D6CIM: 60.4-TOPS/W, 1.46-TOPS/mm^2, 1005-Kb/mm^2 Digital 6T-SRAM-Based Compute-in-Memory Macro Supporting 1-to-8b Fixed-Point Arithmetic in 28-nm CMOS

Jonghyun Oh
Dept. of Electrical Engineering
Columbia University
New York, USA
fodfod6565@gmail.com

Chuan-Tung Lin
Dept. of Electrical Engineering
Columbia University
New York, USA
cl4030@columbia.edu

Mingoo Seok
Dept. of Electrical Engineering
Columbia University
New York, USA
ms4415@columbia.edu

ESSCIRC 2023- IEEE 49th European Solid State Circuits Conference (ESSCIRC) | 979-8-3503-0420-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ESSCIRC59616.2023.10268725

***Abstract*— This paper presents a digital 6T-SRAM-based compute-in-memory macro named D6CIM, which can support 1-to-8b fixed-point arithmetic. Based on the time-sharing (reuse) architecture, D6CIM is designed with three new techniques: static dual-wordline access, hybrid compressor adder tree, and bit-first accumulation. D6CIM is prototyped in 28-nm CMOS. The measurement results show that D6CIM advances the prior arts in the product of the three key metrics: energy efficiency, weight density, and compute density.**

Keywords— Compute-in-memory, in-memory-computing, SRAM, deep convolutional neural networks, digital

I. Introduction

SRAM-based compute-in-memory (CIM) hardware has demonstrated orders-of-magnitude improvement in energy efficiency and throughput for vector-matrix multiplications (VMM). Recently, its digital version, often called DCIM, has received large attention for its superior robustness, precision, and scalability over analog-mixed-signal counterparts [1-6]. However, existing DCIMs exhibit lower weight density (Kb/mm^2) because they employ a large amount of arithmetic hardware. Time-sharing/reusing arithmetic hardware across inputs and weights can improve weight density but naturally degrades the compute density (TOPS/mm^2) [6]. Also, it does not improve energy efficiency (TOPS/W) since the amount of capacitive charging and discharging remains the same for a given computation.

This paper proposes a new digital 6T-SRAM-based CIM macro, called D6CIM, supporting 1-to-8-bit fixed-point arithmetic. We aim to improve energy efficiency and compute density while achieving state-of-the-art weight density. For this goal, we propose the static dual-wordline access scheme, which can access two bitcells in each sub-column, one via a local bitline (LBL) and the other via LBLb. Unlike the prior dynamic dual-wordline scheme [7], our scheme performs no precharge and uses low-swing signals on LBLs, reducing energy consumption. Second, we propose the hybrid compressor adder-tree (HCA) scheme. Compared to the prior works employing only adder trees [1-3,5,6], the proposed scheme substantially reduces the number of transistors, improving energy efficiency and weight density. Third, we propose the bit-first accumulation (BFA) scheme. Existing works would accumulate partial products across inputs and then across input bits, incurring frequent switchings on wordlines (WL) and LBLs. The proposed scheme accumulates across bits from the most-significant bit (MSB) to the least-significant bit (LSB) of a group of inputs and then the next group of inputs. It results in much less switching on WLs and LBLs.

D6CIM is prototyped in 28-nm CMOS. The measurement results show that at 0.6V (1.1V), it achieves an energy efficiency of 60.4 TOPS/W (22.4 TOPS/W), a compute density of 0.12 TOPS/mm^2 (1.46 TOPS/mm^2), and a weight density of 1005 Kb/mm^2. Compared to [6], which achieves the best normalized weight density of 1067 Kb/mm^2, D6CIM marks 8.6X better compute density and 2.2X better energy efficiency at the roughly same weight density.

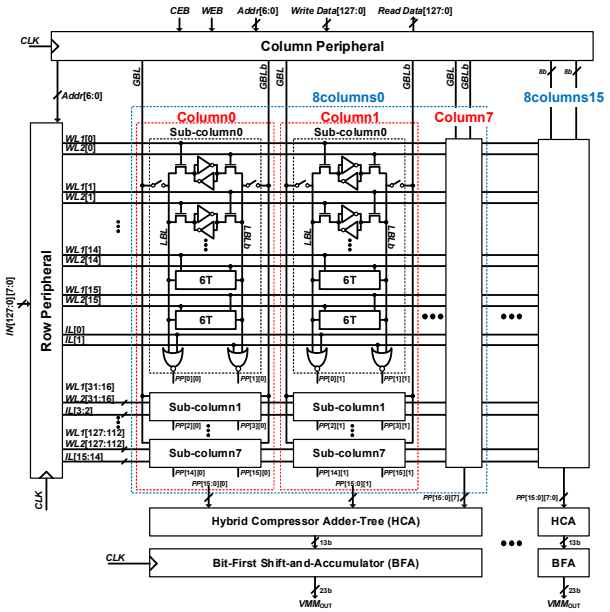

Fig. 1. D6CIM microarchitecture.

II. Proposed Microarchitecture and Circuits

A. Microarchitecture

Fig. 1 shows the microarchitecture of D6CIM. It has a 128×128 6T SRAM array. We target 8b weights; thus, the array can store 128×16 weights. Each column contains eight sub-columns, each having 16 6T bitcells sharing a pair of LBL and LBLb. The LBLs are connected to a pair of global bitlines (GBL) via switches. Each sub-column also contains two NOR gates, each serving as a 1b multiplier. The even-numbered bitcells have WL1 (WL2) for the left (right) access transistors; the odd-numbered bitcells have WL2 (WL1) for the left (right) access transistors. Every *eight* columns (8columns) share one HCA, followed by the BFA. The macro employs 16 HCAs and 16 BFAs. It also contains the row peripheral, which controls WLs and input lines (ILs) for VMM operation; and WLs for SRAM Read/Write (R/W) operation; It contains the column peripheral, which controls GBLs for SRAM R/W.

D6CIM performs an 8b 128×16d (dimension) VMM in 64 clock cycles. It first activates two consecutive WL1s in each sub-column, which transfers two weight bits, via LBL and LBLb, to the two NOR gates in that sub-column. At the same time, the row peripheral feeds the corresponding two input activation bits via ILs to the NOR gates. Since each column has eight sub-columns, one 8columns (64 sub-columns) generates a total of 16 8-b partial products (PP[15:0][7:0]).

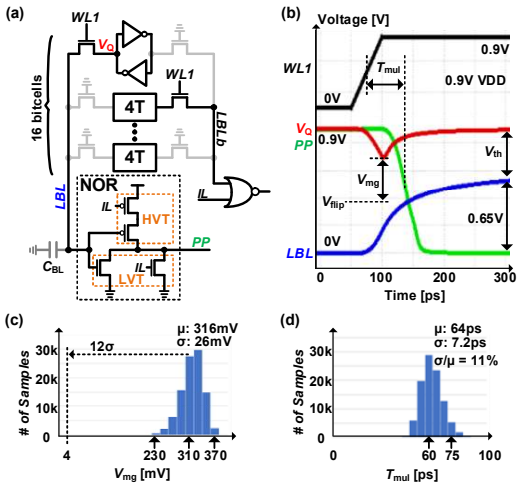

Fig. 2. (a) The proposed static dual-wordline access scheme and (b) its timing diagram. Monte-Carlo simulation results of (c) V_{mg} and (d) T_{mul}.

The HCA then adds up 16 partial products and produces partial sums, and the BFA performs shift-and-accumulate the partial sums. We repeat this process eight times while feeding the rest of the input bits in the bit-serial fashion and then again eight times for providing the rest of the inputs corresponding to the weights in each sub-column. Finally, D6CIM produces the VMM result (VMM_{OUT}), a 23b 16d vector.

B. *Static Dual-Wordline Access*

As shown in Fig. 2(a), D6CIM *statically* accesses two consecutive bitcells in each sub-column using LBL and LBLb. Compared to the conventional dynamic dual-wordline access technique [7], the proposed method eliminates the precharging operation, significantly reducing switching activities on LBLs. Also, it uses low swing signals on LBLs via the threshold voltage (V_{th}) drop of the access transistor. Our simulation shows this technique significantly reduces access energy consumption.

The proposed technique, however, may increase the chance of read-upset since previous access can discharge LBLs to 0V. However, we found that the probability is meager for two reasons. First, the access is single-ended, which exhibits more robustness to read upset than the conventional differential access. Second, each bitcell sees short LBL, shared by only 16 bitcells, exhibiting small parasitic capacitance. To evaluate the read-upset risk, we performed Monte-Carlo simulations on the upset margin (V_{mg}), which is defined as $V_{mg}=V_Q-V_{flip}$, where V_Q is the bitcell's storage voltage, and V_{flip} is the V_Q that upsets the bitcell (Fig. 2(b)). The simulation shows that V_{mg} remains positive even under the 12-σ worst case (Fig. 2(c)), confirming the robustness of the proposed technique.

Recall that the static access technique reduces the voltage swing on LBLs to VDD-V_{th}. This could degrade the noise margin and increase multiplication delay (T_{mul}, defined in Fig. 2(b)). To address these problems, we skew the PN ratio of the NOR gates using high and low V_{th} (HVT, LVT) devices. We also performed 100-k Monte-Carlo simulations, confirming all correct functionality and a small variability of T_{mul} (σ/μ=11%, Fig. 2(d)).

C. *Hybrid Compressor Adder-Tree Scheme*

We propose the HCA hardware. One 8columns in D6CIM produces 16 8b partial products (PP[15:0][7:0]). To sum up all PPs, existing works employ a 16-input 8b adder tree, which incurs a significant area overhead [1]. The proposed HCA can

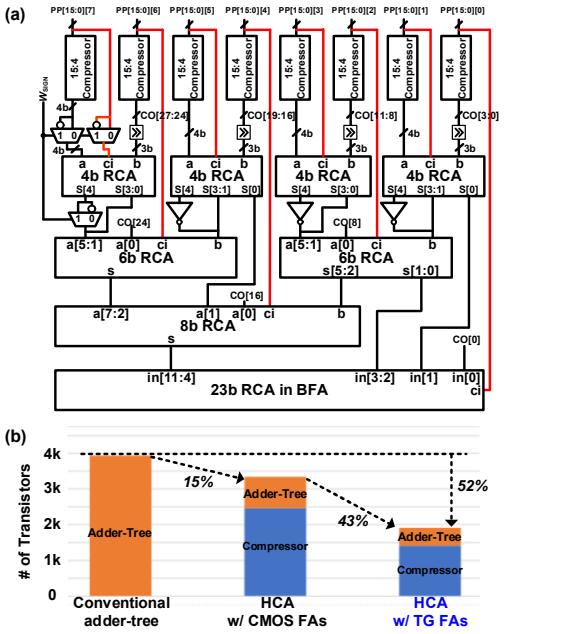

Fig. 3. (a) Hybrid compressor adder-tree (HCA). (b) The proposed techniques (HCA and TG-based FAs) reduce transistor count by 52% as compared to the conventional adder-tree-only scheme using CMOS FA cells.

support the same operation but requires fewer transistors, improving weight density and energy efficiency.

Fig. 3(a) shows the detailed microarchitecture of the HCA. One HCA is shared by one 8columns (the macro has a total of 16 HCAs). It has eight 15:4 compressors followed by one 4b 8-input adder tree. Note that one 8columns produces 8×16b data, and eight compressors can process only 8×15b data. Instead of adding another compressor to deal with the remaining 8b, we use the carry-in port of the 4b ripple carry adders (RCA) in the adder tree (see the red lines in Fig. 3(a)). However, we have only four 4b RCAs and, thus, four carry-in ports. Therefore, we also use the carry-in ports of the two 6b RCAs, one 8b RCA in HCA, and the 23b RCA in the BFA (Sec. II.D). Also, we found that one input of the 4b RCA can be 3b due to the right-shift operation needed for alignment. This reduces the size of the 4b RCAs. It reduces the number of transistors by 15% compared to the conventional adder-tree-only scheme (Fig. 3(b)). Also, the circuit-level optimization on full adder (FA) cells (to be discussed in Sec. II.E) reduces the number of transistors by another 43%.

D. *Bit-First Accumulation Hardware*

We propose the bit-first accumulation (BFA) hardware. Fig. 4 shows the microarchitecture of BFA. It has a 23b RCA, a 30b register, and bi-directional shifters. It performs shift-and-accumulate on the output of HCA (partial sums) based on the proposed BFA scheme.

As shown in Fig. 5(a), the conventional scheme would accumulate the partial products across inputs and input bits first. If applied on D6CIM, it would first accumulate the partial products of 16 inputs' MSBs and 16 weights (all bits), then the next 16 inputs and weights to the last 16 inputs and weights. Then it repeats the process using the next MSBs of the first 16 inputs. This scheme can cause frequent switching on WLs and LBLs since it needs to access new weights every cycle.

Instead, the proposed scheme accumulates the partial products across input *bits* first and then *inputs* (Fig. 5(b)). In other words, D6CIM first accumulates the partial products of

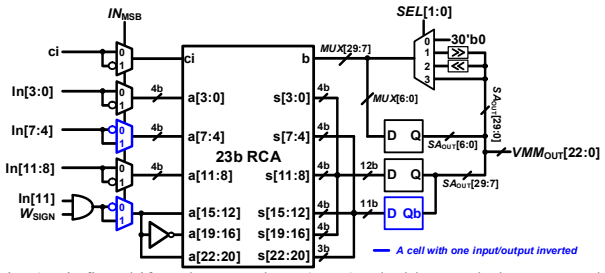

Fig. 4. Bit-first shift-and-accumulator (BFA). The blue symbol represented a cell with one input or output inverted.

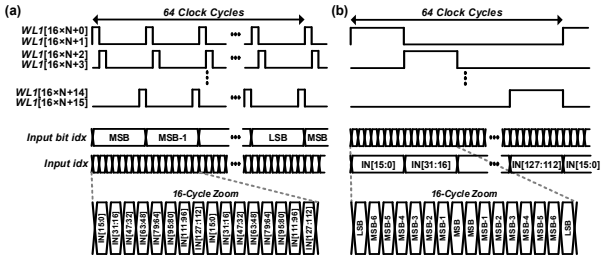

Fig. 5. (a) Conventional input-first accumulation scheme. (b) Proposed bit-first accumulation (BFA) scheme. Note that BFA also employs a bi-directional bit-serial input scheme to reduce the shifter area overhead.

16 inputs' MSBs and 16 weights (all bits), then the same inputs' second MSBs and the same weights to the inputs' LSBs. Then, it repeats the accumulation using the subsequent 16 inputs and 16 weights. This scheme reduces the switching activities on WL and LBLs by ~8X since it does not access new weights until it processes all 8 bits of inputs and thereby does not switch WLs and LBLs. The reduced switching activities largely improve energy efficiency.

Also, we propose the bi-directional bit-serial input scheme to reduce the area of the BFA. As shown in Fig. 5(a), we conventionally feed input bits in one direction, from MSB to LSB. To support 8b inputs in such a uni-directional bit-serial input scheme, we need one 1b left shifter and one 8b right shifter. However, to flexibly support 1-to-8b inputs, we need nine shifters and a 9:1 multiplexer, causing a large area overhead.

The proposed bi-directional bit-serial input scheme alternates the bit-serial direction, e.g., from LSB to MSB, then from MSB to LSB (Fig. 5(b)). In this scheme, we need only 1b left and right shifters and one 4:1 multiplexer, largely helping to reduce the area of BFA while enabling flexible 1-to-8b computation.

E. Circuit and Physical Design Optimizations

The HCA and BFA contain many full adder (FA) cells. The conventional CMOS adder requires ~28 transistors. To minimize the area overhead, we adopt the area-efficient transmission-gate (TG)-based FA and HA (Fig. 6(a,b)). Compared to the pass-gate-based cells [4], the TG-based cells do not induce a V_{th} drop.

We design the RCAs and compressors using the TG-based FAs and HAs (Fig. 7). Connecting too many TGs in series, however, degrades signal slew and delay. Therefore, we added inverters to restore the slew. Those inserted inverters, however, cause polarity changes. Consequently, we develop the FA and HA versions with one input inverted (Fig. 6(c,d)) and replace some regular FAs and HAs with the input-inverted versions. Similarly, in designing the BFA, we use the polarity-inverted multiplexers and D-flip-flops (DFF) to ensure the correctness of logic without adding extra inverters (Fig. 4).

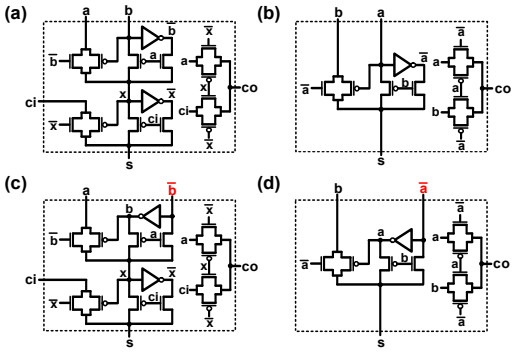

Fig. 6. (a) TG-based FA, (b) HA, (c) one-input-inverted FA, and (d) HA

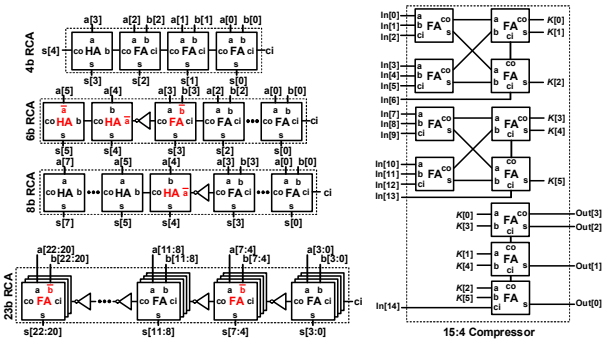

Fig. 7. 4b, 6b, 8b, 23b RCAs, and 15:4 compressor.

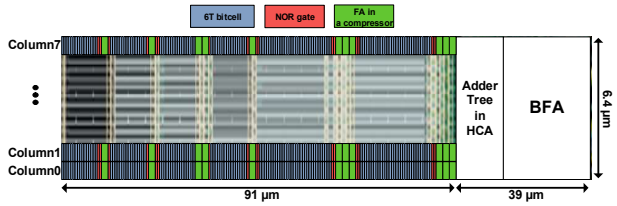

Fig. 8. Layout of the 8columns, HCA, and BFA. We optimize the layout to minimize the lengths of LBLs, WLs, and ILs.

During the physical design, we aim to shorten critical wires such as LBLs, WLs, and ILs since these wires' parasitic resistance and capacitance strongly impact throughput and energy efficiency. Fig. 8 shows the layout of the 8columns and the corresponding HCA and BFA modules. Right below each 16-bitcell sub-column, we place two NOR gates, minimizing the length of LBLs. Next, the 11 FAs, which form one 15:4 compressor, are distributed in each column to reduce the wire length from the NOR gates. Then, we place the adder tree of the HCA and the BFA below the 8columns. This placement minimizes the width of the 8columns, thereby shortening WLs.

III. Measurement Results

We prototyped the D6CIM test chip in a 28-nm CMOS. It takes 0.0159 mm^2 (Fig. 9). The bitcell is drawn in a logic rule, whose footprint is 0.379 µm^2. The 16-Kb bitcells take 41.6% of the total area, while the arithmetic hardware takes 45.8%. The row and column peripherals take 12.6%. The macro achieves an excellent weight density of 1005 Kb/mm^2.

Fig. 10(a) shows the measured shmoo plot of VMM operation across 0.6-1.1V. The maximum clock frequency is 360 MHz at 1.1V. Fig. 10(b) shows the measured compute density, where D6CIM achieve the peak compute density of 1.46 TOPS/mm^2 at 1.1V. Fig. 10(c) shows the energy efficiency measurement across VDDs and input sparsities. D6CIM achieves the peak energy efficiency of 60.4 TOPS/W

TABLE I. COMPARISON TO PRIOR DCIM PROTOTYPES

	This work	ISSCC'21 [1] Y. –D. Chih	ISSCC'22 [2] F. Tu	ISSCC'22 [3] H. Fujiwara	ISSCC'22 [4] D. Wang	VLSI'22 [5] C. –F. Lee	ISSCC'22 [6] B. Yan
Process [nm]	28	22	28	5	28	12	28
VMM Operation	Digital CIM	Digital CIM	Digital CIM	Digital CIM	Digital CIM	Digital CIM	Digital CIM
Transistors of cells and multipliers / bit	6T + 0.5T	6T + 4T	-	12T + 1T	6T + 2T	-	6T + 0.25T
Array Size [b]	16K	64K	96K	64K	16K	8K	32K
Supply Voltage [V]	0.6-1.1	0.72	0.6-1.0	0.5-0.9	0.45-1.1	0.72	0.8
Frequency [MHz]	30-360	500	50-220	360-1140	250	800	333
Input Precision [b]	1-8	1-8	8, 16	4	1-4	4-8	1-8
Weight Precision [b]	8	4, 8, 12, 16	8, 16	4	1	4, 8	1, 4, 8
Full-Precision Output	Yes	Yes	Yes	Yes	No	Yes	Yes
Singed/Unsigned VMM Support	Yes	Yes	Yes	Yes	No	Yes	No
Bitcell Area [μm^2]	0.379	0.379	-	0.075	0.862	-	0.257
Macro Area [μm^2]	0.0159	0.202	0.941	0.0133	0.049	0.0323	0.03
Weight Density [Kb/mm^2]	1005	316	102	**4812**	326	247	1067
[1)] Norm. Weight Density [Kb/mm^2]	**1005**	195	102	153	326	45.4	**1067**
[2)] Energy Efficiency [TOPS/W]	**22.4-60.4**	24.7	**30.8-57.8**	**17.5-63**	9.6-15.5	30.3	27.3
[2)] Compute Density [TOPS/mm^2]	0.12-1.46	4.53	1.43	**13.8-55.25**	2.59	10.40	0.17
[1,2)]Norm. Compute Density [TOPS/mm^2]	0.12-1.46	**2.80**	1.43	0.44-1.76	2.59	1.91	0.17

(1) Area is normalized quadratically to the 28-nm node. (2) One operation is defined as an 8b multiplication or addition.

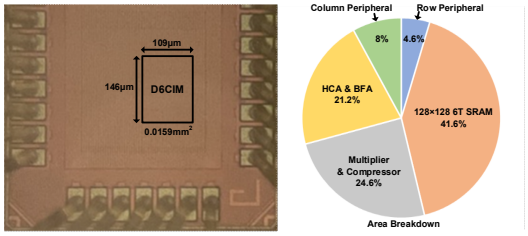

Fig. 9. Die micrograph and area breakdown.

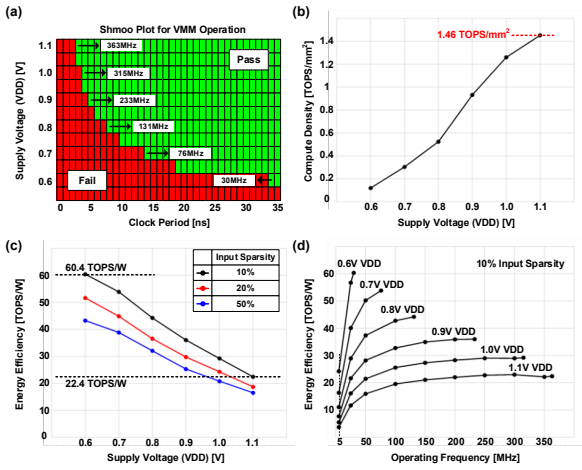

Fig. 10. (a) Measured shmoo plot of VMM operation. (b) Measured compute density (TOPS/mm^2). (c) Measured energy efficiency (TOPS/W) across VDDs and (d) clock frequencies.

at 0.6V and 10% input sparsity. Fig. 10(d) shows the energy efficiency measurement across clock frequencies.

Table I shows the comparisons to the prior DCIM works. As compared to [6], which achieves the best normalized weight density, D6CIM achieves 8.6X better compute density and 2.2X better energy efficiency at a similar weight density. For simpler comparisons, we propose a figure-of-merit (FoM), which is defined as *FoM=(norm. comp. density)×(norm. weight density)×(energy efficiency)*. Fig. 11 shows the comparisons using this FoM. D6CIM achieves up to 8.3X better FoM as compared to the prior works.

IV. CONCLUSION

This paper proposes D6CIM for 1-to-8b fixed-point computation. We create a novel time-sharing architecture with

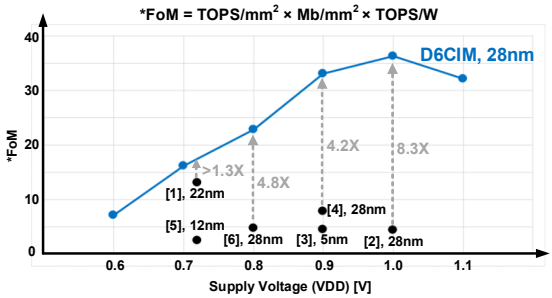

Fig. 11. Comparison with the prior works in the proposed FoM. *FoM=(norm. comp. density)×(norm. weight density)×(energy efficiency).*

three new techniques: static dual wordline access, hybrid compressor adder tree, and bit-first shift-and-accumulation. The measurements of the prototype chip demonstrate that D6CIM advances the prior arts in the product of the three key metrics.

REFERENCES

[1] Y. -D. Chih et al., "16.4 An 89TOPS/W and 16.3TOPS/mm^2 All-Digital SRAM-Based Full-Precision Compute-In Memory Macro in 22nm for Machine-Learning Edge Applications," IEEE International Solid-State Circuits Conference (ISSCC), 2021, pp. 252-254.

[2] F. Tu et al., "A 28nm 29.2TFLOPS/W BF16 and 36.5TOPS/W INT8 Reconfigurable Digital CIM Processor with Unified FP/INT Pipeline and Bitwise In-Memory Booth Multiplication for Cloud Deep Learning Acceleration," IEEE International Solid-State Circuits Conference (ISSCC), 2022, pp. 254-255.

[3] H. Fujiwara et al., "A 5-nm 254-TOPS/W 221-TOPS/mm^2 Fully-Digital Computing-in-Memory Macro Supporting Wide-Range Dynamic-Voltage-Frequency Scaling and Simultaneous MAC and Write Operations," IEEE International Solid-State Circuits Conference (ISSCC), 2022, pp. 186-187.

[4] D. Wang et al., "DIMC: 2219TOPS/W 2569F^2/b Digital In-Memory Computing Macro in 28nm Based on Approximate Arithmetic Hardware," IEEE International Solid-State Circuits Conference (ISSCC), 2022, pp. 266-268.

[5] C. -F. Lee et al., "A 12nm 121-TOPS/W 41.6-TOPS/mm^2 All Digital Full Precision SRAM-based Compute-in-Memory with Configurable Bit-width For AI Edge Applications," IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits), 2022, pp. 24-25.

[6] B. Yan et al., "A 1.041-Mb/mm^2 27.38-TOPS/W Signed-INT8 Dynamic-Logic-Based ADC-less SRAM Compute-in-Memory Macro in 28nm with Reconfigurable Bitwise Operation for AI and Embedded Applications," IEEE International Solid- State Circuits Conference (ISSCC), 2022, pp. 188-190.

[7] M. -F. Chang et al., "A Compact-Area Low-VDDmin 6T SRAM With Improvement in Cell Stability, Read Speed, and Write Margin Using a Dual-Split-Control-Assist Scheme," in IEEE Journal of Solid-State Circuits, vol. 52, no. 9, pp. 2498-2514, Sept. 2017.