

Electronics 2

by Ryan Stapleton

Submission date: 15-May-2023 09:23PM (UTC+0100)

Submission ID: 206335516

File name: 34166_Ryan_Stapleton_Electronics_2_497727_866921712.docx (586.03K)

Word count: 3889

Character count: 22943

2023

Electronics 2

ASSIGNMENT 1 – DIGITAL MULTIPLIERS
RYAN STAPLETON (2108541)

SAFRAN SEATS GB |

Contents

Table of Figures.....	2
Introduction	3
Background Research.....	4
Types of Digital Multipliers	5
8x8 Braun Multiplier	6
Wallace Tree Multiplier.....	6
Circuit Schematics	7
8x8 Braun Multiplier	7
Pseudocode for 8x8 Braun Multiplier	7
8x8 Braun Multiplier Logic Diagram.....	8
8x8 Wallace Tree Multiplier	9
Pseudocode for 8x8 Wallace Tree Multiplier.....	9
8x8 Wallace Tree Multiplier Logic Diagram	10
Circuit Simulations	11
8x8 Braun Multiplier	11
8x8 Wallace Tree Multiplier	13
Timings Summary.....	15
8x8 Braun's Multiplier Timing Summary.....	15
8x8 Wallace Tree Multiplier Timing Summary.....	16
Digital Multiplier Improvements.....	17
Conclusion.....	18
References	19

Table of Figures

Figure 1- 8x8 Braun Multiplier Logic Diagram	8
Figure 2- 8x8 Wallace Tree Multiplier Logic Diagram	10
Figure 3- 8x8 Braun Multiplier FPGA Overview	11
Figure 4- 8x8 Braun Multiplier FPGA Implementation	12
Figure 5- 8x8 Braun Multiplier Power Report.....	12
Figure 6- 8x8 Wallace Tree Multiplier FPGA Overview	13
Figure 7- 8x8 Wallace Tree Multiplier FPGA Implementation	14
Figure 8- 8x8 Wallace Tree Multiplier Power Report	14
Figure 9- 8x8 Braun Multiplier FPGA Longest Path.....	15
Figure 10- 8x8 Braun Multiplier FPGA Shortest Path.....	15
Figure 11- 8x8 Wallace Tree Multiplier FPGA Longest Path	16
Figure 12- 8x8 Wallace Tree Multiplier FPGA Shortest Path	16

Introduction

Digital multipliers are essential building blocks in modern digital signal processing (DSP) systems, [1] enabling efficient multiplication of digital values in various applications such as digital filters, fast Fourier transforms (FFT), convolutional neural networks (CNN), and other digital algorithms. They are widely used in diverse fields such as telecommunications, image processing, audio processing, and sensor networks, among others. Digital multipliers play a crucial role in enhancing the performance and functionality of these systems, making them an integral part of modern electronic devices and communication systems.

Digital multipliers perform the operation of multiplication on digital signals, which are discrete-time signals represented in binary form. They are designed to provide accurate multiplication results while minimizing power consumption, area, and delay. Digital multipliers operate on binary numbers using bitwise operations such as AND, OR, XOR, and bit shifting, and employ arithmetic and logical operations to achieve accurate and efficient multiplication of digital values.

The complexity of digital multipliers depends on the number of bits in the operands and the desired level of accuracy. Commonly used digital multipliers include single-bit multipliers, partial products multipliers, Booth-encoded multipliers, Wallace tree multipliers, and carry-save array multipliers. These multipliers employ different techniques and architectures to optimize performance in terms of speed, power consumption, and area, depending on the specific requirements of the application.

One of the key challenges in designing digital multipliers is achieving high-speed multiplication while minimizing power consumption. High-speed multiplication is crucial in applications where real-time processing is required, such as in wireless communication systems, image and video processing, and digital audio processing. Power consumption is a critical consideration in portable devices and battery-powered systems, where energy efficiency is paramount to extend battery life and reduce operational costs. Hence, digital multiplier design techniques often involve trade-offs between speed, power consumption, and area, depending on the specific constraints of the application.

Another important aspect of digital multipliers is their accuracy. The accuracy of digital multipliers depends on various factors, including the precision of the operands, the word length of the multiplier, and the rounding and truncation methods employed. Achieving high accuracy in multiplication results is crucial in applications such as scientific computing, cryptography, and high-resolution image processing, where precise numerical computations are required.

The advancements in semiconductor technology, such as the scaling of complementary metal-oxide-semiconductor (CMOS) transistors, have greatly impacted the design and implementation of digital multipliers. The shrinking feature sizes of CMOS transistors have led to increased transistor density, higher clock frequencies, and reduced power supply voltages, enabling the integration of complex digital multiplier architectures on a single chip. This has resulted in significant improvements in the performance, power consumption, and area of digital multipliers, making them more efficient and cost-effective for a wide range of applications.

Background Research

Digital multipliers are essential components in digital circuits that perform the multiplication of two binary numbers. They are used in a wide range of applications, including digital signal processing, graphics processing, and cryptography.

Digital multipliers come in various types and architectures, ranging from simple array multipliers to advanced multipliers that use parallelism, pipelining, and optimization techniques to achieve high-speed and power efficiency.

[2] One of the simplest types of digital multipliers is the array multiplier. It uses an array of logic gates to perform multiplication. The input numbers are broken down into individual bits, which are then multiplied and added together to produce the final result. Array multipliers are easy to design and implement, but they can be slow and power-hungry, especially when dealing with large numbers.

To improve the performance of digital multipliers, more advanced architectures have been developed. [3] One such architecture is the Booth multiplier, which uses a combination of shifting and addition operations to perform multiplication. The Booth multiplier reduces the number of partial products generated, resulting in faster and more efficient multiplication. Booth multipliers are commonly used in digital signal processing and graphics processing applications.

Another advanced multiplier architecture is the Wallace Tree multiplier. [4] The Wallace Tree multiplier uses a combination of partial products and carry-save adders to reduce the number of full adders required for the multiplication operation. This results in faster and more efficient multiplication. Wallace Tree multipliers are commonly used in high-performance digital signal processing applications.

[5] The Baugh-Wooley multiplier is another type of array multiplier that uses a combination of XOR and XNOR gates to perform multiplication. This method reduces the number of partial products generated, resulting in faster multiplication. Baugh-Wooley multipliers are commonly used in low-power applications where speed is not the highest priority.

[6] The Dadda multiplier is a more advanced multiplier that uses a combination of partial products and carry-select adders to reduce the number of full adders required for the multiplication operation. This results in faster and more efficient multiplication. Dadda multipliers are commonly used in high-performance digital signal processing and graphics processing applications.

[7] The Karatsuba multiplier is a more advanced multiplier that uses a divide-and-conquer algorithm to perform multiplication. This method reduces the number of operations required to perform multiplication, resulting in faster multiplication. Karatsuba multipliers are commonly used in cryptography applications where speed and security are critical.

In addition to these architectures, digital multipliers can be further optimized using techniques such as parallelism, [8] pipelining, and low-power circuit design. Parallelism involves breaking down the input numbers into smaller segments and processing them simultaneously, which can significantly increase the speed of multiplication. Pipelining involves breaking down the multiplication operation into multiple stages that are executed in sequence. This allows for faster processing of multiple operations simultaneously, resulting in higher speed. Low-power circuit design techniques, such as voltage scaling, clock gating, and power gating, help to reduce power consumption without sacrificing performance.

Digital multipliers are used in a wide range of applications, including digital signal processing, graphics processing, cryptography, and more. They are essential components in modern digital circuits, and their performance and efficiency continue to improve with advancements in semiconductor technology and design techniques.

In conclusion, digital multipliers are critical components in digital circuits that perform the multiplication of two binary numbers. They come in various types and architectures, ranging from simple array multipliers to advanced multipliers that use parallelism, pipelining, and optimization techniques to achieve high-speed and power efficiency. The choice of which type of multiplier to use depends on the specific requirements of the application in terms of speed, accuracy, and complexity.

Types of Digital Multipliers

There are several different types of digital multipliers, each with its own strengths and weaknesses. Here are some of the most common types of digital multipliers:

- **Array Multiplier:** An array multiplier is a simple multiplier that uses an array of logic gates to perform multiplication. The input numbers are broken down into individual bits, which are then multiplied and added together to produce the final result.
- **Booth Multiplier:** A Booth multiplier is a more advanced multiplier that uses a combination of shifting and addition operations to perform multiplication. This method reduces the number of partial products generated, resulting in faster and more efficient multiplication.
- **Wallace Tree Multiplier:** A Wallace Tree multiplier is a high-speed multiplier that uses a combination of partial products and carry-save adders to reduce the number of full adders required for the multiplication operation. This results in faster and more efficient multiplication.
- **Baugh-Wooley Multiplier:** A Baugh-Wooley multiplier is a type of array multiplier that uses a combination of XOR and XNOR gates to perform multiplication. This method reduces the number of partial products generated, resulting in faster multiplication.
- **Dadda Multiplier:** A Dadda multiplier is a more advanced multiplier that uses a combination of partial products and carry-select adders to reduce the number of full adders required for the multiplication operation. This results in faster and more efficient multiplication.
- **Karatsuba Multiplier:** A Karatsuba multiplier is a more advanced multiplier that uses a divide-and-conquer algorithm to perform multiplication. This method reduces the number of operations required to perform multiplication, resulting in faster multiplication.

These are just a few of the many different types of digital multipliers that are commonly used in various applications. The choice of which type of multiplier to use depends on the specific requirements of the application in terms of speed, accuracy, and complexity.

The Braun Multiplier and Wallace Tree Multiplier are both types of binary multipliers that are used to perform multiplication of two binary numbers. However, they differ in their implementation and performance characteristics.

The Braun Multiplier is a type of binary multiplier that uses a combination of logic gates to perform the multiplication operation. It is a relatively simple and low-cost multiplier that is commonly used in digital signal processing applications where speed is not the highest priority. The Braun multiplier is known for its simplicity and ease of implementation, but it can be slower than other more advanced multiplier designs.

On the other hand, the Wallace Tree Multiplier is a more advanced and efficient binary multiplier that uses a combination of partial products and carry-save adders to reduce the number of full

adders required for the multiplication operation. It is a high-speed multiplier that is commonly used in digital signal processing applications where speed is critical. The Wallace tree multiplier is known for its speed and efficiency, but it can be more complex and costly to implement than the Braun multiplier.

In summary, the Braun Multiplier is a simpler and low-cost multiplier that can be slower than other more advanced multiplier designs, while the Wallace Tree Multiplier is a more advanced and efficient multiplier that is faster but more complex and costly to implement. The choice between these two multiplier designs depends on the specific requirements of the application in terms of speed, cost, and complexity.

8x8 Braun Multiplier

An 8x8 Braun multiplier is a circuit that performs multiplication of two 8-bit binary numbers. The Braun multiplier is a type of binary multiplier that uses a combination of logic gates to perform the multiplication operation. The 8x8 Braun multiplier is designed to accept two 8-bit binary numbers as input and generate a 16-bit binary output representing the product of the two numbers.

[9]The function of an 8x8 Braun multiplier is to provide a fast and efficient method for multiplying two 8-bit binary numbers. This type of multiplier is commonly used in digital signal processing applications, such as in digital filters, modems, and other signal processing systems.

The 8x8 Braun multiplier performs the multiplication operation using a series of logic gates that implement the multiplication algorithm. The result of the multiplication is then fed to a series of adders and accumulators to generate the final output.

Overall, the function of an 8x8 Braun multiplier is to provide a high-speed, low-cost method for performing multiplication of two 8-bit binary numbers in digital signal processing applications.

Wallace Tree Multiplier

An 8x8 Wallace Tree Multiplier is a circuit that performs multiplication of two 8-bit binary numbers using the Wallace tree algorithm. The Wallace tree multiplier is a high-speed and efficient binary multiplier that uses a combination of partial products and carry-save adders to reduce the number of full adders required for the multiplication operation.

[10]The function of an 8x8 Wallace Tree Multiplier is to provide a fast and efficient method for multiplying two 8-bit binary numbers. This type of multiplier is commonly used in digital signal processing applications, such as in digital filters, modems, and other signal processing systems where high-speed multiplication is required.

The 8x8 Wallace Tree Multiplier performs the multiplication operation using a series of partial products that are generated by breaking down the 8-bit binary numbers into smaller groups of bits. These partial products are then combined using carry-save adders to reduce the number of full adders required for the final addition stage.

The output of the 8x8 Wallace Tree Multiplier is a 16-bit binary number that represents the product of the two 8-bit binary numbers. The Wallace tree multiplier is known for its speed and efficiency and is commonly used in applications where high-speed multiplication is critical.

Overall, the function of an 8x8 Wallace Tree Multiplier is to provide a high-speed and efficient method for multiplying two 8-bit binary numbers in digital signal processing applications.

Circuit Schematics

8x8 Braun Multiplier

Pseudocode for 8x8 Braun Mul



Inputs:

- 8-bit unsigned integer A
- 8-bit unsigned integer B

Outputs:

- 16-bit unsigned integer C (the product of A and B)

Steps:

1. Initialize C to 0.
2. For i from 0 to 7:
 - a. If the ith bit of B is 1, add A shifted i bits to the left to C.
3. Return C.

Explanation:

The Braun multiplier algorithm is a simple algorithm that performs multiplication by repeated addition and bit shifting. It works by iterating over the bits of the second input, B, and adding A shifted left by the position of the current bit to the product, C, if the current bit is 1. Since the maximum value of each input is 255 (2^8-1), the maximum value of the product is 65,535 ($2^{16}-1$), which can be represented in 16 bits.

The pseudocode starts by initializing the product, C, to 0. It then iterates over the bits of B using a loop that runs from 0 to 7 (since B has 8 bits). For each bit, it checks if it is 1 using a bitwise AND operation. If the bit is 1, it adds A shifted left by the current bit position to C using a bitwise OR operation. After the loop completes, the algorithm returns the product, C.

8x8 Braun Multiplier Logic Diagram

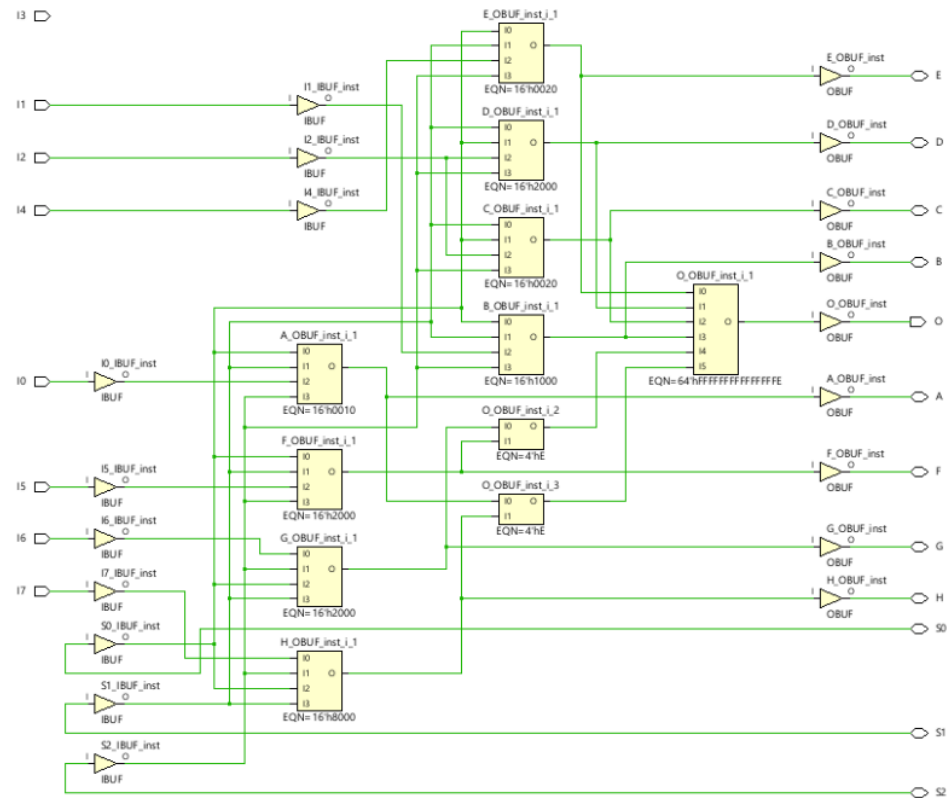


Figure 1- 8x8 Braun Multiplier Logic Diagram

8x8 Wallace Tree Multiplier

Pseudocode for 8x8 Wallace Tree Multiplier

Inputs:

8-bit unsigned integer A

8-bit unsigned integer B

Outputs:

16-bit unsigned integer C (the product of A and B)

Steps:

- Initialize an array of 8 elements, each initialized to 0.
- For i from 0 to 7:
- For j from 0 to 7:
- If the jth bit of B is 1, add A shifted i+j bits to the left to the ith element of the array.
- Perform a carry-save reduction on the array to obtain the product C.
- Return C.

Explanation:



The Wallace Tree multiplier algorithm is a high-speed algorithm that uses a combination of partial products and carry-save adders to reduce the number of full adders required for the multiplication operation. It works by generating partial products by shifting A left by different amounts and ANDing it with B, and then adding the partial products using carry-save adders.

The pseudocode starts by initializing an array of 8 elements, each representing a partial product, to 0. It then iterates over the bits of B using a loop that runs from 0 to 7 (since B has 8 bits). For each bit, it checks if it is 1 using a bitwise AND operation. If the bit is 1, it generates a partial product by shifting A left by the sum of the current bit position and the partial product index, and then adds the partial product to the corresponding element of the array using a bitwise OR operation.

After the loop completes, the algorithm performs a carry-save reduction on the array to obtain the final product, C. Carry-save reduction is a technique that uses multiple full adders to add the elements of the array together, resulting in a carry and two sum outputs. The sum outputs are then added together using another set of carry-save adders to obtain the final product, C.

This pseudocode can be implemented in any programming language that supports bitwise operations and unsigned integers, as well as carry-save adders and other required components. The performance of the algorithm can be further improved by using other optimization techniques, as discussed earlier.

8x8 Wallace Tree Multiplier Logic Diagram

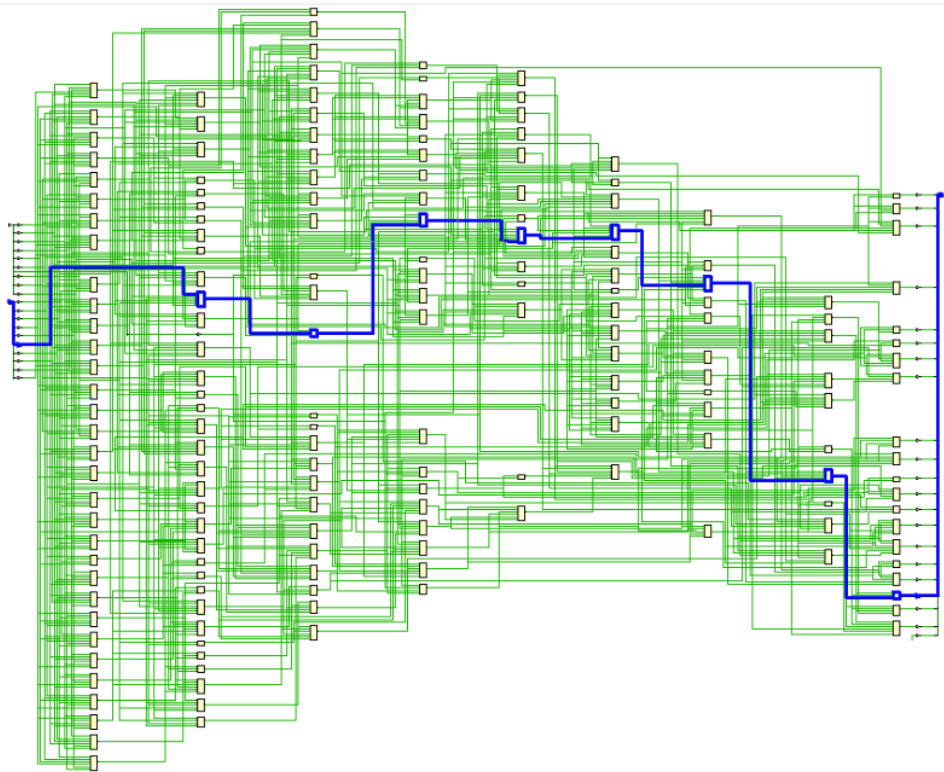


Figure 2- 8x8 Wallace Tree Multiplier Logic Diagram

Circuit Simulations

8x8 Braun Multiplier

Below shows the implementation of the 8x8 Braun Multiplier onto FPGA using the Vivado software.

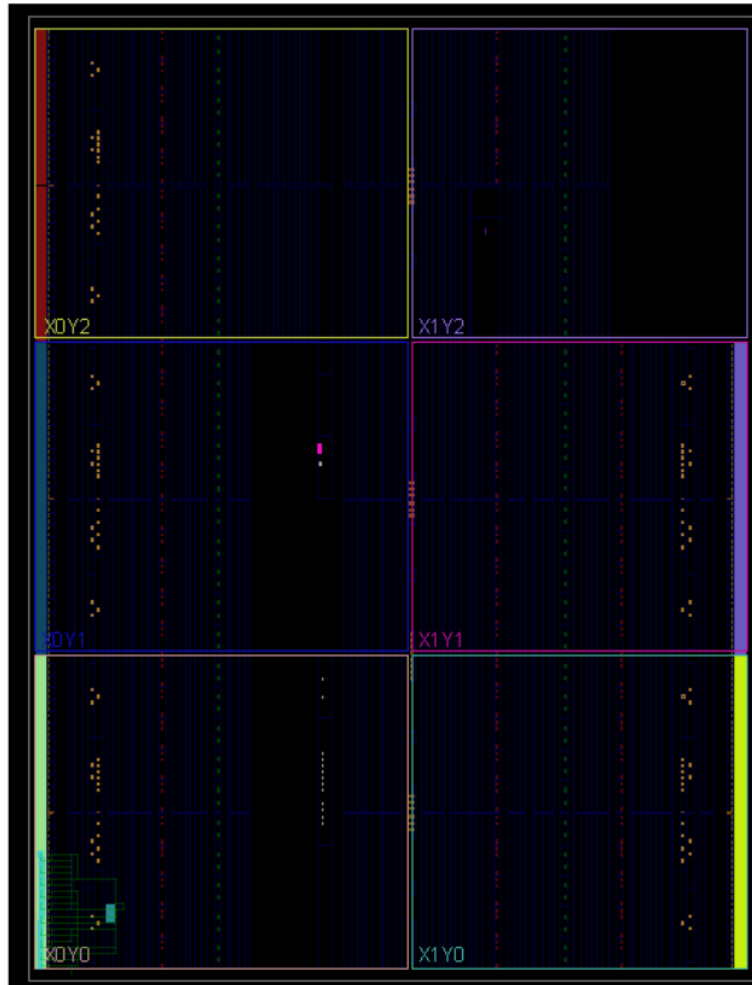


Figure 3- 8x8 Braun Multiplier FPGA Overview

The image below shows a section of the FPGA, showing the LUTs used within the FPGA.

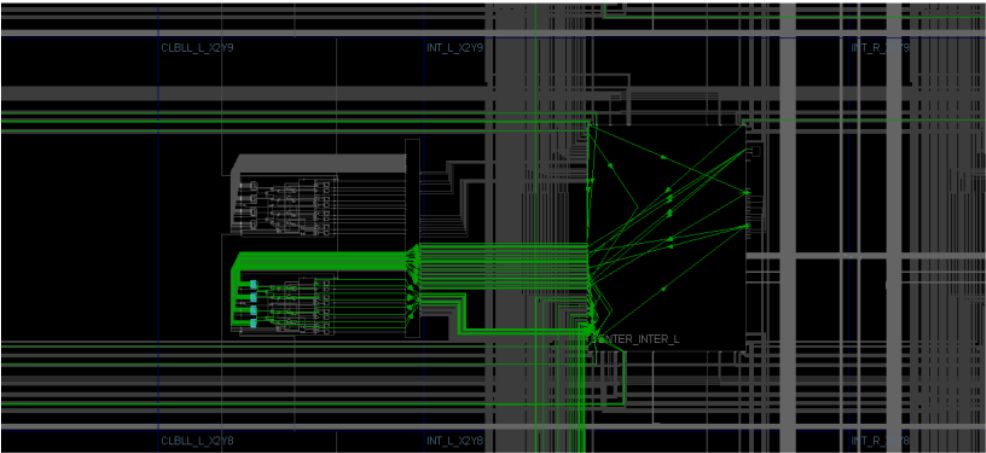


Figure 4-8x8 Braun Multiplier FPGA Implementation

Below shows the report summary from the Vivado software of the implemented Braun Multiplier onto FPGA. This report shows us the total power consumption of this multiplier, being 1.435W. It also shows the junction temperature of 29.1°C, indicating that this running efficiently and effectively.

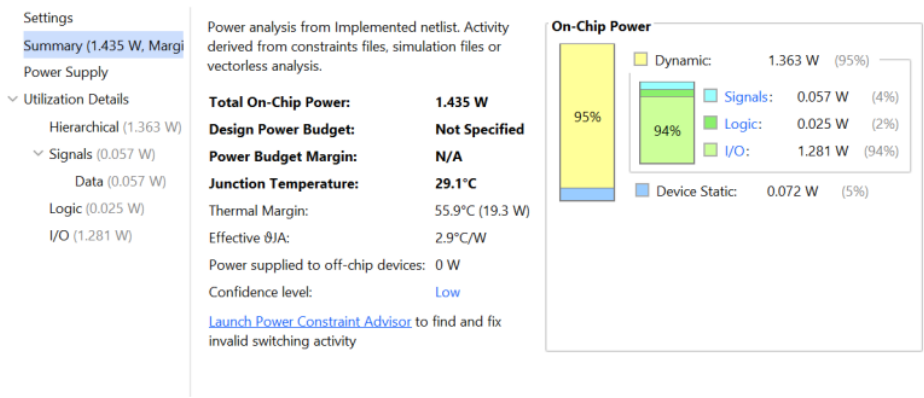


Figure 5- 8x8 Braun Multiplier Power Report

8x8 Wallace Tree Multiplier

Below shows the incorporation of an 8x8 Wallace tree multiplier onto a FPGA with the Vivado Software tool.

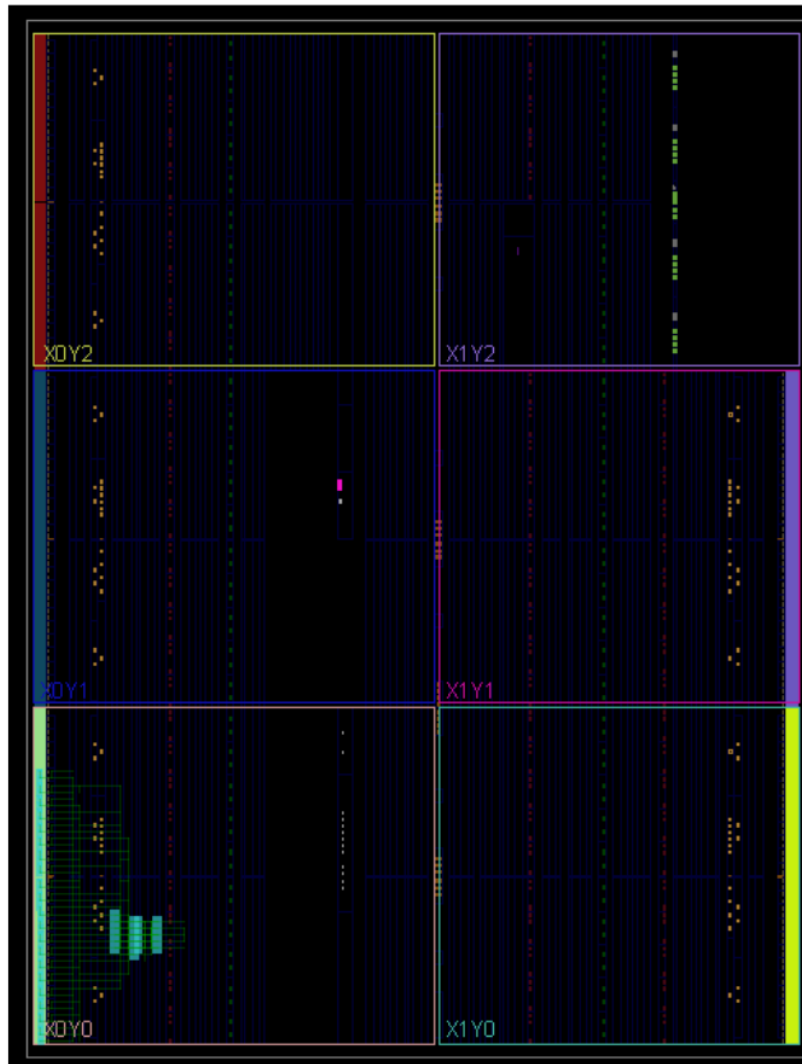


Figure 6- 8x8 Wallace Tree Multiplier FPGA Overview

Below is a section of the FPGA for the 8x8 Wallace tree multiplier. This shows the links between the various LUTs used on the FPGA.

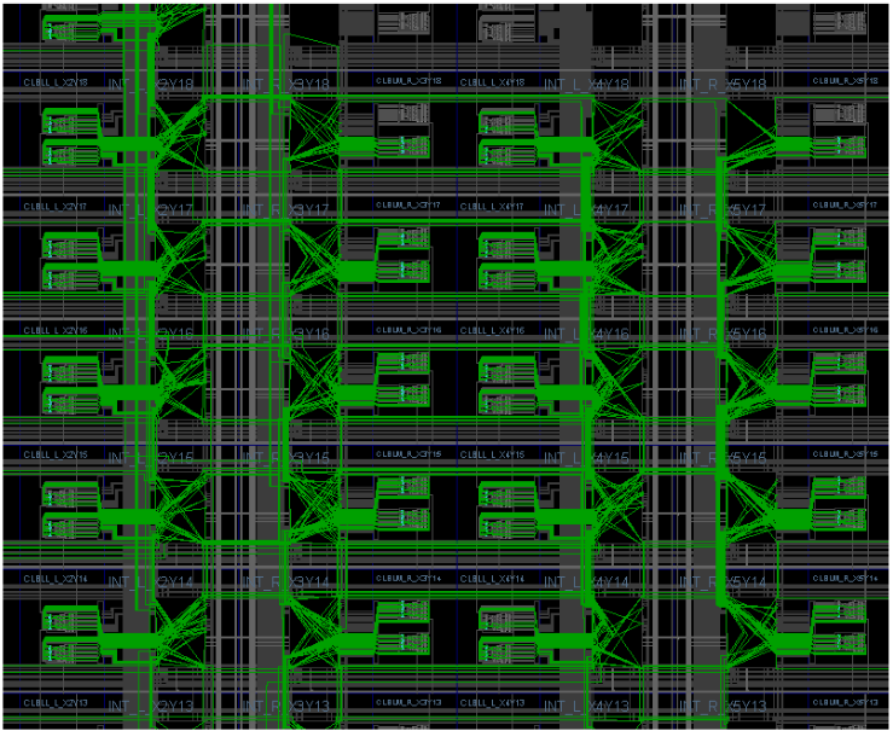
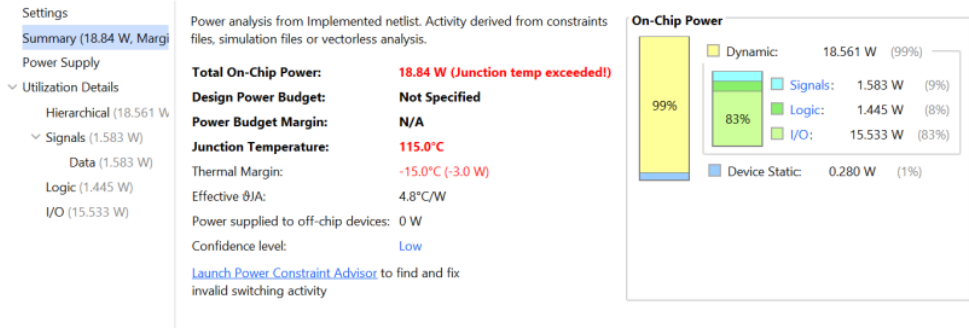


Figure 7- 8x8 Wallace Tree Multiplier FPGA Implementation

Below shows the report summary from the Vivado software after the code was implemented onto the FPGA.

From the report, it shows that the efficiency of this particular 8x8 Wallace Tree Multiplier is pretty poor, with the total power being 18.84W. It can also be seen that the junction temperature is unusually high. This is most likely down the design of the multiplier, with some specifics needing attention to improve the efficiency of this multiplier.



Timings Summary

8x8 Braun's Multiplier Timing Summary

Below shows the longest path of the implemented 8x8 Braun Multiplier compared to the shortest path on the FPGA. The longest path from the timing report is 8.714ns.

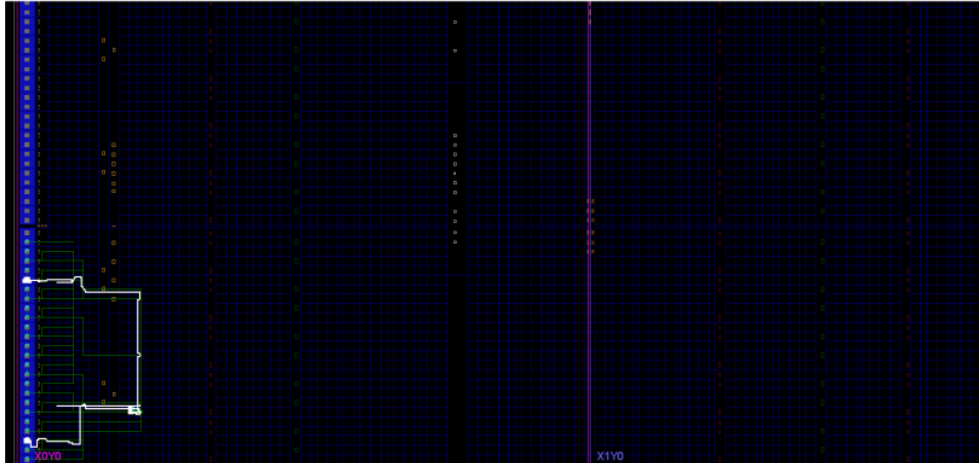


Figure 9- 8x8 Braun Multiplier FPGA Longest Path

Below shows the shortest path of the 8x8 Braun Multiplier implemented onto the FPGA. The shortest path is 2.942ns.

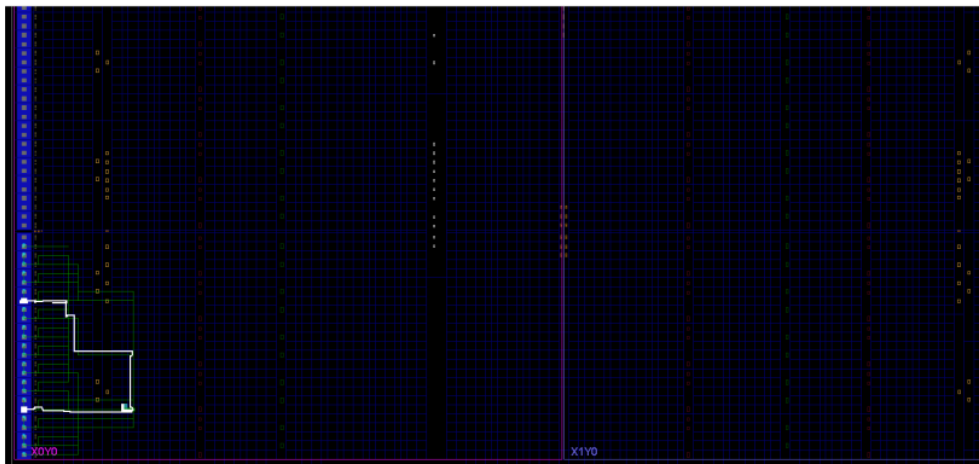


Figure 10- 8x8 Braun Multiplier FPGA Shortest Path

8x8 Wallace Tree Multiplier Timing Summary

11

Below shows the longest path on the 8x8 Wallace Tree Multiplier, compared to that of the shortest path. The longest path shows a delay of 17.54ns. This could be improved with adding some timing constraints to the FPGA.

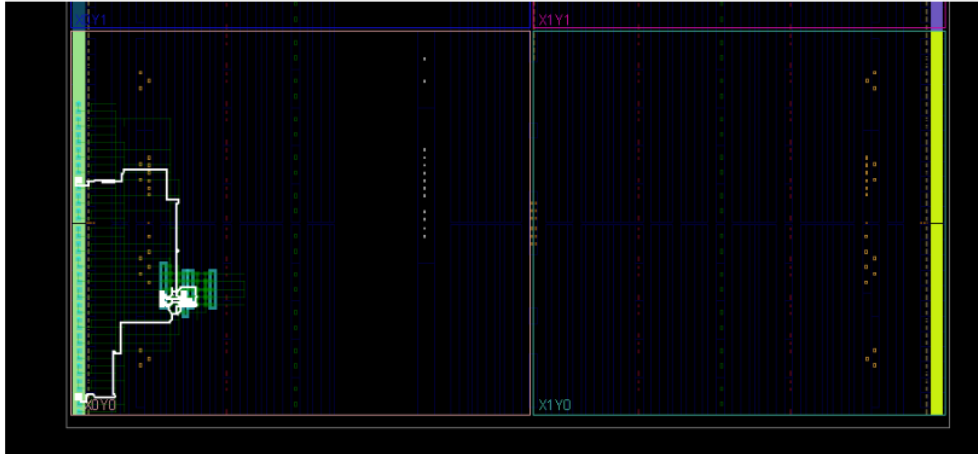


Figure 11- 8x8 Wallace Tree Multiplier FPGA Longest Path

The image below shows the shortest path of the Wallace Tree Multiplier. The shortest path shows a delay of 3.034ns.

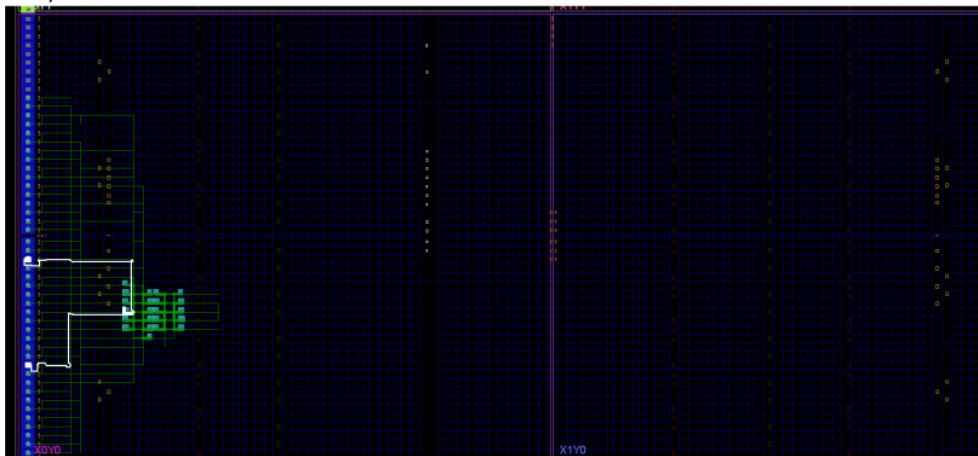


Figure 12- 8x8 Wallace Tree Multiplier FPGA Shortest Path

Digital Multiplier Improvements

There are several techniques that can be used to improve the speed and power efficiency of digital multipliers. Some of these techniques include:

- **Parallelism:** Digital multipliers can be designed to process multiple bits in parallel, which can significantly increase their speed. This is achieved by breaking the input numbers into smaller segments and processing them simultaneously.
- **Pipelining:** [8] Pipelining is a technique used to break down the multiplication operation into multiple stages that are executed in sequence. This allows for faster processing of multiple operations simultaneously, resulting in higher speed.
- **Algorithmic optimization:** Various optimization techniques can be used to minimize the number of operations required to perform multiplication. For example, the Wallace tree multiplier uses partial products and carry-save adders to reduce the number of full adders required, resulting in faster and more efficient multiplication.
- **Low-power circuit design:** Digital multipliers can be designed using low-power circuit design techniques, such as voltage scaling, clock gating, and power gating. These techniques help to reduce power consumption without sacrificing performance.
- **Technology scaling:** Advancements in semiconductor technology allow for the design of smaller, faster, and more power-efficient digital multipliers. Shrinking the size of transistors used in the design can improve performance and reduce power consumption.

Overall, the use of these techniques can help to improve the speed and power efficiency of digital multipliers, making them more suitable for a wide range of applications that require high-performance multiplication.

Conclusion

In conclusion, digital multipliers play a critical role in modern digital signal processing systems, enabling efficient multiplication of digital values in a wide range of applications. They are essential components that contribute to the performance, power consumption, and area efficiency of digital algorithms and electronic devices. Through various architectures and design techniques, digital multipliers are optimized to meet the specific requirements of different applications, such as telecommunications, image processing, audio processing, and sensor networks, among others.

The design of digital multipliers involves trade-offs between speed, power consumption, and area, depending on the application's constraints. Achieving high-speed multiplication while minimizing power consumption is a significant challenge that requires careful consideration of various design parameters, such as operand precision, word length, rounding and truncation methods, and implementation technology. The advancements in semiconductor technology, such as CMOS scaling, have greatly impacted the design and implementation of digital multipliers, enabling higher performance and increased efficiency.

Despite the progress made in digital multiplier design, there are still ongoing research and development efforts to further enhance their performance and functionality. Future advancements in semiconductor technology, circuit design techniques, and algorithm optimizations are expected to continue improving digital multipliers, enabling more efficient and advanced digital signal processing applications.

In conclusion, digital multipliers are vital components in modern digital signal processing systems, providing accurate and efficient multiplication of digital values. They are critical in enhancing the performance and functionality of various applications, including telecommunications, image processing, audio processing, and sensor networks. The optimization of digital multipliers in terms of speed, power consumption, and area is a significant challenge that requires careful consideration of design parameters and advances in semiconductor technology. Further research and development in digital multiplier design are expected to contribute to the advancement of digital signal processing systems and enable more efficient and advanced applications in the future.

References

- [1] "Stanford," [Online]. Available: <https://web.stanford.edu/~nanbhas/research/digital-multipliers/>. [Accessed 5 May 2023].
- [2] "Geeks for Geeks," [Online]. Available: <https://www.geeksforgeeks.org/array-multiplier-in-digital-logic/>. [Accessed 5 May 2023].
- [3] A. D. Booth, "Berkely," [Online]. Available: http://bwrcs.eecs.berkeley.edu/Classes/icdesign/ee241_s00/PAPERS/archive/booth51.pdf. [Accessed 8 May 2023].
- [4] C. Wallace, "Semantics," [Online]. Available: https://web.archive.org/web/20190304091907id_/http://pdfs.semanticscholar.org/bb41/0953db68b9bda88026ade697461154f8a942.pdf. [Accessed 8 May 2023].
- [5] "Springer," [Online]. Available: <https://link.springer.com/article/10.1007/s42452-020-2595-5>. [Accessed 8 May 2023].
- [6] W. J. Townsend, "IEEE Milestones," [Online]. Available: https://ieeemilestones.ethw.org/w/images/d/db/A_comparison_of_Dadda_and_Wallace_multiplier_delays.pdf. [Accessed 10 May 2023].
- [7] "Math World," [Online]. Available: <https://mathworld.wolfram.com/KaratsubaMultiplication.html>. [Accessed 10 May 2023].
- [8] J. G. A. S. Rahul Pal, "IEEE Xplore," [Online]. Available: <https://ieeexplore.ieee.org/document/8783651>. [Accessed 10 May 2023].
- [9] B. Neeraja and R. S. P. Goud, "IEEE Xplore," [Online]. Available: <https://ieeexplore.ieee.org/document/8869307>. [Accessed 6 May 2023].
- [10] M. D. Mounir Bohsali, "Web Archive," [Online]. Available: https://web.archive.org/web/20100215152142/http://www.veech.com/index_files/Wallace%20Tree.pdf. [Accessed 3 May 2023].

FINAL GRADE

64/100

GENERAL COMMENTS

Instructor

a. Introduction to Multipliers/ Various Multipliers/ Advantages and Disadvantages/ Applications of Multipliers (8) 7 Marks

Excellent-Introduction was provided with a clear idea on the different Multipliers. Significance of the Multipliers were captured. Major points are stated clearly and are well supported.

b. Background Research (8) 8 Marks

Exceptional engagement with an extensive range of relevant and credible literature found.

c. Diversity of sources and Referencing Skills(4) 4 Marks

High-level referencing skills consistently applied.

a. Task 1: Identifying the correct Multiplier (4), Architecture of Braun's Multiplier(6) 4 Marks

Good! The correct Multiplier is identified. Key concepts were discussed. The Architecture diagram is missing which is the reason to lose

marks.

**b. Task 2: Dot
Diagram(4), Architecture of Wallace
Multiplier(6) 2 Marks**

The dot diagram and the architecture were missing. Only the explanation of Wallace is found.

**a. Finding
Critical Path, Longest Delay, Min and Max Delay,
Power Consumption(12). 10
Marks**

Good Job- Critical evaluation on finding the Critical path, longest min, max delay, Power consumption were made. The range Wallace in terms of timing should be less when compared to Braun's. All the needed screen dumps were found which evidently proves the work of yours.

**b. Innovative
thinking in problem solving and feature
additions(8) 4 Marks**

Mostly valid arguments and logical judgements on Examining the evidences of the design.

a. Simulation (7) 4 Marks

Good Try- The simulation results were found with some unexcepted results. But, 4 minimal samples were made which is seen in the screen dumps.

**b. VHDL Coding's
(10) 5 Marks**

The codes are written for Braun's Multiplier has some errors but works and gives minimal results.

**b. Circuit
Schematic (4) 3 Marks**

The screen dump provided for the Circuit schematic is clear

**d. FPGA Synthesis
(4) 2 Marks**

The procedures to capture FPGA synthesis were done perfectly

**a. Logical order
of ideas, Flow of ideas in the report from the
beginning to end with
appropriate chapters, sub chapters and
continuation(7) 5 Marks**

Work is coherent, fluent, well-structured and organised.

**b. Definitions,
Acronyms and Abbreviations (3) 2
Marks**

Definitions, Acronyms and Abbreviations used appropriately

and adequately.

C.
Professionalism, Spelling and Grammar,
Complete Report, Proper
Presentation, Quality of the
report(5) 4
Marks

High level of professionalism is maintained throughout the report.

Total Marks 64% Marks

Findings and Simulations

Braun's Multiplier

Variable declarations are found. Simulation runs without and errors or warnings. Code operates only for LSB with some positional change have been observed. No assumptions made with respect to the adders.

Wallace Multiplier

Signal Declaration is made. Here assumptions are made with respect to the sum and carry.

The results are obtained at LSB level and MSBs are obtained with few errors.

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

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

Excellent-Introduction was provided with a clear idea on the different Multipliers. Significance of the Multipliers were captured. Major points are stated clearly and are well supported.

PAGE 5



Comment 2

Good



Comment 3

Well Said



Comment 4

Excellent

PAGE 6

PAGE 7



Comment 5

Exceptional engagement with an extensive range of relevant and credible literature found.

PAGE 8



Comment 6

The Architecture Diagram of both Multipliers are missing

PAGE 9

PAGE 10



Comment 7

The dot diagram and the architecture were missing. Only the explanation of Wallace is found.

PAGE 11



Comment 8

The screen dump provided for the Circuit schematic is clear

PAGE 12

PAGE 13



Comment 9

The procedures to capture FPGA synthesis were done perfectly

PAGE 14

PAGE 15



Comment 10

Critical evaluation on finding the Power consumption were made. The consumption would be reduced if few adders were considered

PAGE 16

PAGE 17



Comment 11

Critical evaluation on finding the Critical path, longest were made. The range Wallace in terms of timing should be less when compared to Braun's.

PAGE 18

PAGE 19

PAGE 20
