ADVANCED DIGITAL SYSTEMS DESIGN

HW_2_

Prepared By:
*Anas Nimer 1180180

Instructor: Dr. Abdallatif Abuissa

Section: 1

Date: 10/10/2020

Table of Contents

❖ **Procedure**: 3
1- mux 4x1 : 3
- Code 3
- Simulation: 3

2- Full adder: 4
- Code: 4
- Simulation: 4

3- Comparator: 5
- Code: 5
- Simulation: 5

4- All circuit structural: 6
- Code: 6
- Simulation: 6

5- All circuit behavioral: 7
- Code: 7
- Simulation: 7

6- All circuit test: 8
- Code: 8
- Simulation: 8

❖ Truth table **Y_S (output of mux 1)** , key_map and the equation : 9
- **Truth table:** 9
- **key_map:** 10
- **equation that we want is :** 10

❖ Truth table **Y_cout (output of mux 2)** , key_map and the equation : 11
- **Truth table:** 11
- **key_map:** 12
- **equation that we want is :** 12

❖ Procedure:

1- mux 4x1 :

- Code:

```
1
2
3   library ieee;
4   use ieee.std_logic_1164.all;
5   entity mux4x1 is
6       port(D: in std_logic_vector(3 downto 0);
7       S: in std_logic_vector(0 to 1);
8       Y: out std_logic);
9   end;
10
11
12  architecture mux of mux4x1 is
13  begin
14
15      Y<= (D(0) and not S(0) and not S(1))
16      or (D(1) and S(0) and not S(1))
17      or (D(2) and not S(0) and S(1))
18      or (D(3) and S(0) and S(1)) ;
19
20  end ;
21
```

Fig_1_(code of mux)

- Simulation:

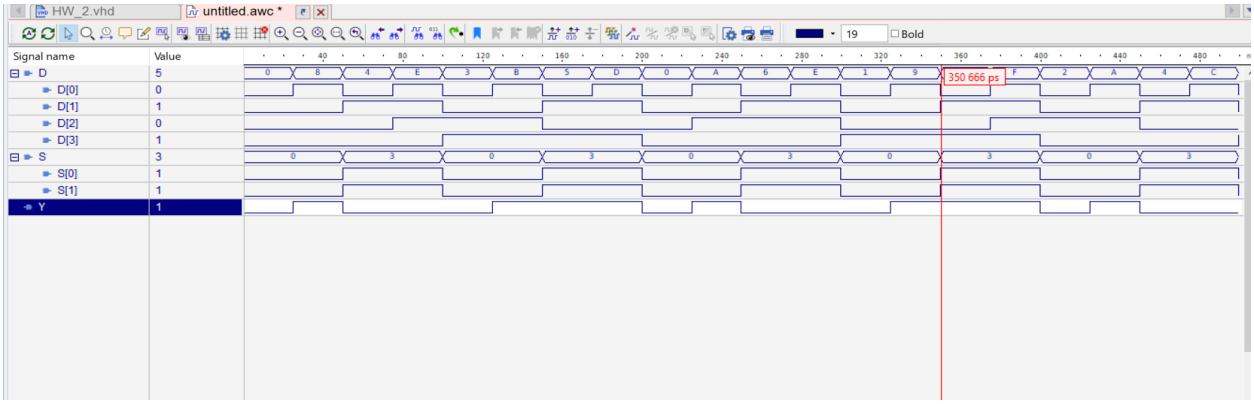

Fig_2_(simulation of decoder)

2- Full adder:

- Code:

```
25  library IEEE;
26  use IEEE.STD_LOGIC_1164.ALL;
27  entity full_adder is
28   Port ( A : in STD_LOGIC;
29   B : in STD_LOGIC;
30   Cin : in STD_LOGIC;
31   S : out STD_LOGIC;
32   Cout : out STD_LOGIC);
33  end full_adder;
34
35  architecture full of full_adder is
36  begin
37
38   S <= A XOR B XOR Cin ;
39   Cout <= (A AND B) OR (Cin AND A) OR (Cin AND B);
40  end;
41
```

Fig_3_(code of Full adder)

- Simulation:

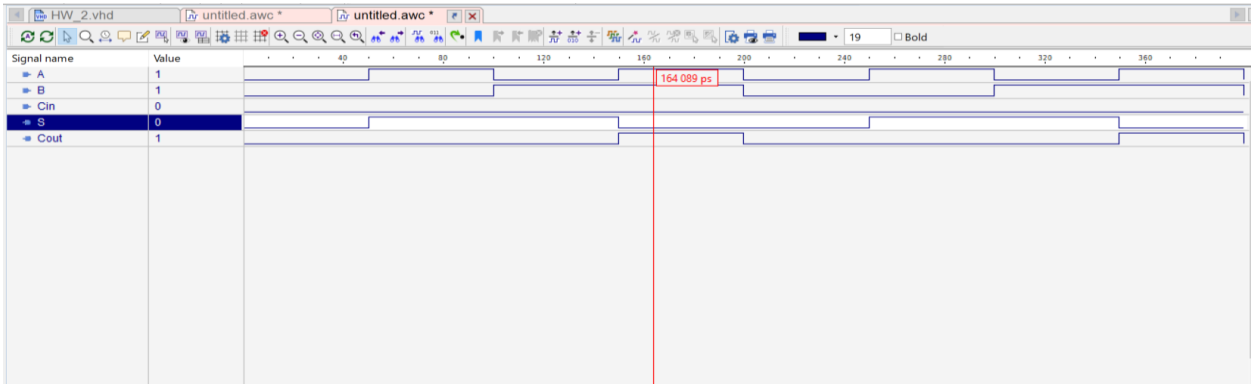

Fig_4_(simulation of Full adder)

3- Comparator:

- Code:

```
43  library IEEE;
44  use IEEE.STD_LOGIC_1164.ALL;
45
46  entity comp is
47      port(A : in STD_LOGIC;
48   B : in STD_LOGIC;
49   Y : out STD_LOGIC);
50  end comp;
51
52  architecture compe of comp is
53  begin
54
55      Y<=(A and not B);
56  end;
57
```

Fig_5_(code of comparator)

- Simulation:

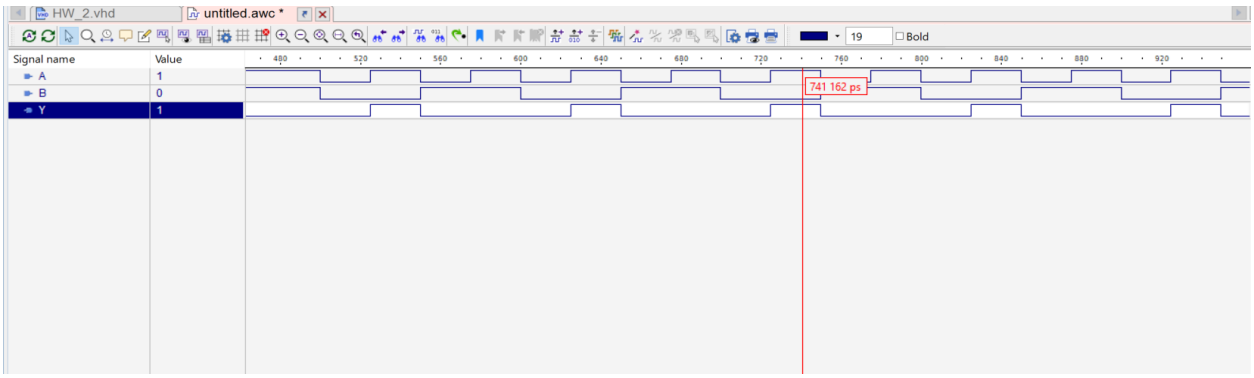

Fig_6_(simulation of comparator)

4- All circuit structural:

- Code:

```
60  library ieee;
61  use ieee.std_logic_1164.all;
62
63  entity allcircuit1 is
64      port(A,B: in std_logic;
65      S: in std_logic_vector(0 to 1);
66      Y_S: out std_logic;
67      Y_Cout: out std_logic);
68  end;
69
70  architecture structural1 of allcircuit1 is
71  signal D1: std_logic_vector(3 downto 0);
72  signal D2: std_logic_vector(3 downto 0);
73
74
75  begin
76      D1(2)<= A and B;
77      D1(3)<= A or B;
78      D1(1)<='0';
79      D2(2)<='0';
80      D2(3)<='0';
81
82      full: entity work.full_adder(full) port map(A,B,'0',D1(0),D2(0));
83      comp: entity work.comp(compe) port map(A,B,D2(1));
84      mux1: entity work.mux4x1(mux) port map(D1,S,Y_S);
85      mux2: entity work.mux4x1(mux) port map(D2,S,Y_Cout);
86  end;
87
```

Fig_7_(code of all circuit structural)

- Simulation:

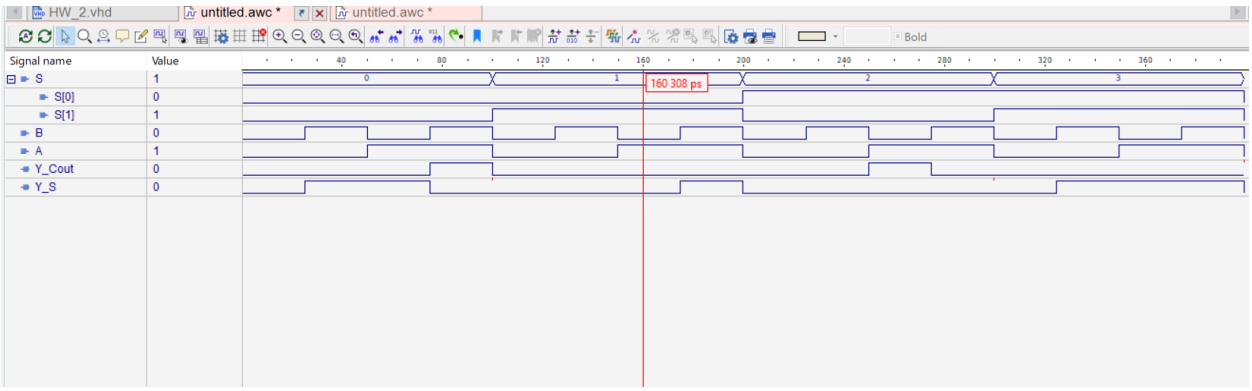

Fig_8_(simulation of all circuit structural)

5- All circuit behavioral:

- Code:

```
architecture behavioral of allcircuit1 is
begin
    Y_S<= (S(1) and A and B) or (S(0) and S(1) and B) or (S(0) and S(1) and A) or (not S(0) and not S(1) and not A and B) or
    (not S(0) and not S(1) and A and not B);
    Y_Cout<= (not S(0) and not S(1) and A and B) or (S(0) and not S(1) and A and not B);

end;
```

Fig_9_(code of all circuit behavioral)

- Simulation:

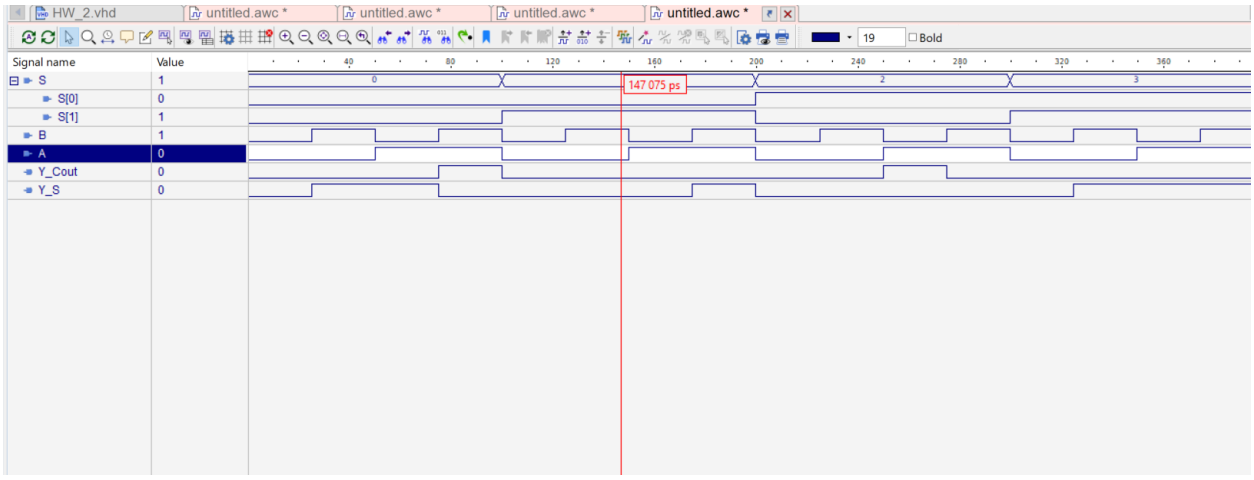

Fig_10_(simulation of all circuit behavioral)

6- All circuit test:

- Code:

```
library ieee;
use ieee.std_logic_1164.all;

entity allcircuit1_Test is
end;

architecture Test of allcircuit1_Test is
signal A,B: std_logic:='0';
signal S: std_logic_vector(0 to 1):= "00";
signal Y_S , Y_Cout: std_logic;

begin


    Y_S<= (S(1) and A and B) or (S(0) and S(1) and B) or (S(0) and S(1) and A) or (not S(0) and not S(1) and not A and B) or
    (not S(0) and not S(1) and A and not B);

    Y_Cout<= (not S(0) and not S(1) and A and B) or (S(0) and not S(1) and A and not B);

    S(0) <= not S(0) after 200 ns;
    S(1) <= not S(1) after 100 ns;
    A<= not A after 50 ns;
    B<= not B after 25 ns;


end;

```

Fig_11_(code of all circuit test)

- Simulation:

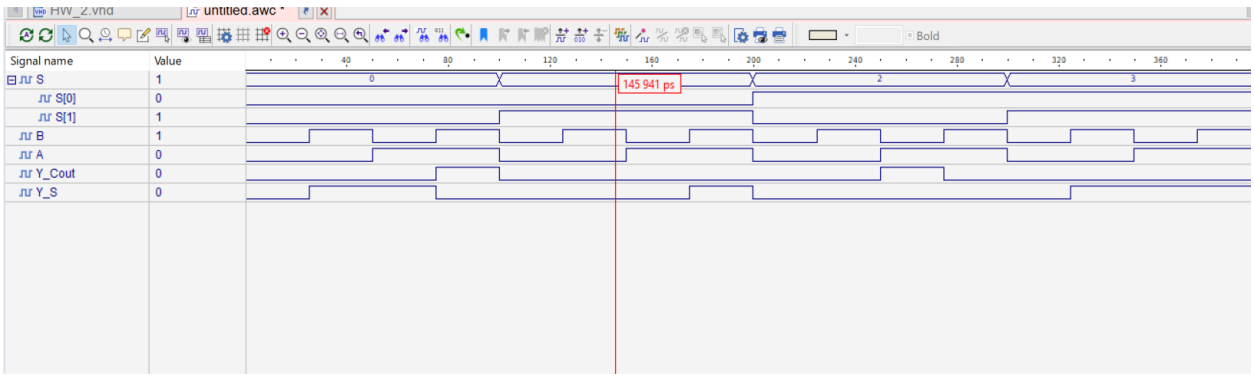

Fig_12_(simulation of all circuit test)

❖ Truth table **Y_S (output of mux 1)** , key_map and the equation :

- **Truth table:**

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

- key_map:

Map

	$\overline{C}.\overline{D}$	$\overline{C}.D$	$C.D$	$C.\overline{D}$
$\overline{A}.\overline{B}$	0	1	0	1
$\overline{A}.B$	0	0	1	0
$A.B$	0	1	1	1
$A.\overline{B}$	0	0	0	0

Map Layout

	$\overline{C}.\overline{D}$	$\overline{C}.D$	$C.D$	$C.\overline{D}$
$\overline{A}.\overline{B}$	0	1	3	2
$\overline{A}.B$	4	5	7	6
$A.B$	12	13	15	14
$A.\overline{B}$	8	9	11	10

- equation that we want is :

Y_S = (S(1) . A . B) + (S(0) . S(1) . B) + (S(0) . S(1) . A) + (S(0)' . S(1)' . A' . B) + (S(0)' . S(1)' . A . B')

❖ Truth table **Y_cout (output of mux 2)** , key_map and the equation :

- **Truth table:**

A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

- key_map:

Map

	$\overline{C}.\overline{D}$	$\overline{C}.D$	$C.D$	$C.\overline{D}$
$\overline{A}.\overline{B}$	0	0	1	0
$\overline{A}.B$	0	0	0	0
$A.B$	0	0	0	0
$A.\overline{B}$	0	0	0	1

Map Layout

	$\overline{C}.\overline{D}$	$\overline{C}.D$	$C.D$	$C.\overline{D}$
$\overline{A}.\overline{B}$	0	1	3	2
$\overline{A}.B$	4	5	7	6
$A.B$	12	13	15	14
$A.\overline{B}$	8	9	11	10

- equation that we want is :

Y_Cout = (S(0)' . S(1)' . A . B) + (S(0) . S(1)' . A . B')