

# A Cell-Based Power Estimation in CMOS Combinational Circuits

Jiing-Yuan Lin, Tai-Chien Liu and Wen-Zen Shen

Department of Electronics Engineering & Institute of Electronics,  
National Chiao Tung University, HsinChu 30050, Taiwan R.O.C.

## **Abstract**

*In this paper we present a power dissipation model considering the charging/discharging of capacitance at the gate output node as well as internal nodes, and capacitance feedthrough effect. Based on the model, a Cell-Based Power Estimation (CBPE) method is developed to estimate the power dissipation in CMOS combinational circuits. In our technique, we first construct a modified state transition graph called STGPE to model the power consumption behavior of a logic gate. Then, according to the input signal probabilities and transition densities of the logic gate, we perform an efficient method to estimate the expected activity number of each edge in the STGPE. Finally, the energy consumption of a logic gate is calculated by summing the energy consumptions of each edge in STGPE. For a set of benchmark circuits, experimental results show that the power dissipation estimated by CBPE is on average within 10-percent errors as compared to the exact SPICE simulation while the CPU time is more than two order-of-magnitudes faster.*

## **1. Introduction**

Recently, due to the advance of integrated circuit technologies, it is possible to integrate several millions of transistors into a small chip area with high performance. However, power consumption problem rises owing to the increased circuit density and speed. Higher power consumption may reduce circuit reliability, shorten the life time and thus require extra device to remove heat. Therefore, low power dissipation has become more important in the modern integrated circuits design. For a low power design environment, an accurate and efficient power estimator is necessary. A direct, simple and accurate approach for estimating power is to use SPICE simulator; however, it becomes inefficient for large circuits. Recently, some researchers have proposed several efficient power estimation methods for CMOS combinational circuits [1-5]. However, they do not know how accurate they are because the authors didn't

compare the experimental results to the exact SPICE simulation.

Most of the power estimation methods except [2] are based on a simple power dissipation model which only consider the charging/discharging of the gate output capacitance, but ignore the power consumption of gate internal nodes. In [2], Tsui et al. have considered the power contribution of gate internal nodes, but they neglected the input temporal correlation. In our experience, for considering the input temporal correlation, if the contribution of internal nodes is neglected, the power consumption will be underestimated by about 10% to 20% in average. In addition to internal node, the power consumption of coupling capacitance, which distribute between gate and source/drain terminals of MOS transistor ( $C_{gs}$ ,  $C_{gd}$ ), is usually ignored too. If the input signals have transitions, the output and internal nodes may produce signal overshoots or undershoots which would affect the power consumption. We refer to this as capacitance feedthrough effect. Obviously, the estimation may be inaccurate if we ignore the contribution of internal nodes and capacitive coupling. In this paper, we propose a power dissipation model considering not only the charging and discharging of both output and internal capacitances but also the capacitance feedthrough effect.

Based on the power dissipation model above, we propose a Cell-Based Power Estimation (CBPE) method to estimate the power dissipation of a CMOS combinational circuit. In our approach, we first construct the modified state transition graphs called STGPEs to model the power consumption behavior for each gate in cell library. Next, for a circuit network, given the input signal probabilities and transition densities, we estimate the signal probabilities and transition densities for each node by using logic simulation. For each gate, based on the input characteristics estimated above, we present an efficient method to estimate the edge activity number of the corresponding STGPE. Finally, the total energy consumption of a gate can be estimated by summing the energy consumption of each edge in the corresponding STGPE.

The rest of this paper is organized as follows. In section 2 we define the signal probability and transition density. In section 3, we propose a graph called STGPE to model the power consumption behavior of a gate. An efficient method for estimating the edge activity number of STGPE is presented in section 4. Experimental results as well as the exact SPICE simulation are presented in section 5. Finally, conclusions are drawn in section 6.

## 2. Preliminaries

Let  $\mathbf{x}(t)$ ,  $t \in (-\infty, \infty)$ , be a **strict-sense stationary (SSS)** and **mean-ergodic** process [4][7] that transitions between its 0 and 1 values at random transition times with zero rise and fall times. Such a process is called a **SSS mean-ergodic 0-1 process**. A logic signal  $\mathbf{x}(t)$  can be thought of as a sample of a SSS mean-ergodic 0-1 process  $\mathbf{x}(t)$ . In this paper, we assume that the primary input processes are SSS mean-ergodic 0-1 processes and mutually independent.

In [4], Najm has presented two continuous time probabilistic measures, equilibrium probability and transition density. In this section, we redefine these probabilistic measures from discrete time point of view. In the following, we assume the combinational circuit is a synchronous digital system controlled by a global clock with cycle time  $T_{cycle}$ . To capture the glitches exactly in general gate delay circuits, we assume  $T_{sd}$  is the smallest gate delay in the circuit, and divide the clock period into  $S = T_{cycle}/T_{sd}$  slots.

The **signal probability**[1] of an input  $x_i$  being one at a given time, denoted as  $p_i^{one}$ , is given by

$$p_i^{one} = \lim_{N \rightarrow \infty} \frac{\sum_{k=1}^{N \times S} x_i(k)}{N \times S} \quad (1)$$

where  $N$  is the total number of global clock cycles and  $x_i(k)$  is the value of input  $x_i$  during the interval of time instances  $k$  and  $k+1$ . Similarly, the probability that  $x_i$  is zero at a given time, denoted as  $p_i^{zero}$ , is  $p_i^{zero} = 1 - p_i^{one}$ .

The **transition probabilities**[1] of  $x_i$  for the transition  $0 \rightarrow 0$ ,  $0 \rightarrow 1$ ,  $1 \rightarrow 0$ , and  $1 \rightarrow 1$  can be denoted by  $p_i^{00}$ ,  $p_i^{01}$ ,  $p_i^{10}$ , and  $p_i^{11}$ , respectively, where for example  $p_i^{10}$  is defined by:

$$p_i^{10} = \lim_{N \rightarrow \infty} \frac{\sum_{k=1}^{N \times S} \overline{x_i(k)} x_i(k+1)}{N \times S} \quad (2)$$

The other transition probabilities follow similarly. It is easy to verify that

$$p_i^{00} + p_i^{01} + p_i^{10} + p_i^{11} = 1 \quad (3)$$

$$p_i^{00} + p_i^{01} = p_i^{zero} \quad (4)$$

$$p_i^{10} + p_i^{11} = p_i^{one} \quad (5)$$

There is another switching activity measure called **transition density** [4], denoted by  $D_i$  for signal  $x_i$ , which is defined as follows:

$$D_i = \lim_{N \rightarrow \infty} \frac{\sum_{k=1}^{N \times S} \overline{x_i(k)} x_i(k+1)}{N \times S} = p_i^{01} + p_i^{10} \quad (6)$$

In general, for a digital signal, the  $0 \rightarrow 1$  transition number are equal to the  $1 \rightarrow 0$  transition number. That means  $p_i^{01}$  is equal to  $p_i^{10}$  and  $D_i$  is twice that of  $p_i^{01}$  or  $p_i^{10}$ . Thus, it is well to know that only two parameters  $p_i^{one}$  and  $D_i$  are needed to determine the characteristics of signal  $x_i$ .

## 3. Our power dissipation model

There are three major sources of power dissipation in CMOS circuits: the charging/discharging of capacitance load, the direct-path short circuit current and leakage current. The first two terms are due to the *input signal transitions*, which are usually called *dynamic* power dissipation. The last term is determined by fabrication technology and is referred to as *static* power dissipation. In the following section, we shall present a power dissipation model which contains all of these sources.

### 3.1 Power dissipation of internal nodes and capacitance feedthrough effect

For a logic gate, the internal capacitance may be charged and discharged without changing the output state. For instance, in Fig. 1 the input is changing from  $10$  ( $t_1$ ) to  $01$  ( $t_2$ ) while the output remains unchanged. Although the output remains unchanged, the internal capacitance  $C_{int}$  is charged at  $t_1$  and discharged at  $t_2$ . Obviously, ignoring this effect would underestimate the power dissipation. Moreover, it is worthy of noting that the input signal transitions may directly affect the charging/discharging of output and internal nodes via capacitive coupling. An example of capacitance feedthrough effect is illustrated in Fig. 2. Because of the capacitive coupling, input transition may cause output and internal nodes overshoots or undershoots. Intuitively, the overshoots or undershoots would affect the power dissipation. Therefore, to estimate the power consumption more accurately, capacitance feedthrough effect must be considered too.

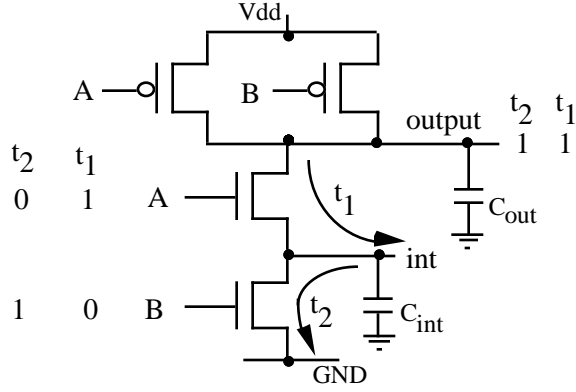


Fig. 1 Charging and Discharging of internal node in 2-input NAND gate

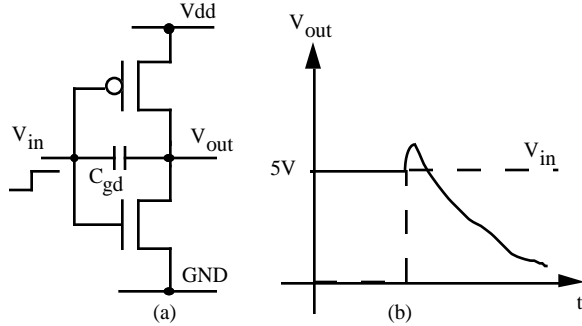


Fig. 2 An inverter with unit step input and its output voltage waveform

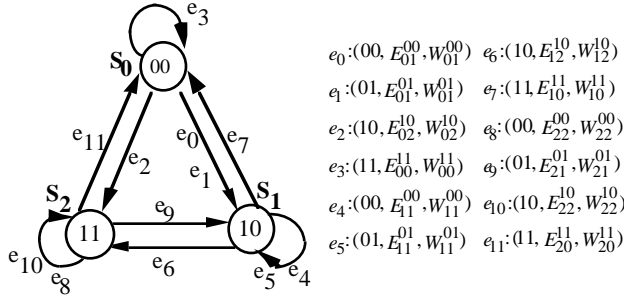


Fig. 3 STGPE of NAND2 (STGPE\_NAND2)

### 3.2 State transition graph for power estimation (STGPE)

To capture the effect of internal and coupling capacitances, we use a modified state transition graph called STGPE to model the power consumption behavior of a logic gate. Without loss of generality, we demonstrate a 2-input NAND gate, denoted by NAND2, shown in Fig. 1 as an example. Fig. 3 is the STGPE of NAND2 which depicts the power consumption behavior in Fig. 1. In Fig. 3 the first and second variables of states stand for the state of output and internal nodes, respectively. For example, state 10 represents "output"=1 and "int"=0. It is worthy to note

that state 01 does not exist because the output discharging path pass through the internal node. So, when output discharge, the internal node between output node and ground node is discharge too.

For each edge in STGPE, we label  $(i_{jk}, E_{jk}^{i_{jk}}, W_{jk}^{i_{jk}})$  to model the power consumption of state transition.  $i_{jk}$  is an *input pattern* which make a state transition from states  $S_j$  to  $S_k$ . In Fig. 3, the first and second variables of the first term in edge label represent the value of input signals A and B, respectively.  $E_{jk}^{i_{jk}}$  is the *edge activity number* of state transition from states  $S_j$  to  $S_k$  when input is  $i_{jk}$  and  $N$  sequential patterns are fed into the gate.  $W_{jk}^{i_{jk}}$  is the *energy consumption* when edge  $(i_{jk}, E_{jk}^{i_{jk}}, W_{jk}^{i_{jk}})$  is travelled. In our approach,  $W_{jk}^{i_{jk}}$  is obtained from SPICE simulation; therefore, all the three sources of power dissipation mentioned above are included in the STGPE model. For each gate in cell library, we built several STGPEs, with different  $W_{jk}^{i_{jk}}$ , for different fanout loading.

Except the energy consumption of state transition, the energy consumption due to capacitance feedthrough effect are also embedded in STGPE. For example, in Fig. 3 the input part of edge  $e_9$  is 01 and make the state transition from states  $S_2$  to  $S_1$ . When edge  $e_9$  is travelled at time  $n$ , there exists two possible inputs 00 and 10 at time  $n-1$ . In other words, there are two kinds of input sequences, (00, 01) and (10, 01), which make  $e_9$  active. Although these two input sequences both make state transition from states  $S_2$  to  $S_1$ , however, they have different energy consumption. Thus, if we want to estimate the power dissipation more accurately, then  $e_9$  can be split into two different edges with different weighting for this two sequences. In this paper, for simplicity, we use only one edge in our model but assigning the average energy consumption of these two sequences as the energy consumption of this edge.

STGPE is easy to be constructed for NAND and NOR gates. If NAND (NOR) gate has  $m$  inputs, the STGPE for NAND (NOR) has  $m+1$  states. However, the construction becomes more complex for AOI or OAI gate. In this paper, we consider NAND and NOR gates only.

For a logic gate  $PG_r$  with  $m$  inputs, if we know the energy consumption and activity number of each edge in the corresponding STGPE, we can estimate the energy consumption of  $PG_r$  as follows:

$$Energy(PG_r) = \sum_{k=1}^{m+1} \sum_{j=1}^{m+1} E_{jk}^{i_{jk}} \times W_{jk}^{i_{jk}} \quad (7)$$

Eqn. 7 means that the energy consumption of a gate can be estimated by summing the total energy consumption of each edge in the STGPE.

Assume that a combinational network  $CNI$  has  $M$  logic gates and  $N$  sequential input patterns are fed into this network with clock cycle time  $T_{cycle}$ . Then, the average power dissipation of  $CNI$  can be calculated as follows:

$$P_{avg}(CNI) = \frac{\sum_{r=1}^M \text{Energy}(PG_r)}{N \times T_{cycle}} \quad (8)$$

#### 4 Estimation of Edge Activity Number in STGPE

Basically speaking, if we neglect the input temporal correlation, the estimation of edge activity number in STGPE become the problem of finding the state probabilities by solving the exact Chapman-Kolmogorov equations[7, 8]. However, if the input temporal correlation is considered, the Chapman-Kolmogorov equations for one step transition Markov Chains can not completely reflect the information of transition probabilities. In the sequel, we will present two strategies to find the edge activity number by considering first order temporal dependent.

The first one is an exact solution method. For clarity, we simplify the notation of edge activity number of edge  $e_k$  in Fig. 3 as  $E_k$ . For instance,  $E_{10}^{11}$  in  $e_7$  is denoted by  $E_7$ . For each state  $S$  in STGPE, we have  $e(S).out_i$  signifying the  $i$ th edge that fans out from  $S$ , and  $e(S).in_j$  signifying the  $j$ th edge that fans out to  $S$ . For a fanout edge  $e(S).out_i$ , its activity number depends not only on the input transition probability but also on the activity number of all  $e(S).in_j$ . From Fig. 3, the equations of the edge activity number can be written as:

$$E_0 = (E_3 + E_7 + E_{11}) P(00/11) \quad (9)$$

$$E_{11} = (E_8) P(11/00) + (E_2 + E_6 + E_{10}) P(11/10) \quad (20)$$

where for example  $P(00/11)$  is the conditional probability of input  $00$  at time  $n$  given that input  $11$  at time  $n-1$ . There are twelve linear equations for 2-input NAND gate. If STGPE has  $m$  inputs ( $m+1$  states), then there are  $2^m \times (m+1)$  linear equations to be solved. Obviously, the complexity of this method is too high and it becomes inefficient when input number is larger than 3.

The second method is an approximation method. There are two major steps in this method. The first step is to find the state probabilities in STGPE. Secondly,

based on the state probabilities calculated, we could estimate the edge activity number for each edge. From Fig. 3, the state probability equations where for example  $P_s^n(00)$  can be written as follows:

$$\begin{aligned} P_s^n(00) &= P_s^{n-2}(00) (P(11->11) + P(10->11) \\ &\quad + P(00->11) + P(01->11)) \\ &\quad + P_s^{n-2}(10) (P(00->11) + P(01->11) \\ &\quad + P(11->11) + P(10->11)) \\ &\quad + P_s^{n-2}(11) (P(10->11) + P(00->11) \\ &\quad + P(11->11) + P(01->11)) \end{aligned} \quad (21)$$

where  $P_s^n(k)$  is the state probability of state  $k$  at time  $n$ .  $P(00->11)$  represents the probability of changing two input signals' level from  $00$  (time  $n-1$ ) to  $11$  (time  $n$ ). The other state probability equations are derived similar to Eqn. (21). Given  $K$  states, we could obtain  $K$  equations out of which any one equation can be derived from the remaining  $K-1$  equations. In addition, the summation of all state probabilities is equal to one. In Eqn. (21), for considering the input temporal correlation, we use  $n-2$  instead of  $n-1$  where  $n-1$  is used in Chapman-Kolmogorov equations[7] for temporal independent inputs. Because the inputs are assumed spatially uncorrelated,  $P(00->11)$  is equal to  $P_A(0->1)P_B(0->1)$ . According to (4), (5) and the relation  $P_i^{10} = P_i^{01}$ , the state probabilities equations can be simplified as:

$$P_s^n(00) = P_A^{one} P_B^{one} \quad (22)$$

$$P_s^n(10) = P_A^{zero} P_B^{one} + P_A^{zero} P_B^{01} + P_A^{00} P_B^{00} (P_s^{n-2}(00) + P_s^{n-2}(10)) \quad (23)$$

$$P_s^n(11) = P_A^{one} P_B^{zero} + P_A^{01} P_B^{00} + P_B^{00} P_s^{n-2}(11) \quad (24)$$

In this approximation method, we assume the state probability of state  $k$  at time  $n$  is equal to the state probability at time  $n-2$ , i.e.  $P_s^n(11) = P_s^{n-2}(11) = P_s(11)$ . In fact, experimental results show that this assumption is reasonable. According to the state probabilities calculated, Eqn. (9) to Eqn. (20) can be rewritten as follows:

$$E_0 = (E_3 + E_7 + E_{11}) P(00/11) = P_s(00) \times N \times S \times P(00/11) \quad (9)'$$

$$E_4 = P_s(10) \times N \times S \times P(00/01) + (E_0 + E_4) (P(00/00) - P(00/01)) \quad (13)'$$

$$E_{11} = P_s(11) \times N \times S \times P(11/10) + E_8 (P(11/00) - P(11/10)) \quad (20)'$$

where  $N$  is the total input patterns applied and  $S = T_{cycle}/T_{sd}$ . There is an interesting property for NAND2. In Fig. 3, no matter where the state stay at, the state will go to state 00 after applying input 11. Similarly, other NAND and NOR gates have the same property too. Based on the property, Eqn. (9)' to (12)' can be written as the simplest forms and  $E_0, E_1, E_2$ , and  $E_3$  can be solved easily. Thus, if STGPE has  $m$  inputs, we only deal with the linear equations with  $2^m$  variables for  $m+1$  times. Obviously, the complexity has reduced dramatically from  $2^m \times (m+1)$  to  $2^m$ .

## 5. Experimental results

A prototype power estimator, called CBPE, has been implemented in C on a SUN SPARCstation 10. Some benchmarks from the cmlex-91 and MCNC-91 benchmarks are used to evaluate the accuracy and efficiency of CBPE. The statistics of the benchmark examples are tabulated in Table 1. In our procedure, logic optimization and technology mapping are performed by using misII where the cell library *minimal.genlib* contains only *nand2*, *nor2*, and *inv1* gates.

From Table 1, we know that *cm150a* has the maximum number of input. In our experiments, the input signal characteristics of *cm150a* which include the signal probabilities and transition densities of each input are generated randomly and listed in Table 2. For a  $m$  inputs benchmark circuit ( $m \leq 21$ ), we take the first  $m$  inputs of *cm150a* listed in Table 2 as the input signal characteristics of the given benchmark. According to the given input signal probabilities and transition densities, a random signal generator generates 1000 random patterns with clock cycle time of 50ns. Because of the lack of efficient and accurate algorithm to estimate the transition number of nodes in the general gate delay circuit, we use VERILOG simulation to find the transition activity number at the output of each gate. Both SPICE and VERILOG simulations utilize the same input sequences for the same benchmark, and 1ns rise/fall time are assigned to the random input signals for SPICE simulation. Moreover, for both CBPE and SPICE simulation, we use the same cell-library with the same SPICE parameters. The transistor models used are the level 3 model of 0.8 $\mu$ m SPDM CMOS technology obtained from CIC (Chip Implementation Center).

In Table 3, we compare the results obtained by using exact SPICE simulation and CBPE. All measured powers are in micro-Watts and CPU time are in seconds.  $A^*$  represents the average power dissipation

by using exact SPICE simulation.  $B^*$  and  $C^*$  represent the average power dissipation by using CBPE with zero and variable gate delays, respectively. Obviously, the difference between  $B^*$  and  $C^*$  is the power dissipation contributed by hazards. Hazards are generated due to non-zero gate delay and make unwanted  $1 \rightarrow 0 \rightarrow 1$  and  $0 \rightarrow 1 \rightarrow 0$  transitions in logic simulation. In fact, hazard in SPICE simulation may not complete a full charging and discharging of a capacitance. In other words, in SPICE simulation the  $1 \rightarrow 0 \rightarrow 1$  transition of static 1-hazard ( $\square$ ) may not discharge the capacitance to 0V and the  $0 \rightarrow 1 \rightarrow 0$  transition of static 0-hazard ( $\sqcap$ ) may not charge the capacitance to  $V_{DD}$ . However, because we use VERILOG simulation to estimate the transition activity at each nodes in combinational network, any signal transition in CBPE are regarded as a complete charge and discharge. Therefore, power consumption due to hazards are overestimated in  $C^*$ . We know the power dissipation due to signal transition is proportional to the square of voltage[1], [2]. So, 50-percent off ( $D^*$ ) and 75-percent off ( $E^*$ ) of the contribution of hazards are reasonable in CBPE.

The "CBPE CPU time" in Table 3 is the CPU time of  $C^*$ . In the experimental results, VERILOG simulation take about 90-percent of CPU time in CBPE; however, it still much less than SPICE simulation. In Table 3, *Error-1*, *Error-2*, and *Error-3* are the absolute error of  $C^*$ ,  $D^*$ , and  $E^*$  with respected to  $A^*$ , respectively. It is worthy to note that there has on average only 10-percent errors larger than SPICE simulation for the worst-case  $C^*$ .

## 6. Conclusions and future work

In this paper, we have proposed a power dissipation model considering the charging and discharging of internal nodes and capacitance feedthrough effect. Based on this model, we presented a cell-based power estimation method to estimate the power dissipation of CMOS combinational network. The attractive property of this paper is that we not only compared the experimental results with exact SPICE simulation but also the result is on average within 10-percent errors.

Several major limitations of our procedure are the extension of STGPE to complex gates, transition density estimation, and the estimation of edge activity number in STGPE. It is easy to construct the STGPEs for multiple inputs NAND and NOR gates, but it become more complex for AOI and OAI gates. In future work, we will extend the construction of STGPE to AOI and OAI gates. Moreover, to reduce the estimation time, an efficient transition density simulator considering variable gate delay needs considerable efforts.

**References**

[1] A. Ghosh, S. Devadas, K. Keutzer, J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," In ACM/IEEE 29th Design Automation Conference, pp. 253-259, 1992.

[2] C. Y. Tsui, M. Pedram, A. M. Despain, "Power Estimation Considering Charging and Discharging of Internal Nodes of CMOS gates," In SASIMI'93, pp. 345-354, 1993.

[3] S. Devadas, K. Keutzer, J. White, "Estimation of Power Dissipation in CMOS combinational Circuits," In IEEE Custom Integrated Circuits Conference, pp. 19.7.1-19.7.6, 1990

[4] F. N. Najm, "Transition Density : A New Measure of Activity in Digital Circuits," IEEE Trans. Computer-Aided Design, Vol. 12, No. 2, pp. 310-323, Feb. 1993.

[5] C. Y. Tsui, M. Pedram, A. M. Despain, "Efficient Estimation of Dynamic Power Consumption under a Real Delay Model," In IEEE international Conference on Computer Aided Design, pp. 224-228, 1993.

[6] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "MIS: A multiple-level logic optimization system," IEEE Trans. Computer-Aided Design, Vol. CAD-6, pp. 1062-1081, Nov. 1987.

[7] A. Papoulis, *Probability, Random variables, and Stochastic Process*, 2nd Edition. New York: McGraw-Hill, 1984.

[8] J. Monteiro, S. Devadas and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits," In ACM/IEEE 31st Design Automation Conference, pp. 12-17, 1994.

Table 1. The statistics of examples

Circuit	Input	Output	Number of Transistors	Number of Gates	Benchmark
C17	5	3	24	6	cmlex
cm150a	21	1	280	79	
cm151a	12	2	136	39	
cm152a	11	1	90	24	
cm162a	14	5	212	60	
cm163a	16	5	202	57	
cm42a	4	10	110	33	
cm82a	5	3	96	28	
cm85a	11	3	190	55	
cmb	16	4	216	60	
con1	7	2	78	22	MCNC
f2	4	4	84	24	
rd53	5	3	244	68	
rd73	7	3	736	203	
misex1	8	7	232	67	
sao2	10	4	774	209	
f51m	8	8	616	168	

Table 2. The input signal characteristics of cm150a

( $P_1^{one}, D_1; P_2^{one}, D_2; P_3^{one}, D_3; \dots$ )  
 (0.2, 0.006; 0.6, 0.014; 0.5, 0.016; 0.8, 0.006;  
 0.7, 0.008; 0.4, 0.01; 0.6, 0.003; 0.8, 0.002;  
 0.3, 0.005; 0.2, 0.006; 0.7, 0.006; 0.5, 0.004;  
 0.1, 0.001; 0.7, 0.007; 0.6, 0.012; 0.5, 0.018;  
 0.3, 0.009; 0.2, 0.005; 0.8, 0.002; 0.5, 0.006;  
 0.4, 0.004)

Table 3. Experimental results of SPICE simulation and CBPE

Circuit	A*	B*	C*	D*	E*	SPICE CPU time	CBPE CPU time	Error-1 ((A-C)/A)	Error-2 ((A-D)/A)	Error-3 ((A-E)/A)
C17	220.7	233.1	237	234.8	233.9	530.4	4.1	7.16%	6.39%	5.98%
cm150a	2771	2809	3075	2942	2876	10051	53.7	10.99%	6.20%	3.80%
cm151a	1368	1361	1476	1419	1390	4049.4	26.4	7.92%	3.74%	1.65%
cm152a	858.9	822.3	863	842.9	832.6	2279.9	16.3	0.52%	1.86%	3.06%
cm162a	1751	1633	2031	1832	1732	6762.6	39.3	16.01%	4.63%	1.06%
cm163a	1889	1806	1996	1901	1854	6111.6	35.5	5.69%	0.67%	2.62%
cm42a	749.3	748.2	815	781.8	765.1	2449.8	12.6	8.80%	4.34%	2.10%
cm82a	1090	1082	1126	1104	1093	2506.8	17.6	3.39%	1.37%	0.35%
cm85a	1661	1689	1933	1811	1750	6051.9	35.5	16.34%	8.99%	5.32%
cmb	1127	1154	1161	1157	1155	4825.5	21.9	2.98%	2.67%	2.44%
con1	837.5	834.3	867	850.7	842.5	1822.4	12.9	3.53%	1.58%	0.60%
f2	776.4	792.2	802	796.9	794.6	1805.2	10.1	3.26%	2.64%	2.34%
rd53	2579	2445	2872	2658	2551	8438.9	44.3	11.35%	3.06%	1.08%
rd73	5406	5241	6490	5865	5553	34352	134.5	20.04%	8.50%	2.72%
misex1	2214	2266	2643	2455	2360	7106.9	39.3	19.37%	8.33%	6.59%
sao2	4625	4485	5088	4787	4636	30453	107.8	10.02%	3.50%	0.24%
f51m	4787	4592	5404	4998	4795	24836	101.1	12.88%	8.84%	0.15%
Avg. Error								9.43%	4.55%	2.48%
Total time						154433	712.9			

A\* : Average Power by using SPICE simulation  
 B\* : Average Power by using CBPE with zero gate delay  
 C\* : Average Power by using CBPE with variable gate delay  
 $D^* : D = B + (C - B) \times 1/2$   
 $E^* : E = B + (C - B) \times 1/4$