

# Implementation of a Parallel Multiplier for FFT Application

Saniya Paulson

M.tech Student,SSET Karukutty, [spsans62@gmail.com](mailto:spsans62@gmail.com), 9496715586

**Abstract**— Multipliers will play an important role in today's digital signal processing and various other applications. The multiplication operation consists of producing partial products and then adding these partial products the final product is obtained. Thus the speed of the multiplier depends on the number of partial products and the speed of the adder. In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. The main aim in VLSI field is to make a multiplier which increases the speed and reduces the area and power. This paper focuses on the implementation of a 8-point and 16-point FFT using a parallel multiplier designed using Advanced Modified Booth Encoding Algorithm(AMBE) for 8-bit signed-unsigned number multiplication. This paper describes comparison of various multipliers based on area and delay. Simulations are done using Xilinx 14.2ISE and implemented on SPARTAN-6 FPGA board.

**Keywords**— AMBE,FFT, Parallel multiplier, Signed, Unsigned, Partial products, Adder, Multiplier

## INTRODUCTION

Enhancing the processing performance and reducing the power dissipation of the systems are the most important design challenges for multimedia and digital signal processing (DSP) applications, in which multipliers frequently dominate the system's performance and power dissipation. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) or discrete wavelet transform (DWT). Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition determines the execution speed and performance of the entire calculation. As the multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined by the multiplier, in general. Multiplication is most commonly used in every step of the world. It is nothing but the addition by the multiplicand adds multiplier no. of times. But, it takes very large hardware resources and simulation takes very large time for the final output. The multiplication mainly consists of two steps:

- i) To generate Partial products and
- ii) To add all these partial products until the final output.

The main aim in VLSI field is to make a multiplier which increases the speed and reduces the area and power. Earlier multipliers were used either for signed or unsigned numbers like Baugh-Wooley multiplier[1][2] for signed numbers,array multiplier[3] for unsigned numbers, Braun array multiplier[4] for unsigned numbers etc. In this paper earlier multipliers are compared and a parallel multiplier[5][6] using Advanced Modified Booth Encoding Algorithm for both signed and unsigned number is being implemented. This multiplier reduces the total chip area as well as the power. Hence it is used to obtain a 8-point FFT and a 16-point FFT .

**RESULTS FROM THE LITERATURE SURVEY**

TABLE1: COMPARISON OF MULTIPLIERS

Parameter	Array Multiplier	Baugh-Wooley Multiplier	Combined Multiplier	MBE Multiplier[7][8][9]	AMBE Multiplier[10][11]
Multiplier Type	Unsigned	Signed	Either signed or unsigned	Signed	Both signed&unsigned
No.of slice LUTS	105	146	247	153	137
No.of bonded IOBs	32	32	33	35	33
Delay(ns)	17.668	21.371	20.245	13.842	11.924

The evaluation results show that the AMBE has better performance, reduced delay and area. Hence it is used to implement 8-point and 16-point FFTs.

**PROPOSED ADVANCED MODIFIED BOOTH ENCODING MULTIPLIER (AMBE)**

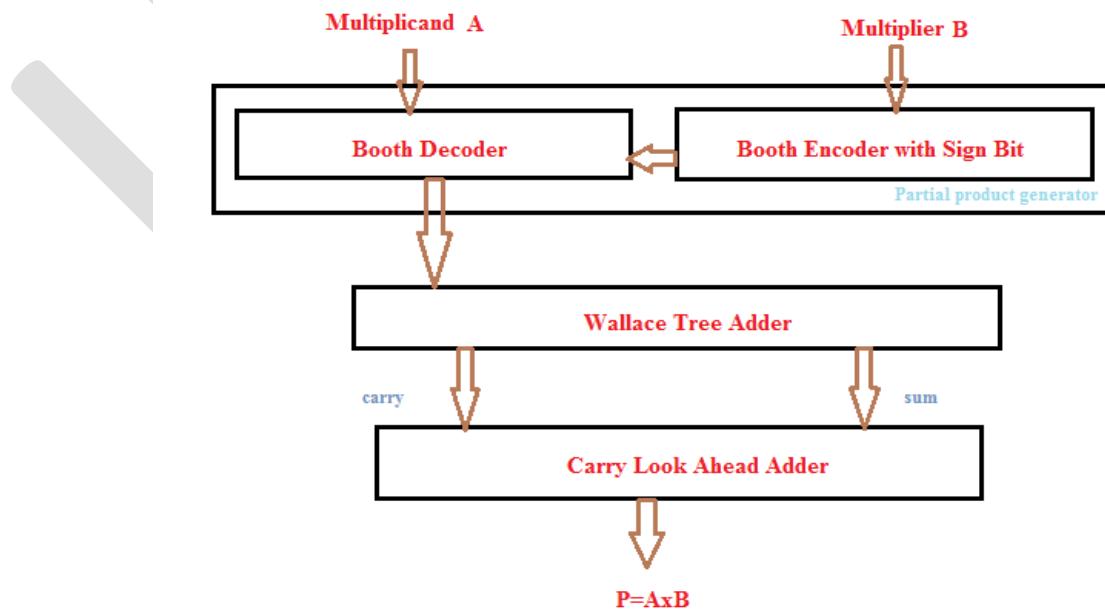


Fig 1: Block diagram of proposed AMBE multiplier

The proposed AMBE follows the same architecture diagram as that of modified booth encoding multiplier. The difference is that the encoder is extended with 2 sign bits. There are Booth Encoder, Booth Decoder, Wallace Tree adder and Carry Look Ahead adder blocks.

The Booth Encoder encodes the multiplier bits and generates the encoded signals  $\{-2, -1, 0, 1, 2\}$ . The logic diagram for Booth Encoder is shown in fig 2. The Booth encoder encodes the signals  $X1\_a$ ,  $X2\_a$ , and  $Z$ .

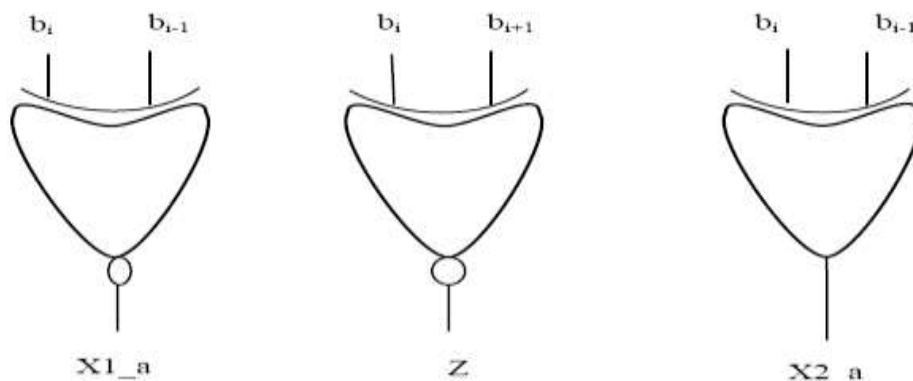


Fig 2: Logic Diagram of Booth Encoder

Using MBE logic, Booth decoder generates the partial product bit, which is given by Equation (1).

$$p_{ij} = (a_i \oplus b_{i+1} + b_{i-1} \oplus b_i) (a_{i-1} \oplus b_{i+1} + b_i \oplus b_{i+1} + b_{i-1} \oplus b_i) \quad (1)$$

Equation(1) is implemented as shown in Fig 3. The proposed AMBE multiplier does not separately consider the encoder and the decoder logic, but instead implemented as a single unit called a partial product generator.

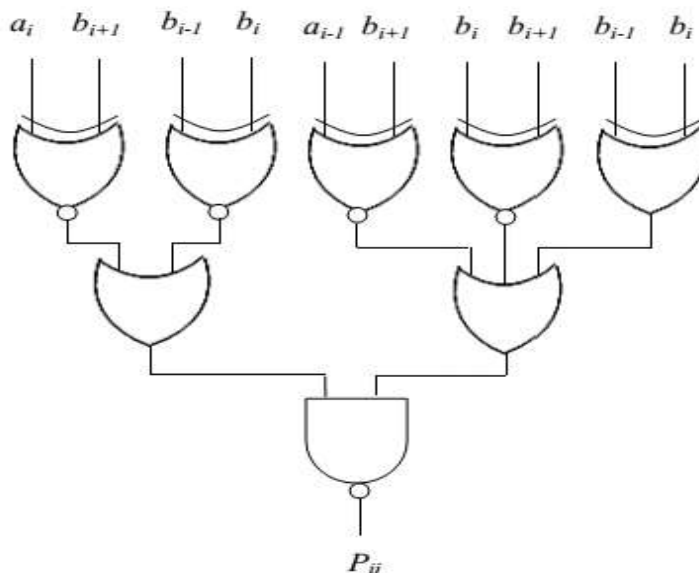


Fig 3: Logic Diagram of 1-bit Partial Product Generator Circuit

The negative partial products are converted into 2's complement by adding a negate ( $N_i$ ) bit. The Negative bit is represented by Equation(2).

$$N_i = b_{i+1} \overline{(b_{i-1} b_i)} \quad (2)$$

The Equation (2) is implemented as shown in Fig 4.

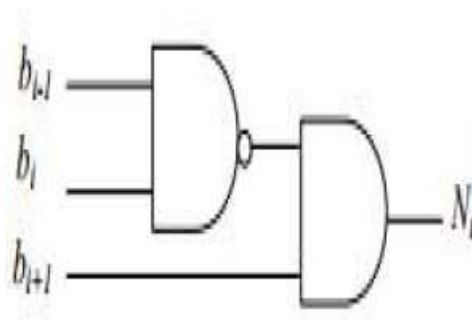


Fig 4: Logic Diagram of Negative bit Generator Circuit

Table 2: Truth table of proposed AMBE

$b_{i+1}$	$b_i$	$b_{i-1}$	Value	X1_a	X2_a	Z	Neg
0	0	0	0	1	0	1	0
0	0	1	1	0	1	1	0
0	1	0	1	0	1	0	0
0	1	1	2	1	0	0	0
1	0	0	-2	1	0	0	1
1	0	1	-1	0	1	0	1
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	0

The sign extension bit is converted to signed-unsigned number by Equations (3) and (4).

$$a8 = s\_u * a7 \quad (3)$$

$$b8 = s\_u * b7 \quad (4)$$

The logic Diagram of Equation (3) and Equation (4) are implemented as shown in Fig 5.

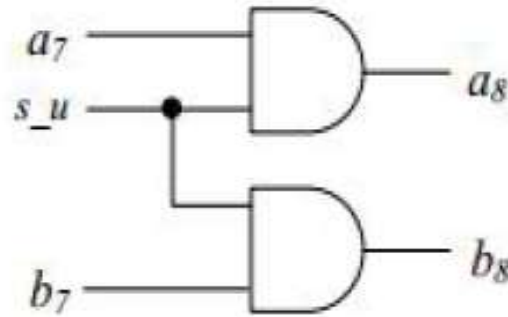


Fig 5: Logic Diagram of Sign Extension bits

One bit control signal called signed-unsigned ( $s_u$ ) bit is used to indicate whether the multiplication operation is signed number or unsigned number. When Sign-unsigned  $s_u = 0$ , it indicates unsigned number multiplication, and when  $s_u = 1$ , it indicates signed number multiplication.

It is required that when the operation is unsigned multiplication the sign extended bit of both multiplicand and multiplier should be extended with 0, that is  $a_8 = a_9 = b_8 = b_9 = 0$ . It is required that when the operation is signed multiplication the sign extended bit depends on whether the multiplicand is negative or the multiplier is negative or both the operands are negative.

For this when the multiplicand operand is negative and multiplier operand is positive the sign extended bits should be generated are  $s_u = 1$ ,  $a_7 = 1, b_7 = 0$ ,  $a_8 = a_9 = 1$ , and  $b_8 = b_9 = 0$ . And when the multiplicand operand is positive and multiplier operand is negative the sign extended bits should be generated are  $s_u = 1$ ,  $a_7 = 0, b_7 = 1$ ,  $a_8 = a_9 = 0$ , and  $b_8 = b_9 = 1$ .

Table 3:  $s_u$  bit operation

SIGNED-UNSIGNED BIT ( $s_u$ )	TYPE OF OPERATION
0	Unsigned
1	Signed

Fig 6 represents the partial products generated from fig 5. There are 5 partial products and Negative bit generator. These partial products are  $X_1, X_2, X_3, X_4$  and  $X_5$ . These partial products are added by Wallace tree Adder which uses Carry Save Adder circuit. It adds partial products until final two outputs are left.

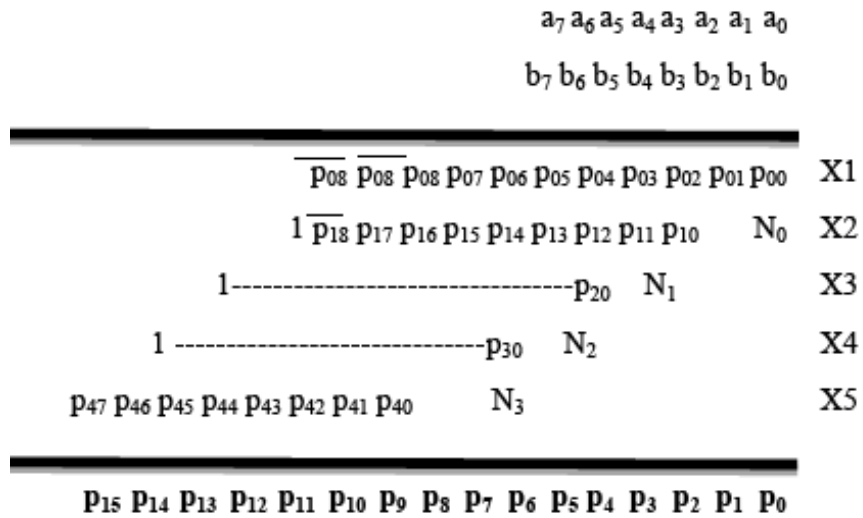


Fig 6: 8x8 signed unsigned generated Partial Product bits

Carry Save Adder (CSA) tree diagram is shown in fig 7 below. When there are only two outputs are left, Carry Look Ahead (CLA) Adder[12] is used to do the final addition and give the Final product.

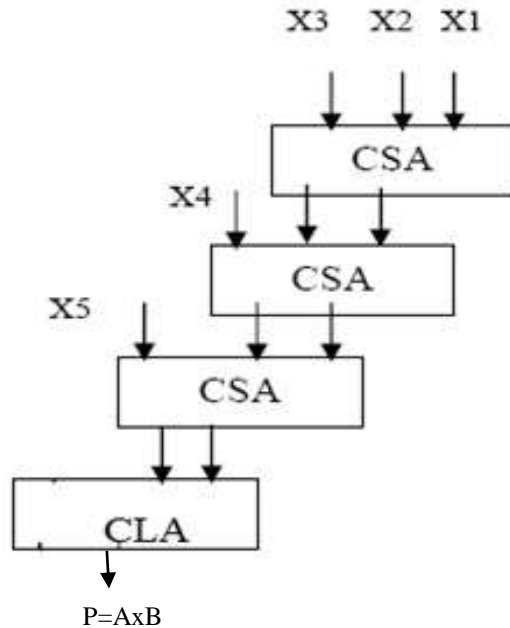


Fig 7: Wallace Tree Adder

The 8-point and 16-point FFTs shown below are implemented using this proposed AMBE multiplier.

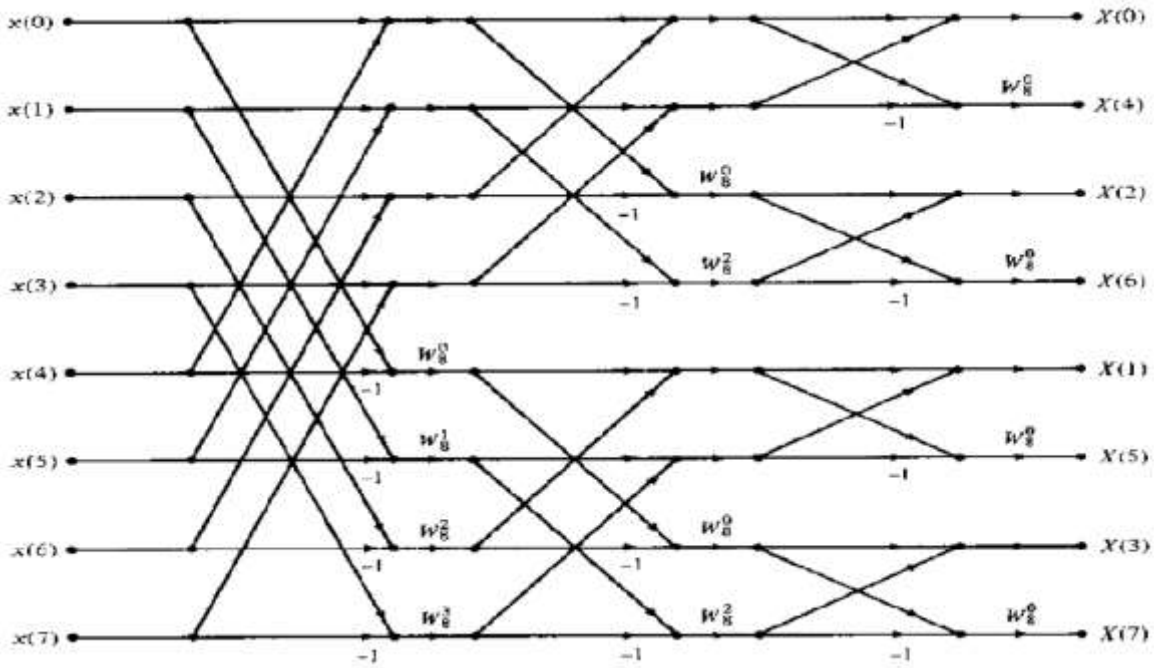


Fig 8: 8-point FFT diagram

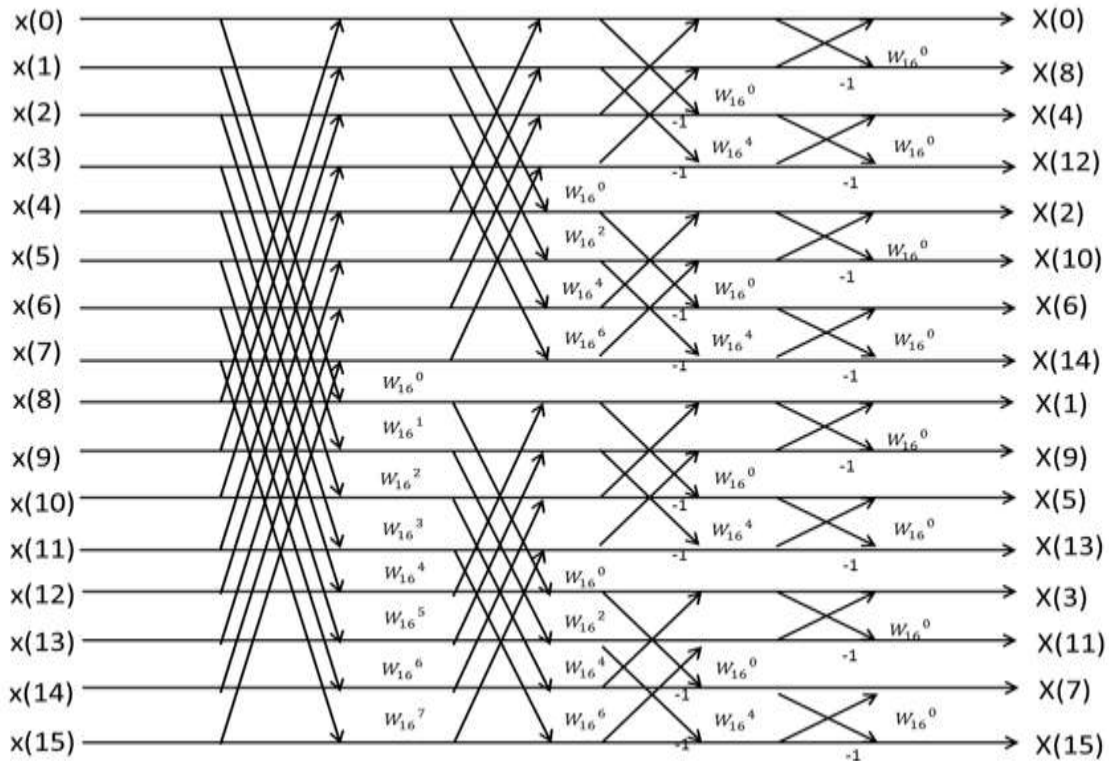


Fig 9: 16-point FFT diagram



**SIMULATION RESULTS**

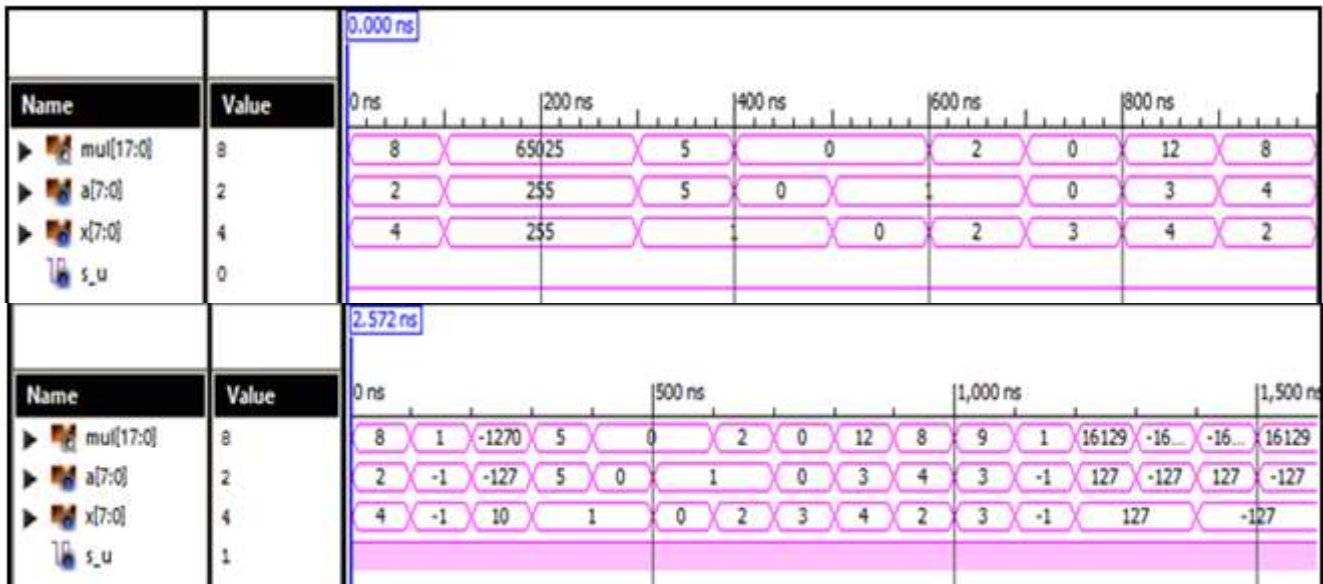


Fig 10: Simulation waveform of 8x8 AMBE

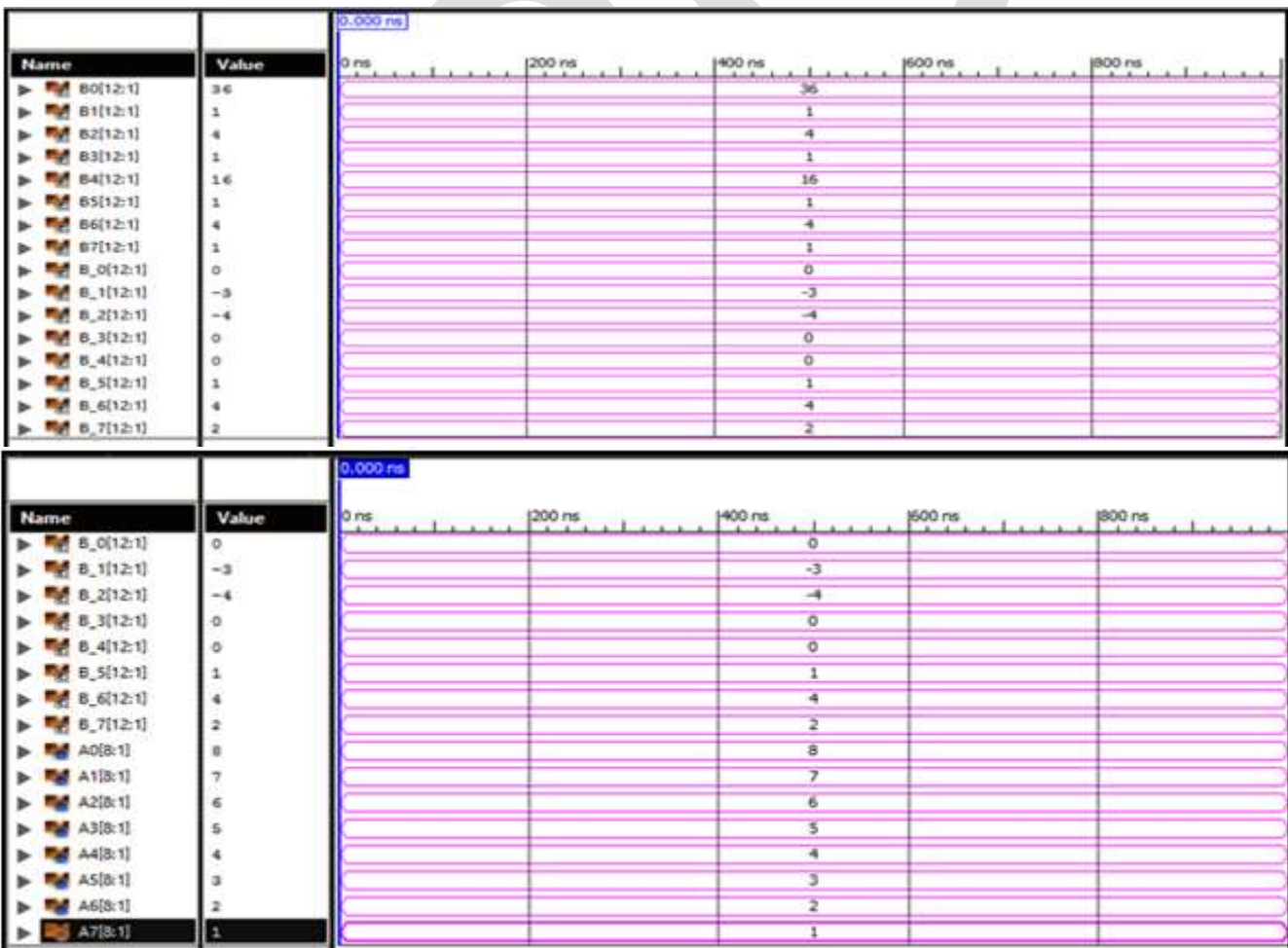


Fig 11: Simulation waveform of 8-POINT DIF FFT



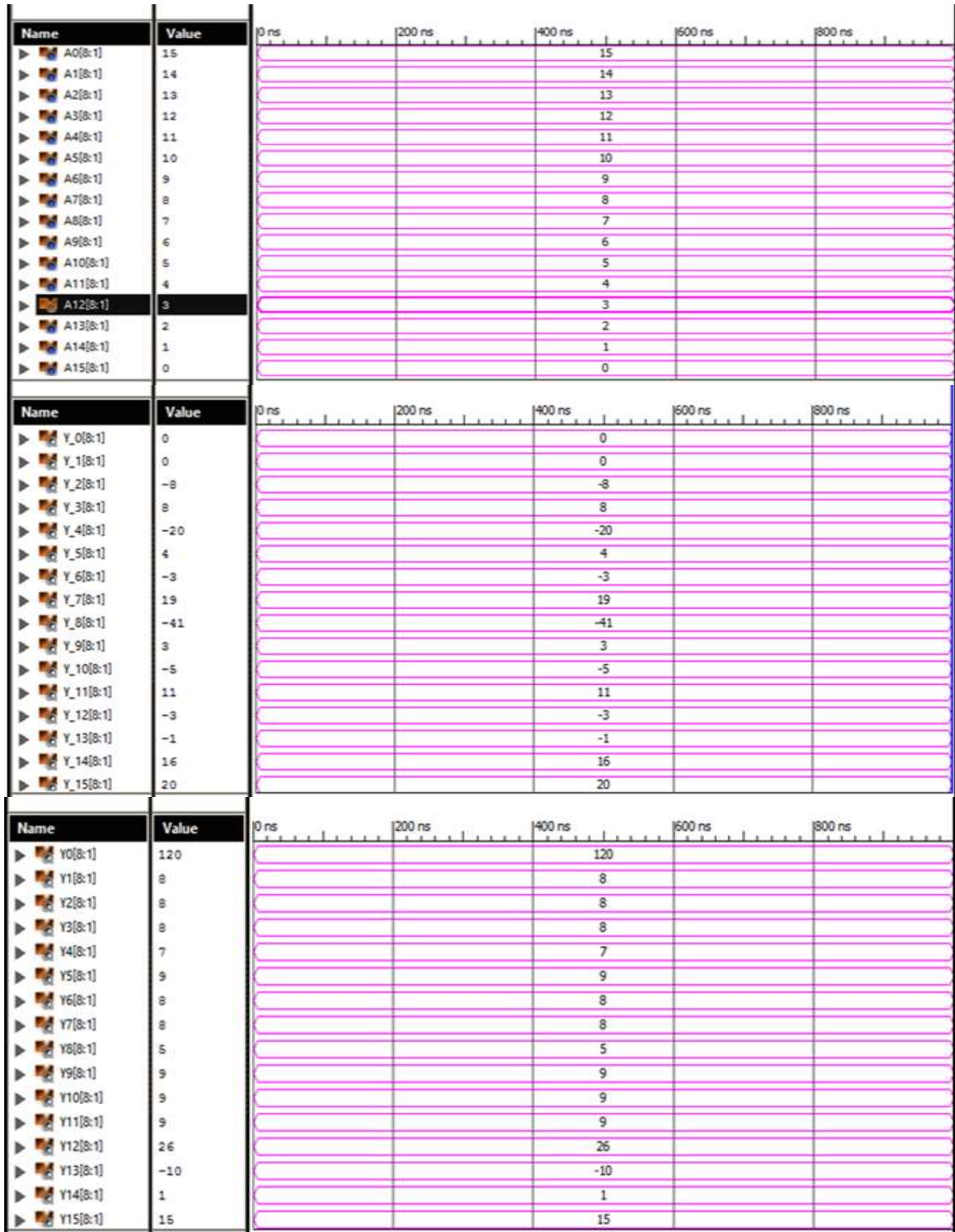


Fig 12: Simulation waveform of 16-POINT DIF FFT

## CONCLUSION

Multiplication is most commonly used in every step of the world. The main aim in VLSI field is to make a multiplier which increases the speed and reduces the area and power. The proposed 8-bit Advanced Modified Booth Encoding Parallel Multiplier has so far obtained the less area and delay than all other multipliers. It performs both signed and unsigned operations in parallel. AMBE uses the same architecture as that of a Modified Booth Encoding Multiplier, but the difference is that the encoder has been extended with a sign bit. The evaluation results show that the proposed design has better performance, reduced delay and area. Hence it is used to implement 8-point and 16-point FFTs. The simulations have been done on Xilinx 14.2ISE and implemented on SPARTAN-6 FPGA board.

## REFERENCES:

- [1] Indrayani Patle, Akansha Bhargav, Prashant Wanjari, "Implementation of Baugh-Wooley Multiplier Based on Soft-Core Processor," IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 10 (October. 2013), ||V3|| PP 01-07 .
- [2] Pramodini Mohanty, VLSI Design, 2011-2012: "An Efficient Baugh-Wooley Architecture for Both Signed & Unsigned Multiplication", Pramodini Mohanty et al./ International Journal of Computer Science & Engineering Technology (IJCSSET).
- [3] N. Ravi, A. Satish, Dr. T. Jayachandra Prasad and Dr. T. Subba Rao, "A New Design for Array Multiplier with Trade off in Power and Area," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, May 2011 ISSN (Online): 1694-0814 www.IJCSI.org.
- [4] Ms. Madhu Thakur, Prof. Javed Ashraf, "Design of Braun Multiplier with Kogge Stone Adder & its Implementation on FPGA", International Journal of Scientific & Engineering Research, Volume 3, Issue 10, October-2012 ISSN 2229-5518 .
- [5] W.-C. Yeh and C.-W. Jen, "High-Speed Booth Encoded Parallel Multiplier Design," IEEE Transactions on Computers, vol. 49, no. 7, pp. 692-701, July 2000.
- [6] T.V. Subbi Reddy and J. Nirmala Bai, "16-Bit High Speed Modified Booth Multiplier for Signed and Unsigned Numbers", International Journal of New Trends in Electronics and Communication (IJNTEC) Vol.1, Issue.3, Oct. 2013.
- [7] Kavita, Jasbir kaur, "Design and Implementation of an Efficient Modified Booth Multiplier using VHDL" , Special Issue: Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013.
- [8] Shiann-Rong Kuang, Jiun-Ping Wang, and Cang-Yuan Guo, "Modified Booth multipliers with a Regular Partial Product Array," IEEE Transactions on circuits and systems-II, vol 56, No 5, May 2009.
- [9] Ravindra p rajput, M.N Shanmukha swamy, "High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers", 2012, 14th international conference on modelling and simulation, doi 10.1109/uksim.2012.99.

[10] Shaikh Kailash Baba& D. Rajaramesh, "Design & Implementation of Advanced Modified Booth Encoding Multiplier", International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726, Volume 2 Issue 8 August. 2013 PP.60-68.

[11] K. Tsoumanis, N. Axelos, N. Moschopoulos, G. Zervakis, and K. Pekmestzi, "Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding", IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 2, FEBRUARY 2016.

[12] C. Padma ,Y. Mahesh, "Design of High Speed Modified Booth Encoded Parallel Multiplier using Carry Look Ahead Adder", i-manager's Journal o ,Vol. No. 1 Embedded Systems 2 February - April 2013.

IJERGS