SPADIC 1.0 status, plans for 2.0

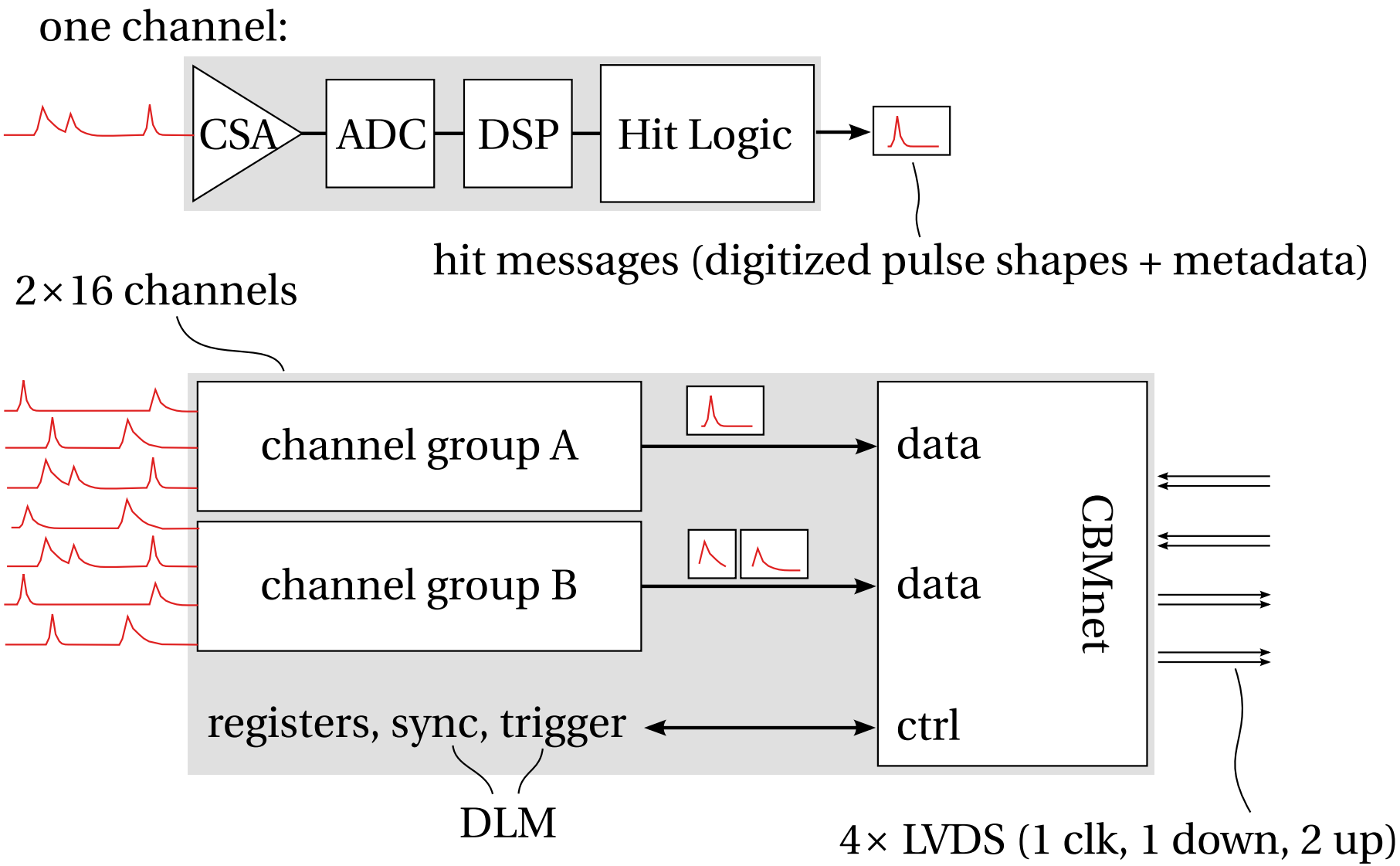
Michael Krieger

TRD Strategy Meeting
04.12.2014, GSI

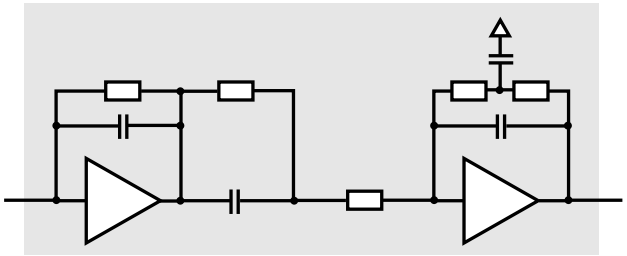
Section 1

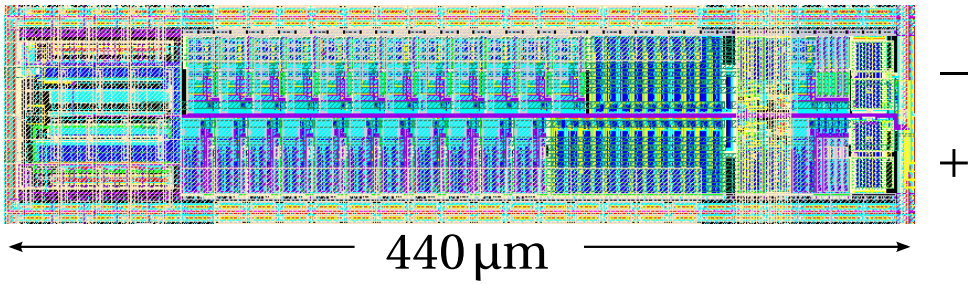
Reminder: SPADIC 1.0 architecture and features

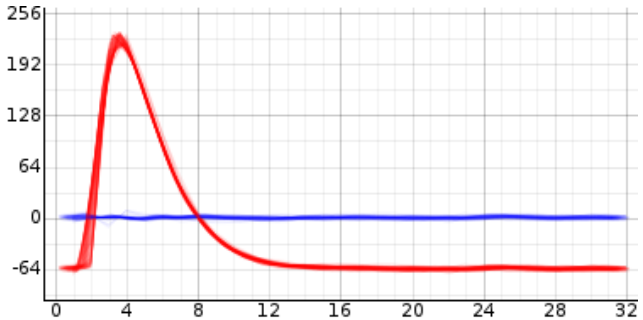
SPADIC 1.0 architecture overview



Charge sensitive amplifier

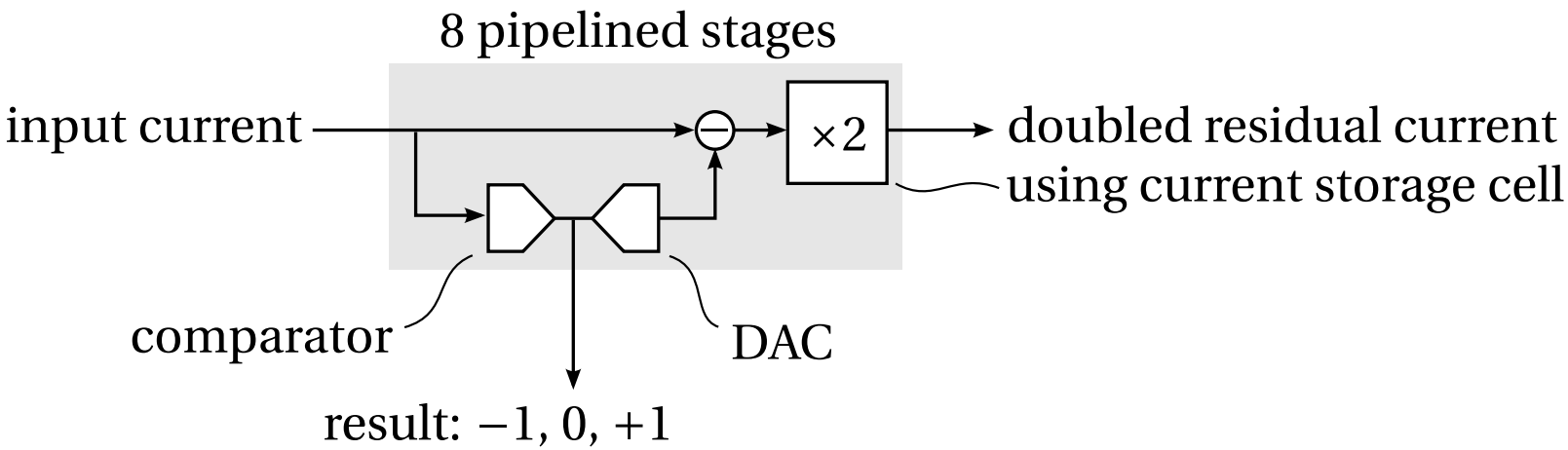






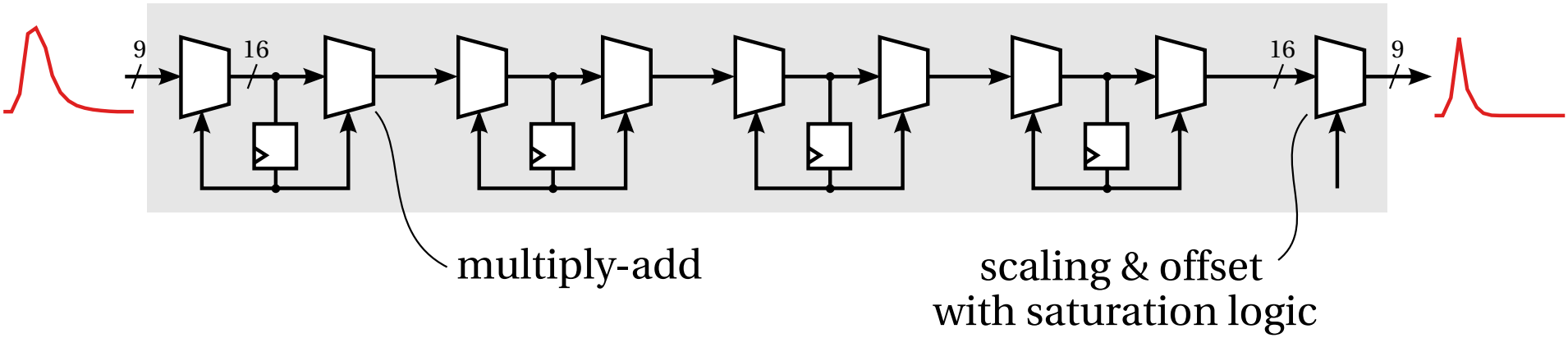
- input range: 75 fC
- $u_{out}(t) = (i_{in} * h)(t)$
- $h(t) \propto t/\tau \cdot e^{-t/\tau}$ (shaping time $\tau = 80$ ns)
- additional amplifier for negative polarity input selectable

ADC

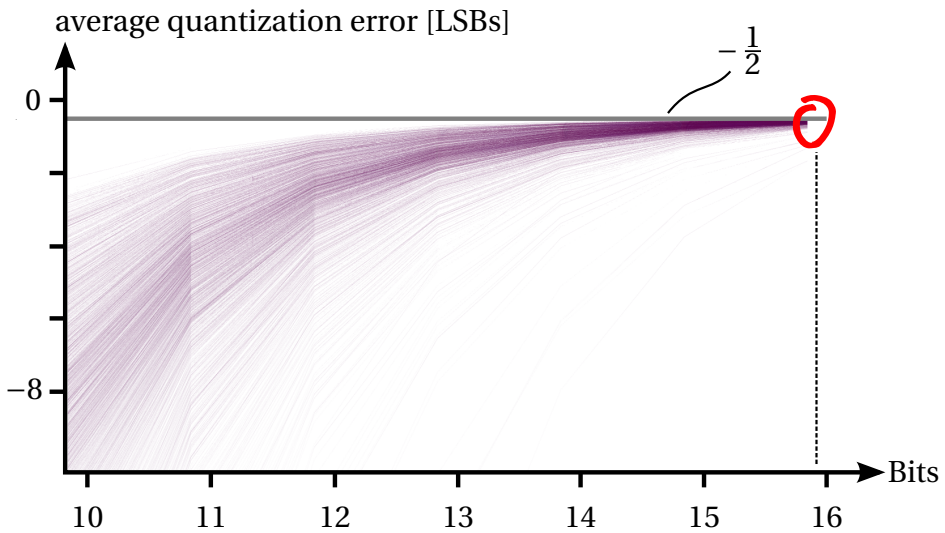


- current mode pipelined design
- resolution ≈ 8 bits
- 25 MHz sample rate, continuously running
- 9 bit signed output (−256..255)

Digital signal processing



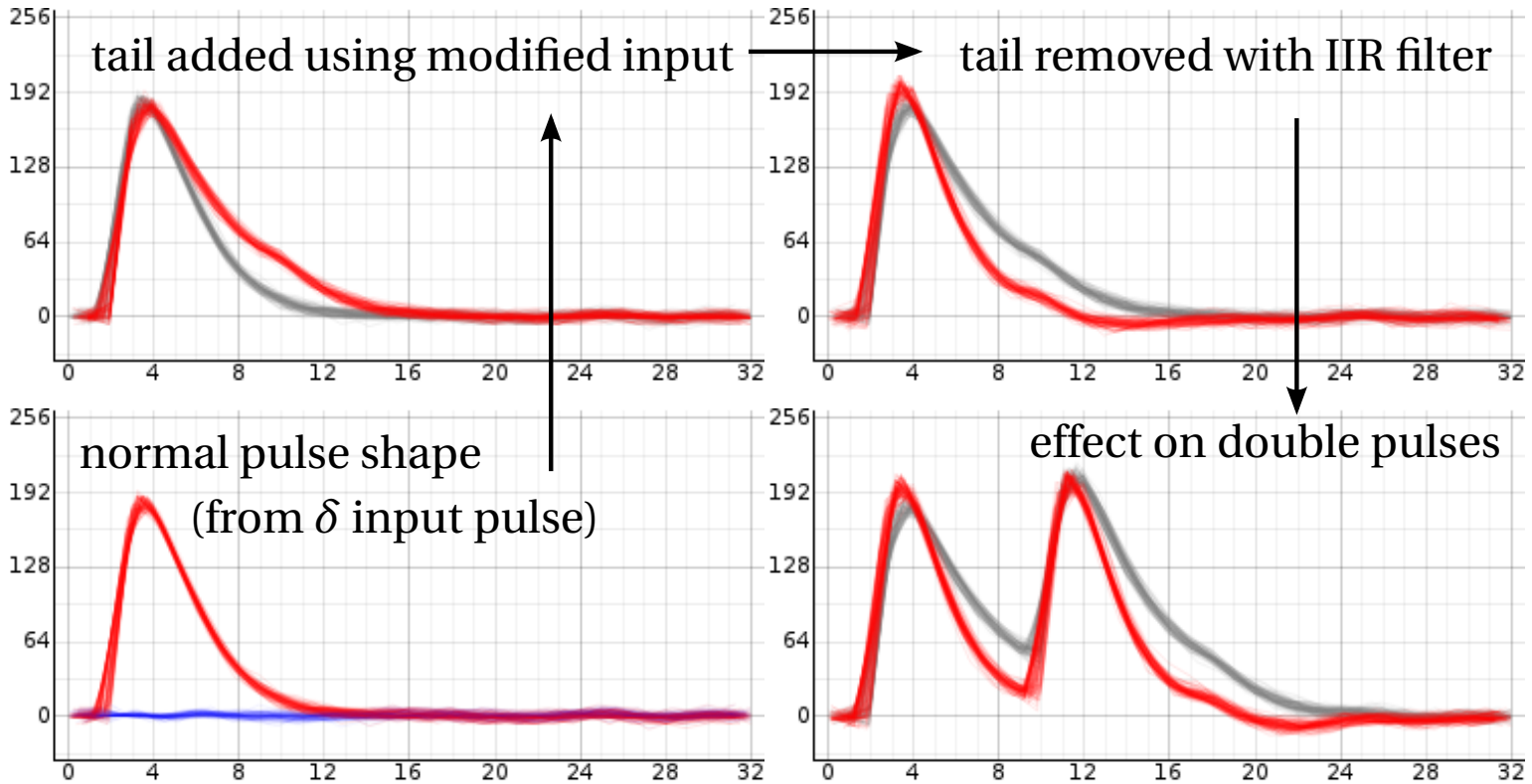
- IIR filter with 4 first order stages
- 16 bit internal resolution
- 6 bit coefficients
- freely programmable $(-\frac{32}{32}, \ldots, +\frac{31}{32})$



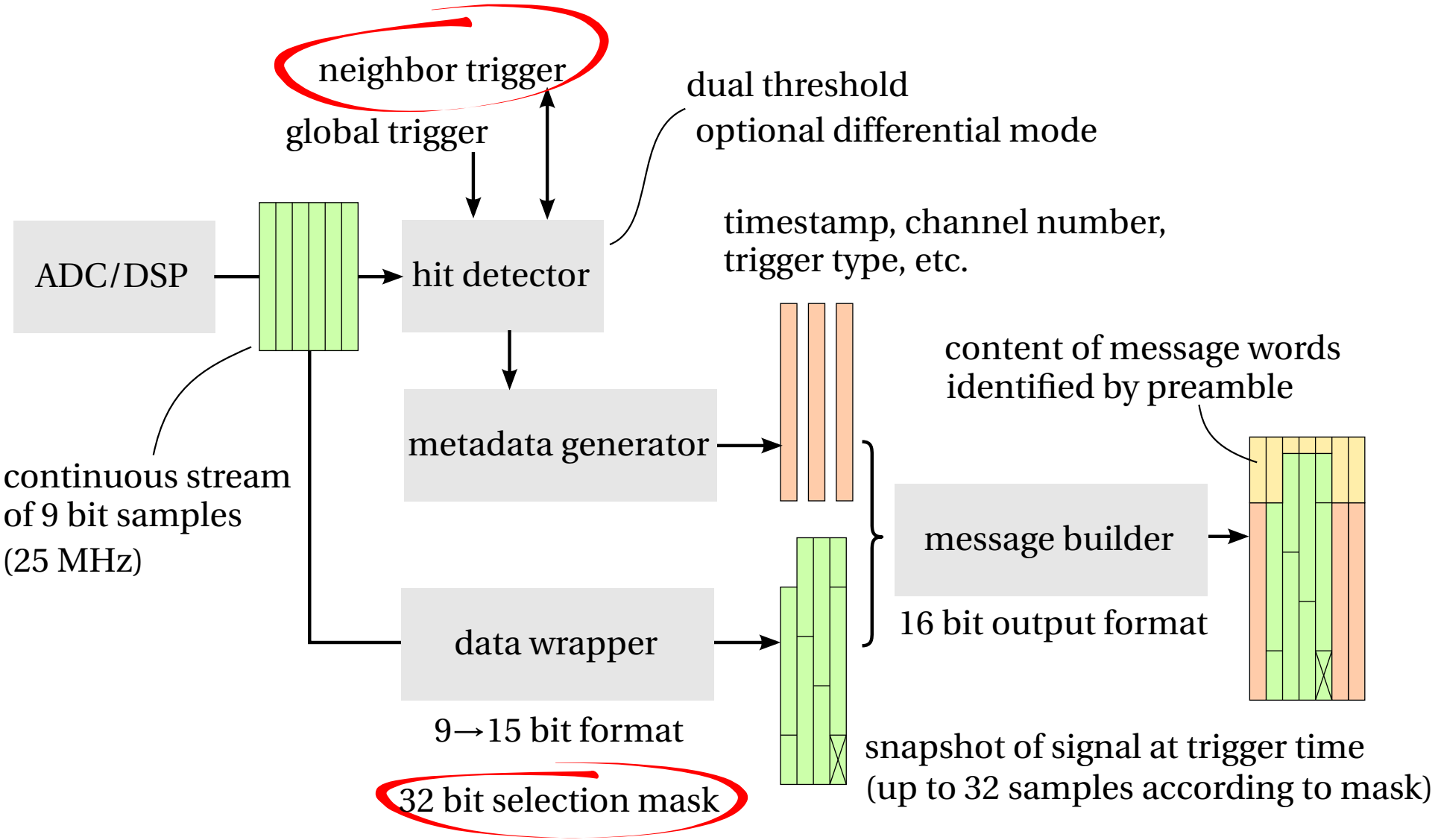
Digital signal processing

purpose: “ion tail” cancellation

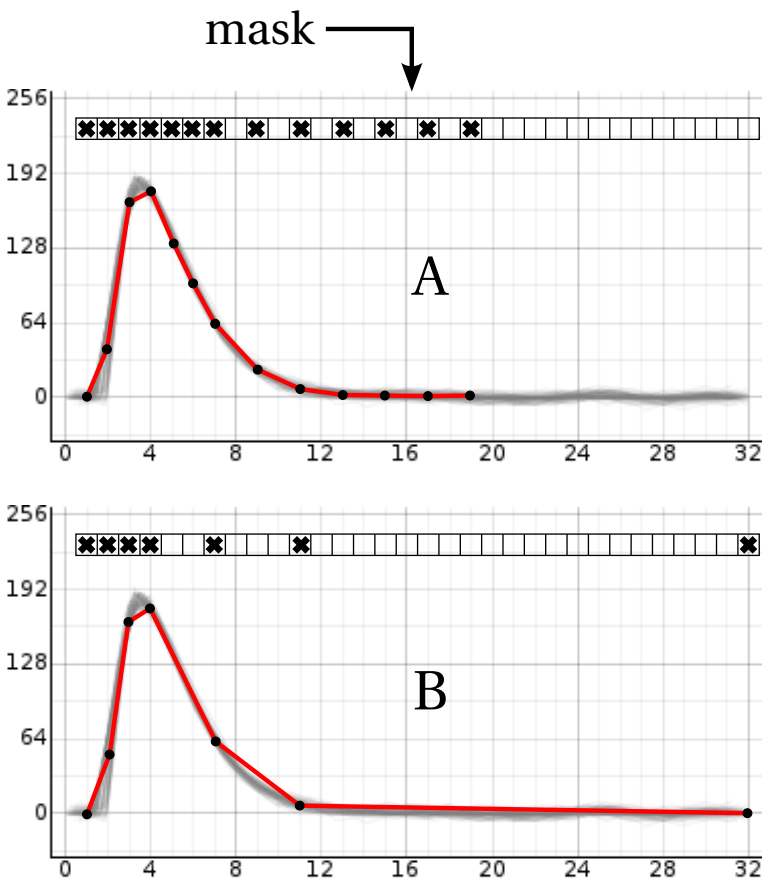
shorten pulses → reduce pileup/improve hit detection



Hit logic


neighbor trigger
global trigger
dual threshold
optional differential mode
ADC/DSP
hit detector
timestamp, channel number,
trigger type, etc.
content of message words
identified by preamble
metadata generator
continuous stream
of 9 bit samples
(25 MHz)
message builder
16 bit output format
data wrapper
9→15 bit format
32 bit selection mask
snapshot of signal at trigger time
(up to 32 samples according to mask)

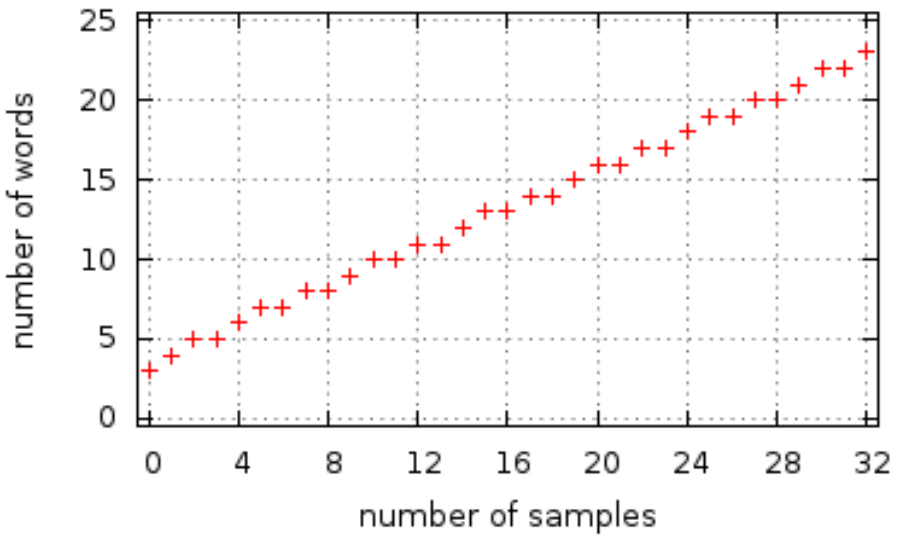
Selection mask examples



→ allows tradeoff between quality of signal reconstruction and data volume

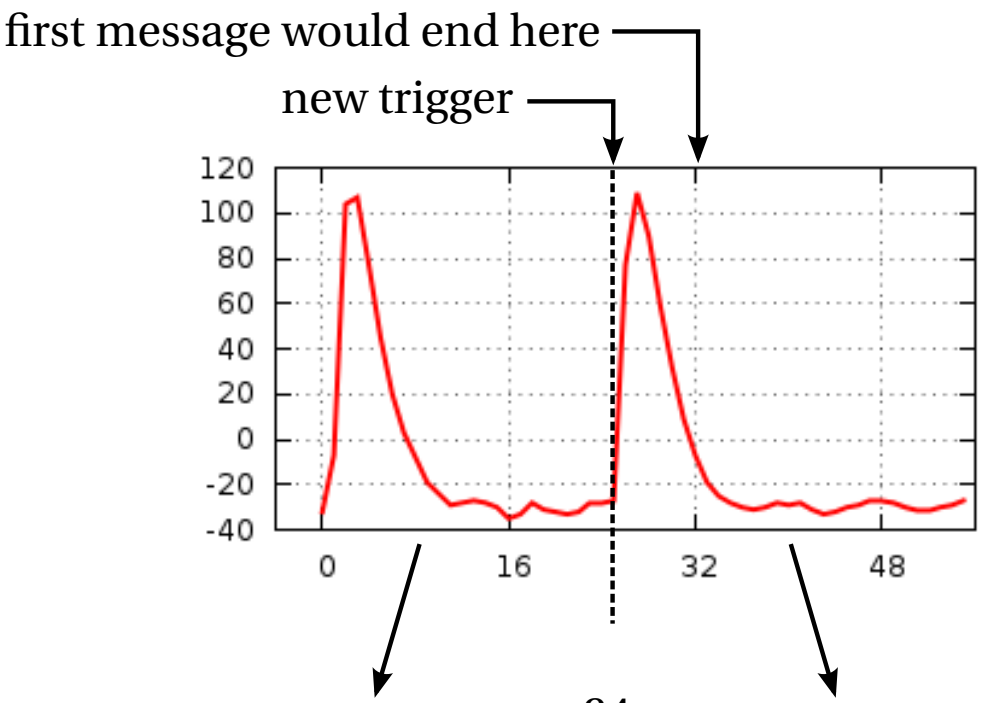
Message size (selection mask)

- $n_{\text{words}} = 3 + \left\lceil \frac{9 \times n_{\text{samples}} + 3}{15} \right\rceil$ for $1 \leq n_{\text{samples}} \leq 32$
- $n_{\text{words}} = 3$ for $n_{\text{samples}} = 0$
- minimal message: 3 words (no samples, only metadata)



Multi hits

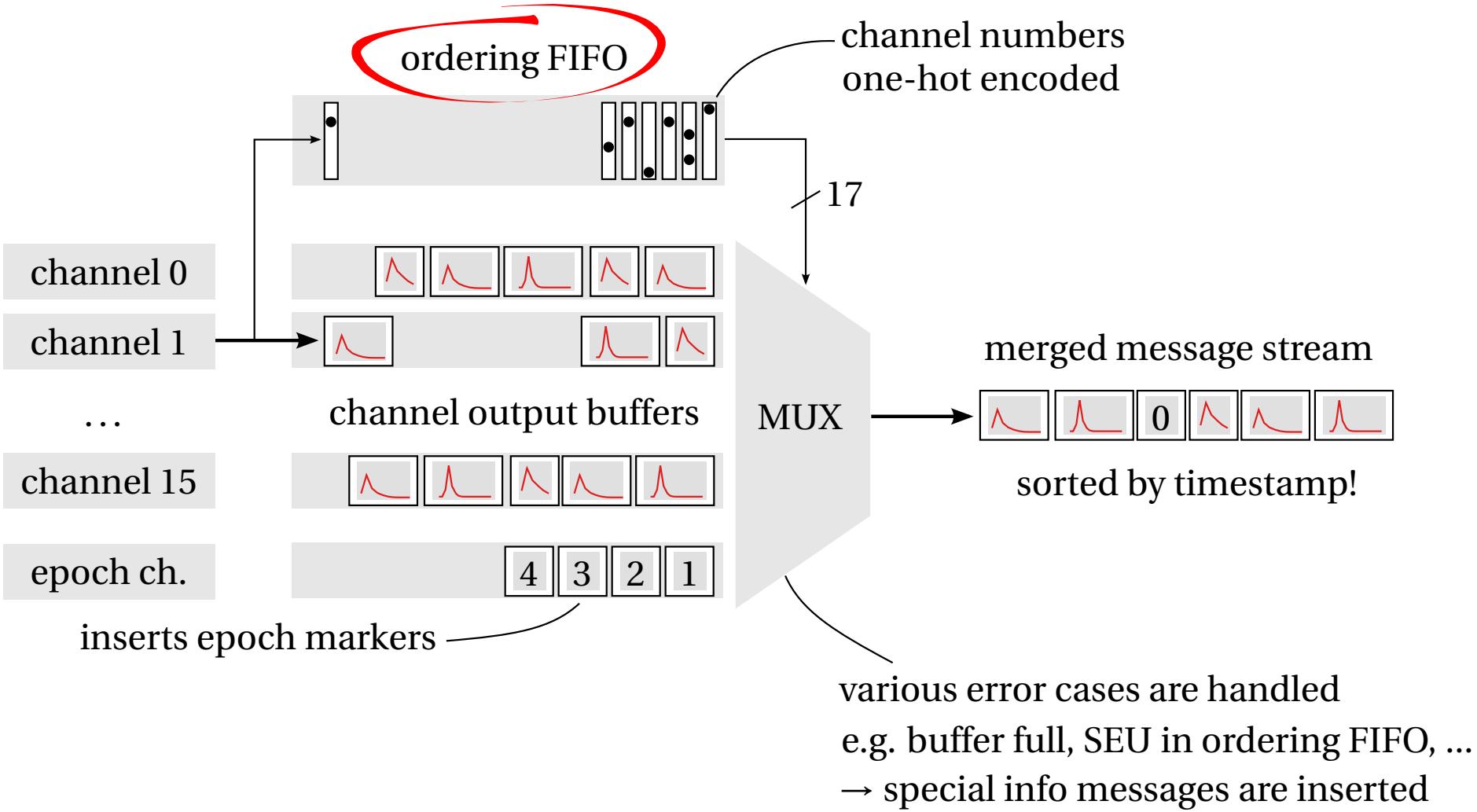
What happens when a channel is triggered again, before the current message is completed?



first message is shorter than usual

selection mask is restarted
(in this example, all bits are selected...)

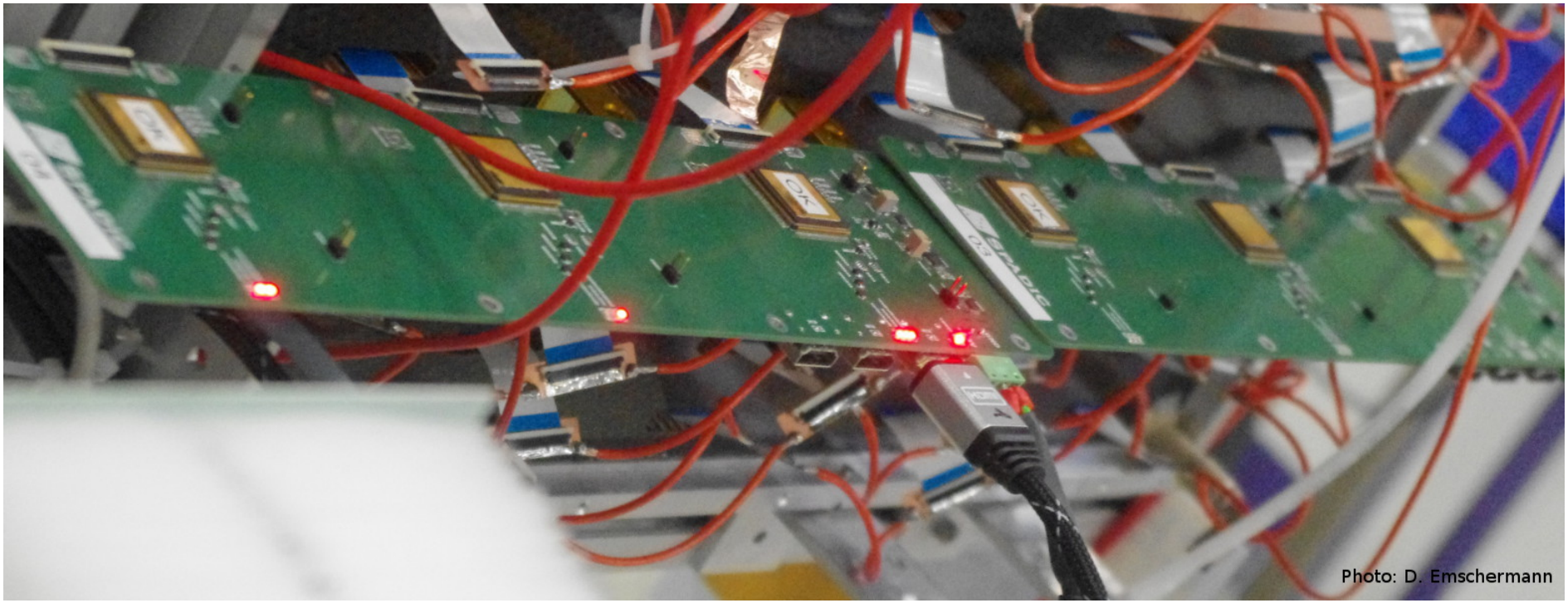
Message output multiplexing



Section 2

Test beam results

Setup



- SPADIC rev.B FEBs (single + triple) on MS, FFM chambers
- CBMnet connection to SysCore v3.1 over HDMI
- optical CBMnet connection to FLIB (acting also as DPB)
- data acquisition using FLESnet
- custom scripts for configuration and online monitoring

Analog performance

- unfortunately: CSA instability observed in every 2–3 chips, took some time to find good ones (combined with finding a good HDMI cable . . .)
- surprisingly low noise, if enough copper tape applied between FEB and backside of TRD (baseline σ = 3–10 ADC units, compared to 1–2 in the lab with no detector)
- → packaging and current FEB design work well
- near optimal use of available dynamic range (baseline at ≈ -230, pulses go above +200)
- available baseline trim (ADC setting) sufficient to equalize all channels

Digital performance

- everything worked as expected (bug in comparator could be worked around)
- some features have not yet been used much, but are still good to have (digital filter, differential trigger mode, digital scaling)
- other features have proven invaluable for debugging: digital offset, seeing full pulse shapes, forced trigger

Section 3

Plans for SPADIC 2.0

Analog frontend and digital part

- fix amplifier instability
- fix bug in comparator (positive $\not>$ negative)
- don't change or remove any existing features unless there are very good reasons

possible new features (must improve performance):

- combine information from neighbor triggered channels in one hit message
- calculate sum/peak/... of a pulse

Communications backend

- replace CBMnet by custom SPADIC–e-link protocol
- adjust interfaces for slow control, synchronization and data acquisition as necessary
- use concepts developed for STS-XYTER if applicable, for example (as far as I have understood them):
  - all downlink communication consists of reading or writing single registers (→ there are virtual registers which, when written, reset the timestamp, trigger all channels, etc.)
  - use variable numbers of links for data transport, distribute data evenly across links

Thank you for your attention.