ASSIGNMENT 4

COMPUTER ORGANISATION AND ARCHITECTURE LAB

(IT 2272)

ADITYA BADAYALA

510819056

IT(HY)

QUESTION 1:

DESIGN AND SIMULATE 1 BIT FA USING IF-ELSE CONSTRUCT

VHDL TESTBENCH:

```
-- Testbench for FULL ADDER
library IEEE;
use IEEE.std_logic_1164.all;

entity FULLADDER is
-- empty
end FULLADDER;

architecture tb of FULLADDER is

-- DUT component
component full_adder is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  carry: out std_logic;
  sum: out std_logic);
end component;

signal a_in, b_in, c_in, carry_out, sum_out: std_logic;

begin

  -- Connect DUT
  DUT: full_adder port map(a_in, b_in, c_in, carry_out,
sum_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
```

```
    assert(carry_out='0' and sum_out='0') report "Fail 0/0/0"
severity error;
    a_in <= '0';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 0/0/1"
severity error;
    a_in <= '0';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 0/1/0"
severity error;
    a_in <= '0';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 0/1/1"
severity error;
    a_in <= '1';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 1/0/0"
severity error;
    a_in <= '1';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 1/0/1"
severity error;
    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 1/1/0"
severity error;
    a_in <= '1';
    b_in <= '1';
```

```
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='1') report "Fail 1/1/1"
severity error;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    assert false report "Test done." severity note;
    wait;
  end process;
end tb;
```

VHDL MODULE:

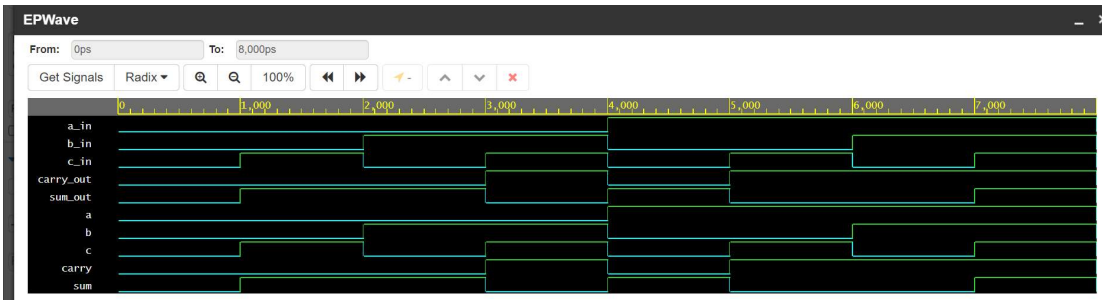
```
--FULL ADDER design
library IEEE;
use IEEE.std_logic_1164.all;

entity full_adder is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  carry: out std_logic;
  sum: out std_logic);
end full_adder;

architecture behaviour of full_adder is
begin
  process(a, b, c) is
  begin
     if (((a and b and c)<='0') and (((a and b)) or ((c and b))
or ((a and c)))='1') then
    sum <= '0';
    carry <= '1';
    elsif ((a and b and c) ='1') then
    sum<='1';
```

```
    carry <='1';
    elsif ((a or b or c)='0') then
    sum<='0';
    carry<='0';
    else
    sum <='1';
    carry <='0';
    end if;
  end process;
end behaviour;
```

SIMULATION:



QUESTION 2:

DESIGN AND SIMULATE 1 BIT FA USING CASE CONSTRUCT

VHDL TESTBENCH:

```
-- Testbench for FULL ADDER
library IEEE;
use IEEE.std_logic_1164.all;

entity FULLADDER is
-- empty
end FULLADDER;

architecture tb of FULLADDER is

-- DUT component
component full_adder is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  carry: out std_logic;
  sum: out std_logic);
end component;

signal a_in, b_in, c_in, carry_out, sum_out: std_logic;

begin

  -- Connect DUT
  DUT: full_adder port map(a_in, b_in, c_in, carry_out,
sum_out);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
```

```
    assert(carry_out='0' and sum_out='0') report "Fail 0/0/0"
severity error;
    a_in <= '0';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 0/0/1"
severity error;
    a_in <= '0';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 0/1/0"
severity error;
    a_in <= '0';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 0/1/1"
severity error;
    a_in <= '1';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='0' and sum_out='1') report "Fail 1/0/0"
severity error;
    a_in <= '1';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 1/0/1"
severity error;
    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='0') report "Fail 1/1/0"
severity error;
    a_in <= '1';
    b_in <= '1';
```

```
    c_in <= '1';
    wait for 1 ns;
    assert(carry_out='1' and sum_out='1') report "Fail 1/1/1"
severity error;

    -- Clear inputs
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    assert false report "Test done." severity note;
    wait;
  end process;
end tb;
```

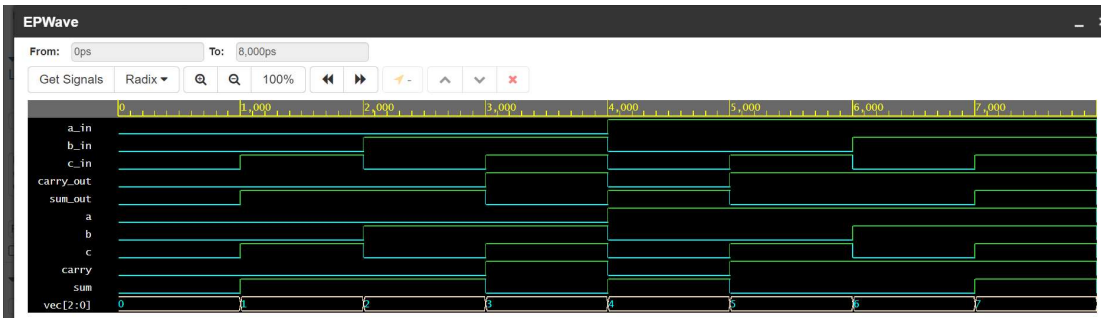
VHDL MODULE:

```
--FULL ADDER design
library IEEE;
use IEEE.std_logic_1164.all;
entity full_adder is
port(
 a: in std_logic;
 b: in std_logic;
 c: in std_logic;
 carry: out std_logic;
 sum: out std_logic);
end full_adder;

architecture behaviour of full_adder is
signal vec : std_logic_vector(2 downto 0);
begin
 process(vec,a, b, c) is
 begin
     vec<=a & b & c;
     case vec is
     when "000"=>
          sum<='0';
            carry<='0';
        when "001"|"010"|"100"=>
          sum<='1';
```

```
            carry<='0';
        when "111"=>
            sum<='1';
            carry<='1';
        when others=>
            sum<='0';
            carry<='1';
        end case;
 end process;
end behaviour;
```

SIMULATION:



QUESTION 3:

DESIGN A 3 TO 8 DECODER USING CASE CONSTRUCT

VHDL TESTBENCH:

```
-- Testbench for 3 to 8 decoder
library IEEE;
use IEEE.std_logic_1164.all;

entity DECODER is
-- empty
end DECODER;

architecture tb of DECODER is

-- DUT component
component decoder_38 is
port(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  d0,d1,d2,d3,d4,d5,d6,d7: out std_logic;);
end component;

signal a_in, b_in, c_in, D0,D1,D2,D3,D4,D5,D6,D7: std_logic;

begin

  -- Connect DUT
  DUT: decoder_38 port map(a_in, b_in, c_in,
D0,D1,D2,D3,D4,D5,D6,D7);

  process
  begin
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
    a_in <= '0';
```

```
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    a_in <= '0';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    a_in <= '0';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;
    a_in <= '1';
    b_in <= '0';
    c_in <= '0';
    wait for 1 ns;
    a_in <= '1';
    b_in <= '0';
    c_in <= '1';
    wait for 1 ns;
    a_in <= '1';
    b_in <= '1';
    c_in <= '0';
    wait for 1 ns;
    a_in <= '1';
    b_in <= '1';
    c_in <= '1';
    wait for 1 ns;
    a_in <= '0';
    b_in <= '0';
    c_in <= '0';
    wait;
  end process;
end tb;
```

VHDL MODULE:

```
--3 to 8 decoder design
library IEEE;
use IEEE.std_logic_1164.all;
entity decoder_38 is
port(
 a: in std_logic;
 b: in std_logic;
 c: in std_logic;
 d0,d1,d2,d3,d4,d5,d6,d7: out std_logic;);
end decoder_38;

architecture behaviour of decoder_38 is
signal vec : std_logic_vector(2 downto 0);
begin
 process(vec,a, b, c) is
 begin
     vec<=a & b & c;
     case vec is
     when "000"=>
          d0<='1';
            d1<='0';
            d2<='0';
            d3<='0';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='0';
        when "001"=>
          d0<='0';
            d1<='1';
            d2<='0';
            d3<='0';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='0';
        when "010"=>
          d0<='0';
```

```
            d1<='0';
            d2<='1';
            d3<='0';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='0';
        when "011"=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='1';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='0';
        when "100"=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='0';
            d4<='1';
            d5<='0';
            d6<='0';
            d7<='0';
        when "101"=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='0';
            d4<='0';
            d5<='1';
            d6<='0';
            d7<='0';
        when "110"=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='0';
```

```
            d4<='0';
            d5<='0';
            d6<='1';
            d7<='0';
        when "111"=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='0';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='1';
        when others=>
          d0<='0';
            d1<='0';
            d2<='0';
            d3<='0';
            d4<='0';
            d5<='0';
            d6<='0';
            d7<='0';
        end case;
 end process;
end behaviour;
```

SIMULATION: