An-Najah National University
Department of Computer Engineering
Digital design 2 LAB -10636391-
Second Semester 2020/2021

جامعة النجاح الوطنية
قسم هندسة الحاسوب
مختبر تصميم دوائر الكترونية 2 - 10636391-
الفصل الأول 2021/2020

Instructor: Anas Toma

NAME	ID NUM	SEC
1- أحمد عبد الله جبر القرم	11819195	3/8:00-11:00
-	-	-

Experiment 2: 4-bit Adder with Structural and Behavioral Implementation

Introduction :

In this experiment I will try to design an 4 Bit Adder using (vhdl) and Zboard from Xilinx

Step by step in tow part :

1- Structural Implementation: starting from half adder .
2- Behavioral Implementation : using IEEE library.

Tools used in Lap :

1- Computer lap.
2- Vivado software .
3- Zboard from Xilinx.
4- VHDL

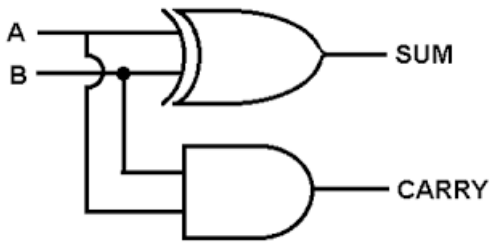
Part 1 : Structural Implementation.

Procedure :

At first build the half adder that contain (XOR) and (AND) gate

$S = A \oplus B \quad Cout = A.B$

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



Project Summary | fa.vhd | bit4Adder.vhd | Adder.vhd *

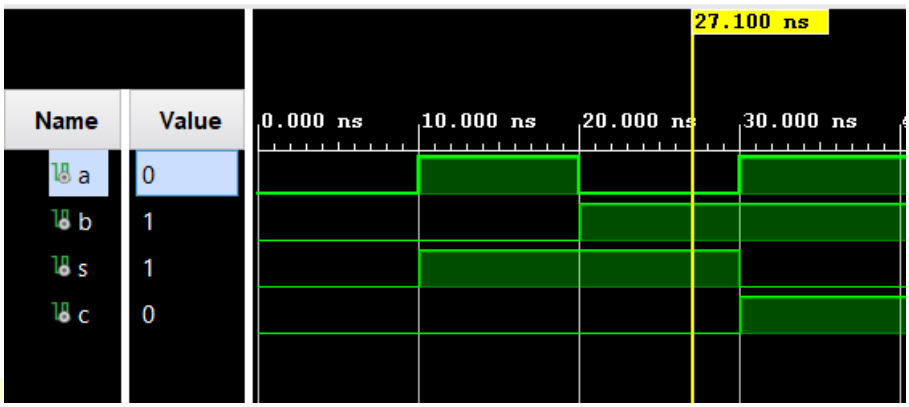
C:/Users/Ahmad/Desktop/A4BitAdder/A4BitAdder.srcs/sources_1/new/Adder.vhd

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity Adder is
4      port (a:in std_logic;
5            b:in std_logic;
6            s:out std_logic;
7            c:out std_logic
8            );
9  end Adder;
10 architecture arc of adder is
11 begin
12 s <= a xor b;
13 c <= a and b ;
14 end arc ;
15
16
```

Simulation for Half Adder:

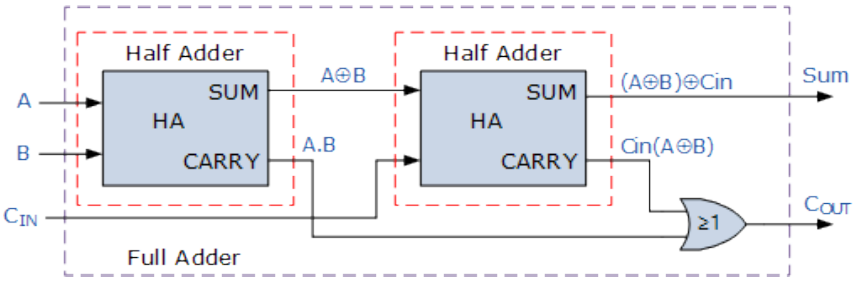
Testbench code:

```
6
7   architecture Behavioral of hadders is
8   component adder port (a:in std_logic;b:in std_logic;s:out std_logic;c:out std_logic);
9   end component;
10  signal a:std_logic:='0';
11  signal b:std_logic:='0';
12  signal s:std_logic:='0';
13  signal c:std_logic:='0';
14  begin
15  uut : adder port map (a=>a,b=>b,s=>s,c=>c);
16  sim:process
17  begin
18
19      a<='0';
20      b<='0';
21      wait for 10 ns;
22
23      a<='1';
24      b<='0';
25      wait for 10 ns;
26
27      a<='0';
28      b<='1';
29      wait for 10 ns;
30
31      a<='1';
32      b<='1';
33      wait for 10 ns;
34  wait;
35  end process;
36  end Behavioral;
```



Then build Full Adder where Composed of 2 (Half Adder) and (OR) gate as shown :

A	*B*	Cin	*S*	*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	1	1



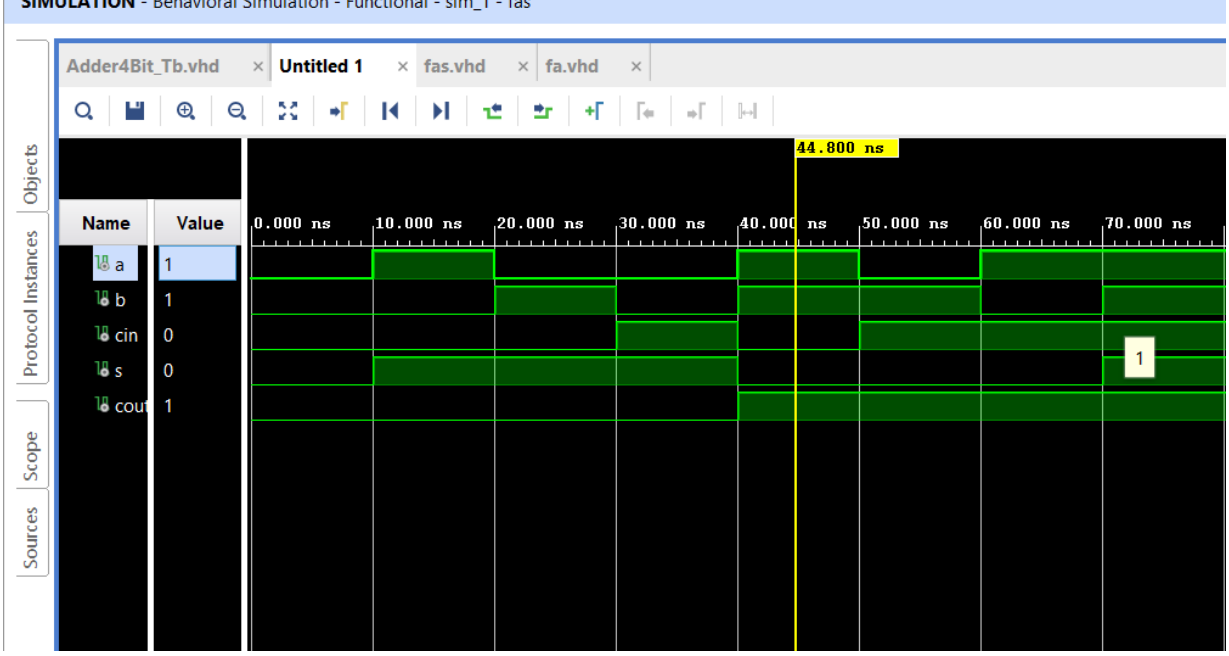
Project Summary | fa.vhd * | bit4Adder.vhd | Adder.vhd *

C:/Users/Ahmad/Desktop/A4BitAdder/A4BitAdder.srcs/sources_1/new/fa.vhd

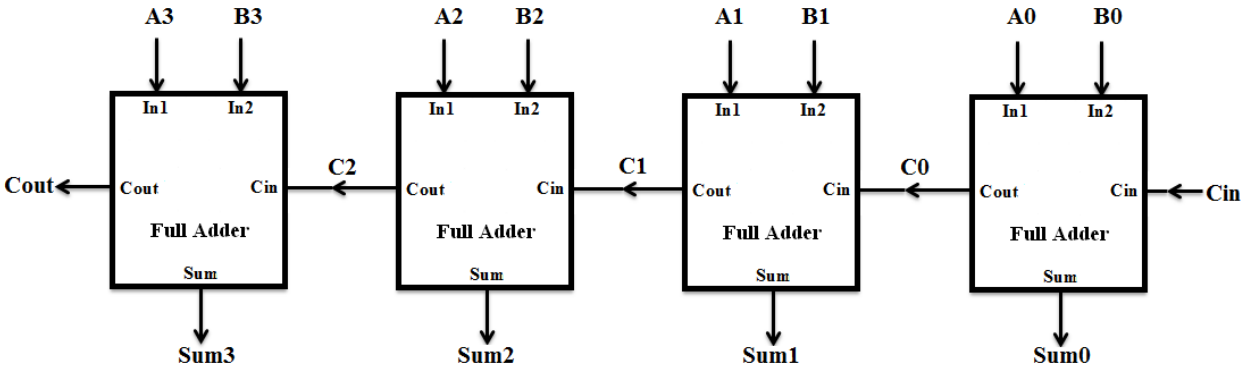
```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity fa is
4      Port ( a : in STD_LOGIC;
5             b : in STD_LOGIC;
6             cin : in STD_LOGIC;
7             s : out STD_LOGIC;
8             cout : out STD_LOGIC);
9  end fa;
10 architecture Behavioral of fa is
11 component adder port (a,b:in std_logic;s,c:out std_logic);
12 end component;
13
14 signal W_sum,w_cry1,w_cry2:std_logic;
15
16 begin
17
18 mod1: adder port map (
19     a=>a,
20     b=>b,
21     s=>w_sum,
22     c=>w_cry1
23 );
24 mod2: adder port map (
25     a=>w_sum,
26     b=>cin,
27     s=>s,
28     c=>w_cry2
29 );
30 cout<= w_cry1 or w_cry2;
31 end Behavioral;
32
```

Simulation and testbench code :

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   entity fas is
4   --  Port ( );
5   end fas;
6   architecture Behavioral of fas is
7
8   component fa port (a,b,cin:in std_logic;s,cout:out std_logic);
9   end component;
10  signal a,b,cin,s,cout:std_logic:='0';
11
12  begin
13  uut: fa port map (
14      a=>a,
15      b=>b,
16      cin=>cin,
17      cout=>cout,
18      s=>s
19  );
20
21  sim:process
22  begin
23
24  a<='0';
25  b<='0';
26  cin<='0';
27  wait for 10 ns;
28
29  a<='1';
30  b<='0';
31  cin<='0';
32  wait for 10 ns;
34  a<='0';
35  b<='1';
36  cin<='0';
37  wait for 10 ns;
38
39  a<='0';
40  b<='0';
41  cin<='1';
42  wait for 10 ns;
43
44  a<='1';
45  b<='1';
46  cin<='0';
47  wait for 10 ns;
48
49  a<='0';
50  b<='1';
51  cin<='1';
52  wait for 10 ns;
53
54  a<='1';
55  b<='0';
56  cin<='1';
57  wait for 10 ns;
58
59  a<='1';
60  b<='1';
61  cin<='1';
62  wait for 10 ns;
63
64  wait;
65  end process;
```



After that i used 4 Full Adder to build the 4 bit Adder




```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity bit4Adder is
5      Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
6             B : in STD_LOGIC_VECTOR (3 downto 0);
7             S : out STD_LOGIC_VECTOR (3 downto 0);
8             Cout : out STD_LOGIC);
9  end bit4Adder;
10 architecture Behavioral of bit4Adder is
11 component Adder port (
12     a:in std_logic;
13     b:in std_logic;
14     cin:in std_logic;
15     s:out std_logic;
16     cout:out std_logic
17 );
18 end component;
19 signal c1,c2,c3,c4:std_logic;
20 begin
21 mod1:Adder port map(
22     a=>A(0),
23     b=>B(0),
24     cin=>'0',
25     s=>S(0),
26     cout=>c1
27 );
28
29 mod2:Adder port map(
30     a=>A(1),
31     b=>B(1),
32     cin=>c1,
33     s=>S(1),
34     cout=>c2
35 );
36
37 mod3:Adder port map(
38     a=>A(2),
39     b=>B(2),
40     cin=>c2,
41     s=>S(2),
42     cout=>c3
43 );
44
45 mod4:Adder port map(
46     a=>A(3),
47     b=>B(3),
48     cin=>c3,
49     s=>S(3),
50     cout=>c4
51 );
52 end Behavioral;
```

After check simulation if running right add constraint file and choose ports in the board .
Synthesis, Implementation, and Bitstream Generation thin program the Zboard and I had checked the circuit .

Inputs	Package Pin	I/O Std
SW0	F22	LVCMOS18
SW1	G22	LVCMOS18
SW2	H22	LVCMOS18
SW3	F21	LVCMOS18
SW4	H19	LVCMOS18
SW5	H18	LVCMOS18
SW6	H17	LVCMOS18
SW7	M15	LVCMOS18
Outputs	**Package Pin**	**I/O Std**
LD0	T22	LVCMOS33
LD1	T21	LVCMOS33
LD2	U22	LVCMOS33
LD3	U21	LVCMOS33
LD4	V22	LVCMOS33

Part two : Behavioral Implementation

Procedure :

Write Behavioral code in vivado as shown :

A4BitAdderB.vhd × A4BitAdder_TB.vhd × Untitled 4 ×

E:/projects/4BitAdderB/4BitAdderB.srcs/sources_1/new/A4BitAdderB.vhd

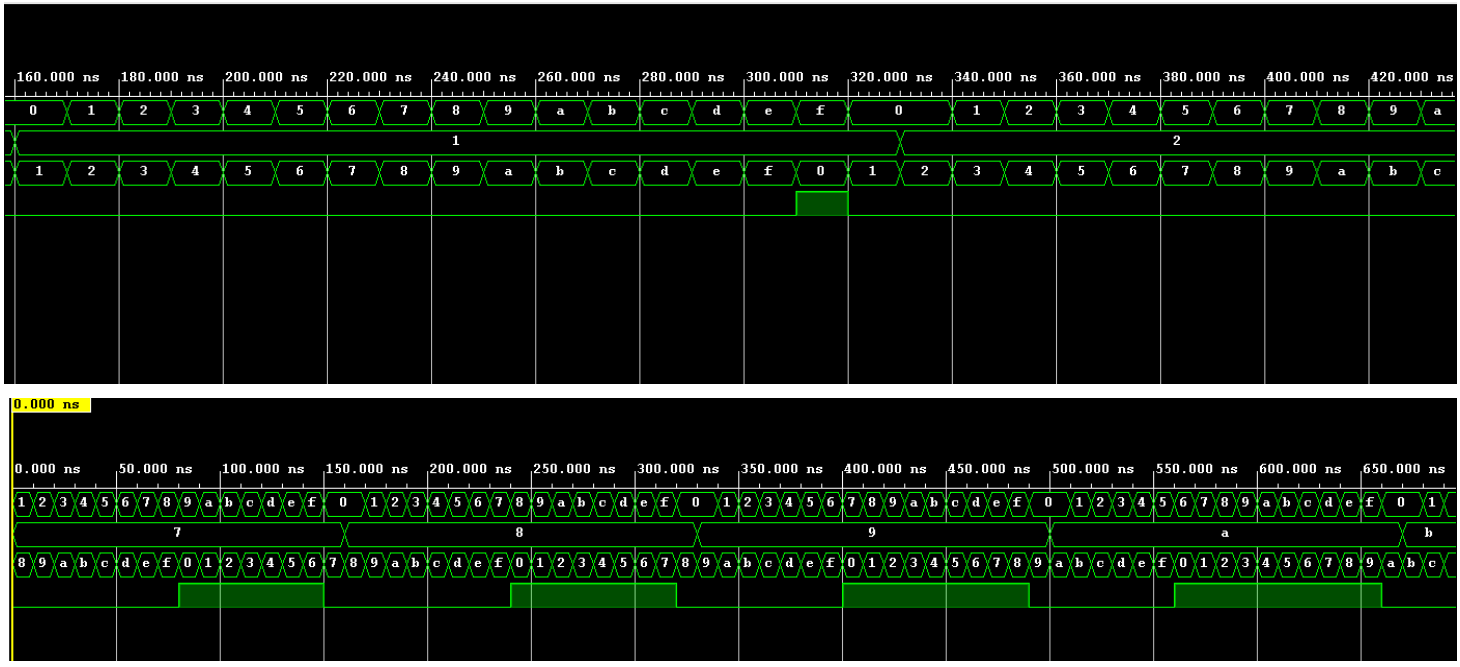
```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3   use IEEE.std_logic_unsigned.ALL ;
4   use IEEE.numeric_std.ALL;
5   entity A4BitAdderB is
6       Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
7              B : in STD_LOGIC_VECTOR (3 downto 0);
8              S : out STD_LOGIC_VECTOR (3 downto 0);
9              Cout : out STD_LOGIC);
10  end A4BitAdderB;
11
12  architecture Behavioral of A4BitAdderB is
13  signal sum:std_logic_vector(4 downto 0);
14  begin
15  sum <= ('0'&A)+('0'&B);
16  S(0)<=sum(0);
17  S(1)<=sum(1);
18  S(2)<=sum(2);
19  S(3)<=sum(3);
20  Cout<=sum(4);
21  end Behavioral;
22
```

Simulation :

There is the testbench code :

```
architecture Behavioral of A4BitAdder_TB is
component A4BitAdderB port (
    A:in std_logic_vector;
    B:in std_logic_vector;
    S:out std_logic_vector;
    Cout:out std_logic
);
end component;
signal A,B,S:std_logic_vector (3 downto 0):="0000";
signal Cout:std_logic:='0';

begin
uut : A4BitAdderB port map (
    A=>A,
    B=>B,
    S=>S,
    Cout=>Cout
);
sim : process
begin
for i in 1 to 16 loop
    for i in 1 to 16 loop
    A<=A+'1';
    wait for 10 ns;
    end loop;
    B<=B+'1';
    wait for 10 ns;
end loop;
end process;
end Behavioral;
```



After Simulate my design . i used the same constraint file from part1
Run the Synthesis, Implementation, and Bitstream Generation processes
Use the Hardware Manger to program your FPGA.
Use the switches and notice the LEDs to verify that your circuit functions correctly.

Conclusion:

In this experiment i learned that i can use switches on the kit as a binary input by specifying in the (.XDC) file. In the same file we can specify the LEDs to show the binary output of our operations.
I got familiar with Zboard kit and i implemented a simple 4BitAdder circuit using VHDL. This helped me learn about the design, simulation and synthesizing processes.