

report

by Matthew Taylor

Submission date: 16-May-2023 01:38PM (UTC+0100)

Submission ID: 206443047

File name: 32583_Matthew_Taylor_report_497727_730026167.docx (49.55K)

Word count: 1987

Character count: 13087



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Electronics II Report
ACDA6001

By Matthew Taylor

Tuesday, 16 May 2023

TABLE OF CONTENTS

Introduction	3
Literature Review	4
PFGA Architecture	4
Designing a FPGA	4
Programming a FPGA.....	4
FPGA Applications	5
FPGA Advancements.....	5
FPGA Limitations	5
FPGA Alternatives	5
Comparison between Braun's And Wallace Tree Multiplier	7
References	9

INTRODUCTION

As part of this report, I will discuss Field-Programmable Gate Arrays (FPGA) and explore their architecture, design flow, programming methodologies, applications and challenges. FPGAs are flexible integrated circuits that offer programmable hardware solutions for a wide range of applications.

FPGAs have emerged as versatile devices that enable rapid prototyping, customisation, and reconfigurability in digital logic design. Unlike Application-Specific Integrated Circuits (ASICs), FPGAs offer flexibility and adaptability, making them suitable for a diverse set of applications.



LITERATURE REVIEW

PFGA ARCHITECTURE

The architecture of FPGAs includes various components that collectively provide the required functionality. These components include Look-Up Tables (LUTs), programmable interconnects, input/output (I/O) blocks, and embedded memory. Configurable logic blocks (CLBs) and routing resources further enhance the flexibility of FPGAs. [1] [2]

DESIGNING A FPGA

The process of designing FPGAs involves a series of steps that turn the design specifications into a working implementation. The process typically includes design entry, synthesis, placement and routing, and configuration. In the design process, Hardware Description Languages (HDLs) like VHDL and Verilog, along with schematic-based entry and high-level synthesis (HLS), are essential components. [3]

PROGRAMMING A FPGA

Field-Programmable Gate Arrays (FPGAs) can be programmed using various methods depending on the design requirements and the preferred programming approach. Here are some common methods used to program FPGAs:



1. **Hardware Description Languages (HDLs):** HDLs, such as VHDL (VHSIC Hardware Description Language) and Verilog, are widely used to describe the functionality of digital circuits. These languages allow designers to specify the behaviour and structure of the desired circuit. HDL-based programming enables a detailed and precise representation of the design, making it suitable for complex designs and customisation.
2. **Schematic Entry:** Schematic entry provides a graphical representation of the digital circuit using symbols and connections. It allows designers to draw the circuit using software tools that provide a schematic editor.
3. **High-Level Synthesis (HLS):** High-Level Synthesis is an approach that enables designers to describe the desired functionality of the design at a higher level of abstraction, such as C/C++. HLS allows for faster design iterations and easier algorithmic exploration, making it suitable for complex designs and system-level optimisations.

FPGA APPLICATIONS

FPGAs are applicable in a diverse set of fields due to their ability to perform parallel processing, handle real-time computations, and deliver low-latency responses. They can be found in applications including digital signal processing, communication systems, image and video processing, cryptography, and embedded systems. [4] [5]

FPGA ADVANCEMENTS

FPGA technology continues to advance, bringing improvements in performance, power efficiency, and capabilities. Notable advancements include the integration of computing elements, such as CPUs and GPUs, within FPGAs. This integration allows for the development of highly efficient and flexible systems. Additionally, the emergence of high-capacity FPGAs enables the implementation of larger and more complex designs. Specialised architectures tailored to specific applications, such as neural network accelerators, have also gained traction. [6] [7]

FPGA LIMITATIONS

Despite their numerous advantages, FPGAs face challenges and limitations that should be considered when evaluating their suitability for a given project. Power consumption is a significant concern, as FPGAs typically consume more power compared to ASICs. Managing power consumption while maximising performance is an ongoing challenge in FPGA design. Design productivity and the learning curve associated with FPGA development can also pose obstacles for designers, particularly those new to FPGA programming like myself. The complexity of FPGA designs increases with the size and scale of the project, making scalability an important consideration.

FPGA ALTERNATIVES

There are several alternatives to Field-Programmable Gate Arrays (FPGAs) that offer different approaches and advantages in various applications. Here are some notable alternatives:

1. **Application-Specific Integrated Circuits (ASICs):** ASICs are custom-designed integrated circuits that are specifically tailored to perform a particular function or set of functions. Unlike FPGAs, ASICs are fabricated for a specific application, resulting in optimized performance, power efficiency, and area use. ASICs offer higher performance and lower power consumption compared to FPGAs, making them suitable for high-volume production and applications with strict performance requirements. However, ASIC development requires significant upfront costs, longer design cycles, and limited flexibility for design modifications.

2. **Digital Signal Processors (DSPs):** DSPs are specialised microprocessors designed for digital signal processing tasks. They offer dedicated hardware resources and optimised instructions for efficient signal processing operations, such as filtering, modulation, and Fourier transforms. DSPs provide high computational performance, flexibility, and real-time processing capabilities. While they may not offer the same level of reconfigurability as FPGAs, DSPs excel in applications that require complex signal processing algorithms, such as audio and video processing, telecommunications, and sensor data analysis.
3. **Graphics Processing Units (GPUs):** Originally designed for graphics rendering, GPUs have evolved into highly parallel processors that excel in data-parallel computations. GPUs are optimised for accelerating graphics rendering, but their parallel architecture can also be leveraged for general-purpose computing tasks. They provide massive parallelism and high throughput, making them well-suited for applications like scientific simulations, machine learning, and data analytics. GPUs offer a balance between performance and flexibility, although they may not provide the same level of configurability as FPGAs.
4. **System-on-Chip (SoC) Platforms:** SoC platforms integrate multiple components, including processors, memory, I/O interfaces, and specialised hardware accelerators, onto a single chip. These platforms combine the advantages of custom hardware accelerators with the flexibility of programmable processors. SoCs provide a balance between performance, power efficiency, and flexibility, making them suitable for a wide range of applications. They are commonly used in embedded systems, mobile devices, and Internet of Things (IoT) applications.
5. **Microcontrollers:** Microcontrollers are compact integrated circuits that combine a microprocessor core with memory, I/O interfaces, and other peripheral components. They are widely used in embedded systems and applications with low-power requirements. Microcontrollers offer low-cost solutions with low power consumption, making them suitable for applications such as consumer electronics, automotive systems, and industrial automation. However, they have limited computational capabilities and are less flexible compared to FPGAs.
6. **Custom Processor Designs:** In some cases, custom processor designs can be developed to meet specific application requirements. This approach involves designing a processor architecture tailored to the specific needs of the application, including custom instructions and hardware accelerators. Custom processor designs offer a high level of optimisation and performance but require significant expertise and development effort.

3 COMPARISON BETWEEN BRAUN'S AND WALLACE TREE MULTIPLIER

Parallel Array Braun's Multiplier and Wallace Tree Multiplier are two commonly used architectures for implementing multipliers in digital circuits. Both architectures are known for their ability to perform high-speed multiplication operations.

Parallel Array Braun's Multiplier: The Parallel Array Braun's Multiplier is a hardware multiplier architecture that utilises a parallel array of adders to perform multiplication. It is based on the concept of shifting and adding intermediate products to obtain the final result. The multiplier operates in parallel, with each bit of the multiplicand being multiplied with all the bits of the multiplier simultaneously. The partial products generated for each multiplication operation are then added together using a series of full adders. This parallel processing enables high-speed multiplication.

Wallace Tree Multiplier: The Wallace Tree Multiplier, also known as the Modified Booth Encoder, is another widely used multiplier architecture. It uses a combination of compressors and carry-save adders to reduce the number of partial products and minimise the number of stages required for addition. The multiplier operates by decomposing the multiplier into smaller sections, reducing the overall complexity of the multiplication process. The Wallace Tree Multiplier employs a tree-like structure, with the partial products generated at each stage being progressively added using compressors and carry-save adders until the final result is obtained. [8]

Both the Parallel Array Braun's Multiplier and the Wallace Tree Multiplier architectures aim to achieve high-speed multiplication operations. They both break down the multiplication process into smaller steps to reduce complexity and improve efficiency. Additionally, both architectures rely on the use of adders to perform the intermediate and final addition stages. Both multipliers are well-suited for applications that require fast multiplication, such as digital signal processing and arithmetic calculations. [9] [10]

Differences:

1. **Complexity:** The Parallel Array Braun's Multiplier has a relatively simpler structure compared to the Wallace Tree Multiplier. It uses a parallel array of adders, resulting in a regular and straightforward layout. In contrast, the Wallace Tree Multiplier involves the use of compressors and carry-save adders, which adds complexity to the design. This increased complexity allows for a reduced number of partial products and stages, leading to improved performance.
2. **Area Efficiency:** In terms of area efficiency, the Wallace Tree Multiplier generally outperforms the Parallel Array Braun's Multiplier. The Wallace Tree architecture offers better use of resources by reducing the number of partial products and stages. This results in a more compact design and a smaller overall footprint compared to the Parallel Array Braun's Multiplier.

3. **Power Consumption:** The power consumption of the two architectures can vary depending on the specific implementation and technology used. However, in general, the Wallace Tree Multiplier tends to consume less power compared to the Parallel Array Braun's Multiplier. This is due to the reduced number of partial products and stages, resulting in lower switching activity and reduced power dissipation.
4. **Delay and Speed:** The Parallel Array Braun's Multiplier architecture typically shows lower delay and faster multiplication speed compared to the Wallace Tree Multiplier. The parallel processing nature of the Parallel Array Braun's Multiplier allows for simultaneous multiplication and addition, resulting in reduced propagation delay. On the other hand, the Wallace Tree Multiplier involves multiple stages of addition, leading to increased propagation delay.



Both the Parallel Array Braun's Multiplier and Wallace Tree Multiplier offer efficient solutions for performing high-speed multiplication operations in digital circuits. The Parallel Array Braun's Multiplier provides a simpler and more straightforward design, while the Wallace Tree Multiplier offers better area efficiency and lower power consumption.

The choice between these architectures depends on specific design requirements. If the priority is to achieve faster multiplication speed with relatively lower concern for area utilisation, the Parallel Array Braun's Multiplier may be preferred. On the other hand, if area efficiency and power consumption are critical considerations, the Wallace Tree Multiplier is a suitable choice.

In conclusion, both the Parallel Array Braun's Multiplier and Wallace Tree Multiplier are effective multiplier architectures, each with its own advantages. Designers should carefully evaluate their specific requirements and trade-offs to select the most appropriate multiplier architecture for their applications.

REFERENCES

- [1] Arrow.com, “FPGA basics: Architecture, applications and uses,” 24 9 2018. [Online]. Available: <https://www.arrow.com/en/research-and-events/articles/fpga-basics-architecture-applications-and-uses>. [Accessed 16 5 2023].
- [2] WatElectronics.com, “Basic FPGA Architecture and its Applications,” 30 7 2019. [Online]. Available: <https://www.watelectronics.com/fpga-architecture-applications/>. [Accessed 15 5 2023].
- [3] Xilinx.com, “Programming an FPGA: An Introduction to How It Works,” [Online]. Available: <https://www.xilinx.com/products/silicon-devices/resources/programming-an-fpga-an-introduction-to-how-it-works.html>. [Accessed 11 5 2023].
- [4] Spiceworks.com, “What is PFPGA,” [Online]. Available: <https://www.spiceworks.com/tech/networking/articles/what-is-fpga/>. [Accessed 12 5 2023].
- [5] Xilinx.com, “What is an FPGA,” [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html>. [Accessed 12 5 2023].
- [6] A. Klotz, “AMD Patent Shows Off Potential CPU With Integrated FPGA,” 5 1 2021. [Online]. Available: <https://www.tomshardware.com/news/amd-patent-shows-CPU-FPGA-integration>. [Accessed 13 5 2023].
- [7] Arrow.com, “FPGA vs CPU vs GPU vs Microcontroller: How Do They Fit into the Processing Jigsaw Puzzle?,” 5 10 2018. [Online]. Available: [FPGA vs CPU vs GPU vs Microcontroller: How Do They Fit into the Processing Jigsaw Puzzle?](#). [Accessed 13 5 2023].
- [8] K. G. S. Himanshu Bansal and T. Sharma, “Wallace Tree Multiplier Designs: A Performance Comparison,” *Innovative Systems Design and Engineering*, vol. Vol.5, no. ISSN 2222-1727, 2014.
- [9] A. LAHOTI, “Comparison between Braun Multiplier and Wallace Multiplier technique,” 3 3 2021. [Online]. Available: <https://medium.com/multiplier-techniques-in-vlsi/comparison-between-braun-multiplier-and-wallace-multiplier-technique-c26c38a17c69>. [Accessed 14 5 2023].
- [10] V. B. Biradar, P. G. Vishwas, C. S. Chetan and B. S. Premananda, “Design and performance analysis of modified unsigned braun and signed Baugh-Wooley multiplier,” in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECOT)*, Mysuru, India, 2017.

FINAL GRADE

21 / 100

GENERAL COMMENTS

Instructor

a. Introduction to Multipliers/ Various Multipliers/ Advantages and Disadvantages/ Applications of Multipliers :2

The work is very General. It could be better if more points are added in relation with the multipliers. The introduction talks about FPGA. But the assessment task has not been identified and diverted from what is required.

b. Background Research :6

The Main parameters are taken for comparisons. It is clear that you have made a research on the multipliers. Theory based comparisons are made while the simulations were missing.

c. Diversity of sources and Referencing Skills :2

Use of referencing, with no/very few inaccuracies or inconsistencies.

a. Task 1: Identifying the correct Multiplier (4), Architecture of Brauns Multiplier :1

The identified Multiplier is not done. Very limited discussions were made related to the concepts of Multipliers. There is no architecture diagram is found

b. Task 2: Dot Diagram(4), Architecture of Wallace Multiplier :1

No dot Diagram found. Very limited discussions were made

related to the concepts of Multipliers. There is no architecture diagram is found

a. Finding

Critical Path, Longest Delay, Min and Max Delay, Power Consumption :0 No
Analysis Found

b. Innovative

thinking in problem solving and feature additions :2

Generalised statements made with scant evidence. Ideas sometimes illogical and contradictory.

a. Simulation :0

No evidence of Simulation is found.

b. VHDL Coding's :0

The codes are not relevant to the Module. Very Poor Assumptions are made.

b. Circuit

Schematic :0

No Schematic Found.

d. FPGA Synthesis :0

No evidence of FPGA is found.

a. Logical order

of ideas, Flow of ideas in the report from the beginning to end with appropriate chapters, sub chapters and continuation :3

The flow of the work is good. But many gaps are found. There is no connection with the assessment Tasks.

**b. Definitions,
Acronyms and Abbreviations**
:2

Definitions, Acronyms and Abbreviations are captured but limited

**c.
Professionalism, Spelling and Grammar,
Complete Report, Proper
Presentation, Quality of the report**
:2

Spelling and Grammar mistakes are limited.
Quality of the
report needs to be improved.

Total Marks :21

[Click here to view the Feed back Report](#)

PAGE 1

PAGE 2

PAGE 3



Comment 1

The work is very General. It could be better if more points are added in relation with the multipliers. The introduction talks about FPGA. But the assessment task has not been identified and diverted from what is required.- 2 Marks

PAGE 4



Comment 2

The same applies here. The literature review is done on a deviated topic and the review of the main topic (Braun's and Wallace) is missing. You might have done the literature review on essential reading. (2 Marks)

PAGE 5

PAGE 6



Comment 3

Its Good to do a comparisons on both the Multipliers.



Comment 4

The Main parameters are taken for comparisons. It is clear that you have made a research on the multipliers. Theory based comparisons are made while the simulations were missing