SIMD and Weight-Output reconfigurable 2D Systolic array-based accelerator

JOEVER

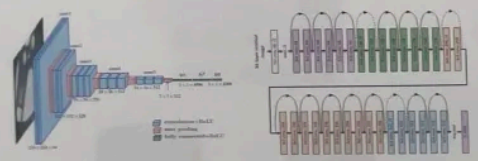
Jesse Vernallis, Stanley Pan, Santhalpa Hota, Rohon Ray
Nazin Bitar, Madeleine Mcswain

Motivation

- Modern AI workloads demand higher throughput and better energy efficiency than conventional CPUs and GPUs can provide. Systolic arrays offer massive parallelism and data-reuse benefits, but real-world models require flexible precision, different dataflow styles, and efficient mapping onto limited hardware resources. This project aims to explore how architectural reconfigurability—across bit-widths (2-bit/4-bit), dataflow modes (weight-stationary vs. output-stationary), and layer-specific channel reductions can improve performance, reduce power, and maximize utilization on FPGA-based accelerators.

Alpha 1. VGGNet vs ResNet (4bit) with quantization-aware training

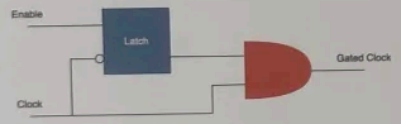
	VGGNet	ResNet
Accuracy (%)	93.03	90.46
Quantization Error	1.52E-05	9.54E-06



(a) VGGNet Architecture (b) ResNet Architecture

- VGG-style networks are easier to map onto a 2D systolic array because they use uniform, stackable 3x3 convolutions that keep the MAC grid busy with regular dataflow.
- ResNet, while often more parameter-efficient and accurate, introduces skip connections and more irregular shapes that can fragment the workload on the array and make it harder to keep all processing elements fully utilized.

Alpha 2. Clock Gating



- Clock gating applied to MAC compute array inside **mac_tile**
- In digital circuits, total power dissipation is divided into static (leakage) power and dynamic power.
  - Dynamic power is primarily influenced by clock signals driving synchronous circuits in an FPGA, where each clock signal transition causes switching activity, leading to increased dynamic power dissipation [1]
- Clock gating reduces this switching by selectively disabling the clock to registers when they do not need to update, thereby reducing toggling and lowering the activity factor **α**

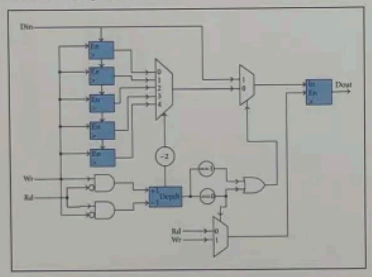
	Before clock gating	After clock gating
Total Power Dissipation (mW)	244.05	245.43
Core Dynamic Thermal Power Dissipation (mW)	34.93	35.37
Core Static Thermal Power Dissipation (mW)	118.76	118.77
I/O Thermal Power Dissipation (mW)	90.36	91.29

	Before clock gating	After clock gating
Max Clock Frequency (MHz)	135.94	137.17

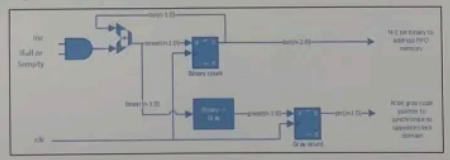
- While our measured power reduction on this specific testbench was small due to minimal OFIFO stalling, the mechanism provides clear architectural scalability and power efficiency benefits under more realistic workloads or deeper network execution
- In addition to lowering the overall power dissipation, our results showed **~1MHz** increase in our max clock frequency, improving the overall performance of our accelerator

Alpha 3. Optimizing FIFO Depth in Weight-Stationary PES

- For our weight-stationary mapping, Large FIFOs provide diminishing returns due to repetitive activations and weight localization



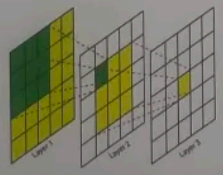
- We therefore reduced the input FIFO depth from 64 → 8 on L0 and cut another FIFO from 64 → 16, which lowers on-chip memory, area, and dynamic power, while still providing enough buffering to smooth minor bandwidth fluctuations.
- Conceptually, this aligns with prior DNN accelerators that show local buffers should be sized to match the chosen dataflow and reuse pattern rather than maximized blindly, especially in weight-stationary designs where weight reuse dominates and activation buffering can be modest.



Smaller FIFO

Alpha 5. LUT- Guided Nij. selection for Efficient 3x3 Convolution

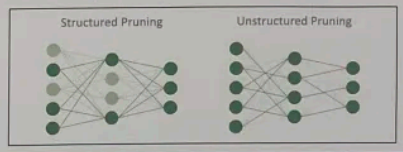
- By pre-selecting only the 16 nij values that actually contribute to each 3x3 kernel tap, we cut per-channel work from 324 to 144 MACs and reduce required L0 input from 36 to 16. On the systolic array, this reduces latency without changing the math of the convolution.



Reducing the size with a LUT effectively reduces value to convolve.

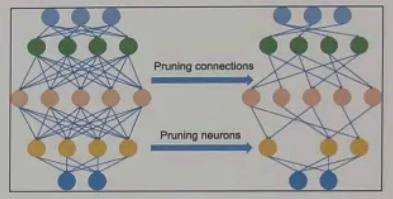
Alpha 4. Combined Structured and Unstructured Pruning

- Unstructured Pruning:
  - Great for compressing weights and potentially lowering energy if you have sparse-aware hardware.
- Structured Pruning:
  - Lower latency, and smaller on-chip buffers because the network's dimensions (channels, filters, branches) actually shrink, making it better fit for fixed-size accelerators and systolic arrays.



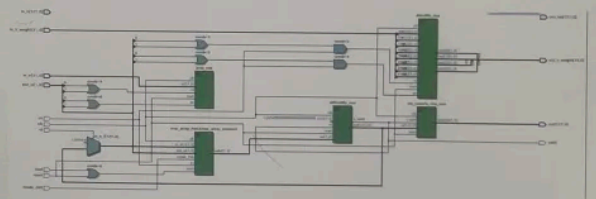
Method	Sparsity	Acc. After Pruning	Acc. After Training
Unstructured	0.8008	10%	89%
Structured	0.8	10%	79%

- Combined pruning
  - By doing both pruning methods, we can minimize the number of calculations while maintaining reasonable sparsity and accuracy.



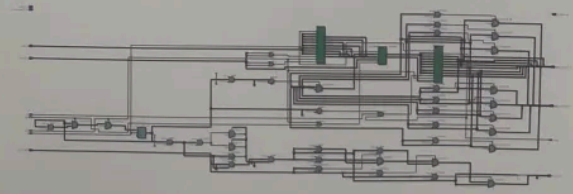
Alpha 6. Reconfigurable OS-WS (4bit-2bit) Systolic Array

It features 2-bit and 4-bit reconfigurable Processing Units and an optimized **mac.v** that processes two weights and two activations for flexible precision operations. The **mac_tile.v** is enhanced to manage WS and OS flows for weights and partial sums, with **os** and **mode_2bit** control signals allowing dynamic switching between modes in the accelerator.



Alpha 5. RTL Layout(Guided Nij)

Observed 66% reduction in Cycles after Synthesis using Quartus Prime.



References.

[1] Prasanth Varasala, Babulu Karapa, Kamaraju Maddu, "Intelligent Clock Gating for FPGA-based RISC Architectures: A Novel Approach to Switching Activity and Dynamic Power Reduction," *IJC*, 2025