Research projects in Collider Electronics Forum of KEK ITDC and developments about AI/ML with FPGA

Yun-Tsung Lai

on behalf of the CEF managers

KEK IPNS

ytlai@post.kek.jp

2025 Belle II TRGDAQ workshop

24th Oct., 2025

Outline

- Introduction on Collider Electronics Forum
- Versal project
  - 1. Hardware fundamental functionalities
  - 2. FPGA algorithm techniques: HLS, ML, AIE, DPU
  - 3. Real application and R&D for hardware device/system
- AI in FEE project
- Summary

Introduction to Collider Electronics Forum (CEF)

- Collide Electronics Forum (CEF):
  - Within Instrumentation Technology Development Center (ITDC) of KEK IPNS.
  - Motivated to provide a platform of common development on the electronics devices with new technologies for future collider experiments.
  - Established by M. Tomoto-san, M. Tanaka-san and Y. Ushiroda-san in 2022, mainly within E-sys, Belle II, and Energy Frontier groups of KEK IPNS.
  - Research proposal from each group can be made and discussed.
    Then the works of the project will be shared with the members in the forum.

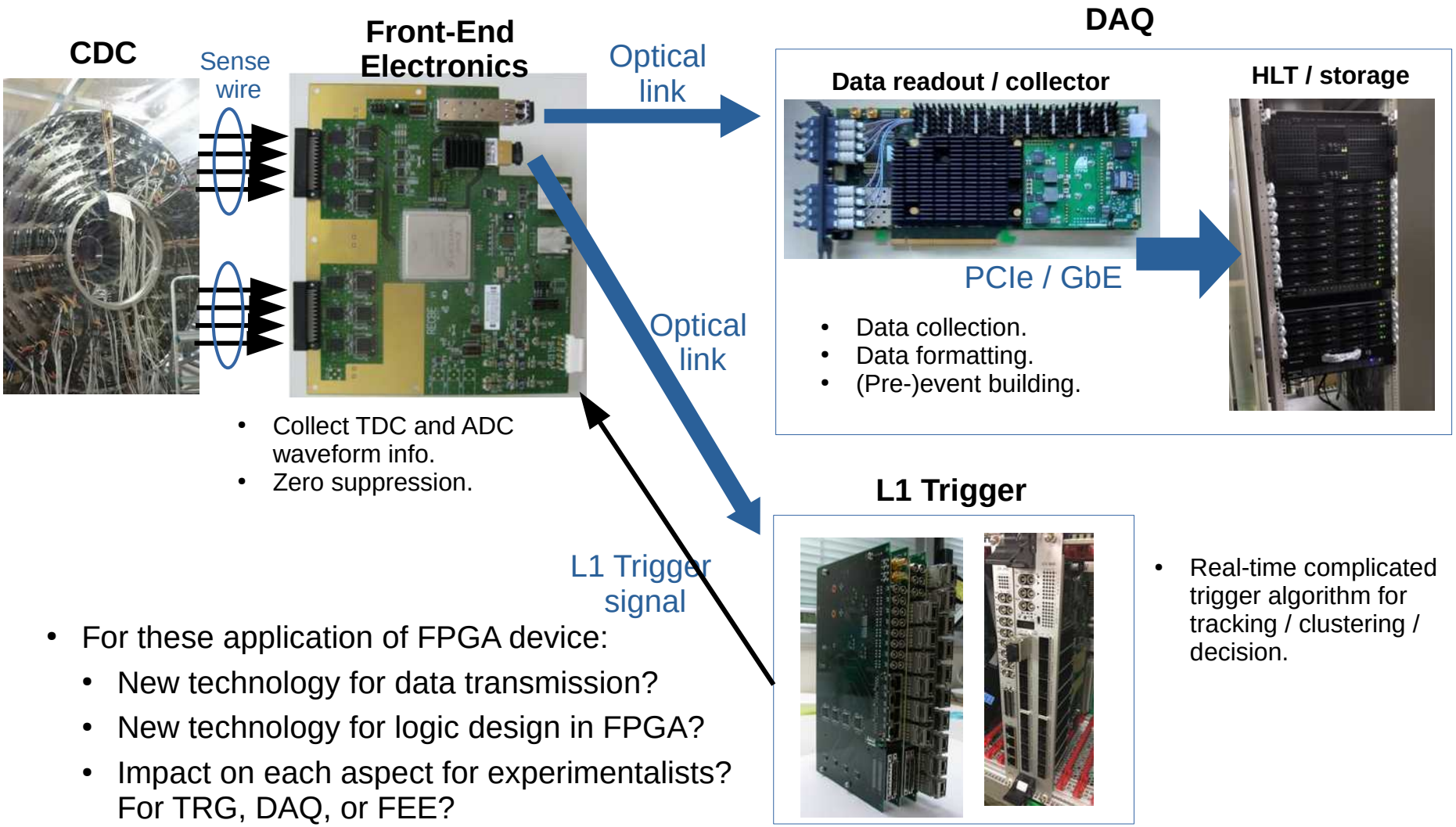
- We will also promote the collaboration with other experimental groups.
  - Communication with SPADI-A: Unified DAQ system design for nuclear experiments.

- Our activities can be found at https://kds.kek.jp/category/2369/

- Core members: from ATLAS, E-sys, Belle II and ALICE/EIC
  - ATLAS: M. Tomoto (KEK), K. Nagano (KEK), J. Maeda (Kobe), Y. Horii (Nagoya)
  - E-sys: R. Honda, M. Tanaka, Y.-T. Lai
  - Belle II: T. Koga, S. Yamada, Y. Ushiroda (KEK)
  - ALICE/EIC: T. Gunji (CNS)

Application of FPGA in HEP experiments

- Belle II Central Drift Chamber (CDC):

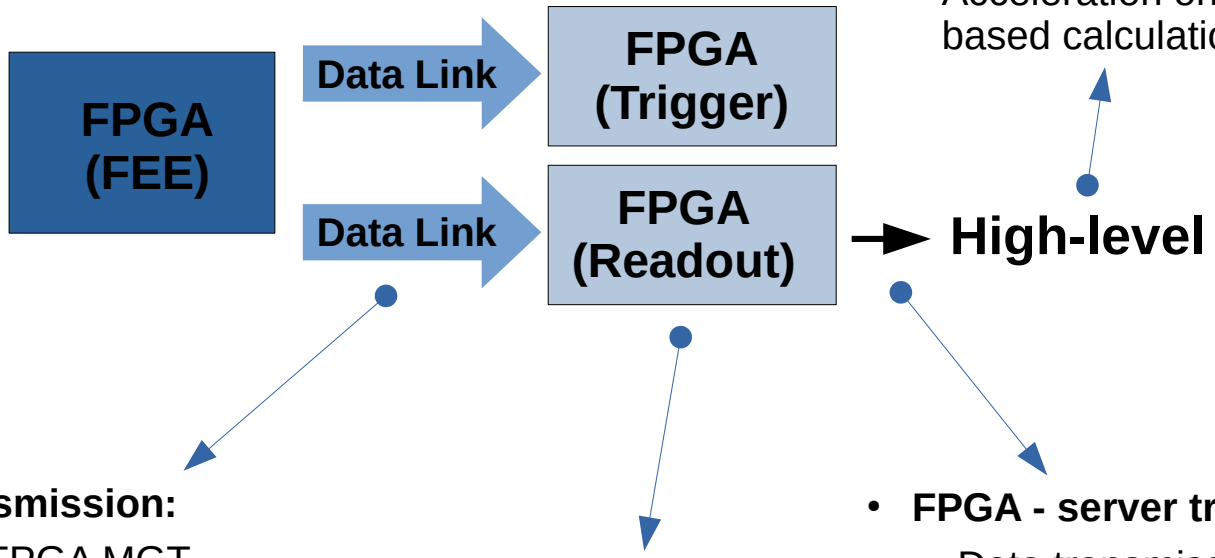


- For these application of FPGA device:
  - New technology for data transmission?
  - New technology for logic design in FPGA?
  - Impact on each aspect for experimentalists? For TRG, DAQ, or FEE?

Application of FPGA in HEP experiments (cont'd)

- **Our target:** Study the latest COTS FPGA devices and their associated new technologies for possible application and upgrade in different aspects of HEP experiments.

- **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.

New technologies for TDAQ in HEP

- **FPGA device**:
  - COTs, ACAP, SoC, such as Xilinx Versal, RFSoC, etc.
- **High-speed serial data transmission**:
  - From Non-Return-To-Zero (NRZ) to Pulse-Amplitude-Modulation (PAM4)
  - Optical module: QSFP, FireFly, other EOM.
- **Data readout**:
  - PCI-Express: PCIe40 (Gen3), PCIe400 (Gen5), FELIX (Gen4), Versal (Gen5)
  - Ethernet
- **Algorithm in FPGA**:
  - High-Level-Synthesis (HLS), ML inference, etc.
  - Xilinx Versal AI engine
- **Hardware acceleration**:
  - Xilinx Versal and Alveo acceleration card, Deep Processing Unit (DPU)
  - GPU

Proposal for R&D: universal electronics device

- The device for low level trigger in Belle II and ATLAS:
  - Strong FPGA with large resource for physics algorithm implementation.
  - Lots of features in common.
  - Teamwork R&D for a new version of universal device.

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

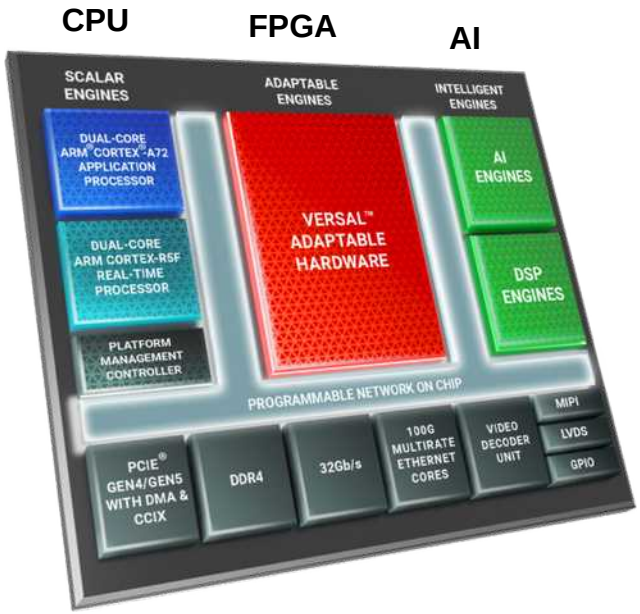
Xilinx UltraScale+
XCVU13P XCZU5EV
GTH,GTY: 16.8 Gbps
with 64B/66B

- Collaboration of multiple groups.
  - PCIe40 readout is the best example: ALICE, LHCb, and Belle II.
  - Design can be optimized with experts from different backgrounds.

Versal project: Ongoing task-force

- Considering the future devices' R&D, KEK together with CEF members purchased a few evaluation kits of the Xilinx Versal series FPGA for common study purpose.
  - Target: Developing a universal FPGA device for general experimental purpose, such as L1 trigger or readout.

- The features of Versal series:
  - ACAP SoC.
  - AI/DSP engine: interface to implement ML core into firmware.
  - High Bandwidth Memory (HBM).
  - Larger number of cells + High transmission bandwidth.

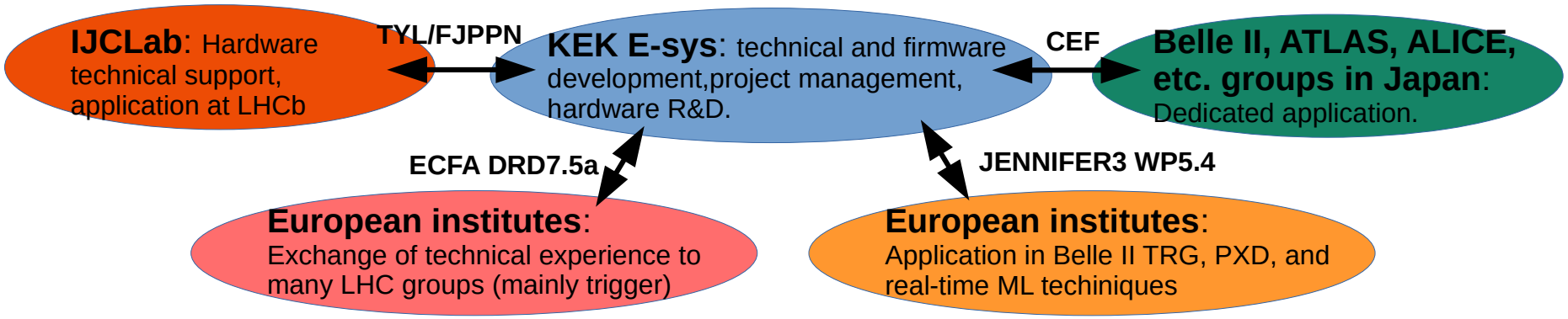


source: Xilinx website

International networking

- Here is our international networking based on this Versal project.
  - There is also communication with individual institute for specific development.

- **ECFA DRD7: 7.5a**
  - TDAQ backend with COTS of heterogeneous computing architecture
  - Utilizing different hardware platforms (FPGA, GPU, CPU) for real-time processing (trigger).
  - Target: Building an open-access, repository-hosted infrastructure for these commonly used tools and algorithms.
    - We will start to construct this repository from the end of 2025!
  - ~20 institutes join 7.5a.
  - I am one of the convener.

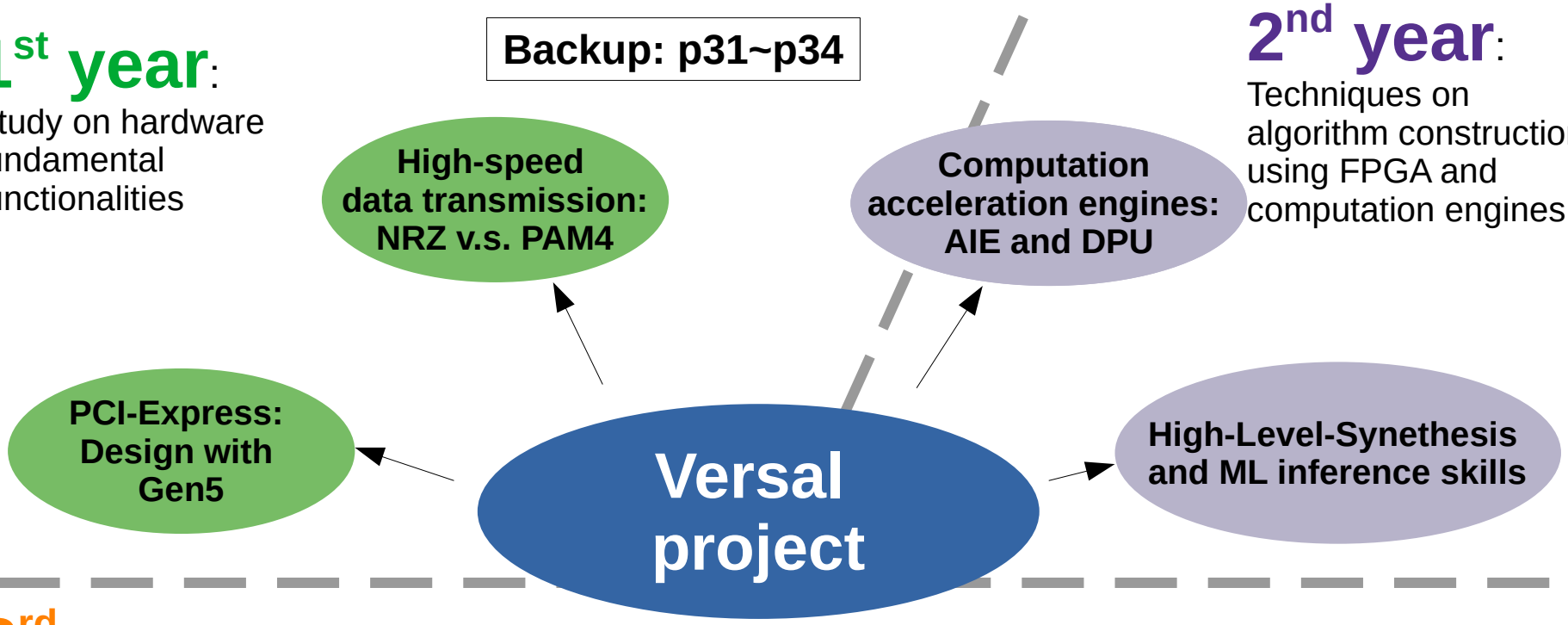


Project overview and working plan

1st year:
Study on hardware fundamental functionalities

Backup: p31~p34

2nd year:
Techniques on algorithm construction using FPGA and computation engines



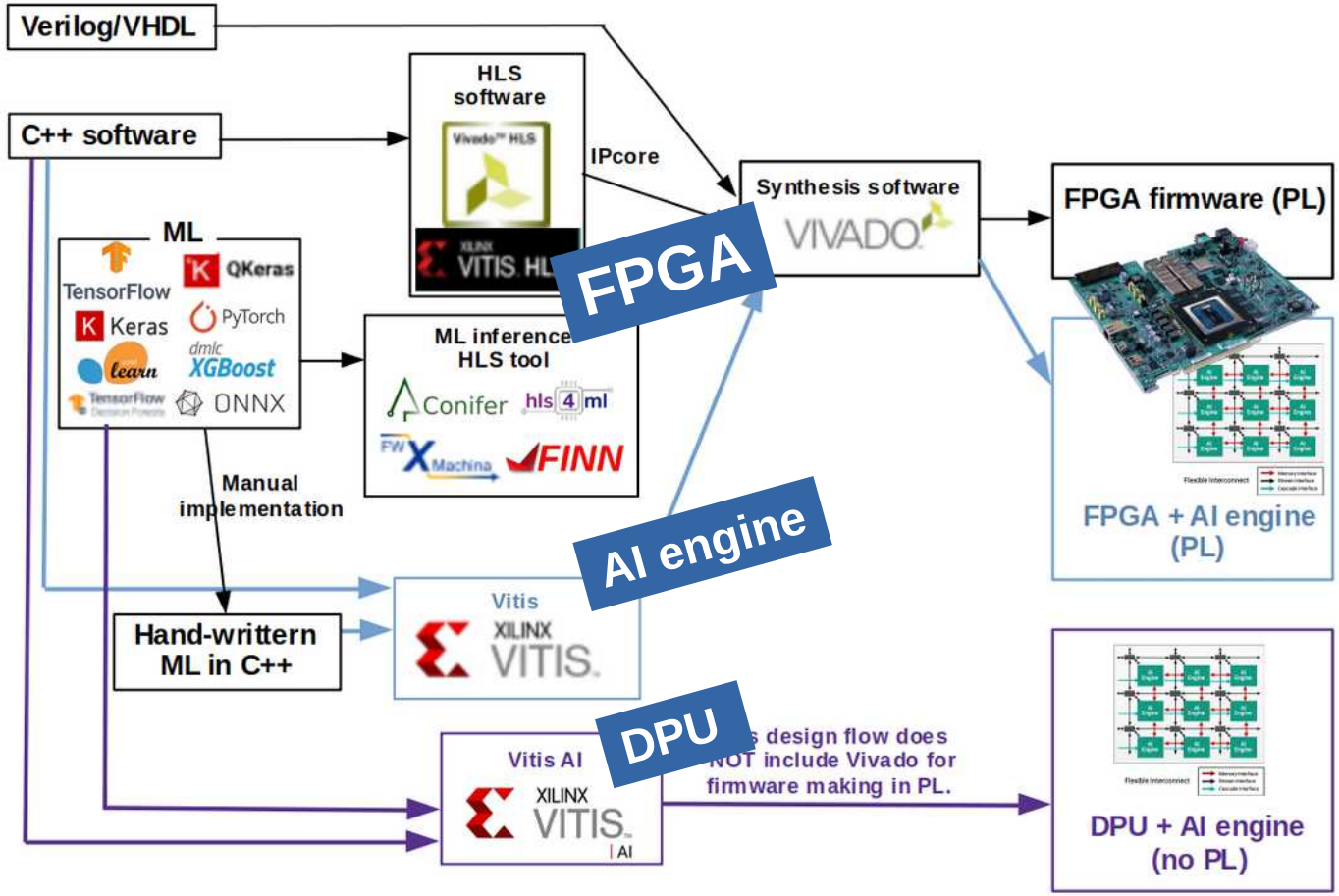
3rd year: R&D works for utilizating Verasl in real experimental systems

- New Level-1 Trigger device for Belle II: UT5
- Upgrade for Belle II HLT with FPGA
- SuperKEKB Bunch Oscillation readout system
- Belle II Readout Upgrade

FPGA methodologies on HLS, ML in FPGA, AIE

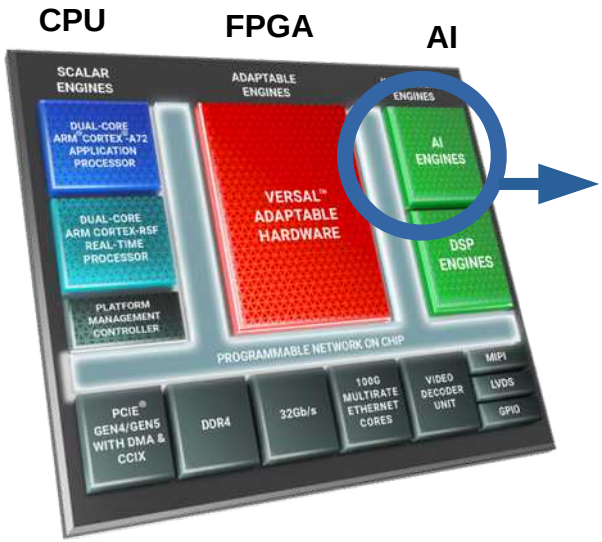
Backup: p35~p40 for details of individuals

- "Not only what kind of logic to make, but also how to make it."
- We almost finished buliding this technical database.
- We also held a summer school in Aug. 2025 for these techniques.
- In addition, we have some ongoing trigger algorithms development associated to it.

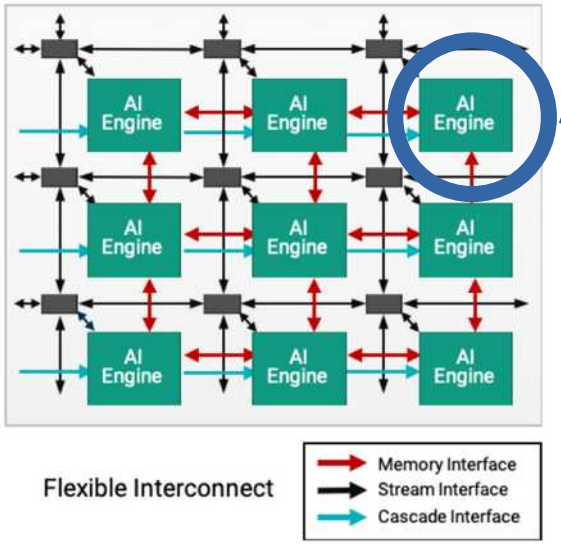


AI engine of Xilinx Versal ACAP

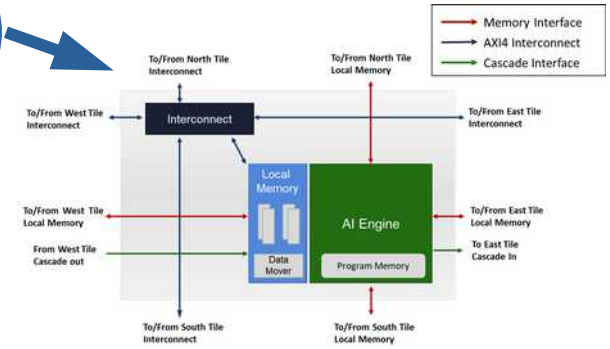
Versal ACAP



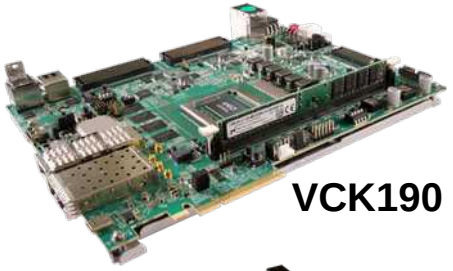
Versal AI engine



An AI engine "tile"



- Computation acceleration engine of Versal ACAP.
- Embedded processor of FPGA.
  - High bandwith between FPGA and AI engine.
  - Outside of FPGA fabric.
- **C programmable**.
  - High precision.
  - No quantization loss on ML.
- **Low latency**.



VCK190 with AIE

VEK280 with AIE-ML

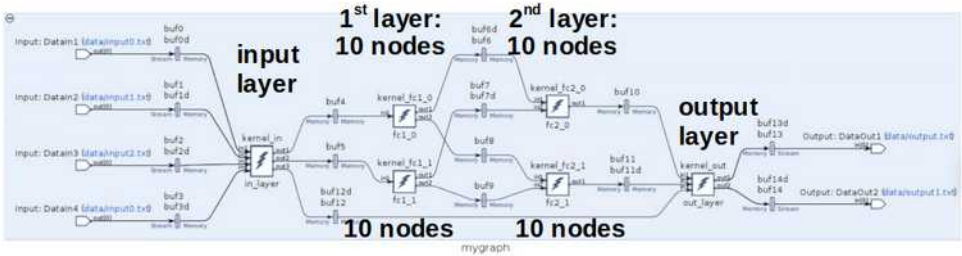
FPGA and AI engine algorithm implementations

- THe original designs are based on HLS inference tools for FPGA implementation. Then, plain C++ is written in Vitis for Versal AI engine.

NN for tau trigger in L1

R. Nomaru (Univ. of Tokyo)

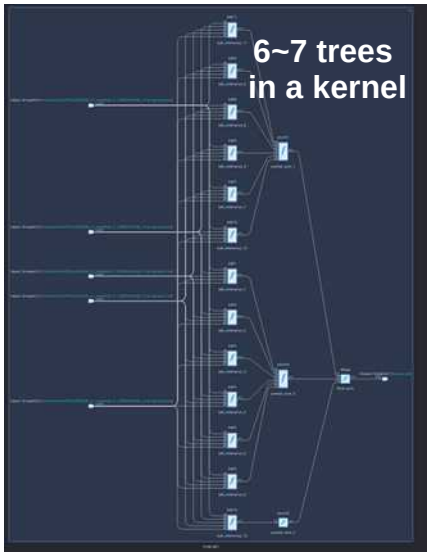
- Neurons: 19,20,20,1



BDT for tau trigger in L1

- N of estimator = 90
- Depth = 3

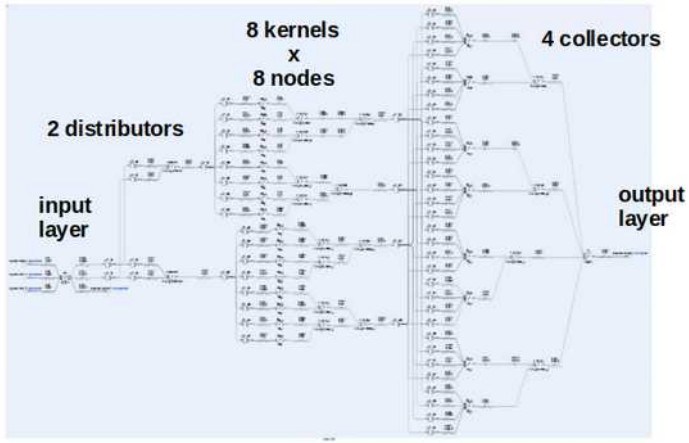
Y. Ahn (Korea Univ.)



NN for KLong-Muon chamber trigger

- Neurons: 8,64,16,3

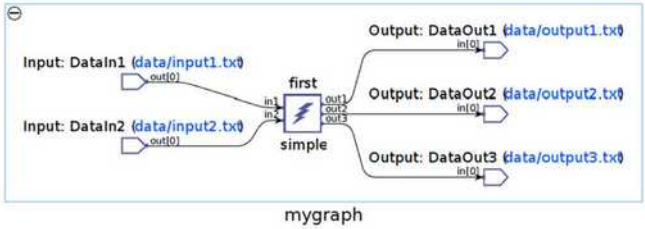
A. Little (Univ. of Sydney)

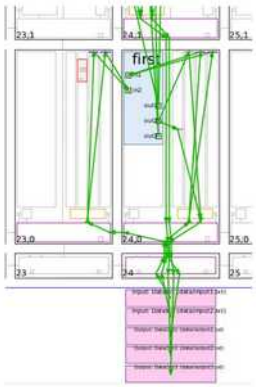


Linear fitter

J. Song (Korea Univ.)

- Based on linear algebra
- C++ in Vitis HLS



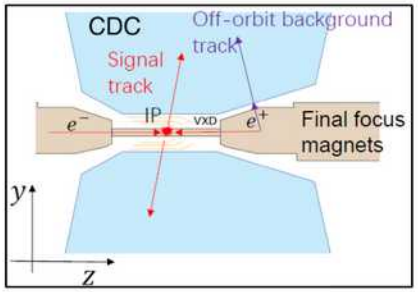


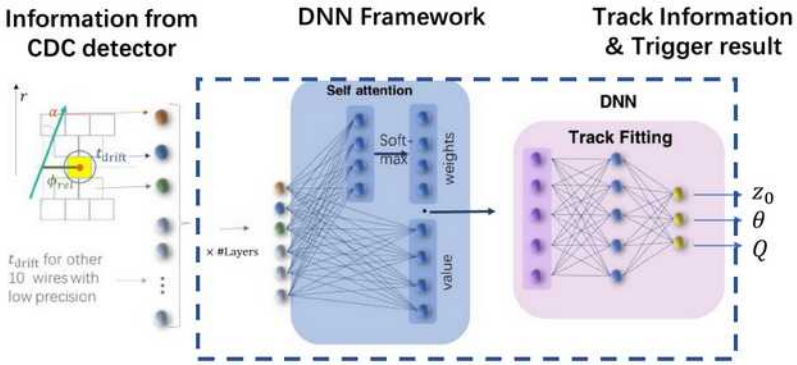
More plans for AI engine

Also see Qi-Dong's talk

- Belle II L1 trigger: Deep-Learning NN for 3D tracking (z-trigger).
  - Original design based on Pytorch and hls4ml.

Original design:
Y. Liu (Sokendai)

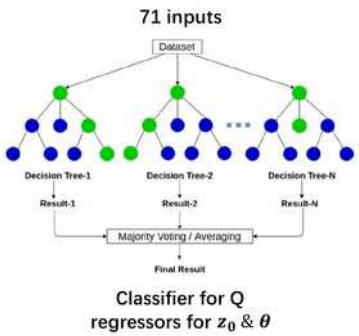


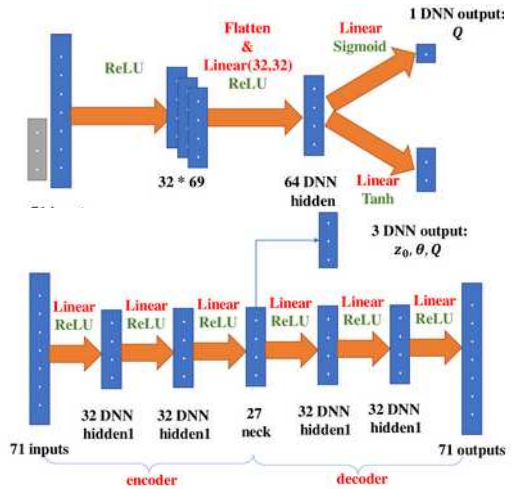


- Ongoing development from the present DNN design:
  - Tuning on DNN
  - CNN
  - Auto Encoder
  - Random Forest
  - Gaussian Processing
  - Support Vector Machine

Y. Yang (Fudan Univ., KEK)

Belle II UT4 → Versal FPGA → Versal AI engine

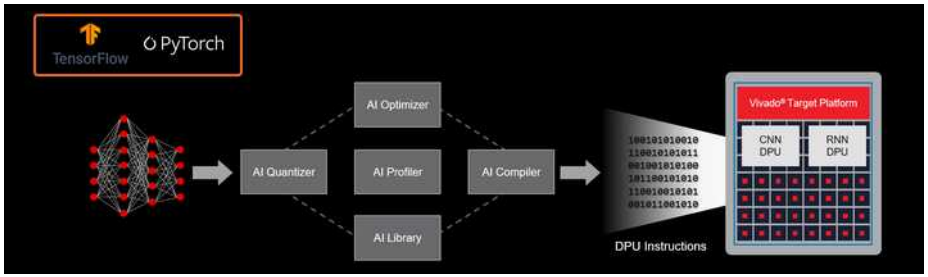


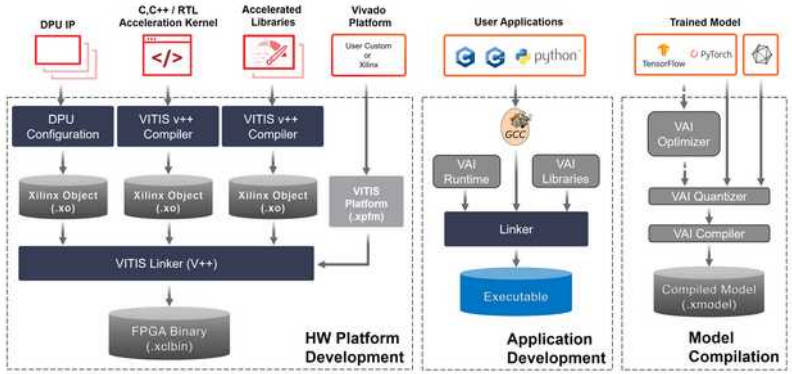


Versal DPU

- **DPU**: Deep Learning Processing Unit
  - Configurable computation engine dedicated to convolutional neural networks.

- DPU takes leverage of the FPGA resource, while the artificial networks inference **does not require touching FPGA PL**.
  - An IPcore of FPGA
  - Network model building by Pytorch, and quantization by Vitis-AI software.
    - Many pytorch models are supported (not yet for GNN).
  - During utilization, the system is operating like a small OS.
    - Replacement of the network model does not require FPGA reprogramming.
    - User operates everything in **command line with ssh and scp**.
  - **Hardware acceleration for high-level application**.

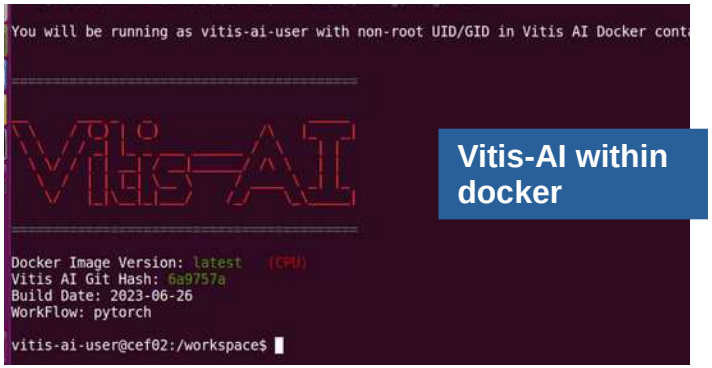




Versal DPU, network building and running

Chaowaroj "Max" Wanotayaroj (KEK ATLAS)

- Pytorch for NN model building.
- Quantization on the model is performed using "VItis-AI" tool from Xillinx inside docker.
  - GPU is also supported.
- After the model is quantized by Vitis-AI, we simply copy the model and data files to the DPU (using scp), then execute the jobs inside DPU.



Vitis-AI within docker

```
root@xilinx-vck190-20222:~/03_vck190_pytorch_atlas-top-tagger# python3 app_mt.py
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
inf> Starting 1 threads...
inf> Throughput=17749.99 fps, total frames = 1000, time=0.0563 seconds
inf> Accuracy= (856/1000)=0.856
root@xilinx-vck190-20222:~/03_vck190_pytorch_atlas-top-tagger#
```

ATLAS top tagging open data inside Versal DPU

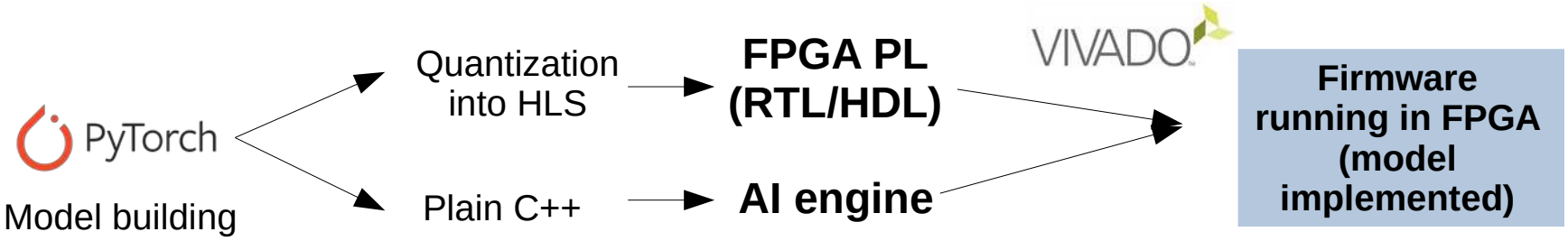
```
root@xilinx-vck190-20222:~/04_vck190_pytorch_muontrigger/Training/Endcap/PtClassification/Training# python app_mt.py
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
inf> Starting 1 threads...
inf> Throughput=16850.75 fps, total frames = 1000, time=0.0593 seconds
/home/root/04_vck190_pytorch_muontrigger/Training/Endcap/PtClassification/Training/app_mt.py:142: DeprecationWarning: elementwise comparison failed; this will raise an error in the f
uture.
  corrects = np.sum(out_q == labels)
inf> Accuracy= (0/1000)=0.0
root@xilinx-vck190-20222:~/04_vck190_pytorch_muontrigger/Training/Endcap/PtClassification/Training# cd ../../../../03_vck190_pytorch_atlas-top-tagger/
root@xilinx-vck190-20222:~/03_vck190_pytorch_atlas-top-tagger# python app_mt.py
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
inf> Starting 1 threads...
inf> Throughput=17163.74 fps, total frames = 1000, time=0.0583 seconds
inf> Accuracy= (856/1000)=0.856
root@xilinx-vck190-20222:~/03_vck190_pytorch_atlas-top-tagger# S
```

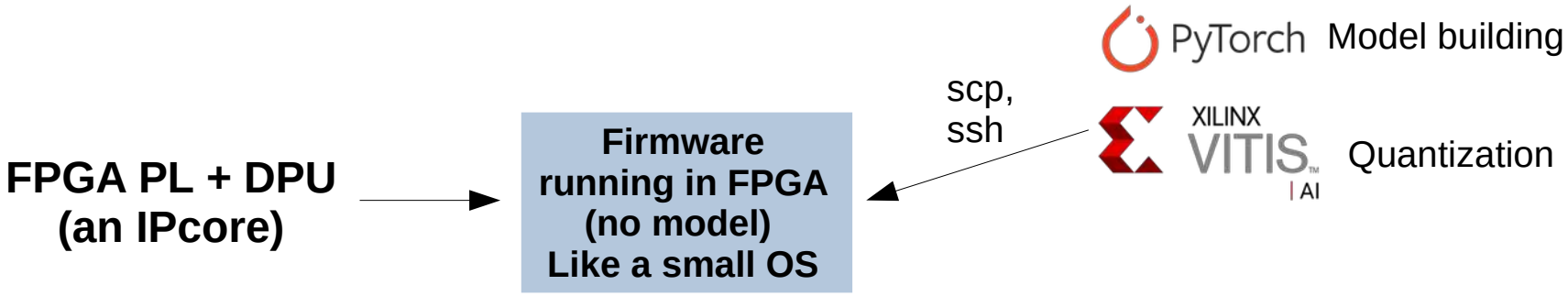
ATLAS muon reconstruction

- By default, connection to DPU is based on ethernet (ssh, scp).
  - Now we are studying data transfer using **PCIe**, which will be helpful for real utilization.

DPU v.s. AI engine v.s. FPGA

- Inference in **FPGA PL or AI engine**:
  - A fixed network has been implemented inside firmware.

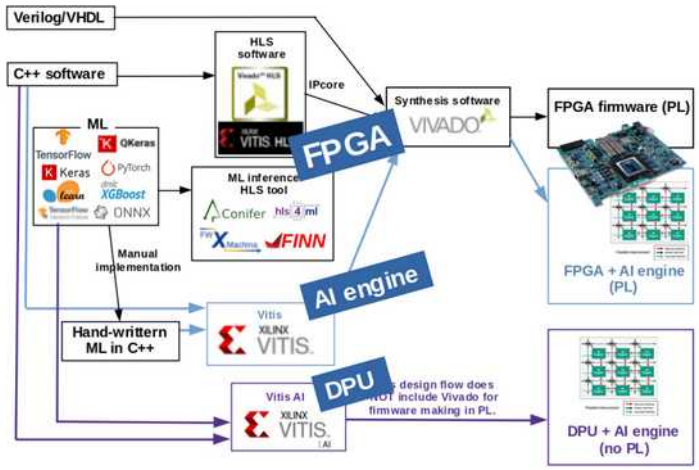




- Inference in **DPU**:
  - Firmware has no model implemented.
  - Model buliding and quantization are done independently.
  - FPGA can be accessed like a server with ssh and scp.
    - Model can be replaced in real-time without touching FPGA firmware.

Summer school on HLS, ML in FPGA, AIE

- Based on the technical database we collected, we held a summer school in 2025.
- In total 25 people from Japan and other countries.
  - Also people from different time zone!
- Content: HLS, hls4ml, Conifer, FINN, Versal AI engine
- Device: Nexys4 boards from Digilent
- We plan to have it once per year.



Belle II Level-1 Trigger board upgrade: UT5

Also see Koga-san's talk

Belle II UT3

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

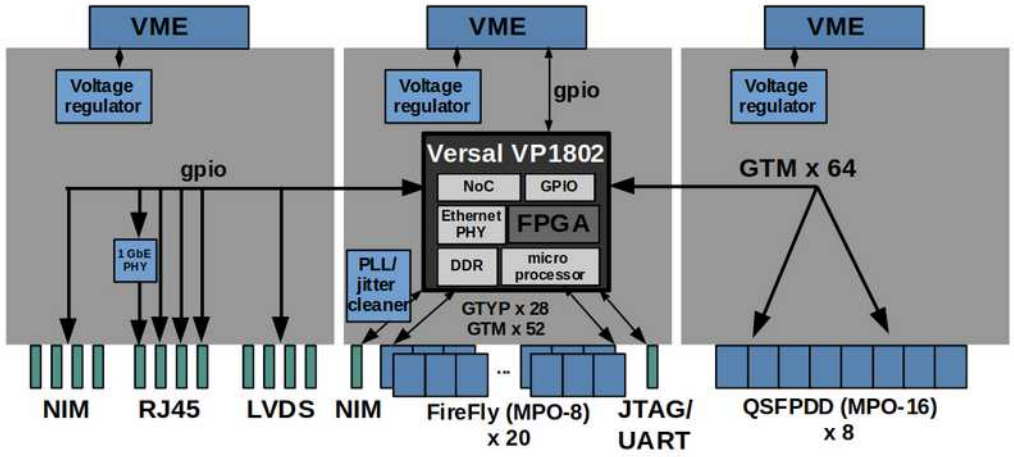
Belle II UT4

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

- Optical link: mainly QSFP28
- No Processing System (PS)
- VME for SLC
- All logics design based on PL

New design: UT5

Preliminary block diagram



- Trying to use other than QSFP28 (FireFly, etc) with smaller form factor.
- QSFP-DD for PAM4 in daughter board.
- Versal has Processing System (PS)
- Still VME for SLC
- Prabably no AI engine in UT5
  - But we are still open for the potential for UT6
- Aiming for prototyping in 2026

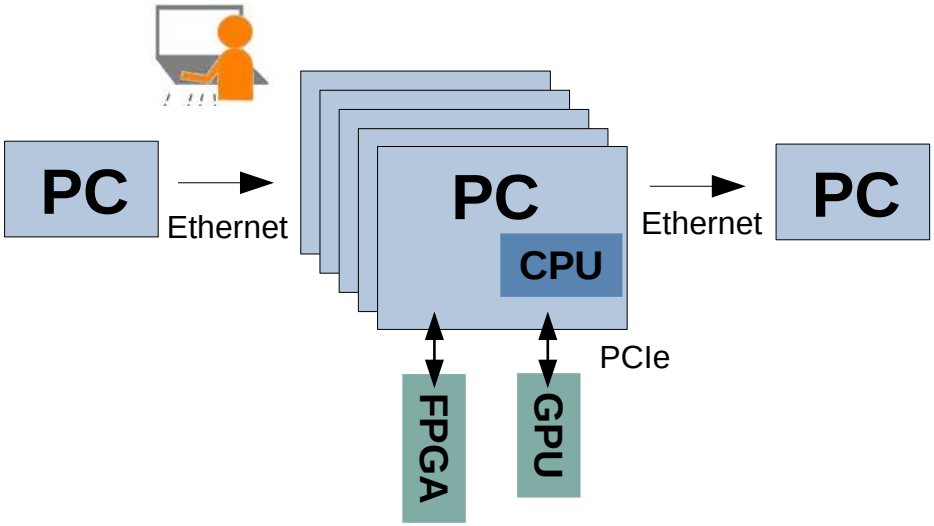
Other than CPU for HLT?

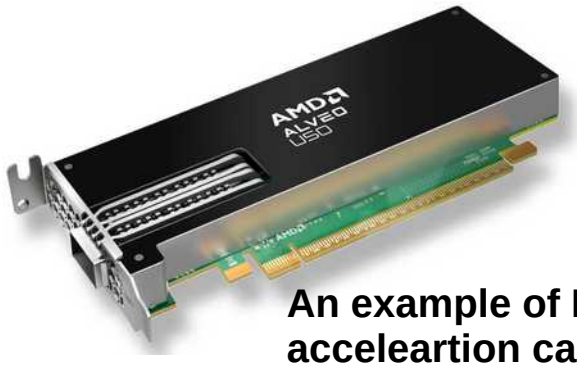
Also see Qi-Dong's talk

- People have been talking about something other than CPU for HLT: GPU or FPGA.
  - In such case, PC is the host.
  - PC transfers the data to external devices, then get the processed output back to PC.

- The design flow for FPGA logic and integration:
  - Developers make design using **C++, python, ML, or HLS tools**.
  - Together with the libraries from vendor (Xilinx), integrate everything into application.
    - DDR memory, data link, Ethernet, PCIe, etc.
  - **User can execute the application in the host PC command line**.

- The design flow mostly **does not require touching FPGA PL, RTL/HDL and Vivado**.
  - "**Hardware acceleration with FPGA**"



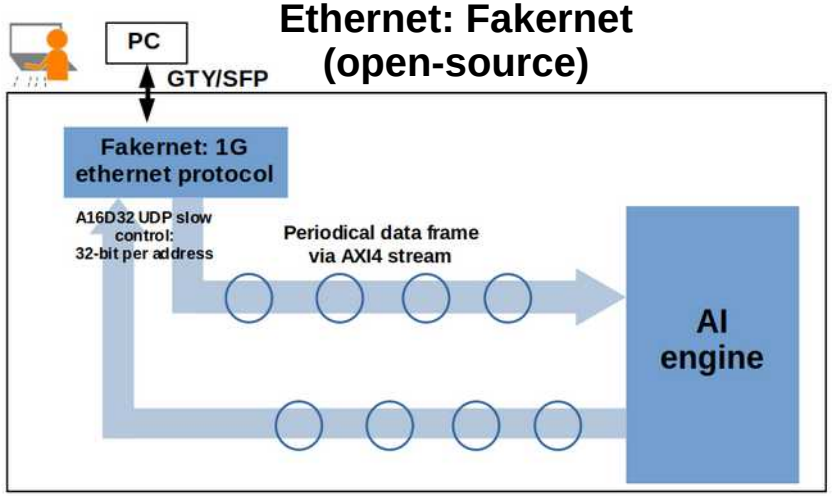


An example of FPGA acceleartion card: AMD Alveo U50

Communication with PC: PCIe or Ethernet

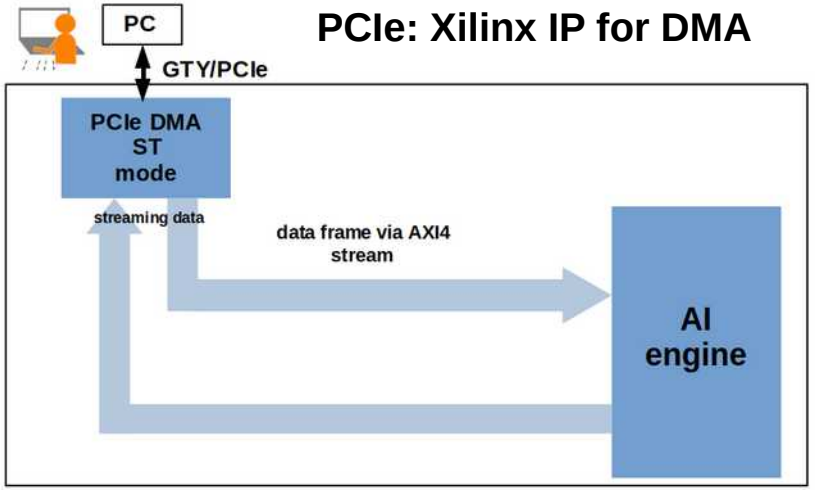
- How about Versal AI engine for HLT?
  - We need **PC-FPGA communication**, and **expertee of integraton in FPGA PL**.
- We tested the designs with Ethernet data link and PCIe of VCK190 for demonbstration.
  - Complicated design. Require expertee in FPGA PL design.

Ethernet: Fakernet (open-source)



- Support 1G and 2.5G
- GTY transceiver with optical SPF at FPGA, NIC at PC
- 1.5 hrs for 200,000 events

PCIe: Xilinx IP for DMA



- Self-defined protocol for data exchange.
- 50 min for 200,000 events.

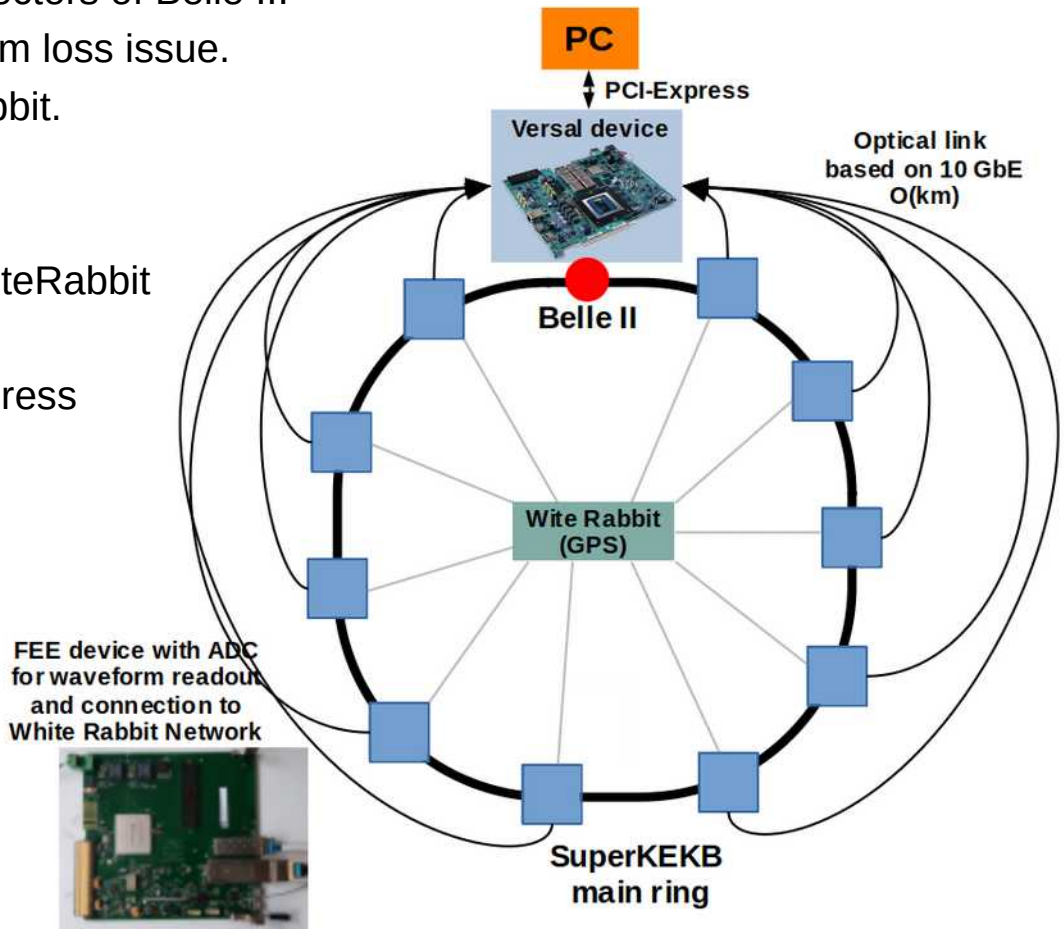
➔ **Potential for HLT application.**

SuperKEKB Bunch Oscialltion Readout system

- Motivation: To handle the sudden beam loss problem in SuperKEKB, we plan to prepare a system to readout the bunch waveform of oscillation
  - Final target: real-time prediction on the sudden beam loss using FPGA readout system.
    - Protection on the inner detectors of Belle II.
  - Feature study for sudden beam loss issue.
  - IDROGEN + ADC + WhiteRabbit.

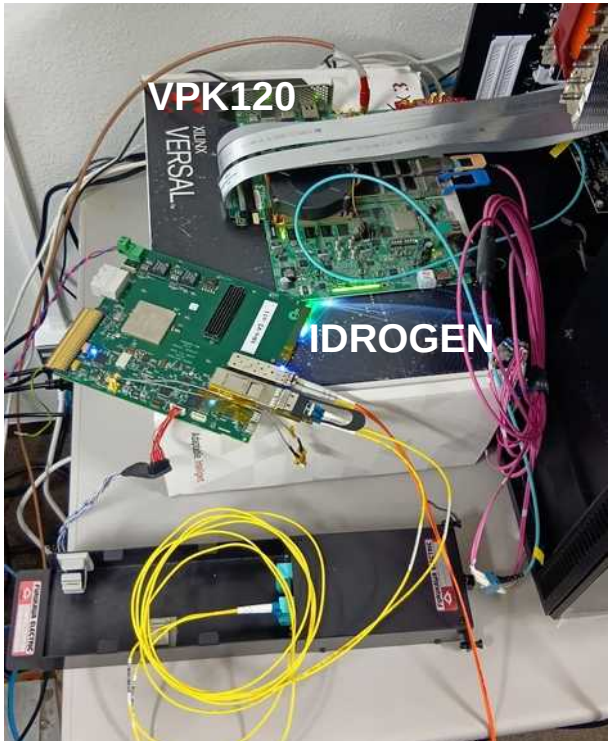
- System:
  - FEE: IDROGEN + ADC + WhiteRabbit
  - Long-distance optical link
  - Readout: Versal with PCI-Express
    - ML-based logic in Versal

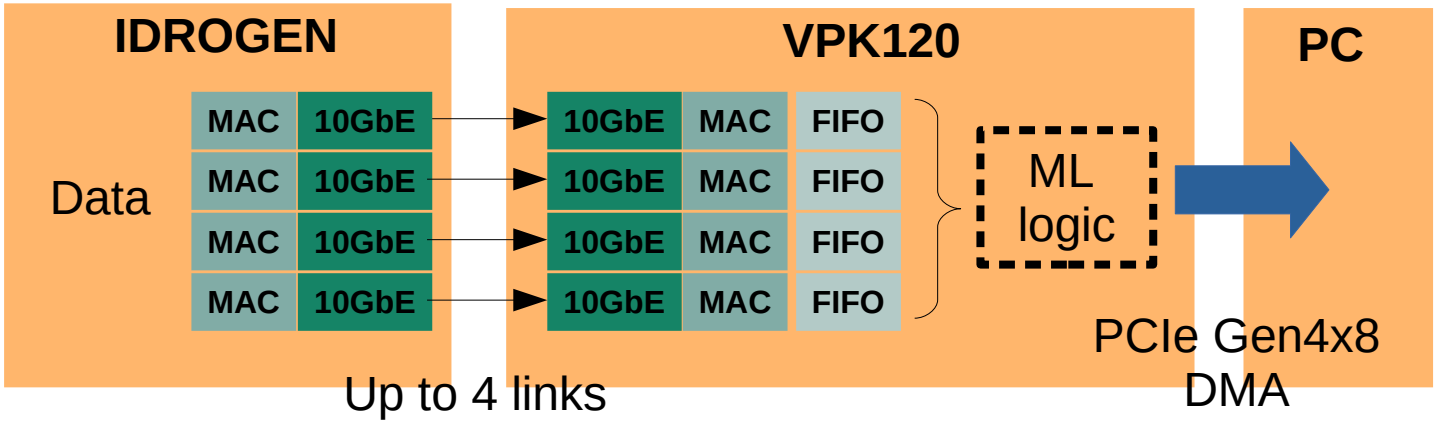
- Collaborators:
  - Univ. of Hawaii: K. Yoshihara
  - KEK ACCL
  - KEK E-sys/CEF
  - IJCLab.



SuperKEKB Bunch Oscialltion Readout system: Progress

- The entire data readout chain has been established:
  - IDROGEN → Optical link → Versal (VPK120) → PCI-Express → PC.
- Data link is based on 10 GbE and MAC.
  - Simplicity for framing transmission and prorocol design.
- PCI-Express: Based on DMA. Tested with Gen4 x 8.
  - VPK120 is up to Gen5.
- ML logic: To be developped.

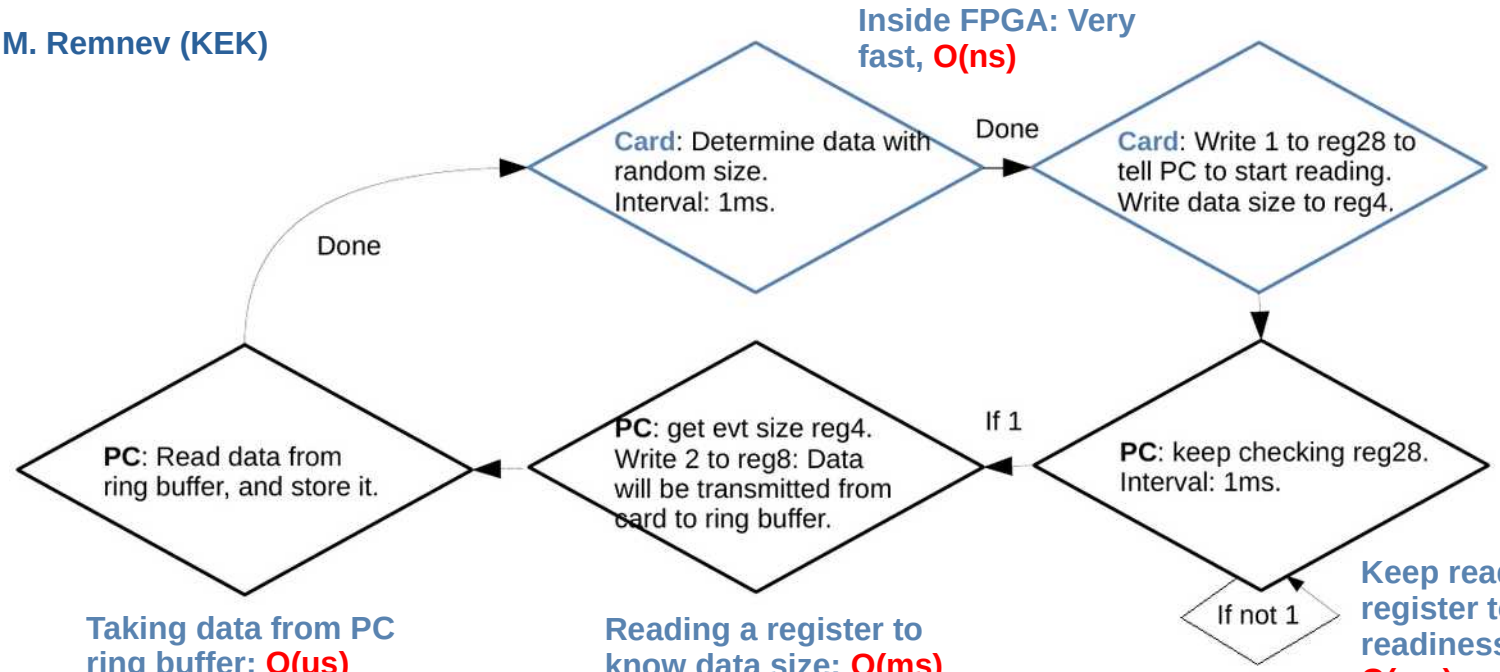




Versal PCIe study

- We are studying the Versal PCIe QDMA IP design and trying to develop a continuous event readout system.
- The original design was slow due to register R/W and memory storage controlled by CPU.
- Now it has been improved with optimizing C program.
  - ~300 MB/s with variable event size is reached per lane (Max. 2 GB/s per lane for Gen4.)
  - Other approaches are under testing with more improvement expected.

M. Remnev (KEK)



Taking data from PC ring buffer: O(us)
Store data into text file: O(ms)
Improved to O(μs) by creating an in-memory partition.

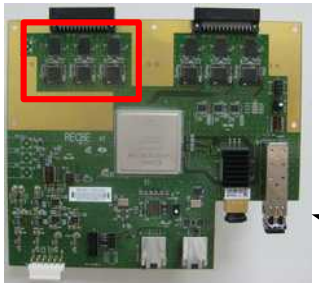
Reading a register to know data size: O(ms)
Writing a register to ask FPGA to dump data: O(ms)
Improved to O(0.1ms) by rewriting register I/O

Keep reading a register to know the readiness of FPGA: O(ms)
Improved to O(0.1ms) by rewriting register I/O

ML in real-time processing: Trigger or FEE?

A cut-view of Belle II CDC:
~14000 sense wires
~300 FEE

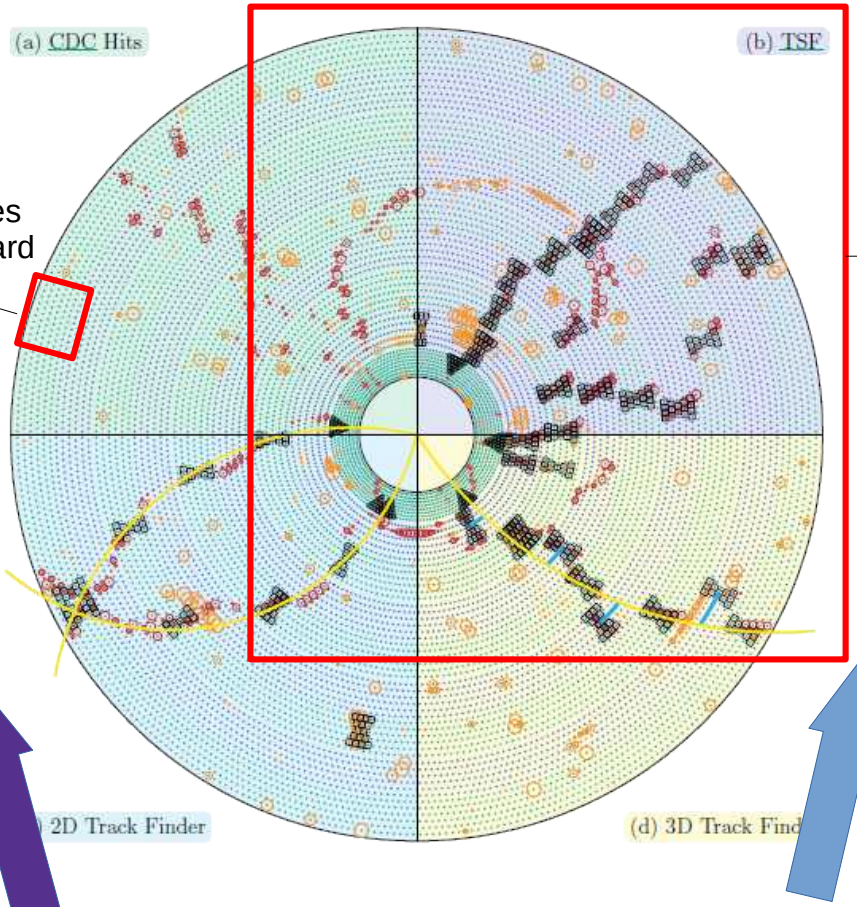
ASIC: Application Specific Integrated Circuit for Analog-To-Digital purpose



CDC FEE
Xilinx Virtex-5
Collect the 48 channels' information (ADC, TDC) to downstream.

FEE FPGA is usually not so strong, since the logic is not so complicated, and need to consider irradiation effect on electronics.

48 wires per board



Almost the entire detector

Belle II UT4
UltraScale

In TRG, FPGA is strong in order to process large date with complicated algorithm: tracking, clustering, topology, etc.

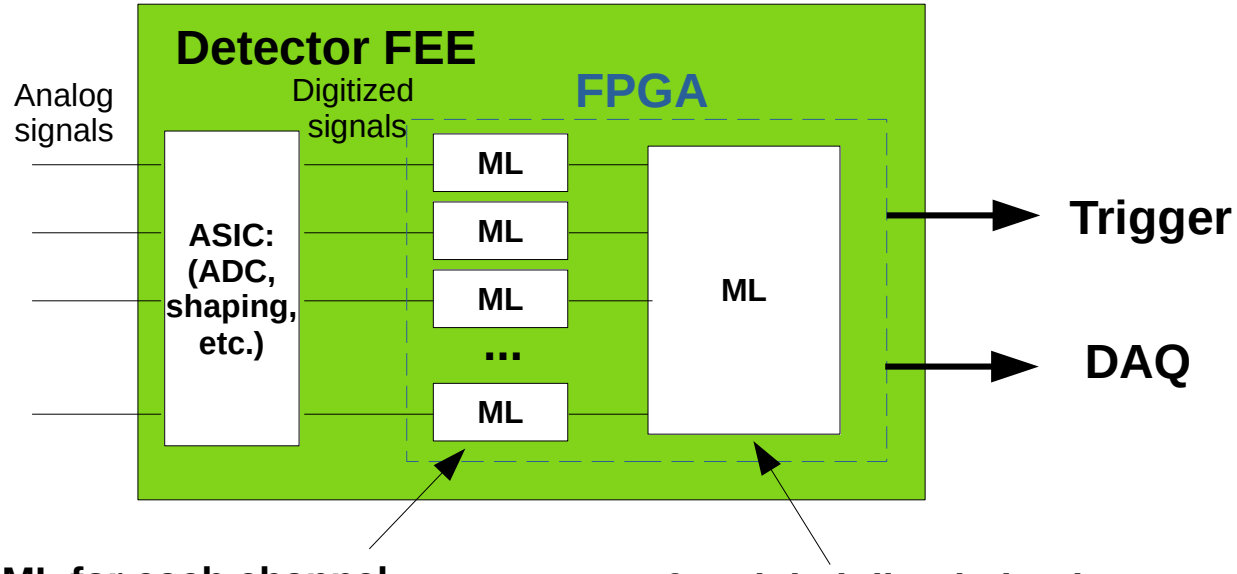
But how about here?

Here, we already knew that lots of thing to play with ML.

AI in FEE: New task-force

- Different from TRG, FEE usually uses smaller FPGA and covers a small region over the detector.

- A FEE receives digitized waveform information for each channel.
  - We can use ML to direct process the **digitized waveform channel-by-channel**.
  - ML can do many things, including those out of our expectation.
  - In R&D, the difficulty in analog design can be eased a bit.
  - FPGA of FEE is usually small, so building a **compact ML model** is critical.

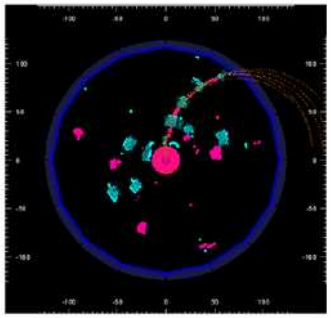


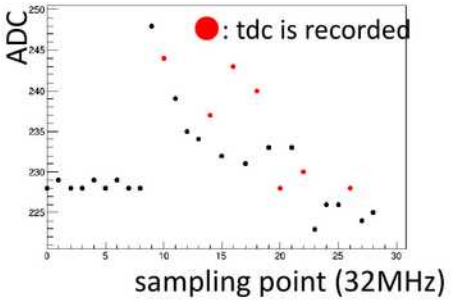
ML for each channel:
Shaping, TDC, energy, signal-noise separation, pile-up, hot channel, etc.

ML for global discrimination:
For some kinds of background/noise which have coinsidence between channels. e.g. crosstalk noise.

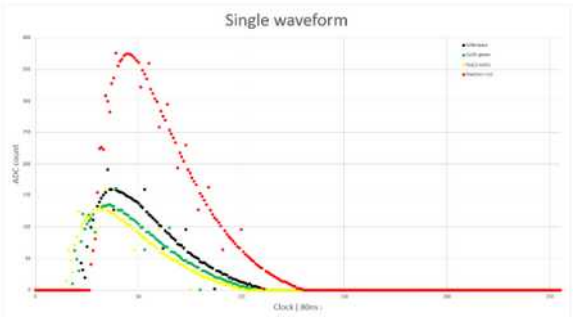
AI in FEE: developments

- **Belle II**: Now K. Yoshihara (Univ. of Hawaii) and I are trying to organize the possible research plans in Belle II into potential general upgrade plans.
  - CDC: ML-based crosstalk noise reduction
  - TOP: feature extraction (time/charge) in backend
  - ECL: hadron and e/γ separation
  - KLM: p.e. counting for improved time resolution





- **Neutron detector** pulse shape discrimination (Tohoku Univ.)



- **Silicon detector** pulse shape discrimination for PID in experimental nuclear physics (Y. Yang, Fudan Univ., KEK)
- **ADC waveform feature extraction** for streaming readout in experimental nuclear physics (SPADI-A)
- **ASIC** development
- **Quantum Error Correction** with small latency (NTHU)

CEF workshop on 1st and 2nd Dec.

- CEF hosts meetings/workshops regularly to summarize our progress.

- It is upcoming on 1st and 2nd Dec.:
  - https://kds.kek.jp/event/57383/
  - Day1: Versal session
  - Day2: AI in FEE session
  - In hybrid
  - Venue in KEK: 4-go-kan 346 (3 階輪講室 2)

- You are more than welcome to join with us!

Summary

- Collider Electronics Forum (CEF) in KEK ITDC is a platform for joint R&D on electronics devices for experimental HEP.

- Versal project: Possible application of high-end SoC device in experimental HEP.
  - 1. Exploration on the fundamental functionalities
  - 2. General techniques on FPGA algorithms: HLS, ML, AIE, and DPU
    - A database is built with providing education activity.
  - 3. Real application in hardware system, and R&D of new device/system

- AI in FEE project: Application of compact AI/ML model in devices in Front-End level
  - A newly started project. We are collecting potential developments.

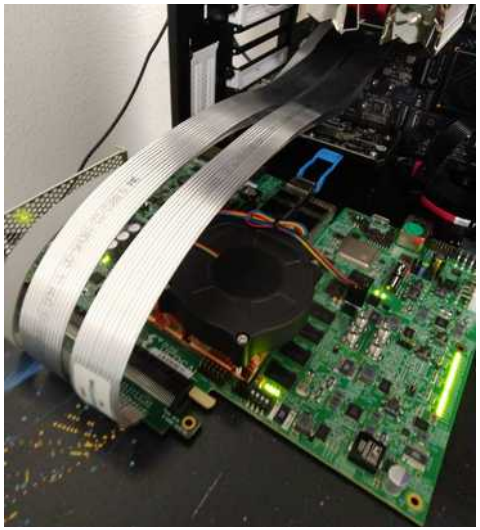
- You are welcome to contact us for any collaboration or inquiry for technical support.
  - Our workshop on 1st and 2nd Dec.: https://kds.kek.jp/event/57383/

Backup

Test benches of Versal kits @ KEK E-sys

- Now we have both VPK120 and VCK190 test benches at KEK E-sys group with host servers.
  - They are opened and shared with our colleagues in CEF.

PC side: PCIe Gen5 x16 slot



VPK120 test bench: 2023 summer

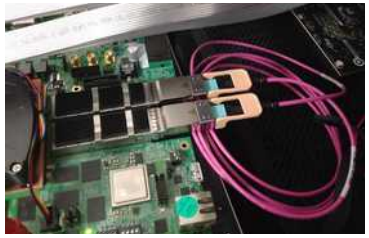
VPK120 connection to Belle II UT4

PC side: PCIe Gen4 x8 slot

VCK190 test bench: 2024 March

Transmission test with PAM4 and QSFPDD

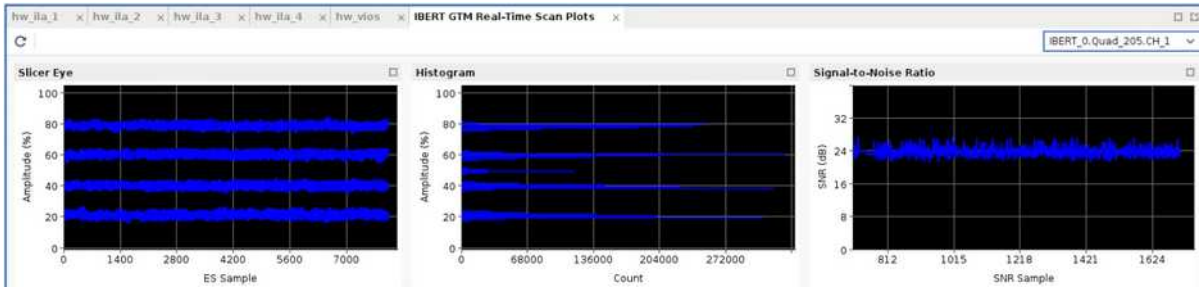
- We have successfully tested the real transmission with PAM4 and QSFPDD:
  - QSFPDD-SR8 with MPO16, from FS company.
  - 53.125 Gb/s x 16 lanes.
    - Only this line rate is supported.



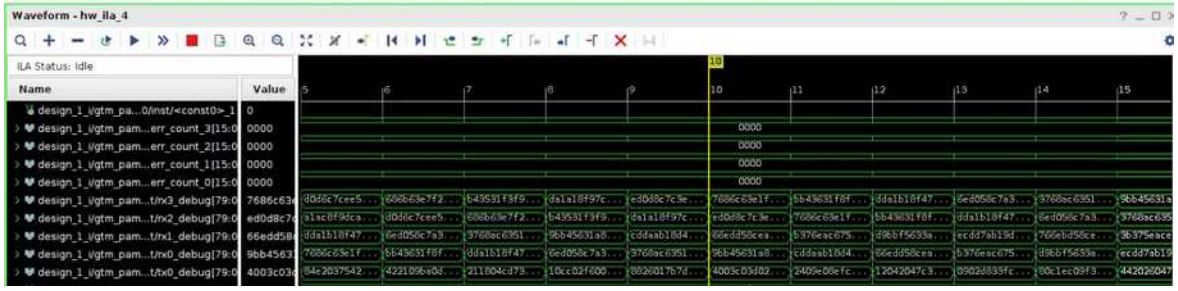
QSFP-DD-SR8

- 3-day BERT, Our self-designed protocol, PRBS16:
  - BER of the worst lane: **9.0×10^{-14}**
  - 16-lane combined BER: **6.7×10^{-15}**
  - Latency: **210~240 ns**

- Based on our experience, NRZ is usually $O(10^{-16})$
- This BER for PAM4 looksl acceptable.

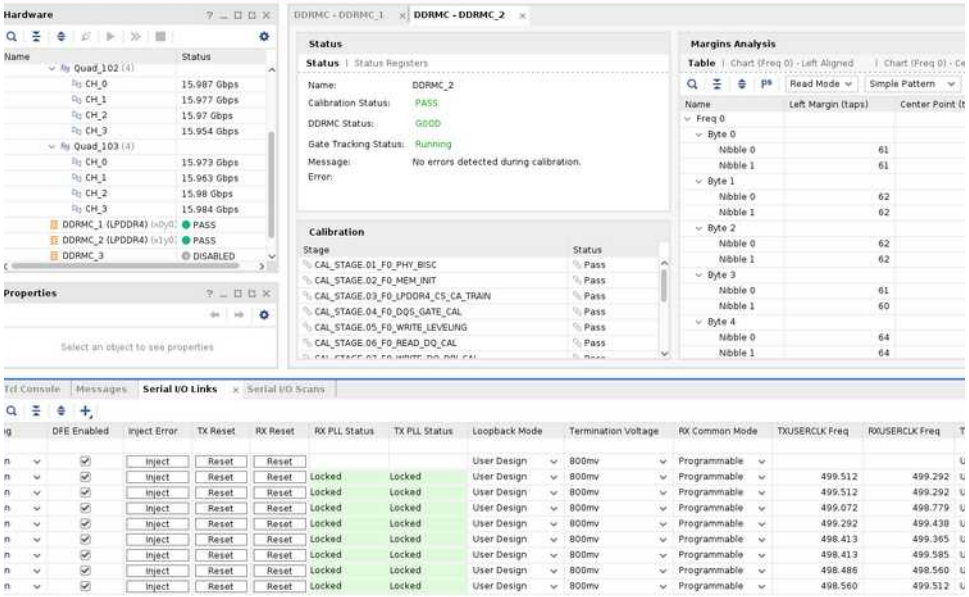


PRBS16 patterns



PCIe-CPM test with Versal kits

- 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.



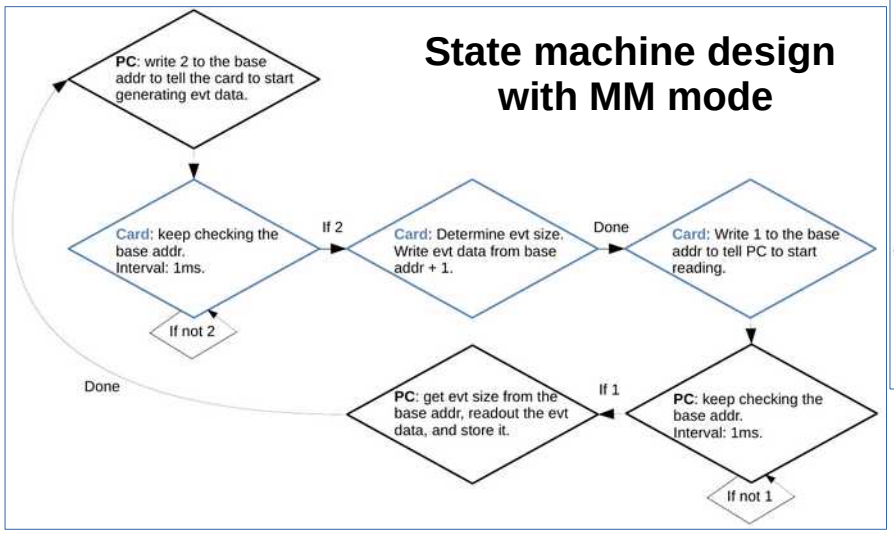
VPK120
Gen5 x 16

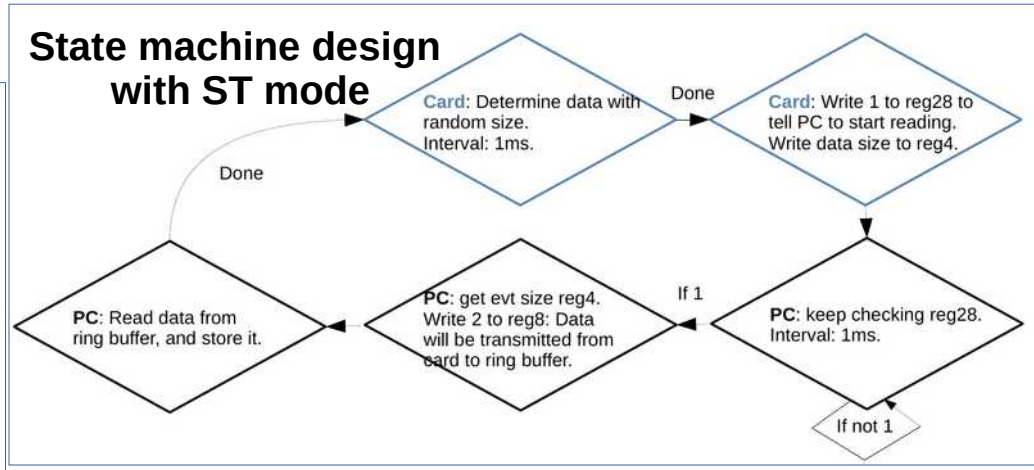
VCK190
Gen4 x 8

```
[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: Event readout

- State machine of the readout protocol between PC and FPGA.
  - Basically, handshake between PC and FPGA to know when is ready, what is data size, and when to take data.
- Random data size in the data generator.
- Not fully optimized yet: Long waiting time.
  - 1 ms waiting time in idle state to repeat.






```
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
```

Manual readout software

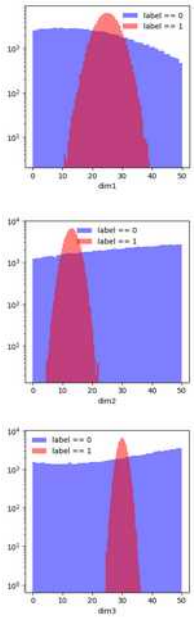
Store event data with random size

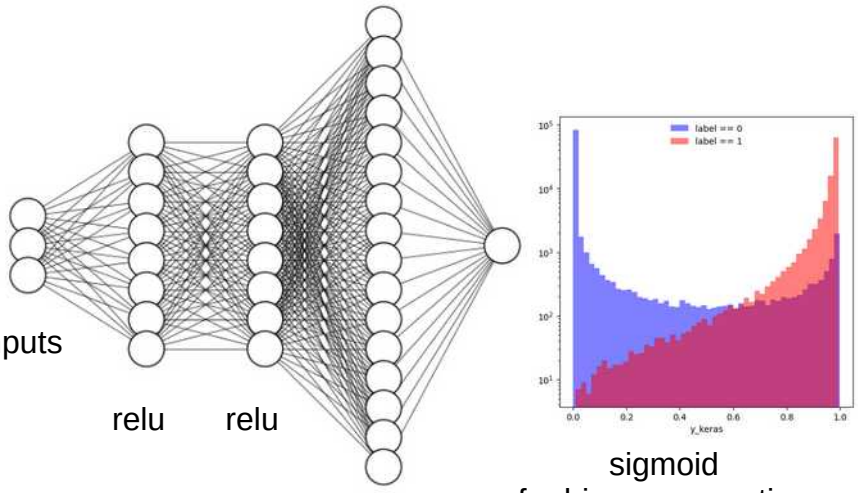
```
512  3月   5 11:01 0.log
192  3月   5 11:01 10.log
512  3月   5 11:01 11.log
448  3月   5 11:01 12.log
384  3月   5 11:01 13.log
128  3月   5 11:01 14.log
  0  3月   5 11:01 15.log
  0  3月   5 11:01 16.log
  0  3月   5 11:01 17.log
576  3月   5 11:01 18.log
512  3月   5 11:01 19.log
448  3月   5 11:01 1.log
448  3月   5 11:01 20.log
```

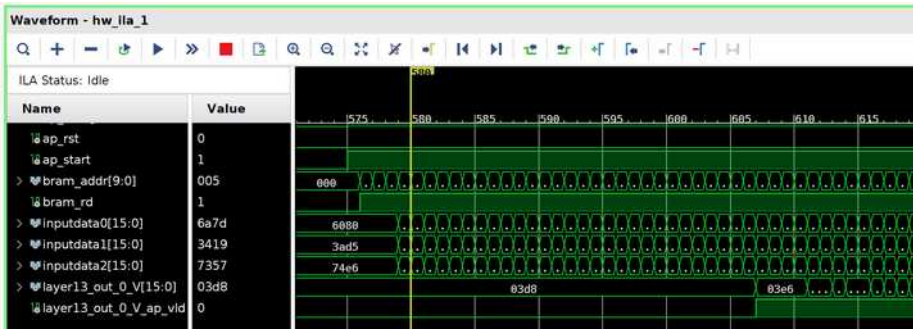
hls4ml

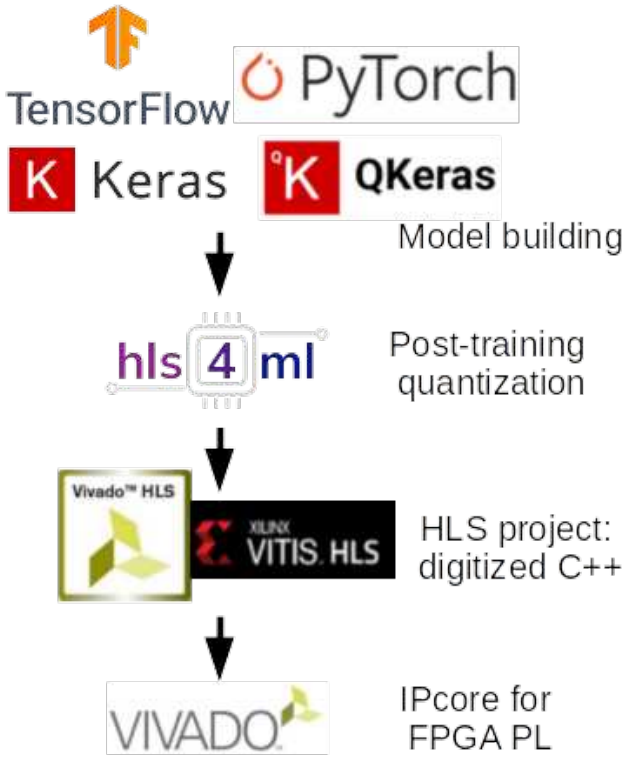
- hls4ml has been widly utilized in our field already.
  - For TensorFlow and Pytorch

- Just a smple demonstration using Nexys Video card and a bipolar separation NN model:







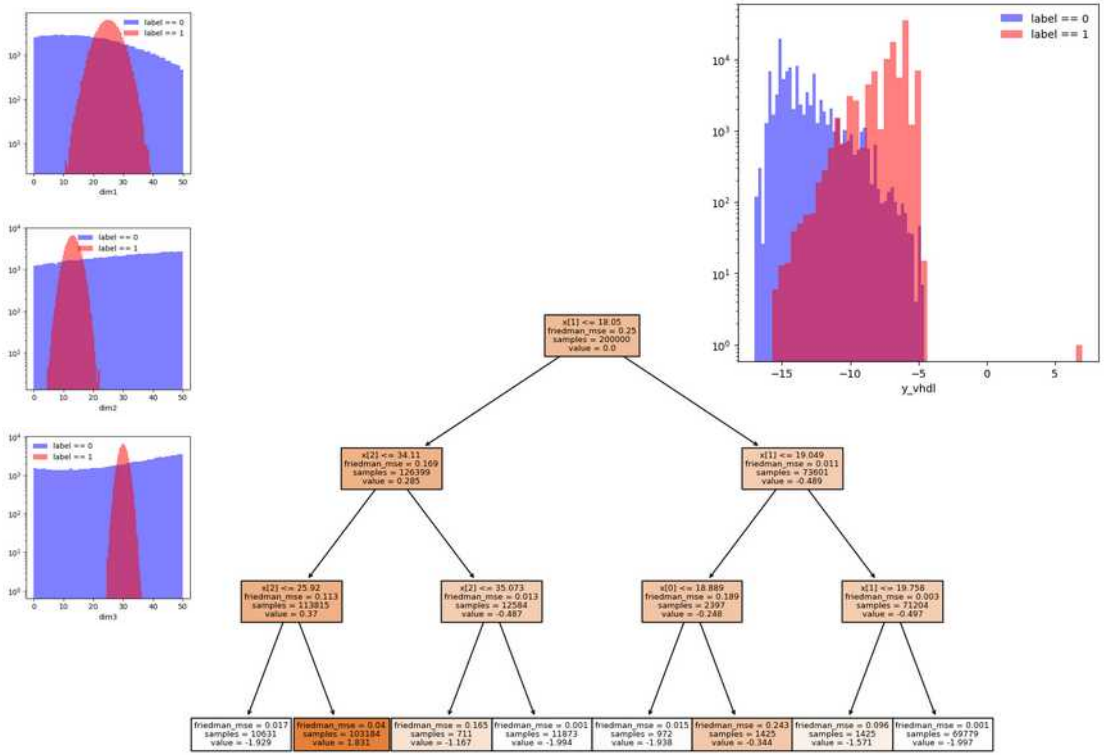


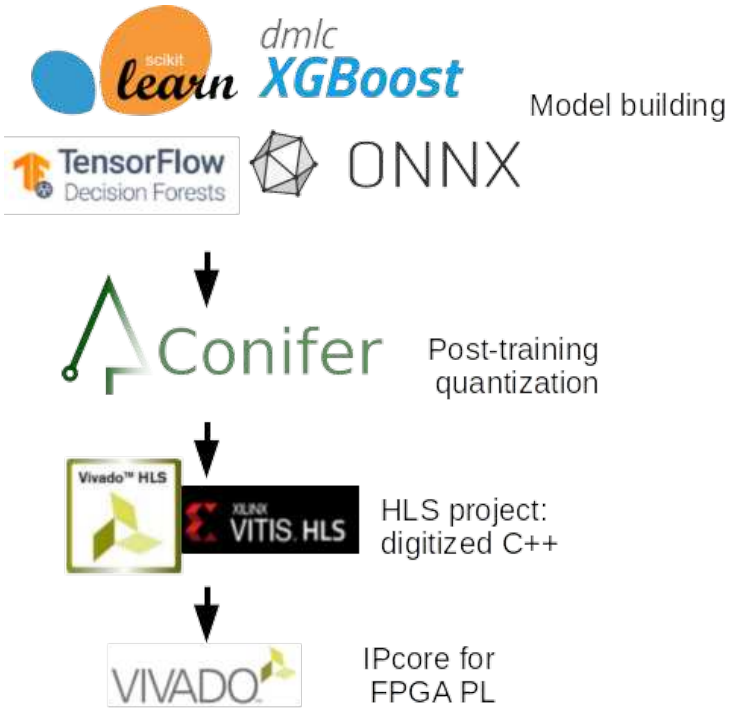
Latency: O(10) clock-cycles

New study: Conifer

- Conifer: a package for BDT inference in FPGA
  - The same developer group as the one for hls4ml.

- Compared to NN, BDT is suitable for separation purpose, but not for regression.

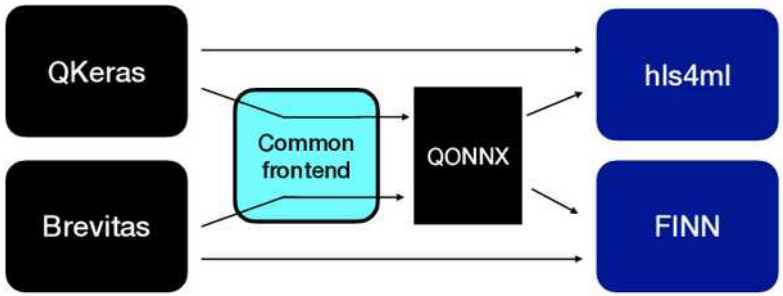




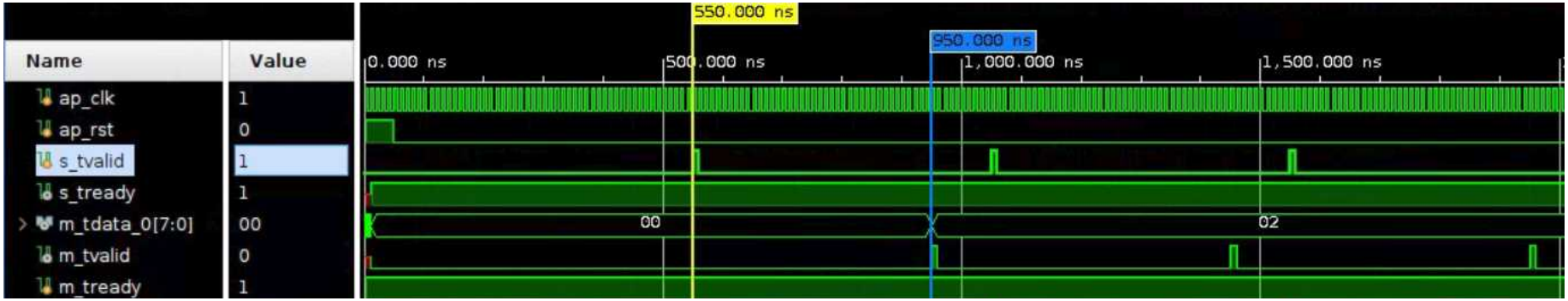
New study: FINN

- Under development by AMD Xilinx.
- The core concept is matrix multiplication.
- Quantization based on Pytorch + Brevitas.
- Model representation by ONNX/QONNX.

- Material is ready.
- Will also use it for our ongoing developments.



source: 10.48550/arXiv.2206.11791

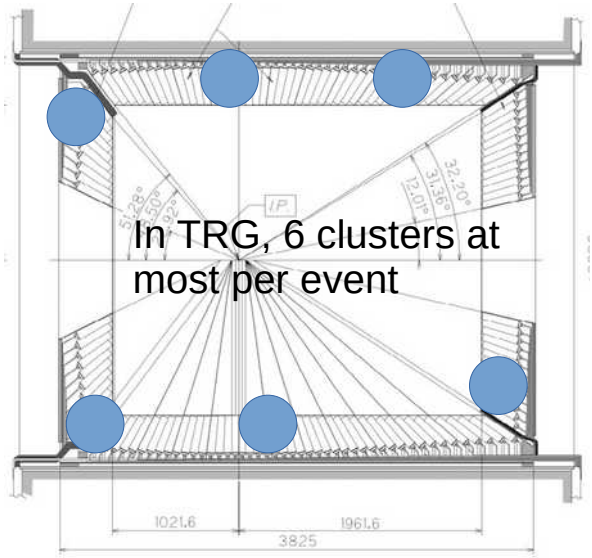


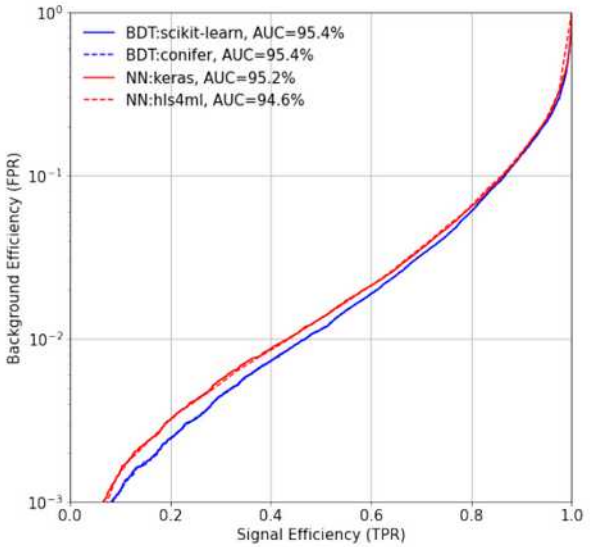
Belle II τ trigger: NN v.s. BDT

R. Nomaru (Univ. of Tokyo)

- Example: Belle II τ event trigger with calorimeter cluster
  - Input: clusters' position and energy
  - Output: Y/N for a $e^+e^- \rightarrow \tau^+\tau^-$ event
  - Original design is based on NN+hls4ml.

- For an alternative way using BDT+Conifer:
  - BDT can achieve the almost same performance.
  - Smaller LUT usage, and 0 DSP usage.





YongHeon Ahn (Korea Univ.)

Resources	BDT with Conifer	NN with Keras
Latency	12 cycles	14 cycles
Initiation Interval	1 cycle	1 cycle
LUT	22,504	28,480
Flip-Flop	11,629	10,632
DSP	0	228

Know-how of ML in AIE

- Here I use this self-defined Keras NN model for demonstration.

- After the model is built, I just obtained the math formula of the model, and write the codes for AI engine in Vitis.
  - **Everything for AI engine is in C++ and single-precision floating point.**
  - **No quantization loss.**

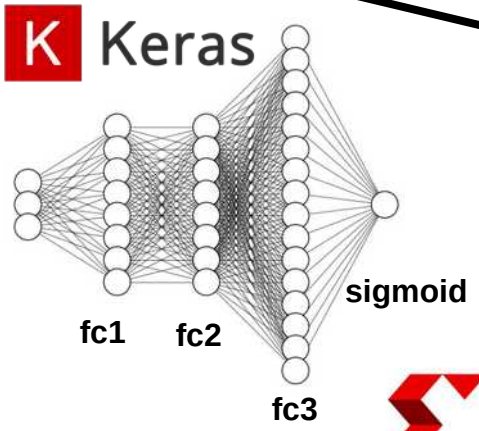
- **Latency: 3.4 μs.**
  - Can be further reduced.

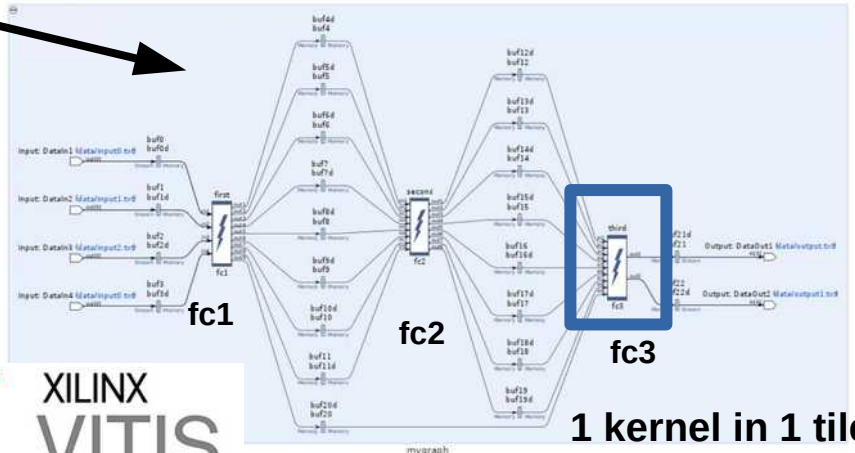
1 block = 1 AI engine "tile"
A unit with 32 kB memory.

TensorFlow

Keras

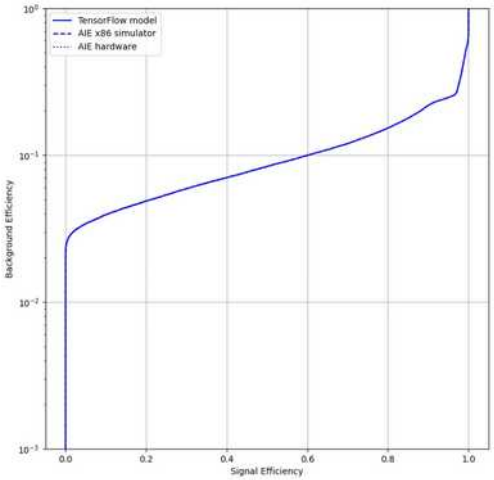
Exact math form in C++ in float





1 kernel in 1 tile

XILINX VITIS



ML in AIE: Belle II τ NN trigger

- Use the same NN model design mentioned in previous pages
- Implement the mathmatic formula of the Keras model in AIE.
  - No quantization
- 19,20,20,1
- Latency: 4.8 μs