Crosstalk Noise based Configurable Computing: A New Paradigm for Digital Electronics

Naveen Kumar Macha, Md Arif Iqbal, Bhavana Tejaswini Repalle, Sehtab Hossain, Mostafizur Rahman
Computer Science Electrical Engineering, University of Missouri-Kansas City, MO, USA
nmhe9@mail.umkc.edu, mibn8@mail.umkc.edu, rahmanmo@umkc.edu

***Abstract*—The past few decades have seen exponential growth in capabilities of digital electronics primarily due to the ability to scale Integrated Circuits (ICs) to smaller dimensions while attaining power and performance benefits. The scalability is now being challenged due to the lack of scaled transistors' performance and manufacturing complexities. Additionally, the growing cyber threat in fabless manufacturing era poses a new front that modern ICs need to withstand. Moreover, fault tolerance through circuit design is mostly achieved only through redundant circuit design which becomes an overhead in ICs. We present a new noise based computing where the interconnect interference between nanoscale metal lines is intentionally engineered to exhibit the programmable Boolean logic behavior. The reliance on just the coupling between metal lines and not on the transistors for computing and the programmability are the foundations for better scalability and security by obscurity and fault tolerance by reconfiguration. Here, we show experimental evidence of a functioning Crosstalk computing chip at 65nm technology. Our demonstration of computing constructs, gate level configurability, and utilization of foundry processes show feasibility. These results in conjunction with our simulation result at 7nm for various benchmarks, which show over 48% density, 57% power, and 10% performance gains over equivalent CMOS counterpart, show potentials. The benefits of Crosstalk circuits and their inherent programmable features set it apart and make it a promising prospect for future electronics.**

Keywords— Crosstalk Computing, Polymorphic logic Circuits, Crosstalk Reconfigurable Logic

I. Introduction

In sub-10nm nodes, device scaling is reaching its physical limits, and the interconnect bottleneck is dominating power and performance [1]. Additionally, the growing cyber threat in the fabless manufacturing era poses a new front that modern ICs need to withstand [2]. Crosstalk computing addresses some of the challenges by proposing a radical new concept for circuit design, scaling, and security [3]-[5]. In Crosstalk, deterministic interference between the adjacent interconnects is utilized for logic computing. The nanoscale interconnects are organized in a manner such that the passing signals (inputs) induce a deterministic voltage in the adjacent floating interconnect (the output). Additionally, with the help of control circuitry, the interference patterns can be controlled to achieve various logic configurations at run-time, hence the reconfigurability [4]-[5]. In this paper, we show experimental evidence of a functioning Crosstalk Computing chip at 65nm node using the TSMC process. We demonstrate both foundational gates and complex logic gates along with reconfigurability. These results in conjunction with our simulation results at 7nm for primitive gates show 30% power reduction and 34.5% performance gain over equivalent CMOS counterparts. The unique capabilities of Crosstalk computing can provide new opportunities for future electronics.

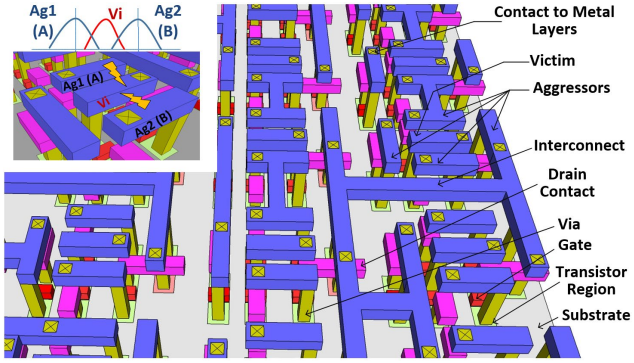

Fig. 1. Abstract view of the Crosstalk computing fabric. Interference between metal nano-lines take place in metal 1 layer. The bottom layer are for transistors that are required to control floating behavior of the output line and to maintain signal integrity. The arrangement of metal lines are according to circuit needs. The inset figure shows Crosstalk principle, where two aggressors (A and B) are transitioning and as a result charges are induced in Victim line (Vi).

II. Crosstalk Computing Overview

Traditionally, the interference between interconnects is considered a curse. This interference is more prominent in nanoscale ICs [1]. With Crosstalk computing, we aim to turn this curse into a feature. The idea is illustrated in Fig. 1, where the metal lines are arranged on the top and the controlling transistors are at the bottom. We intentionally arrange metal lines such that they can interfere in a deterministic manner (see inset figure in Fig.1). Then we capture this deterministic interference in a certain timeframe to ascertain logic. Let us use an example of a 2 input (*A* and *B*) logic (inset figure in Fig. 1). In Crosstalk, we would drive these inputs in two adjacent metal lines, and in between those lines, we will have another metal line to capture the interference charge (or the output). In interconnect terminology, the driving inputs would be called Aggressors (named as *Ag1* and *Ag2* in the inset figure), and the interference capturing line would be called the Victim (*Vi*). For capturing interference, the victim would be intentionally kept floating (not connected to the power supply or ground). As the aggressors transition from 0

to 1 or 1 to 0, corresponding interference would result in voltage gain or drop in the Victim node. If any of the input transitions (*A* or *B*) from 0 to 1, results in a sufficiently high voltage induction in the victim, we would achieve OR logic, and if only when both *A* and *B* transitions from 0 to 1, we notice high voltage induction in victim node, we would call the metal arrangement as performing AND logic. To capture the behavior in circuits for large scale integration, we utilize a control transistor and a clock. With the help of the control transistor, we periodically preset/sink the Victim node to the ground and deterministically keep the Victim node floating (ready for charge induction) during logic computation. An inverter is attached to the victim node is required to achieve complete voltage swing for the next stages. Fundamentally though, the logic computation happens due to the interference between interconnects and without the help of the transistor.

III. Technology Description

A Crosstalk prototype chip is fabricated using TSMC 65nm PDK. Full custom chip design flow is adopted to fabricate the chip. To suffice the coupling capacitor requirement, the NMOS device capacitor and DCAPs (MIM) are used. Custom circuit schematics are initially designed (Cadence Virtuoso) with the couplings. Subsequently, the circuits are fine-tuned for functionality, power, performance, and noise margins through iterative simulations (Synopsys HSPICE). Custom layouts are then designed, and Physical Verification steps are performed. A separate top-level circuit schematic and corresponding layout (Fig. 1D) is also designed, where the custom circuits are instantiated as standard cells. It is an IO limited design that mainly consists of 36 IO pads and IO cells, IO power ring, and core power ring, Power Network to deliver the power to circuits, Crosstalk logic gates. All routing is done manually. The clock network is routed for each cell with buffers to maintain drivability. The final layout (Fig. 1D) is extracted for parasitic RCs and simulated at various corners. The chip is fabricated through the TSMC multi-project-wafer run (MPW).

IV. Post Fabrication Results

Figs. 2.A&B.ii show the experimental results of the AND & OR Crosstalk logic gates. All the input signals along with the discharge signal were of 1V continuous square pulse of 10KHz frequency generated from the arbitrary function generators. The circuit operates in two states, Discharge state (when the victim node is connected to ground), and the Evaluation state (when the victim is disconnected from power

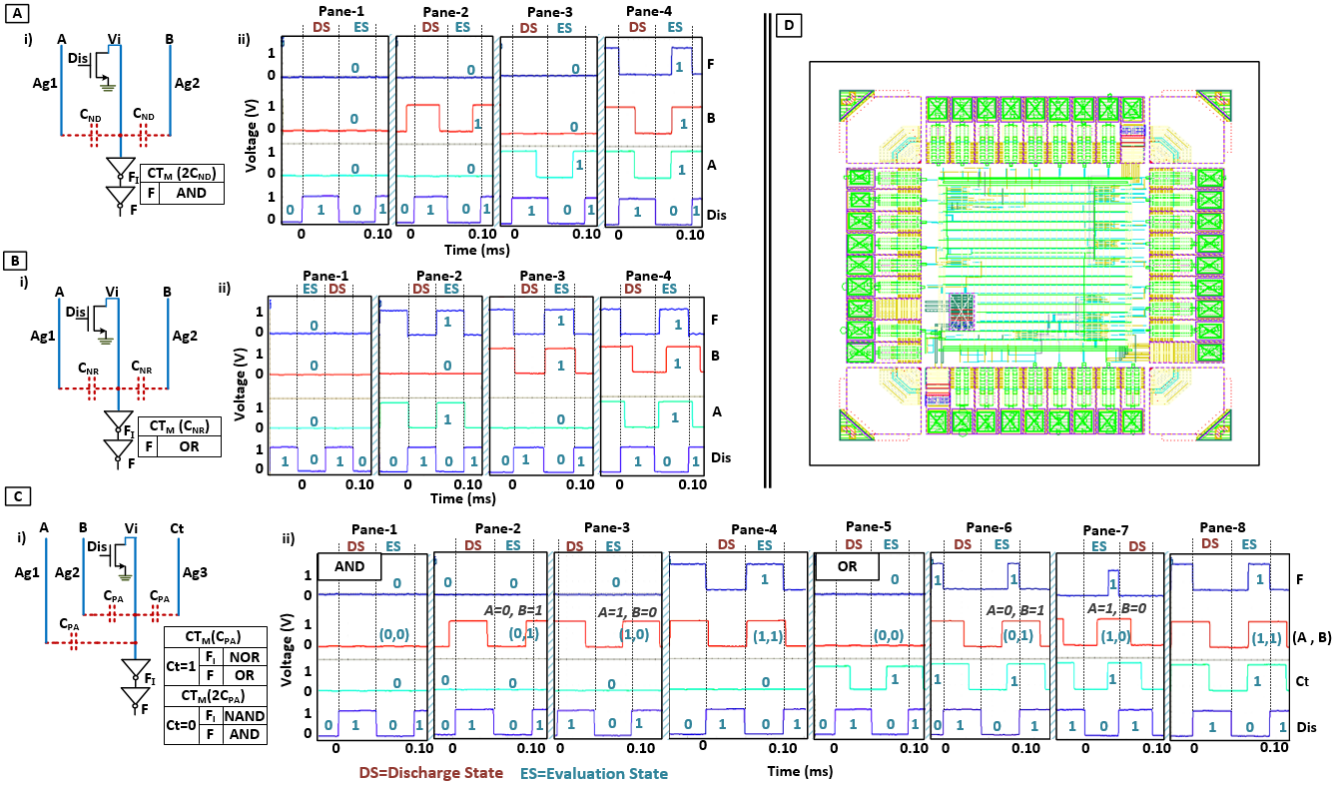

Fig 2. Experimental results of Crosstalk Logic and Reconfigurable gates. A) Crosstalk AND gate; i) Schematic, ii) Experimental results, B) Crosstalk OR gate; i) Schematic ii) Experimental results, C) Reconfigurable Crosstalk AND-OR gate i) Schematic, ii) Experimentals results, D) Layout of fabricated Crosstalk chip at TSMC 65nm with die size of 1mm^2. Input signals are denoted by *A* & *B*, Discharge signal is denoted as *Dis* and for reconfigurable gates additional signal is applied which is denoted by *Ct*. During each Evaluation stage output is taken from node *F*. All the signals are of 1V with 100µs period. For AND gate the coupling between the aggressors and victim node is defined by C_{ND} and for OR gate the coupling capacitance is C_{NR}. Even though both the schematics look the same, the main difference between them is the strength of the coupling capacitance where $C_{NR} > C_{ND}$. For reconfigurable gates, the coupling is the same for all the aggressors and is denoted by C_{PA}. Based on the control input the gate can shift the functionality between AND2 and OR2 gate (*Ct*=0 for AND gate and *Ct*=1 for OR gate). Experimental results are showing the snapshot of functional behaviour for all input combinations at different instances.

supply or ground and ready for capturing interference). Fig. 2.A.ii shows the experimental response of an AND gate where the first row (from bottom to top) shows the discharge signal (*Dis*), the second and third-row show two input signals (*A* and *B*) with 00, 01, 10, and 11 combinations given in Pane-1, Pane-2, Pane-3, and Pane-4 Evaluation states, respectively. The fourth row shows the output response of the AND gate. For all the circuits, the F_1 node gives inverting logic output (NAND, NOR, etc.), and the *F* node gives a noninverting logic output (AND, OR etc.). For input combinations 00 (in Pane-1), 01 (in Pane-2) and 10 (in Pane-3), the output response is logic 0. However, for inputs 11 (in Pane-4), the output is logic 1; which shows AND behavior. Similarly, OR gate implementation is shown in Fig.2.B.i and the experimental response is shown in the 4th row (bottom to top) of Fig.2.B.ii with the same input combination as AND gate. The difference between AND and OR gate is that the coupling strength for the OR gate is higher than the AND gate. We can see from Pane 1 to 4 that when *Dis*=1, irrespective of the input state the output becomes logic '0' but during Evaluation States (*Dis*=0) if there is 0 to 1 transition of either A (in Pane-2) or B (in Pane-3) or both (in Pane-4) the output becomes logic '1'. It can also be observed from the four panes that the Victim node is discharged to 0 before every new logic computation (or inference capture).

Since NAND/NOR are universal gates (can be used to implement any logic), the Crosstalk logic is functionally complete and can be used to implement any large-scale designs. Moreover, the simplicity of logic implementation through signal inference, also implies that more complex logic can be implemented without requiring multiple stages of cascading [3]-[7].

A. Dynamic Configurability (Post-Fabrication)

Another distinguishing feature of Crosstalk computing is the programmability post-fabrication (at run-time). At the application level, this would imply more robustness for functionality (if a functional block is defective, another block can be used to perform the same functions), and enhanced security also through camouflaging. Since Crosstalk circuits are identical (e.g., 2-input AND vs. 2-input OR) with the only difference of coupling capacitance, if the coupling capacitance can be altered at run time, different logics can be achieved in the same gate. We achieve this alternately by introducing a new aggressor called control input *(Ct)*. If the control input is 1, it induces an extra charge on Victim (*Vi)* net and biases the circuit to behave differently (which has the same effect as run time alteration of coupling strengths of input Aggressors). For example, Fig. 2.C.i shows the Crosstalk programmable AND2-OR2 circuit schematic. The inputs *A*, *B,* and *Ct* aggressor receive the same coupling capacitance, C_{PA}. The table adjacent to the circuit diagram (Fig. 2.C.i) lists the margin function and the circuit operating modes. A margin function for the Crosstalk gate indicates that how many and what input needs to be 1 for the inverter to alter its state. The margin function for the AND2-OR2 cell is $CT_M(2C_{PA})$, which makes the circuit behaves as an AND2 gate when the control signal *Ct=0*. On the contrary, when *Ct*=1, the control aggressor induces an extra charge through coupling capacitance C_{PA} and effectively manipulates the margin

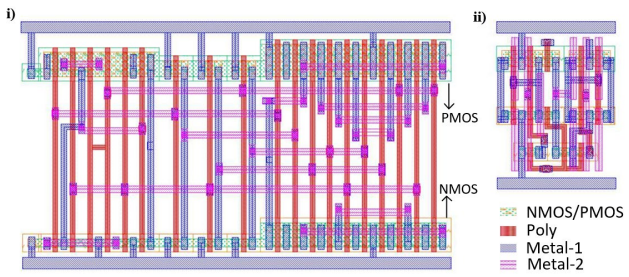

Fig. 3. Layouts of Full Adder Circuit (Sum and Carry): i) CMOS Layout ii) Crosstalk Layout

function to $CT_M(C_{PA})$, making it to behave as an OR2 gate. The response can be observed in Fig. 2.C.ii. Similar to the experimental results of Crosstalk basic gates, the 4th row shows the output *F*, the 3rd row shows input *A* and *B*, the 2nd row shows the control signal (*Ct*), and finally, the 1st row shows the discharge signal (*Dis*). From Fig. 2.C.ii, it can be depicted that the control signal (*Ct*) is kept low in the first 4 panes, during which both inputs *A* and *B* are given different input combinations, that is, 00 in Pane-1, 01 in Pane-2, 10 in Pane-3, and 11 in Pane-3. Only when both the input signals transition to high (Pane-4), the output signal *F* becomes logic 1, hence, behaving as AND gate. In the next 4 panes (Pane 5-8), the control signal is kept high during each Evaluation states (ES). It can be seen (in Pane-5) that all the input signals are kept low except the control signal and subsequently, the output *F* is also low. However, in the next ES states, at least one of the input signals *A* or *B* (01 and 10 in Pane-6, and 11 in Pane-7) is kept high and as a result, the output *F* is high. This shows the OR gate behavior. Since the Mixed Signal Oscillator, we have used has only four channels to observe the signals at runtime, two input signals (*A* & *B*) are synced together and connected to one of the channels in the oscilloscope. However, during 01 and 10 input combinations, one input is tied to the ground and the other is connected to the source (AFG). The Control signal (*Ct*), Discharge signal (*Dis*), and Output signal were connected to the other three channels of the oscilloscope. The results presented in Fig. 2.C.ii shows that based on the state of the control signal (*Ct*), the behavior of the circuit can be programmed to be either the AND or the OR gate at runtime. This dynamic configurability can be a key enabler for anti-counterfeiting, resource sharing, and fault-tolerant computing. This reconfigurability would allow cloaking/camouflaging of logic functionality during anti-counterfeiting. In reference [8], [9] it is also shown that dynamically configurable systems are harder to hack. Besides, the regular layout structure of Crosstalk logic gates also adds a layer of security because they would be identical and difficult to trace during reverse engineering. The run-time programmability can also open up a new front for resource sharing and fault-tolerant computing, as a portion of the circuit can be configured to do the work of other portions. Fig. 2.C shows that the Reconfigurable Crosstalk gates can be also used for resource sharing purposes. For example, it can be used in the systems where AND and OR operations are multiplexed. A single gate can serve both the purpose, thus share the resources [10].

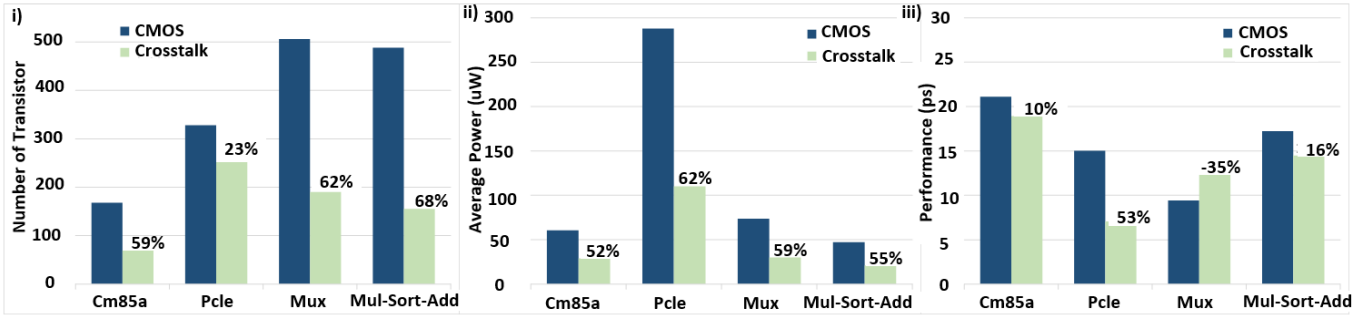

Fig. 4. Comparison of Crosstalk and CMOS large-scale circuits. i) Density Comparison, ii) Average Power and iii) Performance Comparison for Cm85a, Pcle, Mux amd polymorphic circuit (Mul-Sort-Add) at 7nm technology.

Intuitively, we can gather the merits of the Crosstalk Computing approach by inspecting the layouts. Fig. 3 shows the layout of a full adder. For the full adder circuit, CMOS implementation requires 40 transistors using CMOS circuit style (12 transistors for each XOR gates and 12 for carry logic), whereas the Crosstalk implementation requires just thirteen transistors and the interconnection requirements at cell-level are also considerably less. It is evident from Fig. 3 that Crosstalk circuits consume less active device areas compared to CMOS.

V. Gauging the Potentials (Benchmarking at 7nm)

To further evaluate the potential, we have done an extensive comparison between CMOS and Crosstalk. We have implemented 3 MCNC benchmark circuits, a polymorphic circuit (2-bit Multiplier-Sorter-Adder circuit), and compared density, power, and performance results for CMOS at 7nm. For the benchmarking circuits designed at 7nm, there are 48%, 57%, and 10% improvements against CMOS designs in terms of density, power, & performance, respectively. The improvement in power for Crosstalk gates is because of less number of active devices leading to lower overall load power, less cell internal power, and fewer device dissipations. As can be seen from Fig. 4, in terms of transistor count, the highest reduction is for the Mul-Sort-Add circuit, which is 68%. For Cm85a and Pcle circuits, the reduction in transistor count is 59% and 23%, respectively. Crosstalk circuits show on average 57% power benefits over CMOS counterparts. The benefits are primarily due to the reduction in transistor count. However, the reduction in average power for the Mux and the polymorphic circuit is not much even though transistor count reduction is maximum compared to other circuits. This is because these circuit implementations require many auxiliary initializers which results in more switching activities hence, less power reduction. On the contrary, for the Pcle circuit, power reduction is more because it requires less number of buffers and initializer circuits which means less switching activity. However, for Crosstalk design, Cm85a and Pcle circuits have 10% and 53% improvement in performance, respectively.

VI. Conclusion

Here, we have shown the experimental proof of Crosstalk computing technology at 65nm. Our experimental results indicated that the gate level configurability is feasible with the existing process techniques. Our benchmarking results also show significant improvement in density, power, and performance for Crosstalk circuits compared to CMOS even with scaling down of technology nodes for large-scale circuit implementations. The paper lays foundations for practical realization of Crosstalk circuits. Successful implementation of Crosstalk computing at large scale can be a game-changer for future digital electronics.

References

[1] M. V. Fischetti et al., "Scaling MOSFETs to 10 nm: Coulomb Effects, Source Starvation, and Virtual Source," 2009 13th International Workshop on Computational Electronics, Beijing, China, 2009, pp. 1-4, doi: 10.1109/IWCE.2009.5091145.

[2] T. D. Perez and S. Pagliarini, "A Survey on Split Manufacturing: Attacks, Defenses, and Challenges," in IEEE Access, vol. 8, pp. 184013-184035, 2020, doi: 10.1109/ACCESS.2020.3029339.

[3] N. K. Macha, V. Chitturi, R. Vijjapuram, M. A. Iqbal, S. Hussain and M. Rahman, "A New Concept for Computing Using Interconnect Crosstalks," 2017 IEEE International Conference on Rebooting Computing (ICRC), Washington, DC, USA, 2017, pp. 1-2, doi: 10.1109/ICRC.2017.8123636.

[4] N. Macha, S. Geedipally, B. Repalle, M. A. Iqbal, W. Danesh and M. Rahman, "Crosstalk based Fine-Grained Reconfiguration Techniques for Polymorphic Circuits," 2018 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Athens, Greece, 2018, pp. 1-7.

[5] N. K. Macha, B. T. Repalle, M. A. Iqbal and M. Rahman, "A New Computing Paradigm Leveraging Interconnect Noise for Digital Electronics Under Extreme Environments," 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1-8, doi: 10.1109/AERO.2019.8741746.

[6] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, J. D. Meindl., "Interconnect limits on gigascale integration (GSI) in the 21st century," in Proceedings of the IEEE, vol. 89, no. 3, pp. 305-324, March 2001, Doi: 10.1109/5.915376.

[7] M. A. Iqbal, N. Kumar Macha, B. T. Repalle and M. Rahman, "A Logic Simplification Approach for Very Large Scale Crosstalk Circuit Designs," 2019 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Qingdao, China, 2019, pp. 1-6, doi: 10.1109/NANOARCH47378.2019.181306.

[8] Garner, Paul M., Carrie Boone, and Darren J. Cepulis. "Dynamically configurable portable computer system." U.S. Patent No. 5,014,193. 7 May 1991.

[9] N. Rangarajan, S. Patnaik, J. Knechtel, R. Karri, O. Sinanoglu, and S. Rakheja, "Opening the Doors to Dynamic Camouflaging: Harnessing the Power of Polymorphic Devices," in IEEE Transactions on Emerging Topics in Computing, DOI: 10.1109/TETC.2020.2991134.

Rai, Shubham, et al. "Designing efficient circuits based on runtime-reconfigurable field-effect transistors." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 27.3 (2018): 560-572.