Towards the Capability of Providing Power-Area-Delay Trade-off at the Register Transfer Level

Chun-hong Chen Chi-ying Tsui
Department of Electrical and Electronic Engineering
The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
Tel: 852-2358-7071
E-mail: eechench@ee.ust.hk, eetsui@ee.ust.hk

1. ABSTRACT

This paper presents a new register-transfer level (RT-level) power estimation technique based on technology decomposition. Given the Boolean description of a circuit function, the power consumption of two typical circuit implementations, namely the minimum area implementation and the minimum delay implementation, are estimated, respectively. This provides a capability of obtaining a full power-delay-area trade-off curve at the RT level. Our method makes it possible to capture the structural and/or functional information of a circuit without going through actual gate-level implementation. Experimental results show that the accuracy is very reasonable.

1.1 Keywords

RT-level, power estimation, entropy, technology decomposition

2. INTRODUCTION

Power reduction has become one of the primary goals in the design of modern digital systems due to the increasing demand for low power circuits in portable applications. Increasing package and cooling cost is another driving factor. To achieve low power design, the designer has to explore the design space to make the appropriate power-area-delay trade-off decision. Accurate power estimation at different abstraction levels is thus urgently needed to carry out the correct design space exploration.

Recently several RT level power estimation methodologies were proposed based on entropy and information theoretic approaches [1, 2]. However, these approaches are suffering from two main discrepancies. First, entropy of the function of the circuit is used to estimate the circuit switching activity as well as to model the area cost of a circuit [2-4]. Unfortunately, the accuracy of entropy-based power estimation is very limited since the capacitance model using entropy does not work well over a wide range of circuits. Secondly, these methods only give a single power estimate for a given functional description of the circuit. They use very little information about the function and complexity of the circuit at the behavioral level and also do not account for the effect of different potential circuit implementations for different requirements. In this paper, we are targeting to provide a capability to generate the power-area-delay trade-off curve at the RT level. In particular, we propose a method to estimate the power consumption for the *minimum area implementation (MAI)* and the *minimum delay implementation (MDI)* given a functional description of a circuit. These are the two extreme points of the power-area-delay trade-off curve whose power estimation serves as a first step to generate the full trade-off curve.

The remainder of the paper is organized as follows. In Section 3, we describe the power estimation technique for the *MAI* based on technology decomposition. We discuss the modeling of node distribution, capacitance distribution as well as entropy distribution for power estimation. Section 4 is devoted to the power estimation for the *MDI*, including the method of estimating the delay and the total capacitance for the *MDI*. Experimental results and discussions are provided in Section 5 and, finally, conclusion is given in Section 6.

3. POWER ESTIMATION FOR THE MINIMUM AREA IMPLEMENTATION

3.1 Power Model

For a combinational CMOS logic circuit, the average dynamic power dissipation is given by

$$P_{avg} = \tfrac{1}{2} f_{clk} V_{dd}^2 \sum_g C_{load}(g) \cdot sw(g) \qquad (1)$$

where f_{clk} is the clock frequency, V_{dd} is the supply voltage, $C_{load}(g)$ is the load capacitance of gate g , and $sw(g)$ is the average number of transitions at gate g per clock cycle. Here we are ignoring the power consumption due to short-circuit and leakage currents which are negligible for the

This work was supported in part by RGC research grant HKUST 779/96E.

well-designed circuits [5]. Assuming the primary inputs of the circuit are temporally independent, one can reasonably replace *sw(g)* of (1) with ½ *h(g)*, where *h(g)* is the output node entropy of gate *g* [1, 2]. Thus

$$P_{avg} \approx \tfrac{1}{4} f_{clk} V_{dd}^2 \sum_g C_{load}(g) \cdot h(g) \qquad (2)$$

Obviously, the exact values of $C_{load}(g)$ and $h(g)$ are not available until the gate-level circuit implementation is known. If the circuit is *levelized* and, we know the average capacitance per node at each level, then we can approximate the power consumption as

$$P_{avg} \approx \tfrac{1}{4} f_{clk} V_{dd}^2 \sum_{i=0}^{K} \left(\frac{C_i}{n_i} \cdot H_i\right) \qquad (3)$$

where C_i, n_i and H_i are the sum of node capacitances, the number of nodes and the sum of node entropies at level *i*, respectively, and *K* is the largest level number. A logic circuit is levelized such that the output node of each gate is assigned a specific level number which represents its distance from the primary inputs. All primary inputs are said to be at level zero. The output node of a gate whose inputs are all primary inputs is said to be at level one. The output node of a gate whose inputs are either outputs of level one gates or primary inputs is said to be at level two, and so on. The largest level number is also called the *circuit depth*. As we can see from (3), the key factors for power estimation are the capacitance distribution, node distribution and entropy distribution with respect to the circuit levels. Instead of trying to predict these factors just from a given Boolean function, the scheme here is to "*capture*" the information about its implementation as much as possible from the technology decomposition of the Boolean function. Although the effects of logic minimization and technology mapping on the final implementation can not be fully captured, the decomposed network can at least give some idea on the structural and/or functional information such as the distribution of internal nodes, number of logic levels, literal count and the level distribution of the primary outputs. With this information, we can predict the models for the node, capacitance and entropy distribution for the *MAI* of a Boolean function.

3.2 Node Distribution

Let K_d and K_m be the largest level numbers in the *decomposed network* (*DN*) and the *mapped network* (*MN*) for area optimization of a given Boolean function, respectively. In the technology dependent phase of logic synthesis, technology mapping is usually performed after a decomposition step. The mapping itself consists of network covering step which transforms the whole Boolean network into an acceptable design by selecting which subsets of the network nodes shall be collapsed and mapped to a single cell of the target library. Thus, in general, we have $K_d \geq K_m$, i.e., $K_m = \lceil K_d / \alpha \rceil$, where $\alpha \geq 1$. Intuitively, the collapsed nodes in the *DN* are those which are close to one another in terms of their level numbers. Therefore, we can expect the node distributions of the *DN* and that of the *MN* are very similar. In other words, if we assume the *DN* and *MN* have the same total number of nodes, when we compress the node distribution curve (number of nodes against level number) of the *DN* in the level number's axis from K_d to K_m while maintaining the area under the curve (i.e. the total number of nodes), the shape of the resulting curve will be very similar to that of the *MN*. Thus the number of nodes at level *i* ($0 \leq i \leq K_m$) in the *MN*, denoted by n_i, could be estimated as the algebraic average of the number of nodes from level $\lceil (i\text{-}1)\cdot K_d / K_m \rceil$ to level $\lfloor i \cdot K_d / K_m \rfloor$ in the *DN*. Also, it is empirically observed that the ratio of total number of nodes in the *MN* to that in the *DN* is proportional to $\beta(K_m / K_d)$, where $\beta < 1$. This is because some of internal nodes are collapsed in the mapping process. Let $J_1 = \lceil (i\text{-}1)\cdot K_d / K_m \rceil$ and $J_2 = \lfloor i \cdot K_d / K_m \rfloor$. The number of nodes at level *i* in the *MN*, n_i, can be written in terms of the number of nodes at level *j* ($0 \leq j \leq K_d$) in the *DN*, denoted by m_j, as follows

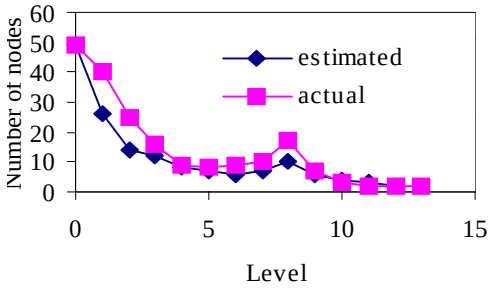
$$\left.\begin{aligned} n_0 &= m_0 \\ n_i &\approx \left\lceil \beta \frac{K_m}{K_d} \cdot \frac{\sum_{j=J_1}^{J_2} m_j}{(J_2 - J_1 + 1)} \right\rceil \qquad i = 1, 2, \cdots, K_m \end{aligned}\right\} \quad (4)$$

where β can be interpreted as the ratio of the total number of nodes in the *MN* and that in the *DN* when $K_m \approx K_d$.

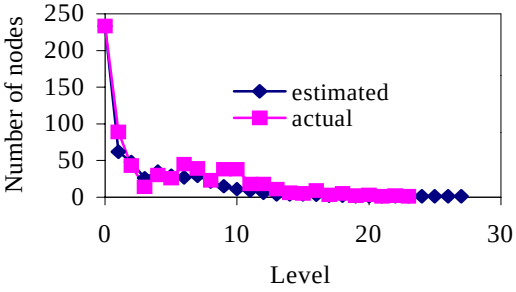
From our experiments with the MCNC'91 benchmark circuits, the value of α ranges from 1.1 to 1.5, and the value of β ranges from 0.5 to 1.0. Actually, α affects only the estimate of K_m, and β determines the estimated number of internal nodes in the *MN*. However, the power estimation is shown to be quite insensitive to both α and β (see [6] for the detailed discussion). In the following, we use $\alpha = 1.3$ and $\beta = 0.7$ unless otherwise stated. To demonstrate the accuracy of predicting node distribution using (4), we plot the node distributions of two example circuits (*apex7* and *C2670*) obtained from (4) and compare them with the actual node distributions in the *MN* of the circuits which are obtained from logic synthesis using area as the optimization objective. The node distribution curves are shown in Figure 1. It shows that there is a good agreement between the two node distribution curves.

3.3 Capacitance Distribution

Prediction of the area cost and hence the total capacitance of a Boolean network is an important step towards power estimation at RT-level. Because a given Boolean function can be implemented in different ways targeting different optimization goals, it is a difficult task to come up with accurate area estimation effectively at RT-level. Previous approaches on the area complexity are entropy-based [2,3]. These approaches break down when the number of inputs is large. Here we use the literal count in the decomposed network, *DN*, as a measure of the total capacitance of the



(a) circuit *apex7*



(b) circuit *C2670*

Figure 1 Node Distribution

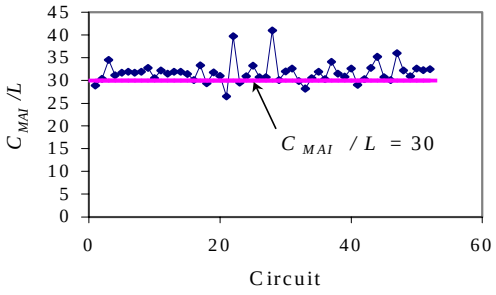


Figure 2 C_{MAI}/L Distribution for 52 Benchmarks

area-minimized mapped network. The reason is that the literal count corresponds closely to the number of transistor pair needed to implement the function as a static CMOS gate, and is a good area, hence, total capacitance estimator. Therefore, it is reasonable to assume that the total capacitance in *MAI*, denoted by C_{MAI}, is proportional to the literal count of the *DN*, denoted by *L*, i.e.

$$C_{MAI} = k_t L \tag{5}$$

where k_t is a proportionality constant that accounts for the gate library used and the effect of logic optimization. We tested the capacitance estimation using (5) on 52 MCNC'91 benchmark circuits with minimum area implementation using an industrial library. The results are shown in Figure 2 where C_{MAI} is in units of *fF* and $k_t \approx 30$ on average. It is shown that the approximation is reasonably accurate in modeling the total capacitance.

In order to extract the level capacitance distribution (i.e. the sum of node capacitances at each level) of the *MN* of the *MAI* from that of the *DN*, we consider the level distribution of the primary outputs of the *DN*. Intuitively, the larger the level number of a specific primary output in the *DN*, the more "*complex*" its logic function would be, and the more "*contribution*" it would make to the level capacitance at the related levels. Here, *contribution* means the number of gates (hence, the amount of capacitance) required to calculate the primary output. This is similar to the transitive fanin of the primary outputs. Since the *DN* is a 2-input gate decomposed network, the deeper the level of the primary output, the larger the transitive fanin cone and the higher the capacitance contribution would be. More specifically, let l_i be the level number of the *i*-th primary output in the *DN*. We assume its contribution to level *j* ($j = 0, 1, \cdots, l_i$), denote by A_{ij} , is given by

$$A_{ij} = k_c \cdot l_i \cdot 2^{-j} \tag{6}$$

where k_c is a proportionality constant that depends on the total capacitance, C_{MAI} , as will be seen later. Thus, we define the level capacitance at level *j* in the *DN* to be

$$C_j^D = \sum_i A_{ij} \qquad j = 0,1,\cdots,K_d \tag{7}$$

where the summation are taken over all primary outputs. In analogy to the derivation of (4) for the node distribution, the total capacitance at level *k* in the *MN*, denoted by C_k , can be written in terms of the total capacitance at level *j* in the *DN* (i.e., C^D_j) as follows

$$\left.\begin{aligned} C_0 &= \frac{1}{2}\left(C_0^D + C_1^D\right) \\ C_k &= \frac{\sum_{k=K_1}^{K_2} C_k^D}{(K_2 - K_1 + 1)} \qquad k = 1,2,\cdots,K_m \end{aligned}\right\} \tag{8}$$

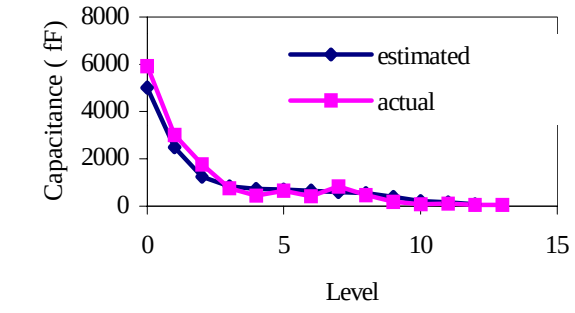
where $K_1 = \lceil (k\text{-}1)\cdot K_d / K_m \rceil$ and $K_2 = \lfloor k \cdot K_d / K_m \rfloor$. It is clear from (6)~(8) that C_k depends on k_c , and the value of k_c can be determined simply by setting

$$C_{MAI} = \sum_{k=1}^{K_m} C_k \tag{9}$$

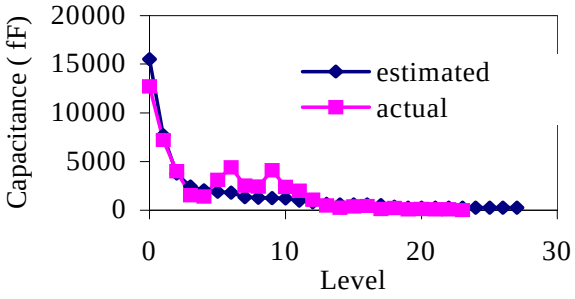
To verify the quality of the approximations made in our capacitance distribution model, the estimated level capacitance distribution against the actual distribution obtained from mapped circuit after logic synthesis is shown in Figure 3 for two example circuits. This comparison indicates that while the agreement is not perfect, our model is nevertheless very reasonable, especially considering that the level capacitances are obtained only from the distribution of primary outputs which is available after technology decomposition.

3.4 Entropy Distribution

From (3), the entropy distribution is another factor required for power estimation. In [2], it is shown that the entropy is varied quadratically with the circuit level. Considering



(a) circuit *apex7*



(b) circuit *C2670*

Figure 3 Capacitance Distribution

that [2] assumed the total number of nodes is $(PI+PO)\cdot(K_m+1)/2$, where *PI* and *PO* are the number of primary inputs and the number of primary outputs, respectively, we modify the entropy model of [2] as

$$H_i = \begin{cases} H_{in} & if \quad i = 0 \\ H_{out} & if \quad i = K_m \\ \dfrac{\sum_{j=0}^{K_m} n_j}{(PI+PO)\cdot(K_m+1)/2} \cdot & \\ \cdot\left(H_{out} + (H_{in} - H_{out})\cdot(1-\dfrac{i}{K_m})^2\right) & if \quad 0 < i < K_m \end{cases} \tag{10}$$

where H_i, H_{in} and H_{out} are the sums of node entropies at level i, *0* (primary inputs) and K_m (the largest level number in the *MN*), respectively. The term Σn_j in (10) represents the total number of nodes, which is intended for obtaining the same average node entropy in the modified entropy model as in the model of [2]. Also, from equation (10), H_i can be greater than n_i (the number of nodes at level i). However, this should not happen because n_i is the possible maximum of H_i from its definition. This suggests that the entropy model should be further modified. We change it by first computing H_i according to (10), then setting $H_i = n_i$ whenever $H_i > n_i$. For the computation of H_{in} and H_{out}, the *Monte Carlo* simulation method can be used [7].

4. POWER ESTIMATION FOR THE MINIMUM DELAY IMPLEMENTATION

4.1 Delay Model

Previous works have been done to estimate the circuit delay given a Boolean function [8,9]. [10] gives a comprehensive survey on this issue. It has been shown in [8] that the delay estimation depends on the number of logic levels and the load capacitance. Although the load capacitance is not available at the RT level, we can use the ratio of the literal count of a Boolean function to circuit depth of the *DN* of the function as a measure of the average load capacitance at each level. The reason is that the literal count of the *DN* is approximately proportional to the total capacitance for *MAI*, as shown in (5). We now define the ratio, L/K_d, to be the *circuit width* of the function, denoted by *W*. Intuitively, the circuit width is a measure of the average capacitive loading at each level of the technology-independent circuit. This leads us to the following simple delay model for the *MAI* of a circuit:

$$\begin{aligned} d_{MAI} &= K_d(a_1 + a_2 W) \\ &= a_1 K_d + a_2 L \end{aligned} \tag{11}$$

where a_1 and a_2 are technology specific parameters. For minimum delay implementation, it is natural to consider that the circuit with smallest circuit depth would result in minimum delay. However, this is not always true when the effect of the capacitive loading on the circuit delay is taken into consideration. If we focus only on the fanout optimization problem which tries to drive a certain fanout loading with a minimum delay, the delay can be modeled as a logarithmic function of the load capacitance, as explained in [9]. Based on these observations, we model the delay of the *MDI* of a circuit using the following equation

$$\begin{aligned} d_{MDI} &= K_d(b_1 + b_2 \log W) \\ &= K_d(b_1 + b_2 \log \frac{L}{K_d}) \end{aligned} \tag{12}$$

where b_1 and b_2 are technology-dependent parameters that are used to fit the abstract model to a specific implementation technology.

4.2 Estimating Capacitance

In order to estimate the total capacitance for the *MDI* of a circuit, consider the fanout optimization problem shown in Figure 4. Assuming that node *A* is driving a large fanout load f in Figure 4(a), the delay can be reduced by inserting $f^{1/2}$ buffers as shown in Figure 4(b). The delay difference between Figure 4(a) and 4(b) will be proportional to $(f - 2f^{1/2} - k_b)$, where k_b accounts for the intrinsic delay of the buffer. In other words, the *delay gain*, denoted by δd, can be expressed as $\delta d \propto (f - 2f^{1/2} - k_b)$. On the other hand, the increase in total load capacitance, denoted by δC, is proportional to $f^{1/2}$. Since the circuit width (L/K_d) can be used as an approximate measure of the capacitive loading, as described in Section 4.1, we obtain the following results

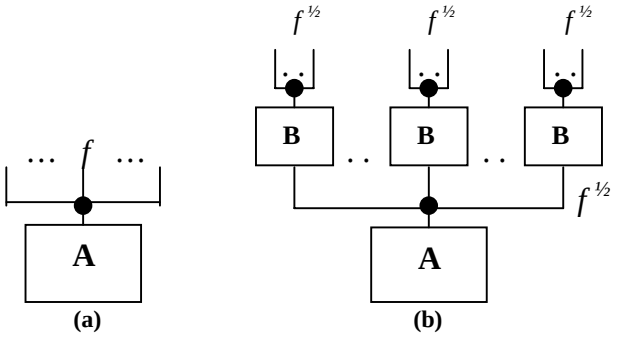


Figure 4 A Fanout Problem

by replacing f with L/K_d:

$$\delta d \propto (L/K_d - 2(L/K_d)^{1/2} - k_b \qquad (13a)$$

$$\delta C \propto (L/K_d)^{1/2} \qquad (13b)$$

Combining (13a) with (13b), we have

$$\left|\frac{\delta C}{\delta d}\right| = \frac{(L/K_d)^{1/2}}{k_1[L/K_d - 2(L/K_d)^{1/2}] + k_2} \qquad (14)$$

where k_1 and k_2 are technology specific constants. Thus, the total capacitance of the *MDI* can be estimated by

$$C_{MDI} = C_{MAI} + \left|\frac{\delta C}{\delta d}\right| \cdot (d_{MAI} - d_{MDI}) \qquad (15)$$

where d_{MAI} , d_{MDI} , C_{MAI} and $|\delta C/\delta d|$ are given by (11), (12), (5) and (14), respectively.

4.3 Power Approximation

We have observed that, for large circuits, the entropy (or, exactly, the average entropy per node) distributions along logic levels in the *MAI* and in *MDI* are very similar on average, although their numbers of logic levels and total capacitance values can be quite different. Furthermore, the shapes of the capacitance distribution with respect to the logic levels of the *MAI* and *MDI* of a circuit have also been found to be quite similar, in most cases, from our experiments. These empirical results suggest that the difference of the power consumption of different implementations for the same circuit depends mainly on the difference of their total capacitance. Thus the average power of the *MDI*, denoted by P_{MDI}, can be simply approximated by

$$P_{MDI} = \frac{C_{MDI}}{C_{MAI}} P_{MAI} \qquad (16)$$

where P_{MAI} represents the average power of the *MAI* of same circuit.

5. RESULTS AND DISCUSSIONS

The techniques described in this paper have been implemented. We carried out experiments using the MCNC'91 benchmark circuits. The input Boolean function is first decomposed using 2-input NAND gates and inverters, and power consumption of the *MAI* and *MDI* are estimated using the methods described in Sections 3 and 4. The actual minimum area and minimum delay implementations are generated under the *SIS* environment using *script.rugged* script for logic optimization. The power consumption of the mapped circuits are then estimated using a real delay gate-level power simulator, assuming a 5*V* supply voltage and 10*MHz* operating frequency.

Table 1 Regression Results

Parameter	a_1	a_2	b_1	b_2	k_1	k_2
Value	.010	.610	.452	.195	3.60e-5	4.60e-4

Table 2 Delay (in *ns*) and Total Capacitance (in *fF*) Estimation on Benchmarks

Example	MAI		MDI	
	actual delay/cap.	estimated delay/cap.	actual delay/cap.	estimated delay/cap.
9symml	17.1/ 17993	16.3/ 16950	13.2/ 24956	16.0/ 24012
C499	17.9/ 24838	19.9/ 24840	16.0/ 46204	17.7/ 50185
C1908	31.1/ 31473	30.5/ 28770	28.6/ 56285	30.0/ 51860
C2670	34.1/ 51166	36.5/ 47610	24.8/ 93258	31.4/ 98056
C3540	60.0/122783	70.3/115800	44.3/194649	46.6/234490
C6288	113/ 147000	117/ 149130	99.1/302208	102/ 304519
C7552	64.5/171307	85.4/163440	48.2/284461	41.6/305204
apex6	29.7/ 47992	29.9/ 46080	19.9/ 83652	21.8/ 95155
apex7	11.6/ 14687	13.0/ 13680	10.5/ 24677	12.7/ 20605
c8	12.1/ 7628	8.2/ 7710	7.6/ 13998	8.1/ 15086
example2	12.6/ 20304	13.2/ 19140	10.7/ 34127	10.2/ 39854
frg2	96.9/ 49762	83.1/ 49230	23.1/ 90455	24.8/101800
i3	4.7/ 18896	5.1/ 18780	3.8/ 34916	3.5/ 38155
i8	47.8/ 69899	33.2/ 68490	23.0/114930	19.5/130635

Before obtaining the experimental results, we derived technology-dependent parameters, such as a_1 and a_2 in (11), b_1 and b_2 in (12) and, k_1 and k_2 in (14) for the target gate library using linear regression technique on five benchmark circuits. Table 1 shows the regression results. Based on these parameters, we predicted the circuit delay using (11) and (12), estimated the total capacitance using (5) and (15), and compared them with the actual values obtained from the mapped circuits. The results are shown in Table 2. On average, our delay model produces the error of about 12.9% and 11.3% for the *MAI* and *MDI*, respectively. The average errors of our total capacitance estimation are 3.9% for the *MAIs* and 10.3% for the *MDIs* of the tested circuits.

To assess the accuracy of power estimation, we tested (3) and (16) on the benchmark circuits. The input signal probabilities are assumed to be 0.5. The output entropies are obtained by *Monte Carlo* simulation. The experimental results are reported in Table 3. The power consumption estimation for the *MAI*s based on the approach proposed in [2] is also included for comparison. Note that the approach in [2] does not include the estimation of the capacitance and here we use the total capacitances as estimated in (5) for power estimation. As shown in Table 3, for the *MAIs*, the average percentage error of our power estimation is 7.5%,

Table 3 Power Comparison on MAIs and MDIs of Benchmarks (in μw)

Example	MAI			MDI	
	sim.value	our est.	est. of [2]	sim. value	our est.
9symml	912.5	883.0	720.5	1315.9	1250.8
C499	1227.8	1370.8	1544.1	2230.0	2769.5
C1908	1383.6	1398.6	1724.6	2441.6	2521.1
C2670	2623.6	2630.2	1831.0	4722.6	5417.0
C3540	4905.7	5488.1	5979.3	7528.2	11113.1
C6288	8104.3	6833.6	9161.4	15666.6	13953.9
C7552	9118.7	8475.8	8871.8	14835.5	15827.6
apex6	2326.3	2458.0	2459.4	3840.9	5075.8
apex7	759.6	744.8	756.4	1237.8	1122.9
c8	404.7	426.4	461.4	700.0	834.4
example2	961.6	1058.0	768.0	1610.8	2203.0
frg2	2269.2	2613.9	2270.3	3895.2	5405.1
i3	922.1	1021.0	773.4	1769.5	2074.3
i8	3124.0	3226.2	3972.1	4998.5	6153.4
Avg. err.		**7.5%**	**15.9%**		**20.6%**

Table 4 Power Estimation on MAIs with Different Values of α and β (in μw)

Example	α = 1.3			β = 0.7	
	β = 0.7	β = 0.5	β = 1.0	α = 1.1	α = 1.5
9symml	883.0	873.1	877.0	883.0	868.3
C499	1370.8	1383.5	1360.8	1364.9	1384.6
C1908	1398.6	1424.1	1381.1	1379.3	1411.1
C2670	2630.2	2748.4	2511.1	2511.8	2699.9
C3540	5488.1	5535.8	5426.9	5477.7	5596.2
C6288	6833.6	6866.7	6849.5	6805.1	6812.8
C7552	8475.8	8667.6	8314.6	8292.7	8665.6
apex6	2458.0	2567.6	2373.1	2379.1	2513.2
apex7	744.8	760.6	723.2	724.8	746.5
c8	426.4	435.6	405.1	409.9	428.6
example2	1058.0	1071.7	1018.3	1024.5	1067.4
frg2	2613.9	2665.5	2573.2	2572.6	2703.7
i3	1021.0	1068.2	975.5	1018.5	1059.8
i8	3226.2	3261.1	3198.2	3089.2	3330.4

while that of [2] produces 15.9% error on average. The error in power estimation of the *MDI* is in general higher than that of the *MAI*. An average 21% error is reported. It is because the capacitance estimation model is not as accurate as that for the *MAI*. The results show in cases where the estimated capacitance distribution and the actual capacitance distribution differ by a significant amount (e.g. *C3540*), the power estimation will lead to large errors.

Finally, we also tested the impact of α and β introduced in Section 3.2 on our power estimation by using different values of α ranging from 1.1 to 1.5 and different values of β ranging from 0.5 to 1.0. Table 4 reports the testing results for the MAIs. It can be seen from this table that the power estimation variations are less than 5% in most cases, and less than 10% in the worst case for all tested circuits. This indicates that our power estimation is quite insensitive to both α and β.

6. CONCLUSION

We have described an entropy-based RT-level power estimation technique using technology decomposition. We focused on generating the power-area-delay trade-off curve by estimating power consumption and delay values of two different circuit implementations, the minimum area implementation and minimum delay implementation. Our approach takes into account the structural information such as the node, capacitance and entropy distribution of a given circuit and hence can achieve higher accuracy. From the experimental results, we have shown that the proposed technique is a significant improvement over previous techniques.

7. REFERENCES

[1] Marculescu D., Marculescu R. and Pedram M.. Information theoretic measures for power analysis. IEEE Trans. on CAD, June 1996, 15(6), 599-610.

[2] Nemani M. and Najm F.N.. Towards a high level power estimation capability. IEEE Trans. on CAD, June 1996, 15(6), 588-598.

[3] Cheng K.T. and Agrawal V.D.. An entropy measure for the complexity of multi-output Boolean functions. In Proc. DAC, 1990, 302-305.Fujita M. and Murgai R.. Delay estimation and optimization of logic circuits: A survey. In Proc. ASP-DAC, 1997, 25-30.

[4] Cook R.W. and Flynn M.J.. Logical network cost and entropy. IEEE Trans. on Computers, Sept.1973, 22(9), 823-826.

[5] Veendrick H.J.M.. Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits. IEEE J. Solid State Circuits, Aug. 1984, 19, 465-473.

[6] Chen C. and Tsui C.. RT-level power estimation using technology decomposition. In Proc. CSIA Chip Technology, April 1998, 205-211.

[7] Najm F.N.. Statistical estimation of the signal probability in VLSI circuits. Coord. Sci. Lab., Urbana, IL, Tech. Rep. #UILU-ENG-93-2211, Apr. 1993.

[8] Brayton R.K., Rudell R., Sangiovanni-Vincentelli A., and Wang A.R.. MIS: A multiple-level logic optimization system. IEEE Trans. on CAD, Nov. 1987, 6(6), 1062-1081.

[9] Wallace D.E. and Chandrasekhar M.S.. High level delay estimation for technology-independent logic equations. In Proc. ICCAD, 1990, 188-191.

[10] Fujita M. and Murgai R.. Delay estimation and optimization of logic circuits: A survey. In Proc. ASP-DAC, 1997, 25-30.