ECE354 – MIDTERM EXAM, April 13, 2009

Instructions:

- Write your name and student/Spire number here:
- The exam is closed book and closed notes. You have 1 hour to complete the exam.
- First read the entire exam and then budget your time accordingly. Don't waste your time giving details that the question does not request.
- Use the space after the question for your answer.

1. (20 pts) Embedded Processors and Testing

a. Difference between RISC, CISC and VLIW processors.

b. Difference between soft core and hard macro processors used in FPGAs. Compare in terms of relative area, power, and performance.

c. What type of processor have you used in the SOPC builder?

d. How was the Altera FPGA programmed in the projects?

e. List the various testing related uses of the JTAG interface.

2. (10 pts). Embedded Memory Systems

(i) Show a design of an embedded memory system with a relocation register.

(ii) Compare your design with a system that has a TLB and page table. What is the disadvantage of your relocation register based system?

3. (10 pts). Interrupts in Embedded Systems

(i) What interrupt type would you use to receive data from a keyboard in a time-shared system that also has connection to an Ethernet network and supports virtual memory:
   a. NMI
   b. Relatively low priority
   c. High priority

Explain your choice:

(ii) Difference between an exception, software interrupt, and hardware interrupt.

3. (10 pts). IO Interfaces

(i) List the parameters of the RS 232 serial communication.

(ii) Describe how the PCIe architecture works briefly. Is it a bus-based system?

4. (10 pts). Software Systems

(i) Why are dynamically linked libraries (DLLs) preferred compared to statically linked libraries in embedded systems?

(ii) Difference between threads and processes.

5. (10 pts) Networks

(i) Describe the TCP/IP packet format you needed to implement in the project.

(ii) How is collision managed on Ethernet?

6. (15 pts) Process State in Memory

Please show the memory layout at runtime for code below including PC, SP, and HP sections for the program below. Partition your memory into regions with text section, stack, heap, and static data segment. Assume the PC points to the "if (b==2) a = 1" line when the memory content is shown. Use space to the right from the code to show your work.

```
int foo (int b){
        int a;
        if (b == 2)  a = 1;
        else a = b;
        return (a);
}


main(){
        int  x = 2;
        int y;
        y= foo (x);
        ….

}
```

7 . (15 pts) Synchronization

The following pseudo-code is the implementation of a test-and-set instruction in processors. It can be used to achieve synchronization between processes accessing a critical section. See the diagram below for two processes running on CPU 1 and CPU 2 accessing the red region in shared memory. Please explain why a test-and-set instruction is necessary for enabling synchronization. Why can't we just use simple flags in shared memory for this purpose?

```
boolean TestAndSet(boolean &target) {
        boolean rv = target;
        target = true;

        return rv;
}
```