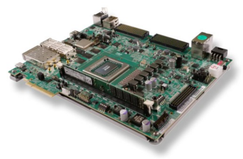
Deploying Inference Models at the Edge

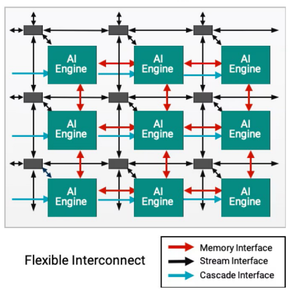
Sapkas Michail, TECH FPA Retreat, Feb 16 – 19, 2026, Catania, LNS

University of Padova – INFN Padova

The AMD VCK190 Versal AI Engine (AIE)

- A 50 by 8 **Tiled Array** of **400 Vectorized Processors**
- Distinct FP32 MAC pipelines
- Each vectorized unit has each own local memory accessed by the Processing System
- Multiple Data Paths inside the Ai Engine:
  - Cascade Streams
  - Shared Memory Banks
  - Stream Interconnect
  - Packet Streams
- 39 PL (128b) to AIE (32b @ 1.25Ghz) Interface Tiles
- Graph - Kernel Programming model

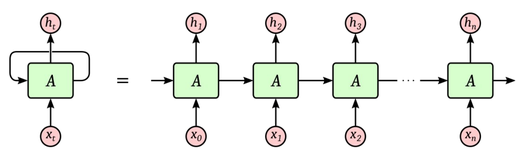




The Gated Recurrent Unit (GRU)

- Belongs to the family of Recurrent Neural Networks, specialized models for timeseries data
- Makes using of the “Gating” Mechanism and the Hidden State Recurrency which imposes Data Flow in the AIE

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
\hat{h}_t &= \phi(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t
\end{aligned}
$$



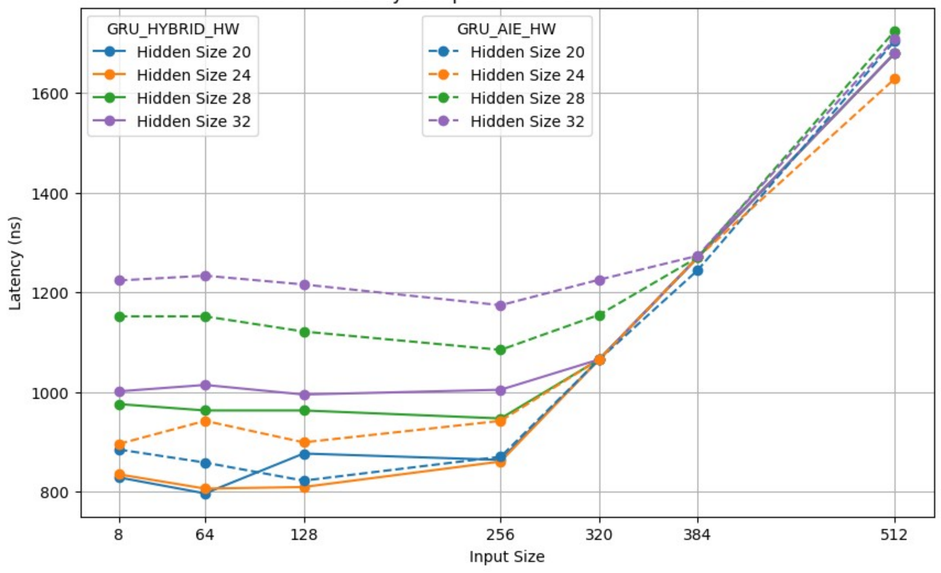
Objective

- To be used as **Real Time Controllers** trained continuously in a **Reinforcement Learning** framework
- For the first time, AIE gives us the possibility to deploy **Large** and **Unquantized (FP32) RNN** models with **microsecond latency**

Results

- Results were presented at **TWEPP 2025** and **published** in proceedings

GRU HW Latency vs Input Size for Different Hidden Sizes

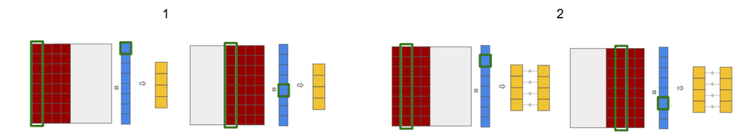


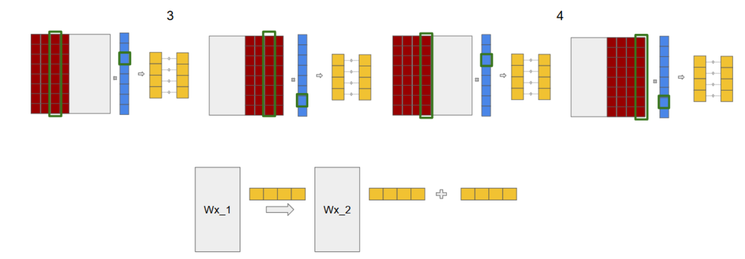
Implementing the Matrix Vector Multiplication

- It's not trivial which Data Path to use inside the AIE nor which Matrix-Vector method to use. The one affects the other.
- Inside the Vectorized Processor we are using the **Multiply Accumulate (MAC)** operation, which has a latency of 8 clock cycles for FP32

Method 1: Column-Element-Cascade

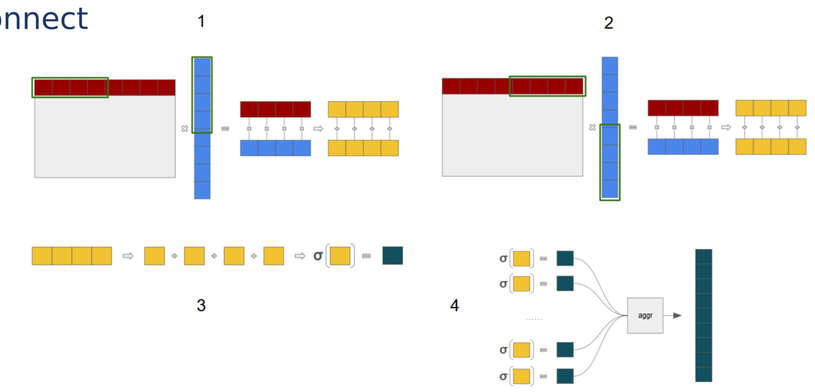
- This method respects the Interface tiles throughput by using the vector elements partially: We MAC elements of the vector with columns of the matrix





Method 2: Row-Vector-Reduce-Stream

- This method breaks pipelining to take advantage as many of the cores as possible exploiting the ability of the AIE to broadcast the same signal via the AXI4 Interconnect



Data Aggregation and Activation Functions

- We explore two possibilities for data aggregation **using only Method 2:**
- **Packet Split aggregation** and **sorting** inside the AIE using specialized kernels
- Or **aggregating**, **sorting** and **applying activation function LUTs in the PL,** using a **32-input HLS kernel connected to 32 interface tiles**.

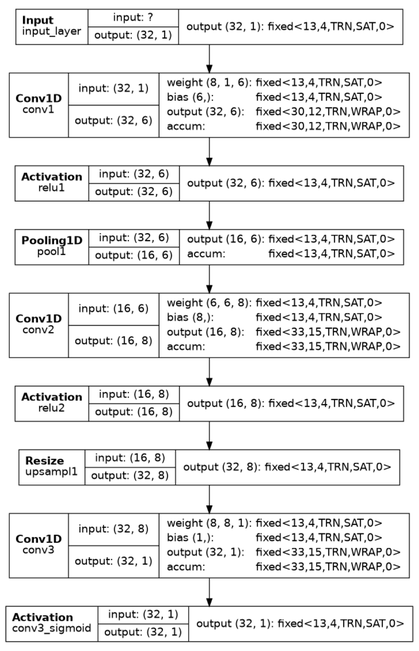
INTERFACE TILES HIDDEN DIM	AIE GRU TILES USED	AIE HYBRID TILES USED	PL kernel II	PL kernel DEPTH	PL kernel FF	PL kernel LUTS	AIE AGGR TILE LAT
20	184	181	6	7 cc = 22.4 ns	5124	9250	229.6 ns
24	220	217	6	7 cc = 22.4 ns	6066	11834	273.6 ns
28	256	253	8	9 cc = 28.8 ns	7057	15663	319.2 ns
32	292	289	8	9 cc = 28.8 ns	8003	19198	363.2 ns

1D CNN for Real Time Trigger using HLS4ML

Motivation

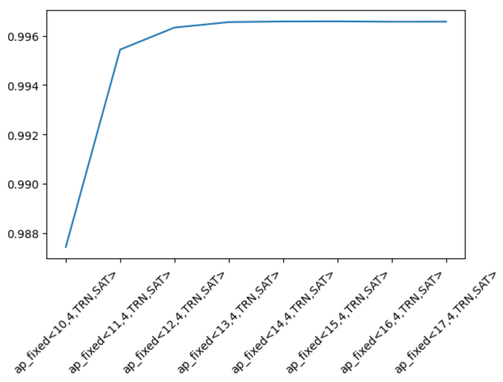
- Within the activity of a 6-month project for my PhD with CAEN SpA
- A Machine Learning system running in the FPGA used by the DT2740 Digitizer
- Zynq Ultrascale + zcu19eg
- The model outputs the probability that at each timestep there is a peak
- This is probably the only way to detect peaks with small SNR

Model



GridSearch Method

- To find the input / output dimension of the model, which affects directly the resources
- In a later stage, to find the best ap_fixed datatype that respect the AUC of the model and doesn’t overuse resources

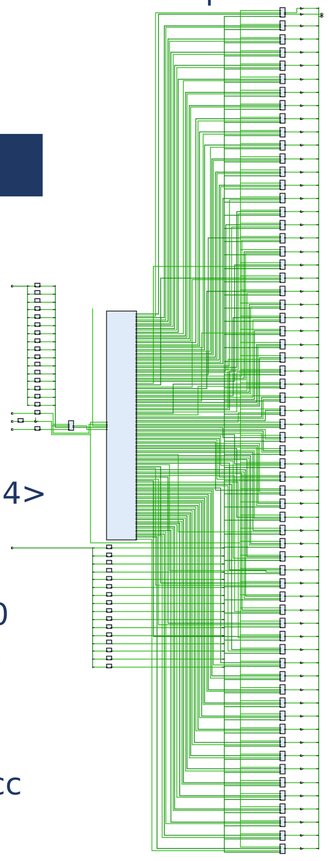


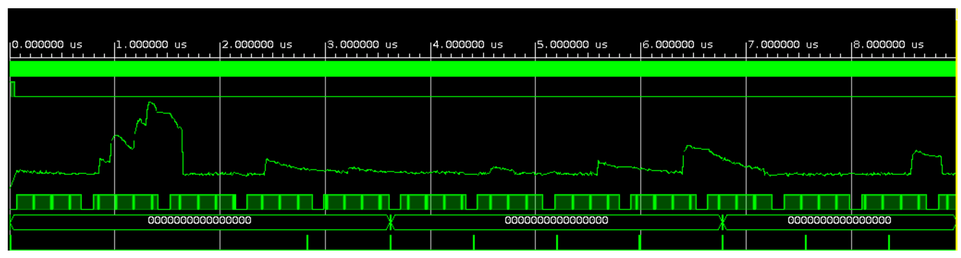
- ap_fixed<12,4> :

LUTs: 123920
FFs: 159183
DSPs: 311
BRAM: 16
Latency: 178cc
II: 64c

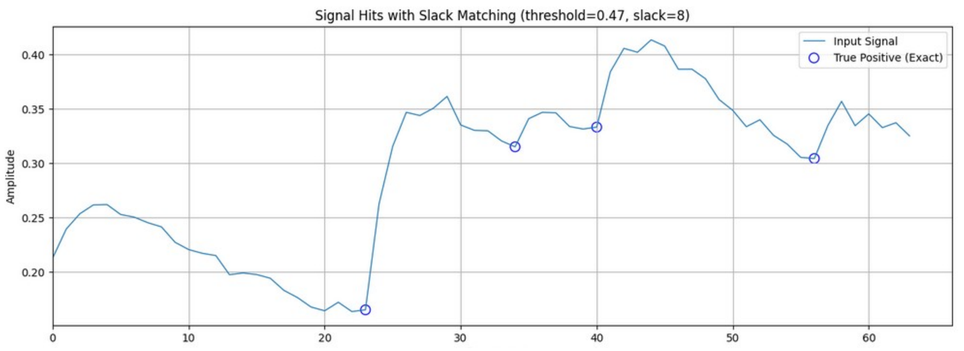
Vivado Implementation

- Implements a buffer for the input
- Comparators at the output that produce the final binary string
  - Tested in Vivado testbench





- Possibility to achieve higher Initiation Intervals to accommodate for on-line overlapping of time frames



- On-line detection of and identification of hits