International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 2, No. 2, July 2013, pp. 55~63
ISSN: 2089-4864

A Novel Evolutionary Method for Designing Optimized Multifunctional Logic Modules

Mehdi Anjomshoa*, Ali Mahani**
* South Pars Gas Complex, Bushehr, Iran
** Department of Electrical and Electronic Engineering, Shahid Bahonar University, Kerman, Iran

Article Info

Article history:

Received Mar 27, 2013
Revised Jun 21, 2013
Accepted Jul 2, 2013

Keyword:

Imperialist Competitive Algorithm
Logic Circuit
Multi Functional Module

ABSTRACT

In this paper, we proposed a novel heuristic method based on Imperialist competitive Algorithm (ICA) to design combinational logic modules which performing different arithmetic functions. According to conventional methods, for multi functional circuit, a distinct circuit is designed for each specific function and then all of them are combined together with multiplexer(s) to have desired circuit. But in our proposed method the whole circuit structure is designed and optimized in one procedure by ICA Algorithm. We tried to optimize the area of circuit by reducing the number of transistors forming logic gates. Simulation results show that our method significantly reduces the number of transistors and gates and accordingly the circuit area.

Corresponding Author:

Mehdi Anjomshoa,
Department of Control and Instrumentation,
South Pars Gas Complex,
Bushehr, Iran
Email: Anjomshoa_m@yahoo.com

1. INTRODUCTION

Arithmetic logic unit (ALU) is one of the most important parts in a microprocessor which runs a lot of functions. Different modules such as adder, subtractor, multiplier and comparator and logic units are designed separately and then come together with multiplexers in ALU and the outputs of these modules are managed with control bits.

Due to the fast improvement in digital electronic devices (regarding to their size), designing compact and optimized circuits, in such devices is so important. In this case, the lower number of transistors which forming logic gates, leads to lower circuit size on chip and production cost will be cheaper.
Actually, many designing approaches, such as Boolean algebra, Karnaugh map [1], and Quine–McCluskey [2] have been widely used in solving the digital circuit optimization problem. These methods depend on human knowledge and creativity and in case of complex circuits; they could not effectively have the satisfactory solution.

Evolutionary hardware (EHW) is an automated design method of circuits based on artificial evolution of natural phenomena. EHW uses evolutionary algorithms (EA) to auto configure and optimize circuits [3], [4]. By exploring a large search space, EHW may find solutions for a task which is unsolvable or difficult to solve.

In this paper we use Imperialist Competitive Algorithm as searching engine. It has been introduced by Atashpaz-Gargari and Lucas [5], which has inspired from a socio-political phenomenon, and it has been applied in various papers for dealing with different optimization tasks [6]-[8].

The aim of this paper is designing and optimization two circuits: first, a full adder-subtractor circuit with common outputs and second, a circuit which running different arithmetic functions at the same time on different outputs.

The rest of this paper is organized as follows. In section II, ICA algorithm will be introduced. Section III describes structure of circuit and the usage of ICA as a new approach for automatic design of an optimized circuit. Section IV shows simulations and results. Finally, Section V concludes.

2. IMPERIALIST COMPETITIVE ALGORITHM

Imperialist Competitive Algorithm (ICA) [5] is a new evolutionary algorithm in the Evolutionary Computation field based on the human's socio-political evolution. Like other evolutionary algorithms, it starts with an initial random population which is called countries. Some of the best countries in the population selected to be the imperialists and the rest form the colonies of these imperialists. In an N dimensional optimization problem, a country is a 1× n array. (Equation (1)).

$$Country = [\, p_1, p_2, ..., p_n] \tag{1}$$

The cost of a country is found by evaluating the cost function f at the variables ($p_1, p_2, ..., p_n$). Then:

$$c_i = f(country_i) = f(p_1, p_2, ..., p_n) \tag{2}$$

The algorithm starts with N initial countries and the N_{imp} best of them (countries with minimum cost) chosen as the imperialists. The remaining countries are colonies that each belong to an empire. The initial colonies belong to the imperialists according to their powers. To distribute the colonies among imperialists proportionally, the normalized cost of an imperialist is defined as Equation (3).

$$C_n = \max(c_i) - c_n \tag{3}$$

Where, c_n is the cost of nth imperialist and C_n is its normalized cost. Each imperialist that has more cost value, will have less normalized cost value. Having the normalized cost, the power of each imperialist is calculated from Equation (4) and based on that the colonies distributed among the imperialist countries.

$$p_n = \left| \frac{c_n}{\sum_{i=1}^{N_{imp}} c_i} \right| \tag{4}$$

On the other hand, the normalized power of an imperialist is assessed by its colonies. Then, the initial number of colonies of an empire will be $NC_n = rand\{p_n \times N_{col}\}$ where, NC_n is initial number of colonies of nth empire and N_{col} is the number of all colonies.

To distribute the colonies among imperialist, NC_n of the colonies is selected randomly and assigned to their imperialist. The imperialist countries absorb the colonies towards themselves using the assimilation policy. The assimilation policy shown in Figure 1 makes the main core of this algorithm and causes the countries move towards to their minimum optima. The imperialists absorb these colonies towards themselves with respect to their power that described in Equation (4).

The total power of each imperialist is determined by the power of its both parts, the empire power plus percents of its average colonies power.

$$T.C_n = Cost(imperialist_n) + \xi\, mean\{Cost(colonies\, of\; impire_n)\} \tag{5}$$

Where $T.C_n$ is the total cost of the nth empire and ξ is a positive number which is considered to be less than one.

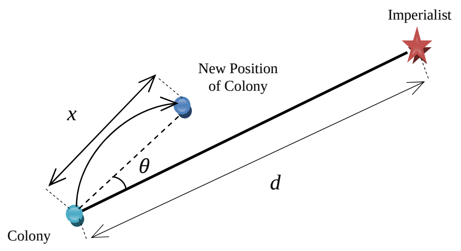


Figure 1. Moving colonies toward their relevant Imperialist [5]

In the assimilation policy, the colony moves towards the imperialist by x unit. The direction of movement is the vector from colony to imperialist, as shown in Figure 1, the distance between the imperialist and colony shown by d and x is a random variable with uniform distribution. Where β is greater than 1 and is near to 2 ($x \sim U(0, \beta \times d)$).

In ICA algorithm, to search different points around the imperialist, a random amount of deviation is added to the direction of colony movement towards the imperialist. In Figure 1, this deflection angle is shown as θ, which is chosen randomly and with a uniform distribution. ($\theta \sim U(-\gamma, \gamma)$).

While moving toward the imperialist countries, a colony may reach to a better position, so the colony's position changes according to the position of the imperialist. In each running cycle, we select some of the weakest colonies and replace them with new ones, randomly. The replacement rate is named as the revolution rate. In ICA, revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis. The revolution increases the exploration of the algorithm and prevents the early convergence of countries to local minimums.

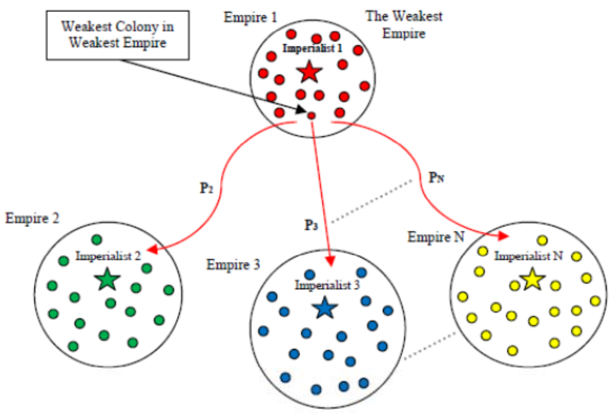


Figure 2. Imperialistic competition

The more powerful an empire is, The more likely it will possess the weakest colony of weakest empire [5].

In this algorithm, the imperialistic competition has an important role. During the imperialistic competition, the weak empires will lose their power and their colonies (Figure 2). To model this competition, firstly we calculate the probability of possessing all the colonies by each empire considering the total cost of empire.

$$NTC_n = \max\{TC_i\} - TC_n \tag{6}$$

Where, TC_n is the total cost of nth empire and NTC_n is the normalized total cost of nth empire. Having the normalized total cost, the possession probability of each empire is calculated from Equation (7).

$$P_{pn} = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \qquad (7)$$

After a while all the empires, except the most powerful one will collapse and all the colonies will be under the control of this unique empire.

3. PROPOSED METHOD

3.1. Encoding of Circuit Structure

Imperialist Competitive Algorithm starts with some random colonies and after evaluation, some of these colonies are selected as imperialist. In this work our logic circuit structures are introduced as colonies. A two dimensional matrix is a common structure that has been used in many research works which was proposed by Louis and Rawlins [9]. That structure is shown in Figure 3. Each cell of the m × n matrix contains the information of the gate type and its corresponding inputs. However, unlike the fixed interconnection rules by Louis and Rawlins [9], the inputs of each unit can be randomly connected by any element output in previous stages and there is no feedback between cell elements. Each cell of the matrix is encoded in an array of five integer numbers. First and second number of the array is the address of the cell which the first input of the gate is connected and third and fourth number of the array is for second input and finally fifth number indicates gate type.

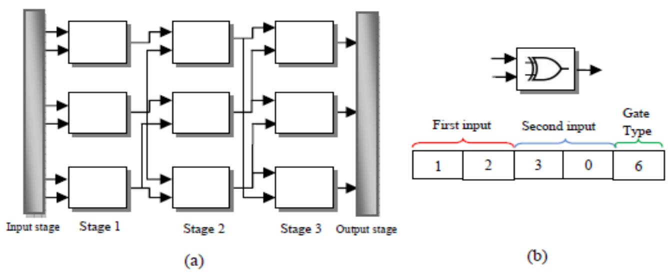


Figure 3. (a) Two dimensional random matrix as circuit structure, (b) Cell encoding

Different logic cells (NOT, AND, OR, NAND, NOR, XOR, XNOR and wire) used in our algorithm. Wire is assumed as a logic gate that transfers data from its input to output without any changes. Cells information which was extracted from the open source standard cell vsclib013 [10] in 0.13 micron technology is shown in Table 1.

Table 1. Standard CMOS cell information from vsclib013 library.

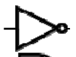
Gate Type Code	Gate Symbol	Area (μm)	Number Of Transistors	Cell Name
0	—	0	0	WIRE
1		1728	2	NOT
2		2880	6	AND
3		2880	6	OR
4		2304	4	NAND
5		2304	4	NOR
6		4608	9	XOR
7		5184	9	XNOR
8		4608	12	MUX(2x1)

Figure 4 shows how logic cells are encoded in a matrix of circuit structure. This circuit has 3 inputs (A, B, C) and one output (OUT). As shown in Figure 4(a), 3×3 matrix is populated with random logic gates and their interconnections. In Figure 4(b) we have encoded matrix and in Figure 4(c) the effective gates of the circuit are shown. Effective gates are those which connect output(s) of circuit to its inputs. Cell (2,3) which attribute is (1,2,2,2,7) is an XNOR gate (according to Table 1).Output of the circuit is the output of this gate and the first input of this cell is connected to the output of the cell (1,2), which is a NAND gate, and the second input is connected to the output of an OR gate in (2,2). Cells in location (1,1), (2,1) and (3,1) are AND,NOR and wire respectively and they are connected to primary inputs A, B and C.

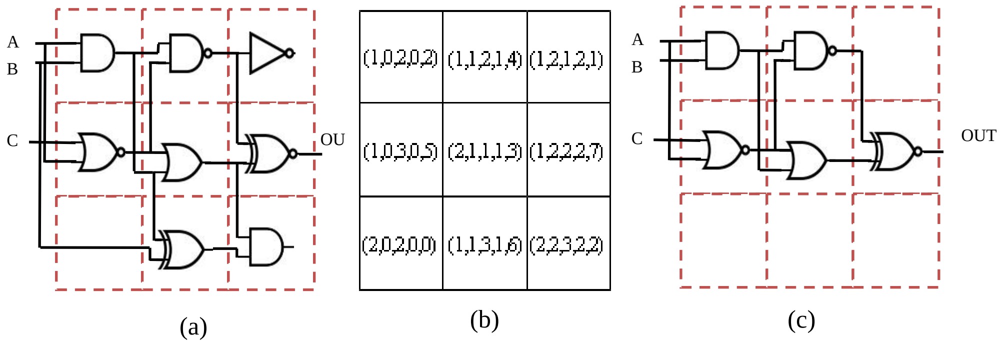
3.2. Circuit Evaluation

After initializing random matrixes of logic cells as colonies, all circuits should be evaluated. Here, to evaluate the evolving circuits, two main issues should be taken into consideration: 1) Functionality, 2) optimization in term of space. Truth table is a table that shows the status of circuit output(s) according to different combinations of circuit inputs. It is used to investigate the functionality of circuit. A multi-objective evaluation mechanism in the form of a weighted cost function introduced to obtain both functional and optimized circuits as follows:

$$f=\frac{W_{match}\times N_{match}+W_{transistor}\times N_{transistor}}{W_{match}+W_{transistor}} \tag{8}$$

Where W_{match} is weight of N_{match} and $W_{transistor}$ is weight for number of transistor representing circuit area.

$$N_{match}=\frac{\text{the number of the correct outputs from circut}}{\text{the number of outputs from truth table}} \tag{9}$$



(1,0,2,0,2)	(1,1,2,1,4)	(1,2,1,2,1)
(1,0,3,0,5)	(2,1,1,1,3)	(1,2,2,2,7)
(2,0,2,0,0)	(1,1,3,1,6)	(2,2,3,2,2)

Figure 4. Example of a circuit and its encoding: (a) circuit schematic with logic gates and interconnections. (b) Encoding of circuit. (c) Effective gates in matrix

Area of a circuit is calculated by the effective area of logic gates without considering interconnections.

Since, the circuits evolving in the algorithm should be functional before optimizing, we consider W_{match} = 10 $W_{transistor}$ which is obtained by experiment.

During search process, we meet some circuits which their output does not match truth table output or N_{match} is less than 50%. For these cases we invert the output gate of the circuit (e.g. XNOR instead of XOR) and change N_{match} as follows:

$$N_{match}=\max\{N_{match},1-N_{match}\} \tag{10}$$

3.2. Modification of Assimilation Policy

The original version of Imperialist competitive algorithm operates on real values. Imperialists tend to improve their colonies behaviour. This fact has been modeled by moving all the colonies toward their Imperialist. As shown in Figure 5(a), colonies move toward their relevant imperialist along a line between them or sometime with a random deviation angle. However, with a simple modification, the Imperialist competitive algorithm can be made to operate on digital circuit structures.

In logic circuit structure some random cell from colonies are removed and then these positions are filled with the cells of imperialist. Figure 5(b), Shows modified assimilation on circuit structures. In this Figure, the left matrix is imperialist and the middle one is colony. Some random cells (i.e. 1, 3, 5, and 7) are selected and replaced in colony structure. Finally right matrix is obtained by assimilation policy.

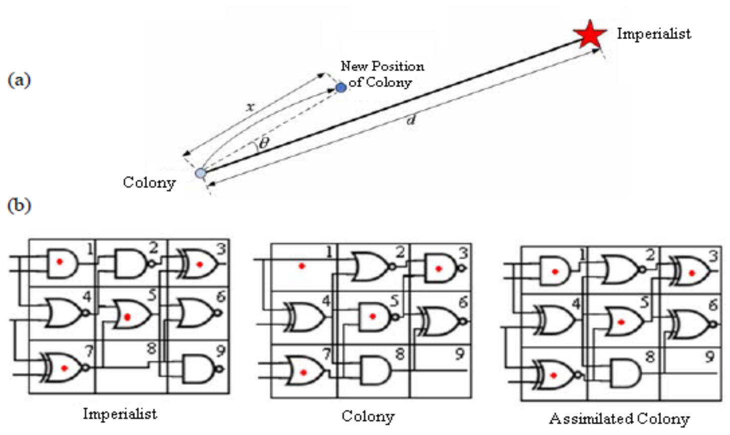


Figure 5. a) Moving colonies toward their relevant imperialist, b) Modification of moving colonies toward Imperialist

4. SIMULATIONS AND RESULTS

The proposed method has been simulated on a PC with Intel(R) Pentium(R) Dual CPU 2.20GHz and 1 GB RAM and in Matlab2009 and 20 independent experiments (runs) were performed for each circuit. To check the correctness of designed circuits, they were tested in MAXPLUS II 10.2.

Parameters of algorithm are: number of countries (350), number of imperialists (35), zeta (0.05), and assimilation rate (0.4) and revolution rate (0.3). The size of Matrix, which was used as circuit structure for first circuit is 3x3 and for second is 4x7.

4.1. Full Adder-Subtractor Module with Control bit

In this case, we tried to design a circuit which performs addition and subtraction functions on the same output(s) and manage circuit output by control bit.

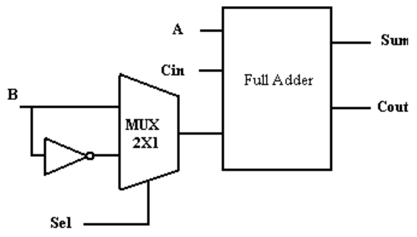


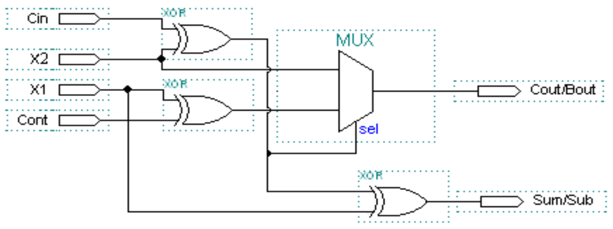
Figure 6. Full adder/subtractor block diagram

Table 2. Truth Table of Full Adder-subtractor

A	B	C_{in}	Sum / Sub	C_{out}	B_{out}
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	1	1

Each full Adder and full subtractor circuit (if optimized) has 5 gates and 30 transistors separately and using both of them in a unit module; needs 10 gates and 60 transistors totally. But it is not a popular method because by one bit complement and using 2x1 multiplexer, subtraction function can be done in adder circuit and then both functions are implemented in one module. As shown in Figure 6 by setting "sel" input of multiplexer ("0" or "1") addition and subtraction functions can be achieved on the outputs. (F=A+ (-B)). Table 2 shows truth table of these functions.

Designing this circuit with Karnaugh method, results in a circuit with 10 gates and 58 transistors, but we designed and optimized this module with ICA and a circuit with 4 gates and 39 transistors obtained as shown in Figure 7, so we achieved a circuit with approximately 38% reduction in space.



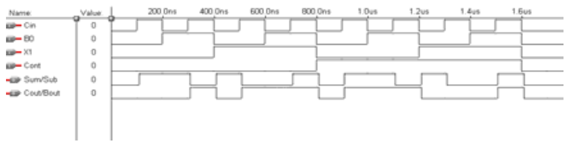


Figure 7. Full adder/subtractor circuit designed by ICA and wave form of inputs/outputs verified in MAXPLUS II

After 20 times running ICA algorithm for this circuit, average iteration (329.3) with standard deviation (84.02) achieved.

In this circuit x1, x2 are inputs and C_{in} is carry-in for adder and barrow-in for subtractor and "Cont" is control bit (0= addition 1=subtraction). By setting control bit, we have two different functions on the same outputs.

To design N-bit full adder-Subtractor we use N proposed module in series as shown in Figure 8.

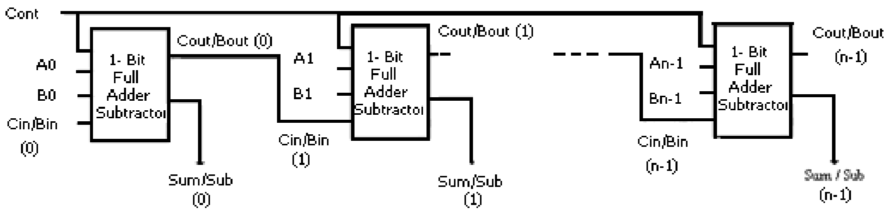


Figure 8. N-Bit full adder-Subtractor resulting from 1-bit optimized modules

4.2. Multi-Function Module

In this case, we designed a circuit that performs different arithmetic functions (Addition, Subtraction, Multiplication, and Comparison) at the same time on circuit outputs. It means when you apply inputs, you have all functions on the outputs simultaneously. Truth table of this module is shown in Table 3.

Table 3. Truth Table of Multi-Function Module

A	B	C_{in}	Sum Sub	C_{out}	B_{out}	Mul	A>B	A=B	A<B
0	0	0	0	0	0	0	0	1	0
0	0	1	1	0	1	0	0	1	0
0	1	0	1	0	1	0	0	0	1
0	1	1	0	1	1	0	0	0	1
1	0	0	1	0	0	0	1	0	0
1	0	1	0	1	0	0	1	0	0
1	1	0	0	1	0	0	0	1	0
1	1	1	1	1	1	1	0	1	0

Our proposed circuit is shown in Figure 9. After 20 times running ICA algorithm for this circuit, average iteration (945.12) with standard deviation (179.26) achieved.

This circuit includes 9 gates and 63 transistors. Using this circuit instead of 4 distinct circuits can save the area on digital chips significantly.

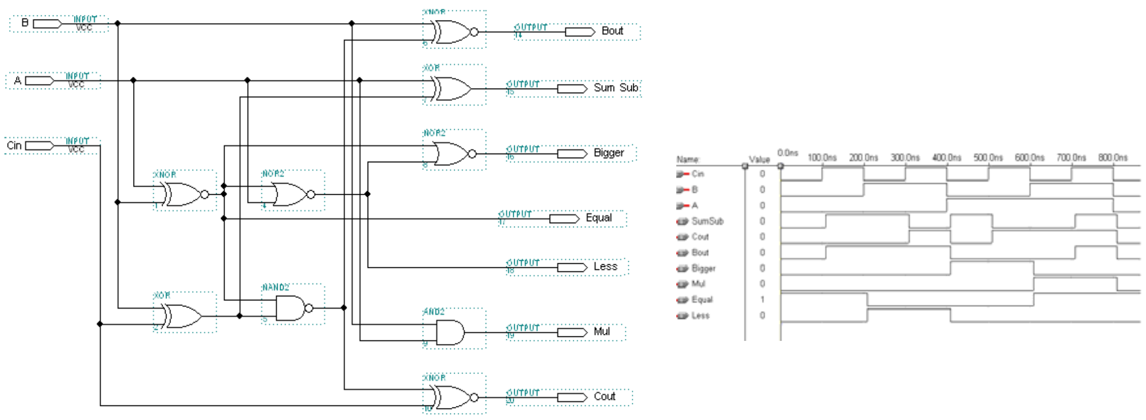


Figure 9. Multi-Functional circuit designed by ICA and wave form of inputs/outputs verified in MAXPLUS II

5. CONCLUSION

In this paper, we proposed a novel evolutionary method based on Imperialist Competitive Algorithm for designing and optimization digital circuits which perform multi arithmetic functions. These circuits are very useful in ALU of microprocessors. By this creative method, compact circuits with various functions are designed instead of different distinct circuits. Simulations clearly show that this method significantly reduce the area of whole circuit and thereby lower number of circuits elements results in lower power consumption of digital chips.

REFERENCES

[1] M Karnaugh. A map method for synthesis of combinational logic circuits. *Transactions of the AIEE, Communications and Electronics*. 1953; 72(1): 593- 599.
[2] EJ McCluskey. Minimization of Boolean functions. *Bell Systems Technical Journal*. 1956; 35(5): 1417-1444.
[3] X Yao, T Higuichi. Promises and Challenges of Evolvable Hardware. *IEEE Trans On Systems Man and Cybernetics, Part C*. 1999; 29: 87-97.
[4] CY Chen, RC Hwang. A new variable topology for evolutionary hardware design. *Expert Systems With Applications*. 2009; 36: 634-642.
[5] E Atashpaz-Gargari, C Lucas. Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. *IEEE Congress on Evolutionary Computation (CEC 2007)*. 2007: 4661-4667.
[6] C Lucas, Z Nasiri-Gheidari, F Tootoonchian. Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Conversion and Management*. 2010; 51(7): 1407–1411.
[7] H Mozafari, B Abdi, A Ayob. Optimization of Transmission Conditions for Thin Interphase Layer Based on Imperialist Competitive Algorithm. *International Journal on Computer Science and Engineering (IJCSE)*. 2010; 2(7): 2486–2490.
[8] H Sayadnavard, A Haghighat, M Abdechiri. Wireless sensor network localization using Imperialist Competitive Algorithm. *3rd IEEE International Conference on Computer Science and Information Technolog*. 2010.
[9] RC Louis, GJE Rawlins. Designer genetic algorithms: Genetic algorithms in structure design. *Fourth international conference on genetic algorithms. San Mateo, CA: Morgan Kauffman*. 1953: 53–60.
[10] G Petley. ASIC standard cell library design. *http://www.vlsitechnology.org/*. (Accessed 5 October 2012).

BIOGRAPHIES OF AUTHORS

Mehdi Anjomshoa (Born 1978) received his B.S in Electronic Engineering from Shahid Bahonar University, Kerman, Iran and his M.Sc. in 2011 in Electronic Engineering from Islamic Azad University, Bushehr Branch, Bushehr, Iran. Now he is working for South Pars Gas Complex as Instrument senior engineer and His current research interests include artificial intelligence, evolutionary electronics, and automatic control systems.

Ali Mahani (Born 1978) received the PhD in Electrical engineering from Iran University of Science and Technology (IUST) in 2009. Since then he has been with the Department of Electrical and Electronic Engineering of Shahid Bahonar University, where he is currently an assistant professor. His interests focus on Fault tolerant digital design and Performance evaluation of Communication Networks.