An Experimental Study Of Combinational Logic Circuit Minimization Using Firefly Algorithm

Aznilinda Zainodin[2], Aida Khairunnisaa Ab. Kadir[1], M. Nasir Ayob[1], Ahmad Fariz Hassan[1]
Amar Faiz Zainal Abidin[2], Fazlinashatul Suhaidah Zahid[2], Hazriq Izzuan Jaafar[3], Ismail Mohd Khairuddin[4]
[1]School of Mechatronic Engineering, Universiti Malaysia Perlis,
Pauh Putra Campus, 02600 Arau, Perlis, Malaysia
(E-mail: aidakhairunnisaa@yahoo.com, nasirayob@unimap.edu.my, farizhasan@unimap.edu.my)
[2]Faculty of Electrical Engineering, Universiti Teknologi MARA (UiTM)
Kampus Pasir Gudang, 81750 Masai, Johor, Malaysia
(E-mail: aznilinda@johor.uitm.edu.my, amarf408@johor.uitm.edu.my, fazlina7803@johor.uitm.edu.my)
[3]Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka,
76100 Durian Tunggal, Melaka, Malaysia
(E-mail: hazriq@utem.edu.my)
[4]Faculty of Manufacturing Engineering, Universiti Malaysia Pahang,
26600 Pekan, Pahang, Malaysia
(E-mail: ismailkhai@ump.edu.my)

Abstract - Combinatorial logic circuit minimization is usually done using Karnaugh's Map or Boolean equation. This paper presents an application of Firefly Algorithm to design combinational logic circuit in which the objective function is to minimize the total number of gates used. Then, the algorithm is benchmarked with other literatures. Result indicates that it able to find optimal solution but further analysis is required for a more complex combinatorial logic circuit minimization.

Keywords - Combinational logic circuit minimization; Computational intelligence; Firefly algorithm; Numbers of gates; Swarm intelligence.

1. Introduction

A minimized combinatorial logic circuit brings a lot benefits to the electronic industries. Based on common sensea minimized combinatorial circuit design should bring at least one out of these four benefits: smaller circuit board, reduces propagation, minimizes error, and reduce cost. Thus, it is essential to ensure any circuit design takes into account the minimization of the number gates used.

Nowadays, there are a lot of ways to minimize the design combinatorial logic circuit. Based on the work done by other researchers, they had proposed different model and algorithms to solve the problem. One of them is using Genetic Algorithm (GA) in design the combinatorial logic circuit. This method proposed by S. J. Louis and G. J. E. Rawlins. Two dimensional array structures (phenotype) will be map for each genotype. This array will be presented as circuit design [1].

On the other hands, X.Wang *et.al* [2] proposed an approach to enhance the designing process by using variable topology cartesian genetic programming for combinational circuit. In this paper it used a combination of GA Evolvable Hardware (EHW), Chromosome Genetic Programming (CGP) and Variable Cartesian Genetic Programming (VCGP). Moreover, it also increases the complexity of space search using random circuit matrix. Besides that, they used symbols to represent a direction such as "*" for horizontal direction and "+" for vertical direction. These symbols were also been used by G.Papa and his co-workers in their binary tree expression [3].

In addition the same method can be used in designing the combinational logic circuit with different algorithms such as GA, Ant Colony System (ACS), and Particle Swarm Optimization (PSO). The model proposes a matrix to represent circuit logic, which is bi-dimensional space, consists of a string of combination gates inputs, gates type, gates outputs. The encoded of the matrix elements for gate type consist of 4 types of gates: AND, NOT, OR, XOR, and WIRE [4].

This paper proposes the application of Firefly Algorithm (FA) in minimizing the number of gates of the combinatorial logic circuit. The model is adapted from A.Tyrell *et.al* [4] which makes it suitable for direct benchmarking between the algorithms.

2. Combinatorial Logic Circuit Minimization

All of the modern computers consist of logic gates which functioning as a building block of a digital circuit. The computers' microchips are arranged in hundred of combinational logical gates to make sure the efficiency and consistently reliable outputs. Even though, there are no limitation in the number of gates that can be used and with the advancement of microchip technology; the designers need to minimize the amount of logic gate [5].

The case study taken for implementing Firefly Algorithm in minimizing the combinatorial logic circuit's gates was referred based on study by C. A. Coello *et al.*

[4]. The truth table is as shown in Table 1. From a work done by Tyrell *et.al* [4] they found that by using PSO-based approach it is able to find the solution by using only 4 gates (2 ANDs, 1 OR, 1 XOR) as stated in equation (1) below.

Table 1 Truth table for Combinational Logic Circuit Minimization using Firefly Algorithm

Input			Output
x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	1	0	1
1	1	1	0
1	0	0	0
1	0	1	1

$$F = ((y + z)x)'yz + ((y + z)x)(yz)' \quad (1)$$

3. Implementation of Firefly Algorithm

Firefly Algorithm was proposed by X. S. Yang *et.al* as another SI algorithm that can compete with PSO [6]. The main motivation for using Firefly Algorithm (FA) to design combinational logic circuit is that this algorithm has been found very efficient in a variety of tasks such as solving Traveling Salesman Problem (TSP) [7], routing Very Large Scale Integration (VLSI) circuit [8] and path optimization in PCB holes drilling process [9].

In implementing any Swarn Intelligence (SI) algorithms, the most importance step is to model the problem so that the algorithm can fit the problem. Generally, firefly positions in the search space represent a candidate solution of the problem. A candidate solution should be a feasible solution where feasible solution is solution that produces the desired output according to the truth table. The firefly position x can be translated into a d-dimension of search space where each dimension represents a part of the candidate solution. For this approach, we adopted the model proposed by [4] which can be translated in equation (2) below.

$$x = [Gate1'sID, Gate2'sID, \dots, Gated'sID] \quad (2)$$

As rule of thumb, the number of dimension is dictated by equation (3) which was proposed by Sadiq M. Sait *et.al* [10] where n_i is the number of input in logic circuit.

$$d = {n_i}^2 \quad (3)$$

Thus, for a 3 inputs logic circuit, the number of dimensions required is 9.

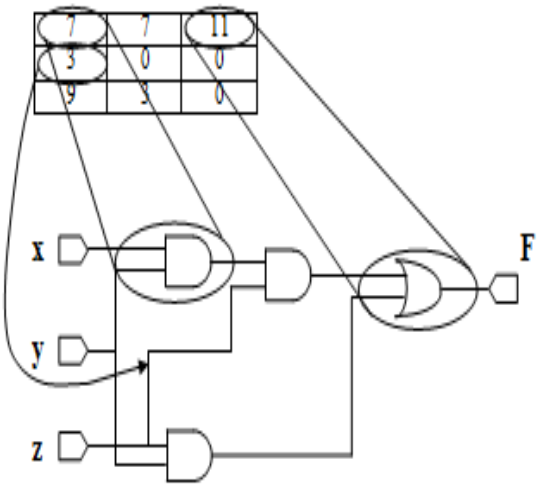

Fig.1. Example of a circuit and its encoding

As stated earlier, the model taken from [4] where each Gate ID can be an integer number range from 0 to 15. Each number has unique combinations of inputs and gate type which presented in Table 2 below.

Table 2 Gate ID and gate types [4]

Gate ID	Inputs	Gate	Output
0	None	None	None
1	x	WIRE1	x
2	y	WIRE2	y
3	z	WIRE3	z
4	x	NOT1	~ x
5	y	NOT2	~ y
6	z	NOT3	~ z
7	x, y	AND1	x .y
8	x, z	AND2	x .z
9	y, z	AND3	y . z
10	x, y	OR1	x + y
11	x, z	OR2	x + z
12	y, z	OR3	y + z
13	x, y	EX-OR1	~x.y + x.~y
14	x, z	EX-OR2	~x.z + x.~z
15	y, z	EX-OR3	~y.z + y.~z

Note that input for 2nd and higher level is based on the input of first level where row 1 is input a, row 2 is input b while row 3 is input c and so on.

3.1 Fitness formulation of Firefly Algorithm

The fitness formulation for FA is the total gates used. The smaller the number of the gates used the better the fitness of that firefly.

Figure 2 displays the flowchart of the implementation of FA in solving the combinatorial logic circuit minimization problem. The program starts by initializing the FA parameters. At the same time, the fireflies' positions are randomly generated. Only feasible solutions are accepted, if there any firefly represented unfeasible solution, the position of the firefly is randomly generated again until it found a feasible solution. Then, the fitness of each firefly is evaluated.

Next, the intensity of each firefly is calculated using formula stated in [6]. After that, the intensity of between two fireflies are compared, firefly with better intensity will attract the firefly with lower intensity towards it. The new position of the firefly will be check if it represents a feasible solution. If the new position is a feasible solution, the new solution is accepted, otherwise, the firefly will remained at the old position. Note that during updating the firefly position, two conditions might occur.

First, out of boundary condition in which the new position of the firefly is outside the desired range, in this case, the range is 0 to 15. If this happened, the modulus after division by 16 of the new position is taken as the new solution. For example, 17 will become 2 and -1 will become 15. Another condition that might occur frequently is the new position is a floating number. For this condition, the new solution will be the rounding of the new position. For example, 3.42 will be 3 while 9.91 become 10.The process will be repeated until the maximum iteration reached.

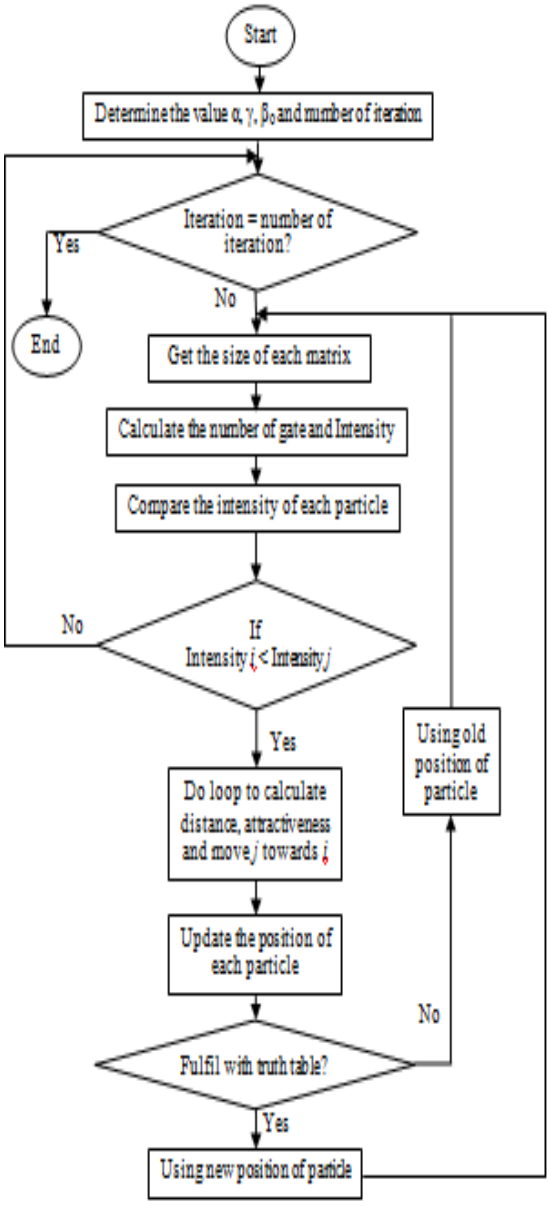

Fig.2. Flowchart of the implementation of FA in solving the combinatorial logic circuit minimization problem.

4. Result and Discussion

4.1 Result and Analysis

The approach is written using MatlabR2010a and run on a laptop with Pentium(R) Dual-Core CPU, 1GB of RAM, 32- bit of operating system and 2.0 GHz of speed. Table 3 stated the parameters used in FA & PSO [4] for the implementation of the proposed approach. Table 4 shows the result obtained from the proposed approach. It can clearly be seen that FA able to find the optimal solution [4].

Table 3 Comparison of the PSO Parameters used by in previous research with this study

	PSO	FA
Common Parameters		
Number of agents, q	90	3
Number of iteration, t	300	15000
Number of computations	20	20
PSO Parameters		
Inertia weight, ω	0.8	Not Applicable
Cognitive component, c_1	Not Applicable	Not Applicable
Social component, c_2	Not Applicable	Not Applicable
r_1 and r_2	Not Applicable	No. of gates for each circuit
FA Parameters		
Attractiveness, β_0	Not Applicable	1
Randomization parameter, α	Not Applicable	0.7
Absorption coefficient, γ	Not Applicable	0.1

Table 4 Comparisons of FA results

PSO [4]	Human Designer 1 [4]
F = ((y + z)x)'yz + ((y + z)x)(yz)' 4 gates 2 ANDs, 1 OR, 1 EXOR	F = x'yz + xy'z + xyz' 11 gates 3 NOTs, 6 ANDs, 2 ORs
FA	Human Designer 1 [12]
F = ((y + z)x)'yz + ((y + z)x)(yz)' 4 gates 2 ANDs, 1 OR, 1 EXOR	F = x'yz + x(x'z + xz') 6 gates 3 ANDs, 1 OR, 1 EXOR, 1 NOT

Figure 4 is the encoded circuit after the simulation of FA using project modeling. From the encoded circuit, project is able to display the circuit using different logic gates. It is shows that circuit has 2 AND gates, 1 OR gate, 1 EXOR gate and 3 wires.

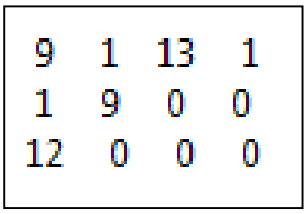

Fig.4. Encoded circuit from the project simulation

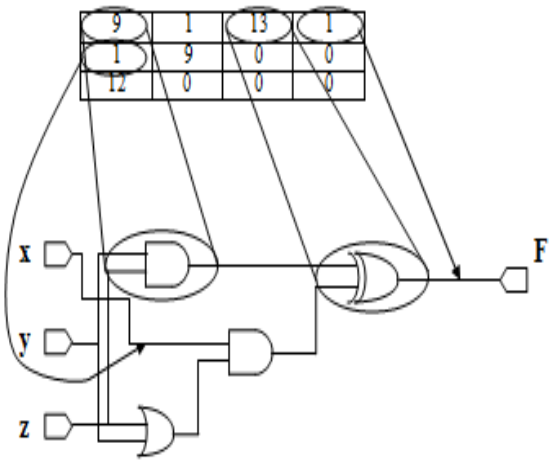

Fig.5. Circuit from the result

5. Conclusion

This paper presents an approach for minimizing number of gates for combinational logic circuit using Firefly Algorithm. The proposed approach is explained extensively and the experimental result also has proved that FA is an efficient algorithm.

Acknowledgment

A special thank to my supervisor Mohd Nasir bin Ayob and for his advice, support and guidance throughout this research.

References

[1] S. J. Louis and G. J. Rawlins, Using Genetic Algorithms to Design Structures, 1991.

[2] W. Xiuqin, W. Hao, and M. Guangsheng, "Variable Topology Cartesian Genetic Programming for Combinational Circuit," in Fourth International Conference on Natural Computation 2008 (ICNC 2008), pp. 306-310, 2008.

[3] G. Papa and J. Å ilc, "Automatic large-scale integrated circuit synthesis using allocation-based scheduling algorithm," Microprocessors and Microsystems, Vol. 26, pp. 139-147, 2002.

[4] A. Tyrrell, P. Haddow, J. Torresen, C. Coello Coello, E.n. Luna, and A.n. Aguirre, "Use of Particle Swarm Optimization to Design Combinational Logic Circuits," in Evolvable

Systems: From Biology to Hardware. Springer Berlin Heidelberg, Vol. 2606: pp. 398-409, 2003.

[5] J. F. Wakerly, Digital Design Principles & Practices: Upper Saddle River, New Jersey Prentice Hall, 1999.

[6] X.S.Yang, Firefly algorithm for multimodal optimization. Stochastic Algorithms: Foundation and Application, Vol. 5th, 2009.

[7] S. F. M. Hussein, A. FaizZainalAbidin and M. AsrafMansor, "A Preliminary Study on Firefly Algorithm Approach for Travelling Salesman Problem," in Science & Engineering Technology National Conference, 2013.

[8] M. N. Ayob, F. Hassan, and A. H. Ismai, "A Firefly Algorithm Approach for Routing in VLSI," in International Symposium on Computer Applications and Industrial Electronics, pp. 98-102, 2012.

[9] M. M. Ismail, M. A. Othman, H. A. Sulaiman, M. H. Misran, R. H. Ramlee, A. F. Z. Abidin, N. A. Nordin, M. I. Zakaria, M. N. Ayob, and F. Yakop, "Firefly algorithm for path optimization in PCB holes drilling process," in Green and Ubiquitous Technology (GUT), 2012 International Conference on, pp. 110-113, 2012.

[10] M. Sadiq M. Sait, Abd-El-Barr, U. Al-Saiari, and B. A. B. Sarif, "Fuzzified Simulated Evolution Algorithm For Combinational Digital Logic Design Targeting Multi-Objective Optimization," in *IEEE International Symposium on Circuits and Systems, ISCAS*, 2004.