CPE 322 **EXAM II**
Spring Semester 2020

INSTRUCTIONS: **You may use any resources available including class notes, textbooks, internet and research articles but you are not to employ any outside help from any individual. This should be your own work. Students who violate this rule will be considered to be guilty of academic misconduct as defined in the UAH Student Handbook. Exam due date: Tuesday April 22, 2020, on-line submission via Canvas.**

Exam Problem

(a) Write a parameterized Verilog module named *ones_count.v* which has an input port declarations for the bus named *in_vec*, and an output port declaration for the bus named *count*. Assume that the number of bits in *in_vec* is given by the the module's Verilog *parameter*, N, which is assigned a default value of 16. Also assume that the number of bits in *count* is derived from N at compile time and is ceil($\log_2$(N)). The module should be combinational in nature employing no clocking signal. It should be encoded in a behavioral manner that incorporates an *always* secton that contains a single *for* loop.

(b) Write a Verilog testbench, named *ones_cout_tb.v* that will provide a new stimulus to the *in_vec* input of the *ones_count* module every 10 ns for the 16 distinct *in_vec* values that are read from the file *in_vec_input.txt* which is a 16 line text file where the data is encoded in hexadecimal format. The Verilog testbench should also record the resulting count value that is produces by the ones_count module for each *in_vec* input and place it in the text output file named *count_output*.txt. The *count* data in this text file should be in decimal format with each *count* value being shown on a separate line.

(c) Simulate this design using ModelSIM^{tm} using the testbench developed in Part (b). Before doing this set the value of N to 32 in your testbench (overriding its default of 16) when you instantiate the *ones_count* module. Also enter 16 stimulus data points in the *in_vec_input.txt* files in hexadecimal format. Chose values of test vectors that should provide a wide range of *count* values in a correctly operating design. Verify the correctness of the results in the *count_output.txt* file.

(d) Create a modified version of the *ones_count.v* count, named *ones_count_task.v* that utilizes a task construct and a call to the task. The task call should have the following format: ***task_name(A,B)*** where ***A*** is the input and ***B*** is the output. Verify the correctness of the design through simulation using the testbench developed in Part (b) and the same input stimulus in the *in_vec_input.txt* file.

(e) Create another modified version of the *ones_count.v*, named *ones_count_function.v* that utilizes a function construct and a call to that function within the *ones_count_function.v* module. Verify the correctness of the design through simulation using the testbench developed in Part (b) and the same input stimulus in the *in_vec_input.txt* file.

Submissions of this assignment should be made on Canvas. It should include the fully commented the *ones_count.v*, *ones_count_tb.v*, *ones_count_task.v*, *ones_count_function.v* and the resulting *count_output.txt* files from part c, d, and e.