International Journal of u- and e- Service, Science and Technology
Vol.9, No. 6 (2016), pp.377-388
http://dx.doi.org/10.14257/ijunesst.2016.9.6.34

Artificial Bee Colony and Neural Networks Optimization Test Generation Algorithm for Multiple Faults of Digital Circuits

Zhao Ying[1], Liu Bai-sheng[2] and Li Yan-juan [3]

[1,2]*Electrical & information Engineering College, Beihua University, Jilin 132021, China*
[3] *College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040,China*
[1]*zhaoying_jl@126.com,* [2]*297744@qq.com,* [3] *aaa123@yahoo.cn*

Abstract

A multiple stuck-at faults test generation algorithm based on artificial bee colony and neural networks for circuits is proposed in this paper because the test generation efficiency for multiple stuck-at faults in digital circuits is low. The algorithm converts multiple stuck-at faults to single stuck-at fault and constructs the constraint circuit of the single stuck-at fault circuit using Hopfield neural networks. The test vectors for multiple stuck-at faults in the original circuit can be obtained by solving the zero value of energy function of the constraint network's interface circuit with artificial bee colony optimization algorithm. The experimental results on ISCAS'85 international standard test circuits show the superiority of the algorithm.

Keywords*: artificial bee colony; neural networks; constraint circuit; energy function*

1. Introduction

The rapid development of microelectronics technology makes the integration and complexity of digital integrated circuit higher and higher, which makes digital circuit test generation more and more difficult. Recently, many local and foreign scholars carried out a wide range of research and made some achievements, but the research has the disadvantage of long test generation time and low fault coverage, and most of the research models are single stuck-at faults and the test generation is very few [1]. In order to guarantee the reliability of a digital circuit, it is usually necessary to have full test for some components, so a test generation for multiple stuck-at faults of the digital circuit has a very important significance. The multiple stuck-at faults are converted to equivalent single stuck-at fault in this paper, so the test generation of multiple stuck-at faults is transformed to the test generation of single stuck-at fault. This paper adopts Hopfield neural network [2] model in the single stuck-at fault test generation, and constructs the constraint circuit of the single stuck-at fault circuit, finally the test vectors for multiple stuck-at faults circuit can be obtained by applying artificial colony algorithm to solve the zero value of energy function of the constraint circuit's interface circuit. Experimental results on ISCAS'85 circuits show that the algorithm can quickly obtain multiple stuck-at faults' test generation vector, comparing with other algorithm; the test generation efficiency is improved obviously.

Corresponding author: LIU Bai-sheng

ISSN: 2005-4246 IJUNESST

2. The Equivalent Transformation of Multiple Stuck-at Faults and Single Stuck-at Fault

Figure 1(a) has V_1, V_2, V_3, V_4 inputs lines, and the respective outputs lines V_5, V_6, V_7, V_8. The two stuck-at-0(s-a-0) faults are on V_1, V_2 and the two stuck-at-1(s-a-1) faults are on V_3, V_4.In order to convert the multiple stuck-at faults to single stuck-at fault, two additional gates should be inserted [3], one is on-line gate and the other is fault gate.

A. On-line gates: A two-input gate should be inserted in each faulty line. An OR (AND) gate is inserted in the line that is stuck-at- 1 (0) fault.

B. Fault gate: It is an n input AND gate and feeds the on-line gates directly if the on-line gate is OR gate, or feeds the on-line gates though an inverter if the on-line gate is AND gate. The fault gate's inputs are derived directly from all s-a-1faults lines and through inverters from all s-a-0 faults lines. The multiple stuck-at faults is equivalent to the single stuck-at-1 fault at the output of the fault gate [4-6], so the multiple stuck-at faults are converted to equivalent single stuck-at fault. The stuck-at-1 fault at f in Figure 1(b) is equivalent to the multiple stuck-at faults in Figure 1(a).

Let's prove the correction of the method produced by the above.

(1) Circuit function's equivalence. For Figure 1(b),the functions are as follows.

$$V_5 = V_1 \cdot \overline{(V_1 V_2 V_3 V_4)} = V_1 \cdot (V_1 + V_2 + \overline{V_3} + \overline{V_4}) = V_1$$

$$V_6 = V_2 \overline{(V_1 V_2 V_3 V_4)} = V_2 \cdot (V_1 + V_2 + \overline{V_3} + \overline{V_4}) = V_2$$

$$V_7 = V_3 + \overline{V_1 V_2 V_3 V_4} = V_3$$

$$V_8 = V_4 + \overline{V_1 V_2 V_3 V_4} = V_4$$

The function is identical to the function of Figure 1(a).

(2) Circuit fault's equivalence. For Figure 1(a):

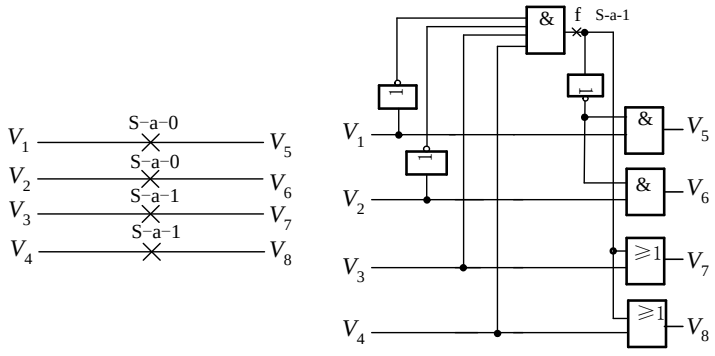
$$V_5 = 0;\ V_6 = 0;\ V_7 = 1;\ V_8 = 1$$

For Figure 1(b):

$$V_5 = 0 \cdot V_1 = 0; V_6 = 0 \cdot V_2 = 0$$

$$V_7 = 1 + V_3 = 1; V_8 = 1 + V_4 = 1$$

The multiple stuck-at faults of Figure 1(a) is identical to the single stick-at fault of Figure 1(b).



(a) The Circuit of Multiple Stuck-at Faults (b) The Circuit After Changing

Figure1. Convert Multiple Faults into Single Fault

3. Neural Network Model for Single Stuck-at Fault

Hopfield two value neural network is applied to construct the model for single stuck-at fault. Hopfield two value neural network's energy function is defined by the formula:

$$E = -1/2\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i + K \quad (1)$$

Here V_i and V_j are state values(0 or 1) of neuron i and j, N is the number of neurons , I_i is threshold value of neuron i,T_{ij} is the weight value between neurons i and j, K is a constant. T_{ij}=T_{ji} and T_{ii}= 0.

Hopfield neural network's model parameters of basic gate circuits are shown in Table1.[7-9] A and B are constants that are greater than zero.

Table 1. Hopfield Neural Network's Model Parameters of Basic Gate Circuits

Gate circuit	input	output	weight matrix	threshold value	K
AND	V_1,V_2	V_3	$\begin{bmatrix} 0 & -B & A+B \\ -B & 0 & A+B \\ A+B & A+B & 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -2A-B \end{bmatrix}$	0
OR	V_1,V_2	V_3	$\begin{bmatrix} 0 & B & -(A+B) \\ B & 0 & -(A+B) \\ -(A+B) & -(A+B) & 0 \end{bmatrix}$	$\begin{bmatrix} A \\ A \\ B \end{bmatrix}$	0
NAND	V_1,V_2	V_3	$\begin{bmatrix} 0 & B & A+B \\ B & 0 & A+B \\ A+B & A+B & 0 \end{bmatrix}$	$\begin{bmatrix} A+B \\ A+B \\ 2A+B \end{bmatrix}$	$2A+B$
NOR	V_1,V_2	V_3	$\begin{bmatrix} 0 & -B & -(A+B) \\ -B & 0 & -(A+B) \\ -(A+B) & -(A+B) & 0 \end{bmatrix}$	$\begin{bmatrix} B \\ B \\ B \end{bmatrix}$	B

Table 2 shows Hopfield neural network's energy functions of basic gate.

Table 2. The Energy Function of Several Basic Gates Circuits

Gate circuit	input	output	**energy function**
AND	V_1, V_2	V_3	$E = -4V_3(V_1+V_2)+2V_1V_2+6V_3$
OR	V_1, V_2	V_3	$E = -4V_3(V_1+V_2)+2V_1V_2$ $+2V_1+2V_2+2V_3$
NAND	V_1, V_2	V_3	$E = 4V_3(V_1+V_2)+2V_1V_2$ $-4V_1-4V_2-6V_3+6$
NOR	V_1, V_2	V_3	$E = 4V_3(V_1+V_2)+2V_1V_2$ $-2V_1-2V_2-2V_3+2$

The digital circuits are composed of basic gates circuits, so neural network model of digital circuits can be obtained by merging neurons of basic gates circuits and adding neurons' state values and weight values[10]. The energy function of digital circuits is the sum of energy function of basic gates circuits. For example, Figure 2 is a simple digital circuit, the neural network model of AND gate and NOR gate are shown in (a-c) of Figure 3. The neural network model of Figure 2 is shown in (d) of Figure 3 by using the method, so the energy function of neural network for Figure 2 is following:

$$E = 4V_4(V_2 + V_3) + 4V_5(V_2 + V_4) + 4V_6(V_1 + V_5) + 2V_2V_3 + 2V_2V_4 + 2V_1V_5 - 2V_1 + 6V_4 + 4V_5 - 2V_6 \quad (2)$$

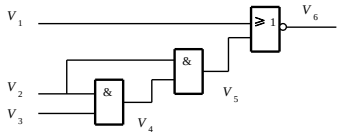


Figure2. A Simple Digital Circuit

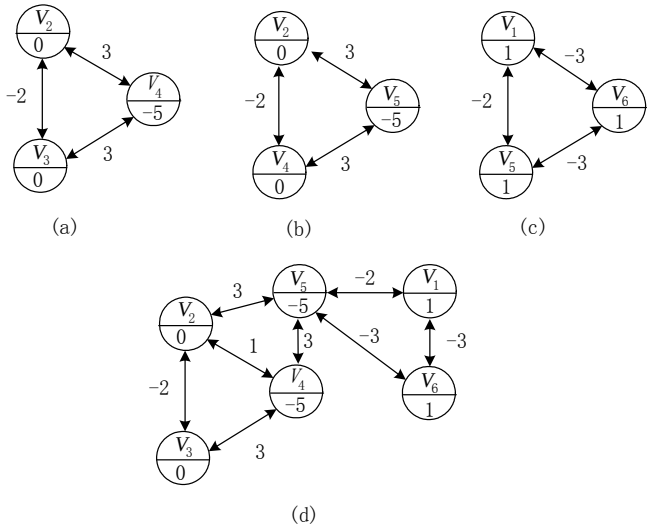


Figure 3. Hopfield Neural Networks Model of Figure2

The states are called consistent states when the states satisfy the circuit's function; other states are called inconsistent states when they can't satisfy the circuit's function [11]. The energy function's value is zero if the neuron states are consistent with the function of the circuit and energy function's value is greater than zero for all inconsistent states.

The constraint circuit should be constructed in order to make the circuit to be in consistent states. The single-output circuit's constraint circuit and m outputs circuit's constraint circuit are shown in Figure4 and Figure5.

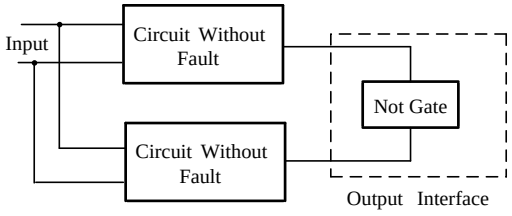


Figure 4. Constraint Networks for Single-output Circuit

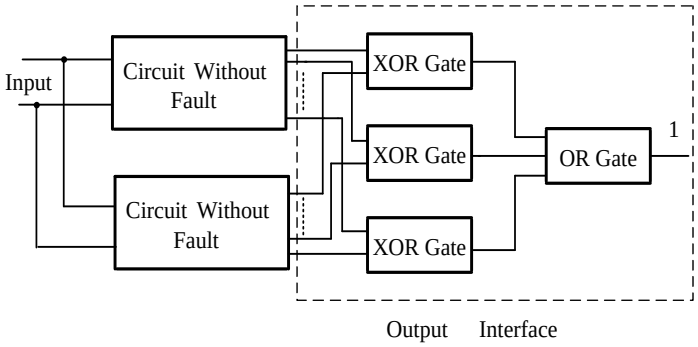


Figure 5. Constraint Networks for M-output Circuit

The test vectors for single stuck-at fault are consistent states for the constraint network and the energy function's value of the constraint network is zero at that time. So the test generation for single stuck-at fault can be transformed to find the zero value of energy function of the constraint network.

4. Algorithm's Realization

4.1. Simplify the Energy Function of Constraint Circuit

The energy function of constraint circuit is often very complicated, especially the complex digital circuit. So it is necessary to simplify the energy function of constraint circuit[12]. For the constraint circuit of single output circuit shown in Figure 4, the outputs of fault-free circuit and fault circuit must be different, so the interface circuit (NOT gate) must be in consistent state, so the test vector for single stuck-at fault can be obtained by solving minimum value of energy function of the interface circuit. The energy function of NOT gate is following:

$$E = 4V(i)V(j) - 2V(i) - 2V(j) + 2 \quad (3)$$

Here, V(i) and V(j) are the outputs of fault-free and fault circuit. For example, s-a-1 fault exists at V_4 in Figure 2, the constraint circuit of Figure 2 is shown in Figure 6.

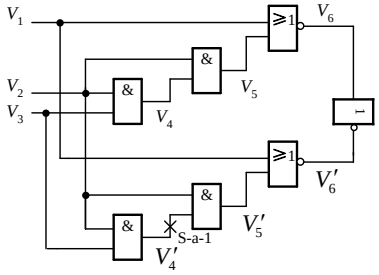


Figure 6. Constraint Circuit of Figure 2

The interface circuit of Figure 6 is a NOT gate, so the energy function of the interface circuit is following:

$$E = 4V_6V_6' - 2V_6 - 2V_6' + 2 \quad (4)$$

The simplified energy function of multiple output circuit is also the energy function of interface circuit.

4.2. Solving the Zero Value of Energy Function with Artificial Bee Colony Algorithm

Artificial bee colony[13] was proposed by the Turkish researcher Karaboga in 2005, it is a bionic optimization algorithm of simulating bee to find the best food source. The algorithm has local and global search in each of the iteration, so it can reduce the probability of entering into the local optimal solution. When using artificial colony algorithm to solve the optimization problem, the location of the food source was abstracted as the optimal solution, the fitness function value of optimization question determines the quality of the food source. Artificial bees mainly include the employed bees, onlookers and scout bees[14-15]. The fitness function is the energy function of interface circuit. Assuming that the algorithm has N initial population[x_i](i=1,2,…,N), [x_i] has m variables(m is the number of circuit input), every variable's value may 0 or 1.The employed bee has neighborhood search for one food source randomly, and updating food source's location according to formula(5).

$$[x_{i+1}] = [x_i] + \{[x_i] - [x_{i-1}]\}\delta_i + [I_i] \quad (5)$$

Where δ_i is random variety belong to [-1,0,1], $[I_i]$ is correction matrix, its inner number is also belong to [-1,0,1].

A food source is selected randomly first, then a new food source is obtained according to formula (5), plug the new food source into energy function E(x), then [x_i] that make E(x) 0 is a test vector for single stuck-at fault, otherwise research should be continued. When all employed bees finish searching, they can convey information to the onlookers by means of wagging dance, onlookers select food source according to roulette rule and keep the food source that has small energy function value.

In the control algorithm, set search control parameter as L, it is upper limit that food source has not been updated. If a food source has not been updated after L times search, the food source has gone into local optimum, the food source should be abandoned, the corresponding employed bee is changed to scout bee, a new food source should be generated according to formula(6) instead of the original food source.

$$[x_{i+1}] = [x_i] + \mathrm{rand}(0,1)[x_{i-1}] \quad (6)$$

The algorithm flow chart is shown in Figure 7.

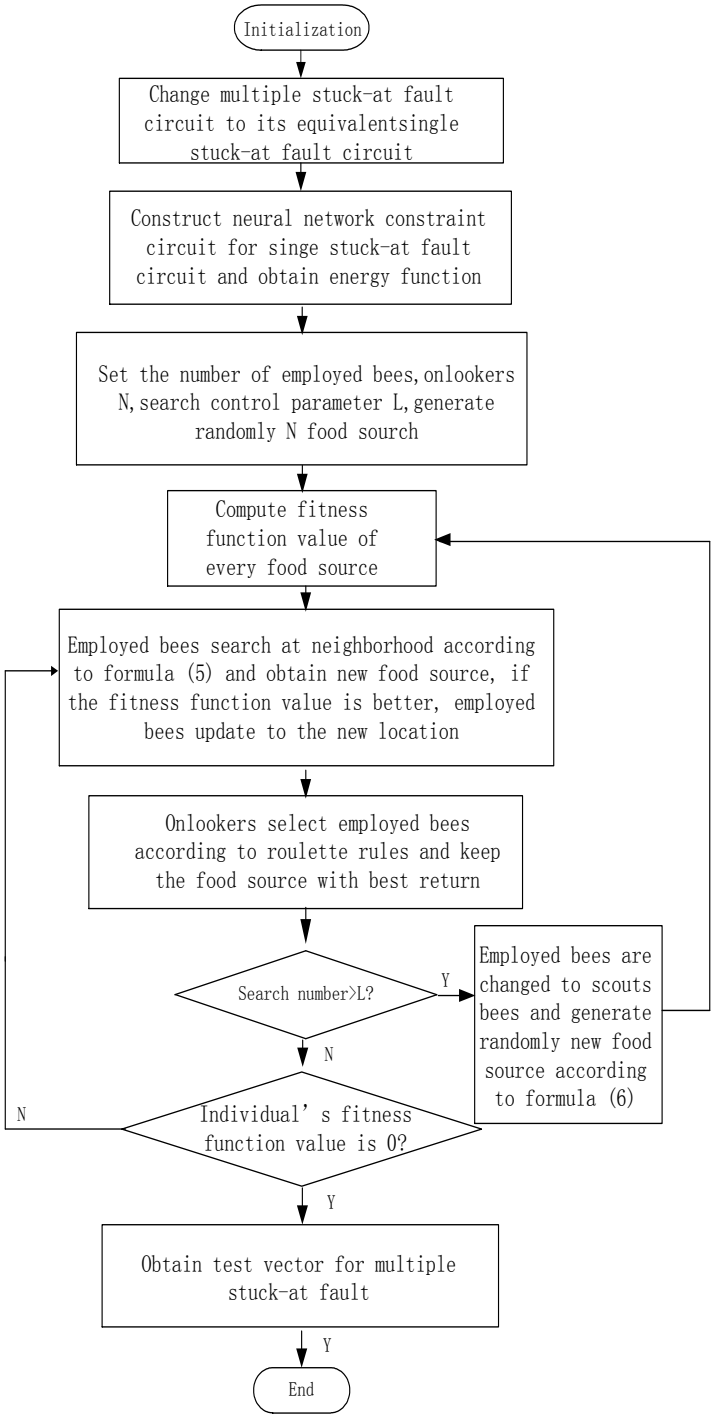


Figure 7. Flowchart of Algorithm

For example, if a circuit's multiple stuck-at fault is equivalent with s-a-1 fault at V_4 in Figure 2, then we solve the test vector for s-a-1 fault at V_4 using the algorithm in this paper.

Set $[x_i] = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$, express 3 input vectors of the circuit,

Set

$$[x_1]=\begin{bmatrix}0\\0\\1\end{bmatrix},[x_0]=\begin{bmatrix}0\\1\\1\end{bmatrix},\delta_1=-1,[I_1]=\begin{bmatrix}0\\1\\1\end{bmatrix}$$

Then according to formula（5）

$$[x_2]=[x_1]+\{[x_1]-[x_0]\}\delta_1+[I_1]$$

$$=\begin{bmatrix}0\\0\\1\end{bmatrix}+\begin{bmatrix}0\\-1\\0\end{bmatrix}\times(-1)+\begin{bmatrix}0\\1\\1\end{bmatrix}=\begin{bmatrix}0\\1\\1\end{bmatrix}$$

so，$\begin{bmatrix}V_1\\V_2\\V_3\end{bmatrix}=\begin{bmatrix}0\\1\\1\end{bmatrix}$，plug them into figure 6，$V_6=0$，$V_6'=0$，plug into formula（4），E=2，energy function is not 0，so the vector is not test vector for s-a-1 fault at V_4.，research should be continued.

Set

$\delta_2=1,[I_2]=\begin{bmatrix}0\\1\\-1\end{bmatrix}$，plug into formula（5），then

$$[x_3]=[x_2]+\{[x_2]-[x_1]\}\delta_2+[I_2]$$

$$=\begin{bmatrix}0\\1\\1\end{bmatrix}+\begin{bmatrix}0\\1\\0\end{bmatrix}\times 1+\begin{bmatrix}0\\1\\-1\end{bmatrix}=\begin{bmatrix}0\\1\\0\end{bmatrix}$$

plug $\begin{bmatrix}V_1\\V_2\\V_3\end{bmatrix}=\begin{bmatrix}0\\1\\0\end{bmatrix}$ into figure6，$V_6=1$，$V_6'=0$，plug into formula（4），E=0，then the vector is　test vector for s-a-1 fault at V_4.

5. Experiment Results

ISCAS’85 benchmark circuits are international standard circuit collection, the merits of the algorithm is usually tested on the circuits. Circuit C17 of ISCAS’85 is shown in Figure 8, there are 5 inputs and 2 outputs, and it is composed of 6 NAND gate circuits. The experiment results of the algorithm on C17 is shown in figure 9(program with C++ language).The experiment results of the algorithm on C432 and C499 are shown in Table 3 and Table 4.The experiment results of the algorithm comparing with other algorithms are shown in Table 5.The experiment results show that the fault coverage increases significantly and the average test generation time is shorter, so the algorithm is better than other algorithm.

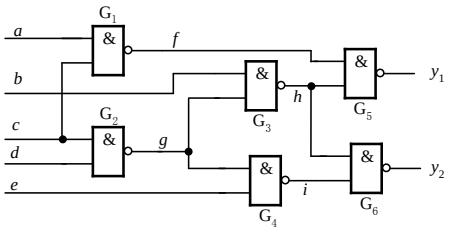


Figure 8. Circuit C17 of ISCAS'85

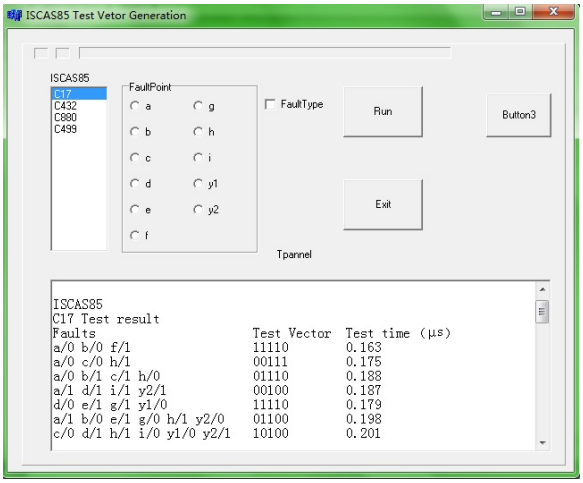


Figure 9. Experiment Results of Circuit C17

Table 3. Experiment Results of C432 Circuit

Circuit	Multiple fault （fault point /fault type）	Test generation time （μs）	Test vector
C	**1/1,4/1,5/0,12/0,27/1,32/0,49/1,85/0**	**0.202**	**00111010110100011101111000011101 0011**
4	**5/0,8/1,9/0,10/0,25/0,38/1,55/0,90/0,105/0, 129/1**	**0.223**	**11011011010100011101010100110100 001**
3	**2/1,3/0,6/1,14/0,28/0,42/1,49/1,180/0,257/1 ,326/0**	**0.214**	**00100010110101011101011001011101 0010**
2	**10/1,21/0,50/0,62/0,75/1,82/0,95/1,108/0,2 88/0,313/1**	**0.211**	**10111011010101010001111010010101 1111**
	7/1,10/1,16/0,32/0,40/1,92/0,108/1,180/0, 183/0,213/1,317/0	**0.241**	**10101010010101100101100110010101 0000**

Table 4. Experiment Results of C499 Circuit

Circuit	Multiple fault （fault point /fault type）	Test generation time（μs）	Test vector
C499	**5/0,11/1,15/0,19/0,29/1,37/0,46/1,88/0,120/0,221/1,305/0**	**0.205**	**10111010110101011101111010011001001100 11**
	7/0,8/0,19/0,30/0,45/0,50/1,55/0,72/0,108/0,128/1, ,288/0	**0.233**	**01110110110101000111010101001101000 01110**
	2/0,8/0,12/1,18/0,25/0,44/1,59/1,115/0,231/1,313/0	**0.234**	**00001000101101010111010110010111010 01010**
	15/1,29/0,55/0,60/0,73/1,86/0,95/1,107/0,248/0,300/1	**0.241**	**01101110100101010100011010100101 01111101**
	7/1, 32/0,40/1,92/0,105/1,180/0, 188/0,213/1,310/0	**0.258**	**11101011100101011001011001100101 01010001**

Table 5. Experiment Test Results of Different Algorithms

circuit		C432	C499	C880
Algorithm in this paper	Average test generation time（μs）	0.201	0.242	0.248
	Fault coverage	100%	98.9%	99.2%
Reference [2] algorithm	Average test generation time（μs）	0.467	0.489	0.491
	Fault coverage	91.4%	96.2%	95.6%
Reference [3] algorithm	Average test generation time（μs）	0.255	0.366	0.381
	Fault coverage	81.8%	86.2%	85.6%

6. Conclusion

In this paper, neural network and artificial colony optimization algorithm are applied to test generation for multiple stuck-at faults of digital circuit, neural network and artificial colony algorithm's advantages are made full use of and avoiding getting into local optimal solution. The experimental results in this paper show that the fault coverage algorithm can achieve 98.5%, average test generation time is less than 0.25μs, the algorithm has obvious advantages compared with other literature algorithm.

Acknowledgements:

This work was supported by the National Natural Science Foundation of China (Study on higher-order logic based inductive logic programming learning algorithm and its application, 61300098)

References

[1] T. Chakrdhar, M. L. Bushnell and V. D.Agrawal, "Toward Massively Parallel Automatic Test Generation", IEEE Trans.Computer_Aided Design, vol. 9,**(1999)**, pp. 81-83.
[2] X. J. Bin and L. Zhi, "The Application of Neural Networks in the Simulation-based Test Generation Algorithm for Hybrid Circuits", Journal of Circuits and Systems, vol. 6,no. 4, **(2001)**, pp. 109-110.
[3] J. Ferrer and P. M. Kruse, "Search based algorithms for test sequence generation in functional Testing",Information and Software Technology, vol. 7, no. 14,**(2014)**, pp. 4-10.
[4] S.Neophytou and M. K. Michael, "Multiple detection test generation with diversified fault partitioning paths", Microprocessors and Microsystems, vol. 38,no. 4,**(2014)**, pp. 85-90.
[5] C. R. WEI, X.-L. YAN and S. Yuling, "Research and Implementation of Boundary Scan Test Generation and Fault Diagnosis **"**, Computer Engineering，vol. 41, no. 1,**(2015)**, pp. 303-306.
[6] C. K.YONG, V. D. Agrawal, "Multiple faults:modeling,simulation and test", Proc.of 15th International Conference on VLSI Design, **(2012)** , pp. 55-58.
[7] H.Q. LUO, L. P. LIANG and T.C. YE, "Genetic Algorithm Based on Random Test Generation: A Case Study", Electronic Test，vol. 45, no. 13,**(2013)**, pp. 75-78
[8] Z. Chen and M. Ding, "A robust algorithm for test generation based on neural network", Journal of Huazhong university of science and technology, vol. 27,no. 9, **(1999)**, pp. 90－91.
[9] X. Liu and S. Sun, " Neural networks based test generation algorithm for combinational logic circuits", Journal of Harbin institute of technology, vol. 34, no. 22, **(2002)**, pp. 56-57.
[10] R. Nourmandi-Poura and N. NafisehMousavian, "A fully parallel BIST-based method to test the crosstalk defects on the inter-switch links in NOC", Microelectronics Journal, vol. 35, no. 1,**(2013)** ,pp. 48-50.
[11] J. Balcarek, P. Fiser and J. Schmidt, "Techniques for SAT-based constrained test pattern generation", Microprocessors and Microsystems, vol. 39, no. 10, **(2001)** , pp. 185-189.
[12] L. W. WU and X. D. WANG, "Three-valued neural network test generation algorithm for Multi-faults based on genetic optimization", Chinese Journa l of Scientific Instrument, vol. 31, no. 8, **(2010)**, pp. 44-49.
[13] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Kayseri:Erciyes University, **(2005)** , pp. 35-45.
[14] N.A. Elkhateeb and R. I. Badr, "Employing artificial bee colony with dynamic inertia weight for optimal tuning of PID controller".ICMIC2013, **(2013)** , pp. 42-45 .
[15] G. P. Dixit and H.M. Dubey, "A rtificial bee colony optimization for combined economic load and emission dispatch", SEISCON2011, **(2011)** , pp. 340-342.

Authors

ZHAO Ying, born in September,1976,Jilin,P R China.She received her MS degree from Harbin University of science and technology in 2005. She is a associate professor in electrical &information engineering College of Beihua University. Her research interests are integrated circuit test and design for measurable performance of circuit. She conducted 5 scientific

research projects, published 15 papers and 5 electronic books.

LIU Bai-sheng,born in 1979, Jilin,P R China.He received his MS degree from Jilin normal university in 2004. He is a intermediate lecturer in electrical &information engineering College of Beihua University. His research interests are signal system and digital signal processing. He conducted 2 scientific research projects, published 1 book.

LI Yan-juan , born in April ,1979,Heilongjiang,P R China. She received her MS degree from Harbin University of science and technology in 2005 and Ph.D degree from Harbin institute of technology in 2010. She is a associate professor in College of Information and Computer Engineering of Northeast Forestry University. Her research interests are inductive logic programming algorithm, neural network and artificial bee colony. She conducted 4 scientific research projects, published 12 papers, and one paper was indexed by SCI.