DIGITAL CIRCUITS LAB
VHDL
LAB-8

200020043
Shashi Prabha

Aim: Write VHDL the code satisfying the requirements given in the questions i.e

- 3-bit adder/subtractor in structural modeling style
- VHDL code for ALU
- VHDL code for 4:2 priority encoder with active high enable pin

Q1)***Write VHDL code for 3-bit adder/subtractor in structural modeling style.***

In this we wrote VHDL code according to these condition

m	operation
0	a+b
1	a-b (2's complement form)

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

--Define input and output ports–
entity bit3_full_adder is
port(i1, i2, i3: in bit;
o1, o2 : out bit);
end bit3_full_adder;
```

```
--Defining the architecture of full adder in dataflow modelling style—-
architecture b_fa of bit3_full_adder is
Begin
o1 <= i1 xor i2 xor i3;
o2 <= (i1 and i2) or ((i1 xor i2) and i3);
end b_fa;


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

  --define the entity with ports
entity TBA is
port (a : in STD_LOGIC_VECTOR (2 downto 0);
b : in STD_LOGIC_VECTOR (2 downto 0);
m : in STD_LOGIC;
sum :out STD_LOGIC_VECTOR (2 downto 0);
cout : out STD_LOGIC);
end TBA;

-----DEFINE THE ARCHITECTURE FOR ADDER_SUBTRACTOR
architecture rtl of TBA is
--DEFINE THE COMPONENT FULL ADDER USED-----
component bit3_full_adder is
port(i1, i2, i3: in STD_LOGIC;
o1, o2 : out STD_LOGIC);
end component;

--DEFINE THE INTERMEDIATE SIGNALS IF REQUIRED----
signal C1, C2, C3: STD_LOGIC;
signal TEMP: std_logic_vector(2 downto 0);

-----DEFINE THE FUNCTIONALITY WITH STRUCTURAL MODELING
begin
```

```
TEMP <= A xor B;
FA0:bit3_full_adder port map(A(0),TEMP(0),m, sum(0),C1);
FA1:bit3_full_adder port map(A(1),TEMP(1),C1, sum(1),C2);
FA2:bit3_full_adder port map(A(2),TEMP(2),C2, sum(2),C3);
Cout <= C3;
end rtl;
```

Q2) ***Write VHDL code for ALU which performs following operation depending on selection lines***

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use ieee.NUMERIC_STD.all;

  --define the entity with ports
entity ALU is
Port (
         A, B : in STD_LOGIC_VECTOR(2 downto 0);        -- 2 inputs 8-bit
        ALU_Sel : in STD_LOGIC_VECTOR(1 downto 0);
       ALU_Out : out STD_LOGIC_VECTOR(3 downto 0);
        Carryout : out std_logic
);

--DEFINE THE INTERMEDIATE SIGNALS IF REQUIRED----
signal ALU_Result: std_logic_vector(2 downto 0);
signal tmp : std_logic_vector(3 downto 0);

end ALU;

-----DEFINE THE FUNCTIONALITY WITH BEHAVIORAL MODELING
architecture Behavioral of ALU is
begin
```

```
process(A,B,ALU_Sel)
begin
        case(ALU_Sel) is
        when "00" => --Addition
        ALU_Result <= A+B;
        when "01" => --Subtraction
        ALU_Result <= A-B;
        when "10" => --A bitwise and B
        ALU_Result <= (A and B);
        when "11" => --A bitwise and B
        ALU_Result <= (A xor B);
end case;
end process;
        ALU_Out (2 downto 0)<=ALU_Result;--ALU_Out
        tmp <= ('0'&A) + ('0'&B);
        ALU_Out(3) <= tmp(3);
end behavioral;
```

Q3) ***Write VHDL code for 4:2 priority encoder with active high enable pin***

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

 --define the entity with ports
entity pri_enco is
port(
D : in STD_LOGIC_VECTOR(3 downto 0);
X : out STD_LOGIC_VECTOR(1 downto 0);
V : in STD_LOGIC
);
end pri_enco;
-----DEFINE THE FUNCTIONALITY WITH BEHAVIORAL MODELING
```

```
architecture behavioral of pri_enco is
BEGIN
        process(V)
BEGIN
        if (V='1')
                then
                X(0)<= (D(2) or D(3)) and (not D(0)) and (not D(1)) ;
                X(1)<= (D(1) or (D(3) and (not D(2)))) and (not D(0)) ;
                else X(0)<= '0';
                X(1)<= '0';
end if;
end process;
end behavioral;
```

Conclusion: This lab was very interesting. We learned how to write code in VHDL for 3-bit adder/subtractor in structural modeling style,VHDL code for ALU,VHDL code for 4:2 priority encoder with active high enable pin . We also saw their output in RTL viewer.
We even cross checked our codes by running them using hardware.