Hazards and Glitch Power Reduction of CMOS Full Adder in 90nm Technology

B. J. Singh
Department of ECE, CVR College of Engineering, Hyderabad, India
Jailsingh4@gmail.com

Abstract — **In This paper we discuss the glitch and hazard power compensation techniques which involves reducing the undesired switching of combinational circuits in order to save the dynamic power for CMOS stander cell designs in 90nm. In nanometer CMOS technologies the power consumption is become a serious concern. The total power consumption is mainly due to the dynamic and leakage power consumptions. In CMOS circuits a glitch occurs when differential delay at the inputs of a gate is greater than inertial delay, which gives an amount of power consumption. In lower technology nodes this glitch power is a major prominent. Experimental results gives 12% to 50% reduction in top 10 peak undesired transition. The proposed methodology has been validated using cadence 90nm gpdk technology libraries.**

Index Terms—**CMOS, Full Adder, Glitch, BTBT, Path Filtering, Sizing.**

I. INTRODUCTION

In a digital CMOS circuits there are three sources of power dissipation. The first is due to signal transitions. As the "nodes" in a digital CMOS circuit transition back and forth between the two logic levels, the capacitance associated with the nodes get charged and discharged. Heat dissipation in the channel of the transistors due to the current flows and electrical energy through the channel of the transistors gets converted into the heat. This portion of heat dissipation is proportional to the supply voltage. Given that the voltage swing in most cases is simply equal to the supply voltage, the power dissipation due to transitions varies over all as the square of the supply voltage.

The second source of power dissipation comes from the short-circuit currents, When the P-network and N-network of a CMOS gate conduct simultaneously then the short-circuit current flows directly from the supply to the ground. However, when the input of a gate changes the output response of the gate is also switches, so both subnetwork conduct simultaneously for a small interval. The duration of interval depends on transition (rise or fall) times of the input and the output and so does the short-circuit dissipation. The power dissipation in CMOS circuits are related to transition (rise or fall) times at gate outputs and are therefore collectively referred to as dynamic power dissipation.

The third and the last source of dissipation is due to the leakage currents, which flow when the inputs to and, therefore, the outputs of a gate are not changing and is called static power dissipation. One of the reasons CMOS circuits are in widespread use is that the only static dissipation in standard CMOS circuits is due to leakage currents and magnitude is very small usually. But to reduce dynamic power as long as the supply voltage is being scaled down, to maintain performance low threshold voltage transistors to be used, yet the lower the threshold voltage, the greater the standby leakage current. The reasons for such behavior of transistors will be explained in the in the following.

II.BACKGROUND

At the macro-chip level, the most significant trend is the increasing leakage power contribution in the total power dissipation of an IC design in CMOS technology. For a long time, the dynamic power was the major component of the total power dissipated due to the switching component by a circuit. However, in order to maintain power dissipation and power delivery costs under control, the supply voltage 'Vdd' was scaled down at the rate of 30% per each technology generation. In conjunction, to improve the performance of transistor and circuit the threshold voltage Vt was also reduced at the same rate, so that a large gate overdrive (V_{dd} -) is maintained. However, transistor sub-threshold leakage current (I_{sub}) is increase exponentially because of reduction in Vt. Furthermore, other components of leakage current, e.g., reverse-biased junction, Band To Band Tunneling (BTBT) and the gate leakage become important as we scale fabrication technology to 90nm and further. In other hands such as gate-induced drain leakage(GIDL) and drain-induced barrier lowering (DIBL) will also become increasingly significant.

Another dimension of drawback is added by the fact that unlike dynamic power, leakage power increases exponentially with temperature. We have been continuously scaling the supply and threshold voltages In order to improve performance. While this results in high frequency of operation, temperatures rise due to large active power consumption. Further increasing temperature there is a strong function of temperature. This circular situation is depicted in Fig.1. A positive feedback between leakage power and temperature can result in thermal runaway, if heat cannot be dissipated effectively. Such a situation can have disastrous consequences, including permanent physical damage of the circuit, the processor if the temperature increases beyond safe limits most processors are now equipped with thermal sensors and hardware circuitry that will stop.

 DOI:10.32377/cvrjst0707

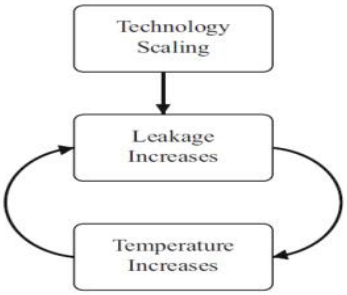


Figure.1. Leakage increases exponentially with the temperature.

A. *DYNAMIC POWER*

The dynamic power dissipation in a CMOS gate is due to the charging and discharging of load capacitance driven by the gate. This capacitance consists of internal capacitances of the gate, wire capacitance of the fanout net and the capacitance of the gate terminals of the transistors being controlled by the fanout net. This power dissipation can be calculated by the following equation(1).

$$P_{dyn} = \frac{1}{2} C_{load} V_d^2 \text{ ……..(1)}$$

Where

- P_{dyn}: dynamic power dissipation of gate,
- C_{load}: load capacitance of the gate,
- f:clock frequency & V_{dd}: supply voltage,
- D: transition density of the output of the gate.

The transition density is the average number of transitions during a clock cycle. The dynamic power dissipated is thus proportional to the number of transitions occurring at agate. Thus, an accurate estimation of the transition density in a circuit will give an accurate estimation of the dynamic power dissipated in the circuit. In earlier technologies, dynamic power accounted for most of the power used by CMOS circuits. But with the advent of deep sub-micron technology of the other components of consumption are also becoming significant. However, several low power techniques have concentrated on minimizing dynamic power, as will be explained later. Dynamic power can be classified into necessary transition due to unbalanced paths in the circuit. The latter component of dynamic power dissipation is the glitch power and is elaborated in the next section.

B. *HAZARDS AND GLITCH POWER*

In a digital circuit signals before to reach steady state, gates can have multiple transitions. Since the power dissipation is directly proportional to the number of transitions, these necessary transitions increase power consumption. These undesired transitions power dissipation are called glitch power or hazards. Glitches are produced in a digital CMOS circuit due to the difference in signal arrival times at the inputs of the gates. Power dissipated by glitches is called glitch power and it typically about 20% of the overall power consumption of the chip and even 70% in some typical cases as the combinational adder[4].

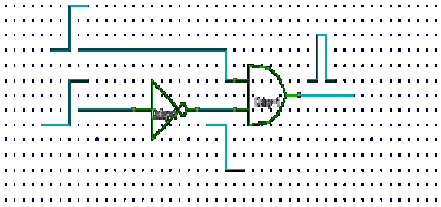


Figure.2. An example of static hazard

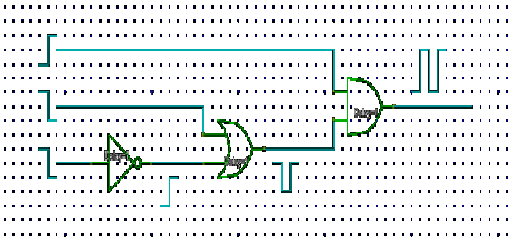


Figure.3. An example of a dynamic hazard

Consider the example of fig.2. with each gate having one unit of delay due to the difference in arrival time of signals at the inputs of the AND gate the logic value glitches or emits a pulse of 1 unit width, which equals the inverter delay. This known as static hazard, In fig.3. the OR gate produces a static hazard of 1 unit. The transient consists of three edges, two rising and 1 falling. This is a dynamic hazard with a combined width of 2 units.

The number of arriving signals at the gate is may equal to the number of edges in transients at the output of a gate. The maximum width of the possible glitch at the circuit output and also the maximum difference in arrival time of signals at the inputs of gate is called differential path delay. A hazard producing gate has more than one input and has non-zero differential path delay. Every gate has a internal delay due to the finite switching speed of the transistors. It is the time a device takes to switch the output after the cause for the change has occurred at the input[5]. Inertial delay plays a major role in distorting glitches produced at the gates and in the next section we present ways of eliminating glitch by either making the differential path delay zero or by increasing the inertial delay of the gate.

III.EXISTING TECHNIQUES

A. *PATH BALANCING*

Path balancing is one of the earliest works that targeted glitch power reduction. The glitches are proposed due to difference in arrival time of signals at gat inputs. The idea behind this technique is to prevent glitches from occurring by balancing the delays of paths such that at any given gate the signals arrive at its input terminal at the same time. Consider the example circuit shown in fig.4, This signals at the inputs of the gate arrive at different times which would probably results in glitches. This behavior can be suppressed by resulting the circuit as shown in fig.5. However, as next example illustrates, a simple resulting of gates may not be sufficient for some circuits.

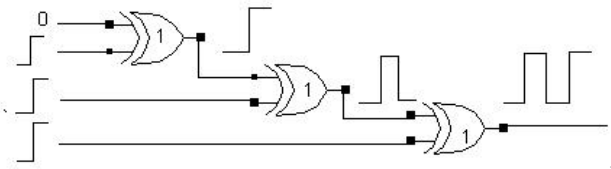


Figure.4. Glitching behavior of an ill-structured circuit

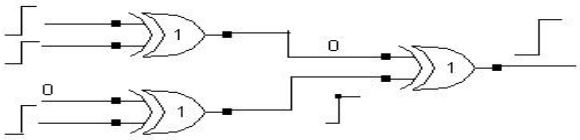


Figure.5. Removal of glitches after restructuring of the circuit in fig.4.

A method of balancing delay for circuit by the inserting buffers at selected inputs of the gates. The addition of buffers is done such that it will not increase the critical delay, but will effectively eliminate spurious transitions. The buffers are inserted only in the fast paths of the circuit and since the slowest paths determine the speed of the chip they are left untouched. The circuit in fig.6 cannot be path balanced without the introduction of buffers as shown in fig.7. However the addition of buffers increases the switching activity of the circuit which may offset the reduction in power due to elimination of glitches.

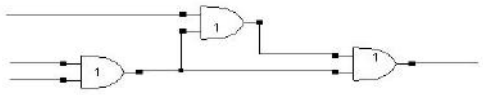


Figure.6. Glitching behavior of a circuit where simple restructuring will not balance paths.

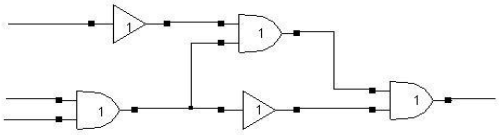


Figure.7. Path balancing of a circuit in fig.6. by buffers.

In CMOS circuits the fanout load significantly affects the gate delay. To account for this developed a delay buffer insertion procedure using a chain of buffers, thus eliminating the need for merging the common delay of all the faout branch buffers into the corresponding fanout stem buffer. The problem of delay buffer insertion on a multiple fanout net simplified by assuming that only one size of delay buffer is used. This method permits a quick estimation of the number of buffers needed to balance the circuit and can therefore be usedin heuristics for logic restructuring.

B. HAZARD FILTERING

This method eliminates glitches in the circuit using inertial gate delays. Inertial delay is the interval that elapses after an input change before a gate can produce an output change. When the time between successive edges in the input signal is less than the inertial delay of the gate, the effects of these edges are suppressed. This is known as filtering effect of the gates [5]. CMOS gates have a built-in delay called the inertial delay of the gate and they suppress signals that are of smaller width than the inertial delay from passing through the gate. This is known as the filtering effect of the gate. Instead of balancing the delays of the gate inputs to be exactly equal, this method evaluates the differential delay of inputs and increases the inertial delay of the gate exceed the differential delay such that the glitch will be suppressed within the gate.

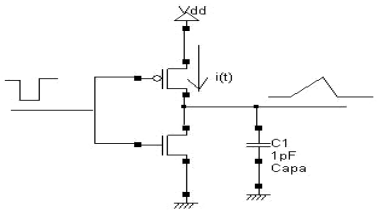


Figure.8. Energy dissipation in an inverter

Consider the example inverter shown in Figure.8 with the total capacitance at the output node as C and the short circuit impedance R. when an input pulse as shown is incident at the input, a charging current i(t) flows through the upper transistor charging the capacitor C. This charging raises the output voltage until time t=τ when the rising edge occurs at the input. The energy consumed by the gate is

$$E(\tau)=CV_{dd}^2\left(1-e^{-\frac{\tau}{RC}}\right) \quad \ldots\ldots\ldots(2)$$

For hazard filtering, we increase the time constant RC of the gate and thus increase the charging time needed for the capacitor reach Vdd. When the time constant is large, there is practically no energy dissipation in the gate due to the short glitch of length τ. Energy dissipation for various values of the time constant with respect to the width of the pulse τ is given in above equation(2).

A simulated example of a full adder circuit showed that the method reduced power by 42%. The glitch- free circuits had gates whose speed was decreased to 20% of their original speed but with very little reduction in overall speed of the circuit. This was because those gates were mainly on non-critical paths and so did not contribute much to the critical path delay of the circuit.

C. GATE SIZING

Gate sizing is defined as assigning load drive capabilities to the gates of a network, such that a given delay limit is obeyed, and the total minimal cost in terms of power consumption[5]. Similar to the hazard filtering technique, this method also uses the filtering effect of the gates but achieves the increase in inertial delay by resizing the gates in the circuits. The delay model frequently used for solving this problem is as shown below [5]:

$$\tau_{gate} = \tau_{int} + c\,C_{load} \quad \ldots\ldots(3)$$

$$C_{load} = C_{wire} + C_{in} \quad \ldots\ldots\ldots(4)$$

where

- τ_{gate} : Delay of the gate
- τ_{in} : Internal delay of the gate
- C_{load}: Capacitive loading experienced by the gate

 DOI:10.32377/cvrjst0707

- C: A constant
- C_{wire}: Routing capacitance of the output net of the gate
- C_{in}:capacitive loading due to gates in the fanout cone

Berkelaar and Jacobs [9] use a parameter called the speed constant to reformulate the delay model but this converts the problem into non-linear domain. Berkelaar et al. [8] have tried to solve the near-LP problem with the use of a piecewise linear simulator ad have published results for all ISCAS benchmark circuits. This method was found to be faster than the LP method for circuits with less than 1000 gates but was not very useful for larger circuits. Berkelaar and Jacobs [9] have tried to formulate the gate sizing problem as an LP problem but the non-linear structure of their delay model posed problems in finding the global optimum solution. The details of this are dealt with in the next sections.

Berkelaar and Jacobs [9] have tried gate sizing under a statistical delay model but the resulting problem was a non-linear problem that caused their program to consume considerable time and resources. Also, this method could not be used for circuits with more than 1000 gates.

D. TRANSISTOR SIZING

Transistor sizing similar to gate sizing but the essential difference is that in gate sizing all the transistors of a gate are sized together but in transistor sizing each transistor can be sized independently. Traditionally, transistor sizing is done to reduce the area and delay of a VLSI chip.

Data et al. [10] have considered static CMOS circuits transistor sizing for low-power and high performance. To obtain a better power and delay performance the transistors on the critical paths of the circuit are scaled size. In the circuit block to improve the switching speed and the output transition characteristics of a particular circuit block on the critical path one may seek to increase the widths of the transistors. This results in an increased current drive and better output transition time. A transition time implies lower rush-through current for faster input/output, hence smaller short –circuit power dissipation. It is to be noted, however, that even through the delay of a particular block and its succeeding block are reduced, an increase in transistor widths increases the capacitive loading of the preceding block and may severely affect its power and delay. Thus, the issues regarding delay and power dissipation are fairly inerter linked. The algorithm described [10] minimizes the delay. The area and the power dissipation of a circuit by optimizing the sizes of the gates on the critical paths of the circuit However. The constrant set for this model becomes non-linear and hence the solution for large circuits becomes tedious and complex.

E. LINEAR PROGRAMMING (LP) APPROACH

A linear program determines a set of variables such that an objective is minimized under given constraints.To eliminate the glitching power from a circuit the inertial delay of the gate has to be alerted as dictated by a hazard filtering technique. But the altering has to be done without affecting the critical path of the circuit and also taking into account the change in delay of the gates in the fanout cone whose delay will effect the delay of the gate in question. There can be infinite number of solutions and finding a global optimum makes this problem an optimtization problem. The linear programming model guarantees the global optimum for every feasible solution of the problem.

A variety of LP models have been investigated to express the glitch removal problem. This have described a gate level model to express the problem. The variables consist of inertial delays of gates in the circuit and also delays of buffers that may need to be inserted in the circuit for complete glitch removal. The constraints are written by path enumeration from the gate inputs to the primary inputs (PIs). Hence, the number of constraints grows exponentially with the size of the circuit. In the mathematical model proposed by Berkelaar et al.[9] the gate delays were described at transistor level. This introduced some no-linearity in the model which increased the complexity and decreased the accuracy of the solution. Another linear programming technique has been proposed by Raja et al.[4]. This is a gate level mathematical model whose constraint set grows linearly with the size of the circuit. Hence, this technique has been proven to be applicable to large circuits also. This detail of this linear program is given below.

IV. CMOS FULL ADDER CIRCUIT

To process the fundamental arithmetic operations addition is an obligatory operation. It is widely used extensively in many VLSI design paradigms and is by far the most frequently used operation in a general purpose system and in application specific processors. Addition is often seen as an indispensable part of the arithmetic unit. It is dubbed the heart of any micro-processor, DSP architecture and data processing system.

In arithmetic operation it usually involves a carry ripple step which must propagate a carry signal from each bit to its higher bit position, so addition is a very crucial operation. This results in a substantial circuit delay. Therefore, which lies in the critical delay path, effectively determines the system's overall speed. On the other hand, the operation of reducing the power consumption of the designed adder, which for many years has been a narrow specialty, has recently been gaining prominence. This can be attributed to the emergence and increasing popularity of smaller and more durable mobile computing and communication systems. Low power dissipation allows a portable system to operate longer with the same battery.

A full adder adds two binary numbers with a carry-in. The structure representation of the conventional CMOS full adder appears in fig.9. It is constructed of the two HA's and an OR gate. There are a total of three inputs for the FA, two for the input numbers A and B, and one for the carry-in, Cin. The outputs are the sum and carry-out. According to the truth table of FA, the logic function corresponding to terminals SUM and CARRY OUT are as follows

SUM=A xor B xor Cin ……..…...(5)

Cout = (A xor B) . Cin + A . B ……(6)

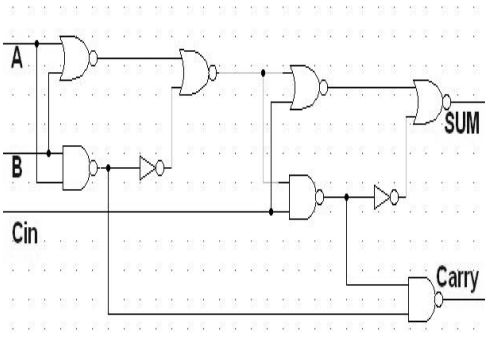


Figure.9. Logic circuit of the conventional full adder.

The SUM (S) output has been generated using two XOR gates. The XOR function is obtained using a pass transistor-based XNOR gate followed by an inverter to restore the logic levels. For generating the carry output, transmission gates have been used instead of single pass transistors to pass inputs A or C_{in}. This resulted in improved logic levels at the output of the transmission gate leading to fully restored logic level at C_{out} and fig.10 shows the circuit diagram of proposed full adder. In practical digital circuits, a functional block is driven through a gate / inverter and is required to drive a gate. The design of a digital circuit, therefore, should be tested with inputs applied through an inverter and another inverter / gate connected as load. The circuit in fig.10 was tested with inputs applied through inverters and similar inverters connected as load. Both the input and load inverters are included in the circuit diagram. The optimum performance of the circuit in terms of speed, fully restored logic levels was obtained by choosing appropriate geometry of individual MOSFET. The pMOS and nMOS transistors used in the inverters have (W/L) ratio of (5/2) and (3/2), respectively. Inverters used in the full adder are also of same type. All transistors in XNOR gates have (W/L) of (6/2). The pMOS and nMOS transistors used in transmission gates have (W/L) of (6/2) and (3/2), respectively.

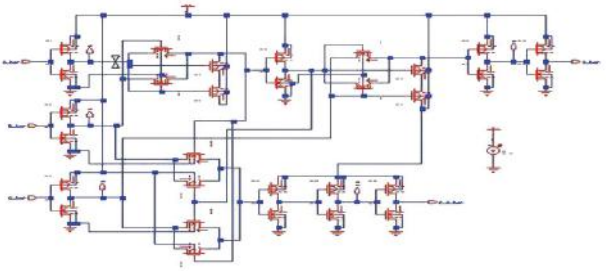
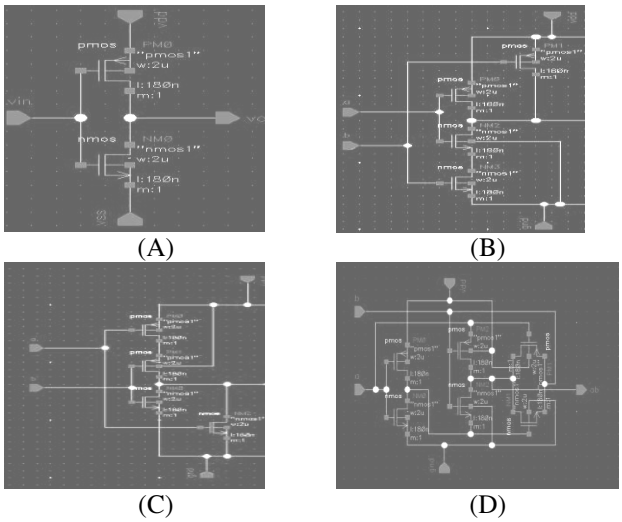


Figure.10. 20-transistor Full Adder circuit

V. SIMULATION RESULTS

Here the present paper is aim to be to reduce power and delay parameters. Cascading of two 1 bit half adder circuits implements a full adder. In order to build transistor level adder, the idea is to begin with the basic gates. So as to reduce the circuit area, power dissipation and propagation delay each individual logic gate used in the circuit is constructed by minimum transistors. The performance criteria of each gate is individually investigated and analyzed for the full adder as shown in Fig.9.

The implementation consists of four types of logic gates such as Inverter, 2-input NAND gate, 2-input NOR gate and may be 2-input XOR gate is also used. The schematic of all the Logic gates implemented using CMOS 90nm technology is shown in Fig.11 below.



(A) (B)

(C) (D)

Figure.11. Schematic of (A) Inverter (B) 2-Input NAND gate (C) 2-Input NOR gate (D) 2-Input XOR gate.

Here both proposed and conventional full adder to show the reduced transistor count & both full adders are implemented in the cadence tool as shown in fig.12 and fig.13.

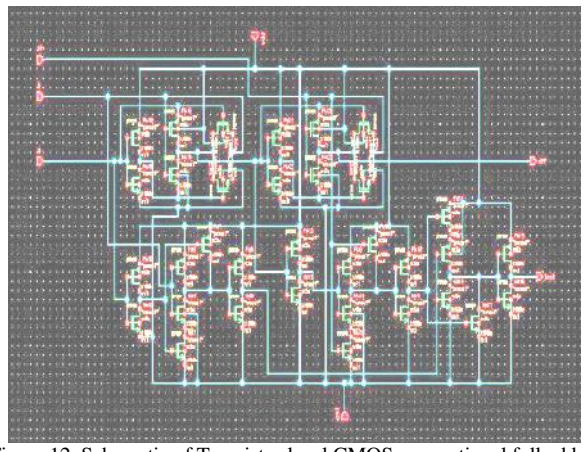


Figure.12. Schematic of Transistor level CMOS conventional full adder.

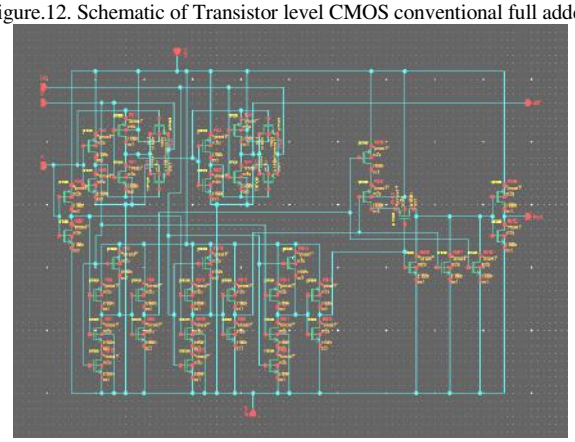


Fig.13. Schematic of Transistor level CMOS Full Adder with minimum glitches

TABLE.I:
PERFORMANCE OF LOGIC GATES IN 90NM

 DOI:10.32377/cvrjst0707

LOGIC GATE	Propagation Delay	Power Dissipation
NOT	7ps	18µW
NAND	14ps	25 µW
NOR	13ps	23 µW
XOR	20.4ps	28 µW

TABLE.II
PERFORMANCE OF FULL ADDER IN 90NM

PARAMETER	CONVENTIONAL FULL ADDER	GLITCH FREE FULL ADDER
Propagation Delay: A to S	110ps	250ps
Propagation Delay: B to S	100ps	140ps
Propagation Delay: C to S	80ps	100ps
Propagation Delay: A to C_0	120ps	220ps
Propagation Delay: B to C0	100ps	120ps
Propagation Delay: C to C0	60ps	120ps
Power Consumption at 500MHz (µW)	94.5	50.7

It can be seen that the output logic levels are fully restored and are free of glitches. The each path propagation delay, and power dissipation as determined are given in Table.II. The conventional full adder path delay is more than compared to the glitch free full adder because to remove the glitches in each path by balance path delay to occur inputs of all gates/transistor at same time. The power consumption was determined by multiplying the power supply voltage with average current drawn from the power supply, the power consumption mentioned in the Table.I and Table.II. The response characteristics such as propagation delay and power dissipation were computed using cadence tool using 90nm technology libraries.

CONCLUSIONS

Unfortunately, the power density increased prominently as a consequence, glitches or hazards are expected to result in further increases in power density for technology node below 90nm. This paper summarized some of the contributions made to reduce glitch power in CMOS digital circuits. A new 20-transistor full adder circuit has been implemented in 90nm technology. In this work different skills and techniques were gained, applied in a fundamental part. Indeed. This full adder is able to design, implemented and successfully analyzed the characteristics. The completion of this task was satisfactory since the theoretical expectations matched with experimental results. This full adder offers better performance than the other full adder circuits (implemented in same technology) reported in the literature. This design is particularly suited for use in a binary multiplier circuit where a number of full adder circuits need to be cascaded. Any logic level degradation in such application would adversely affect the speed of the multiplier. The performance of the full adder was assessed in terms of area, speed and power consumption, also quality of the output signals by comparing the timing measurements of each basic gate.

REFERENCES

[1]. Sung-Mo Kang, Yusuf Leblebici, (2003) CMOS Digital Integrated Circuits: Analysis and Design TATA McGRAW-HILL.
[2]. Microelectronic Circuits by Adel S. Sedra and K. C. Smith. Oxford University Press; Fourth Edition.
[3]. M. S. Elrabaa, I. S. Abu-Khater, M. I. Elmasry, "Advanced low power digital circuit techniques, Kluwar academic publisher 1997.
[4]. T. Raja V. D. Agrawal, and M. L. Bushnell, CMOS circuit design for minimum dynamic power and highest speed, in proc. Of international conference on VLSI design, Jan 2004, pp. 1035-1040.
[5]. V. D. Agrawal, Low power design by Hazard filtering, in proc. of the international conference on VLSI design, Jan 1997, pp. 193-197.
[6]. A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, An average power dissipation and random pattern testability of CMOS combinational circuit, in proc. of international conference on computer aided design, Nov 2001, pp. 402-407.
[7]. K. Roy S. C. Prasad, Low Power CMOS VLSI Circuit Design, New York: Wily Interscience Publications, 2000.
[8]. M. Berkelaar, P. Buurman, and J. Jess, Computing entire area/power consumption versus delay trade-off curve for gate sizing using a piecewise linear simulator, IEEE Transaction on circuits and systems vol. 15 , no. 11, pp. 1424-1434, Nov 1996.
[9]. M. Berkelaar, E. Jacobs, Using gate sizing to reduce the glitch power, in proc. of the pro RISC workshop on circuits, Systems and Signal processing, Nov 1996, 183-188.
[10]. S. Datta, S. Nag, K. Roy ASAP: A transistor sizing tool for area, delay and power optimization of CMOS circuits, in proc. of the international symposium on circuits and systems, May 1999, pp. 61-64.
[11]. C. Y. Chen, and Y. P. Chou, Novel Low Power 1-bit Full Adder Design, IEEE conf. ISCIT, pp. 1348, 2009.
[12]. C. H. Chang, J. M. Gu, and M. Zhang, A Review of 0.18 µm Full Adder Performances for Tree Structured Arithmetic Circuits, IEEE Trans. On VLSI Systems, Vol. 13, no. 6, pp. 686, 2005.
[13]. Cadence –Analog and Mixed Signal Labs Manual. http://cadence.com//