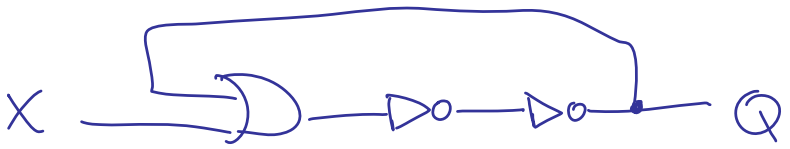
Sequential circuits lecture notes

Last time: Combinational circuits
- implement boolean fn of present inputs
- e.g. adder, decoder, multiplexer

Today: Sequential circuits
- implement boolean fn of present and past inputs
- typically use feedback to achieve this
  i.e. one or more outputs "feed back" as an input.

e.g.



Truth tables of sequential circuits can represent feedback either explicitly (in a separate column) or implicitly (use Q(t) in output column)

inputs		outputs
X	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	1

OR

X	Q(t+1)
0	Q(t)
1	1

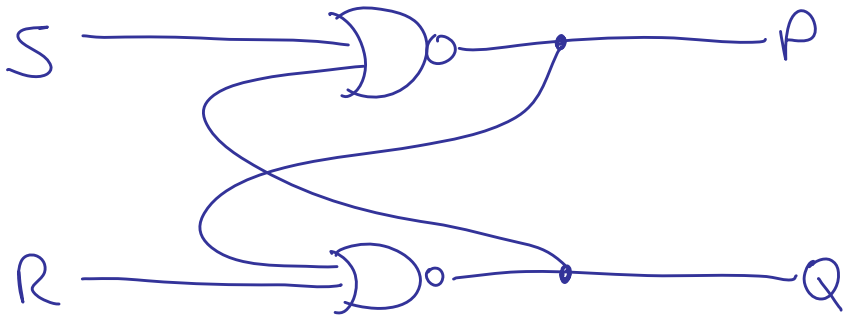
We'll study several important sequential circuits:
1. SR flip-flop (unclocked and clocked versions)
2. JK flip-flop
3. D flip-flop
4. read-write memory
5. binary counter

} clocked versions only

① (a) unclocked flip-flop

terminology: to <u>set</u> a bit means to assign it the value 1
to <u>reset</u> a bit means to assign it the value 0

non-standard representation of S-R flip-flop:

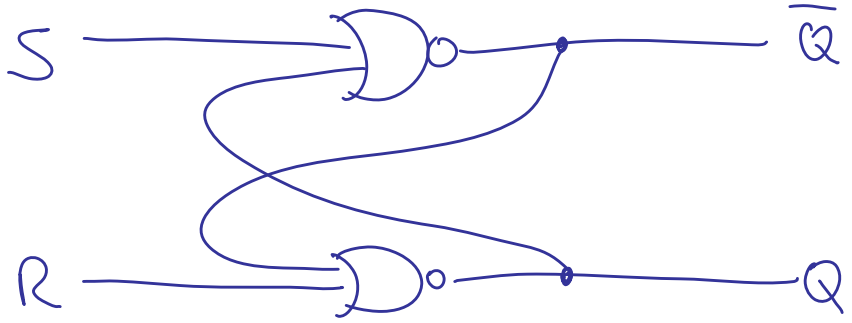


exercise: when R=1, Q = 0
S=1, P = 0
P=1, Q = 0
Q=1, P = 0

Thus have strange truth table:

inputs S	inputs R	outputs P	outputs Q	
0	0	1	0	} either is possible
		0	1	
0	1	1	0	
1	0	0	1	
1	1	0	0	

If we agree that we'll never have S and R = 1 at the same time, we see from the truth table that $P = \overline{Q}$, so we can write the circuit as:



(this is the standard representation)

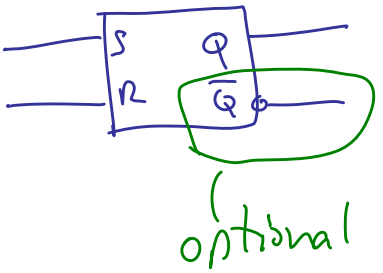
Exercise: fill in the sequence of outputs (this is not a truth table).

time t	S	R	Q	$\overline{Q}$
0	1	0	1	0
1	0	0	1	0
2	1	0	1	0
3	0	0	1	0
4	0	1	0	1
5	0	0	0	1
6	1	0	1	0
7	0	0	1	0

answer

Note that S ≡ "set" i.e. sets Q to 1
R ≡ "reset" i.e. resets Q to 0.

circuit symbol for (unclocked) SR flip-flop:



(1)(b) Clocked SR flip-flop

A clock gives out a repeating on-off signal, e.g. 

Attaching a clock to a circuit can make it change state exactly once per cycle e.g.

- when clock is 1
- when clock is 0

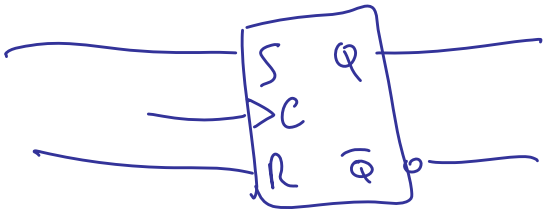
} level-triggered

- when clock changes 0 → 1
- when clock changes 1 → 0

} edge-triggered

We don't study details of how this is achieved.

Symbol for clocked SR flip-flop:



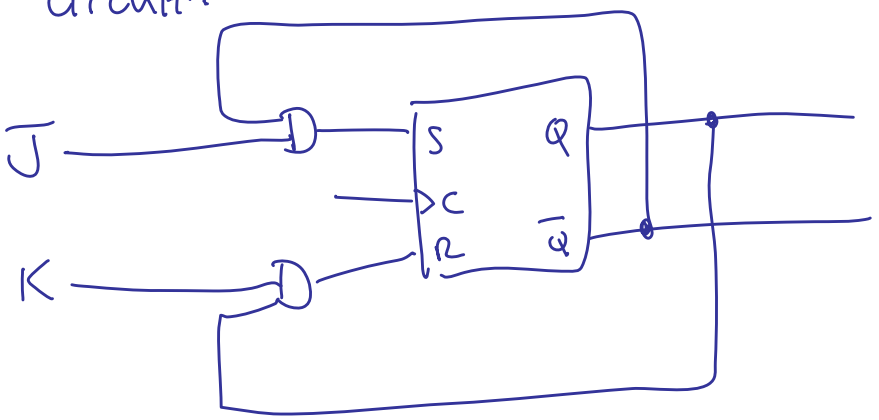
Exercise: fill in the truth table on the handout (puzzle 1).

Better truth table:

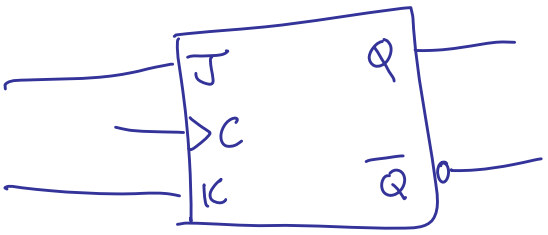
S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	undefined

(2) JK flip-flop

circuit:



symbol:



truth table: same as SR, but output is toggled when J & K are 1:

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

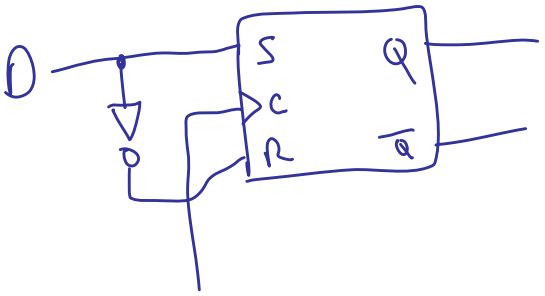
Important lesson from toggling behavior:

At each clock tick, all flip flops compute output based on current input — but this output does not feed back and affect the inputs until the next clock tick

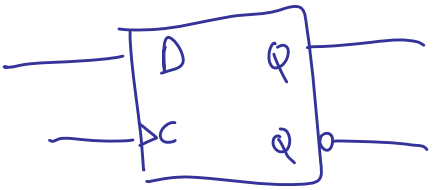
Exercise: Understand this using above circuit diagram for JK flip-flop.

④ D flip flop

circuit:



Symbol:



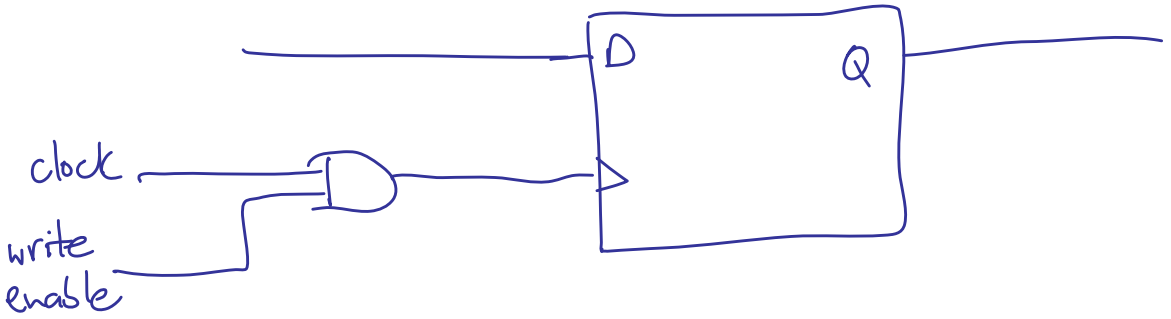
truth table:

D	Q(t+1)
0	0
1	1

Seems pointless, since $Q \equiv D$. Wouldn't a wire achieve the same thing?

Answer: No – it's all about timing. Q stays at a stable level between clock pulses. Meanwhile, we can adjust D.

This leads to circuit for a 1-bit memory:



⑤ Read-write memory

See textbook fig 3.32

Activity: do handout puzzles 2 and 3

⑥ Binary counter

See textbook fig 3.31

Activity: do handout puzzle 4.

If time, check out the online demo on resource page