Statistical Sampling and Regression Analysis for RT-Level Power Evaluation*

Cheng-Ta Hsieh Qing Wu Chih-Shun Ding Massoud Pedram

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

Abstract

In this paper, we propose a statistical power evaluation framework at the RT-level. We first discuss the power macro-modeling formulation, and then propose a simple random sampling technique to alleviate the the overhead of macro-modeling during RTL simulation. Next, we describe a regression estimator to reduce the error of the macro-modeling approach. Experimental results indicate that the execution time of the simple random sampling combined with power macro-modeling is 50X lower than that of conventional macro-modeling while the percentage error of regression estimation combined with power macro-modeling is 16X lower than that of conventional macro-modeling. Hence, we provide the designer with options to either improve the accuracy or the execution time when using power macro-modeling in the context of RTL simulation.

1 Introduction

Power has become an important issue in chip design. In a high density circuit, excessive power consumption may increase the package cost and reduce the reliability. Portable electronics is another driving force for the low power design. To achieve low power, the power optimization and power estimation tools need to be developed. This paper focuses on the RT-level power estimation.

Most RT-level power estimation techniques use capacitance models for circuit modules and activity profiles for data or control signals [8, 4, 6]. Such techniques are commonly known as (power) macro-modeling. The simplest form of the macro-model equation is given by:

$$Power = 0.5V^2 f C_{eff} S$$

where C_{eff} is the effective capacitance, S is the mean of the input switching activity, and f is the clock frequency. The Power Factor Approximation (PFA) technique [8] uses an experimentally determined weighting factor, called the power factor, to model the average power consumed by a given module over a range of designs.

More sophisticated macro-model equations can be used to improve the accuracy. Dual Bit Type model, proposed in [4], exploits the fact that the switching activities of high order bits depend on the temporal correlation of data while lower order bits behave as white noise. Thus a module is completely characterized by its capacitance models in the MSB and LSB regions. The break-point between the regions is determined based on the applied signal statistics collected from simulation runs. The Activity-Based Control (ABC) model [5] is proposed to estimate the power consumption of random-logic controllers. All of the above macro-models assume some statistics about the input sequence.

*This research was supported in part by DARPA under contract number F336125-95-C1627, SRC under contract number 94-DJ-559, and NSF under contract number MIP-9457392.

The register-transfer level (RTL) power evaluation problem can be stated as follows: "Given an RTL description of a *datapath-dominated* circuit consisting of m modules and an input vector sequence of length N, calculate the average power consumption of the circuit over the N cycles". The simulation-based power evaluation process consists of two steps:

1) Perform RTL simulation and collect the input statistics for all modules in the circuit.
2) Evaluate the power macro-model equation for each module and sum over all the modules.

Busses, clock trees, control logic, memory, etc. are processed separately. Circuit power can be evaluated using a *power co-simulator* linked with a standard RTL simulator. The co-simulator is responsible for collecting input statistics from the output of RTL simulator and producing the power value. If the co-simulator is invoked by the RTL simulator at every simulation cycle to collect activity information, then it is called *census macro-modeling* (cf. Figure 1(a)).

There are two problems with the census macro-modeling:

- Input data statistics must be collected for every simulation cycle. The statistic gathering overhead is however large and hence slows down the RTL simulation. For a 16-bit multiplier, the RTL simulation needs only one instruction, while the statistic gathering requires tens of cycles. (See Figure 2.)[1] Thus the macro-modeling can slow down the simulation significantly. This shows that census macro-modeling is costly, especially when the vector sequence is very long (tens or hundreds of thousands of vectors).
- Power macro-models are developed by using a training set of input vectors. The training set satisfies certain assumptions such as being pseudo-random data, speech data, etc. Hence the macro-model is biased, meaning that it produces very good results for the class of data which behaves similarly to the training set; otherwise, it produces poor results.

We have developed two novel macro-modeling schemes:

1) *Sampler macro-modeling* collects and analyzes the input vectors for modules only for a relative small number of cycles (cf. Figure 1(b)). In this manner, the overhead of collecting input statistics at every cycle (which is required by census macro-modeling) is substantially reduced.
2) *Adaptive macro-modeling* not only interacts with the RTL simulator, but also invokes a gate-level simulator

[1] The overhead changes from one macro-model equation to another and from one simulator implementation to another.

ICCAD '96
1063-6757/96 $5.00

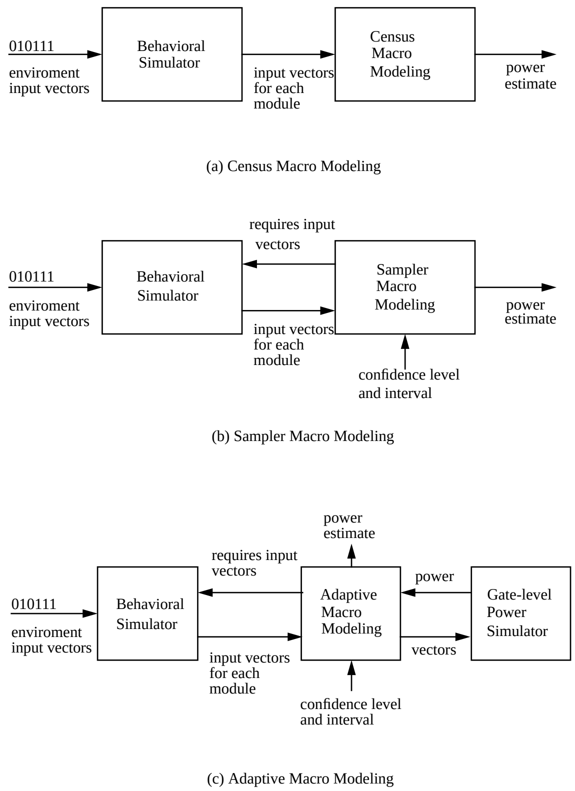


Figure 1: Power macro modeling.

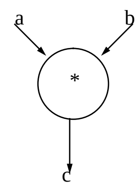
on a small number of cycles to improve the estimation accuracy (cf. Figure 1(c)). In this manner, the "bias" of the static macro-models (which is due to the choice of the training set) is reduced or even eliminated.

The sampler macro-modeling uses a simple random sampling technique to reduce the number of cycles during which data statistics is collected without loss of much accuracy while adaptive macro-modeling relies on regression analysis combined with gate-level simulation on a small number of cycles to "correct" the static macro-model estimate and hence can be thought of as a self-adjusting macro-model. The designer can select either of these techniques to make accuracy versus simulation time trade-off.

This paper is organized as follows. Section 2 discusses the general form of macro-model equations. Section 3 gives background in statistics. Section 4 discusses the basic simple random sampling technique while Section 5 discusses the regression estimator. Section 6 provides a statistical macro-modeling framework for power evaluation at high level. Experimental results and conclusion are provided in Sections 7 and 8.

2 Power Macro Model Equations

Power macro-modeling formulations in general consist of generating circuit capacitance models for some assumed data statistics. The statistics of input data is gathered during RTL simulation of the circuit. Power macro-modeling problem is defined as follows: "Given an input vector sequence of size N, an RT-level circuit with m modules, and assuming N is large enough to capture the typical operation of the circuit, derive a simple function such that the function value of the N vector inputs is as close as possible to the power consumption of the N-vector sequence".




```
c=a*b;
r1 = a^a';
r2 = b^b';
for(i=0;i<16;i++){
    sw_a[i] += r1&1;
    sw_b[i] += r2&1;
    r1 = r1 >> 1;
    r2 = r2 >> 1;
}
a' = a;
b' = b;
```

Figure 2: Overhead of macro modeling

A simple power macro-model equation for the jth module in the circuit could be expressed as:

$$P_j = 0.5V^2 f \sum_{i=1}^{n_j} C_{i,j} S_{i,j} \tag{1}$$

where f is the clock frequency, n_j is the number of inputs of the jth module, $C_{i,j}$ and $S_{i,j}$ are the effective capacitance and switching activity for the ith pin of the jth module. Note that eqn. (1) is only a typical form of macro-model and is not unique. For example, we can include the spatio-temporal correlation coefficients [7] among circuit inputs to improve the prediction accuracy (this will however increase the number of variables in the macro-model equation and thus the evaluation overhead).

Let $P_{j,k}$ denote the power consumption of the jth module at cycle k. We can also write the macro-model equation in a cycle-by-cycle form as follows:

$$P_{j,k} = 0.5V^2 f \sum_{i=1}^{n_j} C_{i,j} S_{i,j,k} \tag{2}$$

where $S_{i,j,k}$ is the switching activity (0 or 1) for the ith input of jth module at cycle k. The above equation also illustrates that macro-modeling can be used to estimate the power consumption at each cycle, this ability is critical to our statistical approach. We thus distinguish between **one-shot** macro-models (such as eqn. (1)) and **cycle-based** macro-models (such as eqn. (2)).

The total power based on one-shot or cycle-based macro-models can be expressed as:

$$P = \sum_{j=1}^{m} P_j \ \text{ or } \ P_k = \sum_{j=1}^{m} P_{j,k} \tag{3}$$

where m is the number of modules used in the circuit. To calculate $S_{i,j}$, RTL simulation is performed from cycle 1 to cycle N.

Let $I_{j,k}$ denote the input vector for module j at cycle k, $0 \leq k \leq N$. A more general macro-model equation for module j at cycle k can be expressed as:

$$P_{j,k} = \mathcal{F}_j(I_{j,k-1}, I_{j,k})$$

where $\mathcal{F}_j$ could be any function of input vector pairs. Let I_k denote the collection of input vectors, derived from simulation, for m modules at cycle k, $0 \leq k \leq N$. Then total power equation for cycle k is:

$$P_k = \mathcal{F}(I_{k-1}, I_k)$$

where $\mathcal{F} = \sum_j^m \mathcal{F}_j$. In general, the three basic criteria for effective macro-model design are:

1. The space and time complexity for collection of parameter values for F and for each evaluation of this function (should be as small as possible).
2. The accuracy of the macro-model (should be as high as possible).
3. The error sensitivity of the macro-model to variations in population behavior (should not be too sensitive).

3 Background

Population refers to the collection of all input vector pairs $\{(I_1, I_2), (I_2, I_3), \ldots, (I_{N-1}, I_N)\}$ collected during RTL simulation. An *individual* is any vector pair in the population. The *characteristics* are the attributes associated with each individual. For instance, a characteristic value may be the macro-modeling power estimate of an individual (vector pair), or the gate-level power value of the individual. The *characteristic under study* is the attribute we want to estimate over the population, denoted as y. The *auxiliary characteristics* which is used to help predict the characteristic under study is denoted by x.

A part or fraction of the population is said to constitute a *sample*. The number of individuals included in the sample is called the *sample size*. The *population mean* (*total*) refers to the mean (total) value of the characteristic under study for the whole population. The *sample mean* refers to the mean value of some characteristic for a sample The sampling theory is mainly concerned with ways of obtaining samples to efficiently estimate the population parameters. Any function of sample values is called a *statistic*. If it is used to estimate any population parameter, it is called an *estimator*. An estimator is a random variable and may take different values from one sample to next.

The value that the estimator takes on in any particular sample is then its estimate. An estimator t is said to be unbiased estimator for parameter θ if $E(t) = \theta$, otherwise it is biased. Thus the bias is given by $E(t - \theta) = B(t)$. The mean of squares of error taken from θ is called mean-square error (MSE). Symbolically, $MSE(t) = E(t - \theta)^2$. The *sampling variance* of t is defined by $V(t) = E[t - E(t)]^2$ and $MSE(t) = E(t - \theta)^2 = V(t) + B(t)$.

Given two estimators t_1 and t_2 of a parameter, the estimator t_1 is said to be more *efficient* than t_2 if the mean square error of t_1 is less than the mean square error of t_2. The *relative efficiency* of t_1 as compared to t_2 , which differs in respect of sample size or sampling method or both, may be defined as the reciprocal of the ratio of the sampling variances of the estimator given by both techniques when the same number of individuals are taken. The total count of all individuals of the population for a certain characteristic is known as complete enumeration, also termed *census survey*. When only a part, called a sample, is selected from the population and examined, it is called sample enumeration or *sample survey*. The sample survey will be less expensive than a census survey and the desired information will be obtained in less time.

A *confidence interval* is an assertion that the unknown parameter θ lies in a computed range, with specified probability $1 - \alpha$ (called *confidence level*). Let θ be the population mean of y characteristic and let t be an estimator for y. Moreover, the distribution of t is assumed to approach a normal distribution. Take m independent samples and let the sample means for these m samples be $t_1, t_2, \ldots, t_m$.

$$\bar{t} = 1/m \sum_{i=1}^{m} t_i \tag{4}$$

$$S^2 = \frac{1}{m-1} \sum_{i=1}^{m} (t_i - \bar{t})^2 \tag{5}$$

The confidence interval of the estimator t can be calculated by the following procedure [3, p.212]:

1) Determine the critical value $t_{\alpha/2}$ such that $F_{m-1}(t_{\alpha/2}) = 1 - \alpha/2$, where $F_{m-1}(t)$ is the t-distribution with degrees of freedom $m - 1$;
2) Compute the mean $\bar{t}$ and standard deviation S of the sample;
3) Compute $k = t_{\alpha/2} S/\sqrt{m}$; and
4) The $100(1 - \alpha)$% confidence interval for $\overline{y}$ is given by $(\bar{t} - k, \bar{t} + k)$.

Confidence interval estimation is used to estimate the error range of the estimate with certain confidence level. The confidence interval is also used as the stopping criterion in the well-known Monte Carlo simulation method [1]. Monte Carlo simulation procedure continues sampling until the confidence interval of the estimate is less than or equal to a user defined value. When m is large, the number of samples needed is inversely proportional to the variance of estimator t. This also explains why the relative efficiency is defined as the reciprocal of the ratio of the sampling variances of two estimators.

We use the following notation:

N	number of individuals in the population
y	characteristic variate under study
x	auxiliary characteristic variate used to estimate y
ρ	correlation coefficient of x and y
y_i	y value of the ith individual in the population or sample
x_i	x value of the ith individual in the population or sample
Y	population total of characteristic y
$\overline{Y}$	population mean of characteristic y
$\overline{X}$	population mean of characteristic x
S_y^2	population variance of characteristic y, $S_y^2 = \frac{1}{N}\sum_1^N (y_i - \overline{Y})^2$
s	a sample, collection of individuals
n	sample size, $\|s\|$

4 Simple Random Sampling

The simplest form of random sampling from a finite population proceeds in one of the following ways:

1) Individuals are drawn at random, one at a time, and characteristic of interest is determined for each one. After each observation and before the next selection, the individual just drawn is replaced and the population is thoroughly mixed. This kind of samples are called simple random samples *with replacement* (wr).
2) Individuals are drawn and observed as in 1), but they are not replaced. This kind of samples are called simple random samples *without replacement* (wor).

The SRS (wr) is used through this paper to make equations easier to read. In practice, the SRS (wor) can be used instead. When the sample size is much smaller than the population size, the SRS (wr) is the same as SRS (wor).

The sample mean of a simple random sample can be used to estimate the population mean, i.e.

$$\overline{y}_{sr} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6}$$

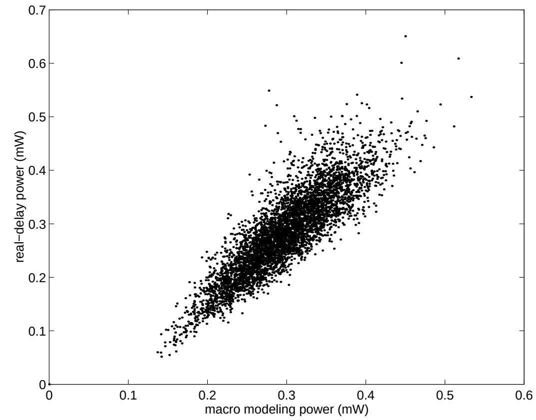


Figure 3: x-y plot of macro-modeling power vs real delay power of a 16-bit adder (on 10,000 vectors).

Theorem 4.1 *In simple random sampling (wr),the sample mean $\overline{y}_{sr}$ is an unbiased estimator of $\overline{Y}$ and its sampling variance is given by*

$$V(\overline{y}_{sr}) = \frac{S_y^2}{n} \tag{7}$$

Eqn. (7) shows that the MSE of simple random sampling is inversely proportional to the sample size. It implies that if n is large enough, we can get very accurate estimation of the original population mean.

5 Regression Estimation

The regression estimator applies to situations where the scatter-plot of y versus x reveals an approximately linear relation of the form:

$$y = \alpha + \beta x$$

Let us study the scatter-plot of gate-level power estimate (y) versus power macro-model equation value (x) for a 16-bit adder (cf. Figure 3). We find that the gate-level power can be roughly predicted by its macro-model equation value and the relationship can be approximated by a line. This motivates the use of a regression linear model.

Our purpose is to estimate the population total Y which can be expressed as:

$$Y = \sum_{i \in s} y_i + \sum_{i \notin s} y_i$$

The quantity $\sum_{i \in s} y_i$ is known (from the samples). The unknown quality $\sum_{i \notin s}$ will obey the approximate relation

$$\sum_{i \notin s} y_i \doteq (N-n)\alpha + \beta \sum_{i \notin s} x_i$$

We now use the method of least-square-error on the available data $\{(x_i, y_i) | i \in s\}$ and find estimates $(\hat{\alpha}, \hat{\beta})$ for (α, β) that minimize

$$\sum_{i \in s} (y_i - \alpha - \beta x_i)^2$$

The solution is

$$\begin{aligned} \hat{\alpha} &= \overline{y} - \hat{\beta}\overline{x} \\ \hat{\beta} &= \frac{\sum_{i \in s}(y_i - \overline{y})(x_i - \overline{x})}{\sum_{i \in s}(x_i - \overline{x})^2} \end{aligned}$$

Hence, we can estimate $\sum_{i \notin s} y_i$ by the quantity

$$(N-n)\hat{\alpha} + \hat{\beta} \sum_{i \notin s} x_i$$

An estimate of the population total is given by

$$\begin{aligned} Y &= \sum_{i \in s} y_i + (N-n)\hat{\alpha} + \hat{\beta} \sum_{i \notin s} x_i \\ &= N(\overline{y} + \hat{\beta}(\overline{X} - \overline{x})) \end{aligned}$$

This leads to an estimator of the population mean as

$$\overline{y}_{lr} = \overline{y} + b(\overline{X} - \overline{x}) \tag{8}$$

In the above equation we have replaced $\hat{\beta}$ with a more familiar notation b which denotes the sample regression coefficient of y on x.

Theorem 5.1 *In SRS, the bias of $\overline{y}_{lr}$ is approximated by*

$$Bias \cong -Cov(b, \overline{x})$$

When the sample size is large, usually the $Cov(b, \overline{x})$ will deceases. It becomes zero if the joint distribution of y and x is a bivariate normal.

Theorem 5.2 *In simple random sampling, wr, the large-sample variance of $\overline{y}_{lr}$ is given by*

$$V(\overline{y}_{lr}) \cong \frac{1}{n} S_y^2 (1 - \rho^2) \tag{9}$$

where ρ is the correlation coefficient of x and y, and the bias is assumed negligible. [2]

Corollary 5.3 *For a large sample size, the variance of regression estimator is less than or equal to the variance of simple random sampling. The equality holds when ρ equals zero. The bias is assumed negligible here.*

The relative efficiency η of regression estimator vs. simple random sample is:

$$\eta = \frac{1}{1 - \rho^2}$$

The higher the ρ value (however, $-1 \le \rho \le 1$), the better the regression estimator.

5.1 A Double Sampling Procedure

To relax the constraint that characteristic x needs to be calculated for all individuals in the population as required by the regression estimator, we propose a two-stage regression estimation procedure. In the first stage, a sub-population s' of size n' is randomly selected (wr). In the second stage, the regression estimator is applied to the first stage sub-population.

The estimator $\overline{y}_{dblr}$ is defined as

$$\overline{y}_{dblr} = \overline{y}(s) - b(s)(\overline{x}(s) - \overline{x}(s')),$$

where subscript *dblr* denotes double-sampling regression.

The exact properties of this estimator on small samples are difficult to study. In general, it is a biased estimator where the bias arises from the term $b(s)(\overline{x} - \overline{x'})$). Assuming that the subpopulation size n' is large enough, we may approximate $\overline{x}(s')$ by $\overline{X}$. Then

$$\begin{aligned} E(\overline{y}_{dblr}) &\cong \overline{Y} \quad (10) \\ V(\overline{y}_{dblr}) &\cong (1/n)S_Y^2(1-\rho^2) + (1/n')S_Y^2 \quad (11) \end{aligned}$$

[2] All proofs can be found in [2].

Let C_x and C_y denote the costs of measuring the x and y characteristics of an individual, respectively. Given a fixed cost C, the optimal allocation problem is to determine optimal values of n and n' such that $V(\overline{y}_{dblr})$ is minimized. This is the same as minimizing

$$V(\overline{y}_{dblr}) + \lambda(n'C_x + nC_y - C)$$

where λ is Lagrange multiplier.

The optimal allocation holds when

$$n' = \frac{C}{C_x + C_y/r}$$
$$n = n'/r$$

where $r = \sqrt{\frac{C_y}{C_x(1-\rho^2)}}$. Note that ρ is not known prior to simulation, but it can be guessed from past experiences.

6 A Statistical Framework

In the previous section, two estimators were introduced: random sampler and regression estimator. The sampler macro-modeling uses simple random sampling to reduce the macro model co-simulator overhead during RTL simulation while adaptive macro-modeling uses regression estimation to improve the estimation accuracy by invoking gate-level simulation on a small number of cycles.

6.1 Sampler Power Macro Modeling

To reduce the run time overhead, one can use simple random sampling to select a sample and evaluate the macro-model equation for all vector pairs in the sample. The sample size is determined before simulation. The sampler macro-modeling randomly selects n cycles and marks those cycles. When the RTL simulator reaches the marked cycle, the power co-simulator is invoked to collect input statistics and calculate the macro model equation value.

The confidence interval of the sample mean can be also derived. However, to calculate the confidence interval, we need more than one sample and the sample mean should be close to a normal distribution. Based on our experiments, the sample mean will approach a normal distribution when the sample size is greater than 30. Instead of selecting only one sample of large size, we select several samples of size 30. Then the estimate of population mean and the sample variance are calculated using eqn. (4) and (5).

Once simulation is completed, the confidence interval for a given confidence level is computed (cf. Section 3). If a fixed-cost approach is taken, the power is reported. Otherwise if the confidence interval is not satisfactory, re-simulation is needed. Note that the cost of re-simulation is very high. An alternative solution is to cache larger set of vector pairs than is though to be necessary for achieving the stopping criterion. If the error level is not satisfied, the co-simulation can take samples from cached vector pairs instead of re-simulation.

6.2 Adaptive Power Macro Modeling

To reduce the error between the power macro-model equation and gate-level power value, one can use a regression estimator. Here characteristic y becomes the gate-level power value and characteristic x the macro-model equation estimate (cf. Section 5).

The regression estimator can achieve very high efficiency if the correlation between the gate-level power value and the macro-model equation estimate is high. The lower the correlation between the two, the slower the convergence rate of the regression sampler. This also points out that the static macro-model equation needs to be designed and trained with great caution. In addition, note that a non-linear regression model can provide better results for certain types of circuits (for example, multipliers).

A double-sampling regression estimator can be built on top of sampler macro-modeling. In the double-sampling regression estimation scheme, n' cycles are selected and marked as 'x' and n cycles randomly selected from the n' cycles and marked as 'y'. Every time the co-simulator encounters cycles marked 'x', the macro-modeling power is evaluated. On the other hand, every time the co-simulation enters cycles marked 'y' then the vector pairs are recorded for later gate-level power simulation. After the RTL simulation is completed, the gate-level simulation is performed on the set of recorded vector pairs. Then, the regression estimation is performed. Similar to sampler macro-modeling, the confidence level and interval can be derived. Note that for the single-stage regression estimator, the power macro-model equation needs to be evaluated for all N, which is not efficient.

We can also use an adaptive macro-modeling approach which would rely on *multi-variate* regression analysis. The variables of regression equation, for example, correspond to input activity and inter-bit correlations. The accuracy of the single-variate regression tends to be lower because of the simple form of the correction to the static macro-model, while its runtime efficiency is higher since it requires smaller number of gate-level simulations and has lower cost associated with the regression analysis itself.

Adaptive macro-models can provide the designers with information (i.e. the confidence level) about the accuracy of predicted value for a given input sequence which is desirable. Adaptive macro-models also guarantee to eliminate the estimator bias given enough gate-level simulations, hence, they tend to be more accurate.

7 Experimental Results

16-bit bus-width is assumed in all of our benchmarks. The macro model we used for the 16-bit adder (subtractor) follows eqn. (1). For the multiplier which has a lot of glitches, we used a multi-variable function to partially account for the signal correlations. All of our results are based on input vector length of 10,000. Full gate-level simulation and census macro-modeling were performed on each circuit for the sake of comparison. The benchmarks shown in Tables 1, 2, and 3 are Differential Equation Solver, Elliptic Filter, Chebyshev Filter, IIR Filter, Discrete Cosine Transform, Robotic Arm Controller, and Adaptive Transversal Filter that are commonly used in high-level synthesis benchmarks. Scheduling and resource binding were performed on these data flow graphs. Confidence level of 95% and error level of 5% were used in Tables 1 and 3. 1,000 runs were performed on each benchmark. All results were generated on a Pentium-120 machine.

Figure 4 shows the plot of error percentage of sampler macro-model power value with respect to census macro-model power value as a function of sample size in the sampler macro-modeling. It can be seen that the error of a sample size > 100 is less than 2%.

Table 1 compares the accuracy of the sampler macro-modeling with that of census macro-modeling. The 'max', 'min' and 'avg' columns show the maximum, minimum and average numbers of required samples that satisfies the stopping criterion. The '$err > 5\%$' column shows the percentage of number of experiment runs that violate the error level. The 'avg err(%)' column shows the average percentage error of the power estimate with respect to the census macro-modeling. The 'speedup' is the simulation time ratio of census macro-modeling over sampler macro-

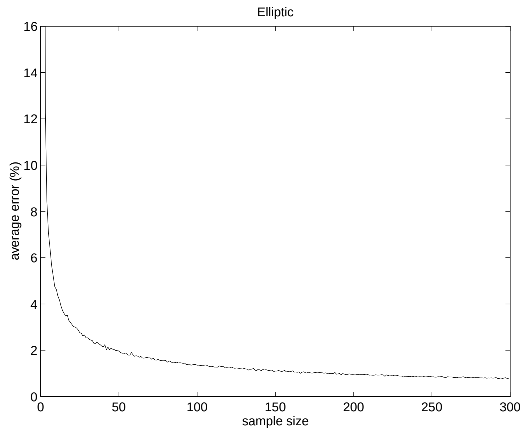


Figure 4: Error percentage vs sample size (sampler macro-modeling)

Table 1: Sampler macro-modeling efficiency (error) compared to census macro modeling (sampe size=30)

bench-mark	max, min	avg	err >5%	%avg err	% bias	speed up
DiffEq	8, 2	3.7	0.1	1.2	-0.04	47.9
Ellip.	6, 2	3.0	0.0	0.8	-0.01	53.2
Cheby.	6, 2	2.9	0.9	0.7	-0.02	53.6
IIR	5, 2	2.8	0.0	0.6	-0.03	54.8
FDCT	8, 2	3.6	0.2	1.2	-0.06	48.0
Robot	5, 2	2.8	0.0	0.7	-0.05	54.8
ATF	8, 2	3.7	0.2	1.2	-0.07	47.8
average	7, 2	3.2	0.2	0.9	-0.03	51.4

modeling. The average speedup of 50X was observed.

Table 2 shows the error percentage of census macro-modeling w.r.t. gate-level simulation power. Results show that the error of macro model is sensitive to the population, and hence is not very reliable.

In Table 3, the error percentage (with respect to gate-level simulation) and the run time of adaptive macro-modeling with two-stage design is reported. The sub-population size was set to 1000 and the sample size was set to 30. The results show that the bias is negligible as expected. Comparing the mean error of adaptive macro-modeling with census macro-modeling, the accuracy of the former is improved by a factor of 16X.

8 Conclusion

In this paper, we presented a statistical framework for RTL power evaluation. Two new power macro-modeling techniques are proposed: the sampler macro-modeling based on the sampling theory and the adaptive macro-modeling based on the regression analysis. Experimental results demonstrated that, compared to census macro-modeling, sampler macro-modeling reduces the simulation time by a factor of 50X while the adaptive macro-modeling lowers the estimation error by a factor of 16X.

There are two situations that may be encountered:

Table 2: Census macro-modeling error compared to gate-level simulation

benchmark	DiffEq	Ellip.	Cheby.	IIR
error(%)	23.1	26.9	30.3	61.8
benchmark	FDCT	Robot	ATF	average
error(%)	1.3	64.4	2.1	30.0

Table 3: Adaptive macro-modeling compared gate-level simulation (sampe size=30)

bench-mark	max, min	avg	err >5%	%avg err	% bias	run time
DiffEq	20, 2	4.2	5.2	1.9	0.18	15.4s
Ellip.	13, 2	3.6	2.2	1.6	0.09	11.1s
Cheby.	11, 2	3.5	0.9	1.4	0.04	11.0s
IIR	15, 2	4.1	2.7	1.7	0.02	13.0s
FDCT	15, 2	3.8	3.4	1.8	0.18	38.8s
Robot	16, 2	3.7	3.8	1.8	0.01	40.4s
ATF	24, 2	7.2	8.6	2.2	0.12	22.2s
average	16, 2	4.3	3.8	1.8	0.09	21.7s

1) The macro-model equation estimate is very accurate (with respect to the gate-level power estimate), but the overhead associated with collecting the input statistics and evaluating the macro-model equation is high. The sampler macro-modeling must be used.

2) The macro-model equation estimate is not accurate, yet its computational overhead is low. The adaptive macro-modeling must be used here.

There is obviously a trade-off between the prediction accuracy and evaluation cost of the static macro-model. This paper suggests that one can use a low-cost static macro-model and then reduce the error by regression sampling or can use a high-cost static macro-model and then improve the efficiency by simple random sampling.

References

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

[2] C-S. Ding, C-T. Hsieh, Q. Wu, and M. Pedram. Statistical techniques for power evaluation. In *CENG Technical Report No. 96-11, University of Southern California*, 1996.

[3] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics, Fourth Edition*. Macmillan Publishing, 1978.

[4] P. Landman and J. Rabaey. Power estimation for high-level synthesis. In *Proceedings of IEEE European Design Automation Conference*, pages 361–366, Feb. 1993.

[5] P. Landman and J. M. Rabaey. Activity-sensitive architectural power analysis for the control path. In *Proceedings of International Symposium on Low Power Desing*, pages 93–98, April 1995.

[6] D. Liu and C. Svensoon. Power consumption estimation in CMOS VLSI chips. In *IEEE Journal of Solid State Circuits*, volume 29, pages 663–670, June 1994.

[7] R. Marculescu, D. Marculescu, and M. Pedram. Logic level power estimation considering spatiotemporal correlations. In *Proceedings of the IEEE International Conference on Computer Aided Design*, pages 294–299, Nov. 1994.

[8] S. Powell and P. Chau. Estimating power dissipation of VLSI signal processing chips: The PFA techniques. In *Proceedings of IEEE Workshop on VLSI Signal Processing IV*, volume IV, pages 250–259, 1990.