UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

Digital System Design with Hardware Description Languages

ECE 247 – Fall 1395

Today's Date:

First Name: ....

Last Name: ....

Student Id Number: ....

Signature: ....

Grade:

Problem 1. ______/15

Problem 2. ______/15

Problem 3. ______/20

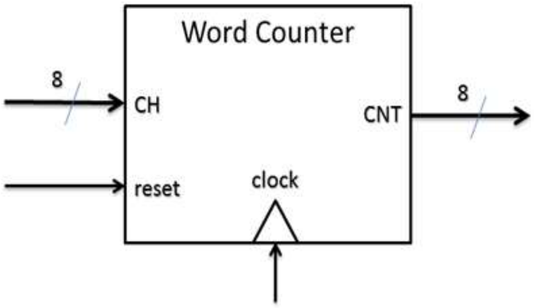
Problem 4. ______/50

Total: ______/100

Khosro Abad, **Sanandaj, Iran**

DO NOT USE LAPTOPS
EXTRA SHEETS WILL NOT BE ACCEPTED
ALL WORKS MUST BE REASONABLY OPTIMIZED
THIS IS AN OPEN BOOK, OPEN NOTE EXAM
YOU MUST SHOW COMPLETE WORK ON ALL PROBLEMS
YOU HAVE EXACTLY 160 MINUTES FOR WORKING ON THIS TEST

1. Write RTL SystemC description of a Word Counter. Characters of an input text appear sequentially on the 8-bit port (CH) of Word Counter. The circuit has an 8 bit outputs that counts number of words that it finds on its CH port after being reset. The words in the text are separated by unknown number of space characters, and the length of each word is unknown. ASCII code of the space character is 32. Show the complete schematic of the circuit. Write RT level SystemC description of this circuit including its datapath and controller.



2. A counter that counts the following sequence in both directions is to be designed. When the *r* input is 1, the counter counts in the right direction, and when 0 it counts in the opposite direction. The counter has a resetting mechanism that starts it in the 100 state. Show RT level description of this counter in SystemC.

100 → 101 → 010 → 110 → 001 → repeats

3. Sayeh processor uses a combinational array multiplier, but has no divider. We have a sequential 16-bit divider that is to be interfaced with the ALU of Sayeh. **Divider**: The divider takes two 16-bit operands and after a complete pulse on its *start* input it start the sequential division process. The division results (Quotient and Remainder) will become available in 16-bit Q and R registers along with a one clock pulse on the divider's *go* output. **Sayeh**: A new DIV Rs, Rd instruction is to be added to Sayeh. This instruction divides Rs ÷ Rd and puts the quotient and remainder in Rs and Rd respectively. **Design**: Show necessary datapath and controller modifications that need to be made to the Sayeh processor. Show all datapath registers, logic units and the necessary control signals. No VHDL is necessary for the datapath. Show controller state changes in VHDL required to make the necessary changes.

4. You are to design a channel with *n* processing requestors (PRs) and *m* special purpose processors (SPs) each of which is responsible for a specific task. The type of the data and the processing that is to be done is packaged in a class of Trans type. This class has an array of 128 integers, an integer identifying the task that is to be done, and a flag indicating the status of the data (New or Completed). **PRs:** A PR places a package in the channel to be processed. The data filed includes the data, the task field indicates the task to be performed on data, and the status field is set to New indicating that data in the data field needs to be processed. Multiple PRs can access the channel simultaneously and they place their data (*placeData*) in a non-blocking fashion in the specific package area dedicated to each PR. When done, the status filed of the package in the channel will be set to Complete, that can be accessed by PRs using *getStatus* interface function. **SPs:** The channel only has bandwidth for allowing only one SP and it uses random monitory scheme for multiple simultaneous accesses. When an SP is free to check the channel, it accesses the channel using *getData* interface method. Once it has given access, it searches the channel for the type of task that it is designed to do. If it finds its task, it will get the Trans class and take it back to the SP module. The module processes the data and when completed, it will return the processed data back to the channel using *returnData* method. Accessing the channel for returning procced data uses the same monitory scheme as receiving data from the channel. **Design:** Show the complete design of the channel including its interfaces and channel implementation. The interfaces and the channel are templated classes. Show how the channel is connected to PRs and SPs. Also show how a typical PR uses its interfaces with the channel, and how an SP uses its interfaces.