iJET EXPLORE International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 2 No. 4, April, 2012

On the Realization of Online Ternary Testable Circuit

Naushin Nower[1], Ahsan Raja Chowdhury[2]
[1]Institute of Information Technology, University of Dhaka,
Dhaka, Bangladesh
[2] Department of Computer Science & Engineering, University of Dhaka,
Dhaka, Bangladesh

ABSTRACT

Ternary Reversible logic got the attention in the recent years for its applications in different sections of Reversible Logic Synthesis. Designing online testable circuits are also considered as the prominent field of research in this domain. This paper presents a novel idea of Online Testable Ternary Reversible Circuits design, where architecture is capable of testing reversible ternary network in real-time (online). The error detection unit, a component of the proposed design, is also designed with reversible gates, for which the entire circuit is considered reversible. The evaluation of the proposed design is carried out with example circuits in terms of well-know design parameters to show the effectiveness and compactness of the circuit.

Keywords: *Garbage output, Minterm, Ternary online testing.*

1. INTRODUCTION

Landauer [1, 2] proves that, logic computations that are not reversible, generate heat *kT*ln2* for every bits of information that is lost, where *k* is the Boltzmann's constant and *T* is the temperature [2]. A **reversible** logic gate is a *k*-input, *k*-output (denoted *k*k*) device that maps each possible input pattern into a unique output pattern [3]. Bennett showed that *kTln2* energy dissipation would not occur, if a computation were carried out in a reversible way [1]. Reversible computation in a system can be performed if the system is composed of reversible gates.

Testing the reversible circuits is a major dispute in today's research because the levels of logic are significantly higher than the standard logic [3]. Online testability is a feature of a circuit that pledges the testing of the circuit at the time the computation is performed. Our proposed work concentrates on the design of ternary testable block using conventional ternary reversible gates. The proposed block is versatile and extendable as it can be used to realize any arbitrary sized **Ternary ESOP** expression. Good number of research works have been performed recently [4-8], focus on binary reversible circuit testing. However, only few of them [9- 10] focus on the ternary circuit testing.

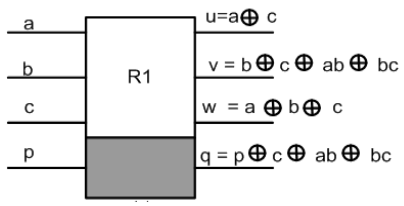
Rest of the paper is organized as follows: Section 2 provides some Basic Definition and Literature Review regarding Ternary Reversible Logic and Online Testing. The proposed idea is introduced in Section 3 where evaluation of the proposed design technique is thoroughly discussed with examples in Section 4. The paper concludes with observations and suggestions for future study in Section 5.

2. LITERATURE REVIEW

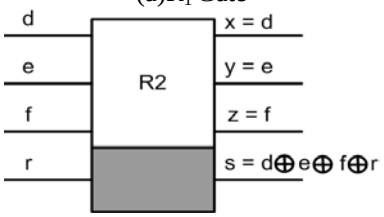
In this section we discuss some terms and definitions used in this paper along with some existing methods.

Definition 1: Online Testability is the ability of a circuit to test a reversible block at the time the circuit is performing an operation. It is the unique feature of reversible circuit first proposed in [3].

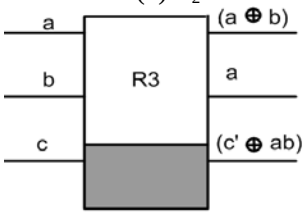
The authors of [3] proposed three reversible logic gates: R1, R2 and R3 (shown in Figure 1), where R1 and R2 can be used in pairs to design testable reversible logic circuits, and R3 that is used to construct two pair two rail checker [8].



(a)R_1 Gate



(b) R_2 Gate



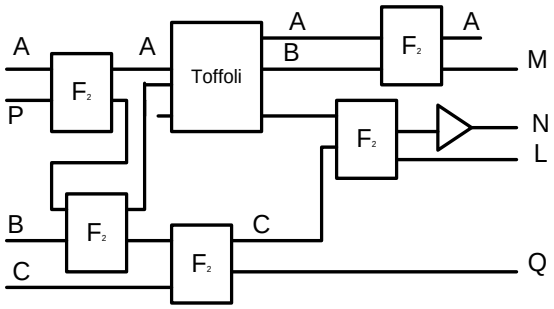
(c) R_3 Gate

Figure 1: Online Testable Reversible Logic Gates [3]

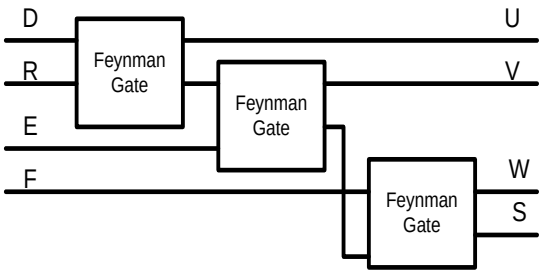
The first gate R1 is used for implementing arbitrary functions while the second gate R2 is employed to incorporate on-line testability features into the circuit. It can be verified that the input pattern of these gates can be uniquely determined from the particular output pattern. Applications of testable block [3] are sketched in [8] on ripple carry adder and carry skip adders. They have shown the implementation of their testable block [3] on the subset of MCNC benchmark circuit. But the designs do not have any implication on multi valued logic synthesis. Since Ternary logic synthesis provides a new-fangled era at present, the necessity for the ternary online testing is a challenging intention for today's research.

In the literature, ternary testable circuits are found in [9-10], where the authors of [9] proposed two gates (TR1 and TR2), which are arranged in cascading fashion to construct Testable Ternary Reversible Block (TR). TR1 is used to generate any ternary expression, while TR2 contains testability features. These two gates are used to build ternary expression and then the outputs from TR1 and TR2 are provided into a rail checker circuit to check the entire circuit. In [10], the modified TR1 of [9] is sketched where the gate cost for construction of TR1 is reduced from 92 MS gate to 49 MS gate. The proposed circuit is able to detect single bit error only where multiple bits failure may be occur in reversible circuits. Moreover, no generalized methods are introduced in [9, 10] for calculating the cost of a ternary expression. In our proposed architecture, we have shown more efficient design than the existing. The proposed TR1 and TR2 [9, 10] are shown in Figure 2 and two pair rail checker is depicted in Figure 3.

Definition 2: Ternary Quantum Logic is the simplest introduction of multi-valued logic which is also referred to as 3VL .To define ternary logic, let T = {0, 1, 2}. A ternary reversible logic circuit with n inputs and n outputs is also called an n-qudit ternary reversible gate can operate on ternary values.



(a) TR_1 Block



(b) TR_2 Block

Figure 2:Two Reversible Online Testable Blocks [9, 10]

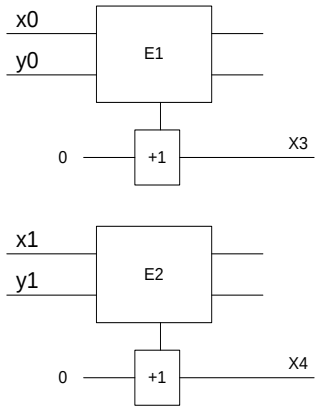


Figure 3: Two pair rail checker [9, 10]

Definition 3: Ternary Galois Field Logic (TGF) [11] consists of the set of elements $T = \{0, 1, 2\}$ and two basic binary operations – ***addition*** (denoted by +) and ***multiplication*** (denoted by · or absence of any operator). These two operations are shown in Table 1.

Table 1: Ternary Galois Field (GF3) operations

(a)Addition

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

(b) Multiplication

•	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

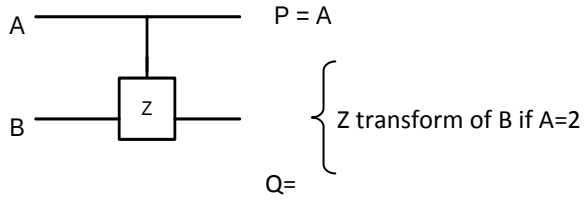


Figure 4: Symbol of ternary M-S gate

Definition 5: Ternary Feynman Gate, counterpart of the Binary Feynman gate, is shown in Figure 3(a) where the outputs of the gate are $Y_1 = X_1$ and $Y_2 = X_1 \oplus X_2$. When $X_1 = 0$, none of the controlling values of the M–S gates will be 2 and no transformation will be applied on X_2. Therefore, Y_2 will be $0 \oplus X_2 = X_1 \oplus X_2$. If $X_1 = 1$, then the controlling value of only the second M–S gate will be 2 and Y_2 will be $X_2 \oplus 1 = X_1 \oplus X_2$. If $X_1 = 2$, then the controlling value of only the first M–S gate will be 2 and Y_2 will be $X_2 \oplus 2 = X_1 \oplus X_2$. The cost of ternary Feynman gate implementation is shown in Figure 5(a) and the copy operation [10] is shown in Figure 5(b) [12].

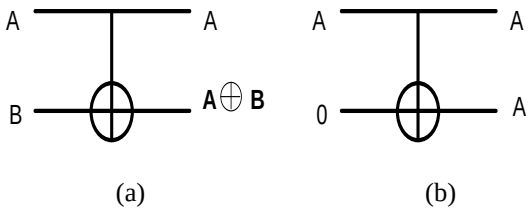


Figure 5: (a) Feynman Gate (b) Copy operation in Feynman Gate

Definition 6: A 3-qutrit Ternary Toffoli Gate is shown in Figure 6(a), where X_1 and X_2 are two controlling inputs and X_3 is the controlled input. If the two controlling input values are 2, then Z transform is applied on the controlled input, otherwise the controlled input is passed unchanged. That is, the outputs of the gate are $Y_1 = X_1$, $Y_2 = X_2$, and Y_3 = Z transform of X_3 if $X_1 = 2 \wedge X_2 = 2$, where Z € {+1, +2, 12, 01, 02}; $Y_3 = X_3$ otherwise. 3-Input Toffoli gate can be act as multiplier by setting third input as a 0 as given in Figure 6(b).

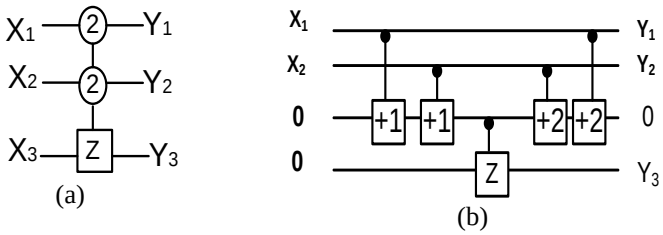


Figure 6. (a) Ternary Toffoli Gate (b) Toffoli gate as a multiplier

Definition 7: Garbage Output indicates the output that is not used as input to other gate or as a primary output. The unutilized outputs from a gate are called "garbage". Heavy price is paid off for every garbage output. Figure 7 shows the garbage output produced in Feynman gate if we want to produce Exclusive OR operation. One extra output should be produced to make the circuit reversible and that unwanted output (P=A, marked as *) is known as garbage.

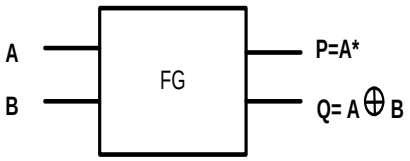


Figure 7: The garbage output A*

Before going to the details of our design we have discussed some recent related work on online testing in this section. In the literature we have found only [9, 10] that incorporates on ternary testing. Besides literature review, some cost effective ternary gate that we used in our design are discussed.

3. ONLINE TESTABLE REVERSIBLE TERNARY CIRCUIT: THE PROPOSED DESIGN

This section introduces the proposed design for online testability of reversible ternary circuits. We present a testable block that is able to test any ternary expression in online. The proposed Ternary Testable Block (TB_3) is implemented using ternary reversible gates, which eventually make the entire circuit reversible. The block, shown in Figure 8, is designed very intelligently so that the output will be 1 if the circuit works correctly.

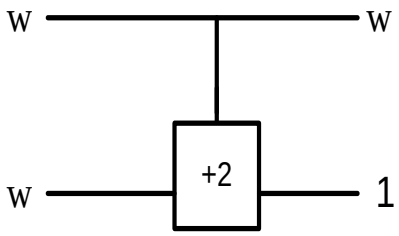


Figure 8: M-S gate with same input

On the way to design the TB_3, we have applied some GF3 [11] operations on each testable minterm in the expression. The required logic operation is illustrated in Table 2. The circuit tests each minterm and the result of the testing propagates to the next minterm checking block. The block is designed in such a way that if every minterm is correct and the entire expression is accurate the final testable block must produce 1.

Table 2: Logic behind the proposed TB_3

x	*y = x*	*x.y*	*w=x.y+2*
0	1	0	2
1	2	2	1
2	0	0	2

**x* denotes a minterm (such as *ab*)

Since we examine ternary logic, the value of the minterm must be in 0, 1 or 2. In the testable block, 1 is added with each minterm by applying *z* transform on it, which results (minterm +1). Then we perform (minterm) * (minterm +1), where the result is either 0 or 2. Then w is produced by applying z transform on the produced result. The value of *w* is limited to 1or 2. The produced resultant (w) is applied on both input of the M-S gate, as shown in Figure 8. Then the output of the M-S gate must be 1, if all the operations are performed correctly. The producing 1 propagates to the next minterm checking block. This testable block multiplies produced 1 with another 1 getting as an input from the previous block. Gate level diagram of the Reversible Ternary Online Testable Function is shown in Figure 9, whereas, block level one is shown Figure 10.

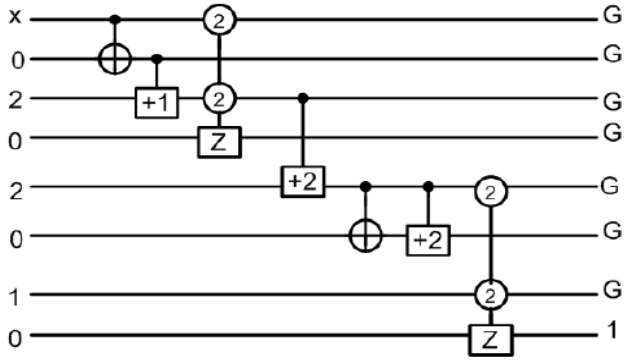


Figure 9: Ternary Testable Block

Now, we are going to develop the entire testable circuit for arbitrary ternary ESOP expressions. Lets' consider a ternary expression x = $ab \oplus cd$ which should be generated first. While generating x, verification should also be performed over the newly created circuit. We need three TB3 to verify ($ab \oplus cd$), two for the minters *ab* and *cd*, and the other one for or generating $ab \oplus cd$ .The first minterm *ab* can generate through Toffoli gate by setting third input 0. Then Feynman gate is used to generate the duplicate of ab. One copies of *ab* is put in to the testable block for evaluation.

Another copy is placed in to the Feynman gate for producing Exclusive OR with cd. The testable block for *ab* performs the logic operation on the *ab* as described in the previous section. The TB_3 produces 1 for *ab* if all operations are acceptable. The same operation performs on cd and the TB3 also produces 1 for *cd*. The 1 resulting from testable block for *ab* participates

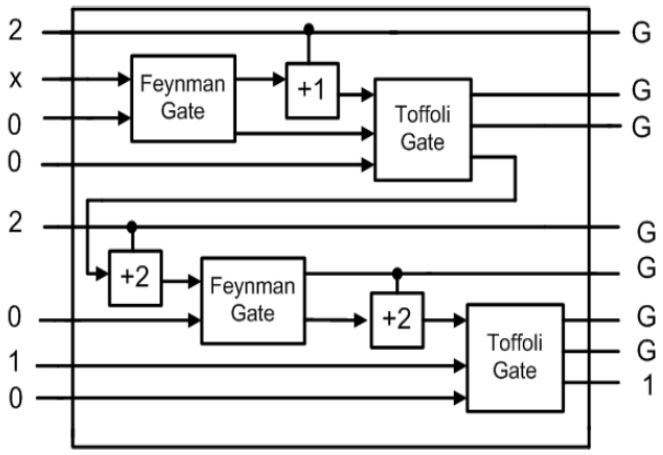


Figure 10: Ternary Testable Block (gates are represented as block)

as an input in the testable for *cd*. The copy of *ab* and *cd* are put into the Feynman gate for producing $ab \oplus cd$. The result is also copied by Feynman gate and one replica is act as an output and another is placed on the testable block for verifying. This TB3 produces 1 for checking $ab \oplus cd$ and getting another 1 from previous the testable block. Their multiplication generates 1. Testable circuit is shown in Figure 11 and hence, the generalized architecture for any arbitrary ternary ESOP function is shown in Figure 12.

4. EVALUATION OF THE PROPOSED BLOCK

To evaluate the proposed block we have calculated both garbage and gate cost. The following theorems for cost calculation can be applicable for any ternary expression.

Theorem 1: Let n be the number of minterm in a ternary expression, g_i be the garbage produced for i^{th} minterm, where $1 \le i \le n$, l and T denotes the number of literals and Toffoli expression, then the total number of garbage T_{GB} produced in the TB_3

$$T_{GB} = \sum_{i=1}^{n} g_i + 15n + 8(l\text{-}T\text{-}1)$$

Proof: Suppose, for i^{th} minterm, g_i garbage is generated. If there are k literals in the minterm, then according to the design $g_i (=k)$ garbage output is produced.
Next we calculate garbage for minterm checking. The TB_3 is 8* 8 blocks where 1 is the actual output and rest of the 7 are garbage output. According to the proposed design, each block produces 7 garbage outputs, and hence the total number of garbage produced for generation and verification (for each minterm) is $\sum_{i=1}^{n} (g_i+7)$. For Ex-or operations, 2 input Feynman gate is required, which generates 1 garbage output only. Another Feynman gate is necessary for copying the output. One output acts as a result and other one is placed in TB_3 for observation. Thus, 8 garbage outputs are produced in x-or. To produce x-or operation of 2 minterms, 8*(2-1) garbage is formed, that is, for n minterms, 8(n-1) garbage outputs are produces for x-or operation. Thus the number of garbage outputs required to Ternary Online Testable Block is for minterm:

$$\sum_{i=1}^{n} (g_i + 7) + 8(n\text{-}1) = \sum_{i=1}^{n} g_i + 7n + 8n - 8$$

$$= \sum_{i=1}^{n} g_i + 15n - 8$$

For x-or operation between two literals, 8(2-1) garbage output is produced. So for l literals number of produced garbage is 8(l-1). Moreover, garbage output for ExOR operation between all minterms and all literals is 8.
To minimize the gate cost we can use 1 Toffoli gate for expression like T= minterm x-or literal. Instead we calculate 2 gates (Toffoli and Feynman) for the same expression. By using 1 Toffoli gate the number of garbage is also reduced, so we need to subtract 8T from the garbage cost.

Thus total number garbage produced in total is

$$\sum_{i=1}^{n} g_i + 15n - 8 + 8(l\text{-}1) + 8 - 8T$$

$$= \sum_{i=1}^{n} g_i + 15n + 8(l\text{-}T\text{-}1).$$

Example 4.1 Lets consider the ternary expression $abc \oplus def \oplus ghi$. The testable circuit is shown in Figure 13.

The minterm *abc* is produced by 4- input Toffoli gate. Since there are 3 literals in each minterm then g_i=3 is created for each minterm. To validate the replica of *abc* is placed in TB_3 where 7 garbage is produced. Thus total 3+7=10 garbage is produced for minterm *abc* generation and validation. There are 3 minterms in the expression so the cost for minterm is 3* 10=30.

Next, $abc \oplus def$ is generated with 1+7 (*1* for Ex-OR and 7 for testing). Then *ghi* is x-ored with the previous result and again the garbage is (1+7=8).The total garbage is 30 + 8 + 8=46.

According to Theorem 1, n=3, k=3, l=0 and T=0 so the total numbers of garbage outputs are:

$$\sum_{1}^{3} g_3 + 15*3 - 8 \quad [\text{where } g_i = 3 \text{ garbage}]$$

$$= \sum_{1}^{3} 3 + 45 - 8$$

$$= 3*3 + 45 - 8 = 46$$

which conforms the number of garbage outputs in Figure 13.

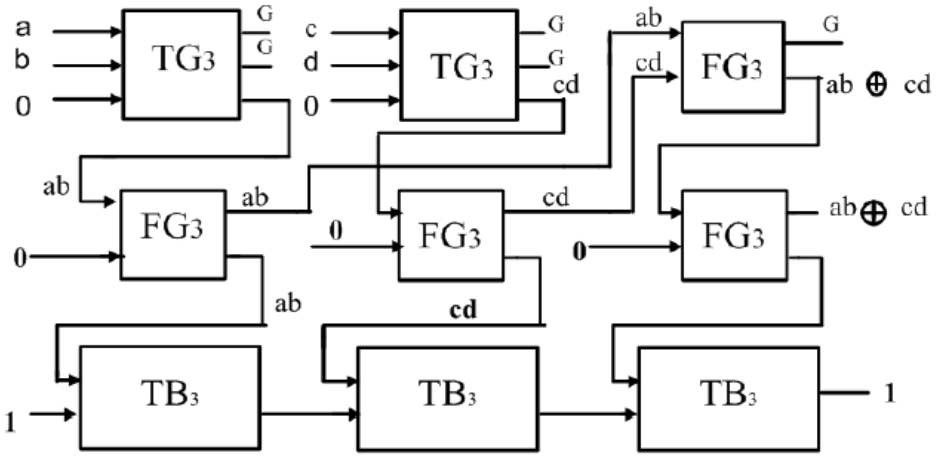


Figure 11: Generation of *ab* ⊕ *cd* with testable block

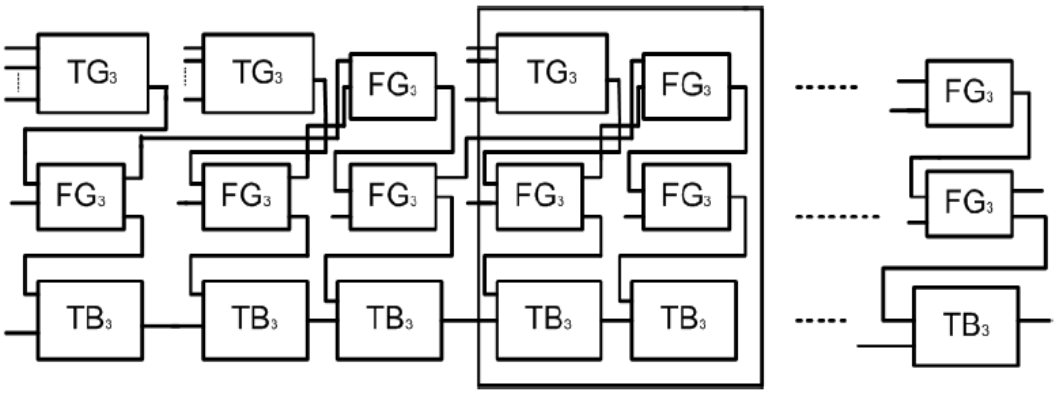


Figure 12: Architecture of Testable Circuit for any arbitrary Ternary ESOP functions

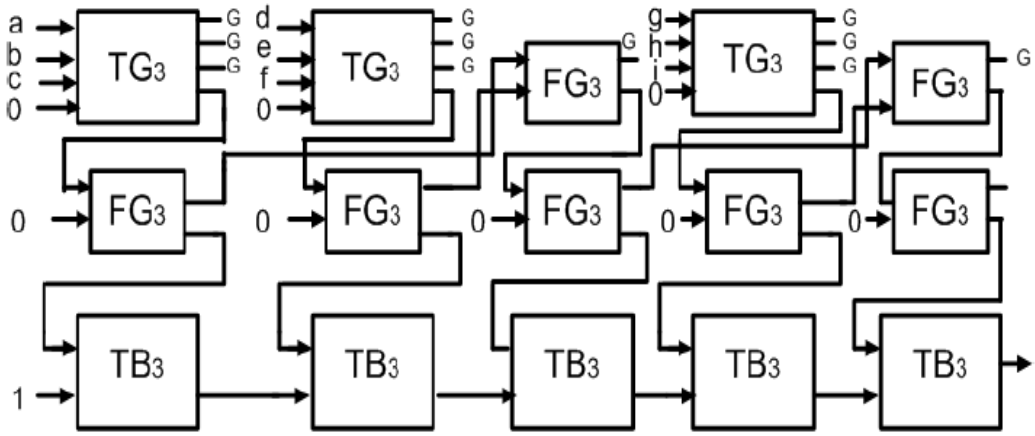


Figure 13: Testable Circuit Architecture of $abc \oplus def \oplus ghi$

Theorem 2: Let n be the number of minterm, l indicates the number of literal, T denotes the number of Toffoli expression. If N_G is the total number of gates required to generation ternary expression then

$$N_G = 9\{(l+2n)-(T+1)\}$$

Proof: To perform x-or operation and verification of 2 literals 9 gates are required. (1 Feynman gate for x-or one for copy and 7 gates in TB_3). Thus for l literals $9(l\text{-}1)$ gates are needed. To generate and verify each minterm we require 9 = (2+7) gates. (2 for generation and 7 for testing) Thus $9n$ gates are intended for n minterm. For Ex-OR operation and verification between 2 minterms 9 gates are required. For 3 minterm 9*(3-1) gates are required and thus for n minterm $9*(n-1)$gates are indented for Ex-OR operation. So, the total required gates for minterm $=9n+ 9*(n-1)= 9n=\mathbf{18n\text{-}9}$.

For performing Ex-OR with literals and minterm we need another **9** gates. (like $a \oplus b \oplus cds$). To minimize the gate cost we can use 1 Toffoli gate for expression like T= minterm x-or literal. Instead we calculate 2 gates (Toffoli and Feynman) for the same expression which requires $18T$ gates. By using 1 Toffoli gate the gates cost is $9T$ so we subtract extra 9T from the total number of gates. Hence, the total number of required gates is:

$$N_G = 9(l-1) + 18n\text{-}9+9\text{-}9T = 9l - 9 + 18n - 9+9\text{-}9T$$
$$= 9l + 18n - 9\text{-}9T = 9\{(l+2n)-(T+1)\}$$

Example 4.2: Let's consider the Ternary expression $abc \oplus def \oplus ghi$. We need 9 gates for abc generation and for verification. So, number of gates for 3 minterm generation and verifying is 3*9=27. Next we calculate the number of gates required for Ex-OR operation. The requirement is 9*(3-1) =18 and hence, the final value becomes: N_G=27+18=45.

According to the theorem n=3, since there are no literals so l=0 and also T=0. So total number of required gates

$$N_G = 9l + 18n - 9\text{-}9T = 9*0 + 18*3 - 9\text{-}9*0 = 54\text{-}9=45$$

5. EXPERIMENTAL RESULTS

Our proposed design for realization and testing of ternary expression have been written using language C and tested extensively on windows workstation. Several experimental results for some of the benchmark circuits are given below in Table 3 using an Intel Pentium II Desktop CPU 300 MHz under Microsoft Windows 98 edition with 128 MB RAM. During the execution it was ensured that no other application is running.

Table 3 shows that, the comparison between the existing testable with that of proposed circuit in terms of gate cost. The number of gates required to realize the benchmark functions are reasonably less than the required number of gates in the exiting method [10]. Though the garbage outputs are little bit higher for the proposed method, more research of the combination of connections among the gates can significantly reduce the total garbage outputs from the designed circuits, and we are working towards achieving the goal.

Table 3: Gate and Garbage count on Proposed Design for various Benchmark Circuits

Ternary Benchmark Circuit	Inputs	Outputs	Gate Cost [10]	Gate Cost [Proposed]
XOR5	5	1	46	36
Mod5	5	1	82	72
6sym	10	1	238	180

We are also working on the realization of larger circuit as the proposed algorithm shows excellent performance for small and medium sized benchmark circuits. The comparison between the existing and proposed one, in terms of number of gate is shown in a chart (Figure 14), to illustrate the consistency of the proposed method.

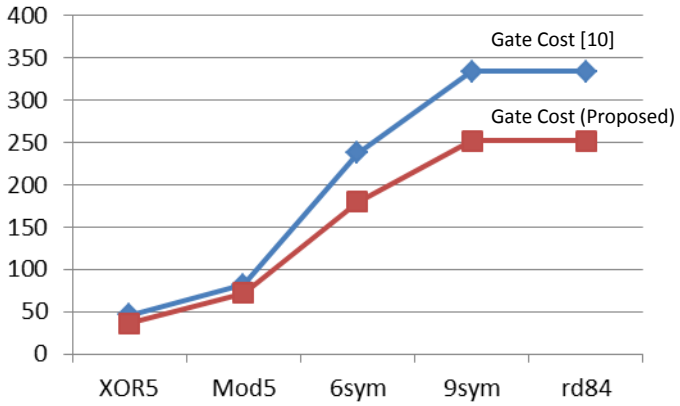


Figure 14: Comparison between Existing and proposed method in terms of Gate Cost

6. CONCLUSIONS

In this paper we introduce a new concept on Ternary Reversible Online Testable Circuit. We have successfully

developed the ternary testable block for online testing which is tested with various expressions. The proposed testable block is designed in such a way that it can be used to test any online ternary circuit. The theoretical underpinnings for the proposed method are also established by various Theorems. Experimental observation shows that the proposed circuit obtains better result in terms of gate cost for benchmark ternary expression.

REFERENCES

[1] Bennett. C. H. 1973, "Logical Reversibility of Computation", IBM J. Research and Development, pp. 525- 532.

[2] Landauer. R. 1961, "Irreversibility and Heat Generation in the Computing Process", IBM J. Research and Development, vol. 3, pp. 183-191.

[3] Vasudevan. D.P., Lala. P.K. and Parkerson. J.P. 2004, "Online Testable Reversible Logic Circuit Design using NAND Blocks", Proc, Symposium on Defect and Fault Tolerance, pp-324-331.

[4] Tara. N., and Chowdhury. A. R., "Design and Analysis of Error Detection Module for Reversible On-Line Testable Circuits**",** Silver Jubilee Conference on Communications and VLSI Design Vellore Institute of Technology (VIT), Vellore, India, Oct 8-10, 2009, pp. 266-267.

[5] Mitra. S. M., Hossain. M. F., Anwar. S., and Chowdhury. A.R., "Online Testable Fault Tolerant Full Adder in Reversible Logic Synthesis", Second International Conference on Signals, Systems & Automation (ICSSA-11), pp. 508-512, January 24-25, 2011, Gujarat, India.

[6] Mitra. S. M., Sultana. T., Anwar. S., and Chowdhury. A. R , "Efficient Approach to design Reversible Fault Tolerant Cyclic Redundancy Check Circuit**",** Second International Conference on Signals, Systems & Automation (ICSSA-11), pp. 503-507, January 24-25, 2011, Gujarat, India.

[7] Mitra. S. M., Anwar. S., and Chowdhury., "Efficient Design of Check Circuit to detect Multiple Cell Errors in Reversible Logic Synthesis", Second International Conference on Signals, Systems & Automation (ICSSA-11), pp. 304-308, January 24-25, 2011, Gujarat, India.

[8] Vasudevan. D.P., Lala. P.K. and Parkerson. J.P., "A Novel Approach for On-line Testable Reversible Logic Circuit Design", Proc. of the 13th Asian Test Symposium, 2004.

[9] M. R. Rahman and J.E. Rice, "Online Testable Ternary Reversible Circuit". In the Proceedings of the Reed-Muller Workshop May 25-26, 2011, Tuusula, Finland.

[10] M. R. Rahman and J.E. Rice, "On Designing a Ternary Reversible Circuit for Online Testability", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, 23-26 August 2011, pp 119-124.

[11] Nower. N. and Chowdhury A. R., "On the Realization of Reversible Ternary Systolic Array", Silver Jubilee Conference on Communications and VLSI Design Vellore Institute of Technology (VIT), Vellore, India, Oct 8-10, 2009, pp. 490-491.

[12] Muthukrishnan. A., Stroud. C. R., 2000, "Multivalued logic gates for quantum computation", Phys. Rev. A 62(5) 052309/1 -8.

[13] Chowdhury. A. R.., Nazmul. R., Babu. H. M. H., "A New Approach to Synthesize Multiple-Output Functions Using Reversible Programmable Logic Array", *IEEE, 19th International Conference on VLSI Design*, Hyderabad, India, 3-7 January 2006, pp. 311-316.