

Design of Test Wrapper Scan Chain Based on Differential Evolution

Aijun Zhu¹, Zhi Li^{1,2,3,*}, Wangchun Zhu² and Chuanpei Xu²

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

³Guilin University of Aerospace Technology, Guilin 541004, China.

Received 25 June 2013; Accepted 15 August 2013

Abstract

Integrated Circuit has entered the era of design of the IP-based SoC (System on Chip), which makes the IP core reuse become a key issue. SoC test wrapper design for scan chain is a NP Hard problem, we propose an algorithm based on Differential Evolution (DE) to design wrapper scan chain. Through group's mutation, crossover and selection operations, the design of test wrapper scan chain is achieved. Experimental verification is carried out according to the international standard benchmark ITC'02. The results show that the algorithm can obtain shorter longest wrapper scan chains, compared with other algorithms.

Keywords: SoC, Differential Evolution, Wrapper Scan Chain.

1. Introduction

Because of the fast development of modern semiconductor technology, the integration of integrated circuit is continuously improved. In this context, the design of SoC enters a fast development track. To reduce the time-to-market, the design of IP-based core is popular nowadays. However, it brings us new problems and challenges, such as the increasing test cost and test time. All these problems become the bottleneck, which impedes development of SoC [1, 5, 12-16]. The main motivation of this paper is to optimize the wrapper scan chains so as to reduce the length of the longest scan chain of IP core. Because the test time of IP module is determined by the length of the longest wrapper scan chain of IP core, by reducing the length of the longest wrapper scan chain of IP module, the test time of IP module is reduced and SoC test costs are minimized.

IYENGAR V, et al [2] proposed a classic algorithm called Best Fit Decreasing (BFD) to design wrapper scan chain. Although BFD algorithm is fast, simple and widely used, it only possesses local optimization capabilities [4]. To improve the BFD algorithm, NIU D H, et al [4] proposed a re-optimization algorithm for SoC wrapper chain balance using Mean-Value Approximation (MVA), which makes use of the average value of the whole internal scan chains to guide the global optimization. However, there are some drawbacks in the MVA algorithm, i.e. it does not always give priority to current longest internal scan chain. In order to improve MVA algorithm, YU Y, et al [3] proposed 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 on the basis of the average value to guide global optimization. However, the MVAR algorithm also has its drawback: it is difficult to select an appropriate residue, for example, if the residue is selected as more than five percent of the mean value, the MVAR algorithm is no better than the MVA algorithm. If the residue is selected as less than one percent of the mean value, there is no apparent difference between the MVAR algorithm and the MVA algorithm.

In a word, all these algorithms aim at shortening the longest wrapper scan chain to reduce test time of the IP core. Because design for test wrapper scan chain is NP hard problem [2], this paper proposes an algorithm based on differential evolution to solve this problem. Differential evolutionary optimization method is different from traditional optimization methods; it belongs to heuristic optimization method. In the traditional optimization methods, additional information such as the gradient of the objective function are required to search deterministically optimal solution; while most of heuristic optimization methods simulate natural optimization mechanism, which are also known as natural computation or evolutionary computation. With a strong ability of global optimization, differential evolution optimization has become a hot area [6-10].

In this paper, differential evolution algorithm is adopted to design wrapper scan chain. The typical SoC IP cores are verified by the experiment and experimental results show the superiority of the DE algorithm.

Organization of the rest of this paper is as follows. Problem formulation is introduced briefly in Section 2. The differential evolution algorithm is presented in Section 3. The proposed algorithm is introduced in detail in Section 4. The proposed algorithm is evaluated by a numerical simulation on ITC02 benchmarks, and experimental results

* E-mail address: zbluebird@quet.edu.cn, cclizhi@quet.edu.cn
ISSN: 1791-2377 © 2013 Kavala Institute of Technology. All rights reserved.

are presented in Section 5. Finally, concluding remarks is given in Section 6.

2. Problem Formulation

When IP core vendors sell their IP core, they also provide some relative information about IP core to help the SoC integration developers. Suppose there is an IP core C_i and its relative information is consist of the number of functional input port, the number of functional output port, the number of functional bidirectional port, the number of internal scan chains and their length. 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}, $j \in [1, N\text{Scan}_i]$ }}, where NumIn_i is the number of functional input port, NumOut_i is the number of functional output port, NumIO_i is the number of functional bidirectional port, NumScan_i is the number of internal scan chains, and {LenScan_{ij}, $j \in [1, \text{NumScan}_i]$ } is the set of the length of all the internal scan chains. When the SoC integration developers design the test wrapper, they combine IP core test ports with internal scan chains in series into a certain number of wrapper scan chains, which are used to load the test vectors and collect test response.

The mathematical model [5] of wrapper design is described as follows: Given an input port set $IN = \{IN_1, IN_2, \dots, IN_i, \dots, IN_m\}$, each input cell corresponds to an input port, and its length $L(IN_i) = 1, i \in [1, m]$. Given an output port set $OU = \{OU_1, OU_2, \dots, OU_j, \dots, OU_g\}$, each output cell corresponds to an output port, and its length $L(OU_j) = 1, j \in [1, g]$. Given an bidirectional port set $BIO = \{BIO_1, BIO_2, \dots, BIO_k, \dots, BIO_u\}$, each bidirectional cell corresponds to an bidirectional port, and its length $L(BIO_k) = 1, k \in [1, u]$. Given a set of n internal scan chains of the IP core $Sc = \{Sc_1, Sc_2, \dots, Sc_v, \dots, Sc_n\}$, and the length of each internal scan chain equal $L(Sc_v), v \in [1, n]$.

Let A be a given subset, $A \subseteq (IN \cup BIO \cup Sc)$. Let $L(A)$ be a sum of length of every element in subset A , $L(A) = \sum_{a \in A} L(a)$. We can partition $(IN \cup BIO \cup Sc)$ into w wrapper scan-in chains, namely, $E = \{E_1, E_2, \dots, E_x, \dots, E_w\}$, $\forall E_x, E_x \subseteq (IN \cup BIO \cup Sc), x \in [1, w]$. We define $Si(E) = \max_{1 \leq x \leq w} L(E_x)$ as the longest wrapper scan-in chain.

Let B be a given subset, $B \subseteq (OU \cup BIO \cup Sc)$. Let $L(B)$ be a sum of length of every element in subset B , $L(B) = \sum_{b \in B} L(b)$. We can partition $(OU \cup BIO \cup Sc)$ into w wrapper scan-out chains, namely, $F = \{F_1, F_2, \dots, F_y, \dots, F_w\}$, $\forall F_y, F_y \subseteq (OU \cup BIO \cup Sc), y \in [1, w]$. We define $So(F) = \max_{1 \leq y \leq w} L(F_y)$ as the longest wrapper scan-out chain.

In BFD algorithm [2], first partition the internal scan chains among w wrapper scan chains to minimize the longest wrapper scan chain 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. Differential Evolution Algorithm

Differential evolution algorithm was proposed by R.Storn in 1997. Because differential evolution algorithm is concise and of less control parameters, it has been widely used in various fields [6, 7, 8, 9, 10].

Differential Evolution algorithm is similar to genetic algorithm, and it also includes crossover, mutation and selection. Standard genetic algorithm adopts roulette selection strategy generally, but differential evolution algorithm uses tournament selection strategy; For the crossover operation, the differential evolution algorithm and genetic algorithm is similar in general, but for the mutation operation, the differential evolution algorithm uses a completely different strategies: it makes use of the difference vector between two different individuals and let the difference vector be scaled, enabling the individual variation of the disturbance, which can compensate for the variation operation of genetic algorithm.

In general, the optimization problem is divided into two kinds: maximization and minimization problems. However, maximization problem can be transformed into minimization problem. Without loss of generality, we only consider the minimization problem:

$$\begin{aligned} \min f(x_1, x_2, x_3, \dots, x_d) \\ s.t \quad x_i^{low} \leq x_i \leq x_i^{up}, \quad i = 1, 2, \dots, d \end{aligned} \quad (1)$$

Where d is the number of the dimension of the solution; x_i^{low} is the upper bound of the i -th component of candidate solution; x_i^{up} is the lower bound of the i -th component of candidate solution.

In Differential evolution algorithm, an initialized population is generally generated randomly, then through mutation, crossover and selection operations, the next generation population is obtained; the evolution does not stop until the maximum number of iterations; at the end of the evolution, the optimal solution is obtained, each part is introduced in detail as follows.

3.1 Generation of Initial Population

Generally speaking, initial population is generated randomly in evolutionary computation algorithm, and there is no exception for differential evolution.

Let R be the initial population, which can be defined as below.

$$R = \{X^1, X^2, \dots, X^k, \dots, X^{NP}\} \quad (2)$$

Where NP is the population size; k is the serial number of individuals, and $k=1, 2, 3, \dots, NP$. Each individual can be expressed as:

$$X^k = (X_1^k, X_2^k, \dots, X_p^k, \dots, X_d^k) \quad (3)$$

Where d is the number of the dimension of the solution (individual); $p=1, 2, \dots, d$; $k=1, 2, \dots, NP$.

$$X_p^k = X_p^k(low) + (X_p^k(up) - X_p^k(low)) \times rand(0,1) \quad (4)$$

Where $X_p^k(low)$ is the lower bound of the p-th component of the k-th individual; $X_p^k(up)$ is the upper bound of the p-th component of the k-th individual; $p=1, 2, \dots, d$; $k=1, 2, \dots, NP$; $rand(0,1)$ represents a random number between 0 and 1.

3.2 Mutation Operation

In most evolutionary algorithms, to achieve the mutation operation, an individual in a viable domain is produced randomly; while in differential evolution algorithm a differential strategy is adopted to generate the variation of an individual. In the standard differential strategy, three individuals that are not the same are randomly selected, and the difference vector of two individuals is zoomed and then it is synthesized with the third individual to achieve the mutation operation.

$$V^i(g+1) = X^{r_1}(g) + F \cdot (X^{r_2}(g) - X^{r_3}(g)),$$

$$r_1 \neq r_2 \neq r_3 \neq i \quad (5)$$

Where g is the generation number; F is the scaling factor; $g=0, 1, 2, \dots, MaxGen$; MaxGen is the maximum of iteration generation.

The above differential strategy is the famous strategy DE/rand/1/bin, which is the most widely used; because it can remain diversity of population. In addition, there is another differential strategy which is widely used, that is DE/best/2/bin. Because its faster convergence speed is an advantage, the DE/best/2/bin differential strategy is as follows:

$$V^i(g+1) = X^b(g) + F \cdot (X^{r_2}(g) + X^{r_3}(g) - X^{r_4}(g) - X^{r_5}(g)), \quad (6)$$

Where $r_4 \neq r_2 \neq r_3 \neq r_5 \neq i$; $X^b(g)$ represents the best individual in current population; In addition to the above two differential strategies, there are other strategies, such as: DE/rand/2/bin, DE/best/1/bin, DE/rand-to-best/bin, DE/current-to-best/bin and DE/current-to-rand/bin.

3.3 Crossover Operation

The g-th generation $\{X^1(g), X^2(g), \dots, X^k(g), \dots, X^{NP}(g)\}$ and its variant are crossed as follows:

$$U_j^k(g+1) = \begin{cases} V_j^k(g+1), \\ \text{if}(rand(0,1) \leq CR \parallel j = j_{rand}) \\ X_j^k(g), \\ \text{otherwise} \end{cases} \quad (7)$$

Where CR represents the crossover probability; j_{rand} is a random integer between 1 and d.

3.4 Select Operation

Differential evolution algorithm uses greedy strategy to select individuals for the next generation. That is to say, if the individual, which is obtained by crossover operation, is better than the original one, then select the new individual as the one in new generation, or the individual maintain unchanged.

$$X^k(g+1) = \begin{cases} U^k(g+1), \\ \text{if}(f(U^k(g+1)) \leq f(X^k(g+1))) \\ X^k(g), \\ \text{otherwise} \end{cases} \quad (8)$$

3.5 Operation for Infeasible solutions

In the differential evolution process, in order to ensure the effectiveness of solutions, each individual must be judged whether each gene is within the specified range. If the current individual is not in feasible region, then a similar method to generate the initial population of individuals is used. Namely, a new individual is generated randomly to replace the one which is not in feasible region.

4. Proposed Algorithm

The proposed algorithm based on differential evolution for wrapper scan chain design 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 scaling factor F; set the value of the crossover probability Pc; set the value of population size NP; set the value of the maximum iterative generation MaxG.

Step (2): in the feasible region, generate a random initial parent population whose size is NP, then use the cost function formula (10) to calculate the cost function value of each individual in the initial parent population.

Step (3): according to the formula (5), generate the mutant population, whose size is NP.

Step (4): check the boundary of each gene of each individual in mutant population; if the value of a gene is less than 1, then change its value to 1, if it is greater than w, then change its value to w.

Step (5): for each gene of the NP individuals, generates a random number rand between 0 and 1, if rand is greater than Pc, the value of a gene of current individual in the child population is set as that of the gene of the corresponding individual in the parent population; otherwise, the value of the gene of current individual in the child population is set as that of the gene of the corresponding individual in the mutant population.

Step (6): use the cost function formula (10) to calculate the cost function value of each individual in the child population.

Step (7): for the NP individuals, if the cost function value of the current individual in the child population is less than that of corresponding individual in the parent population, use the individual in child population to replace the corresponding individual in parent population.

Step (8): through the Step (7), a new generation of population, the parent population, is obtained, and then calculate the cost function value of NP individuals in the parent population.

Step (9): judge whether the number of iterative generation reach MaxG or not; if not, turn Step (3).

Step (10): according to the cost function value, sort the NP individuals in the parent population in ascending order, and then obtain the individual whose cost function is minimum. The individual, whose cost function is minimum, is the optimal solution.

4.1 Solution Encoding

There are two encoding schemes for Evolutionary computation algorithm: 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, 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 (individual): every individual is a candidate solution $X_i=(X_{i1}, X_{i2}, \dots, X_{id})$, where $i=1,2,\dots, NP$; d is the number of dimension of solution; NP is the population size. If there are n internal scan chains in IP core, d equal n ; for arbitrary X_{ij} , X_{ij} is an integer between 1 and w . where w is the number of wrapper scan chains; $i=1,2,\dots, NP$; $j=1,2,\dots, d$. For example, d695.soc in ITC02 benchmark [11] has 16 IP cores (modules); 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 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 $X = (1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2)$ with 16 dimensions, which means odd-numbered series of internal scan chains are connected together to form the first wrapper scan chain and even-numbered series of internal scan chains are connected together to form the second wrapper scan chain.

4.2 System Initialization

The population size NP is set as 20; the maximum number of generation $MaxG$ is set as 1500; the scaling factor F is set as 0.5; the crossover probability P_c is set as 0.2; 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 number of the dimension of the solution space d .

4.3 Population Initialization

The random initial population is generated as follows.

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

$$i = 1, 2, \dots, NP; j = 1, 2, \dots, d \tag{9}$$

Where $\text{round}(x)$ is a function that rounds the elements of x to the nearest integers; a is the minimum value of each gene, here let a be 1; b is the maximum value of each gene, here let b be w , which is determined by the number of wrapper scan chains; d is the number of the dimension of a solution (individual); NP is the population size.

4.4 Cost Function

In order to evaluate every individual, we should 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 (solution).

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

$$g = 0, 1, \dots, MaxG; k = 1, 2, \dots, NP \tag{10}$$

Where $L(D_i)$ is a sum of length of every internal scan chain in the i -th wrapper scan chain; $L(Sc_j)$ is the length of the j -th internal scan chain; n is the number of internal scan chains; w is the number of wrapper scan chains; G_{gk} is the k -th individual of the g -th generation; NP is the population size.

5. Experimental Verification

To compare the various algorithms for wrapper design, we use ITC'02 benchmarks [11]. Because most of internal scan chains in IP Module have little difference in length, so it is difficult to determine which algorithm is better in such IP Modules. To prove the effectiveness of the proposed algorithm, we select two typical unbalanced IP Module (p34392 IP Module 2 and p22810 IP Module 5,) in ITC'02 benchmarks [11].

Table 1. Results of p34392 Module 2

w	The longest wrapper scan chain			
	DE	BFD	MAV	MAVR
2	4534	4538	4536	4536
3	2953	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	1132	1132	1132	1132
15	611	611	611	611
16	570	570	570	570
17	570	570	570	570
18	570	570	570	570

In table 1, w is the number of wrapper scan chains (TAM width); the second column is the results of DE algorithm; the third column is the results of BFD algorithm; the forth column is the results of MAV algorithm; the last column is the results of MAVR algorithm. From table 1, we can conclude that the proposed algorithm can shorten the longest wrapper scan chain when w is less than or equal 7; the longest wrapper scan chain cannot be shortened any more for all the algorithms, if w is more than 8.

Table 2. Results of p22810 Module 5

w	The longest wrapper scan chain			
	DE	BFD	MAV	MAVR
2	1128	1128	1133	1138
3	754	763	757	757
4	567	572	578	578
5	460	461	463	463

w	The longest wrapper scan chain			
	DE	BFD	MAV	MAVR
2	1128	1128	1133	1138
6	387	389	389	387
7	334	342	335	335
8	294	295	295	295
9	266	274	260	260
10	246	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

In table 2, w and the other parameters are the same as those in table 1. From table 2, we can conclude that the proposed algorithm can shorten the longest wrapper scan chain when w is less than or equal 10; the longest wrapper

scan chain cannot be shortened any more for all the algorithms, if w is more than 12.

6. Conclusion

An algorithm based on differential evolution is proposed to shorten the longest wrapper scan chain, so as to minimize the test time of IP module. Experimental results show that proposed algorithm can obtain shorter longest wrapper scan chain in general.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 60766001) and Guangxi Natural Science Foundation of China (Grant No. 2013GXNSFAA019332).

References

- Chakrabarty K, "Optimal Test Access Architectures for System on-a-Chip," ACM Transactions on Design Automation of Electronic Systems, vol. 6, no. 1, 2001, pp. 26-49.
- IYENGAR V, CHAKRABARTY K, MARINISSEN E J, "Test Wrapper and test access mechanism co-optimization for system-on-chip", Journal of Electronic testing: Theory and Application, vol. 18, no. 2, 2002, pp. 213-230.
- YU Y, CHENG Y F, PENG Y, "Wrapper scan chain balance algorithm based on mean-value allowance", Chinese Journal of Scientific Instrument", vol. 32, no. 10, 2011, pp. 2290-2296.
- NIU D H, WANG H, YANG S Y, et al, "Re-optimization algorithm for SoC Wrapper-chain balance using mean-value approximation", Tsinghua Science and Technology, vol.12, no.suppl. 1, 2007, pp. 61-66.
- ZHU A J, LI Z, XU C P, "Wrapper scan chain design algorithm for SoC test based on biogeography optimization", Chinese Journal of Scientific Instrument, vol. 33, no. 12, 2012, pp. 2774-2780.
- Shahryar Rahnamayan, Hamid R. Tizhoosh, "Opposition-Based Differential Evolution", IEEE Trans Evol Comput, vol. 12, no. 1, 2008, pp.64-78.
- Storn R, "System design by constraint adaptation and differential evolution", IEEE Trans Evol Comput, vol. 3, no. 1, 1999, pp.22-34.
- Sharma, Harish, "Fitness based Differential Evolution", Memetic Computing, vol.4, no.4, 2012, pp.303 - 316.
- Carbajal-Gómez, V.H., "Optimizing the positive Lyapunov exponent in multi-scroll chaotic oscillators with differential evolution algorithm", Applied Mathematics and Computation, vol.219, no.15, 2013, pp.8163 - 8168.
- Subudhi, Bidyadhar, Jena, Debashisha, "A differential evolution based neural network approach to nonlinear system identification", Applied Soft Computing Journal, vol.11, no.1, 2011, pp.861 - 871.
- MARINISSEN E J, IYENGAR V, CHAKRABARTY K, "A set of benchmarks for modular testing of SOCs", International Test Conference, 2002, pp. 519-528.
- V. Iyengar, K. Chakrabarty, and E.J. Marinissen. "Efficient test access mechanism optimization for system-on-chip", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.5, May 2003, pp. 635-643.
- Xiaoxia Wu, YiboChen, Krishnendu Chakrabarty, YuanXie. "Test-access mechanism optimization for core-based three-dimensional SOCs", Microelectronics Journal, January 2010, pp. 601-615.
- Dean L. Lewis Shreepad Panth Xin Zhao Sung Kyu Lim Hsien-Hsin S. Lee. "Designing 3D Test Wrappers for Pre-bond and Post-bond Test of 3D Embedded Cores", IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2011, pp. 90-95.
- Qiang Xu, Nicola Nicolici. "Modular SOC Testing With Reduced Wrapper Count", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.24, December 2005, pp. 1894-1908.
- Jaehoon Song, "An Efficient SoC Test Technique by Reusing On/Off-Chip Bus Bridge", IEEE Trans circuits and systems, vol. 56, no. 3, 2009, pp.554-565.