RT LEVEL POWER CONSUMPTION ESTIMATION TOOL

Jaroslav Škarvada
Doctoral Degree Programme (3), FIT BUT
E-mail: skarvada@fit.vutbr.cz

Supervised by: Zdeněk Kotásek
E-mail: kotasek@fit.vutbr.cz

ABSTRACT

This paper deals with RT level power consumption estimation during test application. The goal of this paper is to describe design and implementation of custom power consumption estimation tool that works in collaboration with Mentor Graphic DfT tools. The proposed tool can be used in an early design stage for quick classification of various test sets according to power consumption without need of full synthesis and slow physical level simulation of the design. The tool is implemented as a library that makes it ideal for usage in various power consumption optimization techniques.

1. INTRODUCTION

The need for low power methodologies is still increasing and is mostly driven by the increase in the level of integration. The one reason for this is that market share of mobile devices still grows (only in 1998 mobile devices cover 32% of PC market [1]) Portable devices are mostly driven from batteries, but the battery technology develops slowly over the time and it can be stated that batteries are bottleneck of most portable systems today. Thus for the long operational time the low power design optimizations are highly welcome.

Another good reason for low power optimizations are thermal requirements. Since much of the power consumed by the circuit is dissipated as heat, the relationship between the power consumption and the cooling capacity needs to be taken into account. It is worth to note that 10°C increase in operating temperature leads to doubling the component failure rate [1]. In diagnostic we must also beware of the chip power dissipation limit in order to avoid false error detection during test or even worse destructive test [1] (after several test applications). High power consumption can lead to some reliability issues such as electromigration, voltage drops on supply lines, inductive effects, hot electron effects, etc. [1]. Using low power design methodologies (e.g. from [2]) it is possible to significantly reduce power consumption during normal (functional) mode of circuit operation, but in the test mode this is not so straightforward. As described in various papers (e.g. [3]), the chip under test consumes more power during testing in comparison to normal mode of operation. Greater power dissipation is caused by significantly higher switching activity (when assuming CMOS technology as dominating fabrication technology for implementation of ICs that contain more than 105 transistors [3]). During testing there can be detected much more switching activity due to much lower correlation between input patterns in comparison to functional mode. It can be stated that the vast majority of power reduction techniques con-

centrate on reducing the dynamic power dissipation by minimizing the switching activity in the circuit under test. For successful usage, development and comparison of these techniques, the quick and accurate power estimation tool is needed.

2. RECENT WORKS

Previous RTL power estimation approaches can be classified into three broad categories [4], namely: analytical techniques, characterization macro modeling and fast synthesis based estimation.

Analytical power modeling techniques uses very little information from the functional specification. The power consumption is estimated from general parameters such as a gate count, number and types of logic used, etc. It includes power consumption estimation according to entropy of input and output signals and statistical information such as probability of logical states [5]. These methods are very fast but not too accurate and are rather useful in the very early design stage.

Characterization based macro modeling methods constructs "black boxes" of various macro blocks of circuit logic. Under these macro blocks various sequences of input patterns are trained. Gate level or transistor level tool is used for estimation of power consumption of macro blocks. Based on this data the macro block model for the "black box" is constructed. Early methods simply uses number that denotes the maximal or average power consumption for the block, later methods uses function of statistical parameters of the input signal [6].

Fast synthesis based methods use limited synthesis of the RTL designs. These methods map the design to defined meta-library which consists of small number of primitives. All of them have defined parameters and known power consumption for various input stimulus. Resulting power is obtained by gate-level simulation or probabilistic techniques over these primitives.

3. BASIC DEFINITIONS

With recent designs on the focus it can be stated that the main part of power consumed by CMOS gates are dynamic part of power (about 80% of power) [5]. It is supposed that this number will lower over time as threshold voltages lower (due to supply voltages lowering) that will lead to increase in static power consumption. Physically, the dynamic part of power (P_{dp}) is composed of capacitive switching power (P_{sw}) and short circuit power (P_{sc}) and can be computed by equation (1) (from [1]), where C_L is overall capacitance of gate output, lines and connected gate inputs; V_{dd} is supply power; N is number of transition ($0\rightarrow1$, $1\rightarrow0$) count; f is frequency of the clock signal, K is constant that depends on transistor; V_T is magnitude of threshold voltage; τ is the input rise/fall time.

$$P_{dp} = P_{sw} + P_{sc} = N f \left(\frac{1}{2} C_L V_{dd}^2 + K (V_{dd} - 2V_T)^3 \tau \right) \qquad (1)$$

The equation (1) is too complex for being used for quick power consumption estimation in the early of the design process and in addition some parameters depend on physical properties of used transistors and overall physical layout. That's why some simplificated form needs to be used. For comparison of designs that will be synthesized to the same technology, we can omit the input signal characteristics, transistor constants as well as supply and

threshold voltages. For simplification we assume, that the input capacitance of all gates in design is nearly the same. We also suppose that the clock frequency doesn't change during the test. When we are relatively comparing power consumption of two or more modifications of the same design (or the same design with different test vectors applied) running with the same clock speed it is also possible to omit the frequency parameter. Thus for comparison of influence in design modification or test vectors ordering on power consumption, it is possible to use the WTC (Weighted Number of Transition Count) equation (2).

$$WTC = \Sigma_{i=1..Ng} \left(N_i \times F_i\right) \quad (2)$$

Where N_g is number of gates in the circuit, F_i is number of fan-out factor for gate i. The *NTC* (Number of Transitions Count) can be calculated in the same way with F_i supposed to be one. The scan chain transitions during serial scan-in/scan-out must be also taken into account. The equation (3) can be used for quick scan switching activity estimation. The precondition for usage of equation (3) is that isolable scan registers (registers with SO and Q outputs, that doesn't change the output Q value during serial scan phase) are used.

$$NTC(V_i) = \Sigma_{j=1:(m-1)} \left(V_i(j) \text{ XOR } V_i(j+1)\right) . A \quad (3)$$

- V_i is vector of m bits that is scanned in/out to/from scan chain (with length of m registers)
- Parallel load/unload phase *WTC* can be calculated with equation (2)
- For serial scan-in phase : $A = j$
- For serial scan-out phase: $A = (m - j)$

4. PRINCIPLE OF OPERATION

The estimation tool works on principle that is mixture of the last two approaches mentioned in chapter 2. Models for the AMI 0,5μm library are exploited. The professional synthesis tool Mentor Leonardo Spectrum is used for mapping the design in various level of abstraction into the AMI library. The resulting netlist (in structural Verilog) can be read by described tool as well as the AMI power library. The AMI power library defines tables of transitions for all AMI primitives. Each table consists of exhaustive list of input signal levels ($a_i(t)$) and signal history in (t-1) with appropriate *NTC* (simulated on gate level) for defined signal combination. For registers the internal state (s) is added and it is manipulated in the same way as other input signals. Each table field can be 0, 1 or X (for don't care). Simple example can be seen in table 1. The tool considers only one internal control register (because no more are used in AMI library), but it can be easily extended in the future.

The power library is stored in human readable format so it is possible to use custom models of various elements. When the library is read, it is stored in the memory as fast lookup table. The power estimation tool is able to compact the input netlist for the simulation purposes. It tries to group together as much combinatorial logic as possible. It groups the logic as long as there is available memory and no more than one sequentional element in the group.

a_1 (t-1)	a_2 (t-1)	...	a_n (t-1)	s (t-1)	a_1 (t)	a_2 (t)	...	a_n (t)	s (t)	q (t)	NTC
0	0	...	0	0	0	0	...	0	0	1	10
...	...	...	...	...	...	...	...	...	...	...	...
1	X	...	1	X	1	1	...	1	1	1	5

Table 1: Example of the table used in power library

Then the functional table for the overall group is created. In this step the *NTC* for each element from group is multiplied by fan-out factor of this element to form *WTC*. At the end the resulting table is constructed from the partial tables and then compacted. In the compaction step the rows of the table are scanned. When the two rows of the table have same transition count and are compatible (in signal transitions), these rows are grouped together with don't care value set in columns that doesn't contribute to transitions and the two old rows are combined to form one new row. By this approach the group of logic between registers is "black boxed" and the power consumption for this block can be quickly estimated by simply table lookup. The accuracy of the tool is also acceptable, because accurate power models are used for primitives from the power library. The only drawback is memory requirements for the lookup tables. The tool can read the testing patterns in ASCII text format generated by Flextest and sequentially apply it to the design for predefined number of clock period. The scan chains can be also simulated. For the scan chains the equation (3) is used. As the default, the zero delay model is used for primitives, so no glitches and spurious (hazardous) transmission are count to *NTC*. It is also possible to use non-zero delay model and count every transmission by disabling the "black boxing" behavior. It can be done by special configuration parameter. The delay model for every primitive is than read from the AMI library (or can be globally overridden by config). The tool can report overall *NTC* or *WTC*.

5. EXPERIMENTAL RESULTS

The proposed method was successfully implemented in C++ language with help of the STL library. Experiments were carried on PC (AMD64 Athlon processor @ 2.0GHz, 1GB RAM, Linux). The circuits from ISCAS89 High Level sequentional benchmark set were used as input for the proposed method. The benchmarks circuits were synthesized in Mentor Leonardo Spectrum and mapped to ASIC AMI 0.5μm technology library. The test vectors set were generated by FlexTest tool and analyzed by the described power estimation tool. Full scan test strategy is used with maximal scan length set to the number of flip/flops. The table 2 displays statistics for circuit after mapped to AMI library.

Benchmark name [-]	No of flip/flops [1]	No of nodes [1]	No of I/Os [1]	No of primitives [1]	Fault coverage [%]	No of test vectors [1]	*NTC* [1]
s208	8	111	10/1	112	98.59	36	1980
s298	14	127	3/6	133	100.00	41	4222
s344	15	164	9/11	175	99.70	32	4160

Table 2: Experimental results

6. CONCLUSION

In the previous chapters the method for RTL power consumption estimation during test application was described. So far it was successfully implemented. The AMI 0.5μm power library that is used by this method was also created. The implemented tool can be used in cooperation with Mentor Graphic DfT tools because compatible formats for data interchange were used. The main engine of the tool is implemented as a class so it can be also used as a helper in various power consumption optimization method implementations for comparison of quality of proposed solutions. Namely we are now trying to exploit this tool in process of RTL circuit partitioning to testable blocks where results from the power estimation tool will influence the calculation of the fitness function that controls the circuit partitioning.

REFERENCES

[1] Raghunathan, A., et al.: High-Level Power Analysis and Optimization, Boston, Kluwer Academic Publishers 1998, ISBN 0-7923-8073-8

[2] Schmitz, M. T., et al.: System-Level Design Techniques for Energy-Efficient Embedded Systems, Boston, Kluwer Academic Publishers 2004, ISBN 1-4020-7750-5

[3] Nicolici, N., Al-Hashimi, B. M.: Power-Constrained Testing of VLSI Circuits, Boston, Kluwer Academic Publishers 2003, ISBN 1-4020-7235-X

[4] Ravi, S., et al: Efficient RTL Power Estimation for Large Designs, In: Proceedings of the 16th International Conference on VLSI Design, Washington IEEE Computer Society 2003, ISBN:0-7695-1868-0

[5] Roy, K., Prasad, S. C.: Low-Power CMOS VLSI Circuit Design, New York, John Wiley & Sons, Inc. 2000, ISBN 0-4711-1488-X

[6] Bogliolo, A., Benini, L.: Robust RTL Power Macromodels, In: IEEE Trans. VLSI Systems, vol. 6, 1998, ISSN 1063-8210