14.3 A 28nm SoC with a 1.2GHz 568nJ/Prediction Sparse Deep-Neural-Network Engine with >0.1 Timing Error Rate Tolerance for IoT Applications

Paul N. Whatmough, Sae Kyu Lee, Hyunkwang Lee, Saketh Rama, David Brooks, Gu-Yeon Wei

Harvard University, Cambridge, MA

Machine Learning (ML) techniques empower Internet of Things (IoT) devices with the capability to interpret the complex, noisy real-world data arising from sensor-rich systems. Achieving sufficient energy efficiency to execute ML workloads on an edge-device necessitates specialized hardware with efficient digital circuits. Razor systems allow excessive worst-case V_{DD} guardbands to be minimized down to the point where timing violations start to occur. By tracking the non-zero timing violation rate, process/voltage/temperature/aging (PVTA) variations are dynamically compensated as they change over time. Resilience to timing violations is achieved using either explicit correction (e.g., replay [1]), or algorithmic tolerance [2]. ML algorithms offer remarkable inherent error tolerance and are a natural fit for Razor timing violation detection without the burden of explicit and guaranteed error correction. Prior ML accelerators have focused either on computer vision CNNs with high-power (e.g., [3] consumes 278mW) or spiking neural networks with low-accuracy (e.g. 84% on MNIST [4]). Programmable fully-connected (FC) deep-neural-network (DNN) accelerators offer flexible support for a range of general classification tasks with high accuracy [5]. However, because there is no parameter reuse in FC layers, both compute and memory resources must be optimized.

This paper presents a 28nm SoC with a programmable FC-DNN accelerator design that demonstrates: (1) HW support to exploit data sparsity by eliding unnecessary computations (4× energy reduction); (2) improved algorithmic error tolerance using sign-magnitude number format for weights and datapath computation; (3) improved circuit-level timing violation tolerance in datapath logic via time-borrowing; (4) combined circuit and algorithmic resilience with Razor timing violation detection to reduce energy via V_{DD} scaling or increase throughput via F_{CLK} scaling; and (5) high classification accuracy (98.36% for MNIST test set) while tolerating aggregate timing violation rates $>10^{-1}$. The accelerator achieves a minimum energy of 0.36μJ/pred at 667MHz, maximum throughput at 1.2GHz and 0.57μJ/pred, or a 10%-margined operating point at 1GHz and 0.58μJ/pred.

The SoC (Fig. 14.3.1) is based around an ARM Cortex-M0 cluster. The DNN engine connects through an asynchronous bridge, allowing independent F_{CLK} and V_{DD} scaling to balance throughput and energy efficiency. A 4-way banked on-chip memory (W-MEM) stores the weights for the DNN model (up to 1MB) and provides low-latency access to the DNN engine.

The DNN Engine (Fig. 14.3.2) is a 5-stage SIMD-style programmable sparse matrix-vector (MxV) machine for processing arbitrary DNNs. A sequencer dynamically schedules operations for different DNN configurations—up to eight FC layers with 1-to-1024 nodes per layer. The host CPU loads the input vector into the IPBUF scratchpad and the 8-way MAC datapath processes eight concurrent neuron computations at a time, fed from either IPBUF (input layers) or XBUF (hidden layers). Once the in-flight neurons have accumulated all the weight-activation products, the activation stage adds a bias term and applies a rectified linear unit (ReLU) activation function. The resulting neuron activations are written back to XBUF, which is double buffered to allow simultaneous reads from the previous layer and writes to the current layer. The MAC unit uses optimized 16b fixed-point precision throughout, with support for programmable rounding modes, two's compliment (TC) or sign-magnitude (SM) numbers, and 8b or 16b weight precision.

We exploit the abundant dynamic sparsity in the input data and activations. Prior work clock-gates functional units to save power for zero operands, but still consume cycles for pipeline bubbles [3,5]. Instead, we eliminate bubbles by dynamically eliding all zero operands at XBUF writeback. Moreover, we also skip even small non-zero values, without degrading prediction accuracy, leading to further savings. After ReLU, the activation stage compares output activations against per-layer programmable thresholds to produce a SKIP control signal that predicates writeback of data to XBUF. A small 512B SRAM (NBUF) keeps the list of active node indexes in the previous layer, from which W-MEM addresses are generated. For MNIST, the average number of loads, ops, and cycles are reduced by over 75%, significantly improving energy and throughput (Fig. 14.3.5).

To enhance error-tolerant operation, the DNN accelerator augments two timing-critical stages, W-MEM load and MAC unit, with Razor flip-flops (RZFFs) on timing end-points. The MUX cell in the dual-mode RZFF (Fig. 14.3.2) supports operation as a datapath FF or latch with time borrowing. A global pulse clock (90-to-300ps pulse width) defines both the timing detection window and latch transparency time, while satisfying hold delays set at design time. All other paths include 30% margin.

Figure 14.3.3 plots measured power and timing violation rate (measured at word granularity) results running MNIST inference on a 784×256×256×256×10 DNN for TC and SM number formats with 16b weights. The design targeted a signoff F_{MAX} of 667MHz (1.5ns period) under worst-case conditions (SS, 0.81V, 125°C). At 667MHz, V_{DD} can scale from nominal (0.9V) down to 0.77V before on-chip counters record the first timing violation, translating voltage margin to 30% power reduction (Fig. 14.3.3 top). Alternatively, at 0.9V, F_{CLK} can scale beyond 1GHz before the first timing violations occur.

Further improvements are possible by leveraging inherent resilience of the DNN. Fig. 14.3.4 plots measured classification accuracy vs. timing violation rates for W-MEM loads, datapath MACs, and the combination. For the memory, SM numbering exploits the zero-mean Gaussian distribution of the weights matrix to reduce switching activity in the MSBs and thus bit-flips. Adding a bit-masking (BM) technique to mask individual bit errors in the weight word allows the accelerator to tolerate SRAM read timing violation rates $>10^{-1}$ at 98.36% accuracy. Error tolerance in the datapath is harder to achieve because bit-flips persist in the accumulator. Although SM offers some benefit, circuit-level time borrowing is much more effective, tolerating timing violation rates commensurate to levels seen for the memory. Generous time borrowing from the accumulator is possible through the feedback path of the adder in the MAC unit (Fig. 14.3.2). Together, timing violation tolerance improves by several orders of magnitude at 98.36% accuracy, over the whole 10k vector MNIST test set, which supports further V_{DD} reduction to 0.715V (no margin).

Figure 14.3.5 summarizes energy and throughput improvements offered by different optimizations and techniques. Overall, energy reduces by >9× down to 0.36μJ/pred via 8b weights, aggressive V_{DD} scaling (no margin), and exploiting sparsity (skip). Fig. 14.3.6 shows the Pareto frontier of prediction accuracy vs. energy emerging from running different network topologies on the test chip with comparisons to all other measured HW that reports MNIST energy and accuracy results. Fig. 14.3.7 provides chip details and die microphotograph.

Acknowledgements:
This work is supported in part by DARPA under Contract No. HR0011-13-C-0022. We are grateful to ARM for providing physical IP.

References:

[1] D. Bull, et al., "A Power-Efficient 32b ARM ISA Processor Using Timing-Error Detection and Correction for Transient-Error Tolerance and Adaptation to PVT variation," *ISSCC*, pp.284-285, 2010.

[2] P. N. Whatmough, et al., "A Low-Power 1GHz Razor FIR Accelerator with Time-Borrow Tracking Pipeline and Approximate Error Correction in 65nm CMOS," *ISSCC*, pp. 428-429, 2013.

[3] Y. Chen, et al., "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *ISSCC*, pp. 262-263, Feb. 2016.

[4] J. Kim, et al., "A 640M Pixel/s 3.65mW Sparse Event-Driven Neuromorphic Object Recognition Processor with On-Chip Learning," *IEEE Symp. VLSI Circuits*, 2015.

[5] B. Reagen, et al., "Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators," *ACM/IEEE Int. Symp. Computer Arch.*, pp. 267-278, 2016.

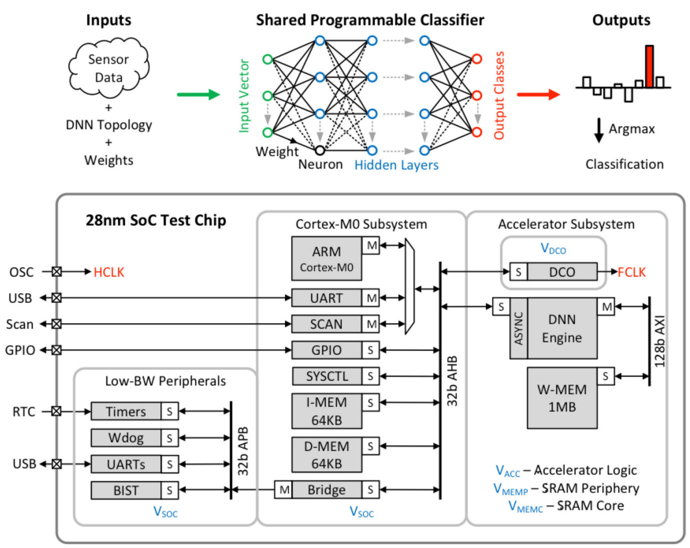

Figure 14.3.1: System block diagram of 28nm SoC with DNN engine.

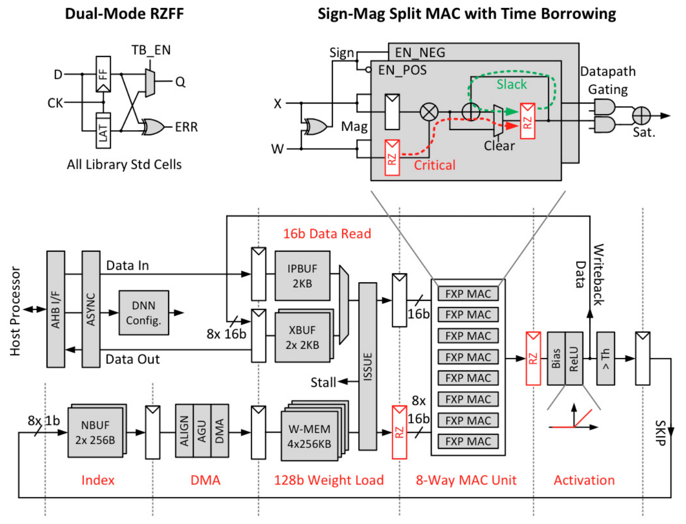

Figure 14.3.2: Simplified microarchitecture of the five-stage DNN accelerator, split sign-magnitude (SM) accumulator design and RZFF.

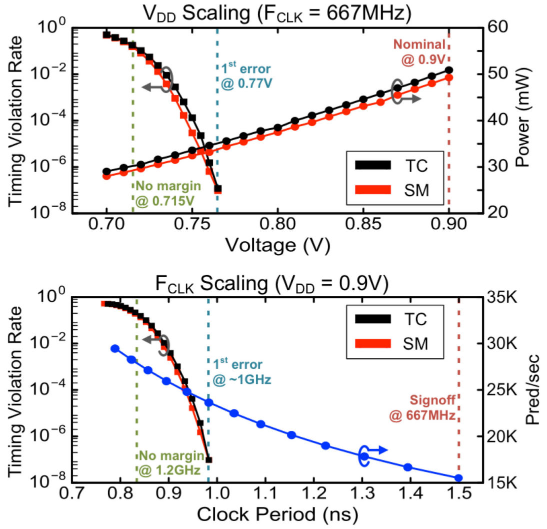

Figure 14.3.3: Power and timing error rate results for voltage scaling at sign-off FMAX of 667MHz (top), and frequency scaling at 0.9V (bottom).

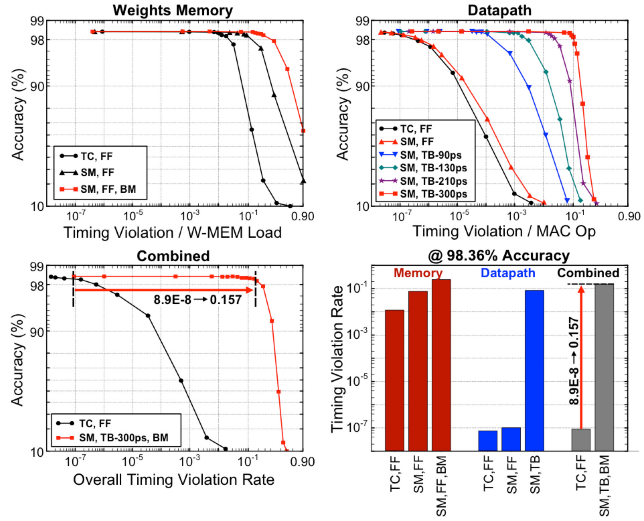

Figure 14.3.4: MNIST accuracy vs. timing error rates in the W-MEM SRAM, MAC datapath, and combined.

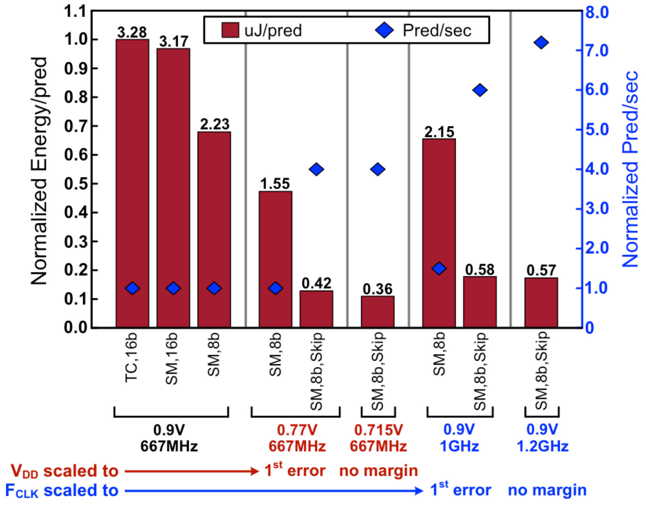

Figure 14.3.5: Energy/pred and throughput across different configurations at 98.36% accuracy for the MNIST test set.

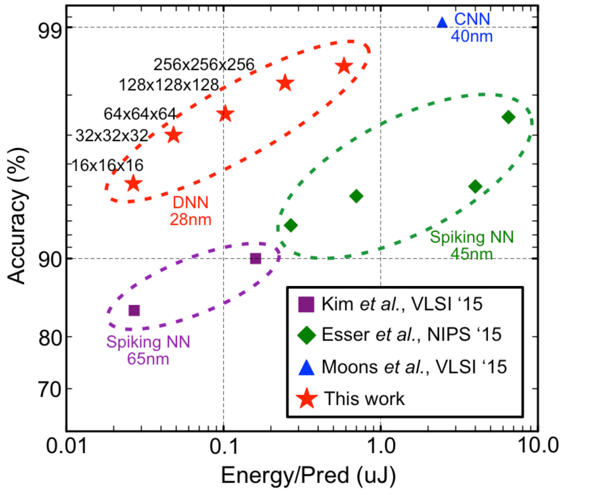

Figure 14.3.6: MNIST accuracy vs. energy/pred for multiple topologies on the test chip and other reported hardware measurements.

Process Tech.	TSMC 28nm HPC 1P10M
Die Size	2.4mm x 2.4mm
Total SRAM	SoC: 128KB / W-MEM: 1MB / DNN-Engine: 6.5KB
Total FFs	8460 (896 RZFFs)
ML Model	FC-DNN Classifier
Weight Precision	8-bit / 16-bit Fixed-Point
Native Model Size	Hidden Layers: 0-6 Nodes/Layer: 1-1024
Error Tolerance	>10^{-1} @ 98.36% Accuracy
Supply Voltage	0.9V (nom.) 0.6 – 1.1 V (operational)
F_{MAX}	667MHz @ 0.9V 1.2GHz @ 0.9V (w/ Razor)
Power Consumption	33.7mW @ 667MHz/0.9V 22.4mW @ 667MHz/0.715V 63.5mW @ 1.2GHz/0.9V
Leakage	3.03mW @ 0.9V
RZFF Overhead	3.24x area per RZFF

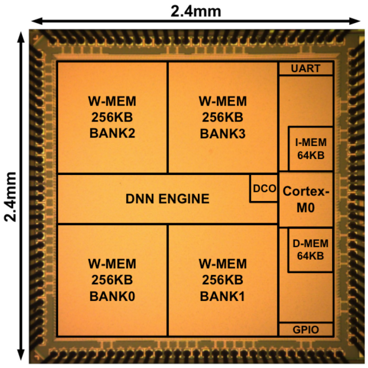

Figure 14.3.7: Chip summary and annotated microphotograph.