

Switching Activity Minimization in Combinational Logic Design

R. V. Menon, S. Chennupati, N. K. Samala, D. Radhakrishnan and B. Izadi

Department of Electrical and Computer Engineering

State University of New York, New Paltz, New York 12561

{menon96, chennu07, samala49}@newpaltz.edu; {damu, bai}@enr.newpaltz.edu

Abstract: *In this paper we focus on the reduction of switching activity in combinational logic circuits. An algorithmic approach using k-map has been proposed which modifies the normal optimal solution obtained from k-map to reduce its switching activity. More than 10% reduction in switching activity has been observed using our method. The final solution gives a good trade off between cost and power consumption.*

Keywords: Switching activity, multi-level implementation, implicant, implicate, low-power, CMOS circuit.

1. Introduction

The continuing decrease in feature size and the corresponding increase in chip density and clock frequency of VLSI circuits have given rise to thermal concerns. Also longer battery life has become vital for battery powered portable systems, such as notepad computers, personal digital assistants and mobile telephones. Thus the need for low power chip design has gained increasing significance.

Managing the power of an Integrated Circuit (IC) design has become a major concern of IC designers. Many researchers have proposed various power estimation and optimization techniques and developed library modules and synthesis tools for the design of low power circuits. Power reduction is needed at different stages in the design process. Ideally, one would like power reduction very early on, such as, when only a high level (behavioral) description of the design is available. Supply voltage reduction is a design level issue that has been addressed by various researchers. While it is highly desirable, supply voltage reduction is ultimately a trade off between speed and power. Hence, it is also important to develop low-level power reduction techniques at the gate or switch-level. Consequently low-level power estimation and reduction problem has become the focus of much research and many estimation and reduction techniques at this level have been proposed [6].

Power consumption in a CMOS circuit is given by the following expression

$$P = \sum_i V_{DD} V_{swing} C_{load} \alpha_{0 \rightarrow 1} f + V_{DD} \sum_i I_{sc} + V_{DD} I_i \quad (1)$$

The three terms in the order written represent the dynamic, short circuit and leakage power respectively of the circuit [4,5]. C_{load} is the load capacitance at node i , V_{swing} is the voltage swing at node i which is same as V_{DD} except in logic circuits implemented using pass logic [3]. $\alpha_{0 \rightarrow 1}$ is the power consuming switching activity at node i , and f is the frequency of operation of the circuit. I_{sc} is the short circuit current and I_i is the leakage current.

The dynamic component of power usually accounts for about 70% of the power consumed in a combinational circuit. Hence methods aimed at minimizing this term will have the highest impact on power reduction. This can be achieved by lowering any one or more of the parameters

like V_{DD} , V_{swing} , C_{load} , $\alpha_{0 \rightarrow 1}$ or f . At the logic design level $\alpha_{0 \rightarrow 1}$ is the principal parameter for power reduction and in this paper we pursue methods for reducing this parameter.

The paper is organized as follows: In Section 2 we discuss previous work done in the area of switching activity reduction, in Section 3 important design issues for switching activity reduction are summarized. In Section 4 we propose the algorithm for combinational logic design with reduced switching activity. Section 5 provides a comparison of results.

2. Previous Work

This section looks at some of the previous work in the area of switching activity reduction. The estimation and minimization of switching activity in combinational circuits by considering both $1 \rightarrow 0$ and $0 \rightarrow 1$ transitions were proposed in [1]. They defined the switching activity for a logic gate as the total probable $1 \rightarrow 0$ and $0 \rightarrow 1$ transitions occurring at the output of the gate. By using their definition the switching activity for a logic gate can be evaluated by the following expression,

$$A_T = \sum_i A_i = \sum_i 2 |F_i| |R_i|$$

where A_T is the total switching activity of the circuit, and A_i is the activity at node i . F_i represents the set of minterms of the logic function realized at node i and R_i represents the set of maxterms. $|F_i|$ and $|R_i|$ represent the cardinality of the sets F_i and R_i respectively. For example, the switching activity of a two input NAND gate whose $|F| = 3$ and $|R| = 1$, evaluates to 6 using the above expression. The switching activity at individual nodes depends on the F set and R set of the output at that node.

From the above definition it may be noted that the switching activity α is maximum for inverters. Therefore the switching activity reduction techniques proposed in [1] mainly aim at eliminating input inverters. The basic approach is to group the negated terms in a three or more variable implicate and applying De-Morgan's laws for product negation that can be implemented using a NAND gate. For example, $W + \bar{X} + \bar{Y}$ can be transformed into $W + \overline{XY}$. A dual to the above method was also proposed in [1] that on implementation eliminates inverters with NOR gates. For example, $W \bar{X} \bar{Y}$ can be transformed to $W(\bar{X} + \bar{Y})$. Three or more variable implicates with single inversion can be similarly transformed using De-Morgan's laws as $W + X + \bar{Y} = \overline{(\overline{W+X})Y}$. A similar operation can be performed on its dual.

The switching activity estimation by assigning an integer value as defined in [1] makes it almost impossible for comparing different circuits since there is no reference metric on which comparisons can be made. Furthermore, there is no upper bound to the switching activity when using the definition provided by [1]. Even though [1] provides a method for reducing the switching activity, it does not provide a formal procedure for the design of a combinational circuit with reduced switching activity. It also does not offer ways to completely eliminate input inverters.

A disjoint implicant approach for reducing the switching activity was proposed in [2] that aimed at increasing the number of inputs to a gate. They partitioned the prime implicants (implicates) of a minimal sum-of-products (product-of-sums) expression that share minterms (maxterms) into disjoint implicants (implicates). The partitioning is done such that no new complementary variables are introduced. The partitioning increases the number of variables in an implicant (implicate), thereby increasing the number of inputs to a gate, which directly reduces switching activity. The methods proposed in [1] for eliminating the input inverters are then applied to the resulting function. The following example may be used to illustrate their technique.

Example 1: Consider a function $F = WZ + \bar{X}Z + \bar{W}X\bar{Y}$. Here we observe that the two, two variable prime implicants share minterms. One of the prime implicants can be partitioned to make these implicants disjoint. According to the procedure defined in [2] priority for partitioning is given to two variable implicants with one variable complemented. This gives us the function $F = WZ + \bar{W}\bar{X}Z + \bar{W}X\bar{Y}$. The last two terms with complementary variables can then be converted to $(\bar{W} + \bar{X})Z + (\bar{W} + \bar{Y})X$, thereby eliminating the input inverters. \square

The switching activity definition given in [1] was modified in [2] to satisfy the classical probabilistic approach that limits switching activity to a maximum value of 1. The switching activity at a node is defined as the sum of the transition probabilities of the node switching from $0 \rightarrow 1$ and $1 \rightarrow 0$, where the transition probability of the node is given as the product of the probabilities of that node staying in 0 state as well as in 1 state. Let P_0 and P_1 be the probability of a zero and one occurring at the output respectively. Now, $P_0 = \frac{|R|}{|R| + |F|}$ and $P_1 = \frac{|F|}{|R| + |F|}$. For

a two input NAND gate with uncorrelated and uniformly distributed inputs, $P_0 = 1/4$ and $P_1 = 3/4$. The transition probability $P_{0 \rightarrow 1} = (1/4)(3/4) = 3/16$. Similarly, $P_{1 \rightarrow 0} = (3/4)(1/4) = 3/16$. Since power is drawn from the battery source only when the output changes from 0 to 1, the discussion on switching activity in [2] accounted only the power consuming transition of $0 \rightarrow 1$.

The procedure proposed in [2] is found to have certain problems associated with it. It does not take into consideration the cases where a circuit with the minimal switching activity is achieved directly from the multi - level implementation of the minimal sum-of-products or product-of-sums function itself. This is illustrated by the following example.

Example 2: Consider the function in Figure 1(a), the minimal sum-of-products expression of which is given by $F = \bar{W}\bar{X} + \bar{Y}Z$. This directly implements to Figure 1(b) using the De-Morgan's transformation, which has the minimum switching activity for this function. The same function when reduced following [2] will lead to the K-map shown in Figure 1(c), the implementation of which is shown in Figure 1(d). It may be noted that the switching activity of the implementation shown in Figure 1(d) is much greater than the modified minimal sum-of-products expression implemented in Figure 1(b). \square

The procedure given in [2] explicitly prohibits the introduction of a new complemented variable. But it may be noted that this restriction can be lifted easily since complemented variables can be eliminated using the NAND-NOR or NOR-NAND implementation. Also, in cases of many implicants (implicates) sharing minterms (maxterms), the method proposed in [2] will lead to experimenting with different partitions before finding the right solution.

3. Factors Affecting Switching Activity

In the last section we discussed previous research that used different methods for switching activity reduction, like eliminating input inverters or increasing the number of inputs to a gate. In this section we will look at how these factors reduce switching activity.

The switching activity is a function of the number of 1's and 0's in the function. Switching activity is minimum, when the difference between $|F|$ and $|R|$ of the function is maximum, and maximum when $|F| = |R|$. A k variable function will have 2^k total possible input combinations. Let x represent the cardinality of the F set in the above function. Then the cardinality of R set in a completely defined function will be $(2^k - x)$. Therefore, $P_0 = (2^k - x) / 2^k$ and $P_1 = x / 2^k$.

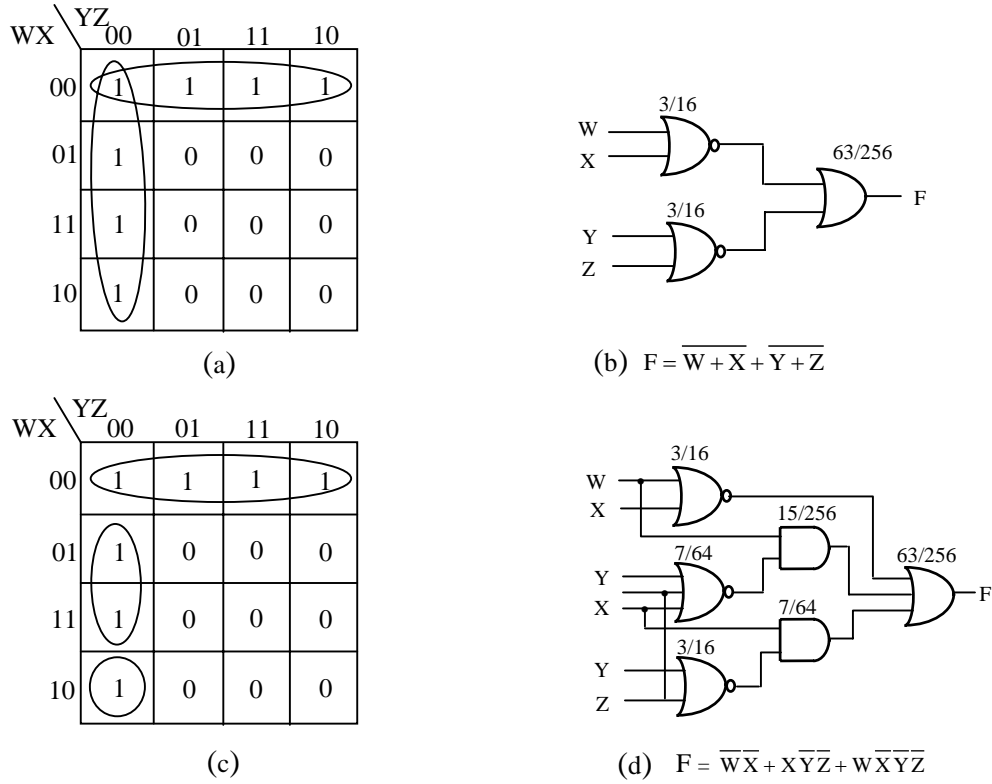


Figure1. Example showing increased switching activity using disjoint approach compared to optimal sum-of-products expression

The power consuming switching activity is then given as $P_{0 \rightarrow 1} = x(2^k - x)/2^{2k}$.

$$\frac{d}{dx}(P_{0 \rightarrow 1}) = \frac{d}{dx} \left(\frac{x(2^k - x)}{2^{2k}} \right) = \left(\frac{2^k - 2x}{2^{2k}} \right) \quad (2)$$

Equating the result in (2) to 0 and solving for x we get

$$\left(\frac{2^k - 2x}{2^{2k}} \right) = 0 \Rightarrow x = \frac{2^k}{2} \quad (3)$$

Hence, the switching activity is maximum when the number of 1s and 0s of the function are equal. Since inverter has equal number of 1s and 0s in its function it has the highest switching activity. Hence, an important design consideration for reducing switching activity is to avoid input inverters.

Switching activity of a gate is also a function of the number of inputs to the gate. A gate with fewer inputs will have higher switching activity compared to a gate with more inputs. An exception is XOR or XNOR gate which has a switching activity of 1/2 irrespective of the number of inputs. Consider an AND/OR/NAND/NOR gate with ‘i’ number of inputs. The F set or the R

set of the gate will either have 1 or $2^i - 1$ elements. Hence $P_0 = \frac{1}{2^i}$ or $\frac{2^i - 1}{2^i}$ and $P_1 = \frac{2^i - 1}{2^i}$ or $\frac{1}{2^i}$. The power consuming switching probability is $P_{0 \rightarrow 1} = \frac{2^i - 1}{2^{2i}}$. This can be

expressed as $P_{0 \rightarrow 1} = \left(\frac{1}{2^i} - \frac{1}{2^{2i}} \right)$. We observe that $P_{0 \rightarrow 1}$ is a function of the number of inputs to the gate; it decreases when the number of inputs increase and vice versa. Hence increasing the number of inputs to a gate is also an important design consideration for combinational logic design for reduced switching activity.

4. Design for Minimal Switching Activity

In this section we develop a method for designing a combinational logic circuit with the objective of reducing its switching activity. The primary goal here is to eliminate input inverters. To achieve this goal we consider all possible types of implicants in an expression.

Consider an n-variable minimal sum-of-products or product-of-sums function. The implicants (implicates) with no negated variables or with all the variables negated can be retained in the function, since they represent minimal switching activity. If two or more implicants in the function have common non-negated variables along with a single different negated variable, then they can be combined for a multilevel implementation using the NAND/NOR approach. For example, $W \bar{X} + W \bar{Y} = W (\bar{X} + \bar{Y}) = W (\overline{XY})$. Terms with three or more variables with single or more inversions can be implemented by eliminating inverters using a multi-level approach. Two variable terms that have a single inversion and do not fall in the earlier category need to be addressed separately since methods mentioned in [1] do not eliminate inversion in this case. Such implicants can be regrouped to introduce an additional variable. Our algorithm includes all such cases into account and is given next.

Algorithm:

1. Obtain a minimal sum-of-products (f_{SOP}), and a product-of-sums (f_{POS}) expressions for the given function f .
2. If either of the functions, f_{SOP} or f_{POS} , does not contain a two variable term with a complementary variable then that function represents an optimal solution. If both functions satisfy the above condition, choose the best among them after comparing the switching activity of each of the functional representations. (The calculation of switching activity is done on a multi-level implementation of the respective functions.) Exit the algorithm.
3. Else, modify the group in the k-map that represents the two variable term (with one variable appearing in complementary form) so that additional variable(s) may be added to the expression, provided that the total number of groups in the K-map is kept the same. The complementary variable then can be removed using a multi-level implementation. The choice of the final f_{SOP} or f_{POS} is made after calculating the switching activities of the two representations. □

Example 3: Consider a four variable function whose K-map is shown in Figure 2(a). The minimal sum-of-products and product-of-sums expressions are: $f_{SOP} = \bar{W} \bar{Y} + X Z + \bar{Y} Z + \bar{W} X$, and $f_{POS} = (\bar{W} + Z) (X + \bar{Y})$ respectively. A direct implementation for f_{SOP} is shown in Figure 2(b). It may be noted that f_{POS} can be easily converted to the above f_{SOP} .

Now applying the above algorithm, we see that f_{SOP} falls into Step 3, since the last two prime implicants of f_{SOP} are two variable terms with a single complementary variable. So, we modify those two prime implicants such that each of them now groups two minterms (as opposed to the original four) as shown in the K-map in Figure 3(a), giving us a modified sum-of-products expression $f_{SOP} = \bar{W} \bar{Y} + X Z + \bar{X} \bar{Y} Z + \bar{W} X \bar{Z}$. Here, it should be noted that while

regrouping, the number of groups in the K-map is kept the same. A three level implementation is shown in Figure 3(b).

A further reduction in switching activity is possible, if the implicants are further modified as single minterms as shown in Figure 4(a). Then, $f_{SOP} = \overline{W} \overline{Y} + X Z + W \overline{X} \overline{Y} Z + \overline{W} X Y \overline{Z}$, and a multilevel implementation is shown in Figure 4(b). Though there is a reduction in switching activity in this case, since two, two-input gates have been converted into three-input gates, the overall power reduction is still a question due to additional transistors being added. A comparison between the switching activities of the implementations in Figures 2(b), 3(b) and 4(b) are shown in Table 1. In the table the vectors X_1 , X_2 and X_3 represent the switching activities of the corresponding figures in the increasing order of their levels starting with the inverter. In vector X_1 the first two values represent the switching activities of the inverters; correspondingly vectors X_2 and X_3 show that two-input and three-input gates have replaced them. More than 20% saving is observed in vector X_2 and more than 40% in case of vector X_3 , due to the removal of both the inverters from the input. \square

A large number of combinational circuits were redesigned using our algorithm. A good majority of them resulted in circuits with 10-20% reduction in switching activity.

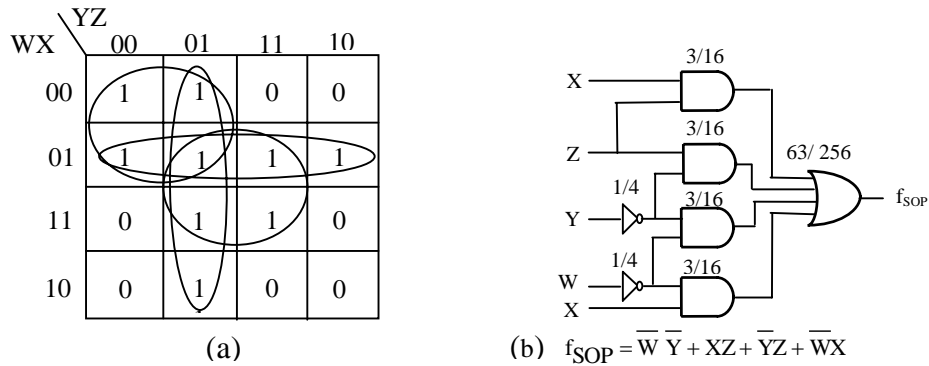


Figure 2. Minimal sum-of-products implementation for Example 3

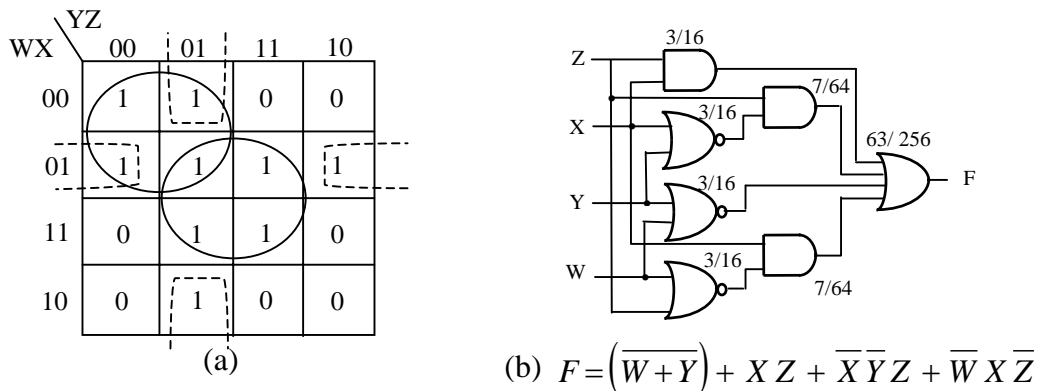


Figure 3. Modified K-map grouping of Figure 2 for reduced switching activity

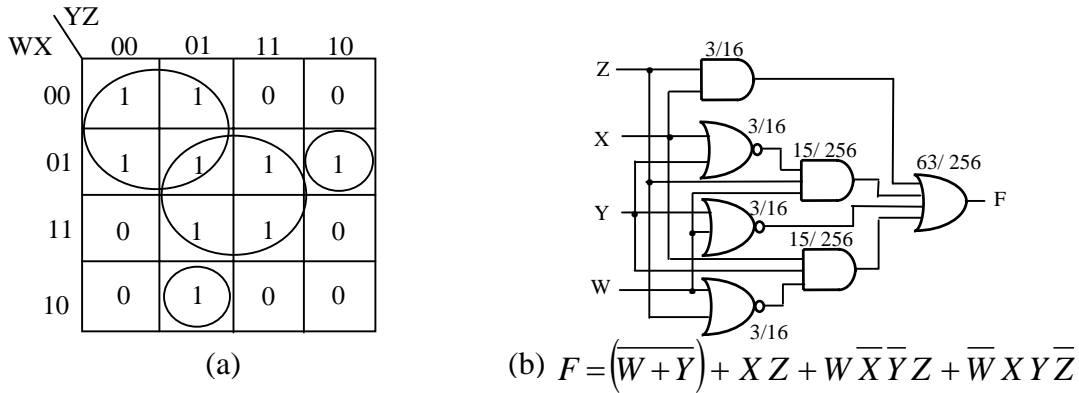


Figure 4. K-map grouping of Figure 2 for the lowest switching activity

Table 1. Comparison of switching activities

	Switching activity of different implementations
Fig. 2(b)	$X_1 = [1/4, 1/4, 3/16, 3/16, 3/16, 3/16, 63/256]$
Fig. 3(b)	$X_2 = [7/64, 7/64, 3/16, 3/16, 3/16, 3/16, 63/256]$
Fig. 4(b)	$X_3 = [15/256, 15/256, 3/16, 3/16, 3/16, 3/16, 63/256]$

5. Conclusion

An algorithmic approach for reducing the switching activity in combinational logic circuits is discussed in this paper. Logic circuits designed using our approach were able to remove input inverters completely, providing for more than 10% reduction in switching activity. Example circuits designed using the proposed algorithm have shown good results. The algorithmic technique discussed in this paper enables an easy integration into modern synthesis tools.

6. References

- [1] I. Brzozowski and A. Kos, "Minimization of Power Consumption in Digital Integrated Circuits by Reduction of Switching Activity," *25th Euromicro Conference*, vol. 1, September 1999.
- [2] R. V. Menon, S. Chennupati, N. K. Samala, D. Radhakrishnan and B. Izadi, "Power Optimized Combinational Logic Design," *Proceedings of the International Conference on Embedded Systems and Applications*, pp. 223 - 227, June 2003.
- [3] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimzu, "A 3.8-ns CMOS 16 x 16-b Multiplier using Complementary Pass-transistor Logic," *IEEE J. Solid-State Circuits*, 25, pp. 388-395, 1990.
- [4] N.Weste and K.Eshraghian, *Principles of CMOS VLSI Design, A Systems Perspective*. Massachusetts: Addison-Wesley, 1993.
- [5] H. J. M. Veendrick, "Short-circuit Dissipation of Static CMOS Circuitry and its Impact on the Design of the Buffer Circuits," *IEEE J. Solid-State Circuits*, SC-19, pp. 468-473, 1984.
- [6] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. of VLSI Systems*, vol. 2, no. 4, pp. 446-455, December 1994.