



ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com



Implementation and Testing of Fredkin Gate based Sequential Circuits

¹E. John Alex and ²M. Vijayaraj¹Research Scholar, Department of Electronics and Communication Engineering, Anna University, Chennai, India.²Associate Professor, Department of Electronics and Communication Engineering, Government College of Engineering, Tirumelveli, India.

ARTICLE INFO

Article history:

Received 16 April 2015

Accepted 12 June 2015

Available online 1 July 2015

Keywords:

Conservative logic, Reversible logic, Fredkin gate

ABSTRACT

In this paper, we propose the design of two vectors testable sequential circuits based on conservative logic gates. The proposed sequential circuits based on conservative logic gates outperform the sequential circuits implemented in classical gates in terms of testability. Any sequential circuit based on conservative logic gates can be tested for classical unidirectional stuck-at faults using only two test vectors. The two test vectors are all 1s, and all 0s. The designs of two vectors testable latches, master-slave flip-flops and double edge triggered (DET) flip-flops are presented. The importance of the proposed work lies in the fact that it provides the design of reversible sequential circuits completely testable for any stuck-at fault by only two test vectors, thereby eliminating the need for any type of scan-path access to internal memory cells. The implementation of different types of shift registers is also explained in the paper.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: E. John Alex and M. Vijayaraj., Implementation and Testing of Fredkin gate based sequential circuit *Aust. J. Basic & Appl. Sci.*, 9(20): 376-386, 2015

INTRODUCTION

CONSERVATIVE logic is a logic family that exhibits the property that there are an equal number of 1s in the outputs as there are in the inputs. Conservative logic can be reversible in nature or may not be reversible in nature. Reversibility is the property of circuits in which there is one to one mapping between the inputs and the output vectors that is for each input vector there is a unique output vector and vice-versa. Conservative logic is called reversible conservative logic when there is a one-to-one mapping between the inputs and the outputs vectors along with the property that there is equal number of 1s in the outputs as in the inputs (Ren *et al.*, 2011). Conservative logic circuits are not reversible, if one-to-one mapping between the inputs and the outputs vectors is not preserved.

Researchers have proved that if the computation is performed in an irreversible manner, each bit of information lost will produce $KT \ln 2$ Joules of heat energy (Murphy *et al.*, 2006). From a thermodynamic point of view, it is also proved that $KT \ln 2$ energy dissipation would not occur, if a computation is carried out in a reversible way. Thus, one of the primary motivations for adopting reversible logic lies in the fact that it can provide a logic design methodology for designing ultralow power circuits beyond $KT \ln 2$ limit for those

emerging nanotechnologies in which the energy dissipated due to information destruction will be a significant factor of the overall heat dissipation.

In this paper, we propose the design of testable sequential circuits based on conservative logic gates. The proposed technique will take care of the fan-out (FO) at the output of the reversible latches and can also disrupt the feedback to make them suitable for testing by only two test vectors, all 0s and all 1s. In other words, circuits will have feedback while executing in the normal mode. However, in order to detect faults in the test mode, our proposed technique will disrupt feedback to make conservative reversible latches testable as combinational circuits. The proposed technique is extended toward the design of two vectors testable master-slave flip-flops and double edge triggered (DET) flip-flops. Thus, our work is significant because we are providing the design of reversible sequential circuits completely testable for any unidirectional stuck-at faults by only two test vectors. The reversible design of the DET flip-flop is proposed for the first time in the literature.

Reversible Logic:

Modern supercomputers are highly scalable devices, of which the performance in principle could grow without limits. However, practical issues such as power dissipation set limits to the size and

therefore to the performance of high-end supercomputers. Energy dissipation in the logic gates of modern supercomputers is still 6 orders of magnitude higher than their thermodynamic threshold:

$$Eth = kBT \ln 2 \tag{1}$$

which depends only on ambient temperature T . It can be easily proved if one realizes that the principal mechanism of the dynamic dissipation could be described as a charging/discharging of the parasitic gate capacitance (Thapliyal *et al.*, 2010). For a reasonable capacitance value C (~ 10 fF) and bias voltage Vb ($\sim 1V$) the recharging energy ($2 C E C * Vb$) is about $5*10^{15}$ J or about $1.7*10^6$ times higher than the thermodynamic threshold that at room (300 K) temperature is $\sim 3*10^{-21}$ J. When taking into account the number of single-bit logic operations associated with one floating point operation, it will help reducing the specific energy dissipation (Tougaw *et al.*, 1994). However, by considering the facts that each floating point

operation requires hundreds of single bit operations and that it is accompanied by a certain number of fixed point operations and by a reading/writing of data and result to/from memory, the energy dissipation per floating point operation can only be reduced to $3*10^{-12}$ J, which is still way larger than the thermodynamic threshold.

A. Conservative Reversible Fredkin Gate:

The Fredkin gate is a popularly used reversible conservative logic gate, first proposed by Fredkin and Toffoli. The Fredkin gate shown in Fig. 1 can be described as a mapping:

$$(A, B, C) \text{ to } (P = A, Q = A_B + AC, R = AB + A_C) \tag{2}$$

where A, B, C are the inputs and P, Q, R are the outputs, respectively. The truth table for the Fredkin gate demonstrates that the Fredkin gate is reversible and conservative in nature, that is, it has unique input and output mapping and also has the same number of 1s in the outputs as in the inputs.

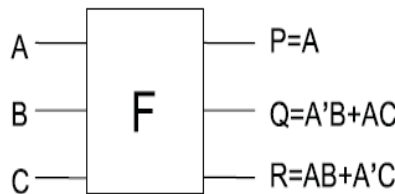


Fig. 1: Fredkin gate.

Existing Reversible Gates:

Several 3x3 reversible gates such as the Fredkin gate, the Toffoli gate and the Peres gate have been reported in the literature. The reversible gate has a cost associated with it called the quantum cost. The quantum cost of a reversible gate is the number of 1x1 and 2x2 reversible gates or quantum logic gates required in its design. The quantum costs of all reversible 1x1 and 2x2 gates are taken as unity. Any reversible gate can be realized using 1x1 NOT gate, and 2x2 reversible gates such as Controlled-V and Controlled -V+ (V is a square-root-of NOT gate and V+ is its hermitian) and the Feynman gate which is also known as the Controlled NOT gate (CNOT). Thus, in simple terms, the quantum cost of a

reversible gate can be calculated by counting the numbers of NOT, Controlled-V, Controlled-V+ and CNOT gates required in its implementation (Tahoori *et al.*, 2004).

3.1 Feynman Gate (CNOT Gate):

The Feynman gate (FG) or the Controlled-NOT gate (CNOT) is a 2-input 2-output reversible gate having the mapping (A, B) to (P=A, Q=A⊕B) where A, B are the inputs and P, Q are the outputs, respectively. Since it is a 2x2 gate, it has a quantum cost of 1. Figure 2 shows the block diagram of the Feynman gate.

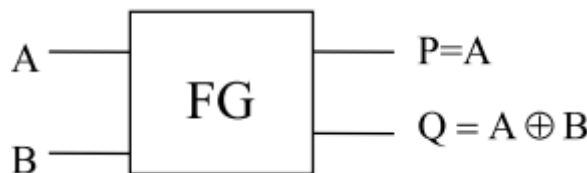


Fig. 2: Feynman Gate (CNOT gate)

3.2 Toffoli Gate:

The Toffoli Gate (TG) is a 3x3 two-through reversible gate as shown in Fig. 3. Two-through means two of its outputs are the same as the inputs

with the mapping (A, B, C) to (P=A, Q=B, R=A ·B⊕C), where A, B, C are inputs and P, Q and R are outputs, respectively. The Toffoli gate is one of the most popular reversible gates and has quantum cost

of 5 (Fredkin and Toffoli, 1982). The quantum cost of Toffoli gate is 5 as it needs 2V gates, 1 V+ gate

and 2 CNOT gates to implement it.

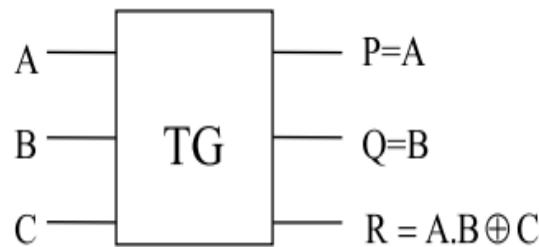


Fig. 3: Toffoli Gate.

3.3 Peres Gate:

The Peres gate is a 3-input 3-output (3x3) reversible gate having the mapping (A, B, C) to (P=A, Q=A⊕B, R=(A·B)⊕C), where A, B, C are the inputs and P, Q, R are the outputs, respectively

(Swaminathan *et al.*,1990) . Figure 4 shows the Peres gate. The quantum cost of Peres gate is 4 since it requires 2 V+ gates, 1 V gate and 1 CNOT gate in its design.

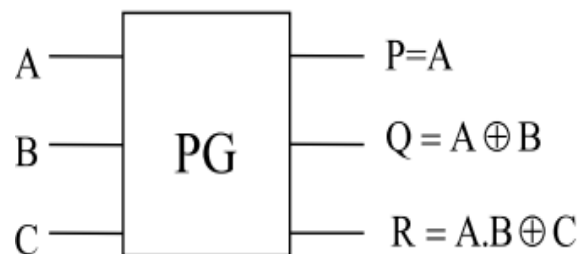


Fig. 4: Peres Gate.

Design of Testable Reversible Latches:

The characteristic equation of the D latch can be written as:

$$Q^+ = D \cdot E + \bar{E} \cdot Q \quad (3)$$

In the proposed work, enable (E) refers to the clock and is used interchangeably in place of clock. When the enable signal (clock) is 1, the value of the input D is reflected at the output that is $Q^+ = D$. While, when $E = 0$ the latch maintains its previous state, that is $Q^+ = Q$. The reversible Fredkin gate has two of its outputs working as 2:1 MUXes, thus the characteristic equation of the D latch can be mapped to the Fredkin gate (F). Fig. 5(a) shows the realization of the reversible D latch using the Fredkin gate. But FO is not allowed in conservative reversible logic. Moreover, the design cannot be tested by two input vectors all 0s and all 1s because of feedback, as the output Q would latch 1 when the inputs are toggled from all 1s to all 0s and could be misinterpreted as stuck-at-1 fault. In this paper, we propose to cascade another Fredkin gate to output Q as shown in Fig. 5(b). The design has two control signals, C1 and C2. The design can work in two modes: Normal mode and Test mode (Mahammad *et al.*,2010).

1) *Normal Mode*: The normal mode is shown in Fig. 5(c) in which we will have $C1C2 = 01$ and

we will have the design working as a D latch without any fan-out problem.

2) *Test Mode (Disrupt the Feedback)*: In test mode, when $C1C2 = 00$ as shown in Fig. 5(d) it will make the design testable with all 0s input vectors as output T1 will become 0 resulting in making it testable with all 0s input vectors. Thus, any stuck-at-1 fault can be detected. When $C1C2 = 11$ as shown in Fig. 5(e), the output T1 will become 1 and the design will become testable with all 1s input vectors for any stuck-at-0 fault. It can see from above that C1 and C2 will disrupt the feedback in test mode, and in normal mode will take care of the fan- out. Thus, our proposed design works as a reversible D latch and can be tested with only two test vectors, all 0s and all 1s, for any stuck-at fault by utilizing the inherent property of conservative reversible logic.

5. Design of Testable Negative Enable Reversible D Latch:

A negative enable reversible D latch will pass the input D to the output Q when $E = 0$; otherwise maintains the same state. The characteristic equation of the negative enable D latch is: $Q^+ = D \cdot \bar{E} + E \cdot Q$. This characteristic equation of the negative enable reversible D latch can be mapped on the second output of the Fredkin gate as shown in Fig. 6.

The second Fredkin gate in the design take cares of the FO. The second Fredkin gate in the design also helps in making the design testable by two test vectors, all 0s and all 1s, by breaking the feedback based on control signals C1 and C2 as illustrated above for positive enable reversible D latch. The negative enable D latch is helpful in the design of

testable reversible master-slave flip-flops. This is because as it can work as a slave latch in the testable reversible master-slave flip-flops in which no clock inversion is required. The details of which are discussed in the section describing reversible master-slave flip-flops.

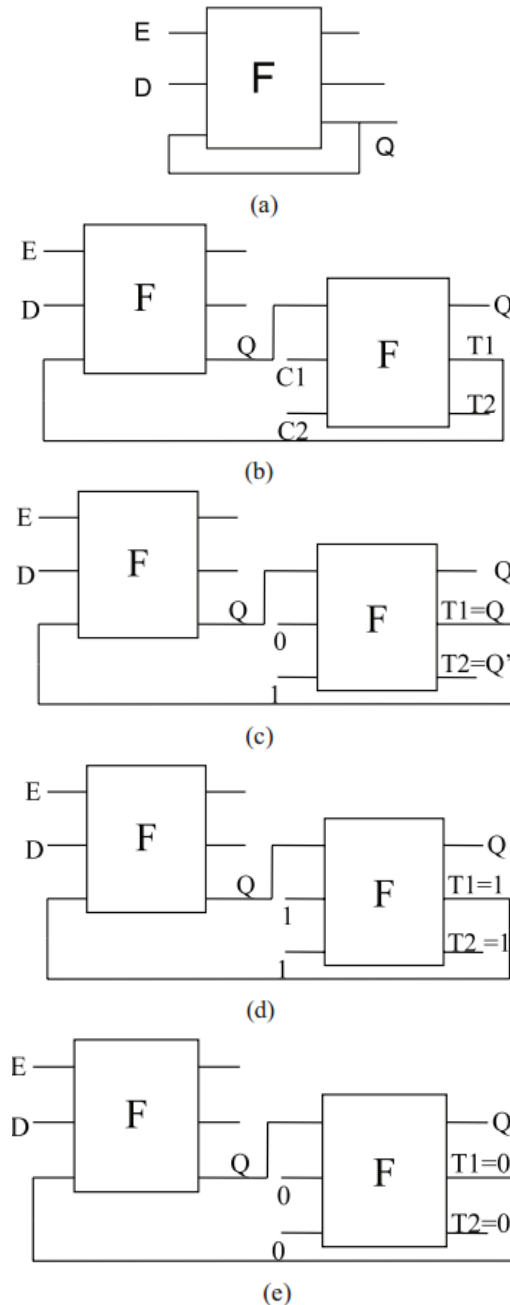


Fig. 5: Design of testable reversible D latch using conservative Fredkin gate. (a) Fredkin gate based D latch. (b) Fredkin gate based D latch with control signals C1 and C2. (c) Fredkin gate based D Latch in normal mode: $C1 = 0$ and $C2 = 1$. (d) Fredkin gate based D latch in test mode for stuck-at-0 fault: $C1 = 1$ and $C2 = 1$. (e) Fredkin gate based D latch in test mode for stuck-at-1 fault: $C1 = 0$ and $C2 = 0$.

6.Design of Testable Master-Slave Flip-Flops:

In the existing literature, the master-slave strategy of using one latch as a master and the other

latch as a slave is used to design the reversible flip-flops. In this paper, we have proposed the design of testable flip-flops using the master-slave strategy that

can be tested for any stuck-at faults using only two test vectors, all 0s and all 1s. Fig. 7 shows the design of the master-slave D flip-flop in which we have used positive enable Fredkin gate-based testable D latch shown in Fig. 5(b) as the master latch, while the slave latch is designed from the negative enable Fredkin gate-based testable D latch shown earlier in Fig. 4. The testable reversible D flip-flops has four control signals mC1, mC2, sC 1, and sC 2. mC1 and

mC2 control the modes for the master latch, while sC 1 and sC 2 control the modes for the slave latch. In the normal mode, when the design is working as a master-slave flip-flop the values of the controls signals will be mC1 = 0 and mC2 = 1, sC 1 = 0 and sC 2 = 1 (as similar to values of the control signals C1 and C2 earlier described for the testable D latches).

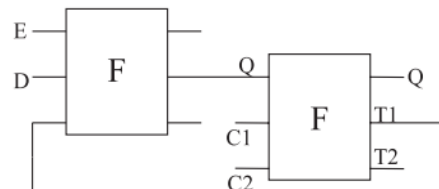


Fig. 6: Fredkin gate-based negative enable testable D latch.

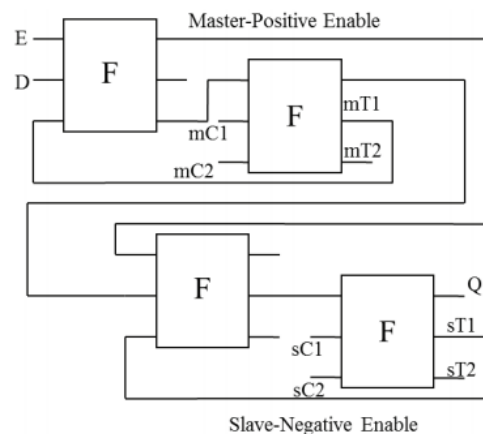


Fig. 7: Fredkin gate-based testable reversible master-slave D flip-flop.

In the test mode:

1) To make the design testable with all 0s input vectors for any stuck-at-1 fault, the values of the controls signals will be mC1 = 0 and mC2 = 0, sC 1 = 0 and sC 2 = 0. This will make the outputs mT 1 and sT 1 as 0, which results in breaking the feedback and the design becomes testable with all 0s input vectors for any stuck-at-1 fault.

2) To make the design testable with all 1s input vectors for any stuck-at-0 fault, the values of the control signals will be mC1 = 1 mC2 = 1, sC 1 = 1, and sC 2 = 1. This will result in outputs mT 1 and sT 1 having a value of 1, breaking the feedback and resulting in the design testable with all 1s input vectors for any stuck-at-0 fault.

The other type of master-slave flip-flops, such as the testable master-slave T flip-flop, testable master-slave JK flip-flop, and testable master-slave SR flip-flop can be designed similarly in which master is designed using the positive enable corresponding latch, while the slave is designed using the negative enable Fredkin gate-based D latch. For example, in the design of master-slave T flip-flop, the master is designed using the positive enable T latch, while the

slave is designed with the negative enable D latch. An example of comparative study is shown in Table I that shows the comparison of proposed reversible sequential building blocks with the existing reversible nontestable sequential building blocks and online testable sequential building blocks in terms of number of gate count, number of constant inputs, and garbage outputs. The number of gate count, number of constant inputs, and garbage output of an online testable sequential building are computed without consideration of IRC blocks used in the designs.

7.Design of Testable Reversible DET Flip-Flops:

The DET flip-flop is a computing circuit that samples and stores the input data at both the edges that is at both the rising and the falling edge of the clock. The master-slave strategy is the most popular way of designing the flip-flop (Pedram *et al.*). In the proposed work, E refers to the clock and is used interchangeably in place of clock. In the negative edge triggered master-slave flip-flop when E = 1 (the clock is high), the master latch passes the input data while the slave latch maintains the previous state. When E = 0 (the clock is low), the master latch is in

the storage state while the slave latch passes the output of the master latch to its output. Thus, the flip-flop does not sample the data at both the clock levels and waits for the next rising edge of the clock to latch the data at the master latch.

In order to overcome the above problem, researchers have introduced the concept of DET flip-flops, which sample the data at both the edges. Thus, DET flip-flops can receive and sample two data values in a clock period thus frequency of the clock can be reduced to half of the master-slave flip flop while maintaining the same data rate. The half frequency operations make the DET flip flops very much beneficial for low power computing as frequency is proportional to power consumption in a circuit. The DET flip-flop is designed by connecting the two latches, viz., the positive enable and the negative enable in parallel rather than in series. The 2:1 MUX at the output transfer the output from one of these latches which is in the storage state (is holding its previous state). The conventional design of the DET flip-flop can be found in. The equivalent testable reversible design of the DET flip flop is proposed in this paper and is shown in Fig. 8(a).

In the proposed design of testable reversible DET flip-flop, the positive enable testable reversible D latch and the negative enable testable reversible D latch are arranged in parallel. The Fredkin gates labeled as 1 and 2 forms the positive enable testable D latch, while the Fredkin gates labeled as 3 and 4 forms the negative enable testable D latch. In reversible logic FO is not allowed so the Fredkin gate labeled as 6 is used to copy the input signal D. The Fredkin gate labeled as 5 works as the 2:1 MUX and transfer the output from one of these testable latches

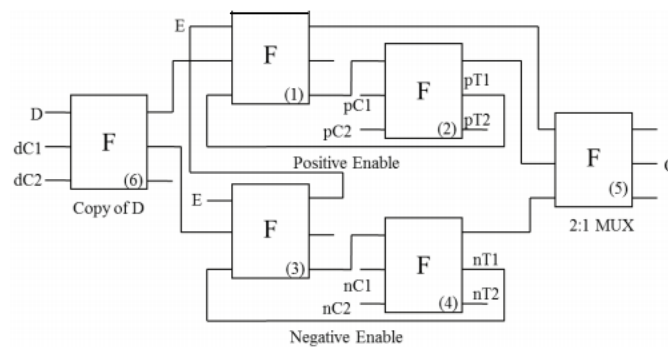
(negative enable D latch or the positive enable D latch) that is in the storage state (is holding its previous state) to the output Q. In the proposed design of testable reversible DET flip-flop, pC1 and pC2 are the controls signals of the testable positive enable D latch, while nC1 and nC2 are the control signals of the testable negative enable D latch. Depending on the values of the pC1, pC2, nC1, and nC2, the testable DET flip-flops work either in normal mode or in the testing mode.

1) Normal Mode: The normal mode of the DET flip-flop is illustrated in Fig. 8(b) in which the pC 1 = 0, pC 2 = 1, nC 1 = 0, and nC 2 = 1. The pC 1 = 0, pC 2 = 1 help in copying the output of the positive enable D latch thus avoiding the FO while the nC 1 = 0 and nC 2 = 1 help in copying the output of the negative enable D latch thus avoiding the FO.

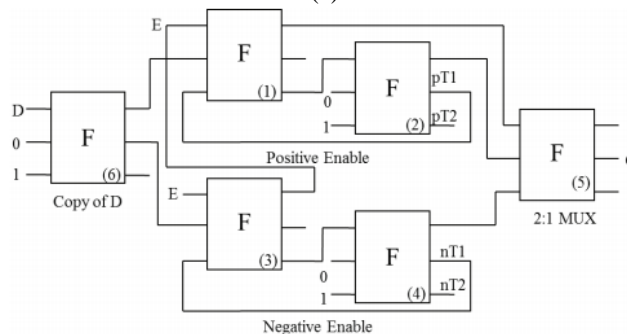
2) Test Mode: There will be two test modes.

a) All 1s Test Vectors: This mode is illustrated in Fig. 8(d) in which control signals will have value as pC 1 = 1, pC 2 = 1, nC 1 = 1, and nC 2 = 1. The pC 1 = 1 and pC 2 = 1 help in breaking the feedback of the positive enable D latch, while the nC 1 = 1 and nC 2 = 1 help in breaking the feedback of the negative enable D latch. This makes the design testable by all 1s test vector for any stuck-at-0 fault.

b) All 0s Test Vectors: This mode is illustrated in Fig. 8(c) in which the control signals will have value as pC 1 = 0, pC 2 = 0, nC 1 = 0, and nC 2 = 0. The pC 1 = 0 and pC 2 = 0 help in breaking the feedback of the positive enable D latch, while the nC 1 = 0 and nC 2 = 0 help in breaking the feedback of the negative enable D latch. This makes the design testable by all 0s test vector for any stuck-at-1 fault.



(a)



(b)

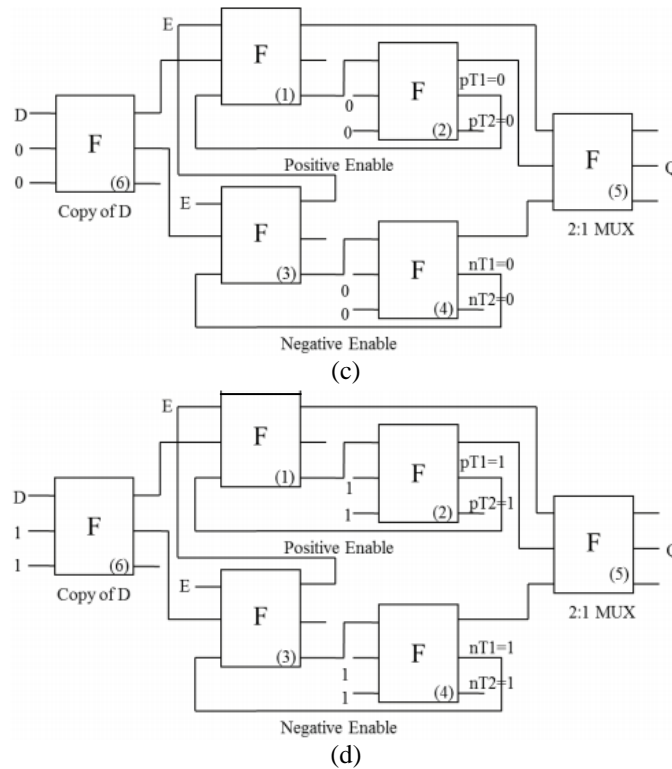


Fig. 8: Fredkin gate-based DET flip-flop. (a) Fredkin gate based DET flipflop. (b) Normal mode. (c) Test mode for stuck-at-1 fault. (d) Test mode for stuck-at-0 fault.

8.New Proposed Reversible Gate and its Applications:

8.1Proposed Reversible AS Gate:

The logic diagram of the proposed reversible gate AS is shown in Figure 9. The proposed

reversible gate AS is a 4x4 reversible gate with inputs(A,B,C,D) and with outputs A', AB + A'C, D (AB + A'C) and B C.



Fig. 9: Proposed 4x4 AS Reversible Gate.

In this gate the input vector is given by IV=(A,B,C,D) and the corresponding output vector is OV=(P,Q,R,S). The quantum cost of the proposed reversible gate AS is 6. The quantum cost represents the number of 1x1 and 2x2 primitive gates used in the realization of the proposed reversible gate AS.

8.2 Realization of the Classical Operations using the Proposed Reversible Gate AS:

The proposed reversible gate AS can implement OR, AND, XOR, NOT and COPY operation. Also since AND, OR and NOT operation can be implemented justifies the aforesaid because any Boolean function can be materialized in product – of

– sum or sum – of – products form. Also the COPY operation is an important operation which can be realized using the proposed reversible gate AS.

8.3 Proposed Design of Reversible DLatch Using AS Gate:

To minimise the transistor count, we have implemented a reversible D-Latch using the proposed reversible gate AS. The symbolic representation of the proposed reversible D-Latch with the un-complemented output is shown in figure 13.

From reversible gate AS, when D is 0, A= Clk, B = D(Data Input), and C = Q(Previous Output) P

$= Clk'$, $R = Q$, $S = D \text{ XOR } Q$ and $Q = ClkD + Clk'Q$, which represents the Boolean Expression of D Latch. The output $P = Clk'$ can be used to realize the master-slave D-flip-flop and $S = D \text{ XOR } Q$ represent

the garbage output. Thus, the proposed Reversible D-Latch requires 1 reversible gate. The circuit accepts 1 constant input and produces one garbage output which is an optimized circuit.

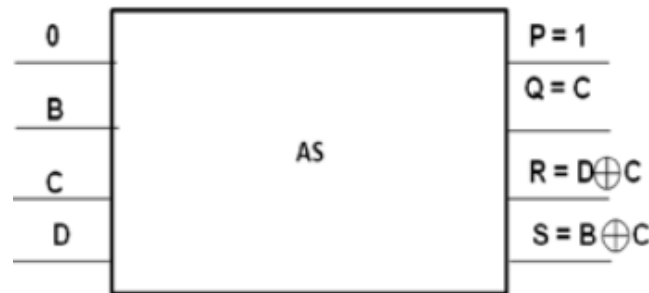


Fig. 10: Reversible Gate AS implementing reversible XOR and COPY operation.

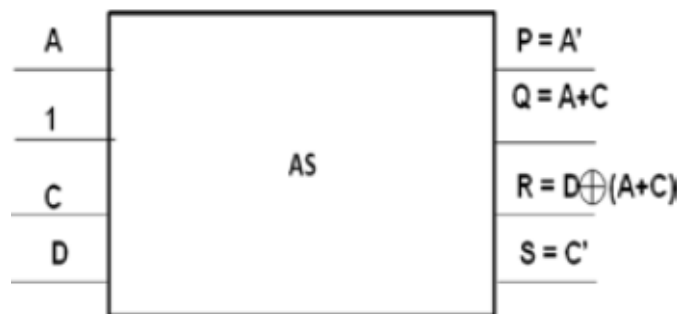


Fig. 11: Reversible Gate AS implementing reversible OR, NOT and XOR operation

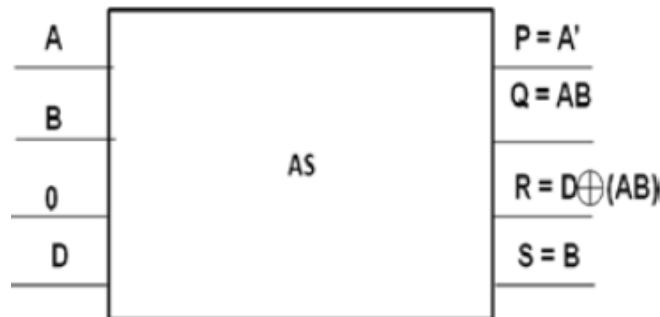


Fig. 12: Reversible Gate AS implementing reversible AND, NOT and COPY operation

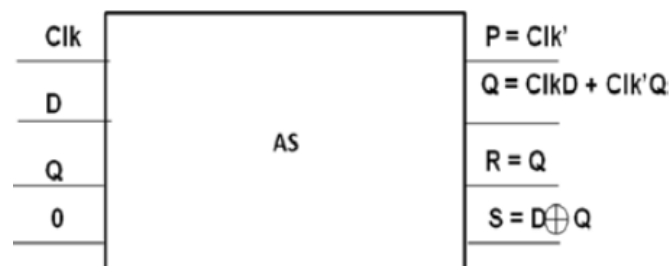


Fig. 13: Symbolic Representation of Proposed Reversible D-Latch with the output Q Using Reversible Gate AS.

9. Design of Shift Registers:

A shift register is a cascade arrangement of flip-flops, sharing the same clock, in which the output of

each flip-flop is connected to the "data" input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the "bit array" stored in it. It

shifts in the data present at its input and shifts out the last bit in the array, at each transition of the clock input. More generally, shift registers can have both parallel and serial inputs and outputs. These are often configured as serial-in, parallel-out (SIPO) or as parallel-in, serial-out (PISO). There are also types that have both serial and parallel input and types with serial and parallel output.

A. Design of Serial In Serial Out Shift Register (SISO):

The serial in /serial out shift register accepts data serially – that is, one bit at a time on a single line. It produces the stored information on its output also in

serial form. The input data is then applied sequentially to the D input of the first flip-flop on the left .During each clock pulse, one bit is transmitted from left to right. A basic four-bit shift register can be constructed using four D flip-flops. The operation of the circuit is as follows. The input data is then applied sequentially to the D input of the first flip-flop on the left. During each clock pulse, one bit is transmitted from left to right. A 4-bit reversible Serial in Serial out Shift Register is realized using the proposed D-Flip-Flop. Thus, 4 reversible D-Flip-Flops are connected in cascade in series to implement the 4-bit reversible shift register.

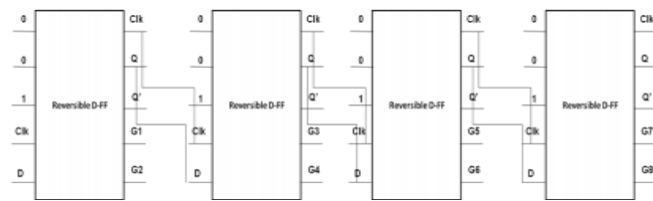


Fig. 14: Serial In Serial Out Shift Register using reversible D flip-flop.

B. Design of Serial In Parallel Out Shift Register:

In this type of shift register, data bits are entered serially that is, one bit at a time on a single line. It produces the stored information on its output in parallel form. The input data is then applied sequentially to the D input of the first flip-flop on the

left .During each clock pulse, one bit is transmitted from left to right. Once the data are stored, each bit appears on its respective output line, and all bits are available simultaneously. A construction of a four-bit serial in - parallel out register is shown in figure 15.



Fig. 15: Serial In Parallel Out Shift Register using reversible D flip-flop.

C. Design of Parallel In Parallel Out Shift Register:

For parallel in - parallel out shift registers, all data bits appear on the parallel outputs immediately

following the simultaneous entry of the data bits. Figure 16 shows the 4-bit Reversible Parallel In n Parallel Out Shift Register.



Fig. 16: Parallel In Parallel Out Shift Register using reversible D flip-flop.

D. Design of Parallel In Serial Out Shift Register:

Figure 17 shows the realization of the reversible 4 – bit Parallel In Serial Out Shift Register. To write the new data to the register, W/S line must be held high. To shift the data, W/S line should be low. Fredkin gate is used as the multiplexer to select whether to shift the data or to load a new data.

The entire unit was functionally verified. At testbench is used to generate the stimulus and applies it to the implemented reversible d-flip-flop, Serial in Serial out Shift Register, Serial in Parallel Out Shift Register, Parallel in Serial Out Shift Register and Parallel in Parallel out Shift Register. The design was simulated using Modelsim and synthesized using Xilinx Virtex5.

Simulation Results:

The simulation results for the proposed reversible sequential circuit designs have been obtained using Xilinx.

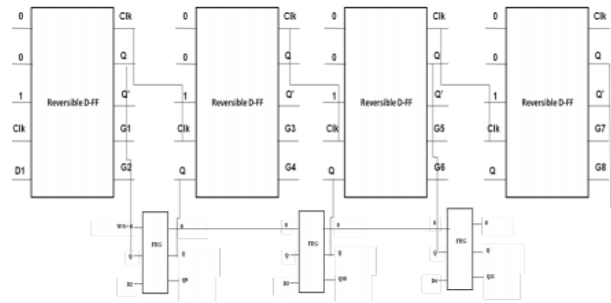


Fig. 17: Parallel In Serial Out Shift Register using reversible D flip-flop.

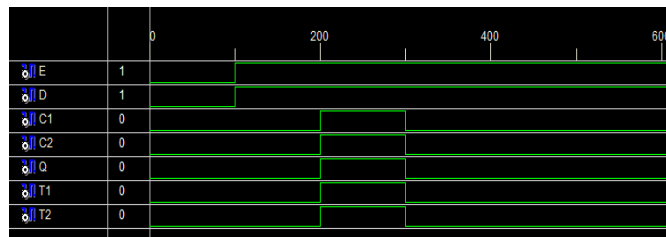


Fig. 18: Simulation result for D latch.

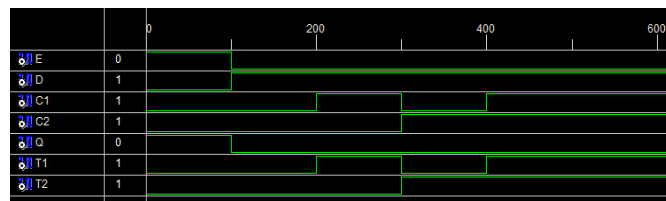


Fig. 19: Simulation result for Negative Enabled D latch.

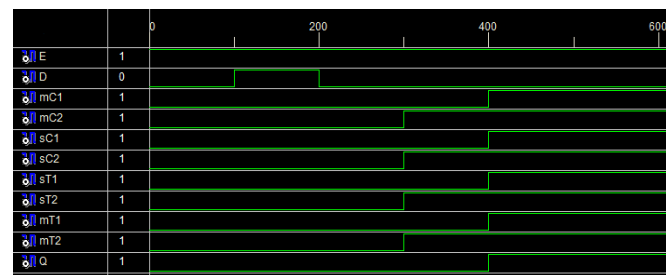


Fig. 20: Simulation result for Master-Slave Flip-flop.

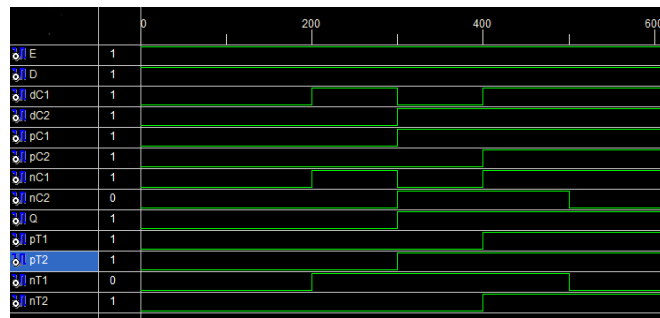


Fig. 21: Simulation result for DET (Double Edge Triggered) Flip-flop.

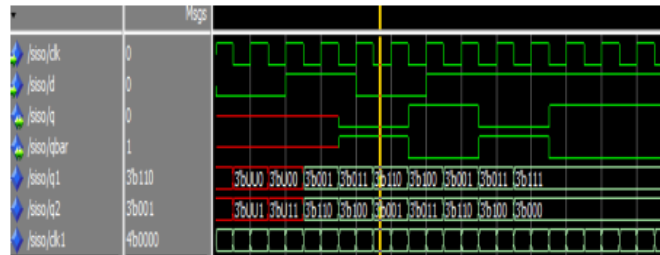


Fig. 22: Simulation Result of the Proposed Reversible 4-Bit Serial In Serial Out Shift Register.

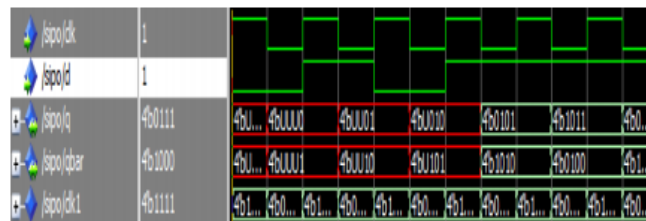


Fig. 23: Simulation Result of the Proposed Reversible 4-Bit Serial In Parallel Out Shift Register.

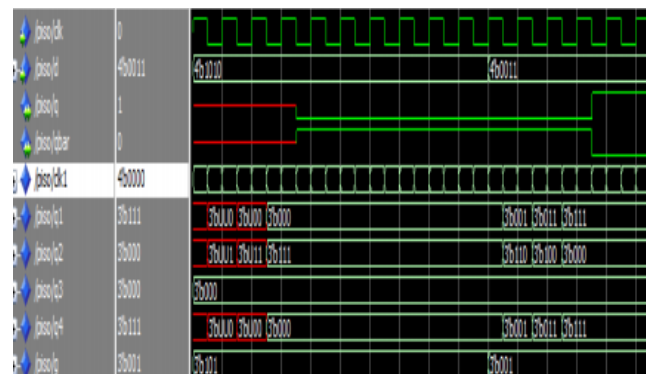


Fig. 24: Simulation Result of the Proposed Reversible 4-Bit Parallel In Serial Out Shift Register.

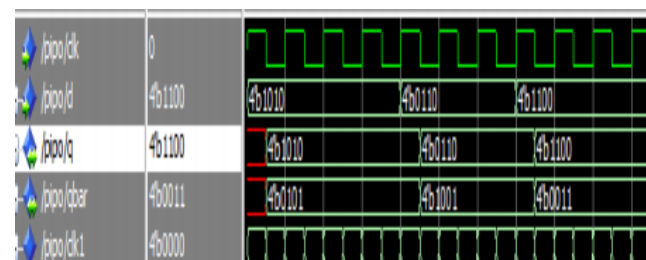


Fig. 25: Simulation Result of the Proposed Reversible 4-Bit Parallel In Parallel Out Shift Register.

REFERENCES

- Fredkin, E. and T. Toffoli, 1982. "Conservative logic," *Int. J. Theor. Phys.*, 21(3-4): 219-253.
- Mahammad, S. and K. Veezhinathan, 2010. "Constructing online testable circuits using reversible logic," *IEEE Trans. Instrum. Meas.*, 59(1): 101-109.
- Murphy, S.F., M. Ottavi, M. Frank and E. DeBenedictis, 2006. "On the design of reversible QDCA systems," Sandia National Laboratories, Albuquerque, NM, Tech. Rep. SAND2006-5990.
- Pedram, M., Q. Wu and X. Wu, 1998. "A new design for double edge triggered flip-flops," in *Proc. Asia South Pacific Design Autom. Conf.*, pp: 417-421.
- Ren, J. and V.K. Semenov, 2011. "Progress with physically and logically reversible superconducting digital circuits," *IEEE Trans. Appl. Super-conduct.*, 21(3): 780-786.
- Swaminathan, G., J. Aylor and B. Johnson, 1990. "Concurrent testing of VLSI circuits using conservative logic," in *Proc. Int. Conf. Comput. Design*, Cambridge, MA, Sep., pp: 60-65.
- Tahoori, M.B., J. Huang, M. Momenzadeh, and F. Lombardi, 2004. "Testing of quantum cellular automata," *IEEE Trans. Nanotechnol.*, 3(4): 432-442.
- Thapliyal, H. and N. Ranganathan, 2010. "Reversible logic-based concurrently testable latches for molecular QCA," *IEEE Trans. Nanotechnol.*, 9(1): 62-69.
- Tougaw, P. and C. Lent, 1994. "Logical devices implemented using quantum cellular automata," *J. Appl. Phys.*, 75(3): 1818-1825.
- Tougaw, P. and C. Lent, 1996. "Dynamic behavior of quantum cellular automata," *J. Appl. Phys.*, 80(8): 4722-4736.