Homework 1

1. Translate the following code to a VHDL if-then-else statement:

```
transmit <= signal_a WHEN state = idle ELSE
            signal_b WHEN state = incoming ELSE
            signal_c WHEN state = outgoing ELSE
            signal_d;
```

2. Translate the following code to a VHDL when-else statement:

```
PROCESS (a, b, j, k)
BEGIN
    IF a = ‘1’ AND b = ‘0’ THEN
        step <= “0100”;
    ELSIF a = ‘1’ THEN
        step <= j;
    ELSIF b = ‘1’ THEN
        step <= k;
    ELSE
        step <= “ --- ”;
    END IF;
END PROCESS;
```

3. Translate the following code to a VHDL case-when statement:

```
WITH state select
  data <= “0000” WHEN idle | terminate,
          “1111” WHEN increase,
          “1010” WHEN maintain,
          “0101” WHEN decrease,
          “----” WHEN OTHERS;
```

4. Translate the following code to two VHDL with-select-when statements:

```
PROCESS (state)
BEGIN
  CASE state IS
    WHEN idle => a <= “11”; b <= “00”;
    WHEN terminate | increase => a <= “01”; b <= “--";
    WHEN maintain | decrease => a <= “10”; b <= “11”;
    WHEN OTHERS => a<= “11”; b <= “01”;
  END CASE;
END PROECSS;
```

5. Write the VHDL entity and architecture for a VHDL model that compares two 8-bit buses, **a** and **b**. The output signal **a_gr_b** is true only when input **a** is greater than input b. Submit source code and a test bench. The test bench does not need to be self-checking.

6. Shown below is the VHDL model and implementation equation for a very simple combinational logic design.

    a. Re-order the three signal assignment statements so their order is STATEMENT #3, STATEMENT #2, then STATEMENT #1 and re-implement the design. What effect did that change have on the implementation equation?

    b. Repeat step (a) with the order STATEMENT #2, STATEMENT #1, and then STATEMENT #3.

    c. Based on your results, what can you say about the order of these statements? Why is this true? Submit the source code and 3 sets of equations from fitter report.

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

entity hw2_3 is
    port (
        a: in STD_LOGIC;
        b: in STD_LOGIC;
        c: in STD_LOGIC;
        z: out STD_LOGIC
    );
end hw2_3;

architecture hw2_3_arch of hw2_3 is
  signal temp1, temp2 : STD_LOGIC;
begin
  temp1 <= a and b;           -- STATEMENT #1
  temp2 <= not c;             -- STATEMENT #2
  z <= temp1 or temp2; -- STATEMENT #3
end hw2_3_arch;


;------------------------------------------------------;
; Implemented Equations.

 z = /c + a * b
```

Where to find equations:

Go to **Tools** --> **Options**
Click **Processing**
Click the checkbox for **automatically generate equations files during compilation**
Click **OK**
Compile the design
When completed, expand **Fitter** option under ***Table of Contents*** for the ***Flow Summary***
Click **Equations** to display the report.