EE214: Homework (Thursday Batch)

Wadhwani Electronics Lab, IIT Bombay

18th February 2021

Please complete the following assignments before your next lab turn, as these designs will be used in next in-Lab experiment.

1. (a) Review Behavioural Architecture Description in VHDL including conditional signal assignment, select signal assignment, process block, if-then-else and for loop.

   (b) Watch Prof. Dinesh Sharma's lecture on Behavioural Description

   (c) Complete the following description of **1-bit Arithmetic Right shifter** in VHDL using conditional signal assignment. (Assume input X to be 8-bit signed number. When S = '1', then Y should be 1-bit right shifted version of X, while preserving sign bit. When S = '0', Y should be same as X)

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

entity ShiftRightByOne is
    port (X: in std_logic_vector(7 downto 0);
          S: in std_logic;
          Y: out std_logic_vector(7 downto 0)
          );
end entity ShiftRightByOne;

architecture Easy of ShiftRightByOne is
begin
Y(7) <= ___ when ____ else ____;
.......
.......
.......
end Easy;
```

   (d) Simulate the above design in Modelsim and validate its functionality using the given Tracefile.
   NOTE: TRACEFILE format
   Input{S X7 X6 X5 X4 X3 X2 X1 X0} Output{Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0} MASK{11111111}

2. (a) Complete the following description of **8-bit Subtractor** in VHDL using process block (Assume inputs A and B as 8-bit unsigned numbers only).
   *Final result is expected to be in 2s complement form
   *The approach in pseudo code given below is quite different/tricky from standard approach, be careful and refer to the TRACEFILE for better understanding.

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

entity sub_8bit is
    port (
        A: in std_logic_vector(7 downto 0);
        B: in std_logic_vector(7 downto 0);
        diff: out std_logic_vector(7 downto 0);
        b_out : out std_logic
    ) ;
end sub_8bit;

architecture Beh of sub_8bit is

begin
    sub : process( A, B )
    variable diff_var, B_comp : std_logic_vector(7 downto 0):= (others=>'0');
    variable carry_prop : std_logic_vector(8 downto 0):= (0 => '1', others=>'0');
    begin
        B_comp := not B;
```

```
            for i in ___________ loop
                ...........
                ...........
            end loop;
            diff <= diff_var;
            b_out <= not carry_prop(8);
        end process ; -- sub
    end Beh ; -- Beh
```

(b) Simulate the above design in Modelsim and validate its functionality using the given Tracefile.
NOTE: TRACEFILE format
Input{A7 A6 A5 A4 A3 A2 A1 A0 B7 B6 B5 B4 B3 B2 B1 B0} Output{b_out diff7 diff6 diff5 diff4 diff3 diff2 diff1 diff0} MASK{111111111}