Analysis of GDI Technique for Digital Circuit Design

Laxmi Kumre
Assistant Professor
Electronics & Comm.Engg. Deptt.
MANIT, Bhopal (M.P.), INDIA

Ajay Somkuwar
Professor
Electronics & Comm.Engg. Deptt.
MANIT, Bhopal (M.P.), INDIA

Ganga Agnihotri
Professor
Electrical Engg. Deptt.
MANIT, Bhopal (M.P.), INDIA

ABSTRACT

Power Dissipation of Digital circuits can be reduced by 15% - 25% by using appropriate logic restructuring and also it can be reduced by 40% - 60% by lowering switching activity. Here, Gate Diffusion Input Technique which is based on a Shannon expansion is analyzed for minimizing the power consumption and delay of static digital circuits. This technique as compare to other currently used logic design style, allows less power consumption and reduced propagation delay for low-power design of combinatorial digital circuits with minimum number of transistors. In this paper, basic building blocks of digital system and few combinational circuits are analyzed using GDI and other CMOS techniques. All circuits are designed at 180nm technology in CADENCE and simulate using VIRTUOSO SPECTRE simulator at 100 MHz frequency. Comparative analysis has been done among GDI and other parallel design styles for designing ripple adder, CLA adder and bit magnitude comparator. Simulation result shows GDI technique saves 53.3%, 55.6% and 75.6% power in ripple adder, CLA adder and bit magnitude comparator respectively as compare to CMOS. Also delay is reduced with 25.2%, 3.4% and 6.9% as compare to CMOS. Analysis conclude that GDI is revolutionary high speed and low power consumption technique.

General Terms

Low power design, Digital circuit Design, VLSI, Logic Style.

Keywords

CMOS, GDI, SOI, CLA.

1. INTRODUCTION

From the day when transistor was invented in 1947, low area, low power and high speed are the primary issue for researcher in the transistor based technology. In the modern technology, low power consumption have emerged as a key design constraint over the last few years due to increasing demand of complex mobile system in the VLSI circuit design. More than ever, circuit designers are recognizing the impact of power consumption on IC performance, as it is directly linked to its reliability. The over-whelming demand for portable and mobile electronics encourages the development of a power optimized structure. Given the increasing complexity of designs, power optimization should be a conscious effort starting from the initial stages of a design, where the opportunity to save power is at a maximum. The proliferation of portable and hand-held electronics combined with increasing packaging costs is forcing circuit designers to adopt low power design methodologies. Low power design of application specific integrated circuits (ASIC) result in increased battery life and improved reliability. Indeed, the Semiconductor Industry Association technology roadmap has identified low power design techniques as a critical technological need. Hence it becomes imperative for circuit designers to acknowledge the importance of limiting power consumption and improving energy efficiency at all levels of the design hierarchy, starting from the lower levels of abstraction, when the opportunity to save power is significant.

There are three components of power dissipation in digital CMOS circuits, which are summarized as [4],

$$P_{avg.} = P_{leakage} + P_{short\ ckt.} + P_{switching}$$

Power consumption due to leakage current which is primarily determined by the fabrication technology consists of reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk regions in a MOS transistor as well as the sub-threshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage. Power consumption because of short circuit current arises due to the DC path between the supply rails during output transitions. The switching/ dynamic component of power consumption arises when the capacitive load, C_L of a CMOS circuit is charged through PMOS transistors to make a low to high voltage power consuming transition, which is usually the supply (V_{dd}). Very high power losses in CMOS circuits are dynamic losses, related to gate output transitions. Since CMOS circuits do not consume much power if they are not switching, a major focus of low power design is to reduce the switching activity or transition activity to the minimal level, required to perform the computation.

Minimization of power dissipation in CMOS based system designs can take place at four levels [5]: technology, circuit, architecture and algorithm. In this paper, this issue is addressed at the technology and circuit level for digital CMOS circuits. At the circuit level, 20 to 30 % power can be saved by choosing appropriate circuit design style [3]. In this paper several digital design circuit techniques have been explained with their advantages and disadvantages. A new low power design technique that solved most of the problems occur in previous techniques – Gate Diffusion Input Technique [1] is explained in detail with its operational and transient analysis. The GDI Technique is superior when dealing with the rising challenges of digital circuit's design [2]. Current methods are based on standard logical gates and are not compatible with the increasing demands for low power designs in the electronic industry. This technology is simple to implement, cost effective and based on multi-functional building blocks.

The aim of this work is to analyze the GDI technique by implementation of logic gates and comparing their properties with their analogues in CMOS, PTL and TG. A variety of logic gates have been implemented in 180 nm technology and results of the comparison are presented. The rest of the paper is structured as follows. Section II provides a comprehensive idea about various circuit techniques used for low power

digital circuit design, their advantages and drawbacks. Section III explains detailed analysis of new GDI technique and its advantages as compare to previous techniques. Section IV gives the design methodology of combinational circuits using GDI technique. Section V presents simulation result of all circuits designed in GDI cell in 180nm standard CMOS process and its comparative performance with respect to other circuit techniques. Section VI discuss the results and concludes the paper.

2. BACKGROUND

The various design techniques for digital integrated circuit are:

A) S*tandard CMOS design technique*: Standard CMOS circuits with complementary nMOS pull-down and pMOS pull-up networks are used for the vast majority of logic gates in integrated circuits. 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 [3]. The power consumption of conventional CMOS circuit is largely determined by the AC power caused by the charge and discharge of capacitances [3]:

$$\text{Power} = CV^2f \qquad \text{---- (1)}$$

Where f is the frequency at which a capacitance charged and discharged. As the circuits get faster, the frequency goes up as does the power consumption. CMOS design technique has relatively simple fabrication process but in order to drive wires quickly, large width transistors are needed, since the time to drive a load is given by:

$$\Delta t = C\,\Delta V / i \qquad \text{---- (2)}$$

Where Δt is the time to charge or discharge the load, C is the capacitance associated with the load, ΔV is the load voltage swing and i is the average current provided by the load driver. Typical voltage swings for standard CMOS are from 3.3 to 5 volts with even smaller swings on the way [8]. All other thing being equal, equation (2) says that a smaller voltage swing will be proportionally faster.

Fig.1. shows a CMOS inverter [1]. There is a pull-up PMOS transistor and a pull-down NMOS transistor. The steady state output of CMOS style will be independent of the ratio of pull-up and pull-down transistor sizes. Because of this, CMOS complementary logic does not have to worry about signal degradation problems in pass-transistor logic [6]. Because the power –to-ground path only closes during the transition, it almost consumes no static power. The CMOS complementary gate has two function determining blocks an n-block and a p-block. There are normally 2n transistors in an n-input gate [8].

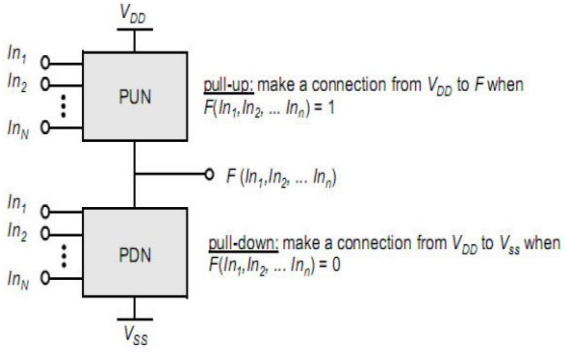


Figure.1. CMOS logic gate

B) *Pass Transistor Logic*: A popularly and widely used alternative to complementary CMOS is pass transistor logic [8], which attempts to reduce the number of transistor required to implement logic by allowing the primary inputs to drive gate terminal as well as source drain terminal. This is contrast to logic family which only allows primary inputs to drive the gate terminal of MOSFET. Figure.2. shows an implementation of the AND gate function using only nMOS transistor [3]. In this gate, if the *B* input is high, the top transistor is turned on and copies the input *A* to the output *F*. When *B* is low, the bottom pass transistor is turned on and passes *O*. The switch driven by *B*' seems to be redundant at first glance. It presence to ensure that gate is static. A low impedance path must exist to the supply rails under all circumstances (in this case, when *B* is low).

The advantage of pass-transistor logic is that it uses fewer transistors to construct complex Boolean function. The reduced number of devices has the additional advantages of lower capacitance. Also this logic style has advantages of high speed due to the small node capacitances, low power dissipation- as a result of the reduced number of transistors and lower interconnection effects due to a small area. However, most of the pass transistor logic implementations have two basic problems [8]. First, the threshold drop across the single-channel pass transistors results in reduced current drive and hence slower operation at reduced supply voltages. Secondly, an NMOS device is effective at passing a 0 but is poor at pulling a node to V_{dd}. When the pass transistor pulls a node high, the output only charges up to $V_{dd} - V_{tn}$. In fact, the situation is worsened by the fact that the devices experience body effect, as there exists a significant source-to-body voltage when pulling high.

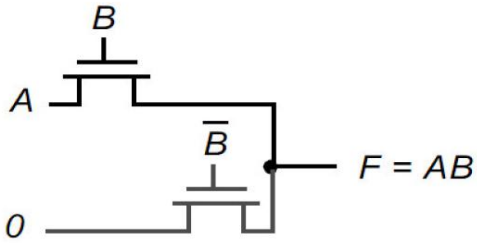


Figure.2. PTL AND gate

C) *Transmission gates*: The structure of a transmission gate is shown in figure 3. It consists of an n-channel transistor and p-channel transistor with separate gate connections and common source and drain connection. The control signal $ is applied to the gate of n- device and its complement to gate of p-device. Operation can be well explained by considering the n and p device separately. When the control signal $ is low i.e. '0' both n and p devices are off and output is high impedance. Similarly when $ is high i.e. '1' both n and p devices are on and input is transferred to output node. The transmission gate may be used as a switch to control the data flow through a static logic network. Also, it is possible to use the transmission gate as a general logic-controlled switch to synthesize complex logic functions.

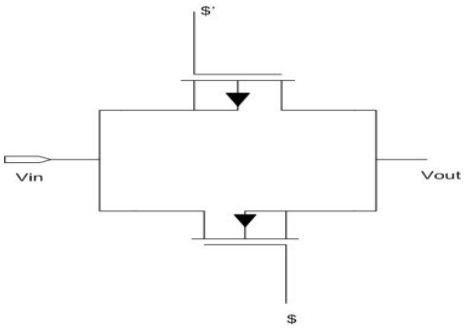


Figure.3. Transmission Gate Logic

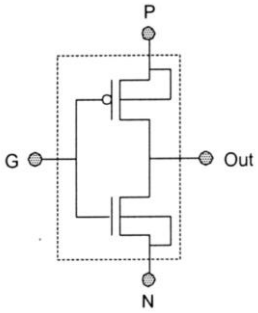


Fig.4. GDI basic cell [1]

Modern high-density, high performance chip designs constraint has led designers to question the need for using the nFET/pFET pair required in the transmission gate [11][12]. The FET itself is not a problem because of its small size. The wiring, on the other hand, can be significant, especially when transmission gates are distributed throughout a complex system layout. Owing to this consideration, many modern designs tend to move away from using transmission gates opting instead for single nFETs in their place. In principle, any transmission gates based network can be converted to using nFETs only so long as we modify the electrical characteristics where needed.

This paper analyses a new low power design technique that allows solving most of the problems mentioned in above digital design circuit techniques- Gate Diffusion Input technique (GDI). The GDI approach allows implementation of a wide range of complex logic functions using only two transistors. This method is suitable for design of fast, low power circuits, using reduced number of transistors (as compared to CMOS and existing PTL techniques), while improving logic level swing and static power characteristics and allowing simple Shannon's theorem-based design by using small cell library.

3. ANALYSIS OF GDI TECHNIQUE

The GDI method which is first proposed by A. Morgenshtein, A. Fish, and I. A. Wagner in 2001 [1], is based on the use of a simple cell as shown in figure.4. At first glance, the basic cell reminds the standard CMOS inverter, but there are some important differences:

1. The GDI cell contains three inputs: G (common gate input of nMOS and pMOS), P (input to the source/drain of pMOS), and N (input to the source/drain of nMOS).

2. Bulks of both nMOS and pMOS are connected to N or P (respectively), so it can be arbitrarily biased at contrast with a CMOS inverter.

It must be remarked that not all of the functions are possible in standard p-well CMOS process but can be successfully implemented in twin-well CMOS or silicon on insulator (SOI) technologies.

Table I: Various Logic Functions of GDI Cell

N	P	G	OUT	FUNCTION
0	B	A	A'B	F1
B	1	A	A'+ B	F2
1	B	A	A+B	OR
B	0	A	AB	AND
C	B	A	A'B + AC	MUX
0	1	A	A'	NOT

3.1 Design of GDI cell

The GDI functions given in table I is nothing but simply the extension of a single input CMOS inverter structure into a triple input GDU cell in order to achieve implementation of complicated logic functions with a minimal number of transistors. Extension of any n-input CMOS structure to an (n+ 2) input GDI cell can be done by using P as input instead of supply voltage in the pMOS block of a CMOS structure and an N input instead of ground in the nMOS block. This extended implementation can be represented by the following logic expression [13]:

$$Out = F'(x_1 \ldots\ldots\ldots\ldots x_n)P + F(x_1 \ldots\ldots\ldots\ldots x_n)N$$

Where $F(x_1 \ldots\ldots x_n)$ is a logic function of an nMOS block not of the whole original n-input CMOS structure. The above equation is based on Shannon expansion, where any function *F* can be written as follows:

$$F(x_1 \ldots\ldots x_n) = x_1 H(x_2 \ldots\ldots\ldots x_n) + x_1' G(x_2 \ldots\ldots x_n)$$

$$= x_1 F(1, x_2 \ldots\ldots x_n) + x_1' F(0, x_2 \ldots\ldots x_n)$$

The output functions of basic GDI cell shown in Table I are based on Shannon expansion where A, B and C are inputs to G, P and N respectively as,

$$OUT = AC + A'B$$

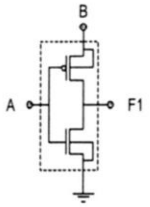
This fact makes a standard GDI cell very suitable for implementation of any logic function that was written by Shannon expansion.

Shannon expansion is a very useful technique for pre-computation based low-power design in sequential logic circuits, due to its multiplexing properties [14]. Hence, GDI cells can be successfully used for low-power design of combinatorial circuits, while combining two approaches - Shannon expansion and combinational logic pre-computation, where transitions of logic values are prevented from propagating through the circuit if the final result does not change as a result of those transitions. Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

3.2 Operational Analysis of GDI cell

We have discussed various circuit techniques currently used for digital design in Section II. The most common problem of all design methods is the low swing of output signals due to the threshold drop across the single-channel pass transistors. Generally to overcome this problem, additional buffer circuit is used. In GDI cell, the effects of low swing problem can be understood by operational analysis of F1 function and it can be easily extend to other functions of GDI cell. Table II shows a full set of logic states and their related functionality modes of F1.

Table II: Input Logic States versus Functionality F1



A	B	Functionality	F1
0	0	*pMOSTrans Gate*	V_{Tp}
0	1	*CMOSInverter*	1
1	0	*nMOS Trans Gate*	0
1	1	*CMOSInverter*	0

From the table, it can be seen that in half of the cases (B =1), the GDI cell operates as a regular CMOS inverter, which is widely used as a digital buffer for logic-level restoration. In the cases, when V_{dd}=1, without a swing drop from the previous stages, a GDI cell works as an inverter buffer and recovers the voltage swing but the only state where low swing occurs in the output value is A = 0, B= 0. In this case, the voltage level of F1 is V_{Tp} instead of expected 0 volt because of the poor high-to-low transition characteristics of the pMOS transistor [6]. Among all the possible transitions, the only case where the effect of low swing occurs is the transition from A = 0, B= V_{dd} to A=0, B=0.

The GDI cell allows a self-swing restoration in certain cases, but the worst case is also assumed in this analysis and additional circuitry is used for swing restoration in the implemented circuits.

3.3 Switching Characteristics

The complexity of the logic function can be implemented in a GDI cell by using only two transistors. So, it is important to *perform a comparison of its switching characteristics with* CMOS gate, whose logic function is of the same order of complexity. This comparison can be used as a base for delay estimation in early stages of circuit design, if GDI or CMOS design techniques are considered. While a GDI cell's characteristics are close to a standard inverter, the gate with equivalent functional complexity in CMOS will be NAND. The switching behavior of the inverter can be generalized by examining the parasitic capacitances and resistances associated with the inverter [12] [16]. Consider the inverter and a NAND gate with a series connection of identical n-channel MOSFETs shown in Fig. 6 with their equivalent digital models.

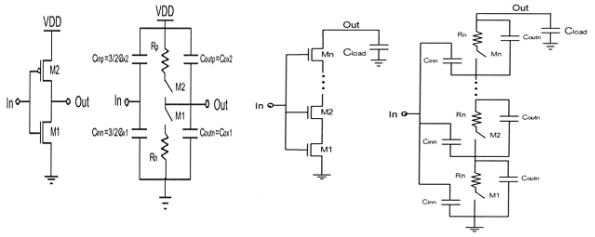


Fig. 5. CMOS inverter and series MOSFET with their equivalent digital model

The propagation delay for an inverter [3] driving a capacitive load is

$$t_{PHL} = R_n . C_{tot} = R_n . (C_{out} + C_{load})$$

Where C_{tot} is the total capacitance on the output of the inverter. The intrinsic switching time of series connected MOSFET with an external load capacitance [12] can be estimate as

$$t_{PHL} = N. R_n. (C_{out} / N + C_{load}) + 0.35. R_n . C_{inn} (N -1)^2$$

The first term represents the intrinsic switching time of the series connection of *N* MOSFETs, while the second term represents RC delay caused by R_n charging C_{inn} .

For C_{inn} = 3/2.C_{ox} and assuming two serial n-MOS transistors, the propagation delay in NAND is

$$t_{PHL} = 1.52 . R_n . C_{out} + 2. R_n . C_{load}$$

Therefore, the delay of a NAND gate compared to a GDI gate is approximated by

$$1.52 \leq [t_{PHL(CMOS)} / t_{PHL(GDI)}] \leq 2$$

Where the high bound is for high C_{load} and the low bound is for low C_{load}.

Note that this ratio will become better if the effect of the body source diode in a GDI cell [1] is considered and the delay formula is used in its improved form.

4. DESIGN METHODOLOGY FOR GDI DIGITAL CIRCUITS

In this paper, GDI technique has been analyzed by designing basic digital gates and few combinational circuits such as ripple adder, carry look ahead adder and comparator for 4 bit binary numbers at 180nm technology using CADENCE EDA VLSI TOOL. The performance of GDI is also measured in high level digital combinatorial circuits. For analysis purpose half adder, full adder, ripple adder, carry look ahead adder and comparator were also implemented using GDI and CMOS design techniques. Half Adder and Full adder are designed using XOR, AND and OR gate combination.

Ripple adder [17] shown in figure 6, is logical circuit to add n-bit numbers using multiple full adders. Each full adder inputs a C_{in}, which is the C_{out} of the previous adder. This kind of adder is called a *ripple-carry adder*, since each carry bit

"ripples" to the next full adder. The ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder.

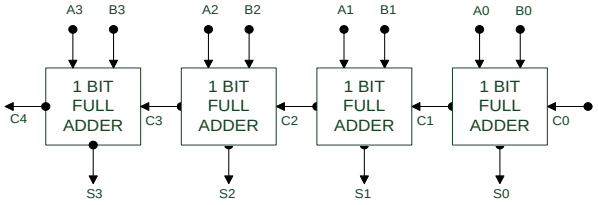


Figure 6: 4 bit Ripple Adder

To reduce the computation time, the faster way is to add two binary numbers by using ***carry look ahead adder*** [17] shown in figure 7. It work by creating two signals *P* and *G* for each bit position, based on whether a carry is propagated through from a less significant bit position (at least one input is a '1'), generated in that bit position (both inputs are '1'), or killed in that bit position (both inputs are '0'). In most cases, *P* is simply the sum output of a half-adder and *G* is the carry output of the same adder. After *P* and *G* are generated the carries for every bit position are created.

The carry of the *i*th stage C may be expressed as

$$C_i = G_i + P_i.C_{i-1}$$

Where $G_i = A_i . B_i$ ……. generate signal

$P_i = A_i \oplus B_i$ …… propagate signal

The sum S_i is generated by

$S_i = A_i \oplus B_i \oplus C_{i-1} = P_i \oplus C_{i-1}$

For 4 bit carry look ahead adder, the four stages of carry generated signals are

$C_0 = G_0 + P_0 C_{in}$

$C_1 = G_1 + P_1 G_0 + P_1 P_0 C_{in}$

$C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{in}$

$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}$

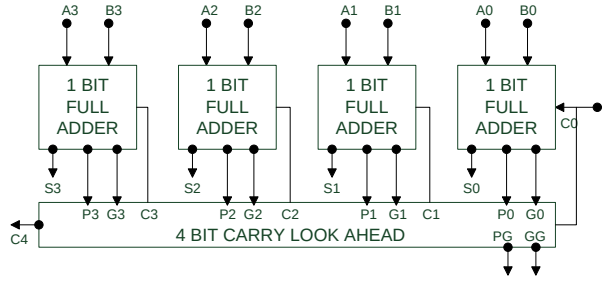


Figure 7: Carry-Look-Ahead Adder

A ***magnitude comparator*** [17] shown in figure 8, is a combinational circuit that compares two numbers A and B and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that include whether A > B, A = B, or A < B. Consider two numbers A and B with four digits each. The coefficients of the numbers with descending significant as follows:

$A = A_3A_2A_1A_0$

$B = B_3B_2B_1B_0$

The two numbers are equal if all pairs of significant digits are equal i.e., if $A_3 = B_3$, $A_2 = B_2$, $A_1 = B_1$ and $A_0 = B_0$. When the numbers are binary, the digits are either 1 or 0 and the equality relation of each pair of bits can be expressed logically with an equivalence function:

$$X_i = A_iB_i + A'_iB'_i \qquad i= 0,1,2,3.$$

Where $X_i = 1$ only if the pair of bits in position i are equal.

To determine whether A is equal, greater than or less than B, the sequential comparison can be expressed logically by the following Boolean functions:

$(A=B) = X_3X_2X_1X_0$

$(A>B) = A_3B'_3 + X_3A_2B'_2 + X_3X_2A_1B'_1 + X_3X_2X_1A_0B'_0$

$(A<B) = A'_3B_3 + X_3A'_2B_2 + X_3X_2A_1B_1 + X_3X_2X_1A'_0B_0$

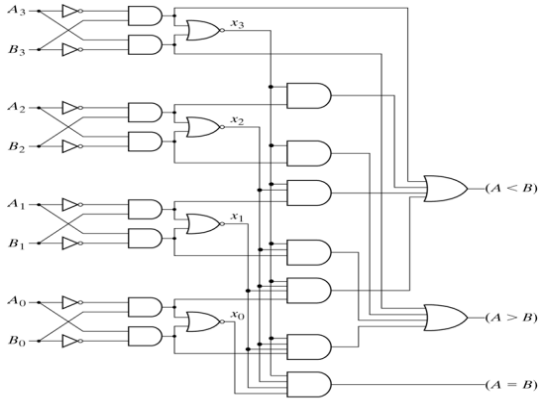


Figure 8: Magnitude Comparator

5. SIMULATION RESULTS

All the basic gates and combinatorial circuits using CMOS, NPG, TG and GDI techniques are simulated in CADENCE VIRTUOSO SPECTRE with 1.8v input voltage supply and at 50MHz frequency. The W/L ratios of both nMOS and pMOS transistors are taken as 540nm/180nm for better power delay performance. To establish an unbiased testing environment, the simulations have been carried out using a comprehensive input signal pattern, which covers every possible transition. The basic gates AND, OR, XOR has been designed and compared using GDI, CMOS, N-PG & TG techniques. The circuit design and comparative analysis are shown in Table III and IV.

The performance evaluation is made with respect to switching delay, transistor count and average power consumed by GDI and other logic design styles. From the analysis it is observed that the GDI performance is better when comparing to CMOS in terms of power consumption and delay, also the number of transistors are very less. In some cases, TG and NPG gates shows less delay compare to GDI, but the power consumption and no. of transistors are very less. Hence, the overall performance of GDI is better than its other parallel design styles.

Wishing to cover a wide range of possible circuits, design methods, and properties comparisons, several digital combinatorial circuits were implemented using GDI and CMOS design techniques, and technology processes. Figure 10 shows circuit implementation for half adder, full adder, ripple adder, carry look ahead adder and comparator using GDI technique implemented during the research with respect to design methods and processes.

Table III: circuits of basic gates in CMOS, NPG, TG and GDI technique

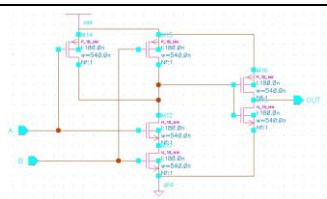
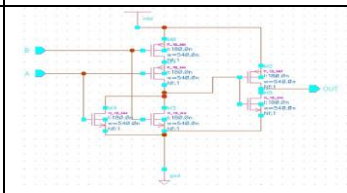
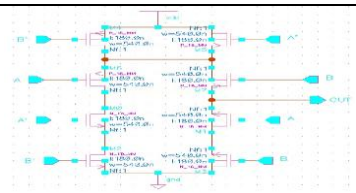
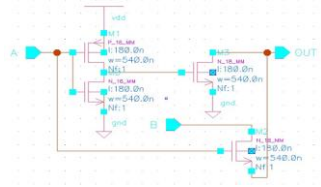
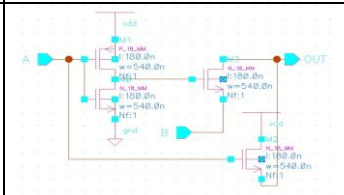
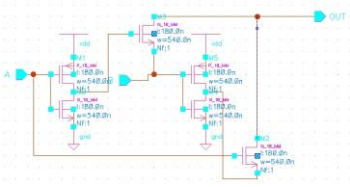
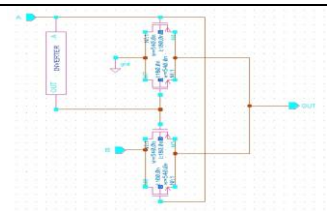
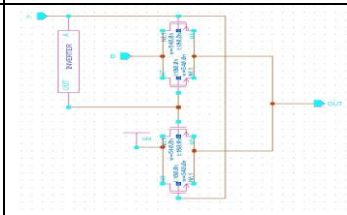
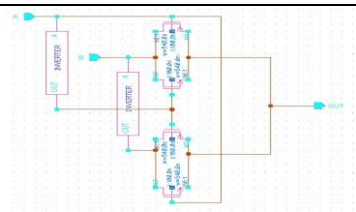
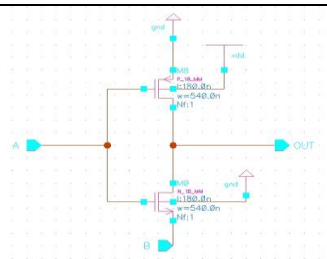
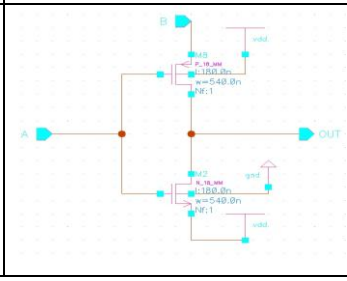
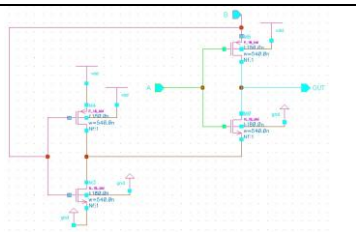
	AND	OR	XOR
CMOS			
N-PG			
TG			
GDI			

Table IV: performance Analysis of GDI and other logic design styles

Gate type	GDI			CMOS			TG			N-PG		
	Power (µW)	Delay (nsec)	No. of Xsistor	Power (µW)	Delay (nsec)	No. of Xsistor	Power (µW)	Delay (nsec)	No. of Xsistor	Power (µW)	Delay (nsec)	No. of Xsistor
AND	0.149	4.948	2	1.459	4.95	6	1.904	4.914	6	1.472	4.928	4
OR	0.123	0.110	2	2.307	0.120	6	2.887	0.104	6	1.727	0.102	4
XOR	0.935	0.016	4	1.671	0.022	12	2.937	0.013	8	2.229	0.104	6

The performance evaluation is made with respect to switching delay, transistor count and average power consumed by GDI and other logic design styles. From the analysis it is observed that the GDI performance is better when comparing to CMOS in terms of power consumption and delay, also the number of transistors are very less. In some cases, TG and NPG gates shows less delay compare to GDI, but the power consumption and no. of transistors are very less. Hence, the overall performance of GDI is better than its other parallel design styles.

Wishing to cover a wide range of possible circuits, design methods, and properties comparisons, several digital combinatorial circuits were implemented using GDI and CMOS design techniques, and technology processes. Figure 10 shows circuit implementation for half adder, full adder, ripple adder, carry look ahead adder and comparator using GDI technique implemented during the research with respect to design methods and processes.

All the circuits are designed at 180nm CMOS technology using GDI and CMOS logic design style. The performances of

GDI circuits have been analyzed in terms of power dissipation, switching delay, transistor count, PD and AT values. The term PD and AT represent product of power – delay and product of area – delay. The parameter AT can be calculated by multiplying the transistor count and delay value. It is observed that designing digital circuits using GDI technique have 16.81% delay reduction and 62.01% less power dissipation as compared to CMOS technique. For high speed digital circuits GDI gives better performance. The

reason for this is that GDI cell uses very less number of transistors as compare to CMOS technique for designing any digital circuits such as GDI full adder uses only 14 transistor where as CMOS full adder designed with 42 transistors. Less transistors results less switching and hence less power dissipation and less delay in any circuits. Only in some circuits GDI needs swing restoration circuits to improve its output voltage level and it can achieve by simply adding inverter after GDI cell wherever it is required. Table IV shows comparative performance of GDI and CMOS based digital circuits in terms of power dissipation, switching delay, transistor count, PD and AT values.

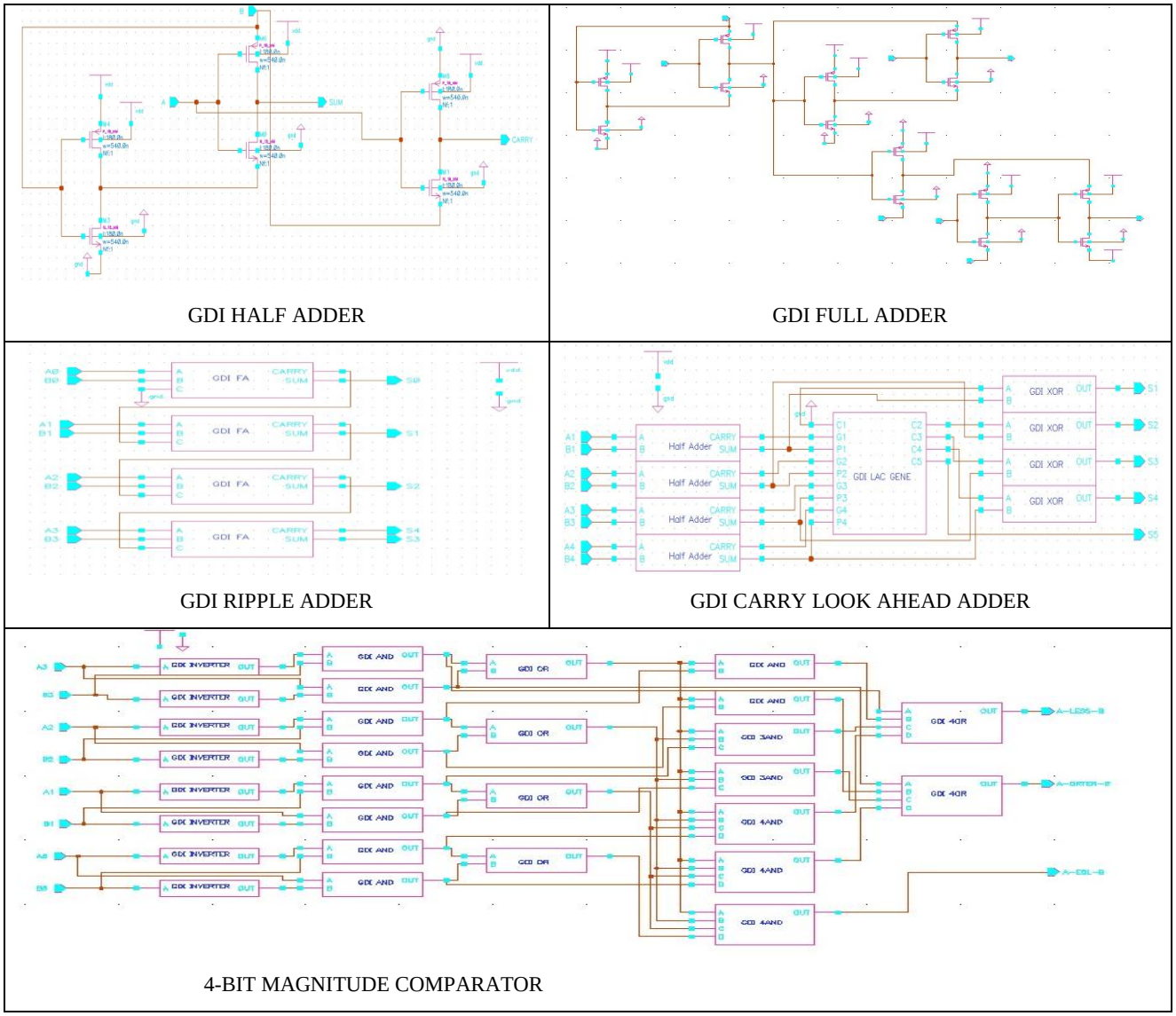


Figure 9: Shows GDI Implementation for Half Adder, Full Adder, Ripple Adder, Carry Look Ahead Adder and Comparator

Table V: Comparative Performance Analysis of GDI and CMOS Based Digital Circuits

Gate type	GDI					CMOS				
	Power (µW)	Delay (nsec)	No. of Xsistor	Power delay product	AT= Area X Time	Power (µW)	Delay (nsec)	No. of Xsistor	Power delay product	AT= Area X Time
Half Adder	1.084	0.0163	6	0.017	0.0981	3.131	0.0229	20	0.071	0.4582
Full Adder	4.287	4.795	14	20.556	67.13	21.73	5.019	42	109.062	210.798
Ripple Adder	46.33	65.02	56	3012.37	3641.12	99.22	87.02	84	8634.12	7309.68
CLA Adder	46.25	3.010	100	139.21	301	104.3	3.118	300	325.207	935.4

Comparator	19.97	59.45	78	1187.21	4637.1	82.02	63.86	214	5237.79	13666.0

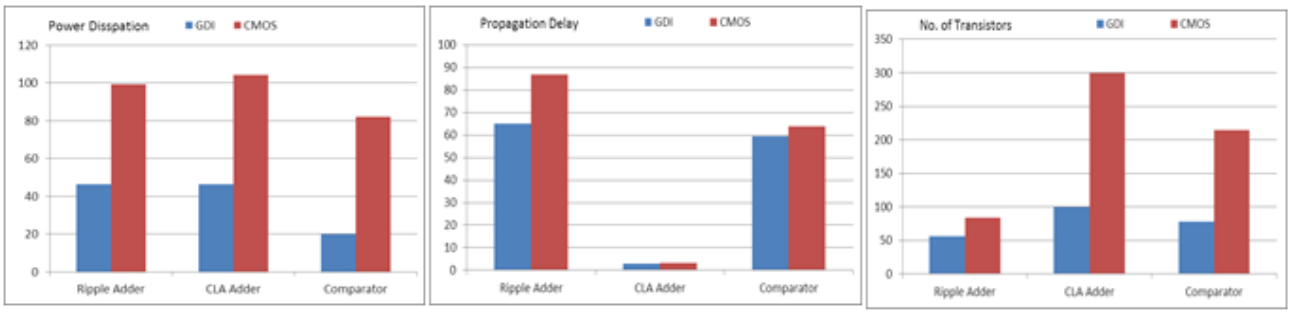


Figure 10: Comparison Graph of Delay, Transistor Count and Power Dissipation between GDI and CMOS Circuits

6. CONCLUSION

Six different digital combinational circuits are designed using AND, OR and XOR gates. Their performances have been analyzed in GDI and CMOS techniques are reported in section IV. The comparison of these circuits was made in terms of power dissipation, switching delay, transistor count, PD and AT values and it is reported in table IV. The comparison of delay, transistor count and power dissipation is depicted in figure 11.

In table V , GDI represent Gate Diffusion Input techniue and CMOS represent Complementary Metal Oxide Semiconductor technique. From the above graphs it is concluded that CMOS technique consumes more power and more delay in its design as compare to GDI technique. To implement full adder, CMOS require 42 transistor wheras GDI needs only 14 transistors that is very less and this only leads to less power dissipation and less delay as compare to CMOS technique. So from this analysis it can be observed that the digital circuits implemented with GDI cell is superior than other CMOS techniques in terms of power dissipation, switching delay, transistor count, PD and AT values and the overall simulation results proved it.

In table IV , GDI represent Gate Diffusion Input techniue and CMOS represent Complementary Metal Oxide Semiconductor technique. From the above graphs it is concluded that CMOS technique consumes more power and more delay in its design as compare to GDI technique. To implement full adder, CMOS require 42 transistor wheras GDI needs only 14 transistors that is very less and this only leads to less power dissipation and less delay as compare to CMOS technique. So from this analysis it can be observed that the digital circuits implemented with GDI cell is superior than other CMOS techniques in terms of power dissipation, switching delay, transistor count, PD and AT values and the overall simulation results proved it.

7. REFERENCES

[1] Arkadiy morgenshtein, Alexander fish & Israel Wagner, "Gate Diffusion input (GDI): A power efficient method for digital combinatorial circuits", IEEE Transaction on very large scale integration (VLSI) systems vol.10, no. 5 October 2002.

[2] Arkadiy Morgenshtein, Idan Shwartz and Alexander Fish, "Gate Diffusion Input (GDI) Logic in Standard CMOS Nanoscale Process " 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel

[3] N.H.E.Weste, David Harris Ayan Banerjee, "CMOS VLSI design, Pearson Education Publication, Sixth Impression, 2008

[4] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low- power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.

[5] A. P. Chandrakasan and R.W. Brodersen, "Minimizing power consumption in digital CMOS circuits," *Proc. IEEE*, vol. 83, pp. 498–523, Apr. 1995.

[6] W. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya, "Pass-transistor logic design," *Int. J. Electron.*, vol. 70, pp. 739–749, 1991.

[7] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design," *IEEE J. Solid-State Circuits*, vol. 31, pp. 792–803, June 1996.

[8] V. Adler and E. G. Friedman, "Delay and power expressions for a CMOS inverter driving a resistive-capacitive load," *Analog Integrat. Circuits Signal Process.*, vol. 14, pp. 29–39, 1997.

[9] J. R. Burns, "Switching response of complementary symmetry MOS transistor logic circuits," *RCA Rev.*, vol. 25, pp. 627–661, Dec. 1964.

[10] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.

[11] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1079–1090, June 1997.

[12] R. J. Baker, "CMOS: circuit design, layout, and simulation," *IEEE Press Series on Microelectronic Systems.*

[13] E. Shannon and W. Weaver, "*The Mathematical Theory of Information"*. Urbana-Champaign: University of Illinois Press, 1969.

[14] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 426–435, Dec. 1994.

[15] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, pp. 584–593, Apr. 1990.

[16] J. P. Uyemura, *Circuit Design for CMOS VLSI*. Norwell, MA: Kluwer Academic, 1992, pp. 88–129.

[17] M.Morris Mano " *Digital Logic and Computer Design*", Pearson Education Publication, Indian reprint