A Faster 2D Technique for the design of Combinational Digital Circuits Using Genetic Algorithm

Vijayakumari. C.K
Department of Electrical Engineering
Rajiv Gandhi Institute of Technology
Kottayam, Kerala, India
vijayakumari@rit.ac.in

Mythili. P
Div. of Electronics, School of Engineering
Cochin University of Science and Technology
Kochi, Kerala, India
mythili@cusat.ac.in

Abstract— **Combinational digital circuits can be evolved automatically using Genetic Algorithms (GA). Until recently this technique used linear chromosomes and and one dimensional crossover and mutation operators. In this paper, a new method for representing combinational digital circuits as 2 Dimensional (2D) chromosomes and suitable 2D crossover and mutation techniques has been proposed. By using this method, the convergence speed of GA can be increased significantly compared to the conventional methods. Moreover, the 2D representation and crossover operation provides the designer with better visualization of the evolved circuits. In addition to this, a technique to display automatically the evolved circuits has been developed with the help of MATLAB.**

Keywords-GeneticAlgorithm;evolvablehardware; combinational digital circuits; 2D crossover; 2D mutation

IV. INTRODUCTION

Genetic algorithm (GA) is a powerful tool which can be used for many of the optimization problems. Conventional GA takes in a linear set of chromosomes on which one dimensional crossover and mutation operators are applied to get an optimal solution. The speed of convergence and the quality of the solutions are influenced by these operators. Hence it is very essential to improve these genetic operators to give a better performance.

Majority of electronic systems with any level of complexity were designed by specialized designers having a complete knowledge of operation of the system and designing rules. Such design methods are obviously limited by the experience and knowledge of the designer [1]. In evolutionary design, knowledge and experience of the designer are replaced by evolution process which has comparatively less constraints. Designers are not only limited by technology but also by their own habits imagination and creative thinking [2]. By applying evolutionary methods to the design of digital circuits, the above mentioned constraints do not come into picture. The main goal of evolutionary algorithm is to evolve a functional circuit with the least possible complexity.

Some of the earlier approaches were focused on evolving a 100% functional circuit, while others focused on arriving at a 100% functional circuit with minimum number of gates [3].The two most popular methods K-Map and Quine-Mc-Cluskey methods do not support the use of XOR/XNOR gates for the design of digital circuits [4]. But the use of XOR gates in arithmetic circuits which is possible with evolutionary design reduces the number of gates in the evolved circuits.

While designing circuits using GA, the combinational circuit is usually represented by a two dimensional array in which each cell represents two inputs and the corresponding gate together which in turn represents a gene. The digital circuit is assumed to be embedded in an n x n array of cells where n represents the number of levels in the circuit and the number of gates in each level.

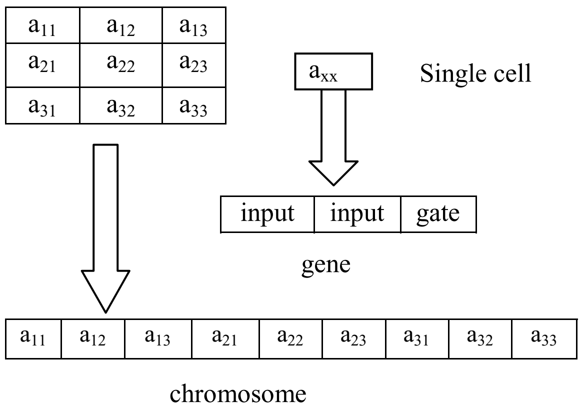


Figure 1. Representation used for encoding matrix form to linear form

Fig.1 shows a combinational digital circuit represented by a 3x3 array of cells and the corresponding chromosome. For two input gates a chromosomal string encodes the matrix by using triplets in which the first two elements refer to each of the inputs used and the third is the corresponding gate [5]. Each cell receives its input from the previous levels only, the inputs to the first set of cells are obtained from the truth table [3].

978-1-4673-0449-8/12/$31.00 ©2012 IEEE

With the one dimensional chromosomal representation, crossover and mutation technique, the computational time involved is more. Moreover it is very difficult to visualize the evolved circuits. To overcome the above constraints a new 2D chromosomal representation instead of the usual linear representation has been proposed. 2D crossover and mutation operators have been designed to operate over this 2D chromosome. A technique to obtain the pictorial representation of the evolved circuits has been included.

This paper is organized as follows: section II describes the problem definition, 2D chromosomal representation, 2D crossover and mutation operations. Section III deals with the pictorial representation of the evolved circuits and section IV gives the results and comparison between the two modes of representation.

IV. PROBLEM DEFINITION

The problem is to evolve a digital circuit that satisfies a specified truth table with minimum number of gates and minimum computational time i.e., to design a black box with known inputs and outputs. The black box is encoded suitably into chromosomes and optimized using evolutionary algorithms [6]. In [4] even though the individual circuits were represented as a two dimensional array, all the elements were then arranged in a linear form to form the chromosome and then crossover and mutation operations were carried out (refer Fig.1) which consumes more computational time. To avoid this, the circuit can be represented as a 2D chromosome itself. It also helps in better visualization. The 2D genetic operators i.e., crossover and mutation to operate over these 2D chromosomes also has to be developed.

A. 2D representation (Encoding of the circuit)

Encoding of a sample circuit (individual) A will be discussed in this section. A is created randomly which contains the information about the type of gates and the corresponding inputs to the gate. Here the gates which are used are XOR, AND, OR and wire. Matrix A for a 2 level circuit is shown in Fig 2. First column represents the inputs for the first level gates. Second column gives details about the gates used in the first level. Third and fourth columns represent the inputs and the corresponding gates for the next level. This type of representation gives a better visualization. All 'A's are binary numbers. For any n level circuit the matrix size will be n x n.

A_{11}	A_{12}	A_{13}	A_{14}
A_{21}	A_{22}	A_{23}	A_{24}
A_{31}	A_{32}	A_{33}	A_{34}
A_{41}	A_{42}	A_{43}	A_{44}

A

Figure 2. Sample chromosome for a 2 level circuit

Proposed 2D crossover technique

Assuming that the possible solution circuits are encoded in to n x n (e.g. 4 x 4) matrices and taking the case of two parent individuals A and B, the 2D crossover technique has been illustrated in this section. Parents A and B are shown in Fig.3. The next step is to prepare the mask matrices to find the region of crossover. A set of 4 random numbers [R1 , R2, C1, C2] are generated where R1 and R2, C1 and C2 are numbers between '1' and 'n' and they indicate the start and end row and the columns of a sub matrix in the parents respectively. The genes in this sub matrix will be swapped between the parents. The sub matrices where the genes are to be swapped are shown in Figure 4.

The corresponding offsprings are shown in Fig.5.

A_{11}	A_{12}	A_{13}	A_{14}
A_{21}	A_{22}	A_{23}	A_{24}
A_{31}	A_{32}	A_{33}	A_{34}
A_{41}	A_{42}	A_{43}	A_{44}

A

B_{11}	B_{12}	B_{13}	B_{14}
B_{21}	B_{22}	B_{23}	B_{24}
B_{31}	B_{32}	B_{33}	B_{34}
B_{41}	B_{42}	B_{43}	B_{44}

B

Figure 3. Parent A and B selected for crossover

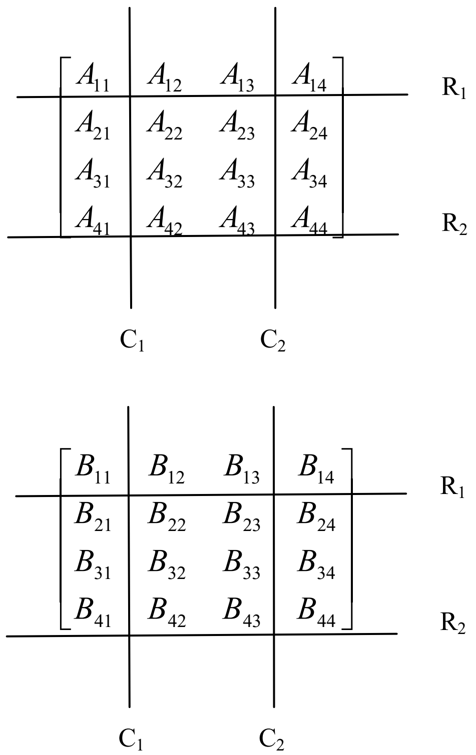


Figure 4. Randomly chosen Sub matrix

Two complementary mask matrices M_1 and M_2 are generated. The elements of matrix M_1 are 1's for rows and columns outside the subset matrix. The rows and columns inside the subset matrix are filled with 0's and 1's randomly and is shown in Fig. 5. The corresponding offsprings obtained from the parents are

$$Offspring_1 = parent_1.M_1 + parent_2.M_2 \quad (1)$$

$$Offspring_2 = parent_1.M_2 + parent_2.M_1 \quad (2)$$

The corresponding offsprings are shown in Fig. 6.

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \quad M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 5. Mask matrices M_1 and M_2

A_1	A_1	A_1	A_1
A_2	B_{22}	A_2	A_2
A_3	B_{32}	B_{33}	A_3
A_4	A_4	B_{43}	A_4

B_{11}	B_{12}	B_{13}	B_{14}
B_{21}	A_{22}	B_{23}	B_{24}
B_{31}	A_{32}	A_{33}	B_{34}
B_{41}	B_{42}	A_{43}	B_{44}

Figure 6. offspring $_1$ and offspring $_2$

B. 2D Mutation

After crossover only a small percentage of the population usually (0.3%) will undergo mutation. For this only a single offspring which needs to be mutated and a single mask is needed. The mutation mask is prepared in the same way as mentioned in the previous section. This mask operator is superposed over the offspring to be mutated. In this region the presence of a '1' indicates a change in the characteristic of the offspring and a '0' indicates no change. Therefore it modifies the gate type and the two inputs, meaning that a completely new cell can appear in the chromosomes.

III . PICTORIAL REPRESENTATION

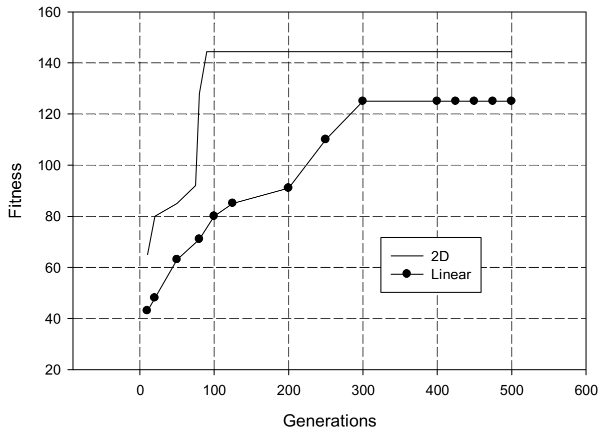
The best chromosome evolved for the circuit generated is a matrix of ones and zeros which can be visualized as a circuit only by the programmer / designer. The matrix obtained is to be decoded to get the evolved circuit. A module has been added to the GA code to represent the circuit pictorially. This module takes in the best chromosome and automatically generates the evolved circuit. The logic symbols and the inputs used are stored as image files and the program retrieves the images whenever needed and the layout is generated as per the evolved circuits.

IV. RESULTS

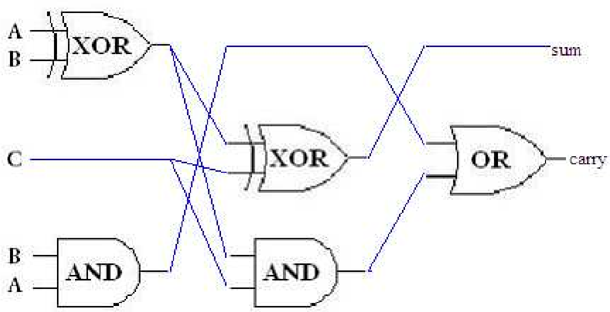
The following parameters were chosen for the GA. Population size=100, number of generations=500, crossover rate=0.7, mutation rate= 0.3%, elitism count=1. Roulette wheel selection technique has been used for selecting the individuals for crossover. Combinational digital circuits with two and three variable inputs and circuits represented by functions in SOP form has been evolved with this proposed 2D representation, crossover and mutation techniques and the results are shown in subsequent sections as examples.

A. Example 1: Full Adder

Fig. 7(a) shows the comparison between the convergence rates of the conventional and the 2D technique for the evolution of a full adder circuit. Fig. 7(b) shows the evolved circuit.



(a)



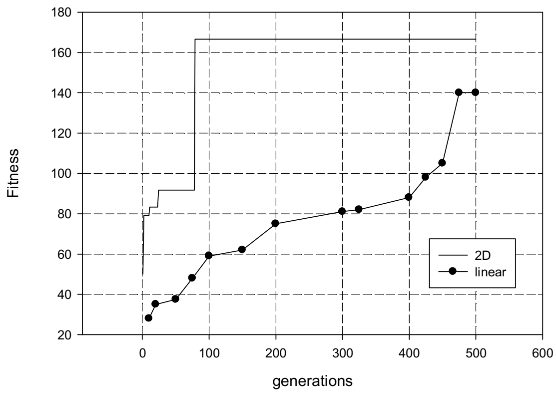
(b)

Figure 7 (a). Comparison of convergence rates between 1D and 2D techniques. (b) Circuit evolved for full adder

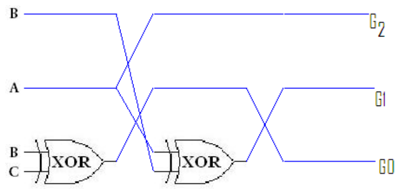
It is evident that only five gates are needed to realize the sum and carry outputs of a full adder. It can be seen from Fig. 7(a) that the convergence rate of the full adder circuit using 2D technique is much faster than the existing techniques. The circuit is evolved within 90 generations using the proposed technique compared to the 300th generation in the other technique.

B. *Example 2: 3 bit binary to gray code converter*

The convergence curves and the corresponding evolved circuit of a 3 bit binary to gray converter is shown in Fig. 8. While it takes around 460 generations to converge using the existing technique it took only 90 generations using the proposed technique.



(a)

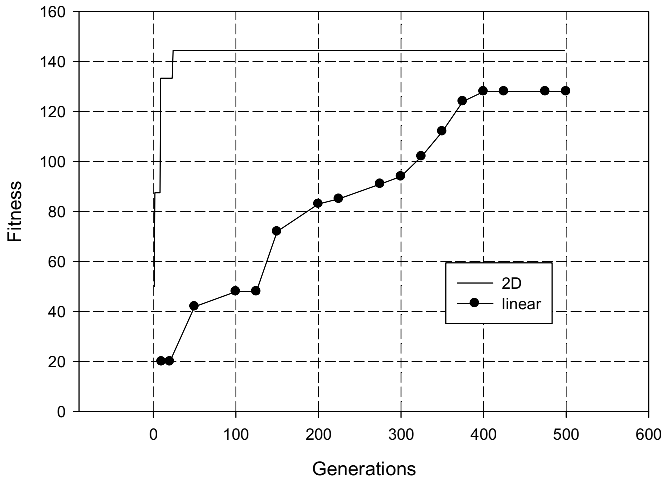


(b)

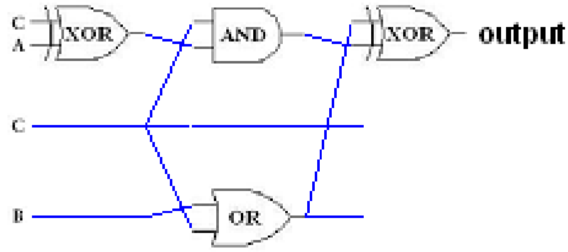
Figure 8 (a). Comparison of convergence rates between 1D and 2D techniques. (b) Circuit evolved for a 3 bit binary to gray code converter.

C. *Example 3: 2 -1 Multiplexer*

The convergence curves and the corresponding evolved circuit of a 2-1 Multiplexer is shown in Fig. 9. It can be observed that the new techniques needs just 40 generations while the existing technique needs 400 generations to evolve the desired circuit.



(a)



(b)

Figure 9 (a). Comparison of convergence rates between 1D and 2D techniques. (b) Circuit evolved for a 2 -1 Multiplexer

D. *Example 4: Function in Sum Of Products Form*

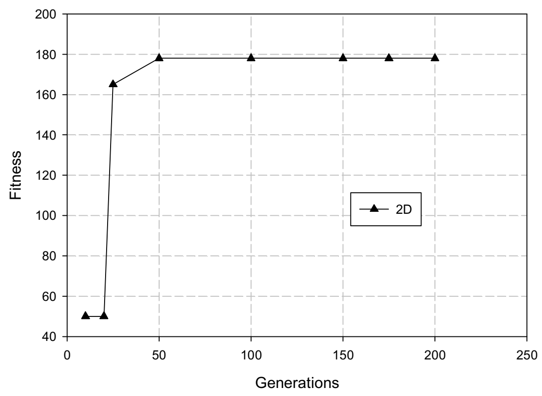
Here an example of a circuit in the SOP form given below is optimized .

$F_1 = \sum (5, 7)$

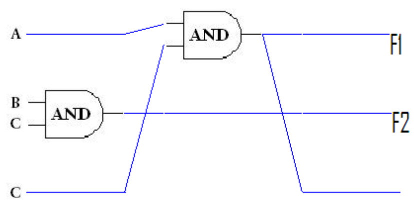
$F_2 = \sum (3, 7)$

The convergence curve using the proposed 2D technique and the corresponding evolved circuit is shown in Fig. 10. From the figure it can be seen that the circuit can be evolved in almost 50 generations.

From the above examples and discussions it can be concluded that the proposed technique is very fast compared to the existing technique and the converged circuit is always obtained in less than 100 generations.



(a)



(b)

Figure 10 (a). Comparison of convergence rates between 1D and 2D techniques. (b) Circuit evolved for a 2 -1 Multiplexer

IV. CONCLUSION

A new 2D representation for the design of combinational digital circuits is proposed. Suitable crossover and mutation techniques are developed for this 2D representation. The fittest circuit evolved is in the form of '1's and '0's. These '1's and '0's are converted to actual circuit diagrams for proper visualization. With the proposed approach, convergence speed of GA has been significantly increased compared to conventional method which in turn reduces the computational time effectively.

REFERENCES

[1] Slowik and Michal Bialko,. "Evolutionary Design of combinational Digital Circuits: State of the Art, Main Problems,and future Trends," Proceedings of the 2008 First International Conference on Information Technology 19-21 May 2008,Gdansk,Poland

[2] J. F. Miller and Peter Thomson, "Discovering Novel Digital Circuits using Evolutionary Techniques", IEE Colloquium on Evolvable Systems, London, 1998.

[3] C. Coello, A. Christianses and A. Aguirre, "Automated Design of combinational Logic circuits using Genetic algorithms," Proc. of the Inf. Con/ on Artificial NeuralNetsand Genetic Algorithms "pp. 3335-3338,1997.

[4] C. Coello and A.Christianses,"Use of Evolutionary techniques to automate the design of Combinational Circuits",International Journal of Smart Engineering System Design 2000.

[5] Ahmed T. Soliman and Hazem M. Abbas," Combinational Circuit Design Using Evolutionary Algorithms", Electrical and Computer Engineering 2003. IEEE CCECE 2003 Canadian Conference vol.1 pp 251-254.

[6] F. Z. Hadjam and M. Benmohammed," Hierarchical Evolution of Digital Combinational Circuits",Information and Communication Technologies, 2006 ictta'06 , pp 2554-2557.

[7] David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley,1989.