DUJASE Vol. 2(1) 31-38, 2011 (July)

Design of A Reversible Random Access Memory

F. K. M. Naimul Huda[1], Shahed Anwar[2], Lafifa Jamal[2] and Hafiz Md. Hasan Babu[2]

[1]Therap Bd Ltd. Dhaka, Bangladesh

[1]Department of Computer Science & Engineering University of Dhaka, Dhaka, Bangladesh.

[2]Therap Bd Ltd. Dhaka, Bangladesh.

E-mail : mdnhuda@gmail.com , shahedanwar01@yahoo.com, lafifa@univdhaka.edu, hafizbabu@hotmail.com

Received on 19. 02. 2011. Accepted for publication on 02. 08. 2011

Abstract

In this paper, we have proposed a new $n \times n$ reversible gate (NH Gate), which is extensively used to design different reversible circuits. On the way to design the RRAM, NH gate based Reversible Decoder, Reversible D-Latch and Reversible Flip-flops are proposed. We have also analyzed the proposed circuits with the existing ones and the comparative result shows that the proposed designs outperform the existing designs in terms of all the parameters of reversible logic synthesis. Finally, we have evaluated the proposed RRAM with Lemmas and Theorems.

Keywords : Reversible Logic, Random Access Memory (RAM), Flip-flop, Latch, Decoder, Garbage Output.

1. Introduction

In a reversible circuit the number of inputs is equal to the number of outputs and there is a one-to-one mapping between the input vector and the output vector, thus the input state can be reconstructed from the output state. In recent years reversible logic has emerged as a promising research area. The main reason for this is the growing demand of low powered device. Our society is moving towards low powered mobile devices. Laundauer's principle [1] state that, logic computations that are not reversible generate heat $kT*ln2$ for every bit of information that is lost.

Reversible logic gates are compatible with the novel computing models such as Quantum and Optical computing [2, 3]. Reversible logic is of growing importance to many future computer technologies. Many researchers are working on reversible combinational logic but little work is done in the area of sequential reversible logic.

In this paper, we first presented a generalized reversible gate. The proposed gate provides much flexibility and opportunity for the researchers to use any of the $n\times n$ version of this family. Another key feature of the paper is to propose a Reversible Random Access Memory (RRAM). The proposed RRAM incorporates all good features of reversible logic synthesis and the efficiency is proved with the help of Lemmas and Theorems. In order to design the RRAM, we proposed some sequential circuits with the help of the newly proposed gate.

2. Basic Definitions of Reversible Logic

In this section, we have presented some definitions and basic idea on reversible logic.

Reversible gate: A reversible gate is an $n \times n$ data stripe block which uniquely maps between input vector $I_v = (I_o, I_1, \ldots I_n)$ and output vector $O_v = (O_0, O_1, \ldots\ldots O_n)$ denoted as $I_v \leftrightarrow O_v$ [4].

Garbage Output: Unwanted or unused output of a reversible gate (or circuit) is known as Garbage output. More formally, the outputs, which are needed only to maintain reversibility, are called garbage outputs.

Delay: The delay of a logic circuit is the maximum number of gates in a path from any input line to any output line. This definition is based on the following assumptions:

i. Each gate performs computation in one unit time.

ii. All inputs to the circuit are available before the computation begins.

Feynman Gate: The input vector, I_v and output vector, O_v for 2 * 2 Feynman Gate *(FG)*[5] is defined as follows: $I_v = (A, B)$ and $O_v = (P = A, Q = A \oplus B)$.

Toffoli Gate: The input vector, I_v and output vector, O_v for 3 * 3 Toffoli Gate *(TG)*[6] is defined as follows: $I_v = (A, B, C)$ and $O_v = (P = A, Q = B, R = AB \oplus C)$.

Fredkin Gate: The input vector, I_v and output vector, O_v for 3 * 3 Fredkin Gate *(FRG)*[7] is defined_as follows: $I_v = (\underline{A}, B, C)$ and $O_v = (P = A, Q = AB \oplus AC, R = AC \oplus AB)$.

Peres gate*:* The input vector, I_v and output vector, O_v for 3 * 3 Peres Gate *(PG)*[8] is defined as follows: $I_v = (A, B, C)$ and $O_v = (P = A, Q = A \oplus B, R = AB \oplus C)$.

Generalized Reversible Family: Input-Output relation of a generalized reversible family of gates can be expressed by the below mentioned equation

$$O_m = I_1 \oplus I_2 \oplus \ldots \oplus I_{m-1} \oplus I_m \quad \ldots \qquad (1)$$

where $I_1, I_2, \ldots, I_m$ denote the inputs and $O_1, O_2, \ldots, O_m$ denote the outputs. According to this equation, mth output of any gate of this family is equivalent to the ex-or of the mth input along with all previous input to the gate. The block diagram of generalized reversible family is shown in Fig 2.1.

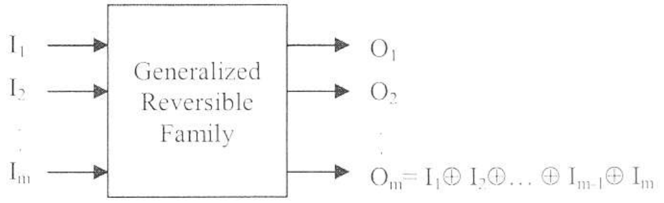

I_1
I_2
I_m
Generalized Reversible Family
O_1
O_2
$O_m = I_1 \oplus I_2 \oplus \ldots \oplus I_{m-1} \oplus I_m$

Fig. 2.1: Block diagram of a generalized reversible family

As the 2*2 Feynman gate[5] remains at the top of this family, we name this family as the GFG or Generalized Feynman Gate family. The input vector, I_v and output vector, O_v for 3 * 3 GFG gate is defined as follows: $I_v = (A, B, C)$ and $O_v = (P = A, Q = A \oplus B, R = A \oplus B \oplus C)$.

Quantum Cost: The 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 0 and that of any 2*2 gate is the same, which is 1 [9]. So, the quantum cost of a 2*2 Feynman gate is 1. Again, the quantum cost of a 3*3 Toffoli, Fredkin and Peres gate is 5, 5 and 4 respectively [10],[9],[5].

3. Implementation of Reversible Decoder using NH Gate

This section is divided into two sub-sections. In the first sub-section we propose a new reversible gate named NH gate and in the next sub-section we define the design methodology of reversible decoder.

3.1. NH Gate: Proposed New Reversible Gate

The proposed NH gate is an $n \times n$ ($n \geq 3$) 1-through gate, where 1 input will remain intact in the output. The proposed NH gate is shown in Fig. 3.1.1. The NH gate can be defined by two different mathematical notations but the resulting values are same.

In the first approach, O_1 is equal to the first input I_1, and the remaining inputs are defined using following equation

$$O_m = ((I_1 \oplus (m\%2)).\ I_2.I_3 \ldots I_{\left\lfloor \frac{m}{2} \right\rfloor+1}) \oplus I_{\left\lfloor \frac{m}{2} \right\rfloor+2} \oplus (I_z \oplus I_{z+1} \ldots \oplus I_m)) \quad \ldots (2)$$

where $z = \left\lfloor \frac{m}{2} \right\rfloor + 3$

Equation (2) can be comprehended using following equation

$$O_m = \begin{cases} I_1 & \ldots\ m = 1 \\ I_1 I_2 \oplus I_3 & \ldots\ m = 2 \\ ((I_1 \oplus m\%2).I_2.I_3 \ldots I_{\lfloor m/2 \rfloor+1}) \oplus I_{\lfloor m/2 \rfloor+2} \oplus I_{\lfloor m/2 \rfloor+3} \oplus \ldots \oplus I_m & \ldots 3 \leq m \leq n \end{cases} \quad \ldots (3)$$

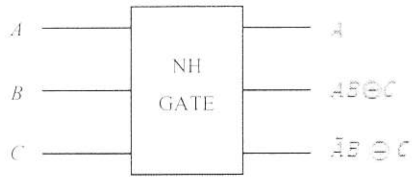


Fig. 3.1.1: 3*3 gate of the proposed family.

Table 3.1.1: Truth Table for the proposed gate

Input			Output		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	1	0	1

Fig. 3.1.2: shows the quantum equivalent representation of NH gate. The circuit consists of two square root of NOT gate (v) each costs 1, one hermitian of square root of NOT (v^+) and one EX-OR gate in the rectangle which costs 1 and two EX-OR gates each costs 1. So, the quantum cost of the proposed gate is (2+1+2) =5 which is equivalent to Toffoli and Fredkin gate [10, 9].

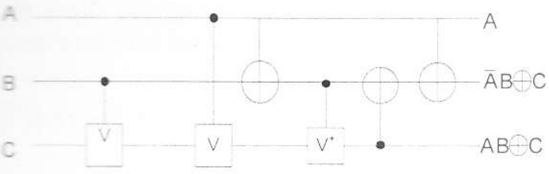


Fig. 3.1.2: Quantum equivalent realization of NH Gate

3.2 Design of a Reversible Decoder

In this section, we construct the reversible decoder using the proposed NH gate. Two 3 x 3 NH gates can be used to produce a reversible implementation of a 2 x 4 decoder. If A, B and 0 are the inputs of a NH gate, the gate generates the output A, AB and $\bar{A}B$ respectively. Thus a NH gate produces two output of a two input decoder. Two other outputs $A\bar{B}$ and $\bar{A}\bar{B}$, can be obtained by presenting input $\bar{A}$ and $\bar{B}$ to another NH gate. Feynman gates can be used to produce complements of the inputs. The proposed 2 x 4 decoder is shown in Fig. 3.2.1

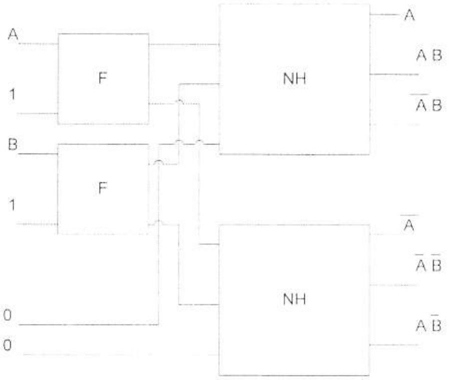


Fig. 3.2.1: 2 × 4 Reversible Decoder

This decoder will be used to realize larger decoders. We now present an algorithm to construct n input decoder. We denote a 3 x 3 NH gate as, N (I_1, I_2, I_3) = (O_1, O_2, O_3), where N [I_1], N [I_2] and N [I_3] denote three inputs to this gate and N [O_1], N [O_2] and N [O_3] are the outputs.

Algorithm 3.2.1: REVERSIBLE_DECODER (*n*)

```
Take one 2 X 4 reversible decoder. Outputs of this decoder are form Level $L_2$
For each level $L_i$, where i = 3 to n Loop
    For j = 1 to $2^{i-1}$ Loop
        Take one NH gate $N_j$
            If j = 1 then
                $N_j [I_1] := A_i$
            Else
                $N_j [I_1] := N_{j-1} [O_1]$
            End if
        $N_j [I_2] :=$ j^th output from $L_{i-1}$
        $N_j [I_3] := 0$
        Set $N_j [O_2]$ and $N_j [O_3]$ as outputs of Level $L_i$
    End loop
End loop
End
```

Fig. 3.2.2 shows the $n \times 2^n$ reversible decoder according to Algorithm 3.2.1 whereas, the 3 x 8 realization is shown in Fig. 3.2.3. This decoder has no garbage outputs, as primary inputs are not considered as garbage [5].

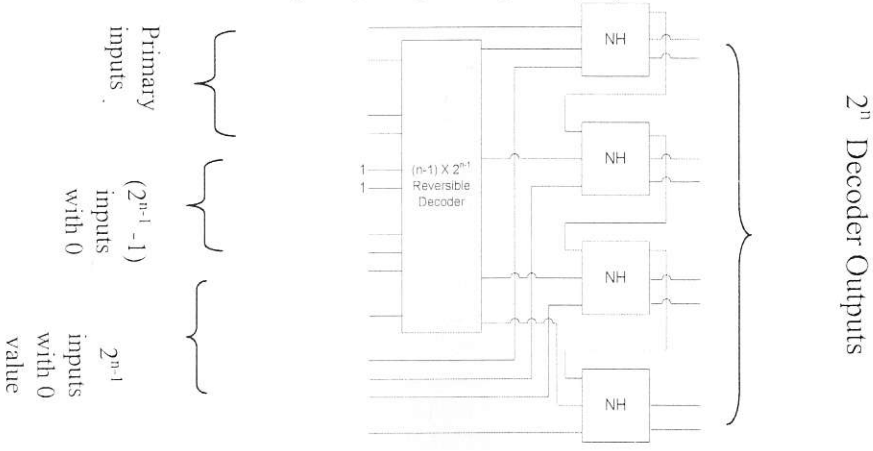

Primary inputs
(2^{n-1} -1) inputs with 0
2^{n-1} inputs with 0 value
1
1
(n-1) X 2^{n-1} Reversible Decoder
NH
NH
NH
NH
2^n Decoder Outputs

Fig. 3.2.2: $n \times 2^n$ Reversible Decoder

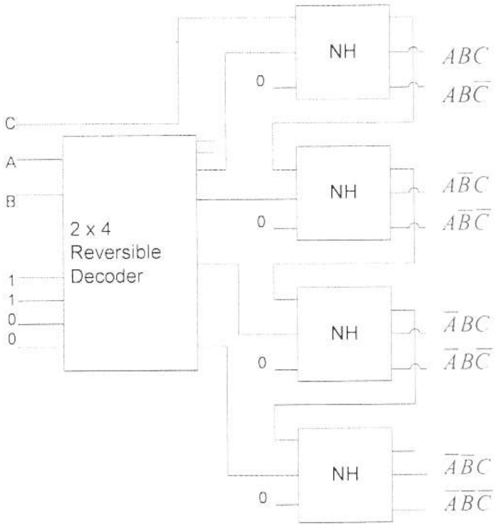

C
A
B
1
1
0
0
2 x 4
Reversible
Decoder
0
NH
ABC
$AB\overline{C}$
$A\overline{B}C$
$A\overline{B}\overline{C}$
$\overline{A}BC$
$\overline{A}B\overline{C}$
$\overline{A}\overline{B}C$
$\overline{A}\overline{B}\overline{C}$

Fig. 3. 2. 3: 3 × 8 Reversible Decoder

Fig. 3.2.3 shows the circuit of the first approach. Fig. 3.2.4 shows an alternative Decoder using another approach where delay can be drastically reduced if primary inputs are directly provided in all the gates of each level other than reusing. As there are three levels, the delay for the decoder of Fig. 3.2.4 is only 3. In this way an n-input decoder has delay at most n, which is better than the delay of the first approach (Fig. 3.2.3). This implementation is obviously better than any other implementation using existing reversible gates.

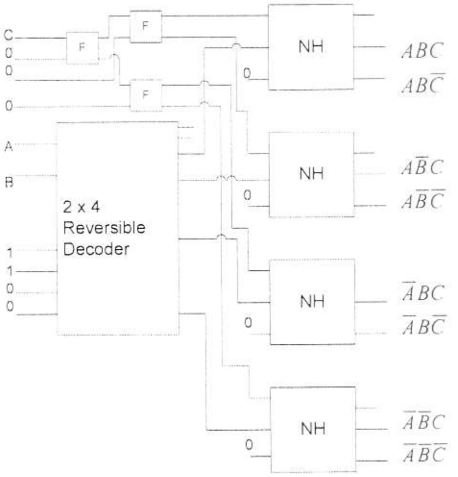

C
0
0
0
F
F
F
A
B
1
1
0
0
2 x 4
Reversible
Decoder
0
NH
ABC
$AB\overline{C}$
$A\overline{B}C$
$A\overline{B}\overline{C}$
$\overline{A}BC$
$\overline{A}B\overline{C}$
$\overline{A}\overline{B}C$
$\overline{A}\overline{B}\overline{C}$

Fig. 3. 2. 4: 3 X 8 Reversible Decoder (less delay)

However, it is possible to design the decoder using other $n \times n$ gates for $n>3$, but to reduce the internal complexity of the circuit, we have selected the value of n less than 4.

Lemma 3.1: *A Reversible $n \times 2^n$ Decoder can be realized by at least 2^n number of reversible gates in approach1.*

Proof: According to Fig. 3.2.1, 2×2^2 decoder can be realized with $4\ (=2^2)$ reversible gates (2 Feynman Gates and two NH Gates). More 2^2 gates are required to realize 3×2^3 decoder (In Fig. 3.2.3). So, $2^2 + 2^2\ (=2^1 \ x\ 2^2 = 2^3)$ reversible gates are required to realize the 3×2^3 decoder in total and hence 2^n reversible gates are required to realize the $n \times 2^n$ decoder.

Lemma 3.2: *A Reversible $n \times 2^n$ Decoder generates at least n garbage outputs in approach1.*

Proof: 2×2^2 decoder generates 2 garbage outputs in the first approach. One extra garbage output (3 in total) should be generated from the last NH gate of the 3×2^3 decode and hence at least n garbage outputs will be generated to realize the $n \times 2^n$ decode using first approach.

Lemma 3.3: *A Reversible $n \times 2^n$ Decoder can be realized with at least 2^n-2 unit delay in approach1.*

Proof: 2×2^2 decoder requires 2 Unit delay in the first approach and more 2^{3-1} Unit delay are required to realize the 2×2^3. That is, $n \times 2^n$decoder requires 2^n unit delay, except 2×2^2 decoder requires 2^n-2 unit delay. As 2×2^2 decoder is the first sub system inside the n × 2^n decode, so n × 2^n decoder requires 2^n-2 Unit delay.

Lemma 3.4: *The Quantum cost of the Reversible $n \times 2^n$ Decoder is at least $5x2^n - 8$*

Proof: An $n \times 2^n$ reversible decoder can be realized with 2^n -2 number of NH gate and two Feynman gate. We have already shown in Fig. 3.1.2, that each NH gate has Quantum cost of 5. Therefore all the NH gates in the circuit cost at least $5(2^n\ -2)$ and two Feynman gate cost 2. So, total Quantum cost is $5 \times 2^n - 8$.

4. Design of the Reversible Random Access Memory (RRAM)

In the first part of this section, different existing D-FFs are discussed and after that, the proposed reversible D-FF and reversible write enabled D-FF are shown. The key contribution of this paper, Reversible Random Access Memory (RRAM), is shown at the end of this section. Proposed RRAM uses the propounded reversible write-enabled D-FF and the reversible decoder.

4.1. Design of Reversible Latch and Flip-Flop

In the first part of this section, different reversible designs of D-FFs are discussed. We have proposed some new implementations and analyzed the Quantum complexity along with the gate and garbage requirements.[11]

The characteristic equation of the D latch can be written as $Q+ = D.E + E'Q$. The characteristic equation of the D latch can be mapped onto the Fredkin gate [12,13]. Fig. 4.1.1 shows the realization of the proposed reversible D latch. To avoid a fan-out problem, another Fredkin gate is used to copy the output. This implementation requires two Fredkin gates to emulate the same behavior of a D latch. Quantum cost of a Fredkin gate is 5. In this circuit, second Fredkin gate is used only to copy the third output of the first Fredkin gate. Same result can be obtained by using a Peres gate, as shown in Fig. 4.1.2. In this case, Quantum cost of the circuit will be less than that of Fig. 4.1.1, since Quantum cost of a Peres gate is 4. The Quantum cost of the circuit of Fig. 4.1.2 is (5 + 4) = 9. Better result is achieved if we use 3*3 GFG gate instead of a Peres gate, as its Quantum cost is only 2. So, the Quantum cost of the circuit of Fig 4.1.3 is $(5 + 2) = 7$. Comparison of Quantum cost of different reversible implementation of D latch is shown in table 4.1.1. Unlike the design of [12,13] we will reuse the D latch enable input (E) or otherwise stated 'Clock Pulse' in the circuit using this D latch. So number of garbage output is reduced to 1 only.

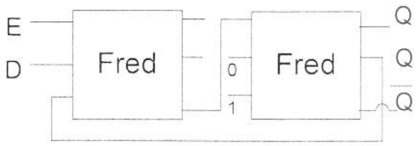


Fig. 4.1.1: Reversible D-Latch [12,13]

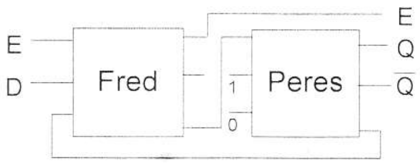


Fig. 4.1.2: Proposed Reversible D-Latch (with one Peres gate)

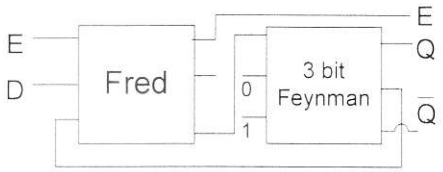
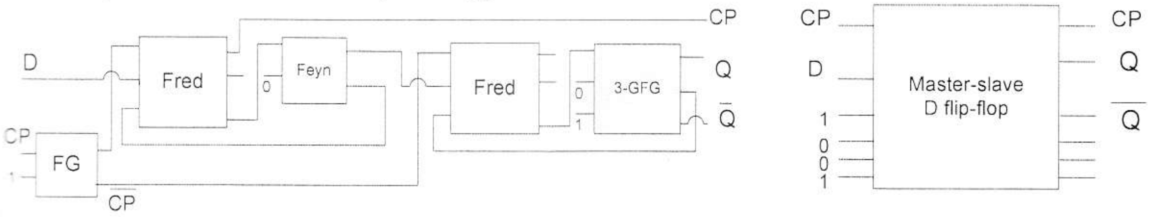


Fig. 4.1.3: Proposed Reversible D-Latch (with one 3-GFG gate)

Table 4.1.1: Comparison between different designs of D latch.

Methods	*Quantum Cost*	*Number of Gates*	*Number of Garbage output*
Existing method [12,13]	10	2	2
Proposed method-1 (figure 4.1.2)	9	2	1
Proposed method-2 (figure 4.1.3)	7	2	1

We propose a new master-slave D flip-flop in Fig. 4.1.4. Since master and slave D latches are activated on opposite pulse of the clock, we are using a Feynman gate with second input as 1. First output of this Feynman gate triggers the master and second output i.e. the inverted clock pulse triggers the slave D latch.



(a) Gate-level Diagram

(b) Block-level diagram

Fig. 4.1.4: Proposed master-slave D flip-flop

The proposed master-slave D flip-flop has two Fredkin gate, two Feynman gate and 1 3-GFG gate. So the Quantum cost of the circuit is $(2 * 5) + (2 * 1) + 2 = 14$.

Table 4.1.2: Comparison between different designs of master-slave D flip-flops.

Methods	*Quantum Cost*	*Number of Gates*	*Number of Garbage output*
Existing method-1 [12,13]	21	5	4
Existing method-2 [14]	47	11	12
Proposed method (figure 4.1.4)	14	5	3

4.2. Design of Write Enabled Master-Slave D Flip-Flop

As data is both read from and written into RAM, each Flip-Flop should work on two modes: read and write. In this sub-section, a new write-enabled FF is presented, which is the modified version of the FF shown in Fig. 4.1.4. The irreversible FF is presented in Fig. 4.2.1 and its proposed reversible implementation is shown in Fig. 4.2.2. In the proposed one, a Fredkin gate is used to multiplex between FF's D input and stored bit Q in the FF. When 'write' is, high input D is to the Fredkin gate is carried to the first D latch and if 'write' is low output of the D-FF fed back to the input so state of the FF remains same. Fig. 4.2.3 delineates the block diagram of a reversible write-enabled master-slave D-FF which is used to construct the RRAM. A Write Enabled Master-slave D flip-flop we propose in Fig. 4.2.2 requires six reversible gates to design. It produces four garbage outputs and it has Quantum cost of 19.

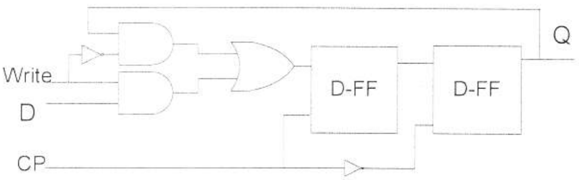


Fig. 4.2.1: Write enabled master-slave D FF

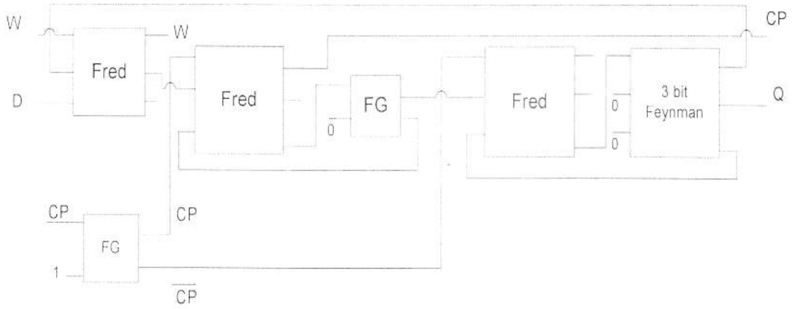


Fig. 4.2.2: Reversible Write-Enabled Master-Slave D-FF

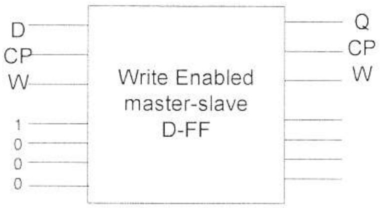


Fig. 4.2.3: Block diagram of Reversible Write-Enabled Master-Slave D-FF

4.3 Proposed Design of RRAM

In this section a reversible Random Access Memory is realized using the proposed reversible decoder and reversible write enabled master-slave D-FF. A RAM is a two dimensional array of flip-flops. There are 2^n rows while each row contains m flip-flops. Each time only one of the 2^n output lines of the decoder is active which selects one row of FFs of the RAM. Whether a read or a write operation is performed depends on the W input. When W is high, m FFs of the selected row of the RAM is written with the inputs D_1 to D_m. When W is low, Q_1 to Q_m contains stored bits in the FFs of the selected row and simultaneously the flip-flops are refreshed with the stored bits. Step by step design algorithm of a reversible RAM is given in Algorithm 4.3.1. Here a write enabled D flip-flop is denoted by FF, for simplicity only the significant inputs, D, CP and W are shown, other inputs, in order to make Flip-Flops reversible, should be given according to the Fig. 4.2.3. The circuit realization of Algorithm 4.3.1 is shown in Fig. 4.3.1.

Algorithm 4.3.1: (Designing the 2^n X m bit RAM)

```
Take one n X 2^n reversible decoder
For each row i = 1 to 2^n, in RAM
  Loop
     Take one Toffoli gate $T_i$
      If i = 1 then
             $T_i$ [ $I_1$ ]: = Write
     Else
             $T_i$ [ $I_1$ ] := $T_{i-1}$ [ $O_1$ ]
     End if
      $T_i$ [ $I_2$ ] := $i_{th}$ output of the decoder
      $T_i$ [ $I_3$ ] := 0
     For each j = 1 to m
       Loop
         Take one reversible D flip-flop $FF_j$
         $FF_j$ [D] := $D_j$, primary data input of the RAM
         If j = 1 then
              $FF_j$ [CP] := $T_j$ [ $O_2$ ]
              $FF_j$ [W] := $T_j$ [ $O_3$ ]
         Else
              $FF_j$ [CP] := CP output $FF_{j-1}$
              $FF_j$ [W] := W output $FF_{j-1}$
         End if
     End loop
  End loop
End
```

Theorem 4.3.1: *If GT_R be the number of gates required to realize a $2^n \times m$ Reversible RAM where n is the number of address bits and m be the selection bits in the RRAM, then*

$$GT_R \geq (7m + 2) \times 2^n - m$$

Proof: A $2^n \times m$ RRAM requires an n input 2^n output decoder and according to Lemma 3.1, 2^n reversible gates are required for this. Moreover 2^n Toffoli gates are required to perform AND operations in RRAM. $m \times 2^n$ FFs are required inside the m x 2^n RRAM whereas each FF required 6 reversible gates (Fig. 4.2.2). $(2^n - 1) \times m$ Feynman gates are required to perform the copy operation. If GT_R is considered as the minimum number of gates to realize the RRAM, so

$$GT_R \geq 2^n + 2^n + 2^n \times m \times 6 + (2^n - 1) \times m$$

$$So,\ GT_R \geq (7m + 2) \times 2^n - m$$

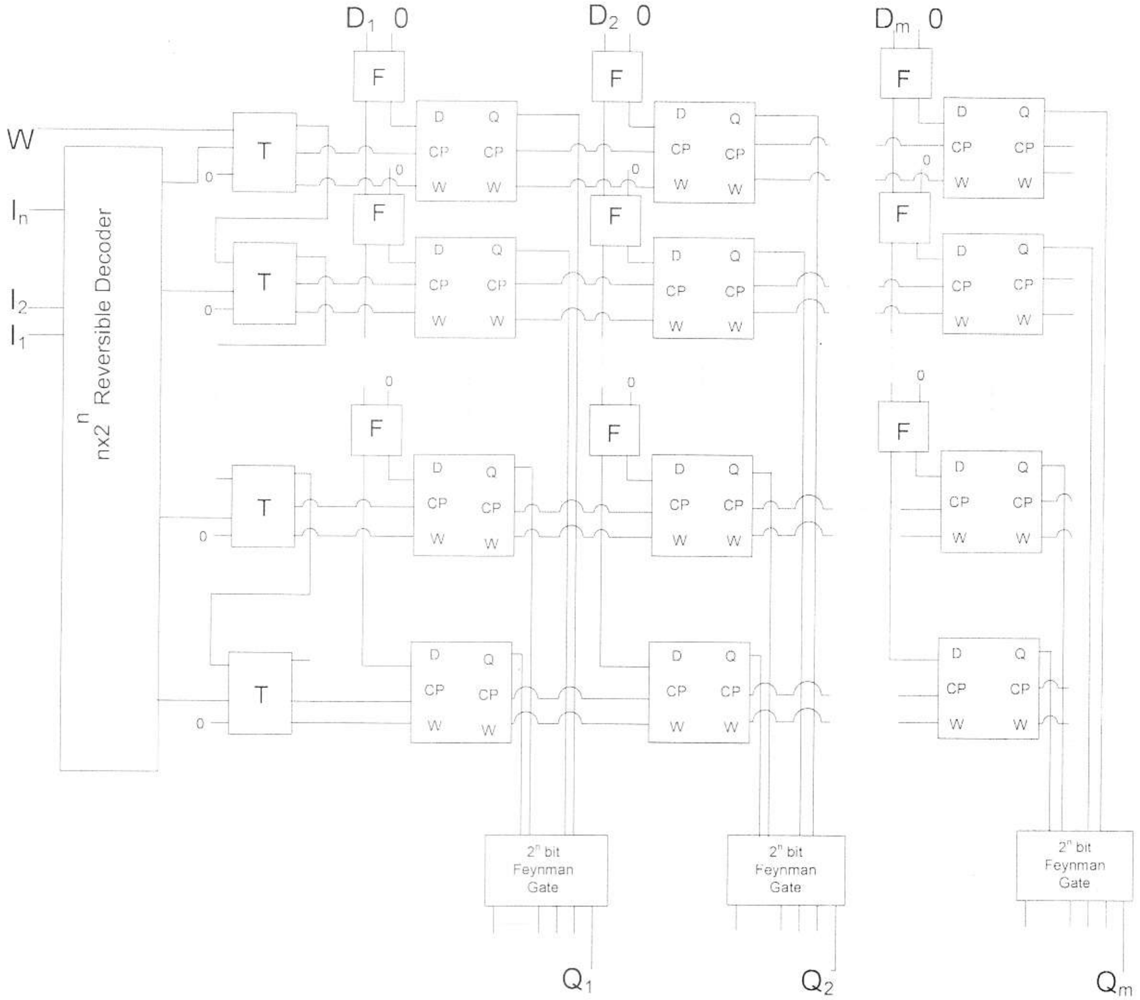


Fig. 4.3.1: Architecture of Reversible RAM

Theorem 4.3.2: *Let n be the number of address bits and m is the selection bits in the RRAM. If GB_R number of garbage outputs are generated from the RRAM, then*

$$GB_R \geq (5m + 2) \times 2^n + n - m.$$

Proof: A $2^n \times m$ RRAM requires one *n input* 2^n *output* decoder and according to Lemma 4.2, the reversible decoder produces only n garbage outputs. 2^{n+1} garbage bits are generate for last CP and Ws in m^{th} column. Each write-enabled D flip-flop produces 4 garbage outputs. Moreover $(2^n - 1)$ x m garbage outputs are generated from the 2^n Feynman Gates shown in Fig. 4.3.1.

If GB_R be the minimum number of garbage outputs generated from RRAM, then

$$GB_R \geq 2^n \times m \times 4 + (2^n - 1) \times x\, m + n + 2^{n+1}$$

$$\text{So, } GB_R \geq (5m + 2) \times 2^n + n - m.$$

Theorem 4.3.2: *Let n be the number of address bits and m is the selection bits in the RRAM. If Q_R* is the total Quantum *Cost of RRAM, then*

$$Q_R \geq (21 .m + 10) \times 2^n - 2m - 8$$

Proof: In decoder part, total number of three input Fredkin/NH gate required is 2^n -2, also 2 Feynman gates are required. In RRAM total of 2^n three input Toffoli gates are required. There are $2^n \times m$ reversible write-enabled D FFs, each contributes Quantum cost of 19. Also, $(2^n - 1) \times m$ two input Feynman and m 2^n input Feynman gates are required. Each 2^n input Feynman gate has Quantum cost of $2^n - 1$.

So total Quantum cost of the circuit at least is,

$$(2^n - 2) .5 + 2 + 2^n .5 + (2^n \times m) .19 + (2^n - 1) \times m + (2^n - 1) \times m$$

$$\text{So, } Q_R \geq (21 .m + 10) .2^n - 2m - 8$$

5. Conclusion

The main contribution of this paper is to design the Reversible Random Access Memory (RRAM). Moreover, we have proposed a new $n \times n$ reversible gate (NH Gate) and mostly used one, 3x3 family, is extensively discussed. We have also improved the sequential circuits, which are used to realize the RRAM, using the proposed NH gate. Efficiency of the proposed circuits is clearly highlighted here, whereas the lower bounds for the number of gates, number of garbage outputs and unit delay are established for the propounded RRAM. Since reversible RAM can be used in various reversible circuits for data storage such as Field Programmable Gate Array (FPGA), reversible central processing unit design and quantum computers etc so we believe that this design of reversible Random Access Memory may find its future use in numerous applications.

References

1. Keyes R, Landauer R., *Minimal Energy Dissipation in Logic*, IBM Journal of Research and Development 1970; 14: 153-7.
2. Charles H.Bennett, *Logical Reversibility of computation*, IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, 1973.
3. Shende VV, Prasad AK, Markov IL, Hayes JP., *Synthesis of reversible logic circuits,* IEEE Transaction on CAD 2003; 22(6): 723-9.
4. H.M.H. Babu, M.R.Islam, A.R.Chowdhury, S.M.A. Chowdhury, *Synthesis of full-adder circuit using reversible logic*, in: 17th International Conference on VLSI Design, 2004,pp.757-760.
5. Feynman R, *Quantum Mechanical Computers*, Optical News 1985:11-20.
6. Tommaso Toffoli, *Reversible Computing*, Automata, Languages and Programming, 7th Colloquium of Lecture Notes in Computer Science, vol. 85, pp. 632-644, 1980.
7. Edward Fredkin and Tommaso Toffoli, *Conservative Logic*, International Journal of Theoretical Physics, vol. 21, pp. 219-253, 1983.
8. A. Peres, *Reversible Logic and Quantum Computers*, Physical Review, vol. 32, pp. 3266-3276, 1985.
9. J. Smolin, D. P. DiVincenzo, *Five two-qubit gates are sufficient to implement the quantum Fredkin gate*, Physical Review A, Vol. 53, no. 4, April 1996, pp. 2855-2856.
10. M. Perkowski and et~al. *A hierarchical approach to computer-aided design of quantum circuits,* Preprint, 2002.
11. Tocci, Widmer. *Digital Systems, Principles and applications*, Prentice-Hall, 2000.
12. Himanshu Thapliyal and A.P Vinod, *Design of Reversible Sequential Elements With Feasibility of Transistor Implementation, Circuits and Systems*, 2007. ISCAS 2007. IEEE International Symposium.
13. Himanshu Thapliyal and Mark Zwolinski, *Reversible Logic to Cryptographic Hardware: A New Paradigm*, proceedings of the 49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2006)
14. Rice, J.E., *A new look at reversible memory elements, Circuits and Systems*, ISCAS 2006. Proceedings. 2006 IEEE International Symposium (May 2006).