# Design and Analysis of Low-Power, High Speed and Area Efficient 2-Bit Digital Magnitude Comparator in 90nm CMOS Technology Using Gate Diffusion Input

Fasil Endalamaw

***Abstract*****—**Digital magnitude comparators based on Gate Diffusion Input (GDI) implementation technique are high speed and area-efficient, and they consume less power as compared to other implementation techniques. However, they are less efficient for some logic gates and have no full voltage swing. In this paper, we made a performance comparison between the GDI implementation technique and other implementation methods, such as Static CMOS, Pass Transistor Logic (PTL), and Transmission Gate (TG) in 90 nm, 120 nm, and 180 nm CMOS technologies using BSIM4 MOS model. We proposed a methodology (hybrid implementation) of implementing digital magnitude comparators which significantly improved the power, speed, area, and voltage swing requirements. Simulation results revealed that the hybrid implementation of digital magnitude comparators show a 10.84% (power dissipation), 41.6% (propagation delay), 47.95% (power-delay product (PDP)) improvement compared to the usual GDI implementation method. We used Microwind & Dsch Version 3.5 as well as the Tanner EDA 16.0 tools for simulation purposes.

***Keywords*****—**Efficient, gate diffusion input, high speed, low power, CMOS.

## I. Introduction

In recent years, demand for high-speed, area-efficient, and ultra-low-power portable battery-operated devices like laptops, mobile phones, and tablets has increased. Very Large Scale Integration (VLSI) designers have traditionally focused on boosting the speed and shrinking the size of digital systems. However, with the advancement of portable systems and improved Deep Sub-Micron (DSM) fabrication processes, power dissipation has become another important design consideration [1]. The increasing amounts of area and power consumption limit the usability of circuits as the scale of integration increases. [2]. A design's power consumption affects how much energy is consumed per operation and how much heat is dissipated by the circuit. These factors influence a great number of critical design decisions, such as the power-supply capacity, the battery lifetime, supply-line sizing, packaging, and cooling requirements [3]. There are certain techniques for reducing power consumption but they are performed at the expense of area, speed, or design complexity. In some cases, smarter architecture at the system level can achieve the goal of significantly reducing power usage without negatively impacting other design aspects [4].

A digital magnitude comparator is a combinational circuit that compares two binary numbers to see if one is more than, equal to, or less than the other. Since a digital magnitude comparator is a decision-making device, it forms an important component in numerous control devices, such as biometric authentication and password verification processes, analog to digital converters, address decoding of computers and microprocessor-based devices (to select a specific input/output device for the storage of data), sensors (where the binary numbers representing measured non-electrical signals such as speed, temperature, position, etc., are compared with a threshold binary value).

It is possible to design and implement a digital magnitude comparator using the GDI technique with a fewer number of transistors as compared to other implementation techniques: Static CMOS, PTL, TG. This results in a reduction of power consumption, delay, and also area in GDI circuits. In this paper, the GDI implementation technique is designed and performance comparison is made with other techniques. A methodology is also proposed.

The organization of the paper is as follows: Section II discusses the basics of a 2-bit magnitude comparator. Section III presents the implementation of a 2-bit magnitude comparator using Static CMOS, PTL, GDI, and TG techniques. Section IV presents the proposed methodology to implement a 2-bit magnitude comparator and in Section V, the simulation results are presented. The results are compared and contrasted with the Static CMOS, PTL, GDI, and TG techniques. Finally, the conclusion is presented in Section VI.

## II. Digital Magnitude Comparator

If A and B are the two numbers being compared, the output of a digital magnitude comparator is in the form of three binary variables reflecting the conditions A = B, A > B, and A < B [5].

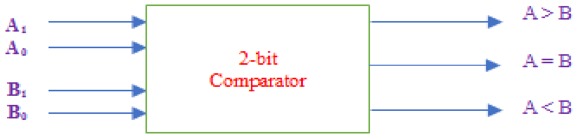


Fig. 1 Bit magnitude comparator block diagram

The output changes state depending on the relative magnitude of the two numbers. The two numbers, represented

Fasil Endalamaw is with Faculty of Electrical & Computer Engineering, Bahir Dar University, Ethiopia (e-mail: fasilend@gmail.com).

in 2-bit binary as (A1A0) and (B1B0) [Fig. 1], are equal if all pairs of significant digits are similar, that is, A1 = B1 and A0 = B0. We examine the relative magnitude of pairs of significant digits, starting with the most significant, to decide if A is greater than or less than B. If the digits of the pair under evaluation are identical, the comparison is made by comparing the next adjacent lower pair of digits. Until a pair of unequal digits is reached, the comparison goes on and on. If Ai = 1 and Bi = 0 in the pair of unequal digits, A > B, and if Ai = 0, Bi = 1 in the pair of unequal digits, A < B.

The truth table for a 2-bit digital comparator is shown in Table I. From the truth table, logical expressions for each output can be expressed as:

$$X = [(A1B1' + A0B0')\ (A1 + B1')]$$
$$Y = [(A0'B0 + A0B0') + (A1'B1 + A1B1')]'$$
$$Z = [X + Y]'$$

where the output variables: X, Y, and Z represent the A > B, A = B, and, A < B conditions.

The logic diagram of a 2-bit digital magnitude comparator is shown in Fig. 2.

TABLE I
TRUTH TABLE OF A 2-BIT COMPARATOR

| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| $A_1$ | $A_0$ | $B_1$ | $B_0$ | A > B | A = B | A < B |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

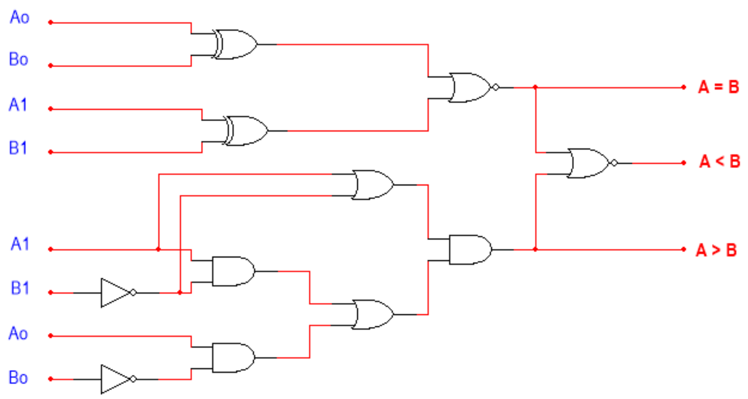


Fig. 2 Logic diagram of a 2-bit magnitude comparator

## III. METHODS TO IMPLEMENT A 2-BIT MAGNITUDE COMPARATOR

A 2-bit digital magnitude comparator, which is a combinational logic circuit, can be realized using a variety of implementation techniques, also called circuit families. Among these circuit families, in this paper Static CMOS, PTL, TG, and GDI techniques are discussed.

### A. Static CMOS Logic

The Pull-Up Network (PUN) and Pull-Down Network (PDN) are incorporated in a static CMOS gate. Fig. 3 shows a generic N input logic gate where all inputs are distributed to both the pull-up and pull-down networks. The PUN and PDN are built in such a way that they are mutually exclusive, with only one of them active in steady-state. [3]. PDN makes a connection from GND to output (F) when F [$In_1$, $In_2$, ……….….., $In_N$] = 0 whereas PUN makes a connection from VDD to F when F [$In_1$, $In_2$, ……….….., $In_N$] = 1. They have good noise margins and are fast, low power, insensitive to device variations, easy to design, widely supported by CAD tools, and readily available in standard cell libraries [6].

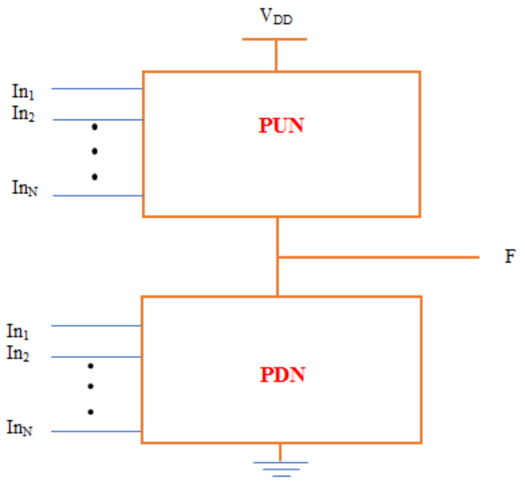
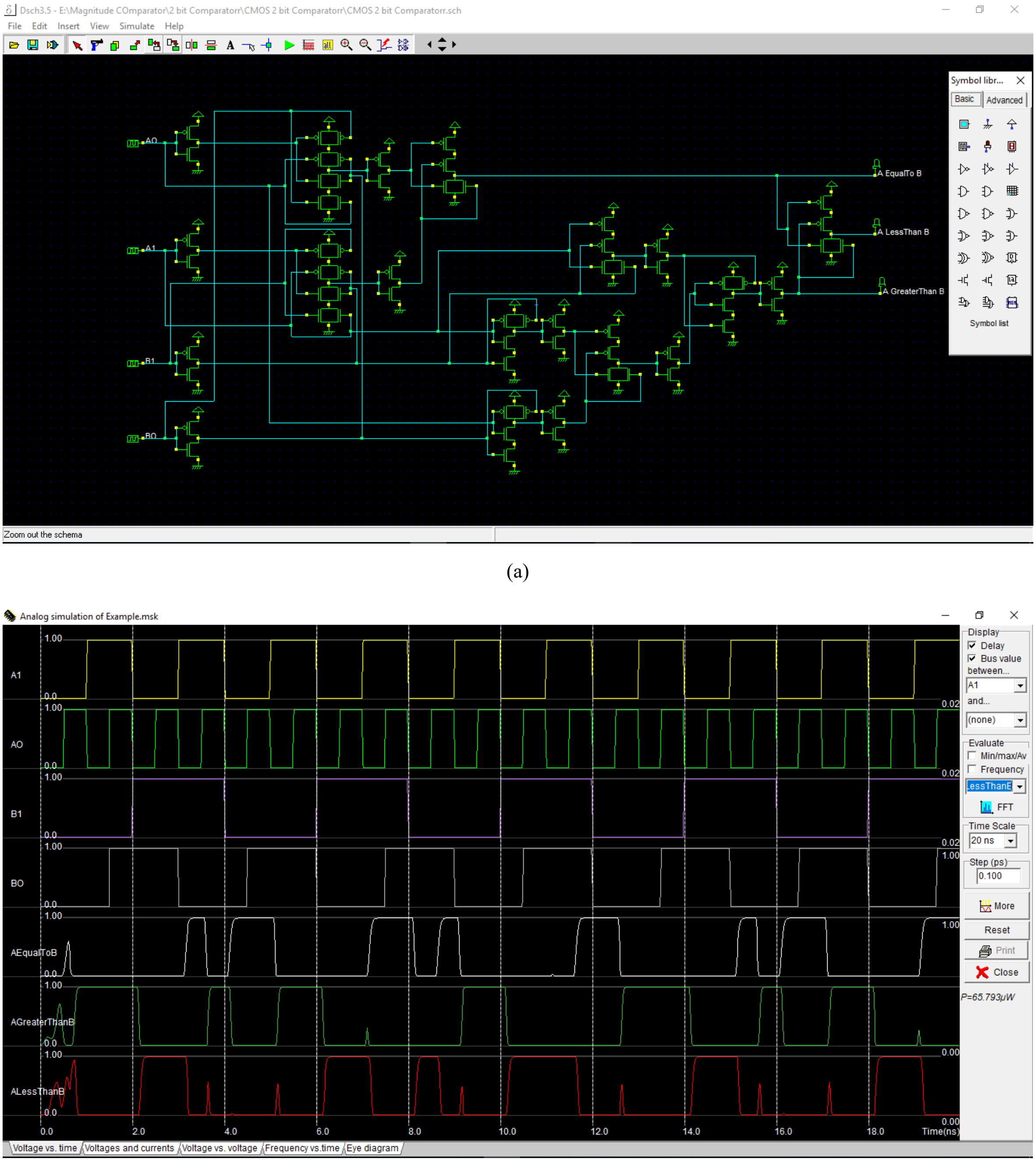


Fig. 3 Static CMOS

Static CMOS implementation of a 2-bit magnitude comparator is shown in Fig. 4 (a). The circuit is simulated using Dsch 3.5 and the input-output waveforms are shown in Fig. 4 (b).



(a)

(b)

Fig. 4 Magnitude comparator (a) CMOS implementation, (b) Input-output waveform

### *B. Pass Transistor Logic*

PTL is a popular and widely used alternative to Static CMOS. By letting the primary inputs to drive gate terminal as well as source/drain terminals, PTL strives to decrease the number of transistors needed to implement logic [7]. In Static CMOS implementation, the primary inputs drive only Gate terminals.

Fig. 5 shows a transistor level implementation of the XOR function using PTL, constructed using NMOS transistors. If the B input is high, the top transistor is turned on, copying the input A' to the output F. The bottom pass transistor is turned on and passes input A when input B is low.

PTL implementation uses fewer transistors as compared to static CMOS logic implementation which indicates less power dissipation. However, PTL has poor performance since PMOS and NMOS transistors are poor passers of logic 0 and logic 1 respectively.

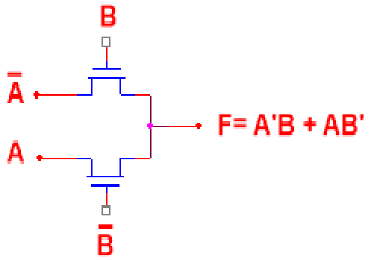
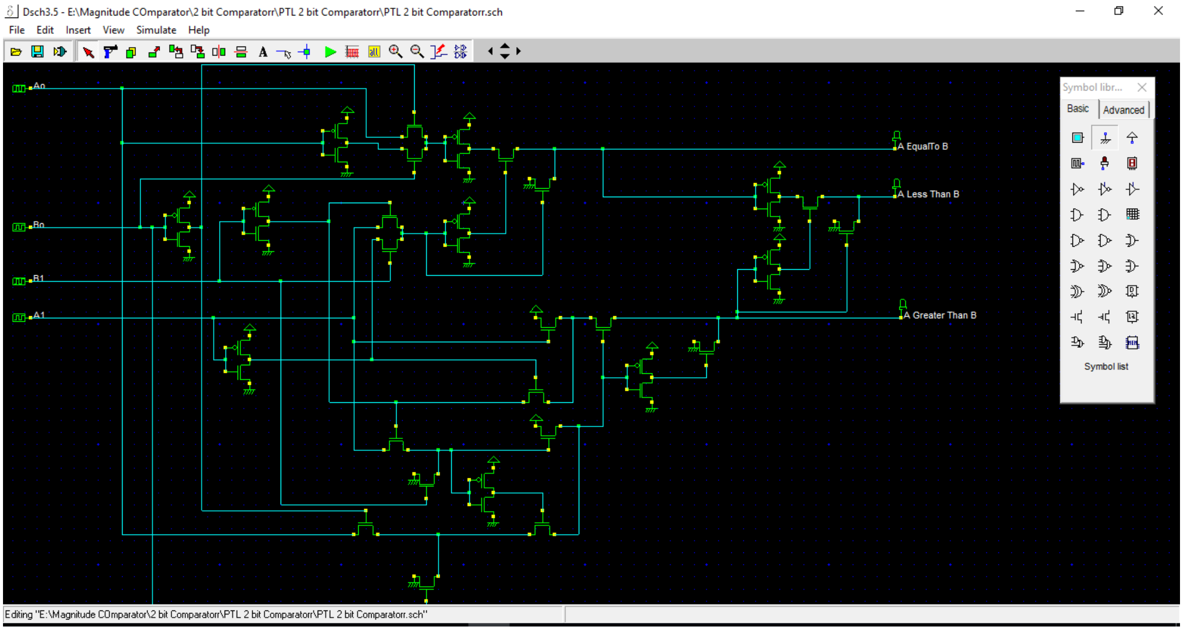


Fig. 5 PTL implementation of XOR gate

The PTL implementation of a 2-bit magnitude comparator is shown in Fig. 6 (a) and the corresponding input-output waveforms in Fig. 6 (b).

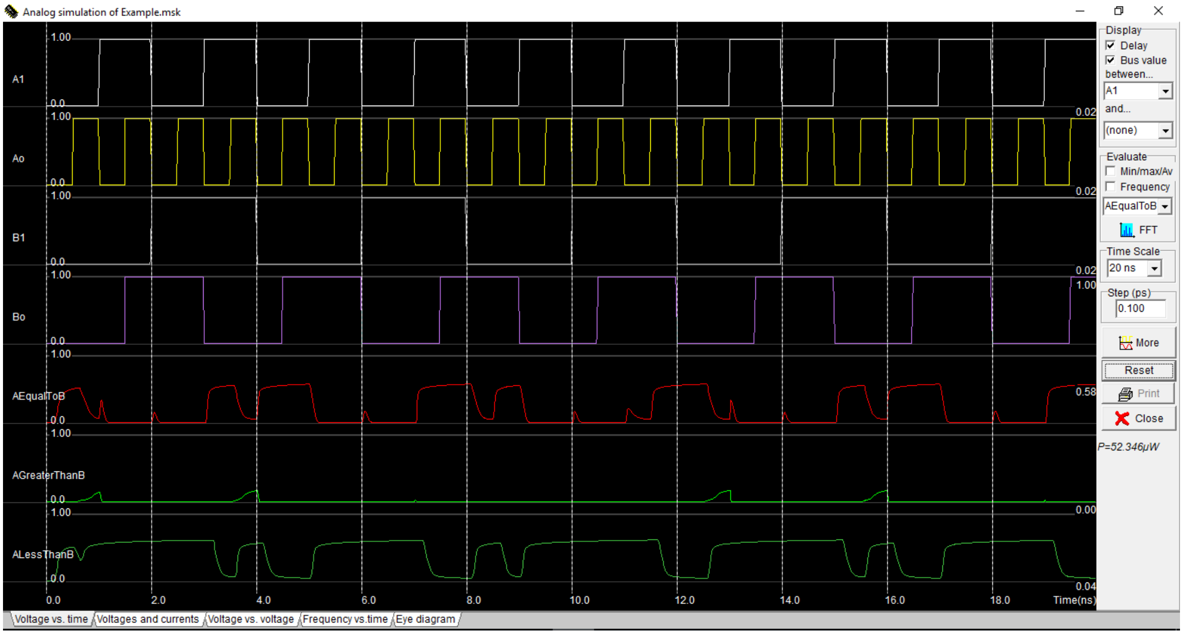
### *C. Transmission Gate (TG) Logic*

The CMOS TG consists of one NMOS and PMOS transistors, connected in parallel, as depicted in Fig. 7.


Dsch3.5 - E:\Magnitude COmparator\2 bit Comparatorr\PTL 2 bit Comparatorr\PTL 2 bit Comparatorr.sch

Editing "E:\Magnitude COmparator\2 bit Comparatorr\PTL 2 bit Comparatorr\PTL 2 bit Comparatorr.sch"

(a)


Analog simulation of Example.msk

Time Scale: 20 ns
Step (ps): 0.100
P=52.346µW

(b)

Fig. 6 Magnitude comparator (a) PTL implementation, (b) Input-output waveform

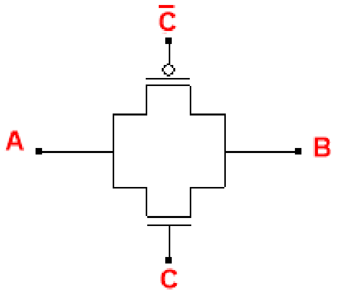
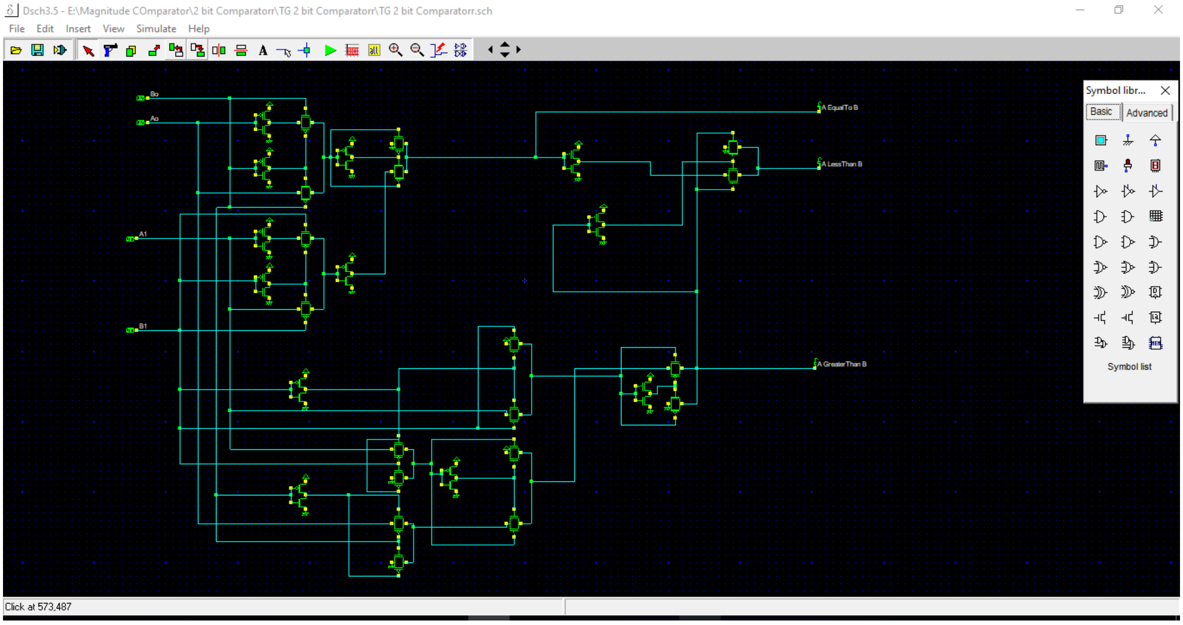


Fig. 7 Basic TG circuit

The gate voltages applied to these two transistors are set to be complementary signals. As such, the CMOS TG operates as a bidirectional switch between the nodes A and B which is controlled by signal C [8].

- Both transistors are turned on and pass A to B if the control input C is high.
- If the control signal C is low, both transistors will be switched off, and the path between A and B will be open.

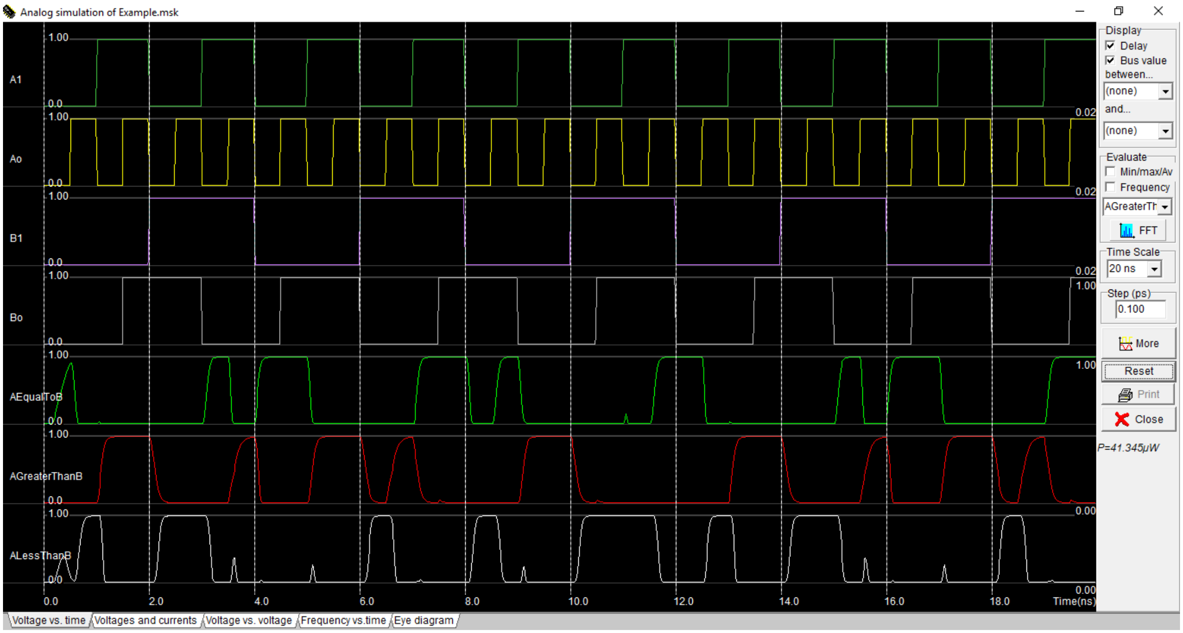
TGs are used to implement basic switching schemes and can be extended to provide advanced logic functions [9]. The TG implementation and simulation outputs are shown in Figs. 8 (a) and (b) respectively.



Dsch3.5 - E:\Magnitude COmparator\2 bit Comparator\TG 2 bit Comparator\TG 2 bit Comparatorr.sch

Click at 573,487

(a)



Analog simulation of Example.msk

Display
☑ Delay
☑ Bus value between...
(none)
and...
(none)

Evaluate
□ Min/max/Av
□ Frequency
AGreaterTh

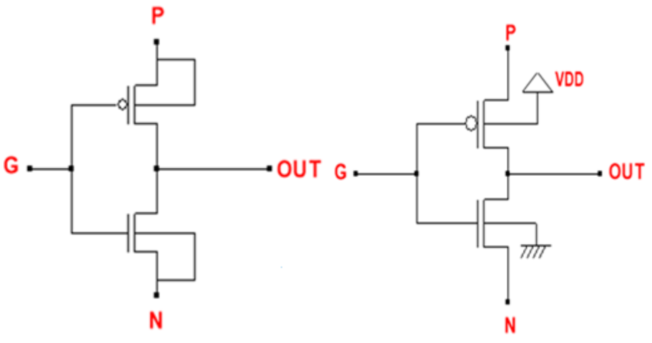
Time Scale 20 ns

Step (ps) 0.100

P=41.345µW

(b)

Fig. 8 Magnitude comparator (a) TG implementation, (b) Input-output waveform



(a) (b)

Fig. 9 GDI (a) Basic circuit (b) Basic modified circuit

### *D. GDI Logic*

Morgenshtein et al. [10] described a new design GDI cell which consists of three inputs – G (common Gate input of the PMOS and NMOS transistors), P (input to Drain/Source of PMOS), N (input to Drain/Source of NMOS). As shown in Fig. 9 (a), the basic GDI cell is very much similar to Static CMOS inverter with the exception, P and N inputs are not always necessarily be connected to VDD and GND, respectively. Since the P and N inputs can take either 0 or VDD, there is a possibility the body terminals of the PMOS and NMOS transistors being connected to 0 and VDD. This causes the threshold voltages to rise. Hence as a solution, [11] described the Modified Gate Diffusion Input (MGDI) cell that overcomes the drawback of GDI. In this Modified GDI Cell [Fig. 9 (b)], the body terminal of the PMOS is connected to the highest voltage supply (VDD) and the NMOS to the lowest voltage source (GND).

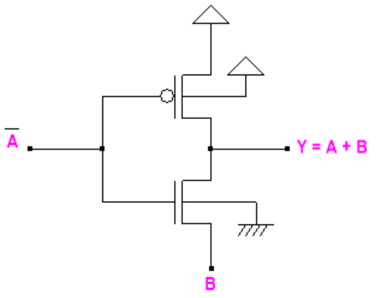
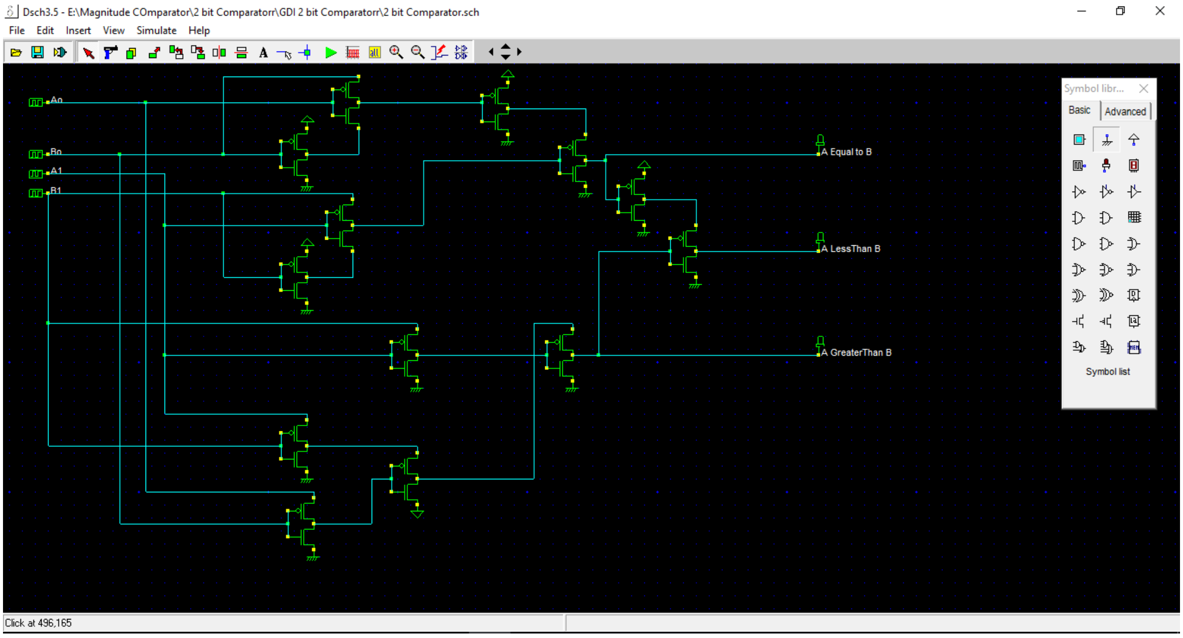


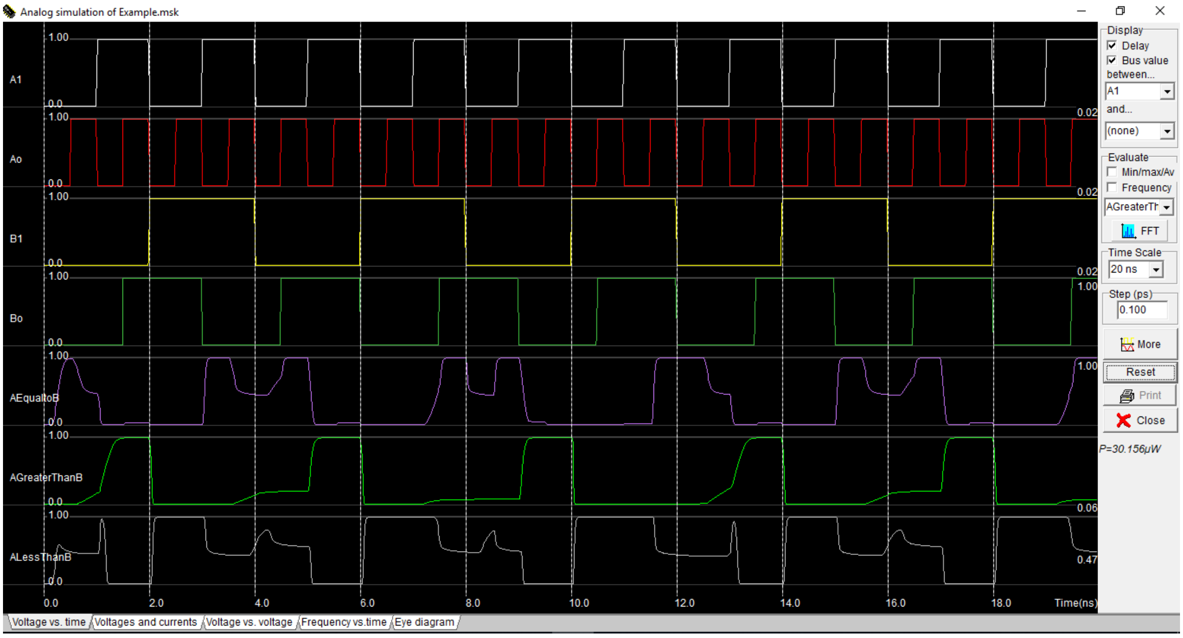
Fig. 10 GDI implementation of OR gate

TABLE II
Summary of Inputs Applied at P, G, and N Terminals to Implement Logic Gates

| | Logic Gate | | | | | | |
|---|---|---|---|---|---|---|---|
| Input Terminal | AND<br>[AB] | OR<br>[A + B] | NAND<br>[AB]’ | NOR<br>[A+B]’ | XOR<br>[A ⊕ B] | XNOR<br>[A ⊕ B]’ | NOT<br>A’ |
| P | B | $V_{DD}$ | $V_{DD}$ | A’ | B | B’ | $V_{DD}$ |
| G | A’ | A’ | A | B | A | A | A |
| N | GND | B | B’ | GND | B’ | B | GND |


Dsch3.5 - E:\Magnitude COmparator\2 bit Comparatorr\GDI 2 bit Comparatorr\2 bit Comparator.sch
File Edit Insert View Simulate Help
A0
B0
A1
B1
A Equal to B
A LessThan B
A GreaterThan B
Symbol libr...
Basic
Advanced
Symbol list
Click at 496,165

(a)



☑ Delay
☑ Bus value between... A1 and... (none)
□ Min/max/Av
□ Frequency
AGreaterTh
Time Scale 20 ns
Step (ps) 0.100
P=30.156μW

(b)

Fig. 11 Magnitude comparator (a) GDI implementation, (b) Input-output waveform

Fig. 10 shows GDI implementation of a two input OR Gate (Y = A + B). For this logic gate, G = A’, P = VDD, and N = B inputs. When A’ = 0, input P i.e. VDD (PMOS Source) is passed to output through the PMOS. But when A’ = 1, input N i.e. B (NMOS Source) is passed to output through the NMOS. If B is VDD, the output results in weak logic 1 otherwise the output will be a good logic 0 as NMOS passes a good logic 0. Similarly, other basic logic gates can be implemented by applying specific inputs at P, G, and N terminals. Table II summarizes the inputs required to realize basic logic gates using GDI implementation technique. Hence, using Table II as a reference more complex functions are built. Fig. 11 (a) shows GDI implementation of a 2-bit magnitude comparator and the simulated waveforms are shown in Fig. 11 (b).

## IV. Proposed Methodology

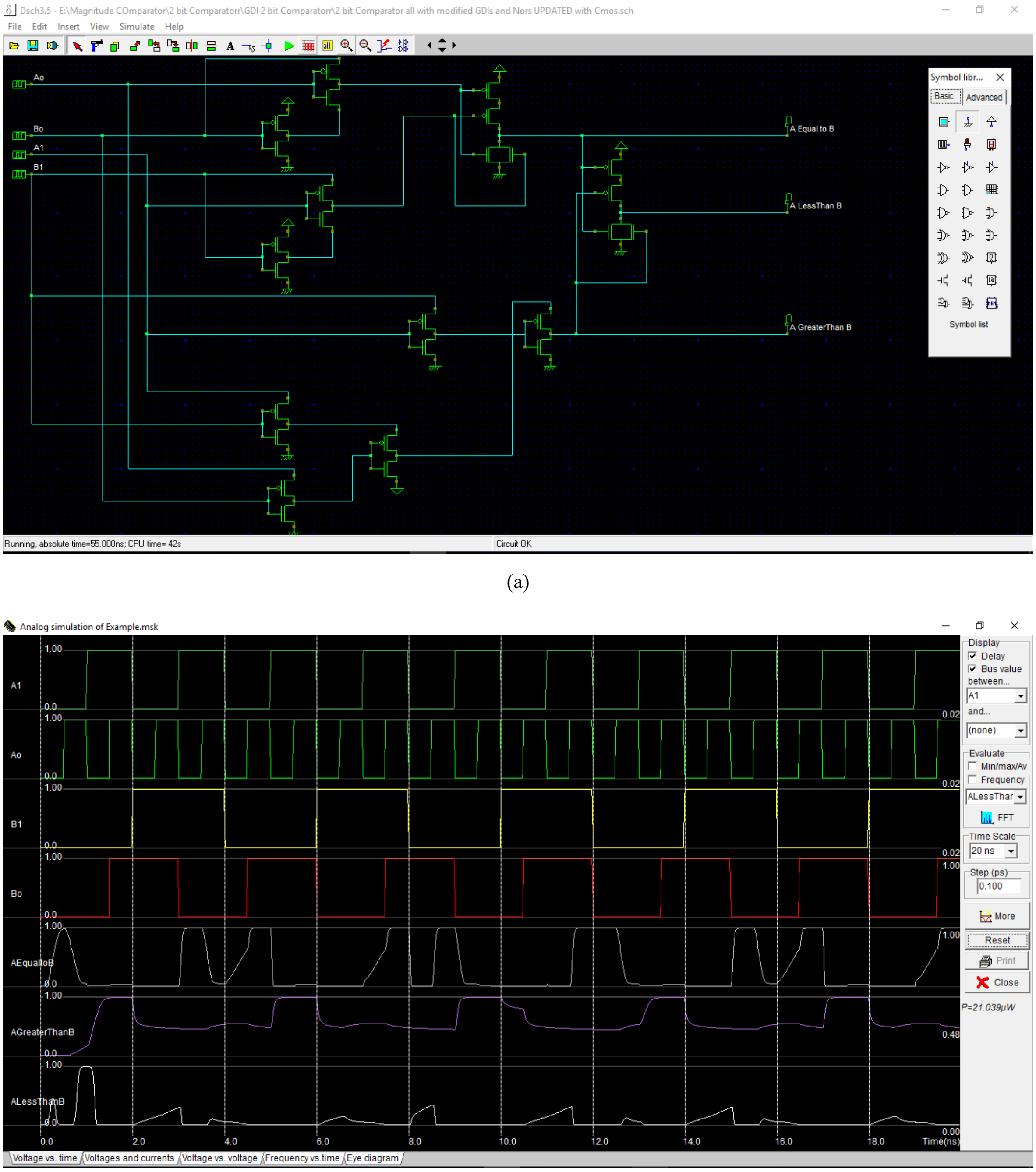
It is well known that, in realizing basic logic gates with the implementation techniques discussed above, they show much wide performance difference. Basic logic gates were realized and compared according to power dissipation, delay, and transistor count using Dsch 3.5. Table III presents the simulation results (power, delay, and transistor count) of basic logic gates in using Static CMOS, GDI, and TG implementation techniques.

As shown in Table III, GDI implementation of AND, OR, XOR, and XNOR logic gates outperforms CMOS and TG implementation techniques in terms of power, delay, and transistor count. Meanwhile, for NOR and NAND gates CMOS implementation is better than GDI and TG techniques. Hence, in applications that use both these sets of logic gates, it is recommended to implement using CMOS and GDI techniques together. In using this hybrid implementation (CMOS implementation for NOR and NAND gates together with GDI implementation for AND, OR, XOR, and XNOR gates), power consumption, delay, and layout area are significantly improved.

The proposed methodology [Fig. 12 (a)] usea both GDI (to implement AND, OR, and XOR Gates) and CMOS (to implement NOR Gate). The corresponding input-output waveforms are shown in Fig. 12 (b).

TABLE III
Power Dissipation of 2 Input Logic Gates in GDI, CMOS, TG, PTL Techniques

| Logic Gates | Power Dissipation [in µw], Propagation delay [in Ps], and Transistor Count<br>[With Supply Voltage of 1V, Input/output Voltage = 2.5V, MOS Model = BSIM 4]<br>[ Inputs used: A = 5ns clock and B = 10ns clock] | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | GDI | | | CMOS | | | TG | | |
| | Power (µw) | Delay (Ps) | Transistor Count | Power (µw) | Delay (Ps) | Transistor Count | Power (µw) | Delay (Ps) | Transistor Count |
| AND | 2.16 | 12 | 4 | 3.83 | 25.5 | 6 | 3.79 | 19.5 | 8 |
| OR | 2.90 | 28 | 4 | 4.63 | 39.5 | 6 | 6.33 | 24.5 | 6 |
| NAND | 2.34 | 7 | 4 | 1.75 | 10 | 4 | 7.77 | 13 | 8 |
| NOR | 2.32 | 15.5 | 4 | 2.16 | 15.5 | 4 | 7.78 | 25.5 | 8 |
| XOR | 0.48 | 22 | 4 | 13.87 | 71.5 | 14 | 7.87 | 27.5 | 8 |
| XNOR | 2.99 | 44 | 4 | 13.47 | 80.5 | 14 | 6.75 | 25.5 | 8 |


Dsch3.5 - E:\Magnitude COmparator\2 bit Comparator\GDI 2 bit Comparator\2 bit Comparator all with modified GDIs and Nors UPDATED with Cmos.sch
File
Edit
Insert
View
Simulate
Help
Ao
Bo
A1
B1
A Equal to B
A LessThan B
A GreaterThan B
Symbol libr...
Basic
Advanced
Symbol list

Running, absolute time=55.000ns; CPU time= 42s

Circuit OK

(a)

Display: ☑ Delay ☑ Bus value between... A1 and... (none)

Evaluate: □ Min/max/Av □ Frequency ALessThar

Time Scale: 20 ns

Step (ps): 0.100

P=21.039µW

(b)

Fig. 12 Magnitude comparator (a) proposed methodology (CMOS + GDI) implementation, (b) Input-output waveform

## V. Simulation Results and Discussion

The simulation of the proposed methodology is investigated and compared with other implementation techniques using delay, power dissipation, transistor count, PDP, and layout area as performance metrics. Tanner EDA 16.0 tools have been used to estimate delay. To compare power dissipation, transistor count, and layout area, Microwind & Dsch Version 3.5 is used.

### *A. Comparison of Implementation Techniques in 90 nm Technology and 1 V Supply Voltage*

All implementation techniques discussed were compared in terms of power dissipation, delay, PDP, layout area, and transistor count in 90 nm technology and 1 V supply voltage. The simulation data are shown in Table IV.

Even though the proposed methodology uses just as equal number of transistors and layout area as the GDI technique, it still outperforms PTL, TG, and CMOS implementation methods in terms of transistor count and layout area. However, comparing them in terms of power consumption, the proposed methodology shows an improvement of 49.11% (as compared to TG), 10.84% (as compared to GDI), 59.8% (as compared to PTL), and 68% (as compared to CMOS).

As shown in Tables IV and V, the worst-case delay of the proposed methodology among all the outputs (A Equal to B, A Greater Than B, and A Less Than B) is 2.05%, 41.6%, 15.27%, and 26.45% better than TG, GDI, PTL, and CMOS implementation techniques respectively. As a result, PDP of the proposed methodology shows an enhancement of 50.15% (as compared to TG), 47.95% (as compared to GDI), 65.95% (as compared to PTL), and 76.477% (as compared to CMOS).

TABLE IV
COMPARISON OF IMPLEMENTATION TECHNIQUES IN 90 NM TECHNOLOGY AND 1 V SUPPLY VOLTAGE

| Implementation Technique [90 nm, 1 V] | No. of Transistors used | Power Dissipation [µw] | Delay [Ps] | PDP [µw * Ps] | Layout Area [µm$^2$] |
|---|---|---|---|---|---|
| TG | 60 [30 NMOS, 30 PMOS] | 41.35 | 243.5 | 10068.7 | 73*10 = 730 |
| GDI | 26 [13 NMOS, 13 PMOS] | 23.6 | 408.5 | 9640.6 | 32*10 = 320 |
| PTL | 38 [28 NMOS, 10 PMOS] | 52.35 | 281.5 | 14736.5 | 47*11 = 517 |
| CMOS | 66 [33 NMOS, 33 PMOS] | 65.79 | 324.3 | 21335.7 | 80*11 = 880 |
| Proposed Methodology | 26 [13 NMOS, 13 PMOS] | 21.04 | 238.5 | 5018.04 | 32*10 = 320 |

TABLE V
COMPARISON OF IMPLEMENTATION TECHNIQUES IN TERMS OF PROPAGATION DELAY

| Implementation Technique [90 nm, 5 V] | Propagation Delay [Ps] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | A Equal to B | | | A Greater Than B | | | A Less Than B | | | Worst- Case Delay [Max $T_P$] |
| | Max $TP_{HL}$ | Max $TP_{LH}$ | $T_P$ | Max $TP_{HL}$ | Max $TP_{LH}$ | $T_P$ | Max $TP_{HL}$ | Max $TP_{LH}$ | $T_P$ | |
| TG | 171.2 | 163.2 | 167.2 | 268.1 | 134.1 | 201.1 | 217.4 | 270 | 243.5 | 243.5 |
| GDI | 109.2 | 376.3 | 242.8 | 244 | 188.9 | 216.4 | 167.5 | 649.5 | 408.5 | 408.5 |
| PTL | 129.5 | 110.5 | 120 | 118.3 | 228.3 | 173.3 | 383.5 | 179.5 | 281.5 | 281.5 |
| CMOS | 208 | 259 | 233.5 | 275.5 | 354.5 | 315 | 373.5 | 275 | 324.3 | 324.3 |
| Proposed Methodology | 90.5 | 360 | 225.3 | 292 | 185 | 238.5 | 186 | 285 | 235 | 238.5 |

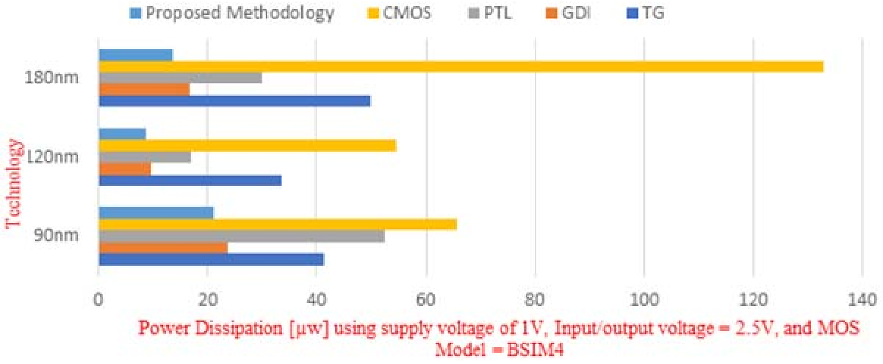


Fig. 13 Comparison of implementation techniques in terms of power dissipation in 90 nm, 120 nm and 180 nm technologies

TABLE VI
COMPARISON OF IMPLEMENTATION TECHNIQUES IN TERMS OF POWER DISSIPATION FOR VARIABLE SUPPLY VOLTAGES

| Implementation Technique [90 nm] | Power Dissipation [in µw] for Supply Voltage of: | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0.5 V | 0.75 V | 1 V | 1.2 V | 1.3 V | 1.4 V | 1.5 V | 1.6 V | 1.71 V | 2 V | 3 V |
| TG | 8.53 | 22.3 | 41.35 | 62.2 | 74.8 | 89.1 | 105 | 124 | 147 | 227 | 850 |
| GDI | 1.76 | 6.11 | 23.6 | 45.4 | 59.5 | 76.7 | 97.9 | 124 | 161 | 309 | 1541 |
| PTL | 2.75 | 14.3 | 52.35 | 105 | 148 | 207 | 284 | 381 | 511 | 995 | 5252 |
| CMOS | 14.6 | 35.5 | 65.79 | 98 | 117 | 138 | 162 | 189 | 222 | 333 | 1294 |
| Proposed Methodology | 1.4 | 5.1 | 21.04 | 43 | 57.2 | 73.9 | 93.6 | 117 | 147 | 252 | 1020 |

### *B. Comparison of Implementation Techniques in Terms of Power Dissipation in 90 nm, 120 nm, and 180 nm Technologies*

Simulations and power dissipation comparison between implementation techniques in 90nm, 120nm, and 180nm technologies using 1V (supply voltage), 2.5V (input/output voltage), and BSIM4 (MOSFET Model) was carried out. Figure. 13 shows the simulation results.

The simulation result reveals that the proposed methodology uses less power in all the technologies used (90 nm, 120 nm, and 180 nm) compared with TG, GDI, PTL, and CMOS implementations.

### *C. Comparison of Implementation Techniques in Terms of Power Dissipation under Variable Supply Voltage*

Varying the supply voltage (from 0.5 V to 3 V), power consumption of the proposed methodology and GDI, TG, PTL, and CMOS implementation techniques was investigated and the data are presented in Table VI. The proposed methodology consumes less power under 1.71 supply voltage. However, power dissipation of TG implementation become less than all the techniques used, including the proposed methodology, above 1.71 supply voltage.

## VI. Conclusions

In this paper, performance of a 2-bit magnitude has been analyzed using GDI, PTL, TG, CMOS, and the proposed method. Using GDI Method, 2-bit magnitude comparator has been implemented with fewer number of transistors as compared to Static CMOS, PTL, and TG logic styles.

GDI technique allows reducing power consumption, propagation delay, and area of digital circuits while maintaining low complexity of logic design [12]. Even though, the GDI implementation technique results in low power consumption and layout area, it gives an increased delay and reduced voltage swing at the output. A method was proposed (combining CMOS and GDI) and significantly improved the total performance of GDI technique. 10.84% (power consumption), 41.6% (propagation delay), 47.95% (PDP) improvements were achieved with this proposed methodology.

Though GDI technique offers low power, less transistor count and high speed, the major challenges occur in the fabrication process. The GDI technique requires twin-well CMOS or Silicon on Insulator (SOI) process to realize a chip which increases the complexity as well as the cost of fabrication [13].

## References

[1] M. W. Allam, New methodologies for low-power high-performance digital VLSI design, 2000.

[2] N. S. Kim et al., "Leakage current: Moore's law meets static power," in Computer, vol. 36, no. 12, pp. 68-75, Dec. 2003.

[3] J. Rabaey and A. Chandrakasan, Digital Integrated Circuits: A Design Perspective. Pearson Education, 2003.

[4] L. Xiu, VLSI Circuit Design Methodology Demystified: A Conceptual Taxonomy. Piscataway, NJ: IEEE Press, 2007.

[5] A. K. Maini, Digital electronics: principles, devices, and applications. Chichester, England: John Wiley & Sons, 2007.

[6] N. H. E. Weste and D. M. Harris, CMOS VLSI design: A circuits and systems perspective. Boston, MA: Addison-Wesley, 2011.

[7] D. Radhakrishnan, S. R. Whitaker and G. K. Maki, "Formal design procedures for pass transistor switching circuits," in IEEE Journal of Solid-State Circuits, vol. 20, no. 2, pp. 531-536, April 1985.

[8] S.-M. Kang and Y. Leblebici, CMOS digital integrated circuits: analysis and design, 3rd ed. New York: McGraw-Hill, 2002.

[9] J. P. Uyemura, Circuit design for CMOS VLSI. New York: Springer Science Business Media, LLC, 1992.

[10] A. Morgenshtein, A. Fish, and I. A. Wagner, "Gate-diffusion input (GDI) - a technique for low power design of digital circuits: analysis and characterization," 2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353), Phoenix-Scottsdale, AZ, USA, 2002, pp. I-I.

[11] P. Balasubramanian and J. Joh, "Low power digital design using modified GDI method," International Conference on Design and Test of Integrated Systems in Nanoscale Technology, 2006. DTIS 2006., Tunis, 2006, pp. 190-193.

[12] A. Morgenshtein, A. Fish and I. A. Wagner, "Gate-diffusion input (GDI): a power-efficient method for digital combinatorial circuits," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 10, no. 5, pp. 566-581, Oct. 2002.

[13] R. Uma and P. Dhavachelvan, "Modified gate diffusion input technique: a new technique for enhancing performance in full adder circuits." Procedia Technology, vol. 6, no. 2, pp. 74-81, 2012.