Plan for new device's R&D using Versal for the experiments at KEK

Yun-Tsung Lai

KEK IPNS

ytlai@post.kek.jp

Workshop on Realtime Machine Learning

11th Apr., 2024

Electronics
System Group

Outline

- Application of FPGA in HEP experiments
  - DAQ, L1 Trigger, HLT systems

- Versal project @ KEK IPNS, Collider Electronic Forum:
  - Introduction & Overview
  - Progress on functionality study: PAM4, PCIe, AI engine, DPU
  - HLS and ML inference study plan
  - Algorithm implementation

- Summary & To do

Application of FPGA in HEP experiments

- Here we use Belle II Central Drift Chamber (CDC) as an example.

CDC

Sense wire

Front-End Electronics

- Collect TDC and ADC waveform info.
- Zero suppression.

Optical link

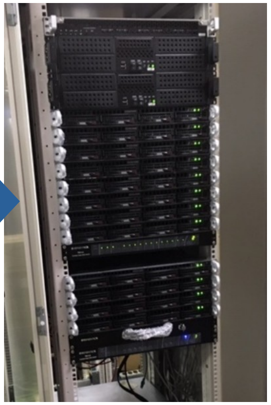
DAQ

Data readout / collector

PCIe / GbE

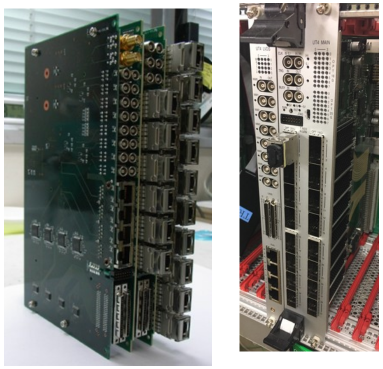
- Data collection.
- Data formatting.
- (Pre-)event building.

HLT / storage



Optical link

L1 Trigger



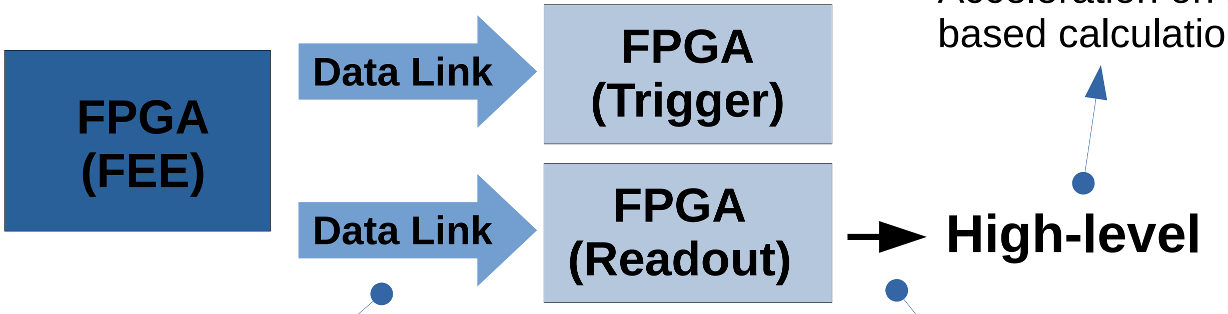
- Real-time complicated trigger algorithm for tracking / clustering / decision.

L1 Trigger signal

- For these application of FPGA device:
  - New technology for data transmission?
  - New technology for logic design in FPGA?
  - Impact on each aspect for experimentalists? For TRG, DAQ, or FEE?

Application of FPGA in HEP experiments (cont'd)

- **Hardware acceleration:**
  - Not only CPU, but also GPU and FPGA.
  - Acceleration on software-based calculation.



- **FPGA - FPGA transmission:**
  - Optical link with FPGA MGT and optical modules.
  - Non-Return-to-Zero (NRZ).
  - Different encoding based on protocol design purposes. e.g. 8B/10B and 64B/66B.
    - <10 Gbps for DAQ.
    - <25 Gbps for TRG.

- Strong **FPGA devices** with:
  - Larger number of cells.
  - Larger data bandwidth.

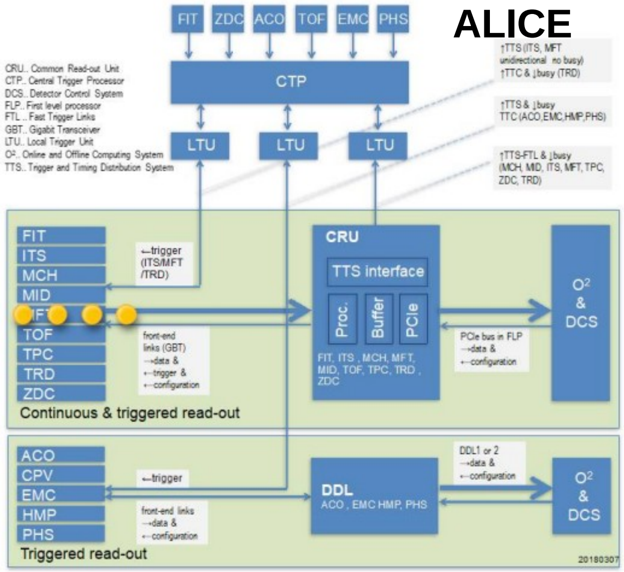
  are critical for the usage in:
  - **TRG**: complicated algorithm implementation.
  - **DAQ**: collect and process large data.

- **FPGA - server transmission:**
  - Data transmission and system slow control.
  - GbE, PCI-express, VME, etc.
  - PCI-Express is the most popular one nowadays: PCIe40 in ALICE, LHCb, and Belle II.

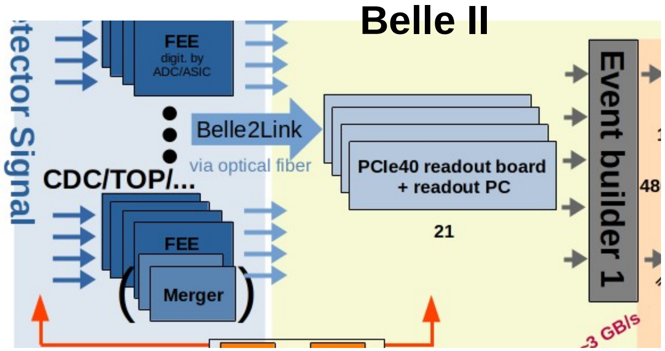
DAQ system

- Readout:
  PCIe has been the most popular solution for electronics → server interface.

ALICE

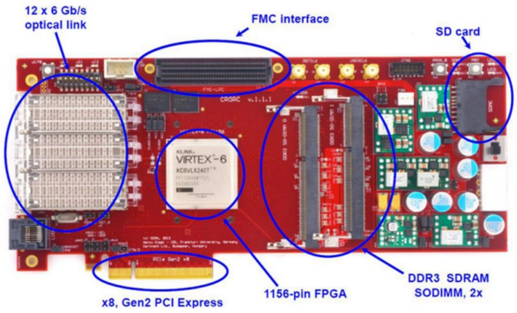

FIT
ZDC
ACO
TOF
EMC
PHS
CTP
LTU
LTU
LTU
CRU.. Common Read-out Unit
CTP.. Central Trigger Processor
DCS.. Detector Control System
FLP.. First level processor
FTL .. Fast Trigger Links
GBT.. Gigabit Transceiver
LTU.. Local Trigger Unit
O².. Online and Offline Computing System
TTS.. Trigger and Timing Distribution System
↑TTS (ITS, MFT unidirectional no busy) ↑TTC & ↓busy (TRD)
↑TTS & ↓busy TTC (ACO,EMC,HMP,PHS)
↑TTS-FTL & ↓busy (MCH, MID, ITS, MFT, TPC, ZDC, TRD)
FIT
ITS
MCH
MID
MFT
TOF
TPC
TRD
ZDC
←trigger (ITS/MFT /TRD)
front-end links (GBT) →data & ←trigger & ←configuration
CRU
TTS interface
Proc.
Buffer
PCIe
FIT, ITS , MCH, MFT, MID, TOF, TPC, TRD , ZDC
PCIe bus in FLP →data & ←configuration
O² & DCS
Continuous & triggered read-out
ACO
CPV
EMC
HMP
PHS
←trigger
front-end links →data & ←configuration
DDL
ACO , EMC HMP, PHS
DDL1 or 2 →data & ←configuration
O² & DCS
Triggered read-out
20180307

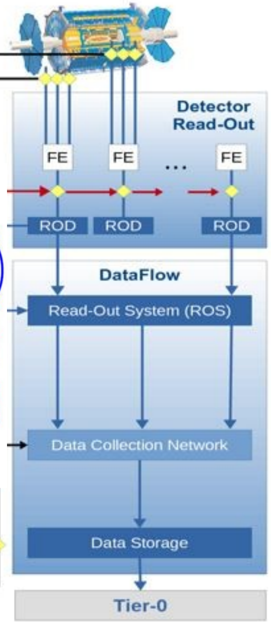
Belle II

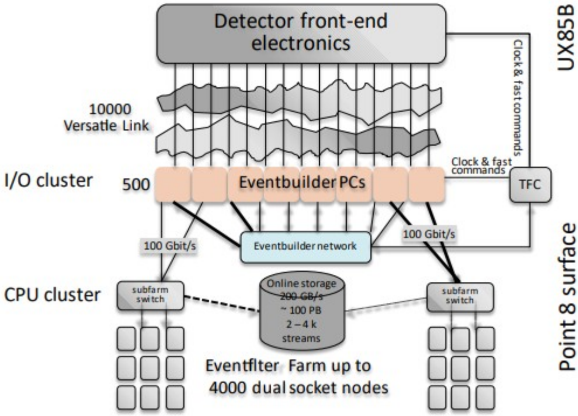


PCIe40: PCIe Gen3

ATLAS RobinNP:
PCIe Gen2



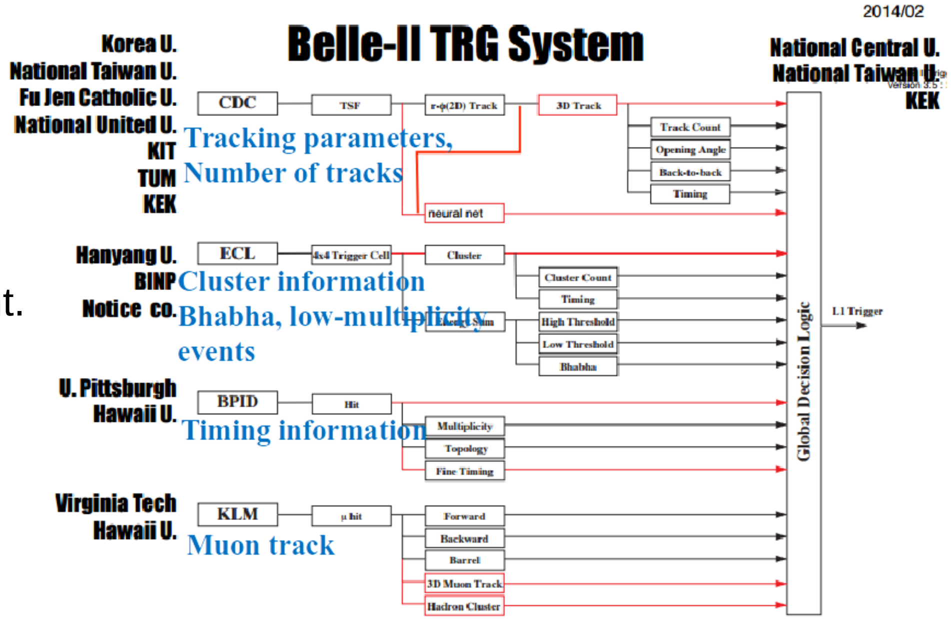




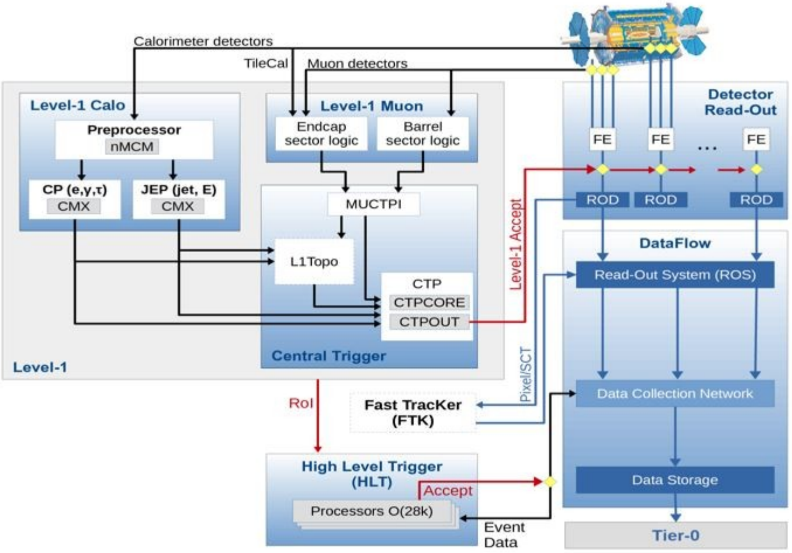
LHCb

L1 Trigger system

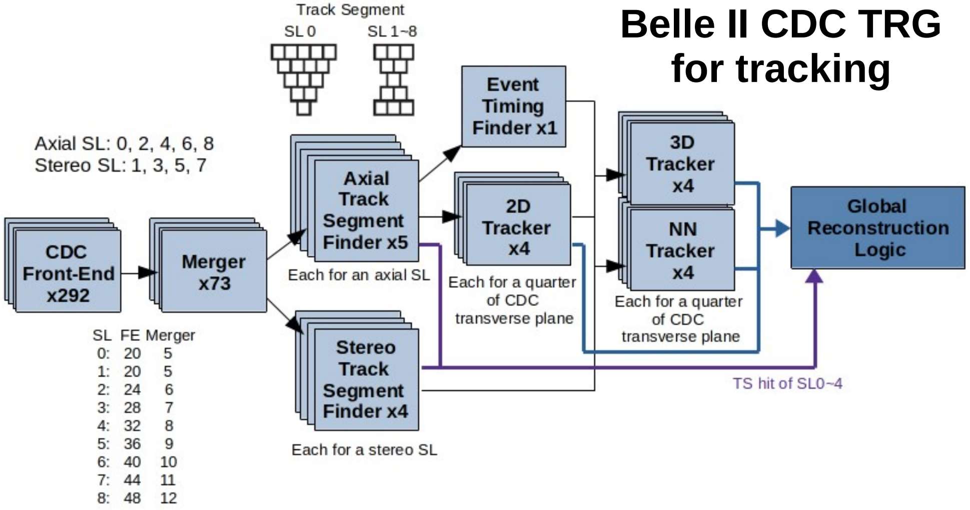
- Provide L1 trigger signal to DAQ using FPGA chips for real-time processing on detector raw data.
- Reason for L1: Buffer storage are not enough for all data due to high event rate and short bunch spacing in collider experiment.



ATLAS L1 TRG



Belle II CDC TRG for tracking



SL	FE	Merger
0:	20	5
1:	20	5
2:	24	6
3:	28	7
4:	32	8
5:	36	9
6:	40	10
7:	44	11
8:	48	12

Trigger device for Belle II and ATLAS

- For TRG purpose, complicated algorithm is implemented to process detector raw data in real-time. Utilization of machine-learning in the logic design became a trend recently.
- Strong FPGA with large resource: improve the logic itself, resolution of triggering, reduce the background rate, and perform everything within a latency limit.

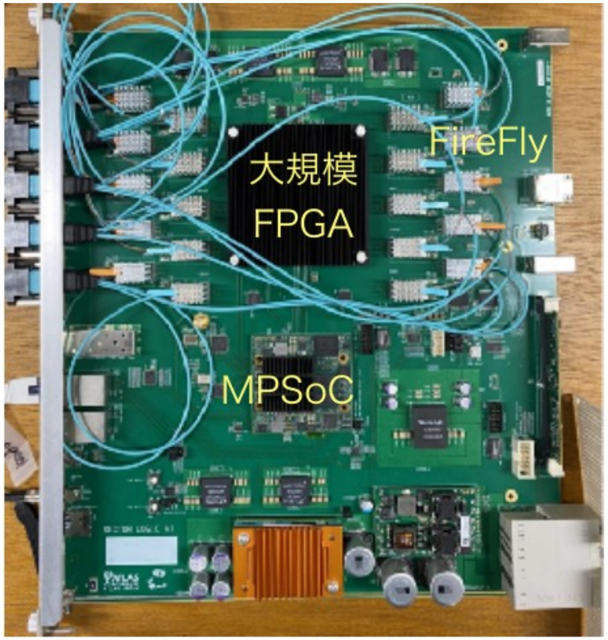
Belle II UT3

Xilinx Virtex-6
xc6vhx380t, xc6vhx565t
11.2 Gbps with 64B/66B

Belle II UT4

Xilinx UltraScale
XCVU080, XCVU160
25 Gbps with 64B/66B

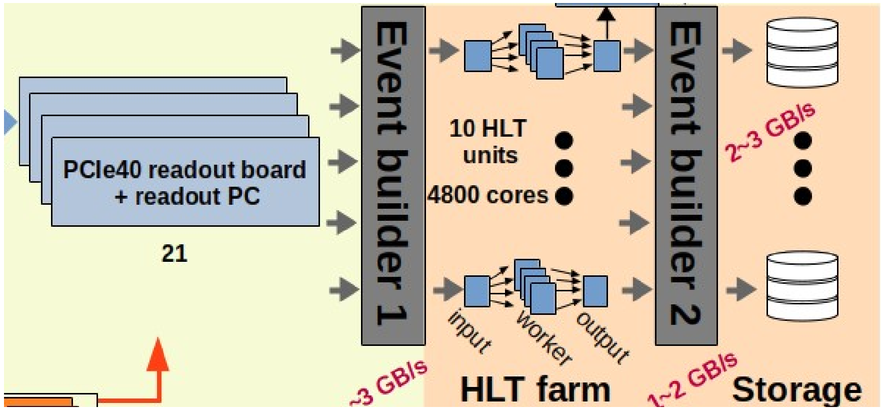
ATLAS Muon Trigger processor



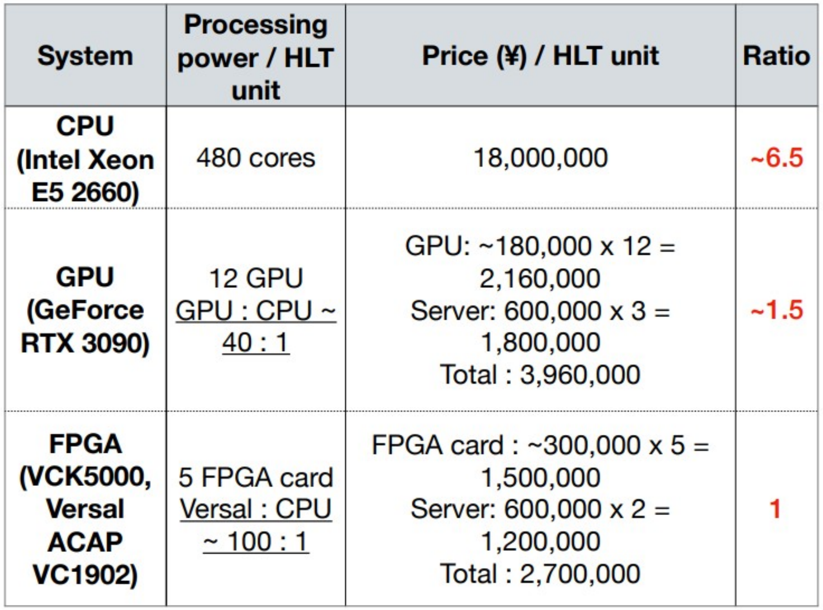
Xilinx UltraScale+
XCVU13P XCZU5EV
GTH,GTY: 16.8 Gbps
with 64B/66B

HLT

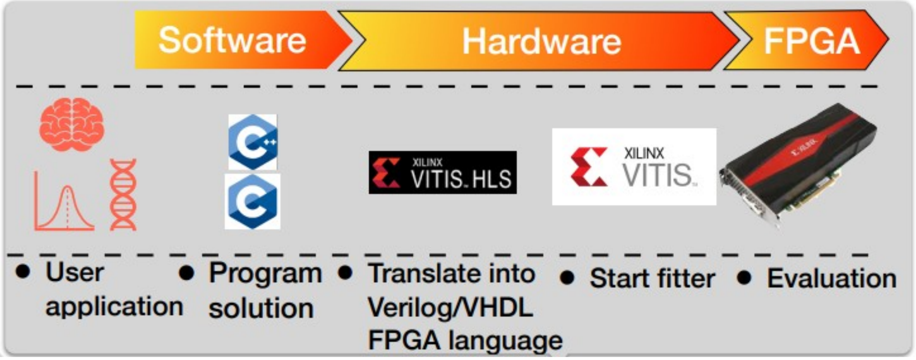
- HLT: Computing servers with reconstruction software.
  - In Belle II: HLT software = offline software.
- How about the options other than CPU?
  - GPU? FPGA for hardware acceleration?



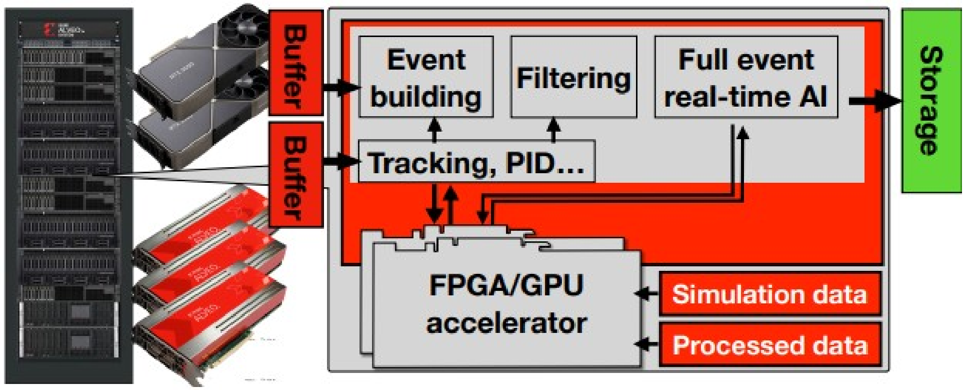
source: Qi-Dong Zhou, Shandong Univ.



System	Processing power / HLT unit	Price (¥) / HLT unit	Ratio
CPU (Intel Xeon E5 2660)	480 cores	18,000,000	~6.5
GPU (GeForce RTX 3090)	12 GPU GPU : CPU ~ 40 : 1	GPU: ~180,000 x 12 = 2,160,000 Server: 600,000 x 3 = 1,800,000 Total : 3,960,000	~1.5
FPGA (VCK5000, Versal ACAP VC1902)	5 FPGA card Versal : CPU ~ 100 : 1	FPGA card : ~300,000 x 5 = 1,500,000 Server: 600,000 x 2 = 1,200,000 Total : 2,700,000	1



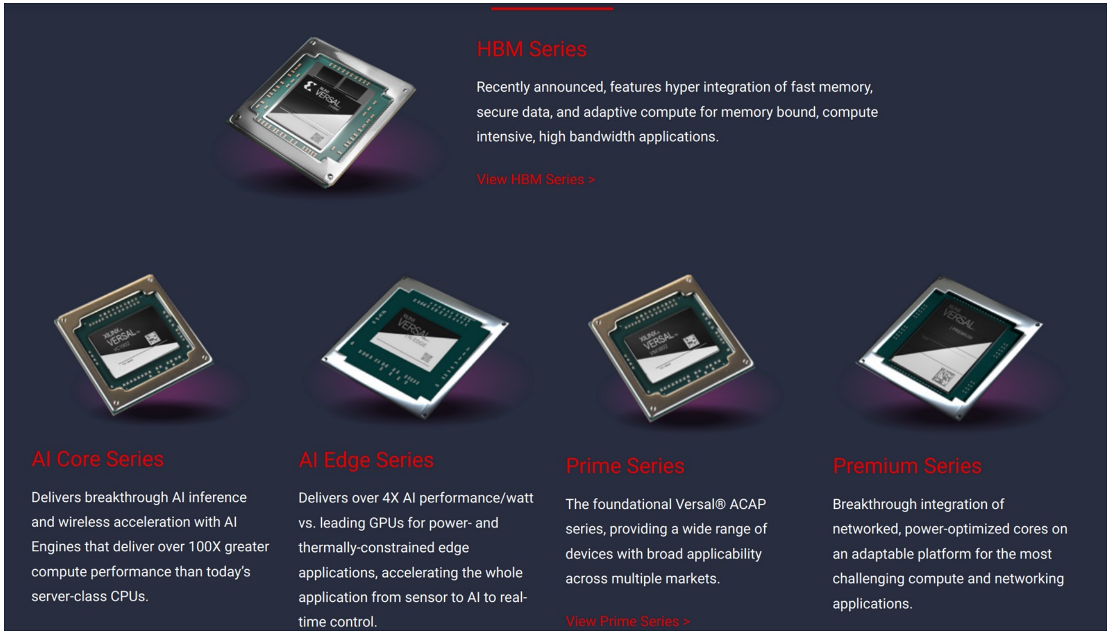
- User application
- Program solution
- Translate into Verilog/VHDL FPGA language
- Start fitter
- Evaluation



Versal project @ KEK IPNS

- "Collider Electronics Forum": A new platform for electronics associated technical communication and common device R&D in Japanese HEP community.
  - KEK IPNS: E-sys, Belle II, Energy Frontier groups.
  - Experiment groups (Belle II, ATLAS, ALICE, nuclear physics) in Japan.

- We purchased a few evaluation kits of the Xilinx Versal series ACAP for joint study.
  - Plan: Common and general studies on the new technologies for future electronics device's R&D. Now we plan to use Versal for L1 TRG, DAQ or HLT purpose.



HBM Series

Recently announced, features hyper integration of fast memory, secure data, and adaptive compute for memory bound, compute intensive, high bandwidth applications.

View HBM Series >

AI Core Series

Delivers breakthrough AI inference and wireless acceleration with AI Engines that deliver over 100X greater compute performance than today's server-class CPUs.

AI Edge Series

Delivers over 4X AI performance/watt vs. leading GPUs for power- and thermally-constrained edge applications, accelerating the whole application from sensor to AI to real-time control.

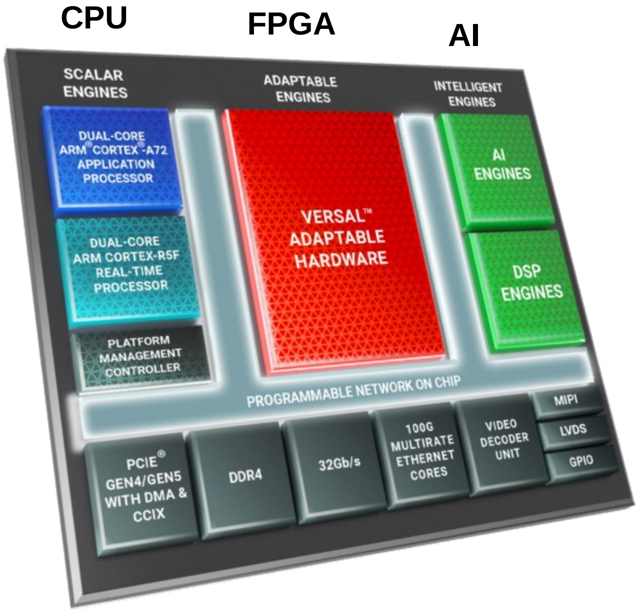
Prime Series

The foundational Versal® ACAP series, providing a wide range of devices with broad applicability across multiple markets.

View Prime Series >

Premium Series

Breakthrough integration of networked, power-optimized cores on an adaptable platform for the most challenging compute and networking applications.


CPU
FPGA
AI
SCALAR ENGINES
ADAPTABLE ENGINES
INTELLIGENT ENGINES
DUAL-CORE ARM® CORTEX-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
VERSAL™ ADAPTABLE HARDWARE
AI ENGINES
DSP ENGINES
PROGRAMMABLE NETWORK ON CHIP
PCIE® GEN4/GEN5 WITH DMA & CCIX
DDR4
32Gb/s
100G MULTIRATE ETHERNET CORES
VIDEO DECODER UNIT
MIPI
LVDS
GPIO

source: Xilinx website

Versal project: General plan and roadmap

- Our goal: R&D of a new general FPGA device using the Versal ACAP.
  - A L1 TRG, DAQ, or HLT device, and also general for different experiments.
  - One clear target is **UT5 for L1 TRG of both Belle II and ATLAS**.

1st year:

- Study the properties of the fundamental functionalities with the kits:
  - GTM (PAM4), PCIe Gen5, AI/DSP engine, CPU acceleration, etc.
- Prepare basic application for each of them for other members.

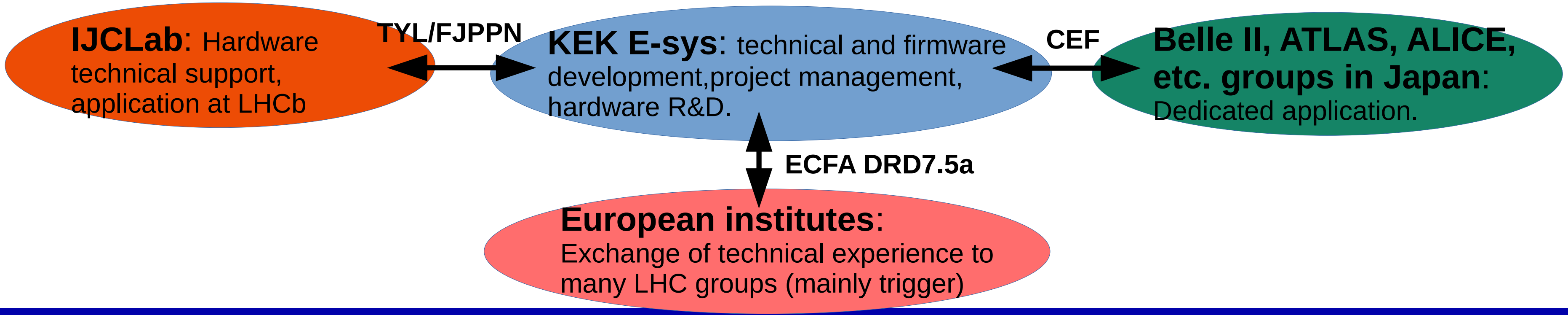
Here we are now with VPK120 and VCK190. →

2nd year:

- Make general transmission protocols for GTM (PAM4), PCIe Gen5, and do performance study.
- Implement various Trigger algorithms (Belle II, ATLAS, etc).
- Connect to existing systems to take real-time data and check performance.

3rd year:

- Future universal device: L1 TRG, DAQ readout, or HLT.
  - Discussion.
  - Schematic/PCB design for the prototype boards.
  - Test with experiments people.



IJCLab: Hardware technical support, application at LHCb

KEK E-sys: technical and firmware development,project management, hardware R&D.

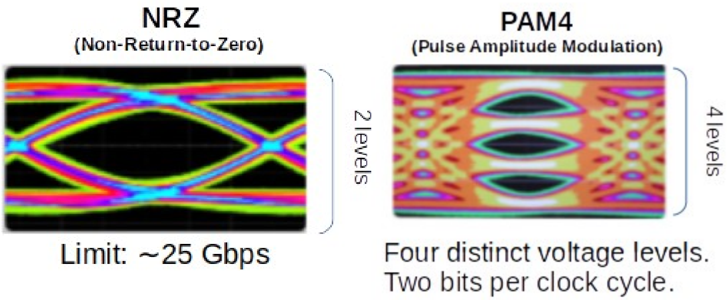
Belle II, ATLAS, ALICE, etc. groups in Japan: Dedicated application.

European institutes: Exchange of technical experience to many LHC groups (mainly trigger)

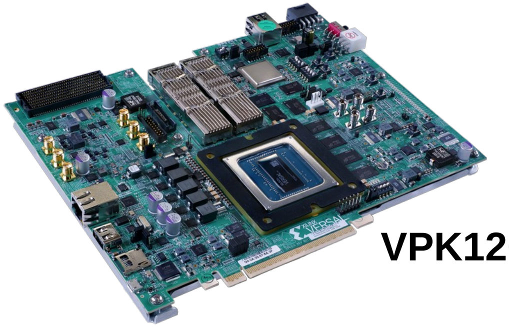
New technology in Versal FPGA: PAM4, PCIe, AI engine

- **Pulse Amplitude Modulation (PAM4):**
  - Four distinct voltage levels to break through the limit of Non-Return-to-Zero (NRZ), which is ~25 Gbps.
  - Using VPK120 to study it.
  - Suitable for high-speed link in L1 TRG.
    Hope to be pioneer to use it in future TRG board.

source: Xilinx

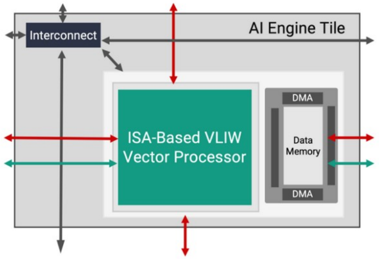

NRZ
(Non-Return-to-Zero)
2 levels
Limit: ~25 Gbps
PAM4
(Pulse Amplitude Modulation)
4 levels
Four distinct voltage levels.
Two bits per clock cycle.

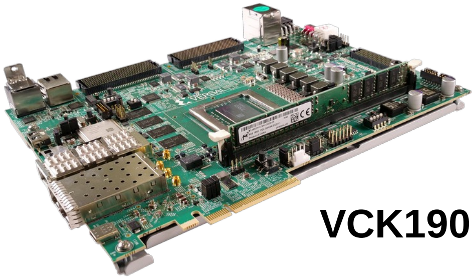
- **PCIe Gen5**:
  - PCIe has been popular option in HEP.
    - ALICE, LHCb and Belle II has been using PCIe40 (Gen3).
  - Study the properties of newer generation of PCIe is beneficial for the future readout device's development.
  - Using VPK120.



VPK120

- **AI engine**: A new technology for data processing.
  - Help for our algorithm construction in TRG.
  - C programmable.
  - Together, we study many options of HLS and ML inference in FPGA, and their performance in different TRG algorithms.
  - Will use VCK190.





VCK190

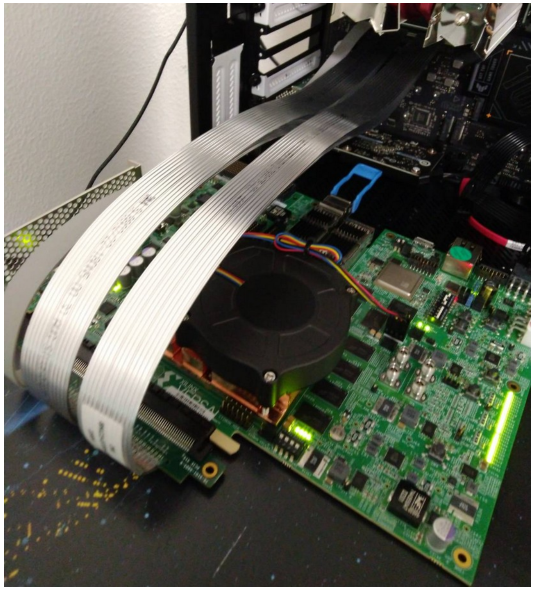
Test bench setup @ KEK E-sys group

- The test bench of VPK120 has been built at E-sys group and released to our members for dedicated studies.

- VCK190 has also arrived at KEK in March. Preparation study is ongoing and will be ready soon.

- Special thanks to Mathis Maurice, internship in E-sys group in 2023 summer, for helping this VPK120 preparation work!

PC side: PCIe Gen5 x16 slot



VPK120 test bench: 2023 summer

PC side: PCIe Gen4 x8 slot

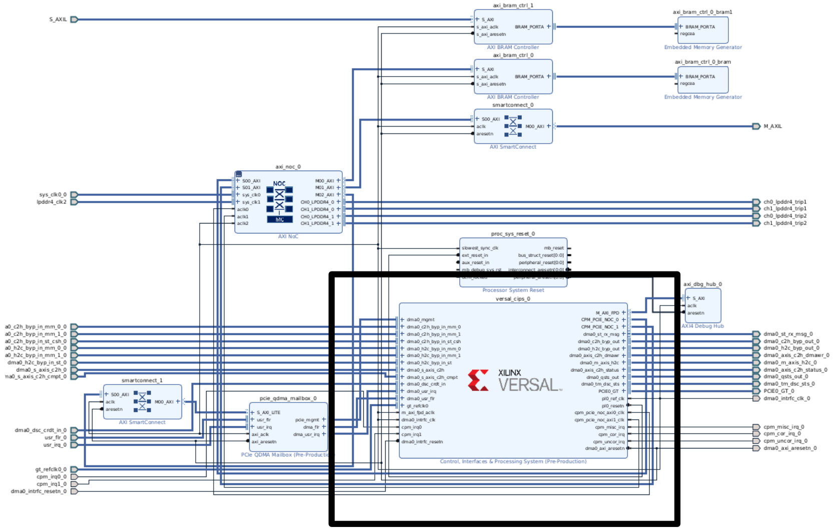
VCK190 test bench: 2024 March

Firmware making with Versal: PS, CIPS and NOC

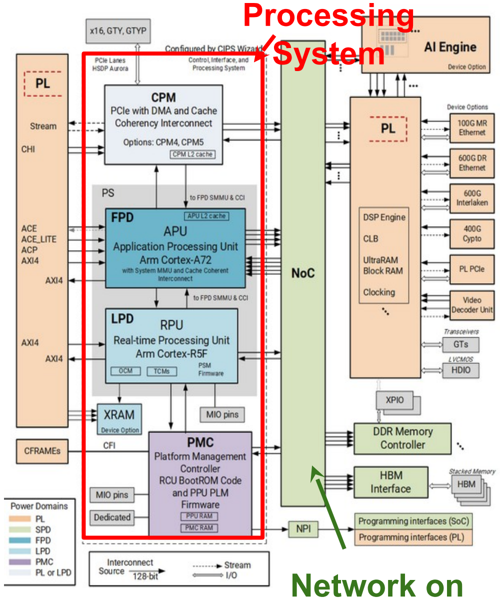
- In our experience, FPGA firmware making is:
  - Writing HDL codes and using IPcore to control all the **Programmable Logic (PL)**.

- But Versal is an ACAP containing lots of sub-systems together with the FPGA.
  - Not only PL, but also **Processor System (PS)**.
  - Firmware making tends to rely on the automatic block design rather than the traditional code-writing way.
  - For now, we still have limited understanding in PS.

A firmware design with PCIe



CIPS (Common Interface Processing System) : Interface referring to the integrated processing sub-system



Control Interface and Processing System

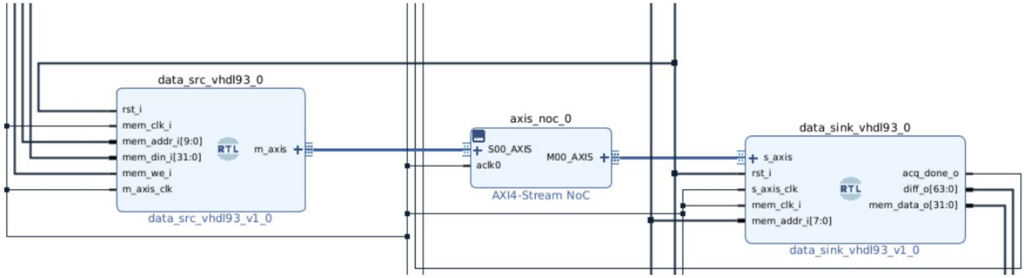
In the future, if we use AI engine for logic design, latency from NOC is important.

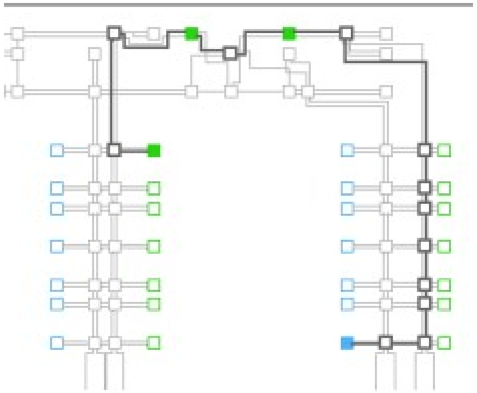
NOC (Network On Chip): Communication network for sub-systems of the FPGA.

Network on Chip

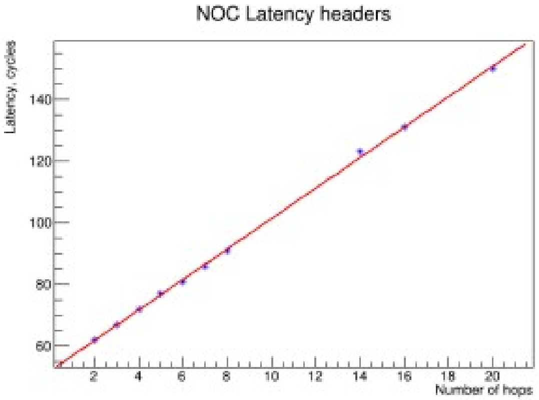
Latency measurement with NOC

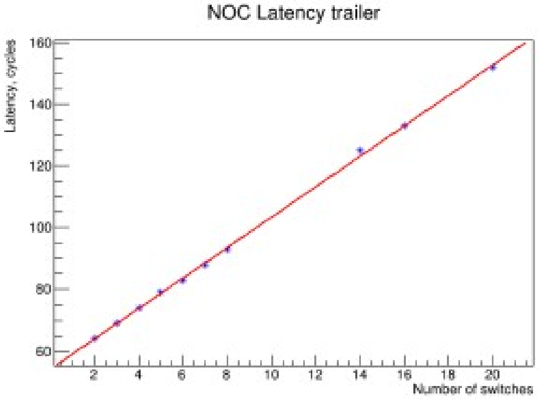
- Study by Dmytro Levit (KEK).

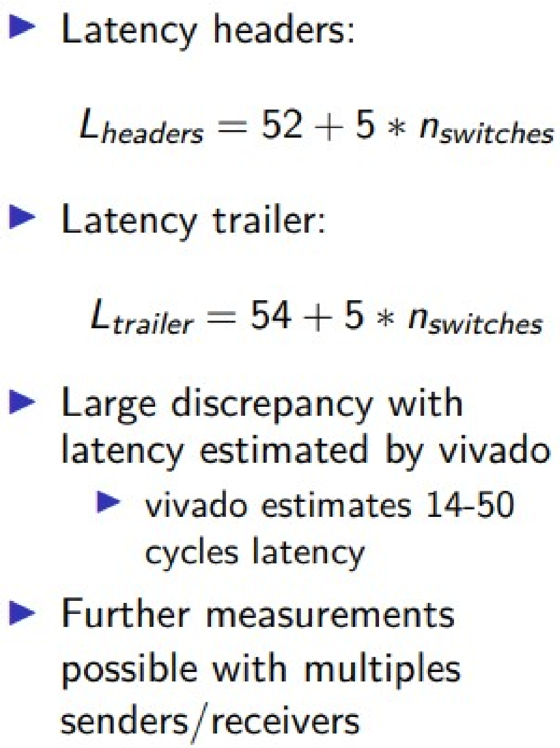




A case of 20 switches







- Latency headers:

$$L_{headers} = 52 + 5 * n_{switches}$$

- Latency trailer:

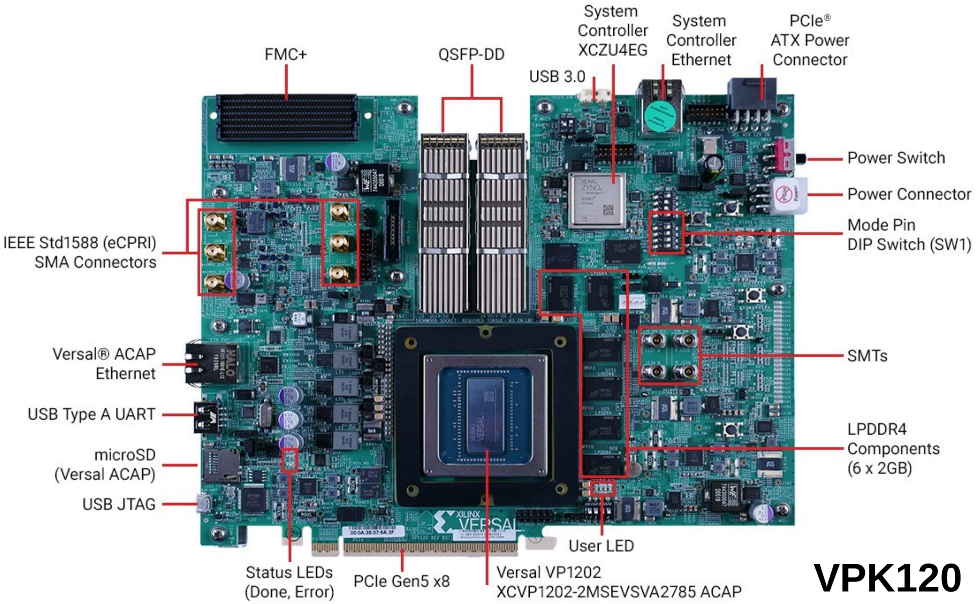
$$L_{trailer} = 54 + 5 * n_{switches}$$

- Large discrepancy with latency estimated by vivado
  - vivado estimates 14-50 cycles latency
- Further measurements possible with multiples senders/receivers

Versal transceivers: GTYP and GTM

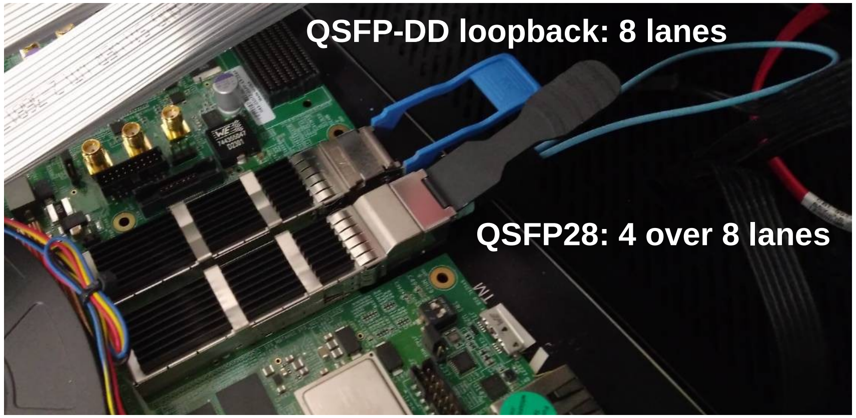
- GTYP: PCIe 5.0 (16) and FMC+ (8)
  - 1.25 ~ 32.75 Gb/s.
  - Various encoder supported.

- GTM: QSFP-DD (8*2)
  - NRZ:
    - 9.5 ~ 15, 19 ~ 29 Gb/s.
  - PAM4:
    - 19 ~ 30, 38 ~ 60 Gb/s
    - 76 ~ 112 Gb/s: "Half density mode" by combining two lanes.
    - No encoding is supported.
      Need to be make them manually in RTL.


FMC+
QSFP-DD
USB 3.0
System Controller XCZU4EG
System Controller Ethernet
PCIe® ATX Power Connector
Power Switch
Power Connector
Mode Pin DIP Switch (SW1)
IEEE Std1588 (eCPRI) SMA Connectors
Versal® ACAP Ethernet
USB Type A UART
microSD (Versal ACAP)
USB JTAG
SMTs
LPDDR4 Components (6 x 2GB)
User LED
Status LEDs (Done, Error)
PCIe Gen5 x8
Versal VP1202 XCVP1202-2MSEVSVA2785 ACAP

VPK120

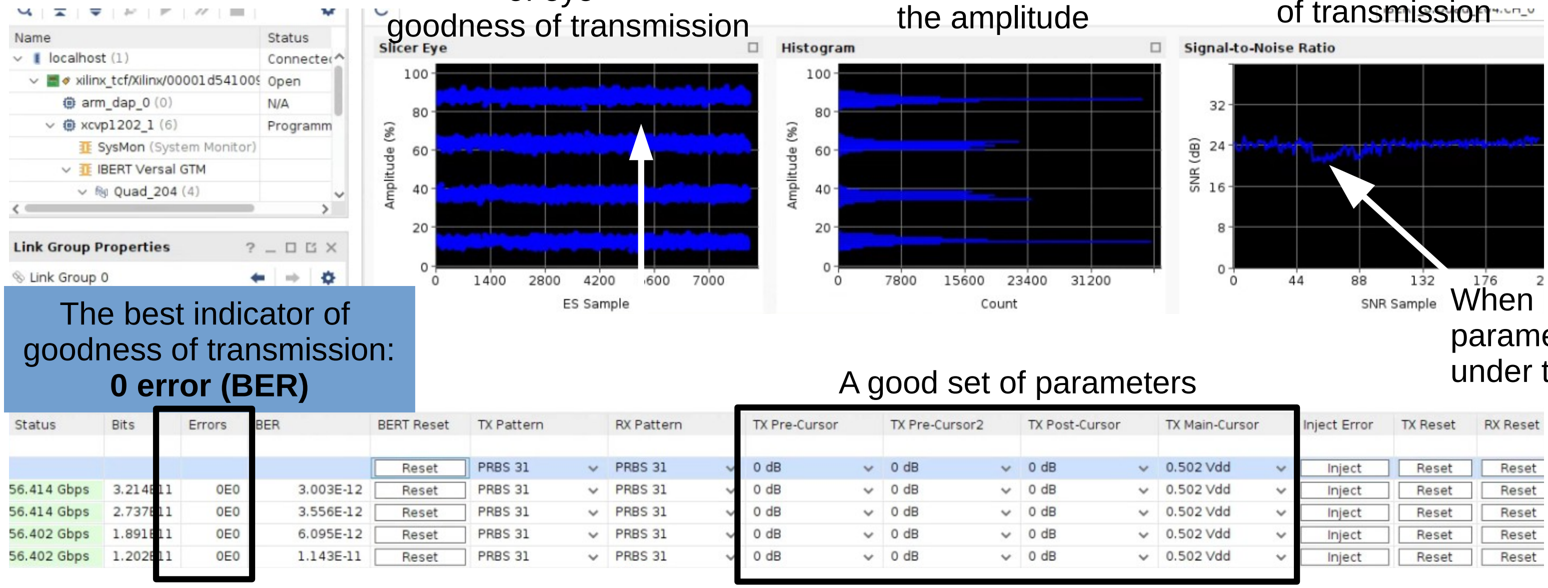
- Our test setup for transceiver study:



PAM4 56 Gbps with GTM IBERT, QSFPDD loopback

- PAM4, 56 Gbps per lane. QSFPDD loopback module.
  - Parameter tuning on cursor position and termination voltage, etc, is necessary to have stable transmission (0 bit error).

DesignCon 2019 Enabling IBIS-AMI Simulations for Systems Containing PAM4 Retimers at 112Gbps



amplitude/height of eye ≠ goodness of transmission

Histogram of the amplitude

SNR ≠ goodness of transmission

When parameters were under tuning.

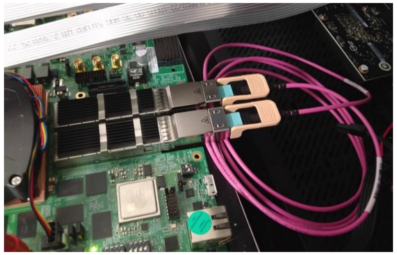
The best indicator of goodness of transmission: **0 error (BER)**

A good set of parameters

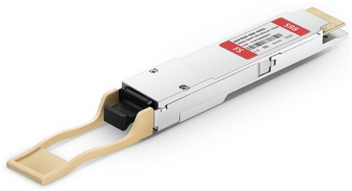
Status	Bits	Errors	BER	BERT Reset	TX Pattern	RX Pattern	TX Pre-Cursor	TX Pre-Cursor2	TX Post-Cursor	TX Main-Cursor	Inject Error	TX Reset	RX Reset
				Reset	PRBS 31	PRBS 31	0 dB	0 dB	0 dB	0.502 Vdd	Inject	Reset	Reset
56.414 Gbps	3.214E11	0E0	3.003E-12	Reset	PRBS 31	PRBS 31	0 dB	0 dB	0 dB	0.502 Vdd	Inject	Reset	Reset
56.414 Gbps	2.737E11	0E0	3.556E-12	Reset	PRBS 31	PRBS 31	0 dB	0 dB	0 dB	0.502 Vdd	Inject	Reset	Reset
56.402 Gbps	1.891E11	0E0	6.095E-12	Reset	PRBS 31	PRBS 31	0 dB	0 dB	0 dB	0.502 Vdd	Inject	Reset	Reset
56.402 Gbps	1.202E11	0E0	1.143E-11	Reset	PRBS 31	PRBS 31	0 dB	0 dB	0 dB	0.502 Vdd	Inject	Reset	Reset

Plan

- Further study with realistic QSFPDD module and MPO-16 is ongoing.
  - Much higher BER (~10^{-6})
  - Forward-Error-Correction will be implemented in our protocol.
  - Also other types of PAM4-supported modules: FireFly, etc.



QSFP-DD-SR8

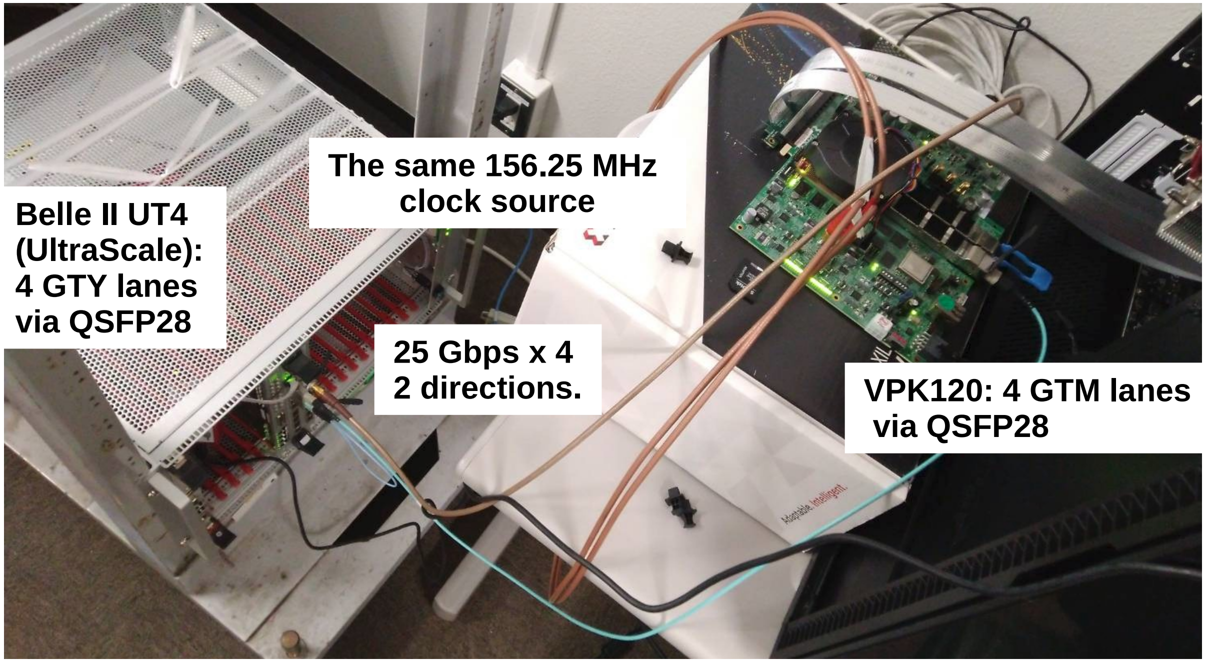


Protocol development and connection test

- Both 8B/10B and 64B/66B (sync. gearbox) are tested with GTM.
- Raw mode with No encoding: A new generalized protocol has been also made.
  - Similar logic to my Belle II TRG protocol design.
  - (de)scrambler for DC balance.
  - Tested to be stable for both NRZ and PAM4.

- Using this new generalized protocol, connection test (25 Gbps x4, NRZ) between Belle II UT4 and VPK120 has been also tested. Stable in few hours.
  - Will test with ATLAS muon board soon.

Belle II UT4


The same 156.25 MHz clock source
Belle II UT4 (UltraScale): 4 GTY lanes via QSFP28
25 Gbps x 4 2 directions.
VPK120: 4 GTM lanes via QSFP28

Latency for Versal GTYP and UltraScale(+) GTY

- Latency is a big concern for L1 TRG system.
  - Since the beginning of Belle II TRG preparation, we have been studying latency reduction in data links.
  - Now we have 25 Gbps running.

- The following are the simulation values from Xilinx website with internal encoder.
  - UT4: Virtex UltraScale

- Measured latency in **bold**: Based on the Belle II TRG protocol.

	Raw (UI)	Raw + Async. 64B/66B (UI)	10 Gbps, Raw (ns)	10 Gbps, 64B/66B (ns)	25 Gbps, Raw (ns)	25 Gbps, 64B/66B (ns)
Versal GTYP 64/64	1127					
Versal GTYP 64/32	688					
UT4 GTY 64/64	768	1458	77 **115**	146 **147**	31 **33**	58 **58**
UT4 GTY 64/32	414	990	41 **90**	99 **122**		

Typical value for 1 link in the present Belle II TRG: **50~100 ns**

Latency for Versal GTM

- If we adapt to use Versal GTM: Larger latency will be introduced.

- The following are the max. simulation values from Xilinx website with No encoding.
  - Measured latency in **bold**: Based on our generalized protocol.

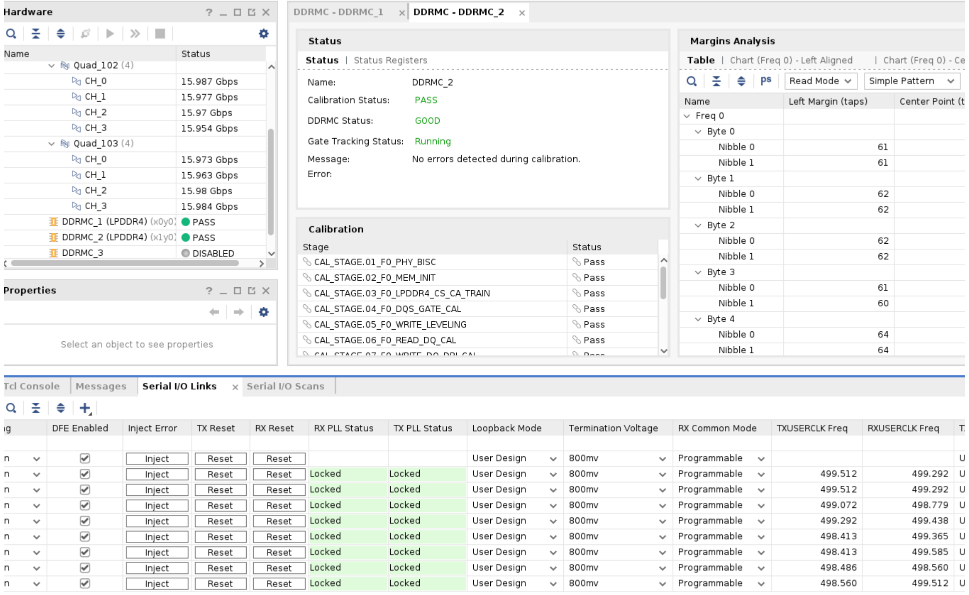
- For the same setup, latency in term of clock-cycle is basically the same.
  - Higher speed is preferred as the processing latency is much smaller.
  - In general, latency of GTM is much larger that that of UltraScale(+) GTY or so.

- If we use GTM, just go with PAM4 with > 50 Gbps.

Versal GTM	Unit Interval (UI)	10 Gbps (ns)	25 Gbps (ns)	56 Gbps (ns)	106 Gbps (ns)
NRZ 64b	5833	583 **640**	233 **256**		
NRZ 160b	4964	496 **730**	198 **237**		
PAM4 160b	2957			53 **97**	
PAM4 256b	3233			57 **133**	
PAM4 320b	3095				29 **66**
PAM4 512b	3690				35

PCIe-CPM test

- CPM-PCIe example from Xilinx: XTP712
  - CPM: building block design for PCIe with integrating DMA, CIPS, NOC, etc.
  - PCIe Gen4 x8:
    GTYP links are up. 16 Gbps per lane.
- Driver software: QDMA, also a Xilinx IP.
- Data exchange test with the QDMA software:
- We spent much time in mine-sweeping
  - Will start to make real protocol for event data readout purpose.
    - Similar to the one in Belle II DAQ.



Hardware

Name	Status
Quad_102 (4)	
CH_0	15.987 Gbps
CH_1	15.977 Gbps
CH_2	15.97 Gbps
CH_3	15.954 Gbps
Quad_103 (4)	
CH_0	15.973 Gbps
CH_1	15.963 Gbps
CH_2	15.98 Gbps
CH_3	15.984 Gbps
DDRMC_1 (LPDDR4) (x0y0)	● PASS
DDRMC_2 (LPDDR4) (x1y0)	● PASS
DDRMC_3	● DISABLED

Properties

Select an object to see properties

DDRMC - DDRMC_1 | DDRMC - DDRMC_2

Status

Status | Status Registers

Name: DDRMC_2
Calibration Status: PASS
DDRMC Status: GOOD
Gate Tracking Status: Running
Message: No errors detected during calibration.
Error:

Calibration

Stage	Status
CAL_STAGE.01_F0_PHY_BISC	Pass
CAL_STAGE.02_F0_MEM_INIT	Pass
CAL_STAGE.03_F0_LPDDR4_CS_CA_TRAIN	Pass
CAL_STAGE.04_F0_DQS_GATE_CAL	Pass
CAL_STAGE.05_F0_WRITE_LEVELING	Pass
CAL_STAGE.06_F0_READ_DQ_CAL	Pass
CAL_STAGE.07_F0_WRITE_DQ_DBI_CAL	Pass

Margins Analysis

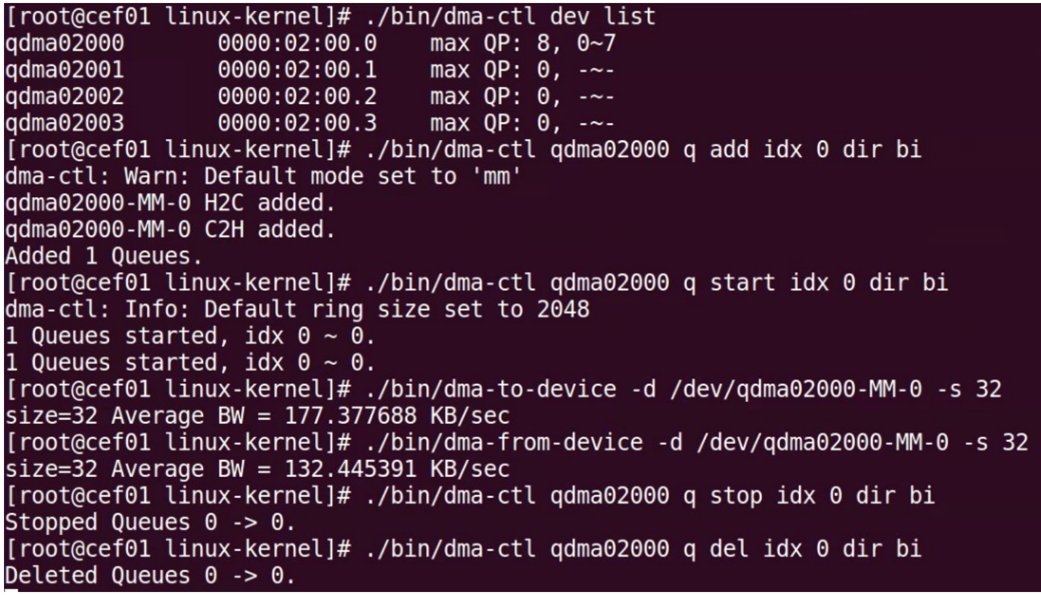
Table | Chart (Freq 0) - Left Aligned | Chart (Freq 0) - Ce

Read Mode | Simple Pattern

Name	Left Margin (taps)	Center Point (t
Freq 0		
Byte 0		
Nibble 0	61	
Nibble 1	61	
Byte 1		
Nibble 0	62	
Nibble 1	62	
Byte 2		
Nibble 0	62	
Nibble 1	62	
Byte 3		
Nibble 0	61	
Nibble 1	60	
Byte 4		
Nibble 0	64	
Nibble 1	64	

Tcl Console | Messages | Serial I/O Links | Serial I/O Scans

ıg	DFE Enabled	Inject Error	TX Reset	RX Reset	RX PLL Status	TX PLL Status	Loopback Mode	Termination Voltage	RX Common Mode	TXUSERCLK Freq	RXUSERCLK Freq	T
n	☑	Inject	Reset	Reset			User Design	800mv	Programmable			U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	499.512	499.292	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	499.512	499.292	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	499.072	498.779	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	499.292	499.438	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	498.413	499.365	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	498.413	499.585	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	498.486	498.560	U
n	☑	Inject	Reset	Reset	Locked	Locked	User Design	800mv	Programmable	498.560	499.512	U

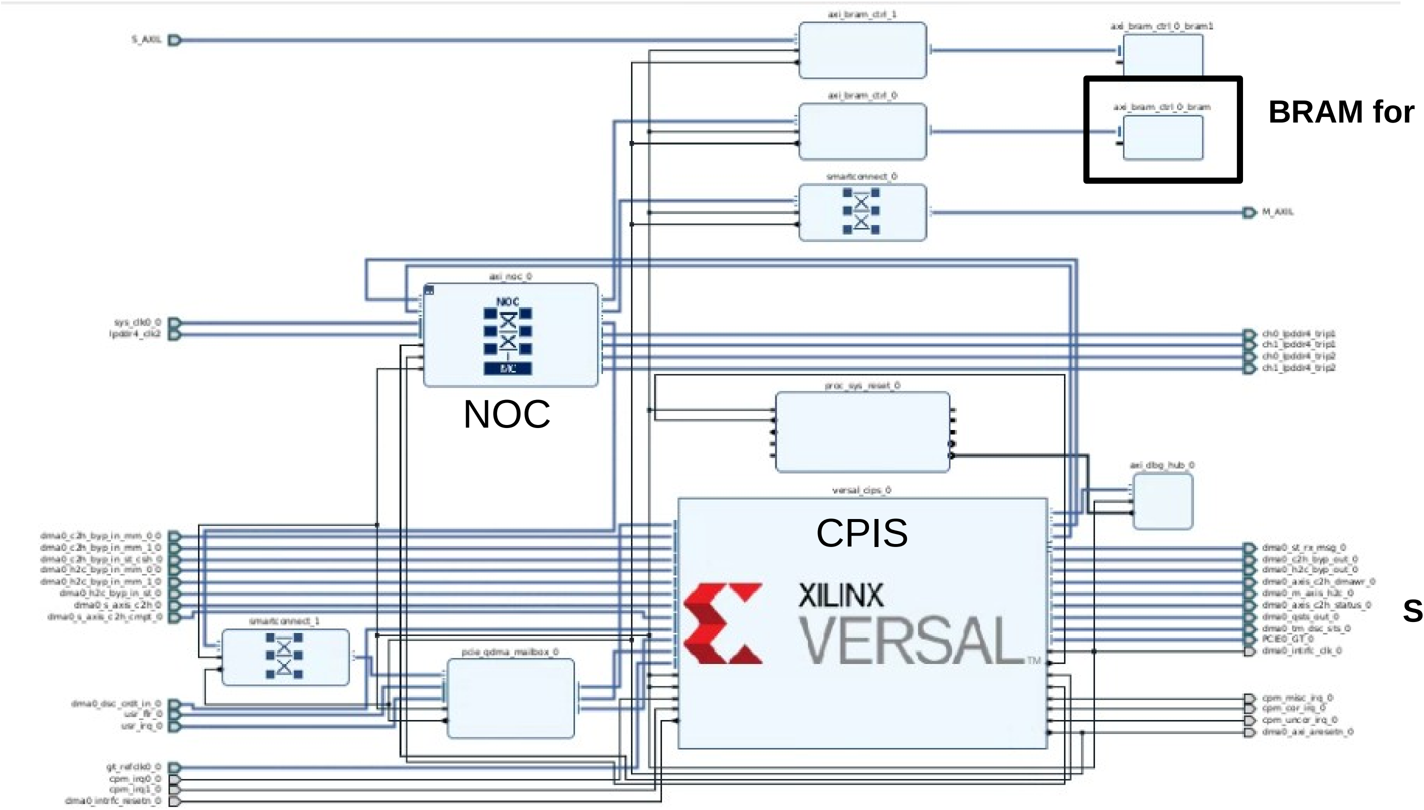


```
[root@cef01 linux-kernel]# ./bin/dma-ctl dev list
qdma02000       0000:02:00.0    max QP: 8, 0~7
qdma02001       0000:02:00.1    max QP: 0, -~-
qdma02002       0000:02:00.2    max QP: 0, -~-
qdma02003       0000:02:00.3    max QP: 0, -~-
[root@cef01 linux-kernel]# ./bin/dma-ctl qdma02000 q add idx 0 dir bi
dma-ctl: Warn: Default mode set to 'mm'
qdma02000-MM-0 H2C added.
qdma02000-MM-0 C2H added.
Added 1 Queues.
[root@cef01 linux-kernel]# ./bin/dma-ctl qdma02000 q start idx 0 dir bi
dma-ctl: Info: Default ring size set to 2048
1 Queues started, idx 0 ~ 0.
1 Queues started, idx 0 ~ 0.
[root@cef01 linux-kernel]# ./bin/dma-to-device -d /dev/qdma02000-MM-0 -s 32
size=32 Average BW = 177.377688 KB/sec
[root@cef01 linux-kernel]# ./bin/dma-from-device -d /dev/qdma02000-MM-0 -s 32
size=32 Average BW = 132.445391 KB/sec
[root@cef01 linux-kernel]# ./bin/dma-ctl qdma02000 q stop idx 0 dir bi
Stopped Queues 0 -> 0.
[root@cef01 linux-kernel]# ./bin/dma-ctl qdma02000 q del idx 0 dir bi
Deleted Queues 0 -> 0.
```

PCIe-CPM firmware

- The Xilinx PCIe-CPM IP provides two modes:
  - Memory-Map (MM)
  - Streaming

- Next, we started to make the firmware/software for continuous event readout for realistic experimental purpose.



NOC

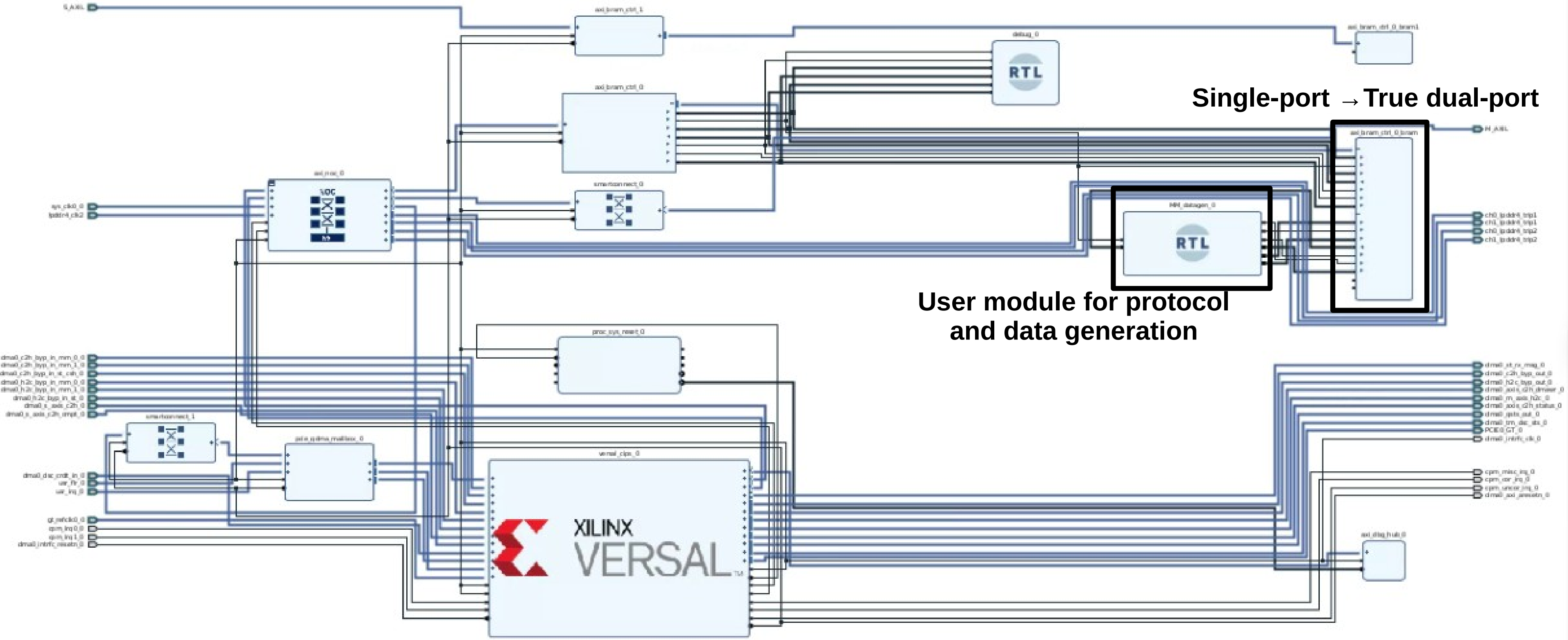
CPIS

BRAM for MM mode

ST mode interface

PCIe-CPM firmware: Event readout using MM mode

- New firmware based on MM mode.

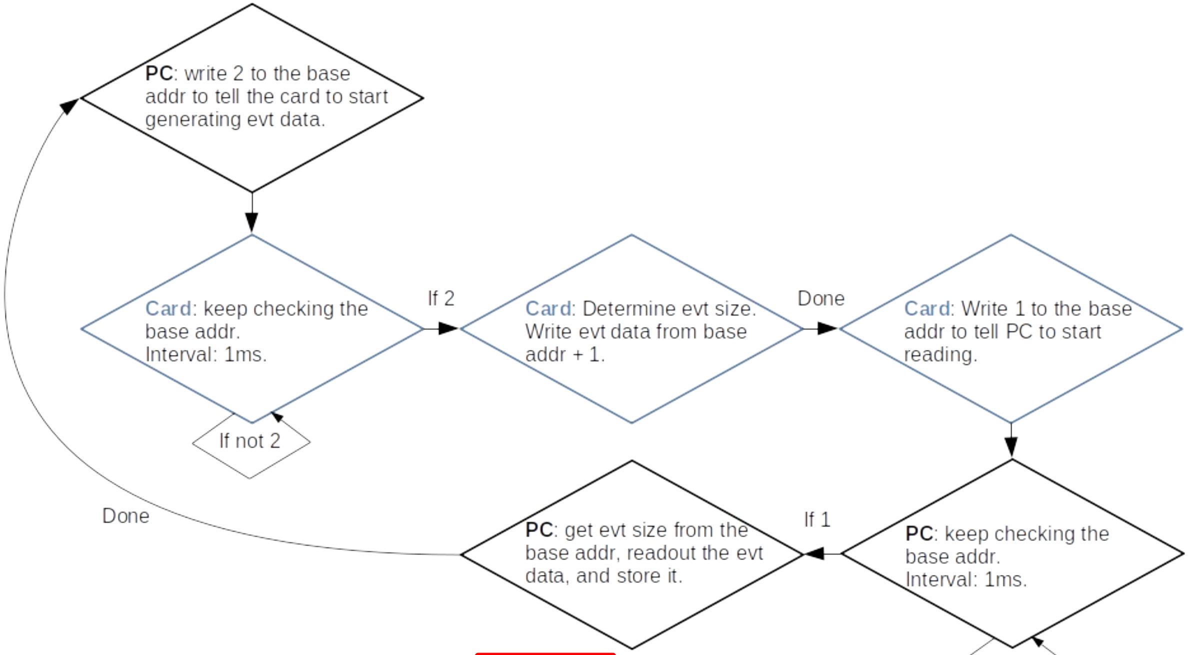


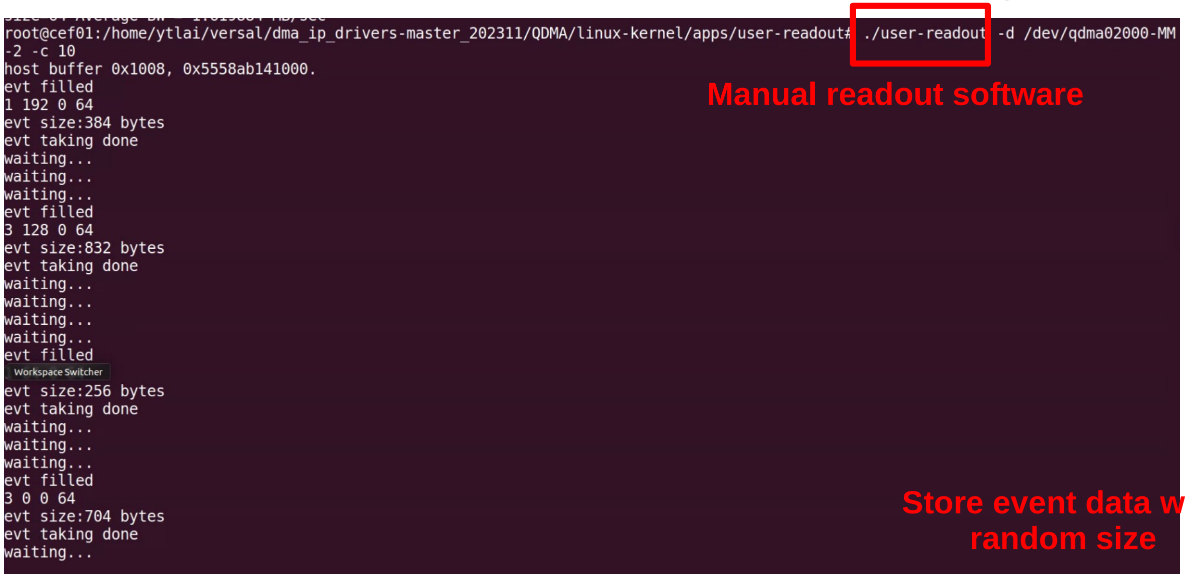
Single-port → True dual-port

User module for protocol and data generation

PCIe-CPM firmware: Event readout using MM mode

- State machine of the readout protocol between PC and FPGA:
- Not fully optimized yet.
- Test:
  - Working correctly.


PC: write 2 to the base addr to tell the card to start generating evt data.
Card: keep checking the base addr. Interval: 1ms.
If not 2
If 2
Card: Determine evt size. Write evt data from base addr + 1.
Done
Card: Write 1 to the base addr to tell PC to start reading.
PC: keep checking the base addr. Interval: 1ms.
If not 1
If 1
PC: get evt size from the base addr, readout the evt data, and store it.
Done

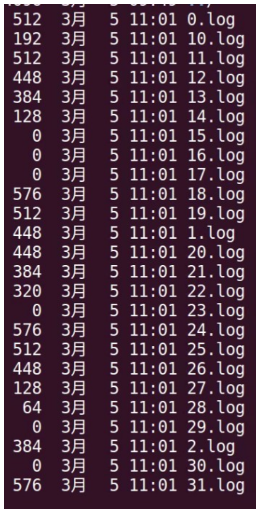


```
root@cef01:/home/ytlai/versal/dma_ip_drivers-master_202311/QDMA/linux-kernel/apps/user-readout# ./user-readout -d /dev/qdma02000-MM
-2 -c 10
host buffer 0x1008, 0x5558ab141000.
evt filled
1 192 0 64
evt size:384 bytes
evt taking done
waiting...
waiting...
waiting...
evt filled
3 128 0 64
evt size:832 bytes
evt taking done
waiting...
waiting...
waiting...
waiting...
evt filled

evt size:256 bytes
evt taking done
waiting...
waiting...
waiting...
evt filled
3 0 0 64
evt size:704 bytes
evt taking done
waiting...
```

Manual readout software

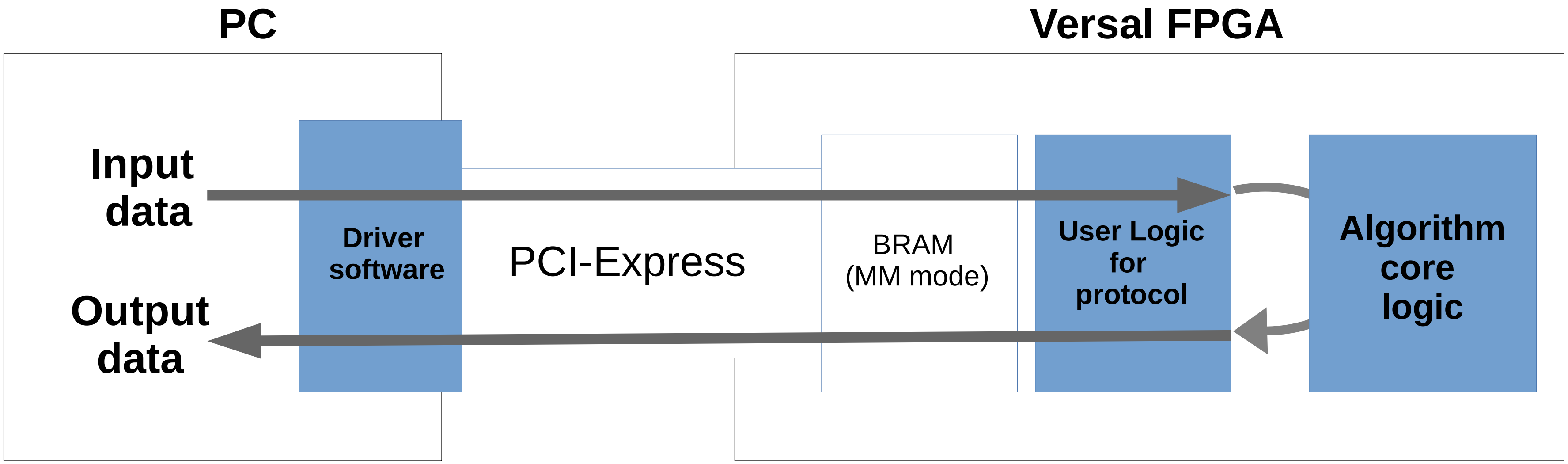
Store event data with random size



```
512  3月   5 11:01 0.log
192  3月   5 11:01 10.log
512  3月   5 11:01 11.log
448  3月   5 11:01 12.log
384  3月   5 11:01 13.log
128  3月   5 11:01 14.log
  0  3月   5 11:01 15.log
  0  3月   5 11:01 16.log
  0  3月   5 11:01 17.log
576  3月   5 11:01 18.log
512  3月   5 11:01 19.log
448  3月   5 11:01 1.log
448  3月   5 11:01 20.log
384  3月   5 11:01 21.log
320  3月   5 11:01 22.log
  0  3月   5 11:01 23.log
576  3月   5 11:01 24.log
512  3月   5 11:01 25.log
448  3月   5 11:01 26.log
128  3月   5 11:01 27.log
 64  3月   5 11:01 28.log
  0  3月   5 11:01 29.log
384  3月   5 11:01 2.log
  0  3月   5 11:01 30.log
576  3月   5 11:01 31.log
```

PCIe-CPM firmware: Event exchange using MM mode

- A data exchange flow is also made for firmware and software.
- 1 event in - 1 event out.
- In order to test the algorithm core logic to be implemented in Versal kits.



Plan

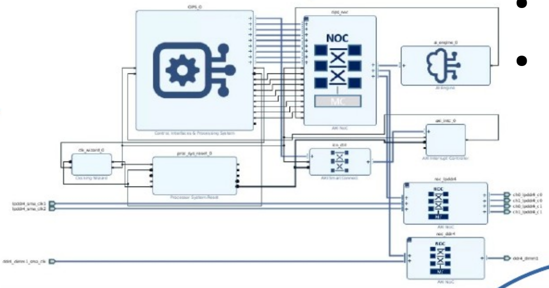
- Further optimize the design and measure the throughput.
- Try to use ST mode: Consulting with Xilinx engineers.

AI engine

- As VCK190 arrived at KEK in 2024 March, we started with learning firmware making to utilize AI engine.
- Our group will prepare a manual for using AI engine for experimental groups' algorithms development.
- Design flow with Vivado/Vitis:
  - C++ programmable design for AI engine.
  - Integrated in PL design.

Vivado

- Open an **example project** for VCK190:
  - Versal Extensible Embedded system. with AI engine.
- Export **platform**



.xsa file

taking this file

Vitis

Workspace

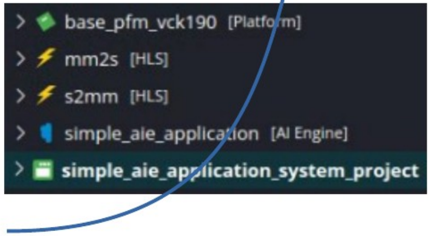
Platform

AI engine component
From example. src, kernel and data.
Algorithm is defined here.

HLS Component: mm2s
with mm2s.cpp

HLS Component: s2mm
with s2mm.cpp

System project Component
for integration

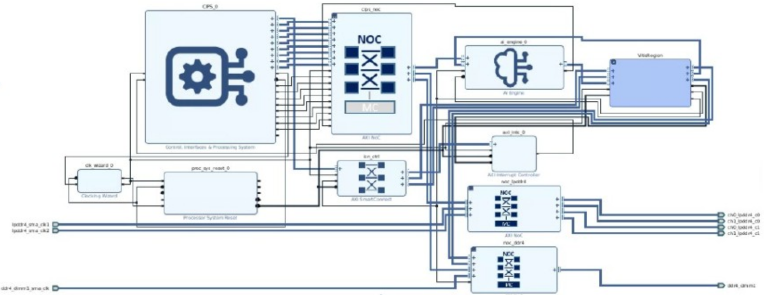


- base_pfm_vck190 [Platform]
- mm2s [HLS]
- s2mm [HLS]
- simple_aie_application [AI Engine]
- simple_aie_application_system_project

building the hardware

Vivado

Get the final **firmware project**
With AI engine, mm2s, s2mm to NOC

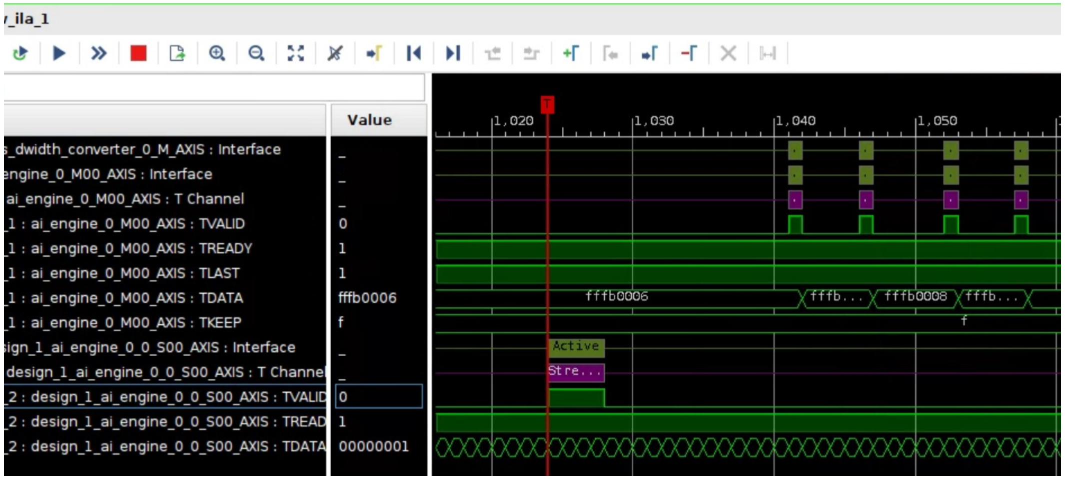


If want to run the system using the Baremetal control application via SD card and UART, need to build up another application in Vitis.

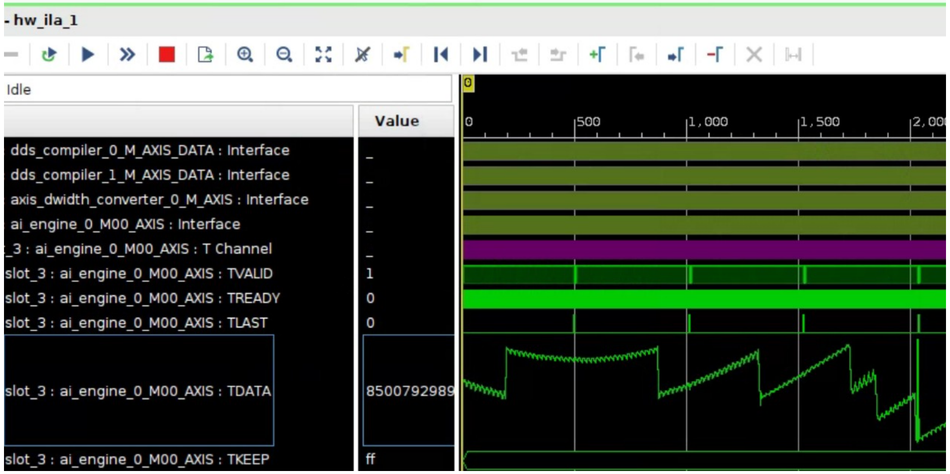
AI engine: test

- The work flow of building up a firmware with AI engine has been studied.
  - PL → AI engine → PL.

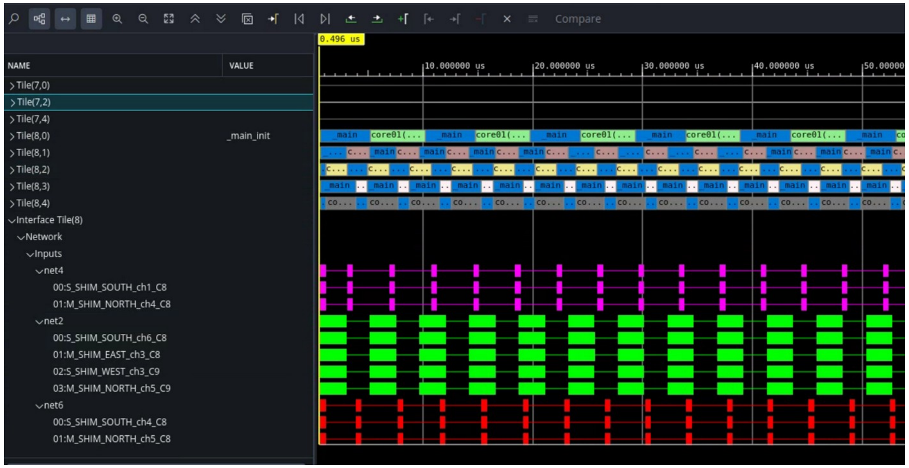
- Some logics were tested.
  - Arithmetic calculation
  - FIR filter
  - leNet



	Value
_dwidth_converter_0_M_AXIS : Interface	_
engine_0_M00_AXIS : Interface	_
ai_engine_0_M00_AXIS : T Channel	_
1 : ai_engine_0_M00_AXIS : TVALID	0
1 : ai_engine_0_M00_AXIS : TREADY	1
1 : ai_engine_0_M00_AXIS : TLAST	1
1 : ai_engine_0_M00_AXIS : TDATA	fffb0006
1 : ai_engine_0_M00_AXIS : TKEEP	f
sign_1_ai_engine_0_0_S00_AXIS : Interface	_
design_1_ai_engine_0_0_S00_AXIS : T Channel	_
2 : design_1_ai_engine_0_0_S00_AXIS : TVALID	0
2 : design_1_ai_engine_0_0_S00_AXIS : TREAD	1
2 : design_1_ai_engine_0_0_S00_AXIS : TDATA	00000001



	Value
dds_compiler_0_M_AXIS_DATA : Interface	_
dds_compiler_1_M_AXIS_DATA : Interface	_
axis_dwidth_converter_0_M_AXIS : Interface	_
ai_engine_0_M00_AXIS : Interface	_
_3 : ai_engine_0_M00_AXIS : T Channel	_
slot_3 : ai_engine_0_M00_AXIS : TVALID	1
slot_3 : ai_engine_0_M00_AXIS : TREADY	0
slot_3 : ai_engine_0_M00_AXIS : TLAST	0
slot_3 : ai_engine_0_M00_AXIS : TDATA	8500792989
slot_3 : ai_engine_0_M00_AXIS : TKEEP	ff

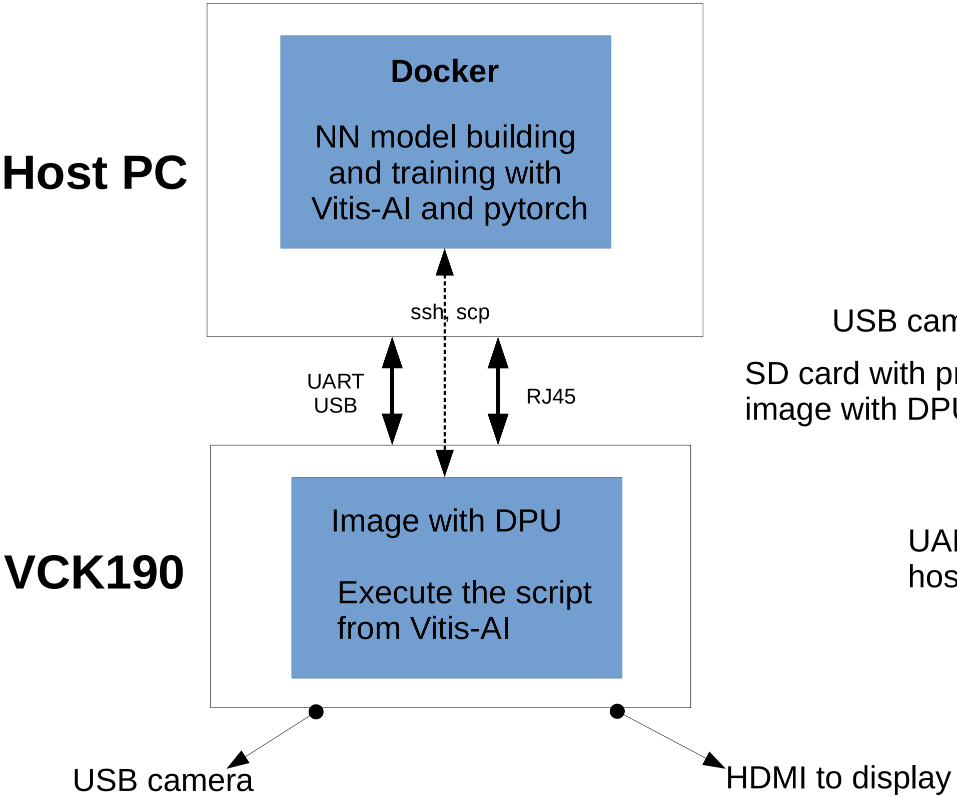


NAME	VALUE
Tile(7,0)	
Tile(7,2)	
Tile(7,4)	
Tile(8,0)	_main_init
Tile(8,1)	
Tile(8,2)	
Tile(8,3)	
Tile(8,4)	
Interface Tile(8)	
Network	
Inputs	
net4	
00:S_SHIM_SOUTH_ch1_C8	
01:M_SHIM_NORTH_ch4_C8	
net2	
00:S_SHIM_SOUTH_ch6_C8	
01:M_SHIM_EAST_ch3_C8	
02:S_SHIM_WEST_ch3_C9	
03:M_SHIM_NORTH_ch5_C9	
net6	
00:S_SHIM_SOUTH_ch4_C8	
01:M_SHIM_NORTH_ch5_C8	

Vitis-AI with DPU

- VCK190 has another feature of Deep Learning Processor Unit (DPU), which is a a configurable computation engine dedicated to convolutional neural networks.

- The design flow does not involve Vivado for PL design. The device is utilized with a small operation system like a server, and works can be executed in it.
  - A higher-level application.



Host PC

Docker

NN model building and training with Vitis-AI and pytorch

ssh, scp

UART USB

RJ45

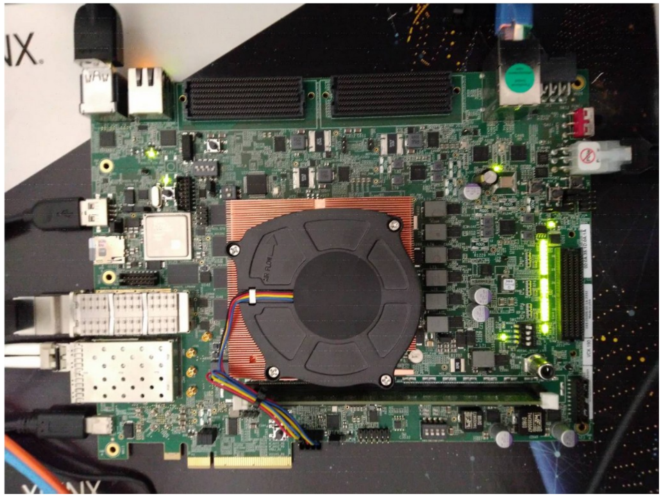
VCK190

Image with DPU

Execute the script from Vitis-AI

USB camera

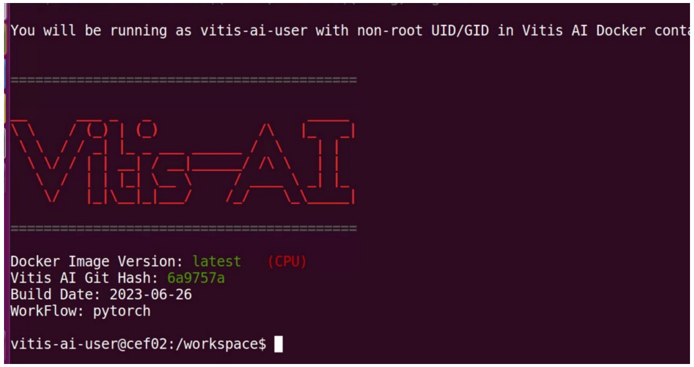
HDMI to display



Vitis-AI with DPU: test

- The environment with docker and DPU setup for VCK190 has been ready.

Vitis-AI within docker



```
You will be running as vitis-ai-user with non-root UID/GID in Vitis AI Docker conta

==========================================

==========================================

Docker Image Version: latest   (CPU)
Vitis AI Git Hash: 6a9757a
Build Date: 2023-06-26
WorkFlow: pytorch

vitis-ai-user@cef02:/workspace$ 
```

Processed image shown on the display

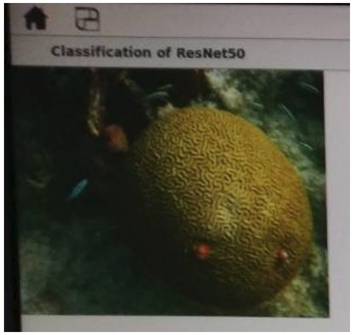
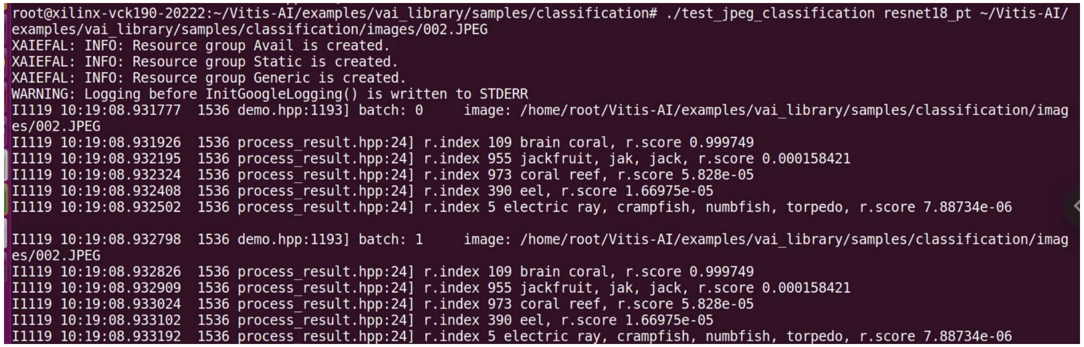


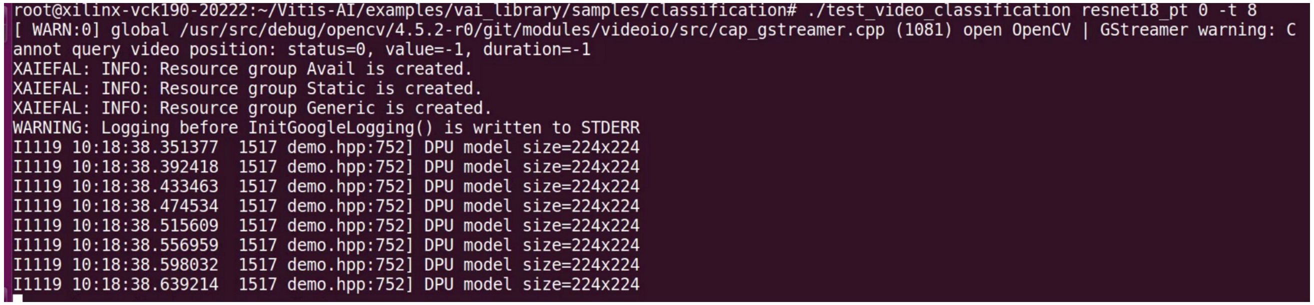
Image processing in DPU



```
root@xilinx-vck190-20222:~/Vitis-AI/examples/vai_library/samples/classification# ./test_jpeg_classification resnet18_pt ~/Vitis-AI/
examples/vai_library/samples/classification/images/002.JPEG
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
WARNING: Logging before InitGoogleLogging() is written to STDERR
I1119 10:19:08.931777  1536 demo.hpp:1193] batch: 0     image: /home/root/Vitis-AI/examples/vai_library/samples/classification/imag
es/002.JPEG
I1119 10:19:08.931926  1536 process_result.hpp:24] r.index 109 brain coral, r.score 0.999749
I1119 10:19:08.932195  1536 process_result.hpp:24] r.index 955 jackfruit, jak, jack, r.score 0.000158421
I1119 10:19:08.932324  1536 process_result.hpp:24] r.index 973 coral reef, r.score 5.828e-05
I1119 10:19:08.932408  1536 process_result.hpp:24] r.index 390 eel, r.score 1.66975e-05
I1119 10:19:08.932502  1536 process_result.hpp:24] r.index 5 electric ray, crampfish, numbfish, torpedo, r.score 7.88734e-06

I1119 10:19:08.932798  1536 demo.hpp:1193] batch: 1     image: /home/root/Vitis-AI/examples/vai_library/samples/classification/imag
es/002.JPEG
I1119 10:19:08.932826  1536 process_result.hpp:24] r.index 109 brain coral, r.score 0.999749
I1119 10:19:08.932909  1536 process_result.hpp:24] r.index 955 jackfruit, jak, jack, r.score 0.000158421
I1119 10:19:08.933024  1536 process_result.hpp:24] r.index 973 coral reef, r.score 5.828e-05
I1119 10:19:08.933102  1536 process_result.hpp:24] r.index 390 eel, r.score 1.66975e-05
I1119 10:19:08.933192  1536 process_result.hpp:24] r.index 5 electric ray, crampfish, numbfish, torpedo, r.score 7.88734e-06
```

Camera video processing in DPU



```
root@xilinx-vck190-20222:~/Vitis-AI/examples/vai_library/samples/classification# ./test_video_classification resnet18_pt 0 -t 8
[ WARN:0] global /usr/src/debug/opencv/4.5.2-r0/git/modules/videoio/src/cap_gstreamer.cpp (1081) open OpenCV | GStreamer warning: C
annot query video position: status=0, value=-1, duration=-1
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
WARNING: Logging before InitGoogleLogging() is written to STDERR
I1119 10:18:38.351377  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.392418  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.433463  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.474534  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.515609  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.556959  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.598032  1517 demo.hpp:752] DPU model size=224x224
I1119 10:18:38.639214  1517 demo.hpp:752] DPU model size=224x224
```

Algorithm making in FPGA: HLS, ML, AI engine

- Next step, we have many algorithms from Belle II, ATLAS, or so, to play in Versal kits.
  - Before that, let's think about the methodologies to do so.

- Considering algorithm implementation:
  - HDL logic in firmware.
  - HLS: software → firmware.
  - ML inference
  - AI engine.

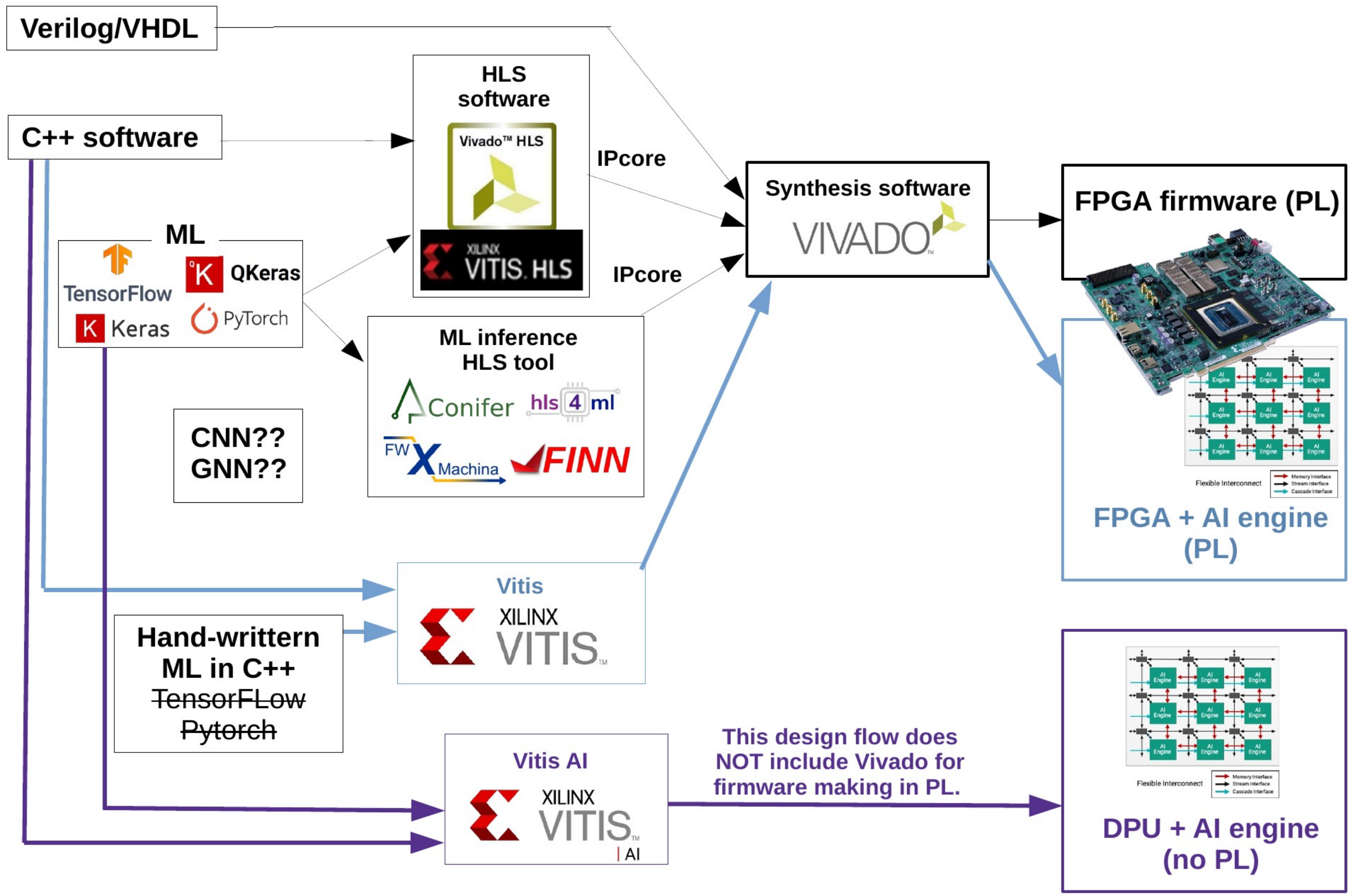
Depend on the different targets, our selection on FPGA differs. A strong FPGA? ACAP with AI engine? DPU?

- Not only the hls4ml, HLS tools has much more for ML and non-ML application.
  - Similarly, Versal AI engine requires a different design flow to make software/firmware.

- For this part of the work, we generalize the work plan into a roadmap in a technical perspective.

HLS, ML, AI engine: roadmap

- As a member of KEK E-sys group, we hope to understand the basic utilization on each, and build a database of such technical knowledge, to support our experimental colleagues.
- We are recruiting young student to learn/work with us.
  - We also plan to make a series of hand-on lecture for each of them.



Verilog/VHDL

C++ software

HLS software

IPcore

IPcore

Synthesis software

FPGA firmware (PL)

ML

CNN??
GNN??

ML inference
HLS tool

Hand-writtern
ML in C++
~~TensorFLow~~
~~Pytorch~~

Vitis

Vitis AI

This design flow does
NOT include Vivado for
firmware making in PL.

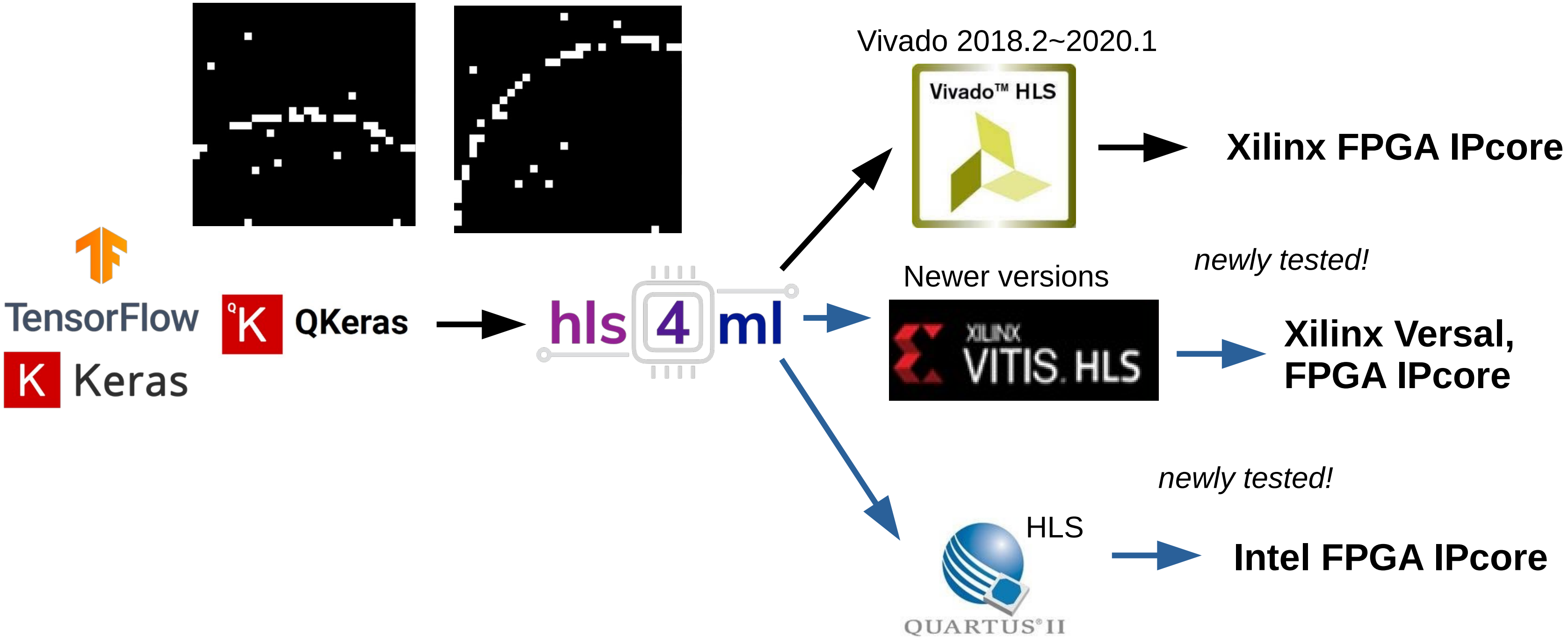
FPGA + AI engine
(PL)

DPU + AI engine
(no PL)

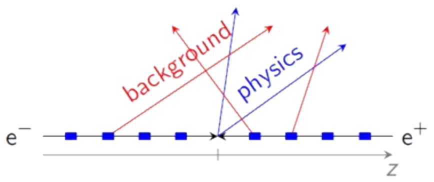
hls4ml

- hls4ml: A package for machine learning inference in FPGA.
  - Already lots of utilizations with Vivado HLS in Belle II and ATLAS.

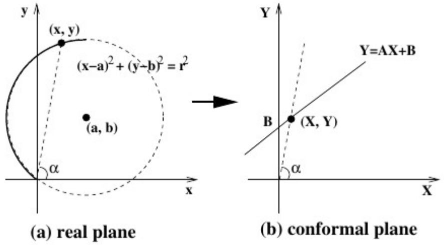
- Yiyang Ding, our summer internship student in 2023, performed general studies on it.
  - A NN model for simple tracker and tested with VPK120!
  - Also tested with Intel FPGA with Quartus.
  - A manual has been prepared.

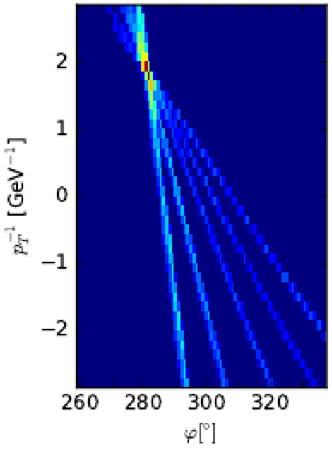


Next step: what kinds of algorithms to implment?

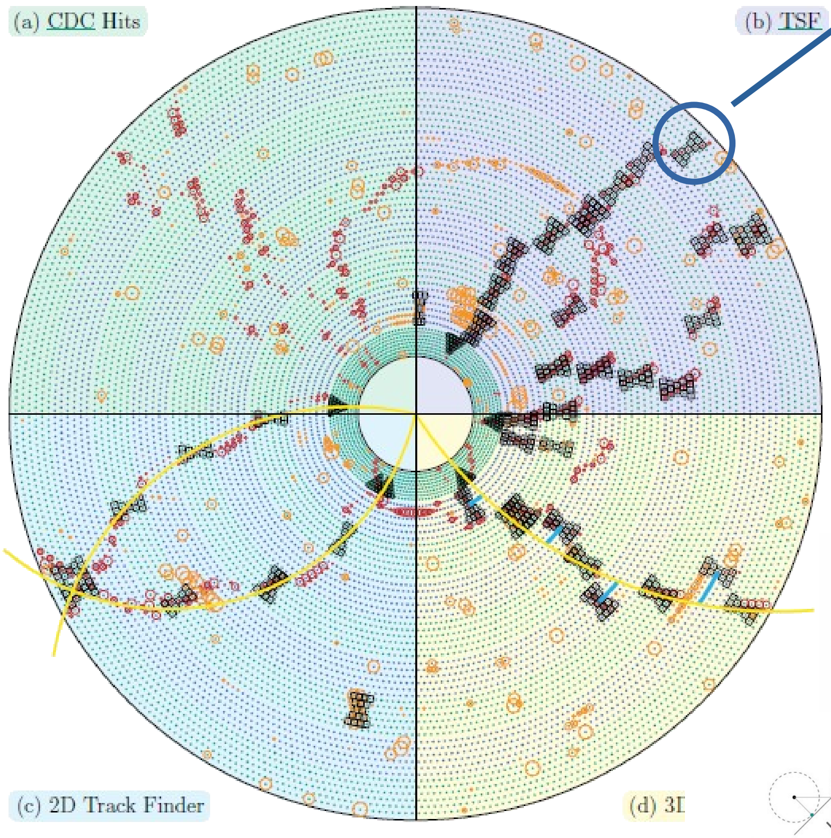


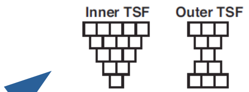
2D Track Finding:
Hough transformation.
Limited track condition.



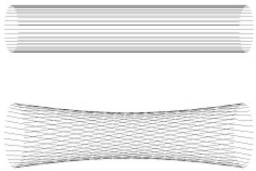


Example: Belle II track Trigger

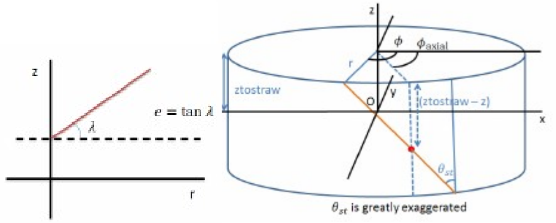




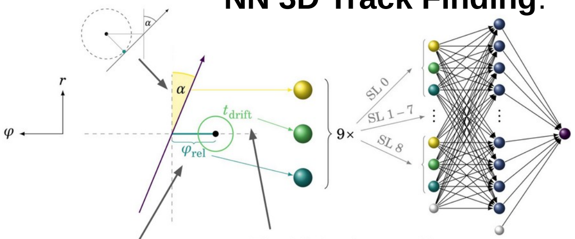
Track Segment:
Simplification on the algorithm



3D Track Finding:
Fitting with Stereo wire info



NN 3D Track Finding:

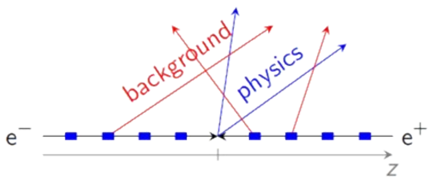


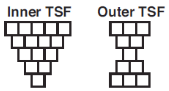
Delta φ relative to the 2DTrack. For Axial TS this is 0.

The drift time has a positive sign, when the track passed on the right side, and a negative sign if it passed on the left.

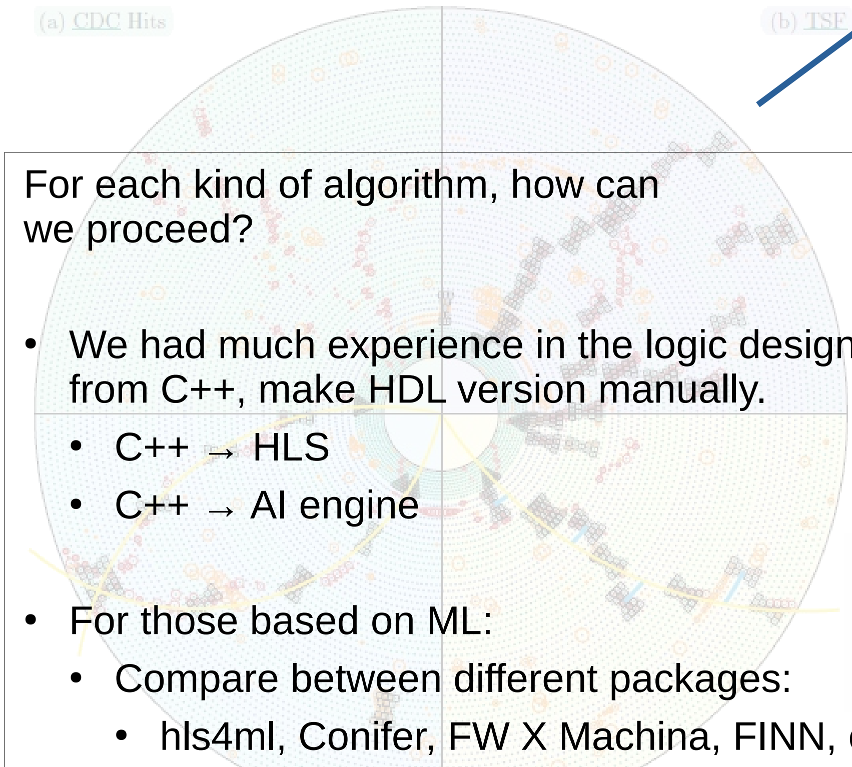
Next step: how to implment?

Example: Belle II track Trigger





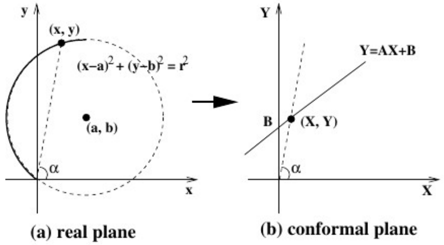
Track Segment:
Simplification on the algorithm

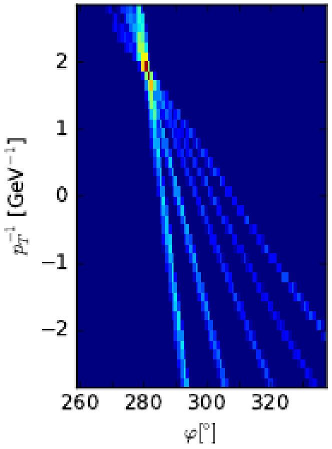


For each kind of algorithm, how can we proceed?

- We had much experience in the logic design from C++, make HDL version manually.
  - C++ → HLS
  - C++ → AI engine
- For those based on ML:
  - Compare between different packages:
    - hls4ml, Conifer, FW X Machina, FINN, etc
  - Manual constructed NN at AI engine

2D Track Finding:
Hough transformation.
Limited track condition.

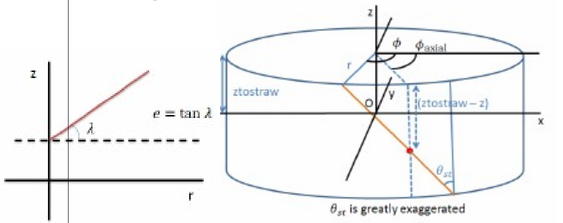




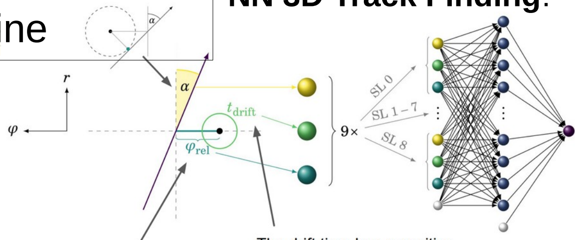
Axial

Stereo

3D Track Finding:
Fitting with Stereo wire info



NN 3D Track Finding:



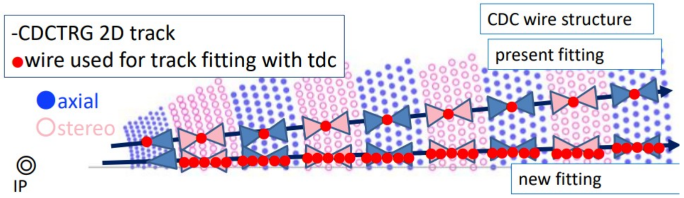
Delta φ relative to the 2DTrack. For Axial TS this is 0.

The drift time has a positive sign, when the track passed on the right side, and a negative sign if it passed on the left.

Prospect: more new ideas

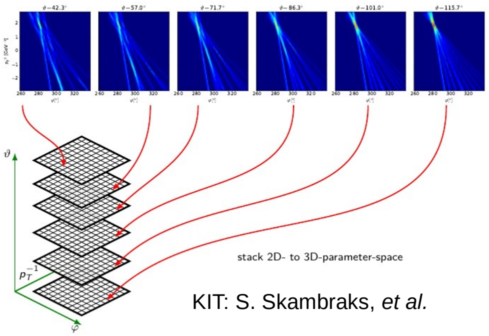
Additional dimension: More resource in FPGA

Extension of more input info:



KEK: T. Koga, *et al*.

2D→3D Hough tracking:

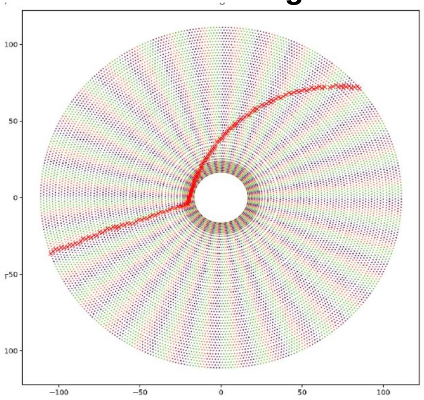


KIT: S. Skambraks, *et al.*

More than NN: CNN or GNN?

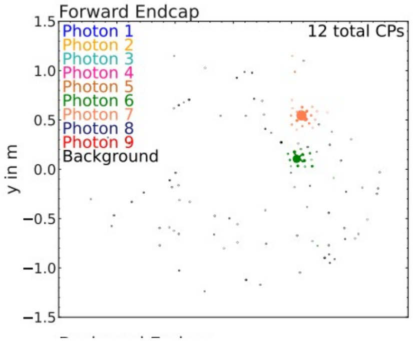
KIT, TUM, MPI: Belle II AI trigger group

CNN tracking

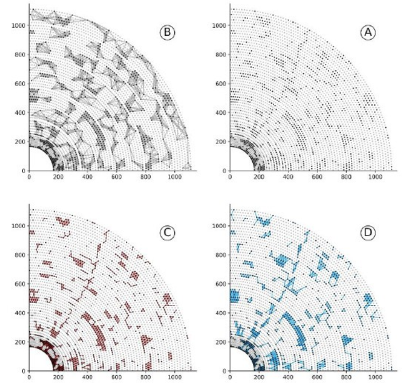


■ Aglining methode ploted on the CDC cross-section

GNN clustering



GNN tracking

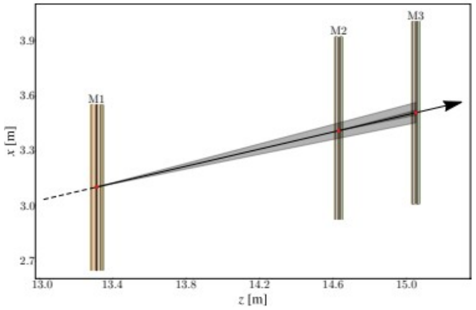


Dataset: displaced_processed_simulated_2_tracks_0_nominal-phase3

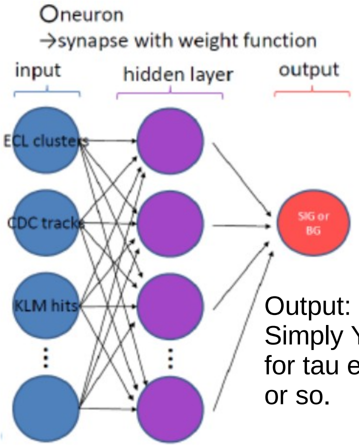
NN: hls4ml

ATLAS fast muon tracking with Neural Network:

arXiv:2202.04976



Neural Network for τ event trigger:



Output:
Simply Y/N
for tau event
or so.

KEK: T. Koga, R. Nomaru.

Summary

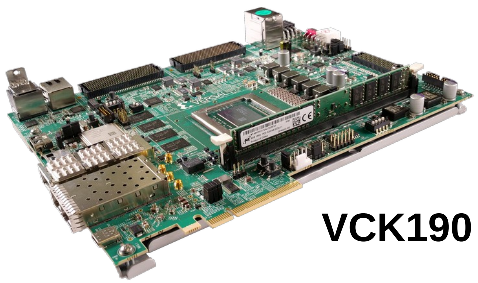
- The Collider Electronics Forum at KEK IPNS and Japanese HEP community started a project using the evaluation kits of the Xilinx Versal ACAP targeting on the future R&D of a new universal FPGA device.

- Some of the fundamental functionalities of the Versal evaluation kits have been studied.
  - Firmware making, high-speed transmission, PCIe, AI engine, DPU, and HLS for ML inference.

- Future plan:
  - More basic studies on HLS tools, ML inference packages, and AI engine will be performed.
  - Implement different physics algorithms for different experiments.
  - We will also discuss about the new device's R&D plan.
    - The next generation of Universal Trigger board (UT5).

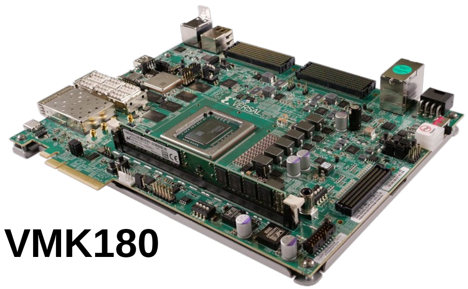
Backup

Evaluation kits for Versal



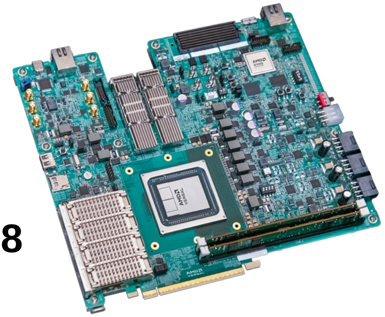
VCK190

- Features the VC1902 Versal AI Core series
- For using AI and DSP engines with greater compute performance that current server class CPUs



VMK180

- Features the VM1802 Versal™ Prime series
- The world’s first ACAP
- A software programmable infrastructure and connectivity

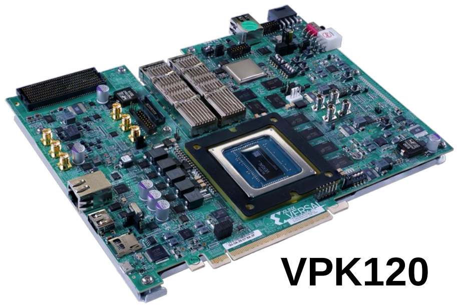


VHK158

- Features the VH1582 Versal™ HBM series
- convergence of memory, compute, and connectivity with 32G HBM and 112G PAM4

VCK5000

- Features the Versal AI Core Series
- For (AI) Engine development with Vitis and AI Inference development
- Not flexible for FPGA firmware



VPK120

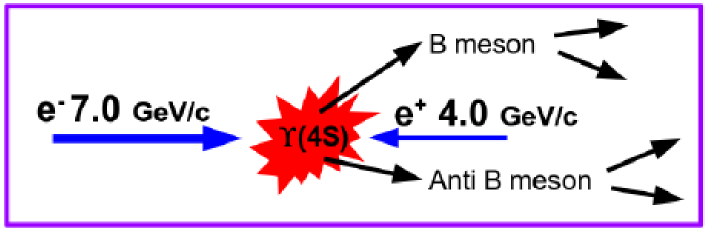
- Features Versal™ Premium series VP1202
- Multiple high-speed connectivity option
- Massive serial bandwidth, security, and compute density

VEK280

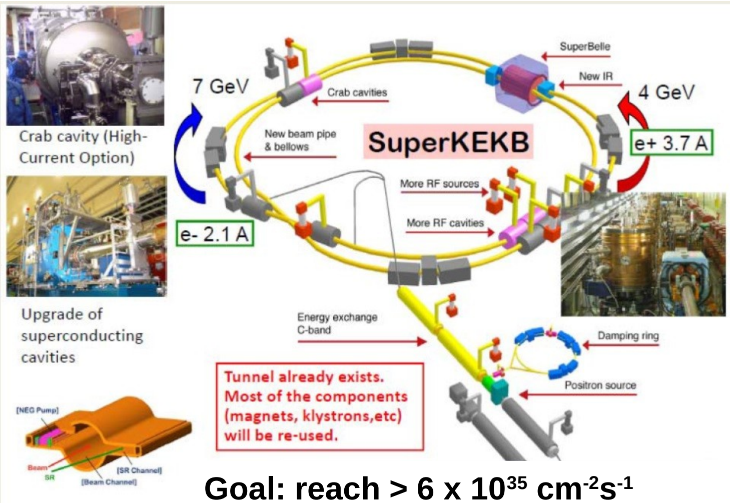
- Features the VE2802 Versal AI Edge series
- Simpler version of VCK190
- Will come out in 2024

SuperKEKB

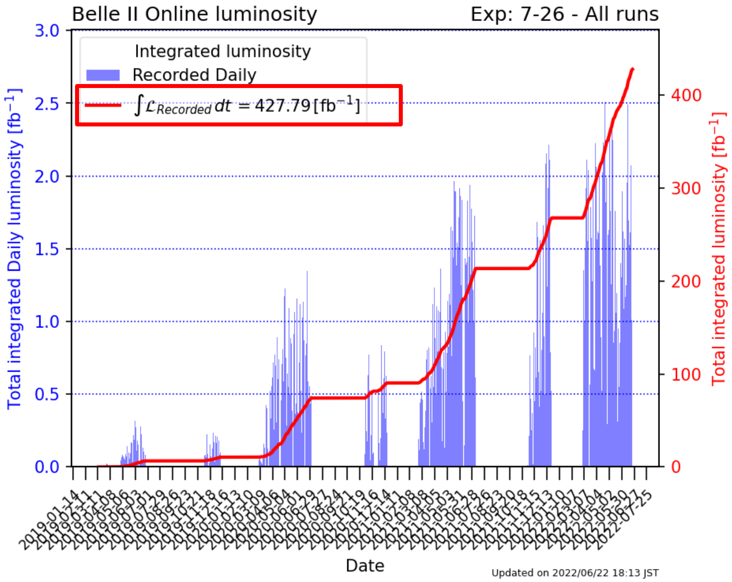
- SuperKEKB: Upgraded from KEKB.
  - More than 30 times larger luminosity of KEKB with nano beam scheme.
- Asymmetric energy collider:
  - 7.0 GeV e^- and 4.0 GeV e^+ for Y(4S) $\rightarrow B\bar{B}$.



- Luminosity achievement:
  - L_{peak} = 4.65 x 10^{34} $cm^{-2}s^{-1}$.
    World record. ~Two times of KEKB record with much smaller beam current.
  - L_{int} = ~427 fb^{-1} up to Jun. 2022.
- Will resume beam collision in 2024 with PXD full installation.

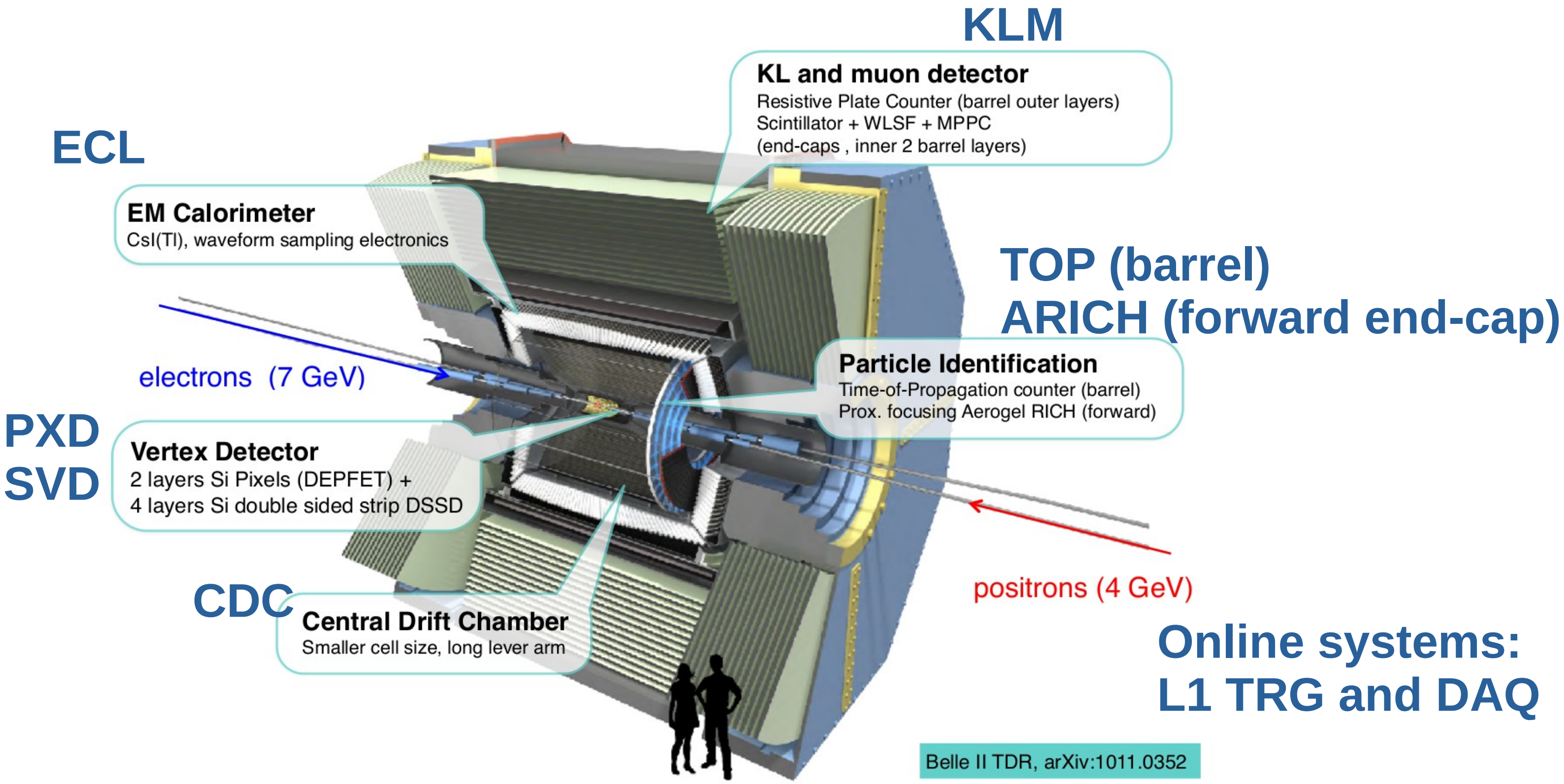


Goal: reach > 6 x 10^{35} $cm^{-2}s^{-1}$



Belle II detector

- Belle II: Newly-designed sub-detectors set to improve detection performance.



KLM

KL and muon detector
Resistive Plate Counter (barrel outer layers)
Scintillator + WLSF + MPPC
(end-caps , inner 2 barrel layers)

ECL

EM Calorimeter
CsI(Tl), waveform sampling electronics

TOP (barrel)
ARICH (forward end-cap)

Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (forward)

electrons (7 GeV)

PXD
SVD

Vertex Detector
2 layers Si Pixels (DEPFET) +
4 layers Si double sided strip DSSD

CDC

Central Drift Chamber
Smaller cell size, long lever arm

positrons (4 GeV)

Online systems:
L1 TRG and DAQ

Belle II TDR, arXiv:1011.0352

- Physics target of Belle II:
  - Rare B, τ, charm physics, Dark Matter search, CP Violation.

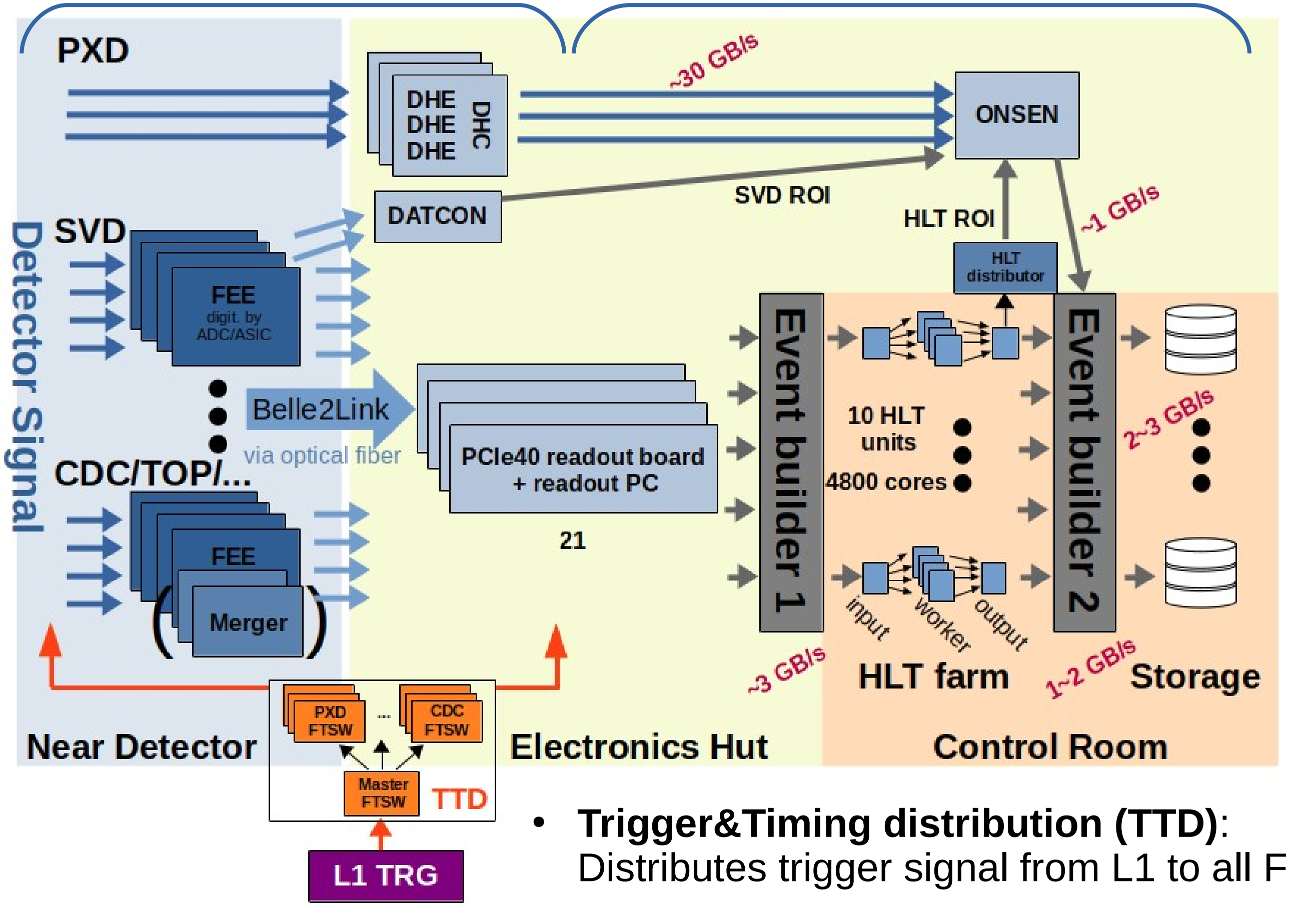
- Requirement for data taking:
  - High L1 trigger rate (~30 kHz), high background, and large event size.

Belle II DAQ system

- Pipeline common readout system for each sub-detector.
  - Except for PXD: data reduction system with ROI.
- Target of performance: 30 kHz L1 rate, ~1% of dead time, and a raw event size of 1 MB.

- **FPGA (FEE - Readout)**: Use universal "Belle2Link" protocol with optical links in between.

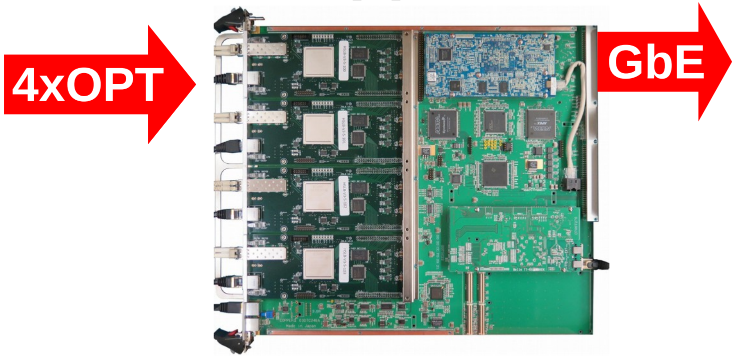
- **Back-end servers**: readout PC, HLT and storage for online procession/reconstruction.



- **Trigger&Timing distribution (TTD)**: Distributes trigger signal from L1 to all FPGA.

Readout device and its upgrade

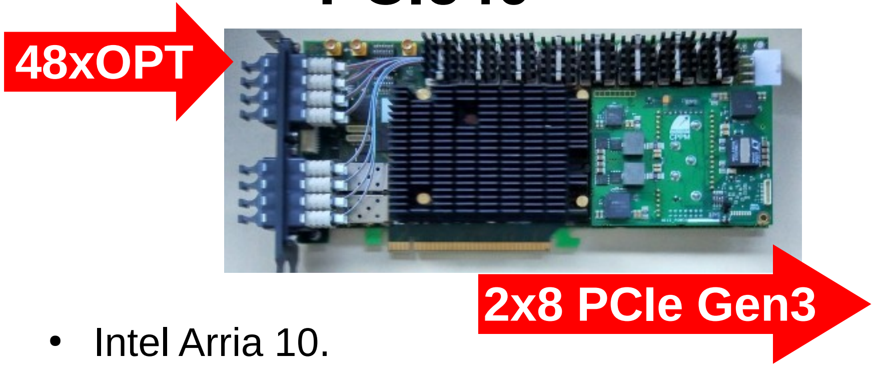
Copper



- 4 Xilinx Virtex-5 receiver boards.
- PrPMC: data procession, pre event building.
- In total 203 coppers were used in Belle II.

Upgrade

PCIe40



- Intel Arria 10.
- Developed in LHCb and ALICE.
- 48 optical links.
- 2x8 PCIe Gen3.
- In total 21 PCIe40 boards will be used in Belle II.

Considerations for upgrade:

- Difficulty of maintenance:
  - Increasing number of malfunctioning pieces.
  - Many different boards in system.
  - Parts out of production already.
- Limit of the system on further improvement:
  - Output throughput by GbE: 1Gbps.
  - CPU usage: ~60% at 30 kHz trigger rate.

Belle II TRG

- 4 sub-trigger systems + 2 global trigger systems.

KEK, NTU, FJU, NUU, KIT, TUM, MPI, KU, KMI Nagoya, U. of Tokyo

Charged tracking: Track parameter, number of tracks, event timing.

Hanyang U., BINP, Notice co.

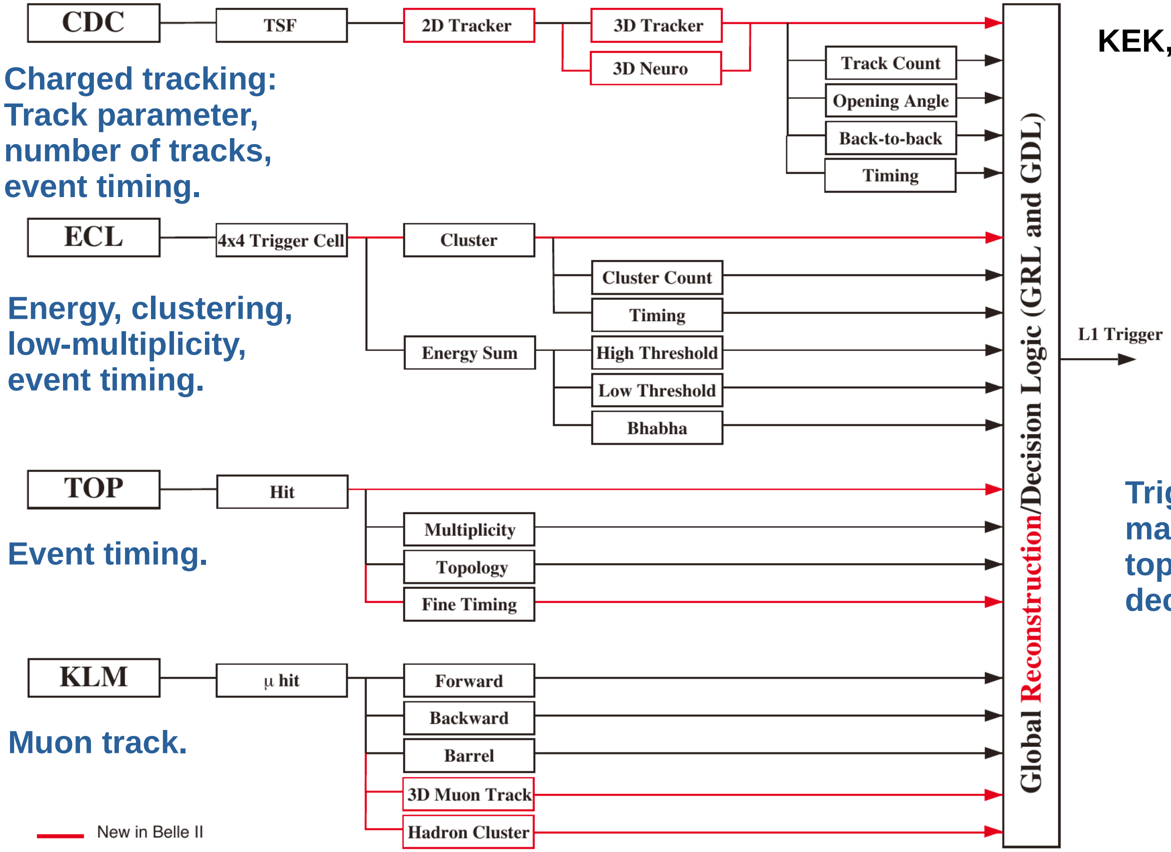
Energy, clustering, low-multiplicity, event timing.

U. Pittsburgh, Hawaii U.

Event timing.

Virginia Tech., Hawaii U.

Muon track.



KEK, NTU, NCU

Trigger summary, matching, topological, final decision.

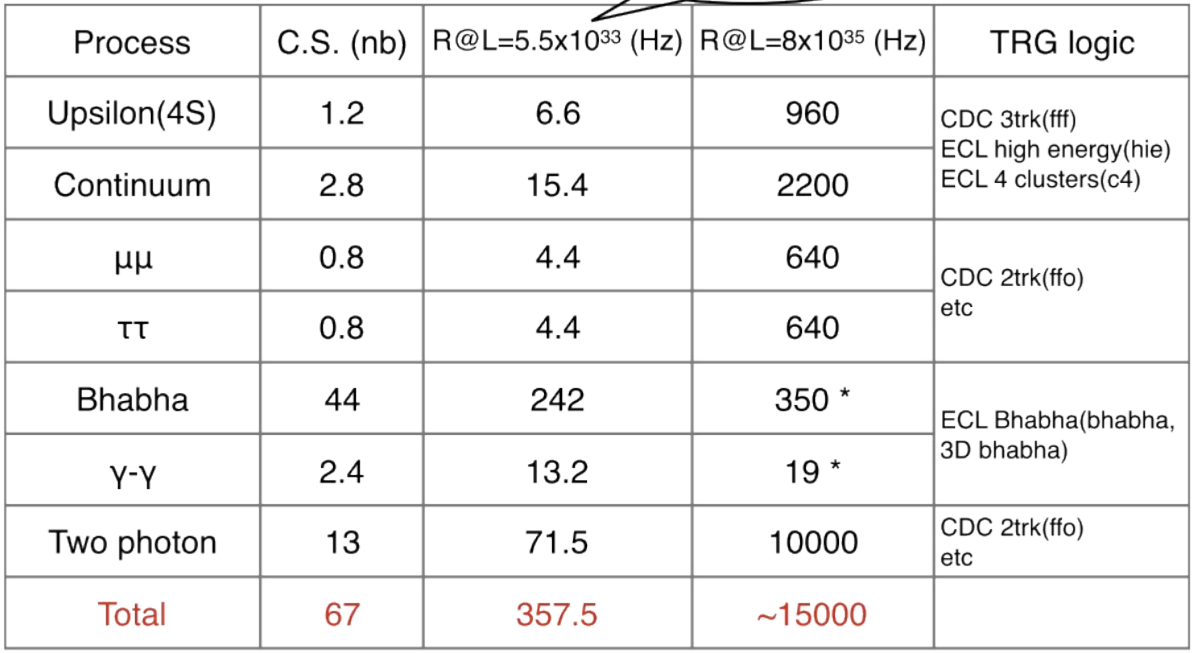
Conditions and requirements for TRG

- Requirements:
  - Overall latency < 4.4 μs.
  - ~100% eff. for hadronic events.
  - Max 30 kHz @ 8*10^{35} $cm^{-2}s^{-1}$
  - Timing precision: < 10 ns
  - Event separation: 500 ns

- Examples of technical challenges so far:
  - Low-multiplicity trigger mainly based on ECL, but contamination from noise, beam bkg or Bhabha.
  - Energy trigger with high eff. but high rate too.
  - Injection bkg.
  - Drawback of track trigger at endcap.
  - High track trigger rate due to crosstalk noise.
  - Latency budget due to transmission or complicated logics.
  - ......

- Physics processes in interest:

Phase2 Lum. Record



Process	C.S. (nb)	R@L=5.5x10^{33} (Hz)	R@L=8x10^{35} (Hz)	TRG logic
Upsilon(4S)	1.2	6.6	960	CDC 3trk(fff) ECL high energy(hie) ECL 4 clusters(c4)
Continuum	2.8	15.4	2200	
μμ	0.8	4.4	640	CDC 2trk(ffo) etc
ττ	0.8	4.4	640	
Bhabha	44	242	350 *	ECL Bhabha(bhabha, 3D bhabha)
γ-γ	2.4	13.2	19 *	
Two photon	13	71.5	10000	CDC 2trk(ffo) etc
Total	67	357.5	~15000	

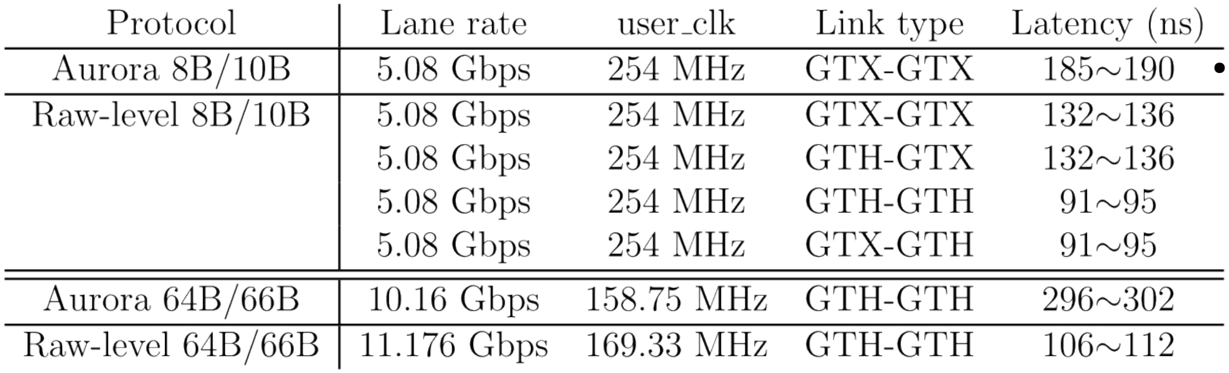
Data transmission protocol at Belle II TRG

- Data transmission in TRG: Xilinx and Altera FPGA MGT, QSFP module, and MPO cable.

- The original plan was to use the open-source Aurora protocol, but large latency was introduced and exceeded the L1 limit (4.4 μs).

- Belle II CDCTRG developed an user-defined transmission protocols:
  - Smaller latency than Aurora's: **Latency reduction is critical for L1!**
  - User-friendly interface.
  - 8B/10B and 64B/66B encoding.
  - Support various Xilinx and Altera MGT.
  - Bit error rate < 10^{-18} /s with few weeks BERT.
  - Flow control and synchronization.

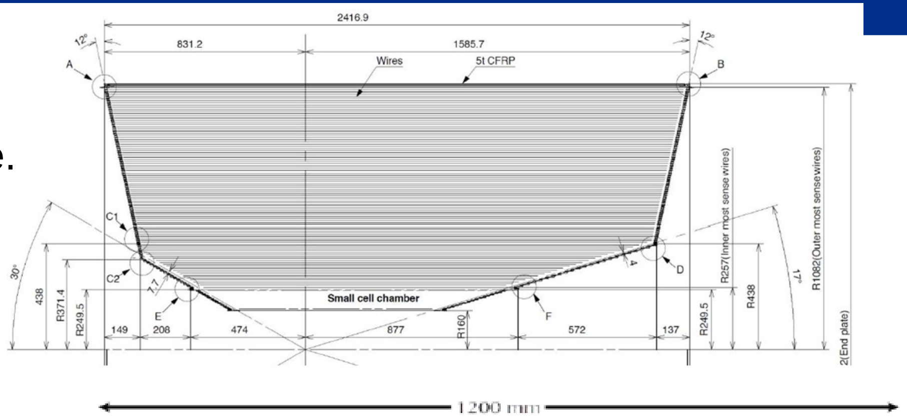
Latency comparison using UT3 (Virtex-6 GTX and GTH)



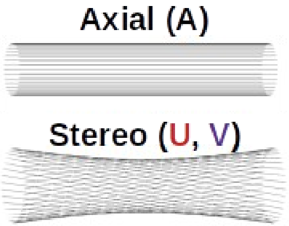
Protocol	Lane rate	user_clk	Link type	Latency (ns)
Aurora 8B/10B	5.08 Gbps	254 MHz	GTX-GTX	185~190
Raw-level 8B/10B	5.08 Gbps	254 MHz	GTX-GTX	132~136
	5.08 Gbps	254 MHz	GTH-GTX	132~136
	5.08 Gbps	254 MHz	GTH-GTH	91~95
	5.08 Gbps	254 MHz	GTX-GTH	91~95
Aurora 64B/66B	10.16 Gbps	158.75 MHz	GTH-GTH	296~302
Raw-level 64B/66B	11.176 Gbps	169.33 MHz	GTH-GTH	106~112

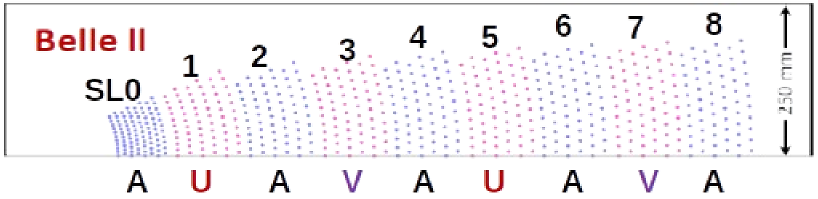
- For **UT4**:
  - Up to 25 Gbps using 64B/66B.
  - Latency: ~ 50ns.

Track trigger with CDC



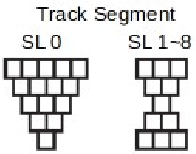
- ~14K sense wires with mixture of He and ethane.
- An alternative **AUAVAUAVA** wire configuration for 3D information:
  A: Axial super-layer (SL) parallel to z-axis
  U, **V**: Stereo SL with two small stereo angles.

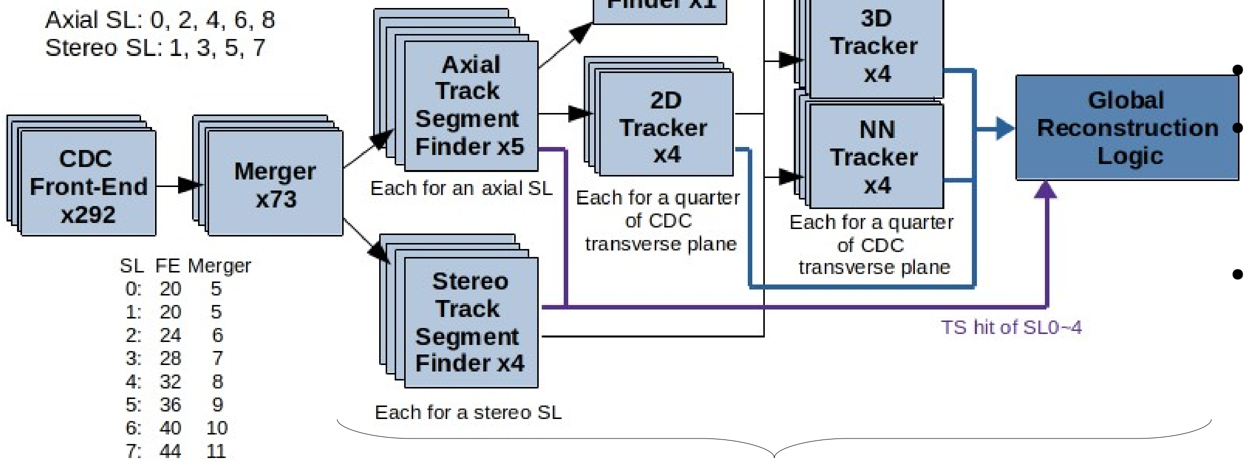




Fonrt-End
Xilinx Virtex-5

Merger
Altera Arria2





SL	FE	Merger
0:	20	5
1:	20	5
2:	24	6
3:	28	7
4:	32	8
5:	36	9
6:	40	10
7:	44	11
8:	48	12

Universal Trigger board: UT3 and UT4

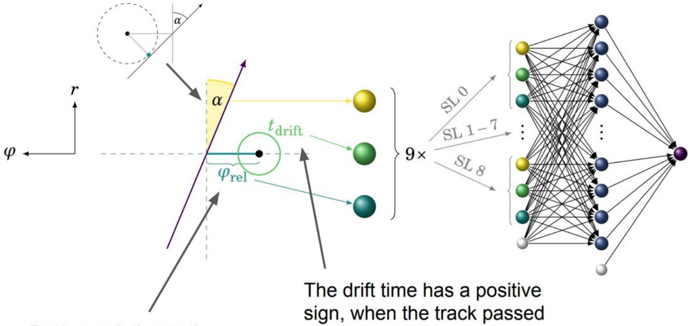
- Track Segment Finder (TSF).
- Tracker: 2D full track, 3D, Neural 3D (NN), and short tracker.
- Event Timing Finder (ETF)

Neural z trigger

Karlsruher Institut für Technologie

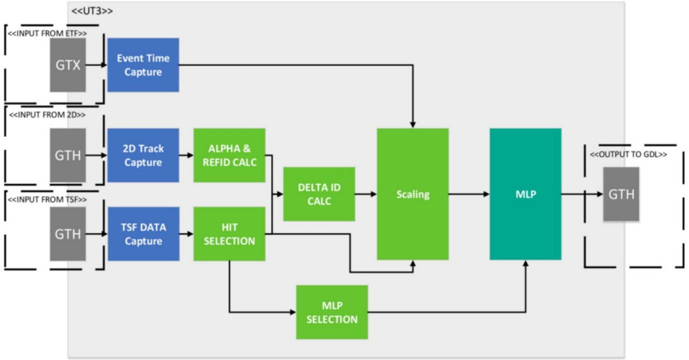
S. Neuhaus et al 2015 J. Phys.: Conf. Ser. 608 012052
Kai Lukas Unger et al 2023 J. Phys.: Conf. Ser. 2438 012056
F. Meggendorfer, DPG Conference 2021
Thesis: S. Skambraks, S. Pohl

- In addition to the conventional 3D tracker based on fitting method, Belle II has a Neural Network 3D tracker (NN) running in parallel in the system.
- Input the 2D tracker and stereo TS info: Crossing angle, drift time, φ relative to 2D Track .
  - Obtain z_0 and θ.



Delta φ relative to the 2DTrack. For Axial TS this is 0.

The drift time has a positive sign, when the track passed on the right side, and a negative sign if it passed on the left.


<<UT3>>

<<INPUT FROM ETF>>

GTX

Event Time Capture

<<INPUT FROM 2D>>

GTH

2D Track Capture

ALPHA & REFID CALC

<<INPUT FROM TSF>>

GTH

TSF DATA Capture

HIT SELECTION

DELTA ID CALC

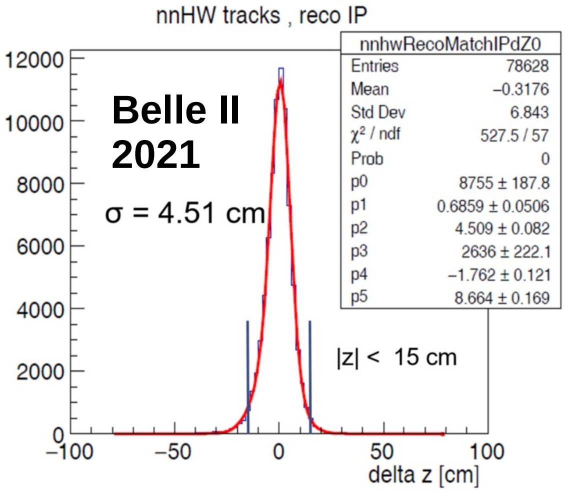
Scaling

MLP

MLP SELECTION

<<OUTPUT TO GDL>>

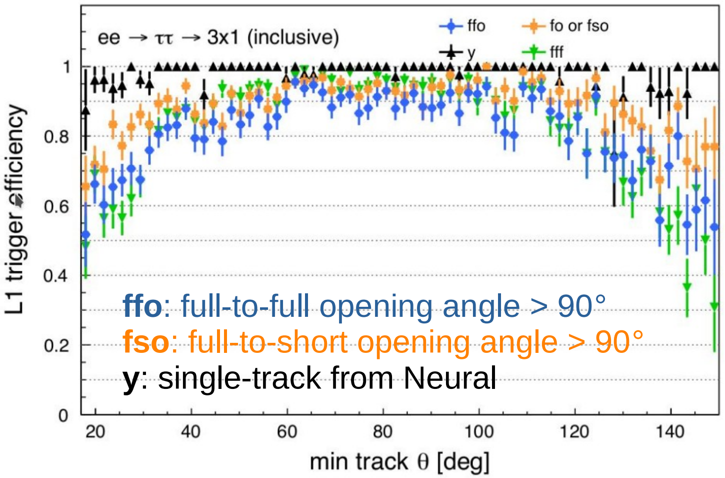
GTH



Belle II 2021

σ = 4.51 cm

|z| < 15 cm



ffo: full-to-full opening angle > 90°

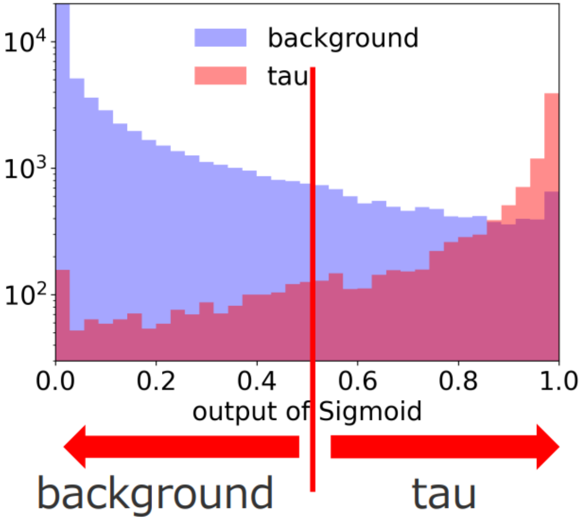
fso: full-to-short opening angle > 90°

y: single-track from Neural

Plots by P.Rados, A.Rostomyan (DESY)

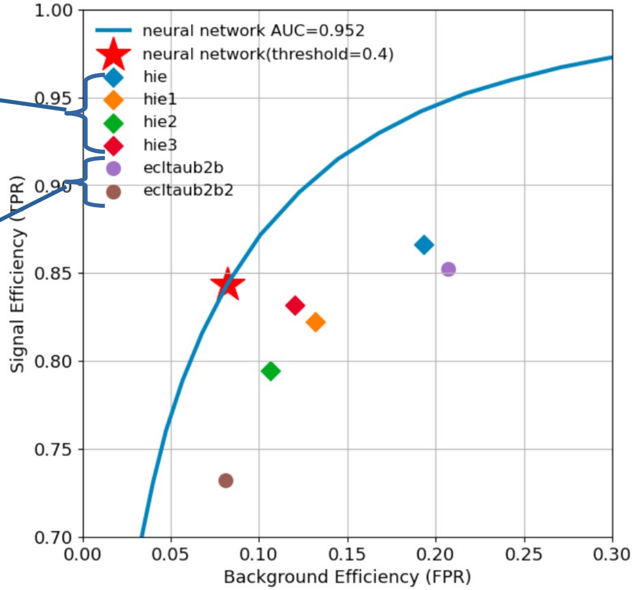
ML tau trigger

- Global trigger receives the cluster information from ECLTRG.
  - Input the position and energy information of clusters to a Neural Network, and determine if it is a tau event or not.
  - A kind of topological application.
  - Based on hls4ml.
  - Validated and will be implemented in 2024 runs.



hie: ECL energy sum

ecltaub2b: ECL cluster based logic


neural network AUC=0.952
neural network(threshold=0.4)
hie
hie1
hie2
hie3
ecltaub2b
ecltaub2b2
Signal Efficiency (TPR)
Background Efficiency (FPR)