Indian Institute of Management Calcutta

Working Paper Series

WPS No. 761
May 2015

A Rule-based Method for Minimizing Power Dissipation by Reducing Switching Activity of Digital Circuits

Subrata Das
Academy of Technology Hooghly (Aedconagar), West Bengal, 712121, India

Sudip Ghosh
School of VLSI Technology Indian Institute of Engineering Science & Technology
(IIEST) Howrah, West Bengal, 711103, India,

Parthasarathi Dasgupta
Professor, IIM Calcutta, Joka, Diamond Harbour Road, Kolkata 700104, India
http://facultylive.iimcal.ac.in/workingpapers

Samar Sensarma
University of Calcutta, Kolkata 92, A.P.C. Road, Kolkata - 700 009, West Bengal, India

A Rule-based Method for Minimizing Power Dissipation by Reducing Switching Activity of Digital Circuits

Subrata Das

Dept. of Information Technology

Academy of Technology

Hooghly (Aedconagar), West Bengal, 712121, India

dsubrata.mt@gmail.com

Sudip Ghosh

School of VLSI Technology

Indian Institute of Engineering Science & Technology (IIEST)

Howrah, West Bengal, 711103, India,

sudip.ghosh@vlsi.iiests.ac.in

Parthasarathi Dasgupta

Management Information System Group

Indian Institute of Management Calcutta, Kolkata

Diamond Harbour Rd, Joka, Kolkata, West Bengal 700104, India

partha@iimcal.ac.in

Samar Sensarma

Department of Computer Science & Engineering

University of Calcutta, Kolkata

92, A.P.C. Road, Kolkata - 700 009, West Bengal, India

sssarma2010@gmail.com

A Rule-based Method for Minimizing Power Dissipation by Reducing Switching Activity of Digital Circuits

Subrata Das, Sudip Ghosh, Parthasarathi Dasgupta and Samar Sensarma

Abstract *— Minimization of power dissipation of VLSI circuits is one of the major concerns of recent digital circuit design primarily due to the ever decreasing feature sizes of circuits, higher clock frequencies and larger die sizes. The primary contributors to power dissipation in digital circuits include leakage power; short-circuit power, and switching power. Of these, power dissipation due to circuit switching activity constitutes the major component. As such, an effective mechanism to minimize power loss in such cases often involves the minimization of switching activity. In this paper, we propose an intelligent rule-based algorithm for reducing the switching activity of digital circuits at logic optimization stage. The proposed algorithm is empirically tested for several standard digital circuits with Synopsis EDA tool and results obtained are quite encouraging.*

Keywords **—** Switching Activity, low-power VLSI circuits, CMOS, Power dissipation, dynamic power, electromigration.

1 INTRODUCTION

Traditionally, the major concerns of the VLSI designers include minimization of chip area, enhancement of performance, testability, reduction of manufacturing cost and improvement of reliability. With increasing use of portable devices and wireless communication systems, reduction of energy consumption and hence reduction of power dissipation and optimization of chip temperature have become some recent major concerns in VLSI design [17]. Power dissipated by a digital system increases the temperature of the chip and affects battery life of the digital devices [14]. Aggressive device scaling also causes excessive increase in power per unit area of the chip. This high power dissipation in VLSI devices is usually manifested in the form of rise of chip temperature. As such, heat generation and its removal from a chip are of serious concern [2]. The heat removal system must be efficient and must keep the junctions below a certain threshold, as determined by reliability constraints. With higher level of integration more and more transistors are being packed into smaller areas. Thus, for high level of integration, heat removal is a dominant design factor. In absence of appropriate removal of the generated heat, the chip temperature may rise causing thermal breakdown. One of the major reasons of VLSI chip failure is due to interconnect failure, often attributed to the phenomenon of electromigration, causing leakage power. The mean time to failure (MTTF) of interconnect due to electromigration is given by Black's equation [7]

$$MTTF = Aj^{-2}e^{(\frac{Q}{KT})} \qquad (1)$$

where A is a constant based on the interconnect geometry and material, j is the current density, Q is the activation energy, K is Boltzman constant, T is the temperature. From equation 1 it is obvious that MT TF decreases exponentially with increase in temperature.

The temperature of a particular region of chip depends on the power density of that particular region and its adjacent regions. Non-uniform distribution of temperature across the chip often creates hot-spots. Hence chips must be designed to avoid hot spots, which necessitates in having almost uniform temperature over the entire chip surface.

In CMOS circuits following are the three primary sources of power dissipation [1].

1. The switching activity occurring due to the logic transitions. When the nodes of a digital circuit make transition back and forth between two logic levels, parasitic capacitances are alternately charged and discharged. Consequently current flowing through the channel

resistance of the transistors consumes electrical energy that is converted into heat [1]. Power loss due to switching activity is given by the following equation

$$P = 0.5 C_L V_{DD}^2 E(sw) f_{clk} \qquad (2)$$

where C_L is the physical capacitance at the output of the node, V_{DD} is the supply voltage, *E(sw)*, the switching activity, is the average number of output transitions per $1/f_{clk}$ time and f_{clk} is the clock frequency. The product of *E(sw)* and f_{clk} is the number of transition per second [22].

2. The short-circuit current that flows from supply to ground when both the p-sub-network and n-sub-network of a CMOS Gate conduct [1].

3. The leakage current [1] caused by substrate injection at p-n junctions and sub-threshold effects determined by the fabrication technology.

The first two sources of power dissipation are known as dynamic power dissipation and the third one constitute the static power dissipation. In the present-day technology about 80% of the total power loss occurs due to switching activity [1]. Thus in order to reduce the power dissipation of VLSI circuits it is desirable to minimize the switching activity of the circuits.

In the paper, we propose an algorithm to obtain for a given logical input expression, an equivalent logical expression with optimal switching activity. Rest of the paper is organized as follows. Section 2 reviews some related recent works and Section 3 introduces the preliminary concepts and computations related to switching activity. Section 4 discusses the motivation for the proposed method. Section 5 provides the formulation of the problem. Section 6 discusses the forming of the rules to be applied in the proposed algorithm for reducing switching activity, and Section 7 describes the proposed rule-based method. Section 8 discusses the experimental results and Section 9 concludes the paper and highlights some of the future scopes of work.

2 LITERATURE REVIEW

Minimization of power consumption of CMOS digital circuits is studied in the past considering all levels of the design such as physical, circuit and logic level [11] [4]. A review on different methodologies for low power VLSI design is also reported in [25]. In digital CMOS circuits the measure of power dissipation is the circuit activity or average number of transitions. Minimization of average number of transition of CMOS digital circuit nodes is discussed in [9]. Estimation of average switching activity in combinational and sequential circuits under random input sequences is presented in [8] using the general delay model. An analytical approach to compute the switching activity of digital circuits at word-level in the presence of glitches and correlation is presented in [16]. The work in [5] provides an interesting repository of recent techniques of power modeling and low power design based on high-level synthesis. The evaluation and reduction of switching activity in combinational logic circuits considering both the transitions 1 → 0 and 0 → 1 at any output node is proposed in [15]. In order to satisfy the classical probabilistic approach that limits the maximum value of switching activity to 1 the definition of switching activity as proposed in [15] was customized in [19]. An algorithmic approach at the Gate level using k-map for reducing the switching activity in combinational logic circuits is presented in [20] and about 10% reduction in switching activity was observed by the proposed method. A unified method to compute the switching activity at the Gate level is also reported in [12]. In [13] the authors proposed a method to estimate the switching activity using a variable delay model. The work of [26] has discussed the system level dynamic power management in chip nanoscale CMOS multiprocessors. An optimal polynomial time algorithm for power minimization of popular media applications, such as audio, video, and sensor network data under QoS requirements and hardware constraints using multiple voltages is described in [21]. Reduction of power dissipation by splitting both NMOS and PMOS transistor of CMOS circuits into two transistors is described in [24]. Such splitting will reduce the power dissipation of digital circuits compared to general static CMOS logic. The work of [28] proposed algorithm to minimize the logic synthesis with reduce area and number of interconnects that will reduce the power consumption and delay fault. Pre-computation-based optimization for low power that computes the output logic values of the circuit one clock cycle before they are computed was discussed in [10].

3 PRELIMINARIES

A definition of switching activity based on classical probabilistic approach is given in [19]. For a logical expression of a switching function for a node *i*, let the sets of minterms and maxterms are represented by N_i and X_i respectively. It is easy to see that a minterm intrinsically represents the number of 1's in the output node, and a maxterm intrinsically represents the number of 0's in the output node. If $| N_i |$ and $| X_i |$ represent the cardinalities of the sets *Ni* and *Xi* respectively, the probabilities of occurrence of a 0 and a 1 respectively at the output node *i* are given by the following equations:

$$P_0 = \frac{| X_i |}{| N_i | + | X_i |} \qquad (3)$$

$$P_1 = \frac{|N_i|}{|N_i| + |X_i|} \qquad (4)$$

Definition: *For a given node of a circuit the probability of transition either from 0 to 1 or from 1 to 0 is known as switching activity (SA) of that node.*

Thus, switching activity of node *i* is given by the composite probability

$$SA = P_0 \times P_1 = \frac{|N_i| \times |X_i|}{[|N_i| + |X_i|]^2} \qquad (5)$$

For instance the switching activity of 2-input AND Gate ($|Ni| = 1$ and $|Xi| = 3$) is given by $\frac{3}{16}$ and this contributes to the dynamic power loss of the AND Gate.

As already mentioned, power dissipation in digital circuits can be reduced by minimizing its total switching activity. In order to minimize the switching activity one must know the minimum and maximum values of switching activity for the logical expression for a particular switching circuit. The total numbers of values (0's and 1's) in the output column of a truth table are dependent solely on the number of inputs and thus may be considered to be invariant for a given function.

Thus, $|Ni| + |Xi| = c$ (constant) for a given switching function. Thus, from Equation 5, minimum and maximum values of SA are determined by the minimum or maximum values of the product $M = |Ni| \times |Xi|$.

$M = |Ni| \times |Xi|$ can be written as $\frac{(|N_i| + |X_i|)^2 - (|N_i| - |X_i|)^2}{4} = \frac{(c)^2 - (|N_i| - |X_i|)^2}{4}$.

Switching activity is maximum when $(|N_i| - |X_i|)^2$ is minimum. Minimum value of $(|N_i| - |X_i|)^2$ is zero and hence Switching activity is maximum when number of 1's (| Xi |) and number of 0's (| Ni |) in the output column of the truth table are equal. Now switching activity is minimum when $M = \frac{(c)^2 - (|N_i| - |X_i|)^2}{4}$ is minimum i.e. $(|N_i| - |X_i|)^2$ is maximum. Thus the value of M and hence the switching activity of the logical expression of the switching function will be minimum when the difference between the number of zeroes and number of ones in the output column in the truth table is maximum.

3.1 Calculation of Switching Activities of Logic Gates:

To calculate the switching activity for a logical expression of a switching circuit it is important to determine the switching activity for the constituent logic gates. Based on the classical probabilistic definition of switching activity [19], we can easily calculate the switching activity of the basic gates. For an AND or an OR, or a NAND, or a NOR gate with *n* inputs the output is 0 or 1 is exactly one for one input only. Hence the value of | *Ni* | or | *Xi* | is 1 or $2^n - 1$. Hence $P_0 = \frac{1}{2^n}$ or $\frac{2^n - 1}{2^n}$ and corresponding $P_1 = \frac{2^n - 1}{2^n}$ or $\frac{1}{2^n}$. Thus, switching activity is given by $\frac{2^n - 1}{2^{2n}}$. Hence as the number of inputs to the above mentioned logic gates increases the switching activity of these gates decreases.

It is clear to see that the switching activity for NOT Gate is maximum and is of value $\frac{1}{4}$ since the number of 0s and 1s in this Gate are equal.

The Switching Activity for XOR Gate and XNOR Gate are independent of the number of inputs to the Gate and is equal to $\frac{1}{4}$. Consider an *n*-input XOR Gate having inputs $x_{n-1}, x_{n-2}, ... x_1, x_0$. The output of the XOR Gate is $(x_{n-1} + x_{n-2} + ... + x_1 + x_0)\%2$ where $x_i \in (0,1)$. Thus, for a particular input of n values the output of XOR Gate is 1 if number of 1s in input is odd. It is clear to see that the number of 1s in the output node of the XOR gate is $2^{(n-1)}$, as the number of inputs having odd number of 1's is $2^{(n-1)}$. Hence $P_0 = \frac{1}{2}$ and $P_1 = \frac{1}{2}$. Therefore switching activity of XOR Gate is $\frac{1}{4}$. Since XNOR gate is the inverse of the XOR gate, the switching activity for the XOR gate is also $\frac{1}{4}$.

Overall switching activity of a given logical expression depends on the number and types of gates required for the implementation of the function. Thus, to reduce the power of digital circuits by minimization of switching activity, it is desired to compute the total switching activity for a logical expression of the circuit. This is discussed in the next Section.

3.2 Calculation of Switching Activity for a Logical expression:

Without loss of generality, we assume the logic gates to have two inputs. Computation of switching activity for a logical expression of a switching circuit is illustrated through an example.

Consider a logical expression $f = a\bar{b}c + \overline{ab} + \overline{c+d}$. The switching activity for $f1 = a\bar{b}c$ is $\frac{7}{64}$. This is due to the fact that the output is 1 only for the input vector 101. As such, the number of 1's and 0's in the output column are respectively 1 and 7. For the implementation of $f1 = a\bar{b}c$, a NOT Gate is required for $\bar{b}$, having switching activity $\frac{1}{4}$, an AND gate is required for ac with switching activity $\frac{3}{16}$. The outputs of these NOT Gate and AND Gate are inputs to a second AND Gate having switching activity $= \frac{7}{64}$. The total switching activity for f1 is thus $\frac{1}{4} + \frac{3}{16} + \frac{7}{64}$.

Clearly, the switching activities for both $f2 = ab$ and $f3 = \overline{c+d}$ are $\frac{3}{16}$. When represented as a minterm, the function is given by $f = \sum\ 0,1,2,3,4,5,6,7,8,9,10,11,12$ and the activity is $\frac{39}{256}$ (as |*Ni*|=3,|*Xi*|=13). Thus, the total switching activity of the circuit is $\frac{1}{4} + \frac{3}{16} + \frac{7}{64} + \frac{7}{64} + \frac{3}{16} + \frac{3}{16} + \frac{39}{256} = \frac{303}{256}$ (Figure 1).

4 MOTIVATION OF THE WORK

According to International Technology Roadmap for Semiconductor (ITRS) one of the major goals of the semiconductor industry is to be able to continue to scale the technology in overall performance. The performance of the components and the final chip can be measured in many different ways; higher speed, higher density, lower power, more functionality, etc [27]. As already discussed, reduction in feature sizes of VLSI circuits results in increased power dissipation. The dissipated power increases the chip temperature causing its malfunctioning. Hence minimization of power dissipation is one of the challenging tasks of the designers. Again according to ITRS $\boldsymbol{V_{dd}}$ is more difficult to scale compared to other parameters, mainly because of the fundamental limit of the sub-threshold slope of 60 mV/decade. This trend will continue and become more severe when it approaches the regime of 0.6 V. This fact along with the continued increase of current density (per area) causes the dynamic power density (proportional to V_{dd}^2) to increase with scaling (although power per transistor is dropping). For high-performance logic, along with the trend of increasing chip complexity and increasing transistor on-current with scaling, controlling chip static power dissipation is expected to become particularly difficult simultaneously with meeting aggressive targets for performance scaling [27]. Minimization of power dissipation of VLSI circuits necessitates the minimization of dynamic power and hence of switching activity of the logical expression of the switching functions. Logic optimization is often achieved using the standard SOP or POS expression. However, the focus of earlier works in logic optimization is primarily on the reduction of number of input terms or number of literals [6]. An algorithm for interconnection-aware two-level logic optimization of multi-output SOP functions appears in [28]. However, to the best of our knowledge, there does not appear any significant effort on logic optimization using SOP or POS that attempts to optimize switching activity and hence dynamic power. An algorithmic approach using k-map for reducing the switching activity in combinational logic circuits is proposed in [20]. However, the use of k-maps restricts the number of variables to around 6. Moreover, for the method proposed in [20], switching activity can be minimized only for some specific switching functions. In this paper we attempt to design a generalized method for any number of variables. The proposed method accepts any SOP or POS expression as input and transforms it into an equivalent expression with minimal switching activity. Figure 2 illustrates a motivating example. Consider the logical expression of Full Subtractor with b_{in}, x, y as input variables and b_{out} and diff as output variables. The logical expression corresponding to b_{out} is given by $\bar{x}y + \bar{x}b_{in} + yb_{in}$ and the total switching activity of this logical expression will be $\frac{83}{64}$. But the logical expression for borrow can also be written as $(x(y+b_{in}) \oplus (y+b_{in})) + yb_{in}$ and the switching activity for this expression would be $\frac{70}{64}$.

5 PROBLEM FORMULATION

Consider a given Boolean (switching) expression *Ei* of *n* input terms in SOP form, each term containing p variables denoted x_i, $0 \leq i < p$. For such a given set of input Boolean terms in SOP (or POS) form, the objective here is to find a set E_o of m Boolean terms, $m \leq n$, not necessarily in SOP, that is equivalent (defined below) to E_i such that the switching activity for realization of E_o is minimal. Thus, the problem considered here is a minimum cover problem, the latter being a well-known NP-hard problem [29], where cost of the cover is measured in terms of its switching activity. In this case, E_o is said to minimally cover E_o. The notion of equivalence of two logical expressions is defined as follows.

Definition 2: Two switching functions $f1(x_{n-1}, x_{n-2}, ..., x_1, x_0)$ and $f2(x_{n-1}, x_{n-2}, ..., x_1, x_0)$ are said to be logically equivalent (or simply, equivalent) if and only if both functions have the same value for each and every combination of the variables ($x_{n-1}, x_{n-2}, ..., x_1, x_0$) [6].

In the next Section, we discuss some results which form the basis of the proposed method for finding the minimal cover.

6 DESIGN FOR MINIMAL SWITCHING ACTIVITY

Based on the discussions in Section 3.1 and Section 3.2, it is observed that the switching activity of a circuit realization depends on its corresponding input logical expression. As such, derivation of an appropriate logical expression equivalent to a given one and ensuring minimal switching activity is very useful.

Switching activity, given by $(\frac{1}{4})$, is maximum for a NOT Gate. Thus, it is desirable to minimize the number of NOT Gates in a logical expression.

The following observation is a clear consequence of applying De Morgan's theorem [6].

Observation 1: *If a product (sum) term contains even number of complemented literals then replacing each pair of complemented literals with their NOR (NAND) combination by application of De' Morgan's theorem will reduce the switching activity of the entire term.*

For instance, consider a logical expression $A\overline{B}C\overline{D}$. Two NOT Gates are required for this logical expression. Thus, the total Switching Activity for this logical expression is $\frac{3}{16}+\frac{1}{4}+\frac{1}{4}+\frac{3}{16}+\frac{15}{256}=\frac{239}{256}$ (Figure 3a). Now this product term can be modified to $AC\overline{B+D}$.

The total switching activity for the modified term is $\frac{3}{16}+\frac{3}{16}+\frac{15}{256}=\frac{111}{256}$ as shown in Figure 3b, which is clearly less than that of the original term.

Lemma 1: The logical expressions for n variable switching functions $f1(x_{n-1},x_{n-2},...,x_1,x_0)=(x_{n-1}*x_{n-2})*(x_{n-3}*x_{n-4})*......*(x_1*x_0)$ when n is even or

$f1(x_{n-1},x_{n-2},...,x_1,x_0)=(x_{n-1}*x_{n-2})*(x_{n-3}*x_{n-4})*......*(x_2*x_1)*x_0$ when n is odd and $f2(x_{n-1},x_{n-2},...,x_1,x_0)=((..((x_{n-1}*x_{n-2})*x_{n-3})*x_{n-4})*......*x_1)*x_0)$ are logically equivalent where * represents Sum or Product operation.

Observation 2: For both AND gate and OR Gates, the switching activity decreases with increasing number of inputs. Thus, if a product (or sum) term contains more than two literals (e.g., $f=x_{n-1}*x_{n-2}*x_{n-3}*x_{n-4}*......*x_1*x_0)$, the function can be implemented as $((..((x_{n-1}*x_{n-2})*x_{n-3})*.....*x_1)*x_0)$ instead of $(x_{n-1}*x_{n-2})*(x_{n-3}*x_{n-4})*......*(x_1*x_0)$ if n is even or $(x_{n-1}*x_{n-2})*(x_{n-3}*x_{n-4})*......*(x_2*x_1)*x_0$ if n is odd, will reduce the switching activity.

Figure 4 shows the implementation of a 6-variable (A, B, C, D, E, F) product term. With the first implementation switching activity $\frac{1643}{4096}$ is less in comparison to second one $\frac{2067}{4096}$. Similar type of operation can be done in sum term are as well.

Observation 3: If a sum-of-product expression contains pair-wise common Boolean literals with multiplication distributed over addition like $f1 = A.B + B.C$ then modification of this expression applying converse distributive property i.e. $f1 = A.B + B.C = f2 = B.(A + C)$ reduces the number of gates as well as switching activity. Figure 5 shows the reduction of switching activity due to such an arrangement. For instance, implementation of *f1* requires 2 AND Gates and 1 OR Gate, whereas that of *f2* requires 1 AND Gate and 1 OR Gate. The following observations follow from Observation 3.

The following observations follow from Observation 3.

Observation 4: Modification of the logical expression $A_1.B_1 + A_2.B_1 + A_1.B_2 + A_2.B_2 + ... + A_1.B_n + A_2.B_n$ to equivalent logical expression $(A_1 + A_2).(B_1 + B_2 + ... + B_n)$ reduces the number of logic gates and switching activity as well. This is illustrated in Figure 6.

Observation 5: Modification of the logical expression $A_1.A_2......A_n.B_1.B_2.....B_m + B_1.B_2.....B_m.C_1.C_2...C_p$ to equivalent logical expression $B_1.B_2.....B_m.(A_1.A_2......A_n. + C_1.C_2.....C_p)$ reduces both the number of logic gates and the switching activity. Here n, m and p are in general different.

Observation 6: Modification of the logical expression $A_1.A_2......A_n.B_1.B_2.....B_m + B_1.B_2.....B_k.C_1.C_2...C_p$ to equivalent logical expression $B_1.B_2.....B_k.(A_1.A_2......A_n.B_{k+1}B_{k+2}...B_m + C_1.C_2...C_p)$ reduces both the number of logic gates and the switching activity. Here n, m, p and k are in general different and $m \geq k$.

Observation 7: *Applying Consensus theorem i.e.* $AB + \overline{A}C + BC = AB + \overline{A}C$ *[6] will reduce the number of gates as well as switching activity as illustrated in Figure 7.*

Lemma 2: *For an n-variable logical expression, (n* $\geq$ *2),* $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_n}$ *is logically equivalent to* $A_1 + \overline{A_1 + A_2 + A_3..... + A_n}$ *i.e.* $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_n} = A_1 + \overline{A_1 + A_2 + A_3..... + A_n}$

Proof: $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_n} = A_1 + \overline{A_2 + A_3..... + A_n} = A_1 + \overline{B}$ *(say). Using perfect induction as shown in* Table 1 *it can be shown that* $A_1 + \overline{B} = A_1 + \overline{A_1 + B}$. *Hence* $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_n} = A_1 + \overline{A_2 + A_3..... + A_n}$ □

Observation 8: *Figure 8 shows the reduction of switching activity to the modification of*

2-variable Boolean expressions according to Lemma 2. This Lemma 2 can be effectively used to reduce the switching activity if the number of complemented term is odd. Hence this Lemma 2 can be applied in the proposed method described in this paper if n $\geq$ 2 and n is even.

Lemma 3: *For an n-variable logical expression, (n $\geq$ 3),* $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_{n-1}}.A_n$ *is logically equivalent to* $A_1 + \overline{A_1 + A_2 + A_3.....+A_{n-1}}.A_n$ *i.e.* $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_{n-1}}.A_n = A_1 + \overline{A_1 + A_2 + A_3.....+A_{n-1}}.A_n$

Proof: ***Let*** $B = A_2 + A_3 + ...+ A_{n-1}$.

$$\therefore A_1 + \overline{A_2}.\overline{A_3}......\overline{A_{n-1}}.A_n = A_1 + \overline{A_2 + A_3.....+A_{n-1}}A_n = A_1 + \overline{B}.A_n = A_1 + (A_1 + \overline{A_1})\overline{B}.A_n$$

$$A_1 + A_1 + \overline{A_1}\,\overline{B}.A_n = A_1 + \overline{A_1}\,\overline{B}.A_n = A_1 + \overline{A_1 + B}.A_n = A_1 + \overline{A_1 + A_2 + A_3.....+A_{n-1}}.A_n \qquad □$$

Observation 9: *Figure 9 shows the reduction of switching activity to the modification of 3-variable Boolean expressions according to Lemma 3..*

Lemma 4: *For two or more variables the Boolean expression* $\overline{A} + AB$ *is logically equivalent to* $((\overline{A+B}) + B)$ *i.e.* $\overline{A} + AB = ((\overline{A+B}) + B)$.

Proof: *This can be proved using perfect induction as shown in* Table 2.

The following Corollary is an obvious consequence of the Lemma 4.

Corollary 1: *A n variable Boolean expression given by* $\overline{A_1} + \overline{A_2} + ...+ \overline{A_{n-1}} + A_1.A_2....A_n$ *is logically equivalent to* $\overline{A_1.A_2......A_{n-1} + A_n} + A_n$.

Observation 10: *For two or more variables modification of Boolean expression* $\overline{A} + AB$ *to* $((\overline{A+B}) + B)$ *reduce the switching activity as shown in Figure 10.*

Observation 11: Modification of Boolean expression $\overline{A_1} + \overline{A_2} + ...+ \overline{A_{n-1}} + A_1.A_2....A_n$ to $\overline{A_1.A_2......A_{n-1} + A_n} + A_n$ reduce the switching activity as shown in Figure 11.

Lemma 5: For two or more variables the Boolean expression $\overline{A}B$ is logically equivalent to $(\overline{AB})B$ i.e. $\overline{A}B = (\overline{AB})B$.

Proof: $\overline{A}B = \overline{A}B + \overline{B}B = B(\overline{A} + \overline{B}) = (\overline{AB})B$. □

Observation 12: Modification of Boolean expression according to Lemma 5 will reduce the switching activity as shown in Figure 13.

Lemma 6: An n variable Boolean expression $\overline{A_1}.A_2.A_3....A_n$ is logically equivalent to $\overline{A_1.A_2......A_n}.A_2.A_3....A_n$.

Proof: Proof can be easily derived from Lemma 5. □

Observation 13: Modification of Boolean expression according to Lemma 6 reduces the switching activity as shown in Figure 12.

Lemma 7 [6]: $A\overline{B}+B$ is logically equivalent to $A+B$. This is known as absorption rule.

Proof: $A\overline{B}+B = A\overline{B}+B(A+\overline{A}) = A\overline{B}+AB+\overline{A}B = A\overline{B}+AB+AB+\overline{A}B =$

$= A(\overline{B}+B)+B(A+\overline{A}) = A+B$ □

Observation 14: *It is clear from Figure 14 that the modification of logical expression according to Lemma 7 will reduce the number of logic Gates as well as switching activity of the logical expression.*

The following corollaries can be easily obtained from Lemma 7.

Corollary 2: $\overline{A}B+\overline{B}$ is logically equivalent to $\overline{AB}$. This can be defined as modified absorption rule.

Observation 15: It is clear from Figure 15 that the modification of logical expression according to Corollary 2 will reduce the number of logic Gates as well as switching activity of the logical expression.

Corollary 3: For three or more variables the Boolean expression $\overline{A}C+A\overline{B}C$ is logically equivalent to $C(\overline{AB})$. This can also be defined as modified absorption rule.

Observation 16: Modification of Boolean expression according to Corollary 3 will reduce the switching activity as shown in Figure 16.

Lemma 8: *For any* $n \geq 2, t_n = \frac{1}{2^n} - \frac{1}{2^{2n}} < \frac{1}{4}$.

Proof: [By Induction]

For n=2, $t_2 = \frac{1}{2^2} - \frac{1}{2^4} = \frac{1}{4} - \frac{1}{64} = \frac{3}{16} < \frac{1}{4}$. Let us assume that the results holds for n=k variables. i.e. $t_k = \frac{1}{2^k} - \frac{1}{2^{2k}} < \frac{1}{4}$.Now we need to show that the result also holds for n = k + 1 variables. $t_{(k+1)} = \frac{1}{2^{(k+1)}} - \frac{1}{2^{2(k+1)}} = \frac{1}{2}[\frac{1}{2^k} - \frac{1}{2^{2k}}] - \frac{1}{2^{(2k+1)}} = \frac{1}{2}t_k - \frac{1}{2^{(2k+1)}} < \frac{1}{4}$ □

Lemma 9: If a product term in a logical expression contains odd number of complemented literals, then switching activity can be reduced by replacing pair of complemented literals with their NOR combination and replacing remaining one complemented literal $\overline{x_i}$ by $x_i \oplus 1$. And if product term contains only one complemented literal $\overline{x_i}$ then switching activity can be reduced by replacing complemented literal $\overline{x_i}$ by $x_i \oplus 1$.

Proof: [By Induction]

We consider a product term in a logical expression containing two literals of the form $x\overline{y}$. From Figure 17a and Figure 17c it is evident that the switching activity for $\overline{y}$ is $\frac{1}{4}$ and that of for $x\overline{y}$ is $\frac{3}{16}$. Thus, the total switching activity in this case is given by

$$TSA_2 1c = \frac{1}{4} + \frac{3}{16} = \frac{7}{16} \qquad (8)$$

($TSA_i jc$ represents Total Switching Activity of a product term containing i number of literals out of which j number of literals are in complemented form and $TSA_i jm$ represents the same of a modified product term.)

Now the expression $x\overline{y}$ can be modified as $x(y \oplus 1) = xy \oplus x$. Corresponding truth table and circuit diagram are shown in Figure 17c and Figure 17b respectively.

From Figure 17c and Figure 17b the switching activity for xy is $\frac{3}{16}$ and that of for $xy \oplus x$ is $\frac{3}{16}$. So total switching activity in this case

$$TSA_2 1m = \frac{3}{16} + \frac{3}{16} = \frac{6}{16} \qquad (9)$$

Comparing the equations 8 and 9 it is clear to see that $TSA_2 1m < TSA_2 1c$.

Now we consider a product term containing three literals with one literal in complemented form, for example, consider the logical expression of the form $zy\bar{x}$.

From Figure 18a and Figure 18c total switching activity for $zy\bar{x}$ is given by

$$TSA_3 1c = \frac{3}{16} + \frac{1}{4} + \frac{7}{64} = \frac{35}{64} \qquad (10)$$

The product term can be modified to $zy(x \oplus 1) = zyx \oplus zy$.

From Figure 18c and Figure 18b, the switching activity of zy, zyx and $zyx \oplus zy$ are respectively given by $\frac{3}{16}$, $\frac{7}{64}$ and $\frac{7}{64}$. Thus, total switching activity for this implementation is

$$TSA_3 1m = \frac{3}{16} + \frac{7}{64} + \frac{7}{64} = \frac{26}{64} \qquad (11)$$

It is clear to see that $TSA_3 1m < TSA_3 1c$.

Consider a product term containing three literals all of which are in complemented form, e.g. $\bar{z}.\bar{y}.\bar{x}$. This product term can be modified to $\overline{(z+y)}..\bar{x}$. The corresponding truth table is shown in Table 3 and the circuit diagram shown in Figure 19a. From Table 3 and Figure 19a the switching activity of $\overline{(z+y)}$, $\bar{x}$ and $\overline{(z+y)}..\bar{x}$ respectively are $\frac{3}{16}, \frac{1}{4}, \frac{7}{64}$. Hence, the total switching activity is given by

$$TSA_3 3c = \frac{3}{16} + \frac{1}{4} + \frac{7}{64} = \frac{35}{64} \qquad (12)$$

The above product term can be modified as $\overline{(z+y)}.(x \oplus 1) = \overline{(z+y)}.x \oplus \overline{(z+y)}$. Figure 19b and Table 3 show that switching activity for $\overline{(z+y)}$, $\overline{(z+y)}x$, $\overline{(z+y)}.x \oplus \overline{(z+y)}$. are $\frac{3}{16}$, $\frac{7}{64}$, $\frac{7}{64}$ respectively. Thus, the total switching activity is given by

$$TSA_3 3m = \frac{3}{16} + \frac{7}{64} + \frac{7}{64} = \frac{26}{64} \qquad (13)$$

Thus, from the equations 12 and 13 it is clear that $TSA_3 3m < TSA_3 3c$. Thus, product terms containing 2 and 3 literals with odd number of complemented literals are in general true.

Now we assume that the result will be true for k literals containing odd number of complemented literals.

We need to show that the result also holds for (k+1) literals containing odd number of complemented literals.

Let the product term be $f = x_k x_{k-1} x_{k-2} .... x_{i+1} x_i x_{i-1} .... x_2 x_1 x_0$ where number of complemented literals are odd. In this case, any of the literals are either in complemented or in regular (uncomplemented) form, of which the total number of complemented terms is always odd. Now suppose x_i is in complemented form ($\overline{y_i}$ say). Thus, $f = (\overline{y_i}) x_k x_{k-1} x_{k-2} .... x_{i+1} x_{i-1} .... x_2 x_1 x_0$. Now $p = x_k x_{k-1} x_{k-2} .... x_{i+1} x_{i-1} .... x_2 x_1 x_0$ contains even number of complemented terms. However, these even numbers of complemented literals can be modified using De' Morgan theorem. It is clear to see that the switching activity for product $p = x_k x_{k-1} x_{k-2} .... x_{i+1} x_{i-1} .... x_2 x_1 x_0$ is fixed, (say *m*). Suppose the switching activity for the logical expression *f* is $n < \frac{1}{4}$ from Lemma 8.

Now let, $f = \bar{y} p$. So using NOT gate total switching activity from Figure 20a is

$$TSA_k qc = m + n + \frac{1}{4}. \quad (q \text{ is odd}) \qquad (14)$$

Now the expression for f can be modified as $f = \bar{y} p = (y \oplus 1) p = yp \oplus p$. Hence, in Figure 20b, total switching activity is given by

$$TSA_k qm = m + n + n \qquad (15)$$

Comparing equations 14 and 15 $TSA_k qc < TSA_k qm$. □

Lemma 10: $A + \bar{B}$ is logically equivalent to AB ⊙ *B.*

Proof: $A + \bar{B} = \bar{\bar{A}} + \bar{B} = \overline{\bar{A}B} = \overline{A \oplus B} =$ AB ⊙ *B.* □

Lemma 11: *If a sum term in a logical expression contains odd numbers of complemented terms then switching activity can be reduced by introducing XNOR gate.*

Proof. *This Lemma is basically the generalized form of Lemma 10 as* $\sum_{i=1}^{p} x_i + \sum_{j=k}^{2k} \overline{x}_j + \overline{x_l} = \sum_{i}^{p} x_i + \overline{x_k x_{k+1}} + ... + \overline{x_{2k-1} x_{2k}} + \overline{x_l} = E + \overline{x_l}$ *where.*

$E = \sum_{i}^{p} x_i + \overline{x_k x_{k+1}} + ... + \overline{x_{2k-1} x_{2k}} + \overline{x_l}$ *. The proof of this lemma is similar to that of Lemma 9.*

From Lemma 9 and Lemma 11 it is clear to see that the introduction of XOR or XNOR Gate in sum term or in product term with odd number of literals reduces the total Switching Activity. Hence it is expected that by the introduction of the above mentioned Logic Gates under above mentioned conditions reduce the power loss the digital circuits.

Lemma 12: *If a product term of n numbers of variables* $n \geq 2$ *consists of odd number of complemented terms then Lemma 5 can be generalized to generate logically equivalent expression with reduced switching activity.*

Proof: Consider a product term containing three literals with one in complemented form say $(\overline{A}BC)$, this can be generalized as $((\overline{ABC})BC)$. Figure 21 clearly shows the reduction of switching activity with this modification. Now consider a product term containing three literals with all in complemented form say $(\overline{A})(\overline{B})(\overline{C})$. This product term can be generalized as $(\overline{A+B})(\overline{C}) = (\overline{(\overline{A+B})C})(\overline{A+B})$. Figure 22 shows the reduction of corresponding switching activity. The rest of the part of the proof can be shown in the same way as the proof of Lemma 9.

7 Algorithm For Minimization Of Total Switching Activity

In this section we propose an algorithm to minimize the total switching activity of digital circuits. The proposed rule-based algorithm for minimization of switching activity in a circuit is shown in Figure 24. The algorithm takes the truth table of any switching function as input and yields an equivalent logical expression of the switching function with minimized switching activity as output. Since NOT has maximum switching activity the algorithm tries to reduce the use of NOT operator. If there are even numbers of complemented terms then using De 'Morgan's theorem a pair of complemented literals is replaced either with their corresponding NOR combination (for a product term) or with a NAND combination (for a sum term). Based on the discussions in the previous Section, the following sets of rules are defined in an attempt to reduce the switching activity.

Rule 1: From Observations 3, 4, 5 and 6 applying $AB + BC = B(A + C)$ and the corresponding equations discussed in Observations 4, 5 and 6;

Rule 2: Applying $x_{n-1} * x_{n-2} * x_{n-3} * x_{n-4} * ...... * x_1 * x_0 = (..((x_{n-1} * x_{n-2}) * x_{n-3}) * ..... * x_0)$ from Observation 2.

Rule 3: Apply De' Morgan's Theorem i.e. $\overline{A}.\overline{B} = \overline{A + B}$ and $\overline{A}. + \overline{B} = \overline{AB}$.

Rule 4: From Observation 7 apply Consensus Theorem i.e. $AB + \overline{A}C + BC = AB + \overline{A}C$.

Rule 5: From Observation 8 if (n≥2) and n is even then apply $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_n} = A_1 + \overline{A_1 + A_2 + A_3 ..... + A_n}$.

Rule 6: Apply $A_1 + \overline{A_2}.\overline{A_3}......\overline{A_{n-1}}.A_n = A_1 + \overline{A_1 + A_2 + A_3 ..... + A_{n-1}}.A_n$ from Observation 9 if $n \geq 3$.

Rule 7: From Observation 10 applying $\overline{A} + AB = ((\overline{A + B}) + B)$ and from Observation 11 applying $\overline{A_1} + \overline{A_2} + ... + \overline{A_{n-1}} + A_1.A_2....A_n = \overline{A_1.A_2......A_{n-1} + A_n} + A_n$.

Rule 8: From Observation 12 applying $\overline{A}B = (\overline{AB})B$ and from Observation 13 applying $\overline{A_1}.A_2.A_3....A_n = \overline{A_1.A_2......A_n}.A_2.A_3....A_n$.

Rule 9: From Observation 14 applying $A\overline{B} + B = A + B$

Rule 10: From Observation 15 applying $\overline{A}B + \overline{B} = \overline{AB}$.

Rule 11: From Observation 16 applying $\overline{A}C + A\overline{B}C = C(\overline{AB})$.

Rule 12: From Lemma 9 applying $\overline{A}B = AB \oplus B$.

Rule 13: From Lemma 10 applying $\overline{A} + B = AB \odot B$

Rule 14 : $Apply (\prod_{i=1}^{n} x_i).(\prod_{j=1}^{2p} \overline{x_j}).(\overline{x_k}) = (\prod_{i=1}^{n} x_i).(\overline{x_1 + x_2})......(\overline{x_{2p-1} + x_{2p}})(x_k \oplus 1) =$

$$(\prod_{i=1}^{n} x_i).(\prod_{j=1}^{2p} \overline{x_j}).(\overline{x_k}) = (\prod_{i=1}^{n} x_i).(\overline{x_1 + x_2})......(\overline{x_{2p-1} + x_{2p}})x_k \oplus (\prod_{i=1}^{n} x_i).(\prod_{j=1}^{2p} \overline{x_j}).(\overline{x_k}) = (\prod_{i=1}^{n} x_i).(\overline{x_1 + x_2})......(\overline{x_{2p-1} + x_{2p}})x_k$$

From Lemma 9

Rule 15: $\sum_{i=1}^{p} x_i + \sum_{j=k}^{2k} \overline{x}_j + \overline{x_l} = \sum_{i}^{p} x_i + \overline{x_k x_{k+1}} + ... + \overline{x_{2k-1} x_{2k}} + \overline{x_l} = E + \overline{x_l}\, Ex_l \odot x_l$ where

$$E = \sum_{i}^{p} x_i + \overline{x_k x_{k+1}} + ... + \overline{x_{2k-1} x_{2k}} + \overline{x_l}$$

Algorithm Minimize Switching Activity ()

```
Input: Truth Table
Output: Function for minimal switching activity

Obtain the minimal Sum-of-Product ($f_{SOP}$) or Product-of-Sum ($f_{POS}$) expression using any standard method for the minimization of given switching function f.
Apply Rule 1 and Rule 2 to reduce the switching activity
If either the ($f_{SOP}$) or ($f_{POS}$) does not contain any complemented variable then
    Calculate the total switching activity of the function corresponding to ($f_{SOP}$) or ($f_{POS}$).
    Take the function with minimum switching activity.
Endif
If the product term in $f_{SOP}$ contains even number of complemented terms then
    Replace a pair of complemented terms by their corresponding NOR combination using De' Morgan's theorem (Rule 3).
Endif
If the sum term in $f_{POS}$ contains even number of complemented terms then
    Replace a pair of complemented terms by NAND Gate using De, Morgan's theorem (Rule 3).
Endif
If applicable then
    Apply Rule 4, Rule 6, Rule 7, Rule 9, Rule 10, and Rule 11 to reduce the switching activity.
Endif
If a $f_{SOP}$ ($f_{POS}$) contains only one complemented term $x_i$ then
    Apply Rule 8 or Rule 12 (in case of SOP) from Lemma 12.
    Apply Rule 5 or Rule 13(in case of POS).
    Take the function with minimum switching activity.
Endif
If the product term in $f_{SOP}$ ($f_{POS}$) contains odd number of complemented terms then
    Apply Rule 14 in case of SOP.
    Or Apply Rule 15 in case of POS
    Take the function with minimum switching activity.
Endif
End.
```

Figure 24: Algorithm for Minimizing Switching activity

8 EXPERIMENTAL RESULTS

The proposed rule-based algorithm is implemented in Xilinx 14.7 and power estimation has done using Synopsys EDA tool -DESIGN VISION version I-2013.12-SP1, 20, 2014 under CENT OS and using TSMC 120 nm library.

Without loss of generality, for our simulations, we consider only 2-input logic gates. The switching activity of some basic circuits and the associated dynamic power dissipation using conventional SOP (POS) method and our proposed method are summarized in Table 4.

We observe that the total number of switching activity for our proposed method never exceeds, and is less in most of the cases than those obtained using the traditional logic optimization.

8.1 Comparison of our proposed method with the existing method of [20]

Reduction of switching activity and hence power dissipation of VLSI CMOS circuits by our proposed algorithm is applicable for any number of input variables and any kind of circuits. In [20] the authors basically modify the k-map to reduce the switching activity. Logic optimization using k-map is limited to 6 variables. Moreover, the method of [20] is not applicable for all type of switching functions. For instance if the logical expression of a 2-variable switching function is $\overline{A}B$ then the method of [20] cannot reduce the switching activity. On the other hand, our proposed method is capable of reducing the same. In [20] around 10% of reduction of switching activity is observed. Our experimental result shows over 34% reduction of switching activity.

8.2 Power-Delay tradeoff

Logic optimization using our proposed method surely minimizes the switching activity. But in order to minimize the switching activity we often use NOR gate instead of a single NOT Gate. In this case the total transition count of the circuit increases which results in increasing circuit delay. In our proposed method we do not consider delay of circuits, even though delay in different Gates may cause glitches resulting in power loss as shown in Figure 23 [23]. Interconnection-aware two-level optimization of multi-output SOP functions was discussed in [28]. Logic optimization to minimize switching activity and delay as joint objective will be one of the future directions of our work.

9 CONCLUSION

In this paper we propose a rule-based approach to reduce the switching activity of combinational logic circuits. This would reduce the dynamic power and total power dissipation, enabling the design of power-efficient circuits with several useful applications. Combinational logic circuits designed using rule-based algorithm proposed in this paper is interestingly applicable for any number of variables. Experimental results show over 34% reduction of switching activity. The works presented in this paper can be further improved by considering both delay and power as objective functions.

REFERENCES

[1] K. Roy, S. C. Prasad, Low Power CMOS VLSI Circuit Design, Wiley India Ed (2011).

[2] P Ghosal, T Samanta, H Rahaman, P Dasgupta, ”Thermal-aware placement of standard cells and gate arrays: Studies and observations” IEEE Computer Society Annual Symposium on VLSI, ISVLSI’08.(2008), pp 369-374.

[3] N. A. Sherwani, “Algorithms for VLSI Physical Design Automation” Springer, 3rd Ed(2008).

[4] N.Wehn and M. Munch, “Minimising power consumption in digital circuits and systems: An overview” Kleinheubacher Berichte, Band 43, Kleinheubach, Germany, Invited Talk(1999).
pp. 308-319,

[5] S. Ahuja, A. Lakshminarayana, S. K. Shukla, ”Low Power Design with High-Level Power Estimation and Power-Aware Synthesis” (2011), Springer Pub.

[6] Z. Kohavi, N. K. Jha:“Switching and Finite Automata Theory” Cambridge University Press, 3rd Edition(2010).

[7] J. Black, Electromigartion-A brief survey and some recent results, IEEE Transactions on Electron Devices (1969), Vol. 4.

[8] A. Ghosh, S. Devdas, K. Keutzer and J. White, “Estimation of Average Switching Activity in Combinational and Sequential Circuits”, 29th ACM/IEEE Design Automation Conference ,(1992), pp 253-259.

[9] K. Roy and S. C. Prasad, “Circuit Activity Based Logic Synthesis for Low Power Reliable Operations” IEEE Transactions on VLSI, (December 1993), Vol-1, No. 4, , pp 503-512.

[10] M. Alidina, J. Monterio, S. Davadas, A. Ghosh, M. Papaeftymiou, “Precomputation-Based Sequential Logic Optiomization for Low Power” IEEE International Conference on Computer Aided design, (1994), pp 74-81.

[11] A. P. Chandrakasan and R. W. Brodersen, :“Minimizing Power Consumption in Digital CMOS Circuits” Proceedings of the IEEE, (April 1995), Vol 83, No. 4.

[12] V. Krishna, R. Chandramouli and N. Ranganathan,: “Computation of Lower and Upper Bounds of Switching Activity: A Unified Approach”

[13] J. Monteiro, S. Devdas, A. Ghosh, K. Keutzer, J. White, “Estimation of average Switching Activity in Combinational Logic Circuits Using Symbolic Simulation” IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems (1997), Vol. 16, No. 1, pp. 121-127.

[14] P. Girard, C.Landraut, S.Pravossoudovitch, D.Severac,“Reducing Power Consumption During Test Application by Test Vector Ordering”, IEEE (1998).

[15] I. Brzozowski, A. Kos, “Minimization of Power Consumption in Digital Integrated Circuits of Reduction of Switching Activity”, 25th Euromicro Conference (1999), Vol. 1,

[16] J. H. Satyanarayana and K. K. Parhi, “Theoretical Analysis of Word-Level Switching Activity in the Presence of Glitching and Correlation” IEEE Transactions on VLSI, Vol. 8, No. 2. pp. 148-159.

[17] S. Chattopadhyay, N. Choudhary,: “Genetic Algorithm based Approach for Low Power Combinational Circuit Testing ” Procedings of the 16th International conference on VLSI Design(VLSI’03).

[18] S. Koziel, W. Szczesniak,: “Reducing Average and peak temparatures of VLSI CMOS circuits by means of evolutionary algorithm applied to high level systhesis”, Microelectronics Journal 34(2003), Elsevier, pp. 1167-1174.

[19] R.V.Menon, S. Chennupati, N. K. Samala, D. Radhakrishna and B. Izadi “Power Optimized Combinational Logic Design” Proceeding of the International Conference on Embeded Systems and Applications (June 2003), pp. 223-227.

[20] R.V.Menon, S. Chennupati, N. K. Samala, D. Radhakrishna and B. Izadi,:“Switching Activity Minimization in combinational Logic Design” Proceeding of the International Conf. on Embedded System and Application (2004), pp. 47-54.

[21] J. L. Wong, G. Qu, M. Potkonjak,“Power Minimization in QoS Sensitive Systems” IEEE Transactions on Very large Scale Integration(VLSI) Systems (2004),Vol. 12, No. 6, pp 553-560.

[22] M. Pedram,: “Power Minimization in IC Design: Principles and Applications”, Dept. of EE-systems, University of Southern California.

[23] D. Mlynek, Y. Leblebici, Design of VLSI System: http://emicroelectronics.free.fr/onlineCourses/VLSI/ch07.html

[24] S. Rao Ijjada, B. Ramparamesh, V. Malleswara Rao,“Reduction of Power Dissipation in Logic Circuits” International Journal of Computer Application, (2011) Vol-24, No. 6, pp 10-14.

[25] K. Kaur, A. Noor, “Strategies & Methodologies for Low Power VLSI Designs” A Review, International Journal of Advances in Engineering & Technology, (2011), Vol 1, Issue 2, pp 159-165

[26] M. Ghasemazar and M. Pedram,“Variation Aware Dynamic Power Management for Chip Mu;tiprocessor Architectures”, 978-3-9810801-7-9/DATE11/ c 2011 EDAA

[27] International Technology Roadmap for Semiconductor 2013 Edition: Process Intregration, Devices and Structures Summery:

[28] P. Dasgupta, P.Dasgupta, D. K. Das , “A Novel Algorithm for Interconnectaware Two level Optimization of Multi-output SOP functions” Proceeding of the 11th International Workshop on Boolean Problems, Freiberg (September 2014), pp 219-226.

[29] M R Garey and D S Johnson: Computers and Intractability: A Guide to Theory of NP-Completeness, (1979), W Q Freeman and Co.

Figures and Tables

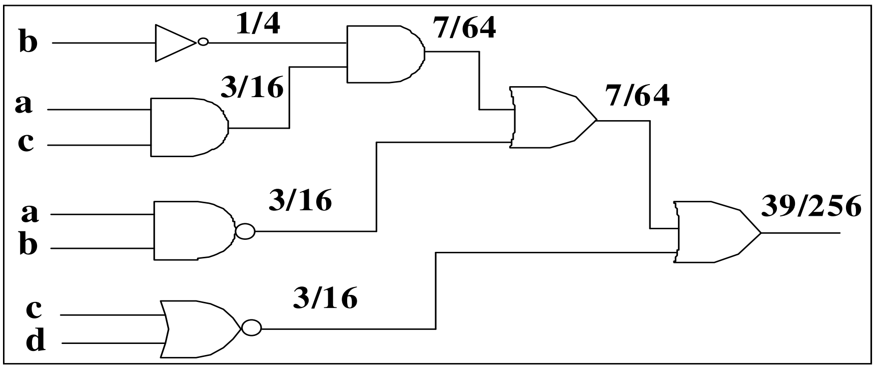


Figure1. Switching Activity for $a\bar{b}c + \overline{ab} + \overline{c+d}$

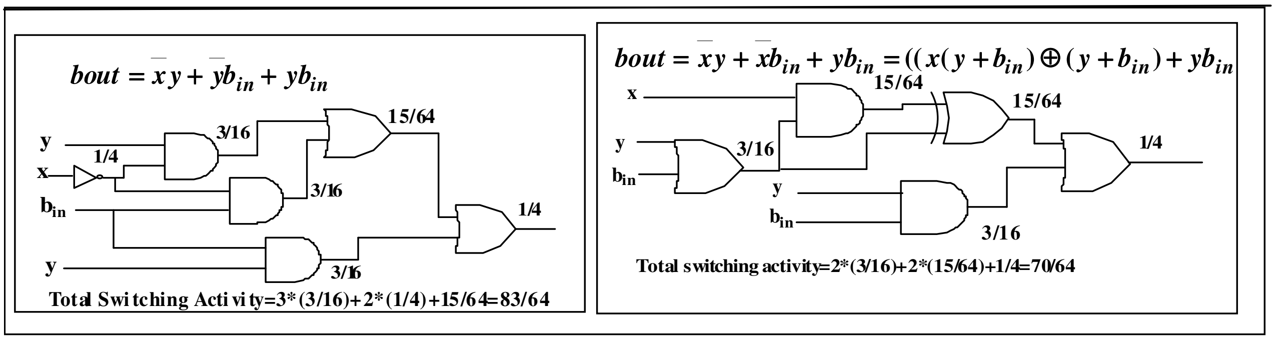
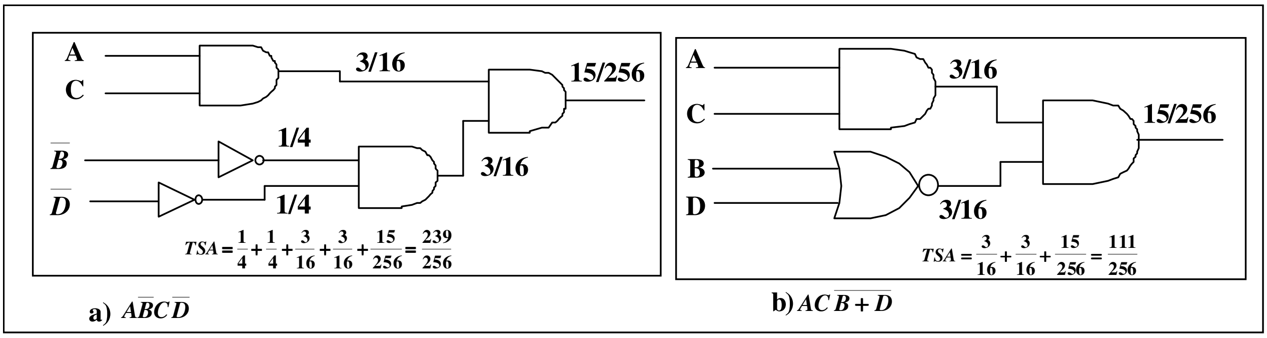

$bout = \bar{x}y + \bar{y}b_{in} + yb_{in}$
y
x
b_{in}
y
1/4
3/16
15/64
3/16
1/4
3/16
Total Switching Activity=3*(3/16)+2*(1/4)+15/64=83/64
$bout = \bar{x}y + \bar{x}b_{in} + yb_{in} = ((x(y+b_{in}) \oplus (y+b_{in}) + yb_{in}$
x
y
b_{in}
y
b_{in}
15/64
15/64
3/16
1/4
3/16
Total switching activity=2*(3/16)+2*(15/64)+1/4=70/64

Figure2. Reduction of Switching Activity for Full Subtractor


A
C
$\bar{B}$
$\bar{D}$
3/16
15/256
1/4
3/16
1/4
$TSA = \frac{1}{4} + \frac{1}{4} + \frac{3}{16} + \frac{3}{16} + \frac{15}{256} = \frac{239}{256}$
A
C
B
D
3/16
15/256
3/16
$TSA = \frac{3}{16} + \frac{3}{16} + \frac{15}{256} = \frac{111}{256}$

a) $A\bar{B}C\bar{D}$

b) $AC\overline{B+D}$

Figure 3: Switching Activity for a) $A\bar{B}C\bar{D}$ **and b)** $AC\overline{B+D}$

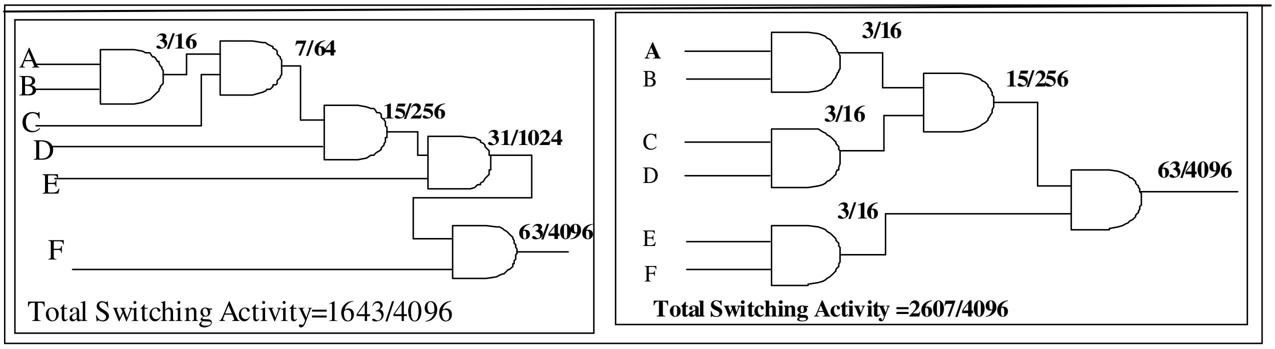

A
B
C
D
E
F
3/16
7/64
15/256
31/1024
63/4096
Total Switching Activity=1643/4096
A
B
C
D
E
F
3/16
15/256
3/16
3/16
63/4096
Total Switching Activity =2607/4096

Figure 4: Comparison of Switching Activity between ((((A*B)*C)*D)*E)*F) and ((A*B)*(C*D)*(E*F))

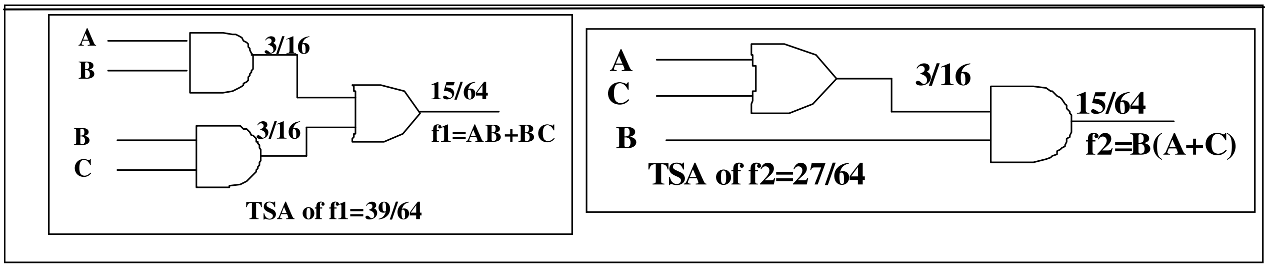


Figure 5: Comparison of Switching Activity between AB+BC and B (A+C)

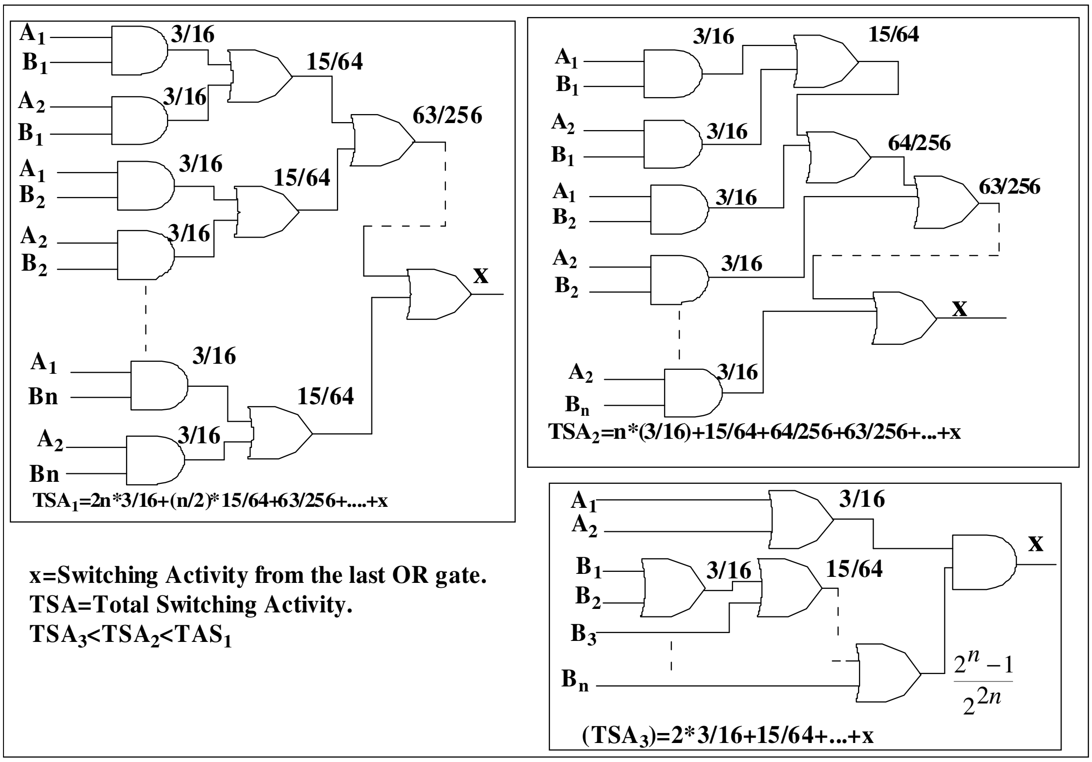


Figure 6. Comparison of Switching Activity for Observation 4

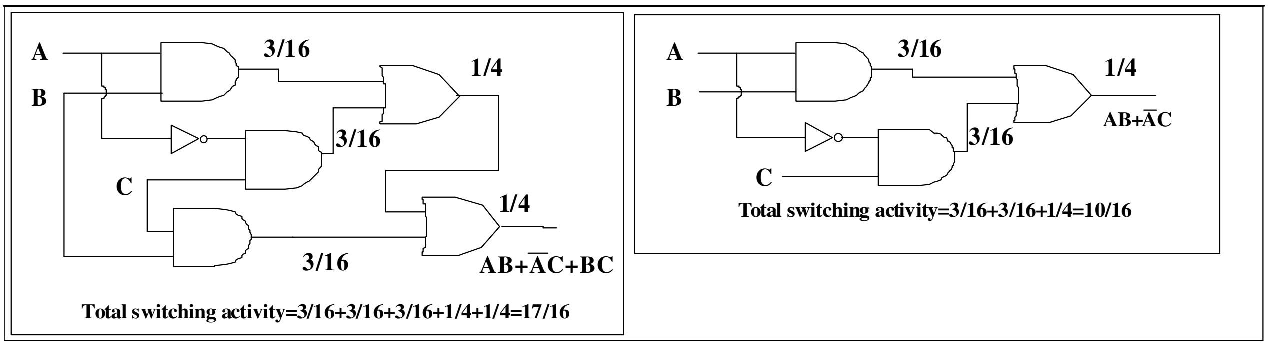

A
B
3/16
1/4
3/16
C
1/4
3/16
AB+AC+BC
Total switching activity=3/16+3/16+3/16+1/4+1/4=17/16
A
B
3/16
1/4
AB+AC
3/16
C
Total switching activity=3/16+3/16+1/4=10/16

Figure7. Reduction of number of Gates and Switching Activity using Consensus theorem

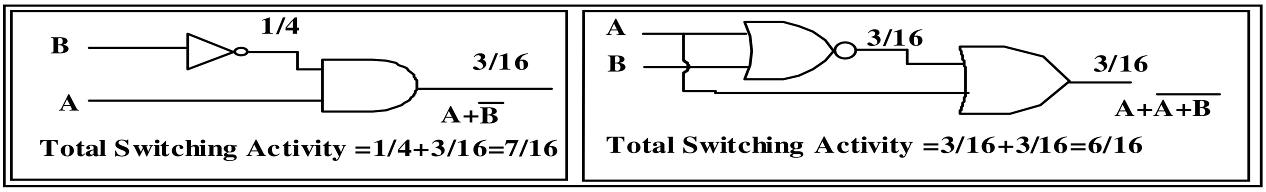

B
1/4
3/16
A
A+B
Total Switching Activity =1/4+3/16=7/16
A
B
3/16
3/16
A+A+B
Total Switching Activity =3/16+3/16=6/16

Figure 8: Switching Activity for $A+\overline{B}$ and $A+\overline{A+B}$

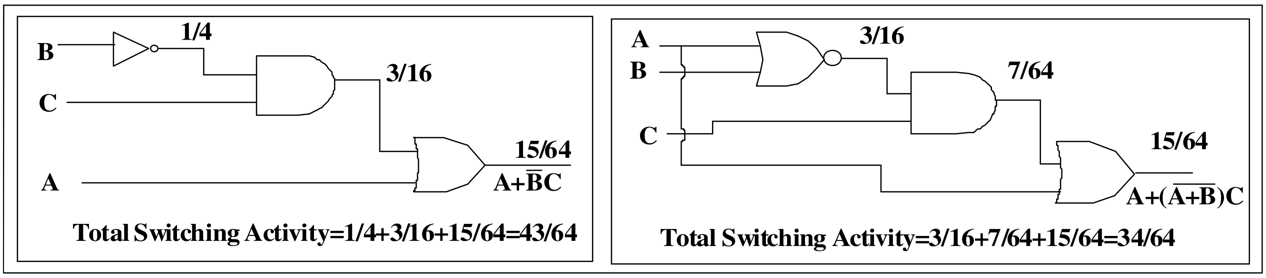

B
C
A
1/4
3/16
15/64
A+B̄C
Total Switching Activity=1/4+3/16+15/64=43/64
A
B
C
3/16
7/64
15/64
A+(A+B)C
Total Switching Activity=3/16+7/64+15/64=34/64

Figure 9: Switching Activity for $A+\overline{B}C$ and $A+(\overline{A+B})C$

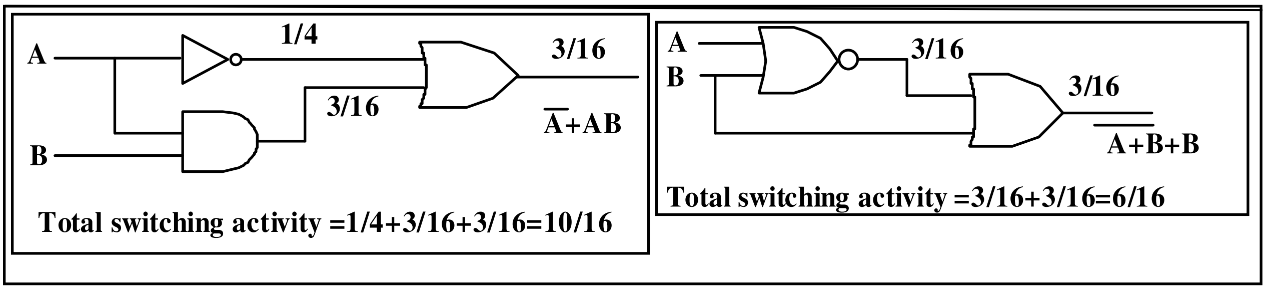

A
B
1/4
3/16
3/16
Ā+AB
Total switching activity =1/4+3/16+3/16=10/16
A
B
3/16
3/16
A+B+B
Total switching activity =3/16+3/16=6/16

Figure 10: Switching Activity for $\overline{A}+AB$ and $(\overline{A+B})+B$

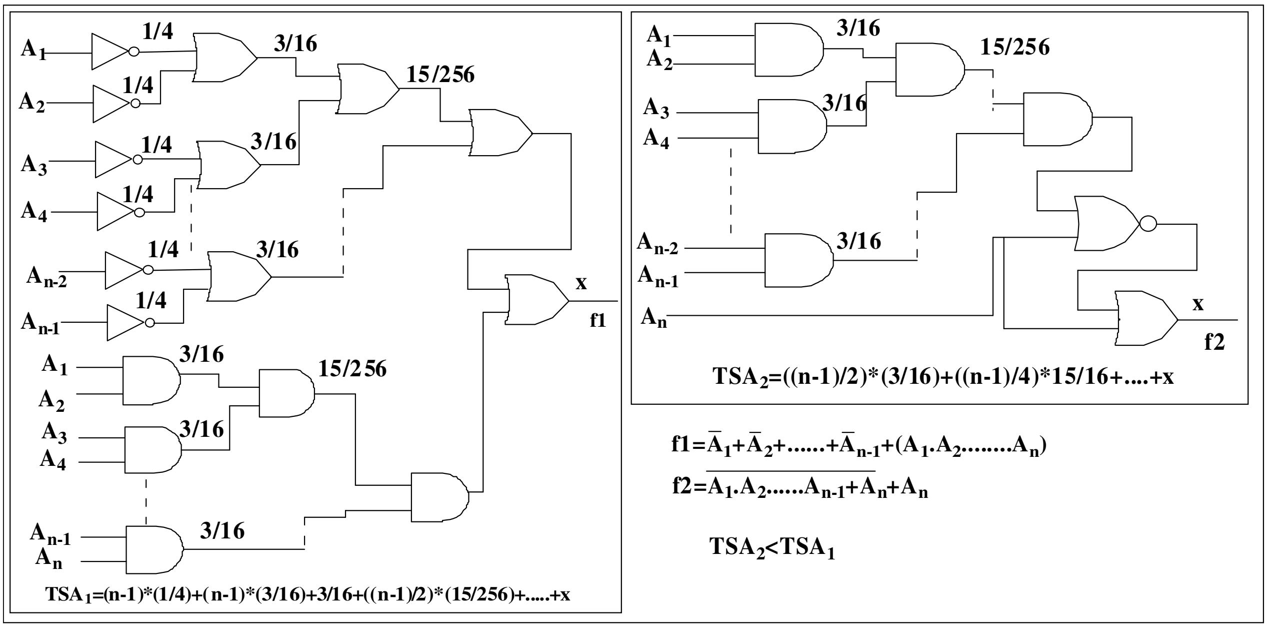


Figure 11: Switching Activity for Observation 11

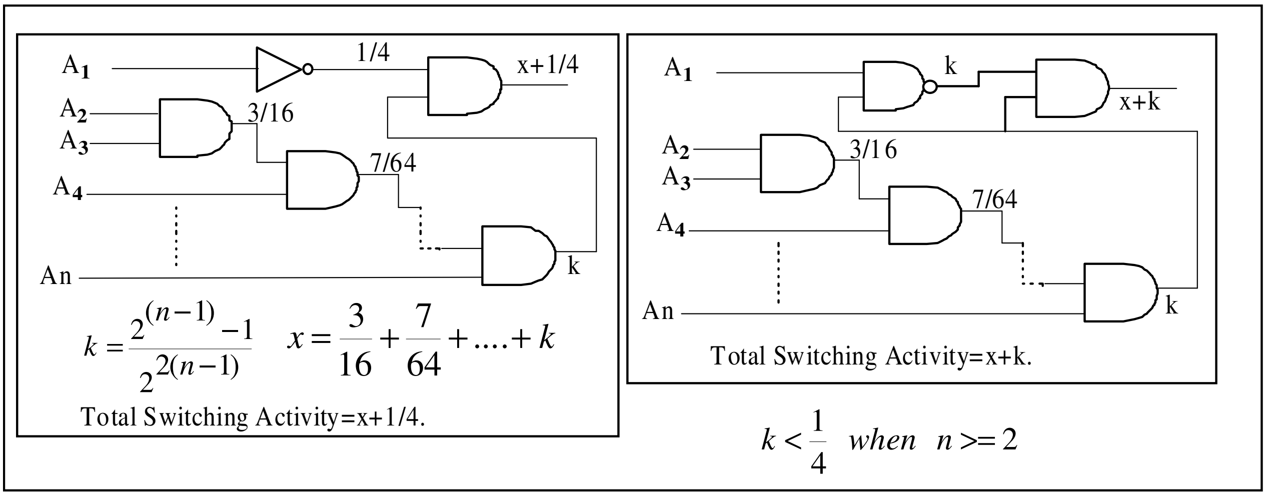

A1
1/4
x+1/4
A2
A3
3/16
A4
7/64
An
k
$k=\frac{2^{(n-1)}-1}{2^{2(n-1)}}$
$x=\frac{3}{16}+\frac{7}{64}+....+k$
Total Switching Activity=x+1/4.
A1
k
x+k
A2
A3
3/16
A4
7/64
An
k
Total Switching Activity=x+k.
$k<\frac{1}{4}$ when $n>=2$

Figure 12: Figure for Observation 13

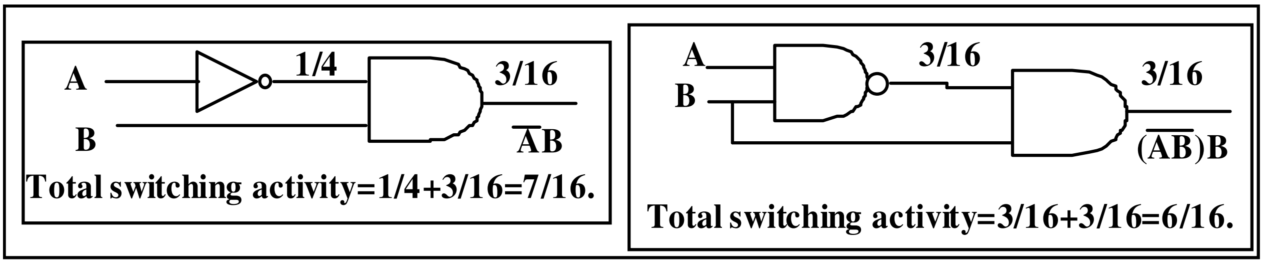

A
1/4
3/16
B
$\overline{A}B$
Total switching activity=1/4+3/16=7/16.
A
B
3/16
3/16
$(\overline{AB})B$
Total switching activity=3/16+3/16=6/16.

Figure 13: Switching Activity for $\overline{A}B$ and $(\overline{AB})B$

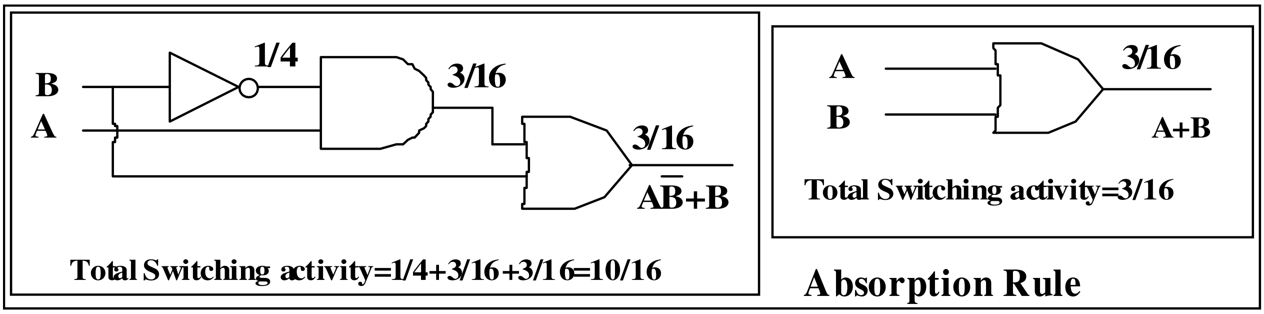

B
A
1/4
3/16
3/16
$A\overline{B}$+B
Total Switching activity=1/4+3/16+3/16=10/16
A
B
3/16
A+B
Total Switching activity=3/16
Absorption Rule

Figure 14: Circuit Diagram and switching activity for $A\overline{B}+B$ and $A+B$

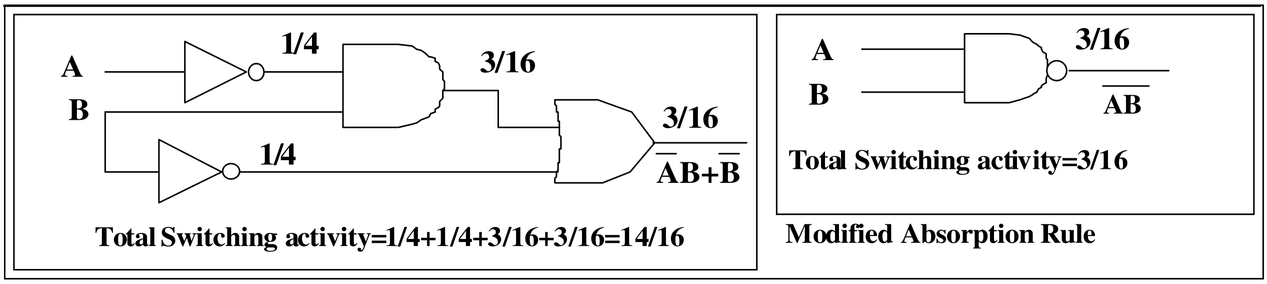

A
B
1/4
3/16
1/4
3/16
$\overline{A}B+\overline{B}$
Total Switching activity=1/4+1/4+3/16+3/16=14/16
A
B
3/16
$\overline{AB}$
Total Switching activity=3/16
Modified Absorption Rule

Figure 15: Circuit Diagram and switching activity for $\overline{A}B+\overline{B}$ and $\overline{AB}$

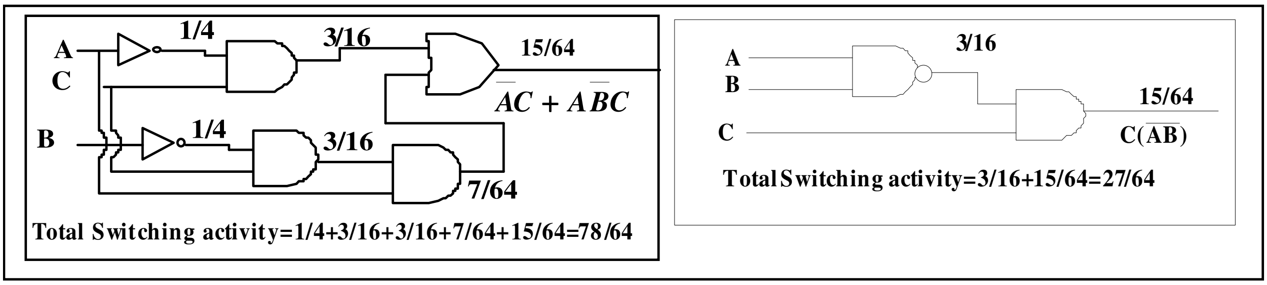
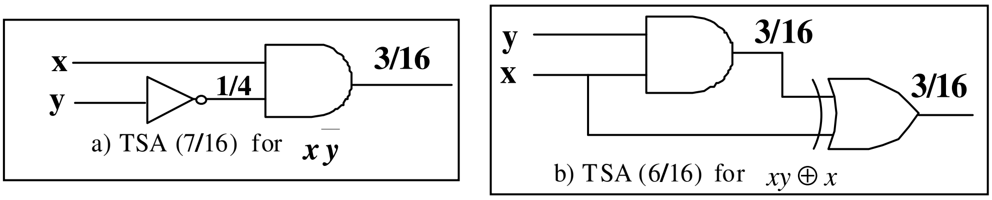

A
C
B
1/4
3/16
15/64
$\overline{A}C + A\overline{B}C$
1/4
3/16
7/64
Total Switching activity=1/4+3/16+3/16+7/64+15/64=78/64
A
B
C
3/16
15/64
C($\overline{AB}$)
Total Switching activity=3/16+15/64=27/64

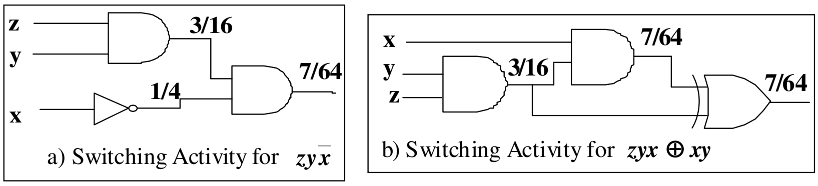
Figure 16: Switching Activity for $\overline{A}C + A\overline{B}C$ and $C(\overline{AB})$


x
y
1/4
3/16
a) TSA (7/16) for $x\overline{y}$
y
x
3/16
3/16
b) TSA (6/16) for $xy \oplus x$

x	y	$x\overline{y}$	$xy \oplus y$
0	0	0	0
0	1	0	0
1	0	1	1
1	1	0	0

c) Truth Table

Figure 17: Switching Activity a) $x\overline{y}$ b) $xy \oplus y$ c) Truth Table for $x\overline{y}$ and $xy \oplus y$


z
y
x
3/16
1/4
7/64
a) Switching Activity for $zy\overline{x}$
x
y
z
3/16
7/64
7/64
b) Switching Activity for $zyx \oplus xy$

x	y	z	$\overline{x}$	yz	$zy\overline{x}$	zyx	$zyx \oplus zy$
0	0	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	0

c) Truth Table

Figure 18: Switching Activity for a) $zy\overline{x}$ b) $zyx \oplus zy$ c) Corresponding Truth Table

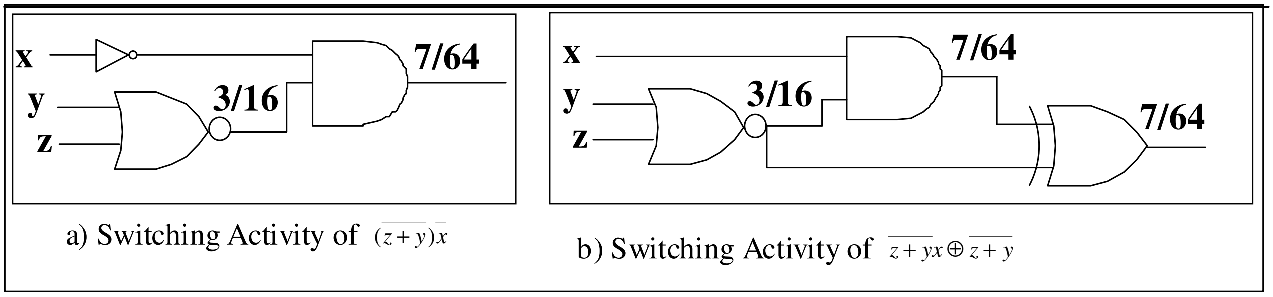

x
y
z
3/16
7/64
a) Switching Activity of $\overline{(z+y)}\overline{x}$
x
y
z
3/16
7/64
7/64
b) Switching Activity of $\overline{z+y}x \oplus \overline{z+y}$

Figure 19: Switching Activity for a) $\overline{(z+y)}\overline{x}$ b) $\overline{(z+y)}\overline{x} \oplus \overline{(z+y)}$

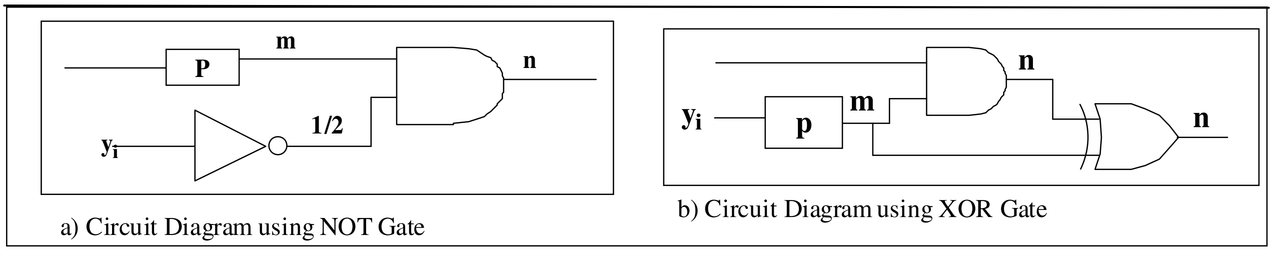

P
m
n
1/2
y_i
a) Circuit Diagram using NOT Gate
n
y_i
p
m
n
b) Circuit Diagram using XOR Gate

Figure 20: Switching Activity for product term *f* containing odd number complimented literals of literals

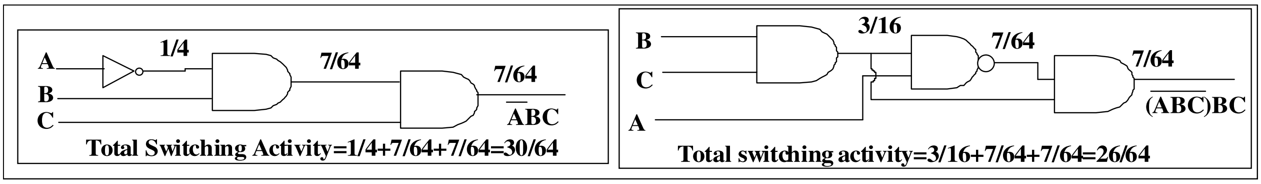

A
B
C
1/4
7/64
7/64
$\overline{A}BC$
Total Switching Activity=1/4+7/64+7/64=30/64
B
C
A
3/16
7/64
7/64
$(\overline{ABC})BC$
Total switching activity=3/16+7/64+7/64=26/64

Figure 21: Switching Activity for $\overline{A}BC$ and $(\overline{ABC})BC$

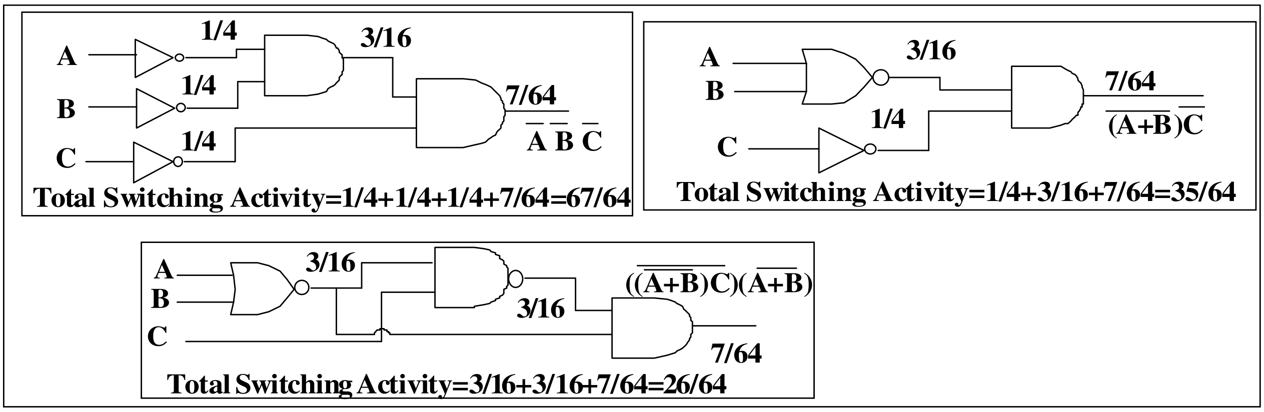

A
B
C
1/4
1/4
1/4
3/16
7/64
$\overline{A}\,\overline{B}\,\overline{C}$
Total Switching Activity=1/4+1/4+1/4+7/64=67/64
A
B
C
3/16
1/4
7/64
$\overline{(A+B)}\overline{C}$
Total Switching Activity=1/4+3/16+7/64=35/64
A
B
C
3/16
3/16
$(\overline{(\overline{A+B})C})(\overline{A+B})$
7/64
Total Switching Activity=3/16+3/16+7/64=26/64

Figure 22: Switching Activity for $\overline{A}\,\overline{B}\,\overline{C}$, $(\overline{A+B})\overline{C}$ and $\overline{(\overline{A+B})C}(\overline{A+B})$

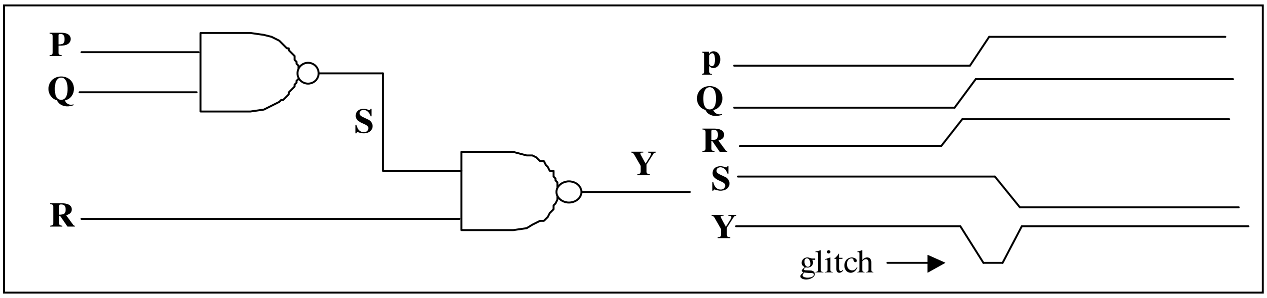


Figure 23: Glitching due to delay

A_1	B	$\overline{B}$	$\overline{A_1+B}$	$A_1+\overline{B}$	$A_1+\overline{A_1+B}$
0	0	1	1	1	1
0	1	0	0	0	0
1	0	1	0	1	1
1	1	0	0	1	1

Table 1: Truth table for $A_1+\overline{B}$ and $A_1+\overline{A_1+B}$

A	B	$\bar{A}$	$\overline{A+B}$	AB	$\bar{A}+AB$	$\overline{A+B}+B$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	1	0	0	1	1	1

Table 2: Truth table for $\bar{A}+AB$ and $\overline{A+B}+B$

z	y	x	$\overline{z+y}$	$\bar{x}$	$(\overline{z+y})\bar{x}$	$\overline{(z+y)}x$	$(\overline{z+y})x \oplus (\overline{z+y})$
0	0	0	1	1	1	0	1
0	0	1	1	0	0	1	0
0	1	0	0	1	0	0	0
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0
1	1	1	0	0	0	0	0

Table 3: Truth Table for Figure 19

Circuits	TSA$_{Conv}$	TSA$_{our}$	% Reduction of TSA	Dynamic Power$_{conv}$	Dynamic Power$_{Our}$
3 : 8 Decoder	176/64	116/64	34%	15.7121 uW	14.3159 uW
4 : 1 MUX	108/64	92/64	14.81%	7.3628 uW	5.2588 uW
Half Adder	7/16	7/16	0%	5.1288 uW	5.1288 UW
Full Adder	99/64	87/64	12.12%	17.8317 uW	11.1890 uW
Half Subtractor	11/16	10/16	9.90 %	6.5248 uW	5.8172 UW
Full Subtractor	29/16	23/16	20.68%	15.3178 uW	14.7618 uW
2 Bit Comparator	1526/256	1226/256	19.65%	18.3713 uW	15.4985 uW
Priority Encoder	57987/16384	47619/16384	17.87%	14.0391 uW	9.0759 uW

Table 4: Comparison of Switching Activity and dynamic power of different combinational circuits