FPGA and ASIC Implementation and Comparison of Multipliers

Salwa Yasmeen Neyaz*, Itisha Saxena†, Naushad Alam‡ and Syed Atiqur Rahman§
Department of Electronics Engineering, Z.H.C.E.T
Aligarh Muslim University (AMU), Aligarh, India
Email: *neyazsalwa@gmail.com, †itisha.0906@gmail.com, ‡itsnaushad@gmail.com, §atiqamu@gmail.com

***Abstract*—The multipliers find extensive use in today's digital world and are even fundamental components in many signal processing applications. This paper presents the implementation and comparison of three types of 4-bit and 8-bit multipliers i.e. Array, Wallace and Baugh Wooley. The implementation is done on both the FPGA and ASIC to compare the designs. The FPGA implementation is done on NEXYS 4 DDR which is equipped with Xilinx Artix-7 FPGA. The ASIC implementation is done using Cadence and Standard Cell Library 90 nm gpdk (generic process design kit). The results indicate that the Baugh Wooley incurs the minimum delay in the FPGA implementation while as the Wallace tree incurs minimum Power Delay Product in the ASIC implementation. The Baugh Wooley is the least area consuming architecture for both the FPGA and the ASIC implementation.**

***Index Terms*—ASIC,FPGA,HDL,RTL,STA,VLSI**

I. Introduction

Multiplication based functions are recurrently used in DSP (Digital Signal Processing) applications. In the arithmetic circuits such as ALU (Arithmetic Logic Unit) of microprocessors, the multiplication operation plays a fundamental role and the performance of many such systems is assessed on the basis of number of multiplications performed in unit time[1].

In a digital system, the presence of a multiplier in the critical path plays a dominant role in deciding the speed of operation, with the multiplier also consuming significant amount of area of the system. Hence, speed, area and power dissipation form the performance metrics used to measure the system quality of the multipliers. In order to achieve the best possible digital system, the performance metrics need to be optimized.

The implementation of the multipliers can be done either on FPGA or ASIC. The underlying difference between the ASIC and FPGA is that the circuitry is forever drawn onto the silicon whereas in FPGA it is a connection of configurable logic blocks. This leads to an enhanced performance in ASIC in terms of area, power and delay. On the other hand, the FPGA offers more flexibility and lesser implementation time. The comparative analysis of the performance design metrics obtained through ASIC and FPGA is worth exploring.

II. LITERATURE REVIEW

Various types of multipliers are discussed in the literature and compared on the basis of area, speed and power consumption. A comparative analysis of many multipliers was reported by K.N.Singh et.al [2] on the basis of delay and power dissipation, and they found that the Array multiplier was not efficient in terms of power and delay. The Booth encoded Wallace Tree multiplier performed the best amongst all.

A qualitative comparison of 4-bit Array, Wallace and Baugh Wooley multipliers has been reported recently by Shanmuganathana et.al[3]. The Wallace tree multiplier was reported to be appropriate for high speed operation while the regularity of Array multiplier was considered beneficial in terms of the layout becoming simpler.

Jaiswal et.al [4] designed a multiplier using multiplexers and XOR gates and compared the proposed structure with the Wallace Tree multiplier. The designs were synthesized in Synopsys Design Compiler using SAED 90nm CMOS technology, concluding that the proposed structure was better. Parate et.al[5] compared 4-bit signed and unsigned multipliers such as Booth, Wallace, Baugh Wooley and Array. The Booth multiplier was found to be inferior.

Pramod et.al [6] proposed a novel 4-bit Baugh Wooley multiplier architecture and compared it with the conventional Baugh Wooley.The results indicated that the novel architecture was faster but it performed poorly in terms of power. Modified algorithms for Vedic, Signed Booth and unsigned Wallace Tree was proposed by Naveen et.al [7] for reducing the delay using novel 4-2 and 5-2 type compressors. Design simulations were carried out in Cadence NC Simulator, synthesized using RTL Compiler (Standard Cell 180nm CMOS technology) and backend design using Cadence RTL to GDSII. The novel 5-2 and 4-2 type compressors incorporated in the above mentioned multipliers outperformed the existing compressor type multipliers in terms of power efficiency.

As can be seen, most of the comparison work is done either only on FPGA or only on ASIC.In this work,we present a comparison of Array,Wallace Tree and Baugh Wooley Multipliers implemented on FPGA and ASIC.

III. TYPES OF MULTIPLIERS

The multipliers that are discussed in this paper are the Array, Wallace tree and Baugh Wooley whose respective algorithms are discussed below.

A. Array Multiplier

The array multiplier is known for its high degree of uniformity making its implementation simple. Its basis is a popular "shift and add" algorithm,and the architecture is shown in Fig.1. The algorithm can be divided into three parts i.e.

978-1-7281-6564-6/20/$31.00 ©2020 IEEE

i) Partial product generation- Multiplication of multiplier bit to the multiplicand (generating first partial product, done using AND gates).

ii) Partial product accumulation-Then, second multiplier bit is multiplied to the multiplicand and the resulting partial product is shifted one bit to the left.

iii) Final addition- repeated for all bits and the result is obtained[8].

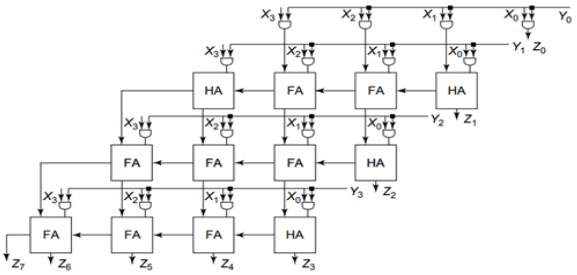

Fig. 1: Architecture of 4-bit Array multiplier [8]

B. Wallace Tree Multiplier

It is a parallel multiplier which uses the "treelike" rearrangement of the partial products. The algorithm combines three functions i.e.

i) Partial product generation beforehand for all bits

ii) Partial product reduction using half-adders and full-adders as shown in Fig.2, and

iii) Final addition using a fast adder.

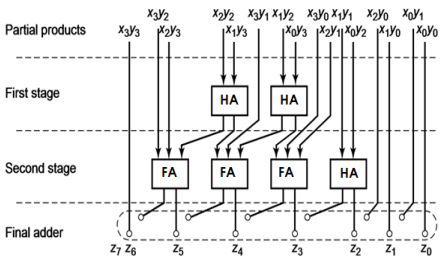

Fig. 2: Architecture of 4-bit Wallace Tree multiplier[8]

C. Baugh Wooley Multiplier

Baugh Wooley multiplier as shown in Fig.3 is used for signed multiplication of numbers. The algorithm can be divided into three stages: i) The MSB (most significant bit) of the N-1 partial product rows and all the bits of the Nth partial product row, except its MSB are inverted. ii) One ('1') is added to the Nth column. iii) MSB of the final result is reversed.

IV. METHODOLOGY

The methodology for the implementation of the multiplier architectures uses FPGA and ASIC design flow as explained in Fig. 4 and Fig. 5.

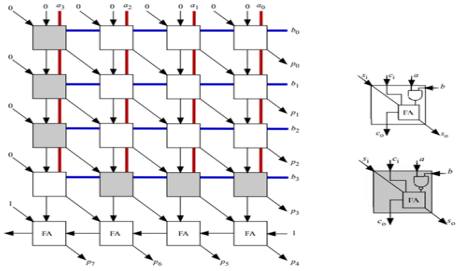

Fig. 3: Architecture of 4-bit Baugh Wooley multiplier with white and grey cells[9]

A. FPGA Design Flow in Xilinx Vivado

The multipliers are implemented on NEXYS 4 DDR board that is equipped with Artix-7 FPGA. The Xilinx Vivado 2018.2 has been used for implementing design on FPGA board. The design flow is described as under.

i) Design Entry- has been made using Verilog HDL (Hardware Description Language) and the behavior of the multipliers is described in modules. For the functional verification of the design, the behavioral simulation is carried out using a testbench. The tool generates an elaborated RTL schematic.

ii) Synthesis- A constraint file (.xdc) is added and the optimization of the design is done by analyzing the hierarchy of the design. Post-synthesis functional and timing simulation is carried to check the correctness of synthesized design[10].

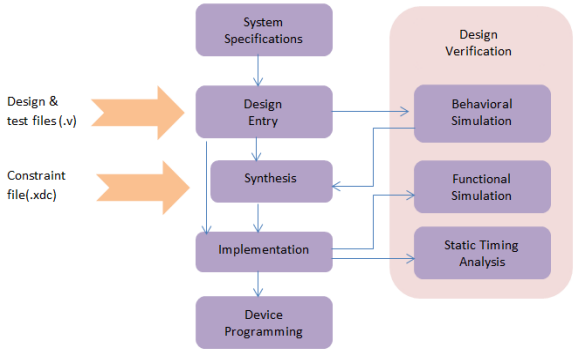

Fig. 4: FPGA Design Flow

iii) Implementation- The place and route of the synthesized netlist is done onto the FPGA device resource. The post implementation timing report is used for checking the slack of the design. If it is violated for any path, it needs to be rectified iteratively.

iv) FPGA programming and Verification- The bitstream file generated from the implementation step is loaded onto the FPGA for programming. By feeding different inputs, the functionality is tested.

B. ASIC Design Flow in Cadence

The ASIC Design Flow is divided into two parts i.e. the Frontend and Backend as shown in Fig 5. The Frontend includes the HDL coding and the synthesis part while the

Backend involves the physical design. The design flow is as under:

i) Specifications: The functionality, power and speed are some of the many specifications that are decided beforehand.

ii) HDL and Verification: The HDL step involves dividing the design into high level and low level modules using synthesizable RTL. The RTL code is then checked for correct functionality using Cadence NC Simulator.

iii)Synthesis: It refers to the mapping of the design onto target technology primitives which is 90 nm gpdk (generic process design kit) here.The constraint file (.sdc) is added which includes all the constraints of the design such as clock period, input delay, output delay, etc and a library file. If any violation is reported, the design constraints need to be tweaked accordingly[11].

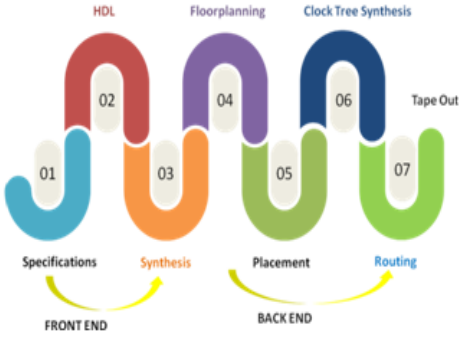

Fig. 5: ASIC Design Flow

The backend design flow is as follows:

i) Floor planning and Placement: The design floor planning in done using Cadence RTL ENCOUNTER. The core utilization, core area, power rings etc are specified here.The standard cells and pins are placed.

iii) Clock Tree Synthesis: The clock tree synthesis is done to evenly distribute clock to all the sequential elements in order to avoid clock skew.

iv) Routing and GDSII: Any buffers that are introduced after Clock Tree Synthesis are also routed and the slack violations are again checked.Then, the bitstream to be sent to the foundry is obtained.

V. IMPLEMENTATION AND RESULTS

The implementation of one of the multipliers i.e. the Wallace tree multiplier for both FPGA and ASIC is shown below.

A. FPGA Results

Behavioral Simulation of 4-bit Wallace Tree multiplier is done for many input combinations .Since both input and output are registered, hence, result is obtained after one clock cycle as can be seen in Fig. 6.

The post implementation schematic is shown in Fig.7 where all the RTL blocks are implemented using LUTs, buffers etc.

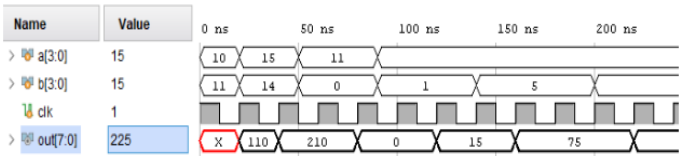

Fig. 6: Behavioral Simulation of 4-bit Wallace Tree

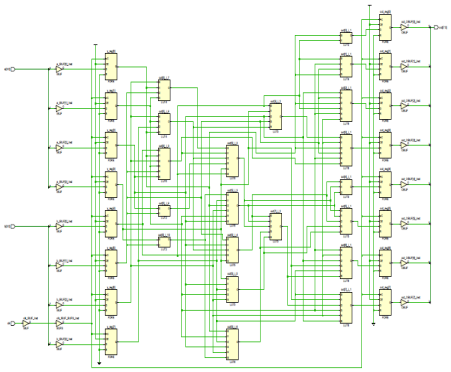

Fig. 7: Post implementation schematic of 4-bit Wallace Tree

B. ASIC Results

Functional Verification is the dynamic timing analysis where different input stimuli are applied in the form of a test bench. This is done in Cadence NC Simulator and the waveforms for the same are shown in Fig. 8.

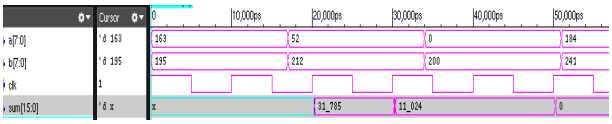

Fig. 8: Functional Verification of 8-bit Wallace Tree

The synthesized design is imported to Cadence ENCOUNTER (RTL-to-GDSII). The design is then placed and routed and a layout is obtained as shown in Fig 9.

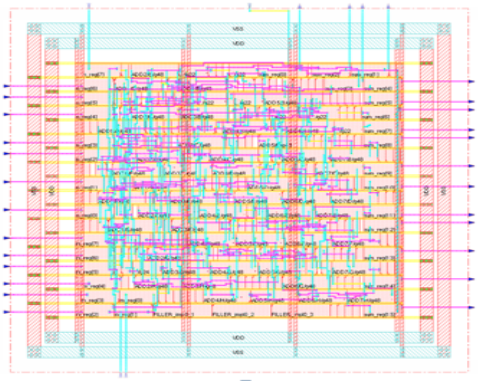

Fig. 9: Post route Layout of 8-bit Wallace Tree

The delay of the three multipliers (4 and 8 bit) post FPGA implementation shows that the Baugh Wooley has the maximum speed of operation as can be seen from Table I.

However, the PDP (Power Delay Product) calculated post ASIC implementation indicates that the Wallace tree outperforms the rest for both 4 and 8 bit. Prior to discussing the results, it is important to understand the reason for choosing only delay for FPGA. The power consumption reported in FPGA is for the entire board including the I/O, and other cores which are not used in our design unlike in case of ASIC, which reports the power for the IC of our design. The power consumption and the delay reported for every 4 and 8 bit multiplier is used to derive the power delay product for all the multipliers in ASIC.

Table I shows that Baugh Wooley has the least delay and Table II indicates that the Wallace tree multiplier has the minimum Power Delay Product.

TABLE I: Delay obtained post FPGA implementation

Type of Multiplier	4-bit (ns)	8-bit (ns)
Array	3.692	7.242
Wallace Tree	3.937	7.689
Baugh Wooley	1.682	5.86

TABLE II: Power delay Product (PDP) obtained post ASIC implementation

Type of Multiplier	4-bit (pJ)	8-bit (pJ)
Array	0.1208	1.090
Wallace Tree	0.0929	0.745
Baugh Wooley	0.0987	0.758

Another important parameter to consider is the area. For the FPGA implementation, the area is reported in terms of the resource utilization. It is defined as the percentage of used logic blocks to the total number of blocks present on the FPGA. In our case, there are 15,850 logic slices that have 4 six-input LUTS's and 8 flip-flops. The area for ASIC implementation is calculated in the Cadence Encounter after the Place and Route. The Table III and IV show the FPGA resource utilization and the area for ASIC implementation of multipliers respectively.

TABLE III: FPGA resource utilization

Type of Multiplier	4-bit (%)	8-bit (%)
Array	0.02	0.14
Wallace Tree	0.02	0.13
Baugh Wooley	0.03	0.11

V. Conclusions

In this paper, we have successfully implemented Array, Wallace Tree and Baugh Wooley multiplier (4 and 8 bit) on FPGA and ASIC. After the implementation, the performance design metrics such as Area, Delay and Power are measured for quantitative comparison between ASIC and FPGA implementations of each multiplier. The quantitative analysis shows that the minimum delay (8-bit) obtained is 5.86 ns for Baugh Wooley multiplier implementation on FPGA. In contrast to this, the PDP of the Wallace Tree multiplier is minimum of all the multipliers(8 bit and 4 bit) in the ASIC implementation. The results indicate that the ASIC implementation is suitable for highly parallel operations as in the case of Wallace Tree multiplier. The FPGA and ASIC implementation support the fact that the Baugh Wooley consumes the minimum area.

TABLE IV: Area obtained post ASIC implementation

Type of Multiplier	4-bit (μm^2)	8-bit (μm^2)
Array	1962.7	4687
Wallace Tree	1963.2	4812
Baugh Wooley	1958.1	4675

References

[1] S. Abraham, S. Kaur and S. Singh, "Study of various high speed multipliers," International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2015, pp. 1-5.

[2] K. N. Singh and H. Tarunkumar, "A review on various multipliers designs in VLSI," 2015 Annual IEEE India Conference (INDICON), New Delhi, 2015, pp. 1-4.

[3] R. Shanmuganathana and K. Brindhadevi "Comparative analysis of various types of multipliers for effective low power," Microelectronic Engineering 214, 2019, pp.28–37.

[4] K. B. Jaiswal, Nithish Kumar V, P. Seshadri and Lakshminarayanan G, "Low power wallace tree multiplier using modified full adder," 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, 2015, pp. 1-4.

[5] P. G. Parate, P. S. Patil and S. Subbaraman, "ASIC Implementation of 4 Bit Multipliers," 2008 First International Conference on Emerging Trends in Engineering and Technology, Nagpur, Maharashtra, 2008, pp. 408-413.

[6] Pramod S. A., Mukesh P. M., Manjul V. N.,and Omkar S. V., "Implementation of Baugh-Wooley Multiplier and Modified Baugh Wooley Multiplier Using Cadence (Encounter) RTL," International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 2, February 2015.

[7] Naveen Kr.G., Prabhat S.,and Jasbir K., " Implementation of Wallace Tree Multiplier Using Compressor", International Journal of Computer Technology and Applications (IJCTA), May 2012,Volume 3 (3),pp. 1194-1149.

[8] Rabaey, Jan. Digital Integrated Circuits : A Design Perspective. Englewood Cliffs, N.J. :Prentice Hall, 1996.

[9] Baugh Wooley Multiplier, University Of Victoria, 2011. Available at :https://www.ece.uvic.ca/fayez/courses/ceng465/lab_465/project2/multiplier.pdf.

[10] Himanshu Bhatnagar, "Advanced ASIC Chip Synthesis", Kluwer Academic Publishers, 2002.

[11] Ian K., and Jonathan R, "Quantifying and exploring the Gap between FPGA's and ASIC, " Springer Science and Business Media, 03-Jul-2010.

[12] Nexys4 DDR FPGA Board Reference Manual, Digilent Inc., September 11, 2014.