Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 7, No. 4, August, 2023

A Unified Power-Delay Model for GDI Library Cell Created Using New Mux Based Signal Connectivity Algorithm

Jebashini Ponnian [1*], Senthil Pari [2], Uma Ramadass [3], Chee Pun Ooi [2]

[1] *Department of Electrical and Electronics Engineering, Universiti Infrastruktur Kuala Lumpur, Malaysia.*

[2] *Faculty of Engineering, Multimedia University, Jalan Multimedia, Selangor, 63100, Cyberjaya, Malaysia.*

[3] *Department of Computer Science, Pondicherry University, Puducherry, India.*

Abstract

The challenges of innovative IC technology typically come with various new design constraints in terms of circuit implementation, behaviour, scaling, and an accurate power-delay model to evaluate the circuit's performance. The circuit realization technique using GDI is gaining popularity because of its power and transistor utilization factors. Considering the core advantage of the GDI technique, this research presents the creation of new GDI library cells implemented using the MUX-based algorithm and its delay-power model. This research defines two goals; the former goal depicts the proposal of GDI library cells with full swing using a MUX-based signal connectivity model, and the later presents the mathematical delay-power model for the proposed GDI library cells. The number of attributes defined in the delay and power model incorporates minimum variables without sacrificing precision. It calculates the delay for simple RC networks and combinational circuits with multiple paths. The power model is given using the node activity factor and the power factor related to the internal node capacitances, wiring, and gate capacitances of the driving and receiving GDI nodes. The experimental results of this study, which conform to the specifications of the sub-micron library supported for the SilTerra 130 nm 6-metal layer fabricated for the CMOS n-well process, demonstrate that the proposed GDI library is indeed superior in terms of delay-transistor and power utilisation to PTL and CMOS technology. The simulation results reveal that there is 55 to 65 % improvement in terms of power and delay factor with the existing CMOS and PTL logic. The proposed delay model demonstrates that GDI cells require less logical effort than CMOS technology. The proposed power model shows that the node activity factor of the proposed GDI cells lies between 0.1 and 0.2, while in CMOS, it is between 0.1 and 0.3.

Keywords:

MUX-Based Connectivity;
Gate Diffusion Technique;
Logical Effort;
Power Model;
GDI Library.

Article History:

Received:	20	December	2022
Revised:	15	June	2023
Accepted:	24	June	2023
Available online:	12	July	2023

1- Introduction

Generally, a delay model can be represented as one, two, or three regions [1, 2] based on the operation of a MOSFET. In the one-region model, the transistor is replaced with an equivalent resistor so that the operation of the transistor is defined in a single region (linear). However, this model is deemed inaccurate since it fails to include the slope of the input transition. For the two-region model, the transistor functions in linear saturation states. The accuracy is improved by incorporating the input transition slope parameter in the model equation. This change reflects a slight accuracy compared to the one-region model at the expense of a more complex equation that includes a three-curve fitting parameter. Finally, the three-region model (cut-off, linear, and saturation) is reported to provide better accuracy while incorporating the velocity saturation effect and channel mobility factor with high levels of the complex equation and 10 curve fitting parameters.

* **CONTACT**: jebashini@iukl.edu.my

DOI: http://dx.doi.org/10.28991/ESJ-2023-07-04-022

A predictive delay analysis reported [3, 4] the empirical fitting parameters of the MOSFET device. This model includes velocity and mobility parameters, neglecting the load and gate-drain coupling capacitors. This analytical approach provides better accuracy and independently lacks the device and model parameters. A physics-based analytical model [5-7] is a surface-inversion charge potential approach. The defined model linearizes surface-inversion charge density factors to improve accuracy with high complex equations and parameters. The model proposed in Sutherland et al. [8] is compact and the fastest candidate to estimate the delay. This approach describes the delay model in terms of logical effort, parasitic effect, and the load capacitance that drives the logic gate. The delay calculation, independent of the technology, characterizes the delay parameters in resistances and capacitances but fails to capture the velocity saturation effects.

The power model [9–11] is a compact analytical approach that comprises the output load capacitance but lacks short channel effects. The model [12, 13] includes an alpha-power-law accounting for the nominal current flowing through the PMOS and NMOS transistors. The influence of internal parasitic capacitances has been included in the model. However, this approach utilizes technology-dependent empirical parameters. This model [14, 15] uses an alpha-power law that incorporates the short channel effects of MOSFETS but fails to include gate-drain and gate-source capacitances. The various power and delay models are reported by [16–23], which lacks to provide parasitic and internal capacitance effects.

1-1- Problem Identification

From the perspectives of the above-mentioned works, the problems identified are: The delay and power models are technology-dependent empirical parameters, more complex, including more curve fitting parameters, fail to include gate-drain and gate-source capacitances, and miscarries to capture the velocity saturation effects. These shortfalls are eliminated in the proposed work by aiming to develop the delay and power model as simple and technology independent.

1-2- Aim, Goal and Objectives

The principle aim of this work is to develop the delay and power model for the proposed GDI library, keeping the constraint that the model should be simple, compact, and accurate. From the perspectives of the above-mentioned works, the delay model [8] is considered as basis for deriving the delay of a GDI circuit using a logical-based approach. The model is developed for un-skewed and skewed gates in single- and multi-stage networks. The delay calculation for a simple RC network and a multi-path combinational circuit is done. The power model [12] is considered as the basis to evaluate the dynamic power dissipation by considering the assumption of a zero-delay gate model where the gate delay and the glitches due to transitions are ignored so that the model becomes simple and compact. The power model is described using two components, namely the node activity factor and the power factor related to internal node capacitances, wiring, and gate capacitances of driving and receiving GDI nodes. This research defines two goals; the former goal depicts the proposal of GDI library cells with full swing using a MUX-based signal connectivity model, and the later presents the mathematical delay-power model for the proposed GDI library cells. The number of attributes defined in the delay and power model incorporates minimum variables without sacrificing precision.

2- Rudiments of GDI Logic Technique

GDI (Gate Diffusion Input) is a new technique [24–29] (logic family) that resembles a CMOS inverter design, and it consists of a series of connected pMOS and nMOS with shorted gate input. In GDI, at the drain terminal of pMOS, the upper region is tied to the P-diffusion, while the lower region of nMOS, at the source terminal, is connected to the N-diffusion input instead of the power rail and ground. Such a topology facilitates and accommodates a greater number of logics with fewer transistors, leading the GDI technology to gain more popularity. The structural representation of GDI is shown in Figure 1.

A logic function implementation in the GDI includes a true and complementary network where the control signal is linked with a series associated with n and p transistor switches (gate terminal). The diffusion of each MOSFET is connected with a true literal ground or power source. Commonly, the switching function in the GDI technique to realize any logic function can be represented as:

$$\mathbb{Z} = V_{P\_diff} \bullet \bar{T} + V_{N\_diff} \bullet T \quad (1)$$

where $\bar{T}$ is constructed by p-MOSFET, and the n-MOSFET transistor realizes T.

The foremost issue in GDI is the threshold discrepancy owing to bulk terminals. Preferably, the charge upshot of gate and bulk terminals is principally lower when contemplating the diffusion of source-drain and bulk terminals. Nevertheless, the oxide-related capacitance (gate and source-drain) will directly connect the gate substrate and source/drain regions. This leads to partial swing output. This problem is surrogated by fabricating GDI cells in SOI CMOS technology or using proper swing restoration logic like incorporating buffers and keeper circuits with an additional penalty of transistor count.

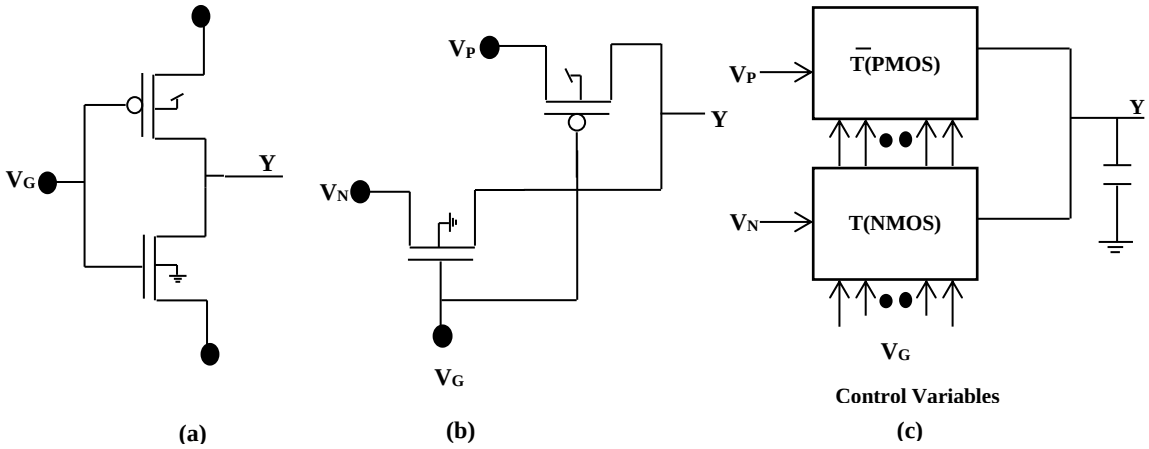


Figure 1. (a) Basic GDI cell using inverter structure (b) alternate basic GDI cell representation using PTL (c) General block diagram of GDI logic

3- Signal Connectivity Model for GDI

Any Boolean countenance in GDI logic technology is realized using Shannon's decomposition theorem to factorize the output variable Z following one of its primary input variables. Consider the illustration of a 2-input AND gate, say $Z = X \cdot Y$, apply Shannon's decomposing to X,

$$\mathbb{Z} = \overline{X}.0 + X.Y \tag{2}$$

The above expression X defines the control variable connected to the shorted P and N-MOSFET transistor. The Y input is tied to the n-MOSFET of the diffusion region, and ground (GND) is routed to the P-MOSFET diffusion region. The basic OR structure realization and its characteristics are depicted in Figure 2.

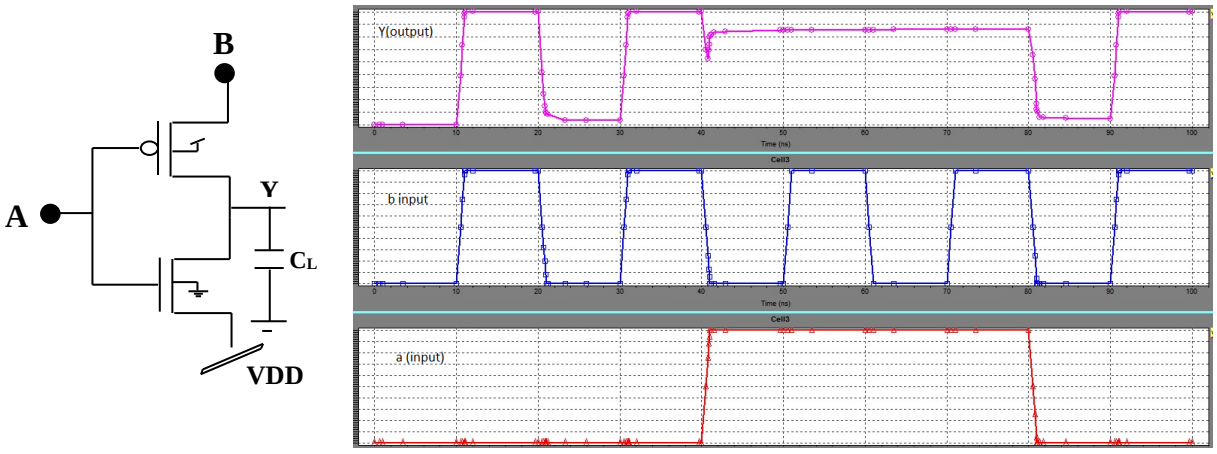


Figure 2. (a) GDI OR gate realization (b) I/O characteristics

The logic-high or logic-low (Y input signal) connected at the diffusion region deteriorates in the P and N-MOSFETs because of its threshold variation (body effect). A buffer or level restoration (like a keeper circuit) must be supplemented at the output node to acquire a complete full swing. Boolean logic in GDI is obtained by changing the inputs of N and P-MOSFET diffusion and gate input, and its depiction is presented in Table 1. The Boolean implementation involves only an extension of the CMOS NOT circuit with 3-inputs (gate, N-MOSFET diffusion, and P-MOSFET diffusion) to accommodate additional logic realization with fewer transistors.

Any Switching function in GDI logic utilizes 2-to-1 MUX with changes in the P and N-MOSFET gates and diffusion regions. This structural realization is also known as a multiplexer-tree (MUX-based). For any Boolean function implementation, the input control variable of the multiplexer must be connected to a couple of the N and P-MOSFETs, which reduces the requirement of an inverter as in the case of an nMOS-based implementation. Nevertheless, the input signal variable may deteriorate while passing the multiplexer-tree towards the output node owing to the inherited characteristics. Therefore, for every output node, a buffer or level restoration is required for full swing output.

Table 1. Gate realization in GDI Technique

N-diff	P-diff	Gate control	Logic output	Gate realization
$\Upsilon 1$	*VDD*	$\Upsilon 2$	$\overline{\Upsilon 1} + \Upsilon 2$	*F2*
GND	$\Upsilon 1$	$\Upsilon 2$	$\overline{\Upsilon 1}\Upsilon 2$	*F1*
$\Upsilon 1$	*GND*	$\Upsilon 2$	$\Upsilon 1\Upsilon 2$	*AND*
VDD	$\Upsilon 1$	$\Upsilon 2$	$\Upsilon 1 + \Upsilon 2$	*OR*
$\overline{\Upsilon 1}$	*VDD*	$\Upsilon 2$	$\overline{\Upsilon 1 \bullet \Upsilon 2}$	*NAND*
GND	$\overline{\Upsilon 1}$	$\Upsilon 2$	$\overline{\Upsilon 1 + \Upsilon 2}$	*NOR*
$\Upsilon 1$	$\overline{\Upsilon 1}$	$\Upsilon 2$	$\Upsilon 1 \odot \Upsilon 2$	*XNOR*
$\overline{\Upsilon 1}$	$\Upsilon 1$	$\Upsilon 2$	$\Upsilon 1 \oplus \Upsilon 2$	*XOR*
W	$\Upsilon 1$	$\Upsilon 2$	$\overline{\Upsilon 2}\Upsilon 1 + \Upsilon 2 W$	*MUX*
GND	*VDD*	$\Upsilon 2$	$\overline{\Upsilon 2}$	*NOT*

The complex function implementation in GDI logic involves Shannon's decomposition until the leaf cell in the GDI network will have the residue of logic 0 or 1, or any true literal value. This research proposes a new signal connection model based on the multiplexer's characteristics for the GDI technique. The signal connectivity via MUX-based construction and BDD (Binary Decision Diagram) is explained in Ponnian et al. [24]. In this work, the MUX-based algorithm is illustrated through an optimized library of primitive cells constructed to illustrate the delay and power models. The MUX-based GDI connectivity model is presented in Figure 3.

```
MUX Mapping Algorithm (GDI)
Algorithm for Any Gate with 2 inputs and 1 output
MUX (gate output, control input1, diffusion connect input2)

  Step 1: Consider variables, // A and B are the control input1 and diffusion
          input2 respectively.
           X – control input1
           Y – diffusion input2
           P-difn, N-difn – drain diffusion of PMOS and source diffusion of
           NMOS
           Z – gate output

  Step 2: Assign,
           X<- Control Signal
           Y<- Select Signal

  Step 3: Construct 2x2 matrix with complement Y and Y as row, P-difn and N-
          difn as column. Map the truth table output of the corresponding gate
          in the constructed matrix

  Step 4: Check for conditions,

            Step 4a: If (P-difn, Y) = 1 and (P-difn,Y^c) = 1, then P-difn ← 1
            Else If (P-difn, Y) = 1 and (P-difn,Y^c) ≠ 1, then P-difn ← Y
            Else If (P-difn, Y) ≠ 1 and (P-difn,Y^c) = 1, then P-difn ← Y^c
            Else If (P-diff, Y) ≠ 1 and (P-diff,Y^c) ≠  1, then P-diff ← 0
            Step 4b: If (N-difn, Y) = 1 and (N-difn,Y^c) = 1, then N-difn ← 1
            Else If (N-difn, Y) = 1 and (N-difn,Y^c) ≠ 1, then N-difn ← Y
            Else If (N-difn, Y) ≠ 1 and (N-difn,Y^c) = 1, then N-difn ← Y^c
           Else If (N-difn, Y) ≠ 1 and (N-difn,Y^c) ≠1, then N-difn ← 0

  Step 5: Construct GDI realization with X, P-difn and N-difn values derived
          at step 4a and 4b respectively.

  Step 6: Return Z
```

Figure 3. MUX mapping algorithm for GDI cell implementation

The MUX-based tactic is constructed using a k-map implementation of the Boolean function. Any m-variable GDI logic can be constructed using (m-1) primitive GDI cells. The algorithm approach consists of constructing a 2x2 K-map

with a P-diffusion and an N-diffusion along the column and an input variable as a complementary and true value. An m-variable Boolean logic in GDI is converged as 2-input GDI primitive cells. Therefore, for a 2-input primitive gate, one input is connected as the control input of MUX (connected across the gate-shorted input terminals of pMOS and nMOS), and the other variable is linked based on the column entities of P-diffusion and N-diffusion. When the column literals of P-diffusion are (1,1) or (0,0), VDD or GND is tied to the P-diffusion. Similarly, if column literals of N-diffusion are (1,1) or (0,0), then the VDD or GND is tied to the N-diffusion. Presumes on the P and N-MOSFET diffusion regions will have column literals of (0, 1) or (1, 0) its equivalent row literal value is linked to the P-diffusion or N-diffusion node.

The considered realization of the three-input XOR function in GDI technology is shown in Figure 4. Initially, the 2×2 K-map is created between inputs X and Y. The variable X is then assigned as a control input tied across the shorted gate input nodes of pMOS and nMOS. The second variable, Y, is then tied to negated and non-negated values across the row. The P-diffusion and N-diffusion are connected along the column side. The two-input XOR function truth table (output) is also mapped in the constructed 2×2 cell. In the column side of P-diffusion contains (0,1), P-diffusion is tied with the Y variable, and N-diffusion contains (1,0); therefore, N-diffusion is connected with the complementary of the b variable. The subsequent factorization is initiated between the output Z and the third input W. The output Z becomes an input for the next stage, which acts as a control variable for the second stage of the GDI cell. As in the first stage, the truth table is implemented for the literal Z and W. For this input, the k-map is formulated for literal W, and its corresponding P-diffusion & N-diffusion are tied depending on the constraints specified in the algorithm. The complete GDI gates' signal connectivity using the proposed MUX-based algorithm is shown in Figure 5.

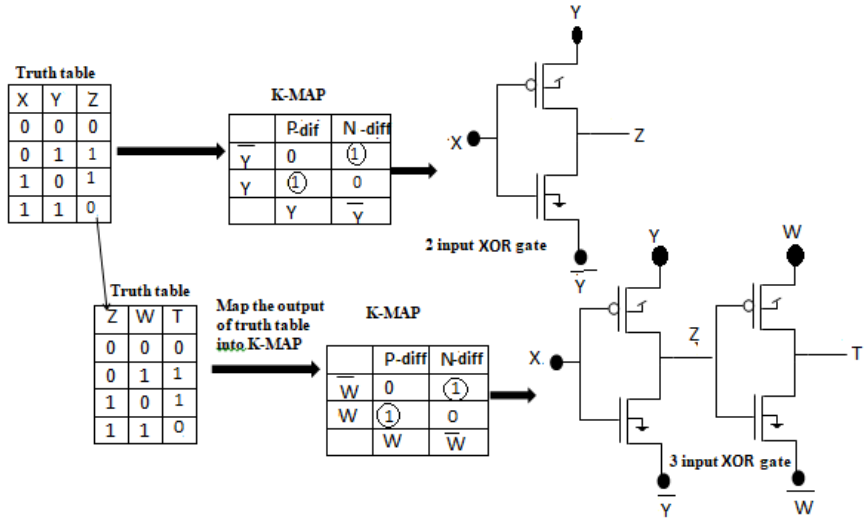


Figure 4. Implementation of 3-input XOR gate in GDI logic

Gate	Truth Table	MUX Based Mapping	Symbolic representation	Circuit Realization In GDI
GDI_AND	Υ1 Υ2 Z 0 0 0 0 1 0 1 0 0 1 1 1	P_difn N_difn $\overline{Υ1}$ 0 0 Υ1 0 1 0 Υ1	**GDI AND**	GND, y2, Z, y1
GDI_OR	Υ1 Υ2 Z 0 0 0 0 1 1 1 0 1 1 1 1	P_difn N_difn $\overline{Υ1}$ 0 1 Υ1 1 1 Υ1 1	**GDI OR**	y1, y2, Z, VDD
GDI_NAND	Υ1 Υ2 Z 0 0 1 0 1 1 1 0 1 1 1 0	P_difn N_difn $\overline{Υ1}$ 1 1 Υ1 1 0 1 $\overline{Υ1}$	**GDI NAND**	VDD, y2, Z, $\overline{y1}$

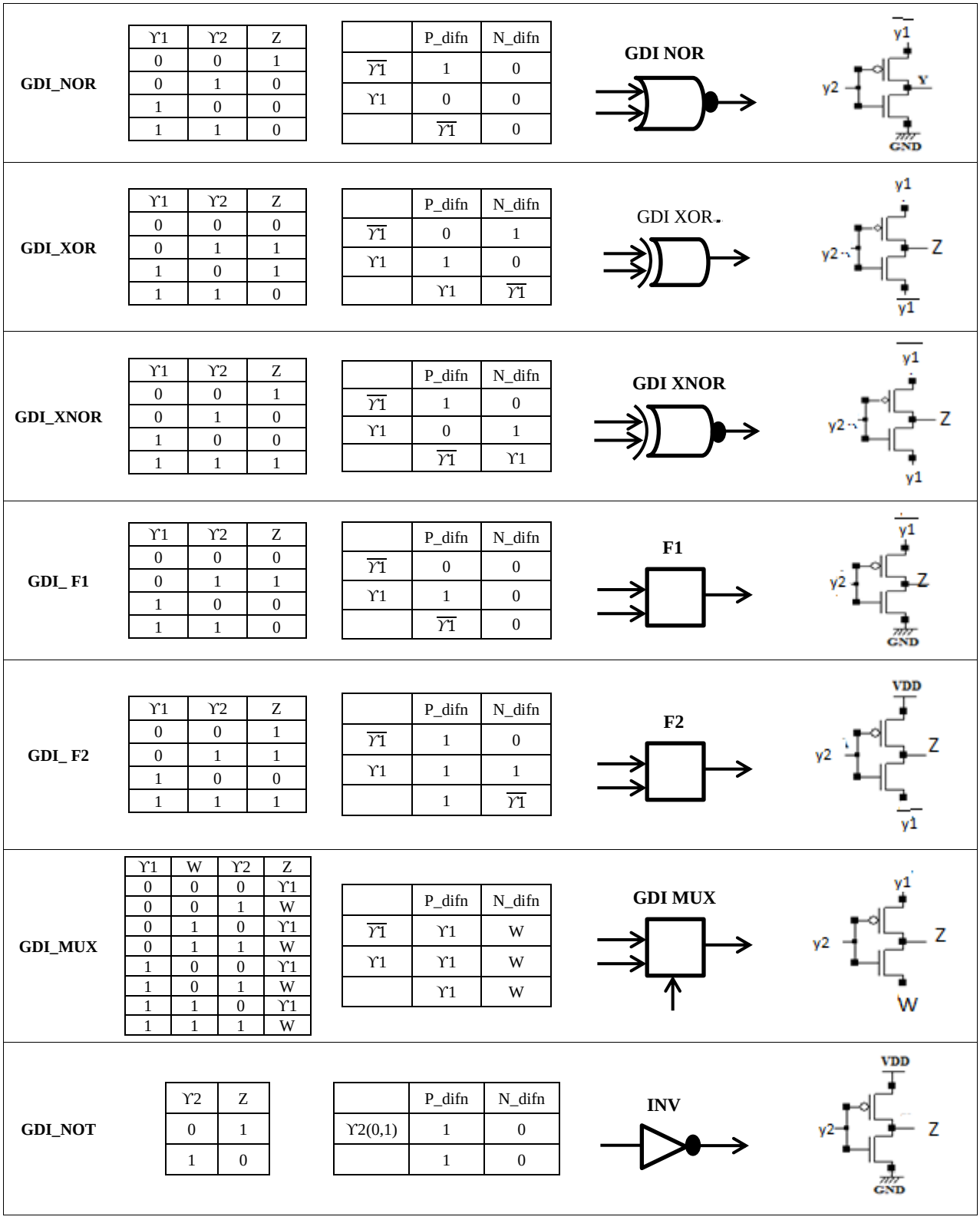


Figure 5. The primitive GDI function implementation using MUX approach

4- GDI Library Cells Creation

Various patterns for the GDI primitive cells are generated using the proposed Mux-based algorithm. The complete implementation strategy and its characteristics for each input transition are explained in Ponnian et al. [24]. The following section explains various primitive GDI cell implementations.

4-1- GDI AND Gate

For AND gate, eight patterns have been generated. The first pattern is implemented with the Boolean function $\mathbb{Z} = \overline{\Upsilon 2} \bullet \Upsilon 1$ the circuit is realized using inverter and F1 gates. The second AND gate are constructed using NAND and

inverter $\mathbb{Z} = \overline{\overline{Y1}.Y2}$, the third circuit is built with $\mathbb{Z} = Y1.Y2$ GDI AND gate. The fourth AND gate is implemented with complemented GDI OR gate with the logic expression as $\mathbb{Z} = \overline{Y2} + \overline{Y1}$, the fifth structure is comprehended using GDI MUX as $\mathbb{Z} = \overline{Y2}.Y2 + Y2.Y1$. The sixth AND formation is done using GDI MUX with the expression as $\mathbb{Z} = \overline{Y2}.\overline{Y2} + Y2.\overline{Y1} \Rightarrow \overline{Y2} + Y2.\overline{Y1} \Rightarrow \overline{Y2} + \overline{Y1}$, The seventh function is implemented using inverter and F2 function $\mathbb{Z} = \overline{\overline{Y2} + \overline{Y1}}$, finally the last AND gate is realized using complemented NOR structure having the Boolean expression of $\mathbb{Z} = \overline{Y2} + \overline{Y1}$. The structure implementation is illustrated in Figure 6.

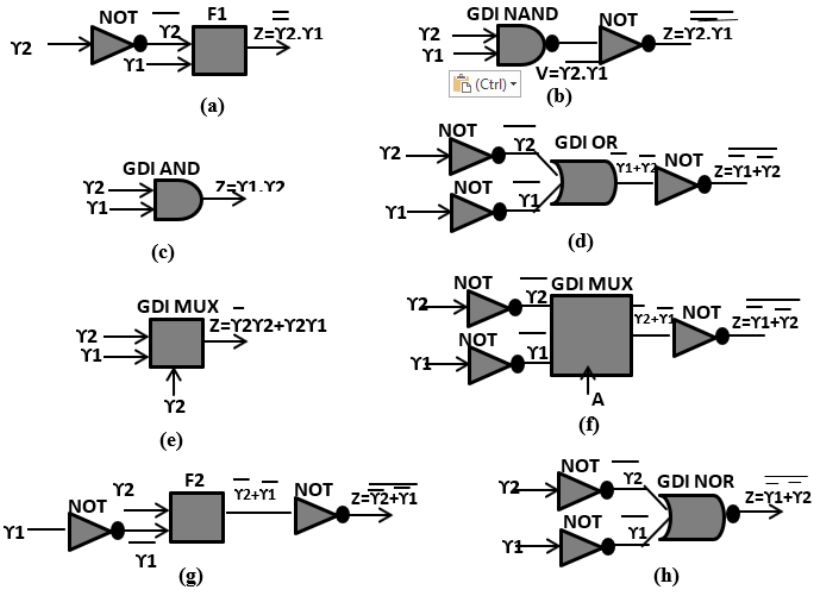


Figure 6. **Various AND patterns**

4-2- GDI OR Gate

For the realization of OR gate 8 patterns have been generated and illustrated in Figure 7. The first OR is built with an inverter-F1-inverter structure with the logic expression as $\mathbb{Z} = \overline{Y1} \bullet \overline{Y2}$. The second and third structure is implemented using GDI NOR-inverter and GDI OR $\mathbb{Z} = Y1 + Y2$. The fourth and fifth OR gate is constructed using complemented AND gate with inverter and GDI MUX with Boolean as $\mathbb{Z} = \overline{Y1}.\overline{Y2}$ and $\mathbb{Z} = \overline{Y2}.Y1 + Y2.Y2$. The sixth and seventh OR function is realized using GDI MUX-inverter and inverter-F2 function with logical expression as $\mathbb{Z} = \overline{Y1}.\overline{Y2} + Y2.\overline{Y2}$ and $\mathbb{Z} = \overline{Y2} + Y1$. The final OR is structure using complemented NAND gate as $\mathbb{Z} = \overline{Y1}.\overline{Y2}$.

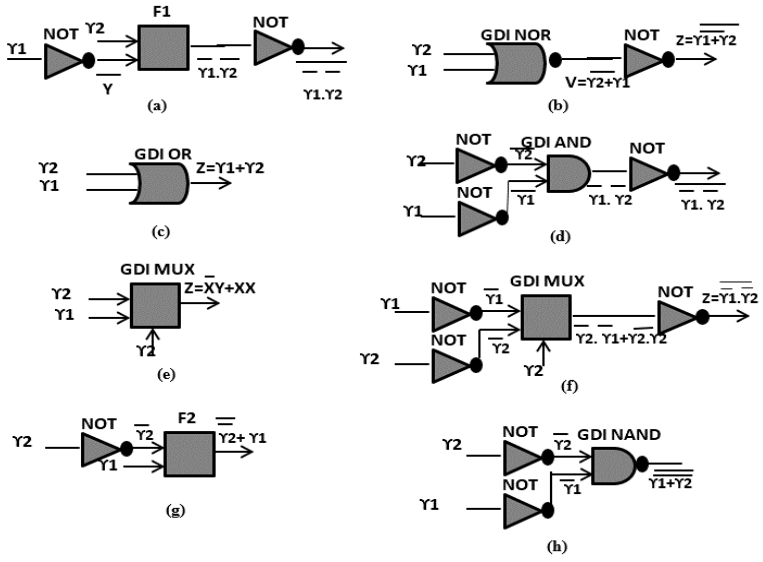


Figure 7. **Various OR patterns**

4-3- GDI NAND Gate

Eight structures of NAND gate have been proposed and presented in Figure 8. The initial and second pattern is proposed with inverter-F1-inverter and GDI AND-inverter with logic expression $\mathbb{Z} = \overline{Y2}.Y1$ and $\mathbb{Z} = \overline{Y1}.Y2$. The third and fourth OR gate is constructed using GDI NAND and complemented GDI OR with Boolean expression as $\mathbb{Z} =$

$\overline{Y1.Y2}$and $\mathbb{Z} = \overline{Y1} + \overline{Y2}$. The fifth and sixth OR gate is realized using GDI MUX with logical expression as $\mathbb{Z} = \overline{Y2}.Y2 + Y2.Y1$ and $\mathbb{Z} = \overline{Y2}.\overline{Y2} + Y2.\overline{Y1}$. The seventh and eighth structure is implemented using inverter-F2 and complemented NOR-inverter with the expression as $\mathbb{Z} = \overline{Y2} + \overline{Y1}$and $\mathbb{Z} = \overline{\overline{Y2} + \overline{Y1}}$.

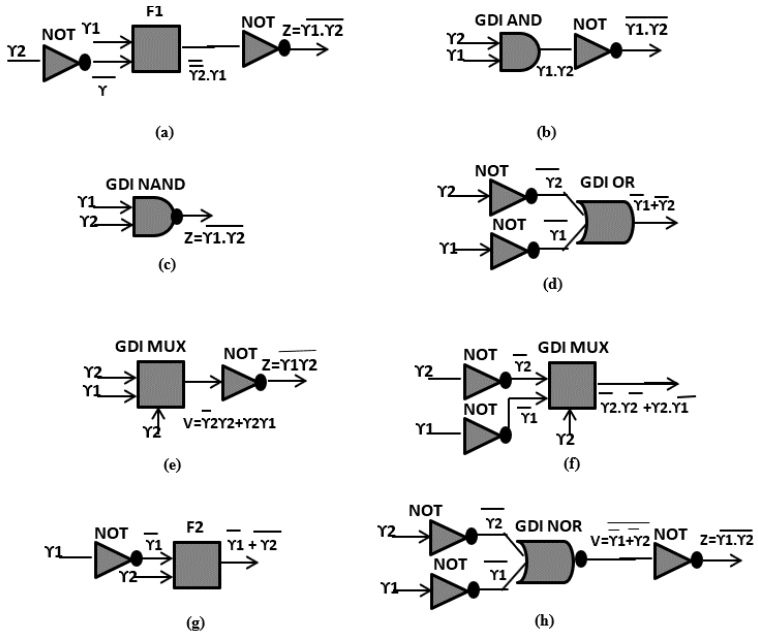


Figure 8. **Various NAND patterns**

4-4- GDI NOR Gate

The GDI NOR cell 8-various patterns have been implemented using inverter-F1, GDI NOR, inverter-GDI AND, GDI OR-inverter, GDI-MUX-inverter, Complemented GDI MUX, Inverter-F1-inverter and inverter-GDI NAND-inverter with Boolean expression as $\mathbb{Z} = \overline{Y1}.\overline{Y2}$ (using F1), $\mathbb{Z} = \overline{Y1 + Y2}$ (using GDI NOR), $\mathbb{Z} = \overline{Y1}.\overline{Y2}$ (GDI AND), $\mathbb{Z} = \overline{Y1 + Y2}$ (GDI OR), $\mathbb{Z} = \overline{\overline{Y2}.Y1 + Y2.Y2}$ (GDI MUX), $\mathbb{Z} = \overline{Y1}.\overline{Y2} + Y2.\overline{Y2}$ (complemented GDI MUX), $\mathbb{Z} = \overline{\overline{\overline{Y2} + Y1}}$ (F2) and $\mathbb{Z} = \overline{\overline{\overline{Y1}.\overline{Y2}}}$ (complemented GDI NAND). The NOR implementation is presented in Figure 9.

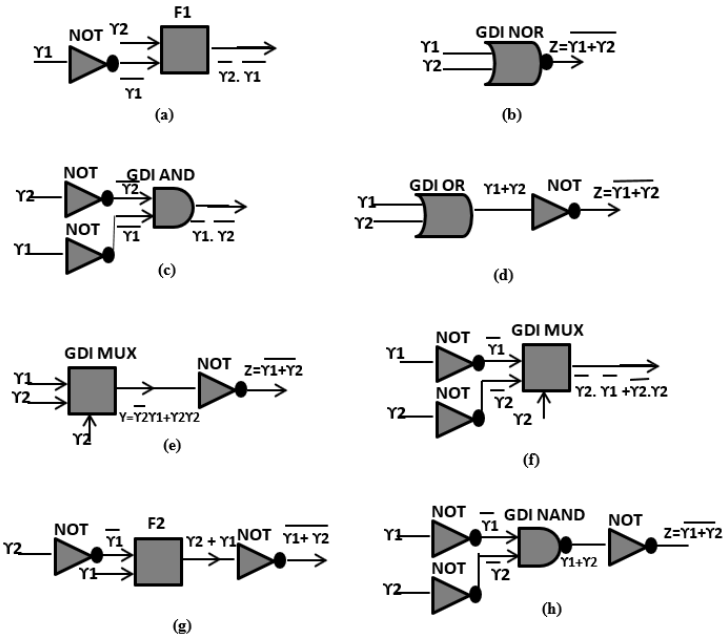


Figure 9. **Various NOR patterns**

4-5- GDI XOR Gate

XOR gate which is one of the fundamental components deployed in adder, subtractor, multiplier and other logic functions. In this research work, six different XOR gate is implemented and shown in Figure 10. The first pattern is implemented with the Boolean function $Y = Y1\overline{Y2} + \overline{Y1}.Y2$ the circuit is realized using F1 and GDI MUX gates. The

second XOR gate is constructed using inverter-AND-MUX, third circuit is built with F1- GDI AND gates. The fourth XOR gate is implemented with complemented F1-GDI MUX-inverter gate, the fifth structure is comprehended using F2-GDI AND-inverter and last XOR gate is implemented with GDI XOR itself. Extensive analysis for XOR gate with existing counterpart is done and its characteristics and complete findings is presented in Ponnian et al. [24].

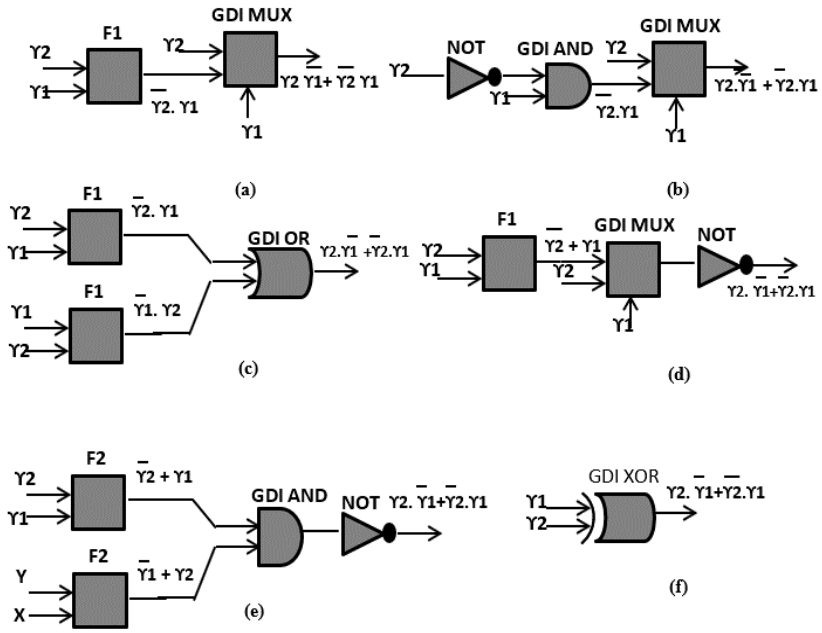


Figure 10. Various XOR patterns

4-6- GDI MUX Gate

Six different MUX patterns have been generated and its structural realization is depicted in Figure 11. The first and second MUX is implemented using inverter-F1-GDI OR and F1-GDI AND-GDI OR gate. The third MUX pattern is realized completely using NAND with the Boolean expression $\mathbb{Z} = \overline{\overline{\Upsilon 2.\Upsilon 1} + \overline{\Upsilon 2.\Upsilon 1}}$. The fourth MUX topology is constructed using GDI AND-GDI OR providing the expression $\mathbb{Z} = \overline{\Upsilon 2}.\Upsilon 1 + \Upsilon 2W$. The fifth MUX is built using F2-GDI AND-GDI OR with the logical expression $\mathbb{Z} = \Upsilon 2(\overline{\Upsilon 2} + \Upsilon 1) + \overline{\Upsilon 2}.W$. The final MUX is GDI MUX structure itself. Exhaustive simulation is done to choose the optimized patterns to incorporate the primitive cell in the proposed GDI library. The GDI library is depicted in Figure 12.

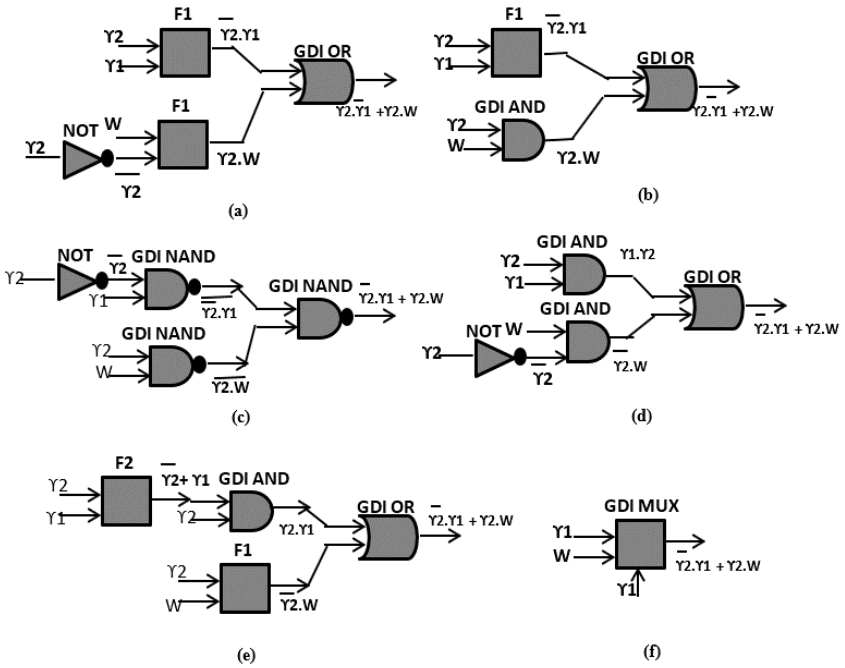
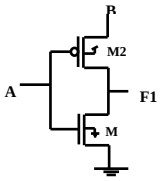
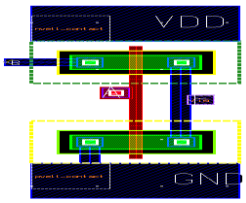
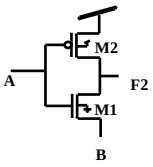
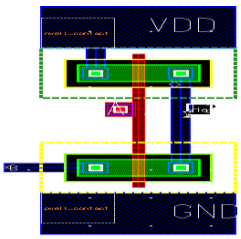
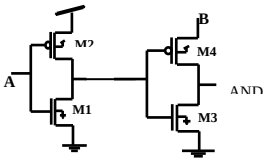
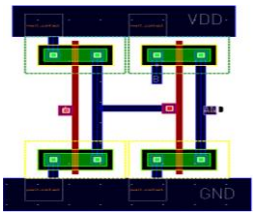
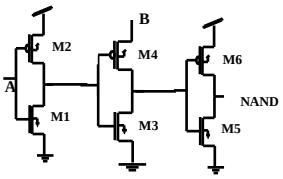
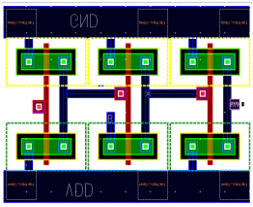
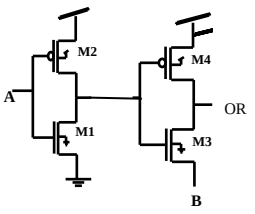
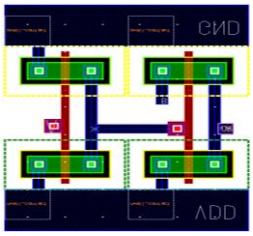
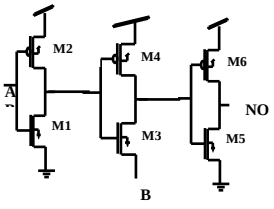
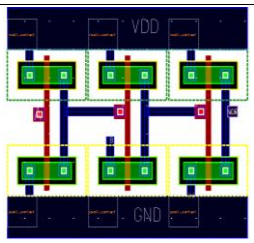
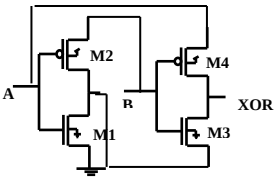
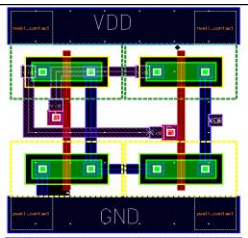
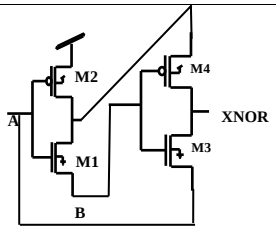
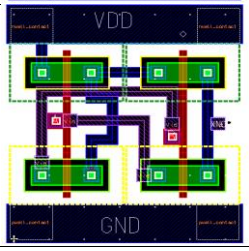


Figure 11. Various MUX patterns

GDI Primitive Cells			
	Symbol Name	**Transistor Presentation**	**Layout**
F!	F1GDI		
F2	F2GDI		
A N D	ANDGDI		
NAND	NANDGDI		
OR	ORGDI		
NOR	NORGDI		
XOR	XORGDI		
XNOR	XNORGDI		

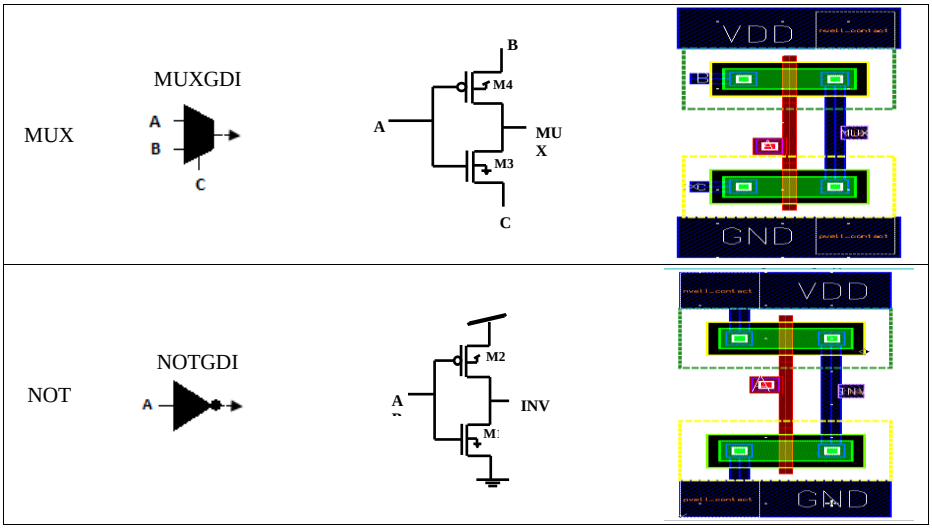


Figure 12. Complete GDI library

5- RC Delay Model of GDI Cell

The delay for the proposed GDI library is developed using Logical Effort [8] approach. This method is compact and easy to approximate the delay of a circuit. Logical effort approach is the fastest way to approximate the delay incurred in different logic structures irrespective of technology. This technique also stipulates the suitable numeral of phases (stages) on a given path and the superlative aspect ratio of a given gate. This model represents a MOSFET in terms of resistances and capacitances. The delay of the circuit is characterized mainly by two factors; the first which signifies the capacitive driving logic gate and the second which uses the network topology of the logic structure. The main characteristics of this delay model are as follows:

- It is independent of the technology and only depends on the transistor level design of the components.
- The path delay or path effort is obtained as the summation of the delays of the gate stages along a particular path.
- It does not consider gate sizing optimizations.
- It does not consider the delay of the wiring interconnects.

The Logical Effort method exemplifies the delay of the logic gate using three parameters: parasitic delay (p), logical effort (g) and electrical effort (h). These parameters can be obtained through modeling the logic gate in terms of capacitors and resistors. The RC model of the GDI basic cell is illustrated in Figure 13. The input to gate and diffusions are designated as V_G, V_P and V_N, assuming V_P = VDD and V_N = GND. The PMOS is modeled as switch and resistance Rpull forming the path between V_p and output Z.

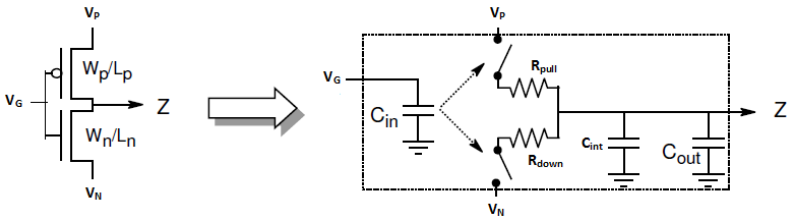


Figure 13. RC delay model of basic GDI cell

Similarly, the NMOS is modeled as switch and resistor R_{down} forming a path between V_N and output Z. The delay of this RC network is obtained via the output node capacitance during its charging and discharging time, which is stated as:

$$Z(t) = VDDe^{\frac{-t_p}{R_t(Cout_{int}())}} \quad (3)$$

where R_t represents the pull up (R_{pull}) and pull down (R_{down}) resistance, C_{out} represents the load output capacitance, C_{int} is the internal capacitances formed between source, drain, bulk and gate region. From equation 3 the delay of the circuit can be obtained as:

$$t_p = R_t(C_{out} + C_{out})\ ln\left(\frac{VDD}{Z(t)}\right) \quad (4)$$

where t_p represents rise or fall delay and assuming the high and low state to be 65% and 35 % of VDD the equation 4 is approximated as

$$t_p \Rightarrow \begin{cases} t_{pull} = R_{pull}(C_{int} + C_{out}) \\ t_{down} = R_{down}(C_{int} + C_{out}) \end{cases} \quad (5)$$

Assuming if fall and rise delay is equal then

$$t_p = t_{pull} = t_{down} = R_t(C_{int} + C_{out}) \quad (6)$$

The expression in equation 5.4 represents the delay of 1-x gate. For n-x gate the equation 5.4 have to be scaled by n and the delay is expressed as:

$$t_p = \frac{R_t}{n}(nC_{int} + C_{out}) \quad (7)$$

For 1-x gate it is not necessary to include the input capacitances since for equal rise and fall delay its value is equal to one. But for n-x gate the effect of input capacitance should be included in the delay expression, and it is represented as:

$$t_p = R_t C_{in} + R_t C_{in}\left(\frac{C_{out}}{nC_{in}}\right) \quad (8)$$

where nC_{in} defines input capacitance of n-x logic gate, the above delay equation is expressed in the form of three parameters viz., the parasitic delay p_{GDI}, logical effort g_{GD_I} and the electrical effort delay h_{GDI}, i.e., $p_{GDI} = \frac{R_t C_{int}}{\tau}$, $h_{GDI} = \frac{C_{out}}{nC_{in}}$, $g_{GDI} = \frac{R_t C_{in}}{\tau}$ where g_{GDI} is Logical effort of GDI gate, h_{GDI} is Electrical effort of GDI gate, p_{GDI} is Intrinsic (parasitic) delay of GDI gate, and τ is represents the characteristics delay for a technology ($\tau = R_{inv}.C_{inv}$) for inverter as the reference circuit.

The absolute delay for single GDI network is given as:

$$t_{pGDI} \Rightarrow d_{GDI} = \tau(g_{GDI}h_{GDI} + p_{GDI}) \quad (9)$$

The proposed delay equation for the GDI cell depends on the input and output resistances and capacitances. For the chain of GDI cells, the delay can be determined through a RC network or a multistage logical effort approach. Any Boolean function in the GDI technique is implemented using the series-connected basic GDI cells as a single path in cascade. A circuit realization with its single path RC network is illustrated in Figure 14, where R_1, R_2, ..., R_n represents the resistances of conducting transistors of GDI cells; R_{b1}, R_{b2}, ..., R_{bn} represents the resistances of conducting transistors of buffer cells; C_1, C_2, ..., C_n and C_{b1}, C_{b2}, ..., C_{bn} are the capacitive loads caused by GDI cells and buffer cells. For full swing output the buffer is mandatory, represented as R_{bout} and C_{bout}.

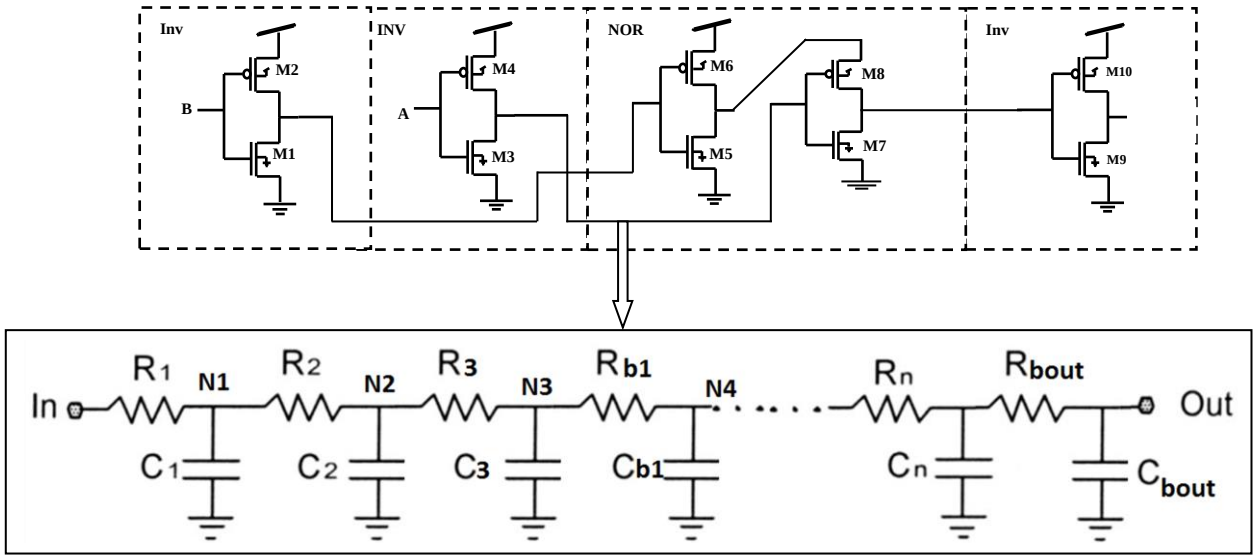


Figure 14. RC Network of GDI logic connected in series

The delay of the GDI RC network can be computed as the sum of resistances along the input and output node, which includes GDI cell resistances and buffer cell resistances along the path. In a cascaded network, the buffer is inserted

between node N3 and N4 since the allowable voltage drop is limited for three consecutive GDI cells in a link. Therefore, for every chain of GDI links (three consecutive basic cells) a mandatory buffer should be included to restore the threshold drop and a level restoring circuit (or buffer) is connected at the final output of the chain for full swing voltage. R_{tt} represents the sum of resistances of R_n, R_{bn} and R_{bout} where R_n is the resistances of GDI cells, R_{bn} resistances of buffers and R_{bout} resistance of output buffer. Therefore, the delay of the RC network is

$$t_p = \left(\sum_{i=1}^{N}(C_i + C_{bn} + C_{bout}) \bullet \sum_{tt=1}^{i} R_{tt}\right) ln\left(\frac{VDD}{Z(t)}\right) \quad (10)$$

where $R_{tt} = \sum(R_n, R_{bn}, R_{bout})$.

The maximum buffer inclusion in the stages depends on the number of GDI cells linked in the entire RC network.

6- Logical Effort Delay Model for Un-Skewed GDI Gates

This section explains the delayed calculation of GDI cells reported by [28, 29] and the proposed GDI and EGDI library for un-skewed gate, offering equal falling and rising time.

6-1- Delay Calculation for Single 2-Input GDI Cell

The calculation of delay for un-skewed gates (MOSFETs) will have an aspect ratio of P- MOSFET while N-MOSFET will be a 2:1, resulting in the circuit having an equal rise and fall delay. For this un-skewed gate βn =βp (where β is the trans-conductance), the nominal threshold voltage V_{inv} is VDD/2. This might be necessary since it exploits the noise margins permitting load capacitance to discharge and charge to provide sourcing and sinking capabilities at an equal time.

Computation of Logical Effort (g_{GDI})

The logical effort g_{GDI} signifies the competence of GDI gate network organization to yield maximum output current which depends upon the width of pMOS and nMOS transistor concerning the width of reference inverter circuit.

$$g_{GD_I} = \frac{R_t C_{in}}{\tau} = \frac{R_t C_{in}}{R_{inv} \cdot C_{inv}} \quad (11)$$

The input capacitance is comparable to the width of the gate capacitance of PMOS and NMOS concerning the width of the gate capacitance of the reference inverter circuit.

$$C_{in} = \left(\frac{W_{GDIgate\_P} + W_{GDIgate\_N}}{W_{inv\_P} + W_{inv\_N}}\right) C_{inv} \quad (12)$$

On substituting C_{in} in (5.9), and if the driving capability of 1-x gate is equal to the reference inverter gate then R_p=R_{inv} therefore logical effort g_{GDI} becomes

$$g_{GD_I} = \left(\frac{W_{GDIgate\_P} + W_{GDIgate\_N}}{W_{inv\_P} + W_{inv\_N}}\right) \quad (13)$$

Computation of Parasitic Delay (p_{GDI})

The parasitic delay of primary influence is the diffusion capacitance connected at the output node (signal). Overall, the parasitic delay is the proportion of the width output GDI gate to the width of the output inverter circuit.

$$p_{GDI} = \frac{R_t C_{int}}{\tau} = \frac{W_{output\_GDIgate}}{W_{output\_inv}} \quad (14)$$

Computation of Electrical Effort (h_{GDI})

The primary contribution to the electrical effort is due to the GDI gate's capacitances of input and load capacitance. It can be demarcated as the fraction of capacitance connected at the output side of the GDI gate to the capacitance connected to the input side of the GDI gate.

$$h_{GDI} = \frac{C_{out}}{nC_{in}} = \frac{C_{output\_GDI}}{C_{in\_GDI}} \quad (15)$$

The delayed calculation of GDI cells [28, 29] is shown in Figure 15. To obtain an equal rising and falling delay, the width of the PMOS transistor is scaled twice of the NMOS. The aspect ratio is chosen as 2:1 while the input and output capacitances are assumed to be equal so that the electrical effort is h_{GDI}=1. For the reference inverter circuit, the logical effort g_{inv}= (2+1)/3=1, the parasitic delay p_{inv}=1 and the electrical effort h_{inv} = 1. For illustration, ruminate that the GDI NAND gate which requires 4 transistors to have true and complementary input B signal. Therefore the logical effort g_A for input A is calculated as the sum of the width of transistors connected by the input A to the width of the reference

inverter circuit, i.e., $g_A=(2+1)/3=1$. Similarly, for input B which has a true and complementary signal, designated as g_B^* is calculated as the sum of the width of transistors that is connected to input B to the width of the reference inverter, i.e., $g_B^*= (2+1+1)/3$. The parasitic delay is computed concerning the transistor capacitances connected in the output side which is equal to 2. Assuming h=1, the delay of NAND gate is $D_{GDI}=\tau(g_{avg\_GDI}h_{GDI}+p_{GDI}) = \tau(7/6+2) = \tau(3.266)$ where τ represents the process parameter for particular technology.

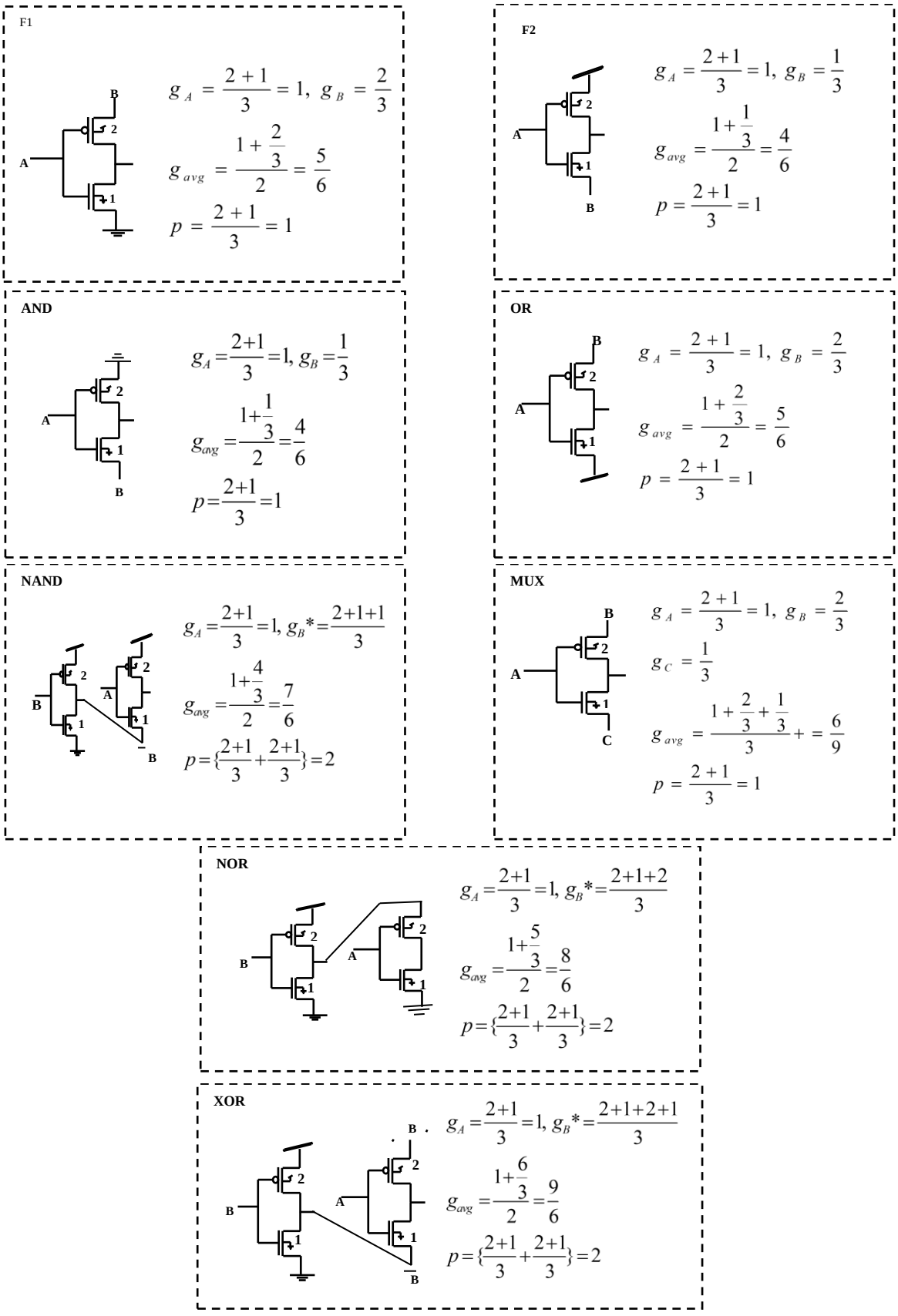


Figure 15. Logical effort and parasitic calculation for basic GDI cells

The computation of logical effort and parasitic delay for the proposed GDI cells are shown in Figure 16. The absolute delay is calculated in a cascaded form for the proposed GDI cells. For example, for the NAND structure, three stages of basic cells are cascaded in a chain i.e, INV-F1-INV, with the logical effort of an inverter calculated as g_A=2+1/3=1. For the next stage the F1 contains input B and complementary input A, with the equivalent logical effort for B, $\overline{A}$ to be g_B=2/3 and $g_{\overline{A}}$=1. For the last inverter cell, when the output of F1 is cascaded, the logical effort of g_{F1}=2+1/3=1.When the output capacitances are contributed through the function F1 and inverter, the parasitic delay will be equal to sum of F1 capacitance and output inverter capacitance. So, in the delay calculation the logical effort has to be calculated for g_A, g_B andg_{F1}. Assuming h=1, the delay of NAND gate is $D_{GDI}=\tau(g_{avg\_GDI}h_{GDI}+p_{GDI}) = \tau(11/12+2) = \tau(2.91)$.

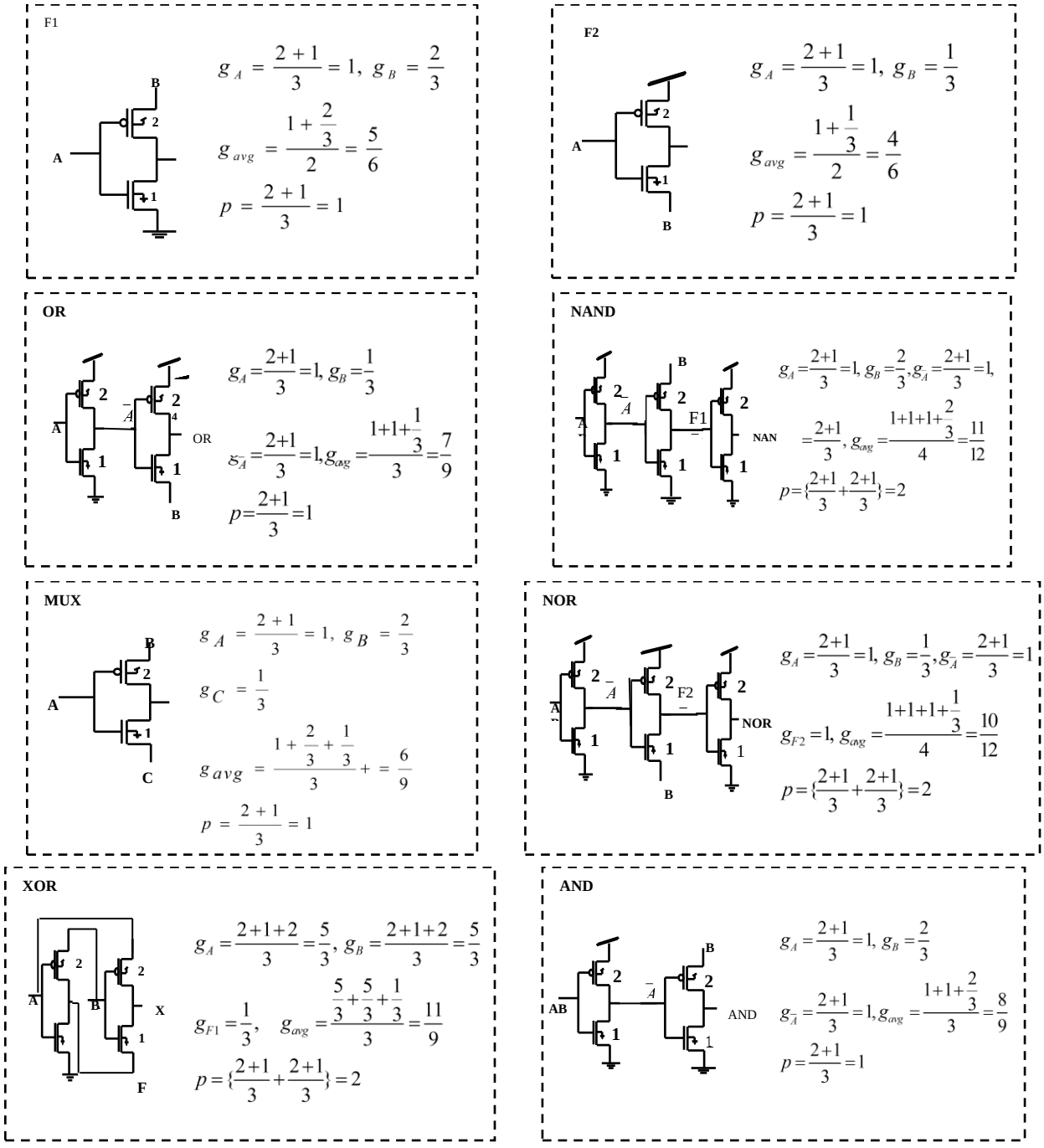


Figure 16. Logical effort and parasitic calculation for proposed GDI cells

The logical effort, parasitic delay and absolute delay of 2-input primitive cells for GDI, proposed GDI and CMOS is presented in Table 2. The graph in Figure 17 illustrates the absolute delay for the existing CMOS logic, GDI and proposed GDI and EGDI cells.

Table 2. Logical delay of un-skewed gates for Effort and Parasitic GDI, and CMOS logic

	$\overline{Y2}$	$\overline{Y2}Y1$	$\overline{Y2}+Y1$	$Y2Y1$	$Y2+Y1$	$\overline{Y2Y1}$	$\overline{Y2+Y1}$	$Y2 \oplus Y1$ /$Y2 \odot Y1$	$\overline{Y2}.Y1+Y2W$
Logical effort (2-input)									
Basic GDI cell]20,21]	1	5/6	4/6	4/6	5/6	7/6	8/6	9/6	6/3
Proposed GDI	1	5/6	4/6	8/9	7/9	11/12	10/12	11/9	6/3
CMOS	1	-	-	-	-	4/3	5/3	4	4
Parasitic delay (2-input)									
Basic GDI cell]20,21]	1	1	1	1	1	2	2	2	1
Proposed GDI	1	1	1	1	1	2	2	2	1
CMOS	1	-	-	-	-	2	2	4	4
Absolute delay (h=1)									
Basic GDI cell]20,21]	2	1.83	1.66	1.66	1.8	3.16	3.33	3.5	3
Proposed GDI	2	1.83	1.66	1.88	1.76	2.91	2.83	3.2	3
CMOS	2	-	-	-	-	3.33	3.67	8	8

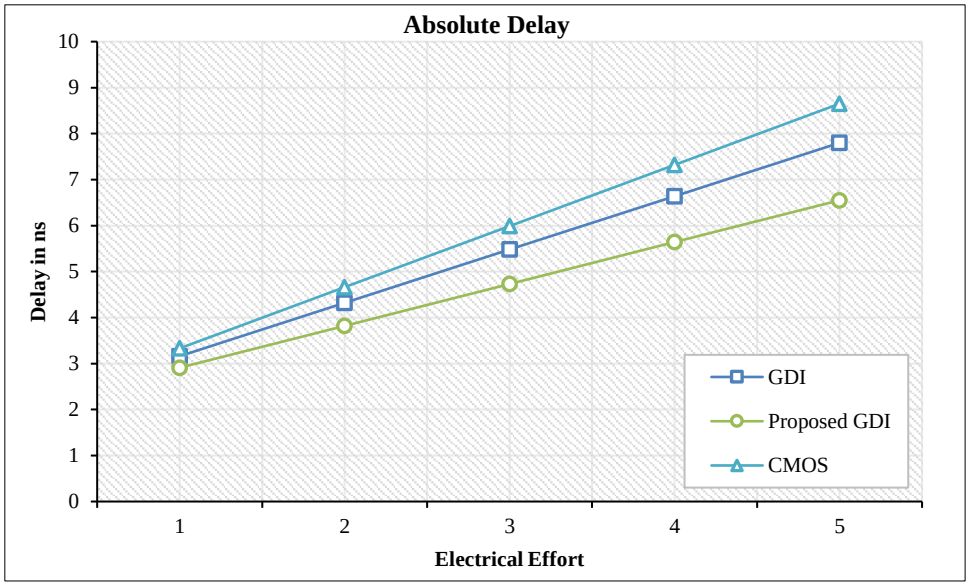


Figure 17. Absolute Delay for 2-NAND gate for varying Electrical Effort

6-2- Delay in Multistage Logic Network for Un-Skewed GDI Gates

The delay incurred in the multistage logic network entails the pathway of parasitic delay and trail of effort delay and the which is stated as:

$$D_{GDI} = N\left(\prod\left(g_{i\_GDI\_cell}\right)\left(g_{i\_buffer}\right) \bullet \prod b_i \bullet H\right)^{1/N} + \sum\left(P_{i\_GDI\_cell}\right)\left(P_{i\_buffer}\right) \quad (16)$$

where N is the no of gates associated in the path, $g_{i\_GDI\_cell}$ is the logical effort for single gate in the path, g_{i_buffer} is the logical effort of the buffer inserted at the output node of each gate, b_i is the branching effort (fan-out) in the path. It calculates the load capacitances along the path and the capacitances that lead off the path whenever fanout occurs along the trail, and H is electrical effort.

This logical effort model estimates the delay of the path in terms of effort and parasitic contribution.

7- Logical Effort Delay Model for Skewed GDI Gates

In combinational circuit design, skewed gates offer enhanced delay and leakage current for particular designs. A traditional static un-skewed circuit does not allow the outputs to change in a particular mode like rising or falling. Nevertheless, skewed network design in static implementation permits the output to change in a particular direction, since the individual logic gate is certain to exclusively toggle either rising (pull-up) or falling (pull-down). This type of changeover improves the enactments and driving competences of the transistor. This can be achieved by varying the

aspect ratio of the pMOS and nMOS transistor. HI-skewed gates have a higher aspect ratio for pMOS, uncertainty the input is VDD/2, then it is predictable that the output must be larger than VDD/2. Consequently, the input threshold will be slightly higher for a skewed gate. Correspondingly, LO-skewed gates have a low aspect pMOSFET transistor and reducing the switching threshold. Skewed logic design permits a compromise between the noise margin and the delay of the gate. Because of the higher noise margin tolerance, skewed gates are preferable for low voltage/low power high performance applications.

The parasitic capacitances in the skewed gates play a significant role in improving noise margin and current driving capabilities. The gate delay versus energy consumption relies on the capacitive effect of the transistor. For optimizing the transistor design, the driving current of a circuit must be increased and the circuit delay is decreased, while considering the parasitic resistance and the capacitance effects. For a MOSFET model the total capacitance will be the summation of gate oxide capacitance C_{ox}, the gate to Source/Drain overlap capacitance $C_{g\text{-}s/d}$ and the sidewall fringing capacitance. As the aspect ratios vary the capacitive effect will be more enunciated. Change in the W/L will show adverse short channel effects. A large aspect ratio estimates small resistance that allows for larger current flows. Since the parasitic resistance is inversely proportional to devise geometry. For HI-skewed inverter the parasitic capacitance is high, allowing larger current flow and smaller resistance.

The delay experienced in skewed logic for the single gate is stated as:

$$t_{puGDI} \Rightarrow d_{uGDI} = \tau(g_{uGDI}h_{GDI} + p_{uGDI}) \quad (17)$$

$$t_{pdGDI} \Rightarrow d_{dGDI} = \tau(g_{dGDI}h_{GDI} + p_{dGDI}) \quad (18)$$

$$d_{avg} = \frac{d_{uGDI}+d_{dGDI}}{2} \quad (19)$$

where g_{uGDI} is Logical effort for rising transition, g_{dGDI} is Logical effort for falling transition, p_{uGDI} is Parasitic delay for rising transition, and p_{dGDI} is Parasitic delay for falling transition.

The rising output transition is high for Hi-Skewed gates and for LO-skewed gates, the falling transition is high. This type of skew will be achieved by reducing the aspect ratio of the non-critical transistor.

For HI-skewed gates:

$$\text{Logical effort } g_{uGDI} = \frac{Input\ capacitance\ of\ HI{-}skew\ gate}{Input\ capacitance\ of\ unskewed\ gate\ (equal\ rise\ resistance)} \quad (20)$$

$$\text{Logical effort } g_{dGDI} = \frac{Input\ capacitance\ of\ HI{-}skew\ gate}{Input\ capacitance\ of\ unskewed\ gate\ (equal\ fall\ resistance)} \quad (21)$$

For LO-skewed gates:

$$\text{Logical effort } g_{uGDI} = \frac{Input\ capacitance\ of\ LO{-}skew\ gate}{Input\ capacitance\ of\ unskewed\ gate\ (equal\ fall\ resistance)} \quad (22)$$

$$\text{Logical effort } g_{dGDI} = \frac{Input\ capacitance\ \ of\ LO{-}skew\ gate}{Input\ capacitance\ of\ unskewed\ gate\ (equal\ rise\ resistance)} \quad (23)$$

The logical g_{uGDI}, g_{dGDI}, p_{uGDI} and p_{dGDI} calculation for the proposed NAND_GDI for high and low skewed logic is illustrated in Figure 18. Tables 3 and 4 presents the logical and parasitic values of proposed GDI skewed gates. A manifestation of the significance of skewed gates for full adder circuits constructed using several topologies is shown in Figure 19. Consider the full adder circuit in Figure 19a where the sum logic is implemented with the GDI_XOR gate and carry logic is implemented using 2-input AND_OR gates. The logical effort of the summing stage can be estimated as $g_{sum} = \prod(g_{XOR}, g_{Buff}) = 11/9 * 11/9 * 1 = 1.49$ and corresponding parasitic delay is $p_{sum} = \sum(AND, OR, Buff) = 2 + 2 + 2 = 6$. The total of GDI cells along the sum path is three while assuming the electrical efforts of H=5. If there is also no branching along with the summing network, then the branching effort will be b=1. The absolute delay along sum network is $d_{sum} = N(g_{sum} \bullet H \bullet b)^{1/N} + p_{sum} = 3(1.49 * 5 * 1)^{1/3} = \tau(11.82)$.Similarly, for carry stage g_{GDI} is $g_{carry} = \prod(g_{AND}, g_{OR}, g_{buff}) = 8/9 * 7/9 * 7/9 * 1 = 0.53$, $p_{carry} = \sum(AND, OR, Buff) = 1 + 1 + 1 + 2 = 5$ and finally the absolute delay is $d_{sum} = N(g_{sum} \bullet H \bullet b)^{1/N} + p_{sum} = 4(0.53 * 5 * 1)^{1/3} = \tau(10.51)$. The total delay of this full adder is 22.3τ. When the circuit delay estimated for HI-gates produce a higher delay, then the optimum result for mixed circuits are produced.

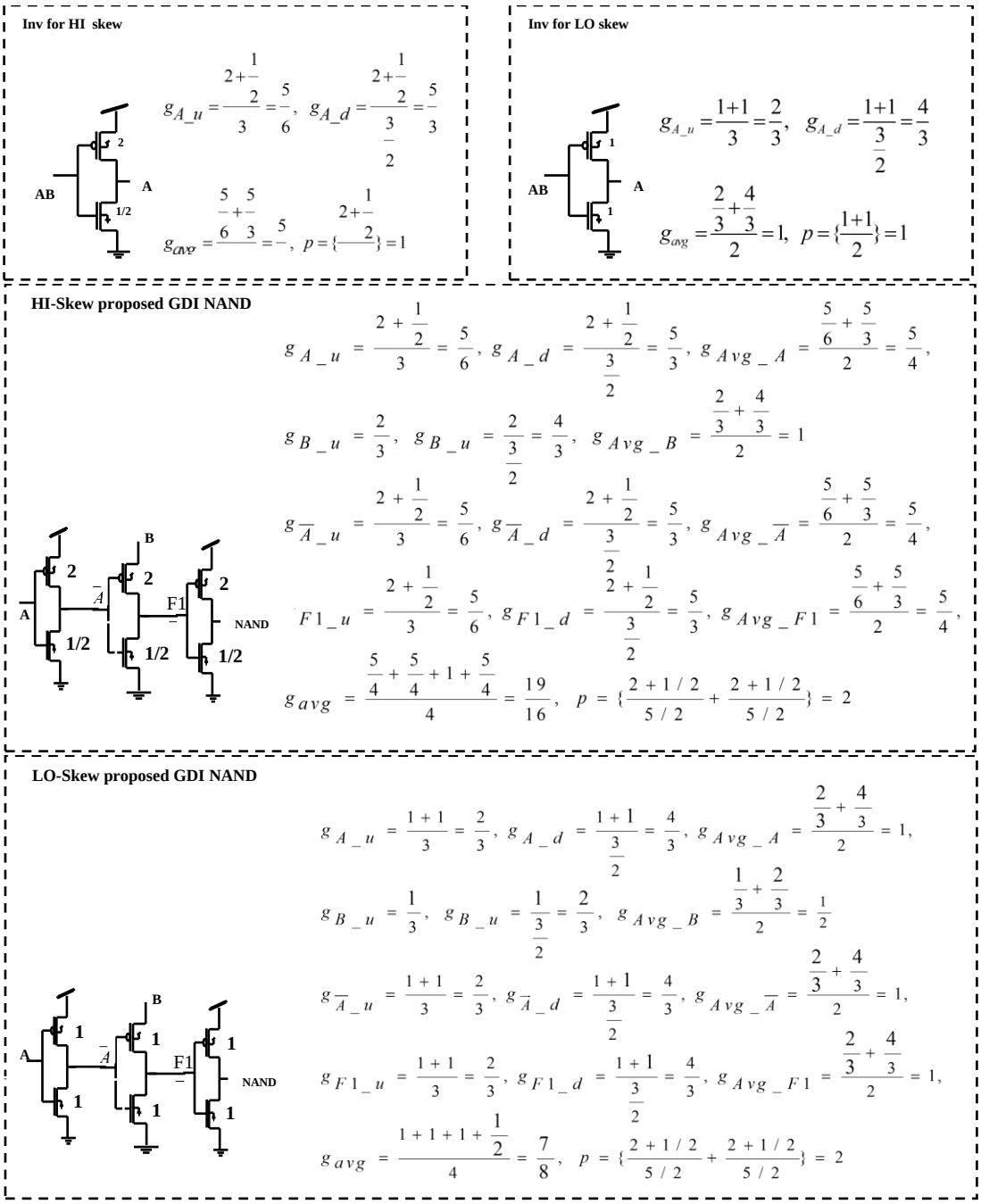


Figure 18. Calculation of Logical effort and parasitic delay for proposed GDI NAND cell in Hi-skew and Lo-skew

Table 3. Tabulation of gGDI/PGDI for HI-skew gates in GDI and CMOS logic

	$\overline{Y2}$	$\overline{Y2}Y1$	$\overline{Y2}+Y1$	$Y2Y1$	$Y2+Y1$	$\overline{Y2Y1}$	$\overline{Y2+Y1}$	$Y2 \oplus Y1$ /$Y2 \odot Y1$	$\overline{Y2}.Y1+Y2W$
Logical effort (2-input)									
Basic GDI cell	5/4	9/8	3/4	3/4	9/8	11/8	14/8	5/4	11/12
Proposed GDI	5/4	9/8	3/4	7/6	11/12	19/16	16/16	17/16	11/12
CMOS	5/4	-	-	-	-	3/2	9/4	3	7/3
Parasitic delay (2-input)									
Basic GDI cell	1	1	1	1	1	2	2	2	1
Proposed GDI	1	1	1	1	1	2	2	2	1
CMOS	1	-	-	-	-	2	2	4	4
Absolute delay (h=1)									
Basic GDI cell	2.25	2.12	1.8	1.8	2.12	3.37	3.75	3.25	1.91
Proposed GDI	2.25	2.1	1.8	2.1	1.91	3.25	3	3.01	1.91
CMOS	2.25	-	-	-	-	3.5	4.25	7	6.33

Table 4. Tabulation of gGDI/PGDI for LO-skew gates in GDI and CMOS logic

	$\overline{Y2}$	$\overline{Y2}Y1$	$\overline{Y2}+Y1$	$Y2Y1$	$Y2+Y1$	$\overline{Y2Y1}$	$\overline{Y2+Y1}$	$Y2 \oplus Y1$ / $Y2 \odot Y1$	$\overline{Y2}.Y1+Y2W$
Logical effort (2-input)									
Basic GDI cell	1	3/4	3/4	3/4	3/4	5/4	5/4	5/2	2/3
Proposed GDI	1	3/4	3/4	5/6	5/6	7/8	7/8	7/6	2/3
CMOS	1	-	-	-	-	3/2	3/2	4	2
Parasitic delay (2-input)									
Basic GDI cell	1	1	1	1	1	2	2	2	1
Proposed GDI	1	1	1	1	1	2	2	2	1
CMOS	1	-	-	-	-	2	2	4	4
Absolute delay (h=1)									
Basic GDI cell	2	1.75	1.8	1.8	1.8	3.25	3.25	4.5	1.667
Proposed GDI	2	1.75	1.8	1.8	1.8	2.95	2.95	3.166	1.667
CMOS	2	-	-	-	-	3.5	3.5	6	6

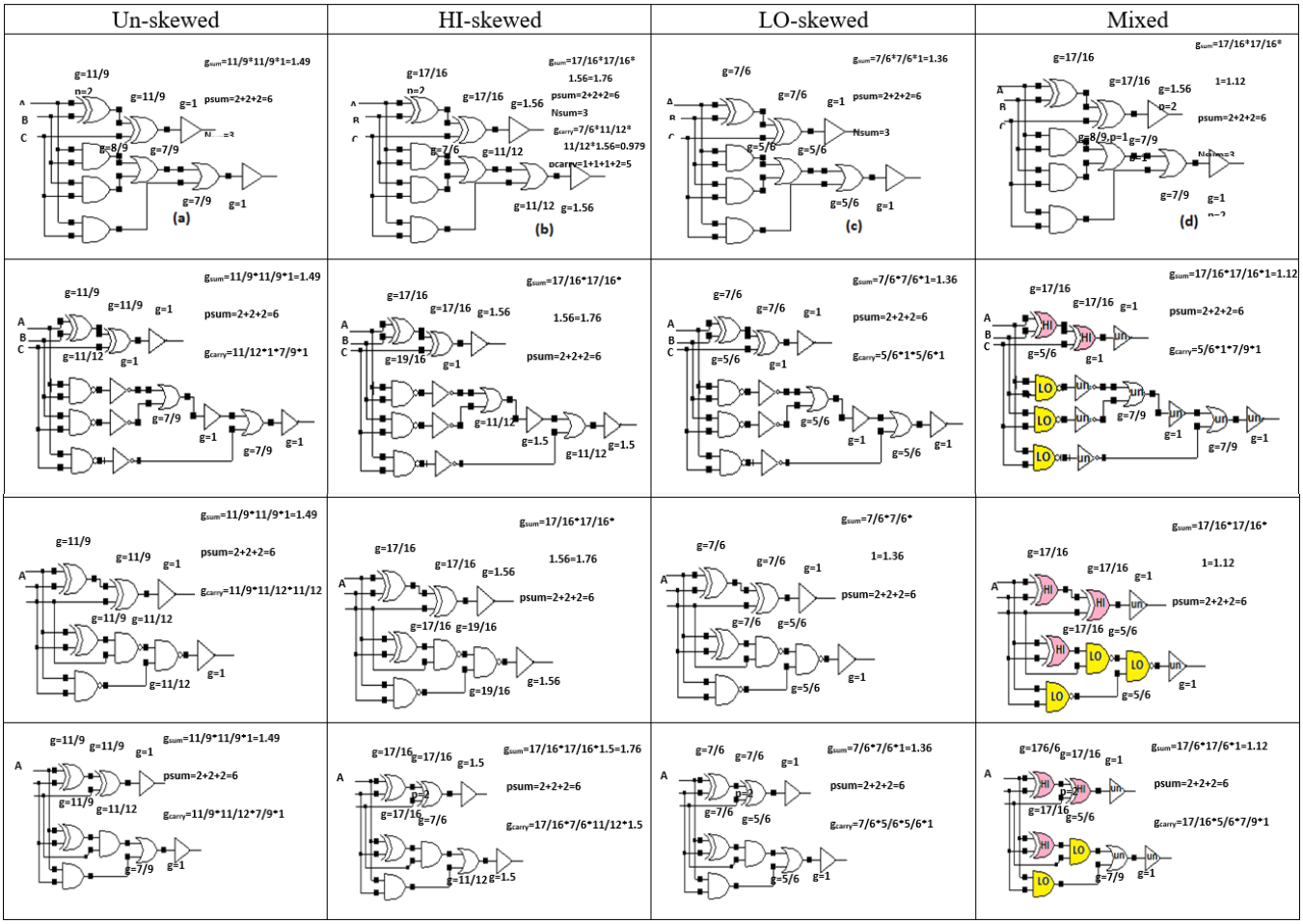


Figure 19. Several full adder circuit which is constructed using un skew, Hi-skew and LO-skew

8- Power Model for GDI Technique

The major component of power dissipation is expressed as

$$P_{avg} = P_{switching(dynamic\_GDI)} + P_{short-circuit} + P_{leakage} = \alpha_{0\to1\_GDI} C_{Load\_GDI} VDD^2 f_{clk} + I_{sc} VDD + I_{leakage} VDD \quad (24)$$

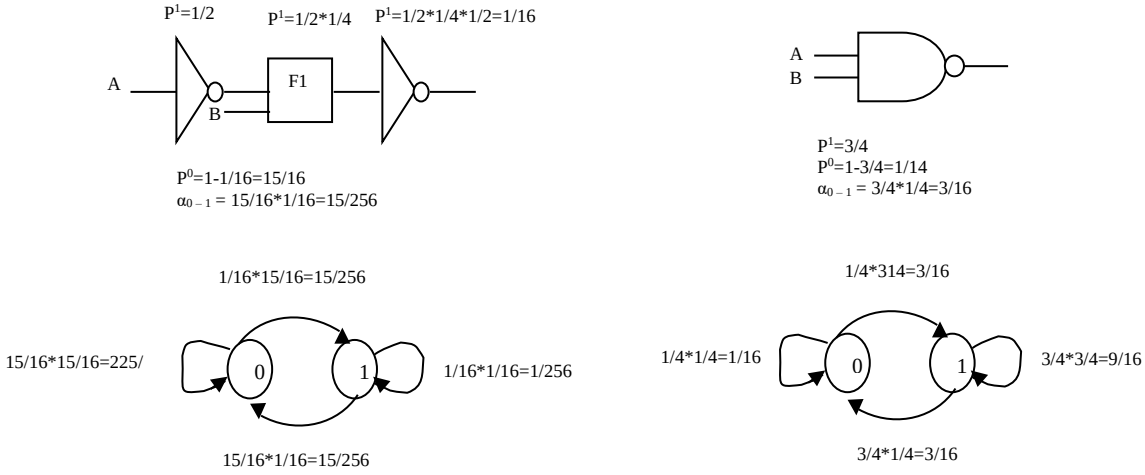
Here the first power dissipation is contributed by switching or dynamic constituent of PWR (power), where $\alpha 0_{\to 1}$ is the node activity aspect or node transition factor, C_{Load_GDI} is the load capacitance of the GDI network and f_{clk} is the clock

frequency of the GDI circuit. The parameter of C_{Load_GDI}, f_{clk} and VDD can be calculated using circuit layout information except the node activity factor α, which relies on the logic (Boolean) function or gate operation and the statistical parameter of the input signals applied to the GDI network.

Two components responsible for the dynamic power dissipation are the node activity factor and charging and discharging of load capacitance. The following assumptions are made to calculate the first component: a zero-delay gate model is considered where the gate delay and the glitches due to transitions are ignored and for single clock cycle, one input transition is allowed. The next assumption is that the inputs to the GDI network have an even supply of high-low and low-high levels. Transition probabilities for the output to be zero and one are denoted as P^0 and P^1. To calculate $\alpha_{0\rightarrow1}$ transition, the probability for GDI gate for the output to be zero is multiplied by the probability of the next state output, and that is expressed as:

$$\alpha_{0\rightarrow1\_GDI} = P^0P^1 = P^0(1-P^0) \quad (25)$$

Consider the calculation of the activity factor for the NAND gate of the proposed GDI library, which has a static 2-input with just one allowed transition. For the 2-input gate, there are then four possible state transitions for A and B as 0→0, 0→1, 1→0 and 1→1. Here, the NAND function is realized using inverter_1-F1-inverter_2, cascaded in the chain. The probability of one (P^1) will be 1/2 for the first inverter and for F1 it is 1/2*1/4=1/8 and for the last inverter the probability of one is 1/8*1/2=1/16. Therefore, the probability of zero (P^0) will be 1-1/16=15/16 and $\alpha_{0\rightarrow1}$ =15/16*1/16=0.058. To demonstrate the significance of the GDI technique, the node activity factor for NAND is calculated for CMOS logic and is illustrated in Figure 20 The probability of one (P^1) will be 3/4 and the probability of zero will be 1-3/4=1/4 and $\alpha_{0\rightarrow1}$ =3/4*1/4=0.1875 which is 69% higher compared to GDI logic. This depicts the dominance of the proposed GDI library cell. The state transition for the proposed GDI library cell and the node activity factor for GDI and CMOS is demonstrated in Figure 21 and Table 5.



a. Node activity factor and state transition for proposed GDI NAND gate

b. Node activity factor and state transition for CMOS NAND gate

Figure 20. Probability of $\alpha_{0\rightarrow1}$ for NAND gate in proposed GDI and CMOS logic

Table 5. Node activity factor for proposed GDI and CMOS

	Proposed GDI			CMOS		
Gate	**P^0**	**P^1**	**$\alpha_{0\rightarrow1}$**	**P^0**	**P^1**	**$\alpha_{0\rightarrow1}$**
F1	3/4	1/4	3/16=0.2	13/16	3/16	13/16*3/16=0.2
F2	1/4	3/4	3/16=0.11	15/16	1/16	15/16*1/16=0.1
AND	7/8	1/8	7/8*1/8=0.11	5/8	3/8	5/8*3/8=0.23
OR	5/8	3/8	5/8*3/8=0.2	7/8	1/8	7/8*1/8=0.2
NAND	15/16	1/16	15/16*1/16=0.1	1/4	3/4	3/4*1/4=0.2
NOR	13/16	3/16	13/16*3/16=0.1	3/4	1/4	3/4*1/4=0.2
MUX	4/8	4/8	4/8*4/8=0.3	4/8	4/8	4/8*4/8=0.3
XOR	28/32	4/32	28/32*4/32=0.11	2/4	2/4	2/4*2/4=0.3
XNOR	20/32	12/32	20/32*12/32=0.23	2/4	2/4	2/4*2/4=0.3
INV	1/2	1/2	1/2*1/2=0.25	1/2	1/2	1/2*1/2=.25
BUF	3/4	1/4	3/4*1/4=0.18	3/4	1/4	3/4*1/4=0.18

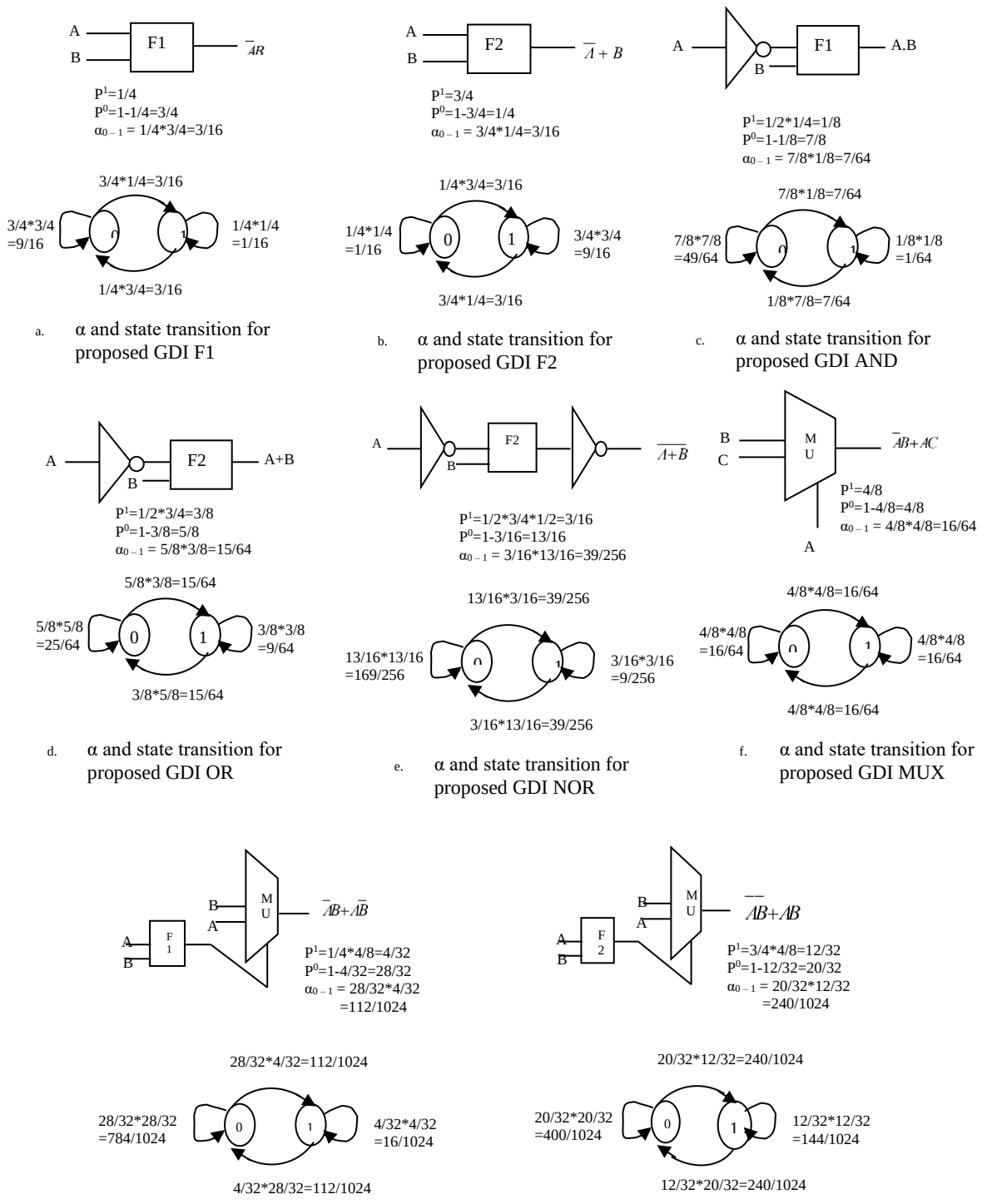


Figure 21. Probability of α 0 →1 for the proposed GDI library cells

From Table 5, it is well tacit that the node activity factors of the proposed GDI cells have better improvement than CMOS logic. The $\alpha_{0\to1}$ of the proposed GDI cells ranges from 0.1-0.2 whereas in CMOS it is 0.1-0.3.

The second component of dynamic power dissipation is contributed by the internal node capacitances and the charging and discharging of load capacitance connected at the output terminal of the GDI network. For a network of cascaded GDI gates operated for a frequency of f=1/t, the dynamic power dissipation is expressed as:

$$P_{charge\_discharge} = P_{charge\_buff} + P_{discharge\_buff} \quad (26)$$

where P_{charge_buff} is the power dissipated during charging the inverter (last buffer stage) of the GDI gate which includes the sum of internal node capacitances of driving the GDI node and the next receiving node m+1. The next capacitance contribution is the sum of the driving GDI node's wiring gate capacitances and the next receiving node. So, this charging power is the sum of power dissipation from VDD-V_{tn} to VDD of the driving GDI node, the power dissipated in driving the wiring and gate capacitances from 0 to VDD and finally the power dissipated by the driven GDI node from 0 to VDD- V_{tn}. The charging power can be expressed as:

$$P_{charge\_buff} = \frac{C_{int\_drive},m}{t}\int_{VDD-V_{tn}}^{VDD} V1dV1 + \frac{C_{wire+gate},m}{t}\int_{0}^{VDD} V1dV1 + \frac{C_{int\_drive\_next},m+1}{t}\int_{0}^{VDD-V_{tn}} V1dV1 \quad (27)$$

where C_{int_drive},m is Internal node capacitances of driving GDI node, $C_{wire+gate}$,m is Wiring and gate capacitances of driving GDI node, $C_{int\_drive\_next}$,m+1 is Internal node capacitances of next driven GDI node, and V1 is the output voltage during charging phase. In Equation 27, $\frac{C_{int\_drive},m}{t}\int_{VDD-V_{tn}}^{VDD} V1dV1$ is Power dissipation by the internal node capacitance of the driving GDI node (m), $\frac{C_{wire+gate},m}{t}\int_{0}^{VDD} V1dV1$ is Power dissipation by the wiring and gate capacitances of the driving GDI node (m), and $\frac{C_{int\_drive\_next},m+1}{t}\int_{0}^{VDD-V_{tn}} V1dV1$ is Power dissipation by the internal node capacitance of the driven next GDI node (m+1).

Similarly, the discharging power can be computed as the sum of power dissipation during the discharging of the internal node capacitance and wiring gate capacitances from VDD-0 can be expressed as

$$P_{discharge\_buff} = \frac{C_{int\_drive\_next},m+1}{t}\int_{VDD}^{0}(VDD-V2)\,d(VDD-V2) + \frac{C_{wire+gate},m+1}{t}\int_{VDD}^{0}(VDD-V2)d(VDD-V2) \quad (28)$$

In Equation 28, $\frac{C_{int\_drive\_next},m+1}{t}\int_{VDD}^{0}(VDD-V2)\,d(VDD-V2)$ is Power dissipation by the internal node capacitance of the last driven GDI node (m+1) and $\frac{C_{wire+gate},m+1}{t}\int_{VDD}^{0}(VDD-V2)d(VDD-V2)$ is Power dissipation by the wiring and gate capacitances of the last driven GDI node (m+1).

V2 defines the output voltage during the discharge phase. If the internal node capacitances of C_{int_drive},m and $C_{int\_drive\_next}$,m+1 are assumed approximately equal the Equations 26 and 28 can be reduced to

$$P_{charge\_discharge} = fclk\left(C_{int} + C_{wire+gate}\right)VDD^2 \quad (29)$$

The complete dynamic power dissipation can be stated as the product of node activity factor and power dissipated during charging-discharging of network, wire and gate capacitances of the GDI network which consisting of logic and buffer circuit is expressed as follows:

$$P_{switching(dynamic\_GDI)} = \alpha_{0-1} * P_{charge\_discharge} = \alpha_{0-1} f_{clk}\left(C_{int} + C_{wire+gate}\right)VVD^2 \quad (30)$$

The short-circuit power dissipation is mainly due to VDD and ground's direct path. Significant short-circuit current induces only for unequal rise-fall time at the input of the gate to that of output gate and if the supply voltage is reduced below the sum of threshold voltages of the PMOS and NMOS on the GDI network, VDD < V_{tn}+V_{tp}. The last power dissipation is the leakage currents due to reverse bias diode and subthreshold leakage of the nominally off transistor. The effect of leakage current is slightly predominant when the GDI cells are fabricated in CMOS n-well or p-well process and very minimum in Silicon-On-Insulator (SOI). The leakage current depends on the process technology and the second-order effects of the transistors. To estimate the performance of the proposed delay and power model of the ISCAS bench mark circuit of 74X series circuits is tested and its results are discussed in the succeeding session.

9- Experimental Results

This research defines two goals; the former goal depicts the proposal of GDI library cells with full swing using a MUX-based signal connectivity model and the later presents the mathematical delay-power model for the proposed GDI library cells. The research goals of this work and research methodology are shown in Figure 22.

The first part of the experimentation involves the simulation of proposed GDI primitive gates. The tool used for simulation is Mentor Graphic EDA. All cells are implemented with a 90 and 130nm process technology. For this simulation, the input supply is applied from 0V to 1.2V with a step size of 0.2V. This setup is maintained for the entire simulation. Exhaustive testing was done with varied design corners. The parameters observed during the simulation were delay (D), rise time, fall time, average power (Avg pwr), the total number of transistors (#Tr), PDP and product of delay and transistor count (#tr*delay). The optimized primitive gates are selected from various patterns based on these parameters. The performance of the proposed library is related to CMOS and PTL logic. The second part of this

experimentation involves the validation of the delay-power model. The delay model for un-skewed and skewed gates performances are measured in terms of simulation and estimation along with the percentage of deviation. The power model performance is observed using ISCAS 74-x combinational bench mark and its evaluation is also reported

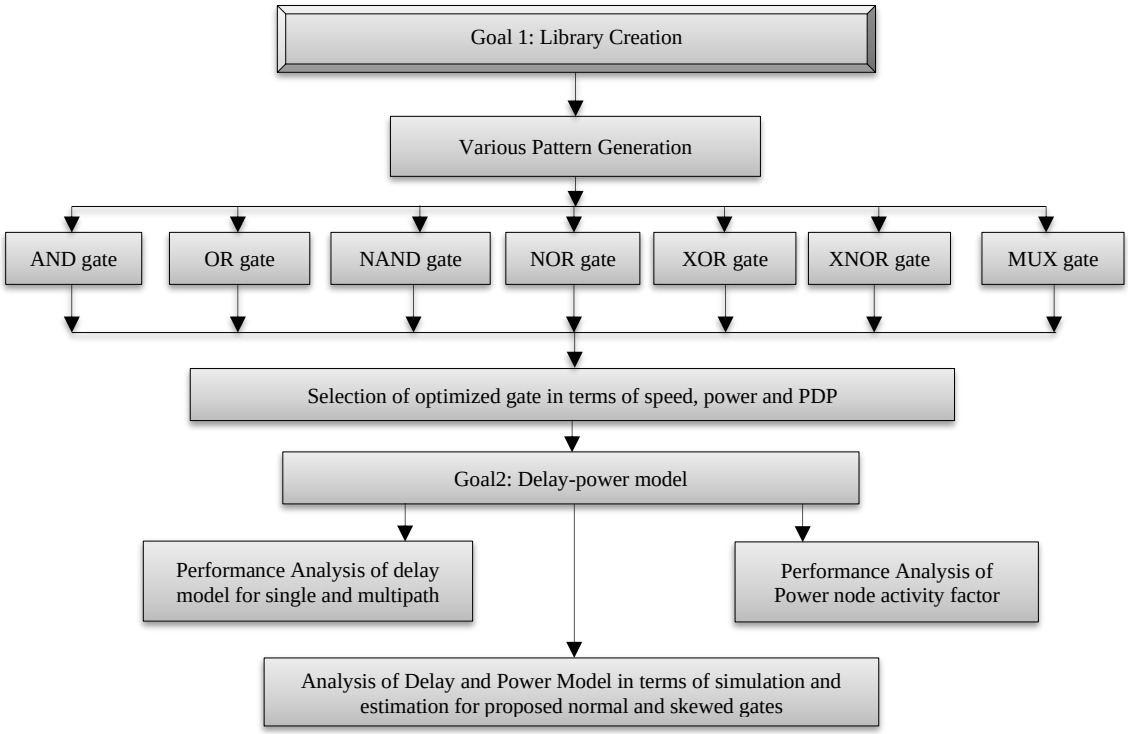


Figure 22. Research goals and methodology

The performances of un-skewed and skewed primitive cells are reported in Table 13. All the circuits are observed for the same experimental setup with a supply voltage of 1.2V. The aspect ratio of the un-skewed gate is taken as 2:1, whereas for Hi-skewed it is 4:1 and for Lo-skewed it is 1:1. The simulated values are compared with the proposed delay and power model. For this experimentation the load capacitance is 60pf and the input capacitance is 10pf. Therefore, the electrical effort H will be 60/10=6 and the characteristics delay τ for this technology is 100ps. The parameter observed for this simulation are rise time, fall time, the average delay in simulation, model delay from section 5 and 6, % deviation for un-skewed, Hi-skewed and Lo-skewed circuits. Similarly, the power is measured for un-skewed and skewed circuits and its performance is compared with the power model. The percentage deviation is calculated as

$$Deviation = \frac{(model - simulation)}{model} \times 100 \quad (30)$$

The simulated results of the primitive gates are presented in Tables 6 to 11. The delay values of logical effort and parasitic values are taken from the Tables 2, 3 and 4. Similarly for power calculation the node activity factor is taken from Table 5.

Table 6. Simulated results GDI AND gate

Pattern	Realization	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 6(a)	NOT+GDI F1	0.572	0.522	1.082	23.12	4	25.015	4.328	Optimal
Figure 6(b)	GDI NAND+NOT	0.422	0.432	1.772	26.04	6	46.142	10.632	
Figure 6(c)	GDI AND	0.213	0.249	2.29	20.20	2	46.258	4.58	
Figure 6(d)	NOT+GDI OR+NOT	0.414	0.466	1.239	25.74	8	29.891	9.912	Moderate
Figure 6(e)	GDI MUX	0.245	0.24	2.166	24.35	2	52.74	4.332	
Figure 6(f)	NOT+GDI MUX+NOT	0.454	0.472	1.253	20.98	8	26.287	10.024	
Figure 6(g)	NOT+GDI F2+NOT	0.499	0.501	2.34	24.51	6	57.353	14.04	High
Figure 6(h)	NOT+GDI NOR	0.432	0.423	1.124	30.44	8	34.214	8.992	

Table 7. Simulated results GDI OR gate

Pattern	Realization	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 7(a)	NOT+GDI F1+NOT	0.42	0.432	1.21	34.99	6	42.512	209.9	Moderate
Figure 7(b)	GDI NOR+NOT	0.57	0.545	1.53	30.78	6	47.093	184.6	
Figure 7(c)	GDI OR	0.22	0.372	1.41	31.76	2	44.813	163.5	
Figure 7(d)	NOT+ GDI AND+NOT	0.59	0.509	2.21	27.73	8	61.366	221.8	High
Figure 7(e)	GDI MUX	0.24	0.232	1.99	28.33	2	56.376	156.6	
Figure 7(f)	NOT+ GDI MUX+NOT	0.57	0.589	1.56	30.32	8	47.299	242.5	
Figure 7(g)	NOT+GDI F2	0.56	0.573	1.55	26.17	4	40.563	104.6	Optimal
Figure 7(h)	NOT+GDI NOR	0.55	0.672	1.33	35.44	8	47.135	283.5	

Table 8. Simulated results GDI NAND gate

Pattern	Combination	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 8(a)	NOT+GDI F1+NOT	0.543	0.671	1.015	22.32	6	22.654	6.09	Optimal
Figure 8(b)	GDI AND+NOT	0.532	0.531	2.15	39.89	4	85.763	8.6	High
Figure 8(c)	GDI NAND	0.521	0.625	2.62	25.2	4	66.024	10.48	
Figure 8(d)	NOT+GDI OR	0.456	0.462	1.53	28.21	6	43.161	9.18	
Figure 8(e)	GDI MUX+NOT	0.545	0.622	2.53	28.55	4	72.231	10.12	
Figure 8(f)	NOT+GDI MUX	0.523	0.521	2.73	20.6	6	56.238	16.38	
Figure 8(g)	NOT+ GDI F2	0.465	0.432	1.46	23.3	4	34.018	5.84	Moderate
Figure 8(h)	NOT+GDI NOR	0.427	0.438	1.87	20.9	10	39.083	18.7	

Table 9. Simulated results GDI NOR gate

Pattern	Combination	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 9(a)	NOT+ GDI F1	0.58	0.552	1.3	40.1	4	52.13	8.2	Moderate
Figure 9(b)	GDI NOR	0.52	0.512	1.51	40.9	4	61.75	6.04	
Figure 9(c)	NOT+GDI AND	0.47	0.421	2.410	33.88	4	81.65	9.64	
Figure 9(d)	GDI OR+NOT	0.55	0.511	2.22	38.8	6	86.13	13.32	High
Figure 9(e)	GDI MUX+NOT	0.43	0.513	1.9	28.32	4	53.80	7.6	
Figure 9(f)	NOT+GDI MUX	0.61	0.612	1.56	30.31	6	47.28	9.36	
Figure 9(g)	NOT+GDI F2+NOT	0.54	0.578	1.57	26.1	6	40.97	9.42	Optimal
Figure 9(h)	NOT+GDI NAND+NOT	0.44	0.433	1.41	45.32	10	63.90	14.1	

Table 10. Simulated results GDI XOR gate

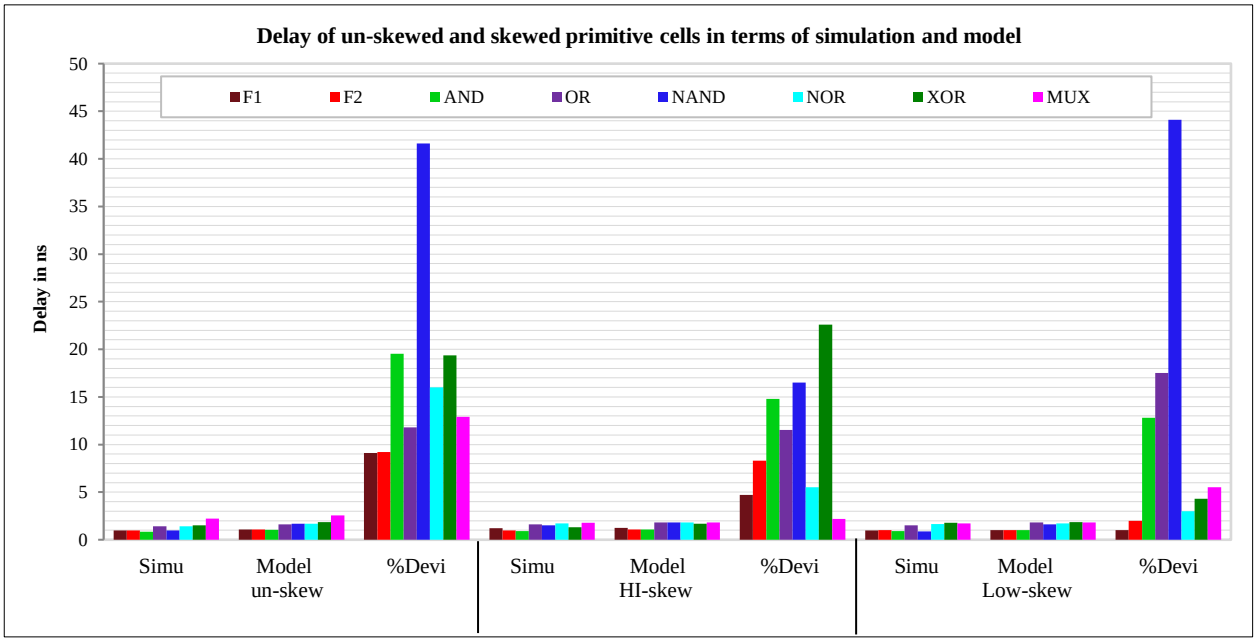
Pattern	Combination	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 10(a)	GDI F1+GDIMUX	0.481	0.472	1.91	29.1	4	55.581	7.64	Optimal
Figure 10(b)	NOT+GDI AND+GDI MUX	0.412	0.476	2.6	25.5	6	66.3	15.6	
Figure 10(c)	GDI F1+GDI OR	0.510	0.566	2.31	23.3	4	53.823	9.24	
Figure 10(d)	GDI F2+GDI MUX+NOT	0.413	0.487	2.51	25.5	6	64.005	15.06	
Figure 10(e)	GDI F2+GDI AND+NOT	0.414	0.479	2.22	22.1	6	49.062	13.32	Moderate
Figure 10(f)	GDI XOR	0.553	0.532	2.91	28.9	4	84.099	11.64	High

Table 11. Simulated results GDI MUX gate

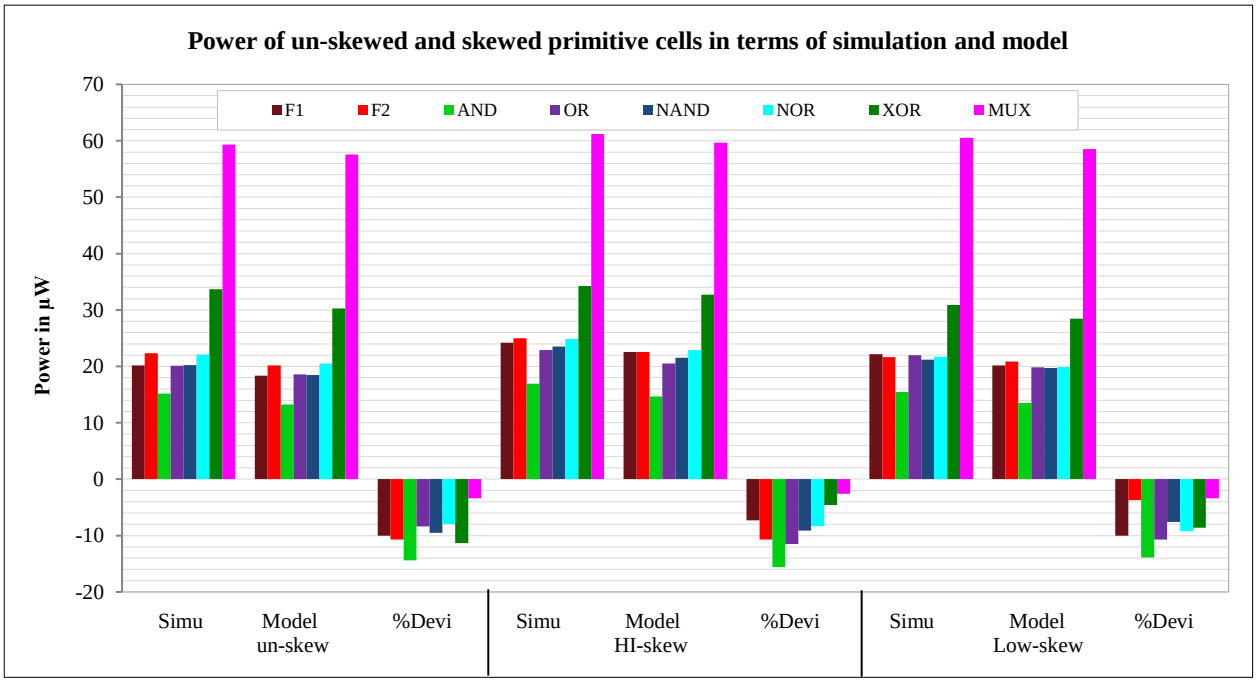
Pattern	Combination	RDT (ns)	FDT (ns)	D (ns)	A_PWR (uW)	TX	A_P*D (fW-S)	Tx*D (ns)	Observation
Figure 11(a)	NOT+GDI F1+GDI OR	0.465	0.498	2.12	63.76	8	135.1	16.9	Moderate
Figure 11(b)	GDI F1+GDI AND+GDI OR	0.532	0.621	2.32	59.32	2	137.6	4.64	Optimal
Figure 11(c)	NOT+GDI NAND	0.523	0.524	2.53	67.64	6	171.1	15.1	
Figure 11(d)	NOT+GDI AND+GDI OR	0.456	0.435	2.27	72.45	8	164.4	18.1	High
Figure 11(e)	GDI F1+GDI F2+GDI AND+ GDI OR	0.432	0.442	2.31	69.99	8	161.6	18.4	
Figure 11(f)	GDI MUX	0.582	0.576	2.61	61.32	14	160.0	36.5	

*RDT-rise delay time, *FDT-fall delay time, *d-delay, *A_PWR-average power, *TX-transistor count, *A_P*D-product of delay and power, *TX*D-product of transistor count and delay

The performance chart for delay and power in terms of simulation and estimation for the primitive cells is shown in Figure 23. Readings from the delay chart in Figure 23-a indicates that the minimum deviation is 2, which exist for F1 gate and a maximum of 44% exist for the NAND gate. The deviation is typically due to the non-inclusion of velocity saturation effects in the model. This deviation is not significant for simple gates. Still, considering the larger circuit, the delay model needs to be revised and the effects of variation in model and simulation are demonstrated in Table 12 for various full adder implementations. The power chart in Figure 23-b shows the deviation is negative since the power approximation model is derived with the assumption that gate delays are zero. So, in the power approximation model the power consumption owing to the glitch caused due to uneven path propagation through the circuit is ignored



(a) Delay in simulation and model



(b) Power in simulation and model

Figure 23. Performance of un-skewed and skewed primitive cells for the proposed GDI logic

Table 12. Performances of un-skewed and skewed primitive cells

	Un-skew					HI-skew					LO-skew				
Gate	**RDT (ns)**	**FDT (ns)**	**Simulated delay (ns)**	**Model delay**	**% Deviation**	**RDT (ns)**	**RDT (ns)**	**Simulated delay (ns)**	**Model delay**	**Deviation**	**RDT (ns)**	**RDT**	**Simulated delay (ns)**	**Model delay**	**Deviation**
F1	0.466	0.421	0.99	1.09	9.1	0.655	0.411	1.2	1.26	4.7	0.311	0.699	0.89	0.996	2
F2	0.513	0.487	0.98	1.08	9.2	0.754	0.325	0.99	1.08	8.3	0.332	0.712	1.001	1.02	2
AND	0.432	0.402	0.845	1.05	19.52	0.665	0.399	0.92	1.08	14.8	0.321	0.653	0.892	1.02	12.8
OR	0.523	0.512	1.41	1.6	11.8	0.698	0.372	1.61	1.82	11.53	0.301	0.754	1.5	1.82	17.5
NAND	0.512	0.554	0.98	1.68	41.6	0.711	0.412	1.5	1.8	16.5	0.299	0.814	0.89	1.6	44.1
NOR	0.599	0.578	1.41	1.68	16.01	0.654	0.358	1.7	1.8	5.5	0.332	0.701	1.65	1.7	3.0
XOR	0.523	0.541	1.5	1.86	19.35	0.789	0.325	1.3	1.68	22.6	0.351	0.654	1.78	1.86	4.3
MUX	0.522	0.612	2.22	2.55	12.9	0.654	0.411	1.78	1.82	2.19	0.452	0.685	1.71	1.81	5.5

	Simulated Power (µW)			Model Power (µW)			% Deviation			#Tr*Simulated Delay			#Tr*Model Delay		
	un-skew	**Hi-skew**	**Lo-skew**	**un-skew**	**Hi-skew**	**Lo-skew**	**un-skew**	**Hi-skew**	**Lo-skew**	**un-skew**	**Hi-skew**	**Lo-skew**	**un-skew**	**Hi-skew**	**Lo-skew**
F1	20.2	24.23	22.18	18.36	22.58	20.15	-10.0	-7.3	-10.0	3.96	4.8	3.96	4.36	5.04	4.36
F2	22.32	24.98	21.65	20.15	22.56	20.87	-10.7	-10.7	-3.7	3.92	3.96	4.004	4.32	4.32	4.08
AND	15.16	16.95	15.45	13.25	14.65	13.56	-14.41	-15.6	-13.9	3.38	3.68	3.56	4.2	4.32	4.08
OR	20.13	22.87	21.98	18.58	20.51	19.85	-8.36	-11.5	-10.7	5.64	6.44	6.0	6.4	7.28	7.28
NAND	20.23	23.51	21.22	18.47	21.56	19.72	-9.51	-9.1	-7.6	5.88	9.0	5.34	10.08	10.8	9.6
NOR	22.13	24.89	21.71	20.50	22.89	19.87	-7.9	-8.3	-9.2	8.46	10.2	9.9	10.08	10.8	10.2
XOR	33.67	34.25	30.89	30.25	32.72	28.45	-11.3	-4.6	-8.6	9.0	7.8	10.68	11.16	10.08	11.16
MUX	59.31	61.23	60.54	57.56	59.65	58.54	-3.4	-2.6	-3.4	4.44	3.56	3.42	5.1	3.64	3.62

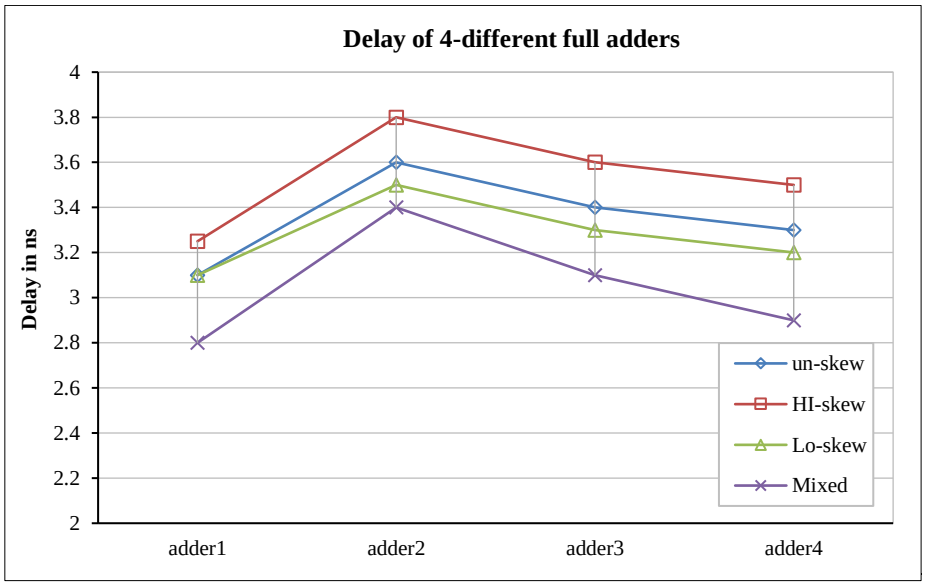
The design of the 4 different full adders that uses un-skewed, skewed and mixed logic is simulated, keeping the sum topology construction same for all logic and varying the carry topology in the adder circuit. The simulation is performed at the same condition taking the characteristics delay as 100ps and branching effort is 1. The delay is calculated for the highest gate count in the path of the network. The simulated results are depicted in Table 13 and the performance parameters of adder circuits are presented in Figure 24. The delay of 4-different full adders is shown in Figure 24-a and it is noticed that the adders designed using Hi-skewed exhibit the highest delay. In contrast, the full adder which uses un-skewed and Lo-skewed circuits produces nearly equal delays while the adders in mixed logic provide a minimum delay in all cases. From the proposed full adders, the adder1 implemented through AND and OR gates produce less delay when compared to adder2, adder3 and adder4.

Table 13. Performance of various full adder circuits in terms of un-skewed and skewed gates

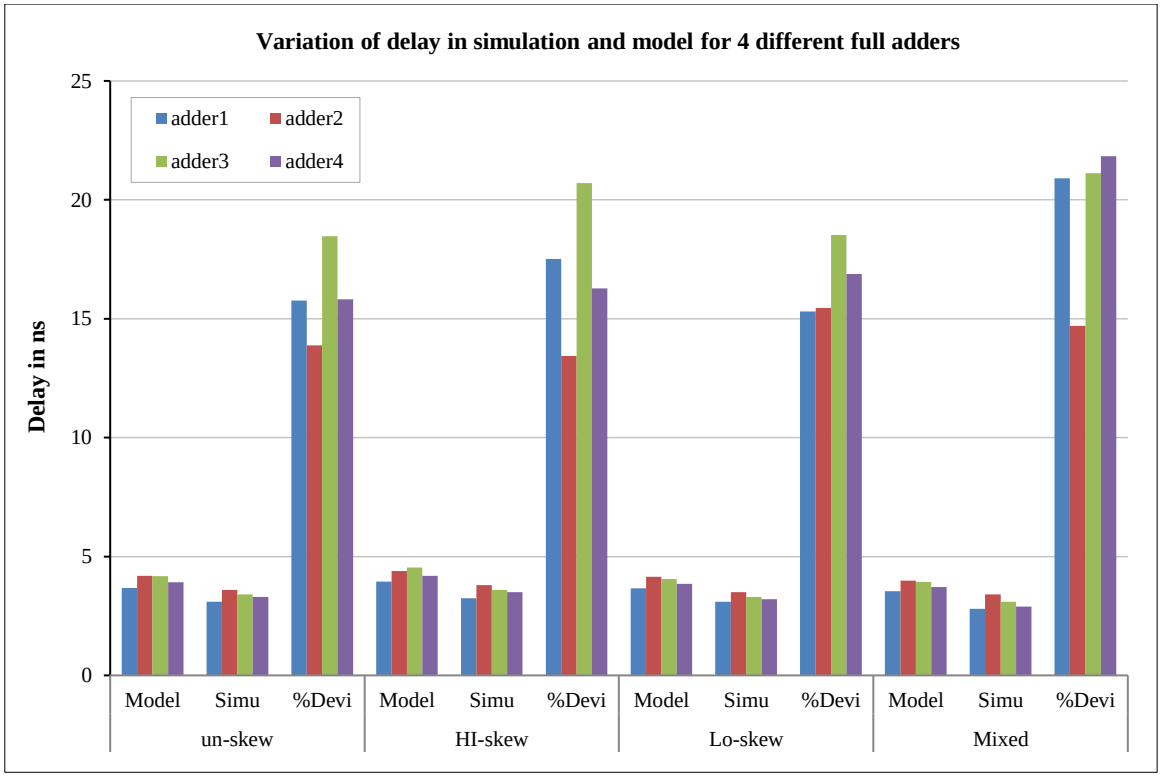
		Un-skew				Hi-skew				LO-skew				Mixed			
		B=1 and H=5															
	Full adder	**g_{GDI}**	**P_{GDI}**	**N**	**M-D**	**g_{GDI}**	**P_{GDI}**	**N**	**M-D**	**g_{GDI}**	**P_{GDI}**	**N**	**M-D**	**g_{GDI}**	**P_{GDI}**	**N**	**M-D**
1	sum	1.49	6	3	21.3	1.76	6	3	22.23	1.36	6	3	20.89	1.12	6	3	19.96
	carry	0.53	5	4	15.5	0.98	5	4	17.25	0.58	5	4	15.75	0.53	5	4	15.5
2	sum	1.49	6	3	21.3	1.76	6	3	22.3	1.36	6	3	20.8	1.12	6	3	19.6
	carry	0.55	9	6	20.4	0.99	9	6	21.6	0.57	9	6	20.5	0.39	9	6	19.9
3	sum	1.49	6	3	21.3	1.76	6	3	22.2	1.36	6	3	20.8	1.12	6	3	19.9
	carry	1.02	8	4	20.3	2.33	8	4	23.2	0.81	8	4	19.6	0.73	8	4	19.3
4	sum	1.49	6	3	21.3	1.76	6	3	22.2	1.5	6	3	20.8	1.12	6	3	19.9
	carry	0.86	6	4	17.6	1.5	6	4	19.6	1.36	6	4	17.6	0.68	6	4	17.1

gGDI – Logical Effort; PGDI - Parasitic delay; N – No of stages (gates in a path); M-D – Model delay T-D – Total delay (sum, carry); S-D – Simulation delay %Dev – Percentage Deviation.

	Un-skew				Hi-skew				LO-skew				Mixed			
FA	**M-D**	**T-D**	**S-D(ns)**	**%Dev**	**M-D**	**T-D**	**S-(ns)**	**%Dev**	**M-D**	**T-D**	**S-D(ns)**	**%Dev**	**M-D**	**T-D**	**S-D(ns)**	**%Dev**
1	21.3 15.5	36.8 3.68	3.1	15.7	22.3 17.2	39.4 3.94	3.25	17.5	20.8 15.7	36.6 3.66	3.1	15.3	19.9 15.5	35.4 3.54	2.8	20.9
2	21.3 20.4	41.8 4.18	3.6	13.8	22.3 21.6	43.9 4.39	3.8	13.7	20.8 20.5	41.4 4.14	3.5	15.4	19.9 19.9	39.86 3.98	3.4	14.7
3	21.3 20.3	41.7 4.17	3.4	18.4	22.2 20.8	45.4 4.54	3.6	20.7	20.8 19.6	40.5 4.05	3.3	18.5	19.9 19.3	39.3 3.93	3.1	21.1
4	21.3 17.8	39.2	3.3	15.8	22.2 19.6	41.8	3.5	16.2	20.8 17.6	38.5	3.2	16.8	19.9 17.1	37.1	2.9	21.8



(a) Delay of full adders in (ns)



(b) Variation of delay through simulation and model

Figure 24. Performance of 4-different full adders in un-skewed, skewed and mixed logic

The delay variation in terms of simulation and estimation is shown in Figure 24-b. The deviation is observed to be a minimum of 13.8 for un-skewed adder2 and a maximum of 21.8 for mixed adder4. This illustration explains the delay reduction of the full adder using mixed logic. However, mixed logic also produces increased static power consumption compared to un-skewed gates due to unequal rise and fall time. Hence, the mixed logic is suitable for high-speed circuits when the power consumption factor is tolerable.

The performance of the ISCAS 85 74-X series combination bench mark circuit in terms of proposed delay-power model and simulation is reported in Table 14. *Gate count/transistor represents the gate count and number of transistors in CMOS logic. **Gate count presents the number of gates and transistors including buffers. The analysis of this circuit in terms of model and simulation delay-power is shown in Figure 25. The findings indicate that nearly 22% variation in delay and 15% variation in power exists between model and simulation. The performance of the proposed delay-power model is evaluated for ISCAS combinational bench mark and its performance deviation between simulation and estimation has been reported.

Table 14. Subset of ISCAS 85 - 74X-Series combinational benchmark Circuits

	Circuit Name	Circuit function	Inputs	Outputs	**Gate count for GDI/ Transistors	Delay (1.2V) T=100 ps		Power (1.2V)	
						Model (ns)	Simulation (ns)	Model	Simulation (uW)
Proposed	**74182**	4-bit carry-look ahead generator	9	4	29/198	4.8	3.5	402	453.3
Ponnian et al. [24]			9	4	34/224	5.6	4.8	506	499.2
Proposed	**74283**	4-bit adder	9	5	62/298	5.1	4.1	523	550.98
Ponnian et al. [24]			9	5	86/348	7.2	6.8	765	722.2
Proposed	**74181**	4-bit ALU	14	8	120/558	5.6	4.7	692	783.67
Ponnian et al. [24]			14	8	160/882	8.2	7.8	865	802.3
Proposed	**74L85**	4-bit magnitude comparator	11	3	64/278	5.3	4.3	501	522.12
Ponnian et al. [24]			11	3	82/334	6.8	6.2	632	602.4

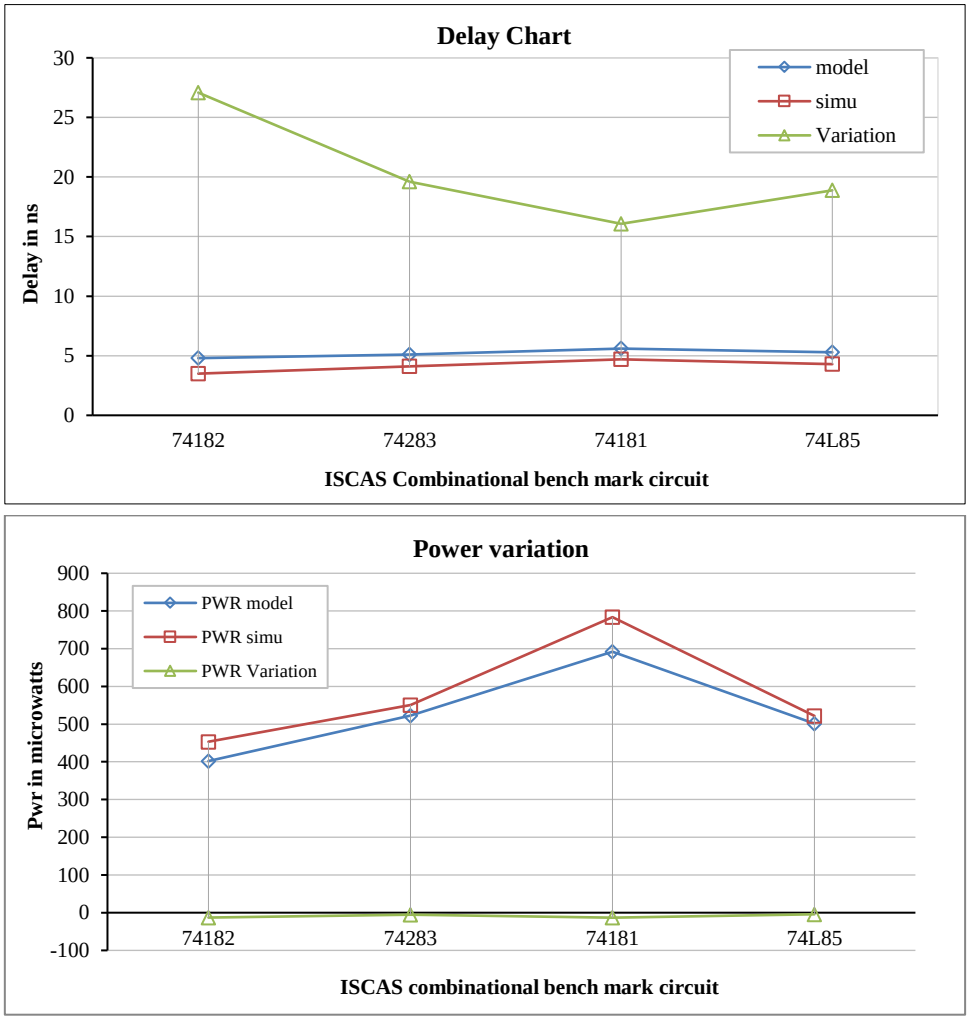


Figure 25. Analysis of 74-X ISCAS combinational bench mark circuit

10- Conclusion

This work investigates an innovative synthesis algorithm for GDI technology using a MUX based decomposition algorithm. The experimental results of the proposed research show its superiority in power-delay concerning CMOS and PTL logic. The propounded delay model, which uses the Logical-based approach, is estimated to be compact and simple. The delay of the circuit is characterized by three components, which are logical effort, electrical effort, and parasitic effort. These components are obtained through internal capacitances (gate and diffusion), input capacitance, output load capacitance, and pull-up – pull-down resistors. In a single gate, the absolute delay is approximated as $\tau(g_{GDI}H_{GDI}+p_{GDI})$ where τ is the characteristic delay for a technology. A multistage GDI network buffer is inserted for each successive three GDI cells to restore the threshold drop. Similarly, a level restoring circuit is also supplemented at the output node at complete swing voltage. The delay calculation for un-skewed and skewed gates for the GDI, EGDI, and proposed GDI, EGDI is reported for logical effort and parasitic delay for single and multistage networks. The proposed power model consists of dynamic, short-circuit, and leakage power. The dynamic power is expressed in the node transition activity factor $\alpha_{0\to1}$ and capacitive power, which is resulted due to the discharge and charge of the driver and driving gate. The total capacitive power is obtained through the summation of the driver and driving gate's internal, wiring, and gate capacitances. The $\alpha_{0\to1}$ of the proposed GDI cells ranges from 0.1–0.2, whereas in CMOS it is 0.1–0.3. Table 5 clearly depicts the notion that the node activity factors of the proposed GDI cells have a considerable amount of improvement over CMOS logic. This proves that the GDI technique is indeed superior to CMOS, PTL, and CPL and substantiates that the logic style reduces power dissipation.

11- Declarations

11-1-Author Contributions

Conceptualization, J.P. and U.R.; methodology, J.P.; software, J.P.; validation, J.P., S.P., and C.P.O.; formal analysis, U.R.; investigation, S.P.; resources, C.P.O.; data curation, U.R.; writing—original draft preparation, J.P.; writing—review and editing, J.P.; visualization, S.P.; supervision, S.P.; project administration, C.P.O.; funding acquisition, C.P.O. All authors have read and agreed to the published version of the manuscript.

11-2-Data Availability Statement

The data presented in this study are available in the article.

11-3-Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

11-4-Institutional Review Board Statement

Not applicable.

11-5-Informed Consent Statement

Not applicable.

11-6-Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

12- References

[1] Bisdounis, L., Nikolaidis, S., Koufopavlou, O., & Goutis, C. (1998). Switching response modeling of the CMOS inverter for sub-micron devices. Proceedings Design, Automation and Test in Europe. doi:10.1109/date.1998.655939.

[2] Nabavi-Lishi, A., & Rumin, N. C. (1994). Inverter Models of CMOS Gates for Supply Current and Delay Evaluation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 13(10), 1271–1279. doi:10.1109/43.317470.

[3] Bhattacharyya, A. B., & Ulman, S. (2002). PREDICTMOS MOSFET model and its application to submicron CMOS inverter delay analysis. Proceedings of ASP-DAC/VLSI Design 2002. 7th Asia and South Pacific Design Automation Conference and 15h International Conference on VLSI Design. doi:10.1109/aspdac.2002.994920.

[4] Chandra, N., Yati, A. K., & Bhattacharyya, A. B. (2009). Extended-Sakurai-Newton MOSFET Model for Ultra-Deep-Submicrometer CMOS Digital Design. 2009 22nd International Conference on VLSI Design. doi:10.1109/vlsi.design.2009.48.

[5] Conti, M., Orcioni, S., Turchetti, C., Soncini, G., & Zorzi, N. (1996). Analytical Device Modeling for MOS Analog IC's Based on Regularization and Bayesian Estimation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 15(11), 1309–1322. doi:10.1109/43.543764.

[6] Austin, B. L., Bowman, K. A., Xinghai Tang, & Meindl, J. D. (2002). A low power transregional MOSFET model for complete power-delay analysis of CMOS gigascale integration (GSI). Proceedings Eleventh Annual IEEE International ASIC Conference (Cat. No.98TH8372). doi:10.1109/asic.1998.722816.

[7] Galup-Montoro, C., Schneider, M. C., & Pahim, V. C. (2005). Fundamentals of Next Generation Compact MOSFET Models. 2005 18th Symposium on Integrated Circuits and Systems Design. doi:10.1109/sbcci.2005.4286828.

[8] Sutherland, I., Sproull, R. F., & Harris, D. (1999). Logical effort: designing fast CMOS circuits. Morgan Kaufmann, Burlington, United States.

[9] Vemuru, S. R., & Scheinberg, N. (1994). Short-Circuit Power Dissipation Estimation for CMOS Logic Gates. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 41(11), 762–765. doi:10.1109/81.331533.

[10] Hamoui, A. A., & Rumin, N. C. (2000). An analytical model for current, delay, and power analysis of submicron CMOS logic circuits. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 47(10), 999–1007. doi:10.1109/82.877142.

[11] Hauser, J. R. (2005). A new and improved physics-based model for MOS transistors. IEEE Transactions on Electron Devices, 52(12), 2640–2647. doi:10.1109/TED.2005.859623.

[12] Bisdounis, L., & Koufopavlou, O. (2000). Short-circuit energy dissipation modeling for sub micrometer CMOS gates. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 47(9), 1350–1361. doi:10.1109/81.883330.

[13] Consoli, E., Giustolisi, G., & Palumbo, G. (2011). An ultra-compact MOS model in nanometer technologies. 2011 20th European Conference on Circuit Theory and Design (ECCTD). doi:10.1109/ecctd.2011.6043403.

[14] Nose, K., & Sakurai, T. (2000). Analysis and future trend of short-circuit power. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 19(9), 1023–1030. doi:10.1109/43.863642.

[15] Sakurai, T., & Newton, A. R. (1990). Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas. IEEE Journal of Solid-State Circuits, 25(2), 584–594. doi:10.1109/4.52187.

[16] Verma, P., Sharma, A. K., Pandey, V. S., Noor, A., & Tanwar, A. (2016). Estimation of leakage power and delay in CMOS circuits using parametric variation. Perspectives in Science, 8, 760–763. doi:10.1016/j.pisc.2016.06.081.

[17] Nandyala, V. R., & Mahapatra, K. K. (2016). A circuit technique for leakage power reduction in CMOS VLSI circuits. 2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA). doi:10.1109/vlsi-sata.2016.7593044.

[18] Kumar, A.P., Aditya, B.L.V.S.S., Sony, G., Prasanna, C., & Satish, A. (2019). Estimation of Power and Delay in CMOS Circuits Using Leakage Control Transistors. Recent Advances in Material Sciences. Lecture Notes on Multidisciplinary Industrial Engineering, Springer, Singapore. doi:10.1007/978-981-13-7643-6_61.

[19] Xue, H., & Ren, S. (2017). Low power-delay-product dynamic CMOS circuit design techniques. Electronics Letters, 53(5), 302–304. doi:10.1049/el.2016.4173.

[20] Saravanan, V., Anpalagan, A., & Woungang, I. (2015). An energy-delay product study on chip multi-processors for variable stage pipelining. Human-Centric Computing and Information Sciences, 5(1). doi:10.1186/s13673-015-0046-x.

[21] Zhao, Q., Sun, W., Zhao, J., Feng, L., Xu, X., Liu, W., Guo, X., Liu, Y., & Yang, H. (2017). Noise Margin, Delay, and Power Model for Pseudo-CMOS TFT Logic Circuits. IEEE Transactions on Electron Devices, 64(6), 2635–2642. doi:10.1109/TED.2017.2695527.

[22] Zhao, Q., Liu, Y., Zhao, J., Guo, X., Li, H., & Yang, H. (2016). Noise margin modeling for zero-VGS load TFT circuits and yield estimation. IEEE Transactions on Electron Devices, 63(2), 684–690. doi:10.1109/TED.2015.2506722.

[23] Han, Z. (2021). The power-delay product and its implication to CMOS Inverter. Journal of Physics: Conference Series, 1754(1), 012131. doi:10.1088/1742-6596/1754/1/012131.

[24] Ponnian, J., Pari, S., Ramadass, U., & Pun, O. C. (2021). A New Systematic GDI Circuit Synthesis Using MUX Based Decomposition Algorithm and Binary Decision Diagram for Low Power ASIC Circuit Design. Microelectronics Journal, 108, 104963. doi:10.1016/j.mejo.2020.104963.

[25] Uma, R., Ponnian, J., & Dhavachelvan, P. (2017). New low power adders in Self Resetting Logic with Gate Diffusion Input Technique. Journal of King Saud University - Engineering Sciences, 29(2), 118–134. doi:10.1016/j.jksues.2014.03.006.

[26] Uma, R., & Dhavachelvan, P. (2012). Modified Gate Diffusion Input Technique: A New Technique for Enhancing Performance in Full Adder Circuits. Procedia Technology, 6, 74–81. doi:10.1016/j.protcy.2012.10.010.

[27] Ramadass, Uma., & Dhavachelvan, P. (2012). New low power delay element in self-resetting logic with modified Gated Diffusion Input technique. 2012 10th IEEE International Conference on Semiconductor Electronics (ICSE). doi:10.1109/smelec.2012.6417197.

[28] Morgenshtein, A., Fish, A., & Wagner, I. A. (2002). Gate-diffusion input (GDI): A power-efficient method for digital combinatorial circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 10(5), 566–581. doi:10.1109/TVLSI.2002.801578.

[29] Morgenshtein, A., Yuzhaninov, V., Kovshilovsky, A., & Fish, A. (2014). Full-swing gate diffusion input logic - Case-study of low-power CLA adder design. Integration, the VLSI Journal, 47(1), 62–70. doi:10.1016/j.vlsi.2013.04.002.