International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 3, March - 2014

Design Optimization of Vedic Multiplier using Reversible Logic

Anuva Das
Department of Electronics and Communication Engineering
SRM University
Kattankulathur - 603203, Tamil Nadu, India

Mrs. J. K. Kasthuri Bha
Department of Electronics and Communication Engineering
SRM University
Kattankulathur - 603203, Tamil Nadu, India

***Abstract*—With DSP applications evolving continuously, there is continuous need for improved multipliers which are faster and power efficient. Reversible logic is a new and promising field which addresses the problem of power dissipation. It has been shown to consume zero power theoretically. Vedic mathematics techniques have always proven to be fast and efficient for solving various problems. Therefore, in this paper we implement Urdva Tiryakbhayam algorithm using reversible logic thereby addressing two important issues – speed and power consumption of implementation of multipliers. In this work, the design of 4x4 Vedic multiplier is optimized by reducing the number of logic gates, constant inputs, and garbage outputs. This multiplier can find its application in various fields like convolution, filter applications, cryptography, and communication. Coding for 4x4 Vedic multiplier is done using VHDL.**

Keywords—Multipliers, Urdva Tiryakbhayam algorithm, Reversible Logic, Vedic Multiplier, Optimization, Quantum cost.

I. Introduction

Continuous research in various technological fields provides us with enumerable useful and handy devices with smaller dimensions. The two major performance factors of these indispensible devices are power consumption and operational speed. The performance of these devices depends on the performance of the processor integrated in it. Multipliers are considered to be one of the prime components in any processor like digital signal processor, microprocessor, micro controllers or other computing machines. So, the efficiency of the aforementioned device components are greatly governed by the effective structure, computation speed, computation method and power efficiency of multiplier they use.

Over the years various techniques have been employed to optimize the design of multipliers. Several types of multipliers like array multiplier, Booth multiplier, Baugh-Woolley multiplier, Wallace tree multiplier are present in literature. The most commonly used multiplier – the array multiplier, has high operational speed but it consumes more power with high requirement of space to implement large number of components required. Wallace tree method offers high speed multiplication but circuit layout is difficult because of the structural irregularity [19]. Again Booth multiplier suffers from complexity in generation of partial products using Booth encoding. Baugh-Woolley multipliers require much area and become slow for higher number of operand bits. In recent years a lesser known branch of mathematics known as Vedic mathematics has been used to construct multipliers. These Vedic multipliers are speed efficient and have a simple and regular structure thereby addressing all the issues faced by other regular multipliers.

Vedic mathematics [1] has its foundation in Vedic scriptures of ancient India. Sri Bharati Krishna Tirthaji reformed a set of 16 Sutras and 13 upa-sutras by solving the mysteries hidden in portion of Atharva Veda called Ganita sutra [2] which means “the easy mathematical formula”. The effectiveness and essence of Vedic mathematics lies in its computational simplicity and closeness to the working of human mind. Vedic algorithms have the wonderful capability to trim down intricate problems into easily understandable calculations so as to yield quicker results. These advantages of Vedic mathematics are found to be very attractive in solving problems in all branches of mathematics including arithmetic, calculus, geometry, computing etc. So, the application of Vedic mathematics in the design of a multiplier yields astounding results. In [17] proposes a Vedic multiplier which is faster than conventional multipliers and its FPGA implementation shows that hardware realization of Vedic algorithms is also easy. In [21] design of multiplier using Urdva Tiryakbhayam sutra performs better in terms of power and delay.

Power dissipation is one of the major aspects involved in designing a circuit. Using Vedic formulae better speed can be achieved but power consumption is still a matter of concern. In high speed devices clock frequency is kept high to perform computations fast with a major drawback of high power dissipation which is directly proportional to the operating clock frequency [9, 10]. This power dissipation problem can be resolved by using reversible logic technology. Reversible logic is recent trend in Low power electronics showcasing the feature of zero power dissipation. This scheme alleviates the problem of energy consumption; however one must take care of delay of the circuit designed with reversible logic.

In [11] a 4x4 multiplier is proposed using Fredkin gates to generate partial products and TSG to design full adders. Again in [14] 4x4 multiplier is developed using HNG gate as full adder to add the partial product generated using Fredkin gate. In [9] a reversible Urdva Tiryakbhayam multiplier has been proposed using Peres gate, BVPPG gate, NFT gate, HNG gate and Feynman gate and proposed design is optimized in terms of various parameters compared to designs present in literature. In this paper we use a different approach followed in [18] and by doing so we reduce the number of gates in adder units thereby reducing the total gate count, number of constant inputs, garbage outputs and quantum cost. The rest of the paper is organized as follows: section II deals with emergence and basics of reversible logic, Section III is about Urdva Tiryakbhayam algorithm, section IV is about reversible implementation of optimized of design, section V shows comparisons of various existing reversible 4x4 multiplier with the optimized design and concludes the work.

II. REVERSIBLE LOGIC

A. Emergence of reversible logic

Physical reversibility [8] signifies a process that dissipates no energy to heat and logical reversibility states that earlier state of a process is ascertainable from the backward computation of a state at any given point of time. Complete physical reversibility is attainable only when the process is logically reversible [5].

Computation processes that erase information bits are irreversible. In 1961 Landauer [3] specified that every time a bit of information is lost equivalent physical entropy is generated. This generated entropy will be transformed into heat as basics laws of thermodynamics say that entropy cannot be destroyed merely [5]. In accordance with Landauer's principle one bit of information loss generates KTln2 joule of heat energy where K is Boltzmann constant and T is the operating temperature in Kelvin scale. In room temperature erasure of one bit information gives off negligible amount of heat but large amount of heat generated as result of huge amount of information loss in high speed operation that engages more operational bits has serious effect on device performance and durability of components.

As predicted in Moore's law we found exponential increment in computer speed and power dissipation due to integration of enormous components in minimized area. This refinement of technology is bounded by lots of fundamental limits that are governed by well-established fundamental laws of physics and are technology independent [5]. At this point low power technologies may find its interest in quantum computing as an alternative to sustain the development flow. Quantum computation is based on unitary transformations that are reversible [11]. In 1973 C. H. Bennett [4] revealed that energy dissipation in a circuit can be avoided completely if the circuit is made up of reversible logic gates. So, reversible logic will play an important role in future technology.

B. Basic terminology, design constraints associated with reversible logic gate

In a reversible logic gate inputs are related with outputs by using one -to-one mapping between them. A reversible logic gate has a very interesting characteristic that for a particular pattern of input it gives a particular output pattern. This phenomenon helps to determine the input pattern by observing the outputs solely as well as input can be retrieved from output [11]. A reversible logic gate with n number of inputs and n number of outputs can be addressed by nxn gate and can be represented as:

$$I_v = (I_1, I_2, I_3 \ldots I_n)$$

$$Ov = (O_1, O_2, O_3 \ldots O_n)$$

Where I_v and O_v are input and out vectors respectively.

1) Design parameters and constraints

a) Design parameters of reversible logic circuits:

The design effectiveness of reversible logic circuits is reflected by the factors [15] mentioned below:

- Gate count: Total number of reversible logic gates used to implement the intended logic circuit.
- Constant Inputs: Inputs that are necessary to accomplish a specific function and remain unaltered all the way through the design.
- Garbage outputs: These are the outputs which are generated due to given inputs but irrelevant to realize the required logic function. These outputs remain unused in the design but are vital to preserve the reversibility of the circuit.
- Quantum cost: Quantum cost logic gate is decided by evaluating the number of primitive gates required to realize that reversible logic gate. Quantum cost of a reversible circuit is thus the overall quantum cost of all the logic gates used to construct it.
- Total Reversible Logic Implementation Cost (TRLIC) [9]: This refers to the summation of gate count, constant inputs, garbage outputs and quantum cost of the circuit.

b) Design constraints:

In a reversible logic circuit design two restrictions should be maintain strictly.

- Fan out of each signal including primary inputs will be one i.e. fan out is not allowed.
- Feedback loops are also prohibited.

Logic synthesis of reversible logic circuits should have the following objectives to achieve optimized structure:

- Design should use minimum number of logic gates.
- Constant inputs should be minimum.
- Number of garbage output should be minimum.
- Quantum cost should be kept as low as possible.

C. Reversible Logic Gates

1) Feynman Gate[6]: It is 2x2 gate. If the first input i.e. A is given as 1 the second output will be the complement of the second input i.e. B. so, this gate is also known as Controlled Not Gate. It can also be used to copy inputs. Quantum cost of this gate is one.

2) Peres Gate[7]: It is a 3x3 gate. It can be used as a half adder with third input i.e. c as 0.It also serves the purpose of fan out. Quantum cost of this gate is four.

3) HNG Gate [12]: It is a 4x4 gate. A single HNG gate can serve as a one bit full adder. Quantum cost of this gate is six.

4) BVPPG gate: In [16] this 5x5 reversible gate is proposed. It is basically for multiplication and can generate two partial products at a time. Quantum cost of this gate is ten.

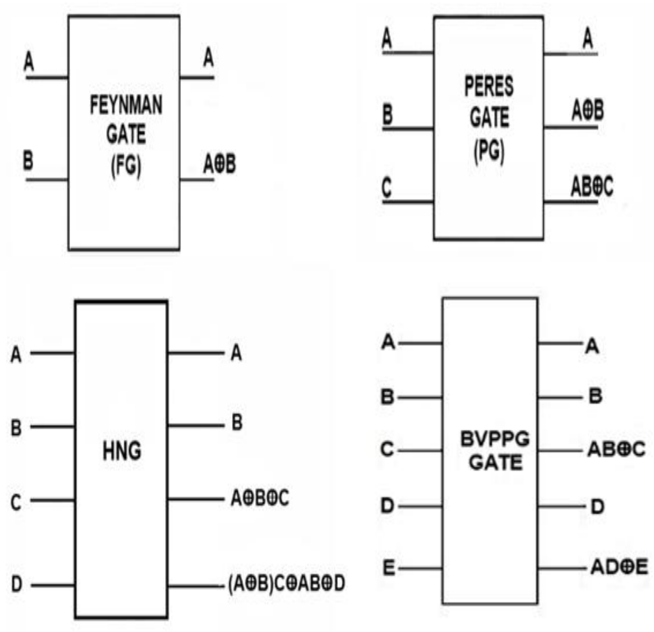


Fig. 1. Reversible logic gates.

III. Multiplication using Urdva Tiryakbhayam Sutra

A Vedic maths offers two sutras – Urdva Tiryakbhayam sutra and Nikhilam Sutra for multiplication. Nikhilam sutra is best used for numbers which are nearer to the base of 10,100, 1000 and increased power of 10 [20], whereas Urdva Tiryakbhayam can be used for any multiplication. The most powerful Vedic multiplication sutra Urdva Tiryakbhayam means 'Vertically and Crosswise". This technique is applicable for any type of number system. General procedure for Urdva Tiryakbhayam Fig. 2.

Algorithm: Let us consider two digit (for binary number system consider 2 bits) multiplicand and multiplier as "A1 A0" and "B1 B0" respectively and the result as R3R2R1R0.

- Multiplication starts with LSB of the operands i.e. vertical multiplication of A0 and B0 will generate the LSB of the result i. e. R0. For binary numbers no carry will be generated at this stage.

$$R0 = A0B0 \quad (1)$$

- R1 is obtained by crosswise multiplication of A0, B1 and A1, B0 and then adding the two products. In this stage crosswise multiplication and simultaneous addition of the product generates R1 as sum and carry say C1.

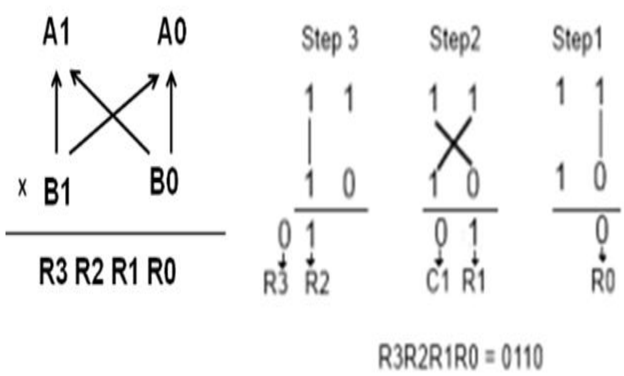
$$C1R1 = A0B1 + A1B0 \quad (2)$$

- Again the vertical multiplication between two MSB of the operands i. e. A1 and B1 takes place and product is added with the generated carry C1 in the previous stage to give the third bit of result i. e. R3 as sum and fourth bit R4 as carry.

$$R3R2 = A1B1 + C1 \quad (3)$$

Final result is obtained by concatenating R3, R2, R1, and R0. This method is applicable for n number of bits.

An illustration of 2x2



multiplication for binary number is shown below.

Fig. 2. General procedure of Urdva Tiryakbhayam sutra with illustration.

A 2x2 multiplier [17] using conventional AND and XOR gate can be implemented as shown in Fig. 3.

For 4x4 multiplication we divide both multiplicand and multiplier into two equal section and consider each section as two bit operand and then we follow the above algorithm for multiplication of all four sections. Let the 4 bit multiplicand be "$A_3A_2A_1A_0$" and multiplier be "$B_3B_2B_1B_0$". The fore sections will be "A_3A_2", "A_1A_0", 'B_3B_2", "B_1B_0". So we need four 2x2 multiplier for a 4x4 multiplication. After all the partial products are generated from four 2 bit multiplications we need to add them in a systematic way to get the final result. This adder section may vary from one design to another. Multiplications between all four sections are depicted in Fig. 4.

Say products generated from four multipliers are "I0 I1 I2 I3", "J0 J1 J2 J3", "K0K1K2K3", "L0L1L2L3". In our work we adopted the addition process [18] to get final 8bit result R7R6R5R4R3R2R1R0. "R1R0" will be same as "I1 I0". To get "R5R4R3R2" we need two 4 bit additions. First we add "J3 J2 J1 J0" and "K3K2K1K0" and we get 4bit sum with fifth bit as carry. This sum is again added with "L1L0I3I2" which we get by concatenating the first two left hand side bits of fourth multiplier result and last two right hand side bits of first multiplier result. The second addition results in a 4bit sum with fifth bit as carry. This 4bit sum is taken as "R5R4R3R2". Now the two carry generated from two additions will be added and will give a sum bit and carry bit as result. These two bits are then added with "L2L3". This addition result will be equal to "R7R6". Concatenation of "R7R6", "R5R4R3R2", and "R1R0" will give the 8bit final product "R7R6R5R4R3R2R1R0". Carry generated in last addition will not be considered in design. Block diagram of this approach is shown in Fig. 6.

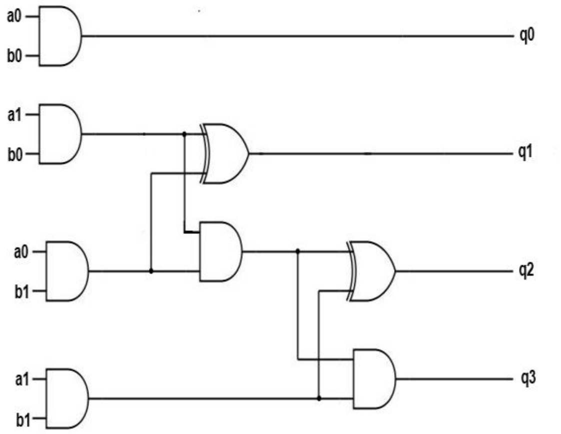


Fig. 3. 2x2 multiplier using conventional logic [17].

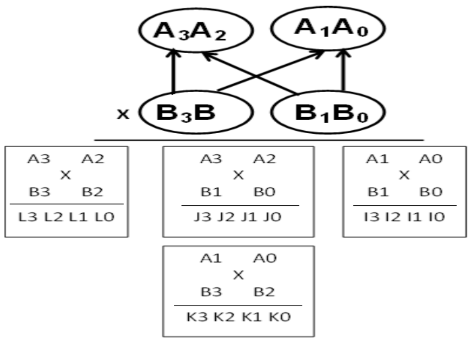


Fig. 4. Procedure of 4x4 multiplication.

IV. Reversible implementation of design

A. 2x2 Multiplier

The 2x2 multiplier is implemented using 4 equations mentioned below [9] derived from Fig. 3.

$$q0 = a0.b0 \tag{4}$$

$$q1 = (a1.b0) \text{ xor } (a0.b1) \tag{5}$$

$$q2 = (a0.a1.b0.b1) \text{ xor } (a1.b1) \tag{6}$$

$$q3 = a0.a1.b0.b1 \tag{7}$$

Reversible implementation [9] is done using a BVPPG gate, three Peres gates and a Feynman gate. BVPPG gate generates two partial products among which, one is Q0. Q1 is obtained from one of the Peres gates and Q2, Q3 are the outputs from Feynman gate. This design needs five reversible logic gates, five constant inputs and generates five garbage outputs. Quantum cost and TRLIC of this implementation are 23 and 38 respectively. In this implementation fan out of every signal including primary inputs is one. Diagram is given in Fig. 5.

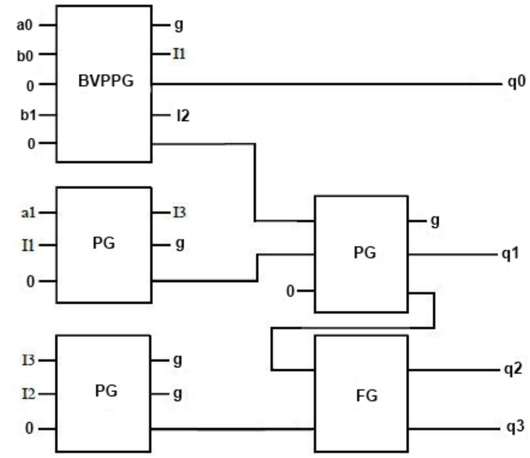


Fig. 5. Reversible implementation of 2x2 Vedic multiplier [9].

B. 4x4 Implementation

Block diagram of 4x4 is shown in Fig. 6. In this block four 2x2 multipliers are arranged systematically. Each multiplier accepts four input bits; two bits from multiplicand and other two bits from multiplier. Addition of partial products are done using two four bit ripple carry adder, a two bit ripple carry adder and a half adder. We obtain the final result by concatenating the last two bits of the first multiplier, four sum bits of the second four bit ripple carry adder and the sum bits of two bit ripple carry adder.

In [9, 10] two four bit and a five bit ripple carry adders are used to add the partial products of 2x2 block. But the block diagram suffers from a serious drawback due to the arrangement of adders. The generated result is accurate only when any of the two operands has '0' in last two left hand side bits i.e. 4 bit operands must be like {0011, 0010}, {0010, 1110}, {1110, 0001} and so on. If any of the third or fourth bit or both the third and fourth bit of any of the operands becomes '1', the result will be erroneous. This is because of the improper way of adding two operands with different place value. We know in addition only the digits having the same place value can be added. The multiplier acts as a 4x2 multiplier. This setback is fixed in our design by using a different block diagram and the number of design parameters is also reduced.

C. Adders

4 bit ripple carry adders consists [9] of three HNG gates and a Peres gate. Peres gate is used as in any addition initial carry is always zero, so we just need a half adder to add first bits of operands. Two bit ripple carry adder needs a HNG gate and a Peres Gate. A Peres gate is used to realize the half adder in 4x4 blocks. Adders are shown in Fig. 7.

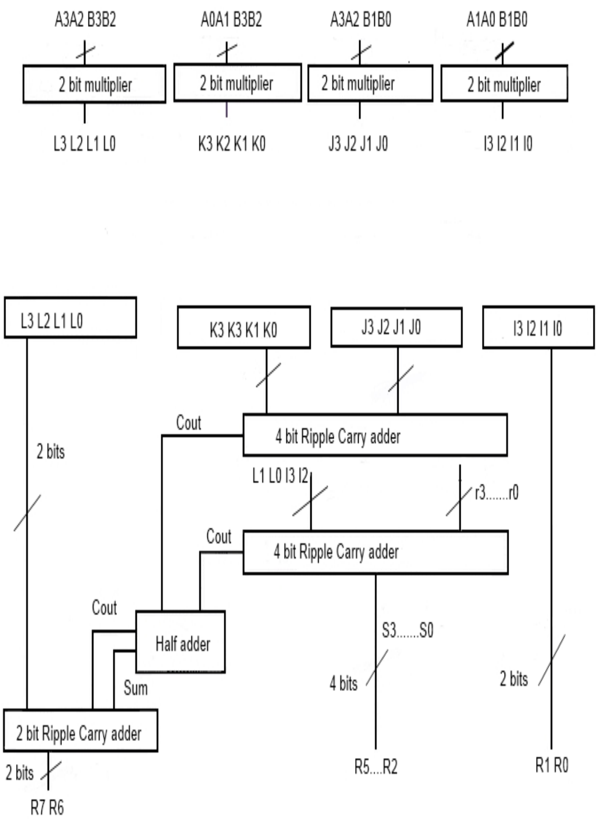


Fig. 6. Block diagram of 4x4 Vedic multiplier.

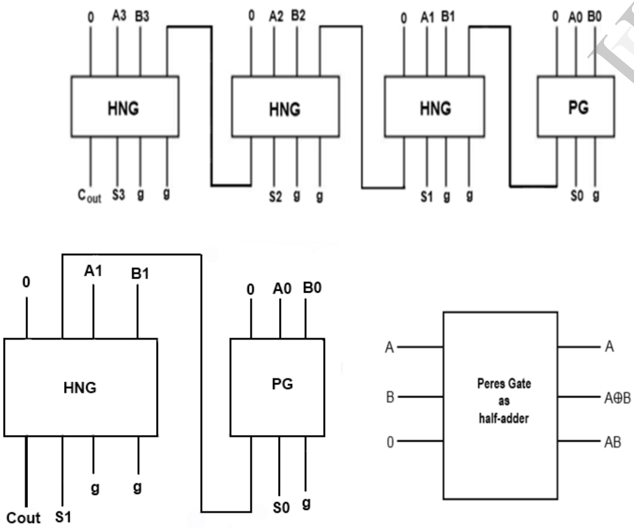


Fig. 7. Diagram of adders.

V. Results and Comparisons

The 2x2 multiplier and 4x4 multiplier are designed in VHDL and functional verification is done using MODELSIM simulator. For 4x4 multiplier design VHDL program of 2x2 multiplier is used. Simulation results are shown in Fig. 8 and Fig. 9. We simulated 2x2 and 4x4 multipliers with various inputs and correct results are obtained.

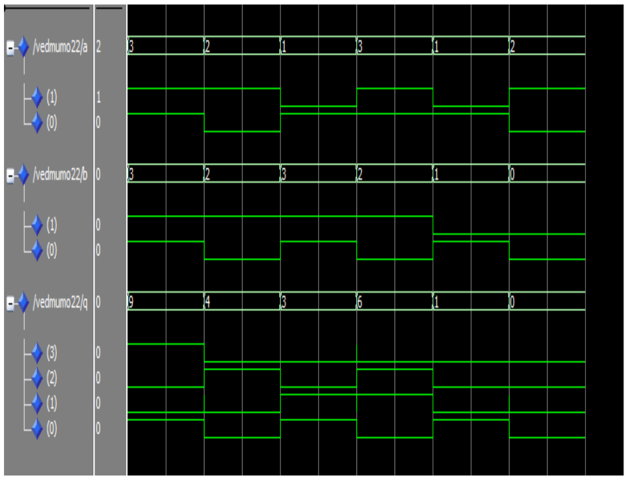


Fig. 8. Simulation result of 2x2 multiplier.

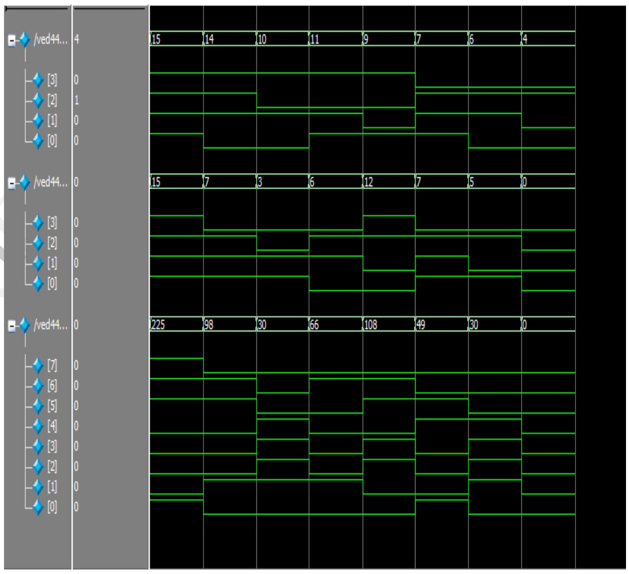


Fig. 9. Simulation result of 4x4 multiplier.

We compare our design with some of the Vedic and non-Vedic reversible 4x4 multiplier in Table I.

From the comparison we can see that our design requires less number of gates compare to other multipliers. Garbage outputs and quantum cost is also less. Constant inputs required is lesser than four other multipliers. Significant reduction in quantum cost and TRLIC is observed. So, we can say our design is optimized as compare to other designs exist in terms of number of gates, constant inputs, garbage outputs, quantum cost, and TRLIC.

TABLE I. COMPARISON OF MULTIPLIER DESIGNS.

4x4 Multiplier	**Performance Parameters**				
	No. of Gates	***Constant Inputs***	***Garbage Outputs***	***Quantum cost***	***TRLIC***
Our work	31	31	38	150	250
Design [9]	33	33	43	164	274
Design [10]	37	29	62	162	290
Design [12]	52	52	52	152	308
Design [13]	52	52	52	168	324
Design [14]	44	56	64	236	400

CONCLUSIONS

In our work, we implemented a 4x4 multiplier using Urdva Tiryakbhayam sutra and four different reversible logic gates. Comparison table establishes that the design is better than other Vedic and non-Vedic multipliers in terms of number of gates used, number of constant inputs, number of garbage outputs, quantum cost and TRLIC. This implementation preserves all the restrictions applicable in reversible logic design and resolves the shortcomings of existing reversible Vedic multiplier. This multiplier can be used in other complex design like squaring circuits, convolution, filters, and other fields like cryptography, communication, nanotechnology etc.

ACKNOWLEDGMENT

We would like to thank our parents, teachers, and friends who supported us and encouraged us during the course of work.

REFERENCES

[1] Jagadguru Swami, Sri Bharati Krisna, Tirthaji Maharaja, “Vedic Mathematics or Sixteen Simple Mathematical Formulae from the Veda, Delhi (1965)”, Motilal Banarsidas, Varanasi, India, 1986.
[2] Vedic Maths India: http://www.vedicmathsindia.org/history-of-high-speed-vedic-maths-2/
[3] R. Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development, 5, pp.183-191, 1961.
[4] C.H. Bennett, “Logical reversibility of Computation”, IBM J. Research and Development, pp.525-532, November 1973.
[5] Michael P. Frank, “Reversibility for Efficient Computing”, Ph. D. Thesis, May 1999. http://www.cise.ufl.edu/~mpf/rc/thesis/phdthesis.html
[6] R. Feynman, “Quantum Mechanical Computers,” Optics News, Vol.11, pp. 11–20, 1985.
[7] A. Peres, Reversible logic and quantum computers, Phys. Rev. A 32 (1985) 3266–3276.
[8] S. Sahni, A. Bakshi, Reversible computing: Looking ahead of the curve. http://www.it.iitb.ac.in/saurabh/documents/revpaper.pdf (Nov. 2004)
[9] Rakshith Saligram, Rakshith T R, "Optimized Reversible Vedic Multipliers for High Speed Low Power Operations", IEEE Intl. Conf. on Info. &Comm. Tech., April 2013.
[10] Rakshith T R, Rakshith Saligram, “Design of High Speed Low Power Multiplier using Reversible logic: a Vedic Mathematical Approach”, Intl. Conf. on Circuit, Power and Computational Technologies [ICCPCT – 2013].
[11] H. Thapliyal and M.B. Srinivas, “Novel Reversible Multiplier Architecture Using Reversible TSG Gate”, Proc. IEEE International Conference on Computer Systems and Applications, pp. 100-103, March 2006.
[12] M. Haghparast, S. Jafarali Jassbi, K. Navi and O. Hashemipour, “Design of a Novel Reversible Multiplier Circuit Using HNG Gate in Nanotechnology”, World Applied Science J.lVol. 3 No. 6, 2008.
[13] M.S. Islam et al., “Low cost quantum realization of reversible multiplier circuit”, Information technology journal, Vol.8, 2009.
[14] Nidhi Syal, Dr. H.P. Sinha, “Design of fault tolerant reversible multiplier”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-6, January 2012.
[15] Rakshith Saligram and Rakshith T.R. "Novel Code Converter Employing Reversible Logic", International Journal of Computer Applications (IJCA), August 2012.
[16] H R Bhagyalakshmi and M K Venkatesha,” Optimized multiplier using Reversible Multi-Control Input Toffoli Gates”, VLSICS, Vol.3. No (6), Dec. – 12.
[17] G. Ganesh Kumara, V. Charishma, “Design of high speed Vedic multiplier using Vedic mathematics techniques”, Itn’l J. of Scientific and Research Publications, Vol. 2 Issue 3 March 2012.
[18] Shamin Akhter, "VHDL implementation of Fast NxN Multiplier Based on Vedic Mathematics", Proc. of IEEE Conference, pp. 472-475, 2007.
[19] Sumit R. Vaidya, D. R. Dandekar, “Performance Comparison of Multipliers for Power-Speed Trade-off in VLSI Design,” RECENT ADVANCES in NETWORKING, VLSI and SIGNAL PROCESSING, ISSN: 1790-5117, ISBN: 978-960-474-162-5.
[20] Dr. S. M. Khairnar, Ms. Sheetal Kapade, Mr. Naresh Ghorpade, “VEDIC MATHEMATICS – THE COSMIC SOFTWARE FOR IMPLEMENTATION OF FAST ALGORITHMS”. http://www.sinhgad.edu/IJCSA-2012/pdfpapers11/4.pdf
[21] Ch. Harish Kumar, “Implementation and Analysis of Power, Area and delay of Array, Urdhva, Nikhilam Vedic Multipliers”, International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.