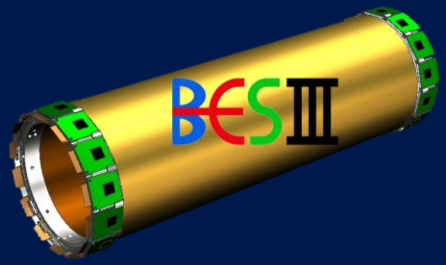
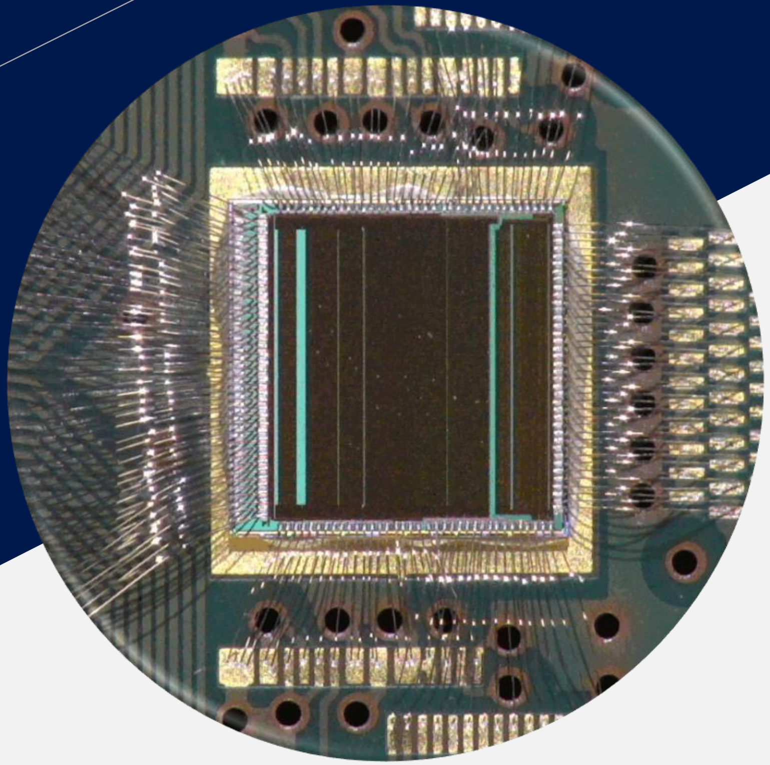




TIGER response

How to model and simulate TIGER response

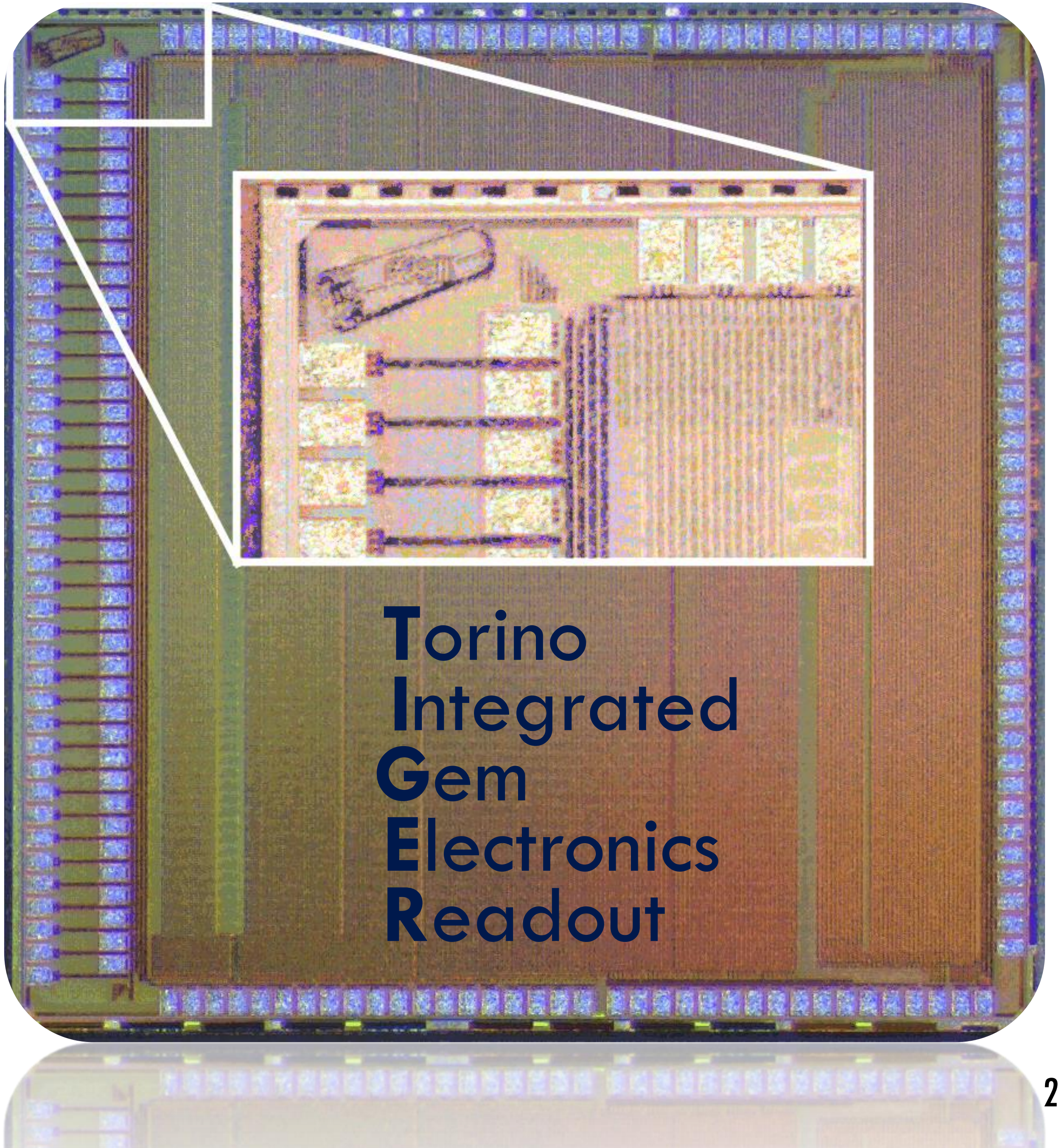
Fabio Cossio

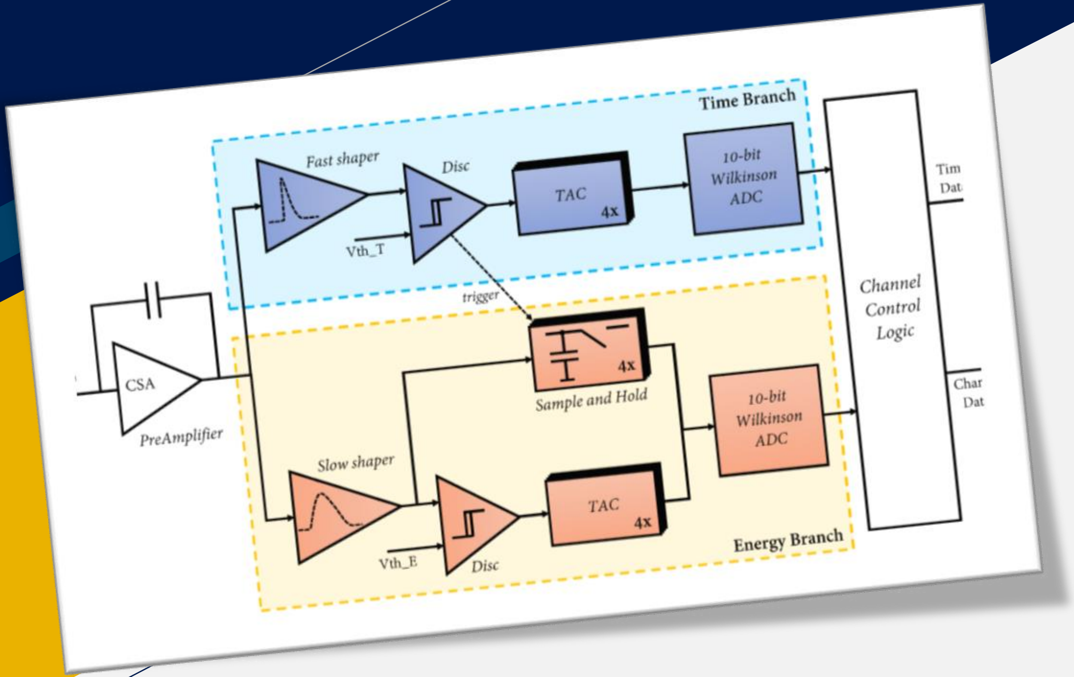
INFN Torino

fabio.cossio@to.infn.it

List of contents

- TIGER channel architecture
- Front-End response model
- Charge and Time measurements
- Calibration files (VTH, TDC, QDC)



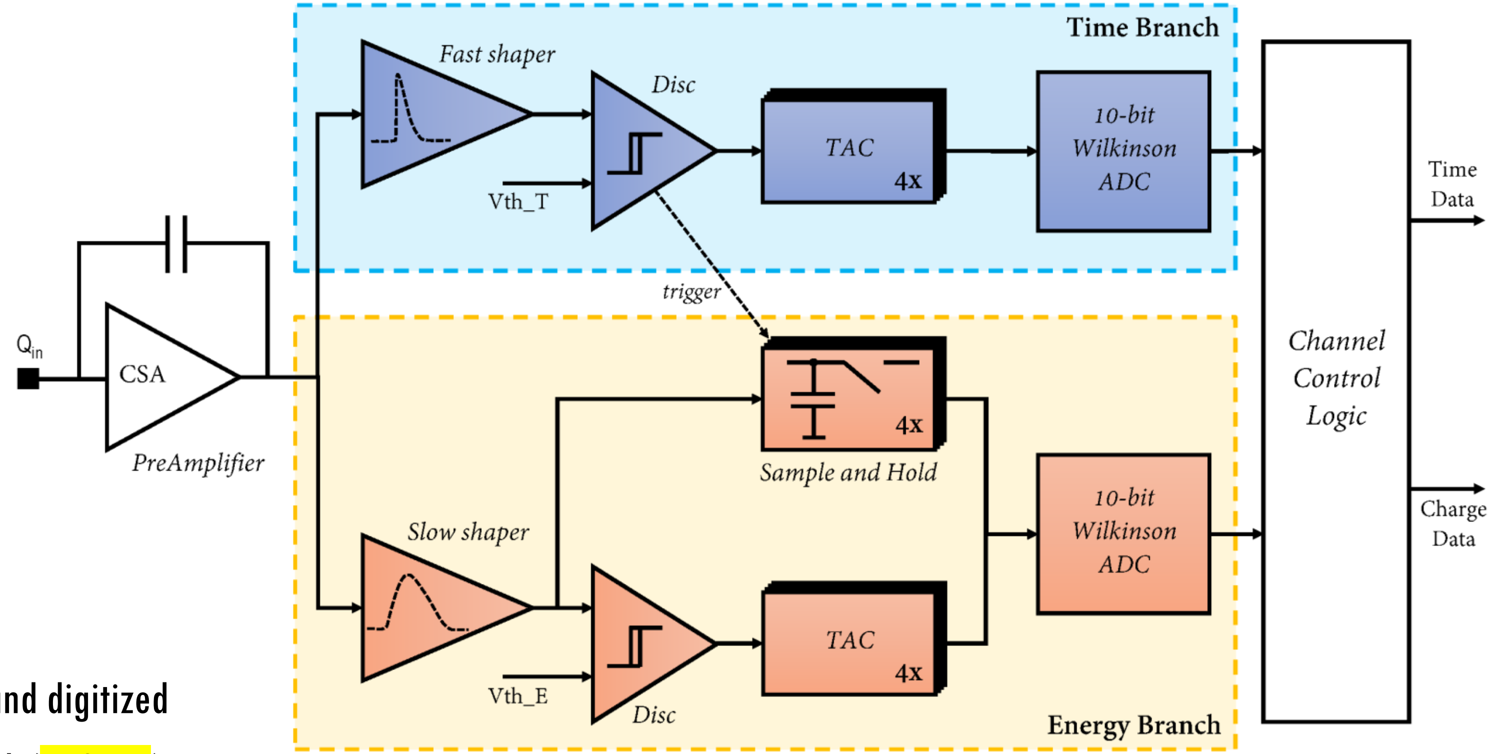


Channel architecture

Main building blocks of one TIGER channel

TIGER channel overview

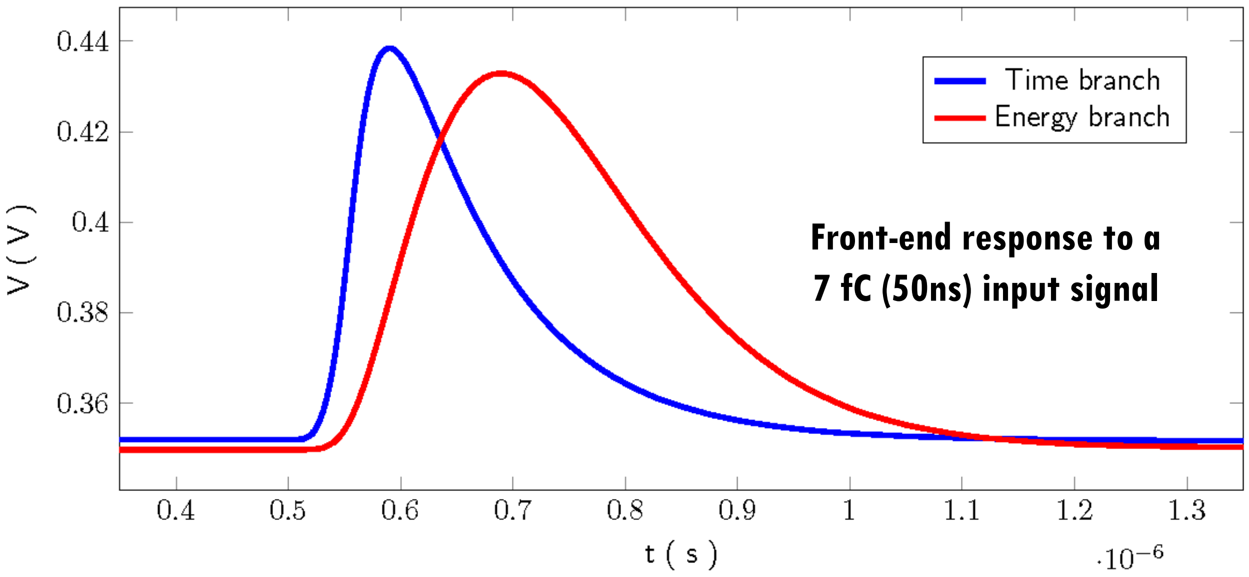
- **Dual-branch** architecture optimized for time and charge measurements
- Two leading-edge **discriminators**:
  - 6-bit DAC for threshold equalization
  - dual-threshold readout mode
- **Timestamp** on rising edge of fast branch using low-power TDCs based on analogue interpolation
- **Charge** measurement
  - **S&H**: slow shaper output peak voltage sampled and digitized
  - **ToT**: timestamp on rising/falling edges of T-branch (BACKUP)

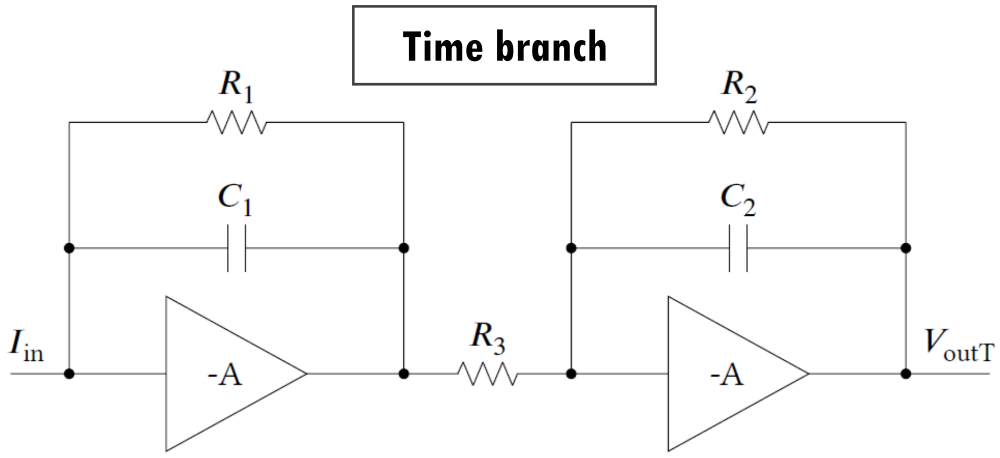


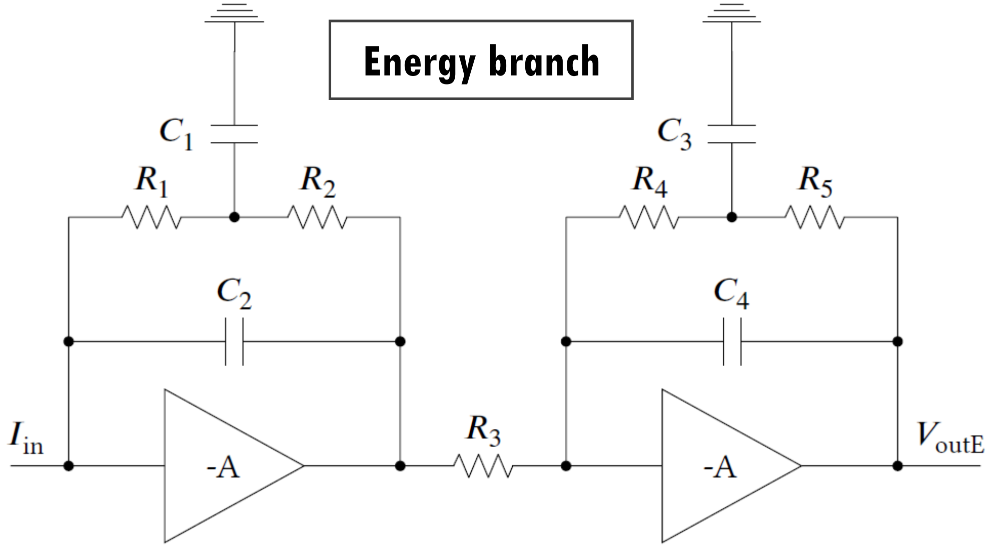
- Charge and time measurements are both encoded using a 10-bit Wilkinson ADC, delivering a **fully-digital** output

Front-End

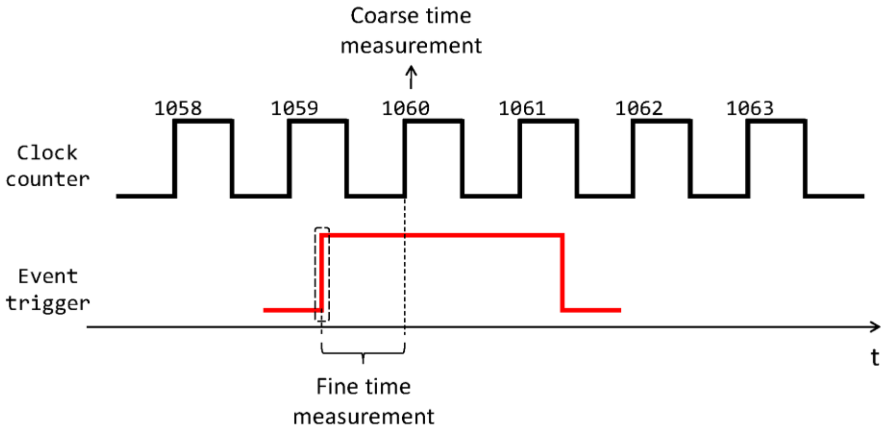
- CSA + dual-branch shaper
  - Time branch: 70 ns peaking time shaper for timing measurements (**timestamp** and ***ToT***)
  - Energy branch: 170 ns peaking time shaper for charge measurement in ***S&H mode***





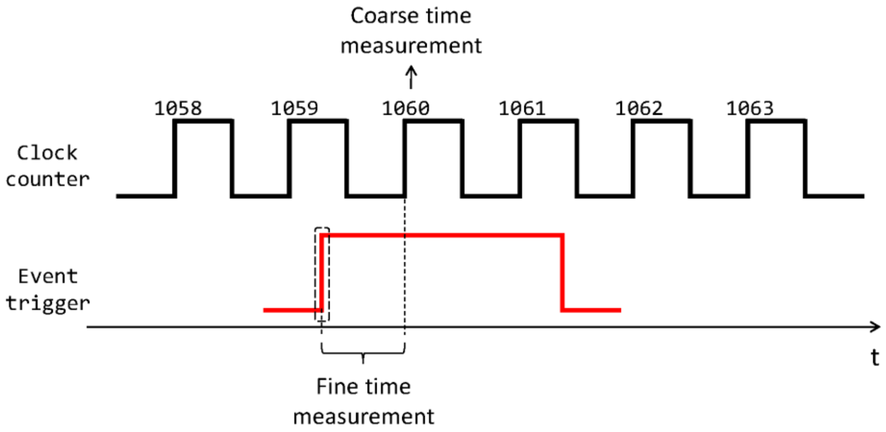


TDC operation

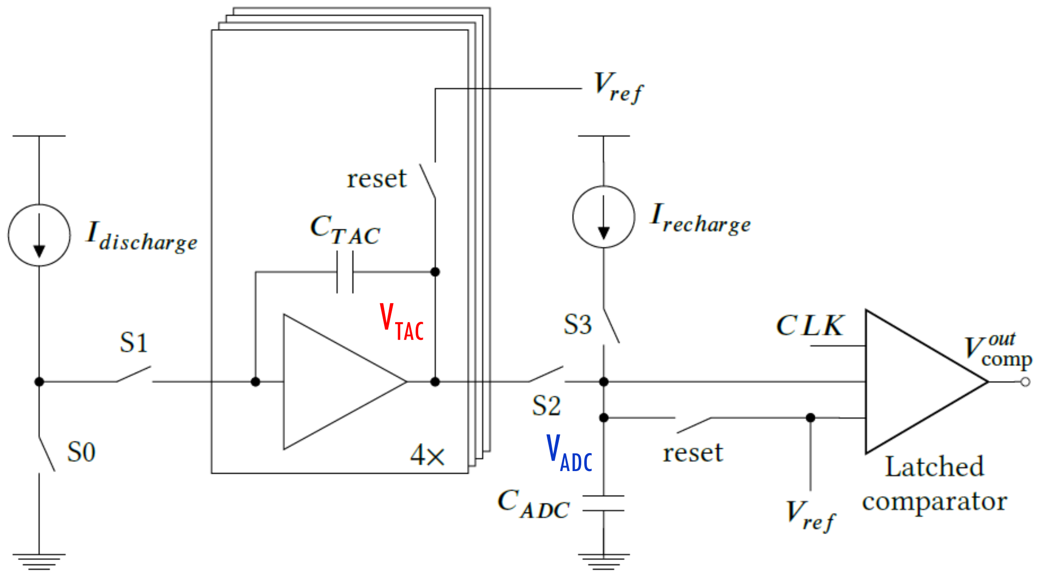


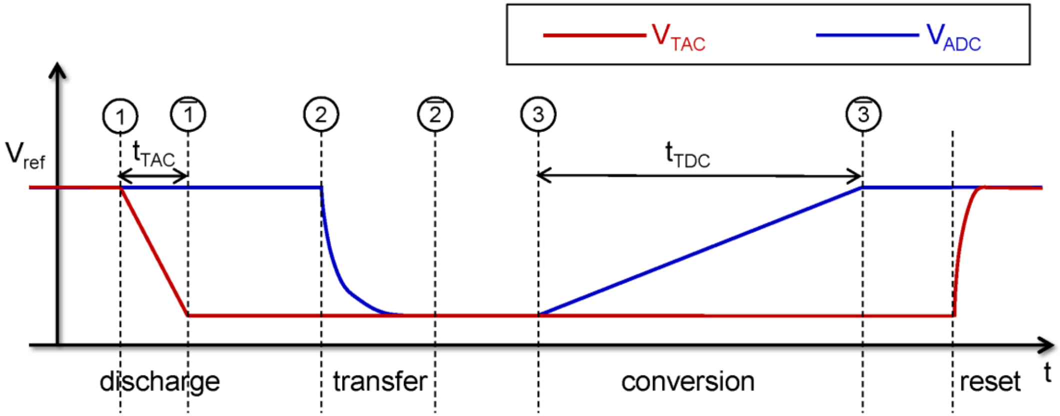
- **Coarse time measurement** from the chip master clock counter (160 MHz)

TDC operation



- **Coarse time measurement** from the chip master clock counter (160 MHz)
- **Fine time measurement** with low-power analogue TDCs based on time interpolation
  - Each TDC has 4 Time-to-Amplitude Converters (TAC)
  - Calibration is performed for each TAC
  - LUT provides gain-offset correction for each TAC





t_{TAC} = fine time measurement [ps]

$C_{ADC} = 4 \cdot C_{TAC}$

Interpolation factor = 32 x 4 = 128

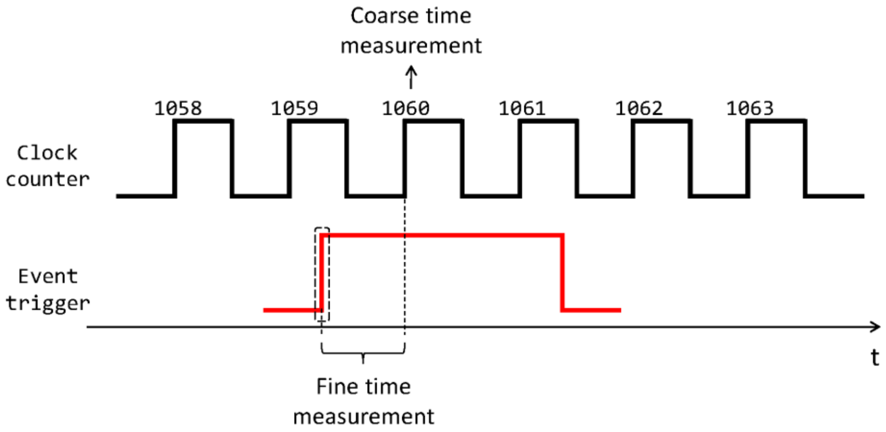
t_{TDC} = Tfine [ADC]

$I_{discharge} = 32 \cdot I_{recharge}$

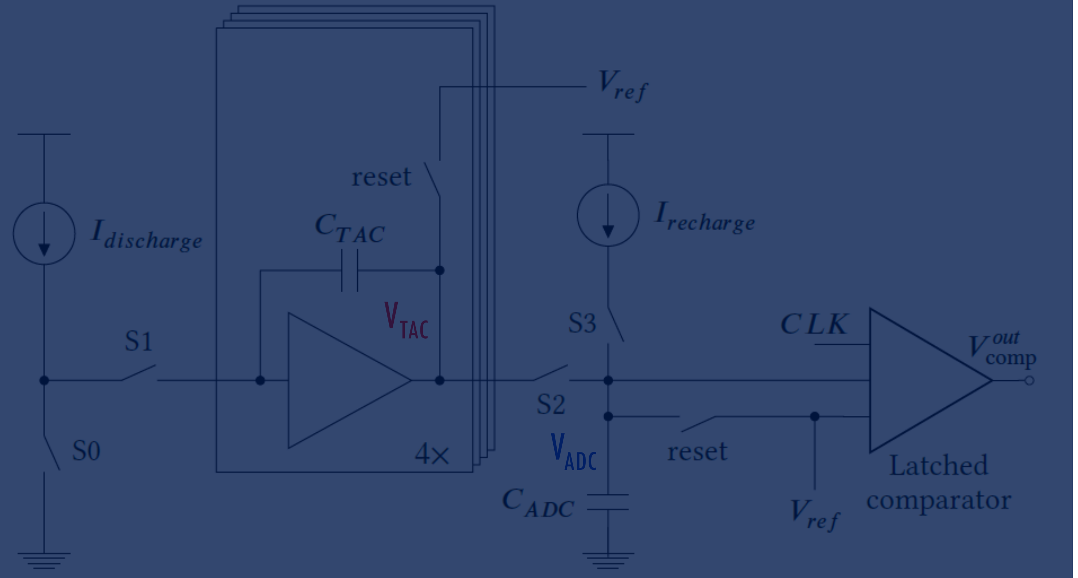
time_bin = 6.25 ns / 128 = 50 ps

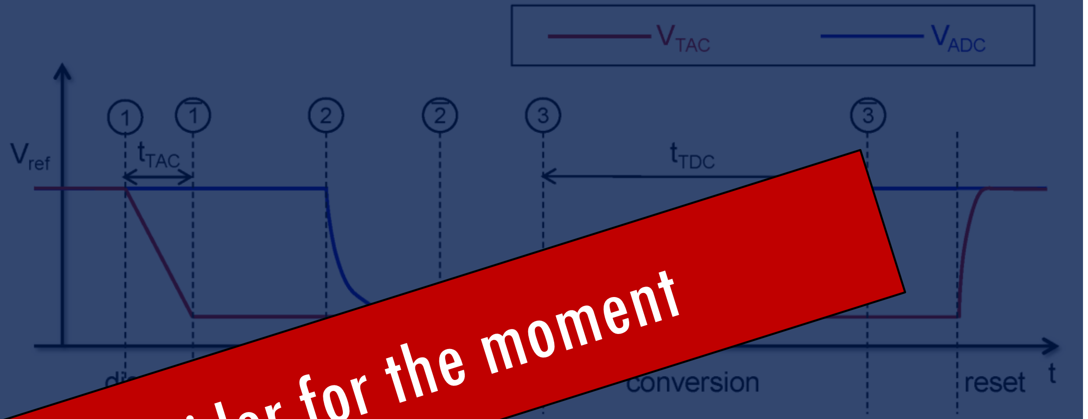
t_{TAC} [ps] = 50 ps · t_{TDC} [ADC]

TDC operation



- **Coarse time measurement** from the chip master clock counter (160 MHz)
- **Fine time measurement** with low-power analogue TDCs based on time interpolation
  - Each TDC has 4 Time-to-Amplitude Converters
  - Calibration is performed for each TAC
  - LUT provides gain-offset correction for each TAC





Don't consider for the moment

...asurement [ps]

... $4 \cdot C_{TAC}$

Interpolation factor = 32 x 4 = 128

t_{TDC} = Tfine [ADC]

$I_{discharge} = 32 \cdot I_{recharge}$

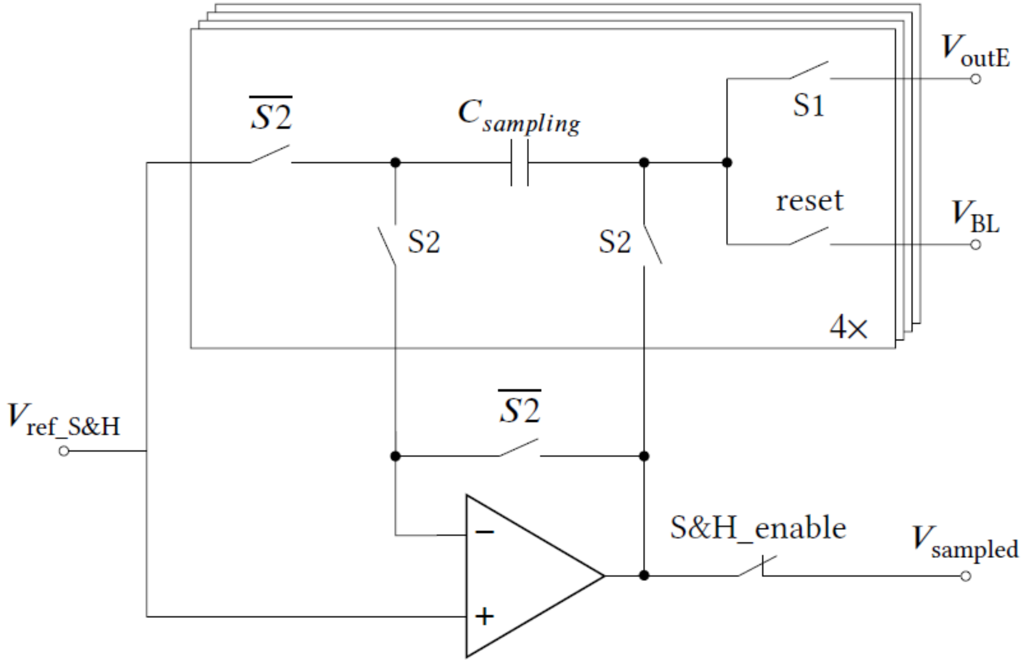
time_bin = 6.25 ns / 128 = 50 ps

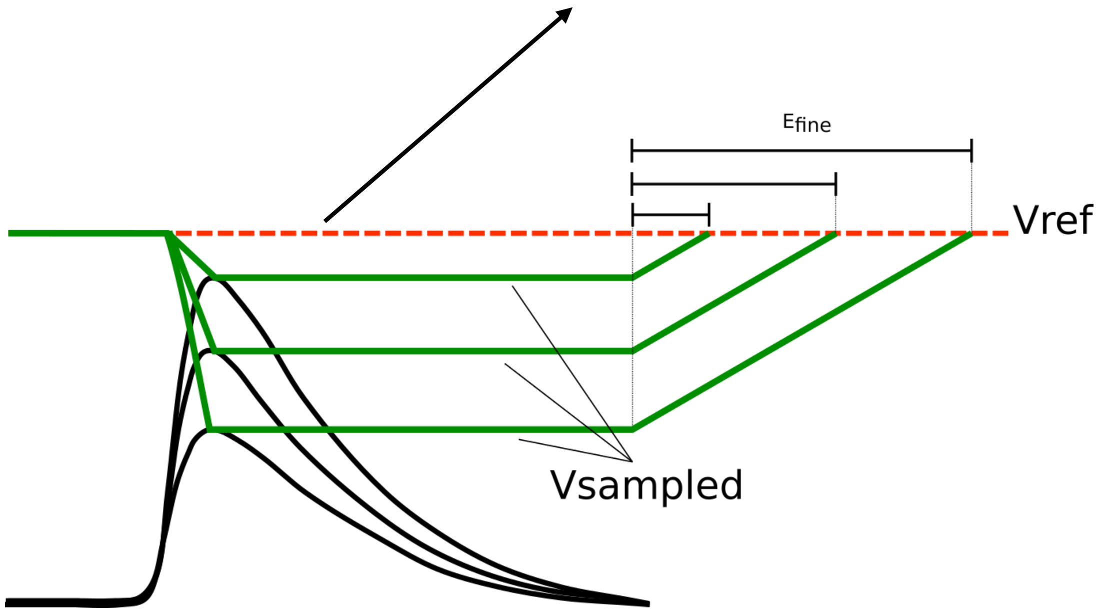
t_{TAC} [ps] = 50 ps · t_{TDC} [ADC]

Sample-and-Hold circuit

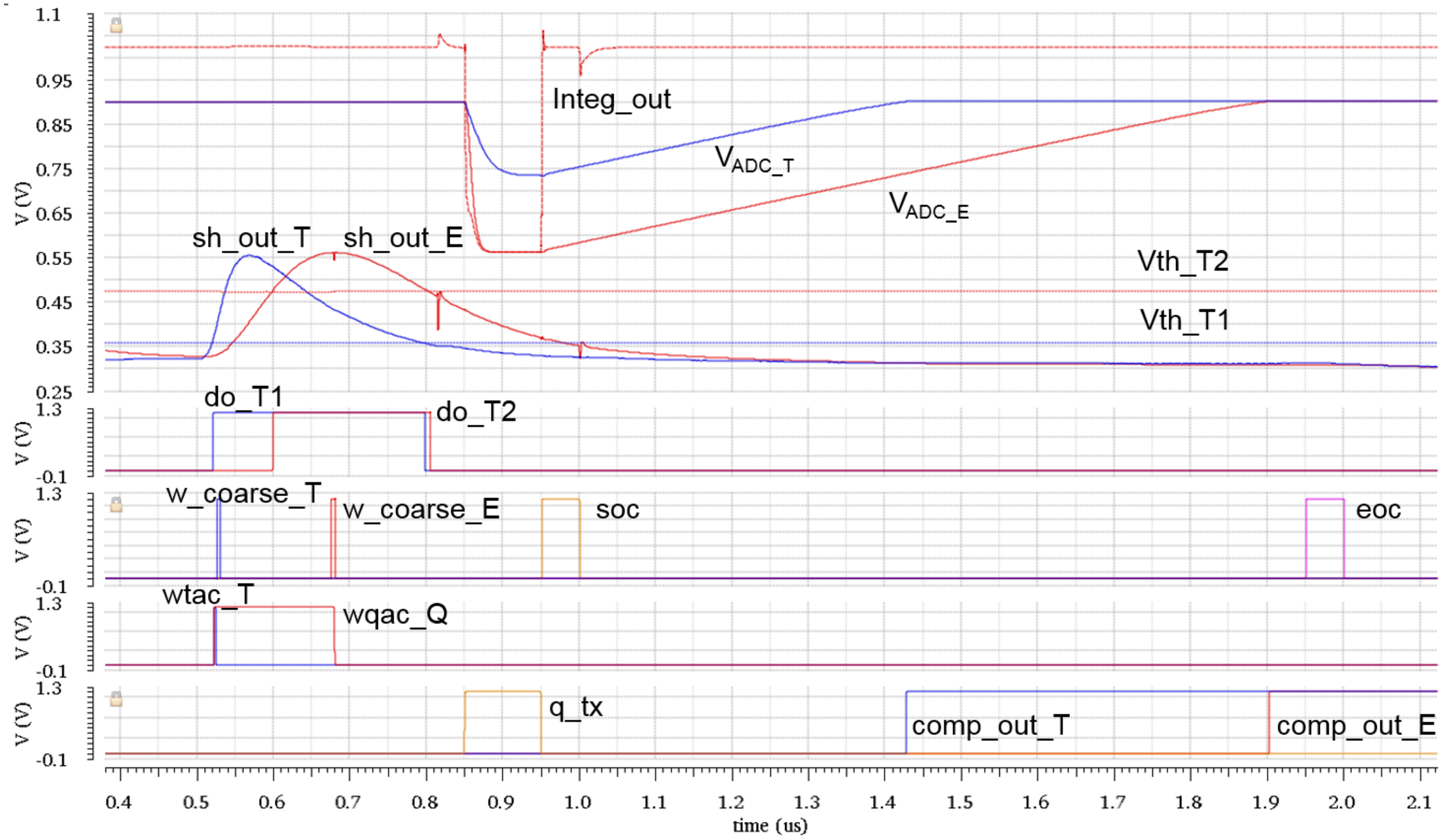
- **Charge measurement** with S&H circuit sampling the E-branch shaper output
- Programmable sampling time targeting the **signal peak**
- Digitization with **Wilkinson ADC** shared with the TDC

S&H saturation when
$\mathbf{V_{sampled} > V_{ref}}$





Full channel simulation



TIGER output words (64-bit)

- Charge and time measurements are both encoded using a 10-bit Wilkinson ADC, delivering a fully-digital output

EVENT WORD

K28.1	0b10	ch_id 6 bits	TAC	Tcoarse 16 bits	Ecoarse 10 bits	Tfine 10 bits	Efine 10 bits

FRAME WORD

K28.1	0x00	reserved	frame count 16 bits	SEU count 15 bits

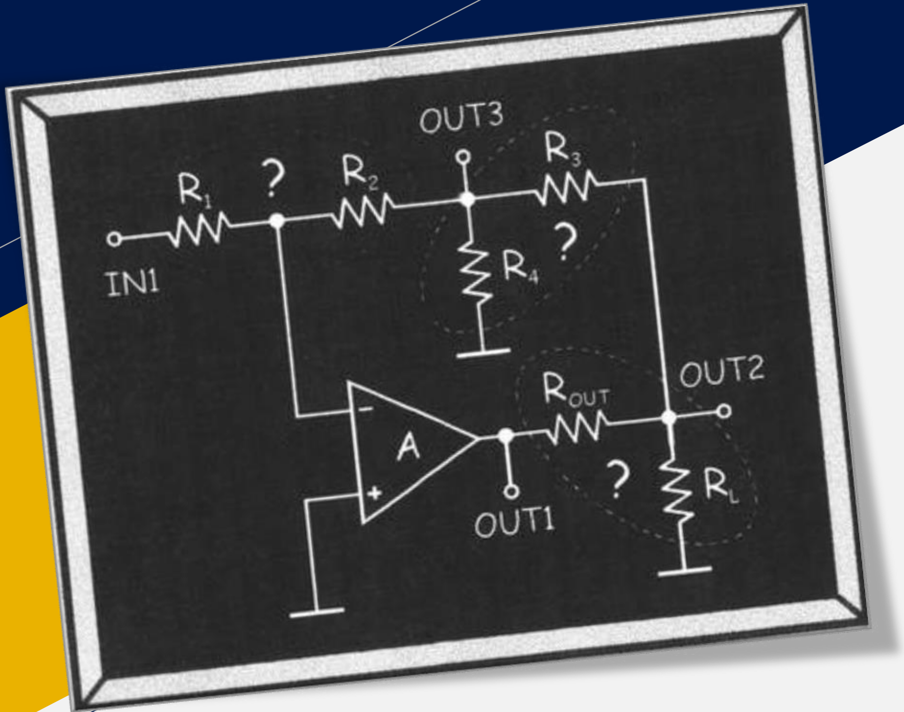
COUNT WORD

K28.1	0x01	reserved	ch_id 6 bits	counter value 24 bits

1. **Event word** digitized output of each hit
2. **Frame word** global timing reference
3. **Count word** read some counters inside TIGER for debug purposes

Event word

Bits	Parameter	Description
63:56	K28.1	Start of the 64-bit word identifier
55:54	0b10	Event word identifier
53:48	Channel_id	Channel identifier
47:46	TAC_id	TAC index
45:30	Tcoarse	Leading edge coarse time tag
29:20	Ecoarse	Falling edge coarse time tag (ToT mode) Sampling stop time tag (S&H mode)
19:10	Tfine	T-branch TDC fine time measurement
9:0	Efine	E-branch TDC fine time measurement (ToT mode) ADC charge value (S&H mode)



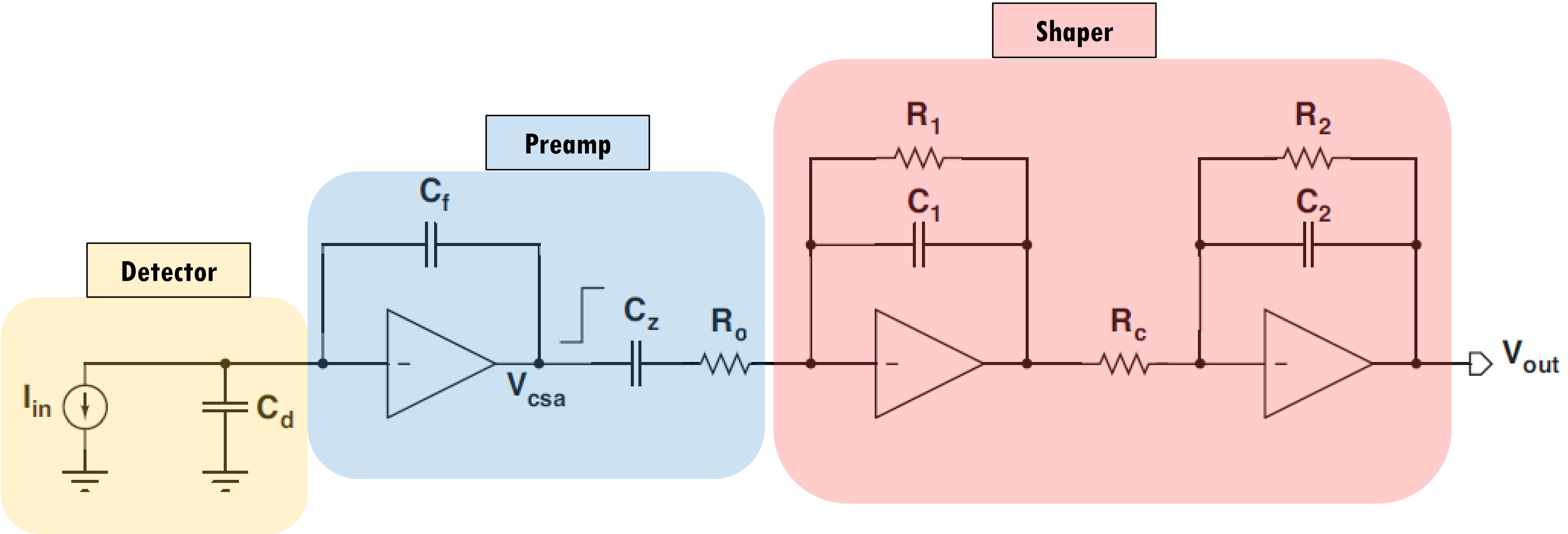
Front-End model

Transfer functions of TIGER analogue very-front-end

TIGER model

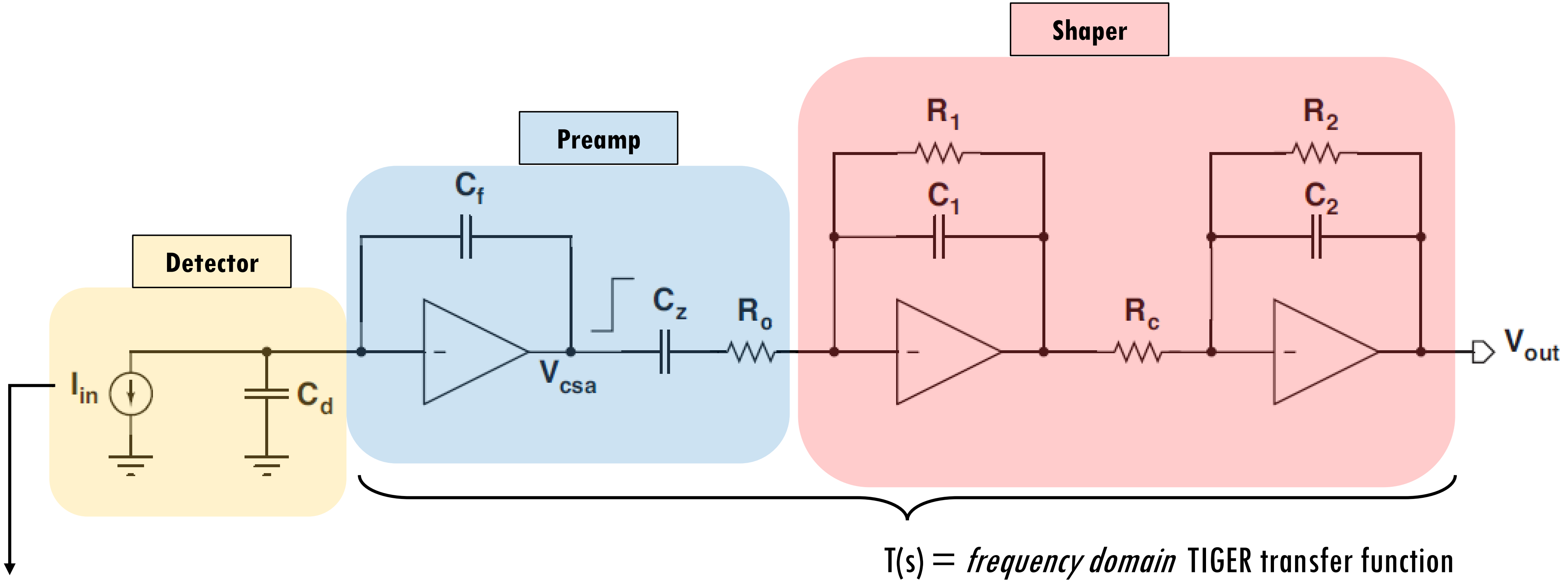
- Build a model to simulate the TIGER response in order to be integrated in the CGEM-IT simulation code
- The model must:
  - ❑ take into account the different duration and shape of input signals
  - ❑ show the different responses from T-branch and E-branch
  - ❑ be faster than computer circuit simulator
  - ❑ anything else?

TIGER T-branch



- $C_f = 0.150$
- $C_z = 20\ C_f = 3$
- $R_0 = 1$
- $R_1 = 100$
- $C_1 = 1$
- $R_2 = 20$
- $C_2 = 1$
- $R_c = 20$
- $V_{bsln} \approx 350\ mV$

TIGER model (T-branch)

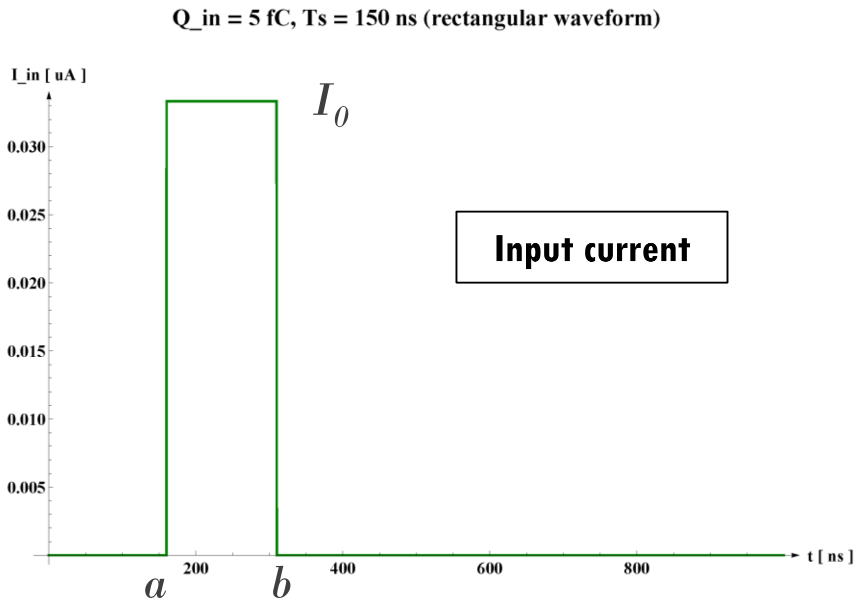


$I_{in}(s)$ = *frequency domain* input signal function (input signal dependent)

T(s) = *frequency domain* TIGER transfer function (fixed)

$$V_{out\_T}(s) = I_{in}(s) \cdot T(s)$$

TIGER model (T-branch)



$$I_{in}(t) = I_0 \cdot (\theta(t-a) - \theta(t-b))$$

$$\begin{cases} T_s = b - a \\ Q_{in} = I_0 \cdot T_s \end{cases}$$

$$I_{in}(s) = L\{I_{in}(t)\}$$

$$T(s) = \frac{1}{C_f} \cdot \frac{R_0 C_z}{(1 + sC_z R_0)} \cdot \frac{R_1}{(1 + sC_1 R_1)} \cdot \frac{1}{R_C} \cdot \frac{R_2}{(1 + sC_2 R_2)}$$

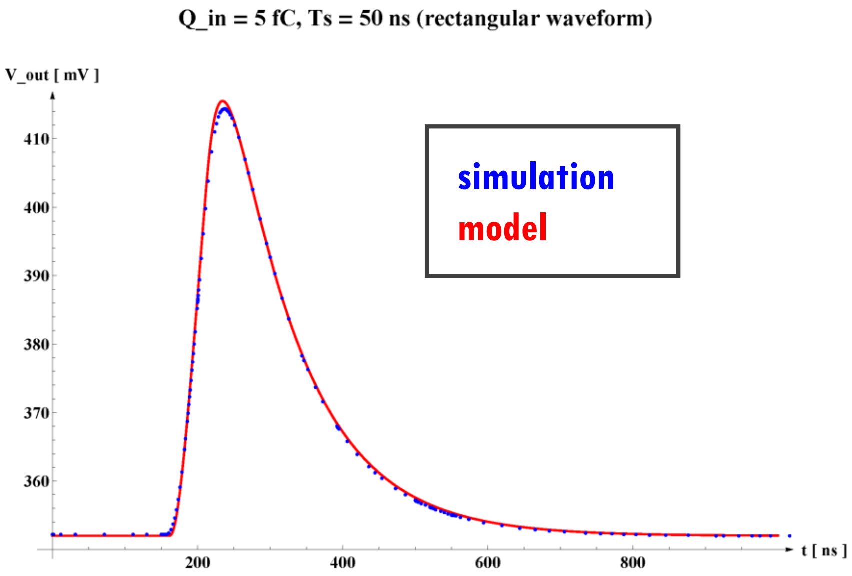
$$V_{out\_T}(t) = L^{-1}\{I_{in}(s) \cdot T(s)\} = V_{bsln} + 2000 \cdot I_0$$
$$((1 - 0.00545785 e^{0.333333(a-t)} + 0.294118 e^{0.05(a-t)} - 1.28866 e^{0.01(a-t)})\theta(-a+t) +$$
$$-(1 - 0.00545785 e^{0.333333(b-t)} + 0.294118 e^{0.05(b-t)} - 1.28866 e^{0.01(b-t)})\theta(-b+t))$$

TIGER response (model vs simulation)

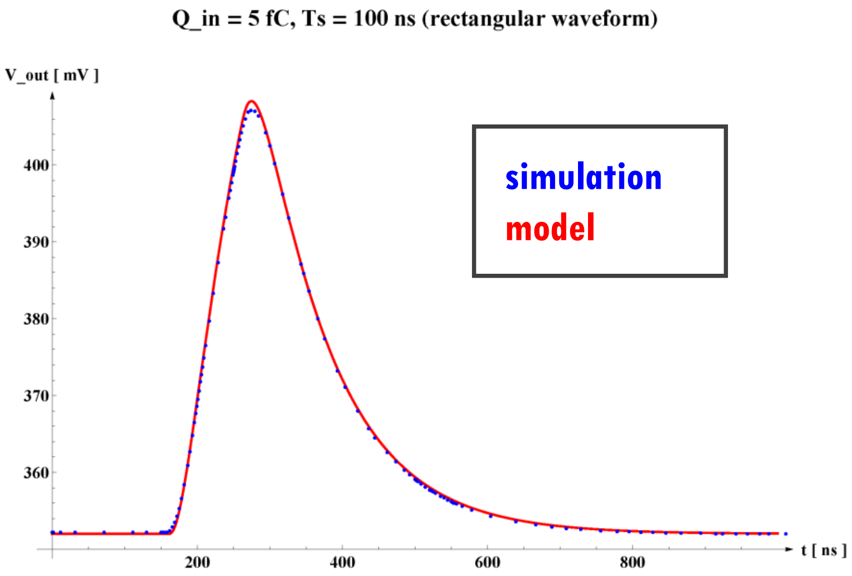
Compare T-branch shaper output for different Q_{in} and T_s:

- Circuit simulator data
- Model

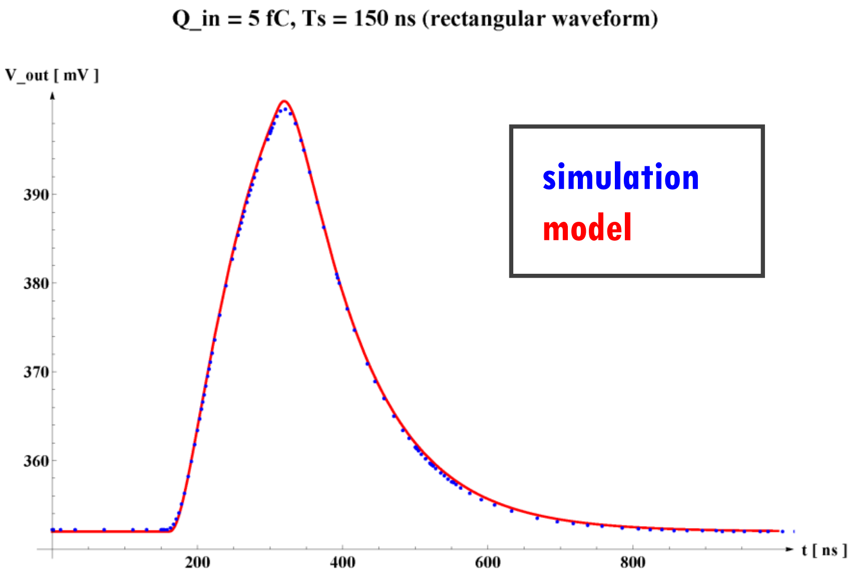
add 10ns delay to the model signals



rectangular waveform
$Q_{in} = 5$ fC
$T_s = 50$ ns



rectangular waveform
$Q_{in} = 5$ fC
$T_s = 100$ ns



rectangular waveform
$Q_{in} = 5$ fC
$T_s = 150$ ns

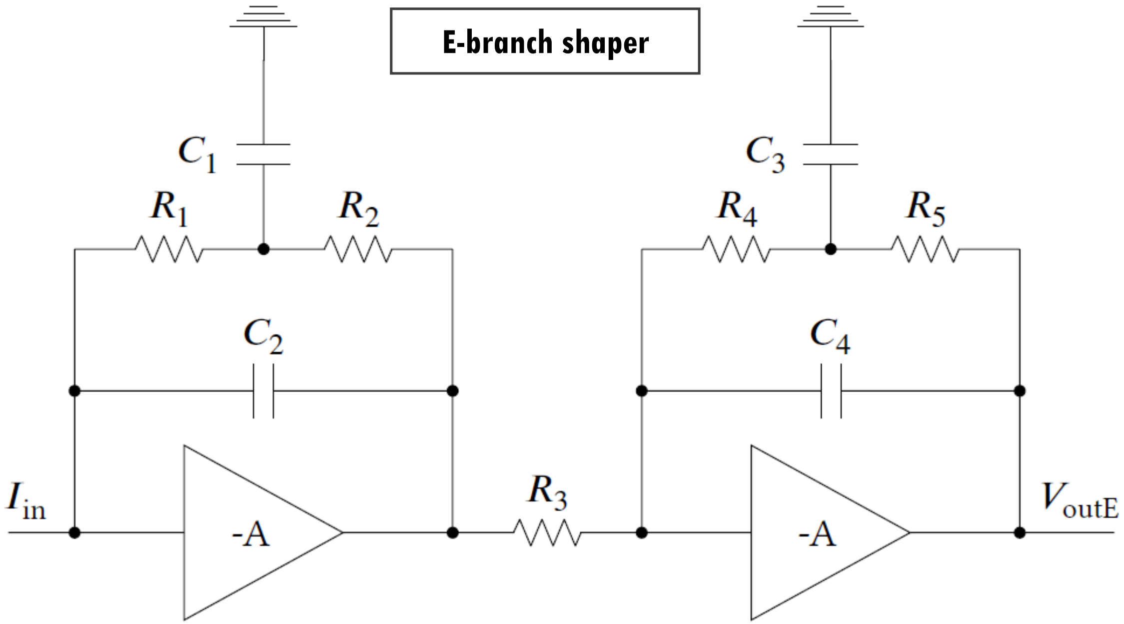
TIGER model (E-branch)

- The E-branch shaper has 2 stages with two complex conjugate poles for each stage
- The transfer function of each stage can be expressed as:

$$T(s) = \frac{G \cdot \omega_0^2}{s^2 + \frac{s \cdot \omega_0}{Q} + \omega_0^2}$$

where:

- G = DC gain
- ω_0 = natural angular frequency
- Q = quality factor



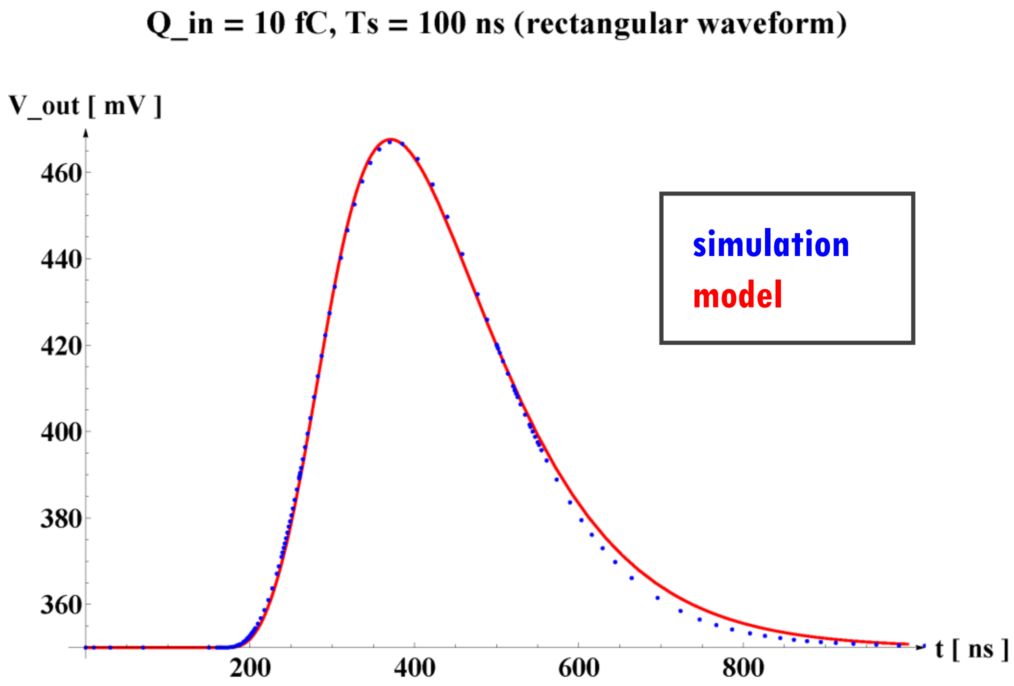
TIGER model (E-branch)

- The overall E-branch transfer function can be approximated as:

$$T(s) = \frac{C_z}{C_f} \cdot \left(\frac{G \cdot \omega_0^2}{s^2 + \frac{s \cdot \omega_0}{Q} + \omega_0^2} \right)^2$$

where:

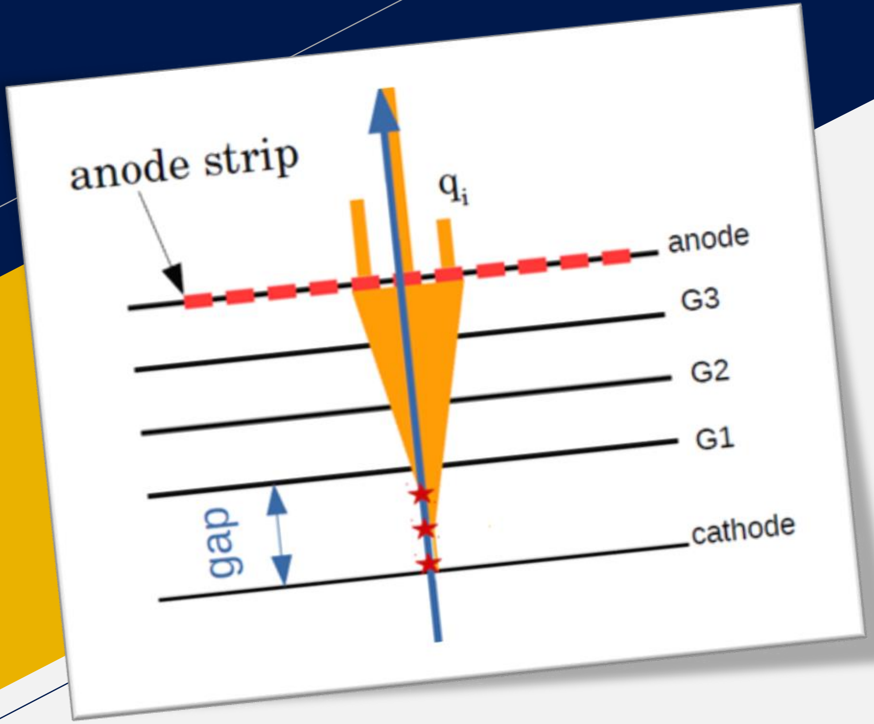
- $G = 12.7$
- $\omega_0 = 0.021$
- $Q = 0.42$
- $C_z = 20\ C_f$



add 5ns delay to the model signals

TIGER Front-End model summary

- A first draft of a model to simulate the TIGER response in order to be integrated in the CGEM-IT simulation code has been developed
- The model:
  - ✓ takes into account the different duration and shape of input signals
  - ✓ well reproduces the T-branch shaper output
  - ✓ provides a good approximation for the E-branch shaper output
  - ✓ validated with “real” CGEM signals
  - ✓ faster than computer circuit simulator (requires *Laplace Transform* evaluation)
- To-Do:
  - ❑ Take into account the saturation of the front-end (signals > 50 fC will have a different response)



Analogue readout concept

Time and charge measurements principle of operation

Time measurement

3 accuracy levels:

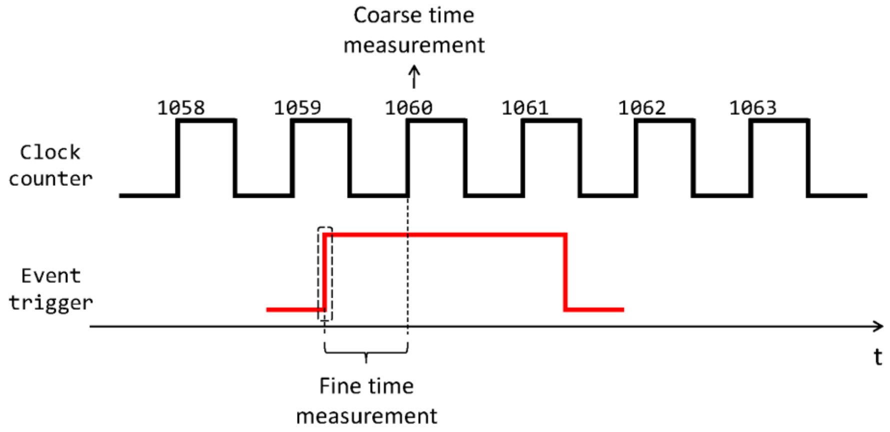
1. **Frame-word**
   - ~200 us time resolution
   - chip data (frame count, common for the 64 channels)
2. **Coarse counter**
   - 6.25 ns time binning (160 MHz)
   - channel data (Tcoarse)
3. **Fine counter**
   - <100 ps time resolution
   - channel data (Tfine)

EVENT WORD

K28.1	0b10	ch_id 6 bits	TAC	Tcoarse 16 bits	Ecoarse 10 bits	Tfine 10 bits	Efine 10 bits

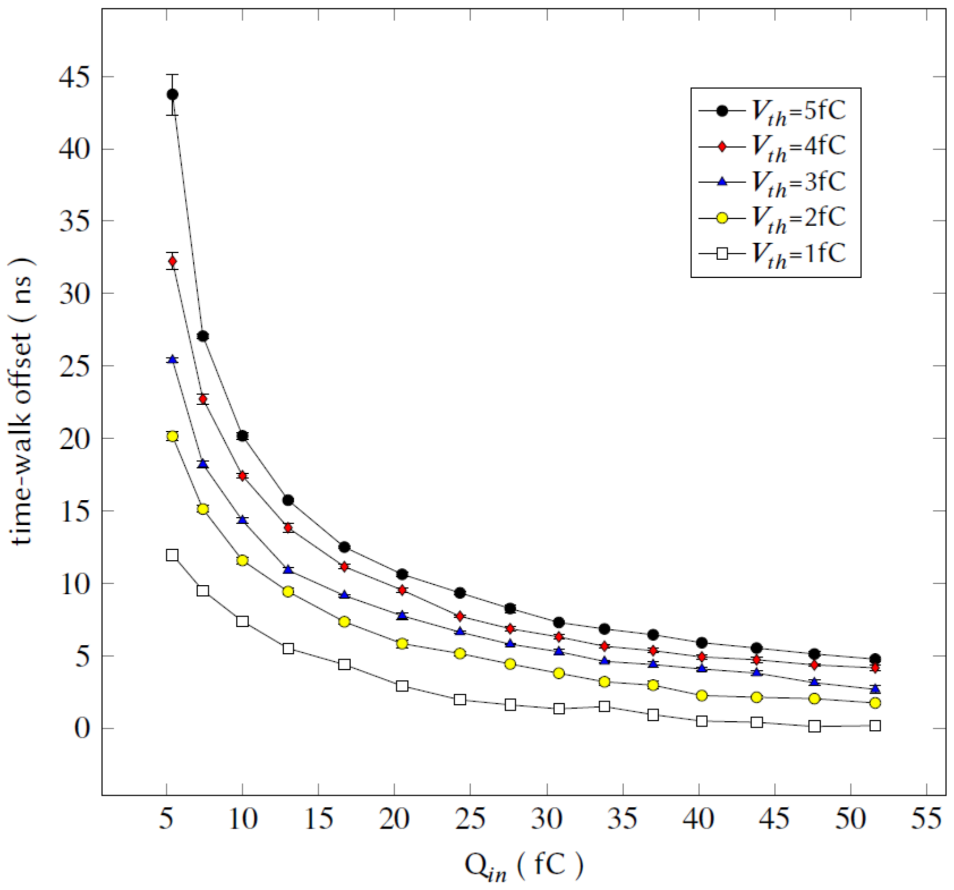
FRAME WORD

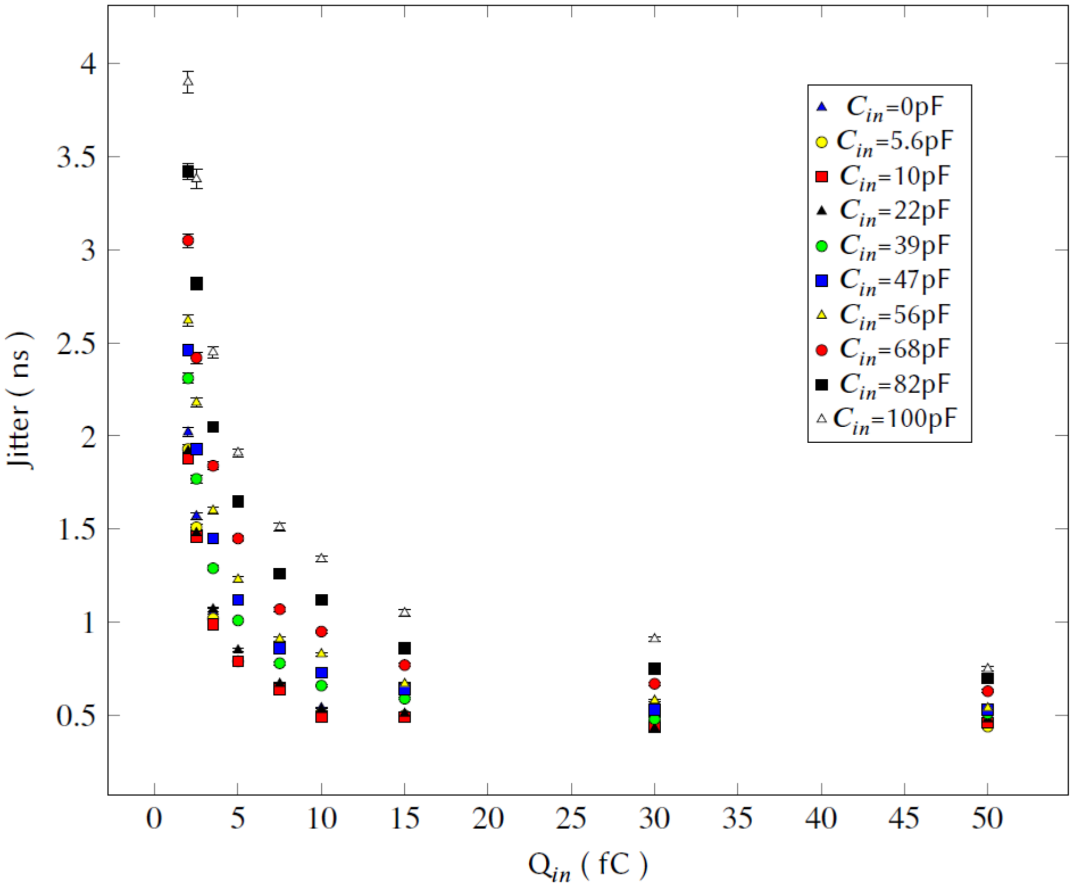
K28.1	0x00	reserved	frame count 16 bits	SEU count 15 bits



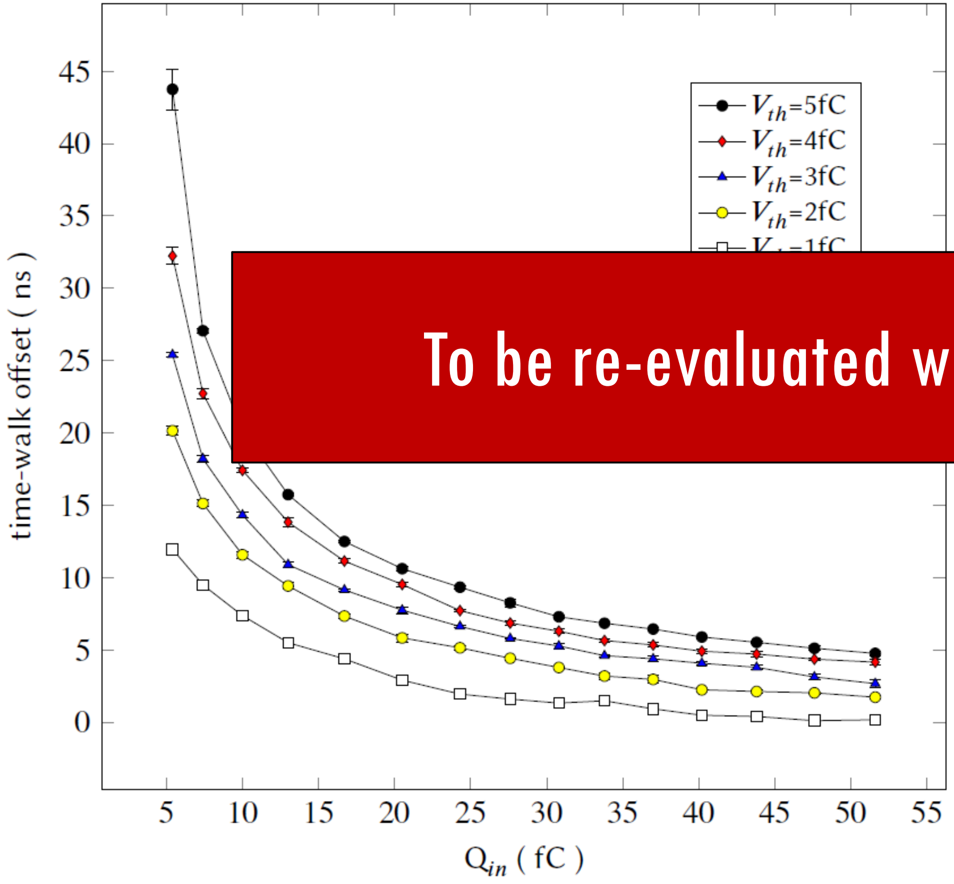
In ToT mode Ecoarse and Efine have the same meaning of Tcoarse and Tfine but for the falling edge

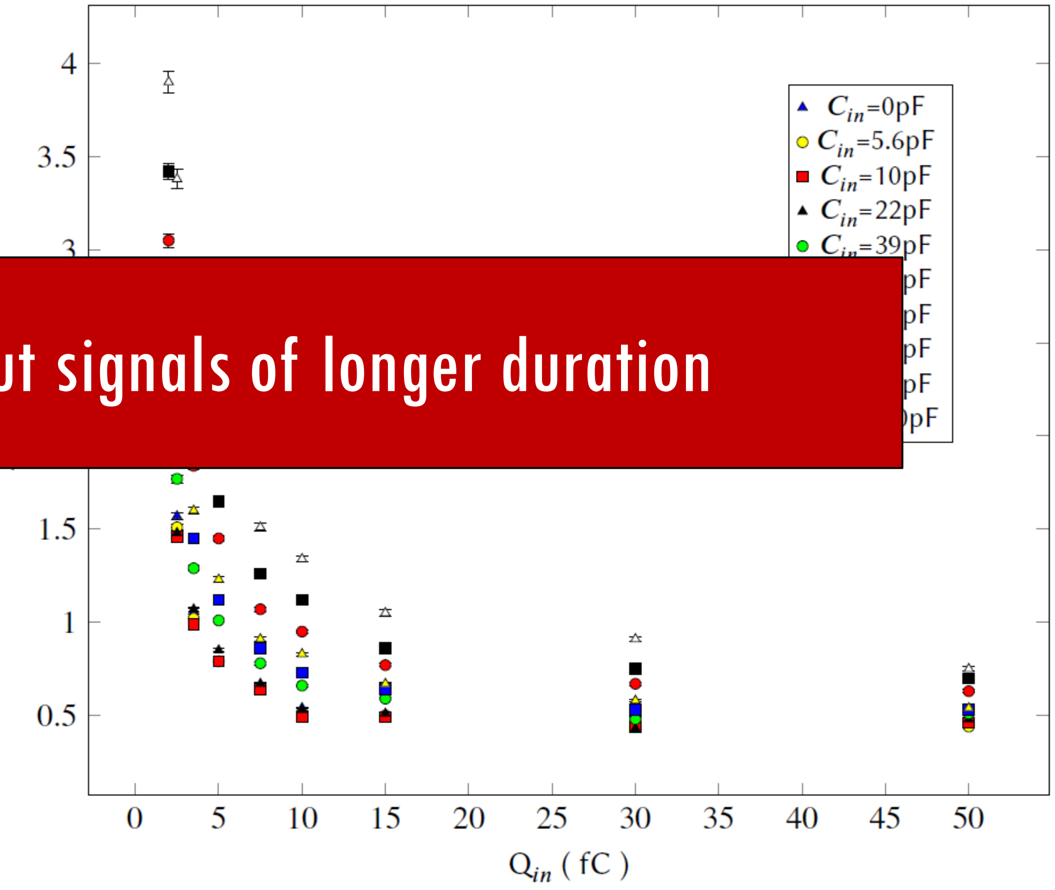
Time-walk and jitter





Time-walk and jitter





To be re-evaluated with input signals of longer duration

Time-walk and jitter

- Use CGEM simulated signals to evaluate TIGER front-end timing performance
  - **Time-walk** depends only on the applied threshold
  - **Jitter** can be estimated with the formula:

$$\boxed{\sigma_t = \frac{\sigma_v}{\frac{dV}{dt}}}$$

where:

$$\begin{cases} \sigma_v & = \text{noise measured at the T-branch shaper output} \\ \frac{dV}{dt} & = \text{T-branch shaper output slew rate around the threshold} \end{cases}$$

Charge measurement

Two mutually exclusive* modes:

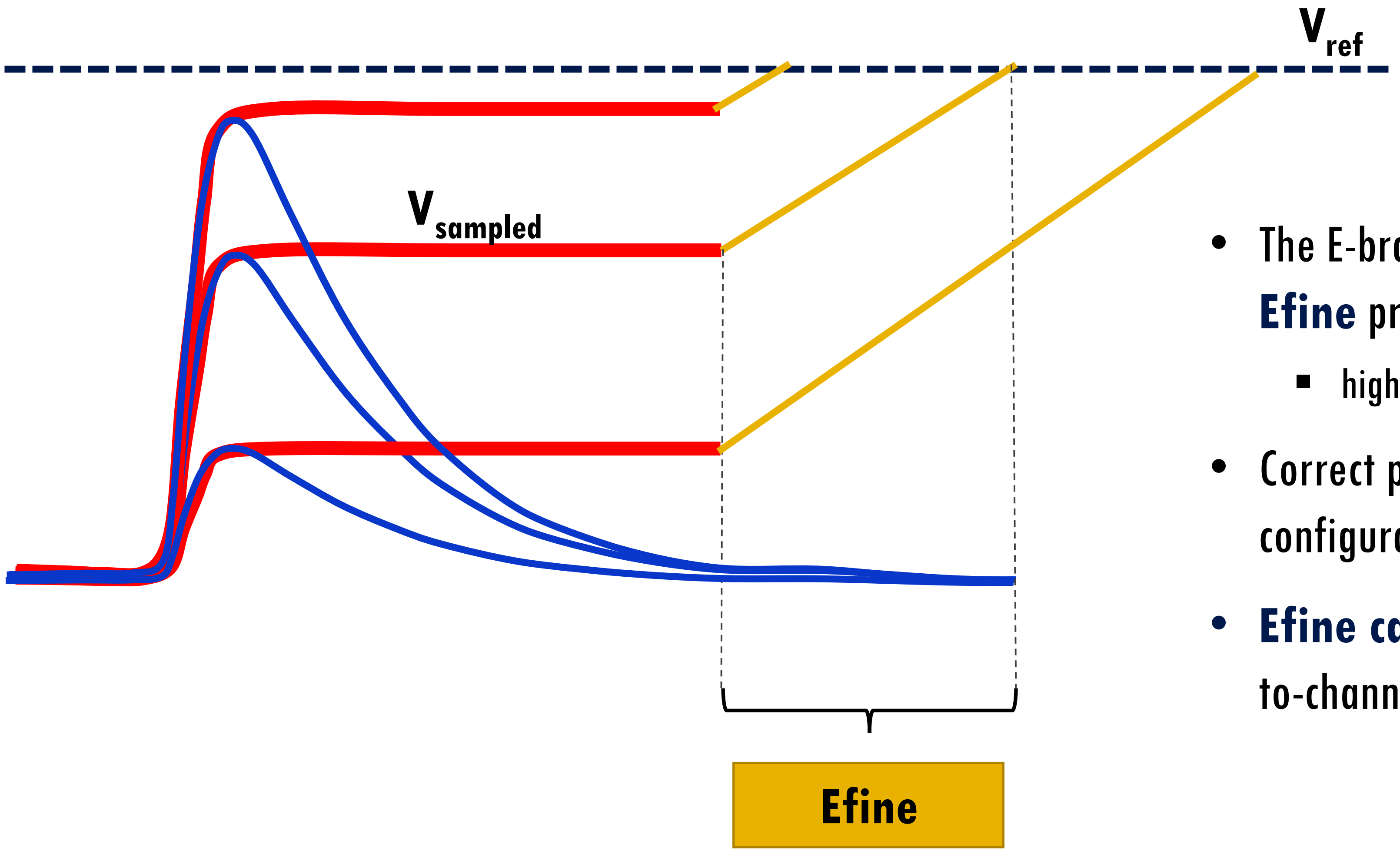
1. **QDC mode** (qdc_mode = 1, integ_en = 1)
2. **ToT mode** (qdc_mode = 0, integ_en = 0)

ToT is only a backup solution

- To check which charge measurement mode was used in a run, one can check the value of *Ecoarse – Tcoarse_10bit*
  - if it is almost constant ***QDC mode*** was selected
  - otherwise ***ToT mode*** was selected and this value is the Time-over-Threshold measurement

* The charge measurement mode is set when the chips are configured before running the acquisition

Charge measurement (S&H mode)

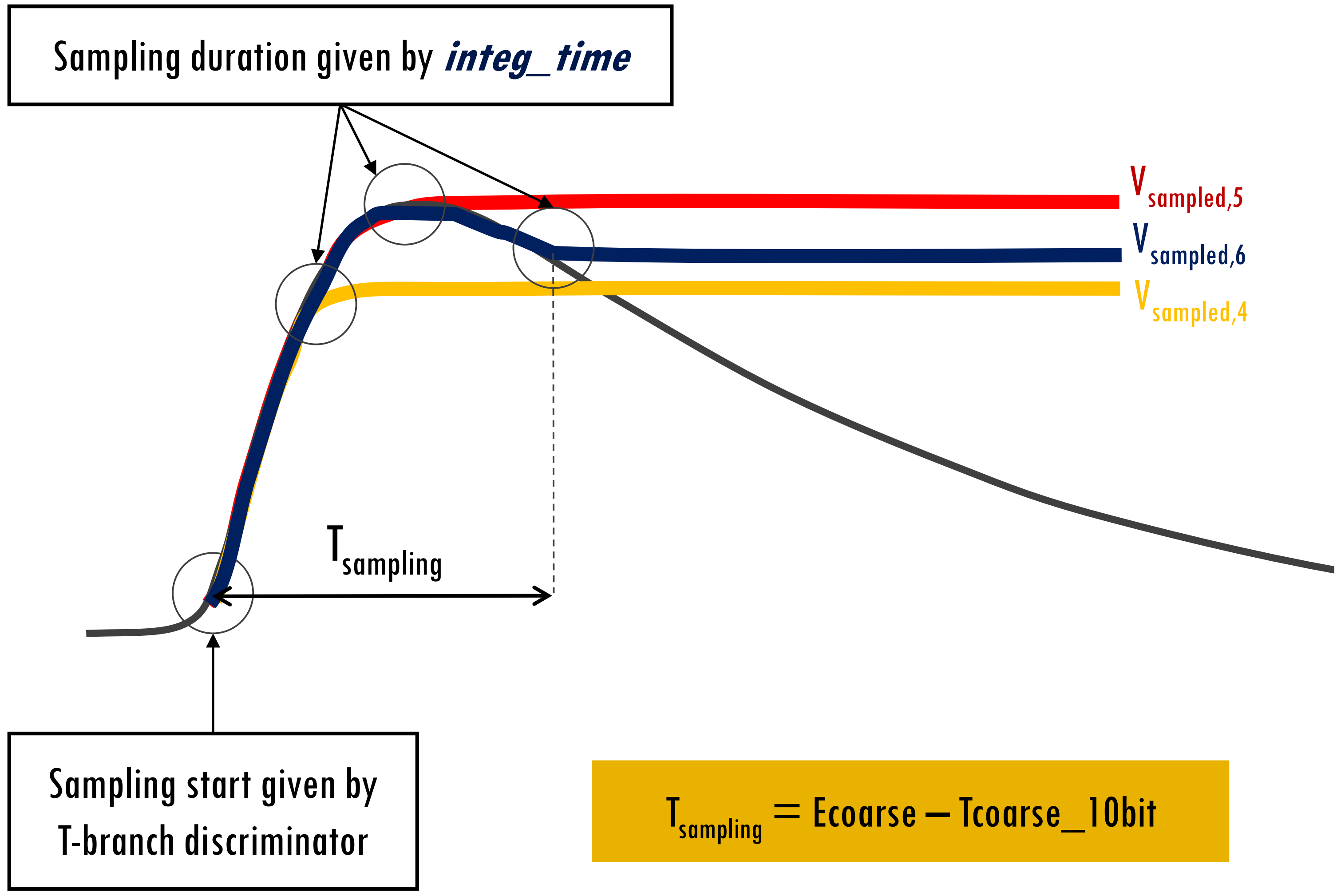


S&H saturation when $V_{sampled} > V_{ref}$

- The E-branch shaper output is sampled and digitized: **Efine** provides the information about the charge
  - high Efine means low charge and vice-versa
- Correct peak sampling set by ***integ_time*** channel configuration register
- **Efine calibration curves** to compensate channel-to-channel and chip-to-chip offsets

Sampling time

$$T_{sampling} = 4 \cdot 6.25ns \cdot (integ\_time + 1)$$

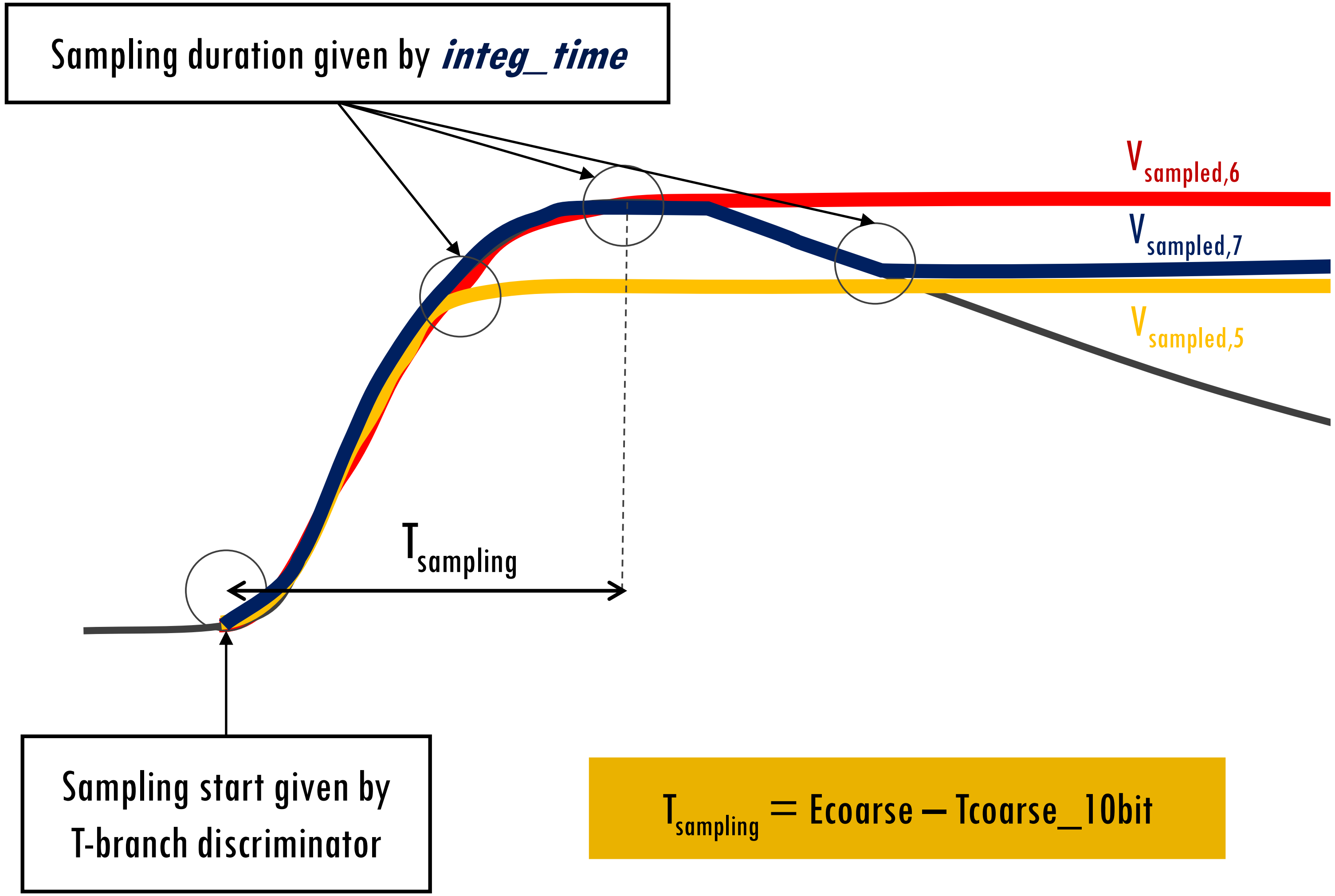


$T_{sampling}$ = Ecoarse – Tcoarse_10bit

- S&H peak detector configured and calibrated for a 170 ns peaking time (***integ_time*** = **5**)
- If signals last longer than the design specification (50 ns) the sampling time (***integ_time***) must be increased → 4 clock cycles steps (= 25 ns)
- **Sampling start** is given by the T-branch, due to its better timing performance

Sampling time

$$T_{sampling} = 4 \cdot 6.25ns \cdot (integ\_time + 1)$$

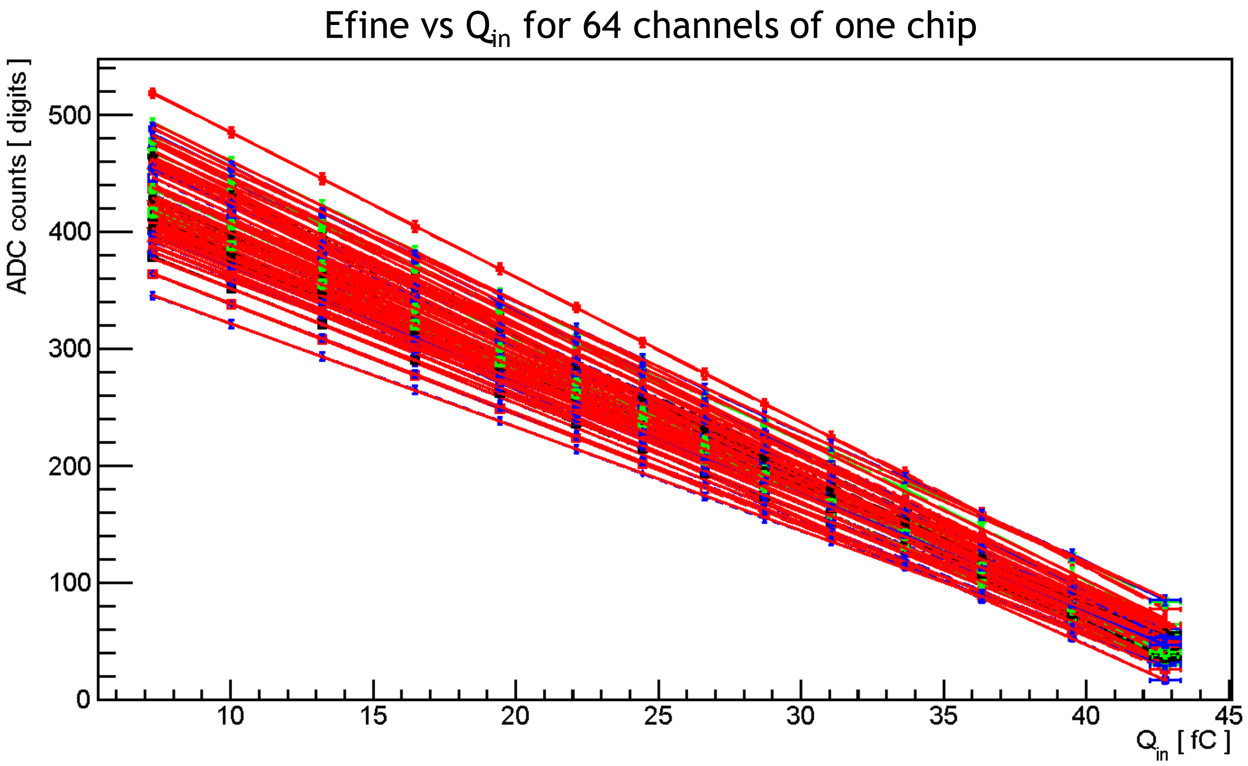


$$T_{sampling} = Ecoarse - Tcoarse\_10bit$$

- S&H peak detector configured and calibrated for a 170 ns peaking time (***integ_time* = 5**)
- If signals last longer than the design specification (50 ns) the sampling time (***integ_time***) must be increased → 4 clock cycles steps (= 25 ns)
- **Sampling start** is given by the T-branch, due to its better timing performance

S&H calibration

Efine calibration curves (linear response) to compensate channel-to-channel and chip-to-chip offsets

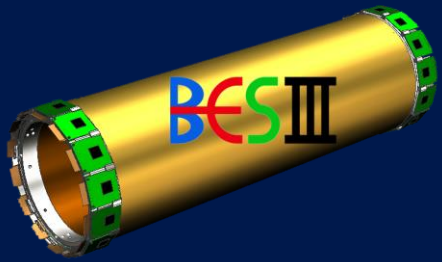


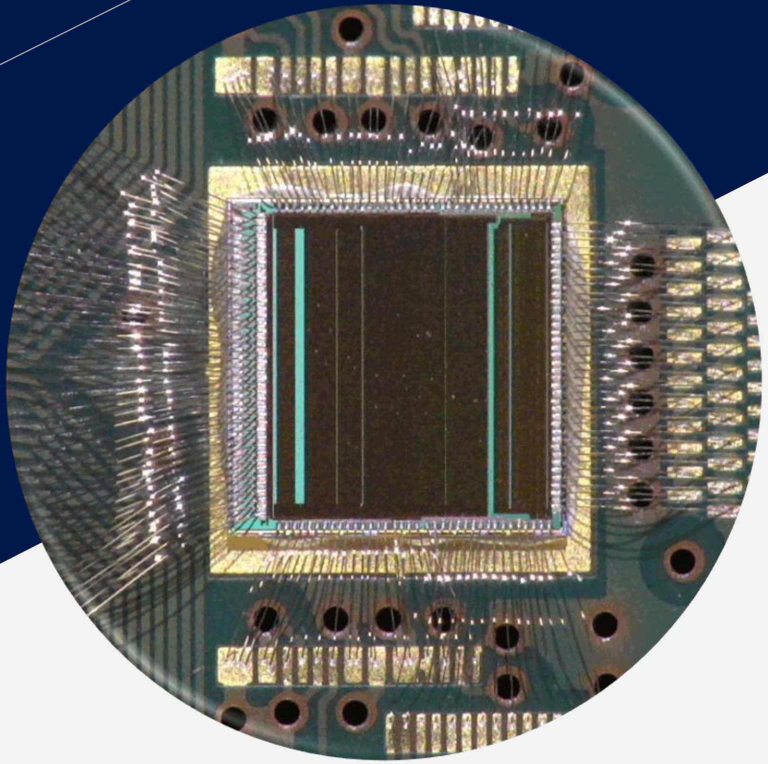
Look-Up-Table (LUT)

- FEB_id
- CHIP_id
- CHANNEL_id
- fit SLOPE parameter
- fit CONSTANT parameter

Threshold calibration

- Threshold scans are performed on both branches of all channels to correct channel-by-channel and chip-by-chip offsets and mismatches
- Equalized threshold values are stored in LUT which are used to configure each channel before the acquisition starts
  - Threshold scans allow also to evaluate the noise for each channel ($\sigma =$ noise)
  - This information is used to set the threshold at a given value (e.g. 3σ) above the baseline in order to equalize the channels noise rate (few kHz)





Backup slides

Fabio Cossio

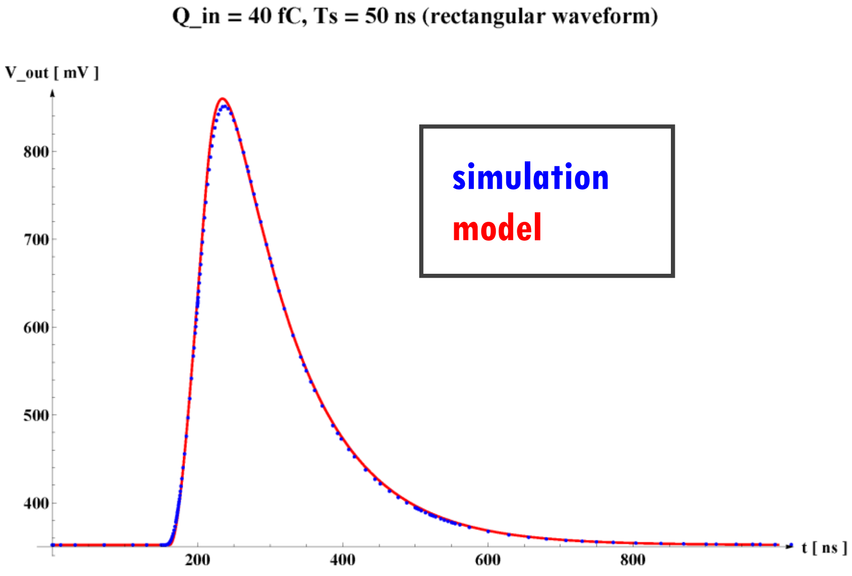
fabio.cossio@to.infn.it

INFN Torino

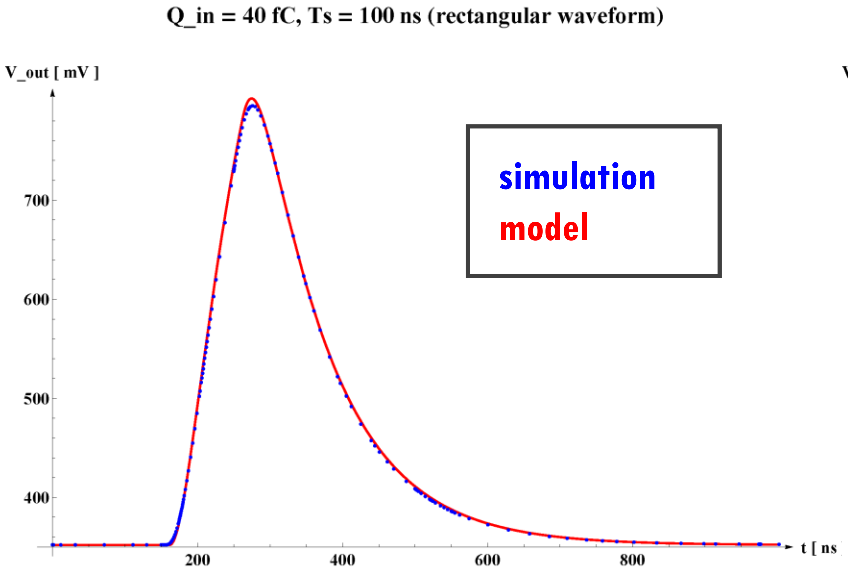
TIGER response (model vs simulation)

Compare T-branch shaper output for different Q_{in} and T_s:

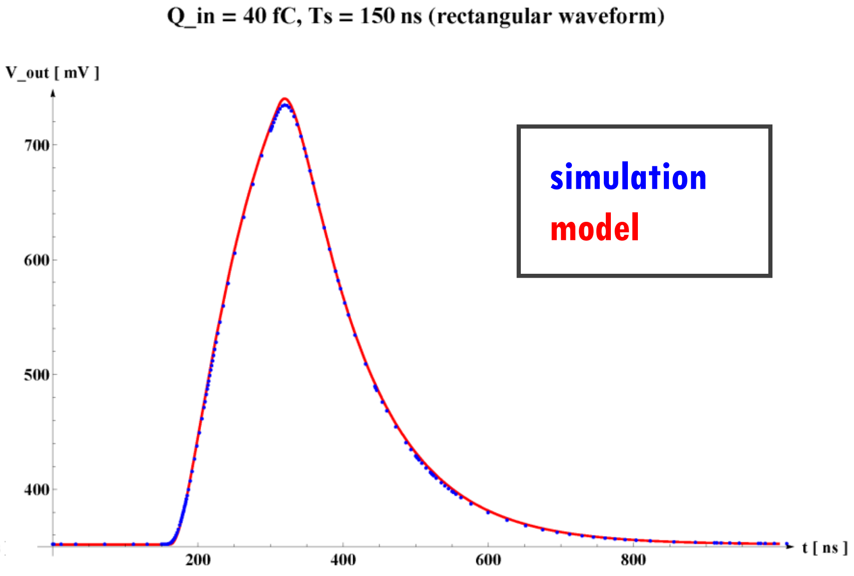
- Circuit simulator data
- Model



rectangular waveform
$Q_{in} = 40$ fC
$T_s = 50$ ns



rectangular waveform
$Q_{in} = 40$ fC
$T_s = 100$ ns

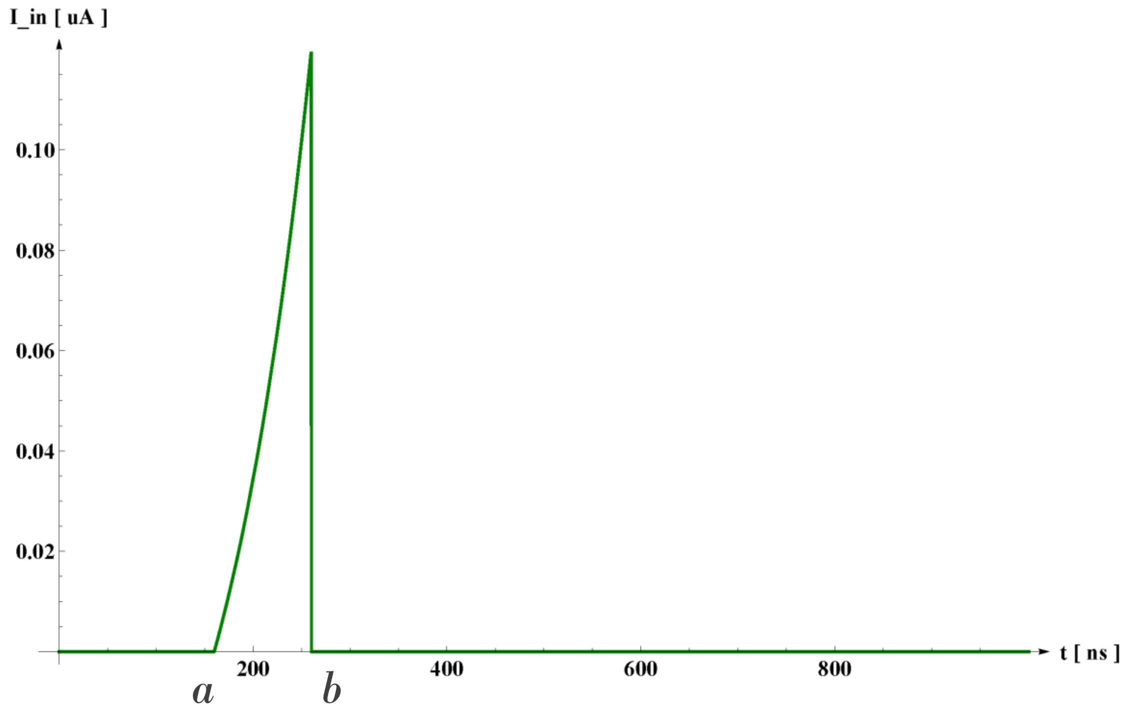


rectangular waveform
$Q_{in} = 40$ fC
$T_s = 150$ ns

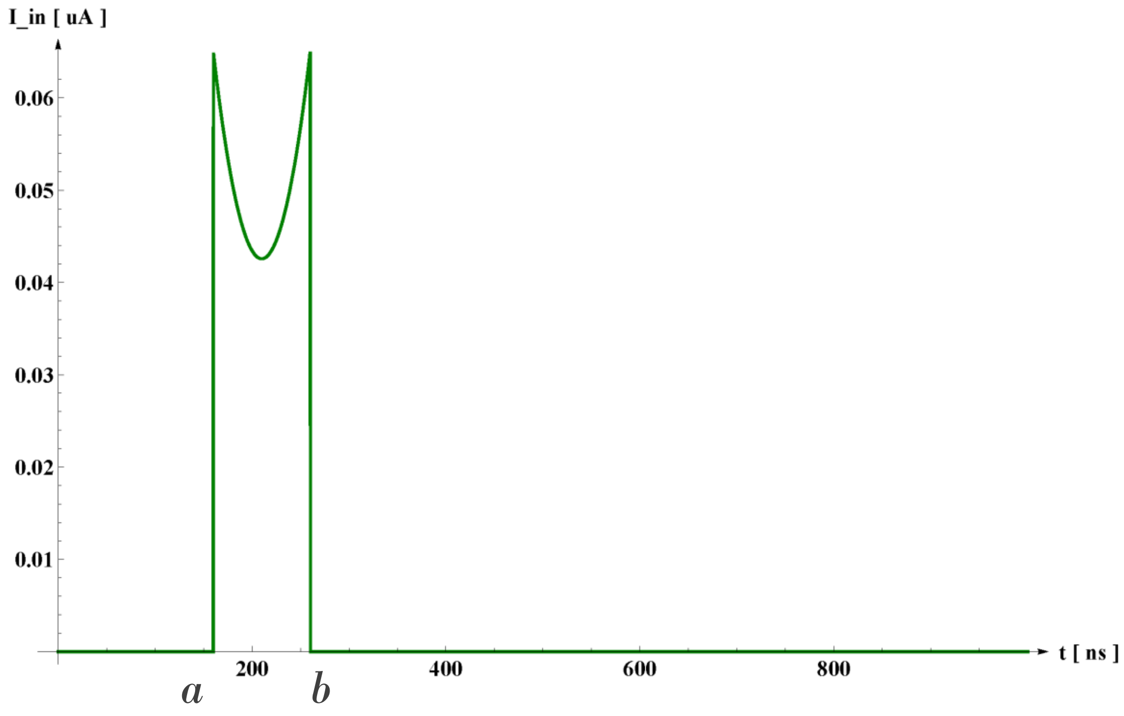
TIGER model (T-branch)

Repeat including also input signals of different shapes

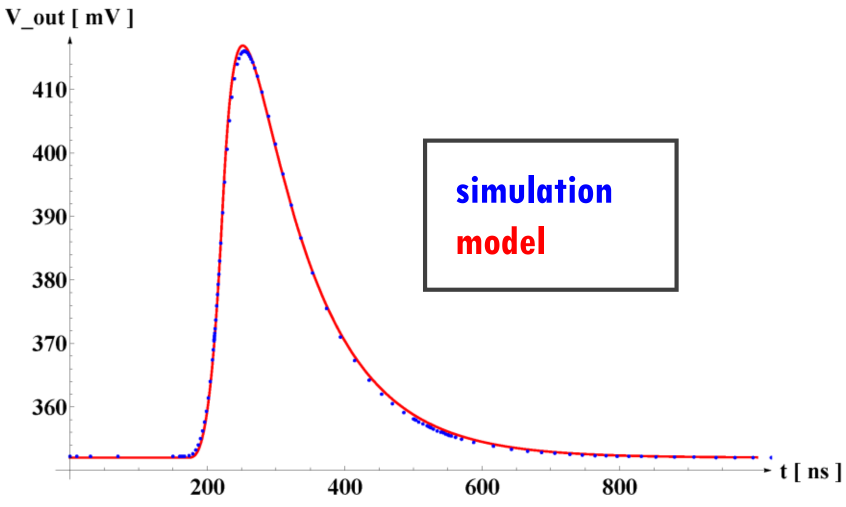
Q_in = 5 fC, Ts = 100 ns (exponential waveform)



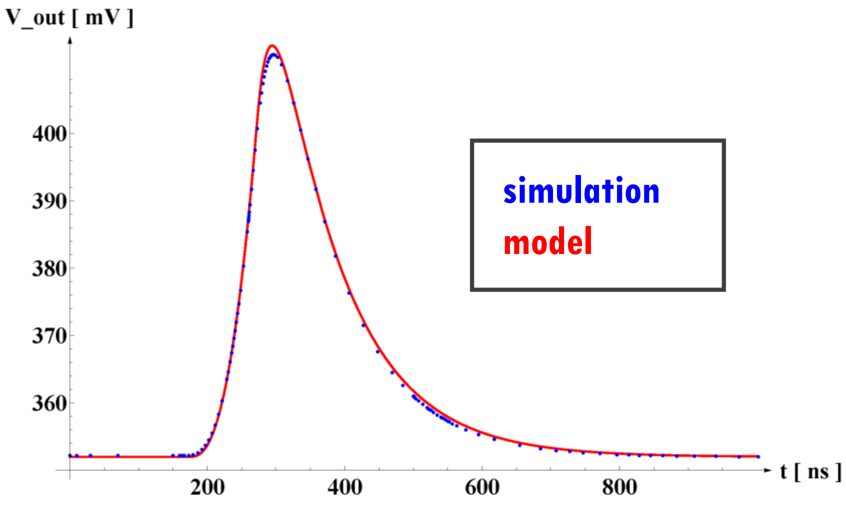
Q_in = 5 fC, Ts = 100 ns (double exponential waveform)



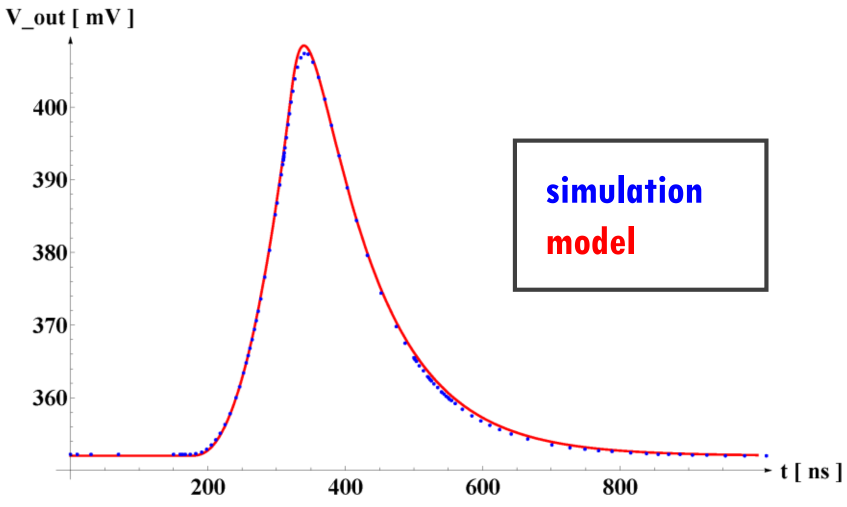
Q_in = 5 fC, Ts = 50 ns (exponential waveform)



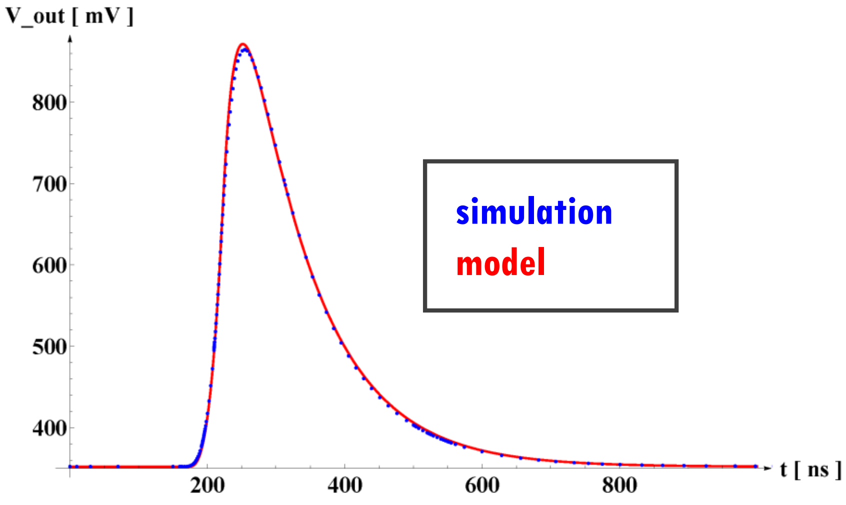
Q_in = 5 fC, Ts = 100 ns (exponential waveform)



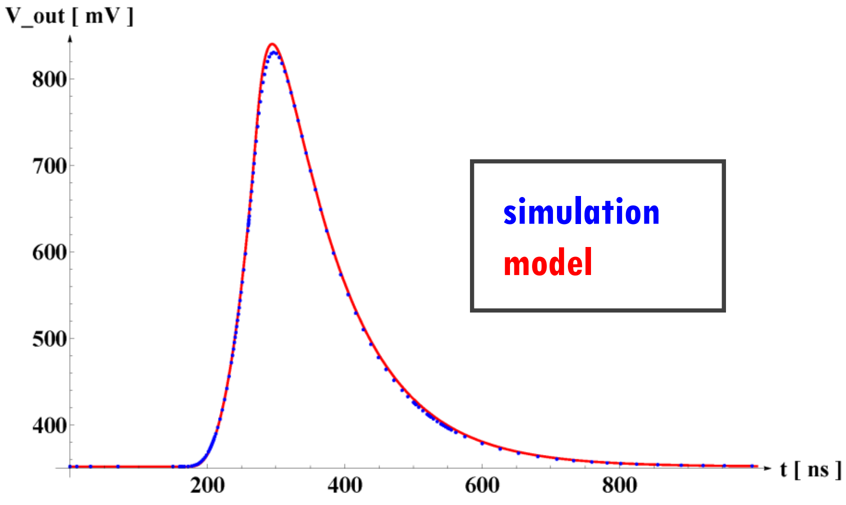
Q_in = 5 fC, Ts = 150 ns (exponential waveform)



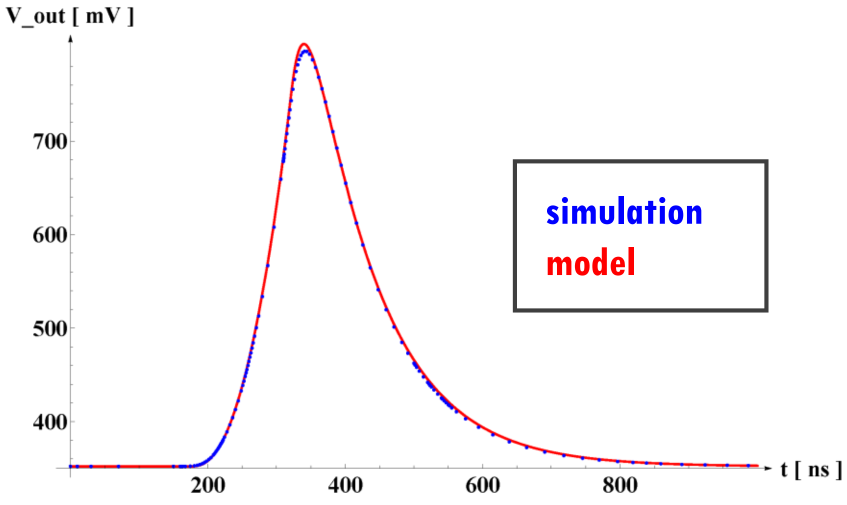
Q_in = 40 fC, Ts = 50 ns (exponential waveform)



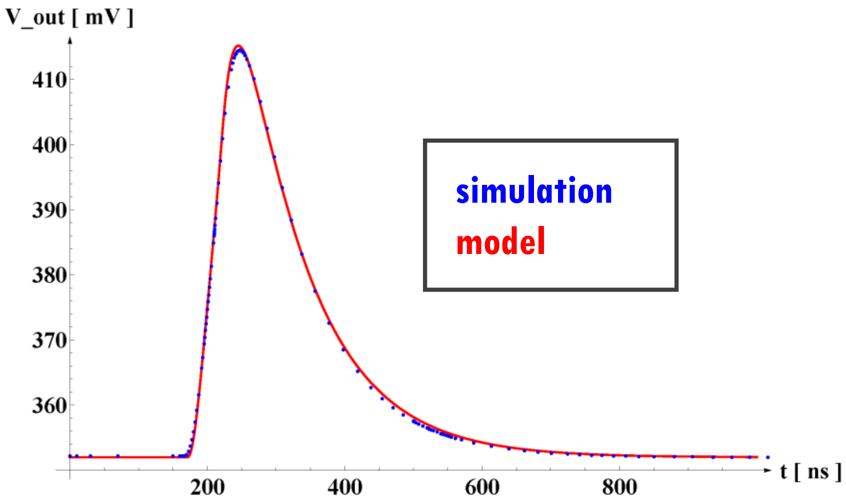
Q_in = 40 fC, Ts = 100 ns (exponential waveform)



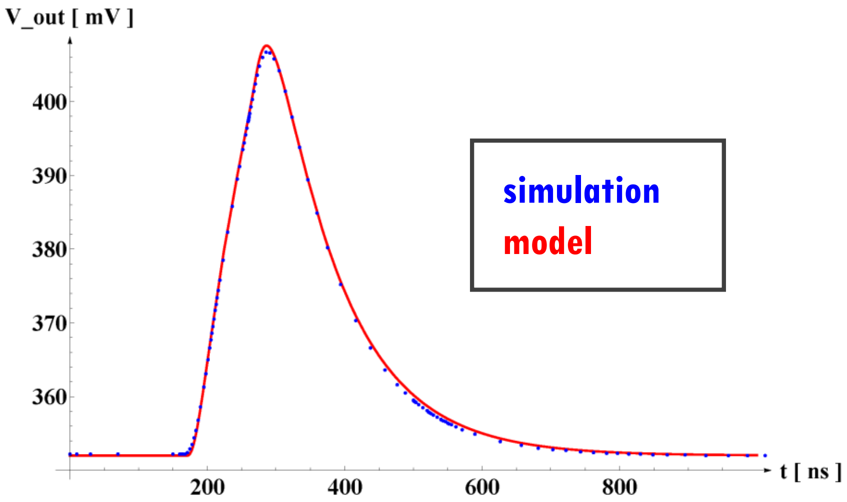
Q_in = 40 fC, Ts = 150 ns (exponential waveform)



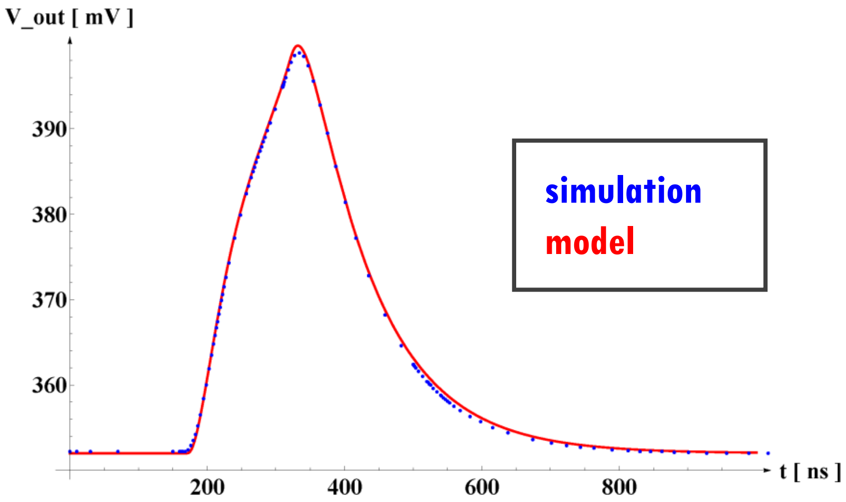
Q_in = 5 fC, Ts = 50 ns (double waveform)



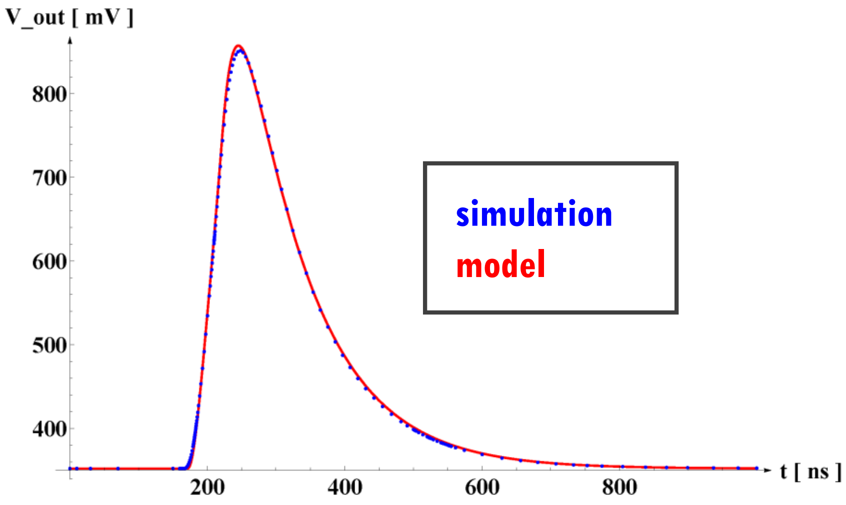
Q_in = 5 fC, Ts = 100 ns (double waveform)



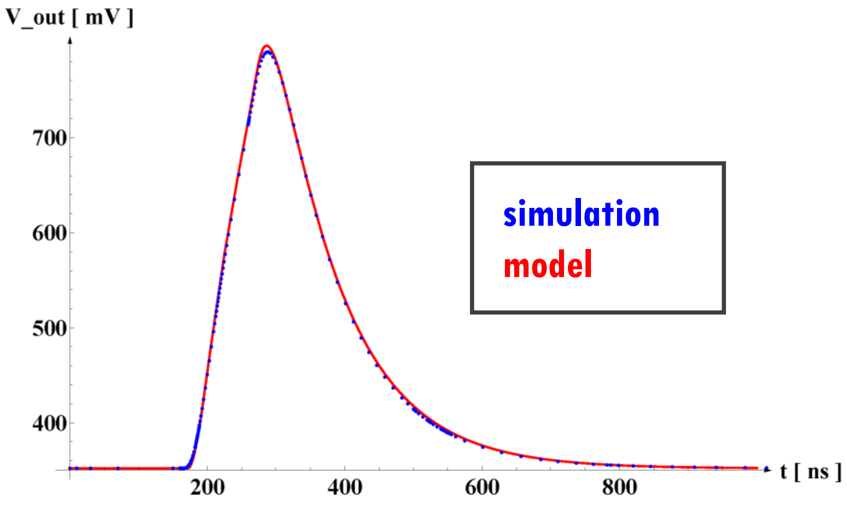
Q_in = 5 fC, Ts = 150 ns (double waveform)



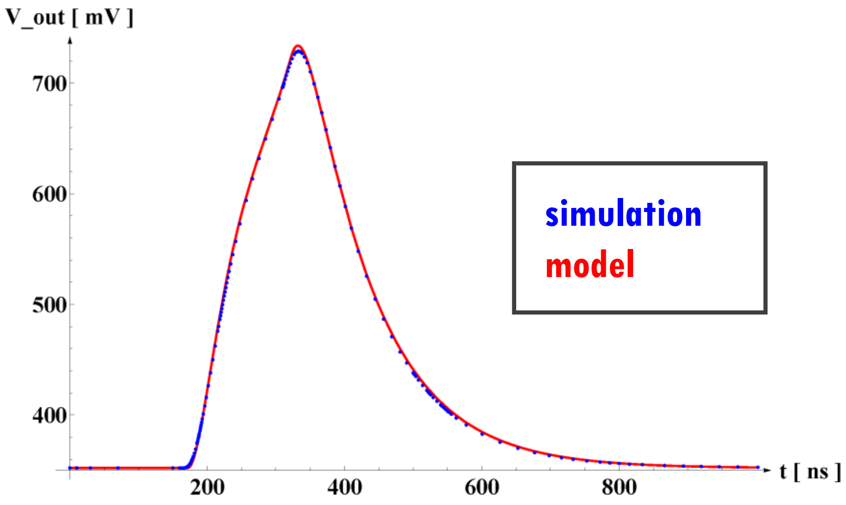
Q_in = 40 fC, Ts = 50 ns (double waveform)



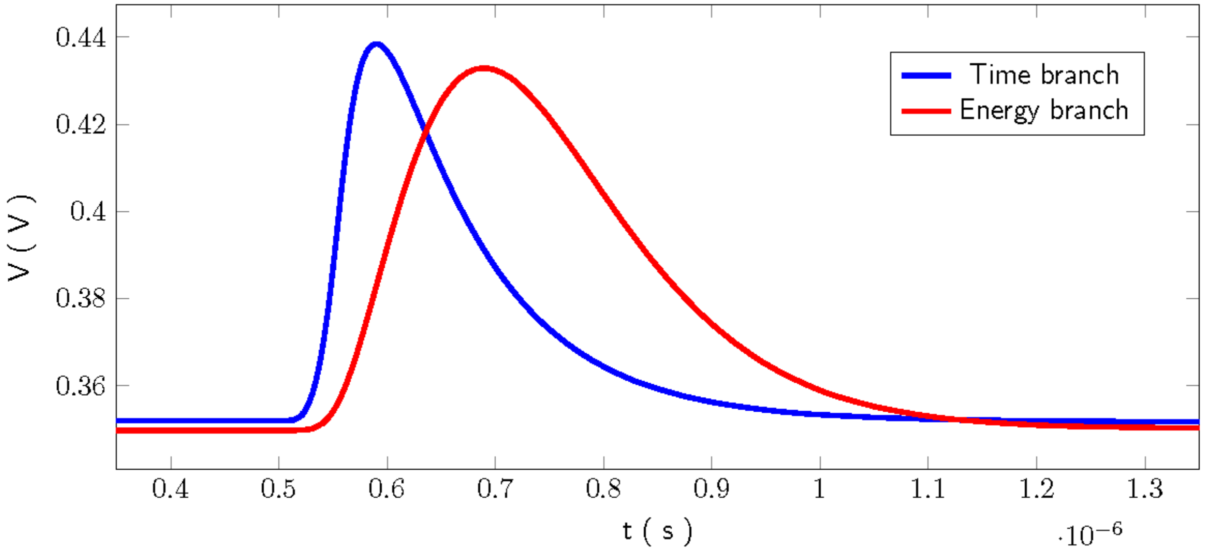
Q_in = 40 fC, Ts = 100 ns (double waveform)



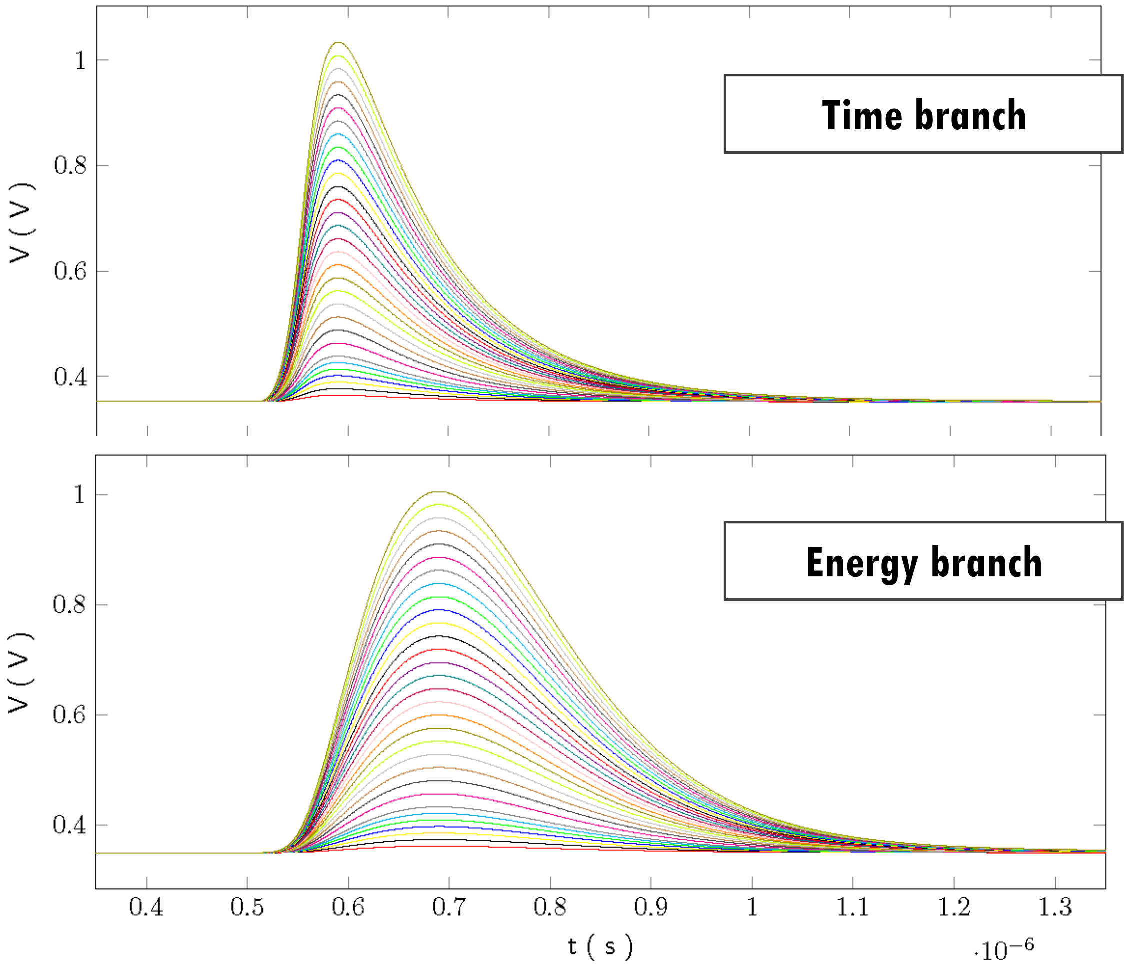
Q_in = 40 fC, Ts = 150 ns (double waveform)



Front-End response



Front-end response to a 7 fC input signal



1-50 fC input signal sweep

Time measurement

1. **Frame-word**
   - Used to reconstruct the global timestamp of each event
   - Sent off-chip with top-priority (common for the 64 channels)
   - 1 frame-word every 2^{15} clock cycles (2^{15} x 6.25 ns = 204.8 us)
   - 16-bit counter (roll-over every 13.4217728 s)
2. **Coarse counter**
   - Combine with frame-word information to reconstruct the global timestamp of each event
   - Data related to single channel-event
   - 1 coarse counter = 1 clock cycle (6.25 ns)
   - 16-bit counter (roll-over every 6.25 ns x 2^{16} = 409.6 us)

Time measurement

Frame-word + Coarse counter examples:

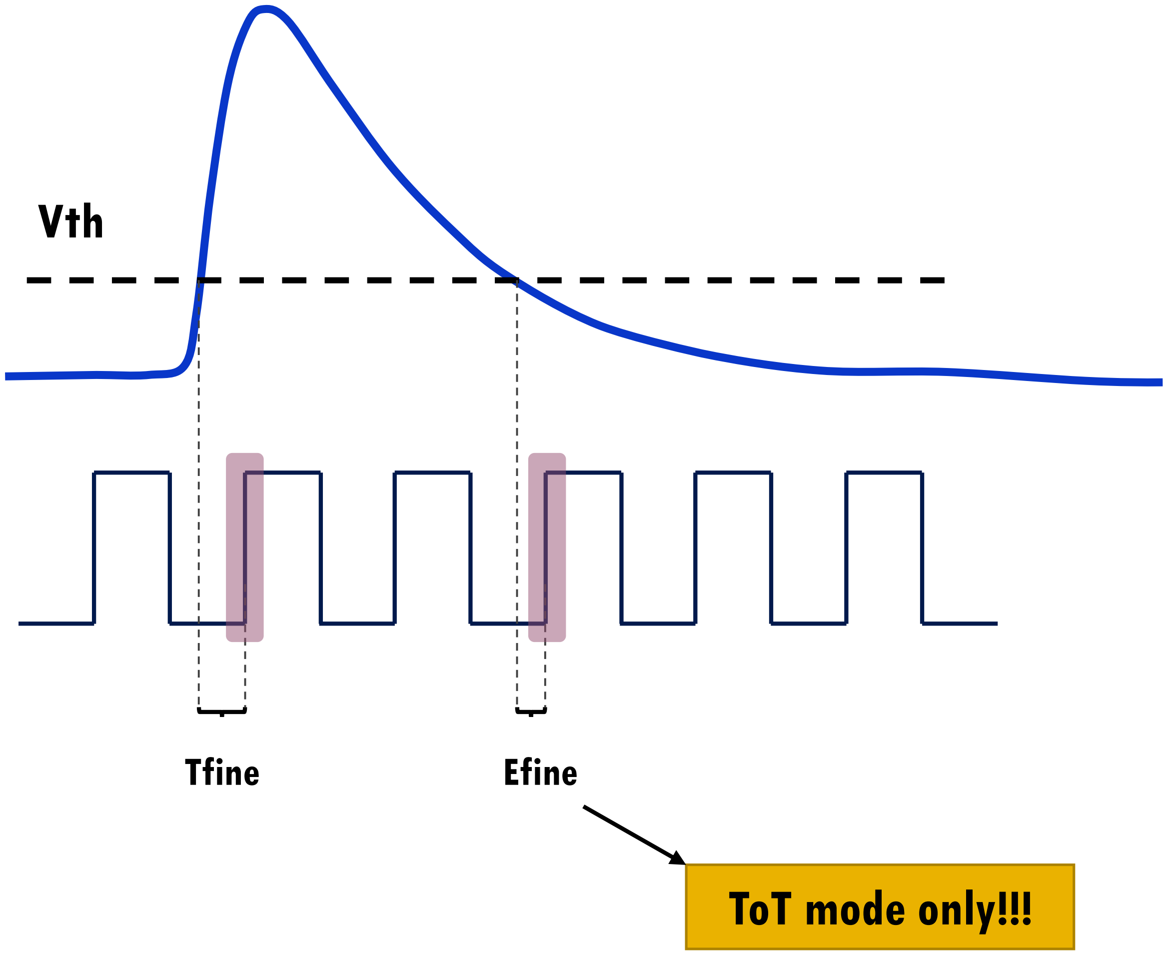
- Frame-word = 6546 → 6546 x 204.8 us = 1340.6208 ms
- Tcoarse = 13465 → 13465 x 6.25 ns = 84.15625 us

➢ Global timestamp = 1340.6208 ms + 84.15625 us = 1340.70495625 ms

- Frame-word = 654**7** → 6547 x 204.8 us = 1340.8256 ms
- Tcoarse = 43465 → $(43465 - 2^{15})$ x 6.25 ns = 66.85625 us

➢ Global timestamp = 1340.8256 ms + 66.85625 us = 1340.89245625 ms

Fine time correction



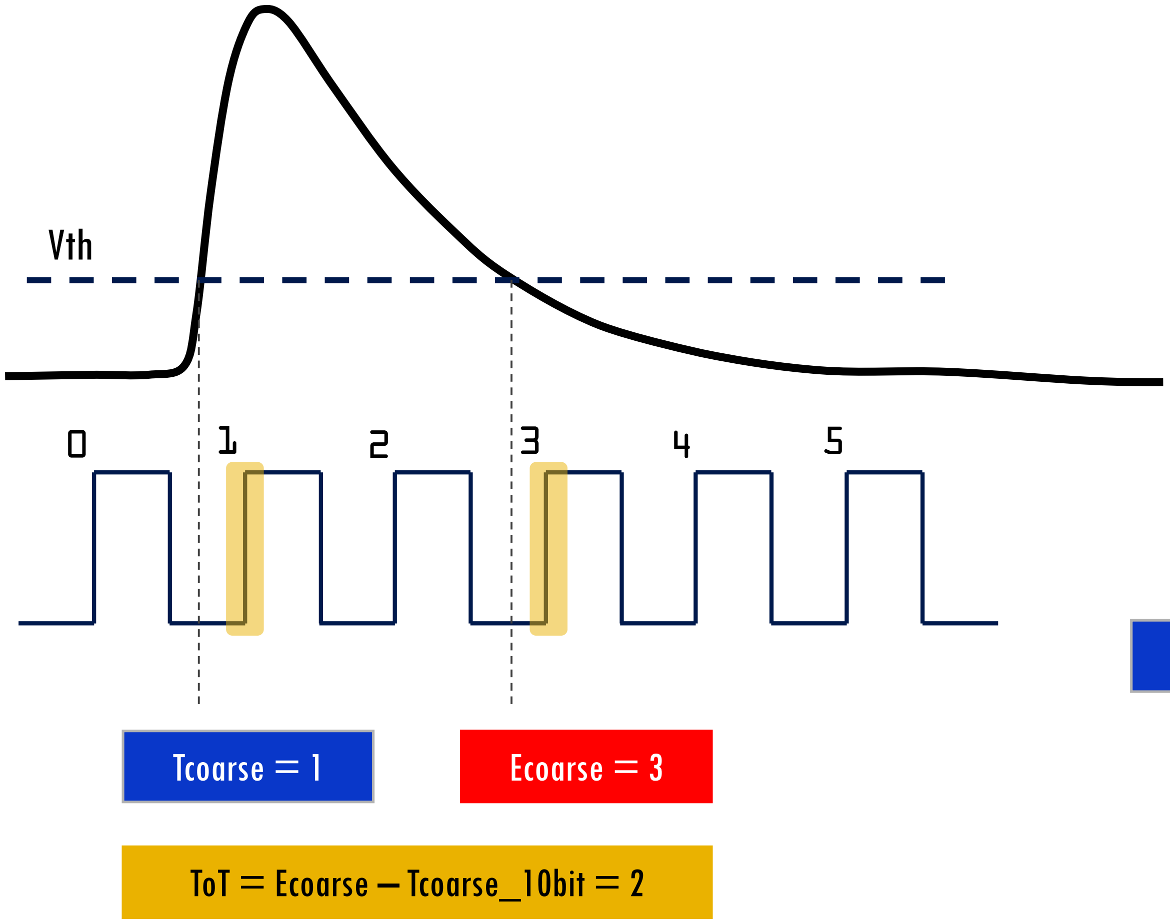
- Tfine counter: 10 bits
- Tfine calibration with a TDC TP phase scan
- This measurement is performed by the 4 TACs (Time-to-Amplitude Converter) of each TDC
- Only in ToT mode apply the same correction also with Efine

Fine time correction

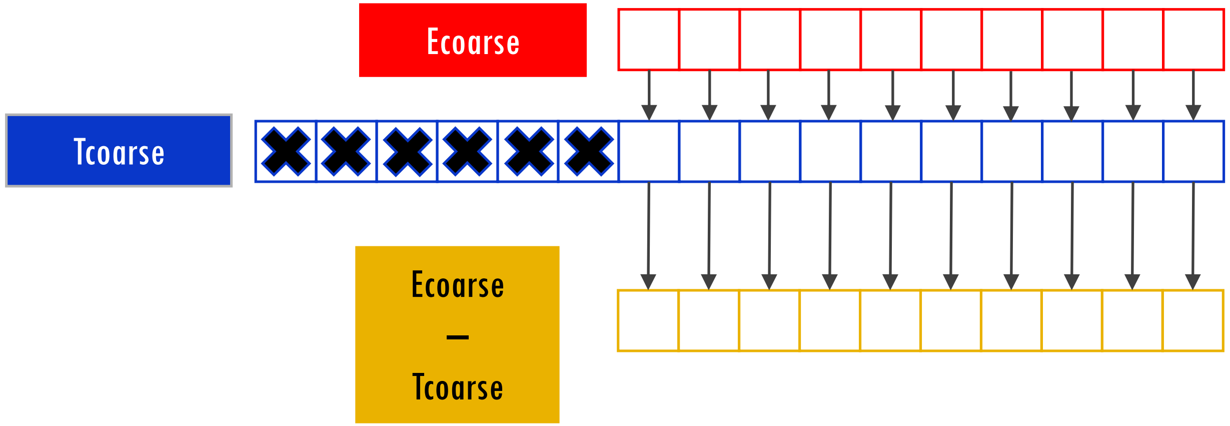
- Timestamp = frame_word + Tcoarse – **(Tfine – *offset*) · *time_bin***
- *offset and time_bin* provided by LUT (64 channels x 4 TACs gain-offsets compensation)
  - *offset* = Tfine_{MIN}
  - *time_bin* = 6.25 ns / (Tfine_{MAX} - Tfine_{MIN})

- The same correction can be applied also to Ecoarse (using Efine) but only when in ToT mode:
- Falling edge trigger time = frame_word + Ecoarse – **(Efine – *offset*) · *time_bin***
  - *offset* = Efine_{MIN}
  - *time_bin* = 6.25 ns / (Efine_{MAX} - Efine_{MIN})

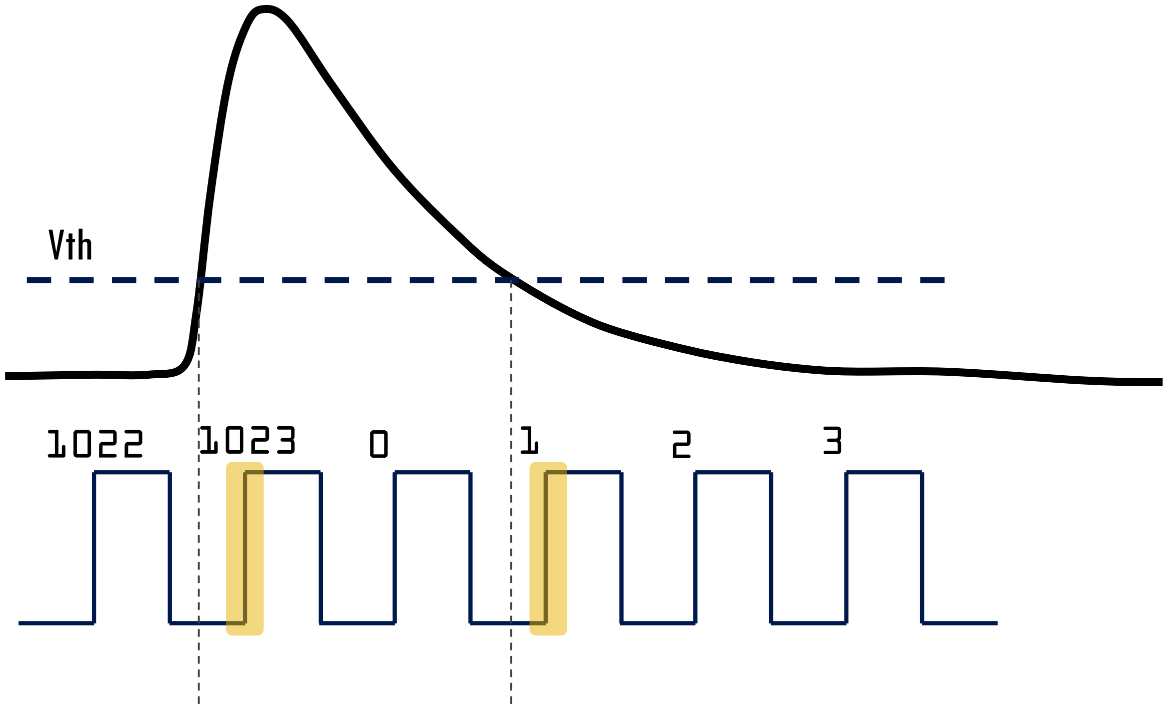
Time-over-Threshold measurement



- Tcoarse counter: 16 bits
- Ecoarse counter: 10 bits
- For this measurement use only 10 LSB of Tcoarse



Time-over-Threshold measurement



Tcoarse = 1023

Ecoarse = 1

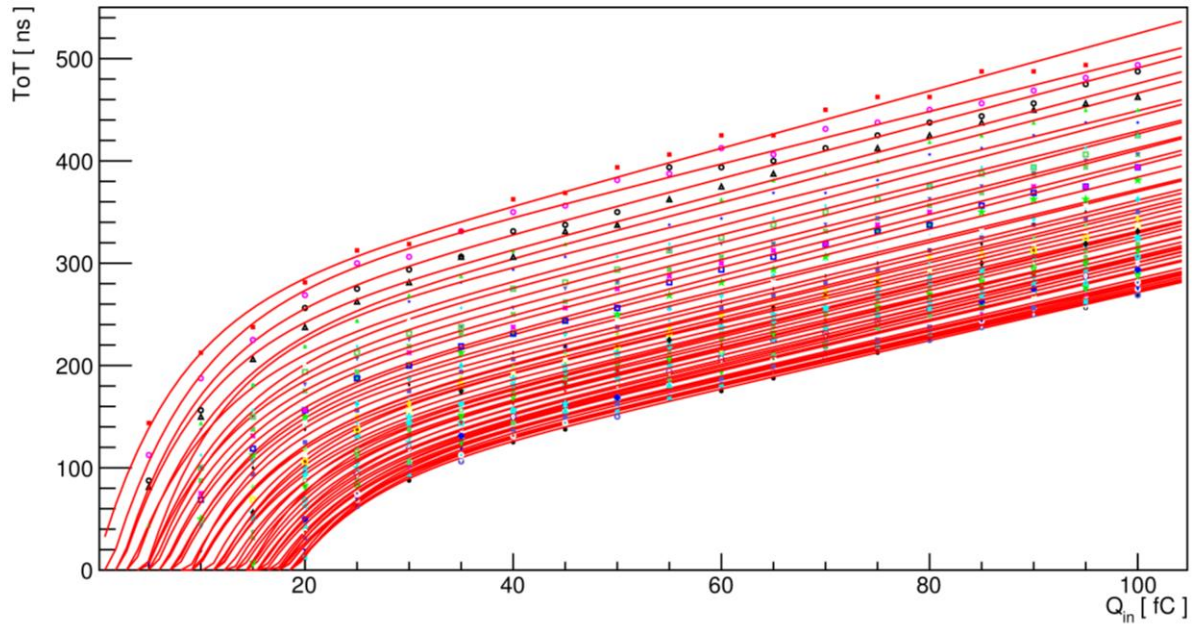
ToT = (Ecoarse + 1024) – Tcoarse_10bit = 2

- Tcoarse counter: 16 bits
- Ecoarse counter: 10 bits
- For this measurement use only 10 LSB of Tcoarse
- Check for counters roll-over (restart from 0 after it reaches 2^10 - 1 = 1023)

Time-over-Threshold calibration

Compensate channel-to-channel offsets due to different thresholds set (depends on acquisition settings) and transistor mismatches (fixed)

ToT curves for different V_{th}



- Response is not linear
- Biggest contribution to channel-to-channel offsets comes from **threshold settings** (not-fixed)
- Fit with the following function:

$$ToT = E \cdot e^{A \cdot Q_{in} + B} + C + D \cdot Q_{in}$$

To be re-evaluated with longer duration input signals