CS2100 COMPUTER ORGANISATION

http://www.comp.nus.edu.sg/~cs2100/

Lecture #19

Sequential Logic

NUS National University of Singapore | School of Computing

https://app.sli.do/event/aanrHmK6Geu3scZbeLh6wx

Lecture #19: Sequential Logic (1/2)

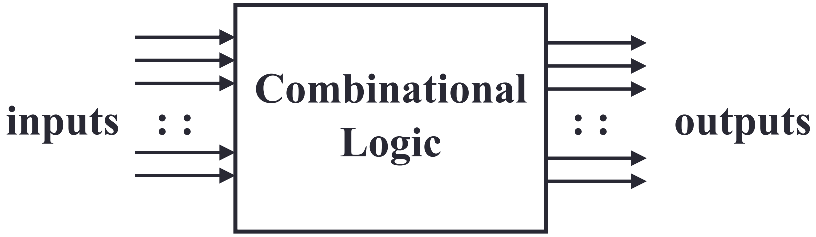
1. Introduction
2. Memory Elements
3. Latches
   - 3.1 *S-R* Latch
   - 3.2 *D* Latch
4. Flip-flops
   - 4.1 *S-R* Flip-flop
   - 4.2 *D* Flip-flop
   - 4.3 *J-K* Flip-flop
   - 4.4 *T* Flip-flop

Lecture #19: Sequential Logic (2/2)

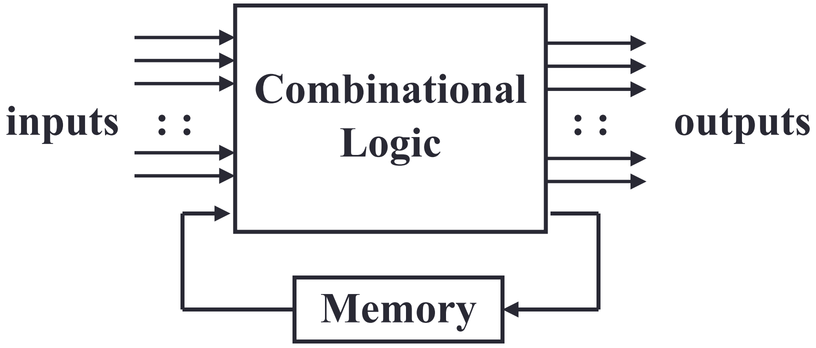
1. Introduction (1/2)

- Two classes of logic circuits
  - Combinational
  - Sequential

- Combinational Circuit
  - Each output depends entirely on the immediate (present) inputs.



- Sequential Circuit
  - Each output depends on both present inputs and state.



1. Introduction (2/2)

- Two types of sequential circuits:
  - Synchronous: outputs change only at specific time
  - Asynchronous: outputs change at any time
- Multivibrator: a class of sequential circuits
  - Bistable (2 stable states)
  - Monostable or one-shot (1 stable state)
  - Astable (no stable state)
- Bistable logic devices
  - Latches and flip-flops.
  - They differ in the methods used for changing their state.

2. Memory Elements (1/3)

- Memory element: a device which can remember value indefinitely, or change value on command from its inputs.

- Characteristic table:

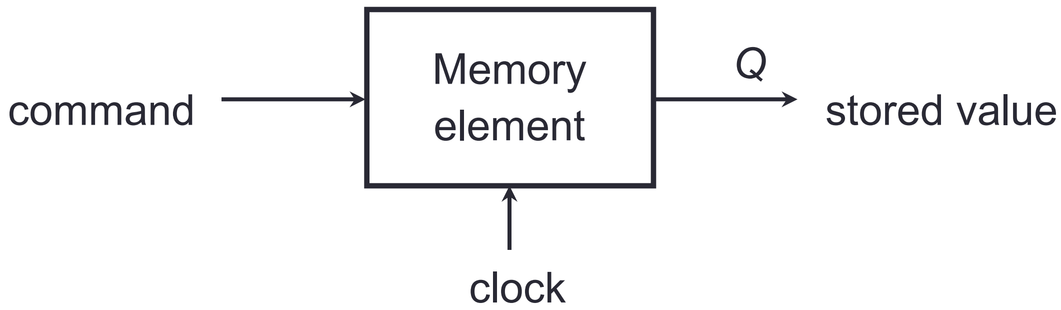
Command (at time *t*)	*Q(t)*	*Q(t+1)*
Set	X	1
Reset	X	0
Memorise / No Change	0	0
	1	1

Q(t) or ***Q***: current state

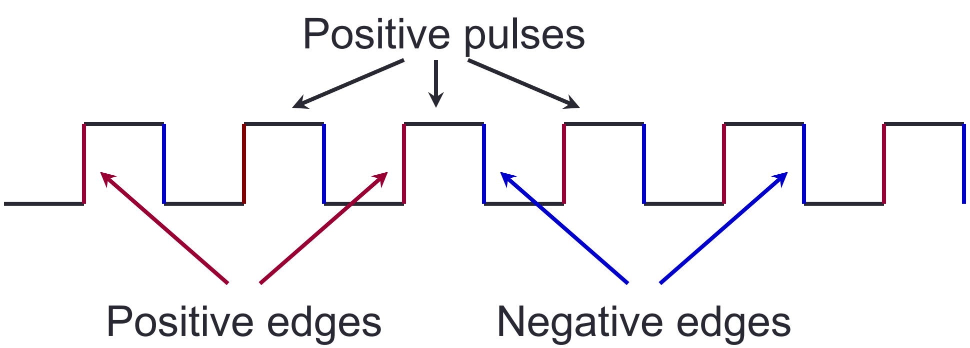
Q(t+1) or $\boldsymbol{Q^{+}}$: next state

2. Memory Elements (2/3)

- Memory element with clock.

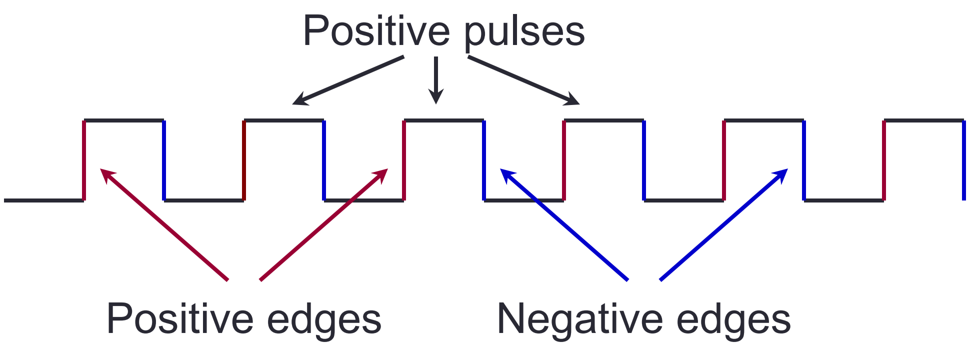


- Clock is usually a square wave.



2. Memory Elements (3/3)

- Two types of triggering/activation
  - Pulse-triggered
  - Edge-triggered



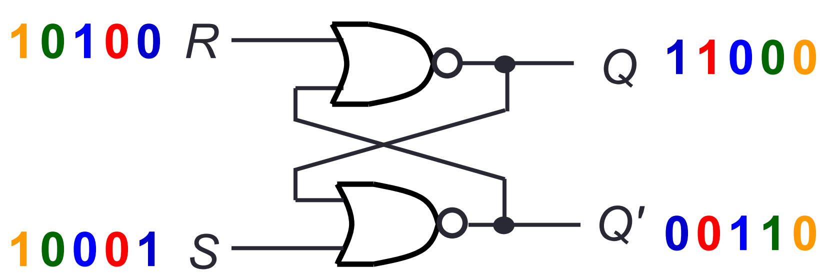
- Pulse-triggered
  - Latches
  - ON = 1, OFF = 0
- Edge-triggered
  - Flip-flops
  - Positive edge-triggered (ON = from 0 to 1; OFF = other time)
  - Negative edge-triggered (ON = from 1 to 0; OFF = other time)

3.1 *S*-*R* Latch (1/3)

- Two inputs: *S* and *R*.
- Two complementary outputs: *Q* and *Q'*.
  - When *Q* = HIGH, we say latch is in SET state.
  - When *Q* = LOW, we say latch is in RESET state.
- For active-high input *S*-*R* latch (also known as NOR gate latch)
  - *R* = HIGH and *S* = LOW → *Q* becomes LOW (RESET state)
  - *S* = HIGH and *R* = LOW → *Q* becomes HIGH (SET state)
  - Both *R* and *S* are LOW → No change in output *Q*
  - Both *R* and *S* are HIGH → Outputs *Q* and *Q'* are both LOW (invalid!)
- Drawback: invalid condition exists and must be avoided.

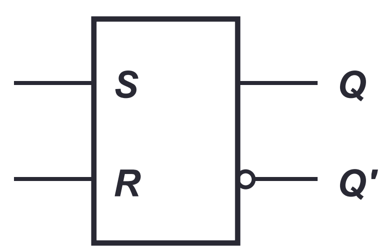
3.1 *S-R* Latch (2/3)

- Active-high input *S-R* latch:

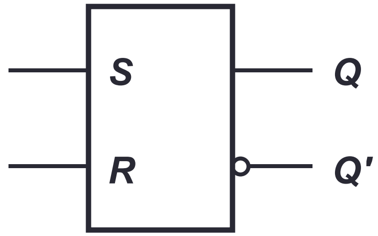


S	*R*	*Q*	*Q'*	
1	0	1	0	initial
0	0	1	0	(afer *S*=1, *R*=0)
0	1	0	1	
0	0	0	1	(after *S*=0, *R*=1)
1	1	0	0	invalid!

- Block diagram:



3.1 *S-R* Latch (3/3)



- Characteristic table for active-high input *S-R* latch:

S	***R***	***Q***	***Q'***	
0	**0**	**NC**	**NC**	**No change. Latch remained in present state.**
1	**0**	**1**	**0**	**Latch SET.**
0	**1**	**0**	**1**	**Latch RESET.**
1	**1**	**0**	**0**	**Invalid condition.**

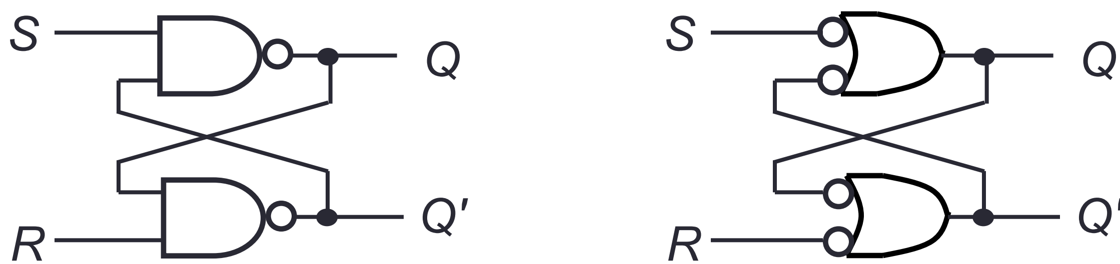
S	***R***	***Q(t+1)***	
0	**0**	***Q(t)***	**No change**
0	**1**	**0**	**Reset**
1	**0**	**1**	**Set**
1	**1**	**indeterminate**	

$$Q(t+1) = S + R' \cdot Q$$

$$S \cdot R = 0$$

3.1 Active-Low *S*-*R* Latch

- (You may skip this slide.)
- What we have seen is active-high input *S*-*R* latch.
- There are active-low input *S*-*R* latches, where NAND gates are used instead. See diagram on the left below.



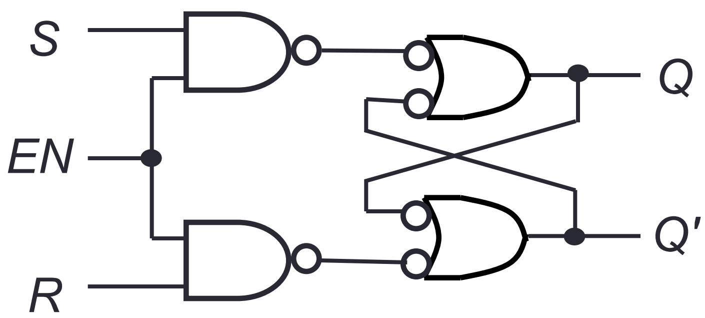
- In this case,
  - when R=0 and S=1, the latch is reset (i.e. Q becomes 0)
  - when R=1 and S=0, the latch is set (i.e. Q becomes 1)
  - when S=R=1, it is a no-change command.
  - when S=R=0, it is an invalid command.

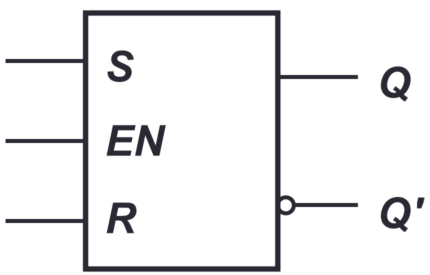
(Sometimes, the inputs are labelled as S' and R'.)

- Sometimes, we use the alternative gate diagram for the NAND gate. See diagram on the right above. (This appears in more complex latches/flip-flops in the later slides.)

3.1 Gated *S-R* Latch

- *S-R* latch + *enable input* (*EN*) and 2 NAND gates → a gated *S-R* latch.

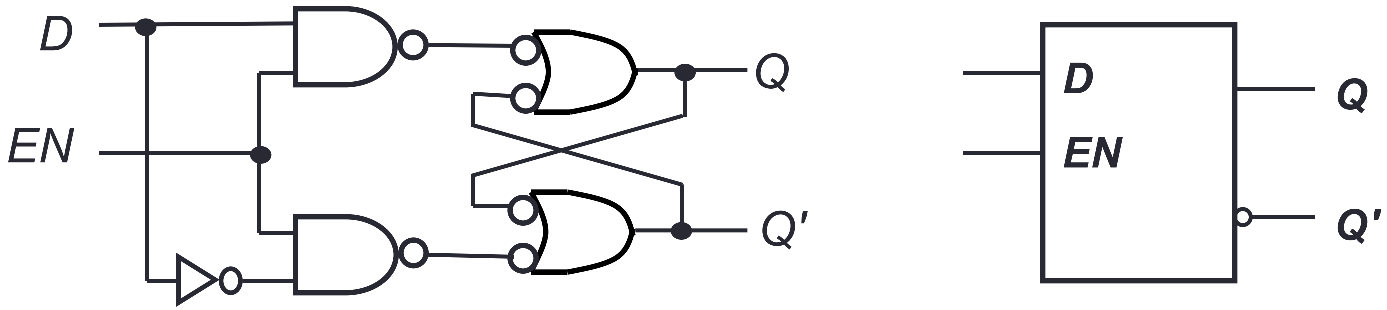




- Outputs change (if necessary) only when *EN* is high.

3.2 Gated *D* Latch (1/2)

- Make input *R* equal to *S'* → gated *D* latch.
- *D* latch eliminates the undesirable condition of invalid state in the *S*-*R* latch.



3.2 Gated *D* Latch (2/2)

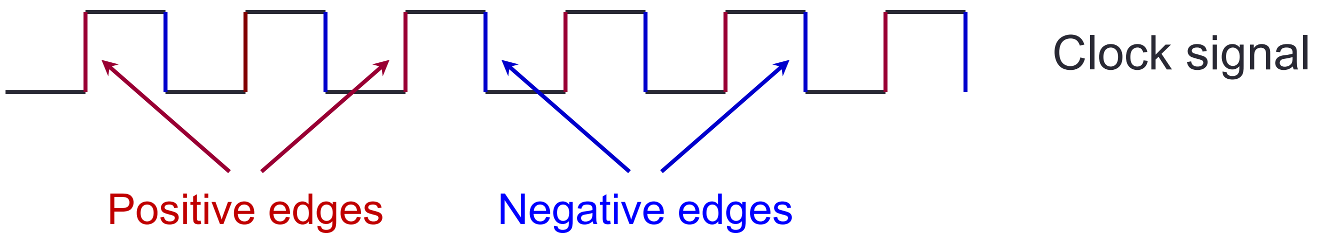
- When *EN* is high,
  - *D* = HIGH → latch is SET
  - *D* = LOW → latch is RESET
- Hence when EN is high, *Q* "follows" the *D* (data) input.
- Characteristic table:

EN	*D*	*Q(t+1)*	
1	0	0	Reset
1	1	1	Set
0	X	*Q(t)*	No change

When *EN*=1, ***Q(t+1) = D***

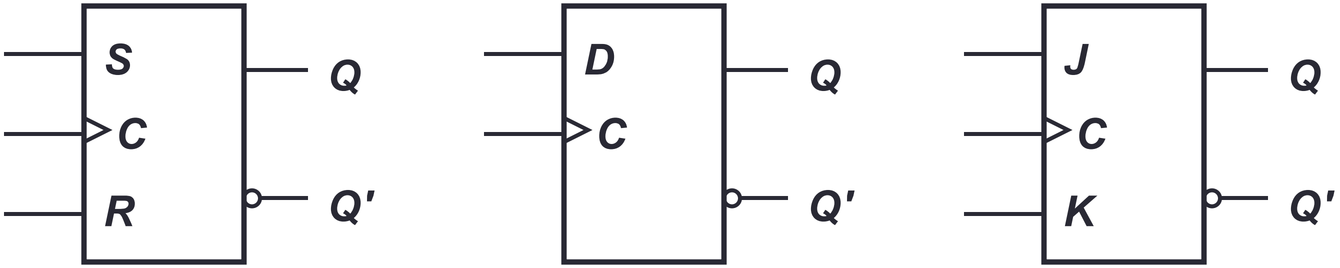
4. Flip-flops (1/2)

- Flip-flops are synchronous bistable devices.
- Output changes state at a specified point on a triggering input called the clock.
- Change state either at the positive (rising) edge, or at the negative (falling) edge of the clock signal.

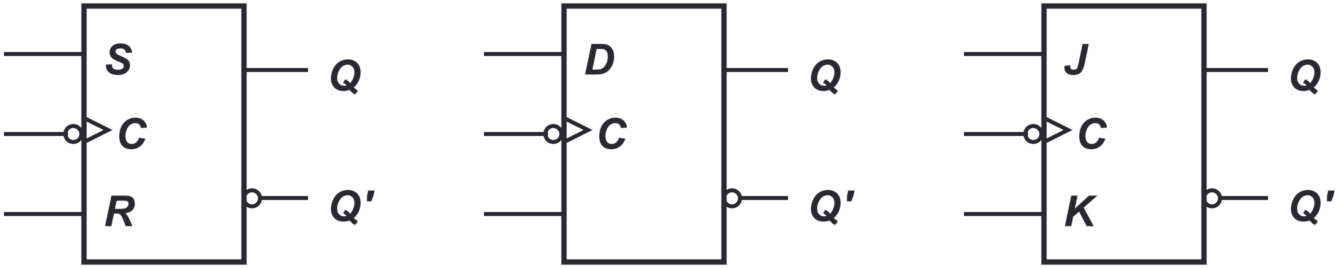


4. Flip-flops (2/2)

- *S-R* flip-flop, *D* flip-flop, and *J-K* flip-flop.
- Note the ">" symbol at the clock input.



Positive edge-triggered flip-flops

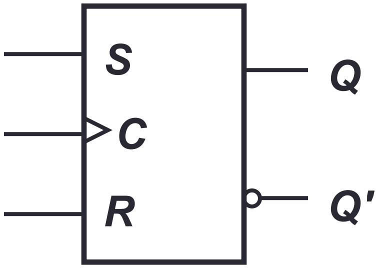


Negative edge-triggered flip-flops

4.1 *S-R* Flip-flop

- *S-R* flip-flop: On the triggering edge of the clock pulse,
  - *R* = HIGH and *S* = LOW ➔ *Q* becomes LOW (RESET state)
  - *S* = HIGH and *R* = LOW ➔ *Q* becomes HIGH (SET state)
  - Both *R* and *S* are LOW ➔No change in output *Q*
  - Both *R* and *S* are HIGH ➔Invalid!

- Characteristic table of positive edge-triggered *S-R* flip-flop:



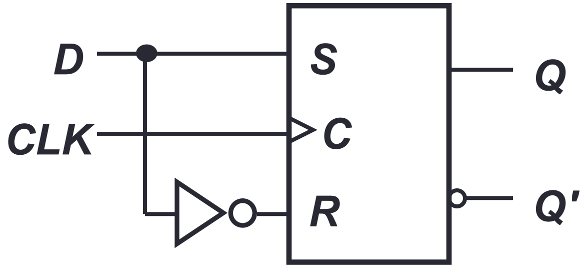
S	*R*	*CLK*	*Q(t+1)*	Comments
0	0	X	*Q(t)*	No change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	?	Invalid

X = irrelevant ("don't care")

↑ = clock transition LOW to HIGH

4.2 *D* Flip-flop (1/2)

- *D* flip-flop: Single input *D* (data). On the triggering edge of the clock pulse,
  - *D* = HIGH ➔ *Q* becomes HIGH (SET state)
  - *D* = LOW ➔ *Q* becomes LOW (RESET state)
- Hence, *Q* "follows" *D* at the clock edge.
- Convert *S-R* flip-flop into a *D* flip-flop: add an inverter.



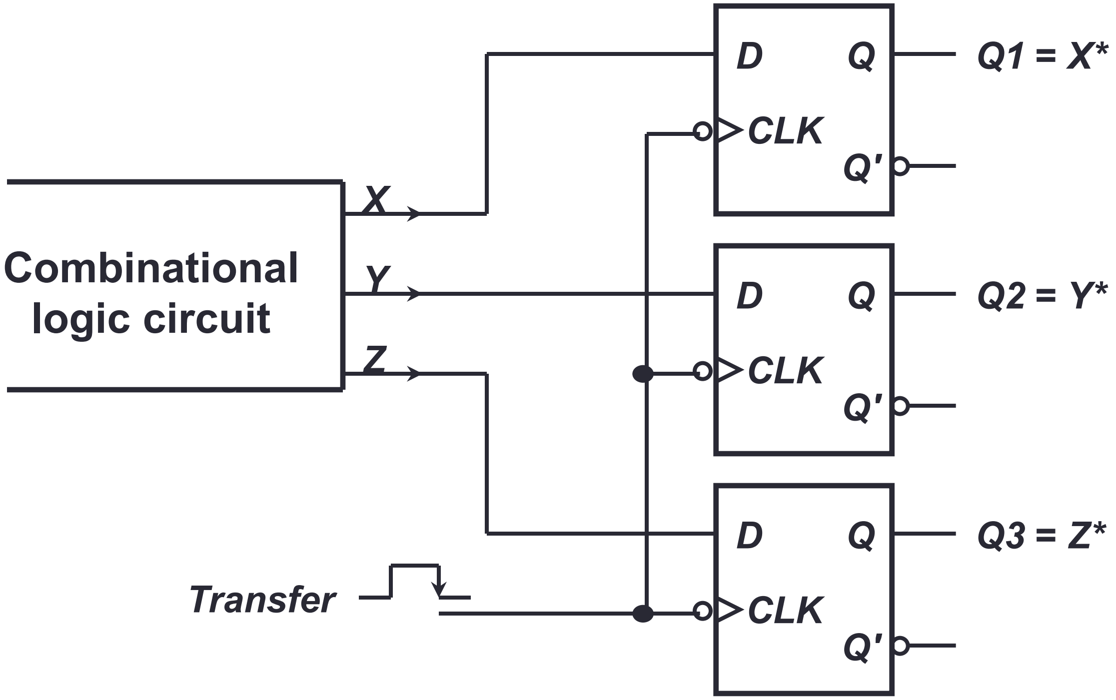
A positive edge-triggered D flip-flop formed with an S-R flip-flop.

D	*CLK*	*Q(t+1)*	Comments
1	↑	1	Set
0	↑	0	Reset

↑ = clock transition LOW to HIGH

4.2 *D* Flip-flop (2/2)

- Application: Parallel data transfer.
  - To transfer logic-circuit outputs *X*, *Y*, *Z* to flip-flops *Q*1, *Q*2 and *Q*3 for storage.



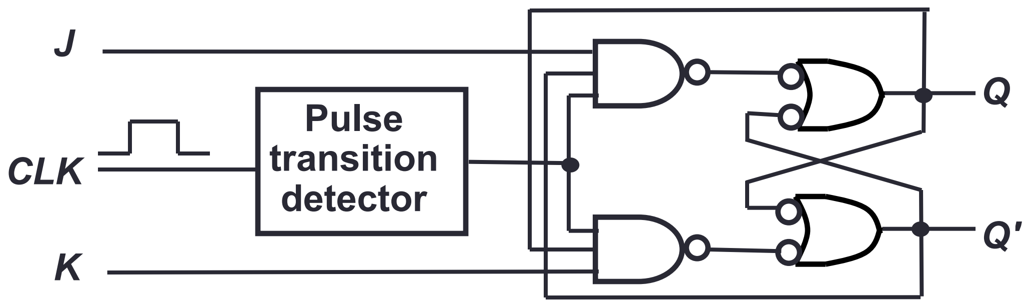
* After occurrence of negative-going transition

4.3 *J-K* Flip-flop (1/2)

- *J-K* flip-flop: *Q* and *Q'* are fed back to the pulse-steering NAND gates.
- No invalid state.
- Include a toggle state
  - *J* = HIGH and *K* = LOW ➔ *Q* becomes HIGH (SET state)
  - *K* = HIGH and *J* = LOW ➔ *Q* becomes LOW (RESET state)
  - Both *J* and *K* are LOW ➔No change in output *Q*
  - Both *J* and *K* are HIGH ➔Toggle

4.3 *J-K* Flip-flop (2/2)

- *J-K* flip-flop circuit:



- Characteristic table:

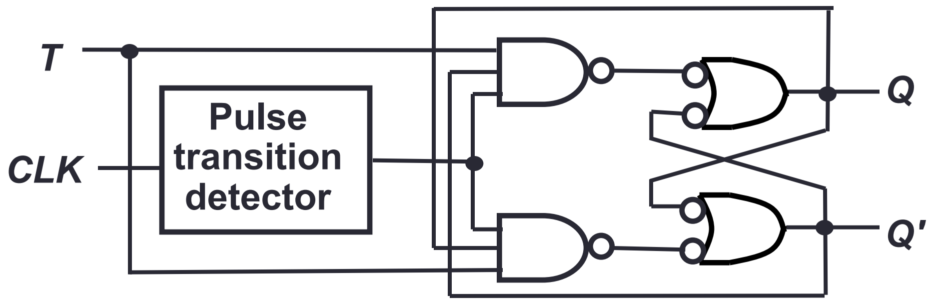
J	K	CLK	Q(t+1)	Comments
0	0	↑	Q(t)	No change
0	1	↑	0	Reset
1	0	↑	1	Set
1	1	↑	Q(t)'	Toggle

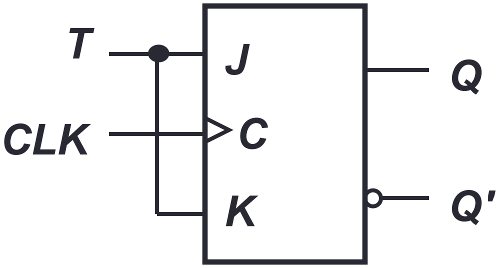
$$Q(t+1) = J \cdot Q' + K' \cdot Q$$

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

4.4 *T* Flip-flop

- *T* flip-flop: Single input version of the *J-K* flip-flop, formed by tying both inputs together.





- Characteristic table:

T	*CLK*	*Q(t+1)*	Comments
0	↑	*Q(t)*	No change
1	↑	*Q(t)'*	Toggle

Q	*T*	*Q(t+1)*
0	0	0
0	1	1
1	0	1
1	1	0

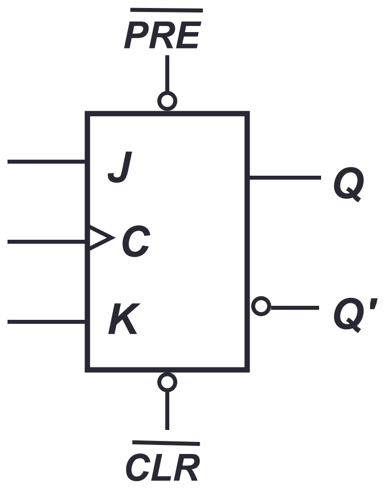
$$Q(t+1) = T \cdot Q' + T' \cdot Q$$

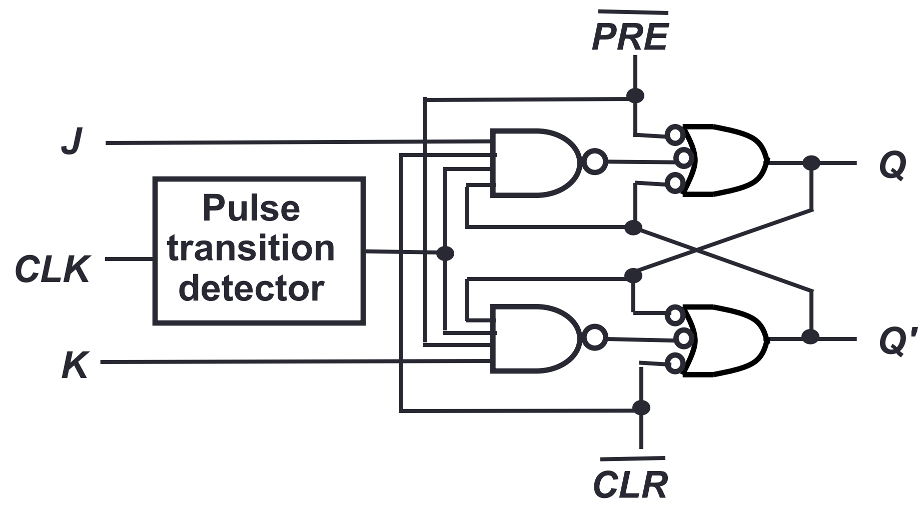
5. Asynchronous Inputs (1/2)

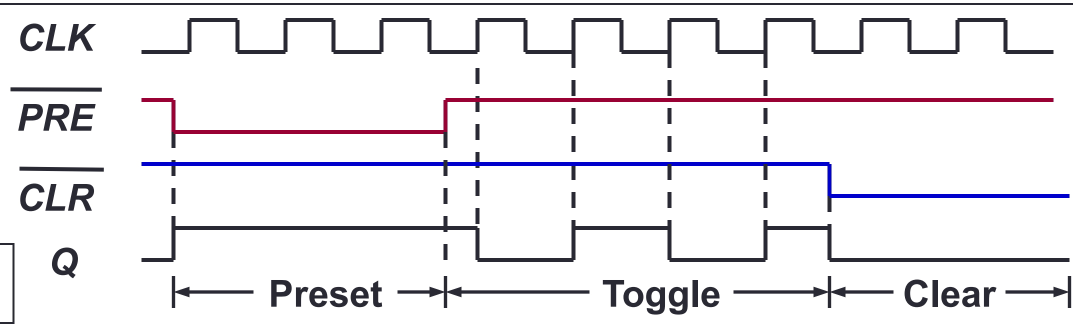
- *S-R*, *D* and *J-K* inputs are synchronous inputs, as data on these inputs are transferred to the flip-flop's output only on the triggered edge of the clock pulse.
- Asynchronous inputs affect the state of the flip-flop independent of the clock; example: *preset* (*PRE*) and *clear* (*CLR*) [or *direct set* (*SD*) and *direct reset* (*RD*)].
- When *PRE*=HIGH, *Q* is immediately set to HIGH.
- When *CLR*=HIGH, *Q* is immediately cleared to LOW.
- Flip-flop in normal operation mode when both *PRE* and *CLR* are LOW.

5. Asynchronous Inputs (2/2)

- A *J-K* flip-flop with active-low PRESET and CLEAR asynchronous inputs.







$J = K =$ HIGH

End of File