

# A Slice Analysis-Based Bayesian Inference Dynamic Power Model for CMOS Combinational Circuits\*

Chen Jie<sup>†</sup>, Tong Dong, Li Xianfeng, Xie Jinsong, and Cheng Xu

(Microprocessor Research & Development Center, Peking University, Beijing 100871, China)

**Abstract:** To improve the accuracy and speed in cycle-accurate power estimation, this paper uses multiple dimensional coefficients to build a Bayesian inference dynamic power model. By analyzing the power distribution and internal node state, we find the deficiency of only using port information. Then, we define the gate level number computing method and the concept of slice, and propose using slice analysis to distill switching density as coefficients in a special circuit stage and participate in Bayesian inference with port information. Experiments show that this method can reduce the power-per-cycle estimation error by 21.9% and the root mean square error by 25.0% compared with the original model, and maintain a 700+ speedup compared with the existing gate-level power analysis technique.

**Key words:** slice analysis; Bayesian inference; power model; CMOS combinational circuit

**EEACC:** 1210; 2570D

**CLC number:** TP391.77

**Document code:** A

**Article ID:** 0253-4177(2008)03-0502-08

## 1 Introduction

Power estimation is the basis of low power design, in which dynamic power estimation composes the main part. Input signal changes bring charge and discharge in node capacity and result in dynamic power dissipation. The dynamic power aroused by a pair of input vectors  $(x_1, x_2)$  can be computed as:

$$P = \frac{1}{2} f V_{dd}^2 \sum_{i=1}^N C_i n_i(x_1, x_2) \quad (1)$$

In Eq. (1), voltage  $V_{dd}$  and frequency  $f$  usually are easy to estimate,  $N$  is the total number of nodes,  $C_i$  and  $n_i(x_1, x_2)$  are the capacity of node  $i$  and the switching number caused by the vector pair  $(x_1, x_2)$  at node  $i$ , respectively. Measuring these two parameters is the main difficulty of dynamic power estimation.

Traditionally, dynamic power is estimated through simulation and computed based on the switching activity record of all nodes<sup>[1]</sup>. Those methods are time consuming, though result in good accuracy. Differing from simulation based methods, analytical methods study the relationship between dynamic power and circuit characteristics, which avoid slow analysis due to massive computation for all nodes<sup>[2~5]</sup>. Reference [2] used coefficients to build a look-up table and computed average dynamic power, but could not compute cycle-accurate power. Though References

[3~5] built a cycle-accurate power macro-model, the building process and format in Refs. [3,4] are complicated, and the model in Ref. [5] only describes the power as a linear function of switching activity, which predigests too much. References [6,7] used Bayesian networks to build models, but these models only support average switching activity computing, and transforming from a gate list to a direct acyclic graph brings massive computing and memory loads. References [8,9] used neural networks to build cycle-accurate power models, but because the establishment of the coefficients depends on circuit structure and power dissipation, the node analysis makes it lack good scalability or the model only investigates the switching density of the input and output signals but ignores abundant internal states.

Based on analysis of signal temporal and spatial correlation, Reference [10] proposed a build dynamic power model using port coefficients and calculated power dissipation by Bayesian inference. To optimize a cycle-accurate dynamic power model accuracy, this paper diagnoses the error in Ref. [10], analyses the power distribution and internal node state under fixed port coefficients, and demonstrates the deficiency of only using port information. Then, we define the concept of slice, propose distilling circuit internal coefficients by slice analysis techniques, and build a Bayesian inference model with port information. The experiments upon ISCAS85 show that this method can,

\* Project supported by the National High Technology Research and Development Program of China (No. 2004AA1Z1010) and the National Natural Science Foundation of China (No. 60703067)

<sup>†</sup> Corresponding author. Email: chenjie@mprc.pku.edu.cn

Received 4 August 2007, revised manuscript received 9 November 2007

compared with Ref. [10], reduce the power-per-cycle estimation error by 21.9%, the root mean square (RMS) error by 25.0%, and maintain 700+ speedup compared with the existing gate-level power analysis technique.

## 2 Bayesian power model

We introduce a multiple coefficients-based Bayesian inference dynamic power model, which is the basis of this paper. Details can be found in Ref. [10].

### 2.1 Bayesian classification and Bayesian theory

Sample space  $S$  is composed of eigenspace  $I$  and class space  $C$ :  $S = \{s_1, s_2, \dots, s_n\} = \langle I, C \rangle$ , each instance is a Descartes product with  $m$  characteristics:  $I = \langle P_1, P_2, \dots, P_m \rangle$ , eigen characteristics is discrete value  $p_{i,k}$  ( $i = 1, 2, \dots, m$ ), and class can be  $l$  discrete values  $C = \{C_1, C_2, \dots, C_l\}$ . The essence of the classification problem is to find a mapping function  $f$  from instance space to class space, for any instance  $A = (p_1, p_2, \dots, p_m) \in I$  and  $P(A) > 0$ , there is only one corresponding  $c_i \in C$  ( $i = 1, 2, \dots, l$ ). Bayesian classification regards the class with posterior probability maximum,  $B_i = \{c = c_i\}$  ( $i = 1, 2, \dots, l$ ), as the image of  $A$ , that is

$$P(B_i | A) \geq P(B_j | A), \quad j = 1, 2, \dots, l \quad (2)$$

And from Bayesian theory<sup>[11]</sup>:

$$P(B_i | A) = \frac{P(A | B_i)P(B_i)}{\sum_{j=1}^n P(A | B_j)P(B_j)}, \quad i = 1, 2, \dots, n \quad (3)$$

Because

$$P(A) = \sum_{j=1}^n P(A | B_j)P(B_j) \quad (4)$$

is fixed, so we only need to maximize  $P(A | B_i) \times P(B_i)$ . In Eq. (3),  $P(A | B_i)$  is priori probability,  $P(B_i | A)$  is posterior probability. Because instance  $A$  has  $m$  kinds of characteristics and it is supposed that each class is independent, so

$$P(A | B_i) = \prod_{k=1}^m P(p_k | B_i), \quad i = 1, 2, \dots, n \quad (5)$$

Considering the historical information of random events in whole space, Bayesian classification has good results for non-linear distributed data classification.

### 2.2 Bayesian inference power model

We build a dynamic power model based on Bayesian classification, regard the power value as a class space  $C$ , and select special circuit coefficients to compose the eigenspace  $I = \langle P_1, P_2, \dots, P_m \rangle$ . These

coefficients represent circuit state changing information and participate in posterior probability computing. This method calculates the probability of  $B_i = \{c = c_i\}$ , which is the image of  $A = (p_1, p_2, \dots, p_m) \in I$ , and finds the classification result  $c_i$ .

Based on the signal temporal and spatial correlation analysis, we define input signal density  $P_{in}$ , input transition density  $D_{in}$ , and output transition density  $D_{out}$  (zero-delay model) as coefficients used in Bayesian classification:

$$P_{in,k} = \frac{1}{n} \sum_{i=1}^n x_{i,k} \quad (6)$$

$$D_{in,k} = \frac{1}{n} \sum_{i=1}^n x_{i,k-1} \oplus x_{i,k} \quad (7)$$

$$D_{out,k} = \frac{1}{m} \sum_{i=1}^m y_{i,k-1} \oplus y_{i,k} \quad (8)$$

Equations (6) ~ (8) define input signal density  $P_{in,k}$ , input transition density  $D_{in,k}$ , and output transition density  $D_{out,k}$  in the  $k$ -th cycle, respectively.  $x_{i,k}$  ( $y_{i,k}$ ) is the  $i$ -th input (output) signal value in the  $k$ -th cycle,  $n$  and  $m$  are the total number of input and output signals, respectively.

## 3 Deficiency of original coefficients

With the port signal coefficients, we can compute power per cycle easily. But after further analysis, we find that the circuit internal states could vary greatly under the same port coefficients values. Because state switching is the reason for dynamic power dissipation, we show the internal node transition under two sets of input signal switching in C17.

In Fig. 1, the  $D_{in}$ ,  $D_{out}$ , and  $P_{in}$  of the input transition  $A_0 A_1$  and  $B_0 B_1$  are all the same, but  $B_0 B_1$  makes 3 nodes change state, while  $A_0 A_1$  changes none (the numbers behind each gate are output under  $A_0 A_1$  and  $B_0 B_1$ , respectively).

Through analyzing the internal node state, we find that three pairs of input switching (N1, N3, N6) vanished in gate G1 and G2, which prevents latter gates from switching. In other words, because of different logics and connections among the gates, input signal switching could be absorbed at different internal nodes and could not pass the state changing to latter gates. Using only the port signal switching as coefficients, internal transition information will be lost, and we cannot know the spread status of state transition and whether most internal nodes make state switching.

Another example is modules with enable signals. Fig. 2 is a 2bit adder with an enable signal. When the enable signal is 0, internal nodes will not switch, but input  $A_0 A_1 B_0 B_1$  change. From this example, we see

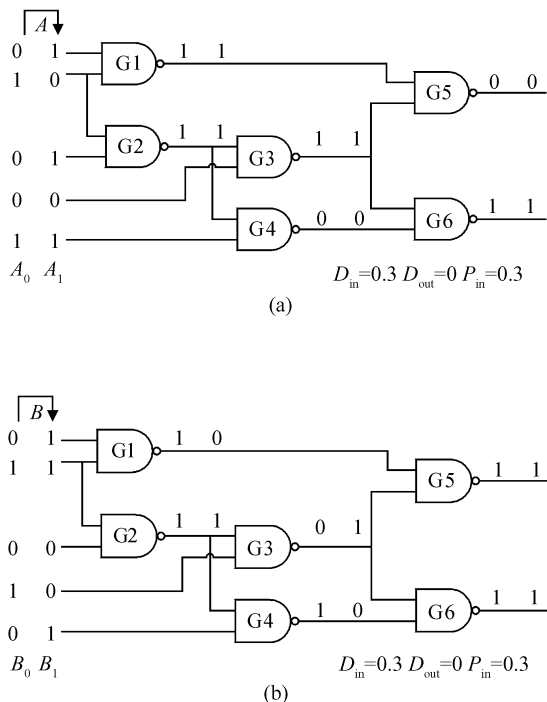


Fig.1 Comparison of C17 node switching under two input sets

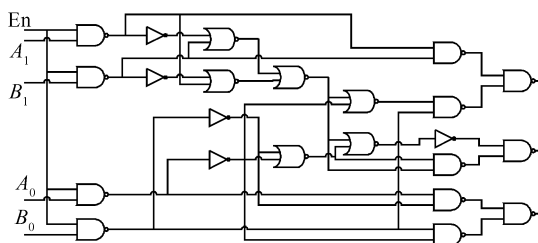


Fig.2 Scheme of 2bit adder with enable signal

that some signals have a more important effect because they or their successive gates have a big fan-out. At that time, we could not get the real working characteristics if our estimation only depends on port information.

By further quantitative analysis, we find that this phenomenon commonly exists in working circuits. Figure 3 shows the first and second level gate switching

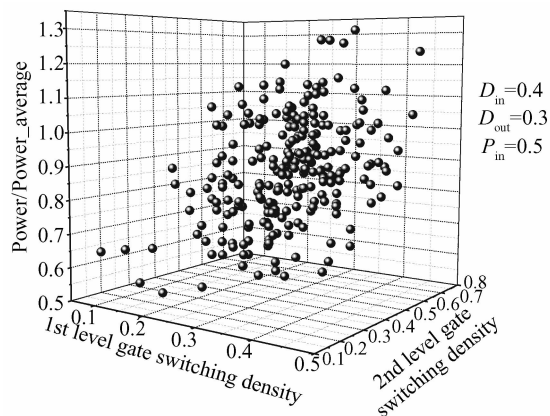


Fig.3 Power and circuit internal level parameters analysis of C880

Table 1 Comparison of C880 power and internal level parameters when  $D_{in} = 0.4$

	Minimum	Maximum	Maximum/Minimum
First level gate switching ratio	0.13	0.53	2.3
Second level gate switching ratio	0.04	0.72	18
Power/Average power	0.52	1.30	2.5

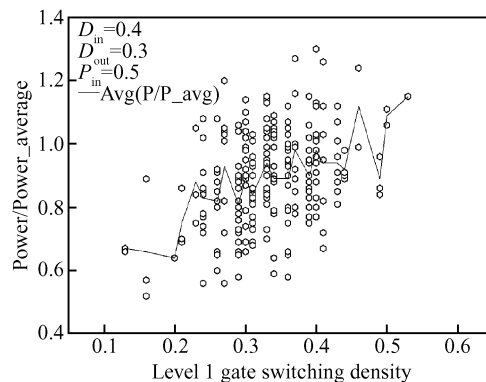


Fig.4 Switching density of the first level gate and total power of C880

density and total power distribution under special  $D_{in}, D_{out}$ , and  $P_{in}$  for 10000 random sample cycles of C880. The switching ratio of successive levels varies widely under the same port signal coefficients values, which results in the circuit total power having a big difference. Table 1 shows the extremes from statistics, from which we find that there are unignorable differences in switching activity and power.

Figures 4 and 5 show the relation analysis result between the power and internal gates' switching ratio. When the internal gates' switching density increases, total power exhibits an increasing trend. The sample number at some density points is small, but that does not affect the trend judgment.

Therefore, under the same port coefficient values, circuit internal states vary widely, while power dissipation and internal gates switching density has a

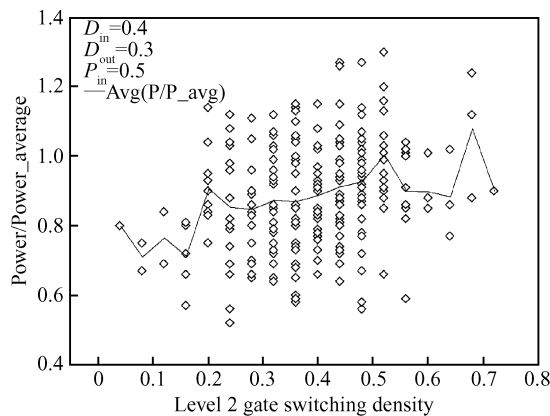


Fig.5 Switching density of the second level gate and total power of C880

positive trend. We introduce this relationship into the power macro model, and propose using circuit slice analysis to obtain internal information, which increases the model's accuracy as a result.

## 4 Circuit slice analysis

### 4.1 Slice analysis

Based on the above analysis, we hope to extract run-time internal information to build a power model and reduce estimation error. Based on logic connections, we assign a depth value to each gate and define it as level, which is computed as follows:

- (1) All primary input levels are 0;
- (2) The level of a non-NOT gate is the maximum of all input signal levels plus 1;
- (3) The gate output signal level equals the gate level;
- (4) The NOT gate output signal level equals the input signal level;

The reason for treating the NOT gate specially is that the input switching of the NOT gate results in output switching. If we increase the NOT gate output signal level value, the anterior level state transition would be counted into the current level repeatedly. So, we treat the NOT gate in a different way, not increasing the output signal level value and not counting it in the current level coefficient calculation.

Based on the above calculations, a circuit module could be divided into  $d$  level gate sets according to each gate level value, which is from 1 to  $d$ . The signal level value is from 0 to  $d$ , in which the 0 level is the primary input, the level 1 signal is the level 1 gate's output signal, and the level  $d$  signal is the output port signal. In Fig. 1, for example, G3 is the second level gate (the 2nd level slice) and G5 is the third level gate (the 3rd level slice).

Based on these calculations, we define that gates with same level value compose a slice and define  $D_{L,k}$ , the switching density of the  $L$ -th level slice in  $k$ -th cycle, as follows:

$$D_{L,k} = \frac{1}{n} \sum_{i=1}^n x_{Li,k} \oplus x_{Li,k-1} \quad (9)$$

where  $x_{Li}$  is the  $i$ -th output signal value in the  $L$ -th slice, and  $n$  is gate number of this level. We call this method slice analysis and it extracts the circuit's internal information effectively. Through this method, different  $D_{L,k}$  are extracted and take part in the computer model with port signal coefficients, resulting in more accurate mapping from model coefficients to power.

### 4.2 Slice method

The coefficient extracting method in the slice method has great impact on modeling efficiency and accuracy. We propose the following extracting method:

(1) Extract the first  $n$ -level signal switching density. This method's advantage is when value  $n$  is not large, we can use script representing first  $n$ -level logical connection, and extract coefficient values without simulation.

(2) Extract the first  $n$ -level signal switching density with most gates. Slices with the most gates usually have typical power dissipation characteristics and adapt to be representative of internal switching.

(3) Divide a module into  $n$  blocks according to the logical depth, and extract the signal switching density of the level with the most gates in each block. Methods 1 and 2 may extract successive slices in logical connection, which results in high correlation in coefficients and reduces information contained in slices or get a slice with too few gates, making it hard to exhibit a difference of varied input vectors. Selecting the slice with the most gates in different sub blocks can compromise those two aspects, and provide representative switching information in varied logical depth for Bayesian inference.

Using slice analysis to extract runtime internal information, new gate level netlists and simulation circumstances need to be generated. We use a script to automatically finish gate depth analysis, and create a new netlist and simulation circumstance generation. Figure 6 shows the modeling process, and Figure 7 shows the slice analysis algorithm (selecting method 2 as an example).

## 5 Experiment and result analysis

Figure 8 shows the slice analysis-based Bayesian inference power modeling and verification process. According to the circuit scale, we select ISCAS85 benchmark circuits as experimental objects. We use gates with two input ports and a NOT gate as a target library and synthesize the circuit into gate level netlist. Slice analysis is realized in Perl, the gate level simulation used Synopsys VCS7.0, the power estimation used PrimePower4.2, and the sample analysis and Bayesian inference were coded in C. All experiments are accomplished on a Sun blade2000 workstation, which has two 900MHz UltraSPARC III processors and 4G memory. We use 1000 cycles of random sample data to build the Bayesian classification, and use 500 cycles of random sample data to verified the pow-

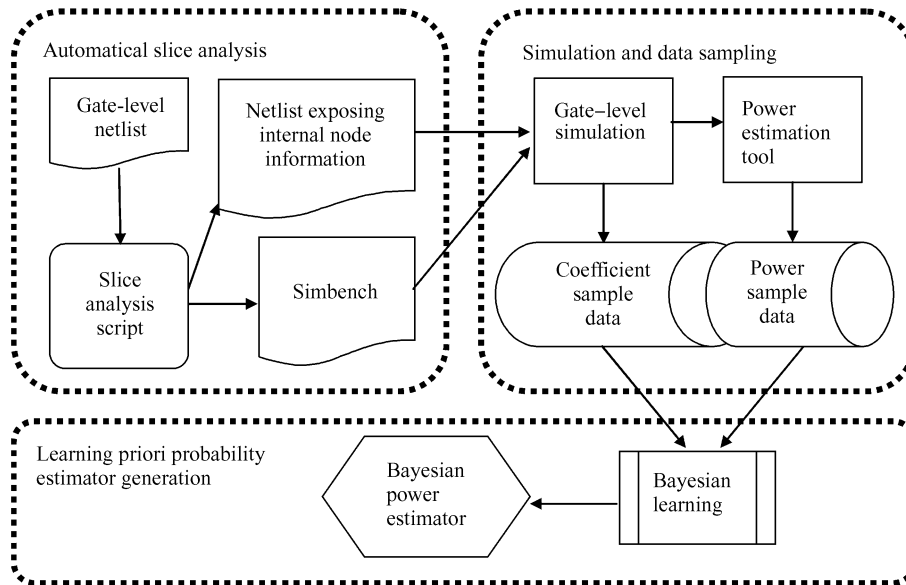
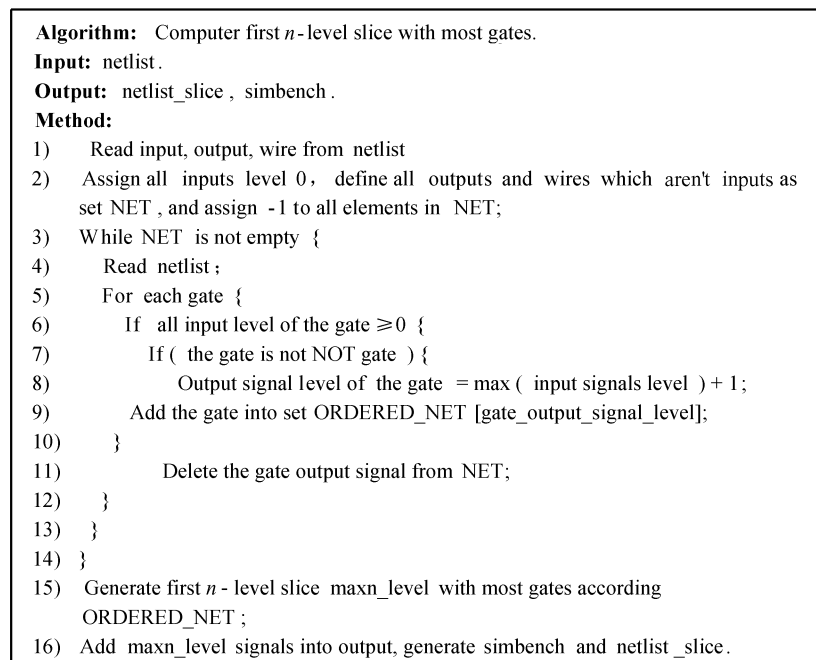


Fig.6 Bayesian model building process based on slice analysis

Fig.7 Algorithm of extract top- $n$  level circuits with most gates

er calculation. The power model error is computed through a comparison between the Bayesian inference result and the gate level estimation result, which is expressed in average error and RMS error of power-per-cycle.

Using the module with 3-level slice coefficients as example, we evaluated the effect of this slice based dynamic power model for three kinds of proposed slice extraction methods. The comparison between the model adding slice coefficients and the original model that only uses port information is shown in Table 2, in which  $E$  is average power-per-cycle error,  $V$  is RMS error,  $\Delta$  is the relative error reduction of the slice

analysis-based model compared with the original model, subscript O means original Bayesian model, and subscript SM, SS and SF stand for three proposed extraction methods; slices with the most gates, slices with the most gates in blocks, first  $n$  level slices.

Table 2 shows that adding slice coefficients brings a notable reduction to the original Bayesian model error. On average, it has better results than extracting slices with the most gates and slices with the most gates in blocks, whose average error and RMS error are all reduced by above 20%; and extracting the first three level slices results in a relatively weak optimization, whose average error and RMS error are

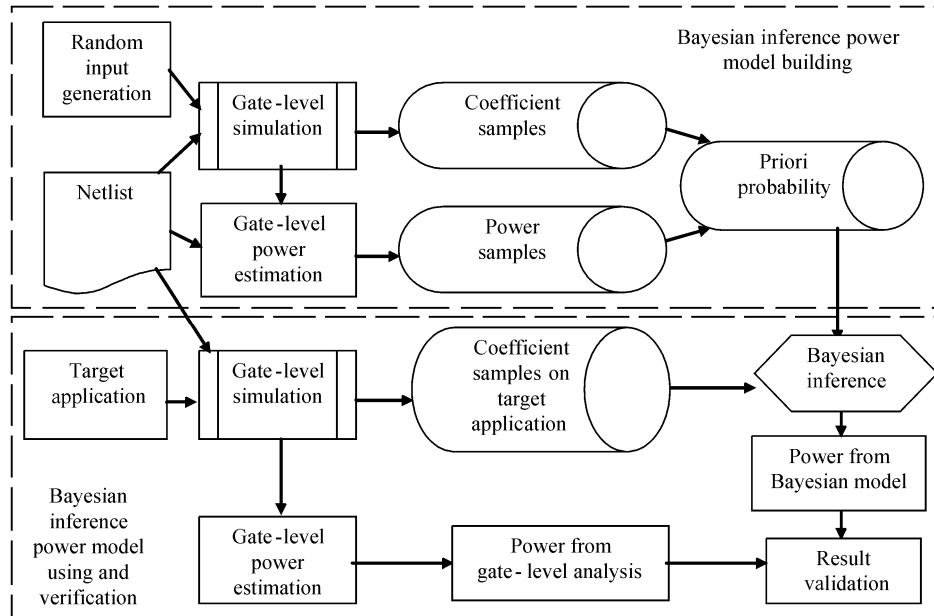


Fig.8 Bayesian inference power model building and validation

Table 2 Power estimation result comparison between 3-level slice analysis and original model

Circuit	Average error of power-per-cycle							RMS error of power-per-cycle						
	Original	SM		SS		SF		Original	SM		SS		SF	
	$E_O$	$E_{SM}$	$\Delta E_{SM}/\%$	$E_{SS}$	$\Delta E_{SS}/\%$	$E_{SF}$	$\Delta E_{SF}/\%$	$V_O$	$V_{SM}$	$\Delta V_{SM}/\%$	$V_{SS}$	$\Delta V_{SS}/\%$	$V_{SF}$	$\Delta V_{SF}/\%$
C432	0.150	0.117	22.0	0.116	22.7	0.146	2.7	0.212	0.153	27.8	0.154	27.4	0.200	5.7
C880	0.149	0.093	37.6	0.096	35.6	0.132	11.4	0.198	0.119	39.9	0.123	37.9	0.171	13.6
C1355	0.078	0.074	5.1	0.071	9.0	0.068	12.8	0.101	0.095	5.9	0.089	11.9	0.086	14.9
C1908	0.102	0.084	17.6	0.087	14.7	0.079	22.5	0.134	0.111	17.2	0.115	14.2	0.105	21.6
C3540	0.111	0.103	7.2	0.097	12.6	0.103	7.2	0.166	0.139	16.3	0.131	21.1	0.139	16.3
C6288	0.087	0.064	26.4	0.065	25.3	0.064	26.4	0.113	0.082	27.4	0.083	26.5	0.081	28.3
C7552	0.096	0.069	28.1	0.077	19.8	0.053	44.8	0.122	0.085	30.3	0.096	21.3	0.069	43.4
Average	0.110	0.086	21.9	0.087	21.2	0.092	16.6	0.149	0.112	25.0	0.113	24.4	0.122	18.6

reduced by 16.6% and 18.6%, respectively.

On the other hand, extracting the first  $n$  level slices has a relatively better optimization in big scale circuit modeling, e. g., estimations on C6288 and C7552 are more accurate. The reason is that the first  $n$  level slices in large scale circuits usually have big number gates and result in a large switching density change range, while deeper level slices have a relatively small switching density change range because anterior switching is absorbed by middle cells. Thus, the first  $n$  level slices could provide a broader coefficient change and reduce model error.

Table 3 uses the SM method as an example to compare the results of our model and Refs. [2, 4]. References [5, 8, 9] did not evaluate ISCAS85, and References [6, 7] aimed at node average switching probability but not power-per-cycle, so they are not proper to compare directly. Compared with the cycle-accurate power estimation in Refs. [3, 4], the average error of our model is reduced by 24.6% and 7.5%. Compared with the total power estimation in Refs.

[2, 3], the error of our model is reduced an order on average. Slice analysis can distinguish different internal power dissipations under the same port switching density, thus increases model accuracy.

Table 4 uses the 500-cycle power estimation as an example to list the time consumption of the slice analysis-based power model. After adding slice analysis, the simulation and model computation time increase over the gate level power estimation time can be almost ignored and it has a 700+ speedup under in-

Table 3 Comparison of different estimation techniques

Circuit	Average error of power-per-cycle			Error of total power		
	Wu <sup>[3]</sup>	Gupta <sup>[4]</sup>	Our model	Wu <sup>[3]</sup>	Gupta <sup>[2]</sup>	Our model
C432	0.193	0.134	0.117	0.0310	0.0441	0.0008
C880	0.143	0.119	0.093	0.0320	0.0362	0.0013
C1355	0.093	0.058	0.074	0.0270	0.0403	0.0021
C1908	0.116	0.089	0.084	0.0200	0.0373	0.0047
C3540	0.125	0.110	0.103	0.0200	0.0322	0.0064
C6288	0.062	0.076	0.064	0.0190	0.0222	0.0036
C7552	0.069	0.065	0.069	0.0110	0.0265	0.0046
Average	0.114	0.093	0.086	0.0229	0.0341	0.0034

Table 4 Time cost and speedup of slice analysis power model

Circuit	Gate-level power ana- lysis time	Simulation time		Computation time		Speedup	
		Original	Slice	Original	Slice	Original	Slice
C432	158.3	0.108	0.117	0.16	0.17	590.7	551.6
C880	210.0	0.150	0.183	0.16	0.17	677.4	594.9
C1355	234.2	0.158	0.217	0.16	0.17	736.5	605.2
C1908	210.8	0.142	0.167	0.16	0.17	698.0	625.5
C3540	340.8	0.217	0.292	0.16	0.17	904.0	737.7
C6288	1894.2	0.808	1.650	0.16	0.17	1956.8	1040.8
C7552	630.0	0.375	0.608	0.16	0.17	1177.6	809.8
Average	525.5	0.280	0.462	0.16	0.17	963.0	709.3

increased accuracy. At the same time, we note that power computation time in the Bayesian model will not change as circuit scale increases. This scalability is extremely suitable for big scale circuit power estimation.

We also analyze the slice number impact on error. Because all circuits have similar results, we only describe those from C880. As Figure 9 shows, when the slice number changes from 0 to 3, power-per-cycle average error and RMS error has a notable reduction; once the slice number is larger than 5, increasing the slice number only brings a relatively small amount of additional information and further reduces errors mildly.

Using 30 groups of random input vectors to build and verify the model, we test model sensitivity for input data. Because the results have similar characteristics, we describe the relation between model error and input group with C432, which is shown in Fig. 10. We see that the RMS error of estimation errors in all groups is 0.24%; this indicates that our model is insensitive to the input data for building models and estimation.

Slice analysis can effectively reduce the Bayesian dynamic power model error using little computation and memory dissipation. Extracting slices with the

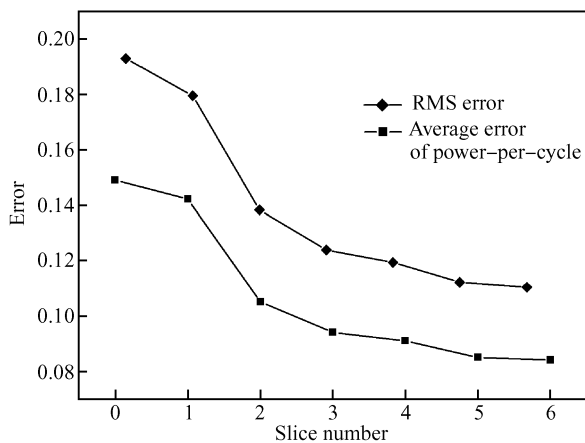


Fig.9 Impact of slice number on power estimation error in C880

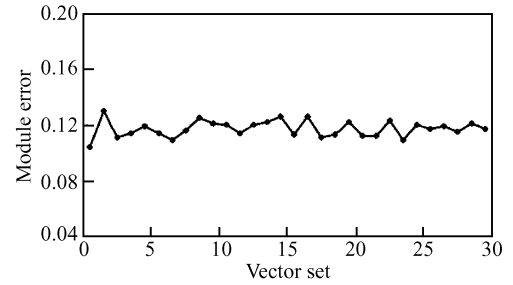


Fig. 10 Impact of input sample on model accuracy in C432

most gates and the most gates in the blocks are relatively more effective for middle or small scale circuits, while extracting the first  $n$ -level slice has better results and time consumption for large scale circuits. When building models, selecting three level slices can usually obtain good accuracy.

## 6 Modeling and estimation cost

Next, we analyze modeling and estimation costs. The time cost of building a slice analysis-based power model is:

$$T_{\text{modeling}} = T_{\text{sample\_simulation}} + T_{\text{sample\_power\_estimation}} + T_{\text{extract\_slices}} \quad (10)$$

where simulation and gate level power estimation time is inevitable in all kinds of analytical modeling methods (e. g. Refs. [2~4,8,9]), because the original data come from this process. Figure 5 shows the modeling time using 1000 cycle samples. Most of the time is consumed during the sample gate-level power estimation, e. g. 98.3% in SM, 99.5 in SF. Extracting slices in modeling is a process composed of node levelization and sorting, whose time complexity are  $n$  and  $n \lg n$ <sup>[12]</sup> (for SF, the cost is much smaller because it only levelizes partial circuits and does not need sorting). So, a circuit scale increase will not bring much complexity to slice processing. Figure 5 shows that extracting slices only adds a 1% time cost based on gate level simulation and power analysis.

The computation cost of our model is small. Using  $D_{\text{in}}$ ,  $D_{\text{out}}$ ,  $P_{\text{in}}$ , and  $n$  level  $D_{\text{L}}$  as coefficients to participate in Bayesian inference, supposing the power class number is  $L$ , and using weighting addition based on probability to calculate power, then  $L(n+3)$  times multiplication and  $L-1$  times addition is needed to calculate the power of each cycle. In high level power estimation,  $L$  usually is not large ( $<20$ ), thus the computation cost is very small. At the same time, port number and slice scale usually increase linearly as circuit scale increases, so the sample and computation consumption of the model are affected little by the circuit, which results in good scalability.

Table 5 Time of building model with 1000-cycle sample

Circuit	$T_{GLPA}$	$T_S$		SM modeling			SF modeling		
		Original	Slice	$T_E$	$T_{Total}$	$T_{GLPA}/\%$	$T_E$	$T_{Total}$	$T_{GLPA}/\%$
C432	466.7	0.167	0.183	1	468.1	99.7	<1	467.6	99.8
C880	590.0	0.250	0.292	3	593.5	99.4	1	591.5	99.7
C1355	724.2	0.267	0.347	5	729.8	99.2	2	726.8	99.6
C1908	610.8	0.255	0.267	3	614.3	99.4	1	612.3	99.8
C3540	1087.5	0.342	0.558	11	1099.4	98.9	5	1093.4	99.5
C6288	6965.0	1.530	3.253	410	7379.8	94.4	48	7017.8	99.2
C7552	2027.5	0.642	1.125	66	2095.3	96.8	26	2055.3	98.6
Average	1781.7	0.493	0.861	71	1854.3	98.3	14	1795.0	99.5

\* :  $T_{GLPA}$  is gate-level power analysis time dissipation,  $T_S$  is simulation time dissipation,  $T_E$  is slice extraction time dissipation,  $T_{Total}$  is the sum of  $T_{GLPA}$ ,  $T_S$  and  $T_E$ .

Our model also has small memory consumption: suppose the class number is  $L$  and the coefficients class number is  $M$ , then the priori probability memory requirement is  $L((3+n)M+1)$ , which will not increase with circuit scale either.

## 7 Conclusion

The slice analysis method extracts a key level coefficients from inside the circuit and uses those coefficients to build a Bayesian power model. This method considers not only port information but also the internal node switching situation, and reduces the error caused by estimating only using port information. Experiments indicate that power estimation based on this method has good accuracy and speedup. At the same time, it is insensitive to input data.

### References

[ 1 ] Najm F. A survey of power estimation techniques in VLSI circuits. *IEEE Trans Very Large Scale Integr Syst*, 1994, 2(4): 446  
 [ 2 ] Gupta S, Najm F. Power modeling for high-level power estimation. *IEEE Trans Very Large Scale Integr Syst*, 2000, 8(1): 18  
 [ 3 ] Wu Q, Qiu Q R, Ding C S. Cycle-accurate macro-models for RT-level power analysis. *IEEE Trans Very Large Scale Integr Syst*,

1998, 6(4): 520  
 [ 4 ] Gupta S, Najm F. Energy and peak-current per-cycle estimation at RTL. *IEEE Trans Very Large Scale Integr Syst*, 2003, 11(4): 525  
 [ 5 ] Chaudhry R, Stasiak D, Posluszny S, et al. A cycle accurate power estimation tool. *Proceedings of the Asia and South Pacific Conference on Design Automation*, Yokohama, 2006: 867  
 [ 6 ] Bhanja S, Ranganathan N. Dependency preserving probabilistic modeling of switching activity using Bayesian networks. *Proceedings of the 38th Conference on Design Automation*, Las Vegas, 2001: 209  
 [ 7 ] Ramani S S, Bhanja S. Any-time probabilistic switching model using Bayesian networks. *Proceedings of the International Symposium on Low Power Electronics and Design*, Newport Beach, 2004: 86  
 [ 8 ] Cao L. Circuit power estimation using pattern recognition techniques. *Proceedings of International Conference on Computer Aided Design*, San Jose, 2002: 412  
 [ 9 ] Hsieh W T, Shiu C C, Liu C N J. A novel approach for high-level power modeling of sequential circuits using recurrent neural networks. *Proceedings of International Symposium on Circuits and Systems*, Kobe, 2005: 3591  
 [ 10 ] Chen Jie, Zhao Xiaoying, Li Xianfeng, et al. Three-dimensional coefficient based Bayesian inference power model. *Journal of Computer-Aided Design and Computer Graphics*, 2007, 19(10): 1241 (in Chinese)  
 [ 11 ] Geng Suyun, Zhang Li'ang. *Probability and statistics*. Beijing: Peking University Press, 1998 (in Chinese)  
 [ 12 ] Even S. *Graph algorithms*. Rockville, MD: Computer Science, 1979

## 基于切片分析的 CMOS 组合电路贝叶斯动态功耗模型\*

陈杰<sup>†</sup> 佟冬 李险峰 谢劲松 程旭

(北京大学微处理器研究开发中心, 北京 100871)

**摘要:** 为改善周期精确级功耗分析的准确度和速度问题, 使用多维特征参数建立贝叶斯推理的动态功耗模型. 基于功耗分布与电路内部节点状态的分析, 发现仅使用端口信息作为参数的不足. 定义了门单元级数的计算和对应切片的概念, 提出使用切片分析的技术提取电路内部关键层的翻转密度作为参数, 与端口信息共同参与贝叶斯推理. 基于 ISCAS85 基准电路的实验结果表明, 该方法使原始模型的误差降低 21.9%, 均方差降低 25.0%, 同时保持了相对现有门级功耗分析 700 倍的加速比.

**关键词:** 切片分析; 贝叶斯推理; 功耗模型; CMOS 组合电路

**EEACC:** 1210; 2570D

**中图分类号:** TP391.77

**文献标识码:** A

**文章编号:** 0253-4177(2008)03-0502-08

\* 国家高技术研究发展计划(批准号:2004AA1Z1010)及国家自然科学基金(批准号:60703067)资助项目

<sup>†</sup> 通信作者. Email: chenjie@mprc.pku.edu.cn

2007-08-04 收到, 2007-11-09 定稿