CERN

picoTDC:
Pico-second TDC for HEP

Moritz Horstmann, Jorgen Christiansen, Samuele Altruda,
Gill Lumer-Klabbers, Jeffrey Prinzie (KU Leuven)

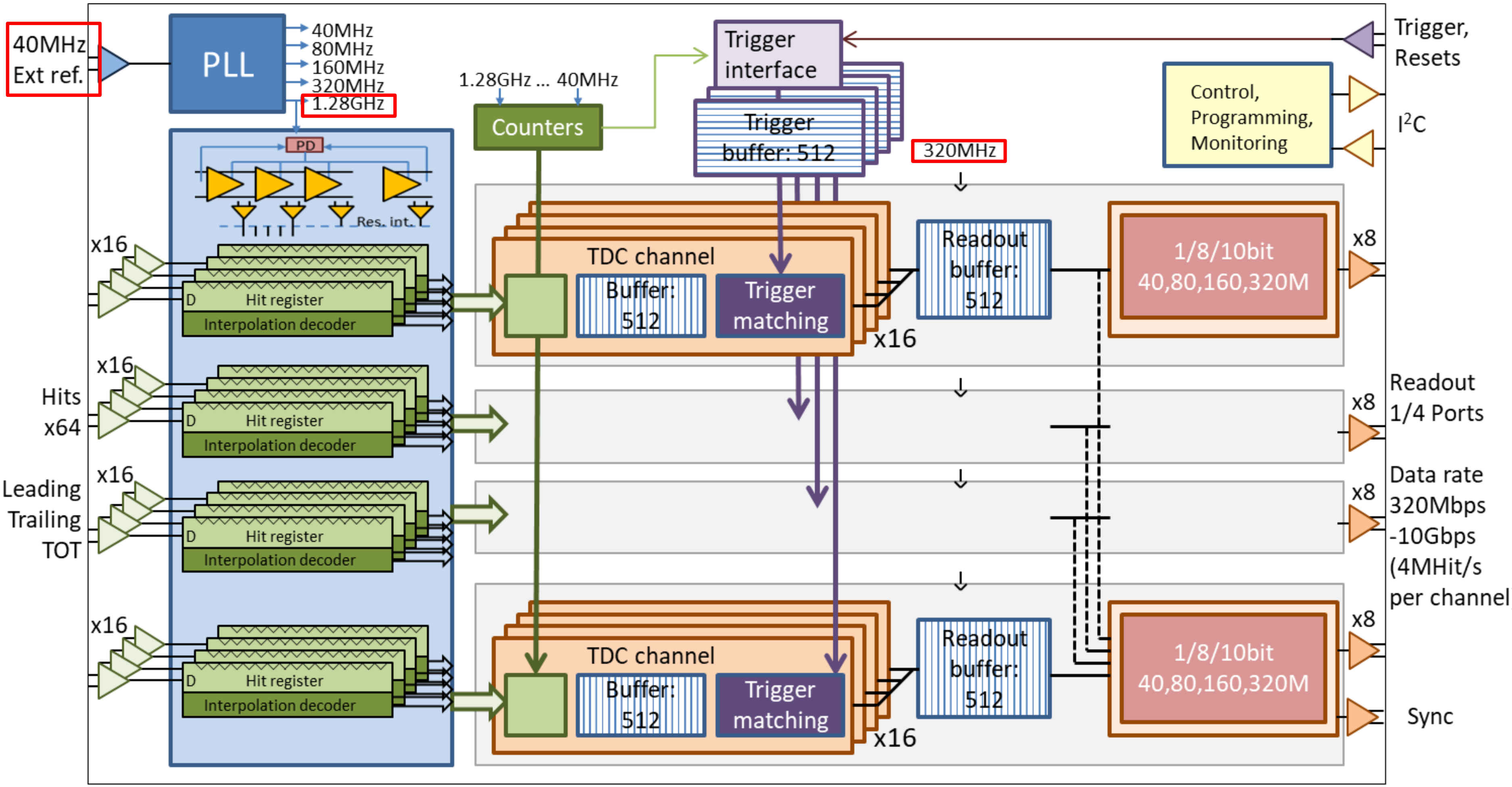
CERN/EP-ESE

PicoTDC: overview

- **PicoTDC Design in a glance**
  - Architecture
- **Testing system**
  - HW,FW,SW
- **Tests**
  - Code density test
  - Sweep test
  - Crosstalk test
  - Cable delay test
- **New PicoTDC version**
  - New features and planning

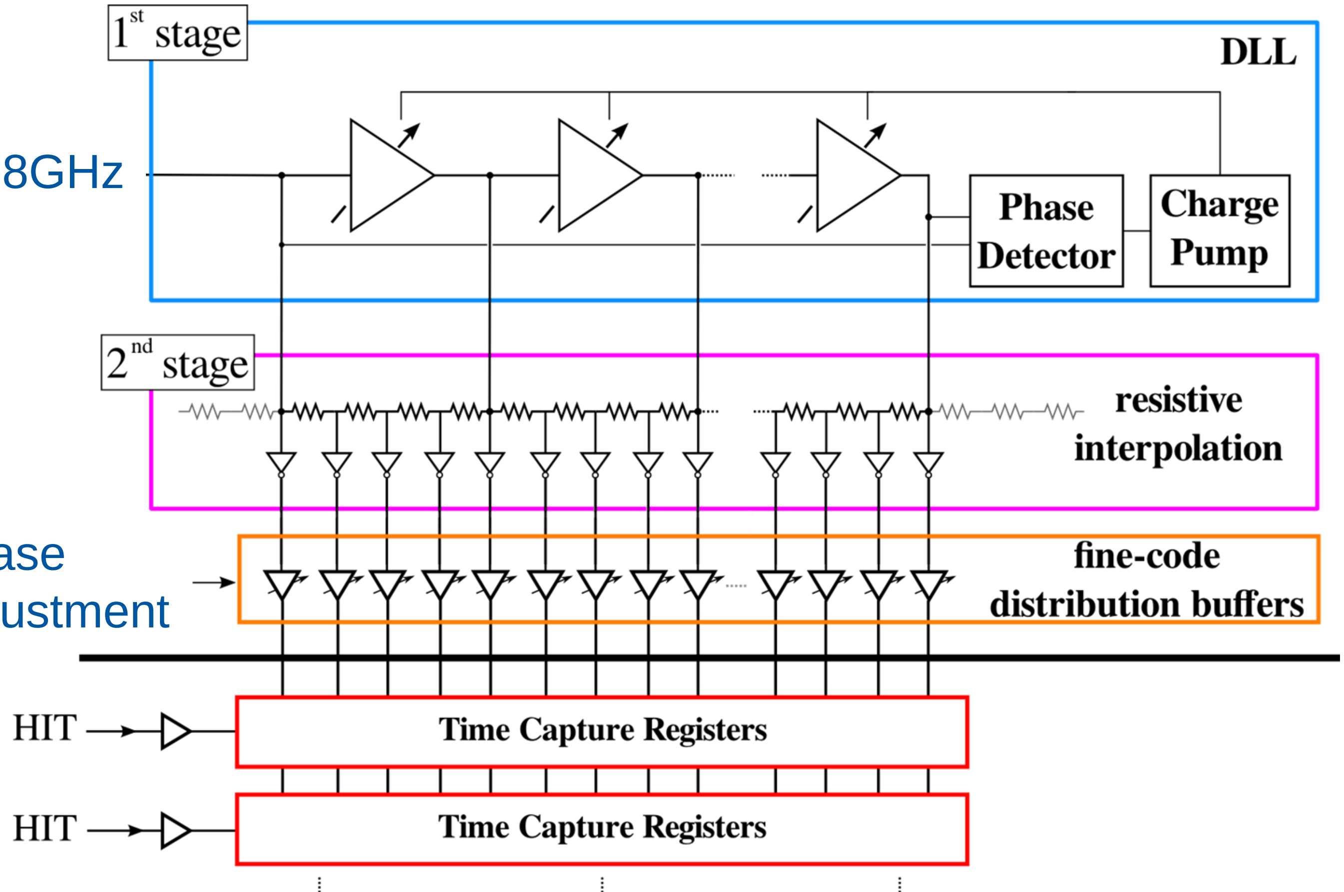
PicoTDC architecture

PicoTDC Architecture



64 channels, 3ps or 12ps time binning, 200us dynamic range

Two Stage Time Interpolation



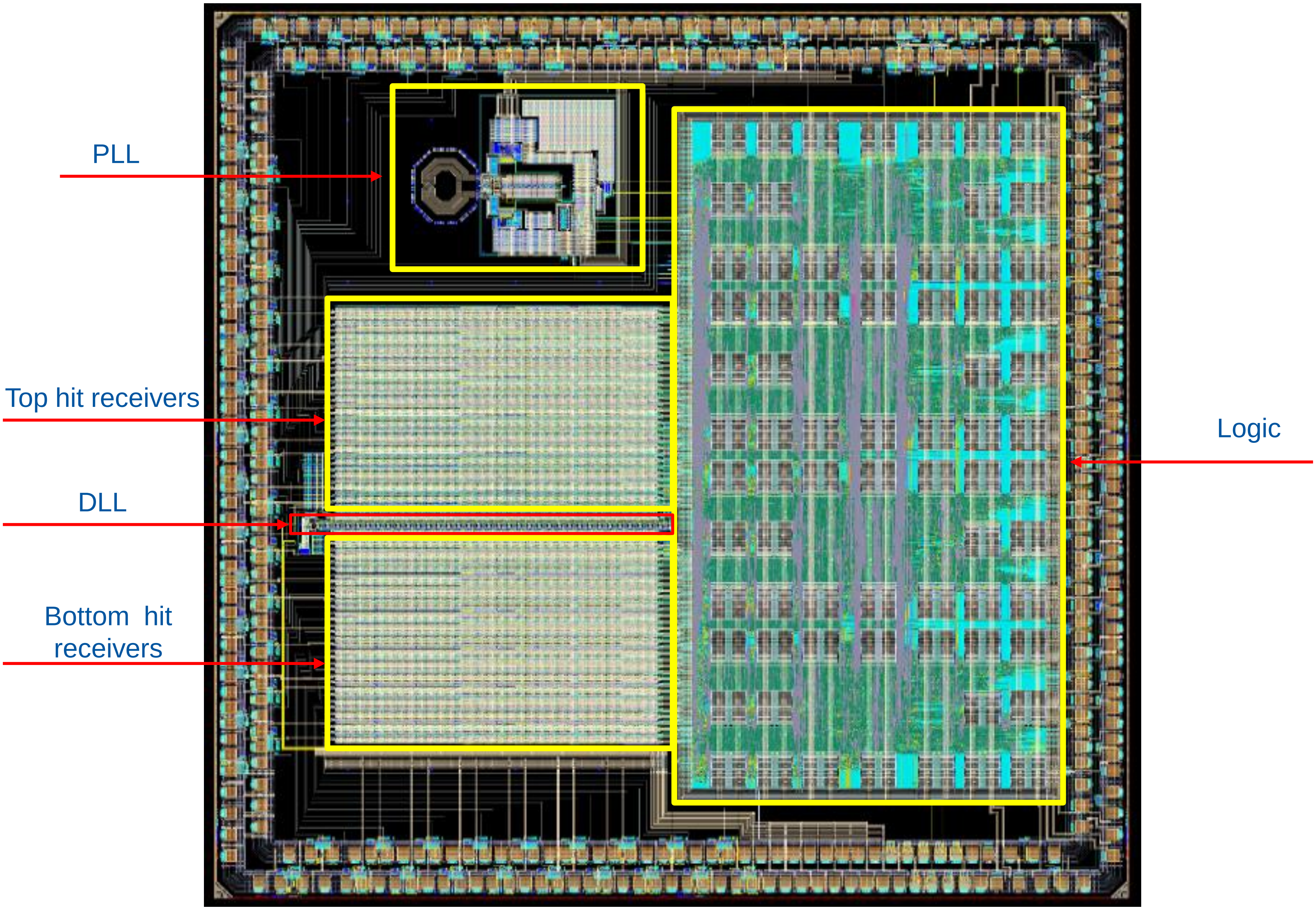
12.2ps delay (Coarse time)

3.05ps delay (Fine time)

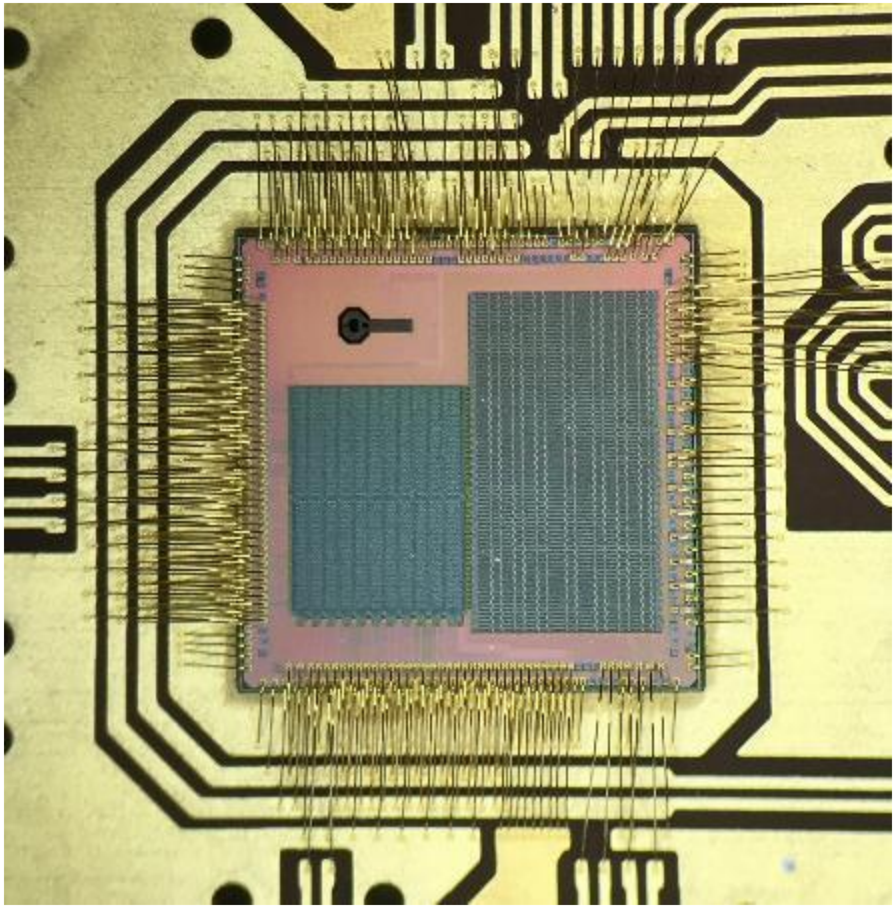
32 steps each with 0.6ps resolution

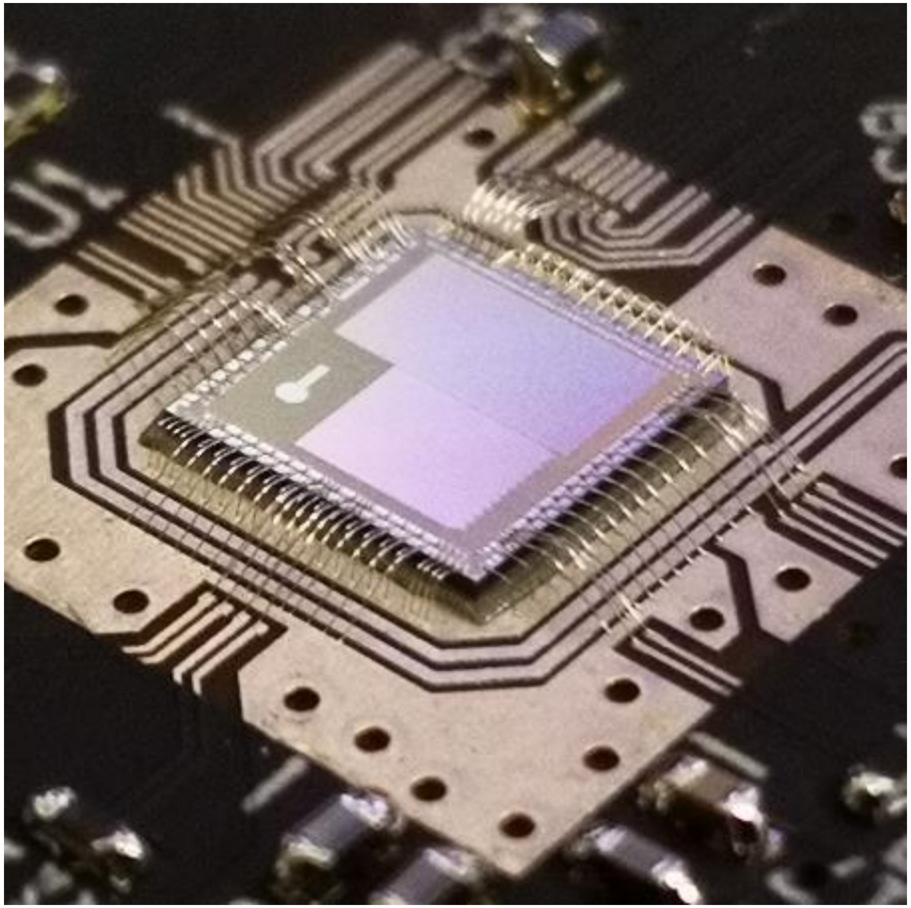
Phase Adjustment

Known issue: Capture FF Mismatch → Solved in next version


PLL
Top hit receivers
DLL
Bottom hit receivers
Logic

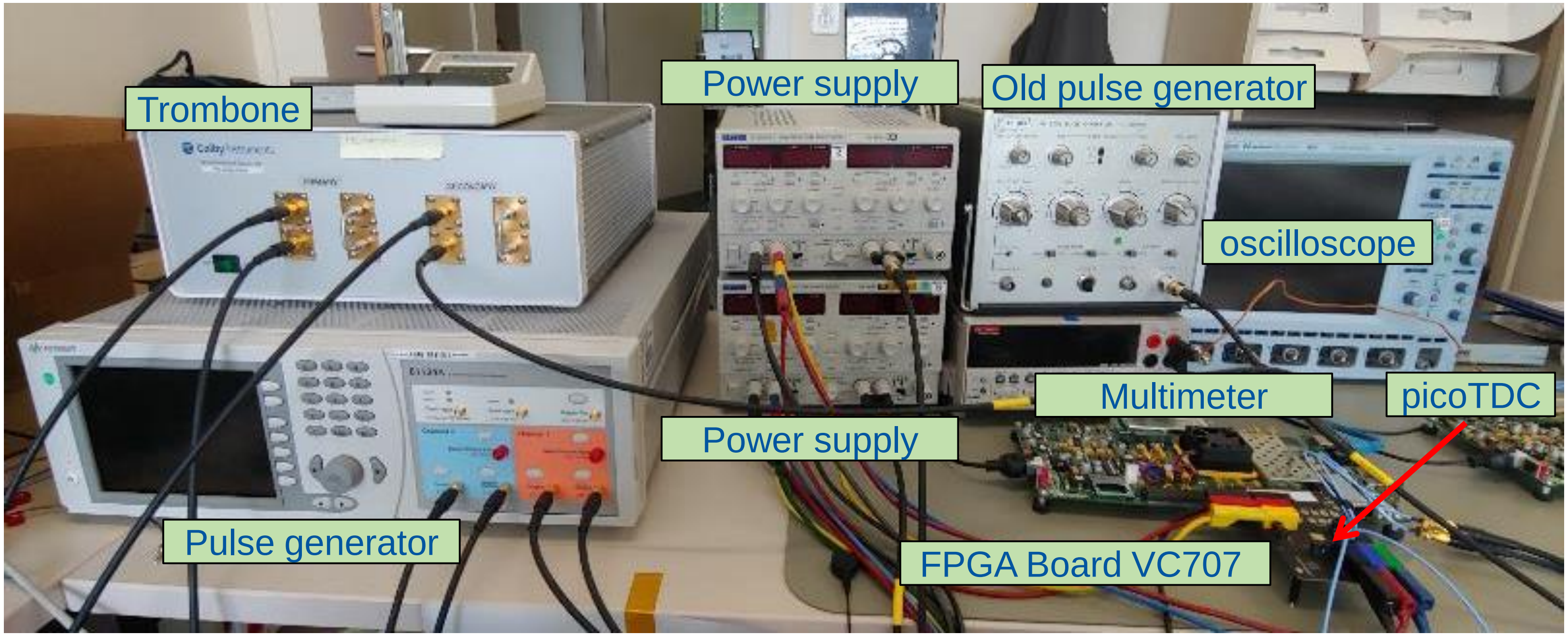
PicoTDC on Test Cards



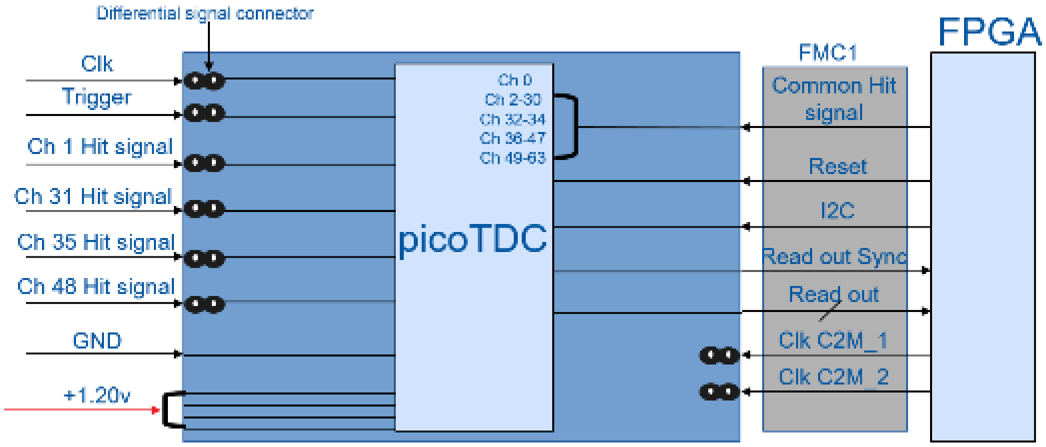


PicoTDC testing system

Instrumentation for testing



PCB mezzanine test board

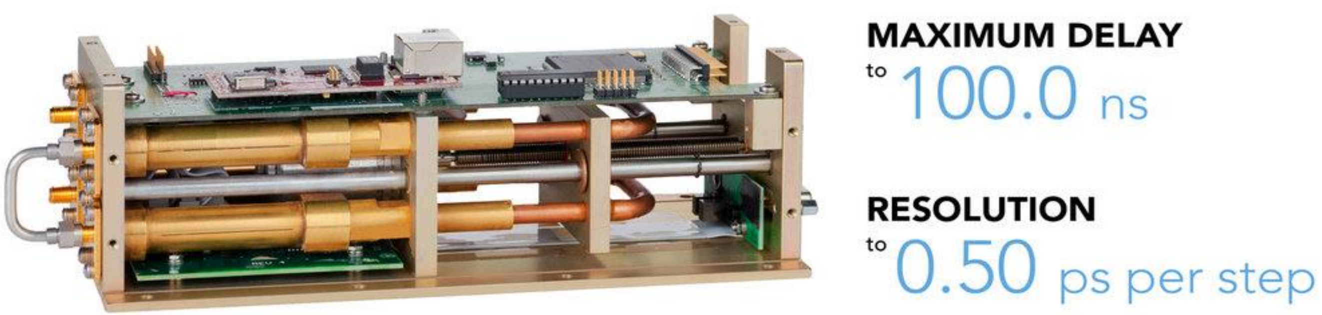


PicoTDC delay sweep test

PicoTDC: Delay sweep test

Sweep test is performed to quantify the linearity of picoTDC

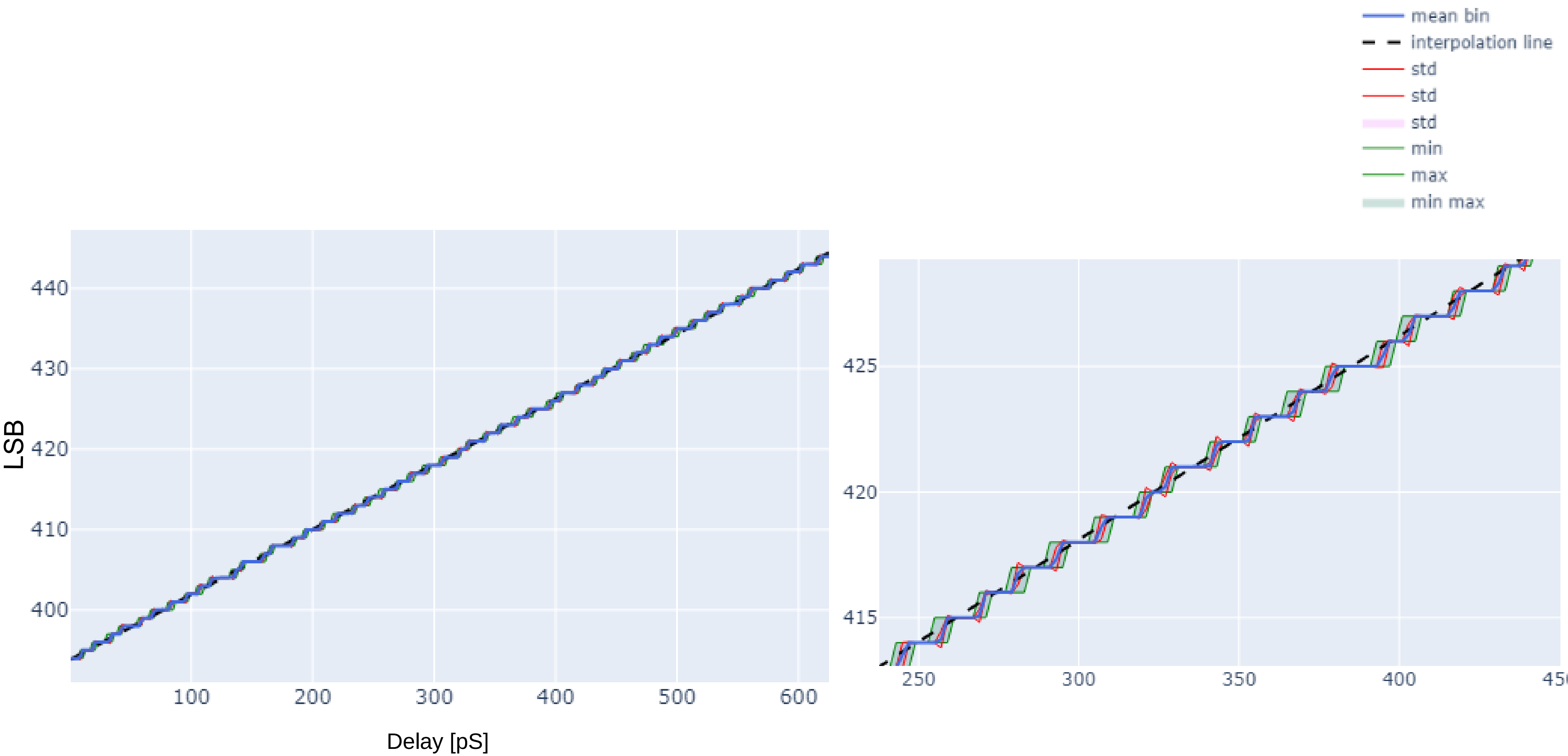
To perform sweep test we use a dual channel pattern generator to provide the clock signal and the hit signal which is then delayed by the trombone.
The trombone is a very precise and repeatable programmable delay line.



PicoTDC: Delay sweep test

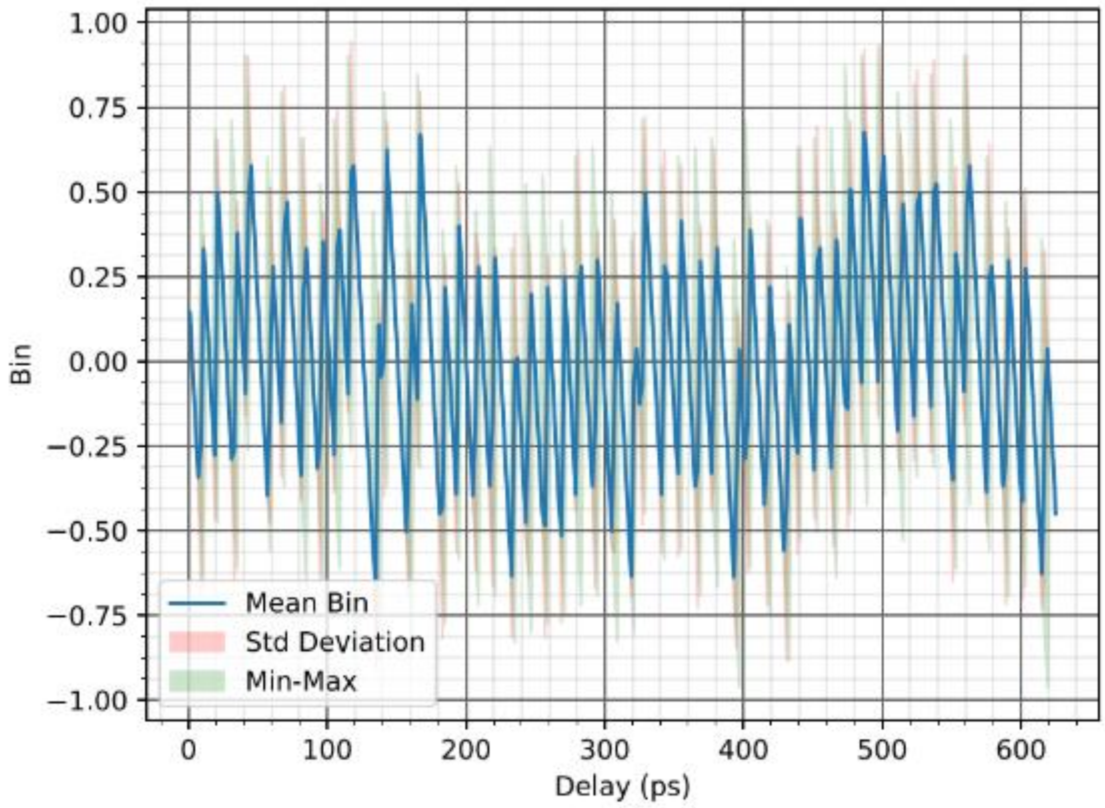
Ch 31, not adjusted, **coarse mode**, bin 12ps, RMS_inl = 4,129ps

Total jitter= jitter from picoTDC + jitter from pulse generator



PicoTDC: Delay sweep test

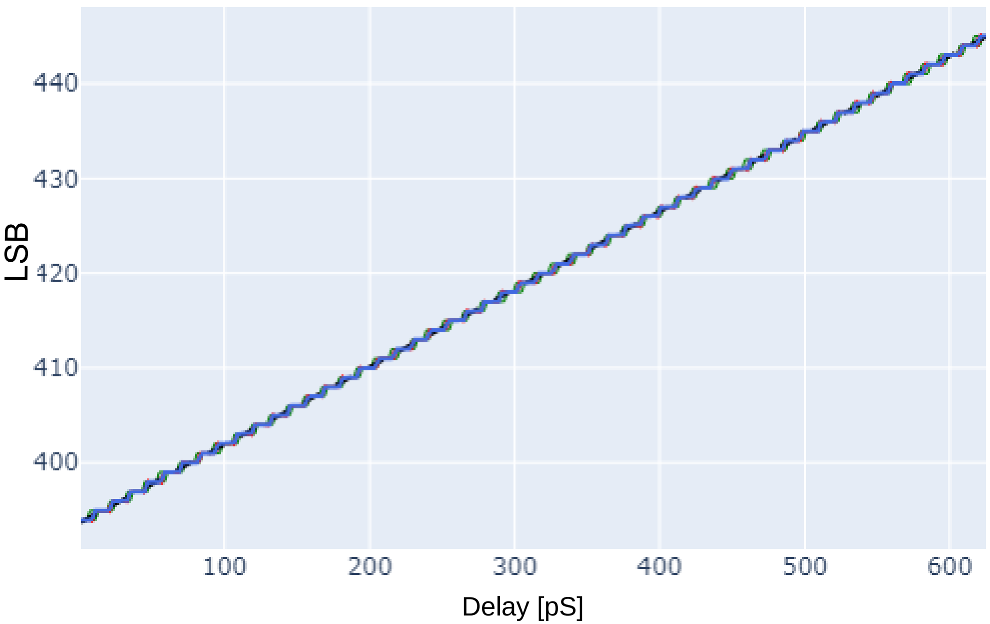
Ch 31, not adjusted, **coarse mode**, bin 12ps, RMS_inl = 4,129ps

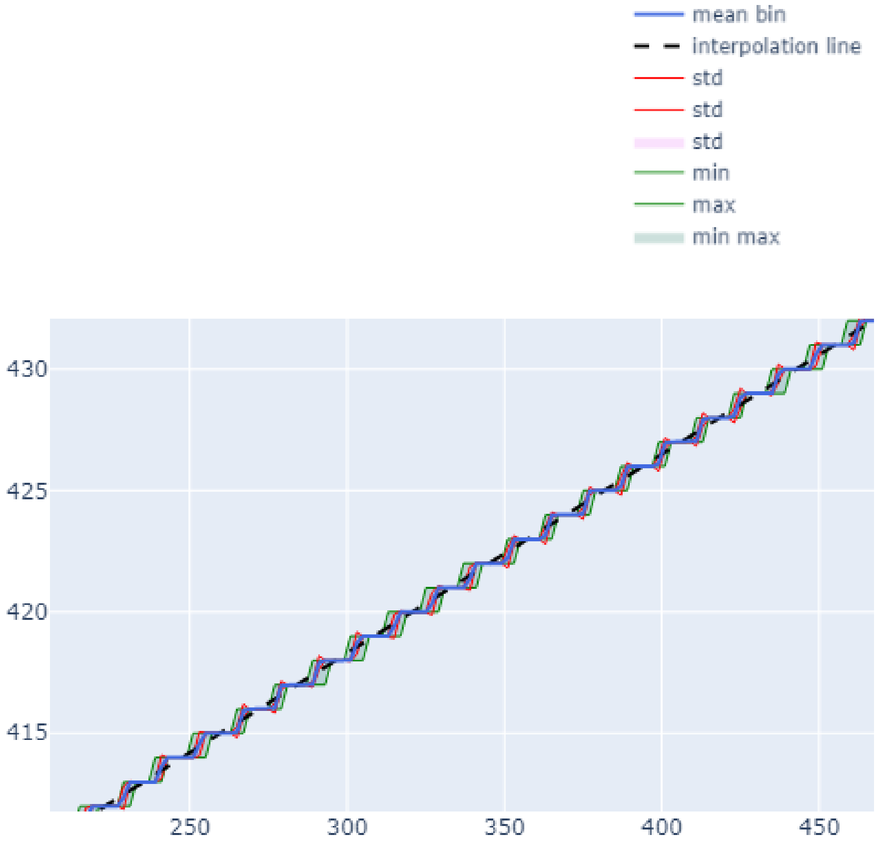


PicoTDC: Delay sweep test

Ch 31, adjusted, **coarse mode**, bin 12ps, RMS_inl = 3,585ps

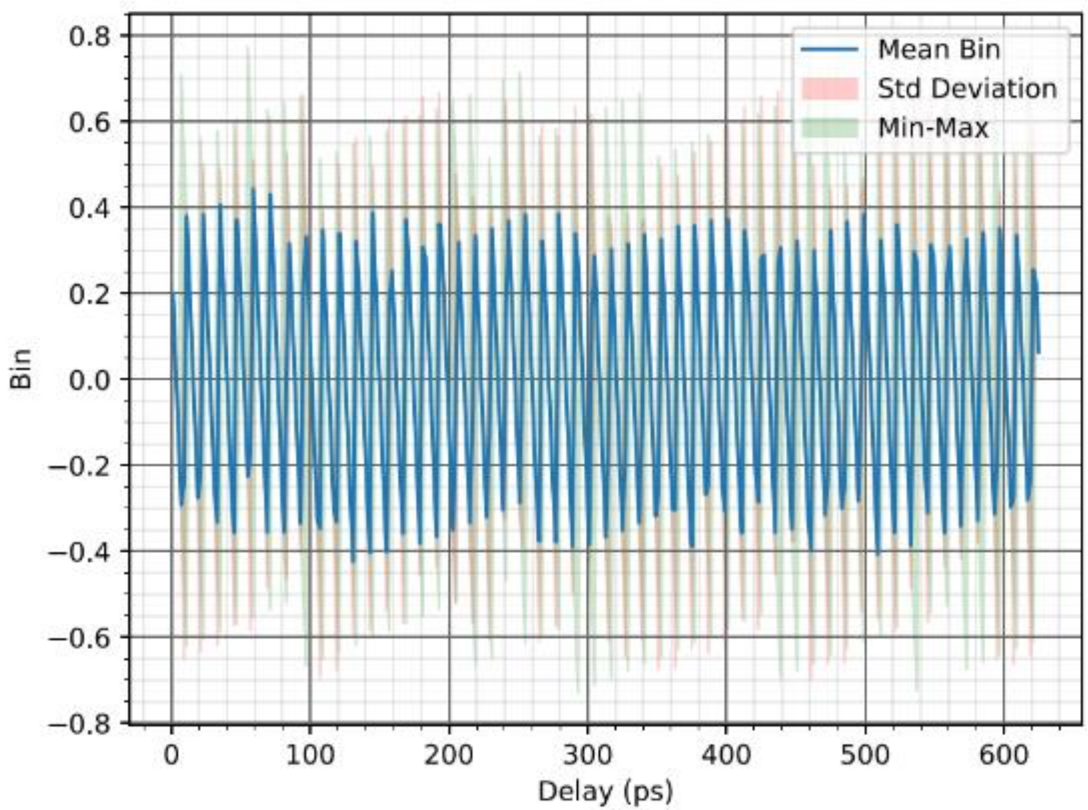
Adjustment for Ch 31 is performed for this channel alone





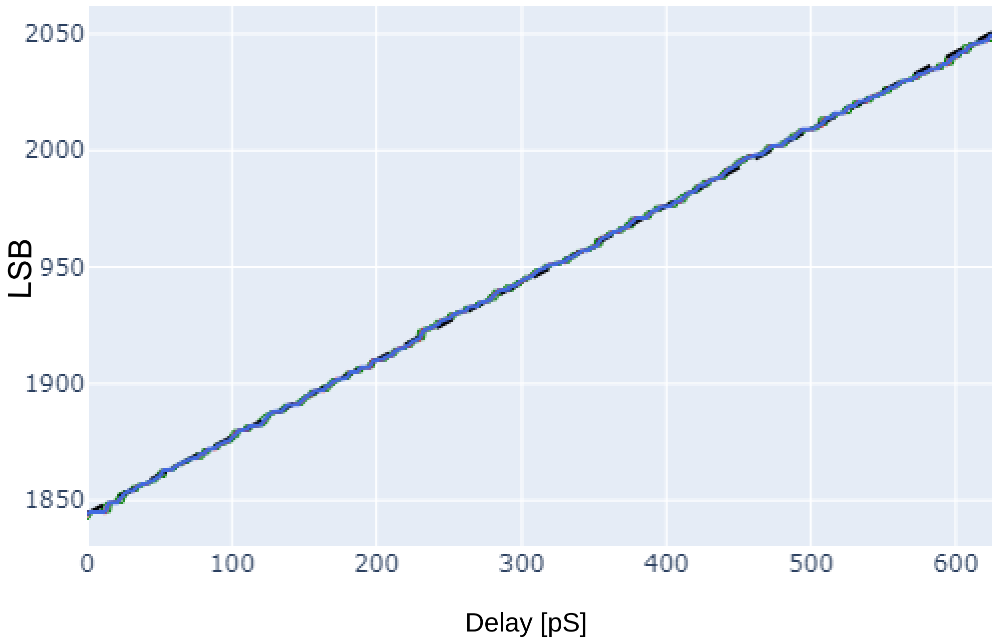
PicoTDC: Delay sweep test

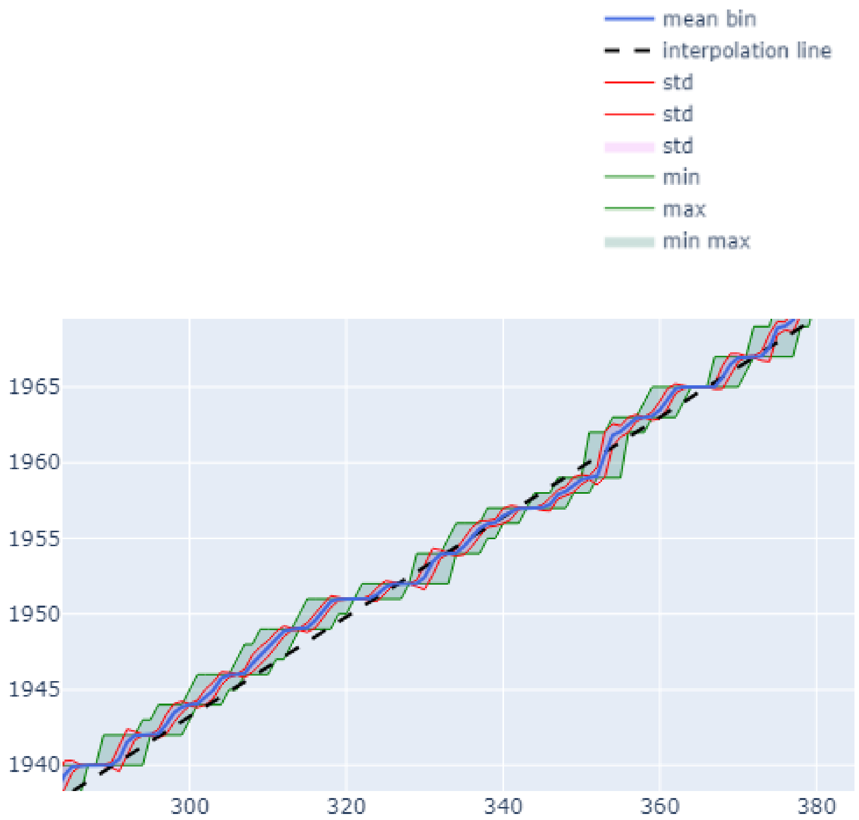
Ch 31, adjusted, **coarse mode**, bin 12ps, RMS_inl = 3,585ps



PicoTDC: Delay sweep test

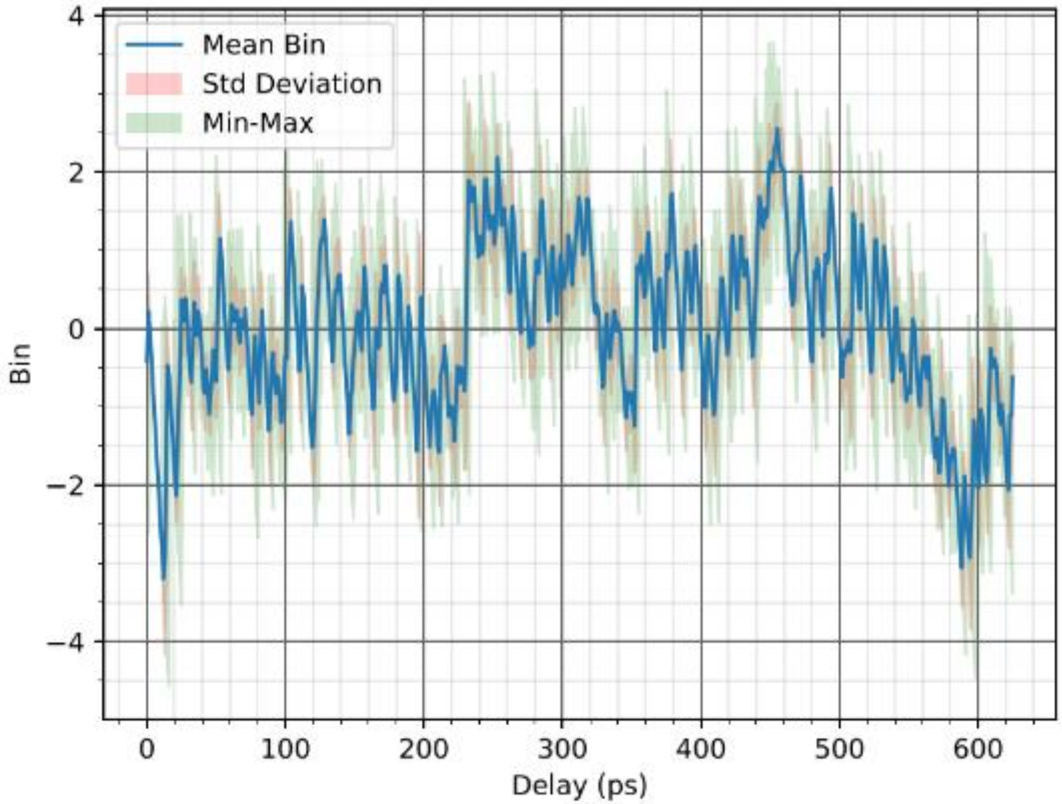
Ch 31, not adjusted, **fine mode**, bin 3ps, RMS_inl = 3,416ps





PicoTDC: Delay sweep test

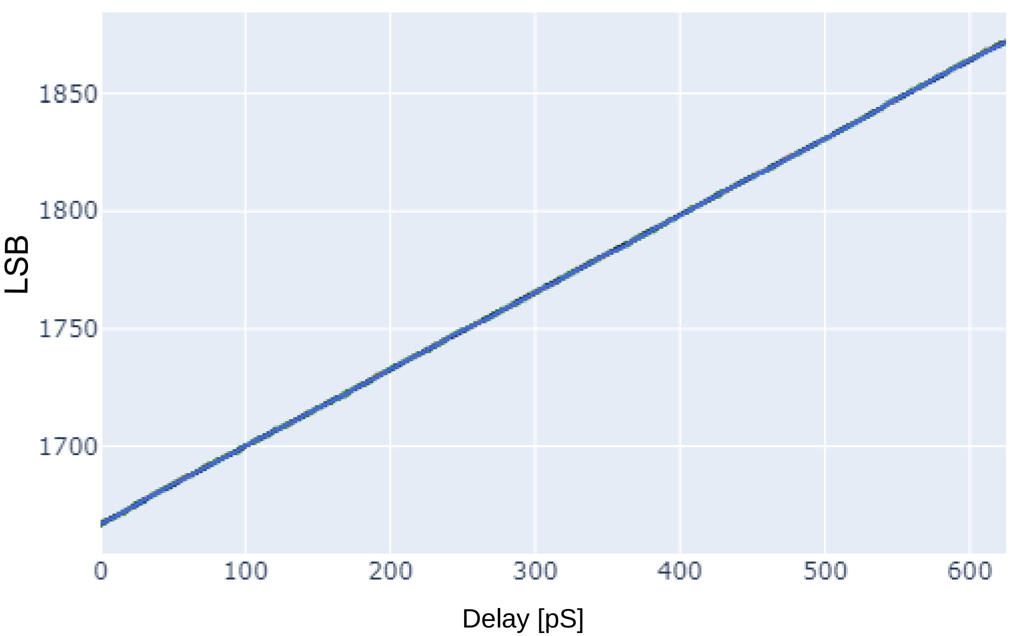
Ch 31, not adjusted, **fine mode**, bin 3ps, RMS_inl= 3,416ps



PicoTDC: Delay sweep test

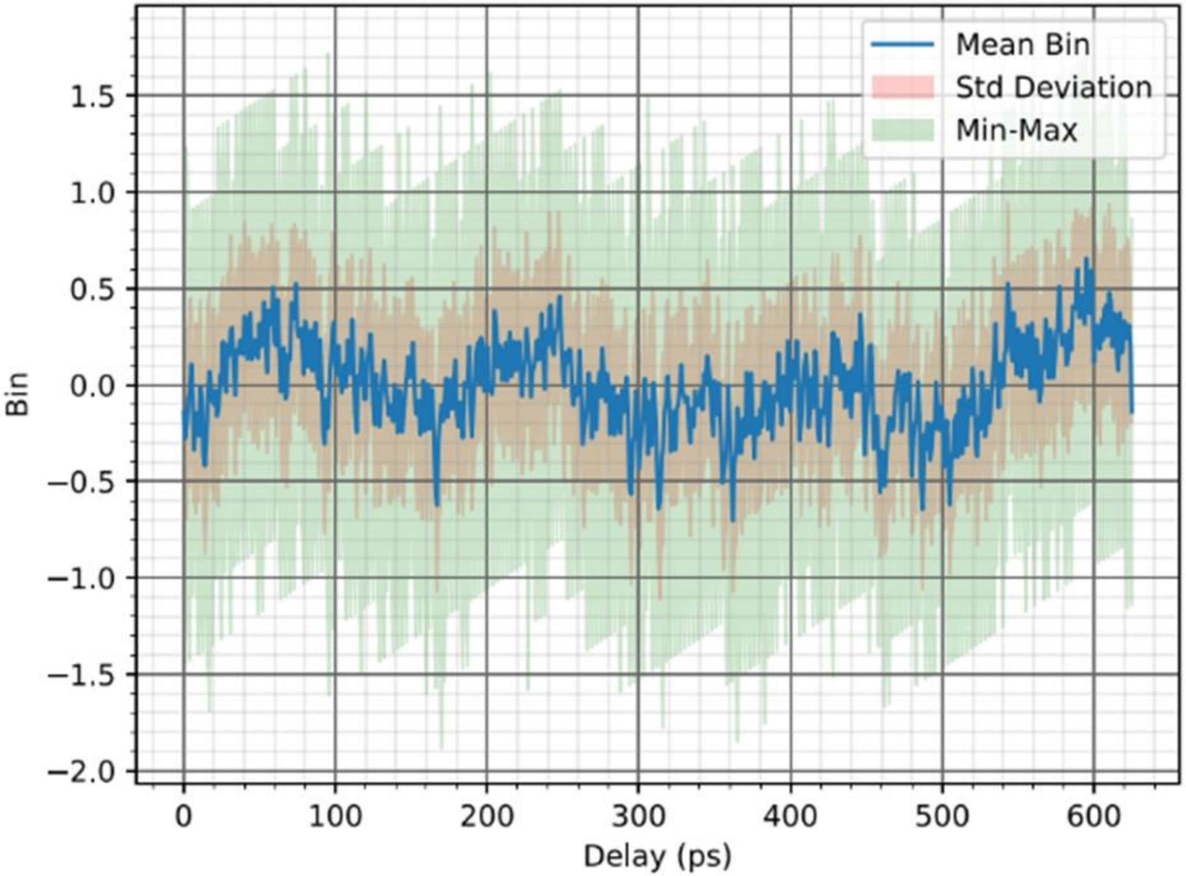
Ch 31, adjusted, **fine mode**, bin 3ps, RMS_inl = 1,296ps

Adjustment for Ch 31 is performed for this channel alone



PicoTDC: Delay sweep test

Ch 31, adjusted, **fine mode**, bin 3ps, RMS_inl= 1,296ps

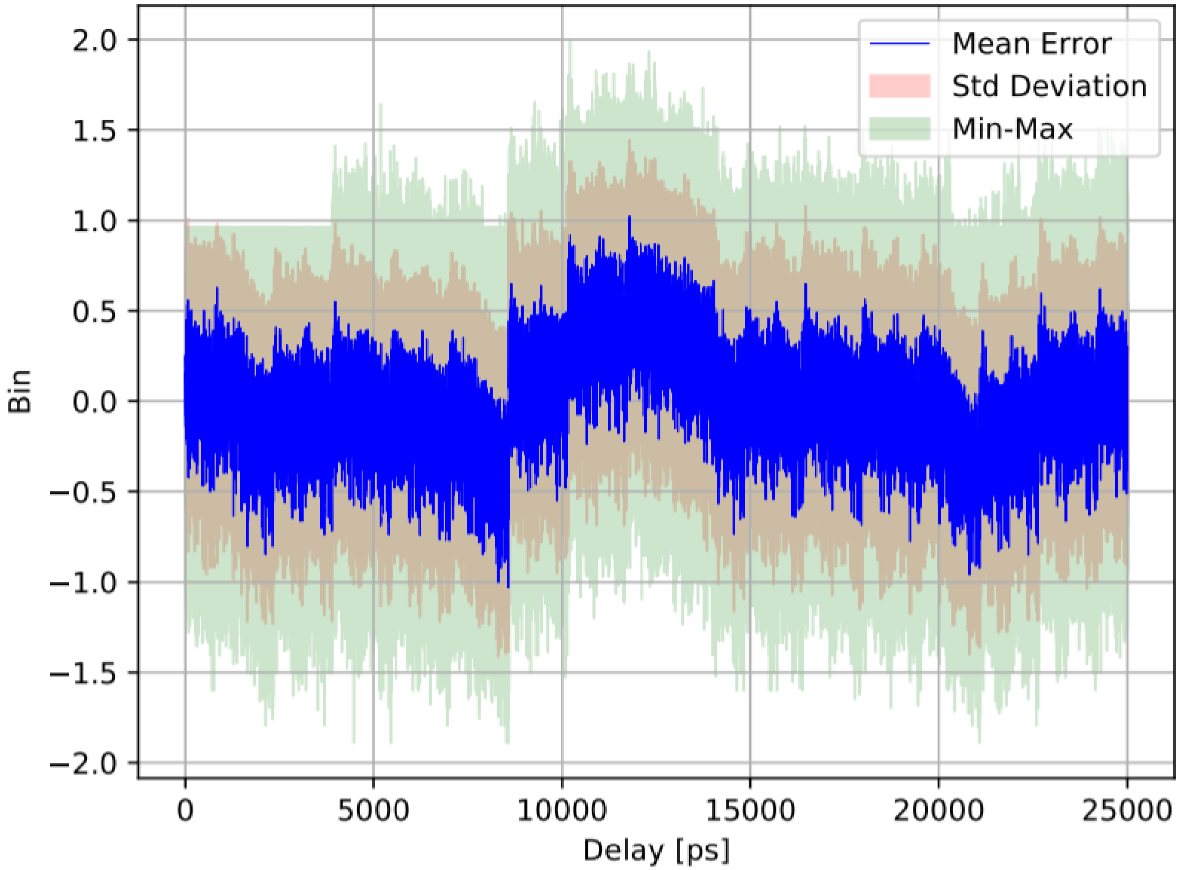


PicoTDC: Delay sweep test 25ns

Ch 31, adjusted, **fine mode**, bin 3ps, RMS_inl = 1,352ps Averaged_inl=0,801ps

Adjustment for Ch 31 is performed for this channel alone

Sweep performed for 25ns



PicoTDC code density test

PicoTDC: Code density test

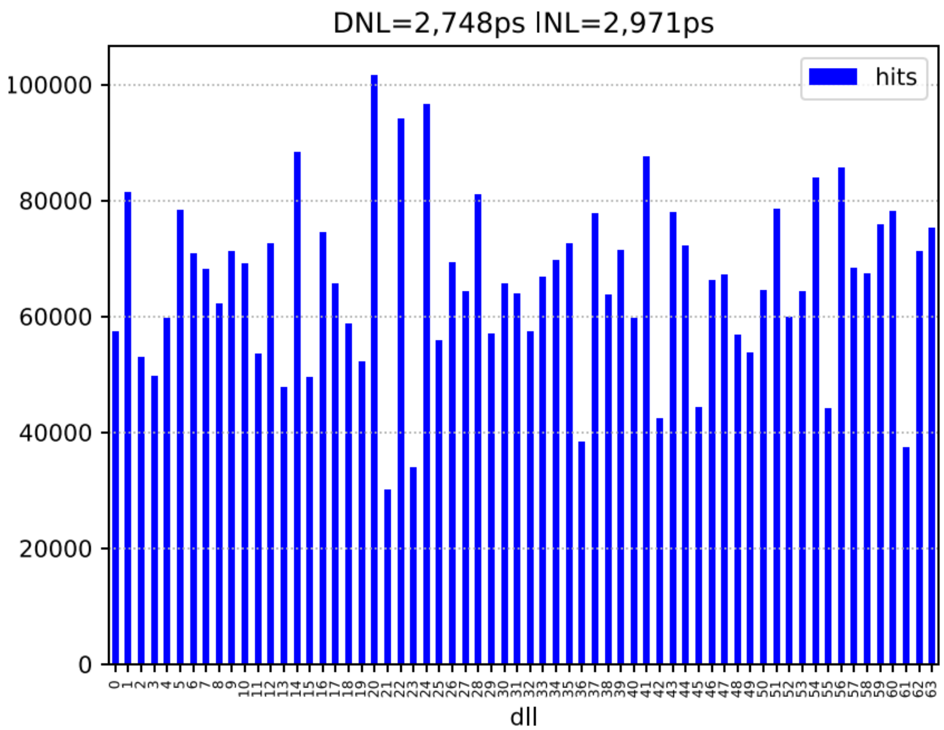
Code density test is performed to measure the effective bin size. The measurement doesn’t include jitter and quantization.

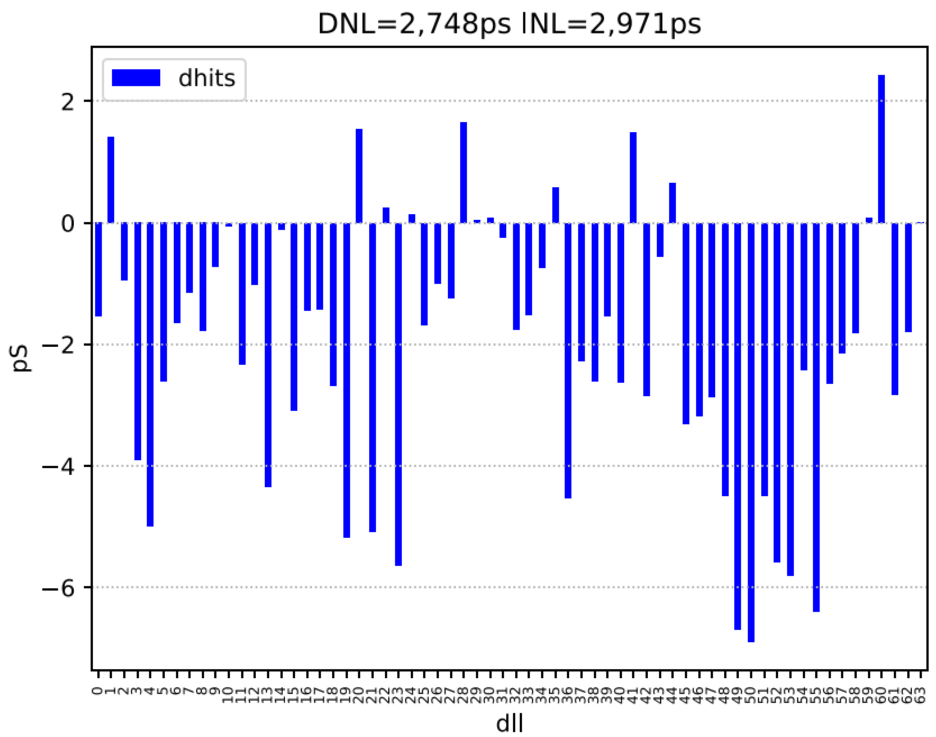
To perform code density test we use an old RC based generator, which is sufficiently “random”, providing us random hits.

As a random source we expect the same amount of hits in each bin.

PicoTDC: Code density test

Ch 31, not adjusted, **coarse mode**, bin 12ps, RMS_dnl= 2,748ps, RMS_inl= 2,971ps

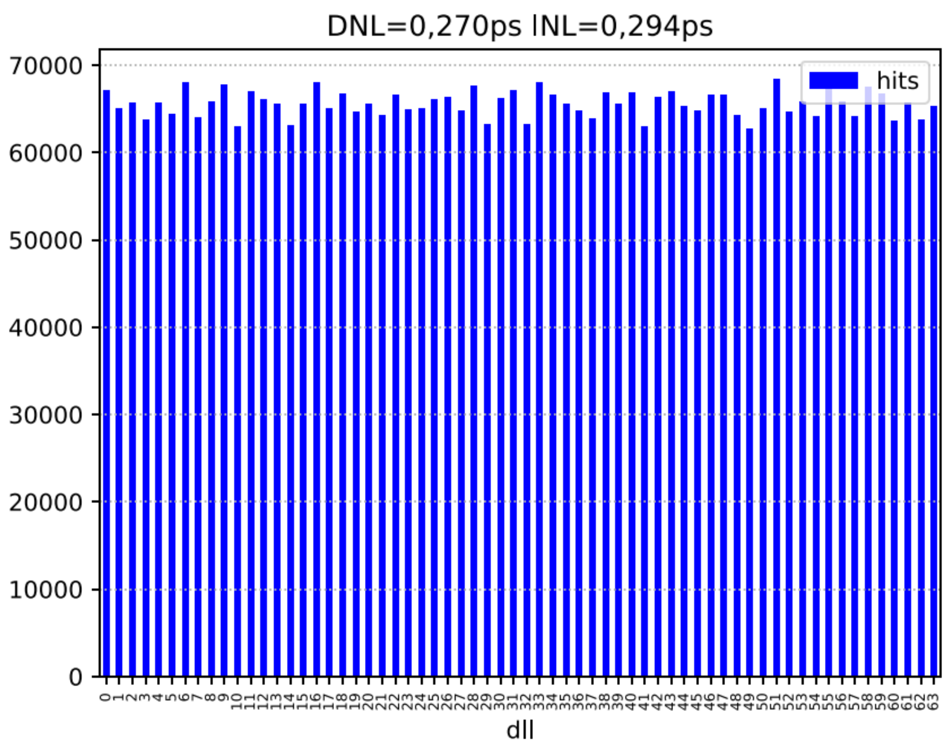


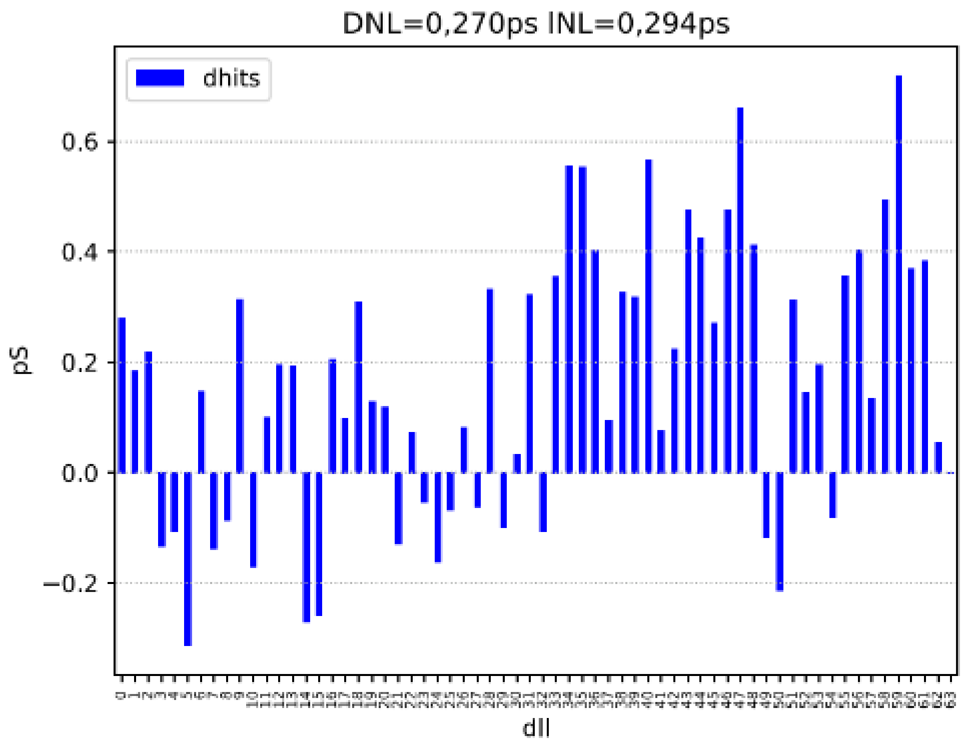


PicoTDC: Code density test

Ch 31, adjusted, **<u>coarse mode</u>**, bin 12ps, RMS_dnl= 0,270ps, RMS_inl= 0,294ps

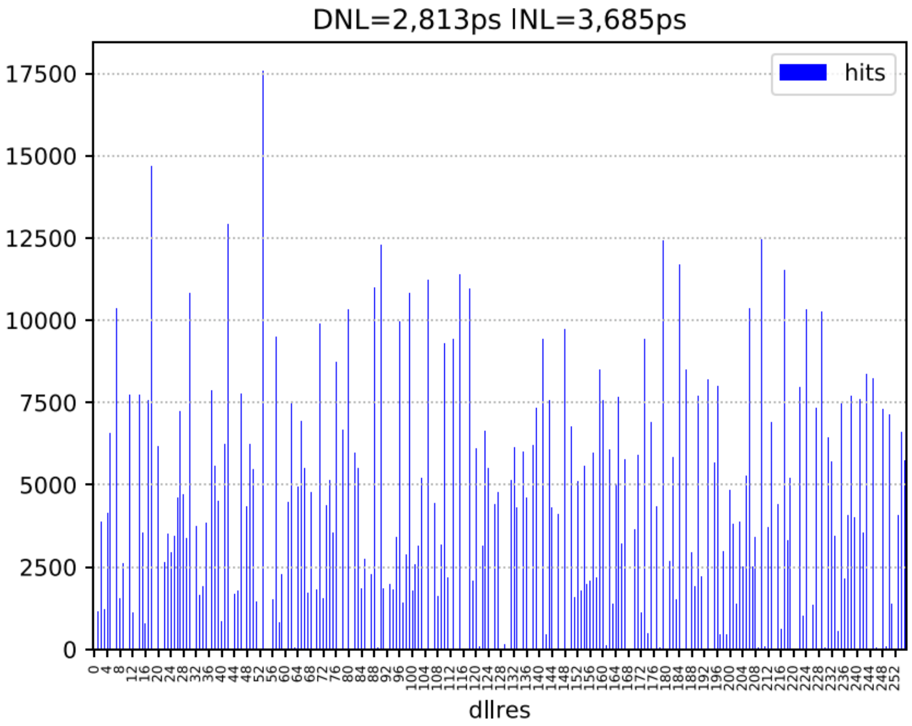
Adjustment for Ch 31 is performed for this channel alone

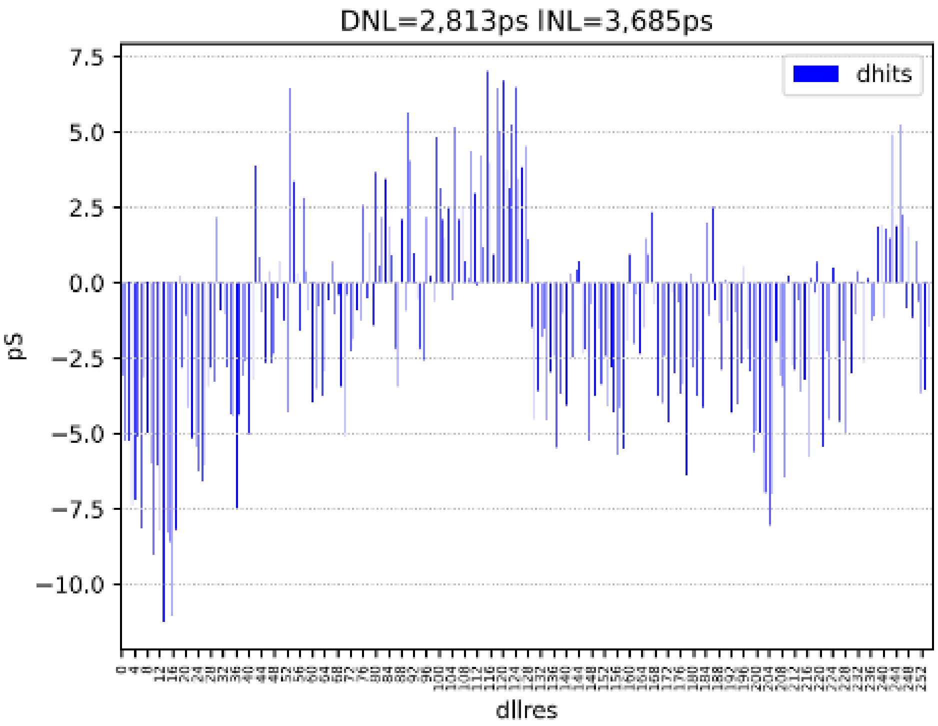




PicoTDC: Code density test

Ch 31, not adjusted, **fine mode**, bin 3ps, RMS_dnl= 2,813ps, RMS_inl= 3,685ps



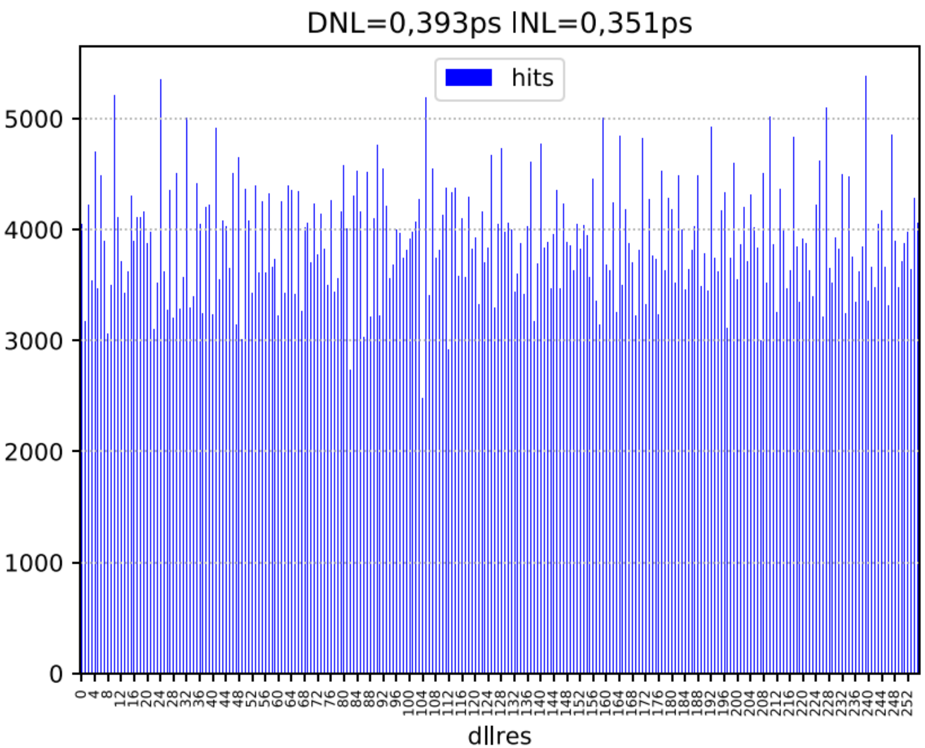


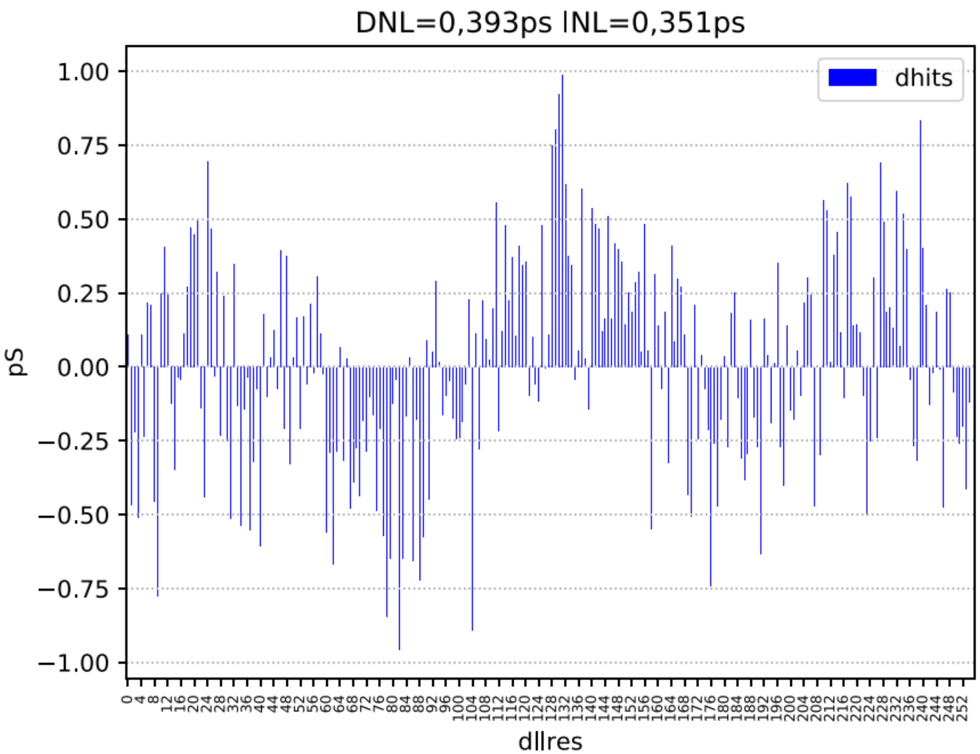
PicoTDC: Code density test

Good adjustment result after 2 step adjustment

Adjustment for Ch 31 is performed for this channel alone

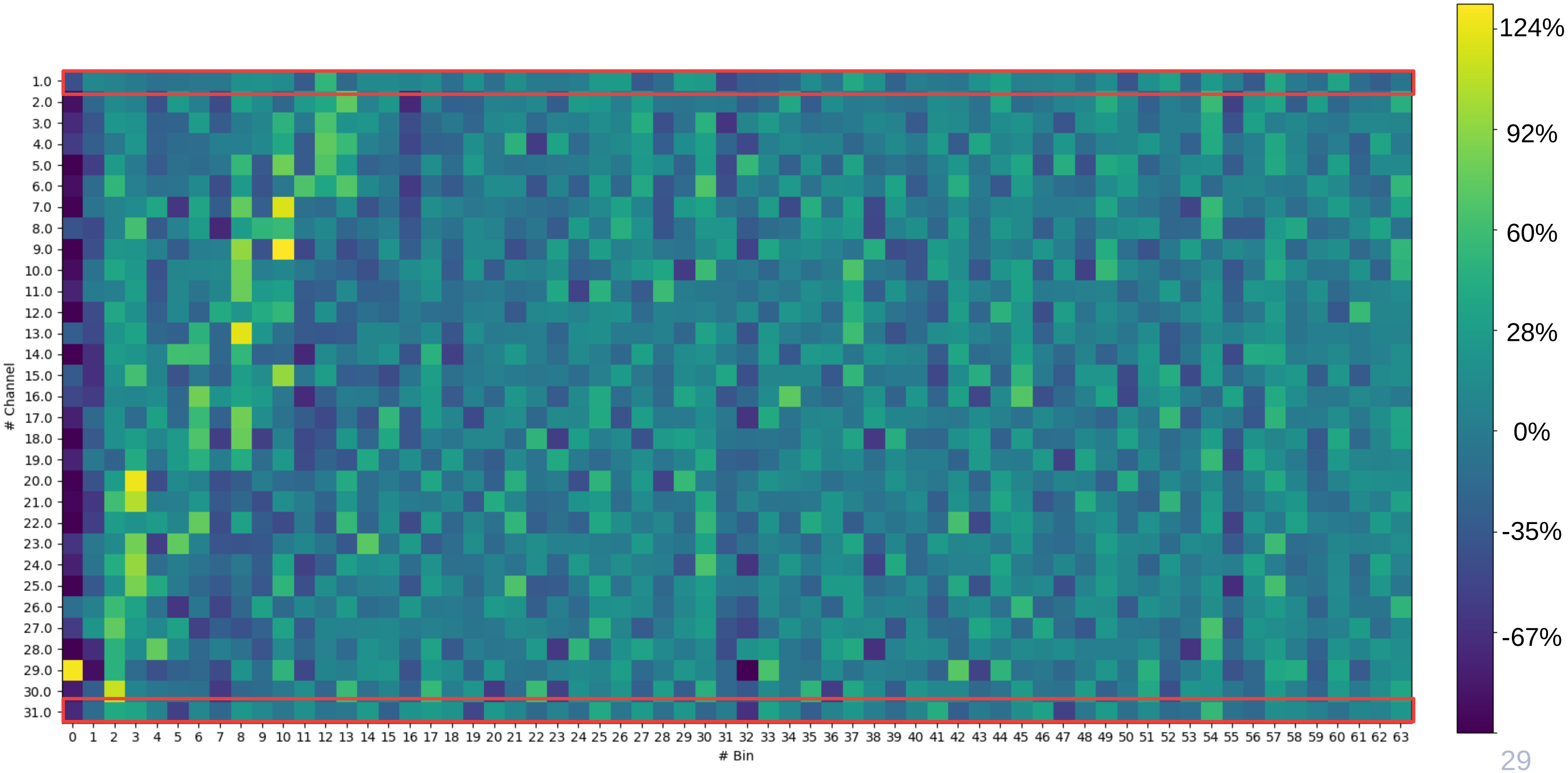
Ch 31, adjusted, **fine mode**, bin 3ps, RMS_dnl= 0,393ps, RMS_inl= 0,351ps





PicoTDC: Code density test

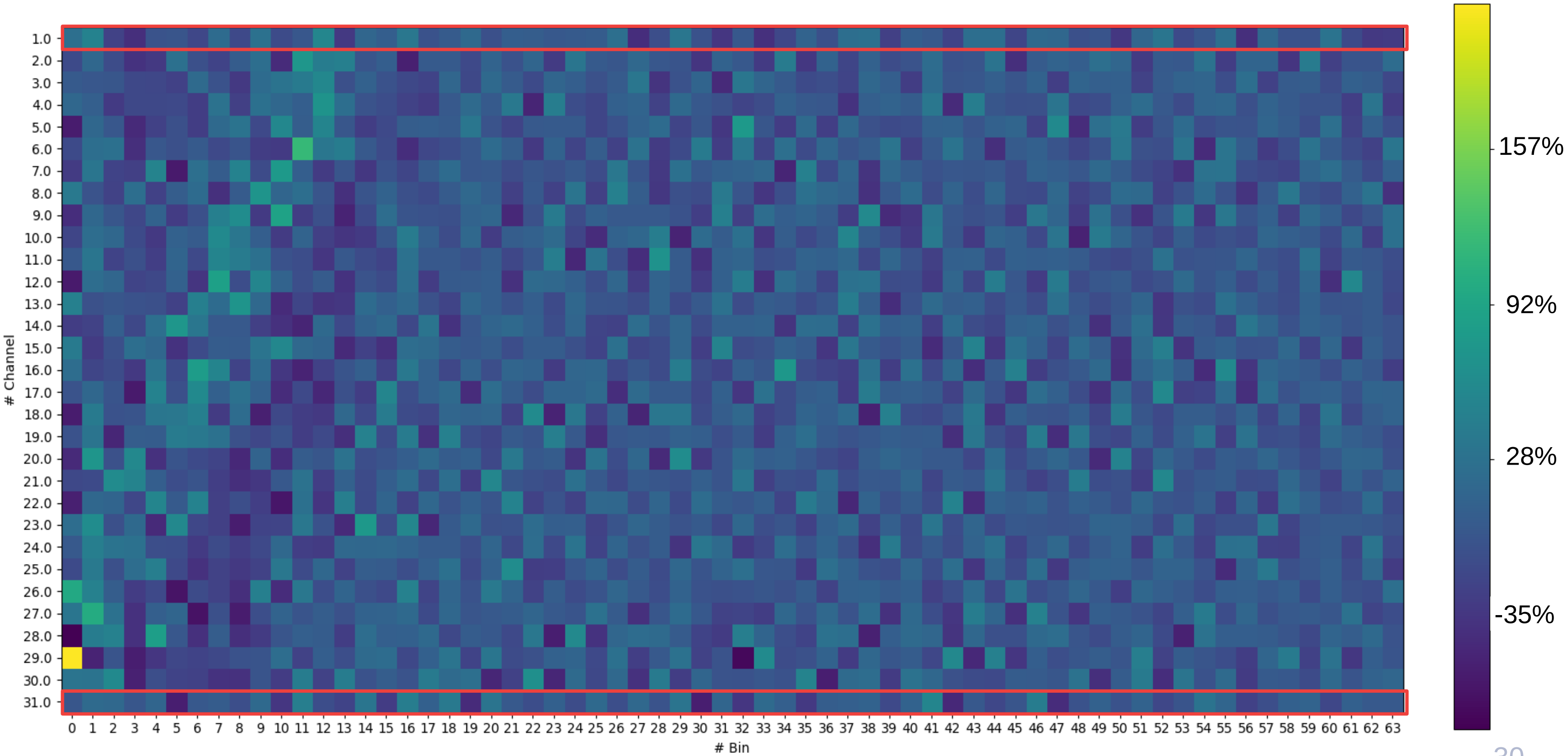
CDT test **coarse time** not adjusted: Grouped vs. single channel RMS_dnl= 3,48ps



29

PicoTDC: Code density test

CDT test **coarse time** adjusted: Grouped vs. single channel RMS_dnl= 3,06ps



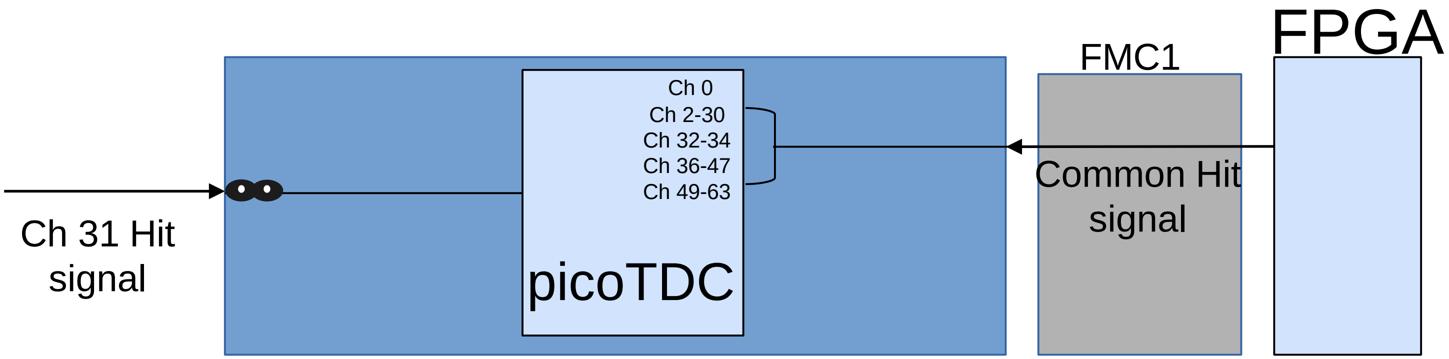
30

PicoTDC crosstalk test

PicoTDC: Crosstalk test

Cross talk test is performed to quantify the noise introduced by the others channels against one. Worst case in exam: one against all.

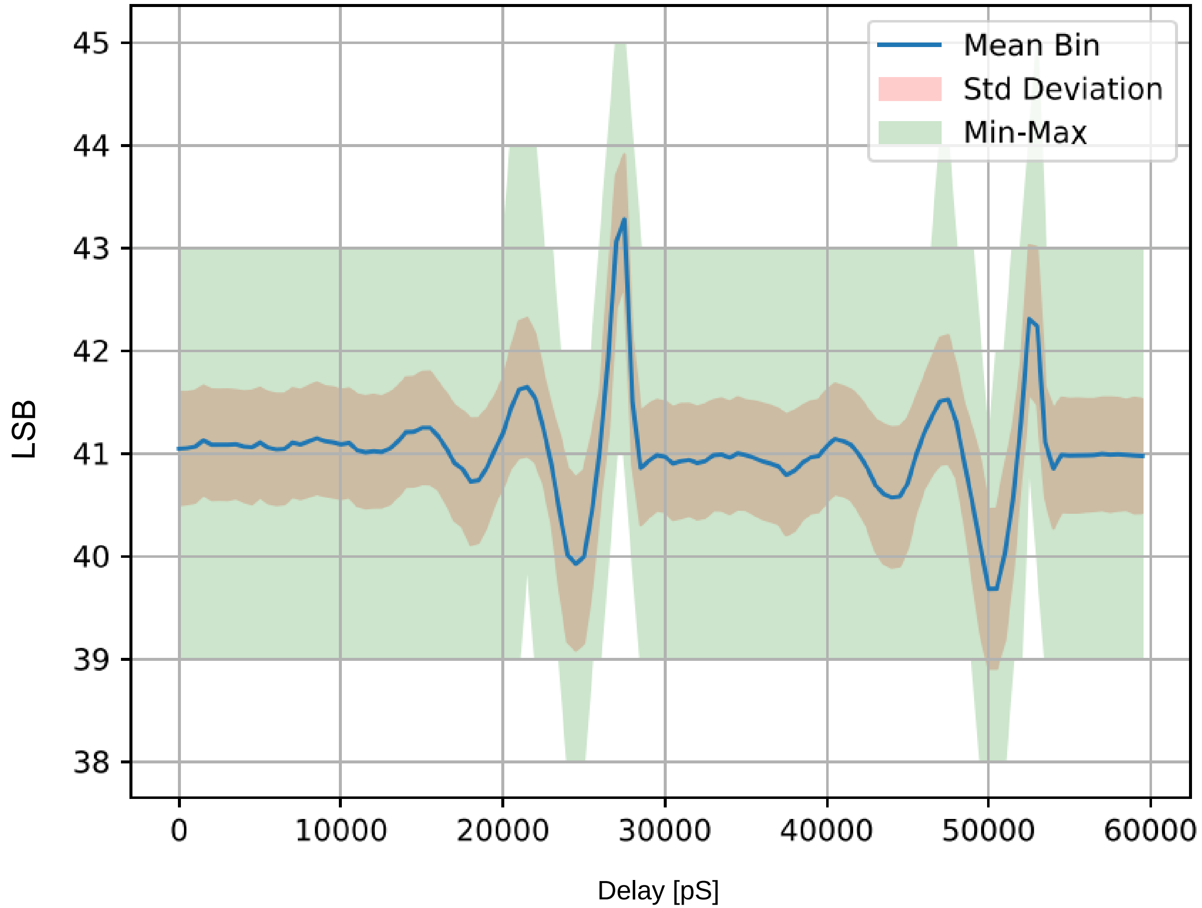
Test scheme simplified



Common Hit signal delay sweep from 0ps to 60000ps
Ch 31 Hit signal at fixed delay

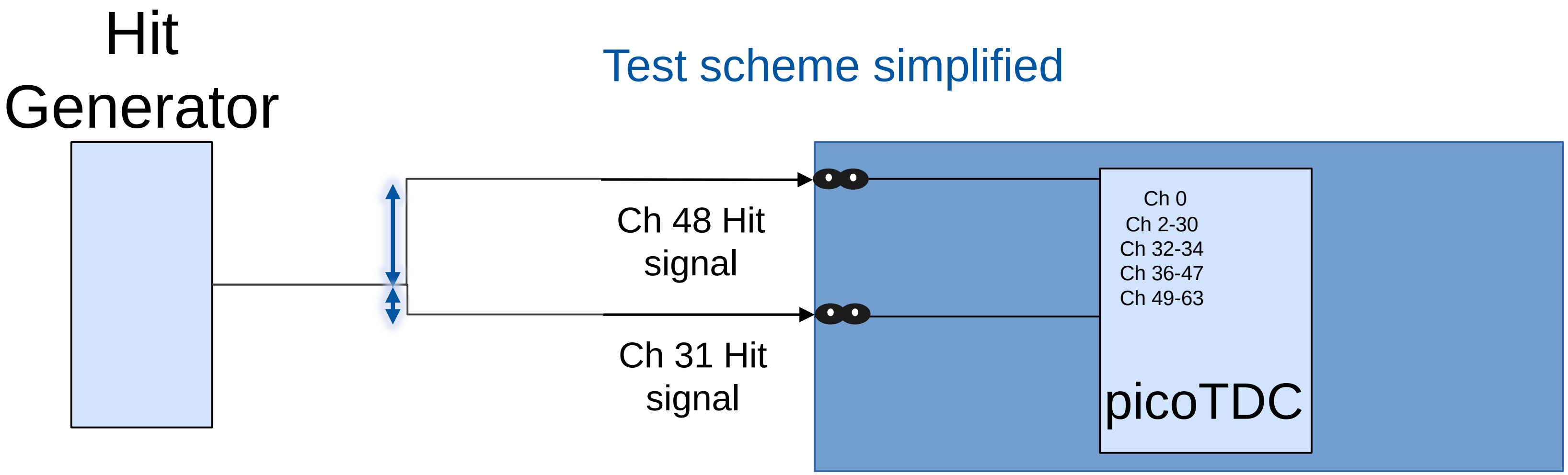
PicoTDC: Crosstalk test

Ch 31 vs All Chs, **coarse mode**, bin 12ps, LSB 12ps



PicoTDC: Cable delay test

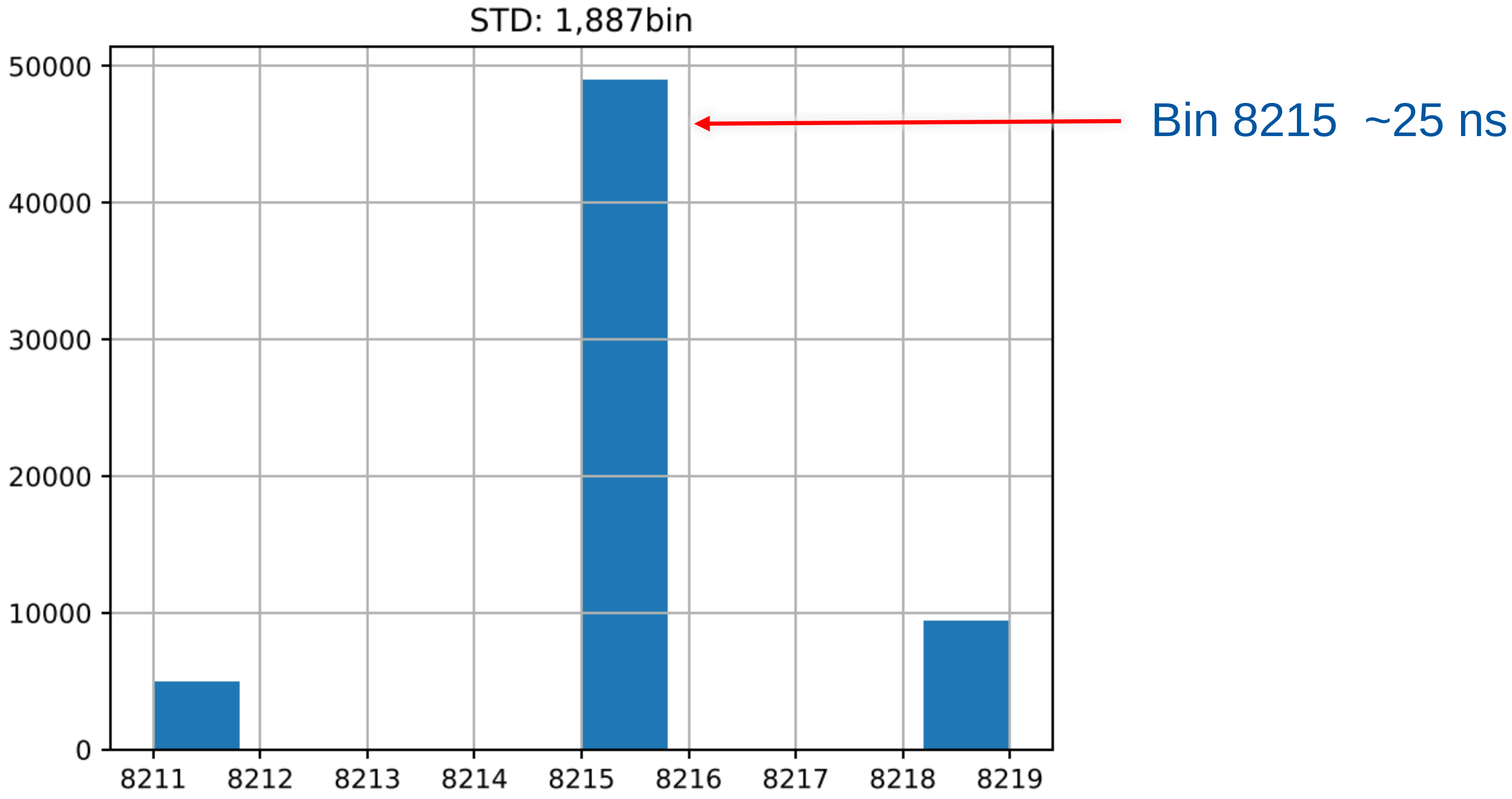
Cross talk test is performed to quantify the linearity of the system over multiple clock cycle, around 200ms



Hit generator outputs a delay sweep hits.
Hits go through different cables with different lengths to channel 31 and 48. Our TDC will measure the time of arrival of the two hits.

PicoTDC: Cable delay test

Ch 31, Ch 48 , **coarse mode**



X axis scale in fine time, 3ps binning

PicoTDC: performances summary

	Ch	adjusted	Code Density Test		Sweep Test
	Ch	adjusted	DNL	INL	INL
Coarse time	31	✗	2.748ps	2.971ps	4.129ps
	31	✓	0.270ps	0.294ps	3.585ps
Fine time	31	✗	2.813ps	3.68ps	3.416ps
	31	✓	0.393ps	0.351ps	1.296ps

CDT doesn't include jitter, quantization

INL include jitter, trombone non linearity, quantization

Temperature performance	Coarse time	Variation limited in min-max 1 LSB pp	<1ps/ºC
Voltage performance	Coarse time	Variation limited in min-max 7 LSB pp	<0.5ps/mV
Crosstalk test	Coarse time	Influence limited to 2 LSB Worst case one channel vs. all	

PicoTDC new version

PicoTDC v2

New features available on version 2:

- Configurable pulse generator on chip
- Trigger from channel 1

Production schedule:

- PicoTDC v2 at CERN now to be bonded on test board
- Functional and timing performance test of PicoTDC v2 bonded on PCB, in March – April 2020
- Functional and timing performance test of PicoTDC v2 on generic package in March – April 2020
- Functional and timing performance test of PicoTDC v2 in final package in July - September 2020

Share point link for info and material: http://cern.ch/PicoTDC

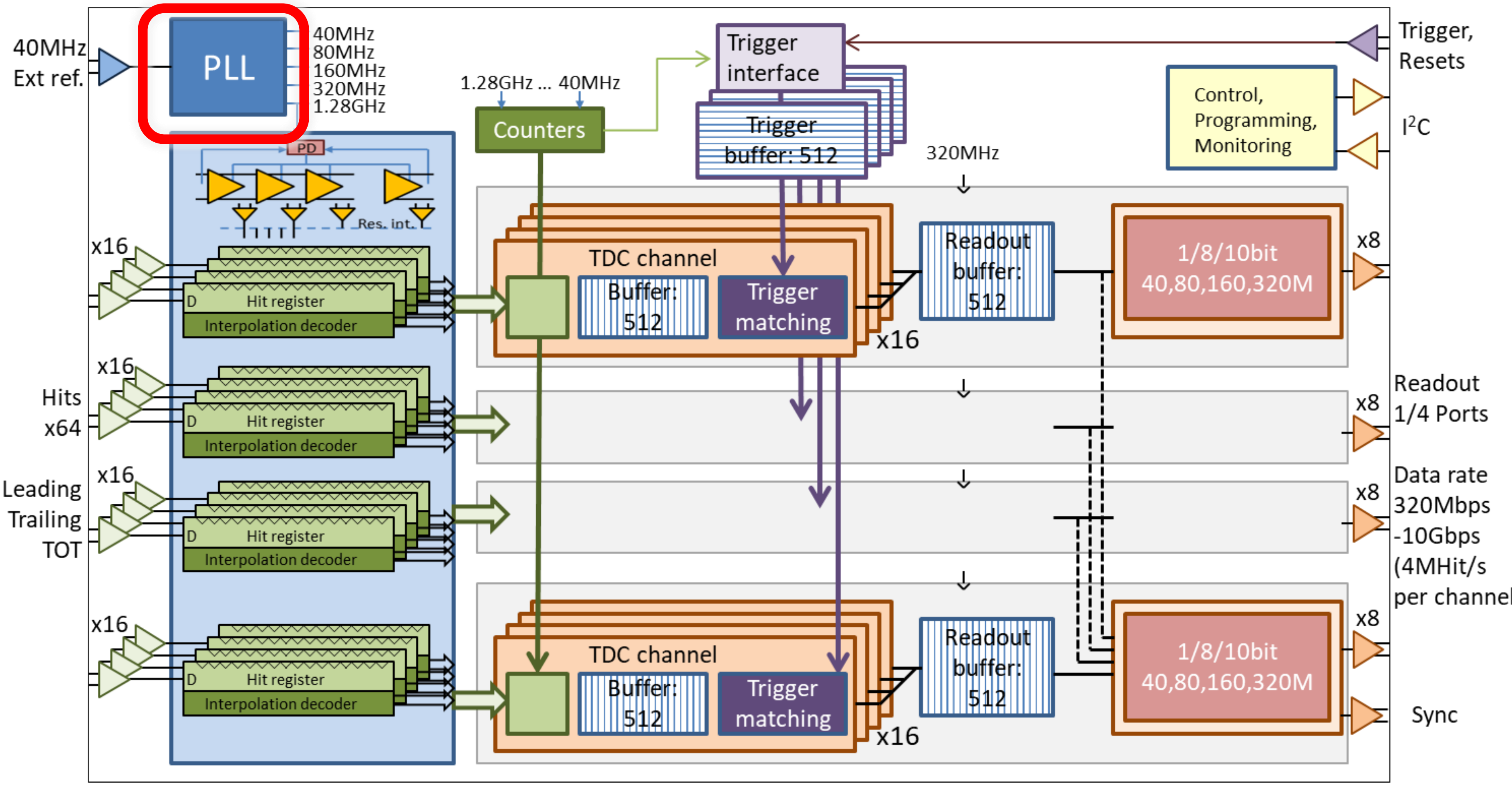
For more info and request access to the material write to:

Jorgen Christiansen <Jorgen.Christiansen@cern.ch>

Samuele Altruda <samuele.altruda@cern.ch>

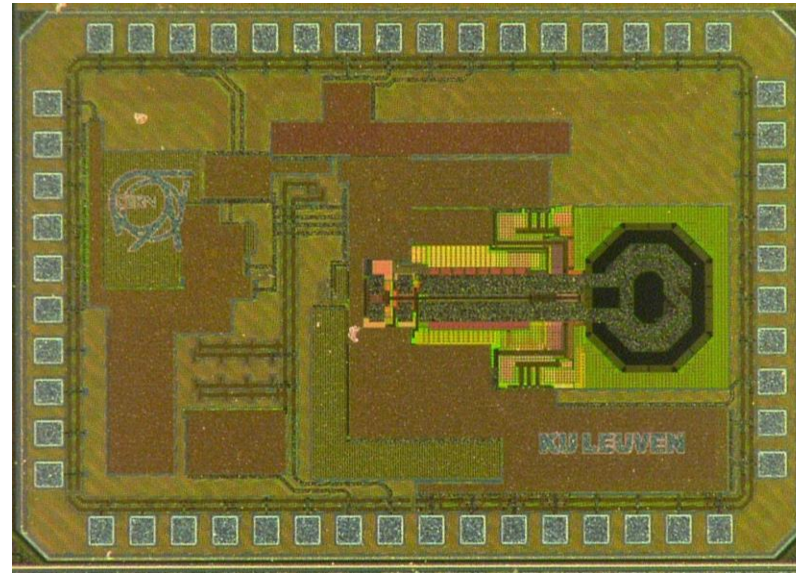
CERN

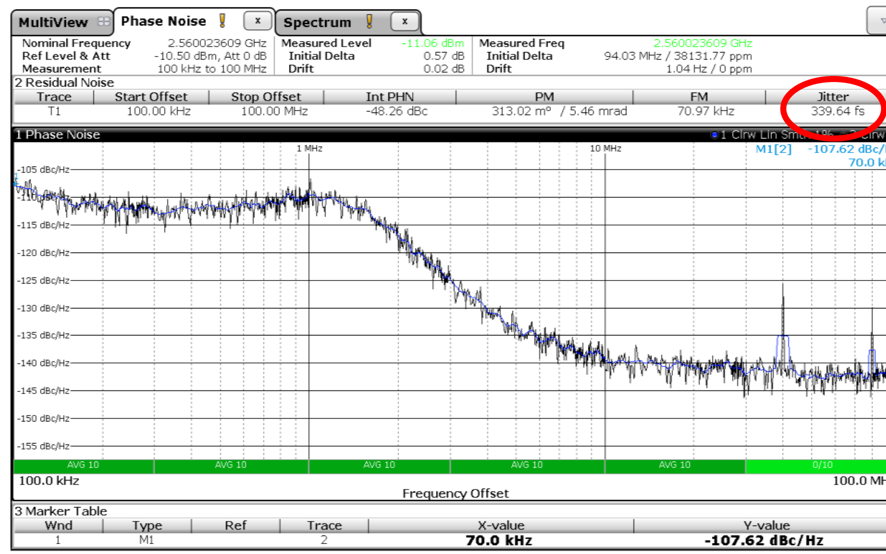
picoTDC Architecture



Low Jitter PLL

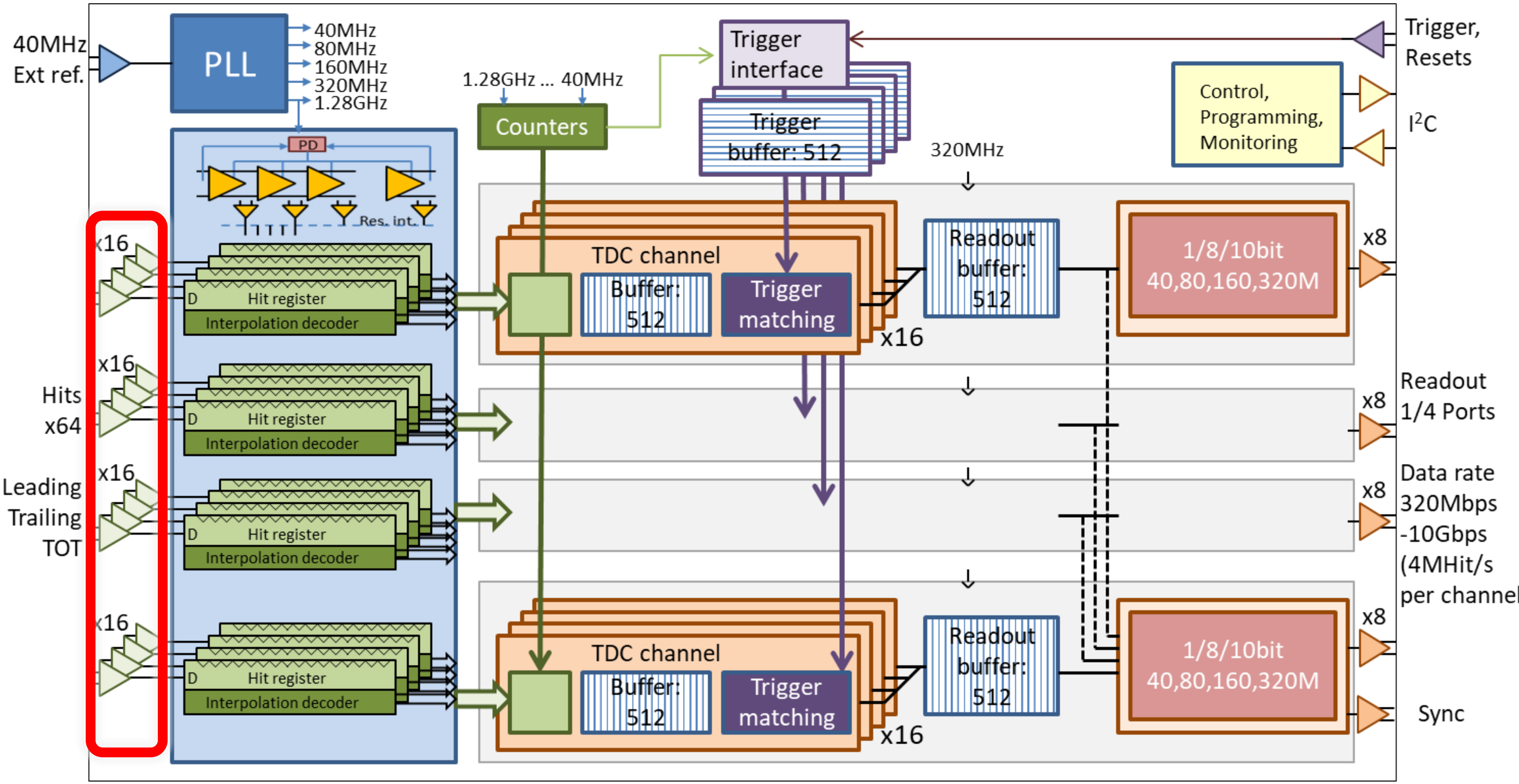
- Clock multiplication from 40MHz to 1.28 (2.56) GHz
  - Low jitter critical
  - Jitter filtering of 40MHz clock to the extent possible
    - 40MHz reference MUST be very clean
  - LC based oscillator
- Design: Jeffrey Prinzie, KU Leuven
- Prototyped & Tested
- Measurements very promising (340fs RMS jitter)





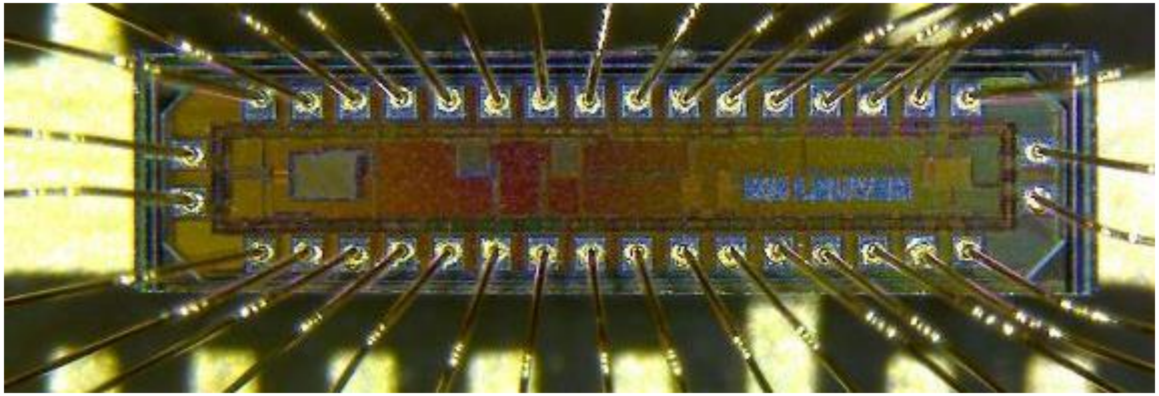
Phase Noise vs. Freq. Offset

picoTDC Architecture

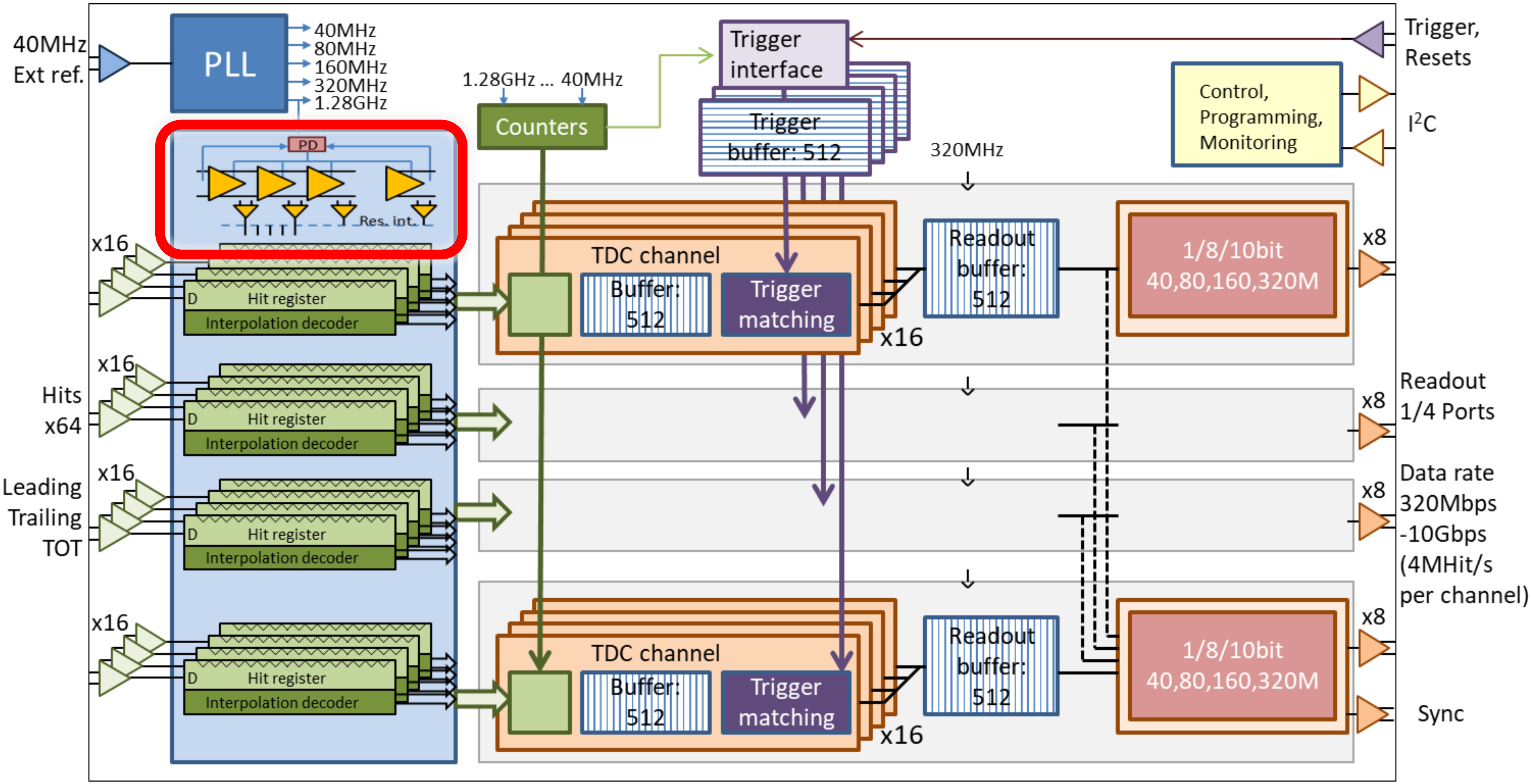


Hit Receivers

- Differential receivers optimized for ultra-low jitter, low power
- Full Range (common mode 0V .. VDD=1.2V), somewhat LVDS-compatible
- Highest speed @~800mV common mode
- Optimized for 200mV Peak-Peak amplitude
- Design: Bram Faes, KU Leuven
- Prototyped & tested

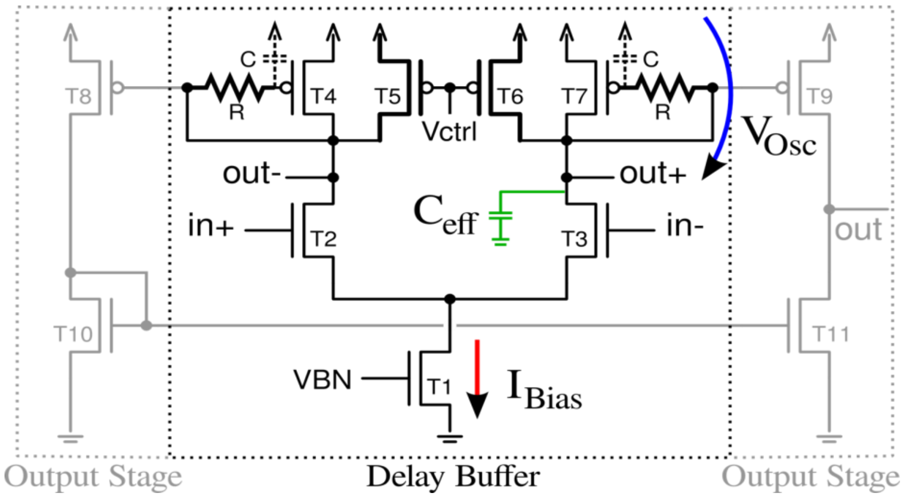


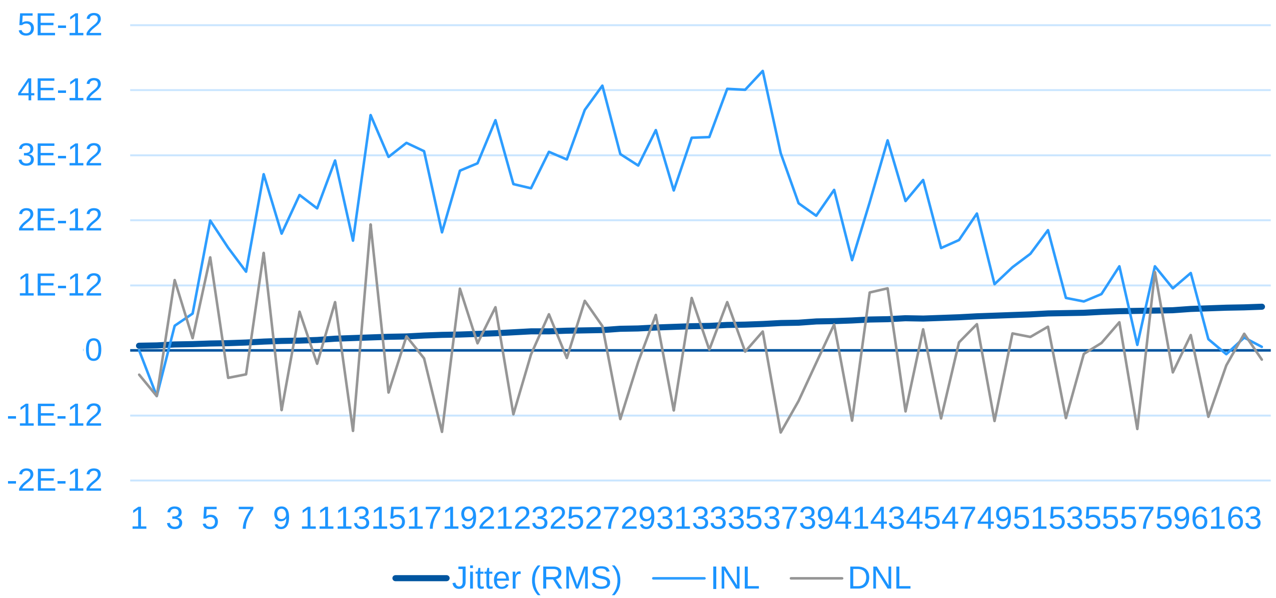
picoTDC Architecture

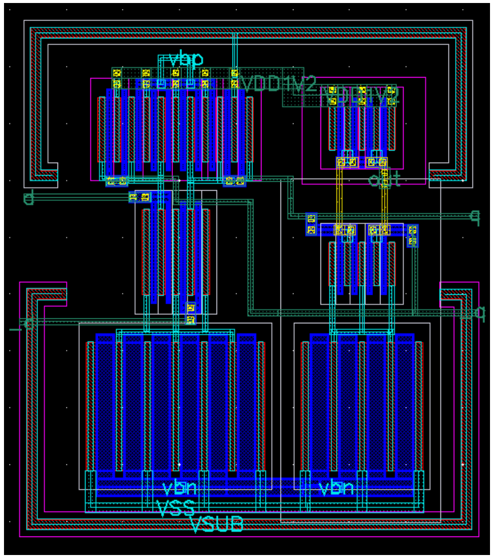


1st Stage: DLL

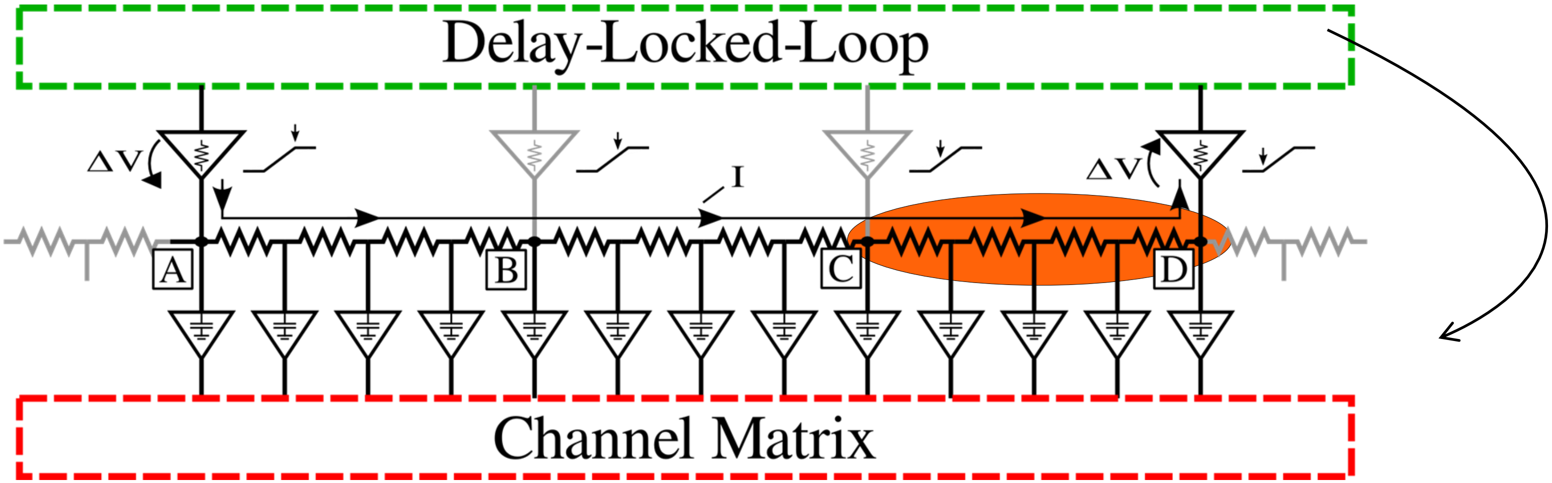
- 64 taps, 12.2ps delay
- Self-Calibrating
- Jitter not as critical, doesn't pile up

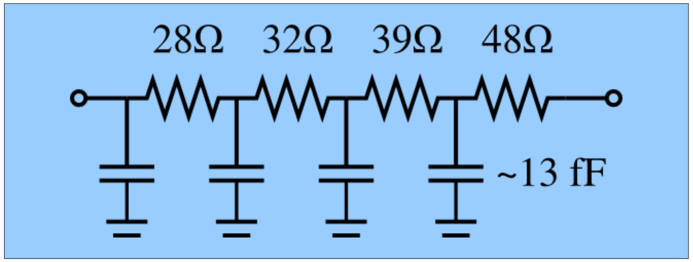


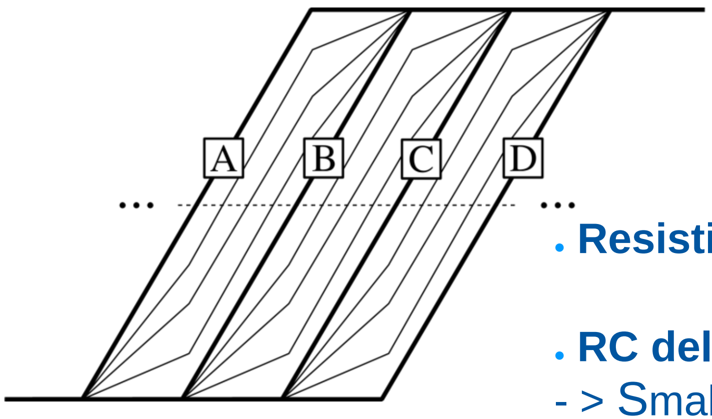




2^{nd} Stage: Resistive Interpolation





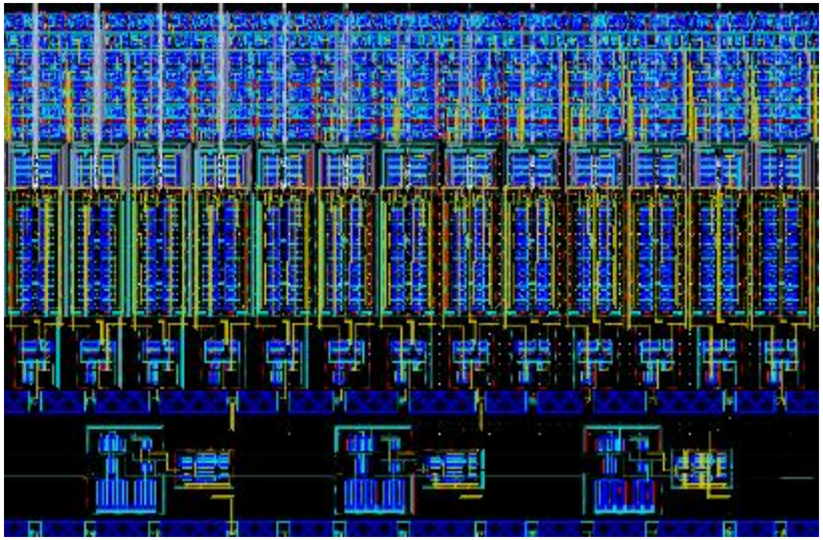


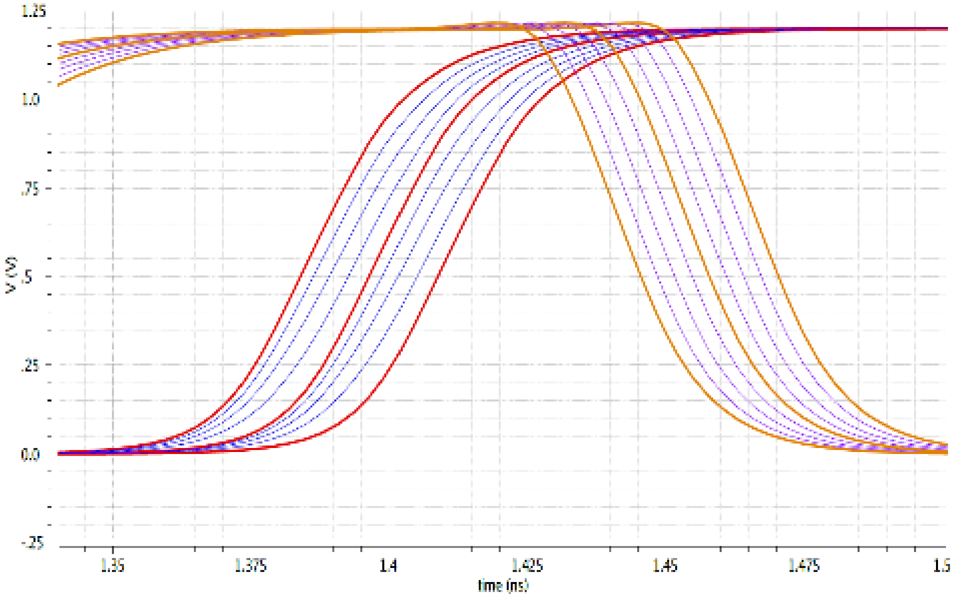
- **Resistive voltage divider**
  -> Signal slopes longer than delay, stabilized by DLL
- **RC delay** (capacitive loading)

- > Small resistances, small loads
- > Simulation based optimization of resistor values

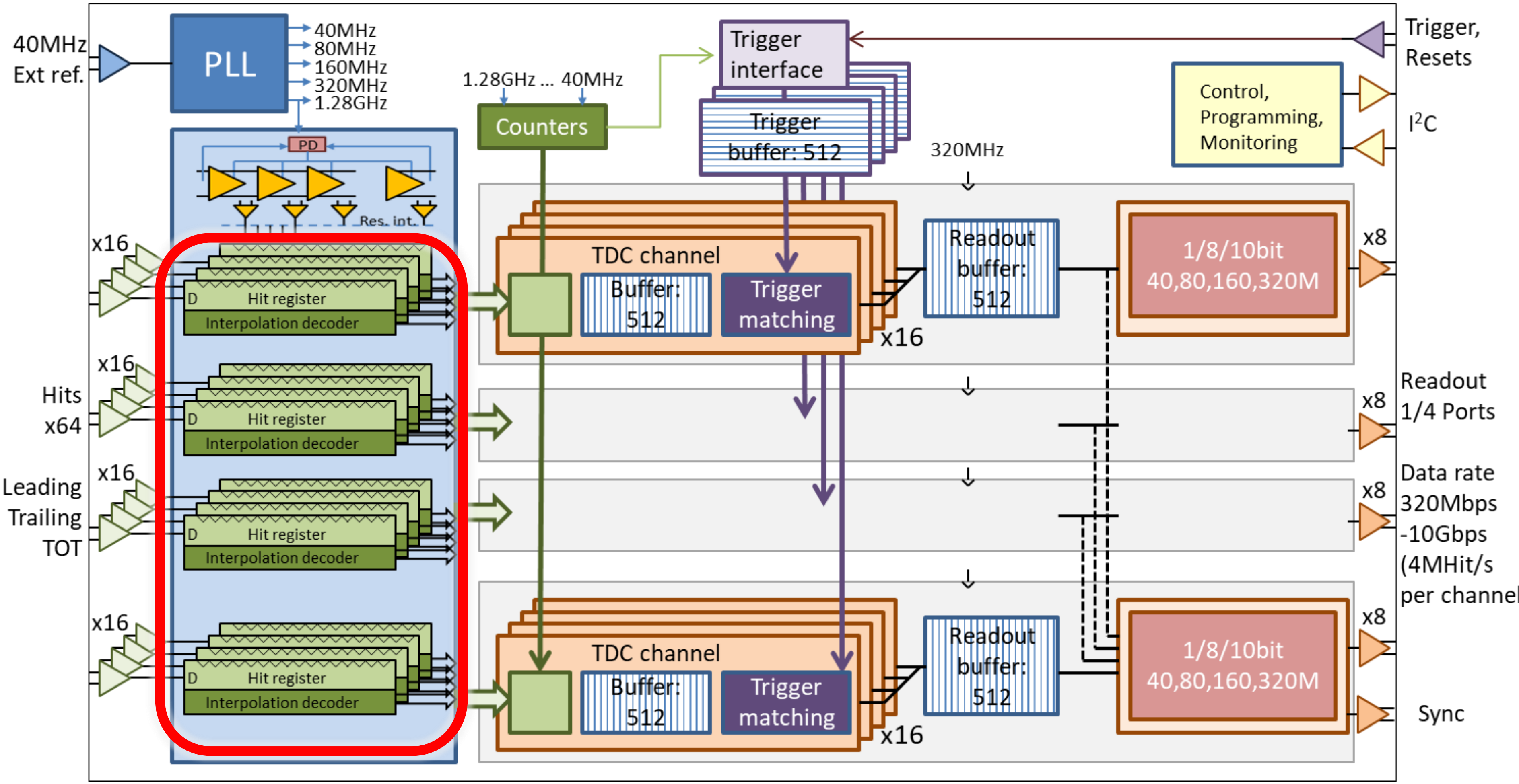
Finecode Drivers and Alignment

- Get down to 3ps bins
- Drivers: tapered buffers, each driving 32 FFs
- Phase alignment separate for each half



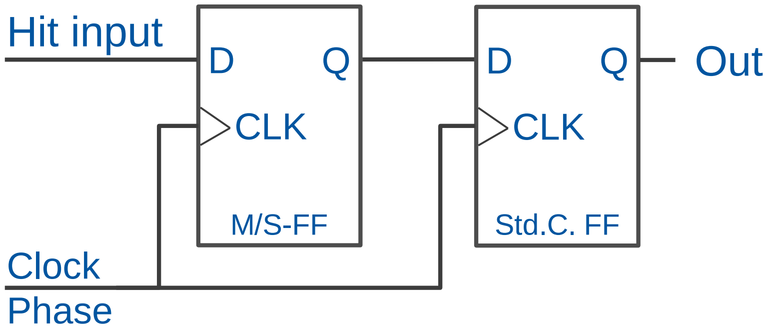


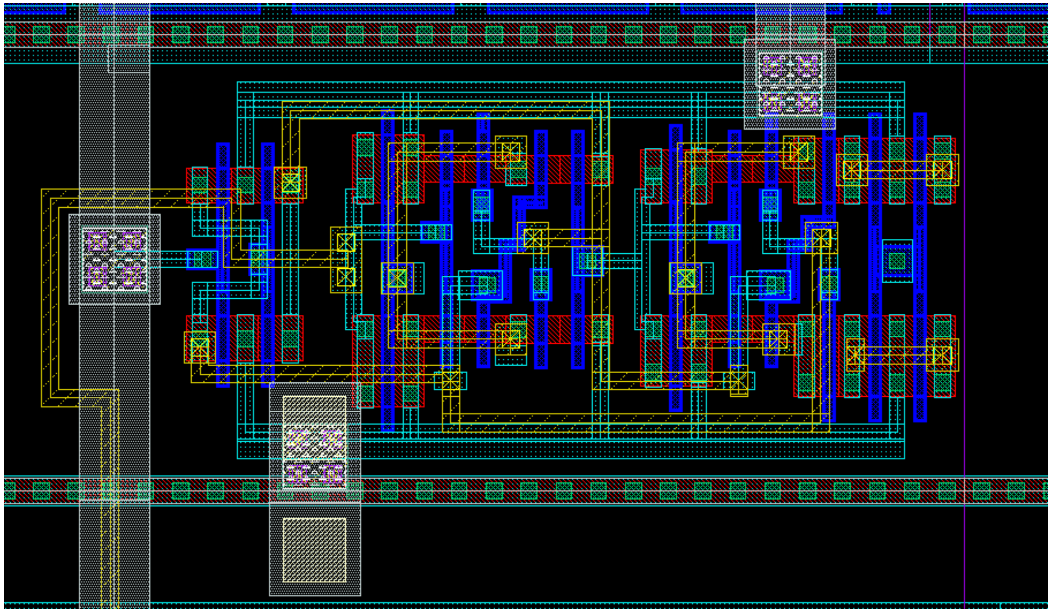
picoTDC Architecture



Capture Flip Flops

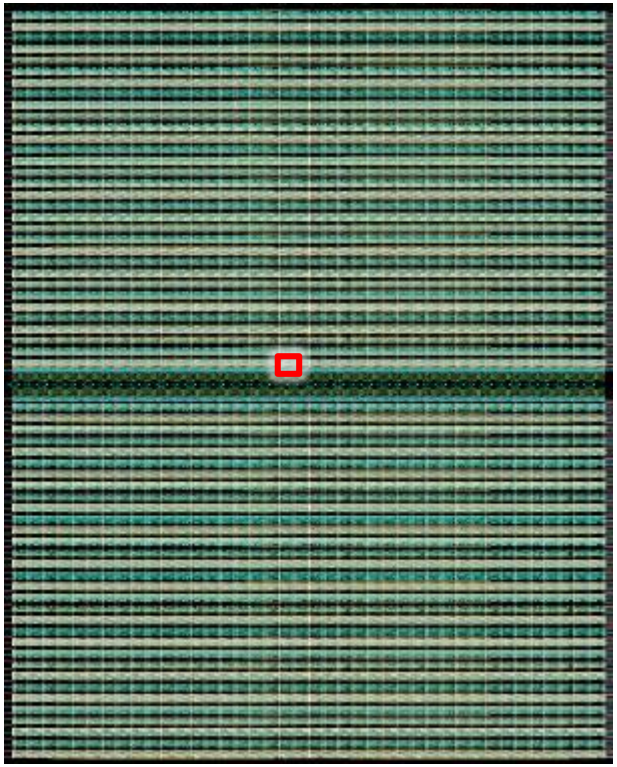
- Revisited design, timing vs. power very critical, 16k capture Flip Flops running @1.28GHz
- Highly optimized M/S Flip Flop followed by standard cell Flip Flop for metastability resolution
- Monte Carlo simulations show a mismatch of 800fs RMS, noise influence of 240fs RMS





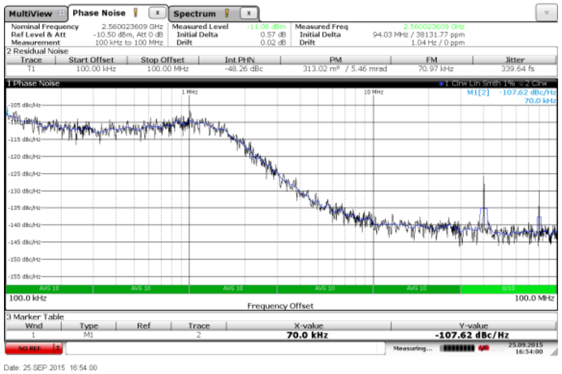
Full Timing Macro

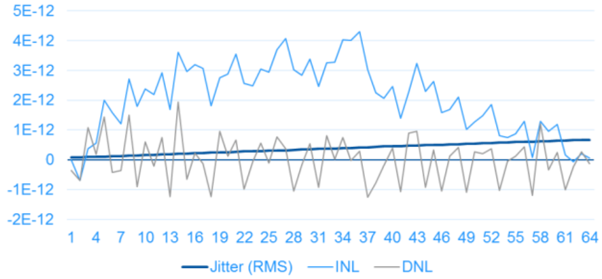
- 64 channels, DLL and resistive interpolator in the center
- Hit signal input on the left, output on the right
- Hit decoding fully synchronous, custom layout with standard cells
  - Decoding of one hit per 0.8ns
- 1.6mm x 2.0mm

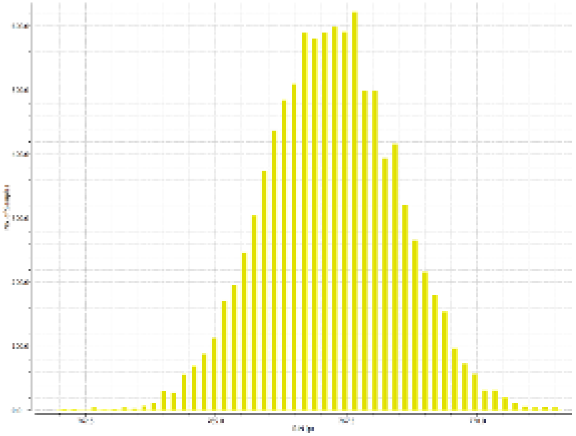


Sources of Measurement Deviation

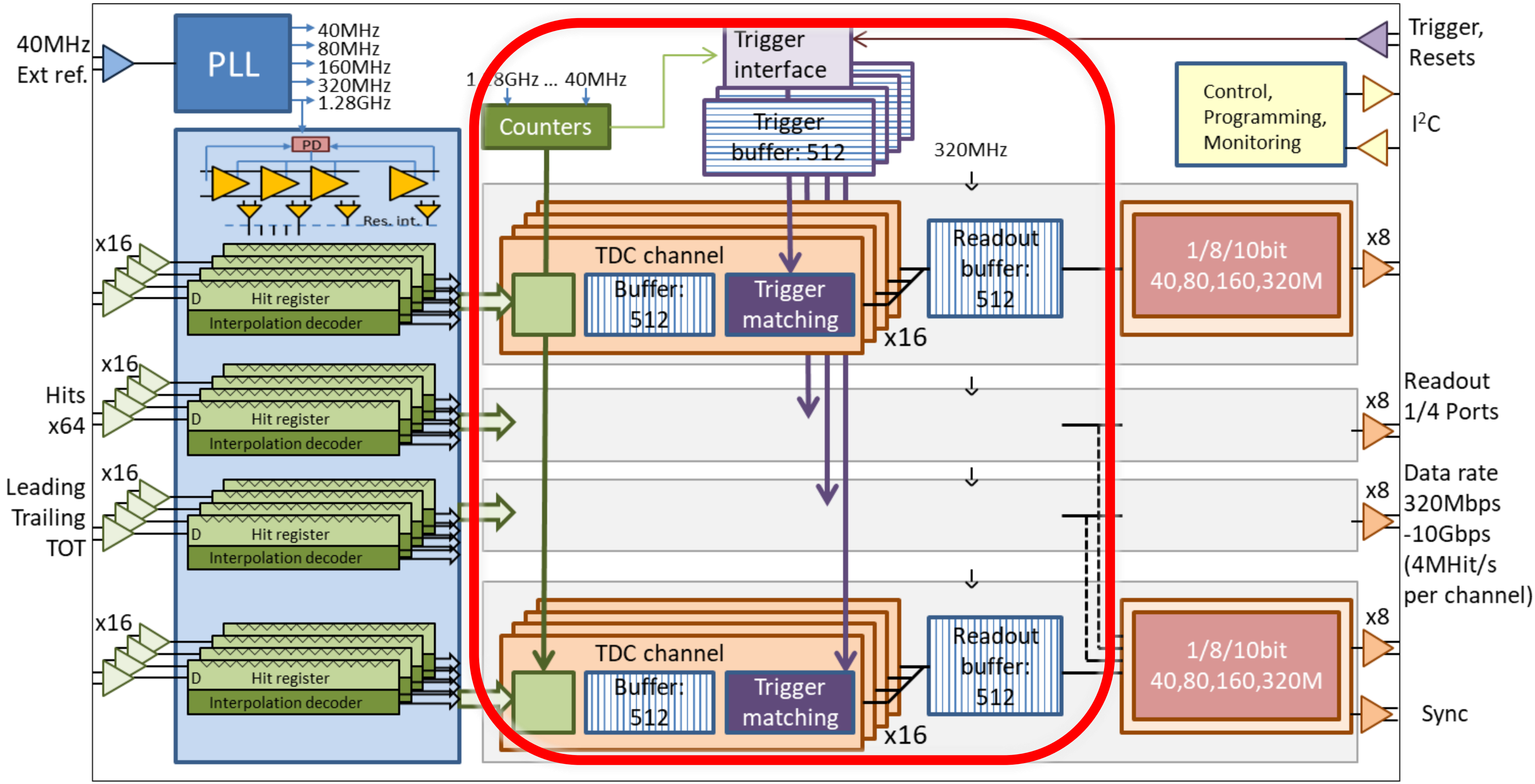
- Bin size 3ps -> 880fs RMS
- PLL: 350fs RMS phase Jitter
- DLL: 400fs RMS phase Jitter, INL/DNL can be adjusted
- Clock Distribution: <500fs jitter
- Capture FFs: <1ps mismatch (DNL)
- Hit receivers: <1ps jitter
- ➢ ~1.75ps RMS total deviation
- External sources: input clock jitter, signal preprocessing







picoTDC Architecture



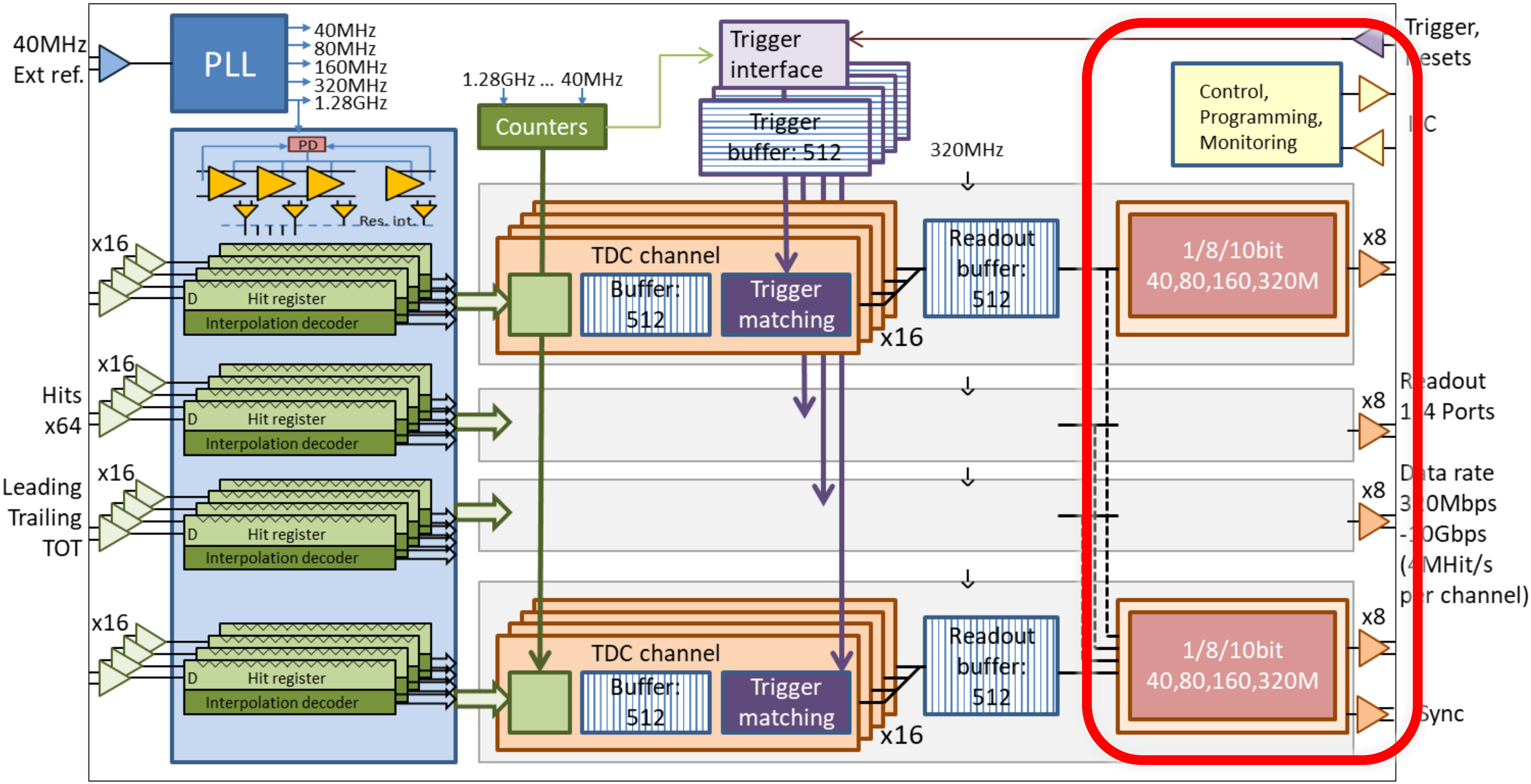
Constraints on Hit Signals

- One edge per 1.28GHz-Cycle (~0.8ns)
- Internal analog glitch filter after hit receiver
  - Filter time can be programmed to ensure 0.8ns
  - Or up to 10ns for filtering e.g. oscillations
- Small derandomizer (4 hits) for each channel running @1.28GHz
- Sustainable rate to channel buffer 320MHz, trigger matching running @320MHz for each channel separate

Logic Features

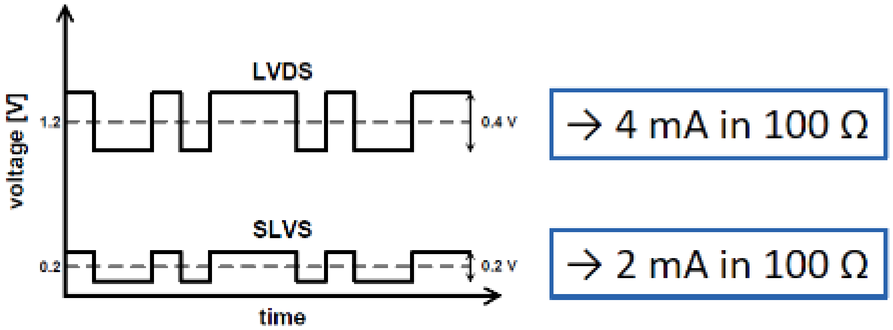
- Triggered with configurable latency and length, overlap possible, or untriggered
- Naturally overflowing counter used for calculating trigger matches, TOT etc.
- Counter with arbitrary overflow and reset for machine cycle, can be inserted in event headers when triggered

picoTDC Architecture



Electrical Interfaces

- Hits: Differential (LVDS “compatible”, common mode from 0.2V to 1.2V)
  - Highest speed (resolution) @ ~800mV common mode
- Time reference: 40MHz differential
  - Low jitter reference critical for high time resolution
- Trigger/Event-Rst/BX-Rst/Reset: Sync Yes/No
- Control/monitoring: I^2C at CMOS 1.2V-levels
- Readout: 4 readout ports of 8 differential signals
  - Common mode 0.6V, programmable current 1-5mA
  - Compatible with LpGBT and FPGAs
- Packaging: 400 BGA (1mm pitch)



Config / Control / Status Interface

- I^2C Interface, up to 1MBit/s
- 1.2V CMOS Levels
- 348 Bytes configuration / control
  - Additional 322 bytes delay adjust
- 300 Bytes status

Readout

- 1 or 4 differential readout ports with 8 bits
  - 40 - 320MHz
  - Bandwidth:
    - Min 320Mbits/s (~0.15 Mhits/s per channel)
    - Max 10Gbits/s (~4 Mhits/s per channel)
- Readout data: 32 bit words
  - TDC data, headers, trailers etc.

32 Bit Frames

TDC measurement

Type (1)=0	TDC data (31)

Event headers (up to two)

Type (4)=100?	Field A (13)	Field B (13)	00

Possible fields: Event ID, Bx ID, Natural ID

Event trailers

Type (4)=1010	Event ID (13)	Hit Count (13)	00

Channel group separator (for single readout port)

Type (4)=1111	Chan-Grp-ID (2)	0x0000000 (26)

Idle frame

Type (4)=1101	0x0D0D0D0 (28)

Absolute TDC data

FULL TDC data, **DEFAULT FORMAT**

Type (1)	Channel (4)	Edge (1)	Coarse cnt (13)	Med. cnt (5)	DLL int (6)	Res int (2)

Relative to Trigger

Triggered with relative time: Same as absolute

Type (1)	Channel (4)	Edge (1)	Coarse cnt (13)	Med. cnt (5)	DLL int (6)	Res int (2)

B: Triggered with relative leading and TOT: Same as absolute Lead. + TOT

Type (1)	Channel (4)	Leading (16)	TOT(11)

Type (1)	Channel (4)	Leading (19)	TOT(8)

Leading + TOT

- Packet Type: 1bit
- Channel ID: 4 bits, for single port readout +2 bit group separator
- Leading: 16/19 bits
  - Large dynamic range
    - 16bit 3ps resolution: 200ns
    - 19bit 3ps resolution: 1600ns
  - **Programmable part of full 25bits leading TDC**
  - (**Relative to trigger to be useable**)
- TOT (Relative to leading): 11/8 bits
  - Short dynamic range:
    - 8bit 3ps resolution: 780ps
    - 11bit 3ps resolution: 6.1ns
  - **Programmable part of full 25bits TOT difference**
    - TOT assumed to be used for offline time-walk correction of leading.
- Alternative: Readout of Individual Leading and Trailing edges with full range/resolution
  - 2x readout bandwidth

Type (1)	Channel (4)	Leading (16)	TOT(11)

Type (1)	Channel (4)	Leading (19)	TOT(8)

Estimated Power Consumption

Highly dependent on hit rate, values based on 1 MHz per channel

- High resolution, 64 channels: 1300mW
- High resolution, 32 channels: 900mW
- Low Resolution, 64 channels: 850mW
- Low Resolution, 32 channels: 550mW