Outline

1. NAND gate latch and D-latch
2. Flip-flop
3. Implementation Technology
4. Transistor in Logic Circuit
4. Programmable Logic Array
Revision: NAND gate Latch

Logic symbol

Ref: NAND gate Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>(AB)'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SET</th>
<th>CLR</th>
<th>Q</th>
<th>Q</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>invalid</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Clear</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0/1</td>
<td>1/0</td>
<td>Retain</td>
</tr>
</tbody>
</table>
D-Latch

For D-latch,
When CLK = 1, D = Q
When CLK = 0, D is no change

Logic symbol

<table>
<thead>
<tr>
<th>D</th>
<th>CLK</th>
<th>Q</th>
<th>Q'</th>
<th>STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>No change</td>
<td>No change</td>
<td>Retain</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Set</td>
</tr>
</tbody>
</table>
Exercise: D-Latch

Draw the waveform of Q

C _______                        ___________

D _______                        ___________

Q

\[ \text{Diagram of a D-Latch with waveforms for C, D, and Q.} \]
Exercise: D-Latch

Draw the waveform of Q

C

D

Q
Flip-Flop

• Flip-flop responds to an active transition.
• Just after an active transition, $Q = D$.
• Otherwise, $Q$ is unchanged.
• Remember flip-flop is edge-triggered, but latch isn’t.

Positive-edge triggered flip flop

Negative-edge triggered flip flop
Exercise: Flip-flop

Draw the waveform of Q

CLK   _______   _______

D     _______   _______

Q for positive-edge triggered

Q for negative-edge triggered

Positive-edge triggered flip flop  Negative-edge triggered flip flop
Exercise: Flip-flop

Draw the waveform of Q

CLK

D

Q for positive-edge triggered

Q for negative-edge triggered

Positive-edge triggered flip flop

Negative-edge triggered flip flop
JK Flip-Flop

\[ D = JQ' + K'Q \]

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t+1)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q(t)</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q'(t)</td>
<td>Complement</td>
</tr>
</tbody>
</table>
Exercise: JK Flip-flop

Draw the waveform of Q

CLK
J
K
Q
Q'

JK flip flop

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t+1)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q(t)</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q’(t)</td>
<td>Complement</td>
</tr>
</tbody>
</table>
Exercise: JK Flip-flop

Draw the waveform of Q

CLK  

J  

K  

Q  

Q'  

JK flip flop

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q(t+1)</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q(t)</td>
<td>No change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Reset</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Set</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Q'(t)</td>
<td>Complement</td>
</tr>
</tbody>
</table>
NMOS and PMOS in Logic Circuit

**NMOS**

- Voltage at Gate: ~0V, 0V (0), short
- Voltage at Gate: ~V_{DD}V, 1V (1), short

**PMOS**

- Voltage at Gate: ~V_{DD}V, 1V (1), Open
- Voltage at Gate: ~0V, 0V (0), short

X = 0

X = 1
NOT gate by NMOS

<table>
<thead>
<tr>
<th>Voltage at Gate</th>
<th>X</th>
<th>Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>~0V</td>
<td>0</td>
<td>Open</td>
</tr>
<tr>
<td>~(V_{DD})V</td>
<td>1</td>
<td>short</td>
</tr>
</tbody>
</table>
NOT gate by PMOS

X

F

F = 1

F = 0

X = 0

X = 1

<table>
<thead>
<tr>
<th>X</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

PMOS

<table>
<thead>
<tr>
<th>Voltage at Gate</th>
<th>X</th>
<th>Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>(~V_{DD}) V</td>
<td>1</td>
<td>Open</td>
</tr>
<tr>
<td>(~0V)</td>
<td>0</td>
<td>short</td>
</tr>
</tbody>
</table>
NOT gate by CMOS

No Current flow through resistor by using CMOS to implement NOT gate, power loss is reduced.
CMOS Realization of a Gate from a given function

**Pull-up network**
1. Complement the input
2. Parallel => OR
   Series => AND

**Pull-down network**
1. Parallel => OR
   Series => AND
2. Complement the output
Exercise: CMOS Realization of a Gate from a given function

**Pull-up network**
1. Complement the input
2. Parallel => OR
   Series => AND

**Pull-down network**
1. Parallel => OR
   Series => AND
2. Complement the output

**DeMorgan’s Theorem**

\[(A+B)' = A'\cdot B'\]
\[(AB)' = A' + B'\]
Exercise: CMOS Realization of a Gate from a given function

**Pull-up network**
1. Complement the input
2. Parallel => OR
   Series => AND

**Pull-down network**
1. Parallel => OR
   Series => AND
2. Complement the output

**DeMorgan’s Theorem**
- \((A+B)' = A'B'\)
- \((AB)' = A' + B'\)
Exercise: Programmable Logic Array

\[ F_1 = x'_1 x_2 + x_1 x_2 \]
\[ F_2 = x'_1 x'_2 + x_1 x_2 \]
Exercise: Programmable Logic Array

\[ F_1 = x_1'x_2 + x_1x_2 \]
\[ F_2 = x_1'x_2' + x_1x_2 \]
Exercise: Programmable Logic Array

\[ F_1 = x'_1 x_2 + x_1 x_2 \]
\[ F_2 = x'_1 x'_2 + x_1 x_2 \]
Exercise: Programmable Logic Array

\[ F_1 = x'_1 x_2 + x_1 x_2 \]
\[ F_2 = x'_1 x'_2 + x_1 x_2 \]