

# A Method for Evaluating Upper Bound of Simultaneous Switching Gates Using Circuit Partition

Kai Zhang, Tsuyoshi Shinogi, Haruhiko Takase and Terumine Hayashi

Department of Electrical and Electronic Engineering, Faculty of Engineering, Mie University

**Abstract:** This paper presents a method for evaluating an upper bound of simultaneous switching gates in combinational circuits. In this method, the original circuit is partitioned into subcircuits, and the upper bound is approximately computed as the sum of maximum switching gates for all subcircuits. In order to increase the accuracy, we adopted an evaluation function that takes account of both the interconnections among subcircuits and the number of generated subcircuits. Experimental results for ISCAS circuits show that the method efficiently evaluates the upper bounds of switching gates.

## 1. Introduction

The concern of power dissipation and device reliability increases in proportion to the level of integration of LSI. The advent of VLSI has led to much recent work on the evaluation of power dissipation and the enhancement of reliability during the design phase.

In the last few years, there were many papers to evaluate the average power dissipation for supporting the lower power design [1]. However, excessive power dissipation in VLSI circuits may reduce the reliability of VLSI chips and peak power dissipation can have a large impact on reliability. Therefore, for high reliability of VLSI, it is essential to evaluate the maximum power dissipation and the analysis for peak power dissipation is more desirable than that for average one. For a CMOS circuit, the power dissipation is mainly due to switching activities charging and discharging capacitances at the gates in a circuit, so the theme of this paper is on evaluating the maximum number of simultaneous switching gates. Accurately evaluating the exact value of maximum simultaneous switching gates in CMOS circuits involves exhaustively searching for two consecutive binary input vector pairs to induce as many simultaneous switching gates in LSI as possible. We call this exhaustively searching "exhaustive enumeration". Unfortunately, the time complexity for this search is  $O(4^n)$ , where  $n$  is the number of primary inputs in the circuit. Therefore, one of the practical way to solve this problem is to obtain an approximation value which includes a lower bound or an upper bound for maximum number of switching gates, where lower bound means the evaluating value that is smaller than the exact value and upper bound means the evaluating value that is

larger than the exact value.

Several effective approaches for evaluating a lower bound of the maximum number of simultaneous switching gates have been proposed until now. They include the partial exhaustive enumeration method [2], the branch-and-bound method [3], the method using genetic algorithm (GA) [4], the approach based on maximum satisfiability via disjoint cover enumeration [5], the maximum power estimation for CMOS circuits using deterministic and statistic approaches [6] and the iterative improvement method [7]. However, a lower bound can not assure the maximum, so it is not safe to use lower bound for VLSI reliability. On the contrary, the upper bound assures the maximum.

To the best of our knowledge, this paper presents a method for computing an upper bound of simultaneous switching gates in combinational circuits for the first time. In this method, the original circuit is partitioned into subcircuits, and the upper bound is approximately computed as the sum of maximum switching gates for all subcircuits. In order to increase the accuracy, we adopted an evaluation function that takes account of both the interconnections among subcircuits and the number of generated subcircuits. Experimental results for ISCAS circuits show that the method efficiently evaluates the upper bounds of switching gates.

The rest of this paper is organized as follows. Section 2 indicates the origin of our research problem. Some key definitions are given in Section 3. The idea of this method is described in Section 4. We show the procedure in Section 5, and Section 6 is devoted to the description of experimental results. Finally, we conclude the paper with the summary.

## 2. Problem Formulation

In this section, we clarify the problem of evaluating the maximum power dissipation in CMOS circuits.

In CMOS circuits, the switching activity of each gate brings most of the power dissipation. The power dissipation with  $N$  gates of  $g_1, g_2, \dots, g_N$  in a circuit is given by ([2] - [7]):

$$P(V) = L * E^2 * \sum_i \{ C(g_i) * T_i(V) \} \quad (i = 1, 2, \dots, N) \quad (1)$$

$V = ((u_1, u_2, \dots, u_n), (v_1, v_2, \dots, v_n))$ ,  $u_j, v_j \in \{0, 1\}$  ( $j = 1, 2, \dots, n$ ) is an input vector pair of  $n$  primary inputs, which means that the values given to the primary inputs are changed from

$u_1, u_2, \dots, u_n$  to  $v_1, v_2, \dots, v_n$ .  $P(V)$  denotes the power dissipation,  $C(g_i)$  is the load capacitance of gate  $g_i$ ,  $E$  is the supply voltage and  $L$  is a constant, and  $T_i(V)$  means that the output of gate  $g_i$  is changed by  $V$  as switching gate, i.e., let  $T_i(V)$  be 1 if gate  $g_i$  switches, otherwise, let  $T_i(V)$  be 0.

The Eq.(1) shows that evaluating the power dissipation is equivalent to counting the weighted number of simultaneous switching gates  $\Sigma\{C(g_i) * T_i(V)\}$ . Although the load capacitance  $C(g_i)$  varies with the characteristic of gates, layout results and the other factors, it is assumed to be identical in order to make the problem simple in this paper. The procedure proposed in this paper can be easily modified by giving the weight, corresponding to the load capacitance to each gate. As  $E$  is a constant as well, the Eq.(1) can be changed to the following Eq.(2).

$$P(V) = L * \sum_i T_i(V) \quad (i=1,2,\dots,N) \quad (2)$$

Here,  $L'$  is a constant and  $\Sigma T_i(V)$  is the number of simultaneous switching gates.

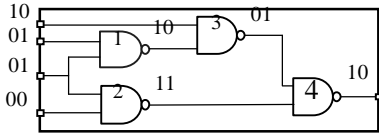


Fig.1 Simultaneous switching gates

Eq.(2) shows that evaluating the power dissipation is taken as counting the number of simultaneous switching gates  $\Sigma T_i(V)$ . Fig.1 shows 3 simultaneous switching gates. Hereafter, the simultaneous switching gates are abbreviated to “switching gates”

### 3. Definitions about Circuit Partitioning

This section gives some key definitions. Let the original circuit  $C_{orig}$  be partitioned into subcircuits  $C_i (i=1,2,\dots,k)$ .

- The Number of Interconnections ( $\#CUT(C_i, C_j)$ )

Let  $G_i$  be the set of gates in a subcircuit  $C_i$ .  $\#CUT(C_i, C_j)$  is defined as the number of interconnections between subcircuits  $C_i$  and  $C_j$ . Now, let  $G_{ij}$  be the set of all gates in  $G_i$  with at least a successor in  $G_j$ . The number of interconnections is calculated as the sum of  $|G_{ij}|$  and  $|G_{ji}|$ , where  $|S|$  denotes the number of elements of the set  $S$ .

- The Number of Interconnections among Partitions ( $\#CUT_{total}$ )

We define  $\#CUT_{total}$  as the number of all interconnections among subcircuits. That is,

$$\#CUT_{total} = \sum_{i=1}^k \sum_{j=i+1}^k \#CUT(C_i, C_j)$$

- Pseudo Primary Inputs (PPIs) for Subcircuit  $C_i$

If the source of a line  $l$  in a subcircuit  $C_i$  is a primary input or a gate outside of  $C_i$ , line  $l$  is a pseudo primary input of subcircuit  $C_i$ .

An example is shown in Fig.2.  $PI_1 - PI_8$  and  $PO_1 - PO_2$  represent the primary inputs and primary outputs in the original circuit respectively. Now suppose that the original circuit  $C_{orig}$  is partitioned into three subcircuits  $C_1, C_2$  and  $C_3$ . Then, line  $l_3$  and  $PI_{5-8}$  are PPIs of subcircuit  $C_2$ , and lines  $l_1, l_2, l_4$ , and  $l_5$  are PPIs of subcircuit  $C_3$ . Also,  $\#CUT(C_1, C_2)$ ,  $\#CUT(C_1, C_3)$  and  $\#CUT(C_2, C_3)$  are equal to 1, 1 and 2 respectively. After all, they make the value of  $\#CUT_{total}$  4.

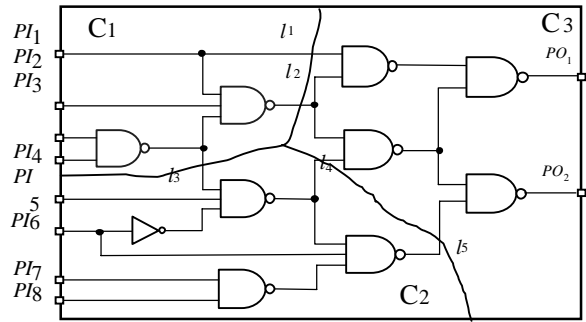


Fig.2 Example for cut definition

## 4. Idea for Evaluating Upper Bound

We present a method for evaluating an upper bound, using the circuit partition. In this section, we present the basic idea, that is, the original circuit  $C_{orig}$  is partitioned into subcircuits  $C_i (i=1,2,\dots,k)$  so as to make exhaustive enumeration for each subcircuit be possible. In order to realize this basic idea, we have to consider two problems, (1) computation time and (2) accuracy of upper bound.

### 4.1 Basic Idea

We present the basic idea for obtaining an upper bound for the number of switching gates. The original circuit is partitioned into subcircuits  $C_i (i=1,2,\dots,k)$ . The maximum number of switching gates for each  $C_i$  can be calculated by the exhaustive enumeration for PPIs, then, the upper bound for the original circuit is obtained by the sum of those maximum numbers for  $C_i (i=1,2,\dots,k)$ .

Our method ensures that we can obtain the upper bound on maximum number of switching gates. We can use the following inequality to describe our basic idea.

$$swg(C_{orig}) \leq \sum_{i=1}^K swg(C_i) \quad (3)$$

Here,  $swg(C_{orig})$  implies the exact value of switching gates for the original circuit.  $swg(C_i)$  is computed by the exhaustive enumeration of vector pair for  $PPIs$ . Because  $swg(C_i)$  is computed under the condition that each subcircuit is taken as independence for  $PPIs$  and the exhaustive enumeration is carried out for each subcircuit, so Ineq(3) holds. Since  $\sum swg(C_i)$  exists the above of exact value, it is an upper bound. In other words, the upper bound problem can be approximately solved by calculating  $swg(C_i)$  for each subcircuit  $C_i$ .

## 4.2 Consideration

In order to implement the above idea, we have to consider two problems, computation time and accuracy of upper bound.

### (1) Computation Time

For the present, the exhaustive enumeration is the only way to obtain the exact value of  $swg(C_i)$ . The number of combinations of all vector pair for  $PPIs$  in each subcircuit equals  $4^n$ . Here  $n$  is the number of pseudo primary inputs  $PPIs$ . The computation time is very long if  $n$  is large, therefore, we use some predetermined number of  $PPIs$  to restrain the number of  $PPIs$  for each subcircuit, so that the maximum number of switching gates for each subcircuit can be calculated by the exhaustive enumeration within permissible time. The constraint for number of  $PPIs$  is denoted by a parameter  $\eta$ .

### (2) Accuracy of Upper Bound

In order to increase the accuracy of upper bound to be obtained, we discuss how the accuracy of upper bound is related to the number of interconnections among subcircuits  $\#CUT_{total}$  and the number of subcircuits. In the following discussion, keep in mind that the lower evaluated number of switching gates means the more accuracy results, since our method evaluates the maximum number of switching gates as an upper bound.

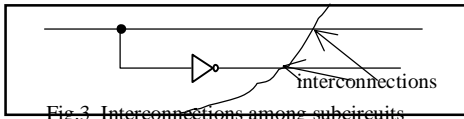


Fig.3 Interconnections among subcircuits

### ● Interconnections

In Fig.3, there exist the impossible vector pairs in the exhaustive enumeration, because of the dependence between the interconnections. In other words, the dependence makes the accuracy of upper bound be decreased if we use exhaustive enumeration. In order to increase the accuracy of upper bound, we should decrease the dependence. If the number of interconnections is decreased, the possibility that the

dependence happens can be decreased. Therefore, we use simple heuristics in circuit partitioning to make the number of interconnections among subcircuits as small as possible.

### ● Number of Subcircuits

It is more difficult for gates to switch in the large circuit than the small one. In order to increase the accuracy of upper bound, we have to make the number of gates in each subcircuit be large, that is, the number of subcircuits is small.

In next section, we present a procedure for the circuit partitioning to make not only the number of interconnections among subcircuits but also the number of subcircuits as small as possible.

## 5. Procedure of Evaluating an Upper Bound

This section presents a procedure for evaluating an upper bound using circuit partition, based on the idea described in Section 4.

### 5.1 Procedure

The original  $C_{orig}$  is bipartitioned into two circuits  $C_{T1}$  and  $C_{T2}$  as described in 5.2. If the number of pseudo primary inputs  $PPIs$  for  $C_{T1}$  or  $C_{T2}$  exceeds  $\eta$  for number of  $PPIs$ , then  $C_{T1}$  or  $C_{T2}$  continues to be bipartitioned, that is, recursively using bipartition. Like that, the original circuit is partitioned until the number of  $PPIs$  in each subcircuit doesn't exceed  $\eta$ . Then the upper bound for  $C_{orig}$  is obtained by the sum of those maximum numbers for subcircuits  $C_i (i=1,2,\dots,k)$ . Here those maximum numbers are computed by the exhaustive enumeration for  $PPIs$ .

### 5.2 Bipartition

The accuracy is increased, through (1) making the number of interconnections among subcircuits  $\#CUT_{total}$  as well as (2) the number of subcircuits as small as possible as discussed in Section 4.2(2). (1) In order to decrease  $\#CUT_{total}$ , at each bipartition, we make interconnections be small. (2) In order to decrease the number of subcircuits, i.e., we have to make the number of gates in each subcircuit be large, so we avoid the very small number of pseudo primary inputs  $PPIs$ . We avoid the number of  $PPIs$  to be small through balancing the number of  $PPIs$  at each bipartition.

#### 5.2.1 Evaluation Function for Bipartition $R(C_i, C_j)$

The circuit  $C$  is bipartitioned into  $C_i$  and  $C_j$ . We use the

following evaluation function for bipartition so as to increase the accuracy of upper bound. The small value of this function makes the number of interconnections among subcircuits  ${}^{\#}CUT_{total}$  and the number of subcircuits as small as possible at the same time.

$$R(C_i, C_j) = \alpha \times {}^{\#}CUT(C_i, C_j) + (1 - \alpha) \times |{}^{\#}PPI(C_i) - {}^{\#}PPI(C_j)|$$

$\alpha$ : a predetermined parameter,  $0 \leq \alpha \leq 1$ .

${}^{\#}CUT(C_i, C_j)$ : number of interconnections between  $C_i$  and  $C_j$ .  
 ${}^{\#}PPI(C_i)$  and  ${}^{\#}PPI(C_j)$ : number of *PPIs* for  $C_i$  and  $C_j$ .

Adjusting the first and second terms realize our purpose described in section 4.2 (2). We can describe the evaluation function by the following three points at each bipartition. (1) The first term makes the number of interconnections among subcircuits  ${}^{\#}CUT_{total}$  as small as possible through letting interconnections at each bipartition be small. (2) The second term makes the number of subcircuits be small through balancing *PPIs* at each bipartition. (3) The parameter  $\alpha$  is predetermined by experiments, so that two terms are made to be as small as possible at the same time.

We devise a procedure of bipartition based on the bipartition procedure in [8] for our purpose, through changing the evaluation function.

## 6. Experimental Results

In order to evaluate our method for evaluating the upper bound of switching gates using circuit partition, we did the experiment with the procedure described in Section 5.

We used ISCAS'85 benchmark circuits [9] for our experiment. The machine with pentium 120 MHz were used for our experiment.

### 6.1 Decision of Parameters

The main parameters for evaluating the upper bound contain  $\eta$  and  $\alpha$ . We give a constant to  $\eta$  and  $\alpha$  is decided by an experiment.

We select  $\eta$  for subcircuits as  $\eta = 10$ , since the exhaustive enumeration with  $\eta = 10$  has at most a million patterns that are acceptable in the computation time.

In order to decide the parameter  $\alpha$ , we perform an experiment with the procedure described in Section 5, in ranging  $\alpha$  from 0.0 to 1.0. Fig.4(a, b) shows that the upper bound varies with adjusting parameter  $\alpha$ . The lower evaluated number of switching gates means the more accuracy results, since our method evaluates the

maximum number of switching gates as an upper bound. The ratios of "diff" to  ${}^{\#}gates$  are 10% and 14% for c2670 and c5315 respectively. Here, the "diff" means the difference between the worst and best upper bound and  ${}^{\#}gates$  means the number of gates in the circuit. Fig.5(a) shows the results that are overlapped and normalized from the results in Fig.4. The upper bound/AVE varies with adjusting parameter  $\alpha$ . Here, AVE denotes the average value of upper bounds for  $\alpha = 0.0 \sim 1.0$ . We can use some value between 0.4 and 0.7 for  $\alpha$  since it exists below the average value.

### 6.2 Discussion of Evaluation Function

Section 4.2(2) describes how the accuracy of upper bound to be obtained depends on both the number of interconnections among subcircuits  ${}^{\#}CUT_{total}$  and the number of subcircuits. In order to increase the accuracy of upper bound to be obtained, our simple heuristics makes  ${}^{\#}CUT_{total}$  and the number of subcircuits as small as possible. In this section, we verify this by the experimental results.

We perform an experiment in ranging  $\alpha$  from 0.0 to 1.0. The results are shown in Fig.5(b, c) for c2670 and c5315. Corresponding to the suitable range at  $\alpha = 0.4 \sim 0.7$  in Fig.5(a), Fig.5(b, c) shows that both  ${}^{\#}CUT_{total}$  and the number of subcircuits are balanced and have the small values simultaneously. This indicates that the experimental results are in good agreement with the discussion in Section 4.2(2).

### 6.3 Results of Upper Bound

Table.1 presents the upper bound by this proposed method for ISCAS'85 benchmark circuits. According to the discussion in Section 6.1, we use  $\eta = 10$  and  $\alpha = 0.5$  for our experiment. To the best of our knowledge, this paper presents a method for evaluating an upper bound of the maximum number of switching gates for the first time. The ratio of ( ${}^{\#}gates$  - upper bound) to  ${}^{\#}gates$  is  $\gamma$ . Table.1 also shows that  $\gamma$  has the 15% average value.

Maximum power evaluation for ensuring reliability has been done by the experience of designers until now. Our proposed method gives the safe and concrete evaluation for the maximum number of switching gates to each different circuit.

## 7. Conclusion

For the reliability for maximum power dissipation,

evaluation of the maximum number of switching gates is essential. This paper presents a novel method for evaluating an upper bound, using a circuit partition method. The original circuit is partitioned into subcircuits under the constraints for numbers of pseudo primary inputs  $PPIs$ , so that the maximum number of switching gates for every subcircuit can be calculated by the exhaustive enumeration in reasonable time. Then, the upper bound for the original circuit is calculated by the sum of maximum number for every subcircuit. In order to increase the accuracy of upper bound to be obtained, our simple heuristics makes interconnections among subcircuits  $^{*}CUT_{total}$  and the number of subcircuits as small as possible through adjusting the parameter  $\alpha$  for the evaluation function. We get that  $\gamma$  has the 15% average value. Maximum power evaluation for ensuring reliability has been done by the experience of designers until now. Our proposed method gives the safe and concrete evaluation for the maximum number of switching gates to each different circuit.

## References

- [1] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," Proc. 29th Design Automation Conference, pp.253-259, 1992.
- [2] H. Ueda and K. Kinoshita, "Evaluation of the maximum number of switching gates for CMOS circuits," Technical Report of IEICE, Japan, vol.93, no.303, pp.49-56, 1993.
- [3] H. Ueda and K. Kinoshita, "Evaluation of the maximum number of switching gates for CMOS Circuits," IEICE Trans, Inf.&Syst., vol.J78-D-I, no.3, pp.367-375, Mar.1995.
- [4] H. Ueda, A. Kiyama and K. Kinoshita, "Evaluation of the maximum number of switching gates for CMOS logic circuits using geneticalgorithm," 33rd FTC Meeting, Japan, 1995.
- [5] S. Devadas, K. Keutzer, White, "Estimation of power dissipation CMOS combinational circuits using booleaned function manipulation," IEEE Trans. on Computer-Aided Design, vol.11, no.83, pp.373-383, March1992
- [6] C.-Y. Wang and K. Roy, "Maximum power estimation for CMOS circuits using deterministic and statistic approaches," 9th International Conference on VLSI Design, pp.364-369, Jan. 1996.
- [7] K. Zhang, H. Takase, T. Hayashi and H. Kita, "An iterative improvement method for evaluating the maximum number of simultaneous switching gates for combinational circuits," ASPDAC'97, pp.107-112, 1997.
- [8] Y.-C Wen and C.-K Cheng, "Towards efficient hierarchical designs by ratio cut partitioning," ICCAD, Nov 5-9, pp.298-301, 1989.
- [9] F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translation in fortran," ISCAS'85: Special Session on ATPG and Fault Simulation, 1985.

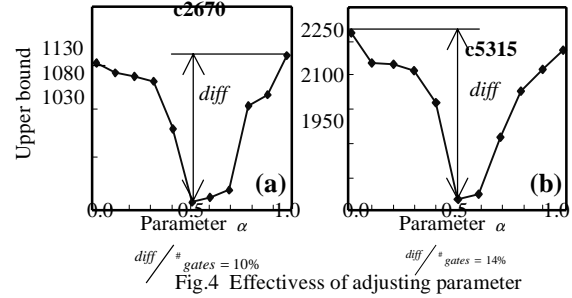


Table.1 Experimental results

| Circuits | #Inputs | #Gates | Upper Bound                  |        | $\gamma$ (%) |
|----------|---------|--------|------------------------------|--------|--------------|
|          |         |        | $\eta \leq 10, \alpha = 0.5$ |        |              |
|          |         |        | Nmax                         | CPU(s) |              |
| c880     | 60      | 383    | 347                          | 174    | 9.4          |
| c1355    | 41      | 546    | 451                          | 126    | 17.4         |
| c1908    | 33      | 880    | 750                          | 608    | 14.8         |
| c2670    | 233     | 1193   | 1008                         | 476    | 15.5         |
| c3540    | 50      | 1669   | 1476                         | 2226   | 11.6         |
| c5315    | 178     | 2307   | 1931                         | 1760   | 16.2         |
| c6288    | 32      | 2416   | 1887                         | 4014   | 21.9         |
| c7552    | 207     | 3512   | 3001                         | 4784   | 14.6         |

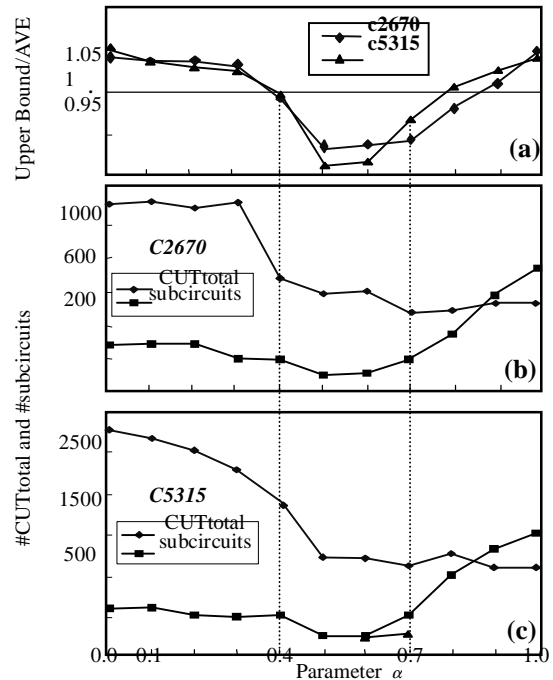


Fig.5 Effectiveness of evaluation function