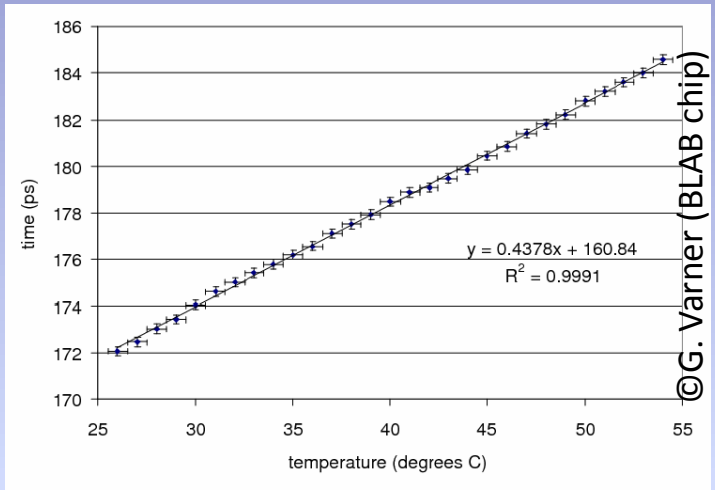
Differential. Low jitter. But Static power.



Modulate PMOS conductance is better for low jitter.
⇒ Separate Vdd for DLL only.
⇒ Vdd – Vcommand easy to filter

Delay control

- Delay elements sensitive to temperature, process, ageing…:



- 2 used philosophies:
  - Servo-control loop (PLL, DLL).
  - No servo-control:
    - Delay control voltages externally generated.
      - Delay= f(Control voltage) first calibrated and stored in a LUT used to command DAC.
      - Temperature dependency can also be calibrated and corrected
    - Delays measured using an extra channel to digitize a clock/ timing signal

Delay Line, Jitter & non linearity

2 sources of aperture jitter :

- Random aperture jitter (RAJ).
- Fixed Pattern Aperture Jitter (FPJ) equivalent to Non Linearity of TDCs

• Along the delay lines, jitters are cumulative. If we consider that there is no correlation of the jitter added by each delay:

• RAJ, the aperture jitter @ tap j will be

$$\sigma_{Rj} = \sqrt{j}.\sigma_{Rd}$$ if σ_{Rd} is the random jitter added by a delay tap

• FPJ

$$\sigma_{FPj} = \sqrt{j}.\sigma_{FPd}$$ for a free running system

$$\sigma_{FPj} = \sqrt{\frac{j.(N-j)}{N}}.\sigma_{FPd}$$ if the total delay is servo-controlled (max @ middle)

if σ_{FPd} is the spread of unitary delays ($=\sigma_{DNL}$) given by transistor matching and N is the DL length.

Short DL => Less Jitter (both kinds)
Fixed Pattern Jitter can be measured and corrected

Timing calibration: statistical method

search of **zero-crossing** segments of a free running sine wave

=> length[position]

- **Calculate the mean value for each position and normalize by the average step value**

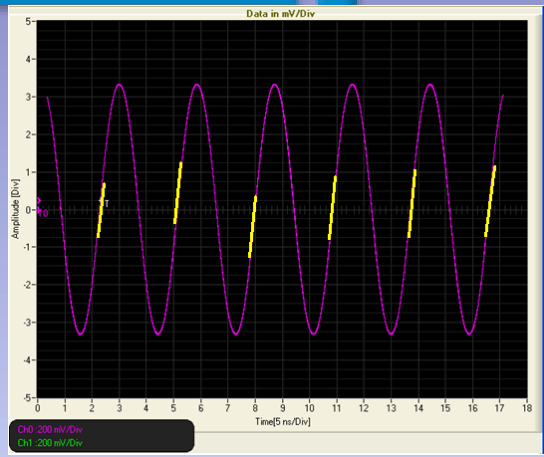
  =>time step duration (DNL)

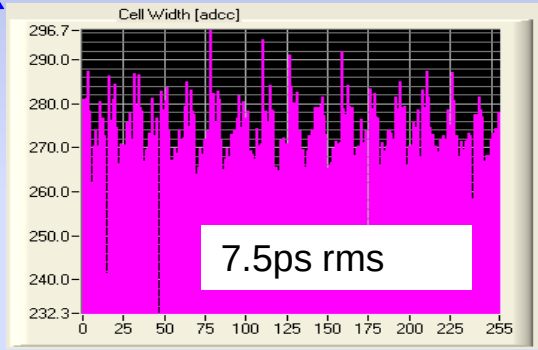
- **Integrate this curve – expected value**

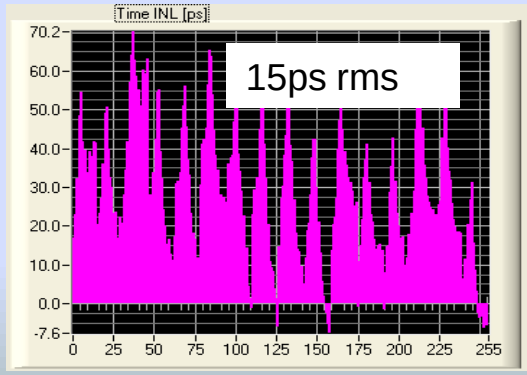
  => Fixed Pattern Jitter = correction to apply to the time of each sample.

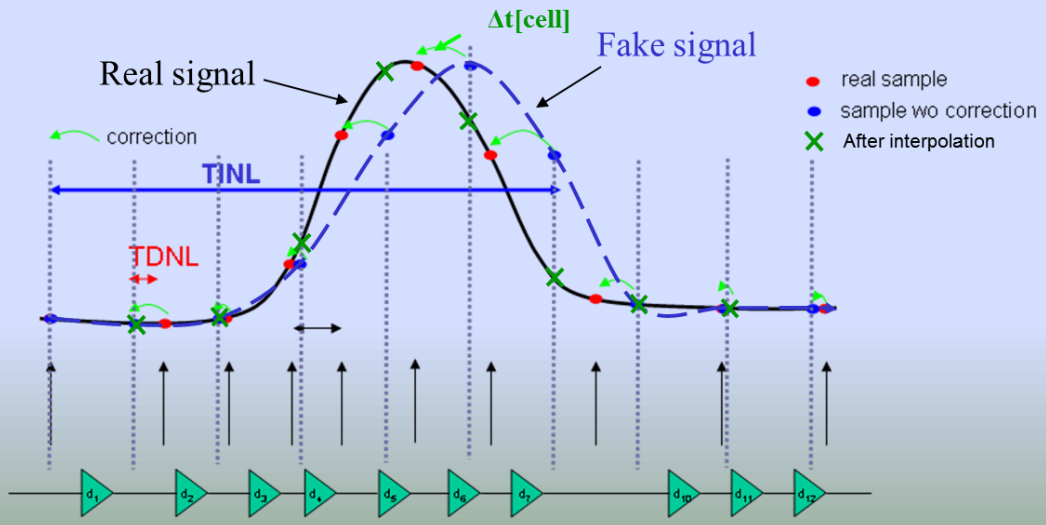
Depending on the timing algorithm:

- **Simple addition on T_{sample}**
- **Calculation of real equidistant samples by interpolation or digital filtering.**

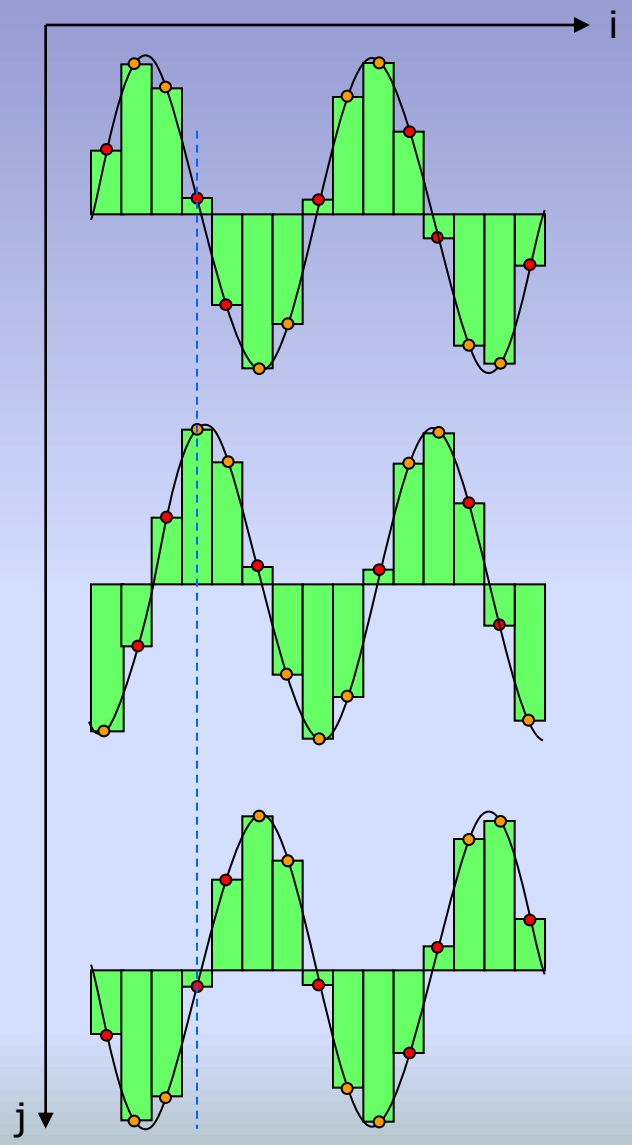








Timing calibration: sinewave fit method



$$\chi^2 = \sum_{j=0}^{500}\sum_{i=0}^{1024}(y_{ji} - (a_j \sin(i\frac{2\pi}{f_j} + \alpha_j + \beta_i) + o_j))^2 \rightarrow \min$$

y_{ji} : i-th sample of measurement j

a_j f_j α_j o_j : sine wave parameters

β_i : phase error $\rightarrow$ fixed jitter

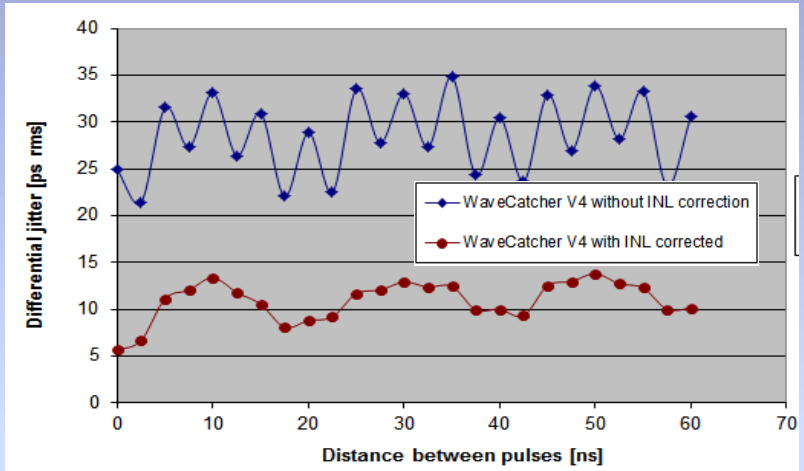
"Iterative global fit":

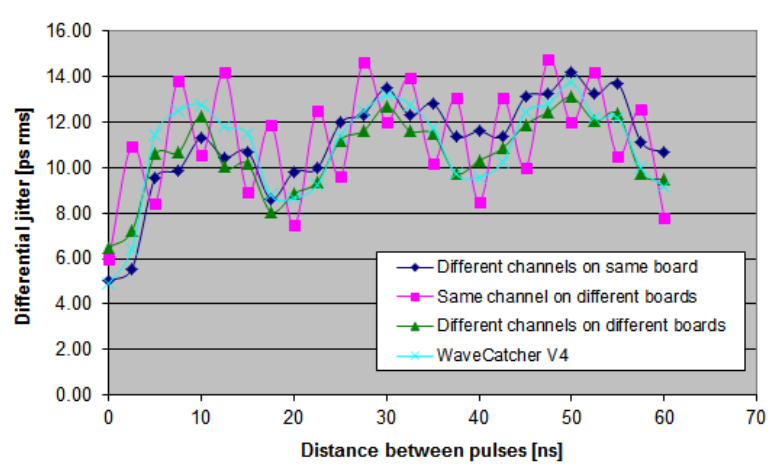
- Determine rough sine wave parameters for each measurement by fit
- Determine β_i using all measurements where sample "i" is near zero crossing
- Make several iterations

S. Lehner, B. Keil, PSI

Fixed Pattern Jitter after correction

Example of SAM/SAMLONG : the correction works very well but is never totally perfect. Checked by sending 2 random pulses with variable distance (differential jitter/ √2)





Mean Jitter = 20ps rms before correction
= 8.5ps rms after correction

Differential jitter is always smaller @ short distance

Remains valid for months.

Same performances reached with chips on **different boards** with the same clk.
=> Similar to what happen on a large timing system

Results similar reported by Hawaii and PSI, but :

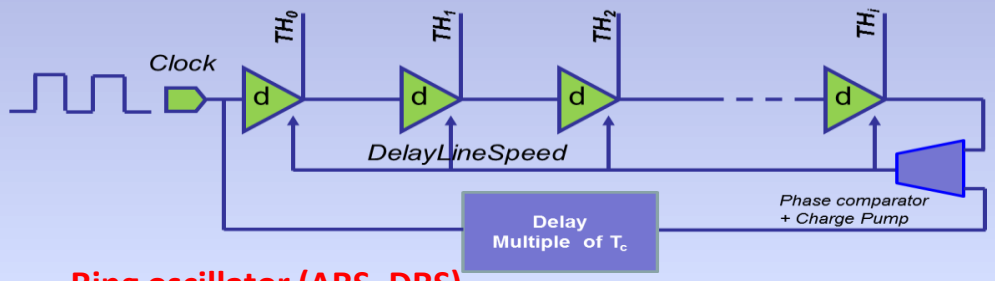
- as delay lines are longer jitter before correction is worst
- very large improvement after calibration

SCAs 2.0

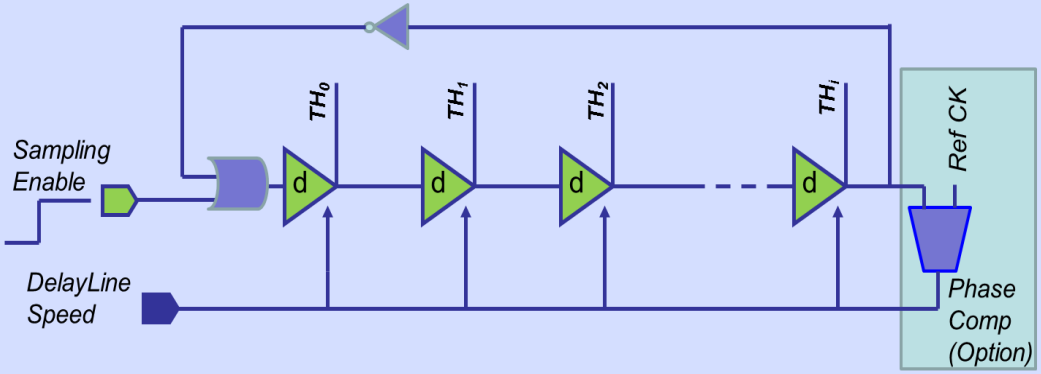
- Continuous operation required to permit Pretrig operation without analogue delay line:
- A rotating sampling pulse is required. Several designs proposed
  - **Pulse regeneration (SAM, MATACQ, PSEC…):**

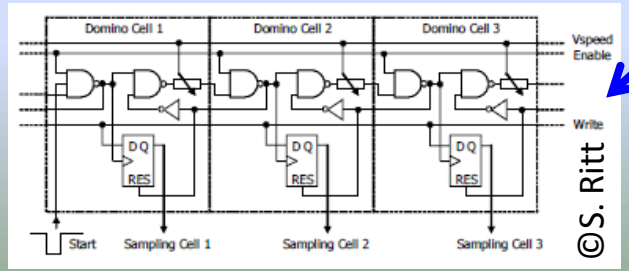
    - A new pulse is generated at input with a $d.N_{TAP}$ periodicity

    => $d.N_{TAP} = N.T_c$ period of an external clock.: servo-control with phase comparator



  - **Ring oscillator (ARS, DRS).**

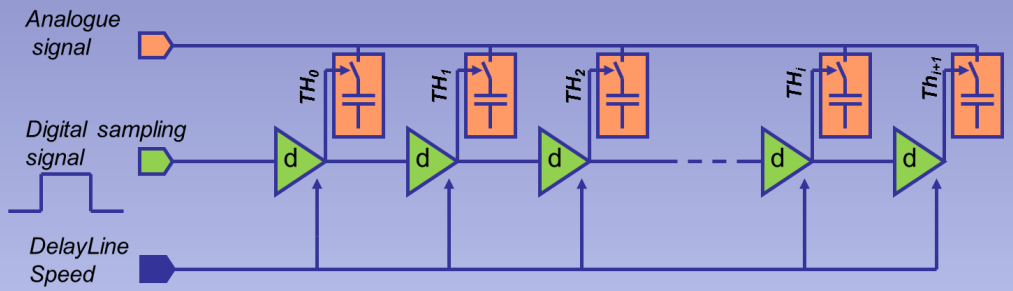


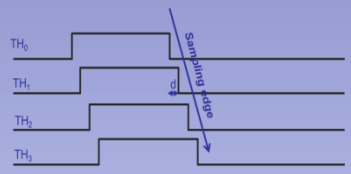


To avoid spread of the pulse length or even vanishing due to different propagations of the 2 edges in a long DLL:

- Use « long » pulses (not the one directly used to sample).
- Ensure edge symmetry
- Servo control of the 2 edges(ARS)
- Pulse biting (DRS): the propagating pulse is intentionally widened in each tap, then cut by the rising edge of the pulse taken on one of the next cells => DL pulse width = fix number of cells.

Sample & Hold command signal





All the cells with switch command = 1 are connected simultaneously to the analogue bus:

⇒ The duration of the sampling pulse must be controlled accurately to guaranty a constant load on the analogue bus => constant bandwidth.

=> In most of the designs, excepted DRS family, the pulse propagating in the delay line is quite long (to avoid vanishing effect).

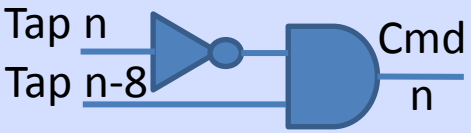
=> Need for a pulse shaping block between the delay line and the switches.

=> monostable

=> use of two taps of the delay line

=> clock period reduce by a fix amount (SAM…)

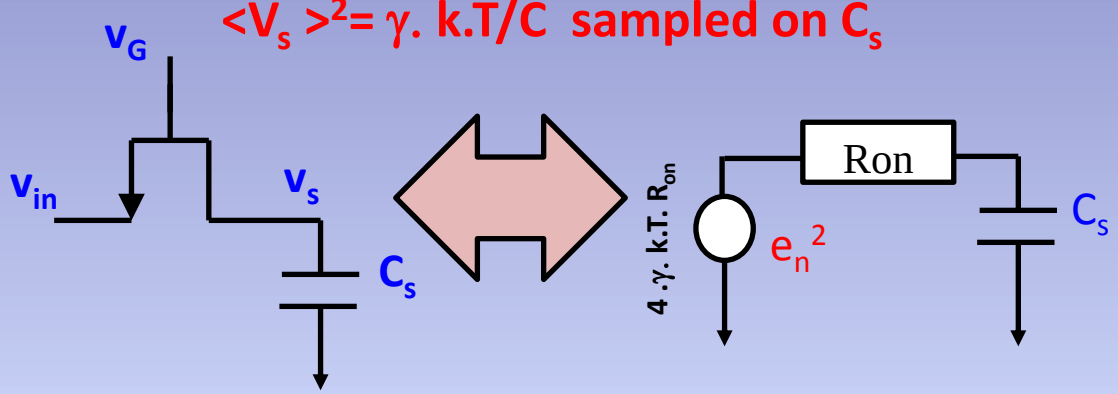
=> Already performed by the “pulse-biting “ cells of the DRS.

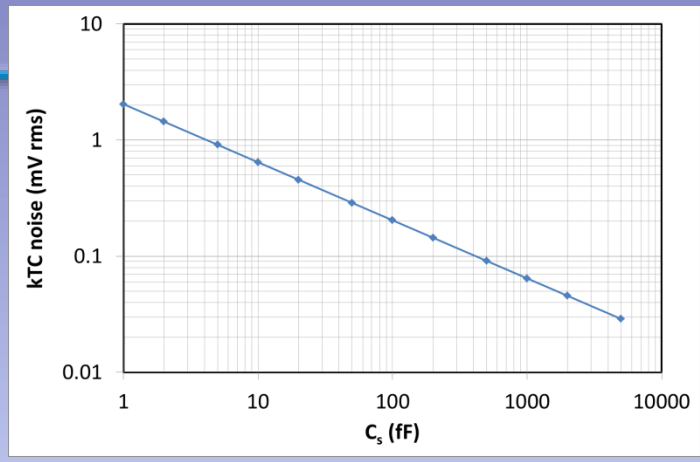


=> in all the case, **the falling edge of the switch command is the important ONE** and must come ~ directly from the DLL out

Storage Cell

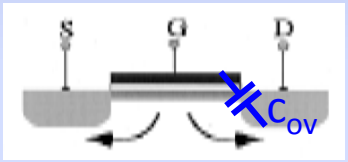
- Noise: absolute noise limite = kTC noise





- Channel charge + command feedthrough injected in Cs when sampling:

$$\Delta v_s = \frac{k.W.L.Cox.(VG-Vin-VT)}{2.Cs} + \frac{(VDD-VSS).Cov}{C_{ov}+Cs}$$

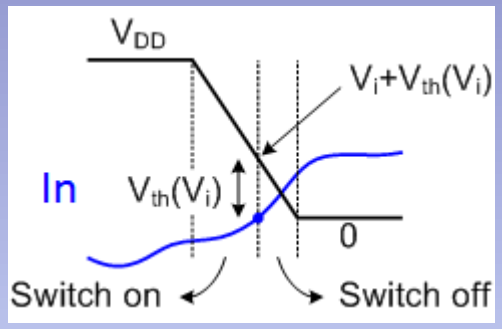


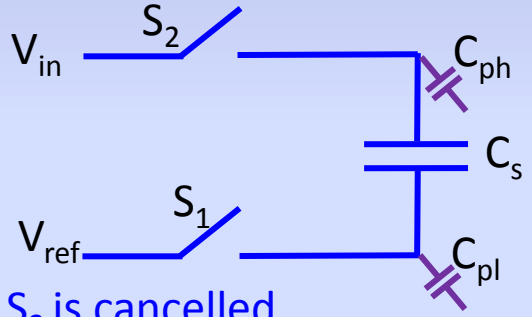
  - First term dominant
  - ~ proportional to $1/C_s$ and to the R_{on} of the switch (if L min)
  - At first order: constant + a term proportional to V_{in} =>Offset + gain different of 1.
  - But transistors mismatches => Offset & gain spread along the SCA.
    => Possible calibration & correction
  - "Dummy switch" technique inefficient => increase of the spread.

Large C_s is good for noise & uniformity !

Storage Cell: Bottom plate sampling

- Edges of the switch command is not infinitely fast.
- Transistor cutoff at $V_G = V_{in} + V_T$
  - =>Dependency of the sampling time with Vin
  - => Distortion, Jitter.
- For a 100ps edge => 50ps error possible !
- Solutions:
  - Live with it, use the fastest possible edges and a reduced dynamic range.
  - Bottom plate Sampling (SAM, DRS):
    - S_1 has a constant source voltage
    - S_1 opened before S_2 => sample
    - Aperture time now independent of V_{in}
    - If "flip around" readout, the charge injected by S_2 is cancelled
      - => Charge injection does not depend on V_{in} .



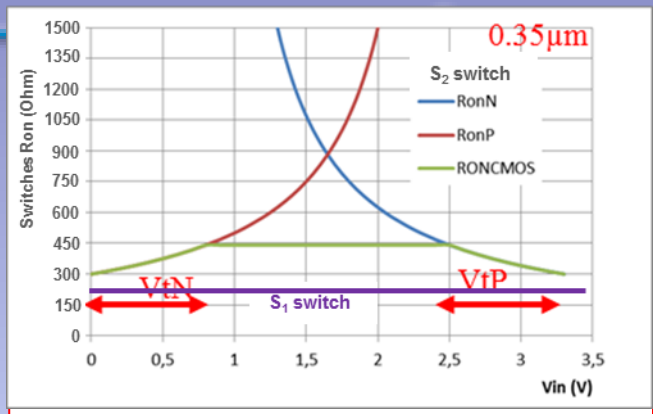


- **Drawbacks:**
  - => S_1 added in serie => lower BW.
  - => generation of S_1 command
  - => Less compact cell => more parasitic capacitance

Storage Cell: Bandwidth

$$BW_{cell} = \frac{1}{2.\pi.Ron.Cs}$$ **small C_s is good for BW**

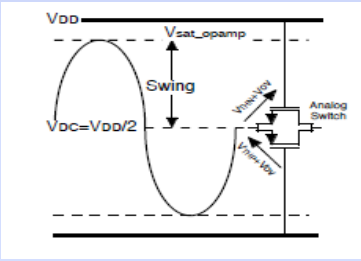
$$with\ R_{on} \approx \frac{1}{gds} = \frac{1}{\mu.Cox\frac{W}{L}(VG - Vin - VT)}$$

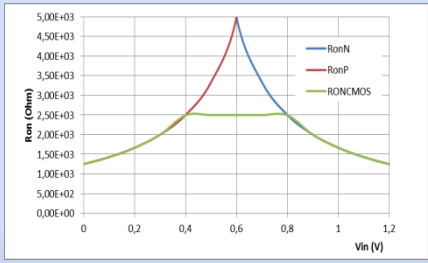


In SAM: S_1 & S_2 switches
$R_{S1}+R_{S2}$ = 600 Ohms
BW_{Cell}= 820 MHz (C_s=300fF)

- Minimum L for max Ron with smaller parasitics.
- BW_{cell} vary with Vin => distortion.
- BW_{cell} is affected by transistor mismatch
- BW_{cell} should not be the contribution limiting the BW
- Possible strategies to limit distortion:
  - Use NMOS only and limit the range to low voltages (DRS)
  - Linearized by using NMOS & PMOS in //and swing centered to vdd/2
  - Bootstrapped switches => never used in SCAs.
- On a given technology for a fix BW_{cell}, Q_{inj} is independent of C_s.
- **Technology scaling:**

  $\Rightarrow$ Lower R_{on} => higher BW but Smaller linear region

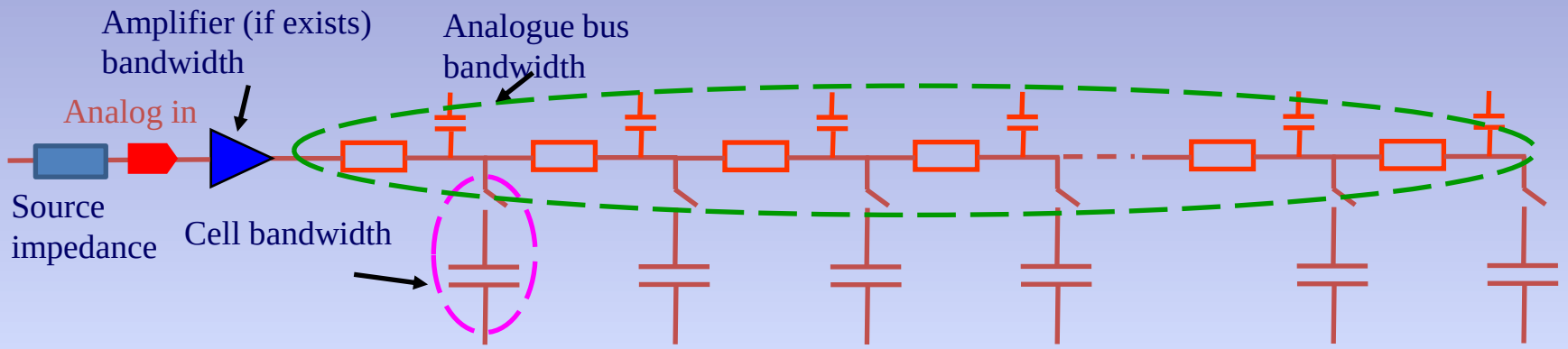




The cell settling time =Ln(precision).R_{on}.Cs must be < switch command duration for good signal tracking.

Global BW

- Combination of the “input bandwidth”, of the possible input buffer, of the analogue bus bandwidth of the cell bandwidth (generally not the dominant term)



Analogue bus resistivity is not 0 => lumped RC filter

=>BW variation along the chip => signal distortion

=> use the metal with lower resistivity

=> large bus width better=> but increase of the overall Cin ! => trade-off

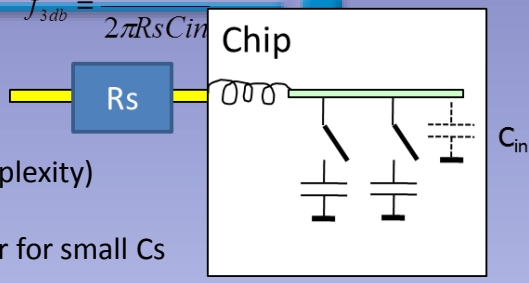
=> same effect with the capacitance reference bus

=> better for shorter bus (narrow cells or less cells/bus)

2 % BW variation measured on SAM which is optimized for this effect, has only 16 cells/bus division and has only a 400 MHz BW !! Huge effect seen on Psec3 chip.

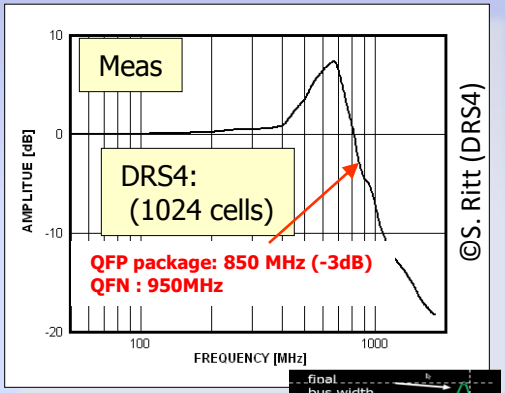
The issue of input Bandwidth

$$f_{3db} = \frac{1}{2\pi RsCin}$$



- Without taking the bonding inductor: InputBW is limited by the $R_s.C_{in}$
- $C_{in} = C_{package} + C_{pad} + N_{cell} \cdot C_{par}$ → C of the metal bus (increase with cell width/complexity) + Cdrain of the switches (prop to $1/R_{on}$) => smaller for small Cs
  - $C_{package}$: 2pF
  - C_{pad}: 2pF
  - $N_{cell} \cdot C_{par}$: SCA dpeth
- **BW_{input} = 800MHz for R_S=25 Ohm & C_{par}= 4pF**

- **1 Solution :** limit the SCA length, small C_s, optimize layout. **PSEC…**



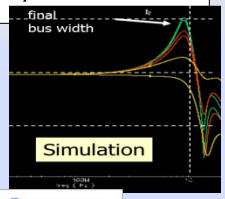
- **2nd Solution:** reduce Rs (ext. low output impedance amplifier)

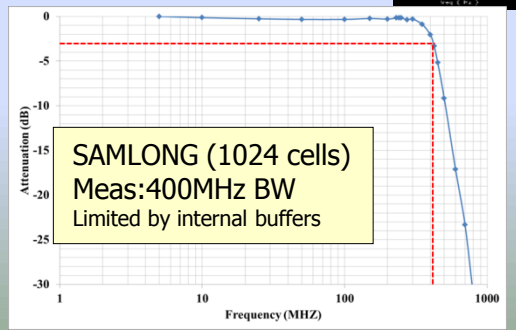
  Bonding -> RLC = 2nd order network => BW reduction and gain peaking increasing when R decreases and L/C increases). **DRS…**

  => use naked dies or very small packages

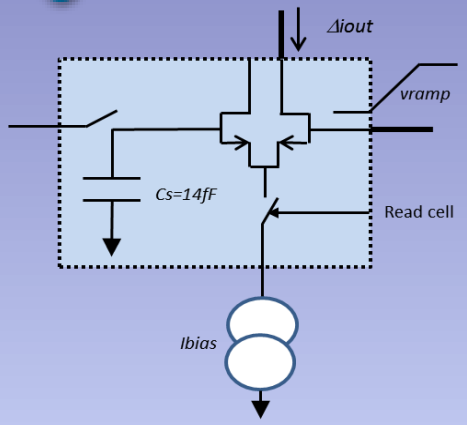
- **3rd solution:** Cut the analogue bus in subdivisions buffered by internal amplifiers with low input impedance => $N_{cell}.C_{par}$ is now replaced by the sum of the buffer capacitances.

  Good High BW and high slewrate buffers are difficult to design and power consuming **BLAB, Target,SAMs…**





Readout



Hawai'i chips:

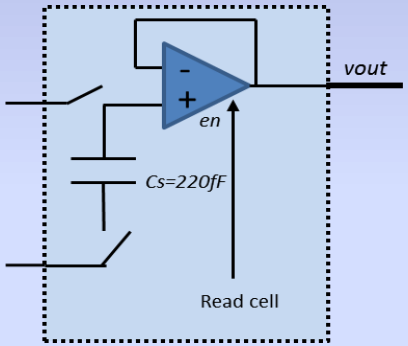
Smart Wilkinson Readout +AD conversion

1 Comparator/Cell

Counters & ramp generators can be inside or outside the chip

Parallel digitization of several cells

Need for one offset & one gain/cell



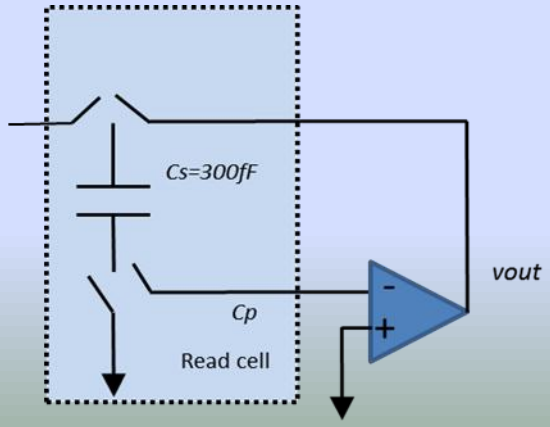
DRS4:

Voltage mode

1 buffer/cell (cut when not used) => low power

Multiplexing toward an external ADC

Need for one offset/cell



MATACQ/SAM…:

« Flip Around » Readout => cancels injected charge

Very well defined gain

1 ampli/ line of cells => critical design very sensitive to C_p

=> speed

=> noise (amplified by $(C_p+C_s)/C_s$

Multiplexing toward an ADC (on-chip in NECTAR)

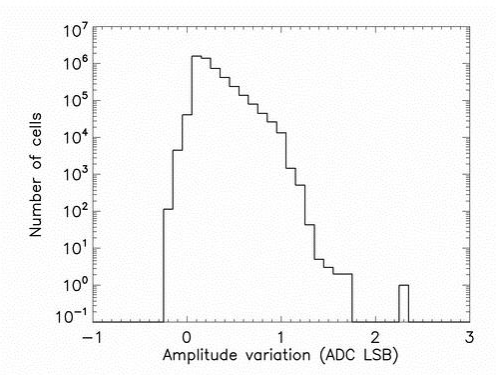
Need for one offset/line

Leakages

- Switches Leakage currents are discharging Cs:

  => voltage drop depending on time between Write and Read.

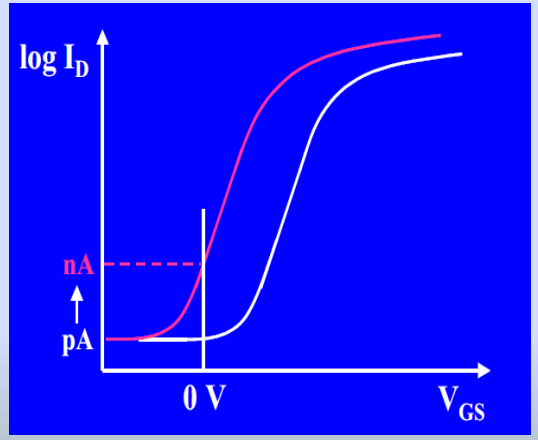
- Not an issue with old fashion (>0.25µm) technologies:



- Not a SD leakage but a current from S/D to bulk

AFTER Chip (T2K TPC) AMS 0.35µm
Distribution of the voltage drop on 120 chips * 65000 cells after 2 ms.
1 LSB = 0.5mV => 55fA. Not gaussian.

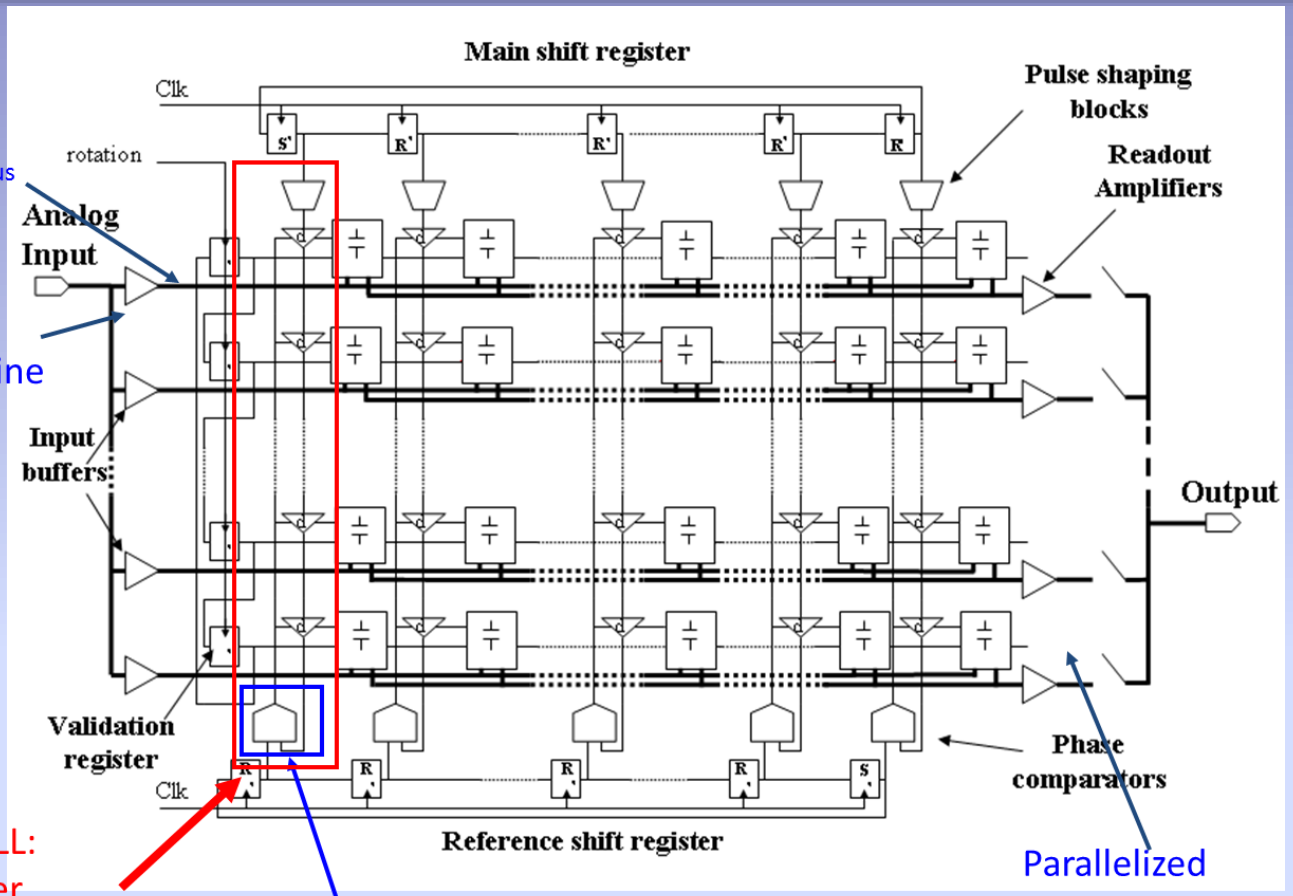
- **A real problem in deep submicron !!!:**



- Now a SD current
- pA scale leakages in 0.18µm
- 10 pA scale in 0.13µm => storage time limited to few µs
- Use of low-leakage transistors (but lower Ron)
- Larger Cs ? => against history !
- Reduce the range to work with negative Vgs in off-mode

Low VT + Low weak inversion slope

The matrix structure (SAM…)



Advantages: robustness => only 1 pedestal/Line to calibrate.
good timing (18ps rms) even with no calibration.

Drawbacks: complexity . Not scalable to a large number of channels/chip

3rd generation of SCAs

Common conclusions of the different groups: need for

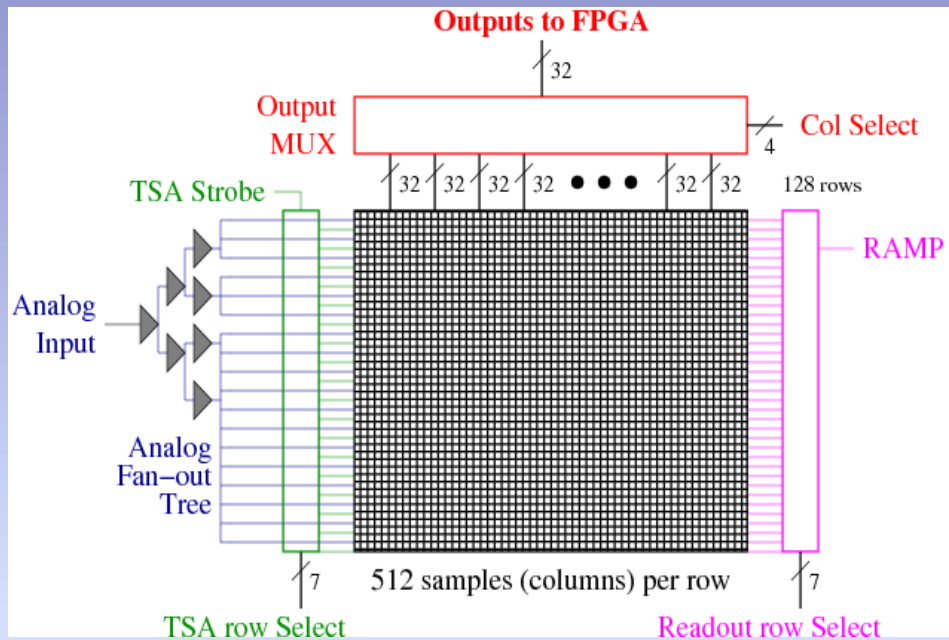
- High bandwidth, low jitter
  - ⇒ short analogue busses
  - ⇒ small C_s
  - ⇒ use of advanced technologies (0.11 to 0.18µm nodes)
- Large depth to accommodate longer latencies…
  - ⇒ Analogue bus segmentation
  - ⇒ And/or two stage architecture
- Fast readout
- Multiple events buffering to derandomize deadtime:
  - ⇒ Simultaneous R/W in a large array with pointer management
  - ⇒ Array of small-size banks of cells.
- Auto-triggering
- **These designs are already existing or being studied**

SCAs 3.0: the BLAB/IRS/Target family

- Very large depth (up to 64k)
- segmented in shorter rows using a tree distribution.
- Lines can be chained or addressed on demand in W or R modes (row select).
- Double port simultaneous RW operation demonstrated.

- 1GHz BW reached with BLAB2

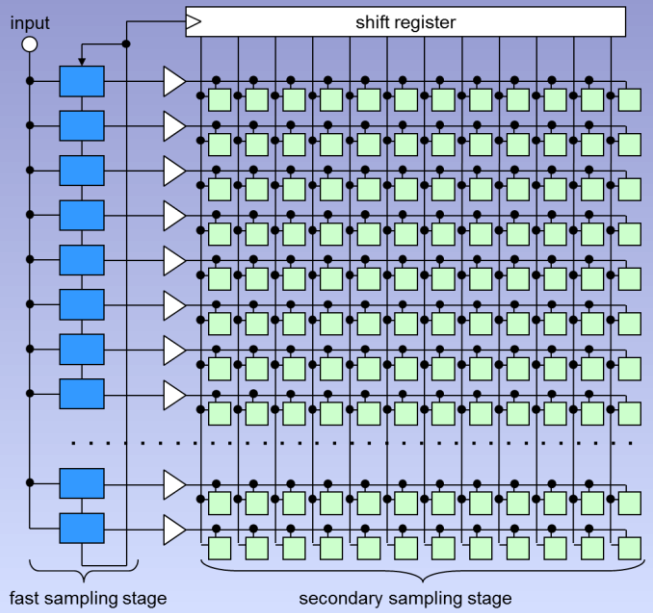
- Several prototype already designed with various :
  - block sizes
  - Number of channels
  - Input amplifiers



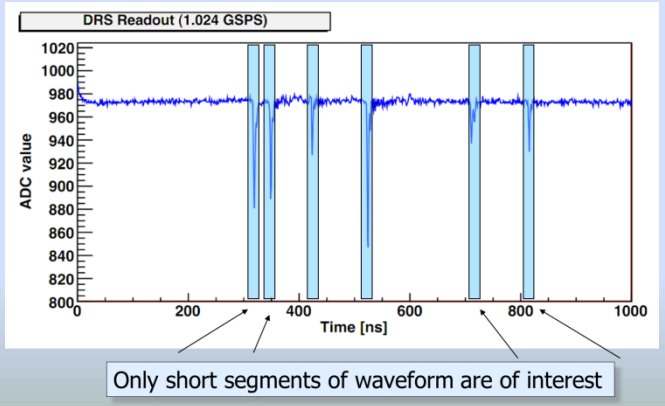
Specifications of the IRS chip

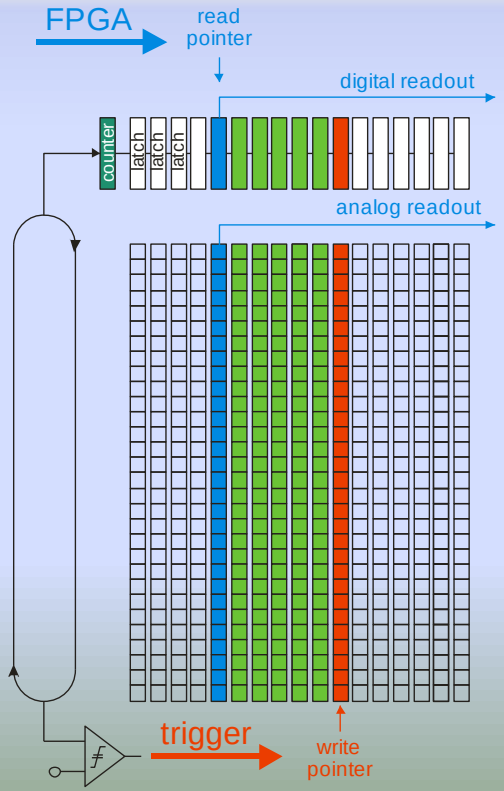
8	channels/IRS ASIC
8	Trigger channels
~9	bits resolution (12-bits logging)
64	samples convert window (~16-64ns)
1-4	GSa/s
1	word (RAM) chan, sample readout
16	us to read all samples
100's	Hz sustained readout (multibuffer)

SCAs 3.0: DRS5. planned for 2013



- 32 fast sampling cells at 10 GSPS
- 100 ps sample time, 3.1 ns hold time
- Hold time long enough to transfer voltage to secondary sampling stage with moderately fast buffer (300 MHz)
- Shift register gets clocked by inverter chain from fast sampling stage
- Multiple buffering => up to 2MHz with negligible deadtime





©S. Ritt

Recent ultra-fast SCAs

ASIC	Design Team	Internal Ampli ?	# chan	Depth /chan	Sampling [GSa/s]	-3dB BW MHz	Dyn. Range Bit rms	Storage Cap (fF)	Techno	Internal ADC?	In this conf.
DRS4	PSI	no	8	1024	1-5	900	12	250	IBM 0.25	no	
SAM SAMLONG NECTAR	Orsay/ Saclay	Buf	2 Fully diff. 1024	256 1024 1024	0.5-3.2 0.5-3.2 0.5-3.2	500 >420 >420	>12 11.3	300	AMS0.35	no no pipelined	N28-6
IRS2	Hawaii	no	8	32536	1-4		10	14	TSMC 0.25	wilkinson	
BLAB3A	Hawaii	Ampli	8	32536	1-4	1000	10	14	TSMC 0.25	wilkinson	
TARGET TARGET2 TARGET3	Hawaii	Buf Ampli Buf	16	4192 16384 16384	1-2.5	150	10	14	TSMC 0.25	wilkinson	
PSEC3 PSEC4	Chicago	no no	4 6	256	1-16	>300 >1600 prelim	10	20 20	IBM 0.13	wilkinson	NP2.S-75

Digital Timing algorithms

Digital Timing Algorithms

There is no Magic universal algorithm working for all setups:

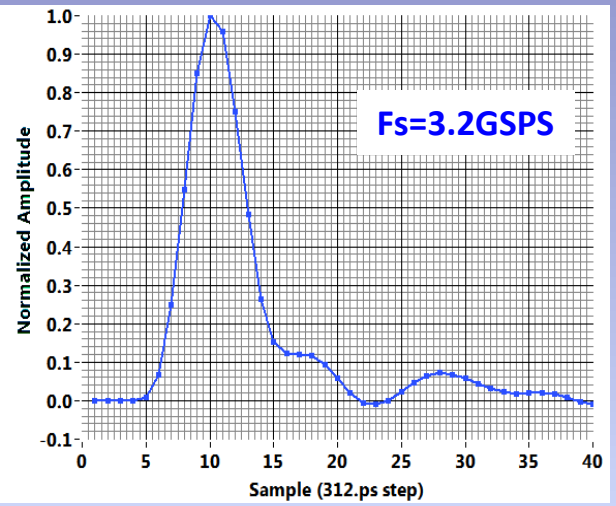
Results depends on:

- Physics: ie in the case of detection of a lot of photons is the best timing given by the first photon or by the average time of photon ?
- Resources available for data treatment.
- Time resolutions better than 1% of the impulsion rise time or few % of the sampling period are possible.

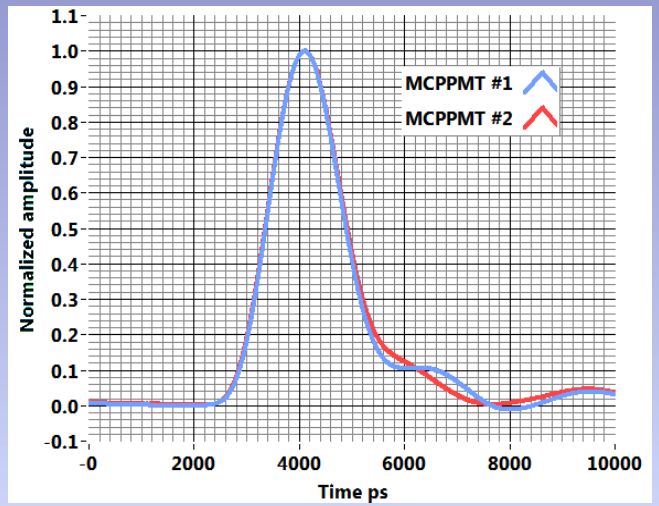
Many technics have been developed to extract timing from sampled data. Some of them (in **red**) all compatible with a reasonable integration in realtime digital electronics are now described and tested on the MCPPMT setup example

- Algorithms inspired from Analogue timing technics: .
  - **d-LED**
  - **Initial Slope**
  - **Interpolation techniques**
  - **d-CFD**
  - **d-ZCCFD**
- True Digital Algorithms.
  - **Optimal Filtering**
  - Deconvolution
  - **Least Square Minimization**
  - Use of neural networks

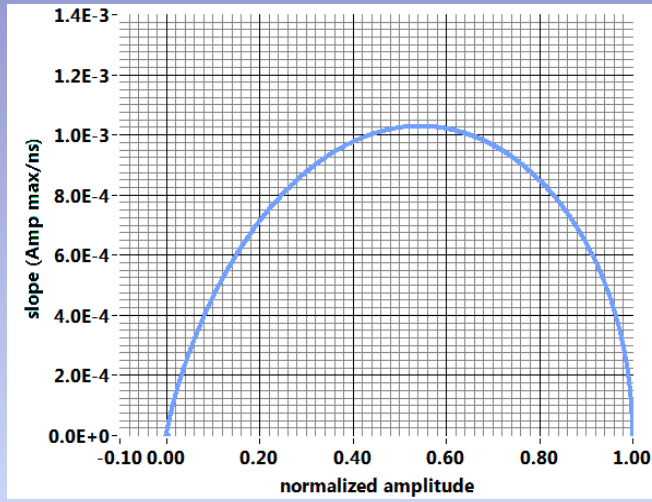
Few characteristics of the MCPPMT used to illustrate the various timing methods



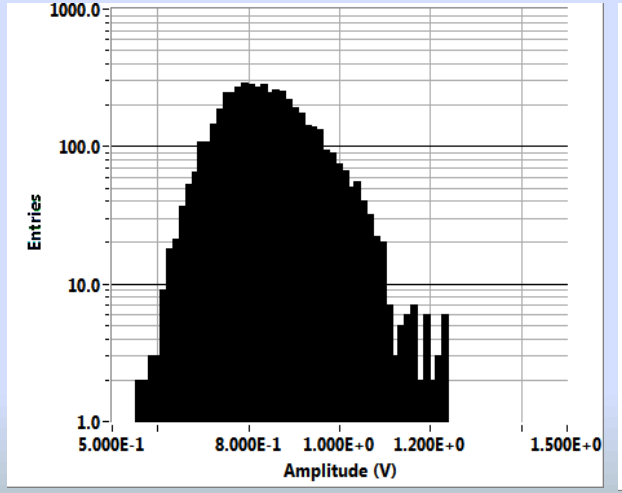
Typical Pulse



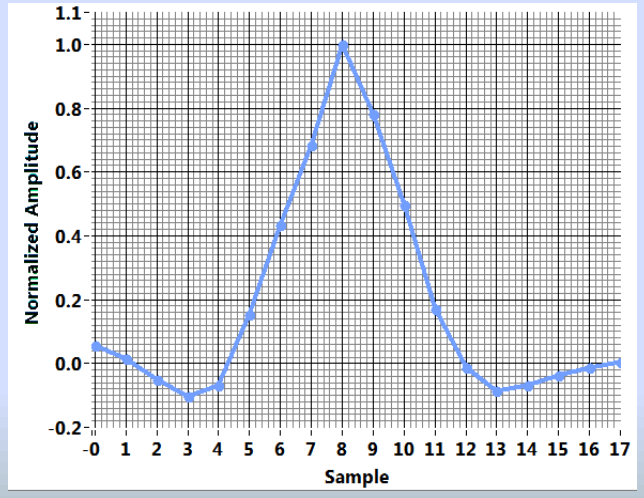
Average Pulses



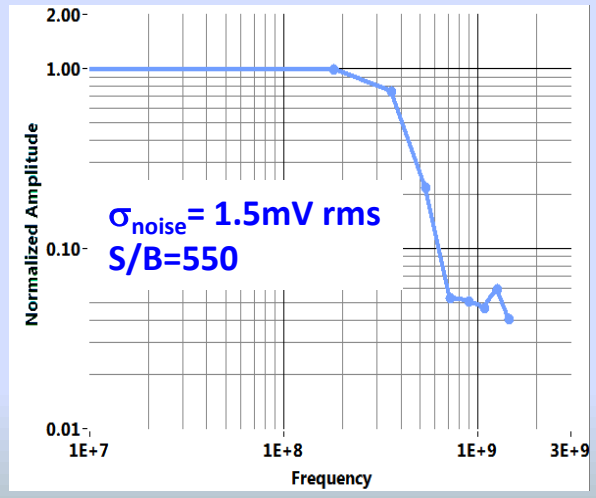
Pulse Derivative maximum
@ ~ half amplitude



Amplitude Distribution



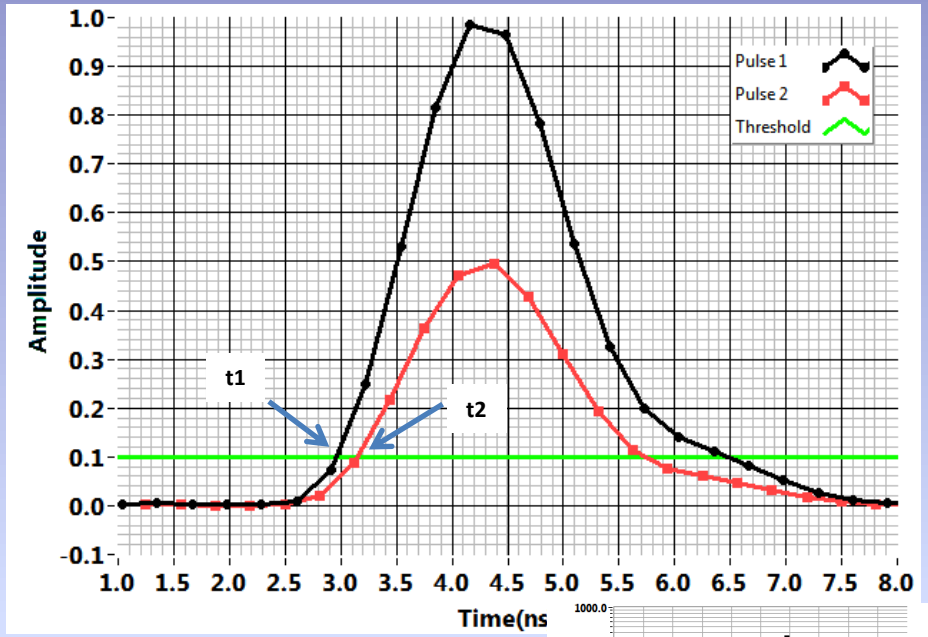
Noise auto-correlation Function:
Strong correlation over >6 samples



Corresponding noise spectrum
370 MHz BW

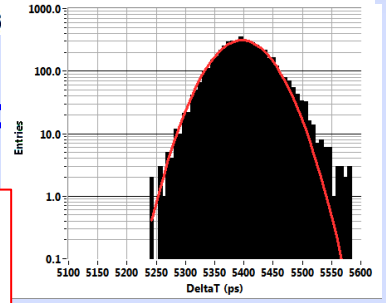
d-LED (Digital Leading Edge Discri)

- Emulation of the analogue leading edge discriminator.
- Time crossing of a fix threshold.



- Same limitations as Analogue LED : timewalk due to amplitude variation:
  t is a decreasing function of Amplitude

- Timewalk can be corrected with a calibrated Look Up Table using :
  - amplitude or charge measurement
  - Time over threshold

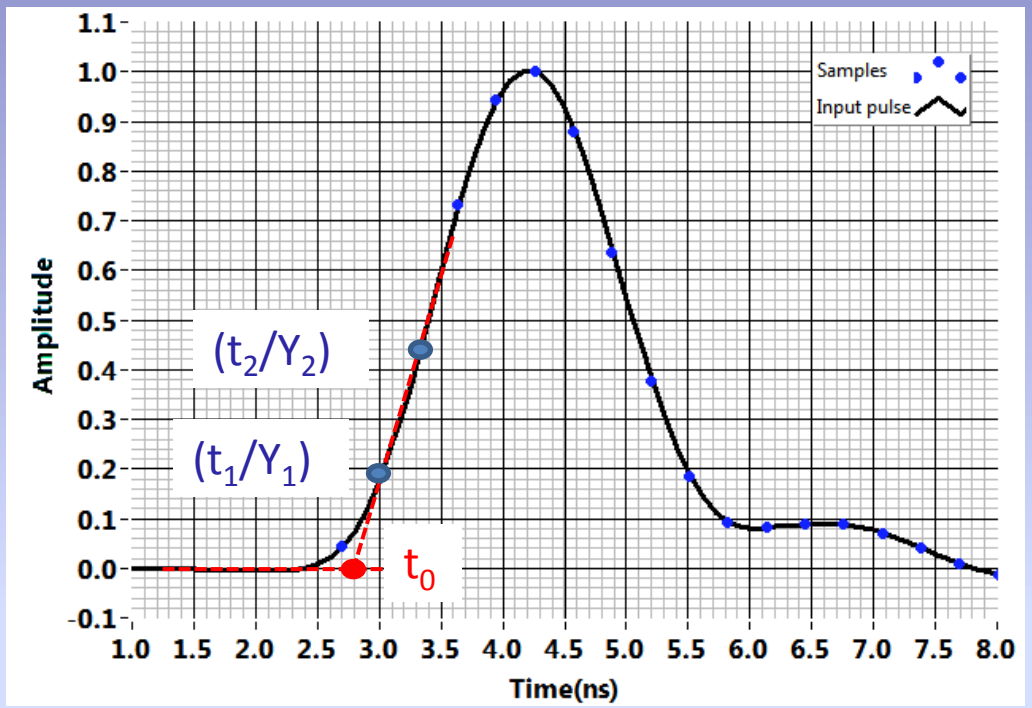


ΔT = 5.397ns/ σ_T =36 ps rms
: very lowTH= 50mV optimum

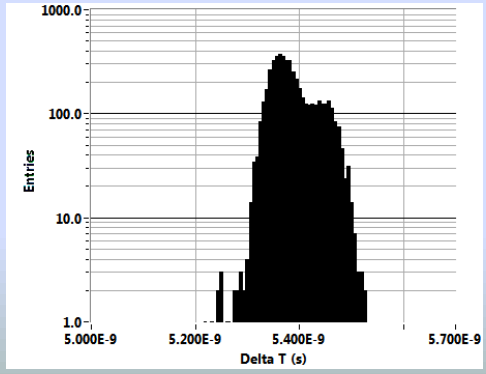
- Can be used only to detect the signal and give a rough timing before applying a more sophisticated algorithm
- In some cases (if very low thresholds are possible) can give good resolutions

ISA: Initial Slope Approximation

- Find the samples with the highest derivative = with the largest amplitude difference.
- Calculate the intersection of the line passing by these samples with the baseline:
- At first order, timewalk effect cancelled.
- Need enough samples on the rise time to catch the highest slope.
- Good resolution obtained with 3 samples on the rise time.



$t_0 = t_1 - (t_2-t_1).y_1/(y_2-y_1)$

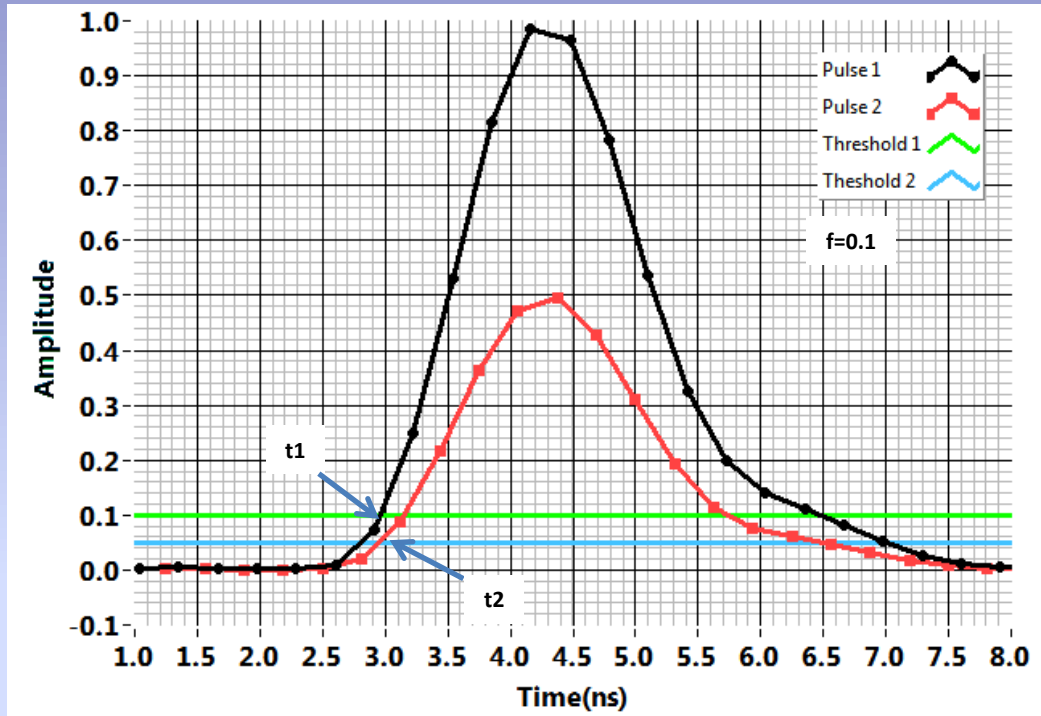


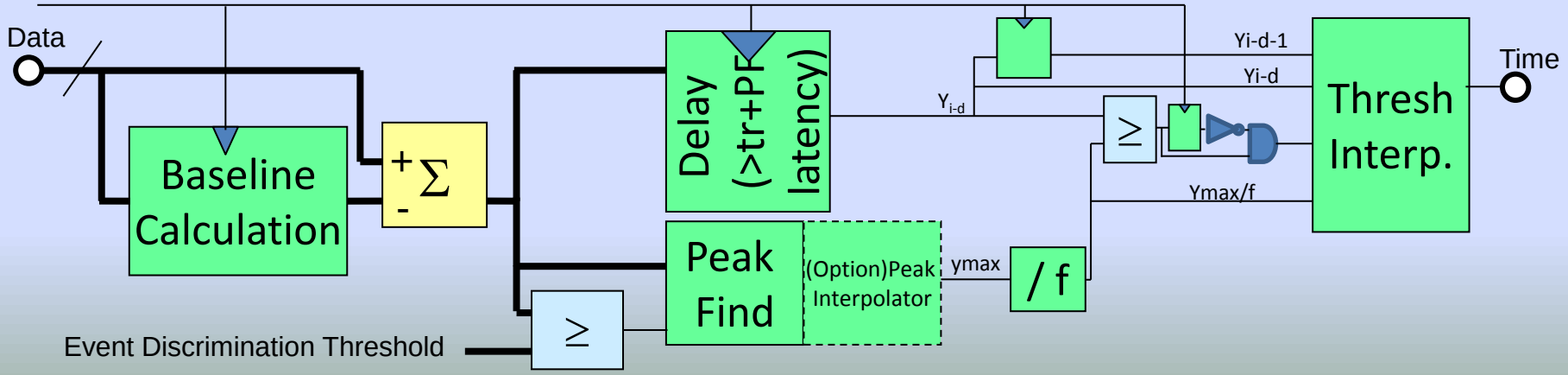
[Streun]: PET LSO + PMT : resolution < 600ps rms with 12 bits/ 40 MHz sampling rate

ΔT = 5.387ns/ σ_T = 30 ps rms optimum for Y_1 first sample above 150mV
Non gaussian distribution due to slope changes.

d-CFD (Digital Constant Fraction)

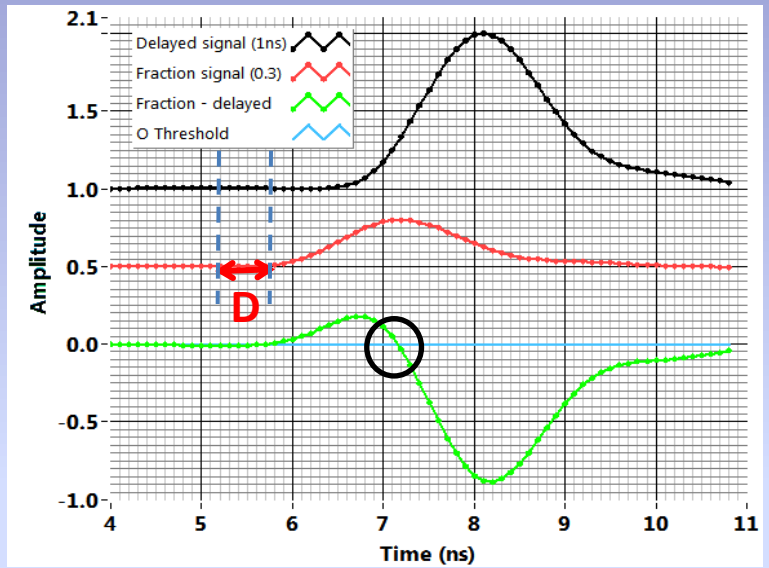
- Time crossing of a threshold set at to a fix fraction of amplitude (or Charge).
- If pulses are homothetic: timewalk is cancelled.
- Compatible with FPGA.
- Easier if f is a power of 2.





d-ZCFD Algorithm

- Emulation of the analogue ZCFD.
- Quite equivalent to CFD (but the threshold is a fraction of a sample not necessary = peak)
- Easier to implement in FPGA for RealTime process.
- No need for peak finding.
- Knowledge of t_{peak} required to tune the delay
- Several possible versions



Simplest expression :

$$V_{ZCFD}(k) = f.V(k) - V(k-D)$$

Typically D = pulse peak/rise time [Hennig], [Bardelli].

Peak estimated through the sliding sum of samples :

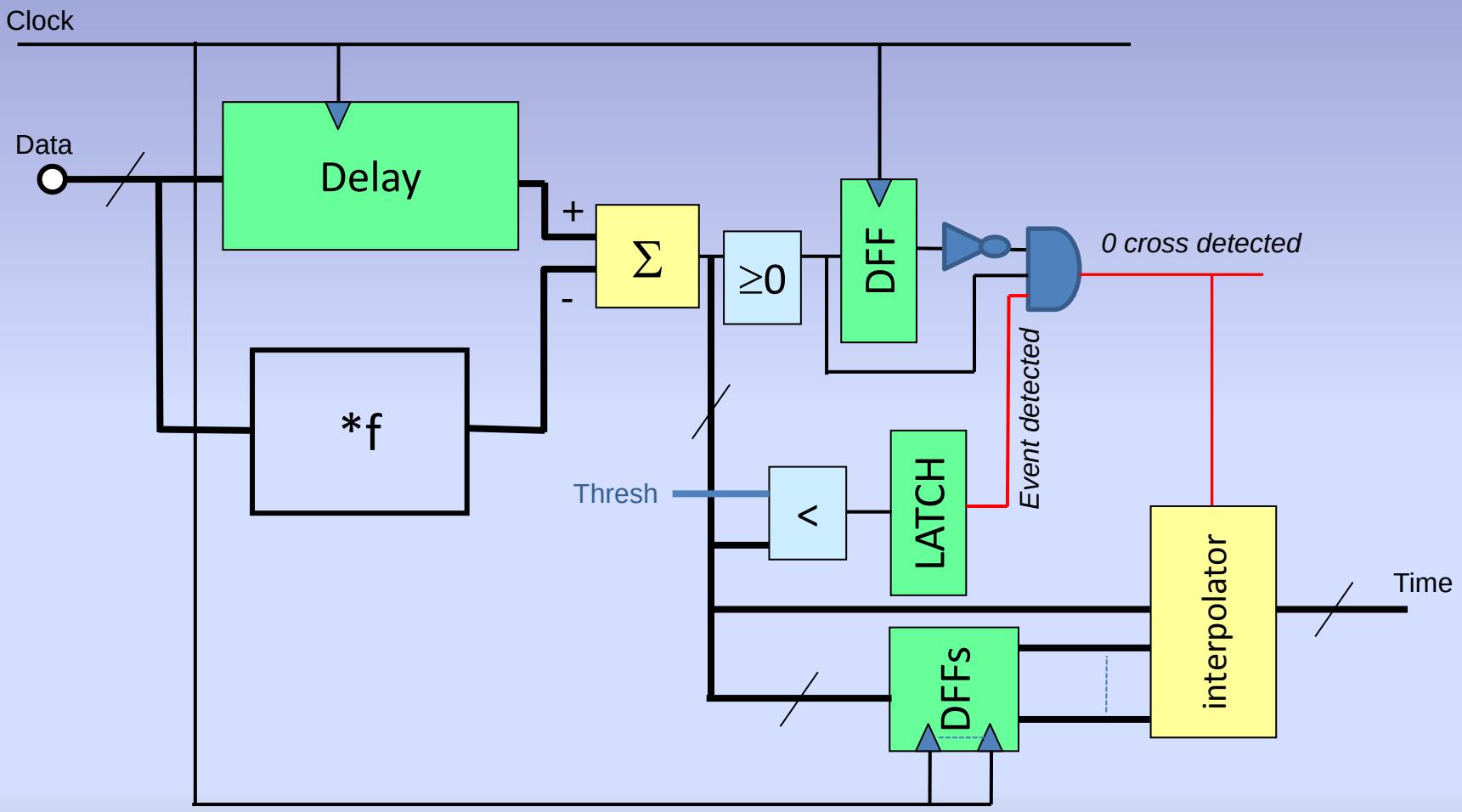
$$V_{ZCFD}(k) = f.V(k) - \sum_{i=1}^{L} V(k-i-D)$$ => Peak is estimated from charge

Both Crossing & Peak estimated through the sliding sum of samples :

$$V_{ZCFD}(k) = \sum_{i=1}^{L} f.V(k) - V(k-i-D)$$ [Fallu-Labruyere] => Q-dZCFD

If D < peak Time => Emulation of ARC (amplitude & risetime compensated) CFD
=> compensate a dependency of the detector signal risetime with amplitude (CdTe) [Nakhostin]

d-ZCFD Algorithm in a FPGA

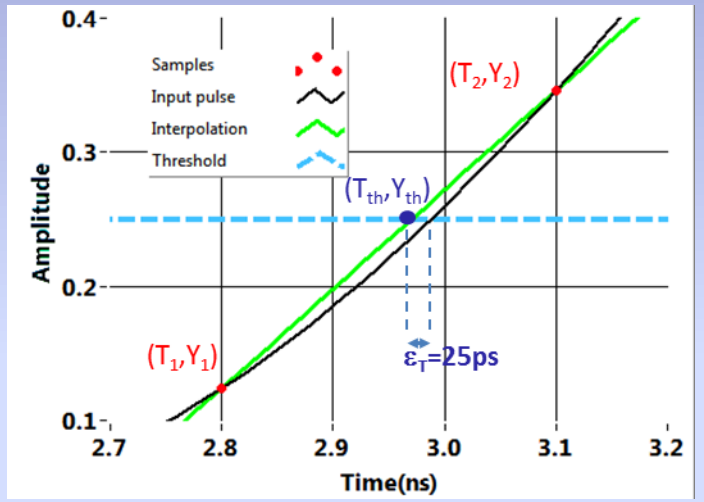


Threshold crossing time pick-off

Without extra calculation, undersampling limits the precision of timing (to Ts/√12) and of amplitude.

Timing can be Improved by using linear interpolation between samples.

$$T_{th} = T_1 + (Y_{th} - Y_1) . (T_2 - T_1) / (Y_2 - Y_1)$$



Can be integrated easily in FPGA or DSP

Interpolation error ε_T due to the waveform curvature depending on the phase of the samples with the threshold

=> produce non gaussian time spectrum.

Possible Solutions:

- Filter the input signal to have more samples on the edge.
- Increase then number of samples => increase the Sampling frequency.
  => trade-off between cost put in extra sampling and cost due to extra digital treatment.
- Calculate new samples:
  - using polynomial interpolation.
  - With digital filter: Nyquist-Shannon-Whittaker theorem

Polynomial interpolations

- Calculate the Lagrange polynomial passing through n+1 samples in the area of interest.

$$Ln(t) = \sum_{i=0}^{n} y_i \,.\, b_i(t) \quad with \qquad b_i(t) = \frac{\prod_{j=0, j\neq i}^{n} (t - t_j)}{\prod_{j=0, j\neq i}^{n} (t_i - t_j)}$$

Easy to code in software. Degree 2 or 3 interpolation compatible with implementation with DSP and (more hardly in modern FPGA).

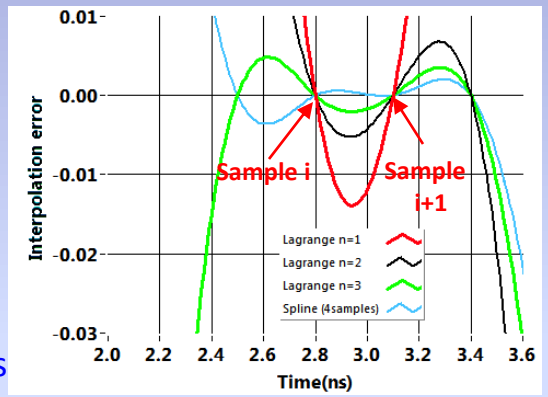
Degree 2: $L_2(t) = a.t^2 + b.t + c$ can be accurate enough :

- for peak finding (parabolic approximation)

  => calculate the a & b coefficients => $y_{max} = c - b^2/2a$

- for threshold crossing if no "flex" of the signal in the area of interest.

Degree 3: implemented by [Bardelli] on a ADSP2189N using very limited resources



1st order => Spline or 3 order interpolation
⇒ Error decreased by 10 or more
(ε_T max decreased from 25ps to less than2)

- Use cubic spline interpolation: set of 3rd order polynomials:
  - each passing through 2 consecutives samples of interest, with continuous first and 2nd order derivatives.
  - Solve N+1 equations with N+1 unknowns.

Successfully implemented by [Semmaoui] using TMS320C6414 DSP

⇒ Two ways to find the threshold crossing after interpolations:

- ⇒ Calculate all the interpolated samples between the two samples across the threshold then use a sequential algorithm similar to the one for zero crossing. (testing all the interpolated samples one after the others).
- ⇒ Solve the f(t)=Th equation by an iterative method (Newton, dichotomy) using the interpolated samples.

A linear approximation using the 2 interpolated samples closet from the threshold can be used to improved again the timing.

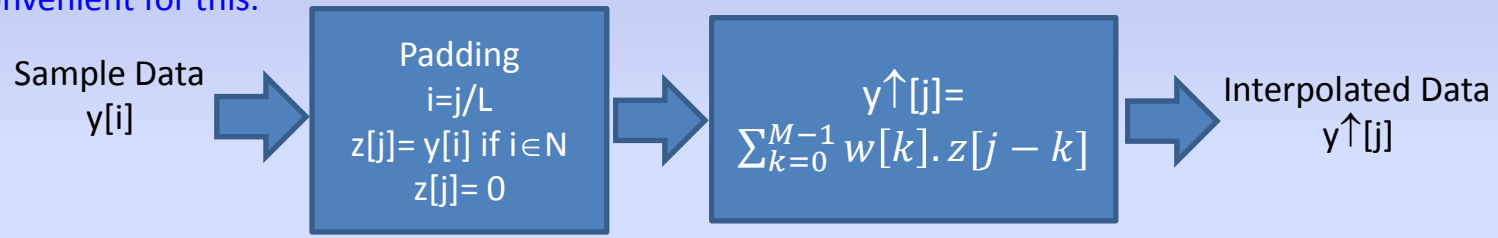
Low Pass Filter Interpolation

Nyquist-Shanon theorem says:" *It is possible to recover a continuous signal from obtained sampled signal if the sampling frequency is twice the signal bandwidth* ".

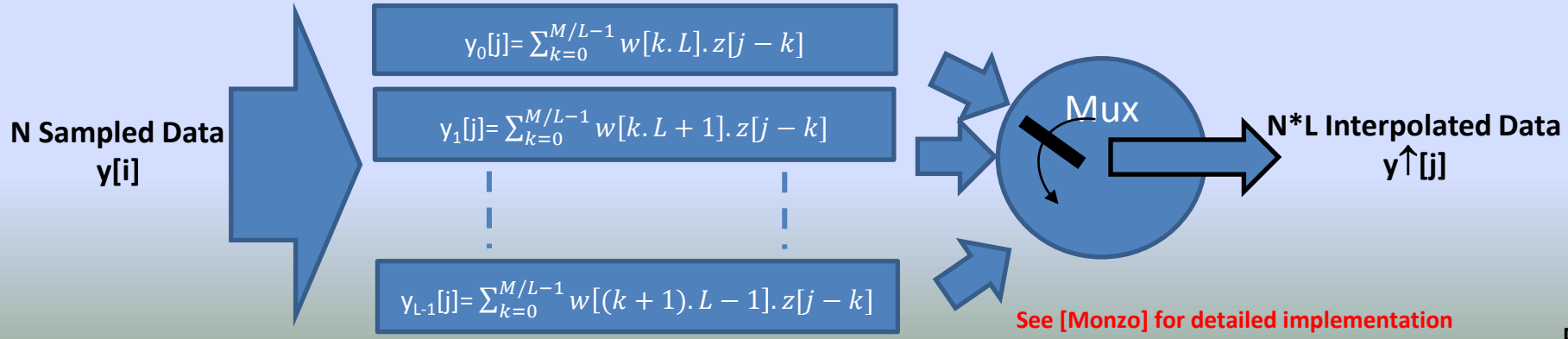
One well know method is the **low pass filter interpolation** [Fontaine], [Monzo]:

For a L interpolation factor:

- The signal is padded with L-1 "0" between each sample.
- A low pass-filter with cut-off frequency <= Fs/2 is then applied to cut the image of the signal created in the higher frequency: a Low Pass Windowed FIR filter, with M w[i] coefficients (easy to implement in FPGA) is convenient for this.

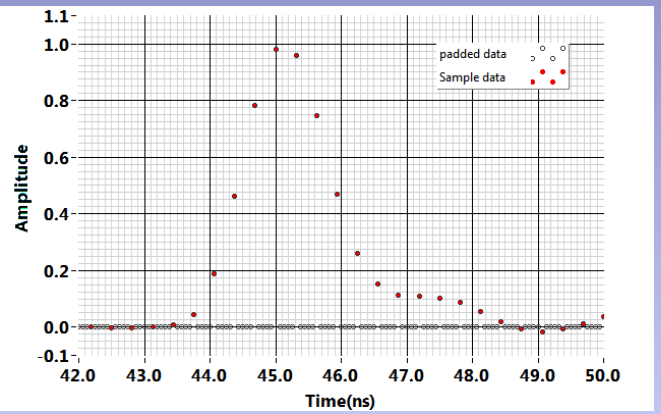


- Special structure of the padded data allows the use of L **polyphase** filters with M/L coefficients working in // at the incoming rate rather than a high frequency one with all coefficients [Bose]. Easier for real time.

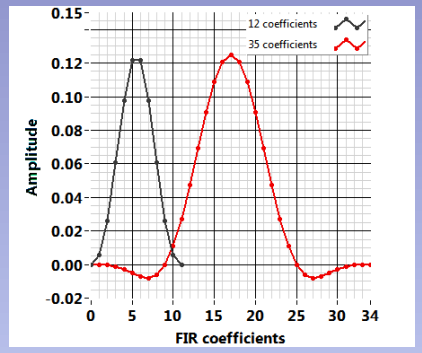


See [Monzo] for detailed implementation In Xilinx Virtex5 with Fs=70MHz

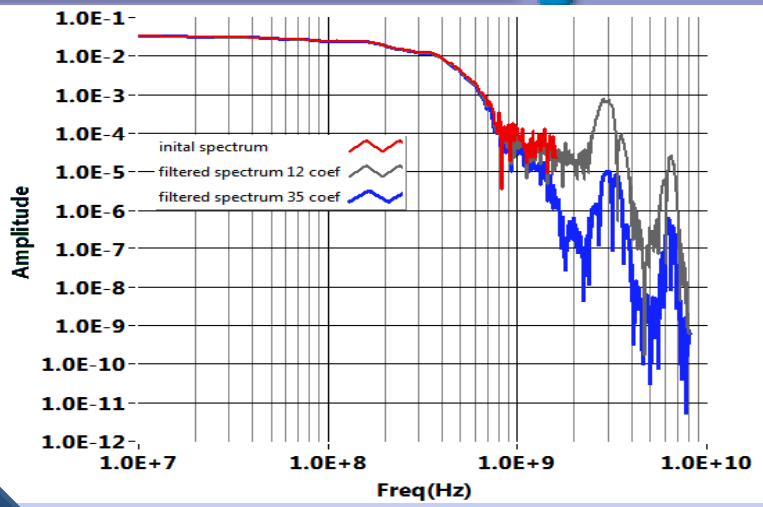
A practical exemple: interpolation by a factor 5



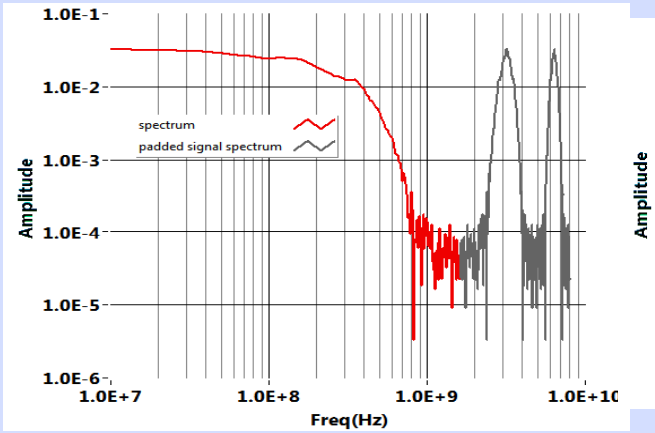
Original pulse: FWHM 1.5ns Fs=3.2GSPS



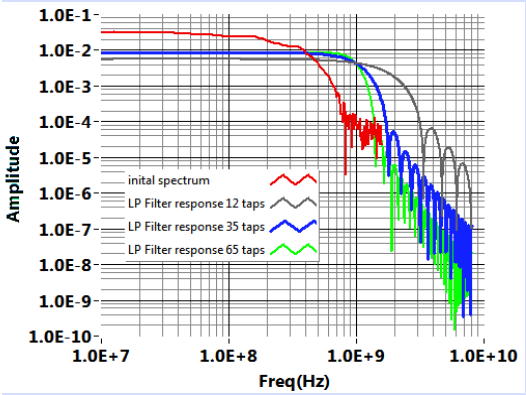
Low pass (Hanning, cutoff 1GHz) windowed filter coefficients



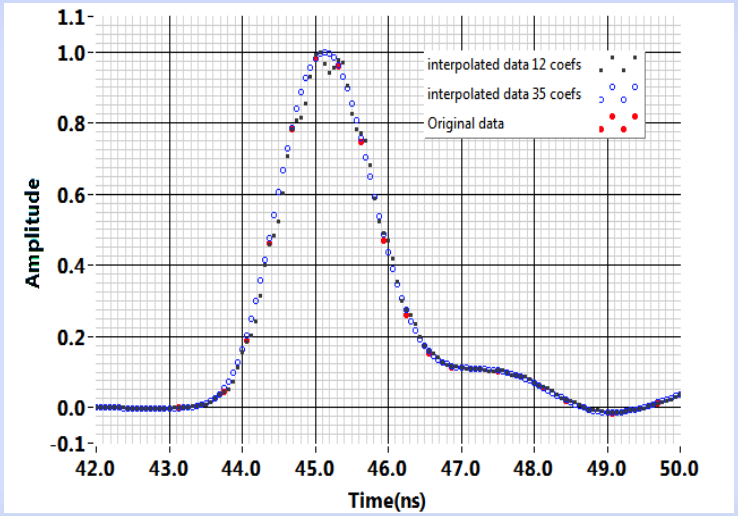
Interpolated signal spectrum:
Check aliasing, check the signal BW



Signal BW= 450 MHz. 120 dB/dec
(SCA input is very good anti-alias filter)

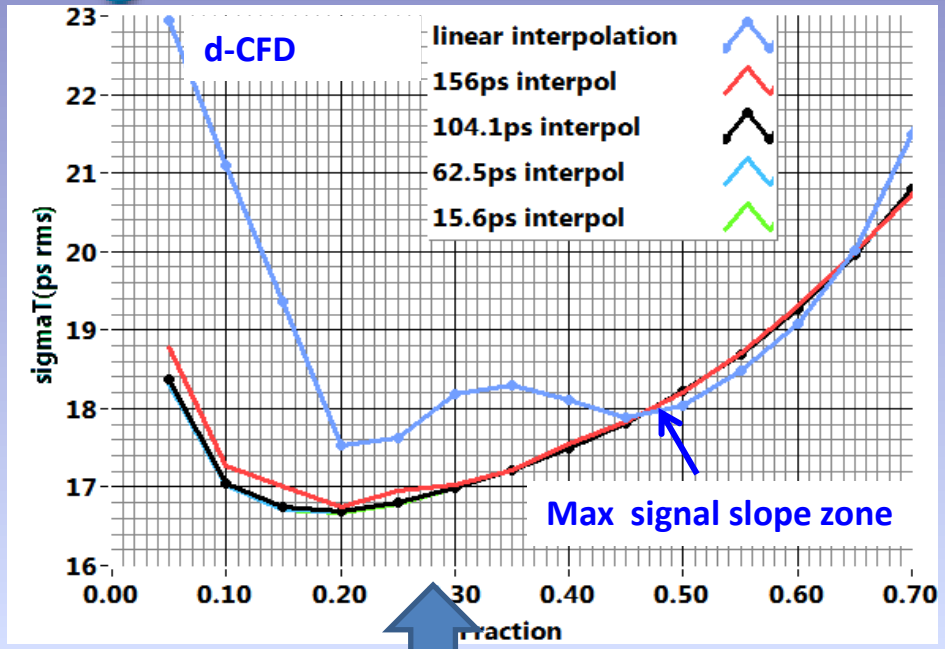


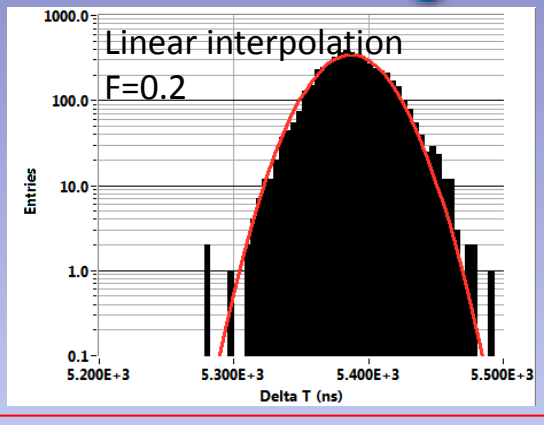
FIR frequency response:
More coef => higher HF rejection
Lower cutoff frequency => more coef
Larger interpolation factor => more coef



Interpolated pulse:
=> Effect of bad HF filtering when the number of coef. Is too low

CFD-ZCFD: results





d-CFD

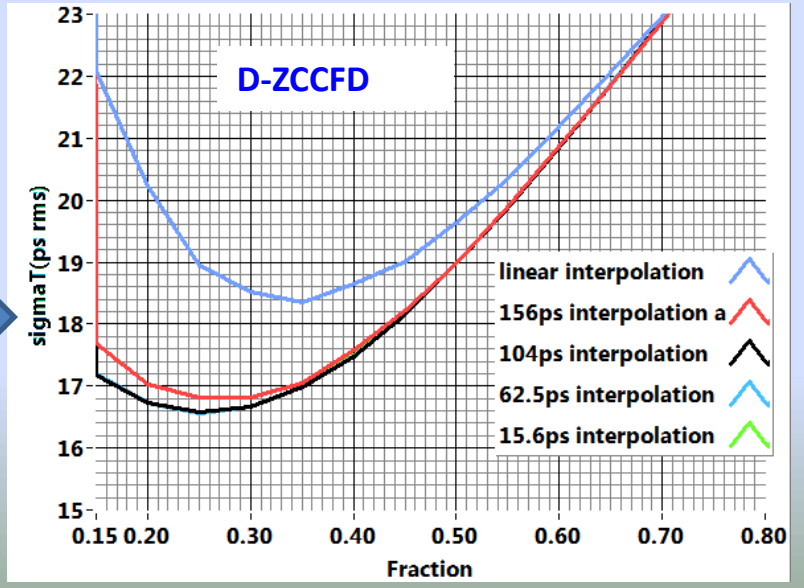
$\Delta T = 5.387ns / \sigma_{T\ opt} = 16.6$ ps rms $N_{ov} \geq 2$

$\Delta T = 5.385ns / \sigma_{T\ opt} = 17.5$ ps rms with linear interpolation

d-CFD: resolution vs fraction with varying interpolation factor:

- Results plotted here for Lagrange 3rd order interpolation
- Exactly the same results with spline interpolation or digital filter (tested up to Nov=5)
- Optimum curve already reached for Nov between 2 and 3.
- Best reolution obtained for F= 0.2

Equivalent results with **d-ZCCFD with D=3 (= peak time)**

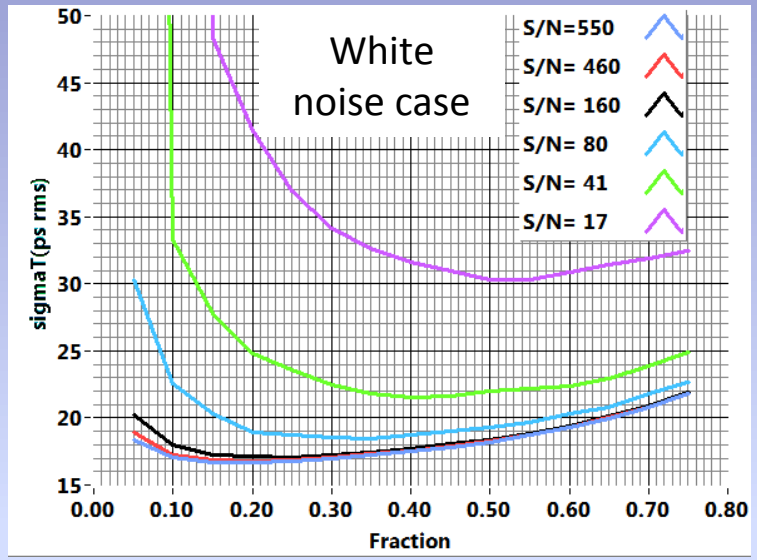


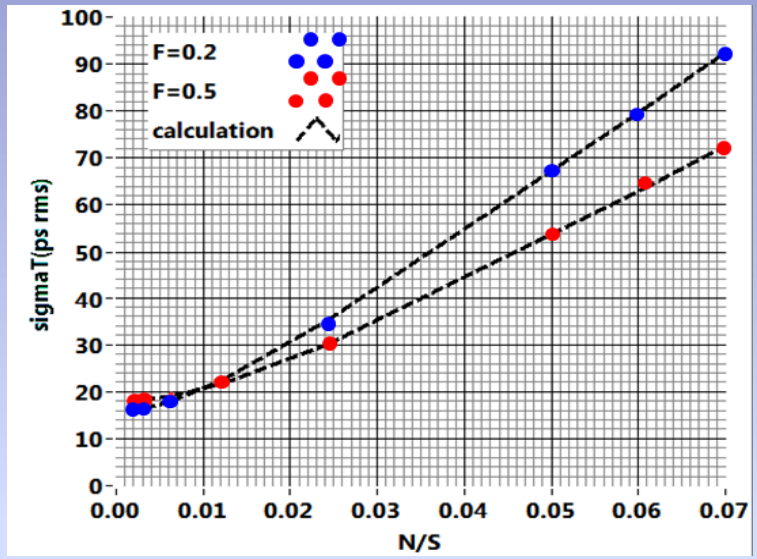
The best timing is obtained at the very beginning of the signal and not at the max slope

Resolution is detector limited

CFD: noise dependency

Noise has been added to the data to check the noise sensitivity





- First degradation appears when noise x 3-5 => resolution is detector dominated
- Optimum fraction progressively move towards the highest slope region when noise increase

- Data consistent with the model:

$$\sigma_T{}^2 = \sigma_0{}^2 + \left[\frac{N}{S} \cdot \left(\frac{dA}{dt}(F)\right)^{-1}\right]^2$$

10% worst if the added noise is only in the signal BW (<300MHz) case of pure white noise. True for all the interpolation modes & also for ZC-CFD

Pulse recognition methods

A reference pulse is computed (can be offline) :

-Using real data, interpolated realigned and normalized (in A or Q) and averaged.

-Or from theoretical response.

A zone of interest of the reference pulse Ref(i) (eventually oversampled by a factor N_{ov}: $t=T/N_{ov}$) is kept.

The pulse is detected, normalized and the only the M samples Vn(j) of a zone of interest are kept . The time of the first sample of this zone (Tz) gives a coarse timing:

Principle: Find the start time for the reference pulse to match the measured one:

- Brut force fit of the data => requires a lot of computing power [Leroux]
- Use of LUT [Haselman]:
- Time shift LSM [Leroux],[Breton]…

Look Up Table

- The Reference pulse REF= REF[i] is inverted, interpolated and stored in a LUT:

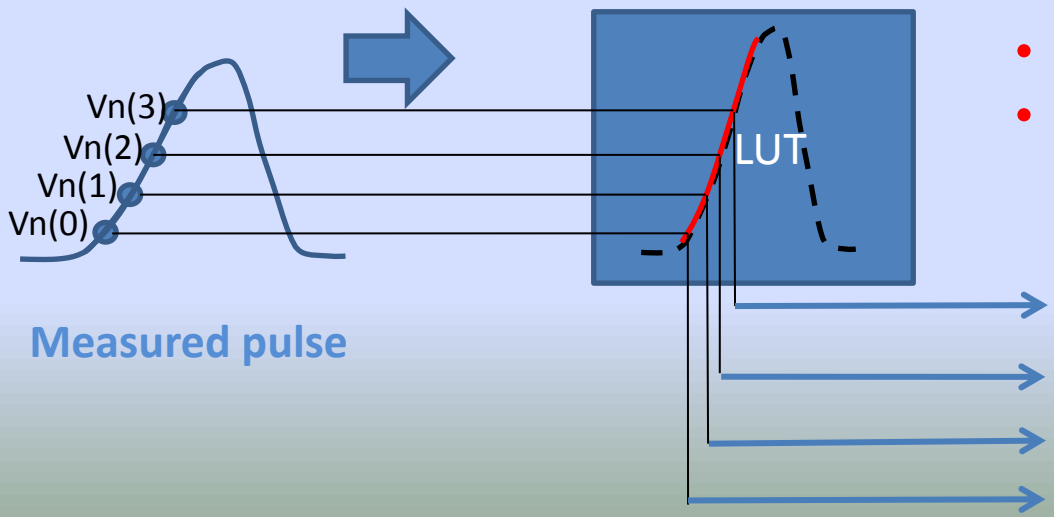
  $T_{LUT} = f^{-1}(REF)$ (TLUT resolution is better than T)

- the first measured sample (normalized) is sent to the LUT

  => the global timing is given by: $T = T_z + T_{LUT}(S_n(0))$

- It can be generalized by using K samples to refine the measurements. In this case the timing is averaged:

$$T = T_z + \frac{\sum_{k=0}^{K-1} T_{LUT}(S_n(k)) - k.T)}{K}$$

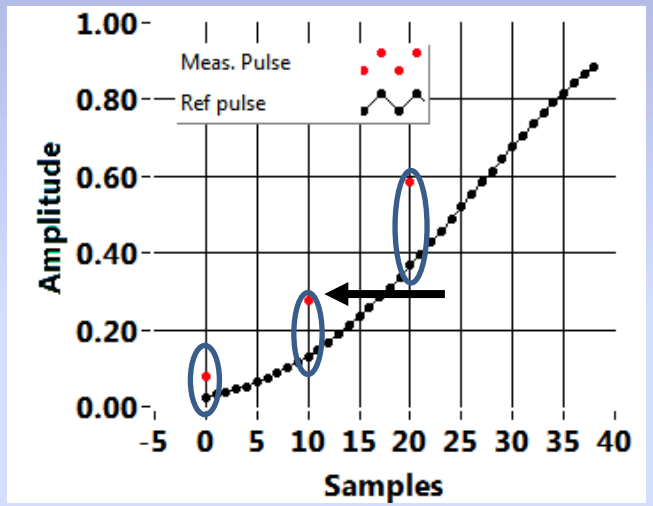


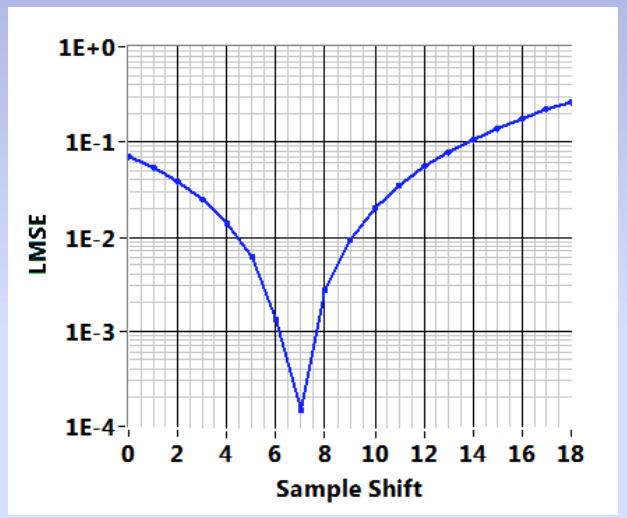
- **FPGA compatible**
- **Fast.**
- **Requires only limited resources**

Time-Shift Least Mean Square Error

- The timing is obtained by minimizing the Least Mean Square Difference between the normalized measured pulse and the reference pulse progressively shifted:

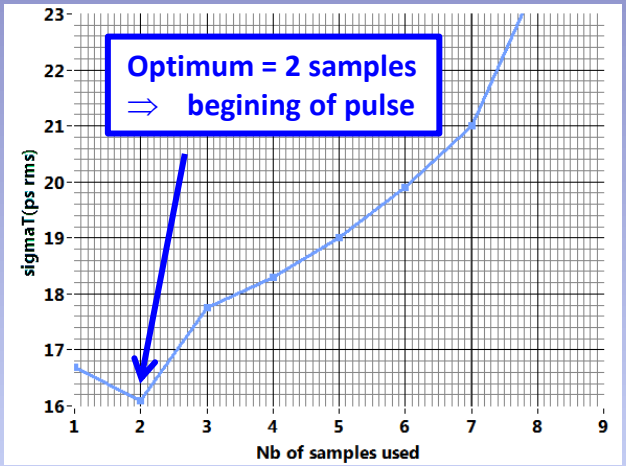
$$LMSE(j) = \sum_{i=0}^{M-1} \left(S[i] - Ref(Nov.i + j)^2\right)$$





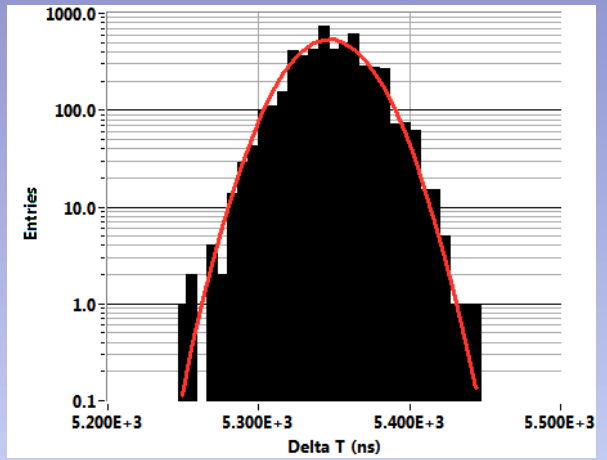
- At least ~ 2*N_{ov} operations required => calculation time.
- The real LMSE minimum can eventually be interpolated from LMSE(j) for a better precision.
- No need for large computing resources. Compatible with FPGA.

LMSE: results

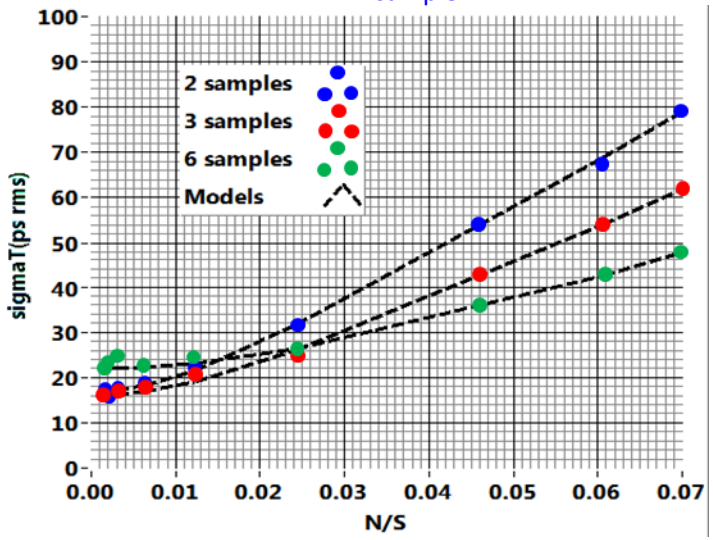


Sliding LMSE
Reference pulse with 10ps step

ΔT = 5.377ns
$\sigma_{T\,opt}$ = 16.1 ps rms

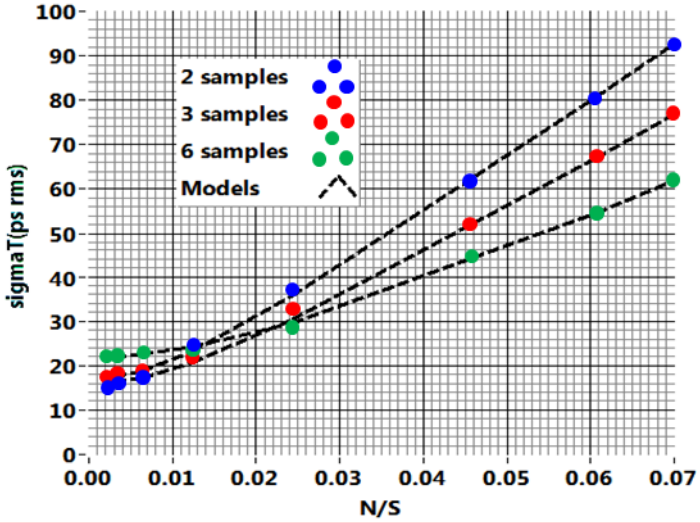


With white noise added: less sensitive than CFD.
Noise averaging when N_{sample} increases



$$\sigma_T{}^2 = \sigma_0(Nsample)^2 + [\frac{N}{S} \cdot \frac{1}{\sqrt{N_{sample}}} \cdot (< \frac{dA}{dt} >)^{-1}]^2$$

With noise added only in the signal BW (<250MHz):
less sensitive than CFD. Same effect but less effective (noise correlation between samples)



$$\sigma_T{}^2 \approx \sigma_0(Nsample)^2 + [\frac{N}{S} \cdot (Nsample)^{-1/5} \cdot (< \frac{dA}{dt} >)^{-1}]^2$$

Few properties of pulse recognition

- Several samples from the waveform are used => improve the noise rejection capability.

- Requires good definition of the Reference pulse and of the zone of interest for timing:
  * samples containing timing information.
  * zone of interest must be reproducible from pulse to pulse.
- Quality of the pulse renormalization affects the results.

- Even in the ZOI, the amount of timing information "associated" for each sample is not uniform:

  => not taken into account in the previous algorithm.

  => The samples should be weighted in LMS or in the calculation using LUT.

"Optimal" digital Filter

- **Widely used in HEP: NA48, ATLAS Calorimeters** [Cleland] with sub ns-resolution @ 40MSPS.
- Evaluated for PET application in [Joly] and compared to dCFD.

- Principle :
  * Find A and tf to make the sampled signal S[i] match as much as possible A.Ref(t_i-t_f)
  * A and tf are calculated by applying a FIR with very few (optimized) coefficients to the signal:

$$u = \sum_i a_i \,.\, Ref[i] = [\mathrm{a}]^{\mathrm{T}}[Ref] \equiv A,$$
$$v = \sum_i b_i \,.\, Ref[i] = [\mathrm{b}]^{\mathrm{T}}[Ref] \equiv \mathrm{A}.\, t_f \Rightarrow\ t_f = v/u$$

- Method described step by step in [Cleland], based on:

  *** signal linearization:**

  S[i]= A.s(t_i-t_f) =A.Ref[i] – A.t_f.Ref'[i] + n[i] where n[i] are the noise contributions to samples.

  * The search of [*a*] and [*b*] **minimizing the variances** of *u* and *v* knowing **the noise auto-correlation matrix (or function) [R_{ij}]** (related its frequency spectrum).

  var(*u*) = [*a*]^T [R_{ij}] [*a*], var(*v*) = [*b*]^T [Rij] [*b*] : Several possible methods (Lagrange multipliers, conjugate gradient)

  Advantages:

  => Naturally gives weight to the samples according to the signal shape.

  => Use the information from several samples and not only 2 samples : good tolerance to noise.

  => Take the noise spectrum into account to calculate the coefficients of the filter.

  => **FIR is straightforard to implement on FPGA or DSP.**

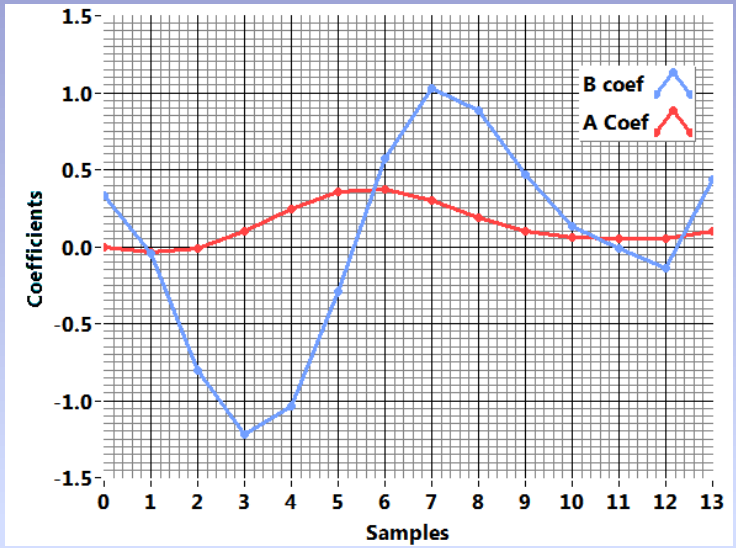
  Practically:

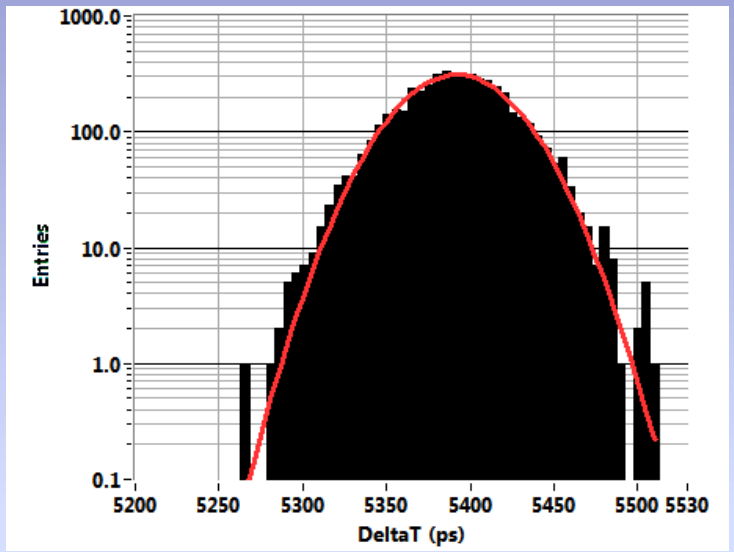
  => the method relies on linearization, A and t_f estimations are good for low tf but systematic bias on when $|t_f|$ increases

  => solutions: use several set of coefficients with Ref signal shifted by a fraction of the sampling clock or calibration and correction.

“Optimal” digital Filter: results

FIR coefficients





Optimum with >7 coefficients for each of the 2 FIR (very small improvement from 7 to 14) :

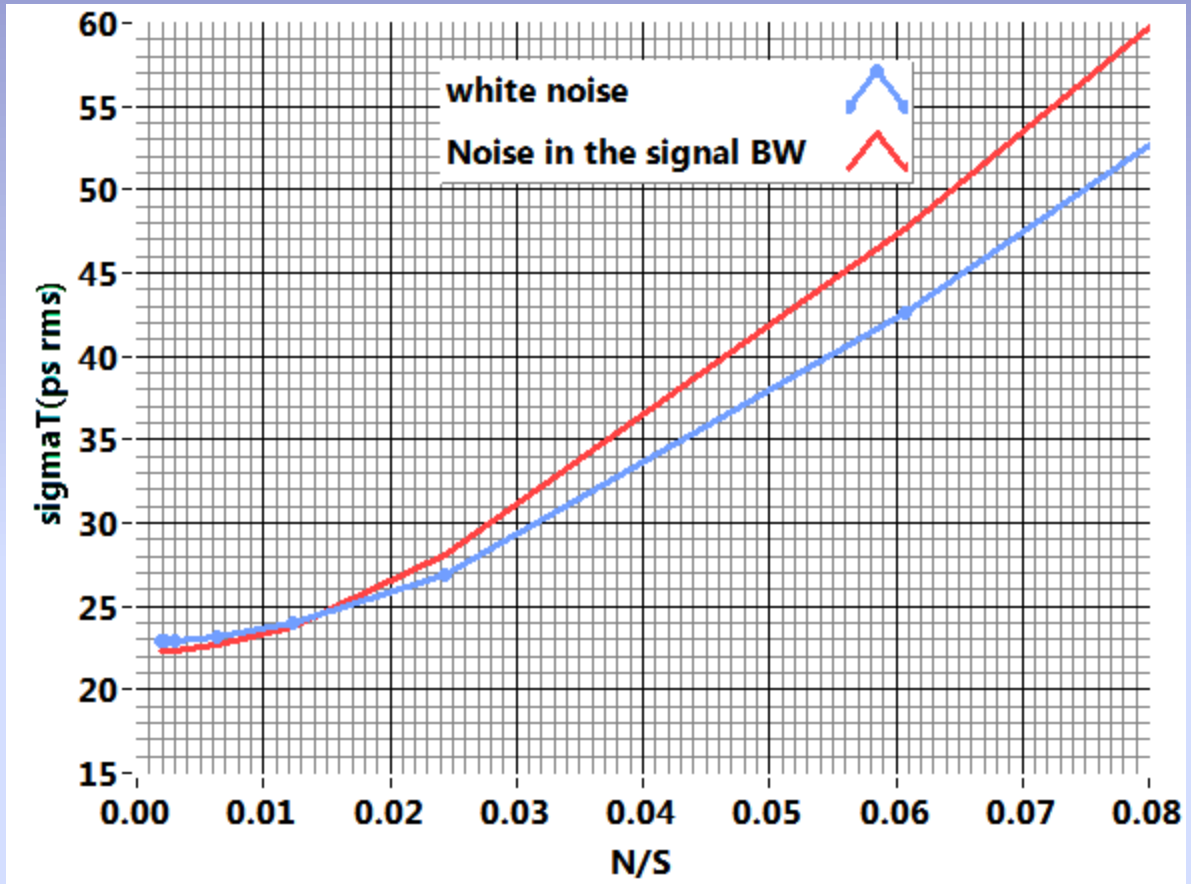
- the signal max must be in the calculation
- must be larger than the “duration” of the noise autocorrelation function

$\Delta T = 5.389ns / \sigma_{T\,opt} = 24$ ps rms

decreased to 22ps rms if 3 sets of coefficients (corresponding to 3 ranges of t_f) are applied

=> Worst than CFD (because variations of the signal after mid amplitude)

“Optimal” digital Filter: behavior with added noise



Optimal filter recalculated each time (7 coefficients)

Results slightly better than those with LMSE method for large N/S.

Thank you for your attention !

References: ultra-fast SCAs

- **ATWD, ATWR (S. Kleinfelder)**
  - ATWR: *S. Kleinfelder's M.S. thesis, Univ. California, Berkeley 1992*
  - ATWD: *IEEE TNS 50-4:955-962 ,2003*

A didactic paper by G.Haller et al:
IEEE JSSC 29-4 (1994)500-508

- **PSI developments (DRS family)**
  - *IEEE/NSS 2008, TIPP09*
  - http://midas.psi.ch/drs
- **Orsay/Saclay**
  - ARS: *IEEE TNS 49-3:1122-1129,2002*
  - MATACQ: *IEEE TNS 52-6:2853-2860,2005 / Patent WO022315*
  - SAM: *NIM A567 (2006) 21-26, IEEE/NSS 2006, NIM A 629 (2011) 123-132, NDIP 2011(Lyon)*
  - NECTAR: *NDIP 2011(Lyon),* ***IEEE/NSS 2011 (N28-6)***
- **Hawai'i developments**
  - STRAW: *Proc. SPIE 4858-31, 2003*
  - LABRADOR: *NIM A583 (2007) 447-460.*
  - BLAB: *NIM A591 (2008) 534-545; NIM A602 (2009) 438-445.*
  - STURM: *EPAC08-TUOCM02, June, 2008.*
  - *Calibration:* Nishimura et al, *Physics Procedia (TIPP 2011)*
- **Chicago activities**
  - http://psec.uchicago.edu
  - Ps timing: *NIM AA607:387-393,2009*
  - PSEC 3: **IEEE/NSS 2011 (NP2S-75),** *TIPP 2011*

References: Time picking methods

- [Abbiati]: Abbiati et al., IEEE TNS (Jun 2006), Vol. 53, NO. 3, p 1270-1274
- [Bardelli]: Bardelli et al., NIM A 521 (2004) 480-492
- [Bose]: T. Bose ,Digital Signal and Image Processing, Wiley (2004) p319-332
- [Breton]: D. Breton et al, NIM A 629 (2011), p123
- [Fallu-Labruyere]: Fall-Labruyere et al., *NIM A* 579 (2007) , p.247
- [Fontaine]: Fontaine et al., IEEE TNS (Feb 2008), VOL.55, NO. 1
- [Haselman]: Haselman et al., IEEE NSS Conference Record
- [Hennig]: Hennig et al. IEEE TNS (Aug. 2010), VOL. 57, NO. 4
- [Joly]: Joly et al, IEEE TNS (Feb 2010),Vol. 57, NO.1
- [Leroux]: Leroux et al., IEEE TNS (Oct 2009), VOL. 56, NO. 5
- [Monzo]: Monzo et al, IEEE TNS (Aug 2011), Vol. 58, NO. 4
- [Nakhostin] : Nakhostin et al, NIM A 614 (2010), p308
- [Semmaoui]: Semmaoui et al., IEEE TNS (June 2009), VOL. 56, NO. 3
- [Streun]: Streun et al., NIM A 487 (2022) 530-534

Some results from the Reference papers

				Source	Signal rise time	σ(ps rms)
Fallu Labruyere	ZCFD. Linear interpol	75MSPS 14 bits	LaBr3 +XP2020	^{22}Na		173
Hennig	CFD	500 MSPS 12 bits	LaBr3+ XP20D0	^{60}Co		177
Bardelli	CFD & ZCFD Cubic interpol	100 MSPS 12 bits	Silicon	Heavy ions	80ns	53
Fontaine	CFD linear + filter interpol	45 MSPS 8 bits	LYSO+ APD	^{68}Ge	~100ns	1796 (linear) 1640 (filter)
Semmaoui	Deconvolution + Adaptative filter	45 MSPS 8 bits	LYSO+LGSO+APD		40ns,65ns	1350 (LYSO) 2470 (LGSO)
Leroux						
Monzo	LPF filter+ Q-ZCFD	70 MSPS 12-bits	LSO + H8500	^{22}Na	45ns ?	545
Streun	initial slope interpolation	45 MSPS 12 bits	LSO+ PMT	^{68}Ge	75ns	600
Nakhostin	ARC-CFD	250-100GSPS	CdTe Schottky	^{22}Na	75ns	5658
Joly	DCFD-OF1-OF2	5GSPS-8b 250GSPS-8b	LaBr3 + XP20D0 LYSO + PAD	^{22}Na	2ns	73-87-61 557-880-536
Breton, Va'vra	DCFD - LMSE	3.2GSPS-12b	MCP-PMT	Laser	1.5ns	~16