Backend electronics function emulation and realization of iRPC

Zhen-An LIU/Jingzhou Zhao/Hanjun KOU*
/Jianing Song/Jia Tao/Qingfeng Hou
Institute of High Energy Physics, CAS

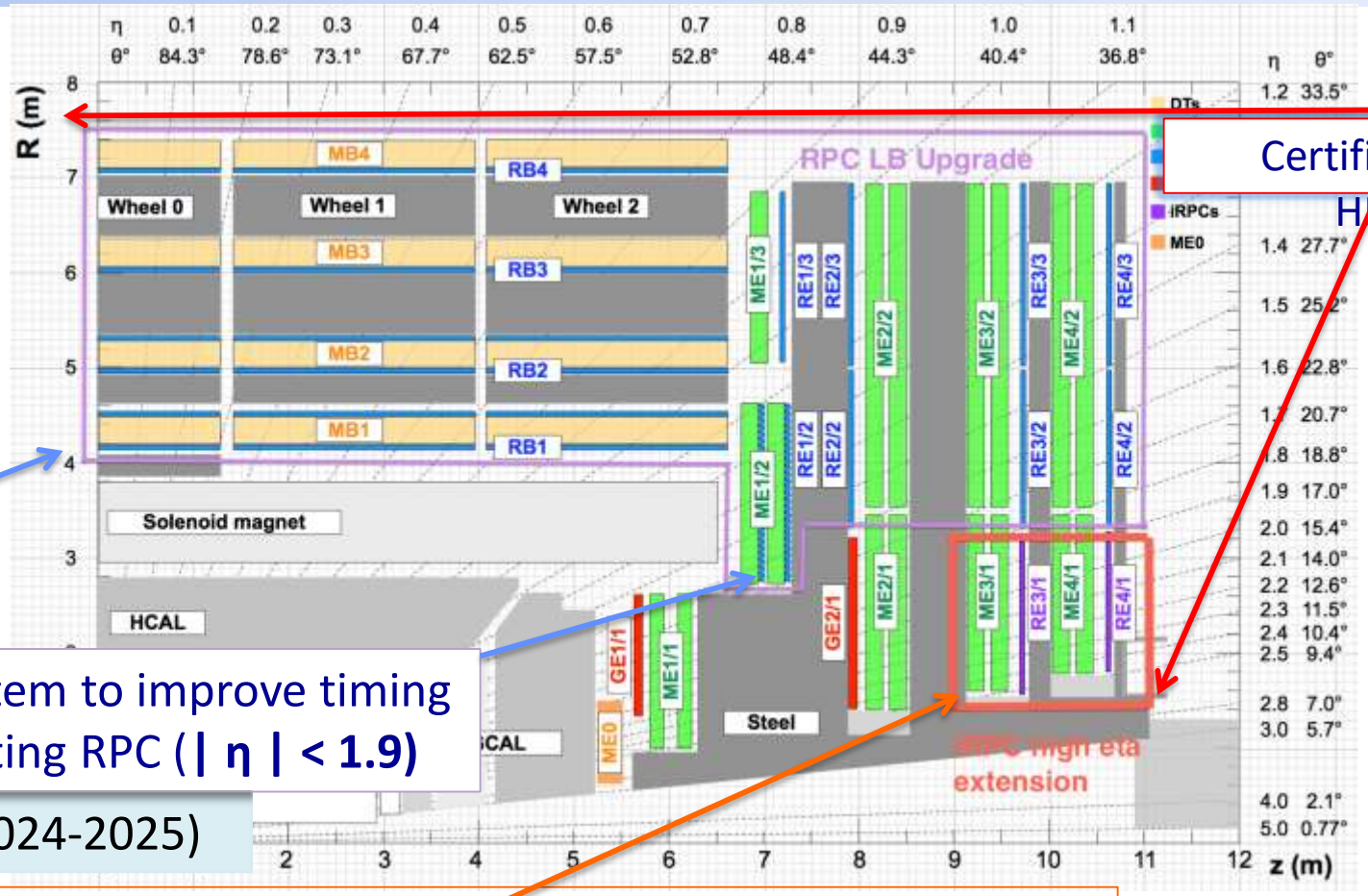
The 6th China LHC Physics Workshop (CLHCP2020)
Nov. 7 2020

Outline

- Introduction to iRPC backend electronics system
- IRPC data source simulation
- FEE function emulation
- BEE design and implementation
- Summary

Phase 2 RPC Upgrade Project Overview

- RE3.1/4.1 (iRPC + FEBs)
- New Link System for present RPC
- New RPC Backend Electronics (3.1/4.1, RPC, TDAQ)



Certification for HL_LHC

Upgrade of Link System to improve timing resolution for existing RPC (**| η | < 1.9**)

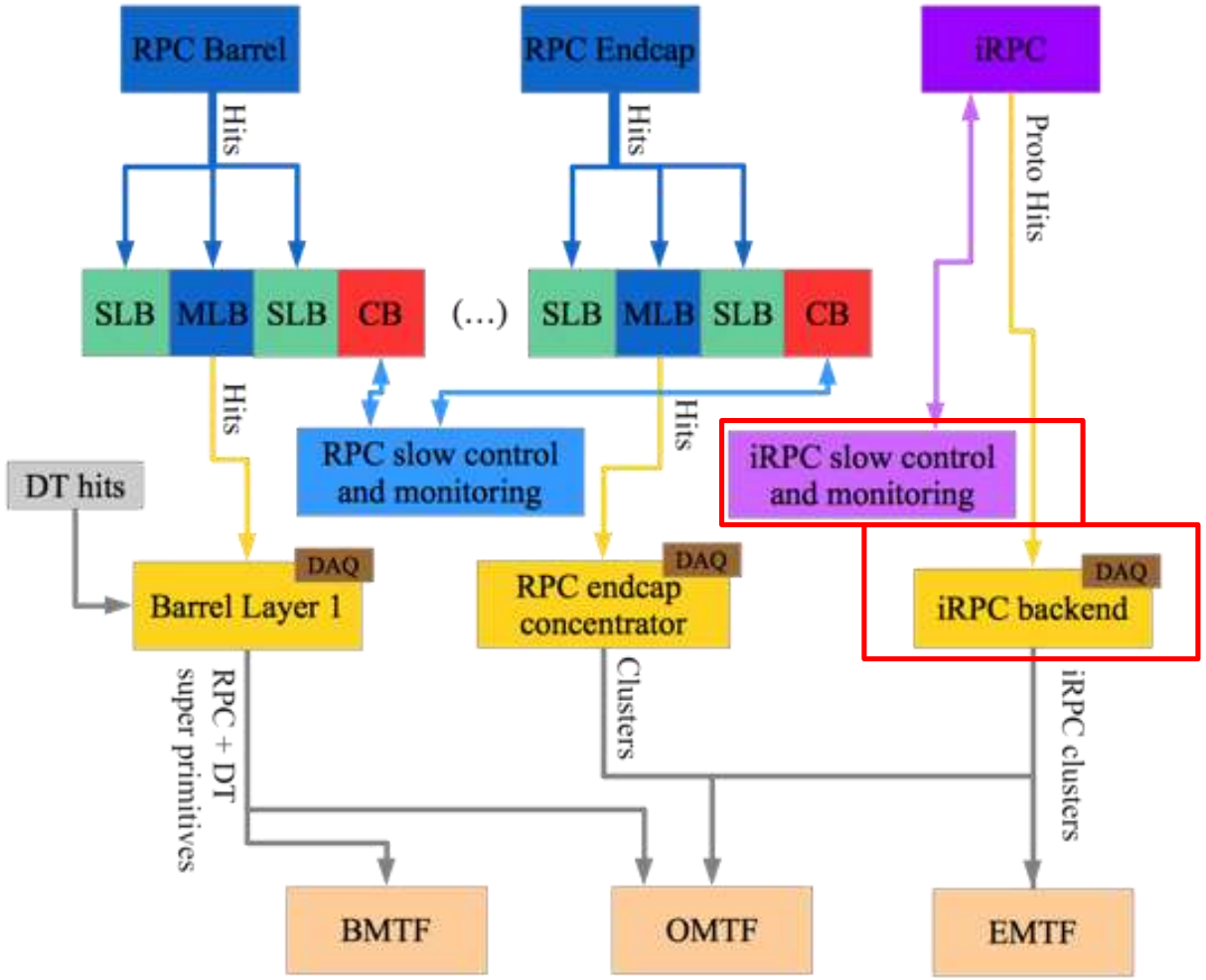
Installation in LS3 (2024-2025)

Extend the RPC coverage up to |η | = 2.4 to increase redundancy in high eta region in stations 3 and 4

Installation in YETS 2021-22 & YETS 2022-23

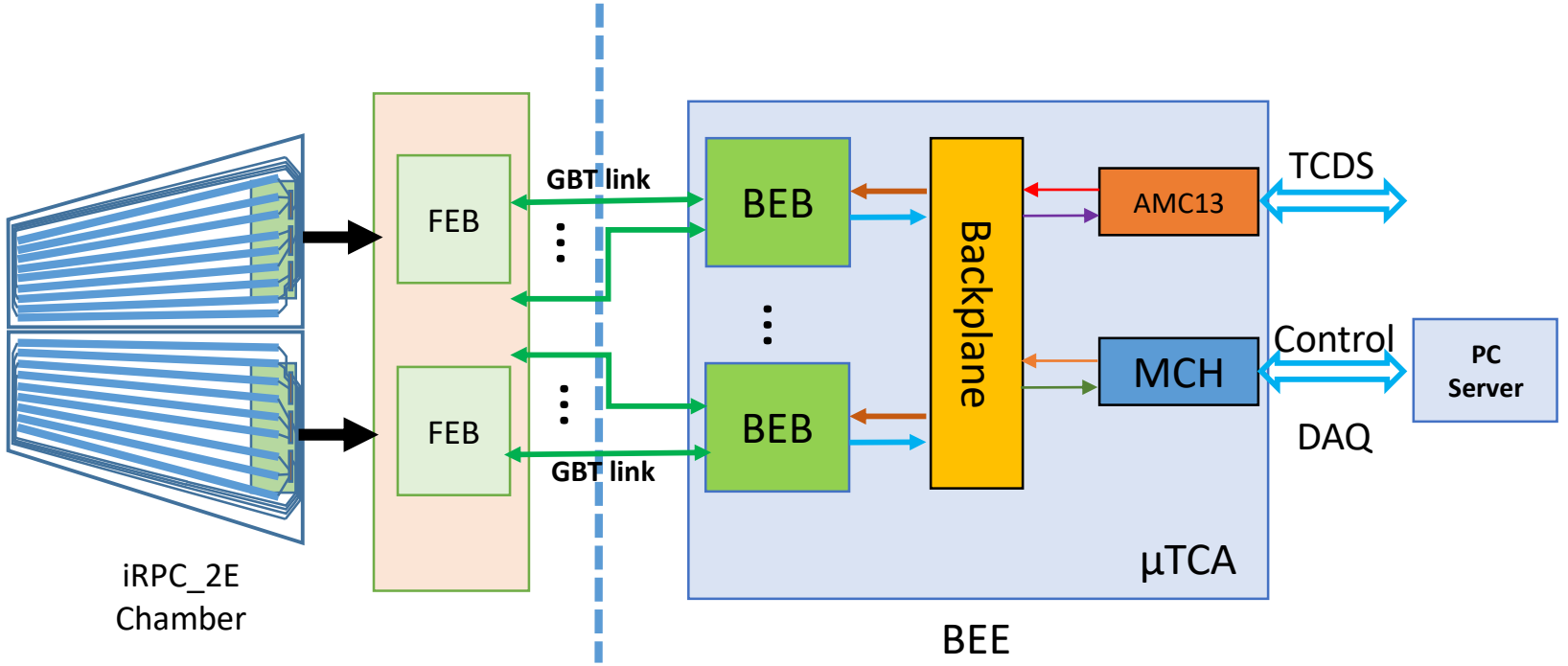
Tasks for IRPC Backend Electronics

- Fast and Slow Control for FEE via TCDS and MCH by GBT link
- Data via GBT link
- Monitoring to central DAQ via MCH/DTH
- Primary trigger algorithm



IRPC Backend electronics system

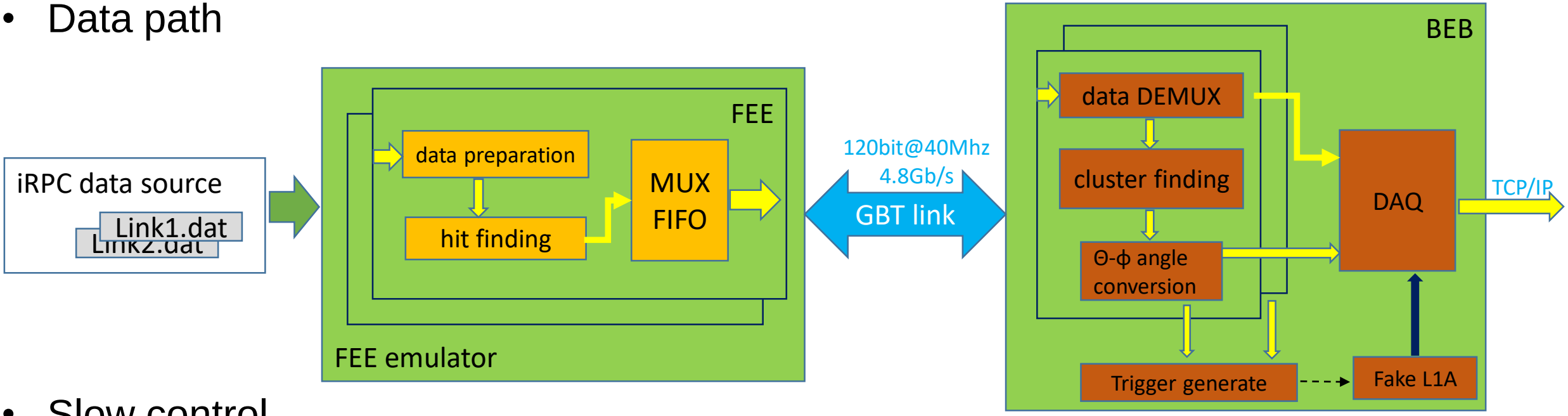
- System design proposal ：
  - μTCA compliant BEB,
    -core board
  - a μTCA crate,
  - an AMC13 card,
    -system clock and fast control
  - a μTCA Carrier Hub(MCH),
    -manage the whole system
  - a sever PC.
    - slow control and DAQ



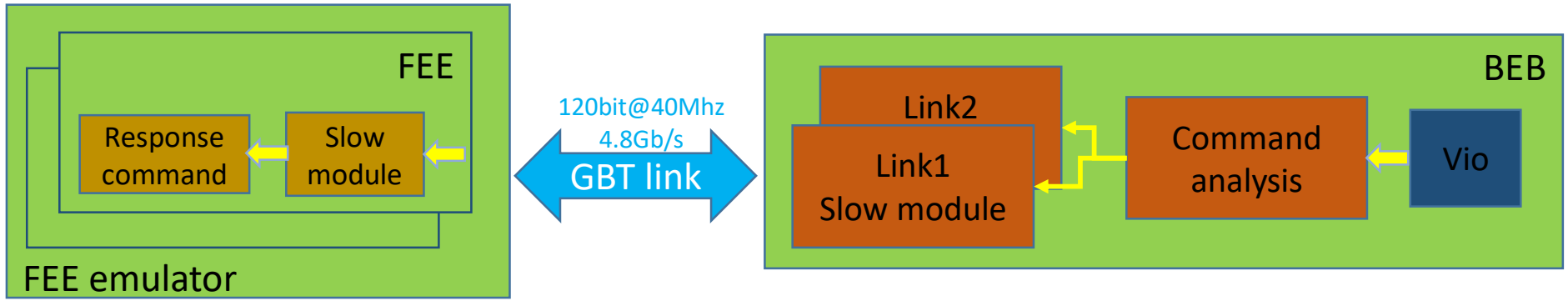
- System emulation ：
  - iRPC data source simulation
  - FEE function emulation
  - BEB design and trigger algorithm implemention

IRPC FEE-BEE emulation system

- Data path

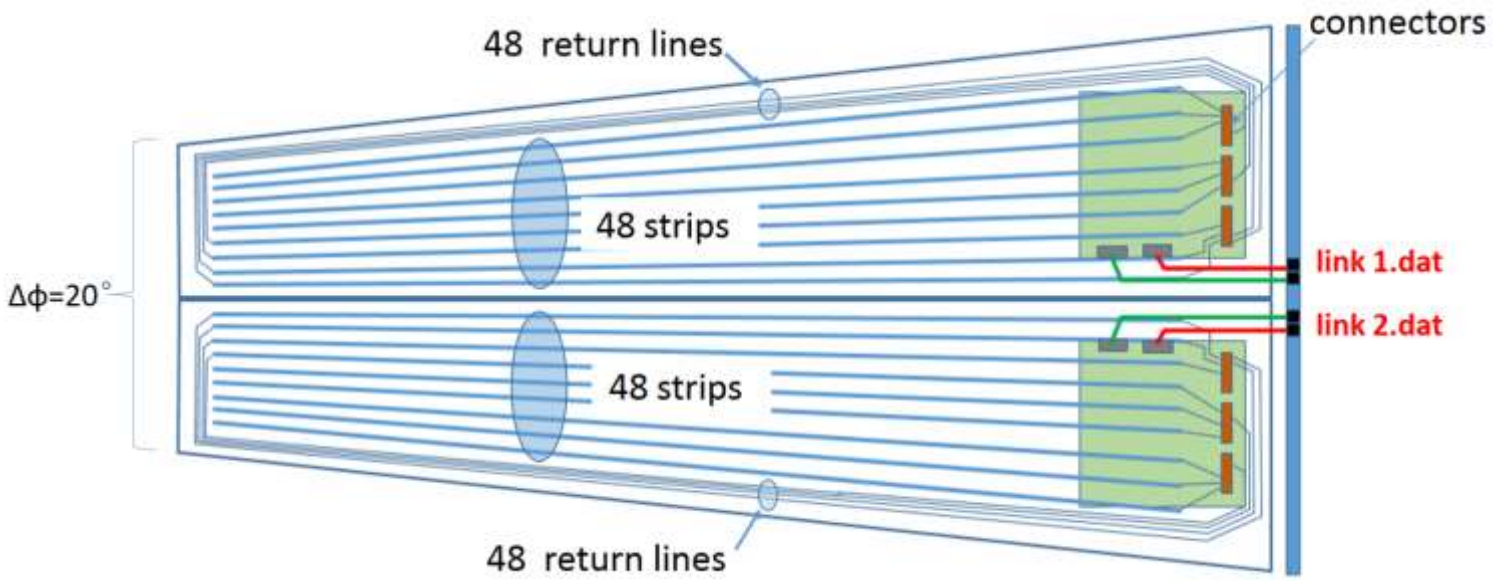


- Slow control



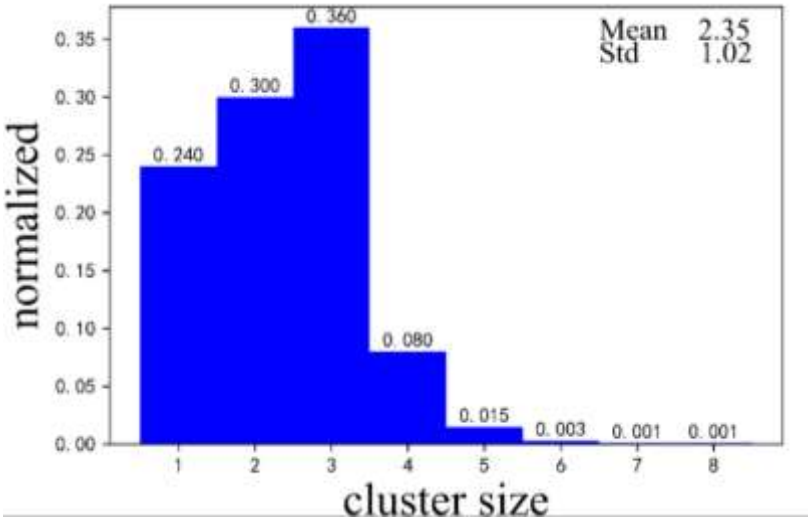
IRPC data source simulation

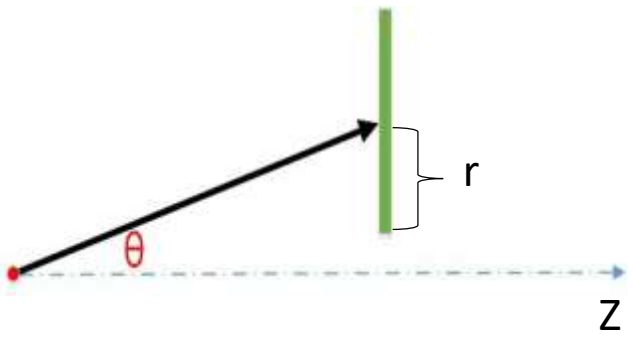
- iRPC data source generation
  - Parameters setting
    - RE3/1 iRPC Chamber and strips
      - For one RE3/1 chamber,R_{in}=153cm,R_{out}=319cm,Area=1.32m^2,covers 20° contains 96 strips，read out by 2 links.
      - For one strip ,$length$=166cm, pitch about 0.60cm at low radius(1.23cm at high radius) connect with a return line (kept as 244cm) within PCB

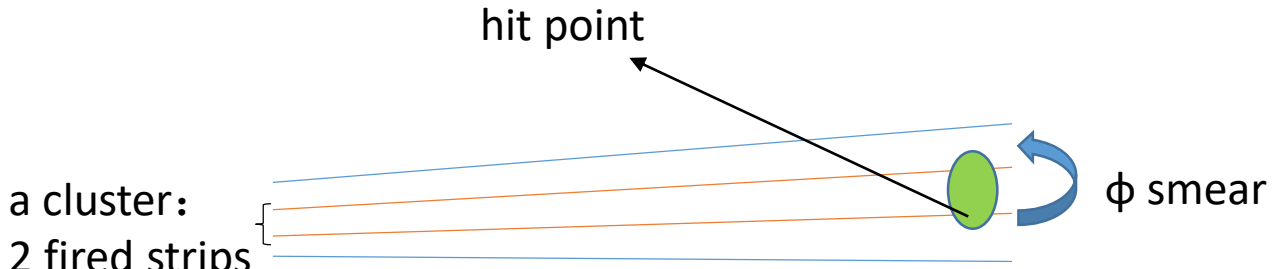


IRPC data source simulation

- 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
    - *reference : Shchablo Konstantin[1] on behalf of the CMS Muon group, Front-End electronics for CMS iRPC detectors（RPC2020, Roma）*
  - Number of clusters :
    - 1(75% probability),2(25% probability， keep the cluster size as 1)





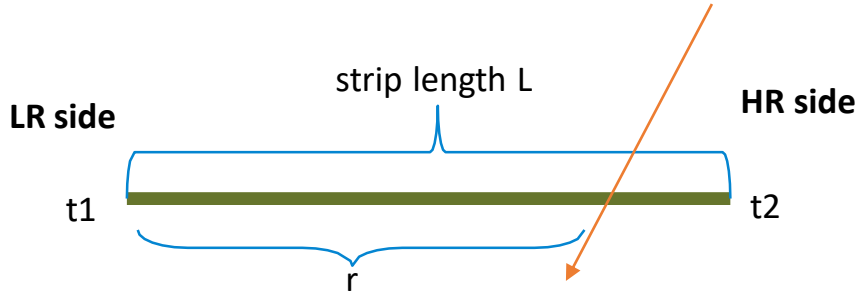


FEE function emulation

- FEE software algorithm
  - Hit position
    - *r* is calculated by the time difference of signals from both ends.
  - Data format

Edge	Asic	Channel	Coarse		Fine
			Preserved	Time	
1	2	5	12	4	8

    - Edge: Rising/Falling edge of the signal
    - Asic: Totally 3 TDCs on FEB
    - Channel: 32 channels for each TDC
    - Coarse: Coarse time, combine BCN(Preserved)and t1, t2
    - Fine: Fine time, responsible to the precision, 2.5ns/256≈10ps
  - Digitization
    - For each fired strip, there are always 2 constraint 32-bits data constructed.
      - Channel-HR Rising Edge + Channel-LR Rising Edge

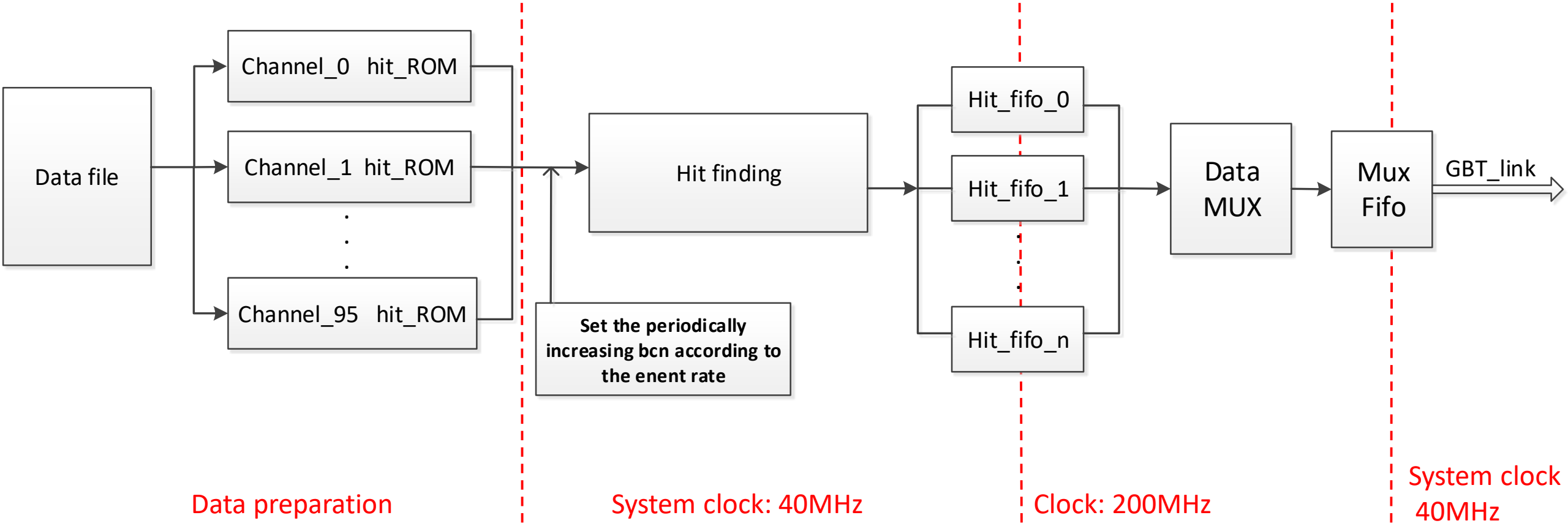


$$r = \frac{1}{2}\mathrm{L} - \frac{(\mathrm{t2} - \mathrm{t1})}{2} * v$$

- Algorithm:
  - Mean cluster size : 2.35
  - 2*32-bit data needed for 1-strip information
  - 1 GBT frames to cover 1 strip

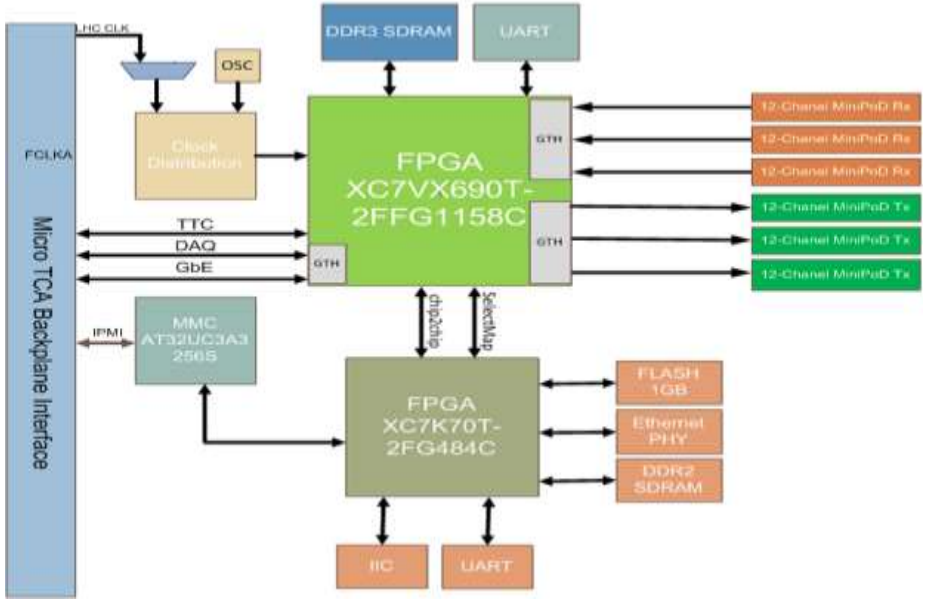
FEE function emulation

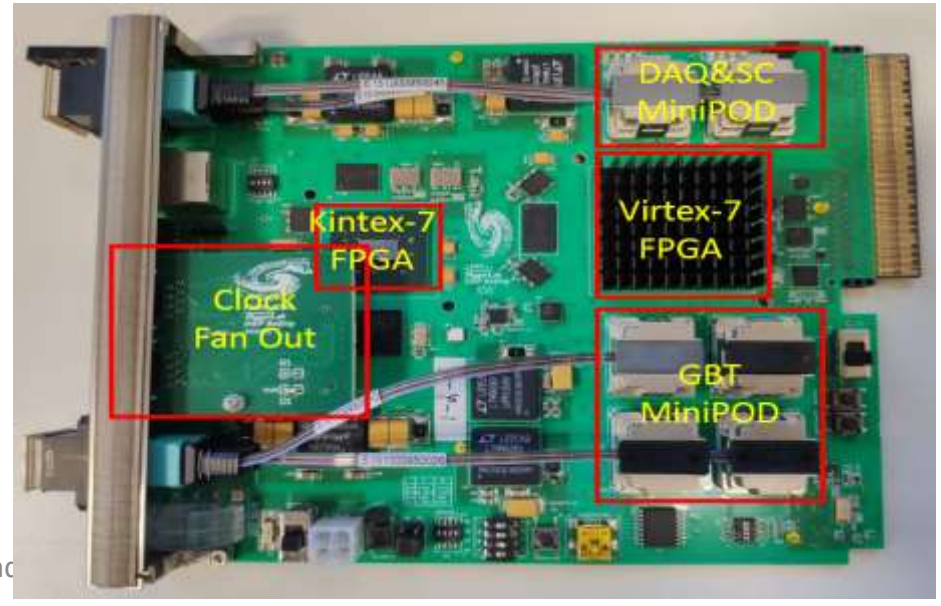
- FEE firmware algorithm



BEB Hardware design

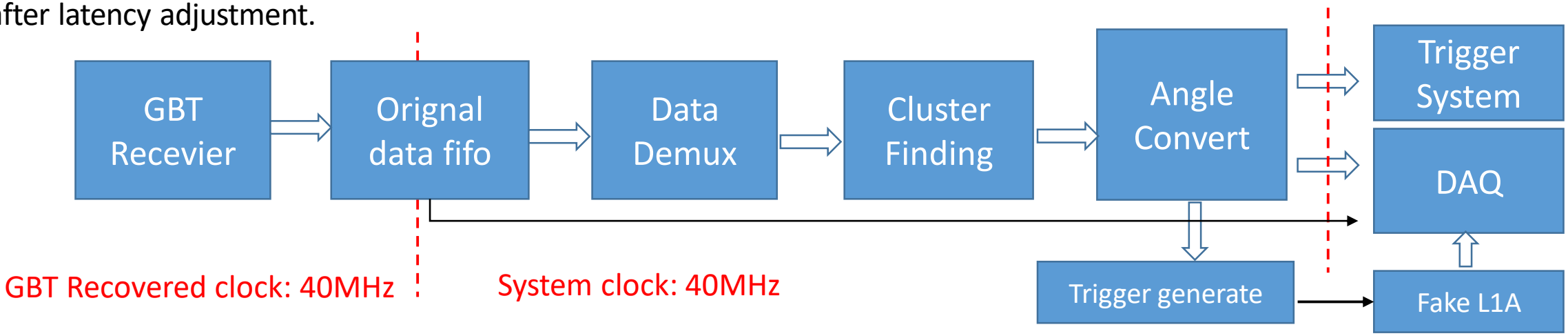
- BackEnd Electronics Board
  - full-height double-width µTCA standard BEB
  - 36 links input/output, a total bandwidth of 400 Gb/s per board
- XC7VX690T-2(Virtex-7)
  - Core FPGA, data processing
- XC7K70T-2 (Kintex-7)
  - Ctrl FPGA, system controlling and clock management
- AT32UC3A1512(Atmel)
  - Intelligent controller that manages the module and is interfaced to the Carrier
- Rj45
  - Clock fan out





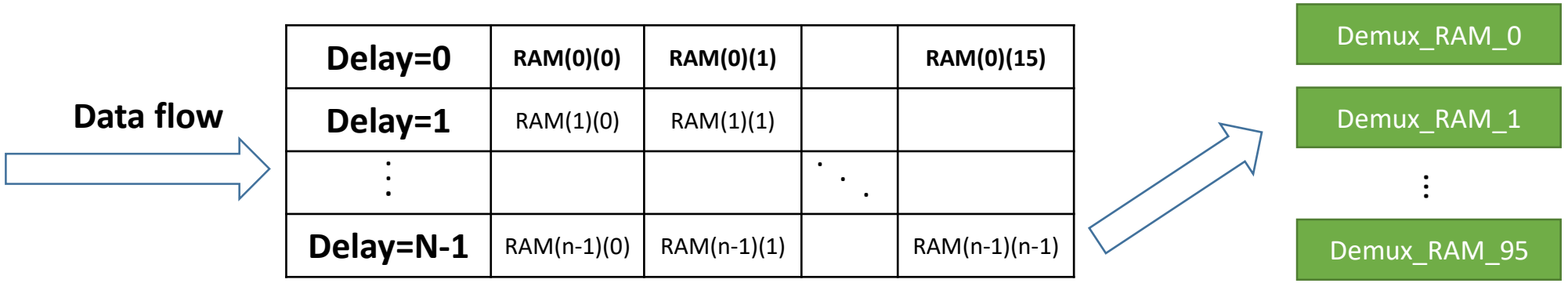
BEE firmware algorithm

- Key point of the algorithm
  - Use pipeline method
  - data decompression
  - clusters whose size is beyond 4 are rejected
  - Maximum number of clusters allowed is 2
  - A fake L1A is generated based on angle conversion results and then feedback to DAQ module
  - DAQ module packs and uploads the original data and angle value based on L1A arriving time after latency adjustment.



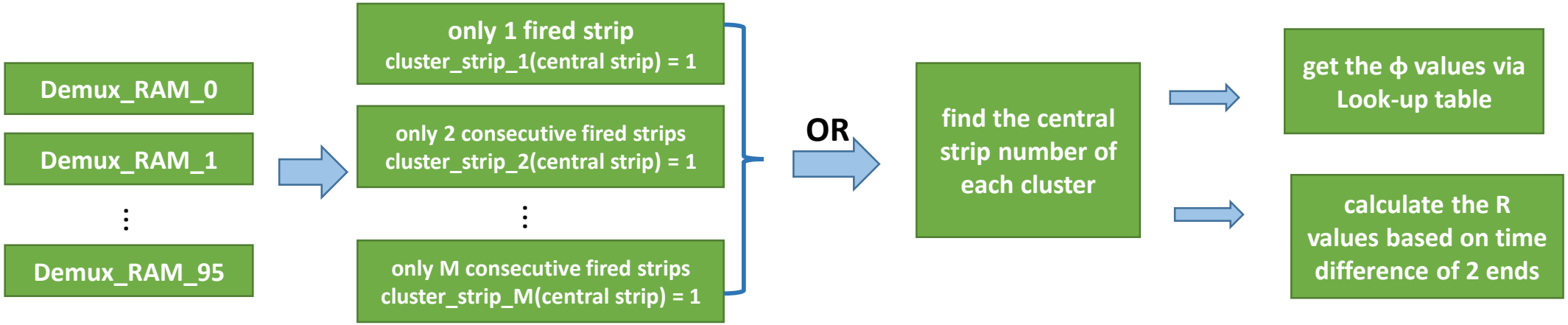
BEE firmware algorithm

- Data Demux



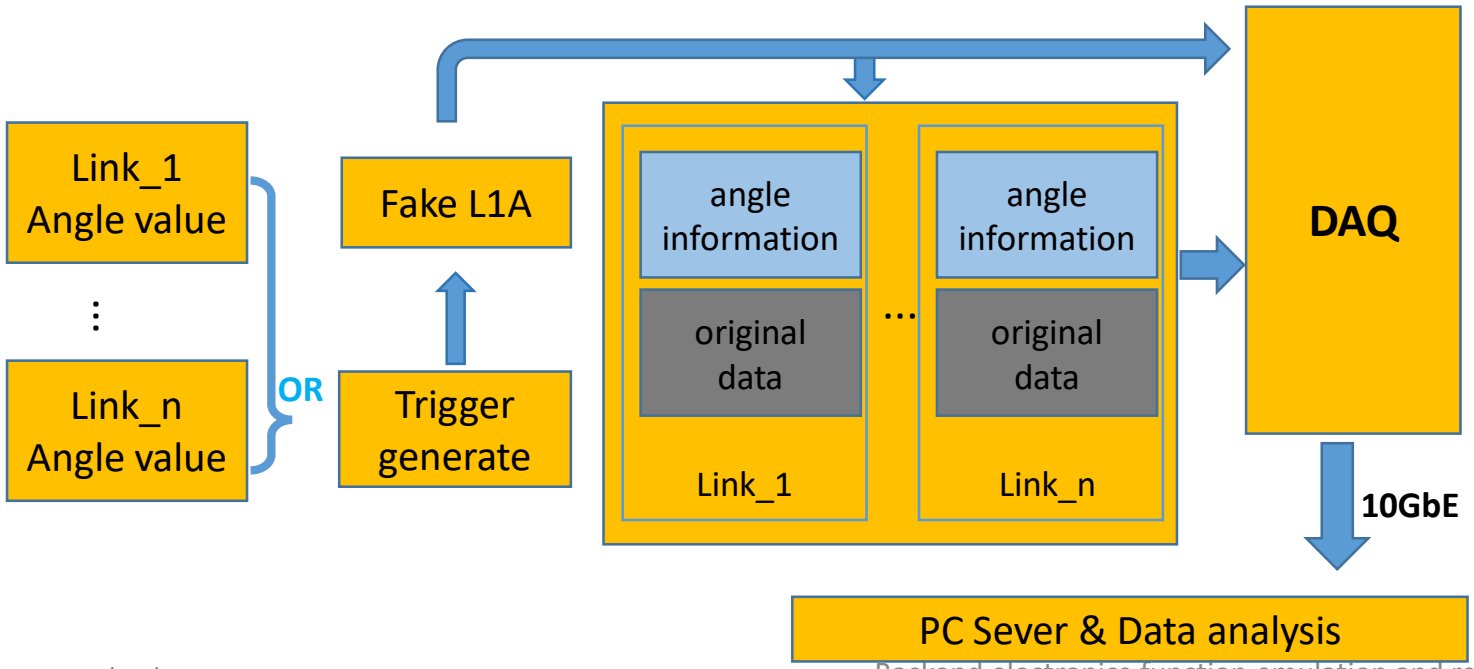
- Cluster Finding

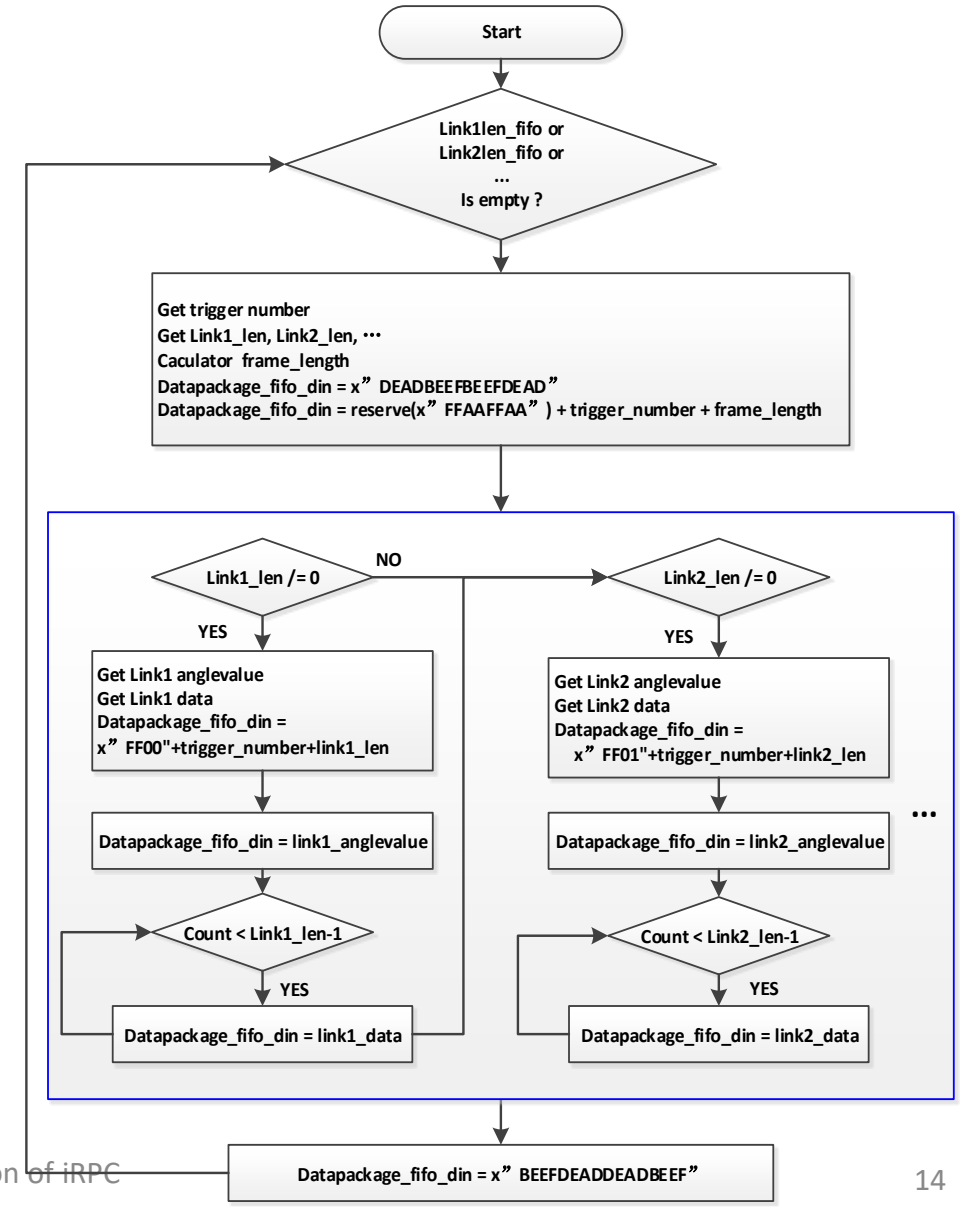
Angle Convert



BEE firmware algorithm

- BEE-DAQ algorithm
  - Trigger generate： use OR logic based on valid angle information from links.
  - Fake L1A signal is generated and then feedback to DAQ module after algorithm processing.
  - When L1A arrives, DAQ module packs and uploads time-relevant data after latency adjustment





BEE-DAQ data format

- Current CPPF system DAQ data format：

CPPF Payload	Header	MP7 header0
		MP7 header1
		MP7 header2
	Data Payload	RX_buf[0] Frame
		TX_buf[0] Frame
		RX_buf[1] Frame
		TX_buf[1] Frame
		RX_buf[2] Frame
		TX_buf[2] Frame
		TX_buf[3] Frame
		TX_buf[4] Frame
		TX_buf[5] Frame
		TX_buf[6] Frame
		TX_buf[7] Frame
		TX_buf[8] Frame
	Trailer	MP7 Trailer

- Used as a reference for iRPC BEE DAQ currently.
- Some parts need to be adjusted based on iRPC DAQ requirements.
- Some fields such as “OrbNum” was unused in lab test at present.

	Byte 7								Byte 6								Byte 5								Byte 4							
	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
MP7 header0	Reserved				nAMC				L1AID																bctr							
MP7 header1	Reserved																Evtp								OrbNum							
MP7 header2	ALGO_REV																								Reserved							
Trailer	CRC_32																								L1AID							

	Byte 3								Byte 2								Byte 1								Byte 0							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MP7 header0	Event length																															
MP7 header1	BoardID																															
MP7 header2	FW_REV																															
Trailer	Event Length																															

BEE-DAQ data format

- Current iRPC BEE DAQ data format:

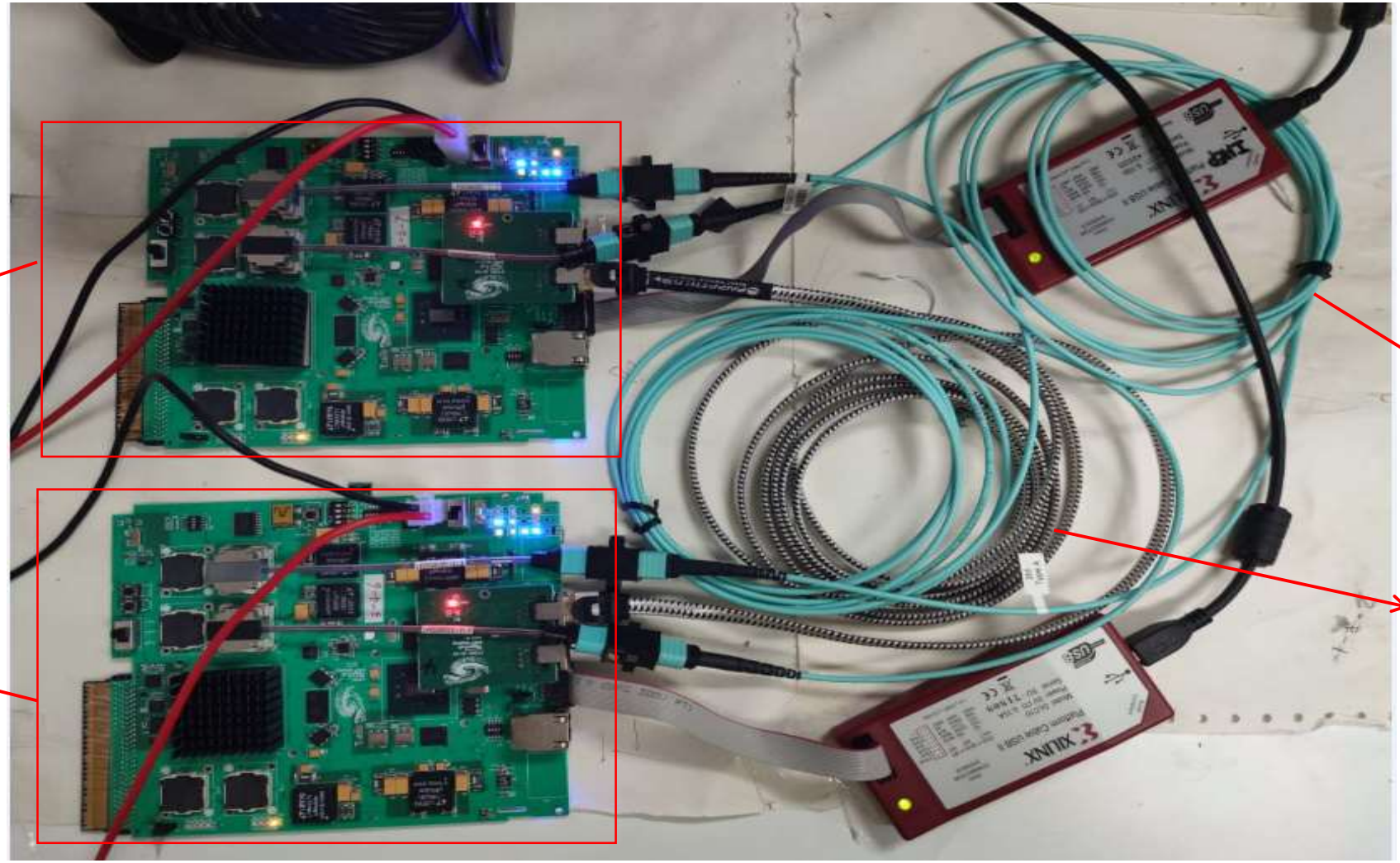
iRPC BEE DAQ Format	Header		Header0
			Header1
	Data Payload	link1 Frame	link_1 Header
			link_1 Angle Value
			link_1 Data
		link2 Frame	link_2 Header
			link_2 Angle Value
			link_2 Data
		…	…
		link_n Frame	link_n Header
			link_n Angle Value
			link_n Data
	Trailer		trailer

- Use different identifiers as Header0 and trailer.
- Pack angle value and original data by link number for events generated in the same BX.
- Not as complete as the CPPF data format for the time being, still working on it!

	Byte 7								Byte 6								Byte 5								Byte 4								Byte 3								Byte 2								Byte 1								Byte 0							
	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header0	x"DEADBEEFBEEFDEAD"																																																																				
Header1	Trigger Number																																Reserved(x"FFAA")																Event Length																			
link Header	LinkID																								Trigger Number																																link Length							
link Angle_value	cluster_1 angle_value																																cluster_2 angle_value																																					
link Data	data1 rising_edge																																data1 falling_edge																																					
	data2 rising_edge																																data2 falling_edge																																					
	…																																…																																					
	data_n rising_edge																																data_n falling_edge																																					
Trailer	x"BEEFDEADDEADBEEF"																																																																				

FEE-BEE test system

- test system setup



BEB

FEE emulator

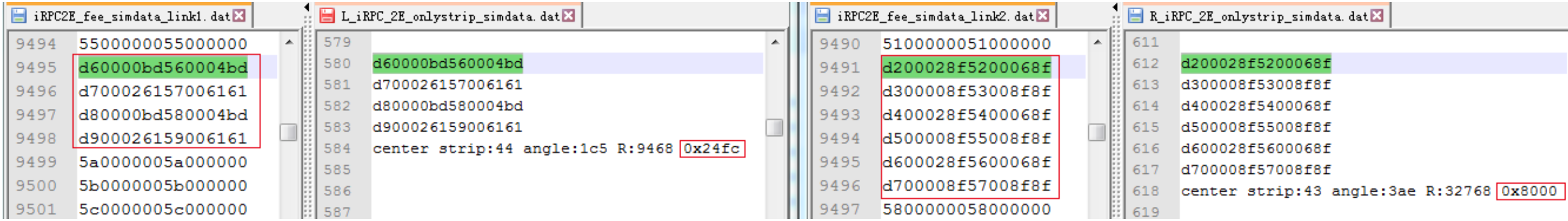
optical fibers

clock fan out: BEE fans out the 40 MHz clock through the on-board clock interface to FEE

BEE firmware test result

- DAQ example：

- Take the first one L1A as example， software simulation data is as below：



- DAQ data frame after packing：
  - **Header0**： identifier x”deadbeefbeefdead”

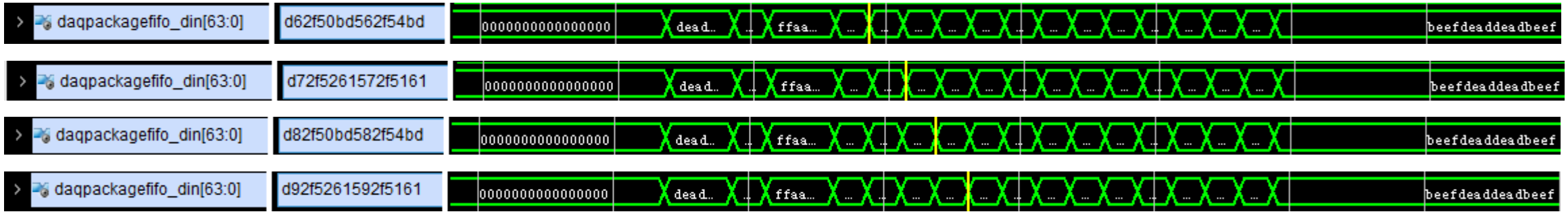
  - **Header1**: trigger_number(32b, x”00000001”)+ reserve(16b, x”ffff”)+ event_length(16b, x”0011”)

  - **Link1 Frame**：
    - **link1_header**: link1_ID (x” FFAA00”) + trigger_number(32b, x”00000001”) + link1_length(8b, x”06”)

BEE firmware test result

- Link1 Frame
  - **link1 angle_value**: cluster1_anglevalue(32b) + cluster2_anglevalue(32b) :
    - *cluster1_anglevalue(32b) = R_value(16b, x"24fc")+phi_value(16b, x"01c5") ,*
    - *only one cluster in this example, so cluster2 angle_value is zero.*

  - **link1 data**: 4 simulator hit data in this simulation, one hit data is inclue 32b rising edge and 32b falling edge.



- **Link2 Frame**:
  - **link2_header**: link2_ID (x" FFAA01") + trigger_number(32b, x"00000001") + link1_length(8b, x"08")

  - **link2 angle_value**: cluster1_anglevalue(32b) + cluster2_anglevalue(32b) :
    - *cluster1_anglevalue(32b) = R_value(16b, x"8000")+phi_value(16b, x"03ae") ,*
    - *only one cluster in this example, so cluster2 angle_value is zero.*

BEE firmware test result

- Link2 Frame：
  - **link2 data**: 6 *64bit data in this simulation, one hit data includes 32b rising edge and 32b falling edge.

- **Trailer**： identifier x”beefdeaddeadbeef”

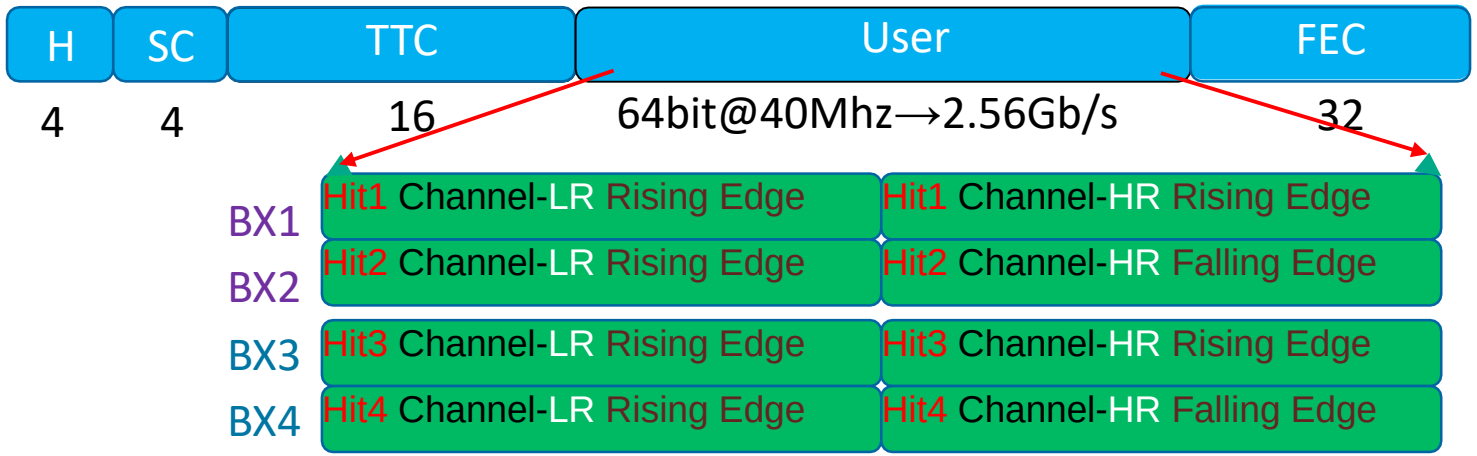
Summary

- FEE function emulation goes well
- BEE firmware development goes well
  - FEE-BEE demo system v1 works well
  - FEE-BEE demo system v2 firmware algorithm is under development
  - Further discussion with FEE group
- Joint test with iRPC FEE at CERN is expected

- Thanks for listening!

Backup

- GBT transmission: 4.8Gb/s (40Mhz@120bit)
  - 80-bit data region fully customized by user
    - the first 16 bits are used for TTC dispatch
    - the remaining 64 bits defined for carrying the detector data
    - 1 GBT frames to cover 1 strip raw data(64bit)
  - So GBT detector data bandwidth :2.56 Gb/s (40Mhz@64bit)

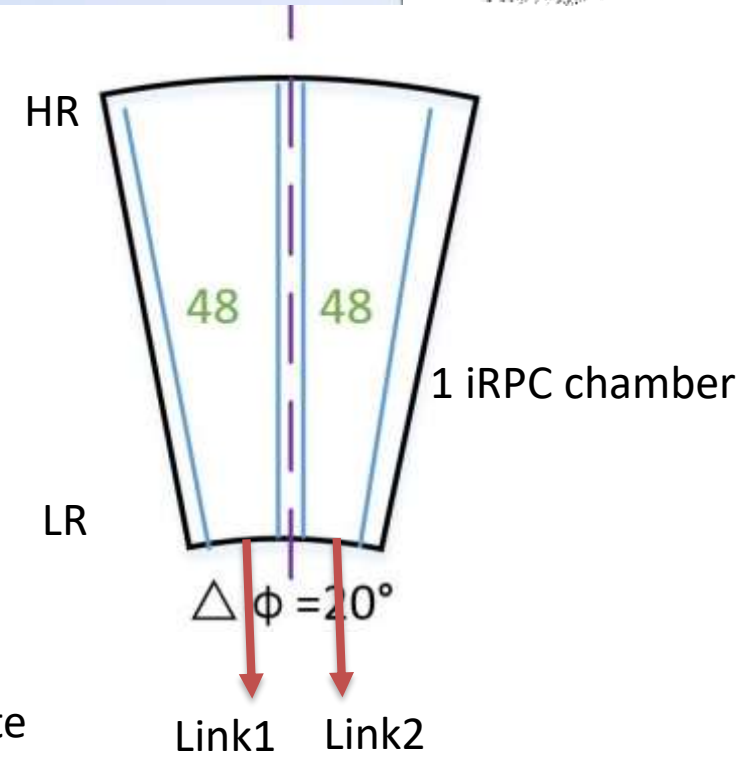


Backup - iRPC bandwidth calculation

- *Rate reference :CMS-TDR-016 p30 and p165*

- Average bandwidth per fiber of iRPC
  - N bit/s = data of one strip *mean cluster size*surface of half chamber*average rate
  - N bit/s = 64 bit_{data}*2.35 $_{cluster\ size}$*6600$cm^2_{surface}$ * 1500 Hz/$cm^2_{average\ rate}$=1.49 Gb/s

- Maximum bandwidth per fiber of iRPC
  - N bit/s= data of one strip *mean cluster size*surface of half chamber*maximum rate
  - N bit/s = 64 bit_{data}*2.35 $_{cluster\ size}$*6600$cm^2_{surface}$ * 2000 Hz/$cm^2_{maximum\ rate}$=1.99 Gb/s



- Raw data readout of half chamber should be possible with a standard GBT link (4.8 Gb/s).