Digital Positron Lifetime:
The Influence of Noise

Arnold Krille[1] Wolfgang Anwand[2] Reinhard Krause-Reberg[1]

[1]Department of Physics, Martin-Luther-University Halle-Wittenberg, 06108 Halle, Germany

[2]Institut of Ion Beam Physics, Research Center Dresden-Rossendorf, 01314 Dresden, Germany

Positron Studies of Defects – September 1st to 5th, 2008

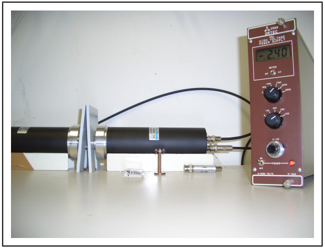
What is this talk about?

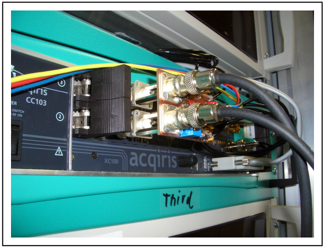
1. What is this talk about?
2. Making your own Artificial Pulses
   - Where we give a recipe to simulate digital positron lifetime pulses similar to the ones acquired from digitizing anode pulses from a photomultiplier.
3. Simulating a 4-GS/s Digitizer
   - Where we are trying to describe and simulate the reality of 4-GS–8-bit–digitizers and predict the performance of 4-GS–10-bit–digitizers.
4. Applying a Lowpass Filter
   - How applying a Butterworth filter (taken from literature) seems to improve the timing resolution by a factor of 2.
5. What did he say?

Analog vs. Digital

The task

Replace all the (50+ years old) analog electronics with PC, digitizer and mathematics.





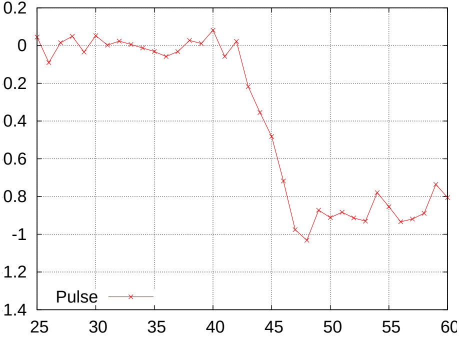
Benefits of digital processing:

- \+ Cheaper (More money left for conferences)
- \+ Simplier (Less cables cluttering the lab)
- \+ Better time base (No more calibration)
- \+ Easy to extent/change (More papers to be written)
- \- Less knowledge available
- ? Better timing resolution

Category “Doing your Homework”

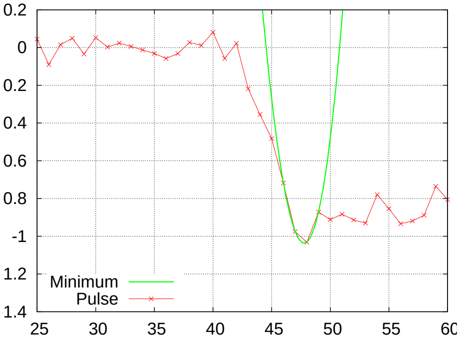
- Result of last years SLOPOS11
- Own “real” results are encouraging but not yet optimal
- Simulations needed:
    - to understand what is really going on
    - to see if other digitizers would give better results

The Algorithm to Extract the Timing Information



1. Find and interpolate the extremum[1]
2. Determine the zeroline (and its deviation) before the extremum
3. Interpolate the constant fraction point on the rising slope between zeroline and extremum[1]
4. Lifetime = $t_{Channel\ 1} - t_{Channel\ 2}$

The Algorithm to Extract the Timing Information

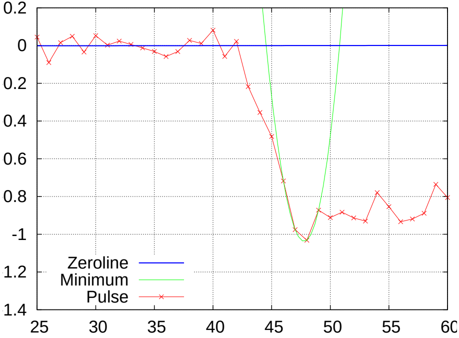


1. Find and interpolate the extremum[1]
2. Determine the zeroline (and its deviation) before the extremum
3. Interpolate the constant fraction point on the rising slope between zeroline and extremum[1]
4. Lifetime $= t_{Channel\ 1} - t_{Channel\ 2}$

---

[1] By simple polynom interpolation of 3rd order.

The Algorithm to Extract the Timing Information

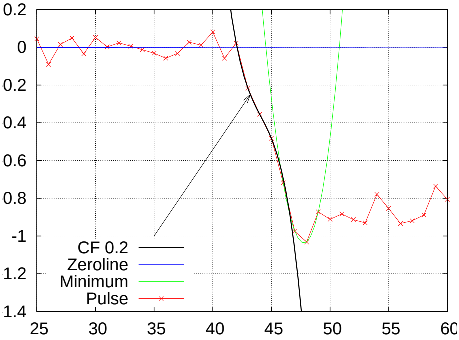


1. Find and interpolate the extremum[1]
2. Determine the zeroline (and its deviation) before the extremum
3. Interpolate the constant fraction point on the rising slope between zeroline and extremum[1]
4. Lifetime = $t_{Channel\ 1} - t_{Channel\ 2}$

---

[1] By simple polynom interpolation of 3rd order.

The Algorithm to Extract the Timing Information

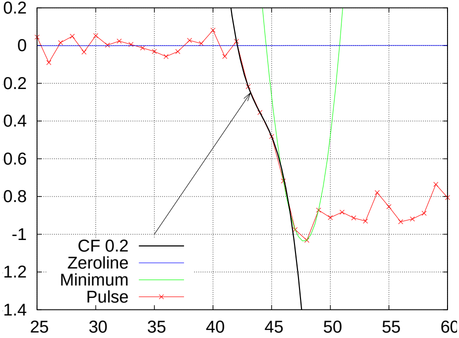


1. Find and interpolate the extremum[1]
2. Determine the zeroline (and its deviation) before the extremum
3. Interpolate the constant fraction point on the rising slope between zeroline and extremum[1]
4. Lifetime = $t_{Channel\ 1} - t_{Channel\ 2}$

---

[1] By simple polynom interpolation of 3rd order.

The Algorithm to Extract the Timing Information



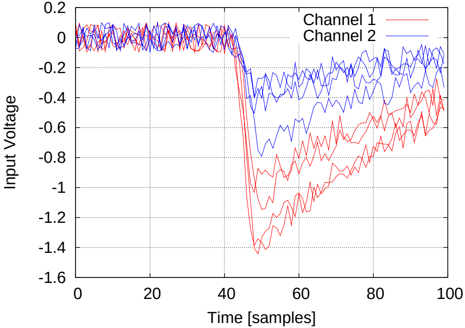
1. Find and interpolate the extremum[1]
2. Determine the zeroline (and its deviation) before the extremum
3. Interpolate the constant fraction point on the rising slope between zeroline and extremum[1]
4. Lifetime = $t_{Channel\ 1} - t_{Channel\ 2}$

Similar to analog constant fraction, called *true constant fraction* by [2].

---

[1] By simple polynom interpolation of 3rd order.

Making your own Artificial Pulses

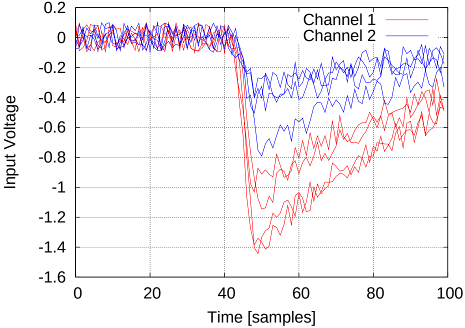


Four pulse pairs generated by EPOS Software

Side Note

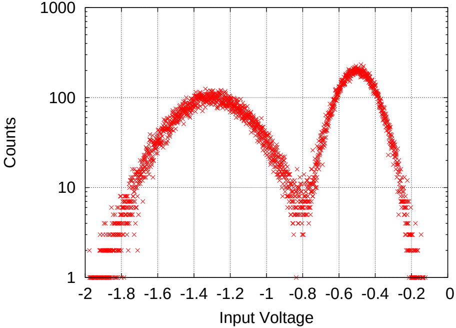
The EPOS Software is gone open-source and looking for users!
Get it for free [1].

Making your own Artificial Pulses: Shape



- Shaped like LSO on Hamamatsu H3378-50
- Risetime like 4 GS/s

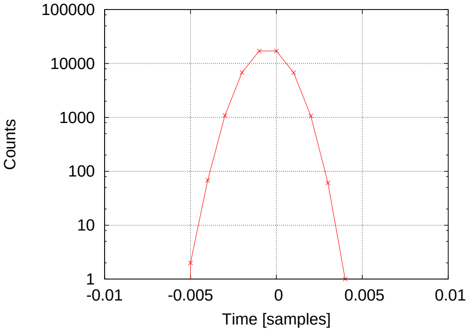
Making your own Artificial Pulses: Energy Spectrum



- Shaped like LSO on Hamamatsu H3378-50
- Risetime like 4 GS/s
- Energy distribution like ^{22}Na

- Energy spectrum closely to ^{22}Na but idealistic.

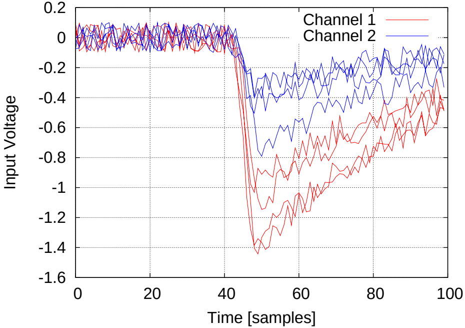
Making your own Artificial Pulses: Timing Distribution



- Shaped like LSO on Hamamatsu H3378-50
- Risetime like 4 GS/s
- Energy distribution like ^{22}Na
- Gaussian distributed timing

- Shift between pulses is Gaussian distributed.
- Shift of pulses to sampling clock is box distributed.

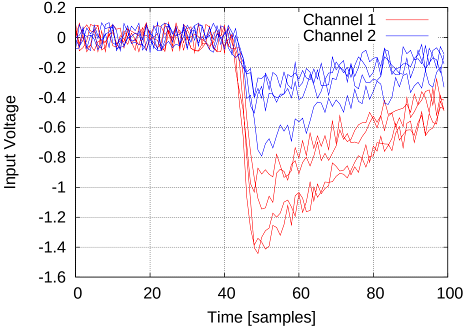
Making your own Artificial Pulses: Bit-depth



- Shaped like LSO on Hamamatsu H3378-50
- Risetime like 4 GS/s
- Energy distribution like ^{22}Na
- Gaussian distributed timing
- Variable bit-depth

- Possible bit-depths: 1-32 bits
- Native double resolution also possible

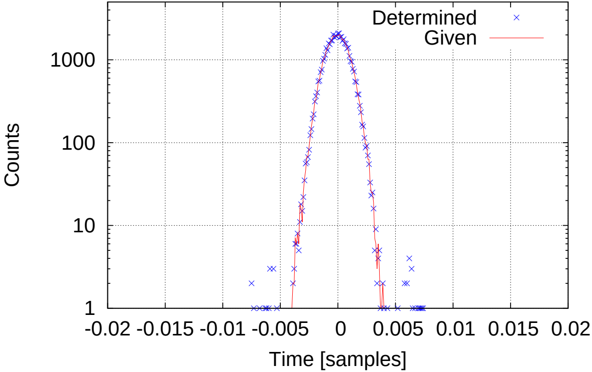
Making your own Artificial Pulses: Adding Noise



- Shaped like LSO on Hamamatsu H3378-50
- Risetime like 4 GS/s
- Energy distribution like ^{22}Na
- Gaussian distributed timing
- Variable bit-depth
- White noise added as wanted

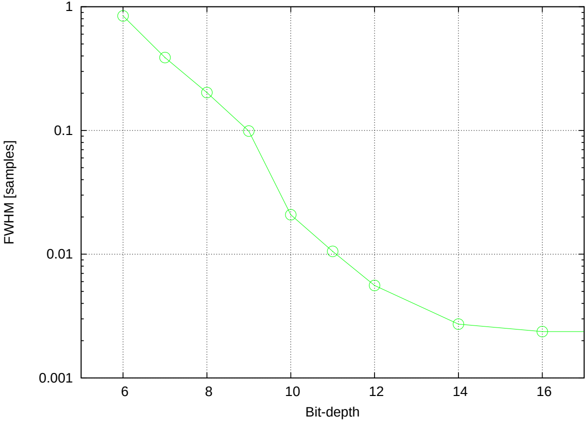
- White noise to simulate the uncertainties of the analog electronics.
- Level can be adjusted as wanted.

Double Resolution without Noise



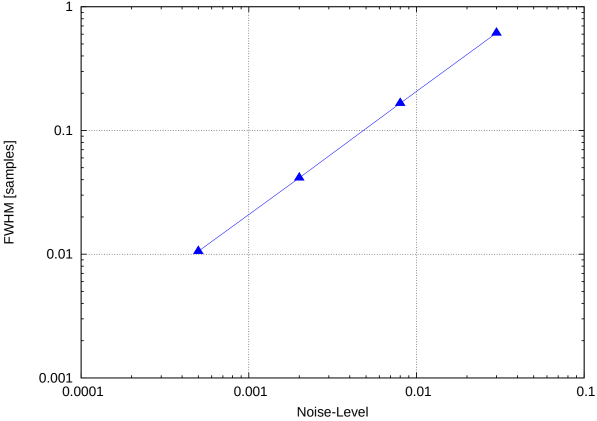
- No noise, native double resolution
- Given timing distribution: FWHM $= 0.0023582$ samples $\equiv 0.589$ ps
- Given distribution (—) $\equiv$ determined resolution ($\times$)
  $\Rightarrow$ Method works

Reducing the Bit-depth



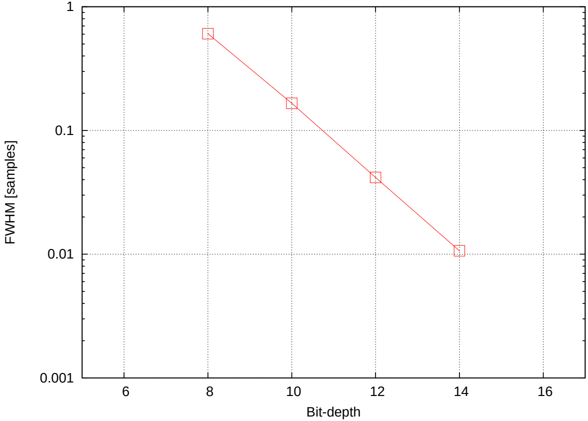
- Reduced bit-depth, no noise
- Timing resolution at 8-bit: 0.202 samples ≡ 50 ps

Noise of Effective Bits



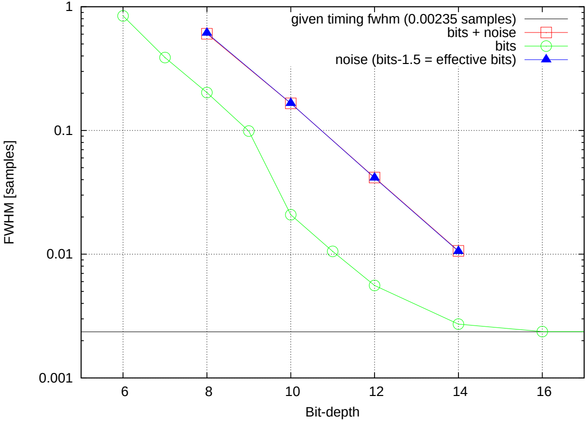
- Native double resolution, noise according to effective bits added
- Strong log-log dependency of timing resolution and noise level.

Noise and Reduced Bit-depth



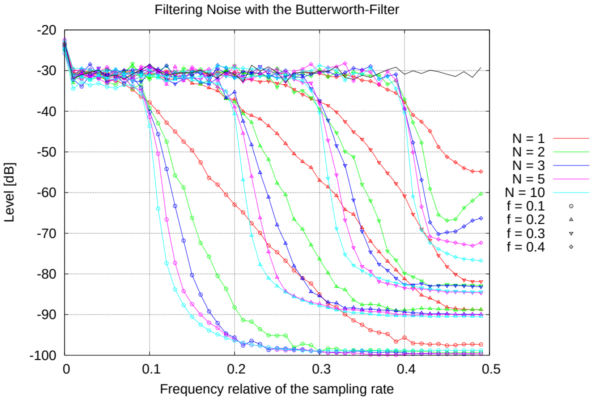
- Reduced bit-depth and noise from effective bits
- Timing resolution at 8-bit: 0.612 samples ≡ 153 ps

Finally Comparing the results



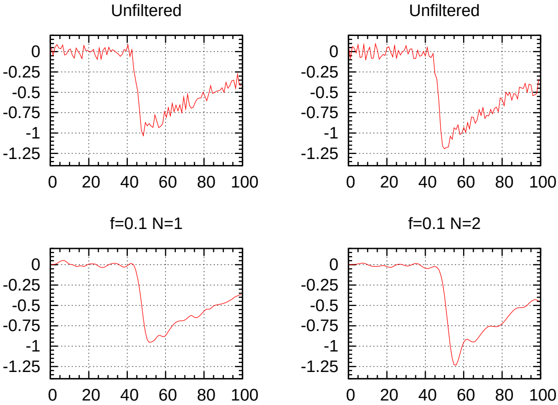
- Noise from effective bits has most influence
- Resulting timing resolutions: 8-bit: 153 ps, 10-bit: 41 ps

Applying a Lowpass Filter



- Butterworth lowpass (implementation taken from literature [3])
- Order and cutoff frequency can be set

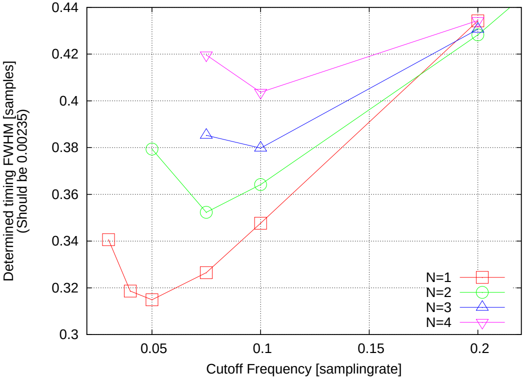
Applying a Lowpass Filter



Upper row Original signals as generated

Lower row Filtered by lowpass

Applying a Lowpass Filter: Results



Best Timing Resolution

$N = 1$ and $f = 0.05$ has FWHM of 0.31 samples $\equiv$ 75 ps.

Comparison of the Results

Method	**Relative Timing FWHM** [samples]	**4-GS/s "real" FWHM** [ps]
Vertical quantization only (8-bit)	0.202 samples	50 ps
Noise of effective 6.5 bit	0.612 samples	153 ps
Butterworth-Lowpass f=0.05 N=1	0.314 samples	75 ps

Comparing the results.

Lowpass filtering can almost remove the effect of the noise added from the analog electronics.

⇒ All with simple polynom interpolation for energy and constant fraction.

Literature, Links, Thanks

Thanks for your attention!

Get the slides from [4].

[1] positron.physik.uni-halle.de/epos/software.

[2] F. Bečvář.
Methodology of positron lifetime spectroscopy: Present status and perspectives.
Nuclear Instruments and Methods in Physics Research B, 261:871–874, 2007.

[3] S. D. Stearns.
Digital Signal Analysis.
Hayden Book Company Inc., 1975.

[4] positron.physik.uni-halle.de.