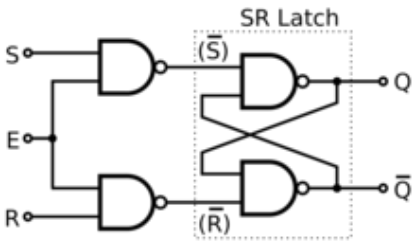
Harold's Flip-Flops
Cheat Sheet
13 June 2020

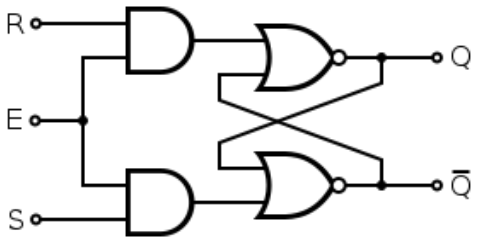
S-R Flip-Flop (Edge-Triggered)

Style

NAND-NAND

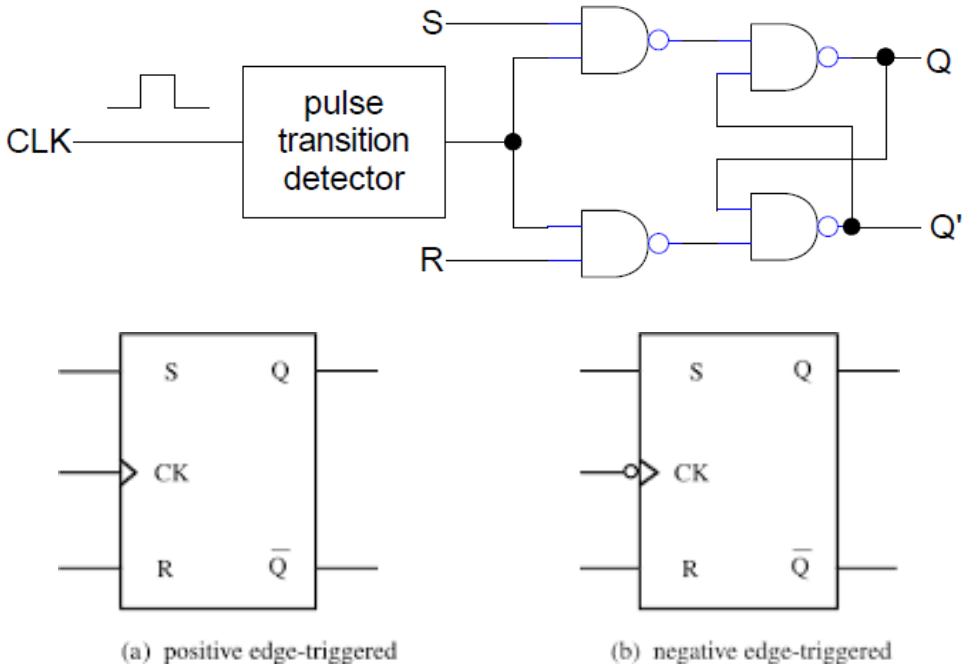


AND-NOR



Circuit

Symbol



Truth Table

S	R	Q_{next}	Action
0	0	Q	No change, Hold
0	1	0	Reset (Q → 0)
1	0	1	Set (Q → 1)
1	1	X	Invalid, Not allowed

Boolean Equation

$$Q_{next} = \bar{R}Q_{prev} + \bar{R}S = \bar{R}(Q_{prev} + S)$$

Name Origin

SR for Set-Reset

Observations

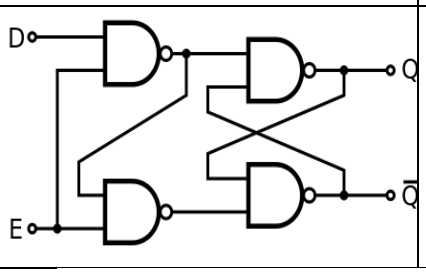
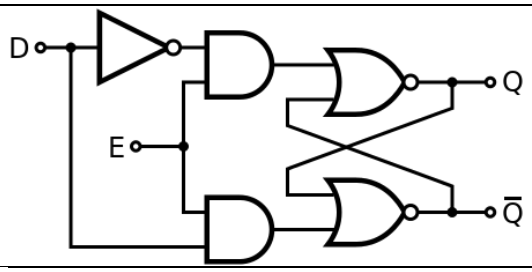
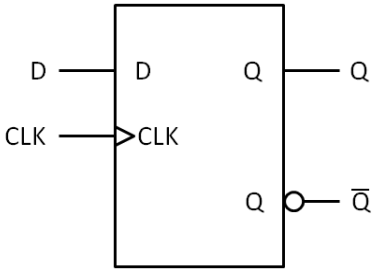
A **Flip-Flop** is a **Latch** with 2 AND/NAND gates added for clock input to trigger data flow from left to right

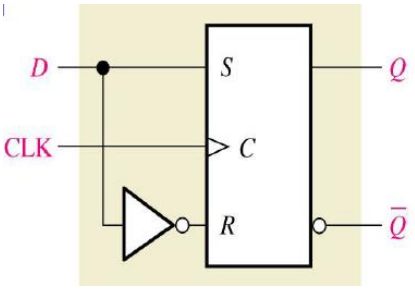
Applications

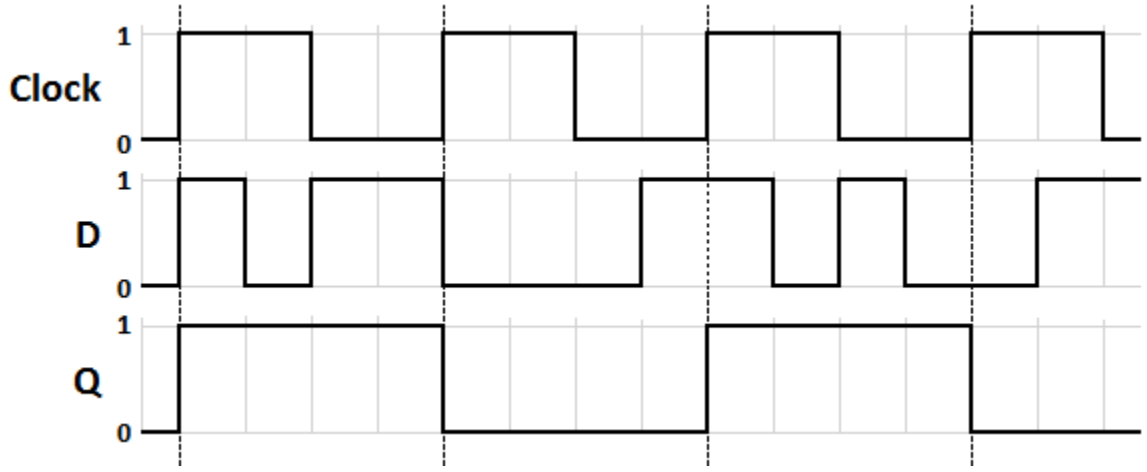
- Storing a single bit of data, 1 or 0

TTL Chips

74x71, 74Lx74

<table>
<tr><th colspan="3">D Flip-Flop (Edge-Triggered)</th></tr>
<tr><td>Style</td><td>NAND-NAND</td><td>AND-NOR</td></tr>
<tr><td>Circuit</td><td> 
</td><td> 
</td></tr>
<tr><td>Symbol</td><td colspan="2"> 

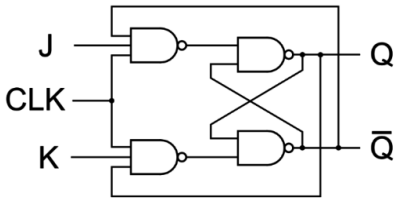

</td></tr>
<tr><td>Truth Table</td><td colspan="2">
<table>
<tr><th colspan="2">Inputs</th><th colspan="2">Outputs</th><th rowspan="2">Action</th></tr>
<tr><th>D</th><th>CLK</th><th>Q_{next}</th><th>Q'_{next}</th></tr>
<tr><td>0</td><td>↑</td><td>0</td><td>1</td><td>Reset (Q → 0)</td></tr>
<tr><td>1</td><td>↑</td><td>1</td><td>0</td><td>Set (Q → 1)</td></tr>
</table>
</td></tr>
<tr><td>Boolean Equation</td><td colspan="2">$Q_{next} = D$</td></tr>
<tr><td>Name Origin</td><td colspan="2">D for Delays, since it delays the signal until the next active clock transition</td></tr>
<tr><td>Observations</td><td colspan="2">• Made with S-R flip-flop with input S inverted for input R
• Stores a single bit after the edge-triggered clock pulse</td></tr>
<tr><td>Applications</td><td colspan="2">• Storing Bits (memory) in a pipeline
• Event Detection</td></tr>
<tr><td>TTL Chips</td><td colspan="2">74x74, 74x79, 74x171, 74x173</td></tr>
</table>



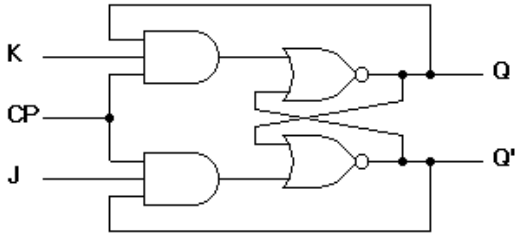
J-K Flip-Flop (Edge-Triggered)

Style

NAND-NAND

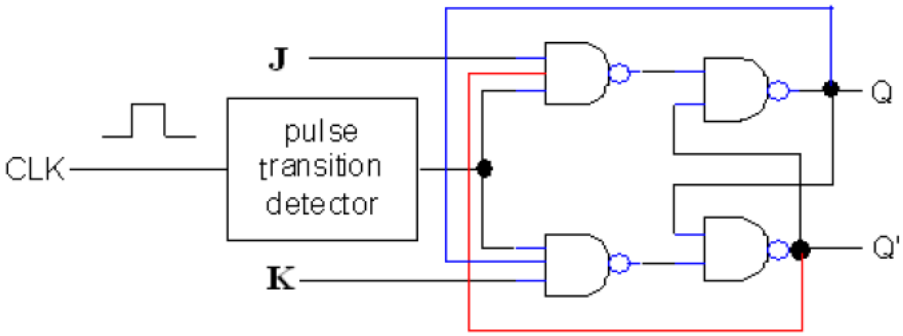


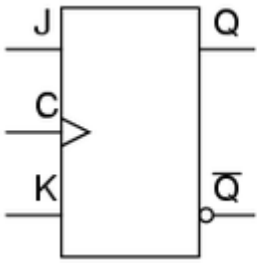
AND-NOR

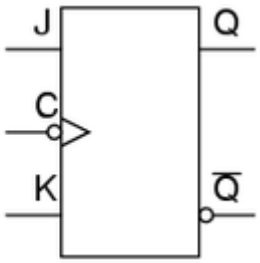


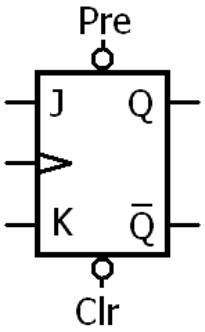
Circuit

Symbol









Truth Table

Inputs			Outputs		
J	**K**	**CLK**	**Q_{next}**	**Q'_{next}**	**Action**
0	0	↓	Q	Q'	Hold, No change
0	1	↓	0	1	Reset (Q → 0)
1	0	↓	1	0	Set (Q → 1)
1	1	↓	Q'	Q	Toggle, Change (1 → 0)

Boolean Equation

$$Q_{next} = J\bar{Q}_{prev} + \bar{K}Q_{prev}$$

Name Origin

None, other than J and K are adjacent letters in the alphabet

Observations

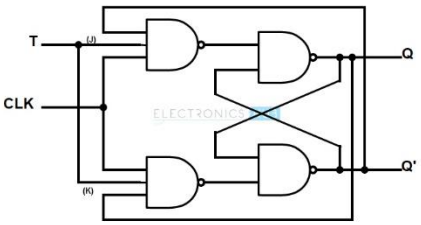
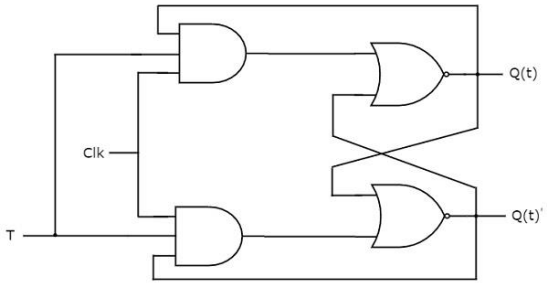
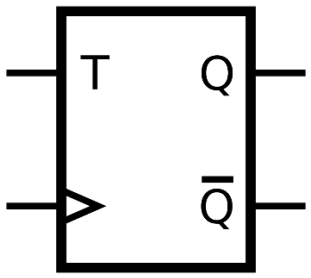
- Same as S-R flip-flop except 2 feedback lines added
- Fixes the invalid 1-1 state

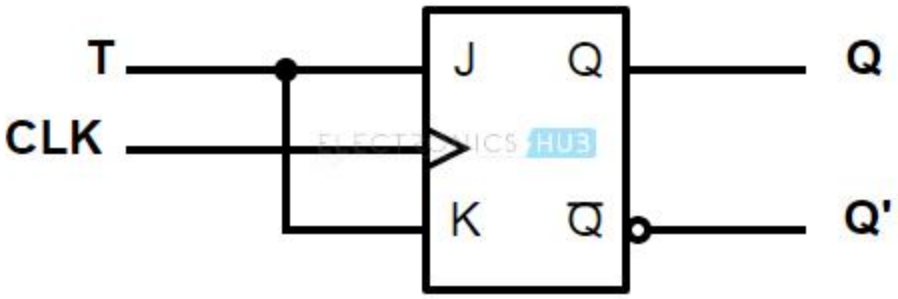
Applications

- Frequency Division: If J = K = HIGH, then clock frequency divider $\left({}^{f}/_{2}\right)$
- Counting: If cascaded with Q_A wired to JK_B CLK, then Q_A = LSB and Q_B=MSB
- Sequence Detection: If cascaded with $Q_A \rightarrow J_B$ and $Q'_A \rightarrow K_B$, then tap Q/Q's for 1/0 pattern, then AND for output

TTL Chips

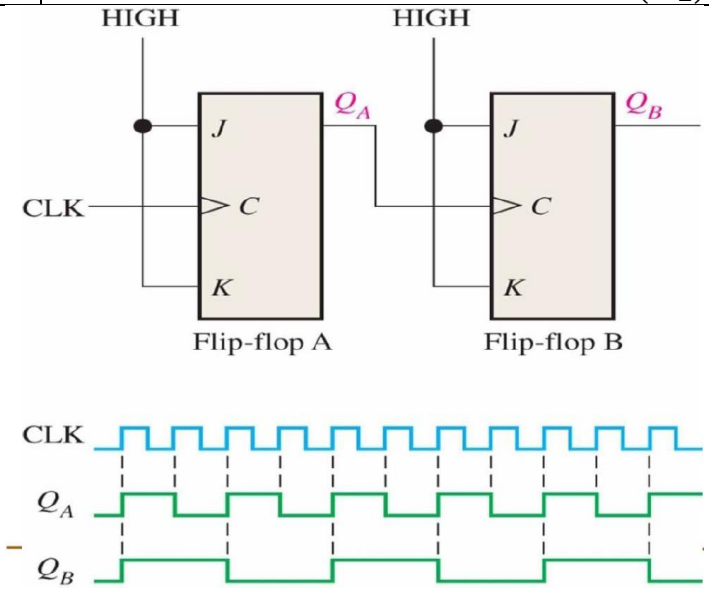
74x68, 74x69, 74x70, 74x73, 74x76, 74x101, 74x102, 74X103, 74x107

<table>
<tr><th colspan="3">T Flip-Flop (Edge-Triggered)</th></tr>
<tr><td>Style</td><td>NAND-NAND</td><td>AND-NOR</td></tr>
<tr><td>Circuit</td><td> 
</td><td> 
</td></tr>
<tr><td>Symbol</td><td colspan="2"> 

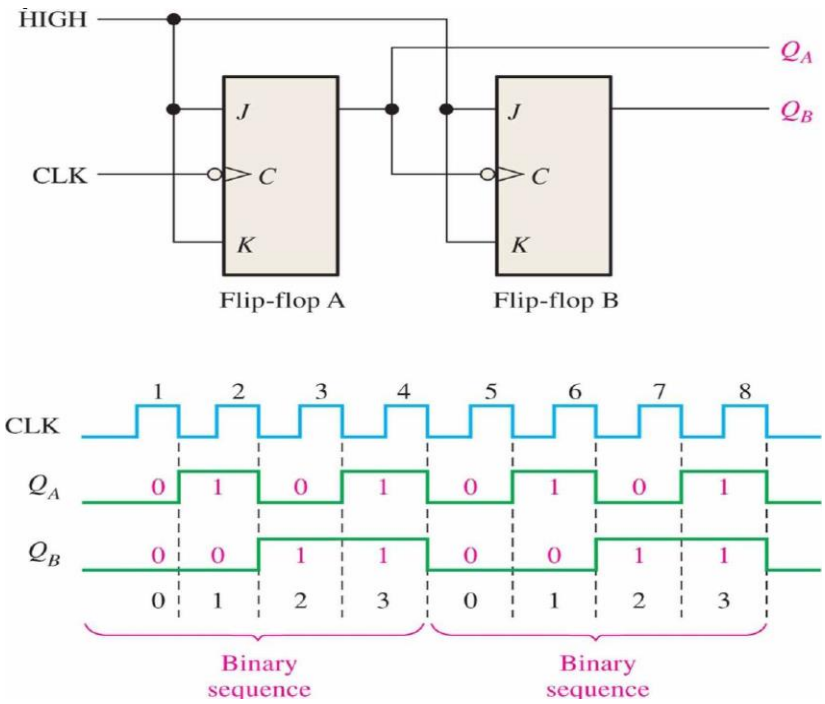

</td></tr>
<tr><td>Truth Table</td><td colspan="2">
<table>
<tr><th colspan="2">Inputs</th><th colspan="2">Outputs</th><th rowspan="2">Action</th></tr>
<tr><th>T</th><th>CLK</th><th>Q_{next}</th><th>Q'_{next}</th></tr>
<tr><td>0</td><td>↑</td><td>Q</td><td>Q'</td><td>No change, Hold</td></tr>
<tr><td>1</td><td>↑</td><td>Q'</td><td>Q</td><td>Change, Toggle</td></tr>
</table>
</td></tr>
<tr><td>Boolean Equation</td><td colspan="2">$Q_{next} = \bar{Q}_{prev}$</td></tr>
<tr><td>Name Origin</td><td colspan="2">T for Toggle, since it changes state on the triggering edge of the clock pulse</td></tr>
<tr><td>Observations</td><td colspan="2">• Made with J-K flip-flop with input T connected to both J and K
• Implements the two middle rows of the J-K flip-flop truth table</td></tr>
<tr><td>Applications</td><td colspan="2">• Frequency Division: If J = K = HIGH, then clock frequency divider $\left({}^{f}/_{2}\right)$</td></tr>
<tr><td>TTL Chips</td><td colspan="2">74x374 or use J-K flip-flop chips</td></tr>
</table>

J-K Flip-Flop Applications

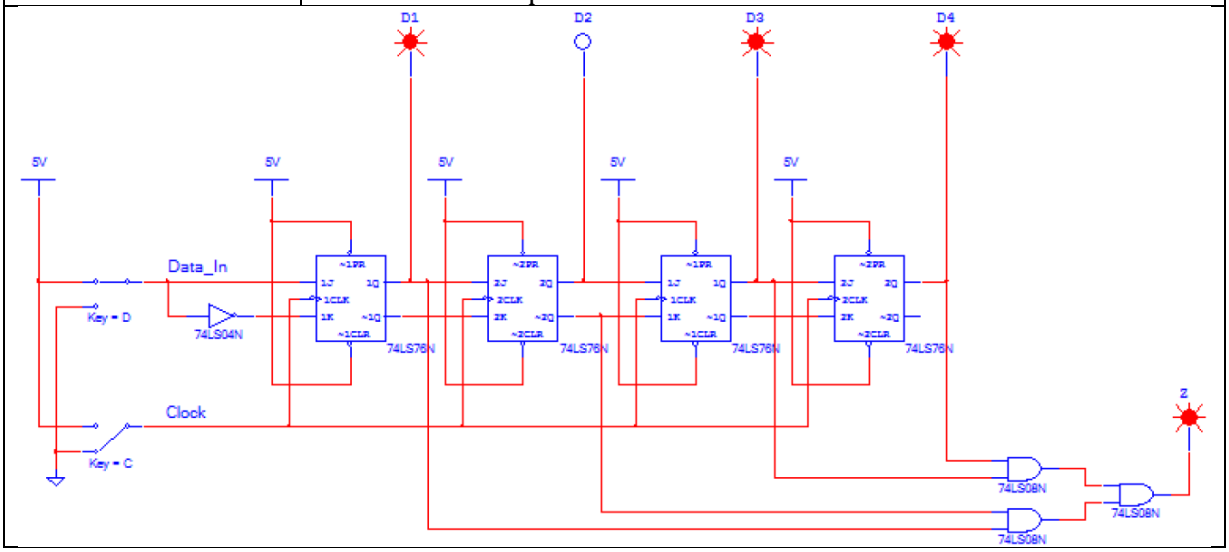
Frequency Division	If J = K = HIGH, then clock frequency divider $\left({}^{f}/_{2}\right)$



Counting	If cascaded with Q_A wired to JK_B CLK, then Q_A = LSB and Q_B=MSB



Sequence Detection	If cascaded with $Q_A \rightarrow J_B$ and $Q'_A \rightarrow K_B$, then tap Q/Q's for 1/0 pattern, then AND for output



Credit: Diagrams taken from "ECPI University EET 230 – Digital Systems II", Wikipedia, and Google images.

