COE 405, Term 041

COE 561 Digital System Design and Synthesis

HW# 1

Due date: Sunday, Oct. 10

Q.1. It is required to model an **n-bit up-down counter**. The counter has a select input, ***sel*** that selects the direction of counting. If *sel=0*, the counter counts up otherwise it counts down. Assume that when the counter reaches the maximum value then it will go to 0. Also, assume that the counter is *rising-edge triggered* and has an *synchronous reset*. Use type **bit** and **bit_vector** for the signals used in your models.

(i) Describe an Entity **UDCOUNT** in VHDL for this n-bit counter. Assume that the size of the counter is specified as a generic parameter.

(ii) Write in VHDL an Architecture **Behave** for modeling this n-bit up/down counter at the behavioral level using a process. Simulate the VHDL model **Behave** and verify that it is working properly. Include a snapshot of the simulated waveform.

(iii) Write in VHDL an Architecture **DataFlow** for modeling this n-bit up/down counter at the data flow level using block and concurrent statements. Simulate the VHDL model **DataFlow** and verify that it is working properly. Include a snapshot of the simulated waveform.

(iv) Write in VHDL an Architecture **Structural** for modeling this n-bit up/down counter at the structural level. You can model the basic components (i.e. FF, MUX, AND, OR, etc) that you will be using at any level you like (i.e. behavioral, data flow, or structural). Simulate the VHDL model **Structural** and verify that it is working properly. Include a snapshot of the simulated waveform.

Q.2. It is required to model a sequence detector that detects the sequence 10111 assuming overlapping sequence detection.

(i) Design the FSM for detecting this sequence assuming overlapping sequences.

(ii) Model your FSM design in VHDL using a behavioral model assuming falling-edge triggering and asynchronous reset.

(iii) Model a test bench that tests the behavior of the sequence detector generating the CLK input with a clock period of 100ns with 50% duty cycle, and the input sequence 00100101110111100010111111011110000.