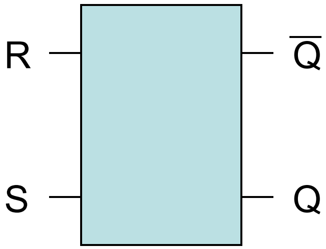
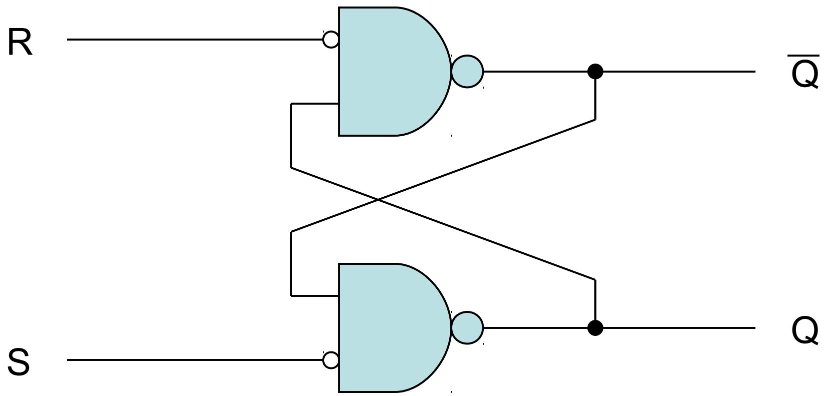
Latch and Flip-Flop roundup…

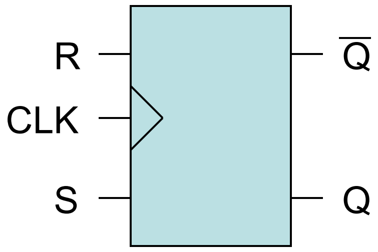
Standard RS Latch

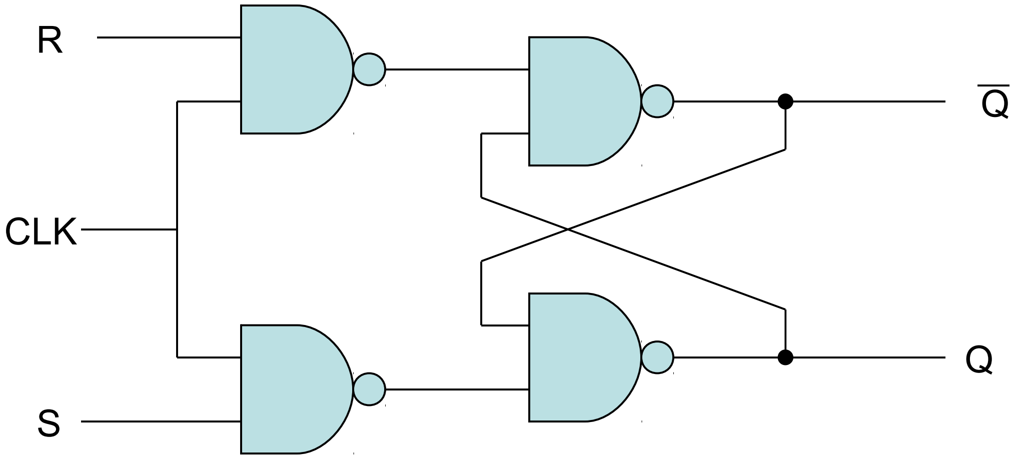


R	S	Q	$\overline{Q}$
1	0	0	1
0	1	1	0
0	0	Q	$\overline{Q}$
1	1	1	1 – not valid!



Latch and Flip-Flop roundup…

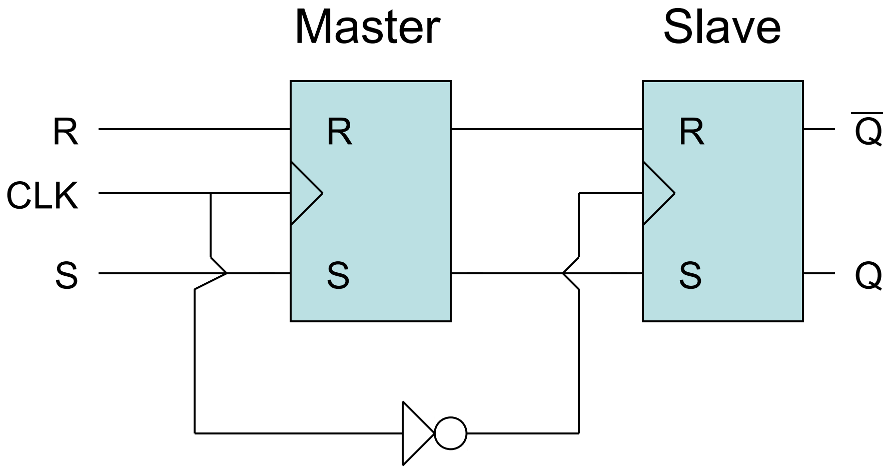




Clocked RS Latch

Output can only change when CLK=1

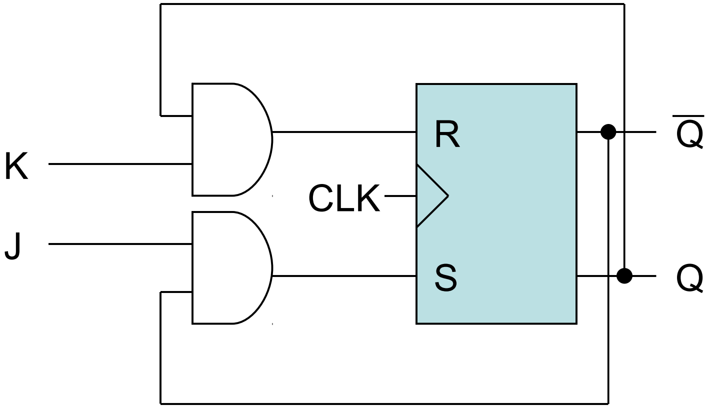
Latch and Flip-Flop roundup…



Edge-triggered R-S Flip-flop

Can only change output state during
the transition from CLK=1 to CLK=0

Latch and Flip-Flop roundup…

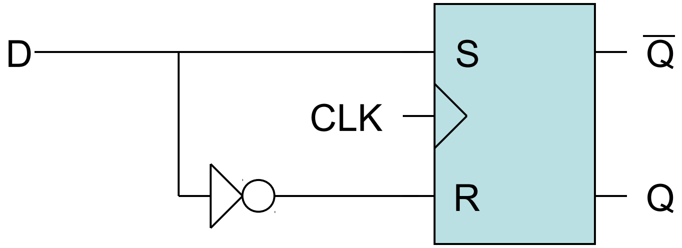


K	J	Q
0	0	Q
0	1	1
1	0	0
1	1	$\overline{Q}$

Standard J-K Flip Flip

Only one of the inputs to the R-S latch can be 1 at any given time, thus the invalid R=1, S=1 condition is resolved

Latch and Flip-Flop roundup…

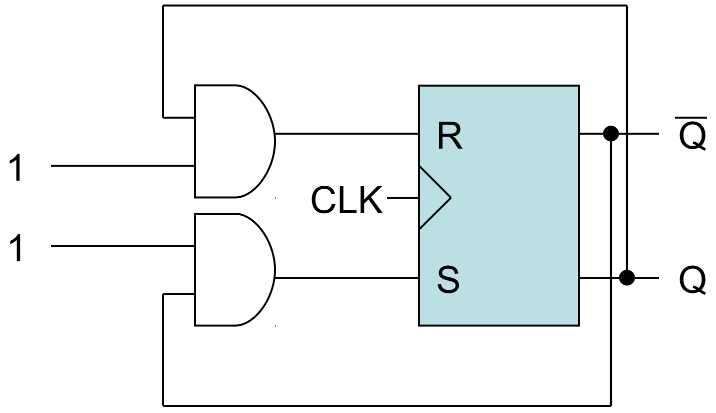


D	Q
1	1
0	0

D (data) Flip Flop

When CLK=1 data in the D line is reflected by Q, remains stored after CLK=0 regardless of D

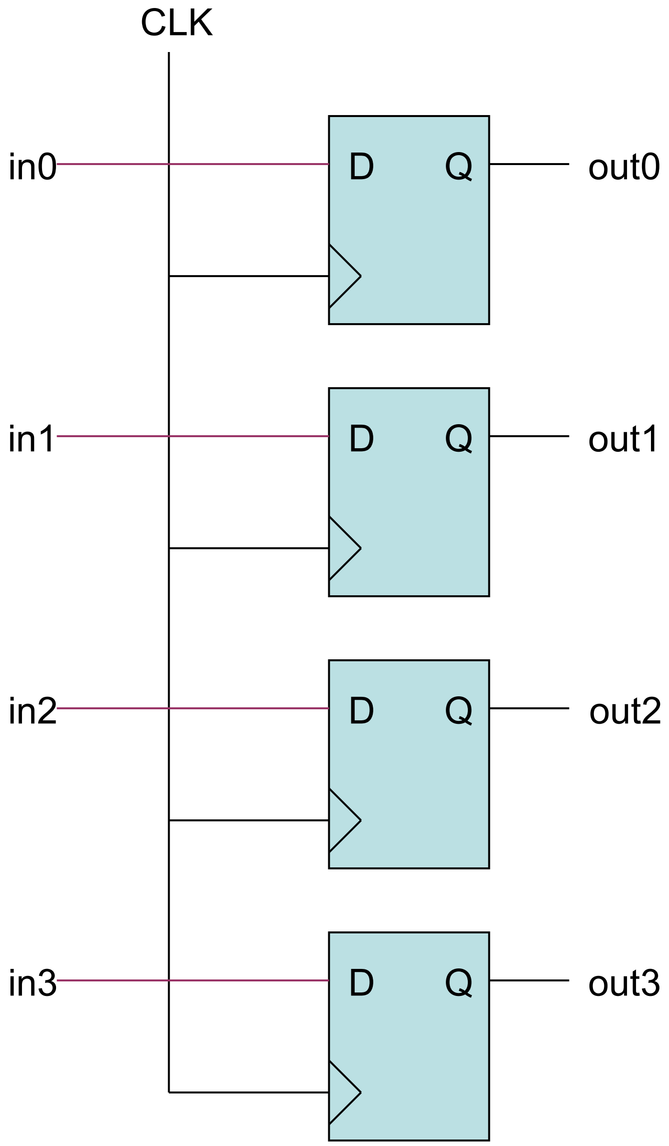
Latch and Flip-Flop roundup…



T (toggle) Flip Flip

Just a J-K with both inputs set to 1,
Output changes state at each clock pulse

Registers: Storing more than 1 bit of data

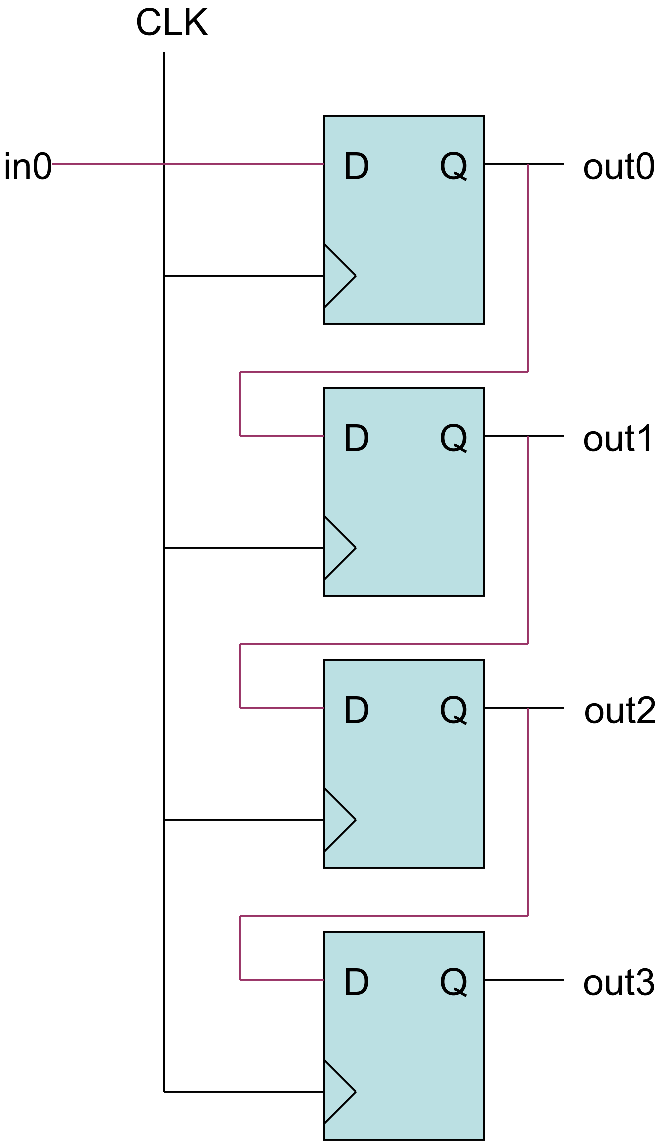


4 bit parallel-load register

Data is loaded on the clock pulse

You know already what to do if you want, say, 2 of these, or 4 of these, or 8 of these: i.e. you can build memory blocks!

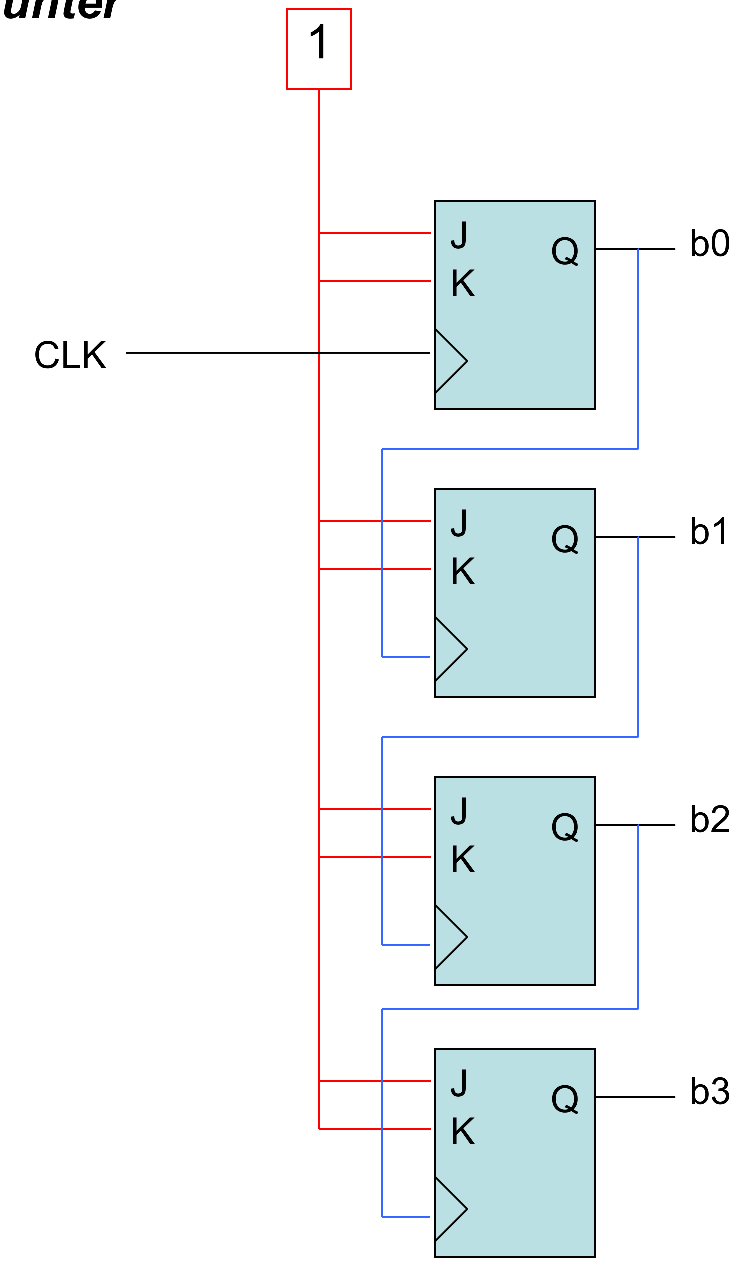
Registers: Storing more than 1 bit of data



4 bit shift register

Data shifts to the ‘left’ (in binary terms)
on each clock pulse

Simple counter



4 bit ripple counter

Counts the number of pulses
In CLK

Each individual output works
also as a frequency divider for CLK