CS 140 Lecture 15
Sequential Modules

Professor CK Cheng
CSE Dept.
UC San Diego
Standard Sequential Modules

1. Serial Adders
2. Serial Multipliers
3. Register
4. Counter
Motivation for Serial Adders and Multipliers

• Tradeoff of silicon area and system performance
  – Perform process in a series of time
• Utilization of FPGA architecture
  – Slice operation bitwise
• Metrics of Cost, Speed, and Power
• Ad: Cheaper hardware, Fit for FPGA architecture, Pipelining for excellent throughput
• Dis: Longer latency
Serial Adder: Perform serial bit-addition

At time $i$, read $a_i$ and $b_i$. Produce $s_i$ and $c_{i+1}$

Internal state stores $c_i$. Carry bit $c_0$ is initialized as $c_{in}$
Feed a_i and b_i and generate s_i at time i. Where are c_i and c_{i+1}?
Serial Adder using a D Flip-Flop

<table>
<thead>
<tr>
<th>id</th>
<th>a&lt;sub&gt;i&lt;/sub&gt;</th>
<th>b&lt;sub&gt;i&lt;/sub&gt;</th>
<th>c&lt;sub&gt;i&lt;/sub&gt;</th>
<th>c&lt;sub&gt;i+1&lt;/sub&gt;</th>
<th>s&lt;sub&gt;i&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</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>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ Q(t) = c_i \]
\[ Q(t + 1) = D = c_{i+1} \]
Serial Adder using a D Flip-Flop
Logic Diagram
Multiplication using Serial Addition

3 X 5 = 15

\[
\begin{array}{ccc}
0 & 1 & 1 \\
x & 1 & 0 & 1 \\
\hline \\
0 & 1 & 1 \\
0 & 0 & 0 \\
+ & 0 & 1 & 1 \\
\hline \\
0 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
a_2 & a_1 & a_0 \\
x & b_2 & b_1 & b_0 \\
\hline \\
a_2b_0 & a_1b_0 & a_0b_0 \\
a_2b_1 & a_1b_1 & a_0b_1 \\
\hline \\
+ & a_2b_2 & a_1b_2 & a_0b_2 \\
\hline \\
m_5 & m_4 & m_3 & m_2 & m_1 & m_0 \\
\end{array}
\]

For \( m=AxB \), set \( m^{(0)}=0 \)

At time \( i \), perform \( m^{(i+1)}=m^{(i)}+Ab_i2^i \)
Register

\[ Q(t+1) = (0, 0, \ldots, 0) \quad \text{if CLR} = 1 \]
\[ = D \quad \text{if LD} = 1 \text{ and CLR} = 0 \]
\[ = Q(t) \quad \text{if LD} = 0 \text{ and CLR} = 0 \]
Counter

- Program Counter
- Address Keeper: FIFO, LIFO
- Clock Divider
- Sequential Machine
Counter

- Modulo-n Counter
- Modulo Counter (m<n)
- Counter (a-to-b)
- Counter of an Arbitrary Sequence
- Cascade Counter
Modulo-n Counter

Q (t+1) = (0, 0, .. , 0) if CLR = 1
  = D if LD = 1 and CLR = 0
  = (Q(t)+1)mod n if LD = 0, CNT = 1 and CLR = 0
  = Q (t) if LD = 0, CNT = 0 and CLR = 0

TC = 1 if Q (t) = n-1 and CNT = 1
    = 0 otherwise
Modulo-m Counter (m < n)

Given a mod 16 counter, construct a mod-m counter (0 < m < 16) with AND, OR, NOT gates

\[ m = 6 \]

Set LD = 1 when \( X = 1 \) and \((Q_3Q_2Q_1Q_0) = (0101)\), ie \( m-1 \)
Counter (a-to-b)

Given a mod 16 counter, construct an a-to-b counter (0 ≤ a ≤ b ≤ 15)

A 5-to-11 Counter

Set LD = 1 when X = 1 and (Q_3Q_2Q_1Q_0) = b (in this case, 1011)
Counter of an Arbitrary Sequence

Given a mod 8 counter, construct a counter with sequence 0 1 5 6 2 3 7

When Q = 1, load D = 5
When Q = 6, load D = 2
When Q = 3, load D = 7
Counter of an Arbitrary Sequence

Given a mod 8 counter, construct a counter with sequence 0 1 5 6 2 3 7

<table>
<thead>
<tr>
<th>Id</th>
<th>Q_2Q_1Q_0</th>
<th>LD</th>
<th>D_2</th>
<th>D_1</th>
<th>D_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

K Mapping LD and D, we get

\[ LD = Q'_2 Q_0 + Q_2 Q'_0 \]

\[ D_2 = Q_0 \]
\[ D_1 = Q_1 \]
\[ D_0 = Q_0 \]
Counter of an Arbitrary Sequence

Example: Count in sequence 0 2 3 4 5 7 6

LD = 1 D = 2 When Q(t) = 0
LD = 1 D = 7 When Q(t) = 5
LD = 1 D = 6 When Q(t) = 7
LD = 1 D = 0 When Q(t) = 6

Through K-map, we derive

LD = Q’_2 Q’_1 + Q_2 Q_0 + Q_2 Q_1
D_2 = Q_0
D_1 = Q’_1 + Q_0
D_0 = Q’_1 Q_0

<table>
<thead>
<tr>
<th>Id</th>
<th>Q_2 Q_1 Q_0</th>
<th>LD</th>
<th>D_2</th>
<th>D_1</th>
<th>D_0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Cascade Counter

A Cascade Modulo 256 Counter
Cascade Counter

TC = 1 when \((Q_3, Q_2, Q_1, Q_0) = (1, 1, 1, 1)\) and \(X=1\)
\[(Q_7^{(t+1)} Q_6^{(t+1)} Q_5^{(t+1)} Q_4^{(t+1)}) = (Q_7^{(t)} Q_6^{(t)} Q_5^{(t)} Q_4^{(t)}) + 1 \mod 16\]
when \(T_{C0} = 1\)

The circuit functions as a modulo 256 counter.

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>…</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_{7-4}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>…</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>T_{C0}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>…</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q_{3-0}</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>…</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Conclusion

• Tradeoff of silicon area and system performance
  – Spatial vs. Temporal Dimension Expansion
  – Combinational vs. Sequential Implementation
  – Wires vs. Embedded States for Communication

• Metrics of Cost, Speed, and Power

• Ad: Cheaper hardware, Fit for FPGA architecture, Pipelining for excellent throughput

• Dis: Longer latency