# DESIGN AND PRACTICAL REALIZATION OF POLYMORPHIC CROSSTALK CIRCUITS USING 65NM TSMC PDK

A THESIS IN

Computer and Electrical Engineering

Presented to the Faculty of the University

of Missouri–Kansas City in partial fulfillment of

the requirements for the degree

MASTER OF SCIENCE

by

Bhavana Tejaswini Repalle

B.TECH., Vignan's Lara Institute of Technology and Science, A.P, India, 2015

Kansas City, Missouri
2019

DESIGN AND PRACTICAL REALIZATION OF POLYMORPHIC CROSSTALK CIRCUITS USING 65NM TSMC PDK

Bhavana Tejaswini Repalle, Candidate for the Master of Science Degree

University of Missouri–Kansas City, 2019

## ABSTRACT

As the technology node scales down, the coupling capacitance between the adjacent metal lines increases. With an increase in this electrostatic coupling, the unwanted signal interference also increases, which is popularly called as Crosstalk. In conventional circuits, the Crosstalk affects either functionality or performance or both. Therefore the Crosstalk is always considered as detrimental to the circuits, and we always try to filter out the Crosstalk noise from signals. Crosstalk Computing Technology tries to astutely turn this unwanted coupling capacitance into computing principle for digital logic gates[1, 2]. The special feature of the crosstalk circuits is its inherent circuit mechanism to build polymorphic logic gates[3]. Our team has previously demonstrated various fundamental polymorphic logic circuits [1-6,16-18]. This thesis shows the design of the large-scale polymorphic crosstalk circuits such as Multiplier–Sorter, Multiplier–Sorter–Adder using the fundamental polymorphic gates, and also analyzes the Power, Performance, and Area (PPA) for these large-scale designs. Similar to the basic and complex polymorphic gates, the functionality of the large-scale polymorphic circuits can also be altered using the control signals. Owing to their multi-functional embodiment in a single circuit, polymorphic circuits find a myriad of useful applications such as reconfigurable system design, resource sharing, hardware security, and fault-tolerant circuit design, etc. [3]. Also, in this thesis, a lot of studies have been done on the variability (PVT analysis) of

Crosstalk Circuits. This PVT variation analysis establishes the circuit design requirements in terms of coupling capacitances and fan-in limitation that allows reliable operation of the Crosstalk gates under Process, Voltage and Temperature variations. As an example, I also elaborate on the reason for which the full adder can't be implemented as a single gate in the crosstalk circuit-style at lower technology nodes.

Though we designed a variety of basic and complex logic gates and crosstalk polymorphic gates, the biggest question is "Will these crosstalk gates work reliably on silicon owing to their new circuit requirements and technological challenges?". Trying to answer the above question, the whole thesis is mainly focused on the physical implementation of the crosstalk gates at 65nm. I will detail the steps that we have performed while designing the crosstalk circuits and their layouts, the challenges we faced while implementing the new circuit techniques using conventional design approaches and PDK, and their solutions, specifically during layout design and verification.

The other potential application of crosstalk circuits is in non-linear analog circuits: Analog-to-Digital Converter (ADC) [4], Digital-to-Analog Converter (DAC), and Comparator. In this thesis, I have shown how the deterministic charge summation principle that is used in digital crosstalk gates can also be used to implement the non-linear analog circuits.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled "Design and Practical Realization of Polymorphic Crosstalk Circuits Using 65nm TSMC PDK" presented by Bhavana Tejaswini Repalle, candidate for the Master of Science 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.
Chair Department of Computer Science & Electrical Engineering

CONTENTS

ILLUSTRATIONS

Figure Page

# TABLES

Tables Page

# ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Mostafizur Rahman for his continuous motivation and guidance in research and my thesis work. His guidance helped me throughout my master's degree, in writing and reviewing the IEEE papers. He showed me the direction whenever I was stuck in my research and pushed me to find out the reasons for it. He allowed me to audit all his courses which enhanced my knowledge in the field. Moreover, he gave me the freedom to attend all meetings and presentations organized by him from which I have learned a lot.

I am thankful to my fellow lab mate and mentor Naveen Kumar Macha who guided me throughout the Crosstalk work. He gave me the training on the concepts and tools in my earlier days in the VLSI field, also directed me to the topics that I need to learn before getting started on my work. He also gave me guidance on how to express and write my work and present it to my advisor.

I am also grateful to my fellow lab mate and mentor Wafi Danesh for his guidance and support in the Multi-Valued Logic (MVL) research work. He helped me to get acquainted with our research lab and the work and continued his guidance throughout my studies.

I would also like to thank Md Arif Iqbal and Sandeep Reddy Geedipally for their guidance in writing the papers and the knowledge of the tools.

CHAPTER 1

# INTRODUCTION AND MOTIVATION

Device scaling and interconnect bottleneck are among the major challenges for CMOS scaling. Furthermore, signal integrity issues like crosstalk – leakage of charge between capacitively coupled nets among neighboring signal lines are becoming inexorable [1]. Our team proposed the Crosstalk Computing which astutely turns this detrimental effect into an advantage by engineering the interference among signal lines. Crosstalk Technology can potentially solve scaling challenges by reducing device and interconnect scaling requirements while complying with the existing manufacturing paradigm.

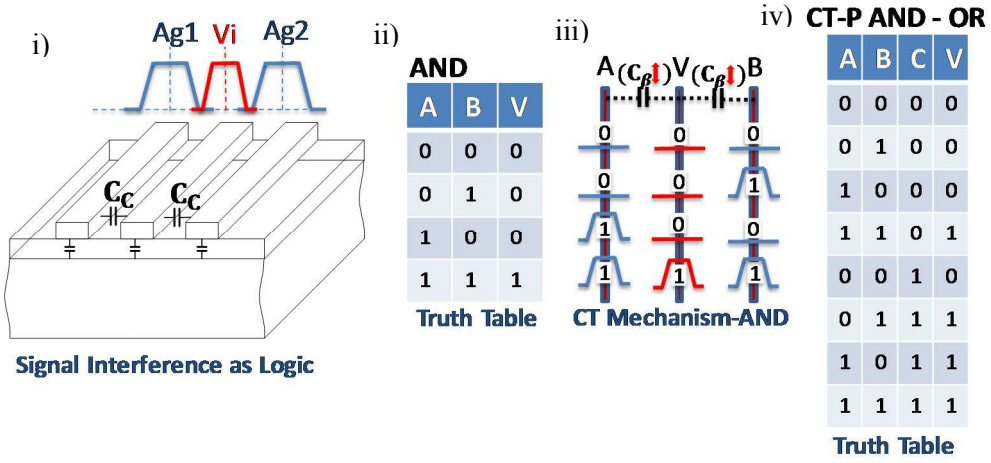


Figure.1.1 Crosstalk Principle

The Fig.1.1(i) 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 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 in nature and possesses the information about signals on two aggressor nets; its magnitude depends upon 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 (which is length times the vertical thickness of metal lines) and inversely proportional to the distance of separation of metal lines. Tuning the coupling capacitance values using the variables mentioned above provides the engineering freedom to tailor the induced summation signal to the specific logic implementation [1]. Fig.1.1 (ii) shows the AND gate truth table where the output of the gate is logic 1 only when both the input signals are logic 1. Fig.1.1 (iii) shows the intuition of the crosstalk AND gate. By incorporating an additional control aggressor, the victim node can be biased to alter the behavior of the AND gate to the OR gate, thus polymorphic gates can be constructed. Fig.1.1 (iv) shows the truth table of polymorphic AND–R gate where the control signal is C. When C=0, the gate should behave as AND gate and when C=1 the gate should behave as the OR gate. This could be enabled by properly engineering the coupling capacitances and the circuit that is specific to Crosstalk Computing.

The actual circuit techniques are discussed next. Fig.1.2 (i) shows the Crosstalk AND circuit in which input aggressor nets (*A* and *B* acting as Ag1 and Ag2) are coupled to victim net (*Vi*) through coupling capacitance $C_C$. A discharge transistor driven by a '*Dis*' signal and an inverter is connected to '*Vi*' net as shown in the figure. The CT-logic operates in two states, logic Evaluation State (ES) and

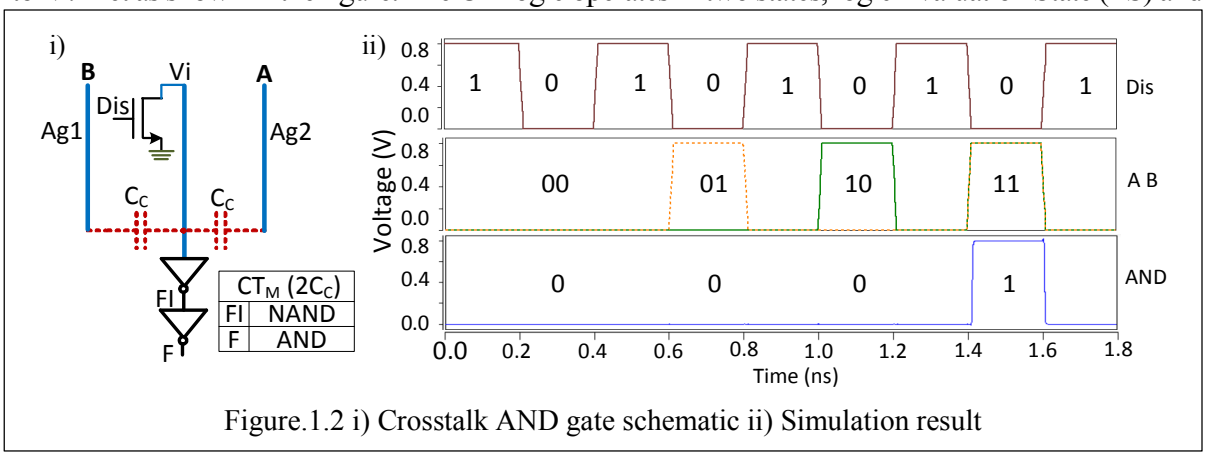


Figure.1.2 i) Crosstalk AND gate schematic ii) Simulation result

discharge state (DS). During ES, the rise transitions on aggressor nets induce a proportional linear summation voltage on *Vi* (through couplings) which is connected to a CMOS inverter. The inverter acts as a threshold function. During the discharge state (enabled by *Dis* signal), the floating victim node is shorted to ground through the discharge transistor, which ensures correct logic operation during the next logic evaluation state by clearing off the value from the previous logic operation. The simulation response of the designed AND gate is shown in Fig.1.2 (ii). The first panel in the figure shows the discharge pulse (Dis), the second panel shows two input signals (*A* and *B*) with 00 to 11 combinations given through successive evaluation stages (when *Dis*=0). The third panel shows the output response of the AND gate. It is to be noted that, as the victim node is discharged to the ground in every DS (*Dis*=1), the outputs of these gates are also logic high.

The rest of the chapters in this thesis are organized as follows: Chapter 2 presents the implementation of large-scale polymorphic circuits and their Power, Performance, and Area (PPA) results. Chapter 3 elaborates on the variability analysis at different processes, voltage, and temperatures (PVT). It also discusses the realization of the high fan-in gates. Chapter 4 presents the physical implementation of the digital crosstalk circuits using 65nm TSMC PDK and its simulation results. Chapter 5 shows difficulties/errors that occurred while designing the crosstalk circuits in silicon using TSMC 65nm PDK. Chapter 6 discusses the nonlinear analog crosstalk circuits and how we can realize them using the charge conservative principle, and its drawbacks. Chapter 7 concludes with the discussion of the benefits of our approach, and what we envision to accomplish in the near future.

CHAPTER 2

POLYMORPHIC CROSSTALK CIRCUIT DESIGN

This chapter introduces the design of the basic polymorphic circuits/gates based on the Crosstalk Computing principle and cascading them to build the large-scale polymorphic circuits.

## 2.1 Basic Polymorphic Crosstalk Gates

The polymorphic logic gates exhibit multiple logic behaviors by virtue of altering a control signal, as a result, it increases the logic expressibility of a circuit. A wide range of polymorphic gates can be implemented using crosstalk circuit techniques, out of which, I show here the circuit reconfigurable AND-OR gates. The technique can be extended to implement many other reconfigurable gates such as OA21-AO21, AND3-AO21, AO21-OR3, etc. All these circuits switch the logic behavior by using an additional control aggressor.

The operation of CT logic gates can be represented functionally using a crosstalk margin function $CT_MI$, which specifies that the inverter of the CT-logic gate flips its state only when victim node sees the input transitions through the total coupling greater than or equal to C. For example, as shown in the Fig.1.1(i), AND CT-margin function is $CT_M(2C_C)$, which states that the inverter flips its state only when the victim node sees the input transitions through total coupling greater

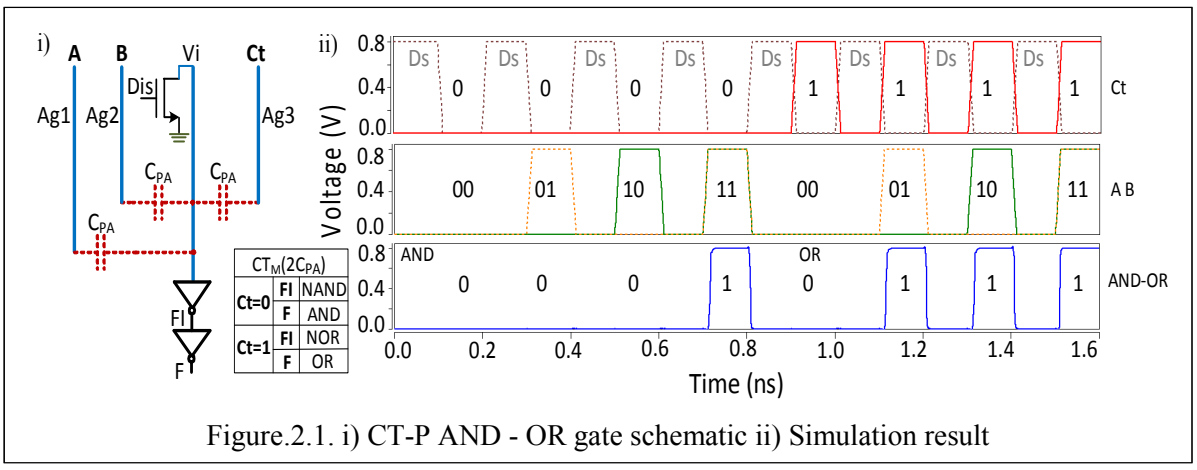


Figure.2.1. i) CT-P AND - OR gate schematic ii) Simulation result

than or equal to $2C_C$, i.e. when both inputs are high. For the CT-margin function of $CT_M(C_C)$, the same gate behaves like an OR gate [3]. These two functionalities can be implemented in a single gate using the control Signal *(Ct)* as shown in Fig. 2.1.

Fig.2.1 shows the Crosstalk Polymorphic (CT-P) AND-OR circuit and its circuit response. As shown in the circuit diagram, inputs (*A* and *B*) and control aggressor (*Ct*) has the same coupling $C_{PA}$. *FI* stage in the circuit (Fig.2.1(i)) gives inverting functions (NAND/NOR) and the *F* stage gives non-inverting functions (AND/OR). The margin function for the AND-OR cell is $CT_M$ ($2C_{PA}$). When control *Ct*=0 it operates as AND, whereas, when *Ct*=1 the *Ct* aggressor *(Ag3)* augments an extra charge through the coupling capacitance $C_{PA}$, hence the cell is now biased to operate as an OR gate. Therefore, the transition of either *A* or *B* is now sufficient to flip the inverter. The same response can be observed

TABLE 1
CROSSTALK LOGIC DESIGN TABLE FOR POLYMORPHIC GATES

| Gate | $C_C$ (fF) | $w_1$ | $w_2$ | $w_3$ | $w_{Ct}$ | Ct | Margin Fuction | Function |
|---|---|---|---|---|---|---|---|---|
| **AND3-OR3** | 1 | 1 | 1 | 1 | 2 | 0 | $CT_M(3C_C)$ | AND3 |
| | | | | | | 1 | $CT_M(C_C)$ | OR3 |
| **AND3-CARRY** | 0.9 | 1 | 1 | 1 | 1 | 0 | $CT_M(3C_C)$ | AND3 |
| | | | | | | 1 | $CT_M(2C_C)$ | CARRY |
| **CARRY-OR3** | 4.5 | 1 | 1 | 1 | 1 | 0 | $CT_M(2C_C)$ | CARRY |
| | | | | | | 1 | $CT_M(C_C)$ | OR3 |
| **OA21-AO21** | 0.7 | 1 | 1 | 2 | 1 | 0 | $CT_M(3C_C)$ | OA21 |
| | | | | | | 1 | $CT_M(2C_C)$ | AO21 |
| **AND3-AO21** | 0.28 | 1 | 1 | 2 | 2 | 0 | $CT_M(4C_C)$ | AND3 |
| | | | | | | 1 | $CT_M(2C_C)$ | AO21 |
| **AND3-OA21** | 0.21 | 1 | 1 | 2 | 1 | 0 | $CT_M(4C_C)$ | AND3 |
| | | | | | | 1 | $CT_M(3C_C)$ | OA21 |
| **OA21-OR3** | 0.97 | 1 | 1 | 2 | 2 | 0 | $CT_M(3C_C)$ | OA21 |
| | | | | | | 1 | $CT_M(1C_C)$ | OR3 |
| **AO21-OR3** | 3 | 1 | 1 | 2 | 1 | 0 | $CT_M(2C_C)$ | OA21 |
| | | | | | | 1 | $CT_M(1C_C)$ | OR3 |
| **CARRY-AO21** | 2.2 | 2 | 2 | 3 | 1 | 0 | $CT_M(4C_C)$ | CARRY |
| | | | | | | 1 | $CT_M(3C_C)$ | AO21 |
| **OA21-CARRY** | 0.6 | 2 | 2 | 3 | 1 | 0 | $CT_M(5C_C)$ | OA21 |
| | | | | | | 1 | $CT_M(4C_C)$ | CARRY |

in the simulation plots shown in Fig.2.1(i).

The first panel shows the discharge (*Dis*) and control ($C_t$) signals, the second panel shows the input combinations fed through *A* and *B*, and 3rd panel shows the response at stage F. It can be observed that the circuit responds as AND when *Ct*=0 for first four input combinations (00 to 11), whereas, it responds as OR when *Ct*=1 during next four input combinations (00 to 11). The polymorphism is shown between many logic functions[3]. Based on coupling capacitance requirements to implement a given logic function, we categorize the logics in Crosstalk Computing into two types. A homogeneous logic if all the aggressors need equal coupling capacitances, and heterogeneous logic if aggressors need unequal coupling capacitances. The polymorphism can be achieved between homogeneous to homogeneous logic: AND-OR, AND-CARRY, OR-CARRY; heterogeneous to heterogeneous logic: AO21-OA21; and homogeneous to heterogeneous logic: AO21-AND3, AO21-OR3, AO21-CARRY, OA21-AND3, OA21-OR3, OA21-CARRY, etc. Table.1 presents the crosstalk logic design table for CT polymorphic gates. The margin functions, as listed in the table, transform from one functionality to others when Ct transitions from 0 to 1 and vice-versa[5].

## 2.2 Large – Scale Polymorphic Crosstalk Circuits

This section shows the design of Multiplier-Sorter Crosstalk Polymorphic Circuits and Multiplier-Sorter-Adder Polymorphic Crosstalk Circuits.

### *2.2.1 Multiplier-Sorter Polymorphic Crosstalk Circuit*

To show the potential of CT polymorphic logic gates, an example circuit of a 2-bit multiplier-sorter (Fig.2.2) is implemented using the above crosstalk polymorphic crosstalk gates. The circuit uses 19 gates in total, 16 CT gates, and 3 inverters. 8 out of 16 CT gates are CT polymorphic gates. Polymorphic gates are efficiently employed to switch between the multiplier and sorter

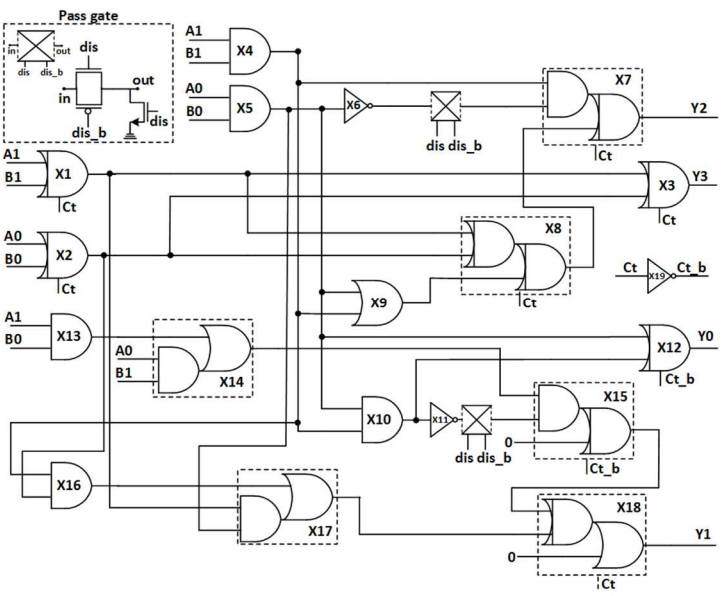


Figure. 2.2 Crosstalk Multiplier – Sorter Circuit Diagram

operations. A control signal *(Ct)* is used switch between this the operations, *Ct=0* is a multiplier and *Ct=1* is Sorter. Fig.2.3 shows the simulation response of the circuit, where different operation modes of the circuit are annotated on top, which is, Discharge State (DS), Multiplier (M) and Sorter (S). The first panel in the figure shows Dis and Ct signals, second and third panels show the two 2-bit inputs A[1:0] and B[1:0], and the following panels show the 4-bit response of the circuit Y[3:0]. To depict multiplier and sorter operations effectively, the Ct signal is given as 0 and 1 alternately which makes the circuit operate as multiplier and Sorter in successive logic states. Also, common inputs, A[1:0] and B[1:0] are given for adjacent M and S modes. It can be observed from the response graphs (Y[3:0]) that, for the same inputs, the circuit gives the multiplier result when Ct=0 and sorter result when Ct=1. For example, for the first set of input combinations, 10 and 11, the M operation gives 0110 as output and S operation gives 1110. Similarly, for the second set of inputs, 01 and 01, M operation gives 0001 and S operation gives 1100. Similarly, M and S outputs are shown for a few other combinations. The circuit consumes only 88 transistors in total. Thus CT-P circuits are compact, possess maximum

reconfigurable features, and can efficiently implement larger polymorphic circuits in cascaded topology.

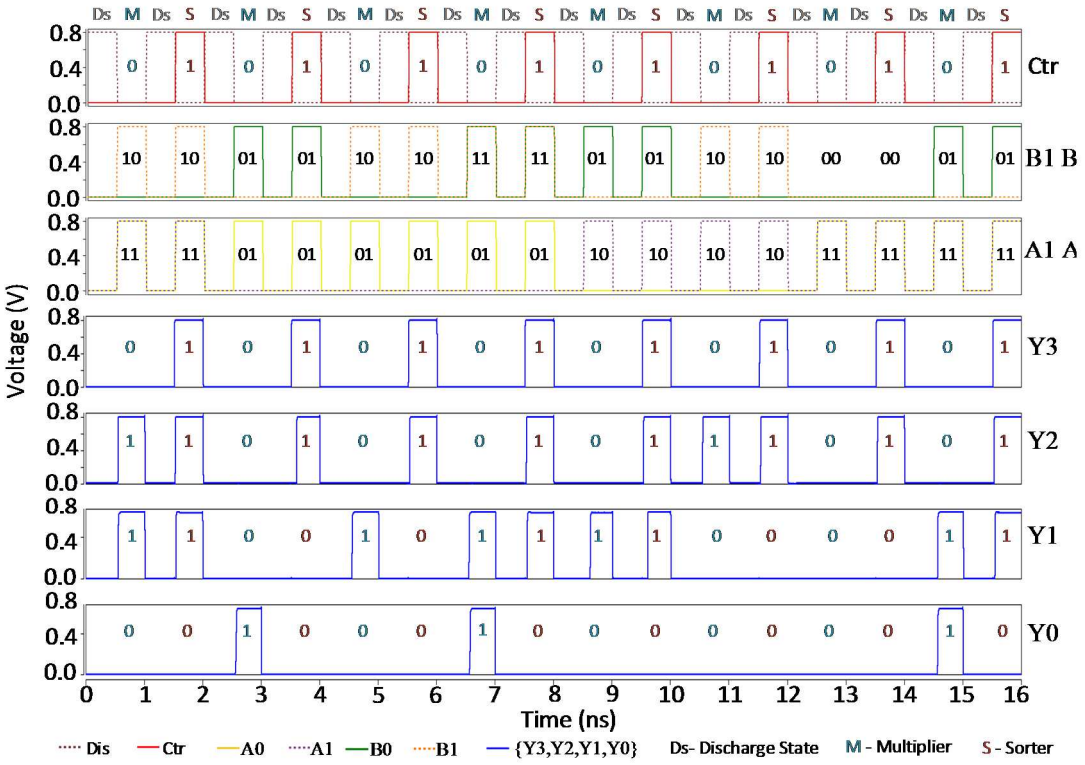


Figure. 2.3 Crosstalk Multiplier – Sorter Circuit Response

*2.2.2 Multiplier-Sorter-Adder Polymorphic Crosstalk Circuit*

This section demonstrates the block level polymorphism using a circuit example of a 2-bit multiplier-sorter-adder (Fig.2.4) which is implemented using the polymorphic gates discussed above. 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 that are efficiently employed to switch the circuit between the multiplier, sorter and adder operations using the two control signals *C1* and *C2*. The inset figure shows the control circuitry (*C1*-*C5*). Fig.2.5 shows the simulation response of the circuit, where different operation modes of the circuit are annotated on top, which are,

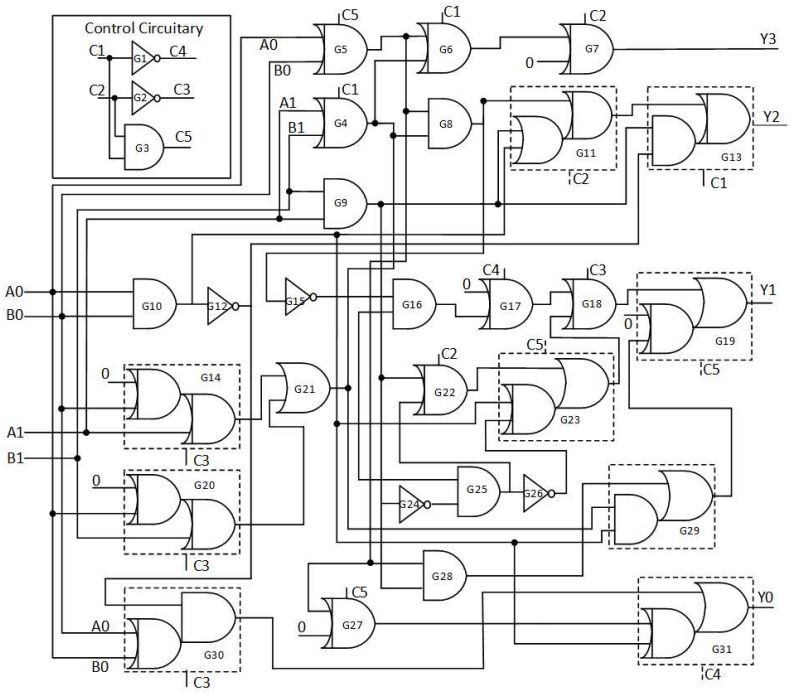


Fig.2.4. Crosstalk Polymorphic Multiplier/Adder/Sorter circuit

Multiplier (M), Sorter (S), and Adder (A). The first panel shows *Dis* signal, *Dis=1* is the discharge state (DS) and *Dis*=0 is the logic evaluation 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. Third and fourth panels show the 2-bit inputs *A[1:0]* and *B[1:0],* respectively. The following panels show the 4-bit response of the circuit, *Y[3:0]*. The circuit is operated alternately in the multiplier, sorter, and adder modes. In each set of these modes, common input values are fed through *A1A0* and *B1B0* which effectively demonstrates the transformation of the circuit in accordance with the control signals. For example, for the first set of 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 set fo inputs, 10 and 01, M, S, and A operations

give 0010, 1100 and 0011 outputs, respectively. In a similar fashion, few other combinations are shown in the next stages. The circuit consumes only 155 transistors in total. Such polymorphic circuits can be employed for fault tolerance at the block level.

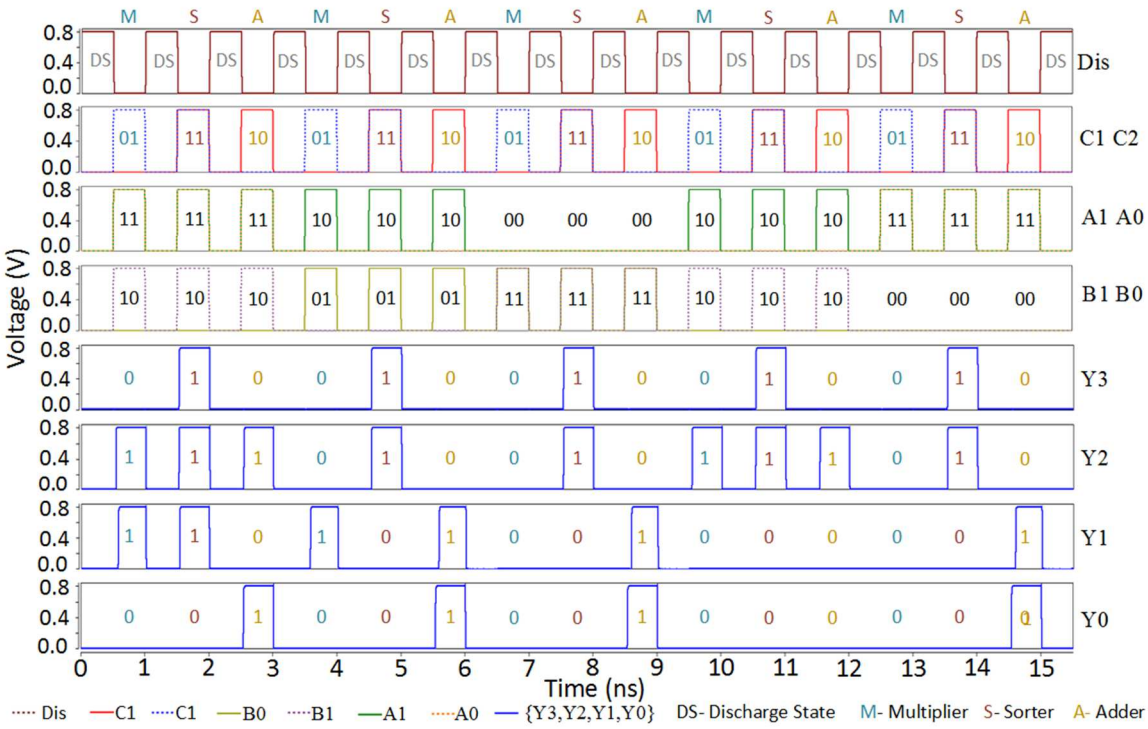


Fig.2.5. Crosstalk Polymorphic Multiplier/Adder/Sorter circuit simulation response

For example, as shown in Fig.2.6, Multiplier, Sorter and Adder operations can be implemented as independent blocks, which also possess the dormant other two operations. During the event of fault detection in one of the blocks, the other blocks can be reconfigured and multiplexed to achieve the correct output. The polymorphic blocks can be also used with traditional voter based [2] fault resiliency techniques.

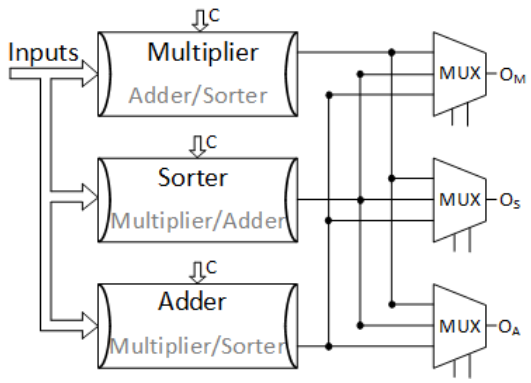

Fig.2.6. Block-Level Polymorphic Fault-tolerant Scheme

# CHAPTER 3

# PRACTICAL REALIZATION OF CROSSTALK CIRCUITS

This chapter discusses, the practical realization of the crosstalk circuits using TSMC 65nm PDK. Because of the novel layout requirements for Crosstalk circuits, I have followed the custom circuit design methodology, involving the following steps, (i) schematic design, (ii) Symbol design, (iii) Functional verification of circuit schematics, (iv) layout design, and (v) layout verification. The following sections describe work in detail, including the tools setup and step by step followed.

## 3.1 Calibre and TSMC libraries setup:

The invocation of Cadence Virtuoso integrating the Mentor Graphics Calibre tool can be done using a single file. Create tsmc_65nm.csh file in the home directory and copy the below lines. Work_65nm is the directory where we installed the TSMC libraries with the help of the README file that TSMC provided. To start the Cadence virtuoso, use the command "source tsmc_65nm.csh".

```
cd /home/students/brkf9/work_65nm
module load MentorGraphics/calibre/current
module switch Cadence/615/Cadence-615 Cadence/new
export CDS_Netlisting_Mode=Analog
setenv CDS_Netlisting_Mode Analog
module load PyCellStudio/current
virtuoso &
```

## 3.2 Physical Implementation of Crosstalk Polymorphic Circuits

We have prototyped our chip using TSMC 65nm logic PDK. The purpose of the current chip prototype is to demonstrate the functionality of the Crosstalk Circuits for proof-of-concept. Though our research envisions for Crosstalk Computing specific 3-D capacitive structures for efficient

implementation, the conventional foundry PDKs lack these features. But, to serve our proof-of-concept purpose, we have chosen the existing capacitance elements in TSMC PDK for our circuit implementations. The capacitance options present in TSMC 65nm PDK are MOSCAP and MIMCAP. We have implemented the crosstalk circuits using both the capacitance types.

Figure 3.1 shows the custom circuit design followed. The first step to be done to design any Crosstalk circuit is to observe the inverter DC characteristics, find out the inverter trip-point/threshold, and noise margins. The above simulation steps need to be done at various variation corners, i.e., Process (P), Temperature (T), Voltage (V). The threshold voltage and noise margins provide a metastable region of the inverter that needs to be avoided while operating the circuits. Then we calculate the input capacitance of the CMOS inverter. We then compute the capacitances values from the voltage division equations formulated for the Capacitive network at the input of inverter *(Vi-node)* for Crosstalk Gates [2]. These capacitance values serve as the starting point for our design. Because of the non-linear nature of the Gate capacitances of CMOS transistor and parasitic RCs, we would require to correct the coupling capacitances by observing the simulation response. The optimal circuit response is achieved for all the gates in a few design iterations. We have used Cadence Virtuoso Schematic Editor for circuit design, and Cadence Spectre Engine for circuit simulations. We then design the layouts for the schematics of the final circuit, perform the RC parasitic extraction, simulate extracted circuits and verify the functionality. The functional verification at this stage is also performed considering the PVT variations. If the added parasitics and/or PVT variations disturb the circuit functionality, we might need to perform design iteration and correct the coupling capacitances to fix the functional failures. We observed that this might happen only for high fan-in/complex gates. Then, we perform the physical design verification steps (DRC, LVS, and Antenna Checks) on these custom

circuits. The final circuit blocks are then integrated at the full-chip level which will be shown in the following sections. I would discuss next the details of the above design steps, challenges and their solutions adopted, including the tool flows.

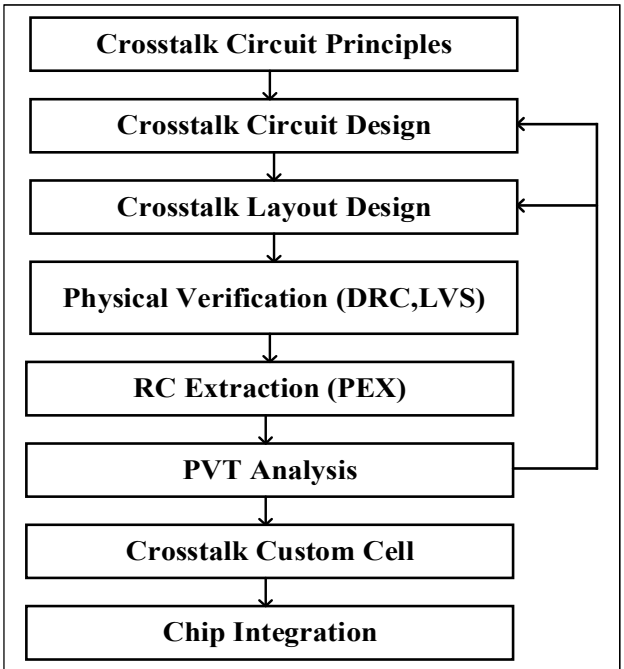

Fig.3.1 Custom Layout Design Flow

3.3 Observations after installing the PDK

We need to observe the characteristics of the PMOS and NMOS transistors and determine its threshold voltage, and input gate capacitance. By knowing these values, we can determine the coupling capacitance values and the inverters sizing ratios required. Observing the on-current, and off-current would also serve as metrics for estimating performance and leakage power. Fig.3.2. shows the experiment setup to extract the transistor characteristics and Fig.3.3 shows the DC transfer characteristic curves.

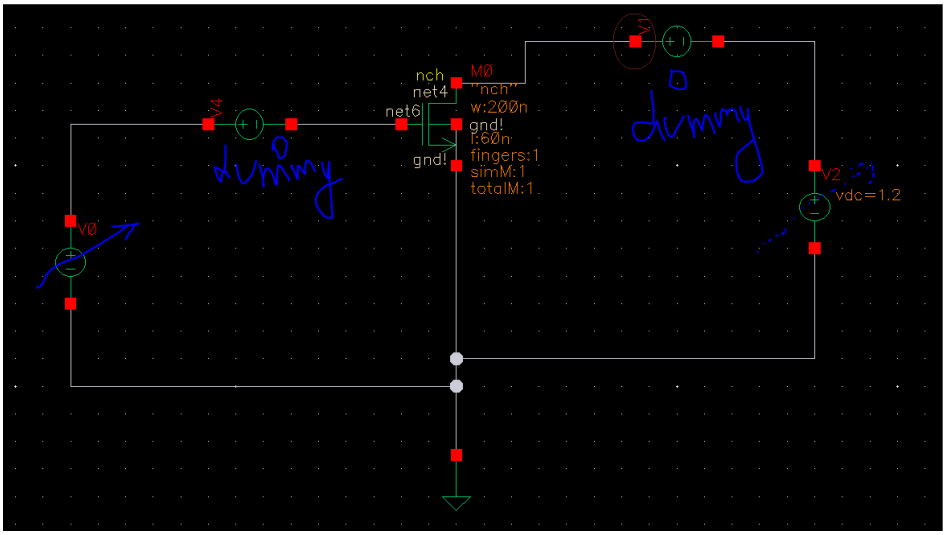


Fig.3.2 DC Sweep of a NMOS transistor

Perform check and save to see if there are any errors in the schematic diagram. For the $V_{GS}$ vs $I_{DS}$ characteristics curve, give the drain to source voltage as 1 volt. If there are no errors, then proceed for the DC simulation. Open the ADE – L from the schematic window and choose the analyses as DC. In the component name section, click on the select component and select the DC voltage at the Gate input and in the sweep range, select 0 volts as the start and 1 volt

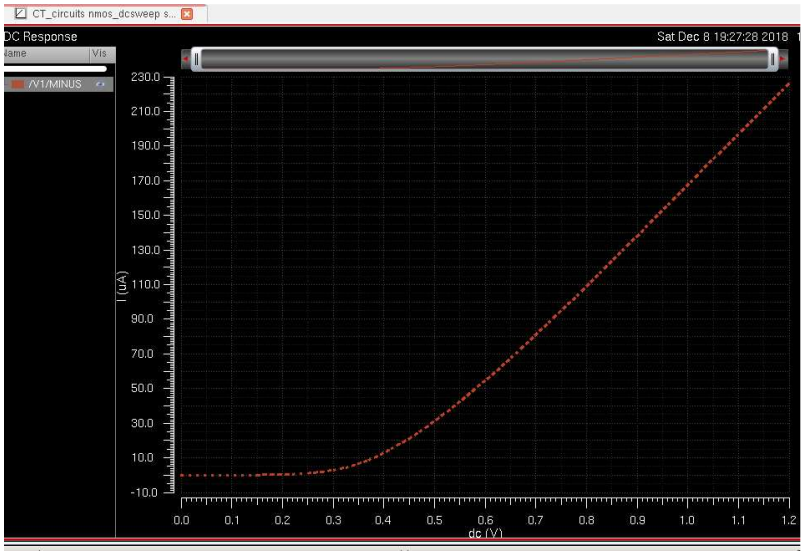


Fig.3.3 $I_{DS}$ vs $V_{GS}$ characteristics curve

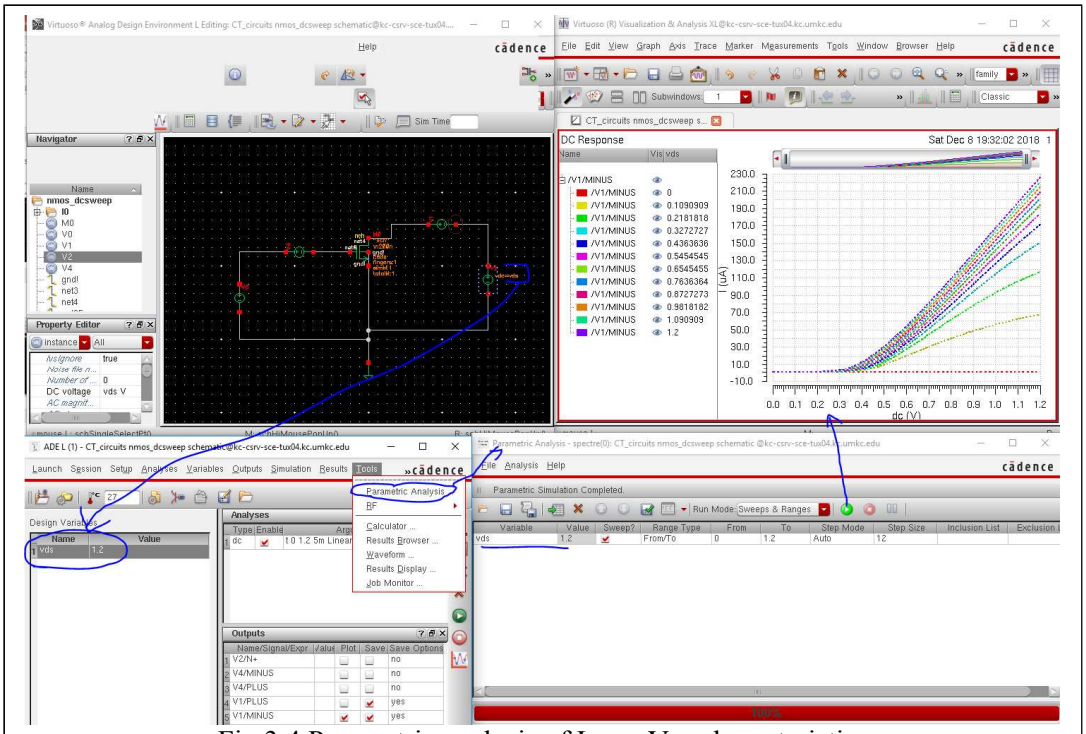

Fig.3.4 Parametric analysis of $I_{DS}$ vs $V_{GS}$ characteristics

as the stop values. In the output's session, select the drain port which indicates that the current flowing through the gate terminal. Now click the play (green in color) button to run the simulation and the result is shown in Fig.3.3. To observe the same plot in Fig.3.3 for different

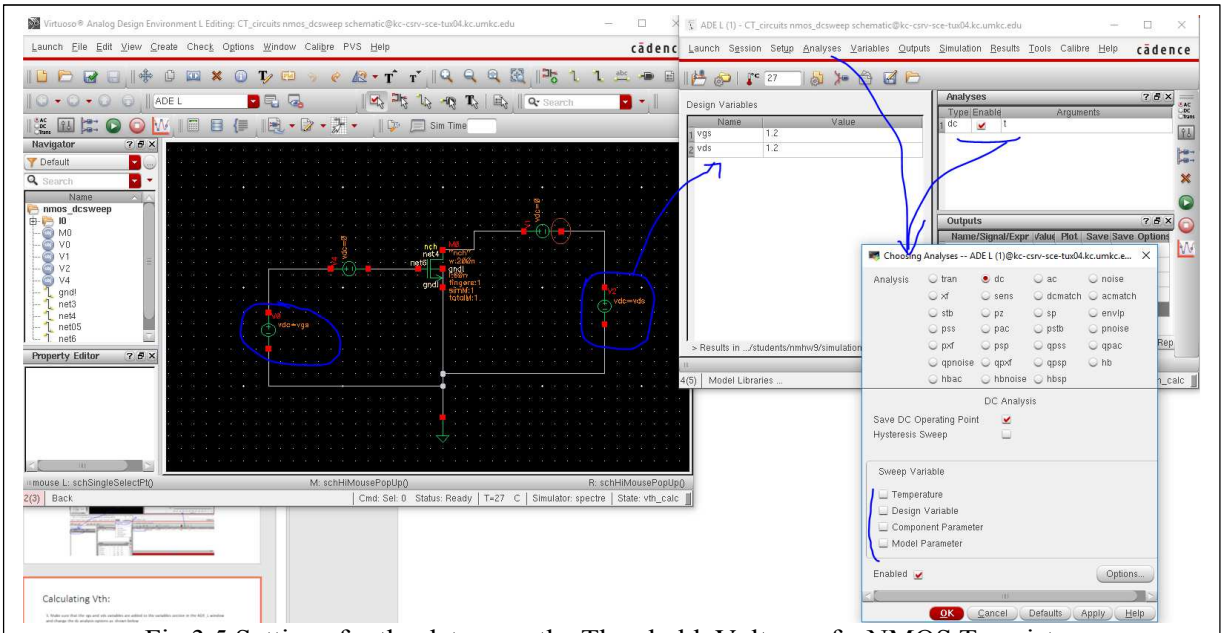

Fig.3.5 Settings for the determne the Threshold Voltage of a NMOS Transistor

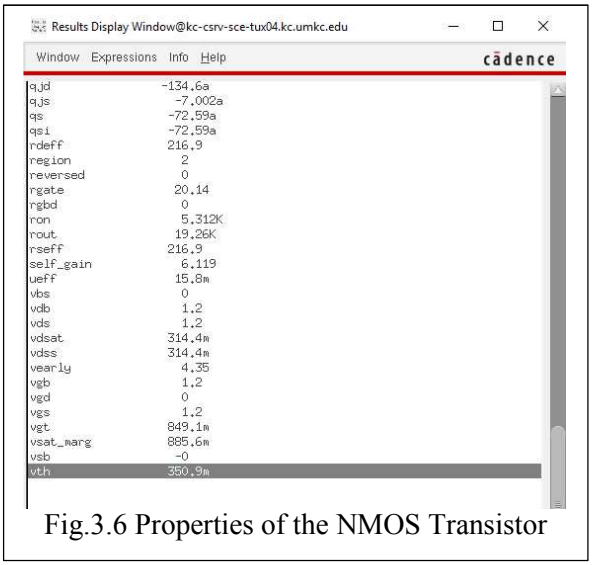


Fig.3.6 Properties of the NMOS Transistor

$V_{DS}$ values, select the $V_{ds}$ as the design variable and follow the process as shown in Fig.3.4 To determine the threshold voltage of a device, make sure that the $V_{ds}$ and $V_{gs}$ are added to the design variables and follow the process as shown in Fig.3.5. After the simulation completes successfully, go to the "results" in the ADE-L window, then "print", then click on the DC Operating points. When we click on it, a window will pop-up which shows all the parameters related to the NMOS transistor as shown in Fig.3.6. In the properties window, we

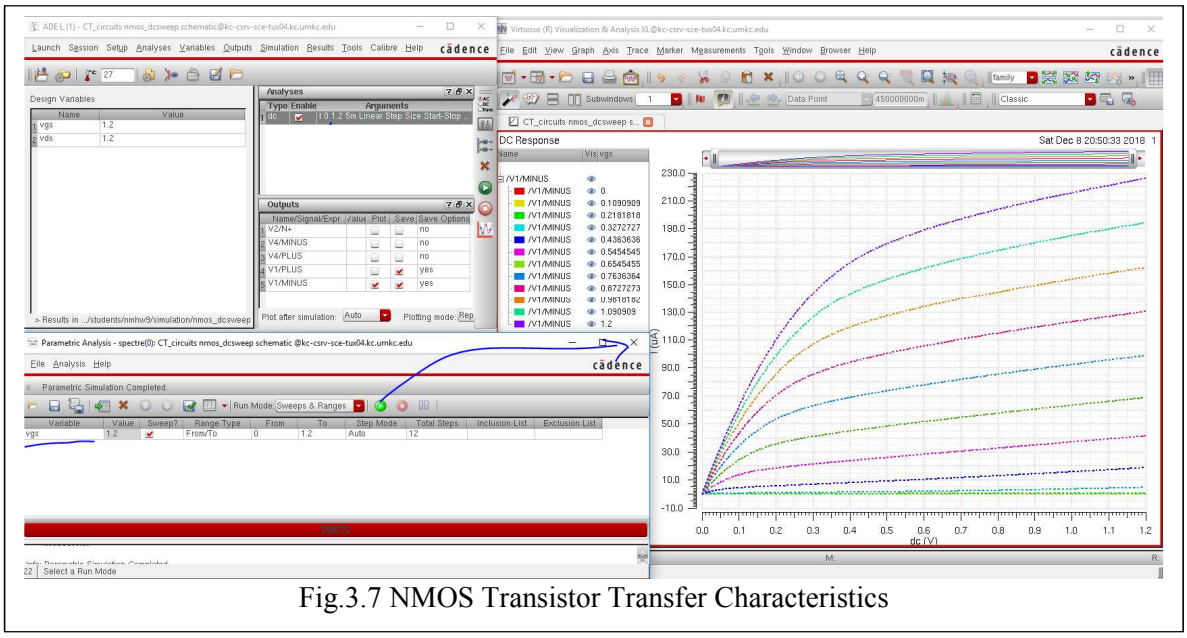


Fig.3.7 NMOS Transistor Transfer Characteristics

can find the threshold voltage as 350.9mV. From Fig.3.3 the average off-current is 206.254nA and the on-current is 225.895uA when the gate and drain voltages are at its maximum. We can also observe that the square law is not followed by the transistors in this 65nm TSMC PDK from the $V_{DS}$ vs $I_D$ curve as shown in the Fig.3.7.

TABLE 2
TSMC 65nm PMOS and NMOS transistor properties

| 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 |

Table.3.1 shows the summary of TSMC 65nm PMOS and NMOS transistor properties. The threshold voltage of the PMOS and NMOS are required to design the threshold device of the crosstalk circuits which is the CMOS Inverter. To choose the coupling capacitances for the Crosstalk

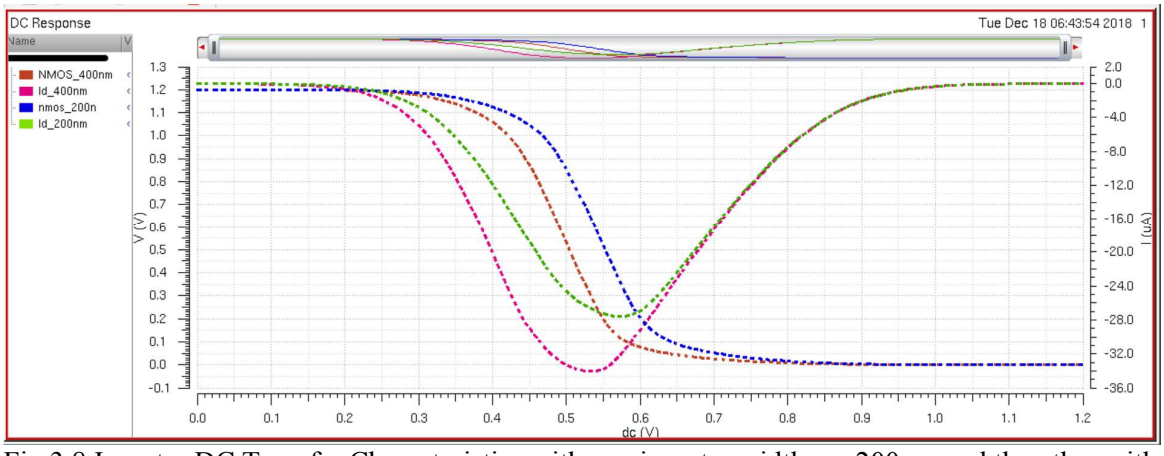

Fig.3.8 Inverter DC Transfer Characteristics with one inverter widths as 200nm and the other with 400nm along with it's drain currents

gates, we would need to find out the input gate capacitance of the CMOS inverter. The voltage induced on the victim node with different logic combinations/transitions on the input aggressors can be calculated using capacitance division equations [16]. By playing with the crosstalk coupling capacitance values, the voltage induced on the victim node can be tailored to implement logic behavior. Thus, the coupling capacitances serve as the first independent variables that can be engineered/controlled to design various logic gates. The second independent variable that can be helpful in designing various crosstalk gates is the threshold-voltage/trip-point of CMOS inverter. For example, Fig.3.8 shows the DC transfer characteristic curve of a CMOS Inverter for NMOS widths as 200nm and 400nm. We can observe that the characteristic curve moved left by doubling the NMOS size. Thus, the trip point of the CMOS inverter (which provides threshold function) can be increased or decreased by different sizing ratios for PMOS and NMOS transistors. Therefore, we use both coupling capacitance and trip-point of the inverter as the design variables that enable building various crosstalk logic gates.

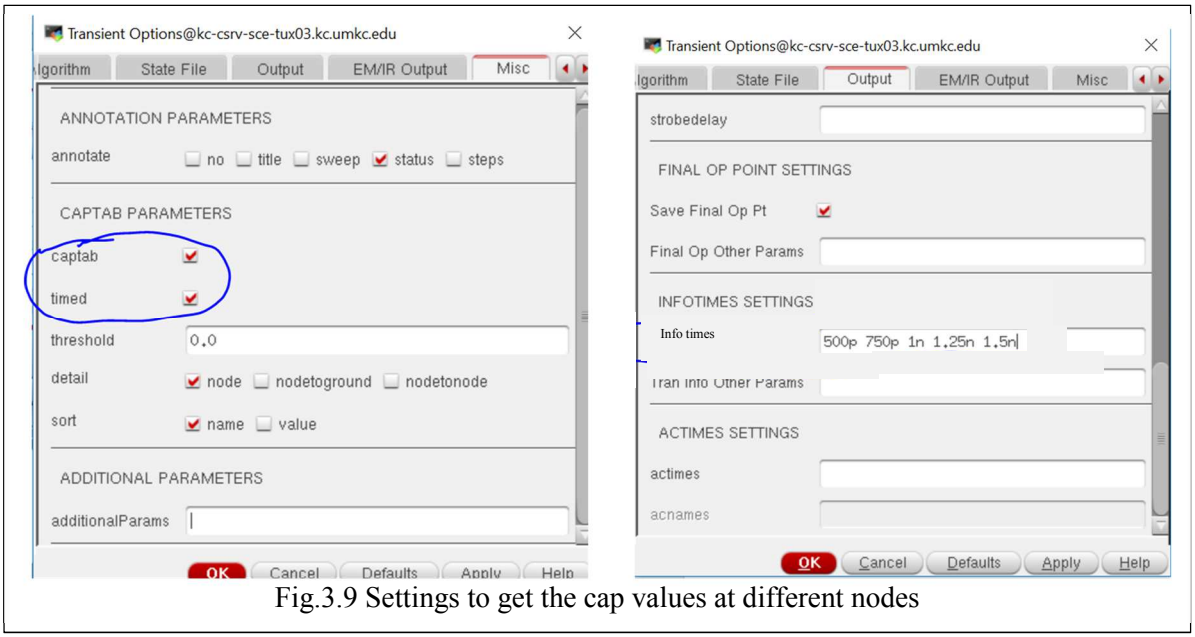

Fig.3.9 Settings to get the cap values at different nodes

Fig.3.9 shows the settings to do to get the cap values at different nodes. In the ADE – L window, go to transient analyses then options then Misc and check the options cap-tab and timed column. Also, in the output tab, give the info times as the time points at which we are interested to find the cap. To print the cap values, go to Results → Print → Cap Table as shown in Fig.3.10. The analysis would give the capacitances values at different info times, which are 314.362af, 322.909af, 322.909af, 343.372af and 314.276af, also, the average value, which is 323.565af.

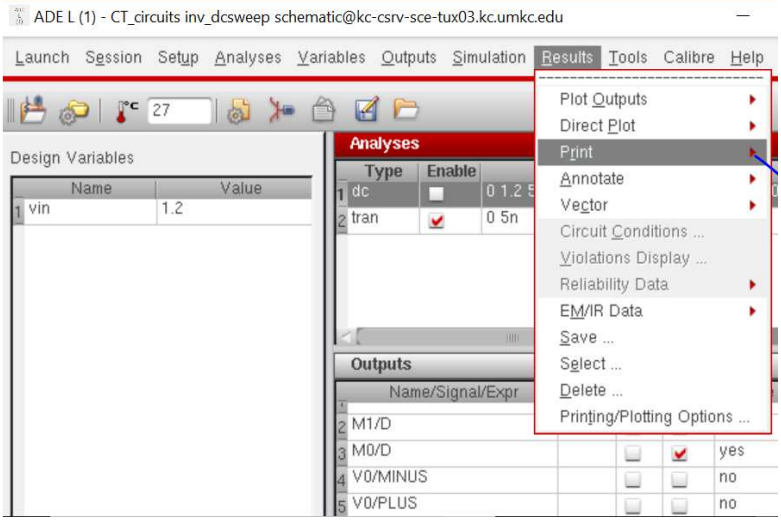

Fig.3.10 Settings to print the cap values

TSMC Provides three types of capacitances. They are MOSCAP (Metal Oxide Semiconductor Capacitor), MOMCAP (Metal-Oxide-Metal Capacitor) and MIMCAP (Metal-Insulator-Metal Capacitor). We get MIM and MOSCAP in the PDK. The MIM cap offers larger capacitance in a small footprint compared to the MOS CAP. So, we used MIM cap, whenever we require the very large capacitance, instead of using a very large width poly MOSCAP. Next few sections I would discuss the custom circuit design flow on Cadence Virtuoso tools.

3.5 Schematic and Symbol Design

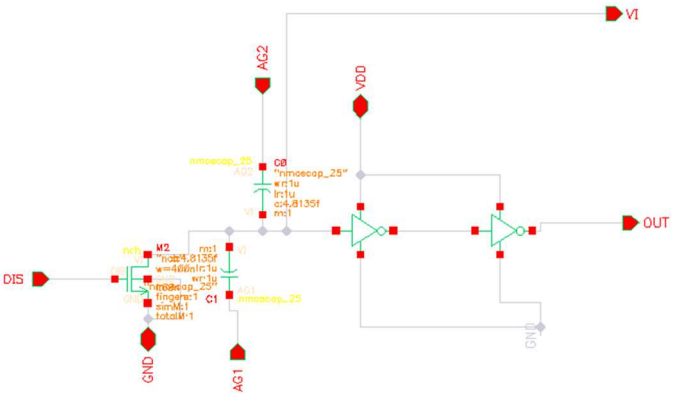


Figure.3.11 AND gate Schematic Diagram

After choosing the coupling capacitances and the inverter sizes, open Cadence Virtuoso Schematic Editor and instantiate the transistors, capacitors and the voltage sources and connect them using wires as shown in Fig.3.11. The inverters used in the schematic are from TSMC 65nm Standard Cell Library. However, we might need to customize the transistor sizes and design our own CMOS inverters to meet the Crosstalk logic requirements. There are three ways to simulate the schematic and verify the functionality. One is using a digital vector file, second is by instantiating the voltage pulse

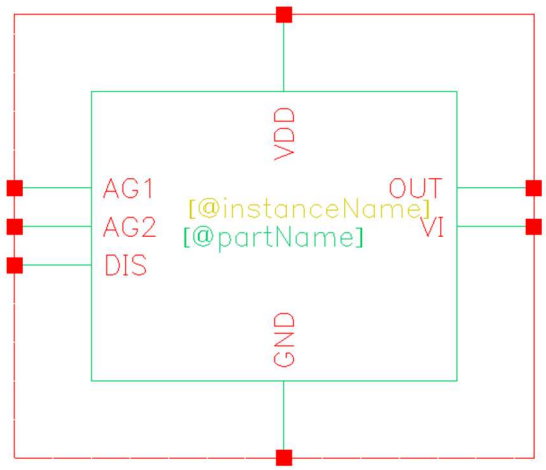


Figure. 3.12 AND gate Symbol

source in the above schematic, and the third way is creating a symbol and then using either digital vector file or simulation sources. We used the third approach. The same setup can be used for the layout simulation also. Fig.3.12 shows the symbol of the schematic shown in Fig.3.11.

It is to be noted that the VI pin is probing the internal victim node of the gate. Observing the *Vi* node voltage would help us both in the design and debug stage, also for extracted layout. Fig.3.13 shows the Design Under Test (DUT) schematic for the Crosstalk AND gate, using its symbol view.

3.6 ADE – XL Schematic Simulations in different PVT corners

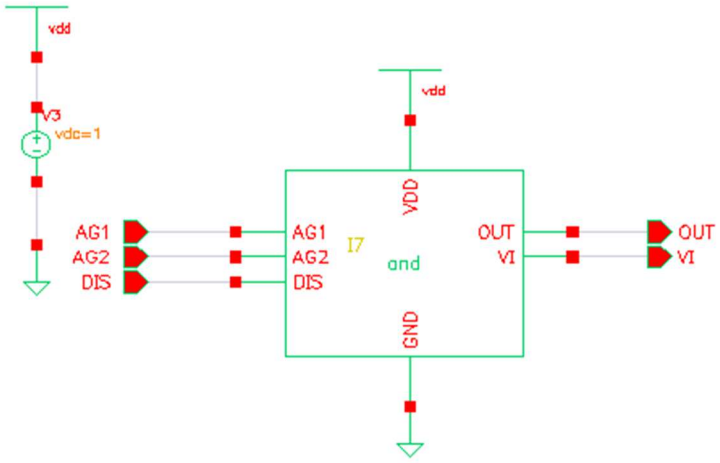


Figure. 3.13 CT AND gate test Schematic

Next save DUT by clicking on the check and save option to see if there are any errors present in the schematic, symbol and test schematic. ADE-L can be used to simulate the schematic and layout at only one corner at a time. To simulate at various corners using ADE-L, we need to manually run each corner. However, we can use ADE-XL to automatically run the simulation in all the corners in one go.

Also, we need to select the appropriate model files for transistors and passive elements. To manually, simulate at a particular corner, we can select different model files. The simulations can be

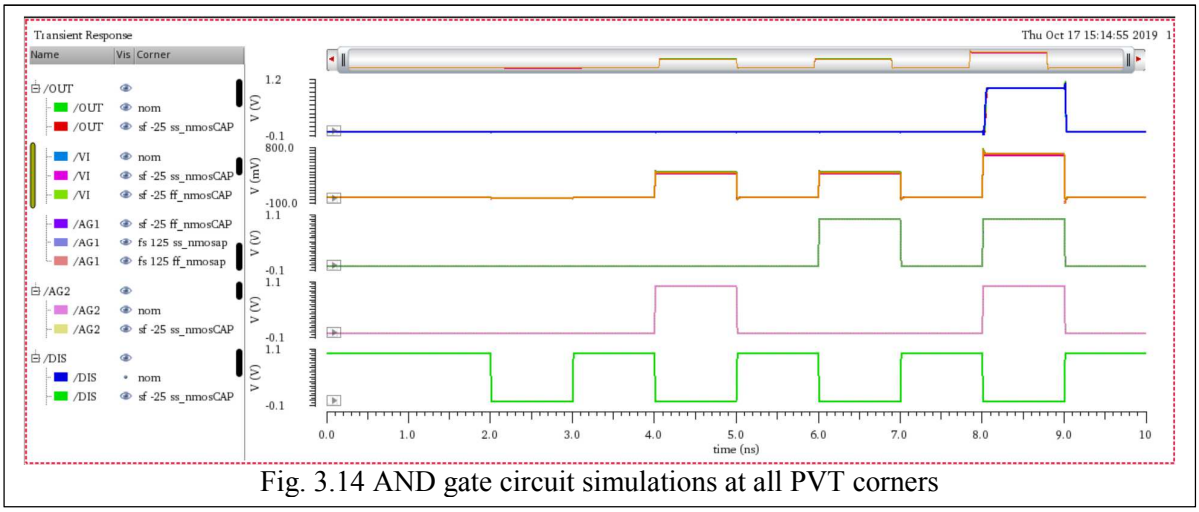


Fig. 3.14 AND gate circuit simulations at all PVT corners

performed on various views using the same DUT setup. To perform simulations on different views, i.e., schematic, caliber extracted, etc., we need to change the view list appropriately in Environment options.

Before starting the simulation, we can select to probe any node that we are interested in the circuit by using using "Outputs → To Be Plotted → Select on the Design". Then select the interested node and run the simulation. After the successful simulation, we can also save the state using "session → Save State" and give a name and folder to save it. This saved state can be used in ADE-XL window later on. After performing all the above simulation setups, Fig.3.14 shows the Crosstalk AND gate schematic simulation response. Next, I will discuss the layout design procedure for this circuit.

## 3.7 Crosstalk Gates Layout Design

Before starting the layout drawing, we need to set up the grid size which can be found in the PDK documents. For this 65nm PDK, the grid size is 0.005 um. We can change these grid settings in "options" → "display". In the Display Options window as shown in Fig.3.15, change the X snap spacing and Y snap spacing from 0.003 (default) to the grid mentioned in the PDK which is 0.005. Also, change the minor spacing to 0.01 and the major spacing to 0.1. Note that we need to set the snap

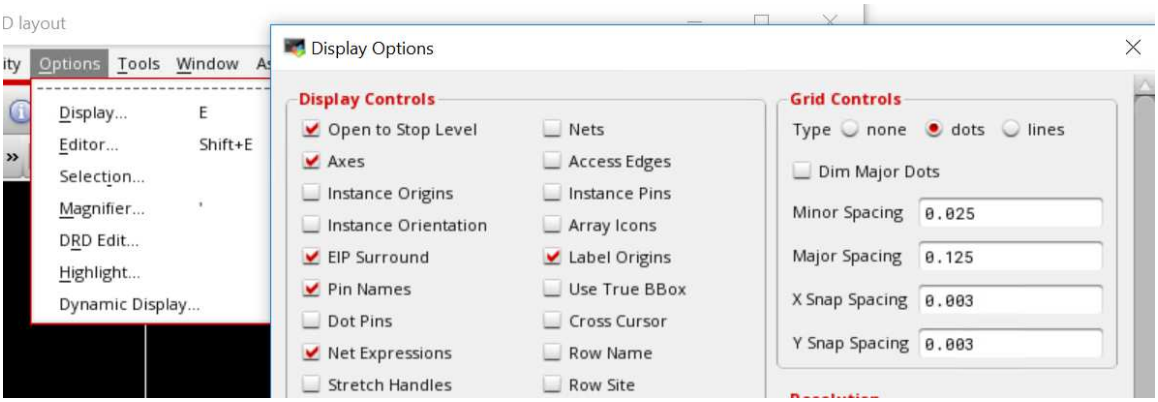


Fig. 3.15 Setting the grid as per the technology node

spacing for each time we open the layout window, otherwise it will create grid DRC errors and we might need to redo the whole design.

After setting the grid, we can start placing the devices and connect them using metal lines. We can practice the good layout design procedure of checking DRC errors for every few steps when we draw the layout. Fig.3.16 shows the final AND gate layout in which two NMOSCAPs value is 4.8fF and the inverter size is 4x. This inverter was taken from the 65nm TSMC standard cell library.

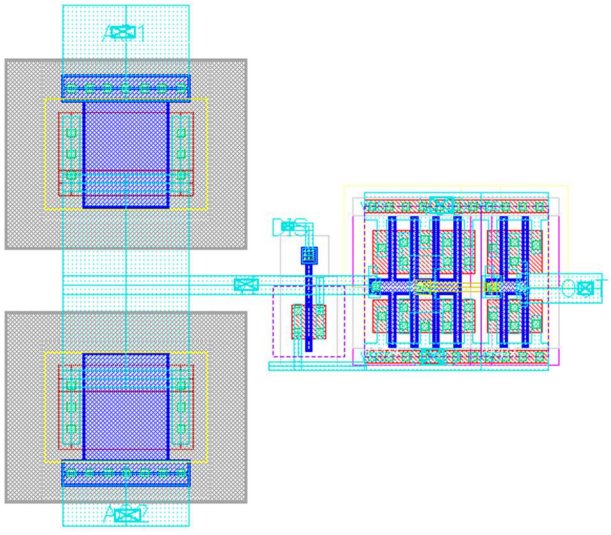

Figure. 3.16 CT AND gate Layout

3.8 Physical Verification and Extraction of the layouts

Physical verification was done using the MentorGraphics caliber tool. To integrate the caliber tool to virtuoso use “module load MentorGraphics/calibre/current”. We can see the caliber plug-ins when we instantiate the layout tool as shown in the Fig.3.17.

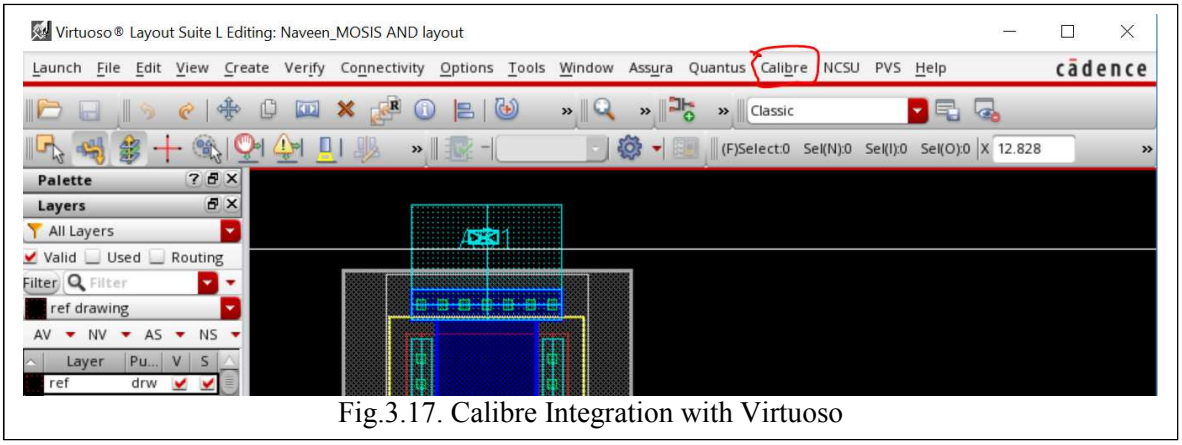


Fig.3.17. Calibre Integration with Virtuoso

The first step that we need to check after the layout design is DRC checking. Foundry will provide the DRC file which contains the various design rules such as minimum width and minimum spacing for all the layers.

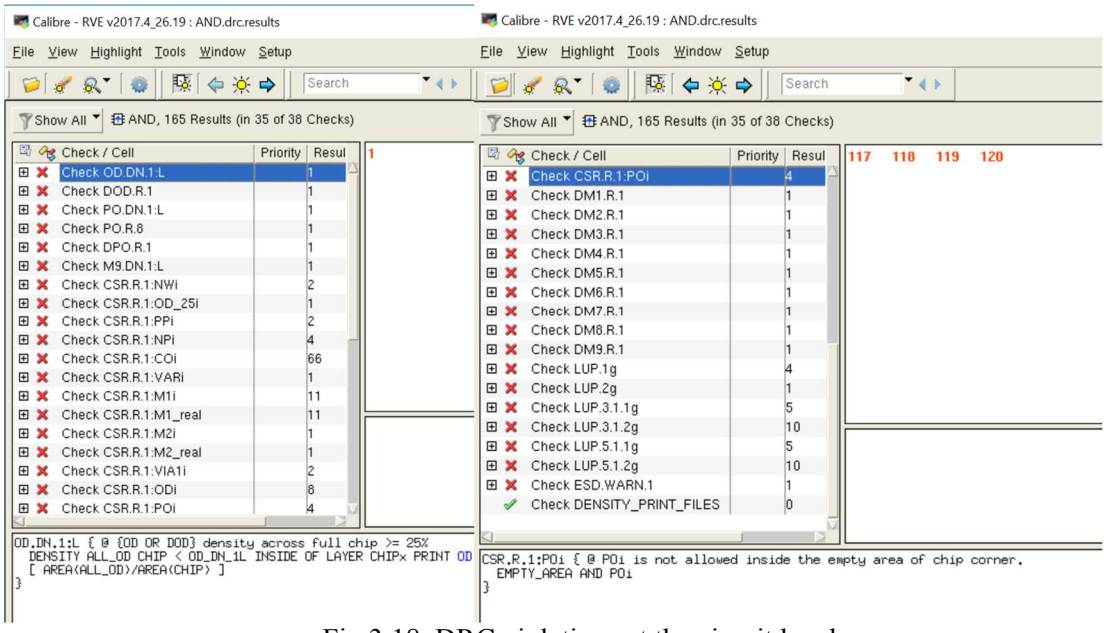


Fig.3.18. DRC violations at the circuit level

When the caliber Interactive nmDRC window opens up, we provide the DRC rules deck and every other field in the input and output section will automatically be populated with its appropriate values. Then we can click on Run DRC. After the DRC check is completed, a results window will pop up with all the DRC violations in the design as shown in the Fig.3.18. We can ignore the full-chip level DRC violations at this stage. They can be addressed at the full-chip level. For example, in Fig.3.18, the violations which have *DN* pattern relate to the density rules for each layer. We can fix these violations after filling the dummy metal-fills at the full-chip level.

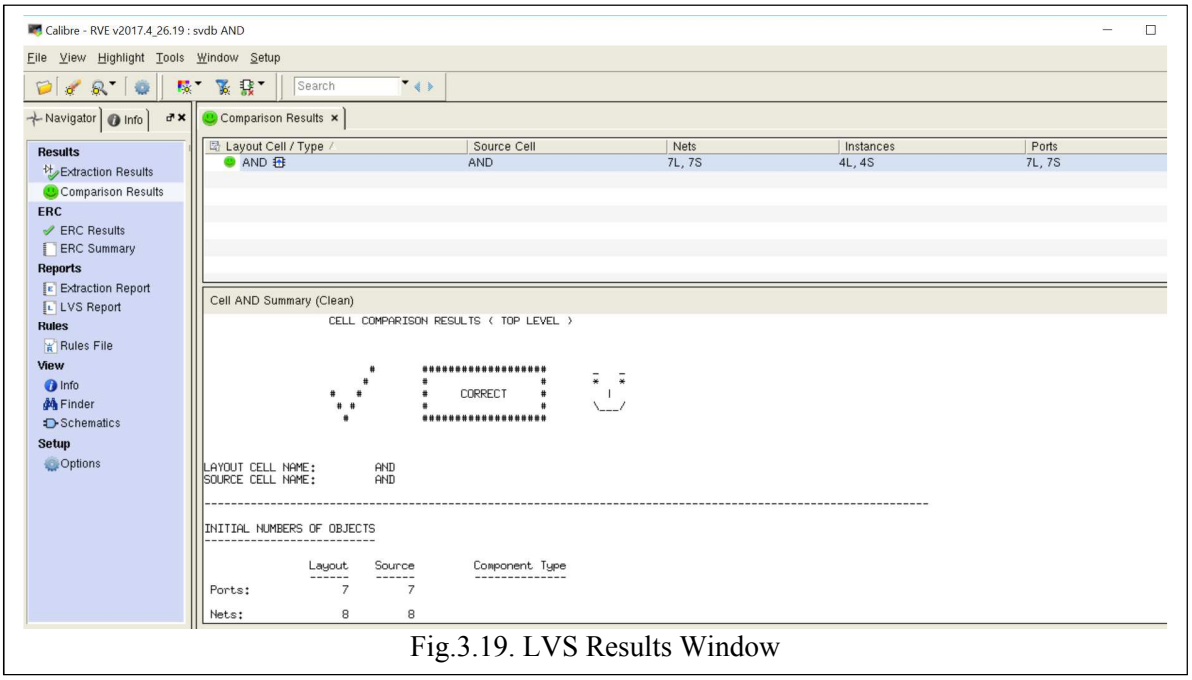

Fig.3.19. LVS Results Window

Similarly, the violations which start with D* pattern, followed by the layer name indicate that the dummy layers related to the drawing layers are missing in the design. These dummy layers are placed in the full chip. We would encounter one more violation "PO.R.8 – floating gate error", which is associated with the discharge transistor gate. This error will be eliminated when the gate of these transistors is connected to the I/O pads in the full chip. Other errors that start with the CSR* pattern are related to the I/O pad corners that can also be ignored at this stage. We can ignore the

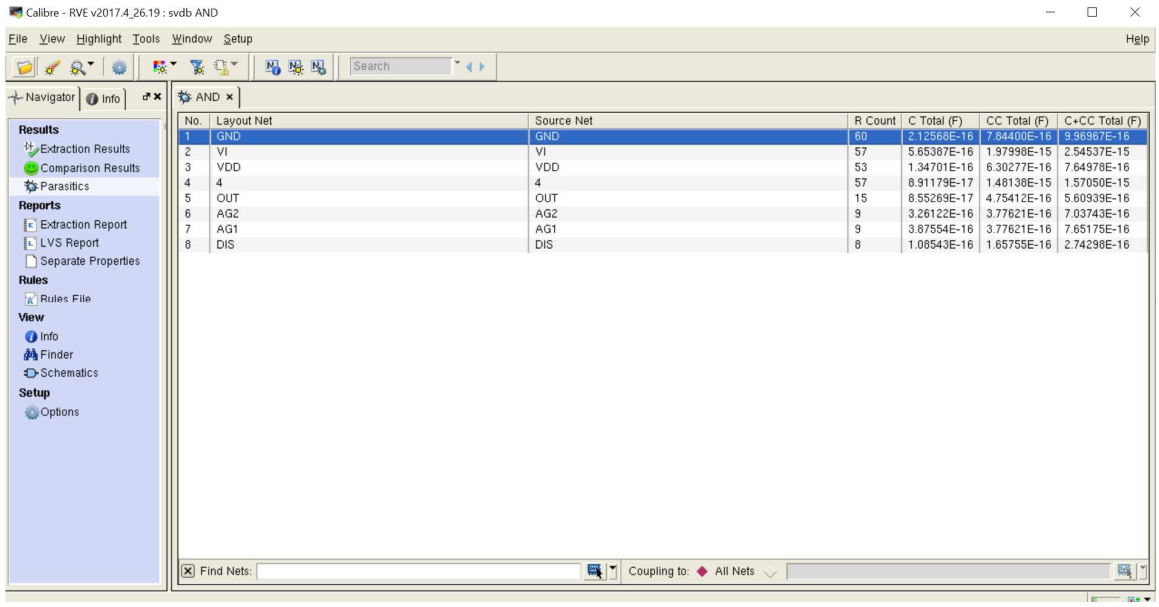


Fig.3.20. PEX Results Window

ESD.WARN.1 which can be fixed at the full-chip level by drawing the SDI drawing around the full chip or else TSMC can waive this violation. Note that by waiving the ESD.WARN.1 violation at the full chip, we are waiving the foundry rules at our risk. Remaining LUP* pattern violations all are related to the latch-up. If we want to waive these violations draw the LUPWDMY drawing layer to fully cover MOS/ACTIVE in the transistor. These errors mainly occur around the discharge transistors, so, we can draw this layer on the discharge transistor. After fixing the DRC's, we will check whether all the interconnects are connected correctly by performing the LVS. In LVS, we compare the layout's extracted circuit netlist w.r.t to the schematic circuit netlist. If any mismatches are found, the Caliber nmLVS will report in the results window. In the setup window, we provide the LVS rule file in the Rules section and in the "Inputs" section, we select the layout format as GDSII and check the option of "Export from layout viewer". Similarly, in the Netlist section, we choose the format as SPICE and check the option of "Export from the schematic viewer". Now, we click on the Run LVS. If everything matches, a green-color-checks with a smiley symbol would appear in the

results window as shown in the Fig.3.19.

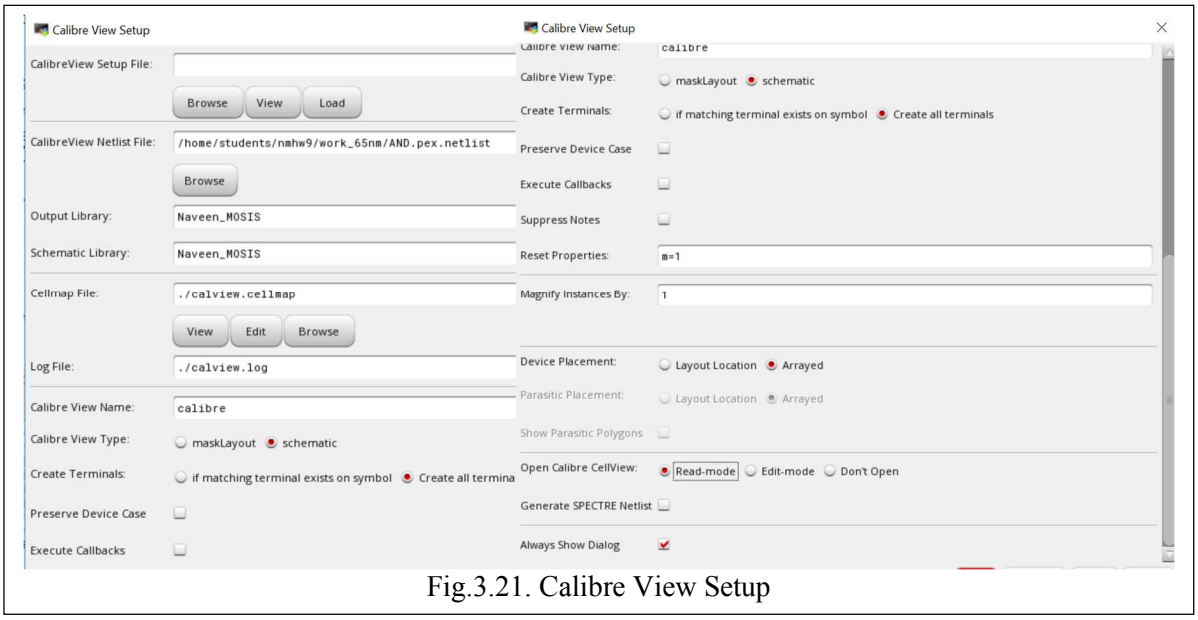

Fig.3.21. Calibre View Setup

If there are any mismatches (if we connect the interconnects wrongly), then the tool shows the errors in the red color and we can debug these mismatches easily from the schematics window in the results window. After the LVS check is completed successfully, we will extract the design using the RC deck and simulate the extracted netlist to check the correct functionality of the layout. In the layout window, we go to calibre and opened the PEX tool. Give the RC deck file as an input to the PEX in the Rules section and in the Outputs section select the format as calibre and click on Run PEX. Now, a results window will pop-up showing either the extraction happened successfully or not as shown in Fig.3.20. Also, an additional calibre view setup would appear as shown in Fig.3.21. We select the calibre view type as schematic, Device Placement as Arrayed and click on OK. We can see the generated calibre view in the AND gate cell in the Virtuoso library manager. Then, we will simulate this extracted netlist using ADE-L by adding the "calibre" in the first argument for the Switch View List (in ADE-L window → setup → Environmental Options). Now, when we click on the green play button, if the simulation is successful, then the waveform will pop-up. Then, we verify whether the

results are the expected outputs. then we proceed for the simulation at all corners.

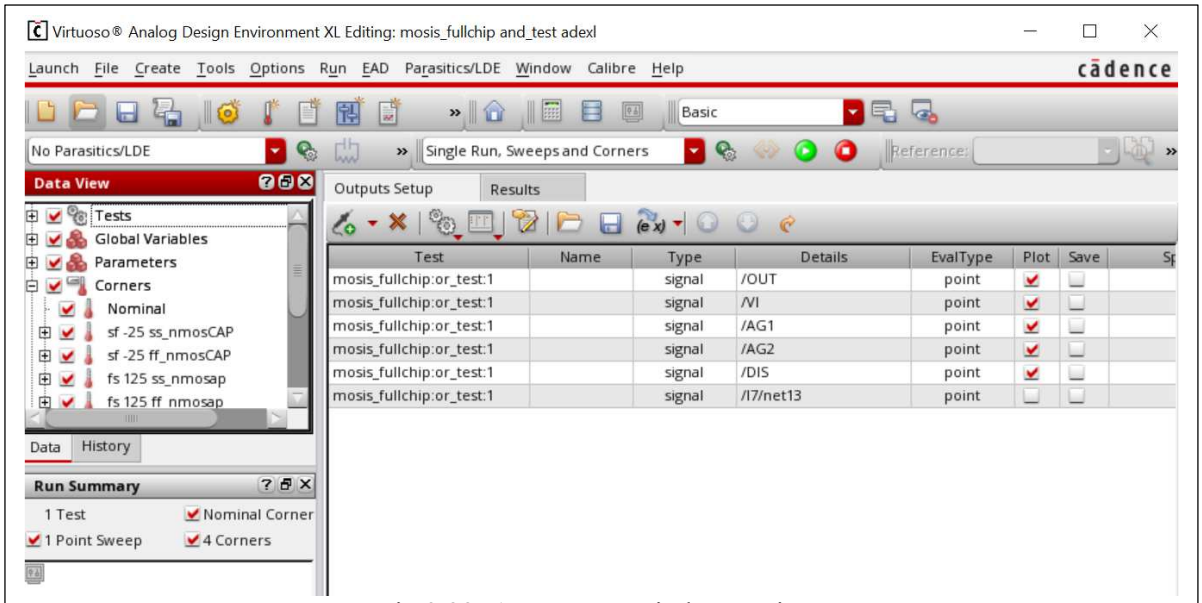


Fig.3.22. ADE – XL Window settings

## 3.9 Simulation of the Extracted Netlist at different PVT corners

We perform different process corner simulations in ADE-XL. We need to make sure that in the Tests section, the correct ADE-L settings are opened for the design to do the simulations. We specify the different corners in the Corners section. Now the window would appear as shown in Fig.3.22. After the successful simulations, we open the waveforms by

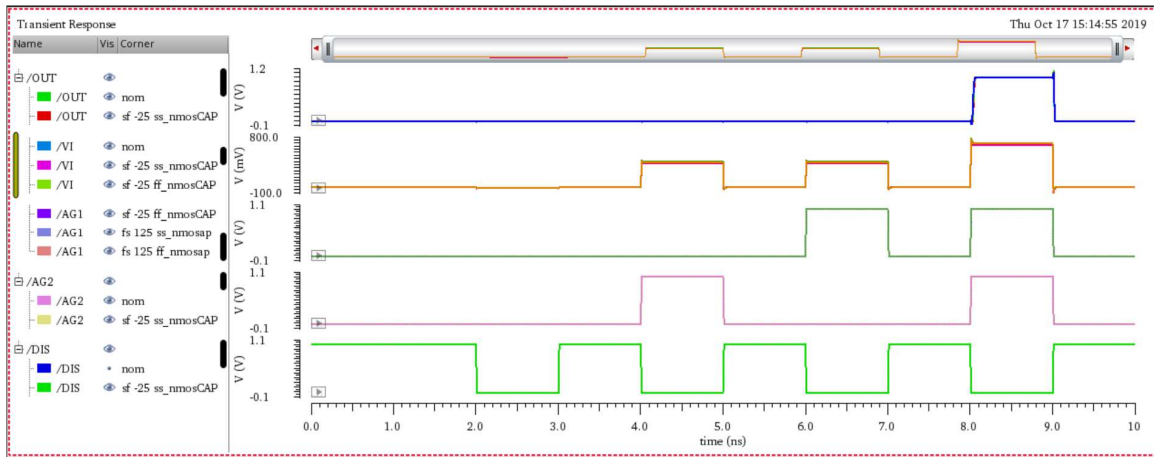


Figure. 3.23 CT AND gate Layout Simulation Results

clicking the Graph button. Fig.3.23 depicts the Simulation response of the Crosstalk AND gate's extracted circuit at different process corners.

# CHAPTER 4

# PVT VARIATION ANALYSIS

This chapter analyzes the PVT variation effects on the Crosstalk gates and discusses the vulnerability of the high fan-in crosstalk gates to variations. As an example, it shows the difficulties in realizing the full adder circuit using the Crosstalk computing technique.

## 4.1 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 the coupling strength of aggressors 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 DC transfer charactristics of the inverter, it's trip points and noise margins can reveal the reliability of the Crosstalk gates. Next, I will discuss the variation effects of individual metrics i.e., Process, Voltage and Temperature and then consider all variations at a time.

### 4.1.1 Considering only process variation

There are three global variation corners that foundry provides as the device models. They are 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 letter represents the process corner for PMOS. Fig.4.1 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.

Thus, the variation would lead to an uncertain shift in the trip point of the inverter. The trip point of the inverter can be calculated from the DC transfer curve. It is the voltage at which the output voltage is equal to the input voltage. I have calculated these points and plotted them in Fig.4.2. This variation becomes an impediment to the Crosstalk gates as the unwanted shift in the threshold curve and the trip point could alter the logic behavior. So, the worst cases would be the curves shifted to left most and right most, which are FS and SF respectively. From Fig.4.1, we can also see that FS has the lowest trip point (0.428V) and SF has the highest trip point (0.503V). This is because, for FS, NMOS becomes stronger and PMOS becomes weaker and thus aids the logic 0 strongly, and vice-versa in SF case. So, the difference between FS and SF processes trip-voltages gives us a process margin for which Crosstalk circuit designs should work reliably. That is the worst-case process shifts should not affect the circuit behavior. So, the process margin that we calculated is 85mV.

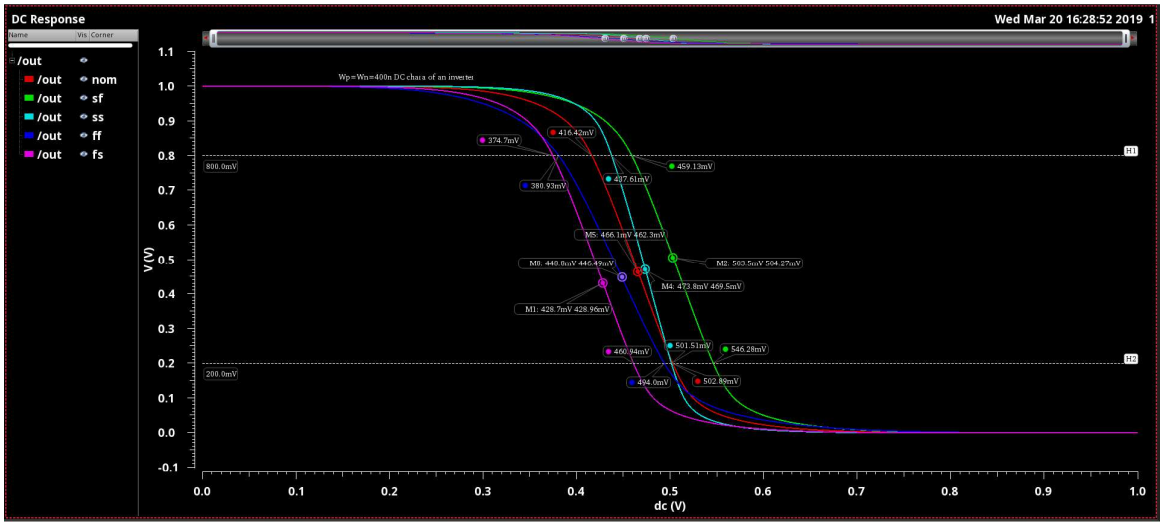

Fig.4.1. Inverter DC characteristics with SF, SS, TT, FS, FF variations

4.1.2 Considering Process and Temperature Variations

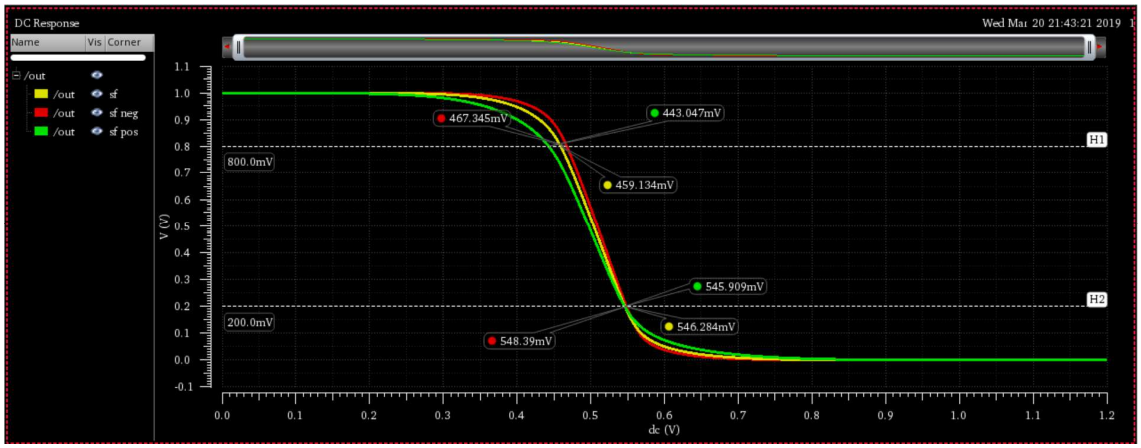


Fig.4.2. Inverter DC characteristics with SF process and Temperature variations

The temperature variation analysis for worst-case FS and SF corners is sufficient as it would give the worst-case variation margin that the Crosstalk circuits have to withstand. Fig.4.2 and Fig.4.3 depicts the DC transfer characteristics of the inverter with added temperature variations for SF and FS corners, respectively. I have considered the temperature extremes, -25 and 125 degrees Celsius, and the typical temperature as 25 degrees Celsius. With the added temperature effect, the variation margin now increased to 105mV which the Crosstalk circuits have to withstand.

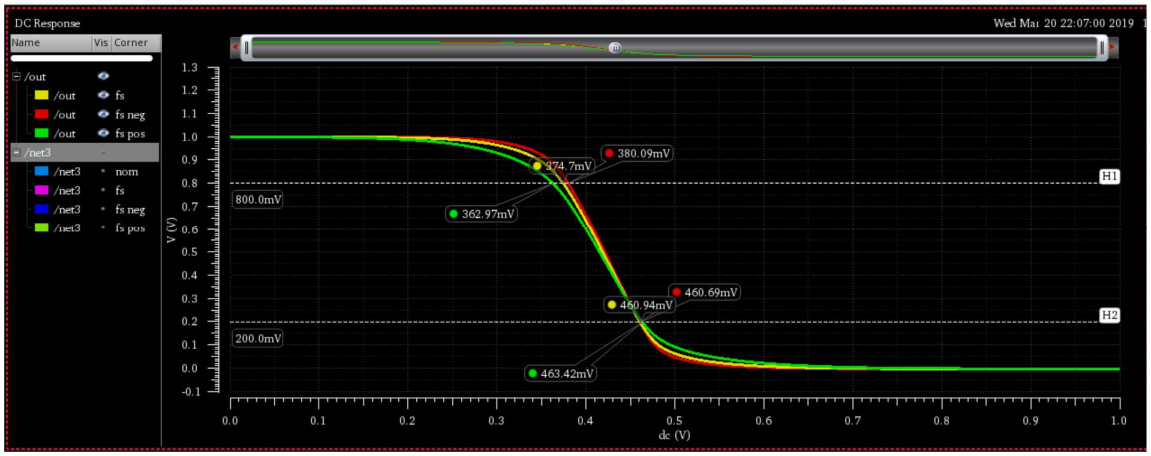


Fig.4.3. Inverter DC characteristics with FS process and Temperature variations

4.2 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 (16). This equation states that the voltage on *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, the voltage half the step size becomes the noise margin that a given gate can withstand and perform functionally correct. For AND Gate, it is 200mV. As 200mV is greater than the variation margin (105mV), the AND gate is functionally stable with a variation. Similarly, for the 2-in OR gate, the noise margin is 125mv.

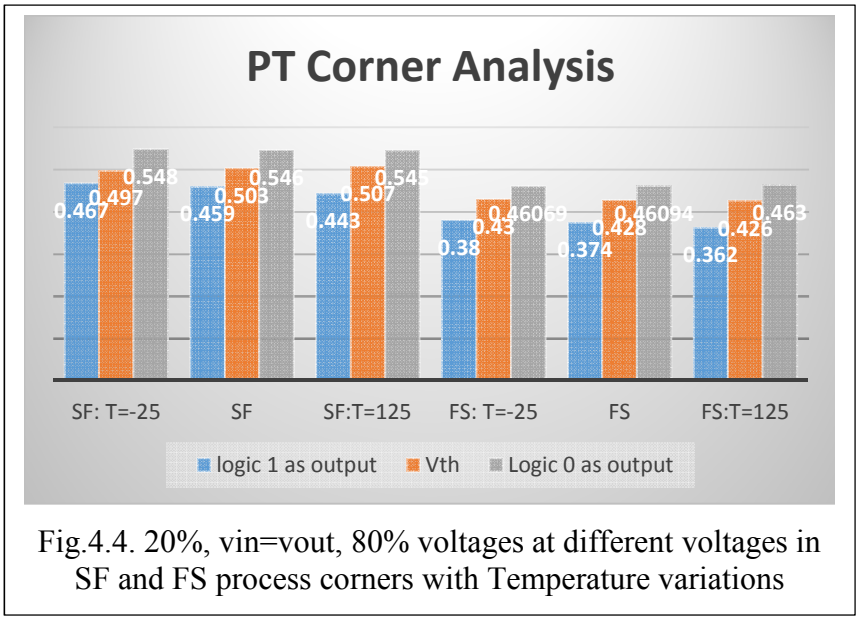


Fig.4.4. 20%, vin=vout, 80% voltages at different voltages in SF and FS process corners with Temperature variations

Similar, in general for all three input Crosstalk gates (AND3, OR3, and Carry), the *Vi* net experiences 4 levels for various input logic combinations. They are 0V, ~300mV, ~600mV, and ~900mV. The step size is this case is 300mV. Therefore the noise margin becomes ~150mV which is again greater than the variation margin. 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 5 input gate, 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. We have implemented a 5-input single-stage full-adder circuit and observed the logic failure with a variation. Therefore, we have designed a two-stage full-adder circuit (by cascading 2 other gates) in this prototype.

CHAPTER 5

DIFFICULTIES OR ERRORS IN LAYOUT DESIGN AND FULL CHIP DETAILS

In this chapter, I discuss the difficulties or DRC, LVS, and PEX errors that occurred while designing the gate level crosstalk circuits and the full-chip level IC using TSMC 65nm PDK.

## 5.1 The errors faced and solutions adopted

### 5.1.1 Grid settings

The grid control settings in the virtuoso layout editor should match the manufacturing grid values provided by the foundry. We need to set the appropriate PDK/Technology grid settings every time we open the layout editor.

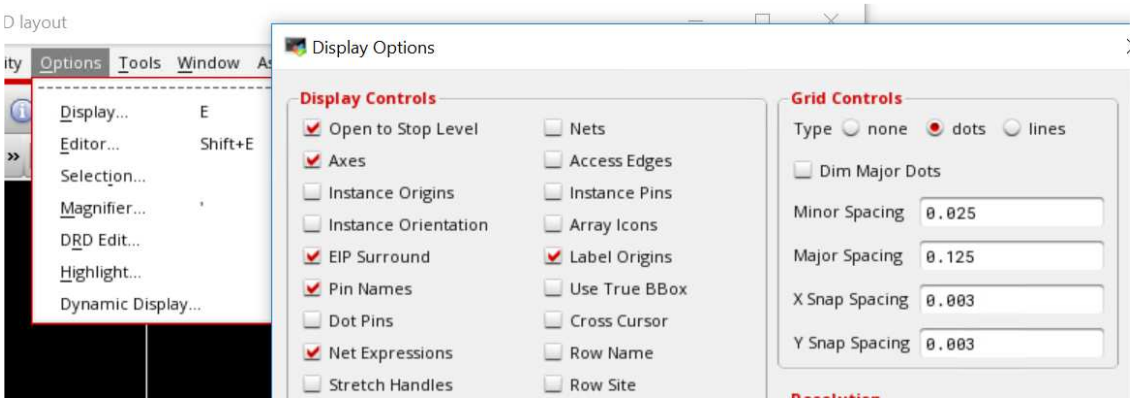


Fig. 5.1 Default Grid settings

Fig.5.1 shows the default grid settings when we open the layout window. These default grid settings need to be updated as shown in Fig.5.2. As the default grid settings are in effect every time we open the tool, we need to make sure we change the spacings before we start any layout drawing.

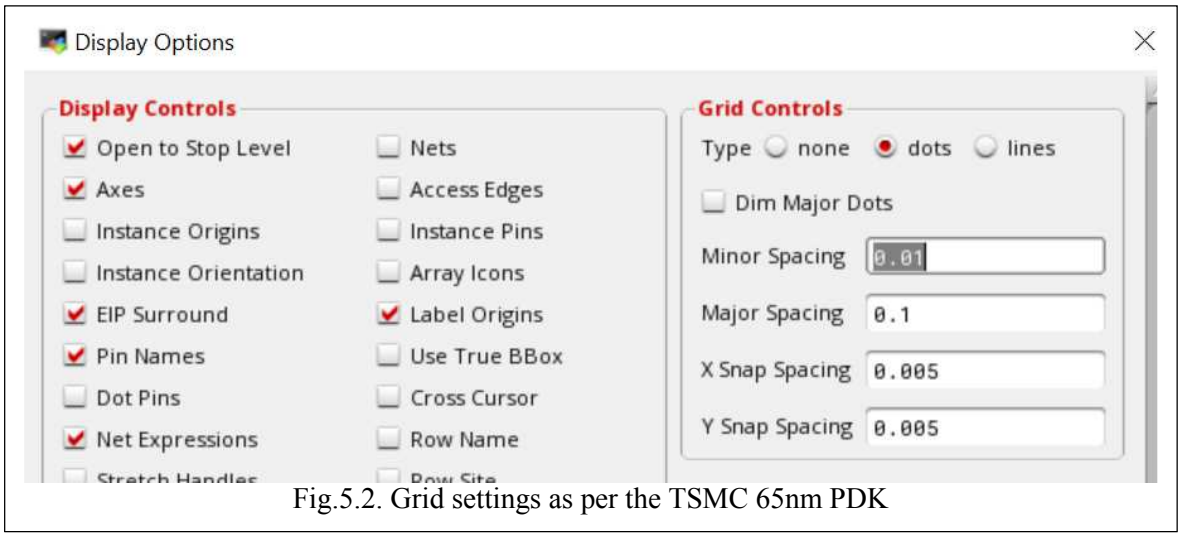


Fig.5.2. Grid settings as per the TSMC 65nm PDK

5.1.2 Warning caused while performing LVS

Fig.5.2 shows the warning that popped up while performing the LVS checking. It is indicating that the NMOSCAP source netlist is not defined in the LVS rules file. To avoid this warning, we need to add the source.added file in the Additional SPEC file section. This file is present in the LVS folder. Fig.5.3 shows the window after adding this file in the LVS run Window.

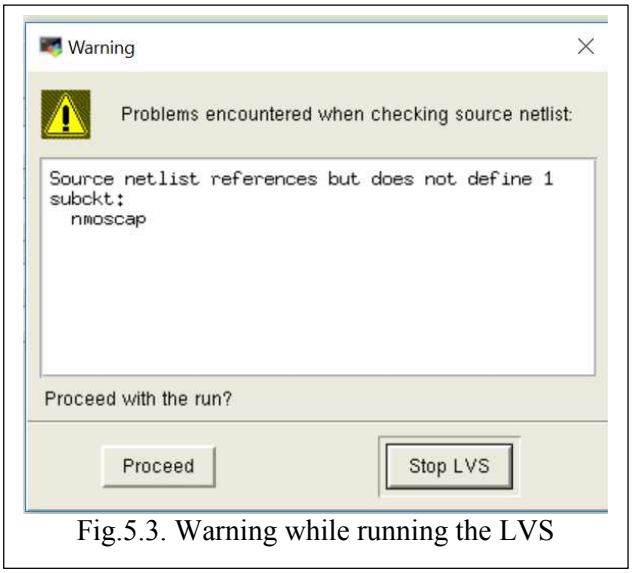


Fig.5.3. Warning while running the LVS

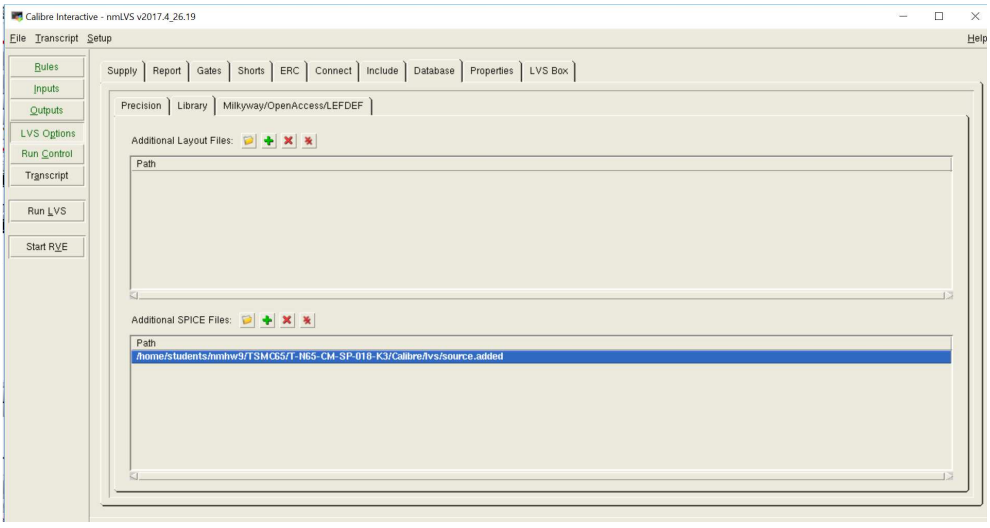


Fig.5.4. LVS run Settings window

5.1.3 XDB issue while running Calibre PEX

Though the DRC and LVS run successfully, we have seen the warning as shown in Fig.5.5 while running the Calibre PEX.

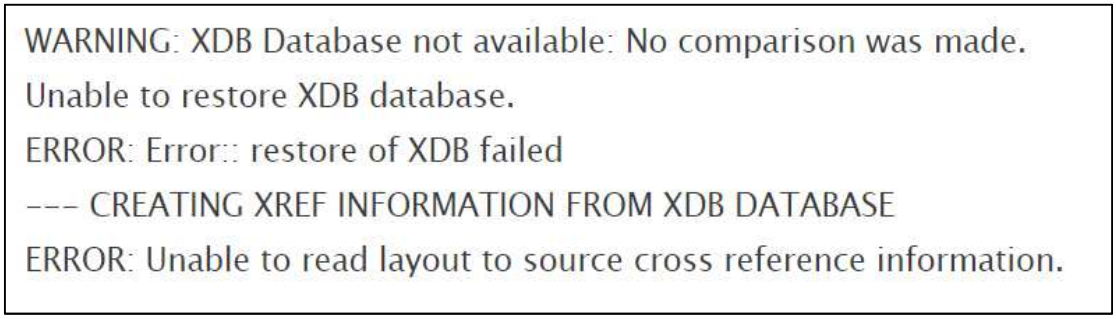


Fig.5.5. Calibre PEX Warning issue

To avoid this warning either we need to add “LVS INJECT LOGIC YES” in the calibre rcx rule deck file or in the Calibre PEX settings window.

5.1.4 Error while compiling the PEX rules file

When we run the PEX to extract the design, we might come across the error as shown

in Fig.5.6. It says that there exists a duplication of the line “PEX IDEAL XCELL YES” in the PEX rules file. So, we can delete the second duplication in the file and rerun it.

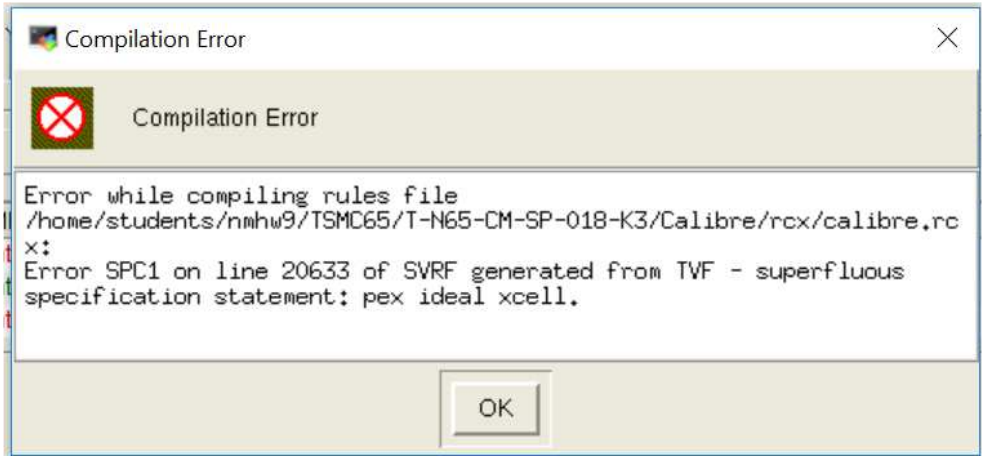


Fig.5.6. Calibre PEX Error

5.2 Details of the Full-Chip

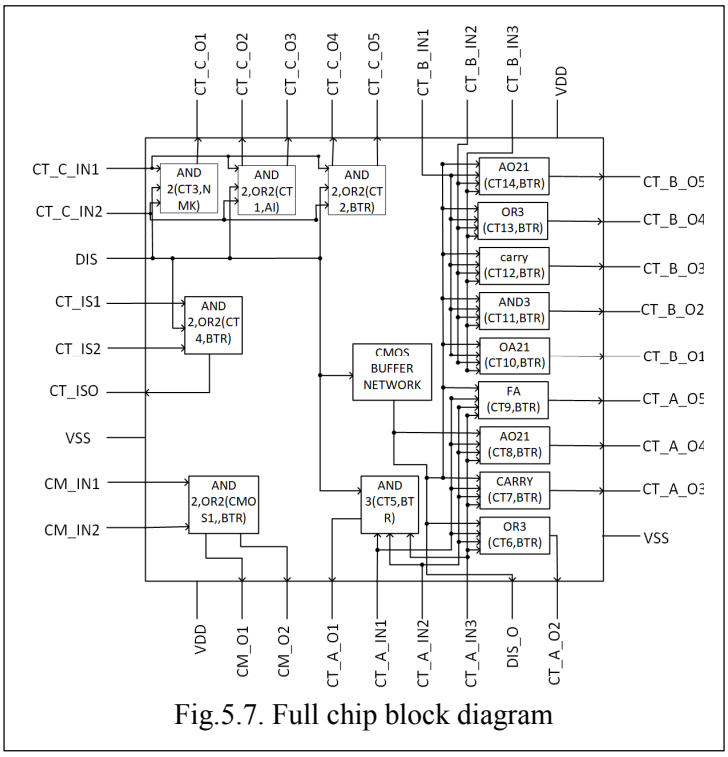


Fig.5.7. Full chip block diagram

We have implemented a crosstalk chip using TSMC 65nm PDK which consists of 9

metal layers. We have designed several crosstalk gates as custom circuit blocks and integrated onto the full chip. The chip is IO limited with 36 pins in total (9 on each side). The size/area of the chip is $1mm^2$ (1mmX1mm). The full chip schematic is shown in the Fig.5.7. It consists of 16 logic gates in total. The types of logic, their input and output pins are shown in the

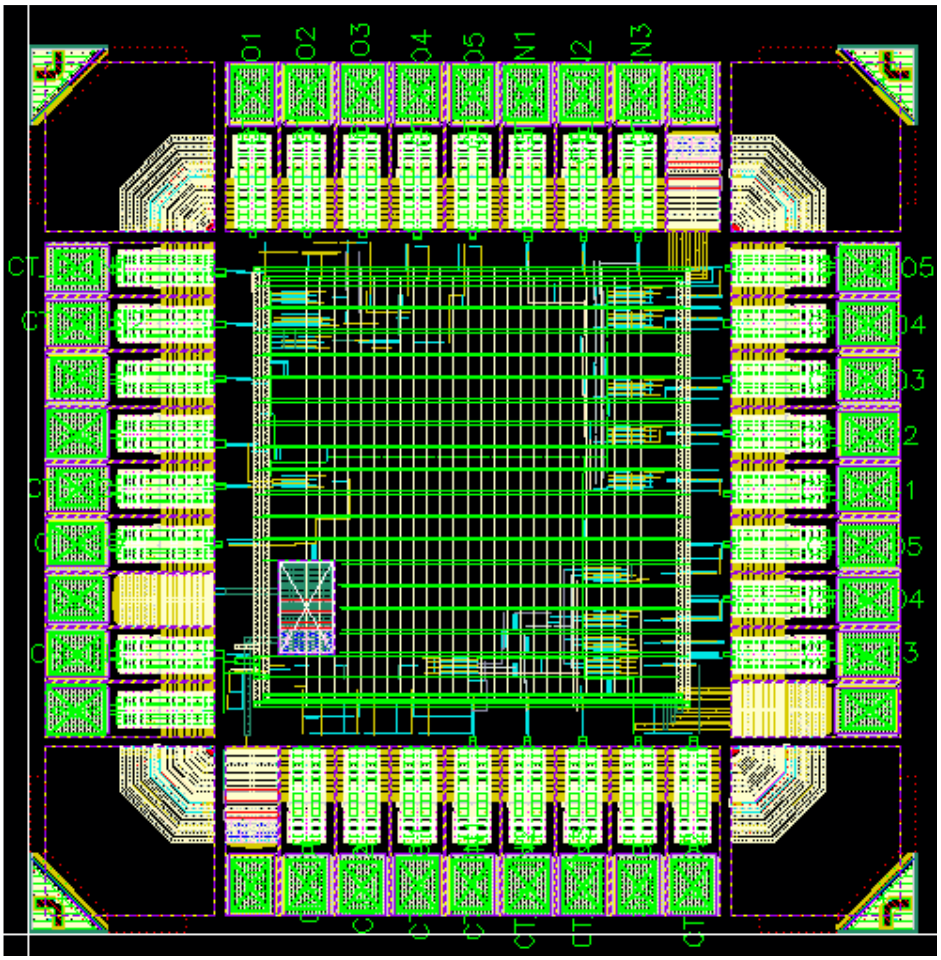


Fig.5.8. Full chip Layout diagram

schematic diagram. The final chip layout is shown in the Fig.5.8, which shows the Crosstalk Circuits we designed, I/O cells, I/O pads, corner cells, filler cells, clamp cells and seal-ring, The capacitors considered are NMOS Capacitors and their sizes are tailored for different logic gates according to the crosstalk logic function requirement. As the crosstalk gates are custom

Fig.5.9.Fabricated chip

gates and they were only a few numbers of gates, we have performed the routing manually. As Crosstalk gates also require a clock signal, we have manually routed a clock network for each gate maintaining the drivability 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. Fig.5.9 shows the fabricated Chip. We have also tested

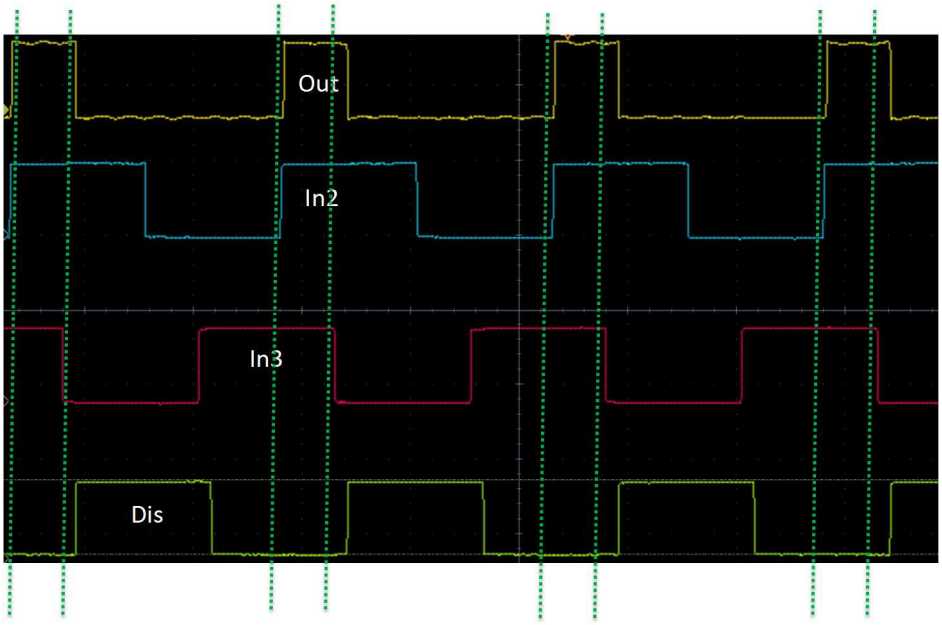

Fig.5.10. OA21 simulation results with In1=0

the Chip. Fig.5.10 shows the test results of the OA21 gate.

# CHAPTER 6

# POTENTIAL MISCELLANEOUS APPLICATIONS

The data-conversion circuits are generally non-linear, in which a continuous time-variant electrical signal is encoded to a stream of binary voltages, or vice-versa. Any signal conversion has to follow a deterministic and faithful flow of information through various proportionate and deliberate mechanisms of signal processing/representations. Following this, our interconnect crosstalk-based signal conversion circuits use the charge conservation-based signal induction and summation mechanisms through engineered capacitive couplings between input and output nets to implement the major portion of the circuit functionality.

## 6.1 Crosstalk DAC

The Crosstalk DAC converts the digital signals into the staircase analog signal using the deterministic charge summation computing technique. The aggressor nets are coupled to the victim nets in the ratios of powers of 2 from the LSB to the MSB, i.e., the coupling capacitances are weighted in ascending manner from LSB to MSB. The digital input transitions on the aggressor nets induce a proportionate summation voltage on the victim net which is precisely proportional to the abstract number the binary bits represent. Next, we present the crosstalk 3-bit and 4-bit DACs and show their simulation results along with the stick diagrams for the layout. The circuits are designed using 16nm PTM transistors and simulated in hspice (at 300C temperature). Fig.6.1(A) shows the crosstalk 3-bit DAC circuit, in which D2, D1, and D0 are the input aggressor nets carrying the digital bits from MSB to LSB, respectively; these aggressors are capacitively coupled to the victim net (Vi) through coupling capacitances 4αCc, 2αCc, and αCc. Cc is the quantized unit capacitance for DAC,

where each input aggressor receives capacitance in multiples of geometrical series of 2. This unit capacitance Cc and the multiplication factor 'α' are shown in Table.1. The crosstalk circuits operate in two states: Evaluation State (Dis=0) when the function/computation is evaluated; and Discharge State (Dis=1) when floating nodes in the circuit are periodically discharged to ground through a discharge transistor which enables control over the floating victim nodes. As the victim node is shorted to ground in the discharge state, the switching of the discharge transistor would not accumulate any erroneous charge onto the succeeding evaluation phase. The victim node is floating only in the evaluation phase, during which, the discharge signal transitions from high to low, thus, it does not induce any extra charge onto the victim net. Our circuit analysis reveals that the high-frequency operations through periodic discharge and evaluation phases does not lead to errors due to charge leakage. During the function evaluation phase, a proportionate charge induction through each of the aggressor coupling capacitances and their net summation on the victim net achieves a continuous piecewise signal with equal step size. The same can be observed in the simulation response of the circuit (Fig.6.1(B)). For instance, with the binary input 001 (for D[2:0]), the crosstalk network computes to voltage level L1, next 010 corresponds to voltage level L2, and so on to voltage level L7 for 111. This signal can be further processed through a reconstructing filter to get a continuous analog signal. Fig.2C depicts a stick diagram layout for this crosstalk based 3-bit DAC circuit. Different layers of diffusion, poly, Metal1, Metal2 etc., are represented and annotated in the figure. Overlap of diffusion and poly layers represents the transistors required for the circuit which are very less in numbers. Metal-1 and Metal-2 layers are used for circuit interconnections whereas Metal-3 is used for crosstalk coupled nets. The required coupling capacitances in this crosstalk network, i.e., between the aggressor (D2, D1, and D0) and victim

(Vi), can be achieved either by different dielectric material choices (represented by ε2, ε1, and ε0) or by specific geometrical arrangements and dimensions for this coupled metal lines. The dielectrics are shown as ε0, ε1 and ε2. They follow the relation ε0 < ε1 < ε2, in accordance with the coupling weights given in the schematic. We have used NCSU 16nm PDK [15] for

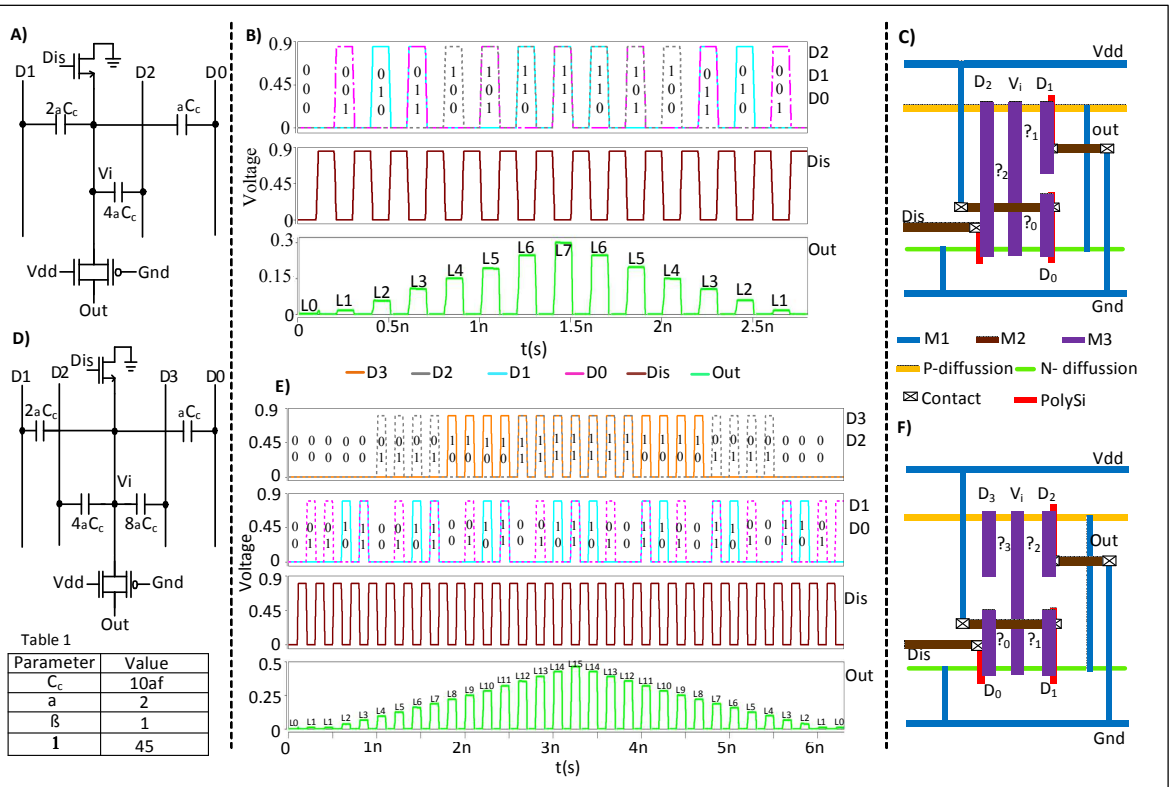


Table 1

| Parameter | Value |
|---|---|
| $C_c$ | 10af |
| a | 2 |
| ß | 1 |
| 1 | 45 |

Fig.6.1. A) 3-bit CT-DAC Circuit B) 3-bit CT-DAC Simulation results C) 3-bit CT-DAC Stick diagram D) 4-bit CT-DAC Circuit E) 4-bit CT-DAC Simulation results F) 4-bit CT-DAC Stick diagram
Table.1. Values of the parameters which are involved in the DAC, ADC and Comparator

the initial evaluation of coupling capacitances achievable in layouts.

An inter-metal-line coupling capacitance data is extracted using Mentor-Graphics-Calibre and it is extrapolated assuming various high-K dielectrics between the metal lines. Thus, we envision to achieve these weighted coupling capacitances by deploying different dielectric materials between the victim and the corresponding aggressors. Furthermore, though the victim net has parasitic capacitances associated with various physical layers, the total capacitance (including plate-to-plate and fringing components) contributed by individual

aggressor nets onto the victim net can be calculated by using accurate capacitive extraction tools and are optimizable to the need. Thus, in addition to the direct plate-to-plate capacitance, both parasitic and fringing capacitances can also be turned as benefits in the crosstalk circuits.

The resolution of the DAC circuit can be increased by employing different innovative circuit techniques and coupling ratios; a straightforward approach is to increase the number of input aggressors and their corresponding couplings (binary-weighted capacitive array). Another approach is by using the segmented input aggressors [14] (Unary and binary-weighted capacitive arrays). We show in this paper the former approach. Fig.6.1(D) represents 4-bit DAC implemented by using an additional input aggressor D4 which receives coupling capacitance $8\alpha C_c$. The simulation response of this circuit is shown in Fig.6.1(E), where 16 binary combinations (0000 to 1111) produce a proportionate 16 voltage levels (L0 to L15) with an equal step size. This output staircase signal can be further processed to a continuous analog signal. A stick diagram representing the layout for this 4-bit DAC circuit is shown in Fig.6.1(F). The arrangement of the crosstalk coupling network for 4 inputs (*D3* to *D0*) and victim net (*Vi*) are shown in metal layer3. It can be noticed from the layout that the number of transistors required for the DAC is just 3 prior to the reconstruction stage.

## 6.2 Crosstalk ADC

This section presents a novel ADC architecture based on the Crosstalk computing concept. Fig.6.2(A) shows the circuit architecture for 2-bit Cross-talk ADC (CT-ADC), which is implemented as a crosstalk tree network of metal lines to convert sampled analog levels to digital code. A sampled analog signal is given through the input aggressor net (Ag) which is coupled to the adjacent victim nets (V11 and V12) through virtual lateral capacitances (12βCc, 6βCc) and these nets are further coupled to sub-victim nets V21&V22 and V23&V24,

respectively (coupling capacitances values are given in Table.1). When a sampled analog signal is driven through input net (Ag), a chain of coupling events takes place on adjacent victim nodes and voltages are induced. These voltages are deterministic and segmented and convey the analog information of the input. The final voltages on leaf victim nodes are given to an encoding block which constitutes of Branch Switching (BS) stage (each leaf node has individual BS-circuitry) and Priority Encoding (PE) stage. Depending upon the series of coupling capacitances leading to each leaf node and threshold of BS, each branch has a margin voltage to switch its output from 1 to 0. Thus, it is designed to generate a thermometer code corresponding to the quantized voltage levels (sample-hold) of the input signal. This

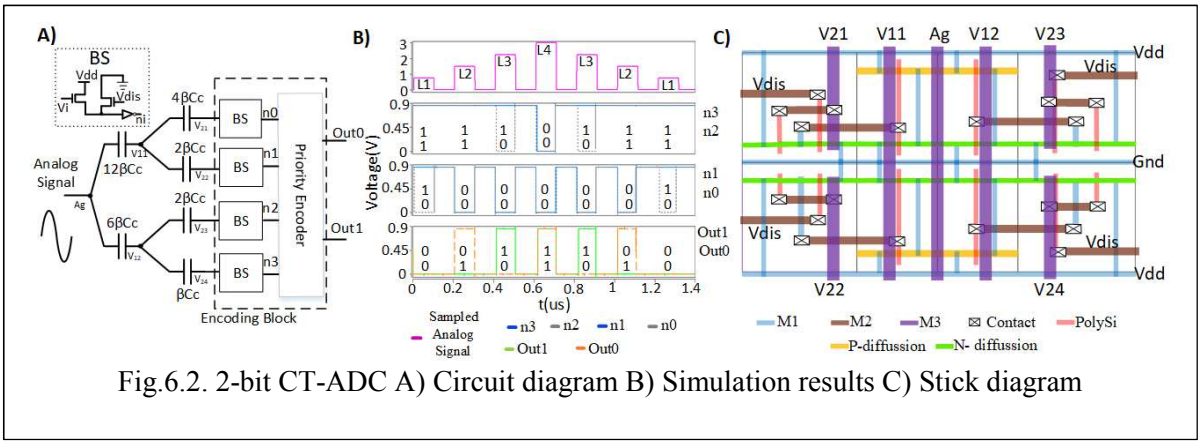

Fig.6.2. 2-bit CT-ADC A) Circuit diagram B) Simulation results C) Stick diagram

thermometer code is next fed to a priority encoder to generate the final digital output.

The Fig.6.2(B) shows the response of the 2-bit ADC circuit. Panel-1 in the figure shows the sampled analog signal consisting of 4 discrete levels (L1-L4). For each level, the BS circuits generate a thermometer code at nodes n3-n0 as depicted in panel-2&3. That is, nodes n3-n0 respond as 1110 for L1, 1100 for L2, 1000 for L3, and 0000 for L4. Passing this code through a PE block gives the outputs 00, 01, 10 and 11 (panel-4) for the input levels L1, L2, L3, and L4, respectively. Thus, a 2-bit analog to digital data conversion is achieved. The layout scheme for achieving this capacitive tree and its crosstalk operations (ADC) is shown

in Fig.6.2(C), where the tree-network is implemented in the metal-3 layer. The center *Ag* metal line is the aggressor which is coupled to two adjacent victim metal-lines V11 and V12. These victim lines are again coupled to sub-victim lines adjacent to them (V21 and V22, V23 and V24). The resolution of the crosstalk-ADC can be increased by increasing the branching stages. Fig.6.3(A) shows the circuit schematic of the crosstalk 3-bit ADC, in which two 2-bit ADC networks are connected in parallel to a root aggressor, forming 8 branches in total. The multipliers β=2 and β=1 (Fig.6.3(A)) are to maintain the coupling strengths for 8 branches in descending fashion.

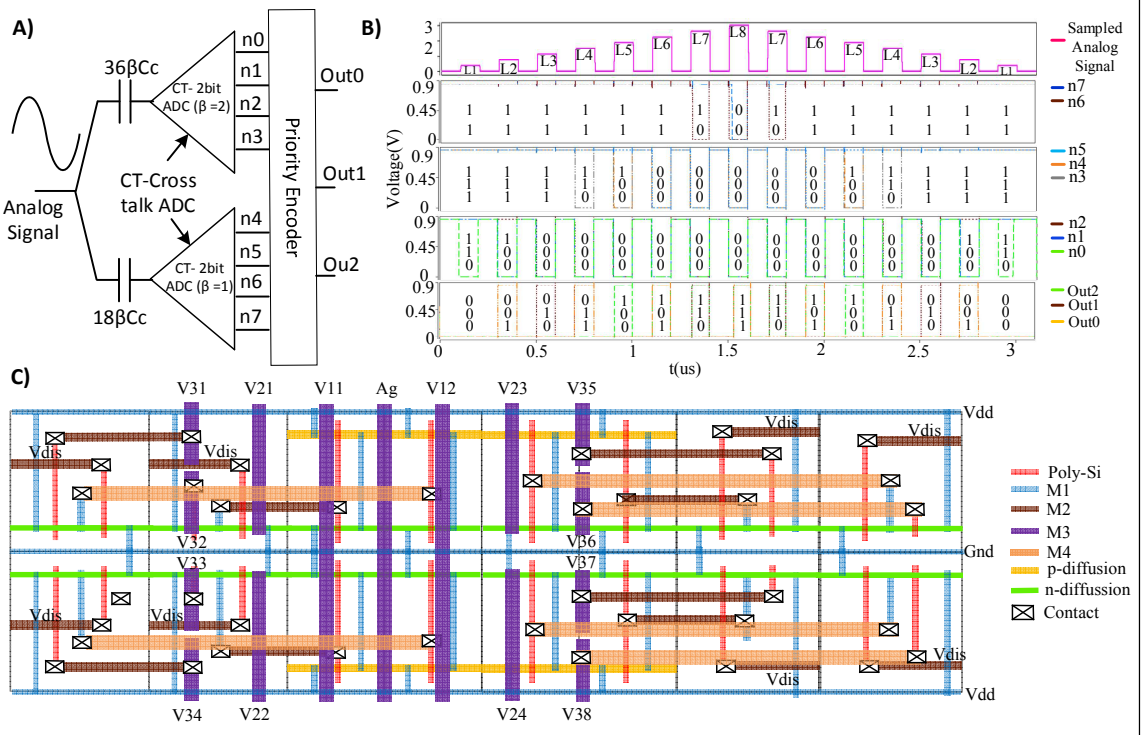


Fig.6.3. 3-bit CT-ADC A) Circuit diagram B) Simulation results C) Stick diagram

Fig.6.3(B) shows the responses of this circuit, where panel-1 shows the input signal which has 8 sample and hold levels, panel-2 to panel-4 show the outputs of 8 BS blocks (n0-n7), and panel-5 shows the PE output. For L1-L8, BSs responses are 11111110-00000000,

and PE responses are 000-111. Fig.4C represents the 3-bit ADC stick diagram, where the tree structure required for crosstalk ADC is implemented in metal layer 3. This layout consumes 32 transistors in total without priority encoder stage.

6.3 Comparator

In this section, we present the idea of a crosstalk-based comparator circuit which compares the sampled analog signal with a dc reference signal. Fig.6.4(A) depicts this circuit, where the sampled input signal and the reference signal are fed through two aggressors

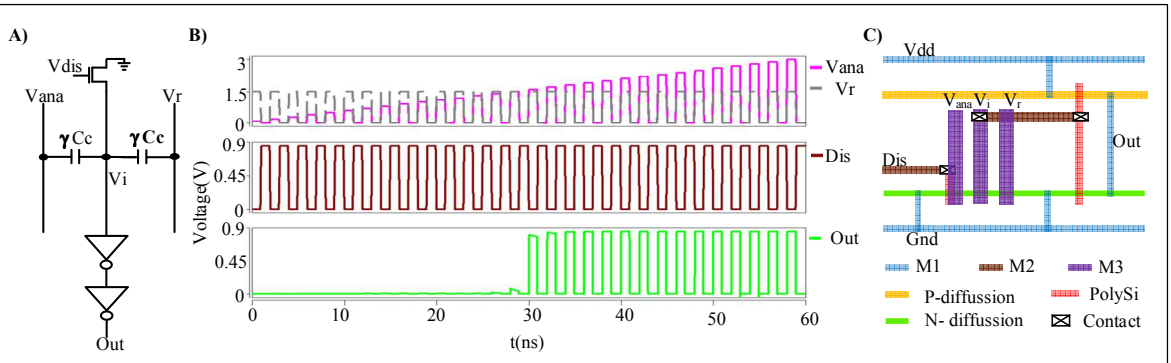


Fig.6.4. CT-Comparator A) Architecture B) Stick diagram C) Simulation results

coupled to the victim node. The coupling values are employed as γCc for both. The victim node is also connected to an inverter which acts as the thresholding function. When the analog signal and reference voltage transition on aggressor nets, they induce a proportionate summation voltage on the victim net whose value is proportional to the coupling strengths. These coupling values are chosen such that the victim node receives enough voltage to flip the state of the inverter only when the analog signal crosses the reference voltage level. This inverter is further connected to another inverter which gives the final output. The response of the circuit is shown in Fig.6B. Panel-1 shows the input signal varying from 0volts to 3volts with 0.1 step size and a reference signal of 1.5volts, panel-2 shows the discharge signal (the comparison operation is performed when dis=0 and when dis=1 the floating victim node is

discharged to ground). It can be observed that *out* node responds as binary level zero when the analog signal ($V_{ana}$) is below the reference ($V_r$) and changes to binary 1 when the analog signal is above the reference signal. Fig.6.4(C) shows the stick diagram layout for this comparator circuit.

## 6.4 Discussion

Our circuit and layout evaluations show that the crosstalk circuit-style requires less number of transistors and therefore the footprint. The transistor count for different circuits presented above is detailed in table.3.

Table 3: Design Metrics for Crosstalk non-linear analog circuits

| 16nm Technology | No. of Transistors | Area ($um^2$) |
|---|---|---|
| 3-bit DAC | 3 | 0.056 |
| 4-bit DAC | 3 | 0.056 |
| 2-bit ADC | 16 | 0.425 |
| 3-bit ADC | 32 | 0.827 |
| Comparator | 3 | 0.086 |

The area estimation and frequency range for the crosstalk 4-bit DAC designed are 0.065um2 and 10MHz-10GHz, respectively; and the rate at which the victim node is discharged and evaluated in the simulation results presented here is 4GHz. Comparing the functional resources requirements, the 4-bit CT-DAC circuit requires 3 transistors and crosstalk metal lines, whereas, the conventional approaches such as Binary weighted DAC [10] requires 4 parallel resistors network and a current-to-voltage converter; likewise, switched-capacitor DAC [11] requires a large number of transistor switches and physical capacitances to achieve the similar staircase signal. For the 3-bit CT-ADC, the area estimation and frequency ranges are 0.827um2 and 1KH - 100MHz, respectively. Finally, the crosstalk

comparator requires just 5 transistors with a footprint area of 0.086um2. Table. 4 shows the comparison of transistor count and power consumption for crosstalk ADC and DAC vs CMOS conventional circuits.

Table 4: Benchmarking results of crosstalk circuits with CMOS

| Technology node (16nm) | Bits | Transistor Count | Power (W) | Supply Voltage (V) |
|---|---|---|---|---|
| Crosstalk ADC | 3 | 32 | 97.64µ | 0.85 |
| CMOS Flash ADC [7] | 3 | 94 | 0.16m | 0.85 |
| Crosstalk DAC | 4 | 3 | 0.1u | 0.85 |
| CMOS Current Steering DAC [8] | 4 | 23 | 0.8m | 0.9 |

For ADC and DAC comparison, we have considered the CMOS Flash ADC [14], and low power current steering DAC [13], respectively. The crosstalk circuits show a significant reduction in transistor count and power consumption. Despite the advantages discussed above, the noise, temperature, and process variation effects pose challenges in the practical realization of the circuit ideas presented here. Though the effects can be alleviated to some extent by using high coupling capacitances, additional design efforts through detailed process corner analysis would mitigate the risks. Thus, further analysis and development would be addressed in the future work.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

Crosstalk Computing is a novel computational circuit style that can offer denser digital logic circuits. The circuits are based on Crosstalk based deterministic charge summation and threshold circuit (CMOS inverter). Previously, our group demonstrated simple, complex and polymorphic logic circuits designed in this approach. For my thesis, I worked on three aspects of Crosstalk Computing: 1) Implementing large-scale Polymorphic circuits using fundamental polymorphic gates, 2) Proof-of-concept prototype of Crosstalk Computing concept, 3) Exploring the potential of Crosstalk Computing in Non-linear Analog Circuits. Along with polymorphic circuits, I discussed in detail from circuit design and layout design of Crosstalk logic gates to and final Chip testing. I also discussed the problems we faced and the solutions adopted while working with the TSMC 65nm.

This thesis also showed the crosstalk techniques that can be used to design the non-linear analog circuits. It provided simulation-based conceptual ideas to implement data converter circuits such as DAC and ADC using the crosstalk computing technique. Our initial explorations and circuit evaluations revealed that these circuit designs consume a smaller number of transistors (3 for DAC and comparator, 32 for 3-bit ADC). Therefore, the required footprint is very less, in fractions of um2, and power consumption is very less compared to the CMOS circuit implementations. Interestingly, unlike the existing circuit solutions, they are compliant with the advancing technology nodes.

The Crosstalk digital logic circuits show potential for applications in radiation-hardened circuits, fault-tolerant circuits and hardware security [1-6,16-18]. In addition, the

ideas presented in chapter 6 can also open up new directions with potential opportunities in very low-power non-linear analog circuits. Therefore, Crosstalk Computing motivates us for further development. Our future work is to make the design using automatic place and route tools.

REFERENCES

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

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

[3]. N. K. Machal, 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.

[4]. R. Desh, N. K. Macha, S. Hossain, R. B. Tejaswini and M. Rahman, "A Novel Analog to Digital Conversion Concept with Crosstalk Computing," *2018 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, Athens, 2018, pp. 1-3.

[5]. 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.

[6]. 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.

[7]. W. Sansen, "Analog design challenges in nanometer CMOS technologies," *2007 IEEE Asian Solid-State Circuits Conference*, Jeju, 2007, pp. 5-9.

[8]. F. D'Agostino D. Quercia "Short-channel effects in MOSFETs" *Proc. Introduction VLSI Design (EECS 467)* pp. 1-15 Dec. 2000.

[9]. D. Arbet, G. Nagy, V. Stopjaková and G. Gyepes, "A self-calibrated binary weighted DAC in 90nm CMOS technology*," 2014 29th International Conference on Microelectronics Proceedings - MIEL* 2014, Belgrade, 2014, pp. 383-386.

[10]. Pradeep Kumar, Amit Kolhe, "Design & Implementation of Low Power 3-bit Flash ADC in 0.18µm CMOS", *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231- 2307, Volume-1, Issue-5, November 2011.

[11]. R. J. Baker, H. W. Li, and D. E. Boyce, "CMOS Circuit Design, Layout, and Simulation" *Piscataway, NJ: IEEE Press*, 1998.

[12]. S. M. I. Huq, S. Islam, N. Saqib and S. N. Biswas, "Design of Low Power 8-Bit DAC Using PTM-LP Technology," *2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*, Warangal, 2017, pp. 64-69.

[13]. Prof. Rahul J. Acharya, "A 45nm CMOS 16-bit Segmented Current Steering Digital-to-Analog Converter", *International Journal of Recent Development in Engineering and Technology*, Vol. 2, Issue 3, March 2014, ISSN 2347 – 6435.

[14]. "Predictive Technology Model (PTM)." Arizona State University, n.d. Web. September 30, 2008.

[15]. N. K. Macha, Bhavana Repalle, Dr. Mostafizure Rahman, "Crosstalk

Computing based Gate-Level Reconfigurable Circuits", *IEEE TNANO*. (Under Review)

[16]. Md Arif Iqbal, Naveen Kumar Macha, Bhavana T. Repalle, and Mostafizur Rahman, "Designing Crosstalk Circuits at 7nm", *2019 IEEE International Conference on Rebooting Computing (ICRC)*, San Mateo, California, 2019.

[17]. Md. A. Iqbal, N. K. Macha, Bhavana T. Repalle, and Mostafizur Rahman, "From 180nm to 7nm: Crosstalk Computing Scalability Study," *IEEE/ACM NANOARCH 2019*, Quingdao, China.

[18]. Md. A. Iqbal, N. K. Macha, et al., "A Logic Simplification Approach for Very Large Scale Crosstalk Circuit Designs," *2019 IEEE S3S Conference*, San Francisco, California.

# VITA

Bhavana Tejaswini Repalle, received her Bachelor of Technology degree in Electronics and Communication Engineering in 2015 from Vignan's Lara Institute Of Technology and Science, Andhra Pradesh, India. She is currently working towards her Master of Science (MS) in Electrical and Computer Engineering at the University of Missouri - Kansas City under supervision Dr. Mostafizur Rahman. Currently, she is a Graduate Research Assistant in Nano IC Computing Lab. Her research interests include Exploring the circuit and design styles of Crosstalk Computing. Her career interests are Physical Design, Circuit Design and RTL Design and Verification.