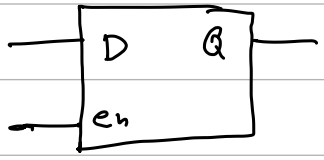
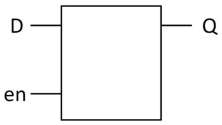
Reading 9

LATCH Video

- **Basic Latch!**

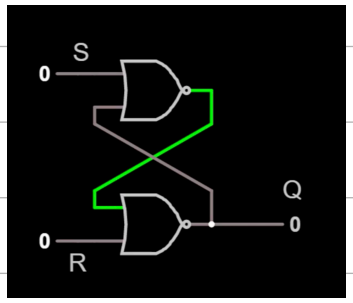


- if en is 1, then Q is whatever D is.
- if en is 0, then Q is whatever D was when en was flipped to 0.
  - ↳ Now even if you change D, Q does not change.

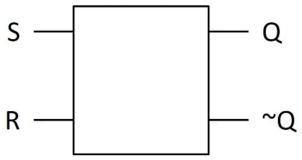


en	D	Q
0	0	hold
0	1	hold
1	0	0
1	1	1

- **SR Latch**



- S is the set control input
- R is the reset control input

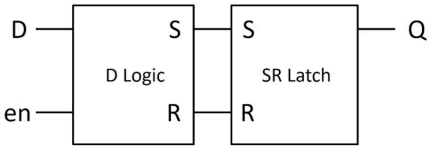


S	R	Q	~Q
0	0	hold	hold
0	1	1	0
1	0	0	1
1	1	0	0

← Mostly want to avoid this situation

D-Latch

- Appending some D logic to an SR latch.



en	D	Q	S	R
0	0	hold	0	0
0	1	hold	0	0
1	0	0	0	1
1	1	1	1	0

S = en D

R = en D'

- Note: A circuit with an odd number of inverters in a loop will oscillate
- Note! An even number of inverters in a loop will take on a stable value of 1 or 0.

FLIP FLOP Video

- Latches are problematic
  - Since they cause continuous oscilation when 'enable = 1' in some circuits.

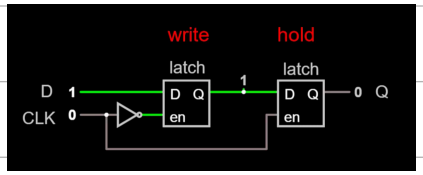
- Flip Flops help resolve this issue.
  - They have clocks instead of enables.

- Analogy: you want to carry in groceries, but dog wants to go out.
  - If you only have one door its a latch (the dog always gets out)
  - If you have 2 doors (i.e. utility room), you can get groceries in without letting dog out. This is the behavior of a flip flop.

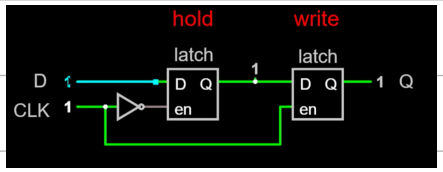
- You can build a flip-flop from from a pair of latches:
  - ↳ First latch enabled when clock is low (flip)
  - ↳ Second latch enabled when clock is high (flop)

- Diagram (Sequence):

1.

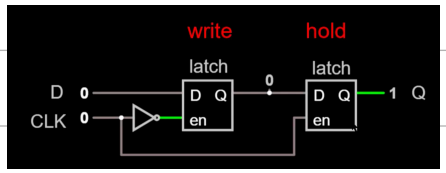


2.



→ here even if D=0, the midpoint = Q = 1

3.



- Note: An individual latch in a flip-flop can go transparent but the entire flip flop will never go transparent.