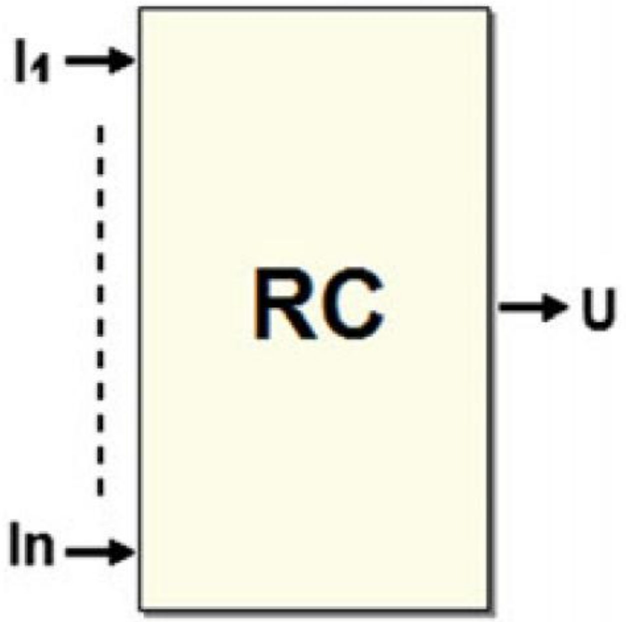
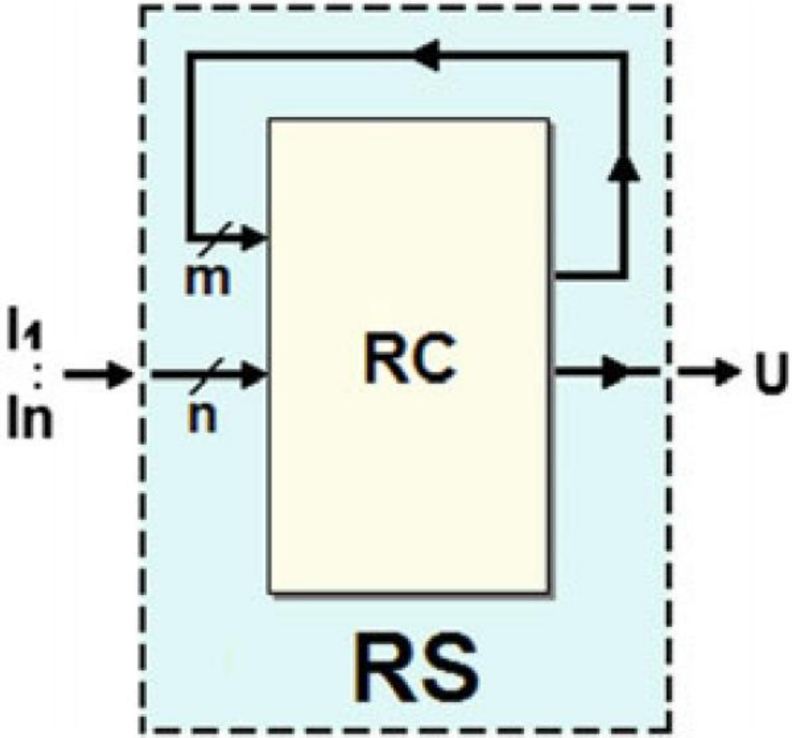
Chapter 5 Introduction to Sequential Networks

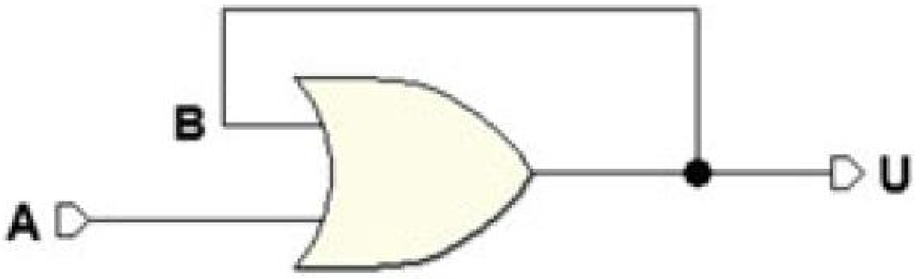
From Combinational Networks to Sequential Networks

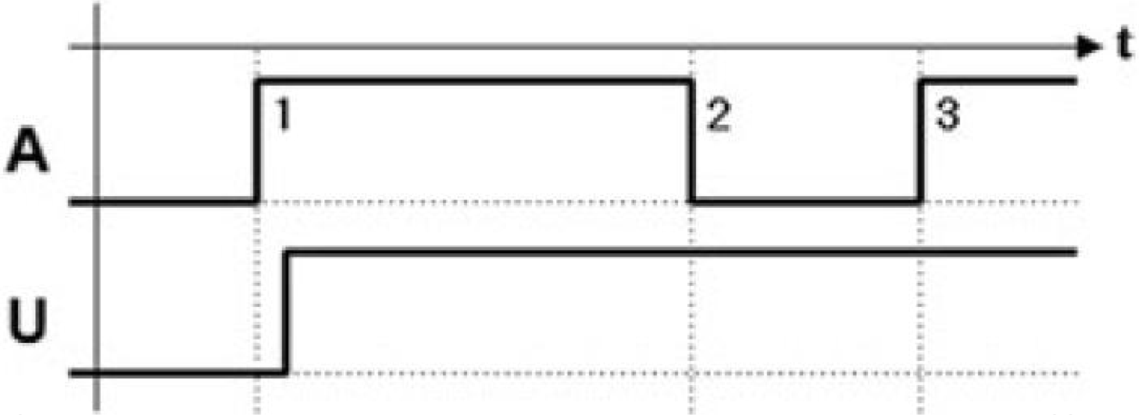


feedback 反饋



$$U = A + U$$

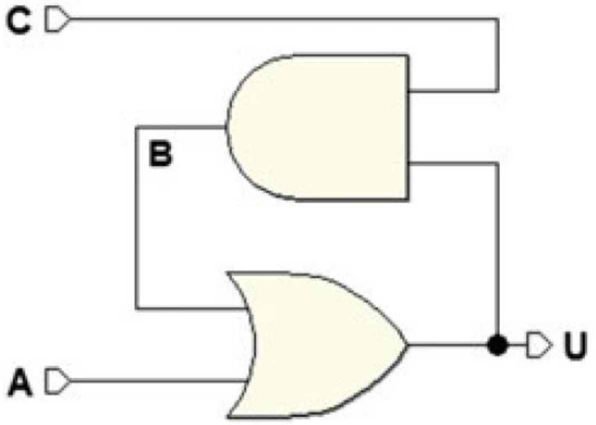


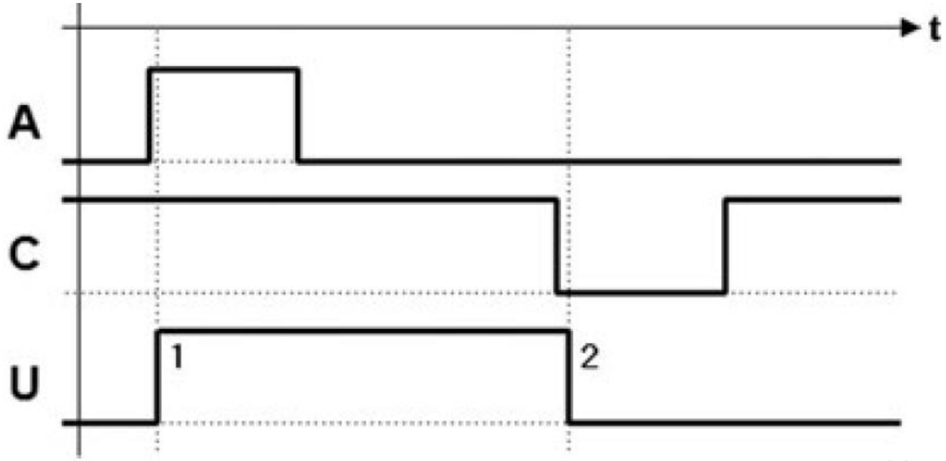


Memorizing an Information Bit: Flip-Flops (觸發器)

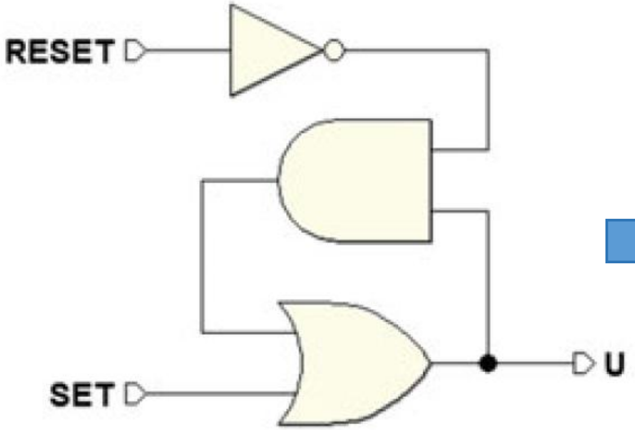
- “A” for memorizing 1 in the output when it is 1. (active-high)
- “C” for memorizing a 0 when it is 0. (active-low)
- This sequential network is one of the ways to create an elementary memory cell, also called bistable element, one-bit register or, more commonly, “flipflop.”
- Flip-flops are the basic logical elements generally used to build sequential digital systems.

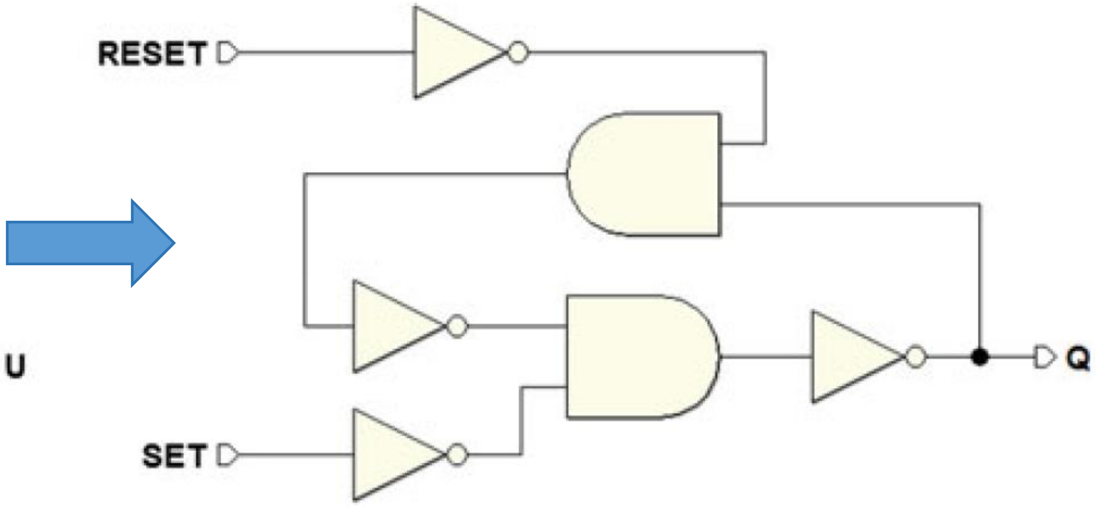
$$U = A + C\,U$$

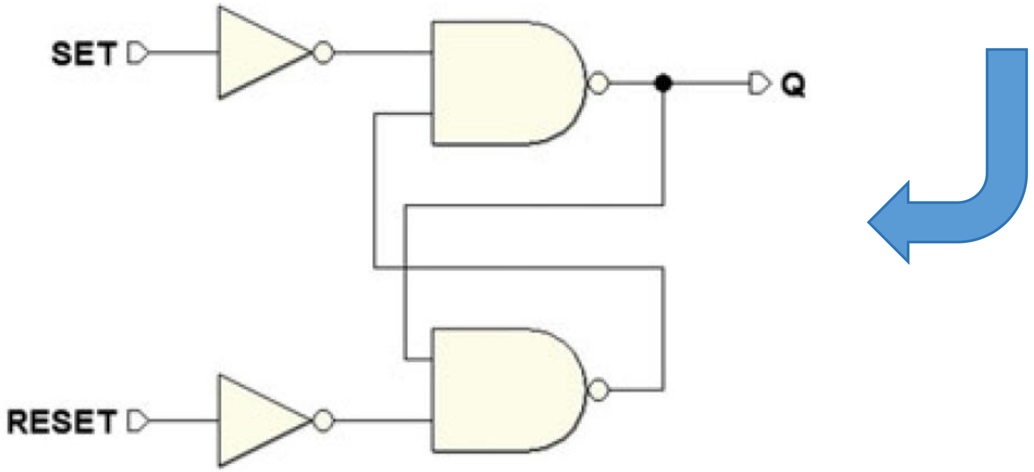




Set-Reset flip-flop





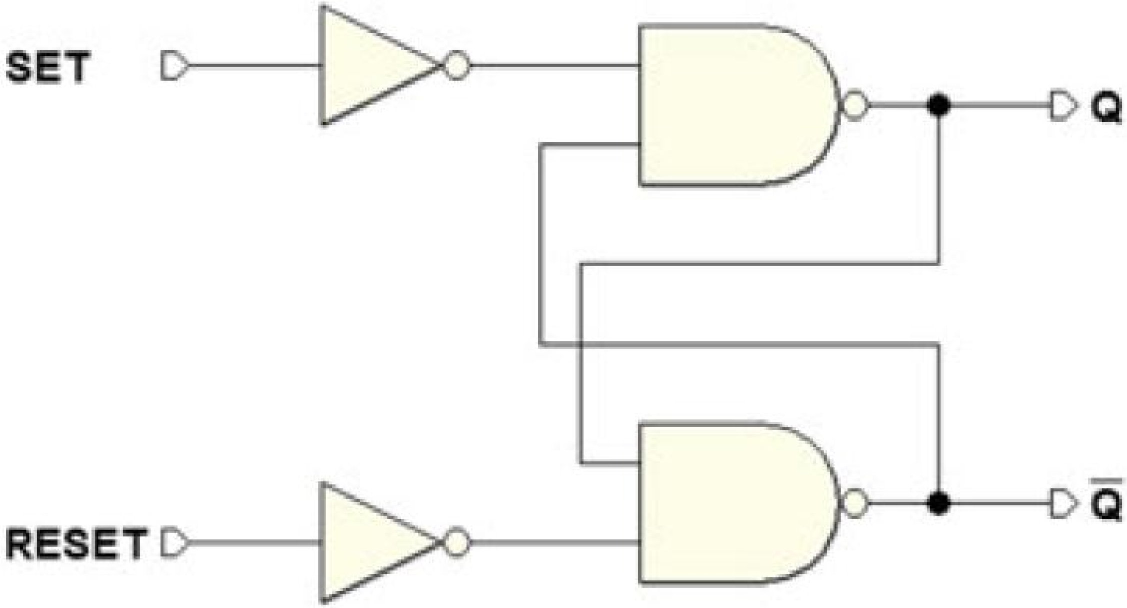


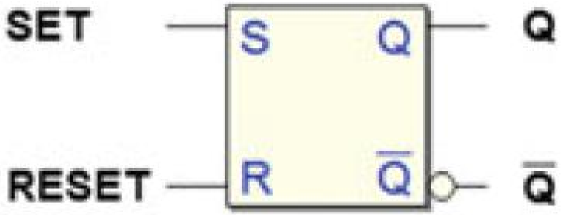
$$Q = \overline{\overline{SET}\ \overline{Q\ \overline{RESET}}}$$

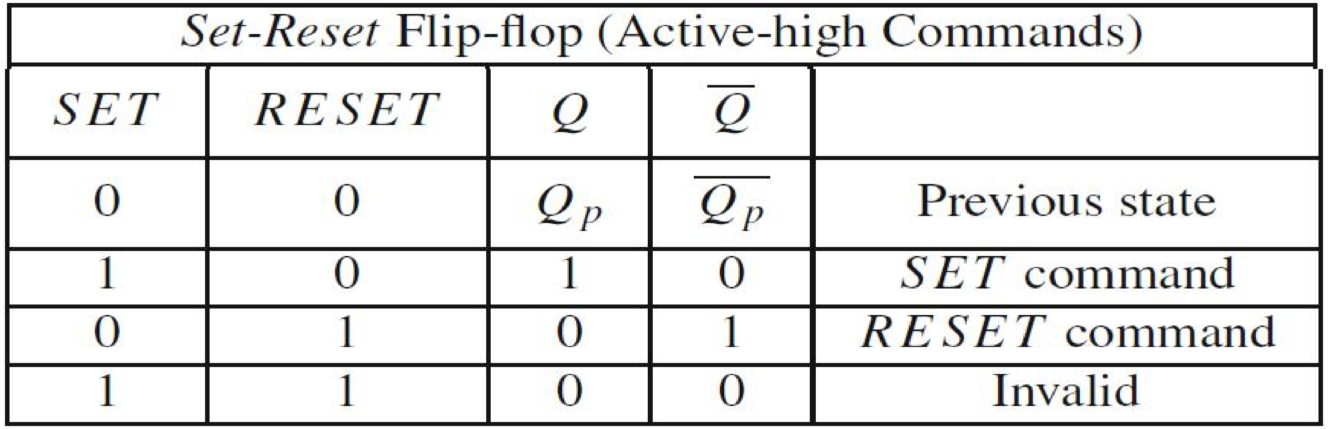
INTRODUCTION TO SEQUENTIAL NETWORKS

Direct Command Flip-Flops

SR Flip-Flop

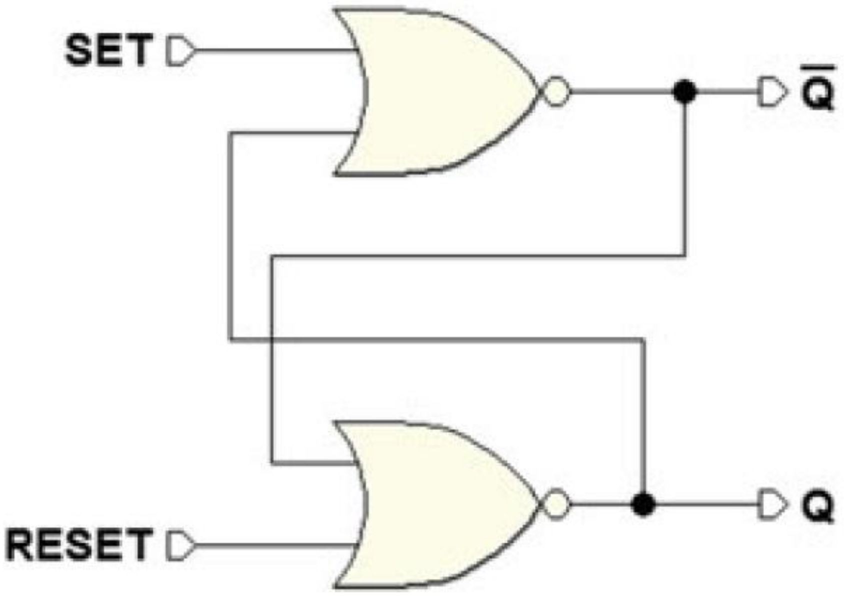






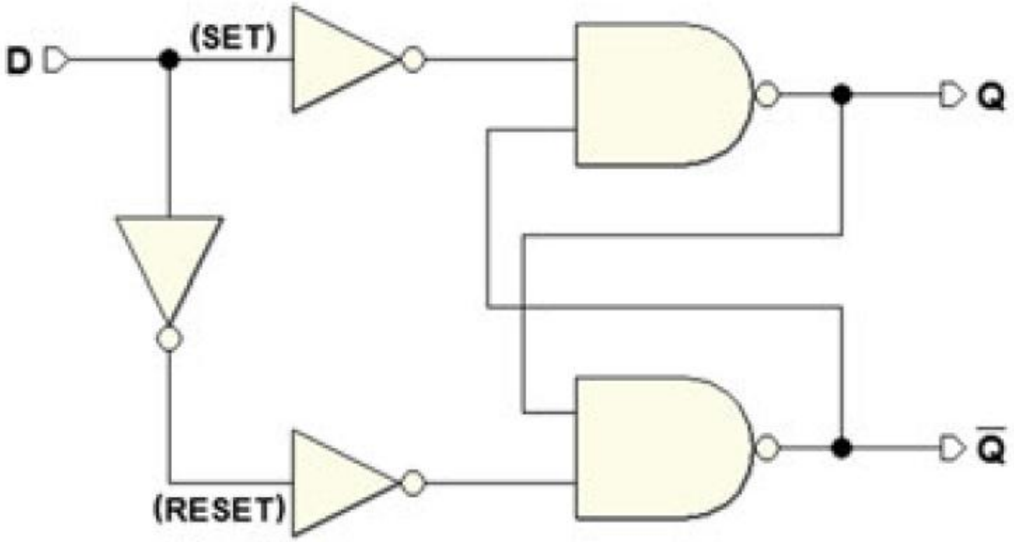
Set-Reset Flip-flop (Active-high Commands)

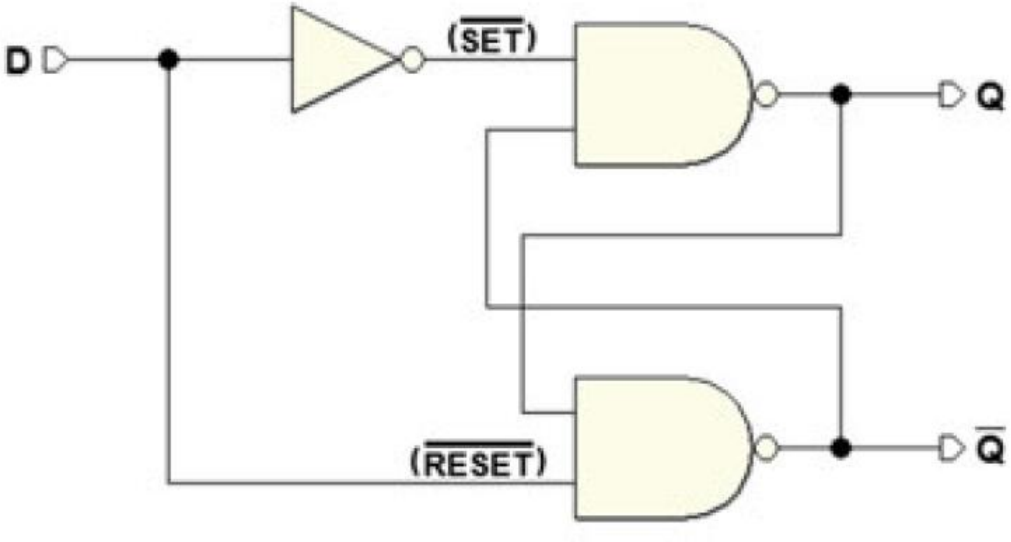
SET	$RESET$	Q	$\overline{Q}$	
0	0	Q_p	$\overline{Q_p}$	Previous state
1	0	1	0	SET command
0	1	0	1	$RESET$ command
1	1	0	0	Invalid

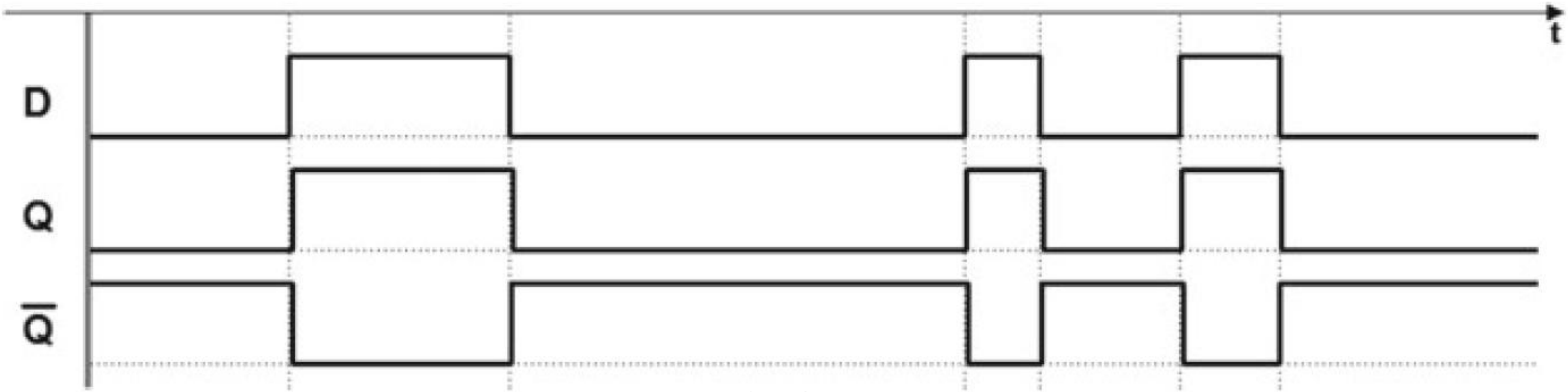


Set-Reset Flip-flop (Active-high Commands)				
SET	$RESET$	Q	$\overline{Q}$	
0	0	Q_p	$\overline{Q_p}$	Previous state
1	0	1	0	SET command
0	1	0	1	$RESET$ command
1	1	0	0	Invalid

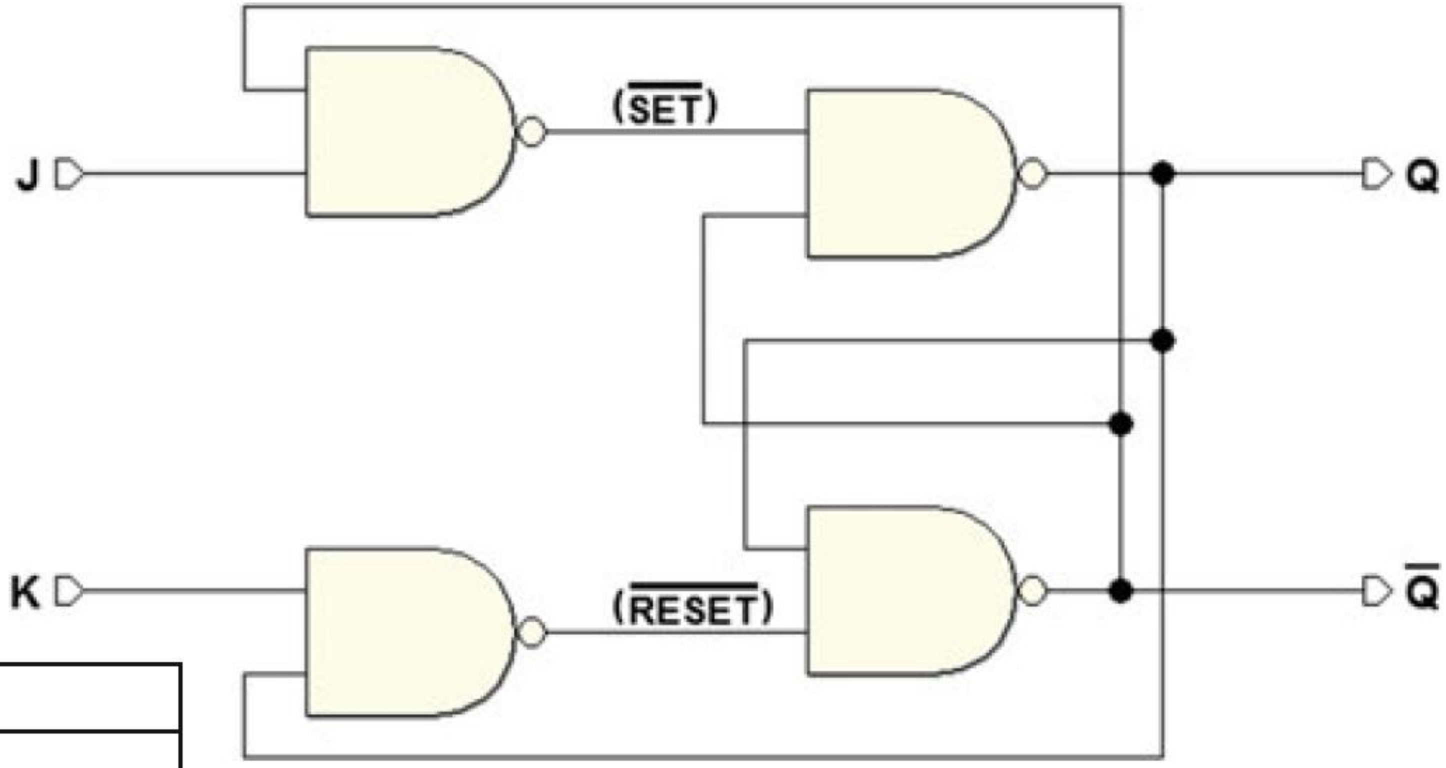
D Flip-Flop


D
(SET)
Q
$\overline{Q}$
(RESET)

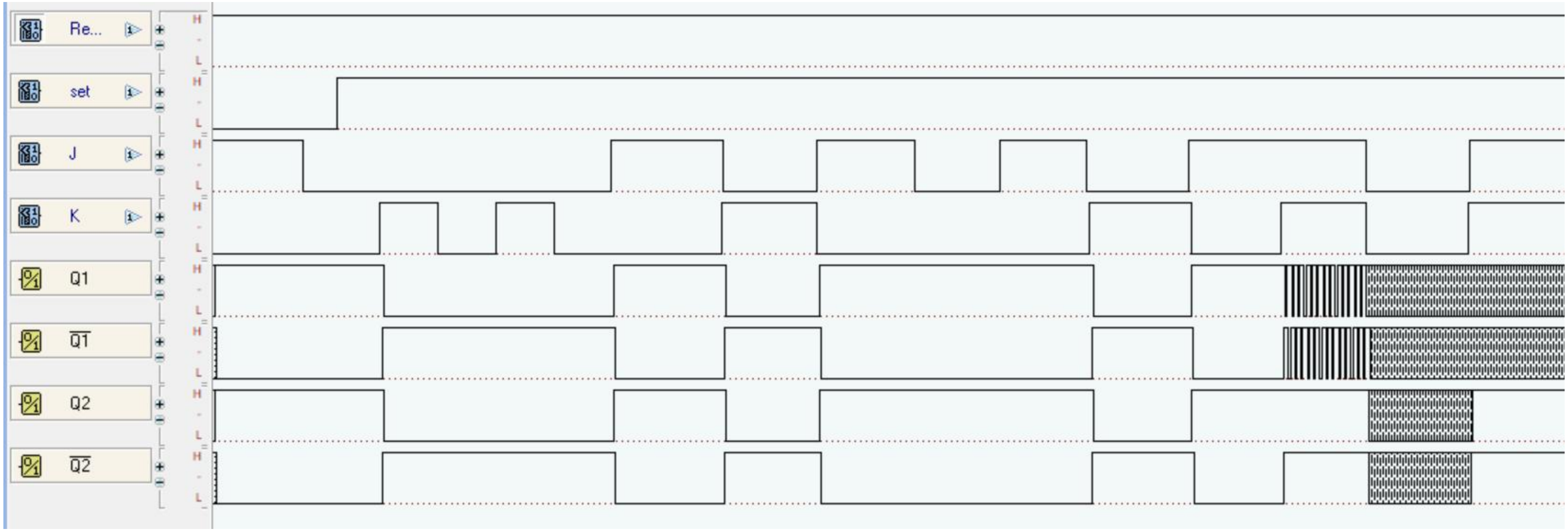

D
($\overline{SET}$)
Q
$\overline{Q}$
($\overline{RESET}$)


t
D
Q
$\overline{Q}$

JK Flip-Flop

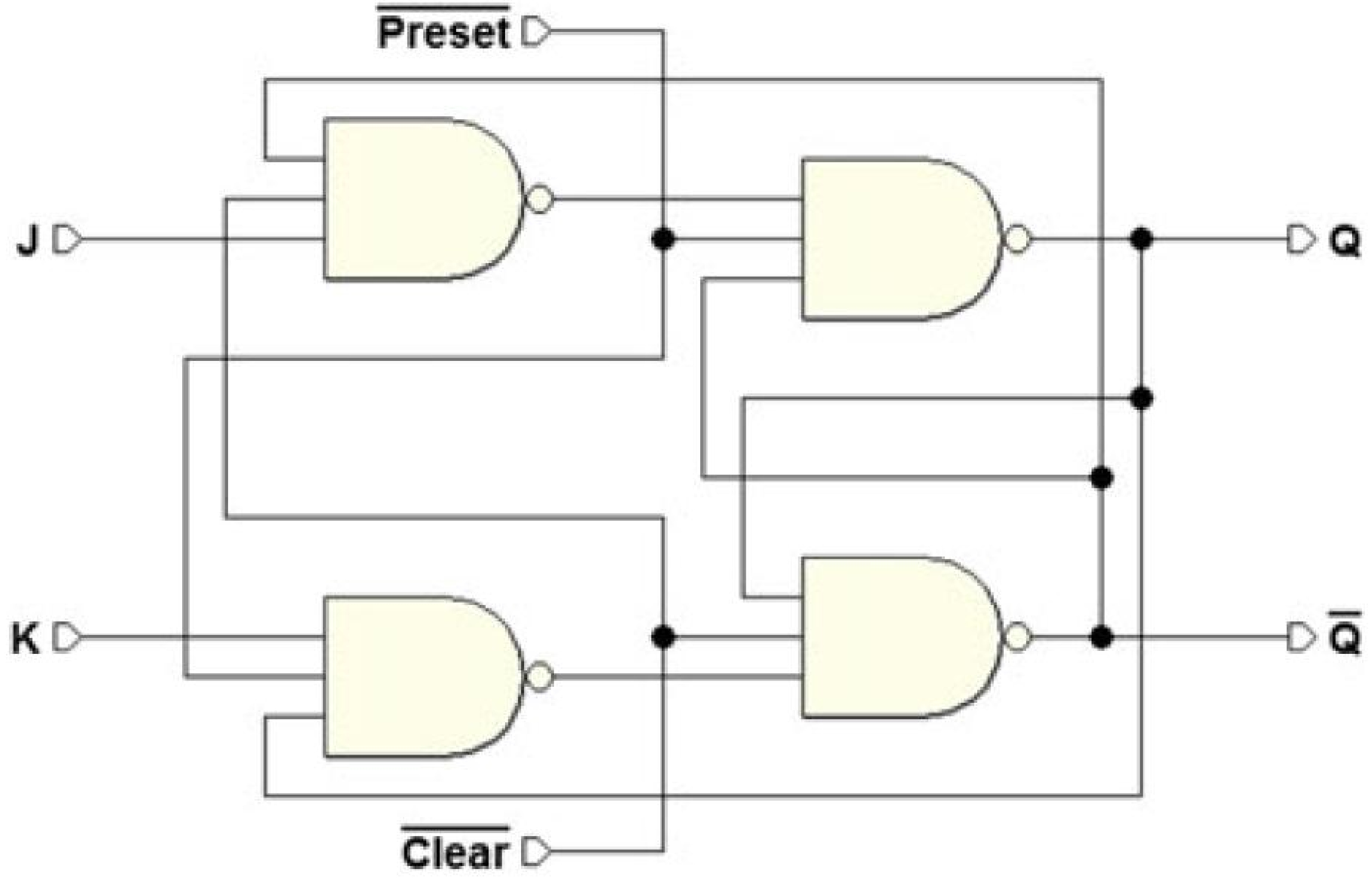


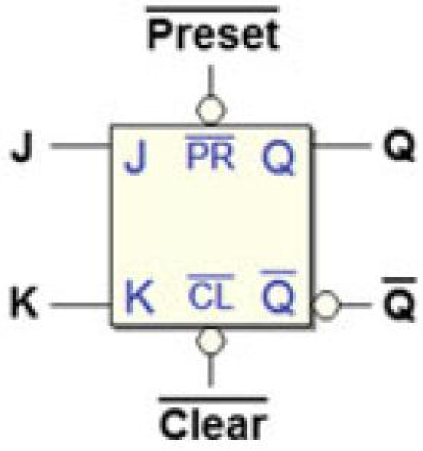
JK Flip-flop				
J	K	Q	$\overline{Q}$	
0	0	Q_p	$\overline{Q_p}$	Previous state
1	0	1	0	SET command
0	1	0	1	$RESET$ command
1	1	$\overline{Q_p}$	Q_p	Toggle



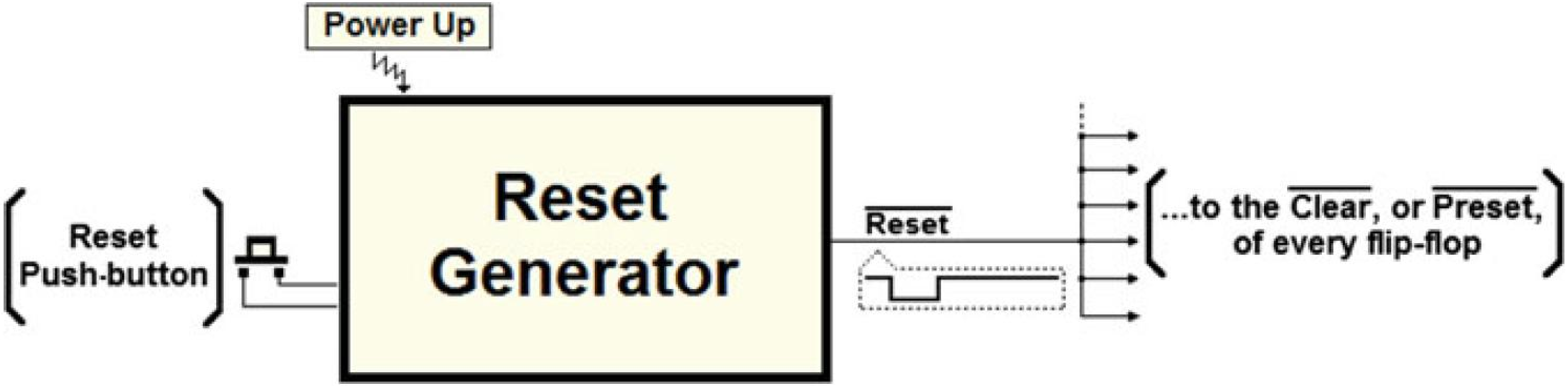
Q1 - JK Flip-Flop
Q2 - SR Flip-Flop

Flip-Flop Initialization Inputs





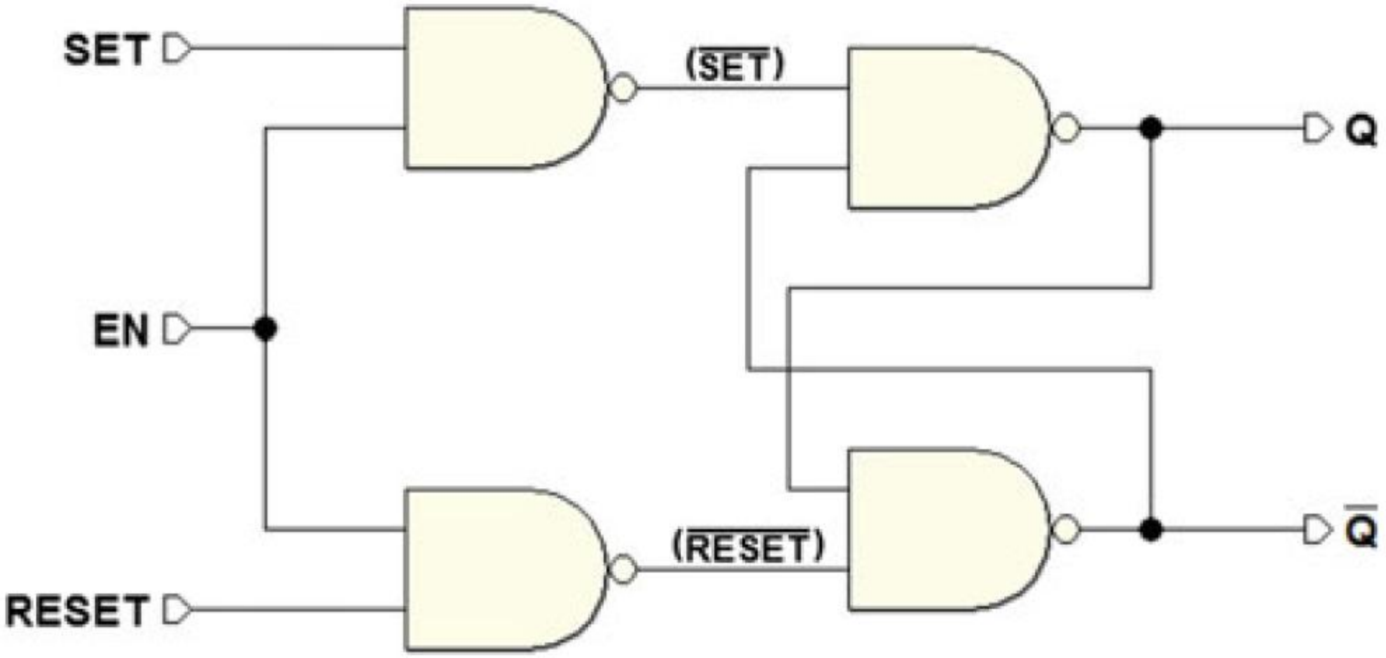
Generating an Initialization Signal

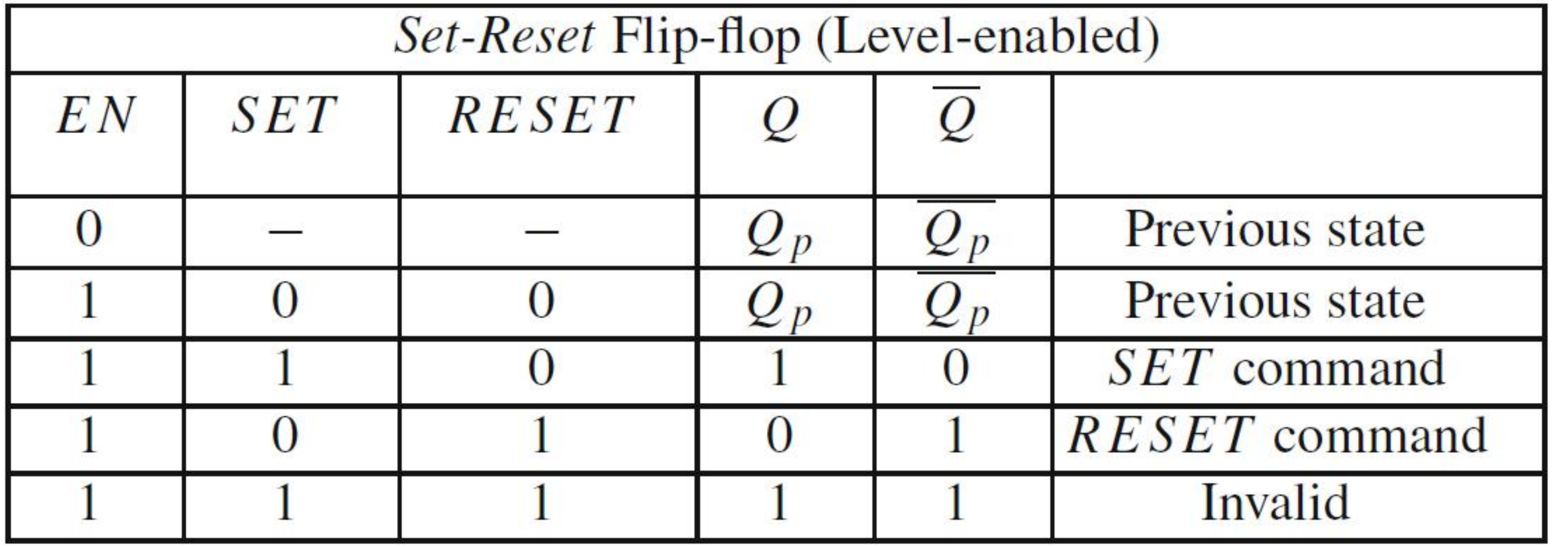

Power Up
Reset Push-button
Reset Generator
$\overline{\text{Reset}}$
...to the $\overline{\text{Clear}}$, or $\overline{\text{Preset}}$, of every flip-flop

Introduction to Sequential Networks

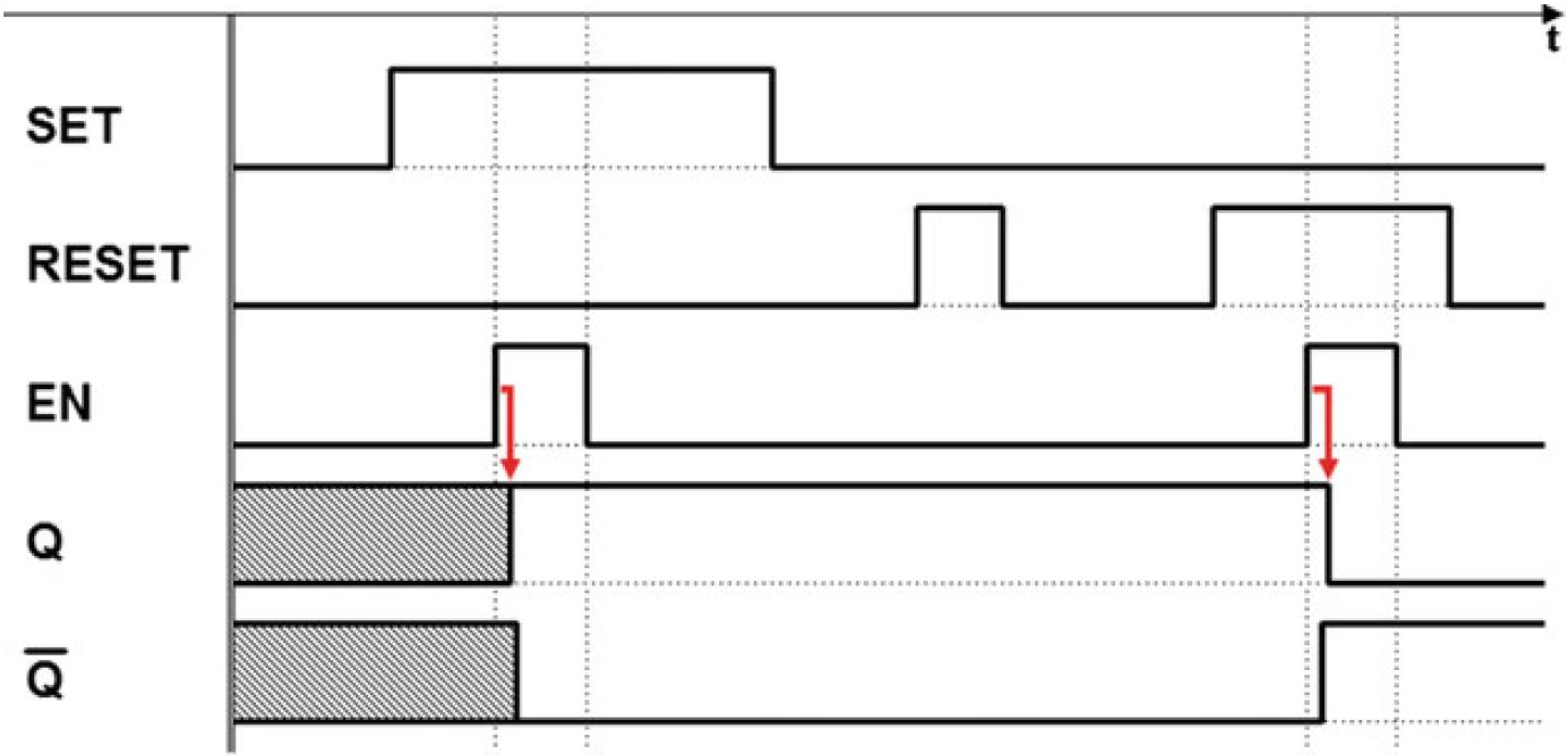
Level-Enabled Flip-Flops

SR-Latch Flip-Flop

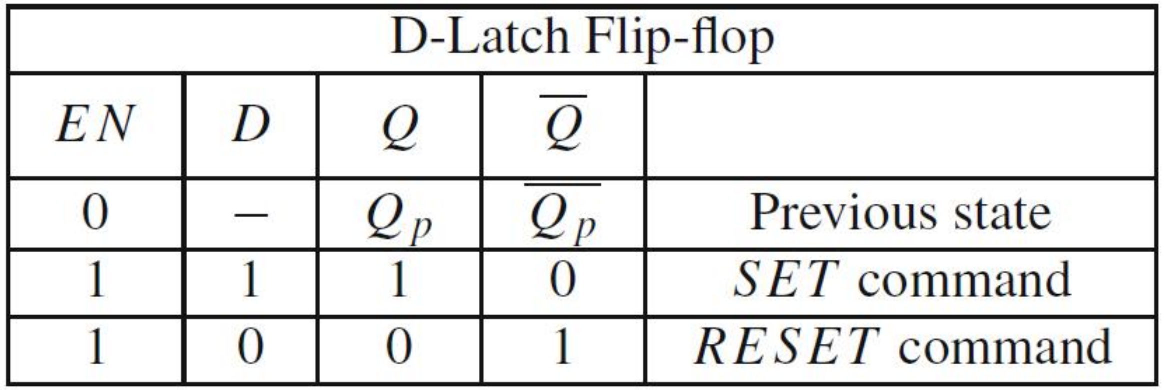




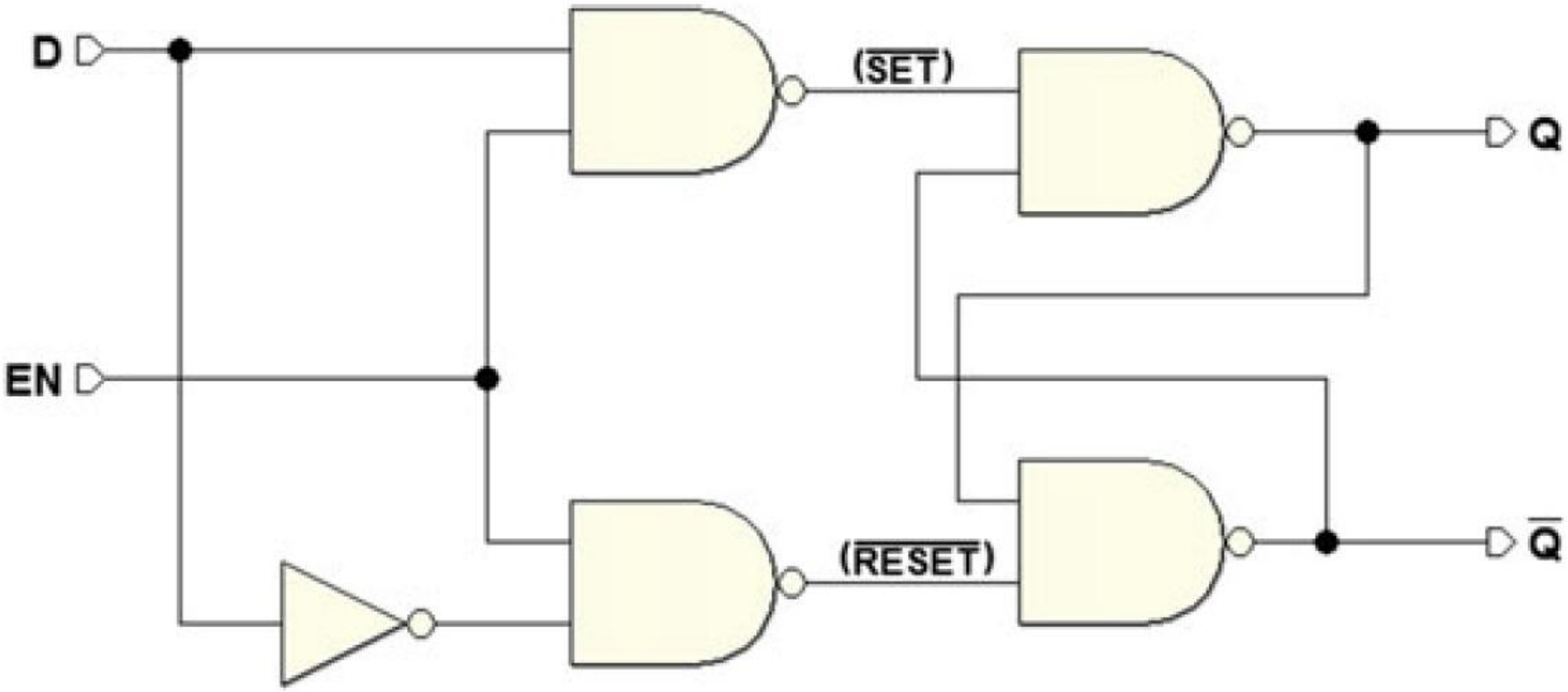
Set-Reset Flip-flop (Level-enabled)					
EN	SET	$RESET$	Q	$\overline{Q}$	
0	–	–	Q_p	$\overline{Q_p}$	Previous state
1	0	0	Q_p	$\overline{Q_p}$	Previous state
1	1	0	1	0	SET command
1	0	1	0	1	$RESET$ command
1	1	1	1	1	Invalid

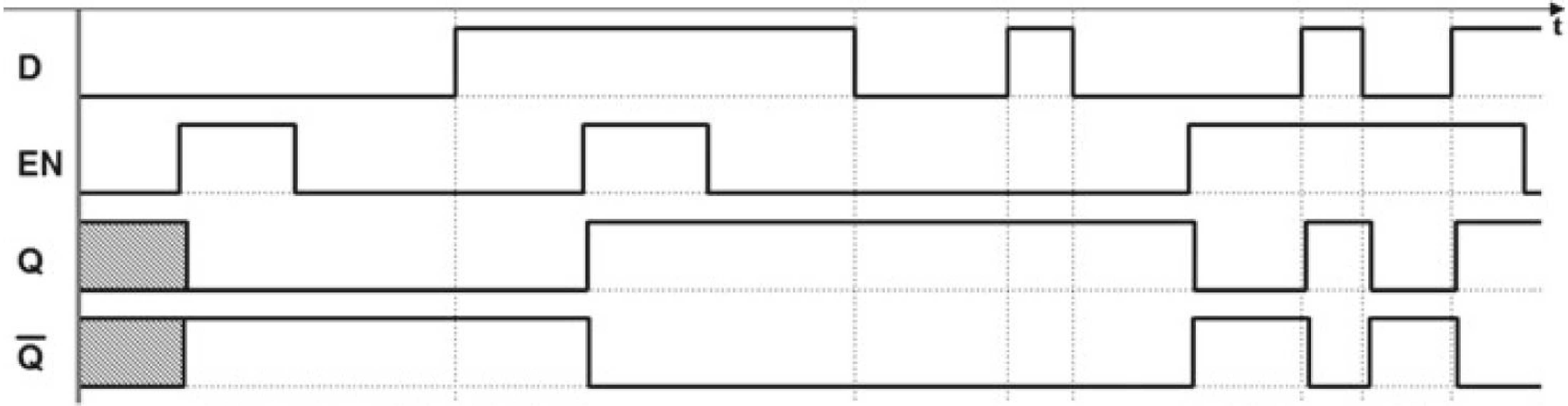

t
SET
RESET
EN
Q
$\overline{Q}$

D-Latch Flip-Flop (D-Latch)

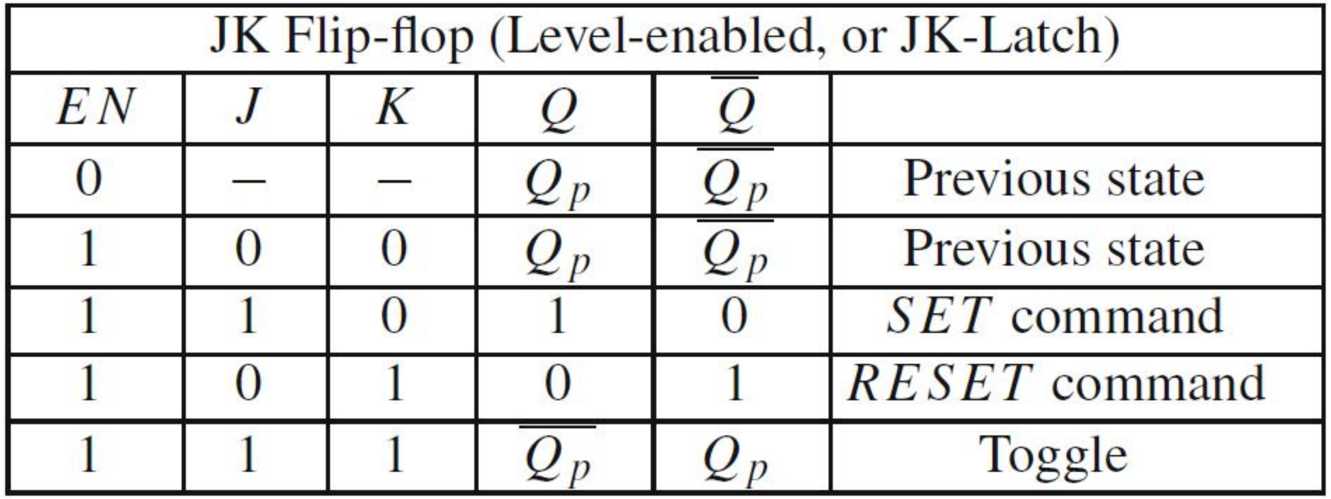


D-Latch Flip-flop				
EN	D	Q	$\overline{Q}$	
0	–	Q_p	$\overline{Q_p}$	Previous state
1	1	1	0	SET command
1	0	0	1	$RESET$ command



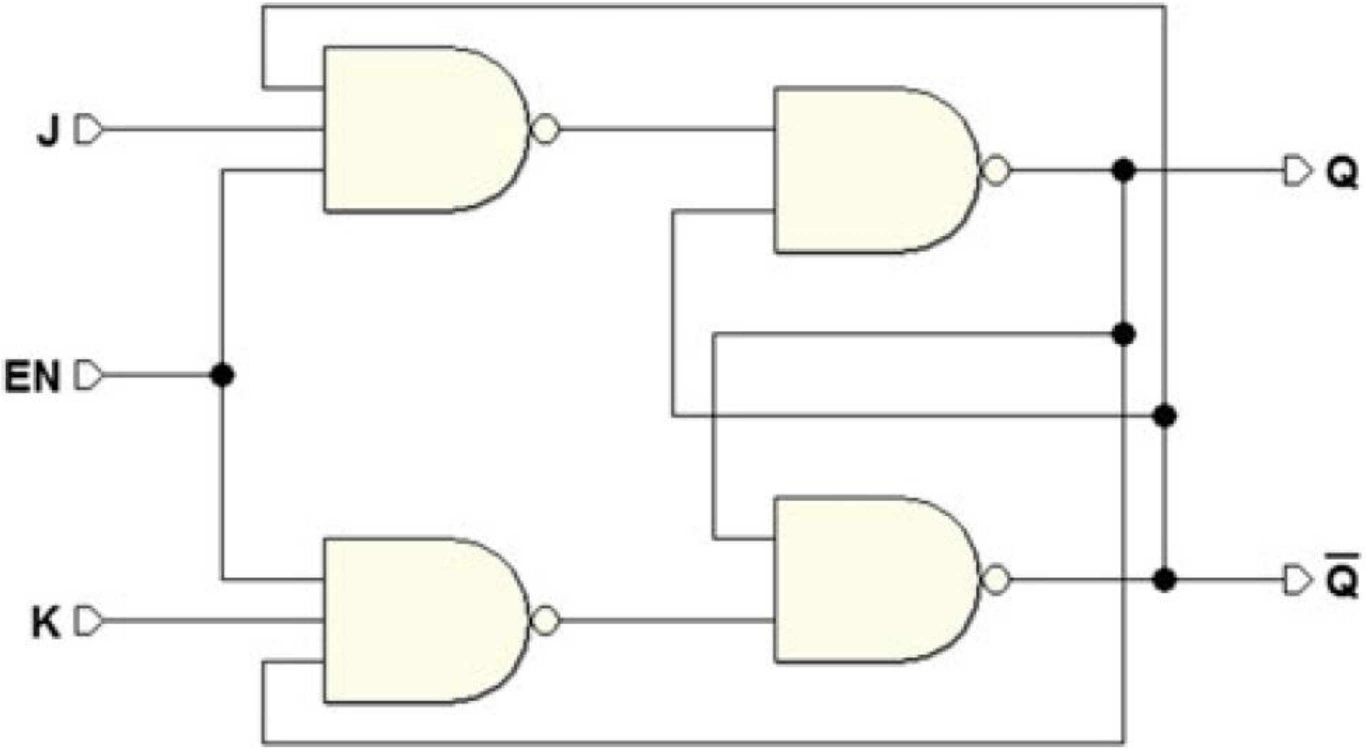

t
D
EN
Q
$\overline{Q}$

JK-Latch Flip-Flop



JK Flip-flop (Level-enabled, or JK-Latch)

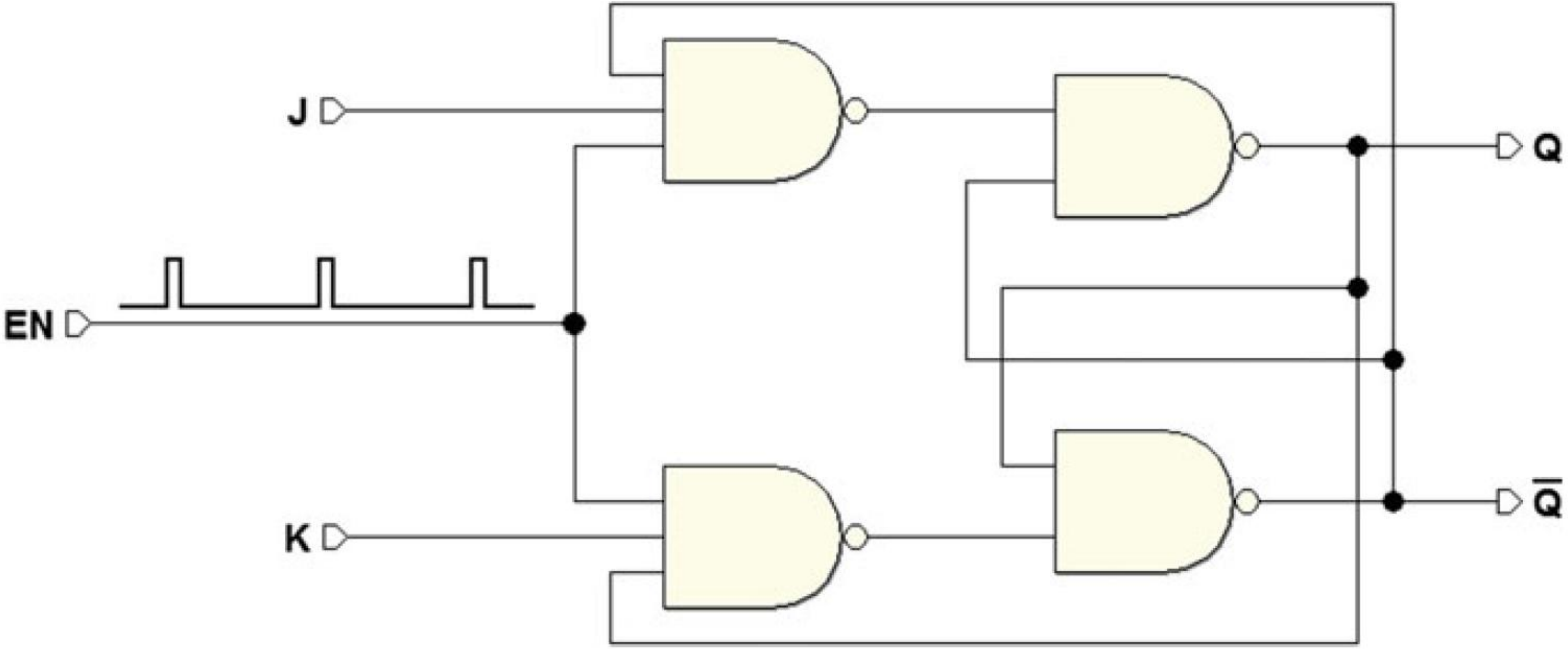
EN	J	K	Q	$\overline{Q}$	
0	–	–	Q_p	$\overline{Q_p}$	Previous state
1	0	0	Q_p	$\overline{Q_p}$	Previous state
1	1	0	1	0	SET command
1	0	1	0	1	$RESET$ command
1	1	1	$\overline{Q_p}$	Q_p	Toggle

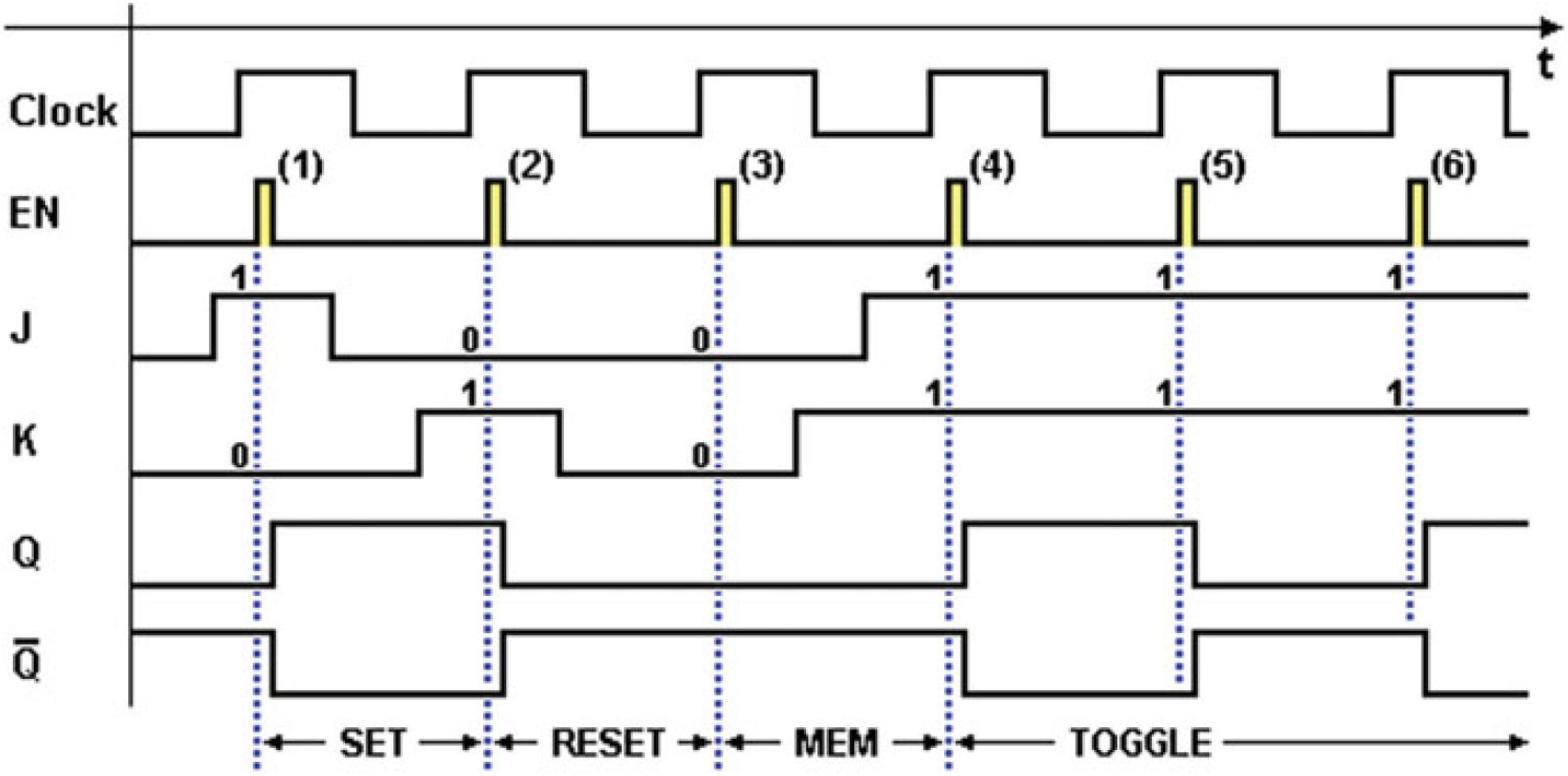


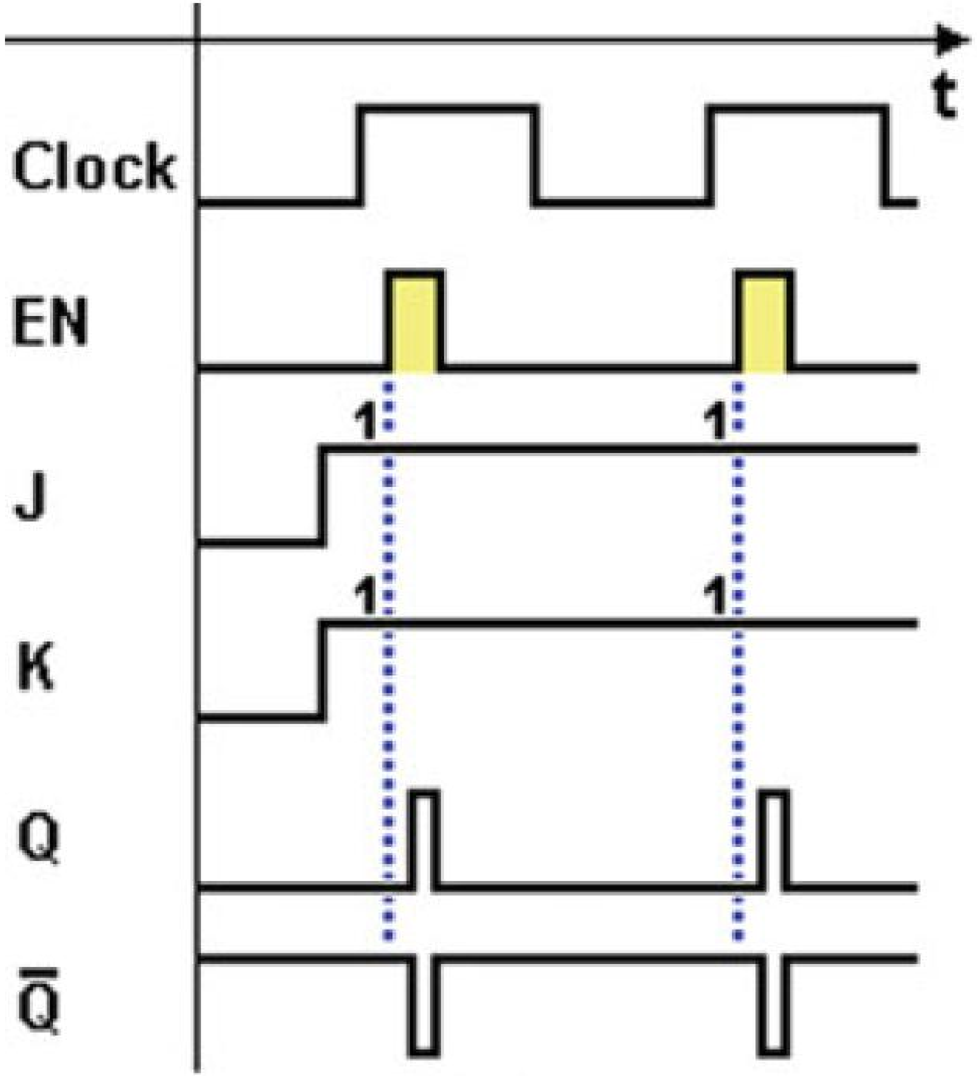
INTRODUCTION TO SEQUENTIAL NETWORKS

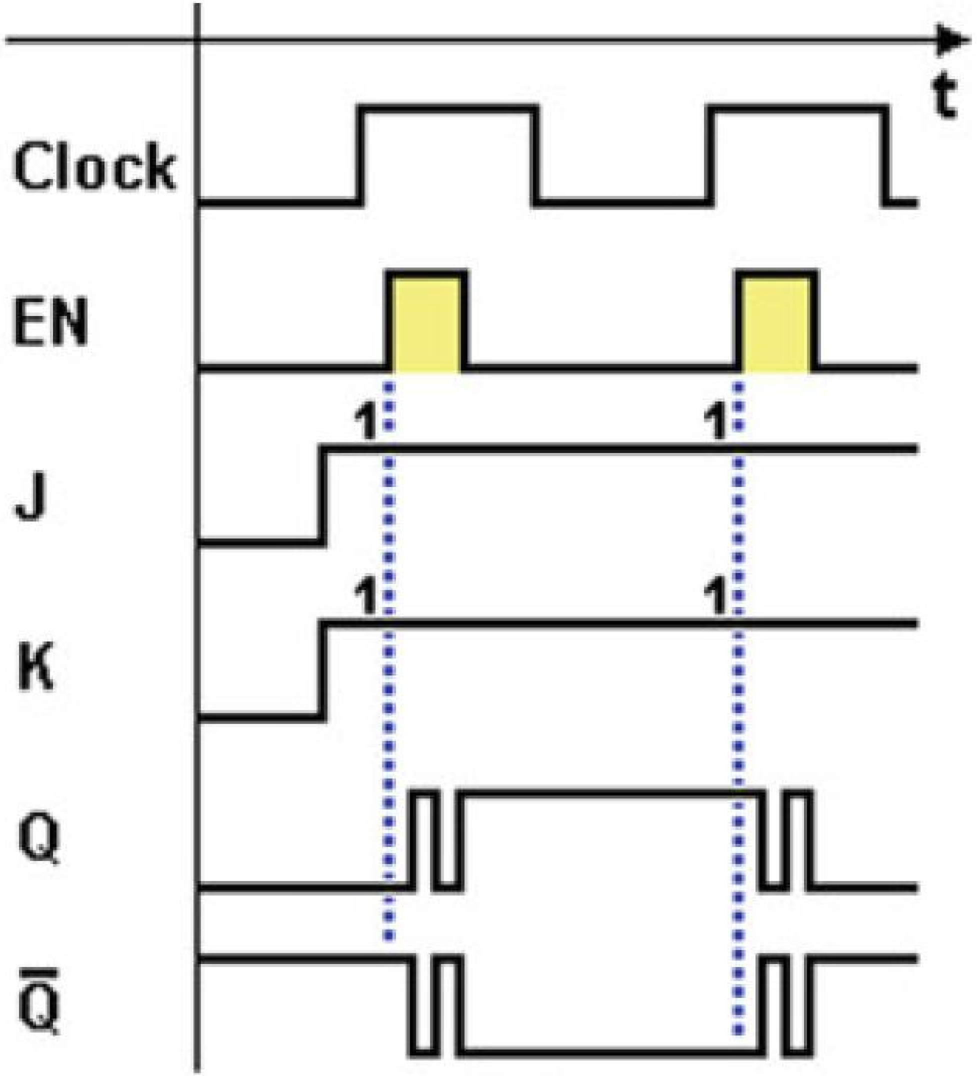
Synchronization of Sequential Networks

The Synchronization Signal

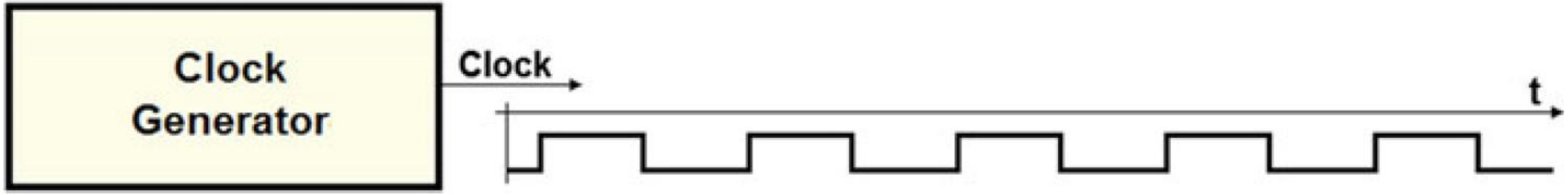


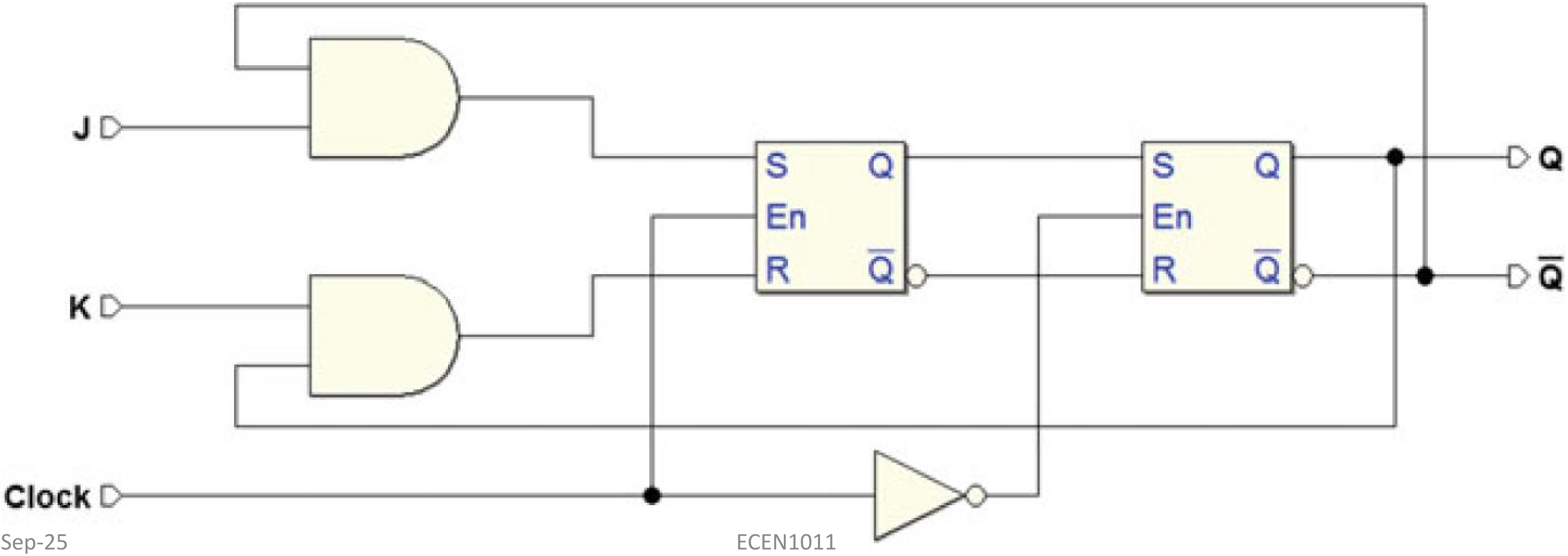

t
Clock
EN
(1)
(2)
(3)
(4)
(5)
(6)
J
1
0
0
1
1
1
K
0
1
0
1
1
1
Q
Q̄
SET
RESET
MEM
TOGGLE

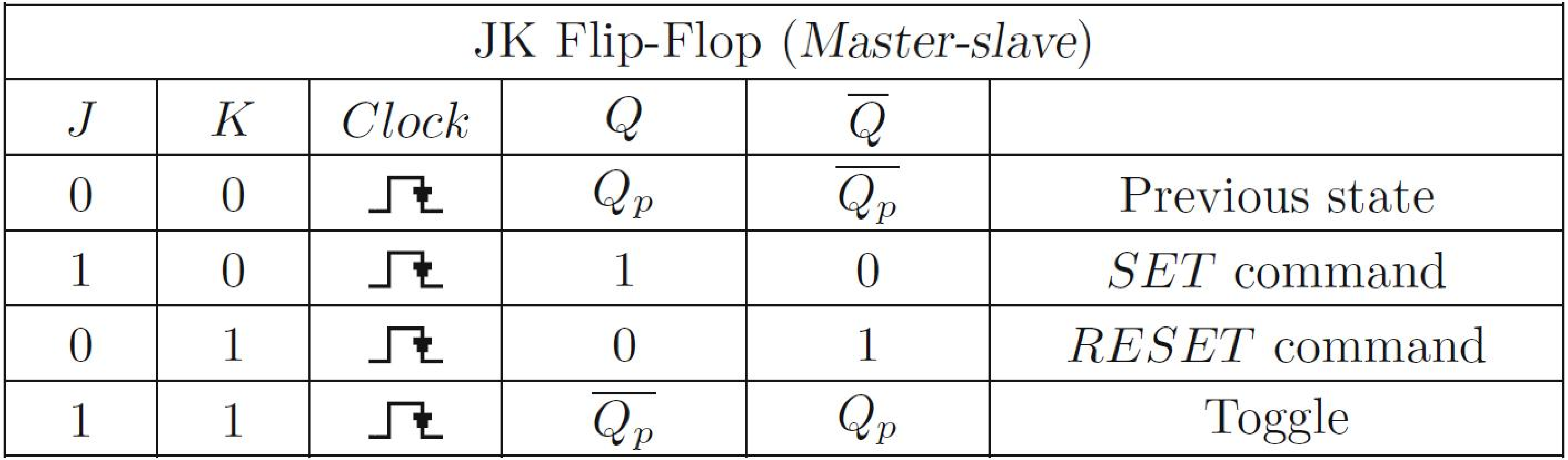

t
Clock
EN
1
J
1
K
Q
Q̄


t
Clock
EN
1
J
1
K
Q
Q̄

The “Clock” and the “Edge-Triggered Command”

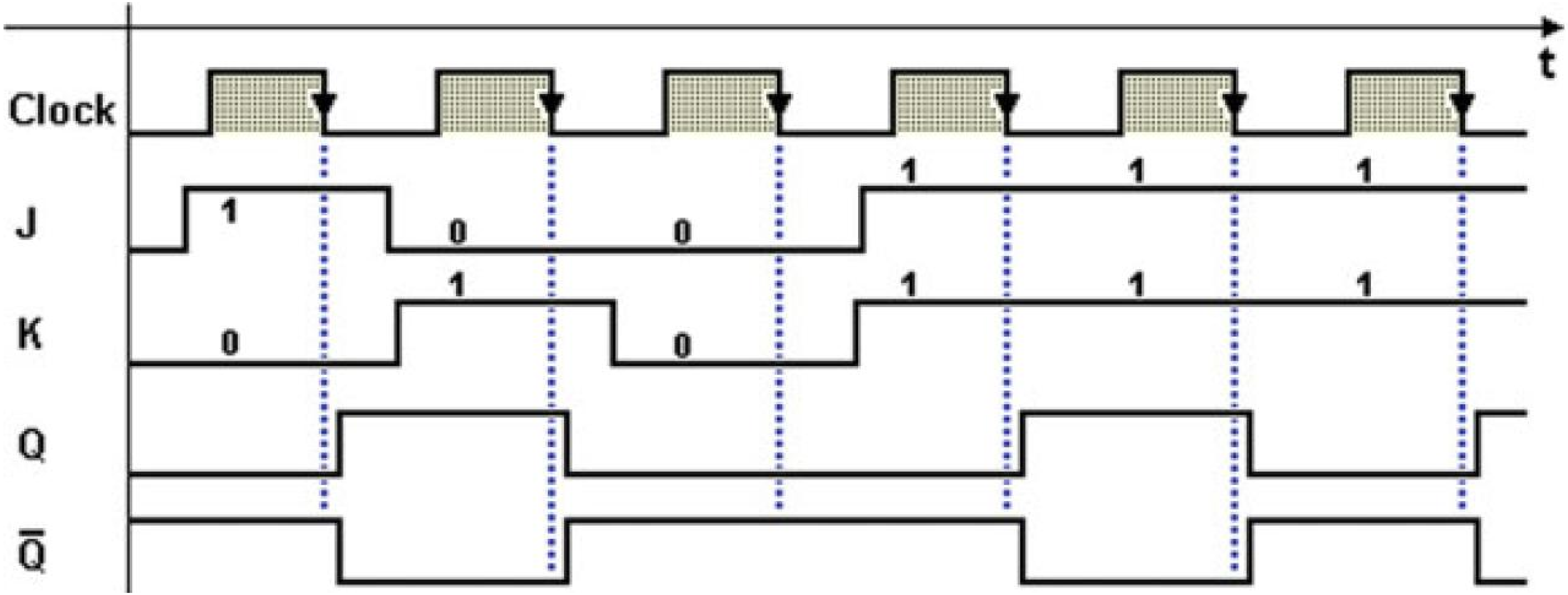




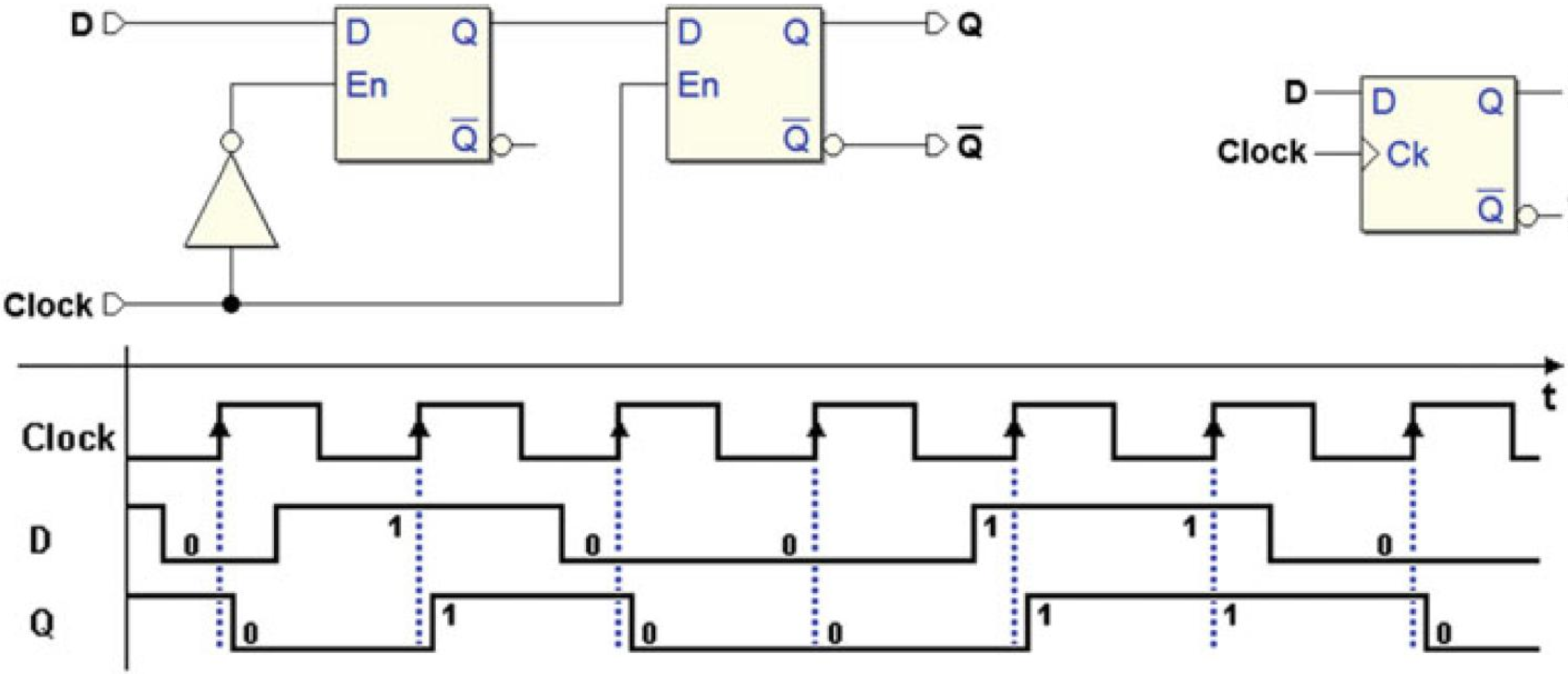


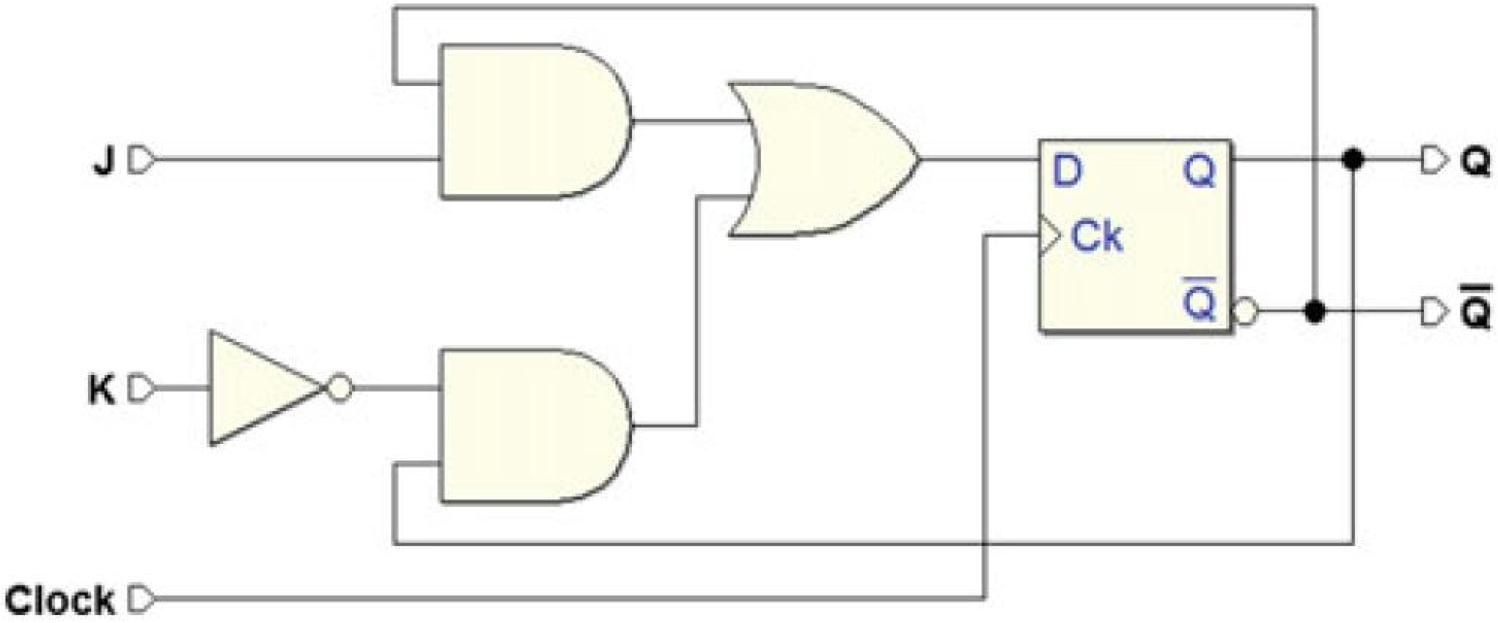
JK Flip-Flop (*Master-slave*)

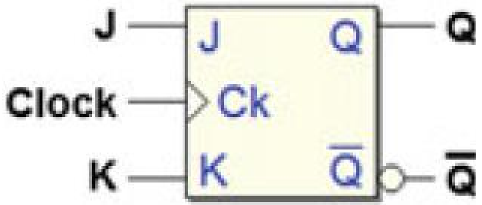
J	K	$Clock$	Q	$\overline{Q}$	
0	0	⎍↓	Q_p	$\overline{Q_p}$	Previous state
1	0	⎍↓	1	0	SET command
0	1	⎍↓	0	1	$RESET$ command
1	1	⎍↓	$\overline{Q_p}$	Q_p	Toggle


t
Clock
J
K
Q
Q̄
1
0
0
1
1
1
1
1
1
1
0
0

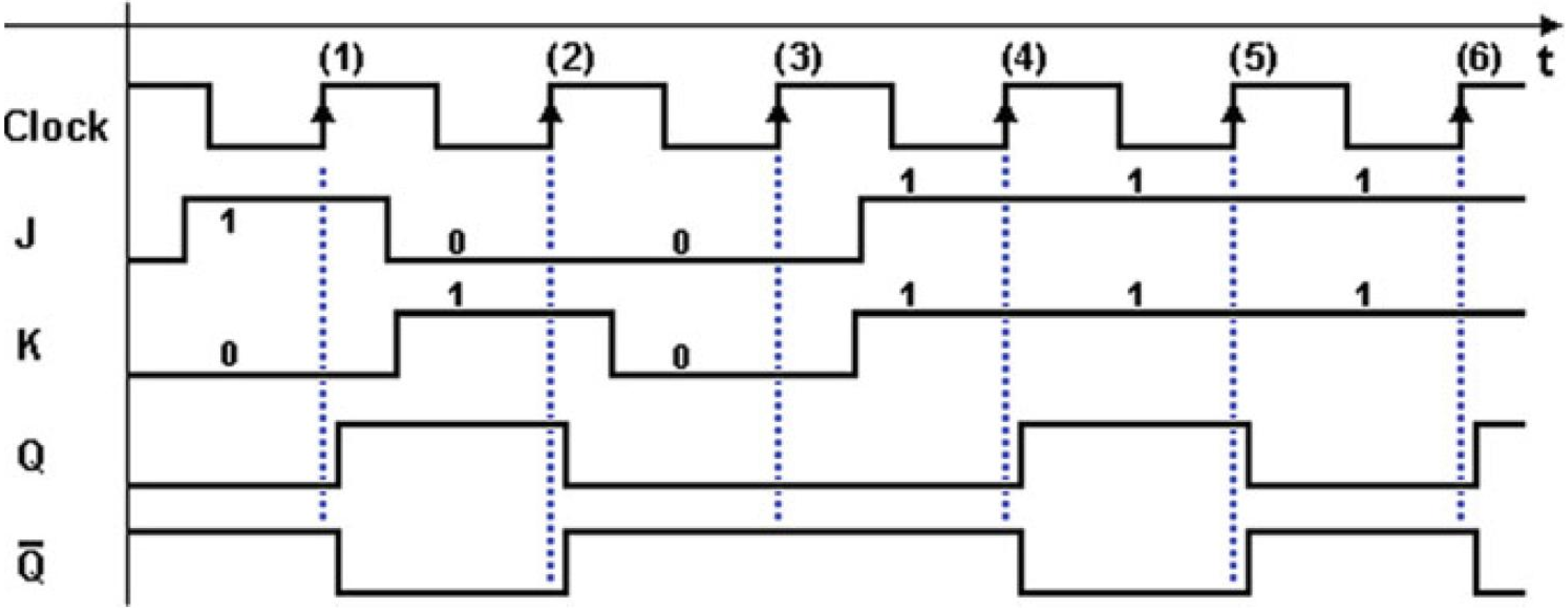
Edge-Triggered Flip-Flops



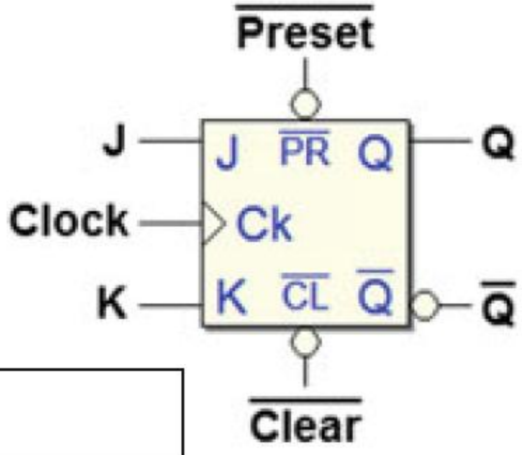




JK-PET Flip-flop					
J	K	$Clock$	Q	$\overline{Q}$	
0	0	↑	Q_p	$\overline{Q_p}$	Previous state
1	0	↑	1	0	SET command
0	1	↑	0	1	$RESET$ command
1	1	↑	$\overline{Q_p}$	Q_p	Toggle

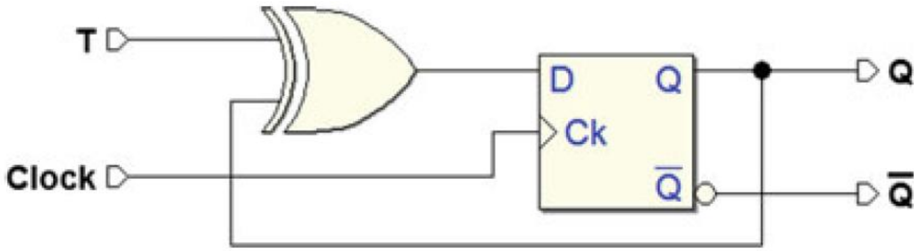

(1)
(2)
(3)
(4)
(5)
(6)
t
Clock
J
1
0
0
1
1
1
K
1
0
0
1
1
1
Q
Q̄

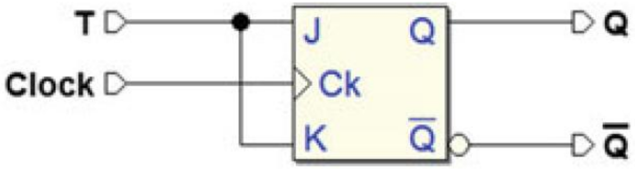
JK-PET Flip-Flop (with $\overline{Clear}$ and $\overline{Preset}$)

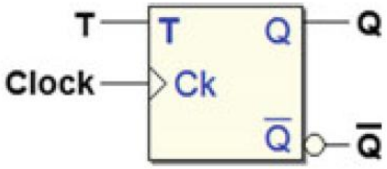


JK-PET Flip-Flop (with $\overline{Clear}$ and $\overline{Preset}$)							
$\overline{Clear}$	$\overline{Preset}$	J	K	$clock$	Q	$\overline{Q}$	
0	1	–	–	–	0	1	Action of Clear
1	0	–	–	–	1	0	Action of Preset
0	0	–	–	–	1	1	(invalid)
1	1	0	0	↑	Q_p	$\overline{Q_p}$	Previous value
1	1	1	0	↑	1	0	SET command
1	1	0	1	↑	0	1	$RESET$ command
1	1	1	1	↑	$\overline{Q_p}$	Q_p	Outputs reversed

T-PET Flip-Flop

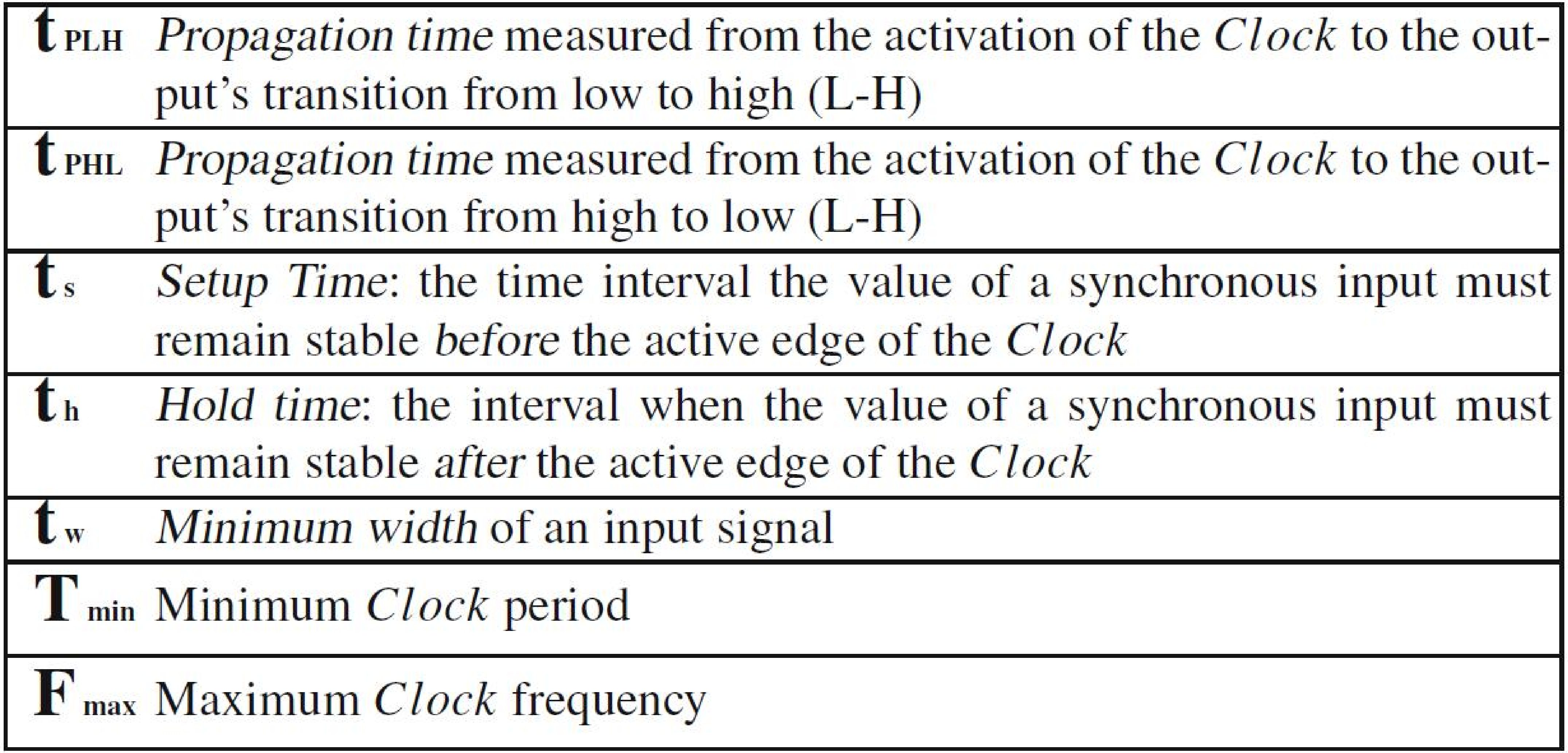




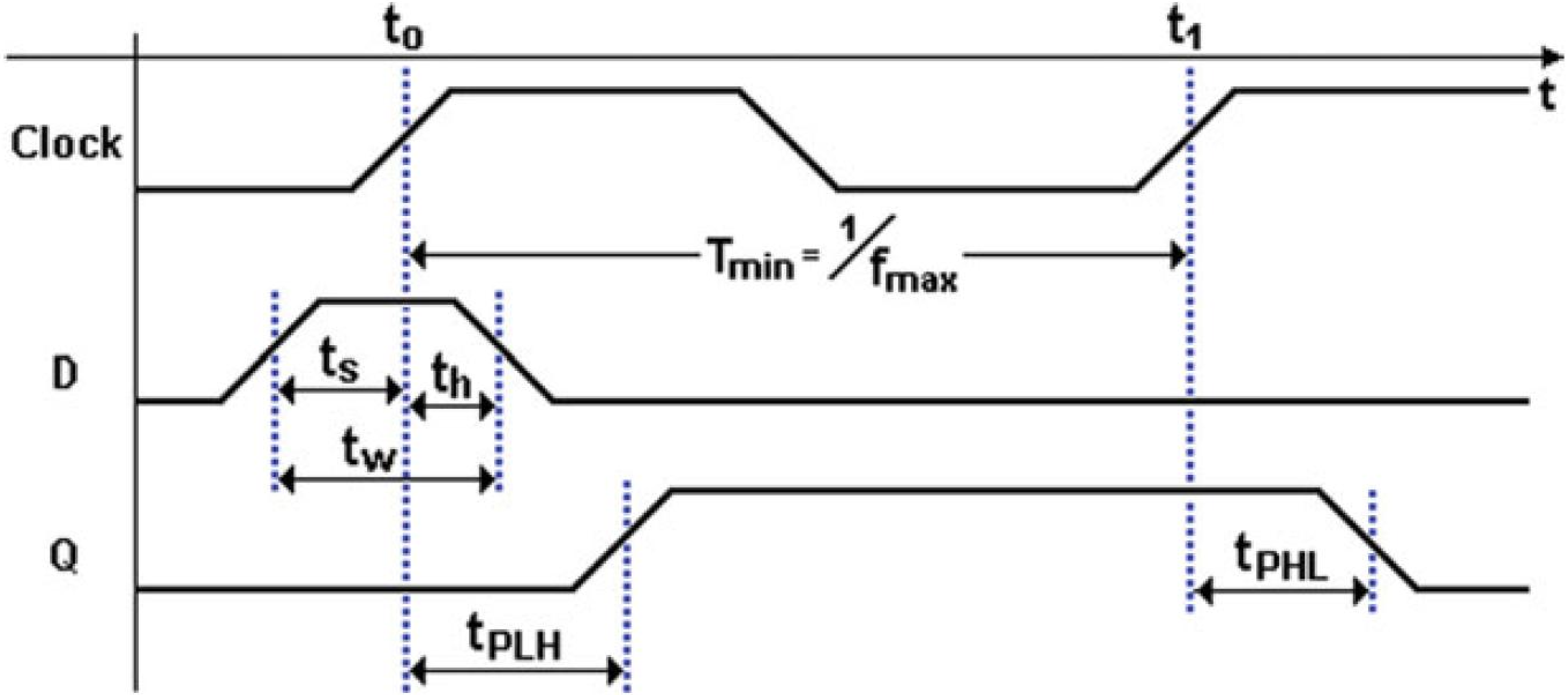


The T-PET Flip-Flop				
T	$Clock$	Q	$\overline{Q}$	
0	↑	Q_p	$\overline{Q_p}$	Previous state
1	↑	$\overline{Q_p}$	Q_p	Toggle

Timing Parameters of Flip-Flops



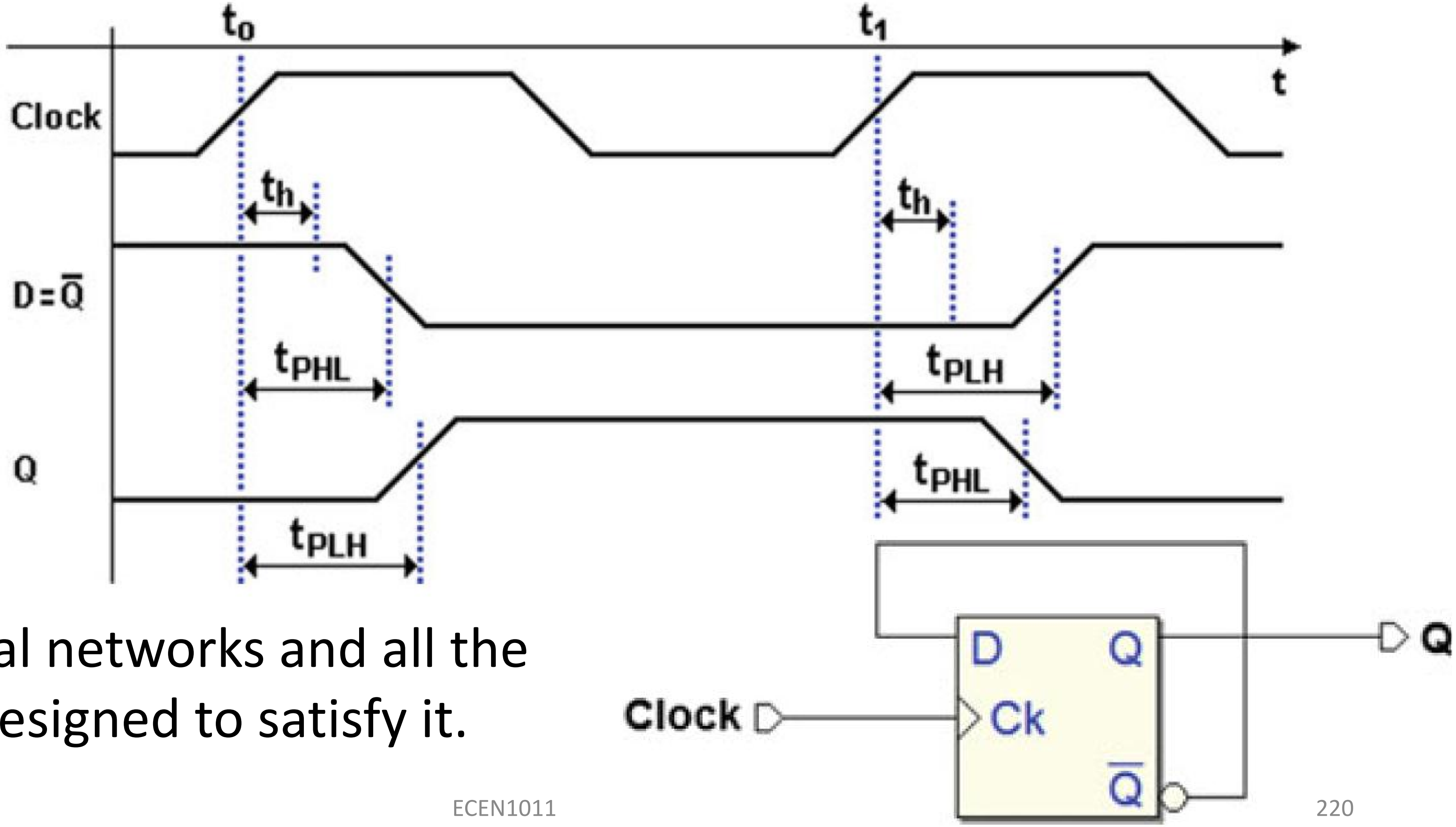
t_{PLH}	*Propagation time* measured from the activation of the *Clock* to the output's transition from low to high (L-H)
t_{PHL}	*Propagation time* measured from the activation of the *Clock* to the output's transition from high to low (L-H)
t_{s}	*Setup Time*: the time interval the value of a synchronous input must remain stable *before* the active edge of the *Clock*
t_{h}	*Hold time*: the interval when the value of a synchronous input must remain stable *after* the active edge of the *Clock*
t_{w}	*Minimum width* of an input signal
T_{min}	Minimum *Clock* period
F_{max}	Maximum *Clock* frequency



t_0
t_1
t
Clock
$T_{min} = 1/f_{max}$
D
t_s
t_h
t_w
Q
t_{PHL}
t_{PLH}

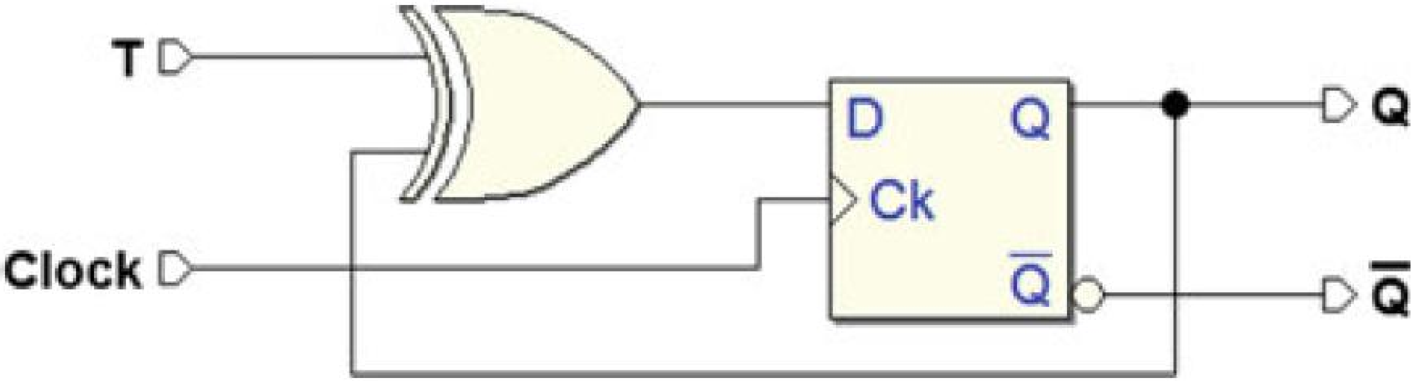
Relationship Between Propagation and Hold Times

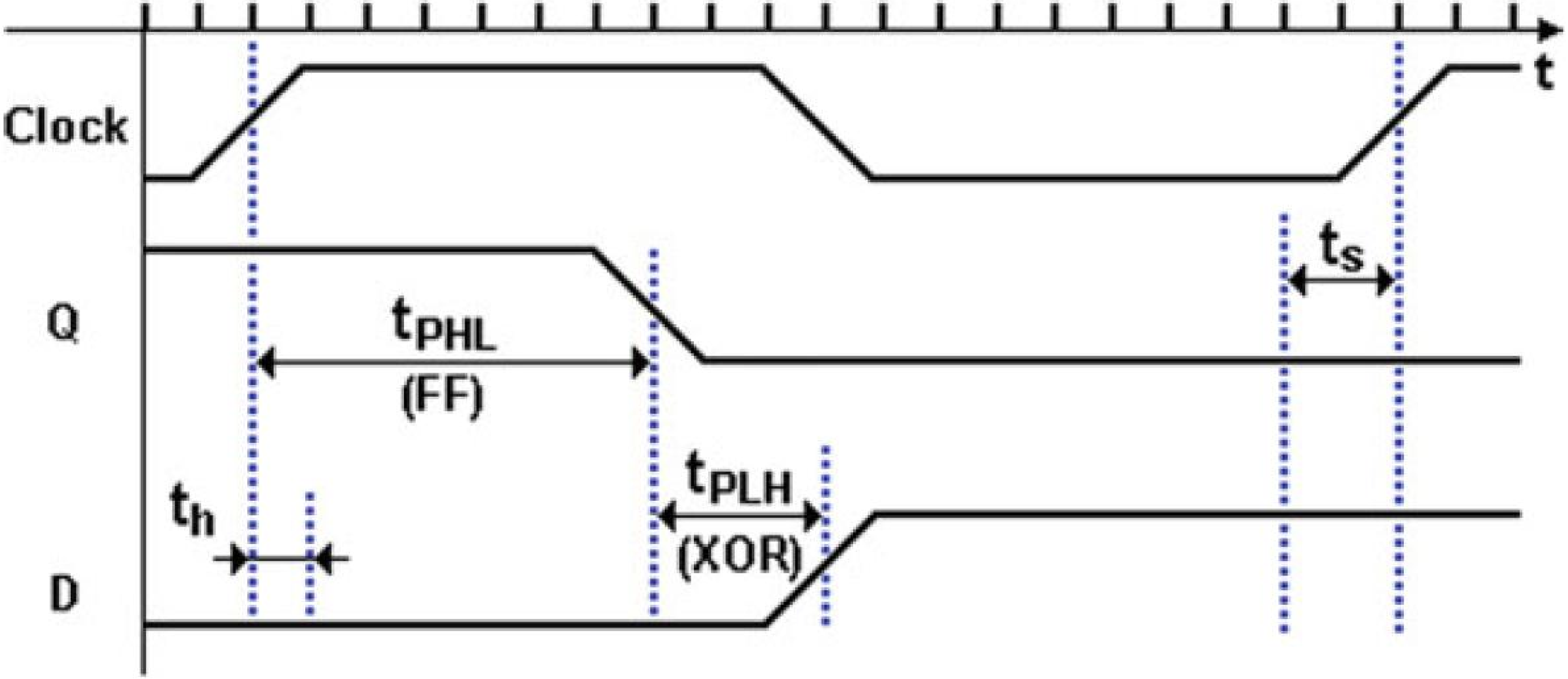
t_h must be shorter than both the two propagation times (**t_{PHL}** and **t_{PLH}**). This condition is at the base of all sequential networks and all the flip-flops are designed to satisfy it.



Maximum Clock Frequency of a Network with Flip-Flops

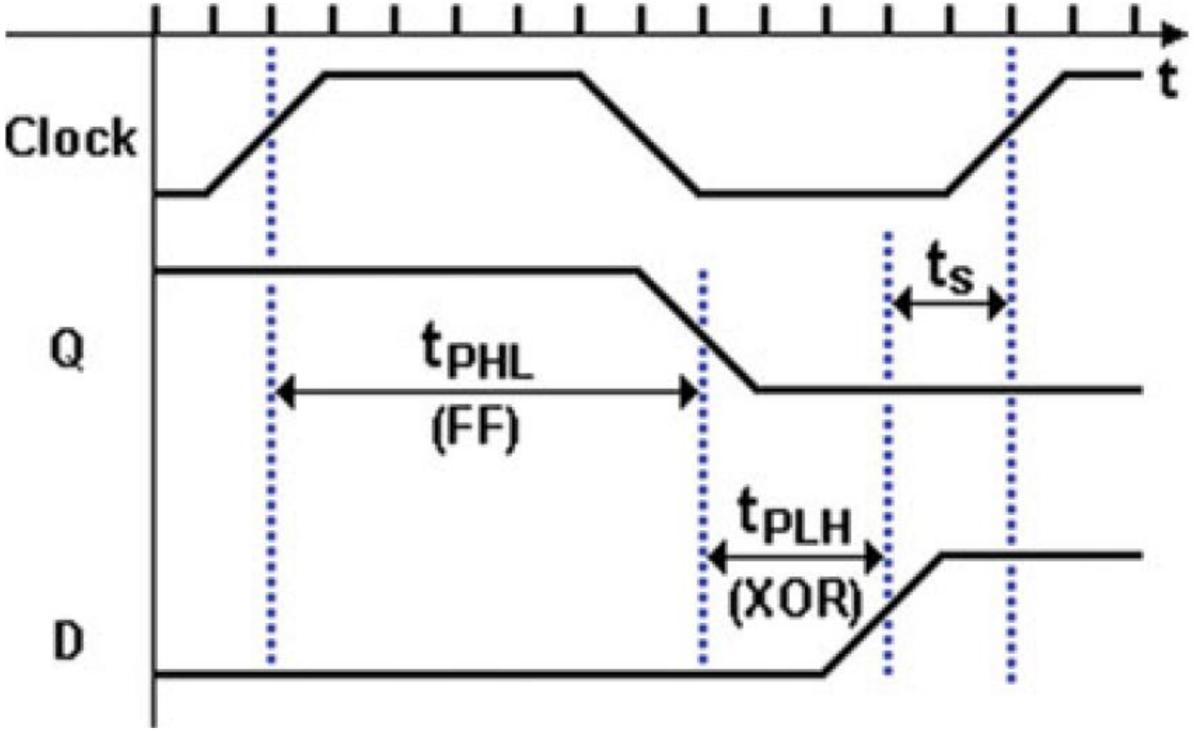
- t_s = 2 nS,
- t_h = 1 nS
- t_{PHL}(FF) = 7 nS
- t_{PLH}(XOR) = 3 nS
- T_{min} = ?
- f_{max} = ?

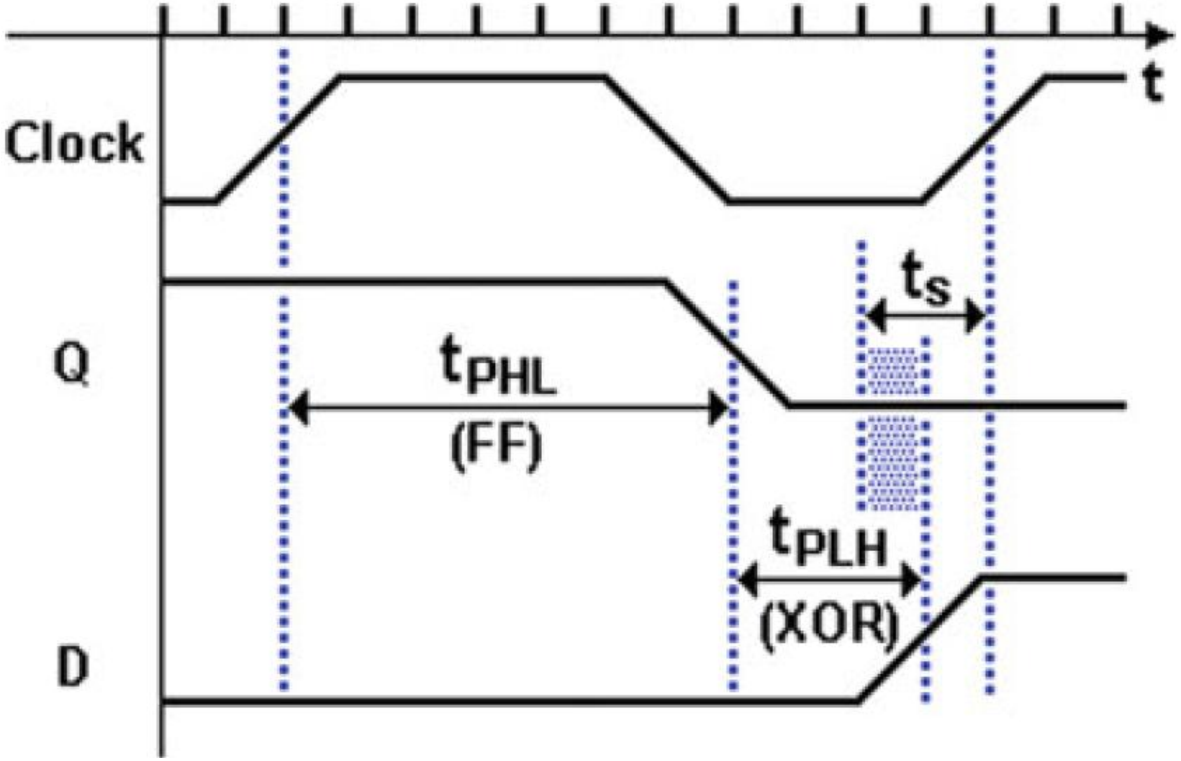




T_{min} = t_{PHL}(FF) +t_{PLH}(XOR)+t_s=7+3+2=12ns

f_{max} = 1/12=0.083GHz=83MHz





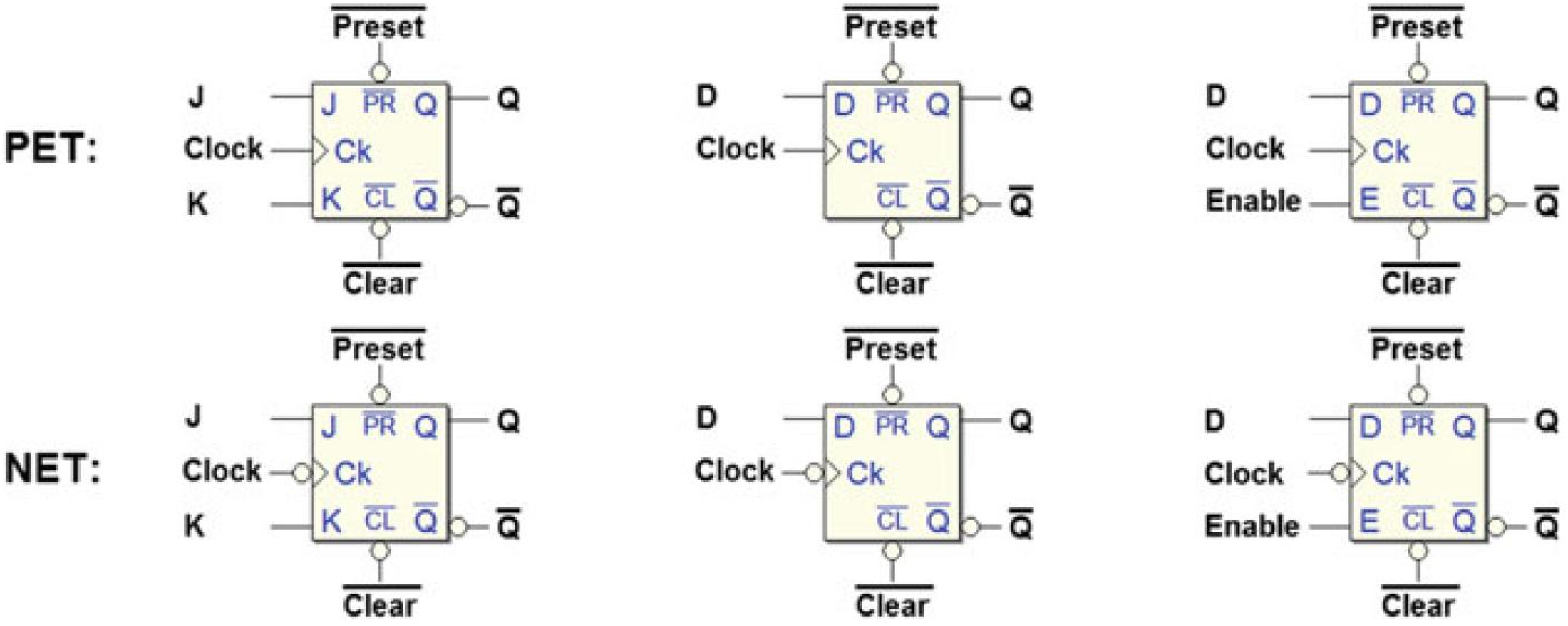
$T_{\mathbf{clock}}$ = 12ns = $T_{\mathbf{min}}$

$f_{\mathbf{clock}}$ = 83MHz = $f_{\mathbf{max}}$

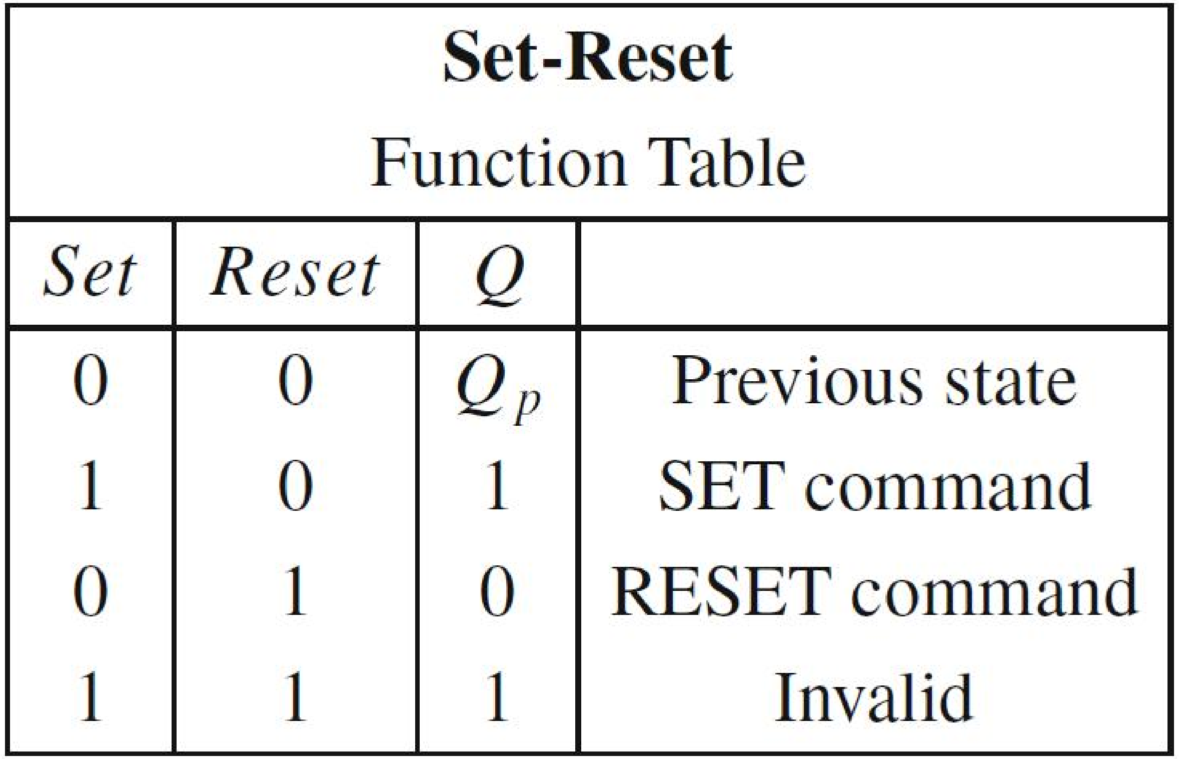
$T_{\mathbf{clock}}$ = 11ns < $T_{\mathbf{min}}$

$f_{\mathbf{clock}}$ = 91MHz > $f_{\mathbf{max}}$

Flip-flop symbols



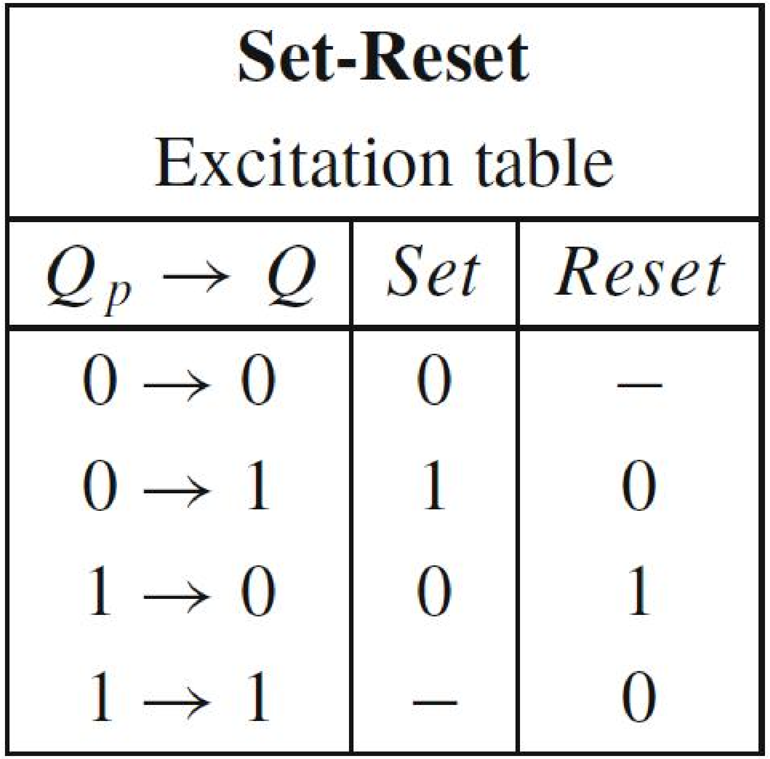
Excitation Tables (S-R)



Set-Reset

Function Table

Set	*Reset*	*Q*	
0	0	Q_p	Previous state
1	0	1	SET command
0	1	0	RESET command
1	1	1	Invalid

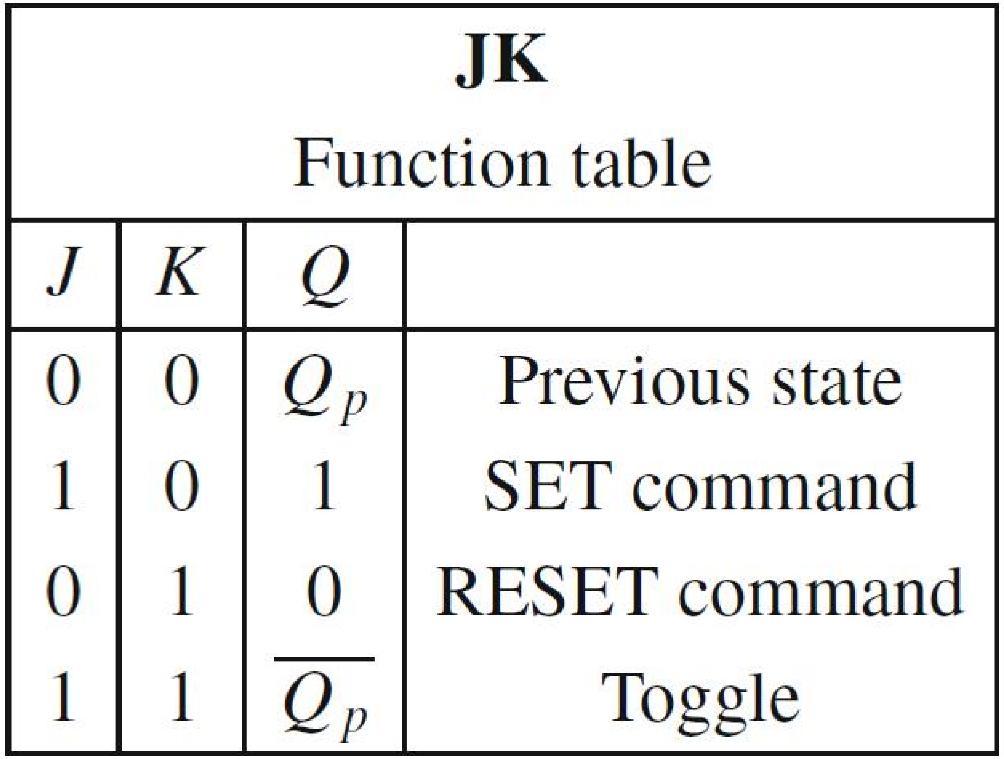


Set-Reset

Excitation table

$Q_p \to Q$	*Set*	*Reset*
$0 \to 0$	0	–
$0 \to 1$	1	0
$1 \to 0$	0	1
$1 \to 1$	–	0

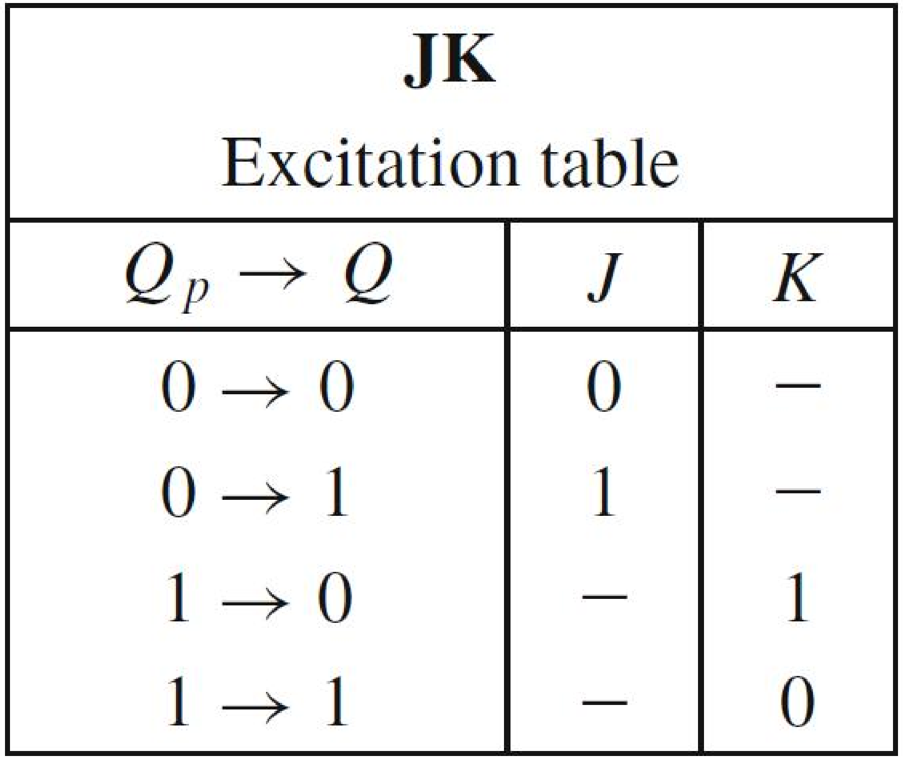
Excitation Tables (JK)



JK

Function table

J	K	Q	
0	0	Q_p	Previous state
1	0	1	SET command
0	1	0	RESET command
1	1	$\overline{Q_p}$	Toggle

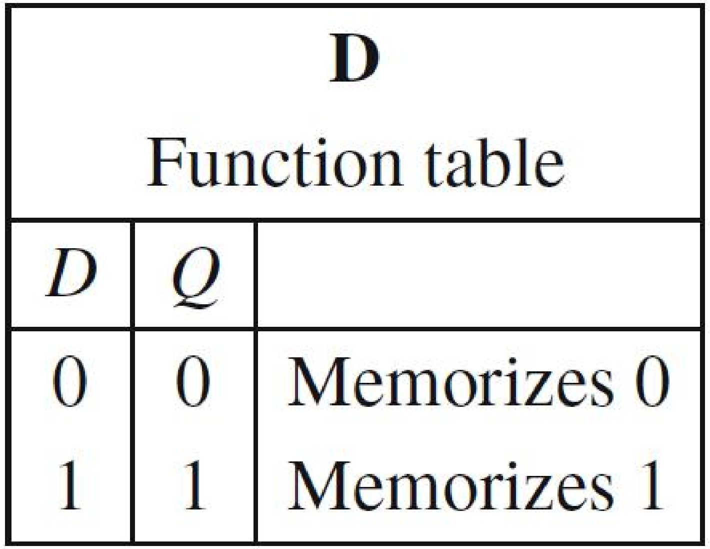


JK

Excitation table

$Q_p \rightarrow Q$	J	K
$0 \rightarrow 0$	0	–
$0 \rightarrow 1$	1	–
$1 \rightarrow 0$	–	1
$1 \rightarrow 1$	–	0

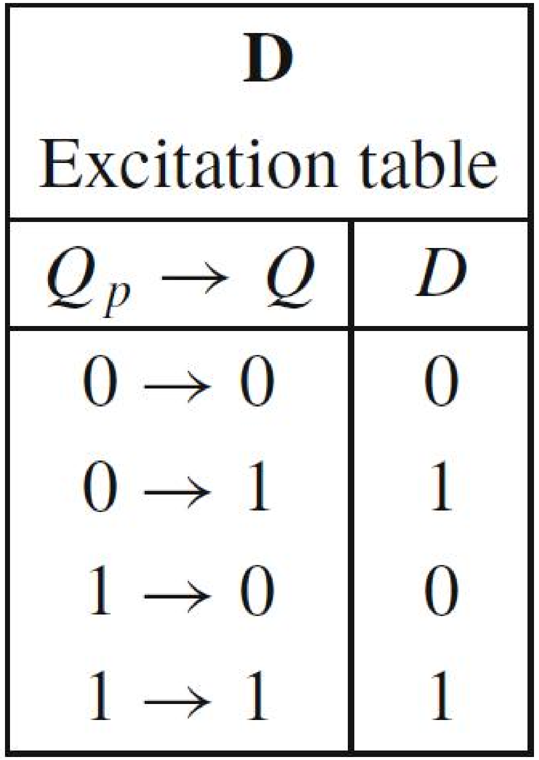
Excitation Tables (D)



D

Function table

D	Q	
0	0	Memorizes 0
1	1	Memorizes 1



D

Excitation table

$Q_p \to Q$	D
$0 \to 0$	0
$0 \to 1$	1
$1 \to 0$	0
$1 \to 1$	1

Synchronous Positive edge JK Flip-Flop with Reset and Clock enable in VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity JK_FF_VHDL is
    port(   J,K: in  std_logic;
            Reset, Clock_enable: in std_logic;
            Clock: in std_logic;
            Output: out std_logic);
end JK_FF_VHDL;

architecture Behavioral of JK_FF_VHDL is
    signal temp: std_logic;
begin
    process (Clock)
    begin
        if rising_edge(Clock) then
            if Reset='1' then
                temp <= '0';
            elsif Clock_enable ='1' then
                if (J='0' and K='0') then
                    temp <= temp;
                elsif (J='0' and K='1') then
                    temp <= '0';
                elsif (J='1' and K='0') then
                    temp <= '1';
                elsif (J='1' and K='1') then
                    temp <= not (temp);
                end if;
            end if;
        end if;
    end process;
    Output <= temp;
end Behavioral;
```

Assignment 4

- P. 179
    - 2, 4, 5