

## An Efficient Design of Fault Tolerant Reversible Decoder Using Verilog HDL

**Mr. Vasam Adinarayana Swamy**

M-Tech Student,  
Department of VLSI & ES,  
Godavari Institute of Engineering and Technology,  
NH – 16, Chaitanya Knowledge City,  
Rajahmundry – 533296, india.

**Mr. G. V. Vinod**

Assistant Professor,  
Department of ECE,  
Godavari Institute of Engineering and Technology,  
NH – 16, Chaitanya Knowledge City,  
Rajahmundry – 533296, india.

### Abstract:

A binary decoder is a combinational logic circuit that converts a binary integer value to an associated pattern of output bits. They are used in a wide variety of applications, including data de-multiplexing, seven segment displays, and memory address decoding. This paper demonstrates the reversible logic synthesis for the n-to-2<sup>n</sup> decoder, where n is the number of data bits. The circuits are designed using only reversible fault tolerant Fredkin and Feynman double gates. Thus, the entire scheme inherently becomes fault tolerant. Algorithm for designing the generalized decoder has been presented. In addition, several lower bounds on the number of constant inputs, garbage outputs and quantum cost of the reversible fault tolerant decoder have been proposed. The gate simulations of the proposed decoder are shown using Xilinx 14.4 ISE, which proved the functional correctness of the proposed circuits. The comparative results show that the proposed design is much better in terms of quantum cost, delay, hardware complexity and has significantly better scalability than the existing approach.

### Index Terms:

Decoder, Delay, Garbage Output, Quantum Cost, Reversible & Fault Tolerant Computing.

### 1. INTRODUCTION:

The encoded input information need be preserved at the output in computational tasks pertaining to digital signal processing, communication, computer graphics, and cryptography applications. The conventional computing circuits are irreversible i.e. the input bits are lost when the output is generated. This information loss during computation culminates into increased power consumption. Reversible logic plays an extensively important role in low

power computing as it recovers from bit loss through unique mapping between input and output vectors [1]. No bit loss property of reversible circuitry results less power dissipation than the conventional one [2]. Moreover, it is viewed as a special case of quantum circuit as quantum evolution must be reversible [3]. Over the last two decades, reversible circuitry gained remarkable interests in the field of DNA-technology [4], nano-technology [5], optical computing [6], program debugging and testing [7], quantum dot cellular automata [8], and discrete event simulation [9] and in the development of highly efficient algorithms [10].

On the other hand, parity checking is a popular mechanisms for detecting single level fault. If the parity of the input data is maintained throughout the computation, then intermediate checking wouldn't be required and an entire circuit can preserve parity if its individual gate is parity preserving [11]. Reversible fault tolerant circuit based on reversible fault tolerant gates allows to detect faulty signal in the primary outputs of the circuit through parity checking [12].

Hardware of digital communication systems relies heavily on decoders as it retrieve information from the coded output. Decoders have also been used in the memory and I/O of micro processors [13]. In [7], a reversible fault tolerant decoder was designed, but it was not generalized and compact. Therefore, this paper investigates the generalized design methodologies of reversible fault tolerant decoders.

### 2. LITERATURE REVIEW:

This section formally defines reversible gate, garbage output, delay, hardware complexity and presents popular reversible fault tolerant gates along with their input-output specifications, transistor and quantum equivalent representations.

## 2.1. Reversible and Fault Tolerant Gates:

An  $n \times n$  reversible gate is a data stripe block that uniquely maps between input vector  $Iv = (I_0, I_1, \dots, I_{n-1})$  and output vector  $Ov = (O_0, O_1, \dots, O_{n-1})$  denoted as  $Iv \leftrightarrow Ov$ . Two prime requirements for the reversible logic circuit are as follows [14]:

- There should be equal number of inputs and outputs.
- There should be one-to-one correspondence between inputs and outputs for all possible input-output sequences.

A Fault tolerant gate is a reversible gate that constantly preserves same parity between input and output vectors. More specifically, an  $n \times n$  fault tolerant gate clarify the following property between the input and output vectors [12]:

$$I_0 \oplus I_1 \oplus \dots \oplus I_{n-1} = O_0 \oplus O_1 \oplus \dots \oplus O_{n-1} \quad (1)$$

Parity preserving property of Eq.1 allows to detect a faulty signal from the circuit's primary output. Researchers [11], [12], [15] have showed that the circuit consist of only reversible fault tolerant gates preserves parity and thus able to detect the faulty signal at its primary output.

## 2.2. Qubit and Quantum Cost:

The main difference between the qubits and conventional bits is that, qubits can form linear combination of states  $|0\rangle$  or  $|1\rangle$  called superposition, while the basic states  $|0\rangle$  or  $|1\rangle$  are an orthogonal basis of two-dimensional complex vector [3]. A superposition can be denoted as,  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , which means the probability of particle being measured in states 0 is  $|\alpha|^2$ , or results 1 with probability  $|\beta|^2$ , and of course  $|\alpha|^2 + |\beta|^2 = 1$  [16]. Thus, information stored by a qubit are different when given different  $\alpha$  and  $\beta$ . Because of such properties, qubits can perform certain calculations exponentially faster than conventional bits. This is one of the main motivation behind the quantum computing. Quantum computer demands its underneath circuitry be reversible [1][6]. The quantum cost for all  $1 \times 1$  and  $2 \times 2$  reversible gates are considered as 0 and 1, respectively [6] [14]. Hence, quantum cost of a reversible gate or circuit is the total number of  $2 \times 2$  quantum gate used in that reversible gate or circuit.

## 2.3. Delay, Garbage Output and Hardware Complexity:

The delay of a circuit is the delay of the critical path. The path with maximum number of gates from any input to any output is the critical path [1].

There may be more than one critical path in a circuit and it is an NP-complete problem to find all the critical paths [17]. So, researchers pick the path which is the most likely candidates for the critical paths [18]. Unused output of a reversible gate (or circuit) is known as garbage output, i.e., the output which are needed only to maintain the reversibility are the garbage output. The number of basic operations (Ex-OR, AND, NOT etc.) needed to realize the circuit is referred to as the hardware complexity of the circuit. Actually, a constant complexity is assumed for each basic operation of the circuit, such as,  $\alpha$  for Ex-OR,  $\beta$  for AND,  $\gamma$  for NOT etc. Then, total number of operations are calculated in terms of  $\alpha$ ,  $\beta$ , and  $\gamma$ .

## 2.4. Popular Reversible Fault Tolerant Gates:

1) Feynman Double Gate: Input vector ( $Iv$ ) and output vector ( $Ov$ ) for  $3 \times 3$  reversible Feynman double gate (F2G) is defined as follows [19]:  $Iv = (a, b, c)$  and  $Ov = (a, a \oplus b, a \oplus c)$ . Block diagram of F2G is shown in Fig.1(a). Fig.1(b) represent the quantum equivalent realization of F2G. From Fig.1(b) we find that it is realized with two  $2 \times 2$  Ex-OR gate, thus its quantum cost is two (Sec. 2.2).

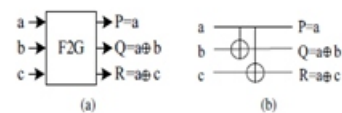


Fig.5.1: Reversible Feynman double gate (a) Block diagram (b) Quantum equivalent realization .

2) Fredkin Gate: The input and output vectors for  $3 \times 3$  Fredkin gate (FRG) are defined as follows [20]:  $Iv = (a, b, c)$  and  $Ov = (a, a \oplus b \oplus ac, a \oplus c \oplus ab)$ . Block diagram of FRG is shown in Fig.2 (a). Fig. 2(b) represents the quantum realization of FRG. In Fig.2 (b), each rectangle is equivalent to  $2 \times 2$  quantum primitives, therefore its quantum cost is considered as one [13]. Thus total quantum cost of FRG is 5.

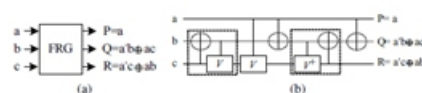


Fig.5.2: Reversible Fredkin gate (a) Block diagram (b) Quantum equivalent realization

Reversible Fredkin and Feynman double gate obey the rule of Eq.1. The fault tolerant (parity preserving) Property of Fredkin and Feynman double is shown in Table 1.

TABLE 1: Truth table for F2G and FRG:

Input			Output of F2G			Output of FRG			Parity
A	B	C	P	Q	R	P	Q	R	
0	0	0	0	0	0	0	0	0	Even
0	0	1	0	0	1	0	0	1	Odd
0	1	0	0	1	0	0	1	0	Odd
0	1	1	0	1	1	0	1	1	Even
1	0	0	1	1	1	1	0	0	Odd
1	0	1	1	1	0	1	1	0	Even
1	1	0	1	0	1	1	0	1	Even
1	1	1	1	0	0	1	1	1	Odd

2.5 Decoder:

Decoders are the collection of logic gates fixed up in a specific way such that, for an input combination, all outputs terms are low except one. These terms are the min-terms. Thus, when an input combination changes, two outputs will change. Let, there are n inputs, so number of outputs will be 2n. There are several designs of reversible decoders in the literature. To the best of our knowledge, the designs from [7] is the only reversible design that preserve parity too.

3. PROPOSED REVERSIBLE FAULT TOLERANT DECODER:

Considering the simplest case, n=1, we have a 1-to-2 decoder. Only a F 2G can work as 1-to-2 Reversible Fault tolerant Decoder (RFD) as shown in Fig.4 and from now on, we denote a reversible fault tolerant decoder as RFD.



Fig.4: Proposed 1-to-2 RFD Architecture

Fig.5(a) and Fig.5(d) represent the architecture of 2-to-4 and 3-to-8 RFD, respectively. From Fig.5(c), we find that 3-to-8 RFD is designed using 2-to-4 RFD, thus a schema of Fig.5(a) is created which is shown in Fig.5(b). Algorithm 1 presents the design procedure of the proposed n-to-2nRFD.

Primary input to the algorithm are control bits. Line 6 of the proposed algorithm assigns the input to the Feynman double gate for the first control bit (S0), whereas line 9 assigns first two inputs to the Fredkin gates for all the remaining control bits. Line 10-12 assign third input to the Fredkin gate for n=2

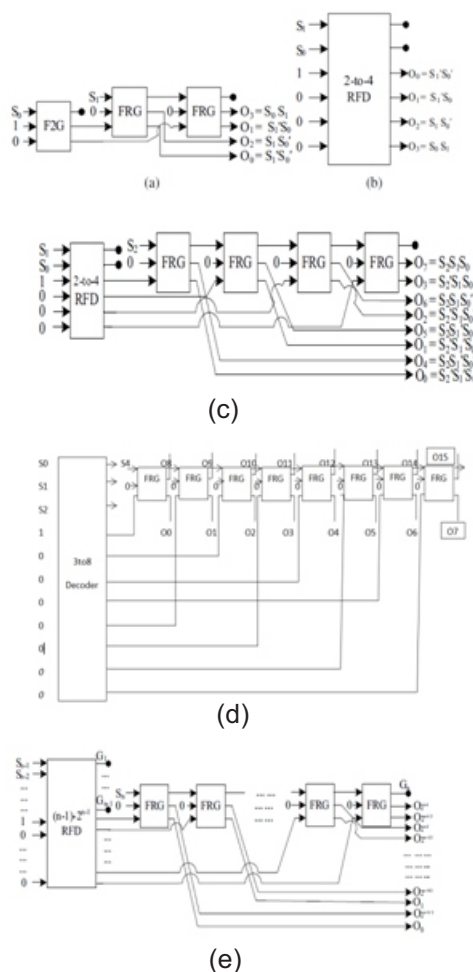


Fig.5.5: (a) Block diagram of the proposed 2-to-4 RFD. (b) Schematic diagram of 2-to-4 RFD. (c) Block diagram of the proposed 3-to-8 RFD. (d) Block diagram of the proposed 4-to-16 RFD. (e) Block diagram of the proposed n-to-2nRFD.

While line 13-15 assigns third input to the Fredkin gate through a recursive call to previous RFD for n>2. Line 18-19 returns outputs. The complexity of this algorithm is O(n). According to the proposed algorithm architecture of n-to-2nRFD is shown in Fig.7. we present the gate representations of FRG and F2G. These representations are finally used to get the gate circuits of the proposed decoder. Each of the proposed circuit are simulated Xilinx ISE simulator. This simulation also show the functional correctness of the proposed decoders. Table.2 shows a comparative study of the proposed fault tolerant decoders with existing fault tolerant one. Theorem 1: An n-to-2n reversible fault tolerant decoder can be realized with at least n garbage outputs and 2n constant inputs, where n is the number of data bits.



Proof: An  $n$ -to- $2n$  decoder has  $n$  inputs and  $2n$  outputs. Thus, to maintain the property of reversibility, there should be at least  $(2n - n)$  constant inputs. However, this  $(2n - n)$  constant inputs don't preserve the parity. To preserve the parity, at least  $n$  more constant inputs are needed. So, there should be at least  $n$  garbage outputs. Example 1: Let the value of  $n$  be 1. Then, we have the 1-to-2 reversible fault tolerant decoder. As shown in Sec.2, for a reversible circuit it is necessary to maintain the one-to-one correspondence between input and output vectors and thus, any reversible circuit should have equal number of inputs and outputs. In the 1-to-2 decoder, there are 2 primary outputs ( $O_0, O_1$ ) but 1 input ( $S_0$ ), hence according to the property of reversibility, 1-to-2 reversible decoder should have at least 1 constant input.

```

Algorithm 1: Algorithm for the proposed reversible fault
tolerant n-to-2^n decoder, RFD(S, F2G, FRG)
Input : Data input set S(S0, S1, ..., Sn-1)
        Feynman double gate (F2G) and Fredkin gate
        (FRG)
Output: n-to-2^n reversible fault tolerant decoder circuit
1 begin
2   i = input
3   o = output
4   for j ← 0 to n - 1 do
5     if j = 0 then
6       Sj → first.i.F2G, 1 → second.i.F2G, 0 →
        third.i.F2G
7     end if
8     else
9       Sj → first.i.FRG, 0 → second.i.FRG
10      if j=2 then
11        third.o.F2G → third.i.FRG
12      end if
13      else
14        call RFD(j-1), RFD.o.j → third.i.FRGj
15      end if
16    end if
17  end for
18  return FRG.o.3 & FRG.o.2 → desired output
19         remaining F2G.o & FRG.o → garbage
        outputs.
20 end
  
```

The value of this constant input can be either 0 or 1. Table.3 shows that whatever the value of this constant input, it will never be able to preserve the parity between input and output vectors, which is the prime requirement of the reversible fault tolerant logic circuit. Therefore, to preserve the parity for the 1-to-2 reversible fault tolerant decoder we need at least one more constant input, i.e., at least 2 constant inputs are required for the 1-to-2 reversible fault tolerant decoder.

**TABLE 5.2: Comparison of reversible fault tolerant decoders**

	GT	GO	QC	HC	UD
2-to-4 Existing Circuit [7]	3	2	15	$6\alpha + 12\beta + 6\gamma$	3
2-to-4 Proposed Circuit	3	2	12	$6\alpha + 8\beta + 4\gamma$	2
3-to-8 Existing Circuit [7]*	$\geq 7$	$\geq 3$	$\geq 35$	$\geq 14\alpha + 28\beta + 14\gamma$	$\geq 7$
3-to-8 Proposed Circuit	7	3	32	$14\alpha + 24\beta + 12\gamma$	4

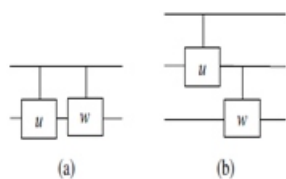
GT = No of Gate, GO = Garbage Output, QC = Quantum cost, HC = Hardware Complexity, UD = Unit Delay.

\* The design is not generalized one, i.e., it is not an  $n$ -to- $2n$  decoder.

**TABLE 3: 1-to-2 decoder with 1 constant input**

Inputs		Outputs		Parity
$CI$	$S_0$	$O_0$	$O_1$	
0	0	1	0	$I_p = E, O_p = 0$
0	1	0	1	$I_p = O_p$
1	0	1	0	$I_p = O_p$
1	1	0	1	$I_p = E, O_p = 0$

Next, we must prove the existence of combinational circuit which can realize the reversible fault tolerant 1-to-2 decoder by 2 constant inputs. This can easily be accomplished by the circuit shown in Fig.4(a). It can be verified that Fig.4(a) is reversible and fault tolerant with the help of its corresponding truth table, there is no need to give more detail. Now, in 1-to-2 reversible fault tolerant decoder there are at least 2 constant inputs and 1 primary input, i.e., total of 3 inputs. Thus, 1-to-2 reversible fault tolerant decoder should have at least 3 outputs, otherwise it will never comply with the properties of reversible parity preserving circuit. Among these 3 outputs, only 2 are primary outputs. So, remaining 1 output is the garbage output, which holds Theorem 1 for  $n=1$ . Theorem 2: A 2-to-4 reversible fault tolerant decoder can be realized with at least 12 quantum cost. Proof: A 2-to-4 decoder has 4 different  $2 \times 2$  logical AND operations. A reversible fault tolerant AND2 operation requires at least 3 quantum cost. So, 2-to-4 reversible fault tolerant decoder is realized with at least 12 quantum cost. Example 2: Fig.5(a) is the proof for the existence of 2-to-4 reversible decoder with 12 quantum cost. Next, we want to prove that it is not possible to realize a reversible fault tolerant 2-to-4 decoder fewer than 12 quantum cost. In the 2-to-4 decoder, there are 4 different  $2 \times 2$  logical AND operations, e.g.,  $S_1 \cdot S_0'$ ,  $S_1' \cdot S_0$ ,  $S_1 S_0'$ ,  $S_1' S_0$ . It will be enough if we prove that it is not possible to realize a reversible fault tolerant  $2 \times 2$  logical AND with fewer than three quantum cost. Consider, i. If we make use of one quantum cost to design the AND, that of course is not possible according to our discussion in Sec. 2. ii. If we make use of two quantum cost to design AND, then we must make use of two  $1 \times 1$  or  $2 \times 2$  gates. Apparently two  $1 \times 1$  gates can't generate the AND. Aiming at two  $2 \times 2$  gates, we have two combinations, which are shown in Fig.6 (a) and Fig.6 (b). In Fig.6 (a), the output must be (a, ab) if the inputs are (a, b). The corresponding truth table is shown in Table.4.



**Fig.6: Combinations of the two 2 × 2 quantum primitive gates**

**TABLE 4: Truth table of Fig. 6(a)**

From Table.4, we find that, outputs are not at all unique to its corresponding input combinations (1st and 2nd rows have the identical outputs for different input combinations). So it can't achieve the reversible AND. For Fig.6(b) if inputs are (a, b, c) then, the outputs of the lower level will be offered to the next level as a controlled input, this means that second output of Fig.6(b) have to be ab, otherwise it will never be able to get outputab since third output of Fig.6(b) is controlled by the second output, thereby according to Table 5, we can assert that the second combination is impossible to realize the AND no matter how we set the third output of Fig.6(b).

**TABLE 5: Truth table of Fig. 6(b)**

a	b	c	a	ab
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

(third column of Table.5), the input vectors will never be one-to-one correspondent with the output vectors. Therefore, we can conclude that, a combinational circuit for reversible fault tolerant 2×2 logical AND operation can't be realized with less than three quantum cost. The above example clarifies the lower bound in terms of quantum cost of 2-to-4 RFD. Similarly, it can be proved that the n-to-2n RFD can be realized with  $5(2n-8/5)$  quantum cost, when  $n \geq 1$ , and by assigning different values to n, the validity of this equation can be proved. Lemma 1: An n-to-2n RFD can be realized with  $(2n-1)$  reversible fault tolerant gates, where n is the number of data bits. Proof: According to our design procedure, an n-to-2n RFD requires an  $(n-1)$ -to- $2n-1$  RFD plus n number of Fredkin gates, which requires an  $(n-2)$ -to- $2n-2$  RFD plus  $(n-1)$  Fredkin gates and so on till we reach to 1-to-2 RFD. 1-to-2 RFD requires a reversible fault tolerant Feynman double gate only. Thus total number of gates required for an n-to-2n RFD is,

$$= \frac{1 + 2 + 4 + \dots + n^{th} \text{ term}}{(2-1)} = 2^n - 1$$

Example 3: From Fig.5(c) we find that the proposed 3-to-8 RFD requires total number of 7 reversible fault tolerant gates. If we replace n with 3 in Lemma 1, we get the value 7 as well.

Lemma 2: Let,  $\alpha, \beta, \gamma$  be the hardware complexity for a two-input Ex-OR, AND and NOT operation, respectively. Then an n-to-2n RFD can be realized with  $(2n+1-2)\alpha + (2n+2-8)\beta + (2n+1-4)\gamma$  hardware complexity, where n is the number of data bits.

Proof: In Lemma 1, we proved that an n-to-2nRFD is realized with a F2G and  $(2n-2)$  FRG. Hardware complexity of a FRG and a F2G are  $2\alpha+4\beta+2\gamma$  and  $2\alpha$ , respectively. Hence, hardware complexity for n-to-2nRFD is

$$= (2^n - 2)(2\alpha + 4\beta + 2\gamma) + 2\alpha$$

$$= (2^{n+1} - 2)\alpha + (2^{n+2} - 8)\beta + (2^{n+1} - 4)\gamma$$

Example 4: Fig.5(c) shows that the proposed 3-to-8 reversible fault tolerant decoder requires six Fredkin gates and one Feynman double gate. According to our previous discussion in Sec. 2, hardware complexity of a Feynman double gate is  $2\alpha$ , whereas, hardware complexity of a Fredkin gate is  $2\alpha + 4\beta + 2\gamma$ . Thus, the hardware complexity of Fig. 5(d) is  $6(2\alpha + 4\beta + 2\gamma) + 2\alpha = 14\alpha + 24\beta + 12\gamma$ . In Lemma 2, if we put  $n=3$ , we get exactly  $14\alpha + 24\beta + 12\gamma$  as well.

## 4.SIMULATION RESULTS AND ANALYSIS:

We have coded the all decoders in Verilog HDL using the proposed reversible design and the existing decoders designs of [6] and [7] for bit-widths 1,2, 3,4 and n bits. All the designs are synthesized in the Xilinx Synthesis Tool and Simulated using Xilinx ISE simulator. After synthesis and simulation, the area and delay values are compared with conventionallogicgates.

The synthesis result confirms that the proposed reversible decoders involves significantly less area and less delay and consumes less power than the existing designs.

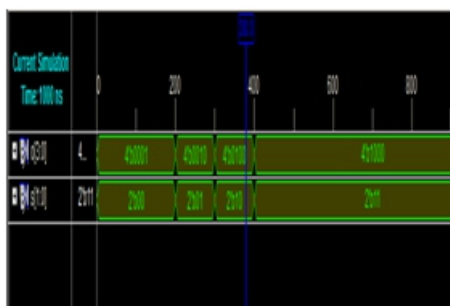


Fig.7 Simulation output for 2to4 reversible decoder

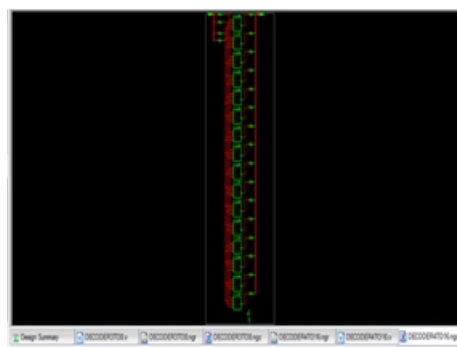


Fig.10 Technology schematic of 4to16 reversible decoder



Fig.8 Simulation output for 3to8 reversible decoder

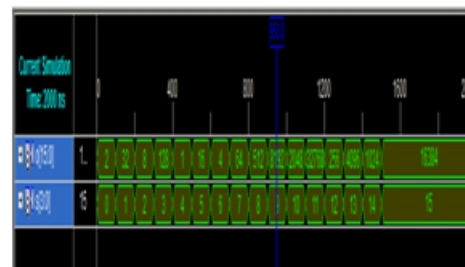


Fig.11 Simulation output for 4to16 reversible decoder

## 5.CONCLUSION:

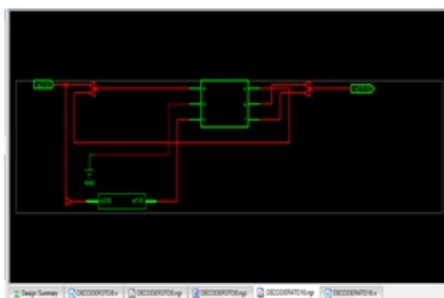
we presented the design methodologies of an 1-to-2,2-to-4,3-to-8,4-to-16 and n-to-2n reversible fault tolerant decoder, where n is the number of data bits. We proposed several lower bounds on the numbers of garbage outputs, constant inputs and quantum cost and proved that the proposed circuit has constructed with the optimum garbage outputs, constant inputs and quantum cost. In addition, we presented the designs of the individual gates of the decoder using reversible logic gates in order to implement the circuit of the decoder with gates. Simulations of the gate level implementation of the decoder showed that the proposed fault tolerant decoder works correctly. The comparative results proved that the proposed designs perform better than its counterpart. We also proved the efficiency and supremacy of the proposed scheme with several theoretical explanations. Proposed reversible fault tolerant decoders can be used in parallel circuits, multiple-symbol differential detection, network components and in digital signal processing etc.

## REFERENCES:

[1] L. Jamal, M. Shamsujjoha, and H. M. HasanBabu, "Design of optimal reversible carry look-ahead adder with optimal garbage and quantum cost," International Journal of Engineering and Technology, vol. 2, pp. 44-50, 2012.



(a)

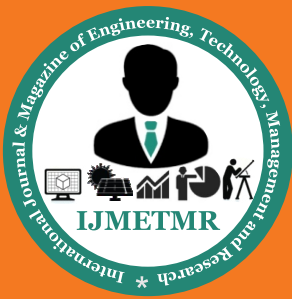


(b)

Fig.9 RTL Schematics of 4to16 reversible decoder(a) Top block(b)Internal block



- [2] C. H. Bennett, "Logical reversibility of computation," IBM J. Res. Dev., vol. 17, no. 6, pp. 525–532, Nov. 1973. [Online]. Available: <http://dx.doi.org/10.1147/rd.176.0525>
- [3] M. Nielsen and I. Chuang, Quantum computation and quantum information. New York, NY, USA: Cambridge University Press, 2000.
- [4] M. P. Frank, "The physical limits of computing," Computing in Science and Engg., vol. 4, no. 3, pp. 16–26, May 2002. [Online]. Available: <http://dx.doi.org/10.1109/5992.998637>
- [5] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. HasanBabu, "Efficient approaches for designing reversible binary coded decimal adders," Microelectron. J., vol. 39, no. 12, pp. 1693–1703, Dec. 2008. [Online]. Available: <http://dx.doi.org/10.1016/j.mejo.2008.04.003>
- [6] M. Perkowski, "Reversible computation for beginners," 2000, lectureseries, 2000, Portland state university. [Online]. Available: <http://www.ee.pdx.edu/mperkows>
- [7] S. N. Mahammad and K. Veezhinathan, "Constructing online testable circuits using reversible logic," IEEE Transactions on Instrumentation and Measurement, vol. 59, pp. 101–109, 2010.
- [8] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. A. Perkowski, "Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 25, no. 9, pp. 1652–1663, 2006.
- [9] D. Maslov, G. W. Dueck, and N. Scott, "Reversible logic synthesis benchmarks page," 2005. [Online]. Available: <http://webhome.cs.uvic.ca/dmaslov>
- [10] D. M. Miller, D. Maslov, and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," in Proceedings of the 40th annual Design Automation Conference, ser. DAC '03. New York, NY, USA: ACM, 2003, pp. 318–323. [Online]. Available: <http://doi.acm.org/10.1145/775832.775915>.
- [11] R. K. James, T. K. Shahana, K. P. Jacob, and S. Sasi, "Fault tolerant error coding and detection using reversible gates," IEEE TENCON, 2007, 2007.
- [12] B. Parhami, "Fault tolerant reversible circuits," Proc. of 40th AsimolarConf. Signals, Systems, and Computers. Pacific Grove, CA, pp. 1726–1729, 2006.
- [13] F. Sharmin, M. M. A. Polash, M. Shamsujjoha, L. Jamal, and H. M. HasanBabu, "Design of a compact reversible random access memory," in 4th IEEE International Conference on Computer Science and Information Technology, vol. 10, June 2011, pp. 103–107.
- [14] H. M. HasanBabu, R. Islam, A. R. Chowdhury, and S. M. A. Chowdhury, "Reversible logic synthesis for minimization of full-adder circuit," Digital Systems Design, Euro-micro Symposium on, pp. 50–53, 2003. [Online]. Available: <http://dl.acm.org/citation.cfm?id=942792.943112>
- [15] J. Mathew, J. Singh, A. A. Taleb, and D. K. Pradhan, "Fault tolerant reversible finite field arithmetic circuits," in Proceedings of the 2008 14th IEEE International On-Line Testing Symposium, ser. IOLTS '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 188–189. [Online]. Available: <http://dx.doi.org/10.1109/IOLTS.2008.35>
- [16] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM J. Comput., vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: <http://dx.doi.org/10.1137/S0097539795293172>
- [17] M. R. Garey and D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co., 1990.
- [18] L. Singhal, E. Bozorgzadeh, and D. Eppstein, "Interconnect criticality driven delay relaxation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, pp. 1803–1817, October 2007.
- [19] R. Feynman, "Quantum mechanical computers," Foundations of Physics, vol. 16, pp. 507–531, 1986, 10.1007/BF01886518. [Online]. Available: <http://dx.doi.org/10.1007/BF01886518>
- [20] E. Fredkin and T. Toffoli, "Conservative logic," International Journal of Theoretical Physics, vol. 21, no. 3, pp. 219–253, Apr. 1982. [Online]. Available: <http://dx.doi.org/10.1007/BF01857727>



[21] DSCH:, "Microwind and dsch information page."  
[Online]. Available: <http://www.microwind.org/>

[22] R. Cleve and J. Watrous, "Fast parallel circuits for the quantum fourier transform," in Proceedings of the 41st Annual Symposium on Foundations of Computer Science, ser. FOCS '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 526–. [Online]. Available: <http://dl.acm.org/citation.cfm?id=795666.796591>

[23] H. M. H. Babu, M. I. Zaber, M. M. Rahman, and M. R. Islam, "Implementation of multiple-valued flip-flops using pass transistor logic," in Euromicro Symposium on DSD, Architectures, Methods and Tools, France. IEEE Computer Society, 2004, pp. 603–606.

[24] A. Schenk, R. F. H. Fischer, and L. Lampe, "A stopping radius for the sphere decoder and its application to msdd of dpsk," *Comm. Letters.*, vol. 13, no. 7, pp. 465–467, Jul. 2009. [Online]. Available: <http://dx.doi.org/10.1109/LCOMM.2009.090940>

[25] M. Chrzanowska-Jeske, "Architecture and synthesis issues in FPGAs," in Proceedings of NORTHCON'93 Electrical and Electronics Convention, ISBN: 0-7803-9972-2, 1993, pp. 102 – 105.

[26] Voyiatzis, D. Gizopoulos, and A. Paschalis, "Accumulator-based test generation for robust sequential fault testing in DSP cores in near optimal time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, pp. 1079–1086, 2005.