Status of the HGND DAQ tests

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

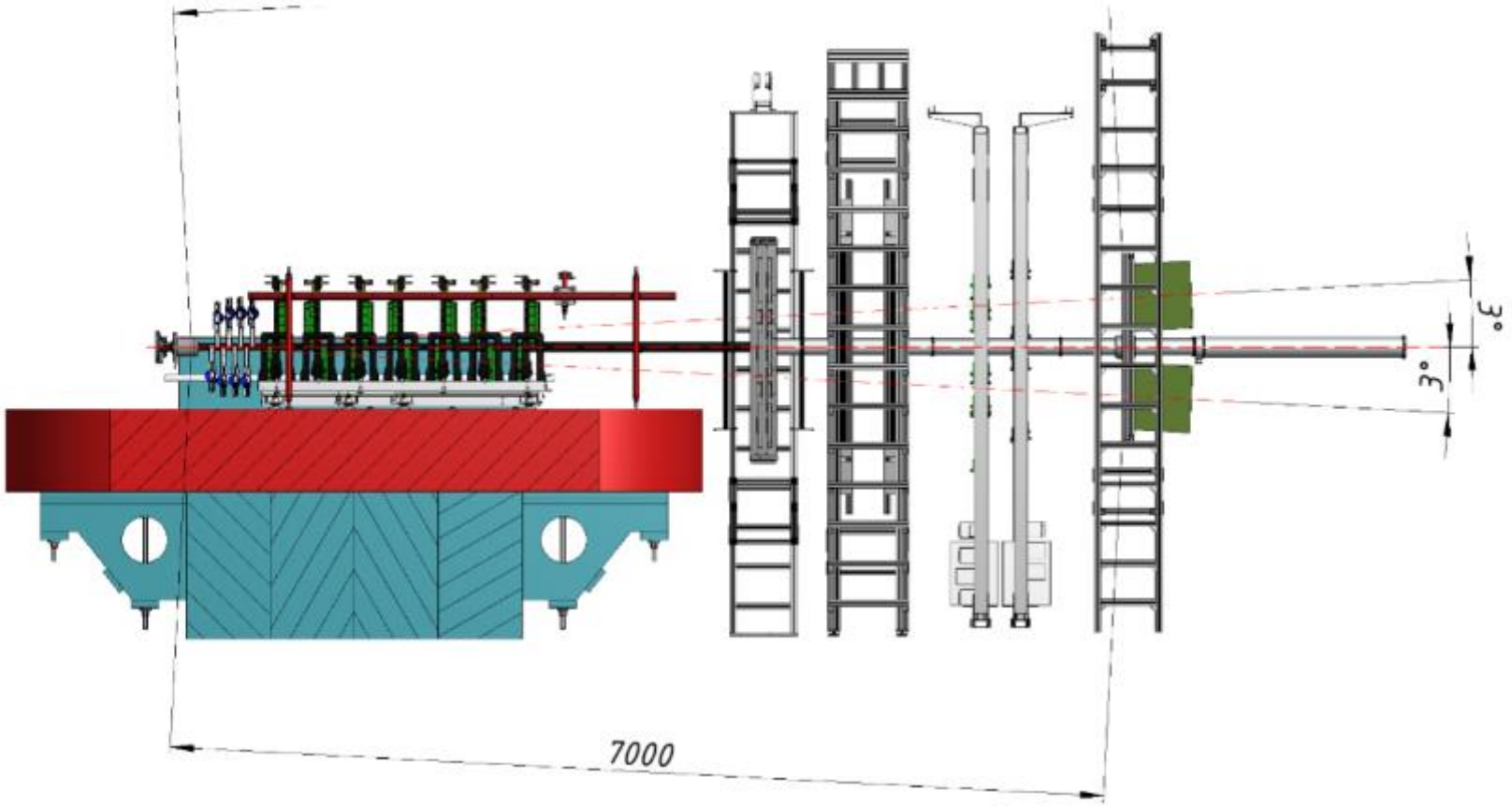
14th Collaboration Meeting of the BM@N Experiment at NICA
2025 May

Outline

- Updates on the software design
- Status of the HW development
- Updates on the FPGA design
- Status of the HGND prototype tests at INR
- HGND prototype setup at BM@N and integration tests

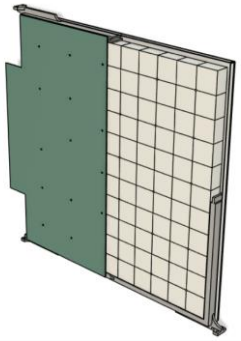
Detector arrangement

- Detector for high-energy neutron flow measurement
- ToF method with T0 as the “start” signal source
- 7m measurement distance
- Detector is split into 2 “blocks” for improved acceptance

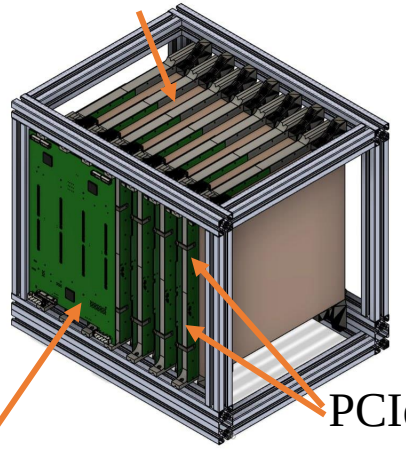


FEE & readout architecture

- 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

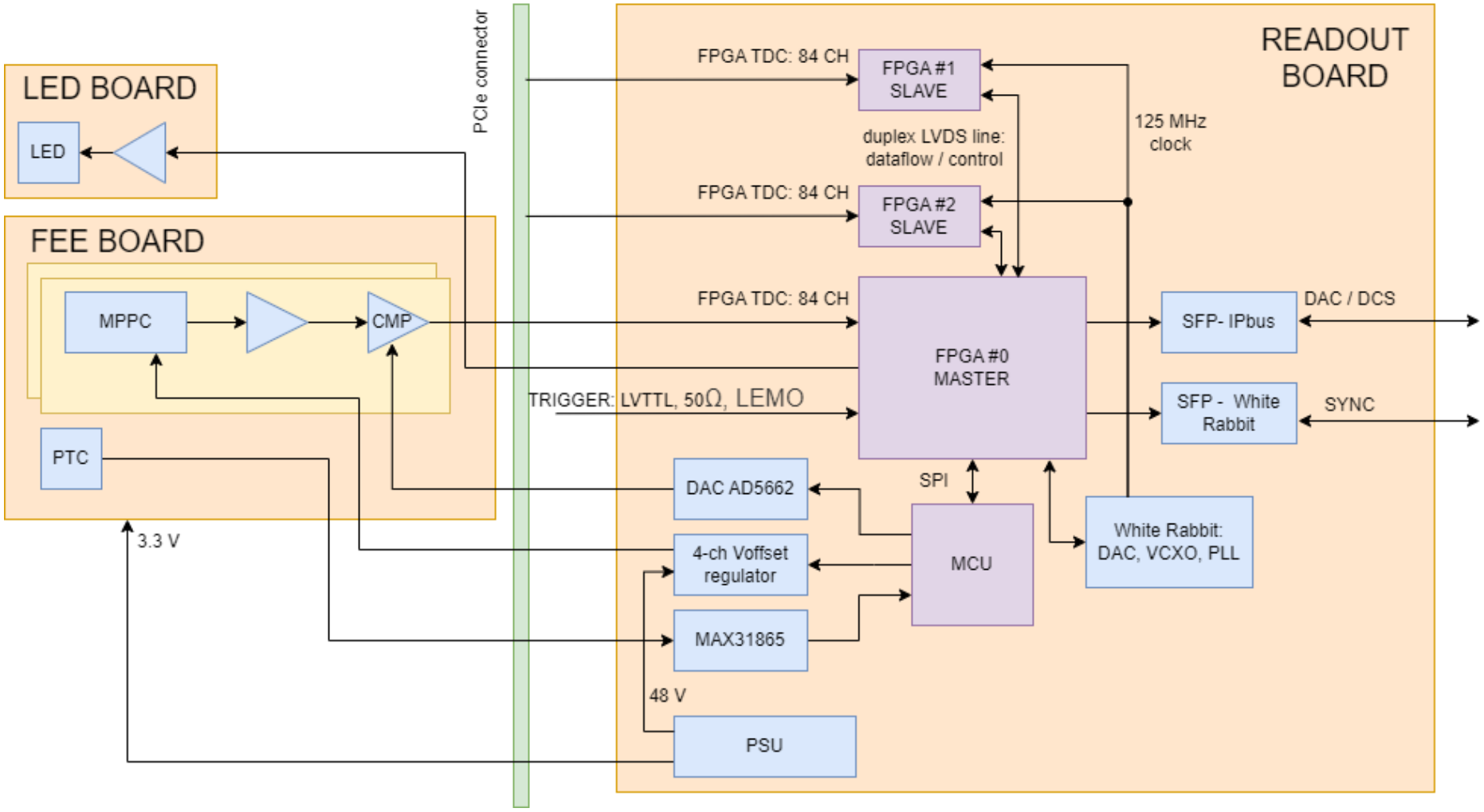


FEE board



PCIe

Readout board



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

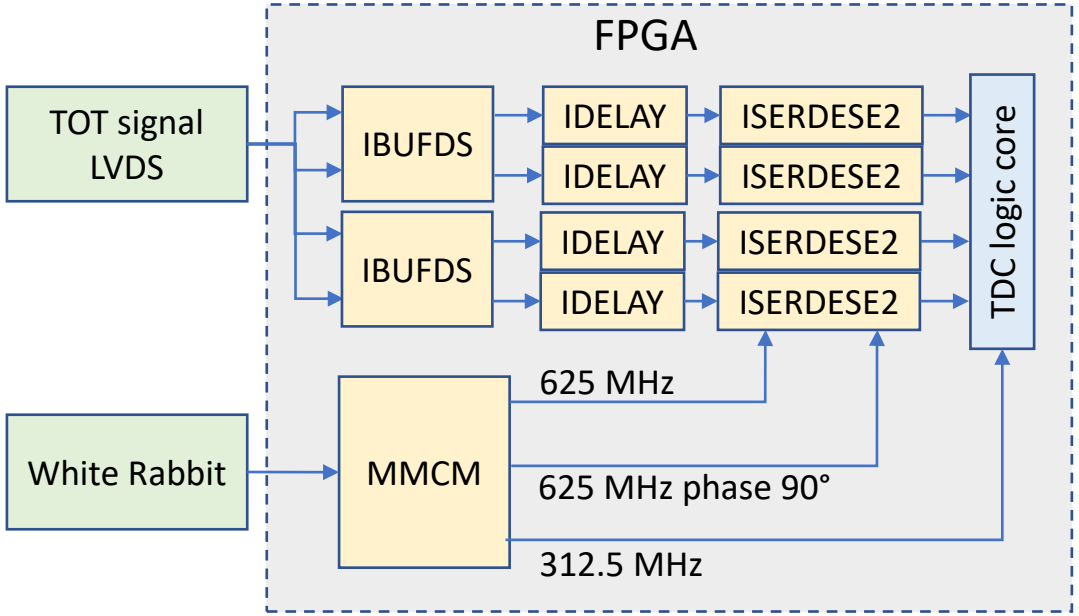
- Ethernet UDP protocol (***IPbus*** [1]) is used for data forwarding and board control
- Local network connect readout boards (8 ethernet links) with FLP

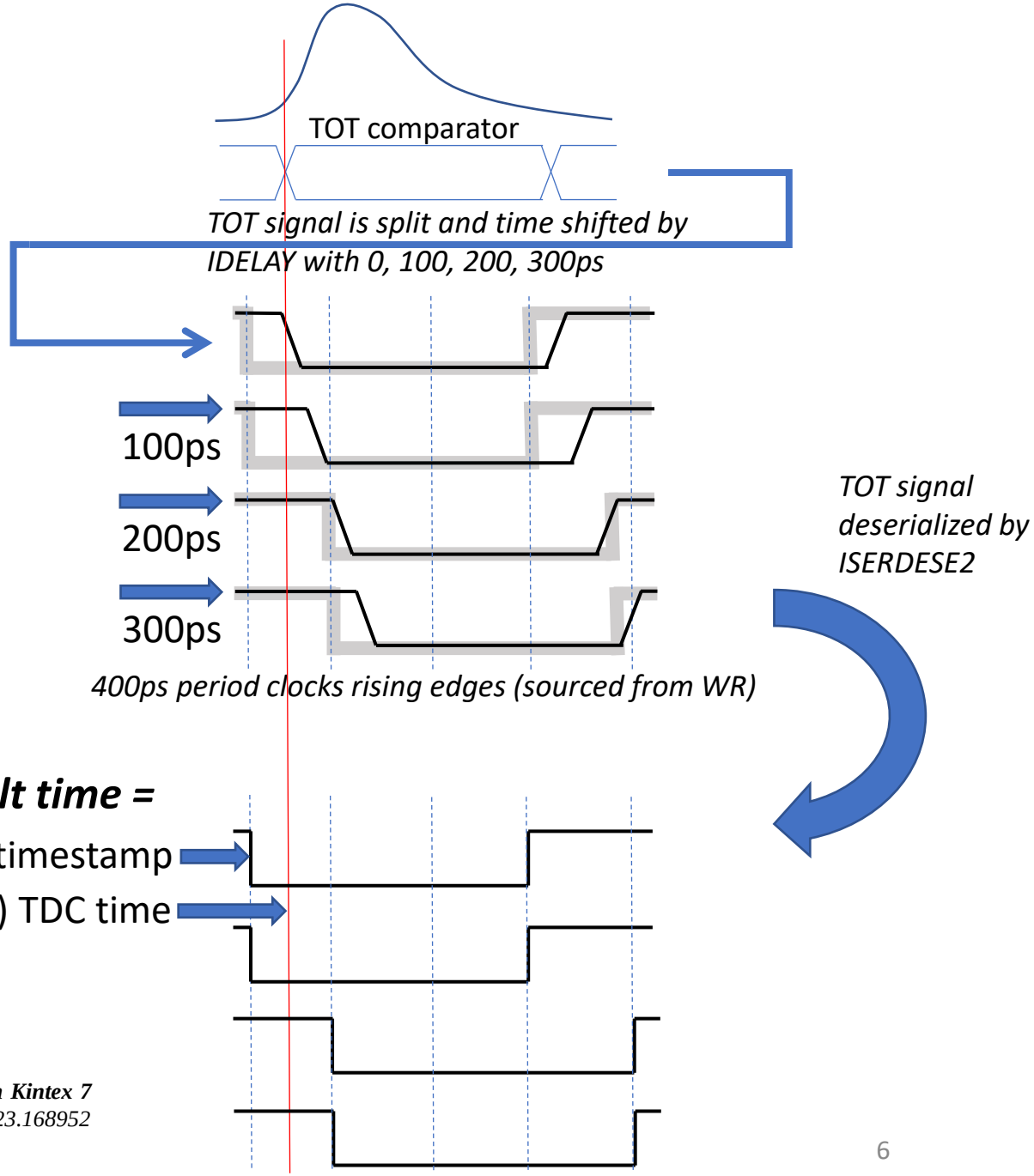
- The maximum HGND channel load is 3 kHz. The event size is 7x16 bits. The upper limit per link is not exceed ***100 Mbit/s***. ***The continuous readout*** is implemented without busy signal.

- 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.*

The 100ps FPGA TDC principle of operation



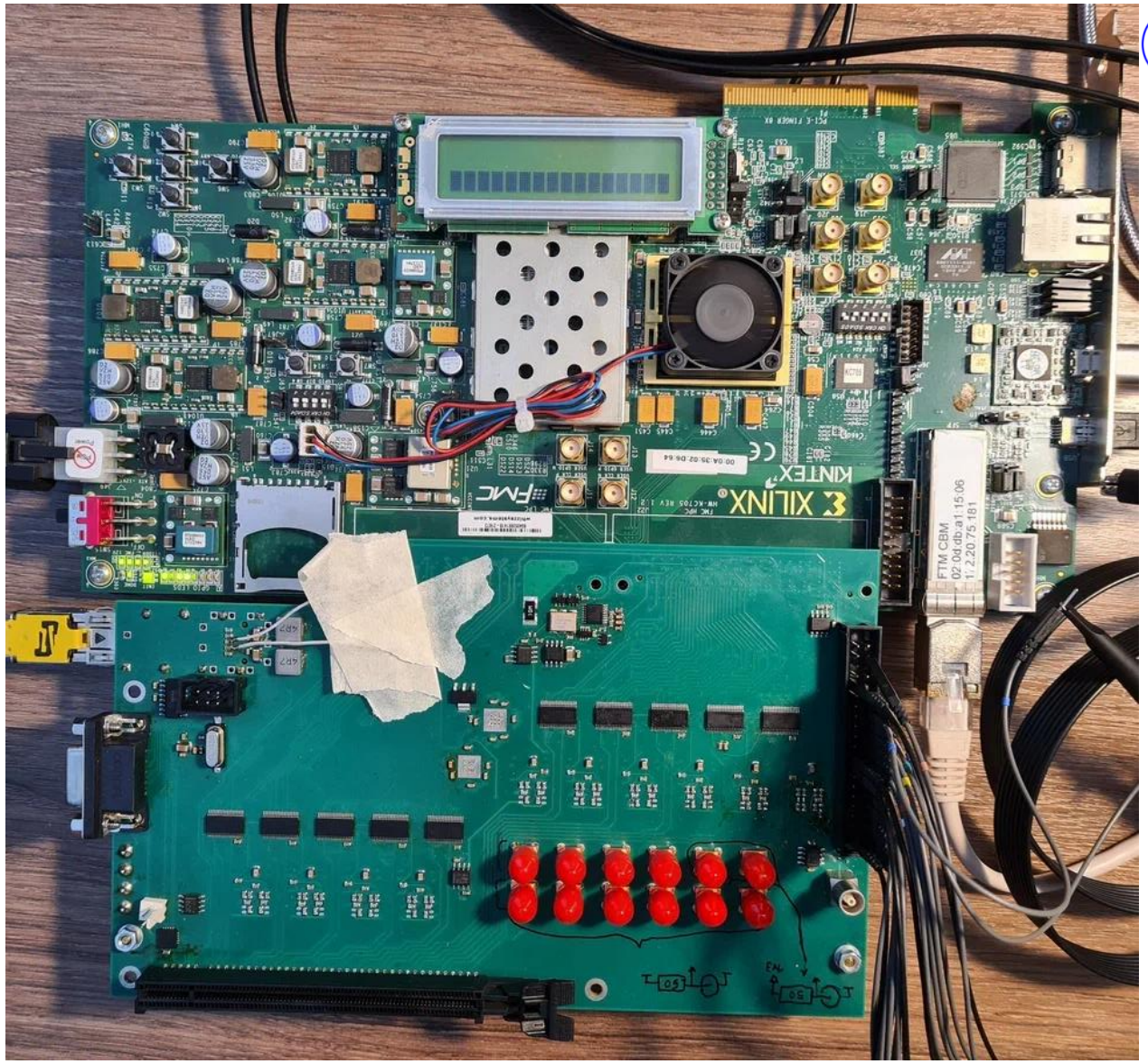


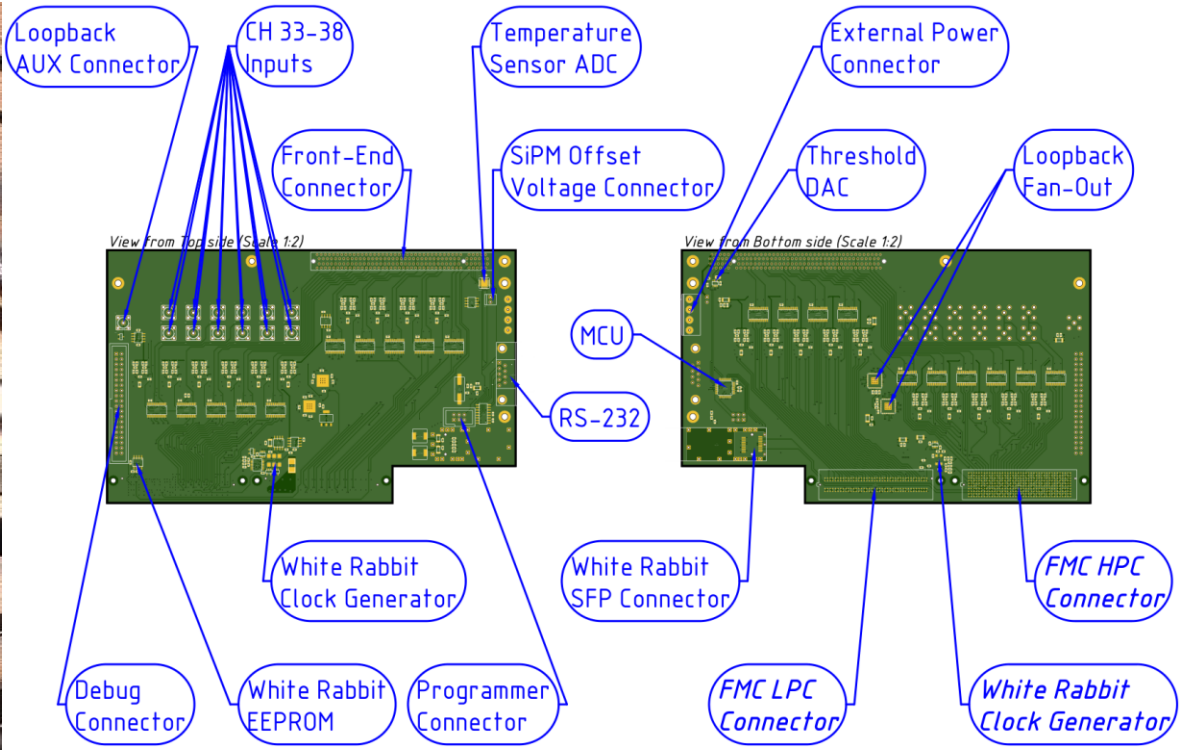
The TDC is **based on the Kintex-7** input serial-to-parallel converter with oversampling capability and programmable delay. The design is **based on Xilinx recommendations**, and uses **only documented features** of the FPGA within its specifications.

[3] 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*

HGND readout v2 prototype (39 channels)

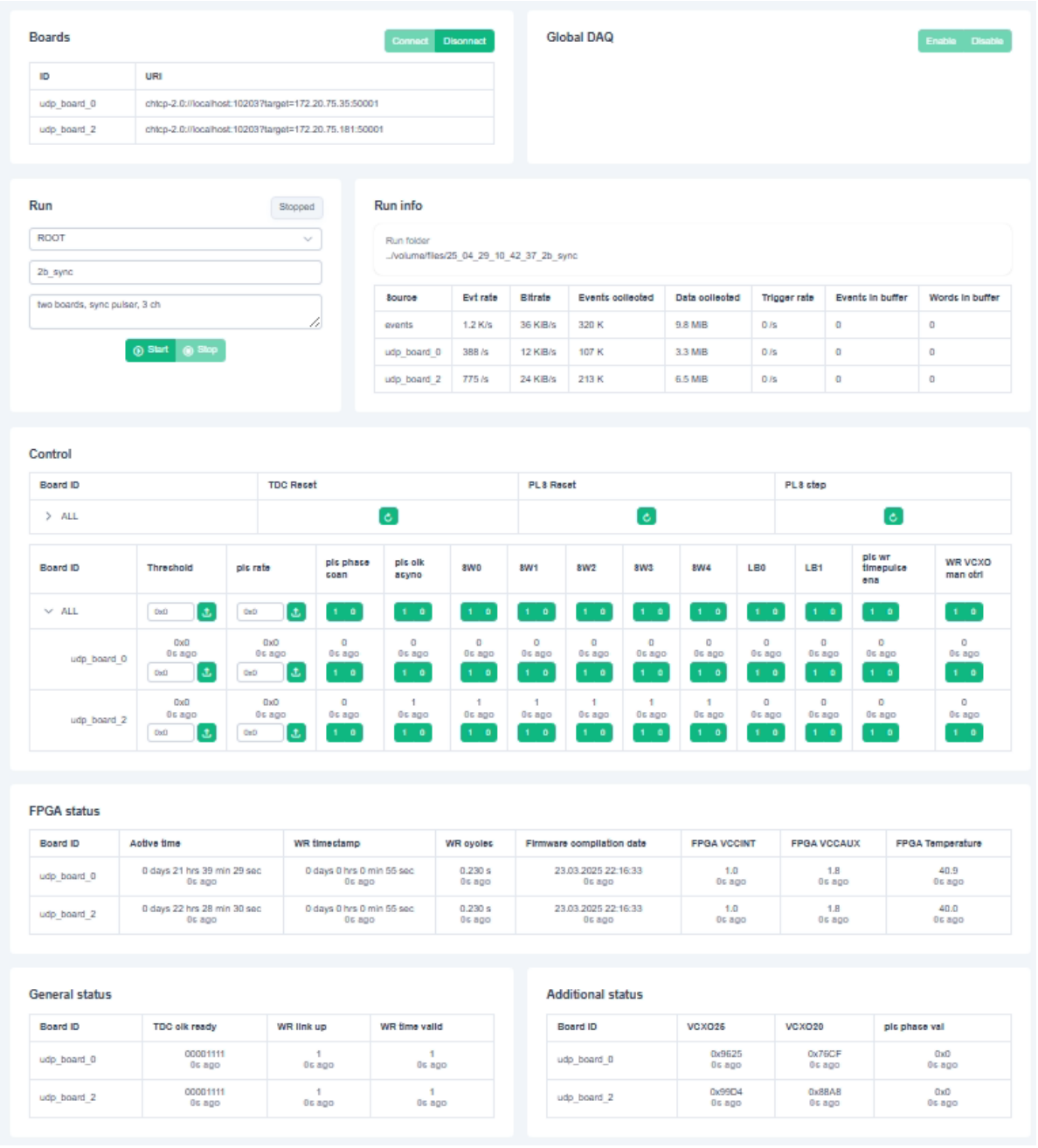
based on the Kintex 7 evaluation board (KC705)

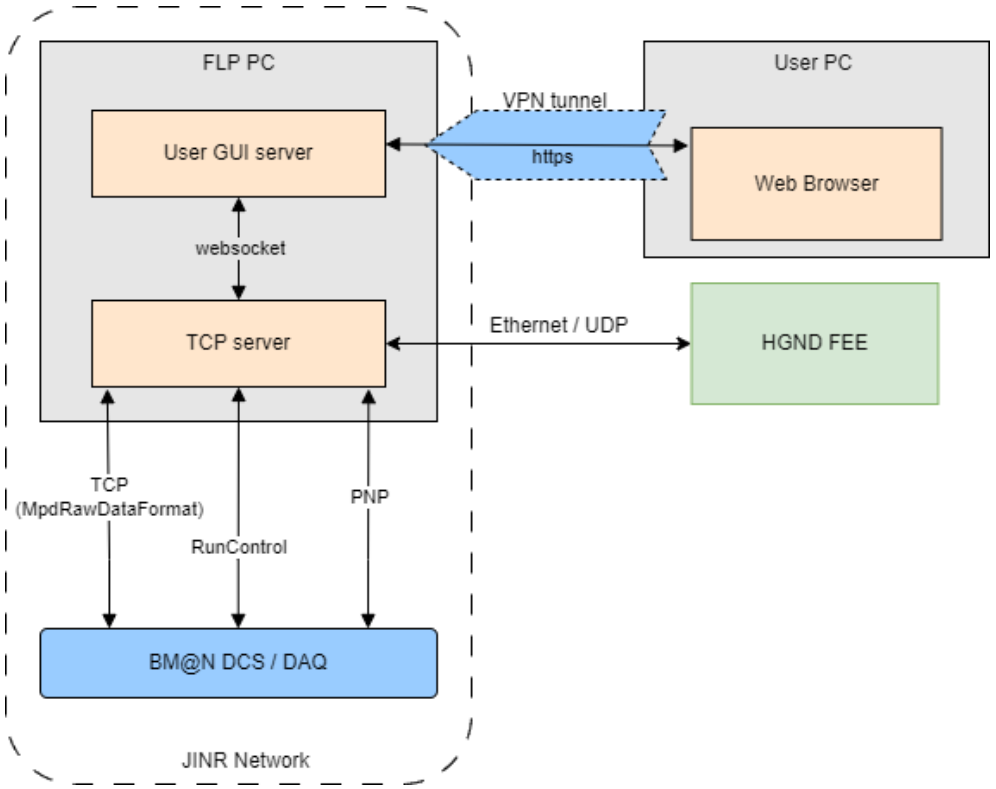




- Addon for Kintex 7 evaluation board
- 33 PCIe + 6 SMA TDC channels
- Ethernet readout
- White Rabbit synchronization
- FPGA loopback for TDC calibration
- Readout board functionality:
  - *PCIe connector for* scintillation matrix*, Temperature sensor, SiPM offset voltage control DAC threshold*

Server development status and preparation for integration into BM@N



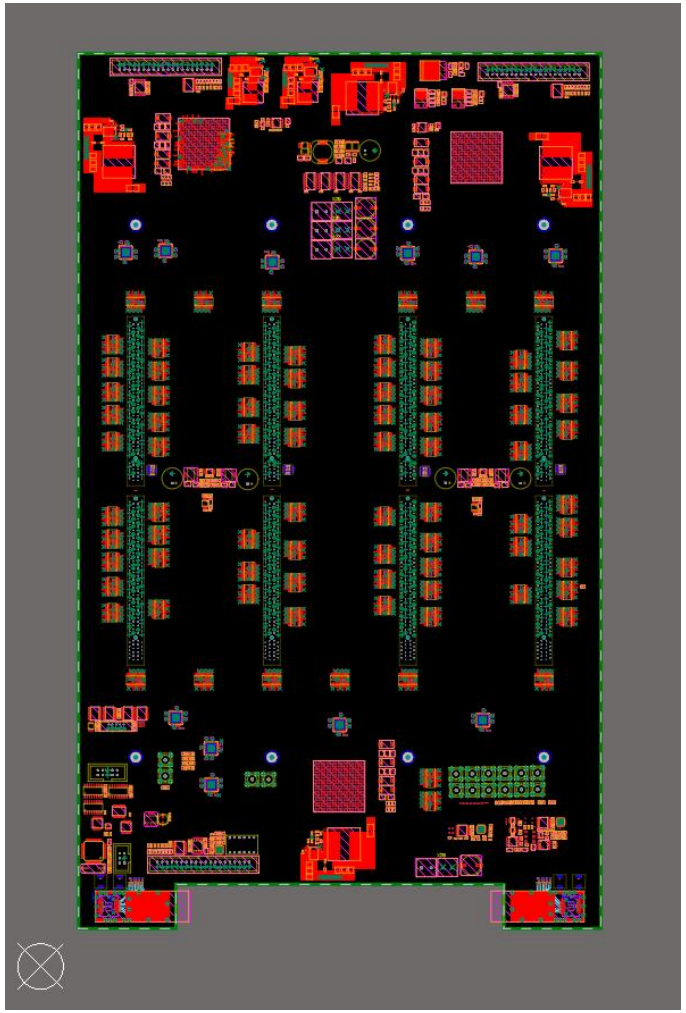


- Web interface: status & configuration
- Global DAQ + DCS
- Standalone data acquisition
- BM@N integration to be tested
- Crucial facilities was tested
- Working on minor tasks

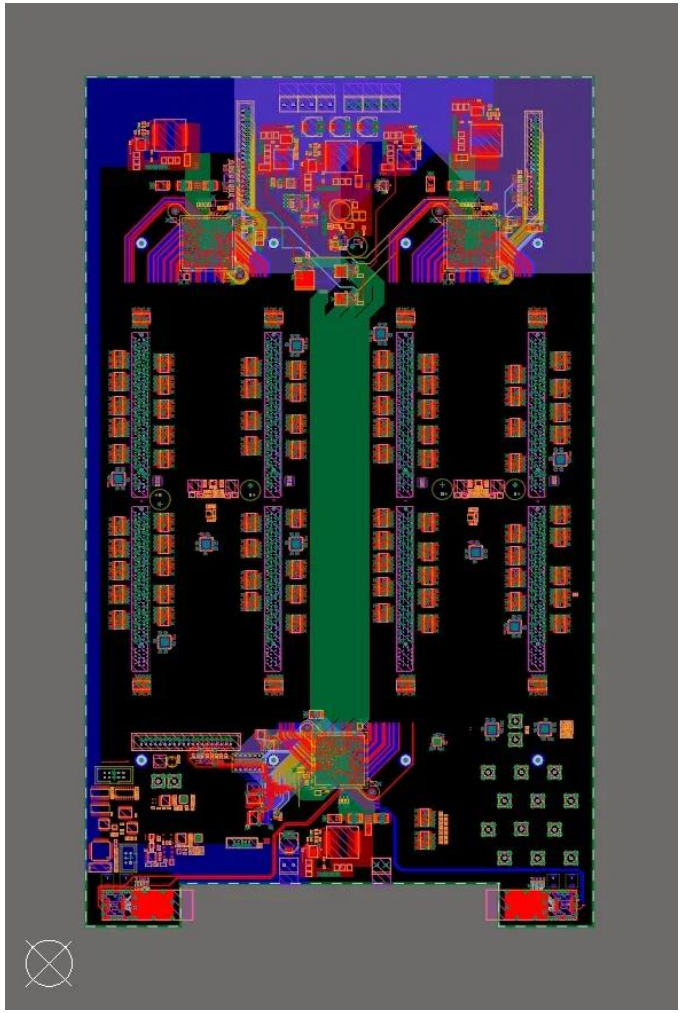
Status of the HW development

Readout board routing status ***Feb 25***

- Elements placement done.



Readout board routing status ***May 25***



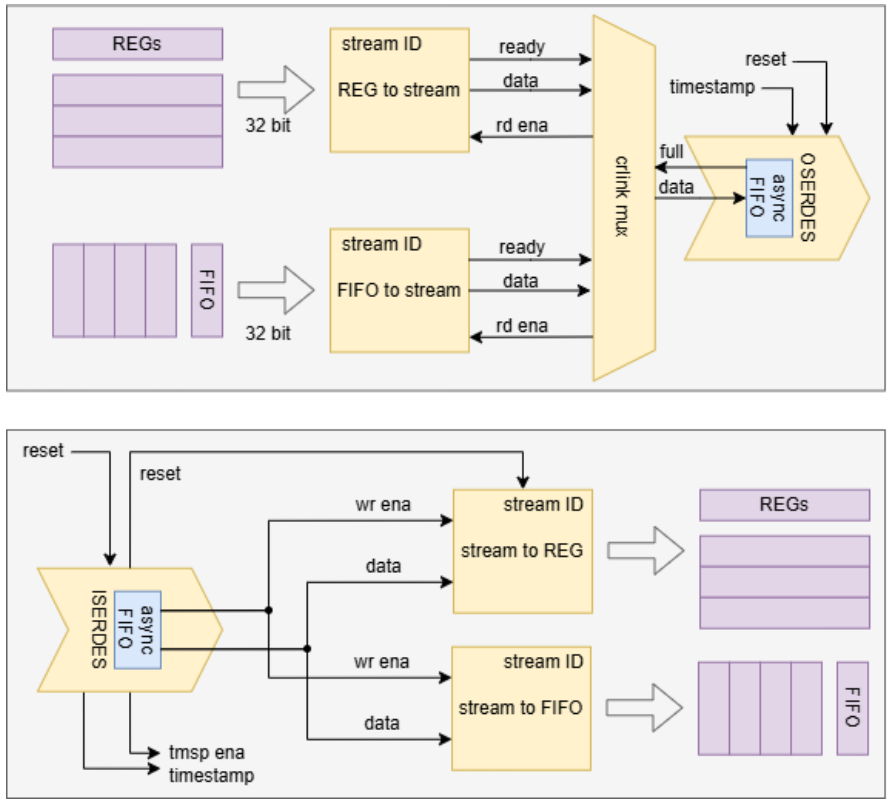
Completed Routing Design:

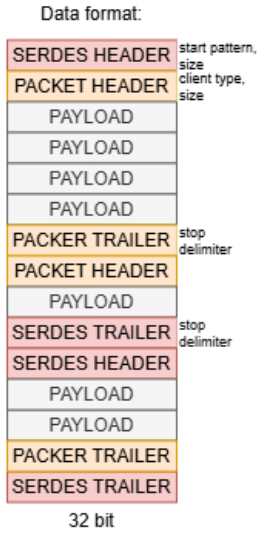
- ✓ Power control circuits
- ✓ FPGA peripheral connection
- ✓ SiPM bias voltage generator

Pending routing tasks:

- ❑ Clocks generator
- ❑ MCU peripheral connection
- ❑ FPGA TDC connections
- ❑ TDC loopback MUX connections

FPGA firmware development status: working on cross FPGA link



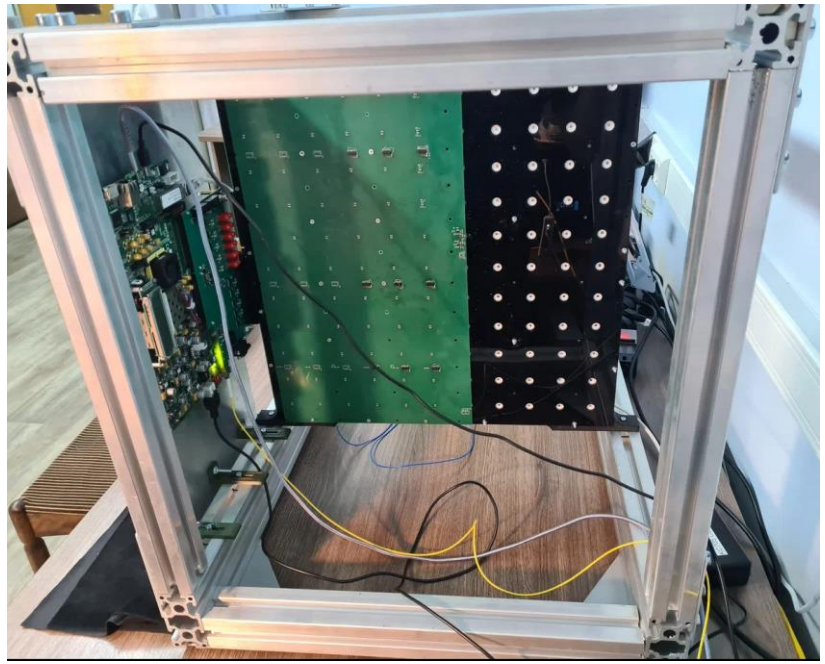


- The slave FPGAs are connected to the master FPGA via full-duplex LVDS lines.
- The connection is implemented using the (IO)SERDESE2 FPGA primitive. Phase alignment is achieved using IDELAYE2 primitive (fixed delay).
- 250 MHz SDR (250 Mbps) is equivalent to an event rate of 10 kHz per channel (84 ch. per FPGA)
- The link supports the transmission of:
  - Detector data
  - Configuration commands
  - White Rabbit (WR) timestamps (fixed latency)
  - Service messages
- Design features: Flexible setup and scalability.
- Current Status: Functionality verified in VHDL simulation.

Status of the HW development

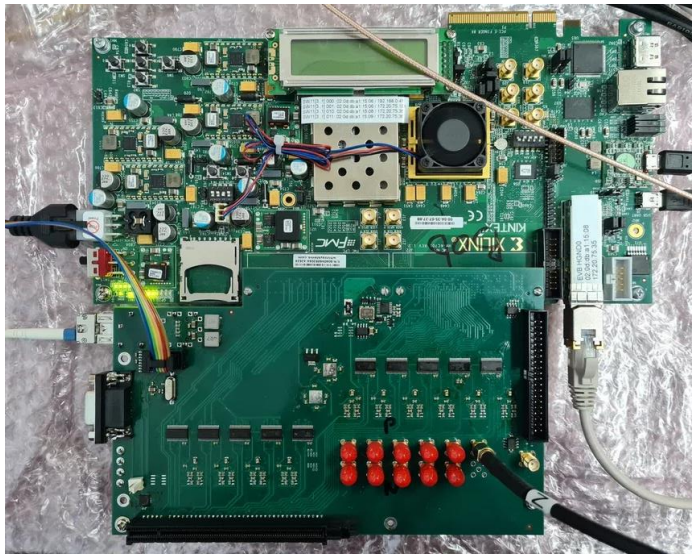
The full functional HGND prototype was assembled.

- Based on readout board proto #1
- ¼ of matrix SiPM & LED channels
- Ethernet readout
- White Rabbit synchronization
- PCIe connector for scintillation matrix
- Temperature sensor
- SiPM threshold DAC control
- Match HGND geometry



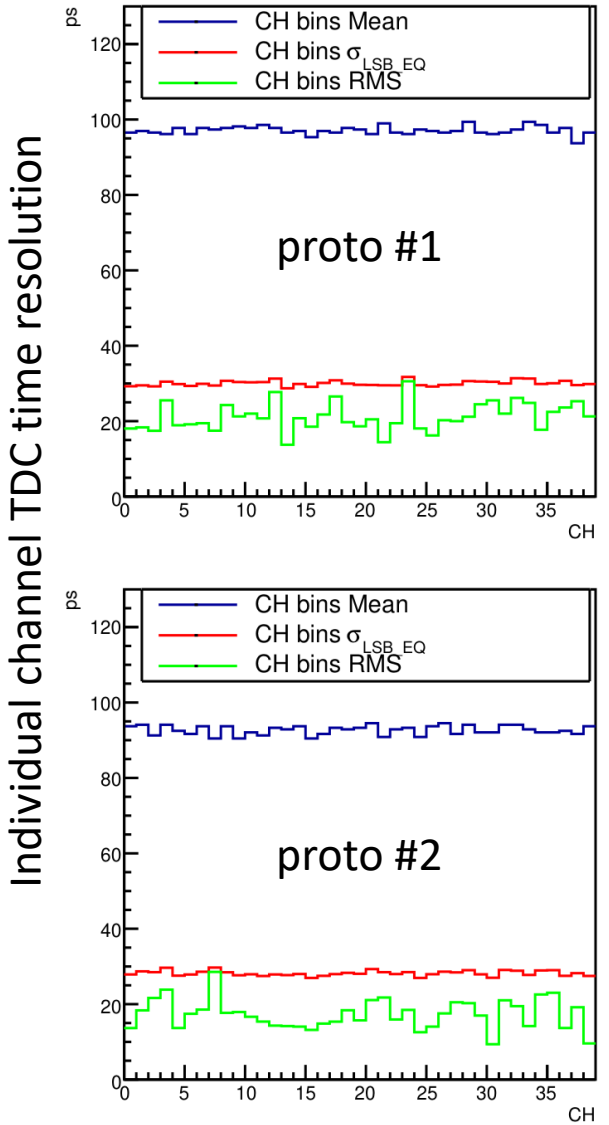
Readout board proto #2

- Standalone tests

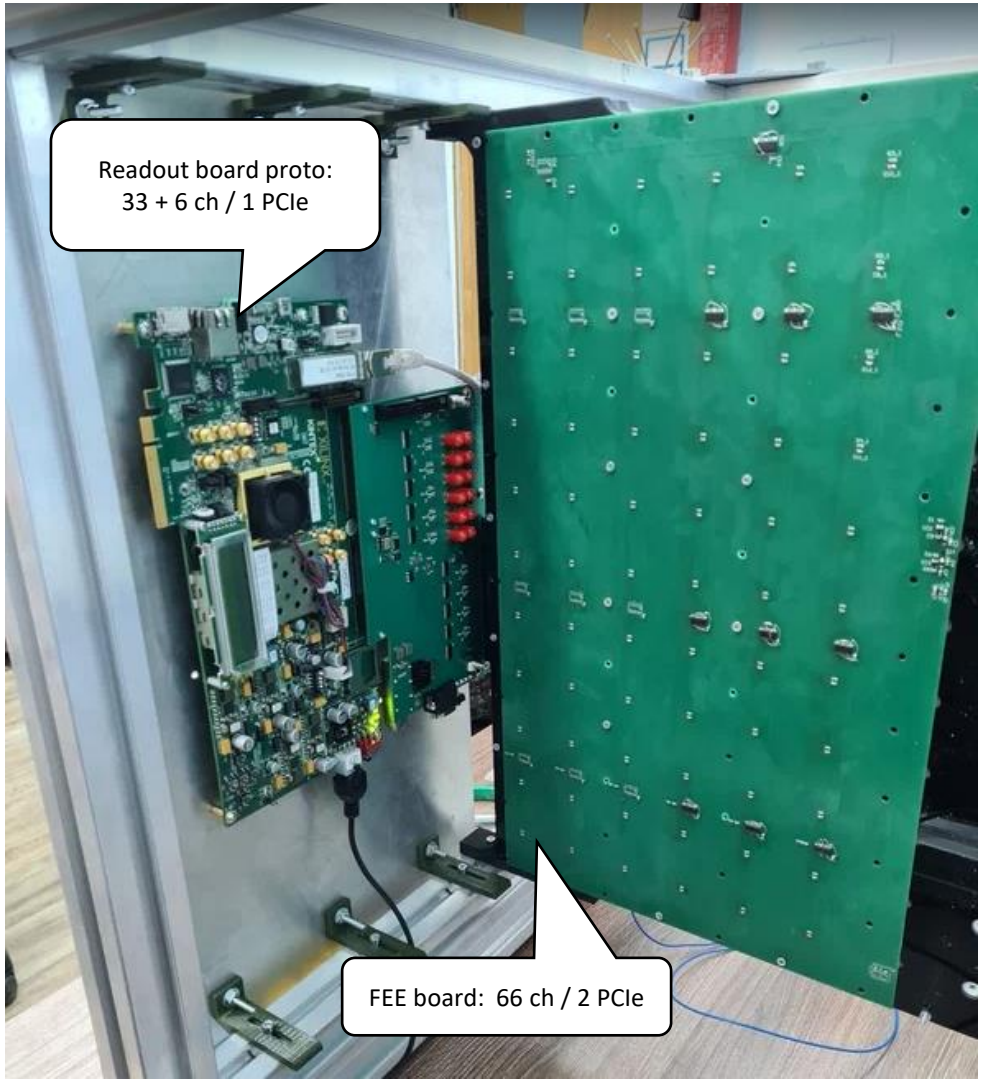


Readout board proto #3

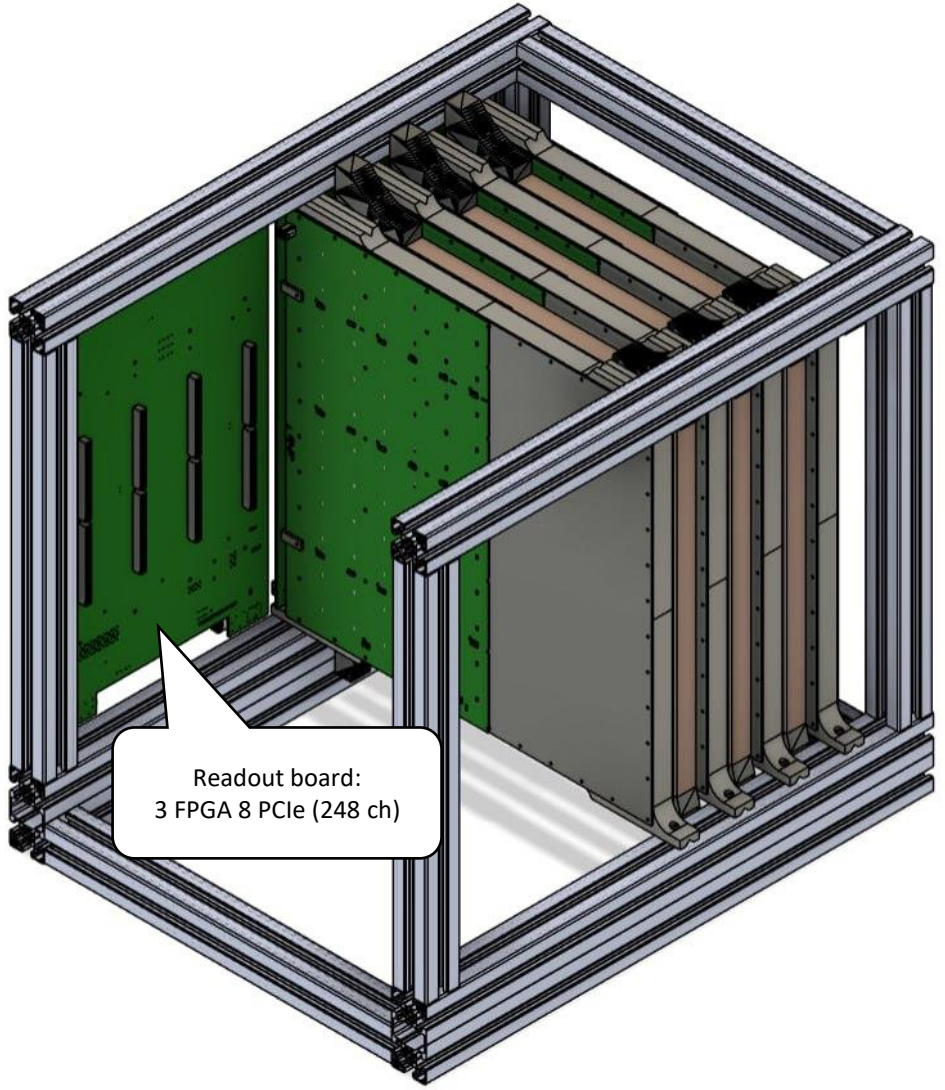
- Assembling



HDND geometry

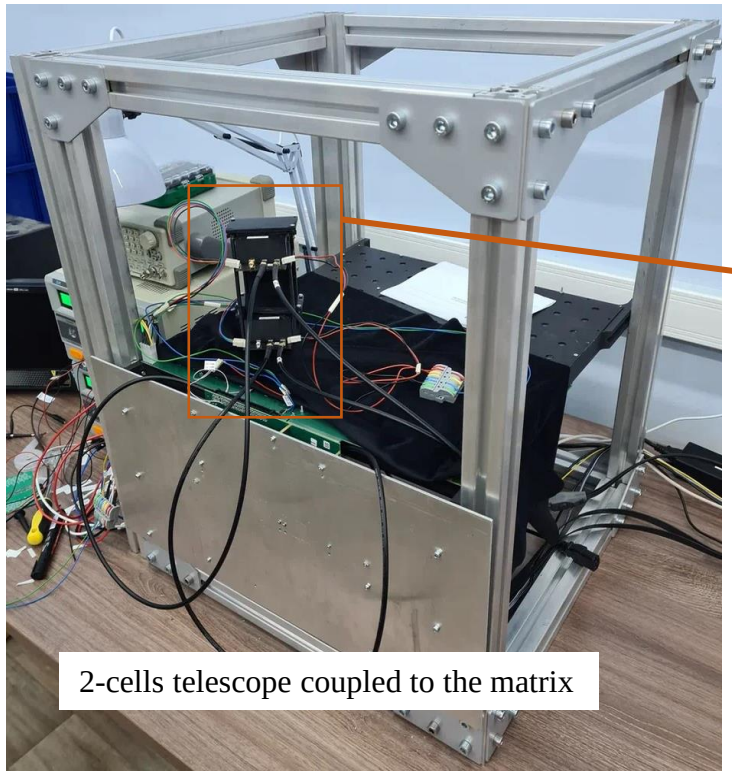


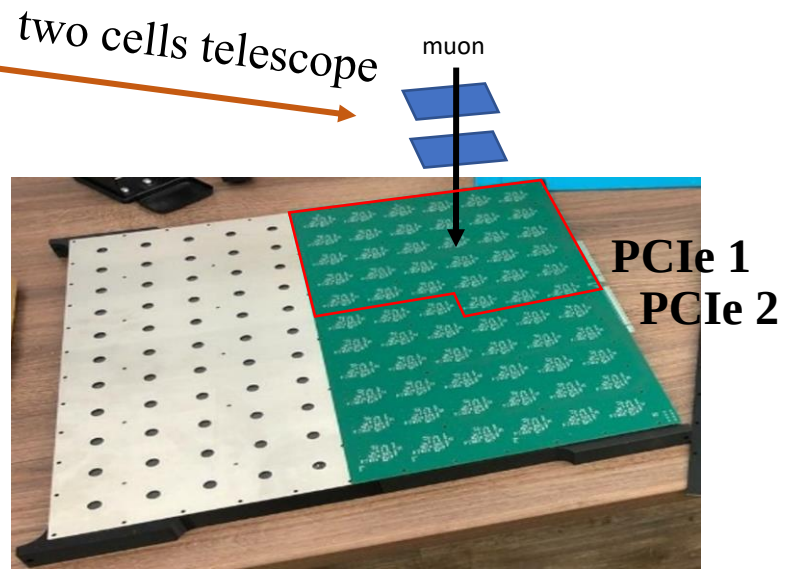
¼ matrix readout prototype



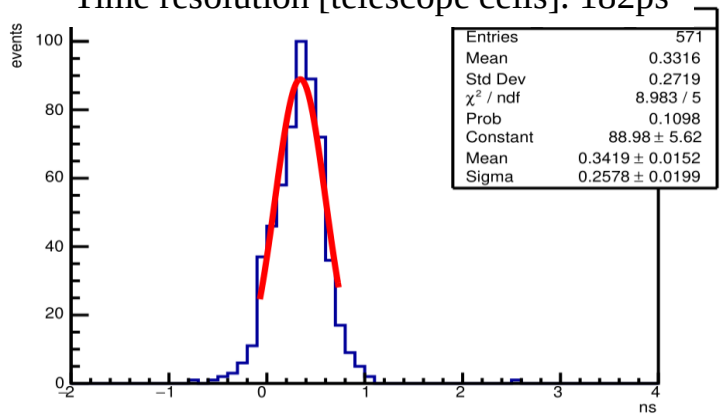
HGND geometry sketch

The HGND prototype tests with cosmic muons

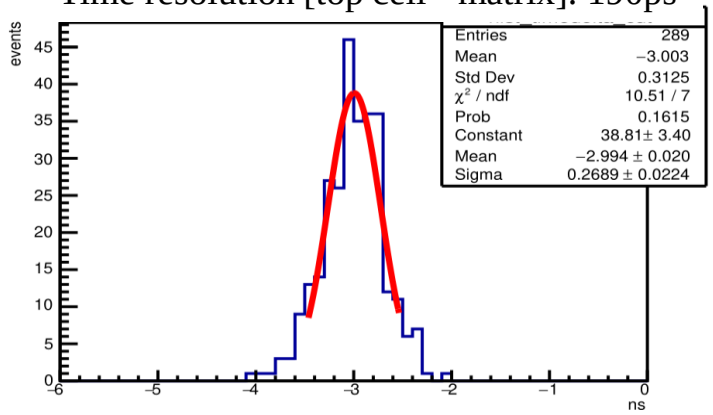




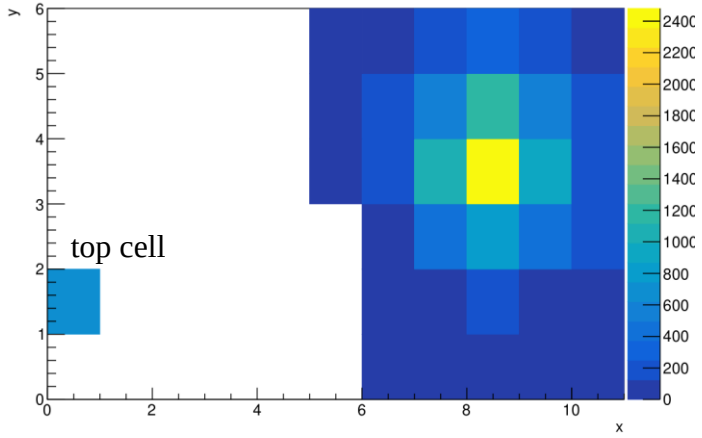
Time resolution [telescope cells]: 182ps



Time resolution [top cell - matrix]: 190ps



The time correlated events: bottom cell – matrix



Cosmic runs was collected in two configuration:

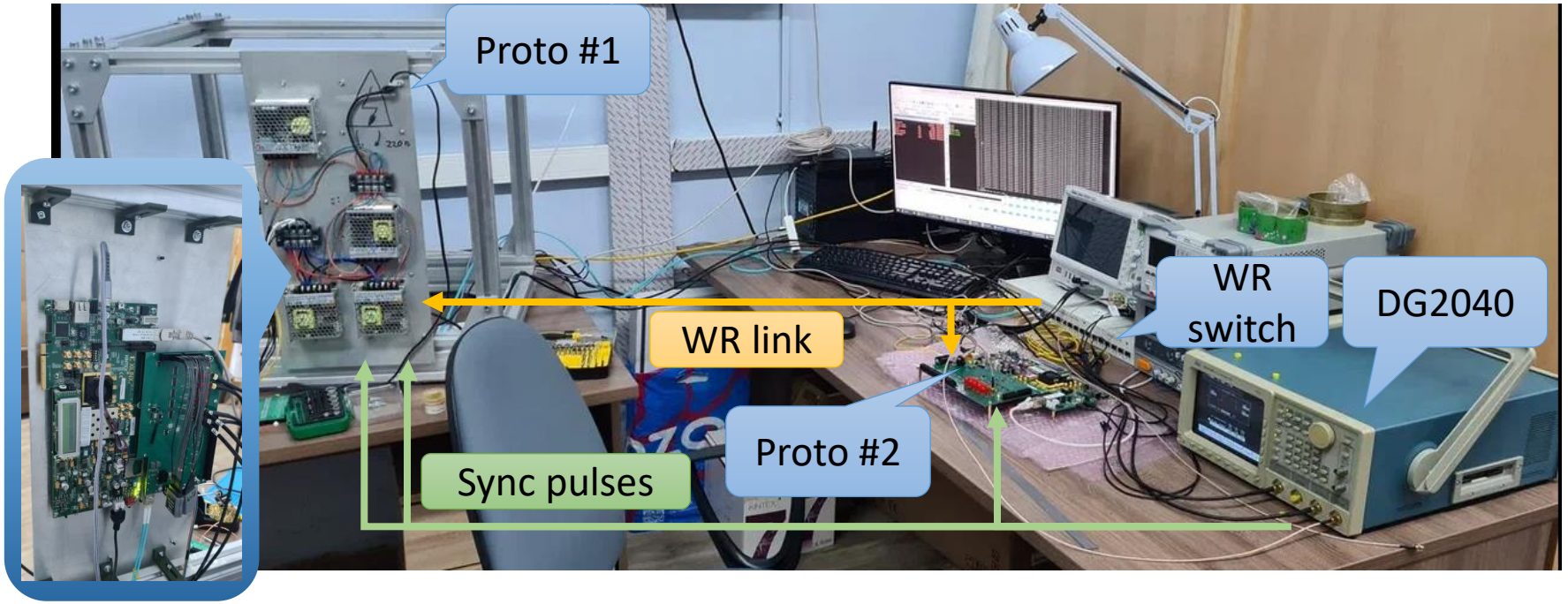
- Normal position (like placement on the BM@N)
- Horizontal position with 2 cells telescope

✓ Results shows the same time resolution between telescope connected via SMA and matrix connected via PCIe: 182 and 190 ps per channel.

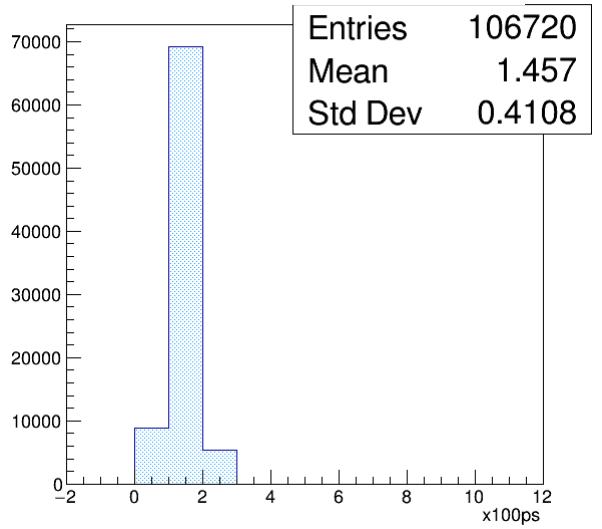
➢ Testing all functionality: readout, LED calibration, temperature sensor, geometry, light-isolating

➢ FEE Mass-production

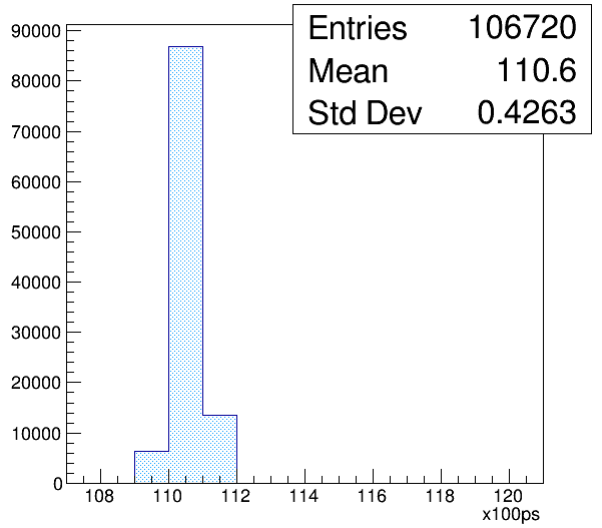
WR cross board synchronization test



Two channels on proto #1

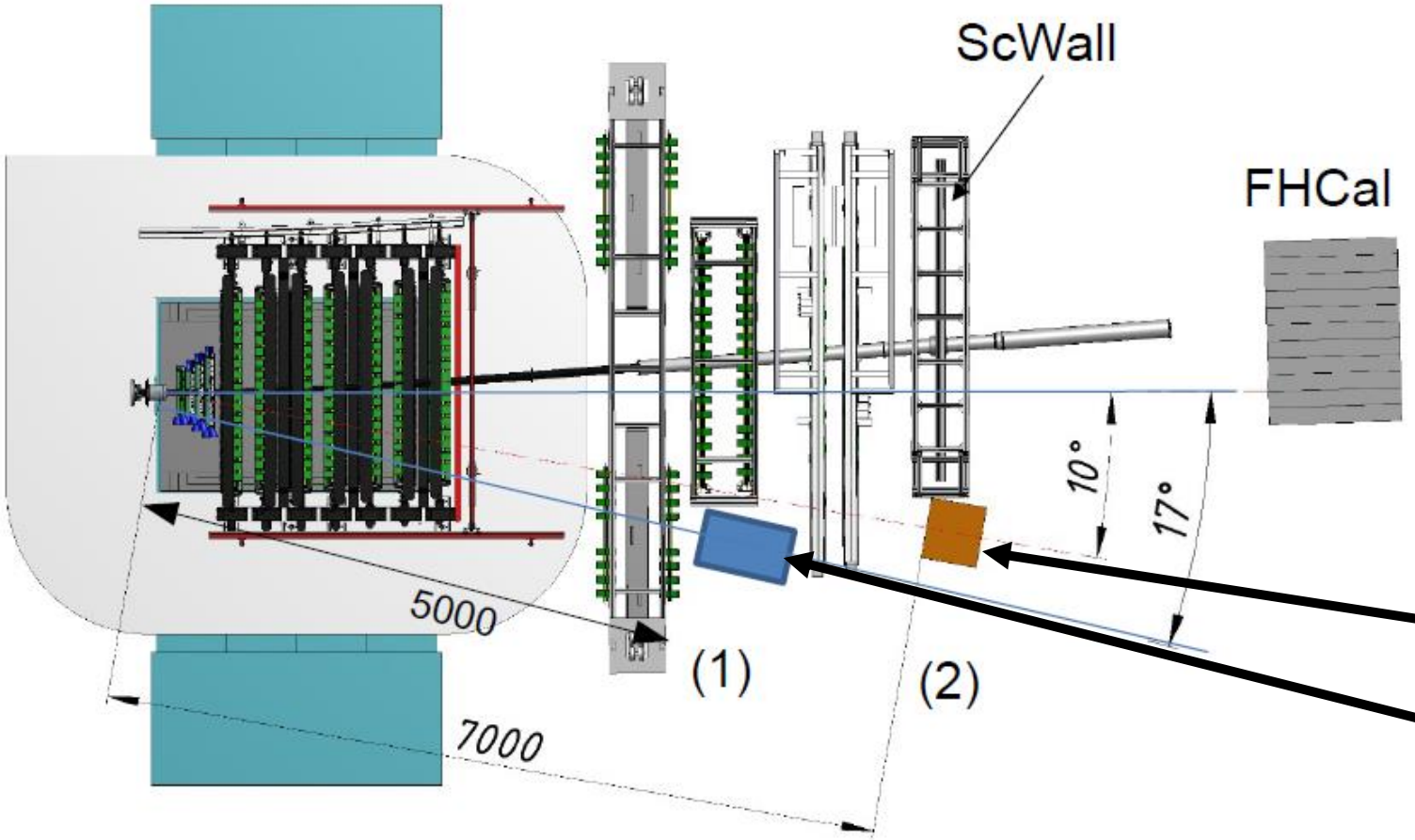


proto #1 – proto #2



- Two prototype readout boards are synchronized using a White Rabbit switch.
  - Proto #1 receives pulses from two channels.
  - Proto #2 receives pulses from a single channel.
- The intrinsic jitter of the **DG2040** generator is **5 ps**.
- The TDC channel's time resolution is **30 ps**.
- The time difference distributions (both cross-board and single-board) is **42 ps** and match the TDC's resolution.

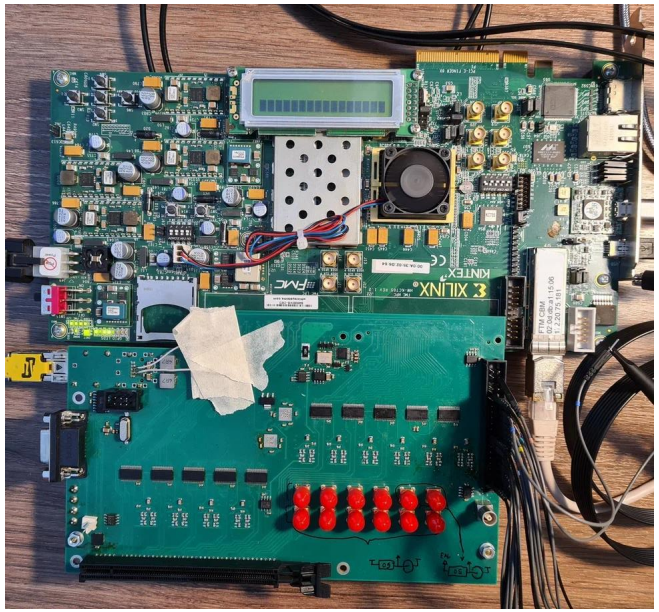
Planned location of the stand with 1 HGND module at the next session on BM@N



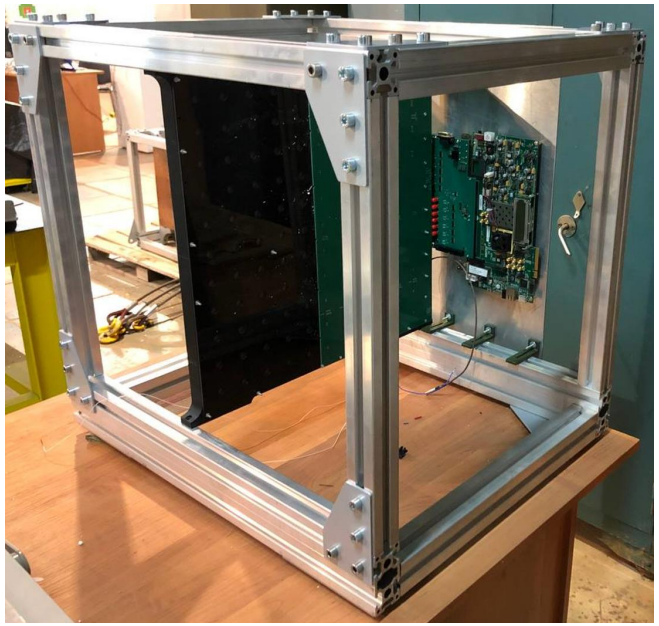
Planned HGND assembly with one module at the next session on BM@N

¼ matrix readout prototype (33 ch)

Planned setup at BM@N: two steps



HGND Proto assembly



Step 1: the setup allows testing:

1. HGND control via common DCS
2. Data readout: rate + digitizing
3. White Rabbit events timestamps
4. Trigger selection & synchronization

Step 2: the setup allows testing (additionally) :

1. HGND real time resolution
2. Physics events time correlation
3. FEE behavior in real beam conditions
4. Detector events load

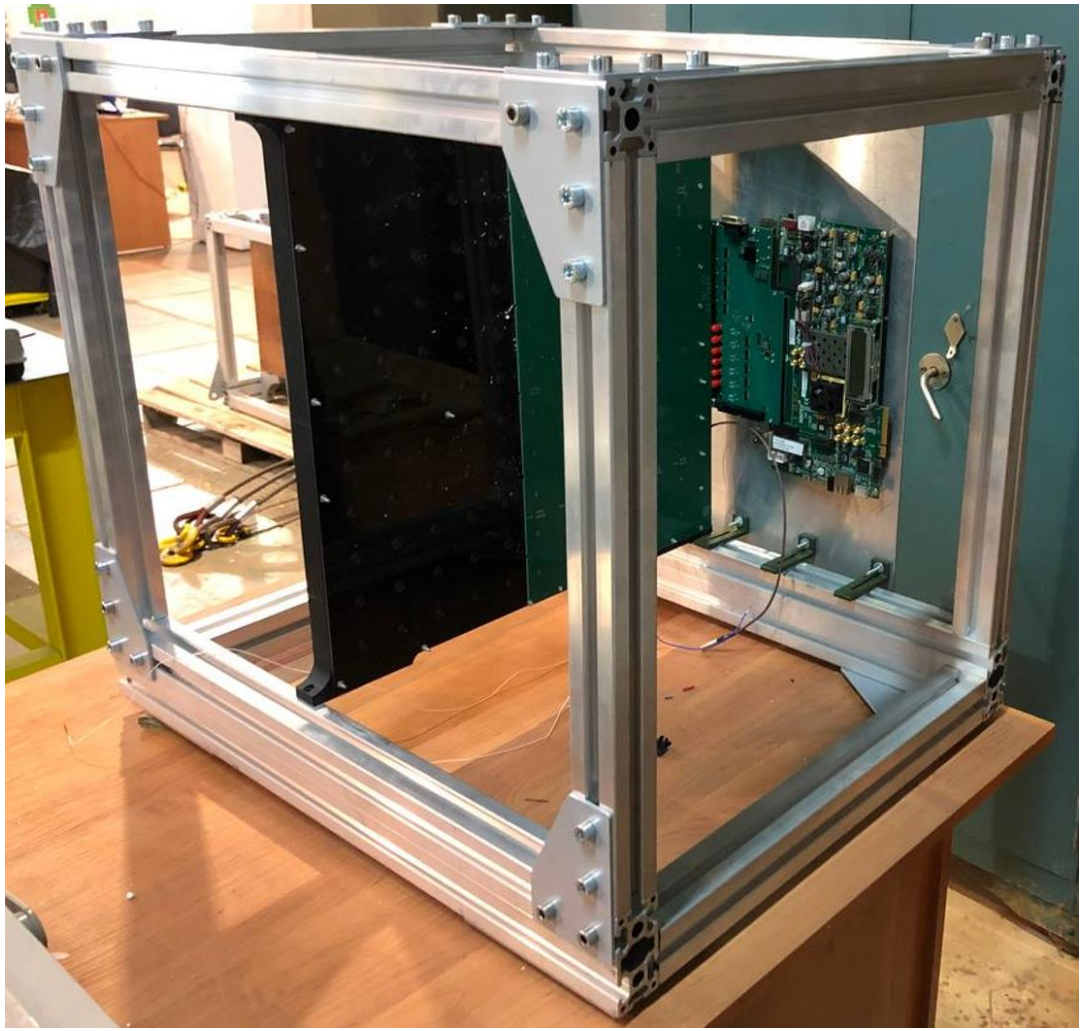
Conclusions

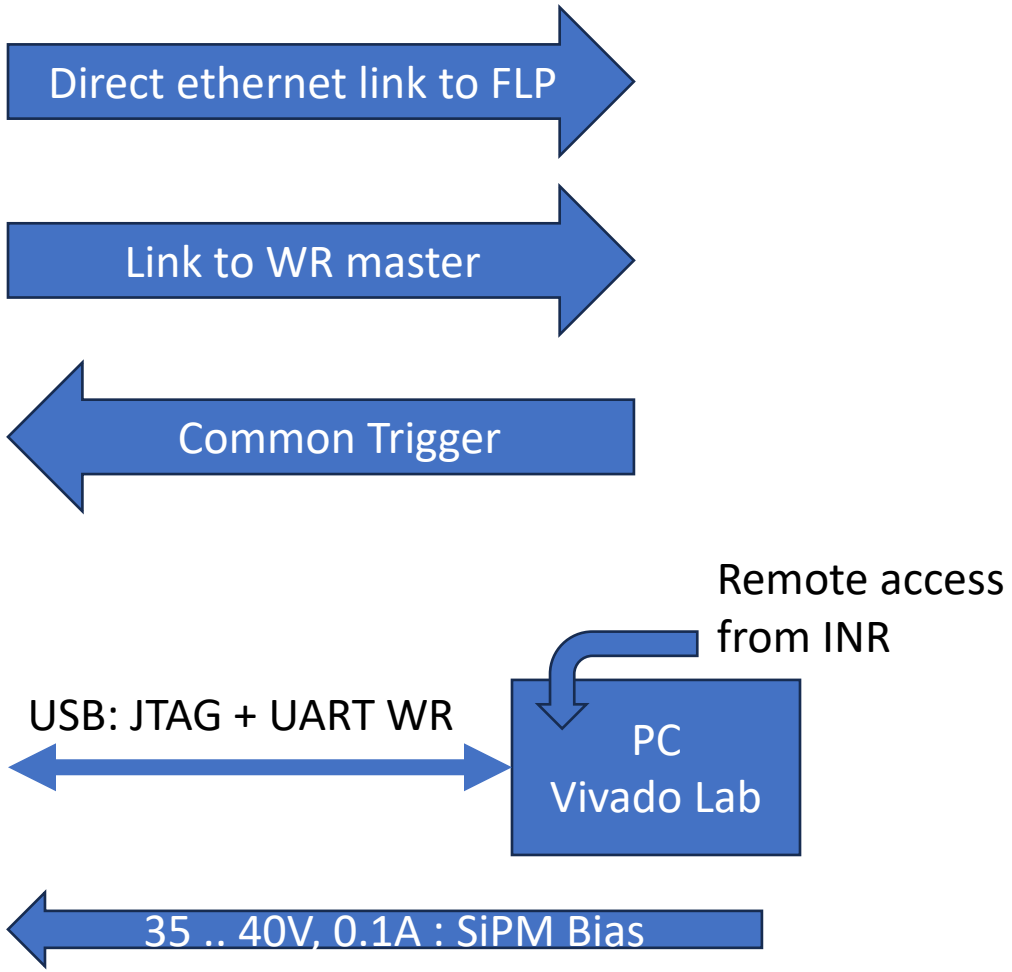
- Status of the HGND readout development:
  - ✓ Working on firmware and software.
  - ✓ Routing the readout board is in progress.
  - ✓ Two boards WR synchronization test done.
  - ✓ Ready for Integration prototype into BM@N and tests on the beam.

Thank you for your attention!

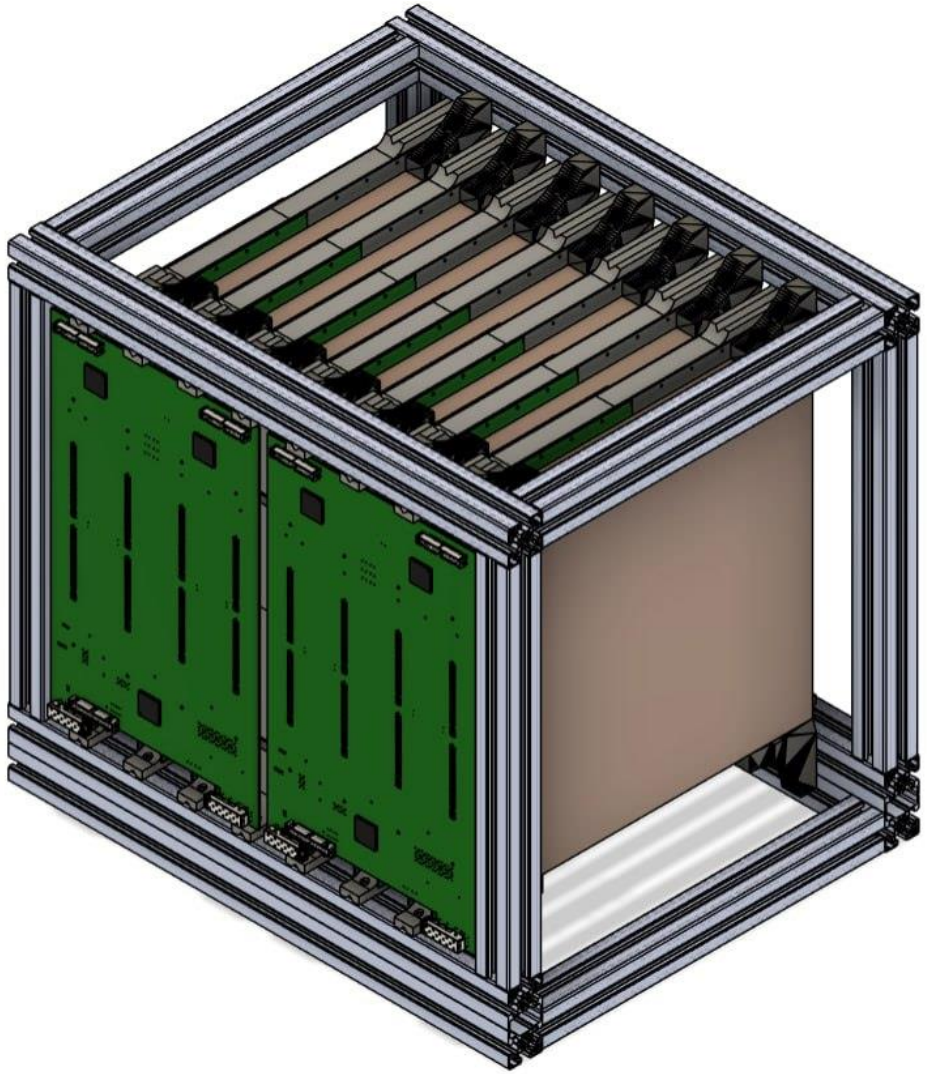
BACKUP

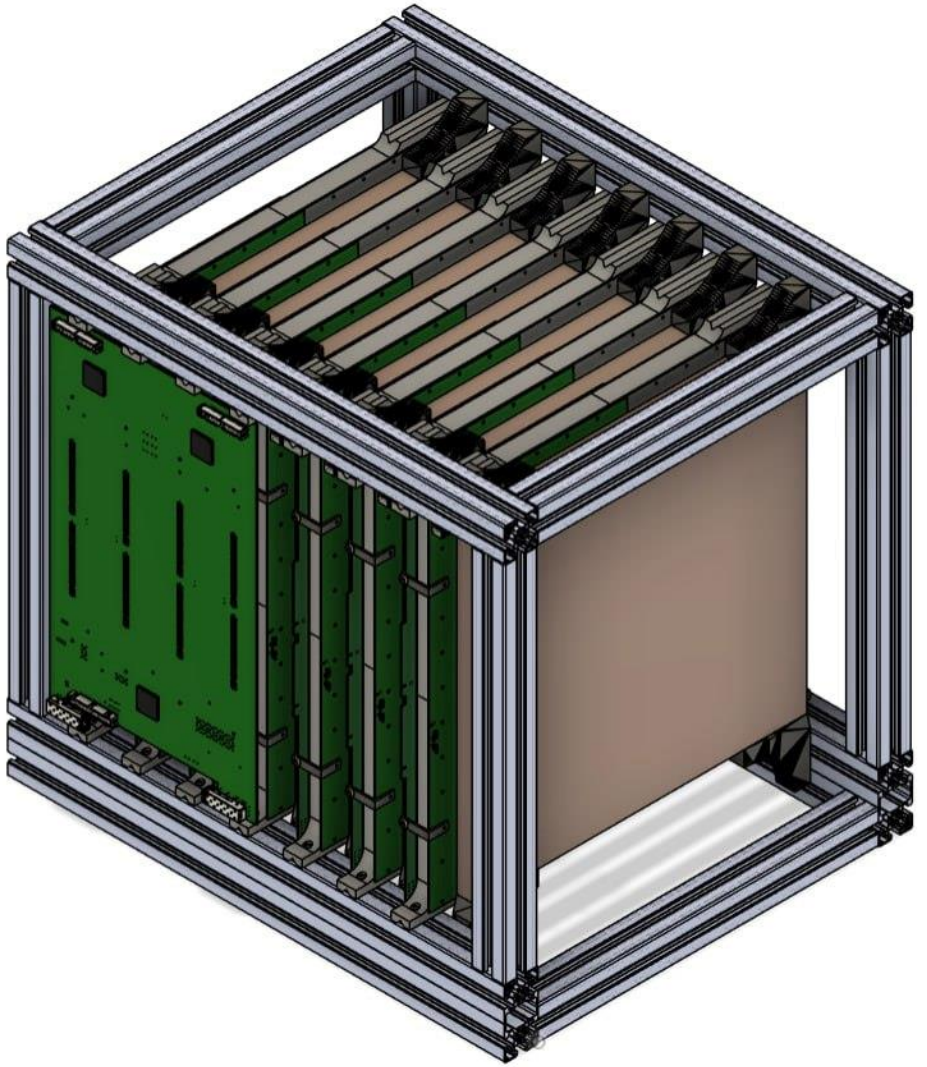
HGND proto setup at BM@N





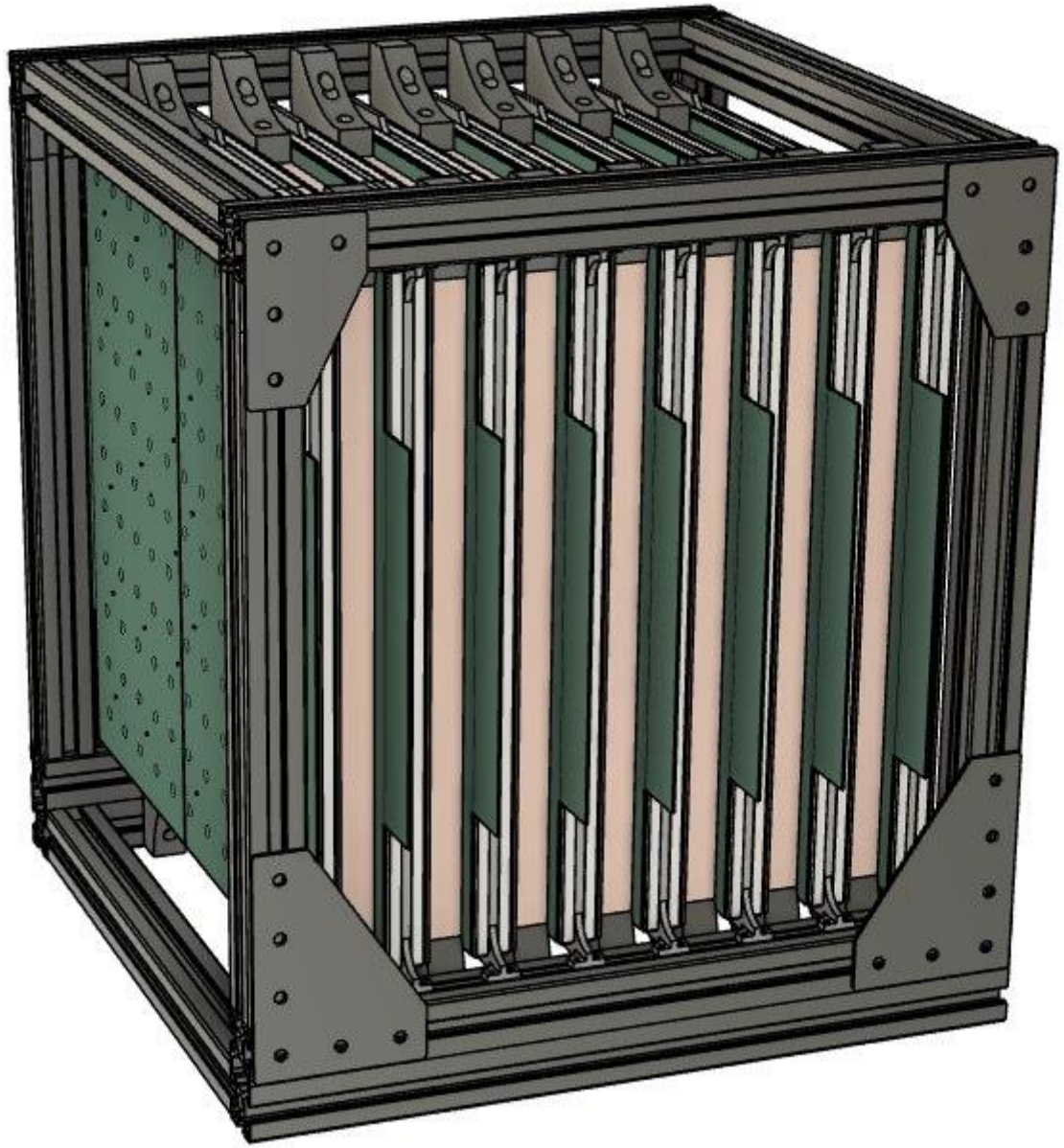
HDND geometry





Detector “block”

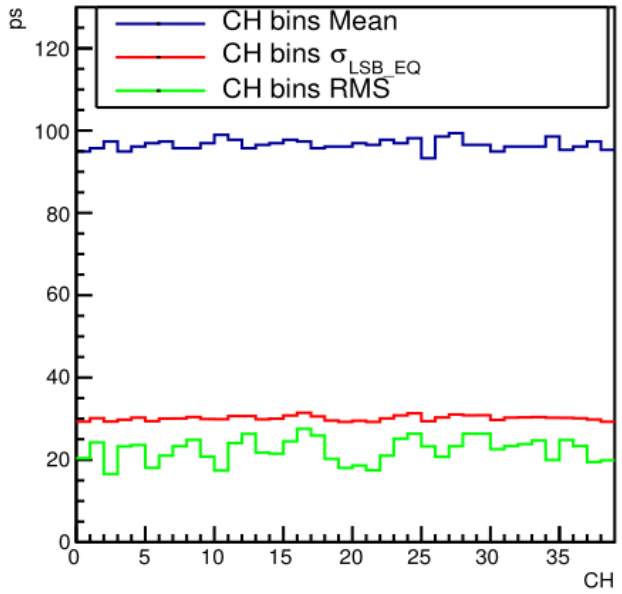
- Each block consists of:
  - A VETO-layer
  - 8 Cu absorbers
  - 8 sensitive layers
    - 11x11 grid of scintillatiors each
- Assembly is light-tight and air-cooled
- Framing is built with light-weight Al profiles



The FPGA TDC test results

PROTO_V2

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

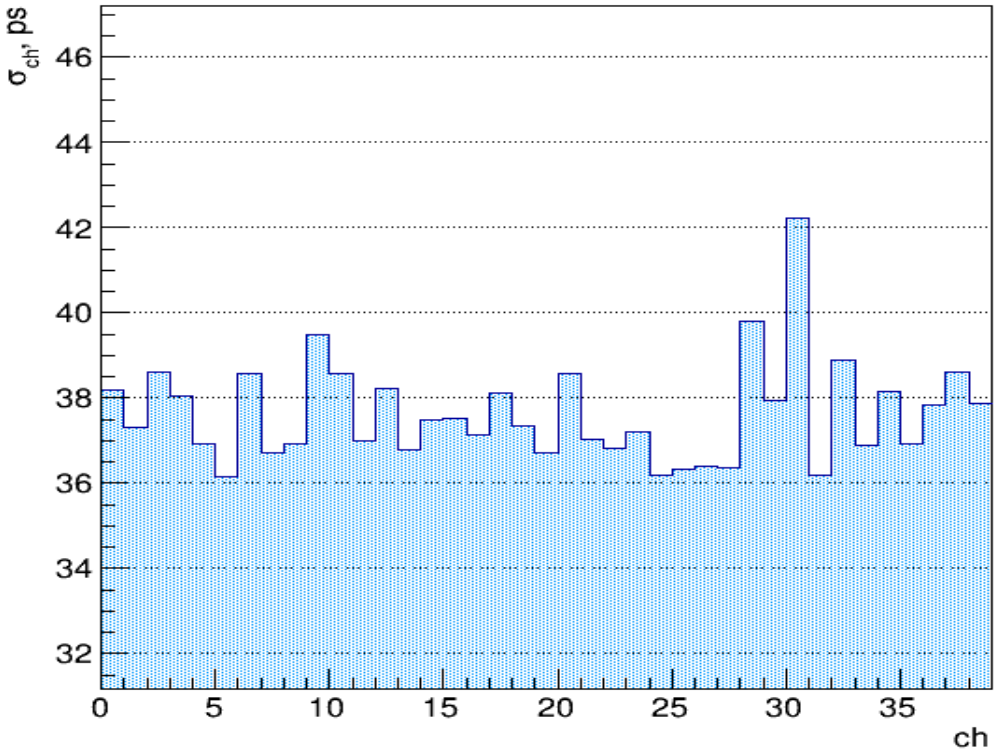


The equivalent LSB precision is ~30ps.

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

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

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

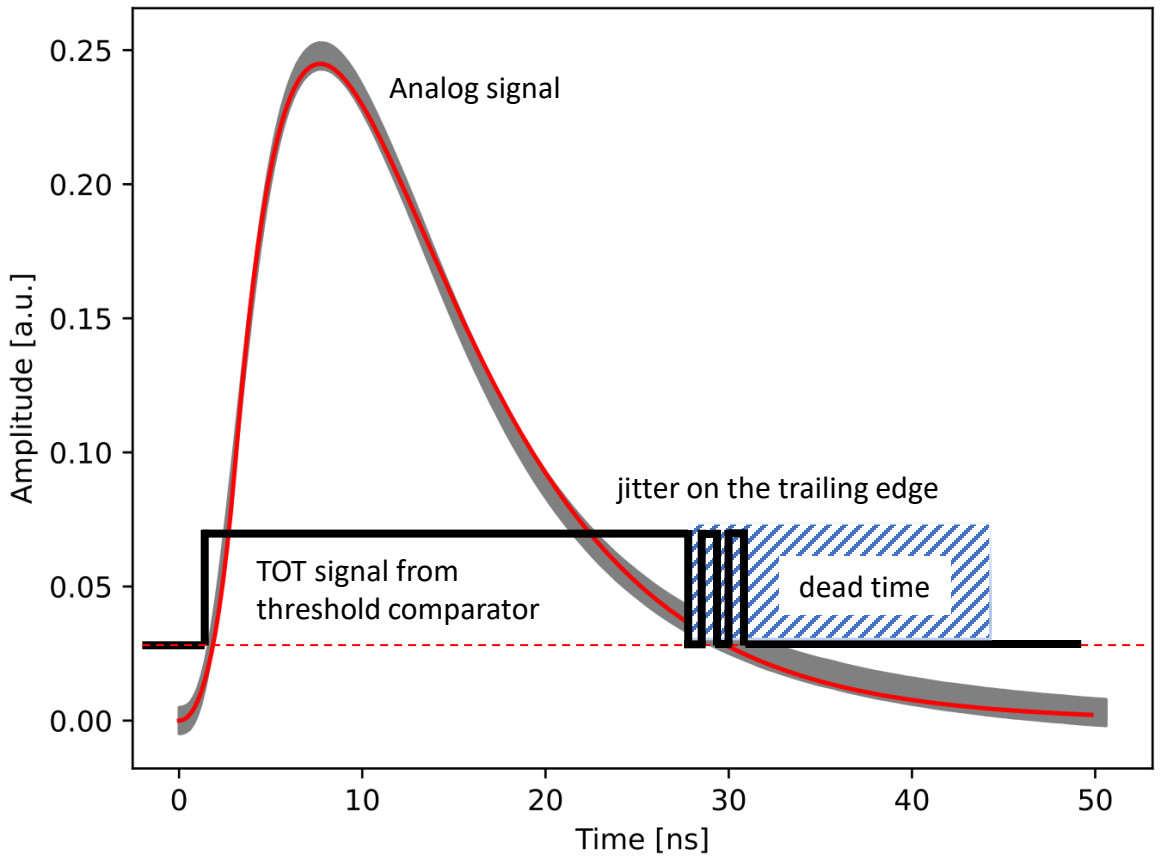


The TDC channel precision. Measured with the data generator DG2040 (Cycle-to-Cycle Jitter 5ps).

The accurate TDC precision measurements:

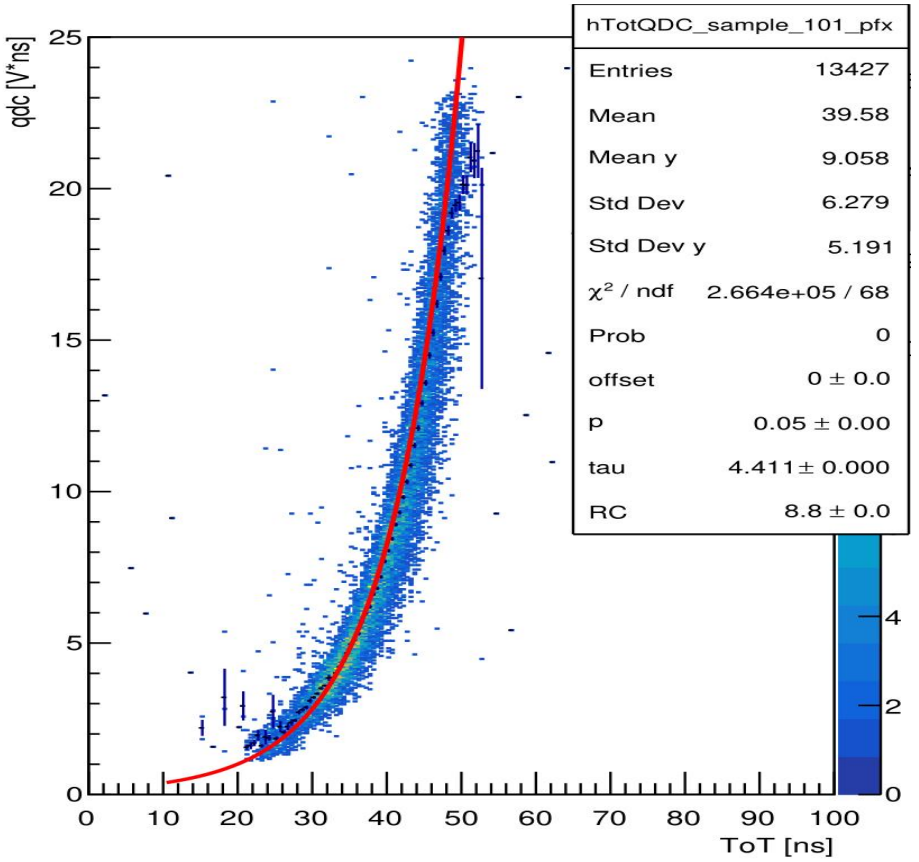
- Pulses are asynchronous to TDC clock.
- Pulses length is various 9.8 .. 10.3 ns with step 50ps
- Source pulses jitter is 5ps

TDC Time Over Threshold (TOT)



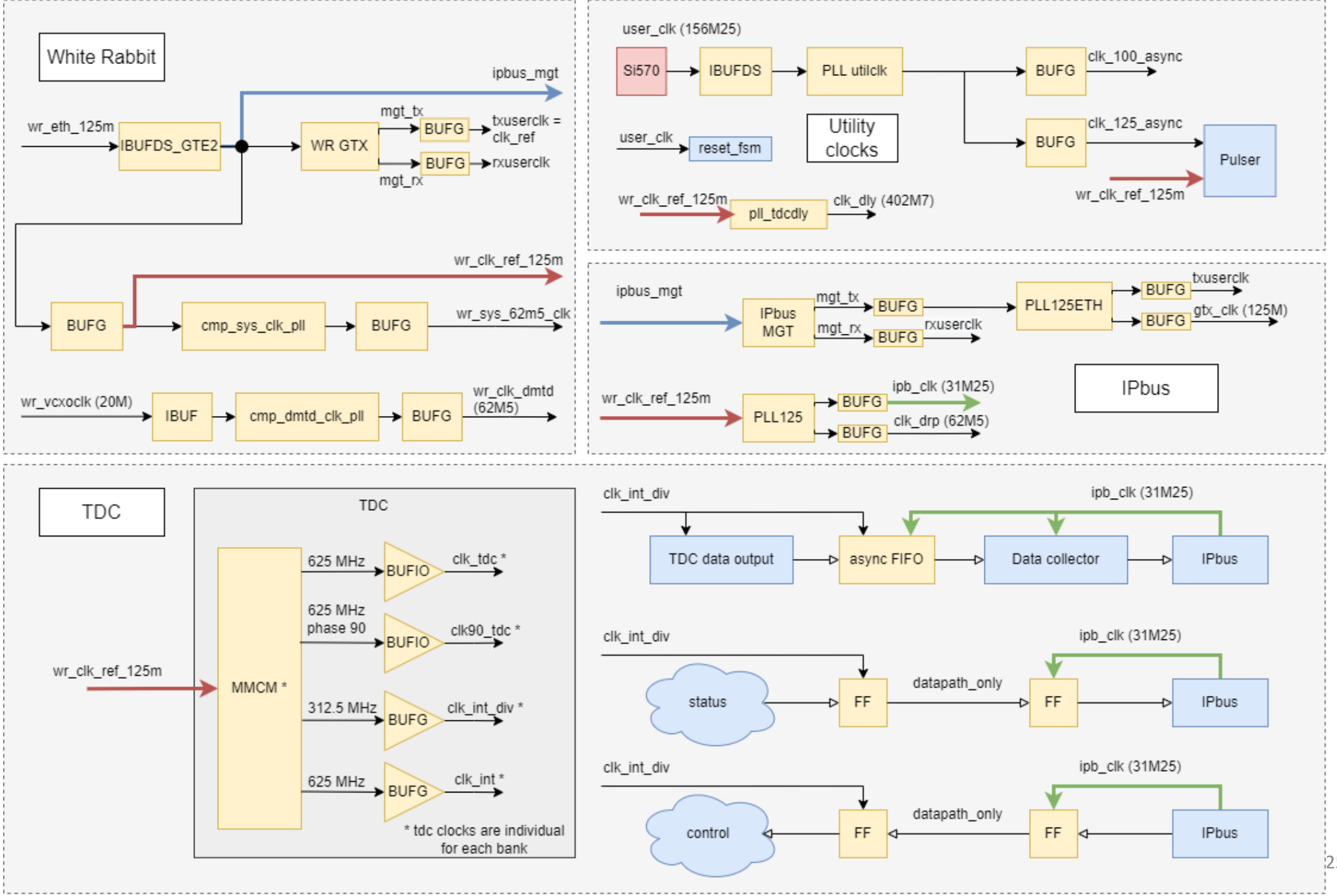
- The threshold is tunable around 20 mV
- Signals length range is 20 – 60 ns
- Signals less than 6.4 ns are rejected for noise reduction
- Dead time is tunable in range 30 – 200 ns for comparator jitter filtering
- Minimum TOT time and dead time available in FPGA TDC are 3.2 ns

Amplitudes vs TOT time with analytical forecast

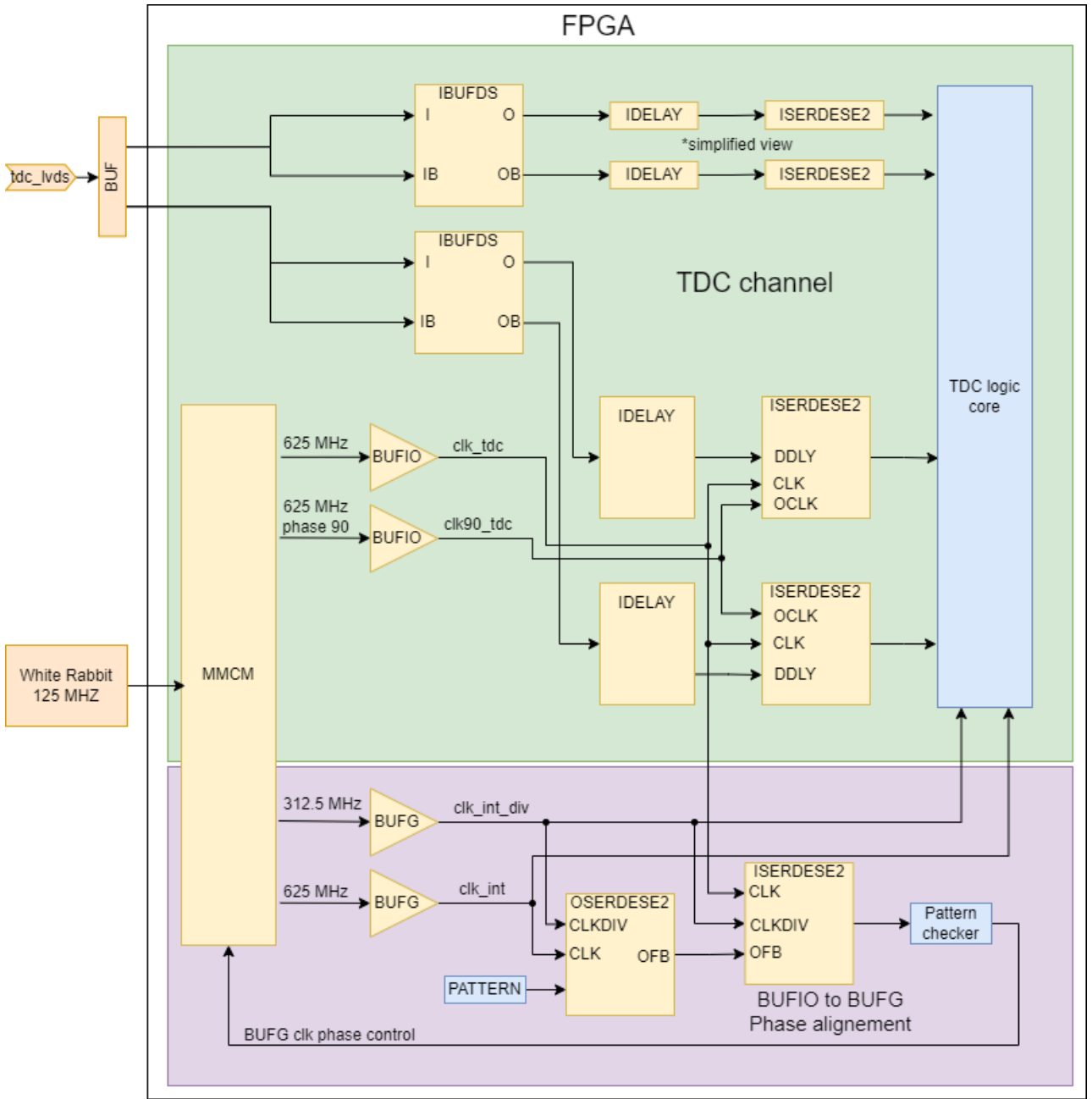


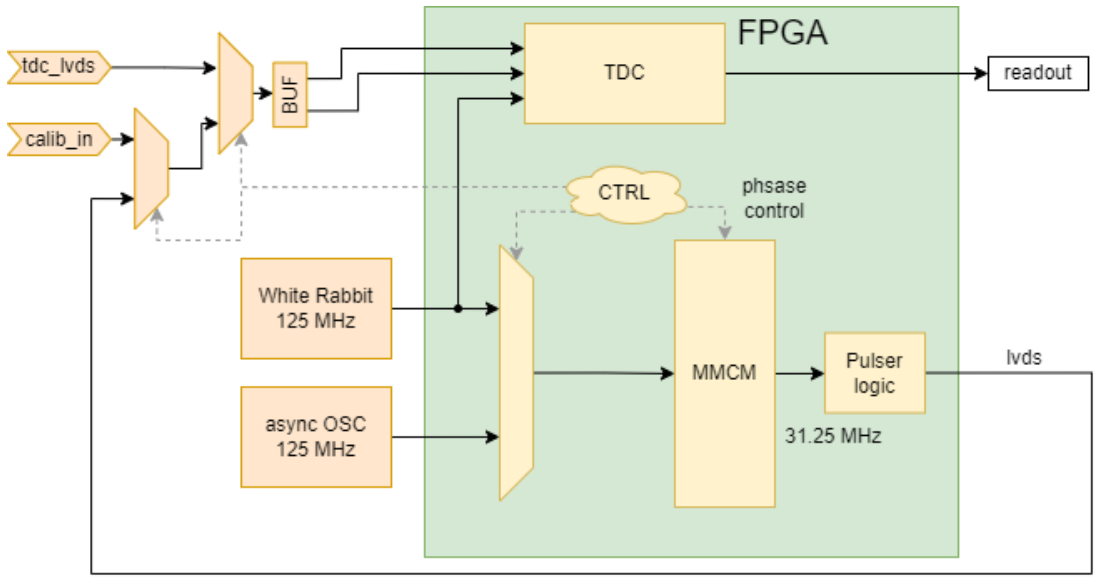
- TOT amplitude resolution is in range 14 - 22%
- Is used for time slewing correction

[2] N. Karpushkin, D. Finogeev, F. Guber, D. Lyapin, A. Makhnev et al., ***Analytical description of the time-over-threshold method based on time properties of plastic scintillators equipped with silicon photomultipliers,*** *DOI: 10.1016/j.nima.2024.169739*

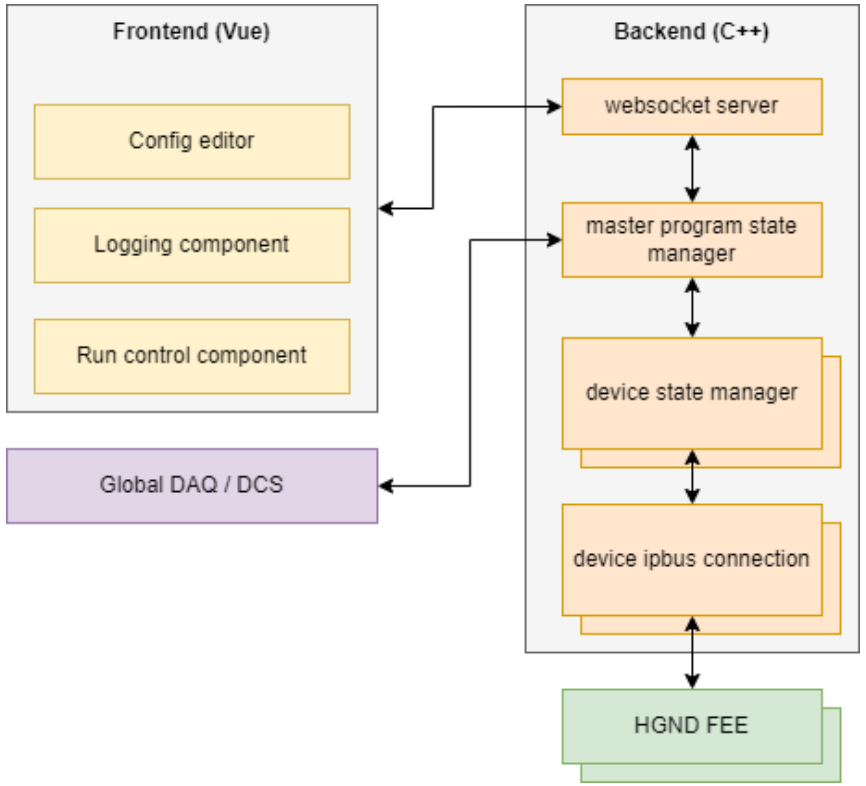

White Rabbit
ipbus_mgt
wr_eth_125m
IBUFDS_GTE2
WR GTX
mgt_tx
BUFG
txuserclk = clk_ref
BUFG
rxuserclk
mgt_rx
wr_clk_ref_125m
BUFG
cmp_sys_clk_pll
BUFG
wr_sys_62m5_clk
wr_vcxoclk (20M)
IBUF
cmp_dmtd_clk_pll
BUFG
wr_clk_dmtd (62M5)
user_clk (156M25)
Si570
IBUFDS
PLL utilclk
BUFG
clk_100_async
user_clk
reset_fsm
Utility clocks
BUFG
clk_125_async
Pulser
wr_clk_ref_125m
wr_clk_ref_125m
pll_tdcdly
clk_dly (402M7)
ipbus_mgt
IPbus MGT
mgt_tx
BUFG
mgt_rx
BUFG
rxuserclk
PLL125ETH
BUFG
txuserclk
BUFG
gtx_clk (125M)
IPbus
wr_clk_ref_125m
PLL125
BUFG
ipb_clk (31M25)
BUFG
clk_drp (62M5)
TDC
TDC
wr_clk_ref_125m
MMCM *
625 MHz
BUFIO
clk_tdc *
625 MHz phase 90
BUFIO
clk90_tdc *
312.5 MHz
BUFG
clk_int_div *
625 MHz
BUFG
clk_int *
* tdc clocks are individual for each bank
clk_int_div
ipb_clk (31M25)
TDC data output
async FIFO
Data collector
IPbus
clk_int_div
ipb_clk (31M25)
status
FF
datapath_only
FF
IPbus
clk_int_div
ipb_clk (31M25)
control
FF
datapath_only
FF
IPbus

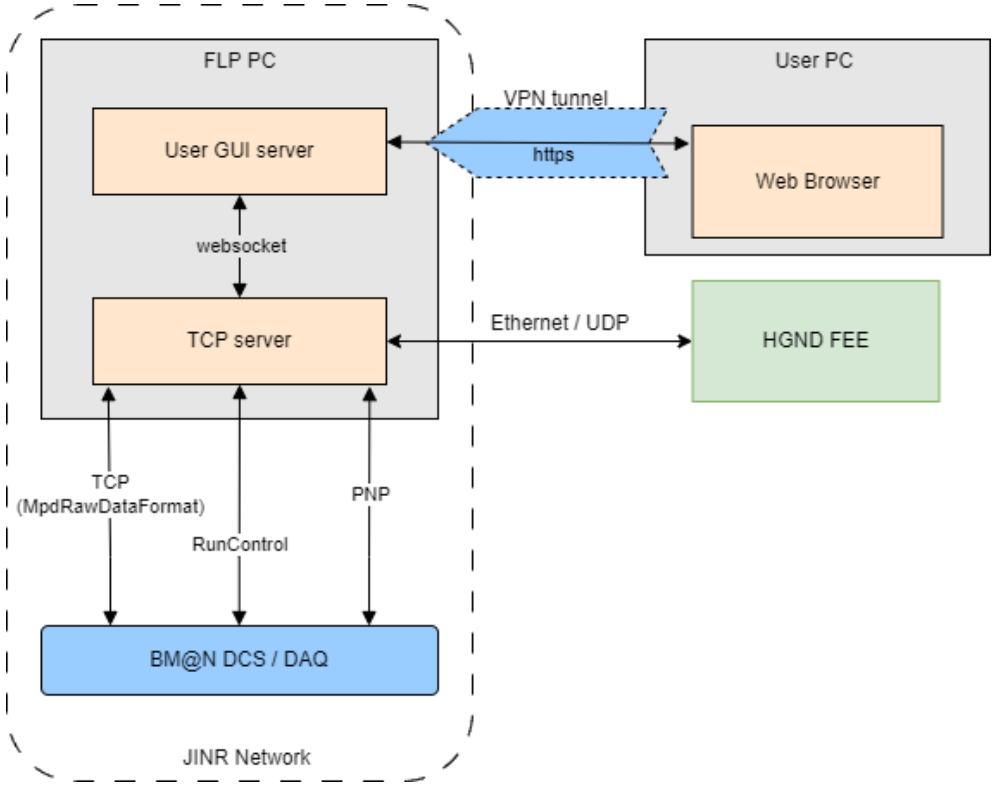
FPGA TDC and calibration pulser clocks layout





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