Analysis and Implementation of a Full Adder Circuit using Xilinx Software

Bobbinpreet Kaur
ECE(AP)
Chandigarh group of colleges
Mohali, India

Girdhari Agarwal
ECE(student)
Chandigarh group of colleges
Mohali, India

Amandeep Kaur
ECE(AP)
Chandigarh group of colleges
Mohali, India

ABSTRACT

This paper presents the novel method to analyze and implement a full adder circuit using VHDL Technology. The results include successful compilation of the VHDL code in the Xilinx software along with the waveforms that prove the legality of the truth table. This paper also shows the effective use of Xilinx software in the analysis of the full adder circuit. It shows the Register Transfer Level (RTL) schematic diagrams and technology schematic diagrams of the different VHDL architectural styles of modeling that include dataflow modeling, behavioral modeling and structural modeling. The analysis includes the detailed analysis of the fitter report and the timing report along with the synthesis report of the design summary. It also shows the chip floor plan of the full adder circuit.

Keywords

Full Adder, Xilinx, VHDL, design.

1. INTRODUCTION

One of the most fundamental arithmetic operations that are widely used in digital system is addition. Addition is the basic operation of all logic and arithmetic operations that can be performed in a digital system. The term 'digital system' is not only limited to a low-level component that can be designed theoretically on paper, but it also extends its hierarchy up to the designing the complete system over a chip or board. It is not always possible to understand the system completely in case of higher levels of hierarchy due to the increased complexity of the digital system at these levels. Thus to overcome this complex situation, VHDL technology is used which makes the design of the system simpler and easy to understand.[6]

This paper describes the analysis and implementation of a full adder circuit with the help of VHDL technology so that the chip or on board design of the full adder system gets easy to implement as it meets the less complexity requirement. One of the most popular and easy approach to VHDL technology is done with the help of Xilinx software that allows us to compile our VHDL code to a machine level program and then implement it on hardware i.e. on the chip or board. The paper also shows that how efficiently our digital system i.e. a full adder can be implemented and analyzed completely using this software.[9] The analysis part shows the RTL view, technological map, chip floor plan and wave forms of voltage v/s time with varying input and their respective output so as to verify the truth table of the full adder circuit.

Following text is divided into five sections. Section 2 refers to the previous papers and the literature reviews of those papers on the same field. Section 3 describes the full adder and its design, section 4 describes the various VHDL architectural styles, section 5 describes the implementation of the full adder circuit using the various architectural designs as discussed in section 4 and the last section i.e. section 6 describes the analysis of the design.

2. LITERATURE REVIEW

Rupesh Prakash Raghatate explained a brief overview about the full adder circuit using VHDL[1] in the paper titled, "Design and implementation of Full adder using VHDL and its verification in analog domain". He implemented the circuit using VHDL in QuartusII.

Rajendra Kumar explained about the implementation of full adder in the addition of multi bit using the carry look ahead[3] in his paper titled, "Performance analysis of different bit carry look ahead adder using VHDL Environment."

3. FULL ADDER DEESIGN

A full adder is a combinational digital circuit with three inputs and two outputs. The full adder circuit has the ability to add the three 1-bit binary numbers (Input1, Input2, Input3) and results in two outputs (Sum, Carry). The fig. 1 below shows the basic logic diagram of the full adder circuit, Table 1 that shows the truth table and fig. 2 shows the K-Map reduction for the two outputs from the truth table.[4]

3.1 Logic Design

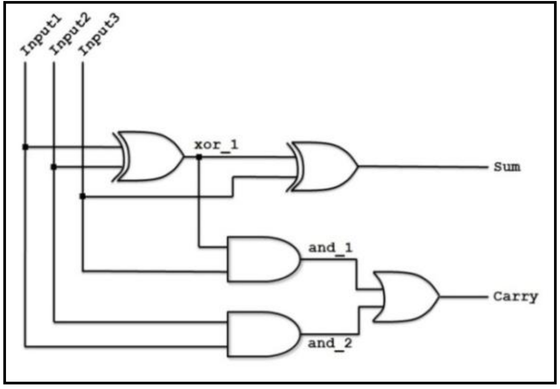
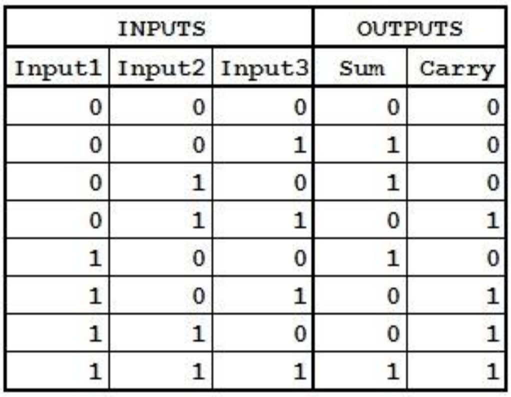


Fig 1: Basic logic diagram for a full adder[5]

3.2 Truth Table

Table 1. Truth Table for full adder



INPUTS			OUTPUTS	
Input1	Input2	Input3	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

3.3 Karnaugh Map

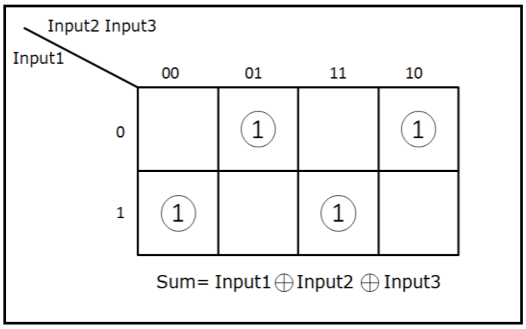


Fig 2: K-Map for logic Sum

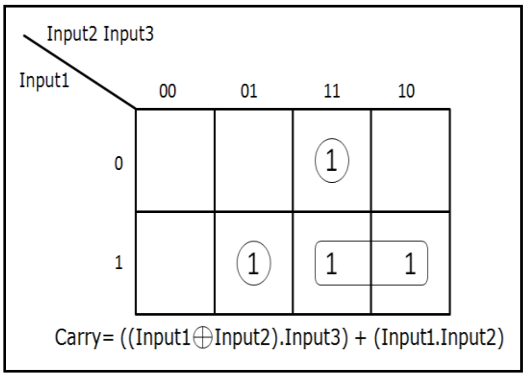


Fig 3: K-Map for logic Carry

4. VHDL

Hardware Descriptive Languages (HDL) are the basic language that are used for the designing of most of the digital circuits using software tools. VHDL is one of most important type of HDL. VHDL is the abbreviation for “Very High Speed Integrated Circuit Hardware Description Language (VHSIC HDL)”.This language is combination of Concurrent, sequential, wave simulation, timing and net list simulation language [2]

VHDL is supposed to be the heart for the production of electronic design of any digital circuit. With the advancement of time and the shrinking of the semiconductor device dimensions into compact sizes, the VHDL has gained a lot of importance.

There are three architectural styles of modeling of a VHDL statement. The internal functionality of digital circuit can be specified using any of the modeling styles as discussed below [8]:

4.1 Structural

A structural architectural design refers to the architectural design where all the used components are interconnected to each other.

4.2 Dataflow

A dataflow architectural design refers to the architectural design where a set of concurrent assignment statements are used to design the program.

4.3 Behavioral

A behavioral architectural design refers to the architectural design where a set of sequential assignment statements are used to design the program.

5. IMPLEMENTATION

Entity statement[7] :

```
entity Full_Adder is
   Port ( Input1 : in  STD_LOGIC;
          Input2 : in  STD_LOGIC;
          Input3 : in  STD_LOGIC;
          Sum : out  STD_LOGIC;
          Carry : out  STD_LOGIC);
end Full_Adder;
```

Architectural design using:

5.1 Structural

```
architecture Structural of Full_Adder is
 signal xor_1, and_1, and_2:STD_LOGIC;
  component XOR1 port(A, B : in STD_LOGIC;
                              X: out STD_LOGIC);
  end component;
  component OR1 port(M, N : in STD_LOGIC;
                            Y: out STD_LOGIC);
  end component;
  component AND1 port(P, Q : in STD_LOGIC;
                              Z: out STD_LOGIC);
  end component;
  begin
 X1:XOR1 port map(Input1, Input2, xor_1);
 X2:XOR1 port map(xor_1, Input3, Sum);
 A1:AND1 port map(Input1, Input2, and_1);
 A2:AND1 port map(xor_1, Input3, and_2);
 O1:OR1 port map(and_1, and_2, Carry);
end Structural;
```

5.2 Dataflow

```
architecture DataFlow of Full_Adder is
signal xor_1, and_1, and_2:STD_LOGIC;
 begin
          xor_1 <= Input1 xor Input2;
          and_1 <= Input1 and Input2;
          and_2 <= xor_1 and Input3;
          Sum <= xor_1 xor Input3;
          Carry <= and_1 or and_2;
end DataFlow;
```

5.3 Behavioral

```
architecture Behavioral of Full_Adder is
begin
process(Input1, Input2, Input3)
begin
if (Input1='0') and (Input2='0') and (Input3='0')
          then Sum<='0'; Carry<='0';
elsif (Input1='0') and (Input2='0') and (Input3='1')
```

```
        then Sum<='1'; Carry<='0';
elsif (Input1='0') and (Input2='1') and (Input3='0')
        then Sum<='1'; Carry<='0';
elsif (Input1='0') and (Input2='1') and (Input3='1')
        then Sum<='0'; Carry<='1';
elsif (Input1='1') and (Input2='0') and (Input3='0')
        then Sum<='1'; Carry<='0';
elsif (Input1='1') and (Input2='0') and (Input3='1')
        then Sum<='0'; Carry<='1';
elsif (Input1='1') and (Input2='1') and (Input3='0')
        then Sum<='0'; Carry<='1';
elsif (Input1='1') and (Input2='1') and (Input3='1')
        then Sum<='1'; Carry<='1';
end if;
end process;
end Behavioral;
```

6. ANALYSIS

6.1 Register Transfer Logic(RTL) schematic of the above written architectural designs:

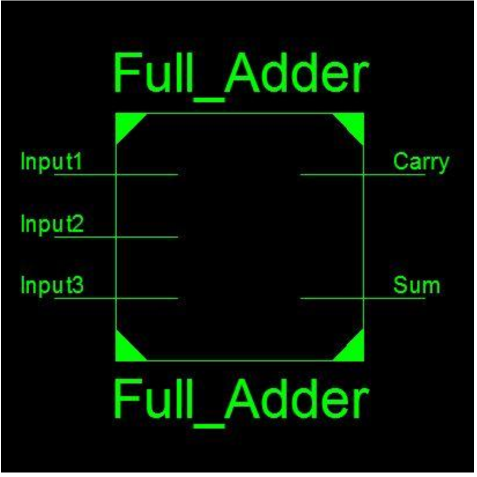


Fig 4: RTL Schematic for the block

6.1.1 Structural

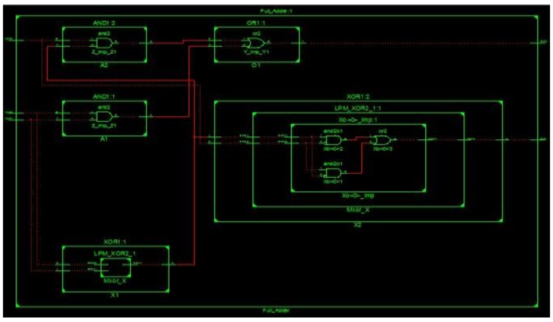


Fig 5: RTL Schematic for structural

6.1.2 Dataflow

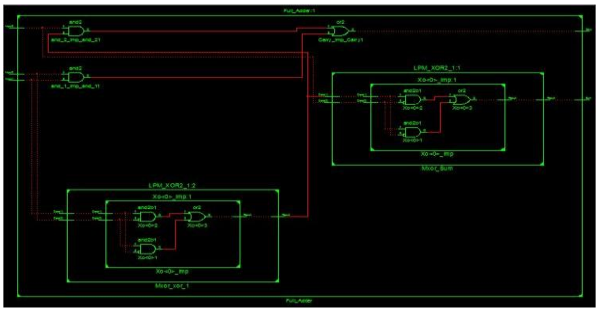


Fig 6: RTL for Dataflow

6.1.3 Behavioral

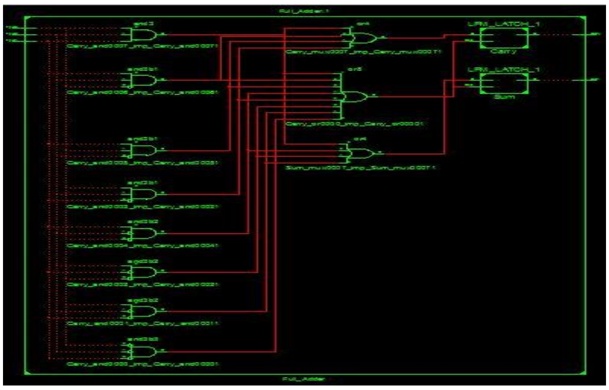


Fig 7: RTL for Behavioral

6.2 Wave Simulation

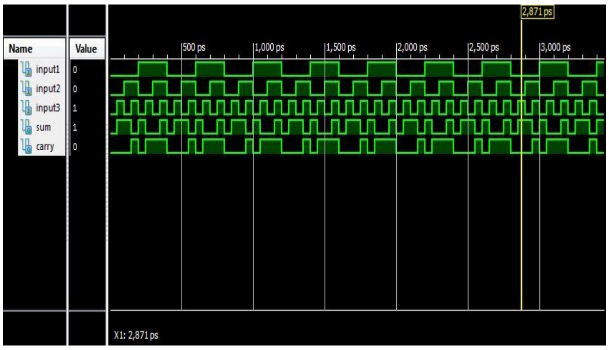


Fig 8 – Wave simulation

6.3 Fitter Report

TABLE II. SUMMARY

Design Name	Full_Adder
Fitting Status	Successful
Software Version	M.81d
Device Used	XA9536XL-15-VQ44

TABLE III. RESOURCES SUMMARY

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
2/36 (6%)	6/180 (4%)	0/36 (0%)	5/34 (15%)	6/108 (6%)

TABLE IV. PIN RESOURCES

Signal Type	Required	Mapped
Input	3	3
Output	2	2
Bidirectional	0	0
GCK	0	0
GTS	0	0
GSR	0	0

TABLE V. EQUATIONS

********** Mapped Logic **********
Carry <= ((Input1 AND Input3) OR (Input1 AND Input2) OR (Input3 AND Input2));
Sum <= NOT (Input2 XOR Sum <= NOT (((Input1 AND Input3) OR (NOT Input1 AND NOT Input3)));

6.4 Timing Report

TABLE VI. DATA SHEET REPORT FOR PAD TO PAD LIST

Source Pad	Destination Pad	Delay
Input1	Carry	15.500
Input1	Sum	15.500
Input2	Carry	15.500
Input2	Sum	15.500
Input3	Carry	15.500
Input3	Sum	15.500

6.5 Chip Floor Diagram

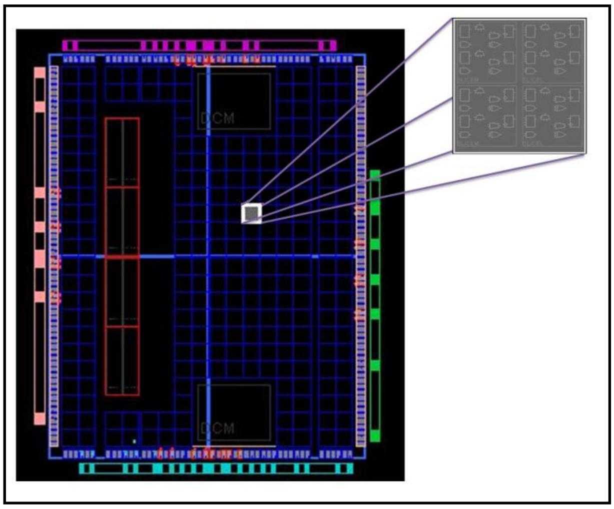


Fig 9 – Chip Floor Diagram

7. CONCLUSION

This paper implements Full adder circuit using Xilinx software that help us to design a full adder with the help of a general 44 pin cmos device.The design can be transferred to a CMOS device.XA9536XL-15-VQ44 is the device used to successfully fit the layout of Full adder on chip, the XA9536XL-15-VQ44 has 44 pins out of which pin 30, 31 and 28 are used as inputs and pin 41 and 38 are used as sum and carry output respectively. The wave simulations show about all the possible inputs for a full adder and their respective outputs to verify the truth table. Further, more improvement can be made in the design for optimized implementation by varying the design methodology.

8. FUTURE SCOPE

Further in future this circuit can be used for addition of multi bit numbers by using the concept of carry look ahead addition in which the carry from the least significant bit is carried to its successive significant bit. Also the parallel adder can be designed with the help of this full adder circuit as a basic design unit. A cascaded network can be created with this circuit as one block and the output of one stage acting as input to the second stage. The effectiveness of the circuit can be checked by calculating the parameters such as Delay and Time for simulation.

9. REFERENCES

[1] Rupesh Prakash Ragahtate, 2013, “Design and implementation of Full adder using VHDL and its verification in analog domain”, International Journal of Engineering Science Invention, Volume 2 Issue 4 ǁ April. 2013 ǁ PP.35-39.

[2] Jayaram Bhasker, A VHDL Primer(PTR Prentice Hall Englewood cliffs, New Jersey 07632).

[3] Rajender Kumar, Sandeep Dahiya, 2013, “Performance analysis of different bit carry look ahead adder using VHDL Environment”, International Journal of Engineering Science and innovative technology, Volume 2 Issue 4 ǁ April. 2013 ǁ PP.80-88.

[4] A. Anand Kumar, Fundamentals of Digital Circuits(PHI 2nd edition).

[5] Prashant Gurjar and Rashmi Solanki, 2011, VLSI Implementation of adders for high speed ALU., International Journal of Computer Applications, vol. 29- No. 10, PP-11-15.

[6] Floyd, Digital Fundamentals, Pearson Publications, 10th edition.

[7] Xilinx inc., Xilinx student edition 4.2i, PHI. 1st edition, 2002

[8] Pedroni, 2004 Circuit design with VHDL, MIT Press, Cambridge, England, PP-159-186.

[9] Chiuchisan, Potorac, 2010, Finite state machine design and VHDL coding techniques, 10th international conference on development and applications systems, Romania, PP-273-278.