Fully Pipelined Reconfigurable 2D Systolic Array

k-furthest-neighbors

Anushka Chaudhary, Arjit Verma, Chainika Shah, Hariram Ramakrishnan, Shail Vaidya

Motivation

To develop a reconfigurable AI accelerator utilizing 2D systolic array architecture, enhanced with advanced techniques such as sparsity-aware clock gating and structured pruning, that is optimized and validated using a VGGNet model trained with quantization-aware training with the CIFAR-10 dataset.

VGG 16 Training

Accuracy	91%
Quantization Error	0.025

Hardware Implementation

Cyclone IV FPGA Mapping of Vanilla Version	
Fmax	122.2 MHz
Switching Activity	20%
PVT	ss-1.2V-100C
Thermal Dynamic Power	18.76mW
Total Logic Elements	6886
Total Registers	4132
Total Operations	8x8x2 = 128
GOPs/s	16
TOPs/W	0.834

Instruction Mapping			
000	Idle	100	Soft_Reset
001	WS_Load	101	OS_Shift
010	WS_Exec	110	OS_Exec
011	Not Used	111	Not Used

Compute Time Optimization

Weight Stationary Mode: 586 cycles

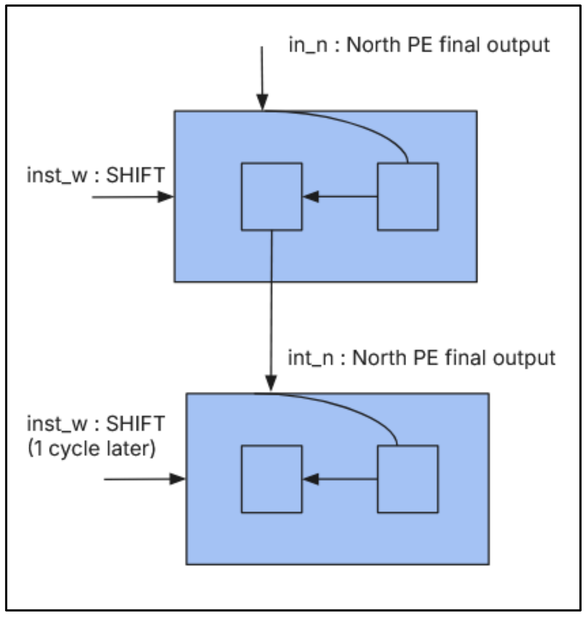
- Instead of resetting the entire MAC array to load the next set of weights, we use a soft reset that trickles across tiles.

Output Stationary Mode: 47 cycles

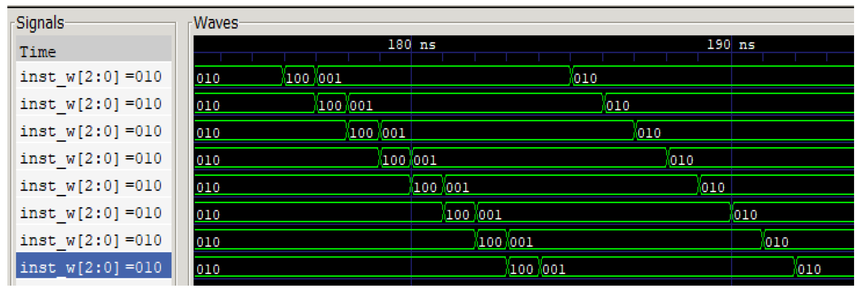
- None of the PEs are idle at any point of time. Right after execution completes, it starts shifting the pixel south.

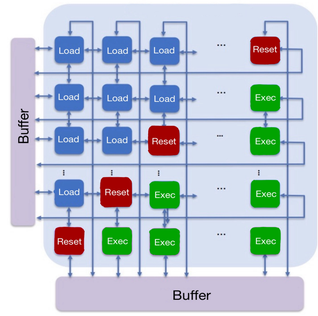
Pipelined output shifting

Implements a fully pipelined output shifting methodology, which allows the calculated output pixel values to begin shifting, as soon as the computation is completed in the PE. This is independent of the state of the neighboring PEs because it is flopped before heading south.



Synchronous Soft Reset



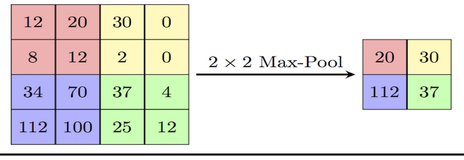


A soft reset instruction is given between every set of weight loads. This ensures that the PE is active in each cycle, allowing for completely pipelined operation.

Testbench Optimizations

The testbench functions as the control unit in our project, and we have designed it to sequence instructions with no idle cycles.

Max Pooling & Batch Norm



- The SFU is capable of implementing a **MaxPool2d** layer. When enabled, the SFU computes the result of the MaxPool operation along with ReLU, with any kernel size and stride. This is verified for the MaxPool2d(2,2) present in VGG16.
- **BatchNorm** layer is planned to be implemented in the systolic array through an enhanced PE capable of normalizing the output, in output stationary mode.

Pruning & Sparsity-Aware CG

- Structured pruning employed to increase the sparsity in the network, thereby reducing the computation load and memory requirements.
- Dynamically disable the computation if the incoming weight or activation is zero, which reduces unnecessary toggling and dynamic power consumption.

References

1. Y. Zhijie, W. Lei, L. Li, L. Shiming, G. Shasha and W. Shuquan, "Bactran: A Hardware Batch Normalization Implementation for CNN Training Engine," in IEEE Embedded Systems Letters
2. T. Sledevic, “Adaptation of convolution and batch normalization layer for CNN implementation on FPGA,” in Proc. Open Conf. Elect. Electron. Inf. Sci. (eStream), 2019, pp. 1–4.