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

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

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

\[
P_{\text{avg}} = \frac{1}{4} f_{\text{crit}} V_{dd}^2 \sum g C_{\text{load}}(g) h(g)
\]

Obviously, the exact values of \( C_{\text{load}}(g) \) 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

\[
P_{\text{avg}} = \frac{1}{4} f_{\text{crit}} V_{dd}^2 \sum_{i=0}^{K} \left( \frac{C_i}{n_i} \cdot H_i \right)
\]

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 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 \( [(i-1) \cdot K_d / K_m] \) to level \( [i \cdot K_d / K_m] \) 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 \( J_i = [(i-1) \cdot K_d / K_m] \) and \( J_2 = [i \cdot K_d / K_m] \). 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

\[
n_0 = m_0

n_i = \frac{\beta}{K_d} \cdot \frac{\sum_{j=j_i}^{j_2} m_j}{(J_2 - J_1 + 1)} \quad i = 1, 2, \ldots, K_m
\]

where \( \beta \) can be interpreted as the ratio of the total number of nodes in the MN and that in the DN when \( K_m = K_d \).

From our experiments with the MCNC’91 benchmark circuits, the value of \( \alpha \) ranges from 1.1 to 1.5, and the value of \( \beta \) ranges from 0.5 to 1.0. Actually, \( \alpha \) affects only the estimate of \( K_m \), and \( \beta \) determines the estimated number of internal nodes in the MN. However, the power estimation is shown to be quite insensitive to both \( \alpha \) and \( \beta \) (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
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 \( j \) (\( j = 0, 1, \ldots, l_i \)), denoted by \( A_{ij} \), is given by

\[
A_{ij} = k_j \cdot l_i \cdot 2^{-j}
\]

where \( k_j \) 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 \( j, j = 0,1,\ldots, K_d \), 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., \( C^D \)) as follows

\[
C_k = \frac{1}{2} \left( C_0^D + C_k^D \right)
\]

\[
C_k = \frac{\sum_{i=1}^{K_k} C_i^D}{K_2 - K_1 + 1} \quad k = 1, 2, \ldots, K_m
\]

where \( K_i = \lceil (k-1) K_d / K_m \rceil \) and \( K_d = \lceil K_d / K_m \rceil \). It is clear from (6)~(8) that \( C_j \) depends on \( k_j \), and the value of \( k_j \) can be determined simply by setting

\[
C_{MAI} = \sum_{k=1}^{K_m} C_k
\]

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
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

\[
H_i = \begin{cases} 
H_m & \text{if } i = 0 \\
H_{out} & \text{if } i = K_m \\
\sum_{j=1}^{K_m} n_j \left( \frac{PI+PO}{2} \right) - \frac{1}{2} \left( \frac{H_{out} + (H_m - H_{out}) \cdot (1 - \frac{i}{K_m})^2}{2} \right) & \text{if } 0 < i < K_m 
\end{cases}
\]

(10)

where \(H_m\), \(H_{out}\) and \(H_{in}\) 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 \(\sum_{j=1}^{K_m} n_j\) 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 \(H_{out}\) and \(H_{in}\), 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:

\[
d_{MAI} = K_d(a_1 + a_2 W)
\]

\[
d_{MAI} = a_1 K_d + a_2 L
\]

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

\[
d_{MDI} = K_d(b_1 + b_2 \log W)
\]

\[
d_{MDI} = K_d(b_1 + b_2 \log \frac{L}{K_d})
\]

(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 \(\Delta d\), 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 \(\Delta C\), 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 3 Capacitance Distribution

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 \(\Delta d\), 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 \(\Delta C\), 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.
by replacing $f$ with $L/K_d$:

$$\delta d = (L/K_d - 2(L/K_d)^{1/2}) - k_1$$  \hspace{1cm} (13a)

$$\delta C = (L/K_d)^{1/2}$$  \hspace{1cm} (13b)

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

$$\delta d = k_1 (L/K_d - 2(L/K_d)^{1/2}) + k_2$$  \hspace{1cm} (14)

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

$$C_{MDI} = C_{MAI} + \frac{\delta C}{\delta d} (d_{MAI} - d_{MDI})$$  \hspace{1cm} (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

$$P_{MDI} = \frac{C_{MDI}}{C_{MAI}} P_{MAI}$$  \hspace{1cm} (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 5V supply voltage and 10MHz operating frequency.

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 MAIs 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%,
while that of [2] produces 15.9% error on average. The error in power estimation of the MAI 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 MAI capacitance distribution differ by a significant amount (e.g., estimated capacitance distribution and the actual capacitance distribution). This indicates that our power estimation is quite insensitive to both $\alpha$ and $\beta$.

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