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: Dr. Ashraf Armoush	**Experiment #: 2**
Academic Year: 2024/2025	**Performed on: 15/2/2024**
Semester: 2st semester	**Submitted on:**
Student Names:	
1-Ahmad Ashayer	2-Asem Diab
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	

-Abstract and Aims-

Design the Full adder by using Half adder description, also design the 4-bit adder by Structural and behavioral description.

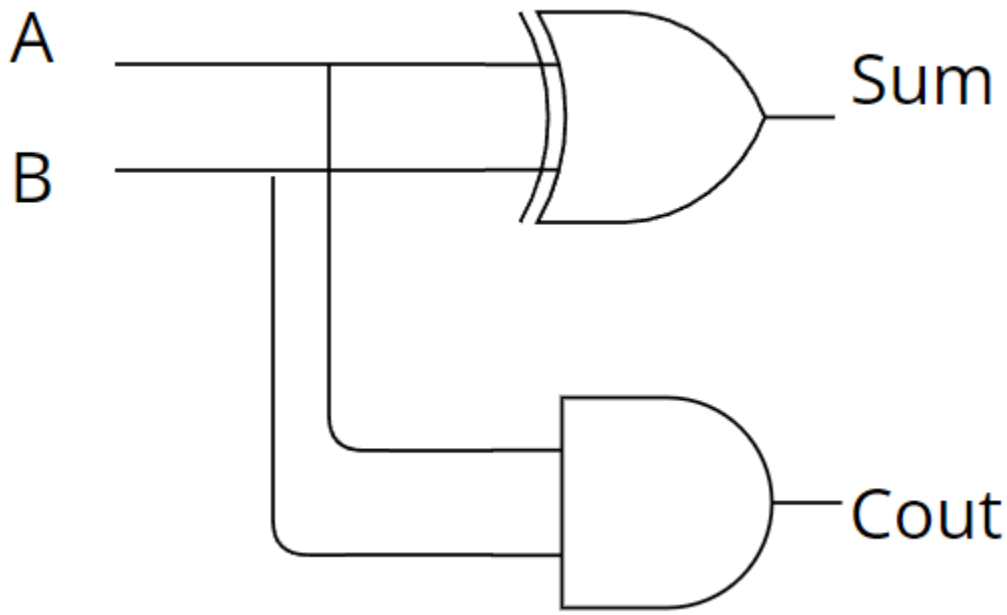
-Introduction-

Half Adder

A circuit that has 2 input and two outputs, one of the outputs is Sum and the other is the carry,

The sum designed with a simple XOR gate between the two inputs, the carry designed with

AND gate as the figure shown:-

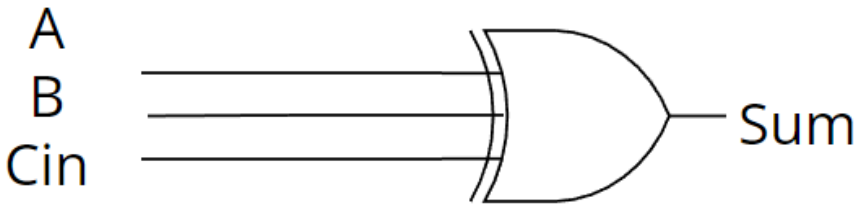


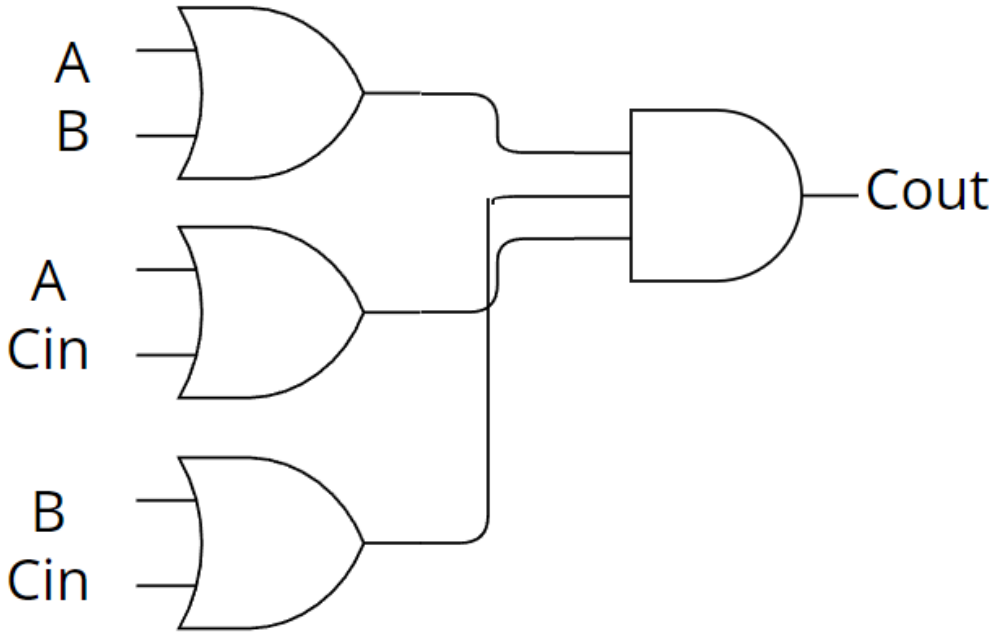
And the truth table of this circuit:-

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Full Adder

The full adder is same as half adder, but with additional input, and the equation of the carry is different from the half adder as figure shown:-





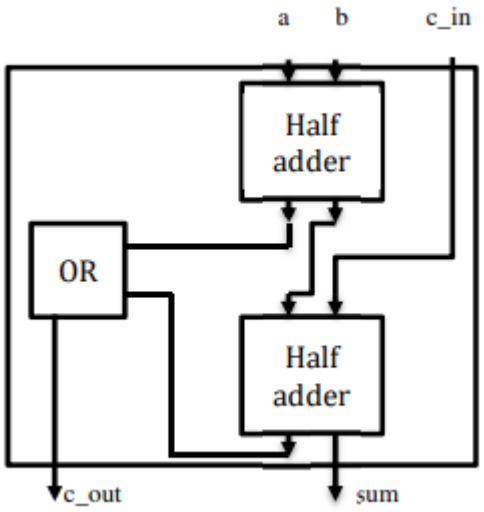
Sum=A xor B xor Cin

Cout=AB + ACin+BCin

And this is the truth table for this digital circuit:-

A	B	Cin	Sum	Cout
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**	**0**	**0**

In the lab the Full adder can be designed by using half adders, in details it designed by two half adders and one OR gate as shown:-

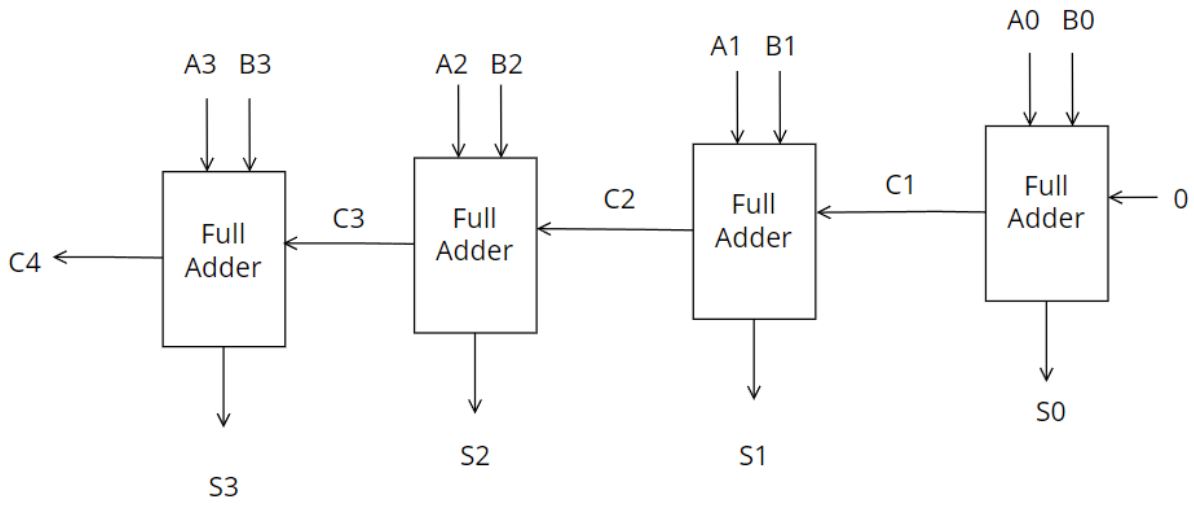


4-Bit-Adder

This digital circuit designed as two inputs and each input described as 4-bit vector with additional carry bit, And designed it in structural and behavioral description, And to design this digital circuit it will designed

by the previous full adder and it will need four of it, with additional carry as shown in the circuit.

In this experiment the Zed-Board has only 8 switches, so we set the carry input as 0.



- Experimental Results -

- Half Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity HA is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           s : out STD_LOGIC;
           c : out STD_LOGIC);
end HA;

architecture Behavioral of HA is

begin
s<= a xor b;
c<= a and b;

end Behavioral;
```

- Full Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FA is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           cin : in STD_LOGIC;
           s : out STD_LOGIC;
           cout : out STD_LOGIC);
end FA;
architecture Behavioral of FA is
component HA is
    Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           s : out STD_LOGIC;
           c : out STD_LOGIC);
end component;
signal s1:std_logic;
signal c1:std_logic;
signal c2:std_logic;
begin
HA1 : HA port map(a,b,s1,c1);
HA2: HA port map(cin,s1,s,c2);
cout<=c1 or c2;
end Behavioral;
```

Figure 1 : implemention of FA

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity FA_tb is...
architecture Behavioral of FA_tb is
component FA is  Port ( a : in STD_LOGIC;...
signal a1:std_logic;signal b1:std_logic;signal cin1:std_logic;
signal s1:std_logic;signal cout1:std_logic;
begin
FA1: FA port map(a1,b1,cin1,s1,cout1);
p1:process
begin
wait for 40 ns;
a1<='0';b1<='0';cin1<='0';
wait for 10 ns;
a1<='0';b1<='0';cin1<='1';
wait for 10 ns;
a1<='0';b1<='1';cin1<='0';
wait for 10 ns;
a1<='0';b1<='1';cin1<='1';
wait for 10 ns;
a1<='1';b1<='0';cin1<='0';
wait for 10 ns;
a1<='1';b1<='0';cin1<='1';
wait for 10 ns;
a1<='1';b1<='1';cin1<='0';
wait for 10 ns;
a1<='1';b1<='1';cin1<='1';
wait for 10 ns;
wait;
end process;
end Behavioral;
```

Figure 2: Testbench for FA

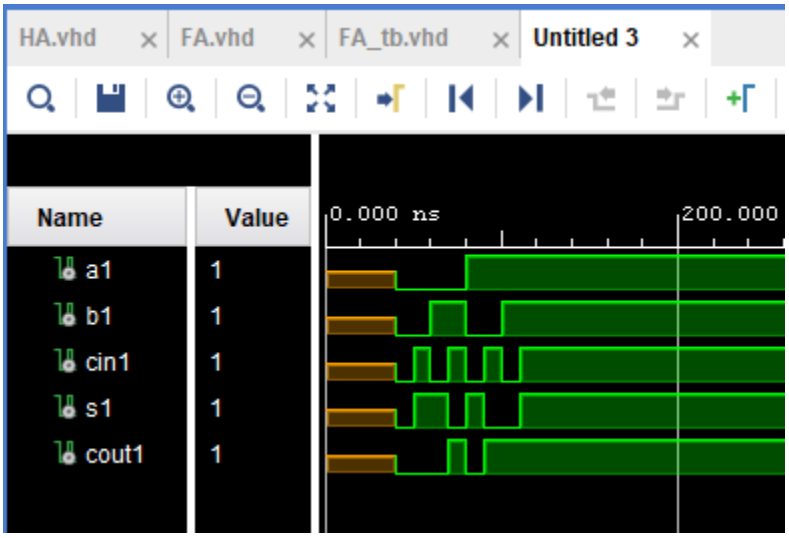


Figure 3: wave for sumlation of FA

- Structure 4bit adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
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
component FA is Port ( a : in STD_LOGIC;
           b : in STD_LOGIC;
           cin : in STD_LOGIC;
           s : out STD_LOGIC;
           cout : out STD_LOGIC);
end component;
signal c : std_logic_vector (3 downto 0);
begin
Fa1: FA port map(a(0),b(0),'0',s(0),c(1));
Fa2: FA port map(a(1),b(1),c(1),s(1),c(2));
Fa3: FA port map(a(2),b(2),c(2),s(2),c(3));
Fa4: FA port map(a(3),b(3),c(3),s(3),cout);

end Behavioral;
```

Figure 4:stucture Adder implemtion

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.ALL;
use IEEE.numeric_std.All;
entity fourbitadder_tb is
--  Port ( );
end fourbitadder_tb;
architecture Behavioral of fourbitadder_tb is
component fourbitAdder is...
signal a :STD_LOGIC_VECTOR (3 downto 0);
signal b :STD_LOGIC_VECTOR (3 downto 0);
signal s :STD_LOGIC_VECTOR (3 downto 0);
signal cout :STD_LOGIC;
begin
adder: fourbitAdder port map( a ,b,s,cout );
p1: process
begin
a<="0000";
b<="0000";
outerloop:for i in 0 to 16 loop
innerloop:for j in 0 to 16 loop
wait for 5 ns;
b<=b+1;
end loop ;
a<=a+1;
end loop ;
end process p1;
end Behavioral;
```

Figure 5: structural Adder testbench

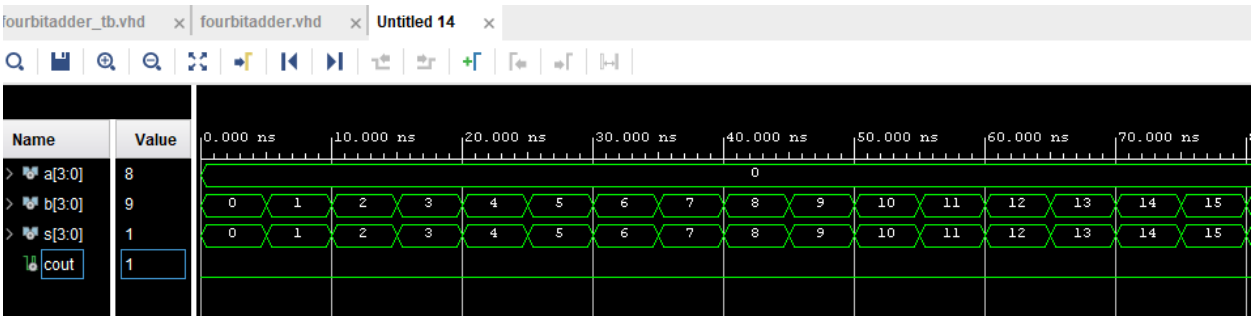


Figure 6 : structural Adder wave

- Behavior 4bit Adder

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.ALL;

entity behAdder 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 (
           cout : out STD_LOGIC);
end behAdder;

architecture Behavioral of behAdder is

signal s1: std_logic_vector(3 downto 0);
signal c:std_logic;
signal res:std_logic_vector (4 downto 0);

begin
p1: process(a,b)
variable asig:std_logic_vector(4 downto 0);
variable bsig:std_logic_vector (4 downto 0);
begin
asig:='0'&a;
bsig:='0'&b;
res <=asig + bsig;
end process;
cout<=res(4);
s<=res(3 downto 0);
end Behavioral;
```

Figure 7 : behavior Adder implemention

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.ALL;
use IEEE.numeric_std.All;
entity behav_tb is--  Port ( );
end behav_tb;
architecture Behavioral of behav_tb is
component behAdder 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 :STD_LOGIC_VECTOR (3 downto 0);
signal b :STD_LOGIC_VECTOR (3 downto 0);
signal s :STD_LOGIC_VECTOR (3 downto 0);
signal cout :STD_LOGIC;
begin
adder: behAdder port map( a ,b,s,cout );
p1: process
begin
a<="0000";
b<="0000";
outerloop:for i in 0 to 16 loop
innerloop:for j in 0 to 16 loop
wait for 5 ns;
b<=b+1;
end loop ;
a<=a+1;
end loop ;
end process p1;
end Behavioral;
```

Figure 8 : behavior Adder testbench

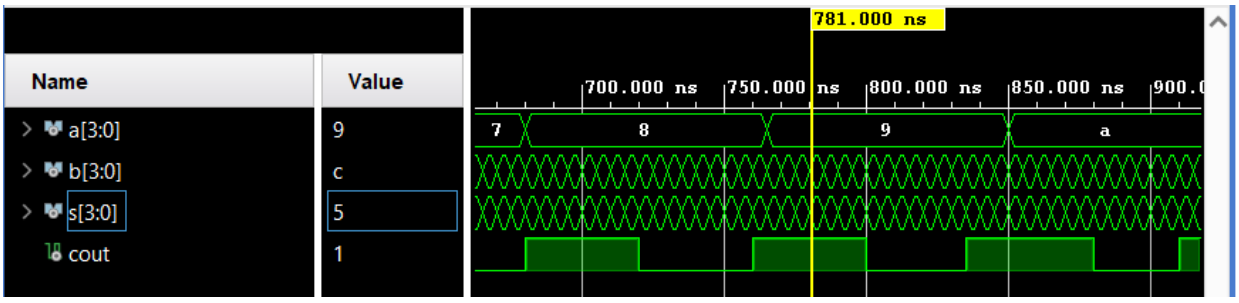


Figure 9 : behavior Adder wave

- Conclusions -

In this experiment we learned how to use Zed-Board and Xilinx Vivado Software for coding how to use it to design digital circuits in Verilog and VHDL, and we learned how to implement the 4-bit-adder in structural and behavioral description by using 4 Full-Adders.

- Appearance-

https://online.visualparadigm.com/app/diagrams/#diagram:proj=0&type=LogicDiagram&width=11&height=8.5&unit=inch