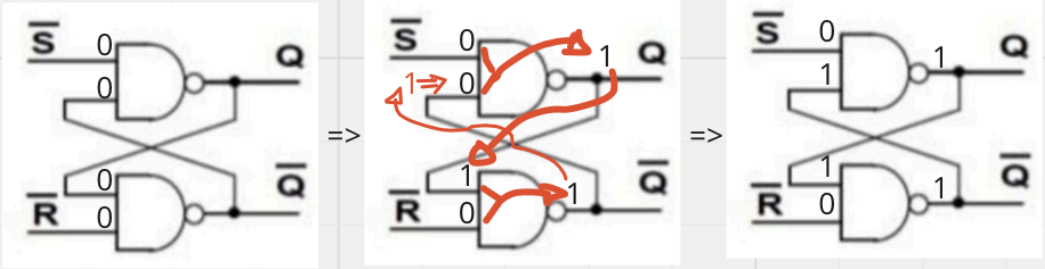
All the memory chips rely on the concept of time introduced into the circuit. Which makes these chips “sequential” instead of “combinational”.

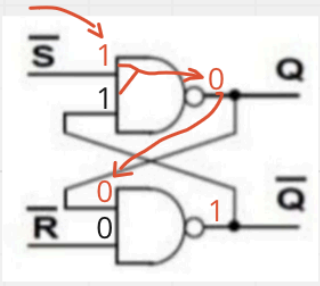
Sequential chips will use a **clock** as an input. From a logic perspective, the clock is assumed **built-in** (it depends on the physics of semiconductors: resistors, capacitors, oscillators..., it will be considered out of the scope for “Elements of Computing System”)

In the book, the D flip flop is also considered **built-in**, (due to the limitation in the HDL of not being capable of recursiveness--mapping the output of one chip into other chips that serve as inputs again, that is, a feedback loop).
However, to understand better how the flip flop gets created, it’s good to see the videos “SR latch” and then “D flip flop” in the references.

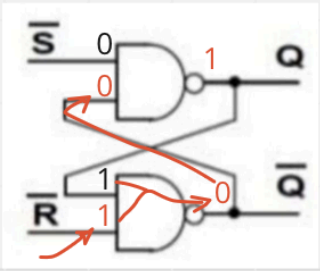
SR LATCH

"invalid" state

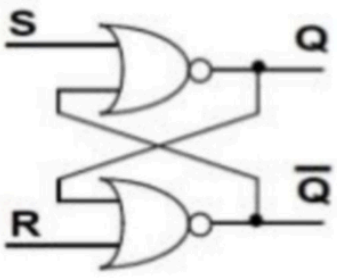




from these "balanced" states, the only remaining option is when both S and R are turn to 1,
in this case the chip just returns the same previous state.

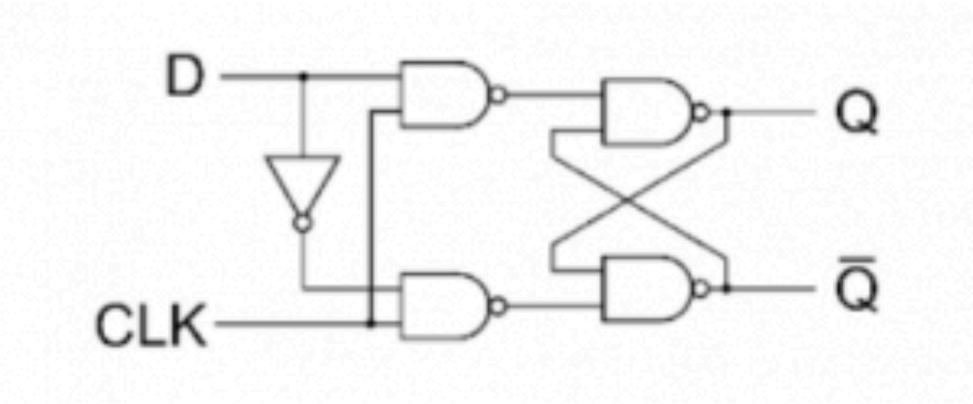


atch



S	R	Q	$\overline{Q}$	Comments
0	0	NC	NC	No Change (remain present state)
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	INVALID

Data flip flop:



Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

To remember:

- Latches serve to store a bit-state. That's the "No Change" value in their truth tables.
- When a latch is regulated by a clock input, it becomes a **flip-flop.**