D_RD_27: Machine Learning inference and computation acceleration engines of FPGA

Yun-Tsung Lai

on behalf of the D_RD_27 group

KEK IPNS

ytlai@post.kek.jp

2025 Joint Workshop of FKPPN and TYL/FJPPN

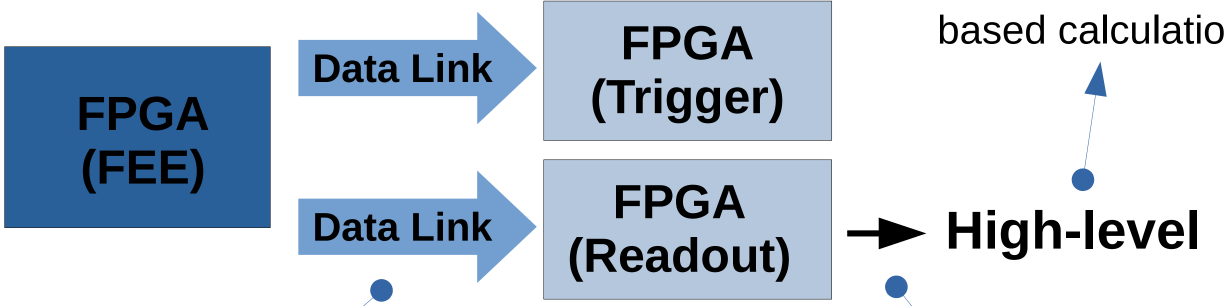
@ L2SN

14th May, 2025

Application of FPGA in HEP experiments

- **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 (FEE)

Data Link

FPGA (Trigger)

Data Link

FPGA (Readout)

High-level

- **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.

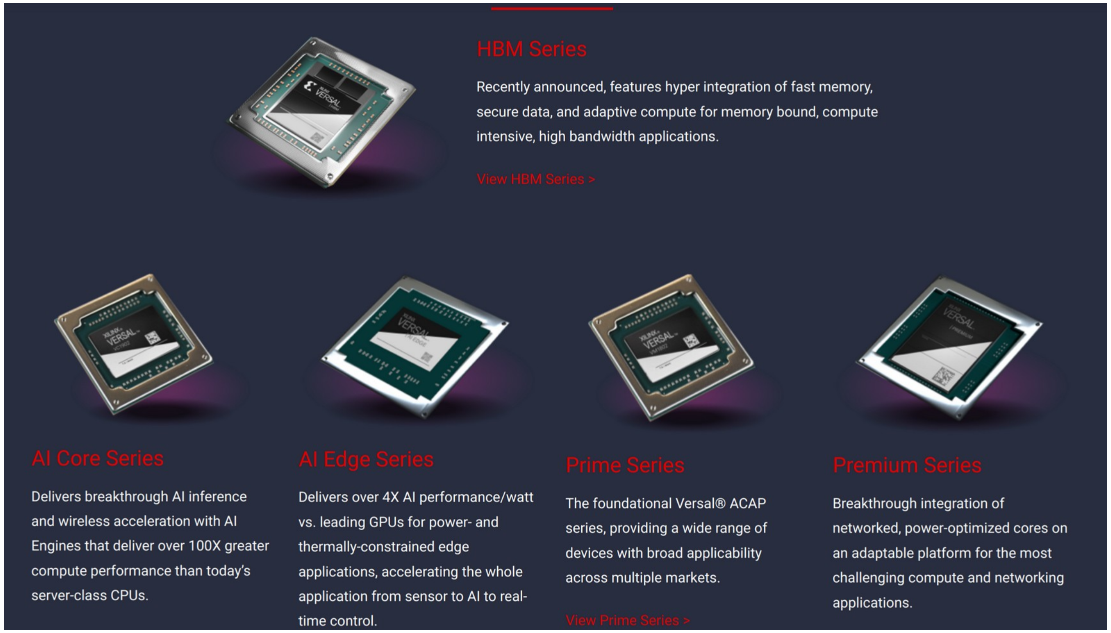
  are critical for the usage in:
  - **TRG**: complicated algorithm implementation.
  - **DAQ**: collect and process large data.

- **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.

D_RD_27: Modern FPGA devices for HEP

- D_RD_27: Study on modern FPGA devices for application in HEP.
  - Mainly based on the **Xilinx Versal series of ACAP**.
- KEK together with Japanese HEP community purchased a few evaluation kits.
  - 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.

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



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.

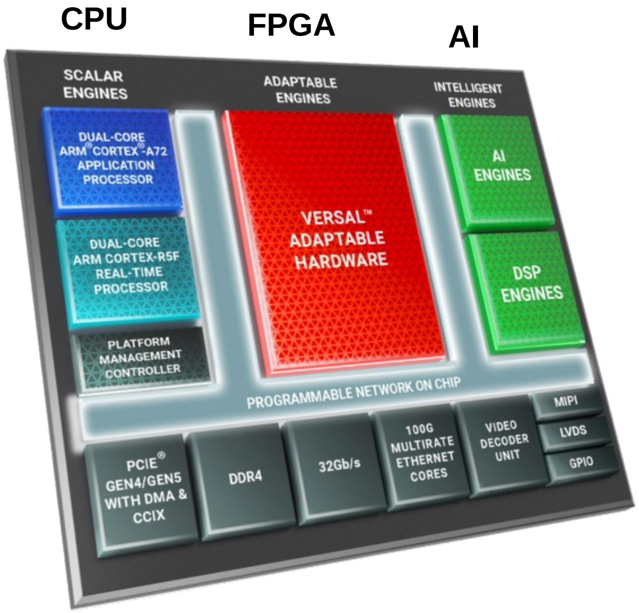
Prime Series

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

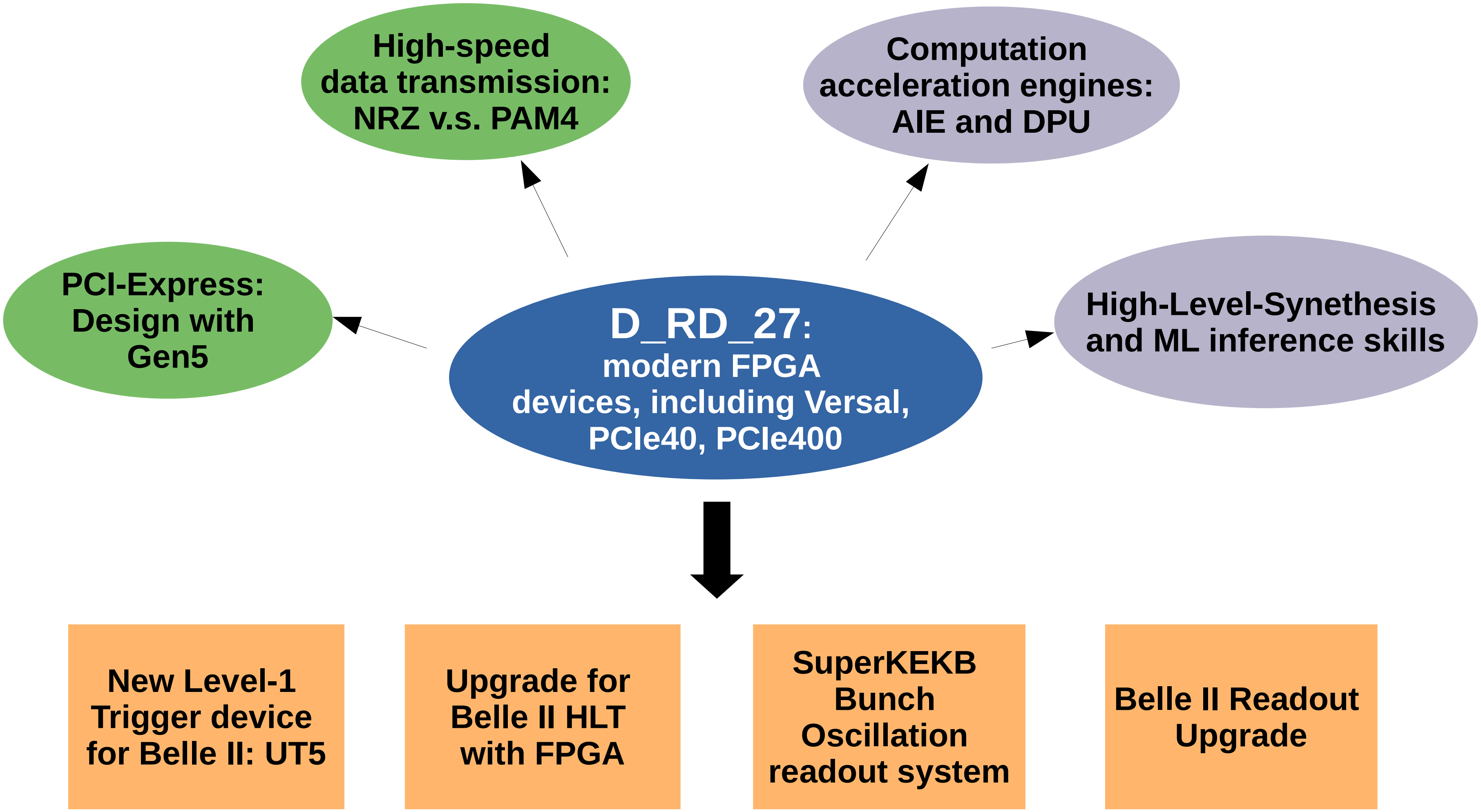
source: Xilinx website

D_RD_27 members

- Activities in 2025:
  - Japan → France: Visited CPPM Marseille for the progress on PCIe400 development and discussion on potential upgrade in Belle II.
  - France → Japan: Viisited KEK for the deployment of IDROGEN/WhiteRabbit in SuperKEKB system.

France			**Japan**		
Name	Institute		Name	Institute	
Daniel Charlet	IJCLab Orsay	PCIe readout device for Belle II / LHCb	Yun-Tsung Lai	KEK IPNS	E-sys, Belle II
Patrick Robbe			Manobu Tanaka		E-sys
Tak-Shun Lau			Makoto Tomoto		ATLAS
Emi Kou			Satoru Yamada		Belle II
			Yutaka Ushiroda		Belle II
			Kunihiro Nagano		ATLAS
			Taichiro Koga		Belle II
			Yu Nakazawa		Belle II
Julien Langouet	CPPM Marseille	PCIe400 readout upgrade	Hiroshi Kaji	KEK ACCL	SuperKEKB
Paul Bibron					
Renaud Le Gac					

Project overview



D_RD_27: modern FPGA devices, including Versal, PCIe40, PCIe400

PCI-Express: Design with Gen5

High-speed data transmission: NRZ v.s. PAM4

Computation acceleration engines: AIE and DPU

High-Level-Synethesis and ML inference skills

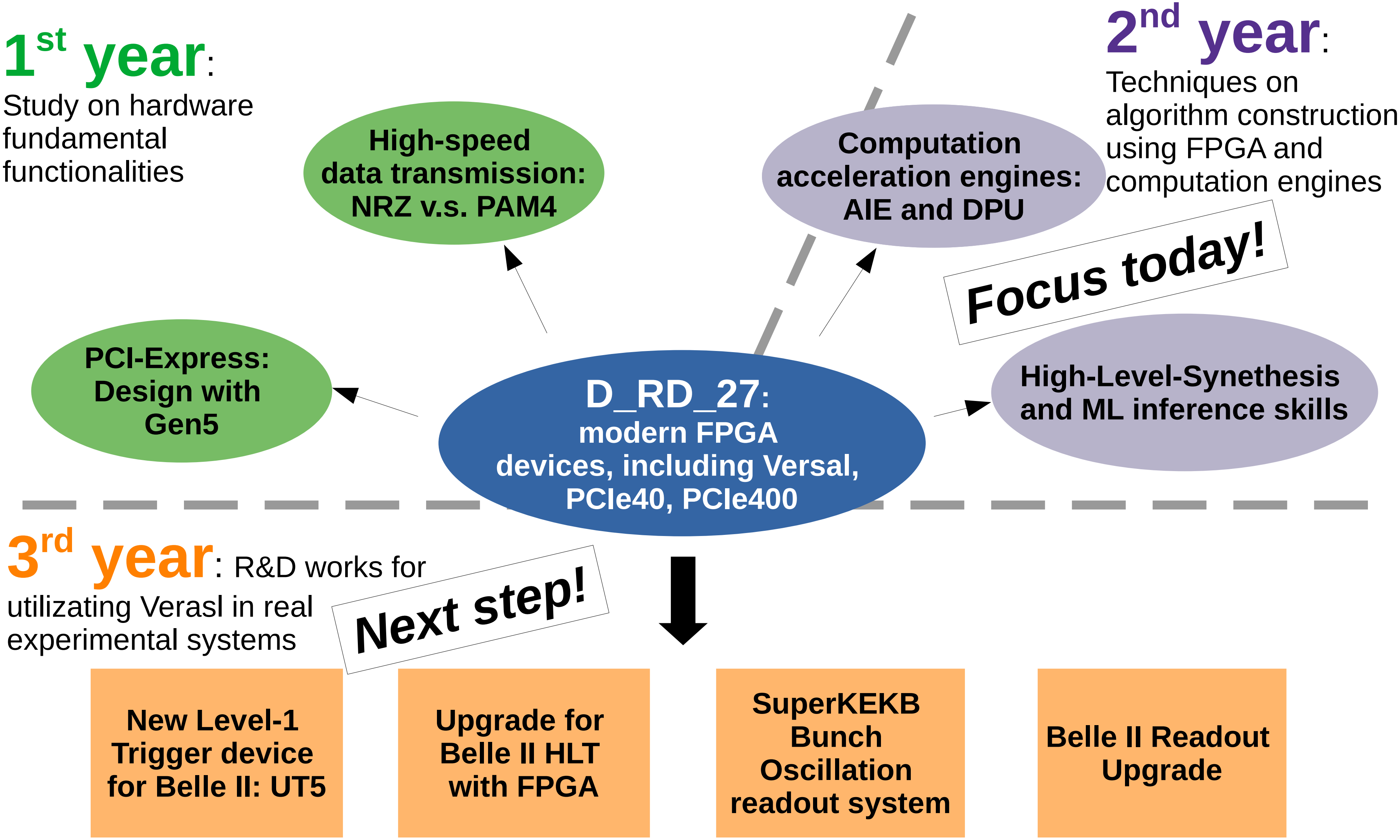
New Level-1 Trigger device for Belle II: UT5

Upgrade for Belle II HLT with FPGA

SuperKEKB Bunch Oscillation readout system

Belle II Readout Upgrade

Project overview and working plan



1st year: Study on hardware fundamental functionalities

High-speed data transmission: NRZ v.s. PAM4

PCI-Express: Design with Gen5

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

Computation acceleration engines: AIE and DPU

Focus today!

High-Level-Synthesis and ML inference skills

D_RD_27: modern FPGA devices, including Versal, PCIe40, PCIe400

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

Next step!

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

2nd year for D_RD_27: How to make algorithm in FPGA

- With consideration on the longer-term plan for application of such advanced FPGA devices:
  - DAQ readout/collection
  - Hardware (Level-1) **Trigger**
  - High-Level **Trigger** (HLT)

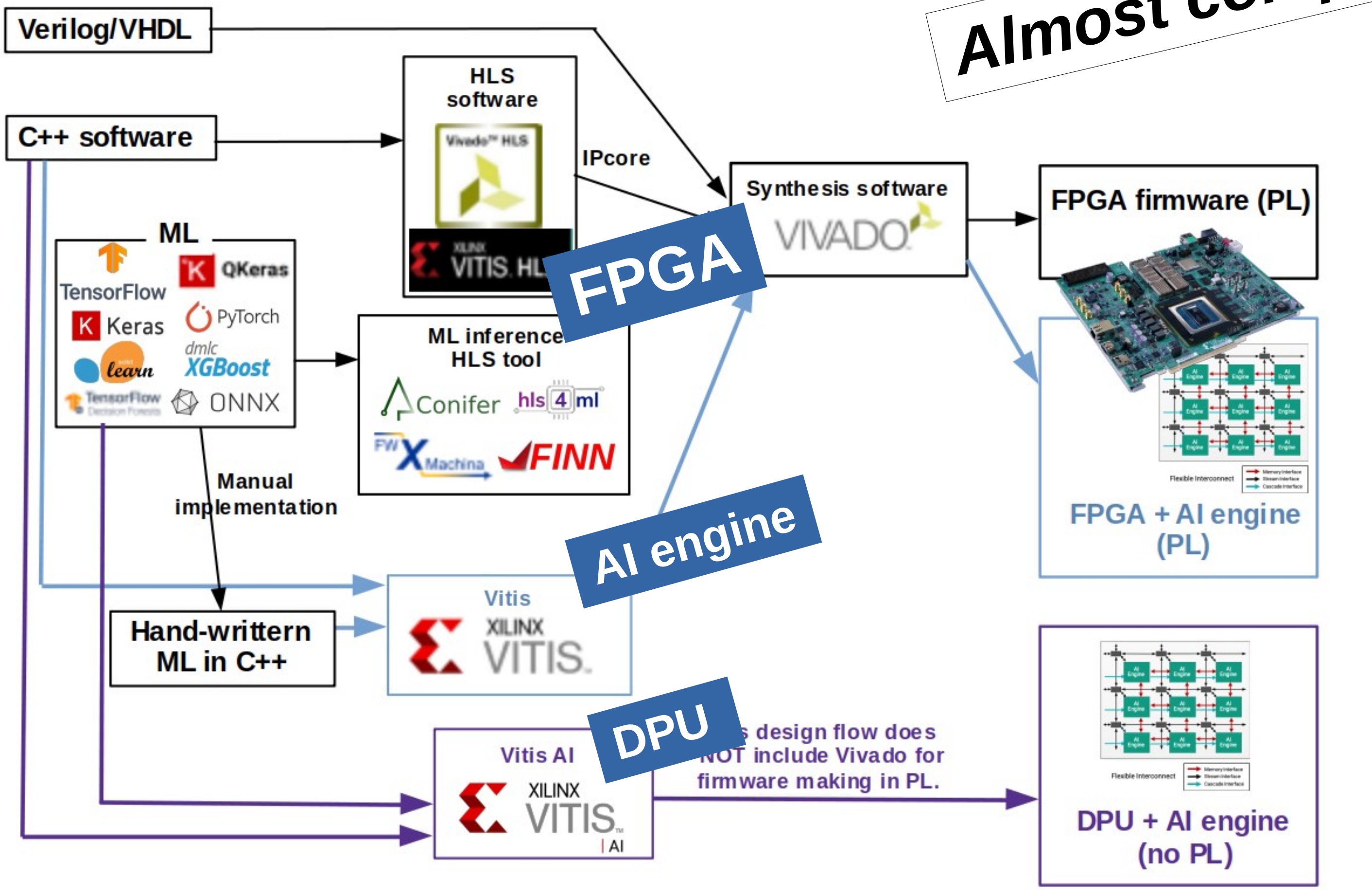
- **Trigger**: real-time data procession with **algorithm in FPGA**
  - For the **two types of trigger system (L1 and HLT)**, what is the major technical difference in terms of algorithm constrction and deploymeny?
  - Any benefit of utilizing new FPGA devices?
  - We have many kinds of logics to be tested with Versal. **But "not only what kind of logic to make, but also how to make it"**.
    - HDL/RTL
    - High-Level-Synthesis (HLS)
    - ML inference with HLS
    - Computation engines: AI engine and DPU

These are our major focus in the 2nd year and will be reported today.

HLS, ML, AI engine: roadmap of FPGA methodology

- Not only "what kind of logic to make", but also "how to make it".
- We hope to perform basic study on each of the items, collect experience, build a database of techinical knowledge, and prepare material to support our experimental colleagues.
  - We believe this kind of effort on fundamental technique is essential.
  - Hand-on lectures in this summer is under planning!

Almost complete!



Verilog/VHDL

C++ software

HLS software

IPcore

Synthesis software

FPGA firmware (PL)

ML

ML inference HLS tool

Manual implementation

Hand-writtern ML in C++

Vitis

Vitis AI

FPGA

AI engine

DPU

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

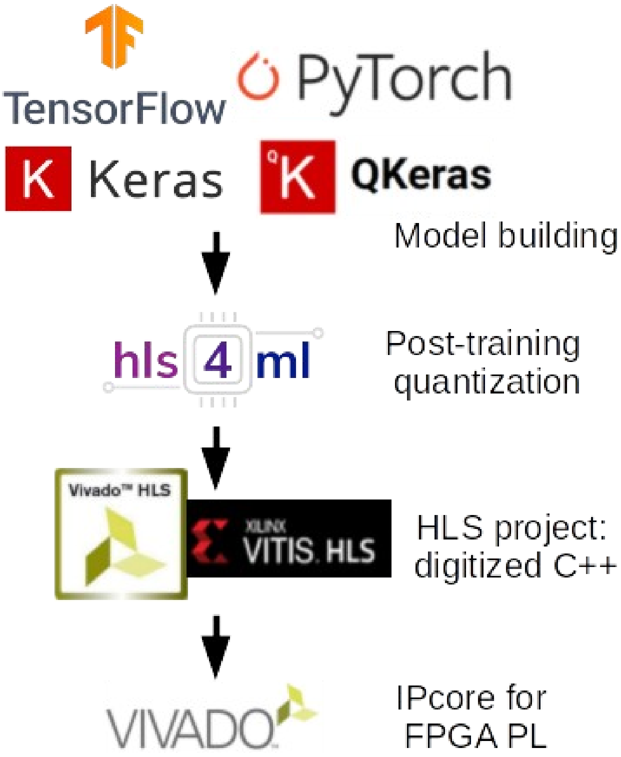
FPGA + AI engine (PL)

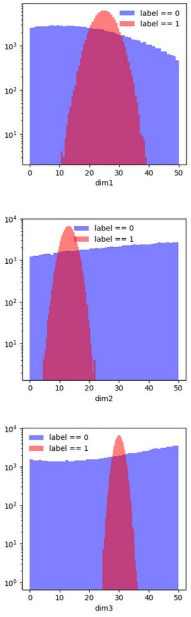
DPU + AI engine (no PL)

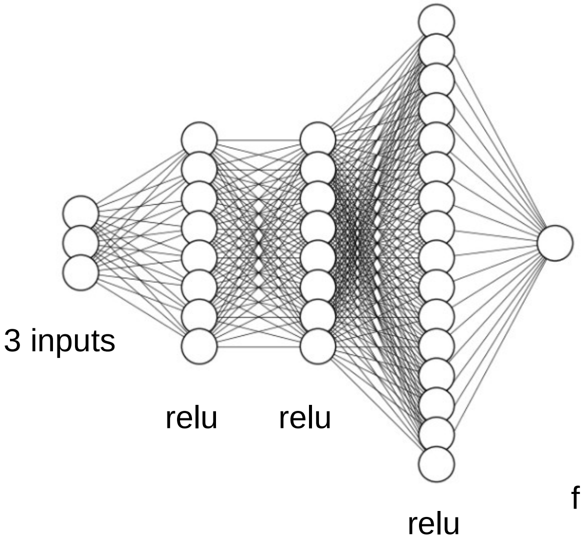
hls4ml

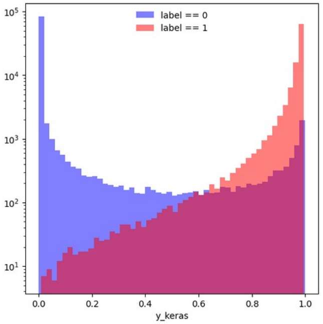
- hls4ml has been widly utilized in our field already.
  - For TensorFlow and Pytorch

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

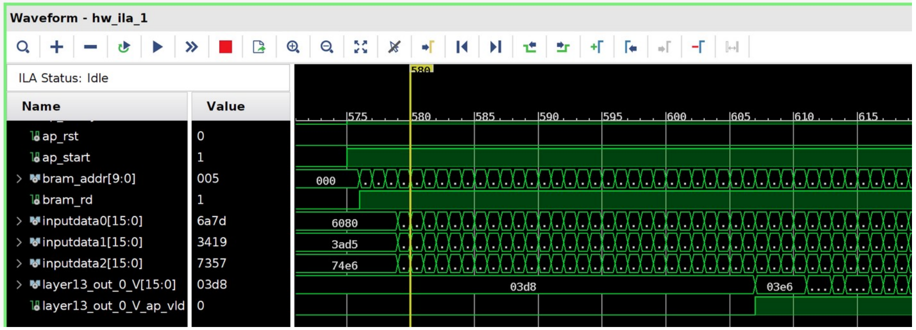








sigmoid
for binary separation


Waveform - hw_ila_1

ILA Status: Idle

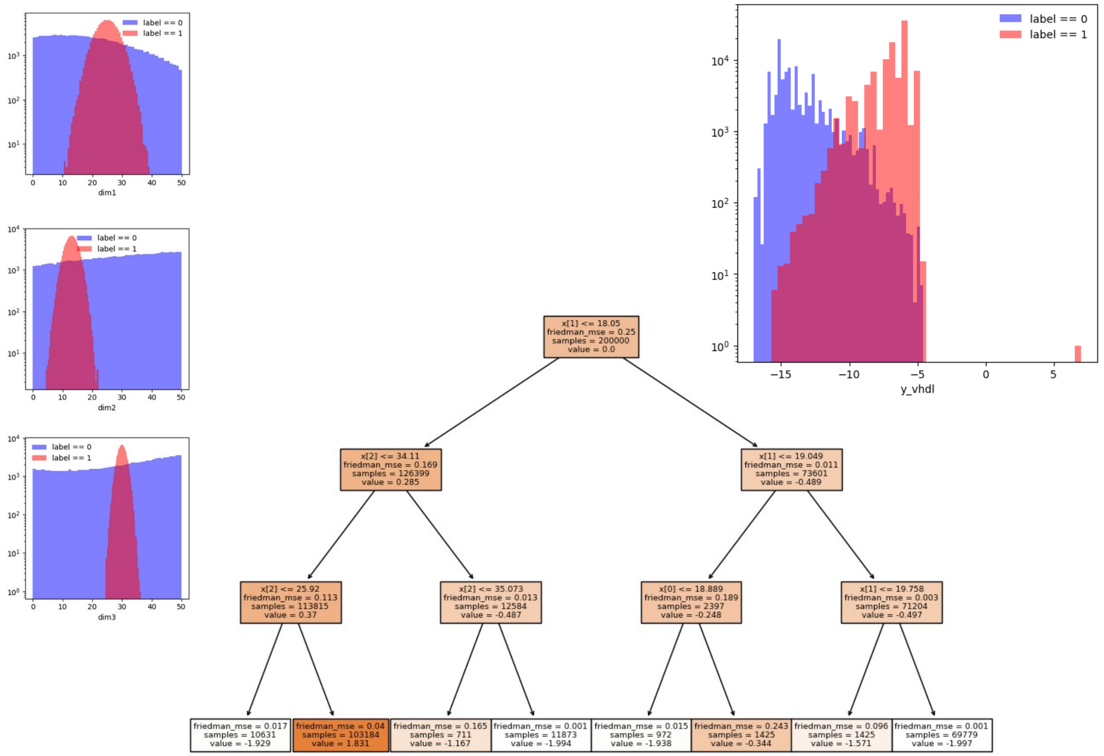
Name	Value
ap_rst	0
ap_start	1
bram_addr[9:0]	005
bram_rd	1
inputdata0[15:0]	6a7d
inputdata1[15:0]	3419
inputdata2[15:0]	7357
layer13_out_0_V[15:0]	03d8
layer13_out_0_V_ap_vld	0

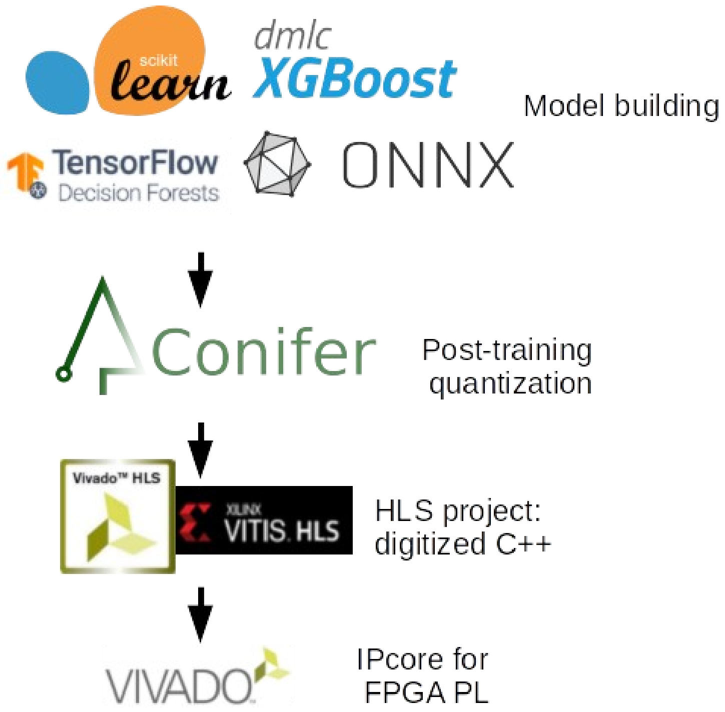
Latency: O(10) clock-cycles

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.





Model building

Post-training quantization

HLS project: digitized C++

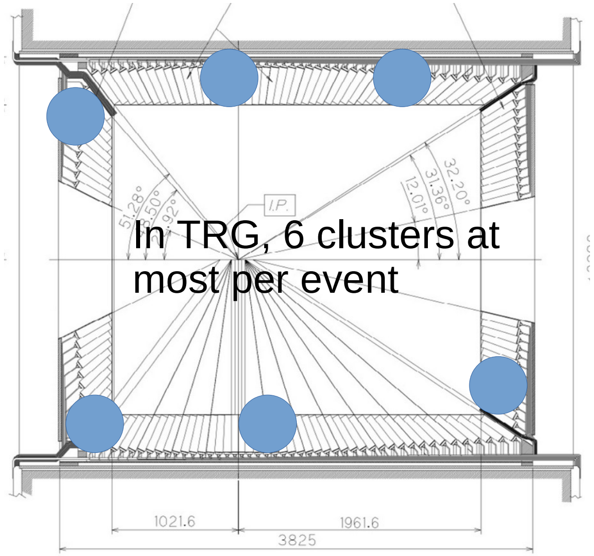
IPcore for FPGA PL

Belle II τ trigger: NN v.s. BDT

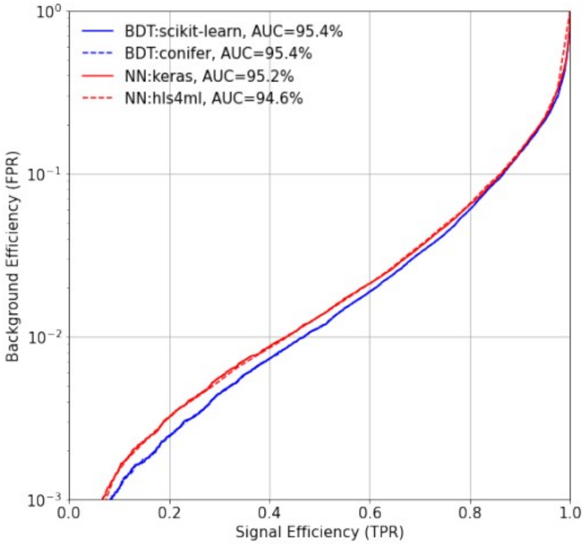
- Example: Belle II τ event trigger with calorimeter cluster
  - Input: clusters' position and energy
  - Output: Y/N for a $e^+e^- \rightarrow \tau^+\tau^-$ event
  - Original design is based on NN+hls4ml.

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

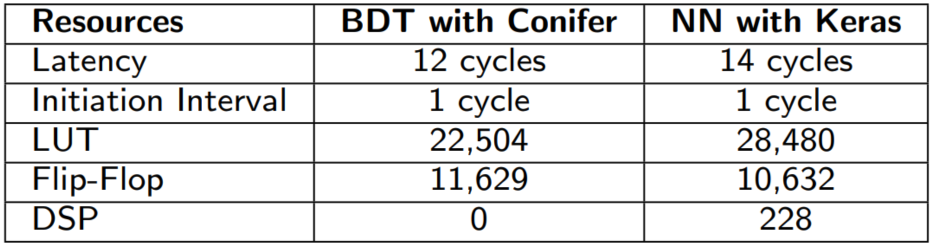
R. Nomaru (Univ. of Tokyo)



In TRG, 6 clusters at most per event



YongHeon Ahn (Korea Univ.)

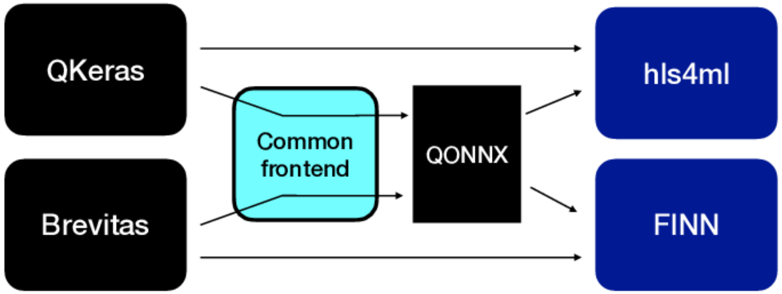


Resources	BDT with Conifer	NN with Keras
Latency	12 cycles	14 cycles
Initiation Interval	1 cycle	1 cycle
LUT	22,504	28,480
Flip-Flop	11,629	10,632
DSP	0	228

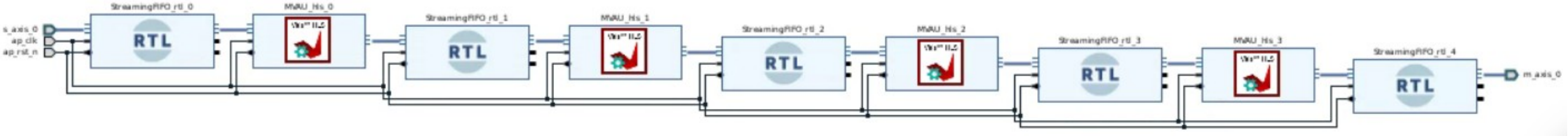
New study: FINN

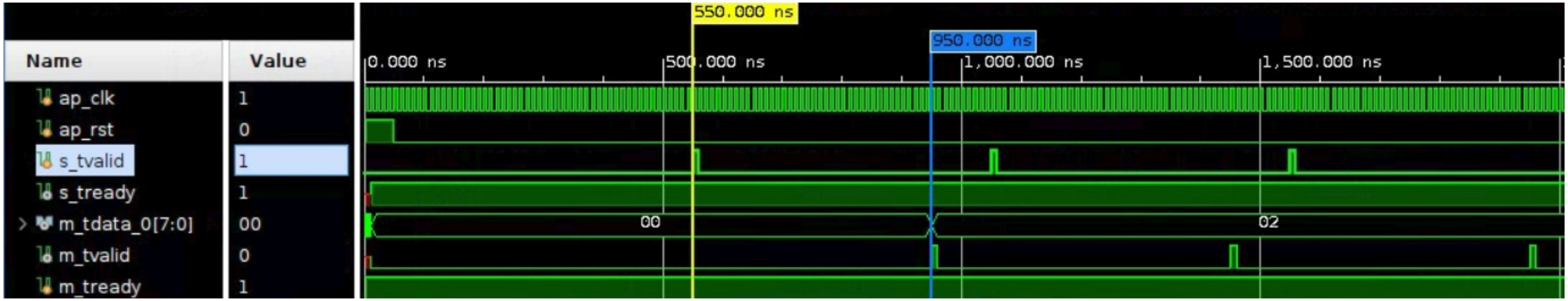
- Under development by AMD Xilinx.
- The core concept is matrix multiplication.
- Quantization based on Pytorch + Brevitas.
- Model representation by ONNX/QONNX.

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



source: 10.48550/arXiv.2206.11791



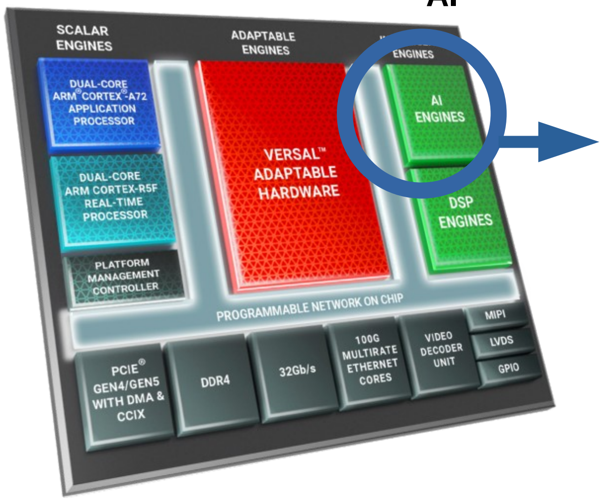


Name	Value
ap_clk	1
ap_rst	0
s_tvalid	1
s_tready	1
m_tdata_0[7:0]	00
m_tvalid	0
m_tready	1

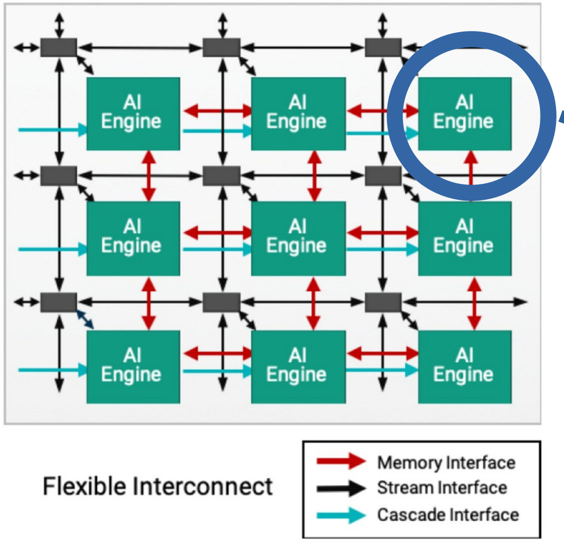
Versal AI engine

Versal ACAP

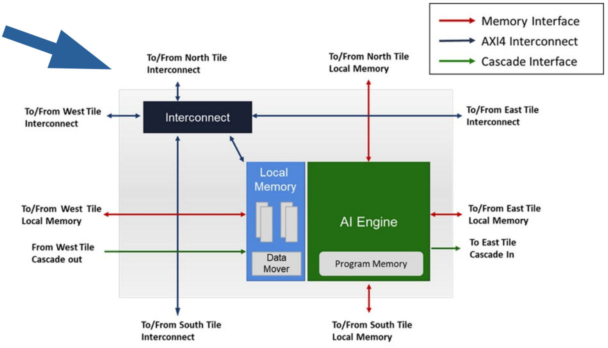
CPU FPGA AI


SCALAR ENGINES
DUAL-CORE ARM® CORTEX®-A72 APPLICATION PROCESSOR
DUAL-CORE ARM CORTEX-R5F REAL-TIME PROCESSOR
PLATFORM MANAGEMENT CONTROLLER
ADAPTABLE ENGINES
VERSAL™ ADAPTABLE HARDWARE
ENGINES
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

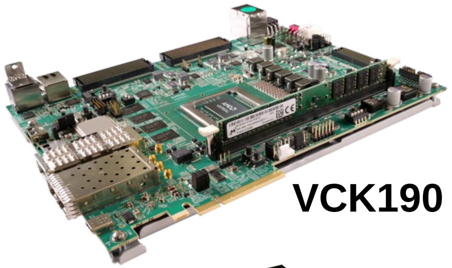
Versal AI engine



AI engine "tile"


Memory Interface
AXI4 Interconnect
Cascade Interface
To/From North Tile Interconnect
To/From North Tile Local Memory
To/From West Tile Interconnect
Interconnect
To/From East Tile Interconnect
Local Memory
AI Engine
To/From West Tile Local Memory
To/From East Tile Local Memory
From West Tile Cascade out
To East Tile Cascade In
Data Mover
Program Memory
To/From South Tile Interconnect
To/From South Tile Local Memory

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



VCK190 with AIE

VEK280 with AIE-ML

You can refer to our mini-WS for many study results: https://kds.kek.jp/event/53369/

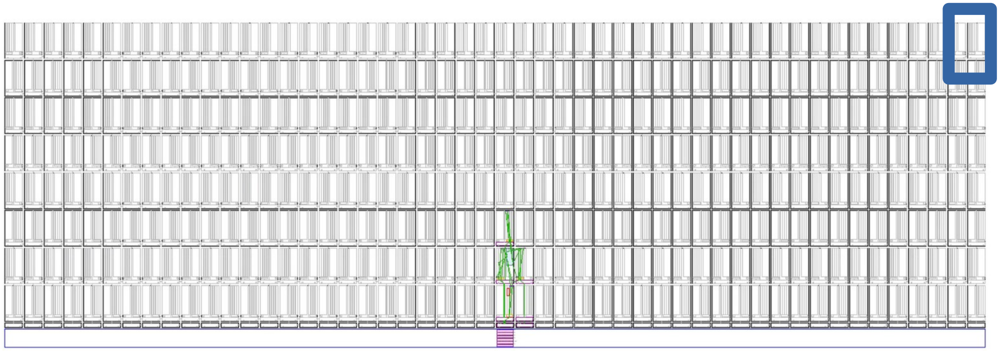
Know-how of ML in AIE

- 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.**

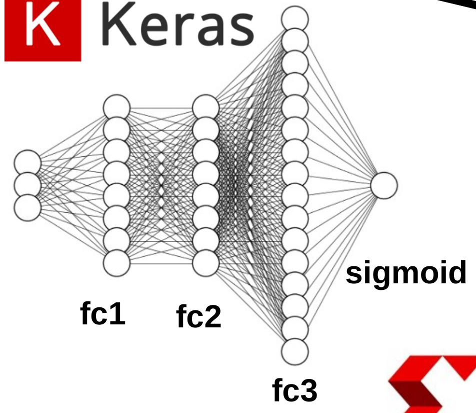
- **Latency: 3.4 μs.**
  - Can be further reduced.

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



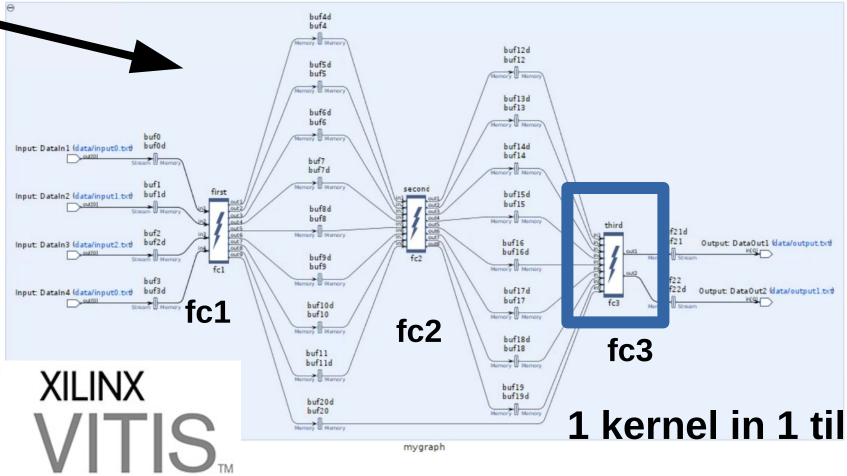
TensorFlow

Keras



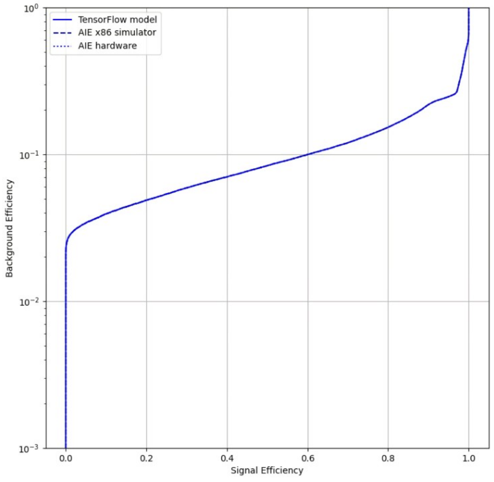
Exact math form
in C++ in float

XILINX VITIS™



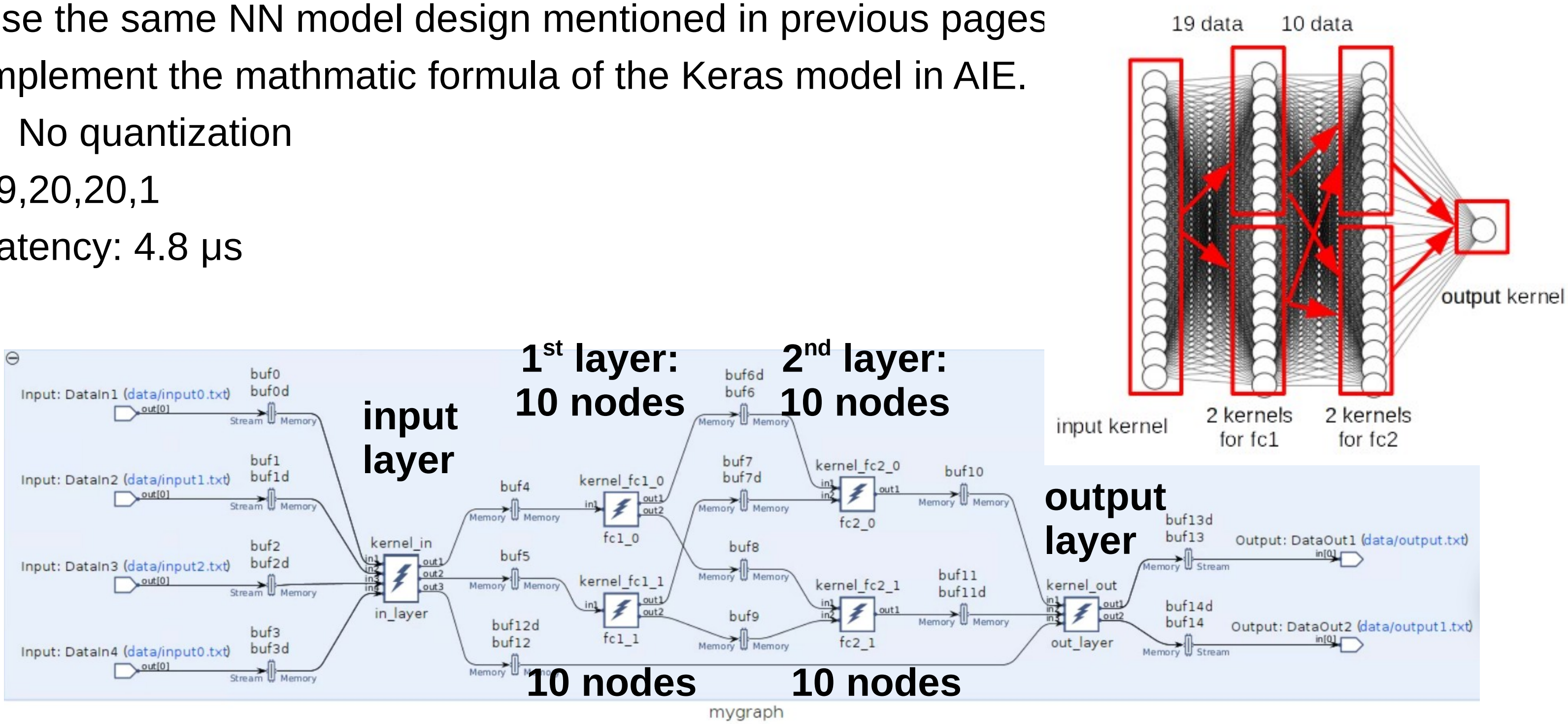
Input: DataIn1 (data/input0.txt)
Input: DataIn2 (data/input1.txt)
Input: DataIn3 (data/input2.txt)
Input: DataIn4 (data/input0.txt)
first
second
third
Output: DataOut1 (data/output.txt)
Output: DataOut2 (data/output1.txt)
mygraph
fc1
fc2
fc3

1 kernel in 1 tile



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



input layer

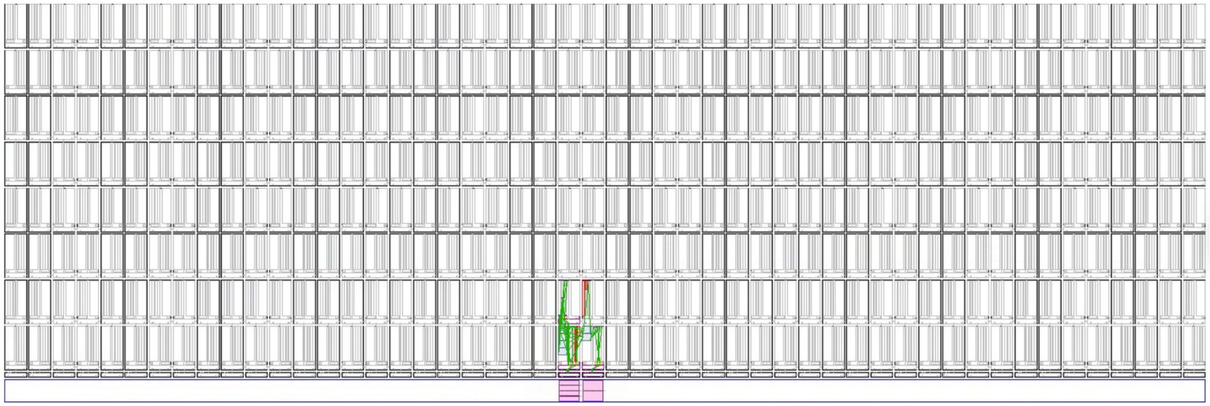
1st layer: 10 nodes

2nd layer: 10 nodes

output layer

10 nodes

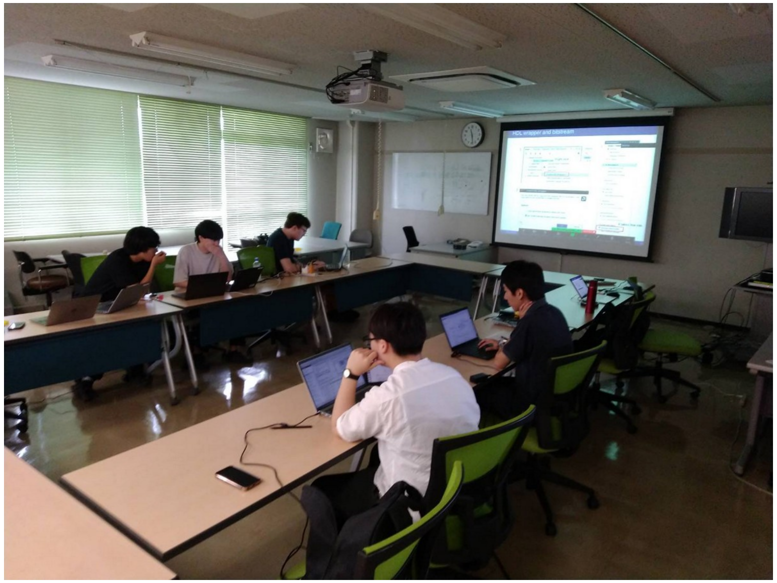
10 nodes

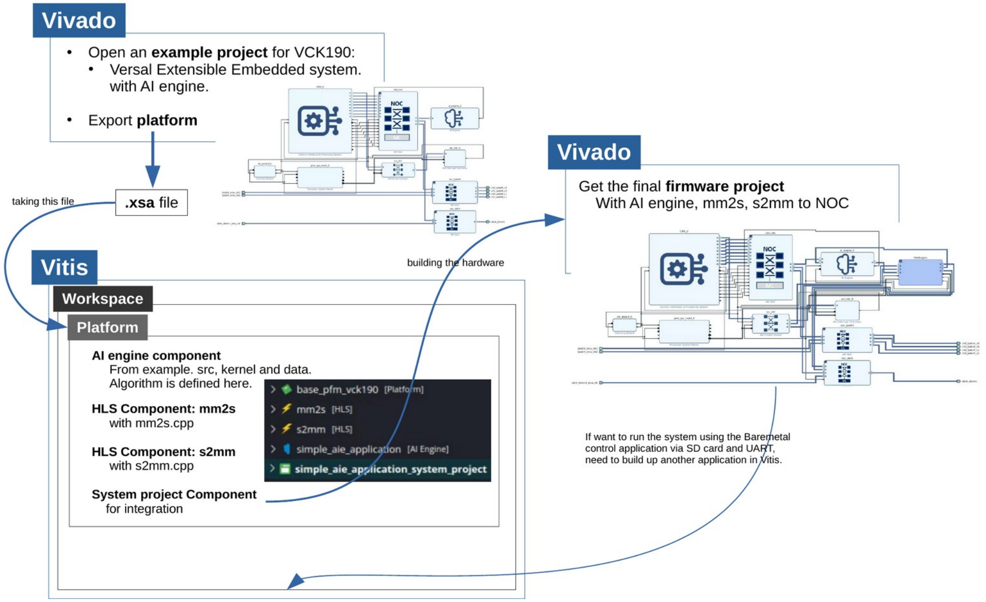


AI engine course @ KEK E-sys, in Aug. 2024

- We tried to hold a course about utilization the AI engine in this Aug. at KEK.
  - Attenders in Japan and other countries remotely.
  - All operations were done using local servers in our laboratory.
  - Almost everything is covered: environment setup, Vivado design flow, Vitis design flow, kernel making, Vitis simulation, hardware test.

- This is our first step. Other courses in summer are under planning.





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

building the hardware

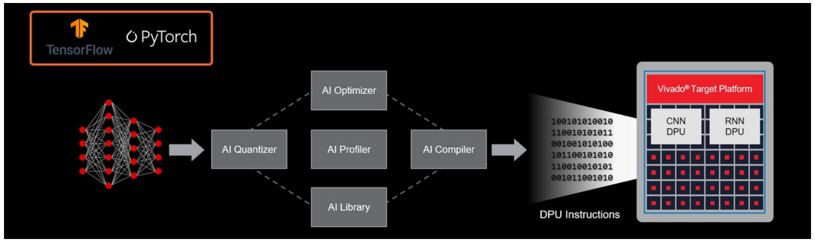
Vivado

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

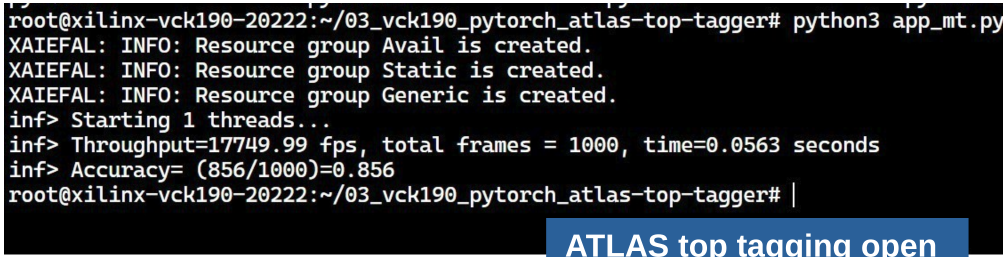
If want to run the system using the Baremetal control application via SD card and UART, need to build up another application in Vitis.

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

Versal DPU

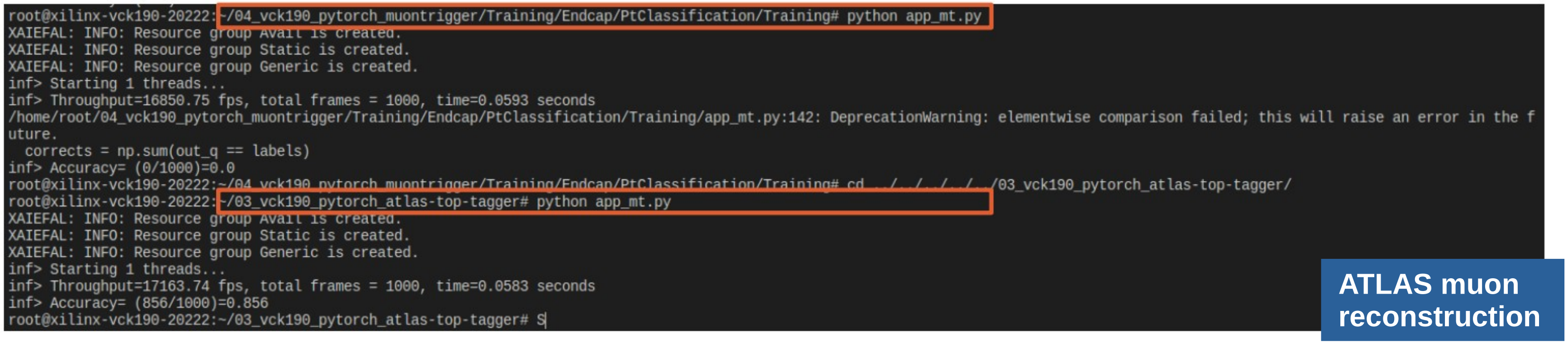


- **DPU**: Deep Learning Processing Unit
  - Configurable computation engine dedicated to CNN
- DPU takes leverage of the FPGA resource, while the artificial networks inference **does not require touching FPGA PL**.
  - Network model building by Pytorch, and quantization by Vitis-AI software. Independent of FPGA.
  - FPGA is served like a server. Operate everything with CUI.
  - **Hardware acceleration for high-level application**.



```
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

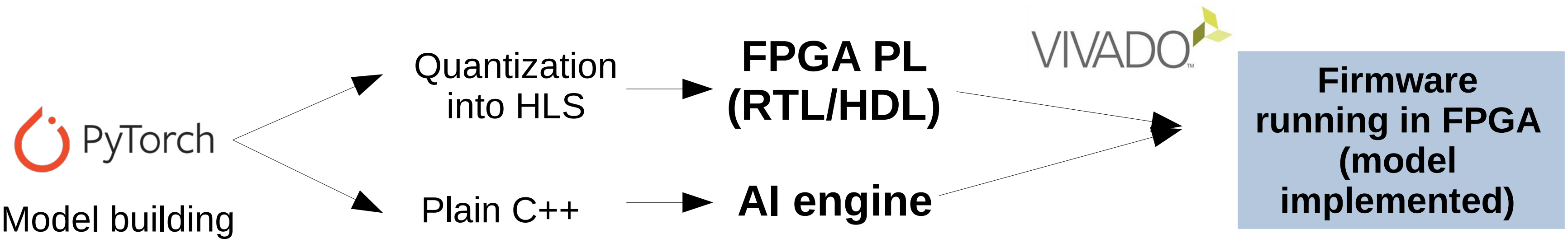


```
root@xilinx-vck190-20222:~/04_vck190_pytorch_muontrigger/Training/Endcap/PtClassification/Training# python app_mt.py
XAIEFAL: INFO: Resource group Avail is created.
XAIEFAL: INFO: Resource group Static is created.
XAIEFAL: INFO: Resource group Generic is created.
inf> Starting 1 threads...
inf> Throughput=16850.75 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

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

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



PyTorch

Model building

Quantization into HLS

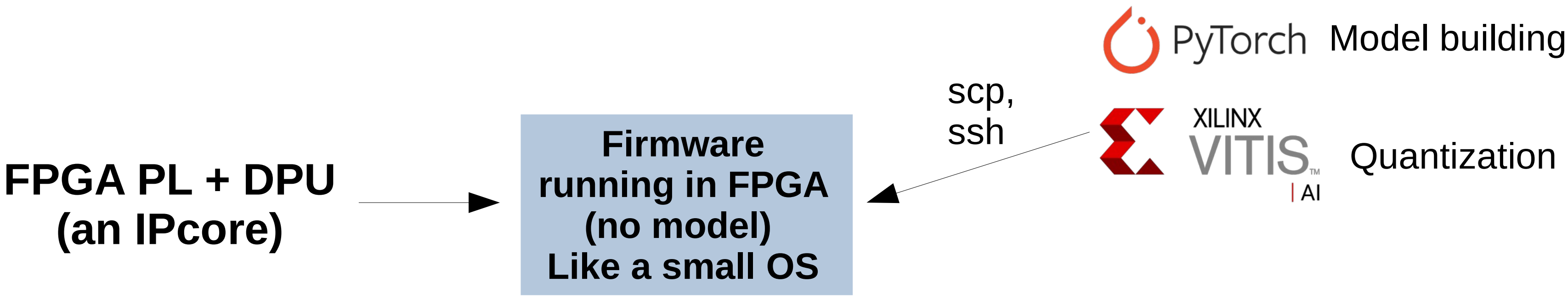
Plain C++

FPGA PL (RTL/HDL)

AI engine

VIVADO

Firmware running in FPGA (model implemented)



PyTorch Model building

XILINX VITIS | AI Quantization

scp, ssh

FPGA PL + DPU (an IPcore)

Firmware running in FPGA (no model) Like a small OS

- 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.

Belle II Level-1 Trigger board upgrade: UT5

Belle II UT3

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

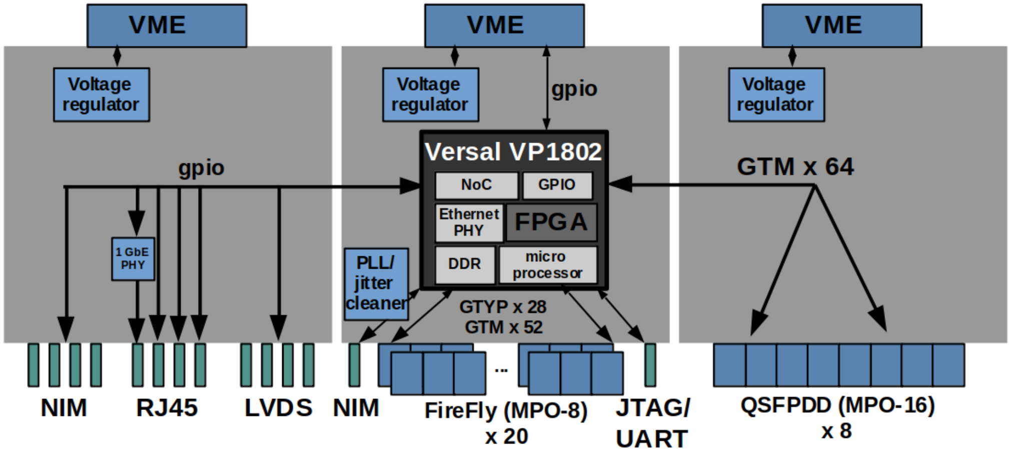
Belle II UT4

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



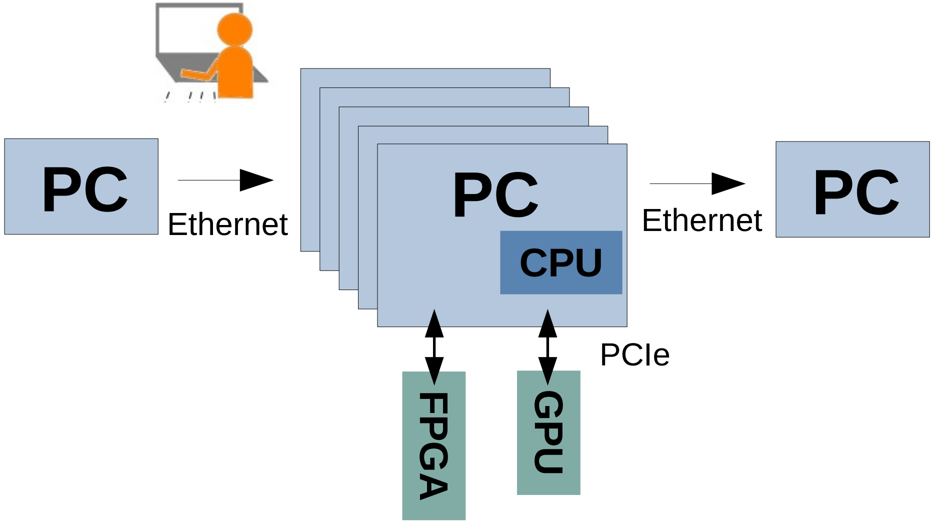
- Trying to use other than QSFP28 (FireFly, etc) with smaller form factor.
- QSFP-DD for PAM4 in daughter board.
- Versal has Processing System (PS)
- Still VME for SLC
- Prabably no AI engine in UT5
  - But we are still open for the potential for UT6
- Aiming for prototyping in 2026

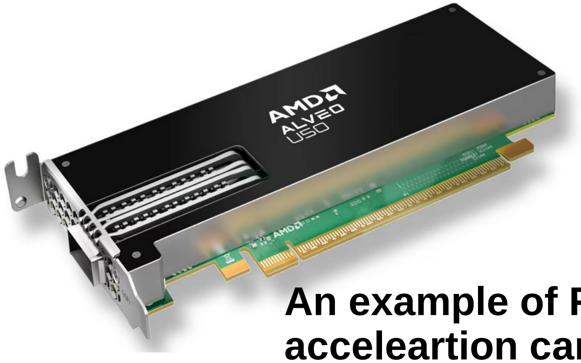
Other than CPU for HLT?

- 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**"

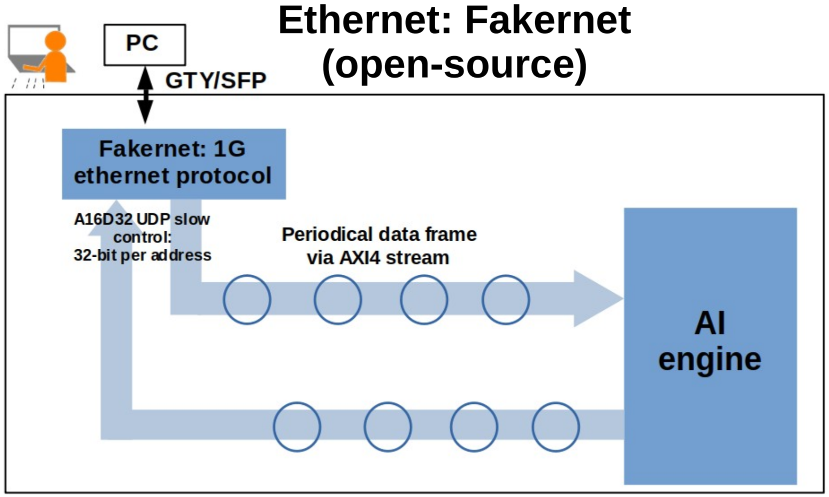


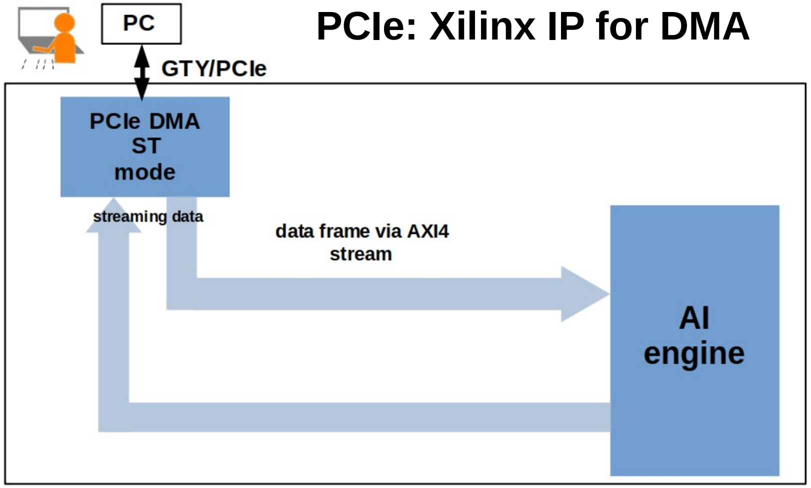


An example of FPGA acceleartion card: AMD Alveo U50

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.





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

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

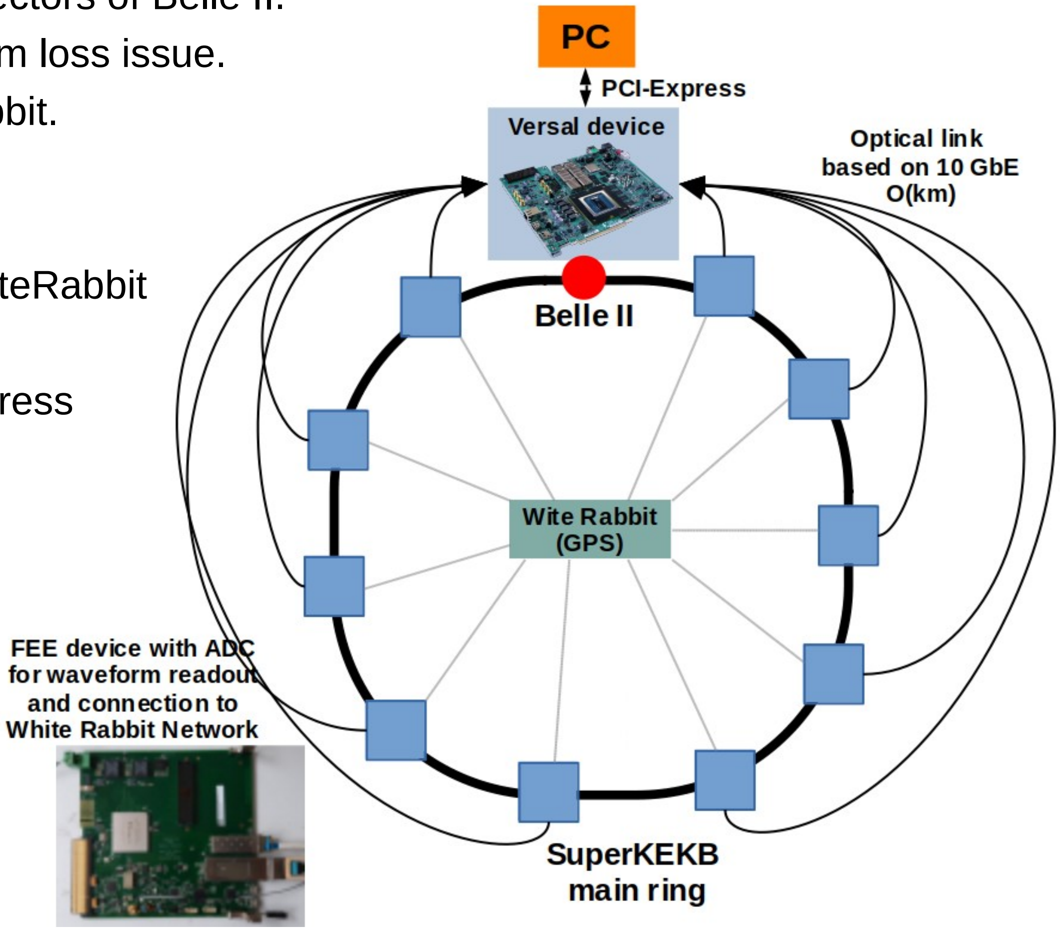
→ **Potential for HLT application.**

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, KEK ACCL, KEK E-sys, and IJCLab.



PC

PCI-Express

Versal device

Optical link based on 10 GbE O(km)

Belle II

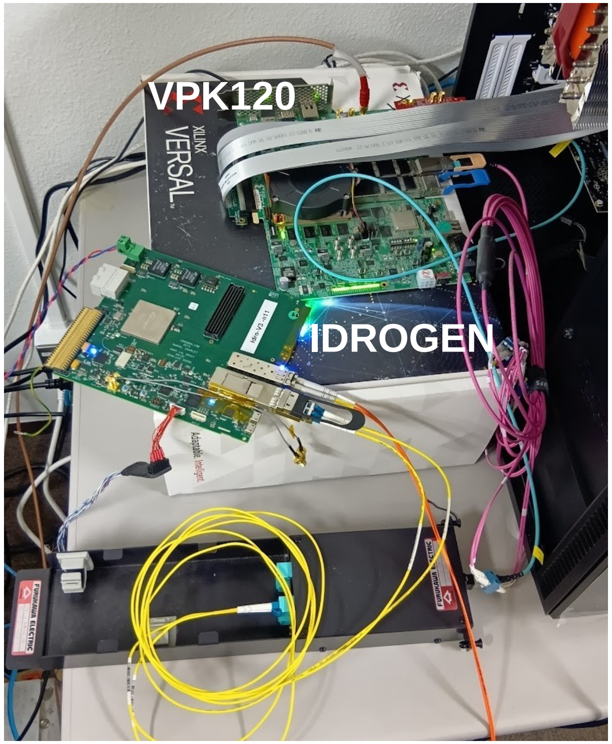
Wite Rabbit (GPS)

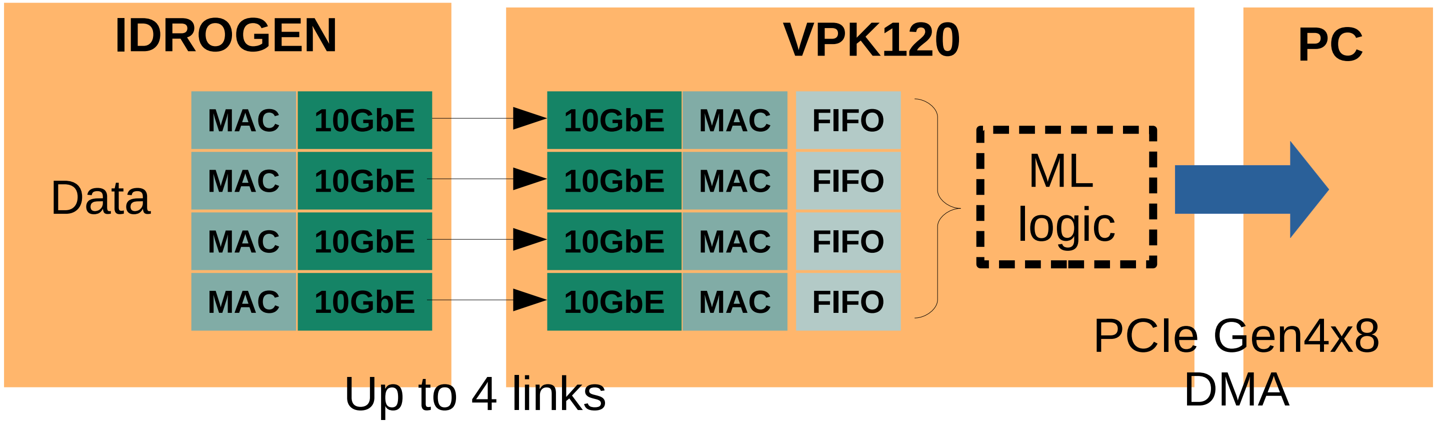
FEE device with ADC for waveform readout and connection to White Rabbit Network

SuperKEKB main ring

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.

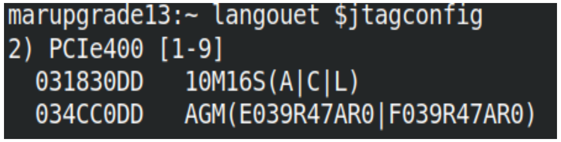




New PCIe device for readout upgrade

source: CPPM Marseille group

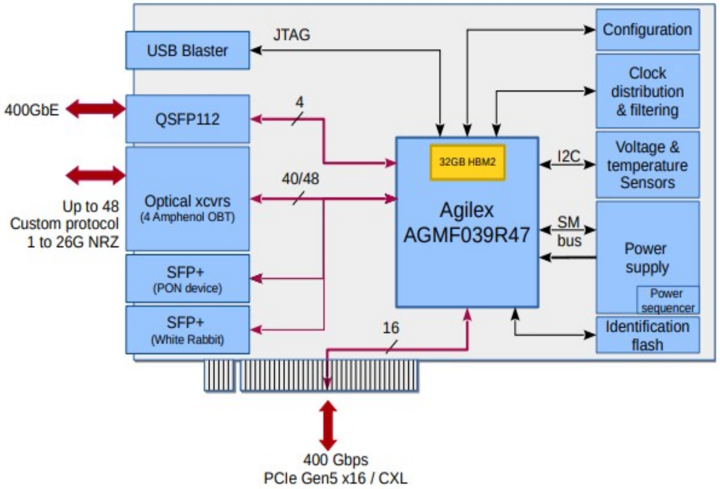
- CPPM Marseille group has received the first prototype of PCIe400 in Jan. 2025.
  - 2 boards + 1 partially equiped with power supply
  - Validation and debugging are ongoing
  - Agilex FPGA can be booted correctly:
  - Aiming for next prototype next year
  - Planning for LHCb upgrade



```
marupgrade13:~ langouet $jtagconfig
2) PCIe400 [1-9]
  031830DD   10M16S(A|C|L)
  034CC0DD   AGM(E039R47AR0|F039R47AR0)
```

- During our visit at CPPM Marseille, we discussed the possibility of Belle II future upgrade.
  - Belle II has just finished the upgrade and commisionning with PCIe40 in 2024.
  - Versal board has similar spec (Gen5 x 16). We are also working on the continuous readout design and throughput test for Versal.

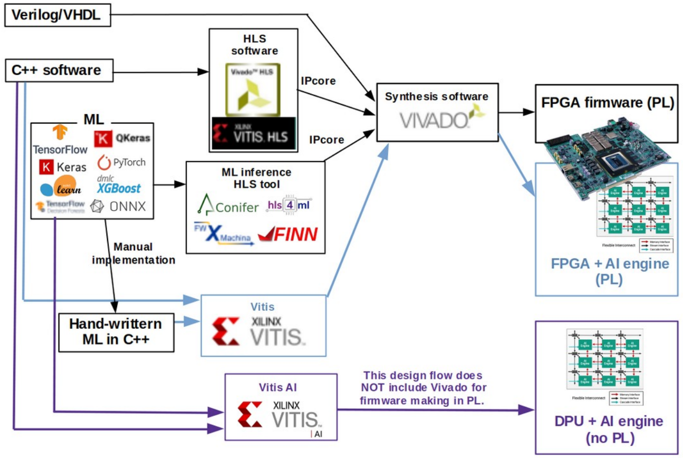
PCIe400 synoptic



Summary

- In our project of D_RD_27, we study the modern FPGA devices for their potential application in experimental HEP for future upgrade.
  - Mainly based on Versal, and also other new PCIe devices.

- In the second year, we focused on the techniques of FPGA algorithm construction for trigger purpose.
  - Not only "what kind of logic to make",
    but also "how to make it".
    - HLS
    - ML inference
    - Computation engines
  - We built up a database of technical knowledge, implemented with exsiting logics in Belle II, and also provided education.

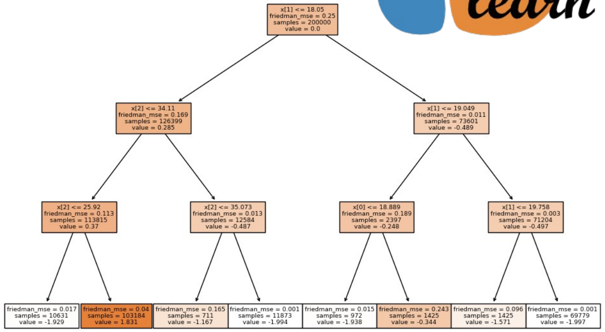


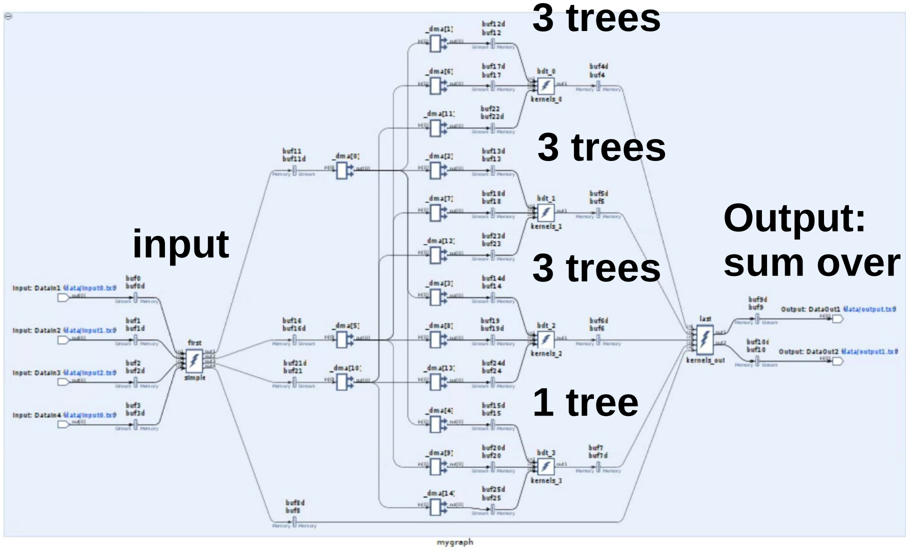
- For our next step, we plan to consider the potential utilization of the devices in our experimental systems in different aspects:
  - L1 trigger: Belle II UT5 upgrade
  - HLT: adoption of FPGA in HLT
  - Readout: SuperKEKB Bunch Oscillation Readout and Belle II readout. Together with PCIe400.

Backup

ML in AIE: BDT

- BDT: Basically a large nested structure of if-else
- Using scikit-learn for model building.
    - Input = 3, N_estimator = 10, depth = 3.
- Parallel kernels for separated estimators, then sum over all the outputs.
- Latency: 2.8 μs




Input: DataIn1 (data/input0.txt)
Input: DataIn2 (data/input1.txt)
Input: DataIn3 (data/input2.txt)
Input: DataIn4 (data/input0.txt)
first
simple
bdt_0
kernels_0
bdt_1
kernels_1
bdt_2
kernels_2
bdt_3
kernels_3
last
kernels_out
Output: DataOut1 (data/output.txt)
Output: DataOut2 (data/output1.txt)
mygraph

3 trees

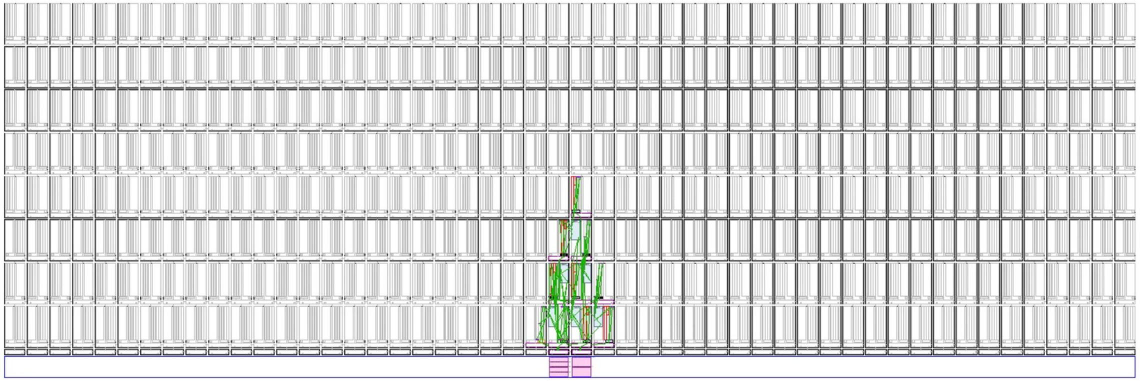
3 trees

input

3 trees

1 tree

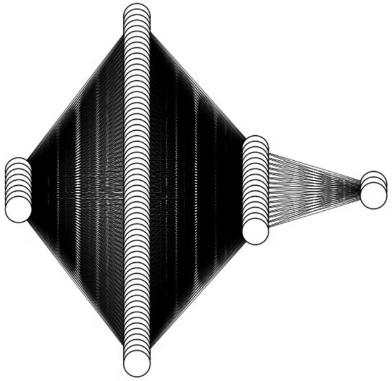
Output:
sum over all trees' output

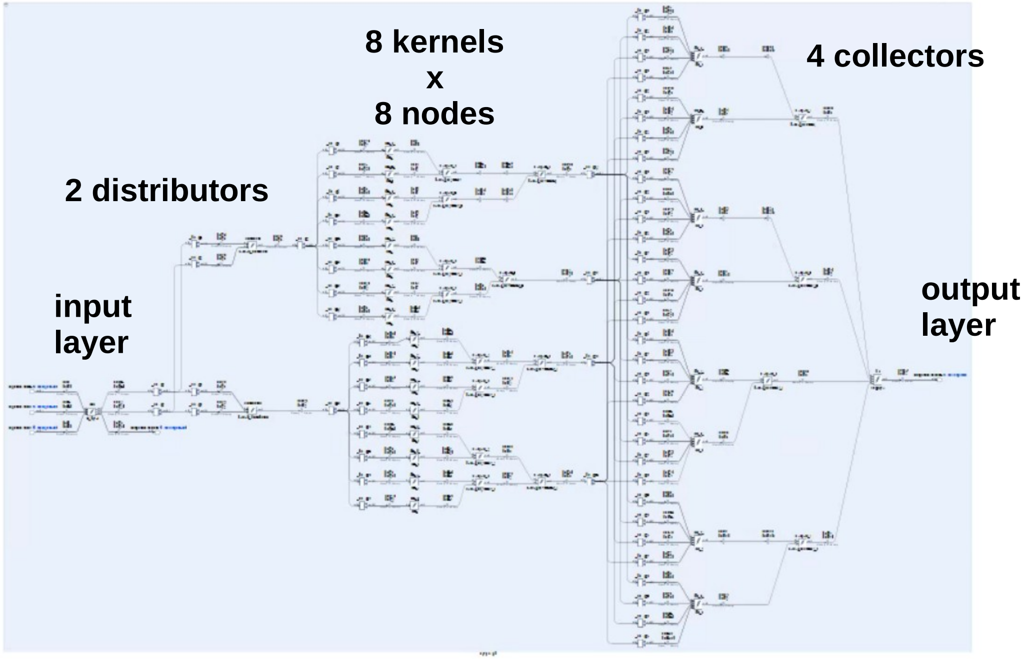


ML in AIE: KLMTRG NN

- Use the pre-tained network by Anthony, then implement the mathmatic formula in AIE.
- 8,64,16,3
  - Hidden layers use tanh. Output layer uses softmax.
- Complicated design!
- Latency: 10 μs

Previous study by Anthony Little (Univ. of Sydney)




8 kernels
x
2 nodes
8 kernels
x
8 nodes
4 collectors
2 distributors
input
layer
output
layer