An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Electrical and Computer Engineering Department
Digital Circuit Design II Lab : (10636391)
Report Grading Sheet

Instructor Name: Ashraf Armoush	**Experiment #: 2**
Academic Year: 2023-2024	**Performed on: 25/4/2024**
Semester: 2nd Semester	**Submitted on: 10/5/2024**
Student Names:	
1-Hamza Younes	2-
3-	4-
5-	6-

Evaluation Criterion	CLO	Grade	Points
Abstract and Aims Aims and idea of the experiment are clearly stated in simple words		10	
Introduction, Apparatus and Procedures Introduction is complete and well-written, all grammar/spelling correct, Appropriate background information related to the principles of the experiment is provided. The list of apparatus and procedures are also provided		15	
Experimental Results, Calculations and Discussion Results analyzed correctly. Experimental findings adequately and specifically summarized, in graphical, tabular, and/or written form. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis as needed.		50	
Conclusions Conclusions summarize the major findings from the experimental results with adequate specificity. Highlighting the most important results		15	
Appearance Title page is complete, page numbers applied, content is well organized, correct spelling, fonts are consistent, good visual appeal. You have also to use reference for the information you provide		10	
Total		100	

An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Aims and Abstract:

In this experiment our goal was to design and implement in VHDL a four-bit adder in both a structural and behavioral manner and simulating them on Vivado. After doing so we materialized our work by implementing the design on an FPGA board (ZedBoard).

Introduction:

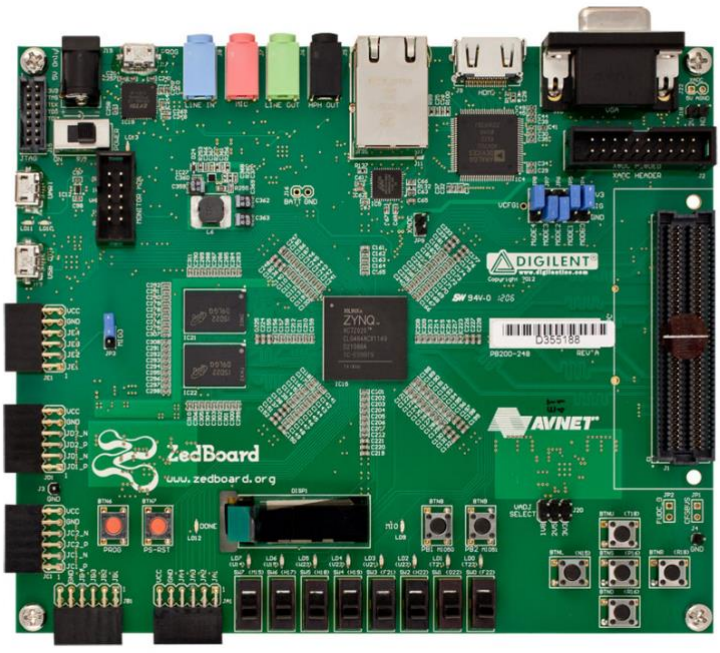
From our prior knowledge gained in the Digital 2 course there are three levels of writing code in hardware description language Behavioral, Structural and gate level. In this experiment we utilized 2 of those 3 methods where we were required to build a four-bit adder in both a structural and behavioral manner.

In Structural coding the focus is on designing detailed building blocks in a black-box manner. We did so by designing half adders which were the backbone of our full adders which were the main building block of our structural design of the four-bit adder. Meanwhile in our behavioral design, no building blocks were required as behavioral design favors high functionality over detailed building blocks therefore we utilized the ieee.std_logic_1164 and ieee.std_logic_unsigned libraries in VHDL to sum the inputs and assign the summation to the output.

To martialize our code we designed a testbench to simulate our code on Vivado. we then synthesized, implemented the code and generated a bit stream file to test our work on an FPGA board (ZedBoard).

Apparatus:

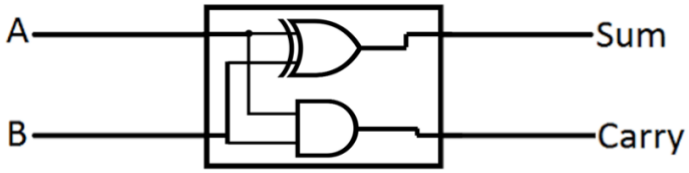
AMD Vivado ML Edition
ZedBoard Zynq-7000 Development Kit



Procedure:

Half adder component

The half adder is made up of two gates an xor gate that provides the sum of the two input bits and an and gate that provides the carry of that summation below is the truth table of the Half Adder, its circuit diagram and its design in VHDL.



A	B	Carry	Sum
0	0	0	0
1	0	0	1
0	1	0	1
1	1	1	0

$$Sum = A \oplus B$$
$$Carry = A \,.\, B$$

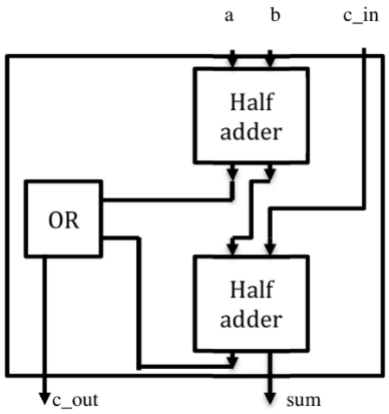
```
entity HA is
    Port ( x : in STD_LOGIC;
           y : in STD_LOGIC;
           z : out STD_LOGIC;
           c : out STD_LOGIC);
end HA;

architecture Behavioral of HA is

begin
z<= x xor y;
c<= x and y;
end Behavioral;
```

Full adder component

The Full adder is made up of two Half adders and an or gate where the two input bits are summed up in the first half adder. The output of that summation is inserted in the 2nd half adder with the C_in bit which provide the final Sum output. The final C_out output is produced by or'ing the carry outputs of the two half adders. Below is the truth table of the full adder, its circuit diagram and it's design in VHDL.



Inputs			Outputs	
A	B	C_{in}	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

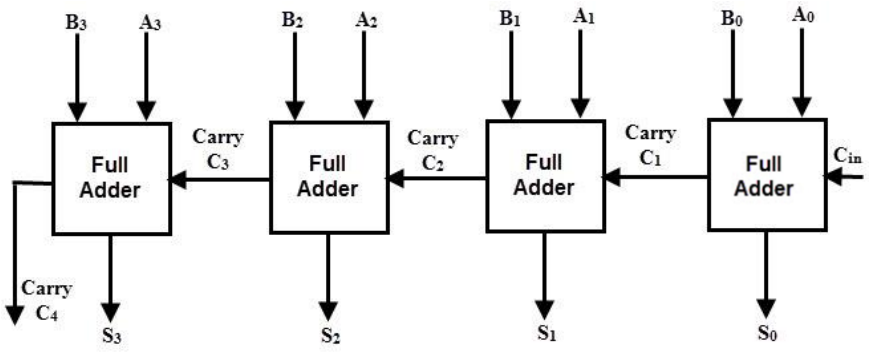
```
entity FA is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c_in : in STD_LOGIC;
           sum : out STD_LOGIC;
           c_out : out STD_LOGIC);
end FA;

architecture Behavioral of FA is
signal sum1,cout1,cout2 : STD_LOGIC;
component HA is port(
x,y:in STD_LOGIC;
z,c:out STD_LOGIC);
end component;

begin
H1:HA Port map(a,b,sum1,cout1);
H2:HA Port map(c_in,sum1,sum,cout2);
c_out<=cout1 or cout2;
end behavioral;
```

Four-Bit adder

The Four-Bit adder is made up of four Full Adders which adds each bit of each variable (1st bit (A) +1st bit (B), 2nd bit (A) +2nd bit (B) etc..) below is the circuit diagram of the Four-bit Adder an it's design in VHDL In both the structural and behavioral manner.



Structural:

```
entity FourBitAdder is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           S : out STD_LOGIC_VECTOR (3 downto 0);
           Cout : out STD_LOGIC);
end FourBitAdder;

architecture Behavioral of FourBitAdder is
signal C0,C1,C2:STD_LOGIC;
component FA is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           c_in : in STD_LOGIC;
           sum : out STD_LOGIC;
           c_out : out STD_LOGIC);
end component;

begin
FA1:FA Port map(A(0),B(0),'0',S(0),C0);
FA2:FA Port map(A(1),B(1),C0,S(1),C1);
FA3:FA Port map(A(2),B(2),C1,S(2),C2);
FA4:FA Port map(A(3),B(3),C2,S(3),Cout);
end Behavioral;
```

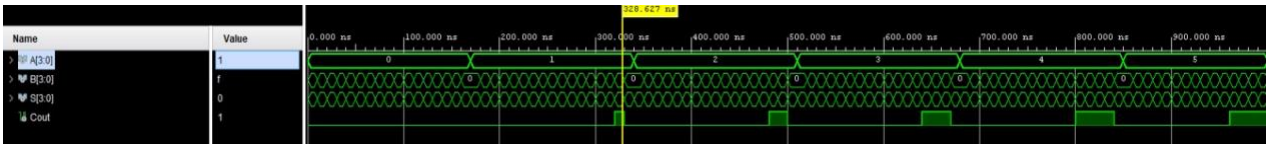
Behavioral:

```
entity FourBitAdder is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           S : out STD_LOGIC_VECTOR (3 downto 0);
           Cout : out STD_LOGIC);
end FourBitAdder;

architecture Behavioral of FourBitAdder is
begin
  process(A, B)
    variable temp_sum: STD_LOGIC_VECTOR (4 downto 0);
  begin
   temp_sum:=('0'&A)+('0'&B);
   S<=temp_sum(3 downto 0);
   Cout <= temp_sum(4);
  end process;
end Behavioral;
```

Four-Bit adder testbench

To test the validity of our design we designed a testbench to test all input cases. Below is the code and output wave we got.




```
architecture Behavioral of FourBitAdder_tb is
component FourBitAdder is
Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           S : out STD_LOGIC_VECTOR (3 downto 0);
           Cout : out STD_LOGIC);
end component;
signal A,B,S : std_logic_vector (3 downto 0);
signal Cout : std_logic;

begin

uut: FourBitAdder port map(
A => A , B =>B , S=> S, Cout => Cout );

stimulus: process
begin
a <= "0000" ;
for i in 0 to 15 loop
b <= "0000";
for k in 0 to 15 loop
wait for 10 ns;
b <= b + '1';
end loop;
wait for 10 ns;
a <= a + '1';
end loop;
wait;
end process;


end Behavioral;
```

Conclusion:

In this experiment, we successfully designed and implemented a 4-bit adder using an FPGA (ZedBoard) and Vivado. We gained hands-on experience with VHDL coding, and explored both structural and behavioral descriptions of digital circuit designing.