IMPLEMENTATION OF 16x16 BIT MULTIPLICATION ALGORITHM BY USING VEDIC MATHEMATICS OVER BOOTH ALGORITHM

Pranita Soni[1], Swapnil Kadam[2], Harish Dhurape[3], Nikhil Gulavani[4]

[1]*Department of E&TC, Sinhgad Academy of Engineering, Pune, Maharashtra, India*
[2]*Department of E&TC, Sinhgad Academy of Engineering, Pune, Maharashtra, India*
[3]*Department of E&TC, Sinhgad Academy of Engineering, Pune, Maharashtra, India*
[4]*Department of E&TC, Sinhgad Academy of Engineering, Pune, Maharashtra, India*

Abstract

Multiplication is one of the important operation in digital signal processors. The speed of processor depends on the hardware architecture, delay and power. Previously implemented Booths algorithm is not very competent in terms of delay and hardware complexity, therefore we have implemented multiplication algorithm which is efficient over booths algorithm. For multiplication, as number of bit increases the respective delay increases. For efficient processors, delay should be minimum, to avoid increase in delay we require minimum hardware architecture for the processor and Vedic multiplier has minimum hardware architecture.

In this paper we have explained 16 bit 'Urdhva Tiryagbhyam' Vedic Multiplier and Booth Multiplier and compared on the various parameters such as speed, delay, hardware complexity. These algorithms are executed in VHDL language by using model sim and synthesis is done in Xilinx software. Spartan 3 family FPGA development board is used for hardware implementation of these algorithms.

Keywords: *Urdhva Tiryagbhyam, Booth, Vedic Multiplier, Spartan 3.*

----------***----------

1. INTRODUCTION

Multiplication is one of the fundamental function in arithmetic operations. In a Digital Signal Processors multipliers are the key components. Since multiplication dominates the execution time of many DSP algorithms, so there is need of high speed multiplier. The demand of high speed processing is increasing in computer and DSP applications. Efficient arithmetic operations are important to achieve the desired performance in many real-time signals.

Hence continuous efforts are taken to improve the performance of multiplier by reducing the speed. There are several methods have been used for multipliers such as Braun-array, Baugh-Wooley method of two's complement and Booths algorithm. Booth's algorithm is the most successful method for multiplication.

In this paper a simple 16X16 bit multiplier is proposed which is based on Urdhva Tiryagbhyam sutra of Vedic mathematics and on Booth's algorithm. The two numbers of 16 bit each are multiplied by using Vedic maths. The main concept behind this is that to reduce the propagation delay of the architecture which is the drawback of Booth's algorithm. As the number of bit increases in the Booth's algorithm delay increases noticeably and for Vedic mathematics as number of bit increases delay increases slightly. In the end we have compared the two algorithms on the basis of certain parameters and proved that Vedic mathematics is the best way to multiplication of higher number of bits like 16X16 bit.

2. BOOTHS MULTIPLIER

The previous method for multiplier is the Booth's multiplier [5] based on Booth's algorithm. This method is given by "Andrew Donald Booth". The given flowchart describes the method.

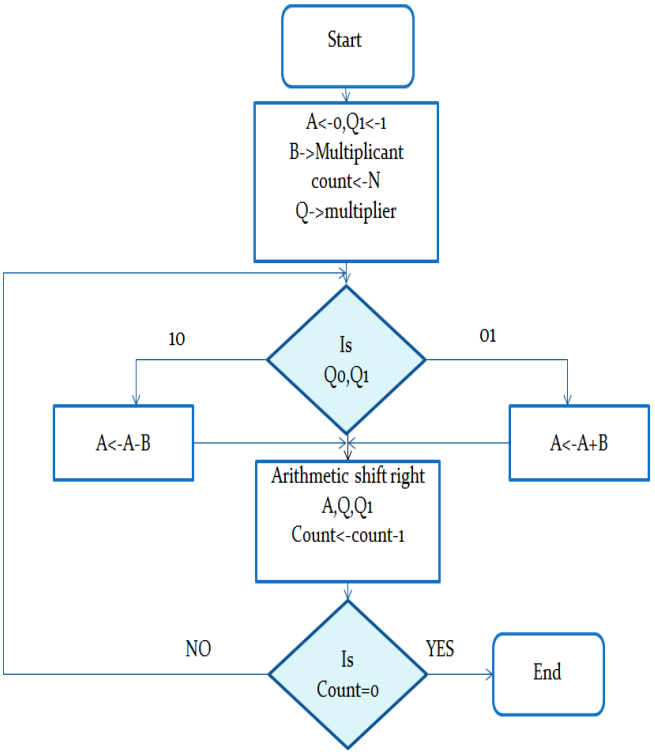

Fig 1: Flow chart for Booths Algorithm

The Booth's multiplier scans the three bits at a time to reduce the number of partial products. These three bits are the two bits from the present pair and third bit from high order bit of an adjacent lower order pair.

From above flow chart we have algorithm as follows:

1. The multiplier and multiplicand are placed in Q and B registers respectively.
2. A one bit Q(-1) register is placed at the least significant bit of Q(0) of register Q.
3. The final result appears in A.
4. A and Q(-1) registers are initialized to 0.
5. Multiplication of no is one in n cycle.
6. In each cycle Q(0) and Q(-1) are examined.
   i. If Q(0) and Q(-1) are (1-1 or 0-0) then all the bits of A ,Q and Q(-1) registers are shifted right by 1 bit.
   ii. If Q(0) and Q(-1) are 01 then multiplicand is added with A. After addition A Q and Q(-1) registers are shifted right by 1 bit.
   iii. If Q(0) and Q(-1) are 10 then multiplicand is subtracted from A . After subtraction A Q and Q(-1) registers are shifted to right by 1-bit.
7. The final result appears in A.

In earlier days this Booth's multiplier is very suitable for multiplication. This will give the accurate result of multiplication. But this multiplier has some drawback.

The drawbacks can be overcome by designing a suitable multiplier for 16X16 bit multiplication. We will design the new multiplier as Vedic multiplier by using the concept of Vedic mathematics.

3. VEDIC ALGORITHM

Vedic multiplier is based on the Vedic mathematics [1]. Vedic mathematics is the ancient method of mathematics. It was rediscovered from Vedas in between 1911 to 1918 by Sri Bharti Krishna Tirthaji (1884-1960). Who was mathematician and Sanskrit scholar. The Vedic mathematics is based on the 16 sutras of Urdhva Tiryagbhyam. It simply means that "vertically and crosswise multiplication".it involves minimum number of calculations, it reduces the space, save the computational time and it is applicable in all cases of multiplication. This method is most efficient when the number of bit increases in multiplication.

The structure of this method is shown in the figure from this we get clear idea of "vertical and crosswise multiplication"

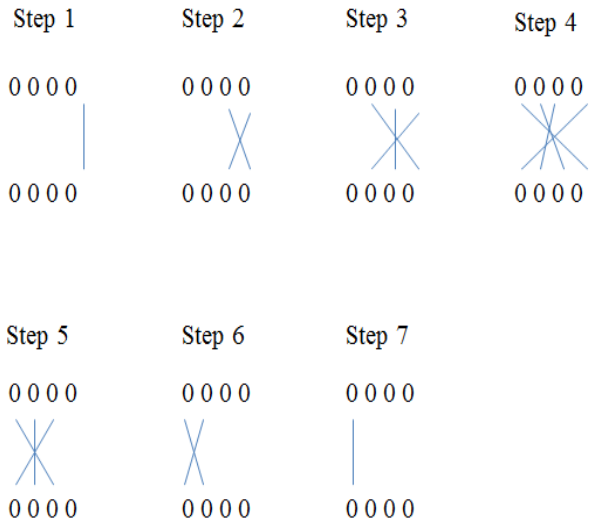

Fig 2: Vedic method for multiplication of 4 bit binary Number

For multiplication of two 4 bit numbers the multiplication process is divided into 7 steps. The first number is a3a2a1a0 which is multiplier and second number is b3b2b1b0 which is multiplicand as shown in the step1 of the figure. The LSB (least significant bit) of multiplier is multiplied with LSB of the multiplicand and the result of this multiplication is stored as the LSB of the final result. Then as shown in the step2 of the above figure The LSB of multiplier is multiplied with second higher bit of multiplicand and second higher bit of multiplier is multiplied with LSB of multiplicand and then these two partial products are added, after adding these two numbers the LSB of addition is taken as the second higher bit of the final result and remaining bits of the addition are taken as carry bit and this carry can be of multi-bit. Then follow the steps which are given in the above figure and after that we get the result of the 4 bit multiplication.

In this paper we have done 16 bit multiplication and for that we require different modules of the 2 bit, 4 bit, 8 bit Vedic multiplier. The different modules of multiplier and their architectures are explained below.

4. THE PROPOSED MULTIPLIER ARCHITECTURE

This method can be used for any N X N bit multiplication and this Vedic multiplier is independent on clock frequency because partial product and their sum calculated in parallel. And because of we don't require high clock frequencies for multiplication and that's why less switching takes place, because of less switching delay and power minimizes and the result of this we get an efficient processor in delay and power. In this paper we are presenting 16 bit multiplication and for that we require 2 bit, 4 bit, 8 bit multiplier. The detail explanation of these architectures is given below.

4.1 Vedic Multiplier for 2X2 Bit Module

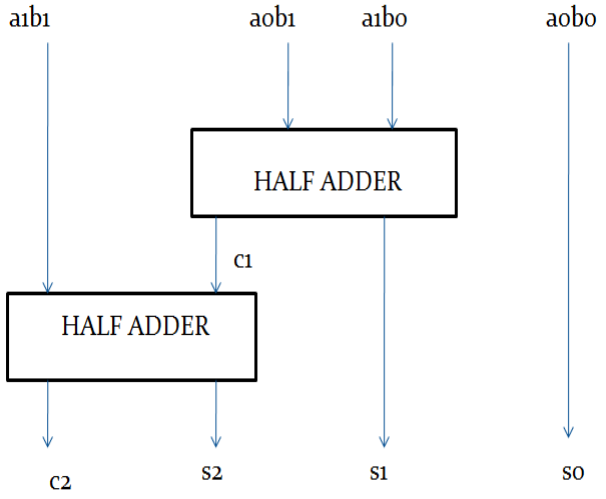

Fig 3: Vedic multiplier for 2X2 bit

A basic block diagram of 2X2 multiplier is as shown. In this diagram a0, a1 are the bits of first digit and b0, b1 are the bits of second digit. In this multiplier we are taking AND of respective bits and then as shown in block diagram we are following the procedure. The multiplication is done on the sutra of Urdhva-Tiryagbhyam. The result obtained is of 4 bits.

4.2 Vedic Multiplier for 4X4 Bit Module

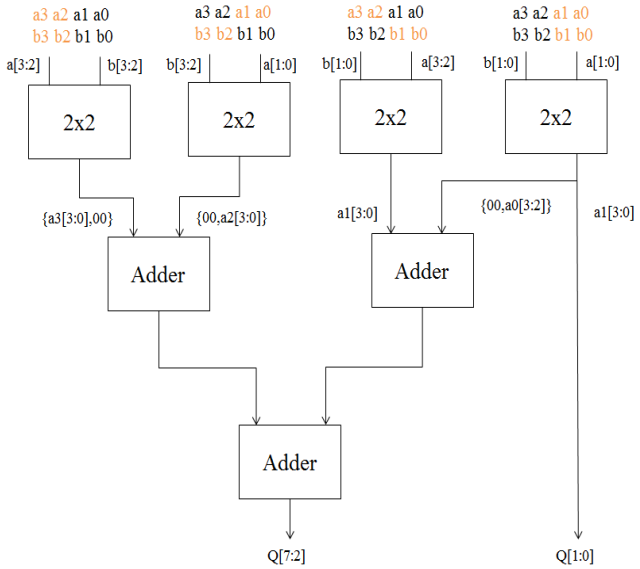

Fig 4: Vedic multiplier for 4X4 bit

After implementation of basic 2X2 multiplier it is easy to implement 4X4 multiplier. In the diagram a0, a1, a2, a3 are the bits of first digit while b0, b1, b2, b3 are the bits of second digit. For 4X4 bit multiplication we require four 2X2 multiplier, three adders. After doing the procedure as shown in figure 4 we can implement the 4X4 Vedic multiplier. The result obtain is of 8 bits.

4.3 Vedic Multiplier for 8X8 Bit Module

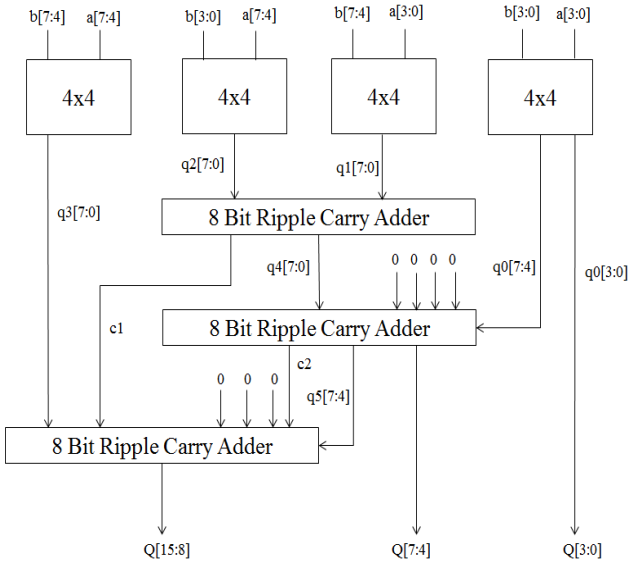

Fig 5: Vedic multiplier for 8X8 bit

This multiplier is implemented from the previous 4X4 multiplier. In this diagram a0 to a7 are the bits of first digit and b0 to b7 are the bits of second digit. For 8X8 multiplier we require four 4X4 multiplier and three 8 bit ripple carry adder after doing the procedure as shown in figure 5 we will get the result of 8X8 multiplier and the result obtained is of 16 bits.

4.4 Vedic Multiplier for 16X16 Bit Module

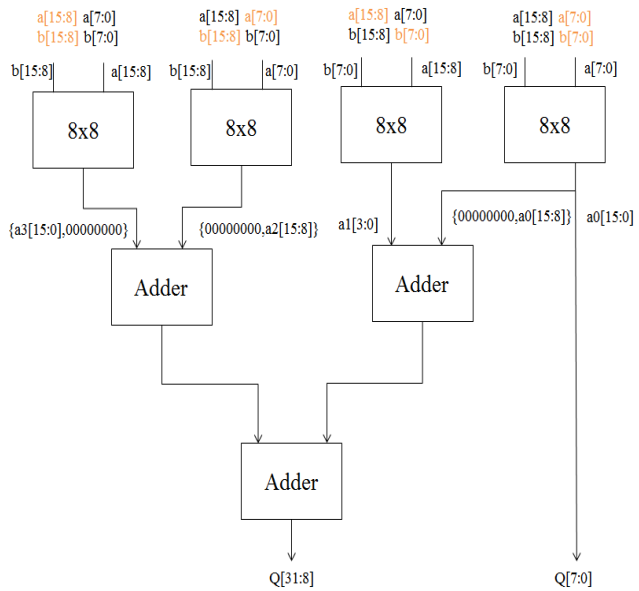

Fig 6: Vedic multiplier for 16X16 bit

This is the last module of the Vedic multiplier, this multiplier is implemented from the 8X8 multiplier. In this diagram a0 to a15 are the bits of first digit and b0 to b15 are the bits of second digit. For 16X16 multiplier we require four 8X8 multiplier and three adders after doing the procedure as shown in figure 6, we will get the result of 16X16 multiplier and the result obtained is of 32 bits.

5. IMPLEMENTATION ON XILINX SOFTWARE

Now we are moving towards main programming part. Using the Vedic Multiplier we have to develop a program. In this work the 16X16 bit multiplier is designed in VHDL (very high speed integrated circuits hardware description language). Synthesis and simulation was done in XILINX ISE 10.1. [6] Project navigator and simulator integrated in the XILINX package. XILINX is software which is purely based on the VHDL language. VHDL is Very large scale integrated circuit Hardware Description Language. We have performed our programming using the Spartan-3 family, Package is selected as PQ208 and Speed grade is -5.

Integrated Software Environment is a software tool produced by Xilinx for synthesis and analysis of HDL design, enabling the developer to synthesize (compile) their design, perform time analysis, examine RTL diagrams, simulate a design reaction to different stimuli, and configure the target device with the programmer. It includes ISE simulator which we can use for functionality verification of program.

The programming is divided into 4 parts. For developing the multiplier of 16X16 bit, we have to first develop the multiplier of 2X2, 4X4 and 8X8.

6. HARDWARE IMPLEMENTATION

After the programming is done in the XILINX software the hardware part comes into account. To see easily the output of multiplication we have executed the output on FPGA (Field Programmable Gate Array) kit. The kit is having many LED's which we are using for input and output purpose.

For downloading the program we require to join the JTAG (Joint Test Action Group) cable. One port is attached to the kit and the other port is attached to the CPU. After the program is downloaded we will see different combinations of 16X16 bit multiplication.

The use of this FPGA kit is that the user will see the output easily.

7. RESULTS

The result of the 16X16 Vedic multiplier is given in the figure's below in which 16 bit number is in binary and decimal form. Also the RTL schematic of Vedic and Booth multiplier is given.

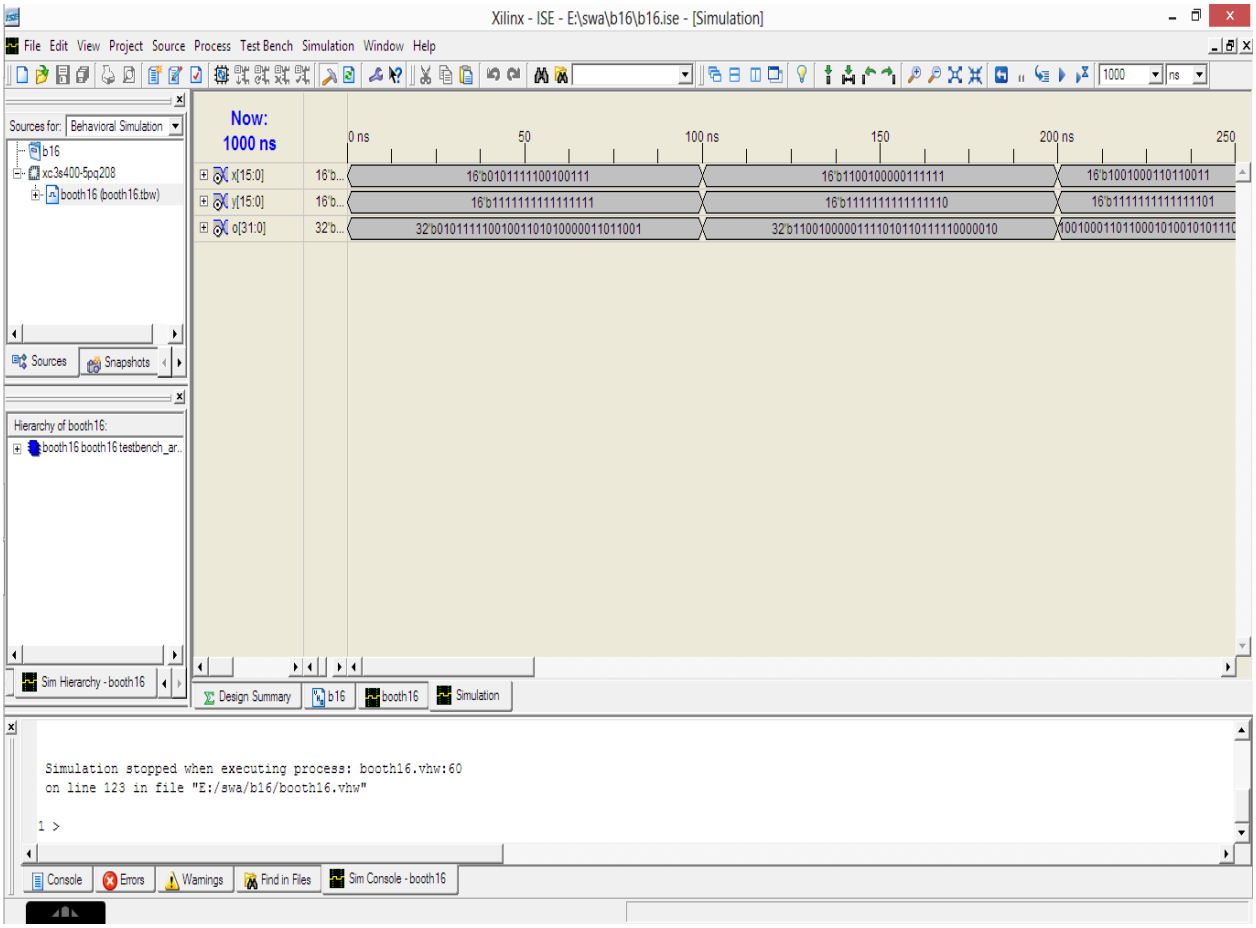

Fig 7: Result of 16X16 multiplier in binary form.

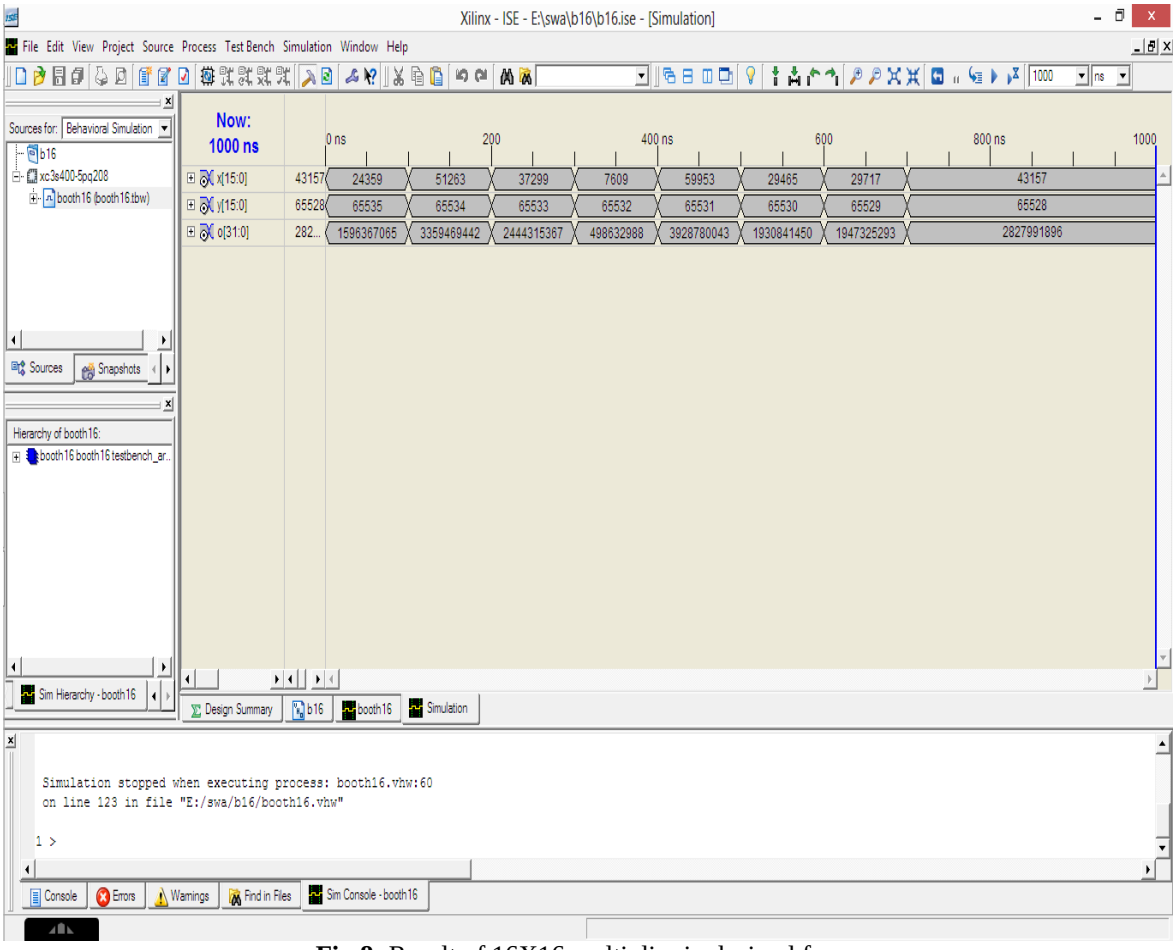

Fig 8: Result of 16X16 multiplier in decimal form.

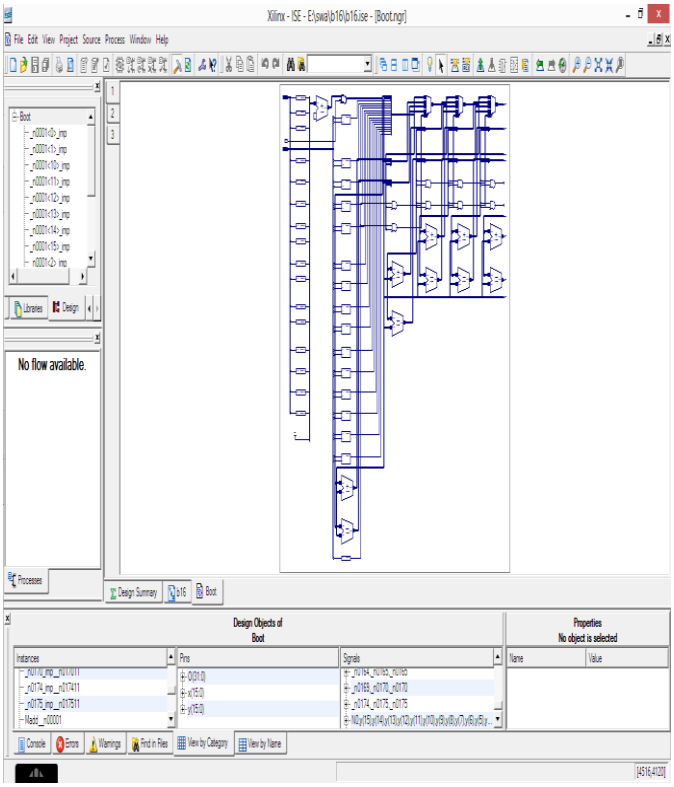

Fig 9: RTL schematic of 16X16 Booth's multiplier.

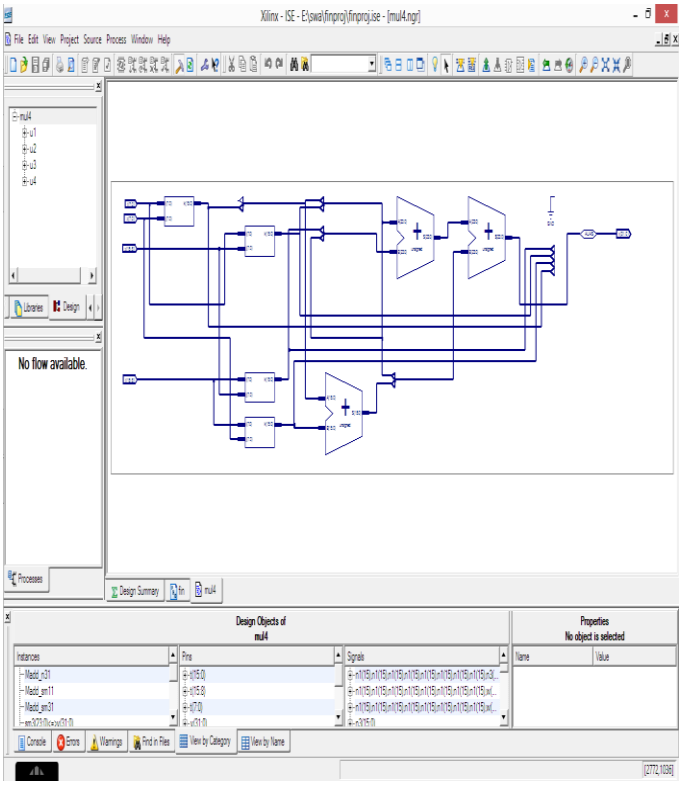

Fig 10: RTL schematic of 16X16 Vedic multiplier.

From the above RTL schematic it is clear that the hardware requirement for the Booths algorithm is much more than the Vedic multiplier, and that's why the hardware complexity of Vedic algorithm is less than the Booth's multiplier.

8. COMPARISON OF DEVICE UTILIZATION BETWEEN BOOTH'S AND VEDIC MULTIPLIER

We have compared Booth's and Vedic multiplier on the basis of device utilization which is as given below.

8.1 4x4 Booth's and Vedic Multiplier

Table 1: Comparison of device utilization theory of 4X4 multiplier

Parameter	Booth's multiplier	Vedic multiplier
Combination path delay	13.224 ns	13.363 ns
No. of slices	23 out of 3584 (0%)	18 out of 3584 (0%)
No of 4 input LUTs	42 out of 7168 (0%)	32 out of 7168 (0%)
No of bonded IOBs	16 out of 141 (11%)	16 out of 141 (11%)

8.2 8x8 Booth's and Vedic Multiplier

Table 2: Comparison of device utilization theory of 8X8 multiplier

Parameter	Booth's multiplier	Vedic multiplier
Combination path delay	50.866 ns	21.782 ns
No. of slices	185 out of 3584 (5%)	75 out of 3584 (2%)
No of 4 input LUTs	330 out of 7168 (4%)	135 out of 7168 (1%)
No of bonded IOBs	32 out of 141 (22%)	32 out of 141 (22%)

8.3 16x16 Booth's And Vedic Multiplier

Table 3: Comparison of device utilization theory of 16X16 multiplier

Parameter	Booth's multiplier	Vedic multiplier
Combination path delay	93.86 ns	28.316 ns
No. of slices	718 out of 3584 (20%)	345 out of 3584 (9%)
No of 4 input LUTs	1281 out of 7168 (17%)	622 out of 7168 (8%)
No of bonded IOBs	64 out of 141 (45%)	64 out of 141 (45%)

8.4 Comparison of Combination Path Delay between Booth's and Vedic Multiplier

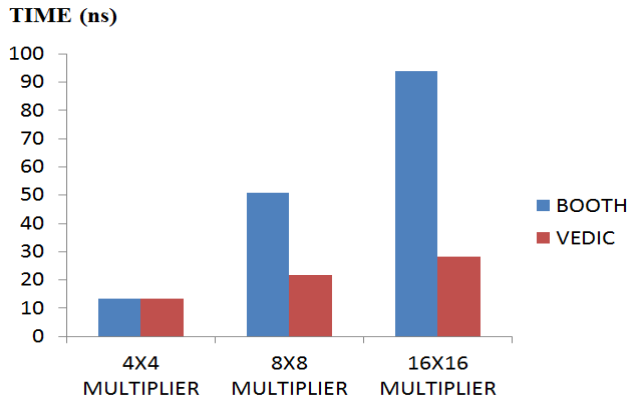

Fig 11: comparison of combination path delay between booth's and Vedic multiplier

In the figure 9 the comparison of combinational path delay between 4,8,16 bit Booth and Vedic multiplier is given. From that we can clearly conclude that the efficiency of Vedic multiplier is greater than Booth's multiplier. As no of bit increases booth become more efficient. Hence for that reason it is very important to use Vedic multiplier instead of Booth's multiplier.

9. CONCLUSION

In this paper we have developed the suitable architecture for multiplication using Vedic multiplier. This multiplier is suitable for multiplication of large number of bits. We compared Vedic multiplier over Booths multiplier and successfully show that Vedic multiplier is more convenient than Booths multiplier. The simulation results clearly shows that the Vedic multiplier is having minimum path delay as compared to the Booth's multiplier also hardware complexity of Vedic algorithm is less than the Booth's algorithm.

REFERENCES

[1]. Jagadguru Swami Sri Bharati Krisna Tirthaji Maharaja, "Vedic Mathematics: Sixteen Simple Mathematical Formulae from the Veda," Motilal Banarasidas Publishers, Delhi, 2009, pp. 5-45

[2]. Virendra Babanrao Magar, "Intelligent and Superior Vedic Multiplier for FPGA Based Arithmetic Circuits", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-3, July 2013.

[3]. Perry, "VHDL", McGraw Hill Publication.

[4]. VHDL Primer by J.Bhaskar.

[5]. A. D. Booth, "A signed binary multiplication technique," Q. J. Mech.Appl. Math., vol. 4, pp. 236–240, 1951.

[6]. Xilinx ISE User manual.

[7] .S.S.Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A "Implementation of Vedic Multiplier for Digital Signal Processing" International conference on VLSI communication & instrumentation (ICVCI), 2011.