A Study on Hybrid SA-GA and SA-EA Methods for Finding the Maximum Number of Switching Gates in a Combinational Circuit

Ichiro R. Ruiz Obregon[1) · Alberto Palacios Pawlovsky[2)

(2009 年 3 月 7 日　受理)

Abstract — Low power and reliable thermal design are very important in developing state-of-the-art circuits. This work shows the results obtained with a hybrid of simulating annealing (SA) and a genetic algorithm (GA) and with a hybrid of SA and an evolutionary algorithm (EA) in the generation of inputs pairs that cause the maximum number of switching gates in combinational circuits. We found that both hybrids, SA-GA and SA-EA, produce better results than those obtained using only SA, GA or EA alone.

Keywords — combinational circuits, switching gates, simulated annealing, genetic algorithm, evolutionary algorithm.

I. Introduction

Modern VLSI designs use integration technologies that make them very complex and with billions of transistors. This complexity now allows us to put a complete system on a chip and has fostered the development of very complex portable devices that to handle their multiple functions require high performance designs. High performance usually implies the use of high frequency clocks and brings the problems of increasing heat and power consumption. This also comes together with the fact that each generation of mobile devices reduces the space allowed to batteries. Moreover, battery technology has not been able to keep pace with this downsizing trend making power reduction a pervasive problem in any new development. We can estimate the power consumption of a design at different levels. At the system level, high-level descriptions of the circuit and abstractions of capacitance and switching are used to estimate power [1]. At the functional level, abstraction of functional blocks (muxes, ALUs, registers) provide the models to get power estimates [2]. At the gate level, we can use BDD (Binary Decision Diagrams) to get some estimates to address power related problems [3]. At the transistor level, we can use model correlations to predict worst case power consumption [4]. We can

Ichiro R. Ruiz Obregon[1) : Graduate School of Biomedical Engineering Toin University of Yokohama
Alberto Palacios Pawlovsky[2) : Department of Robotics and Information Engineering Faculty of Engineering, Toin University of Yokohama, 225-8502, Yokohama, Aoba-ku, Kuroganecho 1614, Japan

break these techniques in non-simulative, simulative, and hybrid techniques. At the gate level we can divide these techniques in probabilistic and sampling-based (deterministic) techniques [5], [6], [7]. This paper study two heuristics of the last type. There are several methods to find an input pair that causes the maximum number of switching gates in a circuit. We have the iterative method of [8], the simulated annealing (SA) method of [9], the genetic algorithm (GA) of [10] and the evolutionary algorithm method (EA) of [11]. Recently we have published some results of a hybrid of SA and GA and of a hybrid of SA and EA [12], [13]. We extended our work on those methods to test them with fixed values of some of their parameters. Those results are shown in the following sections. Section II describes briefly the SA-GA and SA-EA methods. In section III, we show the experimental results we obtained fixing some parameters of these methods. In the last section, we give some conclusions and give some topics for future work.

II Hybrid SA-GA and SA-EA Methods

In this section we will describe the SA-GA and SA-EA methods we developed and used in this work.

A. Hybrid SA-GA Method

Fig.1 shows the flow diagram of pair searching common to both of our hybrids. Its core is the flow diagram of a SA method and in it we also search and evaluate pairs one by one. When a new pair (p_i) is generated it is compared to a current one (p_{i-1}) . If the difference in the number of switching gates $\Delta G(G_i - G_{i-1})$ is greater

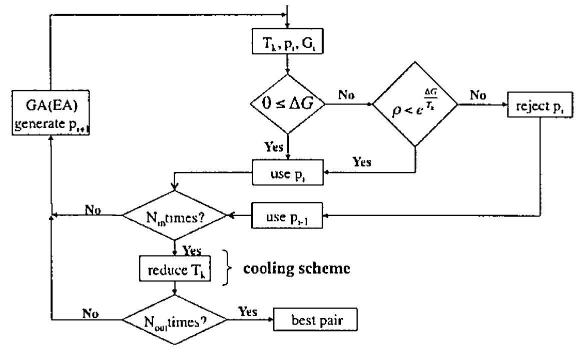

Fig.1. Pair searching flow diagram of the SA-GA and SA-EA methods.

than zero (i.e., there is an increase in the number of switching gates) we use the new pair to generate the next one (p_{i+1}). This iterative process is carried out N_{in} times, dictated by the inner loop of SA. If ΔG shows a decrease in the number of gates, we do not discard immediately the new pair. Instead, we use a decision algorithm to determine stochastically if we use the "bad" pair or not. If we reject it, then the previous one (p_{i-1}) will be used to generate a new pair (p_{i+1}). The decision of rejection or acceptance of a "bad" pair is done using equation (1).

$$p < e^{\frac{\Delta G}{T_k}} \quad (1)$$

Here, T_k is the parameter known as temperature in SA. This parameter is usually set to a high value, and is decreased every time the algorithm leaves its inner loop. The decreasing of parameter T_k is done following a scheme known as the cooling scheme of SA. In this paper, we use the simple geometric cooling scheme given by equation (2).

$$T_{k+1} = \alpha \times T_k \quad (2)$$

Where α was set to 0.7. The number of times the temperature is reduced in the

SA-GA and SA-EA methods is controlled by the outer loop of SA (N_{out}). The number of solutions (pairs) searched by both methods is given by N_{in} × N_{out}, for a total of 10,000 pairs. In both methods, once the number of cycles of the outer loop is completed, the method finishes giving as answer the best pair found in the search. The main difference between both hybrids is the method of generating new pairs (the block on the left in Fig.1).

1) Input Pair Representation in SA-GA and SA-EA: In SA-GA and SA-EA pairs are represented as chromosomes composed of genes following the representation used in [9]. An example of it is shown in Fig.2. The 4-value vector enclosed in dot lines is the chromosome and its values the genes of it. Switching gates are shown in grey in Fig.2 (b).

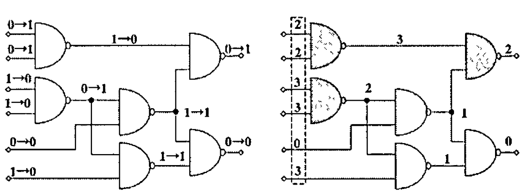

Fig.2. Signal-change (pair) representation (a) binary (b) 4-value.

2) Generation of Input Pairs in SA-GA: In a SA algorithm a new solution is usually generated from a current one and the algorithm itself searches one pair at a time. While in a GA solutions are evaluated in groups called generations. In the SA-GA method we adapted the crossover and mutation operations used in GA to work on only one pair. The details are given in what follows.

3) Crossover and Mutation for SA-GA: In a GA new solutions (pairs in our case) are generated applying crossover and mutation to a current generation (group of solutions). After crossover, mutation is applied to each gene of the new solutions. For crossover in a GA we need two solutions as parents to crossover (to exchange parts of themselves). Since in the SA-GA we also evaluate one solution at a time we had to adapt crossover to work on one solution (pair) at each time. Our approach is shown in Fig. 3. It shows one-point crossover when applied to one pair. Crossover is applied using a probability p_c. Usually before applying crossover a random number is generated and compared to p_c to decide if crossover is applied or not. If

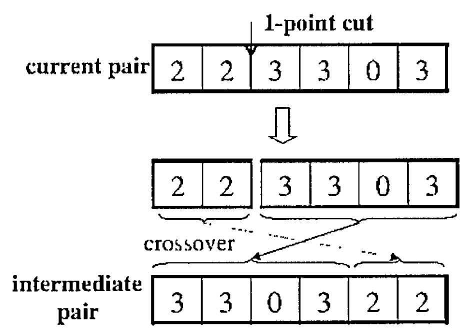

Fig.3. Crossover adapted to work on one pair.

the random number is smaller than the value of p_c, then crossover is applied to the corresponding pair. If the random number exceeds the value of p_c, then crossover is not applied and the pair is left unchanged. After this process mutation is applied to the pair. Mutation is an operation that is applied to each gene (value) of the pair product of crossover. This operation is also applied using a probability p_m. Before applying it we also generate a random number and see if it is smaller than p_m. If it is, then we change the corresponding value in the pair. If the random number is larger than the value of p_m, then the corresponding gene is left unchanged and we move to test the next gene (value) in the pair. Fig.4 shows mutation when applied to the pair of Fig.3. In this example three genes has been changed creating a new pair.

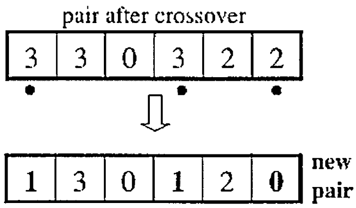

Fig.4. Example of mutation applied to an input pair.

B. Hybrid SA-EA Method

The SA-EA hybrid method shares with SA-GA the same flow diagram (see Fig.1). New pairs in SA-EA are generated using an evolutionary algorithm (EA). Our EA generates new input pairs (chromosomes) taking genes (values) from the best and current pairs during the search. Its workings are detailed in the following subsection.

1) Input Pair Generation Using EA: In SA-EA we generate a new pair from the best pair found so far in the search and from the current pair as is shown in Fig.5. The best pair and current pair are shown as ***b*** and ***c***.

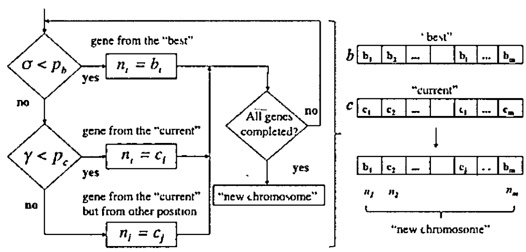

Fig.5. Flow diagram of the EA used in pair generation.

The b_i and c_j represent the genes $(i,j > 0 \forall$ $i,j \in \{1, 2, \ldots, m\}$, where m is the length of the chromosome). A new pair will be formed by genes taken from these two pairs. The gene of the new pair will take the gene of the best pair with a probability p_b. To determine this, a random number is generated and compared to p_b. If it is smaller or equal to p_b, then the value (gene) of the best pair will be copied to the corresponding place in the new pair. Otherwise, we will generate another random number and compare it to another probability p_c to see if we copy the value from the current pair. If this new random number is between 0 and p_c, then the corresponding value of the current pair will be copied to the new pair. If the random number is greater than p_c, then we will choose randomly another value (gene) from the current pair and copy it to the new pair. In [12] we showed results of the SA-GA method with a p_c value of 0.9 and a p_m value that depends on the number of the inputs of the circuit under analysis. Also, we have shown in [13] the results of the SA-EA method with p_b and p_c values that change during the search process. In this paper we controlled those values of SA-GA and SA-EA and determined the best range of values of them. Details of those experiments are given in the following section.

Ⅲ. Experimental Results

We run a set of experiments with the hybrid methods explained above to determine the best set of parameters for each of them. For the SA-GA hybrid method we set the values of parameters p_m and p_c from 0.1 to 0.9 in increments of 0.1 for a total of 9 different values and

81 combinations of them. The same was done for the SA-EA hybrid method for its parameters p_b and p_c. As already indicated for both methods we used a geometric cooling scheme with α set to 0.7 and N_{in} and N_{out} both set to 100. To compare their results to those of [12] and [13] we set T_0 in the SA-GA method to 1,000 and in the SA-EA method to 10,000. With these settings we run the same simulations ten times to see the characteristics of each setting and the quality of their results. We run the simulations on all the benchmark combinational circuits of ISCAS85 [14]. Details of the results are given for each method in the following subsections.

A. SA-GA Method

We found that among the 81 combinations of settings with p_c and p_m the best results are obtained with p_m = 0.9 and $0.1 \leq p_c \leq 0.4$. The best (maximum) number of switching gates obtained with these settings are shown in Table I. This table shows also the comparison of the number of switching gates generated by the hybrid SA-GA taking as reference the SA or GA methods. As we can see in most of the cases a fixed setting increased the number of switching gates in a range between 2% and 19% respect to SA and in a range between 1% and 48% respect to GA. However, the SA-GA method with fixed parameters, when compared to the SA of [9], did not increased those numbers for the c2670 and c7552 circuits (both are adder/compare circuits).

B. SA-EA Method

The results for the SA-EA hybrid method are shown in Table II. The best results were found for $0.3 \leq p_b \leq 0.4$. and for $0.1 \leq p_c \leq 0.9$. With the settings in these ranges, we obtained an increment in the number of switching gates in a range from 1% to 19% when compared them to those of the SA of [9], and between 2% and 22%, when compared to the results of the EA method of [11].

TABLE I
COMPARISON WITH SA AND GA METHODS.

ISCAS85		Max. Switching Gates (*SA*[9]=1) (*GA*[10]=1)			
circuit	gates	[9]	[10]	SA-GA[12]	SA-GA
c432	161	102 (1) (0.99)	103 (1.01) (1)	119 (1.17) (1.16)	121 (1.19) (1.17)
c499	202	157 (1) (0.98)	160 (1.02) (1)	161 (1.03) (1.01)	162 (1.03) (1.01)
c880	383	280 (1) (1.19)	235 (0.84) (1)	304 (1.09) (1.29)	307 (1.10) (1.31)
c1355	546	327 (1) (1.00)	328 (1.00) (1)	326 (1.00) (0.99)	337 (1.03) (1.03)
c1908	880	710 (1) (1.35)	526 (0.74) (1)	748 (1.05) (1.42)	748 (1.05) (1.42)
c2670	1193	1130 (1) (1.22)	927 (0.82) (1)	1161 (1.03) (1.25)	1123 (0.99) (1.21)
c3540	1669	953 (1) (1.14)	834 (0.88) (1)	990 (1.04) (1.19)	990 (1.04) (1.19)
c5315	2307	1912 (1) (1.45)	1321 (0.69) (1)	2057 (1.08) (1.56)	1955 (1.02) (1.48)
c6288	2416	1630 (1) (1.23)	1324 (0.81) (1)	1697 (1.04) (1.28)	1697 (1.04) (1.28)
c7552	3512	2933 (1) (1.31)	2235 (0.76) (1)	2916 (0.99) (1.30)	2871 (0.98) (1.28)

Ⅳ. Conclusions

We have shown in this paper a study of the influence of certain parameters on the results of two new hybrids. We just studied the influence of parameters p_c and p_m for the SA-GA method, and p_b and p_c for the SA-EA

TABLE II
COMPARISON WITH SA AND EA METHODS.

ISCAS85		Max. Switching Gates (SA[9]=1) (EA[11]=1)			
circuit	gates	[9]	[11]	SA-EA[13]	SA-EA
c432	161	102 (1) (0.89)	115 (1.13) (1)	120 (1.18) (1.04)	121 (1.19) (1.05)
c499	202	157 (1) (0.97)	162 (1.03) (1)	162 (1.03) (1.00)	166 (1.06) (1.02)
c880	383	280 (1) (1.13)	272 (0.97) (1)	304 (1.09) (1.12)	307 (1.10) (1.13)
c1355	546	327 (1) (1.00)	328 (1.00) (1)	344 (1.05) (1.05)	349 (1.07) (1.06)
c1908	880	710 (1) (0.99)	716 (1.01) (1)	748 (1.05) (1.04)	748 (1.05) (1.04)
c2670	1193	1130 (1) (1.04)	1087 (0.96) (1)	1164 (1.03) (1.07)	1164 (1.03) (1.07)
c3540	1669	953 (1) (1.02)	934 (0.98) (1)	990 (1.04) (1.06)	990 (1.04) (1.06)
c5315	2307	1912 (1) (1.13)	1691 (0.88) (1)	2065 (1.08) (1.22)	2066 (1.08) (1.22)
c6288	2416	1630 (1) (1.02)	1597 (0.98) (1)	1697 (1.04) (1.06)	1697 (1.04) (1.06)
c7552	3512	2933 (1) (1.03)	2835 (0.97) (1)	2961 (1.01) (1.04)	2961 (1.01) (1.04)

method. The results shown that there exists a range of values for them under which we can expect improvements in the number of switching gates. However, it seems that we can not expect to increase the quality of the results for some circuits. An example of this case appeared for the SA-GA method with circuits c2670 and c7552. These are circuits with similar functions. The c2670 is an ALU with comparator capabilities, the c7552 is an adder with a magnitude comparator [15]. The study of other parameters as the N_{in}, N_{out} and the initial temperature T_0 in both hybrids is left as a topic for further research. It also could be interesting to see the influence of other more elaborated cooling schemes as the one used in [9].

REFERENCES

[1] P. Agrawal, Srinivasa R. STG, A. N. Oke, and S. Vijay, "A Scalable Modeling Technique to Estimate Dynamic Thermal Design Power of Datapath Intensive Designs," Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'07), pp.389-394, March 2007.

[2] L. Zhong, S. Ravi, A. Raghunathan, and N. K. Jha, "Power Estimation for Cycle-accurate Functional Descriptions of Hardware," Proceedings of the 2004 International Conference on Computer Aided Design (ICCAD'04), pp. 668-675. November 2004.

[3] R. L. Wright and M. A. Shanblatt, "Improved Power Estimation For Behavioral and Gate Level Designs," Proceedings of the IEEE Computer Society Workshop on VLSI 2001 (WVLSI'01), pp. 0102, April 2001.

[4] Savithri S., Venkatesan R., and Bhaskar, "An Assertion Based Technique for Transistor Level Dynamic Power Estimation," Proceedings of the 13th. International Conference on VLSI Design, pp. 34, January 2000.

[5] T. Kim, K.S. Chung, C. L. Liu, "A Static Estimation Technique of Power Sensitivity in Logic Circuits," Proceedings of the 38th Design Automation Conference (DAC'01), pp.215-219, June 2001.

[6] C.Y. Wang and K. Roy, "Maximum Power Estimation for CMOS Circuits Using Deterministic and Statistical Approaches," Proceedings of the 9th. International Conference on VLSI Design, pp. 364-369, January 1995.

[7] C.Y. Wang and K. Roy, "COSMOS: A Continuous Optimization Approach for Maximum Power Estimation of CMOS

Circuits," Proceedings of the 1997 International Conference on Computer Aided Design (ICCAD'97), pp. 52-55. November 1997.

[8] K. Zhang, H. Takase, T. Hayashi, and H. Kita, "An Enhanced Iterative Improvement Method for Evaluating the Maximum Number of Simultaneous Switching Gates for Combinational Circuits," Proceedings of the 1997 Asia and South Pacific Design Automation Conference (ASP-DAC'97), Chiba, Japan, pp.107-112, January, 1997.

[9] Alberto Palacios Pawlovsky, "Using simulated annealing to generate input pairs to measure the maximum power dissipation in combinational CMOS circuits," IEICE ELEX, Vol.2, No.4, pp. 115-120, February 2005.

[10] Alberto Palacios Pawlovsky, "On the use of Genetic Algorithms in Generating Input Pairs that Cause the Maximum Power Consumption in CMOS Combinational Circuits," Proceedings of the XIII Iberchip Workshop (IWS-2007), Lima, Peru, pp.183-186, March, 2007.

[11] Alberto Palacios Pawlovsky, "An evolutionary Genetic Algorithm for the Generation of Input Pairs that Cause the Maximum Number of Switching Gates in CMOS Combinational Circuits," Proceedings of the 22nd. International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2007), vol.3, pp. 1047-1048, July 2007.

[12] Ichiro R. Ruiz Obregon and Alberto Palacios Pawlovsky, "A Hybrid SA-GA Method for Finding the Maximum Number of Switching Gates in a Combinational Circuit", The 23rd International Technical Conference on Circuits/Systems, Computers and Communication (ITC-CSCC 2008), pp. 1085-1088, July 2008.

[13] Ichiro R. Ruiz Obregon and Alberto Palacios Pawlovsky, "A hybrid SA-EA method for finding the maximum number of switching gates in a combinational circuit", IEICE Electronics Express, Vol. 5, No. 18, pp. 756-761, Tokyo, September 2008.

[14] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits," Proc. IEEE International Symposium on Circuits and Systems, pp.695-698, 1985.

[15] M. C. Hansen, H. Yalcin and J. P. Hayes, "Unveiling the ISCAS8-85 Benchmarks," IEEE Design & Test of Computers , pp.72-80, 1999.