AI Accelerator with 2D Systolic Array and SIMD with weight/output stationary reconfiguration

Ashwin Rohit Alagiri Rajan, Hejia Zhang, Siyang Lai, Xinjun Hua, Yongshi Zhan, Zihan Ling
Team Vector

Abstract

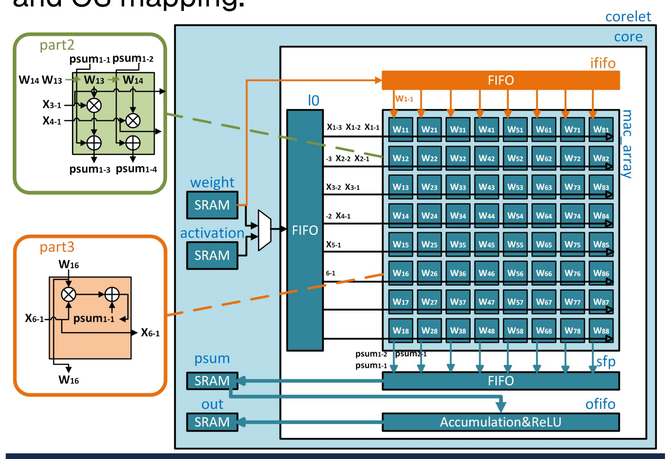
For the final project of ECE 284, our team, Vector, built an AI accelerator with an 8x8 2D systolic array with support for SIMD and weight/output stationary reconfiguring. To evaluate the correctness and performance of our accelerator, we trained and ran inference for VGG-16 with (1) 4 bit weight and activation quantisation, and (2) 4 bit weight and 2 bit activation quantisation schemes. We also implemented a variety of “alphas” on top of the base implementation which further supplanted the functionality and performance of the accelerator.

Software

We trained VGG-16 with custom 8x8 and 16x16 convolution layers for the software section. We trained the model in 2 sequential steps. **Step 1**: Training the model in Quantisation Aware Training (QAT) scheme with 4 bit activation and 4 bit weight (4A4W) quantisation. For the optimiser, we used SGD with starting parameters set to $lr=10^{-2}$, $decay=10^{-5}$, *epochs=300, momentum=0.9.* For the scheduler, we use *CosineAnnealing.* The combination of SGD (w/ momentum) and CosineAnnealing are the recommended methods in industry for QAT. **Step 2**: In this step, we took our previously trained 4A4W model and simply fine tuned it for 2A4W scheme. For fine tuning, we maintained the same setup, except in this case, the quantised parameters were trained with $lr=10^{-3}$, *decay=0.* We were able to achieve 91% and 88% accuracy respectively.

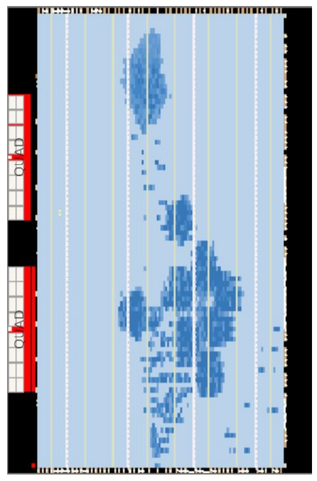
Hardware

Our accelerator features an 8×8 2D systolic MAC array backed by dedicated SRAM blocks for weights, activations, psums, and outputs. An L0 FIFO supplies PEs with weights and streamed activations, while an output FIFO aligns partial sums before they are written to a double-buffered SRAM for accumulation by a special function processor. The core is runtime-configurable to operate in 2A4W or 4A4W precision modes, and support both WS and OS mapping.

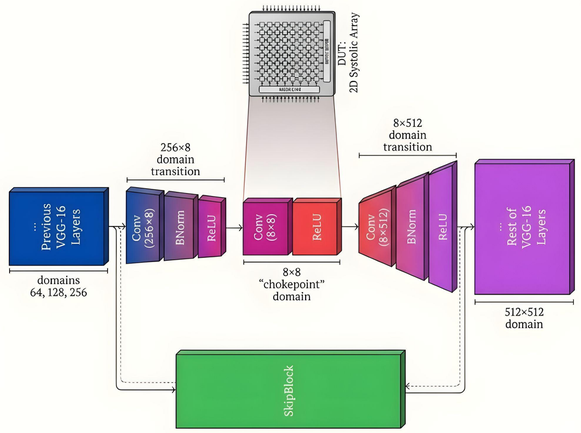


FPGA Synthesis

In FPGA synthesis, our vanilla version design achieves a maximum frequency of 109.96MHz when working at the slow corner(1200mV, 100C), with a total power consumption of 282.14mW. It occupies 16865 4-input LUTs and 12224 registers. TOPs/W=128/0.28214= 453.68



Alpha 1 & 2 (Software Alphas)

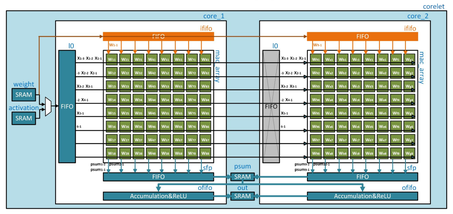


Alpha 1: To assist with training and to improve gradient flow, we created a block called the *SkipBlock*. This is a block made of a Conv, BNorm, and ReLU that sits beside the chokepoint and transitions between the 256 and 512 domains. This is not used during inference but rather during pre-QAT training where the “effective resistance” for gradient flow is minimised and we’re able to achieve >90% accuracy within the first 35 epochs. For QAT, this block is disabled while the weights of the chokepoint are now allowed to be trained. This led to training for 4A4W and 2A4W within 50 and 75 epochs respectively compared to the baseline QAT training where it took >250 epochs.

Alpha 2: We implemented PQT for Bidirectional Encoder Representations from Transformers (**BERT**) model. The base model has an encoder transformer architecture with around 110 million parameters. BERT is a pre-trained model so our evaluation was based on MSE on semi-quantised BERT between our systolic array and NVIDIA RTX 2080 Ti. We ran PQT on the FFN for the BERT model and achieved parity with the PyTorch variant.

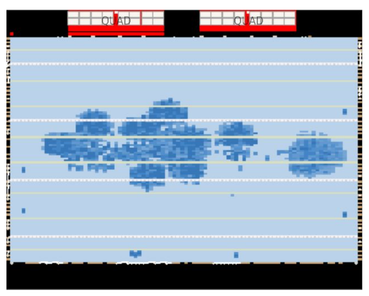
Alpha 3 & 4 (Hardware Alphas)

Alpha 3: Multi-Core Wrapper for Tiled Layer



In 2A4W mode, the convolution layer needs to be tiled into 2 pieces for operation of overall 16 output channels, which can be replaced by multi-core wrapper.
Instantiate two core.v module in the wrapper to implement the function of 8b×16b systolic array. The out_e and inst_e signals of the 1st array's last column are passed to the 2nd array's first column.
This method increases the efficiency of data processing with no need to load weight and activation twice.

Alpha 4: Our fourth alpha will be performing FPGA synthesis for the multi-core wrapper in Alpha 3. By instantiating two core modules and chaining the east out/inst signals from the first array into the second, the system functions as an effective 8×16 systolic array, eliminating the need to tile the convolution layer into two passes in 2A4W mode, nearly doubling output throughput while avoiding redundant weight and activation loads. We expect the multi-core version to finish the same task in roughly half of original time.



Current implementation
freq:103.9MHz
Pow: 291.9MW
LUTs: 18831
reg: 13416