

Performance Analysis of MAC Unit using Booth, Wallace Tree, Array and Vedic Multipliers

Nitin Krishna V

B. E. Student, Dept of Electronics and Communication Engineering, PSG College of Technology,
Coimbatore, India

Abstract— High-performance Digital Signal Processors are the need of the hour in today's world. MAC units being integral parts of such processors are desired to consume low power and to operate at high speeds. In this paper, a detailed analysis of MAC units constructed using four different types of multipliers namely Booth, Wallace tree, array, and Vedic, is carried out. Carry Save Adder, PIPO shift register are used as the adder and the accumulator in the MAC unit. Analyzing the performance of MAC units constructed using different multipliers can help identify the optimum unit to be used in the DSP processors. These designs are constructed for three different bit lengths (4, 8, and 16) and are compared in terms of power consumption, the delay incurred, and FPGA utilization parameters like LUTs, nets, and leaf cells. These designs are analyzed and simulated using the Xilinx Vivado tool and implemented on Zedboard Zynq 7000 Evaluation and Development kit(xc7z020clg484-1).

Keywords— MAC unit; multiply accumulate; Booth's algorithm; Array multiplier; Vedic multiplication; Wallace-tree; DSP processors; Carry-save adder;

I. INTRODUCTION

Recent advances in communication and multimedia systems have resulted in a demand for efficient and fast digital signal processing systems. DSP systems and algorithms are used for processing data streams almost everywhere and therefore they require high precision and timing accuracy. High-performance digital signal processors consume very low power while operating at high speeds. Filtering, convolution, polynomial evaluation, and dot-matrix operations are some of the processes involved in processing signals digitally. These operations usually involve multiplication and addition and are performed by the Multiply and Accumulate(MAC) unit. The MAC Unit is an integral part of all Digital Signal Processors. Fast Fourier Transform (FFT) and DTFT require a large number of multiplication and addition operations. The performance of the processor depends largely on the MAC unit. The area occupancy, power consumption, and delay incurred in the MAC unit can influence the overall performance of the system. A MAC unit consists of an adder, a multiplier, and an accumulator. In general, the delay incurred in DSP systems is mainly due to long multiplication processes in the multiplier. A wide variety of multipliers are known to be in existence. Each multiplier has a unique structure and follows a unique algorithm. Vedic, Booth, array and Wallace tree multipliers are some of the multipliers that are being widely used. Analyzing and comparing these multipliers based on power consumption, the delay incurred and area occupancy can help identify the optimum multiplier to be used in the MAC unit.

In [1], various multiplication hardware realizations have been presented along with a detailed description of parameters to be considered while designing any digital system. A revolutionary multiplication technique for signed binary numbers has been proposed in [2]. This technique widely called Booth's multiplication algorithm is independent of any foreknowledge about the signs of these numbers. In [3], a MAC unit was designed using the Booth multiplier and ripple carry adder. The MAC unit coded in VHDL was analyzed, synthesized, and simulated using Xilinx ISE Design Suite. In [4], Radix-8 Booth multiplier has been designed for signed and unsigned numbers using Verilog. The multiplier has been implemented in Spartan 3 kit. The Wallace tree multiplier has been designed, implemented, and analyzed in terms of speed and equipment cost in [5]. The multiplier has adopted an algorithm which reduces the number of summands and accelerates the formation, and addition of summands. In [6], the Wallace tree multiplier is compared with the array multiplier and it is shown that the Wallace multiplier outperforms the latter in terms of speed and power consumption. In [7] a 32-bit MAC using the Wallace tree multiplier was implemented on the Spartan XC3S500-4FG320 device and synthesized using 90 nm technology using Synopsys Design Compiler and its results were compared with the conventional MAC unit. A 32 bit MAC unit using a Vedic multiplier and carry-save adder was proposed in [8]. The delay incurred, utilization, and the number of LUTs required were compared with existing designs. In [9], the procedure for multiplication using Vedic Mathematics is proposed and the results show that the time complexity involved is lesser than the conventional ways of multiplication. The delay-power performance of various multipliers like Booth, Wallace tree, and Vedic are compared in [10]. The implementation of these circuits in VLSI Design is also described. In [11], the use of reversible logic gates in the design of the Arithmetic Logic Unit is elucidated. Furthermore, a detailed description of various reversible logic gates like Peres, Feynman, DKG, etc., is provided. A comparison between 64 bit MAC units constructed using Vedic and Wallace tree multipliers is provided in [12]. In [13], the array multiplier has been designed and implemented using different adder designs in CADENCE design suite at 180 nm technology. In [14], the array multiplier has been designed without any truncation or addition technique. Parameters like silicon area, delay and power have been analysed for 8, 16 and 32 bit versions.

Section II provides information regarding the basic components of a MAC unit and the types of multipliers used in this paper. The RTL Schematics of all the designs is presented under section III. Simulation results obtained from Xilinx

Vivado tool is provided under section IV. The comparison between the designs in terms of delay, power consumption, and FPGA utilisation parameters is also presented there. The conclusion is presented as section V.

II. COMPONENTS OF A MAC UNIT

A MAC unit consists of three components: a multiplier, an adder, and an accumulator. Words are obtained from memory locations and passed as inputs to the multiplier. The block diagram of a N-bit MAC unit is shown in Fig. 1.

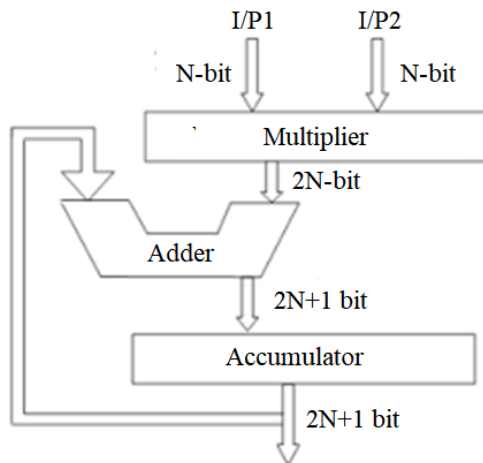


Fig. 1: Block diagram of a N-bit MAC unit

A. Adder

The adder computes the sum of the product from the multiplier and the value stored in the accumulator. The output of the adder is passed onto the accumulator. If inputs to the multiplier are of bit size N, then the adder should be of bit size 2N, producing an output of size 2N+1. Carry save adder, carry select adder, ripple carry adder(RCA), carry look-ahead adder(CLA) are among the widely used adders in the design of digital logic processing devices. Propagation delay and critical delay are two important parameters to be considered while using adders. In this paper, carry save adder is used in the design of MAC unit. It works on the principle of preserving carries until the end. It is one of the widely used circuits for implementing fast arithmetic computations. As the numbers become large, the propagation delay incurred in Ripple carry adder(RCA) and Carry Look Ahead(CLA) adder also increases. Hence, the carry-save adder is much faster than conventional adders. The block diagram of a 4-bit Carry Save Adder is shown in Fig. 2.

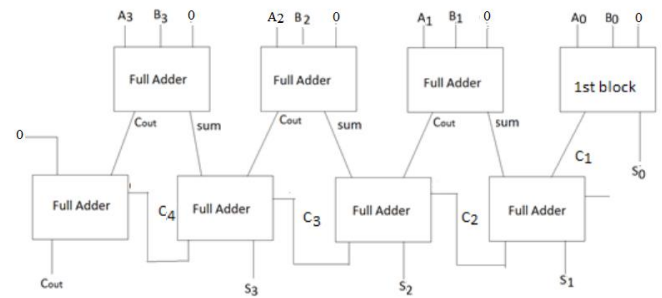


Fig. 2: Block diagram of a 4-bit Carry Save Adder

B. Accumulator

The accumulator is a register which stores the sum of products. It is widely used in Arithmetic Logic Units(ALU) and MAC units. Storing values in the accumulator can eliminate the need for additional summing operations. The response of the accumulator should be fast enough to match with fast adders. Parallel In Parallel Out(PIPO) shift registers are widely used as accumulators as they are fast and output of given within a single clock pulse. The accumulator used in this paper is a parallel in the parallel-out shift register. It is the simplest of all the four configurations of shift registers. Both data loading and data retrieval occur in parallel in a single clock pulse. Hence it serves as the ideal choice for an accumulator. The block diagram of 4-bit PIPO Shift register implemented using D flip flop is shown in Fig. 3.

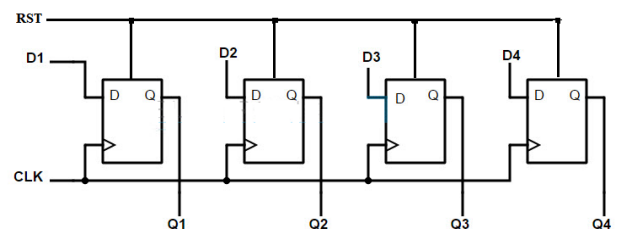


Fig. 3: Block diagram of 4-bit PIPO Shift register using D Flip Flop

C. Multiplier

Multipliers play crucial parts in today's digital signal processing systems and various other applications. Using a high-performance multiplier can boost the performance of the entire system. It is desired to have a multiplier with the following characteristics.

- It should be accurate.
- It should be able to carry out operations at a very high speed
- It should use a fewer number of slices and LUTs.
- It should consume less power.

The process of multiplication involves three steps: partial product generation, partial product reduction, and final addition. There is a wide variety of multipliers. Some of them are array, Booth, modified Booth, Wallace tree, sequential, Vedic, and DADDA multipliers. In this paper, Booth, Wallace tree, array and Vedic multipliers are used in the design of MAC unit.

1) *Booth multiplier*: Booth multiplier follows Booth's multiplication algorithm invented by Andrew Donald Booth in 1950. It multiplies two signed binary numbers in two's complement notation while preserving the sign of the result. It outperforms earlier methods of multiplication by reducing the number of iteration steps. It skims the multiplier operand and skips chains of the algorithm thus reducing the number of additions required to produce the result.

2) *Wallace tree multiplier*: It is an efficient hardware circuit designed to achieve higher speeds of operation. It was designed by Chris Wallace in 1964. It is a variant of the long multiplication method. Wallace tree reduces the number of partial products and uses carry select adder for the addition of partial products. Here, the total delay incurred is proportional to the logarithm of the length of the multiplier operand, This in turn results in faster computations. The Wallace tree method of multiplication has three steps:

- Each bit of the input is multiplied with each bit of the other input.
- The number of partial products are reduced by half by using half and full adders.
- The wires in the two inputs are grouped together and then added.

3) *Array multiplier*: Array multiplier functions based on the add-shift algorithm. It multiplies two binary numbers by using an array of half and full adders. Add and shift operations are simultaneously executed while checking the bits of the multiplier followed by the addition of partial products. This multiplier has a systematic and regular structure. However, when compared with other multipliers, it consumes larger power and suffers from long delays.

4) *Vedic multiplier*: This multiplier is based on Urdhva-Triyakbhyam Sutra, which is the most efficient one in terms of speed among the 16 Sutras. Hence, it is also referred to as UT multiplier. The Urdhva – Triyakbhyam Sutra applies to both division and multiplication. By incorporating the UT formula in the design of multipliers, high-speed digital systems can be designed. Partial product generation and addition are done concurrently. This results in the reduction of delays incurred. Vedic multipliers of larger sizes can be constructed by duplicating 2x2 Vedic multipliers and adding the products using a Ripple Carry Adder. This is shown in Fig. 5 where a 4x4 Vedic multiplier is constructed by using 2x2 Vedic multipliers and ripple carry adders. Using reversible logic gates like Peres, Feynman, DKG in the design can further aid in the reduction of power consumption. An implementation of a 2x2 Vedic multiplier using Peres and Feynman reversible gates is shown in Fig. 4.

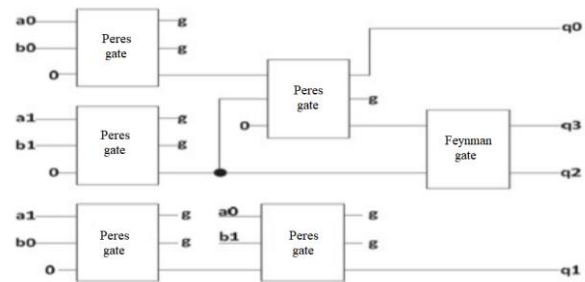


Fig.4 : 2x2 Vedic multiplier using Peres and Feynman gates

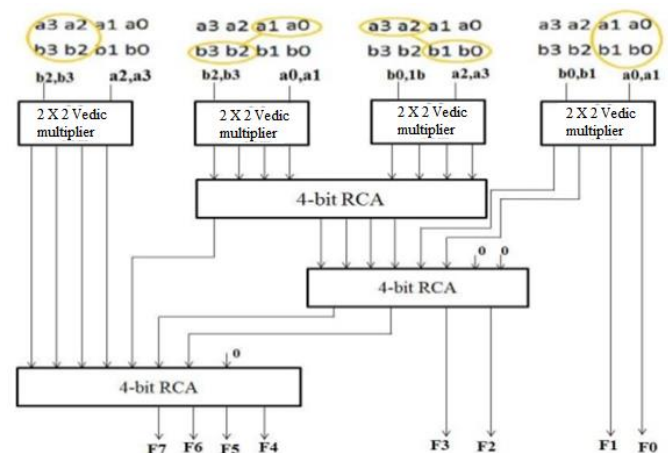


Fig. 5: 4x4 Vedic multiplier using 2x2 Vedic multipliers and Ripple Carry adders

III. DESIGN OF MAC UNIT

A. Design 1 – MAC Unit using Booth multiplier

The RTL schematic of a 4-bit MAC unit using Booth multiplier is shown in Fig. 6.

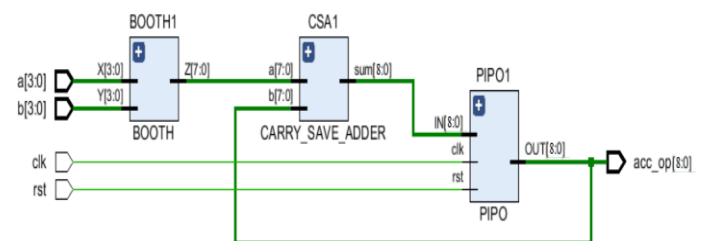


Fig. 6: RTL schematic of a 4-bit MAC unit using Booth multiplier

B. Design 2 – MAC Unit using Wallace tree multiplier

RTL Schematic of a 4 bit Wallace multiplier is shown in Fig. 7. Fig. 8 shows the RTL Schematic of a 4 bit MAC unit using Wallace multiplier.

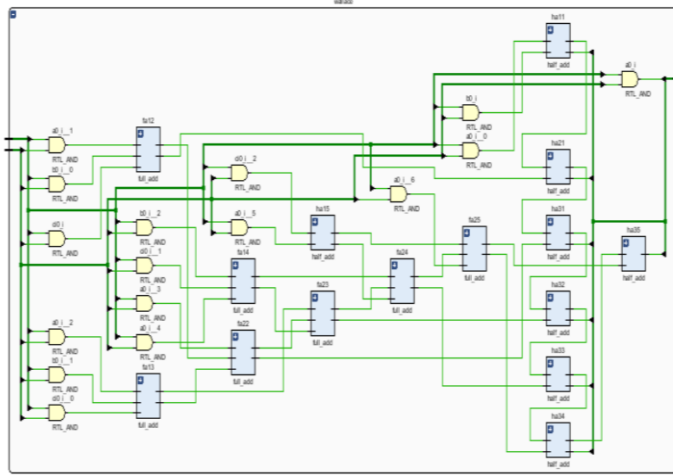


Fig. 7: RTL Schematic of 4 bit Wallace tree multiplier

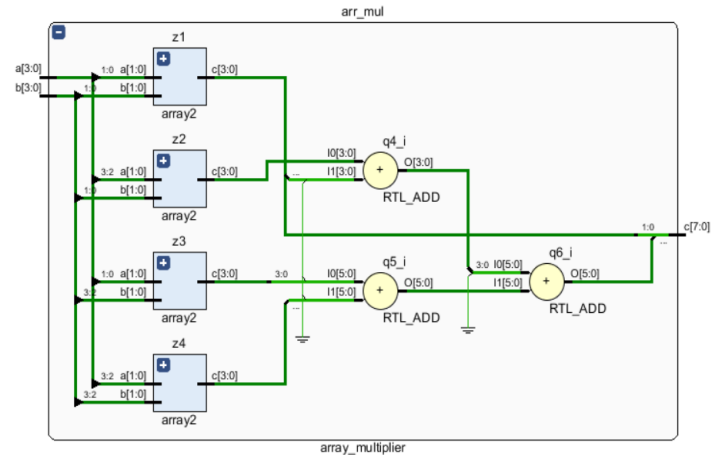


Fig. 10: RTL Schematic of 4 bit Array multiplier

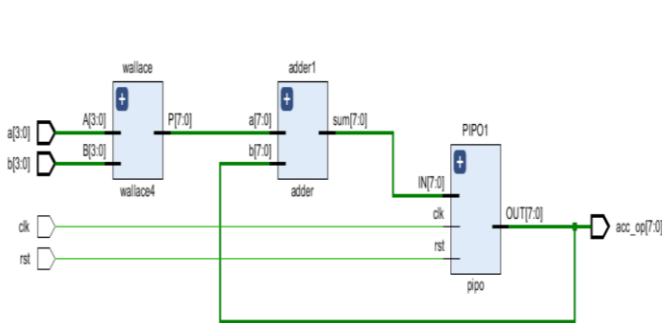


Fig. 8: RTL Schematic of 4 bit MAC unit using Wallace multiplier

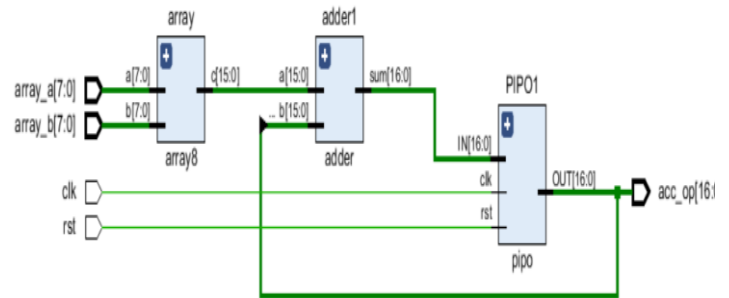


Fig. 11: RTL Schematic of a 8 bit MAC unit using Array multiplier

C. Design 3 - MAC unit using array multiplier

Array multiplier of longer bit lengths can be constructed by using multiple 2 bit array multipliers. RTL Schematic of 2 bit and 4 bit array multipliers are shown in Fig. 9 and Fig. 10. RTL Schematic of a 8 bit MAC unit using array multiplier is shown in Fig. 11.

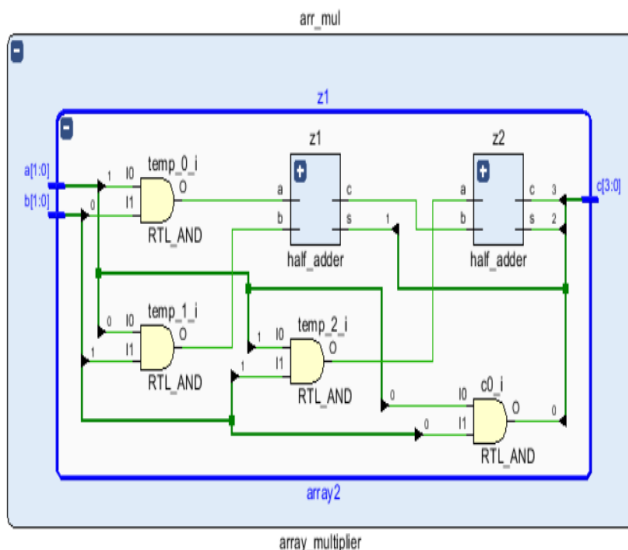


Fig. 9: RTL Schematic of 2 bit Array multiplier

D. Design 4 - MAC unit using Vedic multiplier

RTL Schematic of 2 bit and 4 bit Vedic multipliers are shown in Fig. 12 and Fig. 13. RTL Schematic of a 8 bit MAC unit using Vedic multiplier is shown in Fig. 14.

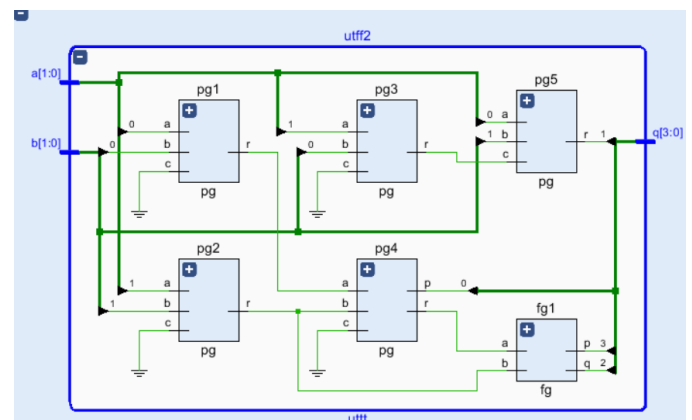


Fig. 12: RTL Schematic of 2 bit Vedic multiplier

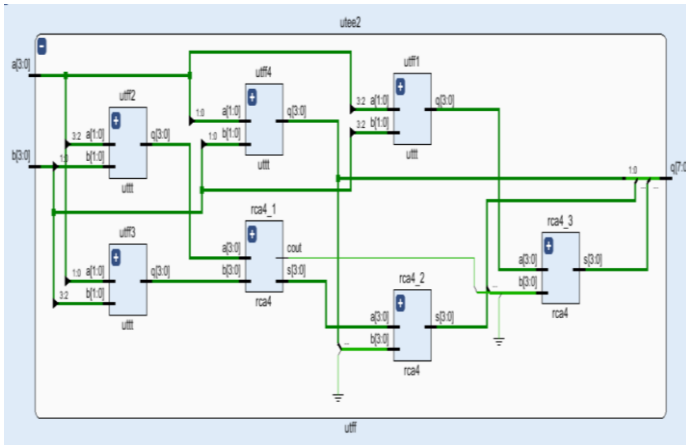


Fig. 13: RTL Schematic of 4 bit Vedic multiplier

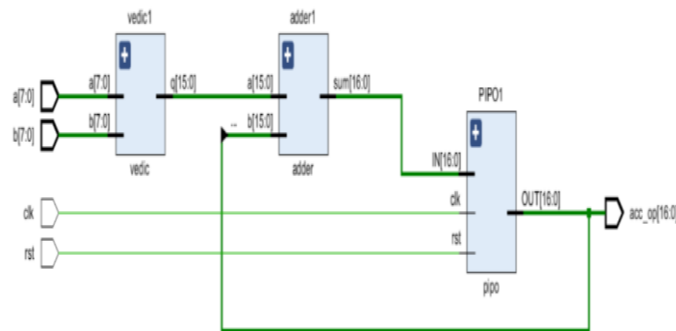


Fig. 14: RTL Schematic of a 8 bit MAC unit using Vedic multiplier

IV. SIMULATION AND RESULTS

A. Design 1: MAC unit using Booth multiplier

Simulation results for different bit lengths(4, 8 and 16) of MAC unit using Booth multiplier from Xilinx Vivado tool is provided in Table 1. Power report of 8 bit MAC unit using Booth multiplier is shown in Fig. 15. FPGA Utilisation report is shown in Fig. 16. Simulation of 16 bit MAC unit using Booth multiplier is shown in Fig. 17.

TABLE 1

Simulation results for design 1 from Xilinx Vivado tool

	4 bit	8 bit	16 bit
Total On chip Power (in W)	0.11	0.117	0.031
Static Power (in W)	0.104	0.104	0.105
Dynamic Power(in W)	0.005	0.013	0.024
Logic Power (in W)	<0.001	0.001	0.002
Delay (in ns)	3.6812	6.9625	14.969
Nets	46	69	165
Leaf Cells	28	35	88
LUTs	29	108	398

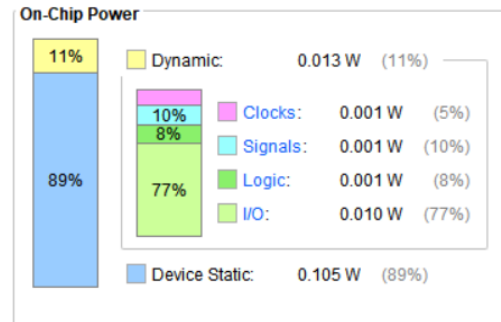


Fig. 15: Power report of 8 bit MAC unit using Booth multiplier

Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (200)	BUFGCTRL (32)
MAC_UNIT	108	16	34	1

Fig.16: Utilization report of 8 bit MAC unit using Booth multiplier

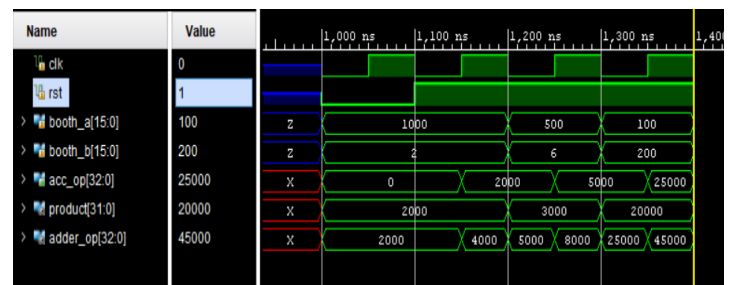


Fig. 17: Waveform of operation of 16 bit MAC unit using Booth multiplier

B. Design 2: MAC unit using Wallace tree multiplier

Simulation results for different bit lengths(4, 8 and 16) of MAC unit using Wallace tree multiplier from Xilinx Vivado tool is provided in Table 2. Power report of 8 bit MAC unit using Wallace tree multiplier is shown in Fig. 18. FPGA Utilisation report is shown in Fig. 19. Simulation of 16 bit MAC unit using Wallace tree multiplier is shown in Fig. 20.

TABLE 2

Simulation results for design 2 from Xilinx Vivado tool

	4 bit	8 bit	16 bit
Total On chip Power (in W)	0.109	0.115	0.124
Static Power (in W)	0.104	0.104	0.105
Dynamic Power(in W)	0.005	0.01	0.017
Logic Power (in W)	<0.001	0.001	0.002
Delay (in ns)	2.9175	6.132	12.793
Nets	37	69	133
Leaf Cells	19	35	67
LUTs	28	114	482

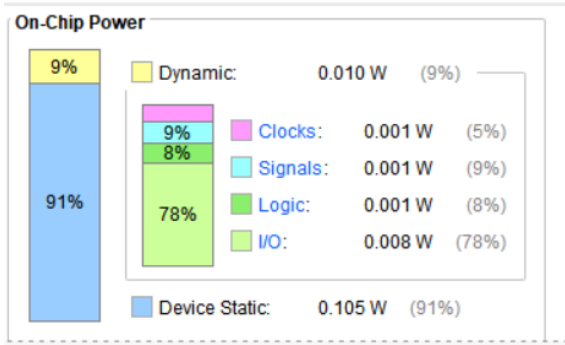


Fig. 18: Power report of 8 bit MAC unit using Wallace tree multiplier

Name	^1	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (200)
MAC_UNIT		114	16	34

Fig. 19: Utilization report of 8 bit MAC unit using Wallace tree multiplier

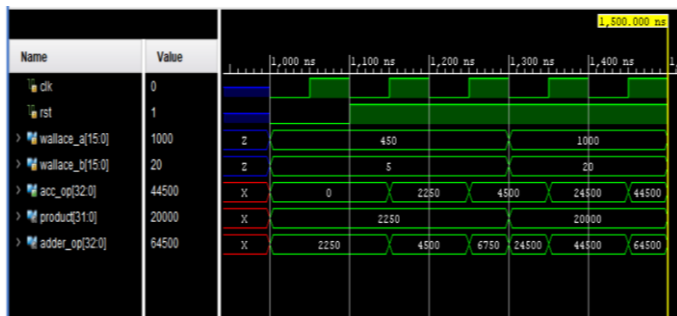


Fig. 20: Waveform of operation of 16 bit MAC unit using Wallace tree multiplier

C. Design 3: MAC unit using Array multiplier

Simulation results for different bit lengths(4, 8 and 16) of MAC unit using array multiplier from Xilinx Vivado tool is provided in Table 2. Power report of 8 bit MAC unit using array multiplier is shown in Fig. 21. FPGA Utilisation report is shown in Fig. 22. Simulation of 16 bit MAC unit using array multiplier is shown in Fig. 23.

TABLE 3

Simulation results for design 3 from Xilinx Vivado tool

	4 bit	8 bit	16 bit
Total On chip Power (in W)	0.11	0.116	0.126
Static Power (in W)	0.104	0.105	0.105
Dynamic Power(in W)	0.006	0.012	0.022
Logic Power (in W)	<0.001	0.001	0.002
Delay (in ns)	9.567	19.896	36.493
Nets	37	88	230
Leaf Cells	19	35	113
LUTs	29	114	272

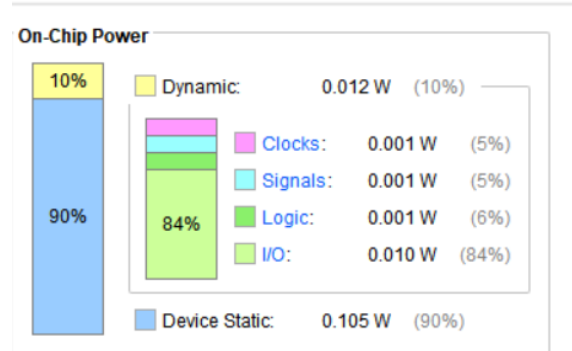


Fig. 21: Power report of 8 bit MAC unit using Array multiplier

Name	^1	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (200)
MAC_UNIT		114	16	34

Fig. 22: Utilization report of 8 bit MAC unit using Array multiplier

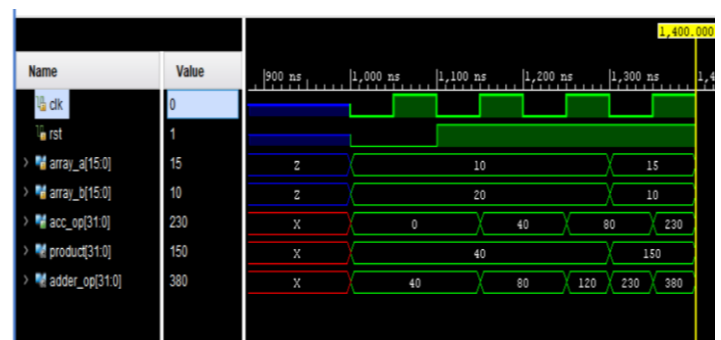


Fig. 23: Waveform of operation of 16 bit MAC unit using Array multiplier

D. Design 4: MAC unit using Vedic multiplier

Simulation results for different bit lengths(4, 8 and 16) of MAC unit using Vedic multiplier from Xilinx Vivado tool is provided in Table 2. Power report of 8 bit MAC unit using Vedic multiplier is shown in Fig. 24. FPGA Utilisation report is shown in Fig. 25. Simulation of 16 bit MAC unit using Vedic is shown in Fig. 26.

TABLE 4

Simulation results for design 4 from Xilinx Vivado tool

	4 bit	8 bit	16 bit
Total On chip Power (in W)	0.108	0.112	0.123
Static Power (in W)	0.104	0.104	0.104
Dynamic Power(in W)	0.003	0.008	0.019
Logic Power (in W)	<0.001	<0.001	<0.001
Delay (in ns)	2.9497	5.8945	11.799
Nets	37	69	139
Leaf Cells	19	45	71
LUTs	29	144	545

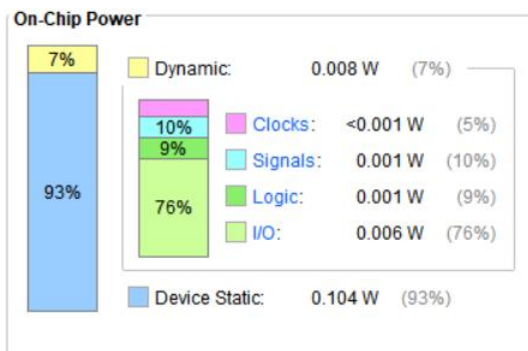


Fig. 24: Power report of 8 bit MAC unit using Vedic multiplier

Name	Slice LUTs (53200)	Slice Registers (106400)	Bonded IOB (200)
MAC_UNIT	144	16	34

Fig. 25: Utilization report of 8 bit MAC unit using Vedic multiplier

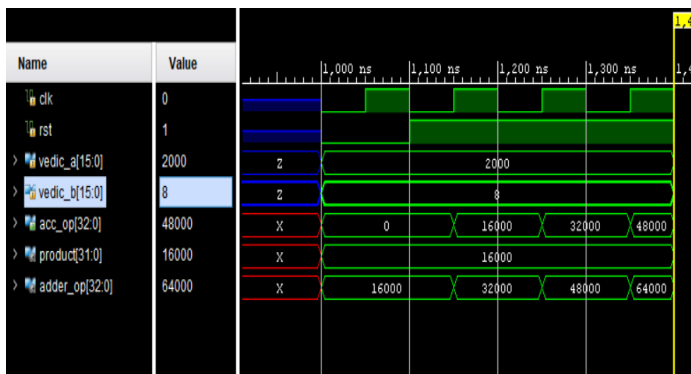


Fig. 26: Waveform of operation of 16 bit MAC unit using Vedic multiplier

TABLE 5

Comparison of total on-chip power and dynamic power between the designs (8 bit MAC unit)

	Total On-chip power (in mW)	Dynamic power (in mW)
Design 1	117	13
Design 2	115	10
Design 3	116	12
Design 4	112	8

Comparison of Total on-chip power and Dynamic power

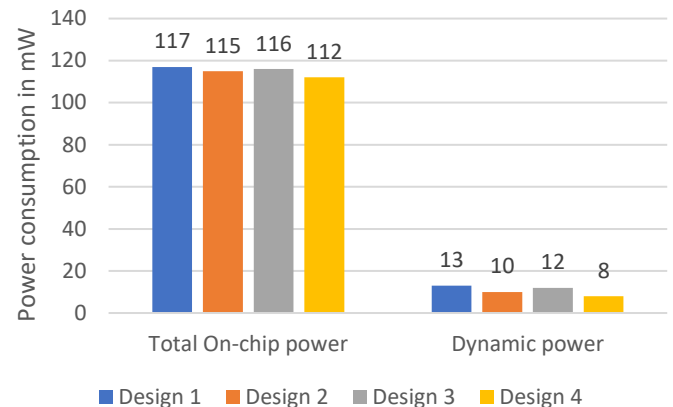


Fig. 27: Comparison chart of total on-chip power and dynamic power for 8 bit MAC unit based on the four designs

Table 5 shows the comparison of power parameters between 8 bit MAC units based on the four designs. It can be seen from tables 1 to 4 that the static power consumption remains fairly the same for all the designs. It can be understood from Table 5 and Fig. 27 that the MAC unit designed using Vedic multiplier consumes the least power. It is closely followed by design 2, design 3 and design 4.

TABLE 6

Comparison of delay incurred for 4, 8 and 16 bit MAC units based on the four designs

	4 bit (in ns)	8 bit (in ns)	16 bit (in ns)
Design 1	3.812	6.9625	14.969
Design 2	2.9175	6.132	12.7931
Design 3	9.567	19.896	36.493
Design 4	2.9497	5.8945	11.799

Comparison of delay

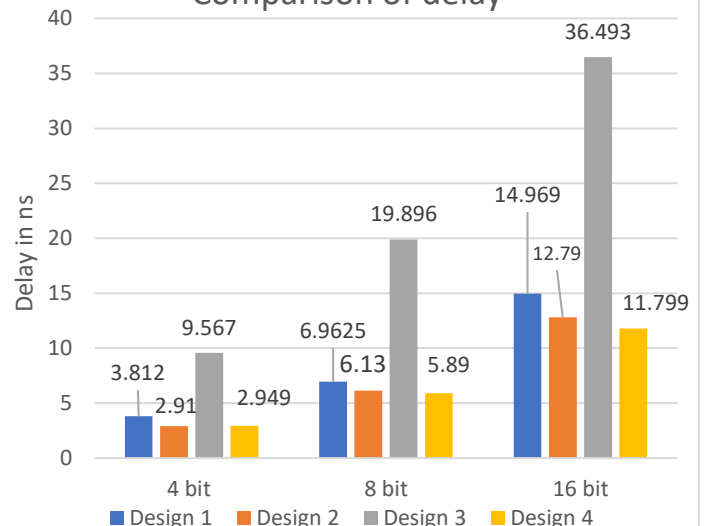


Fig. 28: Comparison chart of delay incurred for 4, 8 and 16 bit MAC units based on the four designs

Table 6 provides a comparison between the designs based on delay incurred. Based on the results as shown in Fig. 28 and tables 1-4 and 6, it is evident that MAC units designed using Vedic multiplier are the fastest. Array multiplier based MAC unit faces the longest delay among the four designs.

TABLE 7

Comparison of number of nets, leaf cells and LUTs required for 16 bit MAC unit based on the four designs

	Nets	Leaf cells	LUTs
Design 1	165	88	398
Design 2	133	67	482
Design 3	230	113	272
Design 4	133	67	545

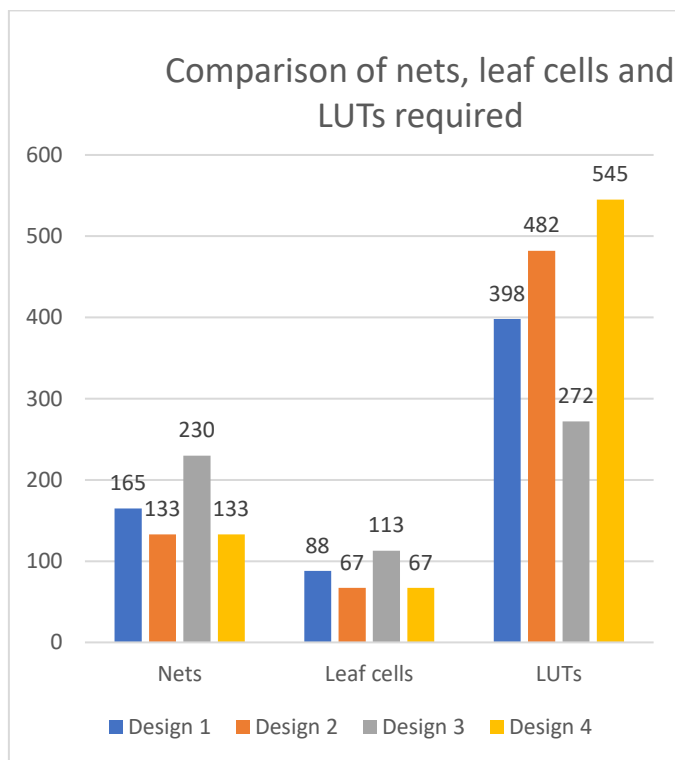


Fig. 29: Comparison chart of number of nets, leaf cells and LUTs required for 16 bit MAC unit based on the four designs

Table 7 provides a comparison between the designs in terms of nets, leaf cells and LUTs required. It is clear from Tables 1-4 and 7 that the MAC unit constructed using array multiplier requires the largest number of nets and leaf cells. However, it requires the least amount of look up tables. Designs 1, 2 and 4 require almost the same number of leaf cells and nets. Design 4 requires the largest number of LUTs.

V. CONCLUSION

In this paper, the Multiply Accumulate Unit is designed using four different multipliers in Verilog HDL, simulated and synthesized in Xilinx Vivado. These designs are compared in terms of delay incurred, power consumption and FPGA utilisation parameters. It is seen that MAC units designed using Vedic and Wallace tree multipliers have smaller delays and thus operate at higher speeds. The MAC unit designed using Vedic multiplier consumes the least dynamic power of all the other designs, however it requires the largest number of look up tables. Furthermore, it can be seen that it requires the least number of nets and leaf cells. Thus, Vedic multiplier designed using Urdhva-Tiryakbhyam Sutra is a better choice among Booth, Wallace tree, and array multipliers.

REFERENCES

- [1] S. Waser and M. J. Flynn, "Introduction to arithmetic for Digital system designers", New York: Holt, Rinehart and Winston, 1982.
- [2] Malik, Swati and Sangeeta Dhall. "Implementation of MAC unit using booth multiplier & ripple carry adder" (2012).
- [3] Pratibhadevi Tapashetti, Dr. Rajkumar, B. Kulkarni and Dr. S. S. Patil, "MAC Architectures Based on Modified Booth Algorithm", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 5, Issue 12, December 2016.
- [4] Minu Thomas, "Design and Simulation of Radix-8 Booth Encoder Multiplier for Signed and Unsigned Numbers", International Journal for Innovative Research in Science & Technology| Vol. 1, Issue 1, June 2014| ISSN(online): 2349-6010.
- [5] C. S. Wallace, "A Suggestion for a Fast Multiplier," in IEEE Transactions on Electronic Computers, vol. EC-13, no. 1, pp. 14-17, Feb. 1964, doi: 10.1109/PGEC.1964.263830.
- [6] Priyanka Mishra and Seema Nayak, "A study on Wallace tree multiplier", International Journal of Advance Research in Science and Engineering, Vol. 7, Special Issue No. 5, April 2018.
- [7] Naveen Kumar, Manu Bansal and Navnish Kumar, "VLSI Architecture of Pipelined Booth Wallace MAC Unit", International Journal of Computer Applications 57 (2012):14-18.
- [8] Design of High Speed MAC (Multiply and Accumulate) Unit Based On "Urdhva Tiryakbhyam Sutra" Parth S. Patel, Khyati K. Parasania
- [9] Asmita Haveliya, "A Novel Design for High Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)", International Journal of Technology and Engineering System (IJTES), Vol.2, No.1, Jan-March, 2011.
- [10] Sumita Vaidya and Deepak Dandekar, "Delay-Power Performance comparison of Multipliers in VLSI Circuit Design", International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010.
- [11] V. Nitin Krishna, "Design and Analysis of Arithmetic Logic Unit using Reversible Logic", Volume 8, Issue III, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 878-893, ISSN : 2321-9653, www.ijraset.com.
- [12] K. Praveen Reddy and S. Aruna Mastani, "Implementation Of High Performance 64-Bit Mac Unit For Dsp Processor", International Journal Of Current Engineering And Scientific Research, Vol.2, Issue-11, 2015.
- [13] K. Asha and Kunjan. Shinde, "Performance Analysis and Implementation of Array Multiplier using various Full Adder Designs for DSP Applications: A VLSI Based Approach", 530. 10.1007/978-3-319-47952-1_59.
- [14] S. Aruna, S. Venkatesh and K. Srinivasa Naik, "A Low Power and High Speed Array Multiplier Using On-The-Fly Conversion", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-5S4, February 2019.