Lecture 23:
System (RTL) Design

CSE 140: Components and Design Techniques for Digital Systems

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego
Consider the given sequential circuit. Assume that the hold time constraint is satisfied for R2, if there is no clock skew.

Q1: If skew is introduced, under which of the following conditions is the hold time constraint likely to be violated (for R2).

A. CLK1 is delayed compared to CLK2 by 0.1*T<sub>c</sub>

B. CLK1 is advanced compared to CLK2 by 0.1*T<sub>c</sub>

C. The clock period is doubled

D. None of the above
Q2: The given tree of 1:2 decoders implements which of the following functions?

A. \( f(x, y, z) = \Sigma m(1, 2, 6) \)
B. \( f(x, y, z) = \Sigma m(1, 5, 6) \)
C. \( f(x, y, z) = \Sigma m(2, 4, 3) \)
D. \( f(x, y, z) = \Sigma m(1, 2) + d(6) \)
Q3: What is the output of the circuit for the input combination $z = 1$?

A. 1  
B. 0  
C. $x$  
D. $xy$
Q4: Which of the following statements is true?

A. HLSMs can have infinite states, while FSMs have finite states
B. HLSMs specify local storage of data elements and arithmetic operations on data, while FSMs do not.
C. HLSMs specify the control signals that must be transmitted between the controller and the datapath
D. HLSMs are special cases of FSMs

B. HLSMs specify local storage of data elements and arithmetic operations on data, while FSMs do not.
Q5: What is the output of the register ‘tot’ in clock cycle 4 for the following sequence of inputs to the datapath?

Clock Cycle 1: \(a = 25, \ s = 50, \ \text{tot}\_ld = 0, \ \text{tot}\_clr = 1\)

Clock Cycle 2: \(a = 25, \ s = 50, \ \text{tot}\_ld = 1, \ \text{tot}\_clr = 0\)

Clock Cycle 3: \(a = 25, \ s = 50, \ \text{tot}\_ld = 0, \ \text{tot}\_clr = 0\)

A. Output of register tot in clock cycle 4 is 0
B. Output of register tot in clock cycle 4 is 25
C. Output of register tot in clock cycle 4 is 50
D. The output of the register cannot be determined
RTL Design (contd)

- Last lecture: Designed datapath
- This lecture: Design Controller

```
Multiply(X, Y, Z, start, done)
{
S0: If start’ goto S0 || done<-1;
S1: A<- X || B <- Y || i<-0 || M<-0 || done <-0;
S2: If B_{15} = 0 goto S4 || i<-i+1;
S3: M <-M+A;
S4: if i>= 16, goto S6
S5: M<-Shift(M,L,1) || B<-Shift(B,L,1) || goto S2;
S6: Z<-M || done<-1|| goto S0
}```

operation
```
A <- Load (X)
B <- Load (Y)
B <- SHL(B)
M <- Clear(M)
M <- Add(M,A)
M <- SHL(M)
i <- Clear(i)
i <- INC(i)
```
Step 2d: Map Control Signals to Operations

operation

A \leftarrow \text{Load (X)}
B \leftarrow \text{Load (Y)}
B \leftarrow \text{SHL(B)}
M \leftarrow \text{Clear(M)}
M \leftarrow \text{Add(M,A)}
M \leftarrow \text{SHL(M)}
i \leftarrow \text{Clear(i)}
i \leftarrow \text{INC(i)}

Source: CK Cheng
Step 2d: Map Control Signals to Operations

operation

A \leftarrow \text{Load (X)} \quad C_0 = 1
B \leftarrow \text{Load (Y)} \quad C_5 = 0 \text{ and } C_3 = 1
B \leftarrow \text{SHL(B)} \quad C_5 = 1 \text{ and } C_3 = 1
M \leftarrow \text{Clear(M)} \quad C_2 = 1
M \leftarrow \text{Add(M,A)} \quad C_4 = 0 \text{ and } C_1 = 1
M \leftarrow \text{SHL(M)} \quad C_4 = 1 \text{ and } C_1 = 1
i \leftarrow \text{Clear(i)} \quad C_6 = 1
i \leftarrow \text{INC(i)} \quad C_7 = 1
The data path specified which signals should be transmitted from the controller to the DP.
Design the Control Subsystem

Multiply(X, Y, Z, start, done)
{
  S0: If start’ goto S0 || done ← 1;
  S1: A ← X || B ← Y || i ← 0 || M ← 0 || done ← 0;
  S2: If B_{15} = 0 goto S4 || i ← i+1;
  S3: M ← M+A;
  S4: if i>= 16, goto S6
  S5: M ← Shift(M,L,1) || B ← Shift(B,L,1) || goto S2;
  S6: Z ← M || done ← 1|| goto S0
}

Multiply(X, Y, Z, start, done)
{
  S0: If start’ goto S0 || done ← 1;
  S1: C_0=1 || C_3=0 and C_5=1 || C_2=1 || done ← 0;
  S2: If B_{15} = 0 goto S4 || ;
  S3: ;
  S4: if i[4], goto S6
  S5: ; || goto S2;
  S6: Z ← M || done ← 1|| goto S0
}
Design the Control Subsystem

Multiply(X, Y, Z, start, done)
{
S0: If start’ goto S0 || done ← 1;
S1: A ← X || B ← Y || i ← 0 || M ← 0 || done ← 0;
S2: If B_{15} = 0 goto S4 || i ← i+1;
S3: M ← M+A;
S4: if i >= 16, goto S6
S5: M ← Shift(M, L, 1) || B ← Shift(B, L, 1) || goto S2;
S6: Z: ← M || done ← 1|| goto S0
}

Multiply(X, Y, Z, start, done)
{
S0: If start’ goto S0 || done ← 1;
S1: C₀ = 1 || C₅ = 0 and C₃ = 1 || C₆ = 1 || C₂ = 1 || done ← 0;
S2: If B_{15} = 0 goto S4 || C₇ = 1;
S3: C₄ = 0 and C₁ = 1;
S4: if i[4], goto S6
S5: C₄ = 1 and C₁ = 1 || C₅ = 1 and C₃ = 1 || goto S2;
S6: Z: ← M || done ← 1|| goto S0
}

operation
A ← Load (X)  \hspace{1cm} C₀=1
B ← Load (Y)  \hspace{1cm} C₅=0 and C₃ =1
B ← SHL(B) \hspace{1cm} C₅=1 and C₃ =1
M ← Clear(M) \hspace{1cm} C₂ =1
M ← Add(M, A) \hspace{1cm} C₄=0 and C₁=1
M ← SHL(M) \hspace{1cm} C₄=1 and C₁=1
i ← Clear(i) \hspace{1cm} C₆=1
i ← INC(i) \hspace{1cm} C₇=1

Source: CK Cheng
Multiply(X, Y, Z, start, done) 
{
S0: If start' goto S0 || done ← 1;
S1: C_{0}=1 || C_{5}=0 and C_{3}=1 || C_{6}=1 || C_{2}=1 || done ← 0;
S2: If B_{15}=0 goto S4 || C_{7}=1;
S3: C_{4}=0 and C_{1}=1;
S4: if i[4], goto S6
S5: C_{4}=1 and C_{1}=1 || C_{5}=1 and C_{3}=1 || goto S2;
S6: Z: ← M || done ← 1 || goto S0
}
Control Subsystem

Multiply(X, Y, Z, start, done)
{
S0: If start’ goto S0 || done ← 1;
S1: C₀=1 || C₅=0 and C₃ =1 || C₆=1 || C₂ =1 || done ← 0;
S2: If B₁₅ = 0 goto S4 || C₇=1;
S3: C₄=0 and C₁=1;
S4: if i[4], goto S6
S5: C₄=1 and C₁=1|| C₅=1 and C₃ =1 || goto S2;
S6: Z: ← M || done ← 1|| goto S0
}
Control Subsystem: One-Hot State Machine Design

1. Use as many bits to encode states as the number of states
2. To transition into a state the corresponding bit should be set high (all other bits should be set low)

State S0 is encoded as

State S0: 00000001
State S1: 00000010

Source: CK Cheng
Control Subsystem: One-Hot State Machine Design

Input: State Diagram
1. Use a flip flop to replace each state.
2. Set the flip flop which corresponds to the initial state and reset the rest flip flops.
3. Use an OR gate to collect all inward edges.
4. Use a Demux (or a network of AND gates) to distribute the outward edges.

Source: CK Cheng
Is the next state logic for S1 correct?
A. Yes
B. No

Source: CK Cheng
If S0 is currently 1 and start is 1, what is the next state value of (S0, S1)
A. (0, 0)
B. (1, 1)
C. (1, 0)
D. (0, 1)
One-Hot State Machine

Source: CK Cheng
Multiply\((X, Y, Z, \text{start}, \text{done})\)
{
\begin{align*}
S0: & \text{ If start’ goto S0 } \| \text{ done } \leftarrow 1; \\
S1: & \quad C_0 = 1 \| C_5 = 0 \text{ and } C_3 = 1 \| C_6 = 1 \| C_2 = 1 \| \text{ done } \leftarrow 0; \\
S2: & \quad \text{If } B_{15} = 0 \text{ goto S4 } \| C_7 = 1; \\
S3: & \quad C_4 = 0 \text{ and } C_1 = 1; \\
S4: & \quad \text{if } i[4], \text{ goto S6} \\
S5: & \quad C_4 = 1 \text{ and } C_1 = 1 \| C_5 = 1 \text{ and } C_3 = 1 \| \text{ goto S2;} \\
S6: & \quad Z: \leftarrow M \| \text{ done } \leftarrow 1 \| \text{ goto S0} \\
\end{align*}
}

<table>
<thead>
<tr>
<th></th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4 (mux)</th>
<th>C5 (mux)</th>
<th>C6</th>
<th>C7</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Source: CK Cheng
<table>
<thead>
<tr>
<th></th>
<th>C0</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4 (mux)</th>
<th>C5 (mux)</th>
<th>C6</th>
<th>C7</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Source: CK Cheng
Data Subsystem

Source: CK Cheng
Control subsystem

Source: CK Cheng
Next week- Review!