Memories - Materials, Devices, Circuits and Systems 7 (2024) 100094

Contents lists available at ScienceDirect

Memories - Materials, Devices, Circuits and Systems

journal homepage: www.elsevier.com/locate/memori

A review of crosstalk polymorphic circuits and their scalability[☆]

Md Arif Iqbal[a], Srinivas Rahul Sapireddy[a], Sumanth Dasari[a], Kazi Asifuzzaman[b], Mostafizur Rahman[a,*]

[a] School of Science and Engineering, University of Missouri Kansas City, Kansas City, MO, USA
[b] Advanced Computing Systems Research, Oak Ridge National Laboratory, TN, USA

ARTICLE INFO

Keywords:
Reconfigurable circuit
Crosstalk computing
Open source PDK

ABSTRACT

Using a control variable, the functionality of Polymorphic circuits can be modified, making them adaptable and useful for reconfiguring circuit behavior — all the way from gate level to system level. State-of-the art polymorphic circuits are based on custom non-linear circuit design or emerging devices such as ambipolar FET, configurable magnetic devices etc. While some of these approaches are inefficient in performance, others involve exotic devices. The Crosstalk computing based polymorphic circuits offer a fresh perspective. In Crosstalk, the interconnect interference between nanoscale metal lines is intentionally engineered to exhibit the programmable Boolean logic behavior. This approach relies on the coupling between metal lines and not on the transistors for computing, resulting in better scalability, security by obscurity, and fault tolerance by reconfiguration. Our novel approach is backed by the mathematical formulation that conveys the rationale to generalize and achieve a wide variety of polymorphic circuits. Our experiments, including design, simulation, and Power Performance Area (PPA) characterization results indicate that crosstalk circuits provide significant improvement in transistor count (about 3x), switching energy (2x), and speed (1.5x) for polymorphic logic circuits. In the best-case scenario, the transistor count reduction is 5x. This paper presents Crosstalk computing's fundamentals, polymorphism and the scalability aspects to compete/co-exist with CMOS for digital logic implementations below 10 nm. Our scalability study uses Open Source 7 nm PDK, considers all process variation aspects and accommodates worst-case scenarios. The study results for various benchmark circuits show that the Crosstalk technology is a viable alternative to CMOS for digital logic implementations below 10 nm, having 48% density, 57% power, and 10% performance gains over equivalent CMOS counterparts. Finally, we compare Crosstalk Polymorphic Circuit design technique with similar approaches described in related works and discuss its features and constraints.

1. Introduction

Polymorphic circuits are a type of digital circuit that can change its behavior between multiple functions through a control variable [1]. For example, an AND gate can switch to an OR gate and vice-versa. This ability to transform its behavior makes them useful in a variety of applications [2–8], such as in reconfigurable circuits and systems [2,3], resource sharing [2,3], multifunctional adaptive systems [4,5] fault tolerance [6], and self-test circuits [7–9].

One of the main features of polymorphic circuits is to be able to reuse circuits for executing different functions, which in turn, helps to retain the same footprint for the circuit. This can be particularly helpful to sustain Moore's Law, which indicates that the number of transistors in a dense integrated circuit is doubled every two years. As the miniaturization of integrated circuits is approaching physical limits, this reconfigurability of polymorphic circuits is becoming increasingly significant.

Several techniques have been proposed for designing and implementing polymorphic circuits, such as genetic algorithms, chaos computing, and emerging tunable polarity transistors. However, to be considered a truly polymorphic circuit, along with the innate multi-functional nature, the control between different functions should be

[☆] Notice: This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

[*] Corresponding author.
E-mail address: rahmanmo@umkc.edu (M. Rahman).

https://doi.org/10.1016/j.memori.2023.100094
Received 31 August 2023; Received in revised form 28 November 2023; Accepted 7 December 2023
Available online 19 December 2023

enabled by inherent device characteristics and/or external environmental influences [1]. Genetic algorithms [10,11] have been extensively researched and can morph their functional behavior based on different environmental control variables such as temperature, supply voltage, control signal, light, radiation, etc. [12–14]. But the circuits evolved using genetic algorithms have some disadvantages [15], such as technology dependency, scalability, and inefficiency in speed and power. Chaos computing [16] is another actively pursued approach in which nonlinear dynamics in transistors and circuits are captured to implement multifunctional circuits. However, these circuits are custom nonlinear/mixed-signal circuit designs for digital circuits and may not be suitable for mainstream digital circuits/systems.

Recently, polymorphic circuits are also designed using emerging tunable polarity transistors [17–21], which can be configured either as p-type or n-type based on a control signal [18]. Such morphable transistors fits into several fine grained polymorphic circuit schemes. However, these novel devices require complex engineering compared to mainstream CMOS devices, and the circuit schemes necessitate additional circuitry to switch the power rails when the transistors change to p-type/n-type. The other alternate approaches using emerging spintronic devices were also proposed [5], but they rely on complex information encoding schemes through spin-polarized currents and bipolar voltages, etc. Consequently, they significantly depart from existing computational device and circuit paradigms.

In our previous work [22], we propose a novel computation scheme named crosstalk-computing (CT-Computing) which implements a wide range of efficient and compact polymorphic circuits [23]. The signal interference between adjacent metal lines is astutely engineered to a logic principle in the CT-Computing technique. Contrary to the traditional switch-based logic computations, a deterministic superposition of crosstalk accompanied input signals are used to conduct the logic operation in CT-Computing.

The counterpart of connecting switches/transistors in different topologies to achieve logic is tuning the crosstalk coupling capacitances in CT-Computing. Furthermore, an additional control signal is used for polymorphism, which biases the circuit to alter its functional behavior [21]. As with any new technology, scalability is a crucial aspect to consider for Crosstalk computing. Therefore, to fully evaluate the efficiency of the design methodologies at advanced technology nodes, a comprehensive process design kit (PDK) that includes all necessary collateral such as schematic entry, layout, design rule checking, parasitic extraction, transistor-level simulation, library generation, synthesis, and automatic placement and routing (APR) is required.

In this paper, we present a comprehensive review of Crosstalk computing and polymorphic circuits and also show scalability aspects. Our review of scalability study involves usage of open-source Arizona State Predictive 7nm PDK (ASAP7) [24]. We present our benchmark circuit designs and their comparison with CMOS equivalent at 7nm. Using ASAP7nm PDK, we also show that Crosstalk circuits can be designed to function correctly for both best-case and worst-case process variations.

The rest of the paper is organized as follows. Section 2 presents CT-Computing concepts, including the assumptions for logic implementation, circuit techniques to implement basic and complex circuits. Section 3 discusses polymorphism in CT-Computing and shows a wide range of CT-Polymorphic gates that can morph between different operations. In Section 4, we discuss the impact of process variation in Crosstalk Computing at 7 nm. The behavior of Crosstalk NAND and NOR gate for different process corners is also shown here. Section 5 explains the larger-scale design aspects using Crosstalk logic cells at 7 nm and gives the benchmarking results. In Section 6, we compare CT-Polymorphic Circuits with similar approaches described in the related works. Finally, we conclude the paper in Section 7.

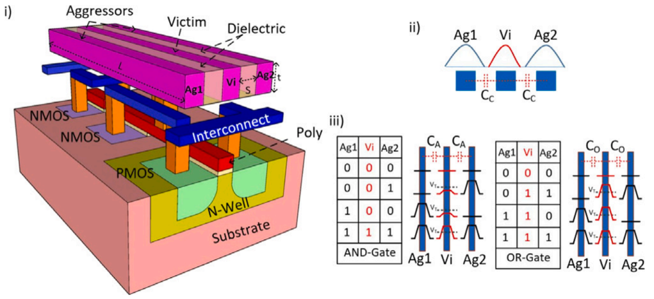


Fig. 1. Abstract view of crosstalk fabric and mechanism of crosstalk-computing fabric. (i) Crosstalk fabric; Coupling capacitances will be on the top, denoted as 'dielectric' between nano-metal lines denoted as 'aggressor' and 'victim.' (ii) Aggressor–victim phenomenon, (iii) Implementing logic gates based on CT-Computing.

2. Crosstalk computing

CT-Computing fabric majorly comprises four components, Crosstalk Layer (CTL), Active Devices, Interconnects, and Vias. The logic is computed in the CT Layer, which is a metal layer constituted by capacitively coupled metal lines named Aggressors (Ag) and Victim (Vi). Interconnects and Vias serve their common purpose and contribution to coupling capacitance in CTL. The active devices are regular FinFETs. In the following sections, we discuss the purpose of the transistors — to accurately control and reconstruct the signals.

Fig. 1(i) shows the abstract view of the CT-Computing fabric where the coupling network is envisioned on a metal layer with closely coupled metal lines. The aggressors are used as inputs and the victim is used as the output. Fig. 1(ii) illustrates the aggressor–victim scenario of Crosstalk logic. It shows the capacitive interference of the signals for logic computation — the transition of the signals on two aggressor metal lines (Ag1 and Ag2) induces a resultant summation charge/voltage on the victim metal line (Vi) through capacitive couplings C_C. As this aspect complies with the charge conservation principle, the net voltage of the victim is deterministic and retains the information about signals on two aggressor nets and its magnitude is dependent on the coupling strength between the victim and aggressor net [25–27]. The coupling capacitance is directly proportional to the relative permittivity of the dielectric and lateral area of metal lines (length multiplied by the vertical thickness of metal lines) and inversely correlated with the separating distance of the metal lines. Adjusting the coupling capacitance using the aforementioned variables provide opportunities to modify the induced summation signal to the individual logic implementation. Fig. 1(iii) depicts the notion of implementing logic gates (AND and OR) using crosstalk signal interference. Input signal transitions constitute a voltage proportional to coupling capacitances. As shown in Fig. 1(iii), for AND gate, the inputs coupling, C_A, can be chosen such that the magnitude of the voltage induced on the output net (Vi) is greater than a selected threshold voltage V_T only when both inputs transition from 0 to 1 (i.e., for input combination 11). Thus, the threshold voltage V_T differentiates logic 0 and 1. When only one of the input transitions (input combinations 01 and 10), the voltage induced on the victim net is below the V_T; hence, the output can be considered logic 0. Therefore, as Fig. 1(iii) shows, an AND gate functionality can be implemented using the crosstalk signal interference scheme. Similarly, OR gate functionality can be realized by increasing the coupling capacitance, which can be done by appropriately tuning the physical dimensions or choosing high-k dielectric material, or both. The intuition for OR gate implementation is also shown in Fig. 1(iii). Compared to the AND gate, for the OR gate, the coupling capacitance C_O is increased ($C_O > C_A$) such that the transition of either of the input signal from 0 to 1 is now sufficient to induce a voltage above the logic threshold (V_T).

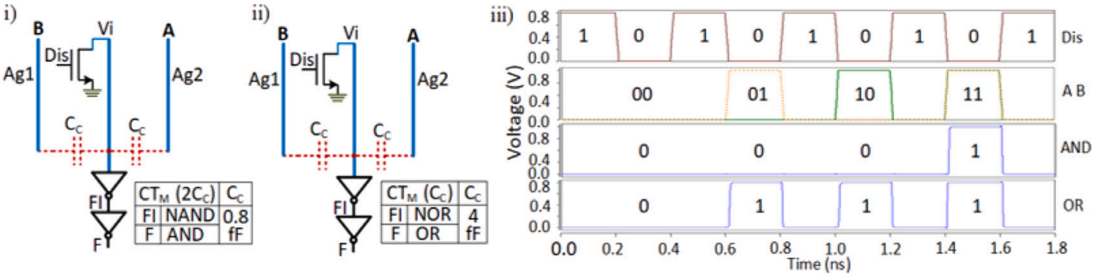


Fig. 2. Crosstalk basic gates: (i) AND gate circuit schematic, (ii) OR circuit schematic, (iii) Simulation response of AND and OR gates.

Table 1
Crosstalk logic design table for basic gates.

Gate	C_C (fF)	Aggressor Weights		Margin Fuction	Width Ratio (PMOS:NMOS)
		w1	w2		
AND2	0.8	1	1	$CT_M(2C_C)$	1 : 1
OR2	4	1	1	$CT_M(C_C)$	1 : 3

Therefore, input combinations 01, 10, and 11 evaluate logic output 1 as an OR gate. Actual implementation of the CT-logic circuits and their dependable operation in cascaded circuits need improved circuit techniques to be adopted with the idea presented above. The circuit techniques are presented next.

2.1. Crosstalk circuit theory and basic logic gates

Although properly engineered and coupled nano-metal lines are sufficient to emulate the logic behavior in CT-Computing [22], the output net (Vi), which collects the crosstalk charge, needs to satisfy three conditions to achieve deterministic functionality in all sorts of real circuit environments. First, the Vi net needs to start from a known initial state. Second, it should remain floating during logic evaluation to collect the crosstalk charge. Third, the output node needs to drive the fanout gates in actual circuits and retain strong binary voltage levels. As shown in Fig. 2, the first two conditions are met by connecting a discharge transistor to the Vi net, and the third condition is met by adding an inverter to the Vi net. The 2-input AND gate is presented in Fig. 2(i). In the AND gate, input aggressor nets (A and B depicted as Ag1 and Ag2) are coupled with Vi net through coupling capacitances of Value C_C (C_C values are given in Table 1).

The Dis signal drives the discharge transistor. The CT-logic gates have two alternate operating states: the logic evaluation state (LE) and the discharge state (DS). During DS (enabled by Dis signal), the floating victim node is shorted to ground through the discharge transistor and thus starts from a known initial condition, i.e., 0. The alternate DS states ensure the correct logic operation during every LE state by clearing off the charge from the previous logic operation. While the Vi net is floating (indicating a LE state) the rising transitions on the aggressor nets exert a linearly correlated summation voltage on the Vi net, which is connected with a CMOS inverter.

The inverter acts as a regenerative threshold function; that is, if the voltage computed on V_i net is above the inverter's threshold-voltage/trip-point (V_{INV}), it outputs the logic level 0, and vice-versa; it regenerates the signals and restores them to full swing. Also, (V_{INV}) is tuneable by changing PMOS to NMOS width ratios if required. The CT-logic gates presented in this paper are designed using the 16 nm Predictive Technology Modeling (PTM) transistors [28] and simulated on SPICE. The simulation response of the designed AND gate is shown in Fig. 2(iii), where the first panel indicates the discharge signal (Dis), the second panel shows two input signals A and B with four combinations (00, 01, 10 and 11) given through consecutive LE stages (having Dis = 0) and the third panel indicates the output of the AND gate. The FI node transmits inverting logic output while the F node transmits noninverting logic output (AND). Likewise, Fig. 2(ii) presents the implementation of an OR gate and its simulation response in panel OR of Fig. 2(iii). We depict the same input signals in panel 1 and 2 for both circuits for the sake of simplicity. The coupling strength (C_C) makes the difference between AND/OR gates. As listed in Table 1, the coupling capacitance for the OR gate is higher (4 fF) than the AND gate (0.8 fF) The coupling capacitance, C_C, is unique to each gate and is directly related to the relative permittivity of the dielectric and the lateral area of metal lines. It is also inversely proportional to the distance of separation of metal lines. By tuning the coupling capacitance values using the variables mentioned above, the induced summation signal on the victim layer is tailored to the specific logic implementation. The calculated capacitance value is then verified through RC extraction and subsequently HSPICE simulation. C_C is the individual quantized capacitance for each gate.

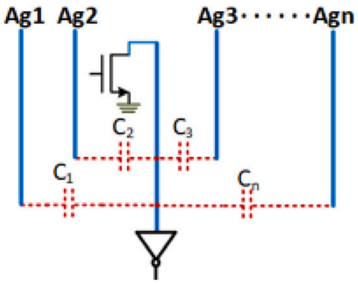
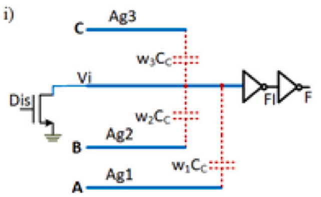


Fig. 3. Capacitive network in a generic Crosstalk gate.

The input aggressors would receive the coupling strengths in integer multiples of C_C, referred to as aggressor weights in Table 1. The design of three parameters, aggressor weights, C_C, and inverter sizing, constitute the logic computed, which is discussed next.

The operation of CT-logic gates would be represented functionally using a crosstalk-margin function, CTM($2C_C$), which specifies that the inverter of the CT-logic gate flips its state only when the victim node sees the input transitions through the total coupling greater than or equal to C_C. For example, AND gate CT-margin function is $CT_M(C_C)$. It states that the inverter flips its state only when the victim node sees the input transitions through total coupling greater than or equal to $2C_C$, which happens only when both inputs are high. Likewise, for the OR gate (shown in Fig. 2(ii)), CTM(C_C) is the CT-margin function, which indicates the transition of any of the aggressors is enough to flip the inverter, therefore implementing the OR functionality. The CT-Circuit design and working mechanism for all logic types will be explained using the CT-margin function in this paper. Therefore, to further elucidate the relationship between the CT-margin function and the working mechanism of CT-logic gates, consider a generic crosstalk capacitive network with 'n' number of input aggressors as shown in Fig. 3. The voltage induced on the victim net can be calculated by



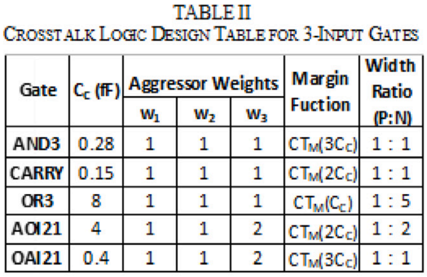


TABLE II
CROSSTALK LOGIC DESIGN TABLE FOR 3-INPUT GATES

Gate	C_C (fF)	Aggressor Weights			Margin Fuction	Width Ratio (P:N)
		W_1	W_2	W_3		
AND3	0.28	1	1	1	$CT_M(3C_C)$	1 : 1
CARRY	0.15	1	1	1	$CT_M(2C_C)$	1 : 1
OR3	8	1	1	1	$CT_M(C_C)$	1 : 5
AOI21	4	1	1	2	$CT_M(2C_C)$	1 : 2
OAI21	0.4	1	1	2	$CT_M(3C_C)$	1 : 1

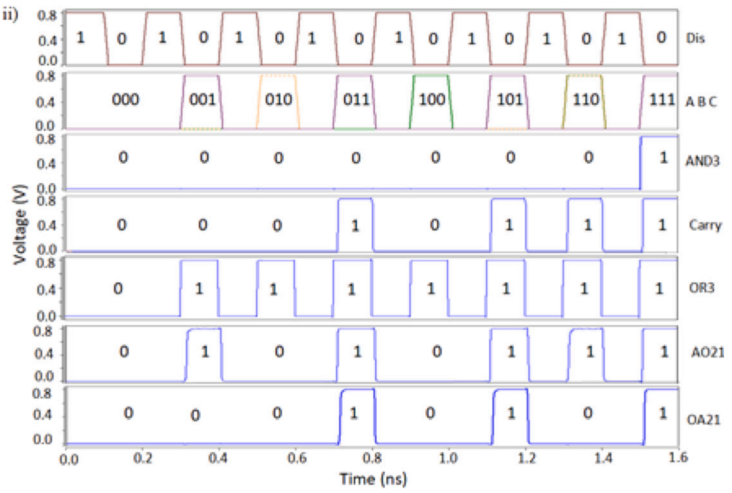


Fig. 4. Crosstalk complex logic gates: (i) A generic schematic representing all 3-input complex logic functions (ii) Simulation response of 3-input complex logic functions (AND3, CARRY, OR3, AO21, OA21).

applying KVL, as follows

$$V_{Vi} = \frac{C_1}{C_T}V_1 + \frac{C_2}{C_T}V_2 + \cdots + \frac{C_n}{C_T}V_n \tag{1}$$

where, C_1, C_2, …, C_n are capacitances from respective aggressors to the V_i net. C_T is the total capacitance on the V_i net, which is given by:

$$C_T = C_1 + C_2 + \cdots + C_{INV} + C_{ds}$$

C_{INV} is the inverter gate capacitance. C_{ds} is the diffusion capacitance of the V_i net.

C_{ds} = Discharge transistors drain to source capacitance

The final voltage levels on input aggressors, which are given by V_1, V_2,…, V_n in Eq. (1), can be formulated as voltage sources, given by,

$$V_i = L_i V_{DD}$$

$$\text{where, } L_i = \begin{cases} 0 & \text{if logic 0} \\ 1 & \text{if logic 1} \end{cases}$$

The capacitances induced to input aggressors are multiples of a constant C_C determined for each gate. Therefore,

$$C_i = w_i * C_C;$$

where w_i is the integer multiplying factor representing the weighted strength of each aggressor. The Eq. (1) now modifies to

$$V_V i = \frac{C_C}{C_T}.V_D D.m \ldots \tag{2}$$

where, $m = w_1L_1 + w_2L_2 + \cdots + w_nL_n$. evaluates integer values. The CT-margin function of each gate can be related to Vi net voltage as follows. Consider given logic gate is associated with the CT-margin function CT_M (k.C_C) (k takes integer values), then for all the input combinations that produce logic output 0, the Vi net voltage computed is greater than inverter trip point (i.e., $V_{Vi} > V_{INV}$) and m is greater than or equal to k (i.e., $m \geq k$). Similarly, for all input combinations that produce logic output 1, where w_i is the weight of input i and L_i is the logic level of input i. , m is less than k (i.e., $m < k$), and Vi net voltage is less than inverter trip-point ($V_{Vi} < V_{INV}$). Therefore, the first step in designing crosstalk logic gates is to derive the CT-margin function for logic computed, i.e., k. For $m = k$, V_{Vi} should be greater than V_INV. So, we start with $m = k$ and $V_{Vi} = V_{INV}$ And compute the C_C value as below from Eq. (2),

$$C_C = \frac{(C_{INV} + C_{ds})}{(V_{DD}k)/V_{INV} - \sum_{i=1}^{i=n} w_i} \tag{3}$$

We start the design with this coupling cap C_C and fine-tune it through simulations to get the symmetric logic response. It can be noticed that C_C is inversely proportional to k, interpreting as the lower margin function (k) gates need the stronger cap. Table 1 lists logic design information for AND2 and OR2 gates. For AND k = 2, for OR gate k = 1, therefore OR needs stronger C_C. Also, from Eq. (3), C_C is the asymptotic function of V_{INV}(directly proportional). For example, for OR gate, C_C increases sharply with a rise in V_{INV} and approaches to infinity if $V_{INV} \to \frac{V_{DD}}{2}$ ideally. Such large values of C_C are impractical in design. Thus, we decrease the inverter trip-point by skewing the inverter.

It should be noticed that skewing inverter increases the gate capacitance C_{INV} in Eq. (3), which conversely relaxes the reduction in C_C achievable through the skewing threshold function of the inverter. Nevertheless, it brings C_C to practical design limits. Also, C_C increases with an increase in fan-in of the gate, i.e., $\sum_{i=1}^{i=n} w_i$.

Therefore, high fan-in and less k values necessitate larger skewing of the inverter, e.g., high fan-in OR gate. Thus, by deriving the three design variables C_C, w_i, and V_{INV} from the above theory and then engineering them through simulations, any linear logic gate can be designed and explored in subsequent sections.

2.2. Complex logic gates

Interesting and complex logic functions can be implemented by expanding the fan-in (number of input aggressors) due to the increased coupling capacitances and CT-margin function choices. Fig. 4(i) presents the circuit schematic of a generic 3-input Crosstalk gate, and Table 2 lists CC and w_i values, CT-margin function, PMOS to NMOS width ratio for all 3-input complex-logic functions that are realized. For logic design, each gate receives an individual quantized C_C value and various aggressor weights (w_i) as listed in the table. Additionally, the input aggressors can be given equal or unequal coupling capacitances. Gates with equally coupled aggressors are defined as Heterogeneous CT-Logic gates. On the other hand, gates with unequally coupled aggressors are defined as homogeneous CT-Logic gates. These options of having homogeneous and heterogeneous coupling further widen the scope of implementing complex logic functions through CT-Computing techniques.

3. Crosstalk polymorphic logic gates

From Tables 1, 2 and circuit schematics it can be observed that dissimilar to CMOS circuits which have fixed patterns of parallel and series connections of switches (transistors) for each logic type, CT-logic circuits are of uniform pattern with the only difference in their coupling capacitances. That means if the coupling capacitances from inputs to

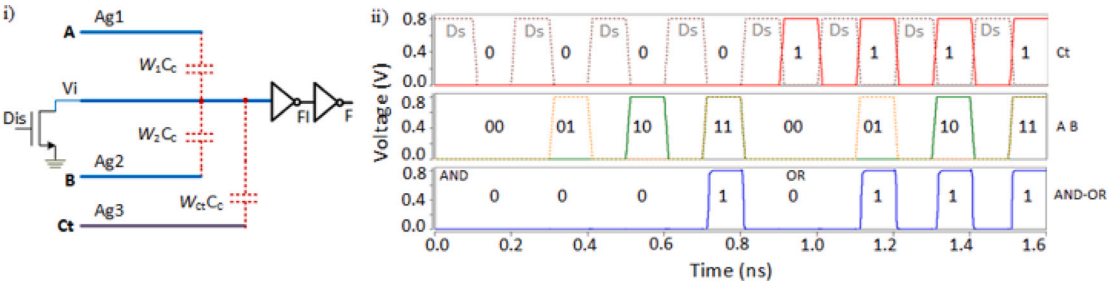
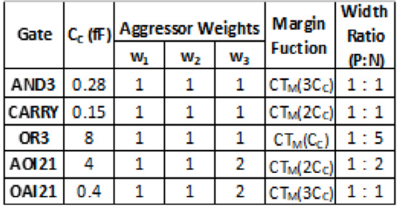


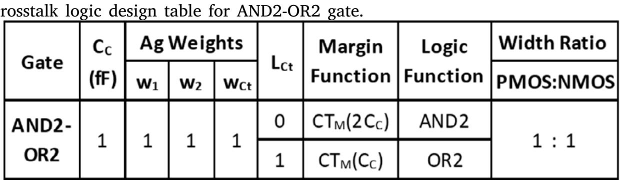
Fig. 5. 2-input crosstalk-polymorphic logic gate: (i) AND2-OR2 schematic, (ii) AND2-OR2 simulation response.

Table 2
Crosstalk logic design table for 3-input gates.



Gate	C_c (fF)	Aggressor Weights			Margin Fuction	Width Ratio (P:N)
		w_1	w_2	w_3		
AND3	0.28	1	1	1	$CT_M(3C_C)$	1 : 1
CARRY	0.15	1	1	1	$CT_M(2C_C)$	1 : 1
OR3	8	1	1	1	$CT_M(C_C)$	1 : 5
AOI21	4	1	1	2	$CT_M(2C_C)$	1 : 2
OAI21	0.4	1	1	2	$CT_M(3C_C)$	1 : 1

Table 3
Crosstalk logic design table for AND2-OR2 gate.



Gate	C_C (fF)	Ag Weights			L_{Ct}	Margin Function	Logic Function	Width Ratio
		w_1	w_2	w_{Ct}				PMOS:NMOS
AND2-OR2	1	1	1	1	0	$CT_M(2C_C)$	AND2	1 : 1
					1	$CT_M(C_C)$	OR2	

the Vi net can be altered at runtime, the logic behavior of the gate can also be altered. Having the potential to be able to modify the runtime logic behavior could open up avenues to design and develop a new category of polymorphic/reconfigurable logic circuits based on CT-Computing. Instead of achieving the runtime modification of coupling capacitances by manipulating material properties or building novel devices, as an alternative, the Vi net can be coupled with an additional control aggressor (Ct). The transition of the signal on Ct would induce an extra charge/voltage on the Vi net — equivalent to the runtime modification of the capacitance coupled with the Vi net. This extra voltage augmented on the Vi net would change the anticipated logic behavior of the gate. If properly engineered, this extra-induced voltage can be intelligently used to realize new functional patterns which can promote polymorphic gates further.

We show polymorphism between the logic functions discussed in the previous section; homogeneous to homogeneous logic: AND2-OR2, AND3-OR3, AND3-CARRY, OR3-CARRY; heterogeneous to heterogeneous logic: AO21-OA21; homogeneous to heterogeneous logic: AO21-AND3, AO21-OR3, AO21-CARRY, OA21-AND3, OA21-OR3, OA21-CARRY.

For a generic CT-Polymorphic gate, the control aggressor Ct will be coupled to Vi net through capacitance $w_{Ct}C_C$ (w_{Ct} is the weight signifying the control aggressor's strength). The Vi net voltage Eq. (2) now turns to,

$$V_{Vi} = \frac{C_C}{C_T} V_{DD}(m + w_{Ct}L_{Ct})$$

where $m = w_1L_1 + w_2L_2 + \cdots + w_nL_n$, and

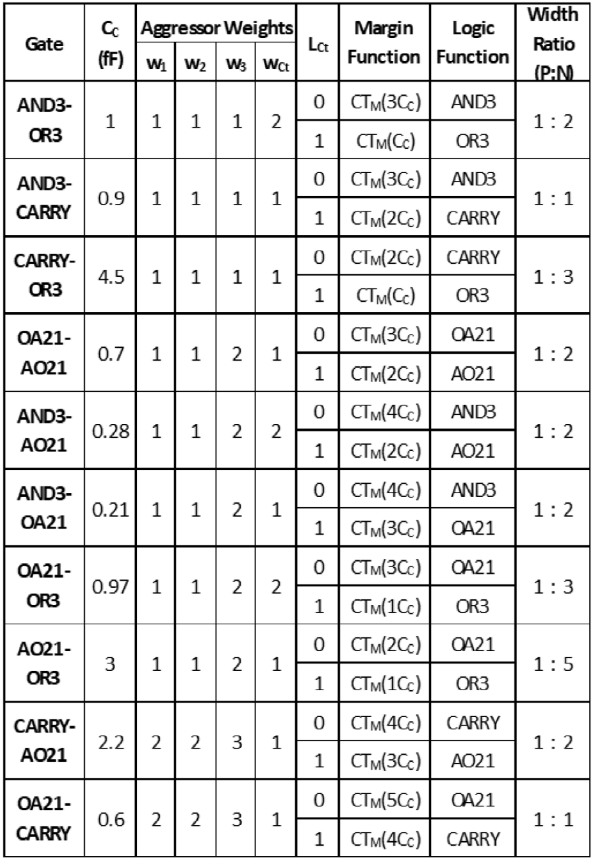
$$L_{Ct} = \begin{cases} 0 & \text{if control signal Ct is low voltage} \\ 1 & \text{if control signal Ct is high voltage} \end{cases}$$

The CT-margin function is an abstraction for logic behavior in CT-Computing. The operation of CT logic gates would be represented functionally using a CT-margin function, $CT_M(C_C)$, which specifies that the inverter of the CT logic gate flips its state only when the victim node sees the input transitions through the total coupling greater than or equal to CC. For example, AND gate CT-margin function is $CT_M(2C_C)$. It states that the inverter flips its state only when the victim node sees the input transitions through total coupling greater than or equal to $2C_c$, which happens only when both inputs are high. Similarly, for the OR gate, the CT-margin function is $CT_M(C_C)$, which means that the transition of any one of the aggressors is sufficient to flip the inverter, thus executing the OR behavior. Thus, converting the CT-logic gate's behavior from one function to another function may also accurately transform their margin functions [29]. The CT-Polymorphic logic gate evaluates to 0 (at node FI) only when, $V_{Vi} > V_{INV}$. The aggressor weights and CC are tuned such that $V_{Vi} > V_{INV}$ only when,

$$m \geq (k - w_{Ct}LCt)$$

Therefore, for a CT-Polymorphic gate to evaluate to 0 at the output node FI, the input logic levels (L_i); thus, m should satisfy the following

Table 4
Crosstalk logic design table for 3-input polymorphic gates.



Gate	C_C (fF)	Aggressor Weights				L_{Ct}	Margin Function	Logic Function	Width Ratio (P:N)
		w_1	w_2	w_3	w_{Ct}				
AND3-OR3	1	1	1	1	2	0	$CT_M(3C_C)$	AND3	1 : 2
						1	$CT_M(C_C)$	OR3	
AND3-CARRY	0.9	1	1	1	1	0	$CT_M(3C_C)$	AND3	1 : 1
						1	$CT_M(2C_C)$	CARRY	
CARRY-OR3	4.5	1	1	1	1	0	$CT_M(2C_C)$	CARRY	1 : 3
						1	$CT_M(C_C)$	OR3	
OA21-AO21	0.7	1	1	2	1	0	$CT_M(3C_C)$	OA21	1 : 2
						1	$CT_M(2C_C)$	AO21	
AND3-AO21	0.28	1	1	2	2	0	$CT_M(4C_C)$	AND3	1 : 2
						1	$CT_M(2C_C)$	AO21	
AND3-OA21	0.21	1	1	2	1	0	$CT_M(4C_C)$	AND3	1 : 2
						1	$CT_M(3C_C)$	OA21	
OA21-OR3	0.97	1	1	2	2	0	$CT_M(3C_C)$	OA21	1 : 3
						1	$CT_M(1C_C)$	OR3	
AO21-OR3	3	1	1	2	1	0	$CT_M(2C_C)$	OA21	1 : 5
						1	$CT_M(1C_C)$	OR3	
CARRY-AO21	2.2	2	2	3	1	0	$CT_M(4C_C)$	CARRY	1 : 2
						1	$CT_M(3C_C)$	AO21	
OA21-CARRY	0.6	2	2	3	1	0	$CT_M(5C_C)$	OA21	1 : 1
						1	$CT_M(4C_C)$	CARRY	

conditions,

$$L_{Ct} = \begin{cases} 0 & m \geq k \\ 1 & m \geq (k - w_{Ct}) \end{cases}$$

Therefore, the CT-margin function transforms as follows,

$$L_{Ct} = \begin{cases} 0 & CT_M(kC_C) \\ 1 & CT_M((k - w_{Ct}).C_C) \end{cases}$$

In other words, when $L_{Ct} = 0$, the inverter can flip its state only when it receives the voltage through a total coupling capacitance of $k.C_C$; therefore, the gate's logic behavior corresponds to the margin function $CT_M(k.C_C)$. However, when $L_{Ct} = 1$, an extra voltage would be applied through capacitance $w_{Ct}.C_C$, leaving capacitance margin to be only $(k-wCt)C_C$; i.e., now the inverter can flip its state just with the voltage applied due to capacitance higher than or equal to $(k-wCt)C_C$. Thus, the margin function and corresponding logic behavior can be transformed to $CT_M(k - wCt)C_C$.

We have implemented various 2-input and 3-input CT-polymorphic logic circuits. Fig. 5 shows the CT-Polymorphic AND2-OR2 Circuit and its simulation response. We list the information on the circuit design parameters: C_C, input and control aggressors' weights, along with PMOS and NMOS widths ratio for the AND2-OR2 gate in Table 3. The Table also presents the effective transformation of the CT-margin function to control logic L_{Ct} and its corresponding function.

From the simulation responses presented in Fig. 5, it can be seen that, while $L_{Ct} = 0$ the circuit operates as an OR gate, and the behavior of this particular case is indicated as CT-margin function $CT_M(2C_C)$ in the Table.

But when $L_{Ct} = 1$, the circuit functions as an AND gate, whose behavior is indicated as $CT_M(C_C)$ in the Table. Next, we have implemented ten different 3-input polymorphic circuits listed in Table 4. To limit the space, these circuits are represented by the single schematic in Fig. 6, as all the gates have uniform circuit topology with only differences in their design parameters. Information on all circuit design parameters is listed in Table 4. The simulation response of all the circuits is presented in Fig. 7, where the first panel shows Dis and Ct signals; the second panel shows the input combinations fed through A, B, and C; and the rest of the panels show the response of different gates at node F. For the AND3-OR3 circuit, inputs A, B, and C have the same coupling CC (i.e., w1 = w2 = w3 = 1), while the Ct aggressor receives $2C_C$ capacitance (i.e., $w_{Ct} = 2$). When $L_{Ct} = 0$, the margin function for the AND3-OR3 gate is $CT_M(3C_C)$, which makes it behave as AND3, as shown in Fig. 7 panel-3. Whereas, when $L_{Ct} = 1$, the C_t aggressor augments an extra charge through coupling capacitance $2C_C$ and effectively manipulates the margin function to $CT_M(C_C)$. Following the $CT_M(C_C)$ function, the transition of either A or B, or C is now sufficient to flip the inverter; thus, the gate biases and operates as an OR3 gate, as shown in Fig. 7, panel 3. It can be observed that the circuit responds as AND3 when $L_{Ct} = 0$, for the first eight input combinations (000 to 111), whereas it responds as OR3 when $L_{Ct} = 1$, during the following eight combinations (000 to 111). For the AND3 gate, if the control aggressor is given just CC coupling strength instead of 2CC in the previous case, $CT_M(3C_C)$ manipulates $CT_M(2C_C)$, which becomes polymorphic AND3-CARRY gate as shown in Table 4. The corresponding simulation response is in Fig. 7, panel 4. For the gate AO21-OA21, it is a heterogeneous-to-heterogeneous logic. In this case, aggressors' coupling weights are $w_1 = w_2 = 1$, $w_3 = 2$, and $w_{Ct} = 1$.

The margin function, $CT_M(3C_C)$, alters to $CT_M(2C_C)$ when $L_{Ct} = 1$, and gives CT-polymorphic AO21-OA21 gate (circuit response is shown in panel-6). The consecutive six gates have a logic type of homogeneous-to-heterogeneous. As an example, the aggressor weights are $w_1 = w_2 = 1$, $w_3 = 2$, and $w_{Ct} = 2$ for the gate AND3-AO21(input weights are heterogeneous). In this case the margin function for AND3 is $CT_M(4C_C)$.

The control aggressor biases it to $CT_M(2C_C)$ and operates the gate as AO21 (circuit response is in panel-7). In the previous case, if C_t is given CC strength instead of $2C_C$, the margin function manipulates from $CT_M(4C_C)$ to $CT_M(3C_C)$, giving CT-polymorphic AND3-OA21 gate as shown in Fig. 7 pane-8. Similarly, CARRY-OR3, OA21-OR3, AO21-OR3, OA21-CARRY, and AO21-CARRY results are shown in Fig. 7. Polymorphic concepts can be extended to cascaded circuits in order to implement different polymorphic functions. It can also be adapted to the module and system level. [30] presents one such module level example on a Mul-Sort-Add circuit.

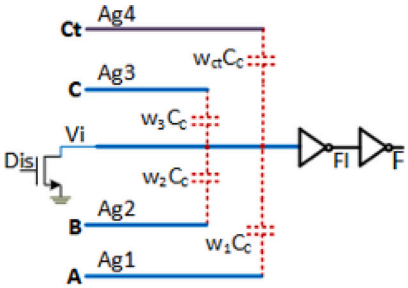


Fig. 6. 2-input crosstalk-polymorphic logic gate: (i) AND2-OR2 schematic, (ii) AND2-OR2 simulation response.

Such examples are explored in our previous work [23,30,31]. We have also discussed the possible applications of Crosstalk polymorphic function in reconfigurable system design, fault-tolerant circuit/system design [30], and radiation-tolerant circuit design [31].

4. Designing under variation at 7 nm

Crosstalk computing fundamentally relies on interference between input nets for computing. Nevertheless, transistors have a crucial function in managing the interference pattern and guaranteeing a full-swing output.

Process variation may arise due to various uncertainties in-process steps [32] and ultimately impact transistor performance, hence the standard in the industry is to name different process corners as FF (Fast–Fast), TT (Typical–Typical), and SS (Slow–Slow), where the first letter refers to NMOS and later one refers to PMOS. Through the amalgamation of FS, FT, and similar parameters, alternative process corners can be derived. Nevertheless, FF, TT, and SS indicate the best, nominal and worst-case situations. Fig. 8 shows the effect of process variation on the transfer characteristics curve of an unskewed inverter at 7 nm. As seen in Fig. 8, the inverter has a weak PMOS transistor causing the switching threshold (V_m) to shift toward zero. In general, the switching threshold voltage is desirable to be equal to exactly half of VDD (0.35 V) since this would provide a higher noise margin. For worst-case process variation (SS), V_m moves more towards the left (Fig. 8) and, thereby, increases the undefined region. Under such conditions, an incoming signal with a noisy zero value would lead to erroneous values at the output.

For Crosstalk computing, when the symmetrical inverter is linked to its Vi node, the incoming input value is evident from Fig. 8. should lie within 0 V to 0.33 V to have perfect logic '1' at the inverter output. However, to have better noise immunity and balanced drive strength, the width of the PMOS needs to be increased since this would shift the switching voltage towards half of the VDD. The properties obtained from Fig. 8 from different process corner is very useful for designing Crosstalk circuits with different fan-ins and are indicative of power and performance profile.

We used the ASAP7nm PDK [24] to evaluate Crosstalk circuits at 7 nm. ASAP7 PDK is compatible with industry CAD tools for full physical verification (Layout design, DRC, PEX, and LVS). The PDK is a 7 nm FinFET technology that comes with transistor models having four different threshold voltage levels The SLVT, LVT, RVT, and SRAM devices are presented within the ASAP7 PDK, ranked in descending

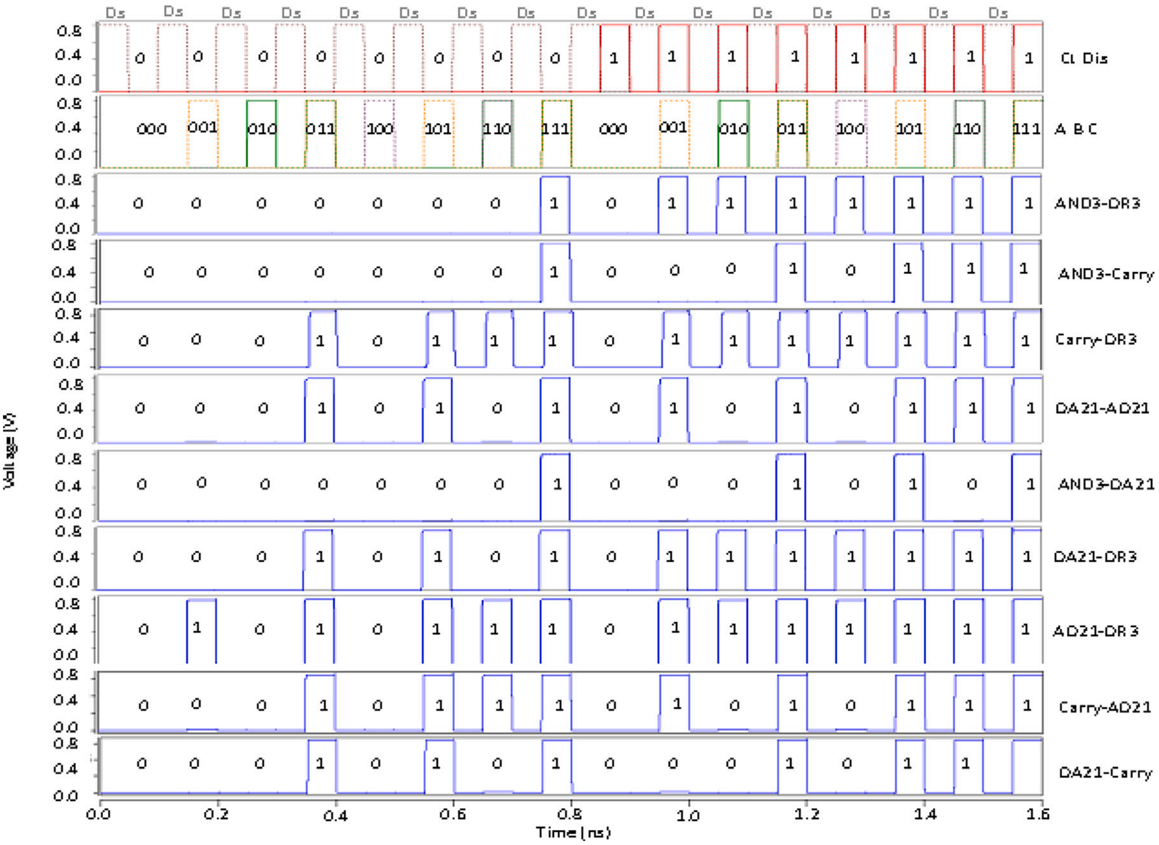


Fig. 7. Simulation responses of 3-input CT-Polymorphic logic gates.

order of drive strength, applying to both NMOS and PMOS transistors. In the case of the RVT-type NMOS transistor, it boasts an Ion measuring 37.85 μA and Ioff of 0.019 μA, resulting in an impressive subthreshold swing of 63.03 mV/decade [11].

Correspondingly, the PMOS transistor exhibits a comparable subthreshold swing of 64.48 mV/decade, featuring an Ion of 32.88 μA and Ioff of 0.023 μA. Through the utilization of these NMOS and PMOS transistors, an unskewed inverter attains a robust high noise margin of 0.33 V and a low noise margin of 0.3 V, while operating with a switching threshold voltage of 0.34 V. Fig. 9(i & ii) depicts the outcomes concerning power and performance for Crosstalk NAND and NOR gates across three process corners, considering both PMOS and NMOS devices: SS, TT, and FF. As revealed in Fig. 9(i & ii), the presence of slower transistors contributes to extended transition periods, measuring 5.53 ps and 8.77 ps for the NAND and NOR gates, respectively. Nevertheless, the essential functionality of these gates remains uncompromised.

The delay is minimum for the FF corner, but the power is also the highest. The bar graphs depict how process variation influences the performance and power of both the NAND and NOR gates, with TT representing the standard scenario

5. Designing large-scale circuits in crosstalk technology at 7 nm

In the previous sections, we have discussed how different Crosstalk primitive logic cells can be achieved and also shown that these cells can be designed at smaller technology nodes with no functionality issue. In this section, we show that Crosstalk cells can be connected in a cascaded manner for large-scale circuit design at smaller technology nodes. In Fig. 10, an illustration is provided to showcase a substantial circuit implementation of Crosstalk computation utilizing the 7 nm ASAP PDK. and Fig. 11 shows the simulation results of the circuit. The circuit is a Cm85a circuit, one of the benchmarking circuits from MCNC suits [33]. The circuit has eleven primary inputs (a-k) and three primary outputs (l-n), as shown in Fig. 10. The netlist of the circuit is obtained from the MCNC and then further simplified to get Crosstalk-friendly netlist according to the process explained in [34]. More benefits can be achieved when the Crosstalk circuits is implemented in a homogeneous (f = ab + bc + ca) or heterogeneous manner (f = a + bc) [34].

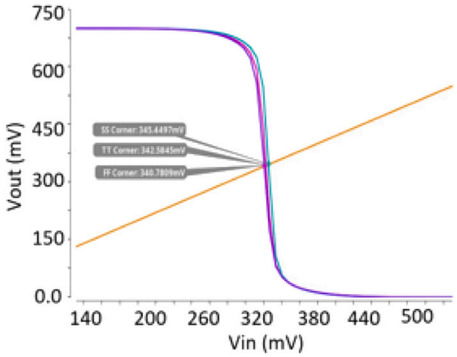


Fig. 8. Voltage characteristics curve of an inverter for different process variation.

When realizing extensive circuits using Crosstalk gates, careful consideration is necessary to uphold signal integrity and drive strength for subsequent stage gates. Addressing these concerns involves the incorporation of an inverter at the end of the victim node. The inverter acts as a thresholding function; that is, if the victim node voltage is below a certain voltage limit, the inverter restores the logic level ‘0’ and vice versa. The strength of the inverter depends on the number of fan-out

Table 5
Comparison of polymorphic technologies.

Technology	CMOS	Evolved Circuits [3]			Ambipolar NWFET [7]	Crosstalk-Polymorphic
Mechanism	Circuit duplication and use of multiplexers to select redundant blocks	A control voltage biases the circuits different operation	Temperature variation effects on devices bias the circuits to different modes	Power supply variation effects on devices biases the circuits to different mode	Band structure of the transistor is altered from p-type to n-type using a control gate	Signal Interference through interconnect crosstalk
Control parameter	Select Signal	Control Voltage	Temperature	Supply Voltage	Control voltage	Control Voltage
Process-Technology Node	16nm (independent)	0.35um (strongly dependent)			30nm (dependent)	16nm (friendly to advanced nodes)
Scalability Dependence	Synthesis	Evolution limitation (Genetic Algorithms)	Evolution limitation (Genetic Algorithms)	Evolution limitation (Genetic Algorithms)	Large scale fabrication of nanowires and reliable ambipolar property	-Crosstalk Coupling network -Noise Margins -Polymorphic Logic Synthesis
Trade-off Vs. Custom ASIC	Density, power and performance penalties for redundant blocks	Power and performance penalties and limited density benefits	Power and performance penalties and limited density benefits	Power and performance penalties and limited density benefits	Limited density benefits	Density, Power and Performance benefits
Transistor Sizing	No	Yes	Yes	Yes	No	Yes

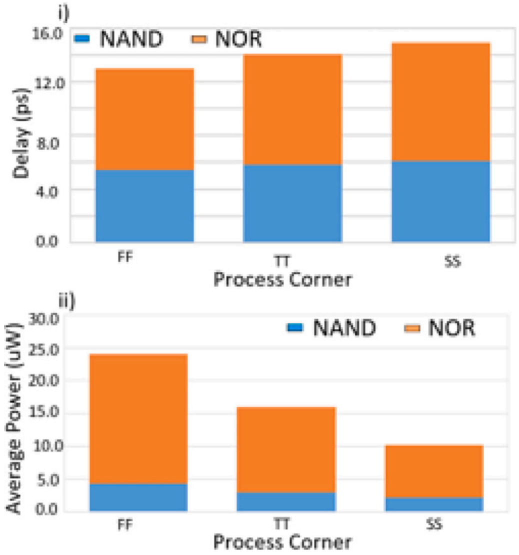


Fig. 9. Impact of process variation on (i) Performance and (ii) Average power for CT NAND and NOR gate.

loads or the type of Crosstalk gates, it is driving. As shown in Fig. 10, at the second level, Crosstalk gates drive two fan-out loads of higher coupling capacitances. To circumvent the decline in signal strength at the fan-in of succeeding stage gates, the inverters are configured as hi-skewed inverters. Another key issue, specific to the Crosstalk circuit, is during each evaluation state, the CT gates need a transition of the input signal from 0 to 1 for correct logic operation. So, if a logic high is retained on the victim node from the previous operation, it leads to logic failure. For example, in Fig. 10, at the third level, a CT-NOR gate, which is an inverting CT-logic gate, is driving the next stage CT-homogeneous gate. During the discharge (Dis) state, it receives a logic high which is carried to the next evaluation state and thereby prevents the signal transition from 0 to 1 leading to logic failure. This issue can be resolved using a Bypass circuit style [29]. In this circuit style, a transmission gate is placed between an inverting and non-inverting gate interface. In the discharge phase, the incoming input signal from the aggressor, linked to the transmission gate, is systematically discharged to the ground. In the evaluation state, the input signal is passed through the transmission-gates thus, creating a signal transition from 0 to 1. In CT-circuits employing three inputs, particularly in the case of CT-OR3 gates, robust coupling strength is imperative to ensure proper logical operation. This necessity arises due to the fact that the victim node necessitates greater charge accumulation to activate the inverter connected to it, where the inverter possesses a higher threshold voltage. However, since ASAP7 PDK, comes with transistor models with four different threshold voltages, the transistor that requires a lower threshold voltage can be used in CT-OR3 gates to avoid higher coupling.

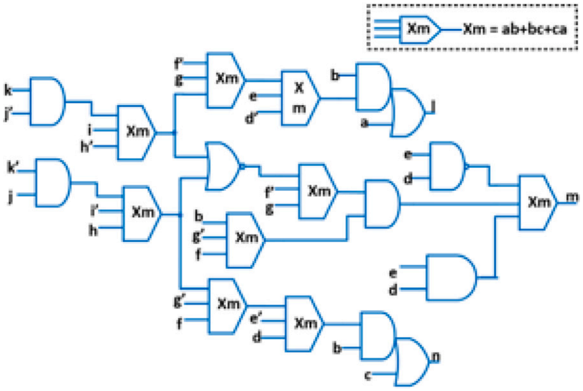


Fig. 10. Schematic of cm85a circuit.

Additionally, buffers can be used to maintain the signal strength to drive the gates that are far placed. Using the above-mentioned steps, the Cm85a circuit is implemented where only three additional buffers were needed, and simulation results in Fig. 11 show the correct functionality is maintained. Large-scale circuits can be implemented even at smaller technologies using CT-gates following a similar methodology while maintaining correct circuit functionality and achieving improved density, power, and performance benefits.

To further evaluate the potential, we have done an extensive comparison between CMOS and Crosstalk circuits. We have implemented 3 MCNC benchmark circuits and compared density, power, and performance results for CMOS at 7 nm. In the context of benchmarking circuits tailored for the 7 nm scale, notable enhancements of 48% in density, 57% in power efficiency, and 10% in performance are achieved compared to CMOS designs, as depicted in Fig. 11. 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 seen from Fig. 11, in terms of transistor count, the highest reduction is for the Mux circuit, which is 62% For Cm85a and Pcle circuits, the reduction in transistor count is 59% and 23%, respectively [29]. Crosstalk circuits show, on average, 57% power benefits over CMOS counterparts. The benefits are primarily

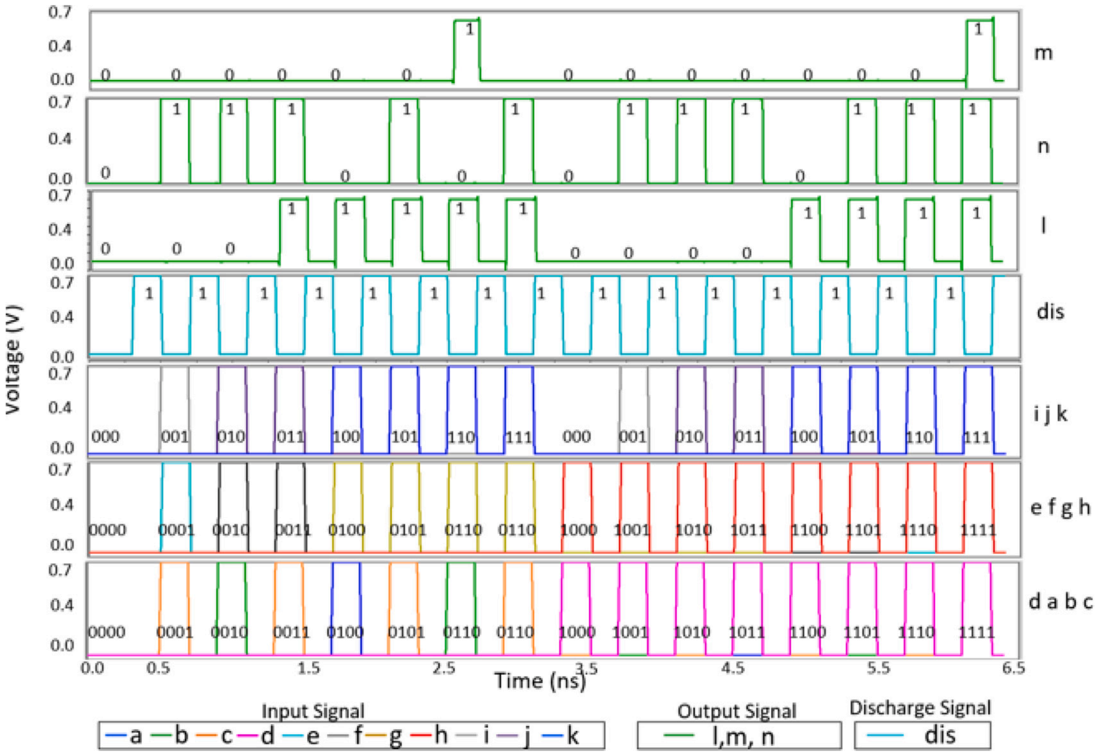


Fig. 11. Simulation results of cm85a circuit design at 7 nm.

due to the reduction in transistor count. However, the reduction in average power for the Mux is not much even though transistor count reduction is maximum compared to other circuits. This is because Mux circuit implementation requires many initializer circuits, which results in more switching activities, hence, less power reduction, and pays the performance penalty. On the contrary, for the Pcle circuit, power reduction is more because it requires fewer buffer and initializer circuits, hence less switching activity. The effect can also be seen for performance (Fig. 11), where the Pcle circuit shows better performance than CMOS circuits. However, for Crosstalk design, Cm85a and Pcle circuits have 10% and 53% improvement in performance, respectively.

6. Comparison

This section undertakes a comparison between CT-polymorphic logic circuits and prevailing polymorphic methodologies found in existing literature, elucidating their pros and cons (as outlined in Table 5). The conventional CMOS polymorphic approach (depicted in Table 5) hinges on a multiplexer-based design, where autonomous circuits are individually conceived and subsequently selected through a multiplexer. The well-known advantages of this approach are its mainstream circuit style that can be implemented in any technology node (we have designed in 16 nm), and the gate responses are robust with high noise margins. However, the drawback is that it is not inherently polymorphic and consumes redundant resources, leading to higher PPA. The unique merits of the evolved circuits, as detailed in Table 5, underscore their distinctiveness. We can construct polymorphic circuits with interesting control parameters such as temperature, supply voltage, light, radiation, etc. [1] (which are not done for Crosstalk circuit style). These attributes render them well-suited candidates for applications necessitating sensor-based functionality and adaptability. Their limitations are that the circuit functionality is technology node dependent (i.e., the same circuit topology is not adaptable to the new technology nodes, instead of needing to be evolved again); they face reliability issues and higher power and delay issues [15]. Moreover, the potential for scalability to encompass a broader array of logic functions or larger circuitry is delimited by the convergence of Evolutionary algorithms [15]. It is to be noted that the CT-circuits presented in this paper are only controlled using a control voltage. Evolution techniques can also be applied in Crosstalk Circuits to explore reconfigurability potentials based on all possible control parameters.

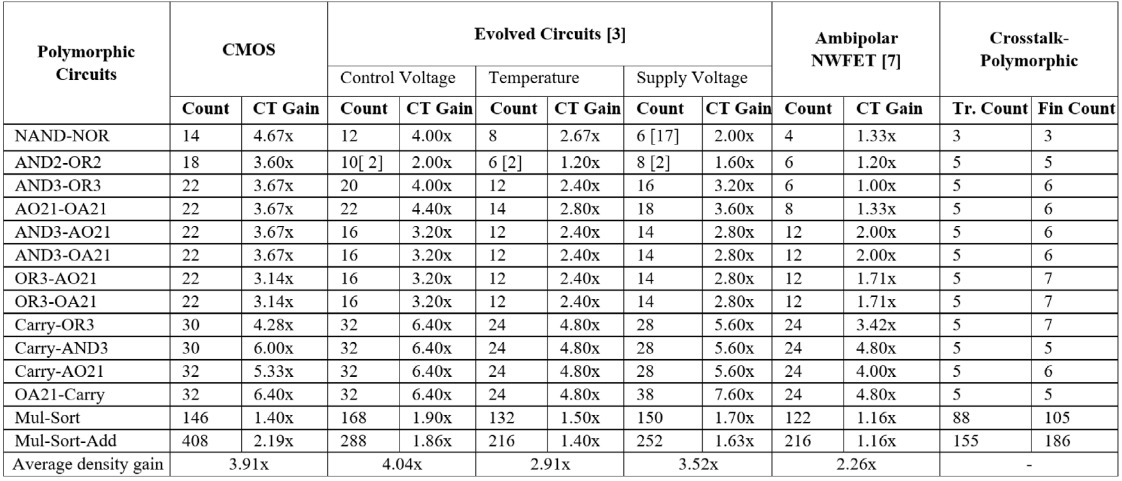
A prominent contender among the emerging reconfigurable transistors is the ambipolar Si nanowire FET (SiNWFET) introduced by De Marchi et al. [18]. In this approach, a nanowire transistor can be configured to function as either n-type or p-type through the manipulation of a control voltage. The advantage here is that the circuit topology remains aligned with the CMOS style, thereby inheriting the advantageous attributes associated with CMOS circuitry.

Nonetheless, limitations within this approach have been identified [17,18], encompassing constraints on density enhancements, the prerequisite for additional circuitry to facilitate the interchange of power rails for pull-up and pull-down networks, a less robust device response, and the need for new fabrication steps within established process flows. In CT-polymorphic circuits, the reconfigurability is achieved using the same Crosstalk aggressor–victim technique that performs the logic computation, which enables deliberate and fast reconfiguration of the gates. Despite its radically different logic and reconfigurability aspects, the working mechanism in crosstalk computing is based on well-known capacitive electrostatics, which makes it easily realizable through existing process setups and fabrication techniques.

In contrast to other unconventional device-based strategies [5], CT-Computing can be effectively accomplished through prevailing fabrication techniques. As such, it complements the traditional paradigms governing CMOS-based devices, circuits, and manufacturing. Furthermore, to the best of our knowledge, a wide range of compact single-stage and cascaded polymorphic complex logic implementations like in Crosstalk logic were not reported in other approaches.

Finally, the CT-Polymorphic approach consumes fewer transistors than any other transistor-based polymorphic circuit approach in the literature, which is discussed next. Table 6 details the transistor count in all circuit approaches and gains for CT circuits. The complex gates listed for other approaches in the Table are constructed by cascading

Table 6
Transistor count and device density gain for CT circuits compared to other approaches.



Polymorphic Circuits	CMOS		Evolved Circuits [3]						Ambipolar NWFET [7]		Crosstalk-Polymorphic	
			Control Voltage		Temperature		Supply Voltage					
	Count	CT Gain	Count	CT Gain	Count	CT Gain	Count	CT Gain	Count	CT Gain	Tr. Count	Fin Count
NAND-NOR	14	4.67x	12	4.00x	8	2.67x	6 [17]	2.00x	4	1.33x	3	3
AND2-OR2	18	3.60x	10[2]	2.00x	6 [2]	1.20x	8 [2]	1.60x	6	1.20x	5	5
AND3-OR3	22	3.67x	20	4.00x	12	2.40x	16	3.20x	6	1.00x	5	6
AO21-OA21	22	3.67x	22	4.40x	14	2.80x	18	3.60x	8	1.33x	5	6
AND3-AO21	22	3.67x	16	3.20x	12	2.40x	14	2.80x	12	2.00x	5	6
AND3-OA21	22	3.67x	16	3.20x	12	2.40x	14	2.80x	12	2.00x	5	6
OR3-AO21	22	3.14x	16	3.20x	12	2.40x	14	2.80x	12	1.71x	5	7
OR3-OA21	22	3.14x	16	3.20x	12	2.40x	14	2.80x	12	1.71x	5	7
Carry-OR3	30	4.28x	32	6.40x	24	4.80x	28	5.60x	24	3.42x	5	7
Carry-AND3	30	6.00x	32	6.40x	24	4.80x	28	5.60x	24	4.80x	5	5
Carry-AO21	32	5.33x	32	6.40x	24	4.80x	28	5.60x	24	4.00x	5	6
OA21-Carry	32	6.40x	32	6.40x	24	4.80x	38	7.60x	24	4.80x	5	5
Mul-Sort	146	1.40x	168	1.90x	132	1.50x	150	1.70x	122	1.16x	88	105
Mul-Sort-Add	408	2.19x	288	1.86x	216	1.40x	252	1.63x	216	1.16x	155	186
Average density gain	3.91x		4.04x		2.91x		3.52x		2.26x		-	

polymorphic NAND-NOR, AND-OR gates presented in [1,18]. It is noteworthy that the configuration of inverter skewing/sizing within CT circuits exerts an influence on the attainable density gain. So, for Crosstalk circuits, fin count is compared with transistor count. The CT Gain column in each approach gives the device density gain the CT circuits could achieve to the complementary approach. Among all circuit techniques compared, CMOS and ambipolar approaches do not require transistor sizing to achieve the logic (mentioned in the 6th row in Table 6). Hence, to fairly compare the device densities and derive the CT Gain, the transistor count in these two approaches is compared with the fin count in Crosstalk circuits. However, the evolved circuits stand as bespoke designs, characterized by ad hoc width ratios. These diverse transistor widths are meticulously chosen by Evolutionary algorithms to achieve the targeted polymorphism. So, to derive the ballpark gains, the transistor count in Evolutionary circuits is compared with the transistor count of CT circuits (width ratios are ignored in both cases). Thus, averaging gain for all circuits in the Table, CT Polymorphic gates achieve 3.9x, 3.5x, and 2.9x gain in device density compared to CMOS, Evolutionary, and Ambipolar circuit approaches.

Nonetheless, the considerable obstacles confronting Crosstalk polymorphic circuits are centered around scalability. These hurdles primarily involve: (i) The challenge of establishing efficient Crosstalk coupling networks. (ii) The constraints imposed by the Noise margins of the CMOS inverter, which consequently curtails the extent of fan-in achievable for these circuits. This, in turn, affects the feasibility of constructing numerous intricate CT-polymorphic logic circuits at the single-stage/gate level (where cascaded polymorphic circuits provide a solution). (iii) The necessity to develop polymorphic logic synthesis algorithms and tools that are conducive to CT-Computing, seamlessly integrating with Electronic Design Automation (EDA) workflows.

Our other works [35,36] have addressed several of these issues.

7. Conclusion

Crosstalk Logic proposes a new paradigm for circuit design, scaling, and security. The inherent hardware-level programmability feature, which allows run-time reconfiguration of computing blocks, is unique and enables enhanced security against cyber-attacks and fault resilience. We have implemented a host of simple, complex, and reconfigurable logic functions (a feature very unique to Crosstalk Computing). Our simulations and analysis work revealed opportunities for density improvements; our benchmarks showed over 48%, 57%, and 10% improvements in density, power, and performance over CMOS implementations, even with scaling down technology nodes for large-scale circuit implementations. Along with the density and power efficiency benefits for mainstream digital electronics, our configurable circuits could also spur novel hardware security, fault tolerance, resource sharing, and radiation hardening solutions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A. Stoica, R. Zebulum, D. Keymeulen, Polymorphic electronics, Evol. Syst. Biol. Hardw. 2210 (2001) 291–302.

[2] A. Stoica, R.S. Zebulum, D. Keymeulen, J. Lohn, On polymorphic circuits and their design using evolutionary algorithms, in: Proc. of IASTED International Conference on Applied Informatics AI2002, Insbruck, Austria, 2002.

[3] L. Sekanina, Principles and applications of polymorphic circuits, in: Evolvable Hardware, in: Natural Computing Series, Springer, Berlin, 2015.

[4] L. Sekanina, L. Stareček, Z. Kotásek, Z. Gajda, Polymorphic gates in design and test of digital circuits, Int. J. Unconv. Comput. 4 (2) (2008) 125–142, Philadelphia.

[5] S. Rakheja, N. Kani, Polymorphic spintronic logic gates for hardware security primitives — Device design and performance benchmarking, in: 2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), Newport, RI, 2017, pp. 131–132.

[6] A. Stocia, D. Keymeulen, V. Duong, C. Salazar-Lazaro, Automatic synthesis and fault-tolerant experiments on an evolvable hardware platform, in: IEEE Aerospace Conference Proceedings, Volume 5, 2000, 46547 I.

[7] R. Ruzicka, V. Simek, More complex polymorphic circuits: A way to implementation of smart dependable systems, ElectroScope Pilsen (ISSN: 1802-4564) 7 (5) (2013) 1–6.

[8] L. Sekanina, Evolution of polymorphic SelfChecking circuits, in: Proc. of Evolvable Systems: From Biology To Hardware, Springer, Berlin, 2007, pp. 186–197.

[9] L. Sekanina, Design and analysis of a new self-testing adder which utilizes polymorphic gates, in: Proc. of the 10th IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop DDECS 2007, IEEE Computer Society, Krakow, Poland, 2007, p. 246.

[10] A. Stoica, G. Klimeck, C. Salazar-Lazaro, D. Keymeulen, A. Thakoor, Evolutionary design of electronic devices and circuits, in: Proc. of the 1999 Congress on Evolutionary Computation, 1999.

[11] Lawrence T. Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, Greg Yeric, ASAP7: A 7-nm finFET predictive process design kit, Microelectron. J. 53 (2016) 105–115.

[12] A. Stoica, R. Andrei, Adaptive and evolvable hardware - A multi-faceted analysis, in: Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2007, pp. 486–498.

[13] A. Stoica, et al., Evolutionary recovery from radiation induced faults on reconfigurable devices, in: Proceedings of Aerospace Conference, 4 6-13 2004, IEEE, pp. 2449–2457.

[14] G.W. Greenwood, On the practicality of using intrinsic reconfiguration for fault recovery, IEEE Trans. Evol. Comput. 9 (4) (2005) 398–405.

[15] A. Stoica, et al., Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration, IEE Proc. - Comput. Digit. Tech. 151 (4) (2004) 295–300.

[16] W. Ditto, et al., Chaogates: Morphing logic gates designed to exploit dynamical patterns, Chaos 20 (2010) 037107.

[17] W.M. Weber, A. Heinzig, J. Trommer, D. Martin, M. Grube, T. Mikolajick, Reconfigurable nanowire electronics - A review, J. Solid-State Electron. 102 (2014) 12–24.

[18] M. De Marchi, et al., Configurable logic gates using polarity controlled silicon nanowire gate-all-around FETs, IEEE Electron. Device Lett. 35 (8) (2014) 880–882.

[19] J. Zhang, P.E. Gaillardon, G. De Micheli, Dual-threshold-voltage configurable circuits with three-independent-gate silicon nanowire FETs, in: Proc. - IEEE Int. Symp. Circuits Syst, 2013, pp. 2111–2114.

[20] W.J. Yu, U.J. Kim, B.R. Kang, I.H. Lee, E.H. Lee, Y.H. Lee, Multifunctional logic circuit using ambipolar carbon nanotube transistor, Proc. SPIE 7399 (2009) 739906.

[21] G. Paasch, Th. Lindner, C. Rost-Bietsch, Operation and properties of ambipolar organic field-effect transistors, J. Appl. Phys. 98 (8) (2005) US.

[22] Naveen kumar Macha, et al., A new concept for computing using interconnect crosstalks, in: 2017 IEEE International Conference on Rebooting Computing (ICRC), Washington, DC, USA, 2017.

[23] Naveen kumar Macha, Sandeep Geedipally, Bhavana Tejaswee Repalle, Md. Arif Iqbal, Wafi Danesh, Mostafizur Rahman, Crosstalk based fine-grained reconfiguration techniques for polymorphic circuits, in: IEEE/ACM NANOARCH, 2018.

[24] Saeyang Yang, Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0, Microelectronics Center of North Carolina (MCNC), 1991.

[25] S.A. Kuhn, M.B. Kleiner, R. Thewes, W. Weber, Vertical signal transmission in three-dimensional integrated circuits by capacitive coupling, in: 1995 IEEE International Symposium on Circuits and Systems (ISCAS), Vol. 1, Seattle, WA, USA, 1995, pp. 37–40, http://dx.doi.org/10.1109/ISCAS.1995.521445.

[26] W.C. Athas, L.J. Svensson, J.G. Koller, N. Tzartzanis, E. Ying-Chin Chou, Low-power digital systems based on adiabatic-switching principles, IEEE Trans. Very Large-Scale Integr. (VLSI) Syst. 2 (4) (1994) 398–407, http://dx.doi.org/10.1109/92.335009.

[27] J. Lee, M. Tehranipoor, A novel pattern generation framework for inducing maximum crosstalk effects on delay-sensitive paths, in: 2008 IEEE International Test Conference, Santa Clara, CA, USA, 2008, pp. 1–10, http://dx.doi.org/10.1109/TEST.2008.470064.

[28] Neil Weste, David Harris, CMOS VLSI Design: A Circuits and Systems Perspective, fourth. ed., Addison-Wesley Publishing Company, USA, 2010.

[29] M.A. Iqbal, N.K. Macha, B.T. Repalle, M. Rahman, Designing crosstalk circuits at 7 nm, in: 2019 IEEE International Conference on Rebooting Computing (ICRC), IEEE, 2019, pp. 1–4.

[30] Naveen kumar Macha, Bhavana Tejaswini Repalle, Sandeep Geedipally, Rafael Rios, Mostafizur Rahman, A new paradigm for fault-tolerant computing with interconnect crosstalks, in: 2018 IEEE International Conference on Rebooting Computing (ICRC), Washington, DC, USA, 2018.

[31] N.K. Macha, B.T. Repalle, M.A. Iqbal, M. Rahman, A new computing paradigm leveraging interconnect noise for digital electronics under extreme environments, in: 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019, pp. 1–8.

[32] Md. Arif Iqbal, Naveen Kumar Macha, A logic simplification approach for very large scale crosstalk circuit designs, in: 2019 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), 2019, pp. 1–6.

[33] Y. Li, C. Hwang, T. Li, M. Han, Process-variation effect, metal-gate work-function fluctuation, and random-dopant fluctuation in emerging CMOS technologies, IEEE Trans. Electron Devices 57 (2) (2010) 437–447, http://dx.doi.org/10.1109/TED.2009.2036309.

[34] NK. Macha, MA. Iqbal, BT. Repalle, M. Rahman, On circuit developments to enable large scale circuit design while computing with noise, Integration 84 (2022) 62–71S.

[35] Arif Iqbal, Naveen Kumar Macha, Bhabhana T. Repalle, Mostafizur Rahman, From 180 nm to 7 nm: Crosstalk computing scalability study, in: IEEE Electron Device Society S3S Conference, 2019.

[36] Predictive Technology Model (PTM), Predictive Technology Model (PTM), Arizona State University, 2011, n.d. Web. 03 Nov..