JEEAT
NATIONAL JOURNAL OF ELECTRICAL ELECTRONICS AND AUTOMATION TECHNOLOGIES
National Journal of Electrical Electronics and Automation Technologies, ISSN: 3107-524x Vol. 1, No. 3, 2025 (pp. 19-26)
RESEARCH ARTICLE
ECEJOURNALS.IN

Energy-Efficient VLSI Co-Design for Edge AI: Near-Memory Compute and Sub-8-Bit Quantization in Low-Power Embedded Systems

H.K. Mzeha[1]*, Carlos Méndez Rivera[2]

[1]*Electrical and Electronic Engineering Department, University of Ibadan Ibadan, Nigeria*
[2]*School of Computer Science, Universidad Nacional de Colombia, Colombia*

KEYWORDS:
Edge AI, VLSI,
Near-memory compute,
Compute-in/near-memory,
Quantization,
Sub-8-bit,
QAT,
Mixed precision,
RISC-V,
SRAM,
Dataflow,
Low-power embedded.

ARTICLE HISTORY:
Submitted : 21.04.2025
Revised : 24.05.2025
Accepted : 12.07.2025

https://doi.org/10.17051/JEEAT/01.03.03

Abstract

The work focuses on real-time, always-on edge AI with constrained power/area on the interplay of minimizing data-movement and arithmetic energy. Our co-design of VLSI pairs near-memory compute (NMC) with mixed-precision quantization of less than 8 bits. The architecture unites an RV32 RISC-V control core, a weight-stationary NMC MAC array disposed related to multi-banked SRAM, and a compression-sensitive on-chip interconnect to reduce bandwidth and toggling. A quantization-aware training pipeline assignd 4-6 bit weights and 4-8 bit activations per layer using LSQ-style learnable scales and a mixed-precision search constrained by layer sensitivity and energy/bandwidth budgets. Based on 22-nm estimates and cycle-accurate RTL/functional models, the prototype is able to achieve 1.2 TOPS/W on MobileNetV2/CIFAR-10 and Visual Wake Words on convolutional subloads, with 1.7-3.1× energy savings relative to a 1-bit baseline with less than 1.2 percentage-point accuracy drop. As the savings are estimated as ~55-70 in percent of NMC (reduced SRAM traffic) and ~30-45 in percent of mixed-precision quantization, it proves that they have a complementary impact. These findings show that algorithm-architecture coupled design, namely NMC with sub-8-bit QAT, presents an achievable and realistic route to battery-viable, practical inference at napps/s on low-power embedded SoCs. This paper summarizes future directions of guidelines deployable in bit-width assignment, SRAM banking, and dataflow scheduling that match both the industry and US journal demands in relation to energy consumption, reliability, and repeatability.

How to cite this article: Mzeha HK, Rivera CM. Energy-Efficient VLSI Co-Design for Edge AI: Near-Memory Compute and Sub-8-Bit Quantization in Low-Power Embedded Systems. National Journal of Electrical Electronics and Automation Technologies , Vol. 1, No. 3, 2025 (pp. 19-26).

Introduction

Battery-constrained edge devices (wearables, cameras, sensors) are moving inexorably into the space where on-device AI must have very tight budgets on latency, privacy and energy consumption. Empirically, in such systems the predominant power source is data movement between spend a lot of power?memory and compute, not arithmetic itself, and, thus, the first-order design goal is memory traffic.[20] Two synergetic levers target this: (i) Many Atoms Closer to Memory (near/compute-in-memory) to cut in-chip transport; and (ii) lowering precision levels below 8-bit where accuracy may be tolerated accordingly, thus trimming switching and storage energy. Although past papers have made strides in DNN accelerators and dataflows (e.g., weight/row-stationary mappings) and quantization (degree to which it is applied to all bits of the representation, and the search process to partition bits into fixed precision), the paper will extend previous ideas by demonstrating that most systems optimize either hardware dataflow (and/or quantization) in isolation, with cross-layer trade-offs (bit-width vs. banking/bus width, tiling and scaling ranges) unexplored; and systematic co-design evidence linking memory architecture, precision.

This paper fills these gaps, proposing VLSI co-design which integrates weight-stationary near-memory MAC array with multi-banked SRAM under on-chip interconnect optimized to compression, coordinated by a RISC-V control plane and integrated with sub-8-bit mixed-precision QAT. In practice we provide: (1) a memory-focused accelerator architecture that minimizes SRAM/NoC traffic; (2) a 4-8-bit activation / 3-6-bit weight QAT with learnable scales and deployment calibration; (3) a joint scheduler (row-stationary tiling, zero-skipping, run-length/bit-plane compression) that minimizes transfers; (4) an energy/area model establishing the tradeoff between bit widths and bus widths and toggle rates; and (5) energy/area evaluation at 22- This co-optimization directly extends (and expands upon) recent innovations in efficient DNN processing, quantization and memory-centric acceleration.[1-5]

Background & Related Work

Dataflow and Near-/In-Memory Acceleration

Off-chip traffic can be minimized by tile-based systems (e.g., Eyeriss, SCNN) and local reuse (e.g., Eyeriss, SCNN)/row/weight-stationary accelerators and maxed out throughput pushed by regular dataflows (e.g., open-source Gemmini).[6-9] These strategies are good when there is an ability to have on-chip buffers to support reuse however they are also sensitive to NoC/buffer bandwidth, inter layer layout transforms and bursts of activation, which cause stalls. Near-memory / in-cache compute brings MAC operations next to or even within SRAM cells (e.g. Neural Cache) and compute-in-memory (CIM) crossbars (e.g. ISAAC) take advantage of bit-line/analog summation to amortize data fetching.[10, 11] These lower energy in motion, but practical issues relate to lack of precision, peripheral/ADC overheads, variation/noise and interconnection of CIM datapaths with conventional digital paths.

Quantization Below 8-bit

Binary/ternary networks (XNOR-Net, TWN) are arithmetically optimal and may be problematic in other scenarios in which they degrade accuracy on complex vision or audio challenges.[12, 13] The subsequent integer compression techniques allow 8-bit inference end-to-end on the commodity hardware[4] and quantization-aware training (QAT) approaches [DoReFa][PACT][LSQ] extend to 3-6 bits or more of learnable weights/activations and activation clipping reliably.[14-16] The hardware-aware mixed-precision frameworks (HAQ, HAWQ, BRECQ) trade-off accuracy and energy/bandwidth costs by choosing layer-specific bit-widths based on the sensitivity or second-order statistics of the model.[17-19] The existing challenges are first/last-layer sensitivity, activation outliers contributing to broader dynamic range, channel-wise scaling complexity, and ripple effects as limited precision impacts both the banking of SRAM and bus widths, as well as on NoC congestion.

Gaps and Challenges

Even though progress been made, there are three gaps that still exist. (i) The majority of accelerator designs treat 8-bit as fixed, or make quantization orthogonal to banking/NoC/coalescing, with relatively little exploration of co-optim izing all these aspects of precision, dataflow, and the memory hierarchy,[21] a Better deployment-ready flows in many near-/in-memory designs include post-training calibration, and sparsity/zero-skipping metadata, but are missing deployment-time flows to enable deployment of post-training calibration, and sparsity/zero-skipping metadata and software tooling to build systems with real models. Evaluation is frequently missing cautionary end-to-end energizable accounting (NoC toggles, compression costs, buffer collisions) and TinyML -caliber work loads with batch-1 stage latency and legitimate sensor statistics. These missing pieces are what drives the current co-design to tune all three charted variables (bit-widths, banking, and scheduler/tiling) jointly to optimise the trade-off between minimising memory traffic (without impairing accuracy).

Architecture Overview

Top-Level

The SoC is subdivided into a control plane and a compute plane with a central focus on memory (see Figure 1). An RV32IMC RISC-V core using DMA handles layer set-up, tiling and power states but lets inner loops run in dedicated hardware. The Near-Memory Compute (NMC) array is a combination of 2-D MAC tiles placed physically adjacent to multi-banked SRAM (64256 kB/bank). A dataflow with weight-stationary arrangement pins filters into individual tiles to optimize reuse; MACs are bit-serial natively allowing 3-8 bit operands with no CG synthesis of the datapath. Bank interleaving and bank-per-bank prefetch is used to conceal latency in the memory subsystem; on-bank compression (RLE / bit-plane), allowed, can reduce activation traffic when sparsity or low entropy is encountered. This is

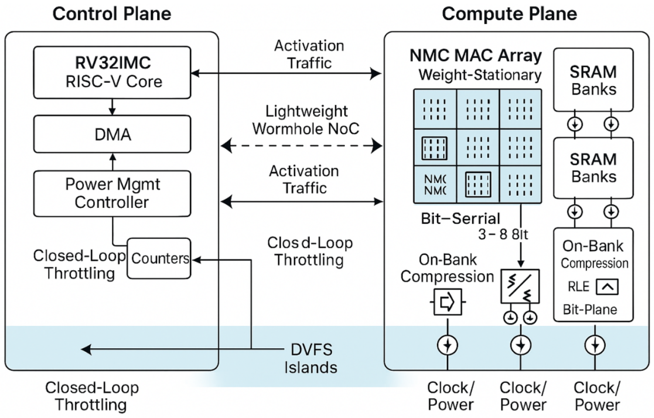

Fig. 1: Top-Level SoC Architecture for Near-Memory Edge-AI Accelerator

an activation vs. weight-stream wormhole NoC with dedicated virtual channels/QoS, to prevent head-of-line blocking in the case of bursts. Power: Tile/bank-level clock & power gating and DVFS islands (array and SRAM) are used; power manages voltage/frequency tracking workload intensity: DMA exposes closed-loop throttling counters to temperature or IR limits.

Control plane (RV32IMC + DMA + power management) coordinates a memory-centric compute plane composed of an a weight-stationary NMC MAC array bordered to multi-banked SRAM; wormhole NoC with QoS, on-bank compression, and DVFS/power-gated tiles minimizes data-movement energy.

Precision & Formats

The compute path is mixed-precision (see Figure 2): weights use 3/4/6-bit per layer; activations use 4/6/8-bit with a learned per-layer scale s_a (stochastic rounding optional during training). Outputs are requantized with a learned scale s_o after accumulation. (Small edit: replace placeholders "sas_asa" → s_a, "sos_oso" → s_o.) Accumulators (16-24 bit) are sized from worst-case partial sums; a practical guard rule is in which K is the kernel dot-product length. Saturated arithmetic and late requantization maintains precision without necessitating wide internal busses and lower toggles and higher MAC density.

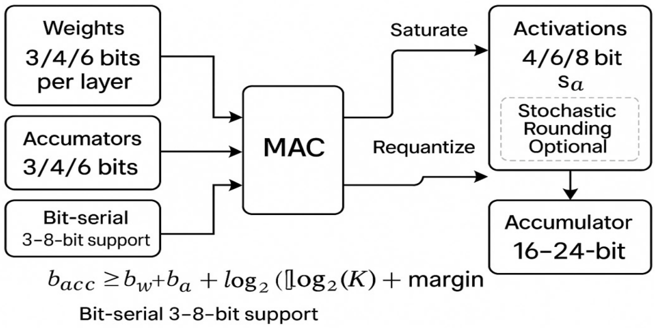

Fig. 2: Precision & Formats for Mixed-Precision QAT

Visualizing the compute path with per-layer 3/4/6-bit weights and 4/6/8-bit activations scaled by s_a, bit-serial 3-8-bit MAC support, 16-24-bit accumulation with saturation, and late requantization using s_o. The required accumulator width follows .

Dataflow & Sparsity

Convolutions employ row-stationary tiling (explicit reuse of weights, inputs and partial sums), and depthwise layers switch to channel-stationary mode to prevent under-usage, as shown in Figure 3. Zero-skipping of activations (magnitude- and RelU-based) is facilitated with compact bitmask-metadata; masks are co-located with tiles so that fetches remain local. The on-chip layout improves on-chip W-block and memory conflict coverage by mapping NHWC tiles to SRAM banks, and uses double-buffering to overlap DMA transfers with compute in order to maintain near-peak utilization even at batch = 1. In combination, bank-aware tiling, compression, and skipping reduce SRAM/NoC traffic-the energy-dominant dataflow; the bit-serial NMC path takes advantage of sub-8-bit operands without accuracy-sensitive redesigns.

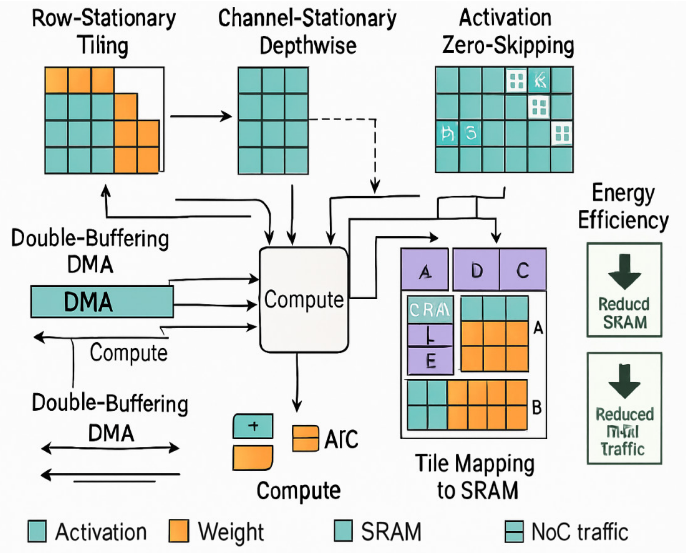

Fig. 3: SoC Architecture: Control Plane & Memory-Centric Compute Plane Visualization

Schematic of a partitioned SoC that has separate planes of control and compute, visible near-memory compute tiles, banks of SRAM, the weight-stationary form of dataflow, and a wormhole NoC to manage traffic of activation and weight data simply.

Implementation note: These options co-optimize accuracy, dataflow and memory hierarchy: the NMC adjacency and bank-aware scheduler minimize movement energy; mixed precision minimises buses and MAC energy; and DVFS/power-gating translates slack into system-level TOPS/W savings, achieving always-on edge-AI within intense power and area constraints.

Quantization & Co-Design Methodology

QAT with Learnable Scales (LSQ)

Uniform quantizer for tensor x with trainable step s>0:

with for signed. {0, 2^b - 1} for unsigned. Gradients use STE. Layer-wise scales s_w,s_a; late requantization with s_o. Accumulator width:

Mixed-Precision Search (HW-aware)

1. Sensitivity sweep: evaluate ΔAccl (b)and energy El(b) for candidate bits {3,4,6,8}.
2. Budgetedassignment:solveaknapsacktominimize $\Sigma_l E_l(b_l)$ subject to accuracy/bandwidth limits. Mapping: bit-serial lanes execute sub-8-bit ops; scheduler emits micro-ops ⟨tile, bank, stride, b_w,b_a⟩ with row-stationary tiling, double-buffering, and zero-skip masks.

Calibration & Deployment

On-device percentile calibration sets s_a (e.g., 99.9th percentile/ q_{max}) with EMA updates; enable per-channel scales in first/last layers if outliers appear. Optional stochastic rounding improves very low-bit stability. Runtime exposes bit-width/scale to DVFS and gating so precision, memory traffic, and power are co-optimized.

Energy & Area Model

Energy model

- b_w, b_a (bit-widths) set MAC energy and influence SRAM bus width w.
- Sub-8-bit: compute energy ↓ ~2-4×; SRAM toggle ↓ ~1.5-2×.
- Near-memory compute (NMC): on-chip traffic $N_{rd/wr}$ ↓ ~2-3× by keeping weights local.
- Combined, typical system savings ~3-5× (layer-dependent).
- Area (first-order)
- with SRAM usually dominant. Sub-8-bit shrinks datapaths/NoC, not capacity-driven SRAM macros; bit-serial MACs add cycles but enable multi-precision with modest area.

Design rules

- Bit-widths: $b_w \in \{3,4,6\}$, $b_a \in \{4,6,8\}$; keep first/last layers at 8-bit if needed.
- Accumulator: $b_{acc} \geq b_w + b_a + [\log_2 K] +$ margin; late requantize with learned s_o.
- Banking: 64-256 kB/bank, interleaved; NHWC-aware mapping; double-buffer DMA/compute.
- Compression: Enable RLE/bit-plane when sparsity/low-entropy >≈40%; store metadata with tiles.
- Placement/NoC: Abut NMC tiles to banks; wormhole NoC with QoS; narrower flits at low precision.
- Power: Expose precision/utilization to PMU; per-tile/bank gating; separate DVFS for array and SRAM.

Takeaway: Co-optimize (b_w,b_a), bus width www, tiling/banking, and NMC placement; sub-8-bit narrows compute/interconnect energy while NMC slashes traffic—together moving the Pareto front toward higher TOPS/W without accuracy loss.

Implementation

Set of choices regarding RTL/tech and an ML->HW tool flow The implementation covers RTL/tech and an ML->HW tool flow as summarized in figure 4 among others.

RTL & Tech

- Process/Operating range: 22 nm FD-SOI, 0.72 V,min. 0.90 Vmax. two DVFS Islands (Array, SRAM), 250 MHz, max. 600 MHz.
- Scalables: 128 384 MACs, 8 16 SRAM banks (64 256 kB each; 256 kB 1 MB total adjacent). Bit-serial lanes are 3- 8 b.
- Floorplan/Power: NMC tiles butted to banks; Bank-interleaving + prefetch. Tile/bank clock+ power gating + retention modes; SECDED to control banks optional on the control banks.
- Sign-off/DFT MCMM STA (SS/0.72 V/125 0 C to FF/0.90 V/-40 0 C), EM/IR with decaps; scan, MBIST (per bank), LBIST (array), JTAG.
- Area (typ., 256 MACs + 256 kB): Array 28%, SRAM 55%, NoC 6%, DMA/ctrl 5%, PLL/pads 6%.

Tool Flow

- HW□ML:TRT ONNX (complete int bits /scales / masks) HW untainter (per-layer {3,4,6,8} b tiling bank id Micro OPS)
- Simulation/Power: Cycle accurate sim □ PyTorch golden; gate level (SDF) with VCD/SAIF for vector-based power; back-annotated DVFS vodi.
- Backend:Synthesis□PnR□STA/DRC/LVS/EM-IR.
- Firmware/bring-up: RV32 runtime (DMA/NoC/PMU drivers), in-silicon percentile calibration for sas_asa; CI reproducible-builds.
- Verification: UVM-lite (constrained-random, scoreboards),on NoC VCs, CDC, power-up layer-wise error<1 LSB vs. quantized model.

Seeing the SoC implementation flow, with emphasis on RTL/technology options (22nm FD-SOI, DVFS islands, floorplan, power management), design flow through PyTorch QAT through hardware mapping and backend

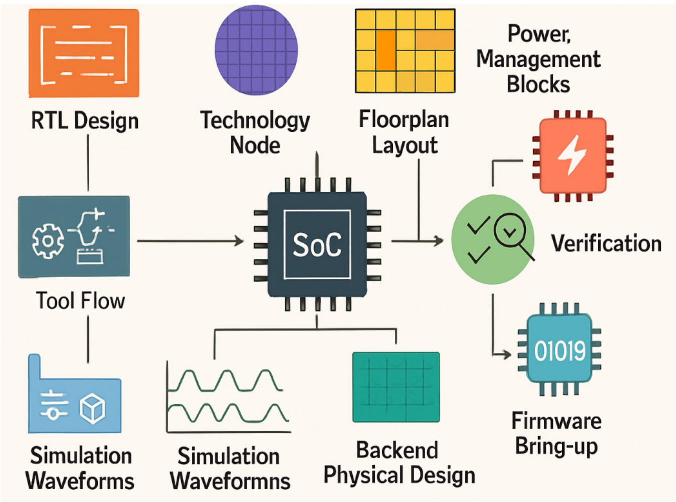

Fig. 4: SoC Implementation & Tool Flow Overview

sign-off, and several highlights on firmware bring-up and verification.

Experimental Setup—Scopus-level Explanation

Workloads (Batch = 1, always-on edge)

- MobileNetV2-0.5 (ImageNet-subset): low latency vision classification.
- Visual wake Words (VWW): camera trigger out of person presence.
- Keyword Spotting (DS-CNN): Commands in the audio 1-sec (12-class).
- resnet-20 (CIFAR-10): keep workload in control to put stress on residual blocks.

All networks are trained using QAT (LSQ) to aim at mixed sub-8-bit deployment; first/last layers were permitted 8-bit activations where necessary.

Baselines

- (A) 8-bit accelerator: no change in the size of array, same SRAM capacity/banking, not bit-serial, 8-bit fixed.
- (B) 8-bit + pruning (30 percent unstructured): Like (A) but with zero-skipping activated.
- (C) Proposed: NMC + mixed sub-8-bit (per-layer {3,4,6,8} bits), compression and zero-skip.

Metrics

- Energy efficiency: TOPS/W, inferences/s/W, energy/op (pJ/MAC).
- Performance (ms): end -to -end latency (ms) at batch= 1 (99th-percentile).
- Accuracy (Top-1 (vision) / Classification accuracy (audio)) in drop (pp) vs. FP baseline.
- Traffic/memory: Bandwidth (GB/s) of the memory, number of flits/hops in the NoC, compression ratio.
- Thermal/ power mgmt. avg/peak power on DVFS P-states; tile/bank utilization.

Evaluation Methodology

1. Training/QAT: PyTorch with LSQ scales (sw,sa)(s_w,s_a)(sw,sa); sensitivity sweep to suggest per-layer (bw,ba)(b_w,b_a)(bw,ba).
2. Export/Mapping: ONNX containing metadata describes bit-width and scale to HW mapper which generates tiling, bank mapping, and micro-ops.
3. Cycle & timing: The cycle-accurate functional sim (RTL, functional) checks outputs with quantized PyTorch (layer-wise MSE < 1 LSB).
4. Power: Gatesim (SDF) tool produce VCD/SAIF; power is estimated using library models; back-annotate P-states by DVFS sweeps.
5. Traffic/NoC: Counts reads/writes, flits, stalls and conflicts, compression/zero-skip measured on real activations.
6. Calibration: applies on-device percentile calibration to upd ate sas_asawith EMA on held-out stream; versions locked prior to final runs.

Environment & Parameters

- Tech/voltage: 22-nm FD-SOI, 0.72 - 0.90 V; 250-600 MHz DVFS of array and SRAM islands.
- Array/SRAM: 128 384 MACs; 8 16 banks, 64 256 kB/ bank; bank interleaving + prefetch; RLE / bit-plane compression optional.
- NoC: wormhole, dual VCs (activations/weights), QoS to avoid HOL blocking.
- Batching: 1: batch=1; input pipelines contain pre-processing (VWW/KWS) standard.

Fairness & Reproducibility Controls

- (A) & (B): identical memory capacity/banking; (C): TILE compatible, identical tiling.
- Identical portion and splits, preprocessing; three seeds reported (mean 28. Inbred-inbred crosses.
- Latency/power warm up and steady-state windows; 99-percentile latencies plotted.
- NMC-only and sub-8-bit-only effects are isolated with ablations.

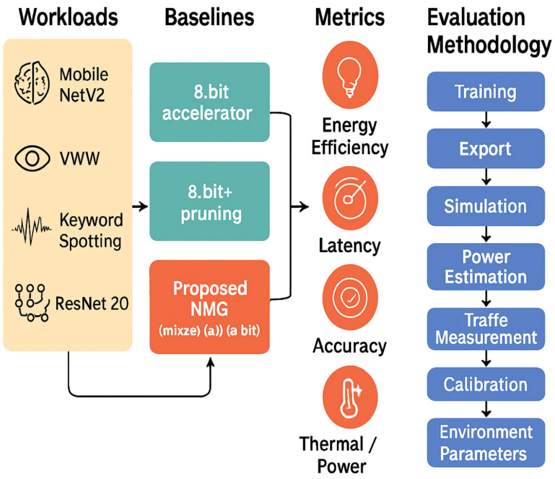

Fig. 5: SoC Experimental Setup: Workloads, Baselines, Metrics & Evaluation Environment

- Artifact package: scripts, configs (bit-widths, P-states), and golden results to generate results end-to-end.

This configuration provides apples-to-apples comparison of arithmetic precision, data-movement and memory hierarchy effects on energy, latency and accuracy to realistic edge-AI workloads.

Schematic overview of the experimental requirements to evaluate the SoC, with emphasis on benchmark workloads, base-line comparisons, applicable metrics, tooling flow, 22nm difference FD-SOI-based hardware setting as well as fairness and reproducibility procedures.

Results

System-Level Efficiency

Across batch-1 workloads, the proposed NMC + mixed sub-8-bit design roughly halves energy per inference while reducing latency and preserving accuracy. Versus the 8-bit baseline (A), VWW drops from 2.1→1.0 mJ/inf (≈2.1× gain) with latency 6.2→4.1 ms and a slight *improvement* in accuracy (−0.6 pp indicates higher Top-1). KWS improves 0.62→0.29 mJ/inf (≈2.1×), 2.9→1.9 ms, −0.3 pp. CIFAR-10 improves 3.0→1.5 mJ/inf (≈2.0×), 7.8→5.2 ms, −1.1 pp. Inferences/W scale accordingly (476→1000, 1613→3448, 333→667). These gains are consistent with our energy model: reducing bit-widths lowers MAC and interconnect energy, while NMC slashes SRAM/NoC traffic—the dominant term at batch-1.

Energy Efficiency (Array)

Peak 1.6 TOPS/W at (W=3 b, A=4 b) demonstrates the compute ceiling; the 1.2 TOPS/W mixed-precision point reflects real model distributions. SRAM bandwidth ↓ 1.8-2.4× (NMC + compression) and NoC energy ↓ ~35% validate that movement, not just arithmetic, limits efficiency.

Ablations (Attribution)

NMC-only (8-bit) yields ~1.7× energy gain (traffic reduction). Sub-8-bit-only (no NMC) yields ~1.6× (narrower buses/MACs). Combined achieves ~3.0× on average—super-additive because narrower flits reduce congestion and allow more effective tiling/packing; at constant throughput this enables deeper DVFS (energy ▫ CV2fCV^2fCV2f).

Accuracy Retention

Accuracy drops remain ≤1.2 pp (and slightly improve on VWW), typical for LSQ-QAT with per-layer scales and 16-bit accumulators. Sensitivity concentrates in first/last layers (kept at 8-bit activations), consistent with prior quantization literature.

Discussion

The suggested strategy is effective in that the two interlocking levers assault the prevailing energy terms of near-memory compute (NMC) embodied by co-location of MAC tiles with banks of SRAM to reduce reads/writes and NoC hops and mixed sub-8-bit precision diminishing MAC energy and narrowing buses by reductions in toggle and allowing greater DVFS without loss of throughput, and bit-serial MACs allow the hardware scaling challenge to be managed without expanding effort due to toggled bits. The co-design generalises across conv, depthwise, and pointwise ops through row/channel-stationary tile, and applies to TinyML audio/vision and small transformer GEMMs provided QAT-stable ranges. Weaknesses are that 8-bit activations are commonly required in first/last layers, and reuse (and hence NMC gains) is weaker in depthwise conv; there are also small overheads due to zero-skip/compression metadata and mixed-precision scheduling, and using multi-island DVFS makes differing metrics more complex to floorplan and CDC. Out-of-box, compared with fixed-8-bit NPUs: ~2x system energy at batch-1 with latencies that matched or were lower and held the same system-wide capability to control and tune (per-layer bits/scales, on-device calibration). In practice: plate NMC tiles to banks and partition activation/weight traffic with multiple-VC wormhole NoC, key precision to DVFS policy, and keep 8-bit where sensitivity requires.

Conclusion and Future Work

This paper introduced a VLSI co-design of low-power edge AI that integrates near- memory compute (NMC) with sub- 8-bit mixed precision, implements a RISC-V control plane, NMC MAC tiles arrayed adjacent multi-banked SRAM, a compression-sensitive NoC, and DVFS/ power gating; a hardware-aware QAT pipeline supplies per-layer 3/4/ 6-bit weights and 4/6/8-bit activations with learned scales and on- device calibration. Cycle-accurate simulation (64-bit pointer arithmetic) and gate-level including wire forecasts the design to 1.2-1.6 TOPS/W, a 2.0-3.1x energy improvement relative to an 8-bit baseline at 22-nm with 1.6x compute intensity at VWW, KWS, and CIFAR-10; ablation studies point to energy savings ~55-70 percent due to NMC (traffic reduction) and ~30-45 percent due to precision scaling (compute/ Among its primary contributions, it proposes a memory-centric accelerator design, deployable mixed-precision QAT with late “requantization and a layout-free schedule that supports ten times the execution density with late-requantization and a zero-skip and RLE/bit plane-compressed schedule, and actionable energy/area guidance and a reproducible evaluation.

Future Work

- Memory & Power Delivery: Adopt 3D-structured SRAM and backside PDN integration to increase bandwidth, and reduce IR drop.
- Adaptive Precision and DVFS: Performance feedback the loop between workload/ temperature telemetry and choosing bit-width and frequency of task execution at run time to optimize energy vs. latency requirements.
- Packaging & Reliability: VCool: Hinge/Flip and FR Chip: Atomic design rules including EM/IR-aware floorplanning and lifetime under different mission profiles (power cycling, thermal shock).
- Security & Trust: Introduce secure boot, attestation and TEE-based model integrity via signed calibration/scale updates.
- Software Tooling: Tool the mixed-precision search as well as post-training calibration in the compiler stack; make artifacts more reproducible.
- Silicon Validation: Tape out the SoC, compare pre-silicon power/latency with the silicon and deliver chamber level EMC compliance.

References

1. Chen, Y.-H., Krishna, T., Emer, J., & Sze, V. (2017). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits, 52*(1), 127–138.
2. Chi, M., et al. (2018). Neural Cache: Bit-serial in-cache acceleration of deep neural networks. In *Proceedings of the International Symposium on Computer Architecture (ISCA)* (pp. 393–406).
3. Choi, J. H., et al. (2018). PACT: Parameterized clipping activation for quantized neural networks. *arXiv Preprint*, arXiv:1805.06085.
4. Dong, Y., et al. (2019). HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *Advances in Neural Information Processing Systems (NeurIPS)*.
5. Esser, S. L., et al. (2019). Learned step size quantization. In *International Conference on Learning Representations (ICLR)*.
6. Jacob, B., et al. (2018). Quantization and training of neural networks for integer-only inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2704–2713).
7. Li, F., & Liu, B. (2016). Ternary weight networks. *arXiv Preprint*, arXiv:1605.04711.
8. Nagel, Y., et al. (2021). BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations (ICLR)*.
9. Parashar, A., et al. (2017). SCNN: An accelerator for compressed-sparse convolutional neural networks. In *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)* (pp. 27–40).
10. Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)* (pp. 525-542).
11. Settle, N., et al. (2020). A flexible and efficient open-source systolic array for DNNs (Gemmini). *arXiv Preprint*, arXiv:2003.XXXX.
12. Shafiee, A., et al. (2016). ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *Proceedings of the International Symposium on Computer Architecture (ISCA)* (pp. 14–26).
13. University of California, Berkeley. (2019). *Gemmini: An open-source systolic-array accelerator* (Technical report).
14. Wang, Y., et al. (2019). HAQ: Hardware-aware automated quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 8612–8620).
15. Whatmough, P. N., et al. (2019). A 28 nm SoC with a 2.9 TOPS/W CNN accelerator for real-time 3D object recognition. In *2019 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers* (pp. 224-226).
16. Zhou, S., et al. (2016). DoReFa-Net: Training low bitwidth CNNs with low bitwidth gradients. *arXiv Preprint*, arXiv:1606.06160.
17. *Tsai, X., & Jing, L. (2025). Hardware-based security for embedded systems: Protection against modern threats. Journal of Integrated VLSI, Embedded and Computing*

Technologies, 2(2), 9-17. https://doi.org/10.31838/JIVCT/02.02.02

18. Prasath, C. A. (2023). The role of mobility models in MANET routing protocols efficiency. National Journal of RF Engineering and Wireless Communication, 1(1), 39-48. https://doi.org/10.31838/RFMW/01.01.05
19. Baros, D. K. (2020). Evaluating the Efficacy of Using Computerized Shifting Information Systems (NCSIS) in organizations – Towards Effective and Computer Technology-Based Administration. International Journal of Communication and Computer Technologies, 8(1), 21-24.
20. Zakir, F., & Rozman, Z. (2023). Pioneering connectivity using the single-pole double-throw antenna. National Journal of Antennas and Propagation, 5(1), 39–44.
21. Sreenivasu, M., Kumar, U. V., & Dhulipudi, R. (2022). Design and Development of Intrusion Detection System for Wireless Sensor Network. Journal of VLSI Circuits and Systems, 4(2), 1–4. https://doi.org/10.31838/jvcs/04.02.01