ISSN 2277-2685
IJESR/October-December 2024/ Volume-14/Issue-4/108-124
J Vasavi *et. al.,* /International Journal of Engineering & Science Research

ARITHMETIC AND LOGICAL UNIT BY USING REVERSIBLE GATES

J Vasavi[1] , Dr S Kishore Reddy[2] , Dr Siddhartha Bangaru[3] and B Dasharadha[4]

[1]M.Tech Student, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.
[2]Associate Professor, HOD, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
[3]Assistant Professor, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.
[4]Assistant Professor, ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., Hyderabad, India.

ABSTRACT: Given the current state of affairs, reversible logic architecture is all the rage due to the fact that it requires very little power. It is essential to use reversible logic when building circuits that use less power. Many different types of reversible gates are used in the process of reversible logic synthesis. These gates include the Feynman, Fredkin, Toffoli, New, Sayem, and Peres gates, amongst others. The more sophisticated circuits that are outlined in this article by using a basic reversible gate may be beneficial to a number of different types of circuits, including combinational circuits, an ALU, and certain sequential circuits. Additionally, a number of basic reversible gates, including as the TSG gate and the Peres gate, as well as instructions on how to utilise these gates to make adder circuits, are given. Surprisingly little attention has been paid to fault localisation in reversible circuits, despite the fact that reversible circuit test generation has been receiving a lot of attention as of late. The primary objective of this study is to identify and locate imperfections in binary reversible (permutative) circuits. The functional test-based fault localisation approach is where we are primarily concentrating our efforts. This approach requires the construction of an adaptive tree in order to recognise and localise "stuck-at" flaws in reversible circuits. A striking characteristic of reversible circuits is the existence of adaptable trees that function in a "symmetric" fashion. We are able to generate one half of the tree as a result of this, and then we are able to make the other half of the tree appear precisely like the first half. This is a great help. It is not very difficult to produce the tree since each test handles half of the problems [1] and the fault table has a high density of ones. This makes the tree production process quite straightforward. As a result of this, the problem of fault localisation in reversible circuits is less complicated than it is in typical irreversible circuits. We provide some preliminary results using a technique that makes use of traditional adaptive tree approaches. On top of that, we provide a fresh and effective method for constructing adaptive fault trees on the fly, which eliminates the need of generating fault tables.

I. INTRODUCTION

One of the most significant issues that has to be resolved in the design of modern VLSI circuits is power consumption. The decrease in heat dissipation and power consumption are two of the biggest problems integrated circuit designers encounter as transistor sizes are lowered. The objective of developing low-power circuits is already being achieved via techniques like voltage scaling and low-power designs. An important driving force behind the study of reversible adiabatic circuits is the growing need for low-energy-loss computing. Furthermore, reversible circuits play a crucial role in other nanotechnologies including quantum

computing [6]. It is crucial to test and look into any faults that arise before and after the design and manufacture of an integrated circuit in order to guarantee that it will operate properly and endure for a long period. Future highly parallel redundant logic systems will use the fault localization approach to locate and replace faulty modules, hence facilitating self-repair. The fact that the circuit may be reversed makes this task considerably simpler, as we demonstrate. Problem localization could be useful for diagnostics conducted throughout the process as well as for human and automated repair procedures. Automated processes include prefabricating a circuit with redundant modules, determining the reason for the failure, and then reconfiguring the circuit by swapping out the troublesome modules with the necessary spares.

An introduction to the basic terms and background knowledge required to perform a risk-free exploration of the realm of reversible logic is given in this chapter. A basic understanding of reversible logic is introduced at the outset. Terms that are fundamental and necessary to reversible logic are defined. After that, we will talk about the Feynman, Toffoli, and Fredkin gates—the three reversible gates that are most often employed. An introduction to reversible logic synthesis and many interesting heuristic formulations are shown here. This is an overview of the literature on the several approaches to reversible logic synthesis. Everything has a cost attached to it, just as in the real world. As previously established, in quantum computing, the outputs that are deemed "garbage" are forbidden and need to be eliminated. This is the cost associated with reversibility. Power dissipation is one of the most important factors in low power VLSI design, which has drawn a lot of attention to reversible logic during the last several years. Among the many possible uses for this material are low power CMOS, optical information processing, quantum computing, DNA computing, nanotechnology, and low power semiconductors. When data loss occurs, usable energy is wasted because hardware computations cannot be reversed. Landauer found that for each irreversible bit, a certain amount of energy was wasted. The formula K=1.3806505*10-23m2kg-2K-1 (joule/Kelvin-1) indicates that KTln2 of joules are required for functioning. where T is the system's operating temperature and is the Boltzmann constant.When dealing with high-speed computing activities or systems that contain a large number of bits, the amount of heat that is lost by these components becomes so big that it significantly influences performance and shortens the lifetime of these components. At room temperature, the amount of heat that is created by the loss of a single bit of information is negligible. Bennett (1973) demonstrated that KTln2 energy would not be wasted as long as the system permits the reproduction of inputs from visible outputs.Reversible logic is the system's foundation, which allows it to operate in both directions. Reversible computations are thus capable of receiving inputs from outputs and restarting from whatever state they were in prior to reaching the present one. For a circuit to be deemed reversible, it must be possible to reconstruct the input vector from the output vector in the same way that the outputs can be recovered from the inputs. The circuit may only be regarded as reversible in this manner. An additional need is that the inputs and outputs have to be assigned one-to-one, meaning that they cannot be swapped. Efficiency gains in computing might lead to a decrease in energy loss or perhaps its total elimination. without information loss

II. LITERATURE REVIEW

(1) Majid Haghparast et al. [1]

They put out an innovative reversible multiplier circuit with four by four bits. The reversible multiplier outperforms its competitors in terms of speed and hardware complexity, while also having fewer gates, less garbage outputs, and constant inputs. Haghparast and Navi suggested the HNG gate, which they used. An independent reversible complete adder is possible with the reversible HNG gate. The suggested HNG gate reversible multiplier circuit adds up two 4-bit binary values.

(2) Biswas, A.K. et al.[2]

Within the scope of this study, the Binary Coded Decimal (BCD) adder and the Carry Skip BCD adder are both examples of reversible logic implementations that have been improved and optimised. The results that were obtained in terms of latency, quantum cost, number of gates, and number of garbage outputs suggest that the modified designs are superior to the ones that were being used before. The efficiency of reversible BCD adders is a primary consideration throughout the design process, and we even specify lower limitations for their gates and garbage outputs to demonstrate this aspect. These circuits may also be used for the production of reversible quick multiplier units, which is an additional application.

(3) James R.K. et al. [3]

They proposed a method for the reversible conservative logic-based rapid addition of decimals (QAD) that is suitable for BCD additions that include several digits. Within the framework of the design, only reversible fault-tolerant Fredkin gates are used. Due to the fact that speed is the most important aspect of high-speed circuits, the objective of the implementation approach is to enhance speed while simultaneously reducing the number of delay levels. It is possible to utilise them for rapid multiplication as well as other jobs that are comparable.

(4) Thapliyal H.M., Arabnia H.R. [4]

The authors of this paper proposed the RPLA architecture, which is characterised by the use of reversible Fredkin and Feynman gates. The RPLA that has been proposed is capable of realising m functions that include n variables, and it accepts n inputs and generates m outputs. In order to demonstrate how the RPLA architecture works, we construct an RPLA with three inputs.

(5) H.M.H. Babu and A.R. Chowdhury [5]

A synthesis strategy was proposed as a means of achieving the goal of developing a binary coded decimal adder circuit that is unidirectional. Putting up a reversible full-adder circuit is the first step towards achieving a solution that is superior to the two circuits that are now in use.

(6) Himanshu Thapliyal and M. B. Srinivas [6]

Their two-concept NXN reversible multiplier makes use of the TSG gate in order to function. To begin, a reversible parallel adder that is built using TSG gates has the potential to reduce the number of steps that are necessary to add the partial products that are formed in parallel with a delay of d to less than two log2N.

(7) Shuli Gao et al [7]

To develop a multiplication and squaring function for large numbers in an effective manner, a methodical approach and design procedure is used, which involves the utilisation of small-size embedded multipliers. In order to assist the implementation of the squarer and multiplier, an overall design is proposed, and a set of equations is derived to facilitate its implementation.

(8) H. Thapliyal and M. B. Srinivas [8]

The first thing that they did was propose a reversible half adder, and then they utilised it to construct a reversible full adder that had multiplexers as its foundation. By using the TKS gate, a newly developed 3x3 reversible gate, which has two outputs that operate as 2:1 multiplexers, they were able to accomplished this.

(9) E. L. Rhod [9]

Field Programmable Gate Arrays, often known as FPGAs, simplify the process of programming hardware devices. This is in addition to the fact that they enable developers to experiment with various degrees of parallel functionality. Sadly, the numerous routing switches and wires that are necessary to give this level of flexibility result in the use of a tremendous amount of circuit space.

III. PROPOSED WORK

The multiplication operation is considered to be one of the most basic mathematical procedures. To put that into perspective, it is not as simple as adding or subtracting numbers since you have to take more time on each step. It is necessary for processors to make advantage of high-speed multipliers in order to attain excellent performance.

The flowchart given below in Figure 4.1 explains about how this work is planned and organized in completing this project.

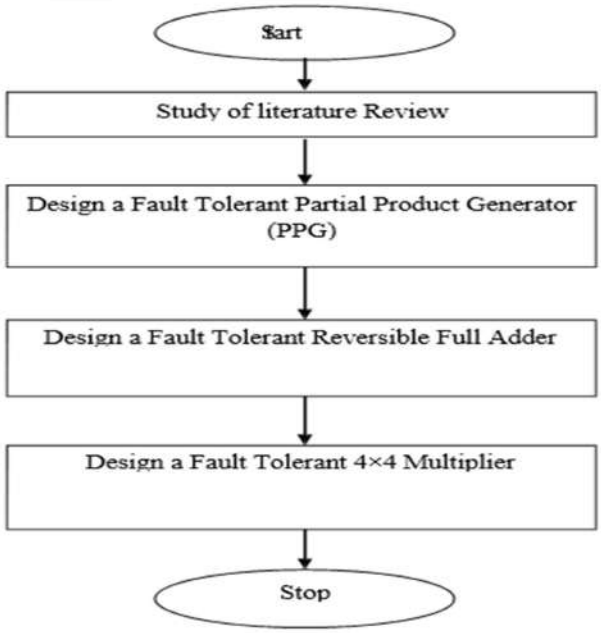

Figure 3.1: Work Plan of Project

3.1 Implementation of reversible fault tolerant 4×4 Multiplier

The design of the multiplier is based on parallel operation is done using two steps. Part I: Partial Product Generation

Part II: Reversible Fault Tolerant Parallel Adder

As was said before, the purpose of this investigation is to optimise the hardware complexity of a reversible fault-tolerant multiplier circuit in such a way that it minimises the amount of garbage outputs and maintains constant inputs while yet maintaining good efficiency.

In order to produce the multiplier, reversible gates were used, and these gates were intended to be fault resistant. One may see the operation of the 4×4 reversible multiplier by referring to Figure. It is formed of sixteen Partial Product bits, which are derived from the four-bit inputs X and Y, in order to perform multiplications of the size of four by four. One example of multiplication using four bits by four bits is seen in the image.

Partial Product				X_3	X_2	X_1	X_0
Generation			X	Y_3	Y_2	Y_1	Y_0
				X_3Y_0	X_2Y_0	X_1Y_0	X_0Y_0
			X_3Y_1	X_2Y_1	X_1Y_1	X_0Y_1	
Multi Operand		X_3Y_2	X_2Y	X_1Y_2	X_0Y_2		
Addition	X_3Y_3	X_2Y_3	X_1Y_3	X_0Y_3			
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Figure 3.2: Multiplication of 4×4 bit

3.1.1 Full Adder Using Peres gate (reversible gate)

The efficient reversible full adder circuit that is supplied includes two 3×3 Peres gates, which are represented in Figure 4.5. These gates are utilised to execute the circuit. An example of the corresponding data for a reversible adder is shown in Table 4.3. The complexity of the hardware, the number of constant inputs, the number of garbage outputs, and the number of gates were all reduced to a minimum. Furthermore, it has been shown in that it is feasible to construct a full adder circuit that is capable of being switched back on with just one constant input and two garbage outputs simultaneously.

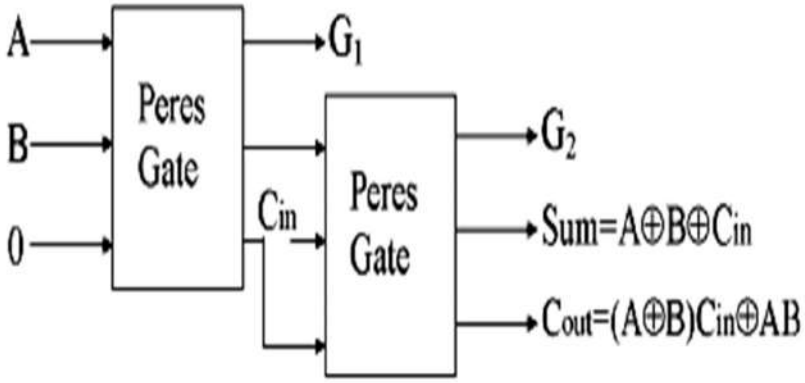

Figure 3.3 A complete adder circuit that is reversible and uses a Peres gate

Table 3.1 Truth Table of reversible Full Adder circuit

A	B	Cin	0	Sum	Carry	G1 (A)	G2 (A Xor B)
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	1	0	0	1
0	1	1	0	0	1	0	1
1	0	0	0	1	0	1	1
1	0	1	0	0	1	1	1
1	1	0	0	0	1	1	0
1	1	1	0	1	1	1	0

3.1.2 Design of Half adder using IG Gate

In order to create the IG gate, it is essential to have two constant inputs that are logic blank out. Additionally, the IG gate produces the necessary sum in addition to two outputs that are not useful any longer. The same circuit is shown in Figure 4.6, with two garbage outputs that are represented by the letters G1 and G2.

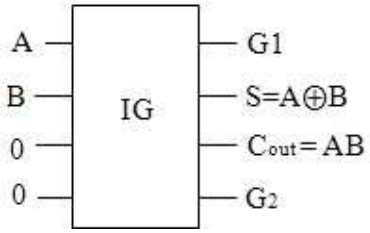

Figure 3.4: IG as Half Adder

The equations for garbage outs: G1
= A
G2 =
B′A

Table 3.2: The table of Half Adder with respects to IG Gate

A (A) (input)	B (B) (input)	C (C1) (constant input)	D (C2) (constant input)	P (g1) (garbage)	Q (S) (Sum)	R (C) (Carry)	S (g2) (garbage)
0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0
1	0	0	0	1	1	0	1
1	1	0	0	1	0	1	0

3.1.3 Design of Full Adder using IG Gate

A complete adder circuit can add a binary number of three bits. This is the usual Boolean expression:

$$\text{Sum} = A \oplus B \oplus C_{in} \quad (3)$$

$$\text{Carry} = (A \oplus B)\, C_{in} \oplus AB \quad (4)$$

When utilizing an IG Gate, you'll need to set two constant inputs to logic zero; the gate will then generate the necessary sum and three useless outputs. Figure 4.7 depicts the same circuit, with G1, G2, and G3 serving as two outputs for trash.

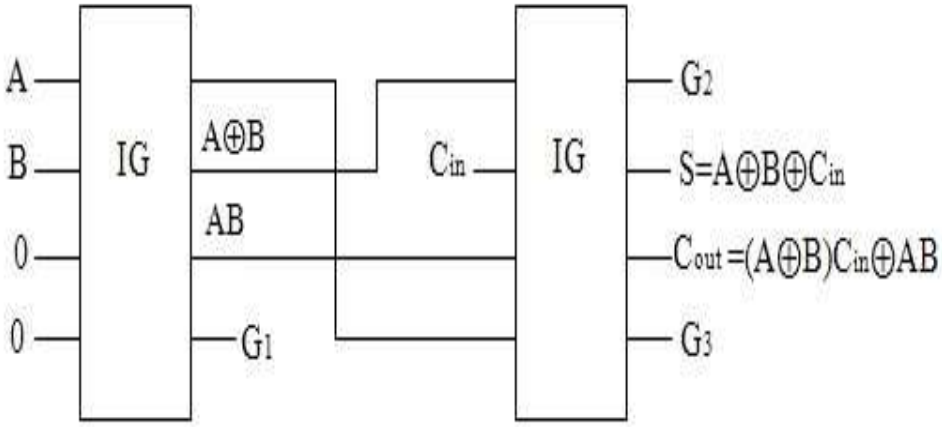

Figure 3.5 Two IG as Full Adder

The equations for garbage bits:

$G_1 = B'A$

$G_2 = A \oplus B$

$G_3 = C_{in} A \oplus C_{in}' B$

Table 3.3: The corresponding full adder table with respect to IG Gate

A (a) (input)	B (b) (input)	C (Cin) (input)	D (c1) (constant)	E (c2) (constant)	P (S) (carry)	Q (g1) (garbage)	R (g2) (garbage)	S (g3) (garbage)	T (sum) (sum)
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	1	1
0	1	1	0	0	1	0	1	0	0

3.2 Implementation of the Fredkin gate in the Design of a Partial Product Generator (PPG)

Multiplier partial products are generated in parallel using 16 Fredkin gates (FRG) as shown in

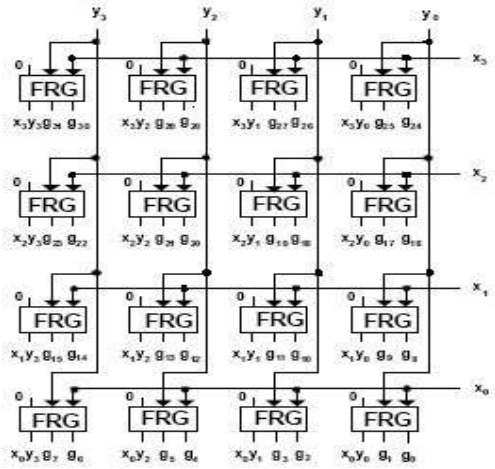

Figure 4.6. This uses 16 FRG gates because they are fault tolerant gates.

In our reversible 4x4 multiplier circuit, there are two components that make up the circuit. Following the creation of the partial products in parallel by the FRG gates, the addition is carried out with the help of the IG gate. There are currently existing circuits in the literature that are less efficient than the fault resistant reversible multiplier circuit. These circuits will be discussed more below. In our approach, the sole difference between the adder block and the partial product generator is that the adder block makes use of fault-tolerant gates. Because these gates maintain parity, we are able to check their parity to determine whether or not there is an error. Parity is defined as the circumstance in which the number of ones at the input and the number of ones at the output are equal to one another. Eight F2G gates are used as inputs to the partial product generator in our design, whereas sixteen FRG gates are used to build the partial product generator itself. The multiplier circuit is made up of a total of forty-four gates, four of which are integrated gate (IG) gates that are used as half adders and eight of which are IG gates that are utilised as full adders.

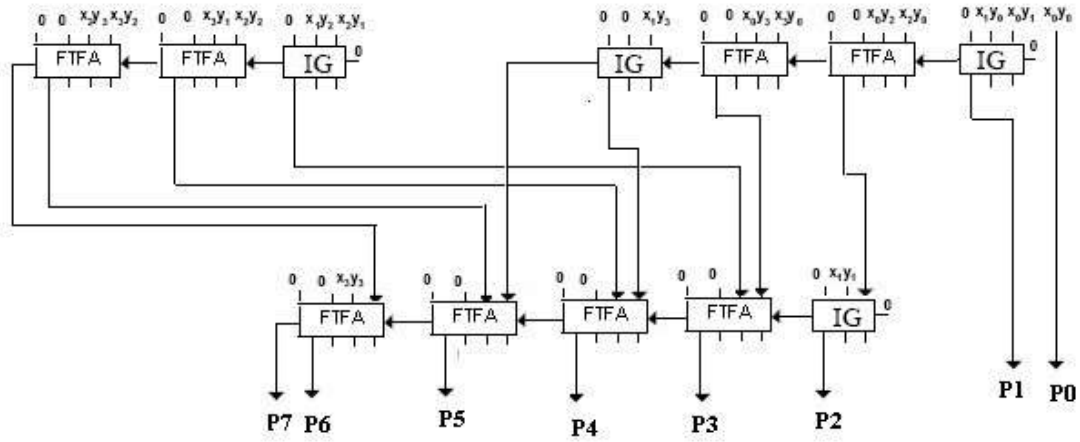

Figure 3.7: Reversible Multiplier with Fault Tolerance

IV. RESULTS AND DISCUSSION

This section presents all of the results of the simulation as well as the report on the synthesis. The Modelsim SE-EE 6.2h simulation tool is used in order to successfully verify the functionality of the design. The findings of time and area are obtained by the synthesis of VHDL code once the relevant simulation results have been obtained. In order to do this, the synthesis tool that was used was Xilinx XST.

4.1 The Partial Product Generator: A Report on Simulation and Synthesis

4.1.1 Fredkin Gate

Description: - The three inputs are A, B, C and the resultant output is P, Q R as shown in

Figure 5.1.

Input A,B,C

Output P=A

$R = A'B \oplus AC$

$S = A'C \oplus AB$

Figure 4.1 Description of the Fredkin fault-tolerant reversible gate (FRG) building block

FREDKIN_GATE1 Project Status			
Project File:	Fredkin_gate1.ise	Current State:	Synthesized
Module Name:	fredkin_gate	• Errors:	No Errors
Target Device:	xc3s500e-4fg320	• Warnings:	No Warnings
Product Version:	ISE 8.2i	• Updated:	Tue Nov 15 12:28:18 2011

Figure 4.2: Fredkin Gate Output

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	4656	0%
Number of 4 input LUTs	2	9312	0%
Number of bonded IOBs	6	232	2%

Figure 4.3: Summary of Device Usage for Fredkin Gate

4.1.2 Fredkin Gate as AND Gate

Description: - The three inputs are A, B, C and the resultant output is P, Q R as shown in Figure 5.4.

Input A= 1, B = 1, C = 0

Output P = A = 1

$Q = A'B \oplus AC = A'B = 0$

$R = A'C \oplus AB = AB = 1$

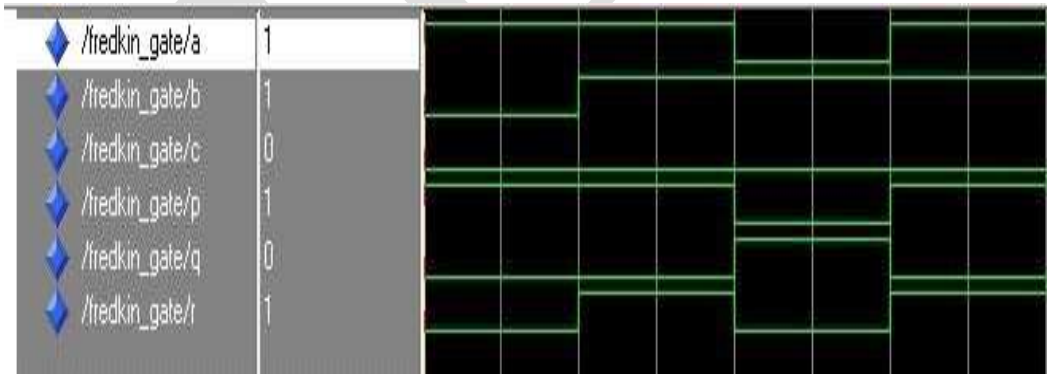

Figure 4.4: After Fredkin, AND Gate

4.1.3 Feyman Double Gate (F2G)

Description: - The three inputs are A, B, C and the resultant output is P, Q R. The output of

Feyman Double gate is shown in figure 5.5. Input

A,B,C

Output P = A,

Q = A ⊕ B,

R = A ⊕ C

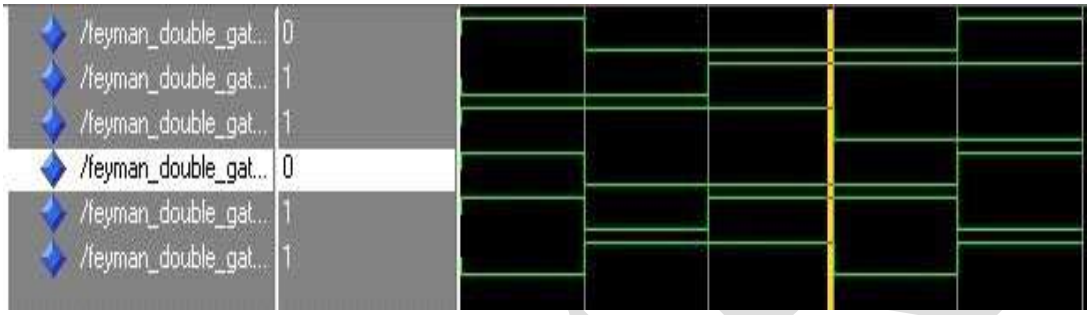

Figure 4.5 Findings from the Feyman Double Gate

4.1.4 Feyman Double Gate as Copying Gate

Description: - The three inputs are A, B, C and the resultant output is P, Q R. To make Feyman Double gate as copying gate put B and C = 0 the the result of three output is same as the input of first. The output of Feyman Double gate as copying gate is shown in figure 5.6.

Input A,B,C

Output P = A

Q =

A R

= A

Figure 4.6 The copying gate output of the Feyman double gate

4.2 Adder

4.2.1 IG Gate

Description:- The IG gate is 4×4 inputs A, B, C and D and outputs are P, Q, R and S, the resultant output is shown in Figure 5.9 whose binary value is given below.

Input: A = 0, B = 1, C = 1, D = 0

Output: P = A = 0

$$Q = A \oplus B = 1$$

$$R = AB \oplus C = 1$$

$$S = D \oplus B'(A \oplus D) = 0$$

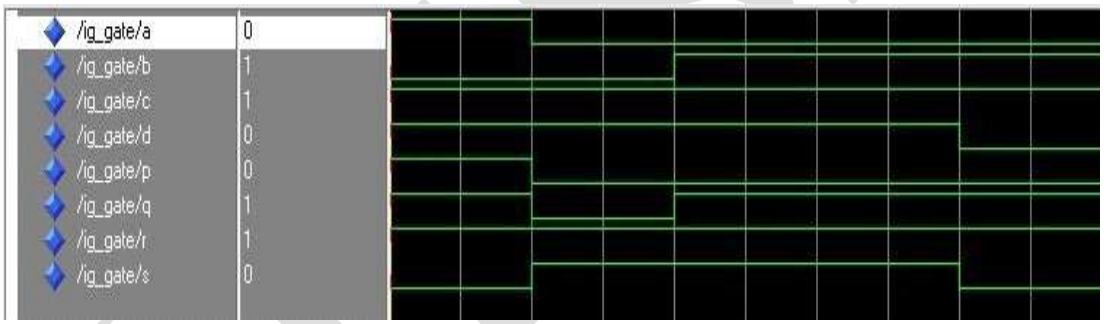

Figure 4.7: Output of IG Gate

PARTIAL_PRODUCT_GEN Project Status			
Project File:	Partial_Product_Gen.ise	Current State:	Synthesized
Module Name:	ig_gate	• Errors:	No Errors
Target Device:	xc3s500e-4fg320	• Warnings:	No Warnings
Product Version:	ISE 8.2i	• Updated:	Tue Nov 15 14:02:47 2011

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2	4656	0%
Number of 4 input LUTs	3	9312	0%
Number of bonded IOBs	8	232	3%

Figure 4.8 Information on the Use of the IG gate Device

4.2.2 IG Gate as Half Adder

Description:- The IG gate is 4×4 inputs A, B, C and D and outputs are P, Q, R and S, set the value of C and D =0 (Constant) the resultant output is shown in Figure 5.11 whose binary value is given below.

Input: A = 1, B = 1, C = 0, D = 0

Output: P = A = 1

$Q = A \oplus B = 0$

$R = AB \oplus C = 1$

$S = BD \oplus B'(A \oplus D) = 0$

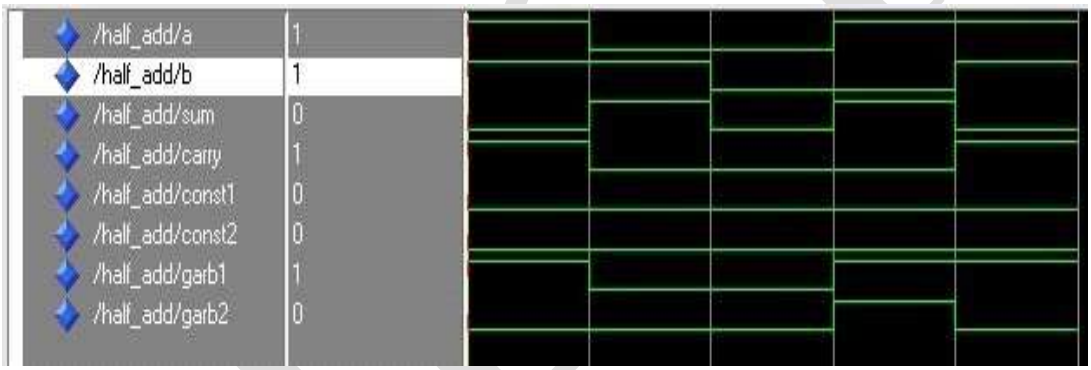

Figure 4.9: The Half Adder's Output

MULTIPLIER_PROP Project Status			
Project File:	multiplier_prop.ise	Current State:	Synthesized
Module Name:	half_add	• Errors:	No Errors
Target Device:	xc3s500e-4fg320	• Warnings:	No Warnings
Product Version:	ISE 8.2i	• Updated:	Tue Nov 15 14:54:27 2011

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2	4656	0%
Number of 4 input LUTs	3	9312	0%
Number of bonded IOBs	8	232	3%

Figure 4.10: An Overview of the IG gate's Use as a Half Adder

4.2.3 IG Gate as Full Adder

Description:- The IG gate is 4×4 inputs A, B, Cin, Const1 and Const2 and outputs are Sum, Carry, Garbage1, Garb2, Garb3, the resultant output is shown in Figure 5.13 whose binary value is given below.

Input: A = 1, B = 0, Cin =1 , Const = 0

utput: Sum = 0

Carry = 1

Garbage =1

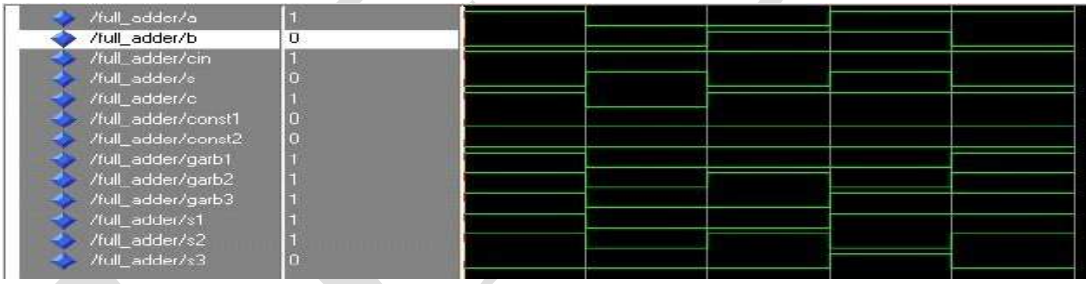

Figure 4.11: Fault-Tolerant IG Gate Output of a Full Adder

MULTIPLIER_PROP Project Status			
Project File:	multiplier_prop.ise	Current State:	Synthesized
Module Name:	full_adder	• Errors:	No Errors
Target Device:	xc3s500e-4fg320	• Warnings:	No Warnings
Product Version:	ISE 8.2i	• Updated:	Tue Nov 15 15:02:14 2011

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	3	4656	0%
Number of 4 input LUTs	5	9312	0%
Number of bonded IOBs	10	232	4%

Figure 4.12: Summary of IG gate device utilisation as full adder

V. CONCLUSION AND FUTURESCOPE

Since that time, the amount of data that can be processed by computers has expanded at a shockingly rapid pace. Energy is the most important resource for computing since it is inextricably linked to the reversibility of the calculation. The primary objective of this research was to get an understanding of reversible computation and its applications in the creation of devices that are both long-lasting and efficient in their use of energy. When performed on a multiplier, which is a basic arithmetic cell in computer arithmetic units, it is claimed that higher-order multiplication makes the most efficient use of the available electrical power. Using fault-tolerant Fredkin, F2G, and IG gates that were already in existence, we constructed a reversible multiplier circuit that maintains parity. This was accomplished by doing an analysis of the advantages of reversibility.

REFERENCES

[1] 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

[2] 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

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

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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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.

[10] 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

[11] 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

[12] 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

[13] 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

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

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