# ECE354 – EXAM, April 26, 2017 – C. Andras Moritz

**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 Labs - all questions have same weight**

a. List the differences between hardware interrupts and software interrupts

b. List the differences between RISC and VLIW microprocessors

c. How is the FPGA based soft-core different from a hard-core processor built into silicon?

d. When is RLE not beneficial as a compression technique?

e. What image transformations did you implement in lab 2?

**2. (10 pts) Embedded Memory Systems**

Show a design (block diagram) of an embedded memory system with a TLB and page table. If you don't remember, describe the role of the TLB vs. the page table.

**3. (15 pts) IO Interfaces**

(i) List at least two parameters of the UART serial communication standard.

(ii) Describe how the PCIe architecture works briefly. Is it a bus-based system? (5 pts)

**4. (15 pts) Software Systems**

(i) What are the various states a process is going through during its lifecycle? Show the state machine for process scheduling.

(ii) Difference between threads and processes. Which one is more lightweight? (5 pts)

**5. (20 pts) Labs Related Questions - all questions have same weight**

(i) Why was FIFO necessary in the hardware implementation of the RLE encoder?

(ii) How did the software decompression you wrote in Lab 4 access the output of the hardware RLE encoder?

(iii) What is the shortest encoding possible (with our Lab setup) with RLE for an image that is all black but has a single white pixel in the middle. What is the size of the RLE encoded image in that case?

(iv) Why is pin assignment necessary (conceptually)?

**6. (20 pts) Synchronization**

The following pseudo-code shows the hardware implementation of a test-and-set instruction in processors.

```
boolean TestAndSet(boolean &target) { // note this is implemented in hardware
        boolean rv = target;
        target = true;
        return rv;
}
```

(a) Please explain why a test-and-set instruction or similar hardware is necessary for enabling proper synchronization. (10pts)

(b) Please explain the difference between a counting semaphore and a simple lock. (10pts)