ASSIGNMENT 3

COMPUTER ORGANISATION AND ARCHITECTURE LAB

(IT 2272)

ADITYA BADAYALA

510819056

IT(HY)

QUESTION 1:

DESIGN A STRUCTURAL MODEL OF A FULL ADDER USING HALF ADDER AS A COMPONENT

VHDL MODULE:

```
library IEEE;
use IEEE.std_logic_1164.all;

entity full_adder is
port(in1, in2, c_in: in std_logic;
     sum, c_out: out std_logic);
end entity full_adder;

architecture structural of full_adder is
component half_adder is
port(x,y : in std_logic;
     z : out std_logic);
end component half_adder;

component or_2 is
port (x,y : in std_logic;
      z: out std_logic);
end component or_2;

signal s1, s2, s3 : std_logic;
begin
H1: half_adder port map(x=>in1, y=>in2, sum=>s1, carry=>s3);
H2: half_adder port map(x=>s1, y=>c_in, sum=>sum, carry=>s2);
O1: or_2 port map(x=>s2, y=>s3, z=>c_out);
end architecture structural;
```

QUESTION 2:

DESIGN A 4-BIT BPA USING STRUCTURAL DESIGN FLOW HIERARCHIALLY

VHDL MODULE:

```
library IEEE;
library libs;
use IEEE.std_logic_1164.all;
use Libs.VHDLPrims.all;
use Libs.mylib.all;

entity add4par is
port (c0,a,b: in std_logic;
      c4,sum: out std_logic);
end add4par;

architecture arch1 of add4par is
     component full_add
          port(a,b,c_in : in std_logic;
               c_out,sum: out std_logic);
     end component;

signal c: std_logic_vector(3 downto 1);

begin

adder1: full_add
     port map(a=>a(1),b=b(1),c_in=>c0,c_out=>c(1),sum=>sum(1));
adder2: full_add
     port map(a=>a(2),b=b(2),
                c_in=>c(1),c_out=>c(2),sum=>sum(2));
adder3: full_add
     port map(a=>a(3),b=b(3),
                c_in=>c(2),c_out=>c(3),sum=>sum(3));
adder2: full_add
     port map(a=>a(4),b=b(4),
                c_in=>c(3),c_out=>c(4),sum=>sum(4));
end arch1;
```

QUESTION 3:

DESIGN A HALF ADDER USING IF ELSE STRUCTURE

VHDL TESTBENCH:

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

entity HALFADDER is
-- empty
end HALFADDER;

architecture tb of HALFADDER is

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

signal a_in, b_in, carry_out, sum_out: std_logic;

begin

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

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

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

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

VHDL MODULE:

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

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

architecture behaviour of half_adder is
begin
  process(a, b) is
  begin
```

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

SIMULATION: