

# A New Look at Reversible Memory Elements

to be presented at ISCAS2006

J. E. Rice  
Dept. of Math & Computer Science  
University of Lethbridge  
Alberta, Canada  
Email: j.rice@uleth.ca

**Abstract**—Although many researchers are investigating techniques to synthesize reversible combinational logic, there is little work in the area of sequential reversible logic. We present an analysis of a basic memory element, the RS-latch, and a number of possible implementations. We then go on to introduce four reversible flip-flop designs based on the reversible RS-latch implementation.

## I. INTRODUCTION

*Reversible computing*, a new paradigm in computing, has recently come to the forefront of many researchers' interest. The primary reason for this is the increasing demands for lower power devices. As our computing demands become more complex, the power requirements tend to increase. At the same time our society is moving more and more towards computing devices that are both *small* and *mobile*. Both of these demand that devices use lower power. Thus the industry is coming to an impasse.

Reversible computing has been introduced as a potential solution to this. As stated by Frank [1],

...computers based mainly on reversible logic operations can reuse a fraction of the signal energy that theoretically can approach arbitrarily near to 100% as the quality of the hardware is improved...

Reversible computing is not particularly new; in fact Tommaso Toffoli characterized reversible logic in his 1980 work "Reversible Computing" [2]. In this work he states that "*Using invertible logic gates, it is ideally possible to build a sequential computer with zero internal power dissipation.*" Toffoli proved that a finite automaton is reversible if its transition function is invertible. In order to realize a finite automaton with a reversible sequential circuit it suffices to build a reversible circuit realizing the transition function and use this as the combinational part of the sequential network.

However, despite the great potential of reversible logic and these endorsements from the leaders in the field, little to no work has been done in this area of sequential reversible logic. Many researchers are investigating combinational logic synthesis techniques but have seemed to overlook the necessity of memory elements if one is planning to implement many of our day to day circuits in reversible technologies.

This paper visits the small amount of existing work in the area, and extends this work to introduce a new reversible implementation of the RS-latch. We then go on to present

	inputs xyz	outputs x'y'z'	
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	100	4
5	101	101	5
6	110	111	7
7	111	110	6

TABLE I

THE TRUTH TABLE OF A 3X3 REVERSIBLE FUNCTION.

flip-flop implementations based on this latch. We hope that this work will begin to fill the gap in this area of research.

## II. BACKGROUND

### A. Reversible Logic

Before discussing sequential reversible logic we first present the basic concepts underlying the idea of reversible logic. According to Shende *et al.* [3],

*Definition 2.1:* a gate is reversible if the (Boolean) function it computes is bijective.

This means that a function is reversible if there is a one-to-one and on-to mapping from the inputs to the outputs (and vice versa) of the function. At the very least, a reversible function must have the same number of inputs as it does outputs. This means, for instance, that the traditional NOT gate is reversible, but that the traditional AND gate is not.

Table I shows the truth table for a 3x3 reversible function. In this figure the inputs are labeled as  $x$ ,  $y$  and  $z$  and the outputs are labeled  $x'$ ,  $y'$  and  $z'$ . The reason for this labeling of the outputs is that each output can be thought of as a transformed version of one of the inputs. Note that a reduced representation can be obtained from the truth table by simply listing the row numbers corresponding to the binary expansions represented by the inputs and by the outputs. If we assume that the inputs are given in numerical order from 0 to  $2^n - 1$  then we can list simply the decimal numbering of the outputs, in this case 0, 1, 3, 2, 4, 5, 7, 6. This is only possible because the function is bijective, and thus there is only one input combination resulting in any given output combination.

Table II lists the behaviour of each of the most commonly used reversible gates. The behaviour describes how each input becomes transformed to produce the outputs when the gate is applied. For instance, if the input values for  $x$ ,  $y$  and  $z$  are

gate	behaviour
Not	$(x) \rightarrow (x \oplus 1)$
Feynman	$(x, y) \rightarrow (x, x \oplus y)$
Toffoli	$(x, y, z) \rightarrow (x, y, xy \oplus z)$
swap	$(x, y) \rightarrow (y, x)$
Fredkin	$(x, y, z) \rightarrow (x, z, y)$ iff $x = 1$

TABLE II

THE BEHAVIOUR OF A SELECTION OF MORE COMMONLY USED REVERSIBLE LOGIC GATES.

110 then the output values will be  $x, y, xy \oplus z$  or 111 after the Toffoli gate is applied.

The symbols for the gates used in this paper are shown in Figure 1.

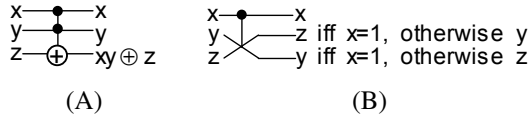


Fig. 1. (A) The Toffoli gate, and (B) the Fredkin gate.

### B. Previous Work

A number of researchers including Toffoli [2] and Frank [4] discuss the potential for sequential reversible logic, but do not present any structures for its realization. Fredkin and Toffoli [5] appear to be the first to suggest a conservative logic sequential element in the form of a JK flip-flop, and Picton [6] suggests a reversible RS-latch. He uses the basic Fredkin gate to build this latch, as shown in Figure 2. Our paper also concentrates on the use of a basic memory element such as the RS-latch, as it is the traditional building block for the clocked flip-flop structures that are our goal. The problem

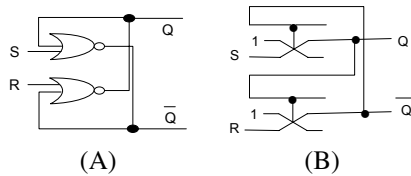


Fig. 2. The RS-latch a) built from traditional logic gates and b) built from two Fredkin gates as designed by Picton [6].

with Picton's model is that the concept of reversible logic is predicated on the fact that not only can one not allow the destruction of data (*e.g.* a signal value) but one can not allow the arbitrary creation of data. This means that fan-out is not permitted. As shown in Figure 2 Picton's latch requires two instances of fan-out. We address this issue further on.

More recent work in the area of sequential reversible logic has been published by Thapliyal, Srinivas and Zwolinski [7]. These authors also discuss the construction of basic memory elements such as flip-flops. However, their designs are ultimately different from those presented in this work, although possibly useful for comparison purposes.

### III. ADDRESSING THE PROBLEM OF FAN-OUT

As pointed out in the previous section, fan-out is required for Picton's model of a RS-latch built using two Fredkin

gates. An examination of the functionality of this latch leads to a relatively simple, almost cosmetic solution; instead of requiring fan-out from the  $Q$  and  $\bar{Q}$  signals to both the outputs and back to the inputs, why not use the outputs of the Fredkin gate for one of these uses. A solution demonstrating this is shown in Figure 3. State tables for the two reversible

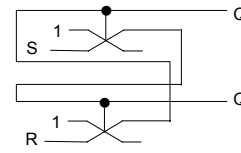


Fig. 3. Picton's RS-latch modified to avoid fan-out.

latches illustrate that the behaviours are the same, as shown in Table III.

	$\bar{Q}$	1	S	Q	1	R	$\bar{Q}^+$	$Q^+$
initial state:	0	1	0	0	1	0	0	0
next state:	1	1	0	1	1	0	1	1
next state:	0	1	0	0	1	0	0	0
							(unstable)	
initial state: (SET)	0	1	1	0	1	0	0	0
next state:	1	1	1	1	1	0	1	1
next state:	0	1	1	1	1	0	0	1
							(stable)	
initial state: (RESET)	0	1	0	0	1	1	0	0
next state:	1	1	0	1	1	1	1	1
next state:	1	1	0	0	1	1	1	0
							(stable)	

TABLE III

THE STATE TABLE FOR THE FREDKIN GATE-BASED RS-LATCH.

### IV. CONSTRUCTING A NEW REVERSIBLE RS-LATCH

One approach to determining reversible memory elements is to take existing memory elements, built from traditional logic, and replace the traditional components with reversible components. For instance, the NOR gate may be used in the design of the RS-latch. The NOR gate is clearly not reversible; one problem is that there is only one output and two inputs. Table IV (A) shows the additions required to create a reversible equivalent of the NOR gate's behaviour. Three outputs are required, and so an additional input must be added. Note that we set this input to 0 and so this is technically only half of the function's truth table; however, we are only interested in the behaviour defined for these inputs. If we rearrange the table as

inputs				outputs			inputs				outputs		
A	B	0		x	y	$\bar{A+B}$	$\bar{A}$	$\bar{B}$	0	x	y	$\bar{A+B}$	
000				0	0	1	110			0	0	1	
010				0	1	0	100			0	1	0	
100				1	0	0	010			1	0	0	
110				1	1	0	000			1	1	0	

(A)

(B)

TABLE IV

(A) A REVERSIBLE GATE WITH THE BEHAVIOUR OF THE NOR GATE, AND (B) WITH SLIGHT MODIFICATIONS TO MATCH THE BEHAVIOUR OF THE TOFFOLI GATE.

shown in Table IV (B) we achieve a reversible function which

happens to match the behaviour of the Toffoli gate. We can thus implement the RS-latch as shown in Figure 4<sup>1</sup>.

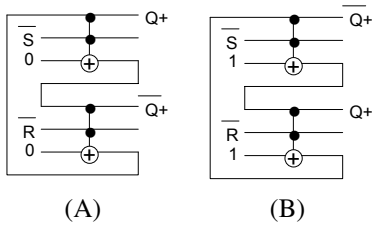


Fig. 4. Two alternate implementations for the RS-latch, using Toffoli gates instead of Fredkin gates.

Another approach to determining reversible memory elements is to define the desired element's state table, manipulate as needed in order to get a reversible function, and then perform reversible logic synthesis techniques. For instance, such a state table might begin as shown in Table V. After

$Q\bar{Q}SR$	$Q^+\bar{Q}^+XY$
0 0 - -	disallowed
0 1 0 0	0 1 0 0
0 1 0 1	0 1 0 0
0 1 1 0	1 0 0 0
0 1 1 1	1 1 0 0
1 0 0 0	1 0 0 1
1 0 0 1	0 1 0 1
1 0 1 0	1 0 1 0
1 0 1 1	1 1 0 1
1 1 0 0	1 1 1 0
1 1 0 1	0 1 1 1
1 1 1 0	1 0 1 1
1 1 1 1	1 1 1 1

TABLE V  
A STATE TABLE FOR THE RS-LATCH.

creating the state table the next stage is to create a cascade of reversible gates that effects the required transitions. However, in the case above the cascade quickly becomes quite involved, and so the designs from Figures 3 and 4 are a more efficient choice.

## V. REVERSIBLE CLOCKED FLIP-FLOPS

According to Sasao [8] there are four standard flip-flop designs. Although one can construct more complex latches with characteristics similar to those of the flip-flops, we argue that the usefulness of a non-clocked memory element is limited. Additionally, we investigate only edge-triggered flip-flops, since clocked latches may be difficult to use due to the restrictions inherent in the period and width of the clock pulse.

1) *Master-Slave flip-flops*: Figure 5 illustrates the standard construction, using traditional logic, of a master-slave flip-flop. Since previous sections have determined structures for reversible RS-latches, if we can also determine reversible structures with behaviours equivalent to the other elements in the master-slave flip-flop then these can be combined to create a reversible implementation. A Toffoli gate can be used for the AND structures, leaving only the problem of fan-out.

<sup>1</sup>The second implementation was suggested by Dr. Michael Miller.

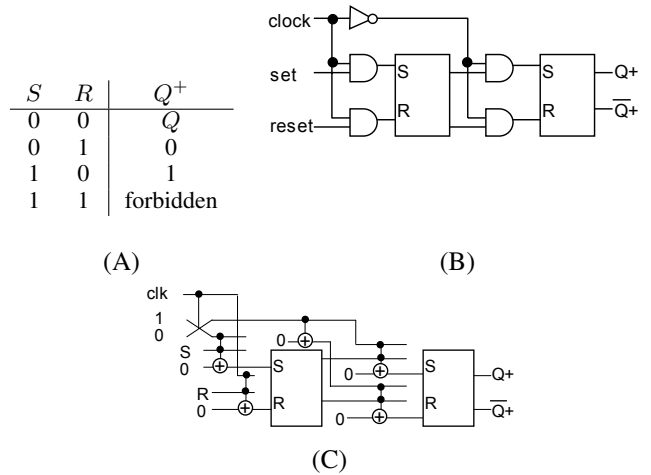


Fig. 5. The (A) behaviour, (B) a traditional logic implementation, and (C) a reversible equivalent for a master-slave flip-flop.

In Figure 5 (C) the fan-out from the clock is generated through the use of a Fredkin gate, which gives two identical outputs and one output that is the negation of the other two. This is useful since we also need a negated clock output, which is then fanned-out to the two inputs to the second RS-latch. Another issue lies in the question of how to generate the clock signal for a reversible circuit; however this is outside the scope of this work.

2) *D flip-flop*: An extension of the Master-slave flip-flop is to disallow  $S = R$ . One way to do this is to use a single input  $D$  which is connected directly to the  $S$  input, and then  $\bar{D}$  is connected to the  $R$  input. The behaviour is then as shown in Figure 6 (A). Figure 6 also shows both the traditional logic implementation of the D flip-flop and an equivalent reversible implementation.

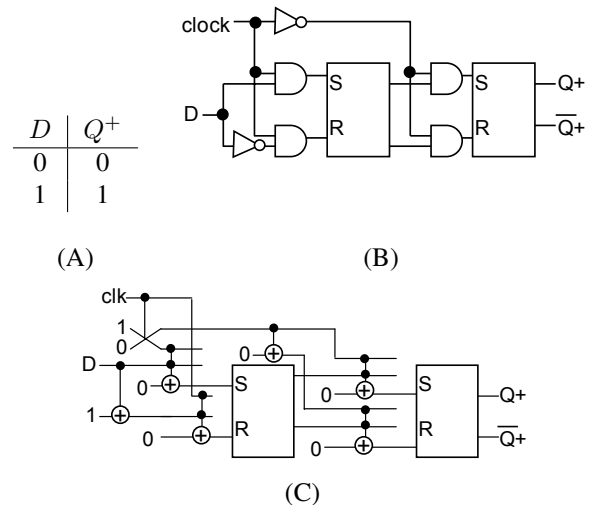


Fig. 6. The (A) behaviour, (B) traditional logic implementation, and (C) equivalent reversible implementation for the D flip-flop.

3) *JK flip-flop*: In many cases it is desirable to be able to retain the value of  $Q$  in the memory element, or even to negate it. The JK flip-flop allows this by making use of additional

feedback. The behaviour is then as defined in Figure 7 (A). Figure 7 also shows a traditional implementation of this flip-flop and an equivalent reversible implementation.

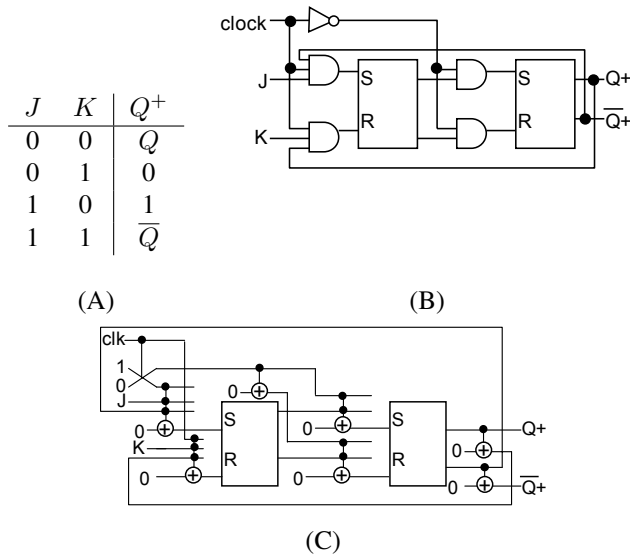


Fig. 7. The (A) behaviour, (B) a traditional logic implementation and (C) a reversible equivalent for the JK flip-flop.

4) *T flip-flop*: Again, a modification to the inputs of the JK flip-flop is possible in order to build a slightly different flip-flop. If we let  $T = J = K$  then the flip-flop behaviour becomes as given in Figure 8 (A). Figure 8 also shows a traditional implementation of this flip-flop and an equivalent reversible implementation.

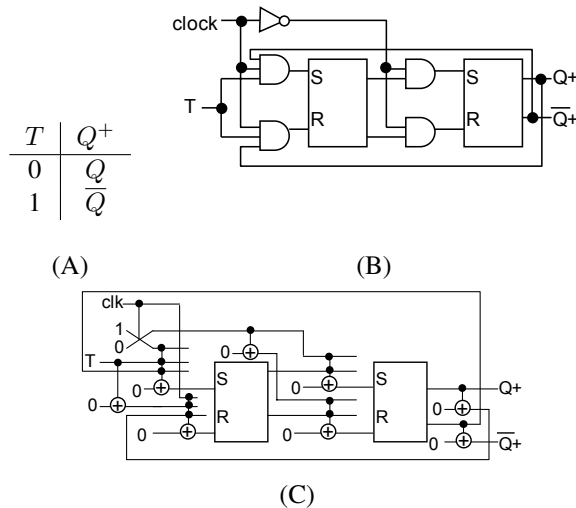


Fig. 8. The (A) behaviour, (B) a traditional logic implementation, and (C) a reversible equivalent for the T flip-flop.

## VI. DISCUSSION

There is some argument as to whether a flip-flop such as the JK flip-flop shown in Figure 7 (C) can be considered *reversible*. Certainly it is constructed entirely of reversible elements; however, by introducing not one but two stages

of feedback is the final structure truly reversible? Further examination is required here in order to satisfy the doubts raised by some researchers.

## VII. CONCLUSION AND FUTURE WORK

This paper presents a discussion of reversible sequential logic. Previous work in the area is examined, with some suggestions as to modifications being presented. A new reversible implementation for a reversible RS-latch is introduced, and four standard types of edge-triggered flip-flops are built exclusively from reversible logic gates.

One particular area of future work, now that reversible memory elements have been presented, is to incorporate these into a logic synthesis process. Existing work on combinational logic synthesis has been presented by researchers such as [9], [10], [11]. It is a logical next step to make use of the most suitable of these in a sequential logic synthesis technique, and future work will address this area.

In addition, the process followed in this paper is fairly pedantic; we simply used known traditional logic implementations for sequential elements and replaced each of the internal elements with reversible equivalents. As Fredkin & Toffoli indicate in [5], this may not result in the most efficient reversible (or conservative) implementation. Thus further investigation into determining optimal implementations for edge-triggered clocked flip-flops is continuing.

## ACKNOWLEDGMENT

The author would like to thank Drs. Claudio Moraga and Michael Miller for enlightening and thought-provoking discussions and emails on this matter.

## REFERENCES

- [1] M. P. Frank, "Introduction to Reversible Computing: Motivation, Progress, and Challenges," in *Proceedings of the 2nd Conference on Computing Frontiers*, 2005, pp. 385–390.
- [2] T. Toffoli, *Automata, Languages and Programming*. Springer Verlag, 1980, ch. title: Reversible Computing, pp. 632–644.
- [3] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Reversible Logic Circuit Synthesis," in *IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2002, pp. 353–360.
- [4] M. P. Frank, "Approaching the Physical Limits of Computing," in *Proceedings of the International Symposium on Multiple-Valued Logic (ISMVL)*, 2005, pp. 168–187.
- [5] E. Fredkin and T. Toffoli, "Conservative Logic," *International Journal of Theoretical Physics*, pp. 219–253, 1982.
- [6] P. Picton, "Multi-Valued Sequential Logic Design using Fredkin Gates," *Multiple-Valued Logic*, pp. 241–251, 1996.
- [7] H. Thapliyal, M. B. Srinivas, and M. Zolinski, "A Beginning in the Reversible Logic Synthesis of Sequential Circuits," in *Proceedings of Military and Aerospace Programmable Logic Devices (MAPLD) International Conference*, 2005, submission 1012.
- [8] T. Sasao, *Switching Theory for Logic Synthesis*. Kluwer Academic Publishers, 1999.
- [9] G. W. Dueck, D. Maslov, and D. M. Miller, "Transformation-based Synthesis of Networks of Toffoli/Fredkin Gates," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, 2003, pp. 211–214, vol.1.
- [10] M. Perkowski, L. Jozwiak, A. Mixhchenko, A. Al-rabadi, A. Copola, A. Buller, X. Song, M. Khan, S. Yanushkevich, V. P. Shmerko, and M. Chrzanoska-Jeske, "A General Decomposition for Reversible Logic," in *Proceedings of the International Workshop on Methods and Representations (RM)*, 2001, pp. 119–138.

- [11] P. Kerntopf, "A New Heuristic Algorithm for Reversible Logic Synthesis," in *Proceedings of the Design Automation Conference (DAC)*, 2004, pp. 834–837.