

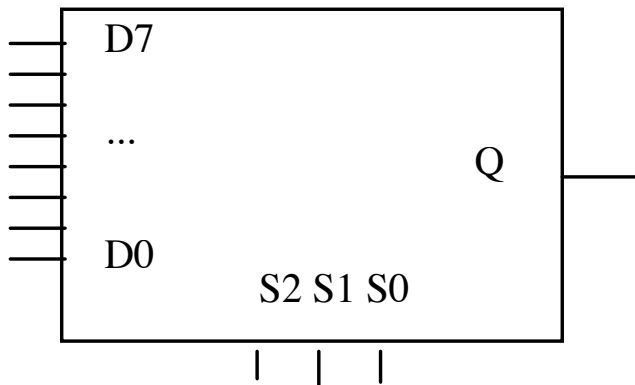
CS240 Laboratory 4
Basic Digital Circuits
and
Introduction to Memory

- **Multiplexer**
- **Decoder**
- **Arithmetic Logic Unit (ALU)**
- **Latches and Flip-Flops**

Multiplexer

- n select lines
- 2^n input lines
- 1 output

One of the possible 2^n inputs is chosen by the n select lines, and gated through to the output of a multiplexer.

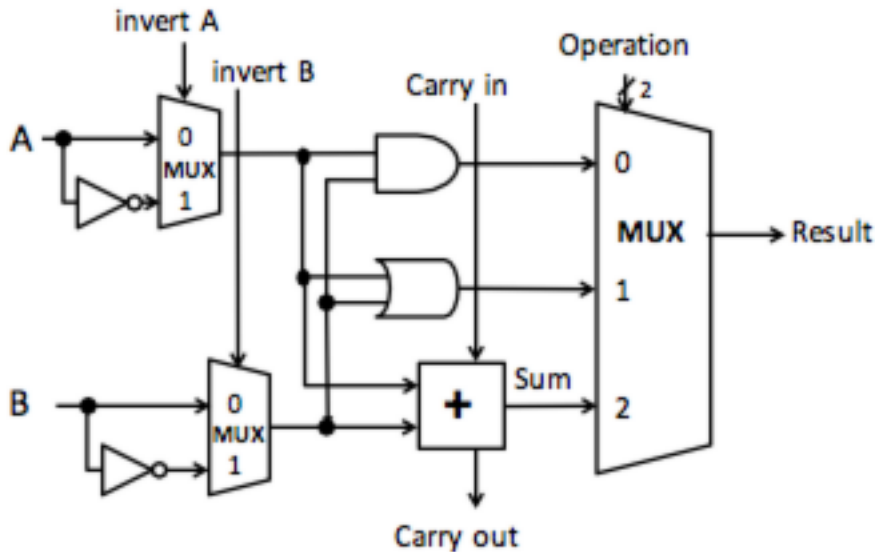


<u>S2</u>	<u>S1</u>	<u>S0</u>	<u>Q</u>
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

Multiplexers are usually used for **selection**, but can also act as code detectors.

Arithmetic Logic Unit

Want to be able to select whether the ALU will produce the bitwise AND, OR, and sum as a result.



The basic operations and results are:

add ($a + b + C_{in}$),

AND ($a \text{ AND } b$),

OR ($a \text{ OR } b$),

Adding the ability to choose whether to invert A or B provides additional operations:

sub (invert b, $C_{in} = 1$, $a + b + C_{in}$)

NOR (invert a, invert b, $a \text{ AND } b$)

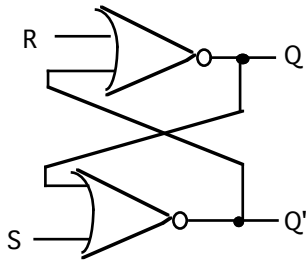
invA	invB	Cin	Op1	Op0	Result
0	0	X	0	0	a AND b
0	0	X	0	1	a OR b
0	0	0/1	1	0	a + b
0	1	1	1	0	a - b
1	1	X	0	0	a NOR b

Basic Memory Circuits

Latch Single-bit memory, level-triggered

Flip-Flop Also single-bit, but edge-triggered

SR (Set Reset) Latch



<u>S</u>	<u>R</u>	<u>Q</u>	<u>Q'</u>	
0	0	Q _p	Q _p '	remember
0	1	0	1	reset (clear)
1	0	1	0	set
1	1			unpredictable

What does **unpredictable** mean? Notice in a NOR gate, if either input = 1 to a gate, its output = 0 (1 is a deterministic input):

<u>A</u>	<u>B</u>	<u>(A+B)'</u>
0	0	1
0	1	0
1	0	0
1	1	0

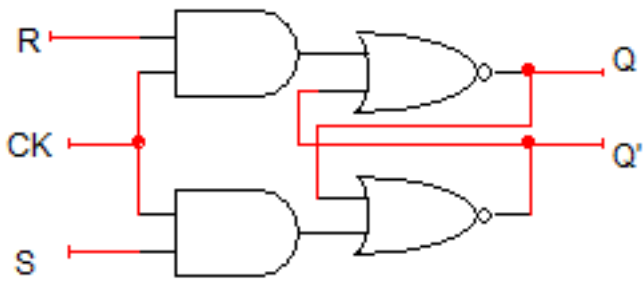
So, although you wouldn't usually try to *set* and *reset* at the same time (it doesn't make sense), if you did, Q and Q' will both be 0 (which is not unpredictable).

However, when you go back to the *remember* state (S=R=0), Q and Q' will not stay at 0 0. The circuit passes through one of either the *set* or *reset* state on its way back to the *remember* state, and Q and Q' change to the complement of one another.

Since the final state depends on which transitional state was sensed on the way back to *remember*, you cannot predict whether the final state of Q will be 1 or 0.

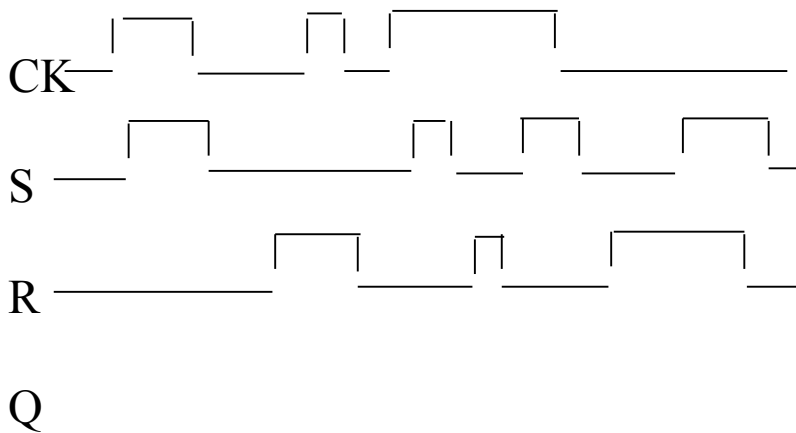
Clocked SR Latch

Incorporates a clock input/level-sensitive



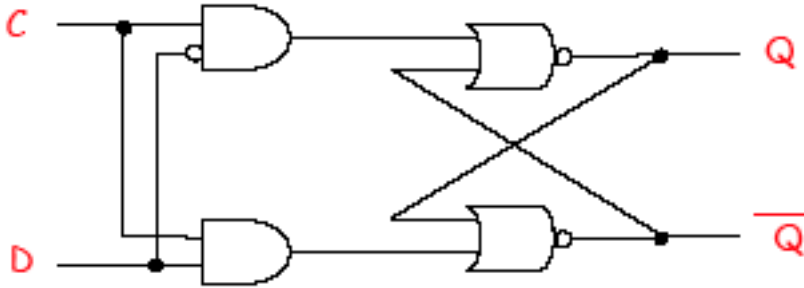
Output Q can change in response to S and R whenever the CK input is asserted.

How does Q respond to the following inputs?



D Latch

Avoids unpredictable state, because a single input D determines the next state of the circuit.

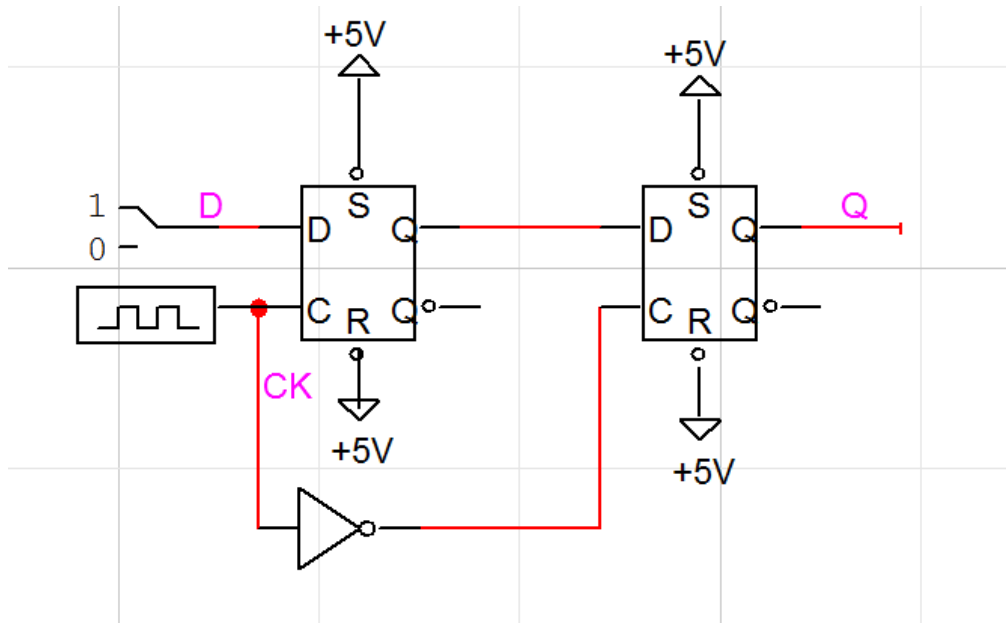


<u>D</u>	<u>Q_{next}</u>
0	0
1	1

D Flip-Flop

Changes state on a clock transition (edge), rather than whenever the clock is asserted.

Internally, a flip-flop is made from 2 latches. The first latch is controlled by the clock, but the second latch is controlled by the *inverse* of the clock:

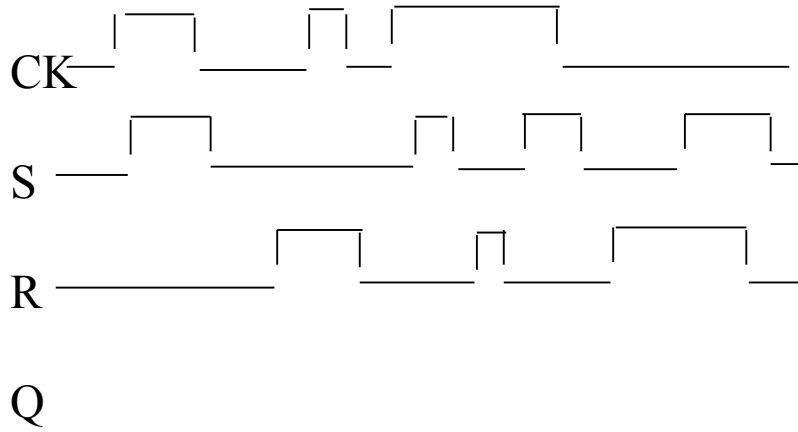


So, the input D will not be passed from the first latch to the second latch until the clock goes low.

Once the clock is low, a new value on D will not store into the first latch. Overall, the flip-flop can change value only *exactly* at the transition of the clock from high to low.

Output Q can change in response to S and R only on the positive edge of the clock.

How does Q respond to the following inputs?



Notice the difference between Q and the output for the earlier clocked latch example.