16.1 DIMC: 2219TOPS/W 2569F/b

Digital In-Memory Computing Macro in 28nm Based on Approximate Arithmetic Hardware

Dawei Wang1, Chuan-Tung Lin1, Gregory K. Chen2, Phil Knag2, Ram K. Krishnamurthy2, Mingoo Seek2

1Columbia University, New York, NY
2Intel, Portland, OR

In-memory-computing (IMC) SRAM architecture has gained significant attention as it achieves high energy efficiency for computing a convolutional neural network (CNN) model [1]. Recent works investigated the use of analog-mixed-signal (AMS) hardware for high area and energy efficiency [2, 3]. However, AMS hardware output is well known to be susceptible to process, voltage, and temperature (PVT) variations, limiting the computing precision and ultimately the inference accuracy of a CNN. We reconfirmed, through the simulation of a capacitor-based IC SRAM macro that computes a 256D binary dot product, that the AMS computing hardware has a significant root-mean-square error (RMSE) of 22.5% across the worst-case voltage, temperature (Fig. 16.1.1 left top) and 3-sigma process variations (Fig. 16.1.1 top right). On the other hand, we can implement an IC SRAM macro using robust digital logic [4], which can virtually eliminate the variance issue (Fig. 16.1.1 top). However, digital circuits require more devices than AMS counterparts (e.g., 28 transistors for a mirror full adder [FA]). As a result, a digital IC SRAM macro shows a lower area efficiency of 3638F/b (22nm, 4b/4b weight/activation) [5] than the AMS counterpart (1170F/b, 65nm, 1b/1b) [3]. In light of this, we aim to adopt approximate arithmetic hardware to improve area and power efficiency and present two digital IC macros (DIMC) with different levels of approximation (Fig. 16.1.1 bottom left). Also, we propose an approximation-aware training algorithm and a number format to minimize inference accuracy degradation induced by approximate hardware (Fig. 16.1.1 bottom right). We prototyped a 28nm test chip: for a 1b/1b CNN model for C1FAR-10 and across 0.5-to-1.1V supply, the DIMC with single-approximate hardware (DIMC-S) achieves 3814F (4b-990TOPS/W (normalized to 1b/1b) 405-19215GOPS (normalized to 1b/1b), and 90.41% accuracy. Double-approximate hardware (DIMC-D) achieves 2569F (1b-1b) 405-19215GOPS (normalized to 1b/1b), and 89.0% accuracy. To compensate for the inaccuracy induced by the approximate hardware, we developed an approximation-aware training algorithm. In this algorithm, the forward path performs the vector-matrix multiplication using a bitwise operation considering the approximate hardware. Gradient calculations are performed using full accuracy. We then benchmarked the proposed hardware against the recently trained VGG-like 1b/1b CNN model and CIFAR-10. The double-approximate version achieved higher accuracy of 86.9%, and the single-approximate version achieved 89.0%—close to the exact hardware (Fig. 16.1.4 top left).

Interestingly, even with the approximation-aware training, the approximate hardware still results in lower accuracy for a multi-bit activation CNN model (Fig. 16.1.4 bottom right) because multi-bit activation tends to require more accurate hardware [3]. Specifically, multi-bit activations are often Gaussian distributed and thus MSBs are sparse and suffer from approximate errors. To improve the accuracy of a multi-bit activation CNN, we propose a new number format called multi-bit XNOR (MB-XNOR). Conventionally, in a 1b/1b neural network, each weight and activation represents ±1 or -1 and XNOR realizes bitwise multiplication. If we use the 2’s complement format for activations, however, the binary weight also needs to be in 2’s complement and can represent only -1 or 0. We found that this results in large degradation to CNN accuracy. Therefore, we extended the format of the binary weight to represent an N-bit activation $b_{1:b}$, $b_{i:b} = \Sigma b_i \times b_{i-2}$, where $b_i$ is +1 or -1. This format cannot represent 0, which disallows some of the activation functions such as ReLU. However, we can still use other popular activations such as hyperbolic tangent (tanh) (Fig. 16.1.4 top right) and leaky ReLU.

We confirmed that the proposed MB-XNOR format improves the accuracy of a multi-bit activation CNN model. We investigated the improvement both in SNR (signal-to-noise ratio) simulation and via CNN accuracy measurement. SNR is formulated as:

$$SNR = \frac{\sum_{i} y_{i}^2}{\sum_{i}(y_{i} - \mu)^2},$$

where $y_{i}$ is the ground truth of the dot product between a 256D Gaussian-distributed input vector quantized to 1-to-4b and a 256D binomial-distributed weight vector. $\mu_{y_{i}}$ is the same dot product but computed with approximate hardware. The DIMC-D macro with the 4b input activations in the MB-XNOR format yields a 0.15 higher SNR than 2’s complement (Fig. 16.1.4 bottom left). The CNN accuracy measurement confirms the same improvement: DIMC-S using the MB-XNOR successfully increases the CNN accuracy by 5.4% (Fig. 16.1.4 bottom right). Despite that DIMC-D also benefits from the MB-XNOR format, the accuracy with multi-bit activations is still lower than that with binary activations, making DIMC-D suitable for only a 1b/1b weight/activation CNN model.

We prototyped the DIMC test chip in 28nm (Fig. 16.1.7). The 16b DIMC-D (DIMC-S) takes 0.033mm$^2$ (0.049mm$^2$), implying an area efficiency of 2569F/b (3814F/b). We measured the area at 0.5-1.1V at 25°C. DIMC-D achieves 932-2219TOPS/W and 475-20023GOPS; DIMC-S 458-990TOPS/W and 405-19215GOPS (normalized to 1b/1b for comparison) (Fig. 16.1.5 top left). We also measured the energy efficiency and throughput across five chips at the nominal voltage 0.9V (Fig. 16.1.5 top right), the energy efficiency across supply voltage at 25% and 50% input toggle rates (Fig. 16.1.5 bottom left). The power breakdown is shown in Fig. 16.1.5 bottom right. The SRAM module takes 340ns (256 cycles at 752MHz) to update in total 16kb weights at 0.9V. Figure 16.1.6 shows the comparison to the recent work. The proposed DIMC macros achieve the high area efficiency, while maintaining the state-of-the-art throughput, energy-efficiency and CNN accuracy.

Acknowledgement:
This work is supported by NSF (PFI-RP1919147) and SRC (TaiTAC 2810.034).

References:
Figure 16.1.1: Voltage and temperature (top left) and process (top right) variations affect AMS computing hardware’s accuracy. Approximate hardware improves the area efficiency of DIMC SRAM (bottom left). The custom training and number format improves CNN accuracy for CIFAR-10 (bottom right).

Figure 16.1.2: Proposed DIMC architecture (left). Three compressor schematics and the corresponding transistor count (middle). The RMSE of 256D binary dot product utilizing three types of compressors (right).

Figure 16.1.3: Two 12T-FA schematics with either regular inputs or inverter-buffered inputs (left). Layout of the 12T FA circuits (top right). Schematics of 4b RCA (middle right) and digital arithmetic hardware of one column (bottom right).

Figure 16.1.4: CIFAR-10 accuracy of conventional training and approximation-aware training (top left). Tanh activation quantized to 3b in the MB-XNOR format (top right). The MB-XNOR format offers better SNR (bottom left) and CIFAR-10 accuracy (bottom right) compared with 2’s complement.

Figure 16.1.5: Measurement results. Energy-efficiency and throughput across different supply voltages (top left). Multi-chip measurements at 0.9V supply (top right). Energy-efficiency at 25% and 50% toggle rates (TR) (bottom left). Power breakdown of two proposed DIMC macros (bottom right).

Figure 16.1.6: Comparison with recent IMC SRAMs using AMS or digital arithmetic hardware.
Figure 16.1.7: Die micrograph and area breakdown.