ECE 4170

Final Examination
May 1st, 2007
2:50 pm – 5:40 pm

1. The Georgia Tech Honor Code governs this examination. Please note you are not to discuss this examination with anyone. Your signature below attests to the same.
2. There are 2 questions.
3. Please write **legibly**.
4. State any assumptions you feel you have to make or ask for clarification
5. Keep in mind it is difficult to give partial credit without written material. Please make sure you document any partial solutions.
6. Submission instructions: A zip file with
    a. Code and trace for problem 1
    b. A Word or Text document with the answer to question 2
    c. All submissions should be sent by 5:40 pm. It is your responsibility to ensure that it is submitted on time.
    d. If you wish, you may submit the same in my office with a memory stick before the end of the exam period.
    e. NO LATE SUBMISSIONS WILL BE ACCEPTED!
7. **Each problem is worth 50 points.**

Problem	Max Points	Graded
1	50	
2	50	
Total	100	

Student Name: _____

Student Number: _____

I have neither received nor given assistance on this examination.

_____

Signature

Question 1

Pick a functioning module from your project. Implement & Test this module in Verilog using ModelSim. The module should not be a trivial module, e.g., a single line of VHDL code. If you are doubtful, ask me. If you feel all modules are individually too large, carve out a sub-function and clearly identify the sub-function. As an example, you should pick a VHDL module with 10-15 lines of code (not counting **entity** and **architecture** statements) or carve out such a piece from your project code. Make sure you precisely identify the module you have taken the code from.

Question 2

Pick a pair of communicating modules from your design. With respect to the communications channel between them

1. Show how you would modify the communication channel so as to be compliant with the open core protocol (OCP)
2. Show the mapping of signals from your existing VHDL interfaces to the OCP signals you have chosen to implement your channel (remember some OCP signals are optional).
3. Pick one of the modules. Produce the state machine for creating an OCP wrapper around that module, i.e., how you add logic to make your module OCP compliant. Precisely describe the project module and the module interface you have chosen. You do not have to implement this wrapper, but the description of the state machine demonstrates that you have thought through how the two modules can communicate over an OCP link.