

# Low-Power Digital Circuit Design Through GDI and CMOS Integration

**M Rama Krishna**

department of Electronics and  
Communication Engineering  
Ramachandra College of Engineering  
Eluru, India  
Ramakrishna05419@gmail.com

**M Suma**

department of Electronics and  
Communication Engineering  
Ramachandra College of Engineering  
Eluru, India  
Vasavisuma9@gmail.com

**Abstract**—One of the methodologies to design low-power combinational circuits is Gate Diffusion Input (GDI). Logic functions F1, F2, OR, AND, MUX, and NOT are implemented with the help of only two transistors. Comparing GDI with traditional CMOS and current pass transistor logic approaches, the former offers less logic complexity and reduces the area, propagation latency, and power consumption of digital circuits. Low power dissipation is achieved through the reduction in the count of transistors. The interconnect effects are reduced as the area is minimized. The logic gates (GDI and CMOS) were first implemented in Cadence at 90 nm in this paper, and gate libraries were prepared for them. Using these libraries, ISCAS-85 benchmark circuits were designed and analyzed in terms of area, power, and latency in order to estimate the complexity and scalability of GDI. GDI circuits turn out to be better

**Key words**-Full Swing, GDI, ISCAS-85,  
VLSI Standard cell library

## 1.INTRODUCTION:

Gate Diffusion Input technology: Propagation delay is one of the significant performance parameters used to measure the efficiency of a digital circuit. For decades, CMOS technology has been the most preferred approach in the designing of digital circuits. Initially, in the days of circuit designing, the circuits were compared based on the speed of their clocks, which gave rise to numerous options that were optimized towards CMOS. Because it required significantly fewer transistors to perform an operation, the first interest was in Pass Transistor Logic (PTL). Another concept that mitigated the threshold drop effects of PTL was the Transmission Gate (TG)

N	P	G	Out	Function
0	B	A	$\bar{A}B$	F1
B	1	A	$\bar{A}+B$	F2
1	B	A	$A+B$	OR
B	0	A	$AB$	AND
C	B	A	$\bar{A}B+AC$	MUX
0	1	A	$\bar{A}$	NOT

, which used a parallel combination of NMOS and PMOS transistors. A similar technique that was explored was Gate Diffusion Input.

GDI, however requires logic level restoration and may well have reduced circuit speeds at lower power supply voltages.

Figure 1 depicts a basic GDI cell. The bulk terminals in both PMOS and NMOS are connected to VDD and GND, respectively. The key difference between GDI and CMOS is that, for a given circuit, in GDI inputs can be provided to VDD or GND besides the N, P, or G terminals. Because there is no strict rule connecting supplies to either PMOS or NMOS, low swing takes place, and therefore, it becomes difficult to use GDI in analog circuits

A basic GDI cell can be utilized for several Boolean functions implementations. In the case of static CMOS, the majority of the functions have complicated implementations which require up to 12 transistors. This implies that only two transistors are required. As a result, it is much better for GDI. The most complicated implementation technique for MUX, as presented in Table I, which may be achieved by utilizing.

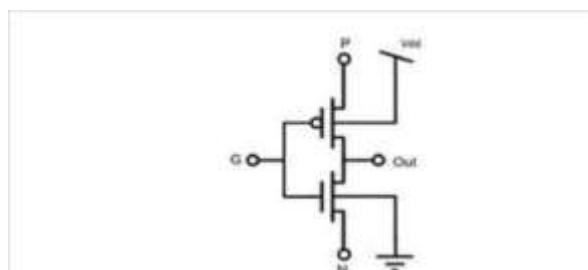


Figure.1 is GDI basic cell

TABLE I. GDI FUNCTIONS

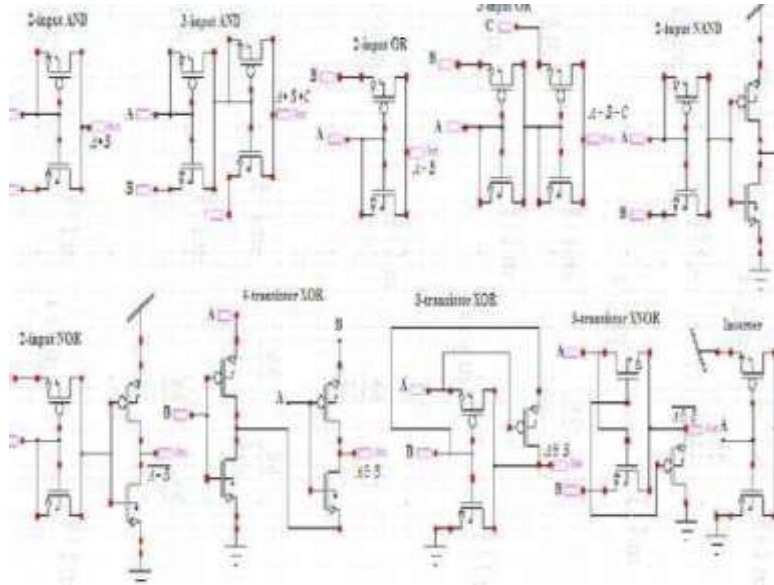


Figure 2. Schematics of gates in GDI

TABLE II. FUNCTIONALITY OF F1 FUNCTION IN GDI

A	B	Functionality	F1
0	0	PMOS Transmission Gate	$V_{TP}$
0	1	CMOS Inverter	1
1	0	NMOS Transmission Gate	0
1	1	CMOS Inverter	0

To improve drivability, buffers can be used to recover swing. Consider the noise function presented in Table II. Only when  $A=0$  and  $B=0$ , while the output is not exactly zero, is the low swing problem present. This is due to models with both structural and behavioral styles of Verilog. Also available in the online documentation is the list below with some high-level descriptions of circuits used in this project.

### B. Circuits for ISCAS-85:

ISCAS-85 Circuits:

- c17 : Parity and syndrome calculator
- c432 : Interrupt controller (27 channel)
- c499/c1355 : Single error check circuit

(32 bit)

- c880 : ALU (8 bit)
- c1908 : SEC/DED circuit (16 bit)
- c2670 : ALU and controller (12 bit)
- c3540 : ALU (8 bit)
- c5315 : ALU (9 bit)
- c6288 : Multiplier (16x16)
- c7552 : Adder/comparator (32 bit)

#### A. ISCAS Benchmarks:

The high level ISCAS-85 can be accessed online at [2]. Demonstrations are further performed at block, circuit, and net-list level using representations. Gate-level netlists appear at RTL blocks. They are quite usually applied in research applications related to digital design, timing analysis, test patterns generation, and mapping technology

## II. BACKGROUND

Advantages of GDI are highlighted by comparing its performance with typical CMOS in terms of area, transistor count, power, and delay. The necessity for faster, more compact design techniques and reduced power dissipation has sparked a number of research efforts in this area

## DESIGN METHODOLOGY

The following steps were followed in this project:

- Generating GDI and CMOS basic gate schematics, symbols and layouts in Virtuoso
- Power dissipation and delay comparison of basic gates (CMOS & GDI)
- Designing gates for full logic voltage swing
- Generation of .lib files for CMOS and GDI
- Synthesis of ISCAS benchmark circuits in CMOS and GDI (90 nm) •  
Comparison of CMOS and GDI with the ISCAS synthesis results

### Full Swing:

GDI is based on Shannon's decomposition theorem. The PMOS transistor is on to pass P to the output when the gate input is low, as shown in Figure 1. The NMOS is on to pass N input when the gate input is high. Output, though is worse when  $G=1$  and  $N=1$ , or when  $G=0$  and  $P=0$ , because PMOS transmits only a weak "0" and

NMOS a weak "1.". Circuits can be adapted slightly to make these latter voltage swings stronger.

One input is applied to the AND gate while the other to N. The P input is shorted to ground. From Shannon's theorem, we can compute the output.  $Y = 0 + A.B = A.B$ . Strong logic swing is developed at the output as depicted in Figure 3 by connecting more transistors. It is a design of the gate with the use of inverted logic.

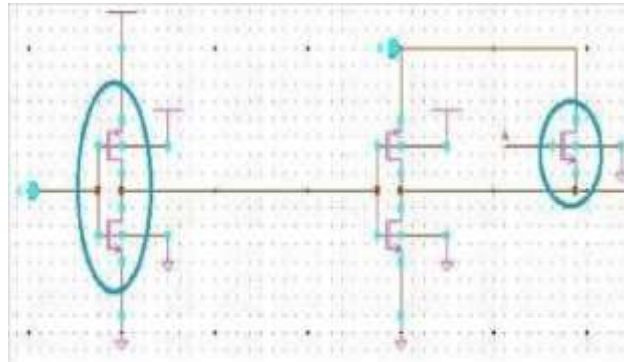


Figure 3. GDI 2-input AND gate for Full Swing

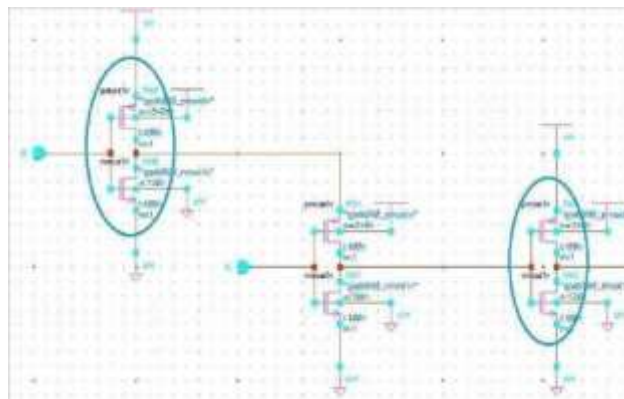


Figure 4. GDI 2-input OR gate for Full Swing

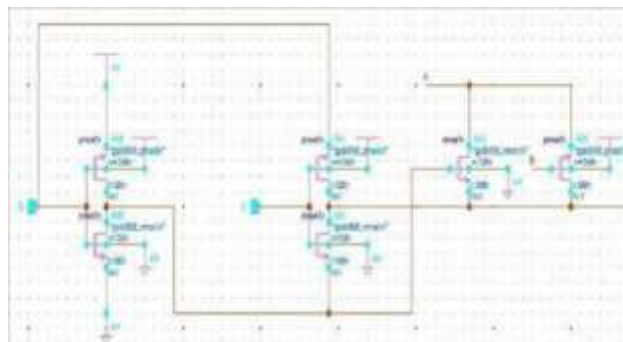


Figure 5. GDI 2-input XOR gate for Full Swing

An inverter is made use of. To draw down a weak zero, you may add another NMOS transistor. As can be seen from Figure 4, you could add four more transistors to an OR gate to increase the swing of its output. As shown in Figure 5, the XOR gate is also constructed with swing restored using two additional transistors. A typical cell library should include the XOR gate since it is very commonly used in adder, multiplier and divider circuits.

### Noise Margin:

Design margins known as noise margins keep the circuits within predetermined limits. Radiation waves (soft error sources), the operating environment, electric and magnetic fields, and supplies are only a few noise sources. Noise is also generated by switching transistors near each other. A set of margins referred to as noise margins are established to ensure reliable operation in these situations.

Let  $V_{IH\ min}$  be the low input (driven) voltage and  $V_{OH\ min}$  be the low output (driver) voltage for a "1". For  $NMH = |V_{OH\ min} - V_{IH\ min}|$  the receiver must interpret it to be "1"; therefore, it should always be a nonnegative quantity. Noise, however takes a high voltage to its lower level by adding together on wires with the signal. Therefore, to attain maximum performance in every parameter of the operation,  $NMH$  must be positive. Similarly,  $NML = |V_{IL\ max} - V_{OL\ max}|$  must also be positive for the logic "0" condition for safe operation. In other words, the noise margin is the tolerance band by which the signal voltages may vary and still be safely detected. Higher the noise margin, the more resilient would be the circuit.

The supply used in the experiment was 1.8V.  $V_{OL} = 0.45V$  and  $V_{OH} = 1.45V$  are the desired margins. Full swing versions were designed for the AND, OR, XOR and MUX gates. The simple AND gate, for instance, had to provide a swing of only 0.27V to 1.54V. Table III's noise margin criteria are already met by the AND & OR gates themselves without any restoration, though GDI is a design technology for non-restoring logic circuits. One must therefore be doubly conservative on the requirements of noise margin.  $V_{OL}$  was hence fixed at 0.25V and  $V_{OH}$  was set at 1.65V for this project. The swing in the new design will thus change from 7 $\mu$ V to 1.79V for the AND gate.

TABLE III. NOISE MARGINS FOR VARIOUS  $V_{DD}$  LEVELS

Technology	$V_{DD}$	$V_{OH}$	$V_{IH}$	$V_{TH}$	$V_{IL}$	$V_{OL}$
5V CMOS	5	4.44	3.5	2.5	1.5	0.5
5V TTL	5	2.4	2	1.5	0.8	0.4
3.3V LVTTTL	3.3	2.4	2	1.5	0.8	0.4
2.5V CMOS	2.5	2	1.7	1.2	0.7	0.4
1.8V CMOS	1.8	1.45	1.2	0.9	0.65	0.45

Principles taken from [6] were adapted for the full swing variant of 2:1 MUX. This design has a more reliable output swing compared to the original gate but at a cost of four additional transistors. Figure 6 displays the new design. With 12 transistors, this is still below the CMOS variant.

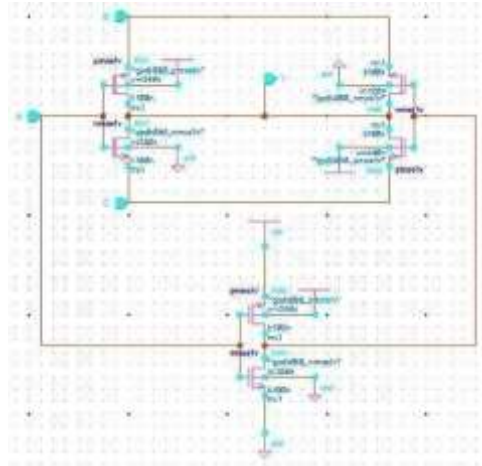


Figure 6. 2:1 MUX for full swing

Figure 7 depicts the waveforms that differ in one another. The basic version has low levels of (0.22) mV and high levels of only 1.58V while the full swing version has a swing of between 0.74mV and 1.79V. Because the input is routed directly to the drain diffusion, negative input voltages should not be used because they result in a number of noise problems in GDI circuits.

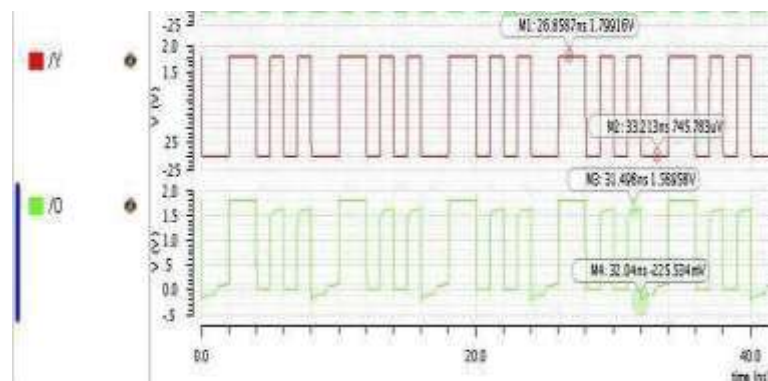


Figure 7. Full swing version (Y) & basic MUX (O)



### C . Standard Cell Library Creation:

The use of cell-based design involves a library [8]. The synthesis tool utilizes the Liberate tool, generating .lib files for the standard cells by finding the optimum circuit topology, designing the function, calculating the timing and power characteristics, and developing a standard cell layout template. It then makes a check for design rules and schematic matches, carries out parasitic extraction, performs back annotation, conducts DC characterization, and provides transient response estimation [20].

First, the area of each cell is extracted from its netlist and written into a text file with appropriate names. net-lists for gates of Following manufacturing in ADEL, the selected gates are translated into Spectre format. Additionally, a template file is made [19] that gives the delay arcs and pin-outs for each gate. A characterisation file is then made. This file invokes all of the SPICE transistor models (gpd90 nm), the process corner models, the gate net-list and template files, and the .lib file and datasheets when this file is run in Liberate [21]. The datasheet also contains the truth tables for each gate, which are calculated

```
// Library name: AND2X1
// Cell name: AND2X1
// View name: schematic
subckt AND2X1 A B Y
  NM2 (B A Y 0) gpd90_nmos1v w=(120n) l=100n as=69.6f ad=69.6f \
    ps=1.16u pd=1.16u m=(1)*(1) simM=(1)*(1)
  NM1 (Y net9 0 0) gpd90_nmos1v w=(120n) l=100n as=69.6f ad=69.6f \
    ps=1.16u pd=1.16u m=(1)*(1) simM=(1)*(1)
  NM0 (net9 A 0 0) gpd90_nmos1v w=(120n) l=100n as=69.6f ad=69.6f \
    ps=1.16u pd=1.16u m=(1)*(1) simM=(1)*(1)
  PM1 (Y net9 B vdd!) gpd90_pmos1v w=(240n) l=100n as=67.2f ad=67.2f \
    ps=1.04u pd=1.04u m=(1)*(1) simM=(1)*(1)
  PM0 (net9 A vdd! vdd!) gpd90_pmos1v w=(240n) l=100n as=67.2f \
    ad=67.2f ps=1.04u pd=1.04u m=(1)*(1) simM=(1)*(1)
ends AND2X1
// End of subcircuit definition.
```

Figure 8. Spectre net-list sample

from its schematic. Areas of cells are added from an externally prepared text file to the .lib file. All three aspects power, area, and timing are accommodated within the .lib file. Selection for Standard Cell Library: All of the functions in Table I are included in the GDI library. An XOR gate and a 2:1 MUX were also added because they are useful for data route circuits such as adders and control logic. The inverter and universal gates were selected for the CMOS library. Two compound gates, AOI21 and AOI22, were included. Compound gates are useful to realize several functions like XOR and MUX. For example, the AOI22 A=S and C=S can be replaced to get Figure 9. GDI Synthesis of c17 ISCAS Circuit.



Figure 9. GDI Synthesis of c17 ISCAS Circuit

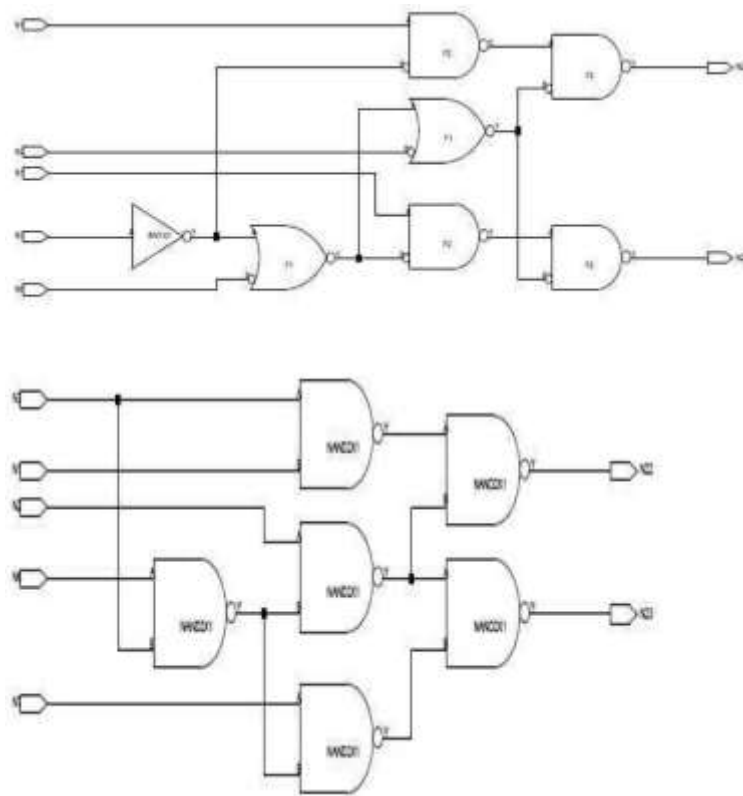


Figure 10. CMOS Synthesis of c17 ISCAS Circuit

tables for each gate, which are calculated from its schematic. Areas of cells are added from an externally prepared text file to the .lib file. All three aspects power, area, and timing are accommodated within the .lib file.

**Selection for Standard Cell Library:** All of the functions in Table I are included in the GDI library. An XOR gate and a 2:1 MUX were also added because they are useful for data route circuits such as adders and control logic.

The inverter and universal gates were selected for the CMOS library. Two compound gates, AOI21 and AOI22, were included. Compound gates are useful to realize several functions like XOR and MUX. For example, the AOI22  $A=S$  and  $C=S$  can be replaced to get the MUX with the gate,  $Y=AB + CD$ . In addition, both the libraries included tri-states. Each gate was developed to work with the same drive currents and have the same drive value of an inverter of  $A$ .

### Synthesis:

CMOS and GDI. F1 and F2 gates are used for GDI, but for the CMOS, only NAND gates are used. The number of transistors in such a simple circuit can easily be counted: 14 for GDI and 24 for CMOS. That will provide GDI-implemented circuits with an edge Current Source Delay Models: CMOS Synthesis of c17 ISCAS Circuit A current source model plots output current as a nonlinear function of input and output cell voltages [23]. A timing analyzer plots voltage versus time by integrating output current and using a random RC network to determine the propagation delay. Output current versus time is tabulated for various values of input slew rate and output

capacitance using Liberty Composite Current Source Model (CCSM). The Effective Current Source Model, ECSM, tabulates the output voltage versus time .

**TCL File for synthesis of ISCAS circuits:** In order to ease the synthesis process, a file was prepared. The file is written in the basic understanding of TCL language synthesis commands [14]. Using this, circuits can be synthesized with only two commands: source the syn.tcl file and then enter the module name when the rc -gui has been opened. The algorithm that the TCL file will be based on is: Take a string input for the file name. The inclusion of the HDL file list, libraries, and module names within the setup file sets library attributes to the environment; reads the HDL file list from the setup; develops the module; synthesizes it with mapping to the library; assigns variables for area, power, and timing filenames before launching the GUI that will create area, power, and timing report.

- Generate report(.rep) files with filename; repeat for each module.

## II. RESULTS

### *Power and Delay Calculation:*

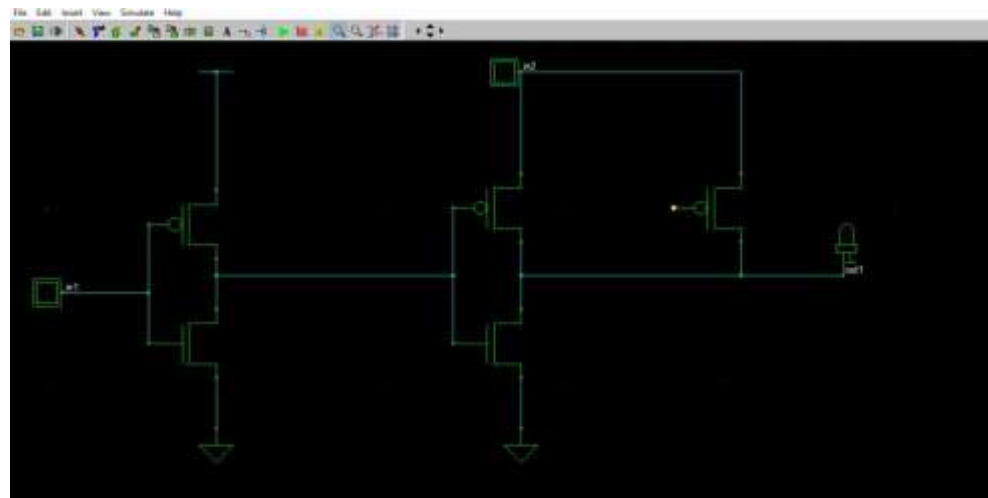
This design used VDD of 1.8V and transistors of Cadence 90 nm technology. Table IV compares the GDI gate with its CMOS counterpart, indicating that GDI gates are much faster than CMOS and consume much less power, particularly at low fan-in.

TABLE V. COMPARISON OF ISCAS-85 CIRCUITS IN CMOS AND GDI

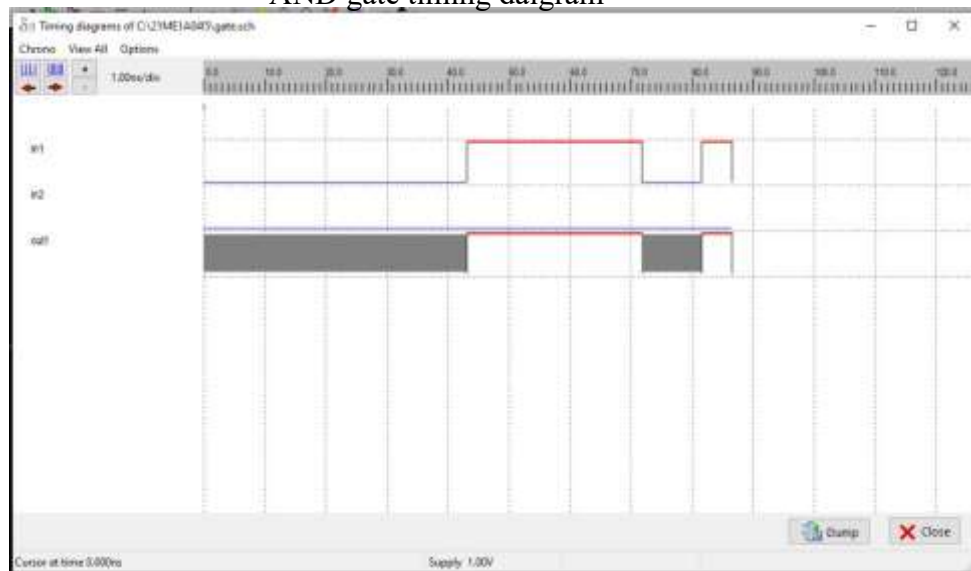
Circuit	No. of cells		Area ( $\mu\text{m}^2$ )		Static pwr ( $\mu\text{W}$ )		Total pwr ( $\mu\text{W}$ )		Delay (ps)	
	CMOS	GDI	CMOS	GDI	CMOS	GDI	CMOS	GDI	CMOS	GDI
c17	6	7	56	33	0.23	0.16	0.24	0.17	164	262
c432	146	142	1409	671	5.87	3.14	6.06	3.24	1052	2234
c499	355	187	3896	1812	15.62	14.74	17.77	28.24	1222	990
c880	288	302	3117	1632	12.03	9.04	12.66	12.41	1672	1962
c1908	353	269	3727	1833	14.87	11.43	16.15	20.59	1873	1577
c2670	423	461	5022	2697	18.95	15.6	20.37	24.07	1428	1636
c3540	739	843	8723	4612	31.81	22.2	33.57	29.03	2312	2142
c5315	988	966	12092	6426	45.38	34.82	48.63	55.55	1967	1469

c6288	2335	1981	21065	9367	97.67	41.09	102.98	44.71	5818	5855
c7552	1244	1050	13820	7114	55.99	47.24	59.67	85.4	2704	2410
Total	6877	6208	72927	36197	298.42	199.46	318.03	303.41	20212	20537
Avg	687.7	620.8	7292.7	3619.7	29.842	19.946	31.803	30.341	2021.2	2053.7

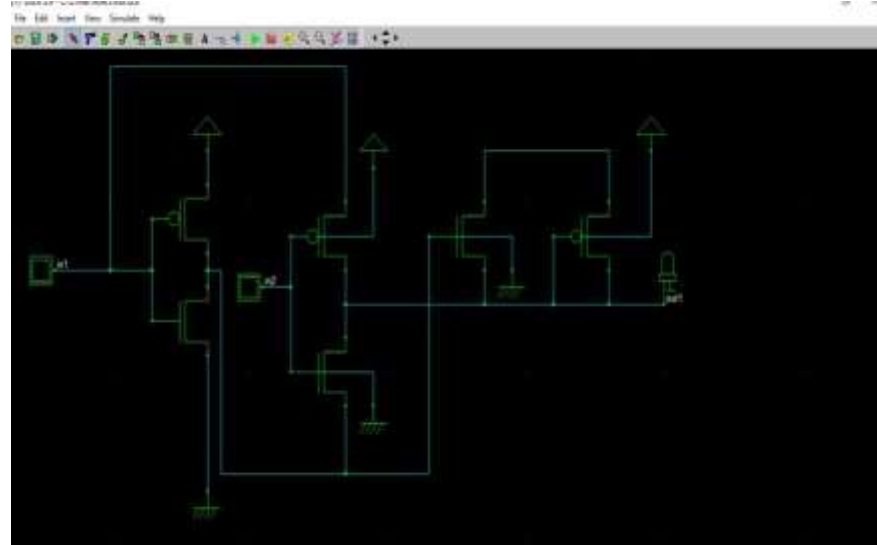
GDI2- input AND gate for full swing AND gate circuit daigram



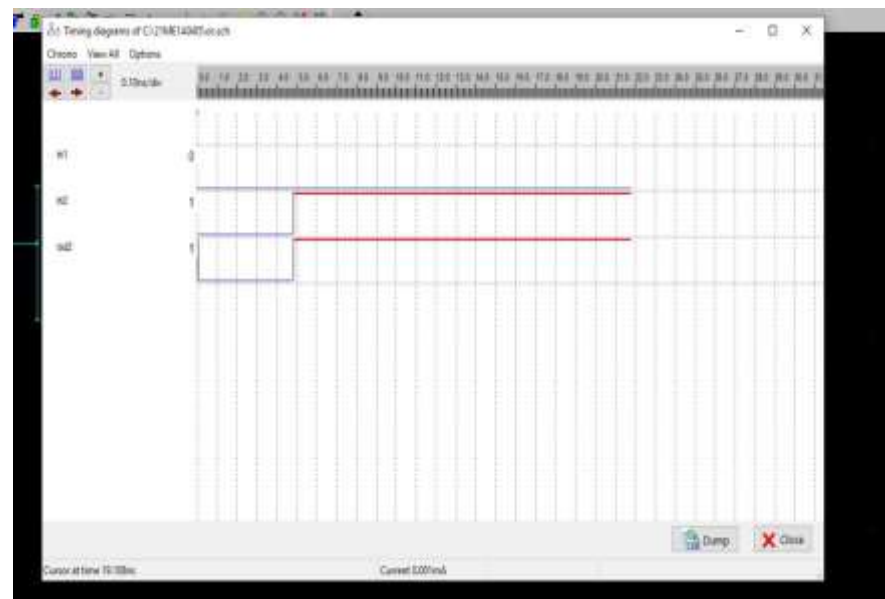
AND gate timing daigram



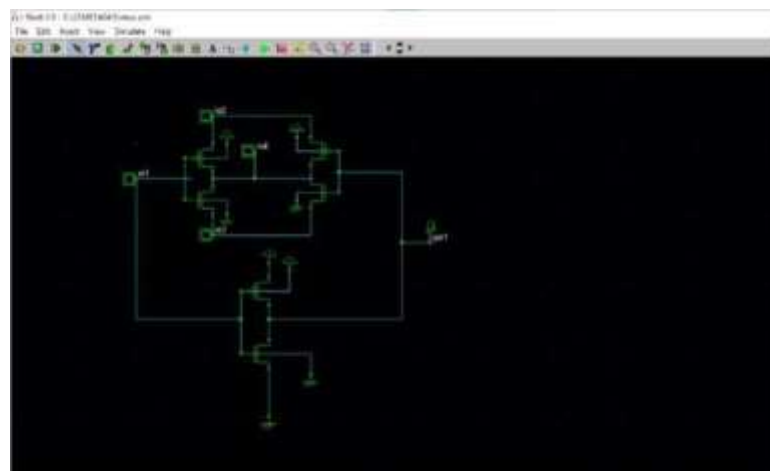
GDI2-input OR gate for full swing OR gate circuit daigram



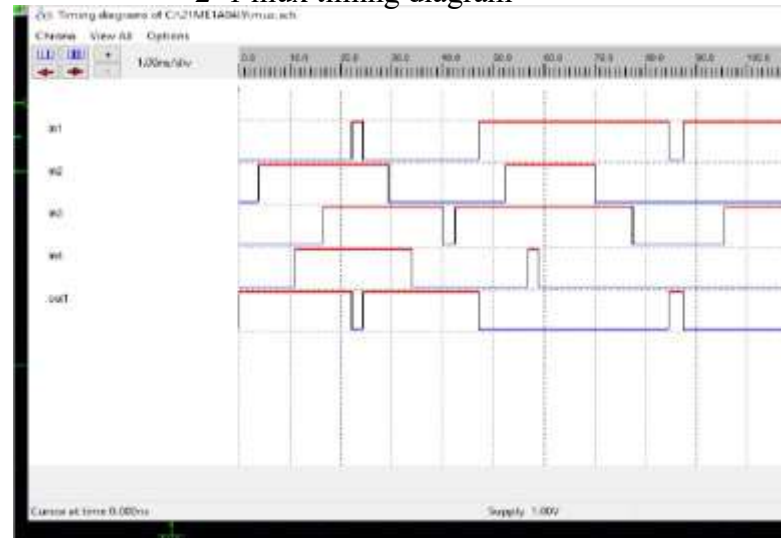
OR gate timing daigram



2\*1 MUX circuit daigram



2\*1 mux timing diagram



### Drawbacks of GDI:

The circuits which involve a transistor's diffusion terminal as an input are quite noise sensitive. One of the major problems associated with injection of minority charges is charge sharing. Assume that the input to the NMOS transistor was driven below GND by coupling or power supply noise to explain the noise problems.

Whenever the VGS exceeds VT, then the transistor is triggered. Let us assume that we build a latch that will implement a transmission gate. Now this noise would allow it to drain down to have a wrong value of "0" if it held the "1." In case input overcomes VDD, the PMOS transistor would again give the incorrect value. Thus dynamic logic circuit are not able paired with GDI. In order to bypass these problems, standard cell latches use buffered inputs. Since GDI combinational logic is by its nature unbuffered, noise affects it too for the same reason. Also, open diffusion leads to back-propagation of noise that contaminates the input signals and complicates delay modeling. Data channels with tightly controlled inputs can utilize GDI since they result in faster circuits.

### CONCLUSION AND FUTURE WORK

Cadence tools are applied in comparison of the values in terms of power and delay between GDI and CMOS basic gates. With 90 nm design done using Cadence Virtuoso, it's inferred that GDI is quicker than CMOS gates. Moreover, GDI utilizes lesser power. Thus to enhance the fan-in level from the previous design, this full swing is applied for an improvement in the same design. Without it, it would be minimal, for sure.

The area, power, and timing reports were obtained. ISCAS-85 benchmark circuits have been created using the gates from the library files in 90 nm technology for both CMOS and GDI

In that sense, we can draw a conclusion from the GDI technology comparing it to the CMOS technology. It merely puts its static power significantly low, and its usage area significantly reduces. One might have inverters and buffers at the end of every stage which would minimize its relatively greater delay along with lesser overall power consumption.

Therefore, GDI is may be a middle way between CMOS and PTL since it achieves full swing outputs and sacrifices uniform design that needs periodic restoration for low area and reduced leakage, even at deep sub-micron levels

Future research on GDI circuits will provide the defects like stuck at faults. Every circuit shall be fault modeled. Afterwards, fault excitation follows, in which the place of the fault is determined using a test vector. A group of fault tests shall be developed Future.

## ACKNOWLEDGMENT :

The authors extend their appreciation to the Department of Electronics and Communication Engineering, Ramaiah Institute of Technology, for help at various stages of research with their professors and VLSI lab personnel.

## References :

- [1] A. Morgenshtein, A. Fish, I. A. Wagner, "Gate-diffusion input (GDI) – a technique for low power design of digital circuits: analysis and characterization", Proc. IEEE International Symposium on Circuits and Systems, vol. 1, pp. 477-480, Arizona, USA, May 26-29, 2002.
- [2] Benchmark circuits: [http://www.pld.ttu.ee/~maksim/bench\\_marks/](http://www.pld.ttu.ee/~maksim/bench_marks/)
- [3] ISCAS-85 circuits block diagrams with functionality: <http://web.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html>
- [4] Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits, Authors: Bushnell, M., Agrawal, Vishwani D.
- [5] Digital VLSI Chip Design with Cadence and Synopsys CAD Tools,
- [6] Author: Eric Brunvand
- [7] S. Hiremath, A. Mathad, A. Hosur and D. Koppad, "Design of low power standard cells using full swing gate diffusion input," 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), Bangalore, 2017, pp. 940-945.
- [8] L. M. Naga and P. Mullangi, "Design and development of an ASIC standard cell library using 90 nm technology node," 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2018, pp. 1-6.
- [9] Introduction to the Custom Design Flow: Building a standard cell EE241 Tutorial 3,
- [10] Author: Brian Zimmer (2013)
- [11] Mahmoud Aymen Ahmed, M. A.
- [12] Mohamed El-Bendary, Fathy Z. Amer, Said M. Singy, "Delay optimization of 4-Bit ALU designed in FS-GDI technique", Innovative Trends in Computer Engineering (ITCE) 2019 International Conference on, pp. 534537, 2019.
- [13] M. Mukhedkar and W. B. Pandurang, "A 180 nm efficient low power and optimized area ALU design using gate diffusion input technique," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), Pune, 2017, pp. 47-51.