POWER ESTIMATION FOR REGISTER TRANSFER LEVEL BY GENETIC ALGORITHM

Yaseer A. Durrani, Teresa Riesgo, Felipe Machado
Universidad Politécnica de Madrid
E.T.S.I. Industriales. División de Ingeniería Electrónica
C/ José Gutiérrez Abascal, 2. 28006 Madrid (Spain)

Keywords: Power estimation, Macromodel, Signal statistics, Genetic algorithm.

Abstract: In this paper, we propose a new genetic algorithm (GA) based macromodeling technique for register transfer level. This technique allows to estimate the power dissipation of intellectual property (IP) components using some statistical knowledge of their primary inputs. During the macromodel construction procedure, the sequence of an input stream is generated by a GA using input metrics. Then, a Monte Carlo zero delay simulation is performed and the power dissipation is predicted by a macromodel function. In our experiments with IP macro-blocks, the results are effective and highly correlated, with an average error of 1%. Our model is parameterizable and provides accurate power estimation.

1 INTRODUCTION

The importance of designing low power digital circuits is being increased rapidly. In order to handle the ever increasingly complexity, CAD tools have been developed that also help in minimizing power dissipation.

In this short paper, we focus on the problem of power estimation at register transfer level (RTL) for IP-based designs. Various power estimation techniques have been introduced previously. These techniques can be divided into two categories: *probabilistic* and *statistical*. Probabilistic techniques (Ghose *et al*, 1992), (Najm *et al*, 1990), (Marculescu *et al*, 1994) use the probabilities of the input stream and their propagation into the circuit to estimate the internal switching activities. These techniques are very efficient, but they cannot accurately capture factors like glitch generation and propagation. On the other hand in statistical techniques (Yacoub & Ku, 1989), (Huizer, 1990), (Deng, 1994) the circuit is simulated under randomly generated input patterns and the power dissipation is monitored using a power estimator. The Monte Carlo simulation technique was developed and presented in (Burch *et al*, 1993). This technique uses input vectors that are randomly generated and its power dissipation is estimated using power estimator.

A look-up table (LUT) based macromodel was presented in (Gupta & Najm, 1997) and further improved in (Gupta & Najm, 1999). The LUT stores the estimates for equi-spaced discrete values of the input signal statistics. The interpolation method was introduced to allow the estimation for the input statistics that not correspond to LUT. In (Chen *et al*, 1997), (Chen & Roy, 1998) interpolation scheme was improved by using power sensitivity concept. For better accuracy, numerous power macromodeling techniques have been introduced in (Gupta & Najm, 1999), (Liu & Papaefthymiou, 2002).

Genetic Algorithms (Davis, 1991) have proved success in solving electronic design problems (O'Dare & Arslan, 1994), (Arslan *et al*, 1996) and have shown a high degree of flexibility in handling power constraints (Davis, 1991). They are more dynamic to combine power of randomness and evolution, and to analyze large solution space.

In this paper, we present a new genetic algorithm based power macromodeling technique for power estimation. The input metrics of our macromodel are the *average input signal probability* P_{in}, *average input transition density* D_{in}, *input spatial correlation* S_{in} and *input temporal correlation* T_{in}. We use intellectual property (IP) macro-blocks for our experiments.

A. Durrani Y., Riesgo T. and Machado F. (2006).
POWER ESTIMATION FOR REGISTER TRANSFER LEVEL BY GENETIC ALGORITHM.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 527-530
DOI: 10.5220/0001214205270530

The rest of this paper is organized as follows. In Section 2 we give the background of the input parameters of our macromodel. In Section 3, we explore our genetic algorithm. In Section 4, we discuss about our macromodel construction. This macromodel is evaluated in section 5. Section 6 summarizes our work.

2 CHARACTERIZATION

Similar power macromodeling techniques were presented in (Chen & Roy, 1998), (Gupta & Najm, 1999), (Liu & Papaefthymiou, 2002). Our macromodel consist of a nonlinear function based on LUT approach:

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

The function f is obtained by a given IP macro-block. The components are simulated under different sample streams with, P_{in}, D_{in}, S_{in}, T_{in}. This model estimates the average power dissipation.

Given an IP macro-block with the number of primary inputs r and the input binary stream $q = \{(q_{11}, q_{12}, ..., q_{1r}), (q_{21}, q_{22}, ..., q_{2r}), ..., (q_{s1}, q_{s2}, ..., q_{sr})\}$ of length s, these metrics are defined in (Gupta & Najm, 1999), (Gupta & Najm, 1999), (Barocci *et al*, 1999), (Bernacchia & Papaefthymiou, 1999):

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

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

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

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

3 EXPLORATION ALGORITHM

In this section, we analyze our genetic algorithm for power macromodeling allowing to estimate the power dissipation. The proposed GA for the IP system is presented in Figure 1. The flow of our GA is explained in next sections.

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

Figure 1: Genetic Algorithm (GA).

3.1 Setting Chromosome

In GA process, chromosomes are exposed to genetic operators like crossover, mutation, selection etc. The objective of these operations is to remove poor strings and produce healthy strings. In this step, we makeup of a given chromosome. It includes the number of genes, and the representation of those genes. In our problem, the chromosomes are the primary inputs.

3.2 Fitness Function Implementation

The fitness function is responsible for performing evaluation and returning fitness value that reflects how optimal the solution is: the higher the number, the better the solution. Given fitness values are then used in a process of natural selection to choose which potential solutions will continue on to the next generation, and which will die out. The natural selection process does not choose the top n number of solutions; the solutions are instead chosen statistically that are more likely with a higher fitness value. Our algorithm is used with measurements to evolve the population of solutions toward a more optimal set of solutions.

3.3 Setup and Creating Population

GA needs to provide three extra pieces of information like fitness function, chromosome setup, and the chromosome population. A large number of population size means more potential solutions to choose and more generic diversity. Each potential solution is represented by a chromosome. A population of chromosome is called a Genotype and that is we need to create our population. We setup a configuration object and return a Genotype with the correct number of chromosomes, each one of this has its genes set to random values. In other words, it generates a random population. It is necessary to create initial population for our potential solutions. We have specified an initial population as an input of GA. The initial population contains M number of random strings of length N, where M and N are the input parameters used in GA.

3.4 Evolutionary Process

After setup and creation of population, our GA evolves the population until it contains satisfying potential solutions. We choose to evolve the population a set number of times and then we check what is the best solution produced by our genetic algorithm.

4 MODEL CONSTRUCTION

Several approaches (Chen & Roy, 1998), (Gupta & Najm, 1999), (Liu & Papaefthymiou, 2002) have been proposed to construct power macromodel on ISCAS-85 benchmark circuits. We have found that the same methodology works as well for IP macro-blocks such as array multipliers or comparators in terms of the statistical knowledge of their primary inputs. By the following method (Gupta & Najm, 1999), we found that a good choice of the P_{avg} in (1), is quadratic for array multipliers and comparators respectively. The sequence of input stream is generated by GA for a desired input metrics. Then, a Monte Carlo zero delay simulation (Burch *et al*, 1993) is performed and the power dissipation is obtained using a power estimator. Different input stream sequences are generated for different signal statistics and the corresponding power consumptions are measured. Using all these measures, the function f in (1) is calculated.

Table 1: Accuracy of Power Estimates.

Circuits	Average Error	Max Error
Mult8x8-1	0.76%	2.35%
Mult8x8-2	1.50%	3.19%
Mult4x4-1	0.67%	2.30%
Mult4x4-2	2.15%	3.00%
Comp-1	1.44%	2.95%
Comp-2	0.33%	1.43%
Comp16x16-1	0.58%	0.95%
Comp16x16-2	0.47%	0.64%

5 MACROMODEL EVALUATION

Experimental results show that our GA can produce sequences with accurate statistics and highly convergence. For the input metrics P_{in}, D_{in}, S_{in}, T_{in}, we specify the range between [0.1, 0.9]. We generated 450 sequences with 8, 16 and 32 bits wide. The sequence length is 2000 and 1000 vectors for macro-blocks. In table (1) the first column shows the name of the circuits. Columns two and three give the average and maximum relative error for the estimates obtained with our macromodel. The reference values for the circuit's power dissipation are obtained using time delays from the Synopsys PowerCompiler. In our experiments, the average absolute error is 0.98%, and the average maximum error is 2.1%. The maximum worst-case error is no more than 3.19%.

In figure 2 we illustrate the combined scatter plot of IP macro-blocks between the macromodel and the reference simulated power. Regression analysis to fit the model's coefficients is performed. For different blocks, the prediction correlation coefficient is measured around 97%. The sequences have high convergence and uniformity. Figure 3 plots the variation of the power value with the trial interval length. This figure shows that the interval length is 1000. The warm-up length is about 400 while the vertical line represents the steady state value at 800.

6 CONCLUSIONS

We have presented a new genetic algorithm based power macromodeling technique for high-level power estimation. Our technique has been applied on IP macro-blocks and has demonstrated good accuracy. Our model shows an average error of 1% and a prediction correlation coefficient of 97%. We are currently evaluating our macromodel on sequential circuits.

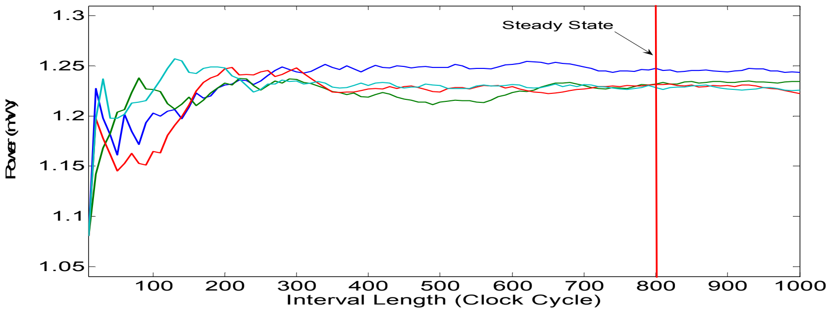

Figure 2: Power comparison between four dimensional macromodel and reference simulated power.

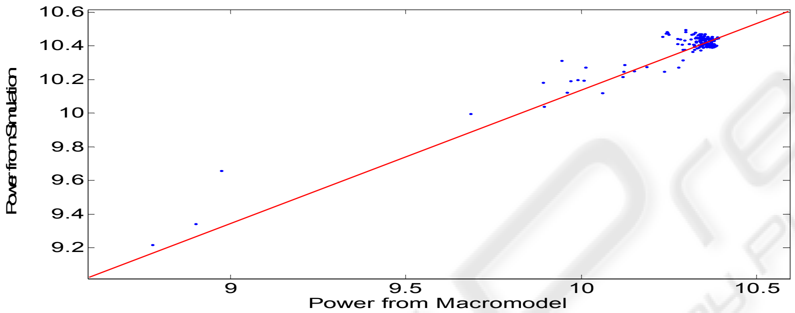

Figure 3: Power changes with respect to sequence length for different IP blocks.

REFERENCES

Ghose, A. A., Devdas, S., Keutzer, K., & White, J., June 1992. Estimation of average switching activity in combinational and sequential circuits. In *Proceedings of the 29th Design Automation Conference*, pages 253-259.

Marculescu, R., Marculescu, D., & Pedram, M., November 1994. Logic level power estimation considering spatiotemporal correlations. In *Proceedings of the IEEE International Conference on Computer Aided Design,* pages 224-228.

Yacoub, G.Y., & Ku, W. H., 1989. An accurate simulation technique for short-circuit power dissipation. In *Proceedings of International Symposium on Circuits and Systems*, pages 1157-1161.

Deng, C., April 1994. Power analysis for CMOS/BiCMOS circuits. In *Proceedings of 1994 International Workshop on Low Power Design*, pages 3-8.

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

Gupta, S., & Najm, F. N., June 1997. Power macromodeling for high level power estimation. In *Proceeding 34th Design Automation Conference.*

Gupta, S., & Najm, F. N., March 1999. Analytical model for high level power modelling of combinational and sequential circuits. In *Proceeding IEEE Alessandro Volta Workshop on Low Power Design*.

Chen, Z., Roy, K., & Chou, T. L., Nov. 1997. Power sensitivity-a new method to estimate power dissipation considering uncertain specifications of primary inputs. In *Proceeding IEEE International Conference on Computer Aided Design.*

Chen, Z., & Roy. K., June 1998. A power macromodeling technique based on power sensitivity. In *Proceeding 35th Design Automation Conference.*

Gupta, S., & Najm, F. N., 1999. Power Macromodeling for High Level Power Estimation. *In Proceeding IEEE Transactions on VLSI.*

Liu, X., & Papaefthymiou, M. C., May, 2002. Incorporation of input glitches into power macromodeling. *In Proceeding IEEE Inter. Symp. On Circuits and Systems*.

Davis, L., 1991. *Handbook of Genetic Algorithm*. Van Nostrand Reinhold, New York.

O'Dare, M.J., & Arslan, T., May 1994. Generating test patterns for VLSI circuits using a Genetic Algorithm. *In Proceeding IEE Electronics Letters,* Vol. 30, No. 10, pp. 778-779.

Arslan, T., Horrocks, D. H., Ozdemir, E., 1996. Structural cell-based VLSI circuit design using a Genetic Algorithm. *International symposium on circuits and systems, Atlanta, USA*.