Modern Electronics for Physicists: Project-Based Learning in PHYS476

Keisuke Yoshihara,

Department of Physics and Astronomy

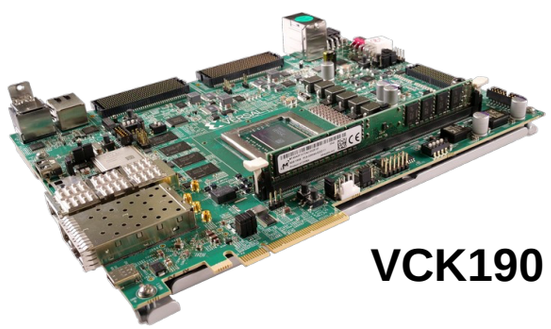
Special thanks to Yun-Tsung Lai, Matt Andrew, Chris Ketter

May19, ML4FE Workshop at University of Hawaii at Manoa

Why AI/ML Education Now?

- **AI/ML is transforming science and industry**
  - HEP: AI/ML used in real-time event triggering
  - Medical: CNNs deployed for on-device diagnosis
  - Industrial robots: AI/ML used for safety-critical control
- **STEM students need more than just Python skills**
  - Hardware integration (FPGA, embedded systems)

    — can it run in 1μs on FPGA?
- **Education must catch up with edge & scientific AI/ML**
  - Modern AI/ML chip requires a unique development platform (e.g., Versal AI Engine, see Yun-Tsung's slides)

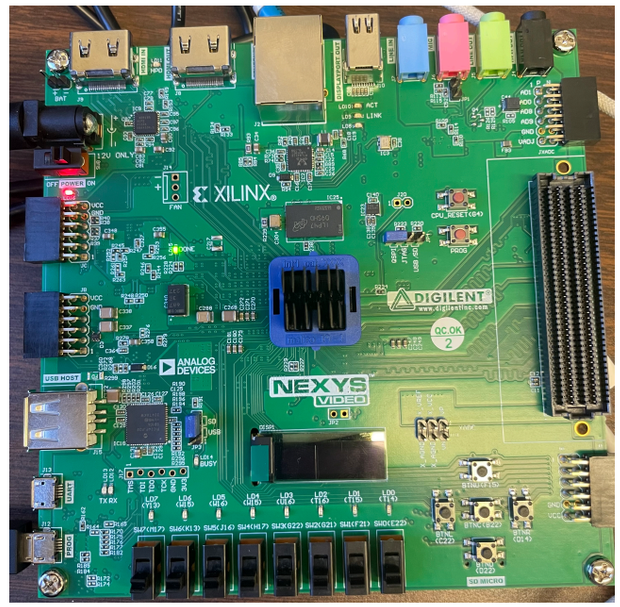


VCK190

Course Overview - PHYS476

- **Upper-division course for physics majors**
  - Undergraduate, graduate, and postdoc
  - Weekly lecture + lab (Tuesday and Thursday, 3 hrs each)

- **From digital circuit design to AI/ML inference on FPGA**
  - First half: logic design (Verilog), simulation & synthesis with FPGA training board.
  - Second half: AI/ML design and optimization



NEXYS-VIDEO

- **Goal: *AI/ML x hardware* literacy for physicists**

Design Tools

1. Digital Circuit Design

VIVADO + VITIS (software development)

2. Neural Network Design

Keras/QKeras
(Tensorflow as backend)

3. hls4ml IP Generation

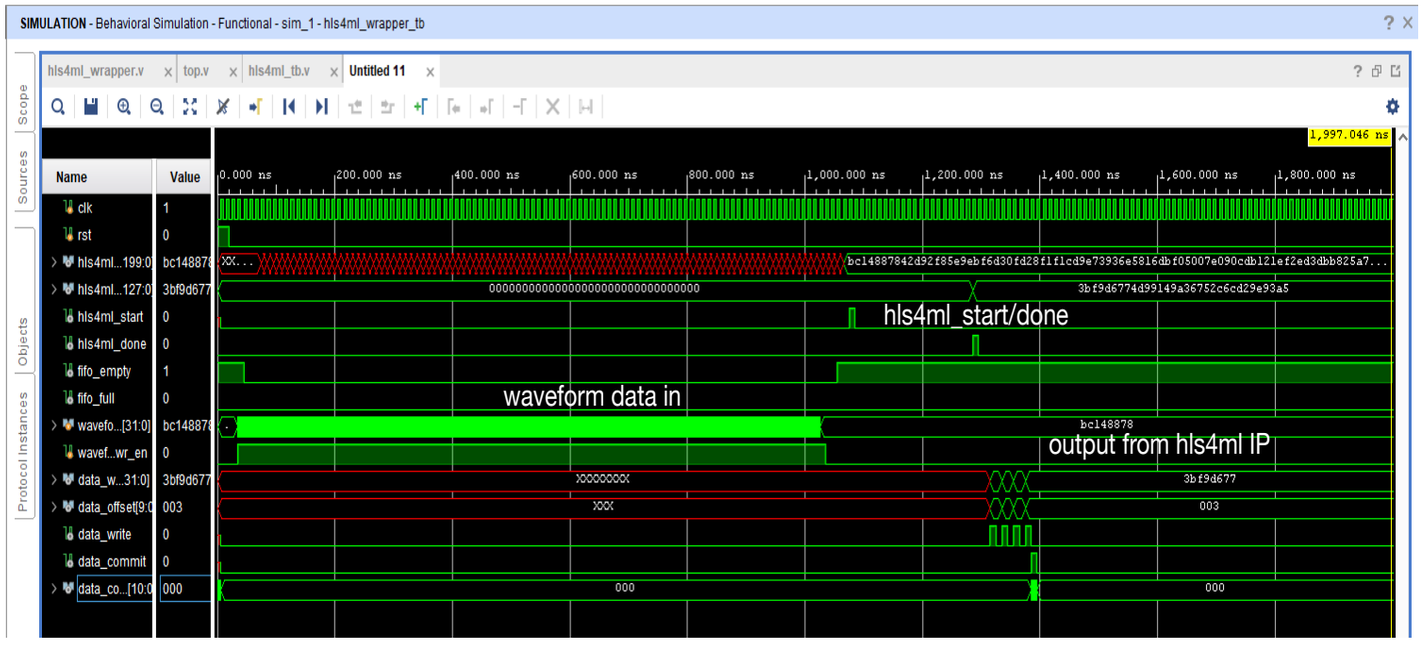
Keras → HLS (C++)

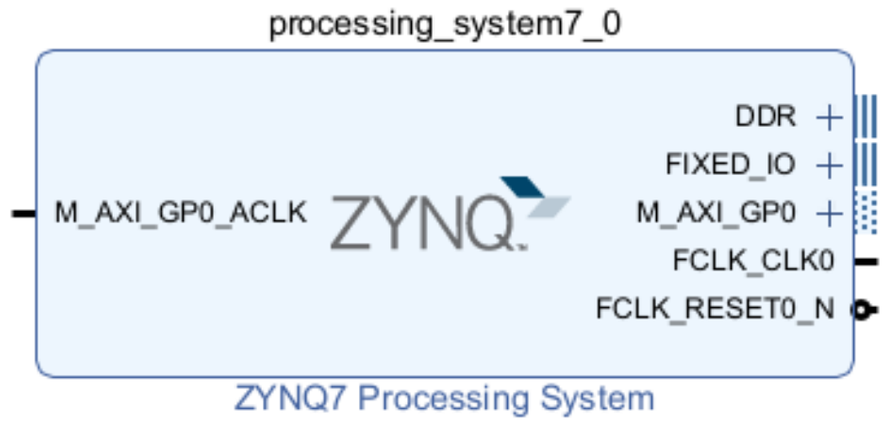
HLS → RTL

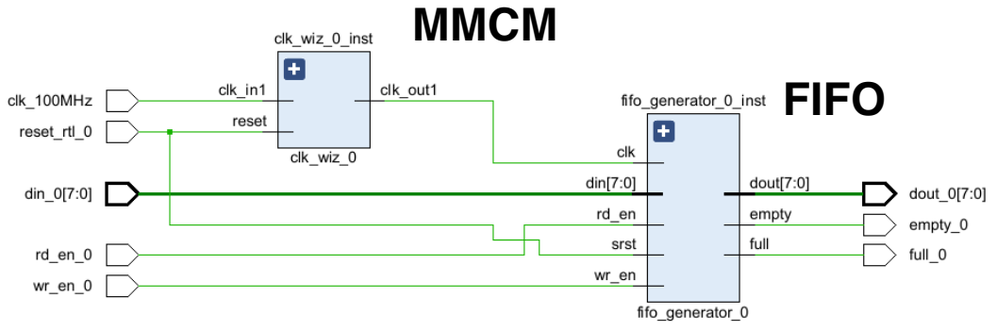
Hands-on (Circuit Design)

- Verilog-based logic circuits:

  Counters, Timers, Clock (PLL/MMCM), FSMs, FIFOs, Ethernet I/O
- Embedded system: MicroBlaze, ZYNQ, SoC
- Vivado **simulation** and **synthesis**





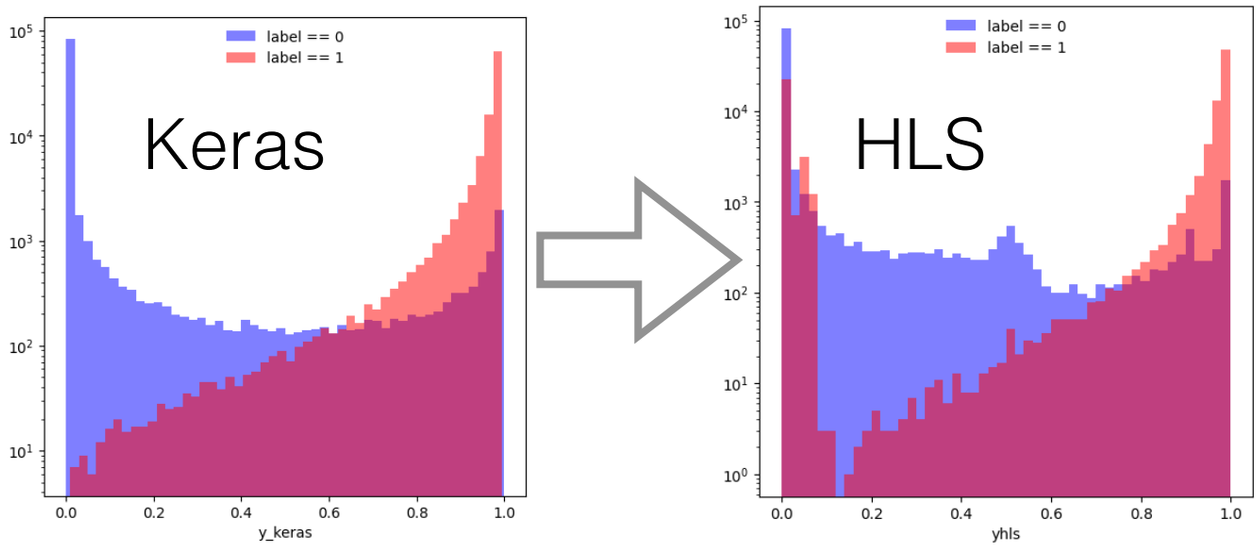


Project-1: Custom digital design

- Cosmic Ray Counting, TDC, OSERDES

Hands-on (AI/ML Design)

- AI/ML basics and model design in Python (**Keras/Tensorflow**)
- **Quantization** and **Pruning** for resource optimization
- HLS model conversion with hls4ml
- Optimize hardware with directives (**pragma** statements) in HLS tool and generate IP core




```
================================================================
== Utilization Estimates
================================================================
* Summary:
+-----------------+---------+-----+--------+--------+-----+
|       Name      | BRAM_18K| DSP |   FF   |   LUT  | URAM|
+-----------------+---------+-----+--------+--------+-----+
|DSP              |        -|    -|       -|       -|    -|
|Expression       |        -|    -|       0|      14|    -|
|FIFO             |        -|    -|       -|       -|    -|
|Instance         |        1|  830|   90006|   35787|    -|
|Memory           |        -|    -|       -|       -|    -|
|Multiplexer      |        -|    -|       -|      36|    -|
|Register         |        -|    -|    2565|       -|    -|
+-----------------+---------+-----+--------+--------+-----+
|Total            |        1|  830|   92571|   35837|    0|
+-----------------+---------+-----+--------+--------+-----+
|Available        |      730|  740|  269200|  134600|    0|
+-----------------+---------+-----+--------+--------+-----+
|Utilization (%)  |       ~0|  112|      34|      26|    0|
+-----------------+---------+-----+--------+--------+-----+
```

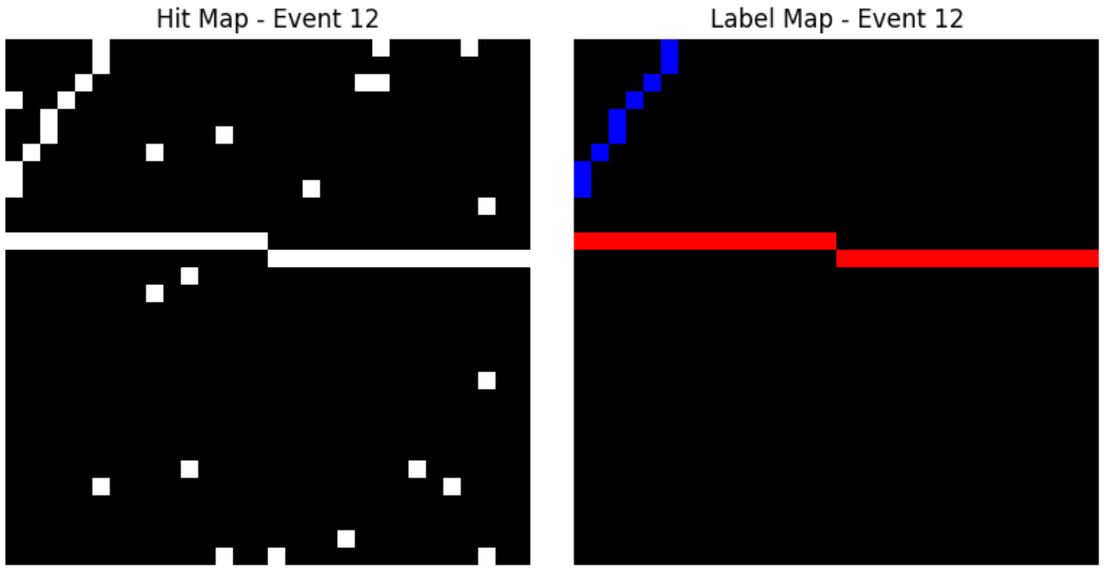
Project-2: NN classifier design/optimization for Belle II PID

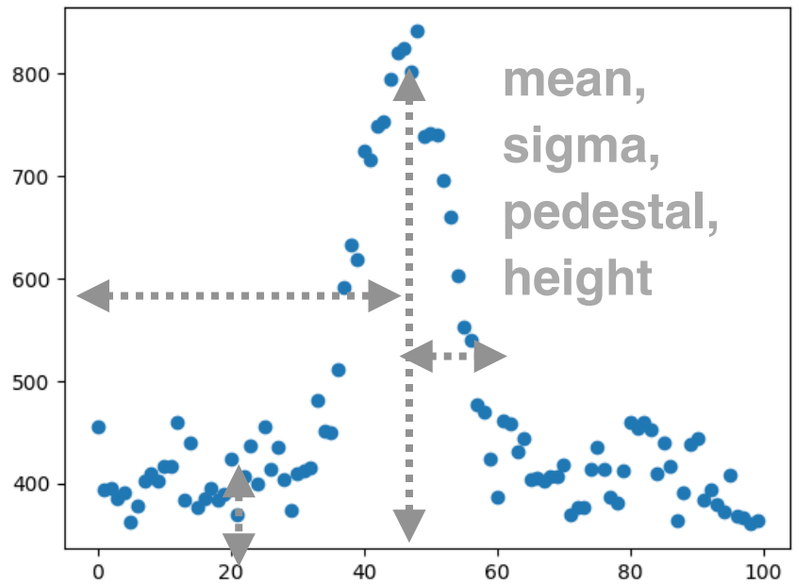
Final Project

Students select one of two projects:

(1) 2D Track Reconstruction (CNN)

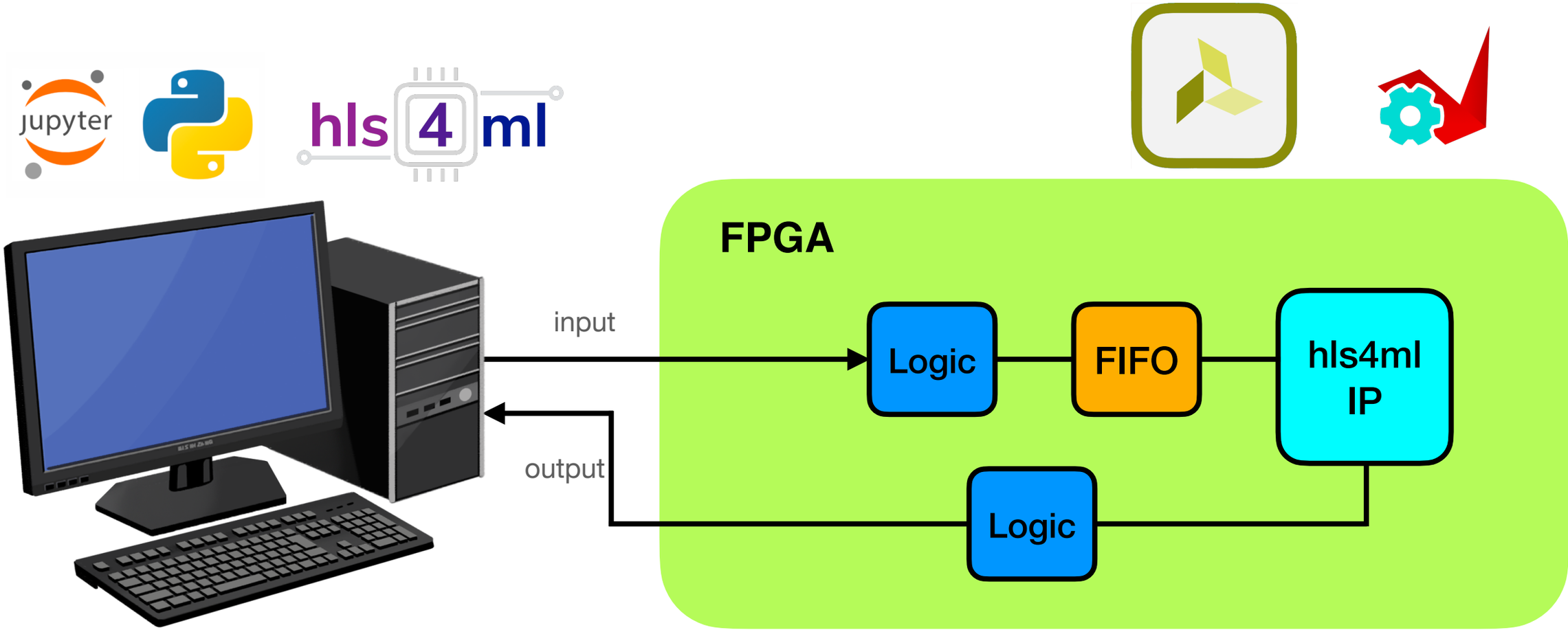
(2) Waveform Analysis (DNN)





- Model design and training
- Resource estimation and IP generation
- Full system integration on FPGA with real I/O.
- **Integrated Logic Analyzer (ILA)** is used for FPGA-level verification.

Design Overview



- The AI/ML alogorithm is developed on your PC.
- FPGA receives inputs and returns outputs for validation
- Output can be compared with the PC-generated result to verify correctness

Challenges & Opportunities

Current Challenges in PHYS476

- Balancing AI/ML theory and hardware implementation in 1 semester
- Students come from diverse backgrounds (undergrad, grad, postdoc)
- Tools are evolving rapidly (Vivado, Vitis, hls4ml, AI Engine, …)

Broader Trends in AI/ML Education

- Rising demand for AI/ML in physics and enginnering
- **Interdisciplinary teaching across departments**
- Hardware-aware AI/ML education beyond CS programs

Outlook

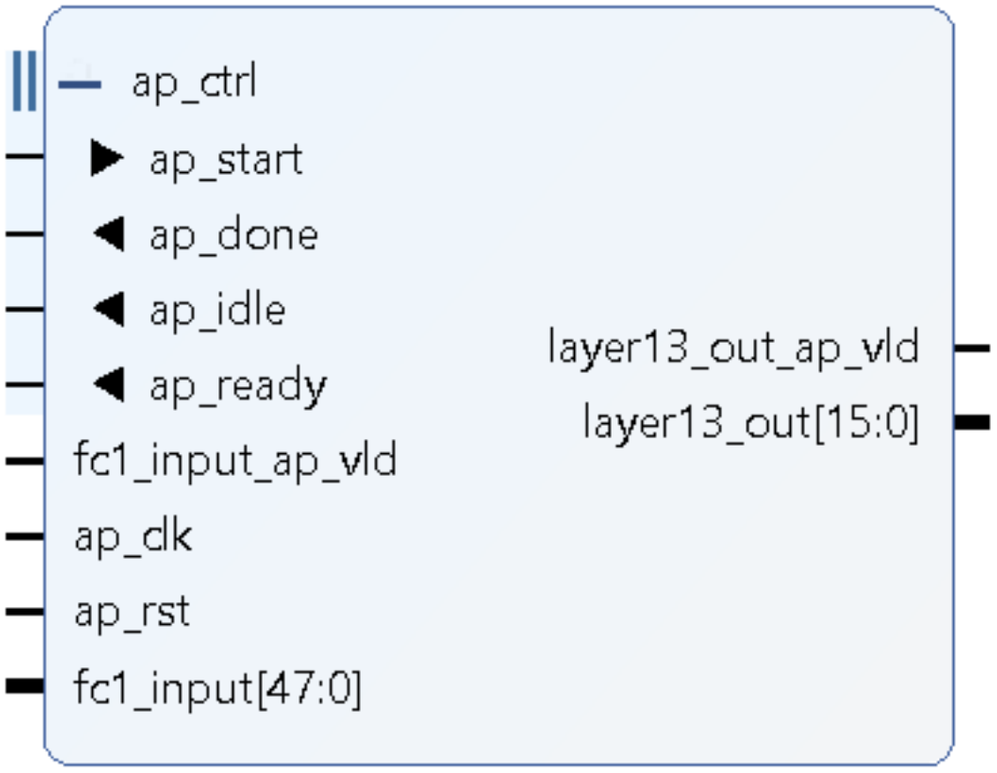
- PHYS476 = **a step toward unified scientific AI/ML education**
- Next steps:
  - Expand curriculum (summer school?)
  - Share materials (open syllabus/code)
  - Collaborate with broader AI/ML education efforts

Acknowledgements:

Thanks to all students of PHYS476, the UH Department of Physics & Astronomy, and DOE-HEP for their support.

Backup

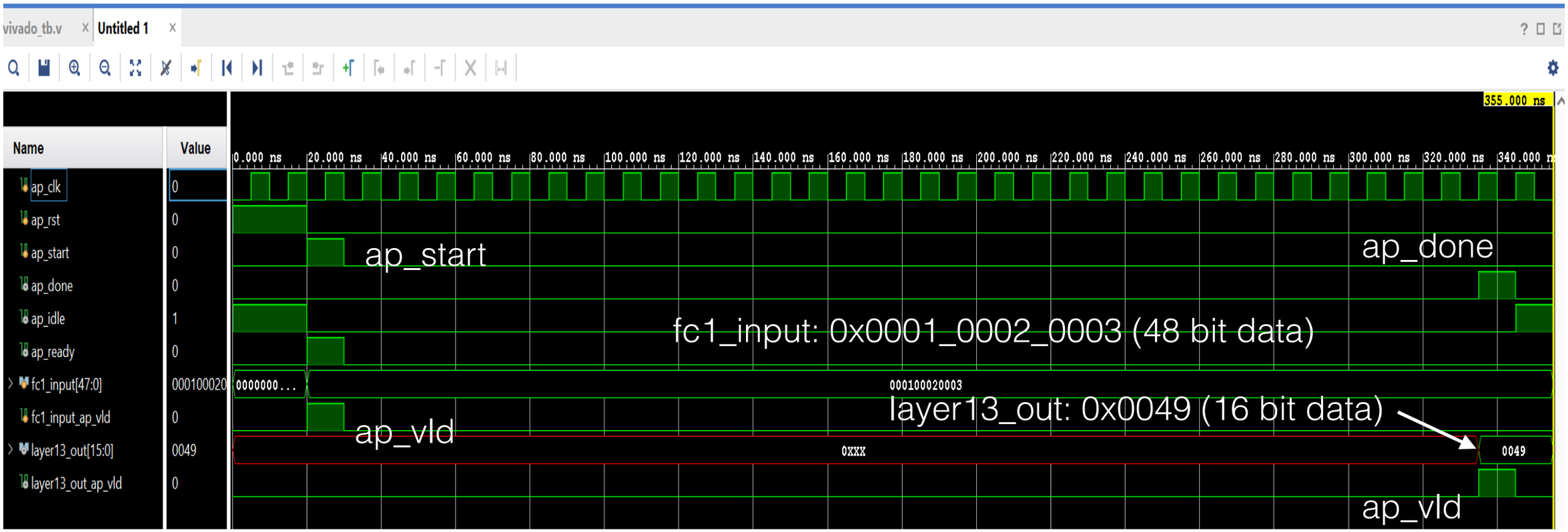
hls4ml IP



- Once registered, hls4ml IP (MyProject) appears in IP Catalog

- **ap_ctrl**: Control signals
- **fc1_input_ap_vld**: Signal indicating the input is valid.
- **fc1_input[47:0]**: 48-bit fixed-length input vector.
- **layer13_out_ap_vld**: Signal indicating the output is valid
- **layer13_out[15:0]**: Output result

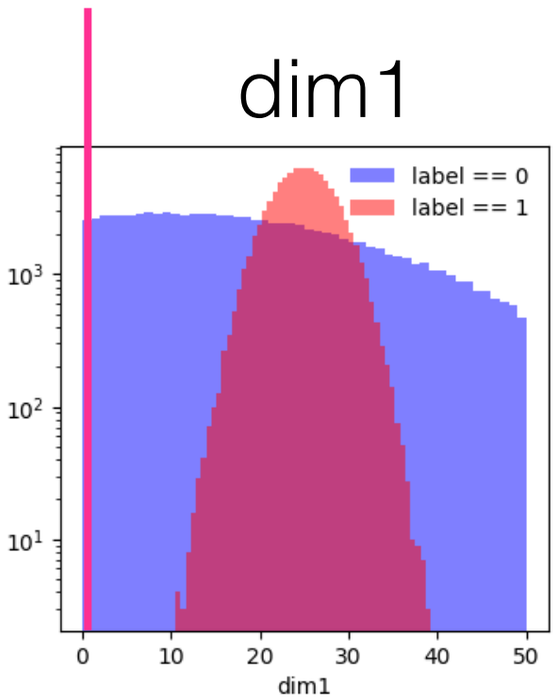
Vivado Simulation

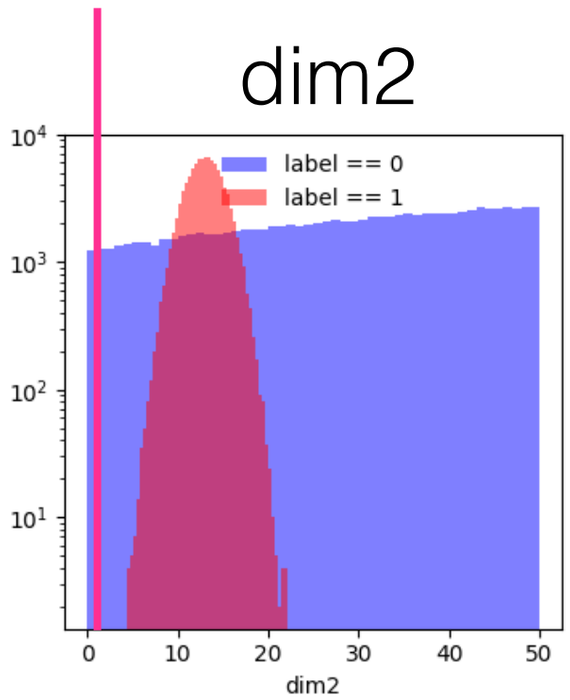


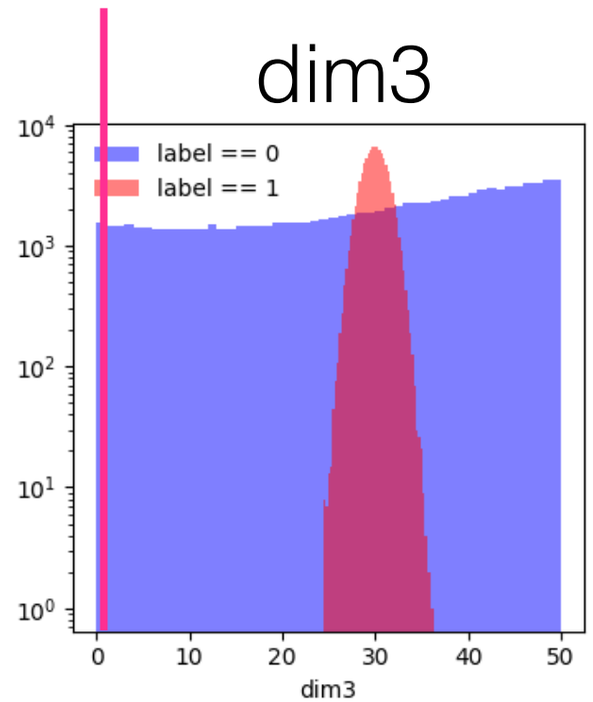
- 48 bit ***fc1_input*** is given as 0x0001_0002_0003, corresponding to three 16-bit fixed-point values for the input features (dim1, dim2, dim3).
- The 16-bit output ***layer13_out*** = 0x0049 (73 in decimal) is valid and marked by ***layer13_out_ap_vld*** = 1.
- NN output is interpreted as 73 x (1/1024) ~ **0.0712**

*This fixed-point format has a fractional bit width of 10, which gives a resolution of 1/1024.

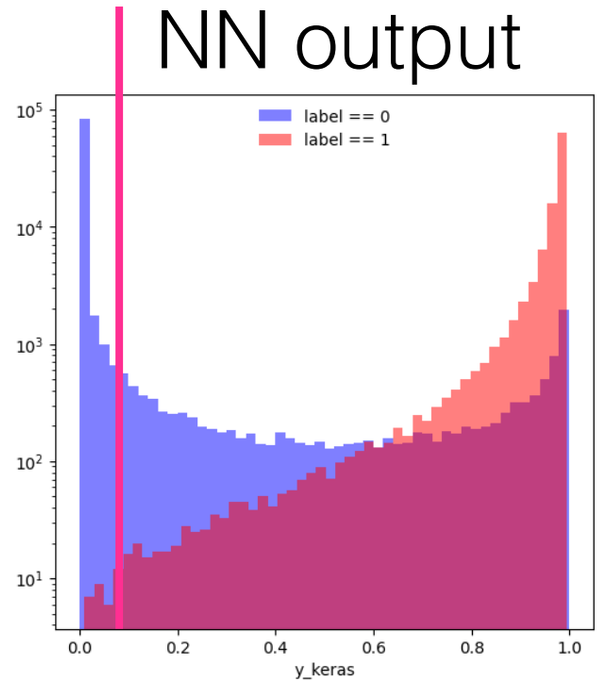
Input/Output Data







- Input data (dim1, 2, 3) used for the NN training ranges from 0 to 50.
- Output data ranges 0 to 1 as a probability (sigmoid).



TOP Module

```
1   `timescale 1ns/1ps
2
3   module top (
4       input  wire        clk,
5       input  wire        rst,
6       input  wire        start,
7       input  wire [47:0] fc1_input,
8       input  wire        fc1_input_vld,
9
10      output wire [15:0] layer13_out,
11      output wire        layer13_out_vld,
12      output wire        done
13  );
14
15    // internal signals
16    wire ap_ready;
17    wire ap_idle;
18
19    myproject_0 hls_inst (
20      .ap_clk(clk),
21      .ap_rst(rst),
22      .ap_start(start),
23      .ap_done(done),
24      .ap_idle(ap_idle),
25      .ap_ready(ap_ready),
26
27      .fc1_input(fc1_input),
28      .fc1_input_ap_vld(fc1_input_vld),
29
30      .layer13_out(layer13_out),
31      .layer13_out_ap_vld(layer13_out_vld)
32    );
33
34  endmodule
```

- The next step is to prepare a top module (**top.v**) for synthesis.
- After synthesis, proceed with implementation and bitstream generation.
- You will also need to define the overall FPGA design, including I/O and how to supply input data to the IP.