7.4 A 4nm 6163-TOPS/W/b 4790-TOPS/mm²/b SRAM Based Digital-Computing-in-Memory Macro Supporting Bit-Width Flexibility and Simultaneous MAC and Weight Update

Haruki Mori[1], Wei-Chang Zhao[1], Cheng-En Lee[1], Chia-Fu Lee[1], Yu-Hao Hsu[1], Chao-Kai Chuang[1], Takeshi Hashizume[2], Hao-Chun Tung[1], Yao-Yi Liu[1], Shin-Rung Wu[1], Kerem Akarvardar[3], Tan-Li Chou[1], Hidehiro Fujiwara[1], Yih Wang[1], Yu-Der Chih[1], Yen-Huei Chen[1], Hung-Jen Liao[1], Tsung-Yung Jonathan Chang[1]

[1]TSMC, Hsinchu, Taiwan
[2]TSMC, Yokohama, Japan
[3]TSMC, San Jose, CA

The computational load, for accurate AI workloads, is moving from large server clusters to edge devices; thus enabling richer and more personalized AI applications. Compute-in-memory (CIM) is beneficial for edge-AI workloads, specifically ones that are MAC-intensive. However, realizing better power-performance-area (PPA) and high accuracy is a major challenge for practical CIM implementation. Recent work examined tradeoffs between MAC throughput, energy efficiency and accuracy for analog based CIM [1-3]. On the other hand, digital-CIMs (DCIM), which use small, distributed SRAM banks and a customized MAC unit, have demonstrated massively-parallel computation with no accuracy loss and a higher PPA with technology scaling [4]. In this paper, we introduce a 4-nm SRAM-based DCIM macro that handles variable 8/12b-inteteger weights and 8/12/16b-integer inputs in a single macro. The proposed 8-transistor 2b OAI (or-and-invert) cell achieves a 11% smaller combined bit cell and multiplier area, and supports ultra-low voltage operation, down to 0.32V. Furthermore, the signed-extended carry-look-ahead adder (signed-CLA) and an adder tree pipeline are introduced to boost throughput.

Figure 7.4.1 shows the implementation of the bit cell structure, and a neural network accuracy comparison with various bit precisions. Since we targeted concurrent write and MAC operations, ping-pong for weight updates and MAC operations, array needs to have an even number of rows: a classical approach is to use two 12T bit cells and a 2-input NOR. The 12T cell supports simultaneous read and write operations, as its read- and write-port are independent. The 2-input NOR is used for bitwise multiplication with input activations (XIN) and weights (W). On the other hand, two 8T cells and an OAI is used in the proposed SRAM-based DCIM macro. In the proposed bitcell topology, the 8T bitcells act as memory data storage and row selection for the write operation. The OAI performs row selection and bitwise multiplication for the MAC operation. The signals for row selection and multiplication are generated by the logic in read WL driver (RWLDRV), and are propagated to the OAIs as RWLBs. Compared to the two 12T cell and NOR2, the area required for data storge and bitwise multiplication is 11% smaller. In addition, three signal tracks in the vertical direction is reduced by adopting proposed bitcell topology, because the OAI logic removes two RWLs and XINB. We also compared the network accuracy with various integer bit precisions, INT16, INT12 and INT8, for different workloads: MobileNet_v2 & ResNet-50, pre-trained with ImageNet. Quantization and de-quantization are introduced prior to and after the compute convolution and fully-connected (CONV/FC) layer, the MAC is performed with integer format in the DCIM. During evaluation the accuracy difference between a MAC operation using FP16 and a MAC operation using INT16 is around 0.2%. We also observed that INT12 results in a negligible accuracy loss, < 0.1%, compared to INT16. INT8 yields a 1~2% accuracy loss, but achieves a higher energy efficiency. The energy efficiency is 1.8x using INT12 and 4x for INT8, compared to INT16. To support higher-accuracy and higher energy-efficiency flexibility, we implement flexible bit width support in our DCIM macro design.

Figure 7.4.2 shows an overview of the DCIM architecture implemented in a 4-nm technology. DCIM supports signed 8/12/16b XIN and signed 8/12b W for the MAC, with 144-activations and 16-output channels. The weights are preloaded before MAC operation through a write port; like a write in a normal SRAM. The signed 8/12b weights are supported by combining 3 local-MAC arrays (LMAC-A to C) and one global-IO (GIO). Each LMAC includes 144 bitwise multipliers, 5 to 12b adders in the adder tree and a 12b partial-sum (PSUM) result output. Like existing CIM designs, interface for XIN signals is a bit serial basis (one bit per cycle). As previously mentioned, XINs are propagated to RWLBs after being combined with the address decode signal (CIMA). To support 8/12b weights, we need to consider weather each MAC is singed or unsigned. LMAC-A for W bits 0 to 3 (LSBs) is unsigned, while LMAC-C for W bits 8 to 11 (MSBs) is signed. On the other hand, LMAC-B for bits 4 to 7 needs to support signed and unsigned operations: signed for 8b and unsigned for 12b weights. The GIO includes 16 and 20b adders, a shift function, and a 36b accumulator. The external input signal, WWIDTH, specifies the current W width: 0 for 8b and 1 for 12b weights. LMAC-B adder tree is switched between signed or unsigned operation based on the WWIDTH signal. In addition, LMAC-C is disabled, to save power, in 8b mode. The INWIDTH[1:0] bus controls the XIN width: 00 for 8b, 01 for 12b, and 10 for 16b modes. To support a signed format with width flexibility, the first 4 cycles are signed 4b operations and the rest of the cycles are unsigned 4b operations, regardless of INWIDTH.

Figure 7.4.3 describes the overall layout floorplan, the transistor threshold voltage (V_t) selection, and the adder tree pipeline. In the DCIM, the MAC array uses a mixed-V_t design to balance performance and leakage; We checked device count and gate delay in the local adders and global adders to place mixed-Vt devices in adder tree. Due to the tree structure of adder, the SRAM bit cell array and adders in the early stage of the adder tree occupies >75% of total device in adder tree. Therefore, we use high-V_t devices for the SRAM array and local adders to reduce leakage. On the other hand, low-V_t devices are used in other portions of adder: semi-global and global adders, shifter, and accumulator. Adopting a mixed-V_t design results in 36.7% of the delay due to high-V_t devices and 63.3% due to low-V_t devices in the first pipeline stage. Leakage is evenly split between high-V_t (49.5%) and low-V_t (50.5%) devices; MAC array leakage is 38.6% smaller than a purely low-V_t design. An adder tree pipeline is introduced for better MAC throughput. A high-V_t device is slow at low voltage, due to smaller overdrive voltage ($V_{gs} - V_t$), and a signal routing with a long wire is slower at high voltage due to the wire delay caused by wire resistance. To balance these effects, the first pipeline stage includes shorter gate stages in adder tree with mixed-V_t and long wire, while the second pipeline stage includes longer gate stages in adder tree with low-V_t and short wire. A static-timing analysis (STA) is used to check the path delays and the maximum operating frequency (f_{MAX}) difference between first and second pipeline stages. As shown by the f_{MAX} vs voltage graph, f_{MAX} between first and second pipeline stages are well balanced. First pipeline stage is 2% faster at 0.9V because of shorter gate stages, while second pipeline stage is 4% faster at 0.5V because of the low-V_t devices. STA results confirm an f_{MAX} with SSG and –40°C is 0.41 and 1.49GHz at 0.5 and 0.9V.

Figure 7.4.4 shows the bit cell layout optimization and circuit optimizations for the final accumulator. The total area of two 8T cells and OAI is 0.161µm². The DCIM design requires more signal tracks in the horizontal direction if the number of XINs and/or the width of W is increased, because the adder tree becomes deeper and wider. To mitigate routing congestion a triple-cell-height style is used for the proposed bit cell. Due to shorter wire routing in the horizontal direction, the wire delay and dynamic power is reduced. The signed-CLA is introduced to speed up the 36b accumulator. Because of the DCIM's bit-serial MAC operation, the final accumulator accumulates two partial sums every cycle; a partial sum from adder tree (PSUM[19:0]) and a partial sum from the previous cycle after one-bit-shift-up operation (NOUT[34:0], signed). Because of the width difference between two partial sums, sign-bit of PSUM[19:0] is extended 16b in the 36b accumulator. As sign extension consists of just coping PSUM[19], the CLA logic is simplified to speed up the 16-b carry-chain critical path for the MSB side. The STA results, in the SSG, 0.9V, and -40°C corner, show that the optimized signed-CLA is 9.6% faster than a conventional CLA; the required area is also smaller, the area overhead with signed-CLA is 0.9% in a macro compared to a ripple-carry adder (RCA).

Figure 7.4.7 shows a 4nm FinFET test chip micrograph. The DCIM macro area is 0.0172mm²; there are three macros on the chip. Figure 7.4.5 summarizes measurement results. the left-hand side graph shows that we observed a minimum operating voltage (V_{MIN}) of 0.32V at –40°C, with a 95% yield. The right-hand side graph shows the energy efficiency (TOPS/W vs voltage) for W=12b and XIN=12b. The weight activation ratio is 50% and different bit sparsities, 12.5%, 25% and 50%, for XIN are considered. The TOPS/W increases when the sparsity and/or voltage is low. A 12.5% results in 42.4TOPS/W at 0.5V and 12.1TOPS/W at 0.9V with 12b W and 12b XIN. Figure 7.4.6 shows a comparison table to prior work. Our DCIM demonstrates a higher TOPS/mm² and TOPS/W; normalized per bit, we achieve 4790TOPS/mm²/b and 6163TOPS/W/b. The flexible weight and input bit width features for a single macro provides the option for more accurate MAC operations with a wider width (42.2TOPS/W with W=12b and XIN=12b) or a higher energy efficiency using narrower width (87.4TOPS/W with W=8b and XIN=8b).

References:

[1] S. Yin et al., "PIMCA: A 3.4-Mb Programmable In-Memory Computing Accelerator in 28nm for On-Chip DNN Inference," *IEEE VLSI*, pp. 1-2, 2021.
[2] H. Wang et al., "A 32.2 TOPS/W SRAM Compute-in-Memory Macro Employing a Linear 8-bit C-2C Ladder for Charge Domain Computation in 22nm for Edge Inference," *IEEE VLSI*, pp. 36-37, 2022.
[3] J.-W. Su et al., "A 28nm 384kb 6T-SRAM Computation-in-Memory Macro with 8b Precision for AI Edge Chips," *ISSCC*, pp. 250-251, 2021.
[4] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm² Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," *ISSCC*, pp. 186-187, 2022.

2023 IEEE International Solid-State Circuits Conference (ISSCC) | 978-1-6654-9016-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISSCC42615.2023.10067555

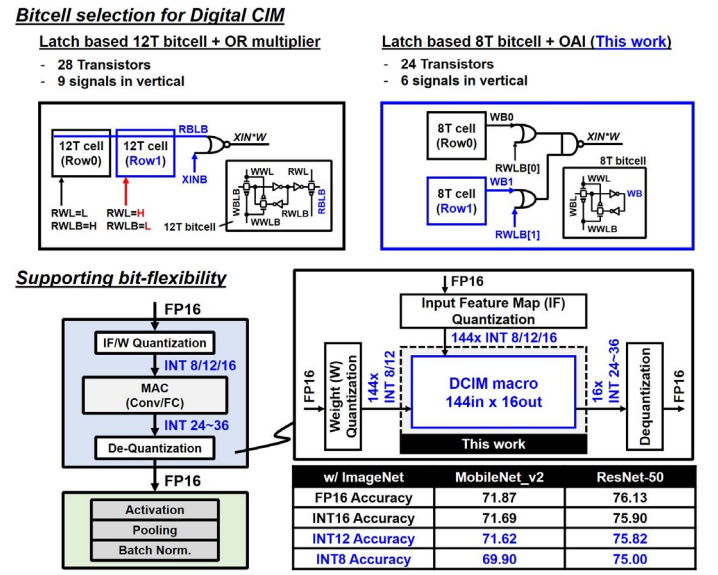


Figure 7.4.1: Bit cell selection for digital-CIM and neural network accuracy with various bit precisions.

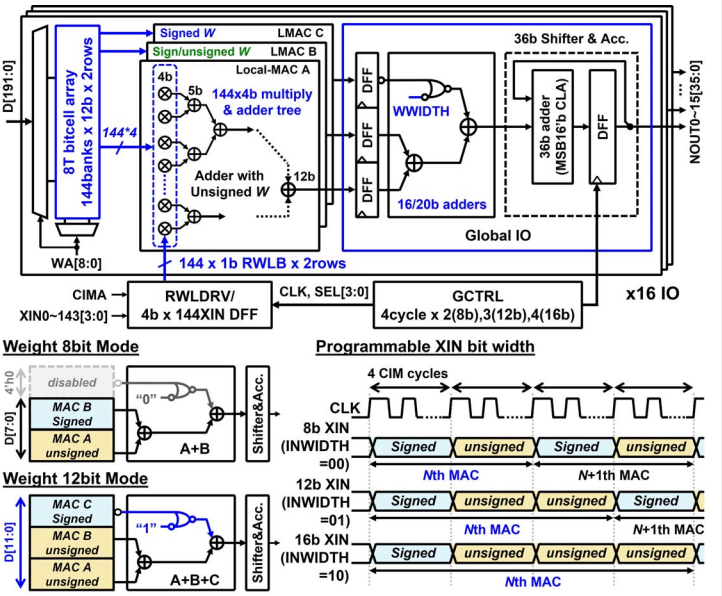


Figure 7.4.2: Overview of our DCIM architecture and flexible width support.

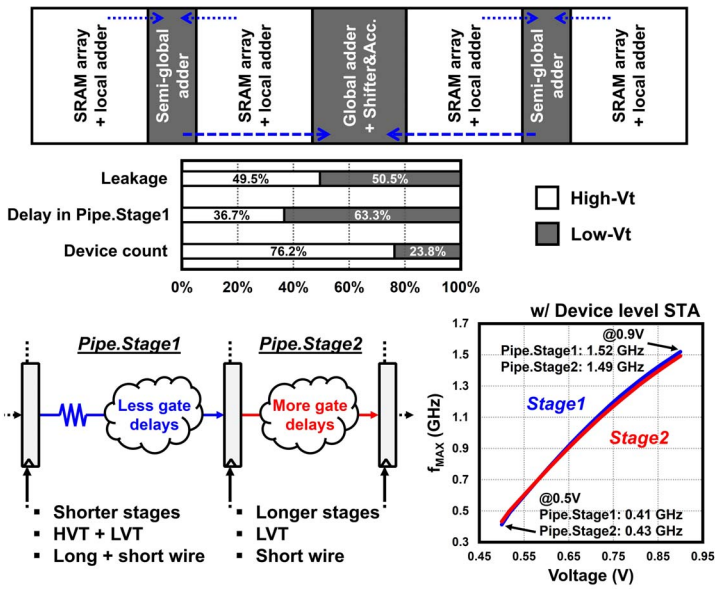
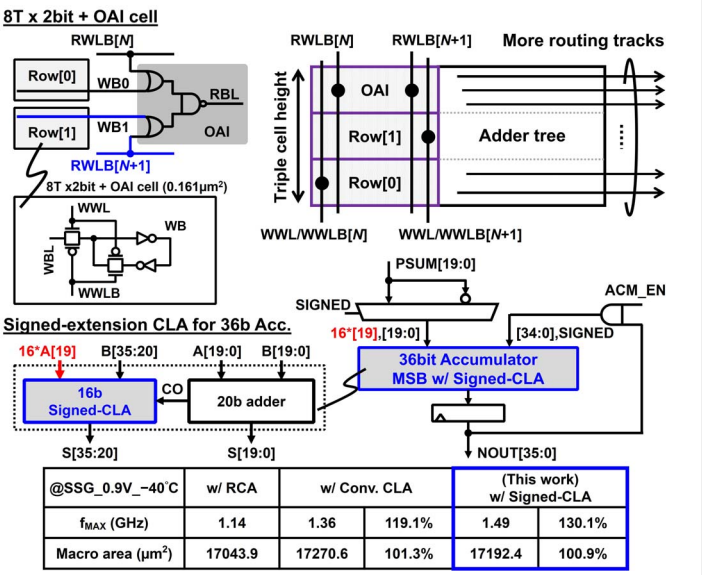


Figure 7.4.3: Layout floorplan, V_t selection and adder tree pipeline.



@SSG_0.9V_−40°C	w/ RCA	w/ Conv. CLA		(This work) w/ Signed-CLA	
f_{MAX} (GHz)	1.14	1.36	119.1%	1.49	130.1%
Macro area (μm²)	17043.9	17270.6	101.3%	17192.4	100.9%

Figure 7.4.4: Layout and circuit optimization for bit cell and final accumulator schematic.

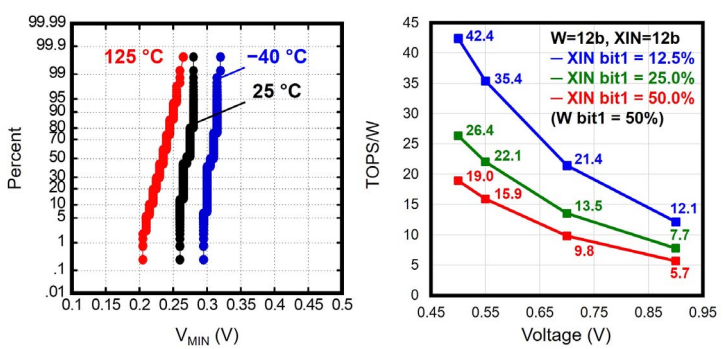
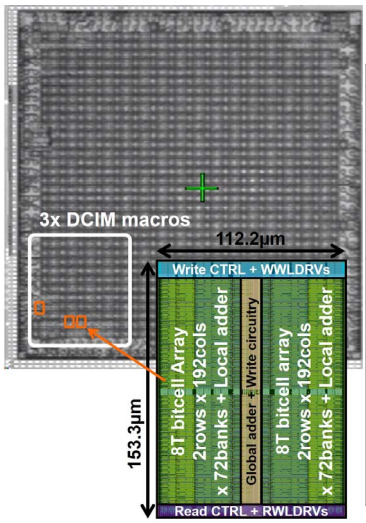


Figure 7.4.5: Measurement results: V_{MIN} and TOPS/W.

	VLSI'21 [1]	VLSI'22 [2]	ISSCC'21 [3]	ISSCC'22 [4]	This work
Technology	28 nm	22 nm	28 nm	5 nm	4 nm
MAC operation	Analog CIM (charge inject)	Analog CIM (charge share)	Analog + Digital CIM (charge share +digital adder)	Digital CIM	Digital CIM
Array size	32 Kb	128 Kb	384 Kb	64 Kb	54 Kb
Bitcell type	10T1C	9T	1R1W 6T	1R1W 12T	8T x 2bit + OAI
Macro area	N/A	0.25 mm²	N/A	0.0133 mm²	0.0172 mm²
Voltage	1.0V	0.6V ~ 1.0V	0.7V ~ 0.9V	0.5V ~ 0.9V	0.32V ~ 1.1V
Input Ch	256	64	16	64	144
Output Ch	128	16	16	64	16
Input bits	1 / 2	8	4 / 8	4	8 / 12 / 16
Weight bits	1 / 2	8	4 / 8	4	8 / 12
Output bits	4	8	12 / 20	14	24 ~ 36
Simultaneous MAC + write	No	No	No	Yes	Yes
TOPS/mm²	N/A	4.0 (8b)	N/A	221 (4b)	49.9 (W8b-XIN8b*) 33.3 (W12b-XIN12b*)
TOPS/mm²/bit	N/A	256 (1b)	N/A	3535 (1b)	4790.2 (1b*)
TOPS/W	588 (1b)	32.2 (8b)	94.3 (4b) 22.8 (8b)	254 (4b)	87.4 (W8b-XIN8b*) 64.1 (W12b-XIN8b*) 42.4 (W12b-XIN12b*) 32.1 (W12b-XIN16b*)
TOPS/W/bit	588 (1b)	2060.8 (1b)	1508.8 (1b)	4064 (1b)	6163.2 (1b*)

(*) TOPS/mm² at 0.9V and TOPS/W at 0.5V

Figure 7.4.6: PPA summary table and comparison to prior work.



Technology	4nm HKMG FinFET
Bitcell	Low voltage 8T x 2bit + OAI ($0.161\mu m^2$)
Macro size	112.2 µm x 153.3 µm
Memory config.	54 Kb 288 addresses x 192 cols w/ MUX1
MAC input size	144ch x 8b/12b/16b Signed Input 8b/12b Signed Weight 16ch x 24b~36b Output
TOPS/mm^2	13.7 ~ 49.9 (W8b–XIN8b) 9.2 ~ 33.3 (W12b–XIN12b)
TOPS/W(*)	W8b–XIN8b: 24.9~87.4 (XIN bit1 = 12.5%) 12.3~41.3 (XIN bit1 = 50%) W12b–XIN8b: 18.3~64.1 (XIN bit1 = 12.5%) 9.0~30.3 (XIN bit1 = 50%) W12b–XIN12b: 12.1~42.4 (XIN bit1 = 12.5%) 5.7~19.0 (XIN bit1 = 50%) W12b–XIN16b: 9.1~32.1 (XIN bit1 = 12.5%) 4.3~14.3 (XIN bit1 = 50%)

(*) W bit1 = 50%

Figure 7.4.7: Test chip micrograph and design summary table.

978-1-6654-9016-0/23/$31.00 ©2023 IEEE