# An Improved Design of a Fault Tolerant Reversible Binary Parallel Multiplier

**Pankaj Bhardwaj, Maninder Singh**

***Abstract*— The complexity of hardware is increasing in day by day portable devices. Power is the basic constraint for any circuit & though for complex hardware the power factor plays vital role. Portable devices demands not only low power but fast speed. Such demand creates digital designers difficult to design complex hardware at the cost of low power & too with fast speed. In the recent years, reversible logic has emerged as a promising technology having its applications in low power CMOS, quantum computing, nanotechnology, and optical computing. This paper presents a new 5*5 parity preserving reversible logic gate, F2PG. The proposed parity preserving reversible gate can be used to synthesize the Boolean function. The proposed fault tolerant reversible full adder circuit can be realized using only one F2PG. and half adder using 2 IG. The proposed fault tolerant reversible binary multiplier has two parts one is fault tolerant partial product generator using FRG and second is adder circuit. It has been demonstrated that the proposed design offers less hardware complexity and is efficient in terms of gate count, garbage outputs and fault tolerant than the existing counterparts.**

***Index Terms*— reversible logic, multiplier, fault tolerant, parity preservation, nano metric scales.**

## I. INTRODUCTION

Power dissipation is one of the important parameters in the digital circuit design. In VLSI circuit designing where power dissipation plays an important role, there has been an increasing trend of packing more and more logic elements into smaller and smaller volumes and clocking them with higher frequencies. The logic elements are normally irreversible in nature and according to Landauer's principle [23] irreversible logic computation results in energy dissipation due to power loss. This is because; erasure of each bit of information dissipates at least KTln2 Joules of energy where K is Boltzamann's constant and T is the absolute temperature at which the operation is performed. By 2020 this will become a substantial part of energy dissipation, if Moore's law continues to be in effect which states that processing power will double every 18 months. This particular problem of VLSI designing was realized by Feynman and Bennet in 1970s. In 1973 Bennet [26] had shown that energy dissipation problem of VLSI circuits can be circumvented by using reversible logic. This is so because reversible computation does not require erasing any bit of information and consequently it does not dissipate any energy for computation. Reversible computation requires reversible logic circuits and synthesis of reversible logic circuits differs significantly from its irreversible counter part because of different factors [22]. The technological requirement of designing of energy dissipation free VLSI circuits, particular characteristics of synthesis and testing of reversible circuits and the tremendous advantage of quantum circuits have motivated scientists and engineers from various background to study various aspects of reversible circuits. But from the construction point of view classical reversible gates are easy to build [20, 21]. A lot of interesting works are already reported in literature in the field of synthesis [12-15], optimization [16], evaluation [17] and testing [13] of reversible circuits. In a short period the reversible computation has emerged as a promising technology having applications in low power CMOS [14], nanotechnology [15], optical computing [16], optical information processing, DNA computing [17], bioinformatics, digital signal processing and quantum computing. 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 uncomputing the computed information by recycling the energy in the system [18]. In the design of reversible circuits two restrictions should be considered [19]; 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 [20]. So, there is a one-to-one mapping between input and output vector. In an n-output reversible gate, the output vectors are permutations of the numbers 0 to 2n - 1.

**Manuscript received September 09, 2014.**

**Pankaj Bhardwaj**, M.Tech Student, Electronics and Communication Department,RIEIT, Rayat Institute of engineering & Technology, Railmajra(near Ropar), Distt. Nawanshahr

**Maninder Singh,** Associate Professor, Electronics and Communication Department,RIEIT, Rayat Institute of engineering & Technology, Railmajra(near Ropar), Distt. Nawanshahr

## II. OVERVIEW ON REVERSIBLE LOGIC

### *A. Basic Reversible Gates:*

There exist many reversible gates in the literature. Among them 2×2 Feynman gate (FG), shown in Fig. 1, 3×3 Peres gate (PG), shown in Fig. 2, 3×3 Toffoli gate (TG), shown in Figure 3 have been studied extensively. Because of their simplicity and low cost there are design approaches and tools that incorporate them separately or in combination with each other [20].

**Feyman Gate**

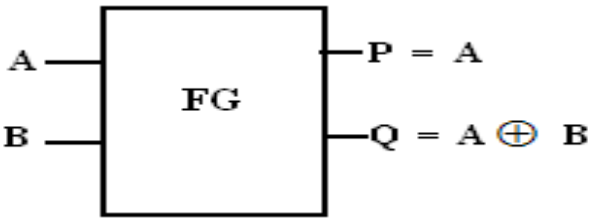


**Figure 1 (a)** Feyman Gate

**Table 1:** Truth Table for Feynman Gate

| Input A | Input B | Output P | Output Q |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

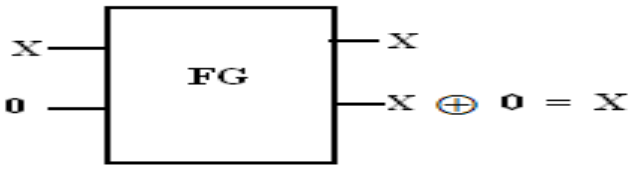


**Figure 1 (b)** Feynman Gate as Copying Output

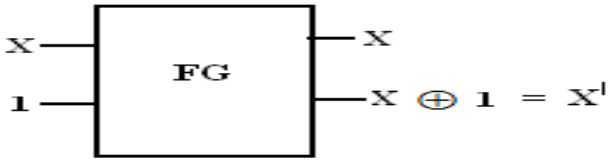


**Figure 1 (c)** Feynman Gate as Not Gate

**Peres Gate**

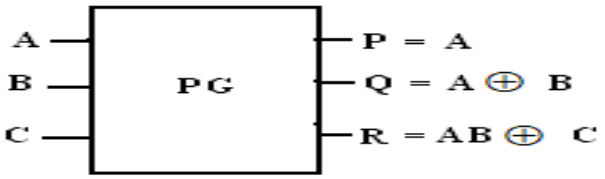


**Figure 2** Peres Gate

**Table 2: Truth Table for Peres Gate**

| Input A | Input B | Input C | Output P | Output Q | Output R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**Toffoli gate**

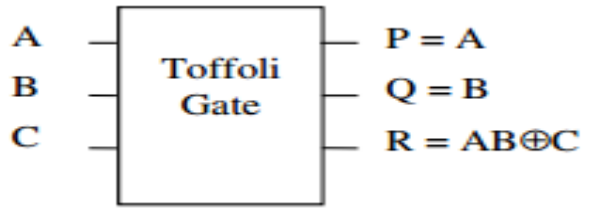


**Figure 3** : Toffoli gate

**Table 3:** Truth table of Toffoli gate

| A | B | C | P | Q | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

## III. OVERVIEW ON FAULT TOLERANT REVERSIBLE GATES

### *A. Parity Preserving Reversible Gates*

Fault tolerance is the property of reversible gates that enables a system to continue operating in the event of the failure of some its components. If the system itself made of fault tolerant components, then the detection and correction of faults become easier and simple. In communication and many other systems, fault tolerance is achieved by parity. Therefore, parity preserving reversible circuits will be the future design trends to the development of fault tolerant reversible systems in nanotechnology. A few parity preserving logic gates have been proposed in the literature. Among them 3×3 Feynman Double gate (F2G) shown in Figure4, 3×3 Fredkin gate (FRG) shown in Figure 5, 4×4 IG gate in Figure 6 are one-through gates, which means one of the inputs is also output. From Table 4, 5 and 6 it can be seen that the gates F2G, FRG and IG gates are parity preserving since they satisfy input parity match to output parity.

**Feyman Double Gate (F2G)**

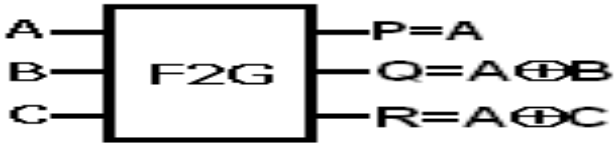


**Figure 4** Feyman Double Gate (F2G)

**Table4: Truth Table of Parity Preserving Feyman Double Gate (F2G)**

| A | B | C | P | Q | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**Fredkin Gate (FRG)**

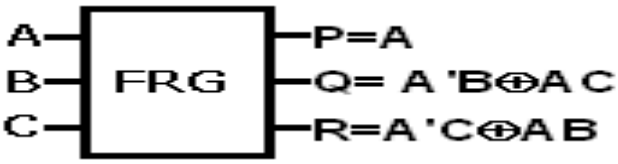


**Figure 5** Fredkin Gate (FRG)

**Table 5: Truth Table of Parity Preserving Fredkin Gate (FRG)**

| A | B | C | P | Q | R |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

**IG Gate**

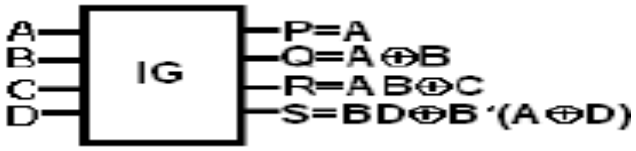


**Figure 6** IG Gate

**Table 6: Truth Table of IG 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 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

## IV. DESIGNING 4×4 MULTIPLIER

The design of the proposed multiplier is based on parallel operation of two steps.
Step I: Partial Product Generation
Step II: Reversible Fault Tolerant Parallel Adder
As mentioned earlier, the purpose of this work is to design a reversible fault tolerant multiplier circuit with the aim of optimizing of its hardware complexity to make it more economical in terms of number of garbage outputs, constant inputs, delay amd gate count without losing its efficiency. The proposed multiplier is implemented using fault tolerant reversible gates. The operation of a 4*4 reversible multiplier is shown in Fig. 7. It consists of 16 Partial product bits of the four bit inputs X and Y to perform 4 * 4 multiplications. Figure 8 shows example of 4*4 bit multiplication.

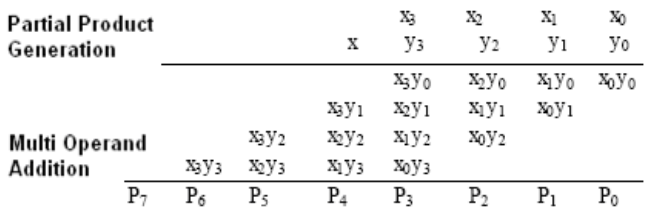


**Figure 7** Multiplication of 4*4 bit

### A. *Multiplication of 4×4 bit*

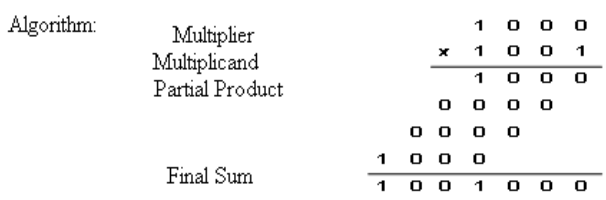


**Figure 8** example of 4×4 bit multiplication

### B. *Partial Product Generation (PPG)*

In this work the partial product generator is made by using the fault tolerant FRG gate. The FRG gate is used to perform AND operation by forcing one constant input as logic 0 whereas it produces required product term along with two garbage outputs. The Fig.9 shows the implementation of AND operation using FRG.

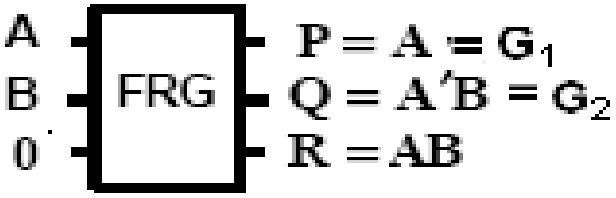


**Figure. 9** FRG as AND gate

Fault tolerant partial products are generated in parallel using 16 Fredkin gates (FRG) as shown in Fig. 10. This uses 16 FRG is a better circuit as it has less hardware complexity compared to other gates and moreover it posses parity preserving logic.

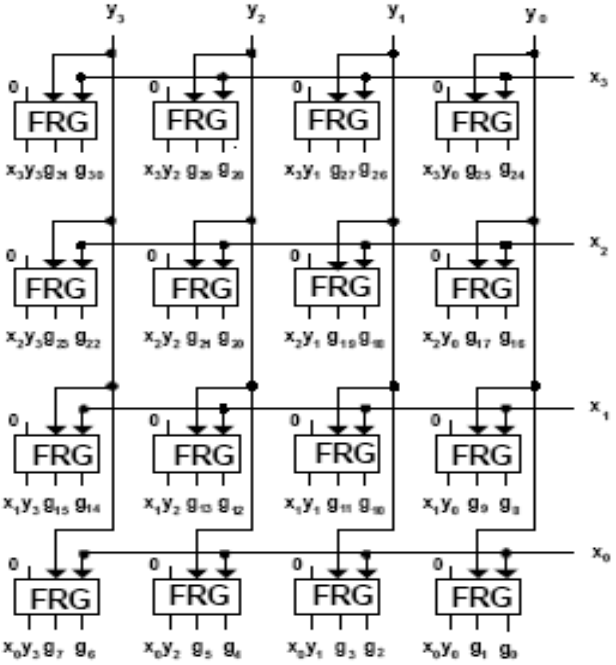


**Figure 10** Partial Product generation circuit using FRG gates

### C. *Design of Full Adder Using Conventional logic (irreversible gates):*

The sum & carry expression of conventional full adder is shown in figure 11
Sum = A⊕B⊕Cin (1)
Carry = (A⊕B)Cin⊕AB (2)

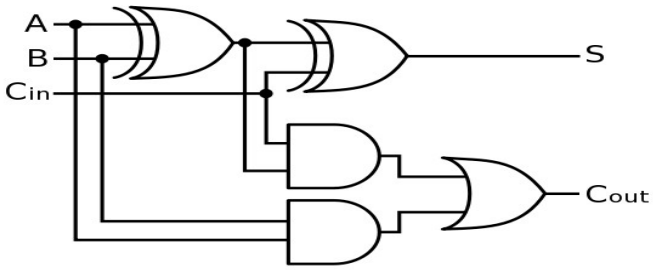


**Figure 11** Conventional Full Adder circuit

**Table 7: Truth Table of conventional Full Adder circuit**

| Input A | Input B | Input Cin | Sum | Carry |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

The conventional design of full adder is shown in Figure 11. From Table 7, it can be easily observed that for input combination of (A, B, Cin) 001, 010 & 100 the output remains same, i.e. sum=1, carry=0. Similarly, for combination 011, 101 & 110 the output remain as sum=0, carry=1. The main drawback of conventional FA is that firstly, as explained earlier it is not reversible; secondly, the adder posses the same output for three different combinations. If in case any of the combination faces stuck-at-fault to any logic then it will become difficult to predict the correct output. In order to avoid such faults the most used technique is based on parity checking.

*D. Using Peres gate (reversible gate):*

Realization of the efficient reversible full adder circuit given, it includes two 3×3 Peres gates as shown in Figure 12. The circuit is minimized in terms of gate count, garbage outputs, constant inputs and hardware complexity. But this is only reversible circuit not fault tolerant.

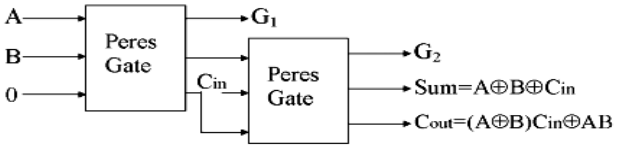


**Figure 12** Reversible Full Adder Circuit

*E. Design of Fault tolerant Half Adder using IG gate*

IG gate used as half adder is shown in Fig. 13. It requires two constant inputs of logic 0 and produces the required sum and carry term with two garbage outputs.

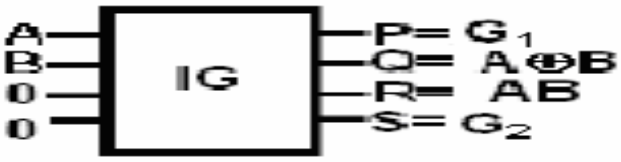


**Figure. 13** IG gate as Half Adder (FTHA)

*F. . Design of Fault Tolerant Full Adder using F2PG Gate*

The proposed fault tolerant full adder circuit adds three bit binary bits. The standard Boolean expression is:

Sum = A⊕B⊕Cin…… (3)

Carry=(A⊕B) Cin ⊕ AB…… (4)

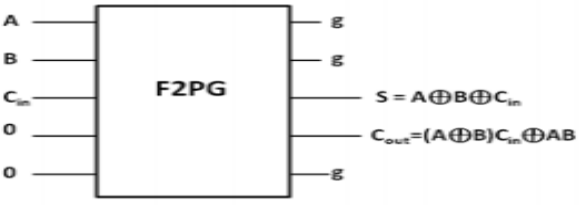


**Figure 14** Fault Tolerant Full Adder Circuit

To implement fault tolerant full adder using F2PG Gate, there is need of two constant inputs forced to logic zero whereas it produces required sum and carry along with three garbage outputs. The equivalent circuit is shown in Fig.14 where G1, G2 and G3 are three garbage outputs.

*G. Design of Fault Tolerant Reversible Multiplier*

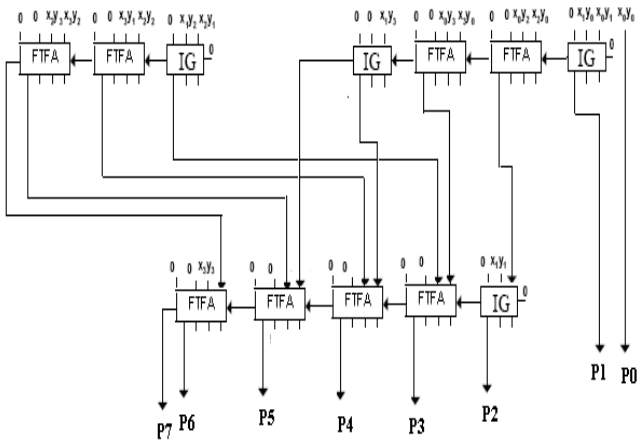


**Figure. 15** The Proposed Fault tolerant Parallel reversible Multiplier

The circuit of proposed Fault tolerant Multiplier shown in Figure. 15. It requires four IG gate FTHA for half adder and eight FTFA for full adder logic implementation

## V. RESULT AND DISCUSSION

The entire architecture is modeled using VHSIC hardware description language (VHDL). The coding is done on Xilinx ISE12.4 on Spartan 3E using target device: 3s500efg320-4 at speed grade of -4. For simulation purpose the Modelsim 6.2h has been used. The simulation result for proposed multiplier is shown in Fig. 16. The proposed multiplier is efficient in terms of number of gates, garbage outputs and constant inputs.

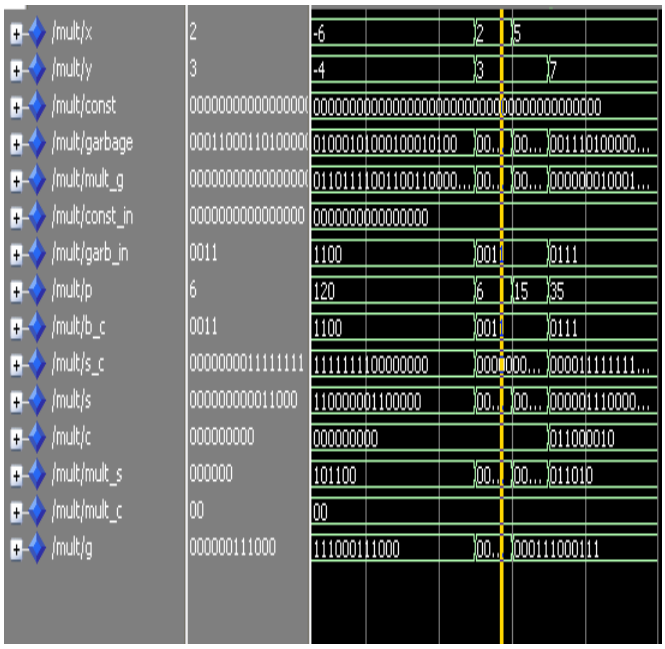


**Figure 16** Simulation result of Proposed Multiplier

**Table 7 Compression of Multipliers**

| Reversible Multiplier | Fault Tolerant Property | No. of gates | No. of Garbage Outputs | No of constant inputs |
|---|---|---|---|---|
| This Work | yes | 36 | 56 | 56 |
| Somayeh Babazadeh et.al, 2012 | yes | 48 | 64 | 52 |
| Haghparast et.al, 2009 | No | 28 | 28 | 28 |
| Haghparast et.al ,2009 | No | 36 | 28 | 28 |
| Haghparast et.al, 2008 | No | 52 | 52 | 52 |
| Shams et.al, 2008 | No | 52 | 56 | 56 |
| Thapyal et.al, 2006 | No | 53 | 58 | 58 |

## VI. CONCLUSION

Multiplier is a basic unit in computer arithmetic architecture. The energy consumption in multipliers turns out to be deeply linked to the reversibility of the computation. The primary objective of this paper is to gain insight into the Fault tolerant Reversible Computation and its use for making circuits energy efficient for long life. In the proposed work, we synthesized a parity preserving reversible multiplier circuit with the help of existing fault tolerant Fredkin, F2G, IG gate and F2PG. The comparison between the proposed multiplier and those of the previous multiplier showed that the proposed work is better in few aspects and can be encouraged due to its additional feature of fault detection technique. Thus, our proposed parity-preserving multiplier circuit can be used in designing fault tolerant reversible complex circuits like ALU. Using such circuit can be helpful not in terms of power saving but also acts as high speed multiplier for dedicated hardware.

## REFERENCES

[1] Mohinder Bassi, Pawandeep Kaur, Amandeep Singh, "Architectures and Methodologies for Reducing Power in Multipliers: A Literature Survey" in International Journal of Computer Applications Volume 85 , January 2014

[2] Raghava Garipelly, P.Madhu Kiran, A.Santhosh Kumar, "A Review on Reversible Logic Gates and their Implementation" in International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013

[3] Somayeh Babazadeh and Majid Haghparast, "Design of a Nanometric Fault Tolerant Reversible Multiplier Circuit" in Journal of Basic and Applied Scientific Research, pp. 1355-1361, 2012

[4] Ravish Aradhya H V, Muralidhara K N and Praveen Kumar B V, "Design of low power arithmetic unit based on reversible logic," in International Journal of VLSI and Signal Processing Applications, Vol.1, pp. 30- 38, 2011

[5] Md. Saiful Islam "Fault Tolerant Variable Block Carry Skip Logic (VBCSL) Using Parity Preserving Reversible Gates" International Journal of Computer and Electrical Engineering, Vol.3, pp 1793-8163 February, 2011

[6] H.R.Bhagyalakshmi et al. / International Journal of Engineering Science and Technology Vol. 2(8), pp 3838-3845, 2010

[7] Anindita Banerjee and Anirban Pathak, "Reversible multiplier circuit," in Third International Conference on Emerging Trends in Engineering and Technology (ICETET), pp 781-786, 2010

[8] S. Offermann, R. Wille, G. W. Dueck, and R. Drechsler, "Synthesizing multiplier in reversible logic," in Symposium on Design and Diagnostics of Electronic Circuits and Systems, pp 35–340, 2010

[9] M. Ehsanpour, P. Moallem and A. Vafaei, "Design of a Novel Reversible Multiplier Circuit Using Modified Full Adder," in Proceeding International Conference on Computer Design And Appliations (ICCDA), Vol. 3, pp 230-234, 2010

[10] M. Nachtigal, H. Thapliyal, N. Ranganathan, "Design of a reversible single precision floating point multiplier based on operand decomposition", Proceedings of the 10th IEEE International Conference on Nanotechnology, Seoul, Korea, pp 233–237August 2010

[11] H.R.Bhagyalakshmi and M.K.Venkatesha, "An improved design of a multiplier using reversible logic gates", International Journal of Engineering Science and Technology, Vol.2, pp 3838-3845, 2010

[12] M. Mahapatro, S. K. Panda, J. Satpathy, M. Saheel, M. Suresh, A. Kumar Panda, M. K. Sukla, "Design of Arithmetic Circuits Using Reversible Logic Gates and Power Dissipation Calculation", Proceeding International Symposium on Electronic System Design (ISED), pp 85-90, 2010

[13] Bruce, J.W., M.A. Thornton, L. shivakuamaraiah, P.S. kokate and X. Li, "Efficient adder circuits based on a conservative reversible logic gate", IEEE computer society Annual symposium on VLSI, Pittsburgh, Pennsylvania, pp 83-88, 2010

[14] Saiful Islam, Muhammad Mahbubur Rahman, Zerina Begum, and Mohd Zulfiquar Hafiz "Realization of a Novel Fault Tolerant Reversible Full Adder Circuit in Nanotechnology" The International Arab Journal of Information Technology, Vol.7, pp 3-7, July 2010

[15] P. K. Lala, J. P. Parkerson and P. Chakarborty, "Adder Designs using Reversible Logic Gates", Wseas transactions on circuits and systems, Vol.9, June 2010

[16] Lihui Ni, Zhijin Guan, and Wenying Zhu, "A General Method of Constructing the Reversible Full-Adder", Third International Symposium on Intelligent Information Technology and Security Informatics, pp 109-113, 2010

[17] R. Zhou, Y. Shi, H. Wang, J. Cao, "Transistor realization of reversible ZS series gates and reversible array multiplier", Microelectronics Journal 2010.

[18] Md. Saiful Islam1, M. M.Rahman1, Zerina Begum "Synthesis of Fault Tolerant Reversible Logic Circuits" IEEE, pp 2583- 2587, 2009

[19] Noor Muhammed Nayeem, Lafifa Jamal and Hafiz Md. Hasan Babu, "Efficient Reversible Montgomery Multiplier and Its Application to Hardware Cryptography," Journal of Computer Science, Vol.5, pp 56, 2009

[20] M. Islam, M. Rahman, Z. Begum, and M. Hafiz, "Low cost quantum realization of reversible multiplier circuit", Information Technology Journal, Vol.8, pp 208–213, 2009

[21] Anindita Banerjee and Anirban Pathak "An analysis of reversible multiplier circuits", pp 1-10, 2009

[22] Islam,M.S., Rahman,M.M., Begum,Z., Hafiz,M.Z., Inst. of Inf. Technol., Univ. of Dhaka, Bangladesh pp 396-401, September 2009

[23] M. Haghparast, M. Mohammadi, K. Navi, M. Eshghi, "Optimized reversible multiplier circuit", Journal of Circuits Systems and Computers Vol.2, pp 311-323, 2009

[24] Islam, M.S., Rahman, M.M., Begum, Z., Hafiz, M.Z., "Fault tolerant reversible logic synthesis: Carry look-ahead and carry-skip adders", Advances in Computational Tools for Engineering Applications, Vol.4, pp 396-401, July 2009.

[25] R. Landauer, "Irreversibility and heat generation in t he computing process," IBM Journal. Research and Development, vol. 3, pp. 183-191, July 1961.

[26] C. H. Bennett, "Logical reversibility of computation," IBM J. Research and Development, pp. 525-532, November 1973.