Power Optimized 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

Abstract

In this paper we address the problem of minimization of power consumption in combinational circuits by minimizing the number of switching transitions at the output nodes of each gate. A powerful combinational logic optimization method using k-map is presented here that is based on disjoint implicants (implicates) of a function. More than 10% reduction in switching activity has been obtained with marginal increase in circuit area and delay.

1. Introduction

Low power has emerged as a major design consideration for the VLSI designer in recent years. With the ever-increasing applications in mobile communications and portable equipment there is a high demand for long running batteries. Apart from the cost considerations low power has gained significance because of thermal factors that emerge as a major concern when integrating large circuits on a single die. High temperatures not only demand adequate cooling mechanisms but also raise reliability concerns, as heat tends to exacerbate silicon failure mechanisms [6]. Hence IC design techniques should not only address the issues of area and performance, but also power. By examining the following expression for the total power consumption in CMOS circuits a better understanding of the power reduction techniques in such circuits can be gained.

$$P = \sum_i V_{DD} V_{swing} C_{load} \alpha_{0 \to 1} f + V_{DD} \sum_i I_{isc} + V_{DD} I_i$$

The first term represents the switching component of power, 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 [2]. $\alpha_{0 \to 1}$ is the node transition activity factor or the switching activity of each node i in the circuit. The second term denotes the short circuit power and the third term represents the leakage power [3,4].

The dominant term in a well designed circuit is the switching component, and low power design thus becomes the task of minimizing V_{DD}, C_{load} and $\alpha_{0 \to 1}$ while retaining the required functionality. Power reduction in high-level design stages has been at the system architecture level, aiming at the minimization and optimization of supply voltages [5]. At the technology level the reduction of loading capacitances has not yet been fully answered and is wholly a design aspect. At the logic level, switching activity of a circuit is the prime factor for optimization. This paper therefore focuses on the switching activity aspect of power reduction techniques.

2. Switching Activity in CMOS Combinational Logic Circuits

The switching activity for a logic gate is the average rate at which its output switches from 0 to 1. Consider a two input NAND gate whose inputs are statistically independent and uniformly distributed. Let P_0 and P_1 be the probability of a zero and one occurring at the output respectively. Then $P_0 = 1/4$ and $P_1 = 3/4$. Therefore, $P_{0 \to 1} = (1/4)(3/4) = 3/16$ and $P_{1 \to 0} = (3/4)(1/4) = 3/16$, where $P_{0 \to 1}$ and $P_{1 \to 0}$ are the transition probabilities of the outputs switching from $0 \to 1$ and $1 \to 0$ respectively. But battery power consumed is only when the output switches from $0 \to 1$, so the switching activity of the two input NAND gate equals **3/16**.

A method for calculating the switching activity for combinational circuits realized using gate logic is presented in [1]. They used a definition for switching activity that is different from the classical one. According to their definition the switching activity for a logic gate refers to the total number of 0 to 1 and 1 to 0 transitions occurring at the output of the gate while all possible two pattern input sequences are applied at the input of the gate. The total switching activity for a complete circuit is expressed as: $A_T = \sum_i A_i = \sum_i 2|F_i||R_i|$, where A_i corresponds to the switching activity at the ith node, $|F_i|$ and $|R_i|$ corresponds to the cardinality of the sets F_i and R_i

respectively. F_i is a set with minterms corresponding to the 1's of the function, and R_i is a set corresponding to the 0's of the function.

A few guidelines were given in [1] to reduce the switching activity in combinational logic circuits. To start with, a minimal product-of-sums (sum-of-products) expression of the function is derived based on the relative magnitudes of the number of 0's (1's) in the truth table. The maximum switching activity has been noted to occur on input inverters. Hence, the design focus is directed to eliminate all input inverters from the circuit. This was achieved as follows: A first level NAND expression of the form $\overline{W\,\overline{Y}\,\overline{Z}}$ can be transformed into the form $\overline{W\,\overline{Y+Z}}$, thereby eliminating the input inverters. Similarly, a first level NOR gate of the form $\overline{W+\overline{Y}+\overline{Z}}$ can be transformed into $\overline{W+\overline{YZ}}$, again eliminating the input inverters.

We applied the guidelines given in [1] to many example circuits. Many example circuits that employed the methods described in [1] do not provide the circuit with minimal switching activity. The paper also lacks any formal techniques for the design of a combinational logic circuit with minimal switching activity. Furthermore, the selection of the sum-of-products (product-of-sums) expression based on the minimum number of 1's (0's) was found to increase the switching activity in certain cases. This was verified with the function $f(W,X,Y,Z) = \prod M(6,7,10,14)$, where the switching activity amounted to 400 while following the guidelines in [1], where as it was only 288 otherwise.

In this paper, we provide a formal technique for the design of combinational logic circuits for minimizing their switching activity.

3. Design of Combinational Logic Circuits with Minimum Switching Activity

In this section we develop formal procedures for the design of combinational logic circuits, with the primary objective of minimizing their switching activity. A number of definitions and lemmas are given first so that the reader can easily follow through the rest of the paper.

3.1. Definitions

Minimal disjunctive (conjunctive) switching function - This is a switching function consisting of the sum (product) of a minimum number of prime implicants (implicates) such that no prime implicant (implicate) from the sum (product) may be removed without changing the function being realized [7].

Disjoint implicants (implicates) - These are implicants (implicates) formed by disjoint sets of minterms (maxterms), i.e., any two implicants (implicates) taken pair wise have no common minterms (maxterms).

Disjoint disjunctive (conjunctive) switching function - This is a (product-of-sums) switching function consisting of only disjoint implicants (implicates). The following lemmas give the necessary conditions for reducing the switching activity of a two level logic implementation.

Lemma 1: The switching activity of a minimal implementation of a logic function f of n variables, with m prime implicants, will be reduced if an i variable prime implicant is replaced by a j variable implicant where $j > i$, such that the (j-i) additional variables introduced do not include complementary variables other than those already present in the remaining m-1 prime implicants, while implementing the same function f.

Proof: Consider an AND-OR / NAND-NAND realization of the function f. The realization with m prime implicants will have m first level AND/NAND gates and one second level OR/NAND gate. The total switching activity for the realization is a collection of the switching activities for each individual gate in the circuit. Assume that the first level k^{th} $(1 \le k \le m)$ AND/NAND gate with i inputs is replaced by a j input AND/NAND gate. Then depending on the j-i additional variables the following two cases occur.

Case 1: *The new (j-i) variables are non-complementary variables*

In this case the added literals do not introduce any input inverters into the circuit. Also, since the number of inputs in the new gate is more than that of the original gate, the switching activity of the new gate will be less than that of the original one, since the difference between the number of zeros and ones in the K-map of the new gate is increased.

Case 2: *One or more of the new (j-i) variables are complementary variables that are already present in the remaining m-1 prime implicants*

In this case also the switching activity of the new gate is less than that of the original one since the number of inputs in the new gate is more than the original gate. Since all the complementary variables are already present in the function, no additional input inverters are added to the circuit. Hence the total switching activity is reduced.

Lemma 2: The switching activity of a minimal product-of-sums implementation of a logic function f of n variables, with m prime implicates, will be reduced if an i variable prime implicate is replaced by a j variable implicate, where $j > i$ such that the (j-i) additional variables do not include complementary variables other than those already present in the remaining m-1 prime implicates, while implementing the same function f.

Proof: The proof of this lemma is similar to that given for Lemma 1 except that product-of-sums instead of is used here.

Based on the lemmas above we can modify a switching network such that the modified network consumes less power compared to the original one. To formally categorize these networks we introduce the following two additional definitions:

Low power disjunctive (conjunctive) switching function - This is a minimal disjunctive (conjunctive) switching function modified into a disjoint disjunctive (conjunctive) switching function using Lemma 1(Lemma 2).

A low power disjunctive (conjunctive) switching function has reduced switching activity compared to the minimal disjunctive (conjunctive) switching function and consumes less power. The following procedure can be used to generate a low power disjunctive switching function.

Procedure 1

1. Generate a minimal (product-of-sums) expression by following the standard k-map reduction rules.

2. If the minimal expression has two or more prime implicants of unequal sizes with shared minterms, then partition those prime implicants into disjoint implicants while satisfying the requirements outlined in Lemma 1 for reducing the switching activity. In doing so, the prime implicant with the largest cover should be kept intact. This eliminates the addition of a large number of disjoint implicants into the function realized.

3. If two or more prime implicants of the same size, each covering 2^n minterms have shared minterms, then all those prime implicants excepting one have to be partitioned into disjoint implicants while satisfying the requirements outlined in Lemma 1 for reducing the switching activity. In such a case, prime implicants, which do not introduce new complementary variables during the partition, must be chosen for partitioning. If none of them satisfies the above condition, then no partitioning is done.

Procedure 1 can also be used for generating a low power conjunctive switching function by initially starting with a minimal product-of-sums expression and then partitioning the prime implicates using Lemma 2 during steps 2 and 3. The following example illustrates the generation of a low power disjunctive switching function using Procedure 1.

Example1

Consider a four variable function f (W,X,Y,Z) = $\sum$m{1,3,4,5,9,11,13,15}. The K-map for the function is shown in Figure 1(a). The prime implicants are shown with circles around them. A minimized switching function in disjunctive form is given as: $F = \overline{W}X\overline{Y} + \overline{X}Z + WZ$, and its NAND-NAND implementation is shown in Figure 1(d). In the k-map in Figure 1(a), two prime implicants share minterms. The shared minterms are m9 and m11. Figure

1(b) shows a disjoint disjunctive minimization of the switching function, where one prime implicant is retained and the other partitioned. The function with this partitioning is given as: $F = \overline{W}X\overline{Y} + \overline{W}\overline{X}Z + WZ$. This adds a complementary variable $\overline{W}$ to one of the prime implicants. But this is agreeable according to Lemma 1. A different partitioning is shown in Figure 1(c). The corresponding switching expression is: $F = \overline{W}X\overline{Y} + \overline{X}Z + WXZ$. This leaves us with two disjoint disjunctive switching functions, both having the same switching activity. The corresponding NAND-NAND realizations are shown in Figures 1(e) and 1(f) respectively.

The logic circuit shown in Figure 1(d) uses three first-level NAND gates with switching activities of 7/64, 3/16, and 3/16. The modified circuits in Figures 1(e) and 1(f) also use three first level NAND gates, but with switching activities of 7/64, 7/64 and 3/16. The switching activities for the rest of the circuit are the same in all three of them. Thus, the modified circuits (Figures 1(e) and 1(f)) have lower switching activity when compared to the minimal disjunctive switching function (Figure 1(d)) as stated in Lemma 1. The two low power disjunctive switching functions generated using Procedure 1 can be further modified using the procedure outlined in [1] for eliminating the input inverters. In [1], the procedure was applied to the minimal sum-of-products (product-of-sums) expression to obtain a function with reduced switching activity. But, when the procedure is applied to the modified circuits discussed in this paper, a much higher reduction in switching activity is obtained. When the input inverters are removed from the minimal sum-of-products expression obtained in Example 2, the function modifies to: $\overline{\overline{W+Y}\ X\ \overline{\overline{X}Z}\ \overline{WZ}}$ as shown in Figure 2(a). Similarly, when the input inverters are eliminated from the low power disjunctive switching functions corresponding to Figures 1(e) and 1(f), the new expressions obtained are: $F_1 = \overline{\overline{W+Y.X}\ \overline{\overline{W+X}Z}\ \overline{WZ}}$ and $F_2 = \overline{\overline{W+Y}\ X\ \overline{\overline{X}Z}\ \overline{WXZ}}$ and are shown in Figures 2(b) and 2(c) respectively. It may be noted at this point that while the two low power disjunctive switching functions (Figures 1(e) and 1(f)) had the same switching activity, their modified versions (obtained by removing the input inverters) exhibited different switching activities. This suggests the need for additional requirements while applying Procedure 1. Adding a 4th step to Procedure 1 as follows will eliminate the above problem:

4. During Steps 2 and 3 of Procedure 1, while selecting the prime implicants for partitioning, priority is given to two variable prime implicants with a complementary variable.
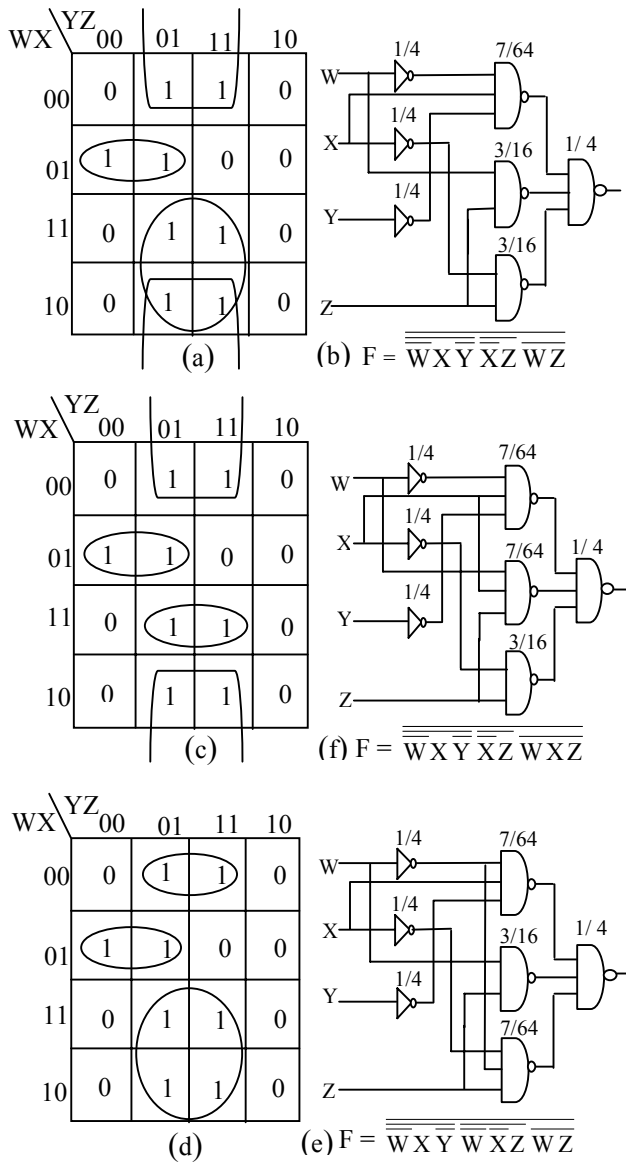
(a)

(b) $F = \overline{\overline{\overline{WX}\,\overline{Y}}\,\overline{\overline{XZ}}\,\overline{WZ}}$

(c)

(f) $F = \overline{\overline{\overline{WX}\,\overline{Y}}\,\overline{\overline{XZ}}\,\overline{WXZ}}$

(d)

(e) $F = \overline{\overline{\overline{WX}\,\overline{Y}}\,\overline{W}\,\overline{XZ}\,\overline{WZ}}$

Figure 1. Alternate sum-of-products implementations for a 4-variable function

(a) $F = \overline{\overline{\overline{W+Y}.X}\,\overline{\overline{XZ}}\,\overline{WZ}}$

(b) $F_1 = \overline{\overline{\overline{W+Y}.X}\,\overline{W+XZ}\,\overline{WZ}}$

(c) $F_2 = \overline{\overline{\overline{W+Y}.X}\,\overline{\overline{XZ}}\,\overline{WXZ}}$

Figure 2. Realizations of the 4-variable function in Example 2 modified using [1]

In Example 2, the application of Step 4 directly results in the circuit shown in Figure 2(b) that has the least switching activity. The switching activities at the outputs of each gate in the circuits of Figures 2(a) and 2(b) are tabulated in Table 1. The circuit shown in Figure 2(a) is claimed as the best circuit in terms of minimal switching activity in [1], while the circuit in Figure 2(b) is the one obtained by our method. Table 1 shows that the circuit in Figure 2(b) has lower switching activity compared to the one in Figure 2(a). More than 10% savings is demonstrated in this case.
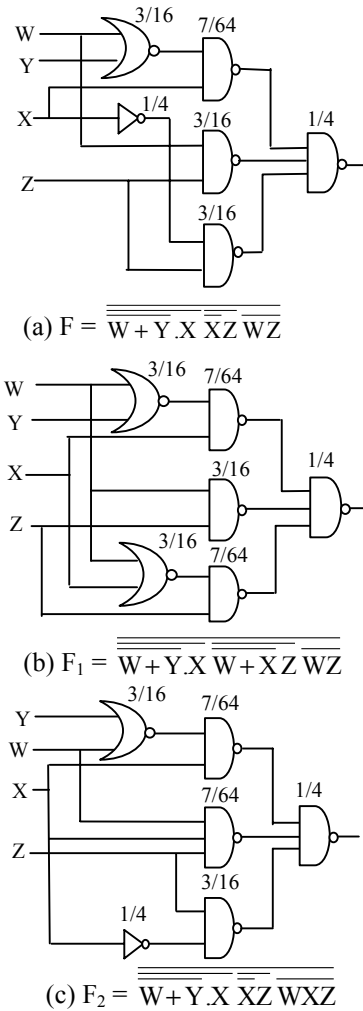
Furthermore, many example circuits designed using our method resulted in more than 10% savings in switching activity compared to the best designs reported so far.

Table 1. Comparison of switching activities

Function in Example 2		Inv	First level gate	Second level gate	Third level gate
	Fig. 2 (a)	1/4	3/16 **	3/16 3/16 7/64	1/4
Our Method	Fig. 2 (b)	**	3/16 3/16	3/16 7/64 7/64	1/4

4. Conclusions

A formal method for the design of combinational logic circuits with reduced switching activity is presented in this paper. Logic circuits designed using our method have been compared to the ones that have been claimed as the best design in [1] and are found to have 5-11% lesser switching activity. Furthermore, more than 12% reduction in switching activity has been demonstrated by our example circuit compared to the best designs reported so far. The present design is based on k-map minimization techniques and hence is limited to a maximum of six variables. By developing an algorithmic technique, the design will be able to handle more than six variables, while enabling an easy integration into modern logic synthesis tools.

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, Sept. 1999.

[2] 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.

[3] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design, a Systems Perspective.* Massachusetts: Addison-Wesley, 1993.

[4] 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

[5] P. Pant, V.K. Chatterjee, and A. De, "Simultaneous Power Supply, Threshold Voltage, and Transistor Size Optimization for Low–Power Operation of CMOS Circuits," *IEEE Trans. VLSI Systems*, vol. 6, no. 4, Dec. 1998.

[6] C. Small, "Shrinking Devices Put the Squeez on System Packaging," *EDN*, vol. 39, no. 4, pp. 41-46, Feb. 1994.

[7] D.D. Givone, *Introduction to Switching Circuit Theory.* New York: McGraw-Hill, 1970.