Project Report - ECE 284

Achuth Krishna
Wuqiong Zhao
Yijie He
Matthew Jarecky
Haochen Jiang

Abstract

This project centers around the task of designing a 2D systolic array architecture for machine learning convolution layer acceleration. In particular, we quantized the VGG16 model trained on the CIFAR dataset and used the 27th layer to validate our hardware design.

1 Part 1 Vanilla

For the software design, the VGG16 model's convolution layers were first quantized to 4-bit weights and 4-bit activations. The 27th layer (a quantized convolution) was reduced to only 8 channels from the original 512, along with a corresponding change to the output channel size of the preceding convolution and the input channel size of the proceeding convolution. Further, the batch normalization layer was removed to simplify construction of the SFP, resulting in a simple convolution$\rightarrow$ ReLU sequence. After training to achieve an accuracy of at least 90% under this scheme, we wrote the activations, weights, partial sums, and ideal output to text files for processing by our Verilog testbench.

The hardware design involved creating a 2-dimensional systolic array with a weight-stationary mapping scheme. Since the convolution kernel for this layer was of size 3x3, we used our testbench to read the nine weight files one at a time, using each for processing of the values in the activations text file. Upon passing the functional verification, we proceeded to obtain the synthesis values for resource utilization, frequency, power, and more.

	Version (Vanilla)
Total Operations per Cycle	128
Frequency	113.2 MHz
Resource Utilization	16,997 Logic Elements (11%)
Dynamic Power(mW)	224.05 mW
TOPs (Trillion Operations per second)	0.0145
TOPS/W	0.0267

Figure 1: The synthesis results for the 2D systolic array

2 Part 2 SIMD

The second phase of the project saw us modifying both the software and hardware designs from the previous part. We first took the same convolution layer from VGG16 with the same 4-bit weight quantization, but instead modified the activations to be two bits instead of four. Training to achieve a higher accuracy was significantly harder under this lowered bit precision, but we nevertheless achieved an accuracy of 89.5% on the test set.

Upon modifying our VGG16 model, we proceeded to modify our hardware to enable a 2-bit and 4-bit reconfigurable SIMD systolic array design. Building off of the same weight-stationary scheme from the vanilla design, we made the requisite changes to both the array tiles and the array itself, along with our testbench. We subsequently passed the functional verification and ensured backward compatibility with our previous 4-bit-only design.

3 Part 3 Reconfigurable

In parallel with our Part 2 design, we implemented a reconfigurable weight-stationary/output-stationary functionality for our systolic array design. The same VGG16 model from Part 1 was used, with 4-bit weights and activations. Our changes to the testbench and core tile design were more substantial, having to change how we loaded the requisite values from the activation/weight text files, as well as the mapping of each processing element in the array. We subsequently achieved functional correctness.

4 Alpha 1 Reconfigurable SFP

Our first addition to the basic design was a reconfigurable SFP module that used look-up tables to compute different activation functions besides just ReLU. Since the activations and weights were quantized, we simply mapped the possible values (with some minor bit precision error) to their corresponding function values under the different schemes we wished to support: namely, ReLU, Softmax, and Tanh.

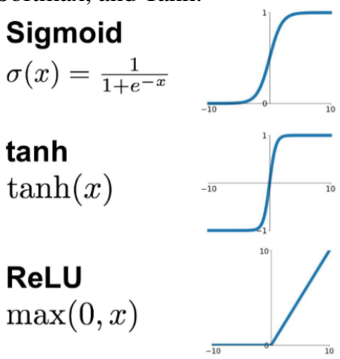


Figure 2: The activation functions we support in our reconfigurable SFP

5 Alpha 2 Physical FPGA Board Mapping

We successfully mapped our weight-stationary 2D systolic array design to a physical Pynq board FPGA, shown below, validated with our VGG16 values for 4-bit weights and activations. The weights were stored in the board's BRAM, and the entire set-up was functionally verified.

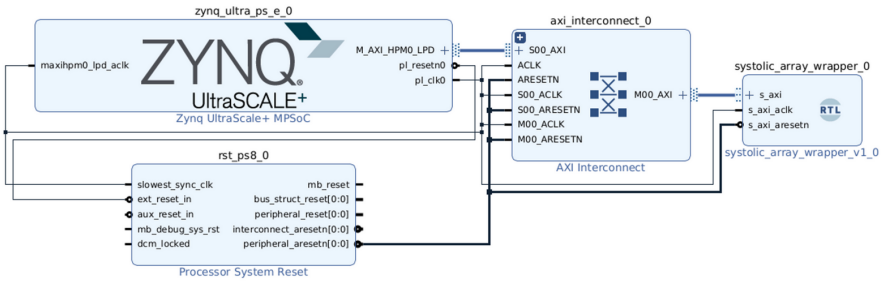
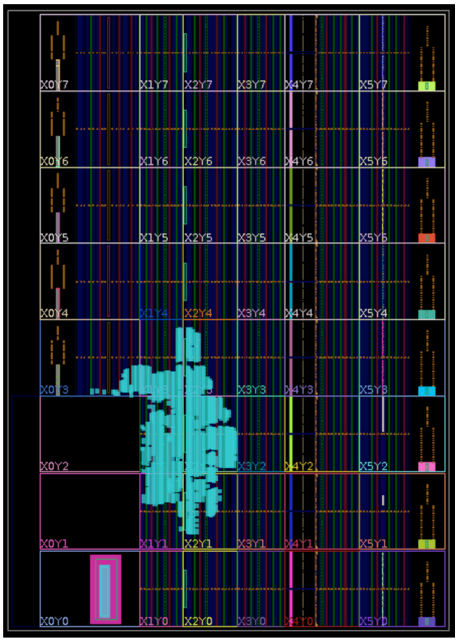


Figure 3: The block diagram for the physical board design



Detailed Power Consumption Breakdown

Type	Component	Power (W)	Percentage
Dynamic	**Total**	**0.724 W**	**60%**
	PS	0.670 W	92%
	Clocks	0.027 W	4%
	Signals	0.010 W	1%
	Logic	0.009 W	1%
	URAM	0.005 W	1%
	BRAM	0.003 W	1%
Static	**Total**	**0.479 W**	**40%**
	PL	0.479 W	100%

Resource Utilization

Resource	Utilization	Available	Utilization %
LUT	8837	425280	2.08
LUTRAM	66	213600	0.03
FF	16525	850560	1.94
BRAM	4	1080	0.37
URAM	2	80	2.50
BUFG	2	696	0.29

Figures 4,5,6: The area, power, and resource usage for the physical board under the weight-stationary mapping

6 Alpha 3 VGG16 Layer Quantization Study

Elaborating on our 2-bit activation quantization from Part 2, we wished to analyze how different convolution layers of the VGG16 model respond to their activations being quantized to two bits instead of four. To that end, we quantized each layer's inputs to two bits and compared the accuracy drop of each layer to the base 4-bit activation model. Then, we used a threshold ϵ to choose only the least sensitive layers for 2-bit activation quantization. From the base model, we were able to retain an accuracy above 90% while further quantizing the activations of five convolution layers.

```
=> Start greedy search from all-4bit baseline: acc=90.68%, epsilon=1.00%

  ACCEPT layer  9 -> 2bit | acc=90.74%, drop=-0.06%
  ACCEPT layer  8 -> 2bit | acc=90.62%, drop=0.06%
  ACCEPT layer 11 -> 2bit | acc=90.30%, drop=0.38%
  ACCEPT layer 12 -> 2bit | acc=90.22%, drop=0.46%
  ACCEPT layer 10 -> 2bit | acc=90.35%, drop=0.33%
  REJECT layer  6 -> 2bit | acc=89.32%, drop=1.36%  (> 1.00%)
  REJECT layer  7 -> 2bit | acc=89.41%, drop=1.27%  (> 1.00%)
  REJECT layer  3 -> 2bit | acc=89.65%, drop=1.03%  (> 1.00%)
  REJECT layer  2 -> 2bit | acc=89.21%, drop=1.47%  (> 1.00%)
  REJECT layer  4 -> 2bit | acc=89.00%, drop=1.68%  (> 1.00%)
  REJECT layer  1 -> 2bit | acc=89.06%, drop=1.62%  (> 1.00%)
  REJECT layer  5 -> 2bit | acc=88.88%, drop=1.80%  (> 1.00%)
  REJECT layer  0 -> 2bit | acc=80.42%, drop=10.26%  (> 1.00%)

==================== Final Mixed-Precision Config ====================
Per-layer activation bits (per QuantConv2d):
[4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2]

Total conv layers = 13
2bit layers       = 5 (38.5%)

Final mixed-precision accuracy     : 90.35%
Accuracy drop vs all-4bit baseline : 0.33%
```

Figure 7: The VGG16 greedy choice of quantizing layer activations further

7 Alpha 4 OpenROAD Synthesis and PnR

Our final addition was exploring a place-and-route flow of the entire design (including memory elements) using the open-source tool OpenROAD. The majority of the difficulty stemmed from the tool not recognizing the SRAM blocks as synthesizable memory elements, prompting us to change our memory block implementation. Below is the resulting resource allocation and area with a 130nm process technology.

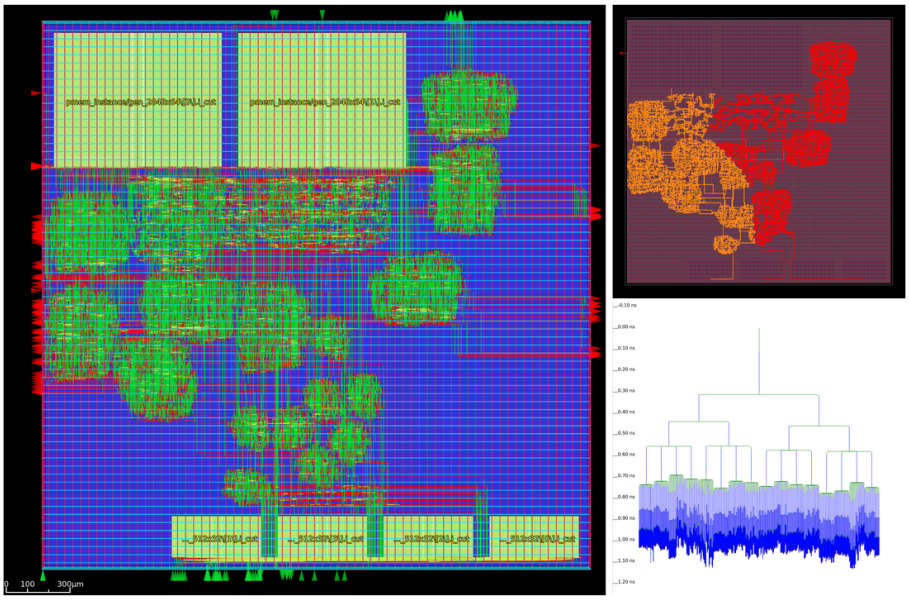


Figure 8: The results of the OpenROAD Synthesis and PnR flow