
# A Novel Evolutionary Method for Designing Optimized Multifunctional Logic Modules

**Mehdi Anjomshoa, South Pars Gas Complex, Anjomshoa_m@yahoo.com**
**Ali Mahani, Shahid Bahonar University of Kerman, Amahani@uk.ac.ir**

Paper Reference Number: 07-62-2001
Name of the Presenter: Mehdi Anjomshoa

**Abstract**

In this paper, we proposed a novel heuristic method based on Imperialist competitive Algorithm (ICA) to design logic modules which performing different arithmetic functions. In conventional methods, for designing multi functional circuit, for each specific function, a distinct circuit is designed and then all of them will be combined together with multiplexer(s) to have desired circuit but in our proposed method the whole structure of circuit is designed and then optimized in one procedure by ICA Algorithm. We tried to optimize the area of circuit by reducing the number of transistors forming logic gates. To show the validity of this method, we design two circuits, first a full adder-subtractor circuit with a control bit and second a multi-function circuit which performs addition, subtraction, multiplication and comparison operation at the same time. Simulation was done in MATLAB 2009 and results shows that our method significantly reduces the number of transistors and gates and accordingly the area of circuits. Using such circuits in Arithmetic logic unit (ALU) of Central Processing Unit (CPU) can reduce the surface and power consumption of IC.

**Key words:** Imperialist Competitive Algorithm, Logic Circuit, Multi Functional Module, Optimization.

## 1. Introduction

Arithmetic logic unit (ALU) is one of the most important components 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, it is important to have compact and optimized circuits, in such devices. 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 hardware design approaches, such as Boolean algebra, Karnaugh map (1953), and Quine–McCluskey (1956) 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 (Yao and Higuichi, 1999, Chen and Hwang, 2009). 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 (2007) which has inspired from a socio-political phenomenon, and it has been applied in various papers for dealing with different optimization tasks (Lucas *et al.*, 2010; Mozafari *et al.*, 2010 and Sayadnavard *et al.*, 2010).
The aim of this paper is to design an optimized circuit for full adder-subtractor on the same outputs with control bit and a circuit to run different arithmetic function 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 use of ICA as an approach for the automatic design of an optimized circuit. Section IV shows simulations and results. Finally, Section V concludes.

## 2. Imperialist Competitive Algorithm

Imperialist Competitive Algorithm (ICA) is an evolutionary algorithm in the Evolutionary Computation field based on the human's socio-political evolution. Like other evolutionary algorithms, it starts with an initial population which is called country and is divided into two types of colonies and imperialists which together form empires. Imperialistic competition among these empires forms the proposed evolutionary algorithm. During this competition, weak empires collapse and powerful ones take possession of their colonies. Imperialistic competition converges to a state in which there exists only one empire and colonies have the same cost function value as the imperialist. The pseudo code of Imperialist competitive algorithm is as follows:
1) Select some random points on the function and initialize the empires.
2) Move the colonies toward their relevant imperialist (Assimilation).
3) Randomly change the position of some colonies (Revolution).
4) If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.
5) Compute the total cost of all empires.
6) Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (Imperialistic competition).
7) Eliminate the powerless empires.
8) If stop conditions satisfied, stop, if not go to 2.

## 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 (1991). That structure is shown in Fig. 1. 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 (1991), 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 number. First and second number of array is related to the first input of gate and third and fourth number of array is for second input and finally fifth number indicates gate type.

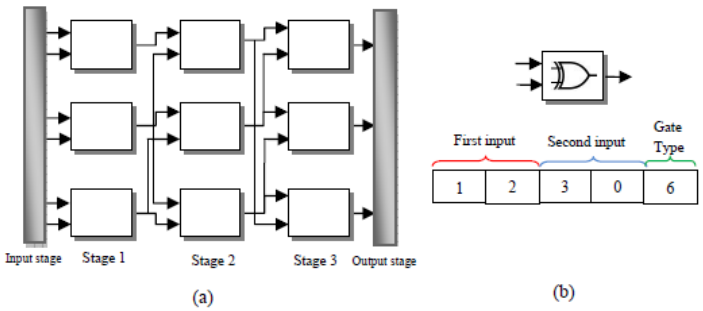


**Fig 1 :**(a) two dimensional random matrix as circuit structure. (b) Cell encoding.

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 change. Cells information which was extracted from the open source standard cells vsclib013 (Petley, http://www.vlsitechnology.org/) in 0.13 micron technology is shown in Table 1.

| Gate Type Code | Gate Symbol | Area ( $\mu m$ ) | 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) |

Table 1. Standard CMOS cell information from vsclib013 library.

Fig. 2 shows how logic cells are encoded in matrix of circuit structure. This circuit has 3 inputs (A,B,C) and one output (OUT). As shown in Fig. 2a, a $3\times3$ matrix is populated with random logic gates and their interconnections. In 2b we have encoded matrix and in 2c the effective gates of circuit are shown. Effective gates are those gates that connect output of

circuit to its inputs. Cell (2,3) whose attribute is (1,2,2,2,7) is an XNOR gate (according to Table 1).Output of circuit is the output of this gate and the first input of this cell is connected to the output of 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.

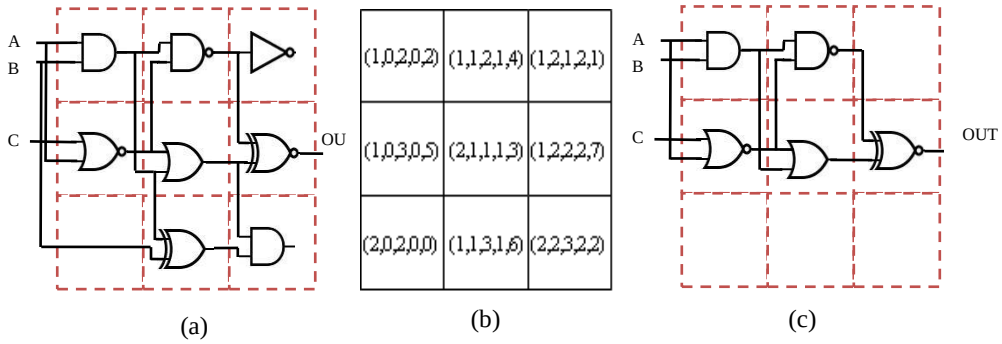


**Fig 2:** Example of a circuit and its encoding: (a) circuit schematic with logic gates and interconnections.(b) Encoding of circuit.(c) Effective gates in matrix.

### 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 taking 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:

$$f = \frac{W_{match} \times N_{match} + W_{transistor} \times N_{transistor}}{W_{match} + W_{transistor}} \tag{2}$$

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

$$f = \frac{the\ number\ of\ correct\ outputs\ \ from\ circut}{the\ number\ of\ \ outputs\ \ from\ truth\ table} \tag{3}$$

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:

$$N_{match} = \max\{N_{match}, 1 - N_{match}\} \tag{4}$$

### 3.3 Modification of assimilation policy

The original version of Imperialist competitive algorithm operates on real values. Imperialists tend to improve their colonies behavior. This fact has been modeled by moving all the colonies toward their Imperialist. As shown in Fig. 3a, 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. Fig. 3b, 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. 1,3,5,7) are selected and replaced in colony structure. Finally right matrix is obtained by assimilation policy.

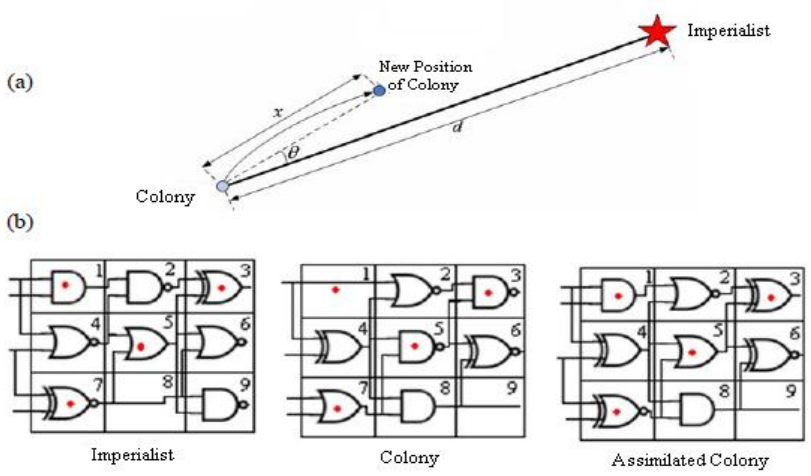


**Fig 3:** a) Moving colonies toward their relevant imperialist b) Modification of moving colonies toward Imperialist.

## 4. Simulations and Results

The proposed method has been simulated on a PC with Intel(R) Pentium(R) Dual CPU 2.20 GHz 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 which were obtained by experiment 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 circuit is 3x3 and for second is 4x7.

### 4.1 Full adder-Subtractor Module with control bit

Here, our aim is to design a circuit which performs addition and subtraction functions on the same output(s) and manage circuit output by control bit.

Each full Adder and full subtractor circuit (if optimized) has 5 gates and 30 transistors and if we use both of them in a module to have two arithmetic functions, we should use 10 gates and 60 transistors totally. But it is not a popular method and since subtraction function can be done in adder by one bit complement and using 2x1 multiplexer as shown in Fig. 4. (F=A+ (-B)), both functions are implemented in one module. Table 2 shows truth table of these functions.

| A | B | Cin | Sum / Sub | Cout | Bout |
|---|---|---|---|---|---|
| 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 |

Table 2. Truth table of full adder-subtractor.

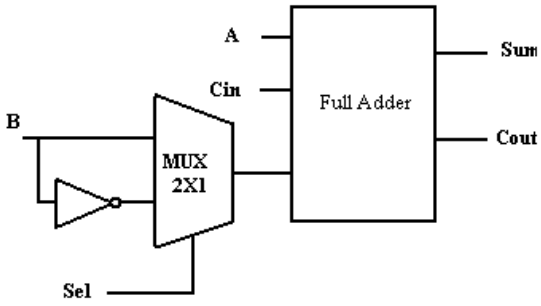


**Fig 4:** Full adder/subtractor block diagram.

If we design this circuit with Karnaugh method, it will results in a circuit with 10 gates and 58 transistors, But we design and optimized this module with ICA and resulting in a circuit with 4 gates and 39 transistors as shown in figure 5, so we achieved a circuit with approximately 38% reduction in space.

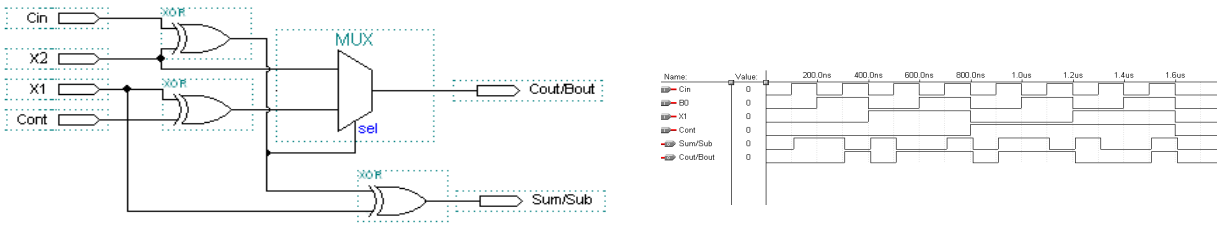


**Fig 5:** Full adder/subtractor circuit designed by ICA and wave form of inputs/outputs verified in MAXPLUS II.

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 Cin 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 with use n proposed module in series as shown in Fig 6.

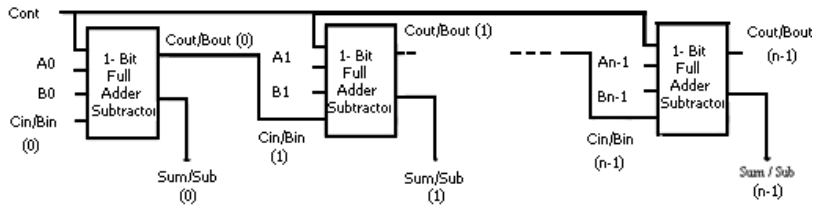


**Fig 6**: n-Bit full adder-Subtractor resulting from 1-bit optimized modules.

### 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 outputs simultaneously. Truth table of this module is shown in table 3.

| A | B | Cin | Sum Sub | Cout | Bout | 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 |

Table 3, Truth table of Multi-Function Module.

Our proposed circuit is shown in figure 7. After 20 times running ICA algorithm for this circuit, average iteration (945.12) with standard deviation (179.26) achieved.

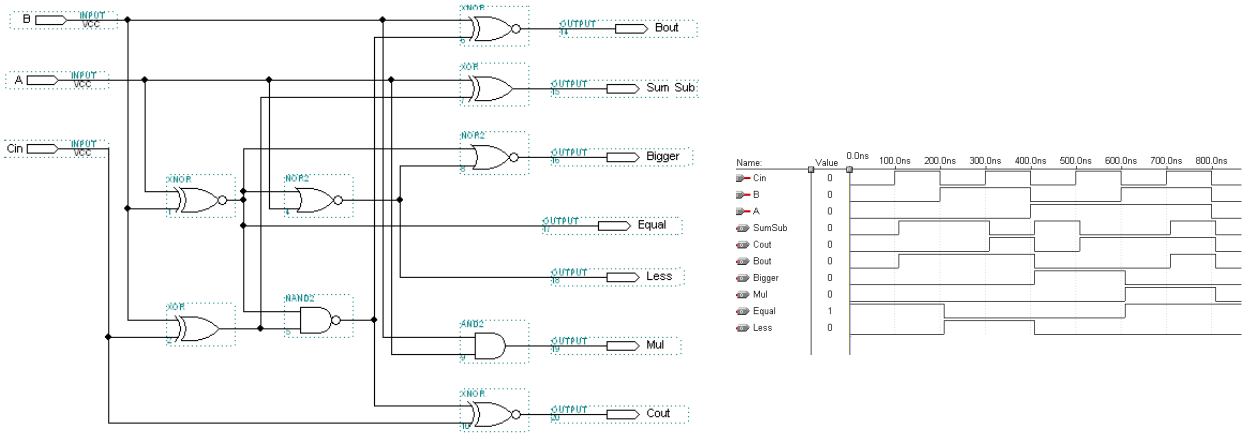

**Fig 7:** Multi-Function circuit designed by ICA and wave form of inputs/outputs verified in MAXPLUS II.

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

## 5. Conclusions

In this paper we proposed a novel method for designing and optimization of digital circuits based on Imperialist Competitive Algorithm which perform multi arithmetic functions and these circuits are very useful in ALU of microprocessors. Using these multi-function circuits instead of different distinct circuits can significantly reduce the area of whole ALU circuit.

## References

Atashpaz-Gargari, E. and Lucas, C. (2007). Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. *IEEE Congress on Evolutionary Computation (CEC 2007)*, 4661-6447.

Chen, C.Y., and Hwang, R.C. (2009). A new variable topology for evolutionary hardware design. *Expert Systems With Applications*. 36, 634-642.

Karnaugh, M. (1953). A map method for synthesis of combinational logic circuits, *Transactions of the AIEE, Communications and Electronics*. 72(1), 593- 599.

Louis, R. C. and Rawlins, G. J. E. (1991). Designer genetic algorithms: Genetic algorithms in structure design. *Fourth international conference on genetic algorithms. San Mateo, CA: Morgan Kauffman.* 53–60.

Lucas, C., Nasiri-Gheidari, Z. and Tootoonchian, F. (2010). Application of an imperialist competitive algorithm to the design of a linear induction motor. *Energy Conversion and Management.* 51 (7), 1407–1411.

McCluskey, E.J. (1956). Minimization of Boolean functions, *Bell Systems Technical Journal*, 35 (5), 1417-1444.

Mozafari, H., Abdi, B. and Ayob, A. (2010). Optimization of Transmission Conditions for Thin Interphase Layer Based on Imperialist Competitive Algorithm. *International Journal on Computer Science and Engineering (IJCSE).* 2 (7), 2486–2490.

Petley, G. *ASIC standard cell library design.* http://www.vlsitechnology.org/. (Accessed 5 October 2011).

Sayadnavard,H., Haghighat,A. and Abdechiri, M. (2010). Wireless sensor network localization using Imperialist Competitive Algorithm. *3rd IEEE International Conference on Computer Science and Information Technology*.

Yao, X., and Higuichi, T. (1999). Promises and Challenges of Evolvable Hardware. *IEEE Trans On Systems Man and Cybernetics, Part C*. 29, 87-97.