FLIP-FLOPS AND LATCHES

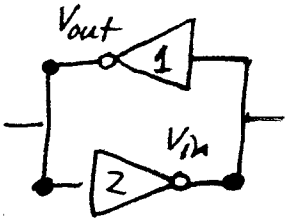
Definitions

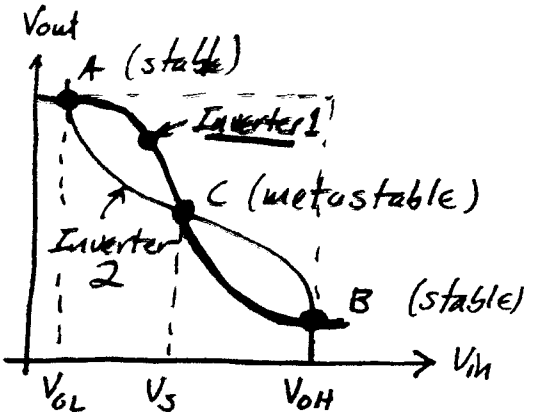
→ Combinational logic circuits – no feedback, output = f(inputs)

→ Sequential logic circuits – new outputs are dependent on the inputs and old outputs, uses positive feedback.

The **bistable** circuit is basic to sequential logic circuits.

Basic Bistable Circuit



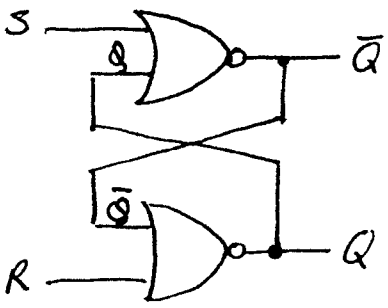


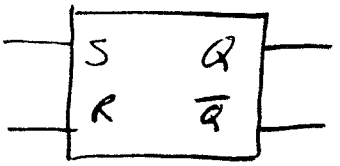
– A bistable circuit can be triggered to move from one state to another by moving past V_S for a duration of $2t_p$, where

$$t_p = \frac{t_{PHL} + t_{PLH}}{2}$$

→ In the absence of a trigger, the bistable remains in that state.

SR Latch with NOR Gates



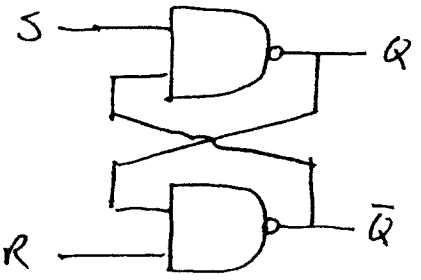


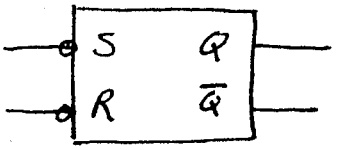
S R	Q Q̄	
0 0	Q Q̄	
0 1	0 1	(Reset)
1 0	1 0	(Set)
1 1	0 0	Ambiguity

Delay from $S \to \bar{Q}$ or $R \to Q$ is one NOR delay.

Delay from $S \to Q$ or $R \to \bar{Q}$ is two NOR delays.

SR Latch with NAND Gates





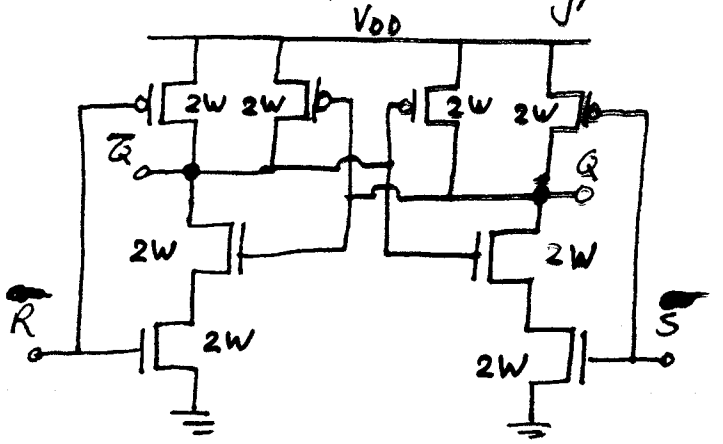
S R	Q $\bar{Q}$	
1 1	Q $\bar{Q}$	
0 1	1 0	(set)
1 0	0 1	(reset)
0 0	1 1	Ambiguity

S→Q & R→$\bar{Q}$ delay = 1 NAND delay

S→$\bar{Q}$ & R→Q delay = 2 NAND delays

Exercise 5.4

Implement the NAND-based SR latch using CMOS gates. Size the transistors to deliver 400 ps propagation delays when driving 100fF in 0.13μm technology. Let L=0.1μm



$t_{pHL} = t_{pLH} = 0.7 R_{eff} C_L$

Let $t_{pHL} = t_{pLH} = 200ps$

$t_{pHL} = 0.7(2 \times 12.5 k\Omega/\square)\frac{L}{W}(100fF)$

$\left(\frac{W}{L}\right)_N = \frac{0.7(25)100}{200} = 8.75 = 9\lambda$

$W_N = 0.875\mu m$

$t_{pLH} = 0.7(30 k\Omega/\square)\frac{L}{W}(100fF)$

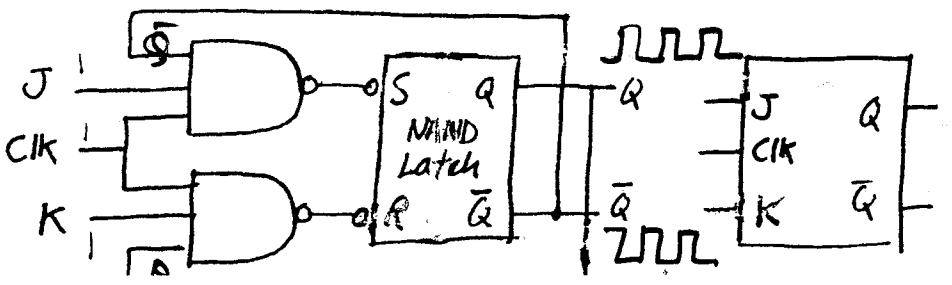
$\left(\frac{W}{L}\right)_P = \frac{0.7(30)(100)}{200} = 10.5$

$W_P = 1.05\mu m$

JK Flip-Flop

Two additional feedback lines remove ambiguity.

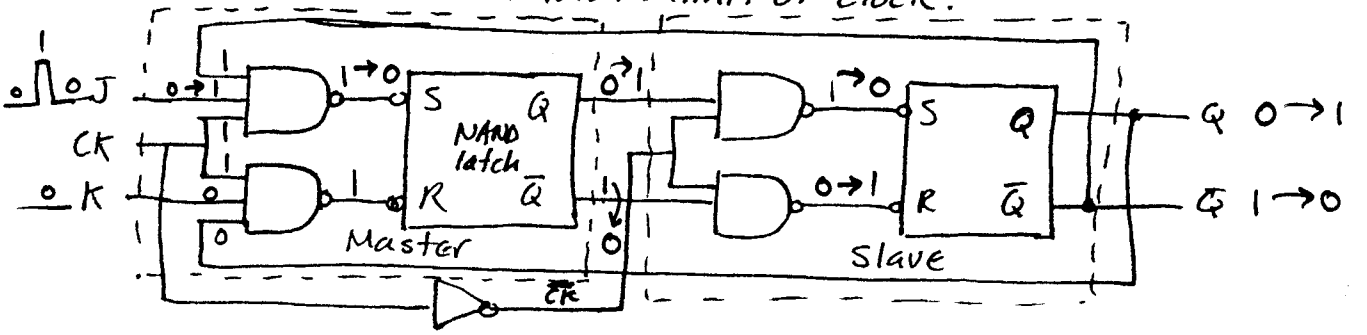
S & R → J & K



J_n K_n	Q_{n+1}	
0 0	Q_n	
0 1	0	(Reset)
1 0	1	(set)
1 1	$\bar{Q}_n$	

JK Master-Slave Flip Flop

Removes minimum width limit of clock.



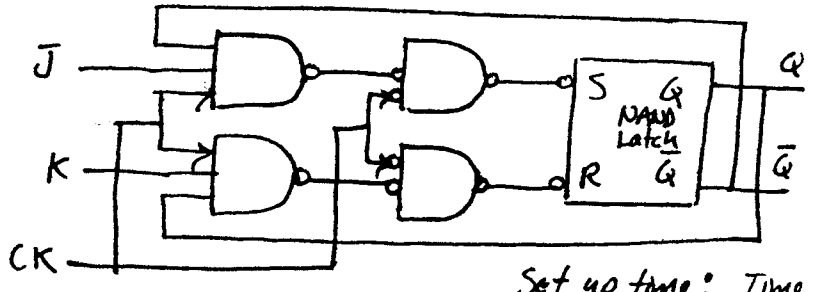
Operation:

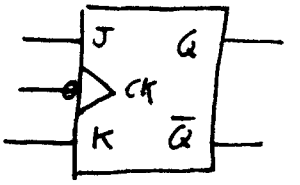
1. CK rises, $\overline{CK}$ goes down and disables the slave NAND gate.
2. CK enables the input NAND gates of the master.
3. The influence of J and K is entered into the master.
4. On the falling edge of CK, the master is disabled and the slave enabled
5. The state of the master is transferred to the slave

One-catching problem:

With CK high and the slave in the reset state, if the J input gate is enabled for a short while and returns to 0, the high value on J will propagate into the master and lock in place

JK Edge-Triggered Flip-Flop





J_n	K_n	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\bar{Q}_n$

Set up time: Time JK must be stable before CK.
Hold time: Time JK must be stable after CK

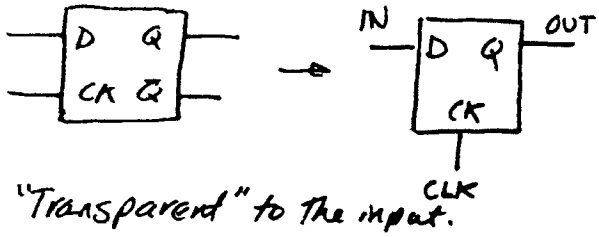
Operation:

1.) With CK high, entry into the input NAND gates is controlled by J & K. feedback lines. Entry into the NAND latch requires CK to go low.

2.) When CK goes low, the input NAND gates are disabled and the output of flip-flop changes due to the state of the J and/or K

D Flip-Flops and Latches (D-Flops)

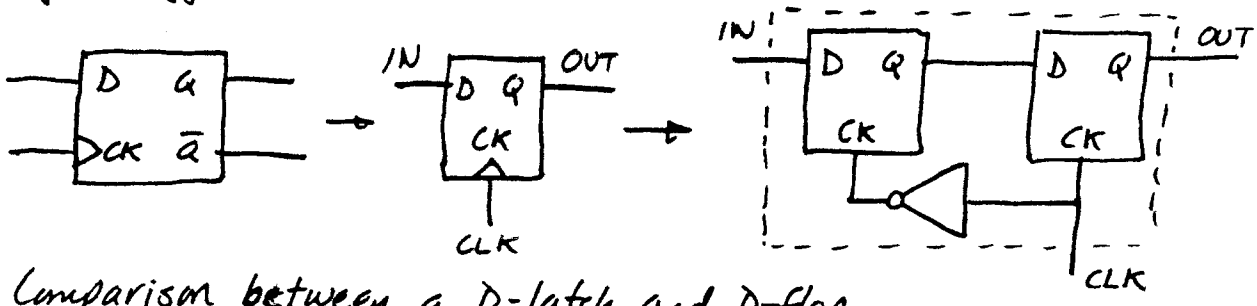
D-Latch –



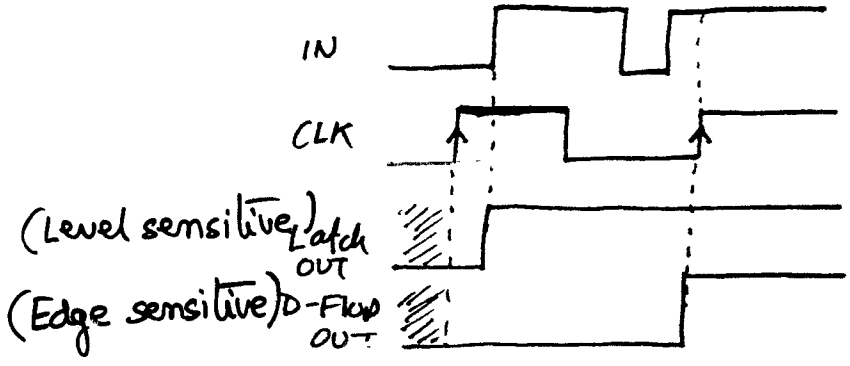
D	Q_{n+1}
0	0
1	1

"Transparent" to the input.

Edge-Triggered D-Flop –



Comparison between a D-latch and D-flop –



Timing Issues –

T_{setup} = time incoming data must be stable before the clock

T_{hold} = " " " " " " after " "

T_{ck-q} = delay from the time the clock arrives to the point when Q stabilizes

T_{d-q} = time it takes the data to propagate thru the D-latch

Implementation of a D-latch –

$Q = (D \cdot CLK) + \bar{Q}$