CROSSTALK COMPUTING: CIRCUIT TECHNIQUES, IMPLEMENTATION AND POTENTIAL APPLICATIONS

A DISSERTATION IN

Computer and Electrical Engineering
and
Computer Science

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by

Naveen Kumar Macha

B.TECH., Jawaharlal Nehru Technological University Hyderabad, Telangana, India, 2014

Kansas City, Missouri
2020

CROSSTALK COMPUTING: CIRCUIT TECHNIQUES, IMPLEMENTATION, AND POTENTIAL APPLICATIONS

Naveen Kumar Macha, Candidate for the Doctor Of Philosophy Degree

University of Missouri–Kansas City, 2020

ABSTRACT

This work presents a radically new computing concept for digital Integrated Circuits (ICs), called Crosstalk Computing. The conventional CMOS scaling trend is facing device scaling limitations and interconnect bottleneck. The other primary concern of miniaturization of ICs is the signal-integrity issue due to Crosstalk, which is the unwanted interference of signals between neighboring metal lines. The Crosstalk is becoming inexorable with advancing technology nodes. Traditional computing circuits always tries to reduce this Crosstalk by applying various circuit and layout techniques. In contrast, this research develops novel circuit techniques that can leverage this detrimental effect and convert it astutely to a useful feature. The Crosstalk is engineered into a logic computation principle by leveraging deterministic signal interference for innovative circuit implementation. This research work presents a comprehensive circuit framework for Crosstalk Computing and derives all the key circuit elements that can enable this computing model.

Along with regular digital logic circuits, it also presents a novel Polymorphic circuit approach unique to Crosstalk Computing. In Polymorphic circuits, the functionality of a circuit can be altered using a control variable. Owing to the multi-functional embodiment in polymorphic-circuits, they find many useful applications such as reconfigurable system design, resource sharing, hardware security, and fault-tolerant circuit design, etc. This dissertation shows a comprehensive list of polymorphic logic gate implementations, which were not reported previously in any other work. It also performs a comparison study between Crosstalk polymorphic circuits and existing polymorphic approaches, which are either inefficient due to custom non-linear circuit styles or propose exotic devices. The ability to design a wide range of polymorphic logic circuits (basic and complex logics) compact in design and minimal in transistor count is unique to Crosstalk Computing, which leads to benefits in the circuit density, power, and performance. The circuit simulation and characterization results show a 6x improvement in transistor count, 2x improvement in switching energy, and 1.5x improvement in performance compared to counterpart implementation in CMOS circuit style.

Nevertheless, the Crosstalk circuits also face issues while cascading the circuits; this research analyzes all the problems and develops auxiliary circuit techniques to fix the problems. Moreover, it shows a module-level cascaded polymorphic circuit example, which also employs the auxiliary circuit techniques developed. For the very first time, it implements a proof-of-concept prototype Chip for Crosstalk Computing at TSMC 65nm technology and demonstrates experimental evidence for runtime reconfiguration of the polymorphic circuit. The dissertation also explores the application potentials for Crosstalk Computing circuits. Finally, the future work section discusses the Electronic Design Automation (EDA) challenges and proposes an appropriate design flow; besides, it also discusses ideas for the efficient implementation of Crosstalk Computing structures. Thus, further research and development to realize efficient Crosstalk Computing structures can leverage the comprehensive circuit framework developed in this research and offer transformative benefits for the semiconductor industry.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a thesis titled "Crosstalk Computing: Circuit Techniques, Implementation and Potential Applications" presented by Naveen Kumar Macha, candidate for the Doctor Of Philosophy degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Mostafizur Rahman, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Masud Chowdhury, Ph.D.
Department of Computer Science & Electrical Engineering

Ghulam Chaudhry, Ph.D.
Department of Computer Science & Electrical Engineering

Dianxiang Xu, Ph.D.
Department of Computer Science & Electrical Engineering

Md Yusuf Sarwar Uddin, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

Figure Page

TABLES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Mostafizur Rahman for his consistent support and guidance throughout my Ph.D. program. His insight and envision in research and discussions brainstorming the new ideas have enhanced my out-of-the-box thinking mindset in solving the research problems. His enthusiastic encouragement, motivation, and patience have enabled me to accomplish my research goals and helped me improve the skills that I lacked. I would also like to thank my committee members, Dr. Ghulam Chaudhry, Dr. Masud Chowdhury, Dr. Dianxiang Xu, and Dr. Md Yusuf Sarwar Uddin, for their guidance and insightful comments.

I appreciate all the assistance I got from my colleagues in Nano-Computing Lab for this research work. To conclude, I cannot forget to thank my family and friends for all the unconditional support and motivation throughout my Ph.D. program.

CHAPTER 1

INTRODUCTION AND MOTIVATION

Miniaturization of Integrated Circuits (ICs), conventionally referred by Moore's Law [1][2], has been offering unprecedented socio-economic benefits. Thus, the past few decades have seen exponential growth in digital electronics capabilities primarily due to the ability to scale ICs to smaller dimensions while attaining power and performance benefits. This scalability is now being challenged [3] due to the lack of scaled transistors' performance, leakage, and manufacturing complexities [4]. Specifically, the challenges are device and material fundamental limitations such as quantum mechanical effects [5][6], short channel effects [7], process variation [3], and device parasitics [3], etc. Fig.1.1 shows the historical scaling trend of leading Chip manufacturers, TSMC [8] and Intel [9]. It can be observed from trends that the scaling used to be steep until ~130nm node, which slowed down later. Subsequently, the gate length scaling has saturated in the sub 10 nm regime facing the fundamental limitations. However, foundries were able to scale other dimensions; thus, the technology node numbers marked no longer reflect the exact channel length but the degree to which features are miniaturized. The current technology node in ramp-up (production) is 5nm,

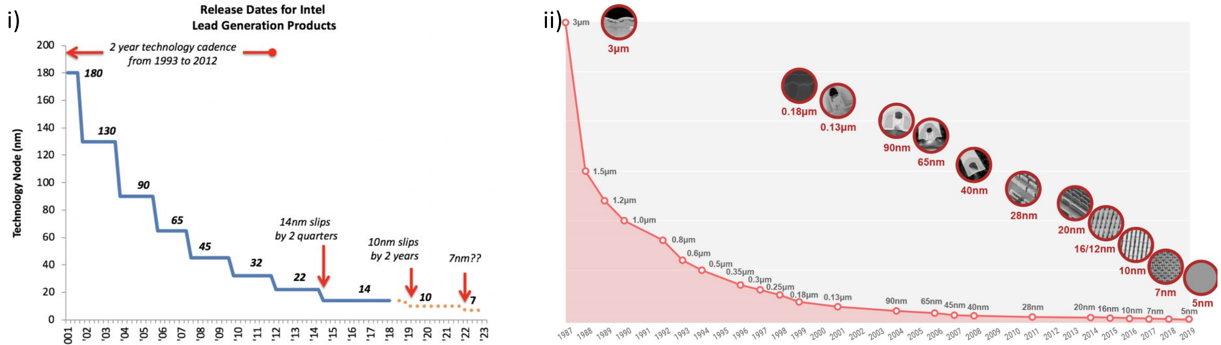


Figure.1.1 Historical Scaling trends: i) Intel Process Technologies [5], ii) TSMC Process Technologies [6]

and 3nm is in development. But the future beyond the 3nm node is gloomy.

Based on the recent research demonstrations, the potential options to continue the scaling benefits can be categorized into three types, 1) Structural innovations of the transistor, such as FINFETs [10], Gate All Around (GAA) Nanowire transistors [13-23], 2) Employing novel materials to improve power and performance [24-27], and 3) Architectural changes of transistors, such as leveraging quantum phenomena for transistor operational mechanism, ex: Tunnel FET (TFET) [28], Negative Capacitance FET (NCFET) [29], etc. (details discussion on these options is presented in the next chapter). Though these techniques might push Moore's law for few more generations, we will soon reach the ultimate atomic and quantum mechanical scaling limit irrespective of the novel channel material choices or structural and architectural changes in transistors [30].

Therefore, industry and academic researchers are actively pursuing various Beyond Moore computing researches to sustain the scaling trend and meet the ever-increasing demand for computational resources on VLSI (Very Large-Scale Integrated circuits) Chips. Some of the directions are neuromorphic computing [34-41], Quantum-dot Cellular Automata (QCA) [42], Single-Electron Transistors (SET) [43], nanomagnetic and spintronic devices [44][45], Quantum Computing [134], etc. Though these directions are promising, they lack significant technology and ecosystem development (from device and circuit to chip design) and suffer from reliability and variability issues (detailed discussion of these directions is presented in the next chapter). Moreover, some of these approaches, like Quantum Computing and Neuromorphic Computing, from their current device and circuit development, reveal that they cannot serve as a complete replacement solution for CMOS digital Chips but can only solve some specific problems efficiently. Therefore, there is a need to explore alternate computing approaches that are based not only on nanoscale mechanisms/effects but also possess the strong merits of conventional CMOS computing and provide Power, Performance, and Area (PPA) improvements over CMOS. This research uses nanoscale crosstalk signal interference for logic computation and achieves PPA benefits over CMOS. Unlike other approaches, it retains the CMOS technology by augmenting it. The next section introduces the signal interference concept, followed by the Crosstalk Computing concept.

Apart from device scaling limits discussed above, another major concern the advanced technology nodes face is the interconnect bottleneck dominating current Chips' power and performance. Moreover, dense placement of devices and interconnects with advancing technology nodes leads to the adverse proximity effect of increasing interference among neighboring signal lines due to strong capacitive coupling [135]. This phenomenon of unwanted signal interference between nearby signal carrying metal lines is traditionally called Crosstalk [136]. Figure.1.2(ii&iii) depicts the Crosstalk effect. The amount of Crosstalk induced noise is proportional to the Coupling capacitance, which is given by,

$$C_C = \epsilon \frac{(L_W x T_W)}{S_W} \quad \dots \; Eq.2(i)$$

where ϵ is the permittivity of the dielectric, L_W is the length of the wire, T_W is the thickness of the wire, and S_W is the spacing of the wire. As shown in Figure.1.2(i), the interconnects were spaced apart in older technologies; hence, signal interference was not a critical issue. However, in advanced technology nodes, as shown in Figure.1.2(ii), the signal interference among adjacent metal lines is becoming a crucial issue because of the close spacing (S_W) of interconnects. Moreover, with advancing technology nodes, increasing the vertical thickness of metal lines (T_W) has been the solution to maintain the contradictory requirement of lateral

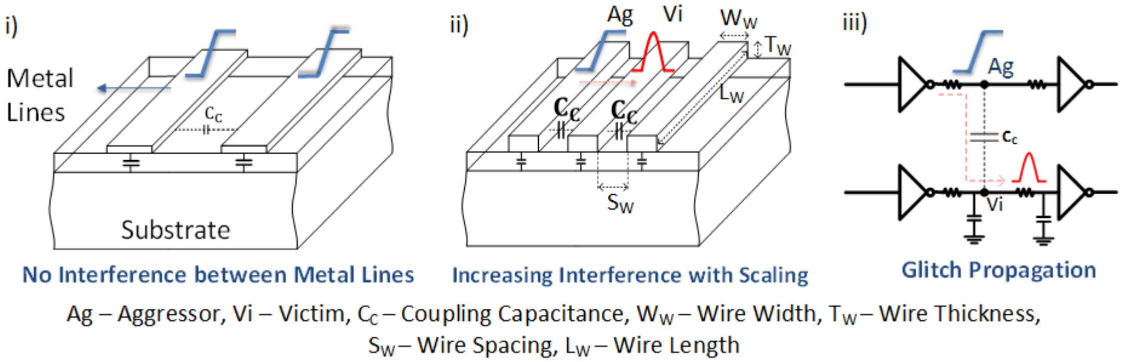

i)
Metal
Lines
C_C
Substrate
No Interference between Metal Lines
ii)
Ag
Vi
W_W
T_W
C_C
C_C
L_W
S_W
Increasing Interference with Scaling
iii)
Ag
C_C
Vi
Glitch Propagation
Ag – Aggressor, Vi – Victim, C_C – Coupling Capacitance, W_W – Wire Width, T_W – Wire Thickness,
S_W – Wire Spacing, L_W – Wire Length

Figure.1.2 Crosstalk Signal Interference: i) No signal interference in older technologies, ii) Increasing signal interference in advanced technology nodes; Aggressor-Victim scenario, iii) Circuit equivalent of Aggressor-Victim scenario

shrinkage of metal lines (W_W) and low sheet resistance (R_{SQ}). But increasing vertical thickness, T_W, of metal lines further increases the lateral capacitance, C_C, hence exacerbates the crosstalk noise. Besides, increasing lengths of semi-global and global interconnects in current IC Chips aggravate Crosstalk issues [137]. Various circuit and layout techniques [138][139] and material choices are applied by industry to damp the crosstalk; however, crosstalk is becoming incrementally inevitable in sub 10nm nodes [138]. In interconnect terminology (as shown in Figure.1.2(ii)), the driving inputs would be called Aggressors (*Ag*), and the interference capturing line would be called the Victim (*Vi*). Figure.1.2(iii) shows the equivalent circuit representation of the Aggressor-Victim scenario.

Conventionally, we treat the Crosstalk as a detrimental effect in circuits and always filter it out. However, the Crosstalk Computing style astutely tries to turn this unwanted coupling capacitance into a computing principle for digital logic gates [56][57]. Figure.1.3 shows a *Vi* net in between two aggressors *Ag1* and *Ag2*. The signal transitions on *Ag1* and *Ag2* nets will induce an effective summation signal on the *Vi* net through coupling capacitances C_C. The magnitude of the signal induced depends on coupling capacitance values. From Eq.2(i), the coupling

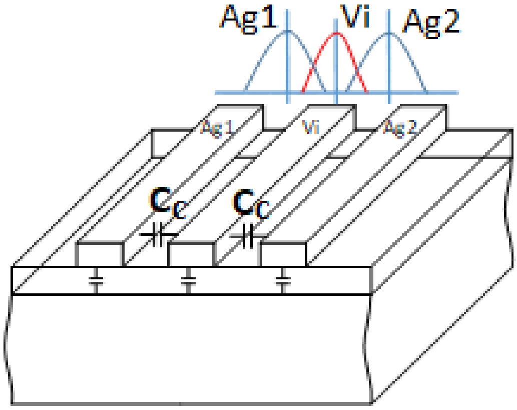


Figure.1.3 Summation of Crosstalk signal interferences

strength is inversely proportional to the distance of separation of metal lines (S_W) and directly proportional to the permittivity of the dielectric and lateral area of metal lines (which is the length (L_W) x vertical thickness (T_W) of metal lines). Tuning the coupling capacitance values using its variables discussed above provides the engineering freedom to tailor the induced summation signal to specific logic implementation or as an intermediate signal for further circuitry. A host of simple and complex logic gates are implemented using the Crosstalk Computing concept. This deterministic, tunable, and controllable signal-interference concept for computing is also extended to implement compact and efficient reconfigurable logic gates (a unique feature to Crosstalk Computing).

The innovation in this research centers on the computing principles, circuit design principles, physical structures, layout arrangements, and scaling. Figure.1.4 shows Crosstalk Computing fabric; an overview of envisioned IC implementation using Crosstalk Circuits. The interference between metal nano-lines occurs in the bottom metal layer/layers, where logic computation majorly happens. The arrangement of nanometal lines, as depicted in the inset figure, is according to logic/circuit needs. The bottom layer is for transistors required to control the output line's floating behavior and maintain signal integrity. As logic computation happens in metal lines, Crosstalk Computing requires fewer transistors to implement logic gates (regular and reconfigurable),

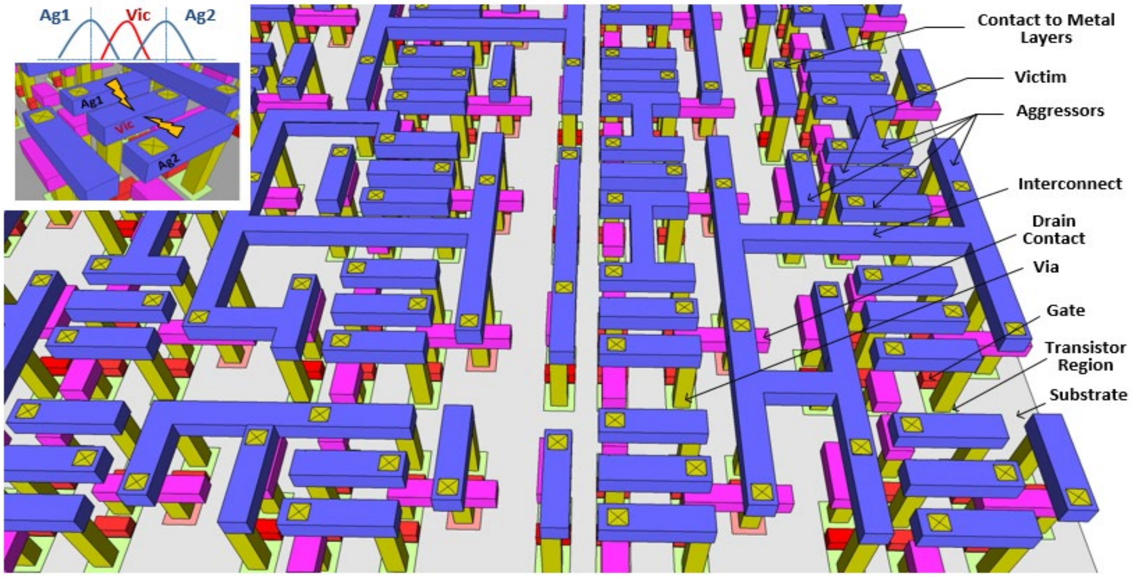


Figure.1.4 Abstract view of the Crosstalk computing fabric. The inset figure shows Crosstalk principle, where two aggressors (Ag1 and Ag2) are transitioning and as a result charges are induced in Victim line (Vi).

which is the foundation for PPA improvements. Therefore, Crosstalk circuits relax the aggressive transistor scaling requirement by alternately scaling down the circuits, which increases the circuit density.

This work also presents the design and analysis aspects of Crosstalk Circuits (simple, complex, and reconfigurable logic gates) along with simulations. The simulations and analysis work revealed tremendous opportunities for density improvements; the benchmarks showed over 48%, 57%, and 10% improvements in density, power, and performance over CMOS implementations. Along with the density and power efficiency benefits for mainstream digital electronics, our configurable circuits could also spur novel solutions in the realm of hardware security, fault tolerance, resource sharing, and radiation hardening. For example, the rise in connected devices due to the advances in Integrated Circuits (ICs) has also increased sophisticated cybersecurity threats. The ability to hack into ASIC hardware due to the de-centralized assembly of ICs makes them directly vulnerable. The Crosstalk Computing and its polymorphic circuit implementations inherently possess dynamic operation and obscurity features that can enhance hardware-security.

Therefore, with looming scaling limitations, security vulnerabilities, and lack of efficient circuit techniques for fault resilience, radiation tolerance, and resource sharing, this technology's successful realization can be transformative for the semiconductor industry.

The rest of the thesis is organized as follows. Chapter 2 discusses the current research directions and potential candidates for More Moore integrated circuits and beyond CMOS options. It finally motivates how inevitable scaling limitations of all these approaches necessitate exploring alternate novel and fundamental computing approaches like Crosstalk Computing. Chapter 3 introduces the crosstalk computing concept and presents initial pilot circuits designed to show the computing concept envisioned. Because of the pilot circuit's practical limitations, Chapter 4 develops an improved Crosstalk Computing concept and circuit style inspired by the perceptron model, which overcomes the functional limitations, and feasible. Chapter 5 presents various flavors of Crosstalk circuits possible within perceptron type circuit style. These circuits face some cascaded circuit issues that lead to functional failure. Chapter 6 discusses the problems in detail and presents the three solutions to fix these issues. The solutions are based on different perceptron type circuits, auxiliary circuits, and inherent circuit modifications. They are discussed in detail with examples, alongside trade-offs.

Chapter 7 introduces the concept of polymorphic circuits, i.e., multifunctional circuits, and surveys the existing polymorphic circuit approaches in the literature. Chapter 8 details the polymorphic/reconfigurable computing concept in Crosstalk Computing and presents a list of polymorphic circuits designed. The circuit design aspects are also detailed, along with simulations. It also shows the cascaded polymorphic circuit examples at gate-level and module-level, which establishes the feasibility of large scale Polymorphic Crosstalk circuits. Next, Chapter 9 discusses the comparison and benchmark results for Crosstalk circuits with respect to CMOS circuits and other polymorphic approaches. Chapter 10 presents the prototype demonstrating feasibility of Crosstalk Computing. The details of the Custom circuit and Chip design, post-silicon functional test, and run-time results are presented in this chapter.

Thanks to the unique, versatile polymorphism feature of Crosstalk circuits, they can be useful for mainstream digital logic circuits and some niche applications. Chapter 11 discusses these potential applications. Finally, Chapter 12 presents future work, where EDA, device, and material researches are proposed for efficient Crosstalk Computing implementation.

CHAPTER 2

MORE MOORE AND RELEVANT BEYOND CMOS RESEARCH DIRECTIONS

2.1 More Moore Research Directions

Figure.2.1(i) shows the six generations of transistor scaling [31] that are used to continue miniaturization in the sub 100nm regime. Strained Silicon and High-k metal gate stack are the material techniques. The strained silicon improved the planar MOSFETs' performance by increasing the channel mobility. The high-k metal gate stack reduced the unwanted gate tunneling current; it improved the gate control over the channel region, alleviating the short-channel-effects/drain-to-source-leakage, thus enhanced power and performance. But beyond 32nm, the techniques over planar transistor structure fall short. Because of the unsurmountable short channel effects in the planar transistors [3], the tri-gated 3-D FinFET transistors [10] were introduced from the 22nm technology node. Figure.2.1(ii) shows the planar MOSFET and FinFET structures. The 3-D structure of FinFET, with gate wrapping on three sides of the elevated fin/channel region, improves the electrostatic control of the channel region, which has bought some time for the semiconductor industry to continue

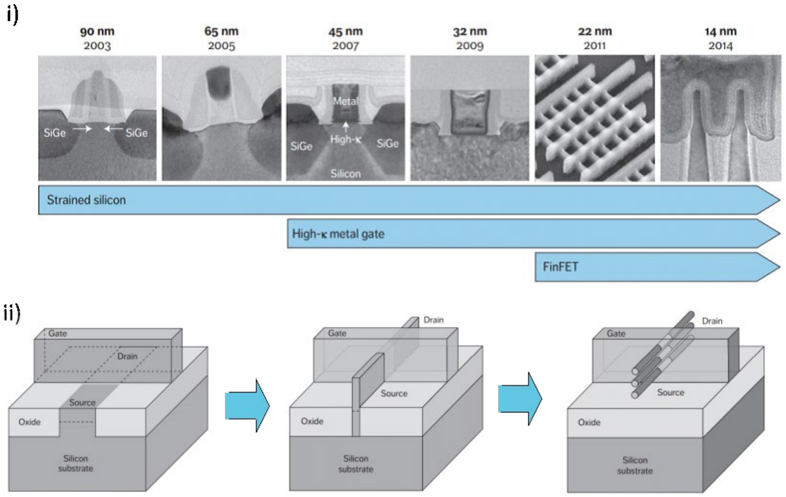
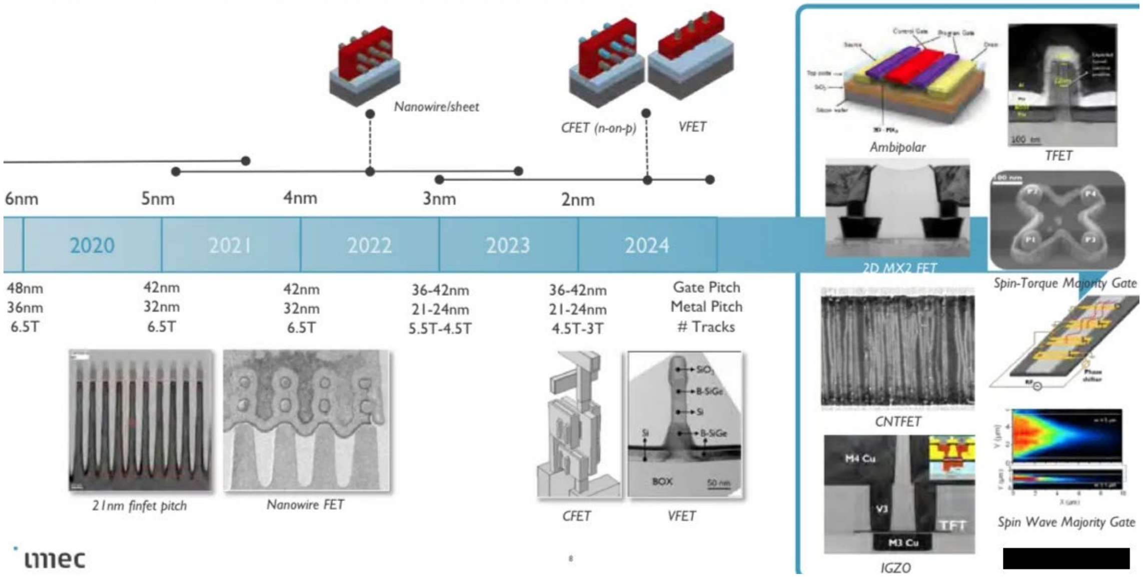


Figure.2.1 i) Six generations of transistor scaling, ii) Evolution of transistor structure: Planar MOSFET, FinFET, Gate All Around Nanowire (GAANW) transistor

miniaturization. But FinFET scaling runs out of steam beyond 5nm technology due to unviable amounts of leakage power and poor performance [11-13].

To gain ultimate electrostatic control, Gate-All-Around (GAA) nanowire/nanosheet transistors [14] (as shown in Figure.2.1(ii)) are the ultimate structural innovation with gate wrapping on four sides of the channel region. Nanowire transistors, as shown in Figure.2.2 [33], are already in the roadmap of the semiconductor industry for beyond 5nm technologies. The nanowire transistor might enable a few more generations of scaling of transistor dimensions, but the current Silicon (Si) transistor-based technology will eventually reach its performance and power limits [15]. Beyond that, there are two directions that the industry and academic researchers are pursuing to sustain Moore's law: 1) 3-D integrated circuits through vertical stacking of devices [16][17], 2) Energy-efficient transistors based on novel materials and architectures [15].



[redacted]

Figure.2.2 Imec Transistor road map (the figure is used from [33] with permission)

The vertical stacking feature of nanowires (Figure.2.2) can open up a new third-dimensional avenue to pack multiple devices vertically and increase density [18]. For example, multiple lateral fins required in traditional FinFET based circuits (for drive strength purpose) can now be achieved by nanowires in a single vertical stack, thus offer density improvements [15]. Also, researchers are developing fabrication methods and tools to enable large scale production of Complementary FETs (CFETs) [19][20], as shown in Figure.2.2. In CFETs, both p-type and n-type devices are vertically stacked in a single nanowire/nanosheet footprint. This complementary stackability enables Stacked horizontal Nanowire based 3-D (SN3D) [21] integrated circuit approaches. In SN3D, both standard cell logic gates [21] and memory (SRAM) [22] circuits can be implemented into just one or two stacks of nanowires and achieve PPA benefits. Though the nanowire stacks of more than 20 layers were demonstrated, the main factors that hurdle the number of vertical stacks are thermal issues [15] [23], mechanical strength [15], and Parasitic Coupling [15], which limits the density benefits achievable through these 3-D approaches. However, combining the 3-D IC architectures with the futuristic energy-efficient transistors and novel materials (to enhance performance and power) [15] can further push the boundaries.

The potential energy-efficient transistor candidates demonstrated in research labs are based on either novel materials or nanoscale/quantum operation mechanisms. Figure.2.2 also shows these options in the semiconductor industry roadmap [33]. Transistors based on Novel Channel Materials: germanium transistors [24], compound material (group III-V) transistors [25], emerging 2-D material transistors [26], Carbon-Nanotube FETs (CNT FETs) [27], etc. Transistors based on Novel Architectures: Tunnel FETs (TFETs) [28], Negative Capacitance FETSs (NCFETs) [29], CNT TFETs [27-28], etc. The current Si-based IC fabrication technology is very matured over 60 years and cost-effective. These emerging transistor technologies need the development of reliable manufacturing methods and tools for cost-effective mass production, and they should overcome variability issues before they can serve as a replacement solution to traditional Si-based transistors. Nevertheless, combinations of novel materials, 3-D structures, and energy-efficient architectures are the only option the semiconductor industry has for pushing the incumbent CMOS computing paradigm's boundaries.

The goal of all More Moore approaches (material, structural, and architectural innovations discussed above) is to improve the PPA of transistor-based digital ICs. Crosstalk Computing also uses transistors for threshold operation and signal boosting purposes. Therefore, Crosstalk Computing can take advantage of any innovations or transistor adoptions of future More Moore technologies. As long as the technology is transistor-based, Crosstalk Computing is implementable and can competitively provide PPA improvements over its counterpart CMOS logic implementation. However, as transistor scaling is eventually bound to hit the miniaturization and 3-D stacking limits, there are different Beyond CMOS Computing approaches pursued by researchers, which propose novel nanoelectronic devices. Some relevant beyond CMOS approaches are discussed next, and Crosstalk Computing is compared with them.

2.2 Relevant Beyond CMOS Computing approaches

Developing novel Beyond CMOS Computing approaches leveraging the atomic level phenomena or nanoscale effects can be game-changers for future digital VLSI circuits. The important research directions are Neuromorphic Computing, Emerging nanoelectronic devices for logic, and Quantum Computing. However, Quantum Computing is not considered here for discussion since it can only solve specific problems based on Quantum algorithms efficiently but not a good substitute for classical CMOS digital circuits [134].

2.2.1. Neuromorphic Computing

Neuromorphic Computing is bio-inspired and imitates the brain-like information processing to attain energy-efficient computation. The brain is 2 to 6 orders of magnitude energy efficient in certain tasks (image recognition, voice recognition, etc.) compared to the most cutting-edge supercomputers today [34]. This efficiency is because of the massive parallel computing and non-von Neumann computing architecture of the brain, where neurons act as computing elements, and their synaptic connections act as memory elements. The analog computation model and massive connections (parallelism) through synapses acting as non-volatile memory give the brain efficient computation capability. Though the software implementation of neural networks on traditional CMOS computing chips is advancing the Artificial Intelligence (AI) capabilities, they are not energy efficient because of Von Neumann architecture. Memory and computing elements are separate in Von Neumann computing, leading to significant power and performance overhead in moving data back and forth. The binary storage of data in conventional Chips also demands extensive resources on Chip, whereas the synapses in the brain store the data in analog format. Therefore, there are significant research efforts and progress in this area, where biological or artificial(software) neural-network type computing models/architectures are attempted to implement directly on hardware.

The key enabler of Neuromorphic Computing is Memristor [35], which is the memory plus resistor. The resistance of the memristor changes with the application of Voltage. Both magnitude and duration/time of the voltage changes the resistance. The resistance state is retained with the removal of voltage, thus acts as non-volatile resistive memory. Multiple reliable resistance states achievable in memristor enable it to be aptly applicable as a synapse for neuromorphic computing. Nevertheless, it can also be used as a resistive switching memory element (RAM) [36]. The popular memristor structure is comprised of Metal-Insulator-Metal layers [37]. Two metal layers are electrodes accessible as two terminals for voltage application. The input voltage varies the devices' resistance by forming and dissolving the conductive bridge of ions in the Insulator layer, the phenomenon identified as resistive switching [37].

Memristors achieving resistive switching based on different nanoscale physical phenomena are demonstrated in the literature and reviewed in detail [38]. The major ones are, i) Filamentary devices: formation of the conductive bridge of metal ions or oxide ions changes the resistance; (ii) phase-change memory: the extend of voltage-controlled amorphous region changes the resistance; (iii) heterostructure graphene/MoSO/graphene: concentration of oxygen vacancies controlled by input signal changes the resistance; (iv) Ferroelectric memristors: control of intrinsic polarization for ferroelectric materials with a control signal changes the resistance; (v) Organic electrochemical device: concentration of positive ions in an electrolyte layer changes the resistance; (vi) Spintronic devices: a) Josephson junction: the magnetic order in the tunneling barrier changes the resistance, (b) Magnetic tunnel junction: the relative orientation of the magnetic layers changes the resistance, (c) spintronic oscillators [38], etc. As all these device operations are non-charge based nanoscale phenomena, and the computation and storage are analog or multivalued, orders (>100x) of energy efficiency are projected [38]. Along with potentials, they do suffer challenges, which are discussed next.

We can mainly implement digital logic gates, neural networks, in-memory/state-full logic circuits, and memory circuits using memristors. Though it is possible to build combinational and state-full logic gates solely using memristors [39], they face severe disadvantages such as static power consumption due to current leakage, poor noise margins, signal degradation due to lack of regenerating circuit. A circuit-style combining CMOS and Memristors circuits [39] can achieve signal restoration using CMOS repeaters but still suffer from static power and noise margin issues. Therefore, the current memristor devices are inefficient in implementing logic gates. Hence in state-full logic, a hybrid implementation, where logic gates are CMOS and memory is the memristor, is efficient [40].

Similarly, to implement neural networks in hardware, a hybrid memristor plus CMOS approach is efficient [40]. In this approach, memristor crossbars serve as massively parallel synaptic connections, and neurons are implemented using CMOS based digital and analog circuits. Deep Neural Networks (DNN) are conventionally implemented in software. Although accuracy is more in software implementation, the power consumption is large, and operation is slow due to Von Neumann architecture's memory wall problem. But in hardware implementation, memristors can store multiple voltage levels or even analog levels, which increases the memory density for weights storage on Chip, thus overcomes the memory wall problem and achieves energy-efficient computation. We can write weights into the memristors in three ways [40], offline training, partially offline and partially online training, and fully online training. The training is performed offline on software in the offline method, and final weights are configured/written onto the Memristor crossbar arrays. The system is then used during the test or inference stage. An additional benefit of memristors is that we can inherently perform weighted summation of neuron inputs using current summation in crossbar columns (voltages through rows are the input signals, and memristors' conductances are the previously configured weights). Also, a hybrid approach of partially offline and partially online training is implemented [40]. A fully online technique lacks accuracy because of the device limitations such as variability in fabrication, asymmetry in memristor behavior, noise, IR drop in crossbar network, etc. [40][41].

So far, only single memristor fabrications and small scale and medium scale crossbar network prototypes are demonstrated [40]. A great deal of technological development is needed to achieve a large-scale reliable crossbar network that enables practical applications. Current full Chip memristor issues to be solved are variability, non-uniformity, asymmetry, and low yield [41][40]. Also, the power consumption of biological synapse is still orders of magnitude lower than is current memristors [40], which signifies the need for further research and improvements. Thus, Neuromorphic computing is a striving field where there are many future opportunities and potentials with ongoing research.

2.2.2. Emerging Nanoelectronic Devices for Logic

Some of the important emerging device technologies for denser logic circuit implementation are Quantum Dot Cellular Automata (QCA) [42], Single-Electron Transistors (SET) [43], Nanomagnetic Logic [44], Spin Logic [45], and Ambipolar Transistors [46], etc.

2.2.2.1 Quantum Dot Cellular Automata

In QCA, four quantum dots are arranged as a square, while two electrons are free to tunnel between any of these four dots/wells. The binary data is encoded as the position of these two electrons. Because of Coulombic repulsion, the electrons can only take the diagonal positions; thus, two diagonal positions possible are encoded as 0 and 1. The fundamental gates, such as the inverter and the Majority gate, can be formed using geometric layout techniques. The majority gate, hence majority logic, becomes the underpinning logic style for the QCA computing. The information propagates in the QCA wires through Coulombic coupling. Quantum dots made up of metals [47][48], molecular [49][50], magnetic [51][52], and semiconductor [53] are experimentally demonstrated. The benefits of QCA computing technology are high device density, ultra-low power, performance up to Thz (Terahertz). But two major drawbacks are thermal fluctuations at room temperature and defects during fabrication [42][54]. Among all, Molecular QCA might be favorable because they can operate at room temperature. The challenges to be solved to enable practical applications are catastrophic high sensitivity to defect density and variation, reliability, and fault-tolerance. Moreover, four clock signals in QCA circuits slow down the large scale designs due to sequencing and synchronization [54].

2.2.2.2 Single-Electron Transistors

Single-Electron Transistors (SET) structure is similar to regular transistors, but instead of a channel between source and drain, we have an island of potential well/dot. A single electron will tunnel from source to quantum dot at a time, then form the dot to drain. The gate voltage from the metal gate on top controls the available energy levels for electrons in the quantum dot. In the off state, when the gate voltage is 0, the dot's energy levels are higher than the source energy levels; hence, tunneling is hampered. But in on state, when the gate voltage is above a threshold, the dot's energy levels are lowered to an extent to facilitate the tunneling. Then, another electron tunnels similarly, and so on, thus conducts a single electron at a time. SET devices can be significantly miniaturized, and power consumption will be very low. They do not need some auxiliary CMOS and analog circuits to makes the signals reliable and robust [43]. However, significant challenges to overcome are sensitivity to thermal fluctuations and process variation. As SET operates at one electron's scale, the thermal fluctuations at room temperature make the electron tunneling random/stochastic. Therefore, in their current development form, these devices need cryogenic temperatures to operate deterministically, making them unsuited for regular digital Chip applications at room temperature.

2.2.2.3 Nanomagnetic and Spintronic Logic Devices

Many device ideas are proposed in the literature to implement nanomagnetic logic (NML) circuits[44] and spintronic logic circuits[45]. They use two ferromagnetic layers (one fixed and one flexible layer) or magnetic tunnel junctions to create ON and OFF states similar to the transistor. The flexible layer's magnetization direction is altered using nanoscale phenomena of spin-current, such as Spin-Hall-Effect, Spin-Transfer-Torque, and Domain-Wall-Motion. If two layers' magnetization is in the same direction (parallel), the stack offers low resistance, leading to ON state, and vice-versa in OFF state. The drawbacks of these technologies are i) switching speed is slower than CMOS, ii) static leakage and ohmic dissipations due to current based operation, which decreases the power savings, and iii) poor reliability and interconnect speed.

2.3 Crosstalk Computing vs. Beyond CMOS approaches

Unlike Neuromorphic computing, Crosstalk Computing serves as a replacement solution to the CMOS logic implementation. Memristors cannot implement logic circuits efficiently and reliably, but Crosstalk logic circuits are efficient and reliable. Thus, Crosstalk Circuits can also be used in the neuromorphic circuits in place of CMOS digital logic circuits. The improvements over CMOS are 5x density, 2x switch energy and 1.5x performance. However, crosstalk circuits cannot perform the analog or current summation-based computations like memristors, and the multistate/analog non-volatility feature is unique to memristors.

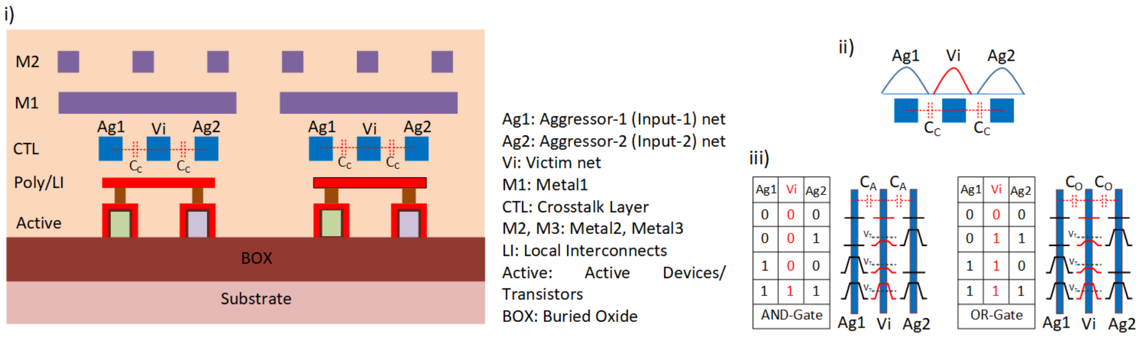
As QCA operates at the electrons' level, they are projected to improve Power greater than five orders, Performance and Area, 10x and 50x, respectively [55]. Similarly, Nanomagnetic and Spintronic devices can also offer enormous power and area improvement, but they are slower than CMOS and Crosstalk implementations [44][45]. Though Crosstalk Computing cannot offer energy-efficient operations as projected for emerging beyond CMOS devices, they do provide significant improvements over CMOS and coexist with CMOS technology.

Interestingly, Crosstalk Computing does not require complete technology development, as in the case of emerging technologies. It can be realized with the established manufacturing setups of foundries; it leverages the existing fabrication flows, techniques, tools, and materials. As Crosstalk Computing uses the Silicon-based fabrication technology matured over 60 years, the defects and variability issues will be less and not as catastrophic as other emerging technologies; thus, reliability and fault tolerance will be better compared to the other emerging nanoelectronics devices and circuits. Like CMOS, Crosstalk Computing uses the capacitance charge to encode information. Hence it is not fatally sensitive to thermal fluctuations; consequently, it does not need cold or cryogenic temperatures to operate. Nevertheless, if the challenges faced in emerging technologies are overcome in the future, these atomic-scale devices and circuits will obviously have great potential to replace the incumbent CMOS digital circuits. Motivated by immediate solution and benefits of Crosstalk Computing, the subsequent chapters develop the comprehensive circuit technology, promising for More Moore and Beyond CMOS computation paradigms.

CHAPTER 3

CROSSTALK COMPUTING

Figure.3.1(i) shows an overview of Crosstalk Computing Fabric, which majorly comprises four components, Crosstalk Layer, Active Devices, Interconnects, and Vias. The Crosstalk layer that computes the logic is a metal layer/layers composed of capacitively coupled metal lines called Aggressors (inputs) and Victim (output). Interconnects and Vias serve their common purpose, along with their contribution to coupling capacitance in Crosstalk Layer. The active devices depicted are FinFETs on SOI substrate. The transistors' goal is to accurately control and reconstruct signals, which would be discussed in the following sections. Figure.3.1(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 rare end aggressor metal lines (*Ag1* and *Ag2*) induce a resultant summation charge/voltage on the victim metal line (*Vi*) through capacitive coupling C_C. Since this phenomenon follows the charge conservation principle, the victim net voltage is deterministic and possesses the information about signals on two aggressor nets; its magnitude depends on the coupling strength between the aggressors and the victim net. The coupling capacitance is directly proportional to the relative permittivity of the dielectric and lateral area of metal lines (length times the vertical thickness of metal lines) and inversely proportional to the distance of separation of metal lines. Tuning



Ag1: Aggressor-1 (Input-1) net
Ag2: Aggressor-2 (Input-2) net
Vi: Victim net
M1: Metal1
CTL: Crosstalk Layer
M2, M3: Metal2, Metal3
LI: Local Interconnects
Active: Active Devices/ Transistors
BOX: Buried Oxide

Ag1	Vi	Ag2
0	0	0
0	0	1
1	0	0
1	1	1
AND-Gate		

Ag1	Vi	Ag2
0	0	0
0	1	1
1	1	0
1	1	1
OR-Gate		

Figure.3.1 (i) Overview of Crosstalk computing fabric, (ii) Crosstalk Computing Mechanism, (iii) Implementing Logic Gates through crosstalk Computing

the coupling capacitance values using the variables mentioned above provides the engineering freedom to tailor the induced summation signal to the specific logic implementation.

The notion of implementing logic gates using crosstalk signal interference is depicted in Figure.2.1(iii) with AND and OR gate examples. Input signal transitions induce a voltage proportional to coupling capacitances. For AND gate, C_A (Figure.3.1(iii)) can be chosen such that the magnitude of the voltage induced is greater than a selected threshold voltage V_T (which differentiates logic levels 0 and 1) only when both inputs transition 0→1 (i.e., input combination 11). For single-input changes (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. Thus, as shown in Figure.3.1(iii), AND gate functionality can be realized using the crosstalk signal interference mechanism. Similarly, OR gate functionality can be realized by increasing the coupling capacitance, which can be done by appropriately tuning the physical dimensions or choice of high-k dielectric material or both. The intuition for OR gate implementation is also shown in Figure.3.1(iii). The coupling capacitance C_O for the OR gate is greater than the AND gate (i.e., $C_O > C_A$), such that the transition of either of the input signal from 0 to 1 is sufficient to induce a voltage above the logic threshold (V_T). Therefore, input combinations, 01, 10, and 11, computes to logic output 1, as an OR gate. Practical realization of Crosstalk logic circuits and their reliable and robust operation in cascaded topology requires additional circuit techniques to be augmented to the intuitive idea described above, which is presented next.

3.1 Pilot Circuits based on pass transistors

The first version of Crosstalk circuits, the pilot circuits, is built solely by employing the Crosstalk Computing principle to demonstrate the deterministic logic behavior. The practical challenges,

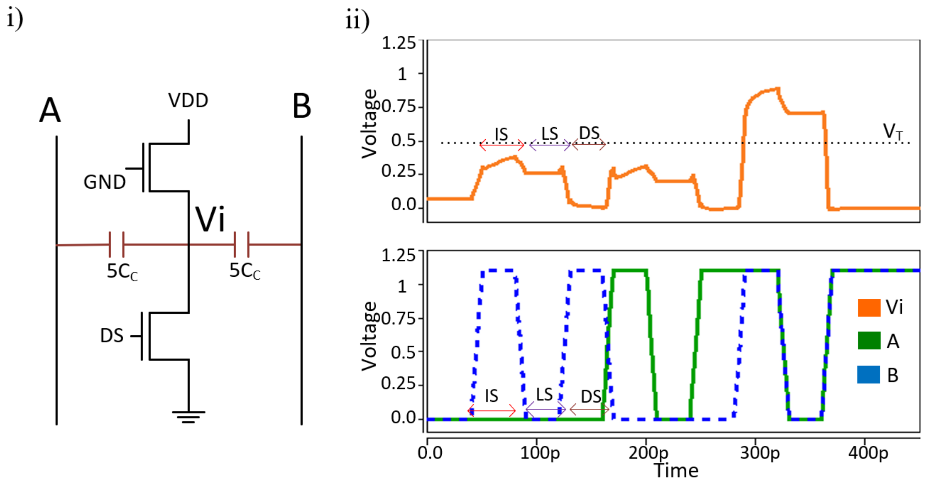


Figure.3.2 Crosstalk based AND gate: i) Circuit schematic, ii) Simulation response

however, are addressed in later chapters, incrementally improving the circuits. A simple AND gate implementation is shown in Figure.3.2(i) [56]. The circuit operation contains three states: Input State (IS) when Inputs are fed through aggressor nets (*Ag1* and *Ag2*); Logic State (LS) when logic is evaluated; and Discharge State (DS) when floating nodes in the circuit are periodically discharged to ground hence gaining control over the floating nodes. Figure.3.2(ii) shows the simulation response of the AND gate. It can be noticed from the figure that inputs *A* and *B* transition on aggressor nets during IS, while logic is evaluated during LS, and during DS, the *Vi* node is discharged to zero. The output

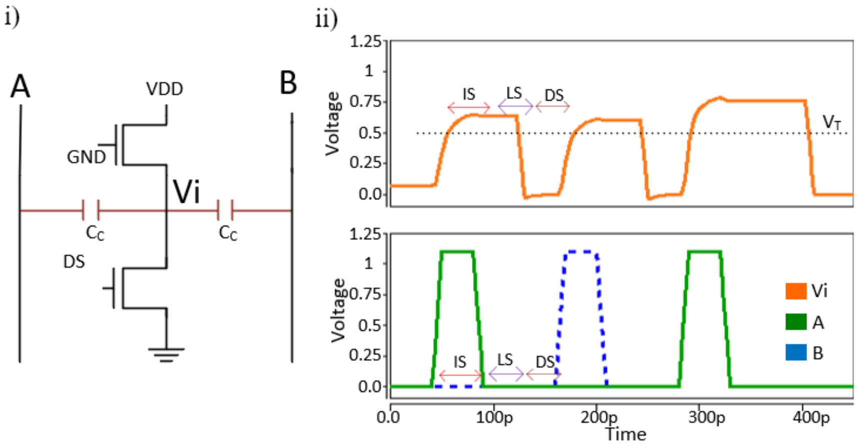


Figure.3.3 Crosstalk based OR gate: i) Circuit schematic, ii) Simulation response

(*Vi*) node voltage goes above the selected threshold V_T only when inputs combination is (1,1); the low bumps during (0,1) and (1,0) inputs are assumed to be 0 as they are below the threshold. Figure.3.3 shows the Crosstalk OR gate implementation [56]. Figure.3.3(i) depict schematic, and Figure.3.3(ii) is the simulation response. It can be seen that the *Vi* node voltage goes above the V_T for all three input combinations (0,1), (1,0), (1,1); thus, it creates the OR logic pattern.

Similarly, the logic implementation of a non-linear circuit like XOR is shown in Figure.3.4 [56]. Figure.3.4(i) is the schematic, and Figure.3.4(ii) is the simulation response. For the XOR gate, the

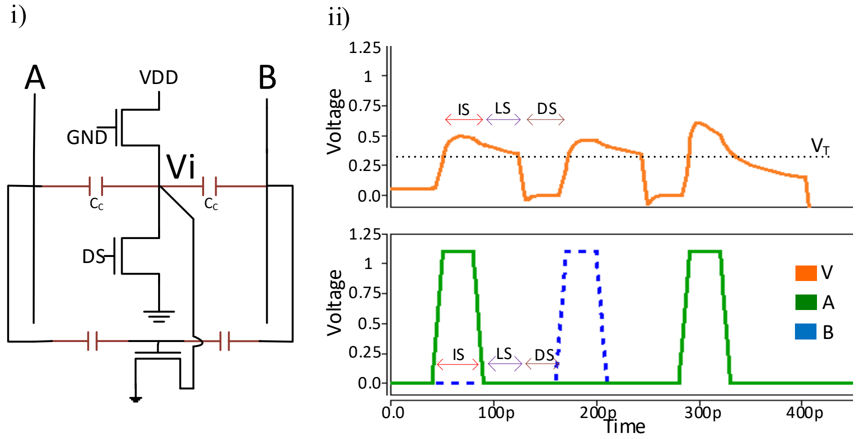


Figure.3.4 Crosstalk base XOR logic gate: i) Circuit schematic, ii) Simulation response

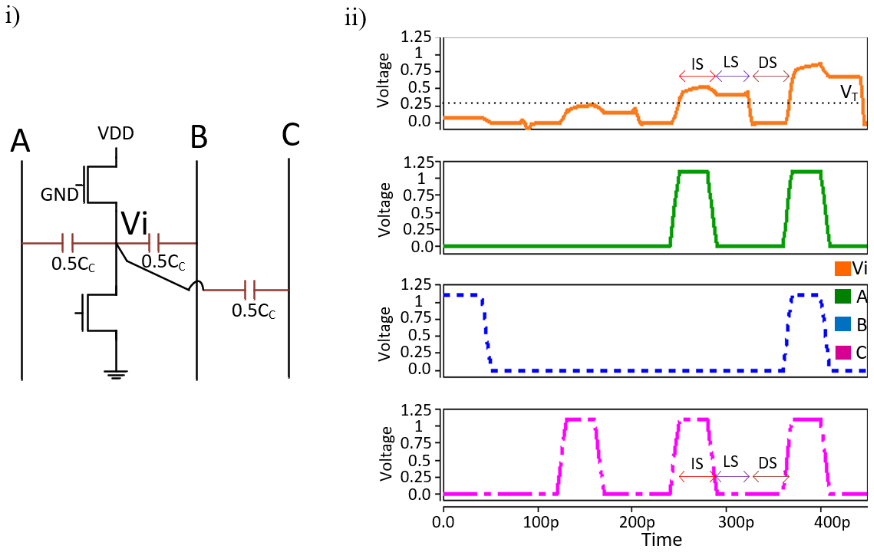


Figure.3.5 Crosstalk base CARRY logic gate: i) Circuit schematic, ii) Simulation response

output during input combination (1,1) reaches zero gradually during LS. Figure.3.5 shows the Crosstalk implementation of CARRY logic [56], whose Boolean expression is (*AC+B(A+C)*). As shown in the schematic Figure.3.5(i), a third aggressor net would serve as the *C* input. The simulation response of the CARRY circuit is shown in Figure.3.5(ii). We can observe that the output is above the selected threshold, i.e., 1, when two or more inputs are 1.

Figure.3.6 shows the stick diagrams of the CARRY circuit in CMOS circuit style (Figure.3.6(i)) and Crosstalk circuit style (Figure.3.6(ii)). It can be observed that Crosstalk circuits need only 2 transistors while CMOS needs 12 transistors, and Crosstalk logic consumes less footprint of .044 mm^2 while CMOS consumes 0.13 mm^2 footprint (the area numbers are estimated from stick diagram applying the minimum rules). The CARRY circuit implementation presented shows the potential of Crosstalk circuits with high fan-in logic. Table.3.1 shows transistor count and area comparisons of all logic circuits. Also, emerging 3-D approaches [4] provide additional 3-D architectural benefits to implement Crosstalk Logic circuits. However, these circuits can only demonstrate the Crosstalk Computing concept but suffer significant drivability and signal integrity issues in practical circuits. The *Vi* node, which is assumed to be the output node, might be connected to any fanout load in actual

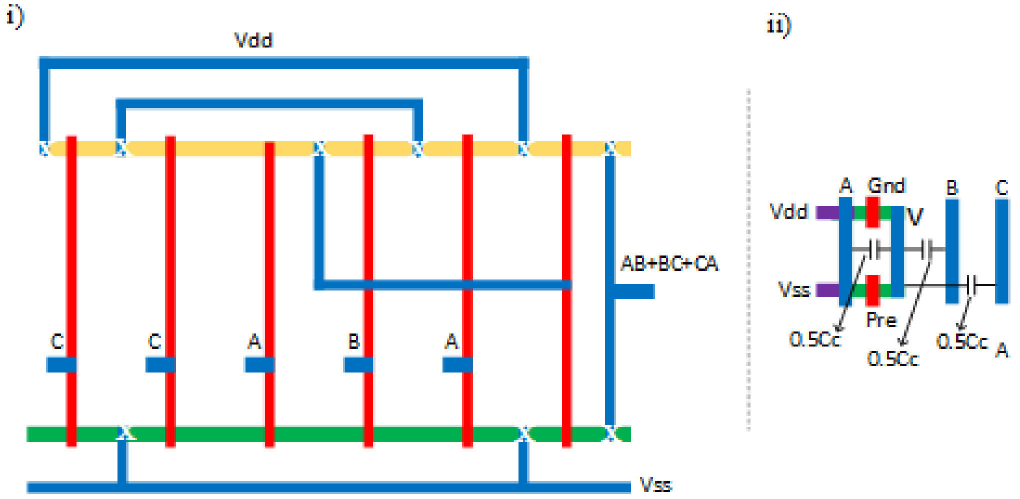


Figure.3.6 Stick diagrams for Carry Circuit i) 2D CMOS circuit style ii) Crosstalk circuit style

TABLE 3.1
Transistor Count and Area Measurement for CMOS and Crosstalk AND, OR, XOR Gates

Logic Gate	Transistor Count		Area	
	CMOS	CT	CMOS	CT
AND	6	2	0.069	0.011
OR	6	2	0.089	0.011
XOR	14	3	0.24	0.018
Carry	12	2	0.13	0.015

circuits. In that case, the voltage induced on Vi net itself depends on the ratio of Aggressor-to-Victim (Input-to-Output) Coupling capacitance to total capacitance on the Vi net, which is,

$$V_{Vi} = \left(\frac{k. C_C}{n. C_C + C_P + C_L} V_{DDD}\right)$$

where, C_C is the unit coupling capacitance value specific to each gate. All input aggressors receive coupling strengths in multiples of C_C. k is the integer multiple that quantifies the amount of total coupling strength through which Vi net experiences voltage induction, i.e., 0 to 1 transition. n is the integer multiple that quantifies the total coupling capacitance associated with the Vi net. C_L is the next stage fanout load attached to the Vi net. C_P is the internal/cell-level device and net parasitic capacitance on the Vi net. Assuming $C_{Int} = n. C_C + C_P$, which is the total internal load on Vi net, the voltage induced becomes,

$$V_{Vi} = \left(\frac{k. C_C}{C_{Int} + C_L} V_{DDD}\right)$$

Therefore, if the fanout load increases, the Vi net voltage will reduce. Thus, a gate logically functioning correct standalone, i.e., zero loads, will not work functionally correct when loaded. Also, as the output node is floating during IS and LS, it lacks drive-strength to drive any output load in the actual circuits, and it is vulnerable to noise. Thus, the output node faces signal integrity issues. The next chapter discusses a different circuit-style that overcomes the above signal integrity challenges and creates robust signals at the output node.

CHAPTER 4

CROSSTALK CIRCUITS BASED ON PERCEPTRON MODEL

The majority of the conventional digital circuits adopted the CMOS circuit style for reliability and robustness reasons. The CMOS circuit-style implements the logic gates by using transistors as gated devices/switches. However, an artificial neural network model, such as a perceptron model (inspired by biological neural networks), shown in Figure.4.1, can serve as a mathematical model that can inspire to implement computing circuits differently. Figure.4.2(i&ii) shows the Crosstalk Circuits implemented based on the mathematical model of perception [57]. The aggressor signals are the inputs, coupling strengths are the weights, summation happens through signal interference, and a CMOS inverter acts as an activation function. As any logical operation can be implemented using one or more Perceptrons, any logical function can also be implemented with the Crosstalk (CT) Circuits.

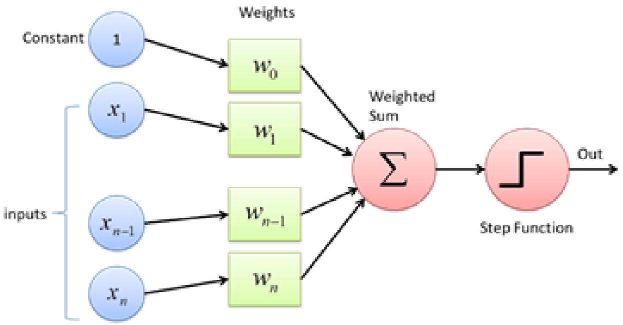


Figure.4.1 Perceptron Model

4.1 Basic Logic Gates

Although properly engineered coupled nano-metal-lines and very few pass transistors are sufficient to emulate the logic behavior in Crosstalk Computing [56], the output net (*Vi*) collecting 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

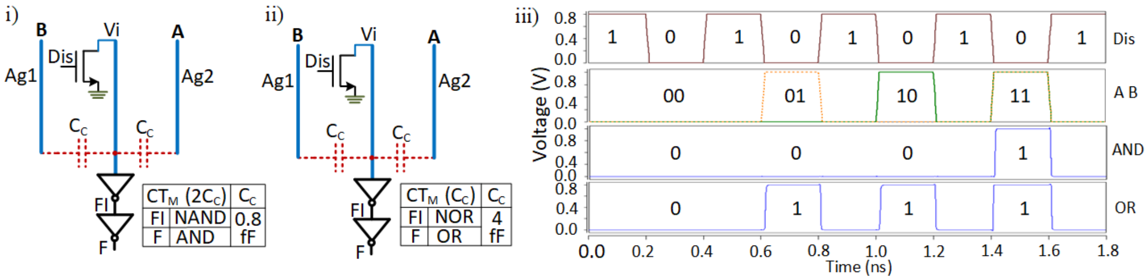


Figure.4.2 Crosstalk Basic Gates: (i) AND gate schematic, (ii) OR Gate Schematic, (iii) Simulation response of AND and OR gates.

known initial state. Second, it should remain floating during logic evaluation to collect the crosstalk charge. Third (as discussed in the previous chapter), the output node should be able to drive the fanout gates in real circuits and maintain the signal integrity of binary voltage levels. As shown in Figure.4.2(i)&(ii), 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. Figure.4.2(i) shows the 2-input AND gate in which input aggressor nets (*A* and *B* acting as *Ag1* and *Ag2*) are coupled to *Vi* net through coupling capacitances of value C_C (C_C values are given in Table.4.1). The *Dis* signal drives the discharge transistor. The Crosstalk logic gates operate in two alternate states, Discharge state (DS) and logic Evaluation state (ES).

TABLE 4.1
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

During DS (enabled by *Dis* signal), the floating victim node is shorted to ground through the discharge transistor and starts with a known initial condition, i.e., 0. The alternate DS states ensure the correct logic operation during every ES state by clearing the previous logic operation's charge. During ES state (*Vi* net is floating), the rise transitions on aggressor nets induce a proportional linear summation voltage on the *Vi* net, connected here to a CMOS inverter. The inverter acts as a regenerative threshold function. That is if the voltage computed on Vi 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 tunable by changing PMOS to NMOS width ratios if required. The Crosstalk logic gates presented in this paper are designed using the 16nm Predictive Technology Modeling (PTM) [58] transistors and simulated on SPICE. The simulation response of the designed AND gate is shown in Figure.4.2(iii), where the first panel shows the discharge signal (*Dis*), the second panel shows two input signals (*A* and *B*) with *00*, *01, 10, and 11* combinations given through successive ES stages (i.e., when *Dis=0*). The third panel shows the output response of the AND gate. For all the circuits, the *FI* node gives inverting logic output (NAND, NOR, etc.), and the *F* node gives a noninverting logic result (AND, OR, etc.). Similarly, OR gate implementation is shown in Figure.4.2(ii), and the simulation response is shown in panel-4 of Figure.4.2(iii) (the input signals in panel-1 and panel-2 are shown common for both the circuits to limit the space). The difference between AND and OR gates is their coupling strength (C_C). As given in Table.4.1, C_C is greater for OR gate than AND, i.e., 0.8fF for AND and 4fF for OR. C_C is the quantized capacitance specific to each gate. The input aggressors would receive the coupling strengths in integer multiples of C_C.

Crosstalk logic gates' operation would be represented functionally using a crosstalk-margin function, $CT_M(C_C)$, specifying that the Crosstalk gate's inverter 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, as shown in the table adjacent to the schematic (Figure.4.2(i)), 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 only happens when both inputs are high. Similarly, for the OR gate (Figure.4.2(i)), the CT-margin function is $CT_M(C_C)$, which means the transition of any one of the aggressors is sufficient to flip the inverter, thus execute the OR behavior.

The Crosstalk circuit design and working mechanism for all logic types will be explained using the CT-margin function in this thesis. Therefore, to further elucidate the relationship between CT-margin function and working mechanism of CT-logic gates, consider a generic crosstalk capacitive network with ‘n’ number of input aggressors, as shown in Figure.4.3. The voltage induced on the victim net can be calculated by applying KVL, as follows,

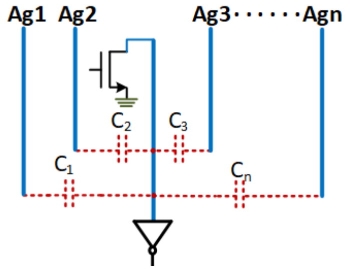


Figure.4.3 Capacitive Network of a Generic Crosstalk Gate

$$V_{Vi} = \left(\frac{C_1}{C_T}V_1 + \frac{C_2}{C_T}V_2 \ldots + \frac{C_n}{C_T}V_n \right) \ldots (Eq\ 4.I)$$

where, $C_1, C_2 \ldots C_n$, are capacitances from respective aggressors to the Vi net. C_T is the total capacitance on Vi net, which is,

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

$$C_{INV} = \text{Inverter Gate Capacitance,}$$

$$C_{ds} = \text{Discharge transistor drain to source capacitance}$$

The final voltage levels on input aggressors, which are given by $V_1, V_2 \ldots V_n$ in equation (*4.I*), can be formulated as voltage sources, given by,

$$V_i = L_i V_{DD};$$

where, L_i represents the logic level, i.e.,

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

The capacitances given to input aggressors are in integer multiples of a constant *C_C* specific to 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 equation (I) now modifies to

$$V_{Vi} = \frac{C_C}{C_T}.V_{DD}.m\ ...\ (4.II)$$

where *m* = $w_1L_1 + w_2L_2... + w_nL_n$; *m* evaluates to integer values. The CT-margin function of each gate can be related to *Vi* net voltage as follows. Consider a given logic gate is associated with the CT-margin function $CT_M(k.C_C)$, where *k* takes integer values. Then for all the input combinations that produce logic output 0, the *Vi* net voltage computed is greater than the inverter trip point ($i.e., V_{Vi} > V_{INV}$) and *m* is greater than or equal to *k* (i.e., m ≥ k). Conversely, for all input combinations that produce logic output 1, *m* is less than *k (i.e., m < k),* and *Vi* net voltage is less than the inverter trip-point ($V_{Vi} < V_{INV}$). Table 4.1 gives the logic design table for AND2 and OR2 gates, which lists the *C_C* values, aggressor weights, and margin function. It also lists the PMOS to NMOS ratio for two gates. The logic design table summarizes the mechanism and circuit aspects of crosstalk logic gates.

3.1 Complex Logic Gates

More interesting complex logic functions can be realized by increasing the fan-in (i.e., the number of input aggressors) because of the increased coupling capacitances and CT-margin function choices. The circuit schematic of a generic 3-input Crosstalk gate is shown in Figure.4.4(i). Table.4.2 is a Logic Design table that lists *C_C* and *w_i* values, CT-margin

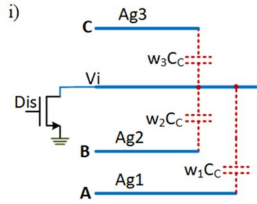


TABLE 4.2
CROSSTALK LOGIC DESIGN TABLE FOR COMPLEX GATES

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

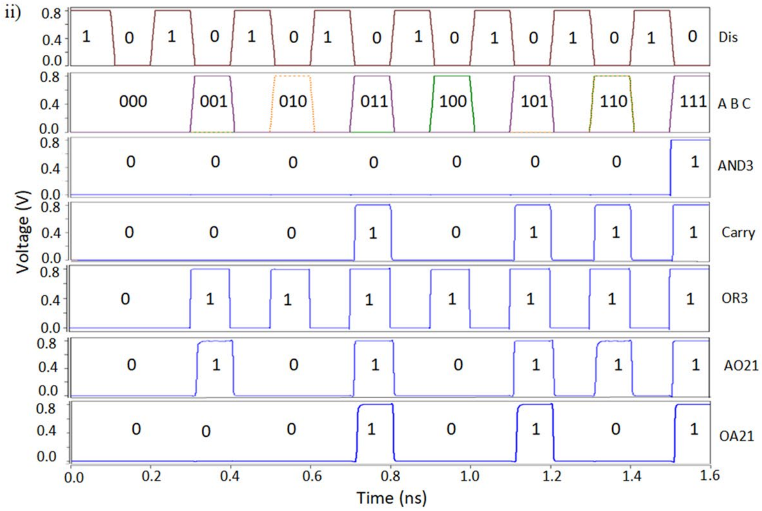


Figure.4.4 Crosstalk Complex logic Gates: (i) A generic schematic representing all 3-input complex logic functions ii) Simulations response of 3-input complex logic functions (AND3, CARRY, OR3, AO21, OA21).

function, and PMOS to NMOS width ratio for all 3-input complex-logic functions that are implemented. For logic design, each gate receives a specific quantized C_C value and different aggressor weights(w_i) as given in the table. The input aggressors can be assigned equal or unequal coupling capacitances. Gates with equally coupled aggressors are called homogeneous Crosstalk logic gates, and unequally coupled aggressors are called heterogeneous Crosstalk logic gates. These homogenous and heterogeneous coupling choices further enhance the scope of complex logic functions that can be implemented efficiently through the Crosstalk Computing mechanism. Starting with homogeneous input aggressor weights, that is when $w_1=w_2=w_3=1$, the Crosstalk margin function of $CT_M(3C_C)$ implements a 3-input *AND* gate, which implies output is 1 only when all three inputs are 1. A CT-margin function of $CT_M(2C_C)$ implements *CARRY (AB+BC+CA)* logic, which means output is 1 when any of the two inputs or all three inputs are 1. The CT-margin function of $CT_M(C_C)$ implements a 3-input *OR* gate, which implies output is 1 when any one or two or all three inputs are 1. The simulation responses of 3-input Crosstalk logic gates designed are shown in

Figure.4.4(ii). The first panel shows *Dis* pulse, the second panel shows the three input signals, *A*, *B,* and *C*, feeding all input combinations from *000* to *111* in successive ES states (i.e., when *Dis=0*). The third, fourth, and fifth panels show the simulation responses of Crosstalk *AND3*, *CARRY*, and *OR3* gates, respectively, for the corresponding inputs in panel-1 and panel-2. Next, by giving weighted/heterogeneous couplings to the input aggressors such that one input has stronger capacitance than the other two (i.e., $w_1=w_2=1$ and $w_3=2$), the functions like *AO21* and *OA21* can be realized as given in Table.4.2. Logic expression of *AO21 (AB+C)* evaluates to 1 when either *AB* or *C*, or both are 1. That means the output is biased towards the input *C*; the output is *1* when *C* is *1* irrespective of *A* and *B* values. Therefore, input *C* has twice the coupling capacitance than *A* and *B*. Similarly, for *OA21* function *(A+B)C*, the output is again biased towards input *C,* i.e., for output to be *1*, *C* should be *1* along with *A+B*. Therefore, as in the previous case, *C* receives twice the coupling $(w_3=2)$ than *A* and *B*. To satisfy the above logic relations, the margin functions for *AO21* and *OA21* gates are $CT_M(2C_C)$ and $CT_M(3C_C)$, respectively. The simulation responses of *AO21* and *OA21* gates, satisfying the logic for all input combinations (*000* to *111*), are shown in panel-6 and panel-7. Thus, it can be observed from the CT-logic-design table (Table.4.2) that a variety of complex logic functions can be generated by engineering the aggressor weights and inverter threshold-voltage.

CHAPTER 5

CROSSTALK CIRCUIT TYPES

The Crosstalk signal interference between two signal-carrying metal lines will happen only when there is a relative change in the two nets' potential/voltage. Therefore, Crosstalk circuits can detect a new logic-level only through signal transition, and logic computation can happen only when input signal changes. The types of signal transitions can be positive (0→1), negative (1→0), or both. Thus, based on the different transition types of signals, the Crosstalk circuits have been classified as the following styles.

5.1 Positive-Transition Crosstalk Circuits

Figure.5.1 depicts the positive-transition Crosstalk circuits. In these circuits, the Vi node's initial state and of all input aggressors during DS state is 0. During the ES state, the input signals will transition from 0→1 on the aggressor lines, which results in charge

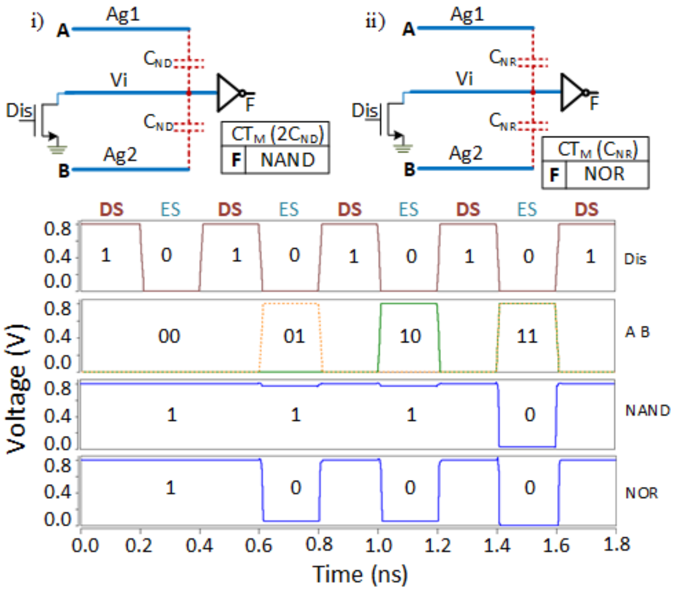


Figure.5.1 Positive Transition Crosstalk Circuits.

accumulation on the victim node. The 0→1 transition on inputs contribute charge on to the *Vi* node, thus identified as the logic 1. Whereas if logic 0, there is no signal transition and no contribution of charge. Therefore, the working mechanism in positive-transition Crosstalk circuits is through the charge summation principle. Figure.5.1 shows an example of such circuits. When signal *A (Ag1)* or *B (Ag2),* or both the signals, transition form 0→1, the charge accumulates on the victim node and results in either NAND or NOR gate based on the input aggressors' coupling strengths and inverter trip-point. The discharge transistor drain is connected to the victim line, and the source is connected to GND. In this way, the victim node is refreshed to ground potential before the arrival of any signal. Previous stage gates will maintain the initial state of 0 on input aggressors.

5.2 Negative-Transition Crosstalk Circuits

The negative transition Crosstalk circuit mechanism is based on high to low signal transition. Figure.5.2 shows the example of a negative-transition Crosstalk circuits. The

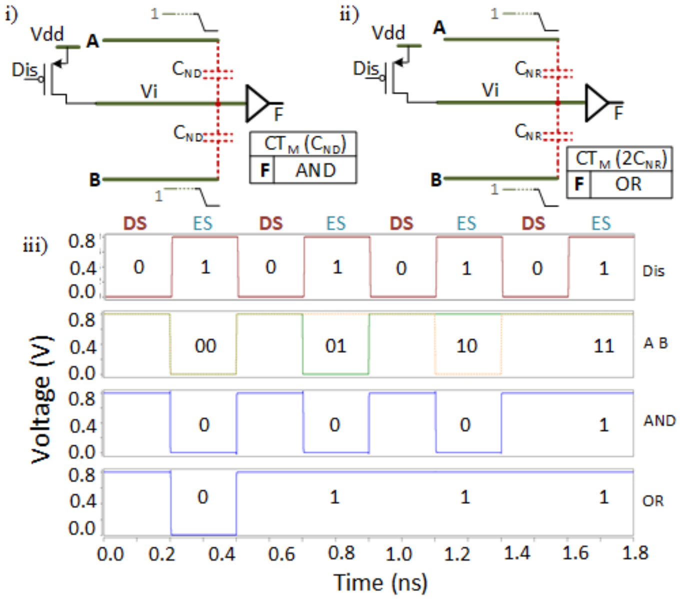


Figure.5.2. Negative Transition Crosstalk Circuits.

Crosstalk circuit's main working principle remains the same, except this time, initial states are set to logic high. In this case, the input logic 1 does not lead to any transition on aggressors. Still, logic 0 makes 1→0 transitions and deterministically subtracts the Vi node's charge, which is also precharged to logic high. In complement to NMOS discharge transistor in positive transition circuits, a PMOS transistor (Figure.5.2(i&ii)) is used as the Precharge transistor, where the source is connected to the Vdd, and the drain is connected to the victim net. Another end of the victim net is connected to a buffer to achieve the AND and OR gate. Figure.5.2(iii) shows that the ES state in Negative transition gates is when the Discharge signal is high (opposite to Positive Transition gates).

It can be noticed from the schematic figures that the margin functions of Crosstalk negative-transition gates (AND and OR) are in complement to Positive transition Crosstalk Gates. $CT_M(C_{ND})$ for AND gate specifies that the output flips its state from 1 to 0 (as output's initial state is 1) when the victim net experiences 1→0 transitions from a total coupling capacitance greater than or equal to C_{ND}, which happens when either one or both the inputs transition 1→0. Similarly, $CT_M(2C_{NR})$ for OR gate states that the output is 0 only when the victim net experiences 1→0 transitions from a total coupling capacitance greater than or equal to $2C_{NR}$, which happens only when both the inputs have 1→0 transitions. Also, the coupling strength $C_{ND} > C_{NR}$ in case of negative-transition Crosstalk gates, whereas $C_{NR} > C_{ND}$ in case of Positive transition Crosstalk gates. Previous stage gates will maintain the initial state of 1 on input aggressors.

5.3 Dual Transition Crosstalk Circuits

The dual transition Crosstalk Circuits leverage both high-to-low and low-to-high signal transition types for logic implementation purposes. In this mechanism, both positive and

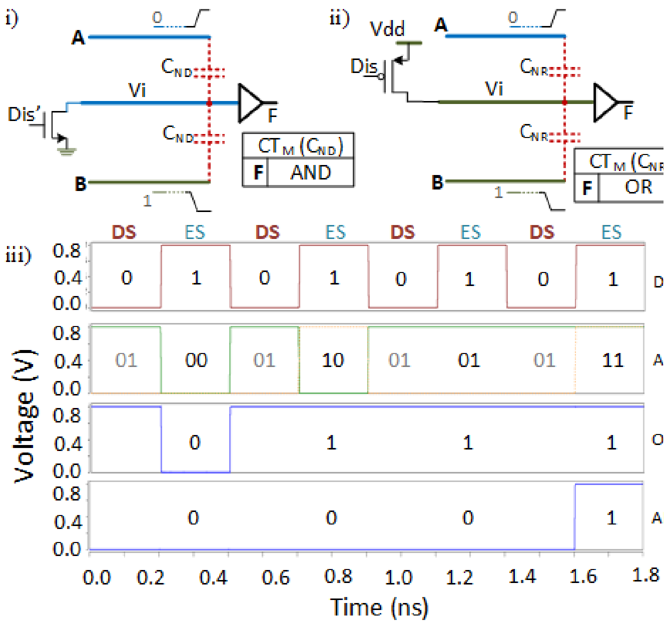


Figure.5.3 Dual Transition Crosstalk Circuits.

negative transition aggressors co-exist within a single gate. In Figure.5.3(i)&(ii), the blue color indicates a positive transition net having 0 as the initial state, and green color indicates a negative transition gate having 1 as the initial state. The main differences in AND and OR gates are the arrangement of the Discharge transistor and the Vi node and aggressors' initial states. As depicted in Figure.5.3(i), the AND gate uses a discharge transistor (NMOS connected to the *GND*), and the OR gate (Figure.5.3(ii)) uses a precharge transistor (PMOS connected to the *VDD*). Figure.5.3(iii) shows the simulation response of AND and OR gates with the correct functional response.

In both circuits, the initial state of input *A* is logic 0, and the initial state of input *B* is logic 1. Input *A* adds the charge when input logic is 1 while input B removes the charge when input logic is 0. *A* and *B* do not affect the *Vi* net when their input logic levels are 0 and 1, respectively; with these inputs, they just maintain their previous logic levels from initial states, leading to no signal transitions (therefore no output change). The margin function for the AND gate is $CT_M(C_{ND})$, which means 0→1 transition on input *A* can flip the inverter state provided the *B* does transition in the negative direction (1→0) and subtract the charge; this can happen only during 11 input combinations (on A and B), thus enables the AND gate design. Note that input *B* experiences only 1→0 but not 0→1 in ES state due to its initial state 1. The margin function for OR gate is $CT_M(C_{NR})$, which means 1→0 transitions on input *B* can flip the inverter provided the input *A* does not transition in the positive direction (0→1); this can happen only during 00 input combinations (on *A* and *B*), thus enables the OR gate design. Note that input *A* never transitions 1→0 in ES state.

5.4 Bypass-Branch Crosstalk Circuits

Crosstalk circuits are analogous to the Perceptron model [6], where the weighted summation of input variables passed through an activation function defines the output. The crosstalk signal interference on the *Vi* net through different coupling strengths is equivalent to weighted input variable summation, and the inverter's threshold function acts as the activation function. Since it is impossible to implement linearly inseparable logic functions (such as XOR, etc.) with a single Perceptron, it would be impossible to implement them with a single-stage Crosstalk gate. However, we can cleverly fix the inseparable problem using a bypass-branch circuits and achieve single-stage XOR circuits. As shown in Supplementary Figure.5.4, the bypass-branch uses stacked PMOS or NMOS transistors acting as AND logic. Four different bypass circuit styles are proposed in the figure. Without a bypass-branch, Figure.5.4(i) is just a NAND gate (which is evident from its margin function $CT_M(2C_{XR})$). NAND gate can be converted to XOR if the gate is somehow tricked for 00 input combination to give 0 as the output instead of 1. During ES state (Figure 5.4(i)), when *A*=0 and *B*=0, the two PMOS transistors turn on and pull the *Vi* node to 1 instead of 0 based on Crosstalk

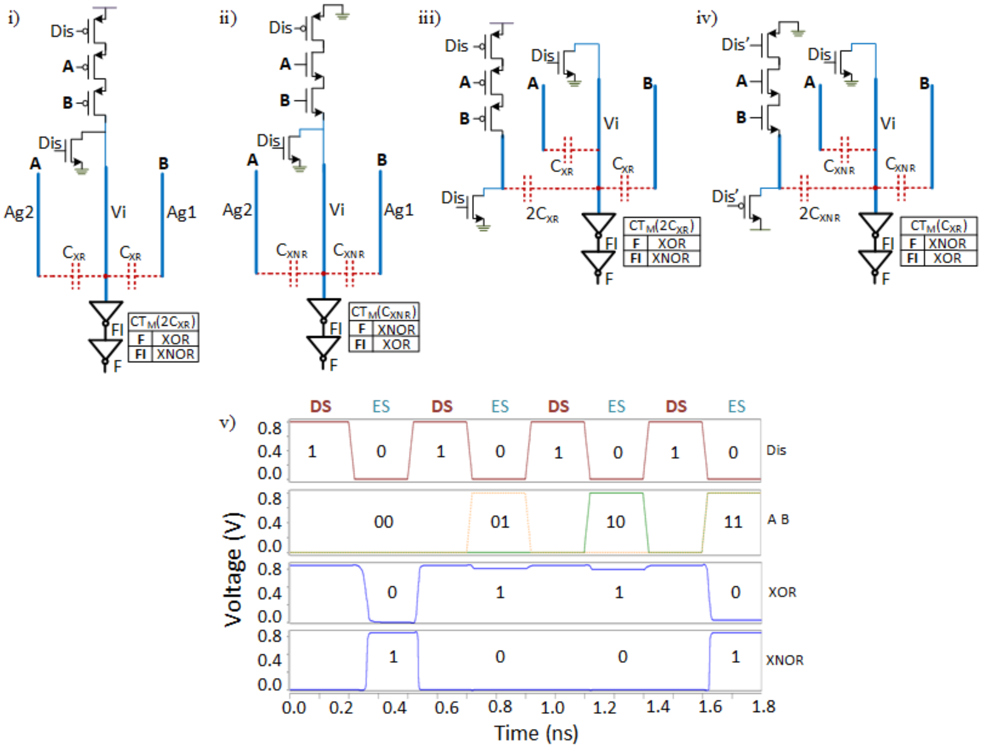


Figure.5.4 Bypass Branch Crosstalk Circuits.

Computing. Thus, the gate behaves as an XOR gate (at node *F*) instead of the NAND gate.

Similarly, Figure.5.4(ii) is a NOR gate (which is evident from its margin function $CT_M(C_{XNR})$). NOR gate can be converted to XNOR if it is tricked for 11 input combinations to output 1 instead of 0. During LE state (for Figure.5.4(ii)), when *A*=1 and *B*=1, the two NMOS transistors turn on and pull the *Vi* node to 0, instead of 1 based on Crosstalk Computing. Thus, the gate behaves as an XNOR gate (at node F) instead of NOR gate. Figure 5.4(iii&iv) are XOR and XNOR gates similar to the last two gates, but the bypass branch is now implemented on the aggressor side instead of the *Vi* node. Figure.5.4(v) shows the simulation result of XOR and XNOR gates using the bypass-branch circuit style. In all cases of Figure.5.4, the usage of an extra transistor driven by *Dis* signal in the bypass branch is to avoid VDD and GND shorts possible during the DS state (when *Dis=1*).

Though the simulation responses are functionally correct, such stacking or bypass branch mechanism has one main disadvantage: leakage current through the bypass branch. When the *Vi* node floats during ES state, the leakage current seeps through the bypass branch and accumulates as the unwanted charge on the *Vi* node, which can lead to logic failure over time. This problem can be overcome by tightening the timing window of ES state, which in turn constricts the overall timing constraints. Besides, this technique increases the transistor overhead due to the bypass branch and increases the delay due to stacking, which are the trade-offs to achieve single-stage non-linear logic gates.

CHAPTER 6

CASCADING CIRCUIT ISSUES AND SOLUTIONS

6.1 Cascading Circuit Issues

The cascading of Crosstalk circuits requires special attention. Crosstalk circuits bring in certain transition constraints that necessitate additional circuit techniques in cascaded circuit topology. As seen in previous sections, for correct logic evaluation, both aggressor and victim nodes need to be either at zero or high before the evaluation period (in DS state) and undergo high (0→1) or low (1→0) transitions, respectively. In the above discussions, the initial state requirement for the *Vi* node is addressed by the discharge/precharge transistor. However, the initial state requirement for input aggressors is mentioned to be taken care of by the previous stage gate or an additional circuit. We will discuss how to take care of these requirements for three kinds of Crosstalk circuits, positive, negative, and dual transition Crosstalk circuits.

For positive transition gates (shown in blue color), input aggressors need to be at logic 0 initially in DS state. This requirement is automatically met if the given gate is driven by non-inverting logic gates (such as AND, OR, etc.), as shown in Figure.6.1(i); this is because, for all non-inverting gates in Crosstalk circuits, the output is 0 in DS state. Conversely, if the gate is driven by the inverting gates, as shown in Figure.6.1(ii), the driving gates' DS state outputs are 1. Therefore, if subsequent ES state evaluates to logic 1, the input aggressor has no 0→1

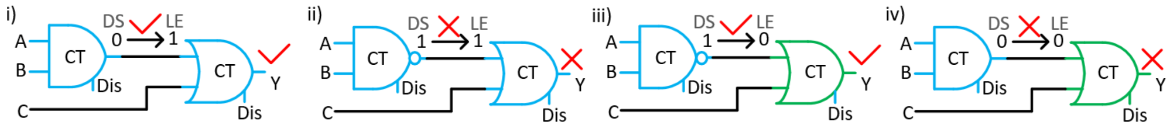


Figure 6.1 Cascading Circuit issues: i)No Transition issue by connecting nodes with same initial state (0) ii) Mismatch node by connecting initial-state-one output to initial-state-zero input iii) No Transition issue by connecting nodes with same initial-state (1) iv) Mismatch node by connecting initial-state-zero output to initial state-one input.

transition (it remains 1→1). Hence, logic 1 will not be detected for logic computation in the gate, leading to incorrect output. Such connection of gates and corresponding nodes can be called as initial-state mismatch-nodes (in this case, logic 0 mismatch-nodes). Initial-state mismatch nodes will be simply called as mismatch nodes in the subsequent discussion.

Similarly, for negative transition Crosstalk gates, all input aggressors require 1 as the initial state. For dual transition Crosstalk gates, some of the input aggressors require 1 as the initial state and others 0. As shown in Figure.6.1(iii), inverting gates driving the negative transition gate (shown in green color) inputs automatically meet the initial state requirement (i.e., 1). But, as shown in Figure.6.1(iv), if a non-inverting gate drives these inputs, there will be again state mismatch (logic 1 mismatch-nodes), leading to logic failures. Because the practical circuits have many such cascaded configurations/connection points, we can formulate two rules that will enable functionally correct cascaded circuits. We can observe that all crosstalk gates have an initial state (either 0 or 1) for their inputs and outputs. Rule number one is that we can connect two nodes (output pin to input pin or input pin to an input pin) only if their initial states match. As mismatch-nodes are inevitable in practical circuits, rule number two is that if there are any mismatch-nodes, they should be connected through some auxiliary interface circuits that can fix the mismatch problem.

6.2 Solutions to fix Mismatch Nodes

This section presents three types of solutions which are incrementally optimal. The first technique is to use some auxiliary/additional circuits at mismatch node interfaces. The second technique is to leverage different Crosstalk Circuit types (positive, negative, and dual transition gates) to fix the mismatch nodes automatically. Finally, the third technique is to modify the CMOS inverter in Crosstalk circuits such that the mismatch nodes at the output are fixed inherently. We can simultaneously employ all these techniques for optimal circuit design.

6.2.1 Auxiliary Initializer Circuits to fix Mismatch Nodes

As these auxiliary circuits' primary purpose is to initialize the input aggressor nodes to the required logic state (0 or 1), it is appropriate to call them as initializers. Figure.6.2 shows all the proposed auxiliary initializers. Figure.6.2(i) shows the Input-Low-Initializer (ILI) to fix logic 0 mismatch nodes, and Figure.6.2(ii) shows the Input-High-Initializer (IHI) to fix logic 1 mismatch nodes. The initializer circuit is a combination of transmission gate and discharge/precharge transistors. The transmission gates in both the circuits (Figure.6.2(i&ii)) turn off in the DS state and thus isolates input aggressor node (*Ag*) from the mismatch signal driven by the previous stage-gate (*Dr*). In the case of ILI (Figure.6.2.(i)), the NMOS will tie

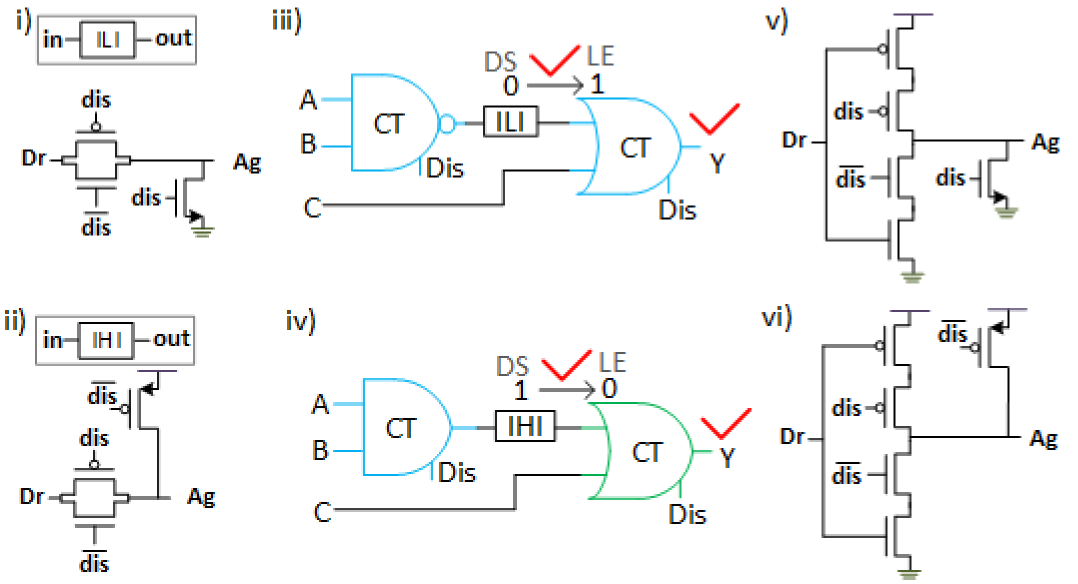


Figure.6.2 Initializers: i) Input-Low-Initializer, ii) Input-High-Initializer, iii) Using ILI, iv) Using IHI, v) Regenerative ILI, vi) Regenerative IH

the *Ag* node to logic-low in DS state, ensuring that the next state gate's input aggressor receives only logic-low in DS state.

Similarly, in IHI, the PMOS transistor will tie the *Ag* node to logic-high in the DS state. Figure.6.2(iii&iv) shows how ILI and IHI circuits can be employed to fix the circuits in Figure.6.1(ii&iv), respectively. Figure.6.2(v&vi) are regenerative versions of ILI and IHI circuits. The only difference is that they use tristate-inverter in place of transmission gate; thus they can be employed in circuit scenarios where drive-strength and signal-integrity are the concerns (for example, when fanout is large).

Next, the mismatch node issue and its fixture using the initializer circuit are demonstrated through simulations. Figure.6.3(i) shows a scenario of logic 1 mismatch node (node *X*) where the NAND gate drives an AND gate. Figure.6.3(ii) employs an ILI circuit to fix the logic 1 mismatch node. Figure.5.4 shows the simulation responses. Panel-3 and panel-4 (from top)

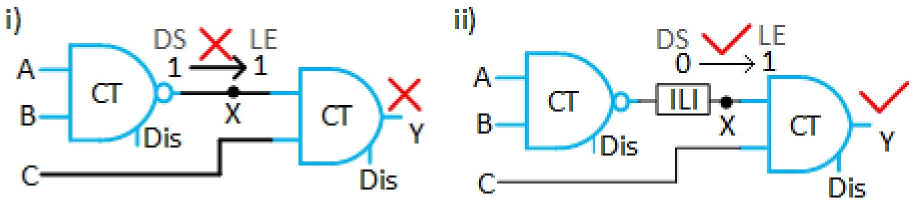


Figure.6.3 Cascading Circuit issues and solutions: i) Logic 1 mismatch node, ii) Employing ILI circuit to fix mismatch node.

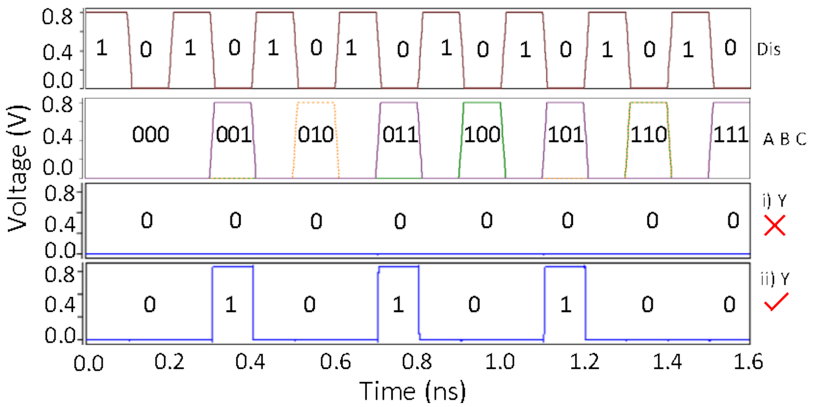


Figure.6.4 Simulation response of the circuits Figure.6.3(i) and Figure.6.3(ii)

correspond to output (*Y*) of Figure.6.3(i&ii), respectively. It can be observed from panel-3 that the output remains zero always, showing incorrect logic behavior because node *X* never sees the 0→1 transition that is required to detect the logic 1 by the next stage AND gate (which is a positive transition gate). However, in panel-4 correct logic behavior is achieved because of the ILI circuit used.

6.2.2 Leveraging Crosstalk Circuit types to fix Mismatch Nodes

Though IHI and ILI circuits fix the mismatch-node problems, they add additional circuitry as each of these circuits requires 3 additional transistors. We can overcome this circuit overhead by using positive, negative, and dual transition Crosstalk gates in combination. We have seen that the initial-states (inputs and output) of the negative transition gate complement a positive transition gate. That means, when we have a mismatch-node, we can alternately swap one of the gates (driver or load) to a different transition gate type and thus fix the mismatch problem. For example, we can swap a positive transition gate to a negative transition gate. Figure.6.5 shows three common circuit configurations, column A: I, II, and III, for which either driver or load gates can be swapped to negative transition gates (Column C-H). A few combinations turn out to be legal connections (shown with red right

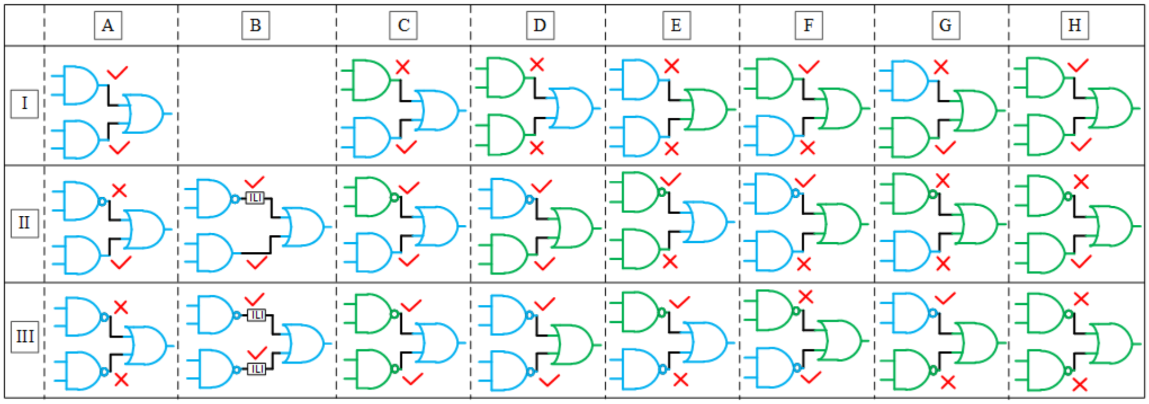


Figure.6.5 Legal and illegal connections for cascading Crosstalk circuits

mark), while others are illegal connections (shown with red cross mark) because of the mismatch nodes. In case-I (first row), both the driver gates are non-inverting logic type (two AND gates). The output of both the AND gates are connected to the OR gate. As all three gates in case-IA are positive transition gates (i.e., low to high transition), and have 0 initial states for all the nodes, such cascading will maintain proper functionality. However, in case-II (second row) and case-III (third row), one or both the driver gates are inverting logic gates (NAND), respectively. These inverting gates create mismatch nodes, as discussed in the previous section. To avoid these mismatch nodes, ILIs can be used, as shown in case-II.B and case-III.B. These ILIs will make sure that the corresponding input aggressor sees the 0→1 transition during every ES state. However, such initializers increase the transistor count.

Alternatively, circuit cases in column C-H show the swapping of driver or load gates to negative transition gates, thus removing the need for ILI circuits as in column B's circuit cases. The column C-H circuit configurations are various combinations of positive and negative transition gates (including legal and illegal connections) to achieve the same logic as in the base cases/configurations, I.A, II.A, and III.A. In Case-I (see first row), I.A and I.H are the legal connections but all other combinations, I.C-I.G, lead to mismatch-nodes, hence illegal. That is, a positive transition load gate (having 0 initial states for inputs) can only have all non-inverting positive transition gates as drivers (Figure.6.5(I.A)) but not one or more negative transition non-inverting gates as in I.C and I.D (green-color gates). Similarly, a negative transition load gate (having 1 initial state for inputs) can only have all negative transition non-inverting gates as inputs (I.H) but not one or more positive transition gates as in I.E, I.F, and I.G (blue-color gates).

The case-II (see the second row) describes the solution to connect both inverting and non-inverting gates to a non-inverting gate. Case-II.B applies the ILI circuit, but II.C and II.D alternately use positive and negative transition gates to fix the mismatch-node problem. All other possible combinations (II.E-II.H) are illegal. That is, as shown in II.C, the inverting input gate can be swapped to a negative transition gate (green colored gate), or, as shown in II.D, the inverting input gate remains as a positive transition gate (blue colored), but the other two gates (input gate and load gate) are switched to negative transition gates (green colored). Any other combinations in II.E-IIH will not work. Finally, if all the inputs are driven by inverting gates, as shown in case-IIIA, there are three solutions. First, as shown in III.B, we can add ILI circuits between each driver and load gate. Second, as shown in III.C, we can switch both the input inverting gates to negative transition gates (green color gates). Third, as shown in III.D, we can instead swap the load gate to a negative transition gate (green colored gate). All other combinations, III.E-III.H, should be avoided. Thus, we can see that six transistors overhead brought in by auxiliary circuits can be reclaimed using positive, negative, and dual transition Crosstalk gates in combination.

The next example shows the usage of a dual-transition gate to reclaim three transistors. Figure.6.6 shows two different cascading styles for implementing a full-adder. In

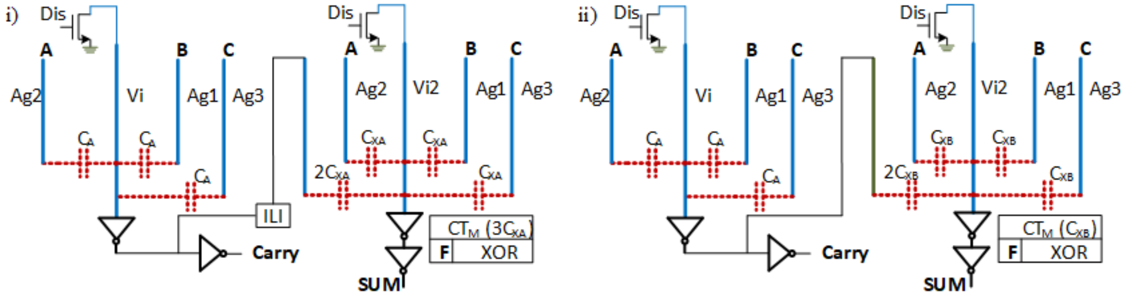


Figure.6.6 Two different cascading styles for implementing Full Adder. i) Implementation of Full Adder using initializers, ii) Implementation of Full Adder using dual transition SUM circuit.

Figure.6.6.(i), ILI is used for connecting inverted Carry output to one of the inputs of the SUM circuit. This circuit needs 13 transistors in total, where three transistors are needed for the ILI circuit. However, the same full-adder can be implemented, as shown in Figure.6.6(ii), using only ten transistors by leveraging a dual transition type SUM circuit.

The above circuit scenarios have a fanout of 2, but practical circuits will also have a fanout greater than 2. Figure.6.7(i.A)&(ii.A) show fanout cases in which we have conflicting initial state requirements from different load gates. When we swap the positive transition gates with negative transition gates in these circuits (to avoid initializers), we might arrive at a situation where a node can have conflicting requirements such as few of the fanout gates might demand a positive transition type driver gate, and remaining gates can demand negative

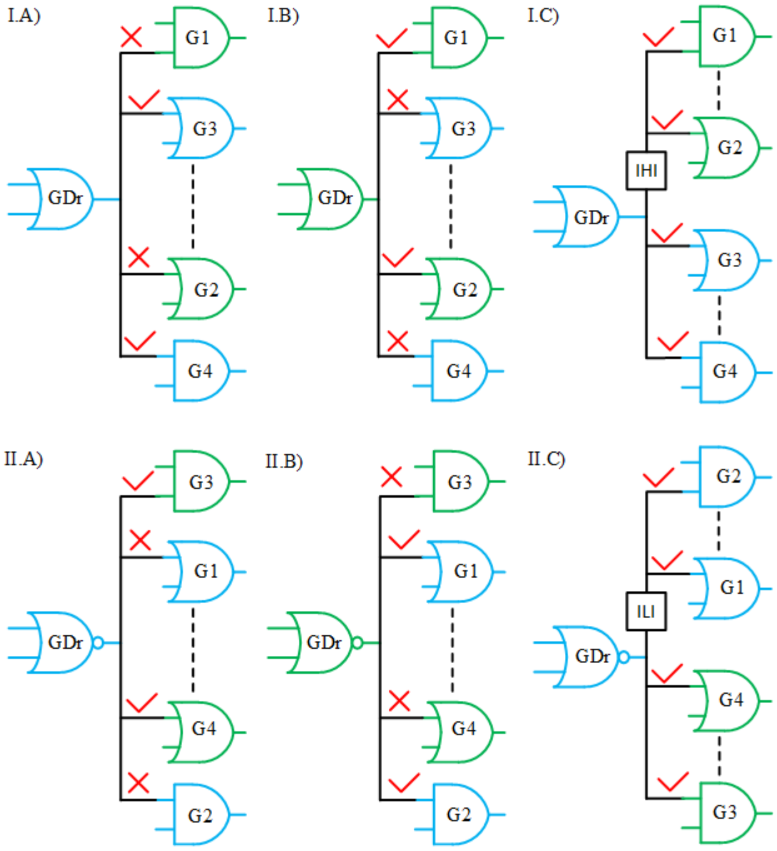


Figure.6.7 Fan-out configuration for Crosstalk circuits

transition type driver gate. Figure.6.7(i.A)&(ii.A) show two such scenarios. Fanout gates G1 and G2 have the mismatch-nodes in two cases. In both cases, if we swap the driver gate (GDr) to a negative transition type, as shown in Figure.6.7(I.B)&(II.B), to fix the mismatch node issue, the inputs of the other two gates, G3 and G4, will turn out to mismatch-nodes. In such a conflicting scenario, we can resort to the initializer circuits, as shown in Figure.6.7(I.C&II.C). In Figure.6.7(I.C), all the negative transition gates' inputs are grouped and driven through a single IHI circuit. Similarly, in Figure.6.7(II.C), all the positive transition gates' inputs are grouped and driven through a single ILI circuit. As drive strength can be a concern for these ILI and IHI circuits in large fanout scenarios, we can employ ILI and IHI circuits' regenerative versions (Figure.6.2(v&vi)).

6.2.3 Crosstalk Circuits with inherent output initializers

The Initializer circuit (ILI or IHI) can be merged with the CMOS inverter of the Crosstalk gate and form an inherent output initializer circuit, as shown in Figure.6.8. Figure.6.8(i) shows the Output Low Initializer (OLI), where the NMOS discharge transistor connected at the output and gated by *Dis* makes sure the output is 0 in DS (*Dis=1*) state irrespective of the *Vi* node initial state. The complementary PMOS transistor (gate by *Dis*) in the circuit avoids the simultaneous turn-on of PMOS and NMOS branches. All down-stream fanout aggressors are

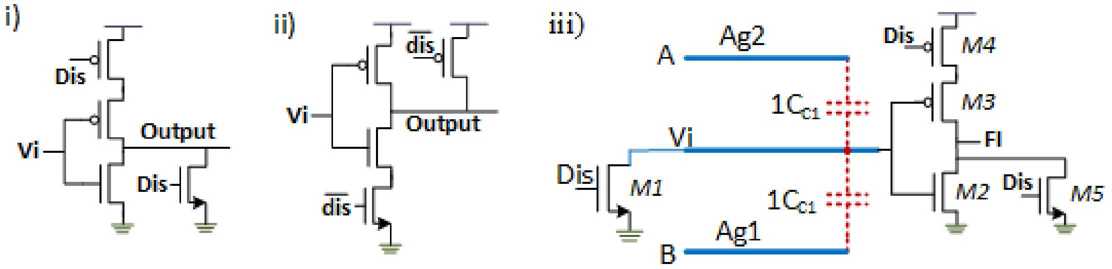


Figure.6.8 i) Output Low Initializer (OLI), ii) Output High Initializer (OHI), iii) Crosstalk Gate with inherent output low initializer.

automatically pulled to the ground using this circuit technique and ensure 0→1 transition in every ES state. Similarly, Figure.6.8(ii) shows the Output High Initializer (OHI), where the precharge (PMOS) transistor connected at the output and gated by the *Dis'* signal will pull the output to 1 irrespective of *Vi* node signal level. All down-stream fanout aggressors are automatically pulled to 1 using this circuit technique, ensuring 1→0 transitions in every ES state. The in-built initializer circuits reduce the transistor count compared to the previous two techniques and also regenerative.

Figure.6.8(iii) shows the Crosstalk circuit using OLI. This circuit inherently fixes all the cascading problems discussed in previous sections. The circuit operates as follows. In every DS state, the *Vi net* is discharged/initialized to ground through M1 transistor, M2 is OFF, M3 is ON, but the *Dis* signal will turn OFF the Pullup branch by turning OFF M4 transistor. Now, the M5 turns ON and shorts the output node *FI* to the ground; thus, the gate's output, in turn, input aggressor/aggressors of next stage gate/gates are initialized to 0.

It should be noticed that all nodes (*Vi*, *FI*, next stage input) that require the initial state in the Crosstalk circuits will be automatically initialized 0 using this circuit-style (Figure.6.8(iii)). Thus, all the circuits can be connected freely without any additional constraints or auxiliary circuits for functional correctness by using such Crosstalk gates. Figure.6.8(iii) implements inverting logic functions. For non-inverting logic functions, we will use plane circuits discussed above because they inherently create an initial state as 0 due to the two inverters connected on the *Vi net*. This circuit also enables the Crosstalk circuits to be compatible with existing Synthesis and Place-and-Route (PnR) flows with only one additional requirement of routing the *Dis* signal, which can be done through a special route step.

CHAPTER 7

EXISTING POLYMORPHIC CIRCUIT APPROACHES

Polymorphic logic circuits are rich in their functional behavior, where a control variable can deterministically morph the circuit's behavior between multiple functions [59]. For example, an AND gate can change as an OR gate and vice-versa. Thanks to their ability to transform intrinsically, polymorphic circuits find their use in a myriad of applications [60-66] such as reconfigurable circuits/systems design [60][61], resource sharing[60][61], multifunctional adaptive systems [62], hardware security [63], fault tolerance [64], and self-test circuits[65-67]. Besides, as scaling down of feature size in Integrated Circuits (ICs) is approaching the physical limits, the miniaturization trend of ICs (Moore's Law) is relaxing. Therefore, developing alternate techniques that try to push the horizons of Moore's Law can be of tremendous potentials. Polymorphic circuits can be one such technique that tries to sustain Moore's Law because they increase the circuit functionality in a given footprint by reusing the circuits to execute different functions. However, the tradeoffs in achieving such polymorphic circuits make them go or no-go for applications.

At the circuit level, the conventional CMOS approach to design such reconfigurable circuits is to have multiple individual functional gates/blocks, which are then selected using multiplexers. However, this approach cannot be genuinely polymorphic because the multiple functions are achieved through circuit redundancy. Also, the transformation mechanism uses transistors as switches (for forming multiplexers). This approach is resource-intensive because of redundancy and multiplexers. Instead of having individual functional units, if a single circuit can exhibit multiple functions, the ensemble will collapse down to a single circuit's footprint; to achieve this, the earlier attempts were based on the design of functionally superimposed circuits, as presented in [68][69] and recently, in [70]. This approach is based on deriving clever circuit topologies within the switch and CMOS based circuit framework. They reduce the transistor count compared to the above multiplexing approach; however, the circuits constitute long series-parallel branches, which reduces the Performance and Area benefits they could offer, and morphing is still achieved using the transistor as a switch.

To be considered as a genuinely polymorphic circuit, along with the innate multifunctional nature, the control between different functions should be enabled by inherent device characteristics and/or external environmental influences [59]. The Polymorphic circuits evolved using genetic algorithms [71] were extensively researched. These evolved circuits can morph their functional behavior based on different environmental control variables such as temperature, supply-voltage, control-signal, light, radiation, etc. They find interesting applications in sensor circuits that morph and adapt their behavior in different environments [72], especially in extreme conditions of temperature, radiation, microwaves, etc. (for example in space electronics) [73][74]. The disadvantages they face are technology dependency, scalability, and inefficiency in speed and power [75], because of which the evolved circuits do not find applications in mainstream digital circuits/systems.

Another approach pursued is chaos computing [76], in which non-linear dynamics in transistors and circuits are captured to implement multifunctional circuits. But these circuits are custom nonlinear/mixed-signal circuit designs for digital circuits. More recently, polymorphic circuits are also designed using emerging tunable polarity transistors [77-81], configured either as p-type or n-type based on a control signal. These morphable transistors foster various fine-grained multifunctional circuit schemes [82]. However, these novel devices require complex device engineering compared to mainstream CMOS devices. The circuit schemes also necessitate additional circuitry to switch the power rails when the transistors change as p-type/n-type [78].

The other alternate approaches using emerging spintronic devices were also proposed [63], but they rely on complex information encoding schemes through spin-polarized currents and bipolar voltages, etc. Consequently, they are a significant departure from the existing computational device and circuit paradigms. The novel exotic-device-based approaches try to create polymorphism based on the transformation of device characteristics, which are experimentally creative and futuristic. But they lack immediate mainstream fabrication developments and adaptability.

CHAPTER 8

CROSSTALK POLYMORPHIC CIRCUITS

Crosstalk Computing enables the implementation of a wide range of compact and efficient polymorphic circuits [57]. For polymorphism, an additional control signal is used to bias the Crosstalk circuit to alter its functional behavior. This chapter discusses the Crosstalk Polymorphic Computing concept and presents a comprehensive list of Crosstalk polymorphic circuit designs and their simulation responses. The polymorphic gates shown are AND-OR, AND-AO21(AND-OR-21, i.e. (AB)+C), AND-OA21 (OR-AND-21, i.e. (A+B)C), AND-CARRY(AB+BC+CA), OR-AO21, OR-OA21, OR-CARRY, AO21-OA21, CARRY-AO21, CARRY-OA21, and Inverter-Buffer (Inv-Buf) [83]. Next, a few polymorphic cascading circuit examples are shown at the gate-level and module level; they depict how reconfigurability can be enriched by cascading Crosstalk polymorphic gates.

8.1 Crosstalk Polymorphic Logic Gates

Unlike the CMOS circuit style, where we have fixed patterns of series and parallel connection of switches (transistors) for each logic type, Crosstalk logic circuits are of uniform pattern with the only difference in their coupling capacitances. That means, if the coupling capacitances from inputs to the *Vi* net can be altered at runtime, the gate's logic behavior can also be altered. This ability to modify the runtime logic behavior could pave the way for designing a new kind of polymorphic/reconfigurable logic circuits based on Crosstalk Computing. Instead of trying to achieve the run-time alteration of coupling capacitances by controlling material properties or by constructing novel devices for this purpose, an alternate path can be chosen where the *Vi* net is coupled with an additional control aggressor (*Ct*). The transition of the signal on *Ct* would augment an extra charge/voltage on to the *Vi* net, which is equivalent to run time alteration of the capacitance coupled to the *Vi* net. This extra voltage induced on the *Vi* net would actually disturb the intended logic behavior of the gate. However, if this extra voltage induced is engineered properly, the gate's logic behavior can be astutely morphed such that a new functional pattern can emerge and give rise to polymorphic gates. By clever designing, the polymorphism is possible between various logic functions. The polymorphic gates are categorized as [83]: homogeneous to homogeneous logic— *AND2-OR2*, *AND3-OR3*, *AND3-CARRY*, *OR3-CARRY*; heterogeneous to heterogeneous logic—*AO21-OA21*; and homogeneous to heterogeneous logic— *AO21-AND3*, *AO21-OR3*, *AO21-CARRY*, *OA21-AND3*, *OA21-OR3*, *OA21-CARRY*.

For a generic Crosstalk polymorphic gate as shown in Figure.8.1, 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 equation (II) now turns to,

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

Where, $m = w_1L_1 + w_2L_2 \ldots + w_nL_n$, and

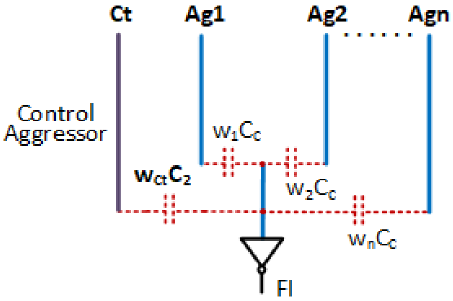


Figure.8.1 Circuit Schematic of a Generic Crosstalk Polymorphic Gate

$$L_{C\,t} = \begin{cases} 0\, if\ control\ signal\ Ct\ \ is\ low\ voltage \\ 1\, if\ control\ signal\ Ct\ \ is\ high\ voltage \end{cases}$$

The CT-margin function is an abstraction for logic behavior in Crosstalk Computing. Therefore, the transformation of the Crosstalk logic gate's behavior from one function to the other function would also mean an effective change in their margin-functions. Let us assume that $CT_M(k. C_C)$ is the CT-margin function for a given polymorphic gate when the control signal is low (i.e., $L_{C\,t} = 0$). The Crosstalk 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,

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

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

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

Therefore, the CT-margin function transforms as follows,

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

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 $L_{C\,t} = 1$, an extra voltage will be induced through capacitance *w_{Ct}.C_C*, leaving only *(k-w_{Ct})C_C* capacitance margin; i.e., the inverter can now flip its state just with the voltage induced due to capacitance greater than or equal to (k- *w_{Ct}*)C_C. Therefore, the margin function and its corresponding logic behavior will be transformed to *CT_M((k-w_{Ct})C_C)*.

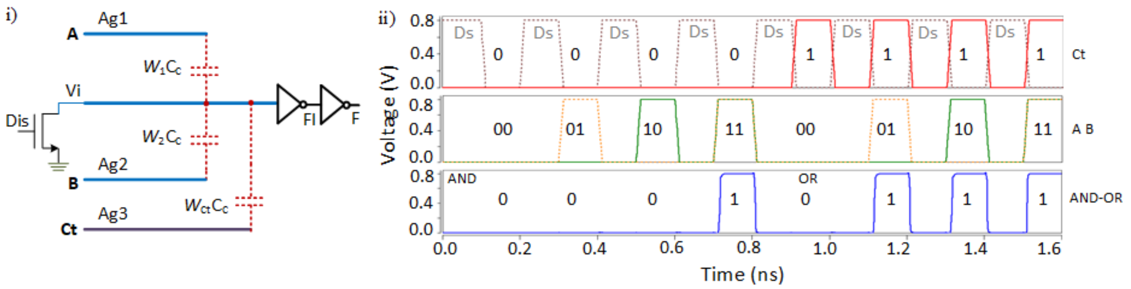


Figure 8.2 2-input Crosstalk-Polymorphic Logic Gate: i) AND2-OR2 Schematic, ii) AND2-OR2 Simulation response

TABLE 8.1
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	

Various 2-input and 3-input Crosstalk polymorphic logic circuits are implemented. Figure.8.2(i) shows the Crosstalk polymorphic AND2-OR2 Circuit. Table.8.1 presents the circuit design parameters for the AND2-OR2 gate, C_C, input and control aggressors' weights, and PMOS and NMOS widths ratio. The table also shows the effective transformation of CT-margin function with respect to control logic $L_{C\,t}$ and its corresponding function. It can be observed from the simulation response Figure.8.2(ii) that when $L_{C\,t} = 0$ the circuit responds as OR gate, whose behavior is abstracted to CT-margin function $CT_M(2C_C)$ in the table. But when $L_{C\,t} = 1$, the circuit responds as AND gate, whose behavior is abstracted to $CT_M(C_C)$ in the table.

Ten different types of 3-input polymorphic circuits are implemented next, which are listed in Table.8.2. All these circuits are represented by a single schematic in Figure.8.3 to limit the space, as all these gates have uniform circuit topology with only difference in their design parameters. Table.8.2 lists all the circuit-design parameters for different gates. The simulation

TABLE 8.2
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	

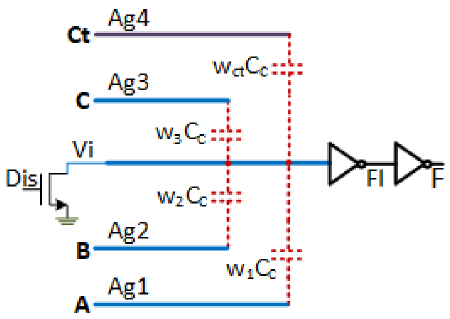


Figure.8.3 Generic 3-input Crosstalk-Polymorphic Logic Gate Schematic

responses of all the circuits are presented in Figure.8.4, 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, the inputs *A*, *B*, and *C* have the same coupling weights of C_C (i.e., w_1=w_2= w_3=1*)*, while *Ct*

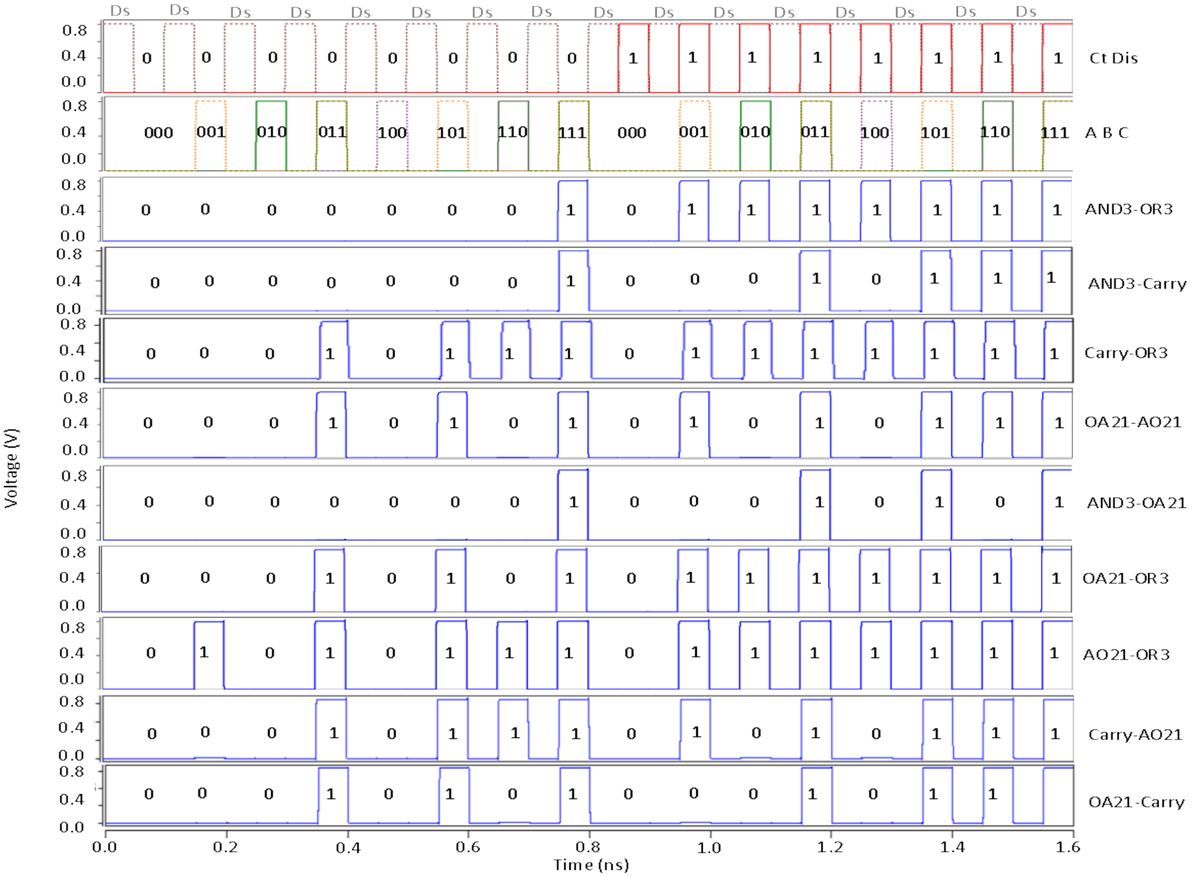


Figure.8.4 Simulation responses of 3-input CT-Polymorphic logic gates

aggressor receives *2C_C* capacitance (i.e., w_{Ct}=2*)*. When $L_{C\,t} = 0$, the margin function for the *AND3-OR3* gate is *CT_M(3C_C)*, which makes it behave as *AND3,* as shown in Figure.8.4 panel-3. Whereas, when $L_{C\,t} = 1,$ the *Ct* aggressor augments an extra charge through coupling capacitance *2C_C* and effectively manipulates the margin function to *CT_M(C_C)*. Following the function *CT_M(C_C)*, the transition of either *A*, *B,* or *C* is now sufficient to flip the inverter. Thus, the gate biases and operate as an OR3 gate, as shown in Figure.8.4 panel-3. It can be observed that the circuit responds as *AND3* when $L_{C\,t} = 0,$ for the first eight input combinations (*000* to *111*), whereas it responds as *OR3* when $L_{C\,t} = 1,$ during the next eight combinations (*000* to *111*). For the *AND3* gate, if the control aggressor is given just *C_C* coupling strength instead of $2C_C$ in the previous case, $CT_M(3C_C)$ manipulates to $CT_M(2C_C)$ instead of $CT_M(C_C)$, which becomes a polymorphic *AND3-CARRY* gate as given in the table. The corresponding simulation response is in Figure.8.4 panel-4. Similarly, for the polymorphic *CARRY-OR3* circuit, the margin function is $CT_M(2C_C)$ when $L_{C\,t} = 0$, during which the circuit behaves as *CARRY* logic (Figure.8.4 panel-5). The control aggressor is given C_C strength, which alters the function to $CT_M(C_C)$ when $L_{C\,t} = 1$ and morphs its behavior to the *OR3* gate (Figure.8.4 panel-5).

The above three gates are homogeneous logic types, where input aggressors receive equal coupling strength (see Table.8.2). The next gate is *AO21-OA21,* which is a heterogeneous to heterogeneous logic. The coupling weights of aggressors are $w_1=w_2=1,\ w_3=2,$ *and* $w_{ct}=1$ (Table.8.2). The margin function, $CT_M(3C_C)$, alters to $CT_M(2C_C)$ when $L_{C\,t} = 1$ and gives *CT*-polymorphic *AO21-OA21* gate (circuit response is in panel-6).

The next six gates are homogeneous to heterogeneous logic types. For the *AND3-AO21* gate, the aggressor weights are $w_1=w_2=1,\ w_3=2,$ *and* $w_{ct}=2$ (note that input weights are heterogeneous). The margin function for *AND3*, in this case, 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 *Ct* is given C_C strength instead of $2C_C$, the margin function manipulates from $CT_M(4C_C)$ to $CT_M(3C_C)$, giving rise to Crosstalk polymorphic *AND3-OA21* gate as shown in Figure.8.4 pane-8. Similarly, the next circuits presented in the table are CT-polymorphic OA21-OR3, AO21-OR3, OA21-CARRY, and AO21-CARRY gates, and their simulation responses are given in panel-9, panel-10, panel-11, panel-12, and panel-13, respectively. Thus, it can be observed from the table that by engineering the design variables,

C_C, coupling weights, transistor widths, and control aggressor's influence, the CT-margin function can be astutely controlled to form various Crosstalk polymorphic logic gates.

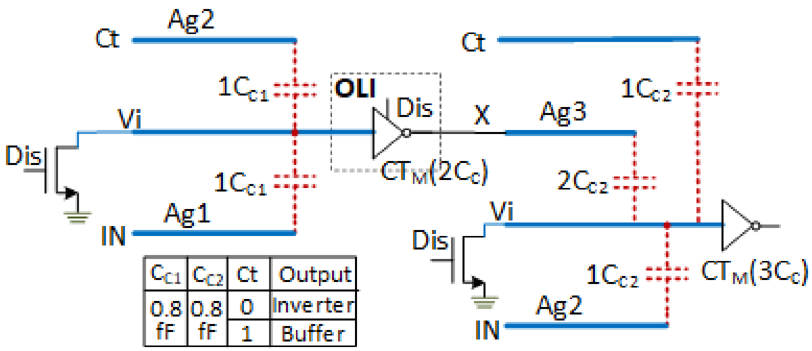
8.2 Cascaded Polymorphic Circuits

So far, we have seen the polymorphic gates only with a single-stage Crosstalk circuit. We can construct many interesting Polymorphic circuits by cascading the Crosstalk gates while employing the cascading techniques discussed in Chapter 6. The next two subsections discuss the cascaded polymorphic gates at the fain-grained level [83], i.e., cascading 2 or 3 gates, and at the modular level [83], i.e., cascading more than 30 gates.

8.2.1 Fine-grained Cascaded Polymorphic Circuits

The circuits discussed so far were reconfigurable between Boolean functions of the same polarity. That is, an inverting logic function transforms into another inverting logic function (ex: NAND to NOR, etc..), and a non-inverting function converts to another non-inverting function (ex: CARRY to AND3, etc..). Polymorphism between opposite polarity functions, e.g., NAND-AND, can enable reconfigurability between all possible Boolean functions. This section shows reconfigurability between opposite polarity functions, i.e., inverting and non-inverting function using fine-grained/gate-level cascading of Crosstalk polymorphic circuits.

The fundamental primitive that could enable the reconfiguration of inverting to non-inverting logic function is polymorphic Buffer-Inverter (Inv-Buf). Figure.8.5 depicts the Inv-Buf circuit. The coupling weights and CT-margin functions are annotated in the schematic, and the table adjacent lists the coupling capacitance values. The simulation response of the Inv-Buf circuit is shown in Figure.8.6. It can be observed that the circuit behaves as an Inverter when *Ct=0*, whereas it acts as a Buffer when *Ct=1*. This circuit enables the transformation



C$_{C1}$	C$_{C2}$	Ct	Output
0.8 fF	0.8 fF	0	Inverter
		1	Buffer

Figure.8.5 CT-Polymorphic Inverter-Buffer Circuit schematic

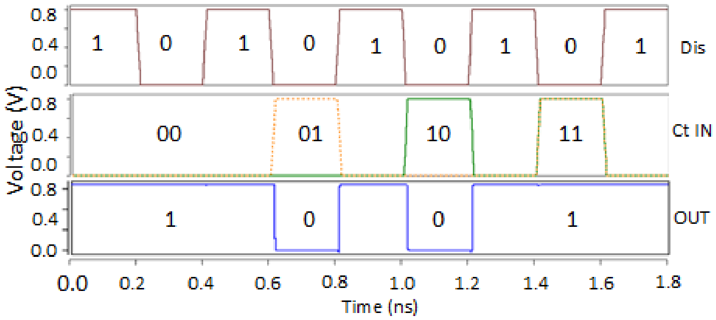


Figure.8.6 Simulation response CT-Polymorphic Inverter-Buffer Circuit

from any inverting-function to a non-inverting function and vice-versa. Figure.8.7 shows a 3-stage polymorphic gate formed by cascading three Crosstalk polymorphic gates (two Inv-Buf gates and one AND-OR gate). It cascaded circuit is a generalized polymorphic circuit implementation that can morph between all two-variable/input linearly separable Boolean logic functions. Table.8.3 lists all the functions that can be obtained. *Ct1*, *Ct2*, *Ct3,* and *Ct4*

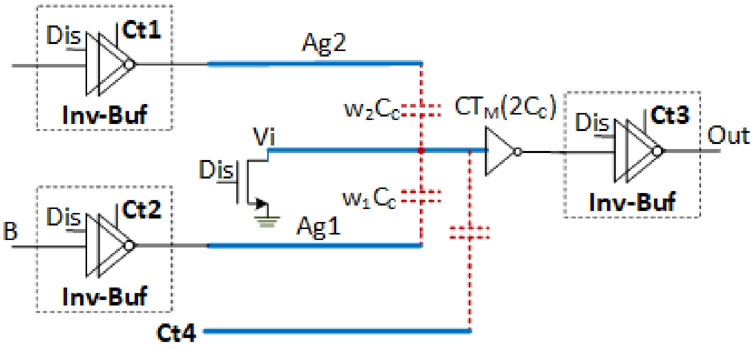


Figure.8.7 Three CT-Polymorphic Gates cascaded to generate 16 functions

TABLE 8.3
Sixteen Reconfigurable functions for the Polymorphic Circuit in Figure 5.6

Ct1	Ct2	Ct3	Ct4	Out
0	0	0	0	(A'.B')
0	0	0	1	(A'+B')
0	0	1	0	(A.B')'
0	0	1	1	(A'+B')'
0	1	0	0	A'.B
0	1	0	1	A'+B
0	1	1	0	(A'.B)'
0	1	1	1	(A'+B)'
1	0	0	0	A.B'
1	0	0	1	A+B'
1	0	1	0	(A.B')'
1	0	1	1	(A+B')'
1	1	0	0	A.B
1	1	0	1	A+B
1	1	1	0	(A.B)'
1	1	1	1	(A+B)'

are the control signals to configure the circuit. This circuit is implemented to show the versatility of Crosstalk polymorphic gates. However, all these functions might not be required in a single setting in real circuits, but a subset of these functions can be created as needed by appropriate changes. The Inv-Buf technique is applied here only to the AND-OR gate. It can be equally applied to all other polymorphic gates discussed in the previous sections. Though the discussion in this section is limited only to the transformation between inverting and non-inverting functions, many other transformations can also be created. For example, the polymorphic circuits like XOR/AND and XOR/OR are also designed by cascading only two Crosstalk gates.

8.2.2 Modul-level Cascaded CT-Polymorphic Circuit

This section demonstrates cascading polymorphic gates to implement a block-level polymorphic

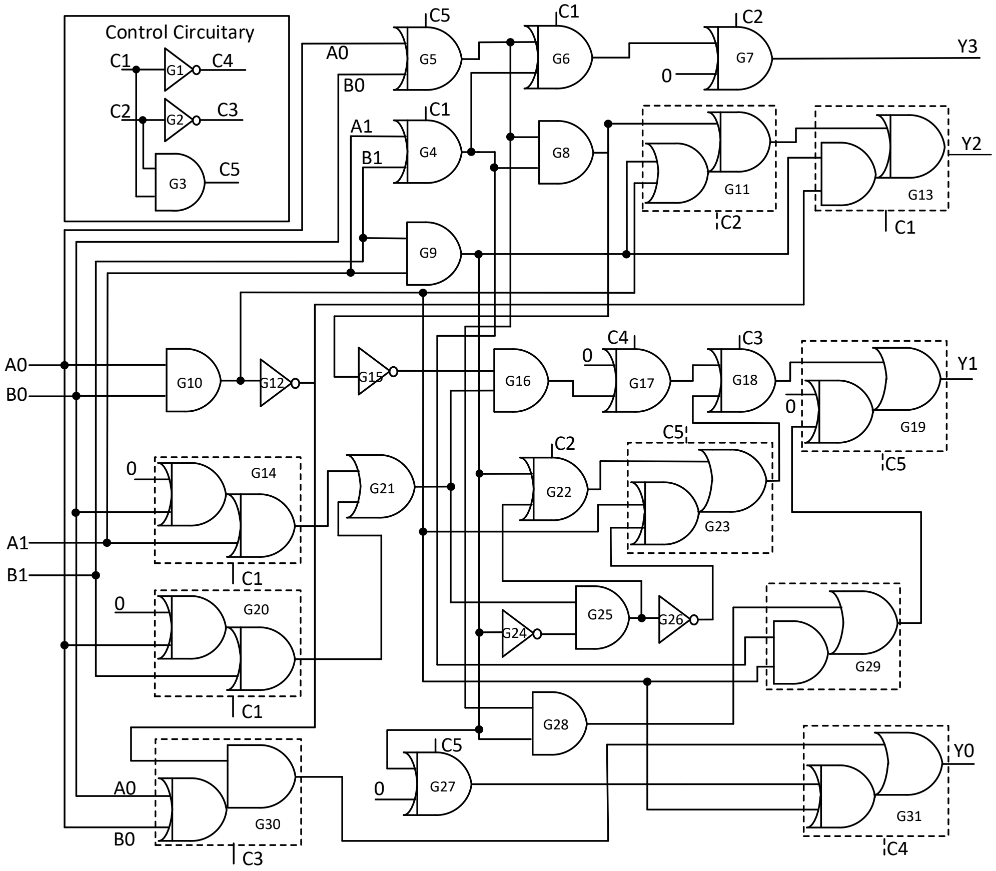


Figure.8.8 Crosstalk Polymorphic Multiplier-Adder-Sorter circuit

circuit. Figure.8.8 is a 2-bit Multiplier-Sorter-Adder circuit. The circuit uses 31 gates in total, out of which 25 are crosstalk gates, and 6 are inverters. 16 out of 25 crosstalk gates are polymorphic gates, which are efficiently employed to switch the circuit between the multiplier, sorter, and adder operations. Two control signals, C1 and C2, are given to a control circuitry shown in the inset figure, which generates C3-C5 signals. C1-C5 signals are employed in the circuit to switch the circuit between three functions. Figure.8.9 shows the simulation response of the circuit; Multiplier (M), Sorter (S), and Adder (A) operation modes are annotated on top. The first panel in the figure shows Dis signal; *Dis*=1 is the discharge state (DS), and *Dis*=0 is the logic Evaluation (ES) state. The second panel shows the control signals *C1* and *C2*, whose values as 01, 11, and 10 corresponds to the multiplier, sorter, and adder operations, respectively. The third and fourth panels show the 2-bit inputs

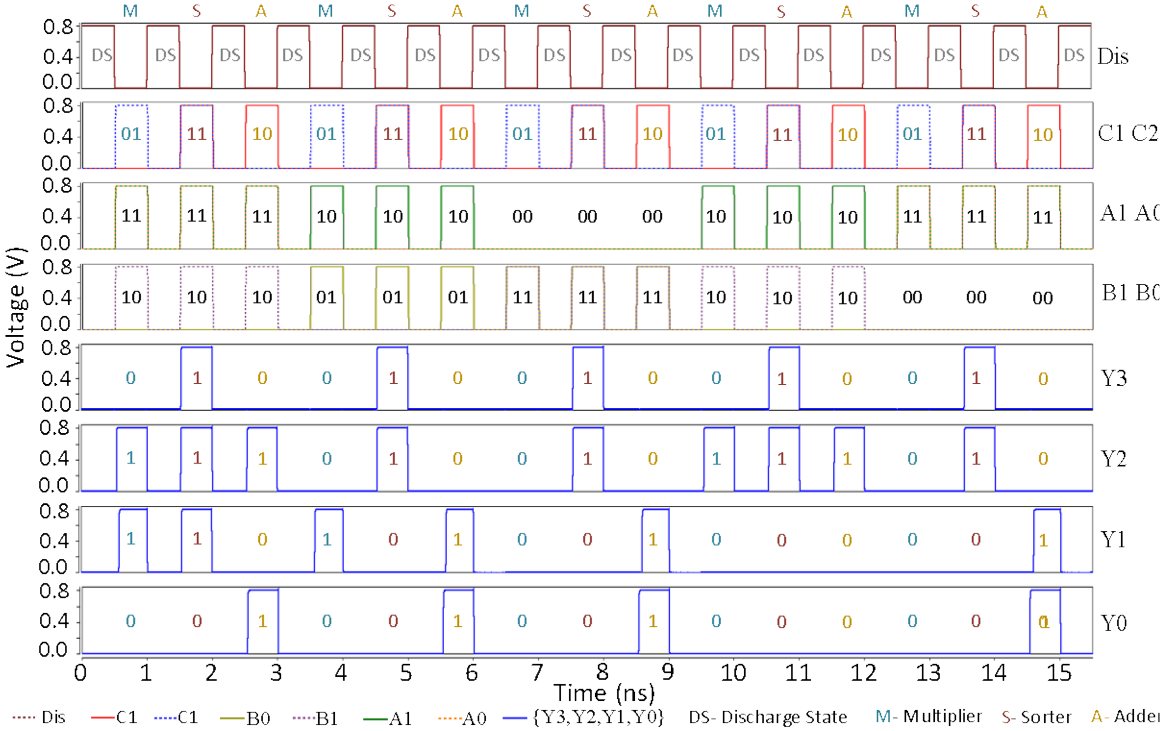


Figure.8.9 Crosstalk Polymorphic Multiplier-Adder-Sorter circuit simulation response

A[1:0] and B[1:0]. The subsequent four panels show the 4-bit response of the circuit, Y[3:0]. Control signals are switched alternately between multiplier, sorter, and adder modes to demonstrate the circuit's transformation effectively. In each set of these modes, common input values are fed through A1A0 and B1B0. For example, for the first input combinations, 11 and 10, the multiplier operation gives 0110 as output while the succeeding sorter and adder operations give 1110 and 0101 outputs, respectively. Similarly, for the second inputs, 10 and 01, M, S, and A operations result in 0010, 1100, and 0011 outputs, respectively. Similarly, few other combinations are shown in the next stages. The circuit consumes only 155 transistors in total.

CHAPTER 9

COMPARISON AND BENCHMARKING OF CROSSTALK CIRCUITS

9.1 Comparison

This section compares the Crosstalk polymorphic logic circuits with respect to existing polymorphic approaches available in the literature and discusses its advantages and disadvantages. The Table.9.1 compares different technology, device, and circuit metrics such as working mechanism, control parameter, process node dependency, scalability, performance

TABLE 9.1
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 technology 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

trade-offs, and transistor count for four types of reconfigurable circuit implementations. The reconfigurability in Crosstalk polymorphic circuits is achieved by using the same Crosstalk aggressor-victim technique that performs the logic computation, which enables deliberate and very fast reconfiguration of the gates.

Despite radical logic and reconfigurability aspects, Crosstalk computing's working mechanism is based on well-known capacitive electrostatics, making it easily realizable through existing process setups and fabrication techniques.

For control-voltage, only binary voltage-levels are used in this paper to create polymorphic circuits; however, it can be of any voltage levels in practice. The multiple voltage levels for control-voltage will enrich the multi-functional embodiment of a circuit, i.e., many other interesting polymorphic functions can be derived in a single gate. For example, we can construct the AND-CARRY-OR circuit by having three voltage levels in the Control signal.

The Crosstalk circuits are technology node independent, i.e., the circuit style is generic to implement at any technology node. Interestingly, the Crosstalk circuits address the impediments advanced technology nodes face in terms of interconnect-crosstalk by astutely leveraging it for the computational purpose; thus, it is friendly to advanced technology nodes.

Compared to other approaches, the Crosstalk polymorphic approach is a very compact implementation. The transistors counts are compared in Table.9.2. The complex gates listed for other approaches in the table are constructed by cascading polymorphic NAND-NOR and AND-OR gates presented in [59] and [78]. The traditional approach (‘CMOS’ column in the table) is multiplexer based, where independent and stand-alone circuits are designed and selected through a multiplexer. Though this approach is mainstream and can be implemented

TABLE 9.2
Transistor Count Comparison

Gates	CMOS	Evolved Circuits [3]			Ambipolar	Crosstalk
NAND2	4	-	-	-	4	3
NOR2	4	-	-	-	4	3
AOI21	6	-	-	-	6	3
OAI21	6	-	-	-	6	3
NAND3	6	-	-	-	6	3
Carry	16	-	-	-	16	3
Polymorphic Gates						
NAND-NOR	14	11[]	8	6 [ ]	4	3
AOI-OAI	18	24	18	14	6	3
AND2-OR2	18	10[ ]	6 [ ]	8	6	5
AND3-OR3	22	20	12	16	6	5
AO21-OA21	22	22	14	18	8	5
AND3-AO21	22	16	12	14	12	5
AND3-OA21	22	16	12	14	12	5
OR3-AO21	22	16	12	14	12	5
OR3-OA21	22	16	12	14	12	5
Carry-OR3	30	32	24	28	24	5
Carry-AND3	30	32	24	28	24	5
Carry-AO21	32	32	24	28	24	5
OA21-Carry	32	32	24	38	24	5
Mul-Sort []	146	168	132	150	122	88
Mul-Sort-Add	408	288	216	252	216	155

in any technology node (we have designed in 16nm), it consumes large resources as listed in the table.

Evolved circuits are unconventional circuit structures evolved/synthesized using genetic algorithms [71][60]. These circuits are strongly technology-dependent (implemented at 0.35um node in [71]) and work only with specific models, conditions, and technology under which they are evolved; therefore, they are not adaptable to advanced technology nodes. Furthermore, they are inefficient in design [71]; they suffer from unreliable responses (weak output logic level), lower input impedance, slow performance, and high-power consumption, etc. Also, the scalability is strictly limited by evolution techniques [71], where the evolution of larger polymorphic circuits are computation-intensive and hard to converge.

In contrast, Crosstalk polymorphic circuits are highly scalable to larger polymorphic systems because of their generic, uniform, and modular circuit topologies that can be extended and permutated to implement many complex polymorphic functions. 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.

However, the scalability limitations to overcome in Crosstalk Computing are i) ability to achieve the efficient Crosstalk coupling networks, ii) Noise margins of the CMOS inverter that limits the fan-in of the circuits, in turn, the ability to construct many single-stage/gate complex Crosstalk polymorphic logic circuits (cascaded polymorphic circuits are the solution), and iii) Crosstalk Computing friendly polymorphic logic synthesis algorithms/tools need to be developed for EDA (Electronic Design Automation) flows.

Lastly, the evolved circuits possess unique merit to construct polymorphic circuits with various control parameters such as temperature, supply-voltage, light, radiation, etc.[59] (which are not done in the Crosstalk circuit style). These features make them ideal candidates for sensor-based and adaptable circuit applications. It is to be noted that the Crosstalk circuits presented in this thesis are only controlled using a control-voltage. Theoretically, evolution techniques can also be applied to Crosstalk Circuits to explore reconfigurability potentials based on all possible control parameters.

The next approach for comparison is emerging reconfigurable-transistor based circuits. Ambipolar Si nanowire FET (SiNWFET) circuits by De Marchi *et.al* [78] are considered. A nanowire transistor can be configured to either n-type or p-type with a control voltage in this approach. This approach's limitations are [77][78], density benefit is limited, additional circuitry required to swap power rails for pull-up and pull-down networks, non-robust device response, and requirement of new fabrication steps in the existing process flows.

Compared to other exotic device-based approaches [63], Crosstalk Computing can be achieved through existing fabrication techniques. Thus it augments the conventional CMOS-based device, circuit, and manufacturing paradigms. Finally, the Crosstalk polymorphic approach consumes fewer transistors than any other transistor-based polymorphic circuit approach in the literature. By averaging the transistor count of all the circuits in Table.V, the Crosstalk Circuits consume 64%, 58%, and 40% fewer transistors than CMOS, evolved circuits, and Ambipolar Circuit techniques, respectively. Therefore, the density, power, and performance benefits are better in Crosstalk Computing than other approaches.

9.2 Benchmarking

The density, switching energy, and performance for all the crosstalk gates presented above are characterized and benchmarked with their counterpart CMOS implementations. Table.9.3 presents these results. The CMOS implementation is multiplexer based, where independent, stand-alone circuits are designed and selected through a multiplexer. Both the circuits are implemented and benchmarked using 16nm PTM tri-gate transistor models. The benefits are huge in all aspects of Crosstalk logic-based implementations. As shown in Table.9.3, the Crosstalk polymorphic logic gates have over 6x density, ~1.5x performance, and ~2x power benefits. The improvement in power for Crosstalk gates is because fewer active devices lead to lower overall load/switching power and less cell internal power due to fewer device dissipations and parasitics' power. The performance improvement in Crosstalk circuits is due to the absence of any series-connected transistors in Pull-up and Pull-down branches (inverter), which leads to shorter RC paths from

TABLE 9.3
Benchmarking of Crosstalk Logic Gates with Respect to CMOS

	Transistor Count		**Switching Energy (aJ)**			**Performance (ps)**		
GATES	**CMOS**	**Crosstalk**	**CMOS**	**Crosstalk**	**%Reduction**	**CMOS**	**Crosstalk**	**% Reduction**
NAND2	4	3	232.1	122.3	47.31	4.12	4.06	1.4325
NOR2	4	3	202.7	260.5	-28.5	5.61	5.86	-4.492
AOI21	6	3	154.4	207	-34.07	5.73	5.51	3.9373
OAI21	6	3	229.3	135.2	41.03	4.36	5.17	-18.52
NAND3	6	3	347.7	112.5	67.65	4.98	4.18	16.11
Carry	16	5	1198.8	326.592	72.757	14.5822	8.6923	40.39102
NAND2-NOR2	14	3	796.14	139.03	82.537	13.46	4.32	67.887
NAND3-NOR3	18	3	1472.6	172.02	88.319	13.21	5.12	61.224
AOI21-OAI21	18	3	698.42	190.52	72.721	9.52	5.39	43.379
NAND3-AOI21	18	3	1091.3	641.38	41.228	14.08	14.14	-0.4211
NAND3-OAI21	18	3	874.99	959.44	-9.651	11.69	19.4	-65.922
NOR3-AOI21	18	3	1030.4	661.67	35.785	17.78	12.65	28.864
NOR3-OAI21	18	3	938.88	546.89	41.75	18.14	11.47	36.807
CARRY-OR3	30	5	4258.6	420.15	90.134	15.02	8.3	44.74035
Carry-AND3	30	5	3059.9	289.6908	90.533	16.7759	7.3955	55.91593
Carry-AO21	30	5	2332.9	481.3124	79.368	28.9174	10.56	63.48219
OA21-Carry	30	5	2004.2	366.9129	81.693	15.67	9.9741	36.34907
MUL-SORT-ADD	408	155	16.2 fJ	6.104 fJ	62.41	61.5	54.4	11.56

VDD/GND to the gate’s output, and also lower gate internal parasitics.

A comparison of CMOS vs. the Crosstalk circuit can illustrate the source of these benefits. For example, the AND3-CARRY polymorphic circuit, with its Boolean expression, ABCS’+ S(AB+BC+CA), requires just five transistors compared to thirty transistors in CMOS based implementation. For the polymorphic Multiplier-Sorter-Adder unit, the benefits were 3.4x, 62% in terms of density and power with comparable performance with respect to CMOS at 16nm. It is to be noted that the interconnection requirements would also be considerably less because of the reduced circuit density.

For any new emerging technology to compete/co-exist with CMOS, scalability study is one of the critical requirements. As a part of the scalability study, Crosstalk logic gates are designed using 180nm, 65nm TSMC PDK, 32nm PTM model, and 7nm ASAP PDK [84][85]. We have designed primitive gates for both Crosstalk and CMOS in all four nodes and analyzed power and performance. Through simulations at worst-case process corners, we demonstrated in [84][85] that even at sub 10nm technologies, Crosstalk logic gates function correctly, and density, power, performance benefits remain intact.

CHAPTER 10

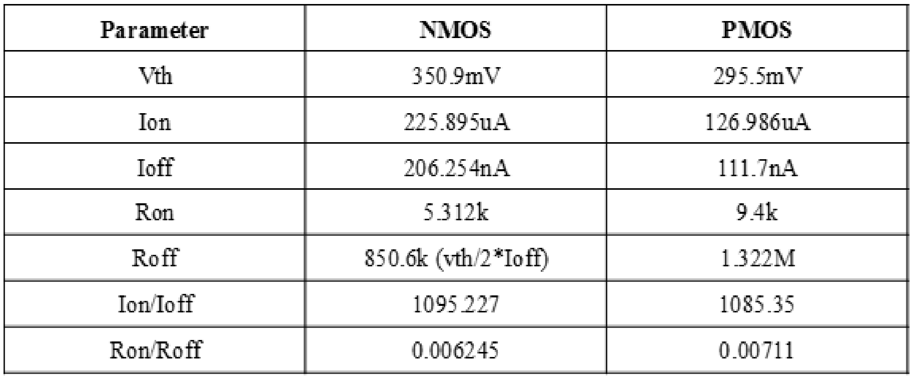
PRACTICAL REALIZATION OF CROSSTALK GATES

10.1 Prototype Circuit Design Flow

This chapter discusses the practical realization of the crosstalk circuits using TSMC 65nm PDK. The die area of the Crosstalk prototype chip is 1mmx1mm. The typical custom circuit and custom chip design flow, as summarized in Figure.10.1, is followed since it is the first prototype demonstrating Crosstalk Computing. The Important DC characteristics of PMOS and NMOS transistors from the PDK are presented in Table 10.1. At first, the inverter DC transfer characteristics are studied to learn about the inverter threshold-voltage (i.e., trip-point) and metastable region to avoid the operation in. Kirchhoff's Voltage Law (KVL) and voltage division principles are applied to the coupling network to formulate the *Vi* net voltage expression. The coupling capacitances required for different logic circuits are computed using the *Vi* net voltage expression (*Eq 4.1*).

The circuit schematics are initially designed in Cadence Virtuoso Schematic Editor [86] with the couplings derived from theory, subsequently, through simulations (Synopsys HSPICE) [87], the circuits are iteratively fine-tuned/optimized for functionality, power,

Table 10.1
Transistor Parameters



Parameter	NMOS	PMOS
Vth	350.9mV	295.5mV
Ion	225.895uA	126.986uA
Ioff	206.254nA	111.7nA
Ron	5.312k	9.4k
Roff	850.6k (vth/2*Ioff)	1.322M
Ion/Ioff	1095.227	1085.35
Ron/Roff	0.006245	0.00711

performance, and noise margins. MOS (NMOS) and MIM Capacitors available in the PDK are used for input aggressors to *Vi* net coupling purposes. Custom layouts are then designed for corresponding circuit schematics and component parameters (in Cadence Virtuoso Layout editor [88]). Then Physical Verification steps, DRC (Design Rule Check) and LVS (Layout-versus-Schematic) verification are performed on the layouts using Mentor's Calibre DRC and Calibre LVS [89]. Next, the layout parasitics (Resistance R, ground capacitance C_G and coupling capacitance C_C) are extracted using Mentor Graphics Calibre PEX [90]. The

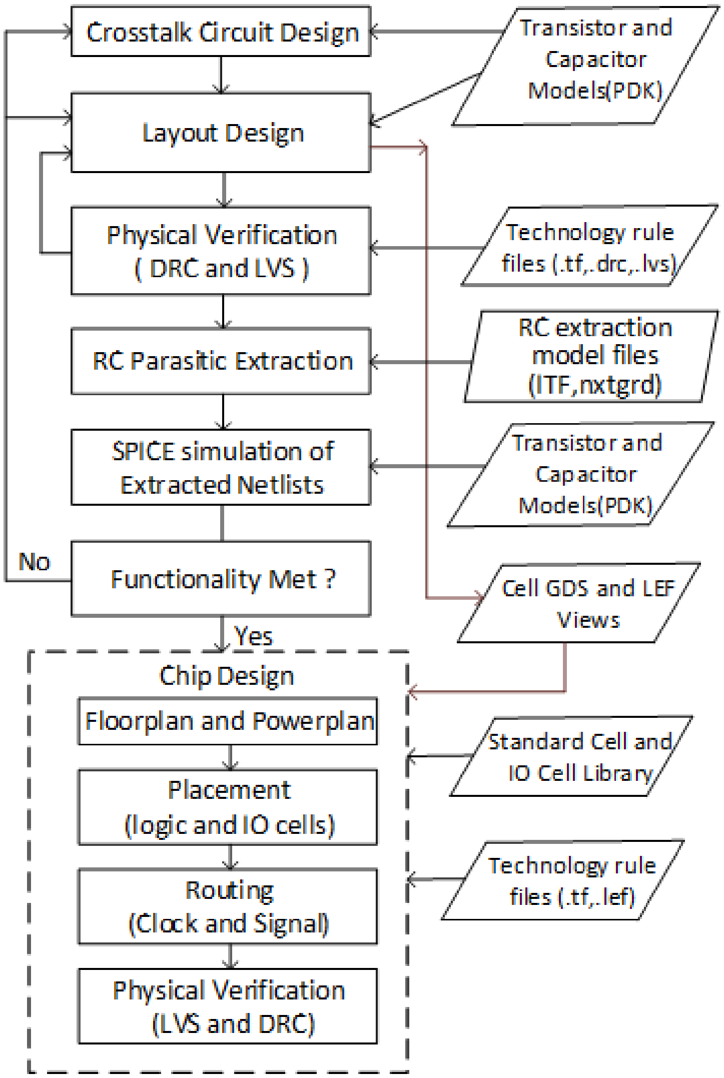

Crosstalk Circuit Design
Transistor and Capacitor Models(PDK)
Layout Design
Physical Verification (DRC and LVS)
Technology rule files (.tf,.drc,.lvs)
RC Parasitic Extraction
RC extraction model files (ITF,nxtgrd)
SPICE simulation of Extracted Netlists
Transistor and Capacitor Models(PDK)
No
Functionality Met ?
Cell GDS and LEF Views
Yes
Chip Design
Floorplan and Powerplan
Placement (logic and IO cells)
Standard Cell and IO Cell Library
Routing (Clock and Signal)
Technology rule files (.tf,.lef)
Physical Verification (LVS and DRC)

Figure.10.1 Circuit Design and Chip Design methodology for Crosstalk Circuit research

extracted parasitics are back annotated to layout extracted circuit netlists (Calibre PEX automatically performs this task). Simulations are performed on the extracted netlists at different Process (P), Temperature (T), and Voltage (V) corners to verify the functionality and quality of the circuits. All Corner simulations are automated through the Corners feature in Virtuoso Analog Design Environment (ADE) XL [91]. The process corners are considered both for active components (transistors) and passive components (Capacitors). Finally, design iterations are performed in case parasitic or PVT variations affect the quality of the circuits. MOS CAPs are used for the circuits requiring small couplings, and MIM CAPs are used for circuits requiring large couplings. The MOS CAPs consumed extra footprint on the substrate; nevertheless, this implementation serves as a proof-of-concept for Crosstalk computing.

10.2 PVT variation Analysis

Process (P) variation may arise due to various uncertainties in-process steps [92], and Temperature (T) variation is an environmental parameter that can be anywhere between -25C to 125C during the operation of the Chip. Both process and temperature variations ultimately impact transistor performance. Hence, the industry standard to name different Process and Temperature corners is FFF (Fast-Fast-Fast), TTT (Typical-Typical-Typical), and SSS (Slow-Slow-Slow), etc.. The first two letters refer to PMOS and NMOS transistors, and the third letter refers to temperature. For process, F, T, and S correspond to Fast, Typical, and Slow binned transistors (in performance) due to random process variation [92]. For temperature, F is low temperature, S is a high temperature, and T is nominal. However, in Crosstalk circuits, the variation affects not only the circuits' performance but also the functionality. Therefore, this section analyzes the PVT variation effects on the Crosstalk gates and discusses the high fan-in crosstalk gates' vulnerability to variations. It also discusses the hurdle in realizing a high fan-in gate and discusses the technique to overcome it. The variation analysis presented in this chapter is performed on all the prototyped circuits (TSMC 65nm node).

10.2.1 Inverter DC characteristics at TSMC 65nm node at different PVT corners

The circuit topology for all the Crosstalk gates looks identical, with the only difference in aggressors' coupling strength to the victim. The threshold circuit, i.e., CMOS inverter, is common in all the Crosstalk logic gates. So, studying the effect of variation on the inverter's DC transfer characteristics, its trip point, and noise margins can reveal the Crosstalk gates' reliability. Next, sub-sections discuss the individual variation effects (i.e., Process, Voltage, and Temperature) and then consider all variations simultaneously.

10.2.1.1Considering only process variation

The foundry provides three global variation corners as the device models: Slow(S), Typical(T), and Fast(F). Because of the uncertainties in the fabrication processes, the PMOS and NMOS devices on a chip can turn out as either S, T, or F. Thus, we can bin the chip into 5 categories based on the process corners that PMOS and NMOS devices can take. They are SF, SS, TT, FS, and FF. The first letter represents the process corner for NMOS and the second

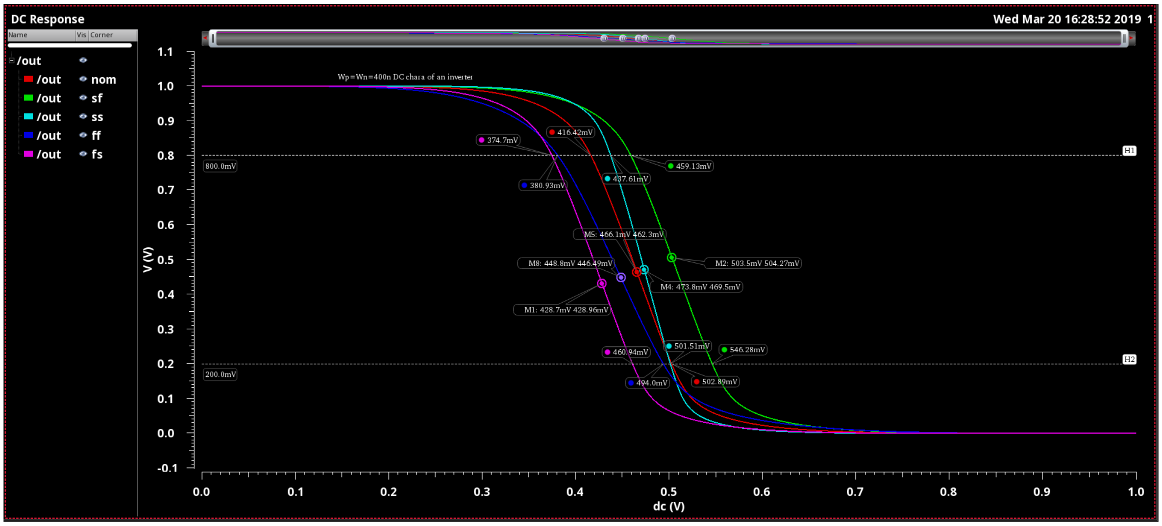


Figure.10.2 Inverter DC characteristics with SF, SS, TT, FS, FF variations

letter represents the process corner for PMOS. Figure.10.2 shows the DC characteristics of the inverter at all these process corners. We can see that the curve shifts left and right in different process corners. This is due to the change in effective ON resistance (R_{ON}) of PMOS and NMOS transistors with process variation. If the inverter has a weak PMOS transistor (Slower corners corresponds to weaker transistors), the switching threshold (Vm) shits the left side. In general, the switching threshold voltage is desirable to be equal to exactly half of VDD, as this would provide symmetrical noise margins for high and low logic levels. Similarly, weak NMOS leads to the right shift of the transfer curve. The inverter's transfer curves at five different process corners (SF, SS, TT, FS, and FF) are shown in Figure.10.2 The trip point of the inverter transfer curves are calculated from the plots in Figure.10.2; the voltage at which the output voltage is equal to the input voltage.

It can be observed that the variation leads to an uncertain shift in the trip point of the inverter. The variation becomes an impediment to the Crosstalk gates functionality as the unwanted shift in the threshold curve, and the trip point could alter the logic behavior. It can be observed that the worst-case corners that affect the functionality of Crosstalk circuits are the curves shifted to left most and right most, which are FS and SF, respectively. This is because for FS, the strong (Fast) NMOS and weak (Slow) PMOS leads to a worst-case left shift of the transfer curve, and for SF, the strong PMOS and weak NMOS leads to a worst-case right shift of the transfer curve. Therefore, from Figure.10.2, the lowest trip point is for the FS corner, which is 0.428V; the highest trip point is for the SF corner, which is 0.503V. The difference between the highest and lowest trip points gives the process margin for which Crosstalk circuit designs can work reliably. That is, the worst-case process shifts should not affect the circuit behavior. So, the process margin that we calculated is 85mV.

10.2.1.2 Considering Process and Temperature Variations

The temperature variation analysis for worst-case FS and SF corners is sufficient. They would give the worst-case variation margin (including temperature) that the Crosstalk circuits have to withstand. Figure.10.3 and Figure.10.4 depict the inverter's DC transfer characteristics with added temperature variations for SF and FS corners, respectively. The temperature extremes, -25C and 125C, and the typical temperature of 25C are considered. The worst slow corner is SF +125C, and the best/fast corner is FS -25C. For these two corners, the variation margin now increases to 105mV, which the Crosstalk circuits have to withstand.

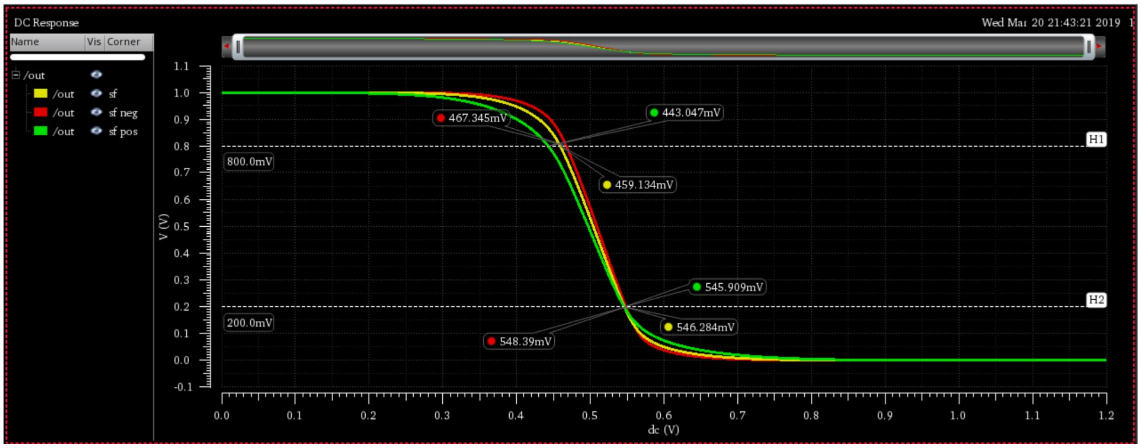


Figure.10.3 Inverter DC characteristics with SF process and Temperature variations

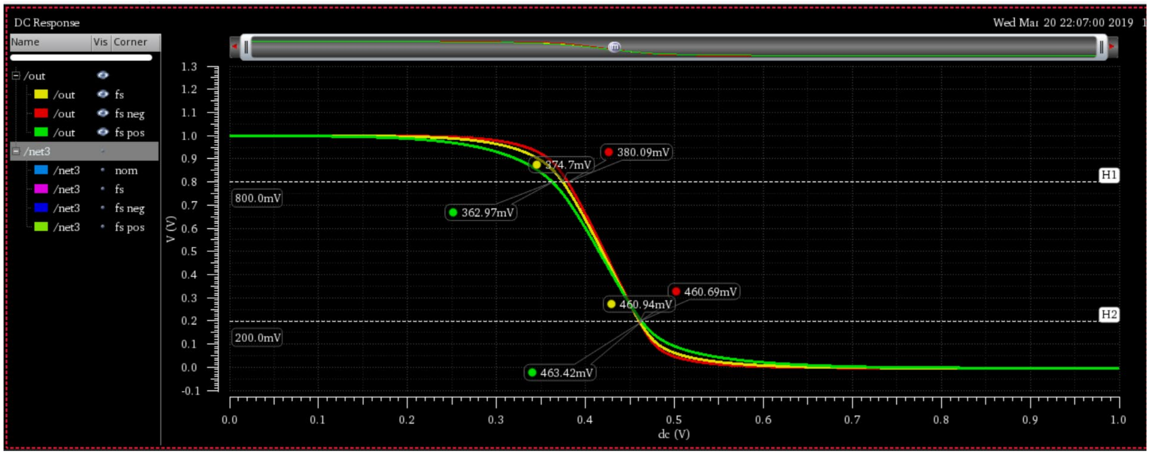


Figure.10.4 Inverter DC characteristics with FS process and Temperature variations

10.2.2 Effect of the functionality margins on the fan-in of the crosstalk gates

The net voltage induced on the *Vi* net can be given by the equation (*4.1*). This equation states that the *Vi* net (for different input logic combinations) takes different intermediate voltages based on the summation of charge induced from all the aggressors. For example, for the AND gate, *the Vi* net voltage will be ~ 400mV for 01 and 10 input combinations and is 800mV for 11 input combinations. From AND gate behavior, 400mV should lead to output logic 0, whereas 800mV should lead to output logic 1. So, the step size from one logic level to the other logic level is 400mV. If we engineer the threshold function to divide the two logic levels in the mid-way, half the step size becomes the noise margin that a given gate can withstand and perform functionally correct. For example, AND Gate has a 200mV noise margin, which is greater than the variation margin (105mV); thus, AND gate is functionally stable with a variation. Likewise, for the 2-in OR gate, the noise margin is 200mV.

Similarly, for all three input Crosstalk gates (AND3, OR3, and Carry), the *Vi* net experiences four levels for various input logic combinations. They are 0V, ~300mV, ~600mV, and ~900mV. The step size, in this case, is 300mV. Therefore, the noise margin becomes ~150mV, which is again greater than the variation margin. Thus, all three input gates function reliably under variation. Figure.10.5 shows the example of a CARRY circuit (3 inputs) simulation. It can be observed that the CARRY

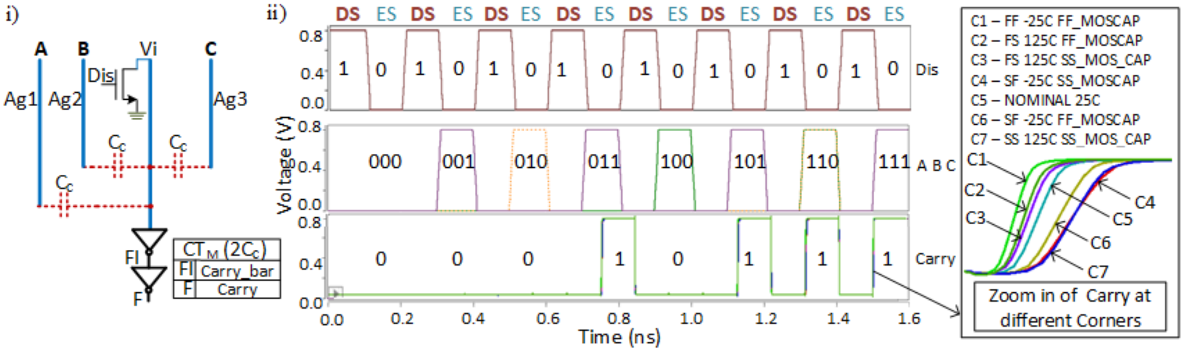


Figure.10.5. i) Crosstalk CARRY Circuit Schematic, ii) Extracted Circuit Simulations at different Corners

circuit works correctly at all process corners (seven worst corners given in the inset figure). The inset figure also depicts the zoom-in of CARRY simulation responses at different corners. We can see that the functionality is not disturbed, but the delay gets affected (similar to CMOS gates).

Similarly, the noise margin calculated for various four-input gates is ~120mV. Though the heterogeneous gates like AO21 and OA21 are three input gates, they would create four voltage levels on the *Vi* net because of their heterogeneous coupling ratios. Thus, we have observed the stable operation of 2-input, 3-input, and 4-input Crosstalk gates with PVT variation. However, for five input gates, the step size is ~200mv, and the noise margin is ~100mv. As the observed variation margin is greater than the 5-input gates' noise margin, they would functionally fail with a variation.

10.2.3 A solution to fix the variation effect on the functionality and achieve high fan-in circuits

Though we have refrained from implementing the five input single-stage circuits in the current prototype, it is possible to reduce the Crosstalk circuit's sensitivity to variation by increasing the threshold voltage. With higher threshold voltage devices, the inverter transfer curve is improved, which reduces the variation margin. Reduction in variation margin makes the Crosstalk circuits more reliable and robust and also enables high fan-in circuits.

The above PVT analysis is performed on all the circuits designed for the prototype Chip. The subsequent section details the Prototype Chip design flow.

10.3 Prototype Chip Design Flow

The Prototype Chip Design flow is also depicted in Figure.10.1. After the final Physical and functional verification of all the custom circuits, they serve as standard cells at the Chip level. A separate top-level circuit schematic (Figure.10.6) is designed for the Chip in Cadence Virtuoso Schematic editor. The Chip layout is designed for the Chip schematic in Layout editor, where the custom circuits designed above are instantiated as standard cells. Figure.10.7

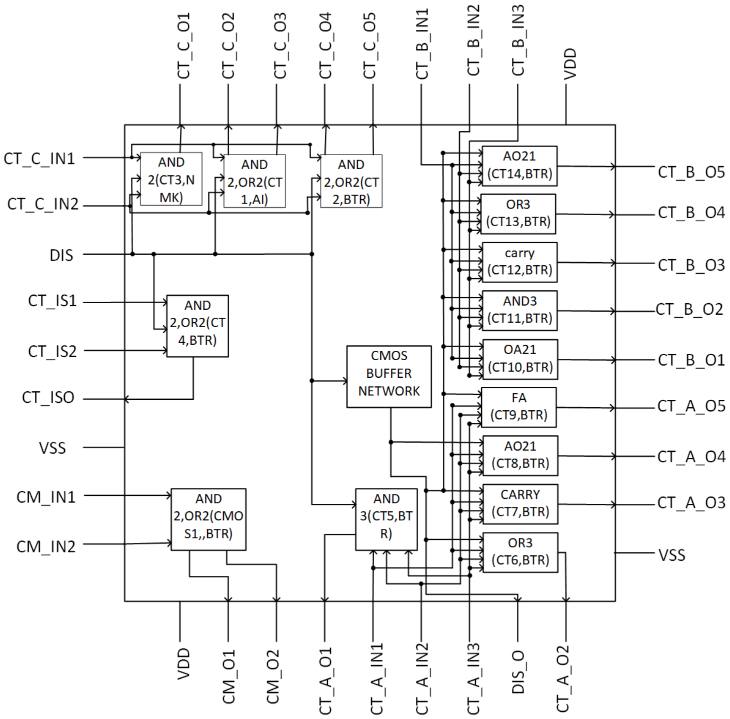


Figure.10.6 Full Chip block diagram

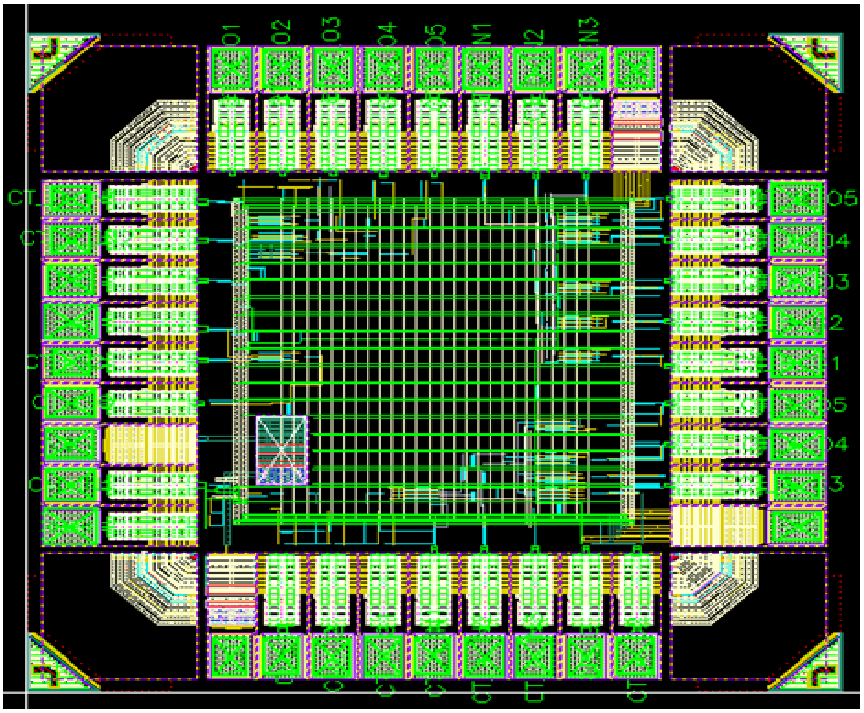


Figure.10.7 Full chip layout diagram

shows the Final Chip layout. It is an IO limited design that mainly consists of the following:

36 IO pads and IO cells, IO power ring, and core power (PG) ring, Power Network to deliver the power to circuits, Crosstalk logic gates, and Clock Network. Only 16 different Crosstalk circuits were instantiated on the die area (near IO pads) to study Crosstalk circuits' practical behavior exclusively. As Crosstalk gates required a clock signal, a clock tree was built, maintaining the drive strength using buffer cells. All routing was done manually. The final chip layout is DRC and LVS verified. The chip was then fabricated through the TSMC multi-project-wafer run (MPW). The chip test and measurement results are presented in the next section.

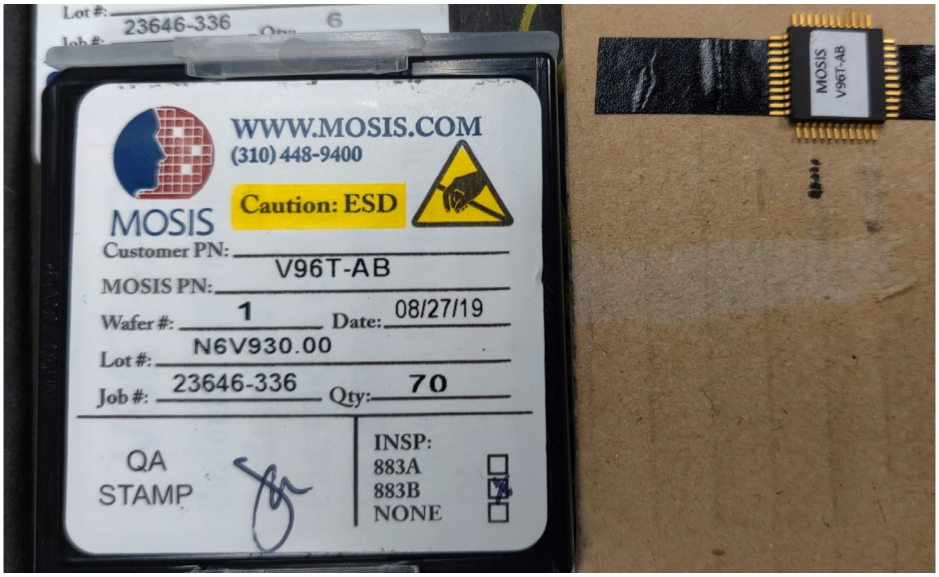
10.4 Details of the Full-Chip

The Crosstalk Chip is implemented using TSMC 65nm PDK, which consists of 9 metal layers. Several crosstalk gates are designed as custom circuit blocks and integrated on the full Chip. The chip's size/area is $1mm^2$ (1mmX1mm), and it is IO limited with 36 pins in total with nine on each side. The full chip schematic is shown in Figure.10.6. It consists of 16 logic gates in total. The type of logic gates, their input, and output pins are shown in the schematic diagram. The final chip layout in Figure.10.7 is comprised of the designed Crosstalk Circuits, clock cells for Clock tree, repeaters/buffer for maintaining signal integrity, I/O cells, I/O pads, corner cells, filler cells, clamp cells, and seal-ring. The capacitors considered are NMOS Capacitors. Their sizes are tailored for different logic gates according to the crosstalk logic function requirement. As Chip consists of only a few numbers of gates, the routing is performed manually. Since Crosstalk gates also require a clock signal (*Dis*), we have manually routed a clock network for each gate, maintaining the drive strength using buffer cells. The operating voltage for the chip is 1V. The Chip is fabricated with TSMC 65nm process technology, under Tiny 2 multi-project-wafer run (MPW), through MOSIS.

Figure.10.8 shows the fabricated Chip. The measurement/test details of the Chip are described in the next section.

10.5 Measurement of fabricated chip:

After obtaining the Chip from the foundry, the measurement is done using the Microxact SPS 100 probe-station [93] and eight micromanipulators. Eight micromanipulators are well aligned as per Crosstalk circuit input and output requirements. A 2-input Crosstalk OR gate requires six I/O connections (two for inputs, one for output, one for discharge signal, two powers and grounds) and a 3-input OR gate requires at least seven I/O connections (three for inputs, one for the output signal, one for Discharge signal, one for *VDD* and one *GND* signal) to the probes. Due to different output pins and shared inputs arrangements among the gates, scope probes are connected to manipulators rather than soldering them for easier connection and movement. For probing, T20-50 Tungsten Probe tips were used as the tip points are thin enough for probing the finest lines on a small pitch. Power (*VDD*) and Ground (*GND*


Lot #: 23646-336 Qty: 6

WWW.MOSIS.COM
(310) 448-9400
Caution: ESD
MOSIS
Customer PN:
MOSIS PN: V96T-AB
Wafer #: 1 Date: 08/27/19
Lot #: N6V930.00
Job #: 23646-336 Qty: 70

QA STAMP

INSP:
883A □
883B ☒
NONE □

Figure.10.8 Fabricated chip

) signals are given from the Keithley 2230G-30-1 DC power supply [94]. A constant 1V DC supply is always supplied to the VDD pin during the measurement. Connection to the GND pin is also ensured by continually checking the current rating at the *GND* pin.

Four input signals are generated using two Tektronix 3100 arbitrary function generators (AFG) [93]. Both the AFGs can generate up to 250 MHz frequency signals, which are sufficient for Crosstalk chip measurement. A 4 channel mixed-signal oscilloscope (Tektronix MSO 4 series) of 1.5 GHz bandwidth [95] is used for measuring both input and output signals. All the input signals and the discharge signal are of 1V continuous square pulse of 10 kHz

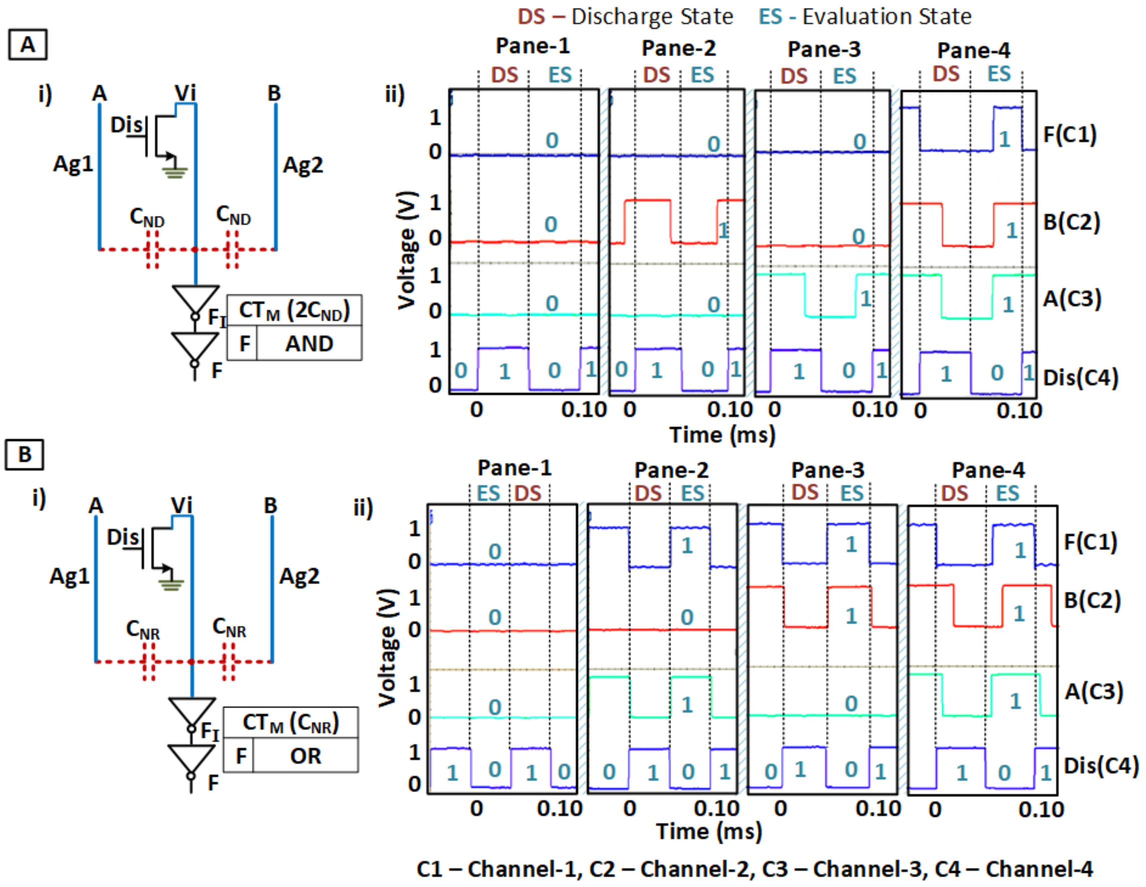


Figure.10.9 Experimental results of Crosstalk Logic gates. A) Crosstalk AND gate; i) Schematic, ii) Experimental results, B) Crosstalk OR gate; i) Schematic ii) Experimental results. Experimental results are showing the snapshot of functional behaviour for all input combinations at different instances.

frequency generated from the AFGs. Figure.10.9 shows the experimental results of the AND & OR Crosstalk logic gates. The circuit operates in two states, the Discharge state DS (when the victim node is connected to ground), and the Evaluation state ES (when the victim is disconnected from power supply or ground and ready for capturing interference). In Figure.10.9.A(ii), 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 ES states, respectively. The fourth row shows the output response of the AND gate. 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, for the OR gate (Figure.10.9.B(i)), the experimental response is shown in the 4th row (bottom to top) of Figure.10.9.B(ii) with the input combinations similar to the AND gate. We can see from Pane-1-4 that during DS (when *Dis*=1), the output becomes logic 0 irrespective of the input signals, from which we can infer that the *Vi* node is discharged to 0 before every new logic computation. But during ES (when *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.

Figure.10.10(i) shows the schematic of the Crosstalk reconfigurable AND2-OR2 circuit fabricated. Its test results can be observed in Figure.10.10(ii). From the bottom, the 1st row shows the discharge signal (*Dis*), the 2nd row shows the control signal (*Ct*), the 3rd row shows inputs *A* and *B*, and finally, the 4th row shows the output *F*. From Figure.10.10(ii), it can be seen that the control signal (*Ct*) is kept low in the first four panes, during which inputs *A* and *B* are given all input combinations, that is, 00 in Pane-1, 01 in Pane-2, 10 in Pane-3, and 11 in

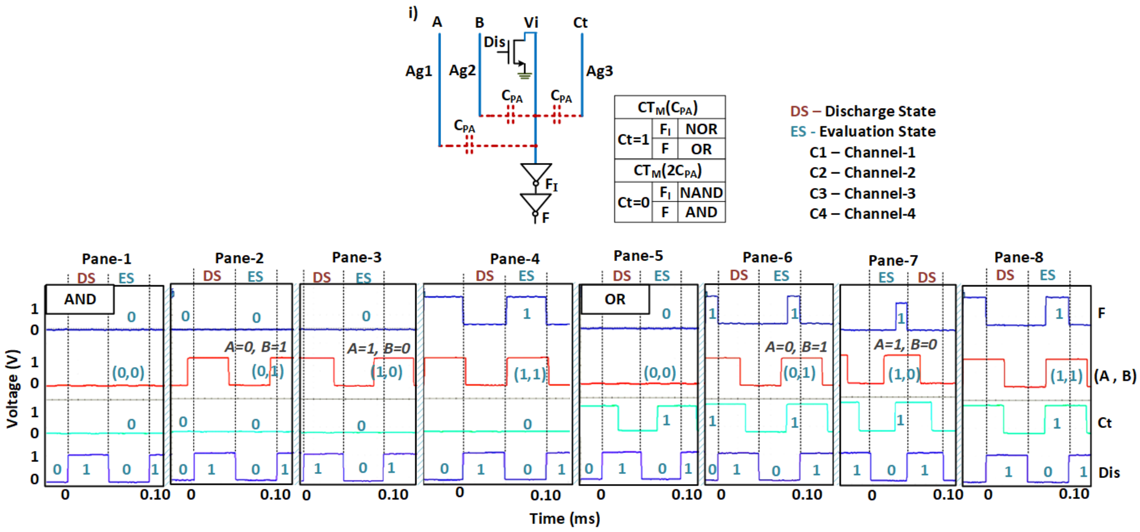


Figure.10.10 Experimental results of Crosstalk Reconfigurable gate; Reconfigurable AND-OR gate: i) Schematic, ii) Experimentals results. Experimental results are showing the snapshot of functional behaviour for all input combinations at different instances.

Pane-4. The output *F* becomes logic 1 only when both the input signals transition to high (Pane-4), hence, behaving as AND gate.

In the next four panes (Pane 5-8), the control signal is kept high during each Evaluation states (ES). It can be seen in Pane-5 that A and B are kept low, whereas *Ct=1*, and subsequently, the output *F* is also low. In the following ES states, pane 6-8, at least one of the input signals *A* or *B* is kept high (i.e., 01 and 10 in Pane-6, and 11 in Pane-7). As a result, the output *F* is high, which shows the OR gate behavior. Since the Mixed Signal Oscilloscope has only four channels to observe the runtime signals, two input signals (*A* & *B*) are synced together and connected to one of the oscilloscope channel (the other three channels probe *Ct, Dis, F* signals). However, during 01 input combination (pane-6), *A* is tied to GND, and B is connected to the source (AFG), and vice-versa for 10 input combinations. The test results in Figure.10.10(ii) show that based on the state of the control signal (Ct), the circuit's behavior can be reconfigured to be either the AND or the OR gate at runtime.

CHAPTER 11

POTENTAIL APPLICATIONS

The inherent hardware-level programmability feature is unique to Crosstalk Computing circuits. This run-time reconfigurability feature could enable a new host of potential applications. Along with the density and power efficiency benefits for mainstream digital electronics, our configurable circuits could spur novel solutions in the realm of hardware security, fault tolerance, and resource sharing, and radiation hardening. The following sections discuss each of the prospects in brief.

11.1 Resource sharing

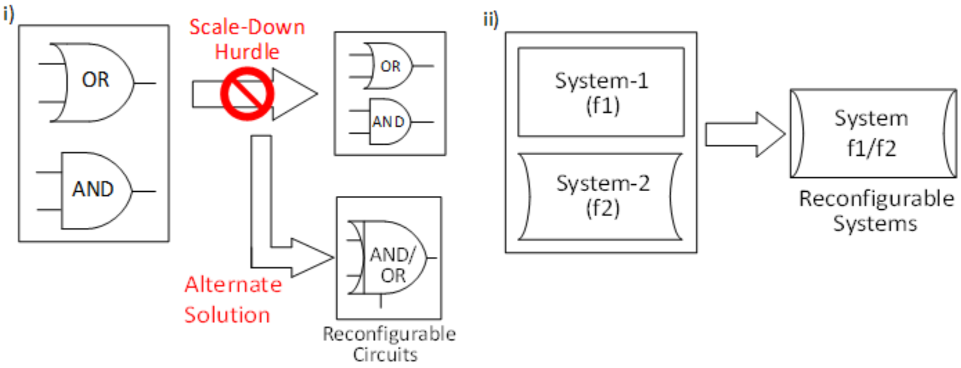


Figure.11.1 Resourse sharing using Crosstalk reconfigurable circuits

The demand for computational resources on a Chip is continuously increasing. For example, the state-of-the-art SoC Chips in High-Performance-Computing (HPC) reported having 54.2 billion transistors on Chip [96]. This demand is ever-growing. As transistors' scaling has slowed down and future nodes face impediments, the frugal usage of resources on Chip will serve as an alternate solution for scaling, thus pushing the boundaries. Resource sharing can serve as one such solution. If the reconfigurable gates can change their functionality at the speed of gate delays using control signals, they can be used for resource sharing. Two example scenarios are shown in Figure11.1. As shown in Figure.11.1(i), if two gates (AND and OR gate) are driven with common inputs and not used simultaneously, we can just use one reconfigurable gate and share the resources instead of using two gates. The concept can also be extended to modular-level (e.g., Multiplier-Adder-Sorter circuit), block-level, and system-level. Figure.9.1(ii) shows two systems with different functions f1 and f2. A reconfigurable system can replace the two systems (able to perform both f1 and f2 functions) to share the resources using the configurable gates judicially.

11.2 Fault Tolerance

The CMOS integrated chips at advanced technology nodes are becoming more vulnerable to various sources of faults like manufacturing imprecisions, variations, aging, etc. Below the 10nm scale, hard and soft errors due to process imprecision, variation, and aging are adversely affecting the yield and reliability of ICs. Additionally, the intentional fault attacks (e.g., high power microwave, cybersecurity threats, etc.) and environmental effects (i.e., radiation) also pose reliability threats to integrated circuits. These risks causing the faults in ICs (both unintentional and intentional fault attacks) are growing in number and severity [97][98]. As a result, reliability concerns are increasing for ICs. Fault-tolerant circuits can help in mitigating the problems and increase reliability. A truly fault resilient circuit scheme can also gracefully recover from run-time faults such as those that incur due to radiation and cyber threats. Traditional approaches for fault tolerance has been concentrated on redundancy based circuits such as CMOS circuit Multiplexing [99], Triple Modular Redundancy (TMR) and its generalized extension N-tuple Modular Redundancy (NMR) [100], Triplicated

Interwoven Redundancy and its generalized extension N-tuple Interwoven redundancy (NIR) [101], and Quadded Logic [102], etc. The need for duplication of logic in all the above approaches/schemes results in a large overhead. A more recent fault tolerance approach looks at circuit level reconfigurability/polymorphism to achieve multiple functionalities with a single logic block. The Crosstalk circuits, which can inherently morph their functionality on the fly (at the speed of gate), could enhance the circuits' fault-resilience/recovery with limited overhead. For example, using such reconfigurable computing units in an ALU would imply that a correct functional output path is possible even when 2/3rd of the ALU is damaged.

The idea of fault tolerance based on reconfigurable circuits is illustrated in Figures.11.2 [103]. Figure.11.2(i) shows a simple circuit having two Crosstalk reconfigurable gates configured to NOR and NAND functions by setting their corresponding *Ct* signals. When a single gate (as shown in Figure.11.2(ii)) is affected by a fault and malfunction, another working gate (the NAND in this example) can be used to perform both functionalities by configuring the corresponding control signal *Ct*. This gate-level reconfigurability concept can be extended to module and system level also.

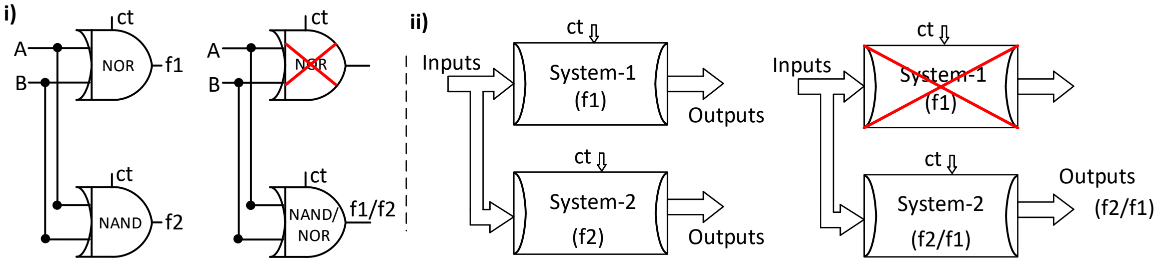


Figure.11.2 Polymorphic/Re-configurable circuit based Fault Tolerance concept, i) Gate-level, ii) System-level

11.2.1 Block-level reconfigurable fault-tolerant scheme

Figure.11.3 [103] shows three Crosstalk polymorphic Adder-Multiplier-Sorter circuits (given in Figure.6.7) configured as independent Multiplier, Sorter, and Adder blocks using

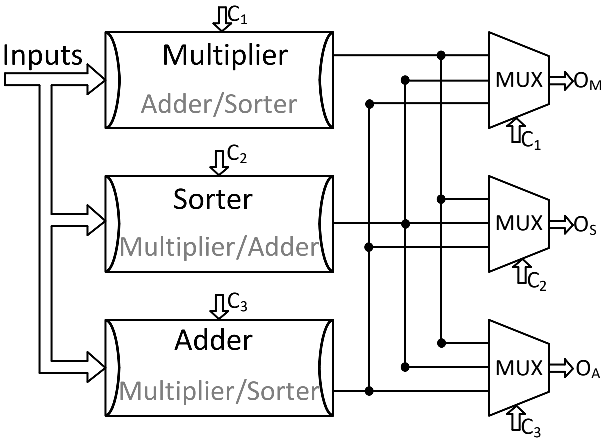


Figure.11.3. Block-Level Polymorphic Fault Tolerant scheme

the control signals (*C₁*, *C₂,* and *C₃*). The three circuits also possess the dormant other two operations. During the event of fault detection in any one of the blocks, the other blocks can be reconfigured and multiplexed to achieve the Adder/Multiplier/Sorter operations as required. The polymorphic blocks can also be used with traditional voter based fault resiliency techniques [100].

11.2.2 System-level reconfigurable fault-tolerant scheme

Figure.11.4 introduces the concept of hardware-software-based fault detection and recovery scheme that can fully utilize the polymorphic circuits to recover from faults at run-time. Here, polymorphic circuit blocks are deployed first and periodically monitored during operation for correctness and recovery. First, a block is configured for one operation, and a known set of inputs are driven to check the functional correctness. If the correct operation is registered, the block and operation are registered in a lookup table. Similarly, all blocks and relevant functionalities are checked, and their information is stored in the lookup table. Upon

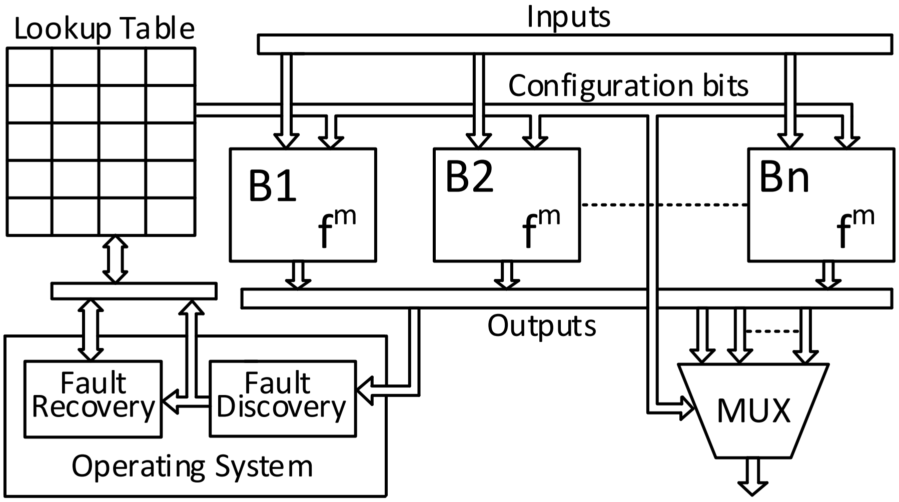


Figure.11.4 System-Level Polymorphic Fault Tolerant scheme

fault detection in one of the blocks, the Software/Assembler will look for alternative blocks

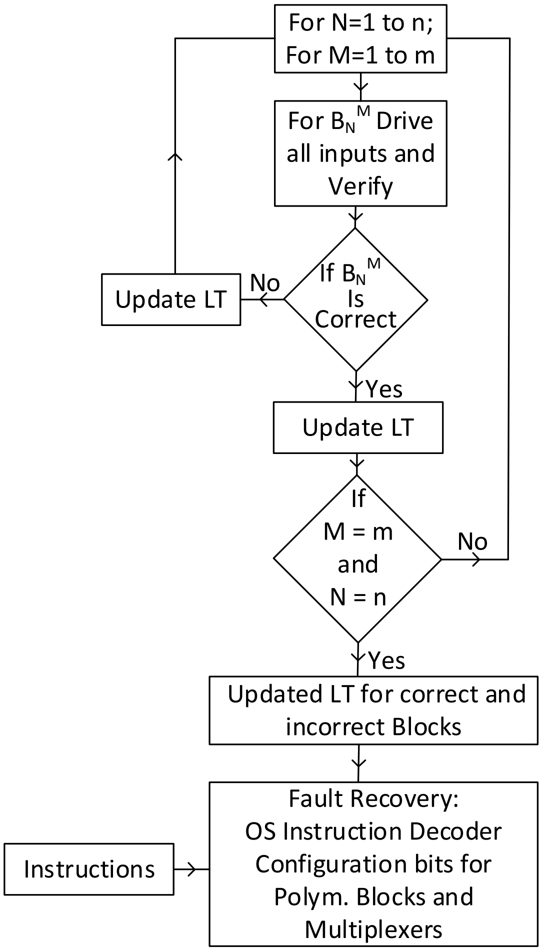


Figure.11.5 Algorithmic Flow chart for proposed system level fault tolerance

in the lookup table and re-route and reconfigure blocks accordingly to achieve correct results. The same Fault Discovery and Fault Recovery steps are detailed in the algorithmic flow chart in Figure.11.5 and detailed algorithm steps in Figure.11.6. The proposed scheme is a straight forward idea to leverage Crosstalk polymorphic gates for system-level fault discovery and recovery purposes, which shows the basic idea. Nevertheless, the proposed schemes can be developed into sophisticated fault-tolerant techniques and methods.

11.3Hardware Security

The rise in connected devices due to the advances in Integrated Circuits (ICs) has also increased sophisticated cybersecurity threats. The ability to hack into ASIC hardware due to the de-centralized assembly of ICs makes them more vulnerable. The identical nature of Crosstalk circuits and the inherent hardware-level programmability feature, which allows run-time reconfiguration of computing blocks, are unique and enables enhance security against hardware-attacks.

Figure.11.7 shows the circuit schematic and layouts of Crosstalk AND gate and OR gate. We can see that circuits and layouts look very similar. Owing to this identical circuit and layout nature of Crosstalk circuits it becomes difficult to counterfeit and reverse engineer the circuits implemented with Crosstalk logic gates. Also, the polymorphic feature in Crosstalk circuits can be leverage to achieve security by obscurity. Hardware obfuscation can disguise the design/circuits and thus make the reverse engineering much more difficult. Moreover, as the polymorphic circuits have other dormant states, we can create hidden states/functionalities in circuits, which can be used for watermarking and fingerprinting.

Fault Discovery

//Assuming n different functional blocks are available and each block can be configured to achieve m different functionalities

Step 1: Configure block1 for functionality 1 by asserting configuration bits

Step 2: Drive known set of inputs for functional verification

Step 3: Check outputs for correctness

Step 4: If outputs are correct/incorrect, mark block1 functionality 1 as correct/incorrect, configure block1 for functionality 2 … functionality m and repeat from Step 2.

Step 5: Repeat Steps 1 to 4 for all computing blocks to discover working functions

Fault Recovery

Step 1: Run Fault Discovery algorithm to discover all correct computing blocks and their respective functions.

Step 2: Operating system stores information about correct blocks and functions in lookup table and generates instructions accordingly

Step 3: From incoming instructions, configure bits are generated during instruction decode phase and all blocks are configured

Step 4: All output multiplexers get proper selection inputs

Step 5: Inputs are driven to computing blocks and outputs observed

Figure.11.6 Algorithm steps for proposed system level fault-tolerance scheme

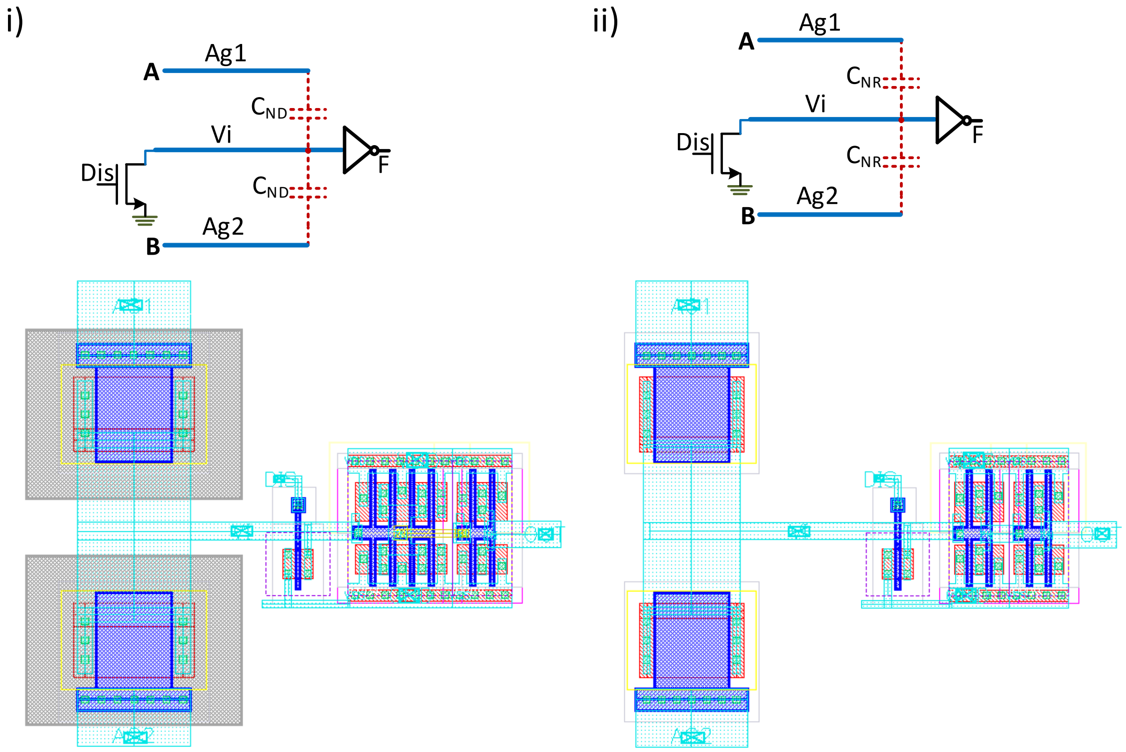


Figure.11.7 i) Crosstalk AND Gate Schematic and Layout, ii) Crosstalk OR Gate Schematic and Layout

It is also shown in [104][105] that dynamically configurable systems are harder to hack. For example, CMOS circuits are vulnerable to side-channel attacks such as differential power analysis attack (DPA), electromagnetic field attack, etc. because of the strong correlation between the information leaking side-channel variables (power, electromagnetic field) and Chip activity. The strong correlation is majorly due to the CMOS circuits' static operation, i.e., the logic states change rarely. However, the Crosstalk circuits are similar to dynamic circuits; they have periodic discharge and logic evaluation states. Therefore, switching activity in Crosstalk Circuits will be more, and hence power profile will be more uniform. This is shown in the simulation plots of CMOS and Crosstalk circuits' instantaneous power in Figure.11.8. In plots, CMOS and Crosstalk gates are of the same logic function and operated

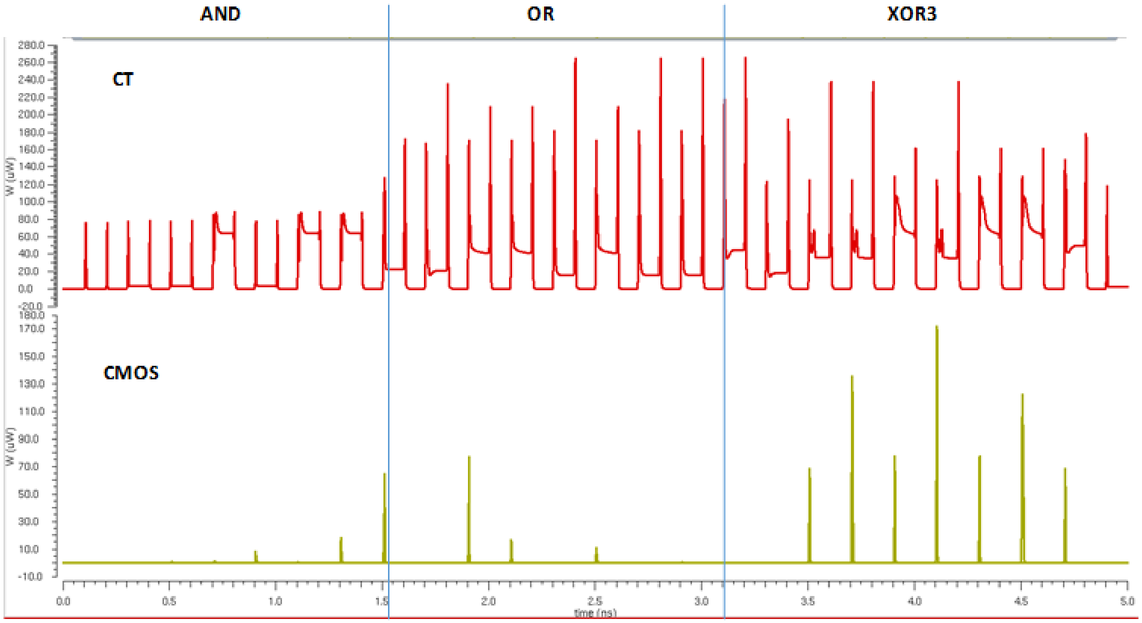


Figure.11.8 Instantaneous power profile for Crosstalk and CMOS gates with same inputs.

with the same inputs. It can be observed that the switching activity in CMOS circuits is less; therefore, power spikes are sparse/intermittent and varying magnitudes. But Crosstalk circuits' power profile is dense/continuous and uniform compared to CMOS. Thus the correlation of power profile to internal information of Chip will be strong in the case of CMOS circuits but weak in the case of Crosstalk circuits. Therefore Crosstalk circuits can be immune to DPA attacks and enhance the security at the hardware level.

11.4 Radiation Hardening

An extensive simulation and analysis work is carried out to investigate the potentials of Crosstalk Circuits in radiation-hardened circuit design [106], which is presented in detail in this section.

The conventional digital ICs based on CMOS circuit style are susceptible to permanent and transient errors in extreme environments such as space due to radiation, primarily due to the sudden build-up of charge in semiconductor devices when stricken by high-energy ions/particle [107]. The failures it can cause are transient errors, MOSFET gate-oxide breakdown [108], junction breakdown due to CMOS latch-ups [109], etc. These effects are expected to worsen with technology scaling. The mitigation techniques to improve the radiation immunity of integrated circuits so far focused on the following approaches: device or substrate engineering [110] using SOI/SOS substrates, radiation hardening by the layout, and circuit optimizations, and device, circuit, and system-level redundancies [111]. While these show some improvements, they incur a density penalty and are inadequate for scaled digital circuits—as the vulnerability of integrated circuits to radiation increases with miniaturization [112]. Moreover, as the demand for reliable and mission-critical applications is ever-growing, the reliability concerns due to the radiation effects on ICs come forefront. Therefore, a scalable solution for radiation resilient digital circuits is essential. In Crosstalk Computing, a byproduct of using interference for computing instead of device switching dependence is that when a high current spike is induced due to radiation, the charges will get shared in coupling capacitances and prevent extra charge build-up in device nodes. For a further increased degree of radiation resilience, we propose an IC fabric that uses ultra-thin body SOI Junction-less nanowire transistors and Crosstalk Computing Circuits in combination. The fabric offers the following radiation-tolerance features inherently:

▪ A novel interconnect-centric circuit implementation style inherently immune to radiation (as the logic computation majorly happens here in interconnects and active device requirement reduces).

▪ The coupling capacitance networks inherent to the fabric would disperse the excess charges deposited by radiation and mitigate the risks.

▪ A fabric-specific dynamic circuit-style that periodically discharges the logic-gates inputs to the ground for the operation shortens the radiation-sensitive window (temporal) of circuits.

▪ Using the junction-less nanowire transistors for the active device and an SOI substrate further damps the radiation effects.

11.4.1 Radiation effects on Integrated Circuits

The sources of radiation (for ground level, aviation, and space applications) are terrestrial radiation (cosmic rays) [107] [113-114], solar radiation [113], and radiation due to radioactive elements [115]. The terrestrial and solar radiation comprises 99% nuclei of well-known atoms and the other 1% of high-energy electrons [116]. The nuclei are mostly high-energy protons (90%), alpha particles (9%), and the remaining 1% are nuclei of heavier elements [116]. Though these radiation particles continuously splatter the earth, most of them (being charged) are deflected into space by the earth's magnetic field. Many are twined and trapped into geomagnetic fields as radiation belts [117]. Because of the varying magnetic field strengths of the earth (weak and strong), few particles make their way into the earth's atmosphere. These high-energy particles reaching the earth's atmosphere undergo a chain of nuclear spallation reactions with the nuclei of atmospheric gases (nitrogen, oxygen, etc.) and produce a shower of secondary particles such as protons, neutrons, heavy-ions, etc. [116]. The secondary neutrons, having longer ranges, reach the earth's surface (as an air shower) through a series of nuclear reactions [116]. These neutrons and all other streams of primary/secondary ions (from ground level to space) act as a threat to ICs deployed at various heights from the ground.

On the other hand, the third source of radiation, radioactive elements, continuously emit the α, β and γ particles. The ICs exposed to these particles get affected, in which α particle, being a heavy ion, affects worse. The radioactive elements can also present internally to the chip if they were contaminated into the die/chip during the fabrication or packaging steps [116].

The radiation particles striking a chip can be categorized into two types based on the physical mechanisms of the radiation effects in ICs: charged particles and uncharged particles. The charged particles are protons, alpha particles, ionized atoms (heavy ions), beta rays (electrons), muons, pions, etc. The uncharged particles are neutrons, photons, neutrinos, gamma rays, etc. The high-energy charged particle (ion) traversing through the semiconductor material ejects the electrons out of the atoms along its track through impact-ionization and thus generates electron-hole pairs. This mechanism is called direct charge generation [118].

The uncharged particles like neutrons do not directly eject the electrons but undergo nuclear reactions with atoms' nuclei and generate secondary ions. These secondary ions, in turn, produce the electron-hole pairs through impact ionization. Therefore, this process is called an indirect charge generation [118]. In both the ways, an excess charge is deposited into the silicon substrate, which gets attracted through drift and diffusion mechanisms towards the sensitive regions such as PN-junctions (Drain-Substrate/Source-Substrate) and collected by the transistor terminals (drain/source).

However, the interaction of metals with radiation is not perceptibly detrimental to circuits [116]. The mobility of electrons in a metal is large, and available energy levels for electrons are continuous (in contrast to semiconductors, metals have no bandgap). Because of these continuous levels, the recombination time of excess carriers is significantly less in metals. When high-energy ions traverse through the metal atoms, they do create excess electrons (excite electrons to higher energy levels) through impact ionization. However, they recombine very fast that they do not show any observable radiation effect [116]. Based on the severity of the deposited charge surge and its consequences, the errors induced in ICs due to radiation are three types, transient errors, permanent errors, and parametric degradation [108]. In case of transient errors, the transistor terminals that are inundated by a surge of unintended charge experience a momentary change in the signals that they carry, i.e., a transient bump occurs in a logic low signal or a transient dip occurs in a logic high signal. This transient pulses might propagate through the circuit as transient errors and get captured in the flip-flops (or they might also get logically/electrically masked). This effect is infamously known as a Single Event Transient (SET)/Single Event Effect (SEE) [118].

Likewise, the permanent faults in ICs majorly occur due to the following catastrophic effects: When an excessive charge sufficient to rupture the gate dielectric of the transistor is deposited [108]; triggering of a latch-up event, which generates an excess disruptive current, leading to burnout [109]; and long-term exposure to radiation leading to gradual deterioration of gate-oxide or other transistor parameters. The additional radiation effects such as Total Ion Dosage (TID) [120], displacement damage [119], etc. can also lead to internal parameter degradation and thus lead to permanent failure of the device/circuit/system. The threats these radiation-induced errors/failures pose in digital-systems are intolerable and catastrophic, and mission-critical in many applications. For example, failure in electronic chips in a spaceship, rockets, automobiles, medical devices, etc. are devastating and life-critical.

As the fluence and intensity of radiation increase with the distance above the earth, the radiation threats on ICs also increase with the height from the earth. Therefore, the applications at the ground-level are at minimal risk, next avionic applications, followed by space applications at maximum risk; for example, at an extreme, the “radiation-belts” around the earth are considered as the dangerous-zone for the satellite’s electronic chips. In addition to these, the radiation-contaminated environments at the ground-level (for example, Fukushima radioactive environment [105], industrial control environments, etc.) also pose high risks. Hence, exploring the techniques for radiation-tolerant integrated circuits is extremely important.

11.4.2 A new integrated circuit fabric for radiation-hardened digital ICs

The proposed IC fabric with nanowire transistors as active devices, SOI substrate, and Crosstalk circuit is depicted in Figure.11.9. It shows the crosstalk NAND gate 3-D view. The Crosstalk Computing layer/layers and circuit aspects are the same as described in previous chapters. Fundamentally, since the mode of operation is different and fewer transistors will be used compared to CMOS for computing, the effect of radiation is expected to be less. Furthermore, instead of relying on semiconductor devices, crosstalk's logic operation is performed in nano-metal lines. Since the metal lines are immune to radiation (because of the reasons discussed in the previous section) [116], the main circuit elements can be said to be immune to radiation.

Unlike CMOS or any other logic styles, the Crosstalk logic circuits are composed of uniform circuit topology, i.e., crosstalk-metal lines, inverter, discharge transistor. Therefore, a uniform grid/template of nanowire fabric effectively implements the crosstalk logic circuits

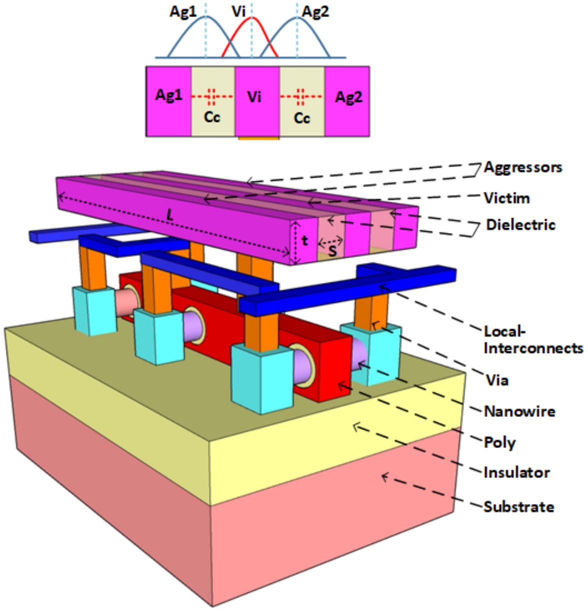


Figure.11.9 Crosstalk NAND gate 3-D view.

with metal lines on the top doing the actual computation (Figure.11.9). The inverters/buffers are used for signal boosting purposes, i.e., maintaining the proper noise margins and signal integrity. The junctionless gate all around Nanowire transistors serve this purpose best at the nanoscale because of their strong electrostatic control over the gate [121,122]. The strong electrostatic control leads to minimum short-channel and second-order effects and provides the best inverter transfer curve for signal regeneration at the nanoscale. These nanowire transistors made on the SOI substrate (Figure.9.9) would also offer reliability and power benefits at the nanoscale. Bulk CMOS circuit implementations (Planar/FinFet) have longer ion track length through transistor drain/source regions, drain-to-substrate PN junctions, and the substrate; therefore, an ion strike results in charge collection from all the areas. However, in the case of SOI circuits, the substrate's charge collection is eliminated, which reduces the severity. Though SOI tri-gate FinFETs have shorter ion track length, they still suffer other radiation effects such as parasitic bipolar amplification [123]. By employing SOI Junctionless nanowire transistors, which are entirely isolated from the substrate, the ion track length is concise. Since there are no PN-junctions, the total charge deposited in this case would be significantly less. Additionally, because of the complete isolation of nanowire transistors by Oxide substrate (SOI) and absence of PN junctions, CMOS like latch-up conditions [124], leading to excess current and permanent failure is also eliminated. Therefore, SOI Junctionless nanowire fabric serves as an excellent candidate to implement crosstalk computing circuits and further enhance radiation tolerance of crosstalk logic circuits.

11.4.3 Methodology for characterizing transient and permanent faults and their mitigation

The capacitive coupling network at the input of inverters in Crosstalk circuits and their mode of operation (alternative discharge and evaluation cycles) both aid in the radiation hardening, which is discussed in this section along with simulation 65nm technology. Two types of fault improvements are detailed, transient, and permanent faults. Transient errors are the momentary glitches in the circuits created by radiation. Figure.11.10 illustrates a transient error scenario both for CMOS (Figure.11.10(i)) and Crosstalk (Figure.11.10(ii)) circuits. Consider an inverter connected to the NOR gate. If radiation strikes at node N (drain on

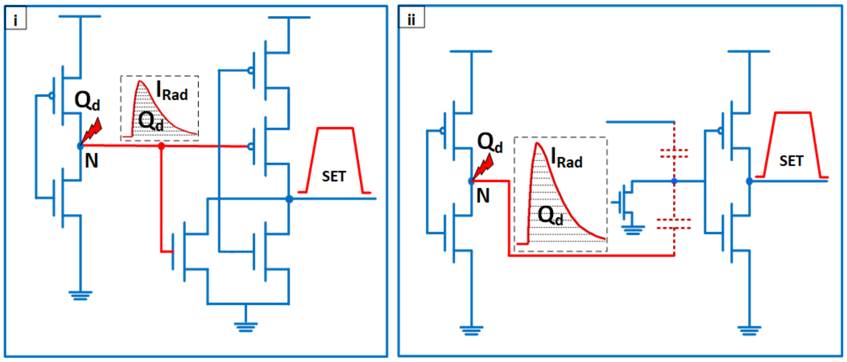


Figure.11.10 Transient Faults in circuits due to Radiation: i) CMOS cascaded circuit; ii) Crosstalk cascaded circuit

NMOS), it deposits a charge Q_D, which is collected by the drain terminal as an unintended current I_{RAD}, which is given by [125],

$$I_{RAD}(t) = I_0(e^{-\frac{t}{\tau\alpha}} - e^{-\frac{t}{\tau\beta}}) \ldots\ldots\ldots\ldots (1)$$

where τ_α is the collection time constant of the junction, τ_β is the time constant for initially establishing the ion, and I_0 in the peak current, which is given by $Q_D/(\tau_\alpha-\tau_\beta)$ [24]. I_{RAD} generates a voltage spike on the input of CMOS/Crosstalk logic gate (here, NOR gate), which propagates through the gate provided the glitch is above the gate's trip-point/threshold. Therefore, the amount of charge required at the input of a gate to trip its state and propagate the glitch is called Critical Charge (Q_C); the error propagates if $Q_D \geq Q_C$, else it gets filtered at gate's input.

11.4.3.1 Charge Sharing in Crosstalk circuits to Minimize the Radiation Effects

Assuming the trip-point for CMOS gates and inverters in Crosstalk gates to be $V_{DD}/2$, the critical charge can be calculated on the first order as $Q_C=C_G(V_{DD}/2)$, where C_G is the input capacitance of the gate. In CMOS gates, $C_G= C_P+C_N$, where C_P and C_N are the input capacitances of PMOS and NMOS transistors, respectively (Figure.11.10). Whereas, for inverters in crosstalk gates, the input is *Vi* node; therefore, $C_G= C_{Vi}= C_C+C_P+C_N$, where C_C is the effective coupling capacitance of inputs. Comparatively, the critical charges for CMOS and Crosstalk are,

$$(Q_C)_{CMOS}=V_{DD}(C_P+C_N)/2 = Q_G$$

$$(Q_C)_{Crosstalk}=V_{DD}(C_C+C_P+C_N)/2 = Q_G + Q_{CC}$$

Thus, the inverter's input capacitance and the coupling capacitance C_C share the charge transferred to the victim node; Hence, the gate's input capacitance effectively increases and makes the Crosstalk gate robust to transient errors. Similarly, if the induced charge/voltage is large enough to break the transistors' gate-oxide or the PN junction, it leads to a permanent error.

Figure.11.11 shows the simulations of transient and permanent errors in CMOS and Crosstalk circuits by incorporating a circuit model for radiation effects. The excess charge deposition by radiation is modeled (in Verilog-A) as a current source given by equation(1). Figure.11.11(i) shows the cascade circuit examples for CMOS and Figure.11.11(ii) for Crosstalk. We have shown the four simulation cases, where case-1 and case-2 are related to the transient failures (in Figure.11.11(iii)), case-3, and case-4 are related to the permanent failures (in Figure.11.12). In all cases, excess charge is induced on output terminal (node *N*) of driving gate G0 by using the current source model; transient errors would receive less Q_D, whereas permanent errors would receive high Q_D. Different to CMOS gates, Crosstalk gates have Discharge (DS) and Evaluation (ES) phases driven by *Dis* signal. Thus, inputs are given only in ES; the CMOS circuit is also given the same inputs at the same time for effective comparison purposes. Case-1 shows the improved radiation immunity of Crosstalk gates

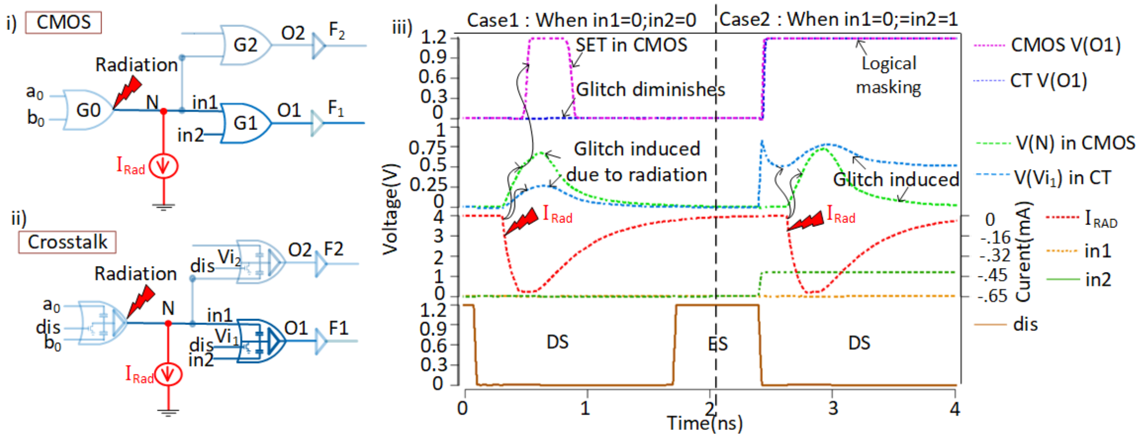


Figure.11.11 Simulation of Transient Errors: i) CMOS cascaded circuit, ii) Crosstalk cascaded circuit iii) Simulation results

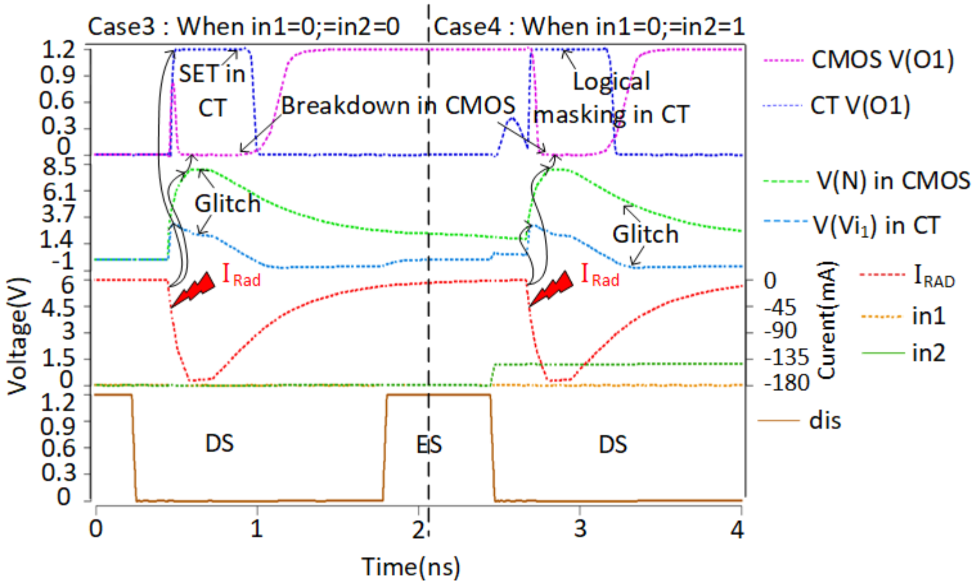


Figure.11.12 Simulation of Permanent Fault in CMOS and Crosstalk Circuits

compared to CMOS gates due to charge sharing. Panel-1 (panel numbers are given from bottom to top) shows the *Dis* signal. G1 receives the inputs 0 0 for inputs in1 and in2, respectively (shown in panel-2). As shown in panel-2, when an excess current I_{RAD} is induced into node N, voltage glitches are developed at the G1 gate's input (panel-3). It can be observed that the CMOS receives a higher glitch, greater than the trip-point of the gate, compared to Crosstalk (for the same Q_D), and flips its output state (panel-4), whereas the crosstalk gate is stable (panel-4). This difference is due to charge sharing discussed above and capacitive voltage division in the network.

The Case-2 shows a logical masking scenario where inputs are 0 and 1 for in1 and in2. With an I_{RAD} (panel-2), voltage glitches are induced on in1. However, inputs 0 1 and 1 1 for the OR gate gives the same output, which is called logical masking. It is to be noted that switch based logic masking in CMOS circuits is maintained in Crosstalk via voltage summation and inverter threshold function. Figure.9.12 demonstrates permanent failure conditions. In cases-

3, both in1 and in2 are given 0 and 0 inputs, and a very large I_{RAD} (~170mA) is induced into the node N. It generates a voltage spike of ~8V at the gate input, that makes the transistors go into the breakdown region. As a result, output goes entirely to zero. The output should have been 1 (due to SET error) if transistors are operated within breakdown limits. However, for the Crosstalk case, the voltage induced on *Vi*-node with this huge current is not as severe as in CMOS (~3V), which makes the spike propagate as a SET (panel-4) but do not cause breakdown to the device. Case-4 is similar to case-3, where inputs are 0 1 instead of 0 0. Here, the CMOS transistors breakdown, whereas, crosstalk gate is still within the breakdown limit (~3V). These scenarios show the improved immunity of crosstalk gates to permanent failures.

11.4.3.2 Temporal Hardening through the periodic discharge

As seen in Chapters 4, 5, and 6, Crosstalk Gates operate dynamically; that is, circuits have an alternate discharge and evaluation states. This operation mode provides the temporal hardening feature to crosstalk gates, which is, crosstalk gates are sensitive to radiation-induced failures only during the logic evaluation phase. If radiation strikes in the discharge stage, the discharge transistors discharge the excess charges to the ground. That means, assuming the 50% duty cycle for *Dis* signal, the radiation immunity of crosstalk gates are 2x more compared to CMOS gates. This scenario is also shown through a simulation in Figure.11.13 for the OR gate (Figure.11.13(i&ii)) with inputs 0 0. As shown in panel-1, radiation current I_{RAD} is induced in both ES and DS phases. During ES, a voltage spike of ~0.26V is generated on the *Vi*-net, which does not propagate as an error at the output (panel-3) in the Crosstalk circuit. But in CMOS, it propagates as a transient error (panel-2 & panel-3). In DS, we only see a small bump on the *vi*-node, which is gradually discharged and damped to the ground. In

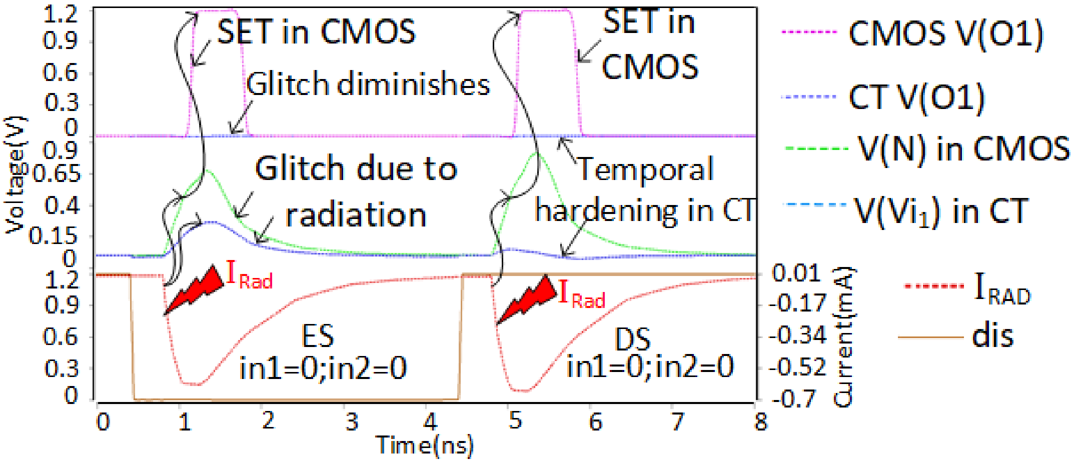


Figure.11.13 Temporal Hardening in Crosstalk circuits

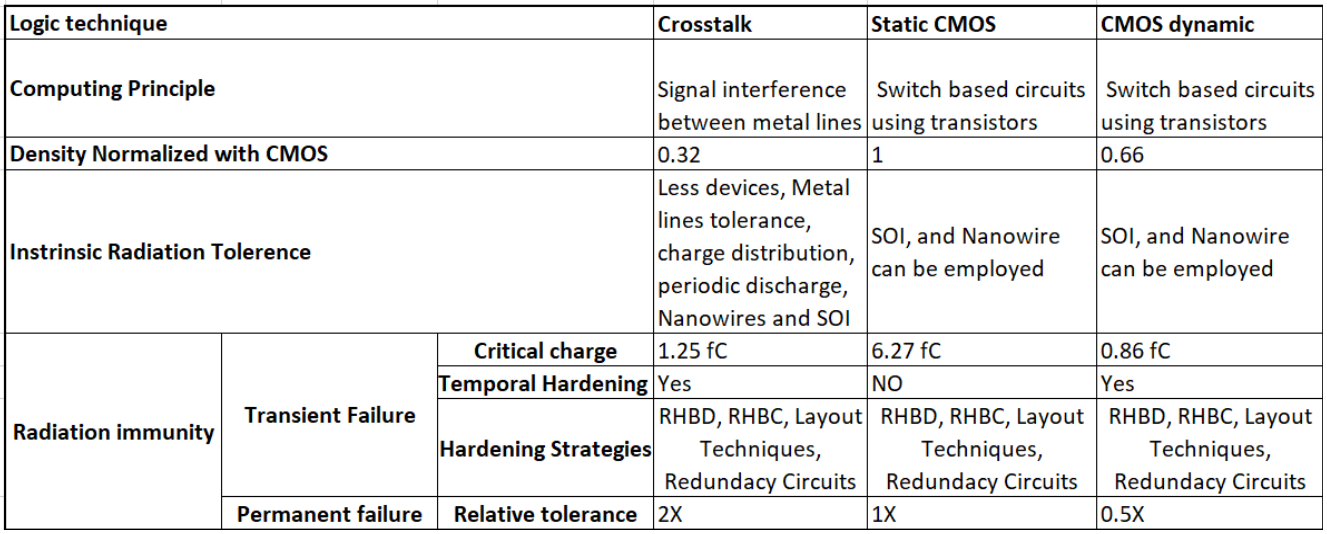
CMOS, it is still a SET error with a significant glitch. Note that the same IRAD is induced both in CMOS and Crosstalk cases. This simulation illustrates the temporal hardening of crosstalk gates to radiation.

11.4.4 Comparison and Summary

The comparative benefits of Crosstalk Circuits concerning static CMOS and Dynamic circuits are presented in Table.11.1. The computing principle is based on signal interference for Crosstalk, whereas it is a switch-based for the other two cases. The density requirement in terms of transistors count is normalized with respect to CMOS. Dynamic logic provides a 40% reduction, whereas Crosstalk logic offers a 68% reduction. Static and dynamic gates do not have any intrinsic radiation tolerance features other than using SOI technology. Whereas, crosstalk has many inherent radiation hardening features: less radiation-sensitive area/regions due to less number of the device; computation in metal lines, which are immune to radiation; charge distribution in capacitive networks; also, SOI and Junctionless nanowire transistors can further enhance the tolerance.

The critical charge calculated based on equation-2 gives 6.27fC, 1.25fC 0.86fC for Crosstalk, Static CMOS, and Dynamic circuits, respectively; hence, Crosstalk circuits are 5x and 2x more robust to radiation compared to static and dynamic circuits. Besides, temporal-hardening is possible for Crosstalk and dynamic circuits but not in Static circuits. Moreover, Crosstalk circuits can equally employ conventional radiation hardening strategies of static and dynamic circuits (circuit techniques, layout techniques, redundancy techniques, etc.). Finally, the crosstalk circuits are 2.5x and 5x more robust for permanent failures compare to static and dynamic circuits, respectively.

TABLE 11.1
Summary of Different Computing Approaches and Radiation Hardening Techniques



Logic technique			**Crosstalk**	**Static CMOS**	**CMOS dynamic**
Computing Principle			Signal interference between metal lines	Switch based circuits using transistors	Switch based circuits using transistors
Density Normalized with CMOS			0.32	1	0.66
Instrinsic Radiation Tolerence			Less devices, Metal lines tolerance, charge distribution, periodic discharge, Nanowires and SOI	SOI, and Nanowire can be employed	SOI, and Nanowire can be employed
Radiation immunity	**Transient Failure**	**Critical charge**	1.25 fC	6.27 fC	0.86 fC
		Temporal Hardening	Yes	NO	Yes
		Hardening Strategies	RHBD, RHBC, Layout Techniques, Redundacy Circuits	RHBD, RHBC, Layout Techniques, Redundacy Circuits	RHBD, RHBC, Layout Techniques, Redundacy Circuits
	Permanent failure	**Relative tolerance**	2X	1X	0.5X

CHAPTER 12

CONCLUSION AND FUTURE WORK

As Crosstalk Computing is novel computing technology rather than just a new circuit technique coherent with incumbent processes and PDKs offered by foundries, it needs many players for efficient implementation. Owing to Crosstalk Computing circuits' benefits and potential applications, it is worth progressing the Crosstalk Computing research towards implementing Hard IPs (Intellectual Property cells/blocks). Towards these goals, the future work can be categorized into two aspects: 1) EDA development for Crosstalk Circuits, and 2) Process, material, and device research to realize Crosstalk Computing specific 3-D capacitances and devices.

12.1 EDA development for Crosstalk Computing

Electronic Design Automation (EDA) for Crosstalk Computing needs research in three aspects: 1) Logic Synthesis, 2) Cell Library development, 3) Place-and-Route flow. The Logic Synthesis tools need enhancement to transform and optimally map the generic structural-netlist to Crosstalk gates (simple and complex logic gates) and polymorphic logic gates. It also needs to perform the transformations required to fix the mismatch nodes, as discussed in Chapter 6. Also, the cell library for Crosstalk gates necessitates some additional characterization features. Finally, Place-and-Route and STA engines need a new flow to handle Crosstalk gates' dynamic logic type circuit architecture. The following three sections present a preliminary exploration in this direction.

12.1.1 EDA flow for Crosstalk Computing

EDA industry is significantly matured over the past five decades and mostly geared for static CMOS circuit style. The adoption of standard-design flows for the Crosstalk circuit can be enabled by incorporating the detailed circuit evaluation and connection policies, and constraints into the design flow [127]. The differences we observe in Crosstalk circuits compared to static CMOS are, 1) Crosstalk gates operate in two states, a pre-discharge state and logic evaluation state, and 2) Crosstalk gates have an additional control signal. Based on these features, Crosstalk circuits appear to be one more kind of dynamic circuits and thus inherit all the EDA challenges of dynamic circuits [128]. But because of their working mechanism, crosstalk gates do not face the dynamic logic issues like monotonicity [129], complimentary cascading links [130][131], trapped inverter problems [132]. However, crosstalk circuits add Crosstalk Computing specific timing and connection constraints to the circuits. These constraints can be modeled into .LIB libraries, which would be honored by the EDA tools during synthesis, and place and route.

12.1.2 Crosstalk Standard Cell Library Characterization

The first task would be to build the crosstalk standard cell libraries. The library files should account for the dual state operation of Crosstalk circuits. I can be accounted for by characterizing two different libraries for Crosstalk Gates, Pseudo-Static Crosstalk Cell Library, and Dynamic Cell Library [133]. The Characterization methodology is shown in Figure.12.1. It consists of Circuit design, layout design, extraction, followed by characterization of extracted SPICE netlists.

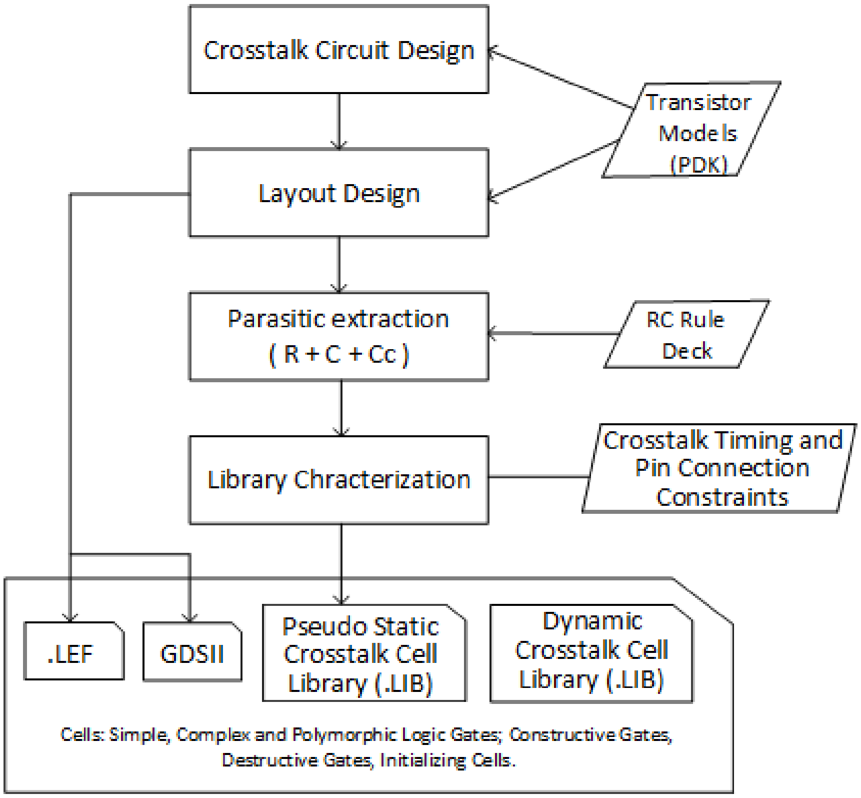

Crosstalk Circuit Design
Transistor Models (PDK)
Layout Design
Parasitic extraction (R + C + Cc)
RC Rule Deck
Library Chracterization
Crosstalk Timing and Pin Connection Constraints
.LEF
GDSII
Pseudo Static Crosstalk Cell Library (.LIB)
Dynamic Crosstalk Cell Library (.LIB)
Cells: Simple, Complex and Polymorphic Logic Gates; Constructive Gates, Destructive Gates, Initializing Cells.

Figure.12.1 Crosstalk Cell Library Characterization Methodology

In Crosstalk circuits, the logic evaluation/propagation happens only in the evaluation state, and there is no effect of the *Clk/Dis* control signal on logic evaluation. Therefore, in the Pseudo static library, the timing arcs are modeled only as data-to-data timing, making characterization similar static CMOS gates, hence called Pseudo-Static Library. Due to the pre-discharge/pre-charge state and logic evaluation state, there would be only one kind of transition in crosstalk gates (either high to low or low to high) during logic evaluation. Therefore, propagation delays and transition times will be accounted only for in these scenarios. Crosstalk Computing specific pin connection constraints will be incorporated in this library. The Synthesis tool will honor these constraints while mapping the library cells. The discharge/pre-charge state also propagates initial states to all *Vi-nets,* and the transition of states is synchronous with the clock. Therefore, these effects are modeled into the Crosstalk Dynamic Cell library using clock-to-data constraints of .LIB standard format.

12.1.3 Synthesis and Place-and-Route Flow

The physical design flow for Crosstalk Computing is depicted in Figure.12.2. The flow inputs are Verilog code, timing constraints, and standard cell library. The HDL design goes

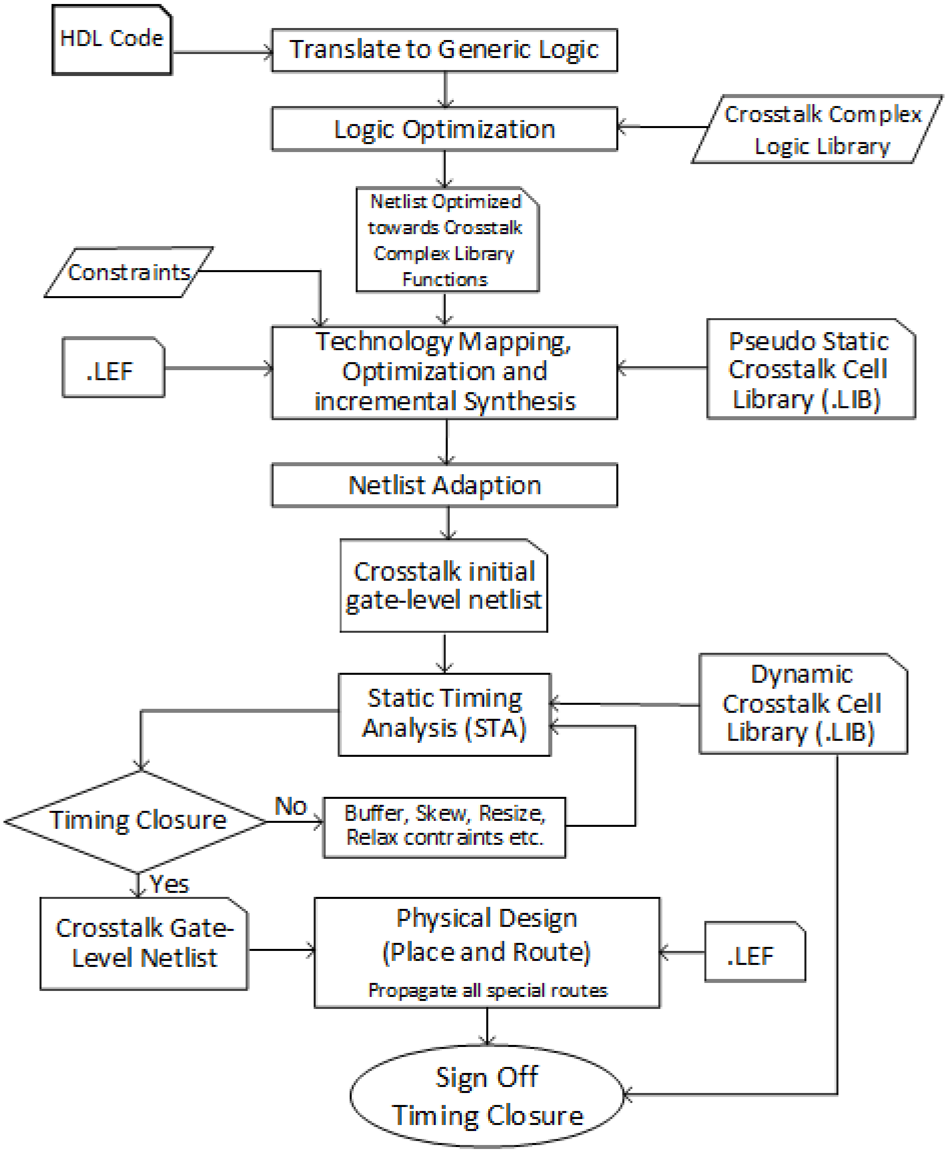

HDL Code
Translate to Generic Logic
Logic Optimization
Crosstalk Complex Logic Library
Netlist Optimized towards Crosstalk Complex Library Functions
Constraints
.LEF
Technology Mapping, Optimization and incremental Synthesis
Pseudo Static Crosstalk Cell Library (.LIB)
Netlist Adaption
Crosstalk initial gate-level netlist
Static Timing Analysis (STA)
Dynamic Crosstalk Cell Library (.LIB)
Timing Closure
No
Buffer, Skew, Resize, Relax contraints etc.
Yes
Crosstalk Gate-Level Netlist
Physical Design (Place and Route)
Propagate all special routes
.LEF
Sign Off Timing Closure

Figure.12.2 Synthesis and Place-and-Route Flow for Crosstalk Computing

through Synthesis and Physical Design steps, as shown in the flow chart. First, the HDL is translated to a netlist consisting of generic logic gates. The generic netlist would then be optimized to use a library of complex functions available in the Crosstalk Cell Library. Then, Technology mapping is performed to the Pseudo-Static Crosstalk Cell library. The gates are accurately modeled for delays in the form of data-to-data timing arcs (clock-to-data is not needed here) in this library. The netlist would now be optimized to meet the design constraints and then incrementally optimized if required. During netlist adaption, the netlist is modified to incorporate *CLK/Dis* signal in all layers of hierarchy. Then STA is performed to check timing. Dynamic Crosstalk Cell Library is given to the timing engine, which would enable us to accurately account for the timing constraints due to crosstalk circuit style and perform buffering, skewing, etc., to close the timing. This prepares the final synthesized netlist, which would be used in place and route physical design steps. The *CLK/Dis* signal can be propagated as a special route. Dynamic Crosstalk Library is for STA checks at various stages and during sign-off timing closure.

12.2 Crosstalk Computing specific 3-D capacitances and devices

The true benefits of Crosstalk Computing can be only unleashed by creating CT-Computing specific Crosstalk coupled capacitive structures/components. Therefore, material and fabrication research is necessary to develop such components/features. As an example, Figure.12.3(i) depicts a vertical/3-D layout style where the coupling-network is envisioned on a metal layer with closely coupled metal-lines, and the three transistors Crosstalk circuitry is underneath the coupling-network. In this way, an additional silicon footprint requirement is mitigated. In conventional Chips, the insulator between the metal lines is always a low-k

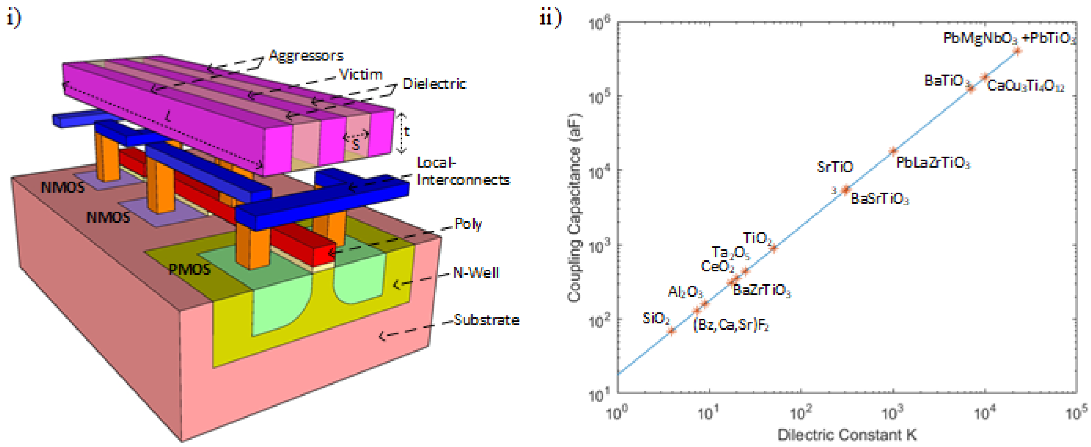


Figure.12.3 i) 3-D Layout for efficient Crosstalk Circuit implementation, ii) Different dielectric materials (K) vs Coupling Capacitances

dielectric. But, by depositing high-k dielectrics in between Crosstalk Computing metal lines, the coupling strengths needed in Crosstalk Circuits can be efficiently achieved. Figure.12.3(ii) shows the dielectric constant (K) of different materials vs. the Coupling capacitance they could offer if placed in between two closely spaced metal lines. Initially, coupling capacitances are extracted for two closely spaced metal-lines from a foundry PDK. Then, by fitting this capacitance value to the standard capacitance equation, the dielectric constant of SiO_2 (used in conventional processes) is replaced with various high-k materials in literature and extrapolated the Coupling capacitance values. It can be observed that we can obtain coupling strengths in the spectrum of three orders, which is promising. Thus high-K material choices would maximize the foot-print benefits in Crosstalk circuits. For example, the OR2 gate requires 4x coupling capacitance compared to the AND2 gate. Using the same dielectric material would mean that the OR gate would consume 4 times the AND2 gate's resources for coupling capacitances (assuming the separation between metal lines is constant). However, using the material with a 4x dielectric constant would result in both AND2 and OR2 gates with the same footprint.

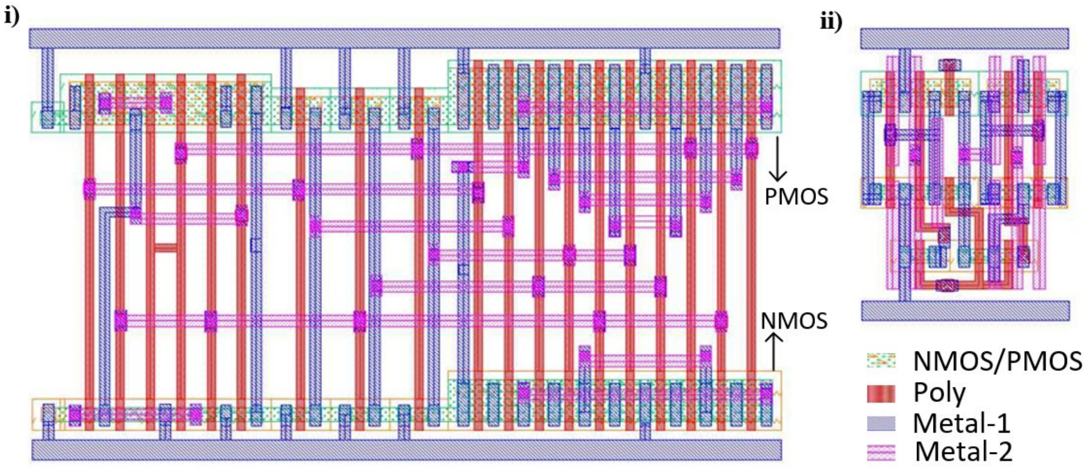


Figure.12.4 Layouts of Full Adder Circuit (Sum and Carry): i) CMOS Layout ii) Crosstalk Layout

Similarly, Figure.12.4 shows the full-adder (SUM and CARRY) layout in CMOS and Crosstalk styles. The Crosstalk layout assumes the Crosstalk couplings are achieved like Figure.12.3(i), by selectively depositing high-k dielectrics between Crosstalk coupled interconnects. The base layer is only for CMOS inverter/buffer and discharge transistor in Crosstalk circuits. The footprints of layouts suggest the density benefit that can be achieved using Crosstalk circuits.

Therefore, future work is to research feasible pathways to achieve the required coupling capacitances in the best possible way. Based on the circuit design requirements and preliminary TCAD simulations of Crosstalk coupled networks, a parameterized capacitor with

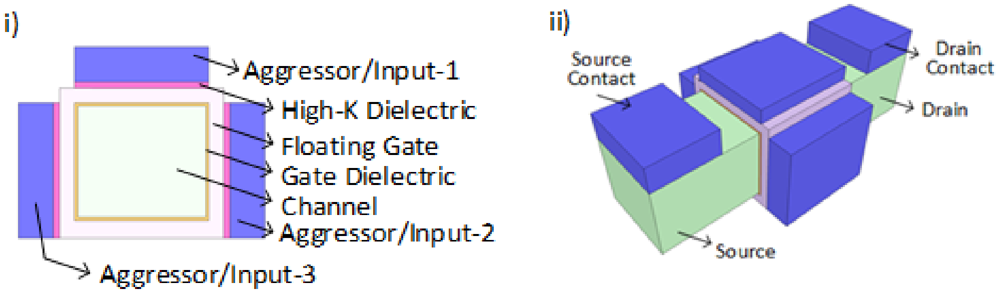


Figure.12.5 i) 3-D FinFET structure with aggressors arranged as three gates and high-k dielectric used for Crosstalk Couplings.

an option to choose between two or three different high-k dielectric materials can serve the purpose and provide optimum designs. Figure.12.5 shows one such idea for implementation; a 3-D FinFET structure with aggressors arranged as gates on three sides and high-k dielectrics used for Crosstalk Couplings. Therefore, the future work is to perform a Crosstalk Computing specific coupling network research exploring the feasible material choices and 3-D structures.

REFERENCES

[1] G. E. Moore, "Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.," in *IEEE Solid-State Circuits Society Newsletter*, vol. 11, no. 3, pp. 33-35, Sept. 2006, doi: 10.1109/N-SSC.2006.4785860.

[2] R. H. Dennard, F. H. Gaensslen, L. Kuhn and H. N. Yu, "Design of micron MOS switching devices," in *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, pp. 35-35, Winter 2007, doi: 10.1109/N-SSC.2007.4785541.

[3] Y. Kim, "Challenges for nanoscale MOSFETs and emerging nano electronics," in *Transactions on Electrical and Electronic Materials,* vol. 11, no. 3, pp. 93–105, 2010, doi:10.4313/TEEM.2010.11.3.093.

[4] N. Z. Haron and S. Hamdioui, "Why is CMOS scaling coming to an END?," *2008 3rd International Design and Test Workshop*, Monastir, 2008, pp. 98-103, doi: 10.1109/IDT.2008.4802475.

[5] P. J. Wright and K. C. Saraswat, "Thickness limitations of SiO/sub 2/ gate dielectrics for MOS ULSI," in *IEEE Transactions on Electron Devices*, vol. 37, no. 8, pp. 1884-1892, Aug. 1990, doi: 10.1109/16.57140.

[6] L. Wang, "Quantum mechanical effects on MOSFET scaling limit", Ph.D. dissertation, Georgia Institute of Technology, August 2006.

[7] V. K. Khanna, "Short-Channel Effects in MOSFETs," in Integrated Nanoelectronics. NanoScience and Technology, New Delhi: Springer, 2016, ch. 5, pp.73-93, doi: 10.1007/978-81-322-3625-2.

[8] Logic Technology, TSMC. Accessed on: October 8, 2020. [Online]. Available: https://www.tsmc.com/english/dedicatedFoundry/technology/logic.htm

[9] D. Brooks, "What’s the future of technology scaling?" Accessed on: October 8, 2020. [Online]. Available: https://www.sigarch.org/whats-the-future-of-technology-scaling/

[10] M. Jurczak, N. Collaert, A. Veloso, T. Hoffmann and S. Biesemans, "Review of FINFET technology," *2009 IEEE International SOI Conference*, Foster City, CA, 2009, pp. 1-4, doi: 10.1109/SOI.2009.5318794.

[11] A. Razavieh, P. Zeitzoff, D. E. Brown, G. Karve and E. J. Nowak, "Scaling challenges of FinFET architecture below 40nm contacted gate pitch," *2017 75th Annual Device Research Conference (DRC)*, South Bend, IN, 2017, pp. 1-2, doi: 10.1109/DRC.2017.7999495.

[12] A. Razavieh, P. Zeitzoff and E. J. Nowak, "Challenges and Limitations of CMOS Scaling for FinFET and Beyond Architectures," in *IEEE Transactions on Nanotechnology*, vol. 18, pp. 999-1004, 2019, doi: 10.1109/TNANO.2019.2942456.

[13] S. Greengard, "Can Nanosheet Transistors Keep Moore's Law Alive?”, in *Communications of the ACM*, March 2020, Vol. 63 No. 3, Pages 10-12, doi: 10.1145/3379493

[14] N. Loubet *et al.*, "Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET," *2017 Symposium on VLSI Technology*, Kyoto, 2017, pp. T230-T231, doi: 10.23919/VLSIT.2017.7998183.

[15] P. Ye, T. Ernst and M. V. Khare, "The last silicon transistor: Nanosheet devices could be the final evolutionary step for Moore's Law," in *IEEE Spectrum*, vol. 56, no. 8, pp. 30-35, Aug. 2019, doi: 10.1109/MSPEC.2019.8784120.

[16] Y. Lee, P. Morrow and S. K. Lim, "Ultra high density logic designs using transistor-level monolithic 3D integration," *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, 2012, pp. 539-546.

[17] N. K. Macha, M. A. Iqbal and M. Rahman, "Fine-grained 3-D CMOS concept using stacked horizontal nanowire," *2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, Beijing, 2016, pp. 151-152, doi: 10.1145/2950067.2950079.

[18] N. K. Macha and M. Rahman, "Cost projections and benefits for transistor-level 3-D integration with stacked nanowires," *2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*, Burlingame, CA, 2017, pp. 1-3, doi: 10.1109/S3S.2017.8309235.

[19] B. Vincent, J. Boemmels, J. Ryckaert and J. Ervin, "A Benchmark Study of Complementary-Field Effect Transistor (CFET) Process Integration Options Done by Virtual Fabrication," in *IEEE Journal of the Electron Devices Society*, vol. 8, pp. 668-673, 2020, doi: 10.1109/JEDS.2020.2990718.

[20] P. Sung *et al.*, "Fabrication of Vertically Stacked Nanosheet Junctionless Field-Effect Transistors and Applications for the CMOS and CFET Inverters," in *IEEE Transactions on Electron Devices*, vol. 67, no. 9, pp. 3504-3509, Sept. 2020, doi: 10.1109/TED.2020.3007134.

[21] N. K. Macha, M. A. Iqbal and M. Rahman, "New 3-D CMOS Fabric With Stacked Horizontal Nanowires," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 9, pp. 1625-1634, Sept. 2019, doi: 10.1109/TCAD.2018.2848588.

[22] N. K. Macha, S. Geedipally and M. Rahman, "Ultra high density 3D SRAM cell design in Stacked Horizontal Nanowire (SN3D) fabric," in *2017 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, Newport, RI, 2017, pp. 155-161, doi: 10.1109/NANOARCH.2017.8053734.

[23] Md. A. Iqbal, N. K. Macha, et al., “Thermal management challenges and mitigation techniques for transistor-level 3-D integration,” in *Microelectronics Journal*, vol. 91, pp. 61-69, Sept. 2019, doi: https://doi.org/10.1016/j.mejo.2019.07.004.

[24] P. S. Goley and M. K. Hudait, “Germanium Based Field-Effect Transistors: Challenges and Opportunities.” *Materials*, 7 (3), 2301-2339, 2014, doi: https://doi.org/10.3390/ma7032301.

[25] H. Riel, L. Wernersson, M. Hong, and J. A. del Alamo, “III–V compound semiconductor transistors—From planar to nanowire structures,” *MRS Bull.*, vol. 39, pp. 668–677, Aug. 2014, doi: https://doi.org/10.1557/mrs.2014.137.

[26] G. Iannaccone, F. Bonaccorso, L. Colombo, et al., "Quantum engineering of transistors based on 2D materials heterostructures," in *Nature Nanotech* 13, 183–191 (2018), doi: https://doi.org/10.1038/s41565-018-0082-6.

[27] G. Hills et al., "Understanding Energy Efficiency Benefits of Carbon Nanotube Field-Effect Transistors for Digital VLSI," in *IEEE Transactions on Nanotechnology*, vol. 17, no. 6, pp. 1259-1269, Nov. 2018, doi: 10.1109/TNANO.2018.2871841.

[28] L. Zhang, M. Chan, et al., *Tunneling Field Effect Transistor Technology*, Switzerland: Springer International Publishing Switzerland, 2016, doi: 10.1007/978-3-319-31653-6.

[29] J. C. Wong and S. Salahuddin, "Negative Capacitance Transistors," in *Proceedings of the IEEE*, vol. 107, no. 1, pp. 49-62, Jan. 2019, doi: 10.1109/JPROC.2018.2884518.

[30] D. Mamaluy and X. Gao, "The Fundamental Downscaling Limit of Field Effect Transistors", in *Applied Physics Letters*, vol. 106, no. 19, pp. 193503, 2015, doi: https://doi.org/10.1063/1.4919871.

[31] M. T. Bohr and I. A. Young, "CMOS Scaling Trends and Beyond," in *IEEE Micro*, vol. 37, no. 6, pp. 20-29, November/December 2017, doi: 10.1109/MM.2017.4241347.

[32] M. T. Bohr and I. A. Young, "CMOS Scaling Trends and Beyond," in *IEEE Micro*, vol. 37, no. 6, pp. 20-29, November/December 2017, doi: 10.1109/MM.2017.4241347.

[33] M. Lapedus, *Transistor Options Beyond 3nm*, Semiconductor Engineering, February 15th, 2018. Accessed on: October 8, 2020. [Online]. Available: https://semiengineering.com/transistor-options-beyond-3nm/

[34] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi and C. S. Hwang, "Memristors for energy-efficient new computing paradigms," *Adv. Electron. Mater.*, vol. 2, pp. 1600090, 2016.

[35] P. Sheridan, W. Lu, "Memristors and Memristive Devices for Neuromorphic Computing," in *Adamatzky A., Chua L. (eds) Memristor Networks*, Springer, Cham, 2014, doi: https://doi.org/10.1007/978-3-319-02630-5_8.

[36] W. Ma, M. A. Zidan and W. D. Lu, "Neuromorphic computing with memristive devices," *Sci. China Inf. Sci.*, vol. 61, no. 6, May 2018, doi: https://doi.org/10.1007/s11432-017-9424-y.

[37] J. J. Yang, D. B. Strukov and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnol.*, vol. 8, no. 1, pp. 13-24, 2013, doi: https://doi.org/10.1038/nnano.2012.240.

[38] D. Marković, A. Mizrahi, D. Querlioz, J. Grollier, "Physics for neuromorphic computing," in *Nat. Rev. Phys.* 2, pp. 499–510, 2020, doi: https://doi.org/10.1038/s42254-020-0208-2, July 2020.

[39] S. Kvatinsky, "Memristor based circuits and architectures," Ph.D. dissertation, Israel Institute of Technology, May 2014. [Online]. Available: http://www2.ece.rochester.edu/users/friedman/Shahar_Kvatinsky_PhD.pdf.

[40] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi and C. S. Hwang, "Memristors for energy-efficient new computing paradigms," in *Adv. Electron. Mater.*, vol. 2, pp. 1600090, 2016.

[41] C. Sung, H. Hwang and I. K. Yoo, "Perspective: A review on memristive hardware for neuromorphic computation," in *J. Appl. Phys.*, vol. 124, no. 15, Oct. 2018.

[42] T. N. Sasamal, A. K. Singh, A. Mohan, *Quantum-Dot Cellular Automata Based Digital Logic Circuits: A Design Perspective*, Singapore: Springer, 2020, doi: 10.1007/978-981-15-1823-2.

[43] K. K. Likharev, "Single-electron devices and their applications," in *Proceedings of the IEEE*, vol. 87, no. 4, pp. 606-632, April 1999, doi: 10.1109/5.752518.

[44] D. Carlton, "Nanomagnetic logic," Ph.D. dissertation, University of California at Berkeley, February 2012.

[45] Z. Liang, "Design, simulation, and optimization of spintronic logic devices," Ph.D. dissertation, The University Of Minnesota, February 2019.

[46] G. De Micheli, Y. Leblebici, M. De Marchi, D. Sacchetto, "Ambipolar silicon nanowire field effect transistor," United States Patent US20130313524A1, November, 2013.

[47] A. O. Orlov, I. Amlani, G. H. Bernstein, C. S. Lent, and G. L. Snider, "Realization of a functional cell for quantum-dot cellular automata," in *Science*, vol. 277, no. 5328, pp. 928-930, 1997, doi: 10.1126/science.277.5328.928.

[48] A. O. Orlov, R. K. Kummamuru, R. Ramasubramaniam, G. Toth, C. S. Lent, G. H. Bernstein, and G. L. Snider, "Experimental demonstration of a latch in clocked quantum-dot cellular automata," in *Applied Physics Letters* 78, no. 11, pp. 1625-1627, 2001, doi: https://doi.org/10.1063/1.1355008.

[49] C. S. Lent, B. Isaksen, and M. Lieberman. "Molecular quantum-dot cellular automata," in *Journal of the American Chemical Society* 125, no. 4, pp. 1056-1063, 2003, doi: https://doi.org/10.1021/acs.jpcc.7b11964.

[50] C. S. Lent and B. Isaksen, "Clocked molecular quantum-dot cellular automata," in *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1890-1896, Sept. 2003, doi: 10.1109/TED.2003.815857.

[51] G. H. Bernstein, A. Imre, V. Metlushko, A. Orlov, L. Zhou, L. Ji, G. Csaba, and W. Porod, "Magnetic QCA systems," in *Microelectronics Journal* 36, no. 7, pp. 619-624, 2005, doi: https://doi.org/10.1016/j.mejo.2004.12.002.

[52] A. Imre, G. Csaba, L. Ji, A. Orlov, G. H. Bernstein, and W. Porod, "Majority logic gate for magnetic quantum-dot cellular automata," in *Science* 311, no. 5758, pp. 205-208, 2006, doi: 10.1126/science.1120506.

[53] F. Perez-Martinez, I. Farrer, D. Anderson, G. A. C. Jones, D. A. Ritchie, S. J. Chorley, and C. G. Smith, "Demonstration of a quantum cellular automata cell in a Ga As⁄ Al Ga As heterostructure," in *Applied physics letters* 91, no. 3, 032102, 2007, doi: https://doi.org/10.1063/1.2759257

[54] C. S. Lent, and G. L. Snider, "The development of quantumdot cellular automata," in *Field-Coupled Nanocomputing*, pp. 3-20. Springer Berlin Heidelberg, 2014, doi: https://doi.org/10.1007/978-3-662-43722-3_1.

[55] R. Tiwari, D. Bastawade, P. Sharan and A. Kumar, "Performance Analysis of Reversible ALU in QCA", in *Indian Journal of Science & Technology*, vol. 10, no. 29, pp. 01-05, 2017, doi: 10.17485/ijst/2017/v10i29/117324.

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

[57] N. K. 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, 2018, pp. 1-7.

[58] "Predictive Technology Model (PTM)." Predictive Technology Model (PTM). Arizona State University. [Online]. Available: http://ptm.asu.edu/

[59] A. Stoica, R. Zebulum, and D. Keymeulen, "Polymorphic Electronics," in *Evolvable Syst. From Biol. to Hardw.*, vol. 2210, pp. 291–302, 2001, doi: https://doi.org/10.1007/3-540-45443-8_26.

[60] A. Stoica, R. S. Zebulum, D. Keymeulen, and J. Lohn, "On polymorphic circuits and their design using evolutionary algorithms," in *Proc. of IASTED International Conference on Applied Informatics AI2002*, Insbruck, Austria, 2002. [Online]. Available: https://www.semanticscholar.org/paper/On-Polymorphic-Circuits-and-Their-Design-Using-Stoica-Zebulum/83e2f0776f0e292983c5f01c2e38ff4ba955c4eb.

[61] L. Sekanina, "Principles and Applications of Polymorphic Circuits," in *Evolvable Hardware*, Natural Computing Series. Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-662-44616-4_8

[62] L. Sekanina, L. Stareček, Z. Kotásek, Z. Gajda, "Polymorphic Gates in Design and Test of Digital Circuits," in *International Journal of Unconventional Computing* 4, pp. 125–142, Philadelphia, 2008. [Online]. Available: https://www.semanticscholar.org/paper/Polymorphic-Gates-in-Design-and-Test-of-Digital-Sekanina-Starecek/4d31e88c2b7847df86d646cf8d9d8b70781975d1.

[63] S. Rakheja and 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), pp. 131-132, Newport, RI, 2017, doi: 10.1109/NANOARCH.2017.8053726.

[64] A. Stocia, D. Keymeulen, V. Duong and C. Salazar-Lazaro, "Automatic synthesis and fault-tolerant experiments on an evolvable hardware platform," *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, Big Sky, MT, USA, 2000, pp. 465-471 vol.5, doi: 10.1109/AERO.2000.878522.

[65] R. Ruzicka and V. Simek, "More Complex Polymorphic Circuits: A Way to Implementation of Smart Dependable Systems", in *ElectroScope Pilsen*, vol. 7, no. 5, pp. 1-6, 2013, ISSN 1802-4564.

[66] L. Sekanina, "Evolution of Polymorphic SelfChecking Circuits," in *Proc. of Evolvable Systems: From Biology to Hardware*, pp. 186 – 197, Berlin, Springer, 2007, doi: https://doi.org/10.1007/978-3-540-74626-3_18.

[67] L. Sekanina, "Design and Analysis of a New Self-Testing Adder Which Utilizes Polymorphic Gates," *2007 IEEE Design and Diagnostics of Electronic Circuits and Systems*, Krakow, 2007, pp. 1-4, doi: 10.1109/DDECS.2007.4295290.

[68] R. Ruzicka and V. Simek, "NAND/NOR gate polymorphism in low temperature environment," *2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, Tallinn, 2012, pp. 34-37, doi: 10.1109/DDECS.2012.6219020.

[69] J. L. Burrows, “Universal logic circuit.” U.S. Patent 4,558,236, issued December 10, 1985.

[70] R. Ruzicka, “New Polymorphic NAND / XOR Gate 2 Known Polymorphic Gates,” *Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science*, vol. 7, pp. 192–196, 2007, ISBN: 9789606766183.

[71] A. Stoica, G. Klimeck, C. Salazar-Lazaro, D. Keymeulen and A. Thakoor, "Evolutionary design of electronic devices and circuits," *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Washington, DC, USA, 1999, pp. 1271-1278 Vol. 2, doi: 10.1109/CEC.1999.782588.

[72] A. Stoica and R. Andrei, "Adaptive and Evolvable Hardware - A Multifaceted Analysis," *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, Edinburgh, 2007, pp. 486-498, doi: 10.1109/AHS.2007.19.

[73] A. Stoica *et al.*, "Evolutionary recovery from radiation induced faults on reconfigurable devices," *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, Big Sky, MT, 2004, pp. 2449-2457 Vol.4, doi: 10.1109/AERO.2004.1368039.

[74] G. W. Greenwood, "On the practicality of using intrinsic reconfiguration for fault recovery," in *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 398-405, Aug. 2005, doi: 10.1109/TEVC.2005.850278.

[75] A. Stoica, R. S. Zebulum, X. Guo, D. Keymeulen, M. I. Ferguson and V. Duong, "Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration," in *IEE Proceedings - Computers and Digital Techniques*, vol. 151, no. 4, pp. 295-300, 18 July 2004, doi: 10.1049/ip-cdt:20040503.

[76] W. Ditto, A. Miliotis, K. Murali, S. Sinha, and M. Spano, “Chaogates: Morphing logic gates designed to exploit dynamical patterns,” *Chaos 20*, 037107, 2010, doi: https://doi.org/10.1063/1.3489889.

[77] W. M. Weber, A. Heinzig, J. Trommer, D. Martin, M. Grube and T. Mikolajick, "Reconfigurable nanowire electronics - A review", *J. on Solid-State Electronics*, vol. 102, pp. 12-24, December 2014, doi: https://doi.org/10.1016/j.sse.2014.06.010.

[78] M. D. Marchi *et al.*, "Configurable Logic Gates Using Polarity-Controlled Silicon Nanowire Gate-All-Around FETs," in *IEEE Electron Device Letters*, vol. 35, no. 8, pp. 880-882, Aug. 2014, doi: 10.1109/LED.2014.2329919.

[79] J. Zhang, P. Gaillardon and G. De Micheli, "Dual-threshold-voltage configurable circuits with three-independent-gate silicon nanowire FETs," *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, Beijing, 2013, pp. 2111-2114, doi: 10.1109/ISCAS.2013.6572291.

[80] 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, 739906, 2009, doi: https://doi.org/10.1117/12.828118.

[81] G. Paasch, T. Lindner, C. Rost-Bietsch, "Operation and Properties of Ambipolar Organic Field-effect Transistors" *Journal of Applied Physics*, vol. 98, no. 8, 2005, doi: https://doi.org/10.1063/1.2085314.

[82] J. Nevoral, R. Ruzicka and V. Simek, "From Ambipolarity to Multifunctionality: Novel Library of Polymorphic Gates Using Double-Gate FETs," *2018 21st Euromicro*

Conference on Digital System Design (DSD), Prague, 2018, pp. 657-664, doi: 10.1109/DSD.2018.00111.

[83] N. K. Macha, B. T. Repalle, Md. A. Iqbal, M. Rahman, "Crosstalk Computing based Gate-Level Reconfigurable Circuits," TNANO, uder review.

[84] Md. A. Iqbal, N. K. Macha, et al., “From 180nm to 7nm: Crosstalk Computing Scalability Study,” *IEEE/ACM NANOARCH*, Quingdao, China, 2019.

[85] M. A. Iqbal, N. K. Macha, B. T. Repalle and M. Rahman, "Designing Crosstalk Circuits at 7nm," *2019 IEEE International Conference on Rebooting Computing (ICRC)*, San Mateo, CA, USA, 2019, pp. 1-4, doi: 10.1109/ICRC.2019.8914701.

[86] Cadence. Virtuoso Schematic Editor. Version -2019. [Online]. Available: https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-schematic-editor.html (2019).

[87] Synopsys. Synopsys HSPICE Simulator. Software. Version K-2015. [Online]. Available: https://www.synopsys.com/verification/ams-verification/hspice.html (2015).

[88] Cadence. Virtuoso Layout Suite. Version -2019. [Online]. Available: https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/layout-design/virtuoso-layout-suite.html.

[89] Mentor. Physical Verification. Calibre nmDRC and Calibre nmLVS. [Online]. Available: https://www.mentor.com/products/ic_nanometer_design/verification-signoff/circuit-verification/calibre-xrc/

[90] Mentor. Parasitic Extraction. Calibre PEX. [Online]. Available: https://www.mentor.com/products/ic_nanometer_design/verification-signoff/physical-verification/

[91] Cadence. Virtuoso Analog Design Environment. Version -2019. [Online]. Available: https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-analog-design-environment.html.

[92] Y. Li, C. Hwang, T. Li and M. Han, "Process-Variation Effect, Metal-Gate Work-Function Fluctuation, and Random-Dopant Fluctuation in Emerging CMOS Technologies," *in IEEE Transactions on Electron Devices*, vol. 57, no. 2, pp. 437-447, Feb. 2010, doi: 10.1109/TED.2009.2036309.

[93] Arbitrary Function Generator. [Online]. Available: https://www.tek.com/signal-generator/afg31000-function-generator.

[94] DC Power Supply. [Online]. Available: https://www.tek.com/tektronix-and-keithley-dc-power-supplies/keithley-2220-2230-2231-series.

[95] Mixed Signal Oscilloscope. [Online]. Available: https://www.tek.com/datasheet/4-series-mso.

[96] Nvidia A100 Tensor Core GPU. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/nvidia-ampere-architecture-whitepaper.pdf

[97] S. S. Sapatnekar, "Overcoming Variations in Nanometer-Scale Technologies," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 1, pp. 5-18, March 2011, doi: 10.1109/JETCAS.2011.2138250.

[98] J. Knechtel, "Hardware security for and beyond CMOS technology: An overview on fundamentals applications and challenges", *Proc. Int. Symp. Phys. Design (ISPD)*, pp. 75-86, Mar. 2020.

[99] J. von Neumann, “Probabilistic Logics and the Synthesis of Reliabl Organisms from Unreliable Components,” *AutomataStudies*, no. 34, pp. 43-99.

[100] E. Dubrova, *Fault-tolerant design*, Springer, 2013, doi: 10.1007/978-1-4614-2113-9.

[101] J. Han, J. Gao, P. Jonker, Y. Qi and J. A. B. Fortes, "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," in *IEEE Design & Test of Computers*, vol. 22, no. 4, pp. 328-339, July-Aug. 2005, doi: 10.1109/MDT.2005.97.

[102] J. Han, E. Leung, L. Liu and F. Lombardi, "A Fault-Tolerant Technique Using Quadded Logic and Quadded Transistors," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 8, pp. 1562-1566, Aug. 2015, doi: 10.1109/TVLSI.2014.2341610.

[103] N. K. Macha, B. T. Repalle, S. Geedipally, R. Rios and M. Rahman, "A New Paradigm for Fault-Tolerant Computing with Interconnect Crosstalks," *2018 IEEE International Conference on Rebooting Computing (ICRC)*, McLean, VA, USA, 2018, pp. 1-6, doi: 10.1109/ICRC.2018.8638601.

[104] 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.

[105] P. M. Garner, C. Boone, and D. J. Cepulis, "Dynamically configurable portable computer system," U.S. Patent No. 5,014,193. 7 May 1991.

[106] 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.

[107] JEDEC Standard JESD89A, "Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray-Induced Soft Errors in Semiconductor Devices, " *JEDEC*, 2006.

[108] J. R. Schwank *et al.*, "Radiation Effects in MOS Oxides," in *IEEE Transactions on Nuclear Science*, vol. 55, no. 4, pp. 1833-1853, Aug. 2008, doi: 10.1109/TNS.2008.2001040.

[109] W. Morris, "Latchup in CMOS," *2003 IEEE International Reliability Physics Symposium Proceedings, 2003. 41st Annual.*, Dallas, TX, USA, 2003, pp. 76-84, doi: 10.1109/RELPHY.2003.1197724.

[110] J. R. Schwank, V. Ferlet-Cavrois, M. R. Shaneyfelt, P. Paillet and P. E. Dodd, "Radiation effects in SOI technologies," in *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 522-538, June 2003, doi: 10.1109/TNS.2003.812930.

[111] R. E. Lyons and W. Vanderkulk, "The Use of Triple-Modular Redundancy to Improve Computer Reliability," in *IBM Journal of Research and Development*, vol. 6, no. 2, pp. 200-209, April 1962, doi: 10.1147/rd.62.0200.

[112] A. H. Johnston, "The Effect of Device Scaling on Single-Event Effects in Advanced CMOS Devices." (2005). [Online]. Available:

https://nepp.nasa.gov/files/11953/Task%2015%20-%20Effect%20of%20Device%20Scaling%20on%20SEE%20in%20Advanced%20CMOS%20Devices%20-%20102197.3.13.4%20Johnston.pdf.

[113] Solar Energetic Particles and Cosmic Rays. [Online]. Available: http://www.solar-system-school.de/lectures/marsch/7.pdf (accessed 14 February 2013)

[114] JEDEC Standard JESD89, "Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Errors in Semiconductor Devices." *JEDEC*, pp. 1–63, 2001.

[115] Technical Committee on Semiconductor Reliability, Semiconductor Jisso & Product Technology Committee, Japan Electronics and Information Technology Industries Association (2011) EITA View Concerning Effects of Radioactive Materials Released from Fukushima Nuclear Power Plant on Semiconductor LSI Products, http://semicon.jeita.or.jp/hp/srg/docs/JEITA-SERPG-View_en.pdf (accessed 9 June 2014).

[116] E. H. Ibe, *Terrestrial radiation effects in ULSI devices and electronic systems*, Wiley-IEEE Press, 2014, ISBN: 978-1-118-47929-2.

[117] N. Merabtine, et al., "Radiation effects on electronic circuits in a spatial environment." *Semiconductor Physics Quantum Electronics & Optoelectronics*, 2004, doi: 10.15407/spqeo7.04.395.

[118] S. Yue, et al., "Modeling and simulation of single-event effect in CMOS circuit," *Journal of Semiconductors,* 36.11: 111002, 2015, doi: 10.1088/1674-4926/36/11/111002.

[119] G. C. Messenger, "A summary review of displacement damage from high energy radiation in silicon semiconductors and semiconductor devices," in *IEEE Transactions on Nuclear Science*, vol. 39, no. 3, pp. 468-473, June 1992, doi: 10.1109/23.277547.

[120] M. Poizat, “Total Ionizing Dose,” *European Space Agency (ESA)*, [Online]. Available: https://indico.cern.ch/event/635099/contributions/2570674/attachments/1456398/2249969/Radiation_Effects_and_RHA_ESA_Course_9-10_May_2017_TID_MP_FINAL_WIN.pdf

[121] L. Ansari et al., “Simulation of junctionless Si nanowire transistors with 3 nm gate length,” *Appl. Phys. Lett.*, vol. 97, no. 6, 2010, 640 Art. no. 062105, doi: https://doi.org/10.1063/1.3478012.

[122] S. Migita, Y. Morita, M. Masahara and H. Ota, "Electrical performances of junctionless-FETs at the scaling limit (LCH = 3 nm)," *2012 International Electron Devices Meeting*, San Francisco, CA, 2012, pp. 8.6.1-8.6.4, doi: 10.1109/IEDM.2012.6479006.

[123] D. Munteanu and J. Autran, "3-D Simulation Analysis of Bipolar Amplification in Planar Double-Gate and FinFET With Independent Gates," in *IEEE Transactions on Nuclear Science*, vol. 56, no. 4, pp. 2083-2090, Aug. 2009, doi: 10.1109/TNS.2009.2016343.

[124] K. Soliman and D. K. Nichols, "Latchup in CMOS Devices from Heavy Ions," in *IEEE Transactions on Nuclear Science*, vol. 30, no. 6, pp. 4514-4519, Dec. 1983, doi: 10.1109/TNS.1983.4333163.

[125] C. Qi, L. Xiao, T. Wang, J. Li and L. Li, "A Highly Reliable Memory Cell Design Combined With Layout-Level Approach to Tolerant Single-Event Upsets," in *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 3, pp. 388-395, Sept. 2016, doi: 10.1109/TDMR.2016.2593590.

[126] M. Li, “Study of Layout Techniques in Dynamic Logic Circuitry for Single Event Effect Mitigation,” Diss. University of Saskatchewan, 2015.

[127] G. D. Hachtel and F. Somenzi, *Logic Synthesis and Verification Algorithms*. Heidelberg, Germany: Springer, 2006, doi: 10.1007/b117060.

[128] G. Yee and C. Sechen, "Dynamic logic synthesis," *Proceedings of CICC 97 - Custom Integrated Circuits Conference*, Santa Clara, CA, USA, 1997, pp. 345-348, doi: 10.1109/CICC.1997.606644.

[129] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Boston, MA, USA: Pearson Education, 2011.

[130] R. H. Krambeck, C. M. Lee and H. -F. S. Law, "High-speed compact circuits with CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 17, no. 3, pp. 614-619, June 1982, doi: 10.1109/JSSC.1982.1051786.

[131] T. Williams, ‘‘Dynamic logic: Clocked and asynchronous,’’ in *Proc. IEEE Int. Solid State Circuits Conf. Tuts.*, Feb. 1996, pp. 1–24.

[132] R. Hossain, *High Performance ASIC Design*, Cambridge, U.K.: Cambridge Univ. Press, 2008, doi: https://doi.org/10.1017/CBO9780511541162.

[133] V. Yuzhaninov, I. Levi and A. Fish, "Design Flow and Characterization Methodology for Dual Mode Logic," in *IEEE Access*, vol. 3, pp. 3089-3101, 2015, doi: 10.1109/ACCESS.2016.2514398.

[134] F. Jazaeri, A. Beckers, A. Tajalli and J. Sallese, "A Review on Quantum Computing: From Qubits to Front-end Electronics and Cryogenic MOSFET Physics," *2019 MIXDES - 26th International Conference "Mixed Design of Integrated Circuits and Systems"*, Rzeszów, Poland, 2019, pp. 15-25, doi: 10.23919/MIXDES.2019.8787164.

[135] N. Weste and D. Harris. 2010. *CMOS VLSI Design: A Circuits and Systems Perspective (4th. ed.)*. Addison-Wesley Publishing Company, USA.

[136] A. Vittal, L. H. Chen, M. Marek-Sadowska, K. Wang and S. Yang, "Crosstalk in VLSI interconnections," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 12, pp. 1817-1824, Dec. 1999, doi: 10.1109/43.811330.

[137] J. A. Davis, et al., "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.

[138] R. A. M. Razif, et al., "Mitigation techniques for crosstalk in ICs," *IOP Conf. Ser.: Mater. Sci. Eng.* 701:012037, 2019, doi: 10.1088/1757-899X/701/1/012037.

[139] Saini S., *Low power interconnect design*, New York, NY: Springer; 2015, doi: https://doi.org/10.1007/978-1-4614-1323-3_3.

VITA

Naveen Kumar Macha received his Masters's degree in Electrical and Computer engineering at the University of Missouri Kansas City (UMKC), Kansas City, Missouri, in 2016. He is pursuing his Doctoral degree in Electrical and Computer Engineering at UMKC. Currently, he is a graduate research assistant at the Nano-Computing group, UMKC. He works on a novel digital integrated circuit research called Crosstalk Computing and Nanowire based 3-D integrated circuits. Naveen has received the patent for his Crosstalk Computing work. He published eleven conferences and four journal papers as author and co-author. Naveen was also awarded the "UMKC School of Graduate Studies Research Grant" in 2019 and 2017. Naveen worked as an Entrepreneurial Lead (EL) for his team in the National Science Foundation (NSF) I-Corps program. During his Ph.D. program, he also pursued three internships to gain the industry experience needed for his research work. He worked as an SoC Physical Design Intern at Synopsys, Digital Circuit and Physical Design intern at Apple, and as a Physical Design Methodology intern at NVIDIA.