



Power estimation for intellectual property-based digital systems at the architectural level



Yaseer Arafat Durrani ^a, Teresa Riesgo ^{b,*}

^a Dept. of Electronic Engineering, University of Engineering & Technology, Taxila, Pakistan

^b Centro de Electrónica Industrial, E.T.S.I. Industriales, Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal 2, 28006 Madrid, Spain

Received 14 December 2012; revised 4 November 2013; accepted 13 March 2014

Available online 19 May 2014

KEYWORDS

Digital system;
Intellectual property;
Genetic algorithm;
Power macro-modeling;
Look-up-table;
Register transfer level

Abstract Estimating power consumption is becoming the critical issue that cannot be neglected in VLSI (very large scale integration) design procedure. Low power solutions are an imperative requirement for the SoC (System-on-Chip) flow that gives designers a powerful methodology to analyze, estimate, and optimize today's increasing power concerns.

We present an efficient power macro-modeling technique at the architectural level for digital electronic systems. This technique estimates the power dissipation of intellectual property (IP) components to their statistical knowledge of the primary inputs/outputs. During the power estimation method, the sequence of an input stream is generated by a genetic algorithm (GA) using input metrics and the macro-model function to construct a set of functions that map the input metrics of a macro-block to its output metrics. Then, a Monte Carlo zero-delay simulation is performed and the power dissipation is predicted by a macro-model function. The most important contribution of the technique is that it allows fast power estimation of IP-based design by the simple addition of individual power consumption. This makes the power modeling of SoCs an easy task that permits evaluation of power features at the architectural level. In order to evaluate our model, we have constructed IP-based digital systems using different IP macro-blocks. In experiments with an individual IP macro-block the average error is 1–2% and for an entire IP-based system with interconnects, the error range is from 9% to 15%. The preliminary results are effective and our macro-model provides accurate power estimation.

© 2014 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

In early VLSI design, the motivation was to find an acceptable balance between often conflicting constraints such as performance, area, reliability and cost. Recently, low power consumption has become the most important objective as a design constraint. A key challenge in low-power systems is accurate and fast power estimation. Power analysis at higher design level, such as computer architecture and software

* Corresponding author. Tel.: +92 519047728.

E-mail addresses: yaseer.durrani@uettaxila.edu.pk (Y.A. Durrani), teresa.riesgo@upm.es (T. Riesgo).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

engineering, is called for to provide new solutions to power problems (Ronen et al., 2001; De et al., 1999). Hence, a design and estimation technique for low power is the key to a successful SoC design.

As rapid growth of a system's complexity and verifications become increasingly difficult and time consuming, power and performance analysis at the early stages of the design flow are essential for shortening the turn-around time. The design cost and time-to-market of the electronic systems can be greatly reduced through the reuse of predesigned circuits. The use of silicon IP has been proposed as one possible solution to the problems associated with SoC design. The designers need to leverage pre-validated components and IPs. Design methodology further supports IP reuse in a plug-and-play fashion, including buses and hierarchical interconnection infrastructure. Reuse design techniques employing IP cores cut down on time-to-market, and fast estimation shortens the design evaluation time, which is more efficiently used in design-space exploration. Power estimation models can be used at different levels of abstraction with corresponding variations in speed and accuracy.

Power analysis of the IP-based system is a particularly challenging task at the architecture level because the designers need to compute accurate power estimates without direct knowledge of the IP design details. With the wide deployment of portable systems, low-power chip design is becoming an increasingly important focus of VLSI research. Thus, at the architecture level the development of an efficient and effective power estimator for IP-based systems is an important and urgent need for the VLSI design communities (Liu and Papaefthymiou, 2001; Landman and Rabaey, 1996).

In this paper, we propose a power macro-modeling technique to solve the problem of high-level power estimation at the register transfer level (RTL). Various power estimation techniques have been introduced previously. The probabilistic technique uses the probabilities of the input patterns/streams and propagates into the circuit to estimate the internal transition activities of the circuit (Monteiro et al., 1997; Ding et al., 1998; Marculescu et al., 1998). These approaches are very effective, but they cannot accurately capture factors like propagation delay and glitch activities. In statistical techniques, the circuit is simulated under randomly generated input streams and the power dissipation is observed using a power estimation tool. The power values obtained are used to estimate the power consumption for every input stream. For more power accuracy, we need to generate the desired number of input vectors, which are usually large and cause run time problem. To solve this issue, a Monte Carlo simulation approach was introduced that uses the input vectors randomly generated to obtain the power values (Burch et al., 1993; Ismael and Breuer, 1991). The large number of samples combined with the previous samples required determining whether the entire process needs to be repeated in order to satisfy a certain given criteria. Most of the common approaches of statistical power estimation consider the input signal probabilities and their average transition activities of the input signal and use signal probabilities propagation methods to estimate the internal transition activities (Gupta and Najm, 1997). In those techniques, there is no guarantee that the estimated power maintains any relation with the real dissipation of the circuit. To handle this problem, a look-up table (LUT) – based macro-model was proposed in Gupta and Najm (2000) and further developed in Kozhaya and Najm

(2001). The model stores the equi-spaced discrete measured power values of the input statistical signals. The interpolation method was presented in Chen and Roy (1998) and further improved by using the power sensitivity concept in Liu and Papaefthymiou, (2002), Liu and Papaefthymiou (2005), Bernacchia and Papaefthymiou (1999), Koriem (2004).

In our previous research, we introduced temporal correlation T_{in} which captures those features that are missed in signal probability P_{in} , transition density D_{in} , and spatial correlation S_{in} (Durrani and Riesgo, 2007, 2009, 2013a,b; Durrani, 2013a). In this paper, we continue our recent work and further improved our power macro-model for IP-based digital systems. The input/output (I/O) metrics of our macro-model are the average input signal probability P_{in} , the average input transition density D_{in} , the input spatial correlation S_{in} , the input temporal correlation T_{in} , the average output signal probability P_{out} , the average output transition density D_{out} , the output spatial correlation S_{out} and the output temporal correlation T_{out} . In experiments, our macro-model $f(.)$ in “(10)” is evaluated on two different IP-based test systems. The most important contribution of our new method is that it allows fast power estimation of IP-based design by the simple addition of individual power consumptions. This makes the power modeling of SoCs an easy task and permits evaluation of the power features at the architectural level. Finally, we performed detailed *statistical error analysis* in “(12)” to find in-affective input metrics in each test system individually and develop a *new* macro-model with only affective metrics in “(13)” and “(14)”. The average error with an individual IP macro-block is 1–2% and for an entire test system (with macro-blocks and interconnects) the average error is estimated as 9–15%.

The rest of this paper is organized as follows. In Section 2, we provide the background of input/output metrics of our power macro-model. In Section 3, we propose the power estimation methodology for IP-based test systems. Our macro-model is evaluated in Section 4 and Section 5 summarizes our work.

2. Power macro-modeling background

One of the most challenging aspects in the construction of a power macro-model is the choice of the model's metrics. These metrics should capture the features that are primarily responsible for a system's dissipation and can thus help in obtaining good estimates of its power dissipation. We focus on the problem of power macro-modeling at RTL for IP-based designs. Our model is LUT based. The input/output (I/O) metrics of our macro-model are P_{in} , D_{in} , S_{in} , T_{in} , P_{out} , D_{out} , S_{out} , and T_{out} .

2.1. Input macro-modeling for IP-based macro-blocks

Once the I/O metrics are selected, the input sequences are computed by our genetic algorithm (GA) Durrani and Riesgo, 2006, and the output metrics are extracted from the functional simulations using a power simulator. Our power macro-model uses statistical techniques and it estimates the average power dissipation for the digital system. Our power macro-model consists of a nonlinear function based on the LUT approach and estimates the average power dissipation PIP_{avg} using “(1)”.

$$PIP_{avg} = f(P_{in}, D_{in}, S_{in}, T_{in}) \quad (1)$$

For a given IP macro-block, the macro-model function f is obtained by simulating different input sample streams with several values of the input metrics: the average input signal probability P_{in} , the average input transition density D_{in} , the input spatial correlation S_{in} and the input temporal correlation T_{in} . For a given IP macro-block with a number of primary inputs r , an input binary stream q of length s is: $q = \{(q_{11}, q_{12}, \dots, q_{1r}), (q_{21}, q_{22}, \dots, q_{2r}), \dots, (q_{s1}, q_{s2}, \dots, q_{sr})\}$ and the input metrics are defined as follows (Ismael and Breuer, 1991; Gupta and Najm, 1997, 2000; Kozhaya and Najm, 2001; Chen and Roy, 1998; Liu and Papaefthymiou, 2002; Liu and Papaefthymiou, 2005) using “(2)”, “(3)”, “(4)” and “(5)”.

$$P_{in} = \frac{\sum_{i=1}^r \sum_{j=1}^s q_{ij}}{r \times s} \quad (2)$$

$$D_{in} = \frac{\sum_{j=1}^r \sum_{i=1}^{s-1} q_{ij} \oplus q_{i+1j}}{r \times (s-1)} \quad (3)$$

$$S_{in} = \frac{\sum_{j=1}^r \sum_{k=1}^r \sum_{i=1}^s q_{ij} \oplus q_{ik}}{s \times r \times (r-1)} \quad (4)$$

$$T_{in} = \frac{\sum_{j=1}^r \sum_{t=1}^{s-t+1} (v_j \otimes q_j)}{r \times s} \quad (5)$$

The macro-model function $f(\cdot)$ in “(1)” is obtained by a given IP macro-block that maps the space of the input signal properties to the power dissipation of a circuit. When the input metrics of $f(\cdot)$ are solely determined by the input signals the computation of the power estimates is a straight-forward and fast-function evaluation. The most commonly used templates for the macro-model function $f(\cdot)$ are low-order polynomial functions. For a k th-order complete polynomial function with n input parameters, a total of S_{n+k}^k coefficients need to be computed. P_{in} , D_{in} , S_{in} , T_{in} can be calculated using “(2)”, “(3)”, “(4)”, and “(5)”, respectively.

2.2. Output macro-modeling for IP-based macro-blocks

Output macro-modeling was first introduced in Liu and Papaefthymiou, (2001) and was further improved in Durrani and Riesgo (2007, 2009, 2013b), Durrani (2013a,b) to predict the output metrics of an individual IP block from input metrics. In the characterization step, the functional simulation of the circuit is performed with different input sequences to obtain the output metrics. The function $f(\cdot)$ in “(1)” constructs a set of functions f_A, f_B, f_C and f_D that maps the input metrics of a macro-block to its output metrics $P_{out}, D_{out}, S_{out}$, and T_{out} , which are derived in “(6)”, “(7)”, “(8)”, and “(9)”:

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

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

$$S_{out} = f_C(P_{in}, D_{in}, S_{in}, T_{in}) \quad (8)$$

$$T_{out} = f_D(P_{in}, D_{in}, S_{in}, T_{in}) \quad (9)$$

The sensitivity of an output metrics with respect to P_{in} , D_{in} , S_{in} , and T_{in} is defined as the partial derivation of the corresponding function f_i .

2.3. Genetic algorithm

We analyze our genetic algorithm in Yaseer et al. (2006) for the power macro-modeling to estimate the power dissipation of the digital system. For a system S based on IP macro-blocks and the statistical signals Q as inputs of the system, our algorithm generates an input stream according to Q . The input Q gives the metrics, P_{in} , D_{in} , S_{in} , and T_{in} at the primary inputs, as shown in Fig. 1. The summary of the proposed GA for the IP system S is presented in Fig. 2.

GA generates the input patterns randomly by conforming to the prescribed input metrics of our macro-model. In the GA process, chromosomes are exposed to genetic operators like crossover, mutation, and selection. The objective of these operations is to remove poor strings and produce healthy strings. During the natural selection phase, the next generation is selected by their “fitness”. The fitness is a measure of how optimal a solution is relative to other potential solutions. The main goal of the GA is to mimic the natural process of evolution in order to produce the best solutions. After setup and creation of the population, our GA evolves the population until it contains satisfying potential solutions. The initial population consists of N random strings of length L . We choose to evolve the population a set number of times and then check what is the best solution produced by our GA. This process continues until the set number of generations reaches a pre-defined optimal solution.

2.4. Monte Carlo simulation

The Monte Carlo approach for power estimation was first proposed by Burch et al. (1993) and further improved in Gupta and Najm (2000), Kozhaya and Najm (2001). Using the same approach, our genetic algorithm generates the corresponding logic input waveforms according to P_{in} , D_{in} , S_{in} , and T_{in} . Then the method estimates the average power by sampling those input waveforms with a certain length l and feeding them into the simulator to derive a sample value. The average power consumption can be estimated with the average of several sample values. We perform the Monte Carlo zero-delay simulation technique for the digital IP-based test system and the power dissipation is obtained by our macro-model function. The interpolation can be applied (to improve the power sensitivity concept), if the input metrics do not match their characteristic scheme (Chen and Roy, 1998).

3. Power Macro-modeling for IP-based digital systems

Recently, we have introduced a power macro-model for different IP blocks in Durrani and Riesgo (2007, 2009). In this section, we present the power modeling methodology for the IP-based digital test systems. Our macro-model uses a nonlinear function to estimate the average power dissipation. In the *estimation phase*, we opted for a simple function with low-order polynomial dependency on $f(\cdot)$ in “(1)” having four input metrics of our power macro-model. In the *characterization phase*, we generate input metrics with the specified range between [0–1] for the given test system.

In our power estimation procedure, the sequence of an input stream is generated for the desired input metrics: P_{in} , D_{in} , S_{in} , and T_{in} . Then using functional simulations and a

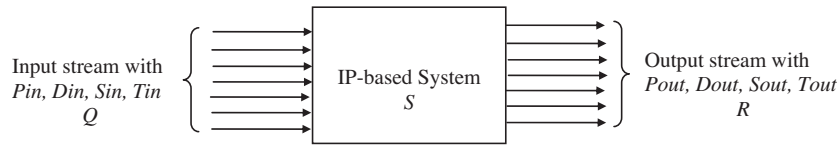


Figure 1 Block diagram of IP-based systems S.

```

GA for sequence of input pattern ()
fitness_value = 0;
num_gen = 0;
Generation of randomly population;
While (num_gen < max_num of generations)
  Compute the fitness values in the population;
  Upgrade the most appropriate fitness_value;
  Crossover;
  Mutation;
  Upgrade population;
  num_gen += 1;
end while;

```

Figure 2 Genetic algorithm.

power estimation tool, the output pattern sequence and the average power dissipation PIP_{avg} are extracted by the output waveforms of the IP macro-block. At this level, the power function in “(1)” can be defined. This method is divided into two steps. In the first step, the metrics of the I/O sequences are computed by our GA and the power function is obtained using PIP_{avg} in “(1)”. In the second step, a Monte Carlo zero-delay simulation is performed with several input sequences of their signal statistics to find the quality of the power function PIP_{avg} and we estimate the power results.

In our preliminary work, the approach intends to reduce the intensive amount of simulations at the RTL level. We use the same IP blocks and their macro-model information for our IP-based systems (Durrani and Riesgo, 2007, 2009). Instead of simulating every IP block, we applied the Monte Carlo zero-delay simulation to the entire test system. These macro-blocks are connected to construct the two different IP-based test systems shown in Fig. 3.

The application of the power macro-modeling on each IP block requires knowledge of the input signal statistics among these blocks. To obtain this information, different functional simulations are performed with different input statistical values of each IP macro-block. For example in Fig. 3(a), the inputs of the block IP-1A are the inputs of the test system I, whereas the outputs of IP-1A are the inputs of IP-1B, and the IP-1C IP blocks can be used as input signal statistics of the reference and so on. The output signal statistical information for each IP block can be used as the input signal statistics of the reference connected IP macro-block. For the IP-1A block, we generate random input vectors of 25 different values using input metrics P_{in} , D_{in} , S_{in} , and T_{in} . Then to construct the LUT, the test IP system is simulated 25 times and for each IP block, 25 different values of input metrics are measured using functional simulations. The average power dissipation P_{system} is extracted using “(10)”.

$$P_{system} = \sum_{i=1}^n P_{IPi_{avg}} \quad (10)$$

We compare the estimated power P_{system} in “(10)” with the simulated power estimation to evaluate the accuracy of the

power macro-model function in “(1)”. The main advantage of our macro-model is that it can provide fast and accurate estimates; thus, it helps designers to explore different complex blocks in real time.

4. Experimental results

In this section, we show the results of our LUT based power macro-modeling approach. We have implemented this approach and built the power macro-model at the architecture level. The accuracy of the proposed model is evaluated for two different IP-based test systems as shown in Fig. 3. For each IP macro-block, a random sequence of test patterns is performed with different values of P_{in} , D_{in} , S_{in} , and T_{in} . The function $f(\cdot)$ in “(5)” constructs a set of functions f_A , f_B , f_C and f_D in “(6)”, “(7)”, “(8)”, and “(9)” that maps the input metrics of a macro-block to its output metrics P_{out} , D_{out} , S_{out} , and T_{out} .

During the characterization phase, the average power consumption is measured using power function $f(\cdot)$, whereas least squares fitting is used to perform linear regression. The input chosen sequences are highly correlated and they are generated by our new method. The accuracy is tested by running gate-level and RTL simulations. The power is estimated using a Monte Carlo zero-delay simulation technique. We compare our power macro-modeling results $P_{estimated}$ with the Synopsys Power Compiler tool $P_{simulated}$ and compute the average absolute and maximum percentage errors using “(11)”.

The experimental results show that the randomly generated sequences have relatively accurate statistics and high convergence. For the verification of our random sequences, we compared our power results with the functional sequence power results and found a 96% correlation. Both random and functional sequences have similar input features. Several sequences that are 8, 16, and 32 bits wide are generated. We performed a synthetic validation by applying a uniform set of stochastically generated test-benches. All the results to be presented were performed with a 5% error-tolerance ($\epsilon = 0.05$) and 95% confidence ($\alpha = 0.05$).

Our pattern generator can generate a set of sequences over the entire space range between [0, 1]. Therefore, it enables us to perform extensive experiments to reveal the relation between IP design power dissipation and specific statistics of the input signals. In our study, we designed different IP macro-block/modules. For each block, we generated 350–1000 sequences with P_{in} , D_{in} , S_{in} , and T_{in} evenly distributed in the four/eight dimensional space. Our parameter granularity is 0.1 over the entire space. In practice, much larger sequences should be used for larger circuits. Roughly speaking, for a given IP module, we empirically observe that sufficiently long input sequences that produce similar steady state power exhibit similar total power. Given an IP module, for all the input sequences that produce a steady state power, we believe that hazardous power corresponding to an input sequences has the behavior of a

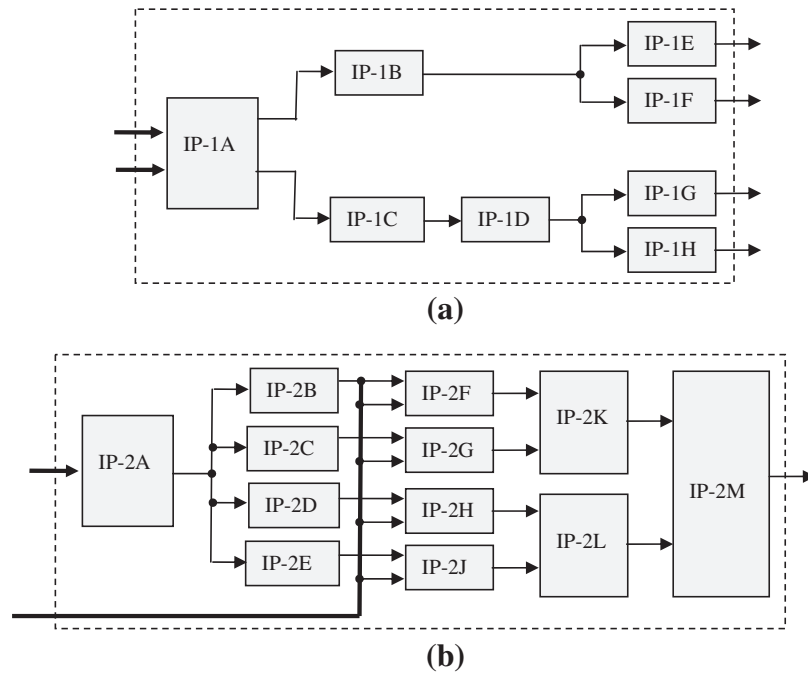


Figure 3 Two different IP-based test systems: (a) combinational logic circuit based test system-I, (b) sequential logic circuit based test system-II.

random variable. Furthermore, among all these input sequences that produce a steady state power, longer sequences tend to have smaller variance than shorter sequences.

As an example, given a three input logic network, assume sequences $\text{Seq}_1 = \{101, 111\}$, $\text{Seq}_2 = \{100, 110\}$, $\text{Seq}_3 = \{110, 011, 010, 011, 101, \dots\}$, and $\text{Seq}_4 = \{011, 101, 110, 110, 111, \dots\}$ all exhibit the same steady power. We believe that the hazardous power produced by these sequences, such as Seq_3 and Seq_4 has a smaller variance to the shorter sequences, such as Seq_1 and Seq_2 .

$$P_{\text{error}} = \frac{|P_{\text{simulated}} - P_{\text{estimated}}|}{P_{\text{simulated}}} \times 100\% \quad (11)$$

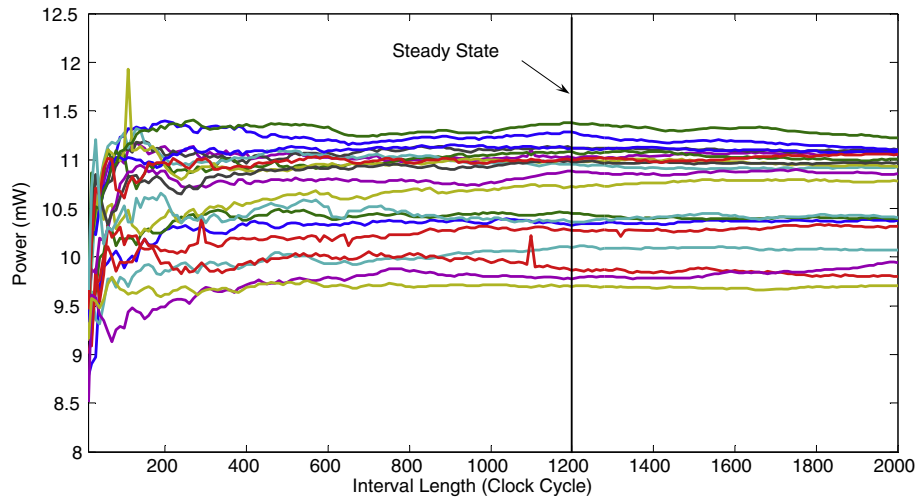
In Table 1, we illustrate the set number of the input vectors and the average relative errors of the estimate values obtained with our macro-model. The function is more accurate estimating the average power in some cases than others. For the input metrics, P_{in} , D_{in} , S_{in} , and T_{in} we specify the range between [0, 1]. The given input metrics values are more accurate for specifying the range between [0.2, 0.8] and less accurate between [0, 0.2] and [0.8, 1]. Our macro-model does not estimate the power consumption of interconnects among different IP macro-blocks. One important source of error is due to interconnects and other factors like glitch activities. For an individual IP block, we measure an error of only 1–2% in Koriem (2004), Durrani and Riesgo (2013a). It is evident from Table 1 that the macro-model function $f(\cdot)$ is accurate for estimating the average power for IP macro-blocks such as array multipliers, adders, registers and comparator circuits. The individual IP block consists of 500–5000 logic gates. In Table 1, the first column shows the name of the macro-blocks. The four dimensional input model estimates the absolute average and maximum relative error, which are shown in columns two and three. In our experiments, the average absolute errors of test

system-I and system-II are 0.94% and 1.94%, whereas the average maximum error is 1.90% and 3.49%, respectively. Columns four and five give the average and maximum relative error for the estimates obtained with the eight dimensional inputs/outputs model. The average absolute errors for both systems are 0.96% and 1.40%, whereas average maximum errors are 1.86% and 2.89%, respectively. We found that considering output metrics in the macro-model can only improve the accuracy 2–5%. These results mirror those obtained for power dissipation, showing that our technique could be used to effectively achieve fast and accurate results in the early stage of digital system design. For the entire IP-based system with interconnects the error increases by 20–30%. This error can be reduced by different techniques that improve the data-path of interconnects among IP macro-blocks. In our experiments, the average errors of the entire IP-based test systems I and II are 22.15% and 27.64%, respectively.

The minimum simulation length can be determined through convergence analysis. Converging on the average power figure helps us to identify the minimum length necessary for each simulation by considering when the power consumption gets close to a steady value given an arbitrary acceptance threshold. Additionally, the convergent sample size is not a function of the circuit size; it depends on how “widely” the power distributes. The sequences generated by our GA have high convergence and uniformity. Fig. 4 plots the variation of the power values with the trial interval length of 2000 for the IP system. The warm-up length is approximately 800 about the vertical line and represents the steady state value at 1200. Regression analysis is performed to fit the model’s coefficients. For IP-based test systems I and II, we measured the correlation coefficient as 96% and 87% and a 98% correlation between the input and the I/O metrics-based macro-models. For different blocks, the prediction correlation coefficient measured

Table 1 Accuracy of the power estimates.

Using input metrics			Using input/output metrics	
IP macro block	Average error (%)	Max error (%)	Average error (%)	Max error (%)
<i>Test system-I</i>				
IP-1A	0.60	2.96	0.68	3.01
IP-1B	0.29	1.10	0.31	1.40
IP-1C	0.73	1.57	0.74	1.67
IP-1D	0.47	0.66	0.50	0.78
IP-1E	1.38	2.36	1.30	2.66
IP-1F	2.15	2.87	2.35	2.17
IP-1G	1.00	1.54	1.10	1.14
IP-1H	0.91	2.12	0.71	2.01
Average error	0.94	1.90	0.96	1.86
<i>Test system-II</i>				
IP-2A	0.96	3.12	0.76	2.36
IP-2B	1.79	3.83	1.53	3.02
IP-2C	0.90	4.19	0.60	2.99
IP-2D	1.40	1.83	0.94	1.71
IP-2E	4.17	5.21	3.57	4.81
IP-2F	1.25	2.56	0.65	2.31
IP-2G	2.95	3.30	2.15	3.01
IP-2H	3.22	5.20	2.92	4.70
IP-2I	1.34	3.67	0.94	2.84
IP-2J	2.35	3.89	1.23	2.95
IP-2K	2.56	3.84	1.21	2.83
IP-2L	1.04	1.23	0.24	1.11
Average error	1.94	3.49	1.40	2.89

**Figure 4** Power changes with respect to sequence length.

around 97%, which is quite good. In macro-model function $f(\cdot)$, the output metrics do not significantly improve the average error, whereas they do improve the average maximum relative error. We have also noticed that the output metrics effectively improve the error for multiplier macro-blocks whereas for the comparator blocks, the result is the opposite. For the individual IP characterization, there would be an increased processor time if the I/O parameters are considered.

The results show that the transition density D_{in} is very effective for estimating power dissipation and is relatively linear to the power measures. In some cases the temporal/spatial correlations T_{in} and S_{in} do not significantly affect power dissipation

and are less sensitive than D_{in} . In other cases, neglecting the correlation metrics at the primary inputs causes inaccurate values for P_{in} and D_{in} . To demonstrate the correlation impact, we performed simulations for different IP macro-blocks with different sequences of input vectors. For example, every input was fixed to $P_{in} = 0.50$ for four simulations. The D_{in} of the primary input was set to 0.50 for the first simulation, 0.25 for the second, 0.10 for the third, and 0.02 for the fourth. The randomly generated uncorrelated input pattern was set to energy $E = 0.50$. Thus, the first simulation determines D_{in} at internal signals with correlation of input increases. With decreasing D_{in} at the inputs, the correlation of the input increases. Techniques

that neglect the correlation at the inputs produce the same values for the transition probability of an internal signal, regardless of the actual transition probabilities at the inputs, i.e., these techniques yield the result of the first simulation for any assignment of E to the inputs. Therefore, the instances where $E \neq 0.50$ at the inputs are compared to the simulations where $E = 0.50$ at the primary inputs.

4.1. Statistical error analysis

It is crucial to understand that all measurements of experiments are subject to uncertainties. It is never possible to measure anything exactly. In order to draw valid conclusions the error must be indicated and dealt with properly. Therefore, we performed statistical analyses to compute the error on the basis of various statistical tests, including standard normal distribution, correlation, covariance, and variance analyses with multivariate graphs, which provide an interesting view into the results of our experiments. A multiple regression model is performed to describe the relationship between the error and the four input variables in “(12)”.

$$\varepsilon = \beta_0 + \beta_1 P_{in} + \beta_2 D_{in} + \beta_3 S_{in} + \beta_4 T_{in} \quad (12)$$

where ε is the error and $\beta_1, \beta_2, \beta_3, \beta_4$, are the coefficients of the input variables.

Table 2, describes the relationship between the error and input variables (P_{in} , D_{in} , S_{in} , and T_{in}) of the two test systems discussed in section III. Table 2, demonstrates the *Pearson product-moment correlation* between each pair of the variables giving a value between +1 and -1 inclusive and measure the strength of the linear relationship between two variables. The number of pairs of data values was used to compute each coefficient. We found a strong correlation with the following pair of variables in test system I: (*Error- D_{in}* , *Error- S_{in}* , *Error- T_{in}* , *D_{in} - S_{in}* , *D_{in} - T_{in}* , and *S_{in} - T_{in}*) and in test system II: (*Error- D_{in}* , and *S_{in} - T_{in}*), respectively. In Table 3, we illustrate the statistical

Table 2 Relationship between input metrics and error.

Variables	Error	D_{in}	P_{in}	S_{in}	T_{in}
<i>Test system-I</i>					
Error	-	0	-0.01	0	0
		+0.95	+0.69	+0.84	+0.78
D_{in}	0	-	-0.01	0	0
		+0.95	+0.97	+0.90	+0.85
P_{in}	-0.01	-0.01	-	-0.03	-0.09
		+0.69	+0.97	+0.88	+0.67
S_{in}	0	0	-0.03	-	0
		+0.84	+0.90	+0.88	+0.87
T_{in}	0	0	-0.09	0	-
		+0.78	+0.85	+0.67	+0.87
<i>Test system-II</i>					
Error	-	0	-0.15	-0.20	-0.31
		+0.96	+0.51	+0.35	+0.15
D_{in}	0	-	-0.03	-0.75	-0.79
		+0.96	+0.90	0	0
P_{in}	-0.15	-0.03	-	-0.11	-0.16
		+0.51	+0.90	+0.63	+0.47
S_{in}	-0.20	-0.75	-0.11	-	0
		+0.35	0	+0.63	+0.92
T_{in}	-0.31	-0.79	-0.16	0	-
		+0.15	0	+0.47	+0.92

Table 3 Statistical error analysis for IP-based test systems.

Parameter	Standard estimates	R-squared statistics	Total error	P-value
<i>Test system-I</i>				
Constant	-5.16	-1.37	3.76	0.19
D_{in}	18.24	6.16	2.96	0.00
P_{in}	0.06	0.07	1.18	0.95
S_{in}	0.27	0.09	3.03	0.93
T_{in}	2.58	0.61	4.24	0.55
<i>Test system-II</i>				
Constant	-10.21	3.39	2.62	0.00
D_{in}	-4.56	-2.07	1.20	0.04
P_{in}	-1.51	-1.03	1.04	0.17
S_{in}	3.45	1.02	3.36	0.32
T_{in}	-11.95	-2.51	4.76	0.02

error analysis results to find the P -value. The P -value tests the statistical significance of the estimated correlations. A P -value less than 0.05 indicates statistically the significance of non-zero correlations at the 95% confidence level. Other inputs are not significant if the P -value is greater than 0.05. The important results for both systems are given below:

4.1.1. For test system-I

In Table 3, the first column shows the input variables of “(12)”. The second column demonstrates the standard error of the estimates with the standard deviation of the residuals, which is found to be 1.61. This value helps to construct prediction limits for new observations. In the third column, the R -squared statistics indicates the model is fit 90.20% of the variability in the error. The adjusted R -squared statistic, which is more suitable for comparing independent variables, is 88.23%. The mean absolute error of 1.14 is the average value of the residuals. In the fifth column of the table, we found the P -value of D_{in} is 0.00, which means, there is a model with a different statistically significant relationship between the variables at the 99% confidence level. Therefore D_{in} is the only significant parameter in the error, which is the main factor of the power consumption for interconnections among IP blocks. The other input variables (P_{in} , S_{in} , T_{in}) are not very influenced in “(12)” for this particular test system. Hence, our model in “(13)” can be further simplified as:

$$\varepsilon = \beta_0 + \beta_1 D_{in} \quad (13)$$

Fig. 5(a) illustrates the correlation between the simulated power, estimated power and the estimated corrected power values. After introducing the simplified model in “(13)”, we measured a further improved correlation coefficient from 96% to 99%. For the entire system, the average error is also improved from 22.15% to 9.23%.

4.2.2. For test system-II

In the second column of the Table 3, the standard deviation of the residuals is 1.21. In the third column, the R -squared statistic indicates the model is fit with 34.72% of the variability in error. The adjusted R -squared statistic with different independent variables is 20.21%. The mean absolute error of 0.83 is the average value of the residuals. In the fifth column of the table, we found the P -values of D_{in} and T_{in} are 0.04 and 0.02, respectively, which means, there is a statistically

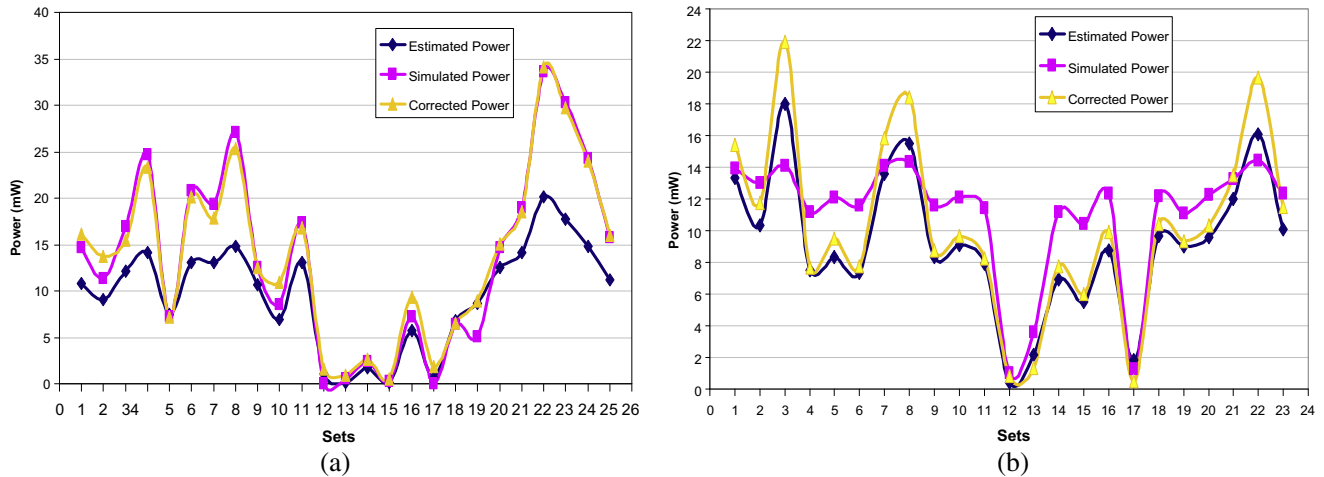


Figure 5 (a) IP-based test system-I (b) IP-based test system-II: correlation between the estimated, simulated and estimated-corrected power.

significant relationship between the variables at the 96–98% confidence level. Therefore D_{in} and T_{in} are the only significant parameters in the error and are the main factors of power consumption for interconnections among IP blocks. The other input variables (P_{in} and S_{in}) are not influenced in “(12)” for this particular IP-based test system. Hence, our model in “(14)” can be further simplified as:

$$\varepsilon = \beta_0 + \beta_1 D_{in} + \beta_2 T_{in} \quad (14)$$

Fig. 5(b) illustrates the correlation between the simulated power, estimated power and the estimated corrected power values. After introducing the simplified model in “(14)”, the correlation coefficient decreased from 87% to 85%. For the entire system, the average error is improved from 27.64% to 15.25%.

5. Conclusion

We have presented an efficient power macro-modeling technique at the architectural level applied to two different IP-based test systems using combinational and sequential circuits. In our preliminary work, for an individual IP block, we measured just 1–2% error. But for an entire IP-based system with interconnects, the error is measured in the range of 20–30%. This is because the macro-model should consider the power consumption of interconnects among different IP macro-blocks and other factors like glitches. We demonstrated relatively better accuracy in some cases than in others. Our improved statistical model showed an average error from 9% to 15% and a correlation coefficient from 96% to 85%. The output metrics in the macro-model can only improve the accuracy of 2–5%. Currently, we are evaluating our macro-model for more complex IP-based systems and are working to further improve its accuracy.

References

Bernacchia, G., Papaefthymiou, M.C., 1999. Analytical macromodeling for high-level power estimation. *Proc. IEEE Int. Conf. Comput. Aided Des.*, 280–283.

- Burch, R., Najm, F.N., Yang, P., Trick, T., 1993. A Monte Carlo approach for power estimation. *IEEE Trans. VLSI Syst.* 1 (1), 63–71.
- Chen, Z., Roy, K., 1998. A power macromodeling technique based on power sensitivity. *Proceeding 35th design automation conference*, June.
- De V., Borkar, S., 1999. Technology and challenges for low power and high performance. *Proceeding of the international symposium on low power electronics and design*, pp. 163–168.
- Ding, C., Tsui, C., Pedram, M., 1998. Gate-level power estimation using tagged probabilistic simulation. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 17 (11), 1099–1107.
- Durrani, Yaseer A., 2013a. High level power optimization for array multipliers. *J. Nucl.* 50 (4), 351–358.
- Durrani, Yaseer A., 2013b. Accurate power analysis for conventional MOS transistors using 0.12 μm technology”. *J. Nucl.* 50 (4), 341–350.
- Durrani, Yaseer A., Riesgo, T., 2006. Statistical power estimation for register transfer level. *Proceedings for International Conference for Mixed Design of Integrated Circuits and Systems*, pp. 522–527.
- Durrani, Yaseer A., Riesgo, T., 2007. Architectural power analysis for intellectual property-based digital system. *J. Low Power Electron.* 3 (3), 271–279, Dec (9).
- Durrani, Yaseer A., Riesgo, T., 2009. Power estimation technique for DSP architecture. *Elsevier J. Digital Signal Process.* 19 (2), 213–219.
- Durrani, Yaseer A., Riesgo, Teresa, 2013a. High-level power analysis for intellectual property-based digital systems. *Springer Circuits, Syst. Signal Process.* 32 (6).
- Durrani, Yaseer A., Riesgo, Teresa, 2013b. High-level power analysis for IP-based digital systems. *J. Low Power Electron.* 9 (4), 435–444.
- Gupta, S., Najm, F.N., 1997. Power macromodeling for high-level power estimation. *Proceeding 34th design automation conference*, June.
- Gupta, S., Najm, F.N., 2000. Power macromodeling for high-level power estimation. *IEEE Trans. VLSI Syst.* 8 (1), 19–29.
- Ismael, A.A., Breuer, M.A., 1991. The probability of error detection in sequential circuits using random test vectors. *J. Electron. Test.* 1, 245–256.
- Korriem, Samir M., 2004. Modeling concurrent, sequential, storage, retrieval, and scheduling activities of multimedia systems. *J. King Saud Univ. Comput. Inf. Sci.* 17, 61–96.
- Kozhaya, J.N., Najm, F.N., 2001. Power estimation for large sequential circuits. *IEEE Trans. Very Large Scale Integ. Syst.* 9 (2), 400–407.

- Landman, P., Rabaey, J.M., 1996. Activity-sensitive architectural power analysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 15 (6), 571–587.
- Liu, X., Papaefthymiou, M., 2001. A static power estimation methodology for IP-based design. *Proceedings IEEE conference on design, automation & test in Europe*, pp. 280–287.
- Liu, X., Papaefthymiou, M.C., 2002. Incorporation of input glitches into power macromodeling. *Proceeding IEEE international symposium on circuits and systems*, May.
- Liu, X., Papaefthymiou, M.C., 2005. HyPE: hybrid power estimation for IP-based systems-on-chip. *Proc. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 24 (7), 1089–1103.
- Marculescu, R., Marculescu, D., Pedram, M., 1998. Probabilistic modeling of dependencies during switching activity analysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 17 (2), 73–83, Feb.
- Monteiro, J., Devadas, S., Ghosh, A., Keutzer, K., White, J., 1997. Estimation of average switching activity in combinational logic circuits using symbolic simulation. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 16 (1), 121–127.
- Ronen, R., Mendelson, A., Lai, K., Lu, S.-L., Pollack, F., Shen, J.P., 2001. Coming challenges in microarchitecture & architecture. *Proc. IEEE* 89 (3), 325–340.