UNIVERSITY OF TEHRAN

Electrical and Computer Engineering Department

ECE (8101) 432

Object Oriented Modeling of Electronic Circuits – Spring 95-96

Final Exam

Computer Account#_____

First Name:_____

Last Name:_____

Student Number:_____

Signature:_____

Grade:

Problem 1. _____/20

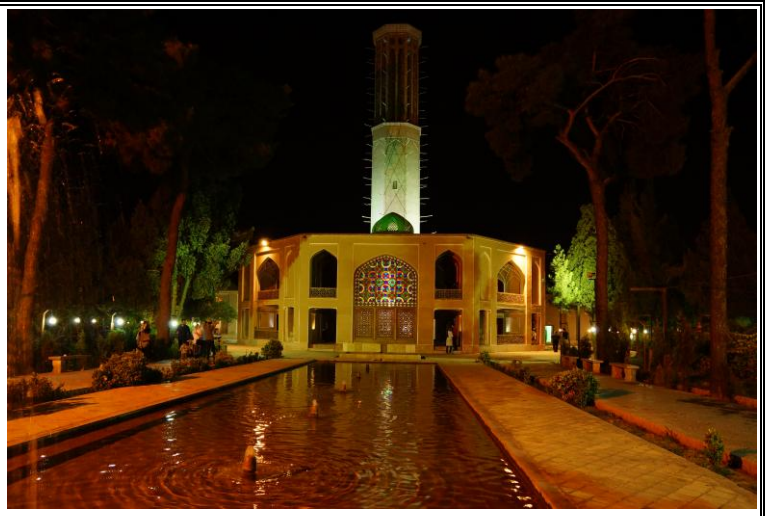
Problem 2. _____/20

Problem 3. _____/20

Problem 4. _____/20

Problem 5. _____/20

Total: _____/100



Baaghe Dolat-Aabaad, Yazd, Iran, Khordad 1396

DO NOT USE LAPTOPS
EXTRA SHEETS WILL NOT BE ACCEPTED
YOU MUST SHOW COMPLETE WORK ON ALL PROBLEMS
YOU HAVE EXACTLY 150 MINUTES FOR WORKING ON THIS TEST
THIS IS AN OPEN BOOK OPEN NOTE EXAM, NO SHARING ALLOWED

SystemC basic concepts

1. You are to write a SystemC description of a memory timer with a configurable wait time. The timer has a clock (*clk*) input, a read-write (*rw*) input, and a memory-ready (*memrdy*) output. When *rw* is issued, the timer waits *n* clock cycles and then it issues *memrdy* for exactly one clock cycle. Number *n* (wait time) is passed to the timer when instantiated.

a. Write SystemC description of the timer. You can use whatever level of abstraction or coding style you want. Your choices are C++-like behavioral or RTL.

b. In a module named *cpu_mem_sys*, show how the timer should be instantiated.

c. Use *cpu_mem_sys* in an **sc_main** and show how timer ports should be displayed for waveform viewing.

SystemC templated component

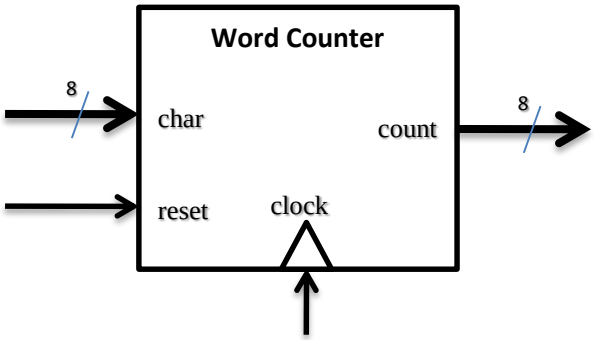
2. Write SystemC description for a dual port clocked register file with two *dataIn* input ports (i.e., *dataIn1* and *dataIn2*), two *writeEnable* control inputs, two *outputPort* outputs, two *addressIn* address inputs, and a clock (*clk*) input. Reading is combinational, and writing requires the positive edge of the clock.

- **a.** Write the templated register file with parameters for the address space and word length. Show the necessary methods in this class definition. Only reading and writing are required, and no initialization or completion methods are needed.
- **b.** Continue Part a with writing SystemC codes for the required methods.
- **c.** Show how this register-file can be declared, instantiated and used in an upper-level design.

SystemC RT level design

3. Show the complete design of a word counter. Design a word counter using only RTL components. Characters of an input text appear sequentially on the 8 bit port (c*har*) of the Word Counter. It has an 8 bit outputs that counts number of words that it sees on its *char* port after *reset*. The words in the text are separated by unknown number of space characters, and also length of each word is unknown.

Note: The ASCII code of space character is 32.



SystemC-AMS basic concepts

4. Answer the following questions:

a. What is the difference between terminal and port in SystemC-AMS?

b. What is the difference between SystemC and SystemC-AMS simulation engines?

c. How can be constructed more than one cluster in a SystemC-AMS design?

d. When do you use **sca_signal** and when do you use **sc_signal** in a mixed-model SystemC/SystemC-AMS design?

SystemC-AMS design

5. In this question you are to design the system shown in the figure shown below using SystemC / SystemC-AMS. This system is constructed using 4 modules (A, B, C and D). Module A is a TDF module that creates samples of a sinusoidal waveform (*sine()*) the amplitude of which is scaled by the value of the input. Module B is an ELN module that models a low-pass filter with time step equal to 10 US. Module C is a TDF module that receives two samples at each activation step, and calculates the subtraction of these samples. For this module, if the result is greater than 0.5 the output becomes true, otherwise false. Module D is a discrete event module that models a multiplexer. If input S is true the output will be equaled to input X, otherwise Y.

a. Define a time step for all ports and modules of the system. Define rate and delay attributes for the ports if needed. Define a scheduling for this system.
b. Write SystemC or SystemC-AMS code for each module.
c. Connect all modules together in an SC_MODULE and construct the complete system. Specify ports, signals, module binding and type of ports and signals.