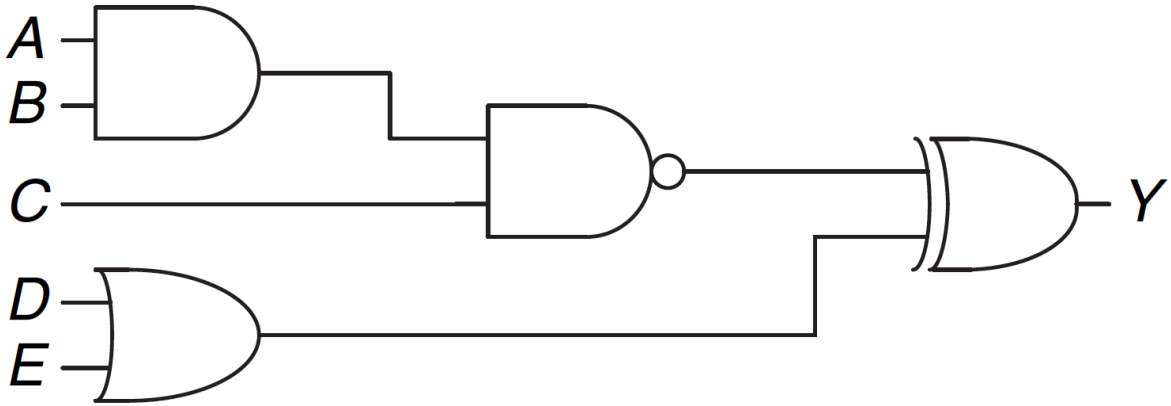
Sequential Logic

Based on slides by Jared Moore

Recall:

Up to now we have talked about combinational logic.

Outputs are dependent upon the the inputs and their flow through logic gates



Sequential Logic

Outputs depend on current and ***prior*** input values.

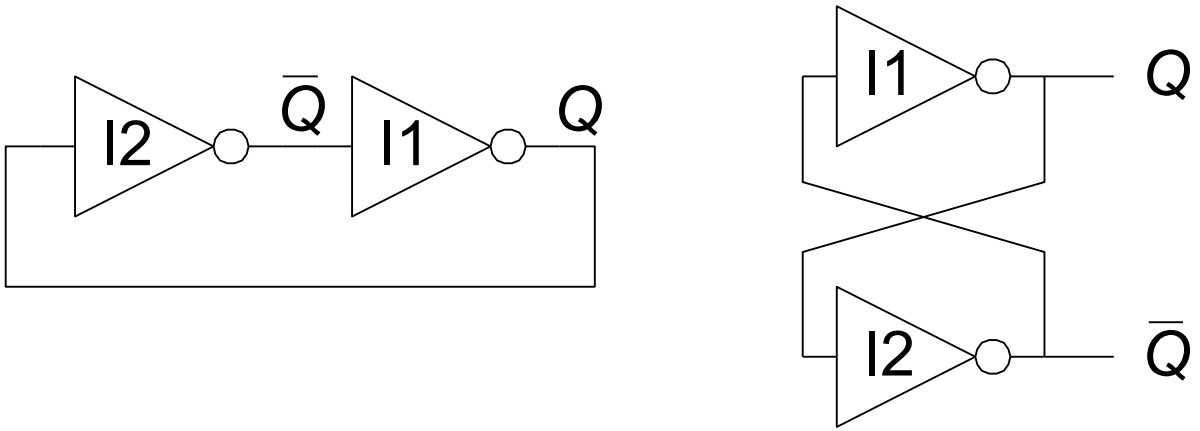
Sequential circuits have *memory.*

Sequential Circuits Can:

- Remember inputs
- Remember information about the system.

Basic Building Blocks

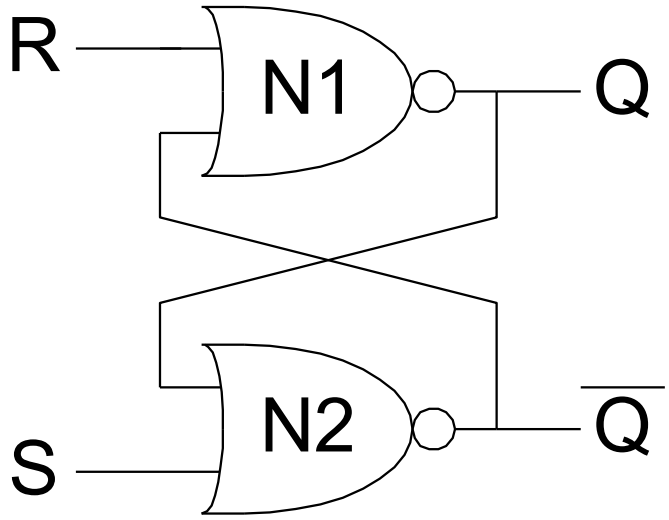
Bistable element: an element with two stable states



SR Latch

Two inputs: S(set) and R(reset)

Two outputs: Q and Q’



Determining SR Latch Behavior

Truth tables don't work for sequential circuits because outputs not dependent only on inputs

We can kind of make truth table, but need to consider two things:

1. Keep tracking until circuit reaches *steady state*
2. Need to consider current outputs as "inputs" for next state (remember – circuit has memory now!)

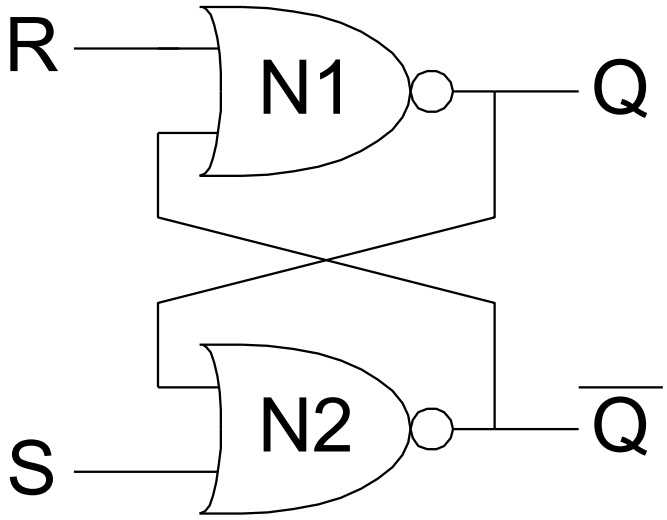
Allows us to build *characteristic table*

SR Latch

Assume R=1, S=Q=1, and Q' = 0

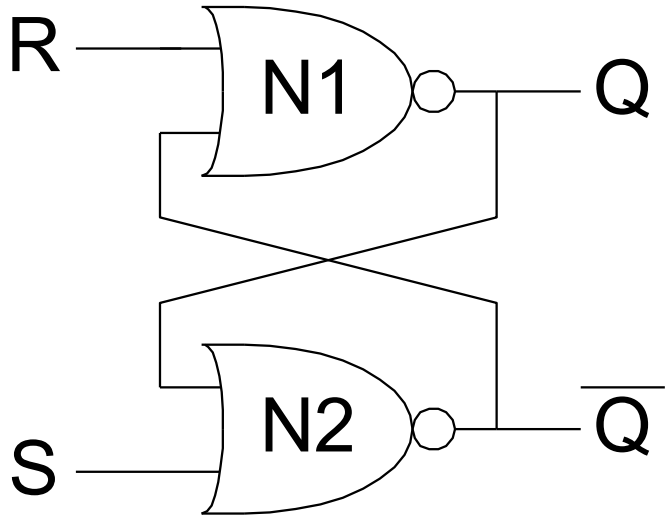
R and S are determined by user – they stay the same until deliberately changed

What is steady state of Q and Q'?



SR Latch

R	S	Q	Qnext	Q'next
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

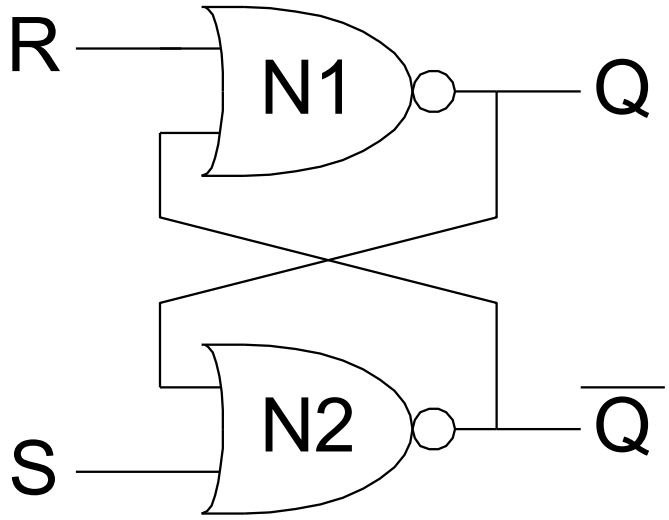


Assume initial Q' is opposite of initial Q

SR Latch

What about Q'? Why not include that as input to circuit?

Ans: In most circuits, we would. Because this one is special, as we'll see shortly, we assumed Q and Q' were opposite to begin with



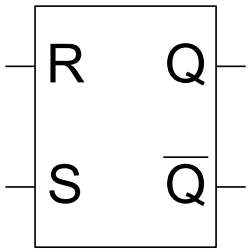
SR Latch Behavior

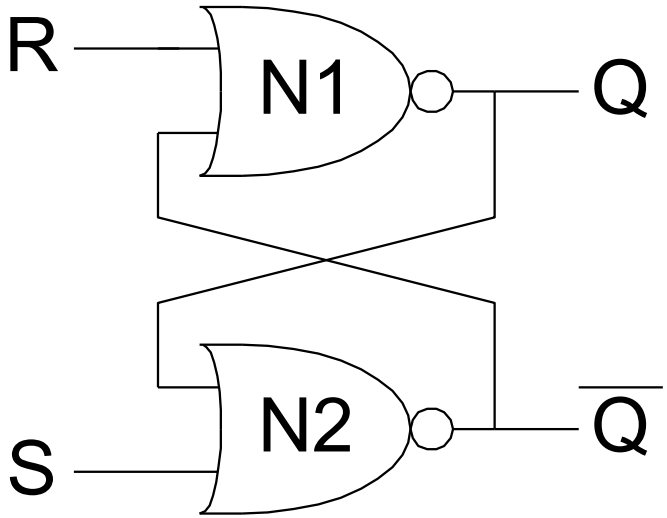
Set makes Q = 1

Reset makes Q = 0

Have the ability to set a piece of data (Q) with two control lines.

Q is state information about the latch





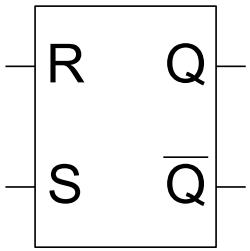
SR Latch Behavior

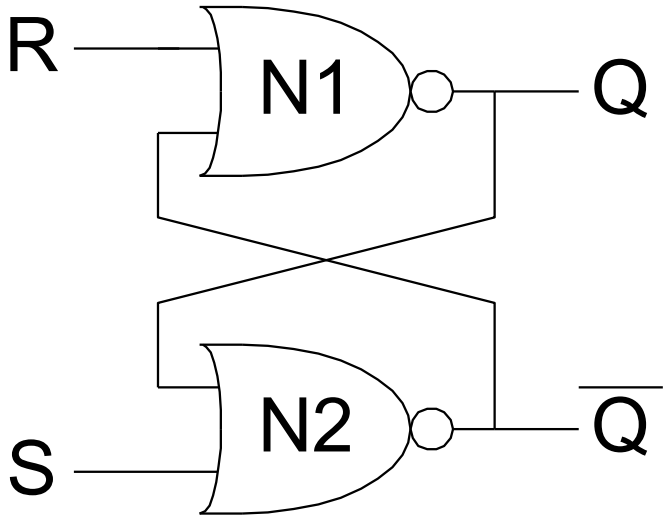
Set makes Q = 1

Reset makes Q = 0

What about pressing both at once?

Don't do that





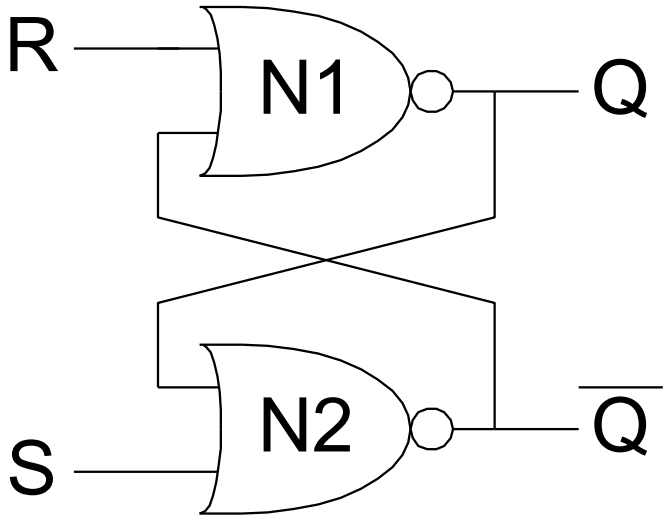
Improve SR Latch?

SR latch holds state only as long as R and S are both 0.

What if there is a glitch in the logic for R or S?

Why two bits of input controlling just one bit of output?

Want a way to control *when* memory changes

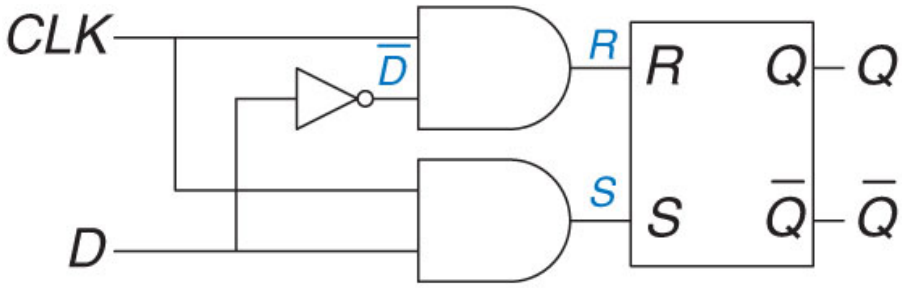


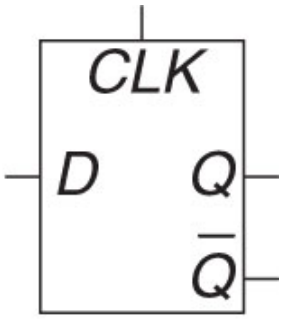
D Latch

Two Inputs:

- D – data
- CLK – clock

Key: Add logic to the front end of an SR Latch.





D Latch Behavior

CLK controls when the latch can be updated.

D controls how the latch is updated.

- Set or reset.

What happens when CLK == 1?