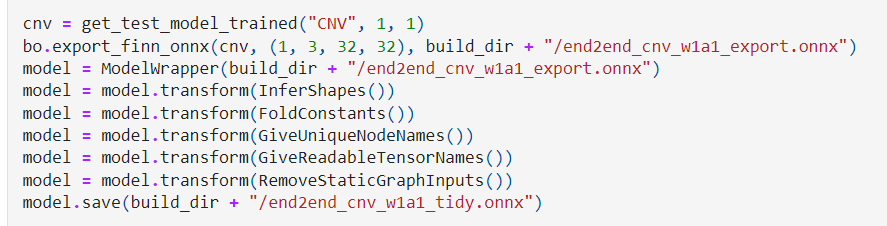
Lab D FINN report

110061638 呂政和 110061639 林致佑 110061641 鄧向凱

Part1:

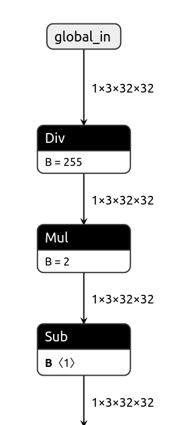
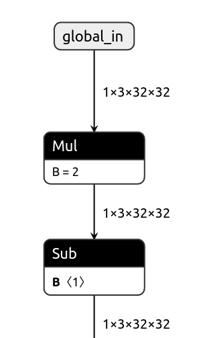
(1)cnv\_end2end\_example:



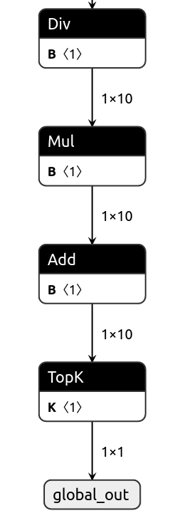
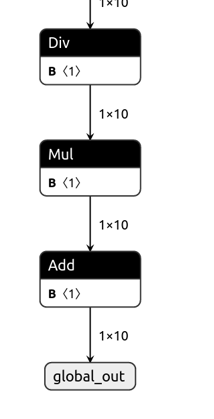
After we get the pretrained network from Brevitas, we have to import it into FINN and tidy-up through several transform functions.

There are some important functions before we get tidy.onnx. In the GiveUniqueNodeNames function, we can notice that inputs and outputs will be different names based on their usage in function. In the FoldConstants function, it will merge some arithmetic functions together to simplify the whole model. The graph below is the layer structure with Neutron.

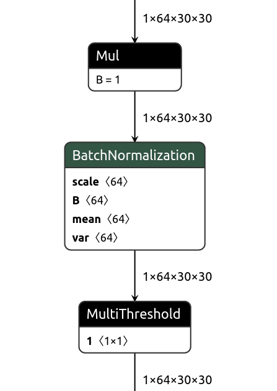
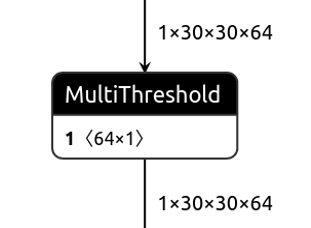
In the pre-processing and post-processing step, the preprocessing step divides the input uint8 data by 255 so the inputs to the CNV-w1a1 network are bounded between [0, 1]. The postprocessing step takes the output of the network and returns the index (0-9) of the image category with the highest probability (top-1). After this step, compared to the lower left graph, we may notice that a new “Div” block had been added in the lower right graph.

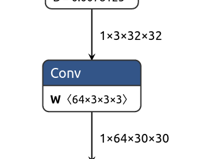
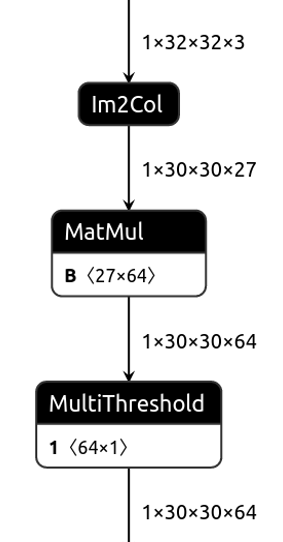


Before global\_out, the topk function is added to return the specified number of series at the top of the ranking. Therefore, we can only get a 1\*1 data from the output side.



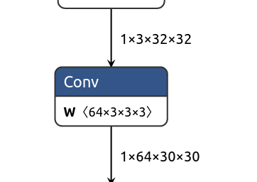
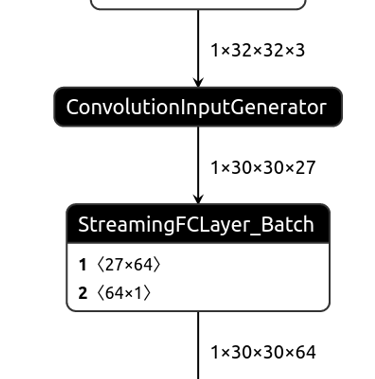
In the lowering and streamlining step, we apply a convolution lowering transformation which can convert convolutions to matrix-matrix multiply operations. As for the streamlining function, it moves floating point scaling and addition operations closer to the input of the nearest thresholding activation and absorbs them into thresholds. Therefore, the convolution node is replaced with Im2Col and MatMul nodes, which we can find in the lower right graph and it is different from the former graph in Pre- and Postprocessing step. In MakeMaxPoolNHWC and AbsorbTransposeIntoMultiThreshold function, they convert the data layout of the network into the NHWC data layout, so the BatchNorm, Mul, Add nodes can be replaced with MultiThreshold node.

------------->

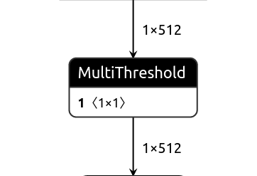
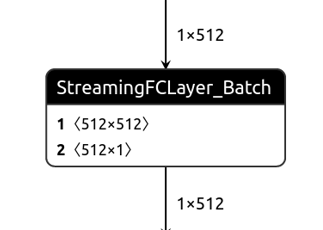
---------->

In Partitioning, Conversion to HLS Layers and Folding step, we delete the Reshape node. Furthermore, ConvolutionInputGenerator, StreamingFCLayer\_Batch and StreamingMaxPool\_Batch nodes implement the sliding window, matrix multiply and maxpool operations in hlslib. StreamingFCLayer can be a convolution or fully connected layer because they can all be done by matrix multiplication.

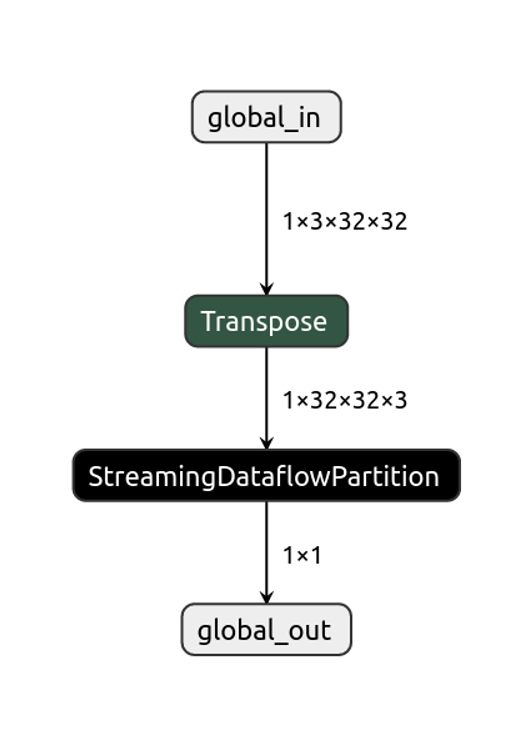
Take first convolution layer as example:

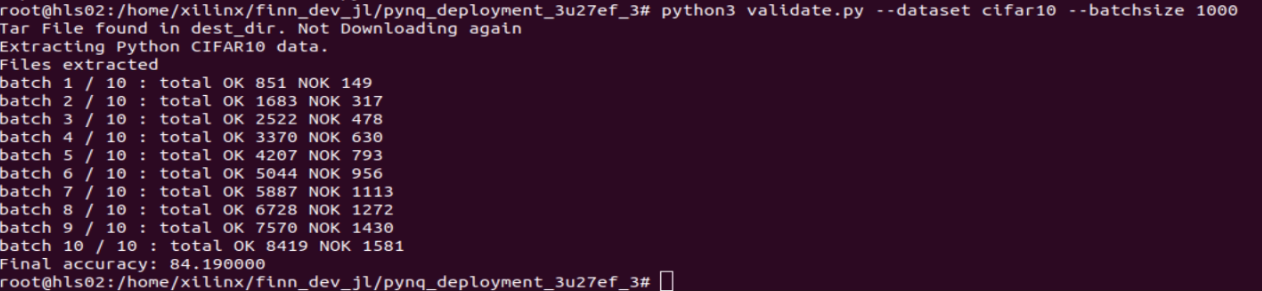
----->

Take last fully connected layer as example:

----->

Below is the end2end\_cnv\_w1a1\_dataflow\_parent.onnx. We only need to transpose the data(NHWC -> NCHW), and then we can obtain the output from StreamingDataflowPartition node.

  
**Model Accuracy:**

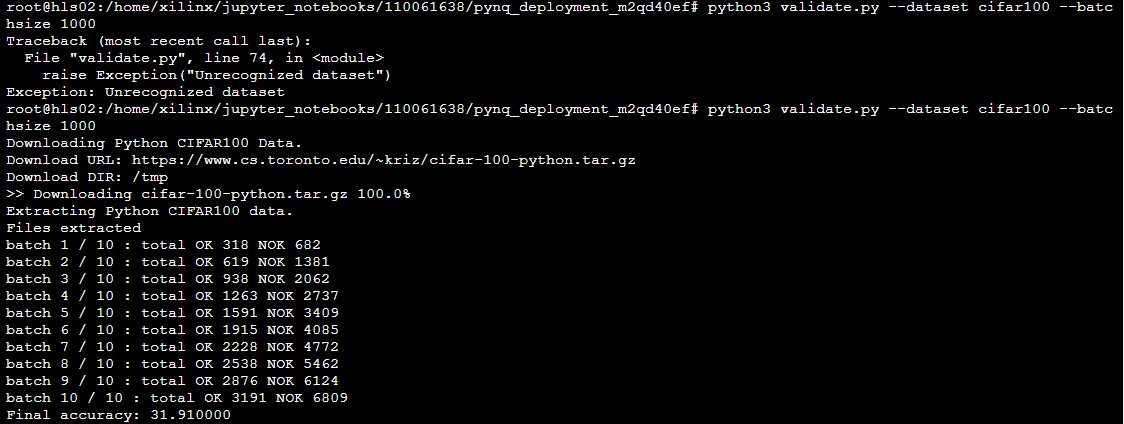


The final accuracy is 84.19%.

Part2:  
VGG9 model accuracy on server:



VGG9 model accuracy on FPGA:

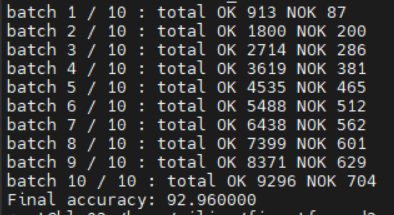


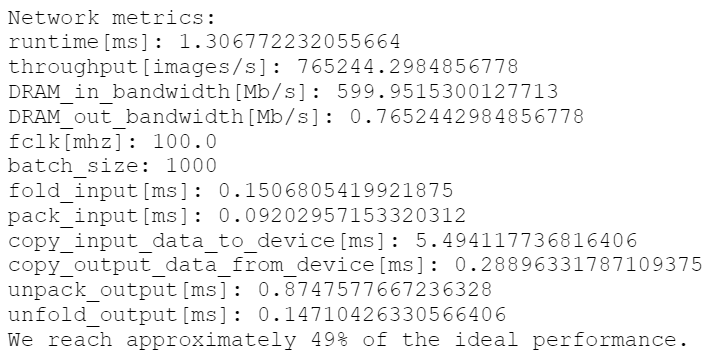
因為在FPGA上跑的準確率會降低很多，這是因為VGG9跑在Cifar10的準確率本來就不太好，會選擇用VGG9主要是因為他剛好可以被塞入Pynq中。

Part3:

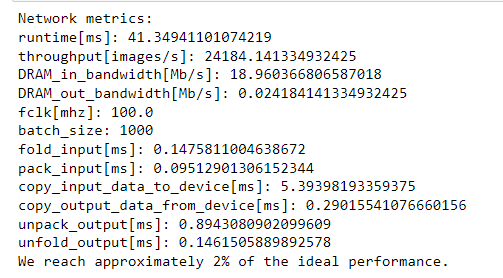
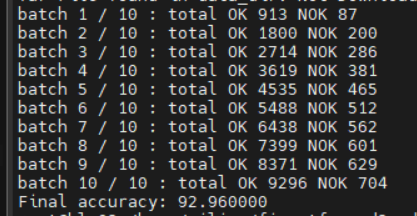
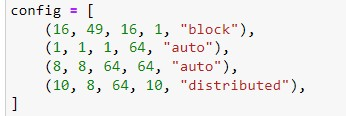
1.Screen Dump:

(1)origin given model:



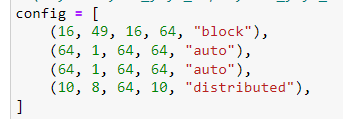


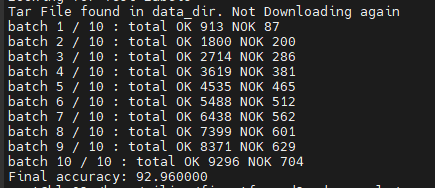
(2)PE = 1, SIMD = 1:

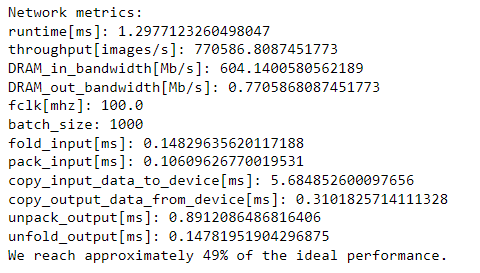


(3)Try your best to achieve the best performance by adjusting the PE and SIMD

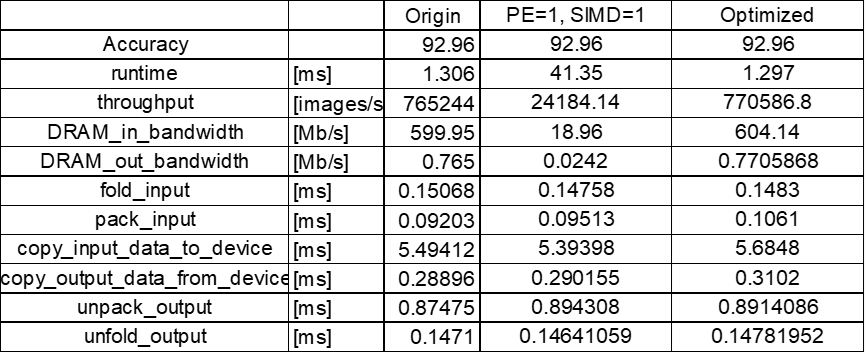
Questions:







2.List all your results in a table:

****

3.• Write down your observations and explain possible reasons for this result (Exp1):

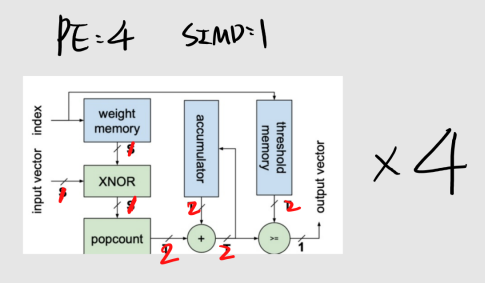
In exp1, we set both layer2’s PE and SIMD which causes the longer runtime and lower throughput and bandwidth. The reason might be the unbalanced computation time of each layer, and layer2, which is the so-called critical path, takes the longest computation time. PE = 1 means there is only one PE in layer2. SIMD = 1 says that each PE only take one byte data every cycle.

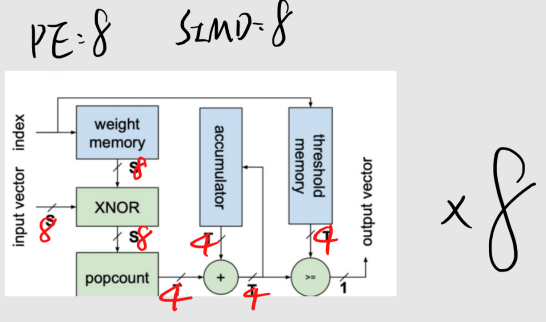
4.

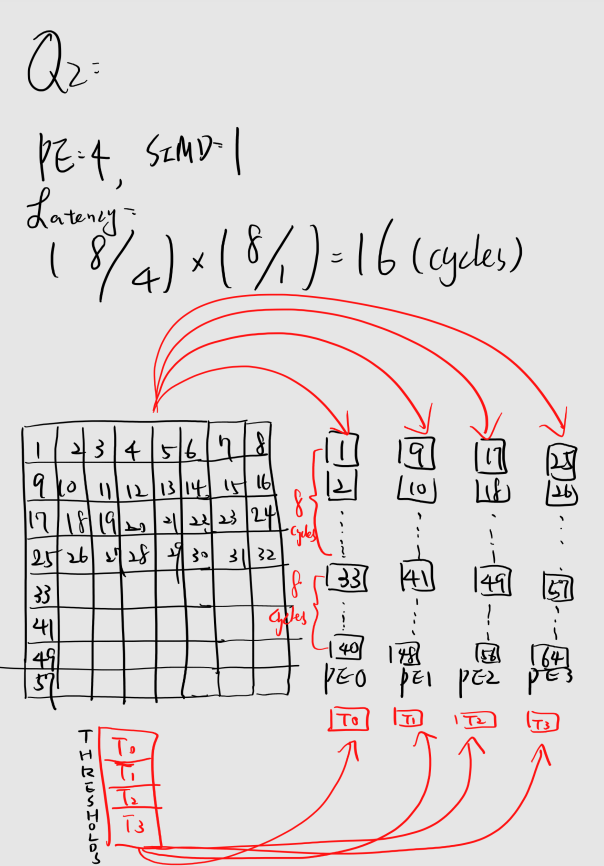
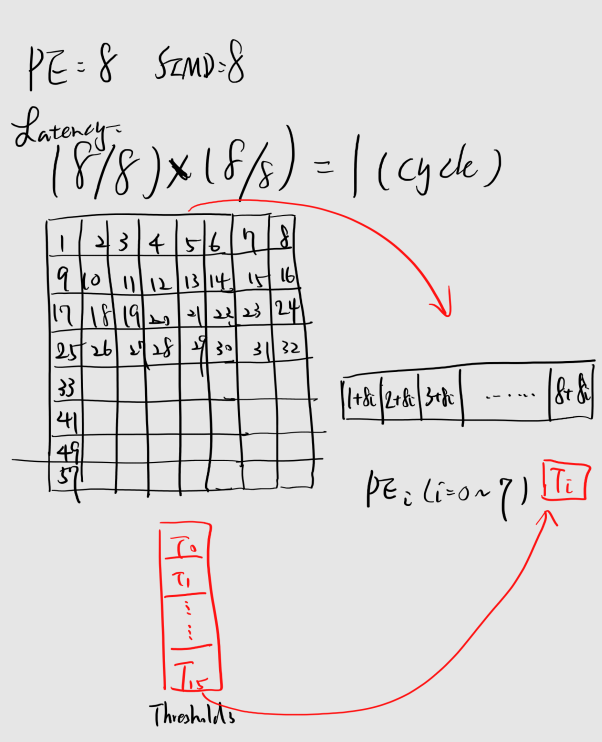
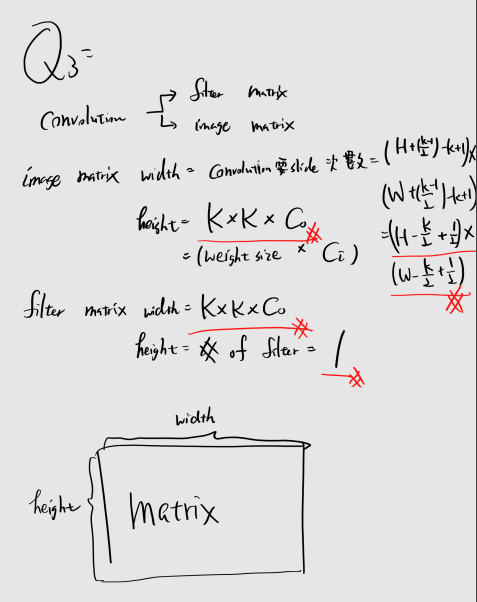
Explain your methodology for adjusting the folding factors and analyze the results. (Exp2):

We try to tune the PE and SIMD in exp2, and the optimized model shows a little improvement compared to the origin model. The main concept we follow is that more PEs can compute more data at the same time, which could reduce each layer’s latency. As a result, despite the fact that there might be unbalanced computation time between layers, the runtime is still reduced and the throughput, bandwidth are improved, too.

1. Please draw the circuit diagrams for the following two PE and SIMD configurations.

• PE: 4 SIMD: 1  


• PE: 8 SIMD: 8  


1. Following the previous question, suppose we need to calculate a 8 x 8 matrix multiplication; please calculate the latency for the two configurations.  
     
   
2. For a 𝐻 × 𝑊 × 𝐶𝑖 feature map, and given 𝐶0 stride 1filters of size 𝐾 × 𝐾 , what is the shape of the filter matrix and image matrix? (Assuming the input images are padded for simplicity, i.e. the output resolution is still H x W.)  
   
3. In this section, the image feature map in the figure is of size 2 x 3. However the real application of the image might be of size 224 x 224. In this case, can we deal with the whole image? Are there any practical solutions in the perspective of hardware design?  
   