

On Design of Multiple-Valued Sequential Reversible Circuits for Nanotechnology Based Systems

Majid Mohammadi
Shahid Beheshti University
Tehran, Iran
mohammadi@mail.uk.ac.ir

Mohammad Eshghi
Shahid Beheshti University
Tehran, Iran
m-eshghi@sbu.ac.ir

Majid Haghparast
Science & Research Branch
Islamic Azad University
haghparast@sr.iau.ac.ir

Abstract

Multiple-valued reversible logic is an emerging area in reversible and quantum logic circuit synthesis. Multiple-valued reversible logic circuits can potentially reduce the width of the reversible or quantum circuit which is a limitation in current quantum technology. In this paper we propose a method to synthesize the multiple-valued reversible sequential circuits. Implementations for ternary D and T flip flops and edge triggered D flip flop are proposed. Synthesis of generalized fanout circuit and generalized r-valued T flip flop are also presented.

Keywords: Multiple-valued logic, reversible logic, sequential circuit, genetic algorithm.

1. Introduction

It is proved that irreversible circuits have power dissipations regardless of their underlying technology [1]. Binary reversible circuits have been studied in low-power CMOS design, quantum computation, nanotechnology-based systems, and optical computing. On the other hand a multiple-valued logic (MVL) reduces the width of the reversible or quantum circuit. At present, quantum implementations are limited to a small number of quantum bits (qubits). MVL coding reduces the number of qubits in the circuit, allowing for larger specifications in current technology [2].

An r -valued $m \times m$ function is reversible if and only if it maps each of input patterns to a unique output pattern [3]. Synthesis of a combinational MVL circuit implements a reversible MVL function by a cascade of reversible gates. In multiple-valued quantum circuits, the quantum digits (qudits) are the signals and quantum gates are operators, represented by unitary matrices [4,5].

Many years, the researches in reversible logic design were limited to the combinational logic circuits. To design a sequential logic circuit, it is necessary to use the feedback, which seems to violate the basic laws of reversible logic circuits. In reversible logic circuits loops was not allowed. However, Toffoli and Fredkin in 1980 in their papers, [6] and [7], presented the sequential reversible circuits. Recently, Banerjee and

Pathak in [8] have shown that the feedback is not a violating condition for reversible logic circuits. A circuit is feasible by feedback if its transition function (or state transition table) remains reversible, or its transfer matrix is unitary.

In this paper, a method to design the sequential MVL reversible circuits is presented. We use the genetic algorithm (GA) [9] as a synthesis tool to design and optimize these circuits. We use the generalized CNOT, Cycle, and Self shift MVL gates which Miller et al [10] have used in their combinational synthesis algorithm. The GA-based synthesis algorithm, used in this research is a simple extension of the work which is done in [11] for binary circuits. Since the MVL reversible gates are realized in quantum technology [2], the proposed method is also applicable in synthesis of quantum sequential MVL circuits.

The background on MVL reversible gates is presented in section 2. GA-based synthesis method for combinational circuits is proposed in section 3. In section 4 we propose an algorithm to synthesize some basic MVL sequential circuits.

2. Multiple-valued reversible gates

An r -valued gate is reversible if and only if it maps each of input patterns to a unique output pattern. As a result, a reversible gate has equal number of inputs and outputs. Two MVL reversible and quantum set of gates are used in the synthesis of MVL circuits which are realizable in existing MVL quantum technology. Generalized ternary gates (GTG) are ternary controlled 2×2 gates which are used in [12] to synthesize the ternary reversible circuits. The generalized CNOT, Cycle, and Self shift gates which are used in [10] have a simpler structure than GTG gates, and can be extended to higher-radix logic with more control inputs. In this paper we use the generalized CNOT, Cycle, and Self shift gates in our synthesis method. Fig.1 shows the symbols and operations of these gates. In this figure r is radix and n ($0 \leq n < r$) is the type of gate (E.g., C_1 , C_2 , ...). Table 1 shows the truth table of these gates in ternary case. Note that the ternary S_2 gate is equivalent to the NOT gate. The C_0 gate is as a buffer.

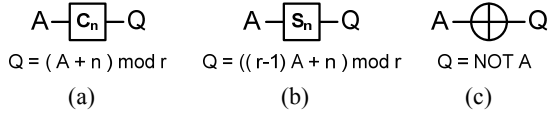


Fig.1. Symbol and functionality of Some MVL gates: (a) Cycle gate, (b) Self shift gate, (c) NOT gate

Table 1: truth table of some ternary gates

input A	Target output (Q)					
	C ₁	C ₂	S ₀	S ₁	S ₂	NOT
0	1	2	0	1	2	2
1	2	0	2	0	1	1
2	0	1	1	2	0	0

Fig.2 shows the symbol of the 2×2 MVL controlled gates. The A_0 input is control input and A_1 is target input. The P value in circle of control input (black circle) represents the threshold value of input. If the control input (A_0) is less than the threshold (P), then the target output (Q_1) is as the input target (A_1); else, the gate's function is applied to the target input (A_1) to generate a new value for the target output (Q_1). Q_0 is always equal to A_0 . Table 2 shows the truth table of controlled gates (only target output) in the ternary case with threshold value of $P = 2$.

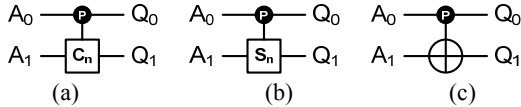


Fig.2. Controlled MVL gates: (1) Controlled Cycle gate, (b) Controlled Self shift gate, (c) CNOT gate. P is threshold of input.

Table 2: truth table of controlled ternary gates

Inputs		Target output (Q_i)				
A_0	A_1	CC1	CC2	CS0	CS1	CNOT
0	0	0	0	0	0	0
0	1	1	1	1	1	2
0	2	2	2	2	2	1
1	0	0	0	0	0	0
1	1	1	1	1	1	1
1	2	2	2	2	2	2
2	0	1	2	0	1	2
2	1	2	0	2	0	1
2	2	0	1	1	2	0

These gates are generalized to the controlled $n \times n$ gates [10] with $n-1$ control inputs and one target input. Some threshold values are assigned to the control inputs. The gate applies its function to the target input if all controls are equal to or greater than their corresponding threshold values.

3. Using genetic algorithms in synthesis of combinational MVL reversible circuits

In this section, application of genetic algorithm (GA)

in synthesis of the *combinational* MVL reversible circuits is described. In the next section the method is extended to *sequential* MVL reversible circuits using state transition table.

We use a simple extension of GA-based synthesis algorithm proposed in [11] to synthesize the MVL reversible circuits. In GA-based synthesis algorithms the circuits are shown in the music line style. In this style an n -input/output circuit is shown as n parallel lines that some reversible gates are placed on them (Fig.3). Each gate is coded to a string as is shown in Fig.3.

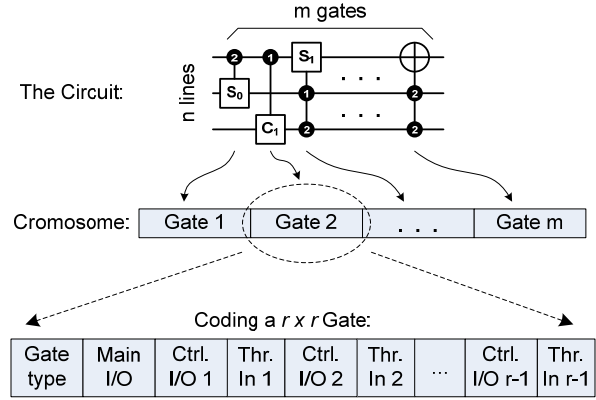


Fig.3. A chromosome and coding a reversible $r \times r$ gate as a string

The first field indicates the type of gate. The second field shows the binary number of the location of the main input/output. The next $2(r-1)$ fields show the number of the location of gate's control inputs and their corresponding threshold values. One chromosome which contains m gates, codes a circuit.

Firstly, we construct a population of chromosomes which represents the MVL reversible circuits. This population is initialized randomly. Then three operators of GA, namely mutation, crossover, and selection are applied to the population of chromosomes to generate a better population. The crossover operator selects two chromosomes randomly and exchanges some of their corresponding segments. The mutation operator applies random changes to the selected chromosome, with a specific probability. The selection operator selects some of the good chromosomes for reproduction of the next population. Selection directs the algorithm toward the optimum. Fig.4 shows the GA synthesis algorithm. In selection operator we calculate a fitness function to evaluate the chromosomes. In synthesis problem, the fitness function is the Hamming distance of the output of desired truth table and output of each circuit as shown in equation 1.

$$HD = \sum_i \sum_j |O_{ij} - S_{ij}| \quad (1)$$

Where O_i is a row vector of the desired truth table and S_i is a row vector of the truth table of the synthesized circuit. O_{ij} is the j^{th} element of the vector O_i and S_{ij} is the j^{th} element of the vector S_i .

We have developed a GA-based synthesis software for synthesis of reversible MVL circuits using the CNOT, Cycle, and Self shift gates.

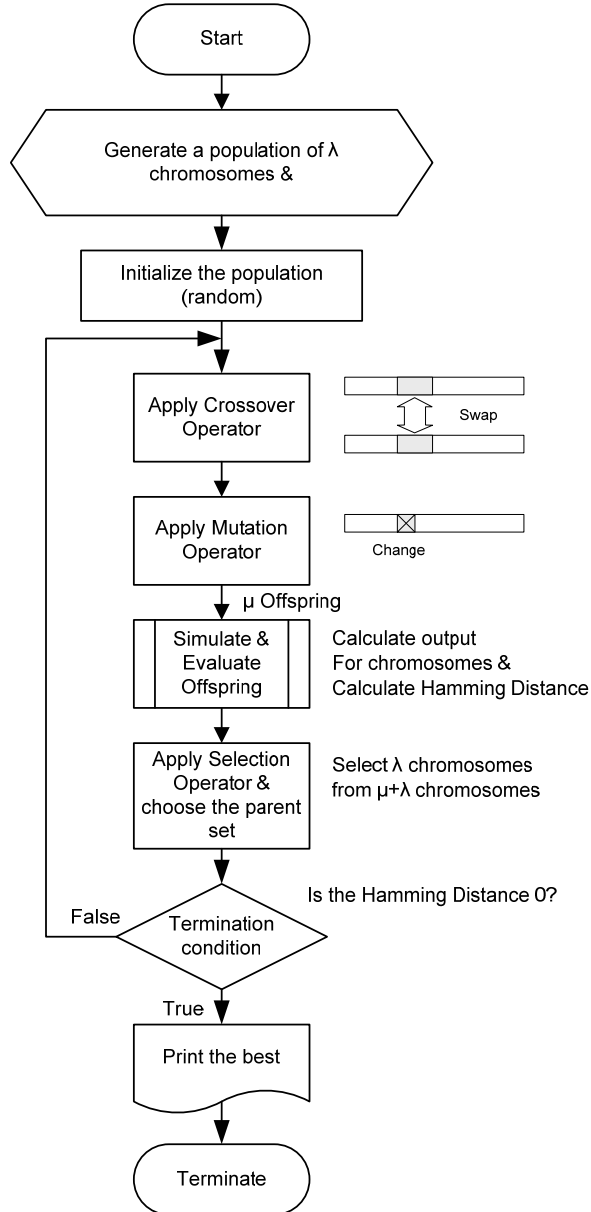


Fig.4. Genetic algorithm used for synthesis of reversible logic circuits

4. Design of feedback circuits using the state transition table

In this section we propose a method to design the flip flops and other sequential circuits, using the state transition table. In this method we use reversible *combinational* MVL synthesis algorithms to design the *sequential* MVL circuits. This method has three major steps as follows:

1. Construct the state transition table of desired sequential circuit. The input section of this table includes all external data inputs, constant inputs (if needed), and the feedback signals of the circuit in time t . Output section includes the main output signals of the circuit at time $t+1$ and garbage outputs (if needed) to maintain the reversibility of the state transition table.
2. Use a proper MVL reversible circuit synthesis tool to implement the transition table.
3. Use the unit wires (as mentioned by Fredkin [7]) to apply proper feedbacks to the circuit according to the input and output variables.

The synthesis algorithm may be the Miller's bidirectional synthesis method [10] or the GA-Based synthesis method. The important advantage of our GA-based synthesis methods to other methods is that various universal and non-universal gates can be used in the synthesis.

Example 1. Clocked ternary D-FF:

The transition table of this circuit is shown in Table 1. As is shown, the Clk input can only accept the 0 and 2 values. Thus, the value of signals for Clk=1 is assumed as don't cares. These values are not shown in the truth table; however, they have to be assigned such a way that the overall table be reversible.

Table 3: State transition Table for a ternary D-FF

Clk	D	Q^t	G_2	G_1	Q^{t+1}
0	0	0	X	X	0
0	0	1	X	X	1
0	0	2	X	X	2
0	1	0	X	X	0
0	1	1	X	X	1
0	1	2	X	X	2
0	2	0	X	X	0
0	2	1	X	X	1
0	2	2	X	X	2
2	0	0	X	X	0
2	0	1	X	X	0
2	0	2	X	X	0
2	1	0	X	X	1
2	1	1	X	X	1
2	1	2	X	X	1
2	2	0	X	X	2
2	2	1	X	X	2
2	2	2	X	X	2

In this table Clk and D are the inputs of flip flop and

Q^t is the past value of the feedback output at time t . Q^{t+1} is the main output and G_1 and G_2 are the *garbage* outputs. Garbage outputs are some outputs which are added to the truth table to maintain the reversibility of the function. Since the number of inputs of Table 3 is 3, at least two garbage outputs have to be added to the table. Generally, if maximum repetition of patterns in the output is λ , at least η garbage outputs are needed to make the truth table reversible; as is shown in equation 2 (r is radix).

$$\eta = \lceil \log_r \lambda \rceil \quad (2)$$

Although, the values of the garbage outputs are not important, they have to be assigned such a way that the reversibility condition is maintained. The values of these garbage outputs affect on the synthesized circuit. Determining the optimal value for these outputs is an open problem in MVL reversible logic synthesis. In this example, we assumed $G_1=D$ and $G_2=CLK$.

The GA synthesis algorithm has generated a circuit for D-FF shown in Fig.5.a. A smaller circuit is obtained using the Cycle gates which is shown in Fig.5.b. In these figures we have also applied the feedback signals, shown as dotted lines with the bold arrows.

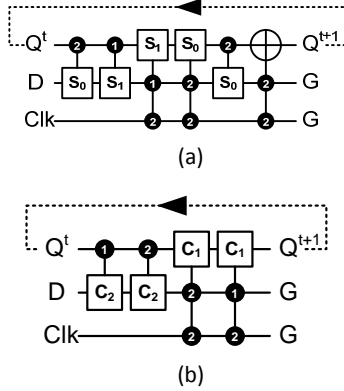


Fig.5. Synthesis of ternary clocked D-FF (a) using Self shift and CCNOT gates (b) using Cycle gates.

Any output of a reversible or quantum gate can drive only one input of another gate. That is, the fanout for a reversible gate is 1. Consequently, to copy the Q^{t+1} signal as the output of the flip flop, we have to use a fanout circuit. We use the GA synthesis algorithm to design this circuit. Table 4 shows the fanout truth table. In this table when A_0 is '0', the Q_0 and Q_1 are equal to A_0 . Other states are *don't care conditions* which we assign some values to keep the reversibility of the function. The A_0 input is a constant input. In the synthesis process the values of constant inputs is also important. In fanout circuit the '0' value results a simpler circuit. The synthesized circuit is shown in Fig.6.a. Fig.6.b shows the synthesized 4-valued (quaternary) fanout circuit. The regularity of these two circuits leads us to a generalized r-valued fanout circuit

which is shown in Fig.6.c.

Table 4: Truth table of fanout circuit.

A1	A0	Q1	Q0
0	0	0	0
0	1	1	1
0	2	2	2
1	0	X	X
1	1	X	X
1	2	X	X
2	0	X	X
2	1	X	X
2	2	X	X

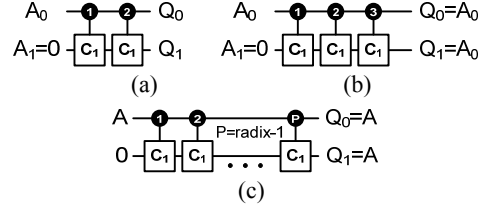


Fig.6. Fanout circuit copies A_0 to outputs Q_0 and Q_1 when $A_1=0$: (a) Ternary circuit, (b) Quaternary circuit (c) Generalized r-valued circuit

Fig.7 shows the complete ternary clocked D-FF in which the fanout circuit is also embedded. As Fredkin in [7] mentioned, the wire which connects Q^{t+1} to Q^t is a unit wire with a specific delay. This delay is the time difference of the Q^t and Q^{t+1} signals. It is necessary in some sequential circuits to let the clock signal return to '0' before a change in the state of the Q^t . For example we can use a 1×1 gate to generate a delay. The fanout circuit has also a delay which is considered as the unit delay in the feedback path.

Example 2. T-FF:

The summary of transition table of T-FF is shown in Table 2. If we set the Clk and T signals to 2, the output of T-FF (Q^{t+1}) is the unit Cycle of its previous value (Q^t). Since the value of '1' for Clk input is not allowed, the output vales are considered as don't care conditions. As the D-FF, the truth table of the T-FF has also two garbage outputs.

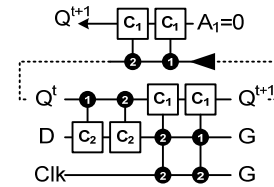


Fig.7: Clocked D-FF with fanout circuit to copy the Q^{t+1} output.

Table 5: Summary of state transition table of the ternary T flip flop

Clk	T	Q^t	G	G	Q^{t+1}
0	X	Q^t	X	X	Q^{t+1}
1	X	X	X	X	X
2	0	Q^t	X	X	Q^{t+1}
2	1	Q^t	X	X	Q^{t+1}
2	2	0	X	X	1
2	2	1	X	X	2
2	2	2	X	X	0

Fig.8.a shows the GA synthesized circuit for this truth table. It has only one unit Cycle gate. The fanout circuit is also included. We have extended this design to the r -valued T-FF as shown in Fig.8.b. Since in the generalized circuit, the T and Clk signals accept just two 0 and $r-1$ values, the inputs thresholds of 3×3 C1 gate are $P = r-1$.

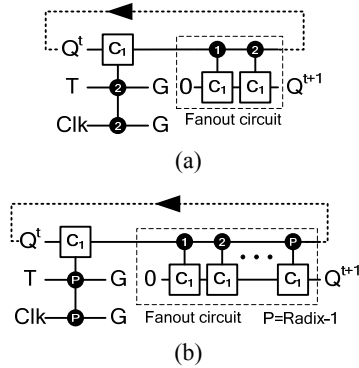


Fig.8. The synthesized T-FF combined with fanout circuit (a) ternary T-FF (b) generalized r -valued T-FF with generalized fanout circuit

A T-FF with an arbitrary operation in each of clock pulses can be designed in the same way. Fig.9 shows this circuit. In this circuit the U gate has a defined function such as Cycle, Shift, NOT, or any combination of such operations.

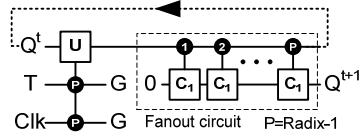


Fig.9. The T-FF circuit with an arbitrary U function.

Example 4. Edge triggered D flip flop:

To avoid some problems, which arise in clock pulse level, we have to design the edge triggered flip flops. As the traditional binary approach we can use two D-FF in series, to design an edge triggered D-FF (Fig.10.a). To implement this design we use the ternary D-FF which is designed in example 1 and shown in Fig.7. The implementation of ternary edge triggered D-FF

using two ternary D-FF's, is shown in Fig.10.b. The Q^{t+1} is output of the D-FF circuit.

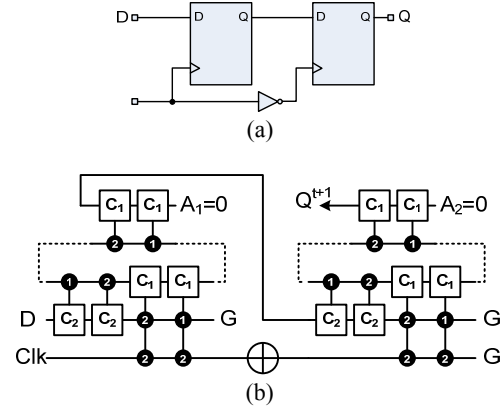


Fig.10. (a) Binary edge triggered D flip flop (b) design of ternary edge triggered D-FF

5. Conclusions

In this paper, we proposed a method to design multiple-valued sequential reversible logic circuits. Many combinational reversible synthesis algorithms can be used. The genetic algorithm is a good candidate because various MVL gates can be used in the synthesis process. To show the efficiency of proposed method, we designed some basic sequential circuits such as clocked D and T flip flops and edge triggered D flip flop. The results of this paper are not presented in the other papers for comparison.

The parameters of the circuits designed in this paper, are presented in table 7. In this Table NOG is number of gates, G_{in} is the number of garbage inputs (or constant inputs), and G_{out} is the number of garbage outputs. For a reversible circuit the number of garbage outputs shows the amount of information or energy losses of the circuit. Therefore, we prefer to design the circuit with the minimum possible number of garbage outputs. In the last column we present the *quantum cost* (QC) of designs. The quantum cost of a circuit in binary case is defined as the number of quantum 1×1 or 2×2 gates for implementation of the circuit. In MVL case we use the same idea. Therefore, the reported quantum cost in the last column of table is the number of 2×2 or 1×1 MVL gates for implementation of the circuits.

The QC of 1×1 and 2×2 gates is 1. For bigger gates we have to implement the gate using 2×2 ones. Our GA-based MVL synthesis tool designed a 3×3 Cycle gate using four 2×2 gates. Thus the QC of a 3×3 Cycle gate is 4. In the same way the QC of the 3×3 controlled-NOT (with two control inputs) is 5 and QC of the 3×3 controlled Self shift gate is 7. A D-FF includes two 3×3 Cycle gates and four 2×2 Cycle gates, resulting to a QC of 12. For other designs QC is calculated in the same

manner.

Table 7: parameters of circuits are designed in this paper

Name	NOG	Gin	Gout	QC
Clocked D-FF	6	1	2	12
Clocked ternary T-FF	3	1	2	6
Clocked r-val. T-FF	r	1	2	r + 3
Edge-Trig. D-FF	13	2	3	25

6. References

- [1] R. Landauer, "Irreversibility and heat generation in the computing processes" IBM J. Res. Develop. July 1961.
- [2] Muthukrishnan, A., Stroud, Jr., C. R. (2000). Multivalued logic gates for quantum computation. Physical Review A, Vol. 62, No. 5, 052309/1-8.
- [3] Al-Rabadi, A., "New Multiple-Valued Galois Field Sum-of-Product Cascades and Lattices for Multiple-Valued Quantum Logic Synthesis," 6th Int. Symp. on Representations and Methodology of Future Computing Technologies, March 2003, pp. 171-182.
- [4] Khan, Mozammel H. A., Marek A. Perkowski and Pavel Kerntopf, "Multi-output Galois Field Sum of Products (GFSOP) Synthesis with New Quantum Cascades," Proc. International Symposium on Multiple-Valued Logic, May 2003, pp. 146-153.
- [5] Picton, P., "A Universal Architecture for Multiple-Valued Reversible Logic," MVL Journal, 5, 2000, pp. 27-37.
- [6] T. Toffoli, "Reversible computing", Tech memo MIT/LCS/TM-151, MIT Lab for Computer Science (1980).
- [7] E. Fredkin, T. Toffoli "Conservative logic", Int. J. Theo. Phys., 21 (1982) 219.
- [8] Anindita Banerjee, Anirban Pathak "On the Synthesis of Sequential Reversible Circuit" arXiv:0707.4233v1 [quant-ph] 28 Jul 2007
- [9] Goldberg, D. E. (1989). "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison Wesley.
- [10] Miller, D. M., D. Maslov, and G. W. Dueck, "Synthesis of Quantum Multiple-Valued Circuits," Journal of Multiple-Valued Logic and Soft Computing, submitted Feb. 2004.
- [11] M.Mohamadi, M.Eshghi, K.Navi, "Optimizing the Reversible Full Adder Circuit", IEEE EWDTs, Yerevan, Sep.7-10,2007, pp. 312-315
- [12] Mozammel H. A. Khan and Marek Perkowski, "Genetic algorithm based synthesis of multi-output ternary functions using quantum cascade of generalized ternary gates" IEEE Congress on Evolutionary Computation, 2004. CEC2004.