Check-Sort-Push: a readout, suppression and transmission on a shared link between frontend and backend electronics

Zhen-An Liu

CEPC2023, Nanjing China

Oct. 23-27 2023

Outline

- Motivation
- TDAQ R&D Setup
- Why do we propose a Check-Sort-Push
- Implementation and test results
- Summary

Motivation

- Questions on TDAQ design: Does CEPC needs a Hardware Trigger(L1 trigger)
- If No (trigger-less/Software trigger/HLT)
  - DAQ: time slice based FEE data readout + eventbuilding + HLT + Saving
  - Clock/Fast control for FEE
  - Slow control for FEE
- If YES
  - L1Pass based data readout in FEE (+eventbuilding + HLT + Saving)
    - Clock/Fast-Slow control via TCDS
  - L1Pass based data readout in BEE (+eventbuilding + HLT + Saving)
    - Unified shared link between FEE/BEE
      - Continuous transmission without compression(Zero suppression)
      - Continuous transmission withcompression(Zero suppression)
      - Local trigger output to subsystem and global trigger
    - Clock/Fast/slow control to FEE via BEE from TCDS
- Motivation
  - Uncover the relative issues/problems

TDAQ R&D proposal in 2021

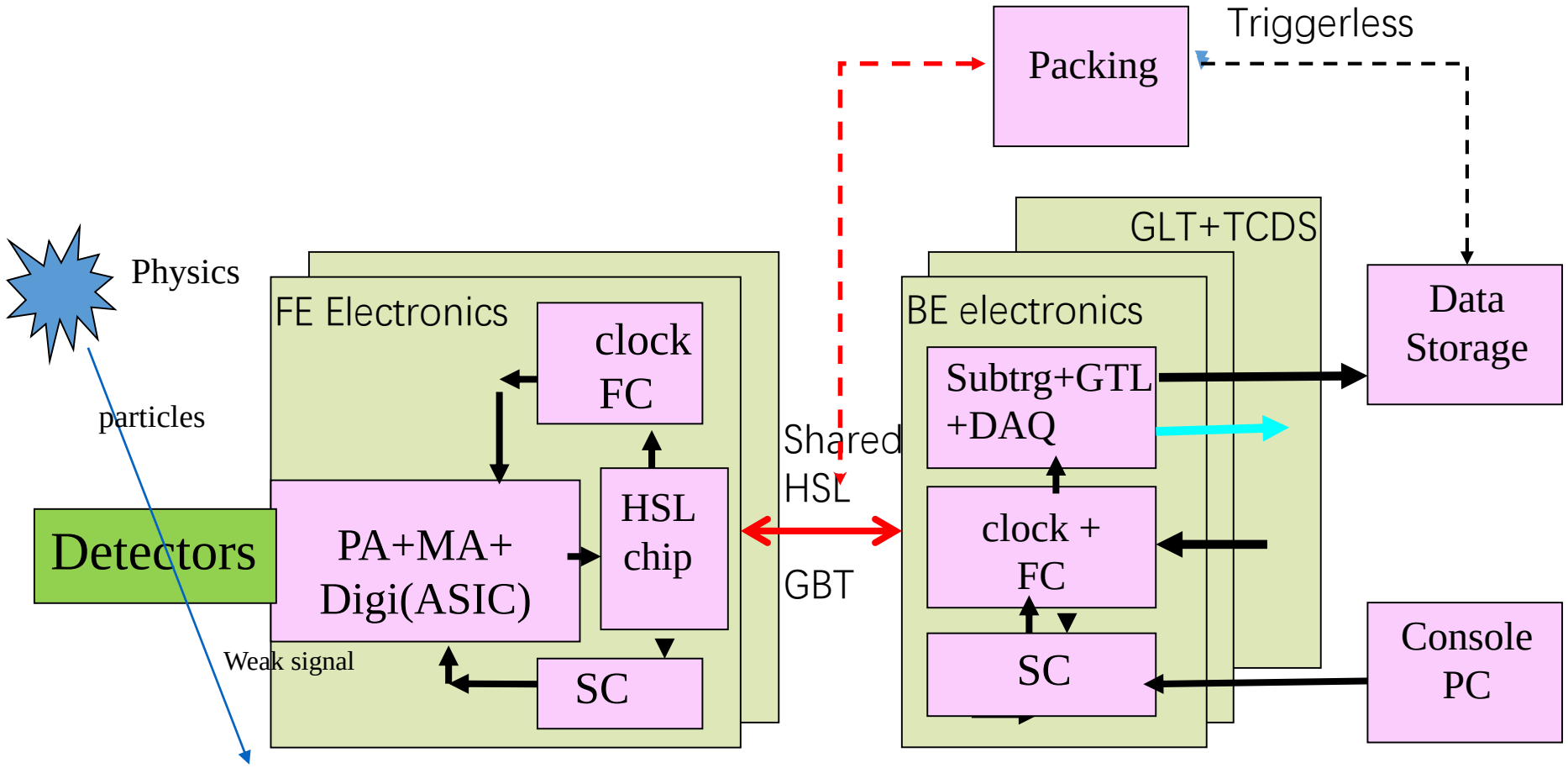
TDAQ Proposal fc

- Baselines
  - Supports both readout
    - Trigger based and
    - Triggerless
  - Supports continuous /data driven
  - Develop unified down link to FEE
    - Fast control(clock, Synch, BXs, L1,..
    - Slow control(parameter setting, run c
  - Provide FEE high BW data way
    - 4.8-10 Gbps links
  - Provides DAQ
    - 10-100 Gbps data links

TDAQ Proposal for CEPC(2)

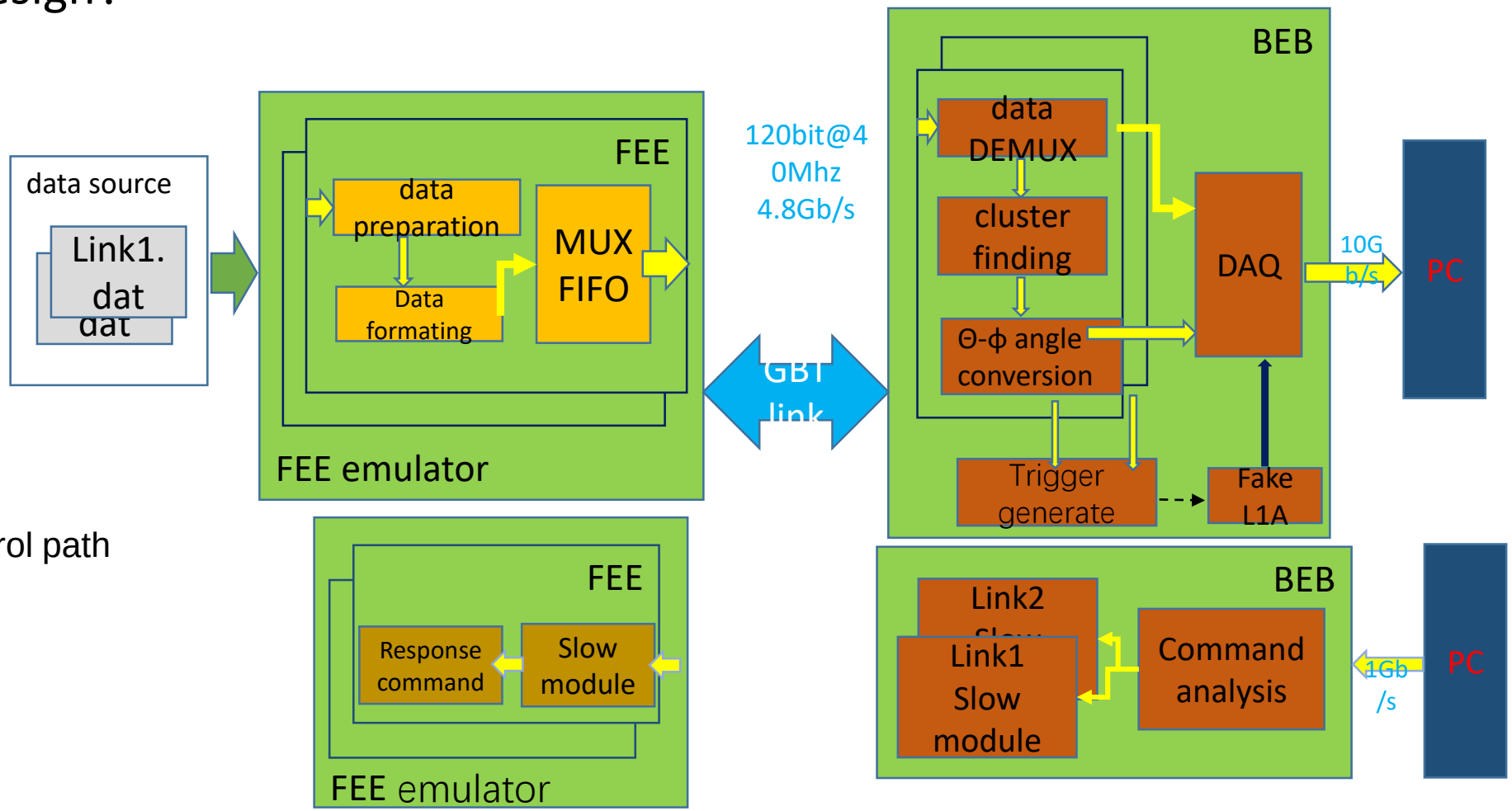
- Baselines
  - Fiber optics data links
  - Open structure to FEE
  - xTCA based architecture
    - High speed interconnection
    - High speed data throughput
  - commercial equipment based Post-DAQ (further discussion with LiFei)

R&D system Components



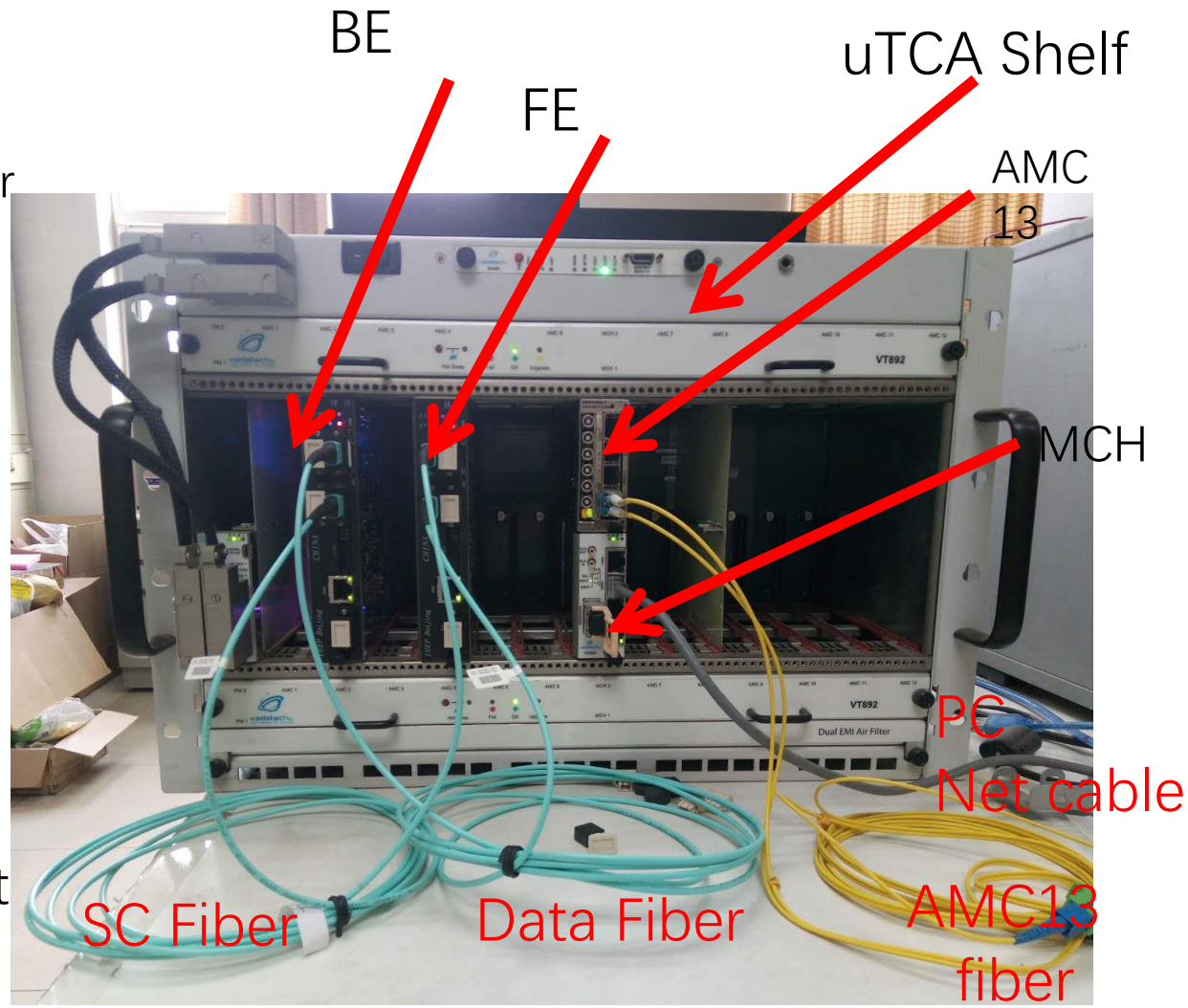
Firmware R&D block diagram

- Firmware design：
  - Data path
  - Slow control path



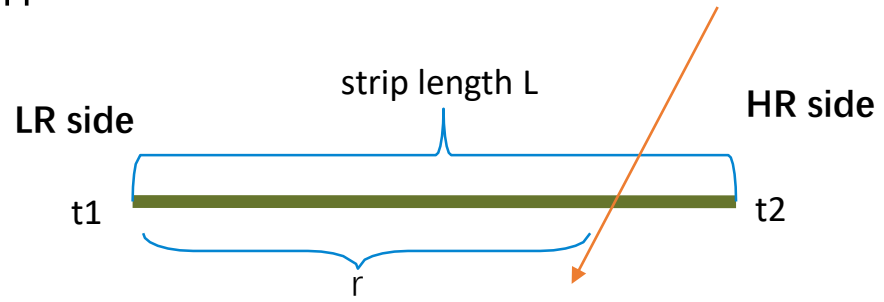
uTCA based R&D System

- Complete R&D System FE/BE
  - 2 ixFP cards
    - 1 ixFP cards for slow control/processor
    - 1 ixFP cards for iRPC data source
  - 1 AMC13
    - TTC/TTS
    - Management
  - 1 MCH + 1 PC
    - Slow control
    - Management + data storage
- Functions
  1. Emulation/Development System
  2. Hardware, GBT/GBT based
  3. Detector Simulator/data source
  4. Fast/Slow control, cluster finding, dat



Emulation Study for RPC detector

- iRPC provides better position by timing measurement from both strip ends
- Frontend Electronics board (FEB) emulator
  - Hit position
    - *r* is calculated by the time difference of signals from both Ends(see next page).
  - Digitization
    - For each fired strip, there are always 2 32-bit TDC data constructed.
      - Channel-HR Rising Edge + Channel-LR Rising Edge
  - Zero-suppression
    - Sends time info from only fired strips
  - TDC Data format(32 bits)
    - devAddr： FPGA ID
    - chanAddr : channel address
    - Coarse time: combine BCN(Preserved, 12 bits) and t1, t2
    - Fine time: responsible to the precision, 2.5ns/256≈10ps



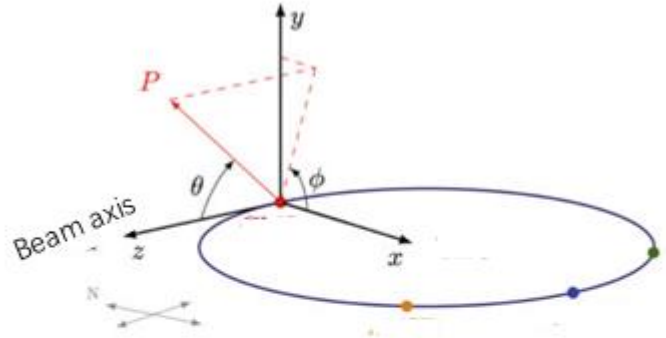
$$r = \frac{1}{2}\mathrm{L} - \frac{(\mathrm{t2} - \mathrm{t1})}{2} * v$$

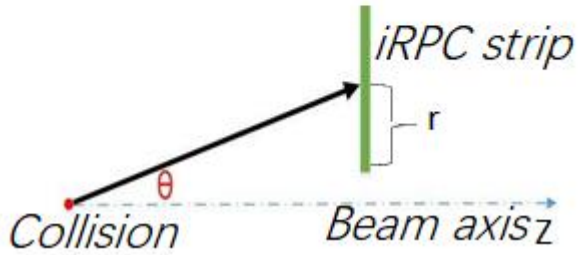
devAddr	chanAddr	TDC data	
		Coarse time	Fine time
2	6	16	8

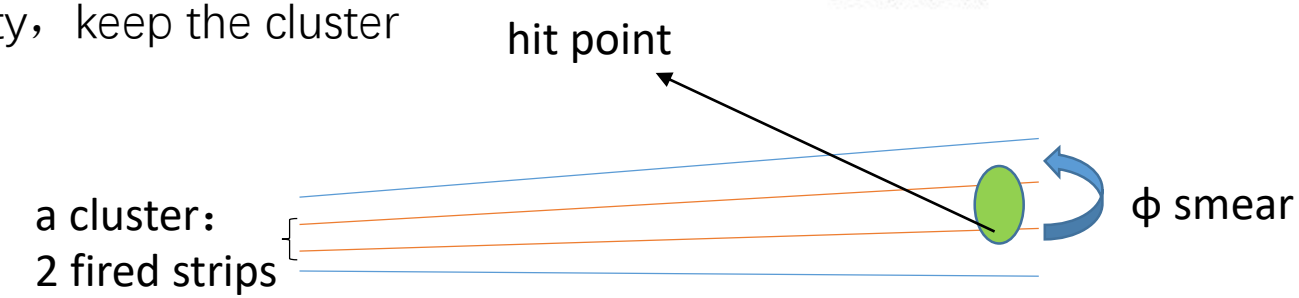
Transmit(Tx) Latency and Receive(Rx) Latency are introduced for MuX/DeMux

iRPC data source from simulation

- Data generation
  - Random hit point(r, φ) in a chamber
    - r: generates the hit point along strip.
    - φ: generates the strip number.
  - Smear of φ direction :
    - Generates a consecutive set of strips as a cluster.
  - Cluster size randomization :
    - 1-8,mean is 2.35
  - Number of clusters :
    - 1(75% probability),2(25% probability，keep the cluster size as 1)

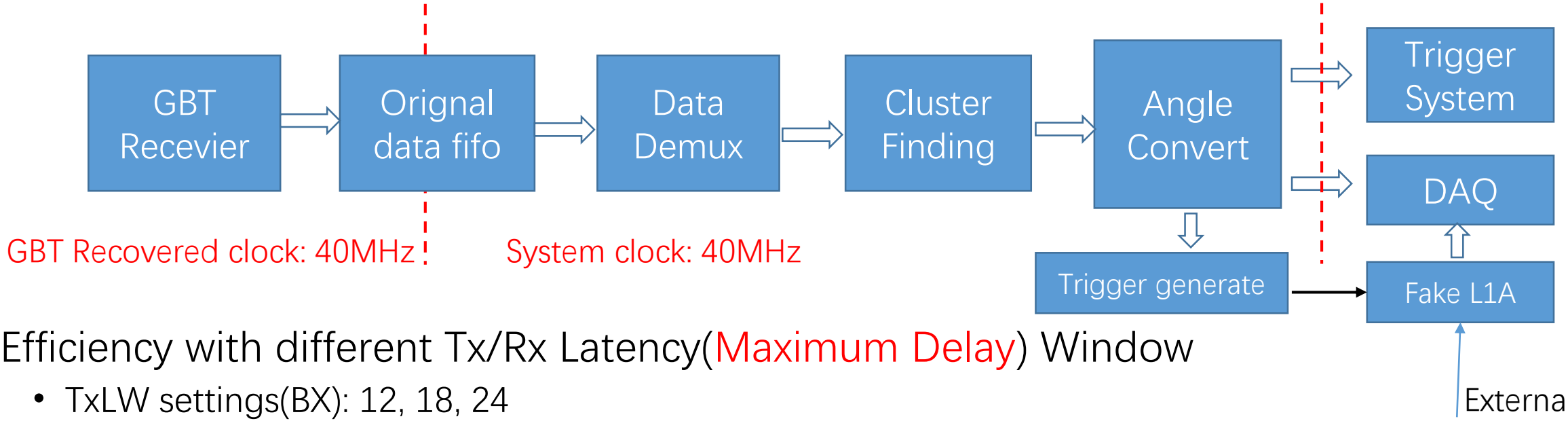






Data frame	Header (status flags)						TDC Data Payload								
Fied Name	MiscStatus	SCFifoFull	DataFifoFull	SCFrame	RSVD	DataValid	devAddr	chanAddr	TDC data	devAddr	chanAddr	TDC data	devAddr	chanAddr	TDC data
Field length	3	3	3	1	3	3	2	6	24	2	6	24	2	6	24
GBT Frame Group	G4						G3 & G2			G1 & G0			G6 & G5		

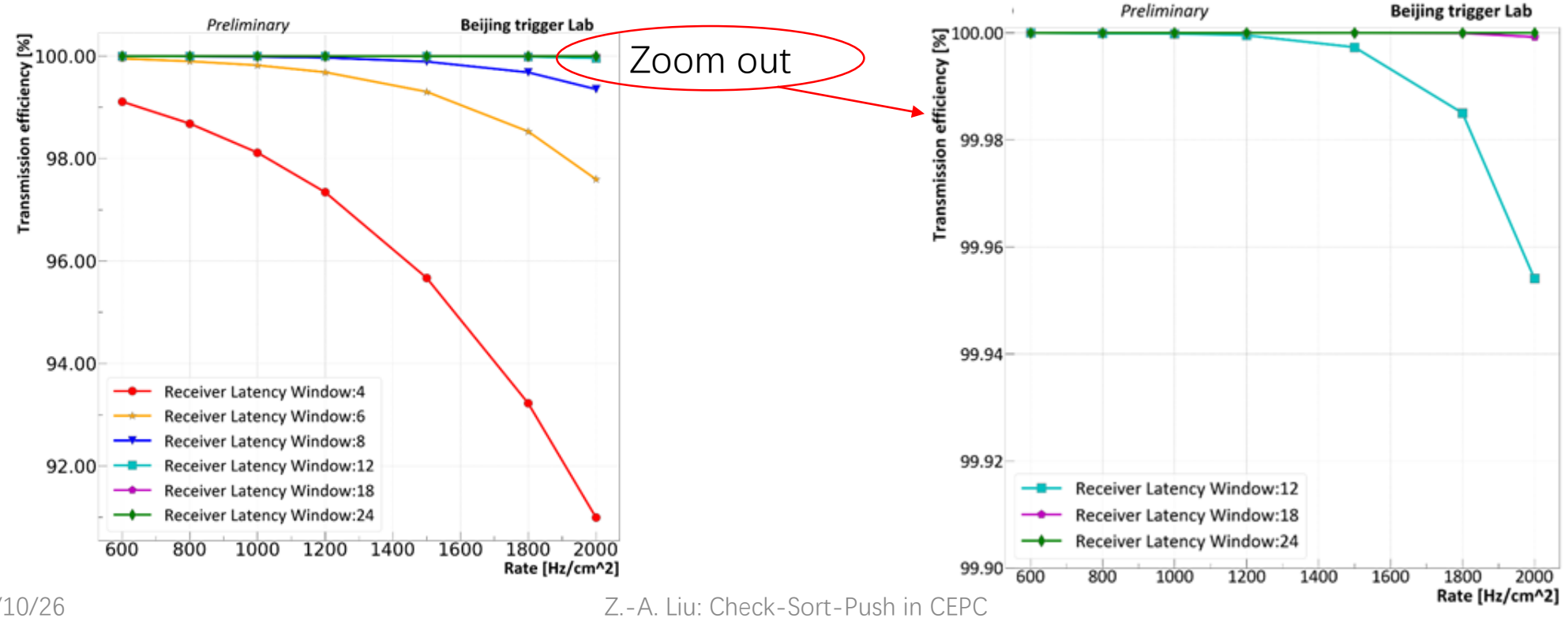
Latency Window optimization



- Efficiency with different Tx/Rx Latency(Maximum Delay) Window
  - TxLW settings(BX): 12, 18, 24
  - Average hit rate(Hz/cm^2 ,*safety factor 3*):600, 800, 1000, 1200, 1500, 1800, 2000
  - Mean cluster size:2.35 ; Surface of half chamber :6600cm^2 (1 fiber)
  - Size of the information/channel : 32bit (only rising edge)
    - For maximum rate 2000 Hz/cm^2,data rate = 2000_{rate}*6600_{surface}* $2.35_{\text{cluster size}}$ *$64_{\text{1 strip info}}$ Gb/s thus data rate is 1.99 Gb/s(smaller than GBT data bandwidth)
    - A larger TxLW leads to increased efficiency.
    - 100 % efficiency is measured with a TxLW setting of 24 BX in our simulation.

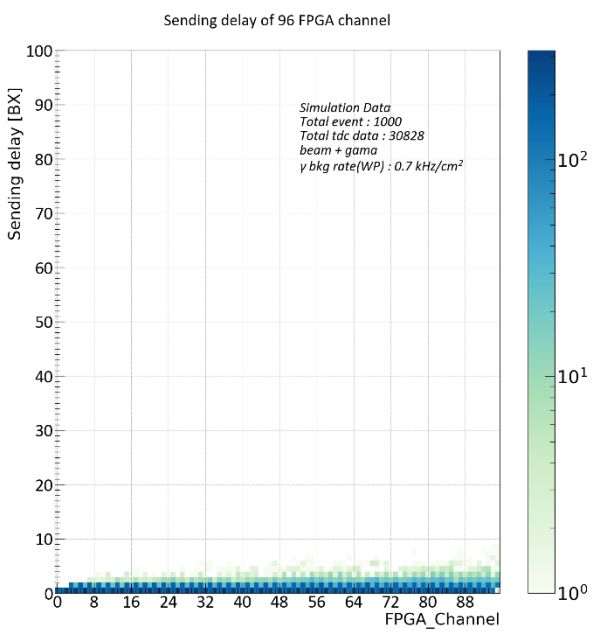
Emulation Study for high occupancy case

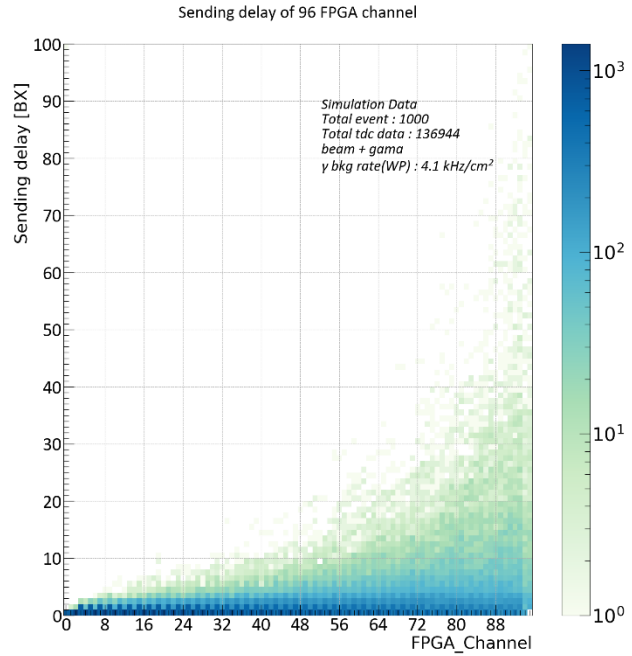
- Emulation Study results with real BEE Board (BEB)
  - FEE sends all data(TxLW: 24)
  - BEE introduce Receive Latency Window(RxLW)
    - The transmission window is calculated by comparing sending BX and the BX the data originates. At high occupancy cases when the data to be transmitted more than the transmission window should be rejected and sending a truncate flag to the backend.



Issues/Problems

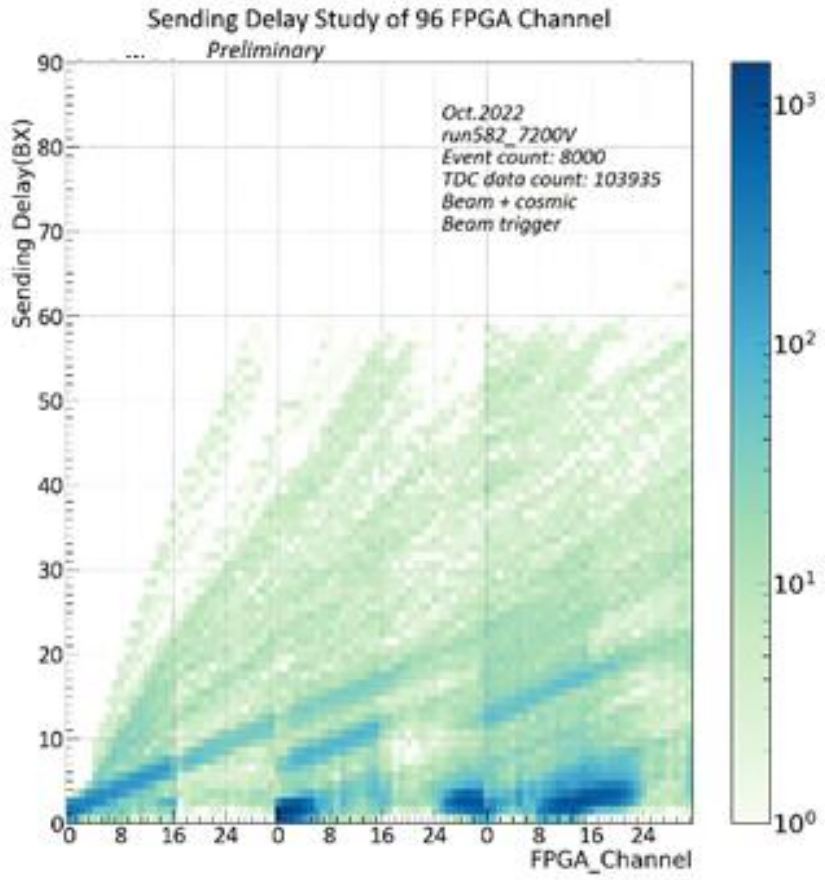
- As hit rate increase
  - Partial DAQ Data readout increase
    - Sequential readout/Frame readout
      - All data are read out which requires additional processing and compression
    - Polling/Sparse readout/Zero suppression
      - **External event sets a flag which is then checked by the readout circuitry**
  - Sending Delay increase in FEE
  - DeMux window in BEE should increase avoid loosing information for trigger also





Issues/Problems(2)

- Real test data showed more interesting for latency
  - Beam data + gamma background
  - Trigger with beam
    - Latency limited by the DeMux buffer length(64BX)
    - Gamma background have larger latency



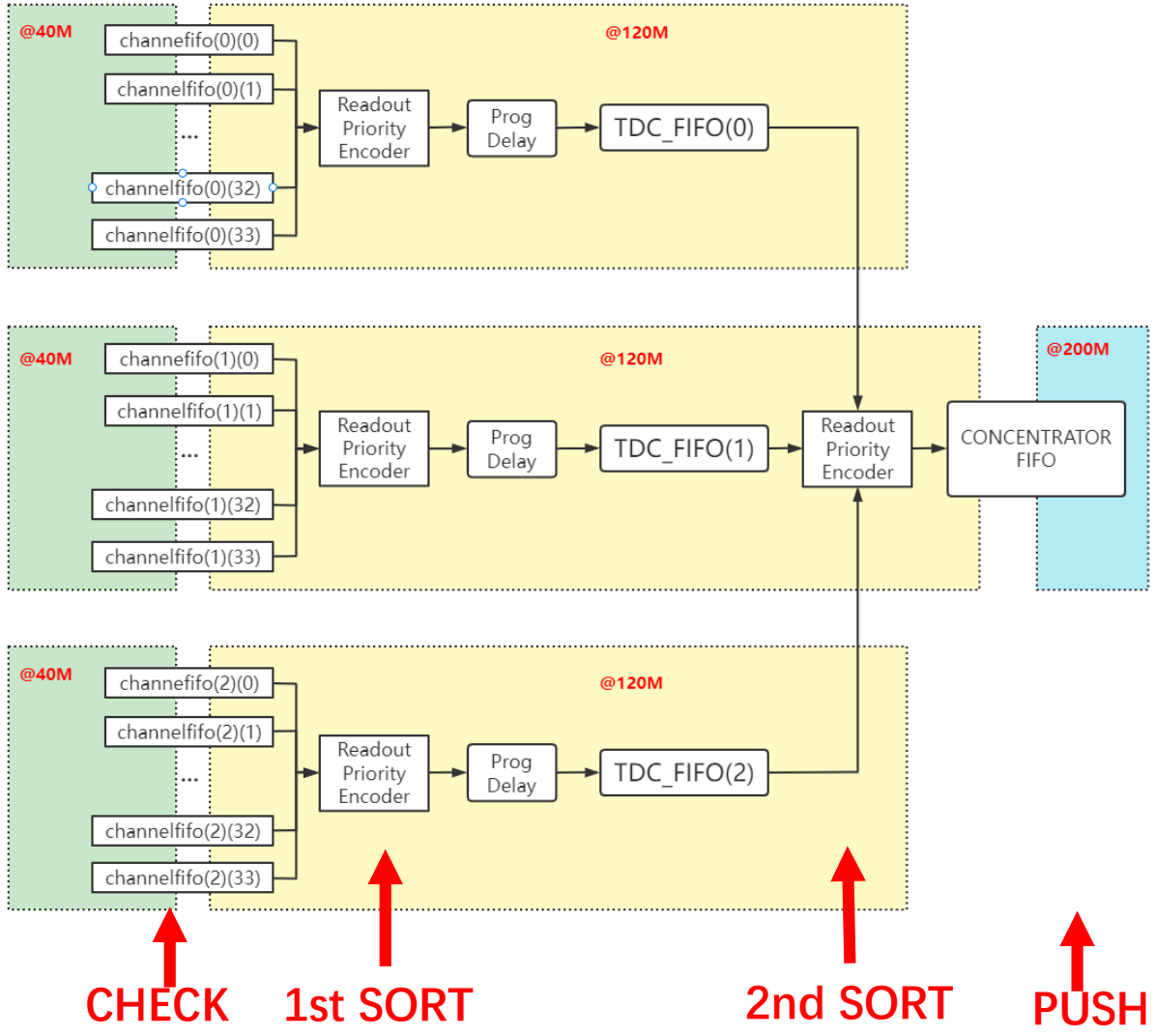
Conclusion: Sequential sending is problematic!
Sending must based on generation time

Check-Sort-Push(CSP) in FEE

- A new protocol ("Check-Sort-Push" algorithm based on timing) is introduced to use shorter Latency Window so to ease backend electronics DeMux design such a way that
  - Check(read) for new hits every BX(25ns) ,
  - Sort the new hits with existing ones in sequence of production in both the local data buffer and the merge buffer
  - Push(send) the concentration frame with earliest data in the merger buffer

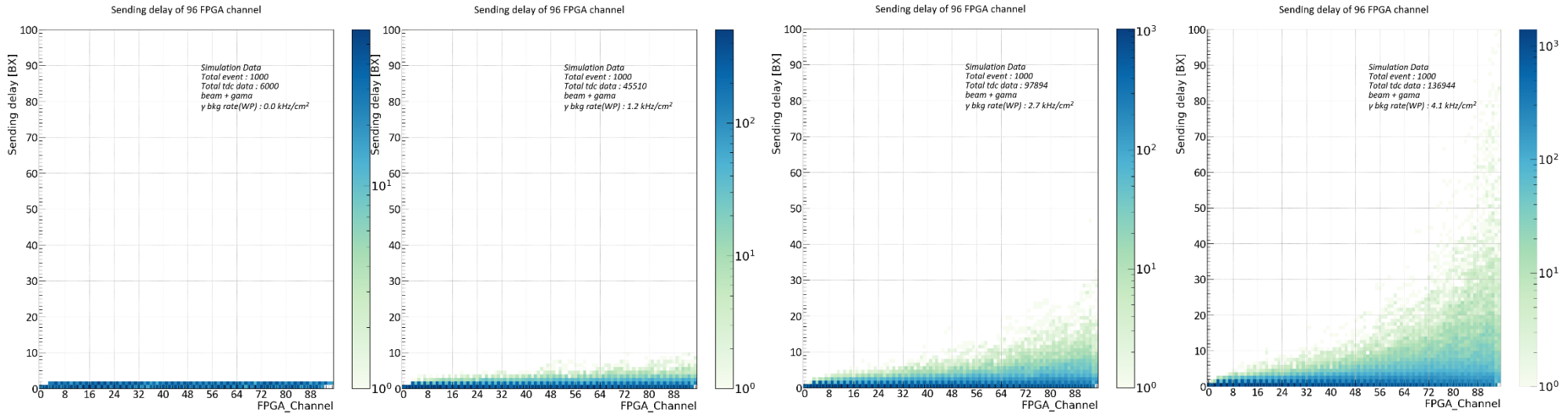
Check-Sort-Push implementation

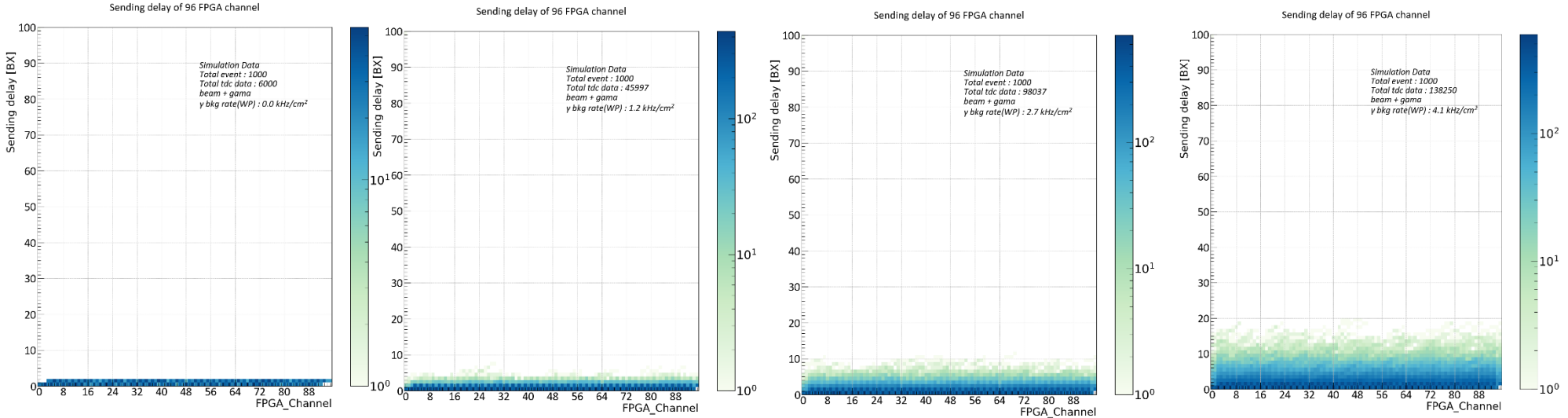
- Check
  - Sequential readout with digitizing clock
  - Data frame with generation time(in B) and channel plus data
- Sort(priority encoding) with merging clock(in shanred link)
  - First sorting by generation time befor moved to FIFO
  - Second sorting by generation time in the merger module before moved to concentrator FIFO
- Push
  - Sending long frame with concentration clock



Verification with simulation data

- Before CSP
  - 0-64 BX sending delay
- After CSP
  - 0-15 BX
  - Sending delay





Much decrease the DeMux resources

- Data De-multiplexing

Data flow

Delay=0	**RAM(0)(0)**	**RAM(0)(1)**		**RAM(0)(15)**
Delay=1	RAM(1)(0)	RAM(1)(1)		
⋮			⋱	
Delay=N-1	RAM(n-1)(0)	RAM(n-1)(1)		RAM(n-1)(n-1)

Demux_RAM_0
Demux_RAM_1
⋮
Demux_RAM_95

- Cluster Finding

Demux_RAM_0
Demux_RAM_1
⋮
Demux_RAM_95

only 1 fired strip
cluster_strip_1(central strip) = 1

only 2 consecutive fired strips
cluster_strip_2(central strip) = 1

⋮

only M consecutive fired strips
cluster_strip_M(central strip) = 1

OR

Angle Convert

find the central strip number of each cluster

get the ϕ values via Look-up table

calculate the r values based on time difference of 2 ends

Further Study

- CSP was proposed in CMS/RPC phase II upgrade
  - https://indico.cern.ch/event/967463/contributions/4071622/attachments/2126016/3579438/RPC_electronic_meeting_20201020.pdf
- Application has been made in CMS iRPC system with success
- Beam test data analysis is under going

Summary

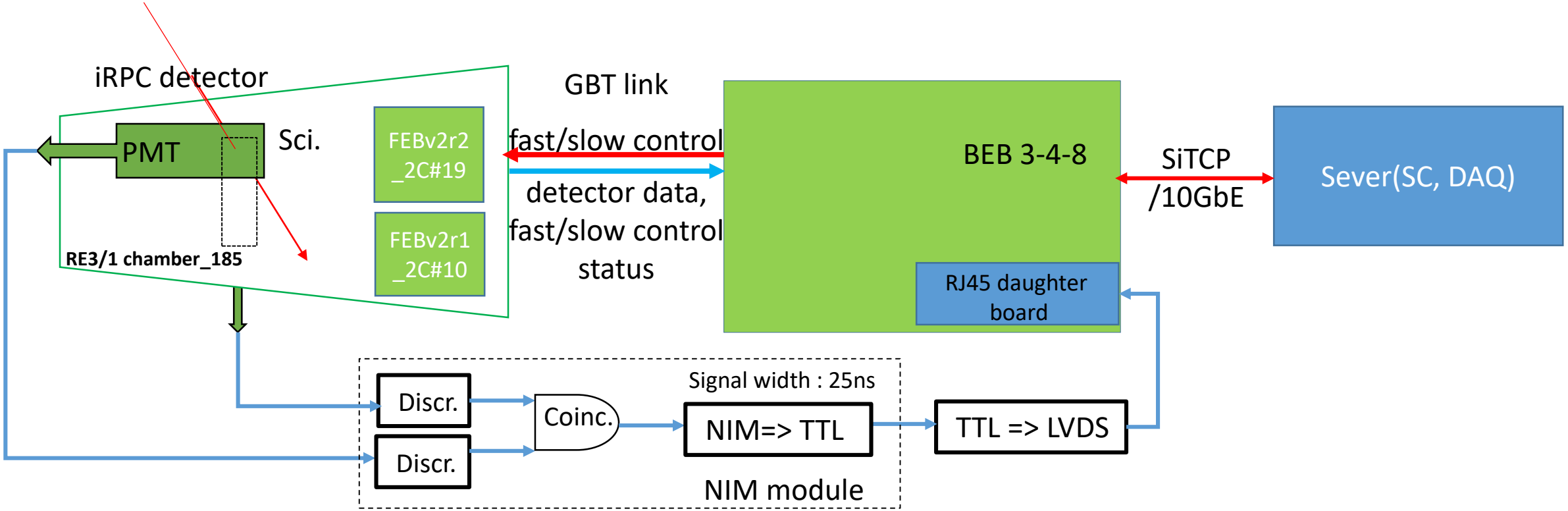
- Sequential data reading and sending introduce a large range of latency which leads to a partial data loss and FPGA resource demands
- Check-Sort-Push Protocol was introduced and proved effective in solving the issue with simulation data and emulation setup
- This R&D study could be a good start of CEPC TDAQ study

backups

backup

- Purpose
  - Studying the data transmission mechanism between the front and backend.
  - Verifying the backend functions(fast/slow control, data transmission mechanism and DAQ data format, etc.)

- System setup



Strange Data reception

Treceive	TDCdata1	TDCdata2	TDCdata3	converted_each_data_to_strip_and_time(ns)		
BX-4	464002d3	454002f2	463ff459	strip_22_HR :40967.1	strip_21_HR :40967.4	strip_22_HR :40930.9
BX-3	474002c8	483ff1f2	473ff29c	strip_23_HR :40967.0	strip_24_HR :40924.9	strip_23_HR :40926.5
BX-2	4b3ff25d	4a3ff245	493ff22e	strip_27_HR :40925.9	strip_26_HR :40925.7	strip_25_HR :40925.4
BX-1	4e3ff426	4d3ff361	4c3ff289	strip_30_HR :40930.4	strip_29_HR :40928.4	strip_28_HR :40926.3
BX+0	003ff73d	4e4002b8	4d4002b0	strip_32_HR :40938.1	strip_30_HR :40966.8	strip_29_HR :40966.7
BX+1	033ff71b	023ff744	013ff74a	strip_35_HR :40937.8	strip_34_HR :40938.2	strip_33_HR :40938.2
BX+2	034002e2	0240030d	0140030e	strip_35_HR :40967.2	strip_34_HR :40967.6	strip_33_HR :40967.6
BX+3	043ff749	01400ed2	00400eae	strip_36_HR :40938.2	strip_33_HR :40997.1	strip_32_HR :40996.7
BX+4	0440030d	053ff767	4f3ff474	strip_36_HR :40967.6	strip_37_HR :40938.5	strip_31_HR :40931.1
BX+5	073ff786	063ff767	054003eb	strip_39_HR :40938.8	strip_38_HR :40938.5	strip_37_HR :40969.8
BX+6	083ff78e	074003b6	0640031d	strip_40_HR :40938.9	strip_39_HR :40969.3	strip_38_HR :40967.8
BX+7	094003c0	084003b7	093ff79f	strip_41_HR :40969.4	strip_40_HR :40969.3	strip_41_HR :40939.1
BX+8	0a4003c7	0b3ff7b1	0a3ff7b4	strip_42_HR :40969.4	strip_43_HR :40939.2	strip_42_HR :40939.3
BX+9	0d3ff7be	0c3ff7b3	0b40036c	strip_45_HR :40939.4	strip_44_HR :40939.2	strip_43_HR :40968.6
BX+10	0e3ff7b3	0d4003a6	0c400380	strip_46_HR :40939.2	strip_45_HR :40969.1	strip_44_HR :40968.8
				----------BEB ±10BX window cut----------		
BX+11	0f4004d1	0e4003d6	0f3ff8c2	strip_47_HR :40972.0	strip_46_HR :40969.6	strip_47_HR :40941.9
BX+12	123ffb14	113ffb26	103ffc29	strip_45_LR :40947.7	strip_46_LR :40947.9	strip_47_LR :40950.4
BX+13	143ffafa	133ffb07	114006d0	strip_43_LR :40947.4	strip_44_LR :40947.6	strip_46_LR :40977.0
BX+14	173ffaa6	163ffa30	153ffaf8	strip_40_LR :40946.6	strip_41_LR :40945.5	strip_42_LR :40947.4
BX+15	1a3ffa7c	193ffc4d	183ffa9b	strip_37_LR :40946.2	strip_38_LR :40950.8	strip_39_LR :40946.5
BX+16	1d3ffc25	1c3ffc37	1b3ffae3	strip_34_LR :40950.4	strip_35_LR :40950.5	strip_36_LR : 40947.2
BX+17	503ff17c	1f3ffc15	1e3ffc32	strip_31_LR :40923.7	strip_32_LR :40950.2	strip_33_LR :40950.5
BX+18	523ff067	513ff07b	4f4002e3	strip_29_LR :40921.0	strip_30_LR :40921.2	strip_16_LR :40967.2
BX+19	533ff028	523ffe8d	5140079c	strip_28_LR :40920.4	strip_29_LR :40956.4	strip_30_LR :40979.0
BX+20	563fefb4	553fefb1	543fefc3	strip_25_LR :40919.3	strip_26_LR :40919.2	strip_27_LR :40919.4
BX+21	593ff044	583ff004	573fefdb	strip_22_LR :40920.7	strip_23_LR :40920.0	strip_24_LR :40919.6
BX+22	5a3ffe73	5b3ff048	5a3ff049	strip_21_LR :40956.1	strip_20_LR : 40920.7	strip_21_LR :40920.7

firstly

finally

I.FPGA1

II.FPGA0

III.FPGA1

32bit received GBT frame from FEB in time seq.

- Problem: Data produced in same time in different channel are transmitted with unexpected delay.
- For example Strip 23 HR was transmitted at BX-2 but LR at BX+21.

- Present FEB sending algorithm:
- Data with smaller FPGA ID and channel number are transmitted firstly.

Eg. 2bit FPGA ID+6bit channel

I. “4x” FPGA 1

II. “0x-1x” FPGA 0

III. “5x” FPGA 1

Channel number is used in each sector I/II/III.

2022/9/25