A Neoteric Approach for Logic with Embedded Memory Leveraging Crosstalk Computing

Prerana Samant, Naveen Kumar Macha*, Mostafizur Rahman
Department of Computer Science & Electrical Engineering, University of Missouri Kansas City, MO, USA
E-mail: pbstfd@mail.umkc.edu , rahmanmo@umkc.edu

Abstract— **One of the essential elements of computing is the memory element. Flip-flops form an integral part of a System-on-Chip (SoC) and consume most of the area on the die. To meet the high-speed performance demands by the data-intensive applications like artificial intelligence, cloud computing, and machine learning, we propose to integrate memory with the logic to get built-in memory Logic circuits that operate on the crosstalk computing logic. These circuits are called Crosstalk Built-in Memory Logic (CBML) circuits which exploit the detrimental interconnect crosstalk and astutely turn this unwanted effect into a computing principle with embedded memory. The logic output of complex circuits is retained irrespective of the change in input until the next evaluation cycle. This neoteric embedding of memory in logic provides high-speed operation with a reduced number of transistors. In this paper we have manifested by experimental evidence, the in-built memory feature of the complex CBML circuits using 16 nanometer (nm) PTM models in HSPICE. Benchmarking is done with the equivalent CMOS circuits to compare the number of transistors, power, and performance. It is observed that the number of transistors consumed by CBML 4-bit Full-Adder, an example of a large CBML circuit, is up to 32% less, and performance is improved by 27% than the equivalent CMOS circuits. It can be used in ALU for implementing a counter or an adder circuit. The performance improvement achieved by 3-input AND and the CARRY logic is up to 60% along with a 20% reduction in the number of transistors. CBML circuits have potential to pave the way for special high-speed macros with specifically engineered strctures.**

Keywords—Crosstalk, Built-in Memory, Flip-Flop (FF), Crosstalk Logic (CL), PMOS, NMOS, Complementary MOSFET (CMOS), Full-Adder

I. Introduction

The advent of artificial intelligence, deep learning, and cloud computing has given rise to the need for System-on-Chip (SoC) and multi-core processors which require a high degree of parallelism. One way to achieve it is by exploiting architectural pipelining. Pipelining consumes excessive hardware due to the use of additional registers/flip-flops employed to shorten the critical path. An SoC consists of mostly combinational circuits and sequential circuits in the form of standard cells. The sequential circuits like registers, flip-flops, and latches, form the basic building block of pipelined architecture. Flip-flops hold the state of the output high or low depending upon the input and change its state only when there is a change in the state of the clock. Flip-flops (FF) consume almost 50% of the area and power on an SoC (System-on-Chip) and hence its optimum design is of crucial importance. Various D-Flip-Flop architectures have been proposed and designed to improve performance and area like the Semi-dynamic FF, Pulse-Triggered FF, Sense Amplifier based FF (SAFF), Topologically Compressed FF (TCFF), Logic structure Reduction Flip-Flop (LRFF), Dual dynamic node FF (DDFF), etc. compared to the traditional Transmission Gate FF (TGFF) [1-9]. But, very few architectures have flip-flops with in-built logic. Integrating logic into memory or vice versa is the latest trend to achieve high performance and reducing latency. The existing architectures have issues like charge sharing and latency. They do not discuss embedding complex logic like full adders into the flip-flops. In this regard, we propound to combine the combinational logic with area-efficient memory elements to get combinational circuits with a built-in flip-flop/register. The concept of the Crosstalk Logic with built-in memory was discussed in [10].

In this paper, we elaborate on the Crosstalk Built-in Memory Logic Circuits (CBML) to build complex circuits like 3-input AND, OR, NAND, NOR, AO21, OA21, and 4-bit adder using cascaded CL. The Crosstalk Logic (CL) gates form the premise for the CBML circuits. We have justified in [11-18] that at small process nodes, the detrimental interference between the interconnects with close proximity, called crosstalk, can be astutely utilized to implement logic functions. These circuits are leveraged to implement built-in memory which can store the output state of the logic. A memory enabler circuit is used which contributes to the enabling of memory feature. The CMOS logic circuits and transmission-gate-based D-flip-flop is used as a baseline for CBML circuits.

II. Analysis of Existing Flip-Flop Architectures with Embedded Logic

Various flip-flop architectures that have been designed and proposed are classified as static and dynamic designs. Few use the hybrid style employing advantages of both static and dynamic styles. The transmission gate-based D-FF (TGFF) [1], which is the most widely used FF, uses 22 transistors and is a static flip-flop. The clock has a capacitive load of 4 transistors. It faces a clock overlap issue and low performance due to greater D-to-Clock delay. The Semi-dynamic Single-Phase Pulsed FF (SDFF), with 26 transistors [2], is a type of dynamic flip-flop that uses a single phase of the clock and is an alternative solution for high-speed operations compared to TGFF. All pulsed flip-flops use narrow pulses derived from the clock locally. When the Clock is low, the internal node value is pre-charged, and output is held at the previous value. On the rising edge of the clock, the FF enters the evaluation phase for a period decided by the pulse and output Q sees the input D. Internal node is discharged due to its pre-charged nature. For the second half of the high clock where the pulse is low, the sampling of the input is disabled while the internal node and Q are retained. Here internal node is truly dynamic in nature. But, SDFF adds back-to-back inverters, buffers the output, and a conditional shut-off circuit to the pulsed FF to improve metastability and sensitivity to noise. Delicate Pulse width control and its distribution is the drawback of SDFF. Logic can be embedded by replacing the input transistor with NMOS pull-down only logic. Implementing complex embedded logic in it increases the NMOS stack and results in increased latency. The Implicit-Pulse Semi-dynamic (ip-DCO) FF [8] has reduced transistor count and has an 8%-10% reduction in D-Q delay than the SDFF. It offers negative slack due to transistor arrangement and pre-charging of the slack node but it has a high capacitive load to the internal node and degrades the overall performance. Implicit pulse-triggered FF with a pulse-control scheme in [3], has better performance than ip-DCO and SDFF but suffers from longer hold time and an increase in the number of transistors.

The Sense-Amplifier-based FF (SAFF) [4] utilizes 23 transistors. It has a sense-amplifier stage and a latch stage. This leads to a transistor overhead but has faster speed due to the absence of crowbar current. It has output noise immunity and can operate at high frequencies than the SDFF but does not have logic embedding capabilities. It has less setup time than the TGFF. The Topologically-Compressed FF (TCFF) in [5] uses an AND-NOR design base instead of a transmission-based design. It utilizes 21 transistors, a single-phase clock, has better performance due to low clock-to-Q delay. But for low-power applications, experiences high delay because of the weak pull-up transistors. The Logic Structure Reduction FF (LRFF) [6], an enhancement over TCFF, is a hybrid FF with static CMOS logic and complementary pass transistor logic. It has less transistor count i.e. 19, logic restructured to achieve shorter setup time, circuit simplification for power consumption reduction. The drawback of LRFF is greater hold time than TGFF and TCFF. The Dual Dynamic Node FF (DDFF) [7] embeds logic in it called as Embedded Logic Module (ELM). It abates the pipeline overhead and uses 21 transistors. The internal node is pull-up by the PMOS transistor and solves the charge sharing problem. The NMOS logic stack increases with an increase in inputs. An Embedded Logic FF (ELFF) introduced in [9] is a hybrid FF that combines logic functions with normal flip-flop operations is an improvement over DDFF. It has a 20% improvement in performance compared to the DDFF-ELM. It also dissipates less power than DDFF. It faces the same issues in terms of the NMOS stack for complex logic embedding.

III. Crosstalk Logic with Embedded Memory Feature

A. Basic Crosstalk Logic

Engineering the interconnect crosstalk to compute logic is demonstrated by us to cope with device scaling limitations and interconnect bottleneck [11]. Fig. 1 shows the Crosstalk Logic (CL) circuits. The inputs, acting as aggressor nets, upon transition, induce a voltage on the victim net (V_i), proportionate to the mutual capacitance (CC) between the nets. The inverter conditions the V_i depending upon the threshold voltage (V_{TH}) and decides its logic level. If $V_i < V_{TH}$, V_i is at Logic 0, and if $V_i > V_{TH}$, V_i is at Logic 1. CL executes its logic in the evaluation state (ES) and requires a discharged state (DS) to bring V_i in its initial floating state. For a Positive CL (PCL), V_i is initialized to Logic 0 and for a Negative CL (NCL), V_i is initialized to Logic 1. Fig. 1.a. shows that PCL requires low to high transition on the inputs whereas Fig. 1.b shows that NCL requires high to low transition for the operation of crosstalk circuits. These inputs are initialized by the previous stage gates or using special initializer circuits [16]. The CC and the V_{TH} decides the type of logic implemented. Combination of PCL and NCL along with memory enabler circuit forms CBML circuits.

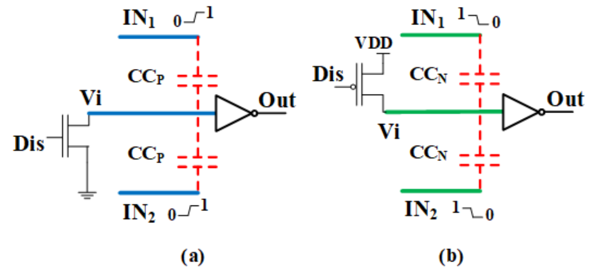

Fig. 1 (a) Positive Crosstalk Circuit (b) Negative Crosstalk Circuit

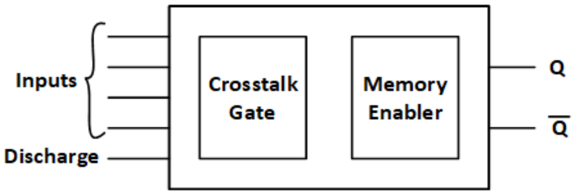

Fig. 2. Logic with Built-in Memory Concept

B. Integrating Memory with Logic

The previous papers discussed in Section II show that logic is implemented as a part of a flip-flop circuit. In this paper, we propose to embed memory with the crosstalk logic to implement functions such as NAND, NOR, AND, OR, etc., which can retain the output state. The concept of logic with built-in memory is shown in Fig. 2. Here, the Crosstalk block consists of PCL and NCL connected in parallel known as Dual CL (DCL) as shown in [16]. The Memory Enabler (ME) circuit discussed in [10] is a special circuitry that provides inherent memory to the crosstalk circuits. The output Q is retained until it receives a Discharge (Dis) cycle. Here, Dis acts like a reset signal. The ME in PCL and NCL consists of pull-up (PU) and pull-down (PD) transistors. The voltage is induced on the victim net V_i as mention in section III-A. The ME pulls up or down the V_i depending upon the logic at the output of the inverter. This statical PU and PD of V_i, reduces its leakage and also retains the output regardless of the change of inputs in the evaluation phase. The PU and PD are disabled in the discharge state. Fig. 3.a&b shows a PCL and NCL with ME. The output of the PCL inverter (Fp) is fed back to the victim net V_i of both PCL and NCL through PMOS P2 whereas the output of NCL (Fn) is given as feedback through NMOS N2 to both PCL and NCL. Depending upon the Fp and Fn, the V_i is pulled up or down. When $Dis = 1, \overline{Dis} = 0$, PMOS P1 and NMOS turns off, PU-PD is disabled, and V_i is

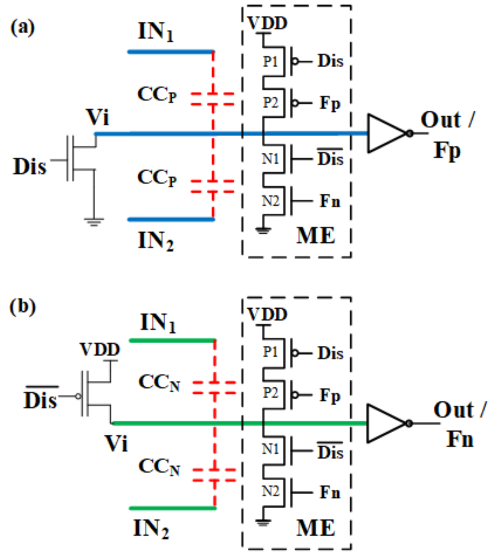

Fig. 3. Memory Enabler circuit integrated with (a) PCL (b) NCL

in its initial floating state. For a particular logic, PCL and NCL are connected in parallel, and Fp and Fn will have the same logic level in the evaluation phase. They differ only in the discharge state because of the different initial states set by the discharge transistors. Proper selection of V_{TH} and accurate transistor sizing enables a built-in memory feature in CBML circuits. Without requiring redundant Master-Slave latches as in the case of traditional Flip-Flops, the CBML circuits offer a novel kind of edge-sensitive memory with an added benefit of computation in memory. Few implementations of CBML logics are presented in the next section.

IV. Crosstalk Logic Flip-Flop Circuits

A. CBML Complex Logic Gates

The Crosstalk circuits are identical in nature if the number of inputs is the same. As the inputs increase, the number of aggressor nets increases accordingly. The 2-input NAND, NOR, AND, OR, and one-bit full adder CBML are discussed in our paper [10]. In this paper, we use the CBML concept to implement 3-input complex circuits. The functionality of the memory block remains the same for all logic circuits. The 3-input AND, OR, AO21, OA21, and, CARRY-3 ($AB + BC + AC$) logic with embedded memory is implemented by using the 3 inputs as aggressor nets and a mutual capacitance for each of the three inputs. The circuit diagram of the 3-input circuits is shown in Fig. 4. The victim nodes are initialized when $Dis = 0$. As discussed in Section III B, the PCL and NCL execute their logic in parallel when $Dis = 0$. The first inverter output of the duals is fed back to statically pull up the V_i when Fp and Fn are at logic 0 and pull down the V_i when Fp and Fn are at logic 1. When the transition of the input happens, due to the mutual capacitances, voltage is induced on the V_i. The values of CC_P, CC_N, and the ratio of the size of PMOS to NMOS ($P{:}N$) of the first inverter determines the logic implemented. The value of K represents the multiple the of the mutual capacitance required for the third input. Table I indicates the required transistor sizing and the coupling capacitances. For instance, for an AND gate, only when all three inputs transition, an equivalent voltage greater than the V_{TH} of the NMOS of the first inverter is induced on V_i and logic 1 is obtained at the output. Output is at Logic 0 when either of them is zero. The number of transistors required for the circuits remains the same as that of the 2-input CBML circuits.

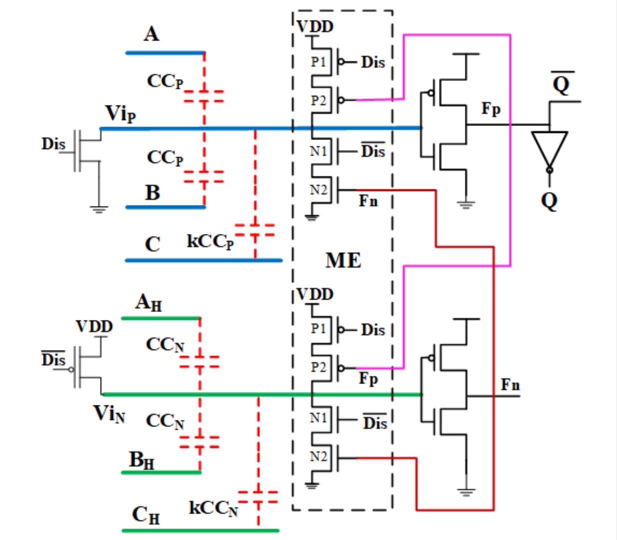

Fig. 4. CBML circuit for 3-input complex gates

TABLE I. Crosstalk Coupling and Transistor Sizing for 3-Input Gates

CL	Gate	CC (aF)	K	Pull-up/Pull-Down Logic				Width Ratio (P:N)
				P1	P2	N1	N2	
PCL	AND3	280	1	3	3	1	1	1:1
	OR3	8000	1	1	2	1	2	1:5
	AO21	600	2	1	2	3	3	1:2
	OA21	500	2	2	3	1	2	1:1
	AB+BC+AC	600	0.67	1	1	3	3	1:2
NCL	AND3	4000	1	1	1	3	3	5:1
	OR3	300	1	1	2	1	2	3:1
	AO21	600	2	1	2	3	3	1:2
	OA21	500	2	2	3	1	2	3:1
	AB+BC+AC	600	0.67	1	2	2	2	2:1

The simulation results for the AO21 ($AB + C$) and OA21 ($A + B$) C are shown in Fig. 5. It can be seen that logic is executed in the evaluation phase (E) and resets in the discharge phase (D). The inputs and output are initialized to zero. The logic with embedded memory feature can be observed for input combination 011 in the simulation. At 18ns irrespective of change in inputs to 000, the output retains the logic 1 for the entire evaluation cycle. It only resets after the

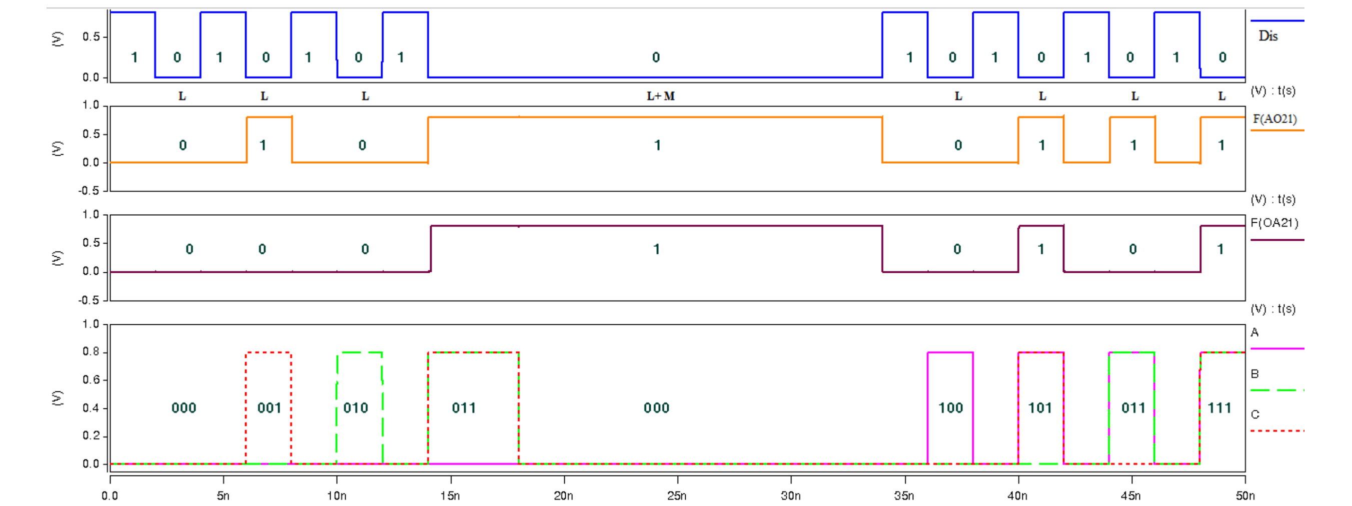

Fig. 5. Simulation of AO21 and OA21 CBML Circuit

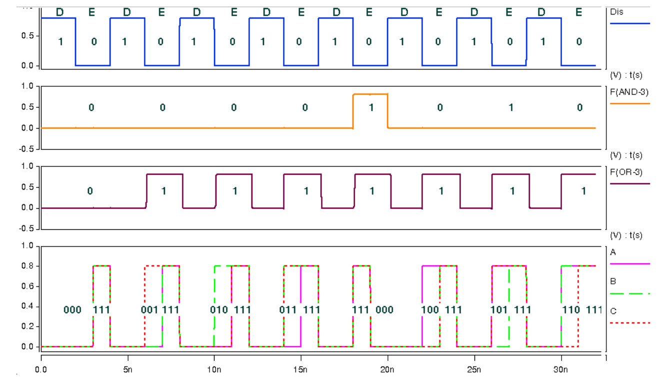

Fig. 6. Simulation of AND-3 and OR-3 CBML Circuit

discharge cycle at 34ns. Likewise, for all input combinations, memory is built-in in the logic. It should be noted that $\bar{Q}$ implements AOI21 ($\overline{AB+C}$) and the OAI21 $\overline{(A+B)\,C}$) logic respectively and also acts as feedback for both PCL and NCL circuits.

Fig. 6 shows simulation results for 3-input AND (AND-3) and OR (OR-3) logic. To verify the memory feature, the inputs are changed after 1ns for each evaluation phase. The change in inputs does not impact the output as it holds the previous state until the discharge state. The output retains only that logic level which it attains at the beginning of the evaluation cycle. For instance, at 19 ns, the inputs are changed from 111 to 000, it can be seen in the simulation results that the AND-3 and OR-3 outputs retain their logic 1 state for the entire evaluation cycle. When $Dis = 1$, the circuit is ready for the next set of inputs. This validates the CBML concept. NAND-3 and NOR-3 outputs are obtained from the $\bar{Q}$.

B. *CBML 4-bit Full-Adder (FA)*

The CBML circuits can be cascaded like the CMOS circuits. The previous stage circuits ensure the transitions on the inputs for the next stage. Few cascading issues and additional initializer circuits are discussed in [16]. Input high initializers (IHI) ensure that the next stage PCL receives 0-to-1 transitions on the inputs whereas the Input low initializers (ILI) ensure that the next stage NCL receives 1-to-0 transition. The logic simplification algorithm for the crosstalk circuits is elaborated in [18]. This algorithm can be used to implement circuits that are not crosstalk-friendly. Implementing a non-linear function like an EXOR or an EXNOR gate requires an additional control signal to exercise the threshold voltage as per the crosstalk logic. A 4-bit full-adder is designed using CBML circuits to verify the working of cascading of complex circuits. Fig. 7 shows the CBML 4-bit full-adder block diagram. Each block has an internal circuit diagram of a 1-bit full-adder as per Fig. 8. The circuit for the CARRY is the same as that discussed for a 3-input CBML. The control signal for the SUM logic is generated by connecting the feedback signals Fp and Fn to the victim net V_i of the SUM logic through a capacitance value twice to that of the mutual capacitances of the SUM CBML circuit (CC_S). This signal having twice capacitance has more control over the V_i and the EXOR operation is executed. The values of the mutual capacitances for the CARRY ($AB + BC + AC$) logic are as per Table I and the values for SUM logic are mentioned in Table II. The number of transistors consumed is 32% less than the equivalent CMOS circuit as discussed in the Section V. Carry-output of the first stage goes as an input to the carry-input of the second stage just like a ripple-carry adder. The simulation results are shown in Fig.9. The SUM and CARRY output continues to be in the same state even after the inputs change in the evaluation cycle. The previous stage carry outputs are also initialized to zero during discharge cycle, hence next stage automatically receives the appropriate transitions. A 4-bit FA is an essential component of Arithmetic Logic Unit (ALU). This CBML FA can be used where an adder or a counter with a storage register is required. It can significantly minimize the number of required transistors and boost performance as discussed in Section V.

TABLE II. CROSSTALK COUPLING AND TRANSISTOR SIZING FOR SUM

Crosstalk Logic	CC_S (aF)	Pull-up/Pull-Down Logic				Width Ratio (P:N)
		P1	P2	N1	N2	
PCL	500	1	1	3	3	1:2
NCL	500	1	1	3	3	2:1

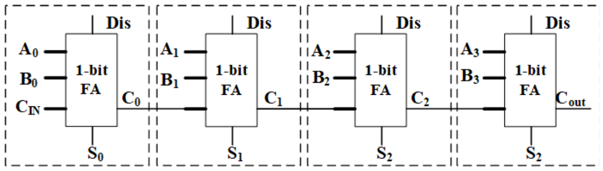

Fig. 7. Cascaded 4-bit CBML full adder block diagram

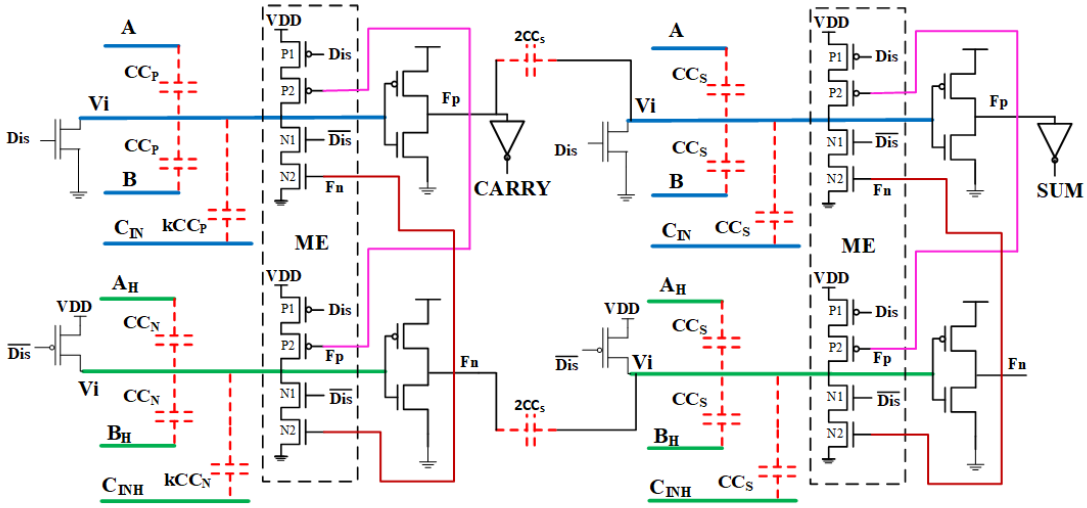

Fig. 8. 1-bit Full-Adder circuit diagram

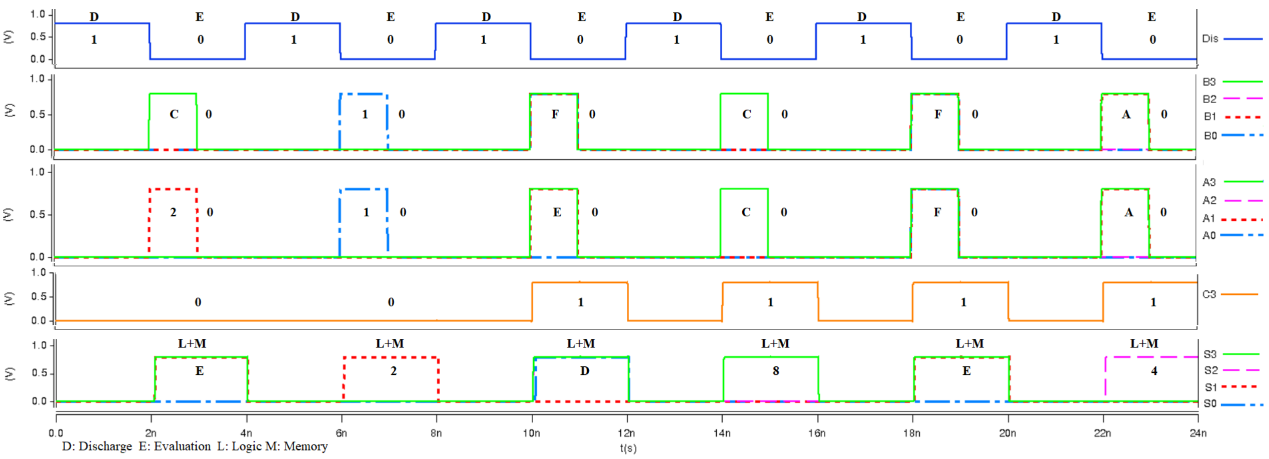

Fig. 9. Simulation of 4-bit CBML Full Adder

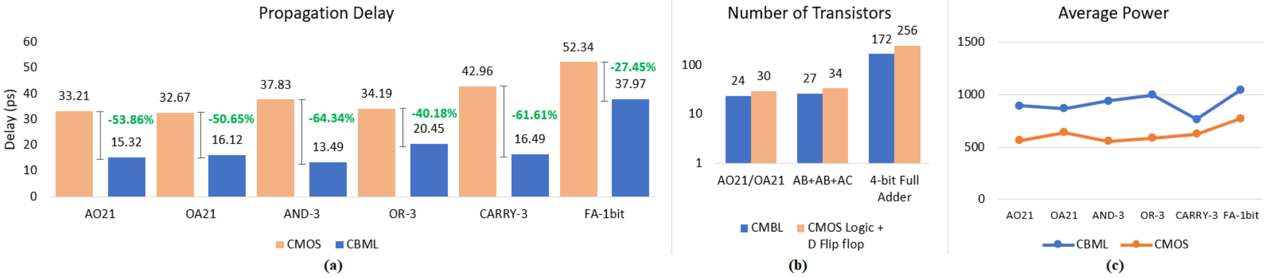

Fig. 10. (a) Comparison of Propagation delay (b) Comparison of Number of Transistors (c) Average Power Comparison

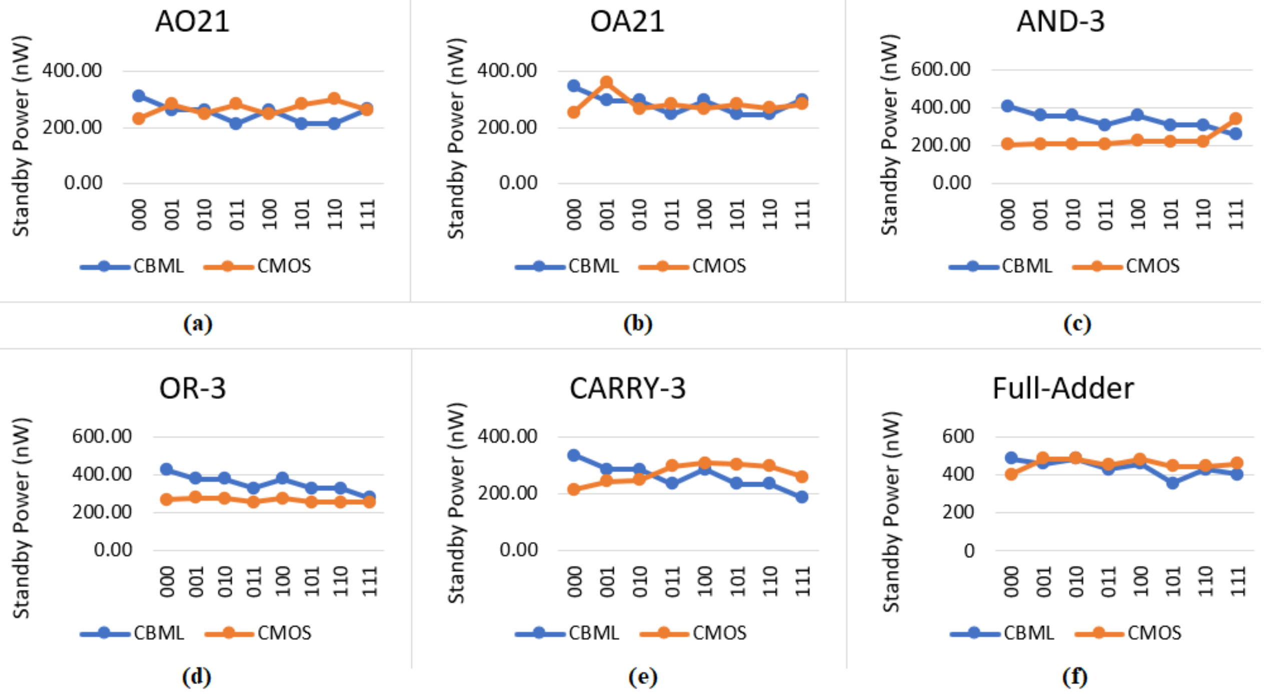

Fig. 11. Standby/Leakage Power for (a) AO21 (b) OA21 (c) AND-3 (d) OR-3 (e) CARRY-3 (f) Full-adder

V. Analysis and Comparison

The CBML circuits do not have setup time or hold time constraints since the basis of crosstalk circuits is the necessity of the transitions on the inputs. But it does have a transition time constraint. The transition time (T_t) is defined as the maximum time from the falling edge of the discharge signal ($Dis = 0$) within which the inputs must transition to get the intended logic. If the inputs change after the transition time, they are not captured by the victim net and the V_i will have incorrect induced voltage and will result in an incorrect output logic state. T_t is calculated from the HSPICE simulations of each circuit by increasing the time between the falling edge of Dis and 0-to-1 transition on the inputs in the evaluation cycle. For AO21 CBML, in Fig. 5, at 40 ns when the discharge (Dis) goes from high to low, the inputs can change from 000 to 101 up to 40.0221 ns. If they change after this time, logic 1 will not be induced on V_i and it will remain at Logic 0. Thus, T_t for AO21 CBML is 22.1 ps. Similarly, transition time (T_t) for other OA21 is 12.71 ps, AND-3 is 11.51 ps, OR-3 is 18.66 ps, CARRY is 17.6 ps and for 1-bit full-adder (FA) SUM output is 31 ps. The SUM logic has more transition time because of the control signal used for its computation as discussed in Section IV.B. The 4-bit FA has the same transition time for its blocks.

The CMOS circuits along with the transmission-gate-based D-Flip-Flop (D-FF) are used for benchmarking of the CBML circuits. For a fair comparison, the logic circuits are implemented with CMOS transistors and the logic output is provided as an input to the D-FF. The delay for the CBML circuit is calculated by measuring the difference in time when output transitions (50%) after the transition of inputs (50%) for all combinations of the inputs. The maximum of the delay is considered as the propagation delay. The delay for CMOS circuits is the summation of propagation delay from input to output of the respective logic, the setup time of D-FF, and clock-to-Q delay. The CBML circuits show almost 50% improvement in performance. The detailed comparison is shown in Fig. 11.a. The propagation delay of the CBML circuits is reduced up to 64% for 3-input AND gate. Hence, CBML can operate at higher speeds. The CBML circuits consume fewer transistors compared to the CMOS circuits as shown in the graph of Fig. 11.b. The 4-bit full-adder requires 32% fewer transistors than the TGFF+CMOS logic circuits.

The power comparison is done in Fig. 12. Average power is calculated in HSPICE over the same time duration and applying the same input combinations for both CBML and CMOS circuits. The average power is more for CBML circuits as shown in the comparison in Fig. 12.a because of the high performance. The leakage power or standby power is calculated for all combinations of inputs by keeping them constant as shown in Fig. 12.b-f. The CBML circuits consume more power due to the transistor sizing of the PU-PD circuit and the required transitions on the inputs but the values are comparable with the CMOS circuits.

VI. Conclusion

The Crosstalk Built-in Memory Logic (CBML) circuits utilize the crosstalk phenomena to embed memory in the high-performance logic operations. They combine the feature of logic and flip-flop. This alleviates the number of transistors consumed by the circuits compared to other architectures as shown by us in [10]. The reduction is up to 32% for a 4-bit full-adder. The simulation results depict performance improvement of up to 64% for a 3-input AND logic and up to 27% for a 4-bit FA. This neoteric 4-bit FA can be deployed in an ALU for integrating adder/counter and register storage. There is some power tradeoff for achieving the performance boost but the analysis shows the power dissipation results comparable with the CMOS circuits. Finally, the CBML circuits broaden the opportunities in accomodating more hardware on an SoC with better performance.

References

[1] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2003.

[2] Klass F., et al., “New Family of Semidynamic and Dynamic Flip-Flops with Embedded Logic for High-Performance Processors,” IEEE Journal of Solid-State Circuits, vol. 34, no. 5, May 1999

[3] Hwang Y.T., et al., “Low-Power Pulse-Triggered Flip-Flop Design With Conditional Pulse-Enhancement Scheme,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp. 361-366, Feb. 2012

[4] Strollo A., et al., “A Novel High-Speed Sense-Amplifier-Based Flip-Flop,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 13, no. 11, pp. 1266-1274, Nov. 2005

[5] Kawai. N, et al., A Fully Static Topologically-Compressed 21-Transistor Flip-Flop with 75% Power Saving, “IEEE Asian Solid-State Circuits Conference (A-SSCC), Nov. pp. 117-120, 2013

[6] Lin J.-F., et al., “Low-Power 19-Transistor True Single-Phase Clocking Flip-Flop Design Based on Logic Structure Reduction Schemes,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 7, no. 11, pp. 3033 - 3044, Nov. 2017.

[7] Absel K., et al., “Low-Power Dual Dynamic Node Pulsed Hybrid Flip-Flop Featuring Efficient Embedded Logic,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 9, pp. 1693-1704, Sept. 2013

[8] J. Tschanz, et al., “Comparative delay and energy of single edge-triggered and dual edge-triggered pulsed flip-flops for high-performance microprocessors,” in *Proc. ISPLED*, pp. 207–212, 2001

[9] Sudheer A., Ravindran A., “Design and Implementation of Embedded Logic Flip-Flop for Low Power Applications”, Procedia Computer Science, vol. 46, pp. 1393-1400, 2015

[10] Macha N.K., et al., “Crosstalk Logic Circuits with Built-in Memory”, ISVLSI-2021, under review. Archived: https://computing-lab.com/wp-content/uploads/2021/04/CBML_blind_review.pdf

[11] Macha N.K., et al., “A New Concept for Computing Using Interconnect Crosstalks,” IEEE International Conference on Rebooting Computing (ICRC), pp. 1-2, 2017

[12] Macha N.K., et.al., “Crosstalk based Fine-Grained Reconfiguration Techniques for Polymorphic Circuits,” IEEE/ACM NANOARCH,2018

[13] Macha N.K., et al., “A New Paradigm for Fault-Tolerant Computing with Interconnect Crosstalks” IEEE Conference on Rebooting Computing, 2018

[14] Iqbal M. A., et.al., “Designing Crosstalk Circuits at 7nm,” IEEE International Conference on Rebooting Computing (ICRC), 2019

[15] Iqbal M. A., et.al., “From 180nm to 7nm: Crosstalk Computing Scalability Study,” IEEE S3S Conference, 2019

[16] Macha N.K. “Crosstalk Computing: Circuit Techniques, Implementation and Potential Applications,” Ph.D. Dissertation, ECE, University of Missouri -Kansas City, MO, USA, 2020. Available: https://mospace.umsystem.edu/xmlui/handle/10355/80941

[17] Macha N.K., et al., “A New Computing Paradigm Leveraging Interconnect Noise for Digital Electronics Under Extreme Environments,” IEEE Aerospace Conference, 2019

[18] Iqbal M. A., et.al., “A Logic Simplification Approach for Very Large Scale Crosstalk Circuit Designs,” IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), 2019