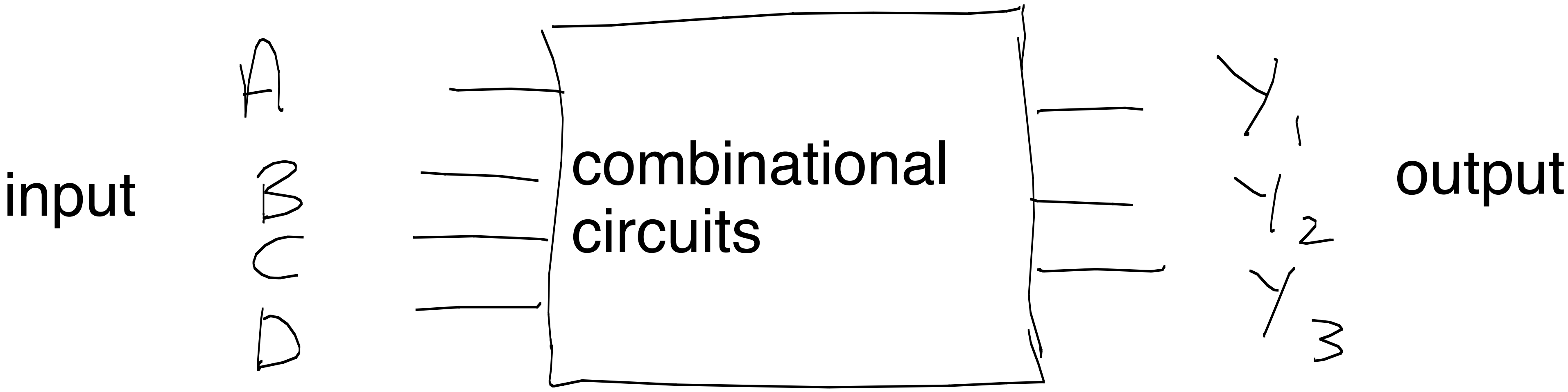
lecture 5

Sequential circuits 1

- RS latch
- D latch
- flipflops (D)
- registers

last week....



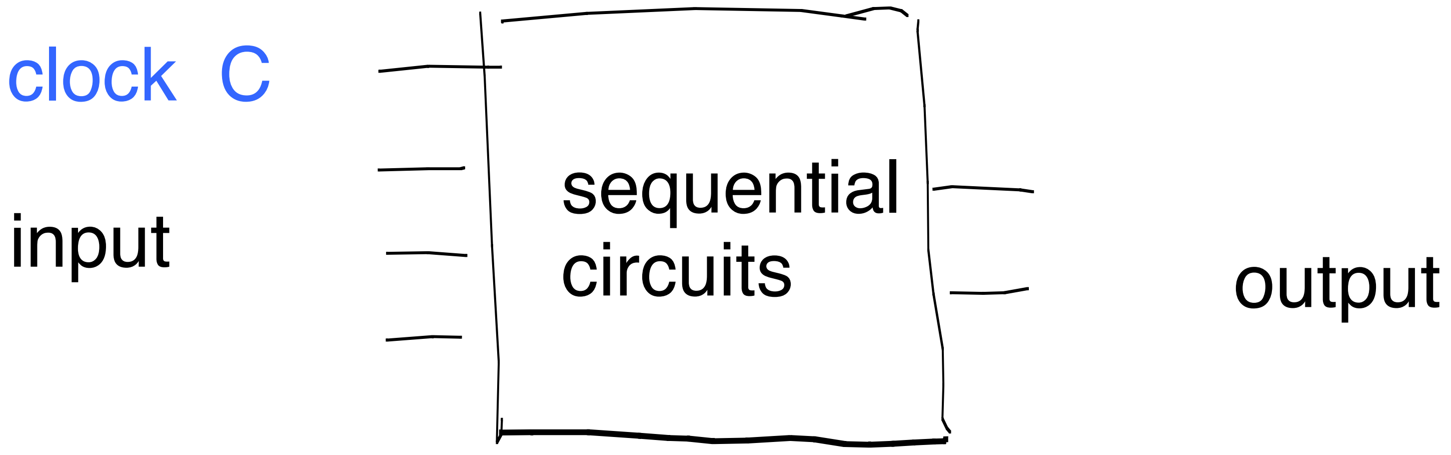
input

combinational circuits

output

- truth tables and circuit diagrams
- 0 and 1 signals are (voltage) values on wires
- circuits take time to "compute" e.g. carries in addition

this week....



clock C

input

sequential circuits

output

- combinatorial circuits + *memory*

- synchronized by a clock C

Memory (two kinds)

- write it down

- repeat it to yourself (feedback)

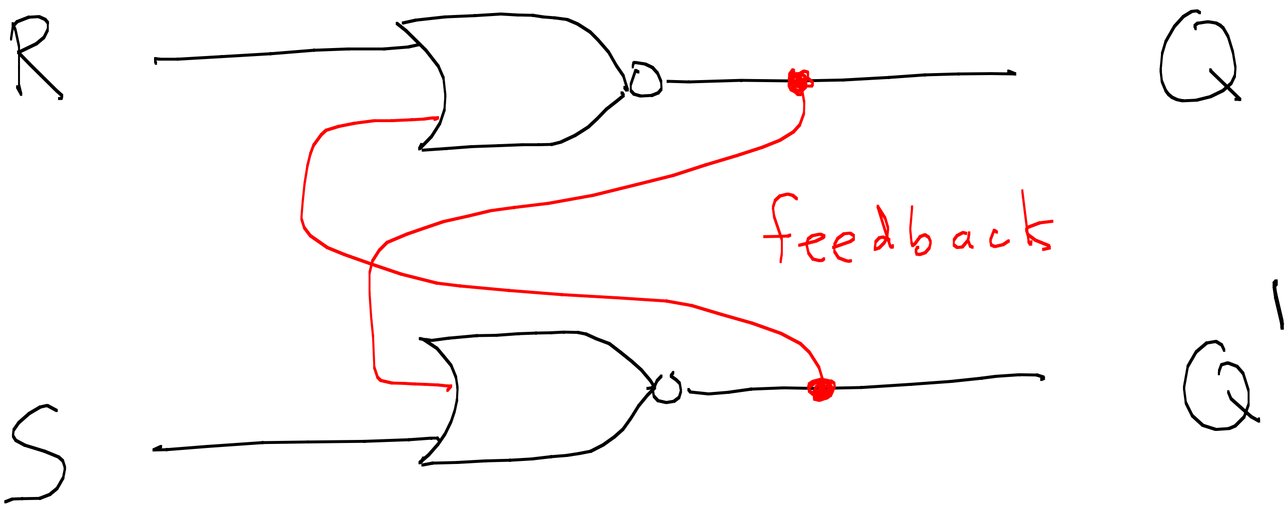
Sequential circuits use the latter.

Latch

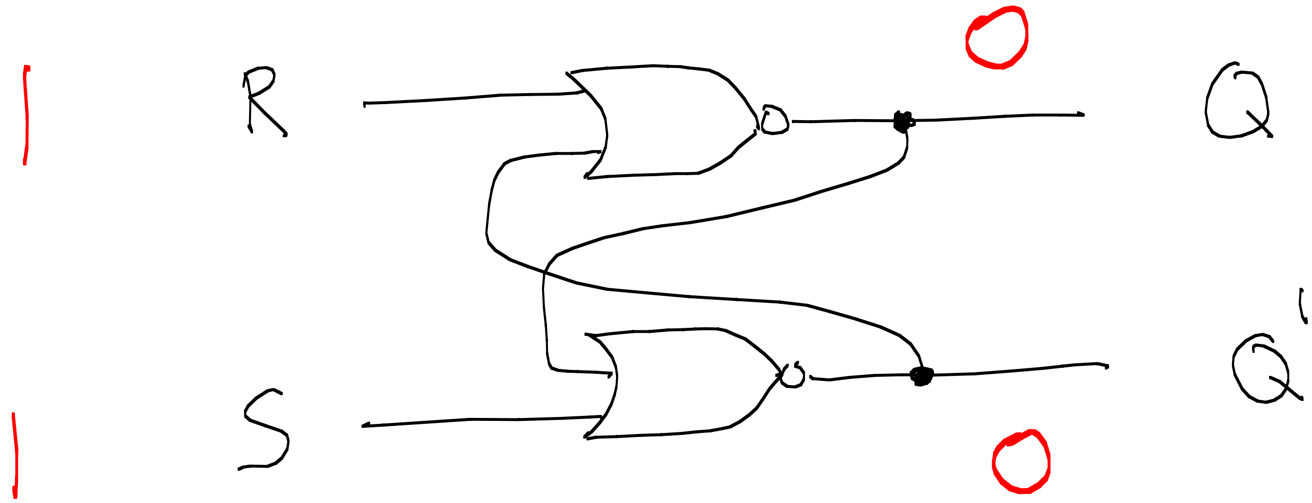
definition (wikipedia); ".. a type of ... fastener that is used to join two objects or surfaces together while allowing for the regular ... separation of the surfaces"

Latches are often (but not always) used to block paths, e.g. close doors.

RS latch ('reset' 0, 'set' 1)

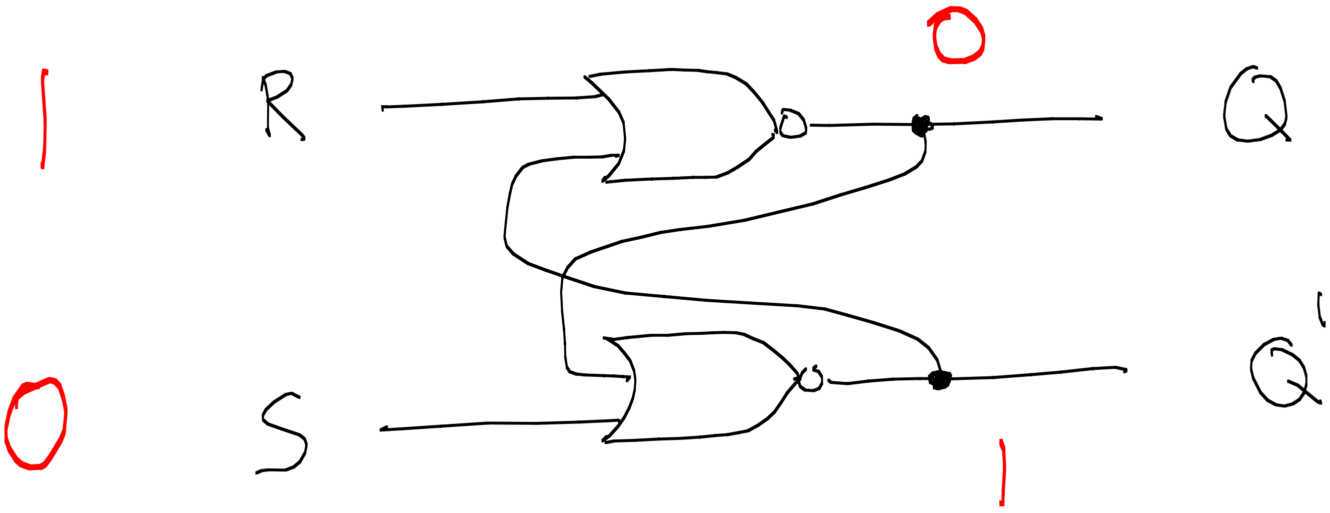


A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

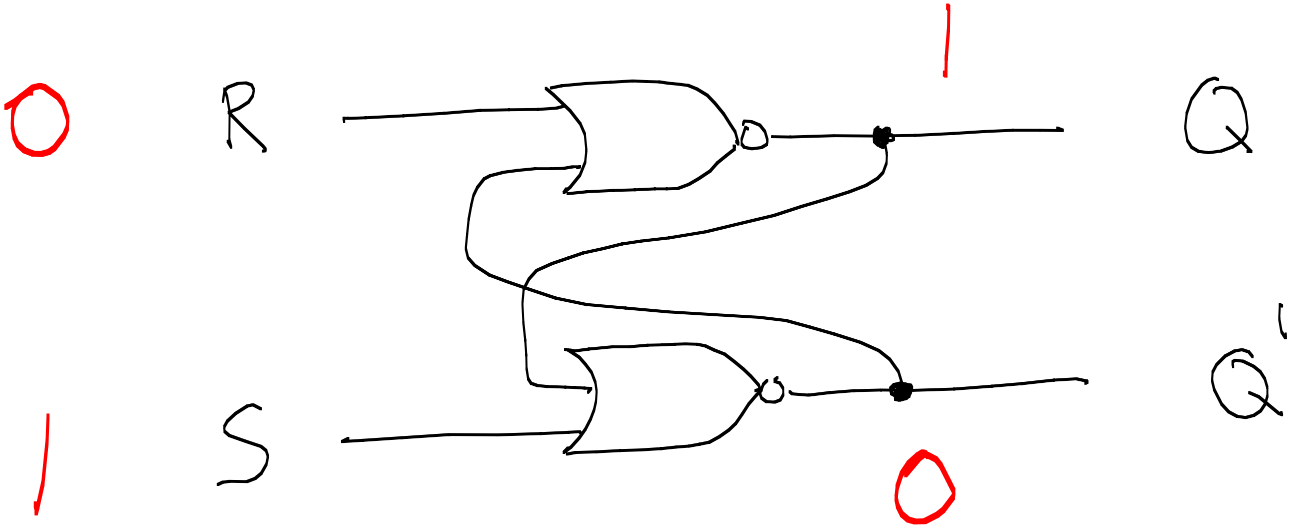


R = S = 1 inputs will not be allowed.

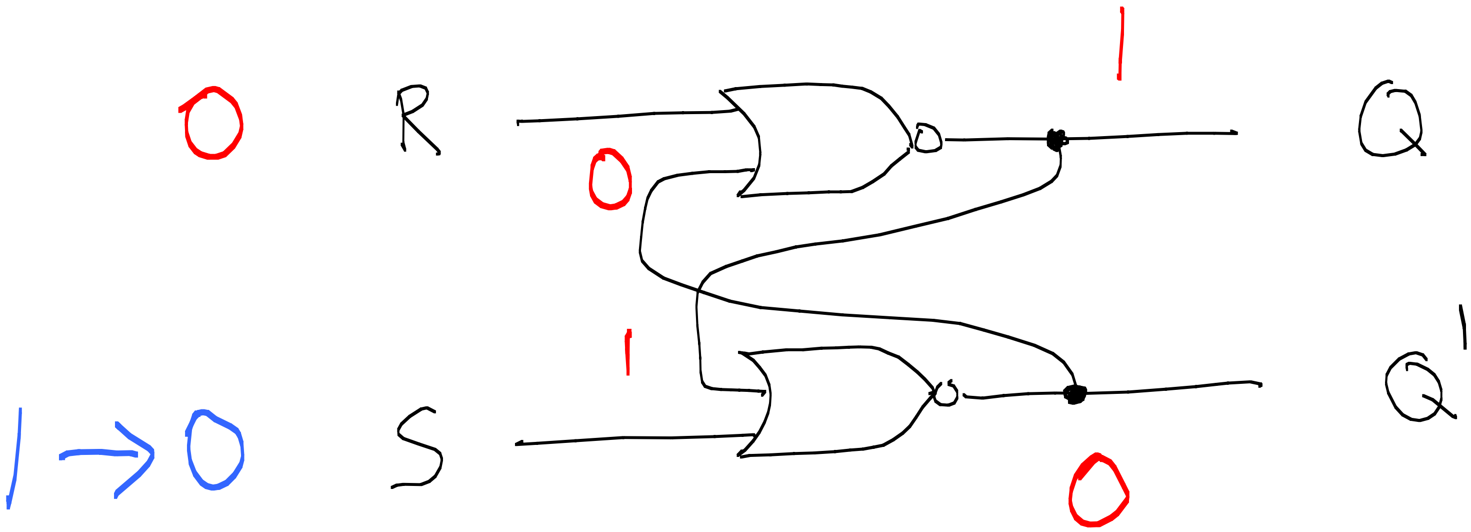
A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

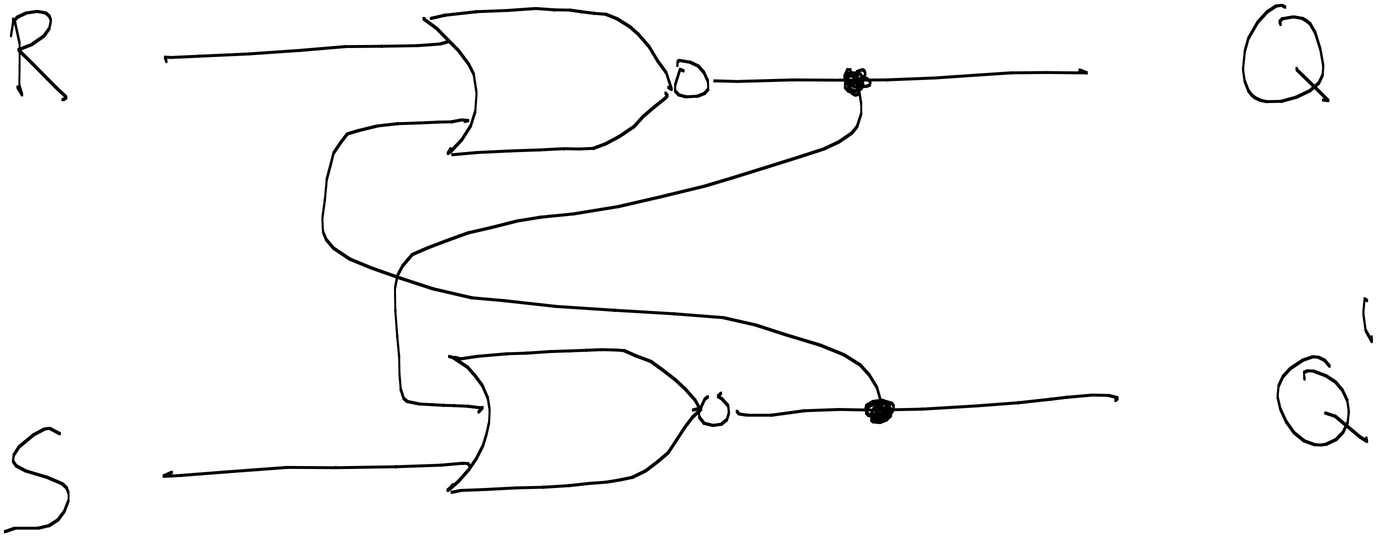


A	B	A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



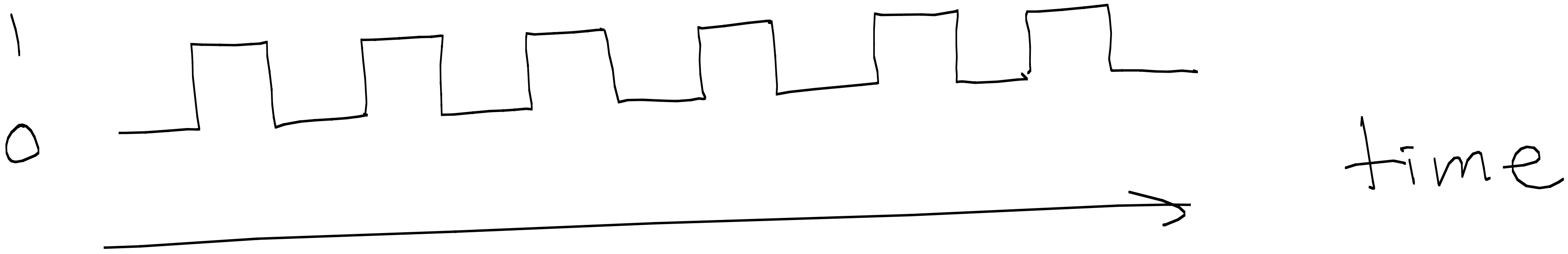
$Q, \overline{Q}$ values do not change (memory)

A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0



R	S	Q	Q'
0	0	hold previous values	
0	1	1	0
1	0	0	1
1	1	0	0

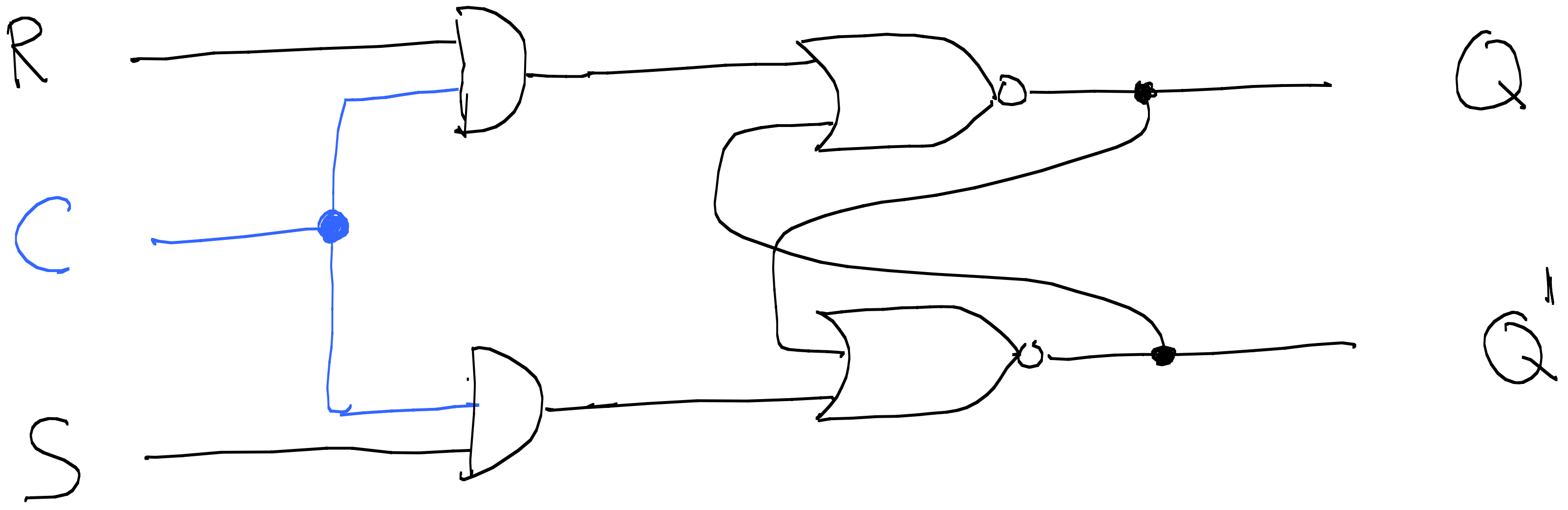
Clock



- electronic implementation uses "crystal oscillator"
  https://en.wikipedia.org/wiki/Clock_signal

- typical clock speed is in gigaherz (10^9 cycles/sec)

Clocked RS latch



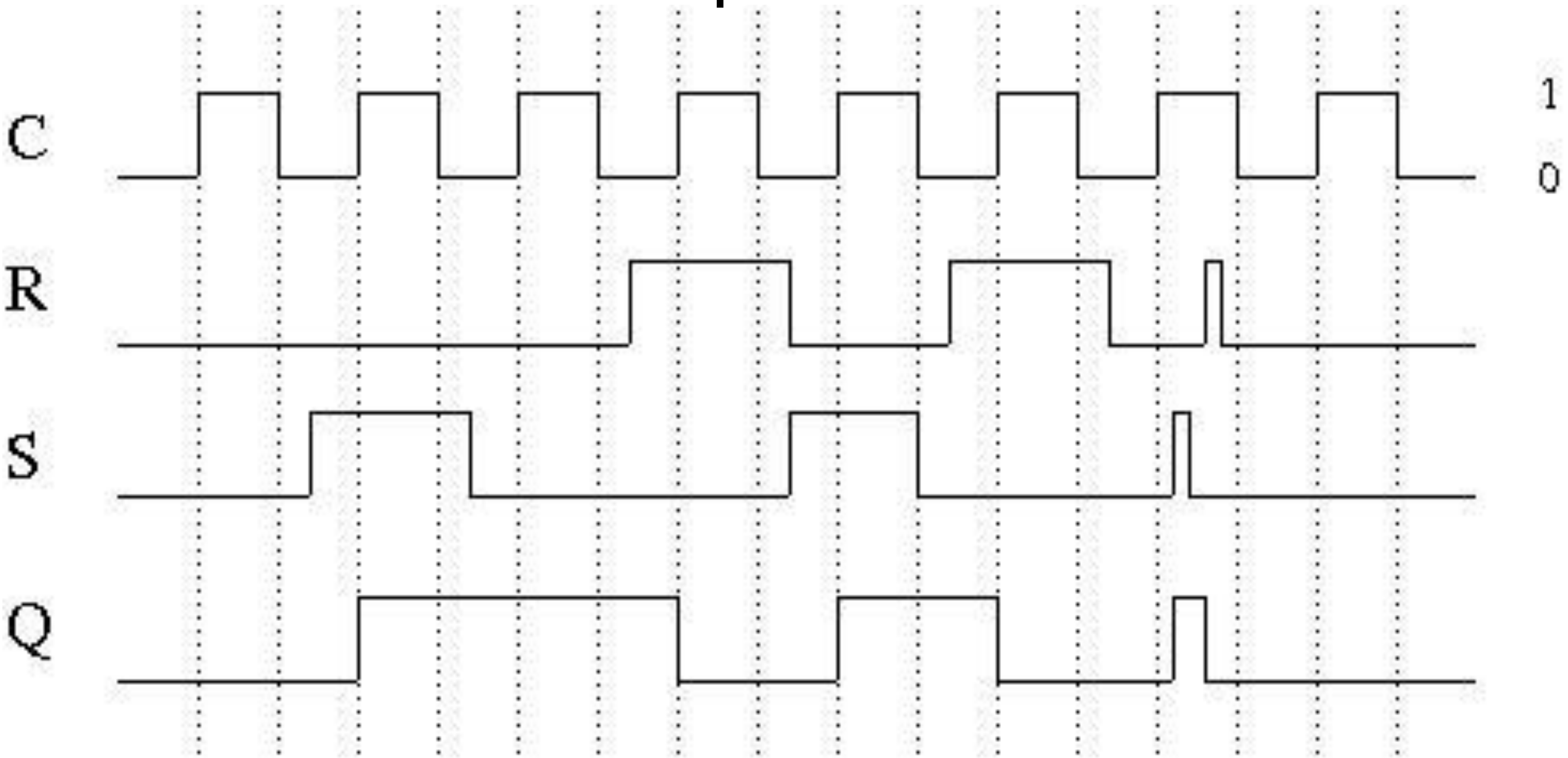
C = 0

R	S	Q	Q'
0	0	hold	
0	1		
1	0		
1	1		

C = 1

R	S	Q	Q'
0	0	hold	
0	1	1	0
1	0	0	1
1	1	0	0

Example



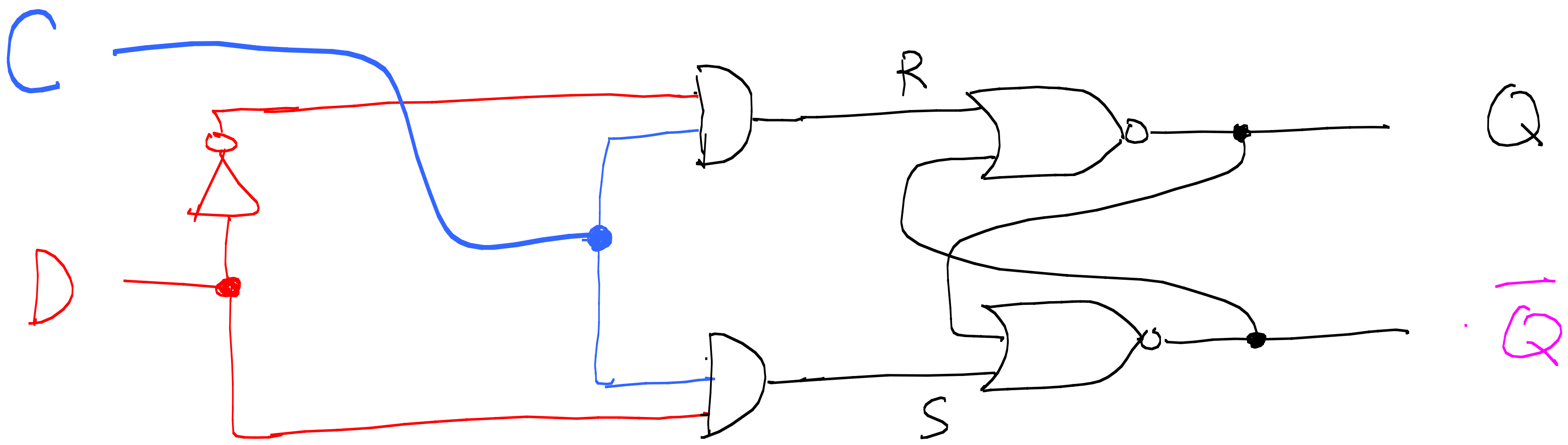
C = 0

R	S	Q	Q'
0	0	hold	
0	1		
1	0		
1	1		

C = 1

R	S	Q	Q'
0	0	hold	
0	1	1	0
1	0	0	1
1	1	0	0

D latch ("D" is for data)



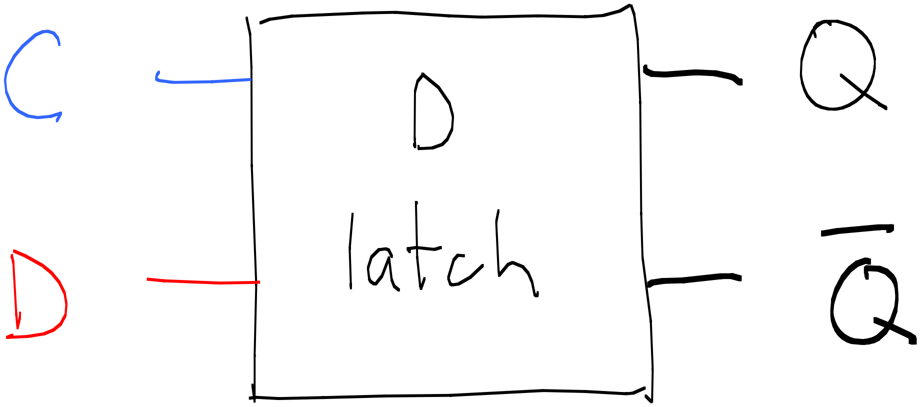
What does this circuit do ?

when $C = 1$, $D = 1 \Rightarrow Q = 1, \overline{Q} = 0$

$D = 0 \Rightarrow Q = 0, \overline{Q} = 1$

when $C = 0$, hold values of $Q, \overline{Q}$

D latch

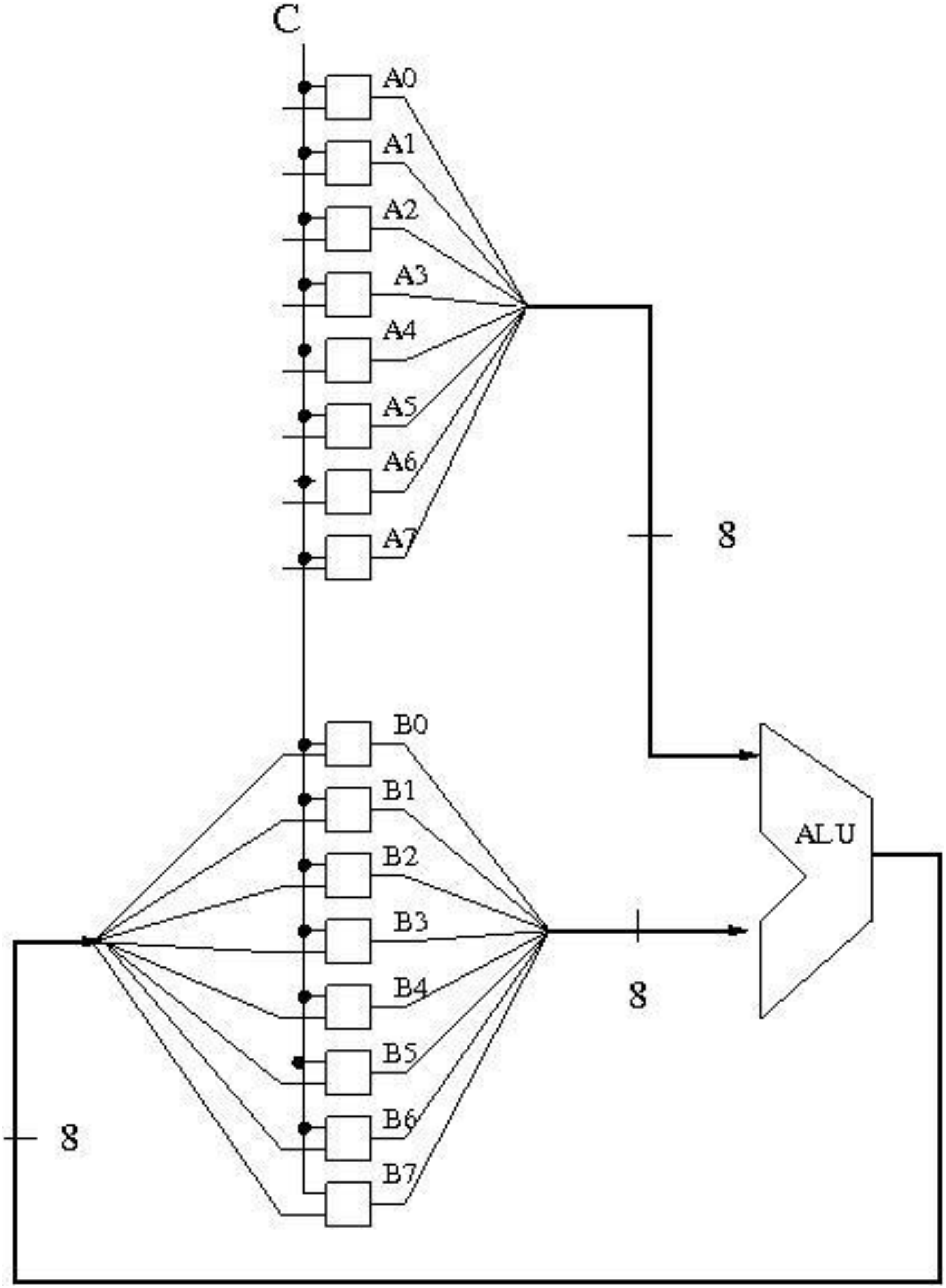


C = 0 holds values in D latches. (Read only)

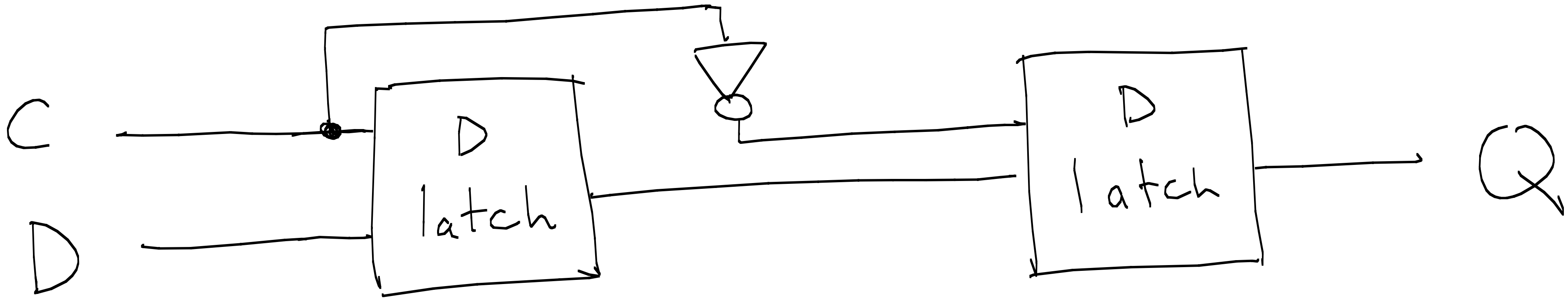
C = 1 allows values in D latches to go through.
(Read and write).

Example:

Suppose we used D latches to store 8 bit numbers A and B. Suppose we added A and B using the circuit below and wrote the new value back into B. Would this work? No, because when C = 1 there would be no control over timing and we could loop through multiple times within a single clock pulse (while C = 1).



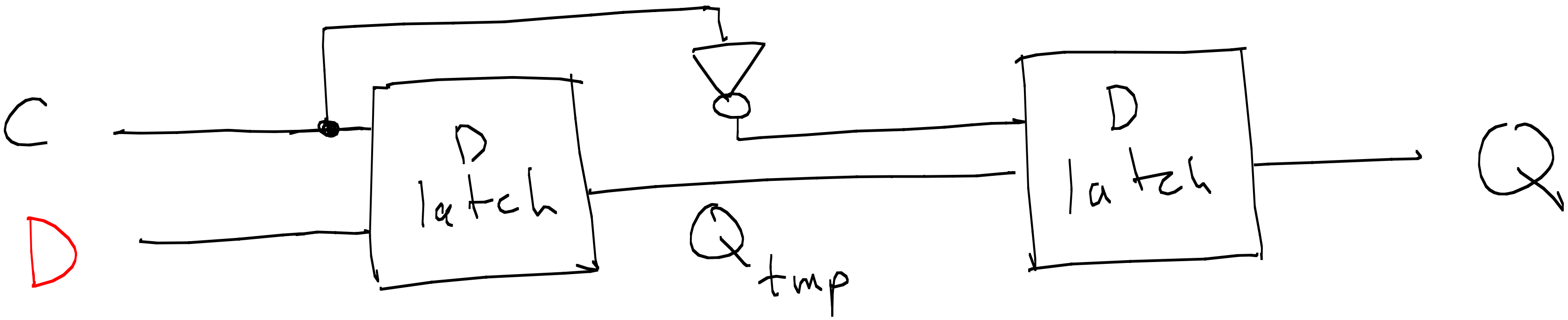
D flip flop



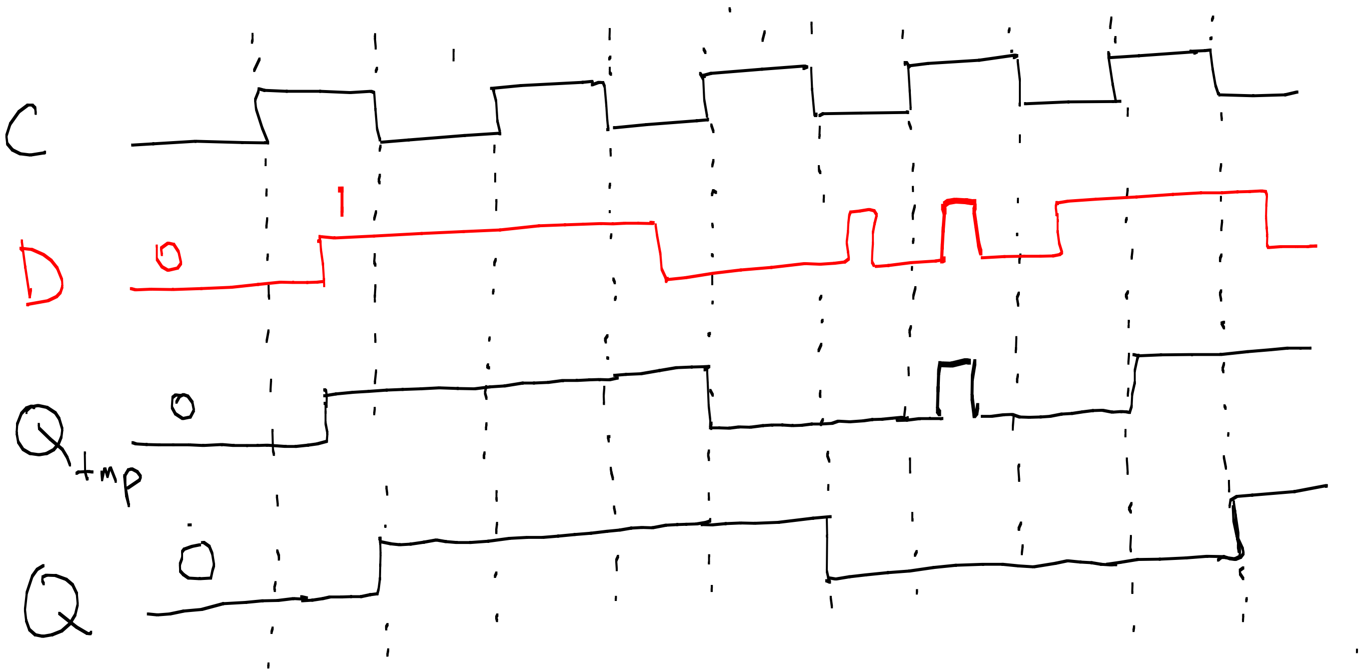
C = 1 Write D value into first D latch.
Q doesn't change.

C = 0 Stop writing D into first D latch.
The D value from first D latch is written into second, so Q gets a possibly new value.

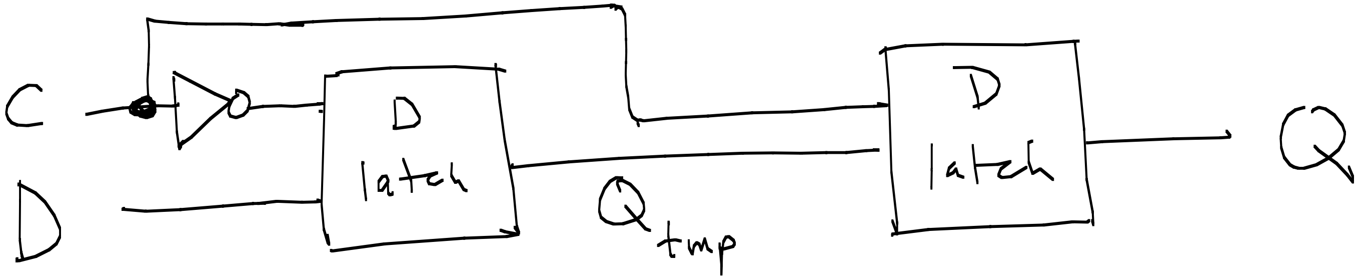
D flip flop ("falling edge triggered")



example



D flip flop ("rising edge triggered")



By putting the inverter on the first D latch, we would make Q change its value on the rising edge of the clock.

Clock cycle must be long enough to allow all gates to stabilize.

Clock synchronizes all flipflops, allowing us to treat time as a sequence of discrete read/write steps (hence 'sequential circuit')

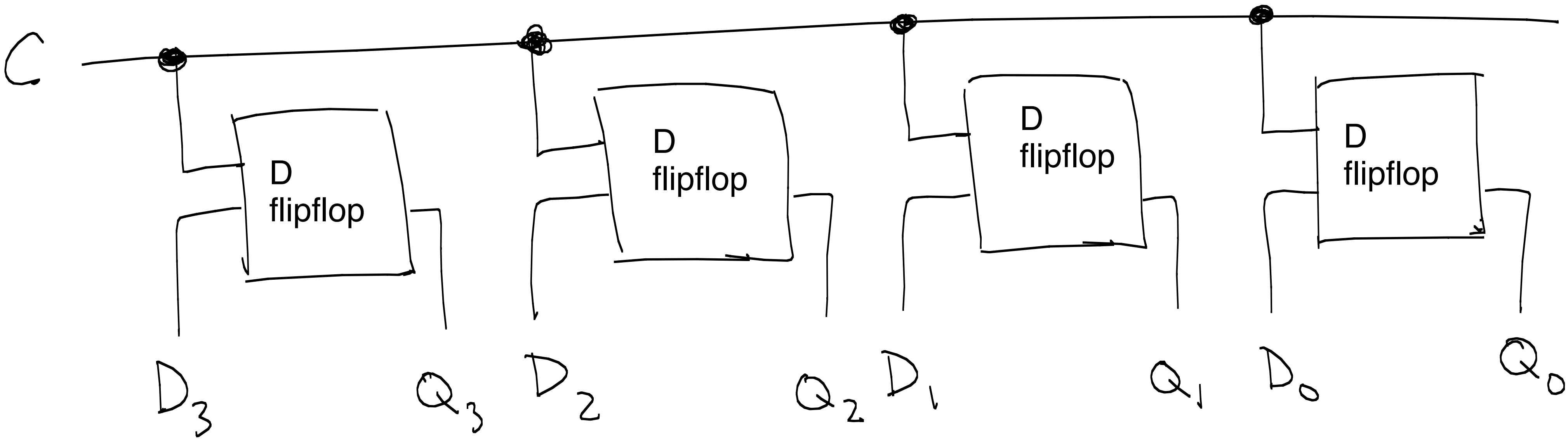
From now on,

- we ignore all variations within a clock cycle
  e.g. carries in the adder.

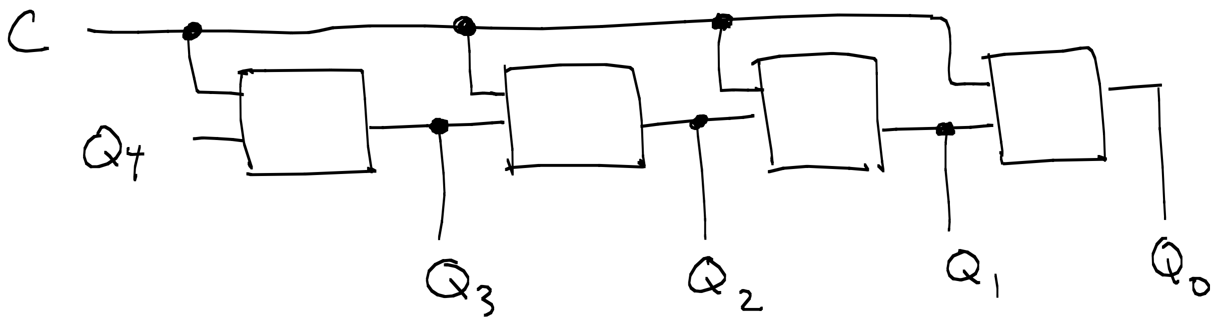
- we work only with D flipflops (no more latches)

Register

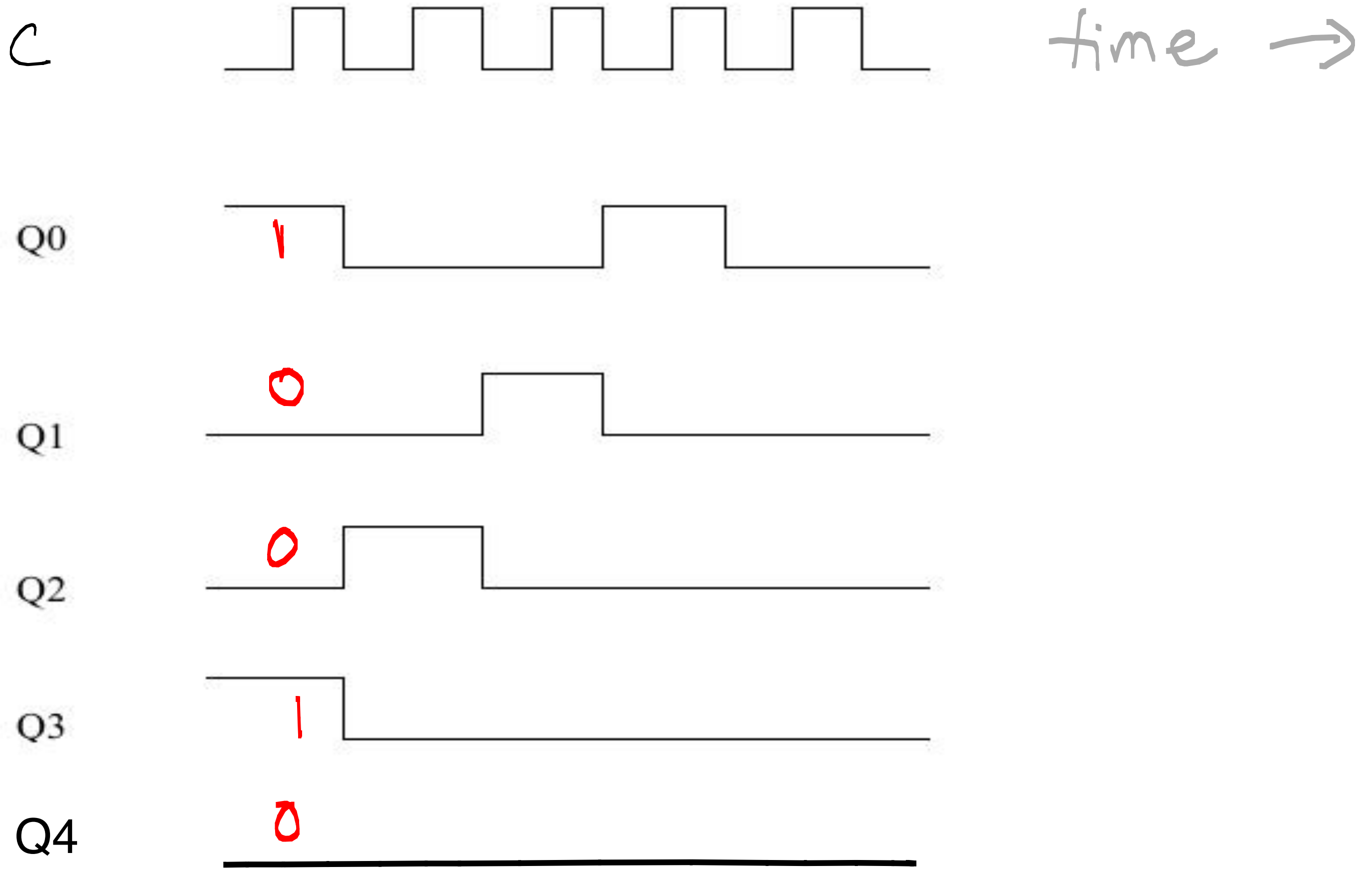
(set of flipflops that are read/written together)



Shift Right Register (falling edge)



Example:



suppose at t = 0.

(Q4, Q3, Q2, Q1, Q0) is (0, 1,0,0,1)

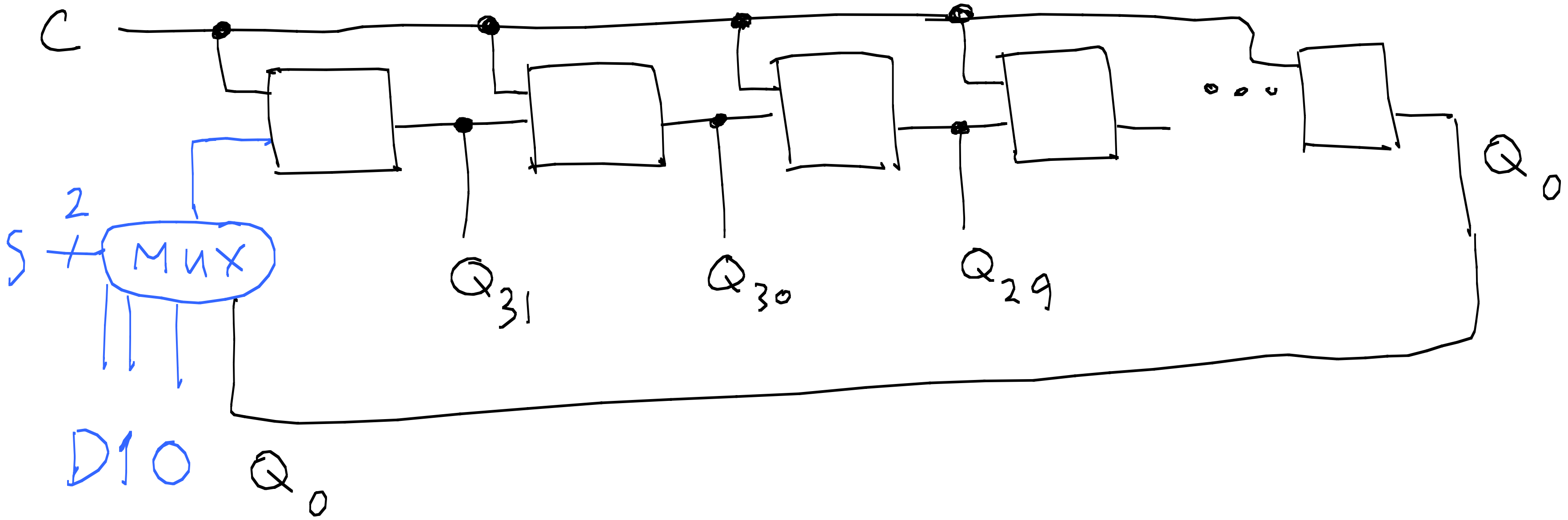
Q4 remains at 0 for the five clock pulses shown.

What happens at each falling edge of clock ?

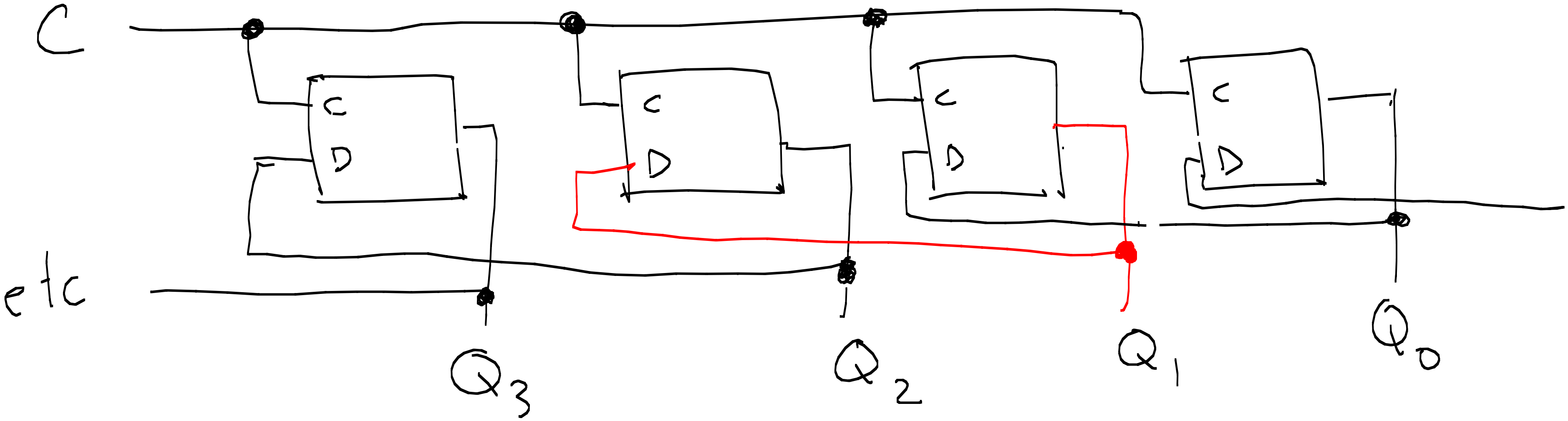
Shift Right Register

We can make Q4 have other values e.g. D (variable), 1, 0, Q0.

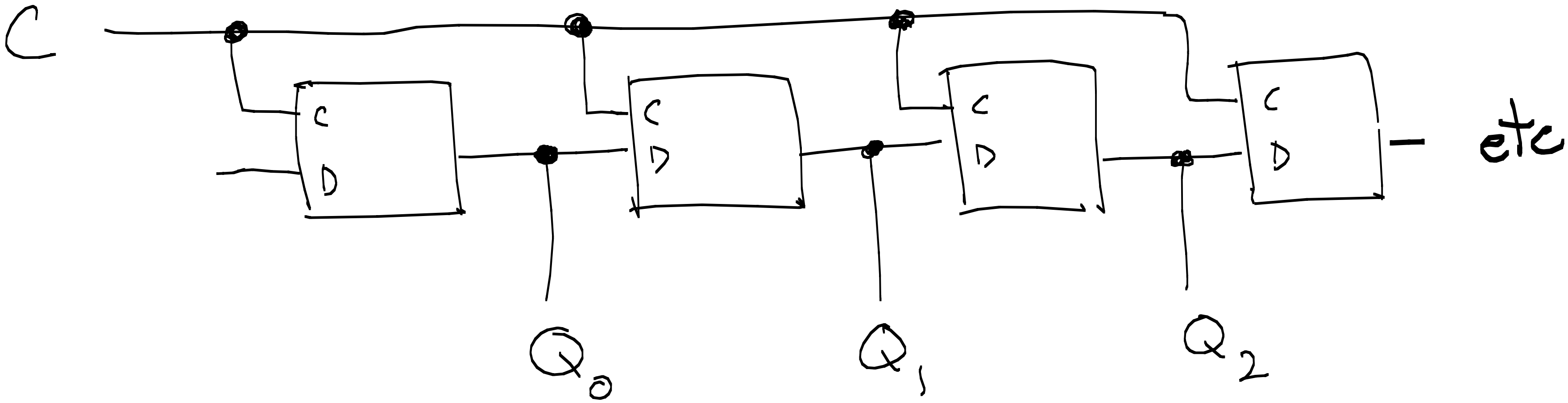
We can then *select* which of these gets put into the MSB.



Shift Left Register



Alternatively, physically order the flipflops in the opposite order



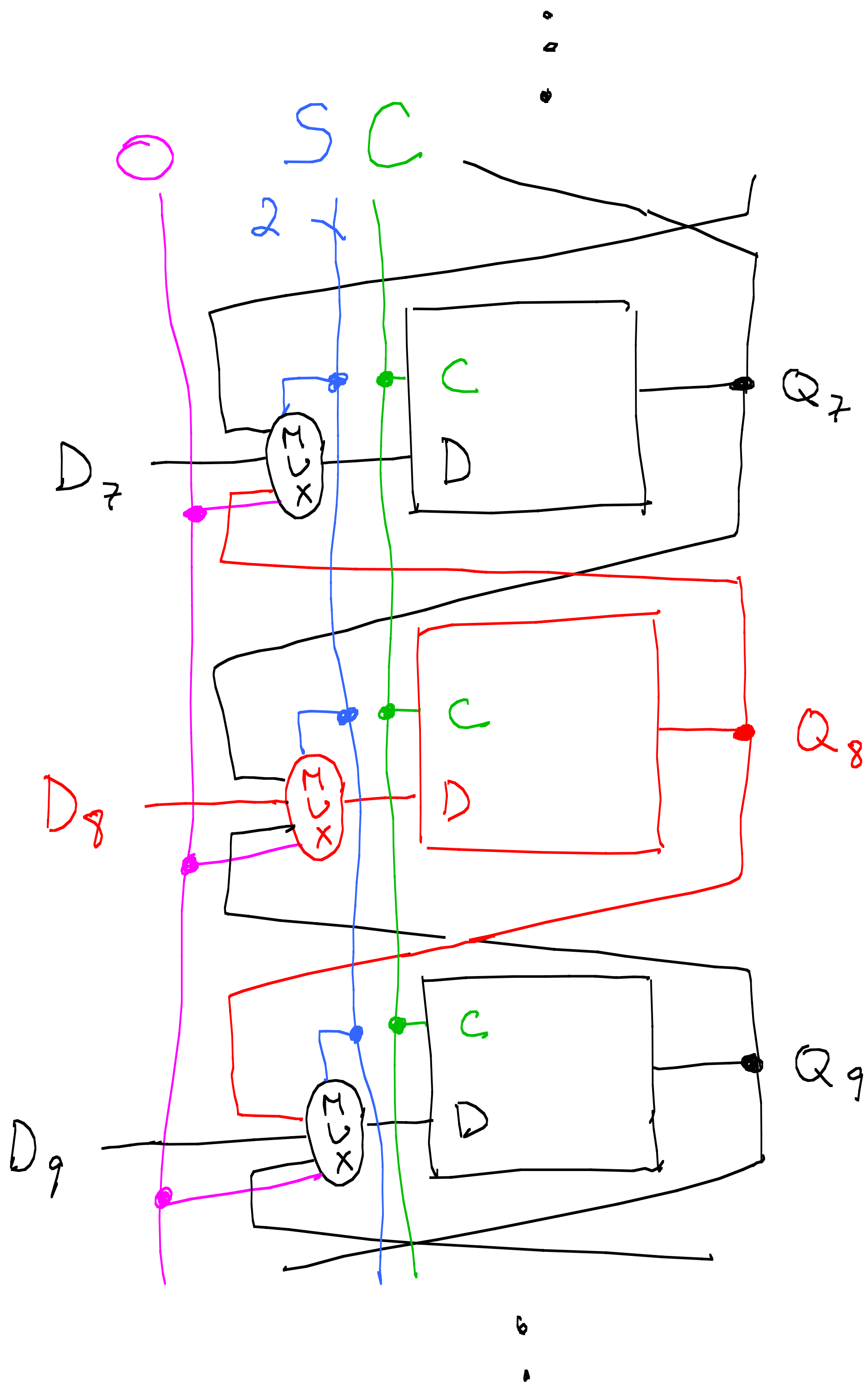
Select from:

- shift left

- shift right

- write data

- clear


0
S
C
2
MUX
C
D
D7
Q7
Q8
D8
Q9
D9