A-B NODES CLASSIFICATION FOR POWER ESTIMATION

*Elías Todorovich and Eduardo Boemo**

School of Engineering
Universidad Autónoma de Madrid
Ctra. Colmenar km. 15, (28049) Madrid, Spain
email: etodorov@uam.es, eduardo.boemo@uam.es

ABSTRACT

In this paper, an optimization for the classical statistical power estimation method is proposed. This technique is applied to the individual nodes. The optimization is based on two observations. Firstly, a small percentage of both the nodes and the estimated power requires nearly a half of the total simulation time. On the other hand, the statistical method produces results with better accuracy than those specified by the user. This additional precision enables to reduce the run time for the slow convergence nodes with no loss of accuracy. A simple partitioning of the nodes into two groups, A and B, with normal and high computational cost respectively, leads to a modified stopping criterion with dramatic savings in the run time.

1. INTRODUCTION

Power consumption is one of the most important design goals together with area and speed in VLSI circuits. This is particularly true for FPGAs where programmability increases the number of transistors per logic gate.

Power-aware design flows require optimization and estimation techniques at all the abstraction levels. In particular, gate-level power estimation methods can be based on statistics or probabilities propagation [1-2]. Probabilistic techniques are fast but can generate low accurate results. On the other hand, statistic-based techniques are accurate and easy to implement using standard simulators with delay models ranging from zero to post place and route (routed delays). However, the main drawback is the execution time as is experimentally shown here. This paper is focused on FPGA technology where there are contributions both for probabilities [3] and statistics-based methods [4]. It is important to note that for FPGAs, gate-level representations can be automatically obtained from synthesizable RTL descriptions.

In statistic-based techniques, randomly generated input patterns are applied to the circuit inputs, whilst the activity per time interval T is monitored by the simulator. The process continues until a stopping criterion is reached. This criterion determines the sample size and thus, the execution time. A stopping criterion is derived under some statistical assumptions like in [5] where the normality of the individual nodes average activity is supposed.

It has been experimentally verified that the nodes with high logic activity rapidly converge given some error and confidence level specification [6]. Nevertheless, these times are significantly higher for low activity nodes. The first approach to solve this slow convergence problem is presented in [5], where the nodes with less activity than a threshold η_{min} are considered low-activity nodes. For these nodes, an absolute error bound $\eta_{min} \times \varepsilon$ is obtained, where ε is the user-specified percentage error for regular nodes. Even with this improvement, high execution times are observed while the accuracy is exceeded for regular nodes. In [7] authors present efficient sampling techniques for estimating the total power consumption of large hierarchical circuits. In [8], circuit nodes are partitioned in M groups according to their contributions to the total power dissipation, gradually decreasing the error to the high power groups. This error-to-group assignment is computed using a quadratic programming formulation.

In this paper, a statistical power estimation technique for individual nodes with a new and simpler partitioning method is proposed. It is based on the experimental observation that a very small percentage (1-2%) of the nodes, and the total power, requires a significant additional execution time. In this way, a simple partitioning method into two groups is derived, A and B, with regular and high computational cost respectively. Although the method can be applied to general CMOS designs, the experimental results are obtained from circuits implemented on Xilinx FPGAs.

2. A-B CIRCUIT NODE PARTITIONING

The stopping criterion for regular nodes in [5] is

$$N \geq \left(\frac{z_{\alpha/2} s}{\overline{n} \varepsilon_1} \right)^2, \qquad (1)$$

*This research has been financed by Project 658007 of the Fundación General de la Universidad Autónoma de Madrid.

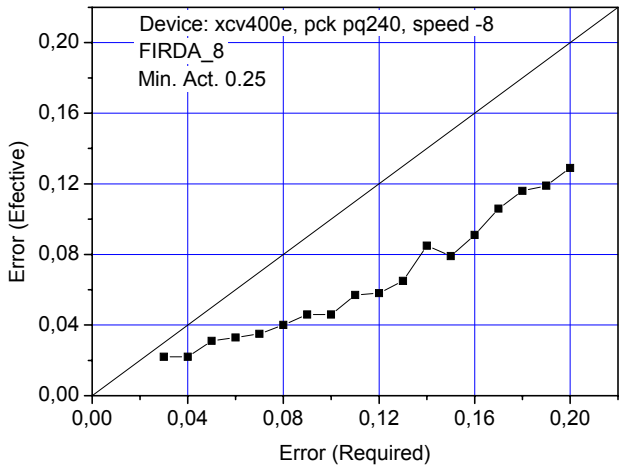

Fig. 1. Effective vs. specified accuracy for the FIRDA_8 test circuit (Table 1) applying the method defined in [5].

where N is the required sample size, $\overline{n}$ and s are the average activity and standard deviation of the random sample, respectively, and $(1 - \alpha) \times 100\%$ is the confidence level that error ε_1 in the estimation is less than a specified value. Finally, $z_{\alpha/2}$ is obtained from the normal distribution.

As the required accuracy increases (ε_1 decreases and, $(1 - \alpha)$ and as a result $z_{\alpha/2}$ increases) the sample size increases, but this is a user decision. Nevertheless, with fixed error for all nodes, the $s/\overline{n}$ ratio can require big samples as $\overline{n}$ decreases and s increases. For this reason, some partitioning of the nodes in M groups, with $M \geq 2$, is necessary to guarantee the accuracy for high activity nodes while the execution time for the lowest active ones is bounded.

In this paper it is shown firstly that applying the estimation technique proposed in [5], the user-required accuracy is exceeded. This is due to the highest activity nodes, which converge earlier in the estimation process, and are over-analyzed, while the low-activity nodes need much more simulation time. In order to quantify this effect, it is interesting to define some "effective" accuracy value that reflects the obtained accuracy with the statistical estimation tool, in opposition to the required accuracy. Fig. 1 shows an example of the relationship between these two variables.

Error and confidence level can be specified independently. However, in order to define the effective accuracy, some normalization is applied to tie together these two user-defined parameters. The confidence-error pairs will be 99/1, 98/2, 97/3… 100-ε%/ε%. This simplifies the study without any loss of generality because the accuracy and simulation time depend on the $z_{\alpha/2}/\varepsilon_1$ quotient. Now, given a power estimation run, it can be defined ε_c values such that the number of nodes with relative error higher than ε_c is less than $\varepsilon_c \times 100\%$ of the nodes. Then, the effective accuracy ε_{ef} is defined by the maximum ε_c that can be obtained from the estimation results. It means that the best precision this given run could satisfy is within a specification where the relative error is less than ε_{ef} with confidence $(1 - \varepsilon_{ef}) \times 100\%$. In Fig. 1 it is observed how the obtained accuracy is always higher than the specified one. This gives the chance to propose optimizations without loss of accuracy. An e value at x and y-axis means an accuracy of e×100% error with (1-e)×100% confidence. The effective accuracy is on average 1.8 times better than the user defined one for the test case in Fig. 1, computed according to (2):

$$\sum_{i=1}^{p} \frac{e_i}{e_{eff,i}} \Big/ p, \qquad (2)$$

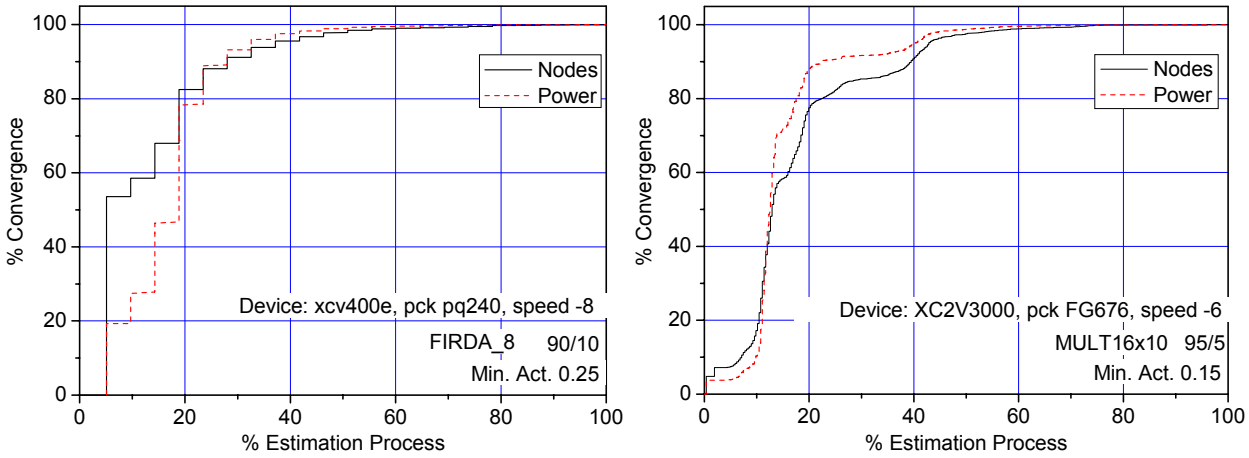

Fig. 2. Nodes convergence for the FIRDA_8 test circuit (left) with 90% accuracy that error is less than 10%, and minimum activity 0.25; and MULT16x10_C (right) with 95% accuracy that error is less than 5%, minimum activity threshold 0.15.

Table 1. Test cases.

	#Slices	#Slice FF	Min. Period (ns)	#nodes
FIRDA_1	159 (3%)	307 (3%)	5.781	1486
FIRDA_2	303 (6%)	597 (6%)	7.305	2774
FIRDA_4	595 (12%)	1177 (12%)	6.484	5245
FIRDA_8	1163 (24%)	2305 (24%)	5.903	9495
MULT32_C	640 (83%)	193 (12%)	40.377	6627
ADDER32_C	49 (6%)	97 (6%)	11.354	2425
MULT16_P	172 (22%)	341 (22%)	9.638	2194
DIV16_P	425 (55%)	831 (54%)	9.899	3257
MULT16x10_C	1654 (11%)	586 (2%)	14.928	27471

where e_i and $e_{eff,i}$ are the specified and the corresponding effective error respectively for the i-th of p estimation runs (p=18 in Fig. 1).

At the same time these better effective accuracies are obtained, it is observed that nodes do not converge linearly. For example, for the FIRDA_8 and MULT16x10_C test circuits (see Table 1), Fig. 2 shows that 98% of the nodes, representing 99% of the power, have met the stopping criteria halfway through the estimation process.

According to these observations, we propose the new power estimation technique called A-B (The A-B name comes from the ABC technique applied in stock control -and other areas in Operations Research- where the articles are classified in three groups, A, B, and C based on the total annual expenditure for each item). Being ε the tolerated error -and $(1-\varepsilon) \times 100\%$ the confidence level-, we can consider the estimation process finished when more than $1-\varepsilon \times St$ % of the normal nodes have converged. The new user-specified parameter St is called optimization strength, and adjusts the estimation process run time and effective accuracy values. For example, with 10% error, 90% of confidence, and 1000 normal nodes, if the optimization strength is 1.0, then the estimation is considered complete when more than 900 nodes have met the stopping criterion defined in (1). If the parameter is set to 0.5, then the estimation is considered finished when more than 950 nodes have converged. In short, the additional condition -besides the stopping criteria at the node level defined in [5]- to finish the estimation is

$$\frac{N_{no}}{N_{reg}} \leq e \times St, \tag{3}$$

where N_{reg} is the regular nodes count, N_{no}, is the number of regular nodes that have not converged yet, ε is the user specified error, and St is the specified optimization strength.

3. EXPERIMENTAL RESULTS

The proposed method is implemented by a Tcl/Tk script that calls several programs and other scripts in order to obtain the average individual node activities and capacitances [6, 9]. The tool is integrated in the Xilinx design flow, but it uses standard formats as far as possible. A simulator (Modelsim) running post PAR VHDL models with routed delays is used in the inner loop of the statistical technique. Several experiments are performed on the circuits listed in Table 1 where the fifth column has the number of nodes counted from the post place and route simulation model which is described in terms of primitive library components.

FIRDA circuits are different implementations of a FIR filter applying distributed arithmetic and the relative placement technique. The filters use 64 6-bit coefficients, 8-bit input and output words, 12.5 MHz fixed sampling frequencies, and a 2/3 cut-off frequency. The difference among these implementations is the internal digit size from bit serial to completely combinational. As the sampling frequency is fixed, the clock must be adjusted to compute each sample before the next is available [10]. These circuits are implemented over a Virtex-E XCV400E-8PQ240 device.

Next, four arithmetical circuits are implemented over a Virtex XCV50PQ240-4 device: A combinational 32-bit multiplier, a combinational 32-bit adder, a pipelined 16-bit multiplier and a pipelined 16-bit divider. All these circuits operate with unsigned integers. The 32-bit adder and multiplier were specified using a simple behavioral VHDL description. For the pipelined multiplier and divider, the corresponding cores are generated with the Xilinx Core

Table 2. Comparison results.

	Base sample size	A-B sample size	Time savings
FIRDA_1	1928	1192	38%
FIRDA_2	2476	652	74%
FIRDA_4	2520	856	66%
FIRDA_8	2467	917	63%
MULT32_C	2576	820	68%
ADDER32_C	578	497	14%
MULT16_P	1602	1039	35%
DIV16_P	1091	528	52%
MULT16x10_C	1681	701	58%

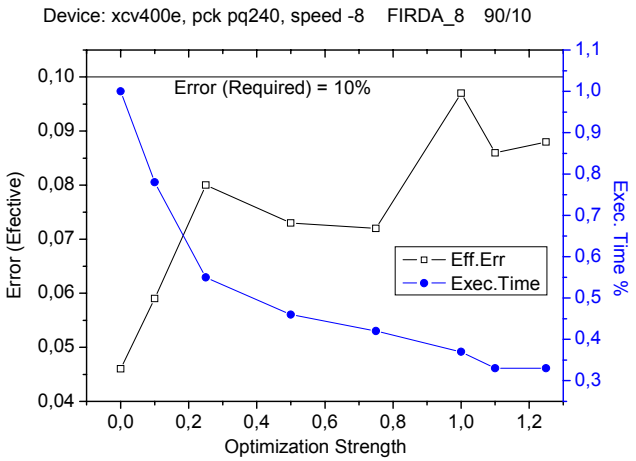

Fig. 3. Execution time and effective accuracy in function of the optimization strength for the FIRDA_8 test circuit.

Generator [11]. MULT16x10_C is implemented over a Virtex-II XC2V3000FG676-6 device. It consists of ten 16-bit combinational multipliers using the general configurable logic.

Table 2 shows the comparison results between the techniques in [5], called Base in the table, and the A-B proposed here. In all the cases, the A-B method requires much smaller samples, with an average saving of 52% for the following specification: 10% error, 90% confidence, and 0.25 min. activity threshold. A-B runs with optimization strength 1.00. Time savings are computed as 1 - column 3/column 2 in Table 2.

The results in Table 2 correspond to specific points in the parameters space. Consequently, it is necessary to make a deeper study of at least one test circuit through a wider range of values. Firstly, the effective error with the A-B technique is revised for different optimization strength values within the 0-1.3 range. Fig. 3 shows for the FIRDA_8 circuit how, as the optimization strength is higher; the effective error approaches the specified one (10%) so that every simulated clock cycle in the taken sample becomes useful and efficient. As it is claimed, there is no loss of accuracy. Furthermore, a dramatic saving in execution time is observed. The savings are expressed in relative terms where 1.0 corresponds to the case without optimization strength. For example, when *St*=1.0, the sample size is less than 40% of the one without optimization.

Another illustration of how the effective error tends to the specified value is Fig. 4. It shows relative error

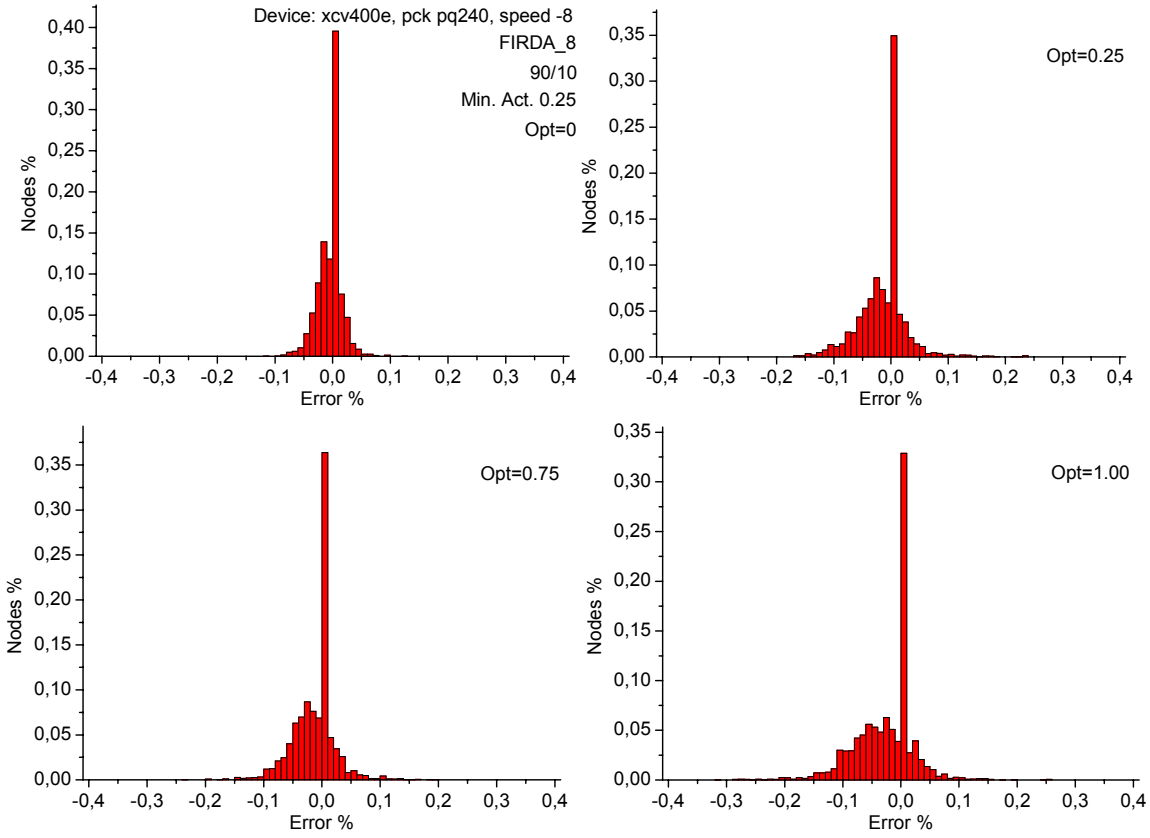

Fig. 4. Individual nodes power: relative error distributions for the FIRDA_8 test circuit.

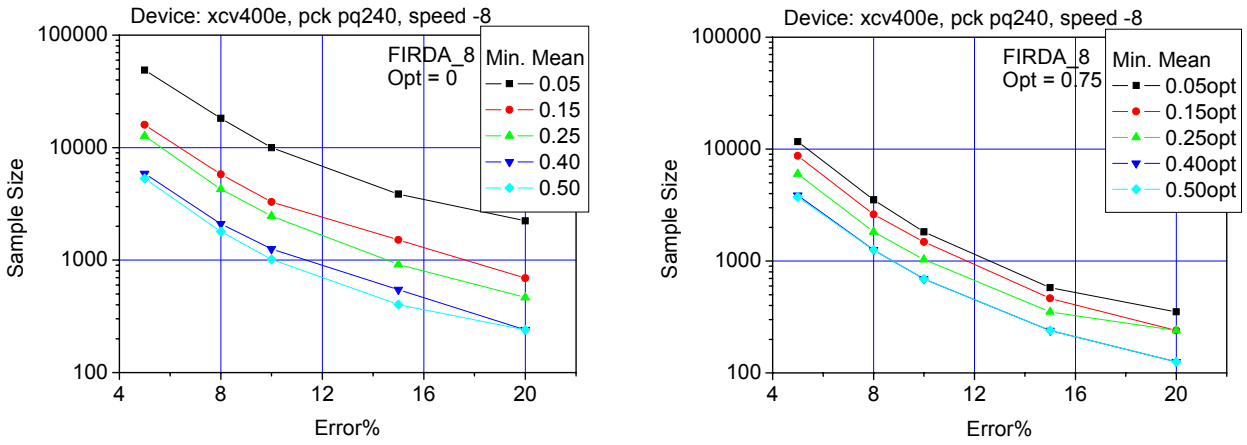

Fig. 5. Characterization of the accuracy vs. execution-time tradeoff for FIRDA_8.

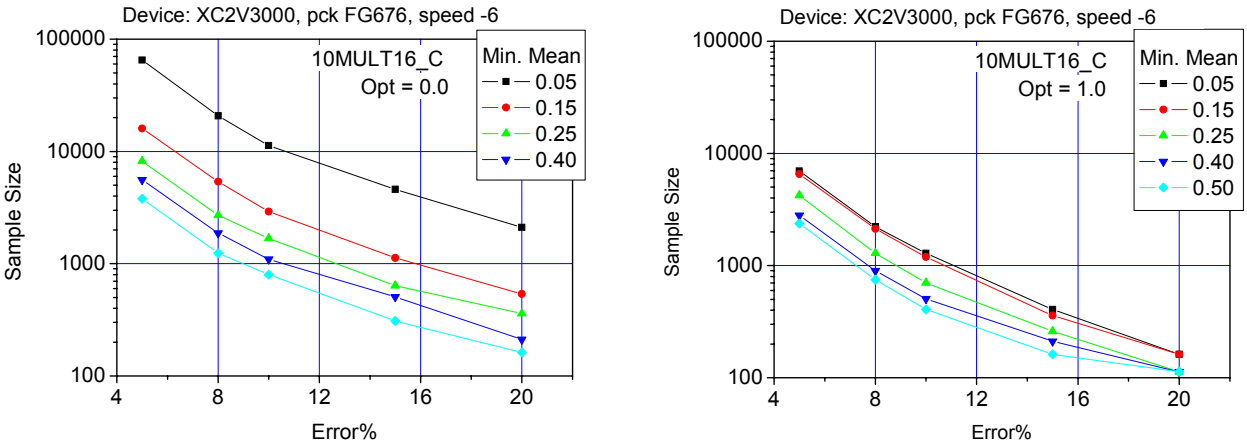

Fig. 6. Characterization of the accuracy vs. execution-time tradeoff for MULT16x10_C.

distributions for different user specified optimization strengths. In these runs, it is specified with 90% confidence that the error is less than 10%; meanwhile the minimum activity threshold is 0.25. Although the effective accuracy is a random variable, it clearly approaches to the one specified by the user as the optimization strength increases. In Fig. 4, when the optimization strength is zero, there are no nodes with error higher than 10% but the user did not specified 99.99% confidence. In this way, the obtained accuracy is higher than the specified one in the classical approach. Note that there are a lot of nodes with zero error: This is because Xilinx reports a zero capacitance for them.

Fig. 5 and 6 represent the accuracy vs. execution time tradeoff for FIRDA_8 and MULT16x10_C circuits respectively, where the behavior of the estimation system is characterized. The *x*-axis represents the accuracy, where an

Table 4. Execution time savings for FIRDA_8. Optimization strength is 0.75.

Error	Min. Activity Threshold				
	0,05	0,15	0,25	0,4	0,5
5	0,76	0,45	0,53	0,35	0,30
8	0,81	0,55	0,57	0,40	0,30
10	0,82	0,55	0,58	0,45	0,32
15	0,85	0,69	0,62	0,56	0,40
20	0,84	0,65	0,49	0,47	0,47

Table 3. Execution time savings for MULT16x10_C. Optimization strength is 1.00.

Error	Min. Activity Threshold				
	0,05	0,15	0,25	0,4	0,5
5	0,89	0,59	0,48	0,50	0,38
8	0,89	0,61	0,52	0,52	0,40
10	0,89	0,59	0,58	0,54	0,49
15	0,91	0,68	0,59	0,58	0,48
20	0,92	0,70	0,68	0,46	0,30

x_i value corresponds to an x_i% error with 100-x_i% confidence level. This experiment confirms the robustness of the statistical technique, allowing a tunable accuracy. In order to give more information about the results in Fig. 5 and 6, Table 3 y 4 respectively, shows the execution time savings with respect to the Base case. It is observed that the best savings are obtained in the most favorable cases, where the required accuracy and execution times are high.

4. CONCLUSION

In this paper, an improvement for the classical Monte Carlo power estimation method for individual nodes has being presented. Although the method is implemented and evaluated within the particular Xilinx ISE design flow, standard formats are used as far as possible. Furthermore, there are no restrictions to apply the technique within other FPGA design environments or even general CMOS design flows.

The problem with the classical statistical estimation method is the execution time. Current big designs could require unacceptable run times when the user specifies medium or high accuracy requirements. The proposed A-B technique takes up reasonable run times enabling its practical use within existing design flows. Moreover, the proposed technique is simple and easy to implement.

It has been shown that the optimization is done without loss of accuracy at the individual nodes level. This is because the A-B method makes use of the extra accuracy generated running the classical approach that is effectively higher than that specified by the user. To quantify and measure this extra precision, a definition of effective accuracy has been proposed.

5. REFERENCES

[1] F. Najm, "Estimating Power Dissipation in VLSI Circuits," *IEEE Circuits and Devices Magazine*, vol. 10, no. 4, pp. 11-19, 1994.

[2] M. Pedram, "Design technologies for Low Power VLSI," in *Encyclopedia of Computer Science and Technology*, vol. 36, Marcel Dekker, Inc., pp.73-96, 1997.

[3] K.K.W. Poon, S.J.E. Wilton, and A. Yan, "A Detailed Power Model for Field-Programmable Gate Arrays," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 10, issue 2, pp. 279-302, April 2005.

[4] J. A. Anderson, and F. N. Najm, "Power Estimation Techniques for FPGAs," *IEEE Trans. on VLSI Systems*, vol. 12, no. 10, pp. 1015-1027, 2004.

[5] F. N. Najm, and M. G. Xakellis, "Statistical estimation of the switching activity in VLSI circuits," *VLSI Design*, vol. 7, no. 3, pp. 243-254, 1998.

[6] E. Todorovich, M. Gilabert, G. Sutter, S. Lopez-Buedo, and E. Boemo, "A Tool for Activity Estimation in FPGAs," in: M. Glesner, P. Zipf, and M. Renovell (Eds.): FPL 2002, *Lecture Notes in Computer Science*, vol. 2438, Springer-Verlag, Berlin Heidelberg, pp. 340-349, 2002.

[7] C-S. Ding, C-T. Hsieh and M. Pedram, "Improving efficiency of the Monte Carlo power estimation," *IEEE Trans. on VLSI Systems*, vol. 8, no. 5, pp. 584-593, 2000.

[8] B. Kwak, and E.S. Park, "An Optimization-Based Error Calculation for Statistical Power Estima-tion of CMOS Logic Circuits," in *Procs. of the Design Automation Conference*, San Francisco, California, USA, pp. 690-693, 1998.

[9] E. Todorovich, Boemo, E.; Angarita, and F.; J. Valls, "Statistical Power Estimation for FPGAs," in *Procs. IEEE 15th Intern. Conf. on Field Programmable Logic and Applications*, pp. 515-518, 2005.

[10] F.E. Angarita, M.J. Canet, J. Valls, and F. Viñedo, "Implementación de un Core IP: Filtro FIR basado en Aritmética Distribuida," in *III Jornadas sobre Computación Reconfigurable y Aplicaciones*, pp. 139-145, 2003.

[11] Xilinx Inc., "CORE Generator Guide", an *Xilinx ISE 7 Software Manual*, available at http://www.xilinx.com, 2004.