SIMD and Weight-Output reconfigurable 2D systolic array-based AI accelerator and mapping onCyclone IV GX

XYZ Group: Changli Liu, Jiabao Xu, Wenhao Zhao, Xinqi Li, Zijian Li

VGGNet

Model Architecture	Precision Config	Accuracy	Quant Error
VGG_4b4b	4-bit/4-bit	91.36	2.82E-07
VGG_2b4b	2-bit/4-bit	89.51	1.39E-06

FPGA synth

	Vanilla
Total Operation per Cycle	128
Frequency	127.99MHz
Dynamic Power(mW)	34.01mW
TOPs	0.0164
TOPs/W	3.80E-09
Resource Utilization	17,157 / 149,760 (11 %)

Alpha1: Resnet20-QuantAwareTraining

We reconstructed Layer 2 to include a specialized "Squeeze-Target-Expand" sequence, featuring a central 8-channel convolution without Batch Normalization

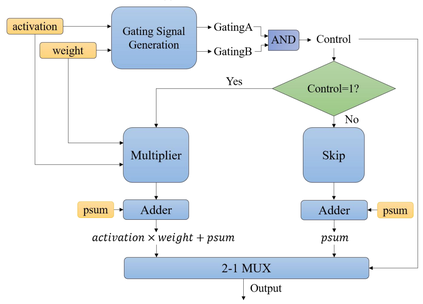
Precision Config	Accuracy	Quant Error
4-bit/4-bit	89.58%	1.69E-06

```
(1): QuantBasicBlockNoBN(
  (conv1): QuantConv2d(
    8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
    (weight_quant): weight_quantize_fn()
  )
  (relu): ReLU(inplace=True)
  (conv2): QuantConv2d(
    8, 8, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False
    (weight_quant): weight_quantize_fn()
  )
```

Alpha2&3: Gating PE

We perform 2 types of gating PE

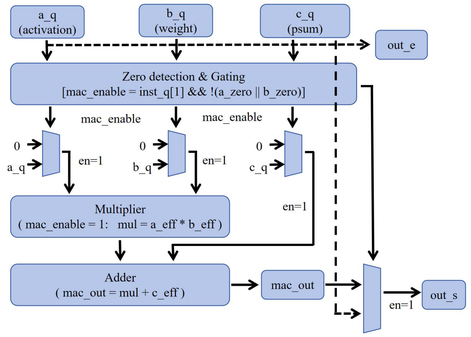
First gates the MAC operation by skipping the multiplier–adder entirely and directly forwarding the partial sum whenever the control signal indicates that the computation should be bypassed.



Data Match	Execute Cycles	Gating Cycles	Gating Ratio
Yes	52	30	57%

The Simulation's Results

Second gates the MAC operation by zeroing out the operands through input multiplexers based on the mac_enable signal, so that the multiplier–adder datapath still operates but performs only effective computations.



We evaluate the dynamic power consumption in comparison to the weight-stationary baseline.

	Vanilla (Weight Stationary)	Data_gating
Core Dynamic Thermal Power Dissipation	160.50 mW	142.55 mW

Alpha4: Part2-tiling

We perform the tiling for 16*16 and verify all the output channel

```
########## Now begin 2b4b testbench ##########
No.          0 th och_tile, kij =          0 execution completed.
No.          1 th och_tile, kij =          0 execution completed.
No.          0 th och_tile, kij =          1 execution completed.
No.          1 th och_tile, kij =          1 execution completed.
No.          0 th och_tile, kij =          2 execution completed.
No.          1 th och_tile, kij =          2 execution completed.
No.          0 th och_tile, kij =          3 execution completed.
No.          1 th och_tile, kij =          3 execution completed.
No.          0 th och_tile, kij =          4 execution completed.
No.          1 th och_tile, kij =          4 execution completed.
No.          0 th och_tile, kij =          5 execution completed.
No.          1 th och_tile, kij =          5 execution completed.
No.          0 th och_tile, kij =          6 execution completed.
No.          1 th och_tile, kij =          6 execution completed.
No.          0 th och_tile, kij =          7 execution completed.
No.          1 th och_tile, kij =          7 execution completed.
No.          0 th och_tile, kij =          8 execution completed.
No.          1 th och_tile, kij =          8 execution completed.
```

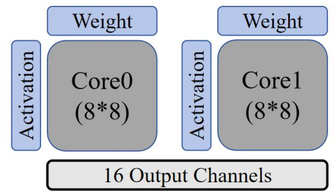
Alpha5: Pruning

We take two strategies based on unstructured pruning: one-shot and gradual pruning.

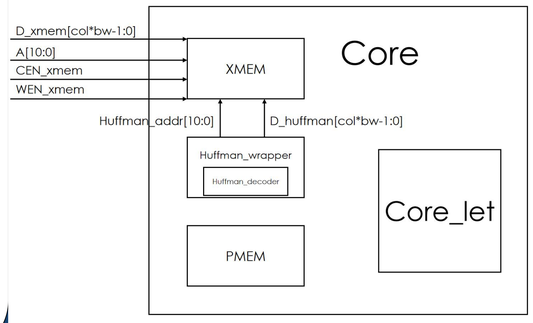
Model	Accuracy	Accuracy decline	Pruning Rate	Fine-Tune Epochs
VGG Baseline	88.670%	0.00%	None	None
One-shot Pruning	87.230%	1.44%	0.8	10
One-shot Pruning	84.940%	3.74%	0.8	None
One-shot Pruning	**88.160%**	**0.51%**	**0.9**	**10**
One-shot Pruning	54.160%	34.51%	0.9	None
Gradual Pruning-2times	82.420%	6.25%	0.8	10
Gradual Pruning-4times	82.750%	5.92%	0.8	10
Gradual Pruning-3times	61.430%	27.24%	0.9	10
Gradual Pruning-6times	70.330%	18.34%	0.9	10

Different Unstructured Pruning Strategy Results

Alpha6: Testbench dual core



Alpha7&8: Huffman encoding&decoder




```
Total bytes processed: 288
Original size: 2304 bits
Compressed size: 560 bits
Compression ratio: 4.11
Encoded data saved to 'output_file'
```