

## Reversible logic in pipelined low power vedic multiplier

Ansiya Eshack, S. Krishnakumar

Department of Electronics, School of Technology and Applied Sciences, Edapally, India

---

### Article Info

#### Article history:

Received Jan 29, 2019

Revised Apr 11, 2019

Accepted Jun 28, 2019

---

#### Keywords:

FPGA

Low power

Pipelining

Reversible logic gate

Toffoli gate

Vedic multiplier

---

### ABSTRACT

With an ever growing demand for low-power devices, it is a general trend to search for ways to reduce the power consumption of a system. Multipliers are an important requirement in applications linked to Digital Signal Processing, Communication Systems, Optical Computing, Nanotechnology, Low-Power Very Large Scale Integration and Quantum Computing. Conventional mathematics makes multiplication a very long and time consuming process. The use of Vedic mathematics has led to great reduction in the time required for such calculations. The excessive use of Urdhava Tiryakbhyam sutra in multiplication surely proves its effectiveness and simplicity in this domain. This sutra supports the process of pipelining, a method employed in reduction of the power used by a system. Reversible logic has been gaining demand due to its low-power capabilities and is currently being used in many computing applications. The paper proposes two multiplier systems: one design employs the Urdhava Tiryakbhyam sutra along with pipelining and the second uses reversible logic gates into the first design. These proposed systems provide very less delay for result computation and low hardware utilization when compared to non-pipelined Vedic multipliers.

*Copyright © 2019 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

### Corresponding Author:

Ansiya Eshack,  
Department of Electronics,  
School of Technology and Applied Sciences,  
Edapally, Ernakulam 682024, India.  
Email: ansiya@yahoo.com

---

## 1. INTRODUCTION

The speed of a system mainly depends on the multiplier used and reducing the delay in multiplication greatly increases the throughput of the system [1]. During multiplication, a large number of partial products are generated. While designing a system, it is always required to reduce the computational delay in producing these partial products. To make these multipliers ready for VLSI implementations, it is necessary that they have high-speed, consume low power and occupy less area.

Multiplication consists of 2 steps, first – partial product generation and second – partial product accumulation. Different methods adopted for the processing in these steps tend to vary the time taken to produce the end result. Vedic mathematics specifies 16 sutras (Formulae) and sub-sutras (Sub-formulae) which can be employed in different arithmetic calculations [2]. It is well-known that with the techniques laid out in Vedic mathematics, the calculations are got at a faster rate than conventional mathematical techniques [3]. The Urdhava Tiryakbhyam (UT) sutra is the most commonly used Vedic formula for multiplication. It employs vertical and crosswise multiplication of the digits which are being multiplied [4]. This sutra supports the use of pipelining to sum the partial products. The Vedic multiplier thus consists of a set of partial product generators whose outputs are summed up by adders which work in parallel.

Reversible logic gates are used in circuits for applications in low power consumption [5]. These circuits dissipate zero heat under ideal physical circumstances, as they do not erase information. There is a one to one mapping between the input and output vectors; that is its input and output can be retrieved uniquely [6, 7].

The two main drawbacks of multipliers are its high delay in generating products and excessive power consumption [8]. This paper proposes a multiplier system combining the technique of Vedic multipliers and use of reversible gates. The algorithm of the system is coded in Verilog hardware description language (HDL). Reversible circuits and Vedic multipliers can be designed efficiently using HDL [9, 10]. The proposed multiplier system is implemented on Spartan 3E series of FPGA. It is understood from the results that the combined use of UT sutra and Reversible gates has led to increase in the speed and reduction in the power consumption of the system. The UT sutra requires less number of steps to calculate the product of multiplication and its parallel computation further assists the reduction in the time for output generation [11]. Reversible logic implemented using the Toffoli gate requires low power during its operation [12]. The proposed multiplier system delivers results faster and consumes less power and thus overcomes the two drawbacks of common multipliers.

This paper is organized as follows: Section 2 contains an overview of pipelined Vedic multipliers, reversible logic and the proposed multiplier system. Section 3 highlights the results obtained by the proposed system. Section 4 concludes the work.

## 2. PROPOSED SYSTEM

### 2.1. Pipelined Vedic Multipliers

Calculations employing the techniques from Vedic mathematics reduces the complexity of the working of a multiplier. This greatly decreases the computational time and outputs are attained faster with lesser delays. UT sutra is the technique implemented in the proposed multipliers. The method of vertical and cross-wise multiplication adopted by this sutra for two 2-bit binary numbers is as described in Figure 1.

As can be seen, Step 1 produces the vertical product 1 which can be represented using P1. Step 2 produces two sub-partial binary products 1 and 1, by cross-wise multiplication, & their sum results in binary 10. This is represented by 2-bit register P2(1:0), where P2(0) is the LSB and P2(1) is the MSB or carry. The sub-partial binary product 1 got in Step 3 is added to the carry of P2, represented by P2(1).

This results in a 4-bit binary number 1001 and which is written as

$$P3(1:0) P2(0) P1$$

where  $P3(1:0) = 10$ ,  
 $P2(0) = 0$  and  
 $P1 = 1$ .

Product of multiplication of two 2-bit binary numbers 11 and 11 is 1001.

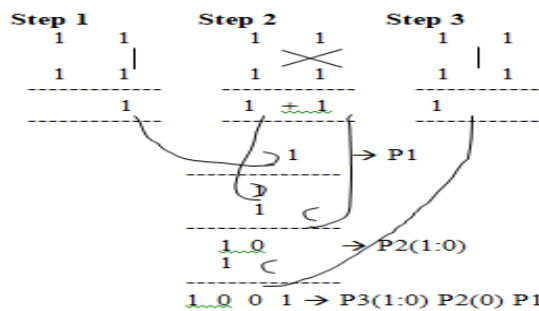
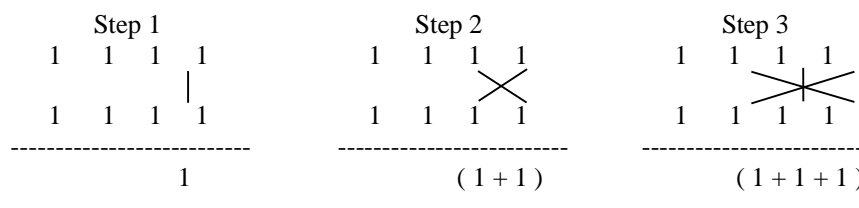
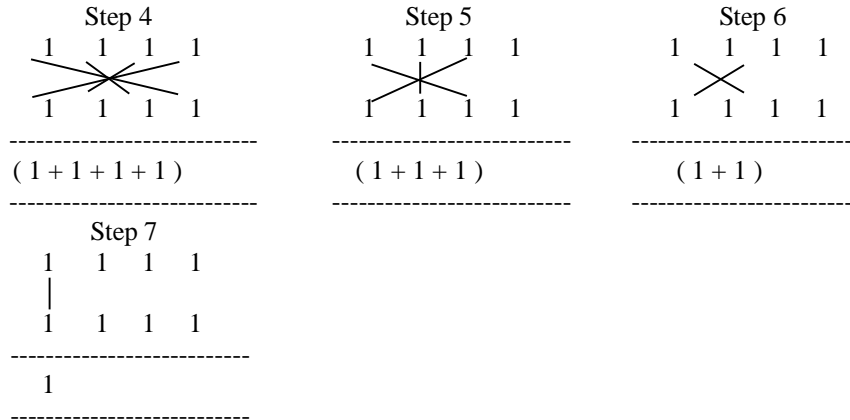


Figure 1. UT sutra for two 2-bit binary numbers

Let us consider the case where two 4-bit binary numbers 1111 and 1111 are multiplied using UT sutra.





The final result of the multiplication is got when the carry of the previous partial products are summed together with the next partial products [11-14]. This is as described as follows:

- Step 1:  $P1 = 1.1 = 1$
- Step 2:  $P2(1:0) = 1.1 + 1.1 = 10$
- Step 3:  $P3(2:0) = 1.1 + 1.1 + 1.1 + P2(1) = 1 + 1 + 1 + 1 = 100$
- Step 4:  $P4(2:0) = 1.1 + 1.1 + 1.1 + 1.1 + P3(1) = 1 + 1 + 1 + 1 + 0 = 100$
- Step 5:  $P5(2:0) = 1.1 + 1.1 + 1.1 + P3(2) + P4(1) = 1 + 1 + 1 + 1 + 0 = 100$
- Step 6:  $P6(1:0) = 1.1 + 1.1 + P4(2) + P5(1) = 1 + 1 + 1 + 0 = 11$
- Step 7:  $P7(1:0) = 1.1 + P5(2) + P6(1) = 1 + 1 + 1 = 11$

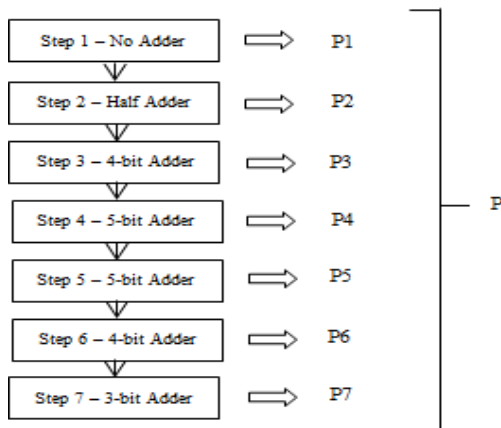
$P2(1) = 1 \ \& \ P2(0) = 0$
$P3(2) = 1, P3(1) = 0 \ \& \ P3(0) = 0$
$P4(2) = 1, P4(1) = 0 \ \& \ P4(0) = 0$
$P5(2) = 1, P5(1) = 0 \ \& \ P5(0) = 0$
$P6(1) = 1 \ \& \ P6(0) = 1$
$P7(1) = 1 \ \& \ P7(0) = 1$

As is seen, the LSB of a step is maintained and all the higher bits are considered as carry and are taken to the next consecutive steps. For example, 100 is the result got in Step 3. The LSB 0 is maintained and MSBs (carry) 1 & 0 are taken to the next two steps, 1 goes to Step 5 and 0 goes to Step 4. In Step 7, the 2-bit result of 11 is maintained and so the final result is an 8-bit binary number:

$$\{P7(1:0)P6(0)P5(0)P4(0)P3(0)P2(0)P1(0)\} = 11100001$$

It can be seen here how the technique of UT sutra is used to generate the partial products. These partial products are got in parallel.

In a 4x4 bit multiplier, two 4-bit binary numbers are multiplied to give an 8-bit output. There are 6 steps where additions take place to produce the output. These additions can be performed in a pipelined manner to compute the final result [15]. It can be understood that the partial outputs are got by addition of the sub-partial outputs produced during single bit multiplication. The addition requires adders and the number of bits added vary in each step.



The partial products P1 to P7 got in each step are added up in parallel. The result of the multiplier is a concatenation of the LSBs got in Step 1 to Step 6 and the partial product got in Step 7 =  $\{P7(1:0) P6(0) P5(0) P4(0) P3(0) P2(0) P1(0)\}$ .

The same concept can be extended to an 8 x 8 bit multiplier which yields a 16-bit product. There are 14 steps where additions are required and performing them in a parallel approach leads to great reduction in the final result computational time of the multiplier.

$$\begin{aligned}
 P1 &= a0b0 \\
 P2 &= a0b1 + a1b0 \\
 P3 &= a0b2 + a1b1 + a2b0 + P2(1) \\
 P4 &= a0b3 + a1b2 + a2b1 + a3b0 + P3(1) \\
 P5 &= a0b4 + a1b3 + a2b2 + a3b1 + a4b0 + P3(2) + P4(1) \\
 P6 &= a0b5 + a1b4 + a2b3 + a3b2 + a4b1 + a5b0 + P4(2) + P5(1) \\
 P7 &= a0b6 + a1b5 + a2b4 + a3b3 + a4b2 + a5b1 + a6b0 + P5(2) + P6(1) \\
 P8 &= a0b7 + a1b6 + a2b5 + a3b4 + a4b3 + a5b2 + a6b1 + a7b0 + P6(2) + P7(1) \\
 P9 &= a1b7 + a2b6 + a3b5 + a4b4 + a5b3 + a6b2 + a7b1 + P6(3) + P7(2) + P8(1) \\
 P10 &= a2b7 + a3b6 + a4b5 + a5b4 + a6b3 + a7b2 + P7(3) + P8(2) + P9(1) \\
 P11 &= a3b7 + a4b6 + a5b5 + a6b4 + a7b3 + P8(3) + P9(2) + P10(1) \\
 P12 &= a4b7 + a5b6 + a6b5 + a7b4 + P9(3) + P10(2) + P11(1) \\
 P13 &= a5b7 + a6b6 + a7b5 + P10(3) + P11(2) + P12(1) \\
 P14 &= a6b7 + a7b6 + P12(2) + P13(1) \\
 P15 &= a7b7 + P13(2) + P14(1)
 \end{aligned}$$

The final result of the multiplier = {P15(1:0) P14(0) P13(0) P12(0) P11(0) P10(0) P9(0) P8(0) P7(0) P6(0) P5(0) P4(0) P3(0) P2(0) P1(0)} which is a concatenation of the LSBs got in Step 1 to Step 14 and the partial product got in Step 15.

Using the same concept, a 16 x 16 bit multiplier is also designed. It has 30 steps containing additions and these are performed in a parallel manner. A 32-bit output is got by the concatenation of the LSBs got in Step 1 to Step 31 and the partial product got in Step 31. The final result of the multiplier = {P31(1:0) P30(0) P29(0) P28(0) P27(0) ... P5(0) P4(0) P3(0) P2(0) P1(0)}.

A 4x4 bit Vedic multiplier is designed to comprise of four 2x2 bit Vedic multipliers. These 2x2 bit multipliers employ the pipelining technique for partial product generation and hence they are now pipelined multipliers. Thus the 4x4 bit multipliers are also now pipelined. The products are got in parallel, from each of the four pipelined 2x 2 bit multipliers. These are partial products and are added in a pipelined manner using the three adders to yield the result of the 4x4 bit pipelined multiplier. Partial product accumulation method is used in these cases. In the system, lower bit multipliers are used to form higher bit multipliers. The proposed system, as shown in Figure 2, thus uses pipelining for partial product generation and accumulation [16].

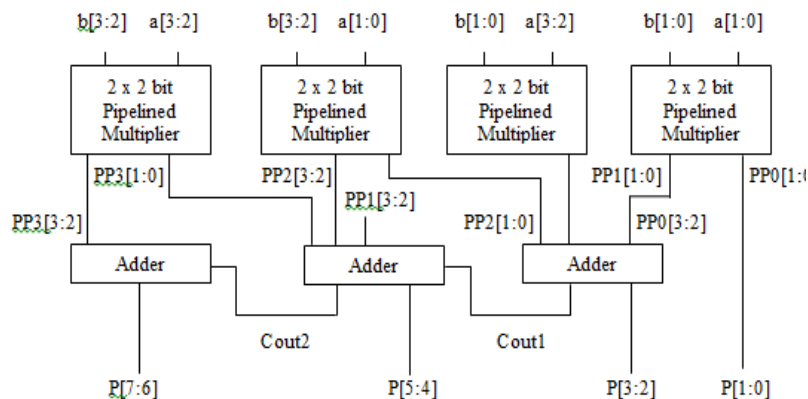


Figure 2. Proposed pipelined 4x4 bit vedic multiplier employing partial product generation and accumulation

PP0, PP1, PP2 and PP3 are the 4-bit outputs of the four 2x2 bit multipliers. They are the partial products and three adders are used for the accumulation of these partial product. The final 8-bit product of the 4x4 bit adder is a concatenation of the outputs of these three adders {P(7:6), P(5:4), P(3:2), P(1:0)}, where:

- Step 1 →  $P(7:6) = PP3(3:2) + \text{Carry of Adder 2}$
- Step 2 →  $P(5:4) = PP3(1:0) + PP2(3:2) + PP1(3:2) + \text{Carry of Adder 1}$
- Step 3 →  $P(3:2) = PP2(1:0) + PP1(1:0) + PP0(3:2)$
- Step 4 →  $P(1:0) = PP0(1:0)$

Similarly a pipelined 8x8 bit Vedic multiplier is designed to consist of four pipelined 4x4 bit Vedic multipliers and a 16x16 bit pipelined multiplier consists of four 8x8 bit pipelined multipliers. The same concept of parallel partial product accumulation is also followed in the 8x8 bit and 16x16 bit multipliers. Hence the use of parallel addition is used in the design of proposed pipelined Vedic multipliers.

**2.2. Reversible Logic**

Presently computing technologies are shifting to reversible computing and this includes the use of reversible logic gates. An m\*m logic gate has m inputs and m outputs. Most commonly used reversible gates are the Feynman gate, Toffoli gate and Fredkin gate [6, 8]. Feynman is a 2\*2 gate while Toffoli and Fredkin are 3\*3 gates. The inputs to the Toffoli and Fredkin gates are A, B and C and the outputs are P, Q and R. Feynman gate has two inputs A and B & outputs P and Q. The equations for these gates are given below.

- Feynman Gate :  $P = A;$   
 $Q = A \oplus B$
- Toffoli Gate:  $P = A;$   
 $Q = B;$   
 $R = AB \oplus C$
- Fredkin Gate:  $P = A;$   
 $Q = B \oplus AB \oplus AC;$   
 $R = C \oplus AB \oplus AC$

Toffoli gate is the reversible gate which is most commonly used [17]. The advantage of this gate is that when two inputs A & B are given with the third input C kept constant at 0, the output R got is always the product of the two inputs given. This makes it suitable for application to the UT sutra and is therefore chosen for multiplication in the proposed system. The truth table of Toffoli gate is shown in Table 1.

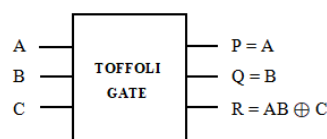
Table 2 shows only the output R and can be observed that it is the product of inputs A and B when the third input C is 0. This makes the Toffoli gate suitable to be used for 1x1 bit multiplication. Using this property, 2x2 bit, 4x4 bit, 8x8 bit and 16x16 bit reversible multipliers are designed for the proposed system. Figure 3 (a) gives the circuit representation and (b) the functionality of the Toffoli gate.

Table 1. Truth Table of Toffoli Gate

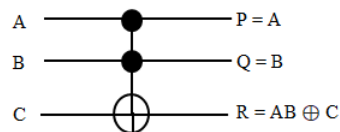
Inputs			Outputs		
A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 2. Truth Table of Toffoli Gate with third input C = 0

A	B	C	R
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1



(a)



(b)

Figure 3. (a) Circuit representation of Toffoli gate (b) Diagram of its functionality

A 2-bit multiplier employing the Toffoli gate is shown in Figure 4.  $x_1x_0$  and  $y_1y_0$  are the two 2-bit numbers we have chosen for multiplication. In all the gates, the third input is maintained at 0, so that the output got is the product of the other two inputs to the gate.  $P_{00}$  is the product got by multiplying the LSBs  $x_0$  and  $y_0$ ,  $P_{01}$  is the product of  $x_0$  and  $y_1$ ,  $P_{10}$  is product of  $x_1$  and  $y_0$  and  $P_{11}$  is the product of  $x_1$  and  $y_1$ .

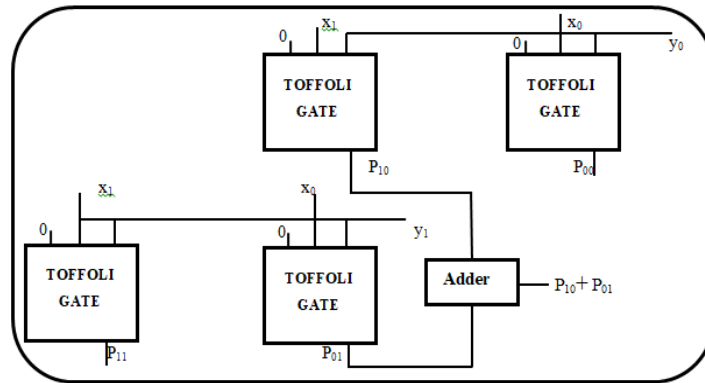


Figure 4. 2x2-bit binary multiplier using toffoli gate

Step 1:  $P_1 \rightarrow P_{00}$

Step 2:  $P_2 \rightarrow P_{10} + P_{01}$

Step 3:  $P_3 \rightarrow P_{11} + \text{Carry of } P_2$

Product of the multiplier: (Carry of  $P_3$ ) (LSB of  $P_3$ ) (LSB of  $P_2$ ) ( $P_1$ )

Four of such 2x2 bit reversible pipelined multipliers are used to create a 4x4 bit reversible pipelined multiplier. Also 8x8-bit and 16x16 bit reversible pipelined multipliers are designed using four 4x4 bit and 8x8 bit pipelined multipliers, employing reversible gates for multiplication.

### 3. RESULTS AND ANALYSIS

The delay of the proposed system employing reversible gates is seen to be less than the delays of a non-pipelined multiplier and pipelined multiplier employing UT sutra. Figure 5, shows a comparison between the delays of the three multiplier systems.

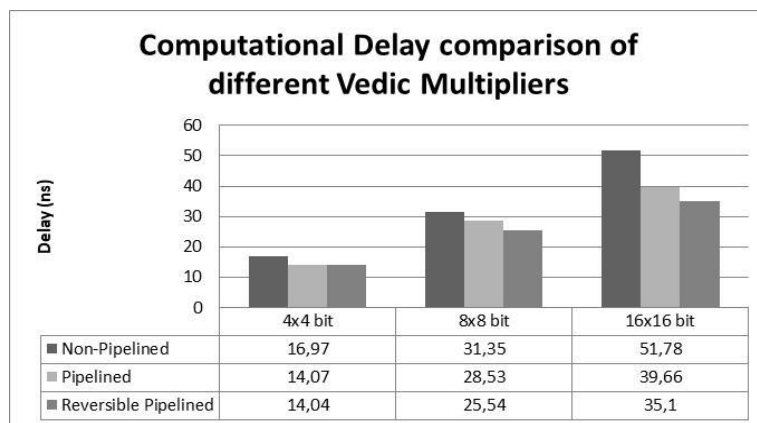


Figure 5. Computational delay comparison of different vedic multipliers

It is observed that the speed of the pipelined Vedic multiplier is greater than its non-pipelined version due to the introduction of the pipelining concept in partial product generation and accumulation. The use of Toffoli gates into these pipelined Vedic multipliers has reduced the computational delay further,

due to the inherent reversible capability of the gate. From Figure 5, it can be noted that the delay for the computation of final result, in the 4x4 bit pipelined multiplier has a 17.08% reduction compared to that of the 4x4 bit non-pipelined counterpart. There is an 8.99% delay reduction in the 8x8 bit pipelined compared to the non-pipelined 8x8 bit multiplier. In the case of 16x16 bit multipliers, the delay reduction is 23.40%. When the pipelined and reversible pipelined multipliers are compared, there is a reduction in delay of 0.21% in the 4x4 bit, 10.48% in the 8x8 bit and 11.49% in the 16x16 bit multipliers, as shown in Table 3.

Table 3. Hardware Utilization Comparison of Different Vedic Multipliers

Type of Vedic Multiplier	4x4 bit		8x8 bit		16x16 bit	
	No. of LUTs	No. of Slices	No. of LUTs	No. of Slices	No. of LUTs	No. of Slices
Non Pipelined	42	22	201	106	841	445
Pipelined	33	17	174	98	722	409
Reversible Pipelined	32	17	171	94	705	390

The proposed system is implemented on Spartan 3E series of FPGA. The hardware on a Field Programmable Gate Array (FPGA) chip is indicated in terms of the number of slices and these slices comprise of the LUTs. Depending on the family of the FPGA chip, the number of LUTs on a slice varies. Comparing the non-pipelined and pipelined Vedic multipliers, 4x4 bit multipliers show a decrease of 21.42% in the number of LUTs and 22.72% in the number of slices used to implement the pipelined design. Similarly in 8x8 bit there is a reduction of 13.43% in the LUTs and 7.54% in the slices used. In 16x16 bit, the number of LUTs used has decreased by 14.14% and the number of slices by 8.08%. The reversible pipelined Vedic multipliers proposed show further reduction in the hardware utilization of FPGA when compared with that of pipelined Vedic multipliers. The percent reduction in the number of LUTs in 4x4 bit are 3.03%. In 8x8 bit, the decrease in number of LUTs and slices are 1.72% and 4.08%. In 16x16 bit, the percent change is 2.35% for LUTs and 4.64% for slices.

In a VLSI design, the important characteristics considered are power, delay and area. For low power consumption of the proposed multiplier, the focus was mainly on reducing the computational delay and the hardware utilization on the FPGA. It is seen that the proposed system employing the UT sutra and the reversible Toffoli gate has greatly increased the throughput of the system and reduced the hardware utilized. The UT sutra uses pipelining technique for addition in the partial products generation & accumulation and so results in faster computation. Further, the reduction in hardware utilization makes the system low-power consuming.

#### 4. CONCLUSION

The paper has proposed two multiplier designs; one design employing pipelining in UT sutra and the other design which employs reversible logic concept in the pipelined multiplier. Vedic concept combined together with pipelining and use of reversible gates in the proposed systems shows great effectiveness in reducing the delay in output calculation and the power consumption of the multiplier system. The delays of 16x16 bit, 8x8 bit and 4x4 bit non-pipelined Vedic multipliers are 51.78 ns, 31.5 ns and 16.97 ns, while the delays of the reversible pipelined Vedic multipliers in the same order have reduced to 35.10 ns, 25.54 ns and 14.04 ns. The LUTs used in 16x16 bit, 8x8 bit and 4x4 bit non-pipelined Vedic multipliers are 42, 201 and 841 while the same for reversible pipelined Vedic multipliers are 32, 171 and 705. The numbers of slices used by 16x16 bit, 8x8 bit and 4x4 bit non-pipelined Vedic multipliers are 22, 106 and 445 while that of reversible pipelined Vedic multipliers are 17, 94 and 390. The reduction in the LUTs and slices used result in low-power consumption of the proposed multiplier systems.

#### REFERENCES

- [1] P. Koti Lakshmi, B Santhosh Kumar and Prof.Rameshwar Rao, "Implementation of Vedic Multiplier Using Reversible Gates," *Fifth International Conference on Advances in Computing and Information Technology (ACITY 2015)*, Chennai, India, 2015, pp. 125-134
- [2] H. D. Tiwari, *et al.*, "Multiplier design based on ancient Indian Vedic Mathematics," *International SoC Design Conference*, 2008, pp. II-65-II-68.
- [3] Salah Alkurwy, "A novel approach of multiplier design based on BCD decoder," *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 14, No. 1, April 2019, pp. 38-44
- [4] K. N. Vijeyakumar, S. Elango and S. Kalaiselvi, "VLSI Implementation of High Speed Energy-Efficient Truncated Multiplier", *Journal of Circuits, Systems and Computers*, Vol. 27, No. 05, 2018, 1850077

- [5] A. S. M. Sayem and S. K. Mitra, "Efficient approach to design low power reversible logic blocks for Field Programmable Gate Arrays," *2011 IEEE International Conference on Computer Science and Automation Engineering*, Shanghai, 2011, pp. 251 – 255.
- [6] Majid Haghparast, *et al.*, "Optimized Reversible Multiplier Circuit", *Journal of Circuits, Systems, and Computers*, Vol. 18, No. 2, 2009, pp. 311 – 323.
- [7] H. V. Jayasree, Sharan Patil and V. K Agrawal, "Design Approaches for Resource and Performance Optimization of Reversible BCD Addition and Unified BCD Addition/Subtraction Circuits", *Journal of Circuits, Systems and Computers*, Vol. 27, No.3, 2018, pp. 1850048
- [8] Chaitanya CVS, *et al.*, "Design of modified booth based multiplier with carry pre-computation," *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 13, No. 3, March 2019, pp. 1048~1055
- [9] Z. Al-Wardi, R. Wille, R. Drechsler, "Towards VHDL-Based Design of Reversible Circuits," *Reversible Computation*, 2017, Lecture Notes in Computer Science, Vol. 10301, Springer, Cham.
- [10] Devika Jaina, Kabiraj Sethi and Rutupama Panda, "Vedic Mathematics Based Multiply Accumulate Unit," *International Conference on Computational Intelligence and Communication Networks*, Gwalior, 2011, pp. 754-757
- [11] Vijaya S. M., Suresh K, "An efficient design approach of ROI based DWT using vedic and wallace tree multiplier on FPGA" *International Journal of Electrical and Computer Engineering*, Vol. 9, No. 4, August 2019, pp. 2433 – 2442
- [12] S.M.R. Taha, "Reversible Logic Synthesis Methodologies with Application to Quantum Computing", *Studies in Systems, Decision and Control*, (Springer International Publishing, Switzerland, 2016), pp. 7 – 16
- [13] S. Vaidya and D. Dandekar, "Delay-Power Performance Comparison of Multipliers in VLSI Circuit Design," *International Journal of Computer Networks & Communication*, Vol.2, No.4, 2010
- [14] D. K. Kundu, *et al.*, "Implementation of optimized high performance 4×4 multiplier using ancient Vedic sutra in 45 nm technology," *2nd International Conference on Devices, Circuits and Systems (ICDCS)*, Coimbatore, India, 2014, pp. 1 – 6.
- [15] Huddar S.R., *et al.*, "Area and Speed Efficient Arithmetic Logic Unit Design Using Ancient Vedic Mathematics on FPGA". *Advances in Computing, Communication, and Control. ICAC3 2013. Communications in Computer and Information Science*, Vol 361. Springer, Berlin, Heidelberg
- [16] A. Eshack and S. Krishnakumar, "Implementation of Pipelined Low Power Vedic Multiplier," *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2018, pp. 171-174.
- [17] A. Eshack and S. Krishnakumar (2018) Design of low-power Vedic Multipliers using Pipelining Technology In: *Computational Intelligence and Informatics. ICCII 2018. Advances in Intelligent Systems and Computing*, Springer, Berlin, Heidelberg (in press)
- [18] R. Drechsler and R. Wille, "Reversible Circuits: Recent Accomplishments and Future Challenges for an Emerging Technology (Invited Paper)," *H. Rahaman et al. (Eds.): VDAT 2012, LNCS 7373*, pp. 383–392, 2012, Springer-Verlag Berlin Heidelberg 2012

## BIOGRAPHIES OF AUTHORS



Ansiya Eshack obtained B. Tech in Electronics & Communication Engineering from MES College of Engineering, Kerala & M. Tech in VLSI & Embedded Systems from Model Engineering College, Kerala. Presently she is working as Associate Professor in Electronics & Communication Engineering Department of KMEA Engineering College, Ernakulam, Kerala. She has a total of 15 years of teaching experience. She is presently pursuing PhD at M. G. University, Kottayam, Kerala. She is a Life member of ISTE & IETE. She was the Student Activities Chair of IEEE Kerala Section. Her areas of interest are Low Power VLSI Design, Embedded Systems, Communication Systems and DSP Applications.



S. Krishnakumar completed his M.Sc. in Physics with Electronics specialization in 1987 from Mahatma Gandhi University, Kottayam. He was awarded Ph.D. in Thin Film Devices in 1995 from Mahatma Gandhi University, Kottayam. He got M.Tech. in Computer Science from Allahabad Agricultural Institute – Deemed University (renamed as Sam Higginbottom Institute of Agriculture, Technology and Sciences) in 2006 and also obtained MCA from IGNOU in 2010. He qualified AICTE-UGC special examination for PG Engineering in 2018. He has 24 years of teaching experience in Electronics and Computer Science subjects for graduates and post graduates. He served as the Regional Director at the School of Technology and Applied Sciences (STAS), during the period 2013-17. Currently he is working in the STAS, Mahatma Gandhi University Research Centre, Edapally, Kochi-24, Kerala, India. His research interest fields include ANN, Analog circuit design and Image processing. Dr. S. Krishnakumar is an Associate member of Institute of Engineers, India. He was a member of Board of studies of University of Calicut and a member of Academic council of Mahatma Gandhi University, Kottayam.