EEL 4783: Hardware/Software Co-design with FPGAs

Mid-Term Exam

(Time allowed: ONE hour)

General Instructions:

1. Do NOT open this exam until instructed to do so.

2. There are EIGHT questions in total. Answer the easy ones first to maximize your score.

3. This exam is close-book and close-notes.

4. Show all of your work on the exam to get credit.

5. Clearly indicate your answers.

Signature: _____

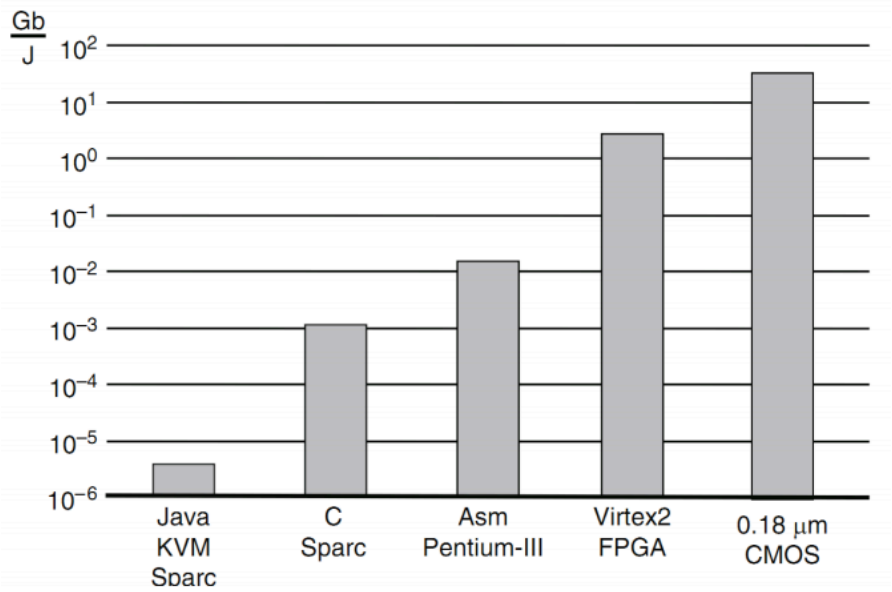
Name (Print): _____

PID: _____

Date: _____

1: (10 pts) Hardware is typically faster and more energy-efficient than its software counterpart. So for any given application, why not using all hardware implementation? What are the benefits of using software for certain parts of the target application?

2: (10 pts)



This figure shows the energy efficiency of different implementations of an application. As it illustrates, CMOS implementation has the best energy efficiency. Based on your understanding, use your own words to explain:

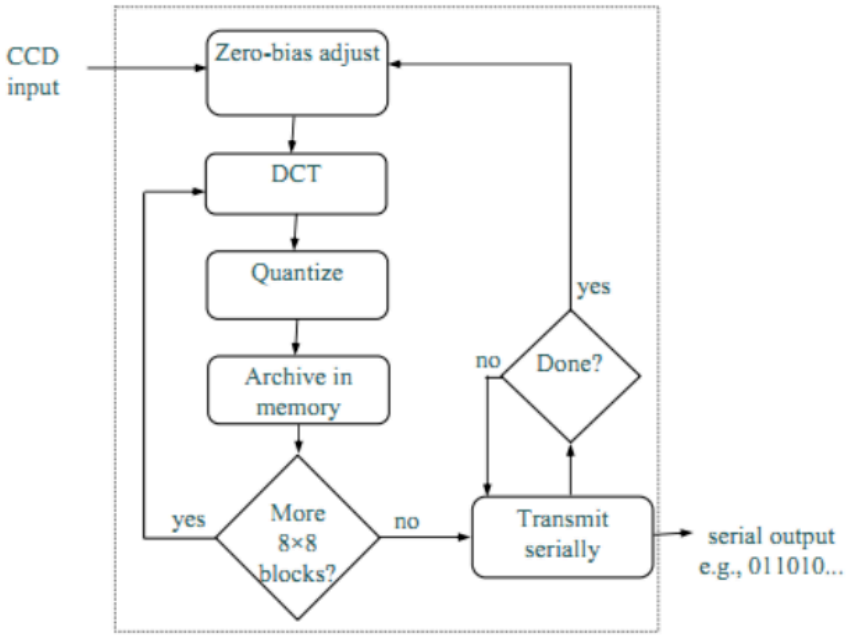
1) Why is hardware implementation so much more energy-efficient than software implementation?

2) Why is ASIC implementation more energy-efficient than an FPGA implementation?

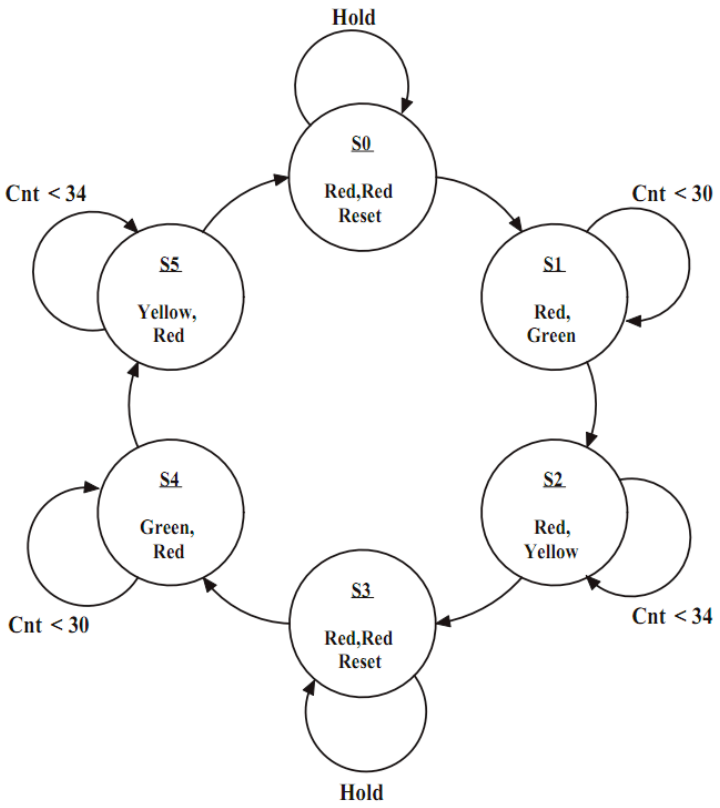
3: (10 pts) What is the main idea behind the Huffman coding algorithm.

4: (10 pts) In our project of Huffman-Lite, we disallow 1 bit Huffman code. Do you think it is possible to use 1 bit as a Huffman code word. If not, why? If yes, How?

5: (15 pts) The following flow diagram shows the task relationship in our software implementation of digital camera. If we would like to add a new functionality called STITCH. This new feature for panoramas enables the photographer to create photos with higher resolution and/or a wider angle of view than their digital camera or lenses would ordinarily allow. As a member of the design team, where should this task module be added to this diagram? Please add the STITCH module to the following task graph. Please state your reasons for your choice. Note that your reasoning is more important than your answers.



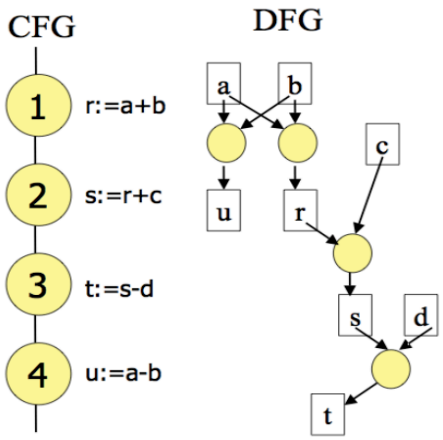
6: (15 pts) For the following state transition diagram, write a Verilog implementation.



7: (15 pts) For the code segment below, draw a Control-Data Flow Graph.

```
If (Condition 1) {
        Switch (Condition 3) {
                Case 1: Code_Block 1;
                  Case 2: Code_Block 2;
                  Default: Code_Block 3;
         }
}
Else If (Condition 2) {
        For (int I=0; I <= 100; I++) {
                Code_Blcok 4;
         }
}
Else {
        If (Condition 4) {
                Code_Blcok 5;
         }
        Else {
                Code_Blcok 6;
        }
}
```

8: (15 pts) Draw the Control Flow Graph, Data Flow Graph for the given code block, and a scheduling diagram to achieve the minimum number of clock cycles for all operations. Assume each operation (in yellow circle) takes exactly one clock cycle. To help you start, we show you an example below.



Code segment:

c = a + b;
d = b + c;
e = b*c;
f = c + d + e;