Synthesis of Reversible Sequential Elements*

Min-Lun Chuang   Chun-Yao Wang

Department of Computer Science,

National Tsing Hua University, HsinChu, Taiwan R.O.C.

{mr934327,wcyao}@cs.nthu.edu.tw

Abstract – To construct a reversible sequential circuit, reversible sequential elements are required. This work presents novel designs of reversible sequential elements such as D latch, JK latch, and T latch. Based on these reversible latches, we also construct the designs of the corresponding flip-flops. Comparing with previous work, the implementation cost of our new designs, including the number of gates and the number of garbage outputs is considerably reduced.

I. Introduction

Power consumption is a very important issue in modern VLSI designs. Part of the problem of power dissipation is resulted from the technological nonideality of switches and materials. Therefore, fabrication processes have been continually improved to reduce power dissipation. The other part of the problem arises from Landauer's principle [5]. Landauer's principle states that logic computations that are irreversible necessarily generate at least $k \cdot T \cdot \log 2$ Joules of heat energy for every bit of information loss, where k is Boltzmann's constant and T the absolute temperature at which computation is performed. This part of energy dissipation is independent of what the underlying technology is. Although power dissipation due to information loss is negligible under current technologies, this part of energy dissipation would become essential in 2020 or earlier if the Moore's Law continues being in effect [20]. This is due to the increasing density in computer hardware. People believe that there will be more than 10^{17} logic devices packed in a cubiccentimeter [11] in the future. Therefore, 10^{17} conventional devices operating at room temperature (T=300° k), at a frequency of 10 GHz would dissipate more than 3,000,000 Watts of power. Moreover, a computer with 1,000 times as many logic elements would still be of reasonable size, but it would dissipate 3,000,000,000 Watts of power [11].

Reversible computing does not result in information loss during the computation process. Thus, it naturally takes care of heating generated due to information loss. Bennett [1] shows that zero energy dissipation would be possible only if the gates in a network are all reversible. As a result, reversibility will become an essential property in future
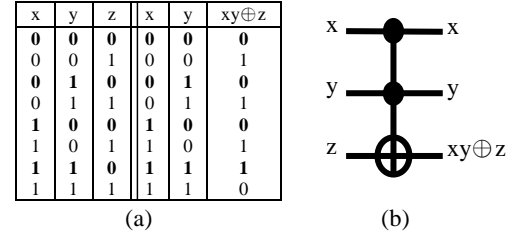
Figure 1: 3-bit Toffoli gate (a) truth table; (b) symbol.

x	y	z	x	y	xy⊕z
0	**0**	**0**	**0**	**0**	**0**
0	0	1	0	0	1
0	**1**	**0**	**0**	**1**	**0**
0	1	1	0	1	1
1	**0**	**0**	**1**	**0**	**0**
1	0	1	1	0	1
1	**1**	**0**	**1**	**1**	**1**
1	1	1	1	1	0

circuit design. Reversible logic has been applied to various future technologies, such as ultra-low-power CMOS design [16], optical computing [3], quantum computing [12], and nanotechnology [6]. These technologies exploit reversible gates to reduce the power consumption [13].

To realize a function with reversibility, many reversible logic synthesis algorithms have been proposed in [4][8][9][13][19]. But most of them address this issue from the aspect of combinational logic. To synthesize reversible sequential circuits, reversible sequential elements such as latches and flip-flops are necessary. Thus, this paper proposes novel designs of reversible sequential elements such as clocked D latch, clocked JK latch, and clocked T latch. The corresponding flip-flop designs are also introduced.

The remainder of this paper is organized as follows. In Section 2, we introduce the background of reversible logic. In Section 3, we review some existing implementations of reversible sequential elements. Then we propose our new designs of these reversible sequential elements and verify the functionalities of them in Section 4. The statistics and comparison of these new designs and previous work are presented in Section 5. Section 6 concludes the work.

II. Background

Definition 1: *A gate is reversible if and only if the (Boolean) function is bijective.*

It means that a reversible function f:$(x_1, x_2, \ldots, x_n) \rightarrow (y_1, y_2, \ldots, y_m)$ satisfies the condition of one-to-one and onto mapping between the input and output domains.

It is obvious that the conventional logic gates are almost irreversible. Among the commonly used gates, only NOT gate is reversible. AND gate and OR gate are irreversible because they violate the requirement of reversible function. One way to make the AND function reversible is to add one

input and two outputs as shown in Figure 1(a). The AND function can be obtained in the third output column $xy \oplus z$ of Figure 1(a), when setting $z = 0$. The truth table of AND function is shown in bold.

Definition 2: *A garbage bit is an additional output that makes an n-input m-output function reversible.*

In the reversible AND function as shown in Figure 1(a), the outputs x, y are garbage outputs which are used to make the function reversible.

A set of reversible gates is needed to synthesize reversible circuits. Several reversible gates have been proposed in the centuries. Here we introduce Toffoli gate and Fredkin gate because they will be used in our work.

Toffoli gate: The truth table of 3-bit Toffoli gate is shown in Figure 1(a) and its symbol is shown in Figure 1(b) [2]. The function of the third_column is $xy \oplus z$. That means when $x=y=1$, the output is z, otherwise the output is z. This gate can be used to realize a 2-input reversible AND function by setting z as a constant 0 as mentioned. A Toffoli gate can be generalized as TOF(C;T), where C is a set of control variables $\{x_1, x_2, \ldots, x_{n-1}\}$, T is a target variable $\{x_n\}$ and $C \cap T = \emptyset$. $TOF(x_1, x_2, \ldots, x_{n-1}; x_n)$ is a gate which maps a Boolean pattern $(x_1, x_2, \ldots, x_{n-1}, x_n)$ to $(x_1, x_2, \ldots, x_{n-1}, x_1 x_2 \ldots x_{n-1} \oplus x_n)$.

Fredkin gate: Fredkin gate is also called controlled SWAP gate. Figure 2(a) is the symbol of Fredkin gate and Figure 2(b) is its truth table [2]. Its behavior can be described as follows: if the control bit x is set to 1, the outputs of y and z are swapped, otherwise they remain unchanged.

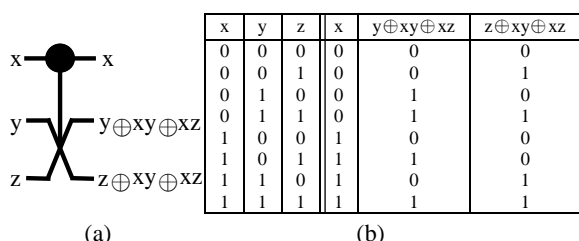| x	y	z	x	$y \oplus xy \oplus xz$	$z \oplus xy \oplus xz$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

(a)　　　　　　　　　　(b)

Figure 2: Fredkin gate (a) symbol; (b) truth table.

A restriction on reversible logic synthesis has to be followed [18]: The fanout count of a signal net must equal one. If two copies of one signal are needed, a duplication is necessary.

This restriction is due to the fact that fanout structure itself is not reversible. For fanout, the number of input signal is one, but there are two or more output signals. Therefore, for this restriction, we use a 2-bit Toffoli gate (also called Feynman gate) to duplicate a signal. The symbol of a 2-bit Toffoli gate and its truth table are shown in Figure 3(a) and 3(b), respectively. The function of the second output column is $x \oplus y$. If y is set as a constant 0, a copy of input variable x

will be obtained in the second output, which is shown in bold. Therefore, the fanout structure in a conventional network can be implemented in this way.

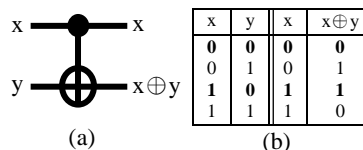| x	y	x	$x \oplus y$
0	**0**	**0**	**0**
0	1	0	1
1	**0**	**1**	**1**
1	1	1	0

(a)　　　　　　　(b)

Figure 3: 2-bit Toffoli gate (a) symbol; (b) truth table.

There are two objectives in reversible circuit synthesis:
1. Minimize the number of gates: the number of gates gives a simple estimation of the implementation cost of a reversible circuit.
2. Minimize the number of garbage outputs: we need extra implementation cost (area and power) for those garbage outputs in reversible circuits. Minimizing the number of garbage outputs leads to minimizing area and power.

III. Previous Work

Fredkin and Toffoli [2] discussed some topics about reversible sequential network and first introduced a sequential element in the form of JK flip-flop. However, they did not discuss other popular sequential elements such as D latch, D flip-flop, etc.

Picton [14] proposed a new design of reversible RS latch without clock signal. This work decomposes a clockless RS latch into two conventional NOR gates as shown in Figure 4. Then each NOR gate is replaced by a Fredkin gate, which can implement the function of NOR gate. This new structure of RS latch is considered reversible because it is entirely constructed by reversible gates as shown in Figure 5. In this work, they did not discuss other reversible sequential elements either.
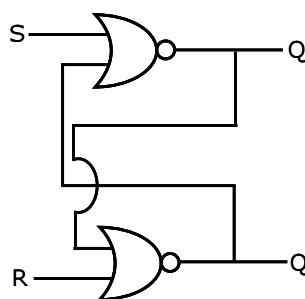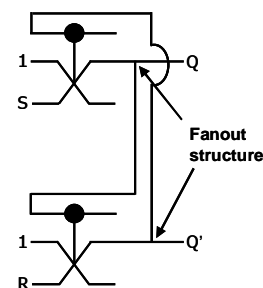

Figure 4: The traditional RS latch.　　Figure 5: The reversible RS latch proposed by Picton [14].

A recent work proposed by Thapliyal et al. [17] exploited similar approach, *direct transformation*, to design other sequential elements such as D latch, JK latch, etc. In addition, not only Fredkin gate, but also some other reversible gates are used to implement conventional logic gates such as NOR gate, AND gate, etc. Although many reversible sequential

elements are considered in this work, the implementation cost of them is quite large because they did not further optimize these reversible sequential elements.

Rice [15] recently proposed a new design of reversible RS latch to avoid fanout structures in the original reversible RS latch design proposed in [14]. The new structure of reversible RS latch is shown in Figure 6. Moreover, since some other sequential elements such as D flip-flop, JK flip-flop, and T flip-flop can be built by RS latch, this work constructs other reversible sequential elements based on this new reversible RS latch design. *Direct transformation* approach is also adapted in this work to design sequential elements.

These three work all use *direct transformation* approach to design reversible sequential elements. Direct transformation approaches, however, would cause a large number of gates and garbage outputs required. Therefore, this paper proposes a new approach to the construction of reversible sequential elements.
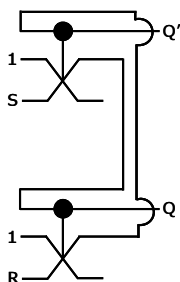

Figure 6: The reversible RS latch proposed by Rice [15].

IV. Novel Designs of Reversible Sequential Elements

This section presents our new designs of reversible sequential elements. Also, our approach to getting these results is introduced in detail.

A. Clocked Reversible Latches

Our synthesis method is a truth table extension method. Unlike the direct transformation based method, we do not replace those irreversible gates with the reversible ones within a sequential element. Alternatively, we extend the original irreversible truth table of a sequential element to an augmented reversible one. We take D latch as an example. First, we get the truth table of D latch and make it reversible. Obviously, the truth table of D latch in Table 1 is not a reversible function. The mapping between the input and output domains is not one-to-one. Therefore we have to add some garbage outputs to make it reversible. The minimum number of garbage outputs required for reversibility is $\lceil \log(q) \rceil$, where q is the maximum number of times an output pattern repeated in the truth table [7]. In this case, 0 and 1 are repeated 4 times in the output column Q_{n+1}.

Therefore, we add $\lceil \log(4) \rceil = 2$ output variables clk' and D' in the truth table and make the table reversible as shown in Table 2. Note that different values assigned to these two output columns will affect the result of our design. Under our assignments in Table 2, we observe that Table 2 is identical to Figure 2. Thus, a D latch can be modeled by a Fredkin gate. This means we only need one Fredkin gate to implement a reversible D latch. However, if we assign these values in different ways, the design may be different.

Table 1: The truth table of D latch.

clk	D	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Table 2: The augmented reversible truth table of D latch.

clk	D	Q_n	clk'	D'	Q_{n+1}
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Comparing with previous work, if we use direct transformation method to implement reversible D latch, the synthesis result would be not good. This is because a traditional D latch is built by many irreversible gates, using direct transformation method to construct a reversible D latch will cause a large number of gates and garbage outputs required.

After synthesizing this augmented reversible function, we need to consider the fanout problem. The input Q_n in the next state comes from the current output Q_{n+1}. Thus, an additional Q_{n+1} is needed for feedback. Here a Feynman gate is used to duplicate the output variable Q_{n+1}. Thus, the final structure of D latch is shown as Figure 7.
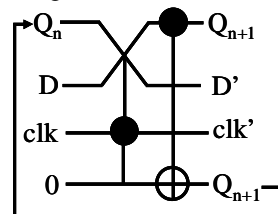

Figure 7: The complete implementation of reversible D latch.

We verify if this reversible D latch design exactly implements the behavior of a D latch. The leftmost part of Figure 8 shows the Boolean functions obtained from the augmented truth table of D latch in Table 2. To simplify the expression of Boolean equations, the symbol "C" is used to represent input variable clk. The rightmost part of Figure 8

shows the functions of implemented reversible D latch. These two expressions are identical, therefore, the functionality of our reversible D latch is correct.
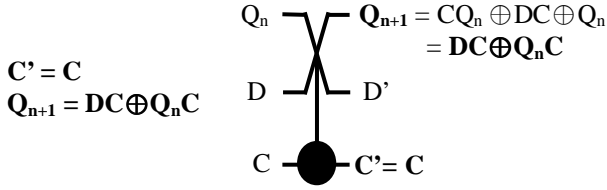

$$C' = C$$
$$Q_{n+1} = DC \oplus Q_n C$$

$$Q_{n+1} = CQ_n \oplus DC \oplus Q_n = DC \oplus Q_n C$$
$$C' = C$$

Figure 8: Functional verification on reversible D latch.

(a)

$$C' = C$$
$$Q_{n+1} = TC \oplus Q_n$$

$$Q_{n+1} = TC \oplus Q_n$$

clk	T	Q_n	clk'	T'	Q_{n+1}
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

(b) (c)
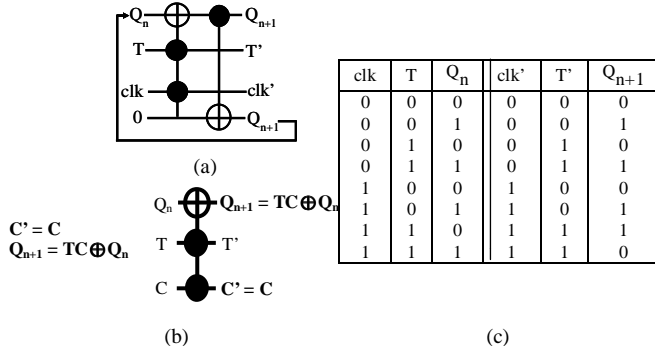
Figure 9: T latch (a) structure; (b) functional verification; (c) the augmented truth table.

Similarly, a reversible T latch can be modeled as a Toffoli gate using the same method. The implementation of reversible T latch is shown in Figure 9(a). Figure 9(b) shows the verification result. Figure 9(c) is the augmented truth table.

As for JK latch, because its function is quite complex, it is not easy to model its function using a single reversible gate. Therefore, we exploit a transformation based synthesis algorithm [9] to construct the reversible JK latch. First, we also derive the augmented reversible truth table as shown in Table 3. Then, we apply the transformation based synthesis algorithm to implement the reversible function. The philosophy of the transformation based algorithm is to cascade some reversible gates such that the output of the truth table is equal to the input. Next, we describe how to construct our reversible JK latch in detail.

Table 3: The augmented reversible truth table of JK latch

clk	J	K	Q_n	clk'	J'	K'	Q_{n+1}
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	**0**	**1**	**1**	**1**	**1**	**1**	**0**
1	1	0	0	1	1	0	1
1	1	0	1	1	0	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0

First, we inspect the augmented truth table in the lexicographical order until the first output assignment differs from the input assignment. The first output assignment which is not equal to input assignment in Table 3 is 1110.

Then we want to add some generalized Toffoli gates from the end of constructed circuit towards its beginning to make the output assignments be equal to input assignments. There are two rules when we choose a generalized Toffoli gate.

1. Deal with those bits that should become to 1 first: we want to change the output assignment 1110 to 1011. Hence the 2nd bit should be changed from 1 to 0 and the 4th bit should be changed from 0 to 1. Therefore, we deal with the 4th bit first.

2. Remain the output assignments which are prior to the current one intact: the output assignments prior to 1110 are identical to their corresponding input assignments, so we leave them unchanged. Using TOF(clk',J',K';Q_{n+1}) or TOF(clk',J';Q_{n+1}) are effective to invert the 4th bit of 1110 and leave the output assignments prior to it unchanged. In our design, we choose a TOF(clk',J';Q_{n+1}) and add it to the end of constructed circuit in this iteration. Note that this process might change other output assignments after 1110, such as 1101 or 1111. Nevertheless, we can reform them in the same way in the later iterations.

In each step, we choose an appropriate generalized Toffoli gate to synthesize the reversible function according to these two rules. The algorithm is terminated until all of the output assignments are equal to the input assignments. Figure 10 shows the process of synthesizing this reversible JK latch. After adding a generalized Toffoli gate in each step, we show those changed output assignments in bold. Note that the gates are identified sequentially from the output side to the input side. The resulting circuit has to be reversed and is shown in Figure 11(a), and the final structure of reversible JK latch is shown in Figure 11(b). The verification of reversible JK latch is shown in Figure 12. We use Figure 13 to illustrate the structure of a reversible sequential circuit with the new proposed reversible D latches. The combinational part and the sequential elements of a reversible sequential circuit have
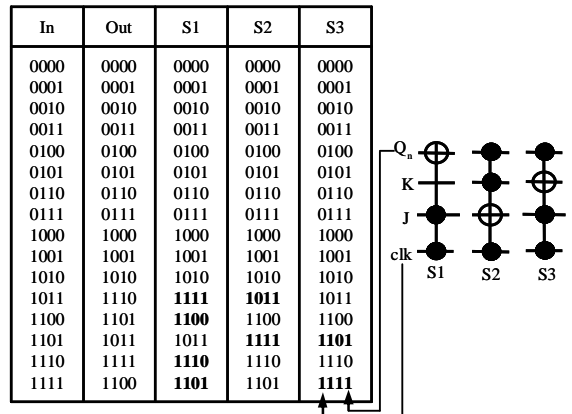
In	Out	S1	S2	S3
0000	0000	0000	0000	0000
0001	0001	0001	0001	0001
0010	0010	0010	0010	0010
0011	0011	0011	0011	0011
0100	0100	0100	0100	0100
0101	0101	0101	0101	0101
0110	0110	0110	0110	0110
0111	0111	0111	0111	0111
1000	1000	1000	1000	1000
1001	1001	1001	1001	1001
1010	1010	1010	1010	1010
1011	1110	**1111**	1011	1011
1100	1101	**1100**	1100	1100
1101	1011	1011	**1111**	**1101**
1110	1111	**1110**	1110	1110
1111	1100	**1101**	1101	**1111**

Figure 10: The synthesis process of reversible JK latch.

$$C' = C$$
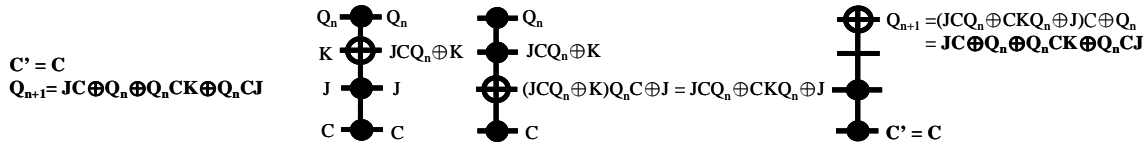$$Q_{n+1} = JC \oplus Q_n \oplus Q_n CK \oplus Q_n CJ$$

Figure 12: Functional verification on reversible JK latch.

(a)　　　　　　(b)
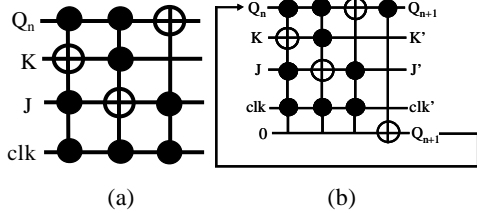
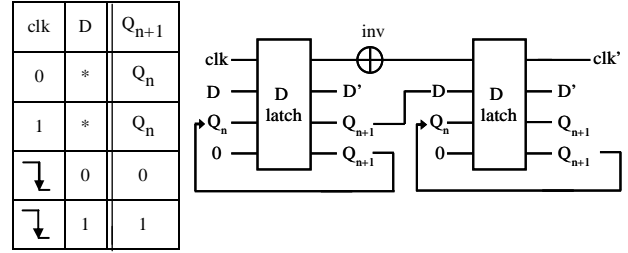Figure 11: JK latch (a) the reversible transition function; (b) the complete implementation.

Figure 13: An illustration of reversible sequential network.

Figure15: D flip-flop (a) behavior; (b) structure.

Figure16: T flip-flop (a) behavior; (b) structure.
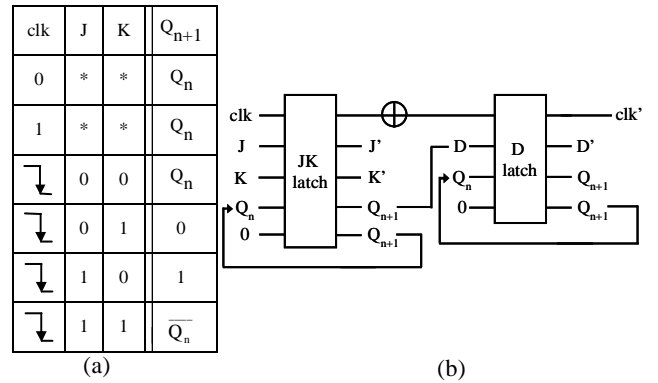
Figure17: JK flip-flop (a) behavior; (b) structure.

to be reversible. The clock signal of each reversible sequential element would be pulsed by a global clock source.

B. Clocked Reversible Flip-Flops

A flip-flop is an edge-triggered sequential element while a latch is a level sensitive sequential element. A traditional D flip-flop consists of two D latches and one inverter, and is shown in Figure 14 [10].The first D latch is called the master and the second one is called the slave. Since a reversible D latch has been built, a reversible D flip-flop can be constructed directly by replacing the D latches and inverter with its reversible versions. The behavior and structure of a reversible D flip-flop are shown in Figure 15(a) and Figure 15(b), respectively. We can trace the D flip-flop design and compare the function with its truth table. The behavior of implemented D flip-flop is the same as that of the truth table in Figure 15(a). The implementations of reversible T flip-flop and JK flip-flop are shown in Figure 16 and Figure 17, respectively.
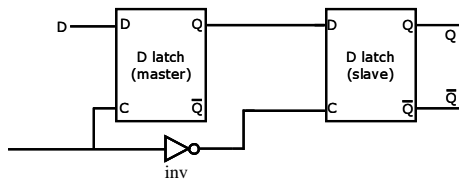

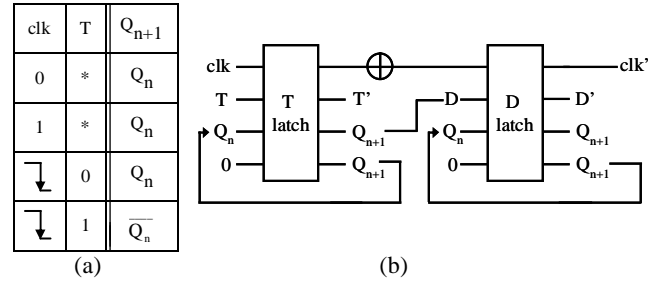Figure 14: An irreversible conventional D flip-flop.

V. The Discussion of Results

Table 4 shows the statistics and comparison of our new designs against that proposed in [17]. Table 5 is another statistics and comparison of our designs against another work proposed in [15]. In these two tables, column 1 shows the types of the sequential element. We use the number of gates and the number of garbage outputs as the cost functions to measure the quality of a design. Each table is separated into two parts by these two costs and each part has three columns. In Table 4, the column "Ours" shows the cost of our design. The column "[17]" shows the cost reported in [17]. The

Table 4: The statistics and comparison of our new designs and previous work [17].

Types	NO. of gates			NO. of garbage outputs		
	Ours	[17]	Ratio (%)	Ours	[17]	Ratio (%)
D latch	2	7	28.6	2	8	25.0
JK latch	4	10	40.0	3	12	25.0
T latch	2	10	20.0	2	12	16.6
JK flip-flop	7	18	38.9	4	21	19.0
Average	—	—	31.9	—	—	21.4

Table 5: The statistics and comparison of our new designs and previous work [15].

Types	NO. of gates			NO. of garbage outputs		
	Ours	[15]	Ratio (%)	Ours	[15]	Ratio (%)
D flip-flop	5	11	45.5	3	12	25.0
JK flip-flop	7	12	58.3	4	14	28.6
T flip-flop	5	13	38.5	3	14	21.4
Average	—	—	47.4	—	—	25.0

column "Ratio" is the percentage of Ours/[17]. For example, for a D latch design, our implementation has 2 gates while [17] has 7 gates. The ratio is 2/7=28.6%. The last row "Average" shows the averaged ratio among these different reversible sequential elements.

In [15], they do not summarize these two kinds of implementation costs of their reversible sequential elements. We count these numbers based on their designs and show them in Table 5.

According to the statistics in Table 4 and Table 5, the implementation cost of our designs is smaller than that of these two previous work.

VI. Conclusion

This paper proposes novel designs of basic reversible sequential elements such as latches and flip-flops. The design process is also introduced in detail. As comparing with previous work, the implementation costs of these new designs are more competitive. Thus, the resulting reversible sequential circuits are more cost efficient.

References

[1] C. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, 17: pp. 525-532, 1973.

[2] E. Fredkin and T. Toffoli, "Conservative logic," International Journal of Theoretical Physics, 21:pp. 219-253, 1982.

[3] E. Knill, R. Laflamme, and G. J. Milburn, "A scheme for efficient quantum computation with linear optics," Nature, pp. 46-52, 2001.

[4] P.Kerntopf, "A new heuristic algorithm for reversible logic synthesis," in Proc. of the IEEE Design Automation Conference, pp. 834-837, 2004.

[5] R. Landauer, "Irreversibility and heat generation in the computational process," IBM Journal of Research and Development, 5: pp. 183-191, 1961.

[6] R. C. Merkle, "Two types of mechanical reversible logic," Nanotechnology, 4:pp. 114-131, 1993.

[7] D. Maslov and G. W. Dueck, "Garbage in reversible design of multiple output functions," in Proc. of 6th International Symposium on Representations and Methodology of Future Computing Technologies, pp. 162-170, 2003.

[8] D. M. Miller, "Spectral and two-place decomposition techniques in reversible logic," in Proc. of the IEEE Midwest Symposium on Circuits and Systems, pp. II493–II496, 2002.

[9] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in Proc. of the IEEE Design Automation Conference, pp. 318-323, 2003.

[10] M. M. Mano, "Computer engineering: hardware design," Prentice-Hall, Englewood Cliffs, NJ, 1988.

[11] R. C . Merkle, and K. Drexler, "Helical logic," Nanotechnology, 7: pp.325-339, 1996.

[12] M. Nielsen and I. Chuang, "Quantum computation and quantum information," Cambridge University Press, 2000.

[13] M. Perkowski, L. Joziwak, A. Mixhchenko, A. Al-rabadi, A. Coppola, A. Buller, X. Song, M. Khan, S. Yanushkevich, V. P. Shmerko, and M. Chrzanowska-Jeske, "A general decomposition for reversible logic," in Proc. of Reed-Muller Workshop, pp. 119-138, 2001.

[14] P. Picton, "Multi-valued sequential logic design using Fredkin gates," Multiple-Valued Logic Journal, l.1:pp. 241-251, 1996.

[15] J. E. Rice, "A new look at reversible memory elements", in Proc. of the IEEE International Symposium on Circuits and Systems, 2006

[16] G. Schrom, "Ultra-low-power CMOS technology," PhD thesis, Technischen Universitat Wien, June 1998.

[17] H. Thapliyal and M. B. Srinivas, "A beginning in the reversible logic synthesis of sequential circuits," in Proc. of Military and Aerospace Programmable Logic Devices International Conference, 2005.

[18] T. Toffoli, "Reversible computing," Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci, 1980.

[19] G. Yang, X. Song, W. N.N. Hung, and M. A. Perkowski, "Fast synthesis of exact minimal reversible circuits using group theory," in Proc. of the IEEE Asia and South Pacific Design Automation Conference, pp. 1002-1005, 2005.

[20] V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, "Limits to binary logic switch scaling–a gedanken model," in Proc. of the IEEE, pp. 1934-1939, 2003.