Lecture 12:
Sequential Networks – Flip flops and registers

CSE 140: Components and Design Techniques for Digital Systems

Diba Mirza
Dept. of Computer Science and Engineering
University of California, San Diego
Announcements

• Section B: Midterm on Friday at lecture time.
How is the pace of the class?

A. Too fast
B. Too slow
C. Okay
Midterm review

True or False

1. In a K-map, a group of two adjacent cells containing zeros is an implicant. **False**

2. A minterm may evaluate to a one for multiple input combinations. **False**

3. An incompletely specified function results in a minimal circuit whose output cannot be determined for some input combinations. **False**

4. When minimizing a function to product of sum form, we always assume that the don't care terms are 1. **False**

5. The output of combinational circuits depends only on current inputs. **True**

A: True
B: False
Reduce to SOP form

\[ F(a,b,c,d) = \Pi M(3,4,5,6,7,11,12). \Pi D(10, 15) \]
Reduce the following to a SOP form

\[ F(a,b,c,d) = \Pi M(3,4,5,6,7,11,12). \Pi D(10, 15) \]
Minimize using Boolean algebra

• $F(x,y,z) = xy + y'((x+y')' + z) + xz$

  
  
  $= xy + ar{y} \left[ (\bar{x} + \bar{y}) + z \right] + xz$

  
  $= xy + \bar{y} \cdot (x + \bar{y}) + \bar{y}z + xz$

  
  $= xy + \bar{y} \cdot \bar{x} \cdot y + \bar{y}z + xz$  (Distributive law)

  
  $= xy + \bar{y} \cdot \bar{x} \cdot y + \bar{y}z + xz$

  
  $= xy + \bar{y} \cdot 0 + \bar{y}z + xz$  (Consume)

  
  $= xy + \bar{y}^2$
Design Problem

Design a circuit that gives the absolute distance between two 2-bit numbers (e.g. \( x=3, y=1, d=2 \))
Design Problem

- Design a circuit that gives the absolute distance between the two numbers (e.g. x=3 y=1 d=2)

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
x_1 & x_0 & y_1 & y_0 & d_1 & d_0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
\hline
\end{array}
\]

\[
d_1 = \Sigma m(8,12) + \Sigma m(12,13) + \Sigma m(24) + \Sigma m(34) \\
= x_1 \bar{y}_0 \bar{y}_1 + x_1 \bar{y}_0 y_1 + \bar{x}_1 y_0 + \bar{x}_1 y_0
\]

\[
d_0 = \Sigma m(13,31) + \Sigma m(4,4,12,14) \\
= \bar{x}_0 y_0 + \bar{x}_0 \bar{y}_0
\]
D Flip-Flop vs. D Latch
How long does a D-flip flop store a bit at its output before the output can potentially change?

A. Half a clock cycle
B. One clock cycle
C. Two clock cycles
D. There is no minimum time
D Flip-Flop (Delay)


characteristic expression

\[ Q(t+1) = D(t) \]

What does the equation mean?
JK F-F

Solve a lot like a latch I know…

State table

<table>
<thead>
<tr>
<th>JK</th>
<th>PS</th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Q(t+1) = Q(t)K’(t)+Q’(t)J(t)

Characteristic Expression
JK F-F

Characteristic Expression
\[ Q(t+1) = Q(t)K'(t) + Q'(t)J(t) \]

State table

<table>
<thead>
<tr>
<th>PS</th>
<th>JK</th>
<th>Q(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00 01 11 10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1 1 0 0 1</td>
<td></td>
</tr>
</tbody>
</table>

Q(t+1)

CLK

J

K

Q

Q'

CLK

J

K

Q

Q(t)

Q(t+1)

Function

Memroy

00  Set to 0'

01  Toggle

11  Set to 1

10  Set to 0
T Flip-Flop (Toggle)

Characteristic Expression
\[ Q(t+1) = Q'(t)T(t) + Q(t)T'(t) \]

State table

<table>
<thead>
<tr>
<th>PS</th>
<th>T</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Q(t+1)
Using a JK F-F to implement a D and T F-F

The above circuit behaves as which of the following flip flops?
A. D F-F
B. T F-F
C. None of the above
Using a JK F-F to implement a D and T F-F

T flip flop
Enabled Flip-Flops

- **Inputs:** $CLK$, $D$, $EN$
  - The enable input ($EN$) controls when new data ($D$) is stored

- **Function**
  - $EN = 1$: $D$ passes through to $Q$ on the clock edge
  - $EN = 0$: the flip-flop retains its previous state
Resettable Flip-Flops

- **Inputs:** $CLK, D, Reset$
- **Function:**
  - $Reset = 1$: $Q$ is forced to 0
  - $Reset = 0$: flip-flop behaves as ordinary D flip-flop
- Two types:
  - **Synchronous:** resets at the clock edge only
  - **Asynchronous:** resets immediately when $Reset = 1$
- Asynchronously resettable flip-flop requires changing the internal circuitry of the flip-flop
- Synchronously resettable flip-flop circuit:

- There are also synch/asynch settable FFs
Bit Storage Overview

**SR latch**
- Feature: S=1 sets Q to 1, R=1 resets Q to 0. Problem: SR=11 yield undefined Q.

**Level-sensitive SR latch**
- Feature: S and R only have effect when C=1. We can design outside circuit so SR=11 never happens when C=1. Problem: avoiding SR=11 can be a burden.

**D latch**
- Feature: SR can't be 11 if D is stable before and while C=1, and will be 11 for only a brief glitch even if D changes while C=1. Problem: C=1 too long propagates new values through too many latches: too short may not enable a store.

**D flip-flop**
- Feature: Only loads D value present at rising clock edge, so values can't propagate to other flip-flops during same clock cycle. Tradeoff: uses more gates internally than D latch, and requires more external gates than SR – but gate count is less of an issue today.
Building blocks with FFs: Basic Register