HIGH SPEED FAULT TOLERANT FLIP-FLOP USING REVERSIBLE GATES

[1] B. Srinivasa Reddy, [2] R. Narayana, [3] Y. Kamala, [4] G. Kesava Sai Krishna Reddy, [5] D. Gowri Sankar Rao

[1,2,3,4] U G Scholars, [5]Assistant Professor

[1]Electronics and Communication Engineering

[1]Godavari Institute of Engineering & Technology, Rajahmundry, Andhra Pradesh

Abstract: In Fault tolerant reversible logic gate circuit, the parity of the input vector must equal the parity of the output vector. It renders to find the fault in the circuit. Thus, the parity preserving reversible logic will be beneficial to the development of fault tolerant systems in nanotechnology. Fault tolerant reversible logic and reversible logic gates are widely known to be compatible with revolutionary computing paradigms such as optical and quantum computing. This thesis presents an efficient realization of D flip-flop and master slave flip-flop using RR- Fault tolerant parity preserving reversible gates. The minimum power, delay, power delay product and hardware complexity to synthesize a fault tolerant reversible D flip-flop and master slave flip-flop circuit has also been given. Finally, this thesis presents a novel fault tolerant reversible D flip-flop and master slave flip-flop circuit and demonstrates its superiority with the existing.

Index Terms **– Reversible gates, RR Gate, MSH Gate, D-Flip-Flop, Master-Slave Flip-Flop.**

I. INTRODUCTION

Reversible logic is an emerging research area. Interest in reversible logic is sparked by its applications in several technologies, such as quantum, CMOS, optical and nanotechnology. Reversible implementations are also found in thermodynamics and adiabatic CMOS. Power dissipation in modern technologies is an important issue, and overheating is a serious concern for both manufacturers (impossibility of introducing new, smaller scale technologies, limited temperature range for operating the product) and customer (power supply, which is especially important for mobile systems). One of the main benefits that reversible logic brings is theoretically zero power dissipation in the sense that, independently of underlying technology, irreversibility means heat generation.

A reversible circuit maps each output vector into a unique input vector, and vice versa. CMOS reversible / adiabatic circuits are currently the most important approaches to power optimization. This paper introduces an approach to synthesize generalized multi-rail reversible cascades for single output Boolean functions. Minimizing the “garbage bits” is the main challenge of reversible logic synthesis.

The key point of reversible computing is that the electric charge on the storage cell consisting of transistors is not permitted to flow away when the transistor is switched. Then it can be reused through reversible computing, which can decrease the energy consumption. When there is no loss of information bits, then the system is reversible. In the VLSI circuits, it means that the circuits consisting of AND gate and OR gate, the bit information presented by charge can be saved when it is not used, which leads to the reversibility of the system. Take the operation of two-bit XOR for example, it has one bit output, which lost a freedom degree. According to the thermodynamics theory, the irreversible operation will inevitably lead to energy consumption. According to Bennet’s theory, every step of irreversible operation can be transformed to reversible operation in the traditional computer, which has the same ability of capability and efficiency for the computer.

There are two major, closely related, types of reversibility that are of particular interest for this purpose: physical reversibility and logical reversibility. A process is said to be physically reversible if it results in no increase in physical entropy; it is isentropic. These circuits are also referred to as charge recovery logic or adiabatic computing. Although in practice n number non stationary physical process can be exactly physically reversible or isentropic, there is no known limit to the closeness with which we can approach perfect reversibility, in systems that are sufficiently well-isolated from interactions with unknown external environments, when the laws of physics describing the system's evolution are precisely known.

If one creates a reversible function by listing the 2n possible rows for any n- input n-output function and creates the outputs by simply permuting the input rows, one might call the resulting logic function a gate". However,there are some characteristics that are desirable before we can label an arbitrary function as a gate. Firstly, it is desirable for the gate to be universal, if possible. This means that it should be possible to create any possible function by using a cascade of the gate in question. For instance, the NAND gate is universal in traditional logic, in that any logic function can be built simply by cascading NAND gates together. The Toffoli gate is also universal. Secondly, it is desirable to have a simple, or even elemental implementation available for each gate. This is so that a conversion from a gate layout to a physical implementation can be performed in a straight- forward way by mapping each gate to its basic underlying implementation. For instance, in traditional logic a NOT gate requires 2 transistors, while a two-input NAND gate requires 4 transistors. Thus, in traditional logic the elementary unit could be considered to be the transistor. Similarly, in quantum and reversible computing, the elementary unit is the quantum gate.

There are a variety of suggestions for how such a gate could be implemented; the reader is directed to works such as for recent developments and for a comprehensive discussion of quantum computing. In general, an elementary quantum gate acts on a single qubit and is described as some mathematical transform that represents the change of state of that qubit to some other state. For instance, the Hadamard gate represents a rotation of about the x and z axes. Many of the researchers in the reversible logic literature refer to Barenco et al. for the numbers of elementary quantum gates required for implementing gates such as the Toffoli gate. More recently Miller, Wille and Sasanian have proposed updates to these quantum costs. Reversible circuits are generally compared based on their quantum cost and number of qubits.

II. LITERATURE SURVEY

Vasudevan D. P., P. K. Lala and J. P. Parkerson, 2004. A novel approach for online testable reversible logic circuit design, Proceedings of the 13th Asian Test Symposium (ATS 2004), pp. 325330. The Vasudevan designed Conventional digital circuits dissipate a significant amount of energy because bits of information are erased during the logic operations. Thus, if logic gates are designed such that the information bits are not destroyed, the power consumption can be reduced dramatically. The information bits are not lost in case of a reversible computation. This has led to the development of reversible gates. This paper proposes three new reversible logic gates; two of the proposed gates can be employed to design online testable reversible logic circuits. Furthermore, they can be used to implement any Boolean logic function. The application of the reversible gates in implementing several benchmark functions has been presented.

The Author Majid Haghparast designedReversible logic is an emerging area of research, having applications in nanotechnology, low power CMOS design, quantum computing, and DNA computing. In this paper, two different parity-preserving reversible error coding and detection circuits are studied. First we propose two new reversible Hamming code generator circuits. One of them is parity-preserve. We also propose a new parity-preserving reversible Hamming code error detector circuit. The proposed parity-preserving reversible Hamming code generator (PPHCG) and error detector circuits provide single error correction–double error detection (SEC–DED). The designs are better than the existing counterparts in terms of quantum cost (QC), number of constant inputs, and number of garbage outputs. Then we propose parity-preserving reversible cyclic code encoder/decoder circuits for the first time. A parity-preserving reversible D flip-flop is also proposed. Equivalent quantum representation of two parity-preserving 4 ∗ 4 reversible gates, IG, and PPHCG, are also proposed. We show for the first time that IG has a QC of only 7 and PPHCG has a QC of only 6.

The Author Mubin Ul Haqueproposed logic elements are master slave Flip Flop, D-Latch and multiplexer. A new 4×4 and a new 6×6 fault tolerant reversible gates are proposed for designing efficient reversible fault tolerant D-latch, master slave Flip Flop and multiplexer, respectively using MSB and MSH gates. The design of the proposed logic elements achieves the improvement of 41.67% in terms of number of gates compared to the best-known existing approach.

Quantum computers require quantum arithmetic. We provide an explicit construction of quantum networks effecting basic arithmetic operations: from addition to modular exponentiation. Quantum modular exponentiation seems to be the most difficult (time and space consuming) part of Shor's quantum factorising algorithm. We show that the auxiliary memory required to perform this operation in a reversible way grows linearly with the size of the number to be factorized.

III. DESIGNING OF FLIP-FLOPS USING MSH GATE

In this section, the basic definitions of reversible fault tolerant gate, garbage output, unit delay, and quantum cost are presented. Besides, the recent related existing works are briefly described.

A reversible gate is an n input In, n output on (denoted by n x n) circuit that produces a unique output pattern for each possible input pattern as $In \leftrightarrow On$, where unused output is known as garbage output.

Fault tolerant gates are reversible gates which maintain parity between input and output vectors as $I1 \oplus I2 \oplus ... \oplus In = O1 \oplus O2 \oplus ... \oplus on$. In this thesis, the fault tolerant gates are used to preserve the parity of the proposed circuits, where the fault tolerant gates are used to detect the faults of the circuits when the fault occurs.

The quantum cost can be derived by substituting the reversible gates of a circuit by a cascade of elementary quantum gates. Quantum Cost of a circuit is the minimum number of 2×2 unitary gates to represent the circuit keeping the output unchanged. The quantum cost of a 1×1 gate is zero and that of any 2×2 gate is one. In other words, number of 2×2 Ex-OR, controlled-V (square root of NOT i.e. SRN) or controlled-*V*+ (Hermitian of SRN) represent the quantum cost of a circuit.

For example, Fig. 3.1 shows the block diagram of two fault tolerant reversible gates named Fredkin (FRG) and Feynman double gate (F2G) along with their quantum realization.

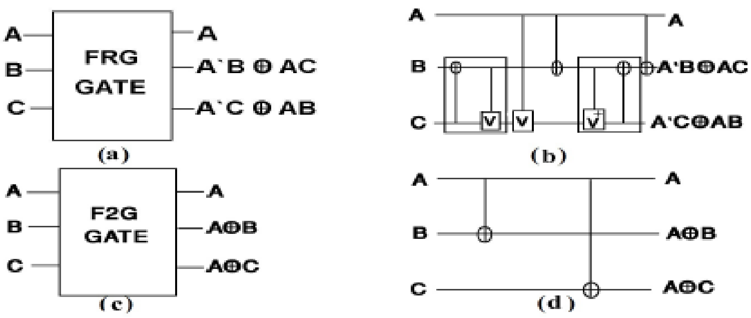

Fig. 1: Fredkin Gate (a) Block Diagram(b)Quantum Realization (Cost 5) Double Gate (c) Block Diagram (d) Quantum Realization (Cost 2)

Delay represents the critical delay of the circuit which considers the following two assumptions. Firstly, each gate performs computation in unit time which means that every gate will take same amount of time for internal logic operations. Secondly, all the inputs are known to the circuit before the computation begins.

Power can be obtained from current using Micro wind DSCH [7] and voltage across each transistor. We have considered the static power consumption of the circuit and power across each transistor is calculated using the following formula:

P ($Power$) = (V_i) (voltage) × (I) (current).

Where *Power* represents power consumed by, *I* number of transistors in a circuit and *Vi* symbolizes voltage required at each transistor. For example, let us consider a half adder circuit which is composed of an *AND* gate and *Ex-OR* gate. Using Micro Wind DSCH [7], critical path delay of each of the *Ex-OR* and *AND* gate is 0.160 *ns*. So, delay of a half adder circuit is 0.160 *ns* and unit delay of the circuit is one. On the other hand, an *AND* gate and *Ex-OR* gate require 6 and 8 transistors, respectively.

3.1 Reversible Fault Tolerant Logic Elements:

3.1.1 Reversible Fault Tolerant Gates:

We Existing two new reversible fault tolerant gates along with their block diagram, quantum realization and truth tables.

1) MSH (Mubin-Sworna-Hasan) Gate:

Figure present the proposed MSH gate (input vector is [A, B, C, D] and output vector is [P, Q, R, S]) with its quantum realization.

2) MSB (Mubin-Sworna-Babu) Gate:

Block diagram and quantum realization of the proposed MSB gate (input vector is [A, B, C, D, E, F] and output vector is [P, Q, R, S, T, U]) are represented in Fig. 2(c) and Fig. 2(d), respectively.

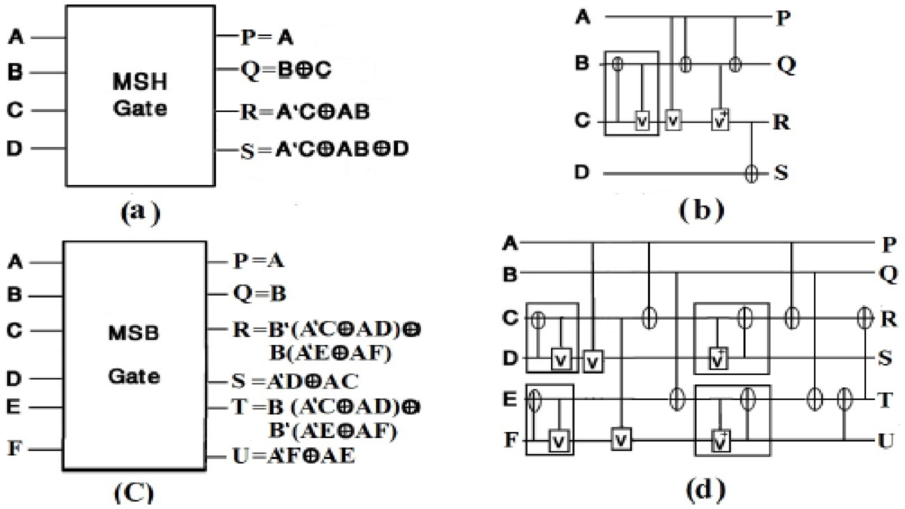

Fig. 2: MSH Gate :(a) Block Diagram (b) Quantum Realization (Cost 6) MSB Gate :(c) Block Diagram (d) Quantum Realization (Cost 12)

3.1.2 Proof of Fault Tolerance Property of Existing Gates:

Table I demonstrates the unique one to one correspondence and parity preservation between input (A, B, C and D) and output (P, Q, R and S) of the proposed MSH gate. For any input combination, corresponding output combination shows the same parity (that is A⊕B⊕C⊕D = P⊕Q⊕R⊕S) as well as reversibility of the proposed gate.

Similarly, Table II clarifies the fault tolerance property and the unique mapping between input (A, B, C, D, E and F) and output (P, Q, R, S, T and U) of the proposed MSB gate.

Table 1: Truth Table of MSH Gate

INPUT				OUTPUT				Parity 1=Odd 0=Even
A	B	C	D	P	Q	R	S	
0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	1	1	1	1
0	0	1	1	0	1	1	0	0
0	1	0	0	0	1	0	0	1
0	1	0	1	0	1	0	1	0
0	1	1	0	0	0	1	1	0
0	1	1	1	0	0	1	0	1
1	0	0	0	1	0	0	0	1
1	0	0	1	1	0	0	1	0
1	0	1	0	1	1	0	0	0
1	0	1	1	1	1	0	1	1
1	1	0	0	1	1	1	1	0
1	1	0	1	1	1	1	0	1
1	1	1	0	1	0	1	1	1
1	1	1	1	1	0	1	0	0

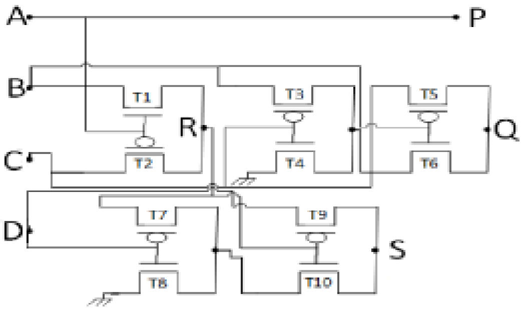

Fig. 3: Transistor Realization of the Existing MSH Gate

Table 2: Truth Table of MSB Gate

INPUT						OUTPUT						Parity 1=Odd 0=Even
A	B	C	D	E	F	P	Q	R	S	T	U	
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	1	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	0	1	0	1	0
0	0	0	1	1	0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	0	0	1	1	1	1
0	0	1	0	0	0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	0	1	0	0	1	0
0	0	1	0	1	0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	0	1	0	1	1	1
0	0	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	0	1	0	0	1	1	0	1	1
0	0	1	1	1	0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	0	1	1	1	1	0
0	1	0	0	0	0	0	1	0	0	0	0	1
0	1	0	0	0	1	0	1	0	0	0	1	0
0	1	0	0	1	0	0	1	1	0	0	0	0
0	1	0	0	1	1	0	1	1	0	0	1	1
0	1	0	1	0	0	0	1	0	1	0	0	0
0	1	0	1	0	1	0	1	0	1	0	1	1
0	1	0	1	1	0	0	1	1	1	0	0	1
0	1	0	1	1	1	0	1	1	1	0	1	0
0	1	1	0	0	0	0	1	0	0	1	0	0
0	1	1	0	0	1	0	1	0	0	1	1	1
0	1	1	0	1	0	0	1	1	0	1	0	1
0	1	1	0	1	1	0	1	1	0	1	1	0
0	1	1	1	0	0	0	1	0	1	1	0	1
0	1	1	1	0	1	0	1	0	1	1	1	0
0	1	1	1	1	0	0	1	1	1	1	0	0
0	1	1	1	1	1	0	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	0	0	1	1	0	0	0	1	0	0
1	0	0	0	1	0	1	0	0	0	0	1	0
1	0	0	0	1	1	1	0	0	0	1	1	1
1	0	0	1	0	0	1	0	1	0	0	0	0
1	0	0	1	0	1	1	0	1	0	1	0	1
1	0	0	1	1	0	1	0	1	0	0	1	1
1	0	0	1	1	1	1	0	1	0	1	1	0
1	0	1	0	0	0	1	0	0	1	0	0	0
1	0	1	0	0	1	1	0	0	1	1	0	1
1	0	1	0	1	0	1	0	0	1	0	1	1
1	0	1	0	1	1	1	0	0	1	1	1	0
1	0	1	1	0	0	1	0	1	1	0	0	1
1	0	1	1	0	1	1	0	1	1	1	0	0
1	0	1	1	1	0	1	0	1	1	0	1	0
1	0	1	1	1	1	1	0	1	1	1	1	1
1	1	0	0	0	0	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	1	0	0	0	1
1	1	0	0	1	0	1	1	0	0	0	1	1
1	1	0	0	1	1	1	1	1	0	0	1	0
1	1	0	1	0	0	1	1	0	0	1	0	1
1	1	0	1	0	1	1	1	1	0	1	0	0
1	1	0	1	1	0	1	1	0	0	1	1	0
1	1	0	1	1	1	1	1	1	0	1	1	1
1	1	1	0	0	0	1	1	0	1	0	0	1
1	1	1	0	0	1	1	1	1	1	0	0	0
1	1	1	0	1	0	1	1	0	1	0	1	0
1	1	1	0	1	1	1	1	1	1	0	1	1
1	1	1	1	0	0	1	1	0	1	1	0	0
1	1	1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	0	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0

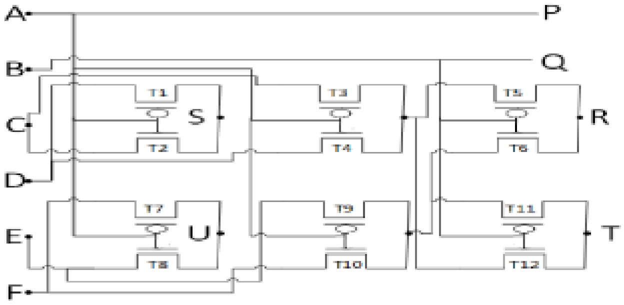

Fig. 4: Transistor Realization of the Existing MSB Gate

3.1.3 Physical Implementation of Existing Gates:

Transistor realizations of the Existing MSH gate and MSB gate are demonstrated in Fig. 3 and Fig. 4 which require 10 and 12 transistors, respectively. Now, for MSH gate, suppose, when inputs of A=0, B=1, C=1 and D=0 then output P is 1, transistor T1 is off and transistor T2 is on, so C is propagated to R. Since C is 1, output at R is also 1. Again, both of B and C is 1, transistors T3 and T5 are off and T4 and T6 are on and pass the *GND* (Ground) voltage or 0 to Q.

Similarly, R=1 and D=0, so transistors T7 and T10 are on, transistors T8 and T9 are off and output found at S is 1. So, finally for input combination (A=0, B=1, C=1 and D=0), output combination found is (P=0, Q=0, R=1 and S=1) as tabulated in Table I which proves the correctness of the physical implementation of the proposed MSH gate.

On the other hand, let us consider MSB gate and for input combination (A=0, B=1, C=0, D=1, E=0 and F=1), transistors T2, T4, T5, T8, T10 and T11 are off and T1, T3, T6, T7, T9 and T12 are on. Input of D and F will be propagated to the output of S and U respectively and S=1 and U=1 since both D and F is 1. Again, transistors T3 and T9 will propagate the input of C and E to the transistors T6 and T12, respectively and outputs obtained at R and T are 0 since both of C and E is 0.

Finally, the generated output is (P=0, Q=1, R=0, S=1, T=0 and U=1) for input combination (A=0, B=1, C=0, D=1, E=0 and F=1). It can be also shown for other input combinations as well to prove the correctness of the physical implementation of the Existing MSB gate.

3.2 Existing Reversible Fault Tolerant Flip-Flops:

The architecture of Existing reversible fault tolerant D-flip-flop using one MSH gate to produce the desired output, S=*Clk_. Feedback* ⊕ *Clk.D* with only two garbage *G*1 and *G*2 and one constant input. Fig.5(b) illustrates the architecture of Existing reversible fault tolerant write enable master-slave flip flop using one FRG, two F2G and two reversible fault tolerant D-flip-flop(which described earlier) to produce the desired output, Q and *Q*+.

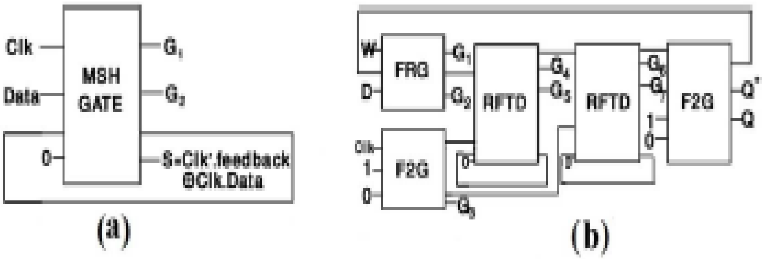

Fig. 5: Reversible Fault Tolerant D-Flip-Flop &Master slave flip-flop

IV. DESIGNING FLIP-FLOPS USING RR GATE

This particular problem of VLSI designing was realized by Feynman and Bennet in 1970s. In 1973 Bennet had shown that energy dissipation problem of VLSI circuits can be overcome by using reversible logic. This is due to the fact that in reversible computation there no loss of information or bits and consequently it does not dissipate any energy for computation. Reversible computation requires circuits with reversible logic and synthesis of such circuits differs significantly from its irreversible counter part because of different factors.[1]A reversible logic gate must have the same number of inputs and outputs, and for each input pattern there must be a unique output pattern.

Thus, Reversible logic circuits avoid energy loss by un computing the computed information by recycling the energy in the system [3]. In the design of reversible circuits two restrictions should be considered [4]; firstly, Fan-out is not permitted and secondly, Feedback from gate outputs to inputs is not permitted. Due to these restrictions, synthesis of reversible circuits can be carried out from the inputs towards the outputs and vice versa [5]. So, there should be one-to-one mapping between input vector and output vector. In an n-output reversible gate, the input number is equal to the no of output. A logic synthesis technique using a reversible gate should have the features like minimum gate count along with less use of constants and garbage generation.

4.1 Reversible Logic Gates:

A logic gate is reversible if the mapping of inputs to outputs is bijective, that is, every distinct input pattern a distinct output pattern will be produced, and equal no of input and output are available. If it has input k*k(and outputs), we call it a reversible gate.[7] Realization of reversible function using gates with smaller width increases the gate count and garbage outputs. Therefore, there must be a tradeoff of using a family of reversible gates.

There are many reversible gates in the literature. Among them are 2*2 Feynman Gate (FG) [3], 3*3 Peres Gate (PG) [11], 3*3 Toffoli Gate (TG) [14], 3*3 Fredkin Gate (FRG) [4], 3*3 Khan Gate (NG) [8], 3*3 double Gate (F2G) [10], and 3*3 NFT [5]. Any realization techniques should keep both the number of constants and garbage's as low as possible [8]. Garbage Outputs, Constant Inputs, Gate Count, Hardware Complexity, Area Consumption, Path Delay should be made minimum.

4.2 Fault Tolerant Reversible Logic:

Fault tolerance enables a system to continue its operations correctly when an error occurs in some parts of it. Parity checking is one of the widely used mechanisms for detecting single level fault in Communication and many other systems. It is believed that if the parity of the input data is maintained throughout the computation, no intermediate checking would be required [11, 12]. Therefore, parity preserving reversible circuits will be the future design trends towards the development of fault tolerant reversible systems.

A gating network will be parity preserving if its individual gates are parity preserving [11]. Thus, we need parity preserving reversible logic gates to construct parity preserving reversible circuits [9][10]. Agrawal [13] has shown that fault-detection probability gets maximum when the output information of a circuit is maximized. Thus, we find that the fault detection would be much easier in the reversible logic, where the information loss is less, than in irreversible ones. Under the multiple fault the full pattern has to be checked in order to find the fault in the circuits for growing number of inputs and the number of gates. This provides additional motivation for studying reversible circuits, namely they may be much easier to test than their irreversible counterparts.

A reversible logic circuit design should be optimized for following criteria:

- minimum number of reversible gates
- minimum number of garbage outputs
- minimum number of constant inputs
- The minimum quantum cost of circuit

We first note that of the gates depicted, only the Fredkin gate is parity-preserving. This is readily verified by comparing the input parity A ⊕ B ⊕ C (or A ⊕ B for the Feynman gate) to the output parity P ⊕ Q ⊕ R (or P ⊕ Q). The Feynman gate is quite useful, but, unfortunately, it is inadequate for the synthesis of efficient reversible circuits.

Given thatsynthesis methods with the Toffoli gate, using Fredkin gates to assist in optimizing cost or performance, are quite advanced, we are motivated to look for additional reversible gates that would lead to similarly efficient designs. In this search, the following impossibility result rules out a fundamental role for 2-input, 2-output gates. No 2-input, 2-output reversible gate can be parity preserving. we next look into 3-input, 3-output gates. We have already observed that the Fredkin gate is parity-preserving. Well known fault tolerant gates are [8] New Fault Tolerant Gate, Fredkin gate, Feynman double gate, Islam gate.

Fredkin gate have been constructed the reversible gates. It uses less no of transistors than conventional CMOS based reversible gates. This reduces the area and as well as to operate the circuit with low power consumption and the speed is increased with the reduction in delay. The source of energy supplied to the system is only through the input signals only which will greatly favor for the future designs. A fundamental conservative reversible logic gate is the Fredkin gate has been designed using 10 transistors.

4.3 Design of RR Parity Preserving Fault Tolerant Reversible Gate:

This paper presents a new 4*4 parity preserving reversible gate, RR Gate (RR), depicted in Figure 2. it has four input and four output wires and it can also be said as parity preserving fault tolerant reversible gate. The gate is one through, which means one of the input variables is also output. The truth table of the RR- gate is shown in Table 1. It can be readily verified from the truth table that the input pattern corresponding to particular output pattern can be uniquely determined. The proposed reversible RR gate is parity preserving. This is readily verified by comparing the input parity A⊕B⊕C⊕D to the output parity P⊕Q⊕R⊕S.

Thereare other two 4*4 reversible gates in the literature, namely TSG and MKG. Though these two gates are reversible but not parity-preserving and therefore do not allow detection of circuit’s faults.

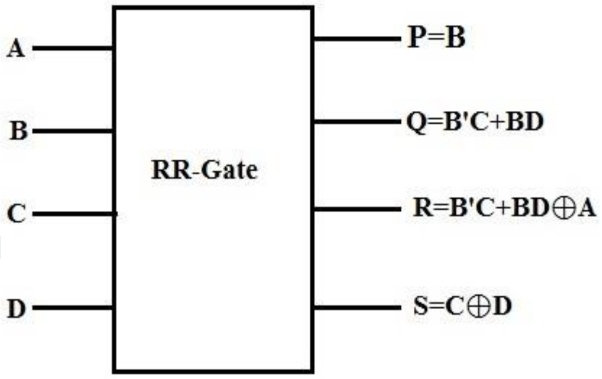

Fig. 6: Proposed RR 4*4 Fault Tolerant gate.

Table 3: Truth table for proposed RR 4*4 fault tolerant gate.

A	B	C	D	P	Q	R	S
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	1	0	0	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	0	1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	1	0	0	1	0	1	0
1	1	0	1	1	1	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	1	0	0

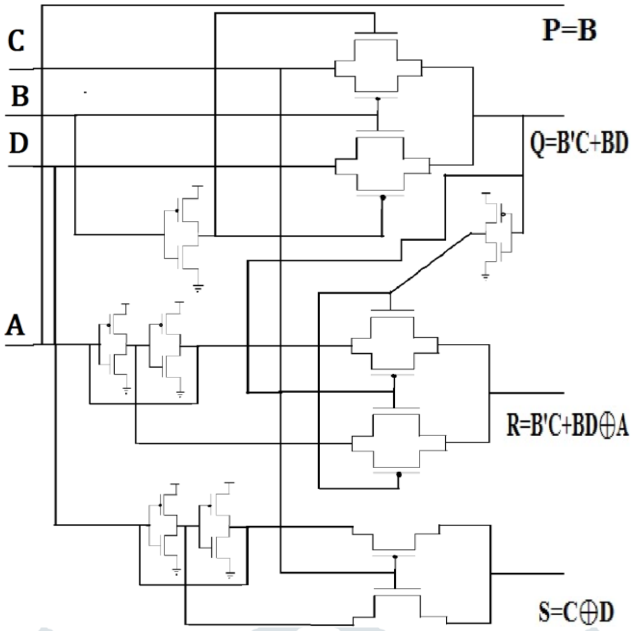

Fig. 7: Transistor Design of RR 4*4 Fault Tolerant gate.

Flip-flops and latches are the one-bit storage element. such circuits which store the data or the state is defined as the sequential circuit. The D Flip-flop make the transaction when it is enabled by the clock. Flip-flops are edge triggered. It can be classified into single edge triggered and double edge triggered. This paper we are going to see about single edge triggered.

When the clock is high then the data will be transmitted to the output making the master to work. When clock is low the storage or the feedback condition take at this time slave will be in the active state and the master will be off condition and vice versa. D Flip-flop is defined as the delay flip-flop because it can't make any necessary transmission unless until it is enabled by the clock. This paper provides the design of D-Flip-flop, using the proposed 4*4 Fault tolerant gate in Fig.4.the advantage of this design is that the designs using this RR fault tolerant gate will preserve parity. Such that the fault or error detection will be simple.

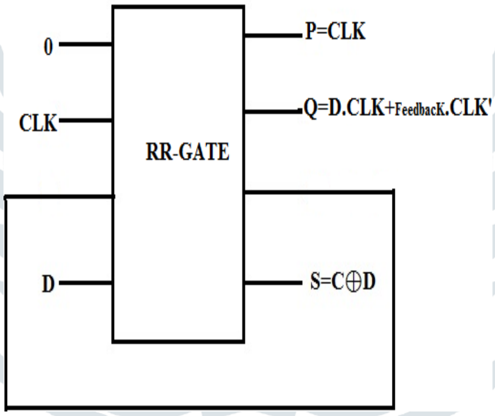

Fig. 8: D-flip-flop using RR Gate.

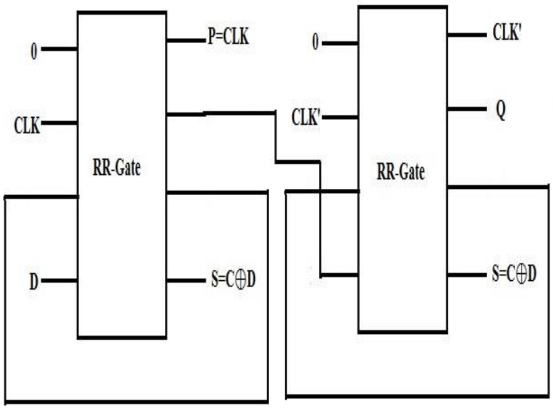

Fig. 9: Master-Slave D-Flip-flop Using RR Gate.

V. RESULTS

The present document introduces the design and simulation of CMOS integrated circuits, in an attractive way thanks to user-friendly PC tools DSCH and MICROWIND.

5.1 Comparison Results of D-Flip-flop:

Table 4: Comparison Results of D Flip-Flop

	CONVENTIONAL	PROPOSED
DELAY (ns)	1.2	0.82
POWER (mw)	1.5	1.2
PDP(PJ)	1.8	0.984
No. OF GATES	1	1

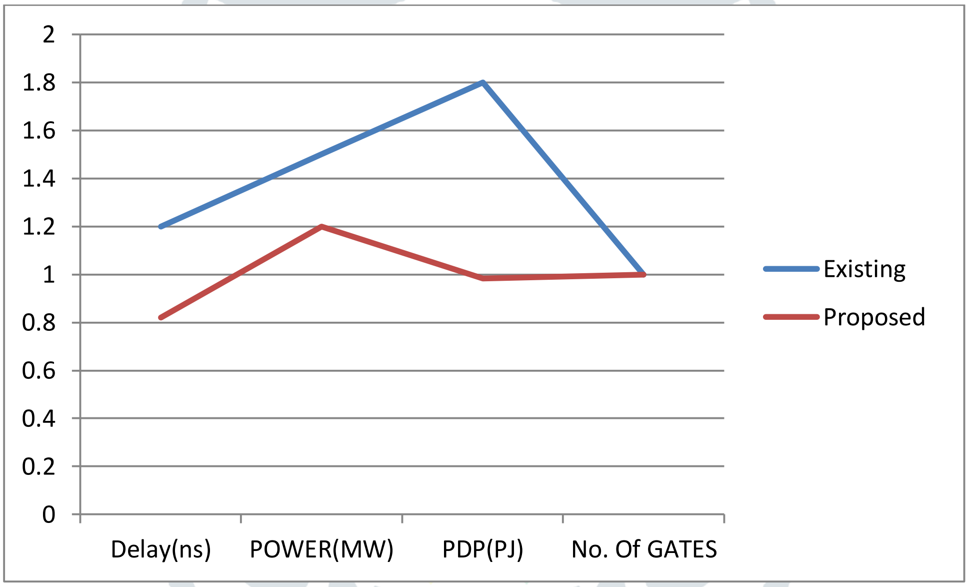

Fig. 10: Comparison Results of D Flip-Flop

5.2. Comparison Results of Master Slave Flip-Flop:

Table 5: Comparison Results of Master Slave Flip-Flop:

	CONVENTIONAL	PROPOSED
DELAY (ns)	**10.2**	**9.8**
POWER (mw)	**7.2**	**2.4**
PDP(PJ)	**73.44**	**23.52**
No. OF GATES	**5**	**2**

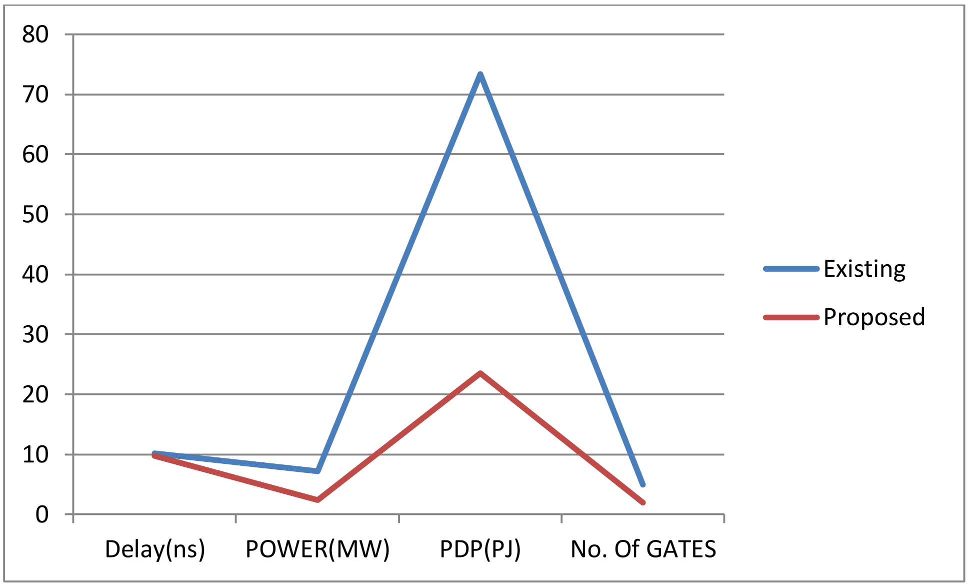

Fig. 11: Comparison Results of Master Slave Flip-Flop

5.3. Digital Schematic Circuits:

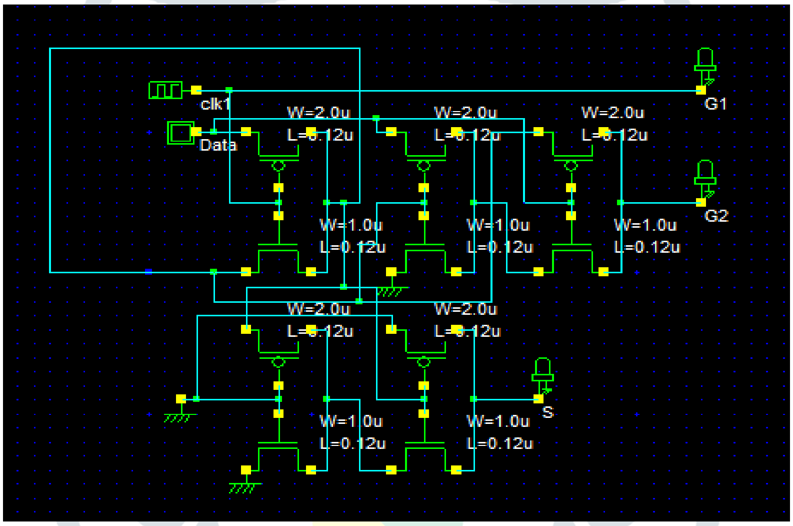

Fig. 12: D-Flip Flop Using MSH Gate

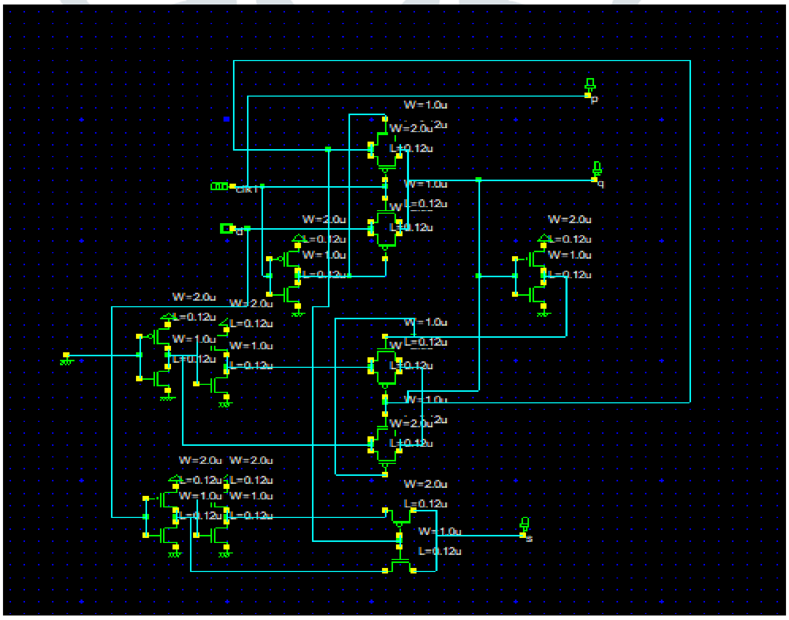

Fig. 13: D-Flip Flop Using RR Gate.

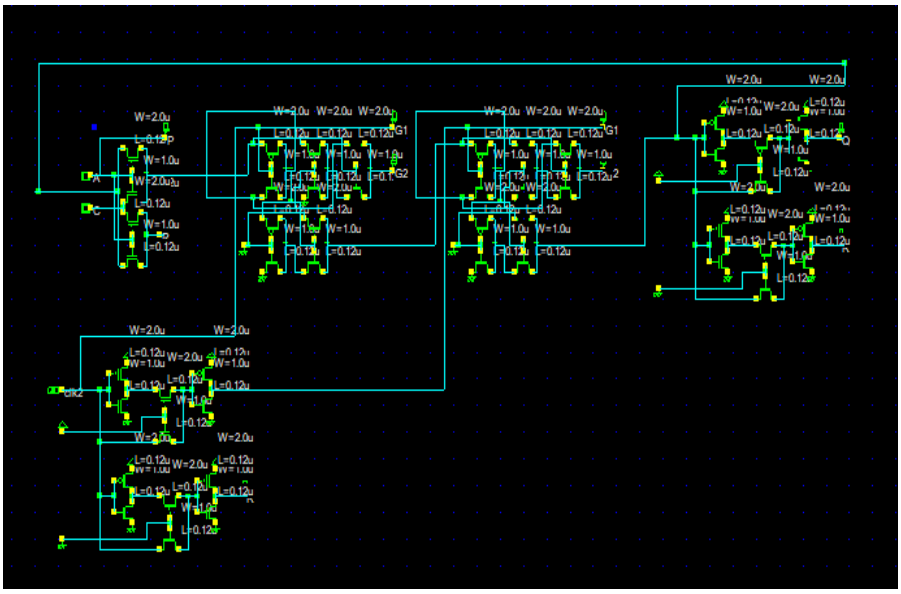

Fig. 14: Master Slave Flip-Flop Using MSH, FRG AND F2G Gate

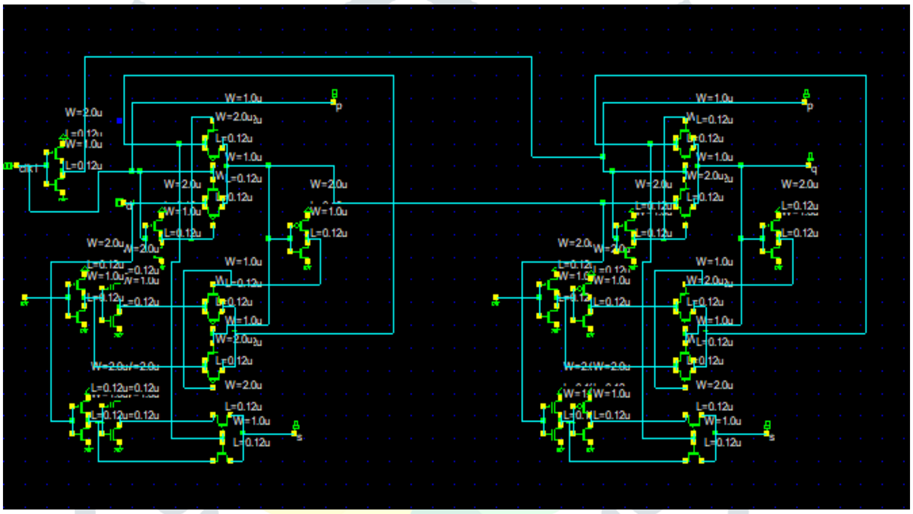

Fig. 15: Master Slave Flip-Flop Using RR Gate

5.4. Simulation Results:

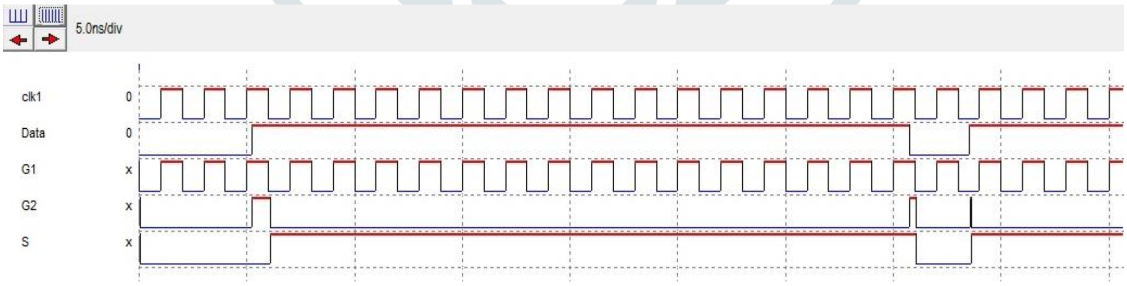

Fig. 16: Simulation of D-Flip Flop Using MSH Gate

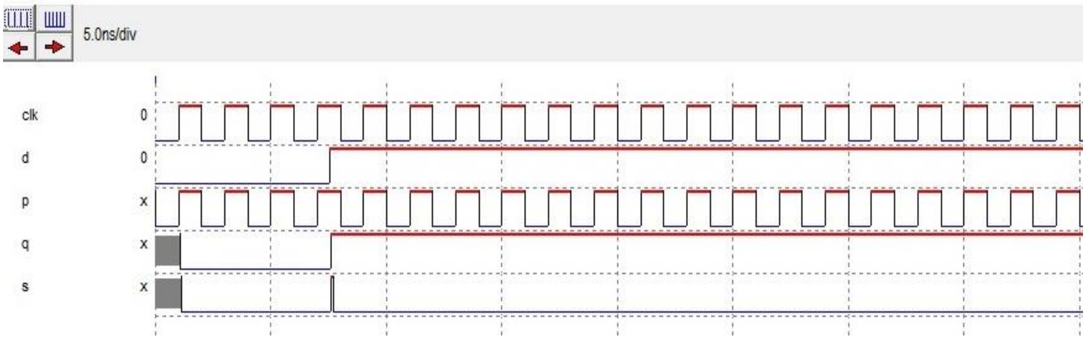

Fig. 17:Simulation of D-Flip Flop Using RR Gate

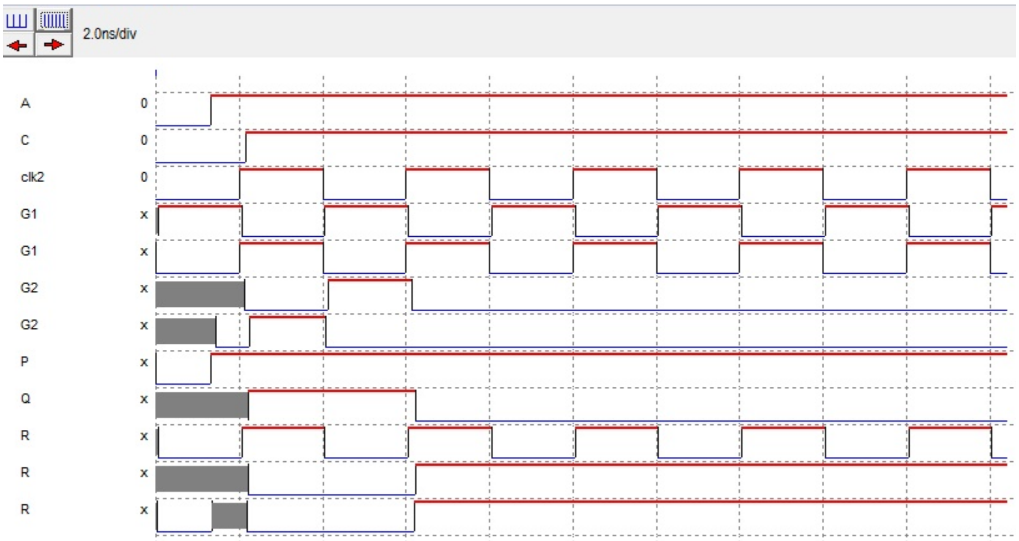

Fig. 18: Simulation of Master Slave Flip-Flop Using MSH, FRG AND F2G Gate

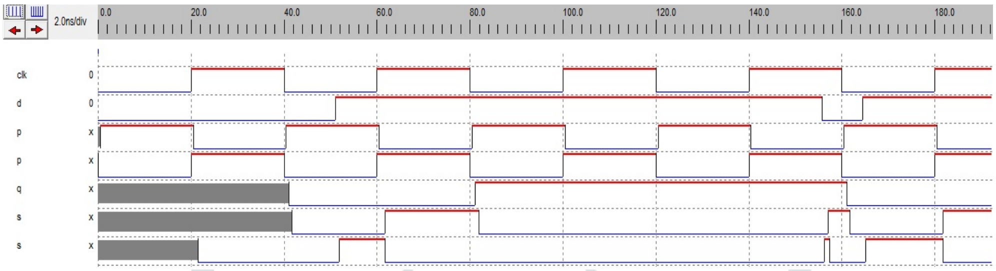

Fig. 19: Simulation of Master Slave Flip-Flop Using RR Gate

VI. CONCLUSION

We have given an overview of the k*k parity preserving reversible gates. An efficient Fault tolerant reversible logic RR-gate circuit has been presented. This thesis presents a novel realization of fault tolerant reversible D-FLIPFLOP AND Master Slave flip-flop using Digital Schematic. The power, delay, power delay product and the hardware complexity are compared with the existing and the proposed design. The obtained values delay (0.82 ns), power (1.2 mw) and power delay product (0.984) of proposed D-flip-flop are better than the existing. The obtained values delay (9.8 ns), power (2.4 mw) and power delay product (23.52) of master slave flip-flop are better than the existing.

REFERENCES

1. R. Landauer, "Irreversibility and heat generation in the computing process," IBM J. Research and Development, vol.3, pp. 183-191, July1961.
2. M. Nielsen, I. Chuang. Quantum computation and quantum information. Cambridge, UK: Cambridge University Press,2000.
3. A. Khazamipour and K. Radecka, “Adiabatic Implementation of Reversible Gates”, Proc. MSCAS,2005.
4. C.Bennett, “Logical reversibility of computation,” IBM J.Res.Dev.,vol.17.
5. R. Feynman, “Quantum mechanical computers”, Optic News, 11:11-20,1985.
6. T. Toffoli. Reversible computing, Tech memo MATLABforComp.Sci,1980.
7. E. Fredkin, T. Toffoli, “Conservative logic”, International Journal of Theoretical Physics.
8. Michael P. Frank, “Reversible Computing,” invited article, Developer 2.0 magazine, JasubhaiDigital Media, January2004.
9. D. P. Vasudevan, P. K. Lala, J. P. Parkerson: CMOS Realization of Online Testable Reversible Logic Gates. ISVLSI 2005:309-310.