

Analysis and logic optimization using logical effort technique of static CMOS circuits

Akansha Rajput, Satyendra Sharma

Department of Electronics and Communication Engineering
Noida Institute of Engineering & Technology,
Greater Noida, 201310

Email: akansha493@gmail.com, satyendracommn@gmail.com

Abstract— Power dissipation became a major challenge in Integrated Circuit (IC) design for both high-performance and portable applications. In the high-performance and high-density chips such as microprocessors, high power dissipation limits the number of on-chip transistors and increases the required heat removal, which tends to lower the performance and increase the system cost, size and weight.

Logical effort technique gives the gate sizing scheme that minimizes the delay at the lowest cost of power or it minimizes the power dissipation for a given delay budget. Designing a circuit to achieve the greatest speed or to meet a delay constraint presents a bewildering array of choices. Which of several circuits that produce the same logic function will be fastest? How large should a logic gate's transistors be to achieve least delay? Sometimes, adding stages to a path reduces its delay

In proposed work, I have implemented logical effort technique in static CMOS circuits like conventional adder, array multiplier, decoder and multiplexer. These circuits are used very frequently in many bigger circuits. So if I change or adjust its transistors sizing such that its delay and PDP reduce then as a result of this bigger circuits also get the benefit of this changes.

Keywords- Transistor sizing, Logical effort, Optimization, Dynamic power dissipation, static CMOS circuit

1 INTRODUCTION :-

In this era the requirement of compact and portable devices has explored to develop new design techniques that minimize delay, power and area of the circuits. And also the growth of the electronics market has driven the VLSI industry towards very high integration density and system on chip designs and beyond few GHz operating frequencies, critical concerns have been arising to the severe increase in power consumption and the need to further reduce it. Moreover, with the explosive growth the demand and popularity of portable electronics is driving designers to strive for smaller silicon area, higher speeds, longer battery life, and more reliability. Nowadays there are an ever-increasing number of portable applications requiring low power and high throughput circuits. Therefore, low-power design has become a major design consideration.

Power dissipation became a major challenge in Integrated Circuit (IC) design for both high-performance and portable applications. In the high-performance and high-density chips such as microprocessors, high power dissipation limits the number of on-chip transistors and increases the required heat removal, which tends to lower the performance and increase the system cost, size and weight. On the other hand, high power dissipation in battery-operated portable devices such as laptops and cellular phones reduces the battery operation duration and life time as well as increases the battery size and weight. This is important especially with the projected slower improvement in the battery-technology compared to the progress pace of the semiconductor industry [1]. Thus, power estimation, analysis and optimization are essential for CMOS IC design. Using circuit simulators such as Spice to predict the power dissipation in large circuits is an unfeasible solution due to large computing-time. Hence, developing accurate power models is necessary for designing and optimizing very large scale integrated (VLSI) CMOS circuits.

In CMOS circuits there are two sources for power dissipation: static and dynamic. Static power dissipation is mainly due to standby leakage current [2, 3] and it is not a function to the switching frequency of a CMOS gate. This source is out of the scope of this work. Dynamic power, in contrast, is the power consumed by a CMOS gate when its output toggles between high and low logic levels [4, 5]. Short-circuit power and switching power are the main components of the dynamic power dissipation. The first component is produced by the direct DC path between the supply voltage and ground when both the nMOS and pMOS transistors are ON during the input transition. Switching power, on the contrary, contributes the major portion of the power consumption in CMOS circuits, and is the result of the charging and discharging of the output capacitance.

Reducing the power dissipation in IC designs was always a key concern and the force behind moving from one technology to another. Under specific delay constraints, power may be reduced at different levels of the design abstractions. At the circuit level, which is the target of this paper, power optimization is achieved by transistor sizing, supply voltage and/or threshold voltage scaling.

The works in [6–9] attempt to optimize switching power through transistor sizing. Turgis et al. [6] consider a chain of inverters where a tapering ratio of 4.25 is found to minimize the power dissipation. In [7] it has been proven that the sum of the input capacitances of an inverter chain is minimized when inverters bear the same fan out. For a path with general gates the minimal energy solution was obtained in [8] by numerically solving a set of equations, which was resulted from LaGrange method. BiCMOS circuits were considered in [9]. This method uses an iterative process to size and optimize the design's gates where the high drive capability buffered gates (i.e., BiCMOS) with sufficiently low fan-out are identified and replaced with a lower power unbuffered (i.e., CMOS) version. This work seeks the minimization of network delay subject to network power dissipation.

Optimizing the supply voltage to reduce the power dissipation was the target of many researchers. Considering micro-processors, Cai et al. [10] Propose a dual supply voltage technique to reduce both the static and the dynamic power dissipation of CMOS circuits. Low supply voltage and low threshold-voltage devices are used for high activity circuits while higher supply voltage and high threshold voltage devices are assigned to the low activity circuitry. In [11] the power optimization has been achieved in two steps. First, maximum delay is assigned to all gates then in the next step each individual gate is optimized iteratively for minimum power by finding the proper combination of the transistor widths; as well as threshold and supply voltages.

This paper survey of logical effort technique in section II, after that in section III shows modified work – logical implementation and simulation result and in section iv concludes this paper

II.SURVEY OF LOGICAL EFFORT TECHNIQUE :-

The method of logical effort is a simple and quick method that estimates the delay in CMOS circuits as normalized to the delay of a unit inverter (τ). In the logical effort technique, τ is defined as the delay of unit-sized inverter driving an identical inverter with no parasitic.

1.1 Logical Effort for individual gates :-

Logical effort technique [12] expresses the delay of CMOS gates D as a normalized value of τ ,

$$D = \tau d, \quad (1)$$

$$d = f + p, \quad (2)$$

$$f = gh, \quad (3)$$

Where d is the gate's normalized delay, f is the gate effort, and p is the parasitic delay. The gate effort is partitioned into two components: the logical effort g and the electrical effort h . The logical effort captures the effect of gate topology on its driving ability and is defined as the ratio of the input capacitance of a template gate (C_{tg}) to the input capacitance of the unit inverter (C_v),

$$\text{So, } g = \frac{C_{tg}}{C_v}, \quad (4)$$

On the other hand, the electrical effort is defined as the ratio of the gate's load capacitance (C_{lg}) to the gate's input capacitance (C_{ing}),

$$\text{So, } h = \frac{C_{lg}}{C_{ing}}, \quad (5)$$

The parasitic delay is given as the ratio of the parasitic capacitance of template gate's (C_{tp}) to the parasitic capacitance of the unit inverter (C_v).

$$\text{So, } p = \frac{C_{tp}}{C_v}, \quad (6)$$

The template gate is defined as gate that is sized to deliver the same output current of the unit inverter.

1.2 Logical effort for logical path:-

The logical effort along a path compounds by multiplying the logical efforts of all the logic gates along the path. We use the uppercase symbol G to denote the *path logical effort*, so that it is distinguished from g , the logical effort of a single gate in the path. So,

$$G = \prod g_i, \quad (7)$$

Where subscript i index the logical stages along the path.

The electrical effort along the path through a network is simply the ratio of the capacitance that loads the logic gate in the path to the input capacitance of the first gate in the path.

$$H = \frac{C_{out}}{C_{in}}, \quad (8)$$

2. Switching power dissipation:-

Switching power dissipation refers to the power consumed by a CMOS gate as a result of the charging and discharging of the gate's output capacitance, henceforth, we call it power dissipation. This source contributes most of the power consumed in current CMOS circuits. It is widely accepted that power dissipation of a CMOS gate is given as

$$P_{sw} = \alpha_g f_{clk} C_{out} VDD^2 \quad (9)$$

Where f_{clk} is the clock frequency, α_g is the active factor of the gate, and C_{out} is the output capacitance of the gate, since the C_{out} consists mainly of two components: the load capacitance (C_{lg}) and the parasitic capacitance (C_{pg}), then can be recast into

$$P_{sw} = \alpha_g f_{clk} (C_{lg} + C_{pg}) VDD^2 \quad (10)$$

To get rid of the mathematical complexity and to achieve a good accuracy at the same time, power dissipation of static CMOS gates can be modeled as normalized to the power dissipation of the unit CMOS inverter. In this work, the unit inverter is considered to have a minimum width nMOS transistor and a twice the minimum size pMOS transistor, which aligns with the logical effort technique.

The load capacitance (C_{lg}) can be described in terms of the gate input capacitance (C_{ing}) and the electrical effort as

$$C_{lg} = h C_{ing} \quad (11)$$

Substituting (3) in (2) yields

$$P_{sw} = \alpha_g f_{clk} VDD^2 (h C_{ing} + C_{pg}) \quad (12)$$

Let P_v be the switching power dissipation of the unit inverter.

$$P_v = \alpha_v f_{clk} VDD^2 C_v \quad (13)$$

Where, α_v is the activity factor of the unit inverter. Thus, the normalized switching power (P_{nm}) of a CMOS gate to P_v can be described as

$$P_{nm} = \frac{P_{sw}}{P_v} = \frac{\alpha_g}{\alpha_v} \left(\frac{C_{ing}}{C_v} h + \frac{C_{pg}}{C_v} \right) \quad (14)$$

Since the input and parasitic capacitances of a CMOS gate are proportional to the widths of the transistors (assuming minimum length), C_{ing} and C_{pg} can be written as

$$C_{ing} = Z C_{tg} \text{ And } C_{pg} = Z C_{tp} \quad (15)$$

in (6) and using logical effort terminology yields

$$P_{nm} = \alpha_{nm} Z (gh + p) \quad (16)$$

Where α_{nm} is the normalized activity factor and is given as

$$\alpha_{nm} = \frac{\alpha_g}{\alpha_v} \quad (17)$$

3. Model validation:-

The performance of the developed model has been tested by comparing its results with the simulation ones when UMC 0.13um process and the predictive high-k 45nm process parameters [24] are targeted. To determine the value of g and p parameters, they have been expressed as [1]

$$g = \kappa g_{eff} \quad p = p_{eff} + \lambda \quad (18)$$

Where g_{eff} and p_{eff} are equivalent to the logical effort and parasitic delay of the logical effort technique and κ and λ are process dependent parameters.

Table given below shows the value of g and p for the target UMC 0.13um and P45 processes where κ and λ have been found to be $\kappa = 0.95$ and $\lambda = 0.38$ for UMC 0.13um; and $\kappa = 1.57$ and $\lambda = 0.9$ for the P45nm process.

Table 1:- extracted value of g and p for selected gates considering UMC 0.13 and P45nm.

Process	Gate	INV	NAND2	NAND3	NOR2	NOR3	GATE
UMC 0.13um	g	0.95	1.2	1.5	1.6	2.2	2.7
	p	1.45	3.45	3.45	2.45	3.45	2.78
P45nm	g	1.57	2.1	2.6	2.6	3.7	3.1
	p	1.9	2.9	3.9	2.9	3.9	3.2

Also, the value of z has been expressed as $z = \frac{W_g}{W_{tp}}$ where W_g and W_{tp} are the widths of the gate and its

template, respectively, and η is a constant that describes the change of a CMOS gate input capacitance due to the change of the gate's size as compared to its template. It has been found that $\eta = 0.8$ for UMC 0.13 nm and $\eta = 1$ for the P45 nm process. It is important to notice that κ , λ and η are process dependent, but not gate nor transistor dependent.

For estimating the switching power I have done it by hand calculation using logical effort model. For example, if we take inverter with 2 fan-out at $z = 5$, in UMC 0.13um, so $g = 0.95$, $h=2$, $z = 0.8*5=4$, $p = 1.45$, put all this in equation (3.8), $P_{nm} = 13.4$, similarly other values can also be calculate.

In case of comparison of model value with simulated one, so I have simulate all gates at 1 to 10 fan-out range at minimum, five times of minimum and ten times of minimum size and compare it with model value. I have made simulated values normalized to the unit inverter. This is done by calculating value of power of the unit inverter.

I have compared the model value with simulated one in which I have used two values of aspect ratio (γ) 2 and 2.5.

Table 2:- Comparison of model value and simulated value of inverter at UMC 0.13um.

FANOUT	NOR POWER AT Y=2, Z=5			NOR POWER AT Y=2.5, Z=5		
	SIM.	MOD.	DIFF.	SIM.	MOD.	DIFF.
1	9.75	9.6	.15	9.78	9.6	.18
2	13.59	13.4	.2	12.59	13.4	.8
3	18.2	17.2	1	18.5	17.2	1.3
4	22.30	21.0	1.3	22.5	21.0	1.5
5	25.97	24.8	1.17	26.1	24.8	1.3
8	37.6	36.2	1.4	37.8	36.2	1.6
10	45	43.8	1.2	45.2	43.8	1.4

Table 3:- comparison of model value and simulated value of inverter at P45nm.

FANOUT	NOR POWER AT Y=2, Z=5			NOR POWER AT Y=2.5, Z=5		
	SIM.	MOD.	DIFF.	SIM.	MOD.	DIFF.
1	17.9	17.4	.5	18.1	17.4	.7
2	26.1	25.2	.9	26.3	25.2	1.1
3	34.3	33.1	1.2	34.4	33.1	1.3
4	42.3	40.9	1.4	42.5	40.9	1.6
5	50.5	48.8	1.7	50.7	48.8	1.9
8	74.0	72.3	1.7	74.0	72.3	1.7
10	89.8	1.8	1.9	89.9	88	1.9

Now i have taken inverter as an example to show graphical view in which comparison is shown for z =1, 5, 10 at UMC 0.13 and P45nm. In this graphical view shown that there is very small error between model value and simulated value.

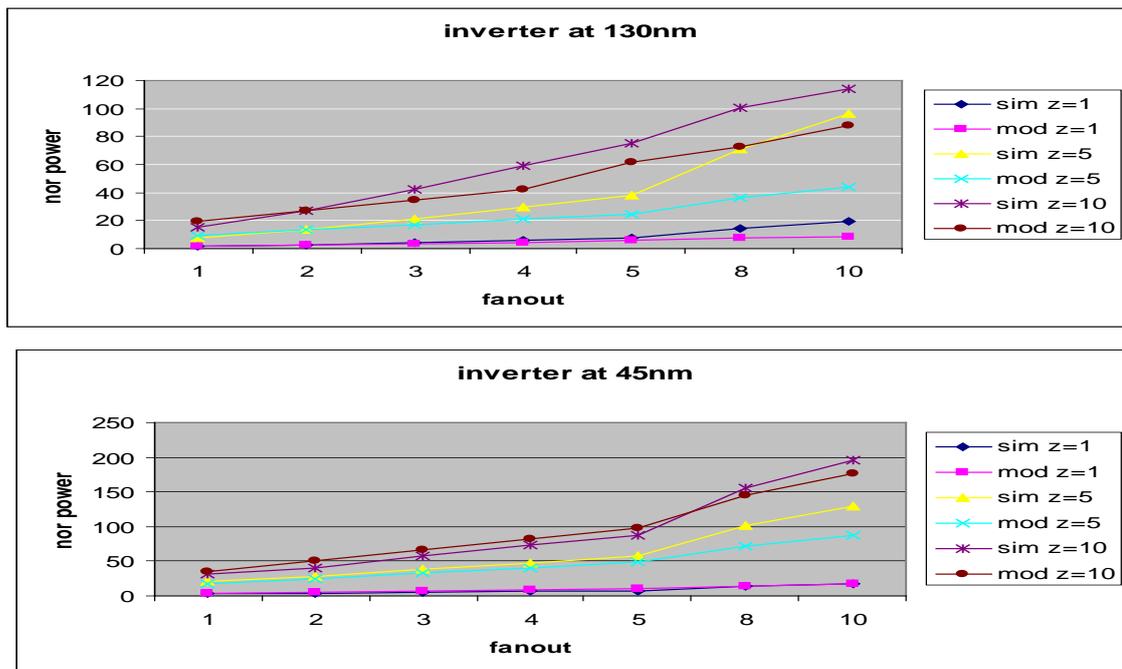


Fig1:-Graphical view of comparison at different sizes of inverter at UMC 0.13 and P45nm.

4. Switching power optimization:-

We know the switching power dissipation of a CMOS gate is given as

$$P_{sw} = \alpha_g f_{clk} C_{out} VDD^2 \tag{19}$$

This equation shows that supply voltage and switching capacitance are the major factors that can be tuned to reduce power dissipation while maintaining a specific operating frequency.

With the fact that switching capacitance can be minimized through a proper transistor sizing, so we analyze power optimization via transistor sizing.

4.1. Transistor size optimization:-

This section examines the optimization of power dissipation through transistor sizing which allows the reduction of the switching capacitance. Transistor sizing problem can be divided into two sub-problems: logic-path gate sizing and gate-transistor sizing. Logic-path gate sizing, determines the sizes of the individual gates in relation to each other, i.e., the sizing scheme of the logic path. Conversely, gate-transistor sizing decides the best actual transistor sizes of each gate to minimize the power while attaining the required speed.

4.1.1. Logic-path gate sizing:-

In this part we are going to show how to minimize the input capacitances of the gates of a path subject to maximum delay of the path. This problem can be articulated as sizing the gates of a logic path to minimize the sum of the path’s gate input capacitances subject to the maximum delay of the path.

Lemma:- Under a maximum delay constrain, the area of a chain of gates is minimized when all the gates have the same effort, i.e.,

$$h_i g_i = h_j g_j \quad \forall (i, j) \in 1 \dots N \tag{20}$$

Proof:- It has been shown in [4] that the input capacitances of a chain of inverters are minimized when all inverters bear the same electrical effort. Nevertheless, logic paths have, in general, different logic gates. For such gates, the value of the input capacitance is determined by the value of the gate’s driving strength (i.e., the gate’s size compared to its template), and the gate’s complexity. For example, an inverter has smaller input capacitance than a 3-input NOR gate when both have the same driving strength. Moreover, if the driving strengths of the inverter and the 3-input NOR gate have changed by the same amount, the input capacitance of the 3-input NOR gate will change more than that of the inverter. To be specific the change will be proportional to the gates’ logical efforts.

For a given chain of gates, the sizing problem aims to determine the driving strength of the gates to achieve the target delay. Thus, the gates can be collapsed into driving-strength equivalent inverters. Since any gate and its equivalent inverter have the same driving strength, the input capacitance of the equivalent inverter (C_{ivq}) can be obtained mathematically via dividing the gate’s input capacitance by the gate’s logical effort

$$C_{ivq} = C_{ing}/g \tag{21}$$

Fig. 3.9 shows a logic path that has two gates, Gate_1 and Gate_2. The equivalent inverter input capacitance of Gate_1 is $C_{ivq1} = C_{ing1}/g1$ and for Gate_2 it is $C_{ivq2} = C_{ing2}/g2$.

It is important to notice that when C_{ing} is implicated as a load, it should not be expressed in terms of C_{ivq} . As a result, the outcome in [4] can be extended to include general CMOS gates; hence, the input gate capacitances of the logic path in Fig. 3.9 are minimized when

$$\frac{C_L}{C_{ing2}/g2} = \frac{C_{ing2}}{C_{ing1}/g1} \tag{22}$$

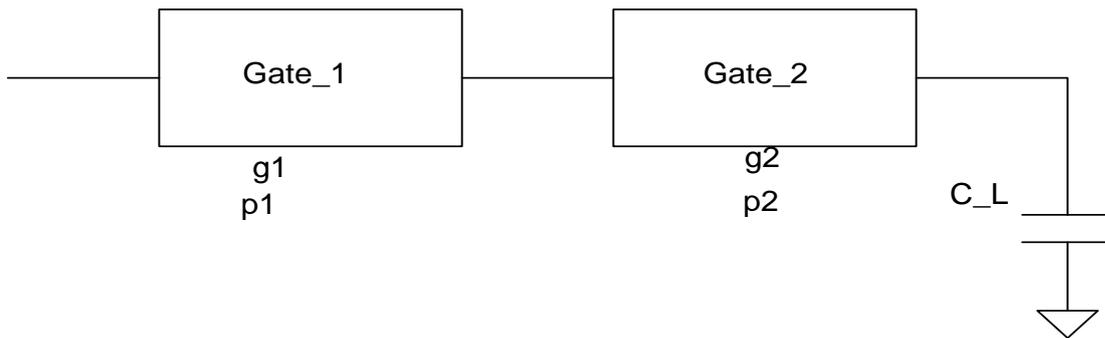


Fig2: A Logic path with two gates

Eq. (22) can be recast into

$$h_1 g_1 = h_2 g_2 \tag{23}$$

Where $h_1 = C_{ing2}/C_{ing1}$ and $h_2 = C_L/C_{ing2}$.

For a logic path with N gates, (23) can be generalized to

$$h_i g_i = h_j g_j \quad \forall (i, j) \in 1 \dots N \tag{24}$$

Comparing (24) to the result in [4] proves the interchange-ability principle of the logical effort [3] which states that two different gates i and j have the same effect on the circuit performance as long as

$$h_i g_i = h_j g_j.$$

This also, extends the result in [4] to be applicable to any chain of similar gates (e.g. NORs, NANDs, etc.) where a chain of inverters is a special case ($g = 1$). More importantly, (24) shows that for a particular logic path with a specific delay budget, the power consumption is minimized when gate efforts of that path are equal. This is the same result obtained by the logical effort to minimize a logic path delay. Therefore, there is no contradiction between designing for high-speed and low power. In other words, logical effort gives the gate sizing scheme that minimizes the delay at the lowest cost of power or it minimizes the power

dissipation for a given delay budget.

Lemma:- The minimum power dissipation (minimum input capacitances) of a logic path that is characterized by a user delay budget (D_{max}), the number of stages and the path input and output capacitances, and is attained when $D_{max} = NF^{1/N} + P_n$.

Proof:- For a chain of inverters, it was proven [4] that the sizes of the inverters are optimized when $D_{max} = D_{LEmin}$. Based on this proof and on the matched solutions of the minimum delay and minimum power problems, one can state that the minimum power is attained when $D_{max} = D_{LEmin}$ as well. Thus, we can write

$$D_{max} = NF^{1/N} + P_n. \quad (25)$$

To compare the performance of the proposed technique with the one in [4], the logic path shown in Fig. 8 has been considered. Different loading conditions have been chosen so the number of stages is less, equal, or more than what is dictated by the logical effort model. Under each of these loading conditions the gates of the path have been sized to have equal fan-out (EFO) as in [4] and to have equal effort delay (EED) as in this paper.

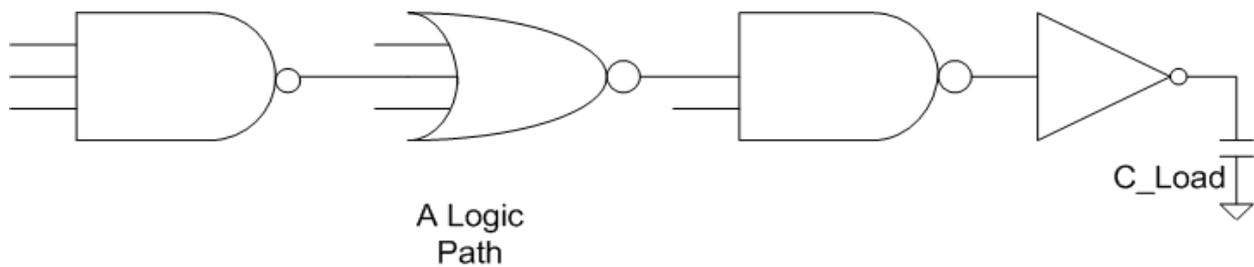


Fig3: Logic path which is used for the analysis.

This is the logic path which is used for the analysis.

Steps for how to calculate size:- Here I have given a brief explanation how to calculate the size for both Equal Fan-Out and Equal Effort Delay.

For EFO:-

- Calculate of c_{in_nand3} .
- Calculate path's electrical effort for a load.
- Calculate equal fanout.
- Calculate c_{in} of other gate corresponds to equal fanout.
- Calculate w of gate corresponds to their c_{in} .

For EED:-

- Calculate of c_{in_nand3} .
- Calculate path's electrical and logical effort for a load and then path effort.
- Calculate equal effort.
- Calculate C_{in} of other gate corresponds to equal fan-out.
- Calculate w of gate corresponds to their C_{in} .

4.1.2. Gate-transistor sizing:-

To analyze the effect of the actual transistor sizes of CMOS gates on the circuit power dissipation and speed, the circuit shown in Fig. 3.11 is considered. In this circuit the inverters have been assigned fixed sizes where INV_D represents the driving circuit and the others (INV_1–INV_M) are the load. The transistor widths of the 3-input NOR gate are swept over a large range during the simulation to determine the value that minimizes power, delay and PDP.

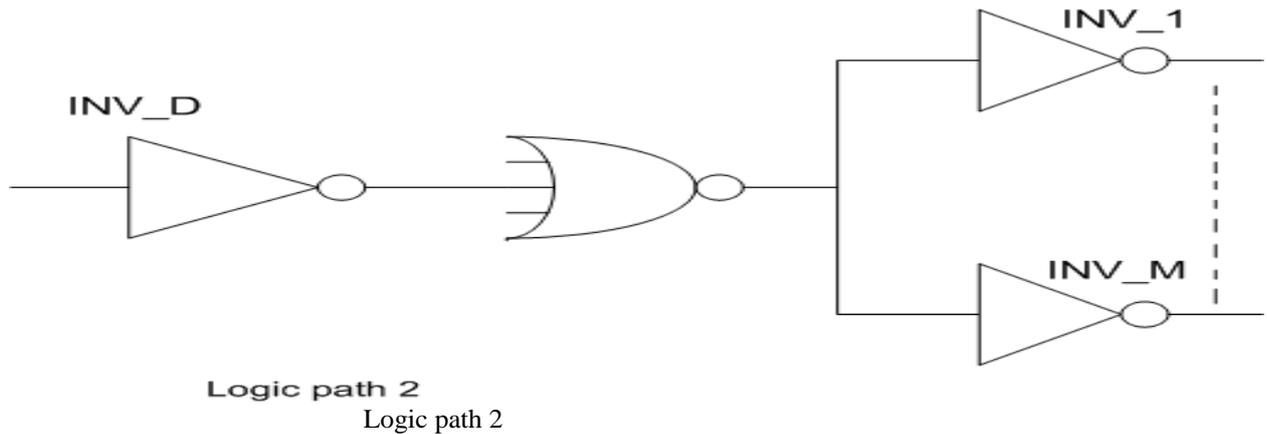


Fig.4:- logic path 2

III. MODIFIED WORK-LOGIC IMPLIMENTATION WITH SIMMULATION RESULT:-

In this work I have implemented the technique of Logical Effort. I have taken well known and popular circuits in digital design systems used logical effort technique on them and compared it with its normal form without the use of this technique. I have implemented logical effort technique in static CMOS circuits like conventional adder, array multiplier, decoder and multiplexer. These circuits are used very frequently in many bigger circuits. So if I change or adjust its transistors sizing such that its delay and PDP reduce then as a result of this bigger circuits also get the benefit of this changes. Logical effort technique mainly deals with scaling of the transistors in such manner that will reduce the delay of the path in which we apply this technique.

So make a module of any circuit with and without use of logical effort technique and

Use this as a basic building block and compare the performance. Here I first used conventional adder. I have taken randomly three different loading conditions, $10fF$, $50fF$ and $100fF$.

Conventional Full Adder: - First of all if I taught about conventional full adder, so the first thing comes in mind that it is a kind of arithmetic function which is used to add two numbers. Here the basic circuit given below:

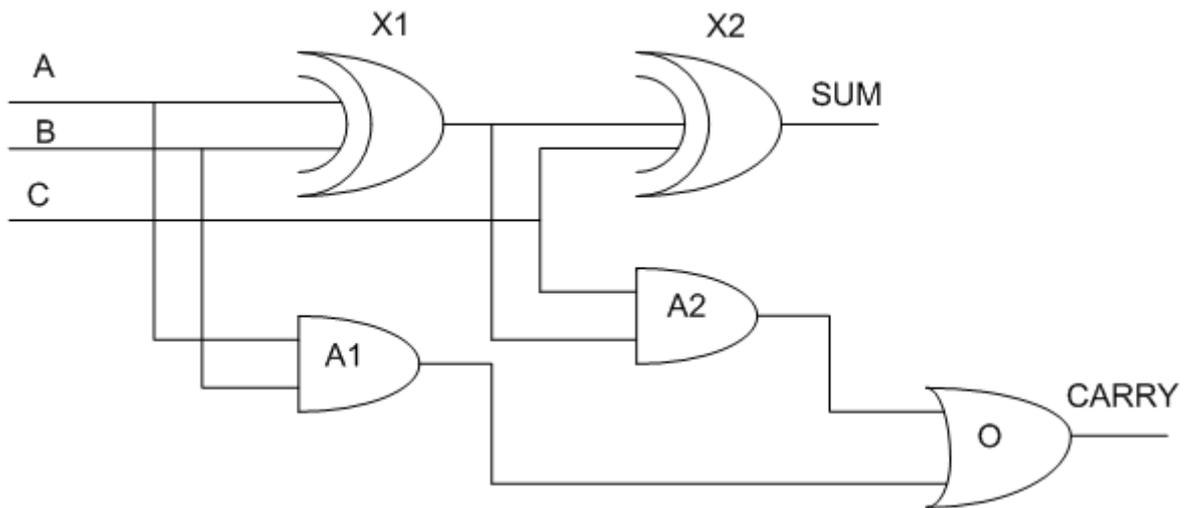
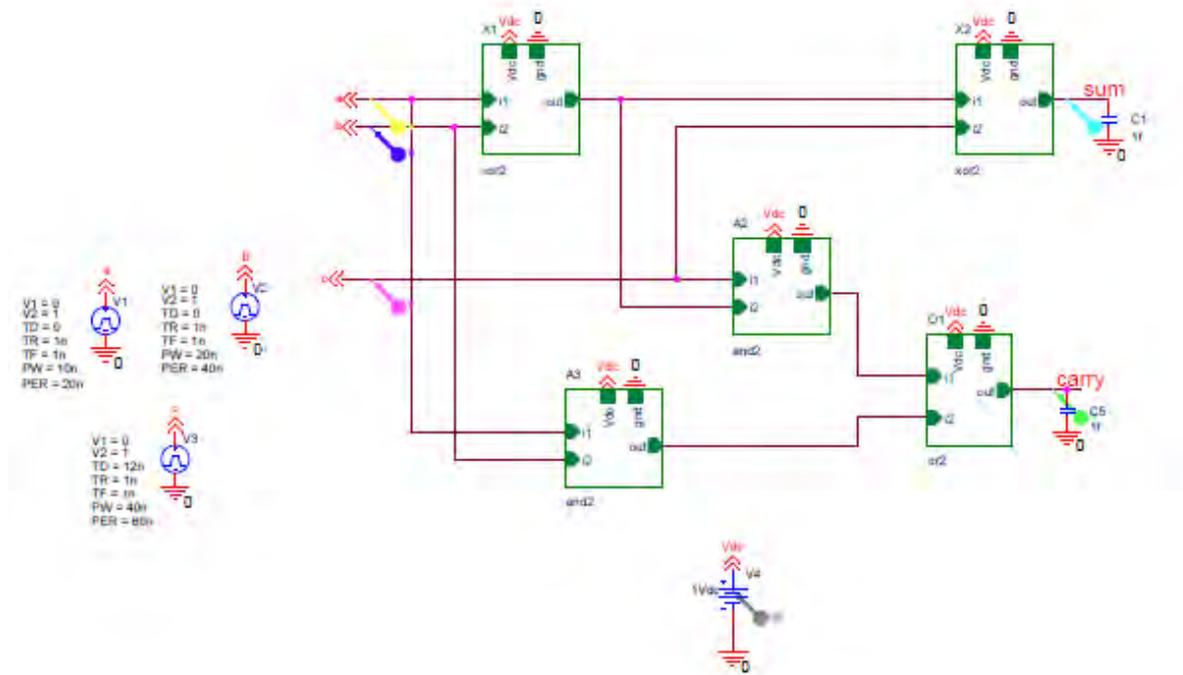


Fig.5:- Conventional Full Adder

Where A and B are two inputs C is input which is carry from previous circuits and SUM and CARRY are two outputs. Here I am not going to fix in details of adder like truth table and other thing because it's well known circuit. Now first I have simulated this circuit without using logical effort technique at different loading conditions. I have used 180nm technology in cadence tool in order to simulate all the circuits.

Schematic and waveforms of adder without using of logical effort:-



Conventional adder with logical effort: - Because logical effort technique could be applying on the circuits which contain path formed by gates. So in adder also I have selected two paths on which I have applied technique which is highlighted below:

Fig.6:- Full adder with highlighted paths on which logical effort applied.

Schematic and waveforms of adder with using of logical effort:-

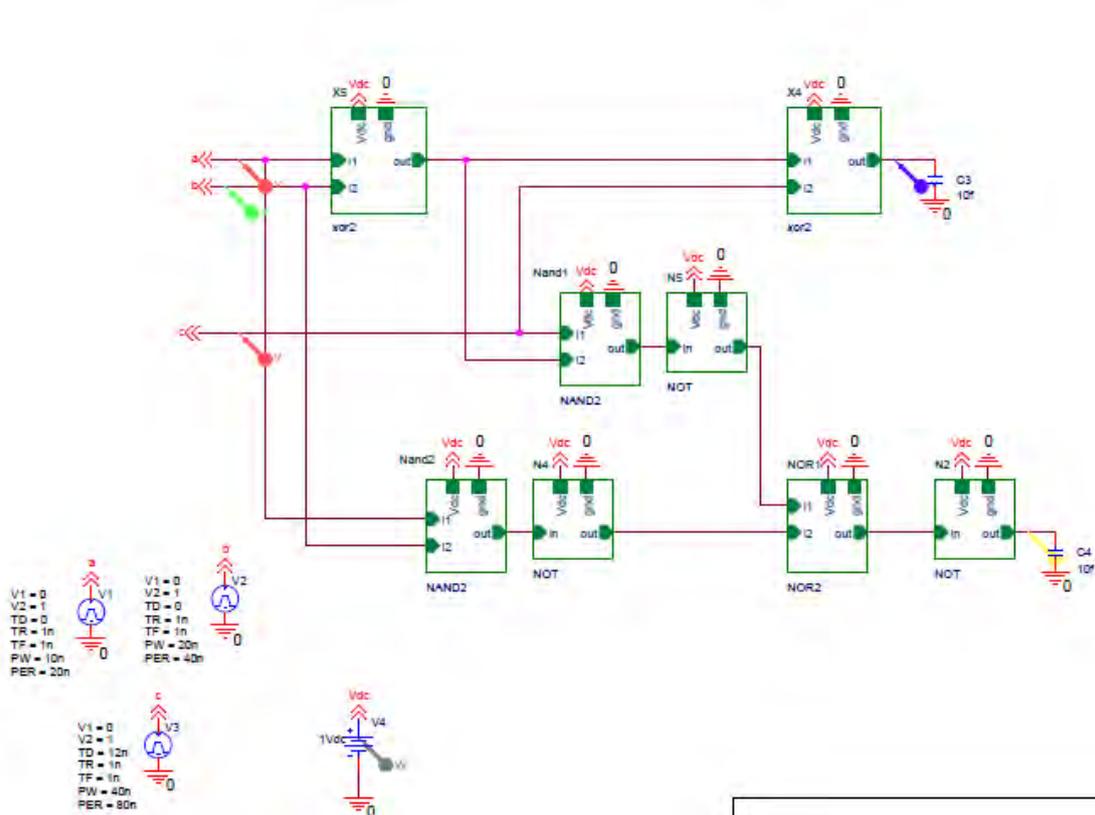
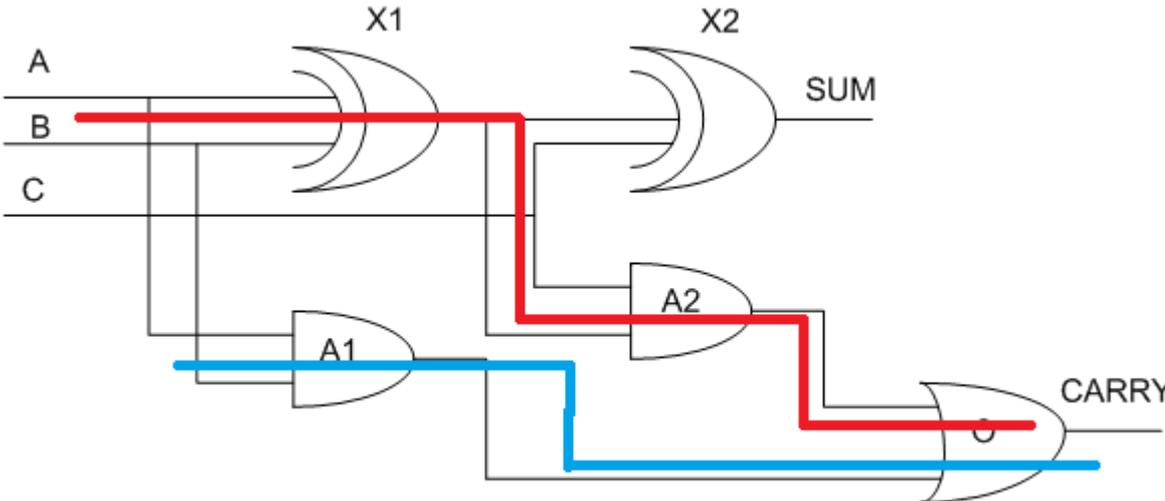


Fig.7:- Schematic view of adder at 10f

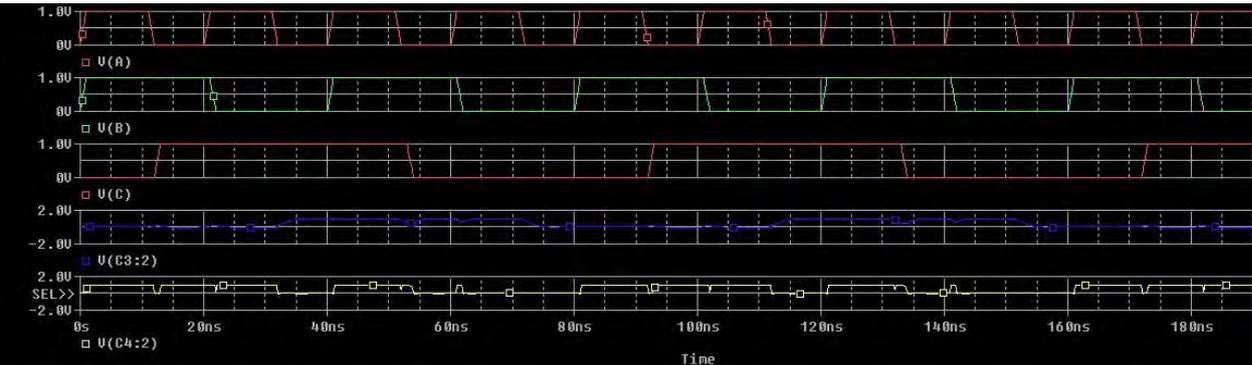


Fig.8:- Waveform of the full adder at 10f

LOAD(fF)	NORMAL			LOGICAL			Delay Decrease%	PDP Saving %
	Pow(uw)	Del(ns)	pdp	Pow	Del(ns)	pdp		
10	1.54	.343	.53	1	.302	.30	13.57	43.39
50	4	2	8	1.5	.44	.66	78	91.75
100	5	2	10	5.6	.75	4.2	62.5	58

From the comparison we can see that delay decrease nearly 52% and PDP saving nearly 65%. So we can say the performance of adder has been improved by using logical effort technique. Now using this adder as a building block I can apply this in 3bit and 5bit multiplier which is well known and acceptable in many digital logic circuits

IV. CONCLUSION:-

Logical effort technique has been used for power estimation and by the scaling of transistors of various gates I have decreases in propagation delay and also saved in Power Delay Product.

It is best transistor sizes for the logic gates and has to be proven an easy way to estimate power and delay. Logical effort technique has been proven to be good agreement for delay and power delay product improvement of adder and array multiplier circuits.

REFERENCES: -

- [1] Logical effort based dynamic power estimation and optimization of static CMOS circuits Original Research Article Integration, the VLSI Journal, Volume 43, Issue 3, June 2010, Pages 279-288, A. Kabbani
- [2] K. Lahiri, A. Raghunathan, S. Dey, D. Panigrahi, Battery-driven system design: a new frontier in low power design, Proceedings of the IEEE 15th International Conference on VLSI Design, 2002, pp. 261–267.
- [3] I. Sutherland, B. Sproull, D. Harris, in: Logical Effort: Design Fast CMOS Circuits, Morgan Kaufmann publishers, 1999 January.
- [4] P. Rezvani, A.H. Ajami, M. Pedram, H. Savoj, LEOPARD: A logical effort-based fan-out optimization for area and delay, Proceedings of IEEE International Conference on CAD, 1999, pp. 516–518.
- [5] S. Turgis, N. Azemard, D. Auvergne, Design and selection of buffers for minimum power-delay product, Proceedings of the European Design and Test Conference, March, 1996, pp. 224–228.
- [6] H.Q. Dao, B.R. Zeydel, V.G. Oklobdzija, Energy minimization method for optimal energy-delay extraction, Proceedings of IEEE Solid-State Circuits Conference, 2003, pp. 177–180.
- [7] Logical effort model extension with temperature and voltage variations Chun-Hui Wu; Shun-Hua, Lin; Herming, Chiu Thermal Investigation of ICs and Systems, 2008. THERMINIC2008. Publication Year: 2008, Page(s): 85 – 88
- [8] D. Markovic, V. Stojanovic, B. Nikolic, M.A. Horowitz, R.W. Brodersen, Methods for true energy-performance optimization, IEEE Journal of Solid-State Circuits 39 (8) (2004) 1282–1293 Aug.
- [9] K.S. Lowe, P.G. Gulak, A joint gate sizing and buffer insertion method for optimizing delay and power in CMOS and BiCMOS combinational logic, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 17 (5) (1998) May.
- [10] J. Cai, Y. Taur, S.F. Huang, D.J. Frank, S. Kosonocky, R.H. Dennard, Supply voltage strategies for minimizing the power of CMOS in: processors, IEEE Symposium on VLSI Technology, Digest of Technical Papers, 2002 pp. 102–103.
- [11] P. Pant, V. De, A. Chatterjee, Device-circuit optimization for minimal energy and power consumption in CMOS random logic networks, Proceedings of the 34th Design Automation Conference, 1997, pp. 403–408.
- [12] A. Shebaita, Y. Ismail, Multiple threshold voltage design scheme for CMOS tapered buffers, IEEE Transactions on Circuits and Systems II 55 (1) (2008) 21–25 Jan.