Shaik Nasar, K. Subbarao / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue5, September- October 2012, pp.1848-1855

Design & Implementation of MAC Unit Using Reversible Logic

Shaik Nasar[1]
Pursuing M.Tech, QCET, Nellore.

K. Subbarao[2]
Asst. Prof, QCET, Nellore.

Abstract

For irreversible circuits, loosing one bit of information dissipates (kTln2) joules of heat energy, where k is Boltzmann's constant and T is the absolute temperature. The reversible circuits do not dissipate energy as much as irreversible circuits. Thus, energy dissipation is proportional to the number of bits lost during computation. The reversible circuits do not lose information and can generate unique outputs from specified inputs and vice versa (there is a one-to-one mapping between inputs and outputs). In order to achieve low power designs Quantum computing and reversible circuits are used. In the majority of digital signal processing (DSP) applications the critical operations are the multiplication and accumulation. Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system. The main aim of the proposed system is to design a MAC unit using reversible logic with least number of gates, number of garbage outputs, delay and quantum cost in order to prove it as an efficient design.

Keywords: Reversible logic, Feynman gate, Peres gate, HNG gate, garbage outputs, Quantum cost, Quantum implementation.

1. INTRODUCTION

1.1 Introduction

Reversible computing is a model of computing where the computational process to some extent is reversible, i.e., time-invertible. A necessary condition for reversibility of a computational model is that the transition function mapping states to their successors at a given later time should be one-to-one. Reversible computing is generally considered an unconventional form of computing. There are two major, closely-related, types of reversibility that are of particular interest for this purpose: physical reversibility and logical reversibility. A process is said to be physically reversible if it results in no increase in physical entropy; it is isentropic. These circuits are also referred to as charge recovery logic or adiabatic computing. Although in practice no nonstationary physical process can be exactly physically reversible or isentropic, there is no known limit to the closeness with which we can approach perfect reversibility, in systems that are sufficiently well-isolated from interactions with unknown external environments, when the laws of physics describing the system's evolution are precisely known.

Probably the largest motivation for the study of technologies aimed at actually implementing reversible computing is that they offer what is predicted to be the only potential way to improve the energy efficiency of computers beyond the fundamental von Neumann-Landauer limit of kT ln(2) energy dissipated per irreversible bit operation. As was first argued by Rolf Landauer of IBM, in order for a computational process to be physically reversible, it must also be logically reversible. Landauer's principle is the loosely formulated notion that the erasure of n bits of information must always incur a cost of nk ln(2) in thermodynamic entropy. A discrete, deterministic computational process is said to be logically reversible if the transition function that maps old computational states to new ones is a one-to-one function; i.e. the output logical states uniquely defines the input logical states of the computational operation.

1.2 Problem Statement

For irreversible circuits, loosing one bit of information dissipates (kTln2) joules of heat energy, where k is Boltzmann's constant and T is the absolute temperature. The reversible circuits do not dissipate energy as much as irreversible circuits. Thus, energy dissipation is proportional to the number of bits lost during computation. The reversible circuits do not lose information and can generate unique outputs from specified inputs and vice versa (there is a one-to-one mapping between inputs and outputs). In order to achieve low power designs Quantum computing and reversible circuits are used.

1.3 *Aim*

In the majority of digital signal processing (DSP) applications the critical operations are the multiplication and accumulation. Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system. The main aim of the proposed system is to highlight an efficient design of a reversible MAC unit in order to prove that new circuit outperforms the previously proposed one in terms of number of gates, number of garbage outputs, delay and quantum cost.

1.4 Proposed System

A MAC unit is used to perform the multiplication and accumulation operations together to avoid unnecessary overhead on the processor in terms of processing time and the on-chip memory requirements. For example, in digital signal processing, FFT (Fast Fourier Transform) is most widely used where number of multiplications and additions should be performed simultaneously. The following expression represents a Fast Fourier Transform.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad 0 \le k \le N-1$$

$$W_N = e^{-j2\pi/N}$$

Most of the power consumption occurs during this data manipulation. Therefore to minimize the power consumption this block should be replaced by a Reversible MAC unit.

For efficient designing of a reversible circuit several criteria are needed to be considered:
Minimize the number of gates as possible.
Minimize the quantum cost of the circuit.
Total number of garbage outputs and usage of constant inputs should be minimized. By maintaining the above parameters and observing the previous design, we have proposed a novel Reversible MAC unit.
The proposed MAC unit is a 4-bit Multiplier along with a 8-bit adder and a 9-bit accumulator Register which uses Feynman Gates for producing fan outs, HNG gates as adders and Peres gates for producing the partial products.

2. THEORETICAL OUTLINE

Reversible computing was started when the basis of thermodynamics of information processing was shown that conventional irreversible circuits unavoidably generate heat because of losses of information during the computation. The different physical phenomena can be exploited to construct reversible circuits avoiding the energy losses. One of the most attractive architecture requirements is to build energy lossless small and fast quantum computers. Most of the gates used in digital design are not reversible for example NAND, OR and EXOR gates. A Reversible circuit/gate can generate unique output vector from each input vector, and vice versa, i.e., there is a one to one correspondence between the input and output vectors. Thus, the number of outputs in a reversible gate or circuit has the same as the number of inputs, and commonly used traditional NOT gate is the only reversible gate. Each Reversible gate has a cost associated with it called Quantum cost. The Quantum cost of a Reversible gate is the number of 2*2 Reversible gates or Quantum logic gates required in designing. One of the most important features of a Reversible gate is its garbage output i.e., every input of the gate which is not used as input to other gate or as a primary output is called garbage output.
In digital design energy loss is considered as an important performance parameter. Part of the energy dissipation is related to non-ideality of switches and materials. Higher levels of integration and new fabrication processes have dramatically reduced the heat loss over the last decades. The power dissipation in a circuit can be reduced by the use of Reversible logic. Landauer's principle states that irreversible computations generates heat of (KTln2) for every bit of information lost, where K is Boltzmann's constant and T the absolute temperature at which the computation performed. Bennett showed that if a computation is carried out in Reversible logic zero energy dissipation is possible, as the amount of energy dissipated in a system is directly related to the number of bits erased during computation. The design that does not result in information loss is irreversible. A set of reversible gates are needed to design reversible circuit. Several such gates are proposed over the past decades. Arithmetic circuits such as Adders, Subtractors, Multipliers and Dividers are the essential blocks of a Computing system. Dedicated Adder/Subtractor circuits are required in a number of Digital Signal Processing applications. Several designs for binary Adders and Subtractors are investigated based on Reversible logic. Minimization of the number of Reversible gates, Quantum cost and garbage inputs/outputs are the focus of research in Reversible logic.

2.1 Reversible Gates

The simplest Reversible gate is NOT gate and is a 1*1 gate. Controlled NOT (CNOT) gate is an example for a 2*2 gate. There are many 3*3 Reversible gates such as F, TG, PG and TR gate. The Quantum Cost of 1*1 Reversible gates is zero, and Quantum Cost of 2*2 Reversible gates is one. Any Reversible gate is realized by using 1*1 NOT gates and 2*2 Reversible gates, such as V, V+ (V is square root of NOT gate and V+ is its hermitian) and FG gate which is also known as CNOT gate. The V and V+ Quantum gates have the property given in the Equations 1, 2 and 3.
V * V = NOT …… (1)
V * V+ = V+ * V = I …….. (2)
V+ * V+ = NOT …… (3)

The Quantum Cost of a Reversible gate is calculated by counting the number of V, V+ and CNOT gates.

2.1.1 NOT Gate

The Reversible 1*1 gate is NOT Gate with zero Quantum Cost is as shown in the Figure 1.

Figure1. NOT gate

2.1.2 Feynman / CNOT Gate

The Reversible 2*2 gate with Quantum Cost of one having mapping input (A, B) to output (P = A, Q = AÅ B) is as shown in the Figure 2. Its Quantum implementation is as shown in Figure 3.

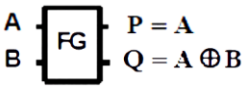


Figure 2. Feynman gate /CNOT gate

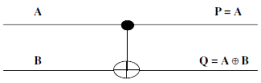


Figure 3. Quantum implementation of Feynman/CNOT gate

2.1.3 Toffoli Gate

The 3*3 Reversible gate with three inputs and three outputs. The inputs (A, B, C) mapped to the outputs (P=A, Q=B, R=A.BÅ C) is as shown in the Figure 4.

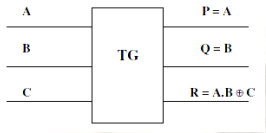


Figure 4. Toffoli gate

Toffoli gate is one of the most popular Reversible gates and has Quantum Cost of 5. It requires 2V, 1 V+ and 2 CNOT gates. Its Quantum implementation is as shown in Figure 4.

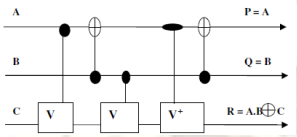


Figure 5. Quantum implementation of Toffoli gate

2.1.4 Peres Gate

The three inputs and three outputs i.e., 3*3 reversible gate having inputs (A, B, C) mapping to outputs (P = A, Q = AÅ B, R = (A.B) Å C). Since it requires 2 V+, 1 V and 1 CNOT gate, it has the Quantum cost of 4. The Peres gate and its Quantum implementation are as shown in the Figure 6 and 7 respectively.

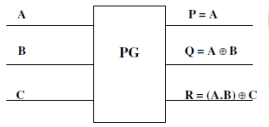


Figure 6. Peres gate

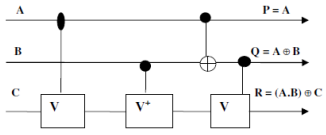


Figure 7. Quantum implementation of Peres gate

2.1.5 HNG Gate

The HNG gate is shown in Fig below, where each output is annotated with the corresponding logic expression. For more information about reversible logic gates see. One of the prominent functionalities of the HNG gate is that it can work singly as a reversible full adder unit.

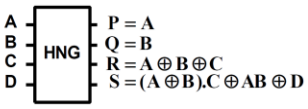


Figure 8. Reversible HNG gate as a reversible full adder

If the input vector is (A, B, Cin, 0), then the output vector P=A, Q=Cin, R=Sum, S=Cout

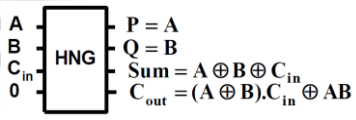


Figure 9. Reversible HNG gate as a reversible full adder

The Quantum cost of HNG gate is 6 with a time delay of 6.

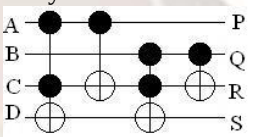


Figure 10. Quantum implementation of HNG gate

3. Implementation of Multiplier and Accumulate (MAC) Unit

In the majority of digital signal processing (DSP) applications the critical operations usually involve many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplier-Accumulator (MAC) is always a key to achieve a high performance digital signal processing system. In the last few years, the main consideration of MAC design is to enhance its speed. This is because, speed and throughput rate is always the concern of digital signal processing system. Pipelined multiplier / accumulator architectures and circuit design techniques which are suitable for implementing high throughput signal processing algorithms and at the same time achieve low power consumption. A conventional MAC unit consists of (fast multiplier) multiplier and an accumulator that contains the sum of the previous consecutive products. The function of the MAC unit is given by the following equation:

$F = \Sigma A_i B_i$

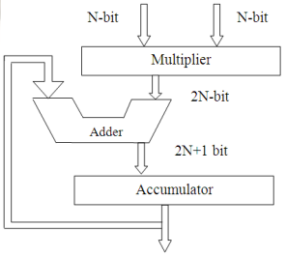


Figure 11: Basic structure of MAC

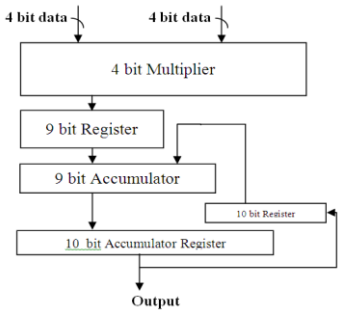


Figure 12: MAC architecture

The main goal of a DSP processor design is to enhance the speed of the MAC unit, and at the same time limit the power consumption. In a pipelined MAC circuit, the delay of pipeline stage is the delay of a 1-bit full adder (Jou, Chen, Yang and Su, 1995) . Estimating this delay will assist in identifying the overall delay of the pipelined MAC. In this work, 1-bit full adder is designed. Area, power and delay are calculated for the full adder, based on which the pipelined MAC unit is designed for low power.

3.1 Multiplier and Accumulator Unit

MAC is composed of an adder, multiplier and an accumulator. Usually adders implemented are Carry- Select or Carry-Save adders, as speed is of utmost importance in DSP (Chandrakasan, Sheng, & Brodersen, 1992 and Weste & Harris, 3rd Ed). One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle (Weste & Harris, 3rd Ed). Figure 12 is the architecture of the MAC unit which had been designed in this work. The design consists of one 9 bit register, one 4-bit
The product of Ai X Bi is always fed back into the 9-bit Ripple Carry accumulator and then added again with the next product Ai x Bi. This MAC unit is capable of multiplying and adding with previous product consecutively up to as many as eight times.
Operation: Output = Σ Ai Bi

In this paper, the design of 4x4 MAC unit is carried out that can perform accumulation on 8 bit number. This MAC unit has 9 bit output and its operation is to add repeatedly the multiplication results. The total design area is also being inspected by observing the total count of transistors. Power delay product is calculated by multiplying the power consumption result with the time delay.

3.2 Multiplication Concepts

There are two types of multipliers which are known as sequential and parallel multipliers. The first type iteratively computes the final product. It needs to use feedbacks and loops to compensate for the iterative portion. This design is too slow and not suitable for the reversible implementation. The second type (i.e., parallel multiplier), conventionally, consists of two main steps:

- Partial product generation
- Multi-operand addition

Partial products are independently computed in parallel–Consider two binary numbers A and B, of m and n bits, respectively.
There are mn summands that are produced in parallel by a set of mn AND gates –n x n multiplier requires n(n-2) full adders, n half-adders and n2 AND gates. The basic cell of the parallel array multiplier is shown in the figure . In this project a 4x4 parallel array multiplier is designed using reversible logic gates: Peres Gate in place of AND gate and PFAG gate in place of Full Adder.

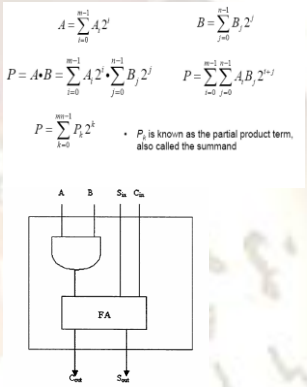


Figure 13. Basic cell of a parallel array multiplier

3.3 Design of Reversible Multiplier

The proposed reversible multiplier is designed in two phases.
Part I: Partial Product Generation (PPG)
Part II: Multi-Operand Addition (MOA)
The operation of a 4*4 reversible multiplier is shown in Figure 15. It consists of 16 Partial product bits of the X and Y inputs to perform 4 * 4 multiplications. However, it can be extended to any other n * n reversible multiplier.

In this we design a multiplier using reversible gates. The reversible gates used in the design of multiplier are Peres gate and Peres full adder gate.

			x	x_3	x_2	x_1	x_0
				y_3	y_2	yx_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	
		x_3y_2	x_2y_2	x_1y_2	x_0y_2		
	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 14. The operation of the 4×4 parallel multiplier

3.3.1 Partial Product Generation:

Partial products can be generated in parallel using 16 Peres gates as shown in Figure 16. This

uses 16 Peres gates and is a better circuit as it has less hardware complexity and quantum cost compared to other gates. An important point that should be considered is that in an n×n parallel multiplier (in reversible logic) for generating partial products in parallel, n copies of each bit of the operands are needed. Therefore, some fan-out gates are needed. The number of fan-out gates needed for the reversible 4×4 multiplier is 24

3.3.2 Reversible multiplier and accumulator circuit

The operation of the 4x4 multiplier is depicted in Figure 2.4. It consists of 16 partial product bits of the form xi.yi.

The reversible 4x4 multiplier circuit has two parts. First, the partial products are generated in parallel using Peres gates shown in Figure 2.3. Then, the addition is performed. The

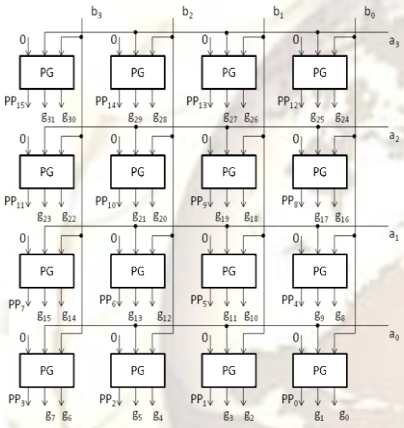


Figure 15. Partial Product generation circuit using Peres gates

basic cell for such a multiplier is a Full Adder (FA) accepting three bits and one constant input. We use PFAG gate as reversible full adder. The proposed reversible multiplier circuit uses eight reversible PFAG full adders. In addition, it needs four reversible half adders. It is possible to use PFAG gate as half adder as mentioned earlier in this study, but we use Peres gate as reversible half adder because it has less hardware complexity and quantum cost compared to the PFAG gate (quantum cost of Peres gate is 4 whereas for PFAG it is 8).

3.4 Accumulator unit

The circuit of figure 15 using the peres gates is a bit-wise multiplier which generates the partial products PP0 to PP15 for a 4x4 multiplication and these partial products will be supplied to the multiplier circuit shown in figure 16. The multiplier's construction concept is shown in figure 17 which developed based on multiplication shown in figure 14. The circuit of figure 16 (using FA, HA) uses 4 Half adders and 8 Full adders. The circuit of the multiplier is in fact an adder producing the 8-bit product output P0 to P7.

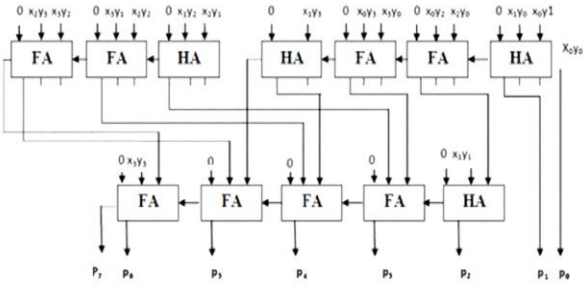


Figure 16. The concept of product generation

The accumulator and buffer both are as shown in figure 18. This circuit is constructed using the HNG, PG and FG gates. HNG gate is used as full adder to serve as the accumulator and the FG gates are used to serve as the buffer circuits. Each HNG gate produces 2 garbage outputs since we have not used the two outputs P & Q as shown in figure 19. The final output

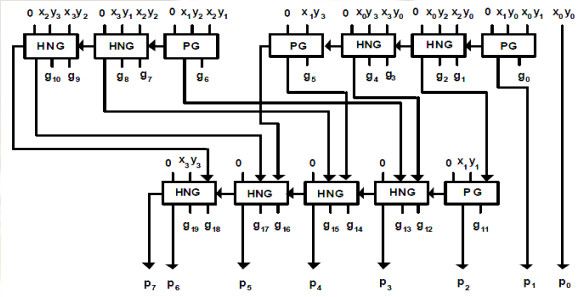


Figure 17. Product generation circuit using HNG & Peres gates

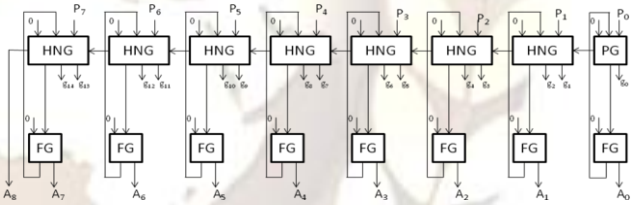


Figure 18. Proposed 4x4 reversible multiply and accumulate circuit using HNG gates and Feynman gates.

contains 9 bits including the carry generated during accumulation. The role of the FG gate is to serve as the buffer which can be cleared referring the figure 20 first input(A) of FG gate is SUM output og the HNG gate which will be brought out unchanged since the other input of the gate is made '0'. The other output, which is A is fed back to the HNG gate to serve as the prevous output. The FG gate is used here since there is no fanout in reversible logic. Furthur, it does not produce any garbage outputs.

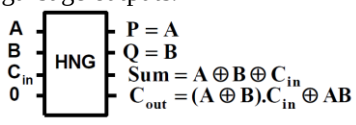


Figure:19. HNG gate

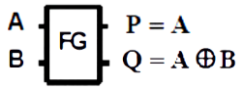


Figure 20. Feynman gate /CNOT gate

4. RESULTS AND DISCUSSION

4.1 Evaluation of the proposed reversible multiplier circuit:

The proposed reversible multiplier circuit is more efficient than the existing circuits presented in [17-19]. Evaluation of the proposed circuit can be comprehended easily with the help of the comparative results in Table 4.1.

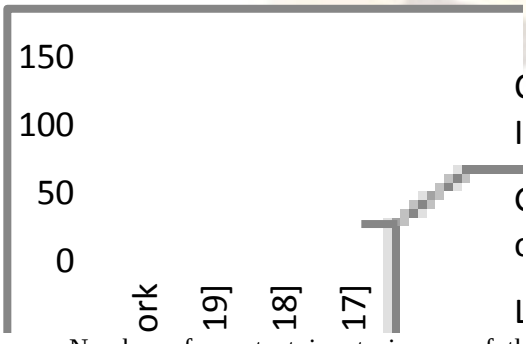
Table 1: Comparative results of various reversible multipliers

Multiplier Design	No of Logic Gates	No of Garbage outputs	No of Constant Inputs	Total Logical Calculations
This Work	28	52	28	80a+36ß
[19]	28	56	32	92a+52ß+36d
[18]	29	58	34	110a+103ß+71d
[17]	40	56	31	80a+100ß+68d

The only difference between partial products generation block in our design with the existing designs in [17, 18] is the use of Peres gates instead of Fredkin gates. This structure is proposed in [19]. We use it because the Peres gates have less logical calculation and less quantum cost than the Fredkin gates.

Garbage output refers to the output of the reversible gate that is not used as a primary output or as input to other gates. One of the other major constraints in designing a reversible logic circuit is to lessen number of garbage outputs. Our proposed reversible multiplier circuit produces 52 garbage output, but the design in [17] produces 56 garbage outputs, the design in [18] produces 58 garbage outputs and the design in [19] produces 56 garbage outputs. So, we can state that our design approach is better than all the existing counterparts in term of number of garbage outputs.

Table 2: Comparision of this work with the earlier ones



Number of constant inputs is one of the other main factors in designing a reversible logic circuit. The input that is added to an nxk function to make it reversible is called constant input. Our proposed reversible multiplier circuit requires 28 constant inputs, but the design in [17] requires 31 constant inputs, the design in [18] requires 34 constant inputs and the design in [19] requires 32 constant inputs. So, we can state that our design approach is better than all the existing designs in terms of number of constant inputs.

Comparing our proposed reversible multiplier circuit with the existing circuits in [17-19], it is found that the proposed design approach requires 28 reversible logic gates but the existing design in [17] requires 40 reversible gates and the existing design in [18] requires 29 reversible gates. So, the proposed circuit is better than [17, 18] in term of number of reversible logic gates, which is one of the other main factors in reversible circuit design. It is to be noted that the existing design in [19] also requires 28 reversible gates.

From the above discussion we can conclude that the reversible MAC unit we designed is best suited for the future technology.

5. SIMULATION RESULTS

5.1 Simulation Results for Reversible gates

The waveform shown below is the simulation results for PG GATE. Here the PG GATE having 3 inputs named as A, B & C and the ouputs are named as P, Q & R. The simulation results for PG GATE are observed by taking all combinations of the inputs. The ouputs verified with reference to the PG GATE definition.

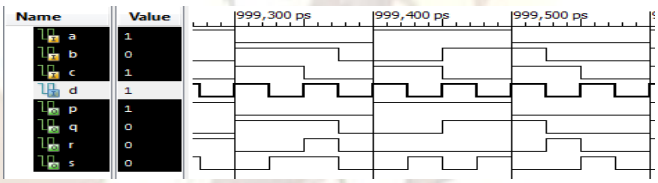


Figure 21: Simulation results of HNG Gate

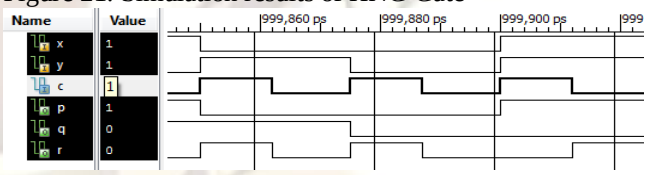


Figure 22: Simulation results of PG Gate

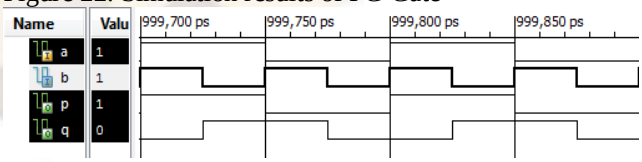


Figure 23: Simulation results of Feynmen Gate

5.2. Simulation Results of Reversible MAC blocks

The Reversible Multiplier is constructed by structural model by using two gates such as PG GATE and HNG GATE as components.

The waveform shown below is the simulation results of final Reversible Multiplier. Here the Multiplier has two four bit inputs x and y. So, the result is eight bit denoted by p. The output

can be verified by taking some possible inputs and observing the outputs.

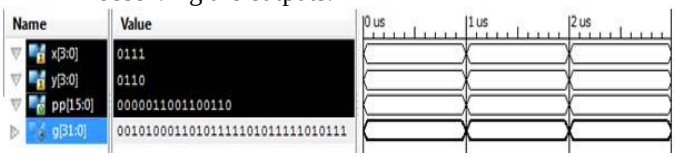


Figure 24:Simulation Results For Partial Product Term Generator Of Multiplier

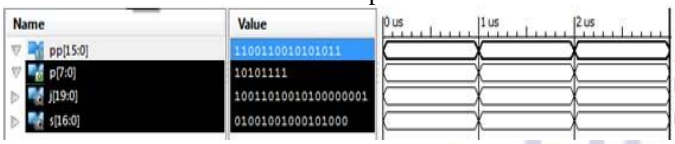


Figure 25: Simulation Results of Proposed Product Generator.

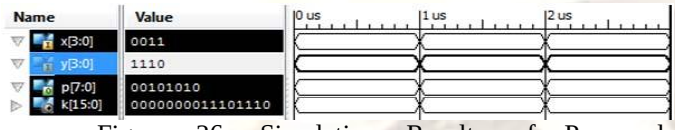


Figure 26: Simulation Results of Proposed Reversible Multiplier.

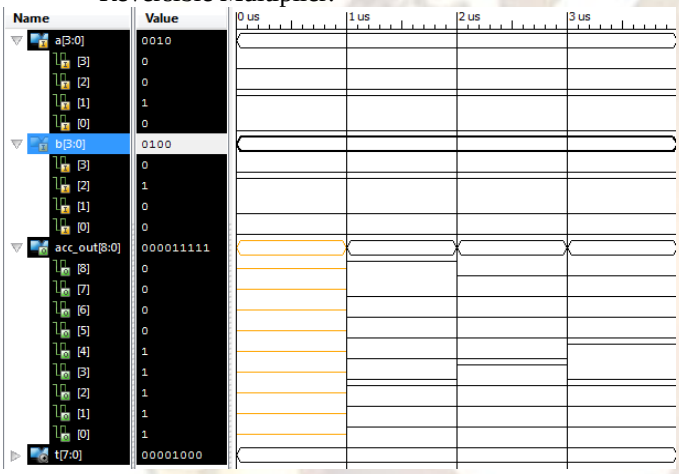


Figure 27: Simulation Results of Proposed Reversible Multiplier.

The Multiplier is designed by using two reversible logic gates: PG Gate and HNG gate. These two gates are described by using Dataflow model. The multiplier is described by Structural model.

6. Conclusion And Future Work

MAC unit is a basic arithmetic cell in computer processing units. Furthermore, reversible implementation of this unit is necessary for quantum computers. Targeting this purpose, various designs can be found in the literature.

We designed a novel 4x4 bit reversible multiplier circuit using Peres gates and HNG gates. Table 4.1 demonstrates that the proposed reversible multiplier circuit is better than the existing designs in terms of hardware complexity, number of gates, garbage outputs and constant inputs. Furthermore, the restrictions of reversible circuits were highly avoided. The proposed reversible 4x4 multiplier circuit can be generalized for N x N bit multiplication.

The prospect for further research includes the reversible implementation of more complex arithmetic circuits with less garbage outputs and low quantum cost.

References

[1]. Landauer, R., 1961. Irreversibility and heat generation in the computing process, IBM J. Research and Development, 5 (3): 183-191.

[2]. Bennett, C.H., 1973. Logical reversibility of computation, IBM J. Research and Development, 17: 525-532.

[3]. Kerntopf, P., M.A. Perkowski and M.H.A. Khan, 2004. On universality of general reversible multiple valued logic gates, IEEE Proceeding of the 34th international symposium on multiple valued logic (ISMVL'04), pp: 68-73.

[4]. Perkowski, M., A. Al-Rabadi, P. Kerntopf, A.Buller, M. Chrzanowska-Jeske, A. Mishchenko, M.Azad Khan, A. Coppola, S. Yanushkevich, V.Shmerko and L. Jozwiak, 2001. A general decomposition for reversible logic, Proc.RM'2001, Starkville, pp: 119-138.

[5]. Perkowski, M. and P. Kerntopf, 2001. Reversible Logic. Invited tutorial, Proc. EURO-MICRO, Sept 2001, Warsaw, Poland.

[6]. Thapliyal Himanshu, and M.B. Srinivas, 2005. Novel reversible TSG gate and its application for designing reversible carry look ahead adder and other adder architectures, Proceedings of the 10th Asia-Pacific Computer Systems Architecture Conference (ACSAC 05). Lecture Notes of Computer Science, Springer-Verlag, 3740: 775-786.

[7]. Saiful Islam, M.D. and M.D. Rafiqul Islam, 2005.Minimization of reversible adder circuits. Asian J.Inform. Tech., 4 (12): 1146-1151.

[8]. Feynman, R., 1985. Quantum mechanical computers, Optics News, 11: 11-20.

[9]. Toffoli T., 1980. Reversible computing, Tech Memo MIT/LCS/TM-151. MIT Lab for Computer Science.

[10]. Fredkin, E. and T. Toffoli, 1982. Conservative logic. Int'l J. Theoretical Physics, 21: 219-253.

[11]. Peres, A., 1985. Reversible logic and quantum computers, Physical Review: A, 32 (6): 3266-3276.

[12]. Azad Khan, Md.M.H., 2002. Design of full adder with reversible gate. International Conference on Computer and Information

Technology, Dhaka, Bangladesh, pp: 515-519.
[13]. Haghparast, M. and K. Navi, 2007. A Novel Reversible Full Adder Circuit for Nanotechnology Based Systems. J. Applied Sci., 7 (24): 3995-4000.
[14]. Haghparast, M. and K. Navi, 2008. Design of a Novel Fault Tolerant Reversible Full Adder For Nanotechnology Based Systems , World Appl. Sci.J., 3 (1): 114-118.
[15]. Haghparast M. and K. Navi, 2008. A Novel reversible BCD adder for nanotechnology based systems. Am. J. Applied Sci., 5 (3): 282-288.
[16]. Haghparast M. and K. Navi, 2008. A Novel Fault Tolerant Reversible Gate For Nanotechnology Based Systems. Am. J. Applied Sci., 5 (5): 519-523.
[17]. Thaplyal, H., M.B. Srinivas and H.R. Arabnia, 2005. A Reversible Version of 4x4 Bit Array Multiplier With Minimum Gates and Garbage Outputs, The 2005 International Conference on Embedded System and Applications (ESA'05), Las Vegas, USA, pp: 106-114.
[18]. Thaplyal, H. and M.B. Srinivas, 2006. Novel Reversible Multiplier Architecture Using Reversible TSG gate. IEEE international Conference on Computer Systems and Applications, pp: 100-103.
[19]. Shams, M., M. Haghparast and K. Navi, 2008. Novel Reversible Multiplier Circuit in Nanotechnology. World Appl. Sci. J., 3 (5): 806-810.
[20]. http://en.wikipedia.org/wiki/Reversible_computing.
[21]. http://en.wikipedia.org/wiki/Binary_multiplier.
[22]. A VHDL Primer – J. Bhasker, Pearson Education/ PHI, 3rd Edition.
[23]. Fundamentals of Digital Logic Design with VHDL – Stephen Brown and Zvonko Vranesic, McGraw Hill, 2nd Edition.
[24]. Digital Systems Design Using VHDL – Charles H. Roth Jr., PWS Publications, 2nd Edition.
[25]. Digital Design Principles & Practices – John F. Wakerly, PHI/ Pearson Education Asia, 3rd Edition.