CS 151
Final

Name : __________________ , __________________
       (Last Name)                 (First Name)

Student ID : _______________

Signature : ________________

Instructions:

1. Please verify that your paper contains 20 pages including this cover.
2. Write down your Student-Id on the top of each page of this final.
3. This exam is closed book. No notes or other materials are permitted.
4. Total credits of this final are 115 + 10 EXTRA CREDIT.
5. To receive credit you must show your work clearly.
6. Calculators are NOT allowed.
7. If necessary, state your assumptions clearly.

<table>
<thead>
<tr>
<th>Total</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>[Extra Credit]</th>
</tr>
</thead>
<tbody>
<tr>
<td>115+10</td>
<td>10</td>
<td>10</td>
<td>20</td>
<td>15</td>
<td>40</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
Q1: [Sequential Circuit Timing Analysis] [10 Points]

The circuit below shows a sequential circuit using D Flip Flops, and Multiplexers. Assuming that $A_1$, $A_0$, and $S$ are the inputs of the circuit, and $Q_1$ is 0, $Q_0$ is 1 when time equals 0 ($t_0$), show the timing diagram for $Q_1$ and $Q_0$.

NOTE: You can assume that gate delay is negligible.
Q2: [Register Design] [10 points]

Design a 4-bit register with 2 control inputs $s_1$ and $s_0$, 4 data inputs $I_3..I_0$, and 4 data outputs $Q_3..Q_0$. Shown below is the function table for the 4-bit register. In this question you have to design the circuit inside the “Black Box”.

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Clear the contents of the register</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Load $I$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Rotate right by one bit</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Rotate left by one bit</td>
</tr>
</tbody>
</table>

---

![Diagram of 4-bit register with control inputs $s_1$ and $s_0$, data inputs $I_3..I_0$, and outputs $Q_3..Q_0$.](image)
Q3: [Counter-based Design] [20 points]

Using only a 4-bit up binary counter, you are going to design a circuit that detects odd numbers in a sequence of numbers, its output (denoted by ODD) is 1 when the sequence value is an odd number, 0 otherwise. The sequence is defined below in Q3a. You can use any of the following components (Specify the bit widths, and name all inputs/outputs):

1) Adders
2) Shifters
3) Comparators
4) Multiplexers
5) Subtractors

Make sure you answer both parts 3a, and 3b.

3a. Using the 4-bit up binary counter, create a circuit that generates the following sequence [10 points]:

48 → 45 → 42 → 39 → 36 → 33 → 30 → 27 → 24 → 21 → 18 → 15 → 12 → 9 → 6 → 3 → 48 → 45 → 42 …
<This page is intentionally left blank>
3b. Using the circuit designed in part 3a, modify it so that whenever an odd number is encountered in the sequence generated, your circuit outputs a 1 (ODD signal is 1), otherwise, it outputs a 0. [5 points]
Q4: [State minimization] [15 points]

4a. Reduce the number of states in the following state machine using the implication table method. Which states are the same? [10 points]
4b. Draw the reduced state machine below. [5 points]
Q5: [C to RTL design] [40 points]

The following C-code computes the intensity histogram of a series of pixel values while keeping track of the largest intensity count.

Inputs: byte pixels[256], bit start
Outputs: byte max_out, bit done

main() {
    byte max;
    byte pvalue;
    byte hvalue;
    byte i;
    byte histogram[256]; // assume when cleared, all entries are set to 0

    while(1) {
        while(!start); // wait for start signal

        done = 0;
        i=0;
        max = 0;
        pvalue = 0;
        hvalue = 0;

        while(i<256) {
            // for simplicity the following lines are split statements
            // from histogram[pixel[i]]++;

            pvalue = pixel[i];
            hvalue = histogram[pvalue];
            hvalue++;
            histogram[pvalue] = hvalue;
            if(hvalue>max) max = hvalue;
            i++;
        }

        max_out=max;
        done=1;
    }
}
To help you understand, below is a block diagram of the datapath. It consists of the datapath black box, and two memory interfaces, the histogram memory (256 bytes), and main memory which contains the pixel array.
5a. Derive the high-level state machine from the provided C-code. [5 points]
5b. Design the data-path for this system. [15 points]

You may use any of the following components:
   1) Registers
   2) Adders
   3) Comparators
   4) Multiplexers
5c. Design the interface of the system and the interface between the controller and the datapath. [5 points]
5d. Design the FSM of the controller. [15 points]

HINT: There is no timing issue for this system so you do not have to consider timing issues in designing the controller’s FSM.
Question 6 [Pipeline utilization] [20 points]

6a. Using only Adders, Subtractors, and Multipliers, design a circuit that performs the following operation: [5 point]

\[ Z = (A-B)*(C+D) + (E+F)*(H-G) \]

A, B, C, D, E, F, G, H and Z are 8 bit registers.

6b. If addition/subtraction takes 10 ns and multiplication takes 30 ns, what is the maximum allowed clock frequency for the above circuit? [5 points]
6c. Add pipeline registers to your design in part (a) to make your design as fast as possible. [5 points]

6d. If addition/subtraction takes 10 ns and multiplication takes 30 ns, what is the maximum allowed clock frequency for the above circuit? [5 points]
EXTRA CREDIT

E1: Consider the schedule below for computing $Z = (A-B)(C+D) + (E+F)(H-G)$ that uses 5 adder/subtractors and 2 multipliers:

A. Draw a new schedule that uses only 2 add/sub modules and 1 multiplier. How many cycles it will take to compute a new value for $Z$? [5 points]
B. Draw the new datapath based on the schedule in part (e) [Hint: Add the required multiplexers] [5 points]