Digital combinational circuit optimization using invasive weed optimization technique

Prabhat K. Patnaik[1], Dhruba C. Panda [2], Santosh Kumar Pantina[1]
[1] Department of Electronics and communication Engineering, School of Engineering and Technology, Centurion University of Technology and Management, Paralakhemundi, India
[2]Department of Electronics Science, Berhampur University, Berhampur, India

E-mail: prabhat.ku.patnaik@gmail.com

(Received 31 January 2013, accepted 27 may 2014)

Abstract
Minimization of digital circuits is required to reduce the area, power consumption, propagation delay and number of active gates. Human methods of minimization like Karnaugh map, Quine' McCluskey, Sasao methods are tedious and are limited to systems with four or five inputs. A new bio-inspired algorithm called Invasive weed optimization (IWO) is used to minimize the combinational circuits. Results are presented to show that IWO based optimization of digital circuits are equivalent to or even with better solution than human design techniques.

Keywords: Combinational logic circuits, Karnaugh Map, Invasive Weed Optimization algorithm.

Resumen
Se requiere minimización de circuitos digitales para reducir el área, el consumo de energía, el retardo de propagación y también el número de puertos activos. Los métodos humanos de minimización como el mapa de Karnaugh, y los métodos de Quine McCluskey y de Sasao son tediosos, y se limitan a sistemas con cuatro o cinco entradas. Un nuevo algoritmo bio-inspirado llamado optimización maleza invasora (OIV) se utiliza para minimizar los circuitos combinacionales. Se presentan los resultados para mostrar que la optimización de circuitos digitales basada en OIV es equivalente o incluso una mejor solución mejor que las técnicas diseñadas por humanos.

Palabras clave: Circuitos lógicos combinacionales, mapa de Karnaugh, algoritmo Optimización de Maleza Invasora.

PACS: 01.40.Fk, 01.40.E- **ISSN 1870-9095**

I. INTRODUCCIÓN

Real world applications require combinational circuits with minimal area, power consumption, propagation delay and minimum number of gates for cost effective and high speed circuit realization. Generally used methods to minimize the combinational circuits are Karnaugh Map [1], Quine' McCluskey [2, 3], Sasao [4] etc. The problem with the human design methods is that they become cumbersome and problematic when number of inputs, number of outputs and the complexity of the function increases.

Process of minimization can be viewed as an optimization process wherein digital circuits seek a best possible/desirable solution for a physical model *i.e.*, combinational circuit. Hence, in order to reduce the problems faced in human design methods, minimization of combinational circuits was done through computational intelligence or more precisely with the use of bio inspired optimization algorithms like Genetic Algorithm(GA) [5], Particle Swarm Optimization(PSO) [6], a hybrid algorithm called Differential Evolution Particle Swarm Optimization(DEPSO) [7] etc. Computational intelligence methods find the significant advantage of being automated through programming over human design methods.

A new technique called Invasive Weed Optimization (IWO) is used to minimize the combinational circuits. IWO is also a bio inspired optimization algorithm which unlike GA and PSO which are Evolutionary and Swarm based algorithms respectively it is an ecology based bio inspired algorithm. IWO is inspired from natural ecosystem which provides rich source of mechanism for designing and solving difficult engineering and computer science problems. The optimization of combinational circuits using IWO is presented and preliminary investigation show that IWO can optimize equally well as other algorithms like GA, PSO, and DEPSO etc. The paper is organized as follows: In section II a brief overview of Invasive weed optimization is given. Section III describes minimization of combinational circuits with IWO. In section IV examples of optimized combinational circuits using IWO are presented and section V compares against human design techniques.

II. INVASIVE WEED OPTIMIZATION ALGORITHM

Invasive Weed Optimization is a numerical stochastic search algorithm proposed by Mehrabian and Lucas in 2006 [8], inspired by the ecological process of weed colonization and distribution. It is capable of solving general multi-dimensional, linear and nonlinear optimization problems with appreciable efficiency.

Weeds are plants whose vigorous, invasive habits of growth pose serious threat to desirable plants. Adapting with their environments, invasive weeds cover spaces of opportunity left behind by improper tillage; followed by enduring occupation of the field. Their behavior changes with time since as the colony become dense there is lesser opportunity of life for the ones with lesser fitness.

Fast reproduction and distribution, robustness and adaptation to the changes in the environment are some of the interesting characteristics have been seen in natural behavior of weeds that have inspired and used in this optimization algorithm. This algorithm has additional desirable properties of capability to deal with complex and non-differentiable objective functions and escapes from local optima.

The algorithm can be summarized in the following four steps [8]:

1. *Initialize a population*: A finite number of seeds composing initial population are being dispread randomly over the problem space.
2. *Reproduction*: Every seed that has grown to new plants is allowed to produce other depending on its fitness. In the simple case, the number of seeds each plant can produce increases linearly from minimum possible seed corresponding to minimum fitness to the maximum number of seeds corresponding to the maximum fitness in the population as illustrated in Figure1.

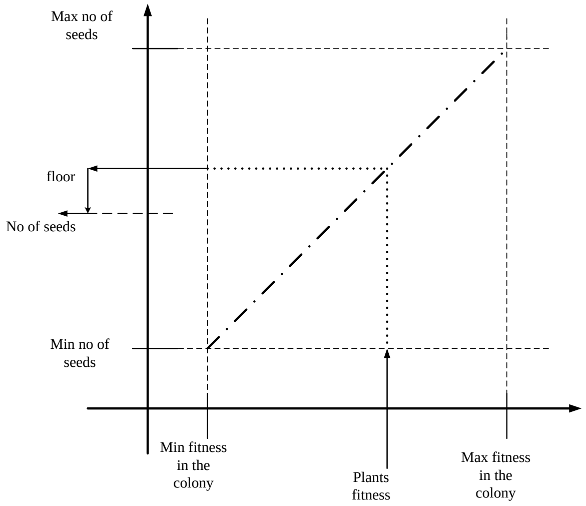

FIGURE 1. Procedure of seeds production in a colony of weeds [7].

3. *Spatial dispersal*: The produced seeds in the previous step are being distributed randomly in the problem space by normal distribution with mean zero and a variance parameter decreasing over time. By setting the mean parameter equal to zero, the seeds are distributed randomly such that they locate near to the parent plant and by decreasing the variance over time, the fitter plants are grouped together and inappropriate plants are eliminated over time. The standard deviation (SD) which is the root square of the variance of this distribution is calculated in every time step as according to (1):

$$\sigma_{iter} = \frac{(iter_{\max} - iter)^n}{(iter_{\max})^n}(\sigma_{init} - \sigma_{final}) + \sigma_{final} \qquad (1)$$

   σ_{init} and σ_{final} are initial and final value of SD for normal distribution respectively, $iter_{\max}$ is the maximum number of iterations before stopping the algorithm, σ_{iter} is the SD present at the present time step and n is the nonlinear modulation index.

4. *Competitive exclusion*: After some iteration, the number of plants in a colony will reach its maximum (pmax) by fast reproduction. However, it is expected that the fitter plants have been reproduced more than undesirable ones. Thus, final step is to eliminate the inappropriate and weaker plants in a competitive manner for limiting the maximum number of plants in a colony. The process continues until maximum iterations or some other stopping criteria is reached and the plant with the best fitness is selected as the optimal solution.

The flow chart depicting IWO algorithm is shown in the Fig.2.

III. OPTIMIZATION OF COMBINATIONAL DIGITAL CIRCUITS WITH IWO

Invasive weed optimization theory described above is used to evolve combinational logic circuits. The basic process of hardware evolution is illustrated in Figure 3. The desired circuit refers to the circuit required to map 100 % exactly the outputs for corresponding inputs typically given by truth table for digital circuits. After each generation, the fitness is evaluated against the desired function to be implemented, given by the truth table. If the output of the circuit is equal to the output of the truth table for the corresponding inputs, then the fitness is increased by one. This is carried out for all inputs listed in the truth table. This process is repeated till we get a weed with the fitness equal to total number of combinations in the truth table for the particular combinational digital circuit under study or till maximum number of iterations is reached. The hardware evolution is carried out until the desired circuit is evolved and then downloaded to a reconfigurable hardware platform.

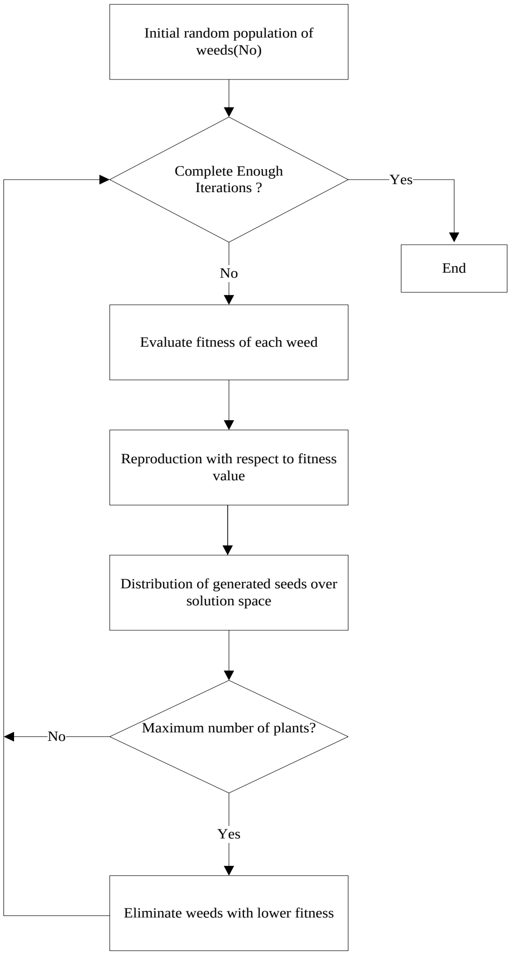
Initial random population of weeds(No)
Complete Enough Iterations ?
Yes
End
No
Evaluate fitness of each weed
Reproduction with respect to fitness value
Distribution of generated seeds over solution space
Maximum number of plants?
No
Yes
Eliminate weeds with lower fitness

FIGURE 2. Flow chart depicting IWO algorithm [9].

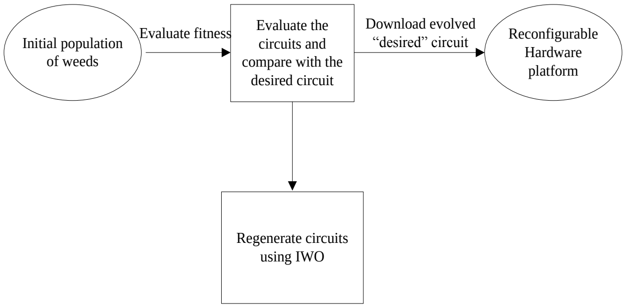
Initial population of weeds
Evaluate fitness
Evaluate the circuits and compare with the desired circuit
Download evolved "desired" circuit
Reconfigurable Hardware platform
Regenerate circuits using IWO

FIGURE 3. Desired circuit hardware evolution.

A. Coding of input matrix

The matrix shown in Figure 4 represents a circuit with M rows and N columns [6]. The elements of the circuit are the logic gates which are selected from a predefined library of 1 or 2-input and 1-output gates. The inputs to the first column of the matrix come from the truth table of the function to be implemented. For all other columns, the input may come from any of the previous column outputs.

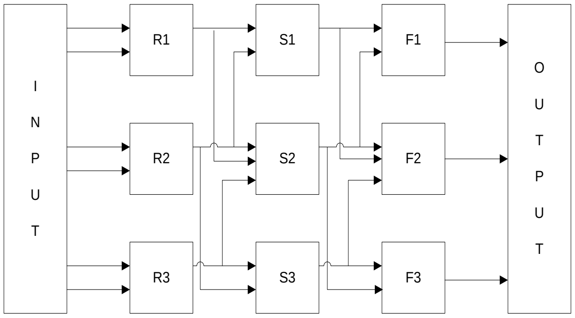

FIGURE 4. Structure of random matrix [6].

For circuit evolution with IWO one matrix is used to represent gates/inputs interconnectivity. The size of the matrix can be taken as g by 3 where g represents the total number of gates in input matrix. Elements in first and third column represent the inputs while the elements in the second column represent the gates. Gates are represented as: AND=1, OR=2, XOR=3, NOT=4 and WIRE=5, the inputs are as well represented for convenience as follows; A=1, B=2, C=3, R1=4, R2=5, R3=6, S1=7, S2=8, S3=9. (R1, R2 & R3) first column gate output, (S1, S2 & S3) second column gate output and F1=F2=F3=FOUT third column output.

B. Implementation of IWO algorithm to combinational digital circuits

1. Initial population of weeds (input matrices) N0 are generated randomly but depending on the constraints based on "coding the input matrix" as discussed previously.
2. Fitness of each weed is evaluated. Fitness of a weed is total number of output combinations of a truth table that matches with the outputs of weed for each particular input combination.
3. Depending on the fitness number of seeds is generated. A maximum number of seeds Smax for maximum fitness and minimum number of seeds Smin corresponding to minimum fitness.
4. Now the generated seeds corresponding to each input matrix are randomly scattered by normal distribution with mean equal to zero and an adaptive standard deviation(SD) given by (1). Produced seeds along with the parents are considered as potential solutions for next generation.
5. If the maximum number of plants Pmax is reached then the weeds with lower fitness is eliminated.

6. Above steps from 2-5 are continued till either of the following conditions is met:
    a. Input matrix (weed) with the highest fitness (total number of input and output combinations in the truth table) is reached. This matrix is the optimizd matrix that is obtained by the algorithm.
    b. Total number of iterations is reached.

A MATLAB program based on the coding of the input matrix discussed in section IIIA and on the IWO implementation for combinational circuits discussed in section IIIB is coded and simulated for the implementation of the IWO algorithm for minimization of the digital circuits.

IV. EXAMPLES OF EVOLVED CIRCUITS

Three examples are presented here to describe the capability of IWO for hardware evolution. The size of the input matrix for these examples is taken as 7 by 3. The IWO parameter values used for optimization of all the three examples is listed in the Table I.

TABLE I. IWO parameter values for optimization of combinational circuits.

Sl.No	*Symbol*	*IWO parameter*	*Value*
1	N_0	Number of initial population	5
2	It_{max}	Maximum number of iterations	2000
3	P_{max}	Maximum number of plant population	10
4	S_{max}	Maximum number of seeds	7
5	S_{min}	Minimum number of seeds	1
6	n	Nonlinear modulation index	1
7	$\sigma_{initial}$	Initial value of standard deviation	1
8	σ_{final}	Final value of standard deviation	0.1

A. 3-even parity problem implementation by IWO

3-even parity problem has three inputs, one output. The truth table for the circuit is shown in Table II. The evolved circuit satisfying the desired circuit is expected to have a fitness of eight in this case. Input matrix of size 7 by 3 is taken with the IWO parameters as mentioned in the Table I. The simplified expression for function F using Karnaugh map (human design method) is shown in (2).

$$F = Z(X \oplus Y) + Y(X \oplus Z)\,. \tag{2}$$

The circuit obtained by Karnaugh map consists of 5 gates *i.e*. two XOR gates, two AND gates and one OR gate as shown in Figure 5. The circuit obtained by using IWO algorithm for the output F, having a fitness of eight obtained after 717 iterations consists of one XOR gate, two AND gates, one OR gate as shown in Fig. 5.

TABLE II. Truth table for 3-even parity generator.

a	*b*	*c*	*F*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

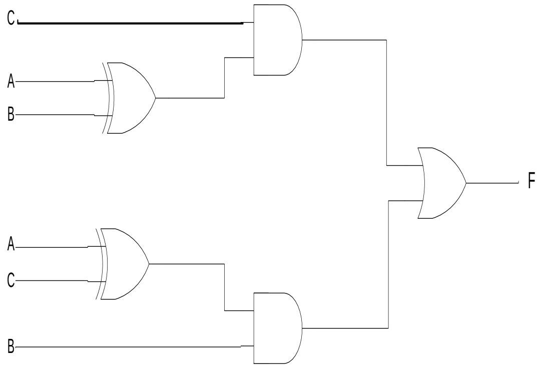

FIGURE 5. Circuit obtained by Karnaugh map for 3-even parity problem.

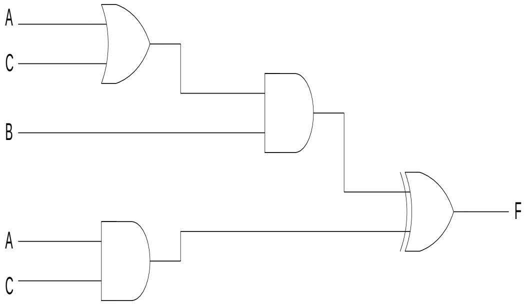

FIGURE 6. Circuit optimized by IWO for 3-even parity problem.

B. Full Adder implementation by IWO

A full adder is a combinational circuit that can be used to add three bits to produce a sum and a carry output [11]. It consists of three inputs, two outputs. Truth table for full adder is shown in the Table III. The evolved circuit satisfying the desired circuit is expected to have a fitness of eight for both the outputs sum and carry. Input matrix of size 7 by 3 is taken with the IWO parameters as mentioned in the Table I. The simplified expressions for *sum* and *carry* using Karnaugh map are shown in (3) and (4) respectively.

$$sum = c \oplus (a \oplus b)\,, \tag{3}$$

$$carry = ab + ac + bc. \tag{4}$$

The circuit obtained by Karnaugh map consists of two XOR gates, three AND gates and one OR gate [10] as shown in Fig. 7. The circuits obtained by using IWO algorithm for the output *sum* having a fitness of eight obtained after 5 iterations consists of two XOR gates as shown in Fig. 8. The circuits obtained by using IWO algorithm for the output *carry* having a fitness of eight obtained after 4 iterations consists of four XOR gates and one OR gate as shown in Fig. 9. The final circuit including both *sum* and *carry* for full adder using IWO consists of four XOR gates and one OR gate as shown in Fig. 10.

TABLE III. Truth table for full adder.

a	*b*	*c*	*carry*	*sum*
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

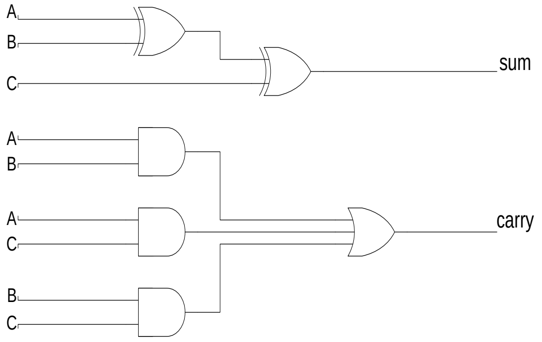

FIGURE 7. Circuit for full adder using Karnaugh map [10].

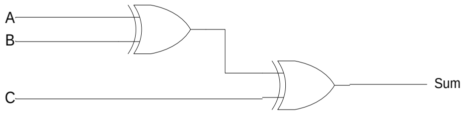

FIGURE 8. Sum circuit for the full adder using IWO.

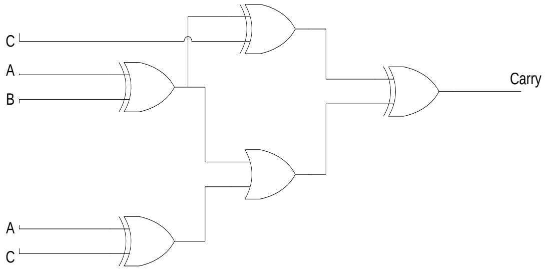

FIGURE 9. Carry for the full adder circuit obtained through IWO.

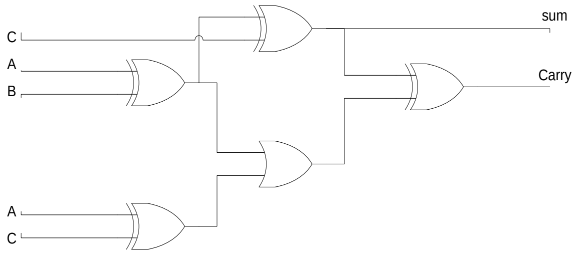

FIGURE 10. Circuit optimized by IWO for full adder.

C. Full subtractor implementation by IWO

A full subtractor performs subtraction operation on two bits, a minuend and a subtrahend, and also takes into consideration whether a ‘1’ has already been borrowed by the previous adjacent lower minuend bit or not. As a result, there are three bits to be handled at the input of a full subtractor, namely the two bits to be subtracted and a borrow bit designated as C. There are two outputs, namely the *difference* output and the *borrow* output. The *borrow* output bit tells whether the minuend bit needs to borrow a ‘1’ from the next possible higher minuend bit. The truth table for the circuit is shown in Table III.

The evolved circuit satisfying the desired circuit is expected to have a fitness of eight for both the outputs *difference* and *borrow*. Input matrix of size 7 by 3 is taken with the IWO parameters as mentioned in the Table I. The simplified expressions for difference and borrow using Karnaugh map are shown in (5) and (6) respectively.

$$difference = c \oplus (a \oplus b), \tag{5}$$

$$borrow = \bar{a}b + (\overline{a \oplus b})c. \tag{6}$$

The circuit obtained by Karnaugh map consists of two XOR, two AND, one OR and two NOT gates [12] as shown in Fig. 11. The circuit obtained by using IWO algorithm for the output difference having a fitness of eight obtained after 5 iterations consists of two XOR gates as shown in Figure 12.

The circuit obtained by using IWO algorithm for the output *borrow* having a fitness of eight obtained after 127 iterations consists of four XOR gates and one OR gate as shown in Fig. 13. The final circuit including both *difference* and *borrow* for full subtractor using IWO consists of four XOR gates and one OR gate as shown in Fig. 14.

TABLE IV. Truth table for full subtractor.

a	*b*	*c*	*Borrow*	*Difference*
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

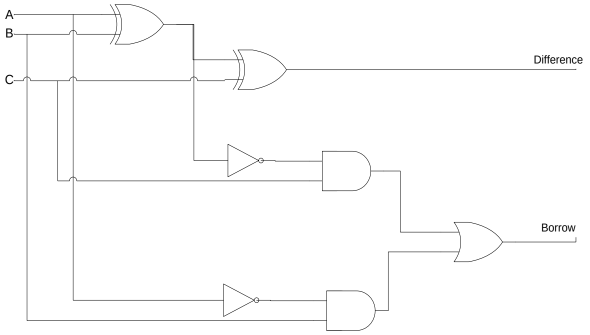

FIGURE 11. Circuit for full subtractor using Karnaugh map [12].

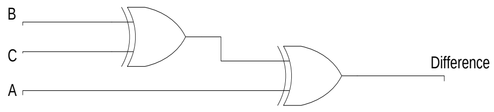

FIGURE 12..Difference circuit for the full subtractor using IWO.

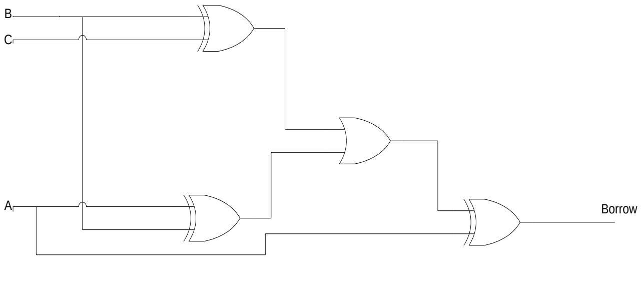

FIGURE 13. Borrow circuit for the full subtractor using IWO.

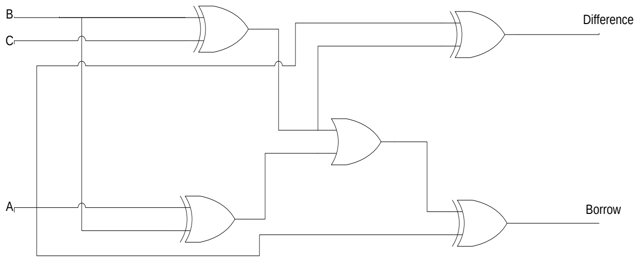

FIGURE 14. Circuit optimized by IWO for full subtractor.

V. COMPARISON OF RESULTS

Results obtained for 3-even parity problem: Full adder, Full Subtractor through IWO are compared with the Karnaugh maps (human design), as shown in the Table V in terms of number of logic gates.

TABLE V. Comparison of results of IWO with Human design for different examples.

Examples	*Karnaugh map (human design)*	*IWO*
3-even parity problem	5 gates	4 gates
	2 XOR, 2 AND, 1 OR	1 OR, 2 AND, 1 XOR
Full adder	6 gates	5 gates
	2 XOR, 3 AND, 1 OR	4 XOR, 1 OR
Full subtractor	7 gates	5 gates
	2 XOR, 2 AND, 2 NOT, 1 OR	4 XOR , 1 OR

VI. CONCLUSIONS

In this paper it is shown that the invasive weed optimization can also be applied to evolve combinational logic circuit.

The emphasis was only on generation of circuit functionality. From the few examples carried out we infer that IWO approach is an improvement over human design method because it has minimum number of gates as summarized in Table V. It is clear that Invasive Weed Optimization has a potential for hardware evolution since it has faster convergence and is able to minimize the total number of active gates.

REFERENCES

[1] Karnaugh, M., *A map method for synthesis of combinational logic circuits,* Transactions of the AIEE, Communications and Electronics **721**, 593-599 (1953).

[2] McCluskey, E., *Minimization of boolean functions,* Bell Systems Technical Journal **35**, 1417-1444 (1956).

[3] Quine, W. V., *A way to Simplify Truth Functions*, American Mathematical Monthly **62**, 627-631 (1955).

[4] Sasao, T., *Logic Synthesis and Optimization*, (Kluwer Academic Press, 1993).

[5] Higuchi T., Iba H. and Manderick T., *Evolvable hardware with genetic learning*. *Massively Parallel Artificial Intelligence*, (MIT press, USA, 1994).

[6] Gudise V. G. and Venayagamoorthy G. K., *Evolving Digital Circuits Using Particle Swarm*, Proc. Of INNS-IEEE Intl. Joint Conf. on Neural Networks, 468-472 (2003).

[7] Moore, P. K. and Venayagamoorthy G. K., *Evolving Digital circuits using Hybrid Particle Swarm Optimization and Differential Evolution*, International Journal of Neural Systems **16**, 163-177 (2006).

[8] Mehrabian, R., and Lucas, C., *A novel numerical optimization algorithm inspired from invasive weed colonization*, Ecological Informatics **1**, 355-366 (2006).

[9] Mallahzadeh, A. R., Es'haghi, S., and Alipour, A., *Design of an E-shaped MIMO antenna using IWO algorithm for wireless application at 5.8 GHz*, Progress In Electromagnetics Research **90**, 187-203 (2009).

[10] Ushie Ogri, J., Etim O. J. A. and Prosper I., *Optimising digital combinational circuit using particle swarm optimisation technique*, Lat. Am. J. Phys. Educ. **6**, 72-77 (2012).

[11] Kumar A., *Fundamentals of Digital Circuits*, 2nd Edition, (PHI Learning Private Limited, New Delhi, India).

[12] Mano, M., *Digital Design*, 4th Edition (Pearson Education, India).