A SoC Test Scheduling Approach based on Simulated Annealing Algorithm with Error Feedback

Huaien Gao[1,2, a], Hui Chen[2] and Xiaoming Xiong[2]

[1]Guangzhou National Integrated Circuit Design Industrialization Center for Modern Service Industry, Guangzhou 510000, China;

[2]School of Automation, Guangdong University of Technology, Guangzhou 510000, China.

[a]gaohuaien@gzicc.cn

Abstract. Because of the uncertainty of optimal solution, SoC test scheduling has attracted major concern in recent year. In this paper, a scheduling approach based on simulated annealing (SA) algorithm with error feedback is proposed. With this model, the error information generated in each iteration of SA is utilized to adjust the new potential solution for the next iteration; therefore, it enhances the learning of the algorithm. The result shows better performance comparing with previous work of authors and also with the traditional methods such as integer linear programming algorithm (ILP), simulated annealing algorithm (SA) and genetic algorithm (GA).

Keywords: SoC test; scheduling; simulated annealing; error feedback.

1. Introduction

Techniques for integrated circuit (IC) have been developed for decades, and computational intelligence become more and more important in IC design. A typical IC chip, particularly a system-on-chip (SoC), contains many Intellectual Property (IP) cores in order to have faster design cycle and robustness of the product. Therefore, the IP cores will also be tested in the procedure of testing of the SoC. In order to reduce the time and so of the cost of testing of SoC, the schedule of testing should be carefully designed. One particular problem lies in the limit width of the test access mechanism (TAM) for the SoC test, which makes the scheduling difficult and need comprehensive algorithm to solve. Many researches have been devoted in this non-deterministic polynomial complete problem. The IEEE Computer Society's Test Technology Committee began to study the testing of embedded cores in 1997, and drafted the IEEE P1500 standard, which unifies the test interfaces between IP core providers and IP core users. The standard was developed into a formal standard in 2005. V. Iyengar et al. [3] transformed the ILP (Integer Linear Programming) problem to test the optimal scheduling to solve the SoC optimization test problem; Yang Jun et al. [4] optimized the test scheduling based on the transfer closed graph representation and combined with simulated annealing algorithm; Rega et al. [1] solved the test scheduling with the adaptability of genetic algorithm (GA). In this problem, the feasible solution set of the test scheduling problem is represented by population, and the approximate solution is generated by successive evolution according to the principle of survival of the fittest. Cui Xiaole et al. [5] proposed a SoC test scheduling method based on ant colony algorithm with the constraint of peak temperature, which can obviously optimize the test time under the condition of thermal safety.

In this paper, an improved simulated annealing algorithm with error feedback is used to optimize the structure of IP core allocated to TAM, in order to reduce the test time. The results show that the test time of this algorithm is shorter than that of traditional integer linear programming (ILP) [3], simulated annealing algorithm (SA) [4] and genetic algorithm (GA) [1], and is better than the previous work of authors [6].

2. Test Scheduling Approach with Simulated Annealing

2.1 Problem Definitions

In the previous work [6] of authors, the problem of SoC scheduling was rewritten as follows:

(1) P_W Problem: A test shell is designed for a given IP core and TAM bit width to minimize the test time and TAM width of the core.

(2) P_AW Problem: Given B test buses, the total width of the test bus is W, and N_C IP cores are allocated to each TAM structure. The width of each test bus is determined to minimize the total test time of the chip.

(3) P_NAW Problem: Given B TAMs and N_C IP cores, the total width of the test bus is W. It is necessary to determine the structure of each IP core assigned to each TAM and the width of each TAM, so as to minimize the total test time of the chip.

(4) P_PNAW Problem: Given the total width of N_C IP cores and test buses is W; the number of TAMs B needs to be determined. Each IP core is allocated to the structure of each TAM and the width of each TAM to minimize the total test time of the chip.

Therefore, to solve the scheduling of SoC, one needs to solve these four problems.

2.2 BFS Algorithm to Solve P_W

Problem (1) of P_W is the same as in the definition from V. Iyengar, which can be solved by using the breadth-first-search (BFS) heuristic algorithm. The following is the Pseudo code for BFS, where vector v here represents the state the answer space. The subroutine of the algorithm is named *design_wrapper ()*.

[1]. push vector v into queue q
[2]. while q is not empty
   a) fetch the front vector v from q
   b) delete the front vector of q
   c) follow the vector v and put all of its sub-nodes into the set s
   d) push all elements of s into queue q
[3]. iterate the step [2]

2.3 Greedy Algorithm to Solve P_AW

The basic scheme of solving this problem is to run the subroutine *design_wrapper()* and find out the bottleneck TAM, which takes the most of time for testing, then increase the width of this particular bottleneck TAM and retry the subroutine *design_wrapper()* again. Such iteration will not be stopped till the test bus width W has been allocated. This procedure is assigned to subroutine *allocate_TAM_Width ()*, with its Pseudo code as followed. Note that the value lambda is an integer and its value can be fixed or be variable.

[1]. Initialization with TAM_i = 1 for all of i
[2]. While sum (TAM_i) < W
   a) for each of $\omega_i = 1 (i = 1,2, \dots, B$,calculate the test time $T(i)(i = 1,2, \dots, B)$for each TAM using subroutine *design_wrapper()*
   b) find the max value among $T(i)(i = 1,2, \dots, B)$as t_max
   c) Mark the TAM with t_max testing time as the Bottleneck_TAM
   d) Increase the TAM value of Bottleneck_TAM by lambda
[3]. The best result remains after the while loop

2.4 Using Simulated Annealing Algorithm to Solve P_NAW and P_PNAW

The remain problems of solving P_NAW and P_PNAW are addressed together by the simulated annealing (SA) algorithm. SA is derived from the principle of solid annealing, and its flow chart is shown as follow. Please note the feedback of E_delta, which does not present in the original SA algorithm.

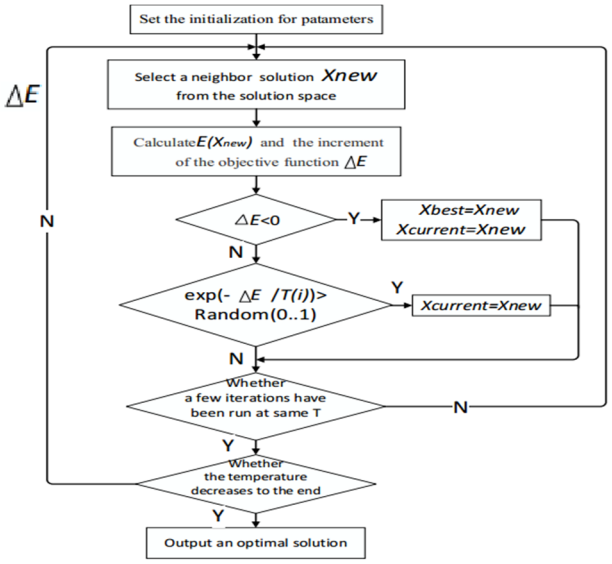

Fig 1. Flow chart of simulated annealing algorithm with error feedback

Three elements are important to understand the simulated annealing algorithm, namely the solution space, the objective function and the initial solution. In this paper, the structure describing the allocation of IP cores to TAM is the solution space, and the objective function refers to the test time of SoC and the initial solution is arbitrary structure of allocation. The Pseudo code of the algorithm can be found as following:

[1]. initialization: TAM_Num_min = 1, TAM_Num_max = min {N_C, W}; let TAM_Num = TAM_Num_min

[2]. While TAM_Num < TAM_Num_max

a) initial a new solution X_new by randomly assign IP core to each TAM
b) allocate the width for the internal TAM and calculate the initial test time E_new by calling the subroutine *allocate_TAM_Width()* with lambda as the input value;
c) let E_best=E_new, E_current=E_new;
d) set the initial temperature T=T_0, let the Markov variable i=1;
e) randomly choose a IP core and move it to another randomly selected TAM, and generate a solution X_new.
f) re-allocate the width for the internal TAMs and update E_new by running the subroutine *allocate_TAM_Width()*
g) update the IP core allocation solution and the E_current value as E_new if E_new < E_current or e ^(E/T) > *Rand ()*;
h) Calculate E_delta as the difference between E_new\E_best and update the allocate solution and the E_best as E_new if E_new<E_best
i) Go to step (d) if i<Markov
j) reduce the temperature by the coefficient alpha (<1), that is, T=T* alpha
k) if T>T_f, go to step (c).
l) let TAM_Num=TAM_Num+ lambda, where lambda= *ceiling(*E_delta/beta*)* and beta is a chosen coefficient differs from applications such that the lambda value can be an integer larger than 1.

3. Experimental Results

Several experiments under the SoC benchmark circuit d695 from ITC'02 [2] were given to examine the effectiveness of the proposed algorithm. The program was implemented in MATLAB code and table-1 shows its results comparing with previous work as well as traditional ILP algorithm [3], simulated annealing algorithm SA [4] and genetic algorithm GA [1].

Table 3 shows the test time for the algorithm, ILP, SA, and GA under 16, 24, 32, 40, 48, 56, and 64 test bandwidths, respectively. Compared with the other three methods, the SoC test time obtained by the improved simulated annealing algorithm is almost shorter, especially when the test bandwidth is larger, the advantage of the algorithm used in this paper is more obvious.

Table 1. result comparing

Test benchmark	algorithm	Test width of TAM						
		16	24	32	40	48	56	64
		Test time of cycles						
D695	ILP	42952	28327	21423	17210	16975	16516	15694
	SA	42361	28638	21967	17414	15141	12692	11242
	GA	41949	28290	21329	17084	16975	14610	12960
	Previous work	42268	28289	21437	17523	14310	12462	10869
	Current work	42126	27864	21393	17177	14227	12304	10743

4. Summary

In this paper, an improved SA algorithm was proposed to solve the IP core allocation problem in SoC scheduling. By giving a feedback of the parameter lambda which is calculated from the error or output difference, the algorithm can result in better value than the previous work and also that of the traditional ILP, SA and GA. Further research can be carried on with the consideration of the constraint of the power consumption of SoC.

Acknowledgments

This work was partially supported by the provincial projects of Guangdong province for the next generation information technology (No.2015B090912001).

References

[1]. Lei Jia and Fang Gang. A Test Scheduling Method of SoC based on Genetic algorithm[J]. Chinese Journal of Scientific Instrument, 2007 28(4):15-18.

[2]. IEEE Std 1500-2005.IEEE 1500 Standard for Embedded Core Test[S].

[3]. V. Iyengar, K. Chakrabarty and E. J. Marinissen, Test wrapper and test access mechanism co-optimization for system-on-chip[C], Test Conference, 2001. Proceedings. International, Baltimore, MD, 2001, pp. 1023-1032.

[4]. Yang Jun and Luo Lan. SoC test schedule based on TCG and simulated annealing algorithm[J]. Journal of Circuits and Systems, 2006,05:37--43.

[5]. Cui Xiaole, Xiong Zhitian, Cheng Wei and Li Chongren. Thermal-aware SoC test scheduling method based on ant colony optimization[J]. Chinese Journal of Scientific Instrument, 2014, 04: 948-953.

[6]. Zheng Jingjing, Shen Zhihang, Gao Huaien, Chen Bianna, Zheng Weida and Xiong Xiaoming, An Improved SoC Test Scheduling Method Based on Simulated Annealing Algorithm, Journal of Physics Conference Series , 2017 , 806 (1) :012011.