CSE140: Components and Design Techniques for Digital Systems

Midterm #2 Sample Problems

Tajana Simunic Rosing
Where we are now…

• What we have covered:
  – Combinational and sequential circuits; Zybook chaps 1-5

• Deadlines:
  – HW#5 due today!!!
  – Quiz #5 average ~80% - Congratulations!!!!
  – Midterm #2 on Thursday at class time
    • All material up to and including today
    • Bring one 8 ½ x 11” paper with handwritten notes, but nothing else
    • Sample midterm problems that are NOT relevant:
      – 1.b) since we did not cover hazards
      – 6.b)-g) we did not cover the process of minimizing states

• Prof. office hours Tuesday & Wed 5-6pm
  – No office hours on Thursday, but can arrange to meet at another time

• Class survey status:
  – More than 90% of students responded so you get your lowest HW grade dropped!!! Congrats!!!!
What do inputs A & B do?

\[ X = Q + A \]
\[ Y = A + B \]
\[ Z = B + Q \]
\[ Q^+ = XYZ \]

Is this a:
- a. Latch
- b. FF

Sources: TSR, Katz, Boriello, Vahid, Perkowski
FSM design example

- Design an overlapping finite string pattern recognizer
  - output is 1 whenever the input sequences 101 and 011 are observed
### Sequential circuit design

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next state</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>W=0</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>W=1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>W=0</th>
<th>W=1</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ z = q_1 q_0 \]
FSM Design Example

• Write the state table and implement the following state machine:
  \[ Q_0(t+1) = Q_0(t) + Q_1'(t) \cdot x(t), \]
  \[ Q_1(t+1) = Q_0'Q_1(t) + x'(t), \]
  \[ y(t) = Q_0(t)Q_1(t). \]

Is this design
a. Moore
b. Mealy
c. None of the above
Logic & Timing diagrams

- $T_{pcq} = 3\text{ns}$; $T_{\text{gate}} = 1\text{ns}$
- All other timing constraints are zero

\[
T_c \geq t_{pcq} + t_{pd} + t_{skw} + t_{	ext{setup}}
\]
Setup and hold constraints

\[ T_c \geq 2 + 3 \cdot 3 \cdot 1 \text{ns} + 1 \text{ns} + 1 \text{ns} \quad T_c \geq 13 \text{ns} \quad f = \frac{1}{13 \text{ns}} \]

**Setup:**
\[ T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}} + t_{\text{skew}} \]

**Hold:**
\[ t_{ccq} + t_{cd} - t_{\text{skew}} > t_{\text{hold}} \]

\[ t_{pd} = 1 \text{ns} \quad t_{cd} = 0.5 \text{ns} \quad t_{ccq} = 2 \quad t_{\text{hold}} = 1 \quad t_{\text{setup}} = 1 \]

**Full Adder:**
\[ C_{\text{out}} = C_{\text{in}} (A + B) + A B \]

**Sum:**
\[ \text{Sum} = C_{\text{in}} \oplus A \oplus B \]

Sources: TSR, Katz, Boriello, Vahid, Perkowski
What does this circuit do?

<table>
<thead>
<tr>
<th>A</th>
<th>Q_1</th>
<th>Q_0</th>
<th>D_1</th>
<th>D_0</th>
<th>Y_1</th>
<th>Y_0</th>
<th>Y_2</th>
<th>Y_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

D_0 = Y_4 \oplus A' 
D_1 = Y_2 \oplus A
What does this circuit do?

Assume that both D-FFs are reset at start

This design contains:
- a. Mux
- b. Decoder
- c. ALU
- d. None of the above

How many inputs does the circuit have:
- e. 0
- f. 1
- g. 2
- h. More than 2

00 → 10 → 00 → 10 → clock → transient state
11 → 00 → 10 → clock → transient state
01 → 01 → 01 → boring circuit

Sources: TSR, Katz, Boriello, Vahid, Perkowski
ALU design

- Design at 2 bit ALU using the specification given below with maximum two full adders

<table>
<thead>
<tr>
<th>$S_1$</th>
<th>$S_0$</th>
<th>ALU Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$F_i = (A_i \equiv B_i)$ (bitwise equality)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$F_i = (A_i &lt; B_i)$ (bitwise strictly less than)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$F_i = A_i + B_i + 1$ (addition, then increment)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$F_i = A_i - B_i - 1$ (subtraction, then decrement)</td>
</tr>
</tbody>
</table>

\[ A + \overline{B} + 1 \Rightarrow A - B \]

\[ 2s \text{ comp} \]

\[ A - B - 1 \]