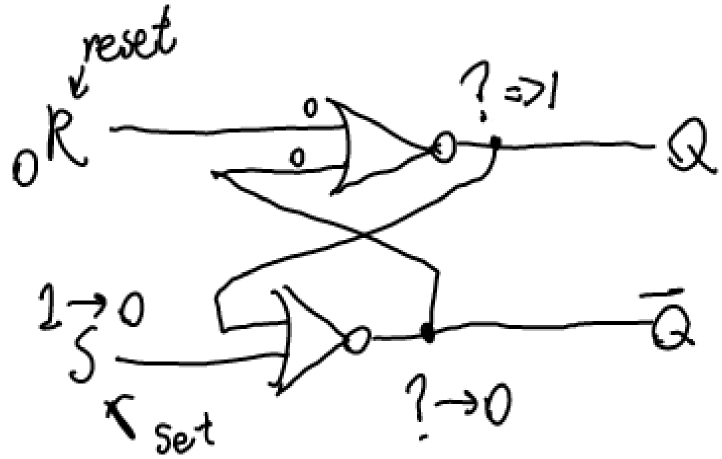
Latches - one type of memory element.

SR-Latch

0	0	1
0	1	0
1	0	0
1	1	0

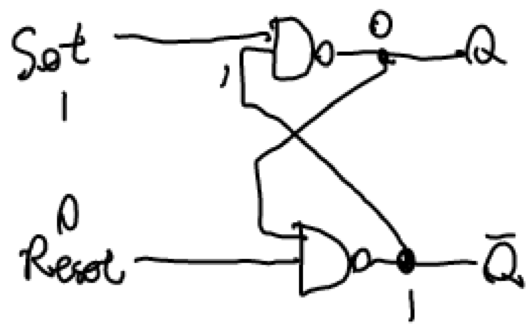


How does this work + how does it have "memory"

S	R	Q	$\overline{Q}$	
1	0	1	0	⇐ "set"
0	0	1	0	⇐ "hold" (after SR=10)
0	1	0	1	⇐ "reset"
0	0	0	1	⇐ hold" (after SR=01)
1	1	0	0	⇐ undesirable

* Can set Q=1, can reset Q=0, can hold previous value.

$\overline{S}\,\overline{R}$ - Latch



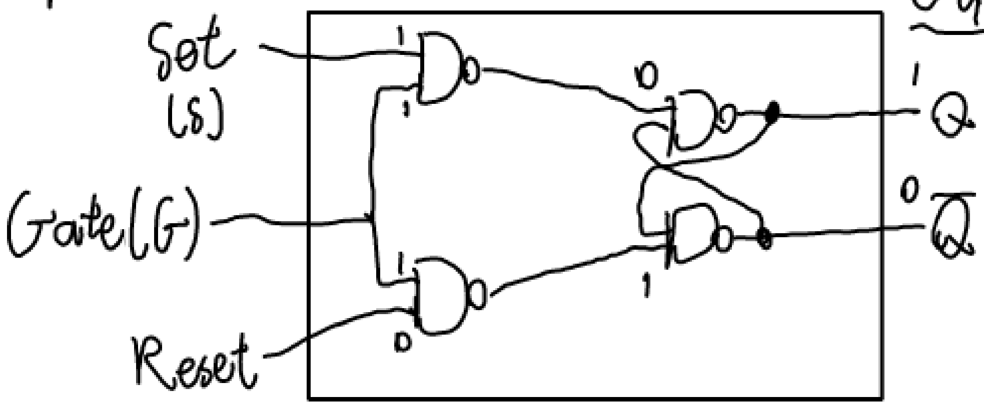
0	0	1
0	1	1
1	0	1
1	1	0

S	R	Q	$\overline{Q}$	
1	0	0	1	reset
1	1	0	1	hold
0	1	1	0	set
1	1	1	0	hold
0	0	1	1	← Undesreadable

Gated Latch

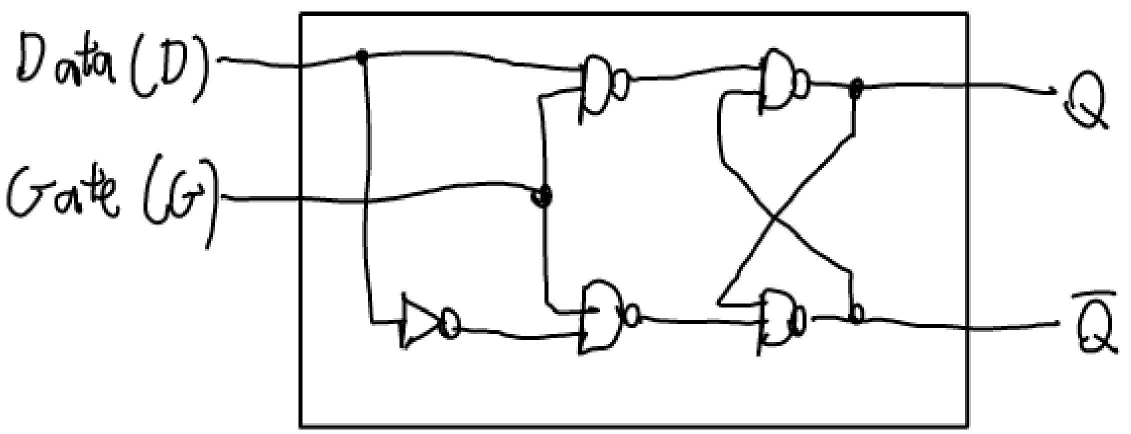
* Add a "gate" to control whether or not the Latch operates.

Gated SR Latch



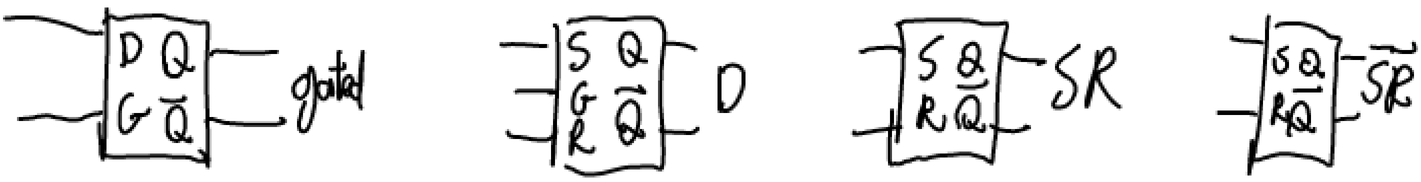
Gate	Set	Reset	Q	$\overline{Q}$	
1	1	0	1	0	Set
1	0	1	0	1	reset
1	0	0	previous value		hold
1	1	1	1	1	undesreable
0	X	X	previous hold		

D-Latch



Gate	Data	Q	$\overline{Q}$	
1	1	1	0	set
1	0	0	1	reset
0	X	prev. val.		hold

* first set data
wait a bit — (let values become stable)
change the gate *



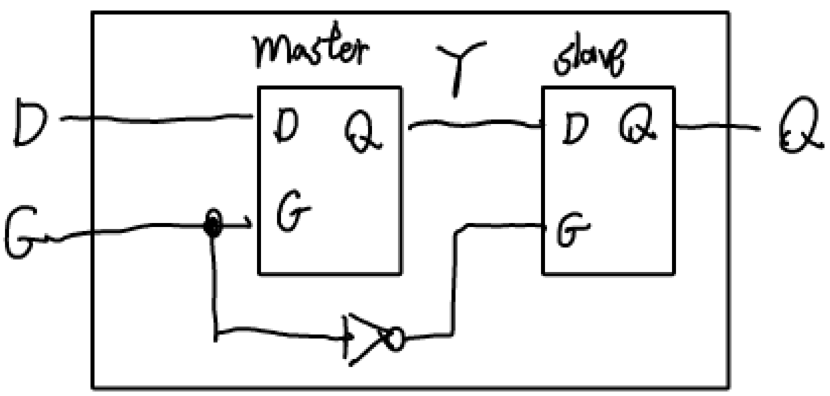
Latches are "level sensitive" memory elements.

Q: When can the output change?

* Output changes whenever G=1 (i.e., on the "level of G").
* Would like a bit more control as to when Q can change.
    * The "interval" should be an "instant".

Flip Flops

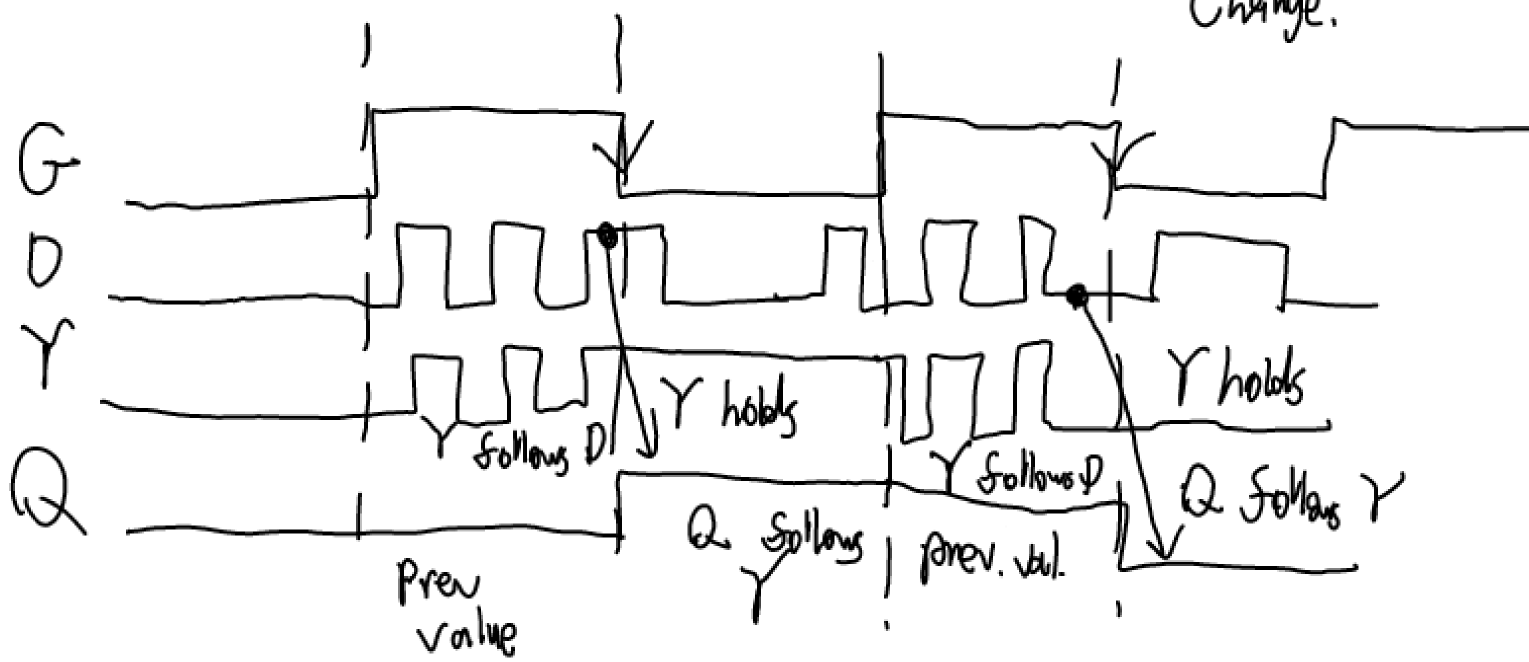
Master-Slave FlipFlop



when G=1
Y=D (Y follows D)
Q=Holding

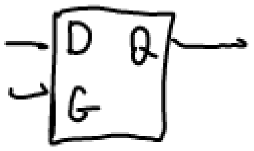
Change G from 1 to 0.
Y=holding the value of D just before G changed

Q=Y (D follows Y)
= D just before change.

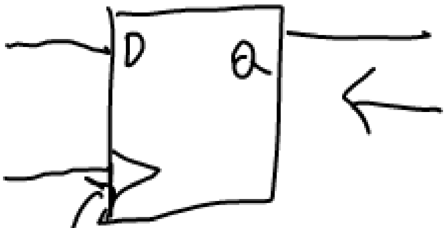


⇒ Edge Sensitive | Negative-Edge Triggered flipflop

Symbol:



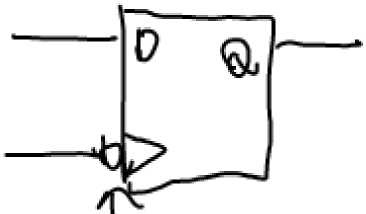
D-Latch
"Level sensitive"



← +ve edge triggerd

D-flipflop

means edge triggerd



-Ve edge triggered

Gate

get data ready

store instantly