

A New Wrapper Scan Chain Balance Algorithm for Intellectual Property in SoC

Aijun Zhu¹, Zhi Li^{1,2,3,4}, Chuanpei Xu^{2,3} and Wangchun Zhu^{2,3,*}

¹School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China

²School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China

³Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin 541004, China

⁴Guilin University of Aerospace Technology, Guilin 541004, China

Abstract: Recent patents and progress on scan chain balance algorithms have been reviewed. With a significant increase of the SoC (System on Chip) integration and scale, the test time of SoC increase dramatically, and this makes the test cost of SoC grow rapidly. In order to reduce test cost and expense, the paper proposes an OBBO (Opposition-based learning and Biogeography Based Optimization) algorithm and designs wrapper scan chains for the IP (Intellectual Property) using OBBO algorithm, which can make wrapper scan chains equilibration so that we can make the test time of IP be minimum. The new method is a random optimization algorithm which combines BBO (Biogeography Based Optimization) algorithm with OBL (Opposition-based learning). By using migration operation, mutation operation and OBL operation, we achieve a balance between different wrapper chains so that we can shorten the wrapper scan chain which is longest. Experimental results show that OBBO can obtain shorter longest wrapper scan chain in most case and at the same time the convergence speed can be faster.

Keywords: Optimization algorithm, SoC test, wrapper Scan Chain.

1. INTRODUCTION

With the development of modern semiconductor manufacturing processes and the application of nanotechnology, the integration of the LSI (Large Scale Integration) is constantly increasing; therefore, SoC obtains a good opportunity of rapid development. To achieve the aim of reducing the time-to-market of SoC, without reducing the stability and reliability of the system, the SoC design with reuse-based IP core has gradually become the mainstream style. Because of the substantial increase of the SoC integration and scale, this brings new challenges and problems, for instance dramatic increase in the SoC test time and rapid growth in the SoC test costs. Therefore, SoC test has become a difficulty problem in the SoC design and manufacturing process, which is also a bottleneck restricting its development [2-5, 19-23]. The main motivation of this paper is to optimize the wrapper scan chains so as to make their balance become better and reduce the length of the longest scan chain of IP core. Since an IP's test time must be mainly determined by the longest scan chain's length of an IP, therefore, by making the balance of wrapper scan chains be better, we can reduce the length of the longest scan chain of IP core. Thus, the IP module's test time is reduced and SoC test costs are

minimized. For example, module 4 of SoC d695 in ITC02 benchmark [10] has four internal scan chains whose lengths are 54, 53, 52 and 52 respectively.

Now, we want partition four internal scan chains to three wrapper scan chains. There are three partitions ($\{54+52, 53, 52\}$, $\{54, 53+52, 52\}$, $\{54, 53, 52+52\}$), whose longest lengths are 106, 105 and 104 respectively. The third partition is the best, because this longest scan chain's length is the shortest. In other word, the third partition has better balance.

To achieve the aim of reducing the SoC's test time, VIK-RAM IYENGAR, *et al.* [2] partitions Wrapper / TAM and test scheduling combination optimization problem into the P_W , P_{AW} and P_{PAW} three sub-problems. It is easy to find out from the division, that test Wrapper design (P_W) is the basis of the above sub-problems. If the designed wrapper does not reach the best balance, then we can not obtain the optimal solution for the other problems. Since the test time of IP module is determined by the longest scan chain of IP core, and therefore how to optimize balance design for test wrapper scan chain will directly determine the time and cost of the test.

The classic method for test wrapper design is BFD (Best Fit Decreasing) [2]. This heuristic method has two steps: (1) partitioning every internal scan chain in the module, (2) assigning wrapper input and output cells to the partially designed scan chains generated in the first step. In Step (1) each internal scan chain could be allocated to the scan chain, whose current length after such allocation should be nearest

to, but does not outnumber the current longest scan chain' length. If we could not find such scan chain, then the internal scan chain could be allocated to the present shortest scan chain. 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. However, the BFD algorithm only has local optimization ability [4]; the fundamental reason is that it only considers the length of each current scan chain when every internal scan chain is sequentially appended to the corresponding scan chains.

Since the BFD algorithm is lack of global optimization ability, NIU D H, *et al.* [4] introduces a global optimization ideology and proposes a re-optimization algorithm for SoC wrapper-chain balance using Mean-Value Approximation (MVA), which uses the average value of the whole internal scan chains to guide the global optimization. However, the MVA algorithm also exist some problems, i.e. it does not always give preferential treatment to the current longest internal scan chain.

In order to improve MVA algorithm, YU Y, *et al.* [3] proposes a wrapper scan chain balance algorithm based on Mean-Value Allowance Residue (MVAR). The MVAR algorithm first calculate the average value of all the internal scan chains, and then add an appropriate residue to guide global optimization on the basis of the average value. However, the MVAR algorithm also has its drawback: it is hard to select an appropriate residue, in other words, it is sensitive to the residue. For example, if the residue is selected as more than 5 percent of the mean value, the MVAR algorithm is worse than the MVA algorithm.

In a word, all these algorithms aim at making wrapper scan chains as balanced as possible, thus shortening the longest wrapper scan chain to reduce test time of the IP module. Since wrapper design problem is NP hard [2], this paper proposes an OBBO algorithm to solve this problem, because it is based on swarm intelligence method and it is suitable for NP problems. With the advantage of population, it can partition the serious "unbalanced" internal scan chains into balanced wrapper scan chains. The typical SoC "unbalanced" IP modules are verified by the experiment and experimental results show the superiority of the OBBO algorithm.

Many patents on scan chain were invented for IP core test wrappers [10-13]. For example, 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].

Wrapper scan chain balance algorithm based on OBBO for IP module has not been reported, so the algorithm is proposed to solve the NP hard problem. The rest of this paper is organized as follows. The problem formulation is given and discussed in Section 2. Then the original BBO is introduced briefly in Section 3. The learning based on opposition is presented in Section 4. The new and novel algorithm is introduced in detail in Section 5. The OBBO algorithm is evaluated by a numerical simulation on ITC02 benchmarks, and experimental results are given in Section 6. Last, Section 7 concludes the whole paper.

2. PROBLEM FORMULATION

After IP cores are sold by the IP core vendor, some relative information about IP cores is also provided to help the SoC integration developers. Given a set called CoreInfor_i, which denotes relative information about IP core C_i, CoreInfor_i={NumIn_i, NumOut_i, NumIO_i, NumScan_i, {LenScan_{ij}, j∈[1, NScan_i]}}, where NumIn_i is the functional input port's amount, NumOut_i is functional output port's amount, NumIO_i is functional bidirectional port's amount, NumScan_i is internal scan chains's amount, and {LenScan_{ij}, j∈[1, NumScan_i]} is the set of all the internal scan chains' length. As the SoC integration developers design the test wrapper, they combine IP core test ports with internal scan chains in series to form several wrapper scan chains, which could be used for loading the test vectors and collecting test response.

We could represent mathematic model of wrapper design as the following: Given an input port set IN = {IN₁, IN₂, ..., IN_i, ..., IN_m}, each input cell corresponds to an input port, and its length L(IN_i) = 1, i ∈ [1, m]. Given an output port set OU = {OU₁, OU₂, ..., OU_j, ..., OU_g}, each output cell corresponds to an output port, and its length L(OU_j) = 1, j ∈ [1, g]. Given an bidirectional port set BIO = {BIO₁, BIO₂, ..., BIO_k, ..., BIO_u}, each bidirectional cell corresponds to an bidirectional port, and its length L(BIO_k) = 1, k ∈ [1, u]. Given a set of n internal scan chains of the IP core Sc = {Sc₁, Sc₂, ..., Sc_v, ..., Sc_n}, and the length of each internal scan chain equal L(Sc_v), v ∈ [1, n].

In BFD algorithm [2], first allocate the internal scan chains within w wrapper scan chains so as to make the longest wrapper scan chain be minimum in part one; Next the above process is repeated for part two, considering the input cell and output cell as internal scan chains of length 1. 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. 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, $D = \{D_1, D_2, \dots, D_z, \dots, D_w\}$, $\forall D_z, D_z \subseteq Sc, z \in [1, w]$. We define $S(D) = \max_{1 \leq z \leq w} L(D_z)$ as the longest wrapper scan chain.

3. BIOGEOGRAPHY BASED OPTIMIZATION

Biogeography Based Optimization algorithm is one type of random optimization approach, which was proposed by

D.Simon [1] in 2008. The BBO algorithm is composed of a population of candidate individuals or solutions which are named habitats or islands [7-9]. Every dimension of a candidate solution is known as a suitability index variable (SIV). The fitness of every individual or candidate solution could be measured by habitat suitability index (HSI) that is the same as the fitness in the other evolutionary computation algorithms with a candidate population. Habitats with many species are called good solutions with high HSI, and habitats with few species are called poor solutions with low HSI. Good solutions or individuals are easy to give their wonderful features to poor individuals. At the same time, poor individuals are much more willing to adopt other fresh characters of good individuals.

For original BBO algorithm, every individual or candidate solution is of the corresponding immigration rate Im and emigration rate Em . Both Im and Em are the function of species' amount in the island or habitat. Suppose the island has few species, then its Im is high and Em is low. Similarly, Suppose the island has many species, then its Im is low and Em is high. The Im and Em could be formulated as the following:

$$Em_k = E.k / n \quad (1)$$

$$Im_k = I.(1 - k / n) \quad (2)$$

Where k indicates the species' amount in current island and n indicates the largest number of species contained in current habitat; E indicates the largest emigration rate when current island has no species; I indicates the largest immigration rate when the island has contained the largest species' amount; Suppose all the habitats contain the same largest number of species.

Suppose the probability of occurrence for an island having just right k species is Pr_k , and Pr_k varies between time t and time $t + \Delta t$ just like follows [1]:

$$Pr_k(t + \Delta t) = Pr_k(t).(1 - Im_k.\Delta t - Em_k.\Delta t) + Pr_{k-1}(t).Im_{k-1}.\Delta t + Pr_{k+1}(t).Em_{k+1}.\Delta t \quad (3)$$

Where Im_k is immigration rate when the current habitat has k species; Im_{k-1} is immigration rate when the current habitat has $k-1$ species; Em_k is emigration rate when the current habitat has k species; Em_{k-1} is emigration rate when the current habitat has $k-1$ species; Pr_{k+1} is the probability of occurrence for the current island having just right $k+1$ species; Pr_{k-1} is the probability of occurrence for the current island having just right $k-1$ species.

Suppose we let Δt be as small as possible, then we can ignore the probability that an island has more than one emigration or immigration. Let $\Delta t \rightarrow 0$, we could calculate the limit of (3), obtain (4):

$$Pr_k = \begin{cases} -(Im_k + Em_k).Pr_k + Em_{k+1}.Pr_{k+1}, & k = 0 \\ -(Im_k + Em_k).Pr_k + Im_{k-1}.Pr_{k-1} + Em_{k+1}.Pr_{k+1}, & k \in [1, S_{max}-1] \\ -(Im_k + Em_k).Pr_k + Im_{k-1}.Pr_{k-1}, & k = S_{max} \end{cases} \quad (4)$$

Where S_{max} indicates the largest species' amount in a single island, other parameters are the same as equation (3). If a given individual X with k species is of a small Pr_k , then

we regard it almost impossible that it will be selected as a solution. Therefore, such solution is easier to change into another better individual or solution. Similarly, if a given habitat X with k species has a big Pr_k , then we regard it almost impossible that it will mutate to other worst individuals. We can define a mutation rate m_k , which is in inverse proportional to Pr_k .

$$m_k = m_{max}.(1 - Pr_k) / Pr_{max} \quad (5)$$

Where m_{max} is the maximum number of mutation rate, which is set at system initialization; Pr_{max} is the maximum probability in all solutions, which is presented as follows.

$$Pr_{max} = \max_{0 \leq k \leq S_{max}} Pr_k \quad (6)$$

4. OPPOSITION-BASED LEARNING

Evolution computation methodologies generally begin from a random population (solutions). Then they improve the fitness of the population towards some optimum solutions through the crossover, mutation and the other operations, gradually approaching the optimum individual. When they are searching the optimum individual, the searching procedure does not stop until the conditions defined in advance are met. The conditions defined in advance can be maximal iterative generations or a predetermined range of precision. At the beginning, we almost do not know anything about the optimum solution. Therefore, we begin from some stochastic guesses. Therefore, the time for such computation is highly determined by the distance between initial guesses or candidate solutions and the optimum solution. Through examining the opposite individual, of course we could obtain better opportunity of beginning from a nearer individual. Through such method, a better solution could be selected to be a beginning solution. Actually, as we all know, the probability that any one random guess is further than its corresponding opposite guess is fifty percent. So beginning from the closer guess of a stochastic guess and its corresponding opposite individual obtains a possible opportunity for accelerating convergence speed. So, we could use this strategy not only on the beginning individuals, but also on every individual of each generation. We first give the definition of opposite numbers before the OBL is introduced [6].

Let $x \in [c, d]$ be an integral number between c and d , the opposite number y is defined as follows:

$$y = c + d - x \quad (7)$$

The above definition is the case of the one-dimensional space; similarly, the above definition could be expanded to multi-dimensional space as the following.

Suppose $X = (X_1, X_2, \dots, X_i, \dots, X_n)$ be a vector which belongs to n -dimensional space, where any X_i is an integral number between c and d , and i is between 1 and n . The opposite vector $Y = (Y_1, Y_2, \dots, Y_n)$, and its i -th dimension is as follows:

$$Y_i = c + d - X_i, i \in [1, n] \quad (8)$$

Given the definition of opposite vector, the learning based on opposition could be defined:

Let $X = (X_1, X_2, \dots, X_n)$ be a vector which belongs to n -dimensional space, in other words, X is a candidate individual. Suppose $cf(\cdot)$ be a cost function that is to calculate the candidate individual's cost. Based on the above equation, we regard that $Y = (Y_1, Y_2, \dots, Y_n)$ could be the Opposition of X . If we obtain $cf(Y) \leq cf(X)$, the point X can be replaced by the point Y , because the cost of X is greater than that of Y ; if not, we could go on with point X . Therefore, the vector and its opposite vector are assessed meanwhile so as to continue with the fitter one with smaller cost.

5. PROPOSED ALGORITHM

As we all know, similar to other evolutionary computation methods, two primary procedures must be popular for BBO, in other words, how to initialize population and how to generate a new generation through computation operations or whatever migration and mutation. Of course, we introduce a new operation, namely, OBL operation, to strengthen the above two steps. The motivation is to speed up the convergence speed for original BBO through adding the learning based on opposition operation into original BBO [6], at the same time, in order to reduce the longest wrapper scan chain's length, so as to make IP module's test time be minimum.

5.1. Encoding

Evolutionary computation algorithm usually has two encoding scheme: binary encoding and real number encoding. However, there is a big redundancy in binary coding scheme in general. Since this study is to solve the problem of the partition of internal scan chains within w wrapper chains, at the same time, the internal scan chains' amount and their corresponding scan chain length are all discrete integers, so the real number (integer) coding scheme is adopted [5]. For example, d695.soc in ITC02 benchmark [18] has 16 IP cores (s); where IP core 6 has 16 internal scan chains, namely, $\{S_1, S_2, \dots, S_{16}\}$; their length are 41, 41, 40, 40, 40, 40, 40, 40, 40, 40, 40, 39, 39, 39, 39, respectively. Since d depends on internal scan chains' amount in IP module, 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 chain or the second wrapper 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.

5.2. System Initialization

The largest emigration rate E is set as 1 and the largest immigration rate I is set as 1. We set population size NP as 50; the maximum generation' amount $MaxGen$ could be set as 100; maximum mutation rate is set as 0.005; the maximum generation jumping rate Jo is set as 0.7

5.3. Opposition-based Population Initialization

Since we have little priori knowledge about the initial solutions, it is popular for us to build the beginning population through random operation.

1) The stochastic beginning solutions are generated as follows.

$$G_{0,ij} = \text{round}(c + (d-c).\text{rand}),$$

$$i = 1, 2, \dots, NP; \quad j = 1, 2, \dots, D \quad (9)$$

Where $\text{round}(x)$ should be a function that rounds the elements of x to the nearest integers; c is the minimum value of each SIV, here let c be 1; d is the maximum value of each SIV, here let d be w , which is determined by wrapper scan chains' amount.

2) According to such random beginning population, opposite population can be calculated by

$$OG_{0,ij} = c + d - G_{0,ij}, \quad i = 1, 2, \dots, NP;$$

$$j = 1, 2, \dots, D \quad (10)$$

Where $G_{0,ij}$ and $OG_{0,ij}$ indicate the j -th feature (SIV) of the i -th island (solution or individual) of the beginning population and the beginning opposite population, respectively; the rest parameters are the same as equation (9).

3) According to the cost function value of each individual, choose NP competent individuals within such set $\{G_0, OG_0\}$ to be final beginning population G_0 .

5.4. Cost Function

In order to evaluate every individual, we can define a cost function, whose value is inversely proportional to the fitness of the corresponding individual. In other words, the greater the cost function value, the less the fitness and the worse the individual.

$$cf(G_{gk}) = \sum_{i=1}^{i=w} (L(D_i) - \frac{1}{n} \sum_{j=1}^{j=n} (L(Sc_j)))^2, \quad g = 0, 1, \dots, MaxGen;$$

$$k = 1, 2, \dots, NP \quad (11)$$

Where $L(D_i)$ is 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; G_{gk} is the k -th individual of the g -th generation.

5.5. Opposition-based Generation Jumping

Through employing the opposition-based generation jumping for the present population, the procedure of evolution maybe be forced to form a fresh candidate individual that may be better compared with the present individual. After new population is generated by migration and mutation operation, according to the jumping possibility Jo , we can obtain the opposite population by equation (12).

$$OG_{g,ij} = c + d - G_{g,ij}, \quad g = 1, 2, \dots, MaxGen; \quad i = 1, 2, \dots, NP;$$

$$j = 1, 2, \dots, D \quad (12)$$

Table 1. The Experimental Results of p34392 module 2.

w	The Longest Wrapper Scan Chain			
	OBBO	BFD	MAV	MAVR
2	4534	4538	4536	4536
3	2952	2954	2954	2953
4	2269	2269	2269	2269
5	1772	1773	1772	1786
6	1701	1701	1701	1701
7	1699	1700	1699	1699
8	1135	1135	1135	1135
9	1134	1134	1134	1134
10	1134	1134	1134	1134
11	1134	1134	1134	1134
12	1134	1134	1134	1134
13	1133	1133	1133	1133
14	1133	1132	1132	1132
15	611	611	611	611
16	570	570	570	570
17	570	570	570	570
18	570	570	570	570

Then, NP competent solutions are chosen within the unified set which is composed of the present population and corresponding opposite population. Such generation jumping is similar to opposition-based initialization. Unlike the former one, the opposite population is generated through generation jumping operation in dynamical case according to a jumping possibility J_0 .

6. EXPERIMENTAL VERIFICATION

To compare the algorithms for wrapper design, we use ITC'02 benchmarks [18]. While most of internal scan chains in IP module are balanced, so it is hard to determine which algorithm is better in these IP modules. To prove the effectiveness of the proposed algorithm, we select an unbalanced IP module.

In Table 1, w is the wrapper scan chains' amount (TAM width); the second column is the results of OBBO algorithm; the third column is the results of BFD algorithm; the fourth column is the results of MAV algorithm; the last column is the results of MAVR algorithm. From Table 1, we can conclude that the new method can shorten the longest wrapper scan chain when w is less than 7; the longest wrapper scan chain cannot be shortened any more for all the algorithms, if w is more than 15.

The case of Table 2 is similar to that of Table 1. From Table 2, we could obviously conclude that such proposed algorithm can shorten the longest wrapper scan chain when w is less than 8; the longest wrapper scan chain cannot be shortened any more for all the algorithms, if w is more than 10.

Through calculating the number of generation calls (NGC), we could compare BBO's convergence speed with OBBO's. Such NGC is similar to functional calls' amount (NFC) [6]. If we have a smaller NGC, it indicates that we obtain faster convergence speed. The stop condition is to obtain a lesser than value-to-reach (VTR), without outnumbering the maximum generation calls' amount MaxGen. To reduce the random character, the obtained NGC is the average value over 20 independent runs.

For the convenience of convergence speed's comparison, we could employ AR which is defined in (13), according to the NGC for BBO and OBBO.

$$AR = NGC_{BBO} / NGC_{OBBO} \quad (13)$$

If we obtain an AR which is larger than 1, it shows that OBBO must be faster. We could also define another variable SR as (14) to calculate the probability that the algorithm completely reach the VTR:

Table 2. The Experimental Results of p22810 module 5.

w	The Longest Wrapper Scan Chain			
	OBBO	BFD	MAV	MAVR
2	1128	1128	1133	1138
3	752	763	757	757
4	566	572	578	578
5	452	461	463	463
6	379	389	389	387
7	331	342	335	335
8	286	295	295	295
9	262	274	260	260
10	237	239	247	247
11	214	214	214	216
12	214	214	214	214
13	214	214	214	214
14	214	214	214	214
15	214	214	214	214
16	214	214	214	214
17	214	214	214	214
18	214	214	214	214

Table 3. The Experimental Results of Convergence speed of p34392 module 2.

w	BBO	OBBO	AR
	NGC SR	NGC SR	
2	56 0.8	41 0.9	1.37
3	51 1	33 1	1.55
4	60 0.9	45 1	1.33
5	69 1	46 0.95	1.50
6	68 1	51 1	1.33
7	67 1	63 1	1.06
8	62 1	55 1	1.13
9	66 1	50 1	1.32
10	73 1	59 1	1.24
11	77 1	73 1	1.05
12	67 1	56 1	1.20
13	70 1	55 1	1.27

Table 3. contd...

	BBO	OBBO	
w	NGC SR	NGC SR	AR
14	77 1	64 1	1.20
15	80 1	69 1	1.16
16	91 1	75 1	1.21
17	80 1	72 1	1.11
18	79 1	73 1	1.08

Table 4. The Experimental Results of Convergence speed of p22810 5.

	BBO	OBBO	
w	NGC SR	NGC SR	AR
2	38 1	35 1	1.09
3	58 1	45 1	1.29
4	65 1	55 1	1.18
5	49 1	41 1	1.19
6	37 1	40 1	0.93
7	43 1	31 1	1.39
8	36 1	34 1	1.06
9	44 1	38 1	1.16
10	48 1	41 1	1.17
11	50 1	42 1	1.19
12	49 1	41 1	1.19
13	46 1	32 1	1.44
14	51 1	42 1	1.21
15	68 1	46 1	1.48
16	73 1	59 1	1.24
17	85 1	64 1	1.33
18	71 1	63 1	1.13

$$SR = \text{NumberToVTR} / \text{TotalTrials} \quad (14)$$

Where NumberToVTR is the number of times got to VTR; TotalTrials is the total trials' amount (independent runs).

In Table 3, w is wrapper scan chains' amount; NGC is the generation calls' amount; SR is the success rate and AR is acceleration rate. From Table 3, since most of AR is greater than 1, we can come to a conclusion that the proposed algorithm is obviously faster in general.

The case in Table 4 is similar to that in Table 3. From Table 4, since most of AR is greater than 1, we can come to a conclusion that the proposed algorithm is obviously faster in general.

CONCLUSION

A new algorithm is proposed to reduce the longest wrapper scan chain's length, so as to make the test time of IP module be minimum. Experimental results show that OBBO

can obtain shorter longest wrapper scan chain in general and the convergence speed is faster, compared with BBO.

CURRENT & FUTURE DEVELOPMENTS

The paper has reviewed the recent patents on the scan chain for test of IP module, and presented a novel technique to shorten the longest wrapper scan chain, with the aim of reducing an IP module's test time. The technique is applied on the internal scan chain of IP module using the proposed algorithm OBBO 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 an IP module's test time. Such new method is a random optimization algorithm which combines BBO algorithm with OBL. By using migration operation, mutation operation and OBL operation, we achieve a balance between different wrapper scan chains in order to reduce the longest wrapper scan chain's length. Experimental results show that OBBO can obtain shorter longest wrapper scan chain (LWSC) in most case, at the same time, such convergence speed is faster. Furthermore, the proposed algorithm is simple and flexible enough to apply to wide fields, such as engineering optimization for NP hard problems in the real world. In the near future, we will apply the proposed technique on wrapper design for 3D stacked SoC.

CONFLICT OF INTEREST

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

ACKNOWLEDGEMENTS

This work was subsidized and supported by the National Natural Science Foundation of China (Grant No. 60766001), Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (Grant No. YQ14110), Science and Technology Research Project of Guangxi Department of Education (Grant No. 2013ZD024) and Guangxi Information Science Experiment Center Project (Grant No. 20130323).

REFERENCES

- [1] D. Simon, "Biogeography-based optimization", *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702-713, 2008.
- [2] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "Test Wrapper and test access mechanism co-optimization for system-on-chip", *J. Electron. Test. Theory Appl.*, vol. 18, no. 2, pp. 213-230, 2002.
- [3] Y. Yu, Y. F. Cheng, and PENG Y, "Wrapper scan chain balance algorithm based on mean-value allowance", *Chinese J. Sci. Instrum.*, vol. 32, no. 10, pp. 2290-2296, 2011.
- [4] D. H. Niu, H. Wang, S. Y. Yang, B. Cheng, and Y. Jin, "Re-optimization algorithm for SoC Wrapper-chain balance using mean-value approximation", *Tsinghua Sci. Technol.*, vol.12, no. suppl., pp. 61-661, 2007.
- [5] A. J. Zhu, Z. Li and C. P. Xu, "Wrapper scan chain design algorithm for SoC test based on biogeography optimization", *Chinese J. Sci. Instrum.*, vol. 33, no. 12, pp. 2774-2780, 2012.
- [6] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution", *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp.64-78, 2008.
- [7] X. T. Li, and M. H. Yin, "Hybrid differential evolution with biogeography-based optimization for design of a reconfigurable antenna array with discrete phase shifters", *Int. J. Antennas Propag.*, vol. 2011, pp. 1-12, 2011.
- [8] L. Wang, and Y Xu, "An effective hybrid biogeography-based optimization algorithm for parameter estimation of chaotic systems", *Expert. Syst. Appl.*, vol. 38, no. 12, pp. 15103-15109, 2011.
- [9] X. T. Li, J. Y. Wang, J. P. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization", *Appl. Math. Comput.*, vol. 218, no. 2, pp. 598-609, 2011.
- [10] W.D. Lee, "Input, output, and link instruction circuits for hierarchical P1500 wrappers", U. S. Patent 8,667,351, March 4, 2014.
- [11] W. D. Lee, "Core circuit test architecture", U. S. Patent 8,656,237, February 18, 2014.
- [12] C. Wu-Tung, S. Manish, D. Avijit, B.R. Brady, and M.A. Kassab, "Test access mechanism for diagnosis based on partitioning scan chains", U. S. Patent 8,607,107, Dec. 10, 2013.
- [13] R. Kapur, J. Saikia, R. Uppuluri, P. Notiyath, T. Fernandes, S. Kulkarni and A. Anbalan, "Test design optimizer for configurable scan architectures", U. S. Patent 8,584,073, November 12, 2013.
- [14] J. Dirks, M. Dinter, 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] A. Rich, and R. John, "Scan chain fault diagnosis", U. S. Patent 8,572,543, Oct. 29, 2013.
- [16] C. R. Conklin, M. F. Fee, A. Martens, W. Niklaus, S. B. Swaney, and T. Weibel, "Updating settings of a processor core concurrently to the operation of a multi core processor system", U. S. Patent 8,499,144, July 30, 2013.
- [17] H. Kukreja, and D. Agrawal, "Method for partitioning scan chain", U. S. Patent 8,504,886, August 6, 2013.
- [18] E.J. Marinissen, V. Iyengar, and K. Chakrabarty, "A set of benchmarks for modular testing of SOCs", *Int. Test Conf.*, pp. 519-528, 2002.
- [19] Y. Huang, W-T Cheng, C-C Tsai, N. Mukherjee, O. Samman, Y. Zaidan, and S. M. Reddy, "On concurrent test of core-based soc design", *J. Electron. Test. Theory Appl.*, vol. 18, no. 4-5, pp. 401-414, August/October, 2002.
- [20] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. "Efficient test access mechanism optimization for system-on-chip", *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 5, pp. 635-643, May 2003.
- [21] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test-access mechanism optimization for core-based three-dimensional SOCs", *Microelectron. J.*, pp. 601-615, Jan. 2010.
- [22] D.L. Lewis, S. Panth, X. Zhao, S.K. Lim, and H-H. S. Lee, "Designing 3D Test Wrappers for Pre-bond and Post-bond Test of 3D Embedded Cores", *Pr. IEEE Comp. Design*, pp. 90-95, 2011.
- [23] Q. Xu, and N. Nicolici. "Modular SOC testing with reduced wrapper count", *IEEE Trans. Computer-Aided Design*, vol. 24, pp. 1894-1908, Dec. 2005.

Received: June 09, 2014

Revised: June 22, 2014

Accepted: July 24, 2014

© Zhu et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.