Three-dimension Test Wrapper Design based on Multi-objective Cuckoo Search

Aijun Zhu[1,2,3], Chuanpei Xu[1,3], Jun Wu[1,3], Zhi Li[1,2,3,4] and Junhao Niu[1,2,3,*]

[1]School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin, 541004, P.R. China; [2]School of Mechano-Electronic Engineering, Xidian University, Xi'an, 710071, P.R. China; [3]Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin, 541004, P.R. China; [4]Guilin University of Aerospace Technology, Guilin 541004, P.R. China

Abstract: The main methodologies and recent patents on test wrapper design have been reviewed in this paper. The design of three-dimension test wrapper for IP module in System on Chip is a NP Hard problem. As the test time of an IP module is determined by the longest wrapper scan chain and each TSV has area costs and is a potential source of defects, it is necessary to shorten the longest wrapper chain and use less TSV. The paper proposes MOCS (Multi-Objective Cuckoo Search) algorithm to achieve the purpose of minimization of the IP module test time and the number of TSV used. The proposed algorithm, which is based on swarm intelligence, through levy flight operation and discovery rate, can achieve balance of the wrapper scan chains and use less TSV. Typical IP modules in ITC'02 benchmarks are used to prove the effectiveness of the proposed algorithm. Experimental results show that the algorithm can get better Pateto solutions Set, compared with NSGAII, and can reduce the test cost.

Keywords: Multi-objective cuckoo search, swarm intelligence, test wrapper.

1. INTRODUCTION

With the rapid development of modern microelectronic technology and nano-technology, integrated chip has entered the era of IP-based core design of SoC (System On Chip). Previous work is mainly confined to the traditional 2D (Two Dimensional) Integrated Chip [1-9] with the appearance of 3D (Three-Dimensional) Integrated Chip, design based on the embedded IP core would become a popular style for 3D integrated chip, there is an urgent need to study the design for testability methodologies and optimization problems of 3D SoC.

3D Integrated Chips overcome the shortcoming of 2D integrated chip, which is the rapid growth of internal wiring length caused by the increase of circuits size; therefore it would improves the performance of integrated circuits. TSVs (Through Silicon Vias) are used between layers of silicon chips. Using vertical connection method instead of the early edge alignment method makes internal wiring dramatically shortened, thereby reducing the transmission power and transmission delay and further increasing packaging density of integrated chips. In modular test framework based on IP core, the test resource allocation and test scheduling will directly affect the overall test performance of 3D SoC.

Design for testability of 3D SoC is a complex and difficult task, and there are a number of issues need to be done. Test wrapper design proved to be NP hard problem [1]. The test time of an IP core is determined by the longest wrapper scan chain of IP core, so the design of wrapper scan chain determines the test time and cost directly. Since input and output ports and TSV in 3D SoC are limited resources, the optimization design of wrapper scan chain can shorten test time and reduce the number of TSV used. Therefore, the total testing costs to address the key issue are reduced.

V.lyengar proposed a wrapper-design algorithm [2], which is based on the best fit decreasing (BFD) heuristics. The algorithm first allocate the internal scan chains within w wrapper scan chains in Step (1) to minimize the longest wrapper scan chain in part one; then, such procedure is repeated for Step (2), with each input cell and output cell being regarded as an internal scan chain whose length is one. But the BFD algorithm only possesses a local optimum capacity [5]. In addition, the BFD algorithm is only for two-dimensional SoC, and it did not consider the use of TSV situation.

B.Noia and K.Chakrabarty proposed a model based on integer linear programming algorithm [3]. The minimum goal of test time is achieved under the constraint condition of the finite TSV available. Since TSVs available are limited, the test time can not be guaranteed to be globally optimal. In addition, a method based on integer linear programming is only suitable for small -scale problem. Wu XX, etc. [1] proposed an optimization method, under the constraint of the number of TSV and power consumption, test time is to be minimized. This optimization method combines integer linear programming (ILP), LP-relaxation and randomized rounding to reduce testing time and test costs; it is also not suitable for solving large-scale problems. Yu Yang [6], etc.

proposed an algorithm called MVR (Mean Value Residue), but it does not consider the number of TSV used. Several methods above did not consider the balance between test time and TSV resource used. Integer Linear Programming, which is used frequently, is not suitable for problems with large-scale case. Because evolutionary algorithm based on swarm intelligence is suitable for large-scale problems, a multi-objective evolutionary algorithm based on swarm intelligence is proposed to solve the problem of test time minimization and the number of TSV used minimization.

Many patents on scan chain were invented for IP core test wrappers [10-13]. For example, the input, output and link instruction circuits for hierarchical 1500 wrappers were proposed, to give a method for testing the module and the interconnections within different modules [10]. Such test structure enables every one of the plural wrappers, which includes wrappers in modules embedded within other modules. Another scan test structure was proposed [11], for the convenience of semiconductor circuits' low power testing through partitioning the serial scan portions into shorter parts. A test access mechanism for diagnosis based on partitioning scan chains was proposed [12], to divide the scan cells into a set of non-overlapping partitions, and based on the partition pass/fail signals, a failure diagnosis process can be performed. There are also some other patents on scan chain we can find in [13-17].

Organization of the rest of this paper is as follows. Then the problem definition is introduced briefly in Section 2. Cuckoo Search with Levy flight is presented in Section 3. Three-dimension test wrapper design based on Multi-Objective Cuckoo Search is introduced in detail in Section 4. The MOCS algorithm is evaluated by a numerical simulation on ITC02 benchmarks, and experimental results are presented in Section 5. Finally, concluding remarks and current & future developments is given in Section 6.

2. PROBLEM DEFINITION

Generally, to facilitate the test of 3D SOC, a test wrapper is designed for each embedded IP core. According the actual need of design, the TAM (Test Access Mechanism) bus width is equal to the number of test wrapper chain. A test wrapper chain is usually composed of function inputs, function outputs and internal scan chain. During test course, TAM buses are used to feed test patterns to the CUT (Core Under Test) through its test wrapper. The test patterns act as stimuli to activate the circuit under test. After that, test responses are captured into scan registers. At last, the test responses are sent back to the test pins of ATE (Automatic Test Equipment) through TAM buses.

In the classical wrapper design method of BFD algorithm [2], Step (1), first allocate the internal scan chains within w wrapper scan chains to minimize the longest wrapper scan chain in part one; then, such procedure is repeated for Step (2), with each input cell and output cell being regarded as an internal scan chain whose length is one. Because part two is similar to part one, which is also a special case of part one, we only discuss part one in this paper. Suppose there is a core with n internal scan chain, which are Sc1, Sc2,…, Scn, respectively. We define a set Sc = {Sc1, Sc2,…, Scn}. The longest wrapper scan chain can be defined as follows:

Let C be a given subset, $C \subseteq Sc$. Let L(C) be a sum of length of every element in subset C, $L(C) = \sum_{c \in C} L(c)$. We can partition Sc into w wrapper scan chains, namely, $D = \{D_1, D_2, ..., D_z, ... D_w\}$, $\forall D_z, D_z \subseteq Sc,$ $z \in [1, w]$. We define $S(D) = \max_{1 \le z \le w} L(D_z)$ as the longest wrapper scan chain.

To illustrate our motivation, we choose core 8 of h953 from the International Test Conference 2002 Test Benchmarks, and design it with three layers. Suppose the TAM bus width is two. As we all know, the test time of a core is determined by the longest wrapper chain. So, if we make the longest wrapper chain be shorter, the test time can be reduced.

In Fig. (**1**), the number of TSV used by TAM1 is 4; the number of TSV used by TAM2 is 4, too. Therefore, the number of total TSV used is 8, and the length of the two wrapper scan chains are 940(188+188+188+188+188) and 567(189+189+189), respectively. Therefore, the length of the longest wrapper scan chain is 940.

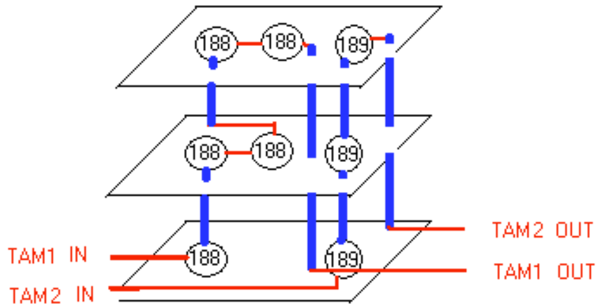

Fig. (1). Example of case a.

In Fig. (**2**), the number of TSV used by TAM1 is 4; the number of TSV used by TAM2 is 4, too. Therefore, the number of total TSV used is 8, and the length of the two wrapper scan chains are 753(188+188+188+189) and 754(188+188+189+189), respectively. Therefore, the length of the longest wrapper scan chain is 754. Compared with Fig. (**1**), the longest wrapper scan chain in Fig. (**2**) is shorter and need less test time, although the number of total TSV used in Fig. (**2**) is the same as that in Fig. (**1**).

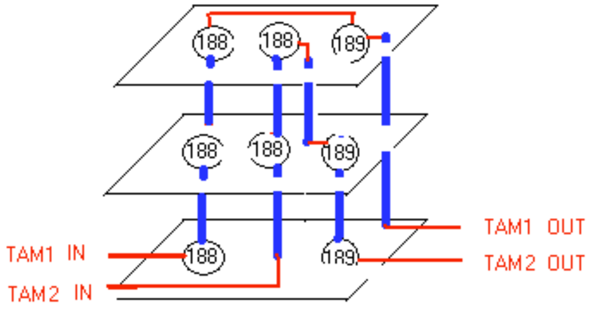

Fig. (2). Example of case b.

In Fig. (**3**), the number of TSV used by TAM1 is 4; the number of TSV used by TAM2 is 2. Therefore, the number of total TSV used is 6, and the length of the two wrapper scan chains are 753(188+188+188+189) and 754(188+188+

189+189), respectively. Therefore, the length of the longest wrapper scan chain is 754. Compared with Fig. (**2**), the longest wrapper scan chain in Fig. (**3**) is the same and need the same test time, but the number of total TSV used in Fig. (**3**) is fewer than that in Fig. (**2**). Compared with Fig. (**1**), not only the longest wrapper scan chain in Fig. (**3**) is shorter, but also the number of total TSV used in Fig. (**3**) is fewer. Therefore, Fig. (**3**) needs the fewest test time and need lest number of total TSV used. Through the design of three-dimensional test wrapper, we can make the test time shorter and need less number of TSV used, and this is the motivation of this paper.

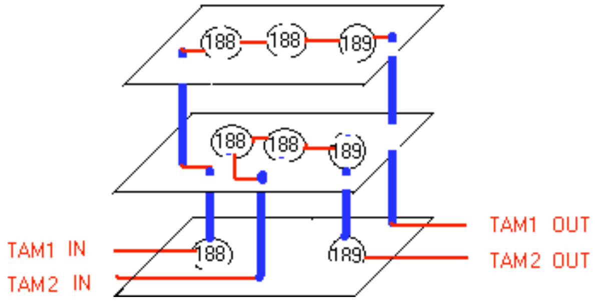

Fig. (3). Example of case c.

The problem of three-dimension test wrapper scan chain design is described as follows: Given a IP core, which contains n internal scan chains Sc1, Sc2, ..., Scn, and the corresponding length is Len(Sc1), Len (Sc2), ..., Len (Scn); Given the n internal scan chains are distributed on m layers, and the corresponding layers are Lay(Sc1), Lay(Sc2), ..., Lay(Scn), respectively; then, allocate the n internal scan chains within w wrapper scan chains, so that the length of the longest wrapper scan chain and the number of total TSV used are minimized.

Recent works have not considered the balance between the test time and the number of TSV used, while the popular method ILP is only suitable for problems with small size. Cuckoo search is a new method based on swarm intelligent and its global search capacity is very strong; Cuckoo search method has been widely used [18-20]. Therefore, multi-objective Cuckoo search algorithm is adopted to get the pateto optimal solutions.

3. CUCKOO SEARCH WITH LEVY FLIGHT

3.1. Cuckoo Typical Behavior

The sound that a cuckoo can make is beautiful, but what is profoundly interesting is its aggressive breeding strategy. A cuckoo may lay eggs in the other bird's nest; at the same time, in order to improve the hatching probability of its own eggs, it pushes the original eggs off the nest. If the host bird finds the alien eggs, it either throws away these eggs or builds a new nest in other place. Some cuckoos are good at laying eggs which are similar to the host bird' eggs and this reduces the probability of being discarded. Some infantile cuckoos can mimic the sound of the host bird, which improve the probability of being fed. Finally, the survival probability is improved [21].

3.2. Levy Flight

In the natural world, animals always find food randomly or in a quasi-random manner. In the general condition, the path by which animals find food is a random walk in effect; because the next movement is based on the current state or location and the transition probability to the next state. Which real direction it selects depends on a probability, which can be modeled in math. In fact, all kinds of researches have implied that the flight behavior of lots of birds and insects has shown the classical characteristics of levy flight [22, 23]. A recent research has shown that fruit flies can explore their landscape in a levy-flight -style manner [23]. Therefore, such typical behavior can be applied to optimal search and optimization. Recent experimental results have shown it is of promising capability for optimal search [24].

Generally speaking, Levy flight is a random walk whose step length complies with the levy distribution. Mathematically speaking, a straightforward version of Levy distribution [19] can be defined as follows:

$$L(s,\gamma,\mu)=\begin{cases}\sqrt{\frac{\gamma}{2\pi}}\exp\left[-\frac{\gamma}{2(s-u)}\right]\frac{1}{(s-u)^{3/2}} & if\ 0<\mu<s<\infty \\ 0 & if\ s\le 0\end{cases} \tag{1}$$

here μ is a minimum step size and γ is a scale factor. Evidently, if s is big enough in above equation, that is $s\to\infty$, then we have the following equation:

$$L(s,\gamma,\mu)\approx\sqrt{\frac{\gamma}{2\pi}}\frac{1}{(s)^{3/2}} \tag{2}$$

The above equation is a special case of the levy distribution.

4. THREE-DIMENTION TEST WRAPPER DESIGN BASED ON MULTI-OBJECTIVE CUCKOO SEARCH

Cuckoo search for optimization [20] adopt the following three idealized rules:(1) every cuckoo lays one egg at a time, and put it in a randomly selected nest; (2) elite strategy, that is the best nest with high quality egg can be kept to the next generation; (3) the number of the total host nests is constant, and a host bird can find an alien egg with a probability pa∈ [0,1]. If this case happen, the host bird either push the egg out of the nest or build another new nest in a new place.

Generally speaking, the first rule can be changed into a randomization operation; therefore, we can generate a new solution by random walk or by levy flight. The localized permutation is conducted over different solution, which can be also regarded as a form of crossover operation. In fact, the second rule is the corresponding elitism, which makes sure the best solution can be passed onto the next generation. Such strategy can make the algorithm converge appropriately. Besides, the third rule can be regarded as the mutation operation. Therefore, the worst solution is abandoned with a probability pa and a new solution is generated.

According to the three rules of cuckoo search, the proposed algorithm based on multi-objective Cuckoo search is described as follows:

Step (1): system initialization: according to the number of internal scan chains in the IP core, set the number of the dimension of the solution space d; according to the number of wrapper scan chains needed, set the value of w; set the value of population size NP; set the generation counter G=0; set the value of the probability of discovering alien eggs Pa; randomly generate the layer number of all the internal scan chains in the IP core; set the value of the maximum iterative generation MaxG.

Step (2): In the feasible region, generate a random initial population whose size is NP. That is to say, the NP nests are randomly generated.

Step (3): Use the objective function formula (7) and formula (8) to calculate the objective function values of each individual in the initial population.

Step (4): None-dominate-sort the initial population, then get the pateto set and pateto font of the initial population.

Step (5): According to Levy distribution, update the population, while those individuals with pateto font kept unchanged.

Step (6): Calculate the objective function values of each individual in the updated population, None-dominate-sort the updated population, then get the pateto set and pateto font of the updated population.

Step (7): According to the probability of discovering alien eggs Pa, abandon a few worse nests and build new nests, then get an updated population.

Step (8): Calculate the objective function values of each individual in the updated population, None-dominate-sort the updated population, then get the pateto set and pateto font of the updated population..

Step (9): Generation counter G=G+1;.

Step (10): Judge whether the number of iterative generation G reach MaxG or not; if not, turn Step (5). If it reaches MaxG, then output the pateto set and pateto font.

4.1. Solution Encoding

Because this paper is to solve the problem of the partition of internal scan chains, and the number of internal scan chains and their corresponding scan chain length are all discrete integers, so the real number (integer) coding scheme is adopted here.

Definition 1: a cuckoo, an egg or a nest is a solution $\mathbf{H}_k, \forall \mathbf{H}_k \in L^{\mathrm{d}}, k \in [1, NP]$, here NP is the population size of cuckoo; d is number of dimension of a solution; $\mathbf{H}_k = (L_1, L_2, ..., L_d)$ is a candidate solution, which is a solution for the three dimensional test wrapper design.

Definition 2: for the variable L_i, $\forall L_i, i \in [1, d], L_i \in Z$, here Z is an integer set and $Z = \{1, 2, ..., w\}$; here w is the number of wrapper scan chains.

For example, d695.soc in ITC02 benchmark [25] has 16 IP cores (modules); where IP core 6 has 16 internal scan chains, namely, {S1, S2,…, S16}; their length are 41, 41, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 39, 39, 39, 39, respectively. Since d is determined by the number of internal scan chain in IP core, d equals 16. If we want to partition 16 internal scan chains of IP core 6 in d695.soc among 2 wrapper scan chains, that means w = 2. As a matter of fact, every internal scan chain belongs to either the first wrapper scan chain or the second wrapper scan chain; therefore, the value of each gene of a candidate solution is either 1 or 2. Assume a candidate solution X is equal to (1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2) with 16 dimensions, which means odd-numbered series of internal scan chains could be connected together to make up the first wrapper scan chain, while even-numbered series of internal scan chains could be connected together to make up the second wrapper scan chain.

4.2. System Initialization

The population size NP is set as 50; the maximum number of generation MaxG is set as 500; set the generation counter G=0; set the value of the probability of discovering alien eggs Pa as 0.25; according to the number of current wrapper scan chains needed, set the value of w; according to the number of internal scan chains in the current IP core, set the value of the dimensional number of the solution space d.

4.3. Population Initialization

The initial population (NP nests) is generated as follows : Nest(i,:)=round(Lb+(Ub-Lb). rand(1,d)), where i=1,2,…NP; Ub=w; Lb=1; d is the number of the solution space ; round(x) is a function which rounds the elements of x to the nearest integers.

4.4. Strategy of Generating New Solution

When a new solution $\mathbf{H}_k^{(t+1)}$ is generated for, say a cuckoo k, a levy flight is carried out.

$$\mathbf{H}_k^{(t+1)} = \mathbf{H}_k^{(t)} + \alpha \oplus Levy(\beta) \tag{3}$$

where α is greater than zero and it is the step size which is related to the size of the problem to be solved. Generally speaking, we can make α equal 1. We can also make use of the difference of two random solutions to generate α as follows:

$$\alpha = \alpha_0 (\mathbf{H}_j^{(t)} - \mathbf{H}_i^{(t)}) \tag{4}$$

where the expression in the bracket is the difference between two random solutions; α_0 is a constant. This expression mimics the fact that similar cuckoo eggs are hardly to be found out. Therefore, new solutions are generated according to the proportion of the difference.

The symbol $\oplus$ expresses an entry-wise multiplication. In essence, Levy flight is a random walk and the random steps are calculated by a levy distribution for large steps

$$Levy \sim u = t^{-1-\beta}, (0 < \beta \le 2), \tag{5}$$

4.5. Abandon bad Solutions with a Probability of Pa

According to the probability of discovering alien eggs Pa, a few bad nests are abandoned. By None-dominate-sorting operation, we can get the worst nests. Naturally, we should use the conception pateto dominate, which is defined as follows:

Definition 3: (pateto dominate) if a candidate solution $\mathbf{H}_s$ dominate another candidate solution $\mathbf{H}_t$, that is $H_s \prec H_t$, then both the following conditions should be satisfied: (1) for all objective functions, $\mathbf{H}_s$ is not worse than $\mathbf{H}_t$; that is to say, $\forall i \in [1,k]$, $f_i(H_s) \le f_i(H_t)$; Here k is the number of objective functions; (2) for all objective functions, there is at least one objective function, which can make $\mathbf{H}_s$ be strictly better than $\mathbf{H}_t$; that is to say, $\exists i \in [1,k]$, $f_i(H_s) < f_i(H_t)$.

At the same time new nests are built according to the difference of two randomly selected nests.

$$\mathbf{H}_k^{(t+1)} = \mathbf{H}_k^{(t)} + rand * (\mathbf{H}_i^{(t)} - \mathbf{H}_j^{(t)}) \qquad (6)$$

where rand is a function which returns a random value between zero and one.

4.6. Objective Functions

To evaluate the fitness of cuckoo, we define the following two objective functions

$$\min obj1 = \sum_{i=1}^{i=w} (L(P_i) - \frac{1}{n} \sum_{j=1}^{j=n} (L(Sc_j))^2, \qquad (7)$$

where $L(P_i)$ a sum of every internal scan chain's length in the i-th wrapper scan chain; $L(Sc_j)$ is the j-th internal scan chain's length; n is the internal scan chains' amount; w is the wrapper scan chains' amount. This objective function is to evaluate the balance of all the wrapper scan chains. The smaller the objective function obj1 is, the shorter the longest wrapper scan chain is.

To evaluate the number of TSV that all wrapper scan chains need, we define objective function obj2 as follows:

$$\min obj2 = \sum_{i=1}^{i=w} N_{TSV_i} \qquad (8)$$

here N_{TSV_i} is the number of TSV used by the i-th wrapper scan chain, $i \in [1, w]$, N_{TSV_i} is calculated as follows:

$$N_{TSV_i} = 2 \times Max_{1<=j<=x}(Lay_{ij}) \qquad (9)$$

here lay_{ij} is the layer number of the j-th internal scan chain in the i-th wrapper scan chain; x is the number of internal scan chain in the i-th wrapper scan chain. All the wrapper scan chains start and end at the bottom layer, because all test pins are located at the bottom layer. Suppose the number of the bottom layer is 0, and the number of the next one is 1, and so on.

4.7. Boundary Checking

Check the boundary of each gene of each individual in the updated population; if the value of a gene is less than Lb, then change its value to Lb, if it is greater than Ub, then change its value to Ub. Here Ub=w and Lb=1.

5. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed algorithm, we select two classical benchmarks p22810 core 5 and p34392 core 2 from ITC'02 benchmarks [25]. The reason is that the two SoC are almost the most complex and the corresponding cores are unbalanced seriously in the internal scan chains.

The proposed algorithm is programmed by MATLAB code. We set the maximum iteration number MaxG as 500;the population size is set as 50; set the value of the probability of discovering alien eggs Pa as 0.25; according to the number of current wrapper scan chains needed, set the value of w; according to the number of internal scan chains in the current IP core, set the value of the dimensional number of the solution space d. Because there are no information of layer for the internal scan chains in the ITC'02 benchmarks, we suppose all internal scan chains in every core are randomly distributed on three lays in this paper, and the number of the bottom layer is 0; the number of layer is increased progressively by one upward from the bottom lay, that is to say, the number of the layer is 0, 1 and 2 respectively from the bottom to the top. Because every input of Test Access Mechanism starts on the bottom layer and every output of Test Access Mechanism ends on the bottom layer, every wrapper scan chain starts and ends on the bottom layer.

In Table **1**, the first column M means the methods to solve the three-dimension test wrapper design. To illustrate the effectiveness, we compare our proposed algorithm MOCS with the classical method NSGAII (Nondominated Sorting Genetic Algorithm II) [26]. The second column Pateto set means the optimal set (Pateto set) obtained by corresponding method. The third column L is the length of the longest wrapper scan chain, and the fourth column Ntsv is the number of TSV used. In each of methods, the third column L along with the fourth column Ntsv is called Pateto Front. The parameters in NSGAII are as follows: the maximum iteration number is the same as that of MOCS, which is 500; the population size is the same as that of MOCS, which is 50. Table **1** is the experimental results of p22810 core 5. Table **1** show that, when the number of TSV used in MOCS and NSGAII are both 12, the length of the longest wrapper scan chain L in MOCS is 759 and that in NSGAII is 803. Obviously, the proposed algorithm can get shorter L; therefore, we can reduce the test time for corresponding IP core. When L equal 10 or 8, we also get shorter L. We conclude that the proposed algorithm can get shorter the longest wrapper scan chain with the same number of TSV used.

Table 1. Experimental results of p22810 core 5, w=3.

		Pateto Front	
M	**Pateto set**	**L**	**Ntsv**
MOCS	(2,3,1,2,3,3,3,2,2,1,1,1,2,3,3,1,1,1,2,2,3,3,1,3,1,3,1,1,1)	759	12
	(2,1,3,1,1,3,3,1,1,2,3,3,2,3,1,2,2,1,3,3,2,3,1,2,1,1,2,1,2)	783	10
	(3,2,2,1,1,1,1,1,2,3,2,1,3,1,2,2,2,2,1,2,2,2,3,2,2,1,1,2,2)	1001	8
NSGAII	(2,1,3,1,3,1,3,2,1,3,3,2,3,2,2,3,1,3,2,2,1,2,1,2,2,1,3,1,1)	803	12
	(3,1,2,1,3,3,2,1,2,2,1,2,1,1,2,2,3,3,1,3,3,3,2,2,2,2,3,2,2)	824	10
	(3,2,2,3,2,2,1,1,1,1,1,2,3,2,1,3,1,2,2,2,2,1,2,2,2,3,2,2,1)	1024	8

Table **2** is the experimental results of p34392 core 2. All the symbols in Table **2** are the same as in Table **1**. We also can conclude that in the case of w=3, the proposed algorithm can get shorter L with the same number of TSV used.

Table 2. Experimental results of p34392 core 2, w=3.

		Pateto Front	
M	**Pateto set**	**L**	**N t s v**
MOCS	(1,3,3,2,1,3,2,1,3,3,1,2,2,2,1,3,1,3,1,1,3,3,3,1,3,2,3, 2,1)	3016	12
	(2,3,3,2,1,3,2,1,3,2,3,1,2,2,1,2,2,1,1,3,1,3,2,3,3,1,3,3, 3)	3508	10
NSGAII	(3,2,1,3,3,3,2,1,1,1,1,2,2,3,2,2,2,3,3,2,2,1,2,2,2,2,2,3,1)	3062	12
	(1,3,2,1,3,1,2,3,2,2,2,2,3,1,1,2,2,1,1,3,2,2,2,2,2,2,2,2,1)	3616	10

To compare the proposed algorithm with NSGAII further, we introduce the concept SC (Set Coverage) [27] as follows.

Definition 4: suppose a none dominant solutions set A={a_1,a_2,…,a_n} and another none dominant solutions set B={b_1,b_2,…,b_m}, we define SC(A,B) as the part of none dominant solutions in set B, which are covered by none dominant solutions set A.

$$SC(A,B) = \frac{|\{b_i \in B; \exists a_i \in A : (a_i \prec b_i) \vee (a_i = b_i)\}|}{|B|} \qquad (10)$$

where |B| is the number of elements in set B. the above definition can be explained like this: suppose for a member in B, we can find another member in A, which pateto dominants the member in B; then calculate all such members in B, and SC(A,B) is the number of such members divided by the number of members in B. The bigger SC(A,B) is, the higher the proportion of B covered by A. Therefore, The bigger SC(A,B) is, the better is A than B. While SC(A,B)=1 and SC(B,A)=0, it means B is totally covered by A. while SC(A,B)=1 and SC(B,A)=1, it means B = A.

Because operation of pateto dominant is asymmetric, SC(A,B) maybe not equal 1- SC(B,A). It means we must calculate SC(A,B) and SC(B,A) at the same time. To compare the average performance of MOCS and NSGAII, we calculate the value of SC, when w is assigned different value (w=2,3,…,16).

In Table **3**, A and B is the set in SC(A,B). We calculate different SC(A,B) when w is assigned from 2 to 16. Experimental result show that the average value of SC(MOCS, NSGAII) = 94.2 % is bigger than the average value of SC(NSGAII, MOCS) = 31%. We can conclude from Table **3** that the none dominant solutions set obtained by the proposed algorithm MOCS is better than the none dominant solutions set obtained by the classical method NSGAII as a whole.

In Table **4**, experimental result show that the average value of SC(MOCS, NSGAII)= 84.44% is bigger than the average value of SC(NSGAII, MOCS,) = 20.68%. We can conclude from Table **4** that the none dominant solutions set obtained by the proposed algorithm MOCS is better than the none dominant solutions set obtained by the classical method NSGAII as a whole.

Table 3. Experimental results of p22810 core 5, SC(A,B).

A **B**	**MOCS** **NSGAII**	**NSGAII** **MOCS**
w=2	100%	0%
w=3	100%	0%
w=4	100%	0%
w=5	80%	25%
w=6	100%	33%
w=7	85%	57%
w=8	100%	30%
w=9	100%	20%
w=10	83%	50%
w=11	100%	60%
w=12	90%	20%
w=13	100%	40%
w=14	100%	80%
w=15	75%	30%
w=16	100%	20%
Average	94.2%	31%

Table 4. Experimental results of p34392 core 2, SC(A,B).

A **B**	**MOCS** **NSGAII**	**NSGAII** **MOCS**
w=2	100%	0%
w=3	100%	0%
w=4	100%	0%
w=5	90%	30%
w=6	75%	50%
w=7	80%	50%
w=8	66.6%	33.3%
w=9	83%	67%
w=10	100%	0%
w=11	100%	0%
w=12	80%	0%
w=13	75%	25%
w=14	67%	0%
w=15	50%	25%
w=16	100%	30%
average	84.44%	20.68%

6. CURRENT & FUTURE DEVELOPMENTS

The paper has reviewed main methodologies and recent patents on the test wrapper scan chain for IP module, and presented a novel technique to shorten the longest wrapper scan chain and minimize the number of TSV used, in order to reduce the test cost of IP module. The technique is applied on the internal scan chain of IP module using the proposed algorithm MOCS to test its efficacy. In almost all the benchmarks of International Test Conference 2002 considered in the research have proven that the proposed algorithm is worth potential to reduce the test time of IP Module. The new algorithm is based on swarm intelligence, through levy flight operation and discovery rate, can achieve balance of the wrapper scan chains and use less TSV. Typical IP modules in ITC'02 benchmarks are used to prove the effectiveness of the proposed algorithm. Experimental results show that the algorithm can get better Pateto solutions Set, compared with NSGAII, and can reduce the test cost. Furthermore, the proposed algorithm is simple and flexible enough to apply to wide areas in the real world. In the near future, we will apply the proposed technique on test scheduling for 3D stacked SoC.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

We wish to acknowledge the National Natural Science Foundation of China (60766001). This work was also supported by the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (Grant No. YQ14110), Program for Innovative Research Team of Guilin University of Electronic Technology (IRTGUET).

REFERENCES

[1] X. X. Wu, Y. B. Chen, and K. Chakrabarty, "Test-access mechanism optimization for core-based three-dimensional SOCs", *Microelectron. J.*, vol. 41, no. 10, pp. 601-615, 2010.

[2] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and test access mechanism co-optimization for system-on-chip", *Int. Test Conf.*, USA, pp. 1023-1032, 2001.

[3] B. Noia, and K. Chakrabarty, "Test-wrapper optimization for embedded cores in TSV-based three-dimensional SOCs", *IET Comput. Dig. Tech.*, vol. 5, no. 3, pp. 186-197, 2011.

[4] X. L. Cui, and W. Cheng, "Test wrapper and test access mechanism co-optimization for SoC based on ant colony algorithm", *J. Comput. Aided Des. Comput. Graph.*, vol. 21, no. 4, pp. 461-466, 2009.

[5] D. H. Niu, H. Wang, S. Y. Yang, and B. Cheng, "Re-optimization algorithm for SoC wrapper-chain balance using mean-val-ue approximation", *Tsinghua Sci. Tech.*, vol. 12, no. S1, pp. 61-66, 2007.

[6] Y. Yu, Y. F. Cheng, and Y. Peng, "Wrapper scan chain balance algorithm based on mean-value allowance", *Chin. J. Sci. Instrum.*, vol. 32, no. 10, pp. 2290-2296, 2011.

[7] E. J. Marinissen, G. S. Kumar, and L. Maurice, "Wrapper design for embedded core tes", *Int. Test Conf.*, USA, pp. 911-920, 2000.

[8] A.J. Zhu, Z. Li, and W.C. Zhu, "Wrapper scan chain design algorithm for SoC test based on biogeography optimization", *Chin. J. Sci. Instrum.*, vol. 33, no. 12, pp. 2774-2780, 2012.

[9] A. J. Zhu, Z. Li. W. C. Zhu, and C. Xu, "Design of test wrapper scan chain based on differential evolution", *J. Eng. Sci. Tech. Rev.*, vol. 6, no. 2, pp. 10-14, 2013.

[10] W. D. Lee, "*Input, output, and link instruction circuits for hierarchical P1500 wrappers*", U. S. Patent 8667351, March 4, 2014.

[11] W. D. Lee, "*Core circuit test architecture*", U. S. Patent 8656237, February 18, 2014.

[12] W. T. Cheng, M. Sharma, A. Dutta, R. B. Benware, and M. A Kassab, "*Test access mechanism for diagnosis based on partitioning scan chains*", U. S. Patent 8607107, December 10, 2013.

[13] R. Kapur, S. Jyotirmoy, U. Rajesh, P. Notiyath, T. Fernandes, S. Kulkarni, and A. Anbalan, "*Test design optimizer for configurable scan architectures*", U. S. Patent8584073, November 12, 2013.

[14] D. Juergen, D. Matthias, and R. Leuchter, "*Automation of tie cell insertion, optimization and replacement by scan flip-flops to increase fault coverage*", U. S. Patent 8,572,543, October 29, 2013.

[15] R. Ackerman, and J. Raykowski, "*Scan chain fault diagnosis*", U. S. Patent 8572543, October 29, 2013.

[16] C. R. Conklin, M. F. Fee, A. Martens, W. Niklaus, S. B. Swaney, and T. Webel, "*Updating settings of a processor core concurrently to the operation of a multi core processor system*", U. S. Patent 8499144, July 30, 2013.

[17] H. Kukreja, and D. Agrawal, "*Method for partitioning scan chain*", U. S. Patent 8504886, August 6, 2013.

[18] X. S. Yang, and S. Deb, "Multi-objective cuckoo search for design optimization", *Comput. Oper. Res.*, vol. 40, no. 2013, pp. 1616-1624, 2013.

[19] X. S Yang, *Engineering optimisation*: *An introduction with met heuristic applications:* John Wiley and Sons, 2010.

[20] X. S Yang, and S. Deb, "Cuckoo search *via* Levy flights", In: *Proc. World Congr. Nat. Biol. Insp. Comput.,* (NaBIC2009India), *IEEE Publications*, USA, pp. 210-4, 2009.

[21] R. B. Payne, M. D. Sorenson, and K. Klitz, *The Cuckoos*. Oxford University Press; 2005.

[22] C. Brown, L. S. Liebovitch, and R. Glendon, "Levy flights in Dobe Ju/'hoansi foraging patterns", *Hum. Ecol.*, vol. 35, pp. 129-38, 2007.

[23] A. M. Reynolds, and Frye MA, "Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search", *PLoS One* vol. 2, p. 354, 2007.

[24] I. Pavlyukevich, "Levy flights, non-local search and simulated annealing", *J. Comput. Phys.*, vol. 226, pp.1830-44, 2007.

[25] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, "A set of benchmarks for modular testing of SOCs", *Int. Test Conf.*, USA, pp. 519-528, 2002.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm NSGA-II", *IEEE Transact. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.

[27] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multi-objective optimizers: An analysis and review", *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117-132, 2003.

Received: September 22, 2014 Revised: November 30, 2014 Accepted: December 02, 2014