

EE-313 LAB #4
BASIC AND COMBINATIONAL LOGIC GATES

1. Purpose: Review Altera Quartus II Design Software to simulate the truth tables for combinational logic using block diagrams and a hardware description language (VHDL). Review implementation of logic design in a field-programmable logic array. Introduce implementation of combination logic using NAND gate equivalent. Introduce entering a truth table in VHDL using a vector signal.

2. Equipment: Desktop with Altera Software
Altera PLD Board
Altera Quartus II Quick Reference

3. Procedure
 - a. You will be designing a basic logic gate circuit that monitors the RPM, Pressure, Oil Level, and Temperature values of an engine using sensors that operate as follows:
 - Pressure (P) sensor output = 0 when pressure < 300 PSI
 - RPM (R) sensor output = 0 when the speed < 5200 RPM
 - Oil Level (L) sensor = 0 when level is > 75%
 - Temperature (T) sensor output = 0 when temperature < 300° F

 - b. Using block diagrams design, compile and simulate a system that will illuminate a Warning (W) light under the following set of circumstances:
 - If the engine pressure exceeds 300 PSI when the temperature is above 300°, **or**
 - When the pressure is over 300 PSI while the engine is at high RPM (above 5200 RPM) and the oil level is low (below 75%), **or**
 - When the temperature is over 300° when the oil level is low.

Note: The engine's normal operating conditions are P=R=L=T=0. Fill in Table 1 for the system you design.

 - c. Change the logic circuit that you designed above into a NAND gate equivalent circuit (i.e. use ONLY NAND gates to implement the logical combination that you derived above. Compile and simulate your NAND gate equivalent circuit.

 - d. Complete the table below:

 - e. Using VHDL, enter the truth table above using a vector signal, compile, simulate the combinational logic, and implement your VHDL design into the Altera PLD board.

 - f. Using toggle switches and LEDs on the Altera PLD board, demonstrate to the instructor that your VHDL design on the Altera PLD board provides the same outputs as table 1.

Inputs				Output W1 (Predicted)	Output W1 (Simulated)	Output W1 (Simulated NAND gates only)
P	R	L	T			
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Table 1: Truth Table for your Logic Circuit

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY fig5_62 IS
    PORT(
        a,b,c      : IN std_logic;
        x          : OUT std_logic);
END fig5_62;

ARCHITECTURE arc OF fig5_62 IS
    SIGNAL input : std_logic_vector(2 DOWNTO 0);
BEGIN
    input(2)<=a; --move a to element 2 of the internal vector signal
    input(1)<=b; --move b to element 1 of the internal vector signal
    input(0)<=c; --move c to element 0 of the internal vector signal
    WITH input SELECT
        x <= '1' WHEN "000", -- x equals 1 when input equals "000"
             '0' WHEN "001", -- x equals 0 when input equals "001"
             '1' WHEN "010", -- x equals 1 when input equals "010"
             '0' WHEN "011", -- x equals 0 when input equals "011"
             '1' WHEN "100", -- x equals 1 when input equals "100"
             '1' WHEN "101", -- x equals 1 when input equals "101"
             '1' WHEN "110", -- x equals 1 when input equals "110"
             '0' WHEN "111", -- x equals 0 when input equals "111"
             '1' WHEN others;
END arc;

```

*Note this is a sample of VHDL code using a truth table not the actual code for this lab.

4. Deliverables

- a. Create an entry in your lab notebook as per the course guidelines, with a copy of your combinational block diagram logic and its simulation, your NAND block diagram and its simulation, and a copy of your VHDL code for said combinational logic and its simulation. Your lab notebook should have an emphasis on lessons learned.