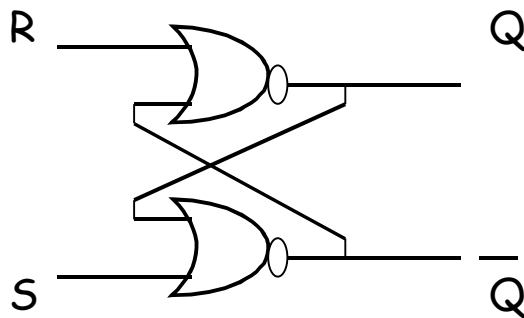


Sequential Circuits

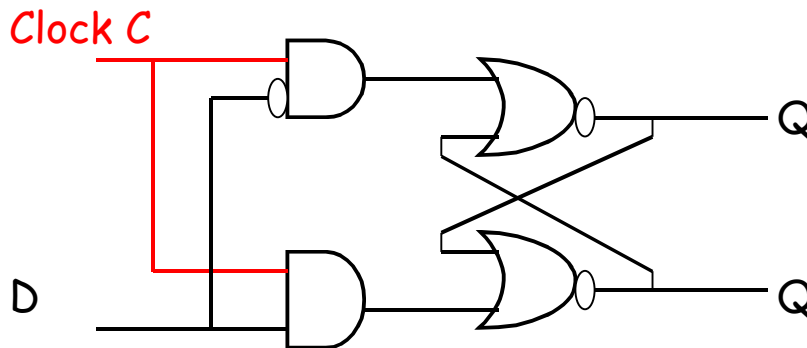
The output depends not only on the current inputs, but also on the past values of the inputs.



An SR Latch

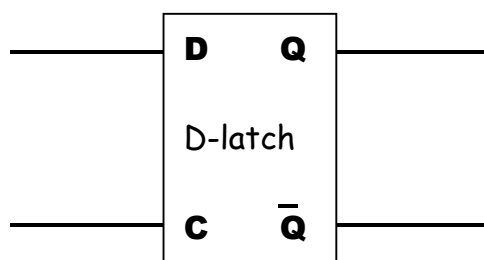
S	R	Q	\bar{Q}	Comment
0	0	0/1	1/0	Old state continues
1	0	1	0	Set state
0	1	0	1	Reset state
1	1	0	0	Illegal inputs

A clocked D-latch



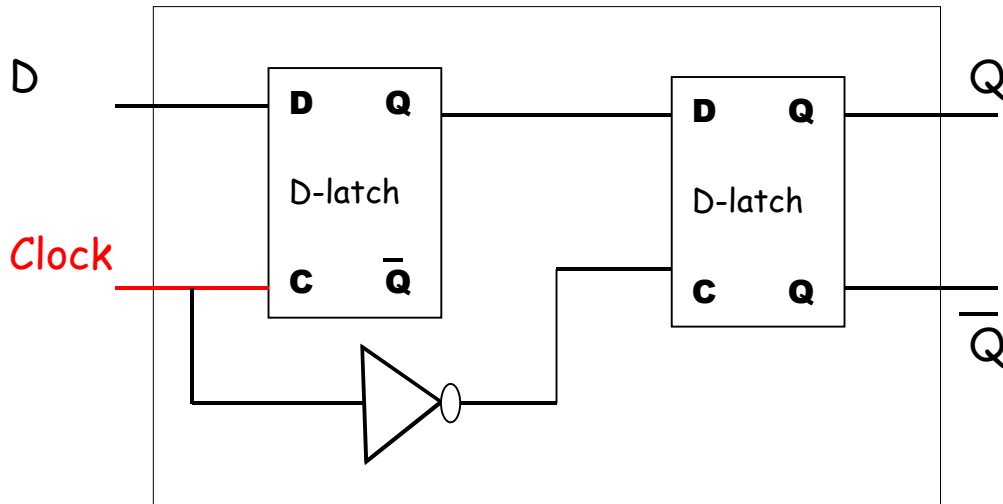
Clock is the enabler. If $C=0$, Q remains unchanged.

When $C=1$, then Q acquires the value of D . We will use it as a building block of sequential circuits.

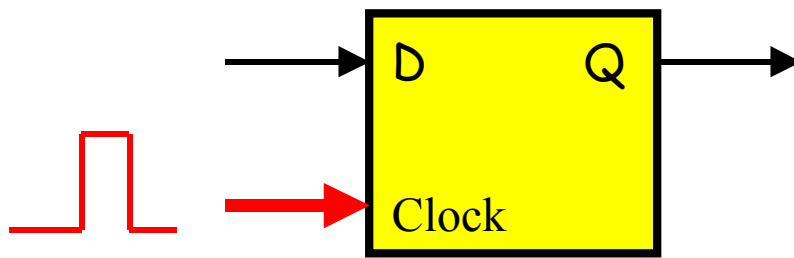


There are some shortcomings of this simple circuit. An **edge-triggered** circuit (or a **master-slave** circuit) solves this problem (to be discussed in the class)

Master-Slave D flip-flop



Internal details shown



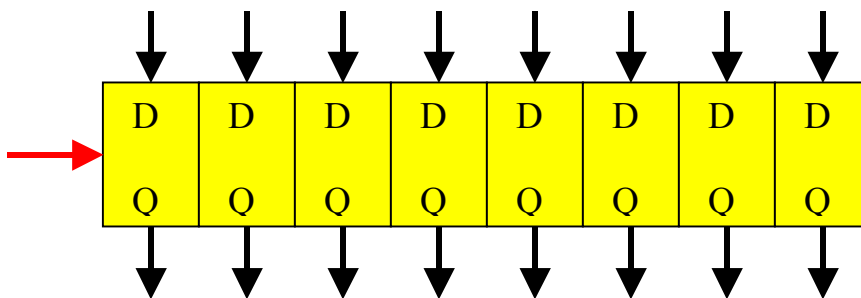
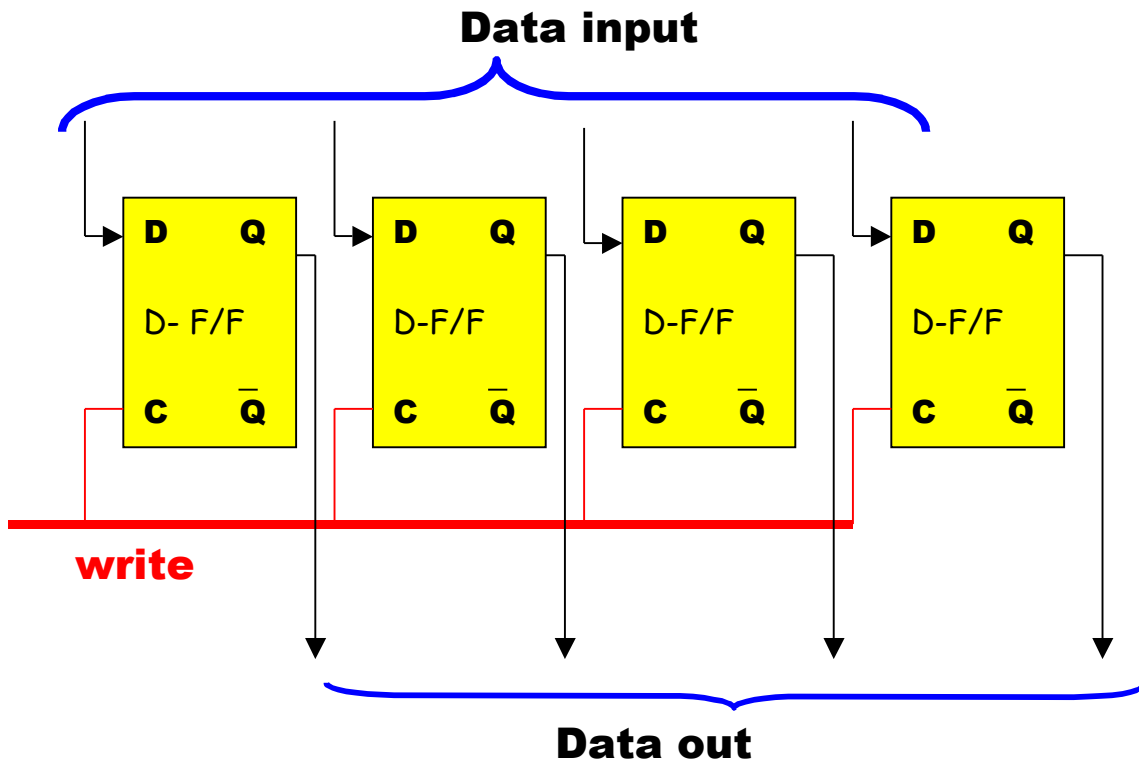
Clock pulse

Abstract view

The output Q acquires the value of the input D, only when **one complete clock pulse** is applied to the clock input.

Register

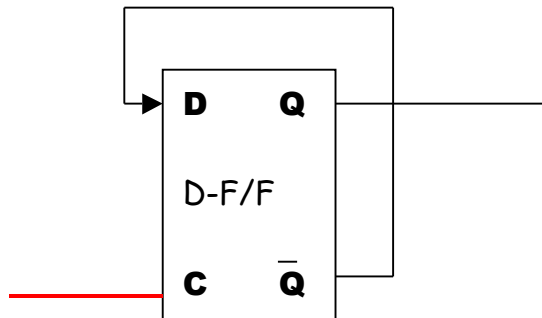
A 8-bit register is an array of 8 D-flip-flops.



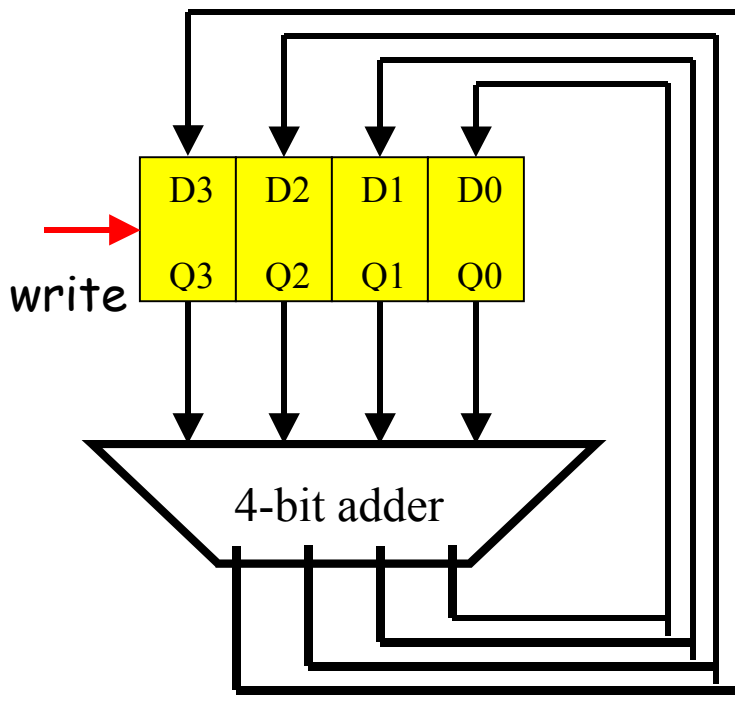
Abstract view of a register

Binary counter

Counts 0, 1, 2, 3, ...



A toggle flip-flop (T) is a **modulo-2 counter**

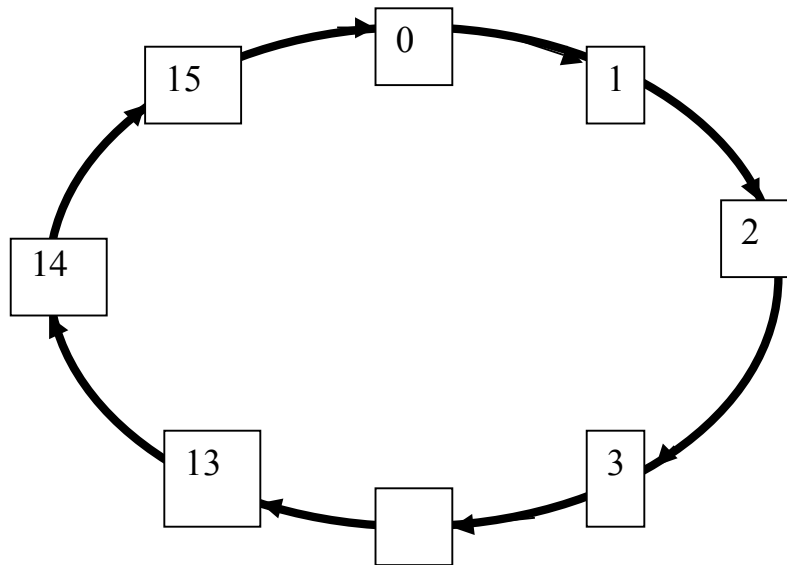


A 4-bit counter
(mod-16counter)

Observe how Q3 Q2 Q1 Q0 change when pulses are applied to the clock input

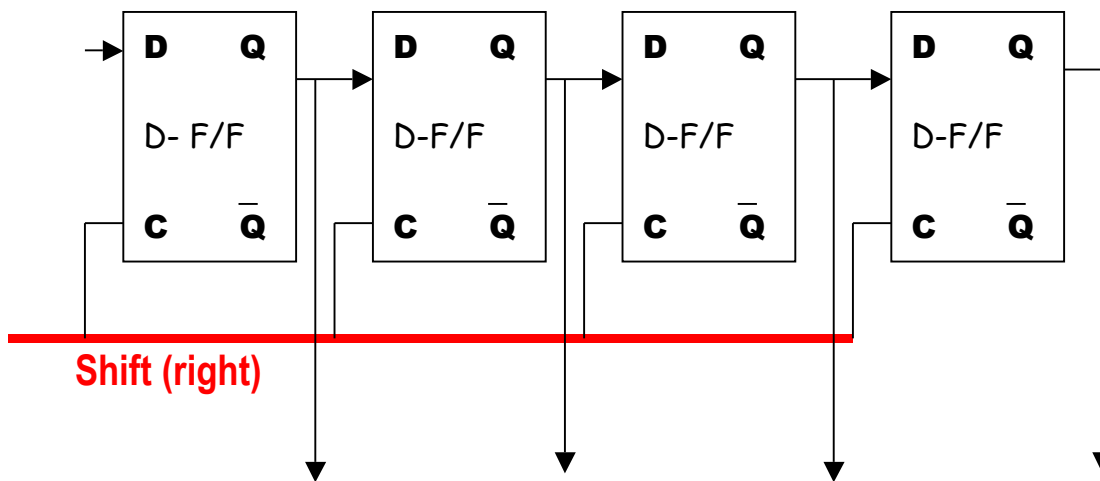
State diagram of a 4-bit counter

Here state = $Q_3Q_2Q_1Q_0$



Recall that the program counter is a 32-bit counter

A shift register



With each pulse on the shift input, data moves by pulse to the right.