Integration of the HGND readout into the BM@N experiment data acquisition system

D. Finogeev, F. Guber, A. Izvetnyy, S. Musin, D. Serebryakov
INR RAS, Moscow

Acknowledgements:
This work was carried out at the Institute for Nuclear Research, Russian Academy of Sciences, and supported by the Russian Scientific Foundation grant №22-12-00132.

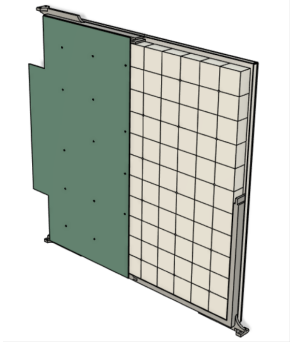
LXXIV International conference Nucleus-2024: Fundamental problems and applications
July 3, 2024

Outline

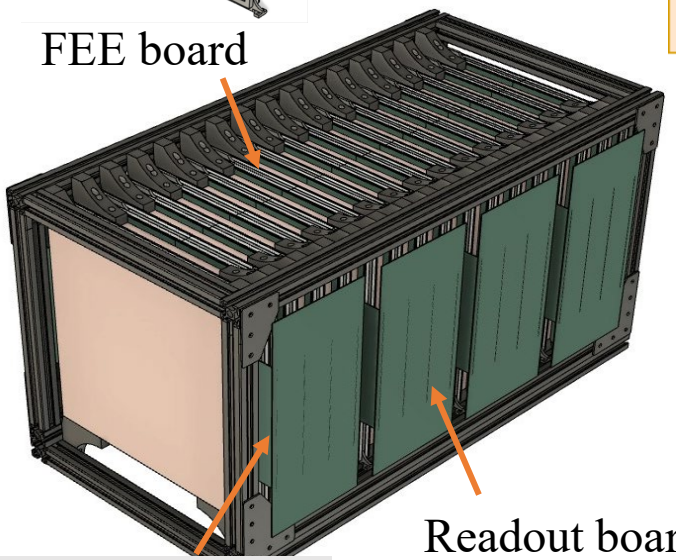
- HGND readout topology
- 100ps FPGA based TDC
- Status of the readout board development
- DCS & readout software architecture

FEE & readout architecture

- 16 layers with scintillation matrix 11X11
- 16 LED boards
- 32 FEE boards
- 8 Readout boards
- 3 FPGA per board
- 84 channels per FPGA
- 2000 channels in total

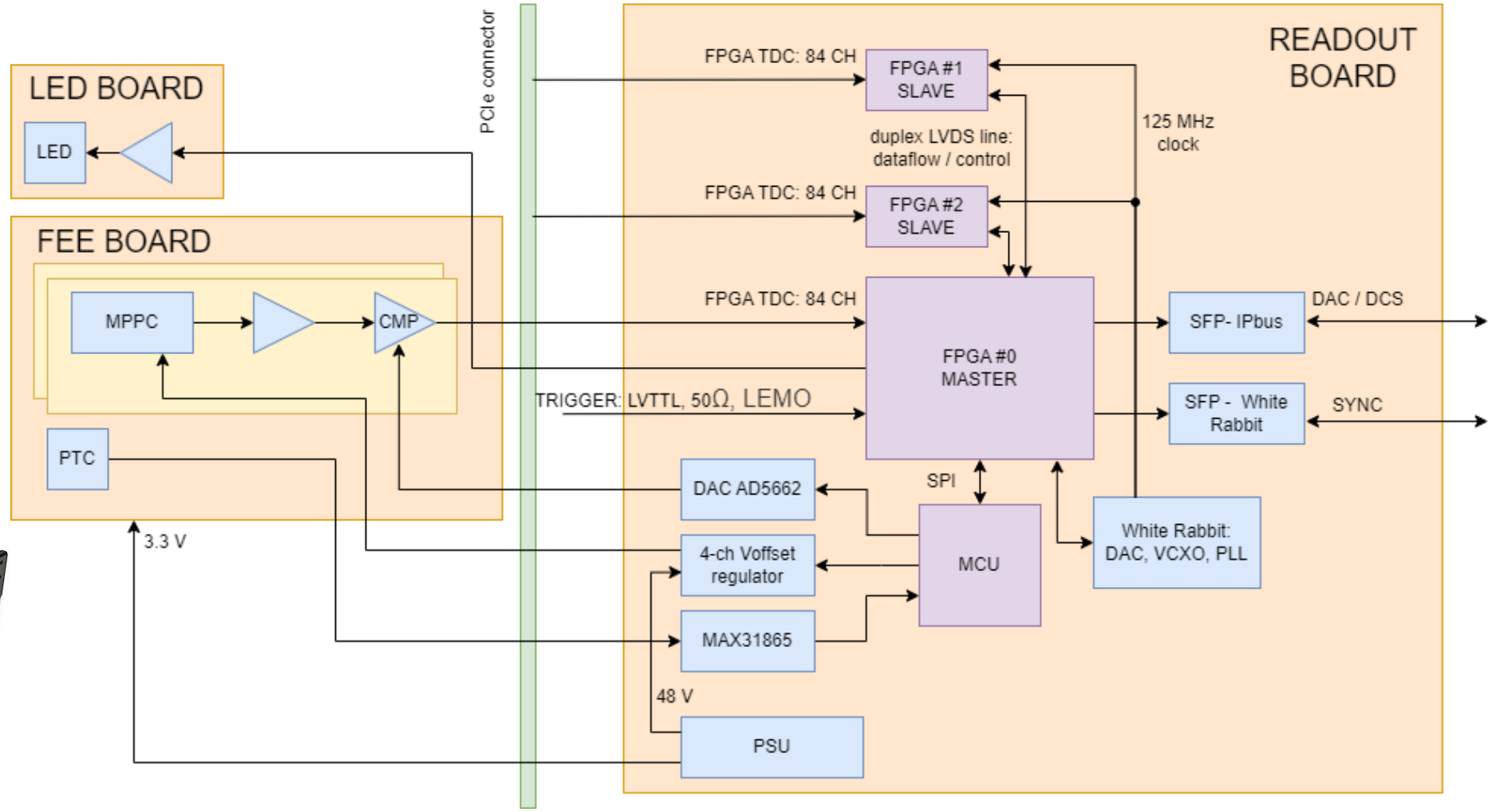


FEE board



Readout board

PCIe connector



Readout & trigger

- ***100 ps*** TDC is implemented in Kintex 7 FPGA, 84 channels per FPGA chip (2000 total)
- ***White Rabbit*** (WR) is used for event's time synchronization (8 links total):
  - TDCs use clock sourced from WR synchronous to whole BM@N
  - WR timestamps are assigned to measured events

- Ethernet UDP protocol (***IPbus*** [1]) is used for data forwarding and board control
- Local network connect readout boards (8 ethernet links) with FLP

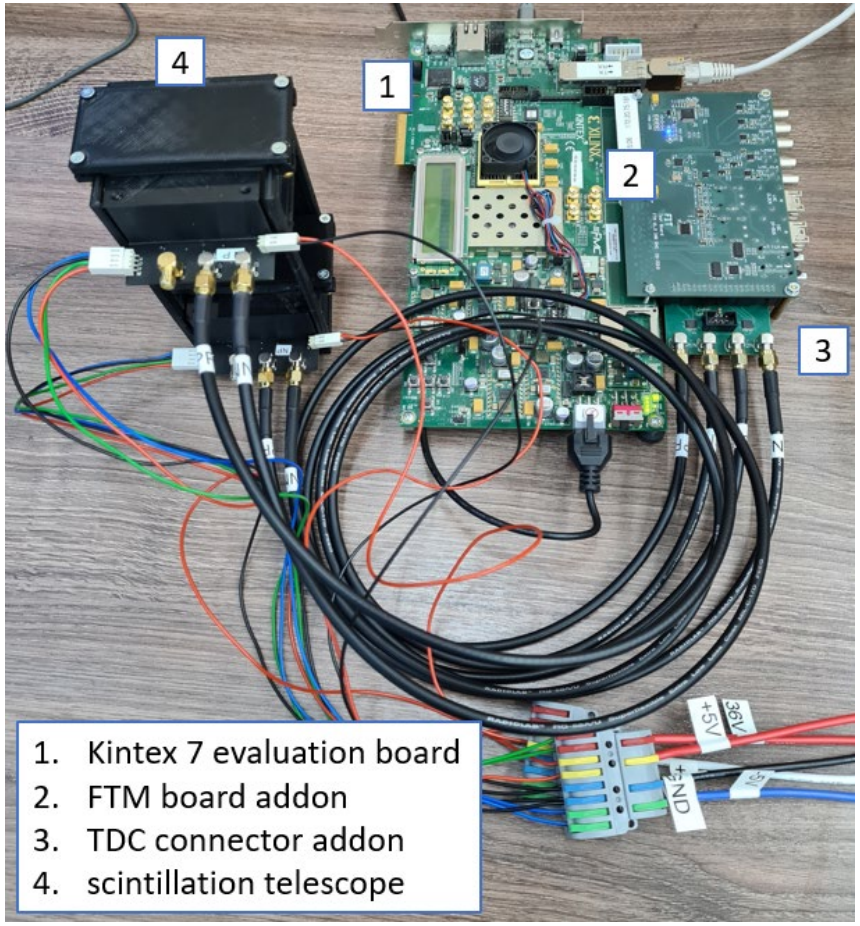
- The maximum HGND channel load is 3 kHz. The event size is 7x16 bits. The upper limit per link is not exceed ***100 Mbit/s***. ***The continuous readout*** is implemented without busy signal.

- The trigger is processed on FLP site:
  - Trigger signal is connected to TDC channel and digitized with WR timestamp in FPGA
  - Message trigger accompanied by a timestamp is transmitted to FLP for event selection

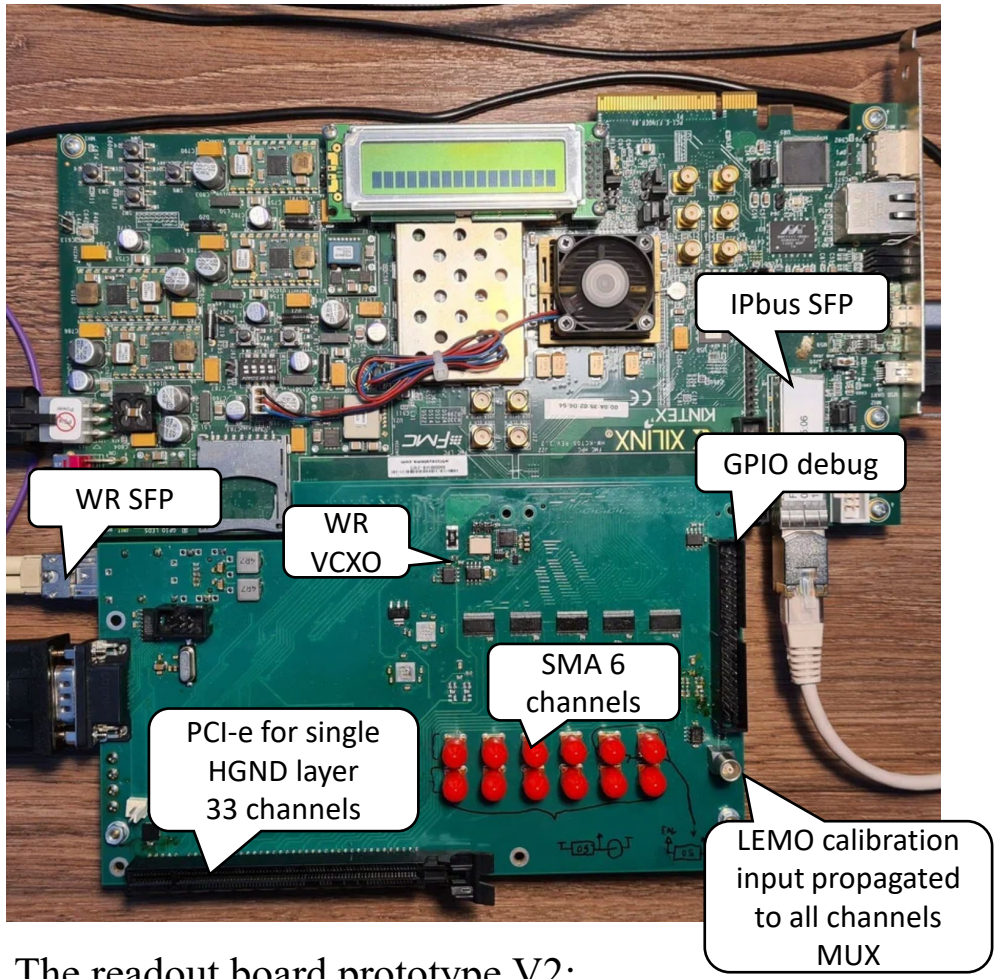
[1] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, A. Rose, A. Thea and T. Williams, ***IPbus: a flexible Ethernet-based control system for xTCA hardware***, *JINST 10 (2015) no.02, C02019.*

HGND readout v1 and v2 prototypes

based on the Kintex 7 evaluation board (KC705)

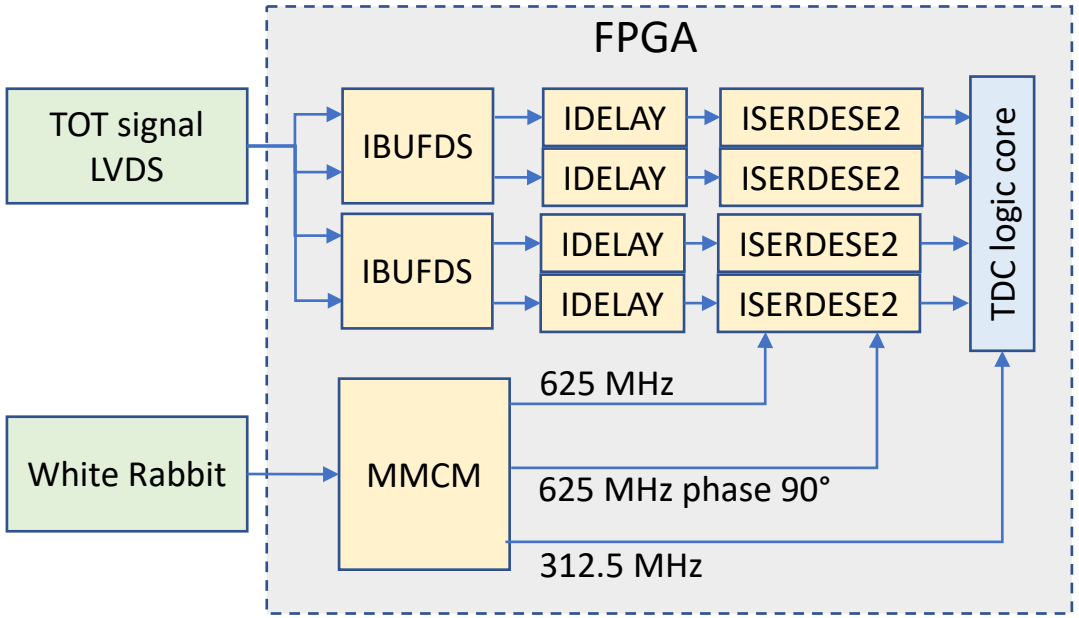


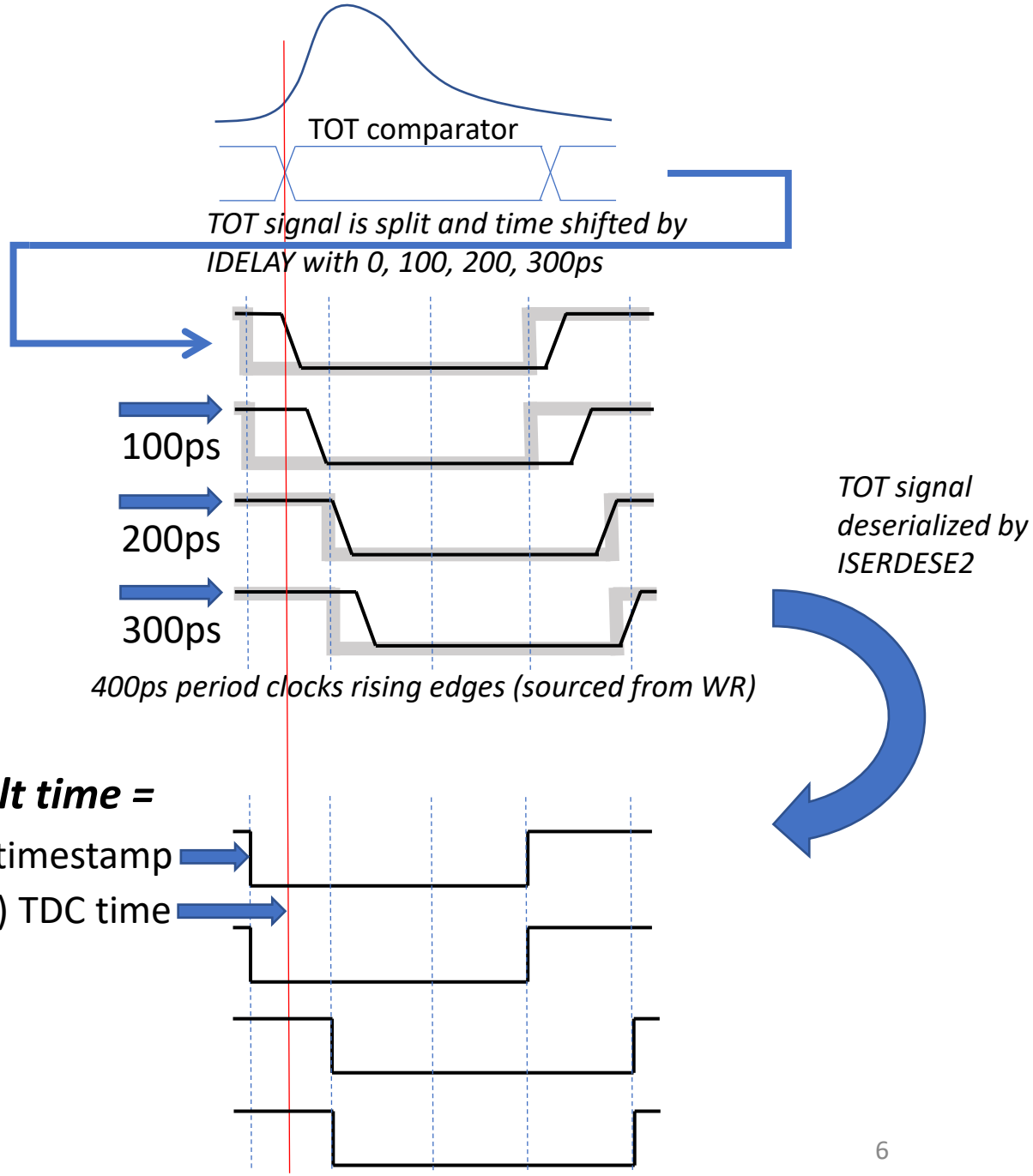
The 2-channels TDC prototype V1 connected to HGND scintillator cells telescope



The readout board prototype V2:

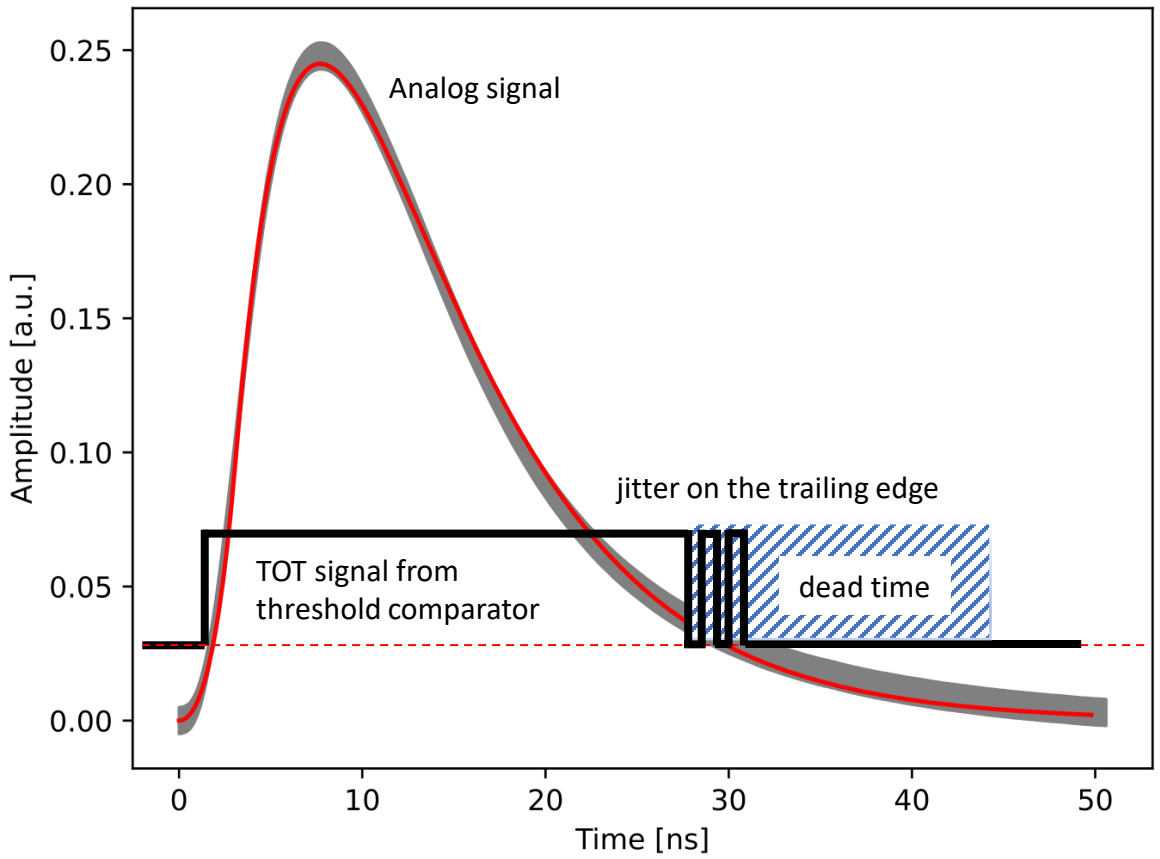
The 100ps FPGA TDC principle of operation



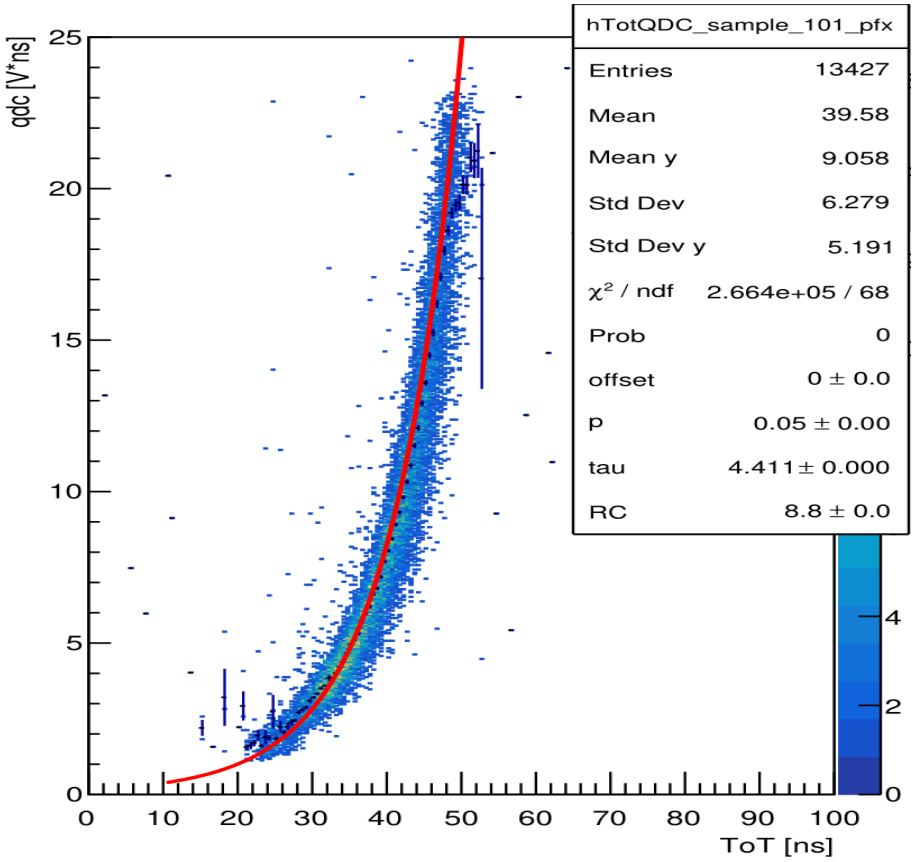


The TDC is **based on the Kintex-7** input serial-to-parallel converter with oversampling capability and programmable delay. The design is **based on Xilinx recommendations**, and uses **only documented features** of the FPGA within its specifications.

TDC Time Over Threshold (TOT)



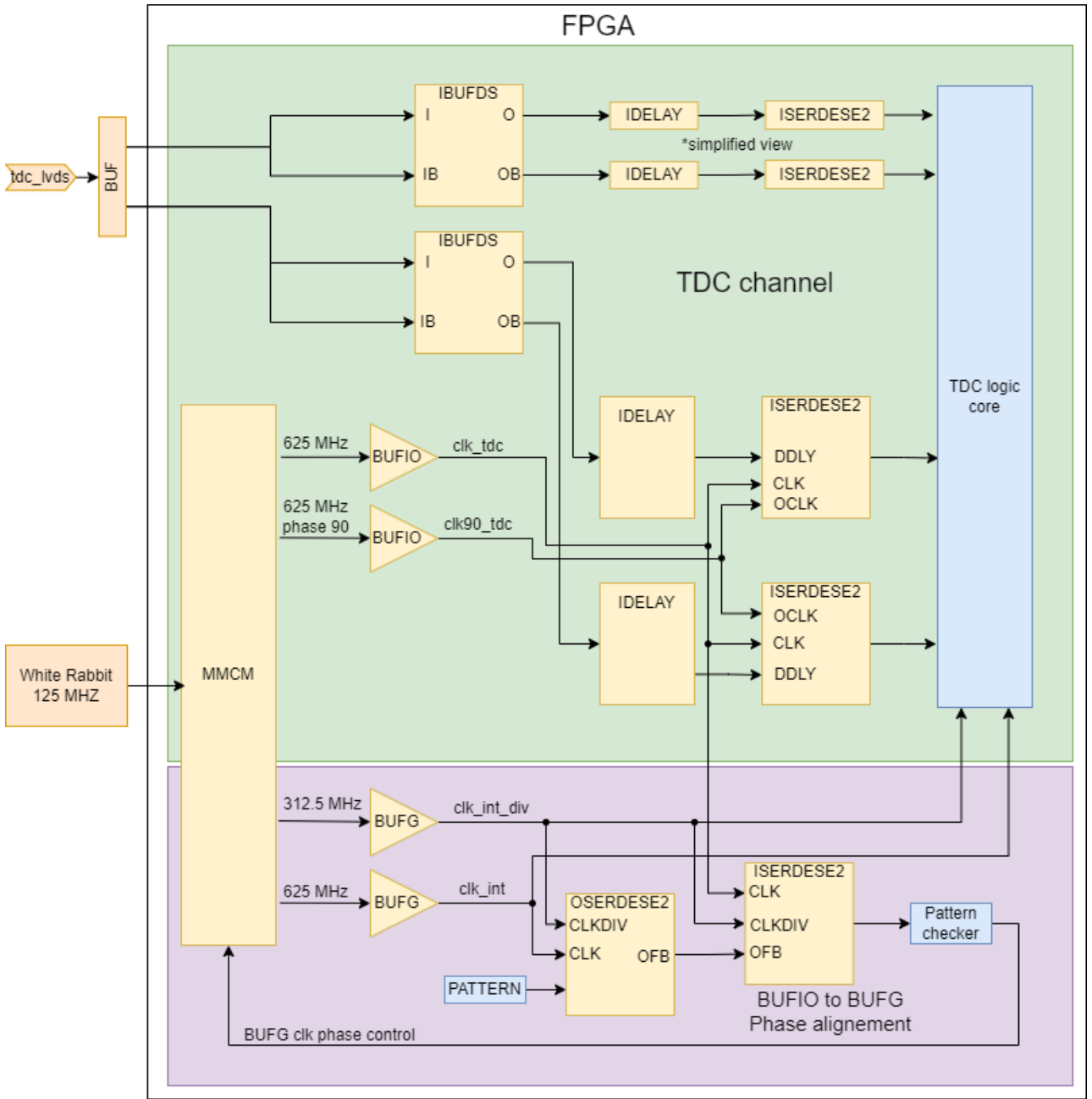
Amplitudes vs TOT time with analytical forecast

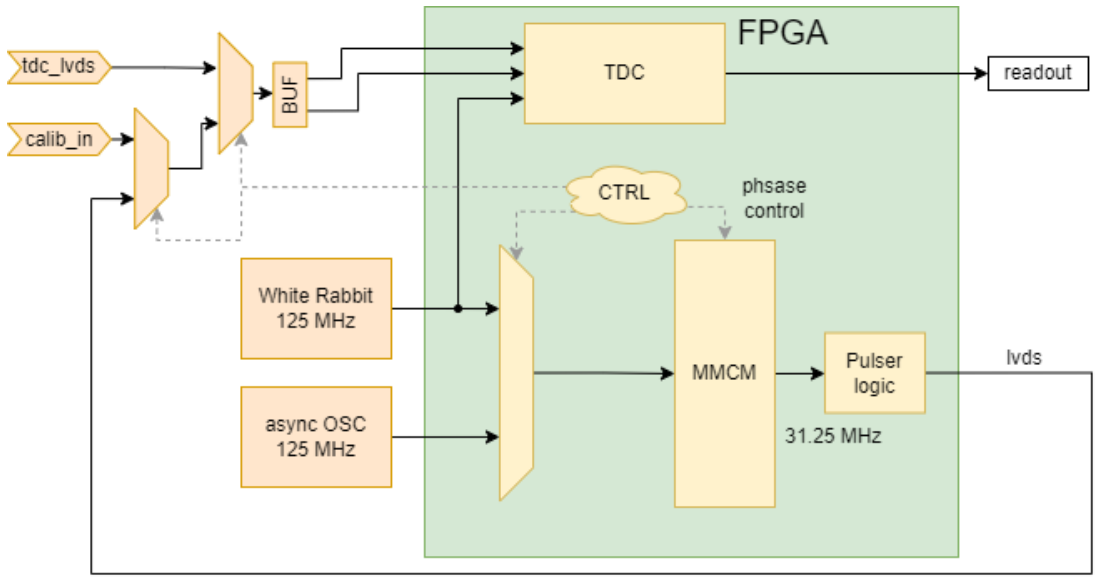


TOT amplitude resolution is in range 14 - 22%

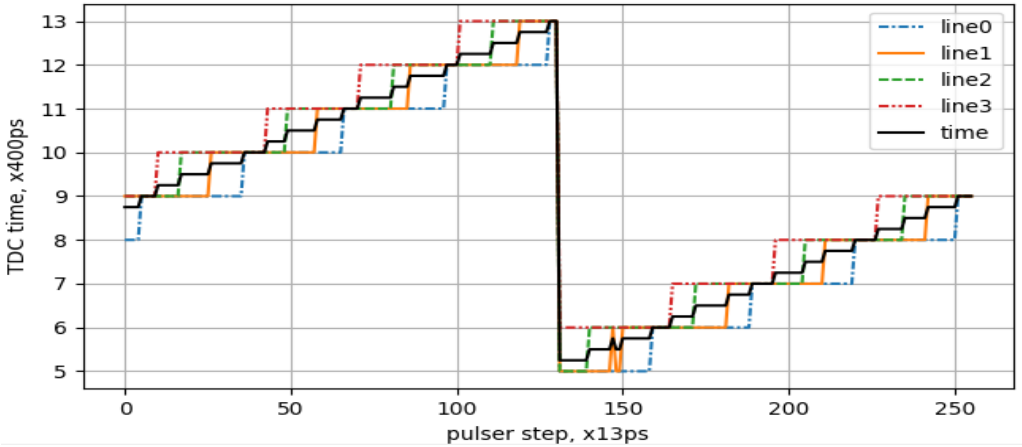
- The threshold is tunable around 20 mV
- Signals length range is 20 – 60 ns
- Signals less than 6.4 ns are rejected for noise reduction
- Dead time is tunable in range 30 – 200 ns for comparator jitter filtering
- Minimum TOT time and dead time available in FPGA TDC are 3.2 ns

FPGA TDC and calibration pulser clocks layout

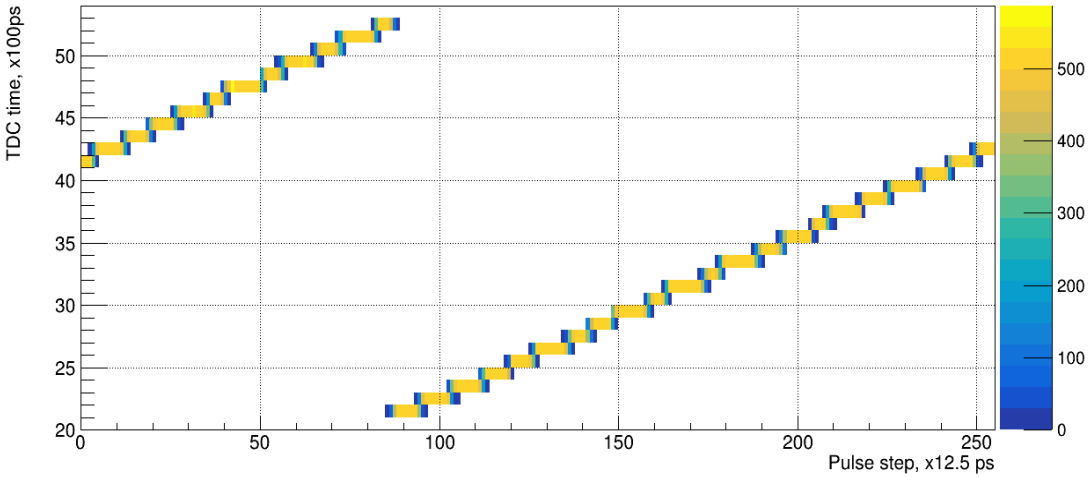




The FPGA TDC time scan results



Four TDC lines and the resulting time dependence on the pulse time shift are shown. Pulses are generated by FPGA MMCM synchronously to the TDC clock with a phase step of 12.5ps. The single scan pass was taken with a digital FPGA logic analyzer.

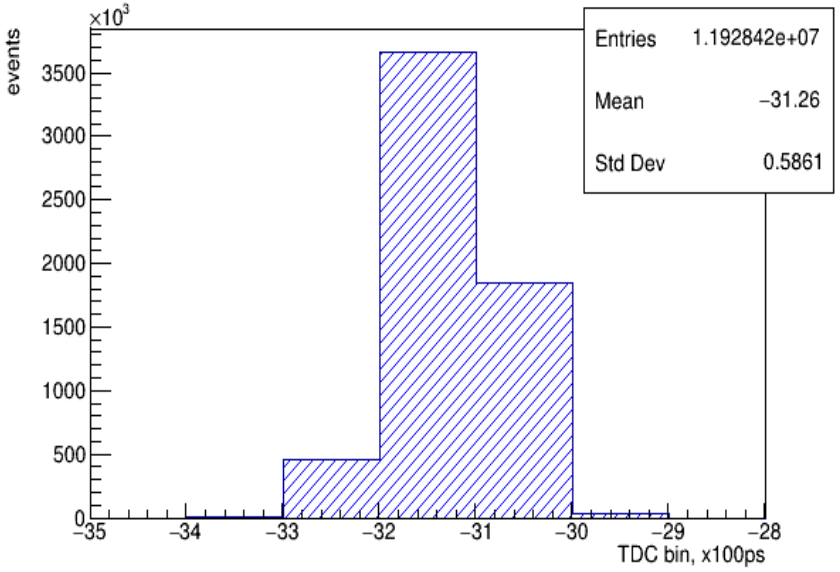


The TDC time dependence on pulse time shift. Pulses are generated by FPGA MMCM synchronous to TDC clock, and the time step is 12.5 ps. Data was taken with PC readout, 1000 events per single time shift step.

The FPGA TDC test results

PROTO_V1

The time difference distribution between two FPGA TDC channels measured with the data generator DG2040 (Cycle-to-Cycle Jitter 5ps).

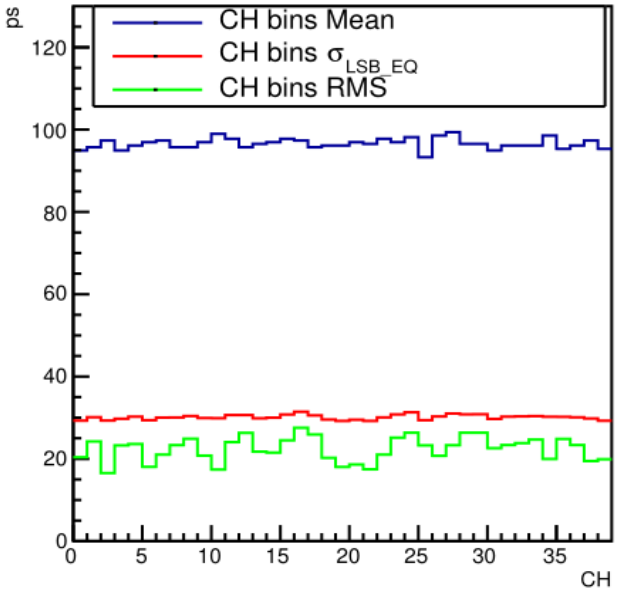


The RMS of single TDC channel is 42ps.

[1] D. Finogeev, F. Guber, A. Izvestnyy, N. Karpushkin, A. Makhnev et al., ***Development of 100 ps TDC based on Kintex 7 FPGA for the High Granular Neutron Time-of-Flight detector for the BM@N experiment,*** *DOI: 10.1016/j.nima.2023.168952*

PROTO_V2

TDC bins with for all 39 channels: mean value, RMS, equivalent LSB precision. Calculated with synchronous scan



The equivalent LSB precision is ~30ps.

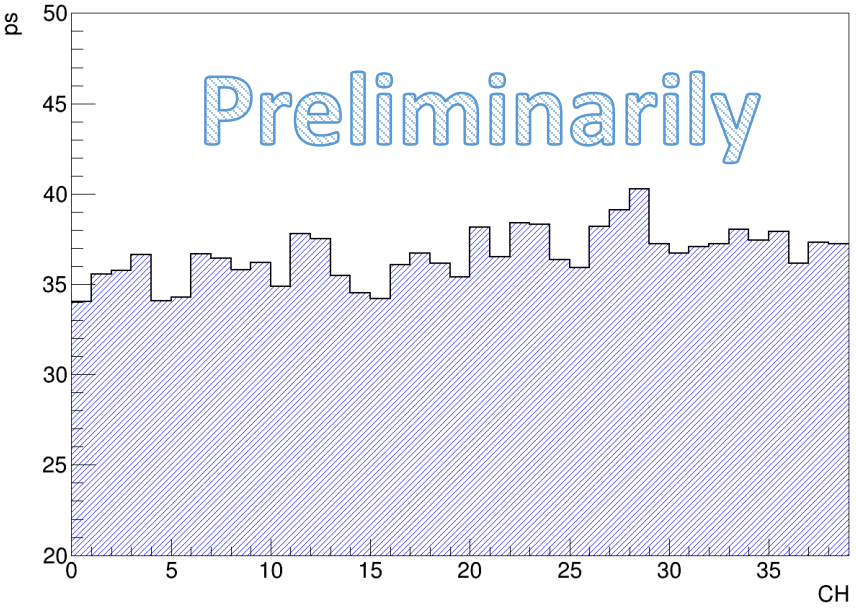
$$LSB_{EQ} = \sqrt{\frac{1}{\sum_{i=0}^{i=N-1} BinWidth[i]} \cdot \sum_{i=0}^{i=N-1} BinWidth^3[i]}$$

$$\sigma_{LSB\_EQ} = {LSB_{EQ}}/{\sqrt{12}}; \begin{cases} BinWidth[\forall i] = 100ps \\ \sigma_{LSB\_EQ} = 29ps \end{cases}$$

[2] N. Lusardi et al., "Quantization noise in non-homogeneous calibration table of a tcd implemented in fpga," DOI:10.1109/NSSMIC.2014.7431149

PROTO_V2

TDC bins precision measured with pulse length 101.2ns



The measured TDC precision is 35 – 40 ps.

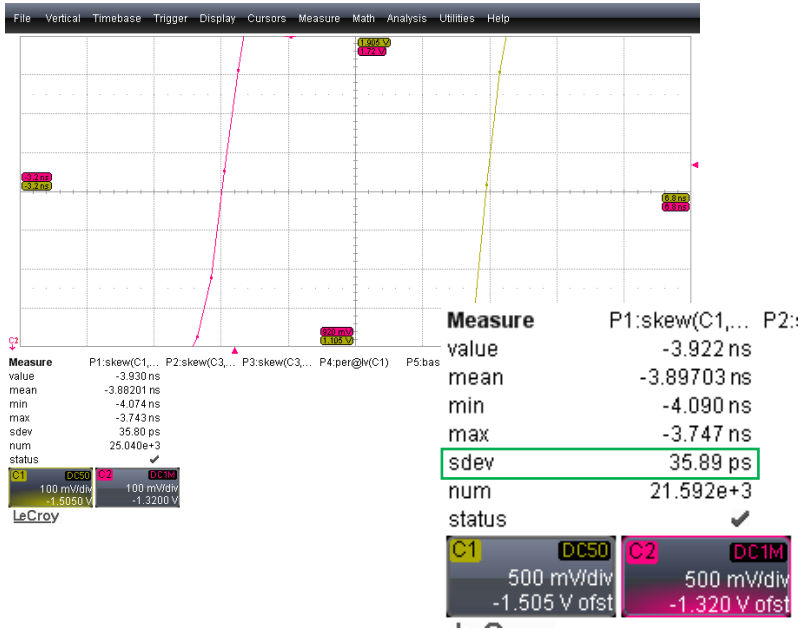
The time resolution of the scintillation cell is 130 ps

HGND readout proto_v2 WR tests at JINR (16 – 17 of April)

HGND readout proto_v2 connected to WR master at JINR

WR TX clocks 62.5 MHz:
orange – WR master; pink – WR slave (hgnd)

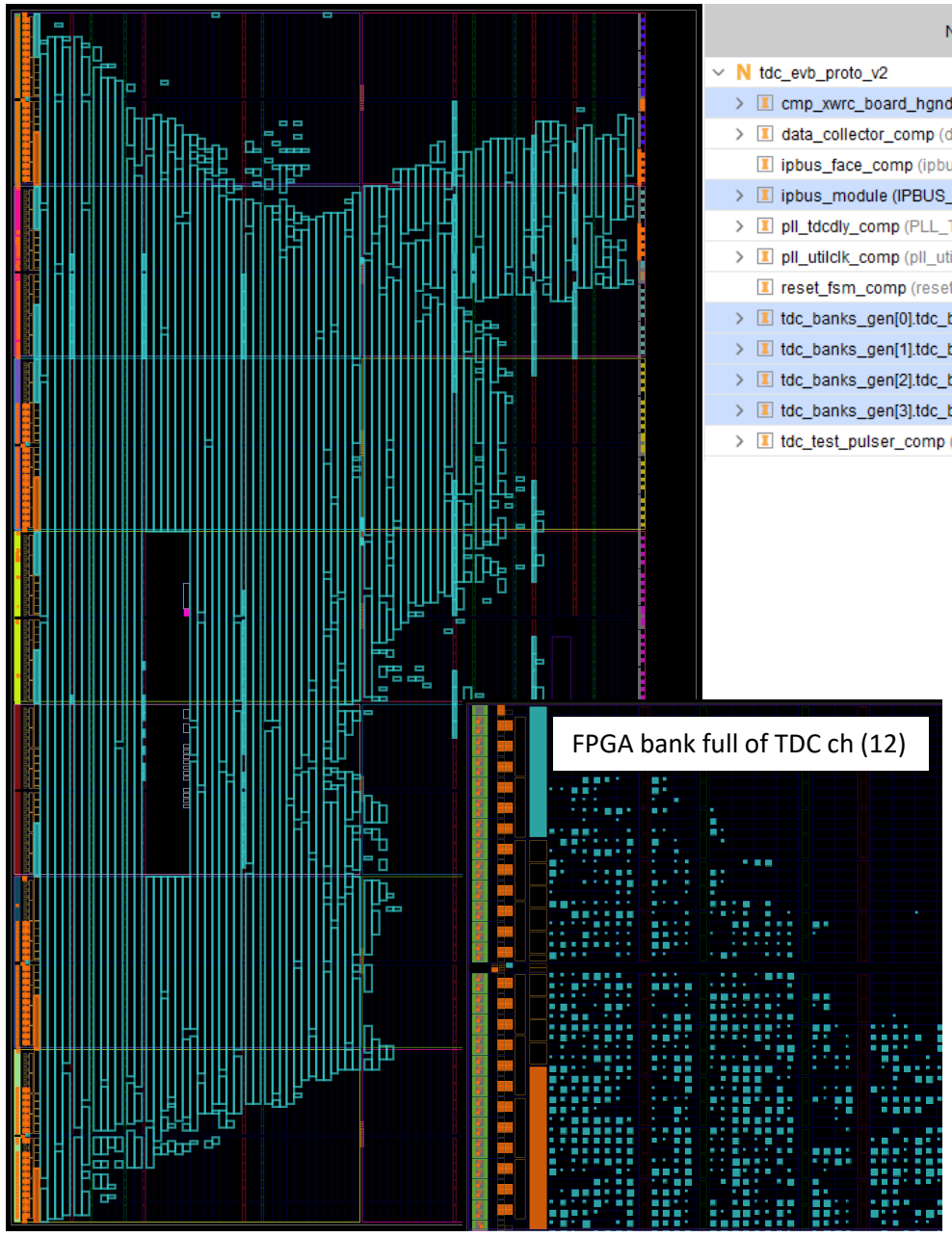
Clock sync jitter ~35ps (measure on FPGA output buffers)



WR tests at JINR conclusions:

- WR synchronization works
- Tx timestamp error (reason found: generic fifo synthesis bug)
- EEPROM not works (fixed – I2C wires swapped)

39 TDC + WR + IPbus FPGA design

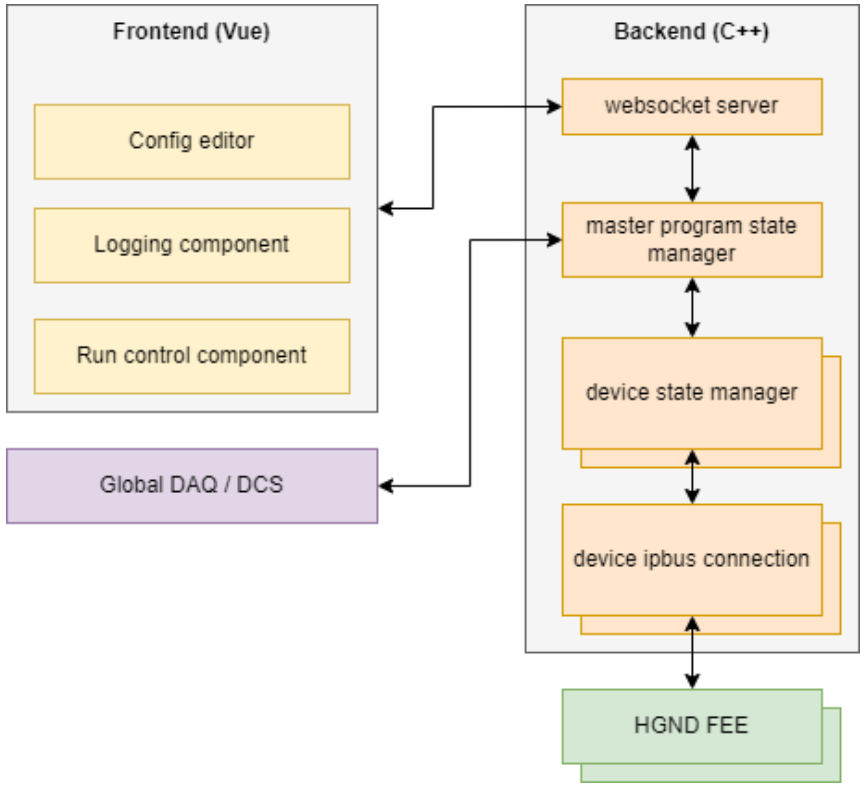


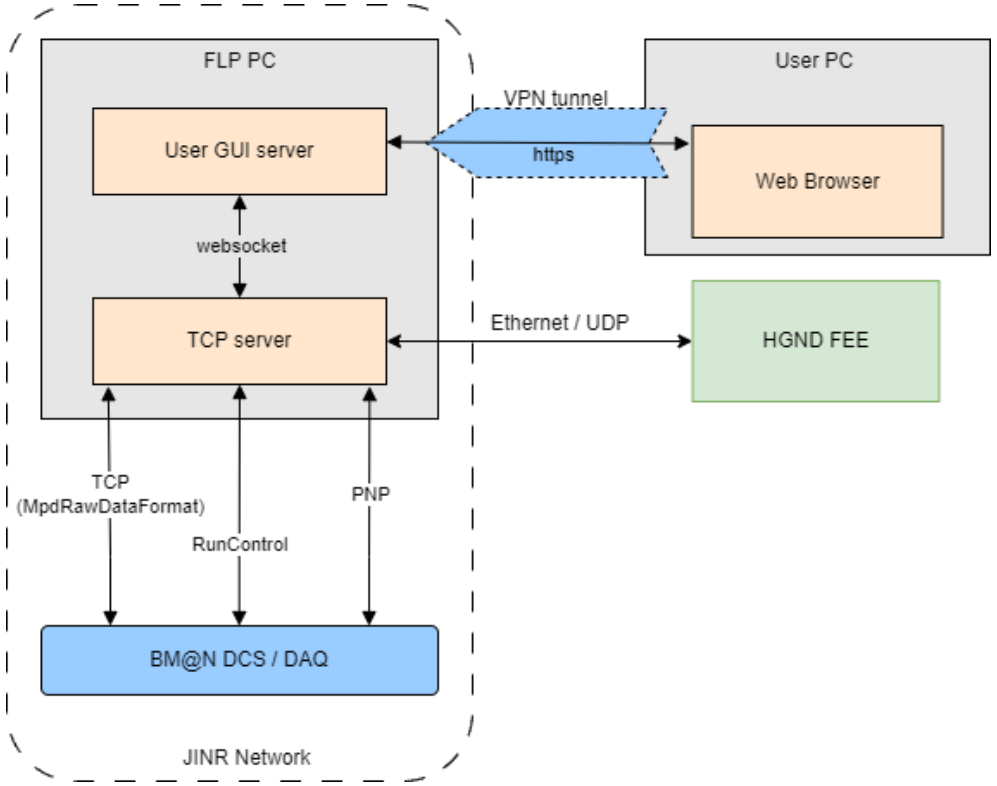
Name	Slice LUTs (203800)	Slice Registers (407600)	Block RAM Tile (445)	Slice (50950)	LUT as Logic (203800)	LUT as Memory (64000)	IBUFDS (480)	GTXE2_CHANNEL (16)	BUFGCTRL (32)	BUFHCE (168)
tdc_evb_proto_v2	51768	71169	106	23849	51676	92	157	2	23	5
cmp_xwrc_board_hgnd (xwrc_board_afi_hgnd)	4776	4838	39.5	2015	4768	8	0	1	6	0
data_collector_comp (data_collector)	678	194	30	343	678	0	0	0	0	0
ipbus_face_comp (ipbus_interface)	3204	10681	0	4037	3204	0	0	0	0	0
ipbus_module (IPBUS_basex_infra)	12015	5023	17	4217	11931	84	0	1	3	5
pll_tdcdly_comp (PLL_TDCDLY)	0	0	0	0	0	0	0	0	2	0
pll_utilclk_comp (pll_utilclk)	0	0	0	0	0	0	0	0	2	0
reset_fsm_comp (reset_fsm)	49	52	0	28	49	0	0	0	0	0
tdc_banks_gen[0].tdc_bank_comp (tdc_bank)	7237	11617	4.5	4161	7237	0	36	0	2	0
tdc_banks_gen[1].tdc_bank_comp (tdc_bank__parameterized1)	7192	11617	4.5	3693	7192	0	36	0	2	0
tdc_banks_gen[2].tdc_bank_comp (tdc_bank__parameterized3)	7067	11617	4.5	3955	7067	0	36	0	2	0
tdc_banks_gen[3].tdc_bank_comp (tdc_bank__parameterized5)	9477	15442	6	5398	9477	0	48	0	2	0
tdc_test_pulser_comp (tdc_test_pulser)	71	70	0	47	71	0	0	0	1	0

- 39 TDC FPGA design was prepared
- Now xc7k325 is used (EvB – 10 banks), xc7k160 will be used on readout board
- xc7k160 has 8 banks: 7 * 12 = 84 TDC channels + 1 bank for utilities

- TDC pairs routing per bank in sequential and mixed pattern will be tested
- TDC + WR + IPbus design prove of the concept will be shown

The topology of the DCS & readout software

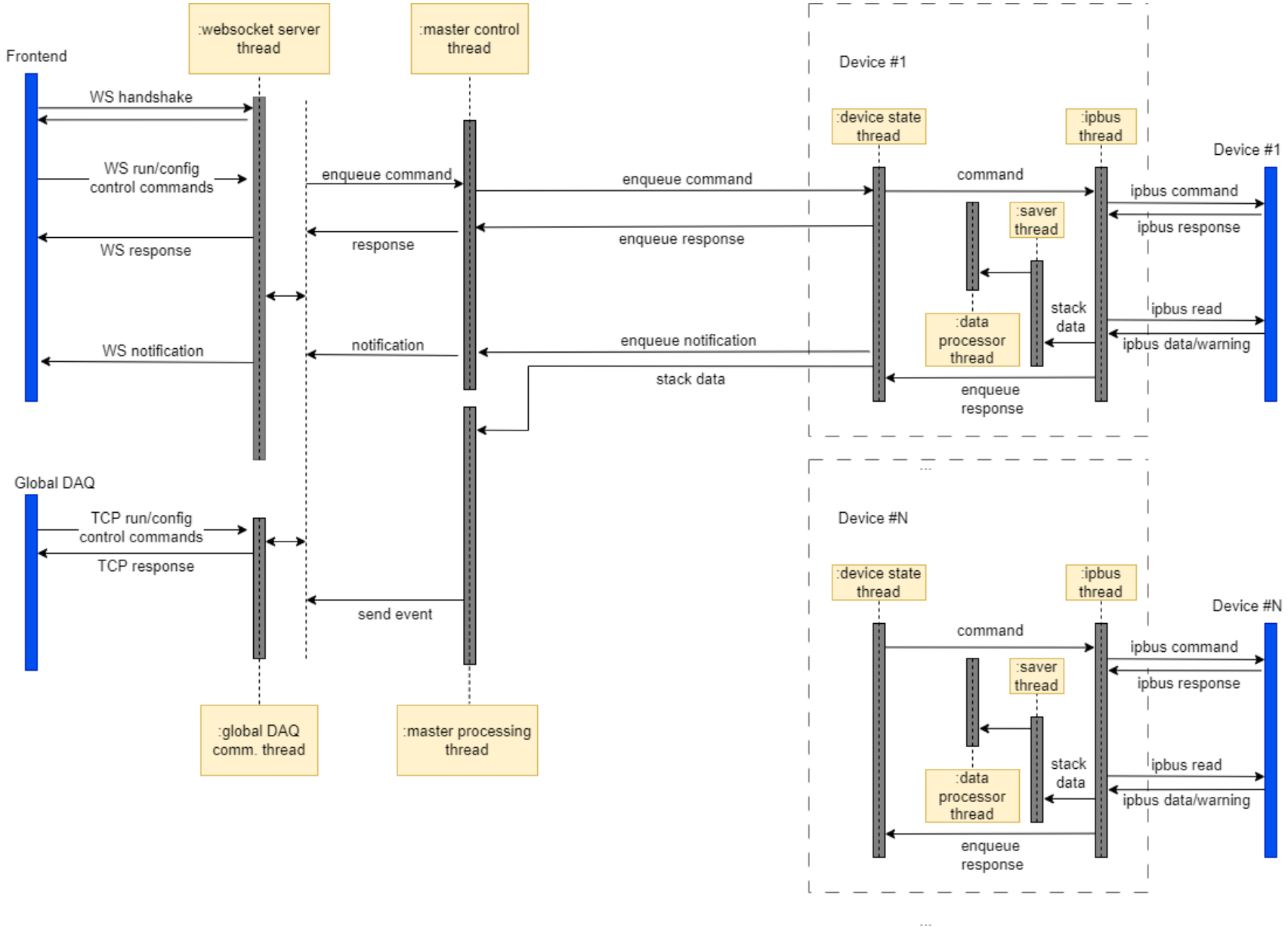




- The detector control system consists of 2 independent modules: frontend and backend
- The backend is a C++ server running on a DCS computer in the same network with detectors.
- The server part provides write/read operations to the detectors via an IPBus connection to the control and data acquisition board.

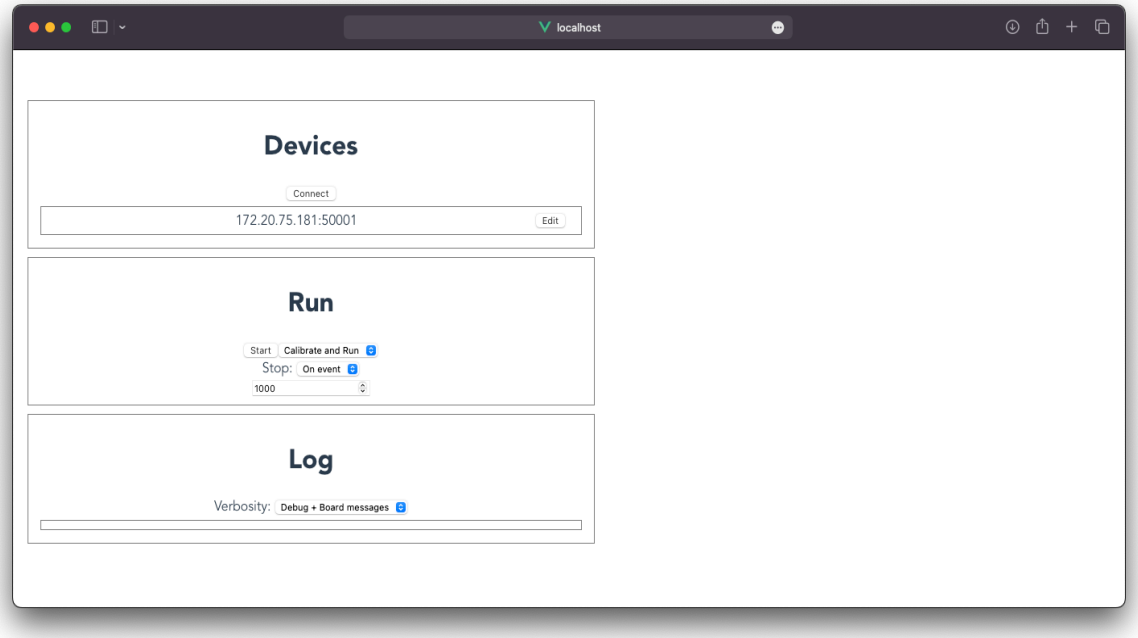
- Frontend is the user interface for this backend. It can be implemented in various ways; the key parameters are reactivity and an ability to establish a connection via websocket with the server part.
- The current version is the Vue web application (open-source JavaScript framework).
- The interface can be run either on the DCS computer or on the operator's computer, provided that the port is available.

The threading scheme of a C++ backend



HGND readout & DCS readout test setup

HGND proto_v2 and EvB connected to gigabit switch for readout tests



HGND DAQ & DCS GUI current view

Synthetic tests for proving the concept of developed software architecture

- ✓ Data & trigger FPGA emulator
- ✓ 400 Mbit/s data readout per board (100 Mbit is required)
- ✓ 800 Mbit/s readout rate with 2 boards was achieved (is the estimated rate for HGND with 8 links)
- ✓ Data sorting by trigger selection with two links readout: data rate 2.6 MHz (10 kHz/channel), trigger rate 10kHz.
- ✓ Data flow (soft emu) & DCS commands test on BM@N FLP
- ➢ Preparing software for TDC tests

Conclusions

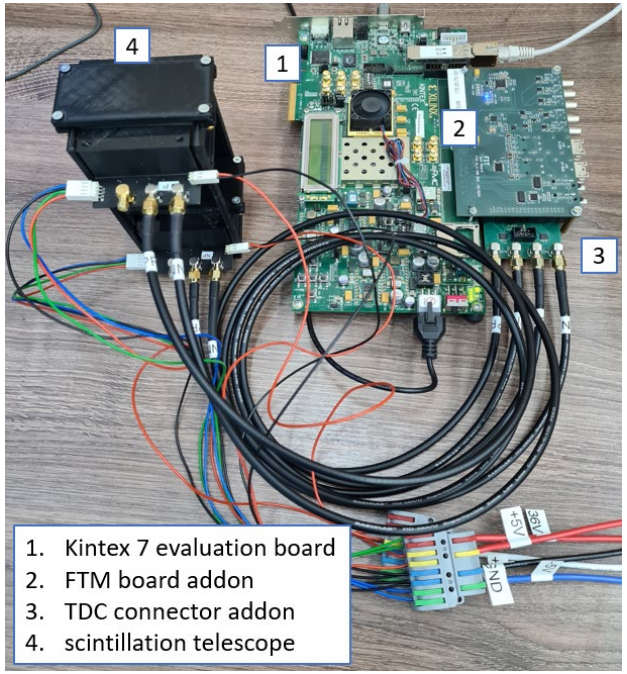
- Status of the readout board development:
  - The White Rabbit setup was tested at JINR: Clock synchronization works but the timestamp synchronization failed, reason found, required new test
  - Performing TDC tests with 33 channels (debugging, calibration, time scan, pulser, cosmic)
  - Working on the design of the full scale readout board: board routing

- The DCS & readout software:
  - Required throughput was achieved
  - Data sorting tests was performed
  - Working on software integration tests (FLP at BM@N)
  - June – July: Using software readout & control for TDC tests

Thank you for your attention!

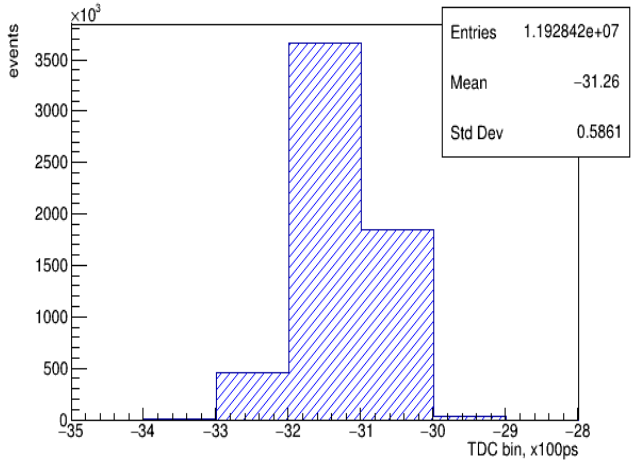
BACKUP

The FPGA TDC test results



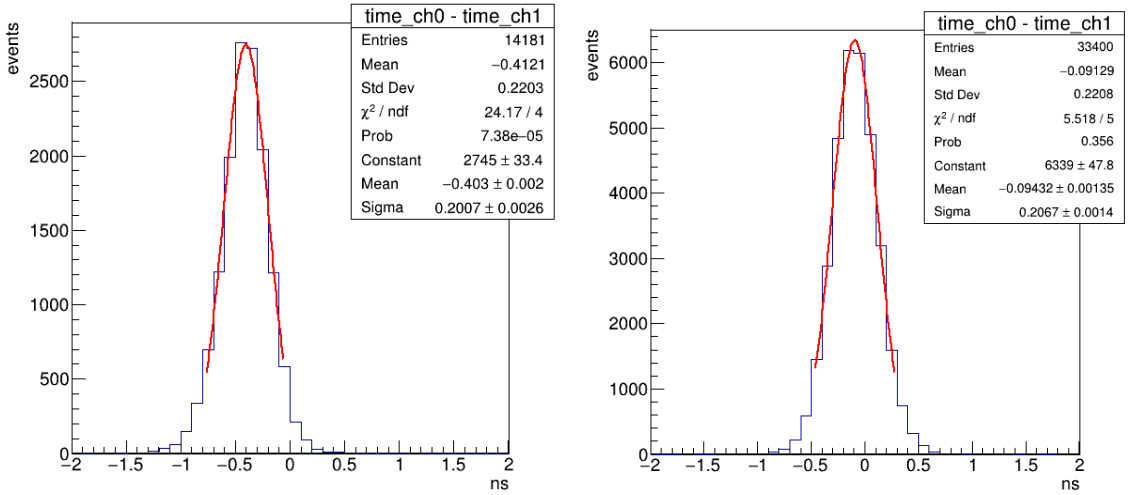
HGND scintillator cells telescope is connected to 2-channels TDC prototype based on KC705 evaluation board

The time difference distribution between two FPGA TDC channels measured with the data generator DG2040 (Cycle-to-Cycle Jitter 5ps).



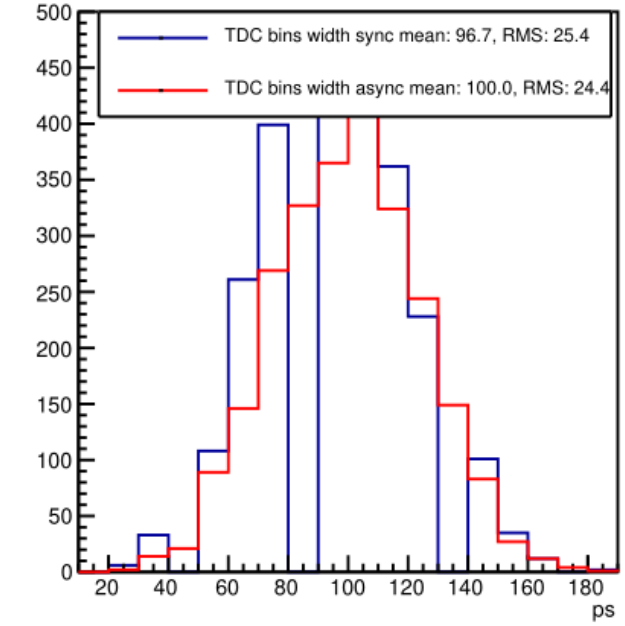
The RMS of single TDC channel is 42ps.

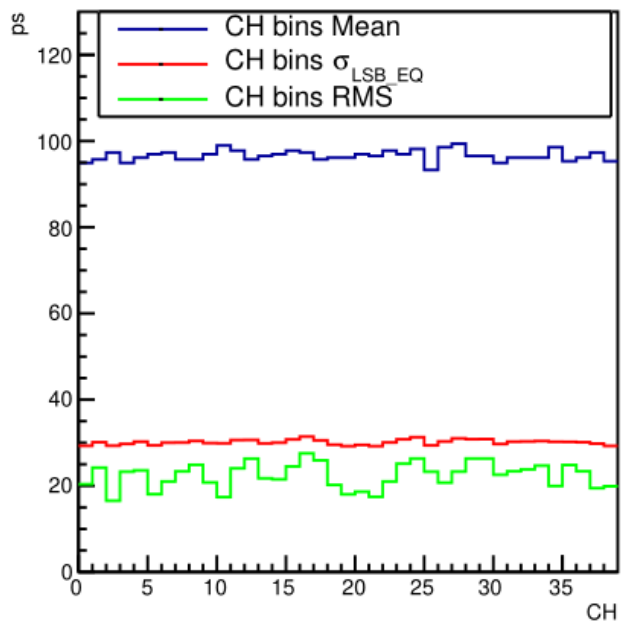
The time resolution measurements with the FPGA TDC prototype board were performed with the 280 MeV electron beam on the “Pakhra” synchrotron in LPI (Moscow, Russia).

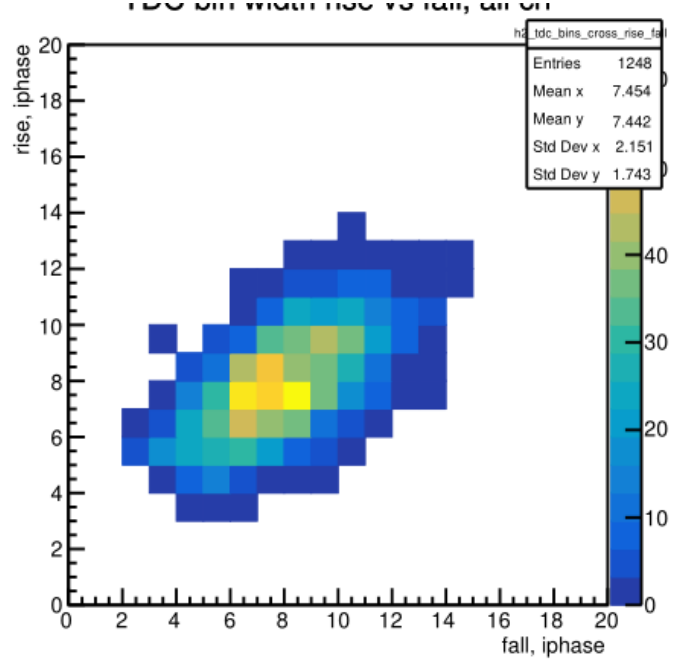


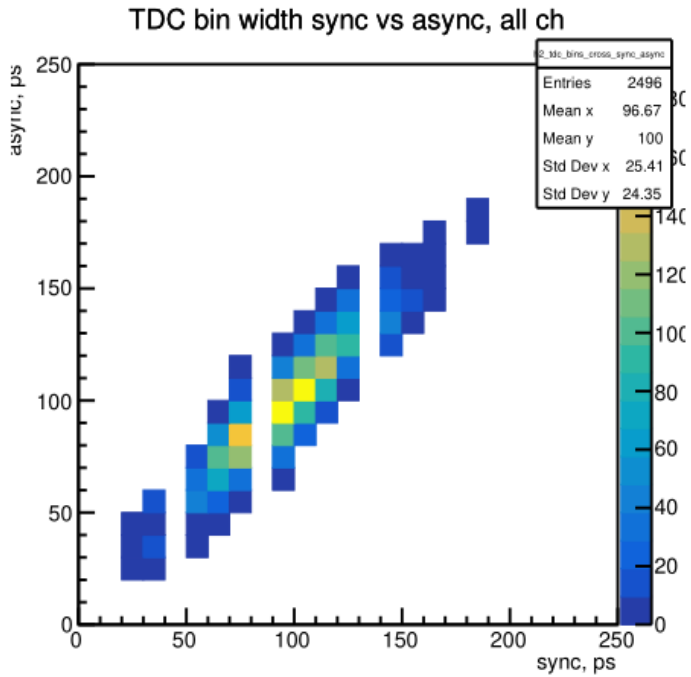
The time difference distributions of two cells of the telescope measured with the CAEN digitizer (left) and the FPGA TDC prototype board (right). Time resolution is ***146 ps*** per single HGND channel.

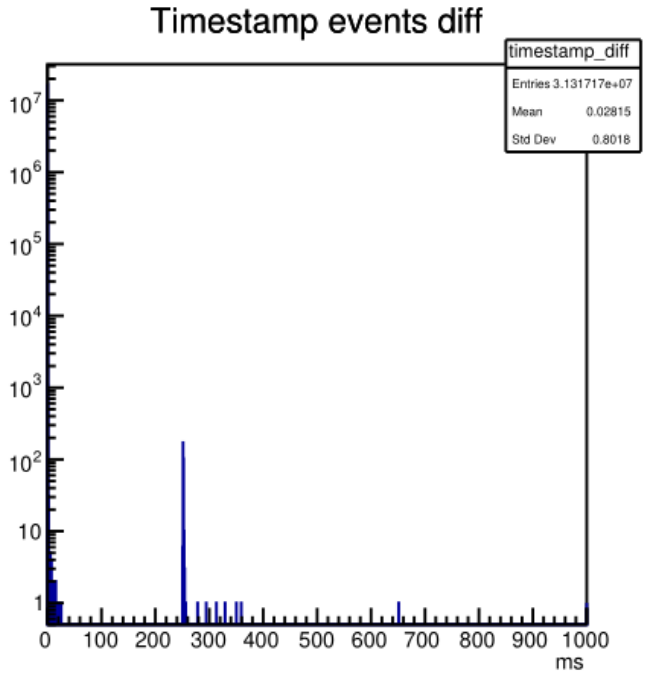
[2] D. Finogeev, F. Guber, A. Izvestnyy, N. Karpushkin, A. Makhnev et al., ***Development of 100 ps TDC based on Kintex 7 FPGA for the High Granular Neutron Time-of-Flight detector for the BM@N experiment,*** *DOI: 10.1016/j.nima.2023.168952*

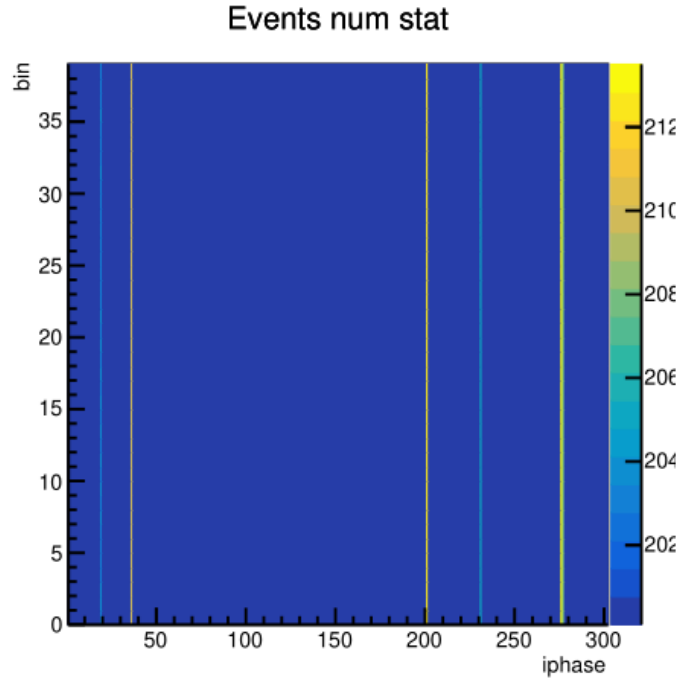

TDC bins width sync mean: 96.7, RMS: 25.4
TDC bins width async mean: 100.0, RMS: 24.4
ps

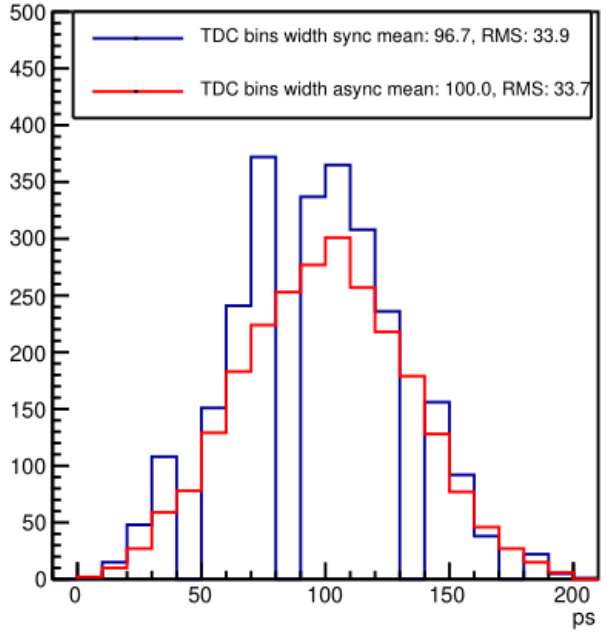

CH bins Mean
CH bins σ LSB_EQ
CH bins RMS
ps
CH

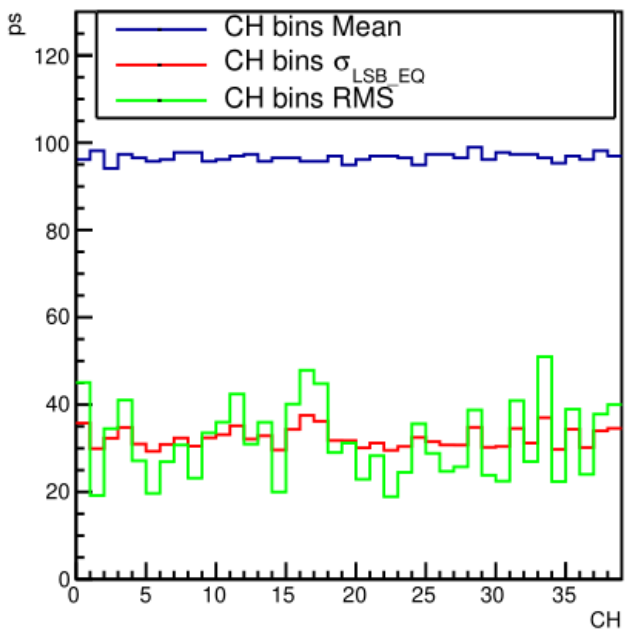

tdc_bins_cross_rise_fall
Entries 1248
Mean x 7.454
Mean y 7.442
Std Dev x 2.151
Std Dev y 1.743
rise, iphase
fall, iphase

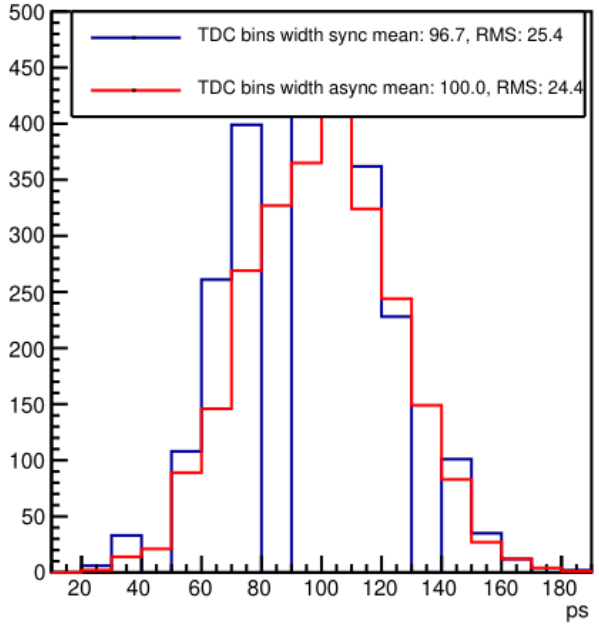

TDC bin width sync vs async, all ch
tdc_bins_cross_sync_async
Entries 2496
Mean x 96.67
Mean y 100
Std Dev x 25.41
Std Dev y 24.35
async, ps
sync, ps

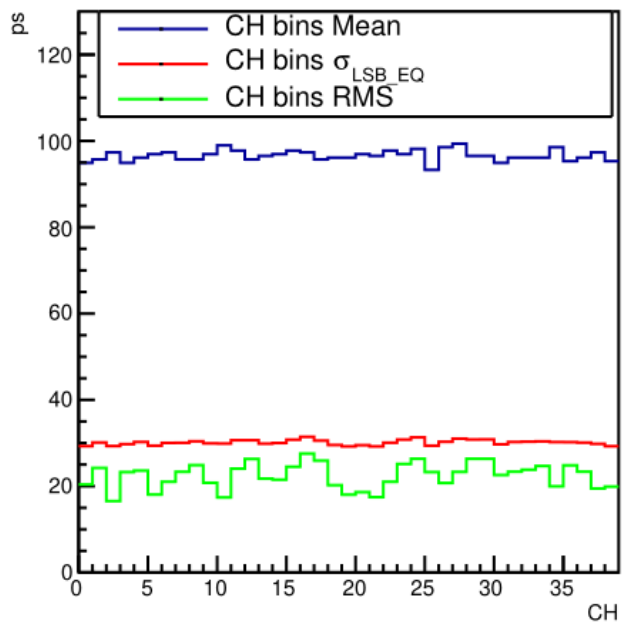

Timestamp events diff
timestamp_diff
Entries 3.131717e+07
Mean 0.02815
Std Dev 0.8018
ms

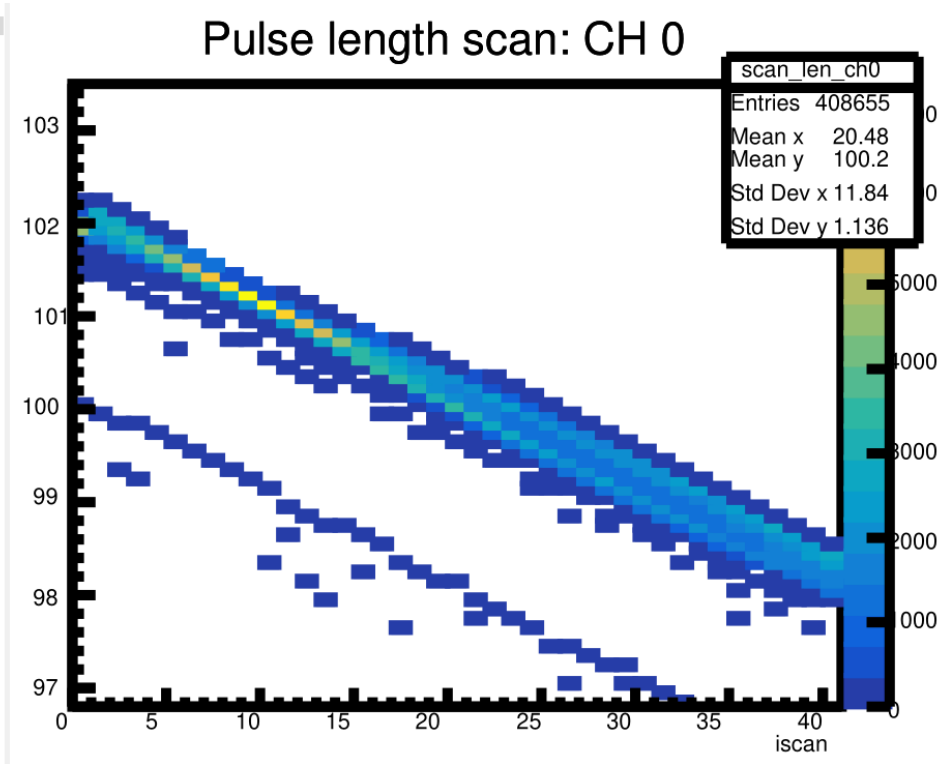

Events num stat
bin
iphase

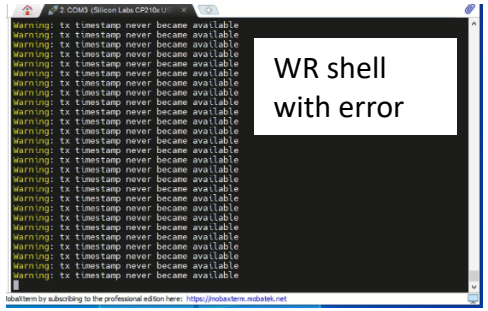

TDC bins width sync mean: 96.7, RMS: 33.9
TDC bins width async mean: 100.0, RMS: 33.7
ps

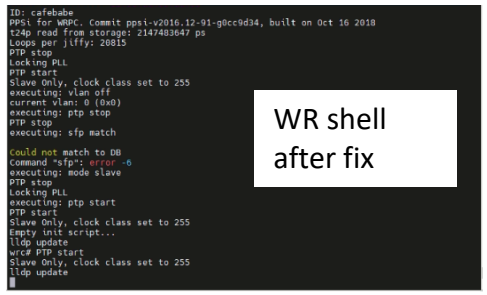

CH bins Mean
CH bins σLSB_EQ
CH bins RMS
ps
CH


TDC bins width sync mean: 96.7, RMS: 25.4
TDC bins width async mean: 100.0, RMS: 24.4
ps


CH bins Mean
CH bins σLSB_EQ
CH bins RMS
ps
CH


Pulse length scan: CH 0
scan_len_ch0
Entries 408655
Mean x 20.48
Mean y 100.2
Std Dev x 11.84
Std Dev y 1.136
iscan





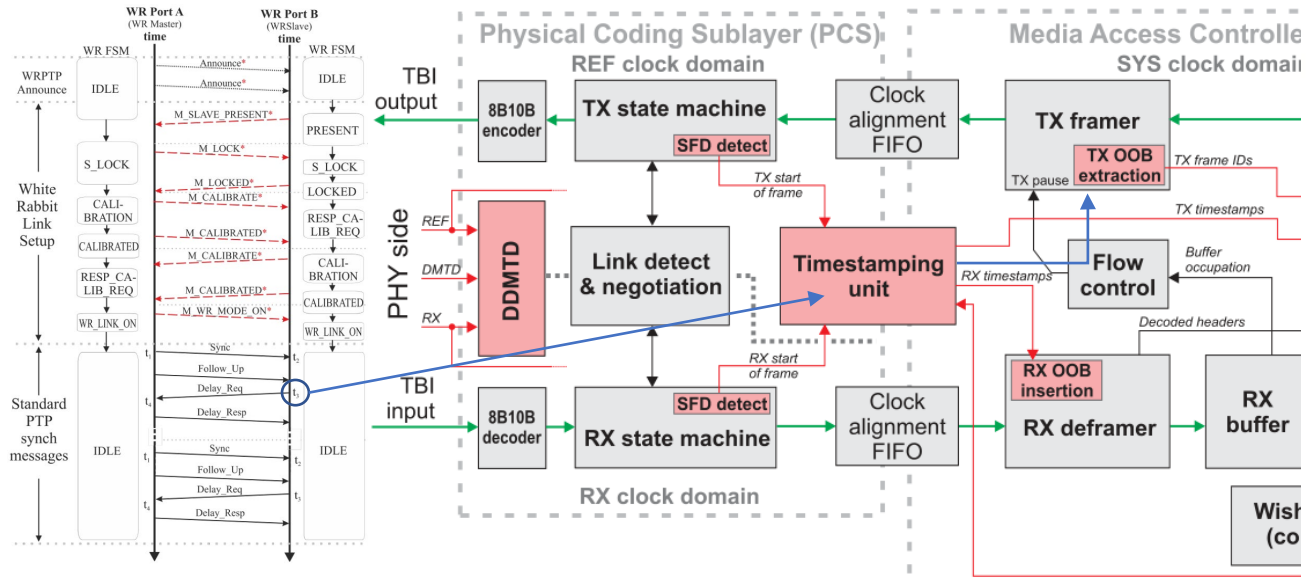


Figure 21: White Rabbit PTP extension

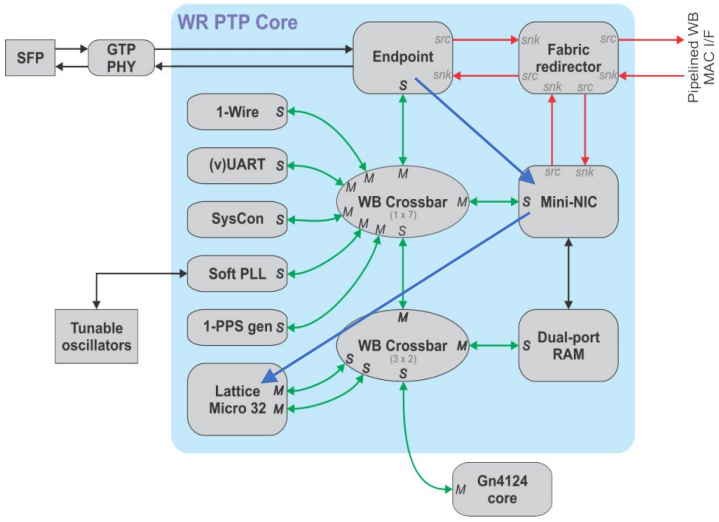
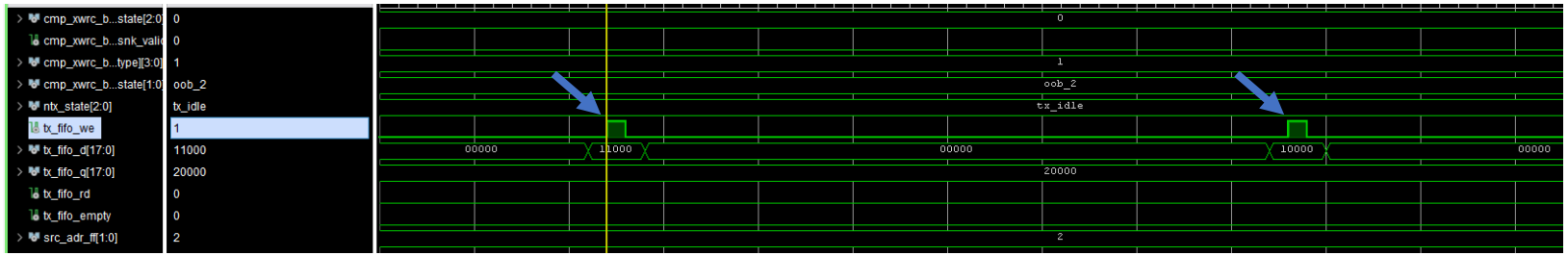
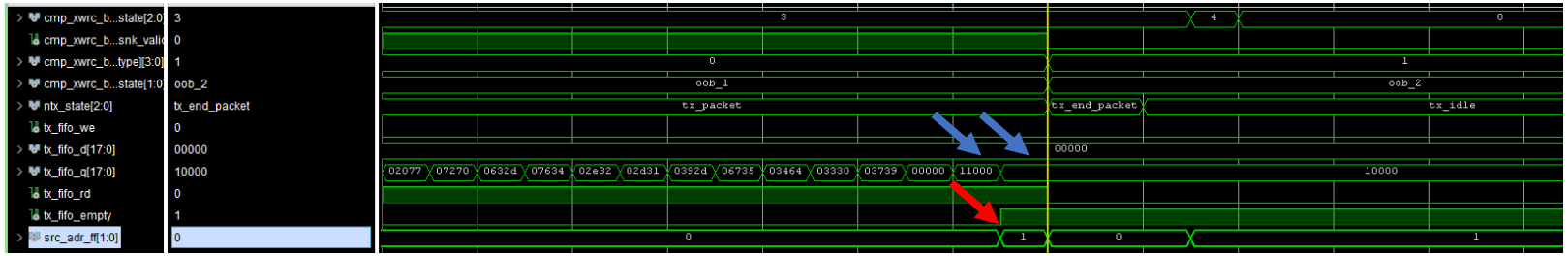


Figure 23: White Rabbit PTP Core block diagram

- TX timestamps is sent with two OOB packets
- LM32 request 2 OOB packets for reading TX timestamp: packets addr 11000 and 10000:

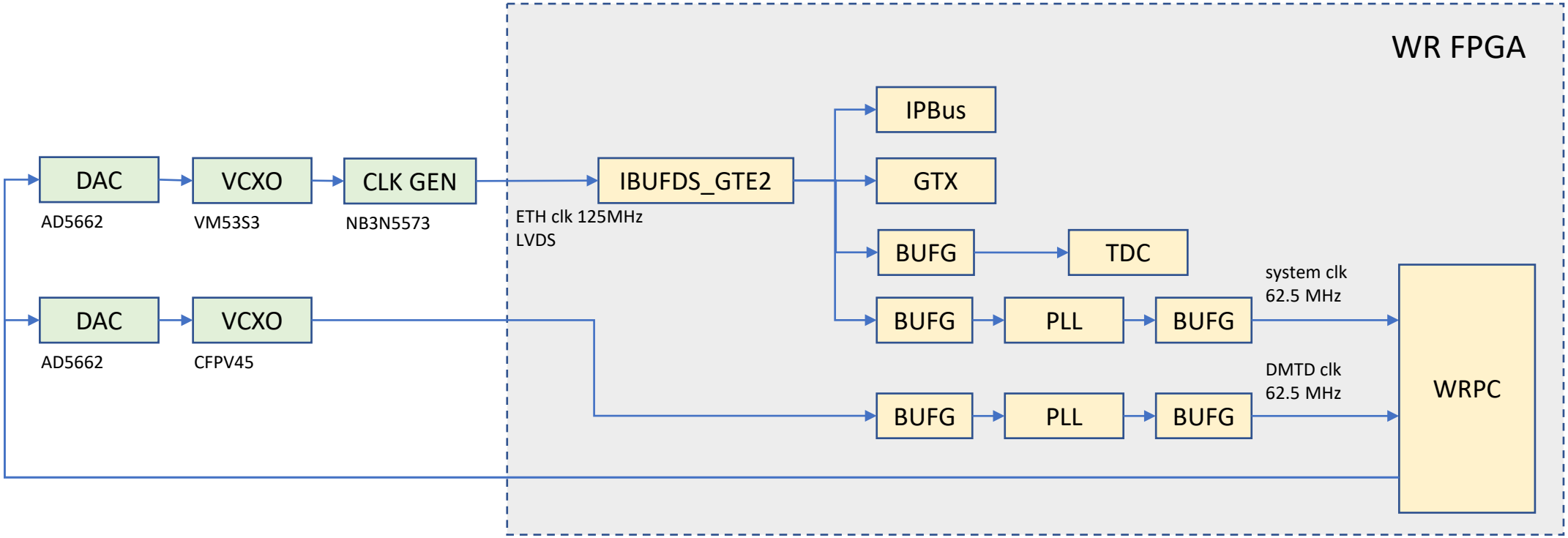


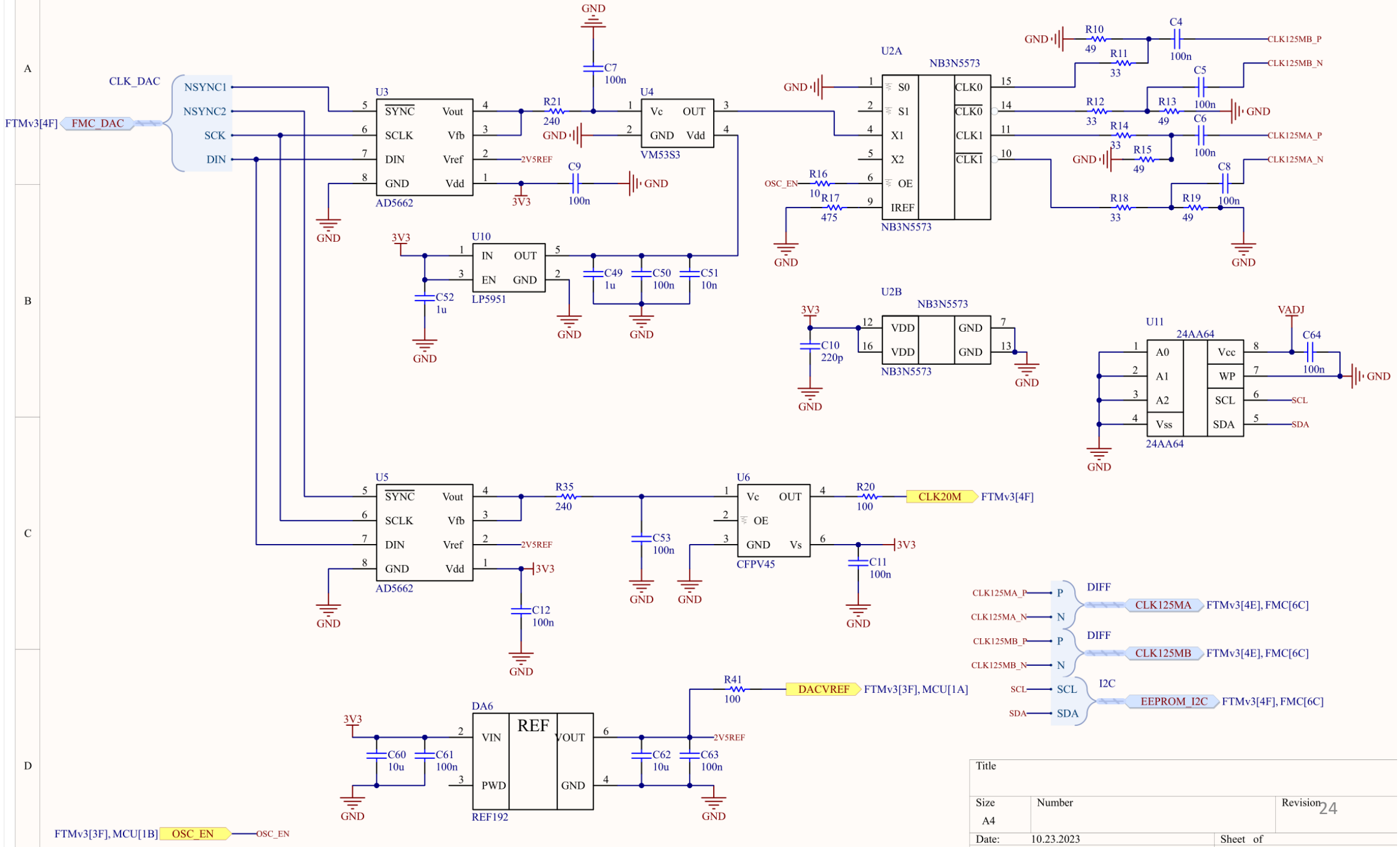
- While reading two OOB packets in 'TX framer' fifo rise empty one cycle early
- latching fifo_empty signal for one cycle mitigates the error



- Module wr-cores\modules\wr_endpoint\ep_tx_header_processor.vhd (line 341) send tx timestamp while processing two OOB packets
- OOB packets are requested by LM32. Request is processed by wr-cores\modules\wr_mini_nic\wr_mini_nic.vhd with pushing data to 'TX_FIFO: generic_sync_fifo'.
- This fifo is read in wr_mini_nic while sending packets in ep_tx_header_processor.
- While reading the fifo the empty flag is raised one cycle earlier
- This move FSM to tx_end_packet state and second OOB is not sent
- Tx timestamp is missed because of single OOB packet
- By delaying (latching) fifo_empty by single cycle, mitigates the error "tx timestamp never became available".

The White Rabbit implementation





Title		
Size A4	Number	Revision
Date: 10.23.2023	Sheet of	