Improved Synthesis of Reversible Sequential Circuits

Mozammel H A Khan
Department of Computer Science and Engineering
East West University
Aftabnagar, Dhaka 1212, Bangladesh
mhakhan@ewubd.edu

Jacqueline E. Rice
Department of Mathematics and Computer Science
University of Lethbridge
Lethbridge, AB T1K 3M4, Canada
j.rice@uleth.ca

***Abstract*—Synthesis of reversible sequential circuits is a very new research area. It has been shown that such circuits can be implemented using emerging technologies such as quantum dot cellular automata. Earlier work uses traditional designs for sequential circuits and replaces the flip-flops and the gates with their reversible counterparts. Our earlier work used a direct feedback method without any flip-flops, which proved to be better than the replacement technique in terms of both quantum cost and ancilla inputs. This work is an improved version of our direct feedback method, which uses a different approach to the reversible mapping of sequential circuits. Design examples show that the proposed method produces better results than our earlier method in terms of both quantum cost and ancilla inputs.**

Keywords-reversible logic, sequential circuit, direct feedback method, modified next state, counter, register

I. INTRODUCTION

It has been generally said that feedback is not allowed in reversible circuits and, therefore, that reversible sequential circuits are not possible. However, Toffoli [1] argued that if the feedback is provided through a delay element, then the feedback information will be available as the input to the reversible combinational circuit in the next clock cycle and thus a reversible sequential circuit is possible. Moreover, Thapliyal *et al*. [2] showed that reversible sequential circuits can be implemented using quantum dot cellular automata (QCA) technology.

This paper presents a further improvement of the work in [3]. We propose a new technique for representing the next states using exclusive-OR sum-of-product (ESOP) expressions. ESOPs are similar to the traditional sum-of-product (SOP) representation except that product terms are combined with exclusive-OR operators rather than OR operators. Using this technique we present designs for a four-bit falling-edge triggered up/down counter with asynchronous loading and a four-bit falling-edge triggered universal register.

A. Reversible Circuits

The term reversible function refers to a function that is bijective. A logic gate can be considered reversible if the function it computes is bijective [4]. Our work uses only Toffoli, Feynman and Fredkin gates. The Toffoli gate shown in Fig. 1(a) behaves as follows: $p = x;\ q = y;\ r = z \oplus xy$. The inputs x and y are called control inputs and the input z is referred to as the target input. The Fredkin gate in Fig. 1(b) behaves as follows: $p = x;\ q = z$ and $r = y$ iff $x = 1$, otherwise $q = y$ and $r = z$ (controlled swap). The input x is referred to as the control input while the inputs y and z are called target inputs. The Feynman gate is a variant of the Toffoli gate in which only one control is used (either of the x or y line). A NOT gate can be obtained by removing both controls from the Toffoli gate. Reversible circuits are created by cascading reversible gates.

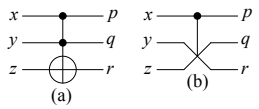

Figure 1. (a) Toffoli gate. (b) Fredkin gate.

B. Related Work

Limited attempts have been made in the field of sequential reversible circuit synthesis [2], [3], [5]–[11]. [5]–[8] present reversible designs for latches and flip-flops using reversible gates and suggest that sequential circuits be synthesized by replacing the latches, flip-flops, and gates of the combinational part of the traditional irreversible designs with their reversible counterparts. Following this replacement approach, the work in [9] presents a design for a four-bit falling-edge-triggered universal register and the work in [10] presents a design for a four-bit level-triggered up counter with asynchronous parallel load. The work in [2] again uses the replacement technique, and demonstrates that reversible circuits can be implemented using QCA technology with off-line testing of the QCA-based sequential circuit. The work in [11] offers the first attempt to synthesize a sequential circuit using direct feedback of the state output and without using any flip-flops. [11] also presents a design for a level-triggered up counter using positive polarity Reed–Muller expressions for representing the next states. The up counter design is more efficient than the replacement design in [10] in terms of both quantum cost (number of elementary quantum gates needed in the design) and garbage outputs

978-1-4799-5341-7/16/$31.00 ©2016 IEEE

(outputs that are not used for the intended circuit realization). The next step to this work is presented in [3], which details the direct design of arbitrary sequential circuits using pseudo Reed-Muller expressions for representing the next states. This work also introduces modifications making the circuit falling-edge triggered and asynchronous loadable. The up/down counter in [3] is better than that in [10] in terms of both quantum cost and garbage outputs, and the universal register in [3] is better than that in [9] in terms of both quantum cost and garbage output.

The remainder of the paper is organized as follows: in section II, we present our technique for representing the next state using ESOP expressions. We also present designs of a four-bit falling-edge triggered up/down counter with asynchronous loading and a four-bit falling-edge triggered universal register in sections III and IV respectively. Finally, we conclude the paper in section V.

II. Representing the Next State

Let Q and Q^+ be the present state and the next state of a sequential circuit, respectively. The next state Q^+ can be expressed as a function of the present state Q and the next state Q^+ as

$$Q^+ = Q \oplus Q \oplus Q^+ = Q \oplus Q^*, \quad (1)$$

where

$$Q^* = Q \oplus Q^+. \quad (2)$$

We call Q^* the modified next state. In our proposed technique the modified next state is determined using (2) and expressed as a minimized ESOP expression. The next state is then expressed using (1).

III. Example 1: 4-bit up/down counter

In this section we illustrate our technique by designing a four-bit falling-edge trigged up/down counter with asynchronous loading. The truth table representing the next states and the modified next states of a four-bit up counter is shown in Table I. The modified next states of the four-bit up counter from Table I are minimized as ESOP expressions as follows:

$$Q3^* = Q2Q1Q0 \quad (3)$$

$$Q2^* = Q1Q0 \quad (4)$$

$$Q1^* = Q0 \quad (5)$$

$$Q0^* = 1 \quad (6)$$

Similarly, the modified next states of a four-bit down counter are minimized as ESOP expressions as follow:

$$Q3^* = Q2'Q1'Q0' \quad (7)$$

$$Q2^* = Q1'Q0' \quad (8)$$

$$Q1^* = Q0' \quad (9)$$

$$Q0^* = 1 \quad (10)$$

The four-bit falling-edge triggered up/down counter with asynchronous loading is designed using the ESOP expressions of (3) to (10) and the resulting reversible circuit is shown in Fig. 2. In Fig. 2, C is the clock input. The input L performs the asynchronous load with $L = 0$ for normal operation and $L = 1$ for asynchronous loading. The input M determines the count direction with $M = 0$ for up and $M = 1$ for down. The operation of the circuit in Fig. 2 is discussed below:

TABLE I. Truth Table Representing the Next States and the Modified Next States of a Four-Bit Up Counter

Present State $Q3Q2Q1Q0$	Next State $Q3^+Q2^+Q1^+Q0^+$	Modified Next State $Q3^*Q2^*Q1^*Q0^*$
0000	0001	0001
0001	0010	0011
0010	0011	0001
0011	0100	0111
0100	0101	0001
0101	0110	0011
0110	0111	0001
0111	1000	1111
1000	1001	0001
1001	1010	0011
1010	1011	0001
1011	1100	0111
1100	1101	0001
1101	1110	0011
1110	1111	0001
1111	0000	1111

1. ***C* = 1 and *L* = 0: maintain present states.** The Feynman gates of the Feedback section of the circuit make copies of the present state outputs $Q3$, $Q2$, $Q1$, $Q0$.These copies are then fed back through the two Feynman gates in cascade in the Modified Next State Logic section where the values are unchanged by the two cascaded Feynman gates. They then move still unmodified through the Next State Logic section, and become inputs to the Fredkin gates in the Falling-Edge Trigger section of the circuit where the target inputs are swapped as a result of the control value $C = 1$. The values are not modified in the Asynchronous Load section since the control value for those gates (L) has the value 0, and thus present states are maintained.
2. ***C* = 1 and *L* = 1: asynchronous load.** The data inputs $D3$, $D2$, $D1$, $D0$ are loaded to the present state outputs $Q3$, $Q2$, $Q1$, $Q0$ (respectively) through the Fredkin gates of the Asynchronous Load section of the circuit. L having the value 1 will result in the swaps taking place with the loaded values, and if we then set $L = 0$ the asynchronously loaded present state outputs will be maintained as described above.
3. ***M* = 0: modified next state generation (count up).** The six M-controlled Feynman gates of the Modified Next State Logic section will not modify the logic values of their targets and the fed-back state values $Q3$, $Q2$, $Q1$, $Q0$ will remain unchanged. These fed-back state values are used in the Modified Next State Logic section to generate modified next states $Q3^*$, $Q2^*$, $Q1^*$, $Q0^*$ using (3) to (6).
   ***M* = 1: count down.** The first three Feynman gates controlled by M complement the fed-back present states $Q3$, $Q2$, $Q1$, $Q0$. These complemented values are used

in the Modified Next State Logic section to generate modified next states $Q3^*$, $Q2^*$, $Q1^*$, $Q0^*$ using (7) to (10). The last three Feynman gates of the Modified Next State Logic section restore the fed-back present states for use in the Next State Logic section of the circuit.

4. **Next State Generation:** The Next State Logic section generates the next states $Q3^+$, $Q2^+$, $Q1^+$, $Q0^+$ using formula (1), which is implemented with Feynman gates.
5. ***C* = 0 and *L* = 0: Falling-Edge Triggering.** This functionality allows the transfer of the generated next states $Q3^+$, $Q2^+$, $Q1^+$, $Q0^+$ to the present state outputs $Q3$, $Q2$, $Q1$, $Q0$. The Fredkin gates of the Falling-Edge Trigger section pass the generated next states to the present state outputs; the gate does not swap any values since the control C = 0. The Fredkin gates of the Asynchronous Load section also do not alter the values since their control line L has the value 0. After transferring the generated next states to the present state outputs setting C = 1 maintains the state outputs unchanged as described above.

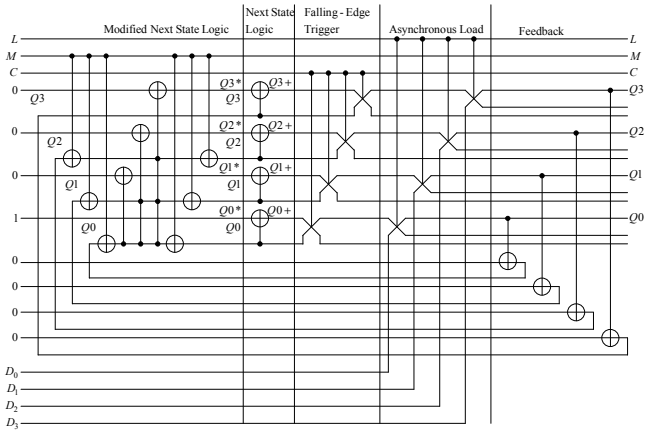

Figure 2. Reversible realization of the four-bit falling-edge triggered up/down counter with asynchronous load.

The circuit in Fig. 2 requires 8 constant-initialized inputs other than the primary inputs required for synthesizing the circuit. These are referred to as *ancilla* lines. It is assumed that reversible gates will be designed using some quantum technology and thus designs are generally measured in terms of their quantum cost. The quantum cost of a Feynman gate is one while the quantum cost of a three-input Toffoli gate is five, as is the quantum cost of a Fredkin gate [12], [13]. In this work we compute the quantum cost of Toffoli gates with more than three inputs as proposed in [14]. Using these individual gate costs the quantum cost of Fig. 2 is 74. The circuit complexity of Fig. 2 is compared with that in [3] in Table III.

IV. Example 2: 4-bit Universal Register

In this section we illustrate the synthesis process for a four-bit falling-edge triggered universal register. The truth table representing the next states and the modified next states of a four-bit serial-in serial-out right-shift register is shown in Table II, where DR is the serial data input. The modified next states $Q3^*$, $Q2^*$, $Q1^*$, $Q0^*$ from Table II are minimized as ESOP expressions as follows:

$$Q3^* = DR \oplus Q3 \tag{11}$$

$$Q2^* = Q3 \oplus Q2 \tag{12}$$

$$Q1^* = Q2 \oplus Q1 \tag{13}$$

$$Q0^* = Q1 \oplus Q0 \tag{14}$$

Using (1) the next states corresponding to (11) to (14) can be determined as follows:

$$Q3^+ = DR \oplus Q3 \oplus Q3 = DR \tag{15}$$

$$Q2^+ = Q3 \oplus Q2 \oplus Q2 = Q3 \tag{16}$$

$$Q1^+ = Q2 \oplus Q1 \oplus Q1 = Q2 \tag{17}$$

$$Q0^+ = Q1 \oplus Q0 \oplus Q0 = Q1 \tag{18}$$

TABLE II. Truth Table Representing the Next States and the Modified Next States of a Four-Bit Serial-In Serial-Out Right-Shift Register

Present State $DRQ3Q2Q1Q0$	Next State $Q3^+Q2^+Q1^+Q0^+$	Modified Next State $Q3^+Q2^+Q1^+Q0^*$	Present State $DRQ3Q2Q1Q0$	Next State $Q3^+Q2^+Q1^+Q0^+$	Modified Next State $Q3^+Q2^+Q1^+Q0^*$
00000	0000	0000	10000	1000	1000
00001	0000	0001	10001	1000	1001
00010	0001	0011	10010	1001	1011
00011	0001	0010	10011	1001	1010
00100	0010	0110	10100	1010	1110
00101	0010	0111	10101	1010	1111
00110	0011	0101	10110	1011	1101
00111	0011	0100	10111	1011	1100
01000	0100	1100	11000	1100	0100
01001	0100	1101	11001	1100	0101
01010	0101	1111	11010	1101	0111
01011	0101	1110	11011	1101	0110
01100	0110	1010	11100	1110	0010
01101	0110	1011	11101	1110	0011
01110	0111	1001	11110	1111	0001
01111	0111	1000	11111	1111	0000

Similarly, the next states of a four-bit serial-in serial-out left-shift register can be determined as follows, where DL is the serial data input:

$$Q3^+ = Q2 \tag{19}$$

$$Q2^+ = Q1 \tag{20}$$

$$Q1^+ = Q0 \tag{21}$$

$$Q0^+ = DL \tag{22}$$

The four-bit falling-edge triggered universal register is designed using the next state expressions of (15) to (22) and the resulting reversible circuit is shown in Fig. 3. Multiplexing is needed for implementing right-shift and left-shift: between DR and $Q2$ for $Q3^+$, between $Q3$ and $Q1$ for $Q2^+$, between $Q2$ and $Q0$ for $Q1^+$, and between $Q1$ and DL for $Q0^+$. This is implemented using four Fredkin gates controlled by the shift direction control input M in the Next State Logic section of the circuit. Implementations of the

other sections are similar to those in Fig. 2. The different modes of operation are discussed below:

1. ***C* = 1 and *L* = 0: Serial-In Serial-Out Register.** If *M* = 0, expressions (15) to (18) are implemented by the Next State Logic section and data is shifted right. If *M* = 1, expressions (19) to (22) are implemented and data is shifted left. When *C* is changed to 0, the next states are passed to the present state outputs through the Asynchronous Load section of the circuit.
2. **Serial-In Parallel-Out Register:** The operation is similar to step 1 and the present state outputs are taken in parallel.
3. ***C* = 1 and *L* = 1: Parallel-In Parallel-Out Register.** The parallel input values *D*3, *D*2, *D*1, *D*0 are loaded to the present state outputs through the Asynchronous Load section by setting *L* = 1 and then changing to *L* = 0. Outputs are taken in parallel.
4. **Parallel-In Serial-Out Register:** The asynchronous load operation is similar to step 3. After setting *L* = 0, if *C* is changed to 0, the states will be shifted to either right or left based on the value of *M*.

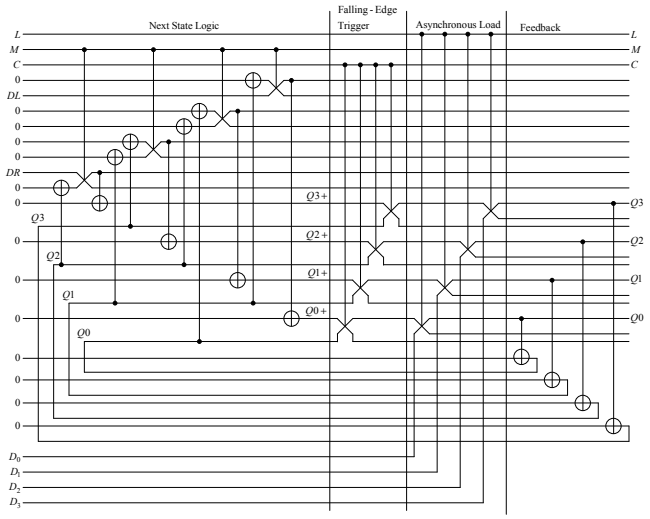

Figure 3. Reversible realization of the four-bit falling-edge triggered universal register.

The quantum cost of the circuit in Fig. 3 is 74 and 14 ancilla lines are required. The circuit complexity of Fig. 3 is compared with that in [3] in Table III.

V. Conclusion

In this work an improved synthesis approach for sequential reversible circuits is presented. Design examples of a four-bit falling-edge triggered up/down counter with asynchronous loading and a four-bit falling-edge triggered universal register are shown. We compare the quantum cost and the ancilla inputs of the two designs with those offered in [3]. The new counter design saves 21.28% in quantum cost with the same number of ancilla inputs as compared to the design in [3]. The new register design saves 33.93% in quantum cost with a 16.67% increase in ancilla inputs over that in [3]. Future work includes automation of the mapping process and consideration of testability.

TABLE III. Comparison of Circuit Complexity of the Proposed Method with that in [3]

Circuit	Present work		Work in [3]		% Improvement over [3]	
	QC	AI	QC	AI	QC	AI
Four-bit falling-edge trigered up/down counter with asynchronous load	74	8	94	8	21.28	0
Four-bit falling-edge triggered universal register	74	14	112	12	33.93	-16.67

QC = quantum cost, AI = ancilla inputs

Acknowledgments

The first author has prepared this work at the University of Lethbridge, Lethbridge, AB, Canada while on sabbatical from East West University, Dhaka, Bangladesh.

References

[1] T. Toffoli, "Reversible computing," MIT Lab. Comput. Sci., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TM-151, 1980.

[2] H. Thapliyal, N. Ranganathan, and S. Kotiyal, "Design of testable reversible sequential circuits," *IEEE Trans. VLSI Syst.*, vol. 21, no. 7, pp. 1201-1209, 2013.

[3] M. H. A. Khan, "Design of reversible synchronous sequential circuits using pseudo Reed-Muller expressions," *IEEE Trans. VLSI Syst.*, vol. 22, no. 11, pp. 2278-2286, 2014.

[4] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, pp. 710–722, Jun. 2003.

[5] J. E. Rice, "A new look at reversible memory elements," in *Proc. IEEE ISCAS*, May 2006, pp. 243–246.

[6] S. K. S. Hari, S. Shroff, S. N. Mohammad, and V. Kamakoti, "Efficient building blocks for reversible sequential circuit design," in *Proc. 49th IEEE MWSCAS*, Aug. 2006, pp. 437–441.

[7] H. Thapliyal and A. P. Vinod, "Design of reversible sequential elements with feasibility of transistor implementation," in *Proc. ISCAS*, 2007, pp. 625–628.

[8] M.-L. Chuang and C.-Y. Wang, "Synthesis of reversible sequential elements," *ACM J. Emerg. Technol.*, vol. 3, no. 4, pp. 1–19, 2008.

[9] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 4, pp. 14:1–14:35, 2008.

[10] M. Haghparast and M. S. Gharajeh, "Design of a nanometric reversible 4-bit binary counter with parallel load," *Austral. J. Basic Appl. Sci.*, vol. 5, no. 7, pp. 63–71, 2011.

[11] M. H. A. Khan and M. Perkowski, "Synthesis of reversible synchronous counters," in *Proc. 41st IEEE ISMVL*, May 2011, pp. 242–247.

[12] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, *et al.*, "Elementary gates for quantum computation," *Phy. Rev. A*, vol. 52, no. 5, pp. 3457–3467, 1995.

[13] J. A. Smolin and D. P. DiVincenzo, "Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate," *Phys. Rev. A*, vol. 53, no. 4, pp. 2855–2856, 1996.

[14] D. M. Miller, R. Wille, and Z. Sasanian, "Elementary quantum gate realizations for multiple-controlled Toffoli gates," in *Proc. 41st IEEE ISMVL*, May 2011, pp. 288–293.