CSE 140 Midterm 3 - Solution
Prof. Tajana Simunic Rosing
Spring 2013

- Do not start the exam until you are told.
- Write your name and PID at the top of every page. Do not separate the pages.
- Turn off and put away all your electronics.
- This is a closed-book, closed-notes, no-calculator exam. You may only refer to one 8 ½ x 11” page of your handwritten notes.
- Do not look at anyone else’s exam. Do not talk to anyone but an exam proctor during the exam.
- If you have a question, raise your hand and an exam proctor will come to you.
- You have 80 minutes to finish the exam. When the time is finished, you must stop writing.
- Write your answers in the provided space.
- No credit will be given if you do not show all steps of your work.

<table>
<thead>
<tr>
<th></th>
<th>1. 15 points</th>
<th>2. 20 points</th>
<th>3. 15 points</th>
<th>4. 15 points</th>
<th>5. 20 points</th>
<th>6. 15 points</th>
<th>Total (100 pts.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 1

a) True or False?

1-1. _T_ The following two circuits have the same functionality.

1-2. _F_ One advantage of ROM over RAM is that ROM is volatile.
1-3. _T_ The output of a Mealy state machine changes asynchronously.
1-4. _F_ A logic function is shown in the Karnaugh map below. The minimal SOP implementation of this function, AB+A’C, does not have a static timing hazard.

1-5. _T_ Output Y of the circuit below computes the sum bit of a full adder.

1-6. _T_ The delay of a ripple-carry adder is linearly dependent on the number of bits.
1-7. F A ripple-carry adder is never faster than carry-lookahead adder.

b) Multiple Choice

1-8. What sequence does the following circuit detect?

a) 011  
 b) 100  
 c) 010  
 d) 101

1-9. What is the canonical form for Y based on the truth table below?

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

a) \( \Sigma m(0,2,5)+d(1,5) \)  
b) \( \Pi M(0,1,2)*d(1,5) \)  
c) \( \Sigma m(3,4,5,6)+d(1,7) \)  
d) \( \Pi M(3,4,6)*d(1,5,7) \)
1-10. What does the circuit in figure below represent? How many transistors does it have?

![Circuit Diagram]

a) D-latch with 6 transistors  
b) SR-flip-flop with 8 transistors  
c) D-flip-flop with 6 transistors  
d) SR-latch with 8 transistors

1-11. What is the minimum number of D flip-flops required to design a counter circuit that outputs the first seven Fibonacci numbers and then wraps around? Fibonacci numbers are defined by \( F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2} \).

a) 3  
b) 4  
c) 5  
d) 6
Problem 2 – ALU
a) Draw the schematic for an ALU with unsigned 3-bit inputs A and B and two control bits C1 and C0. The ALU implements functionality shown in the table below. Use a minimum number of 2:1 MUXs, a single 3-bit adder (with a carry-in input), and a minimum number of inverters.

<table>
<thead>
<tr>
<th>$C_1C_0$</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>$A-B-1$</td>
</tr>
<tr>
<td>01</td>
<td>$A-B$</td>
</tr>
<tr>
<td>10</td>
<td>$B-A-1$</td>
</tr>
<tr>
<td>11</td>
<td>$B-A$</td>
</tr>
</tbody>
</table>

with overflow bit

without overflow bit
b) Using the minimum number of 2:1 MUXs, a single 3-bit adder, and as few gates as possible, implement the ALU functionality shown in the table below. The logical outputs should be 0-extended, meaning that if A=001 & B=100, then A<B is true and the output is 001, and B<A is false, so that output is 000.

Hint: Think about how to use the first four arithmetic functions to perform the four logical operations. Your solution should be an extension of your solution to part a)

<table>
<thead>
<tr>
<th>C2C1C0</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>A-B-1</td>
</tr>
<tr>
<td>001</td>
<td>A-B</td>
</tr>
<tr>
<td>010</td>
<td>B-A-1</td>
</tr>
<tr>
<td>011</td>
<td>B-A</td>
</tr>
<tr>
<td>100</td>
<td>A&lt;=B (zero-extended)</td>
</tr>
<tr>
<td>101</td>
<td>A&lt;B (zero-extended)</td>
</tr>
<tr>
<td>110</td>
<td>B&lt;=A (zero-extended)</td>
</tr>
<tr>
<td>111</td>
<td>B&lt;A (zero-extended)</td>
</tr>
</tbody>
</table>

with overflow bit

without overflow bit
**Problem 3 - Rotator**
Design a 4-bit right rotator with 4-bit input $A_{3:0}$ and 4-bit output $B_{3:0}$. The two-bit control signal $S_{1:0}$ is used to determine the number of bits for $A_{3:0}$ to be shifted, as shown in the table below. For instance, suppose $S_{1:0}=10$ & $A_{3:0}=1100$; after rotate is completed the output $B_{3:0}=0011$. You may use eight 2:1 MUXes and a minimum number of gates to implement this functionality.

<table>
<thead>
<tr>
<th>$S_{1:0}$</th>
<th>Rotator’s Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No shift</td>
</tr>
<tr>
<td>01</td>
<td>Right shift by 1 bit</td>
</tr>
<tr>
<td>10</td>
<td>Right shift by 2 bit</td>
</tr>
<tr>
<td>11</td>
<td>Right shift by 3 bit</td>
</tr>
</tbody>
</table>

![Diagram of 4-bit right rotator](image)
Problem 4 - Timing
You are given a sequential circuit design as shown below.

- $R_pC_g = 10\ \text{ps}$, $R_nC_g = 5\ \text{ps}$
- D-FF clk-to-q propagation delay $t_{pcq}=10\text{ps}$
- D-FF clk-to-q contamination delay $t_{ccq}=5\text{ps}$
- D-FF data setup time $t_s=10\text{ps}$
- Clock skew = 0 ps

<table>
<thead>
<tr>
<th>Delay</th>
<th>$T_{pd} (\text{ps})$</th>
<th>$T_{cd} (\text{ps})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adder($\Sigma$)</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Left shift by 1 bit($&lt;&lt;1$)</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Left shift by 2 bits($&lt;&lt;2$)</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Inverter</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Mux (2:1)</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>
a) Calculate the maximum clock frequency for reliable operation.

\[
\text{period} > (\text{DFF propagation delay}) + \\
(\text{max combination circuit delay}) + \\
(\text{DFF setup time}) + \\
(\text{max clock skew}) \\
= 10 + (50 + 30) + 10 + 0 \\
= 100 \text{ ps}
\]

\[
\text{frequency} < \frac{1}{(100\text{ps})} \\
= 10 \text{ GHz}
\]

b) How long should the hold time be for safe operation?

\[
\text{hold time} \leq \text{DFF contamination delay} + \\
\text{min combinational circuit delay} - \\
\text{max clock skew}
\]

\[
\text{Min combinational circuit delay} = \text{transistor delay} + \text{inverter delay} \\
\text{transistor delay} = \frac{R_p}{2} \times 2C_g = R_p C_g = 10 \text{ ps} \\
\text{inverter delay} = 5 \text{ ps}
\]

\[
\text{hold time} \leq 5 + (10 + 5) - 0 \\
\leq 20 \text{ ps}
\]
Problem 5 – Mealy Sequence Detector
Design a sequence detector for ‘11011’ using D flip-flops. Overlap is allowed between neighboring bit sequences. For instance, let X denote the input and Z denote the output. Assume X=’11011011011’ and the detector will output Z=’00001001001’.

a) Draw the Mealy FSM.

![Mealy FSM Diagram]

b) Fill the state transition table given below using the above FSM. Y_{2:0} are state variables.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Input</th>
<th>Next State</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol</td>
<td>Y2</td>
<td>Y1</td>
<td>Y0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
c) Implement the next state logic for $Y_1^+$ & $Y_0^+$ with the hardware listed below. Use minimum number of additional gates.

- $Y_1^+$ using a 8:1 Mux with $(Y_2,Y_1,Y_0)$ as select lines.
- $Y_0^+$ using a 4:1 Mux with $(Y_2,Y_1)$ as select lines.
Problem 6 – RTL Design

void main()
{
    unsigned int i, start, data, mod4_count, A[128];

    while(1) {
        while(!start);

        i=0;
        mod4_count = 0;

        while(i<128) {
            data = A[i];
            if(data%4==0) {
                mod4_count++;
            }
            i++;
        }
    }
}
a) Use the code shown above to create the high level FSM

\[
\begin{align*}
i &= 0 \\
\text{mod4\_cnt} &= 0 \\
(i &< 128) \\
\text{start} \\
\text{start}' \\
\end{align*}
\]

\[
\begin{align*}
(i &< 128)' \\
\text{B} &\rightarrow \text{C} \\
\text{C} &\rightarrow \text{D} \\
\text{D} &\rightarrow \text{F} \\
\text{F} &\rightarrow \text{E} \\
\text{E} &\rightarrow \text{A} \\
\text{A} &\rightarrow \text{B} \\
\end{align*}
\]

\[
\begin{align*}
i &= i + 1 \\
\text{mod4\_cnt} &= \text{mod4\_cnt} + 1 \\
\end{align*}
\]

b) Show all components in the datapath for this design.
c) Draw the interface between the controller and the datapath, show all inputs and outputs.
This page is intentionally left blank. Use it as scratch paper or to provide additional answers.