This article has been accepted for publication in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits. This is the author's version which has not been fully edited ar
content may change prior to final publication. Citation information: DOI 10.1109/JXCDC.2023.3322292

Energy Efficient Logic and Memory Design with Beyond-CMOS Magneto-Electric Spin-Orbit (MESO) Technology toward Ultra Low Supply Voltage

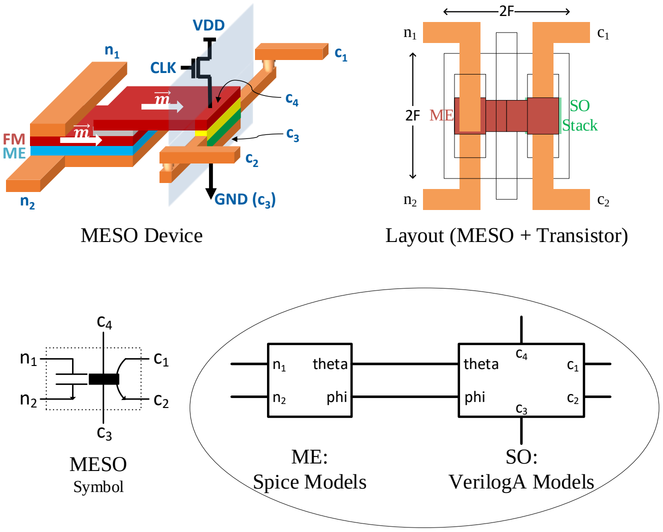
Rohit Rothe, *Student Member*, *IEEE*, Hai Li, *Member*, *IEEE*, Dmitri E. Nikonov, *Senior Member*, *IEEE*, Ian A. Young, *Life Fellow*, *IEEE*, Kyojin Choo, *Member*, *IEEE*, David Blaauw, *Fellow*, *IEEE*

***Abstract*— Devices based on the spin as the fundamental computing unit provide a promising beyond-complementary metal-oxide-semiconductor (CMOS) device option, thanks to their energy efficiency and compatibility with CMOS. One such option is a Magneto-Electric Spin-Orbit (MESO) device, an attojoule-class emerging technology promising to extend Moore's law. This paper presents circuit design and optimization techniques such as device stacking and a canary circuit-based asynchronous clock pulse generation scheme for Magneto-Electric Spin-Orbit (MESO) device technology. With these targeted circuit techniques, the MESO energy efficiency can be improved by ~1.5x. Novel architectures for arithmetic logic and effective realization of in-memory computing are also proposed that utilize the unique properties of this promising new technology.**

***Index Terms*— Beyond-CMOS logic, magnetoelectric (ME), SPICE, spin-orbit (SO), canary circuits, in-memory computation**

I. Introduction

The VLSI industry has always strived for improvements in performance, power, size, and cost with each technology generation. However, the returns from CMOS scaling have started to diminish with recent technology nodes. The CMOS operating voltage has not reduced at the same rate as density gains due to the marginal reduction of the threshold voltage. The supply voltage scaling has become increasingly challenging, and the off-transistor current leakage has limited the system's energy efficiency. This has hampered strategies for overcoming the CMOS power dissipation concern. An important avenue in the search for lower power and better performance is exploring beyond CMOS approaches. Many alternatives have been proposed to complement CMOS and sustain the trajectory of Moore's law [1]. One of the leading candidates is the Magneto-Electric Spin-Orbit (MESO) device [2-4] which promises to be in the attojoule energy efficient class with supply voltage in the range of < 100 mV. MESO compares very well with other beyond-CMOS technologies as well as advanced CMOS processes [5, 6]. MESO technology exhibits an excellent throughput at very low power density and delay [6].



MESO Device

Layout (MESO + Transistor)

MESO
Symbol

ME:
Spice Models

SO:
VerilogA Models

Fig. 1. Modelling MESO device using a hybrid Verilog-A & spice model [8].

A MESO device consists of two primary blocks: 1) a input voltage driven magnetoelectric (ME) capacitor that switches a

Manuscript received July 30, 2023; revised xxx x, 2023; accepted xxx x, 2023.
This work was funded in part by Intel Semiconductor Corporation.

Rohit Rothe and David Blaauw are with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: rohitrr@umich.edu).

Hai Li, Dmitri E. Nikonov, and Ian A. Young are with the Components Research Group, Intel Corporation, Hillsboro, OR 97124 USA (e-mail: hai.li@intel.com).

Kyojin Choo, was with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. (e-mail: kyojin.choo@epfl.ch).

ferromagnet (FM) and 2) a spin-orbit (SO) output module in which the spin current from the FM layer creates a positive or negative output charge current, depending on the magnetization in the FM. This spin current flows into the inverse spin-orbit coupling (ISOC) conversion stack beneath the FM, which performs spin-to-charge conversion based on the Inverse Spin-Hall Effect (ISHE) and the inverse Rashba–Edelstein Effect (IREE) [7–9]. Depending on the spin current polarity, either positive or negative charge current flows into the metallic interconnect that drives the next logic gate.

To regulate the amount of charge current flowing through the ISOC stack, some combination of header or footer power gating CMOS transistors are used. These transistors are clocked using multi-phase clock pulses. This ensures that the device consumes DC current only when required for logic gate evaluation.

Fig.1 shows the mapping of the MESO device to the simulation model. The nodes n_1 and n_2 are the ME capacitor nodes. Whereas nodes c_1-c_4 represent the charge-based terminals of the SO module [10]. A SPICE and Verilog-A hybrid model, as shown in Fig. 1, based on the physics of the MESO device, was described in [10]. Multiphysics coupling is computationally intensive for circuit design and simulation. This model comprehends all the primary physics behavior using a circuit approach. Using this model, this paper demonstrates an implementation of arithmetic operations, such as addition, multiplication, and in-memory computing. An asynchronous clock generation scheme is proposed to address the power gated through current consumption of the MESO logic family. Furthermore, a memory architecture demonstrating how the MESO device is ideally positioned to perform efficient in-memory computing is also proposed.

II. Device Stacking

The fundamental logic unit using the MESO device is a majority gate. A 3-input majority gate would only require 4 MESO devices whereas implementation in CMOS would require at least 14 transistors. Moreover, as the complexity of logic increases this difference escalates (as shown in Table 1), for example, a 5-input majority gate requires 6 MESO devices as compared to 62 CMOS transistors. A single sided MESO (area: 2Fx1F = $2F^2$) can overlay over a CMOS transistor [5] (F = smallest feature size). The area of a CMOS transistor is 2Fx2F = $4F^2$. The differential MESO device area can be extrapolated to 2Fx2F = $4F^2$ (same as a CMOS transistor). An example layout of a differential MESO device along with its header transistor is shown in Fig. 1. Table 1 also accounts for the area overhead of header transistors required for MESO implementation (area shown as addition of MESO area + CMOS header area).

Header transistors control the flow of the power supply current into the MESO device so that alternating clock phases allow devices to cascade in logic gate stages. The stacking of MESO logic devices is proposed to reuse this current and allow multiple devices in a column of logic to function in parallel. Compared to the standard implementation [2], this enables ~1.5x energy saving as verified in simulations and is shown in Table 2. In the current implementation, up to 3 devices can be stacked. The marginal energy saving, and the requirement of a much higher supply voltage is a deterrent against stacking more than 3 MESO devices.

Table 1: Comparison of number of devices required to implement logic gates in MESO vs. CMOS transistors.

Gates	MESO	Area (um^2)	CMOS	Area (um^2)
NOT	2	$4F^2 + 4F^2$	2	$4F^2$
NAND	4	$8F^2 + 8F^2$	4	$8F^2$
NOR	4	$8F^2 + 8F^2$	4	$8F^2$
3-Maj	4	$8F^2 + 8F^2$	14	$28F^2$
5-Maj	6	$12F^2 + 12F^2$	62	$124F^2$
7-Maj	8	$16F^2 + 16F^2$	282	$564F^2$

Fig. 2 shows the stacking of two inverters. Devices (#INP1, #INP2) and (#INV1, #INV2) will share the DC through current controlled by the shared NMOS header. Another advantage of stacking devices is that it reduces the overhead of NMOS devices used for MESO clock pulse gating.

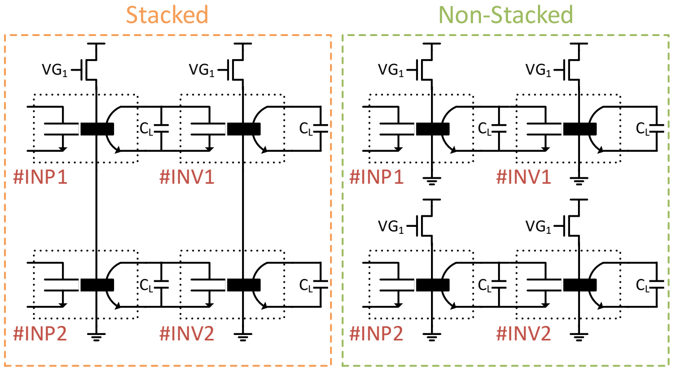


Fig. 2. Stacking MESO devices to efficiently reuse the DC through current.

Table 2: Comparison of Energy per operation for stacked vs. non-stacked version of MESO inverters.

Stacked Inverter			Non-stacked Inverter		
Time Period (ns)	VDD (mV)	Energy (fJ)	Time Period (ns)	VDD (mV)	Energy (fJ)
1	450	0.889	1	300	1.526
2	250	0.547	2	150	0.768
4	175	0.529	4	100	0.681
8	125	0.537	8	75	0.768

Comparing the stacked and non-stacked versions of MESO inverters vs. CMOS technology yields the results shown in Fig. 3. Supply voltage as low as 100 mV can be used for the stacked inverter. The load capacitance doesn't severely impact the energy for the MESO versions, as shown in Fig. 3. This is because the maximum contribution of energy in a MESO device comes from the DC current that flows from c_4 to c_3. The generated charge current associated with the load capacitance compared to the DC current is extremely small in comparison. The canary circuit enables the minimum time of the DC current to flow. With more efficient SO modules in the future, the percentage of DC power to SO generated power will reduce enabling lower overall total power. As expected, the energy consumed by the CMOS inverter goes up almost linearly as the load capacitance goes up.

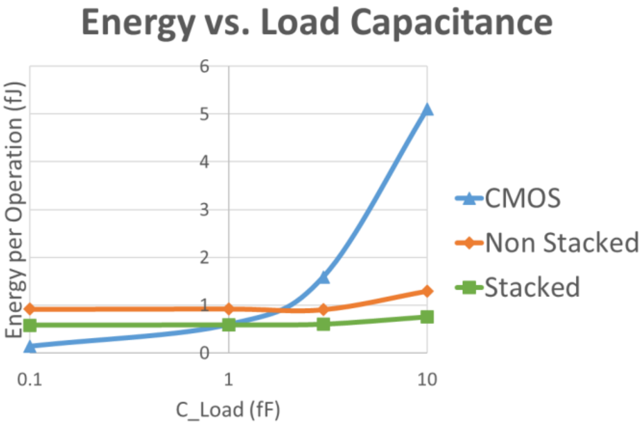


Fig. 3. Energy Comparison for MESO vs. CMOS (12 nm FinFET).

III. Asynchronous Clock Generation

Using multi-phase clocks for timing the stages of logic has two main issues: 1) The clock pulses must be generated externally and distributed globally, incurring energy overhead. 2) The clock phases are not adjusted to the complexity of the logic gates it is clocking. The width of the logic pulse required for the magnetization to flip for a 3-input majority gate is shorter than that required for a 5 or a 7-input majority gate.

A lot of complex logic operations in CMOS can be expressed simply using majority gates. Full adders' Carry-out and Sum-out can be generated by using 3 and 5-input majority gates respectively. Priority encoders are essential digital components of many modern architectures. They require high fan-in logic gates [11] such as 4-AND, 8-AND, etc. to support higher bit-widths. These can be easily implemented in majority gate logic.

$$AND4 = 7 - Majority(A, B, C, D, '0', '0', '0')$$

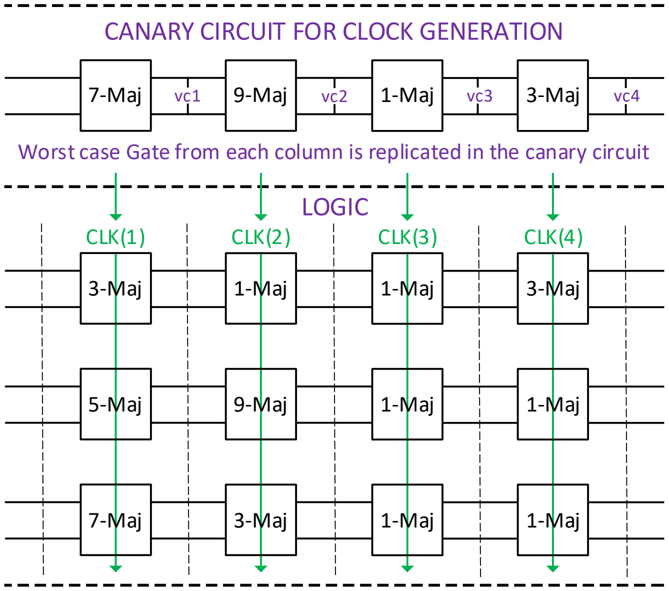


Fig. 4. An example of logic data-path along with the corresponding canary circuit for clock generation.

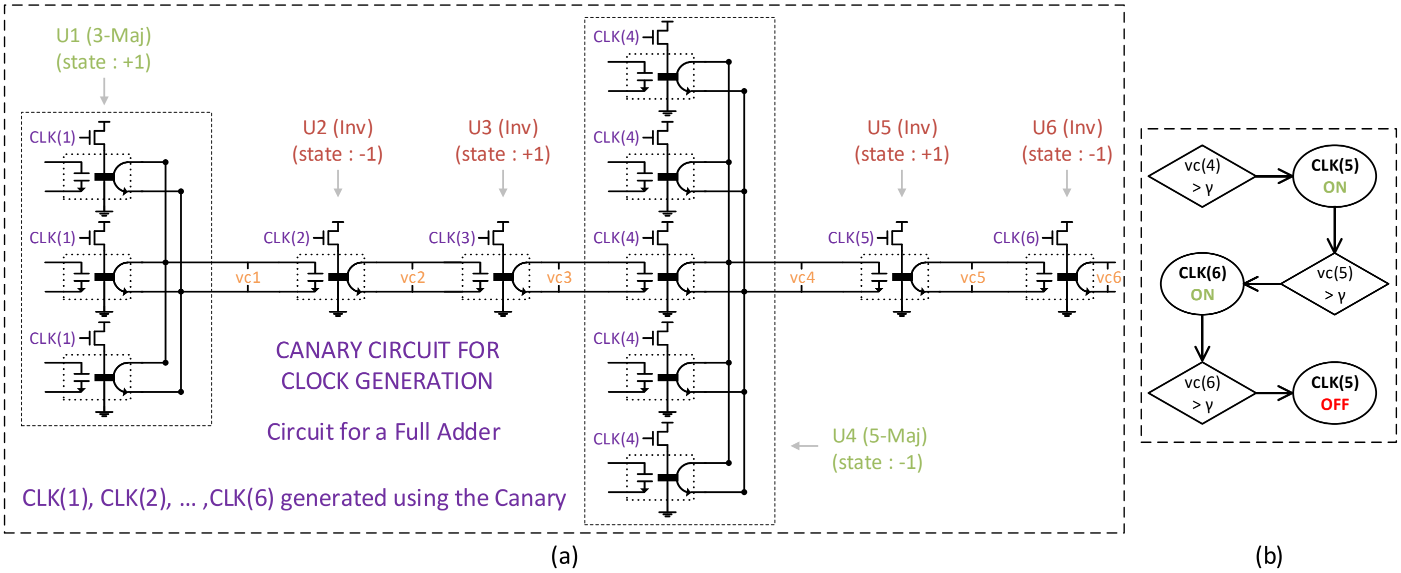


Fig. 5. (a) An example of a canary circuit used for asynchronous clock generation (Full Adder implementation for carry and sum generation). (b) Flow chart of triggers and states involved in clock generation for CLK (5).

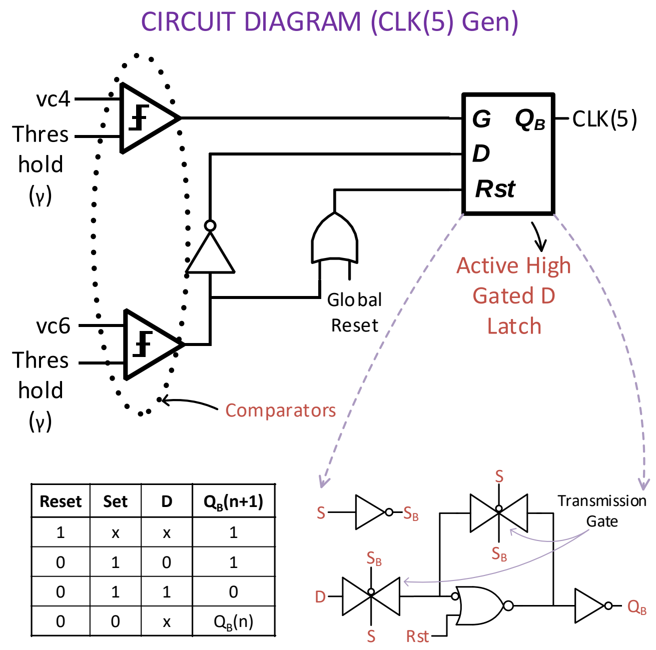
Table 3: Comparison of time required for magnetization to flip for different number of input complexity of the majority gates (No canary circuit involved).

Gates	Time Required
1-Majority	0.62 ns
3-Majority	1.41 ns
5-Majority	2.32 ns
7-Majority	4.45 ns

Table 3 shows that the time required for magnetization to flip varies a lot across the varying complexity of the MESO majority gates (time difference of 6.93x from 1-majority (buffer/inverter) to 7-input-majority). If no canary circuit was used, a pessimistic clock pulse of 5-Majority gate (2.32 ns, assuming 5-Majority is the highest complexity) would have to be used for every logic stage. In that case the 1-Majority is unnecessarily ON for an extra (2.32-0.62) = 1.7 ns. Without a canary circuit it is necessary to allocate some margin for the worst-case timing. One option is to consider a 7-majority gate as the worst-case pulse width.

Designing a data-path using MESO devices involves various combinations of majority gates, all of them requiring different lengths of time. Given the magnitude of the current when the headers are on, it is paramount that a gate is ON for the minimum required amount of time.

Canary-based clocking, as shown in Fig. 4 (an example of datapath design with canary circuit), is proposed to address this. The clock for each logic stage is generated locally depending on the type of majority gate(s) used in that logic stage, reducing overall circuit delay.



Reset	Set	D	$Q_B(n+1)$
1	x	x	1
0	1	0	1
0	1	1	0
0	0	x	$Q_B(n)$

Fig. 6. Circuit implementation for clock generation at each stage.

The worst-case gate delay from each column/datapath logic stage is replicated and used as a representative in the canary circuit. This ensures that the generated clock pulse can satisfy the time requirement for each gate in that datapath bit. For the worst-case evaluation the devices in the canary circuit are initialized such that each stage flips the magnetization of the next one. An example of such a canary circuit is shown in Fig. 5 (a). It represents the canary circuit for a full adder. The 3-input majority gate generates the carryout. The generated carry out along with the inputs and carry-in facilitate the sum bit generation using the 5-input majority.

The principle is to propagate the signal in the canary chain from the MESO device U1 to U6 and flip the magnetization of every gate. A signal can be considered to have propagated through a MESO gate (e.g. U4), and its headers can be disabled when the magnetization of the *next* gate (U5) flips completely. However, magnetization cannot be sensed electronically. Hence to ensure that this device (U5) has completely flipped, the differential output voltage of the *following* device (U6) is monitored. Once the threshold (γ) is crossed, the U2 device can be safely switched OFF. The state machine for CLK(5) generation is shown in Fig. 5 (b).

This sequence is implemented using an active high-gated D Latch and comparators to compare the differential output voltages, as shown in Fig. 6. The simulation results for the canary circuit and the clock signals generated are shown in Fig. 7. The extra hardware required for the clock generation can be amortized over several logic stages running in parallel with the same clock and with same logic complexity (multiple bits in a word). The canary circuit tunes the clock pulse width based on each majority gate's logic size.

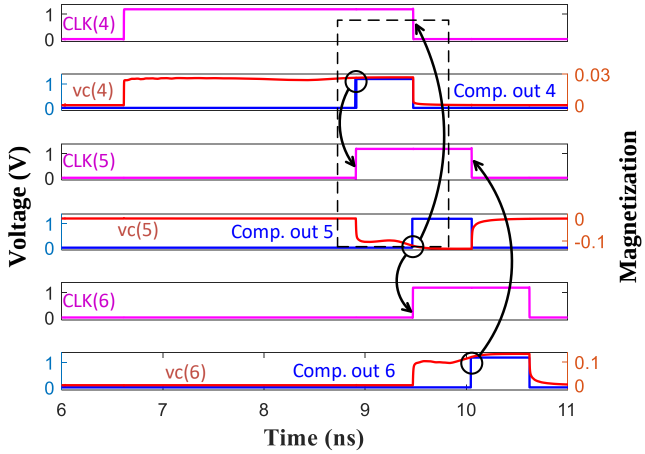


Fig. 7. Simulation results show the generated clock phases. Arrows show the triggers for CLK (5) generation. Here vc is the differential voltage at each output.

With the canary circuit the computation of Fig. 5 (a) would finish in 6.68 ns. Whereas in its absence it would take 26.77 ns (7-majority time*6 stages = 4.45*6 = 26.77 ns). The 7-majority gate time is assumed to be the worst-case to accommodate margin for variations. The total energy for the circuit of Fig. 5

(a) (including the digital auxiliary circuits, excluding the comparators) is 220.4 fJ. Without the canary circuit if each clock pulse is assumed to be of a 7-majority gate, the energy would be 341.5 fJ. It should be noted that without the canary circuit a clock pulse generator or clock tree would be required which usually consumes power in the mW range and involves significantly more design complexity to manage global timing skew.

Accounting for comparator energy would add to the canary circuit energy. A state-of-the-art comparator in 65 nm [12] consumes 30 fJ per conversion at a supply voltage of 1.2 V. An improvement of 5x can be assumed by scaling to 12 nm CMOS FinFETs. Moreover, since MESO devices support ultra-low voltages, reducing the supply voltage to 0.6V would be highly beneficial. Assuming a 6 fJ energy overhead per comparator would result in total energy of 220.4 fJ (canary) + 6*2*6 fJ (comparators) = 292.4 fJ for the entire canary circuit. Low voltage operation also enables inverter-based comparators [13] which have the potential to be lower in energy.

Table 4: Energy (amortized for 4 blocks of each) and Latency comparison for without and with canary circuit.

Components	Without Canary Circuit		With Canary Circuit	
	Energy (pJ)	Latency (ns)	Energy (pJ)	Latency (ns)
Full Adder	1.02	13.42	0.90	6.400
4-bit Ripple Carry Adder	4.32	22.29	3.72	14.13
4-bit Tree Multiplier	14.2	40.07	10.7	24.89

Table 4 compares the latency and energy of a canary based vs. non canary based implementation. The energy is amortized for 4 circuit blocks each. For example, the canary for a 4-bit ripple carry adder consumes 0.68 pJ (accounting for the energy of comparators). Each 4-bit ripple carry adder would consume 0.76 pJ using the clock pulses generated using the canary circuit. Therefore, the total energy for the canary-based approach is 0.68 pJ + 4*0.76 pJ = 3.72 pJ. Whereas, without the canary circuit, the 4-bit ripple carry adder would consume 4*1.08 pJ = 4.32 pJ.

IV. Arithmetic Logic Design

A. 4-bit Ripple Carry Adder

Having a majority gate as the fundamental logic unit allows generation of complex logic with a much lower number of devices.

Typically, it's observed that the carry bit is the bottleneck in any architecture of an adder. However, in the MESO technology carry is the output of a 3-input majority gate as used in [14].

$$C_{out} = 3 - Majority(A, B, C_{in})$$

$$S_{out} = 5 - Majority(A, B, C_{in}, \sim C_{out}, \sim C_{out})$$

A and B are the two inputs along with the input carry, C_{in}. Using the generated output carry bit and the three inputs, sum out (S_{out}) can be obtained by using a 5-input majority gate. The advantage with this design is that the carry bit can ripple through quickly and the corresponding sum bits can be generated later.

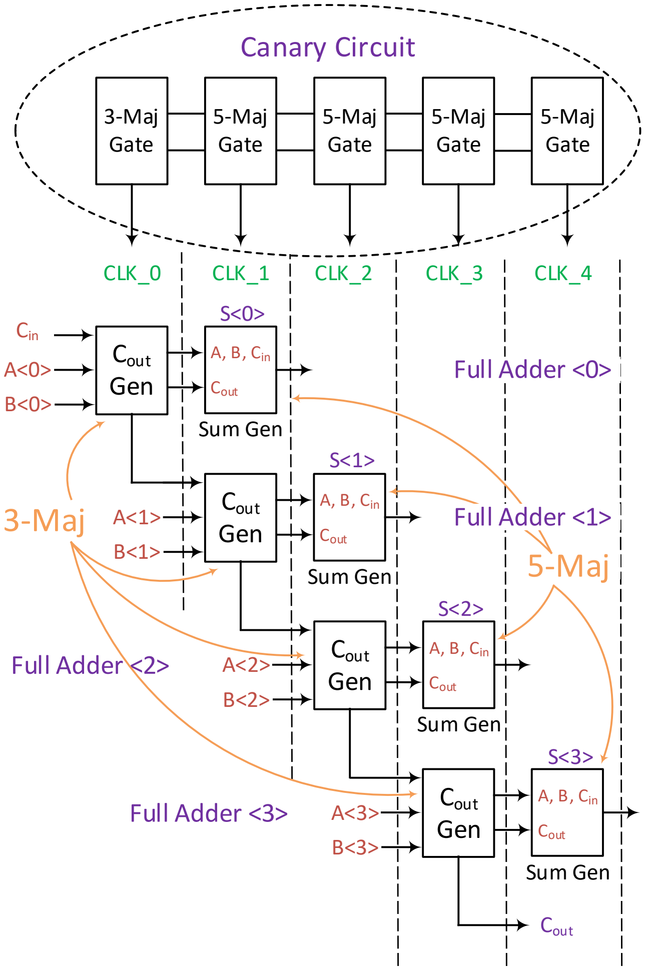


Fig. 8. Block diagram for the 4-bit ripple carry adder along with its canary circuit for clock generation.

Implementation of a 4-bit ripple carry adder is shown in Fig. 8. Carry generation for a full adder is just a 3-input majority gate. Since the carry generation is only a one-gate operation, the carry can ripple very quickly. The sum is generated by using the generated carry output and the inputs.

The circuit block diagram of the 4-bit ripple carry adder consists of 4 cascaded full adders. Each full adder includes a carry generation block (3-input majority gate) and a sum generation block (5-input majority gate). The carry-out generation for the next bit and the sum generation of the current bit happens in parallel.

The operation is as follows:

1) CLK_0 → Generate C_{OUT}<0> using A<0>, B<0> and C_{IN}.

2) CLK_1 → Use C_{OUT}<0>, A<1>, B<1> to generate C_{OUT}<1>. Parallelly using A<0>, B<0>, C_{IN}, ~C_{OUT}<0>, ~C_{OUT}<0> to generate SUM<0>.

Step 2 is repeated for the subsequent full adders enabling the rippling of carry.

The optimal design of a CMOS full adder requires 28 transistors. Comparing this to a MESO full adder which requires 4 devices for carry generation and 6 devices for sum generation. Accounting for the overhead of the header transistors, the MESO full adder requires 10 MESO devices and 10 transistors.

The 4-bit addition takes 5 clock phases to finish all the sum bits and final carry out generation. As seen in Fig. 8, each clock phase except CLK_0 has a sum generation block. The sum generation block is more complex than the carry generation block. (5-input majority vs. 3-input majority). Hence, the canary circuit for this adder will consist of 5-input majority gates for CLK_1 to CLK_4 and 3-input majority gate for CLK_0.

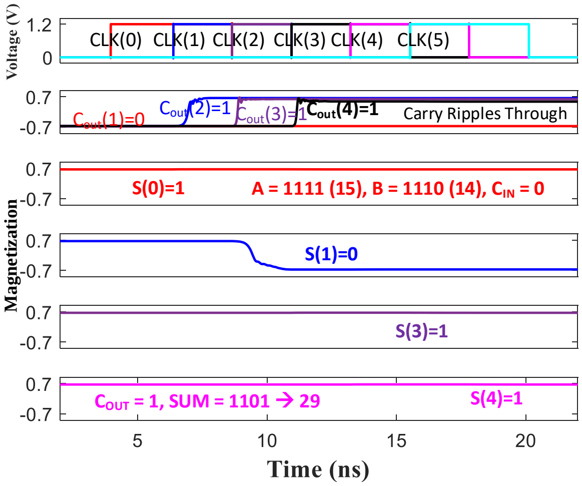


Fig. 9. Simulation result of the 4-bit ripple carry adder.

Fig. 9 shows the MESO circuit simulation result of addition of 1111(15) and 1110(14). The output matches the expected output of 15+14 = 29 (C_{OUT} = 1 and SUM = 1101).

B. *4-bit Tree Multiplier*

The compression tree for a 4-bit tree multiplier is shown in Table 5. The tree is compressed using the Wallace multiplier reduction scheme [15]. As seen in Table 4, the partial products must go through 2 stages of reduction before a 4-bit ripple carry adder in the final stage.

Table 5: Wallace Tree Compression for a 4-bit multiplier.

Weight	128	64	32	16	8	4	2	1
Wires	0	1	2	3	4	3	2	1
Pass		1	2	1	1			1
HA				1			1	
FA					1	1		
Wires	0	1	3	3	3	2	1	1
Pass		1					1	1
HA						1		
FA			1	1	1			
Wires	0	**2**	**2**	**2**	**2**	1	1	1

The circuit block diagram for the multiplier is shown in Fig. 10. The first stage is generation of partial products in CLK_0.

$$PP = 3 - Majority(A, B, '0')$$

The partial product is the AND operation on the two input bits. This is implemented in the MESO technology using a 3-input majority with one input set as ‘0’.

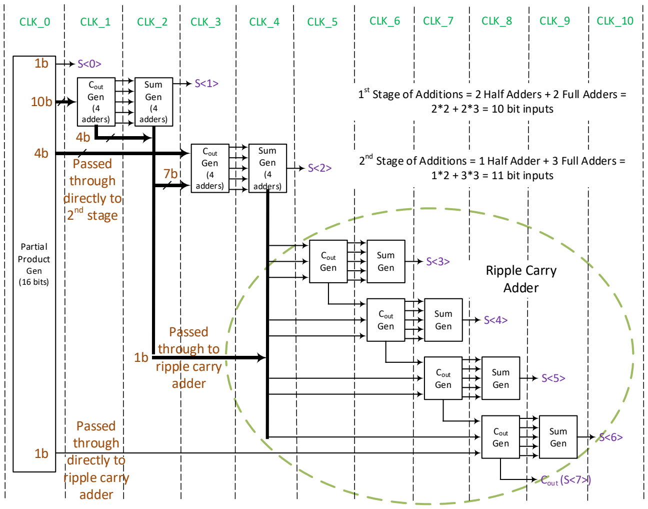

CLK_0
CLK_1
CLK_2
CLK_3
CLK_4
CLK_5
CLK_6
CLK_7
CLK_8
CLK_9
CLK_10
1b
S<0>
10b
C_{out} Gen (4 adders)
Sum Gen (4 adders)
S<1>
4b
4b
Passed through directly to 2nd stage
7b
C_{out} Gen (4 adders)
Sum Gen (4 adders)
S<2>
1st Stage of Additions = 2 Half Adders + 2 Full Adders = 2*2 + 2*3 = 10 bit inputs
2nd Stage of Additions = 1 Half Adder + 3 Full Adders = 1*2 + 3*3 = 11 bit inputs
Partial Product Gen (16 bits)
C_{out} Gen
Sum Gen
S<3>
Ripple Carry Adder
S<4>
S<5>
S<6>
1b
Passed through to ripple carry adder
1b
Passed through directly to ripple carry adder
C_{out} (S<7>)

Fig. 10. Block diagram of the 4-bit multiplication. Ripple carry adder for final addition after Wallace tree compression.

The two 4-bit inputs would result in 16 partial products being generated. Once the partial products are ready, CLK_1 to CLK_4 are used for the two stages of reduction shown in the Wallace tree. The least significant partial product is passed on to the output directly. The next 10 bits of partial products are used in the first stage of the Wallace tree. The outputs from this stage and the next 4 bits of partial products are used in the second stage of the Wallace tree. Further, once the ripple carry adder inputs are generated, pulses - CLK_5 to CLK_9 are used to perform the final ripple addition. At the end of CLK_9 all the multiplication outputs are available.

The canary circuit is used to generate the clock phases for multiplication operation. Fig. 11 shows the MESO circuit simulation result for a multiplication of A = 1101 (13) and B =

1011 (11). The output matches the expected result = 10001111 (143).

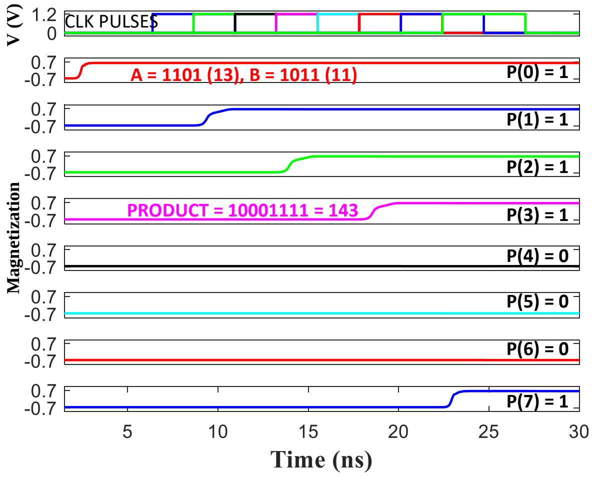


Fig. 11. Simulation result for multiplication operation.

V. Memory Design & In-Memory Computation

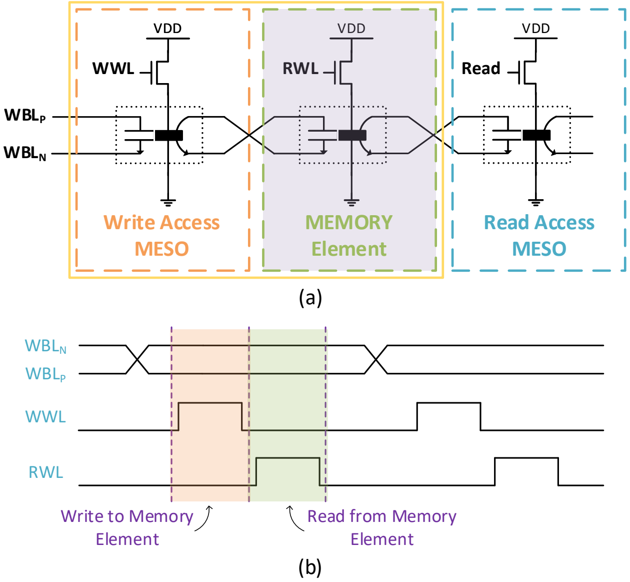


Fig. 12. Memory Design (Read and Write mode)

The magnetization in the MESO device ferromagnet is retentive [3, 4] and acts as the state output that can be monitored. Accordingly, a memory can be organized as shown in Fig. 12 (a).

The structure makes use of differential bit-lines, write word-lines, and read word-lines. Each memory element is preceded by a write access device. A read access MESO device is connected to a column of memory elements as can be seen in an example of in-memory addition of Fig. 13. Compared to an SRAM memory bank, MESO technology does not require any sense amplifier. Fig. 12 (b) shows the timing information of signals WBL, WWL and RWL required for writing and reading from memory.

WRITE: WBL enables the input data to be written to write access MESO → WWL = ON.

READ: RWL = ON → Read Access MESO's magnetization is programmed with the data.

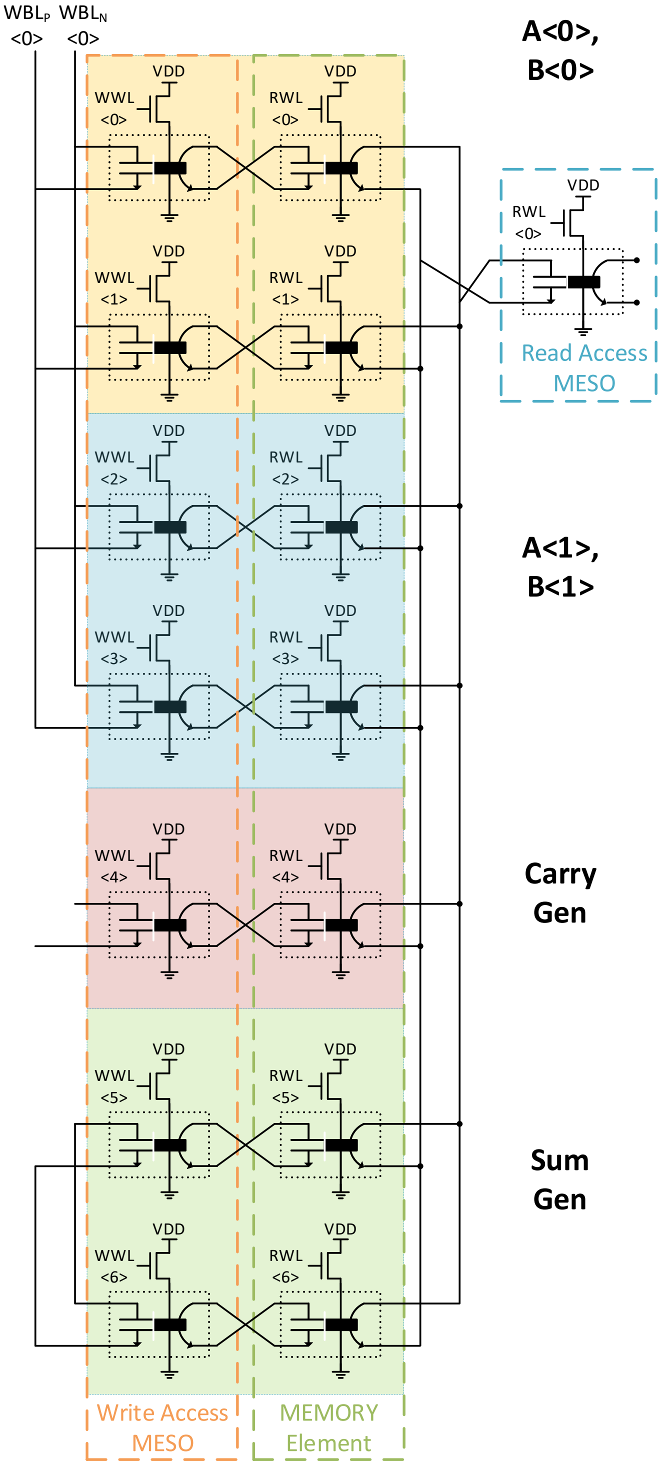


Fig. 13. Memory structure enabling in-memory addition operation.

A. *In-memory Addition*

This memory organization enables in-memory computation. Multiple memory elements can be enabled simultaneously to obtain a majority gate operation at the bit-line output of the read access MESO.

The in-memory compute operation for 2-bit addition is shown in Fig. 13. The steps for the in-memory addition are as follows:

1) Write inputs A<1:0> and B<1:0> in the memory WWL<0:3> (Clock count = 4).

2) A<0> and B<0> are used to generate carry in the WWL<0> and WWL<1> devices. (Clock count = 5).

3) Write COUT<0> in write access devices to prepare for SUM<0> generation (Clock count = 6).

4) Write COUT<0> in the memory for SUM<0> generation (Clock count = 7).

5) Generate SUM<0> (Clock count = 8).

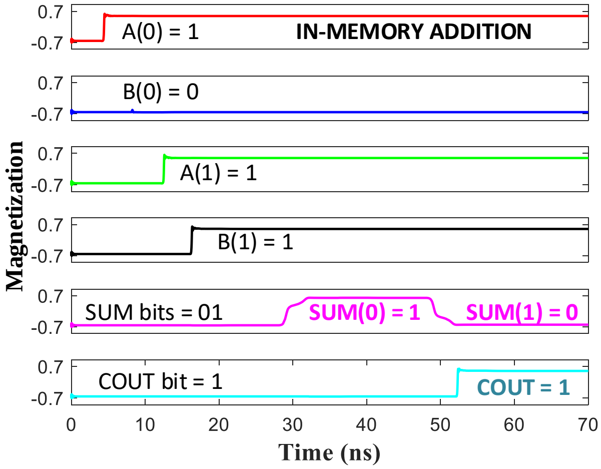


Fig. 14. In-memory addition simulation result. Here A = 11 (3), B = 10 (2) resulting in output sum = 5.

The inputs are written in the memory in step 1 (takes 4 clock cycles). Steps 2 to 5 explain the generation of SUM<0>. Steps 2 to 5 are repeated to generate SUM<1>. An extra transition clock cycle is required at the end to prepare for the next computation. Hence, it takes a total of 13 clock phases to perform the in-memory addition. Fig. 14 shows the simulation result for the in-memory addition. The inputs used are A = 11 (3) and B = 10 (2). The output matches the expected output of 5 with SUM = 01 and Carry Out = 1. This series of steps can be performed in parallel in multiple memory columns leading to a high throughput.

B. *In-memory multiplication*

For the in-memory multiplication the read access MESO devices are also used hierarchically to provide flexibility and parallelism.

The read devices are further used in a hierarchical fashion to enable complex operations. This is another advantage of the MESO memory fabric that allows for hierarchical construction which allows growth in a multiplicative fashion. For the circuit example of Fig. 15, the $READ_1$ device is capable of reading from 15 memory elements directly (via $READ_{01-04}$) or after a majority gate operation on this data. The circuit block diagram is shown in Fig. 15. for 2-bit multiplication. The in-memory multiplication is achieved by obtaining the product in a bit-serial fashion [16].

The step-by-step operation for the in-memory multiplication is as follows:

1) P_0 = AND (A_0, B_0)

2) P_1 = SUM_OUT (A_0B_1, A_1B_0). The carry out ($Carry_1$) is passed to the next step.

3) P_2 = SUM_OUT (A_1B_1, $Carry_1$).

4) P_3 = $Carry_2$.

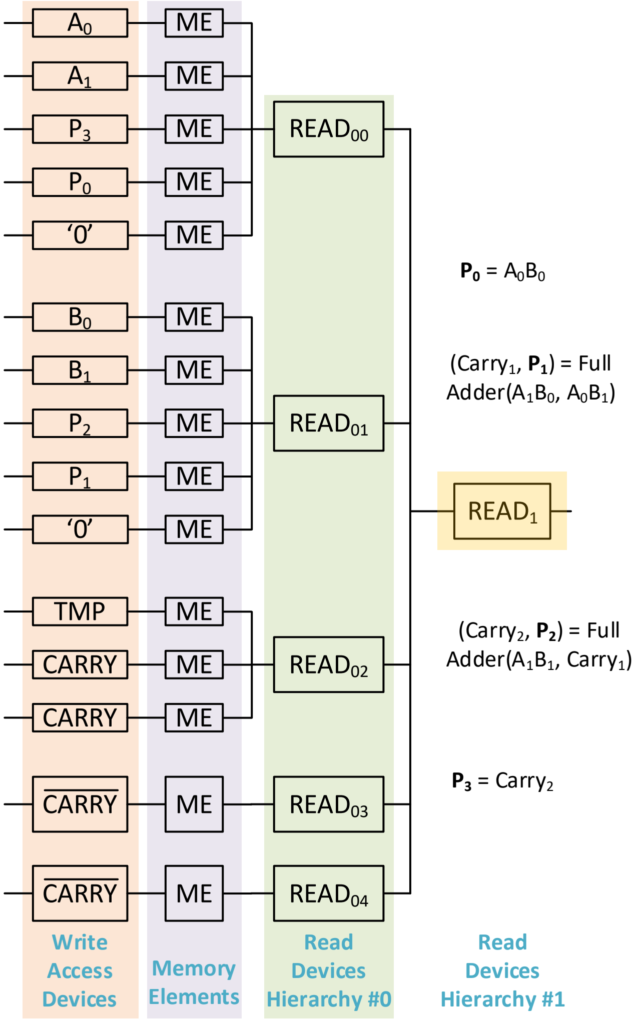

A_0
A_1
P_3
P_0
'0'
B_0
B_1
P_2
P_1
'0'
TMP
CARRY
CARRY
$\overline{\text{CARRY}}$
$\overline{\text{CARRY}}$
ME
$READ_{00}$
$READ_{01}$
$READ_{02}$
$READ_{03}$
$READ_{04}$
$READ_1$
$P_0 = A_0B_0$
$(Carry_1, P_1)$ = Full Adder(A_1B_0, A_0B_1)
$(Carry_2, P_2)$ = Full Adder$(A_1B_1, Carry_1)$
$P_3 = Carry_2$
Write Access Devices
Memory Elements
Read Devices Hierarchy #0
Read Devices Hierarchy #1

Fig. 15. Memory structure enabling in-memory multiplication operation by using multi-level hierarchy.

The in-memory multiplication can be easily extended to arbitrary bit-width in a bit-serial fashion. Furthermore, as described in [16], an in-memory CMOS multiplication needs 148 transistors per multiplication operation. However, the proposed architecture using MESO devices reduces the number of devices to 36 transistors and 36 MESO devices.

Fig. 16 shows the simulation result for the in-memory multiplication. The inputs used are A = 11 (3) and B = 10 (2). The output matches the expected output of 6 with P = 0110. These steps can be performed in parallel in multiple memory columns leading to high throughput.

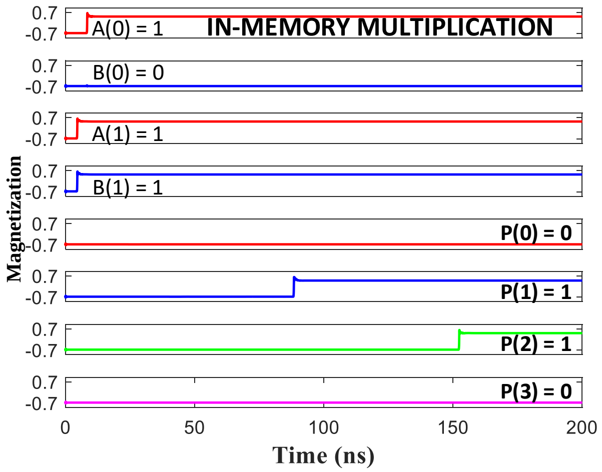


Fig. 16. In-memory multiplication simulation result. Here A = 11 (3), B = 10 (2) resulting in output product = 6.

VI. Conclusions

MESO technology presents unique circuit opportunities to implement logic functions operating at ultra-low supply voltage that enables significant energy efficiency improvements. Using a circuit simulation device model that has been verified with physics simulation [2, 10], this paper presented the use of device stacking to reuse power supply current, canary clocking to minimize current, and in-memory computing exploiting the state retention and functional operation of read-out innate to the MESO structure and capitalizing on its capabilities. This will further enable the ever-growing need of energy efficient computation required for general purpose compute and artificial intelligence.

VII. Acknowledgement

This work is supported by the Intel Science and Technology Center (ISTC) - Valleytronics.

References

[1] D. E. Nikonov and I. A. Young, "Overview of Beyond-CMOS Devices and a Uniform Methodology for Their Benchmarking," in Proceedings of the IEEE, vol. 101, no. 12, pp. 2498-2533, Dec. 2013, doi: 10.1109/JPROC.2013.2252317.

[2] H. Li, C. -C. Lin, D. E. Nikonov and I. A. Young, "Differential Electrically Insulated Magnetoelectric Spin-Orbit Logic Circuits," in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, vol. 7, no. 1, pp. 18-25, June 2021, doi: 10.1109/JXCDC.2021.3105524.

[3] D. C. Vaz et al., "Functional Demonstration of a Fully Integrated Magneto-Electric Spin-Orbit Device," 2021 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2021, pp. 32.4.1-32.4.4, doi: 10.1109/IEDM19574.2021.9720677.

[4] P. Debashis et al., "Low-voltage and high-speed switching of a magnetoelectric element for energy efficient compute," 2022 International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2022, pp. 36.4.1-36.4.4, doi: 10.1109/IEDM45625.2022.10019505.

[5] S. Manipatruni, et al. "Scalable energy-efficient magnetoelectric spin–orbit logic." Nature 565.7737 (2019): 35-42.

[6] D. E. Nikonov and I. A. Young, "Benchmarking of Beyond-CMOS Exploratory Devices for Logic Integrated Circuits," in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, vol. 1, pp. 3-11, Dec. 2015, doi: 10.1109/JXCDC.2015.2418033.

[7] E. Saitoh *et al.*, ‘‘Conversion of spin current into charge current at room temperature: Inverse spin-Hall effect,’’ *Appl. Phys. Lett.*, vol. 88, no. 18, p. 182 509–1–182 509–3, 2006.

[8] J.-C. Rojas-SÆnchez *et al.*, ‘‘Spin-to-charge conversion using Rashba coupling at the interface between non-magnetic materials,’’ *Nature Commun.*, vol. 4, Dec. 2013, Art. no. 2944.

[9] Y. Shiomi *et al.*, ‘‘Spin-electricity conversion induced by spin injection into topological insulators,’’ *Phys. Rev. Lett.*, vol. 113, no. 19, 2014, Art. no. 196601.

[10] H. Li et al., "Physics-Based Models for Magneto-Electric Spin-Orbit Logic Circuits," in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, vol. 8, no. 1, pp. 10-18, June 2022, doi: 10.1109/JXCDC.2022.3143130.

[11] S. K. Maurya and L. T. Clark, "Fast and scalable priority encoding using static CMOS," Proceedings of 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 2010, pp. 433-436, doi: 10.1109/ISCAS.2010.5537688.

[12] H. S. Bindra, C. E. Lokin, A. -J. Annema and B. Nauta, "A 30fJ/comparison dynamic bias comparator," ESSCIRC 2017 - 43rd IEEE European Solid State Circuits Conference, Leuven, Belgium, 2017, pp. 71-74, doi: 10.1109/ESSCIRC.2017.8094528.

[13] Meng-Tong Tan, J. S. Chang and Yit-Chow Tong, "A process-independent threshold voltage inverter-comparator for pulse width modulation applications," ICECS'99. Proceedings of ICECS '99. 6th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.99EX357), Paphos, Cyprus, 1999, pp. 1201-1204 vol.3, doi: 10.1109/ICECS.1999.814385.

[14] A. Jaiswal, A. Agrawal and K. Roy, "Robust and Cascadable Nonvolatile Magnetoelectric Majority Logic," in IEEE Transactions on Electron Devices, vol. 64, no. 12, pp. 5209-5216, Dec. 2017, doi: 10.1109/TED.2017.2766570.

[15] C. S. Wallace, "A Suggestion for a Fast Multiplier," in IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14-17, Feb. 1964, doi: 10.1109/PGEC.1964.263830.

[16] J. Wang et al., "A 28-nm Compute SRAM With Bit-Serial Logic/Arithmetic Operations for Programmable In-Memory Vector Computing," in IEEE Journal of Solid-State Circuits, vol. 55, no. 1, pp. 76-86, Jan. 2020, doi: 10.1109/JSSC.2019.2939682.