Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

Implementation of Radix-4 (32 bit) Booth Multiplier using VHDL

Sulbha Bhongale [*1], Rajendra Patel[2],
Department of electronics and communication
[1, 2] Infinity Management & Engineering College, Sagar

Abstract -

Area, power, delay is the key design factor in VLSI circuit. In the design of microprocessor, graphical systems, multimedia systems, and DSP system Multiplier is a very essential part.

Multiplication alone uses about 15% of the IC's total power. It is crucial to have a design that is efficient in terms of speed, area, and performance of the multiplier, and for the same, Booth's multiplication algorithm offers a relatively simple strategy with a noticeable improvement in performance.

In this article, we offer a signed radix-4 Booth multiplier implementation and design. There have been discussions about a number of collation parameters, including the Wallace tree multiplier, the Booth radix-2, and the Modified radix-4 Booth multiplier. A different multiplier is compared to the radix-4 Booth 32-bit implementation.

Any multiplier has limitations, one of which is the quantity of partial products. This limitation is overcome by the multiplication Booth algorithm, which allows 32-bit multiplication to be calculated in 16 latches. In this study, we present a multiplier that, with reduced latency, minimises the partial product row via improved booth technique. We created a radix-4 booth multiplier, and the software Xilinx ISE 14.4 was used to synthesise it. By multiplying two 32-bit signed numbers, the design has been confirmed. The results are compared to published literature. Utilising the most recent version of the Xilinx ISE Design Suite tool, the simulation and synthesis results are provided.

Keywords -

Radix-4 Booth multiplier, radix-2, Ripple carry adder, Xilinx 14.4.

INTRODUCTION

The multiplication is one of the most important arithmetic operations and it is an important factor in signal processing algorithm, in digital signal processor, multimedia and microprocessor application. The high-speed processing is the key requirement of modern days applications. In many Digital Signal Processing (DSP) applications, including convolution, fast Fourier transform (FFT), filtering, in the arithmetic and logic units of microprocessors, and in graphics processors, multiplication-based operations like multiply and accumulate (MAC) and inner product are currently implemented [1]. The most commonly used elements in any digital circuit design are digital multipliers. They are used to carry out any operation since they are quick, dependable, and effective components. The add-and-shift method is used for multiplication.

The multiplicand is a number which is to be added and number of times it is called as multiplier. The repetitive addition is considered as slow multiplication. The speed of multiplier is a key factor in digital signal processing and general-purpose processor. The design of multiplier inconsistent with delay, time, area, size to minimum power dissipation. Therefore, continue high performance, the speed and area of multiplier are major design issues and they need to be improved for enhancing performance [2]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

The architecture of multiplier can be divided into three stages namely, partial product generation stage, partial product reduction stage and final addition of reduced partial products. Partial products are generated by multiplying the multiplicand with each bit of the multiplier. Generally, multiplication is done by generating partial products and then added. The speed of the multiplier depends on fast the generation of partial products and addition of these partial products. The reduction of no of partial product and use of efficient adder circuit for speeding up the process is the desired approach. Booth multiplier offers an advantage and accomplish both the steps and hence performance can be improved. In addition, the Radix-4 Booth multiplier can be performed both signed and unsigned number [3]. The aim of booth algorithm scheme to increase the speed of multiplication process, as compared to conventional multiplier such as array multiplier, Wallace tree multiplier, radix-2 multiplier. Booth multiplication helps to reduce the number of iteration step and result is faster computation. In this paper we present 32-bit multiplication by using Modified Booth (radix-4) algorithm and its implementation using VHDL platform [4].

LITERATURE REVIEW

Various multiplier is contrast by using commercial variable like delay, power, and area etc to interpret which multiplier was applicable for situation. After contrast all, it completes that radix-4 booth multiplier was applicable for situation where speed is the critical concern [5]. Implementation of radix-4 booth multiplier is done by using ripple carry adder and came to concluded that parallel multiplier is much better than serial multiplier due to less area occupied and low power consumption.

The Booth algorithm was invented by Andrew D. Booth [**6**]. The radix-2 algorithm treats both signed and unsigned number. Partial product is generated by using this algorithm, which firstly encode the multiplier bits. The radix-2 and radix-4 are two algorithms to obtained partial product generation and reduce no of partial product and final addition of reduced partial product.

In various DSP application, all of multiplier output bits were not used, but only upper bits of output were used. Kidmahi proposed prune unsigned multiplier for this idea. This prune scheme can be applied for booth multiplier which can be used in real DSP system more efficiently. Also prune booth multiplier guaranteed 0 input to 0 output that was not provided before. Prune booth multiplier reduced about 37 to 48 % of area and about power consumption.

Several designs and methods are suggested to begin a modular point of view for enhancing power efficiency [7]. It initiates that algorithm-based structure decrease gate switching activity considerably. As a result, power consumption in multiplier is reduced [8]. It is initiate that data complication various distribution of gate level digital circuit has significant effect on power dissipation. Regarding this actual design of the chip can be employed by using historical algorithm by analysing placement option subject to optimum space allocation similarly selection booth algorithm and modified booth algorithm may reduce power consumption as outcome of data complexity. It is initiate that in multiplier circuit, Modified booth algorithm reduces power consumption as compared to other method of multiplication on making implementation of radix-4 booth multiplier has power reduction than the conventional multiplier.

Booth multiplier can be composed based on dynamic span observation of multiplier and which can be compose either 32-bit multiplication.

In reported literature, several approaches and designs have been reported [9-12]. 16 bits Booth multipliers with radix-2, radix-4 methods using VHDL have been demonstrated and their performance have been analysed. Niharika and Dilip Kumar, “Analysis of Booth multiplier using radix-2 and radix-4 using VHDL”. They have demonstrated 16-bit Booth multiplier using Radix-2 and radix-4 method. The designed has been proposed which can raise the performance in term of the area, and the speed of the multiplier.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

The optimization of digital circuit can lower the number of transistors resulting into reduced area on the chip. Thus, the low power consumption can also be achieved [6].

Sagar Pradhan et al have proposed the design and implementation of low power and area efficient 32-bit multiplier. They have demonstrated two high speed multipliers, namely, Modified booth multiplier (MBM) and Wallace tree multiplier are combined with carry select adder and then they formed a hybridized that gives high speed computation and power consumption is low. Wallace tree multiplier is maintained for fast addition and then last CSA is used for final accumulation. By considering factor such as speed, power, the hybrid multiplier gives better result than conventional design. The proposed design considering the term such as delay, area, power, and their product by hand with logical effort and through custom design. It has been proposed that the Booth multiplier with Wallace tree implementation energy efficient relative to the other proposed design.

Manpreet Manna & Sukhmeet kaur, have also carried out the implementation of Modified Booth algorithm (radix-4) and compare with Booth algorithm (radix-2). They proposed the method of implementing the parallel MAC with the smallest possible delay. A new architecture of multiplier and accumulator (MAC) for high-speed arithmetic by combining multiplication with accumulation and devising a carry look ahead adder (CLA). Modified Booth multiplication algorithm is designed using high speed adder. High speed adder is used to speed up the operation of multiplication. The radix-4 booth multiplier has higher computation speed than radix-2 booth multiplier, coding of both the multiplier is done in VHDL and simulated using Xilinx ISE 9.1i software has been used and implemented on FPGA Xc3550-5pq208 [13].

A demonstrative radix-2 Booth Multiplier is described below. The faster and smaller multiplication of 2' complement of binary integer are employ by Booth technique. Therefore, Multiplication completed by Booth recoding algorithm. So first we have to recode the multiplier first. Every bit of recoded multiplier can take any value from 0, -1,1. In order to obtain, 2 bits of multiplier are conduct at a time by conjoining technique. Thus, in radix-2, clustering of multiplier bits start from LSB for which the first block utilizes only single bit of multiplier and one more bit supposed to be zero [5]. By using following steps recoded multiplier for radix-2 are obtained.

1) Add a zero to the LSB side of given multiplier.
2) By using conjoining technique, clustering of two bits of multiplier and recode the number using following table.

Q	Q_{n+1}	Recorded bits	Operation
0	0	0	Shift
0	1	+1	Add
1	0	-1	Subtract
1	1	0	Shift

Table 1: Truth table for Booth Multiplier

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

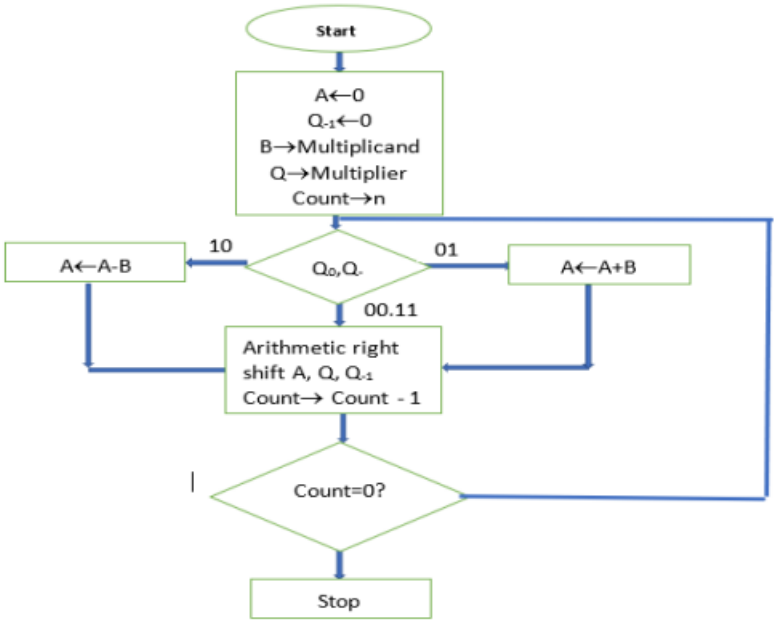


Figure 1: Flow chart of Booth Multiplier

Procedure;

M is the multiplicand

Q is the multiplier

Consider 1 bit register Q-1 and initialize to 0

Consider register A and initialize to 0

Condition;

1. If Q_0, and Q_{-1} are same 11 or 00 then perform arithmetic right shift by 1 bit
2. If Q_0 and Q_{-1} = 01 then perform A = A + M and then perform arithmetic right shift
3. If Q_0 and Q_{-1} = 10 then perform A = A - M then perform arithmetic right shift
4. Example 5 x -4 = -20
5. Multiplicand = 5
6. Multiplier = -4

Table 2 presents step by step implementation of multiplication (5 x -4) of the two numbers by Booth approach

Table 2: Multiplication of two number (5*-4) in step by step using Booth algorithm

Count	A	Q	Q_{n-1}	Operations
	0000	1100	0	Initialization
1	0000	0110	0	Right shift
2	0000	0011	0	Right shift
3	1011	0011	0	A = A-M = A + (M` +1)
	1101	1001	1	Right shift
4	1110	1100	1	Right shift
	00010011			
	Ans = 00010100			2`s complement of 20

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

- The main version of Booth algorithm (radix-2) has certain drawbacks. It becomes inconvenient while designing parallel multiplier and there is a string of acute 1s, the algorithm becomes ineffective. These drawbacks are overcome by using Modified Booth algorithm

Modified Radix-4 Booth Multiplier

One more advancement in the multiplier is by reducing the number of partial products. To attain high speed multiplication, algorithm using parallel counter like modified booth algorithm has been exhibit and accomplished. It decreases the number of partial products and reduces number of multiplicands [9]. It groups three consecutive bits at a time, in that three-bit 2 bits on left hand side are present bit and third bit is the carry bit from previous pair of bits. The multiplier performs much faster than array multiplier for extended operands because it's time to ended the operation is proportional to the log of the word length of operand. The major advantage of this multiplier is that if the successive bits in multiplicand are same, then addition can be skipped [14]. The number of partial products is downed by a factor of half by using the technique of booth recoding [15]. The grouping of bits and booth recording determine the number of partial products.

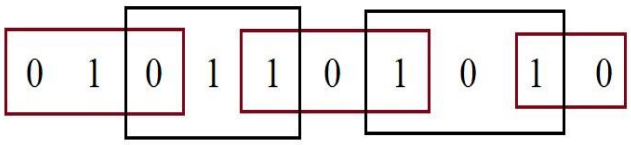


Figure 2: Grouping of bits from the multiplier term

In a given multiplier, the group of three bits are considered, it initiates from the LSB bit and first group is formed by two bits, onward consider a group of three bits in which one bit will be overlapped on the previous group as shown in figure 4. Thus, groups multiplier will result in the production of bits between these five bits are as follows -2, -1, 0, +1, and +2 which are tabulated in table 3.

Table 3: Recoding table of radix-4 Booth multiplier

Block	Recoded digits	Operation
000	0	0
001	+1	+1
010	+1	+1
011	+2	+2
100	-2	-2
101	-1	-1
110	-1	-1
111	0	0

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

Algorithm:

1) Check in case the multiplier and multiplicand are having negative sign if so convert 2's complement
2) Add implied zero to the multiplier and then convert the multiplier into other form using Booth recoding table shown in above table
3) Now multiply the multiplier with multiplicand
    a. If the multiplier was zero then the multiplicand multiplied with zero will result in the output is zero
    b. If the multiplier was +1 then, the result of the multiplicand with the multiplier was same as the multiplicand
    c. If the multiplier was -1 then, the result was calculated by 2's complement of the multiplicand
4) Finally add all generated partial products in ripple carry adder.

Methodology

From the principle of Booth Multiplication, it can be manifested that the addition/subtraction operation can be omitted if the consecutive bits in the multiplicand are same. If 3 consecutive bits are same then addition/subtraction operation can be omitted. therefore, normally the delay related with Booth Multiplication are smaller than that with Array Multiplier. Although, the execution of Booth Multiplier for delay is input data dependant. In the worst case the delay with booth multiplier is on par with Array Multiplier [16]. The method of Booth recording reduces the numbers of adders and hence the delay required to produce the partial sums by examining three bits at a time. The high performance of booth multiplier comes with the drawback of power consumption. The reason is large number of adder cells required that consumes large power [3].

Implementation and Result

Multiplication based on Radix-4 Booth algorithm has been simulated on ISIM simulator of Xilinx 14.4 software and it is shown in Fig. 3 and implemented on FPGA platform for which the results are obtained.

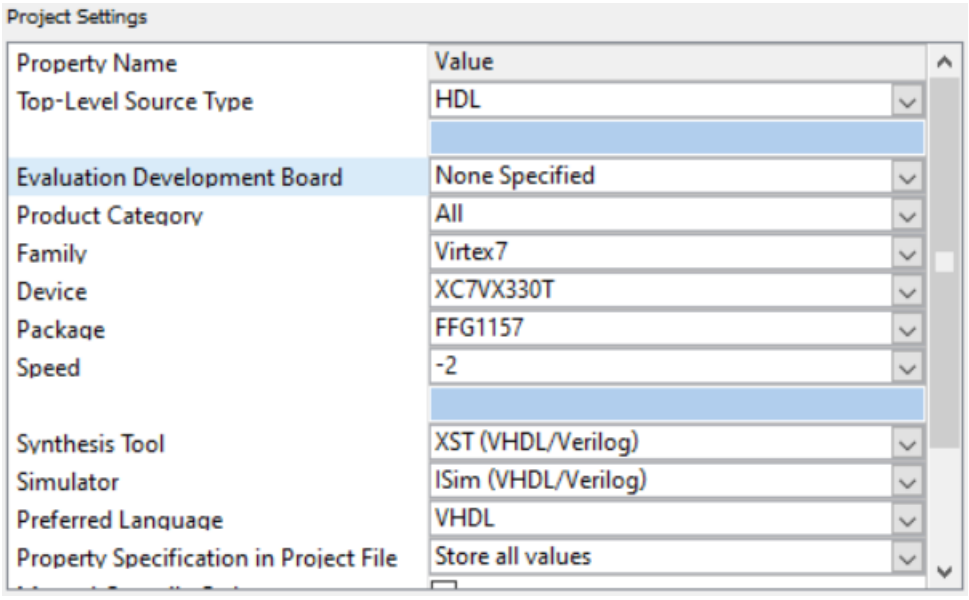


Figure 3: Properties of Xilinx 14.4i Module

In the design of Booth multiplier for Radix-4, there are two inputs namely Multiplier [32:0] and Multiplicand [32:0] and single output product [64:0] and the carry output. The schematic is presented in Fig. 4.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

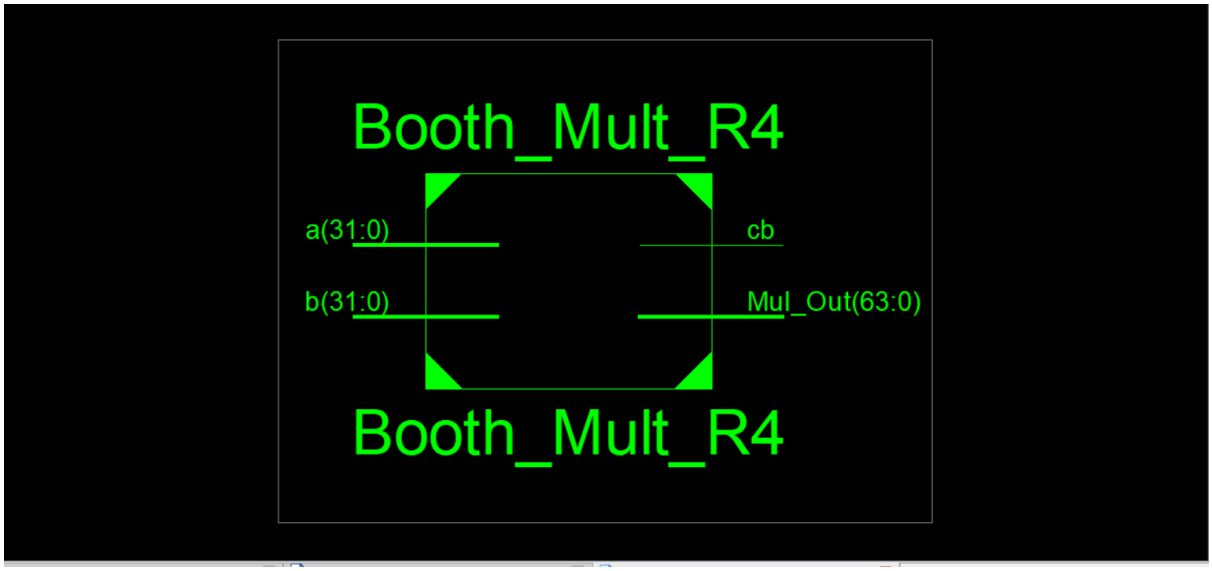
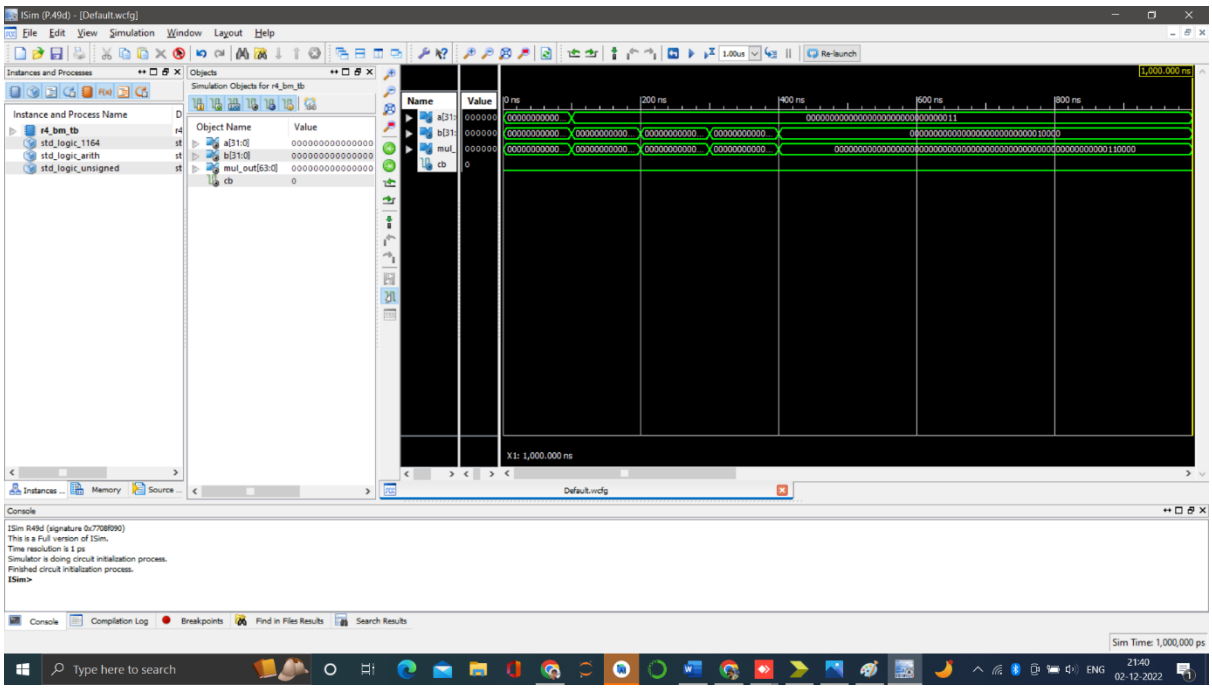


Figure 4: shows RTL of Booth Multiplier using Radix-4

Output is verified for input of different ranges. In this case the simulation results are shown for input value in binary for multiplier "00000000000000000000000000000011" and a multiplicand "00000000000000000000000000010000". The result of above multiplication is given in binary form and the product is "00000000000000000000000000000000000000000000000000000110000". The simulation result is presented.

Synthesis result of radix-4

The multiplier has been synthesized using Xilinx ISE 14.4. The RTL schematic of Booth multiplier using Radix-4 has been shown in Fig.5



International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

Figure 5: shows simulation result using Radix-4

In the RTL schematic of Booth multiplier multiplicand [32:0] and multiplier [32:0] represent the 32 bit input operands and product [64:0] represent 64 bit output product. Figure shows the internal RTL of the Booth multiplier and figure the device utilization of Booth multiplier using Radix-4.

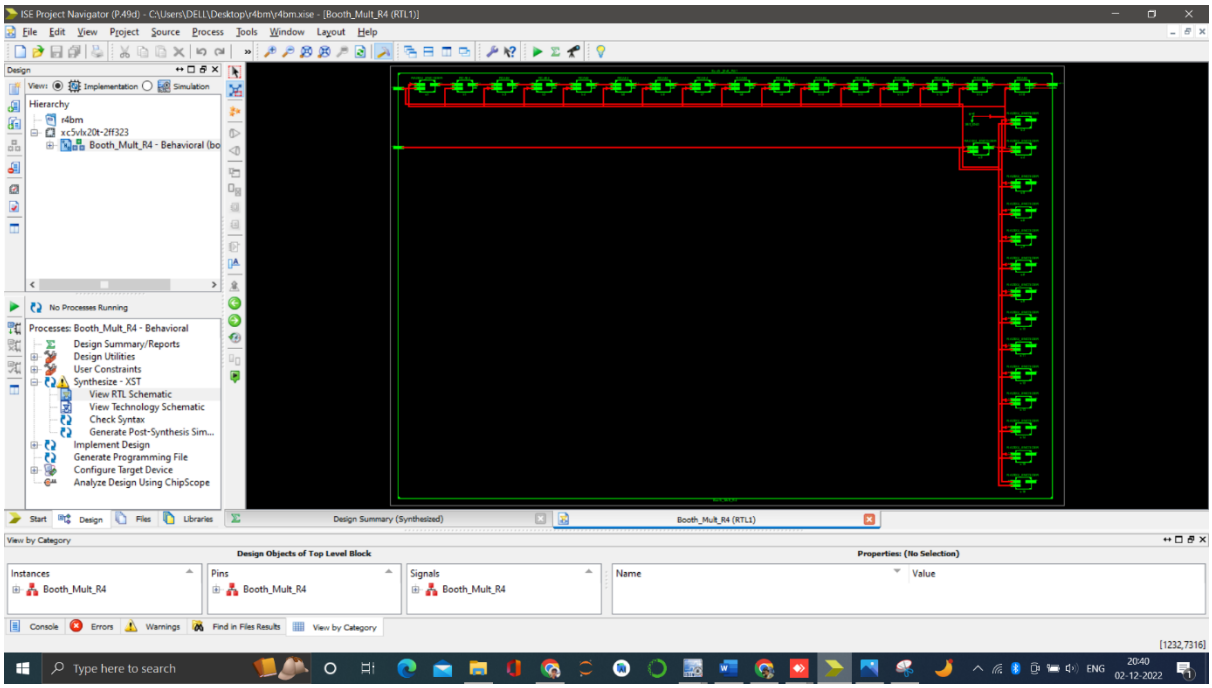


Figure 6: shows Internal RTL of Booth Multiplier using Radix-4

The delay analysis is performed using various device family. The details are tabulated in table 4-6 as shown below.

Table 4: Device utilization of Booth Multiplier using Radix-4

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	2514	204000	1%
Number of fully used LUT-FF pairs	0	2514	0%
Number of bonded IOBs	129	600	21%

Table 5: Shows Device utilization proposed Radix-4 Booth multiplier

Device & Family	Vertex 5	Vertex 6	Vertex 7	Spartan 3	Spartan 6
Number of slices	2522	2514	2514	1881	2514
Number of fully used LUTFFF pair	0	0	0	3542	0
Number of Bonded IOB	129	129	129	129	129

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

Table 6: Comparison table of Radix-4 Booth Multiplier

Design	Proposed Radix-4 Booth Multiplier			Reference paper number 2
Device and Family	Vertex 5	Vertex 7	Spartan 3	Spartan 3
Delay(ns)	35.80ns	26.32ns	111.29ns	500ns

Conclusion and future work:

After doing some work and observed a lot of difficulty, we maintain to accomplish the main aim of the work is to implement Booth algorithm for the design of radix-4 Booth multiplier using ripple carry adder architecture and performed delay analysis using various device family that shown in table 4-6. We evaluate the time delay in design with Ripple carry adder. Design and implementation of radix-4 Booth multiplier using VHDL and synthesized results were presented. The Radix-4 Booth multiplier have been simulated using Xilinx platform and implemented on vertex XC7z010-Zcg400. The proposed Modified Booth Multiplier has taken less computation time to multiply the two 32 bits binary number and analysis shows that the delay obtained for Modified Booth multiplier using RCA is 26.32 ns which is quite low compared with reported literature [1]. Ramteke et al reported delay of 500ns to complete 8-bit multiplication. It has proved that that it can be useful to apply a Radix-4 Booth Multiplier in high-speed multiplier because gain in time obtained due to reduction in partial product to N/2, by using efficient encoding method, which save multiplier area and reduced delay at the same time. The high-speed implementation of such multiplier has wide range of application in image processing, arithmetic logic unit and VLSI signal processing. In future the algorithm can be extended to Radix-8 for which complexity is somewhat high, but the generated partial product will reduce to N/3.

REFERENCES:

[1] Sneha Manohar Ramteke, Alok Dubey, Yogeshwar Khandagare, “VLSI desiging of low power Radix-4 Booth Multiplier”, International journal of recent Advances in Engineering & Technology (IJARET), volume-10, issue-2,2022

[2] Bhawna Singroul, Pallavee Jaiswal, “A review on performance Evaluation different digital multiplier in VLSI using VHDL”, International journal of Engineering research & technology, vol-7, issue-5 ,2018

[3] Sagar Pradhan, “Design and Implementation of low power and area efficient 32-bit booth multiplier”, An international E-journal on Engineering trends in science Technology and management, special issue-2016

[4] B. Lamba and A. Sharma, "A review paper on different multipliers based on their different performance parameter", 2nd International Conference on Inventive Systems and Control, pp 324-327, 2018

[5] Bhavya Lahari Gundapaneni, JRK kumar Dabbakutti, “A review on Booth Algorithm for the design of multiplier”, International journal of innovative technology and Exploring Engineering, vol-8, issue-7, pp 1506-1509, 2019

[6] A. D. BOOTH, “A signed Binary multiplication technique”, in journal of Mech. APPL. Math, Oxford University press, vol-4, pp 236-240,1951

[7] Niharika, Dilip Kumar, “Analysis of booth multiplier radix-2 and radix-4 using VHDL”, International journal of research and development in applied science and engineering, Vol 8, issue 2, January 2016.

[8] Sandip Srivastava and Mukesh Tiwari, “Implementation of radix-2 booth multiplier and comparison with radix-4 Encoder Booth Multiplier, International journal on engineering technology, volume 2, February 2011.

[9] Shweta S. Khobragade, Swapnil P. Kormore, “A review on low power VLSI design of Modified Booth multiplier”, International journal of Engineering and Advanced Technology, vol-2, issue-5, pp 463-466, 2015

[10] Bhavya Lahari Gundapaneni, JRK kumar Dabbakutti, “A review on Booth Algorithm for the design of multiplier”, International journal of innovative technology and Exploring Engineering, vol-8, issue-7, pp 1506-1509, 2019

[11] Niharika, Dilip Kumar, “Analysis of booth multiplier radix-2 and radix-4 using VHDL”, International journal of research and development in applied science and engineering, Vol 8, issue 2, January 2016.

[12] Hasan Krad and Aws Yousif, “Design and Implementation of a fast unsigned 32-bit multiplier using VHDL”

[13] Dr. Manpreet Manna and Sukhmeet Kaur, “Implementation Modified booth algorithm(radix-4) and its comparison with Booth algorithm (radix-2), Advance in Electronics and Electric Engineering”, Advance in Electronics and Electric Engineering vol.3, November 6 (2013), pp.683-690.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 06, June-2023

[14] Savita Nair, "A review paper on comparison of multiplier based on performance parameter", International journal of computer application, vol-2, pp 6-9, 2014

[15] Sumit Vaidya, Deepak Dandekar, "A review on delay performance comparison of multiplier in VLSI circuit design", International journal of computer network & communication, Vol 2, issue 4, pp 47-55, 2010

[16] Rubi Choubey, Md. Arif , " Area optimized and low power using Modified Booth multiplier for unsigned number, International journal of Engineering Science and Engineering (IJESE), vol 2, issue-6, April 2014.