

A Static Power Estimation Methodology for IP-Based Design

Xun Liu Marios C. Papaefthymiou
Department of Electrical Engineering
and Computer Science
University of Michigan
Ann Arbor, Michigan 48109

Abstract

This paper proposes a novel system-level power estimation methodology for electronic designs consisting of intellectual property (IP) components. Our methodology relies on analytical output and power macromodels of the IP blocks to estimate system dissipation without performing any simulation. We derive upper bounds on the estimation error of our methodology and demonstrate the relation of this error to the sensitivities of the macromodeling functions. For circuits without feedback, we give a sufficient condition for the worst-case power estimation error to increase only linearly with the length of the IP cascades. We also give a tighter sufficient condition that ensures error boundedness in IP systems of any topology. Experiments with signal processing and data encryption systems validate the accuracy and efficiency of our approach. For designs of up to 576 IP blocks, power estimates are obtained within 0.2 seconds. In comparison with switch-level simulation results, the average error of our power estimates is 7.3%.

1. Introduction

Low power consumption has become one of the most important objectives in electronic design due to continuously growing portability considerations and reliability concerns. A key challenge in the design of low power systems is the fast and accurate estimation of power dissipation. System designers need to evaluate power dissipation at an early design stage, when detailed knowledge of physical design is still not available [7, 10].

Macromodeling is a promising approach to the problem of high-level power estimation. The basic idea behind power macromodeling is to generate a mapping between the power dissipation of a circuit and certain statistics of its input signals such as the average signal probability or average transition density [9]. This abstract mapping hides the implementation details of a circuit while allowing its eval-

uation under the workload of the specific application it is used for. Although power macromodeling has been proved to be effective for individual intellectual property (IP) components, it is inadequate for modeling systems of IP components. In such IP-based designs, multiple IP blocks are connected together. The application of power macromodeling on each IP block of the system requires knowledge of the signal statistics among these blocks. To obtain this information, architects must perform extensive functional simulations, an often time-consuming task that significantly limits the potential for exploring alternative system implementations.

In this paper, we present a novel power estimation methodology for IP-based designs that relies on power and output macromodeling. The proposed methodology proceeds in two phases. In the characterization phase, power and output macromodels are obtained for each IP block using simulation. In the estimation phase, an iterative algorithm computes the signal statistics of the inter-IP nodes using the output macromodels. The dissipation of the entire system is subsequently obtained by applying these statistics to the power macromodels of the individual blocks. The estimation phase of our methodology is *fully static*, that is, it does not perform any simulations. Thus, once the IP blocks have been characterized, the power dissipation of alternative system architectures can be obtained extremely fast. To our knowledge, our scheme is the first-ever fully-static macromodeling methodology proposed for estimating the power dissipation of IP systems.

We give a rigorous analysis that relates the sensitivity of the macromodel functions with the error of our power estimation algorithm. A surprising result of our analysis is that when the macromodel functions are relatively smooth, the error does not accumulate at exponential rates and remains bounded even around loops. Specifically, for IP-based designs with no feedback loops, we give sufficient conditions on the macromodel functions for the worst-case error to accumulate only linearly along cascades. For IP systems of general topology that may include loops, we give a tighter

sufficient condition that guarantees the boundedness of the estimation error.

Our proposed methodology was evaluated on a collection of IP-based designs, including a variety of digital signal processing systems and a standard encryption/decryption system. The systems in our test suite comprised up to 576 IP blocks. In comparison with switch-level simulation, the average power estimation error of our methodology was 7.3% with a standard deviation of 0.08. For more than 91% of the obtained estimates, the error was less than 15%. In general, the error of our methodology was lower on circuits with feedback than on circuits without feedback. For all designs, power estimates were obtained within at most 0.2 seconds on a Sun UltraSparc II workstation while the corresponding switch-level simulation took several hours.

Previous macromodeling research has focused on individual IP blocks. A look-up table (LUT) approach was introduced in [5] and improved in [1]. The LUT stores power estimates for equally-spaced discrete values of the input statistics. Interpolation is used to obtain estimates for statistics not in the LUT. The notion of power sensitivity was introduced in [3, 4] for improving the accuracy of interpolation. In this approach, discrete planes are used to approximate the power surface and reduce the large memory requirements of the LUT. Analytical power macromodeling uses mathematical expressions to map input signal statistics to power dissipation, thus avoiding the space cost of LUT approaches [2, 6]. The idea of output macromodeling, that is, the mapping of input signal statistics to output signal statistics, was first introduced in [2] to predict output signal statistics from input. This concept was only discussed in the context of a single IP block, however, without investigating its application to IP-based system design.

The remainder of this paper has 5 sections. In Section 2, we give background of analytical power and output macromodeling. In Section 3, we present our static power estimation methodology for IP-based designs and describe our iterative estimation algorithm. The error and convergence rate of our algorithm are analyzed in Section 4. Experimental results are presented in Section 5. Section 6 summarizes our paper and gives directions for further research.

2. Background

In analytical power macromodeling, a function g maps the space of input signal statistics to the power dissipation of a circuit. Similarly, in analytical output macromodeling, a function f is used to map input to output signal statistics. When the input parameters of all macromodeling functions are solely input signal statistics, the computation of power estimates or output statistics is a straightforward and fast function evaluation. The key challenges in analytical macromodeling are the choice of appropriate input param-

eters for the macromodels and the derivation of the macromodel functions.

The estimation methodology we describe in this paper chooses the average input signal probability P_{in} , the average input transition density D_{in} , and the input spatial correlation S_{in} as the input parameters for the macromodels. Given an IP block with M binary inputs and a binary input stream $B = \{(b_{11}, b_{12}, \dots, b_{1M}), \dots, (b_{N1}, b_{N2}, \dots, b_{NM})\}$ of length N , these metrics are defined as follows [2]:

$$P_{in} = \frac{\sum_{j=1}^M \sum_{k=1}^N b_{kj}}{M \times N}, \quad (1)$$

$$D_{in} = \frac{\sum_{j=1}^M \sum_{k=1}^{N-1} b_{kj} \oplus b_{k+1j}}{M \times (N-1)}, \quad (2)$$

$$S_{in} = \frac{\sum_{j=1}^M \sum_{k=1, k \neq j}^M \sum_{n=1}^N b_{nk} \oplus b_{nj}}{N \times M \times (M-1)}. \quad (3)$$

To obtain the power function g of a given IP block, the component is first simulated under sample input streams with various P_{in} , D_{in} , and S_{in} . The set of power dissipation points \mathcal{P} obtained by this procedure is subsequently curve-fitted to derive an analytical expression g using a minimal mean-square error criterion so that

$$\mathcal{P} = g(P_{in}, D_{in}, S_{in}). \quad (4)$$

Power sensitivity to an input metric K shows how K affects \mathcal{P} and is defined as:

$$\lim_{\Delta K \rightarrow 0} \frac{\Delta \mathcal{P}}{\Delta K} \quad (5)$$

Given an analytical expression g , the power sensitivity can be calculated by the partial derivative of g .

In output macromodeling, a method similar to the computation of g can be used to map input to output signal statistics. In the characterization step, functional simulation of a block is performed with different input sequences to obtain data points for the metrics P_{out} , D_{out} , and S_{out} . Using a minimal mean-square error criterion, analytical functions f_1 , f_2 , and f_3 are derived so that

$$P_{out} = f_1(P_{in}, D_{in}, S_{in}), \quad (6)$$

$$D_{out} = f_2(P_{in}, D_{in}, S_{in}), \quad (7)$$

$$S_{out} = f_3(P_{in}, D_{in}, S_{in}). \quad (8)$$

Similar to power sensitivity, the sensitivity of an output metric with respect to P_{in} , D_{in} , or S_{in} is defined as the partial derivative of the corresponding f_i .

3. Power Estimation Algorithm

Our power estimation algorithm combines output and power macromodeling to estimate the power dissipation of

```

HIPPE ( $\mathcal{S}, \mathcal{T}$ )
1 sort all IP blocks using breadth-first search
2 initialize inter-IP signal statistics to arbitrary values
3 repeat
4   for each IP block  $n$  in sorted order
5     do  $P_{out} = f_1(P_{in}, D_{in}, S_{in})$ 
6        $D_{out} = f_2(P_{in}, D_{in}, S_{in})$ 
7        $S_{out} = f_3(P_{in}, D_{in}, S_{in})$ 
8 until (reached max iteration count
        or all inter-IP signal statistics converged)
9  $power = 0$ 
10 for each IP block  $n$ 
11   do  $power = power + (\text{power dissipation of } n)$ 
12 return  $power$ 

```

Figure 1. Algorithm HIPPE for estimating the power dissipation of an IP system \mathcal{S} for a set \mathcal{T} of primary input statistics.

an IP-based design. Given a system \mathcal{S} consisting of IP components and signal statistics \mathcal{T} at the primary inputs of \mathcal{S} , our algorithm estimates the power dissipation of \mathcal{S} under \mathcal{T} . The input \mathcal{S} includes the individual power and output macromodels of each IP block in the system. It also includes the interconnection topology of the system. The input \mathcal{T} gives the metrics P , D , and S at the primary inputs.

The pseudocode of our high-level IP system power estimation algorithm *HIPPE* is given in Figure 1. Algorithm *HIPPE* first sorts the IP blocks of the given system \mathcal{S} in the order of their shortest distance from the primary inputs. It then initializes the metrics P , D , and S of the inter-IP signals to arbitrary values. As we argue in Section 4, this choice does not affect the error bound of the derived estimates.

The main processing step of Algorithm *HIPPE* is the **repeat** loop in lines 3–8. In each iteration of the loop, the output signal statistics P_{out} , D_{out} , and S_{out} of each IP block n are updated by applying the current input statistics of n to the block’s macromodel functions f_1 , f_2 , and f_3 . (For simplicity, we do not index the macromodel functions over the set of IP blocks, even though different IP blocks will in general have different macromodel functions.) Since the output signals of a block n are a subset of the input signals of some other block j , each update of n may trigger a new update for j at a later iteration. These iterative updates continue until the signal metrics converge or a maximal number of iterations is reached. The bound on the number of iterations depends on the topology of the system and the macromodels of the individual IP blocks and can be calculated using the error analysis formulas given in Section 4.

The power dissipation of the entire system is computed in the loop of lines 10–11, where the final signal statistics are applied to the power macromodel functions of the individual IP blocks. Line 12 returns the total power dissipation of the system, and the algorithm terminates.

4. Algorithm Analysis

In this section, we analyze the estimation accuracy and convergence rate of Algorithm *HIPPE*. We first introduce some notation and definitions used in our analysis. We subsequently proceed to analyze the accuracy of the algorithm first for linear cascades and then for general topologies.

Without loss of generality, we assume that 3 metrics (P , D , and S) are used to characterize binary sequences. For IP block n , the vectors $\vec{I}_n = (i_{n1}, i_{n2}, i_{n3})$ and $\vec{O}_n = (o_{n1}, o_{n2}, o_{n3})$ denote the input and output sequence statistics, respectively. Moreover, the vector $\vec{F}_n = (f_{n1}, f_{n2}, f_{n3})$ denotes its output macromodel functions, where f_{ni} is a scalar function of \vec{I}_n that predicts o_{ni} . The power macromodel is a scalar function g_n of \vec{I}_n that predicts the power dissipation \mathcal{P}_n . The 3-dimensional vector of 1’s is denoted by $\vec{1}_3$.

The comparison between two vectors \vec{A} and \vec{E} is defined as:

$$\vec{A} \leq \vec{E} \Leftrightarrow \forall j, a_j \leq e_j.$$

The absolute value matrix $|\overline{\overline{R}}|$ is obtained by taking the absolute value $|r_{ij}|$ of each element r_{ij} in the matrix $\overline{\overline{R}}$. The function $H(\overline{\overline{R}}, \overline{\overline{Q}})$ gives the maximal absolute difference between the corresponding elements of $\overline{\overline{R}}$ and $\overline{\overline{Q}}$:

$$\forall j \in \{1, m\}, k \in \{1, l\}, H(\overline{\overline{R}}, \overline{\overline{Q}}) = \max |r_{jk} - q_{jk}|,$$

where $\overline{\overline{R}}, \overline{\overline{Q}}$ are two $m \times l$ matrices.

The partial derivative vector of a scalar function $x(\vec{I})$, where $x \in \{g_n, f_{ni}\}$, is defined as:

$$\vec{x}'(\vec{I}) = (\partial x / \partial i_1, \partial x / \partial i_2, \partial x / \partial i_3)$$

The partial derivative matrix of a function vector $\vec{F}_n = (f_{n1}, f_{n2}, f_{n3})$ is defined as:

$$\overline{\overline{F}}'_n = (\vec{f}'_{n1}, \vec{f}'_{n2}, \vec{f}'_{n3})^T$$

The absolute sum of power macromodeling sensitivities, $|\vec{g}'_n| \cdot \vec{1}_3$, signifies how the input parameters affect the power estimate. Similarly, the vector $|\overline{\overline{F}}'_n| \cdot \vec{1}_3$ describes how the input parameters affect the estimates of the output statistics. It is assumed that for each IP block n , there exists $\epsilon_n > 0$ such that

$$H(\vec{O}_n, \vec{F}_n(\vec{I}_n)) \leq \epsilon_n \quad (9)$$

$$H(\mathcal{P}_n, g_n(\vec{I}_n)) \leq \epsilon_n \quad (10)$$

4.1. Linear Cascades

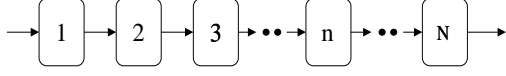


Figure 2. N cascaded IP blocks

A cascade of N IP blocks is shown in Figure 2. In Algorithm HIPPE, the input statistics of block n are computed by the output macromodeling functions of block $n - 1$. Due to the errors in the input statistics, the output statistics and power estimation errors of block n are not bounded by ϵ_n . Theorem 1 gives a sufficient condition for the worst-case error to increase only linearly with the number of cascaded blocks. Intuitively, the theorem states that if the sensitivities of the output and the power macromodels are low, then the errors are not amplified as they propagate along the cascades.

Theorem 1 *Let \mathcal{S} be an IP system with N cascaded blocks. For each block n , let ϵ_n be the maximal error of the output and the power macromodeling functions. The worst-case power estimation error of Algorithm HIPPE is proportional to the number of blocks cascaded from the primary inputs if the power and the output sensitivities satisfy the conditions*

$$\left| \vec{g}_n \right| \cdot \vec{\mathbb{I}}_3 \leq 1, \quad (11)$$

$$\left| \overline{\vec{F}}_n \right| \cdot \vec{\mathbb{I}}_3 \leq \vec{\mathbb{I}}_3. \quad (12)$$

Proof. Let \vec{O}_n and \vec{I}_n represent the statistics of the real signals at the output and input of block n . We first use induction to prove the following bound on the output signal statistics of the inter-IP nodes:

$$\forall n \in \{1, N\}, \quad H(\vec{O}_n, \vec{F}_n(\vec{F}_{n-1})) \leq n \cdot \epsilon, \quad (13)$$

where $\epsilon = \max \{\epsilon_j \mid j \in \{1, N\}\}$ and $\vec{F}_0 \stackrel{\text{def}}{=} \vec{I}_1$.

The base case for $n = 1$ is straightforward. Next, we assume that for $n \leq j$, we have

$$H(\vec{O}_n, \vec{F}_n(\vec{F}_{n-1})) \leq n \cdot \epsilon, \quad (14)$$

and prove that Inequality (13) holds for $n = j + 1$. Since block $(j + 1)$ uses the output estimates of block j as its input parameters, using the first-order Taylor expansion approximation

$$\vec{F}_{j+1}(\vec{F}_j) = \vec{F}_{j+1}(\vec{I}_{j+1}) + \overline{\vec{F}}_{j+1}(\vec{I}_{j+1}) \cdot (\vec{O}_j - \vec{F}_j) \quad (15)$$

we have

$$\left| \vec{O}_{j+1} - \vec{F}_{j+1}(\vec{F}_j) \right|$$

$$\begin{aligned} &= \left| \vec{O}_{j+1} - \vec{F}_{j+1}(\vec{I}_{j+1}) - \overline{\vec{F}}_{j+1}(\vec{I}_{j+1}) \cdot (\vec{O}_j - \vec{F}_j) \right| \\ &\leq \epsilon_{j+1} \cdot \vec{\mathbb{I}}_3 + \left| \overline{\vec{F}}_{j+1}(\vec{I}_{j+1}) \right| \cdot H(\vec{O}_j, \vec{F}_j) \cdot \vec{\mathbb{I}}_3 \\ &\leq \epsilon_{j+1} \cdot \vec{\mathbb{I}}_3 + \left| \overline{\vec{F}}_j(\vec{I}_{j+1}) \right| \cdot j \cdot \epsilon \cdot \vec{\mathbb{I}}_3 \\ &\leq (j + 1) \cdot \epsilon \cdot \vec{\mathbb{I}}_3. \end{aligned} \quad (16)$$

From Inequality (16), it follows that Inequality (13) holds for $n = j + 1$, and the proof is complete.

The proof for the bound of the power estimation error is straightforward and can be derived by combining the first-order Taylor expansion approximation of g_n with Inequalities (11) and (13). \square

4.2. General Topologies

We now extend our analysis to IP systems of general topologies. The following theorem shows that if output sensitivities are low, then for any system \mathcal{S} , the power estimation error of Algorithm HIPPE is bounded.

Theorem 2 *Let \mathcal{S} be an IP system of arbitrary topology. For each block n , let ϵ_n be the maximal error of the output and power macromodeling functions. The power estimation error of Algorithm HIPPE is bounded if the power sensitivities $\left| \vec{g}_n \right|$ are bounded for all n , and the output sensitivity of each block n satisfies the inequality*

$$\left| \overline{\vec{F}}_n \right| \cdot \vec{\mathbb{I}}_3 \leq \delta \cdot \vec{\mathbb{I}}_3, \quad (17)$$

where δ is a real number such that $0 < \delta < 1$.

Proof. We prove that if the estimation errors of the signal statistics on the inter-IP nodes do not exceed $\epsilon/(1 - \delta)$, then they stay within this bound throughout the execution of Algorithm HIPPE. We also show that if the initial errors exceed $\epsilon/(1 - \delta)$, then they approach this bound after each iteration of the algorithm.

We use the first-order Taylor expansion approximation to relate the errors of the input signals before an iteration of the **repeat** loop in Algorithm HIPPE with the errors of the output signals after the iteration. This relation is given by the inequality

$$\forall n, \quad \vec{\epsilon}_n^{\vec{o}} \leq \epsilon_n \cdot \vec{\mathbb{I}}_3 + \left| \overline{\vec{F}}_n \right| \cdot \vec{\epsilon}_n^{\vec{i}}, \quad (18)$$

where $\vec{\epsilon}_n^{\vec{o}}$ and $\vec{\epsilon}_n^{\vec{i}}$ are the signal statistics estimation errors at the output and input of block n , respectively. Without loss of generality, we assume that before the j th iteration, the maximum error of the output signal statistics is $k_j \cdot \epsilon/(1 - \delta)$,

where $k_j > 0$, and $\epsilon = \max_n \{\epsilon_n\}$. For all n , Inequality (18) implies that

$$\begin{aligned} \epsilon_n^{\vec{\delta}} &\leq \epsilon \cdot \vec{I}_3 + \left| \overline{F'_n} \right| \cdot \frac{\epsilon}{1-\delta} \cdot k_j \cdot \vec{I}_3 \\ &\leq \epsilon \cdot \vec{I}_3 + \delta \cdot \frac{\epsilon}{1-\delta} \cdot k_j \cdot \vec{I}_3 \\ &= \frac{1 + (k_j - 1) \cdot \delta}{1 - \delta} \cdot \epsilon \cdot \vec{I}_3. \end{aligned} \quad (19)$$

If $k_j \leq 1$, Inequality (19) implies that for all n , we have

$$\epsilon_n^{\vec{\delta}} \leq \frac{\epsilon}{1-\delta} \cdot \vec{I}_3. \quad (20)$$

Therefore, if the maximum error of the output statistics is at most $\epsilon/(1-\delta)$, then it will never exceed $\epsilon/(1-\delta)$.

If $k_j > 1$, Inequality (19) implies that for all n , we have

$$\epsilon_n^{\vec{\delta}} \leq \frac{k_j \cdot \epsilon}{1-\delta} \cdot \left(\delta + \frac{1-\delta}{k_j} \right), \quad (21)$$

where $\delta + (1-\delta)/k_j < 1$. Inequality (21) shows that if the original maximum error of the output statistics exceeds $\epsilon/(1-\delta)$, it will decrease after each iteration, eventually approaching $\epsilon/(1-\delta)$. \square

From Inequality (21) in Theorem 2, it follows that after j iterations of the **repeat** loop in Algorithm HIPPE, we have

$$k_{j+1} - 1 \leq (k_1 - 1) \cdot \delta^j.$$

At that point, the difference between the output error and $\epsilon/(1-\delta)$ is proportional to $k_{j+1} - 1$. Since the upper bound of $k_{j+1} - 1$ decreases at an exponential rate with the iteration count j , the output estimation error of Algorithm HIPPE converges quickly to the bound $\epsilon/(1-\delta)$.

For the simple case of a system that consists of a single loop, we can relax Inequality (17) and still guarantee that the estimation error remains bounded. Specifically, if the IP blocks along the loop are labeled from 1 to N , we can replace Inequality (17) by the inequality

$$\left(\prod_{j=1}^N \left| \overline{F'_j} \right| \right) \cdot \vec{I}_3 \leq \delta \cdot \vec{I}_3,$$

where $0 < \delta < 1$.

5. Experimental Results

In this section, we discuss some empirical results from the application of our power estimation methodology on several IP-based designs. We first describe the flow of our characterization and estimation procedure. We subsequently describe our test circuits. Finally, we discuss our

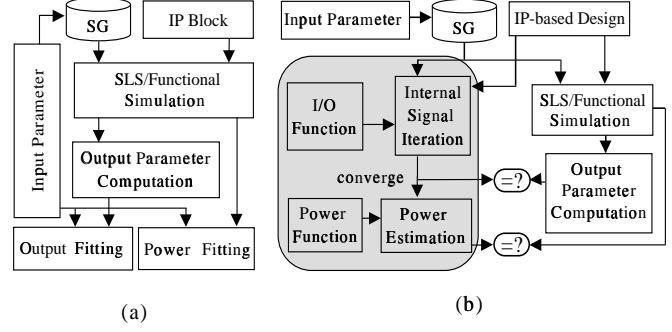


Figure 3. Experimental procedure: (a) Characterization (b) Estimation

power estimation results in comparison with those obtained by switch-level simulations.

Our power estimation framework consists of two phases, the characterization phase and the estimation phase. Figure 3(a) shows the characterization phase. The first step in the characterization of an IP block is the generation of input sequences of different and diverse statistics. To that end, we have developed a sequence generator (SG) based on a Markov chain model [8]. SG is capable of generating an input sequence of given P_{in} , D_{in} , and S_{in} or report that no such sequence exists. It furthermore guarantees the uniformity of the signal statistics for the entire length of the sequence. In our experiments, we generated 238 sequences with P_{in} , D_{in} and S_{in} evenly distributed in the normalized 3-dimensional space. For all three parameters, the granularity was 0.1. Each sequence had 2,000 vectors.

After the generation of the input signals, the next step of the characterization phase is circuit simulation. Our setup relied on Delft University's switch-level simulator SLS to calculate the power dissipation of each IP block. In conjunction with SLS, we used the `scmos_n` transistor model, which has the same model for transistor resistance and parasitic capacitance as SPICE. Our switch-level simulations considered the contribution of transient effects, such as glitches, to the power dissipation of a design. The output signal statistics were computed using functional simulation.

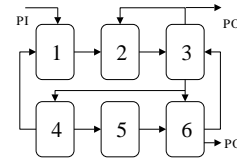


Figure 4. An IP system.

The final step of our characterization phase was the derivation of output and power macromodel functions by curve fitting. To improve the accuracy of analytical macro-

modeling, we partitioned the input signals of each block into groups, based on the logic connections among the IP blocks. This partitioning increases the characterization dimension of the input stream and thus results in more accurate estimates. Our input partitioning scheme can be described using the simple IP system in Figure 4. The inputs of block 3 can be partitioned into two groups, for example, one from block 2 and one from block 6. The outputs of block 3 can also be divided into two groups, because they are connected to block 2 and blocks 6 and 4, respectively. In our experiments, we chose the second degree complete polynomial as the template for output fitting and power fitting. This choice of the macromodel function template is based on [2] and considerations for low characterization complexity. For an IP block with n input groups, each macromodel function required the calculation of $1 + C_{3n}^1 + C_{3n}^2$ coefficients, where C is the binomial coefficient function. In our experiments, with I/O partitioning, the mean error rate of all analytical macromodel functions was at most 4%.

The estimation phase of our experiments is described in Figure 3(b). The shaded block highlights the steps of this phase that are specific to the static estimation algorithm we propose in this paper. Given an IP system, SG generates sequences as primary inputs. The signal statistics of the internal nodes among the IP blocks are then computed iteratively using the **repeat** loop in Algorithm HIPPE. These statistics are applied to the power macromodel functions to compute the power dissipation of the system. To evaluate the accuracy of our estimation methodology, we independently use SLS to simulate the entire system at the switch level. The inter-IP signal statistics and the overall power dissipation are then compared with the results obtained using our static estimation algorithm.

For our experiments, we designed seven IP blocks at the transistor level in static CMOS technology. Our suite of IP blocks comprised a ripple-carry adder, a subtractor, and an amplitude modulator, elements frequently used in digital signal processing systems. The schematics of these circuits were taken from [12]. Our IP blocks collection also included components of an encryptor/decryptor system: a modulo-2 adder, a substitution block, a permutation block, and an extension block. The functionalities of these blocks are described in the Data Encryption Standard [11]. Their synthesis was performed using standard cells and the Berkeley SIS tools. The adders, subtractors, and modulo-2 adders had their inputs partitioned into two groups. All other IP blocks has their inputs grouped together. The outputs of each block also formed only one group. Using the 7 IP blocks, we built several IP systems of up to 576 IP blocks. Our designs included different types of Hadamard transform modules (HT), fast Fourier transform modules (FFT), infinite impulse response filters (IIR), finite impulse

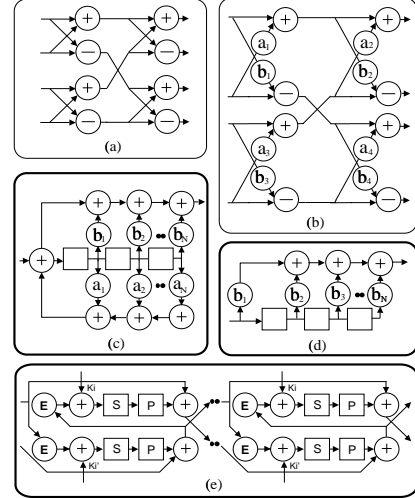


Figure 5. Sample IP systems from our test suite: (a) 2-bit HT (b) 2-bit FFT (c) IIR (d) FIR (e) DES

response filters (FIR), and a data encryption standard encryptor/decryptor (DES). Figure 5 shows the structure of our systems for low-bit, small-size modules.

The results from the application of our static power estimation methodology to the IP systems in our test suite are given in Table 1. Columns 1 and 2 give the name and size of each system. Columns 3–5 give the average relative error in the estimation of the inter-IP signal statistics P , D , and S , in comparison with functional simulation results. Column 6 gives the average relative error $\Delta\mathcal{P}$ in the estimation of the power dissipation, in comparison with SLS simulation results. Column 7 shows the standard deviation $STD(\mathcal{P})$ of the power estimation error. Column 8 gives the maximum cascade length ML in each design. Column 9 gives the iteration count IC of Algorithm HIPPE, that is, the maximum number of iterations before the difference in the signal statistics between two consecutive iterations is less than 0.1%. The first eight IP systems have no feedback loops, and the corresponding IC values are ignored. The last three columns of Table 1 give the percentage of estimation runs whose power estimation error was below 5%, 15%, and 25%, respectively, in comparison with SLS. Each power estimate in the table was obtained within 0.2 seconds of CPU time on a Sun UltraSparc II workstation.

The results in Table 1 show that our static power estimation methodology achieves very high accuracy with remarkable speed. The average output macromodeling error does not exceed 5.7%. The average power estimation error is 7.24%. For all but four of our IP systems, the power estimation error is within 15% of the SLS result for more than 98% of the simulation runs. Our estimation algorithm

Design Name	IP #	$\Delta(P)$ (%)	$\Delta(D)$ (%)	$\Delta(S)$ (%)	$\Delta\mathcal{P}$ (%)	$STD(\mathcal{P})$	ML	IC	$\Delta\mathcal{P} \leq 5\%$ (%)	$\Delta\mathcal{P} \leq 15\%$ (%)	$\Delta\mathcal{P} \leq 25\%$ (%)
ht8	24	1.82	3.44	1.33	3.26	0.024	3	-	80.7	99.6	100.0
ht64	384	1.48	3.71	1.16	1.38	0.004	8	-	100.0	100.0	100.0
fft8	36	2.14	5.72	1.57	2.43	0.023	3	-	90.3	100.0	100.0
fft64	576	2.38	4.49	1.53	1.92	0.006	8	-	100.0	100.0	100.0
fir1	59	4.48	12.24	5.58	6.28	0.088	30	-	63.5	90.8	95.0
fir2	119	4.64	7.93	3.57	15.17	0.229	60	-	46.2	74.0	83.2
fir3	239	3.06	5.96	2.47	44.88	0.612	120	-	13.0	35.3	61.3
des	64	6.03	11.88	20.76	11.88	0.115	64	-	27.7	68.6	91.8
htfb8	24	1.33	3.46	0.98	2.20	0.017	3	3	92.4	100.0	100.0
htfb64	384	1.49	3.91	1.17	1.13	0.002	8	2	100.0	100.0	100.0
iir1	16	1.59	3.64	0.88	4.49	0.037	4	7	66.4	98.3	99.6
iir2	16	1.51	3.17	0.76	3.37	0.035	4	7	78.6	98.7	99.6
iir3	16	1.44	4.92	0.90	4.61	0.037	4	7	66.0	98.3	99.6
iir4	60	1.70	5.16	1.11	3.78	0.036	15	17	75.2	99.2	99.6
iir5	120	1.51	3.70	0.77	3.66	0.030	30	33	73.1	99.6	100.0
iir6	240	5.61	7.79	4.88	5.36	0.035	60	62	53.4	98.3	99.6
average	148	2.63	5.69	3.09	7.24	0.083	26.5	17	70.4	91.3	95.6

Table 1. Comparison of estimates obtained by Algorithm HIPPE and switch-level simulations

performs worst on the FIR designs, which include long cascades. In FIR designs, the error grows linearly with the length of the cascade. The circuit *fir3* has the longest cascade of 120 IP blocks and the largest power estimation error. For the HT and FFT designs, the power estimation error decreases as ML increases. In these modules, the number of parallel paths increases exponentially faster than the maximum cascade length. Consequently, the likelihood of the power estimation errors canceling each other increases, and the resulting error is much smaller than our theoretical linear bound. We also observe that the power estimation error is smaller for the circuits with feedback loops than those without loops. The reason for this seemingly counterintuitive result is that the feedback systems usually have stable operating points unique to their inputs. Our methodology can get to these points very well, and therefore compute highly accurate power estimates.

The power estimation error of our static methodology degrades gracefully, as evidenced by the histogram in Figure 6. The x axis in this histogram is the power estimation error. The y axis is the percentage of the estimation runs that resulted in a given error range. The runs with estimation error rate greater than 50% have all been aggregated in the last column of the histogram. This histogram shows that our methodology results in a low power estimation error for the vast majority of estimation runs. Specifically, for 91% of the runs, the power estimation error is less than 15%. For 2% of the runs, the error exceeds 50%. Most of these runs are related to the FIR designs that contain long cascades of IP blocks. Another reason for observing this small fraction

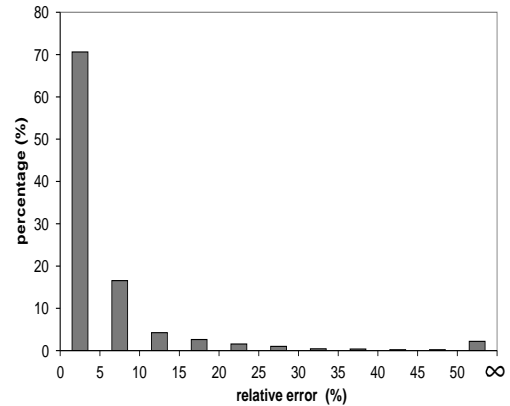


Figure 6. The distribution of the power estimation error rate

of high-error estimates is the use of the minimum mean-square error criterion for macromodel characterization. Although this criterion reduces the absolute estimation error, when the estimation result is small, the relative error becomes disproportionately high. Finally, the few high-error instances are also attributed to corner cases of the input sequences that include highly-correlated inputs. For example, if the two inputs of a subtractor are identical, all the output bits are zero, a situation that generates a relatively high error.

Design Name	P	D	S
adder	1.07	1.23	0.97
subtractor	1.07	1.23	0.97
amplitude modulator	0.77	1.00	0.83
substitution box	0.17	0.77	0.10
permutation box	1.00	1.00	1.00
extension box	0.07	1.00	0.03
modulo-2 adder	0.59	0.95	1.07
average	0.68	1.02	0.71

Table 2. Average output sensitivities of IP blocks.

A key issue in our methodology is the magnitude of the output sensitivities $\left| \overline{F'_n} \right| \cdot \vec{1}_3$ of the IP blocks. Table 2 shows the average output sensitivities of the 7 IP blocks in our experiments. In general, the average output sensitivities are small, in most cases less than 1. Although in certain cases the sufficient conditions from Section 4 are violated, our methodology still results in a very low power estimation error. This observation is explained by the fact that our analysis is a worst-case one and assumes no error cancellation.

6. Conclusion

We have presented a new static power estimation methodology for IP-based systems. Our power estimation procedure is completely simulation free, and therefore it greatly reduces estimation time. To our knowledge, this is the first fully-static, macromodeling-based methodology to be published for estimating the power dissipation of IP systems. Through rigorous analysis, we give sufficient conditions on the macromodel sensitivities that ensure the boundedness of the power estimation error of our methodology for designs of arbitrary topology. In experiments with several IP systems, the average power estimation error of our methodology is 7.3%, in comparison with switch-level simulation, with a standard deviation of 0.08.

We are currently analyzing results from the application of our methodology on control intensive systems containing IP blocks of sequential logic. We are also investigating the inaccuracy led by the glitches propagated between IP blocks and improving the output macromodeling metrics for the data-dependent branch prediction.

Acknowledgments

This research was supported in part by the National Science Foundation under Grant No. MIP-9423886 and Grant No. MIP-9610108 and by the US Army Research Office under Grant No. DAAD-19-99-1-0304.

References

- [1] M. Barocci, L. Benini, A. Bogliolo, B. Ricco, and G. D. Micheli. Lookup table power macro-models for behavioral library components. In *Proc. IEEE Alessandro Volta Workshop on Low Power Design*, Mar. 1999.
- [2] G. Bernacchia and M. Papaefthymiou. Analytical macromodeling for high-level power estimation. In *Proceedings of IEEE International Conference on Computer Aided Design*, Nov. 1999.
- [3] Z. Chen and K. Roy. A power macromodeling technique based on power sensitivity. In *Proc. 35th Design Automation Conference*, June 1998.
- [4] Z. Chen, K. Roy, and T. L. Chou. Power sensitivity—a new method to estimate power dissipation considering uncertain specifications of primary inputs. In *Proceedings of IEEE International Conference on Computer Aided Design*, Nov. 1997.
- [5] S. Gupta and F. N. Najm. Power macromodeling for high level power estimation. In *Proc. 34th Design Automation Conference*, June 1997.
- [6] S. Gupta and F. N. Najm. Analytical model for high level power modeling of combinational and sequential circuits. In *Proc. IEEE Alessandro Volta Workshop on Low Power Design*, Mar. 1999.
- [7] P. Landman. High level power estimation. In *Proceedings of International Symposium on Low Power Electronics and Design*, Aug. 1996.
- [8] X. Liu and M. C. Papaefthymiou. A macromodeling sequence generator based on Markov chains. *University of Michigan, Department of EECS, Technical Report*, Sept. 2000.
- [9] F. N. Najm. Transition density: A stochastic measure of activity in digital circuits. In *Proc. 28th Design Automation Conference*, June 1991.
- [10] F. N. Najm. A survey of power estimation techniques in VLSI circuits. *IEEE Trans. VLSI Systems*, 2(4):446–455, Dec. 1994.
- [11] National Bureau of Standards. *Data Encryption Standard*. Federal Information Processing Standard (FIPS), Publication No. 46, Jan. 1977.
- [12] J. M. Rabaey. *Digital Integrated Circuits*. Prentice Hall, New Jersey, 1996.