Improvement of Optimization in Design of Synchronous Sequential Circuits by Using Evolvable Hardware

P. Soleimani, S. Mirzakuchaki, R. Sabbaghi-Nadooshan, and M. Bagheri, *Member, IACSIT*

***Abstract*—Evolvable hardware is a new method for designing the digital logic circuits. In this paper, a method has been presented for designing the synchronous sequential logic circuit by using the evolvable hardware. In this approach, the sequential logic circuit is divided into two sections; the combinational logic circuit and DFFs. The combinational logic part is designed by using a constant structure and their connections are set with genetic algorithm (GA). The results show that our method can reduce the average number of generations by limitation the search space.**

***Index Terms*—Sequential circuit, combinational circuit, genetic algorithm, evolvable hardware, evolutionary algorithm.**

I. Introduction

The aim of evolvable hardware is the self-sufficient reconfiguration of hardware structure in order to advance performance. In designing and optimizing of the evolutionary circuit, an optimization algorithm searches the space of all possible circuits and determines solution circuits with desired functional response. Simpler structure of combinational circuits in compare with sequential circuits and the lack of feedback in this circuits is caused more researches have been done in this field.

However, relatively few efforts have been done to evolving sequential logic circuits [1]. The works have been done in the field of sequential circuits are presented in Table 1.

In the rest of this paper, Sections 2 consider the main idea of the proposed method. Section 3 describes GA operators. Section 4 describes details of process to define structure of chromosomes. Section 5 explains fitness evaluation process to evaluate the performance of evolved circuits. Simulation environment has been described in Section 6.

Section 7 summarizes the experiment of proposed method on the sequential circuit; Section 8 shows the simulation results for target circuit. Finally, in section 9 the conclusion of this paper is presented.

II. The Proposed Method

As Fig. 1, the structure of sequential logic circuits comprises a set of two sections of combinational logic circuit and D flip-flops [2].

In this approach, for designing combinational parts, we present one type rectangular array of logic gates and use it to build next state of DFFs and primary outputs. Fig. 2 shows block diagram of proposed method.

As Fig. 3, this array has R rows and C columns and their gates are chosen from OR, AND and NOT gates. Except NOT gate, the other gates have two inputs and one output. Each gate input can be taken from primary inputs, DFFs outputs or output of each gate that is neighbor to the left.

In our approach, one multiplexer is added to the inputs of gates in each cell array (see Fig. 4), input of DFFs and before primary output.

TABLE I: Related Works.

Author	year	Target Application
A. Thomson[3]	1995	Robotics
C. Manovit[4]	1998	Frequency detector, odd parity detector,module-5 counter, serial adder
C.Aporntewan[5]	2000	Serial adder, 0101 detector, module-5 counter, Reversible 8-counter
T.Kalganova[2]	2004	Module-4 counter, 1010 detector
A.T.Soliman[6]	2004	3-bit up-counter
A.P.shanthi[7]	2009	Module-6 counter ,'lion' circuit

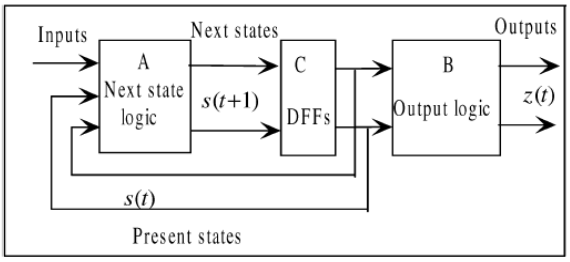


Fig. 1. Discription on the circuit parts [2].

Then, by determining the proposed structure of chromosome encoding (Section 4) and by using genetic algorithm, we have evaluated the different states of logic gates connection for each array separately, to achieve correct functionality and minimum number of logic gates. The structure of multiplexer has been used, is shown in Fig. 5.

Manuscript received June 20, 2011; revised August 30, 2012.

P. Soleimani and R. Sabbaghi-Nadooshan are with the Department of electronic engineering, Islamic Azad University, Central Tehran Branch (e-mail: parisa.soleimani@ gmail.com, r_sabbaghi@iauctbt.ac.ir).

S. Mirzakuchaki and M. Bagheri are with the Department of electronic engineering, Iran University, of Science and Technology (e-mail: M_kuchaki@iust.ac.ir, m-bagheri@elec.iust.ac.ir).

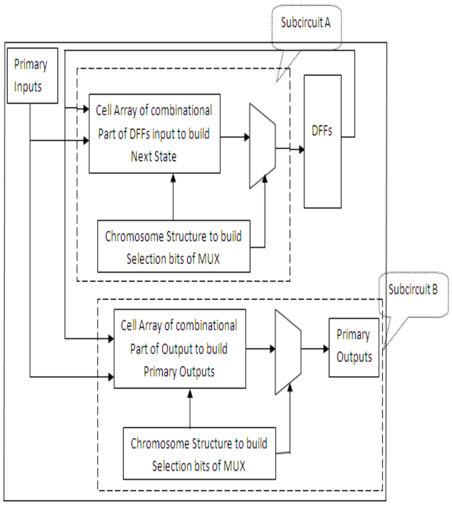

Subcircuit A
Primary Inputs
Cell Array of combinational Part of DFFs input to build Next State
DFFs
Chromosome Structure to build Selection bits of MUX
Subcircuit B
Cell Array of combinational Part of Output to build Primary Outputs
Primary Outputs
Chromosome Structure to build Selection bits of MUX

Fig. 2. Block diagram of proposed method.

Evolving of each array separately, has been lead to increase the speed of evolution the circuit and decrease the run time.

III. Genetic Algorithm Operators

In this paper, genetic algorithm has been used to evolve the particular circuit. Individuals have been defined in type of bit string. We described genetic algorithm operators as follows:

1) Selection: we chose the roulette wheel as a method for parent selection.
2) Crossover: a pair of parents by using scatter crossover produce child.
3) Mutation: mutation is described as a random change of genes in the chromosome. The mutation method that has been used in this study is the uniform mutation.

Population size has been defined as 10 and maximum number of generation is set to 30000 .The algorithm is stopped if there is no improvement in the fitness function for 10,000 consecutive generations. This is for overcome the stalling effect.

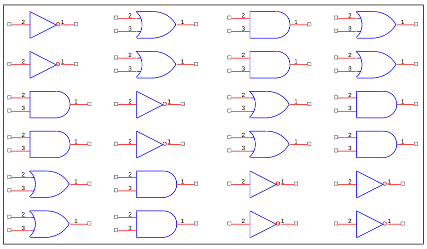


Fig. 3. Schematic of the rectangular array structure has been used in the combinational logic parts.

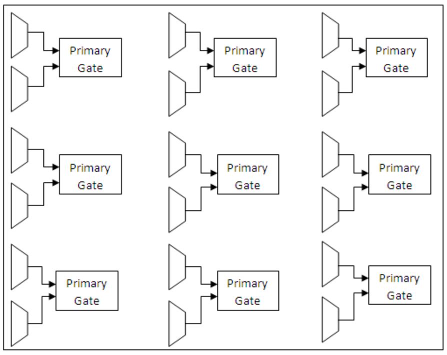


Fig. 4. Block diagram of cell array after adding the multiplexer to array.

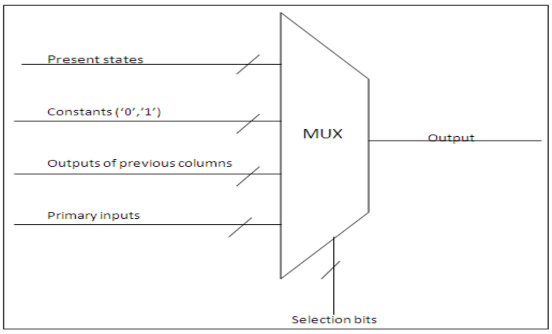


Fig. 5. Structure of multiplexer

IV. Chromosome Encoding

The chromosome defines the construction of the logic circuit and the connectivity between logic gates.

In this approach, we put a multiplexer to input of each gate, DFFs and before primary outputs. We change connection between gates and DFFs by changing the selection bits of multiplexers. Inputs of multiplexers of logic gates are taken from primary inputs, present states of DFFs, outputs of all gates that is the neighbor left column and constant values that set equal '0' and '1'. Also inputs of multiplexers of DFFs and primary outputs are taken from primary inputs and outputs of all logic gates that are on the all left columns.

Changing selection bits of used multiplexers leads to different connectivity between logic gates of circuit. We have used the selection bits of multiplexers as chromosome genes (Fig. 6).

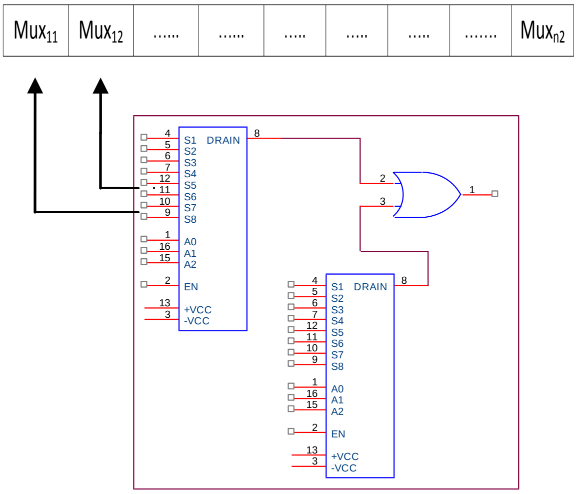


Fig. 6. Structure of chromosome encoding.

V. Fitness Evaluation Process

In this section, the fitness evaluation process has been explained. We used finite state machine (FSM) for evaluation of sequential circuits. In this method, first the desired state is set in the circuit flip flops and then we change the value in primary inputs and compare the output of circuit with the desired ones. At any state, desired output for input combinational logic circuit is the next state of the DFFs but desired output for output combinational logic circuit is primary output (output columns of STT). If in any state, the corresponding output and desired output are equal, then the fitness value is increased.

In proposed method, we measured fitness function by two main criteria: design and optimization. In the first criteria, functionality of the circuit is evaluated. The goal of first criteria is to evolve a circuit that has 100% functionality. Then in the second criteria, optimization has been performed by reducing the numbers of logic gates that is used in the target circuit. Fitness optimization is activated once design fitness value reaches 100% functionality.

The design criterion of any individual is evaluated as these steps:

1) The initial value for design fitness has been considered to zero.
2) The primary inputs and present state of DFFs have been set externally. Then the value of output of the circuit is measured after sending a clock signal to DFF.
3) The corresponding output with desired output has been compared. We can use this equation to measure fitness:

$$F_{Design}=F_{Design}+\text{ number of equal output bits.}$$

4) The steps 2-3 have been repeated for the remaining states of FSM and functionality of circuit has been evaluated.

The optimization criterion has been calculated as follow steps:

1) The initial value for optimization criterion has been considered as:

$$F_{Optimization} = R\times C$$

2) For each individual, total number of logic gates have been calculated. So, we can use this equation to find optimization fitness:

$F_{Optimization} = (R\times C)$ - number of logic gates that is used in new circuit.

Now, the final fitness of individual could be calculated by using this equation:

$$F_{Final} = F_{Design}+ F_{Optimization}$$

Both of the procedures described above are applied for evaluation of both combinational parts of sequential logic circuit.

VI. Simulation Test Bed

In this paper, we used Modelsim as VHDL hardware programming language simulator and MATLAB software for implement GA. Also we used GA toolbox in MATLAB Revision 2010 software to run the evolutionary algorithm. In addition we used simulator link TM MQ toolbox in this software. It can access to Modelsim, open HDL code, run it for different inputs that are determined in MATLAB code and save outputs in the variables of MATLAB codes. Fig.7 shows block diagram of this process. Hence this toolbox is as a link between Modelsim and MATLAB.

VII. Experiment on the Sequential Circuit

In this section, the proposed method is experimented on the sequential circuit. State transition graph of this circuit has been shown in Fig. 8. Also, in Fig. 9 step1 shows the symbolic state table of FSM and state assignment to each state. In step2, STT of the target circuit is shown. In step3, STT of the circuit is divided into input combinational logic subcircuit A and output combinational logic subcircuit B. ([2]).

This circuit has six states that has been lead to use three

DFFs. As we explained in previous sections, we evaluated each subcircuit A and B, separately. Finally, the sequential circuit is assembled.

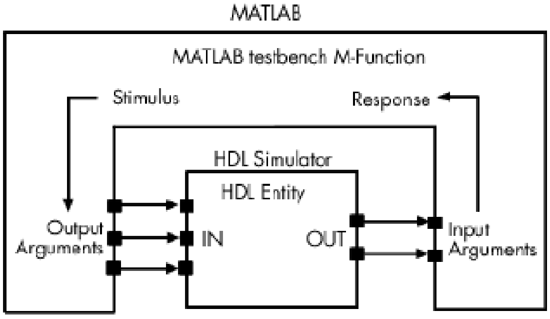
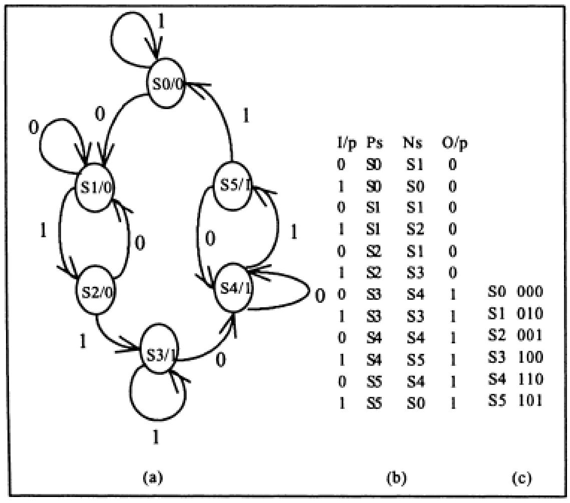


Fig. 7. Connection of MATLAB and modelsim[8].



I/p	Ps	Ns	O/p
0	S0	S1	0
1	S0	S0	0
0	S1	S1	0
1	S1	S2	0
0	S2	S1	0
1	S2	S3	0
0	S3	S4	1
1	S3	S3	1
0	S4	S4	1
1	S4	S5	1
0	S5	S4	1
1	S5	S0	1

S0	000
S1	010
S2	001
S3	100
S4	110
S5	101

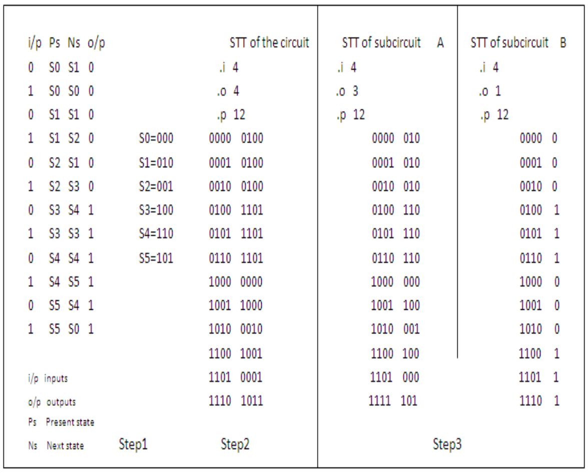
Fig. 8. Sequential detector (a) state transition graph, (b) state transition table, (c) state assignment [2].

VIII. Results

In this section, structure of the evolved circuit has been shown in Fig. 10. The results that have been achieved by proposed method in compare with [2] have been shown in Table II. The solution obtained by manual method, uses almost 2 times more gates than the circuit created by the proposed method, and the solution reported in [2] uses one gate more than our method and with more generations in compare with our method.

IX. Conclusion

This paper, presented a method to design and optimize the synchronous sequential circuits. In this method, we have used GA as the evolutionary algorithm and describe digital circuit based on gate level. Combinational blocks have been produced from AND, OR and NOT gates and each of them evaluated separately. The achieved results by proposed method have been compared with other method [2]. This comparison shows our method can design sequential logic circuits better than [2] and need to less time for evaluating.



i/p	Ps	Ns	o/p		STT of the circuit	STT of subcircuit A	STT of subcircuit B
0	S0	S1	0		.i 4	.i 4	.i 4
1	S0	S0	0		.o 4	.o 3	.o 1
0	S1	S1	0		.p 12	.p 12	.p 12
1	S1	S2	0	S0=000	0000 0100	0000 010	0000 0
0	S2	S1	0	S1=010	0001 0100	0001 010	0001 0
1	S2	S3	0	S2=001	0010 0100	0010 010	0010 0
0	S3	S4	1	S3=100	0100 1101	0100 110	0100 1
1	S3	S3	1	S4=110	0101 1101	0101 110	0101 1
0	S4	S4	1	S5=101	0110 1101	0110 110	0110 1
1	S4	S5	1		1000 0000	1000 000	1000 0
0	S5	S4	1		1001 1000	1001 100	1001 0
1	S5	S0	1		1010 0010	1010 001	1010 0
					1100 1001	1100 100	1100 1
i/p inputs					1101 0001	1101 000	1101 1
o/p outputs					1110 1011	1111 101	1110 1
Ps Present state							
Ns Next state				Step1	Step2	Step3	

Fig. 9. Process of STT of sequential circuit where .i input=input+present state bits, .o defined the number of outputs calculated, outputs=next state+output bits, .p is the number of product terms

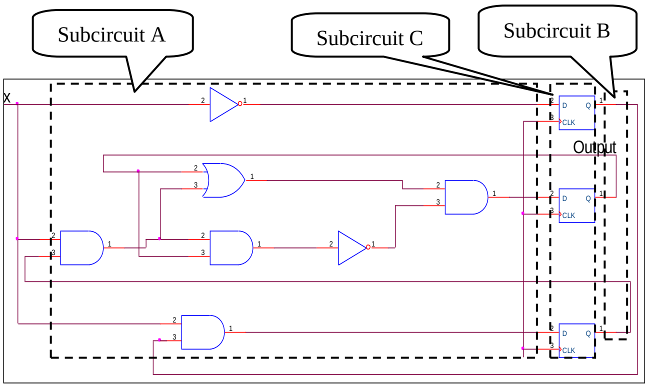


Fig. 10. Evolved optimal circuit solution for sequential detector.

TABLE II: SOLUTION OBTAINED FOR SEQUENTIAL DETECTOR.

Proposed method	**T.kalganova[2]**	**Manual method**
DA=XB	*DA =XB*	*DA=AC'+AX'+B CX'*
DB=X'	*DB=X'*	*DB=BX+A'CX*
DC=(*XAC*)'(*C+XA*)	*Dc=XAC'+X'C+A'C*	*DC=BX+A'C'X'+A'B'X'+AC'X*
Z=C	*Z=C*	*Z=A+BC*
Subcircuit *A*=7	Subcircuit *A*=8	Subcircuit *A*=17
Subcircuit *B*=1	Subcircuit *B*=1	Subcircuit *B*=2
Subcircuit *C*=3 D flip-flops	Subcircuit *C*=3 D flip-flops	Subcircuit *C*=3D flip-flops

For future works it can be considered the evolution of the large scale sequential circuits by using proposed method that is used more in industry.

REFERENCES

[1] H. Liang, W. Luo, and X. Wang, "A tree-step decomposition method for the evolutionary design of sequential logic circuits," in *Genet Program Evolvable Mach*, vol. 10,pp. 231-262, 2009.

[2] B. Ali, A. Almaini, and T. Kalganova, "Evolutionary algorithms and their use in the design of sequential logic circuits," in *Genetic Program, Evolvable Machine*, vol. 5, pp.11-29, 2004.

[3] A. Thompson, "Evolving electronic robot controllers that exploit hardware resources," vol. 929, pp. 640-656, 1995.

[4] C. Manovit, C. Aporntewan, and P. Chongstitvatana, "Synthesis of synchronous sequential logic circuits from partial input/output sequence," vol. 1478, pp. 98-105, 1998.

[5] C. Aporntewan and P. Chongstitvatana, "An on-line evolvable hardware for learning finite-state machine," in *Proceeding of int*, pp. 13-15, 2000.

[6] A. T. Soliman and H. M. Abbas, "Synchronous sequential circuits design using evolutionary algorithm," vol. 4, pp. 2013-2016, 2004.

[7] A. P. Shanthi and L. K. Singaram, "Evolution of asynchronous sequential circuits," in *Proceeding of the 2005 NASA/DoD Conference on Evolvable Hardwarex*, pp. 93-96, 2005.

[8] MathWorks - MATLAB and Simulink for Technical Computing. [Online]. Available: http://www.Mathworks.com.

P. Soleimani received the B.S. degree in bio electronic engineering from the Science and Research Branch, Islamic Azad University, Tehran, Iran in 2007 and received the M.S degree in electronic engineering from Central Tehran Branch of Islamic Azad University, Tehran, Iran in 2011. Her research interest includes Evolvable Hardware, digital signal processing.

S. Mirzakuchki received the BS in Electrical Engineering from the University of Mississippi in 1989, and the MS and PhD in Electrical Engineering from the University of Missouri-Columbia, in 1991 and 1996, respectively. He has been a faculty member of the College of Electrical Engineering at the Iran University of Science and Technology, Tehran, since 1996. His current research interests include characterization of semiconductor devices and design of VLSI circuits. Dr. Mirzakuchaki is a member of IEEE and IET (formerly IEE) and a Chartered Engineer.

R. Sabbaghi-Nadooshan received the B.S. and M.S. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 1991 and 1994 and the Ph.D. degree in electrical engineering from the Science and Research Branch, Islamic Azad University, Tehran, Iran in 2010. From 1998 he became faculty member of Department of Electronics in Central Tehran branch, Islamic Azad University, Tehran, Iran. His current research interests include design of digital circuits, Nanocomputing, and Networks-on-Chips. He is a member of IEEE.

M. Bagheri received the M.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran in 2011. His research interests are in evolvable hardware evolutionary computation, design of digital circuits, and bio engineering applications. He is currently a member of IEEE since 2008.