Design of 32-Bit Multiplier using Reversible Logic Technology based on Urdhva Tiryakbhyam Sutra

Amit Kumar[1] Shri A.G.Rao[2]
[1,2]Department of Electronics Design & Technology
[1,2]NIELIT, Gorakhpur, India

Abstract— In this paper the most significant aspect of the proposed method is that, the developed multiplier architecture is based on vertical and crosswise structure of Ancient Indian Vedic Mathematics. As per this proposed architecture, for two 32-bit numbers; the multiplier and multiplicand, each are grouped as 16-bit numbers so that it decomposes into 16×16 multiplication modules. It is also illustrated that the further hierarchical decomposition of 8×8 modules into 4×4 modules and then 2×2 modules will have a significant Verilog HDL(hardware description language) coding of for 32x32 bits multiplication and it was observed that the parameters like Hardware Complexity, power and Delay are improved over other Reversible multipliers. The design is simulated, synthesized and power estimation was done using 14.3 version Xilinx tools.

Key words: Reversible Logic and Gates, Reversible Multiplier Circuit (RMC), Partial Product, Vedic Mathematics, Urdhva Tiryakbhyam, RLC, Compressor

I. INTRODUCTION

Vedic Mathematics is a system of reasoning and mathematical working based on ancient Indian teaching called Veda [1]. It is fast, efficient and easy to learn and uses all arithmetic and algebraic operations which are accepted by worldwide. Vedic Mathematics, can be implemented both in decimal and binary number. Two most common multiplication algorithms are Array [2] and Booth [3]. The array multipliers take less computation time because the partial products are calculated independently in parallel but delay is large due to the time taken by the signals to propagate through the gates which forms array multiplication. Booth's Algorithms are used for signed multiplication but storing larger partial sum and carry bits, requires huge memory space. Now-a-days, these algorithms are designed at hardware level using low power VLSI design technologies like CMOS, lowering feature size MOS transistors. The huge amount of operand associated with these multiplier unit results in high power consumption and high heat dissipation, because these technologies uses conventional logic gates like AND, NAND, XOR etc, which erased inputs and output bits, every time a logic operations carried out .According to Landauer [3, 4], these conventional logic operations, which are also called as irreversible operation generates $k \cdot T \cdot \ln 2$ joules of heat energy for every bit of information loss where 'k' is the Boltzmann's constant ($1.3807 \times 10^{-23} JK^{-1}$) and '$T$' is the operating temperature ($300\ K$) of the environment. At room temperature the dissipating heat is around $2.8 \times 10^{-21} J$ which is small but not negligible and this causes increase in power consumptions and heat dissipation. Bennett [4] showed that zero energy dissipation would be possible if the computation is carried out by reversible gates only, which does not lose any information and thus power consumption and heat dissipation can be reduced. Now a day's reversible logic gates are finding its applications in optical computing [5], ultra low power CMOS design [6] and nanotechnology [7].To overcome these problems, in this paper a new multiplier architecture is proposed based on Indian Vedic Mathematics algorithm called as Urdhva Tiryakbhyam sutra for high speed multiplication [8], and implemented using Reversible logic gates for low power consumption.

II. REVERSIBLE LOGIC GATES PROPERTIES

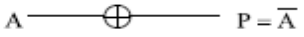
It is an n-input n-output logic function in which there is a one-to-one correspondence between the inputs and the outputs. Because of this injective mapping the input vector can be uniquely determined from the output vector. This prevents the loss of information which is the root cause of power dissipation in irreversible logic circuits. In the design of reversible logic circuits the following points must be considered to achieve an optimized circuit.

They are

- Fan-out is not permitted.
- Loops or feedbacks are not permitted
- Garbage outputs must be minimum
- Minimum delay
- Minimum quantum cost

A. *The NOT Gate*

A NOT gate is a 1x1 gate represented as shown in Fig... Since it is a 1x1 gate, its quantum cost is unity.



B. *Feynman Gate*

Feynman gate is a 2*2 one through reversible gate as shown in figure 1. The input vector is I (A, B) and the output vector is O (P, Q). The outputs are defined by P=A, Q=A B. Quantum cost of a Feynman gate is 1. Feynman Gate (FG) can be used as a copying gate. Since a fan-out is not allowed in reversible logic, this gate is useful for duplication of the required outputs

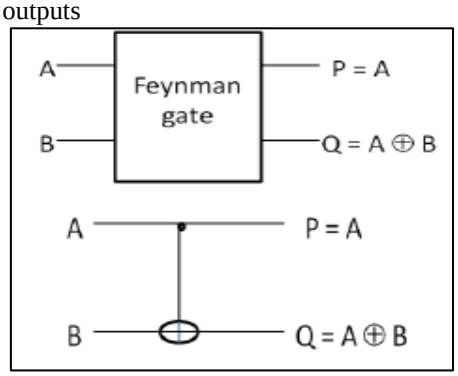


Fig. 1: Feynman gate

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig. 2: Truth table of Feynman gates

C. Toffoli Gate:

Fig 3 shows a 3*3 Toffoli gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The outputs are defined by P=A, Q=B, R=AB C. Quantum cost of a Toffoli gate is 5.

The Toffoli Gate (TG) is a 3x3 two-through reversible gate as shown in Fig. Two-through means two of its outputs are the same as the inputs with the mapping (A, B, C) to (P=A, Q=B, R=A ・B⊕ C), where A, B, C are inputs and P, Q, R are outputs, respectively. The Toffoli gate is one of the most popular reversible gates and has quantum cost of 5The quantum cost of Toffoli gate is 5 as it needs 2V gates, 1 V + gate and 2 CNOT gates to implement it.

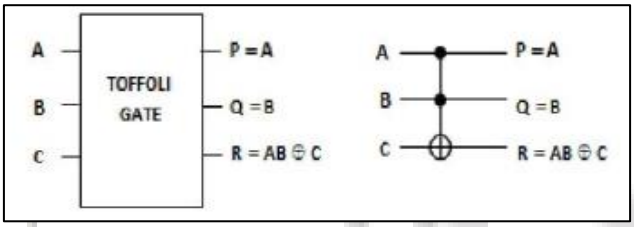


Fig. 3: Toffoli gate

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

Fig. 4: Truth table of Toffoli gate

III. PROPOSED MODEL

Reversible logic plays an important role in recent years due to its ability to reduce the power dissipation which is the main requirement in Low power VLSI design [15]. It is based on the Quantum computing using a physical mechanism that is thermodynamically as well as logically reversible. According to Landauler's research the amount of energy dissipated for every irreversible bit operation is at least kTln2 joules[15], where k=1.3806505*10- 23m2kg-2K1(joule/Kelvin-1).A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there exist one to-one correspondence between its input and outputs[15]. In this paper a basic 4x4 input TSG gate is taken and used as Full Adder[5][6] and the same is used to build 4:2 compressors.[1][2]. Basic gates used in this paper are Peres gates, TSG gates.

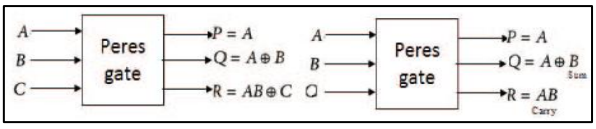


Fig. 5: 3x3 Peres Gate as And gate/Half Adder

In Figure Peres gate is used as AND gate to generate Partial Products by making input c zero and the same can be used as half adder with Output Q as Sum and R as Carry output.

Fig a: Basic TSG gate

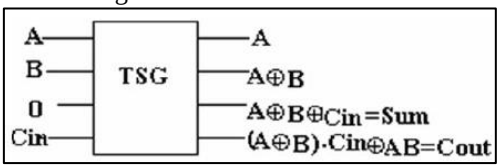


Fig. 6: b: TSG gate as Full Adder

Figure a describes an 4 input TSG reversible logic gate with 4 outputs as shown above and by making input C as '0' it acts as Full Adder

IV. COMPRESSOR

Compressors are used to implement arithmetic and digital signal processing architectures for high performance applications. These are used especially in adder structures to reduce the Complexity and time delay. These are also used in Multiplier architectures to add all partial products and for final addition. In multiplier architectures the main source of power, delay and area consumption are from how these partial products are accumulated. These compressors are used to reduce time delay and increase its speed for specific architecture. Generally compressors reduce N-input bits to a single sum bit of equal weight to that of the inputs and carry out bit. In usage we had 3:2, 4:2, 5:2, etc. In this paper we used only 4:2 compressors with four inputs (x1, x2, x3, x4) and two outputs sum and carry. The 4:2 compressors receive an input Cin from the preceding module of one binary bit order lower in significance, and produce an output Cout to the next compressor module of higher significance as shown in figure. Besides, to accelerate the carry save summation of the partial products, it is imperative that the output, Cout be independent of the input Cin.

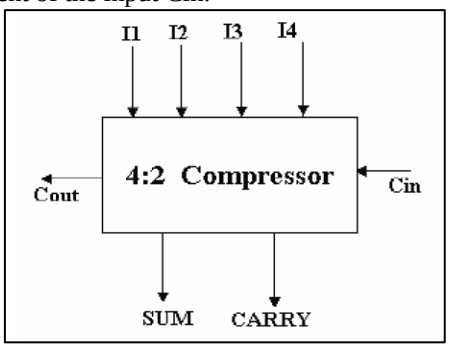


Fig. 7: 4:2 Compressors

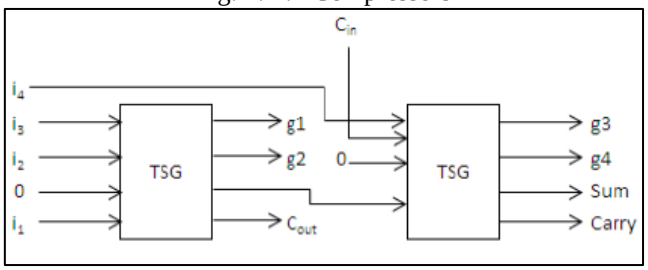


Fig. 8: Reversible 4:2 Compressor using TSG gates

A. Proposed Multiplier

Basically Multiplier consists of 3 stages 1. Partial Product Generation, 2.Partial Product Addition 3. Final Product Addition. Multiplier essentially consists of 2 operands a multiplicand "Y" and Multiplier "X" and produces a product. In stage 1 each bit is multiplied to produce Partial products. Stage 2 is an important stage where all partial products gets added using various adder structures in a tree like fashion [4]. Stage 3 is used to generate the Final Product. Our proposed multiplier uses Peres gates to generate partial products, 4:2 TSG based compressors to add partial products and the same is compared with conventional Multipliers.

B. Partial Product Generation:

To generate Partial Products we used Peres gates because quantum cost per gate is less when compared to other reversible gates [17].In literature to generate partial products they used Fredkin gates or Peres gates [5]. Quantum Cost for Fredkin gate is 5 but for Peres gate it is 4. So we chosen Peres gate to reduce QC. We require 64 gates to generate 64 partial products and the same is shown in figure

C. Partial Product Addition:

This paper combines two different technologies like Compressor logics and Reversible Logics for adding Partial Products on Vedic Multiplier and it was observed that number of stages and number of gates used reduces when compared to other existing structures.

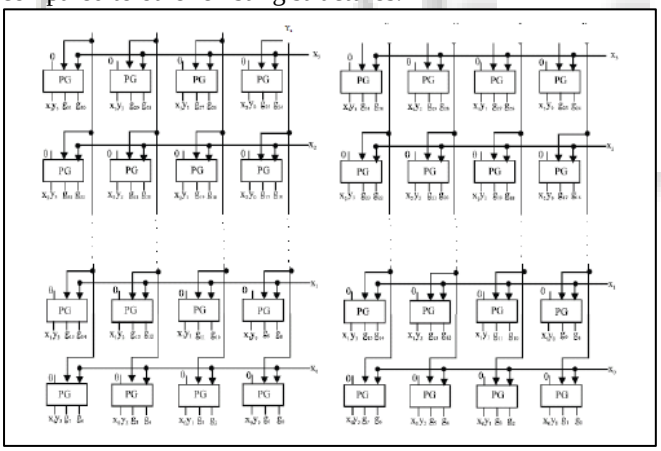


Fig. 9: Partial Production generation using Peres gates.

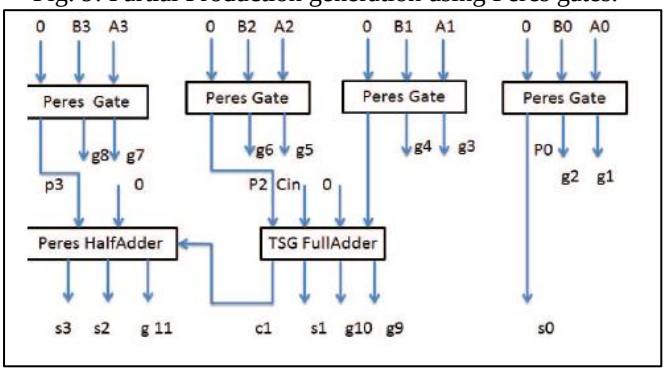


Fig. 10: 2x2 bit Reversible Vedic Multiplier

V. SOFTWARE REQUIREMENTS

Verilog was as soon as soon as begun out in the path of the 300 and sixty 5 days 1984 by the use of Gateway Design Automation Inc as a prohibitive hardware demonstrating vernacular. It can be perceived that the predominant vernacular was as soon as once as rapidly as brought on through taking factors from most commonly almost typically relatively commonly just about essentially the most unmistakable HDL language of the time, almost always called Hilo, and in addition from normal programming lingos, for occasion, C. Round then, Verilog was as soon as once as speedily as not systematized and the tongue modified itself in each one of the most critical priceless transformations that became out inside of 1984 to 1990.

Verilog scan framework throughout the opening used as a contact of 1985 and created unmitigated via 1987. The execution of Verilog scan constitution purchased by way of utilisingutilising Gateway. The main detail bona fide progress of Verilog most definitely will not be geared up to abstain from being Verilog-XL, which joined a pair highlights and utilized the high-high-quality "XL huge kind" which is an exceptionally trained strategy for doing part measure duplication.

Later 1990, Cadence Design process, whose key part round then integrated dainty movie method scan structure, secured Gateway Automation procedure, neighboring unique Gateway matters., Cadence now alternate into the proprietor of the Verilog vernacular, and persevered advancing Verilog as every a dialect and a experiment structure. Then, Synopsys used to be as soon as swiftly as soon as pushing the very first-rate-down setup procedure, utilizing Verilog. This was once as quickly as a gainful mix.

In 1990, Cadence sorted out the Open Verilog world (OVI), and in 1991 gave it the documentation for the Verilog Hardware Description Language. This was as soon as speedily because the celebration which "opened" the vernacular. General standards Hardware Description Language Two issues see a HDL from a straight tongue like "C":

A. RTL Description

– Conversation of Specification in coding format using CAD Tools.

B. Coding Styles:

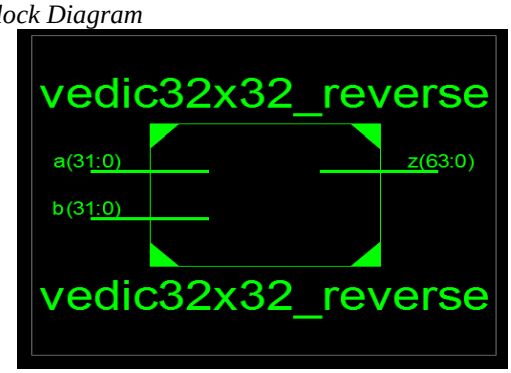
– Gate Level Modeling
– Data Flow Modeling
– Behavioral Modeling

C. RTL Coding Editor:

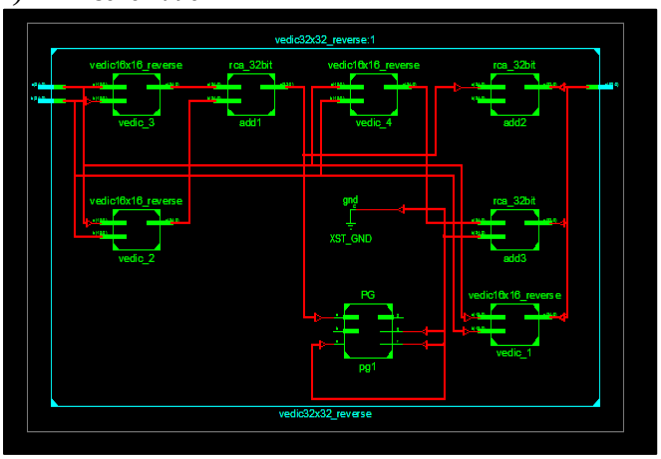
Vim, Emacs, conTEXT, HDL Turbo Writer
Simulation: Modelsim, VCS, Verilog-XL, Xilinx.

D. Simulation Result:

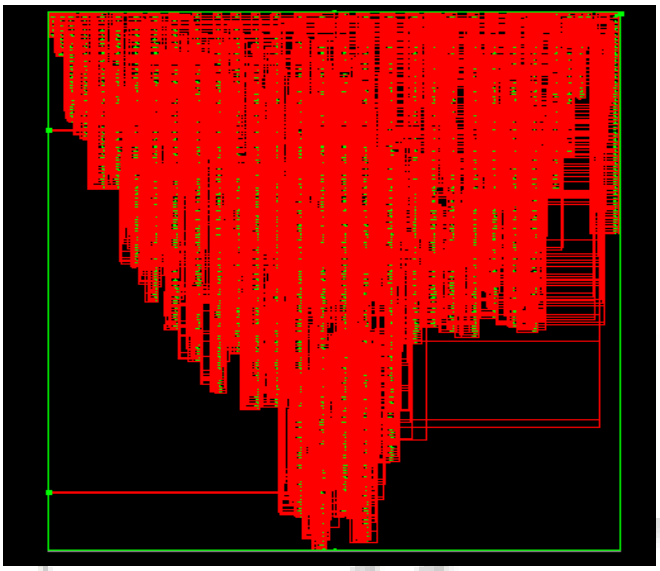
1) Block Diagram



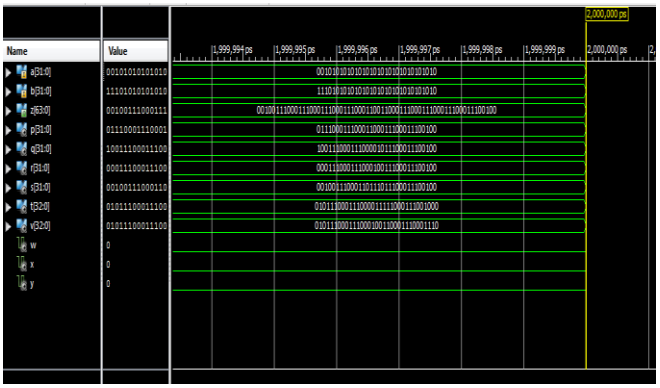
2) RTL schematic



E. Technology schematic



F. Simulation results



1) Proposed (32 bit compressor based reverse Vedic) Area:

vedic32x32_compressor Project Status (07/13/2017 - 18:36:38)			
Project File:	reverse_vedic.xise	**Parser Errors:**	No Errors
Module Name:	vedic32x32_compressor	**Implementation State:**	Synthesized
Target Device:	xc7a100t-3csg324	**• Errors:**	No Errors
Product Version:	ISE 14.3	**• Warnings:**	43 Warnings (0 new)
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	2033	63400	3%
Number of fully used LUT-FF pairs	0	2033	0%
Number of bonded IOBs	128	210	60%

Number of slice LUTs=2033

G. Delay:

```
Delay:               18.662ns (Levels of Logic = 47)
  Source:            a<1> (PAD)
  Destination:       z<63> (PAD)

  Data Path: a<1> to z<63>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ----------
     IBUF:I->O             57   0.001   0.667  a_1_IBUF (a_1_IBUF)
     LUT4:I0->O             2   0.097   0.299  vedic_2/vedic_1/vd1/vd3/gate6/Mxor_q_xo<0>
     LUT6:I5->O             2   0.097   0.688  vedic_2/vedic_1/vd1/add4/Mxor_r_xo<0>1 (ve
     LUT5:I0->O             2   0.097   0.515  vedic_2/vedic_1/vd1/add5/Mxor_s_xo<0>1 (ve
     LUT3:I0->O             2   0.097   0.561  vedic_2/vedic_1/vd1/add7/Mxor_r_xo<0>1 (ve
     LUT6:I2->O             3   0.097   0.521  vedic_2/vedic_1/add2/gate2/Mxor_s_xo<0>1 (
     LUT3:I0->O             3   0.097   0.566  vedic_2/vedic_1/add3/gate1/Mxor_r_xo<0>1 (
     LUT6:I2->O             3   0.097   0.521  vedic_2/vedic_1/add4/gate1/Mxor_r_xo<0>1 (
     LUT6:I3->O             3   0.097   0.305  vedic_2/vedic_1/add5/gate2/Mxor_r_xo<0>31
     LUT5:I4->O             3   0.097   0.521  vedic_2/vedic_1/add6/gate1/Mxor_r_xo<0>1 (
     LUT5:I2->O             3   0.097   0.521  vedic_2/vedic_1/add7/gate2/Mxor_r_xo<0>31
     LUT3:I0->O             2   0.097   0.688  vedic_2/vedic_1/add7/gate2/Mxor_r_xo<0>1 (
     LUT6:I1->O             3   0.097   0.389  vedic_2/add2/gate3/Mxor_s_xo<0>1 (vedic_2/
     LUT5:I3->O             3   0.097   0.389  vedic_2/add2/gate5/Mxor_s_xo<0>1 (vedic_2/
     LUT5:I3->O             3   0.097   0.693  vedic_2/add2/gate7/Mxor_s_xo<0>1 (vedic_2/
```

Delay= 18.662ns

VI. CONCLUSION

Compressor based Vedic Multiplier has been designed using Reversible logics and the functional correctness of the proposed Vedic multiplier. We have proposed novel high speed architecture for multiplication of two 8 bit numbers, combining the advantages of compressor based reverse logic adders and also the ancient Vedic math's methodology. A new 4:2 compressor designed with reversible gates architecture was also discussed. Upon comparison of the area occupied by the multiplier and also its speed, with two other popular multipliers. We can conclude that the compressor based reverse Vedic math's multiplier proves to be a better option over conventional multipliers used in several expeditious and complex VLSI circuits.

REFERENCES

[1] D. Radhakrishnan A.P. PreethySingapore "Low Power CMOS Pass Logic 4-2 Compressor for High-Speed Multiplication", circuits and systems, 2000, Proceedings of the 43rd IEEE Midwest Symposium, pages 1296-1298.

[2] S. F. Hsiao, M.R. Jiang and J.S. Yeh, "Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers," Electronics Letters, vol. 34, no. 4, pp. 341-342, Feb. 1998

[3] M.Margala and N.G. Durdle, "Low-Power Low-Voltage 4-2 Compressors for VLSI Applications," Proc. Workshop on Low Power Design, 1999.

[4] S.Veeramachanemi, K.Krishna, L.Avinash, S.R.Puppola, M.B.Srinivas, "Novel architectures for high speed and low power 3-2, 4-2 and 5-2 compressors", IEEE Proc .Of VLSID'07,pp.324-329,2007.

[5] HimanshuThapliyal and M.B Srinivas "Novel Reversible Multiplier Architecture Using Reversible TSG Gate" Computer Systems & Applications, 2006 IEEE International Conference, pages 100-103.

[6] Md. M. H Azad Khan, "Design of Full-adder With Reversible Gates", International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2002, pp. 515-519.

[7] Hafiz Md. HasanBabu, Md. Rafiqul Islam, Syed MostahedAliChowdhury and Ahsan Raja Chowdhury,"ReversibleLogicSynthesis for Minimization of Full Adder Circuit", Proceedings oftheEuroMicro Symposium on Digital System

Design(DSD'03), 3-5 September 2003, Belek- Antalya, Turkey,pp-50-54.
[8] HimanshuThapliyal and M.B Srinivas "High Speed Efficient NxN Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics"Transactions on Engineering, Computing and Technology, Dec 2004.
[9] JeganathanSriskandarajah, "Secrets of Ancient Maths: Vedic Mathematics", Journal of Indic Studies Foundation, California, pages 15 and 16.
[10] Fundementals and Applications of Vedic mathematics in engineering domains, State Council of Educational Research & Training, New Delhi 2014.
[11] R. Landauer, "Irreversibility and Heat Generation in the Computational Process", IBM Journal of Researchand Development, 5, pp. 183-191, 1961.
[12] Nabila Abdessaied, Robert Wille,, Mathias Soekenand Rolf Drechsler1"Reducing the number of lines in Reversible circuits" , Design Automation Conference, 2010 47th ACM/IEEE pp 647- 652.
[13] MajidHaghparast,SomayyehJafaraliJassbi,KeivanNavi, OmidHashemipour "Design of a Novel Reversible Multiplier Circuit Using HNG Gate in Nanotechnology "World Applied Sciences Journal 3 (6): 974-978,2008.

