

Electronics 2 - ACDA6001

by Rebecca Coles

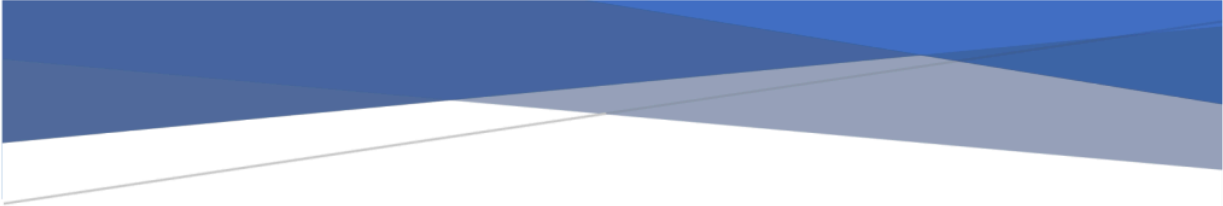
Submission date: 17-Apr-2025 01:19PM (UTC+0100)

Submission ID: 255954568

File name: 41397_Rebecca_Coles_Electronics_2_-_ACDA6001_654553_842132712.docx (6.16M)

Word count: 3628

Character count: 17655



CRITICAL EVALUATION OF BRAUN'S MULTIPLIER & WALLACE TREE MULTIPLIER USING VHDL

Electronics 2
ACDA6001

Rebecca Coles
2218001

Electrical & Electronic Engineering

Table of Contents

Table of Figures	3
Introduction.....	5
Background Research.....	5
Applications & Advantages of Multipliers.....	6
Flowchart	7
Braun's Multiplier	7
Wallace Tree Multiplier	7
Methodologies	8
VHDL Codes	9
8x8 Braun's VHDL Code.....	9
8x8 Wallace VHDL Code	15
Timing Diagrams & Schematics of Design	20
Braun's Multiplier – Timing diagrams	20
Timing 1.....	20
Timing 2.....	21
Timing 3.....	22
Timing 4.....	23
Timing Constraints – Longest delay	24
Timing Summary.....	24
Schematic	28
Wallace Tree Multiplier – Timing diagrams	29
Diagram 1.....	29
Diagram 2.....	29
Diagram 3.....	30
Diagram 4.....	31
Timing Constraints – Longest delay	32
Timing Summary.....	32
Schematic	36
FPGA Design & Pin Allocation	37
Braun's Multiplier Pin Allocation.....	37
Wallace Tree Multiplier Pin Allocation	38
Results.....	39
Braun's Multiplier	39
Battery Summary	39
RTL Analysis.....	40
LUT's.....	40
Wallace Tree Multiplier	41
Battery Summary	41

RTL Analysis..... 41

LUT's.....42

Recommendations..... 43

Conclusion 43

References..... 44

Table of Figures

Figure 1 – Braun’s Multiplier	7
Figure 2 - Wallace Tree Multiplier	7
Figure 3 - Dot Diagram.....	8
Figure 4 - Braun’s VHDL - 1	9
Figure 5 - Braun’s VHDL - 2	9
Figure 6 - Braun’s VHDL - 3	9
Figure 7 - Braun’s VHDL - 4	10
Figure 8 - Braun’s VHDL – 5.....	10
Figure 9 - Braun’s VHDL - 6	10
Figure 10 - Braun’s VHDL – 7.....	11
Figure 11 - Braun’s VHDL – 8.....	11
Figure 12 - Braun’s VHDL – 9.....	11
Figure 13 - Braun’s VHDL – 10.....	12
Figure 14 - Braun’s VHDL – 11.....	12
Figure 15 - Braun’s VHDL – 12.....	12
Figure 16 - Braun’s VHDL – 13.....	13
Figure 17 - Braun’s VHDL – 14.....	13
Figure 18 - Braun’s VHDL - 15	14
Figure 19 - Wallace VHDL - 1.....	15
Figure 20 - Wallace VHDL - 2.....	15
Figure 21 - Wallace VHDL - 3.....	16
Figure 22 - Wallace VHDL - 4.....	16
Figure 23 - Wallace VHDL – 5	17
Figure 24 - Wallace VHDL - 6.....	17
Figure 25 - Wallace VHDL - 7.....	18
Figure 26 - Wallace VHDL - 8.....	18
Figure 27 - Wallace VHDL - 9.....	19
Figure 28 - Braun’s simulation timing 1 (Max value)	20
Figure 29 - Braun’s simulation timing 2 (Min Value)	21
Figure 30 - Braun’s simulation timing 3 (Midd Value 1)	22
Figure 31 - Braun’s simulation timing 4 (Midd Value 2)	23
Figure 32 - Braun’s longest delay diagram.....	24
Figure 33 - Braun’s timing summary & device.....	24
Figure 34 - Braun’s time delays - 1.....	25
Figure 35 - Braun’s time delays - 2.....	25
Figure 36 - Braun’s time delays - 3.....	26
Figure 37 - Braun’s time delays - 4.....	26
Figure 38 - Braun’s time delays - 5.....	26
Figure 39 - Braun’s time delays - 6.....	27
Figure 40 - Braun’s time delays – 7.....	27
Figure 41 - Braun’s Schematic.....	28
Figure 42 - Wallace simulation timing 1 (Max Value)	29
Figure 43 - Wallace simulation timing 2 (Min Value)	29

Figure 44 - Wallace simulation timing 3 (Midd Value 1)	30
Figure 45 - Wallace simulation timing 4 (Midd Value 2)	31
Figure 46 - Wallace longest delay diagram	32
Figure 47 - Wallace timing summary & device	32
Figure 48 - Wallace timing delay - 1	33
Figure 49 - Wallace timing delay - 2	33
Figure 50 - Wallace timing delay - 3	34
Figure 51 - Wallace timing delay - 4	34
Figure 52 - Wallace timing delay - 5	35
Figure 53 - Wallace timing delay - 6	35
Figure 54 - Wallace timing delay - 7	36
Figure 55 - Wallace Schematic	36
Figure 56 - Braun's pin allocation	37
Figure 57 - Braun's device	37
Figure 58 - Wallace pin allocation	38
Figure 59 - Wallace device	38
Figure 60 - Braun's power summary	39
Figure 61 - Braun's set power value	39
Figure 62 - Braun's RTL analysis	40
Figure 63 - Braun's LUT's	40
Figure 64 - Wallace power summary	41
Figure 65 - Wallace RTL analysis	41
Figure 66 - Wallace LUT's	42

Introduction

This report looks at both the theory and practical use of Braun and Wallace multipliers using VHDL. It starts with background research on where and how multipliers are used, and explains the features, benefits, and working methods of Braun and Wallace multiplier designs. The report then moves on to show how each multiplier is built and tested in practice. Both types of multipliers are compared based on their speed, complexity, and power use, highlighting the strengths of each design. [1]

Background Research

Multipliers play an important part of digital systems, they are used in many areas like signal processing, image processing and computer systems where fast calculations are needed. The Braun's and Wallace Trees are two common types of multipliers, each with their own advantages and disadvantages. The Braun's multiplier is known for its simple and regular design, which makes it easier to build. However, the Wallace Tree multiplier is designed to be faster, this is due to the reduction of steps needed to complete the process.

Choosing the right type of multiplier depends on the needs of the application. Things to be considered would be speed, size and power. As signal processing tasks are becoming more demanding, the need for faster and more efficient multipliers has increased. In real-time systems, it's important to reduce any delays and power usage to meet performance requirements. By using the application VHDL to design and test these multipliers, we are able to compare how they perform and understand which one works better on different tasks. The choice of multiplier depends on the specific needs of the system and can be categorised as 'signed' or 'unsigned'. This is based on how fast they can handle the values inputted into the system.



The Braun's and Wallace Tree multipliers each have their own strengths and weaknesses. The Braun's multiplier is known for its simple and regular design, making it easy to understand and implement in any software. It is a good option to choose when the design needs to be straightforward and easy to implement. However, there are disadvantages to using the Braun's Multiplier, this includes slower processing and the use of more power due to longer delays in the system. On the other hand, the Wallace Tree multiplier is designed for speed, it does this by reducing the number of steps it takes to process the data, making it faster and more efficient for high-performance tasks. While it offers better speed and is more suitable for larger operations, its structure is more complex, which can make it harder to design and implement. In comparison, when choosing between the two multipliers, it would really depend on the needs of the application, whether simplicity or speed is more important. [1] [2]

VHDL is used as it allows us to describe, simulate and test systems before they are built on a software. VHDL makes it easier to design complex systems, detect errors throughout the simulation and make changes without physically rebuilding the circuit.

One key advantage of using VHDL is its public availability, as it is defined by the IEEE as an industry standard language. This standardisation ensures that VHDL is both machine readable and human readable, which helps improve the design, synthesis and testing of the software. VHDL follows a common structure and format, it also simplifies the process of sharing, maintaining and modifying any software designs. This is especially useful in long term projects, where updates or repairs may be needed, as it ensures that the data remains accessible and useable over time.

Applications & Advantages of Multipliers

Braun's multiplier is commonly used in digital systems where simplicity and regular designs are important. It is mostly used for applications such as image processing, signal processing and embedded systems that require basic, unsigned multiplication. One of the main advantages of Braun's multiplier is its simple structured layout, which makes it easy to design and implement on any software. Especially when used on FPGAs (Field Programmable Gate Array). Its structure allows for easy scaling when working with larger bit-widths. While it may not be the fastest, Braun's Multiplier is reliable, efficient for smaller systems and is a straight forward logic that makes testing and fault finding easier. This makes the Braun's multiplier a good choice for simple projects, low power designs and systems where speed isn't as critical.

FPGAs (Field Programmable Gate Arrays) are types of an integrated circuits that can be programmed to perform different tasks after they are manufactured. They are commonly used in custom software design, prototyping, high-performance computing and digital signal processing. FPGAs are flexible, allowing engineers to test and modify designs quickly, making them ideal for tasks that require high speed processing or real time data handling. [3]

The Wallace Tree multiplier is widely used in applications that require high speed operations. These include digital signal processing (DSP), microprocessors and image processing. The Wallace Multiplier is particularly useful for large bit multiplications where speed and efficiency are critical. One of the main advantages of the Wallace Tree multiplier is its ability to reduce the number of stages, speeding up the multiplication process. By using carry and sum adders in a tree-like structure, it minimises delays and improves performance. However, the increased complexity in both the design and software may demand more resources, which may affect speed. [4]

Flowchart

Braun’s Multiplier

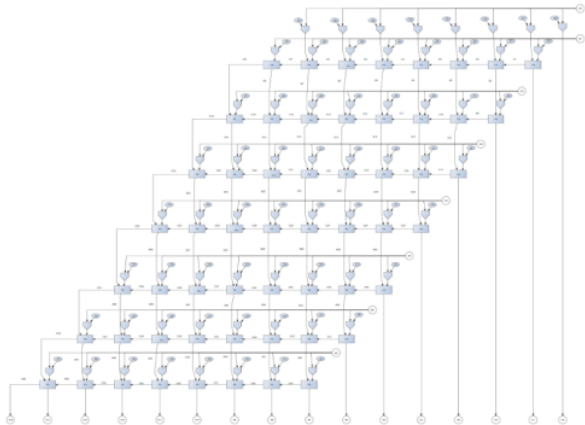


Figure 1 – Braun’s Multiplier

Wallace Tree Multiplier

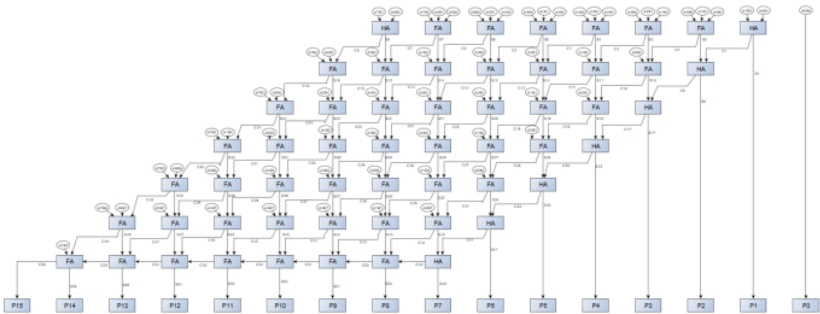


Figure 2 - Wallace Tree Multiplier

Methodologies

In this project we were required to design two 8x8 multipliers. One being the Braun's Multiplier and the other the Wallace Tree Multiplier. Both multiplier were designed, implemented and compared using VHDL and the Vivado software. The Braun's multiplier was built using AND gates and then adding them using either full or half adders, this was dependant on the inputs into each gate. Each component was written in VHDL and connected using port maps to build the complete multiplier, which was then simulated and tested on FPGA. For the Wallace Tree multiplier, the design focused on reducing multiple stages using a structured tree of adders to achieve faster performance. A dot diagram below is used to calculate half or full adders in the multiplier. As shown below, anything grouped as a three is a full adder, showing 3 inputs, where as anything grouped as a 2 is a half adder with only 2 inputs, refer to figure.3 below. It is also then followed by VHDL implementation, simulation and FPGA synthesis. . In the final step, both multipliers were compared by calculating delays, finding the critical path and checking power use. This helped to understand which design worked better in terms of speed and efficiency.

6

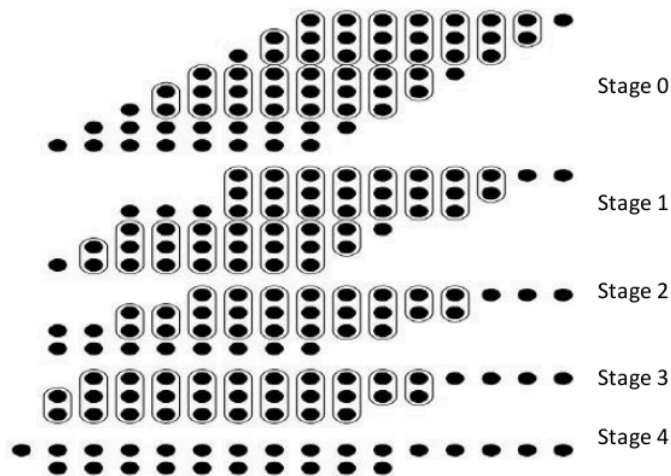


Figure 3 - Dot Diagram

7

VHDL Codes

8x8 Braun's VHDL Code

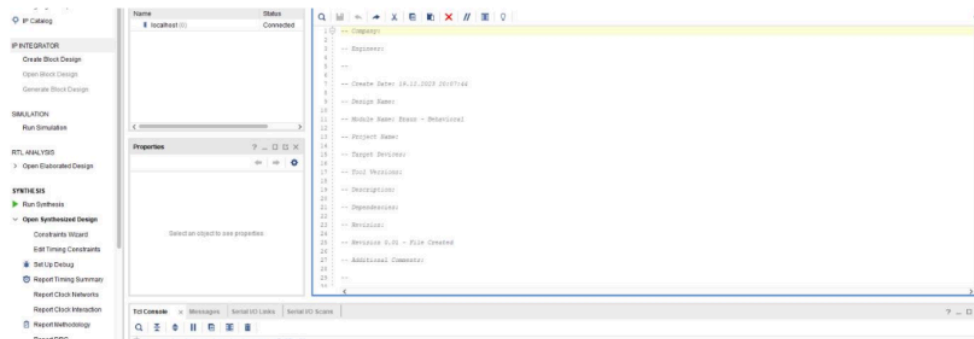


Figure 4 - Braun's VHDL - 1

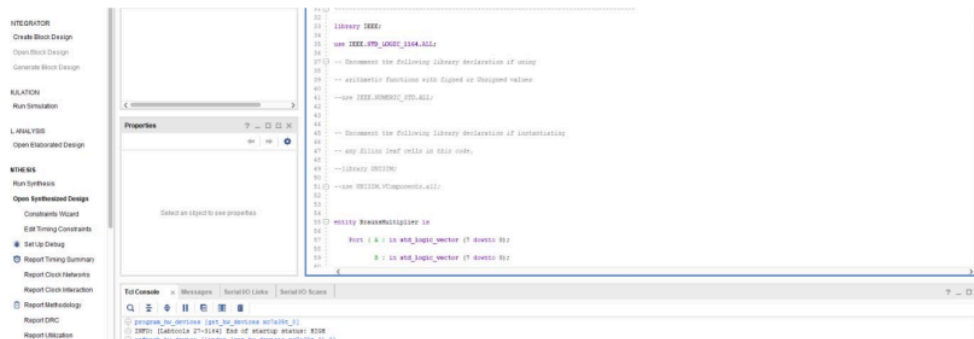


Figure 5 - Braun's VHDL - 2

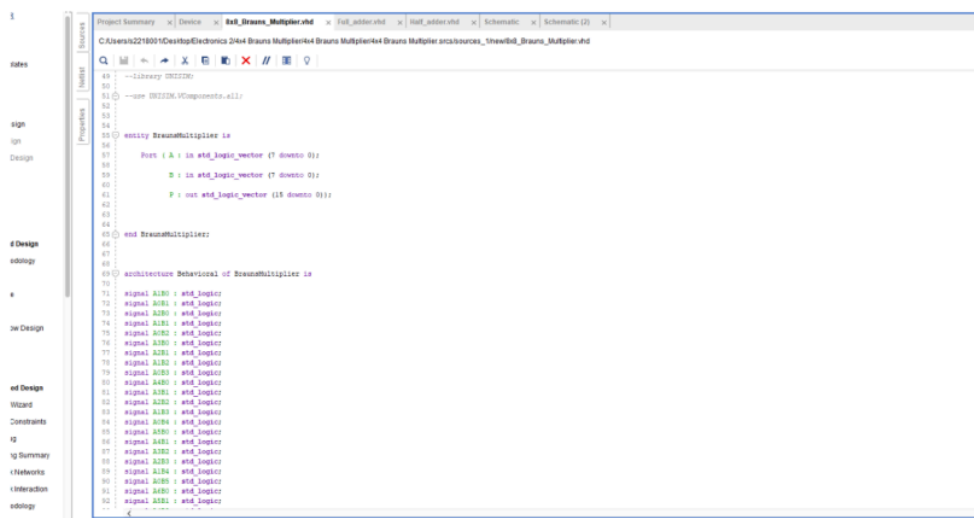


Figure 6 - Braun's VHDL - 3

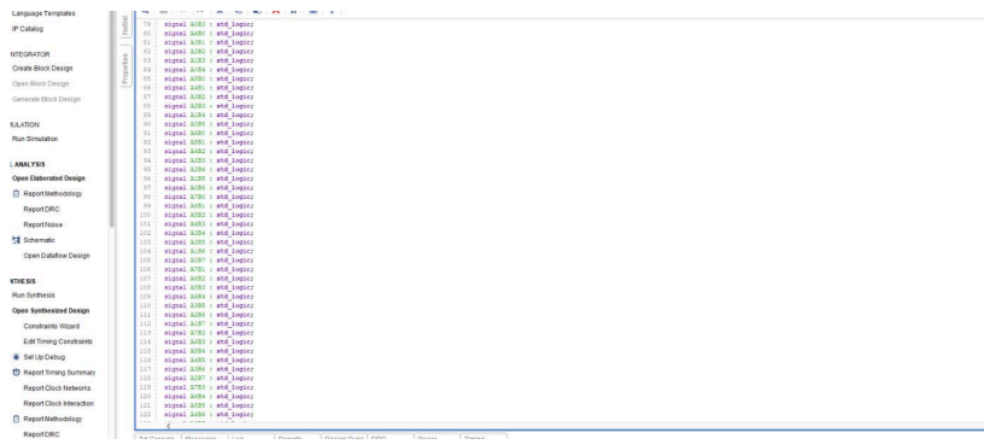


Figure 7 - Braun's VHDL - 4

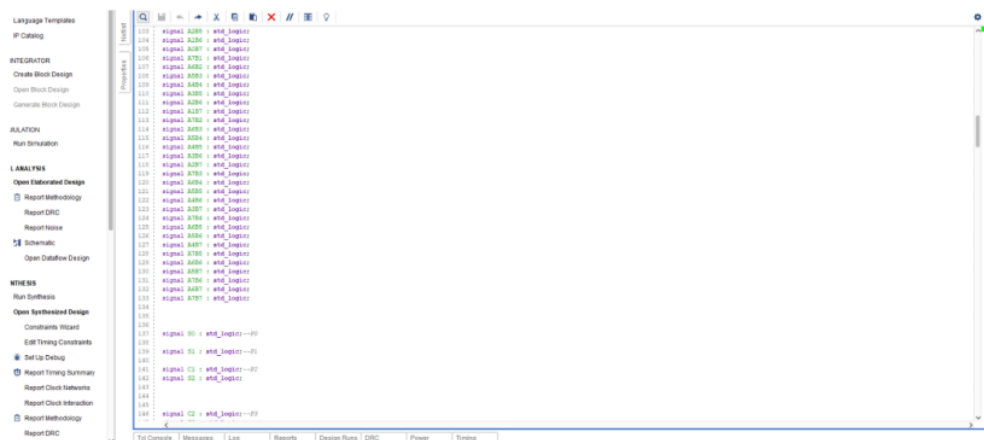


Figure 8 - Braun's VHDL – 5



Figure 9 - Braun's VHDL - 6

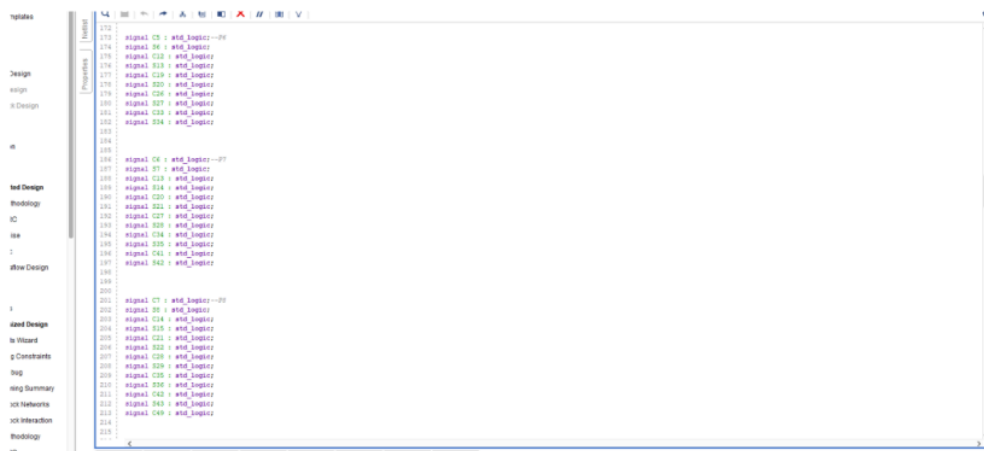


Figure 10 - Braun's VHDL – 7



Figure 11 - Braun's VHDL – 8

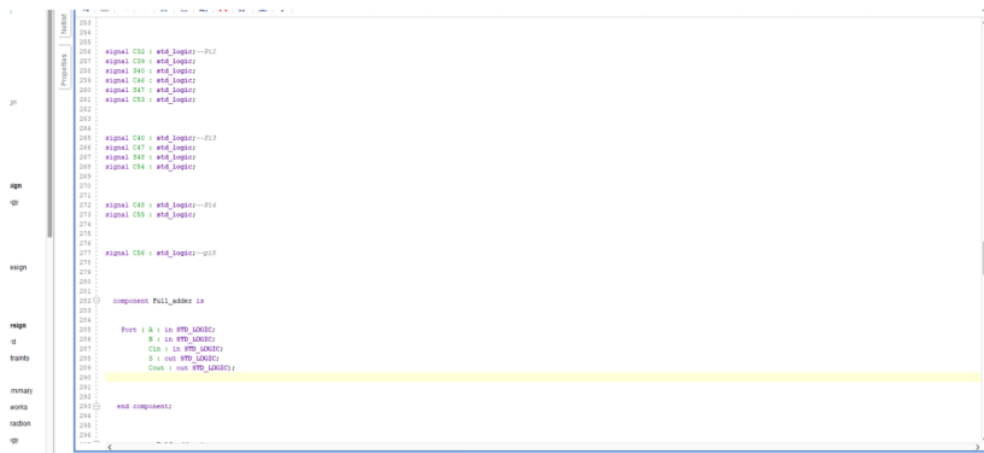


Figure 12 - Braun's VHDL – 9


```
C:\Users\12218001\Desktop\Electronics 2\44 Braun's Multiplier\44 Braun's Multiplier\44 Braun's Multiplier\srcs\newVhdl_Brauns_Multiplier.vhd

439 523_i : Full_adder port map (a=>A514, b=>C16, Cin=>C22, S=>S23, Cout=>C23);
440 530_i : Full_adder port map (a=>A524, b=>S23, Cin=>C29, S=>S30, Cout=>C30);
441 537_i : Full_adder port map (a=>A535, b=>S30, Cin=>C36, S=>S37, Cout=>C37);
442 544_i : Full_adder port map (a=>A536, b=>S37, Cin=>C43, S=>S44, Cout=>C44);
443 P9 : Full_adder port map (a=>A287, b=>S44, Cin=>C50, S=>P(9), Cout=>C51);
444
445
446
447 554_i : Full_adder port map (a=>A783, b=>C16, Cin=>C22, S=>S52, Cout=>C52);
448 555_i : Full_adder port map (a=>A484, b=>S24, Cin=>C20, S=>S55, Cout=>C55);
449 556_i : Full_adder port map (a=>A585, b=>S51, Cin=>C37, S=>S58, Cout=>C58);
450 545_i : Full_adder port map (a=>A486, b=>S58, Cin=>C44, S=>S45, Cout=>C45);
451 P10 : Full_adder port map (a=>A287, b=>S45, Cin=>C51, S=>P(10), Cout=>C52);
452
453
454
455 532_i : Full_adder port map (a=>A784, b=>C24, Cin=>C31, S=>S32, Cout=>C32);
456 539_i : Full_adder port map (a=>A485, b=>S32, Cin=>C38, S=>S39, Cout=>C39);
457 546_i : Full_adder port map (a=>A586, b=>S39, Cin=>C45, S=>S46, Cout=>C46);
458 P11 : Full_adder port map (a=>A287, b=>S46, Cin=>C52, S=>P(11), Cout=>C53);
459
460
461
462 540_i : Full_adder port map (a=>A785, b=>C22, Cin=>C39, S=>S40, Cout=>C40);
463 547_i : Full_adder port map (a=>A486, b=>S40, Cin=>C46, S=>S47, Cout=>C47);
464 P12 : Full_adder port map (a=>A287, b=>S47, Cin=>C53, S=>P(12), Cout=>C54);
465
466
467
468 548_i : Full_adder port map (a=>A786, b=>C40, Cin=>C47, S=>S48, Cout=>C48);
469 P13 : Full_adder port map (a=>A287, b=>S48, Cin=>C54, S=>P(13), Cout=>C55);
470
471
472
473 P14 : Full_adder port map (a=>A787, b=>C48, Cin=>C55, S=>P(14), Cout=>C56);
474
475
476
477 P15 : P(15) <= C56;
478
479
480
481 end Behavioral;
482
```

Figure 18 - Braun's VHDL - 15

8x8 Wallace VHDL Code

```

1  -- Company:
2  -- Engineer:
3  --
4  -- Create Date: 07.12.2023 10:10:47
5  -- Design Name:
6  -- Module Name: wallace8 - Behavioral
7  -- Project Name:
8  -- Target Device:
9  -- Tool Versions:
10 -- Description:
11 --
12 -- Dependencies:
13 --
14 -- Revision:
15 -- Revision 0.01 - File Created
16 -- Additional Comments:
17 --
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41 library IEEE;
42 use IEEE.STD_LOGIC_1164.ALL;
43
44

```

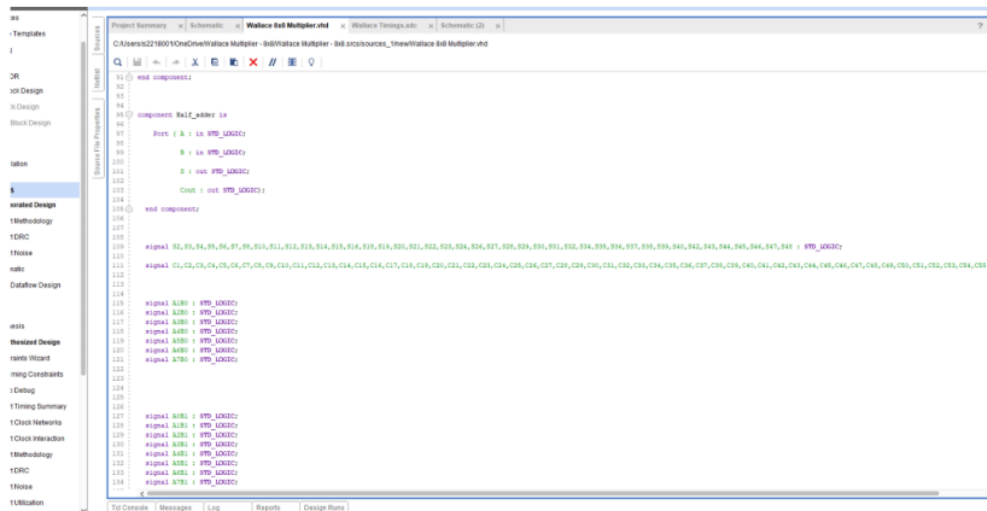
Figure 19 - Wallace VHDL - 1

```

45
46
47 -- Uncomment the following library declaration if using
48 -- arithmetic functions with Signed or Unsigned values
49 --use IEEE.NUMERIC_STD.ALL;
50
51
52
53
54
55 -- Uncomment the following library declaration if instantiating
56 -- any Xilinx leaf cells in this code.
57 --library UNISIM;
58 --use UNISIM.VComponents.all;
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

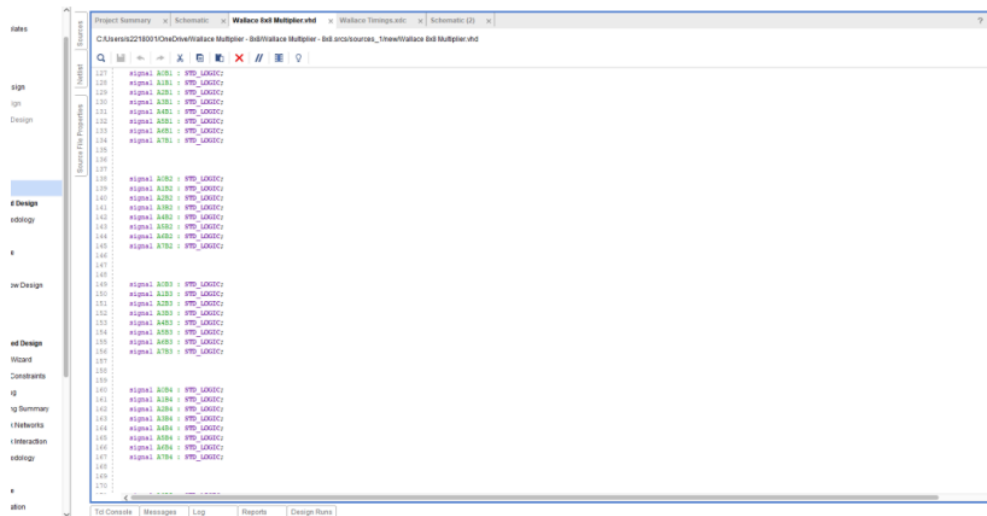
```

Figure 20 - Wallace VHDL - 2



```
100 library ieee;
101 use ieee.std_logic_1164.all;
102
103 component Wallace_Mult
104     port (
105         A : in std_logic_vector(255 downto 0);
106         B : in std_logic_vector(255 downto 0);
107         Y : out std_logic_vector(255 downto 0);
108     );
109 end component Wallace_Mult;
110
111 -- Main body of the file
112 signal A : std_logic_vector(255 downto 0);
113 signal B : std_logic_vector(255 downto 0);
114 signal Y : std_logic_vector(255 downto 0);
115
116 -- Instantiate the Wallace_Mult component
117 Wallace_Mult_inst : Wallace_Mult
118     port map (
119         A => A,
120         B => B,
121         Y => Y
122     );
123
124 -- End of file
```

Figure 21 - Wallace Vhdl - 3



```
125 library ieee;
126 use ieee.std_logic_1164.all;
127
128 component Wallace_Mult
129     port (
130         A : in std_logic_vector(255 downto 0);
131         B : in std_logic_vector(255 downto 0);
132         Y : out std_logic_vector(255 downto 0);
133     );
134 end component Wallace_Mult;
135
136 -- Main body of the file
137 signal A : std_logic_vector(255 downto 0);
138 signal B : std_logic_vector(255 downto 0);
139 signal Y : std_logic_vector(255 downto 0);
140
141 -- Instantiate the Wallace_Mult component
142 Wallace_Mult_inst : Wallace_Mult
143     port map (
144         A => A,
145         B => B,
146         Y => Y
147     );
148
149 -- End of file
```

Figure 22 - Wallace Vhdl - 4

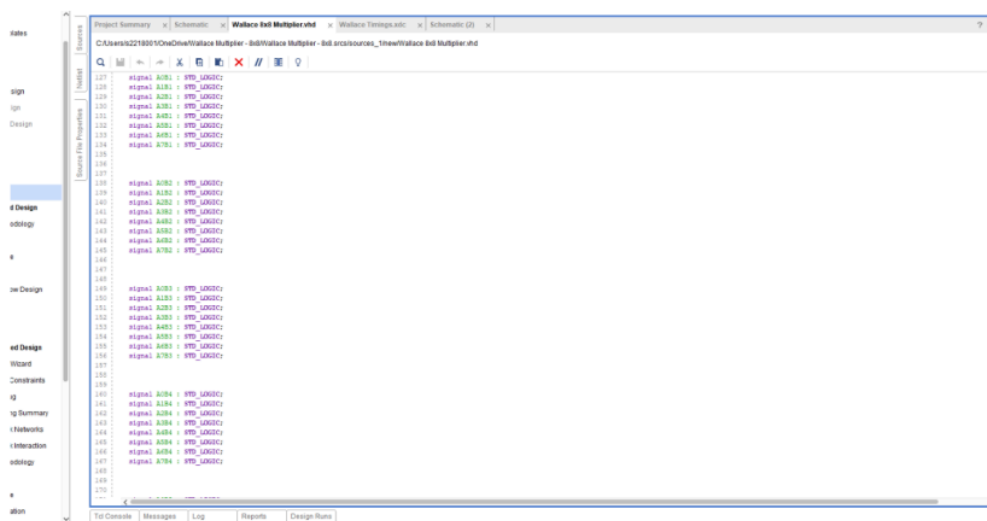


Figure 25 - Wallace VHDL - 7

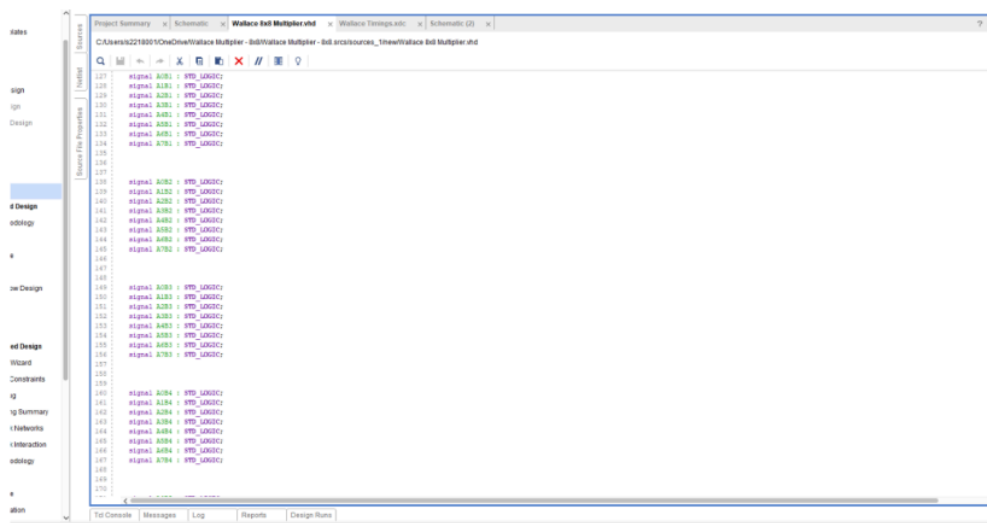


Figure 26 - Wallace VHDL - 8

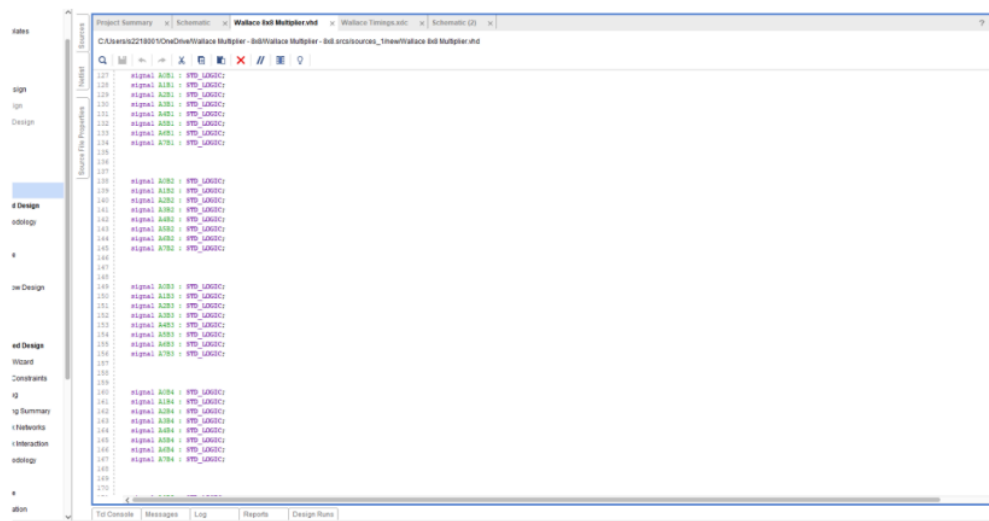


Figure 27 - Wallace VHDL - 9

Timing Diagrams & Schematics of Design
Braun’s Multiplier – Timing diagrams

The figures below show different timing constants applied to the Braun’s simulation. There has been 4 timing constraints applied. Each simulation has a forced clock on A & B with the leading edge value of 1 and the trailing edge value of 0. The constants for each simulation has then been set to max, min and two middle values.

Timing 1

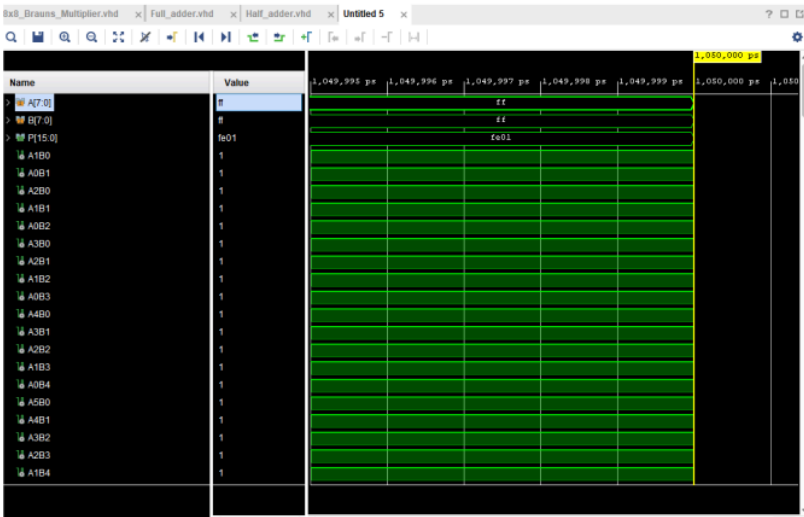


Figure 28 - Braun's simulation timing 1 (Max value)

Figure 28 shows the simulation being run and the different behaviours. This figure shows the maximum value that can be simulated on the multiplier.

A x B = P

Hex-
ff x ff = fe01

Binary-
1111 1111 x 1111 1111 = 1111 1110 0000 0001

Timing 2

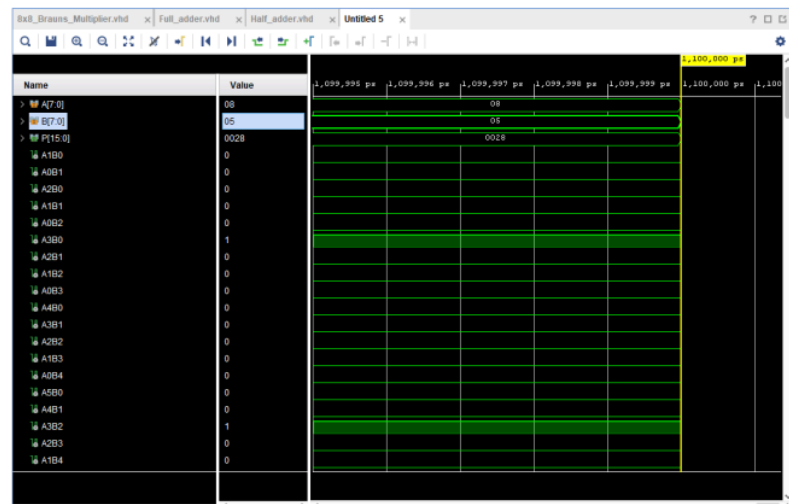


Figure 29 - Braun's simulation timing 2 (Min Value)

Figure 29 shows the minimum value input into this simulation.

A x B = P

Hex-

08 x 05 = 0028

Binary-

0000 1000 x 0000 0101 = 0000 0000 0010 1000

Timing 3

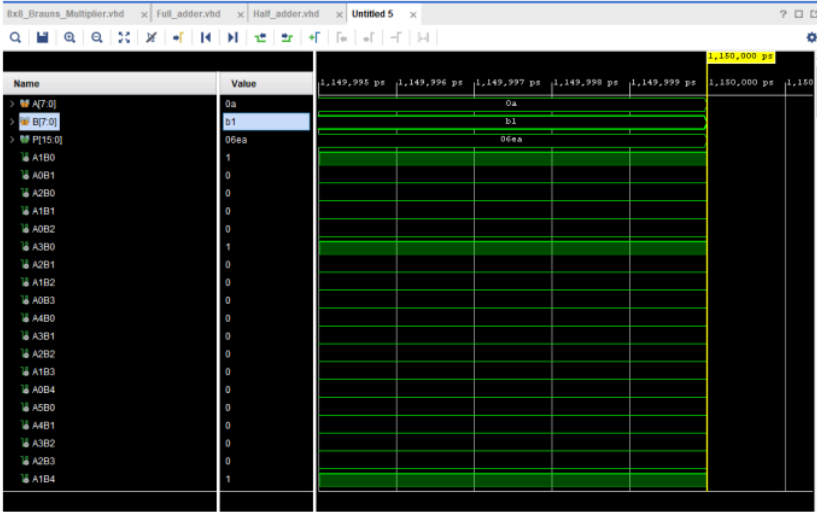


Figure 30 - Braun's simulation timing 3 (Midd Value 1)

Figure 30 shows the first middle value input into this simulation.

A x B = P

Hex-

0a x b1 = 06ea

Binary-

0000 1010 x 1011 0001 = 0000 0110 1110 1010

Timing 4

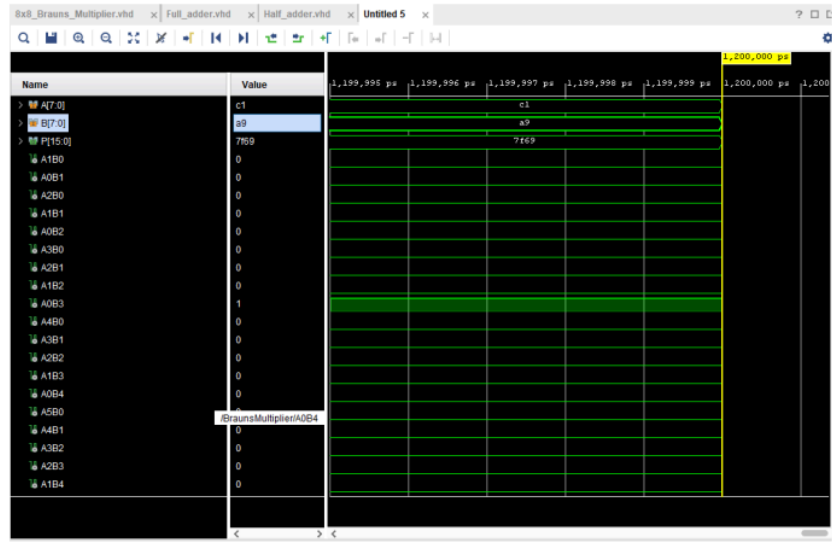


Figure 31 - Braun's simulation timing 4 (Middle Value)

Figure 31 shows the second middle value input into this simulation.

$$A \times B = P$$

Hex-
 $c1 \times a9 = 7f69$

Binary-
 $1100\ 0001 \times 1010\ 1001 = 0111\ 1111\ 0110\ 1001$

As demonstrated throughout each simulation each input has given the correct value output. This proves both coding and simulation are working correctly.

Timing Constraints – Longest delay

Given that AND gate delay = 1ns

Full adder/half adder delay = 2ns

$$\text{Longest delay} = 1\text{ns} + 14 * (2\text{ns}) = 29\text{ns}$$

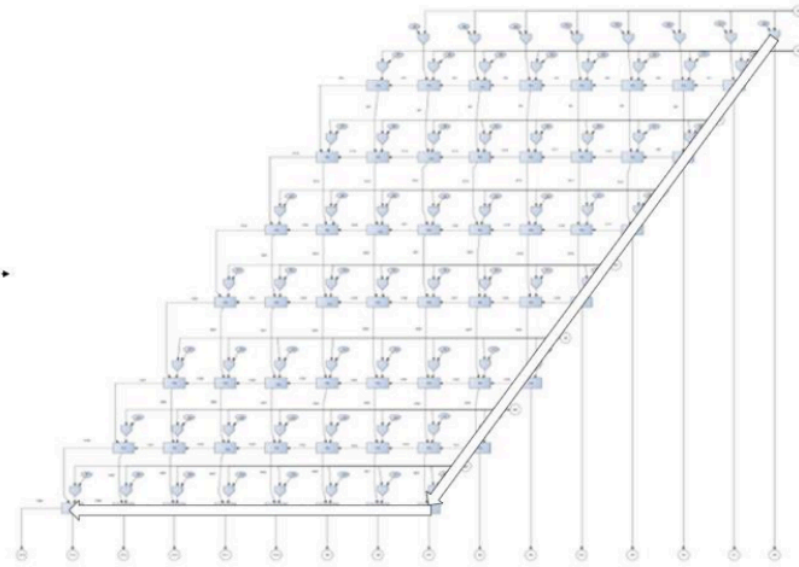


Figure 32 - Braun's longest delay diagram

10

Timing Summary

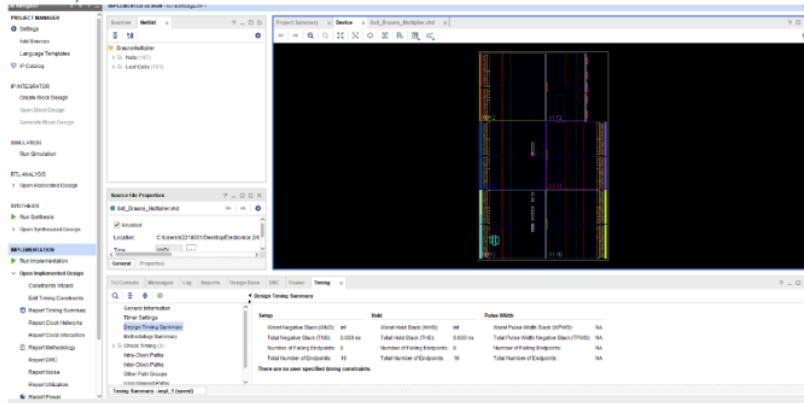


Figure 33 - Braun's timing summary & device

The figure shows the timing summary for the Braun's multiplier. It indicates that there are no timing violations, with zero negative slack and zero failing endpoints, meaning the design meets the

required timing constraints. This confirms that the multiplier operates correctly within the expected timing, making it reliable for FPGA implementation.

The next figures show the time delays of all signals in the Braun’s Multiplier.

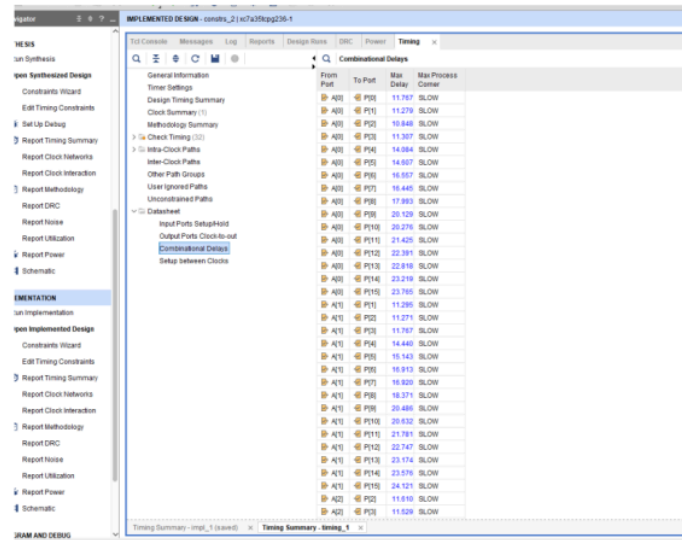


Figure 34 - Braun's time delays - 1

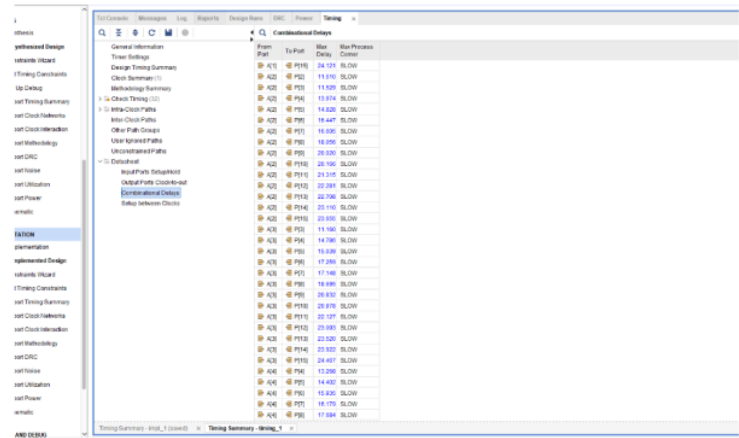


Figure 35 - Braun's time delays - 2

From	To	Max Delay	Max Process
P101	P102	20.484 SLONG	
P101	P103	21.430 SLONG	
P101	P104	21.807 SLONG	
P101	P105	22.258 SLONG	
P101	P106	22.694 SLONG	
P101	P107	23.110 SLONG	
P101	P108	23.510 SLONG	
P101	P109	23.894 SLONG	
P101	P110	24.264 SLONG	
P101	P111	24.624 SLONG	
P101	P112	24.974 SLONG	
P101	P113	25.314 SLONG	
P101	P114	25.644 SLONG	
P101	P115	25.964 SLONG	
P101	P116	26.274 SLONG	
P101	P117	26.574 SLONG	
P101	P118	26.864 SLONG	
P101	P119	27.144 SLONG	
P101	P120	27.414 SLONG	
P101	P121	27.674 SLONG	
P101	P122	27.924 SLONG	
P101	P123	28.164 SLONG	
P101	P124	28.394 SLONG	
P101	P125	28.614 SLONG	
P101	P126	28.824 SLONG	
P101	P127	29.024 SLONG	
P101	P128	29.214 SLONG	
P101	P129	29.394 SLONG	
P101	P130	29.564 SLONG	
P101	P131	29.724 SLONG	
P101	P132	29.874 SLONG	
P101	P133	30.014 SLONG	
P101	P134	30.144 SLONG	
P101	P135	30.264 SLONG	
P101	P136	30.374 SLONG	
P101	P137	30.474 SLONG	
P101	P138	30.564 SLONG	
P101	P139	30.644 SLONG	
P101	P140	30.714 SLONG	
P101	P141	30.774 SLONG	
P101	P142	30.824 SLONG	
P101	P143	30.864 SLONG	
P101	P144	30.894 SLONG	
P101	P145	30.914 SLONG	
P101	P146	30.924 SLONG	
P101	P147	30.934 SLONG	
P101	P148	30.944 SLONG	
P101	P149	30.954 SLONG	
P101	P150	30.964 SLONG	
P101	P151	30.974 SLONG	
P101	P152	30.984 SLONG	
P101	P153	30.994 SLONG	
P101	P154	31.004 SLONG	
P101	P155	31.014 SLONG	
P101	P156	31.024 SLONG	
P101	P157	31.034 SLONG	
P101	P158	31.044 SLONG	
P101	P159	31.054 SLONG	
P101	P160	31.064 SLONG	
P101	P161	31.074 SLONG	
P101	P162	31.084 SLONG	
P101	P163	31.094 SLONG	
P101	P164	31.104 SLONG	
P101	P165	31.114 SLONG	
P101	P166	31.124 SLONG	
P101	P167	31.134 SLONG	
P101	P168	31.144 SLONG	
P101	P169	31.154 SLONG	
P101	P170	31.164 SLONG	
P101	P171	31.174 SLONG	
P101	P172	31.184 SLONG	
P101	P173	31.194 SLONG	
P101	P174	31.204 SLONG	
P101	P175	31.214 SLONG	
P101	P176	31.224 SLONG	
P101	P177	31.234 SLONG	
P101	P178	31.244 SLONG	
P101	P179	31.254 SLONG	
P101	P180	31.264 SLONG	
P101	P181	31.274 SLONG	
P101	P182	31.284 SLONG	
P101	P183	31.294 SLONG	
P101	P184	31.304 SLONG	
P101	P185	31.314 SLONG	
P101	P186	31.324 SLONG	
P101	P187	31.334 SLONG	
P101	P188	31.344 SLONG	
P101	P189	31.354 SLONG	
P101	P190	31.364 SLONG	
P101	P191	31.374 SLONG	
P101	P192	31.384 SLONG	
P101	P193	31.394 SLONG	
P101	P194	31.404 SLONG	
P101	P195	31.414 SLONG	
P101	P196	31.424 SLONG	
P101	P197	31.434 SLONG	
P101	P198	31.444 SLONG	
P101	P199	31.454 SLONG	
P101	P200	31.464 SLONG	
P101	P201	31.474 SLONG	
P101	P202	31.484 SLONG	
P101	P203	31.494 SLONG	
P101	P204	31.504 SLONG	
P101	P205	31.514 SLONG	
P101	P206	31.524 SLONG	
P101	P207	31.534 SLONG	
P101	P208	31.544 SLONG	
P101	P209	31.554 SLONG	
P101	P210	31.564 SLONG	
P101	P211	31.574 SLONG	
P101	P212	31.584 SLONG	
P101	P213	31.594 SLONG	
P101	P214	31.604 SLONG	
P101	P215	31.614 SLONG	
P101	P216	31.624 SLONG	
P101	P217	31.634 SLONG	
P101	P218	31.644 SLONG	
P101	P219	31.654 SLONG	
P101	P220	31.664 SLONG	
P101	P221	31.674 SLONG	
P101	P222	31.684 SLONG	
P101	P223	31.694 SLONG	
P101	P224	31.704 SLONG	
P101	P225	31.714 SLONG	
P101	P226	31.724 SLONG	
P101	P227	31.734 SLONG	
P101	P228	31.744 SLONG	
P101	P229	31.754 SLONG	
P101	P230	31.764 SLONG	
P101	P231	31.774 SLONG	
P101	P232	31.784 SLONG	
P101	P233	31.794 SLONG	
P101	P234	31.804 SLONG	
P101	P235	31.814 SLONG	
P101	P236	31.824 SLONG	
P101	P237	31.834 SLONG	
P101	P238	31.844 SLONG	
P101	P239	31.854 SLONG	
P101	P240	31.864 SLONG	
P101	P241	31.874 SLONG	
P101	P242	31.884 SLONG	
P101	P243	31.894 SLONG	
P101	P244	31.904 SLONG	
P101	P245	31.914 SLONG	
P101	P246	31.924 SLONG	
P101	P247	31.934 SLONG	
P101	P248	31.944 SLONG	
P101	P249	31.954 SLONG	
P101	P250	31.964 SLONG	
P101	P251	31.974 SLONG	
P101	P252	31.984 SLONG	
P101	P253	31.994 SLONG	
P101	P254	32.004 SLONG	
P101	P255	32.014 SLONG	
P101	P256	32.024 SLONG	
P101	P257	32.034 SLONG	
P101	P258	32.044 SLONG	
P101	P259	32.054 SLONG	
P101	P260	32.064 SLONG	
P101	P261	32.074 SLONG	
P101	P262	32.084 SLONG	
P101	P263	32.094 SLONG	
P101	P264	32.104 SLONG	
P101	P265	32.114 SLONG	
P101	P266	32.124 SLONG	
P101	P267	32.134 SLONG	
P101	P268	32.144 SLONG	
P101	P269	32.154 SLONG	
P101	P270	32.164 SLONG	
P101	P271	32.174 SLONG	
P101	P272	32.184 SLONG	
P101	P273	32.194 SLONG	
P101	P274	32.204 SLONG	
P101	P275	32.214 SLONG	
P101	P276	32.224 SLONG	
P101	P277	32.234 SLONG	
P101	P278	32.244 SLONG	
P101	P279	32.254 SLONG	
P101	P280	32.264 SLONG	
P101	P281	32.274 SLONG	
P101	P282	32.284 SLONG	
P101	P283	32.294 SLONG	
P101	P284	32.304 SLONG	
P101	P285	32.314 SLONG	
P101	P286	32.324 SLONG	
P101	P287	32.334 SLONG	
P101	P288	32.344 SLONG	
P101	P289	32.354 SLONG	
P101	P290	32.364 SLONG	
P101	P291	32.374 SLONG	
P101	P292	32.384 SLONG	
P101	P293	32.394 SLONG	
P101	P294	32.404 SLONG	
P101	P295	32.414 SLONG	
P101	P296	32.424 SLONG	
P101	P297	32.434 SLONG	
P101	P298	32.444 SLONG	
P101	P299	32.454 SLONG	
P101	P300	32.464 SLONG	
P101	P301	32.474 SLONG	
P101	P302	32.484 SLONG	
P101	P303	32.494 SLONG	
P101	P304	32.504 SLONG	
P101	P305	32.514 SLONG	
P101	P306	32.524 SLONG	
P101	P307	32.534 SLONG	
P101	P308	32.544 SLONG	
P101	P309	32.554 SLONG	
P101	P310	32.564 SLONG	
P101	P311	32.574 SLONG	
P101	P312	32.584 SLONG	
P101	P313	32.594 SLONG	
P101	P314	32.604 SLONG	
P101	P315	32.614 SLONG	
P101	P316	32.624 SLONG	
P101	P317	32.634 SLONG	
P101	P318	32.644 SLONG	
P101	P319	32.654 SLONG	
P101	P320	32.664 SLONG	
P101	P321	32.674 SLONG	
P101	P322	32.684 SLONG	
P101	P323	32.694 SLONG	
P101	P324	32.704 SLONG	
P101	P325	32.714 SLONG	
P101	P326	32.724 SLONG	
P101	P327	32.734 SLONG	
P101	P328	32.744 SLONG	
P101	P329	32.754 SLONG	
P101	P330	32.764 SLONG	
P101	P331	32.774 SLONG	
P101	P332	32.784 SLONG	
P101	P333	32.794 SLONG	
P101	P334	32.804 SLONG	
P101	P335	32.814 SLONG	
P101	P336	32.824 SLONG	
P101	P337	32.834 SLONG	
P101	P338	32.844 SLONG	
P101	P339	32.854 SLONG	
P101	P340	32.864 SLONG	
P101	P341	32.874 SLONG	
P101	P342	32.884 SLONG	
P101	P343	32.894 SLONG	
P101	P344	32.904 SLONG	
P101	P345	32.914 SLONG	
P101	P346	32.924 SLONG	
P101	P347	32.934 SLONG	
P101	P348	32.944 SLONG	
P101	P349	32.954 SLONG	
P101	P350	32.964 SLONG	
P101	P351	32.974 SLONG	
P101	P352	32.984 SLONG	
P101	P353	32.994 SLONG	
P101	P354	33.004 SLONG	
P101	P355	33.014 SLONG	
P101	P356	33.024 SLONG	
P101	P357	33.034 SLONG	
P101	P358	33.044 SLONG	
P101	P359	33.054 SLONG	
P101	P360	33.064 SLONG	
P101	P361	33.074 SLONG	
P101	P362	33.084 SLONG	
P101	P363	33.094 SLONG	
P101	P364	33.104 SLONG	
P101	P365	33.114 SLONG	
P101	P366	33.124 SLONG	
P101	P367	33.134 SLONG	
P101	P368	33.144 SLONG	
P101	P369	33.154 SLONG	
P101	P370	33.164 SLONG	
P101	P371	33.174 SLONG	
P101	P372	33.184 SLONG	
P101	P373	33.194 SLONG	
P101	P374	33.204 SLONG	
P101	P375	33.214 SLONG	
P101	P376	33.224 SLONG	
P101	P377	33.234 SLONG	
P101	P378	33.244 SLONG	
P101	P379	33.254 SLONG	
P101	P380	33.264 SLONG	
P101	P381	33.274 SLONG	
P101	P382	33.284 SLONG	
P101	P383	33.294 SLONG	
P101	P384	33.304 SLONG	
P101	P385	33.314 SLONG	
P101	P386	33.324 SLONG	
P101	P387	33.334 SLONG	
P101	P388	33.344 SLONG	
P101	P389	33.354 SLONG	
P101	P390	33.364 SLONG	
P101	P391	33.374 SLONG	
P101	P392	33.384 SLONG	
P101	P393	33.394 SLONG	
P101	P394	33.404 SLONG	
P101	P395	33.414 SLONG	
P101	P396	33.424 SLONG	
P101	P397	33.434 SLONG	
P101	P398	33.444 SLONG	
P101	P399	33.454 SLONG	
P101	P400	33.464 SLONG	
P101	P401	33.474 SLONG	
P101	P402	33.484 SLONG	
P101	P403	33.494 SLONG	
P101	P404	33.504 SLONG	
P101	P405	33.514 SLONG	
P101	P406	33.524 SLONG	
P101	P407	33.534 SLONG	
P101	P408	33.544 SLONG	
P101	P409	33.554 SLONG	
P101	P410	33.564 SLONG	
P101	P411	33.574 SLONG	
P101	P412	33.584 SLONG	
P101	P413	33.594 SLONG	

File Edit View Layout Tools Design Tools DRC Power Timing				Combinational Delays			
				From Port	To Port	Max Delay	Max Process
gpi	General Information						
	Timer Settings						
gbs	Design Timing Summary			IN10	IN146	20.507 SL,0.00	
	Clock Summary (1)			IN10	IN105	21.002 SL,0.00	
may	Methodology Summary			IN14	IN146	8.895 SL,0.00	
	↳ Clock Timing (2)			IN14	IN105	11.658 SL,0.00	
fclk	↳ Inter-Clock Paths			IN14	IN105	13.612 SL,0.00	
	Inter-Clock Paths			IN14	IN107	13.893 SL,0.00	
clock	Other Path Groups			IN14	IN105	15.344 SL,0.00	
	User Ignored Paths			IN14	IN106	16.025 SL,0.00	
clock	Unconstrained Paths			IN14	IN102	16.663 SL,0.00	
	↳ Datasheet			IN14	IN111	17.812 SL,0.00	
	Input Ports: Subtotal			IN14	IN102	18.718 SL,0.00	
	Output Ports: Clock-to-out			IN14	IN110	19.226 SL,0.00	
	Combinational Delays			IN14	IN114	19.697 SL,0.00	
	Setup between Clocks			IN14	IN110	20.152 SL,0.00	
				IN10	IN105	9.914 SL,0.00	
				IN10	IN106	12.881 SL,0.00	
				IN10	IN107	13.131 SL,0.00	
				IN10	IN102	14.562 SL,0.00	
gpi				IN10	IN106	16.763 SL,0.00	
				IN10	IN102	16.981 SL,0.00	
may				IN10	IN111	17.089 SL,0.00	
				IN10	IN102	18.016 SL,0.00	
fclk				IN10	IN110	18.443 SL,0.00	
				IN10	IN114	18.645 SL,0.00	
clock				IN10	IN110	19.390 SL,0.00	
				IN10	IN105	9.777 SL,0.00	
				IN10	IN107	13.387 SL,0.00	
				IN10	IN106	15.148 SL,0.00	
				IN10	IN105	16.740 SL,0.00	
				IN10	IN110	16.689 SL,0.00	
				IN10	IN111	17.648 SL,0.00	
				IN10	IN112	18.614 SL,0.00	
				IN10	IN113	18.041 SL,0.00	

Figure 39 - Braun's time delays - 6

	From Path	To Path	Max Delay	Max Process Center
General Information				
Timer Settings				
Design Timing Summary	IP-REG1	IP-P113	19,205	SLOW
Clock Summary (1)	IP-REG1	IP-P114	19,687	SLOW
Methodology Summary	IP-REG1	IP-P115	20,152	SLOW
↳ Clock Timing (2)	IP-REG1	IP-P161	9,034	SLOW
↳ Intra-Clock Paths	IP-REG1	IP-P161	12,851	SLOW
Inter-Clock Paths	IP-REG1	IP-P171	13,131	SLOW
Other Path Groups	IP-REG1	IP-P161	14,582	SLOW
Never Ignored Paths	IP-REG1	IP-P161	15,763	SLOW
Unconstrained Paths	IP-REG1	IP-P116	16,891	SLOW
↳ Datasheet	IP-REG1	IP-P116	17,050	SLOW
Input Ports Setup/hold	IP-REG1	IP-P112	18,016	SLOW
Output Ports Setup/hold	IP-REG1	IP-P113	18,443	SLOW
Combinational Delay	IP-REG1	IP-P114	18,845	SLOW
Setup between Clocks	IP-REG1	IP-P115	19,390	SLOW
	IP-REG1	IP-P161	9,377	SLOW
	IP-REG1	IP-P171	13,367	SLOW
	IP-REG1	IP-P161	15,148	SLOW
	IP-REG1	IP-P161	16,740	SLOW
	IP-REG1	IP-P115	16,489	SLOW
	IP-REG1	IP-P116	17,648	SLOW
	IP-REG1	IP-P112	18,514	SLOW
	IP-REG1	IP-P113	19,041	SLOW
	IP-REG1	IP-P114	19,443	SLOW
	IP-REG1	IP-P115	19,888	SLOW
	IP-P171	IP-P171	9,719	SLOW
	IP-REG1	IP-P161	11,403	SLOW
	IP-REG1	IP-P161	12,584	SLOW
	IP-REG1	IP-P110	12,700	SLOW
	IP-REG1	IP-P111	13,773	SLOW
	IP-P171	IP-P112	14,639	SLOW
	IP-REG1	IP-P113	14,882	SLOW
	IP-REG1	IP-P114	14,888	SLOW
	IP-REG1	IP-P115	15,482	SLOW

Figure 40 - Braun's time delays – 7

Schematic

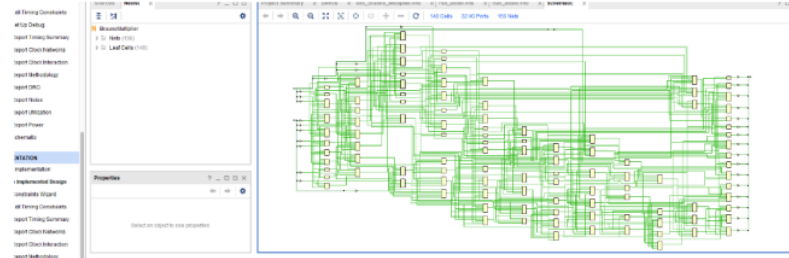


Figure 41 - Braun's Schematic

Figure 40 shows the Schematic layout for the Braun's multiplier designed in Vivado. The green lines represent the interconnections between components in the circuit, which also include components like AND gates and adders. This diagram help understand, test and improve the circuit.

Wallace Tree Multiplier – Timing diagrams

The same timing constant were applied on the Wallace Multiplier. See below figures of all 4 simulations.

Diagram 1

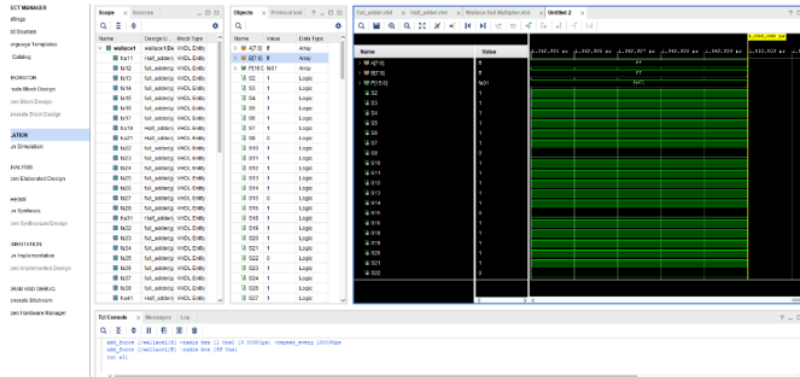


Figure 42 - Wallace simulation timing 1 (Max Value)

$$A \times B = P$$

Hex-
ff x ff = fe01

Binary-
1111 1111 x 1111 1111 = 1111 1110 0000 0001

Diagram 2

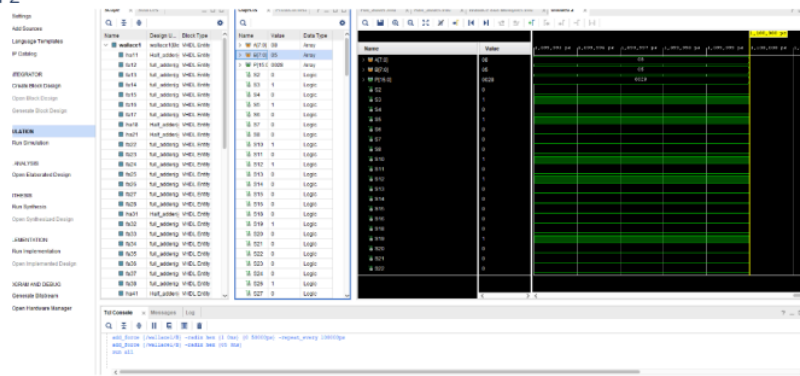


Figure 43 - Wallace simulation timing 2 (Min Value)

A x B = P

Hex-
08 x 05 = 0028

Binary-
0000 1000 x 0000 0101 = 0000 0000 0010 1000

Diagram 3

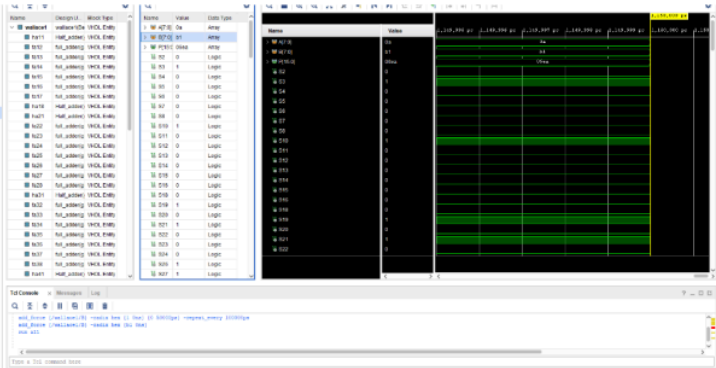


Figure 44 - Wallace simulation timing 3 (Midd Value 1)

A x B = P

Hex-
0a x b1 = 06ea

Binary-
0000 1010 x 1011 0001 = 0000 0110 1110 1010

Diagram 4

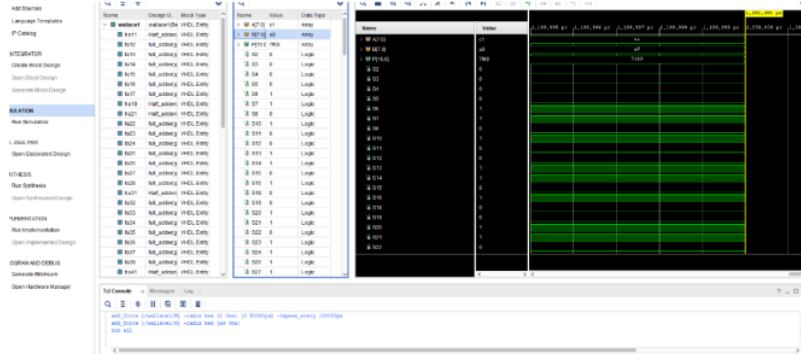


Figure 45 - Wallace simulation timing 4 (Mid Value 2)

$$A \times B = P$$

Hex-
 $c1 \times a9 = 7f69$

Binary-
 $1100\ 0001 \times 1010\ 1001 = 0111\ 1111\ 0110\ 1001$

As shown in all 4 simulations, each input has given the correct value output again using the Wallace Tree multiplier. This proves both coding and simulation are working correctly.

Timing Constraints – Longest delay

Given that AND gate delay = 1ns

Full adder/half adder delay = 2ns

$$\text{Longest delay} = 1\text{ns} + 20 * (2\text{ns}) = 41\text{ns}$$

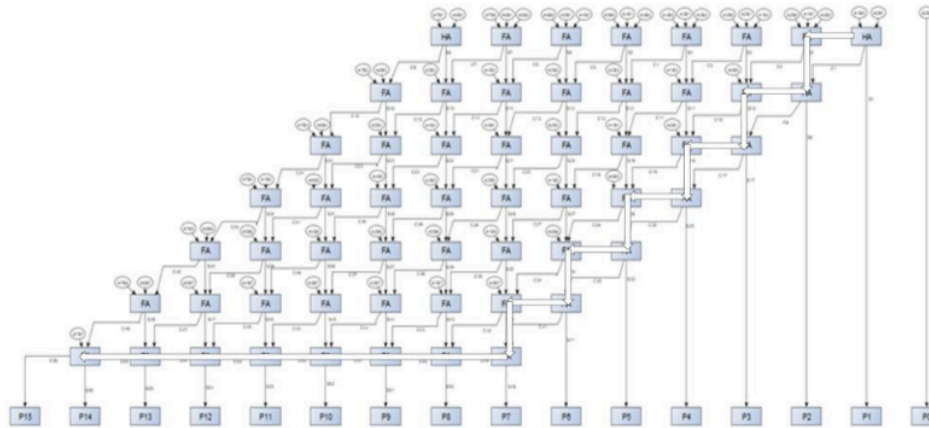


Figure 46 - Wallace longest delay diagram

Timing Summary

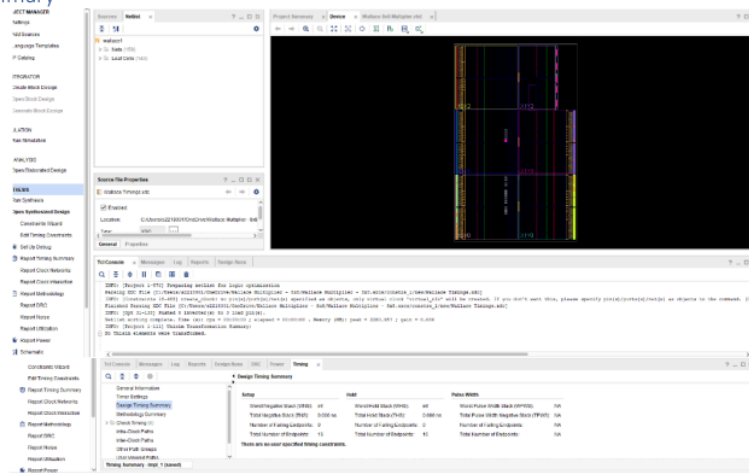


Figure 47 - Wallace timing summary & device

The timing summary for the Wallace Tree multiplier indicates that there are no timing violations, with zero negative slack and zero failing endpoints, meaning the design meets the required timing constraints. This again confirms that the multiplier operates correctly within the expected timing, making it reliable for FPGA implementation. As it stands, using the same inputs with either multiplier's, we are receiving the same output.

The next figures show the time delays of all signals in the Wallace Tree Multiplier.

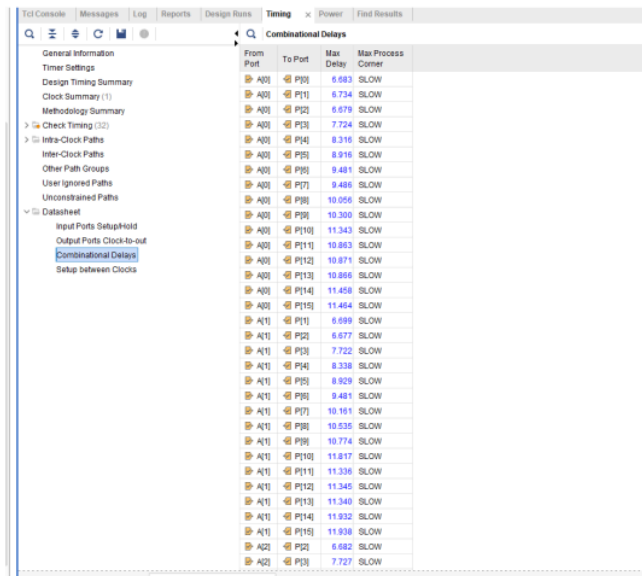


Figure 48 - Wallace timing delay - 1

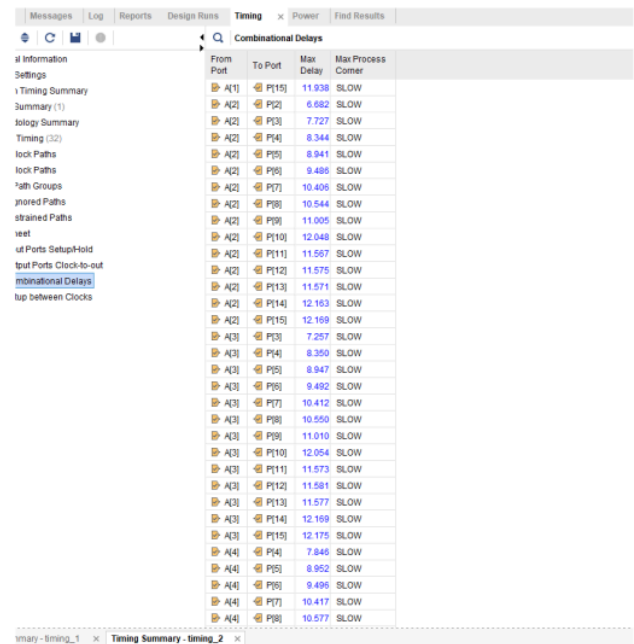


Figure 49 - Wallace timing delay - 2

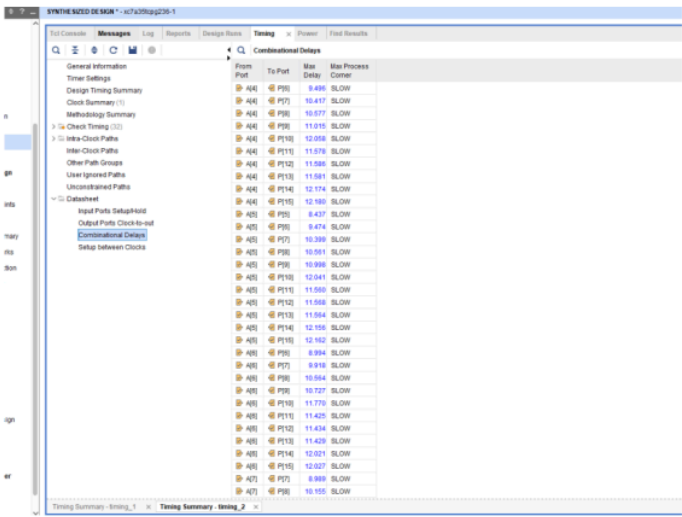


Figure 50 - Wallace timing delay - 3

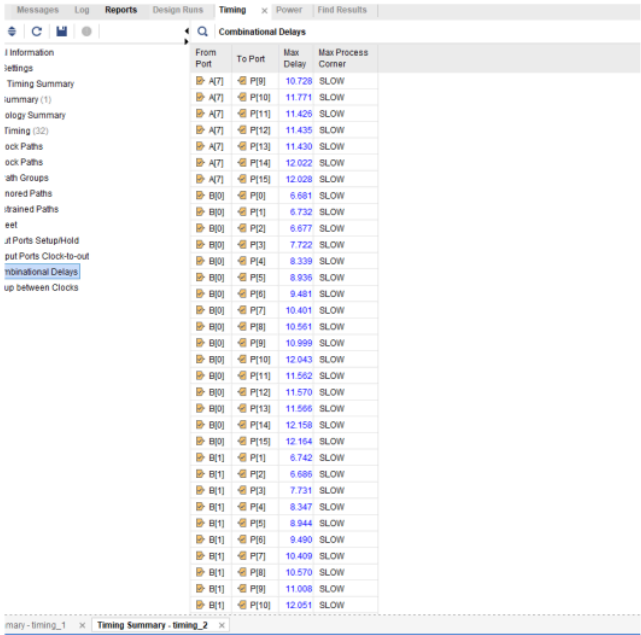


Figure 51 - Wallace timing delay - 4

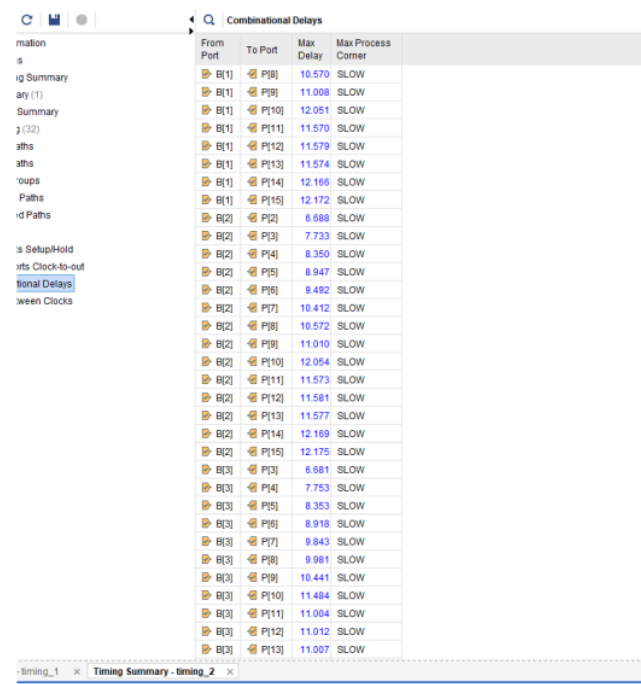


Figure 52 - Wallace timing delay - 5

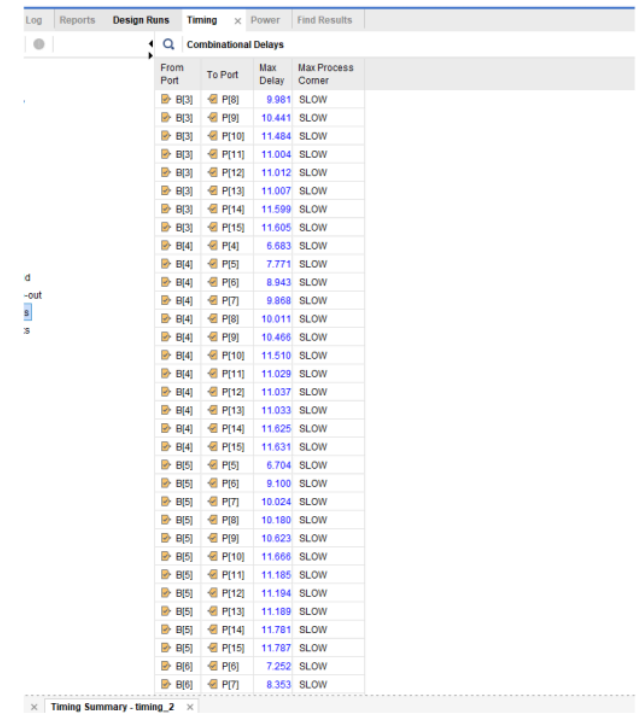


Figure 53 - Wallace timing delay - 6

Combinational Delays

From Port	To Port	Max Delay	Max Process Corner
B[4]	P[13]	11.033	SLOW
B[4]	P[14]	11.625	SLOW
B[4]	P[15]	11.631	SLOW
B[5]	P[5]	6.704	SLOW
B[5]	P[6]	9.100	SLOW
B[5]	P[7]	10.024	SLOW
B[5]	P[8]	10.180	SLOW
B[5]	P[9]	10.623	SLOW
B[5]	P[10]	11.666	SLOW
B[5]	P[11]	11.185	SLOW
B[5]	P[12]	11.194	SLOW
B[5]	P[13]	11.189	SLOW
B[5]	P[14]	11.781	SLOW
B[5]	P[15]	11.787	SLOW
B[6]	P[6]	7.252	SLOW
B[6]	P[7]	8.353	SLOW
B[6]	P[8]	8.923	SLOW
B[6]	P[9]	9.500	SLOW
B[6]	P[10]	10.544	SLOW
B[6]	P[11]	10.063	SLOW
B[6]	P[12]	10.071	SLOW
B[6]	P[13]	10.067	SLOW
B[6]	P[14]	10.659	SLOW
B[6]	P[15]	10.665	SLOW
B[7]	P[7]	6.583	SLOW
B[7]	P[8]	7.287	SLOW
B[7]	P[9]	9.084	SLOW
B[7]	P[10]	10.127	SLOW
B[7]	P[11]	9.646	SLOW
B[7]	P[12]	9.654	SLOW
B[7]	P[13]	9.650	SLOW
B[7]	P[14]	10.242	SLOW
B[7]	P[15]	10.248	SLOW

Figure 54 - Wallace timing delay - 7

Schematic

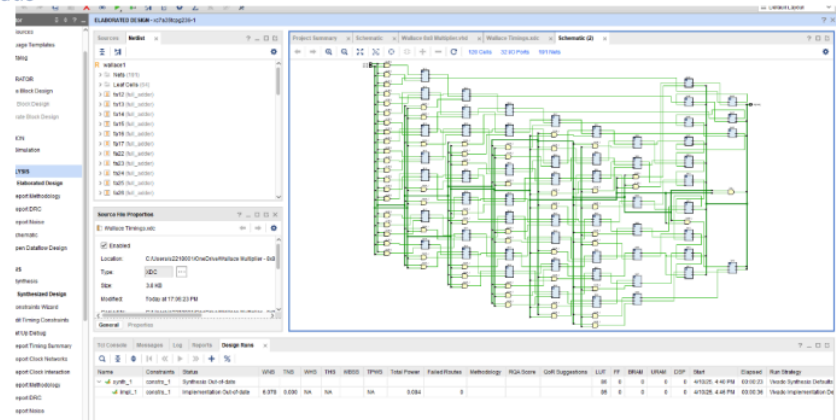


Figure 55 - Wallace Schematic

FPGA Design & Pin Allocation

Allocation of port is an important part in the system. The input ports receive data, such as the numbers to be multiplied, while the output ports deliver the result of the operation. Without properly assigned I/O ports, the design cannot interact and provide any signals.

Braun's Multiplier Pin Allocation

File Edit View Window Design Tools Help																
Q Messages Log Reports Results Timing																
Name	Direction	Interface	Req Off Pin	Package Pin	Fixed	Bank	IO Bus	Max	Val	Drive Strength	Drive Type	Pull Type	IO Config Termination	PULL_TERM	Package Pin Location	
B_A[7] IN			R2	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[6] IN			T1	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[5] IN			U1	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[4] IN			R2	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[3] IN			R3	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[2] IN			T2	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[1] IN			T3	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_A[0] IN			U2	✓	34	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[7] IN			W13	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[6] IN			W14	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[5] IN			V15	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[4] IN			W16	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[3] IN			W17	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[2] IN			W18	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[1] IN			V19	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_W[0] IN			V17	✓	14	LXVIOSS2P	3.300					NONE	✓	NONE	✓	NA
B_P[15] OUT			L1	✓	35	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[14] OUT			P1	✓	35	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[13] OUT			R3	✓	35	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[12] OUT			P3	✓	35	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[11] OUT			U3	✓	34	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[10] OUT			W3	✓	34	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[9] OUT			U5	✓	34	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[8] OUT			V13	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[7] OUT			V14	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[6] OUT			U14	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[5] OUT			U15	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[4] OUT			W19	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[3] OUT			V19	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[2] OUT			U19	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[1] OUT			E19	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA
B_P[0] OUT			U16	✓	14	LXVIOSS2P	3.300	12	✓	✓	✓	NONE	✓	PP_VTT_50	✓	NA

Figure 56 - Braun's pin allocation

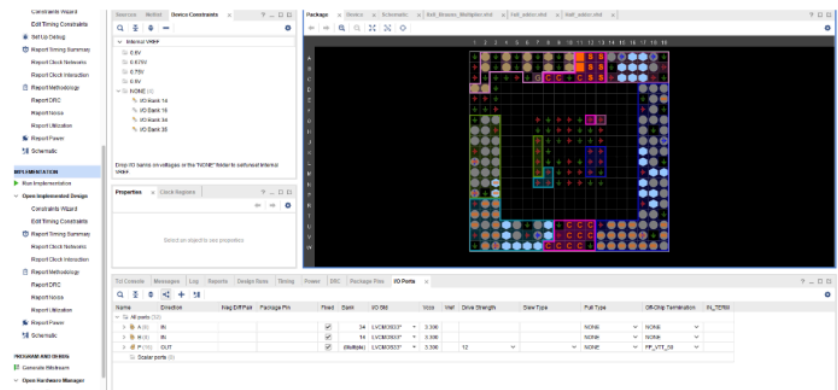


Figure 57 - Braun's device

Wallace Tree Multiplier Pin Allocation

[illegible]

Figure 58 - Wallace pin allocation

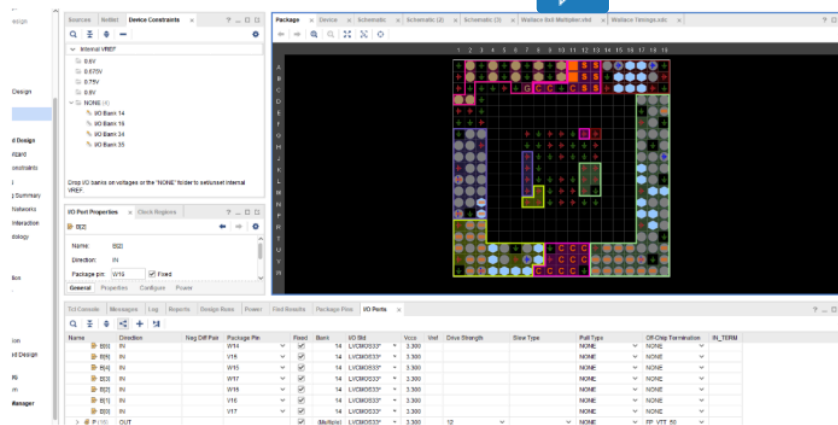


Figure 59 - Wallace device

Results

Braun's Multiplier

Battery Summary

The figure below shows the power summary of the Braun's Multiplier.

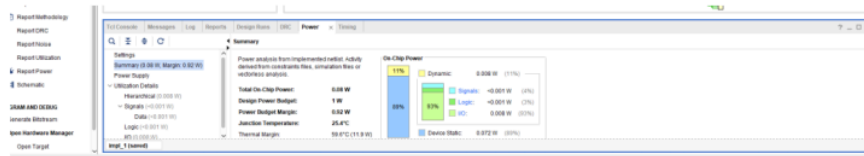


Figure 60 - Braun's power summary

By knowing the power summary of the Braun's multiplier it tells us how much power the circuit uses when running. This helps in understanding if the design is energy efficient and suitable for use in real software's such as FPGAs. If the power is too high, it can cause heating issues or reduce battery life in portable devices. By checking the power summary, we are able to make changes to improve performance and save energy.

The figure below shows what design power value we set, which is 1W.

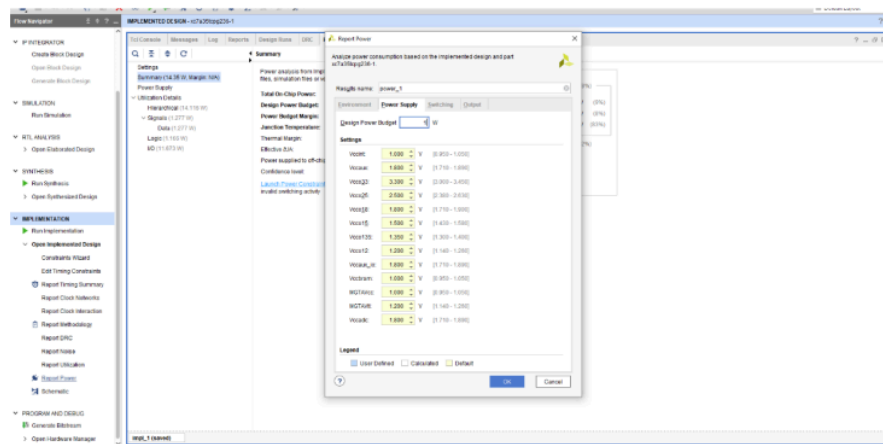


Figure 61 - Braun's set power value

RTL Analysis

The following figure demonstrates the RTL analysis of the design.

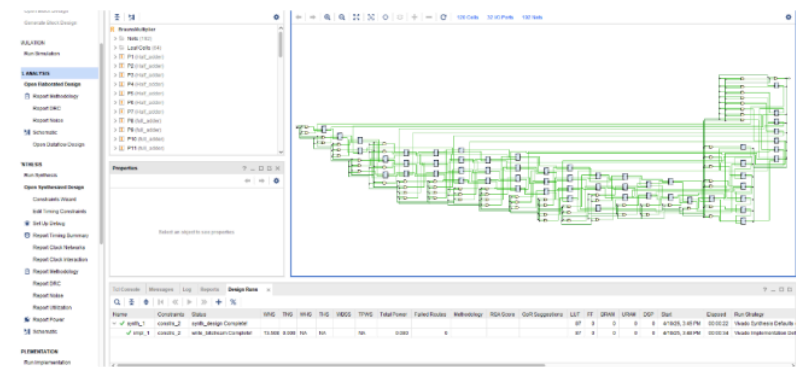


Figure 62 - Braun's RTL analysis

RTL (Register Transfer Level) analysis is important because it shows how data moves between registers and logic in a digital circuit. By checking the RTL, we can make sure the logic is correct, identify errors, and improve the design before moving to implementation.

LUT's

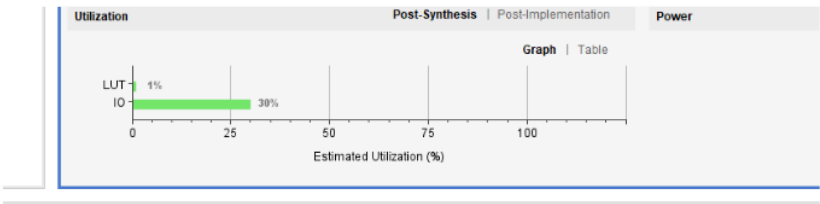


Figure 63 - Braun's LUT's

Utilization in the multiplier refers to how efficiently the software resources of an FPGA, such as LUT's, etc that are used to implement the design. It provides information on how much of the available FPGA capacity is consumed by the multiplier. Monitoring utilization helps designers understand the complexity and size of the design. High utilization may indicate a need to simplify the design, while low utilization shows that more resources are available for other components. [5]

Wallace Tree Multiplier

Battery Summary

The figure below shows the power summary of the Wallace Tree Multiplier.

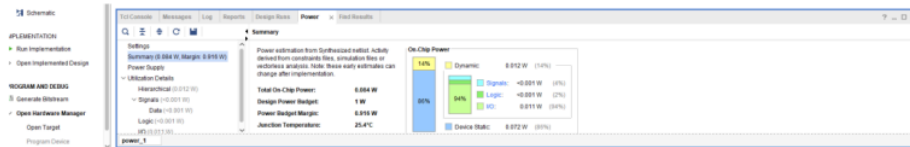


Figure 64 - Wallace power summary

Both Multiplier's were given a design power value of 1W. Both values were kept the same for comparison.

RTL Analysis

The following figure demonstrates the RTL analysis of the design on the Wallace.

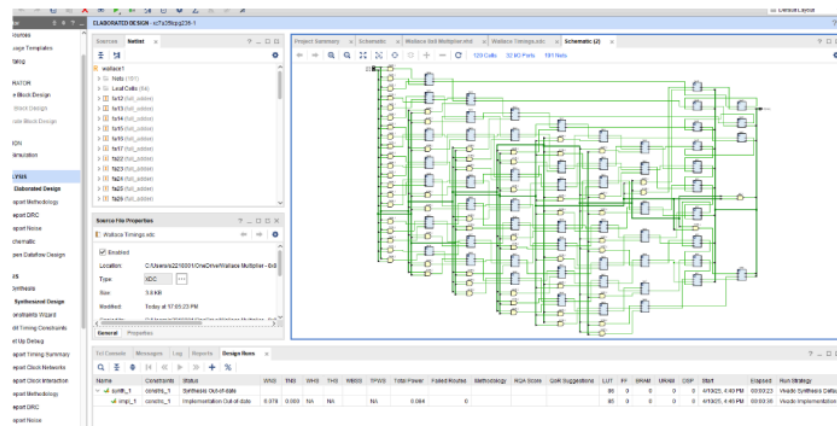


Figure 65 - Wallace RTL analysis

LUT's

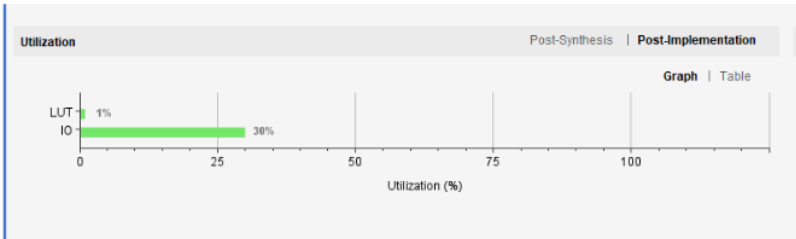


Figure 66 - Wallace LUT's

Recommendations

Recommendations for the design and analysis of the Braun's and Wallace Tree Multiplier are listed below. Firstly design selection based on application, if the application requires a more simple and lower resource usage, then the Braun's multiplier would be suitable. This is due to its smaller systems where power efficiency and ease of implementation are more important than speed. Secondly optimisation for speed. For high speed or performance critical tasks like real time processing, the Wallace Tree multiplier is more suited to the task. It is quicker due to reduction in stages, despite its complexity and slightly higher power consumption. Next would be the FPGA's. Although both designs use FPGA's, monitoring LUT's and power consumption in detail and would help identify any design issue's throughout the system. Another recommendation would be power efficiency. When designing and choosing what multiplier to use, considerations like low power design techniques to further enhance energy efficiency. Finally simulation and testing. Additional testing and verification could be included to improve the design, especially when implementing into real world applications.

Conclusion

In conclusion, this project implemented, analysed and compared two multiplier's, one being Braun's and the other the Wallace Tree Multiplier. The Braun's multiplier and the Wallace Tree multipliers, both use VHDL and FPGA. As shown throughout the report, both simulations, including timings constraints, power usage, input power, etc, have performed and worked correctly when tested.

The Braun's multiplier demonstrated a simpler and more structured design, making it easier to use and suitable for low complexity applications. However, the Wallace Tree multiplier, even though it is a more complex design, it has proved to be more of an efficient system in regards to speed due to its reduction process on the stages.

Both designs met the required timing constraints and power use, making them both suitable options for software implementation. Overall, when choosing between both designs would really depend on the requirements of the application, whether it's simplicity and ease of implementation or speed and performance. [6]

References

- [1 in *Computer Architecture*, John.L.Hennesy & David.A.Patterson, 2019.
]
- [2 "Ripublication," 11 04 2025. [Online]. Available:
] https://www.ripublication.com/acst17/acstv10n5_54.pdf.
- [3 "AMD," 12 04 2025. [Online]. Available: <https://www.amd.com/en/products/adaptive-socs-and-fpgas/fpga.html>.
]
- [4 "IEEE Explore," 04 04 2025. [Online]. Available:
] <https://ieeexplore.ieee.org/abstract/document/8605173>.
- [5 "iaENG," 09 04 2025. [Online]. Available:
] https://www.iaeng.org/publication/IMECS2012/IMECS2012_pp275-277.pdf.
- [6 "Academia," 02 04 2025. [Online]. Available:
] https://www.academia.edu/114227557/FPGA_Based_High_Performance_Double_Precision_Matrix_Multiplication?uc-sb-sw=7975393.

59/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6



Comment 1 | Engagement with literature(20)

The inclusion of their applications and advantages provides useful context and helps in understanding their practical relevance.

PAGE 7



Comment 2 | Engagement with literature(20)

It will be helpful to start with the basic idea of binary multiplication and explain logic circuits like full adder and half adder clearly.



Comment 3 | Knowledge & Understanding(20)

Basic logic circuit explanation and diagrams for binary operations are missing.



Comment 4 | Engagement with literature(20)

It
will be helpful to start with the basic idea of binary multiplication and
explain logic circuits like full adder and half adder clearly.

PAGE 8



Comment 5 | Knowledge & Understanding(20)

Architecture
diagrams for both Braun and Wallace are shown clearly with labels.

PAGE 9



Comment 6 | Engagement with literature(20)

More
in-depth explanation of the working process, signal flow, and technical
comparisons between the two multipliers would strengthen this section.



Comment 7 | Cognitive skills (20)

Wallace
multiplier's dot diagram is neat.

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15



Comment 8 | Practical Application (25)

Give
more detail about port mapping logic.

PAGE 16

PAGE 17

PAGE 18

PAGE 19

Comment 9 | **Practical Application (25)**

Simulation
and hardware results are shown properly, which proves the design works.

Comment 10 | **Cognitive skills (20)**

Delay
diagrams and critical path calculations for both designs are shown clearly.

Comment 11 | **Practical Application (25)**

Screenshots
like schematics, RTL diagrams, and I/O planning are provided is appreciable.



Comment 12 | **Practical Application (25)**

The report mentions pin mapping in the design flow, indicating awareness of hardware setup.



Comment 13 | **Practical Application (25)**

The report includes critical path diagrams, power analysis screenshots, LUT usage data, and area details, which shows effort towards evaluating the design performance and resource utilisation.



Comment 14 | **Cognitive skills (20)**

Future recommendations are noted. They could be improved by incorporating additional criteria.



Comment 15 | **Transferable skills (15)**

The structure and organisation of the report are well-executed, contributing to a clear and logical flow of information.



Comment 16 | **Cognitive skills (20)**

Although results are presented, there is no in-depth discussion linking them to the overall performance or design choice, and comparison is minimal.



Comment 17 | **Engagement with literature(20)**

Adding
more recent papers and doing a proper review of existing multiplier methods
will improve the quality.

GRADING FORM: UNTITLED

REBECCA COLES

58.80

ENGAGEMENT WITH LITERATURE(20)

Assesses clarity of definition, technical explanation, advantages, disadvantages, applications, and use of relevant references. Also includes critical analysis, research gaps, and proper citation and referencing.



Few relevant references are used, and they are poorly integrated. Analysis lacks originality or critical thought. Research gaps are not clearly identified. Referencing is inconsistent and lacks attention to formatting. Minimal variety in sources.

8.2

KNOWLEDGE & UNDERSTANDING(20)

This section evaluates understanding of binary multiplication, multiplier architectures, logic design, and adders. It also includes key concepts like power efficiency, delay path, LUTs, and slack, reflecting application of core engineering principles.



Adequate but limited understanding. Key topics are mentioned (e.g., Braun, Wallace, logic design),

10.1

but with shallow explanation. Necessity for the aspects like delay paths, power basics, and slack needs more explanation.

COGNITIVE SKILLS (20)

This section includes analysis of design structures, performance interpretation, and comparison of results using appropriate techniques. Emphasis is placed on understanding technical relationships, interpreting data, and drawing meaningful conclusions with supporting evidence such as calculations, graphs, and recommendation



Some understanding of analysis, but work is mostly descriptive. Timing and performance metrics are discussed with limited accuracy. Graphs may be missing or poorly used. Conclusions are present but lack depth, and recommendations are vague or underdeveloped.

11.3

PRACTICAL APPLICATION (25)

This section assesses the practical skills in VHDL implementation, port mapping, FPGA pin assignment, simulation, and hardware proof. It also includes synthesis, applying XDC constraints, and interpreting reports on LUTs, power, delay, and timing, reflecting effective tool usage.



Very good execution of practical tasks. VHDL code is functionally sound; port mapping and pin assignments are mostly correct. Simulation and hardware outputs are included with minor gaps. FPGA synthesis and report handling are well managed. Constraints are applied with general accuracy.

20.3

TRANSFERABLE SKILLS (15)

This section evaluates the ability to communicate work clearly and effectively. It covers report writing, clarity of diagrams, structured presentation, and appropriate use of screenshots. It also assesses how well findings are explained, the accuracy with which procedures are followed, and how results are summarised. Communication should be suited to both technical and non-technical audiences, demonstrating professionalism and clarity.



Very good communication. Report is coherent and includes clear diagrams and screenshots. Findings are explained well and structure is mostly consistent. Procedural accuracy and summary are evident. Minor improvements could be made in tailoring language for diverse audiences.

8.9