Improving Sampling Efficiency for System Level Power Estimation*

Chih-Shun Ding
Rockwell Semiconductor Systems
Rockwell International Corporation
Newport Beach, CA 92660

Cheng-Ta Hsieh Massoud Pedram
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

Abstract

In this paper, we propose an efficient statistical sampling technique which is suitable for estimating the total power consumption of a large VLSI system. The basic idea is to generate simulation units for each module in the system independently and then form samples of the system power by randomly selecting simulation units for each module. Hence, sampling is performed both temporally (across different clock cycles) and spatially (across different modules). A module clustering step ensures that the module types are compatible with this sampling strategy. Experimental results show a 4x reduction in the simulation time compared to existing Monte-Carlo simulation techniques.

1 Introduction

Power dissipation, along with area and speed, has become one of the important design metrics in VLSI designs. As a result, the issue of estimating the power dissipation of a VLSI system which consists of a large number of modules has become a critical task. Notice that this problem is different from the problem of estimating the power dissipation of an individual module in the system.

One way to improve the simulation efficiency is to apply statistical sampling techniques to select a small subset of the applied vectors for power simulation. Such techniques use a Monte Carlo iteration loop to incrementally improve the estimation accuracy until user-specified error and confidence levels are met [1, 2, 3]. A common characteristic of these techniques is that the simulation time, in terms of the number of simulated clock cycles, is directly proportional to the cycle-by-cycle fluctuation of power dissipation, or in statistical terms, the *relative variance* of the power dissipation per cycle (Relative variance is defined as the ratio of the variance to the mean squared).

Previous sampling techniques can be described as *clock-based* techniques since the total power dissipation of the target system at a single clock is the basic unit of sampling. Conceptually, one can view the cycle-based sampling strategy at the system level as sampling each module in a system simultaneously and *synchronously.* That is, if the power measurement of a module in a particular clock cycle is included in a power sample, the power measurements of the remaining modules in the same clock cycle must also be included in the same sample. These strategies essentially sample in the temporal domain only.

In this paper, we will show that it is possible to improve the efficiency by sampling in the spatial domain as well. More precisely, we will show that as the number of modules in a system increases, the relative variance of the cycle-by-cycle power decreases. This results in a smaller number of samples needed for convergence in a Monte-Carlo simulation. We will next show that as the number of samples required for convergence becomes smaller than 10, Monte-Carlo simulation oversamples by more than 20%. In fact, the smaller the number of samples required for convergence, the higher the oversampling.

*This research was supported in part by DARPA under contract no. F336125-95-C1627, SRC under contract no. 98-DJ-606 and a grant from Rockwell Semiconductor Systems.

The primary cause of this problem is that each sample unit in clock-based sampling techniques is very large. If we reduce the size of each sample unit, we can avoid the problem of oversampling in Monte-Carlo simulation. The sampling techniques proposed in this paper use the collection of the power dissipations of each module at each cycle as the population for sampling. Furthermore, estimation of total system power is performed with one Monte-Carlo simulation. This is different from the strategy which estimates each module in one Monte-Carlo simulation and sums up the estimates to produce the total system power estimate. The drawback of the latter strategy is that it is difficult to determine *a prior* what confidence and error levels should be assigned for each module so that the sum of individual power estimates satisfies user-specified error and confidence levels of the total system power estimate.

The rest of the paper is organized as follows. In Section 2, we consider sampling issues for system level power estimation. We propose a cluster-based technique in Section 3. Experimental results are presented in Section 4 followed by concluding remarks in Section 5.

2 System Level Issues

A *system* is defined as a collection of *modules* in which all inputs come either from system inputs or from other modules in the system. We further assume that the vector traces of all sequential elements are given as a result of functional simulation of the system.

Let m and n denote the total number of modules and clock cycles, respectively. Power dissipation of the jth module, M_j, at the ith clock cycle is denoted as $p_{i,j}$ and stored at the $< i, j >$ entry of a *power log matrix.* Note that this matrix is just for conceptual convenience; the actual power values $p_{i,j}$ are not known before simulation. Only the input vectors that will be used to simulate the jth module at the ith clock cycle are known at this time.

To compare different sampling strategies, we define the notion of (*simulation*) *workload.* The workload for a sample, is calculated as the product of the number of transistors being simulated times the number of simulated cycles. Therefore, if a system consists of modules A and B (each having 10k transistors) is simulated for 30 clock cycles and the average power over the 30 cycles is used to produce one sample value, then the workload for the sample is measured as 20k×30=600k transistor-cycles. On the other hand, if module A is simulated for 15 cycles and module B is simulated for 15 cycles and the 30 simulation results are averaged and multiplied by a factor of 2 to produce the sample value, then the workload for the sample is measured as 10k×15+10k×15=300k transistor-cycles.

From the analytical point of view, if the population mean and variance are known *a prior*, then $E[X]$ and $V[X]$ of samples are also known. In addition, if the samples follow normal distribution, we can compute the ideal number of required samples, k_{ideal}, to achieve an error level ϵ and

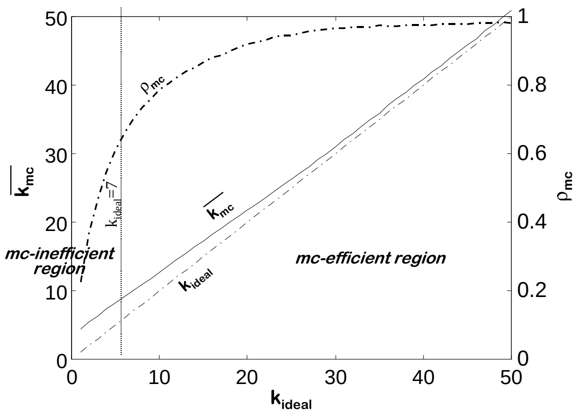

Figure 1: Oversampling in Monte Carlo simulation.

a confidence level $(1-\alpha)$, as follows:

$$k_{ideal} = (\frac{z_{\alpha/2}}{\epsilon})^2 V_{rel}(X) \tag{1}$$

where $z_{\alpha/2}$ is defined such that the area to its right under a standard normal distribution is equal to $\alpha/2$ and $V_{rel}(X)$ is the relative variance of X.

If we perform a large number of runs using Monte Carlo simulation on the same population for the same values of ϵ and $(1-\alpha)$, we obtain an average value for the number of required samples, denoted as $\overline{k_{mc}}$. The relationships for $\overline{k_{mc}}$ and ρ_{mc} versus k_{ideal} are plotted for a 99% confidence level in Figure 1 ($\overline{k_{mc}}$ is computed by 100,000 runs of Monte Carlo simulation). Note that both $\overline{k_{mc}}$ and ρ_{mc} are independent of ϵ.

The efficiency coefficient ρ_{mc} for Monte Carlo simulation is defined as

$$\rho_{mc} = \frac{k_{ideal}}{\overline{k_{mc}}} \times 100\%$$

The Monte Carlo efficient region, called *mc-efficient* region, is defined as the range of k_{ideal} values where $\rho_{mc} \geq 70\%$. The compliment set of the mc-efficient region is defined as the *mc-inefficient* region.

From Figure 1, the mc-efficient region is found to be in the range $[7, \infty]$. In this region, the smaller k_{ideal}, the lower ρ_{mc}. For instance, when $k_{ideal} = 2$, $\rho_{mc} = 37\%$ whereas $k_{ideal} = 1$, $\rho_{mc} = 22\%$. On the other hand, when $k_{ideal} = 10$, $\rho_{mc} = 80\%$, a mere 20% loss in efficiency. In a large VLSI system, the power simulation for a single clock cycle requires a significant computation time. Therefore, although k_{ideal} is very small in the mc-inefficient region, the impact of inefficient estimation is still enormous.

While we have demonstrated that Monte Carlo simulation may oversample, the question remains: "Will we run into mc-inefficient region when using clock-based Monte Carlo simulation for estimating the system power?" In another words, "what k_{ideal} values do we encounter in practice?" It is difficult to give a precise quantitative answer. Our approach is to derive the relation between k_{ideal} at the module level and that at the system level. Based on results reported at the module level [1, 2], we can infer the possible range of k_{ideal} values for system power estimation. We consider both cases where the modules are uncorrelated and correlated.

Let k_{ideal}^{system} denote the value for k_{ideal} for the whole system, whereas $MAX(k_{ideal}^{module})$ denote the maximum k_{ideal} for the modules in the same system.

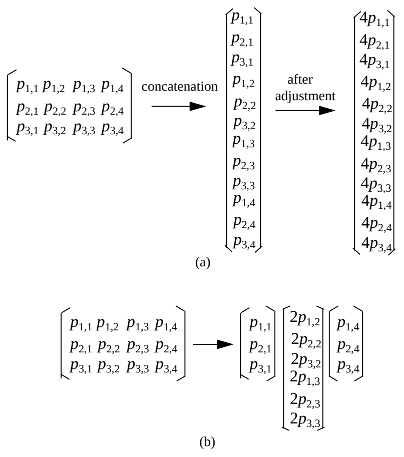

Figure 2: Illustration of the sampling strategy based on the power log matrix

Theorem 2.1 *Consider a system which has m uncorrelated modules $M_1, M_2, ..M_m$. Let the random variable representing the cycle-by-cycle power of module M_i be P_i and its mean be $E[P_i]$,*

$$k_{ideal}^{system} \leq \frac{\sum_{i=1}^{m} E^2[P_i]}{(\sum_{i=1}^{m} E[P_i])^2} MAX(k_{ideal}^{module}) \tag{2}$$

Theorem 2.2 *Let the system be the same as the one defined in Theorem 2.1, except that the modules are correlated, we have*

$$k_{ideal}^{system} \leq MAX(k_{ideal}^{module}) \tag{3}$$

For the proofs of the above Theorems, please refer to [4]. Note that if the covariances among module power dissipation are positive, the k_{ideal}^{system} becomes comparable in magnitude to $MAX(k_{ideal}^{module})$. In the case of some negative covariances, the k_{ideal}^{system} is smaller than $MAX(k_{ideal}^{module})$.

3 Cluster-based Power Estimation

To give an intuitive rationale for our approach, we need to revisit (1).

From (1), the total number of required simulation units (vector pairs) is given by

$$k_{ideal}^{system} \cdot n_s = (\frac{z_{\alpha/2}}{\epsilon})^2 V_{rel}(P_s) \tag{4}$$

where, n_s is the number of units per sample and P_s is the random variable representing the clock-based system power. Our goal is how to ensure the normality of samples while reducing n_s, which will in turn increase k_{ideal}^{system} and thus avoid the oversampling problem. We can achieve this by sampling each module independently and thus increasing the number of independent random variables included in a sample to ensure the normality of samples as described next. The rule of thumb is that there should be at least 30 independent (and comparable) random variables in a sample in order to maintain the normality of samples.

Consider a fairly homogeneous system (for instance, a system consisting of adders and subtractors). The case for heterogeneous systems will be explained later. There are n clock cycles and m modules in the system. The average power dissipation in this system is calculated as:

$$E(P_A) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \tag{5}$$

Next, consider system B which consists of a single adder but there are $n \cdot m$ clock cycles. Furthermore, the power log matrix of system B is obtained by concatenating all columns of the power log matrix of system A into a single column as shown in Figure 2(a). The average power of this system is:

$$E(P_B) = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \qquad (6)$$

Obviously, (6) is simply scaled down from (5) by m. Therefore all entries in the single column matrix should be multiplied by m if we want to estimate the average system power of system A by sampling on system B.

Our sampling strategy, which we call *mix-and-stratify*, is explained next. Consider system A as described above. First the zero-delay power estimates are calculated for all clock cycles and all modules, and the units of all modules in the system are put together. The mixture is stratified into m' equal size strata. One unit is randomly sampled from each stratum. We know which module this unit is from and can then perform transistor-level simulation on that module under the corresponding vector pair. The observed power from the simulation is multiplied by m and treated as a unit drawn in the standard stratified sampling [2], that is, when calculating the sample value, we follow the steps in the standard stratified sampling. This approach can easily be extended to non-equal size strata or non-equal sample sizes.

The advantage of the mix-and-stratify approach is that the minimum simulation workload in a sample to maintain normality of samples can be less than that of simulating the entire system for one clock cycle. To give an example, consider a system with 60 adder modules. These 60 modules are lumped into a cluster (super module). Let the number of strata be 30. One unit is randomly drawn from each stratum and collectively becomes a sample. The simulation workload of this sample is same as simulating only one half of the system for one clock cycle.

By heterogeneous systems, we mean systems which consist of more than one type of module. The type classification is based on the glitch activity in each module. For instance, multipliers are classified as different from adders. All modules of the same type are put into one cluster. In terms of the power log matrix, all the columns corresponding to the modules in the same cluster are concatenated into a single column matrix. Let the number of clusters be c and the number of modules in each cluster be m_i, where $i = 1, \ldots, c$. After the column concatenation, all the entries in the ith single column matrix are multiplied by m_i, as shown in the previous section. Then the estimation can be reduced to that of estimating the power on a new system consisting of c modules with each module representing a single column matrix as shown in Figure 2(b).

4 Experimental Results

We compared the sampling efficiency of two techniques: clock-based simple random sampling (SRS) and cluster-based stratified sampling(CSTS).

The circuits are selected from popular high level synthesis benchmarks: a Chebyshev filter (CHEB), a differential solver (DS), an IIR filter (IIR), and a discrete cosine transformation circuit (DCT). CHEB, DS, IIR, and DCT have 8(5), 4(7), 8(5), and 48(10) adders (multipliers), respectively. The input sequences are obtained from music CD's. For CSTS, we use zero-delay power estimate as the predictor. The target simulator is a real-delay gate-level simulator. The total circuit power of each circuit is first obtained by simulating the circuit over the entire sequence.

Table 1: Results of 100,000 Monte Carlo simulation runs for 5% error

	CHEB		DS	
Scheme	SRS	CSTS	SRS	CSTS
k_{avg}	6.0	8.7	11.6	10.7
W_{rel}	180	52	348	46
v.r.(%)	0.2	0.6	0.4	0.9
r.t	7.5	2.4	18.9	3.1
	IIR		DCT	
Scheme	SRS	CSTS	SRS	CSTS
k_{avg}	7.1	8.7	5.2	7.5
W_{rel}	180	53	156	17
v.r.(%)	0.7	0.6	0.2	0.4
r.t	9.0	2.5	16.2	2.1

We set the error and confidence levels to 5% and 0.99, respectively. We perform 10,000 simulation runs for each sampling methods. For stratified sampling, to reduce the overhead in calculating the predictor values, a different subpopulation of size 1000 is first randomly selected in each run. The stratification (predictor calculation, sorting) is only performed on the subpopulation. The sample size is set such that there are at least 30 independent random variables in a sample. We use two clusters. All adders are put into one cluster whereas all multipliers in another. Each cluster is stratified into the same number of stratum as the sample size and one unit is sampled from each stratum. The sample size for both techniques is 30.

The results are summarized in Table 1. The 'k_{avg}' rows list the average number of samples required in each method. The 'W_{rel}' rows list the average relative workload, defined as k_{avg} times the ratio of the workload per sample to the workload of simulating entire system for one clock cycle. The 'v.r' rows list the percentage of simulations that have greater than 5% error. The 'r.t' rows list the run time in seconds on a Pentium Pro machine. This table clearly shows that CSTS is better than SRS. The run time improvement is a factor of 4.

5 Conclusion

In this paper, we proposed efficient statistical sampling techniques suitable for system level power estimation. We first showed that Monte-Carlo-based statistical power estimation techniques tends to oversample when the relative variances of samples are small. To address this issue, we attempted to reduce the simulation workload per sample while maintaining the normality of samples. We thus proposed a cluster-based Monte-Carlo simulation technique to achieve this goal. We demonstrated that the proposed technique provides a reduction of 4x in terms of the simulation workload compared to existing Monte-Carlo simulation techniques.

REFERENCES

[1] R. Burch, F. N. Najm, P. Yang, and T. Trick. A Monte Carlo approach for power estimation. IEEE *Transactions on VLSI Systems*, 1(1):63–71, March 1993.

[2] C.-S. Ding, C.-T. Hsieh, Q. Wu, and M. Pedram. Stratified sampling for power estimation. In *Proceedings of the* IEEE *International Conference on CAD-96*, pages 577–582, November 1996.

[3] L.-P. Yuan, C.-C. Teng, and S.-M. Kang. Statistical estiamtion of average power dissipation in CMOS vlsi circuits using nonparametric technique. In *Proc. 1996 International Symp. on Low Power Electronics and Design*, pages 73–78, 1996.

[4] C.-S. Ding. Probabilistic and Statistical Sampling Techniques for Efficient Power Estimation in VLSI Circuits. In *Ph.D Dissertation, USC*, May 1998.