A 35.6 TOPS/W/mm2 3-Stage Pipelined Computational SRAM With Adjustable Form Factor for Highly Data-Centric Applications

Jean-Philippe Noel Noel, Manuel Pezzin, Roman Gauchi, Jean-Frédéric Christmann, Maha Kooli, Henri-Pierre Charles, Lorenzo Ciampolini, Mariam Diallo, Florent Lepin, Benjamin Blampey, et al.

A 35.6 TOPS/W/mm^2 3-Stage Pipelined Computational SRAM With Adjustable Form Factor for Highly Data-Centric Applications

J.-P. Noel, M. Pezzin, R. Gauchi, J.-F. Christmann, *Member, IEEE*, M. Kooli, H.-P. Charles, L. Ciampolini, M. Diallo, F. Lepin, B. Blampey, P. Vivet, *Member, IEEE*, S. Mitra, *Fellow, IEEE*, and B. Giraud

***Abstract*—In the context of highly data-centric applications, close reconciliation of computation and storage should significantly reduce the energy-consuming process of data movement. This letter proposes a computational SRAM (C-SRAM) combining in- and near-memory computing (IMC/NMC) approaches to be used by a scalar processor as an energy-efficient vector processing unit. Parallel computing is thus performed on vectorized integer data on large words using usual logic and arithmetic operators. Furthermore, multiple rows can be advantageously activated simultaneously to increase this parallelism. The proposed C-SRAM is designed with a two-port pushed-rule foundry bitcell, available in most existing design platforms, and an adjustable form factor to facilitate physical implementation in a SoC. The 4-kB C-SRAM testchip of 128-b words manufactured in 22-nm FD-SOI process technology displays a sub-array efficiency of 72% as well as an additional computing area of less than 5%. The measurements averaged on 10 dies at 0.85 V and 1 GHz demonstrate an energy efficiency per unit area of 35.6 and 1.48 TOPS/W/mm^2 for 8-b additions and multiplications with 3- and 24-ns computing latency, respectively. Compared to a 128-b SIMD processor architecture, up to 2× energy reduction and 1.8× speed-up gains are achievable for a representative set of highly data-centric application kernels.**

***Index Terms*—Artificial intelligence, in-memory computing (IMC), near-memory computing (NMC), SRAM circuit, vector processing unit.**

I. Introduction

In- and near-memory computing (IMC/NMC, depending on pre/post-sense-amplifier computation) are considered today as promising solutions to overcome the energy wall problem raised with highly data-centric applications [1], [2]. This problem corresponds to the huge amount of energy to move the data between memory and processing element compared to the small amount of energy required for the computation itself. Moreover, the technology scaling only exacerbates this phenomenon because although it reduces the computing energy, it increases the impact of moving data (by the increase of parasitic elements) that is predominant [3]. Recent works demonstrate the interest to use SRAM-based IMC architectures enabling multirow selection during the read operation to perform *in-situ* logic operations (NOR, AND, etc.), also called scouting logic [3]–[5]. This leads to significant energy reductions for representative sets of edge-AI and security-oriented kernels. Nevertheless, most of them require custom bitcells designed at logic-rules, at the cost of a drastic loss in memory density. Furthermore, to perform bitwise IMC operations, the operands must be physically aligned at the array level. This forbids to interleave the words on a single row, fixing the macro form factor to MUX-1.

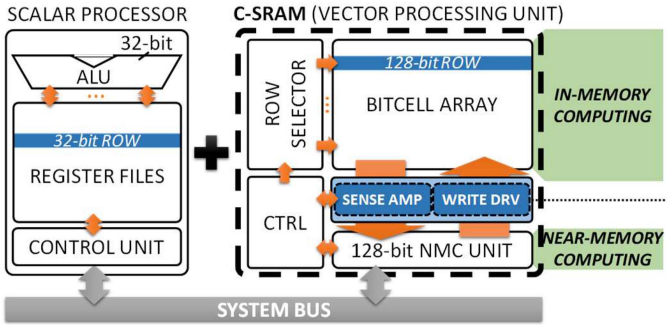


Fig. 1. Integration scheme of the proposed C-SRAM used as an energy-efficient vector processing unit.

In this letter, we propose a 128-b computational SRAM (C-SRAM) architecture with an adjustable form factor to facilitate physical implementation and compatible with two-port (8T) pushed-rule (SRAM-rules) foundry bitcells to keep a high memory density. Furthermore, we have paid particular attention to minimize the additional computing area and the performance penalty, while limiting the computing latency with a 3-stage pipeline. The main contributions of this letter are as follows.

1) To quantify the energy and power contributions (computing, fetch, store, and NOP) of IMC and NMC operations with a 3-stage pipeline.
2) To demonstrate the feasibility of designing high-density C-SRAM with the adjustable form factor.

The remainder of this letter is organized as follows. Section II details the proposed 128-b C-SRAM architecture as well as IMC/NMC operations and the associated pipeline flow. Section III explains the circuit design choices enabling the form factor adjustment to facilitate physical implementation. Then, the measurement results are depicted and compared with the prior art in Section IV. Application benchmarks versus 128-b SIMD processor are also presented. Finally, Section V summarizes this letter.

II. Proposed C-SRAM Architecture

The proposed C-SRAM architecture can be used to design an energy-efficient vector processing unit performing heavily parallel operations which can be finely interleaved with sequential operations of a scalar processor (Fig. 1). Fig. 2(a) shows the baseline IMC approach based on a three-port (10T) custom bitcell [5], which is not supported by most of the foundries in advanced technology nodes. The proposed IMC approach [Fig. 2(b)] is based on dual arrays of two-port (8T) foundry bitcells per bank containing complementary data to enable simultaneously NOR and AND

Manuscript received May 4, 2020; revised June 24, 2020; accepted July 1, 2020. Date of publication July 20, 2020; date of current version September 1, 2020. This article was approved by Associate Editor Stefan Rusu. This work was supported by French ANR via Carnot Funding. *(Corresponding author: J.-P. Noel.)*

J.-P. Noel, M. Pezzin, R. Gauchi, J.-F. Christmann, M. Kooli, H.-P. Charles, L. Ciampolini, M. Diallo, F. Lepin, P. Vivet, and B. Giraud are with LIST, CEA, Université Grenoble Alpes, 38000 Grenoble, France (e-mail: jean-philippe.noel@cea.fr).

B. Blampey is with LETI, CEA, Université Grenoble Alpes, 38000 Grenoble, France.

S. Mitra is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA, and also with the Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

Digital Object Identifier 10.1109/LSSC.2020.3010377

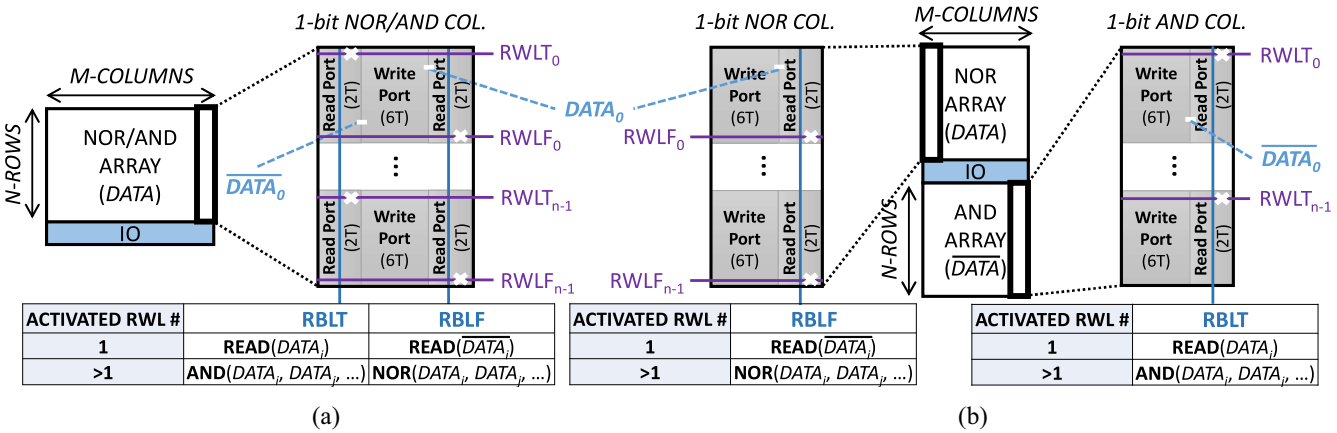


Fig. 2. (a) Reference (10T bitcell) and (b) proposed IMC solution based on 8T foundry bitcells to implement dual arrays enabling both NOR and AND operations.

TABLE I
COMPARISON WITH PREVIOUS WORKS @0.9 V

	This Work	JSSC 2020 [4]		JSSC 2018 [5]	VLSI 2019 [6]
Process technology node	22nm	28nm		40nm	65nm
SRAM bitcell	foundry (SRAM-rules) 8T	custom (logic-rules) 8T		custom (logic-rules) 10T	custom (logic-rules) 6T
Memory capacity	4kB	4kB (instance)	128kB (full chip)	8kB	8kB
Memory density	**1.87Mb/mm²**	0.816Mb/mm²		0.3Mb/mm²	0.01Mb/mm²
Adjustable form factor	**yes**	no		no	no
Computing type	IMC+NMC (bit-parallel)	IMC+NMC (bit-serial)		IMC+NMC (bit-parallel)	IMC+NMC (bit-parallel)
Supported IMC operation	AND/OR (multi-row selection)	AND/OR (multi-row selection)		AND/OR (2-rows selection)	XNOR
Supported NMC operation	Logic/Add/Sub/Comp/Shift/Mul	Logic/Add/Sub/Comp/Mul/Div/FP		Logic/Shifter/Rotator/S-BOX	MAC
Targeted usage	vector processing unit	vector processing unit		accelerator (cryptography only)	accelerator (RNN only)
Operating frequency	**1GHz** @0.9V	375MHz @0.9V		90MHz @0.9V	75MHz @0.9V
Peak performance (GOPS)	16 [a] (8-bit ADD)	3 [b] (8-bit ADD)	96 [b] (8-bit ADD)	not reported	614
	0.67 [a] (8-bit MUL)	0.23 [b] (8-bit MUL)	7.5 [b] (8-bit MUL)		
Computing latency @Op. freq. (ns)	**3** [a] (8-bit ADD)	21 (8-bit ADD)		not reported	not reported
	24 [a] (8-bit MUL)	271 (8-bit MUL)			
Max. energy per computed bit (fJ/bit)	248 @0.9V [a]	1536 @0.9V [b]	48 @0.9V [b]	152 @0.9V [c]	2.7 @0.9V [d]
Peak energy efficiency per unit area (TOPS/W/mm²)	**30.5** [a] (8-bit ADD)	0.08 (8-bit ADD) [b]	2.6 (8-bit ADD) [b]	not reported	15.1
	1.27 [a] (8-bit MUL)	0.009 (8-bit MUL) [b]	0.28 (8-bit MUL) [b]		

[a] constant whatever computing load [b] peak performance when the pipeline is full [c] assuming 128-bit words [d] assuming 256-bit words

operations when multirow selection is activated. Fig. 3 shows the proposed C-SRAM architecture, including two memory banks in MUX-1 configuration (no word interleaving). Each bitcell array (NOR and AND) is connected to a local I/O (LIO), while a global I/O (GIO) vertically interfaces the two banks of the page with a vectorized NMC unit. Based on this architecture, a representative set of logic and arithmetic operations is supported [Fig. 4(a)]. To ensure conflict-free memory access, while performing energy-efficient operations, the six types of C-SRAM instruction are executed in a 3-stage pipeline [Fig. 4(b)]. Depending on the instruction sequence [Fig. 4(c)], minimum and maximum power consumption occur when the three pipeline stages are either idle (NOP) or simultaneously operating (IMC+NMC+WR), respectively.

III. ADJUSTABLE FORM FACTOR DESIGN AND IMPLEMENTATION

A 256×128 b (4 kB) C-SRAM testchip has been manufactured in 22-nm FD-SOI process technology, featuring one page of two banks organized in dual arrays of 128-rows of 128-columns. In order to demonstrate the feasibility of designing a C-SRAM with an adjustable form factor, an extra memory page including only interconnecting

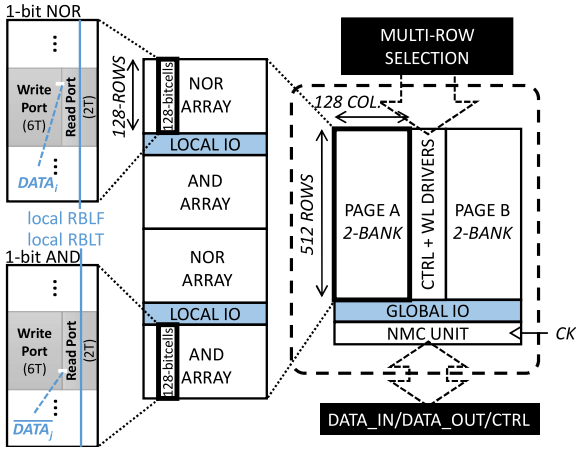


Fig. 3. Proposed C-SRAM architecture based on two adjacent memory pages.

layers taking the parasitic elements into account has been implemented. To ensure IMC operations (NOR/AND) between the two adjacent pages, an original horizontal BL metallization scheme is

COMPUTING TYPE	IMC	NMC
MAX. OPERAND #	*128*	*2*
LOGIC OPERATION	AND	AND/NAND
	NOR	OR/NOR
	-	XOR/XNOR
		INV
		IMPLY
ARITHMETIC OPERATION	-	ADD/SUB
		COMP
		SLI/SRI
		MUL[1]

(a)

PIPELINE STAGE		*1st*	*2nd*	*3rd*
INSTR. TYPE	RD	RD	*NOP*	*NOP*
	WR	*NOP*	*NOP*	WR
	COPY	RD	*NOP*	WR
	MO-LOG[2]	IMC	*NOP*	WR
	MO-LOG + LOG/ARITH	IMC	NMC	WR
	LOG/ARITH	RD[3]	NMC	WR

[1]*shift-and-add* multiplication
[2]multi-operand (>2) logic operations
[3]2 different addresses can be accessed at the same time

(b)

INSTR.	*i*	*i+1*	*i+2*	*i+3*	*i+4*	*i+5*	...	...
STAGE-1	RD	RD	*NOP*	RD*	IMC	IMC	...	...
STAGE-2	...	*NOP*	*NOP*	*NOP*	NMC	*NOP*	NMC	...
STAGE-3	...	...	*NOP*	WR	WR	WR	WR	WR

Min. power consumption

Max. power consumption

(c)

Fig. 4. IMC and NMC operations: (a) supported functions, (b) instruction types, and (c) instruction flow example.

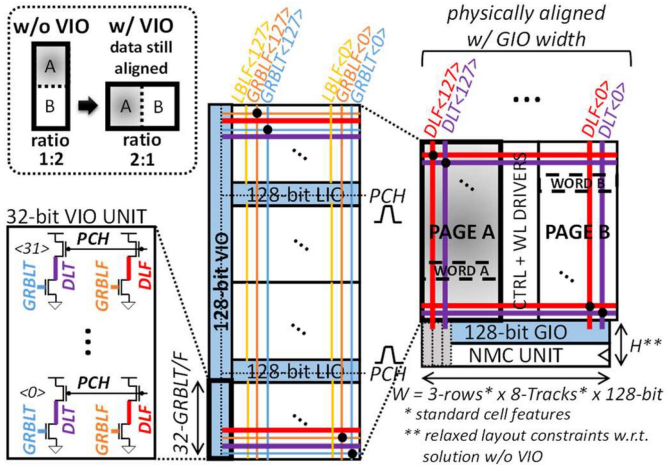


Fig. 5. Horizontal BL metallization scheme enabling adjustable form factor for interpage IMC operands (e.g., "word_A" AND "word_B").

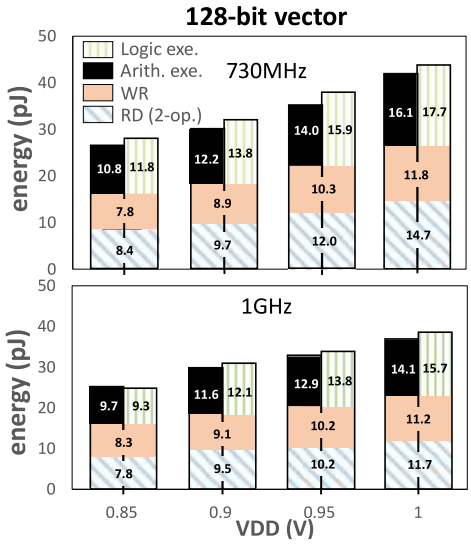


Fig. 6. Measured logic and arithmetic instruction energy versus VDD.

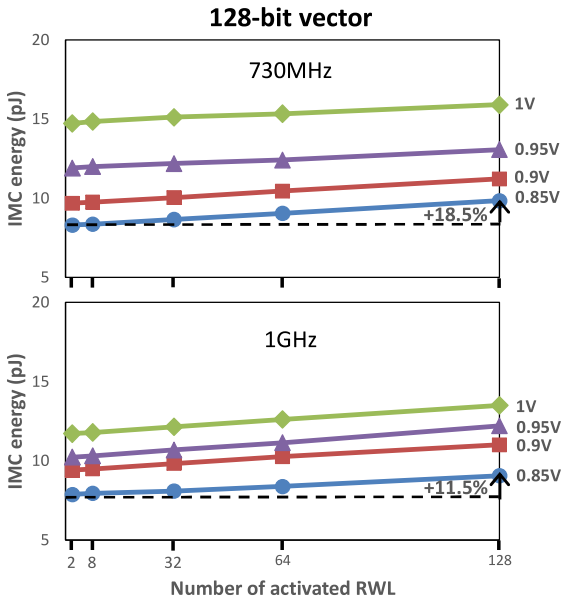


Fig. 7. Measured IMC energy versus number of activated RWL.

INSTR. TYPE	PIPELINE STAGE 1	2	3	Energy (fJ/bit) @0.85V Min.	Max.
RD	RD	*NOP*	*NOP*	118	
WR	*NOP*	*NOP*	WR	122	
COPY	RD	*NOP*	WR	154	
MO-LOG*	IMC	*NOP*	WR	155	164
MO-LOG + LOG/ARITH	IMC	NMC	WR	199	211
LOG/ARITH	RD	NMC	WR	199	202

(a)

PIPELINE FILLING SCENARIO			Power (mW) @0.85V/1GHz
STAGE-1	*STAGE-2*	*STAGE-3*	
NOP	NOP	NOP	11
RD (*2-OPERANDS*)	NMC	WR	25.5
IMC (*2-OPERANDS*)			25.5
IMC (*128-OPERANDS*)			27.1

*multi-operand (>2) logic operations

(b)

Fig. 8. Measured (a) instruction energy and (b) pipeline power consumption.

used to connect together the vertical global BLs of each bank to a vertical I/O (VIO) (Fig. 5). Multioperand IMC is then enabled by bitwise row selection inputs providing different test scenarios (each bit activates a WL). The proposed metallization scheme only requires two additional metal layers compared to a conventional SRAM in the same process technology.

IV. Measurement Results and Applications

All reported data are measurements averaged on 10 dies. Fig. 6 shows that the computing part performed in the second

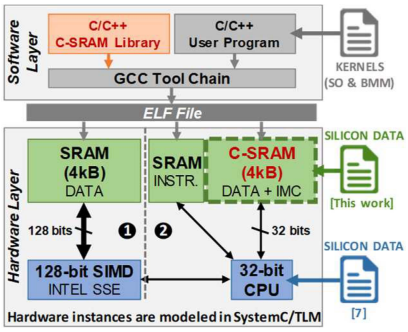


Fig. 9. Benchmark platform using SystemC/TLM calibrated with silicon data.

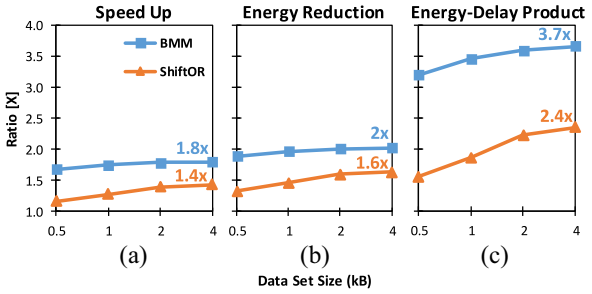
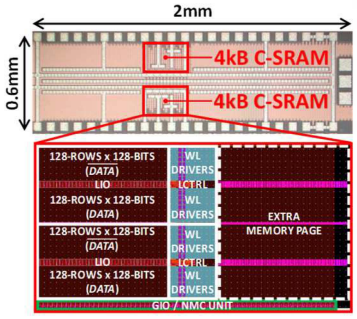


Fig. 10. Simulated (a) speed-up, (b) energy reduction, and (c) EDP ratio of 128-b C-SRAM versus 128-b SIMD processor (baseline).

stage of the pipeline (operation execution) represents more than 40% of the total instruction energy. The rest is spent by fetching the operands (∼30%) and storing the results (∼30%). Note that the difference between logic and arithmetic operations (except for multiplication) is negligible (<4%). These results highlight the advantages of performing NMC operations when the data to be processed are already in memory. Fig. 7 represents the IMC operation (NOR/AND) energy according to the number of activated RWL (corresponding to operands). The difference between 2 and 128 activated RWL is below 20% because most of the energy is spent in the IOs (local, global, and vertical) and the controller. Fig. 8(a) shows that the instruction energy ranges from 118 to 211 fJ/bit (1.79×) at 0.85 V according to the instruction type. Fig. 8(b) lists the power consumption for different pipeline filling scenarios, varying from 11 to 27.1 mW (2.46×) at 0.85 V and 1 GHz. Using a 3-stage pipeline allows to limit the variation in energy and power consumption, which will make easier the power management at the system level.

To evaluate the benefits at the architecture level, the C-SRAM is compared to a 128-b SIMD processor architecture (using Intel SSE instructions) for two representative application kernels: Shift-OR (database oriented) and Boolean matrix multiplication (AI oriented) using a simulation platform. Fig. 9 describes this SystemC/TLM-based platform power annotated with silicon measurements of the C-SRAM and a baseline 128-b SIMD processor architecture [7]. Fig. 10 shows gains up to 1.8× in speed-up, 2× in energy reduction, and 3.7× in energy–delay product (EDP) at 0.85 V and 1 GHz. Table I compares the C-SRAM to previous works and shows significant improvements in terms of memory density (up to 1.87 Mb/mm^2), form factor flexibility, computing latency (down to 3 ns), and energy efficiency per unit area (up to 30.5 TOPS/W/mm^2) for the highest frequency (1 GHz). For the sake of fairness, energy-related performance is compared at the same VDD (0.9 V). Fig. 11 shows the die micrograph and the testchip summary at nominal VDD (0.85 V). In this condition, the energy efficiency per unit area is increased by



Process technology		22nm FD-SOI
Metal scheme		8ML @ macro level
Die size		2 x 0.6mm
4kB macro size		0.130 x 0.135mm
Bitcell	Type	Two-Port (8T)
	Area	0.185µm²
Memory	Capacity	4kB (2 x 2kB)
	Density	1.87Mb/mm²
Sub-array efficiency		72%
Data type		boolean/integer
Vector size		128-bit
Instruction flow		1 instr./cycle (3-cycles latency)
Computing area		<5% overhead
Operating frequency		1GHz
Leakage power		212µW
Read power		15.1mW
Write power		15.6mW
Computing power		25.5mW*
Computing energy		202fJ/bit*
TOPS/W/mm²		35.6* 1.48**
Measured chip #		10 (0.3Mb)

@0.85V (nom. VDD)

*8-bit addition **8-bit multiplication

Fig. 11. Die micrograph and chip summary @0.85 V.

16% to achieve 35.6 and 1.48 TOPS/W/mm^2 for 8-b additions and multiplications, respectively.

V. Conclusion

This letter presents a 4-kB C-SRAM circuit based on two-port pushed-rule foundry bitcells manufactured in 22-nm FD-SOI process technology. The testchip achieves the highest memory density (1.87 Mb/mm^2) compared to previous works, while facilitating physical implementation at SoC level thanks to a novel design technique enabling the form factor adjustment. Based on a 3-stage pipeline, access to memory as well as execution of operations (IMC/NMC) can operate up to 1 GHz with a latency of 3 ns (except for 8-b multiplication which has a 24-ns latency). This small pipeline depth limits the energy (118–211 fJ/bit) and power consumption (11–27.1 mW) variations as well as the additional area of the unit managing operations (<5%). Thanks to the combined benefits of all these circuit design choices, the proposed C-SRAM solution achieves an energy efficiency per unit area of 35.6 and 1.48 TOPS/W/mm^2 at 0.85 V for 8-b additions and multiplications, respectively. At the architecture level, compared to a 128-b SIMD processor dealing with a 4-kB data set, up to 3.7× EDP is obtained with a representative set of edge-AI and database application kernels.

References

[1] N. Verma *et al.*, "In-memory computing: Advances and prospects," *IEEE Solid-State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Aug. 2019.

[2] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-Mb in-memory-computing CNN accelerator employing charge-domain compute," *IEEE J. Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, Jun. 2019.

[3] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, San Francisco, CA, USA, 2014, pp. 10–14.

[4] J. Wang *et al.*, "A 28-nm compute SRAM with bit-serial logic/arithmetic operations for programmable in-memory vector computing," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 76–86, Jan. 2020.

[5] Y. Zhang, L. Xu, Q. Dong, J. Wang, D. Blaauw, and D. Sylvester, "Recryptor: A reconfigurable cryptographic cortex-M0 processor with in-memory and near-memory computing for IoT security," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 995–1005, Apr. 2018.

[6] R. Guo *et al.*, "A 5.1pJ/neuron 127.3us/inference RNN-based speech recognition processor using 16 computing-in-memory SRAM macros in 65nm CMOS," in *Proc. Symp. VLSI Circuits*, Kyoto, Japan, 2019, pp. C120–C121.

[7] J.-F. Christmann *et al.*, "A 50.5 ns wake-up-latency 11.2 pJ/Inst asynchronous wake-up controller in FDSOI 28 nm," *J. Low Power Electron. Appl.*, vol. 9, no. 1, p. 8, 2019.