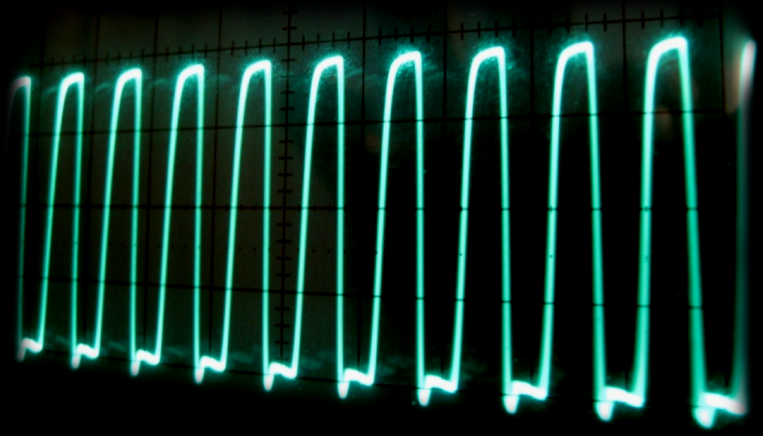
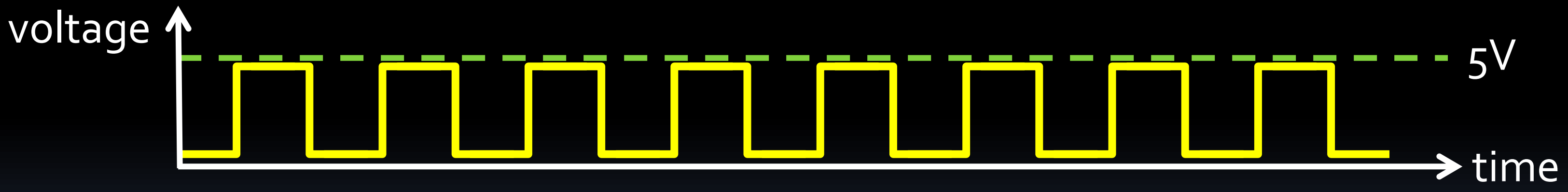
Lecture 4: Sequential Circuits Continued

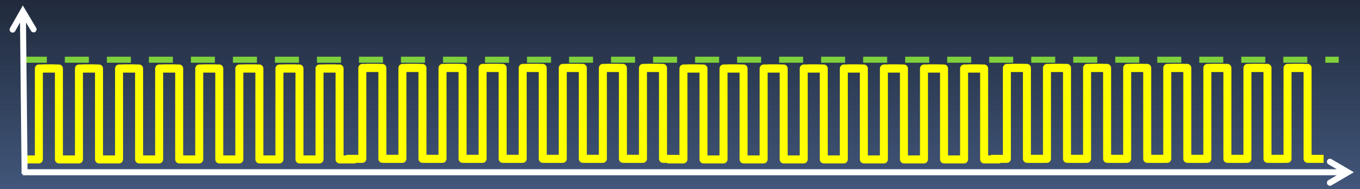
Clock signals



- "Clocks" are a regular pulse signal, where the high value indicates that the output of the latch may be sampled.
- Usually drawn as:

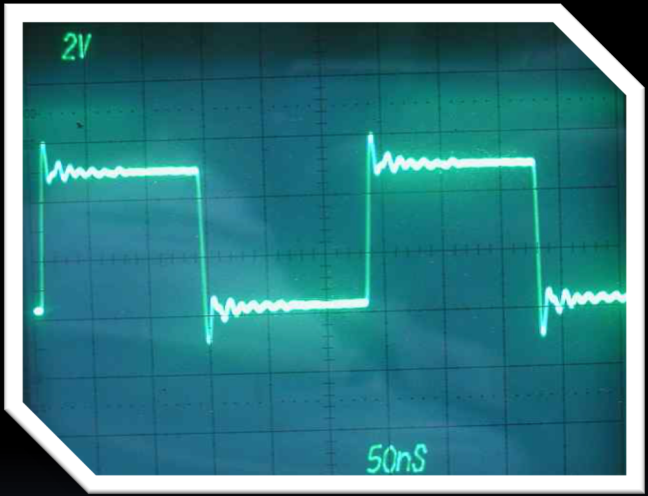


- But looks more like:



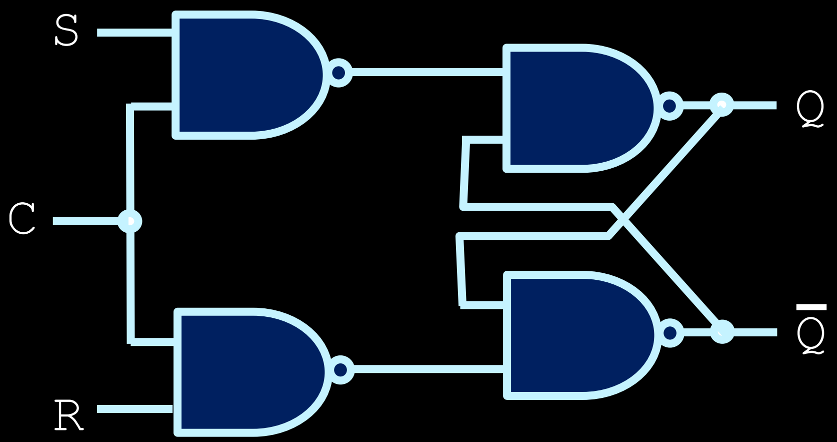
Signal restrictions

- What's the limit to how fast the latch circuit can be sampled?
- Determined by:
  - latency time of transistors
    - Setup and hold time
  - setup time for clock signal
    - Jitter
    - Gibbs phenomenon



- Frequency = how many pulses occur per second, measured in Hertz (or Hz).

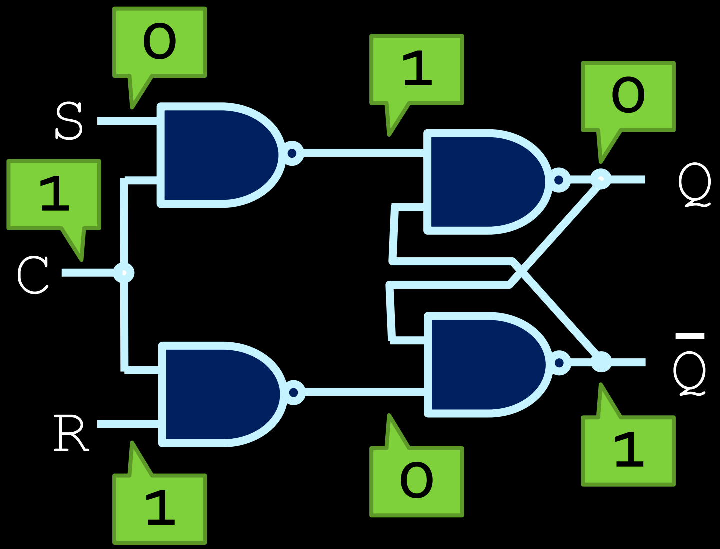
Clocked SR latch

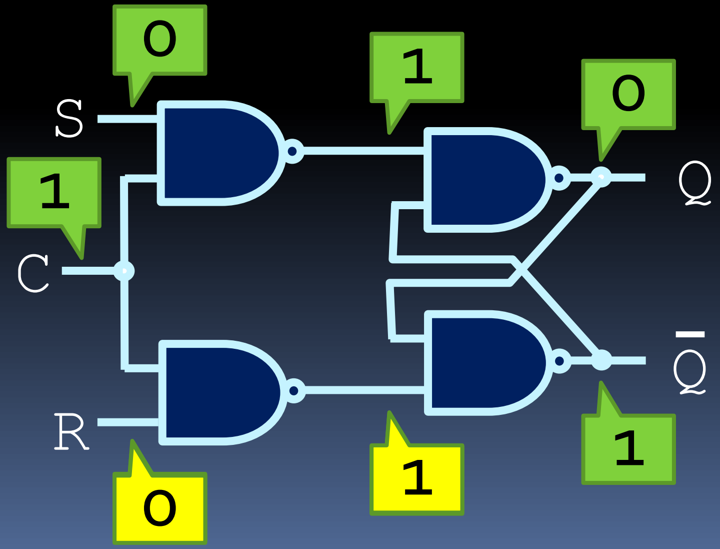


- Adding another layer of NAND gates to the $\overline{S}\overline{R}$ latch gives us a clocked SR latch or gated SR latch)
  - Basically, a latch with a control input signal `C`.
- The input `C` is often connected to a pulse signal that alternates regularly between `0` and `1` (`clock`)

Clocked SR latch behaviour

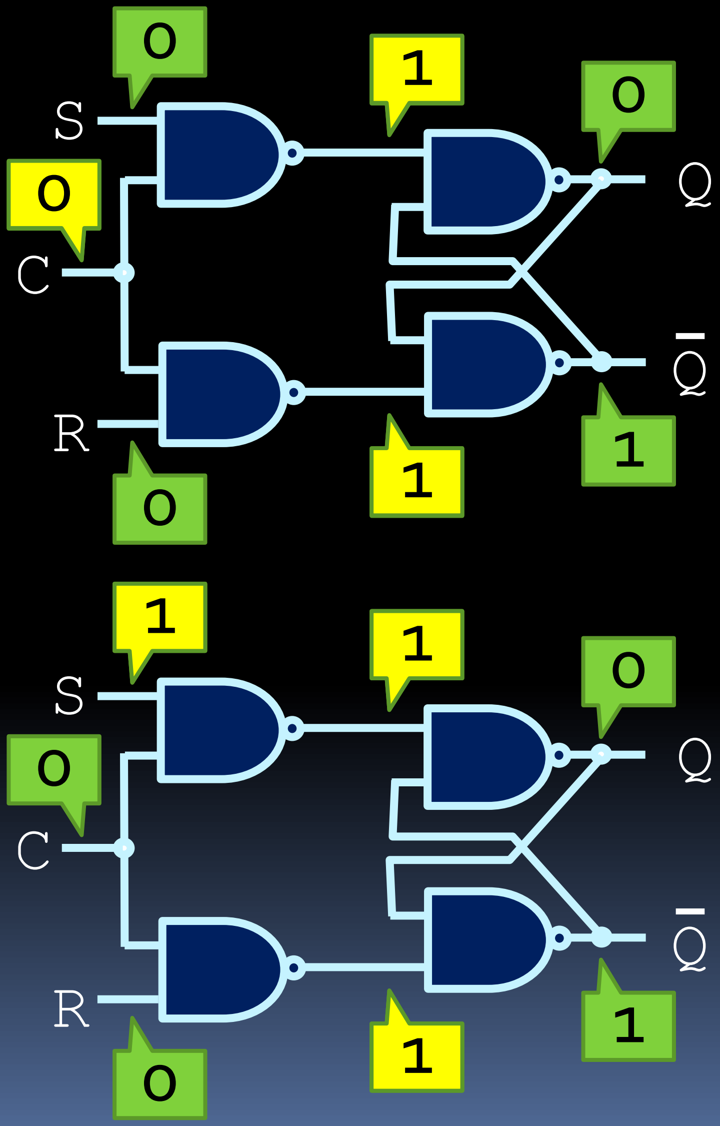
- Same behaviour as SR latch, but with timing:
  - Start off with `S=0` and `R=1`, like earlier example.
  - If clock is high, the first NAND gates invert those values, which get inverted again in the output.
  - Setting both inputs to `0` maintains the output values.



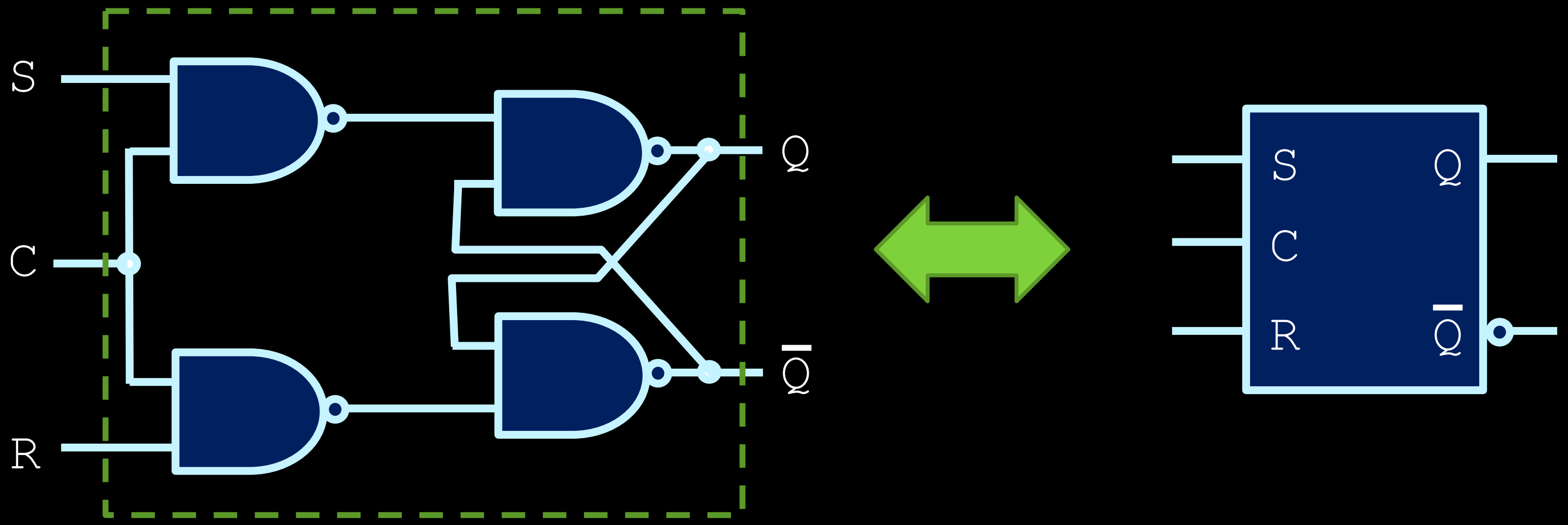


Clocked SR latch behaviour

- Continued from previous:
  - Now set the clock low.
  - Even if the inputs change, the low clock input prevents the change from reaching the second stage of NAND gates.
  - Result: the clock needs to be high in order for the inputs to have any effect.

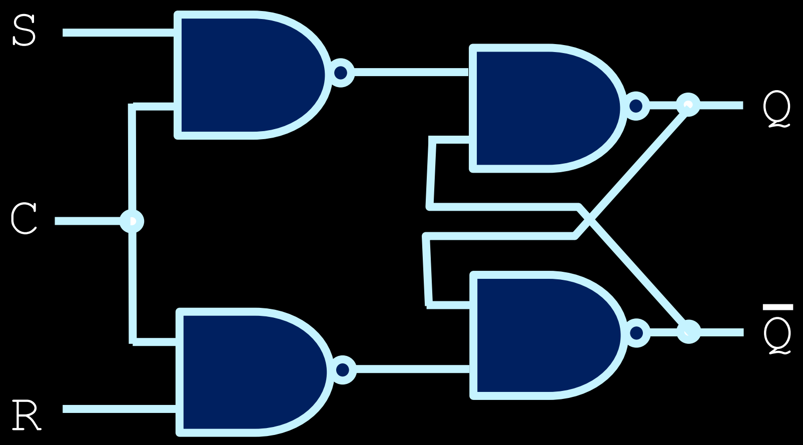


Clocked SR latch



- This is the typical symbol for a clocked SR latch.
- This only allows the S and R signals to affect the circuit when the clock input (C) is high.
- Note: the small NOT circle after the $\overline{Q}$ output is simply the notation to use to denote the inverted output value. It's not an extra NOT gate.

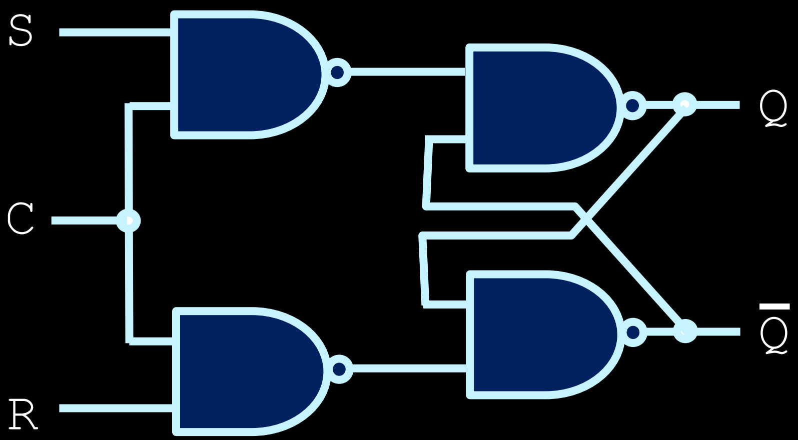
Clocked SR latch behaviour



Q_T	S	R	Q_{T+1}	Result
0	0	0	0	no change
0	0	1	0	reset
0	1	0	1	set
0	1	1	?	???
1	0	0	1	no change
1	0	1	0	reset
1	1	0	1	set
1	1	1	?	???

- Wait!
- Where’s the clock?
- There’s a better way to look at this….

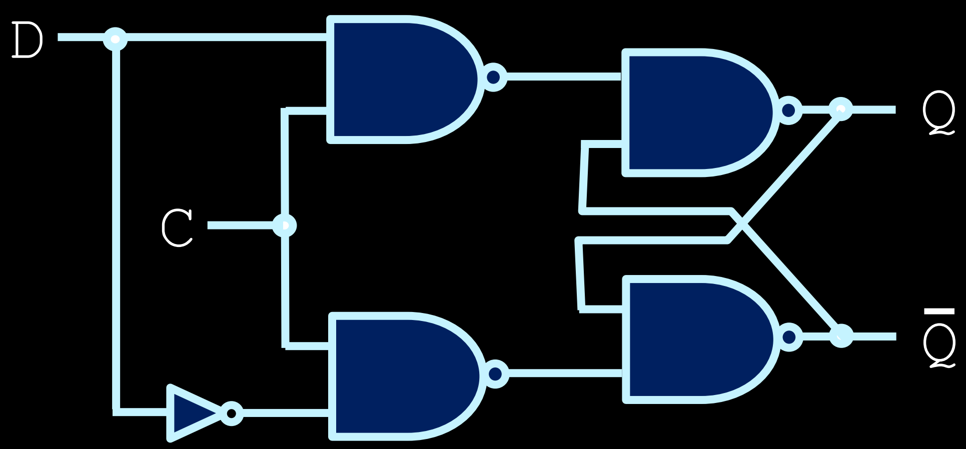
Clocked SR latch behaviour



C	S	R	Q_{T+1}	Result
0	X	X	Q_T	no change
1	0	0	Q_T	no change
1	0	1	0	reset
1	1	0	1	set
~~1~~	~~1~~	~~1~~	~~?~~	~~Undefined~~

- Assuming the clock is 1, we still have a problem when S and R are both 1, since the state of Q is indeterminate.
  - Better design: prevent S and R from both going high.

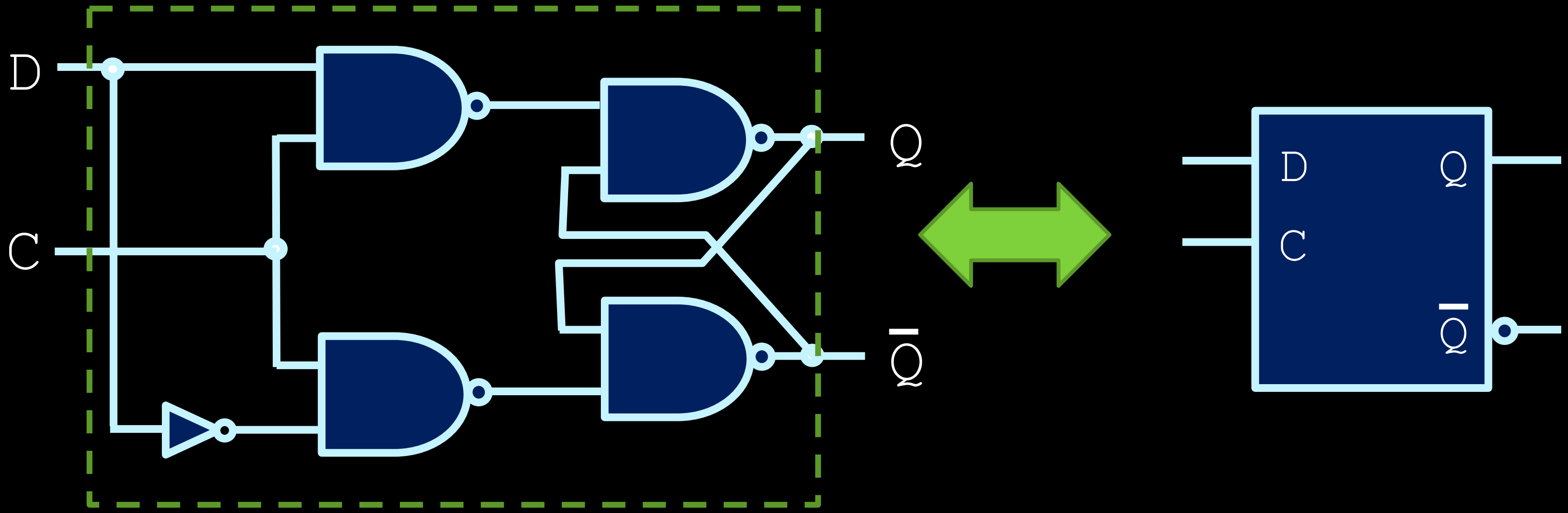
D latch



Q_T	D	Q_{T+1}
0	0	0
0	1	1
1	0	0
1	1	1

- By making the inputs to R and S dependent on a single signal D, you avoid the indeterminate state problem.
- The value of D now sets output Q low or high whenever C is high.

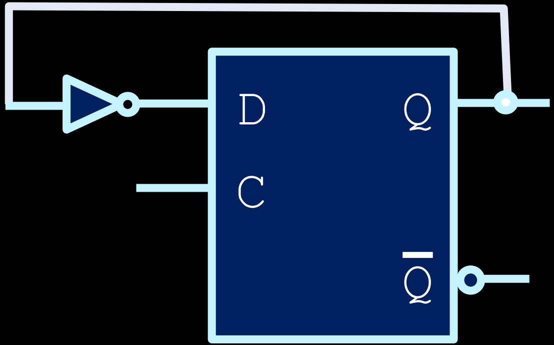
D latch

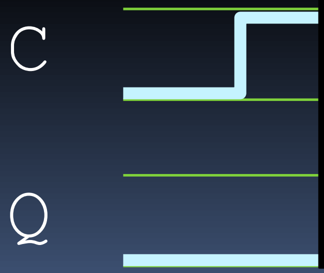


- This design is good, but still has problems.
  - i.e. timing issues.
  - How can we maintain state?

Latch timing issues

- Consider the circuit on the right:
- When the clock signal is high, the output looks like the waveform below:
  - Output keeps toggling back and forth.

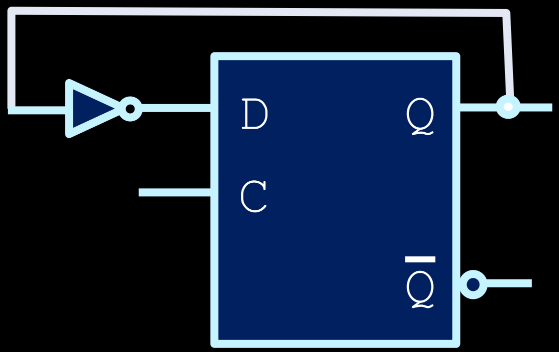


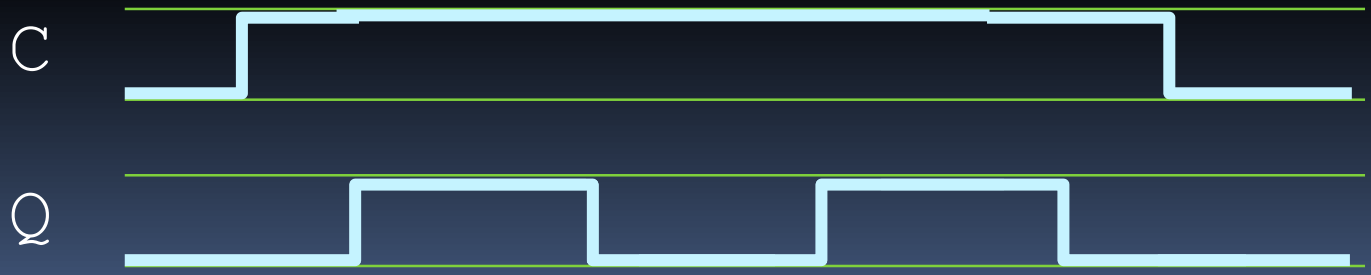


...what happens next?

Latch timing issues

- Consider the circuit on the right:
- When the clock signal is high, the output looks like the waveform below:
  - Output keeps toggling back and forth.





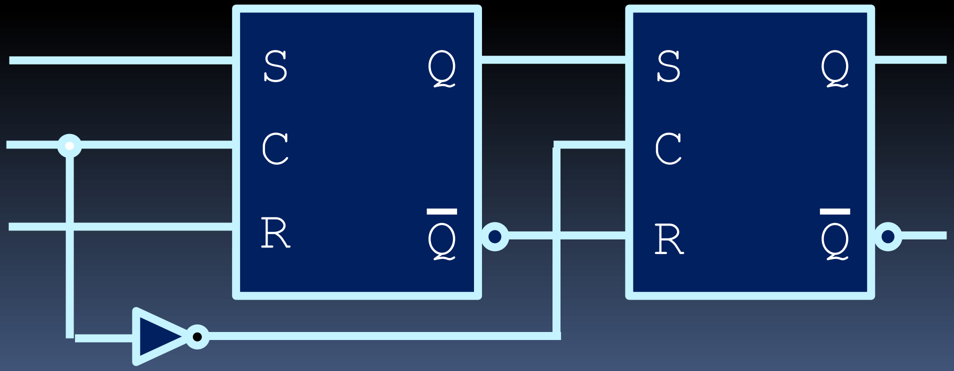
D-Latch is transparent!

- Transparent means that
  - Any changes to its inputs are visible to the output when control signal (Clock) is 1.

- Key Take-away: The "output of a latch **should not** be applied directly or through combinational logic to the input of the same or another latch when they all have the same control (clock) signal."

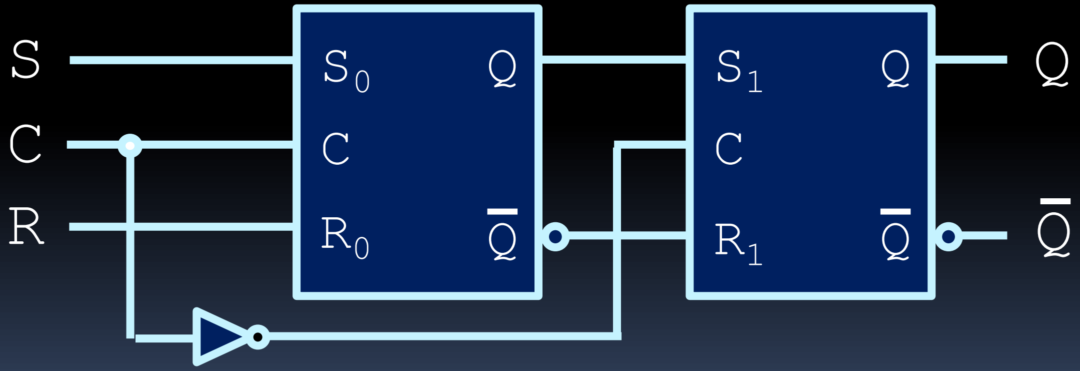
Latch timing issues

- Preferable behaviour:
  - Have output change only once when the clock pulse changes.
  - Solution: create disconnect between circuit output and circuit input, to prevent unwanted feedback and changes to output.

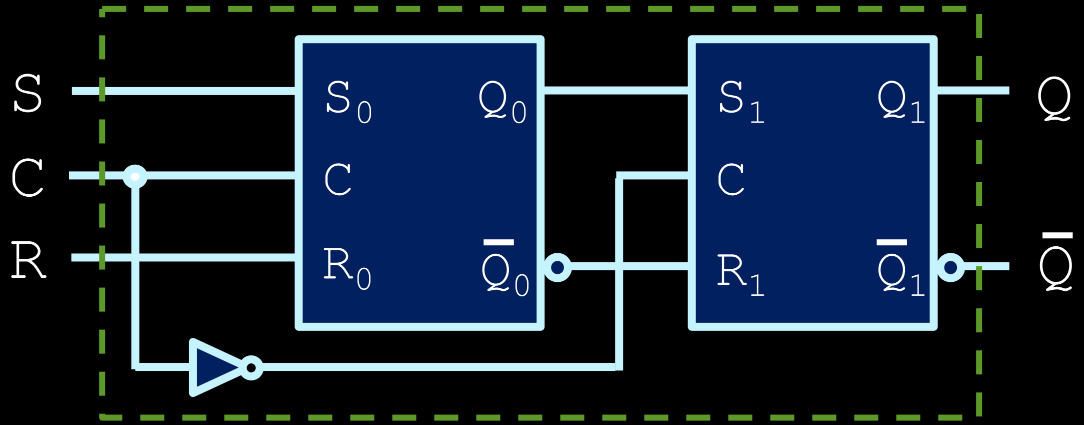


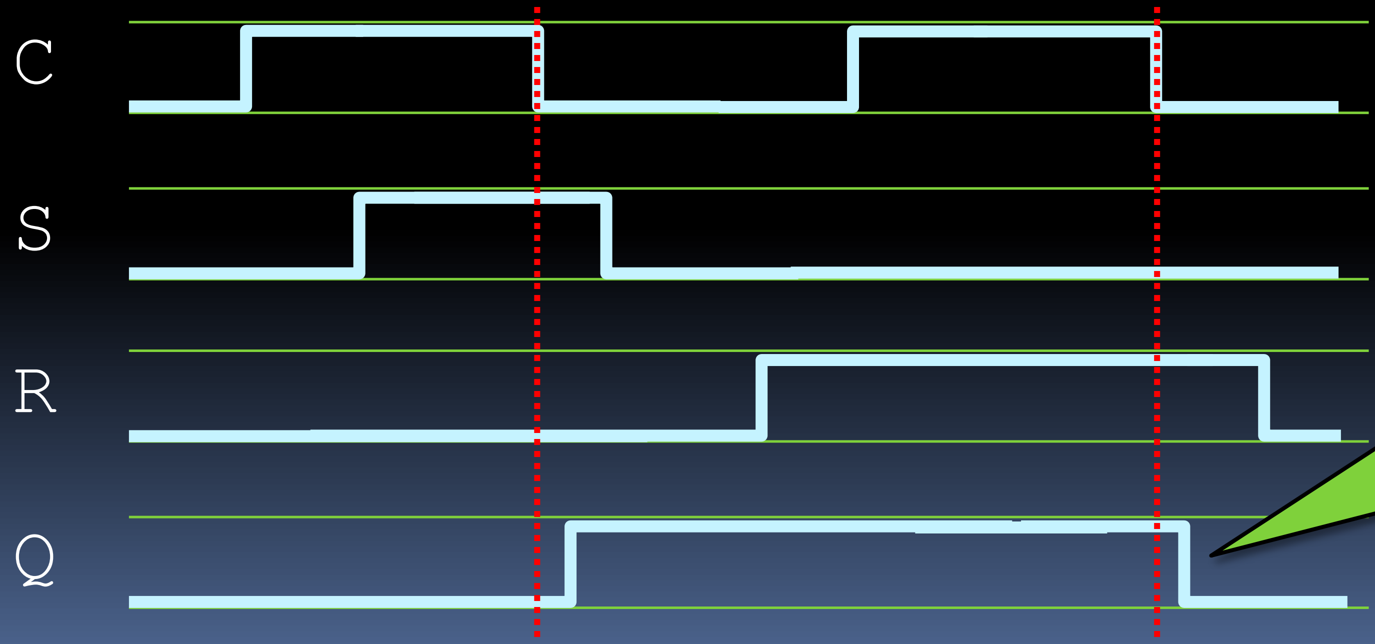
SR master-slave flip-flop

- A flip-flop is a latched circuit whose output is triggered with the rising edge or falling edge of a clock pulse.
- Example: The SR master-slave flip-flop



SR master-slave flip-flop

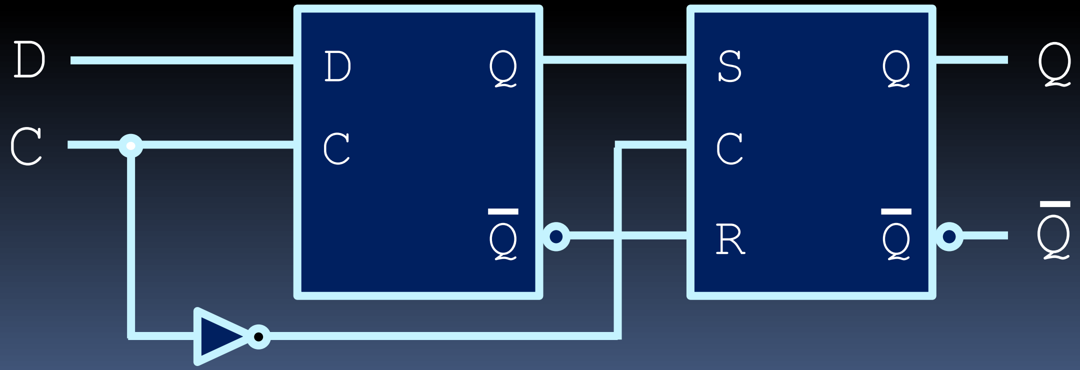




propagation delay

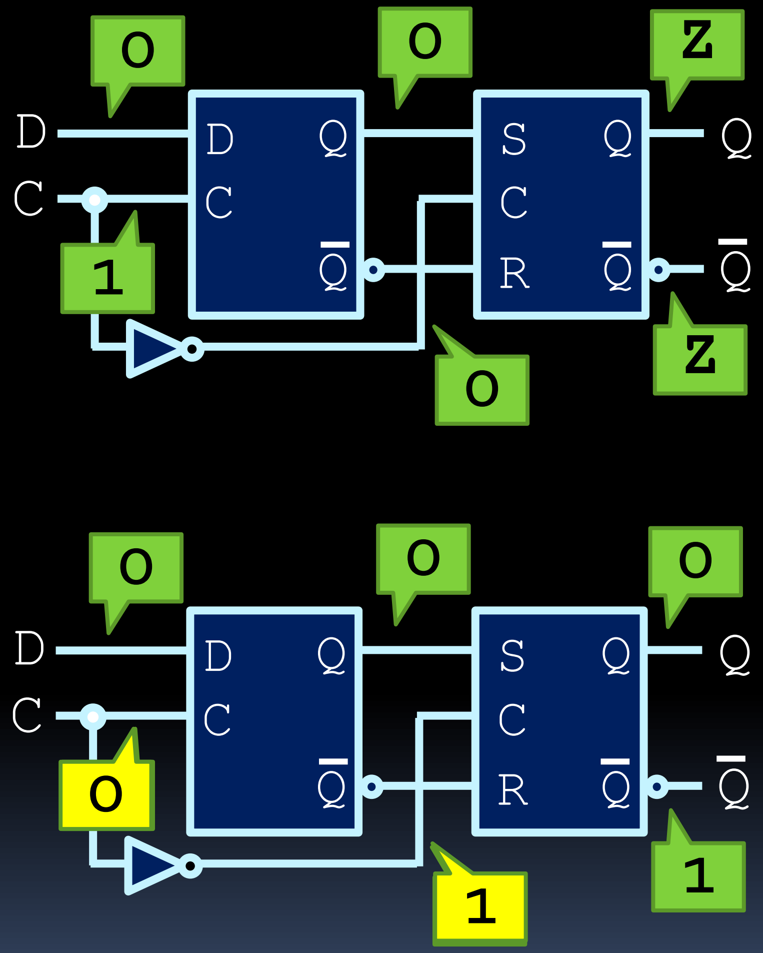
Edge-triggered D flip-flop

- SR flip-flops still have issues of unstable behaviour.
- Solution: D flip-flop
  - Connect D latch to the input of a SR latch.
  - Negative-edge triggered flip-flop (like the SR)



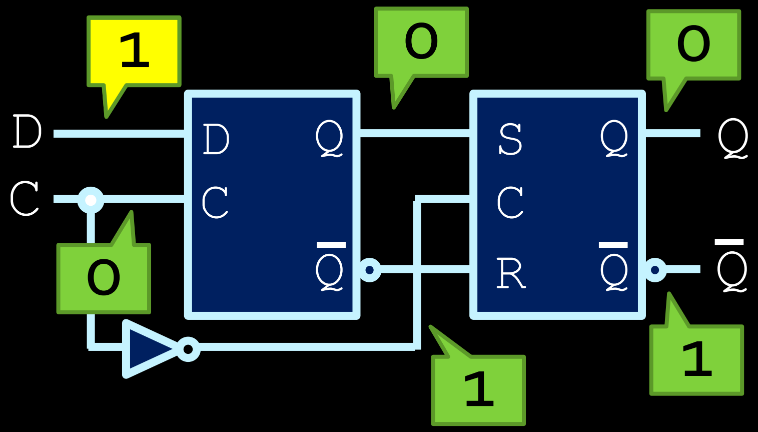
Flip-flop behaviour

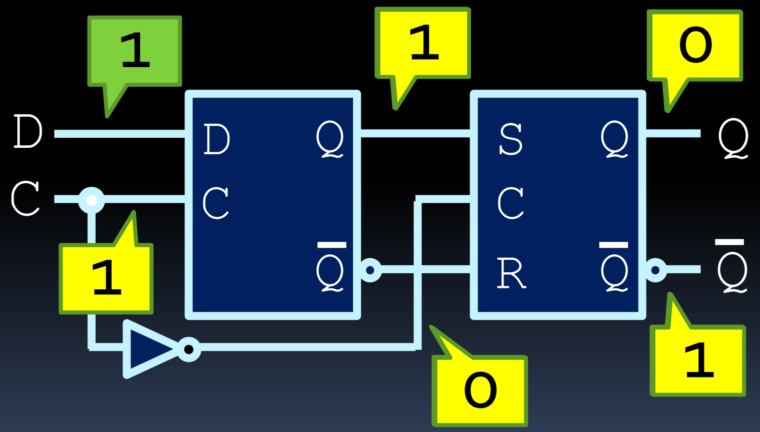
- Observe the behaviour:
  - If the clock signal is high, the input to the first flip-flop is sent out to the second.
  - The second flip-flop doesn't do anything until the clock signal goes down again.
  - When it clock goes from high to low, the first flip-flop stops transmitting a signal, and the second one starts.



Flip-flop behaviour

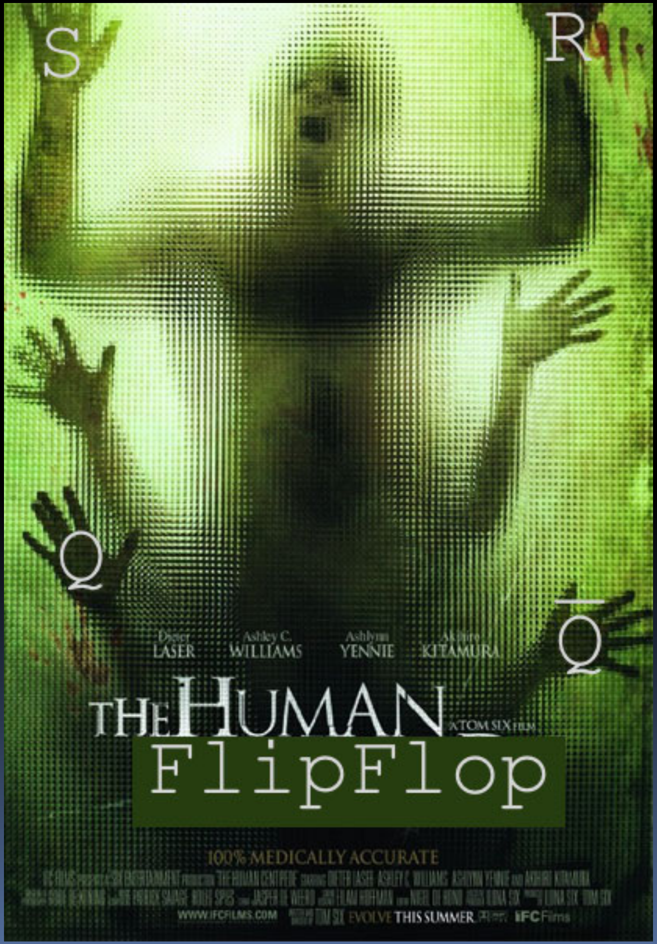
- Continued from previous:
  - If the input to D changes, the change isn't transmitted to the second flip-flop until the clock goes high again.
  - Once the clock goes high, the first flip-flop starts transmitting at the same time as the second flip-flop stops.





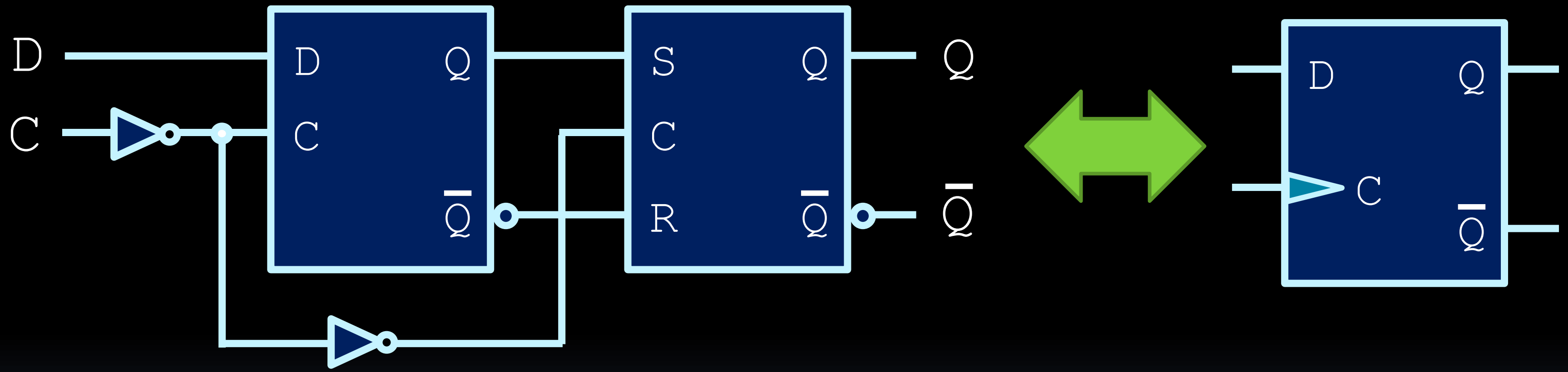
Confused yet?

- Maybe a demonstration will help



Edge-triggered flip-flop

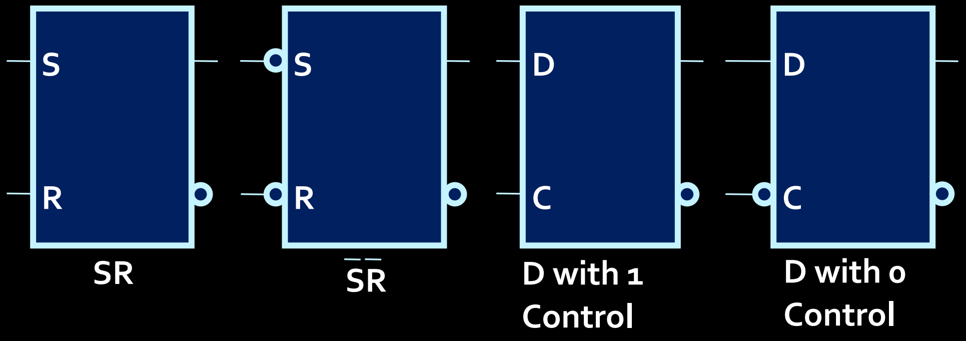
- Alternative: positive-edge triggered flip-flops



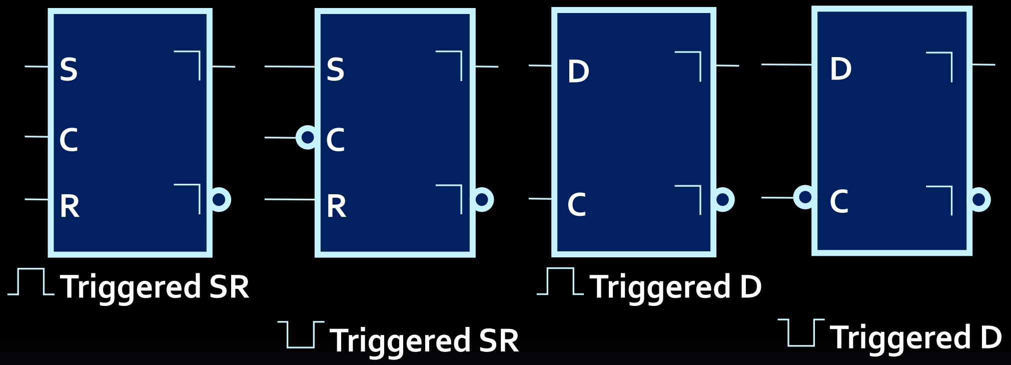
- These are the most commonly-used flip-flop circuits (and our choice for the course).

Notation

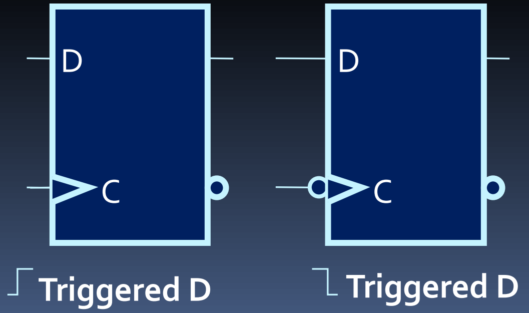
- Latches



- Master-slave flip-flops



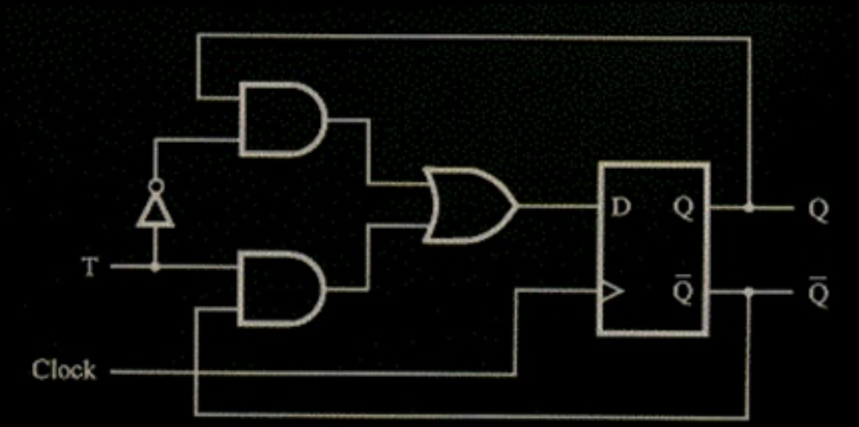
- Edge-triggered flip-flops

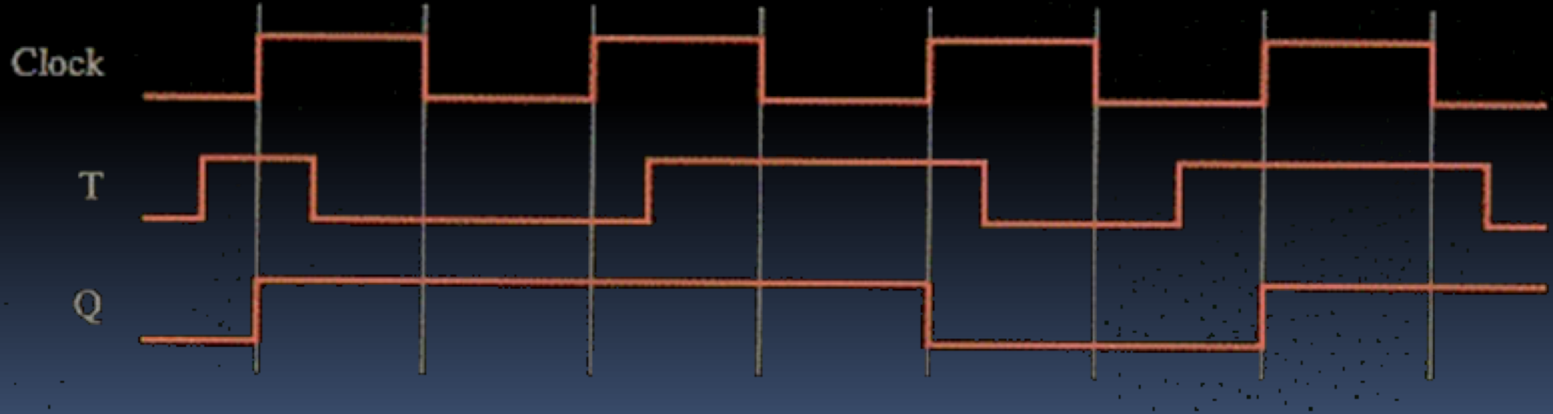


Note: While all these are possible, we mainly use edge-triggered D flip-flops in our designs.

Other Flip-Flops

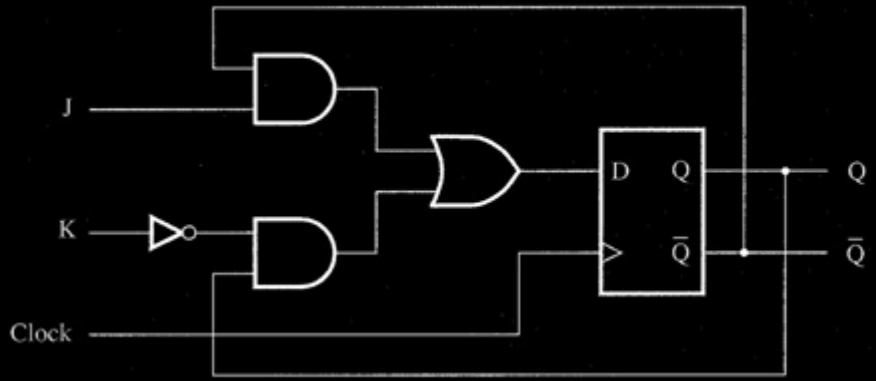
- The T flip-flop:
  - Like the D flip-flop, except that it toggles its value whenever the input to T is high.

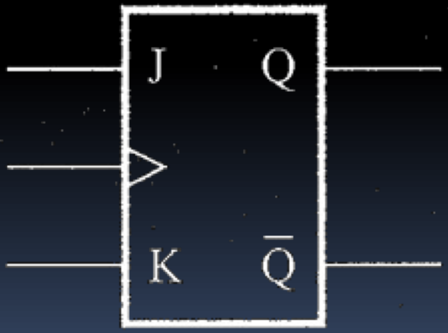




Other Flip-Flops

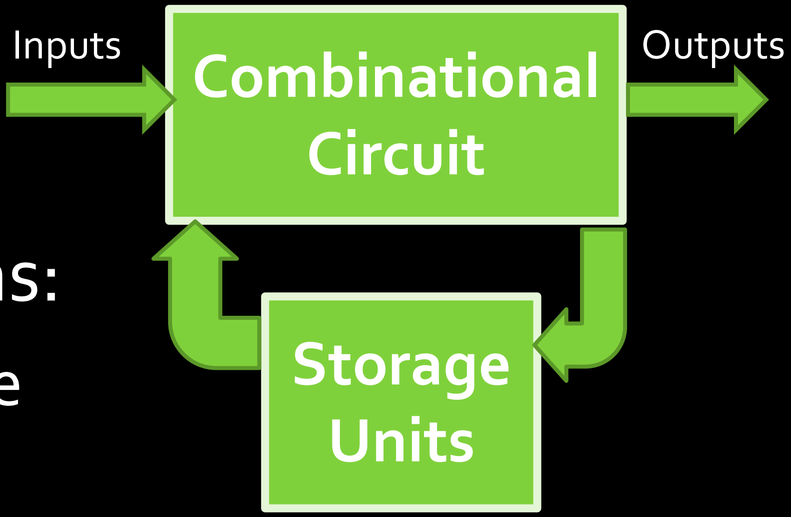
- The JK Flip-Flop:
  - Takes advantage of all combinations of two inputs (J & K) to produce four different behaviours:
    - if J and K are 0, maintain output.
    - if J is 0 and K is 1, set output to 0.
    - if J is 1 and K is 0, set output to 1.
    - if J and K are 1, toggle output value.





Sequential circuit design

- Similar to creating combinational circuits, with extra considerations:
    - The flip-flops now provide extra inputs to the circuit
    - Extra circuitry needs to be designed for the flip-flop inputs.
    - ...which is next ☺


Inputs
Combinational Circuit
Outputs
Storage Units