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
RTL Design (contd)

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

```plaintext
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 ← Load (X)
B ← Load (Y)
B ← SHL(B)
M ← Clear(M)
M ← Add(M, A)
M ← SHL(M)
i ← Clear(i)
i ← INC(i)

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

operation
A ← Load (X)  \( C_0 = 1 \)
B ← Load (Y)  \( C_5 = 0 \) and \( C_3 = 1 \)
B ← SHL(B)  \( C_5 = 1 \) and \( C_3 = 1 \)
M ← Clear(M)  \( C_2 = 1 \)
M ← Add(M, A)  \( C_4 = 0 \) and \( C_1 = 1 \)
M ← SHL(M)  \( C_4 = 1 \) and \( C_1 = 1 \)
i ← Clear(i)  \( C_6 = 1 \)
i ← INC(i)  \( C_7 = 1 \)

Source: CK Cheng
Design the Control Subsystem

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

Multiply(X, Y, Z, start, done)
{
S0: If start’ goto S0 || done \( \leftarrow 1 \);
S1: || || || || done \( \leftarrow 0 \);
S2: If \( B_ {15} = 0 \) goto S4 || ;
S3: ;
S4: if \( i[4] \), goto S6
S5: || || || goto S2;
S6: Z \( \leftarrow M \) || done \( \leftarrow 1 \) || goto S0
}

operation
\[\begin{align*}
A & \leftarrow \text{Load} \ (X) & C_0 = 1 \\
B & \leftarrow \text{Load} \ (Y) & C_5 = 0 \text{ and } C_3 = 1 \\
B & \leftarrow \text{SHL} (B) & C_5 = 1 \text{ and } C_3 = 1 \\
M & \leftarrow \text{Clear} (M) & C_2 = 1 \\
M & \leftarrow \text{Add} (M, A) & C_4 = 0 \text{ and } C_1 = 1 \\
M & \leftarrow \text{SHL} (M) & C_4 = 1 \text{ and } C_1 = 1 \\
i & \leftarrow \text{Clear} (i) & C_6 = 1 \\
i & \leftarrow \text{INC} (i) & C_7 = 1
\end{align*}\]

Source: CK Cheng
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_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
}

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
}

Source: CK Cheng
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

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
One-Hot State Machine

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, 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
}

<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
<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!