ShanghaiTech University

EE 115B: Digital Circuits

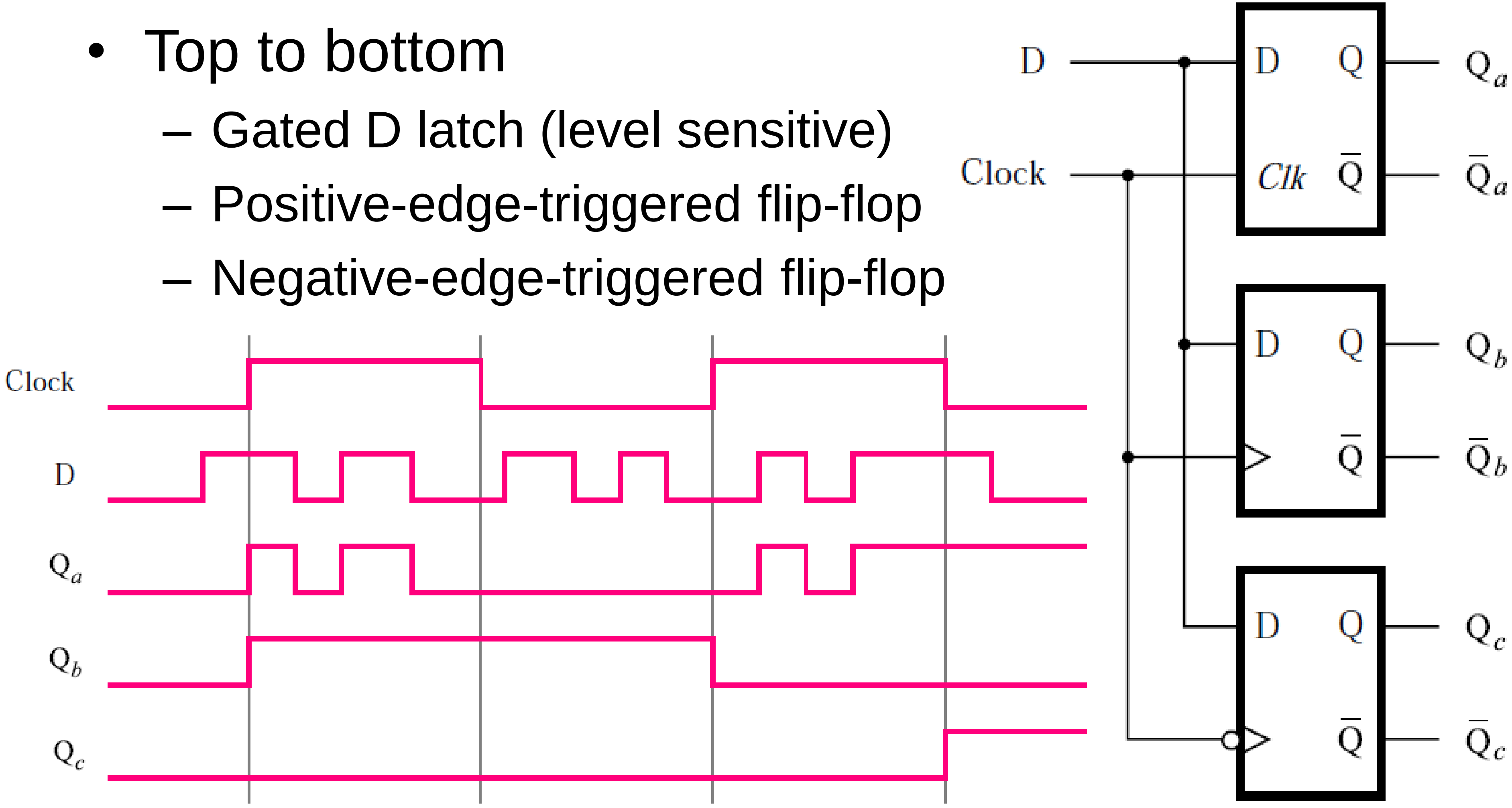
Fall 2022

Lecture 17

Hengzhao Yang
December 1, 2022

D: latch vs. flip-flop

- Top to bottom
  - Gated D latch (level sensitive)
  - Positive-edge-triggered flip-flop
  - Negative-edge-triggered flip-flop

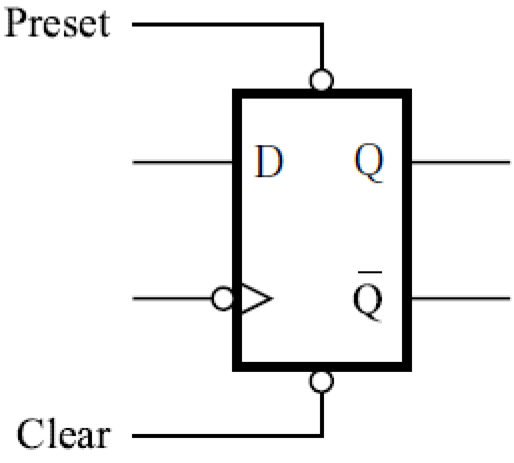


D flip-flops with CLEAR and PRESET

- Control signals: CLEAR and PRESET
  - Use CLEAR to set initial condition Q=0
  - Use PRESET to set initial condition Q=1
- Implementations
  - Based on master-slave D flip-flop
  - Based on edge-triggered D flip-flop

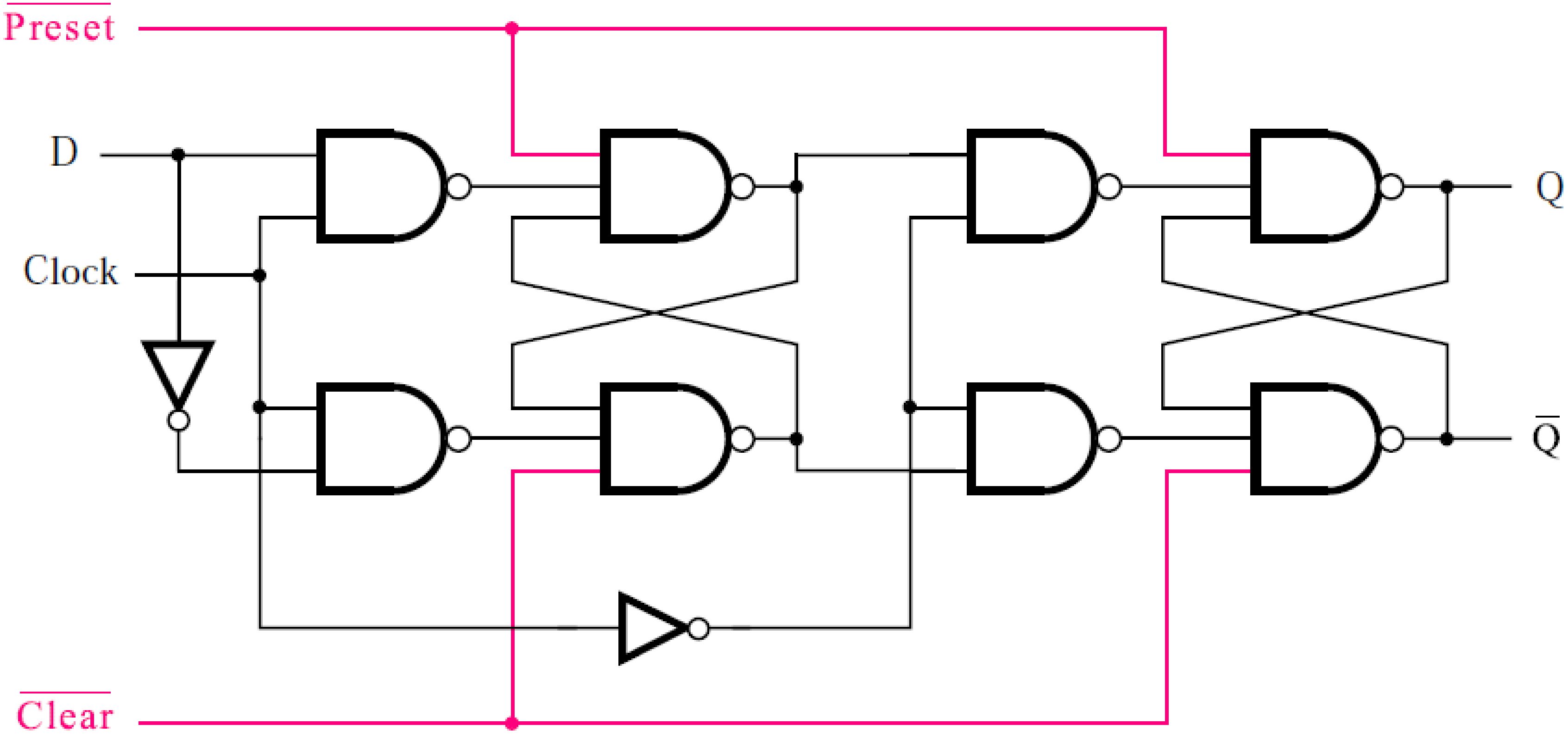
Master-slave D flip-flop with CLEAR and PRESET

- CLEAR and PRESET are active low
  - CLEAR=0: Q=0
  - PRESET=0: Q=1
  - CLEAR=1 or PRESET=1: no effect
- Circuit symbol (negative edge)



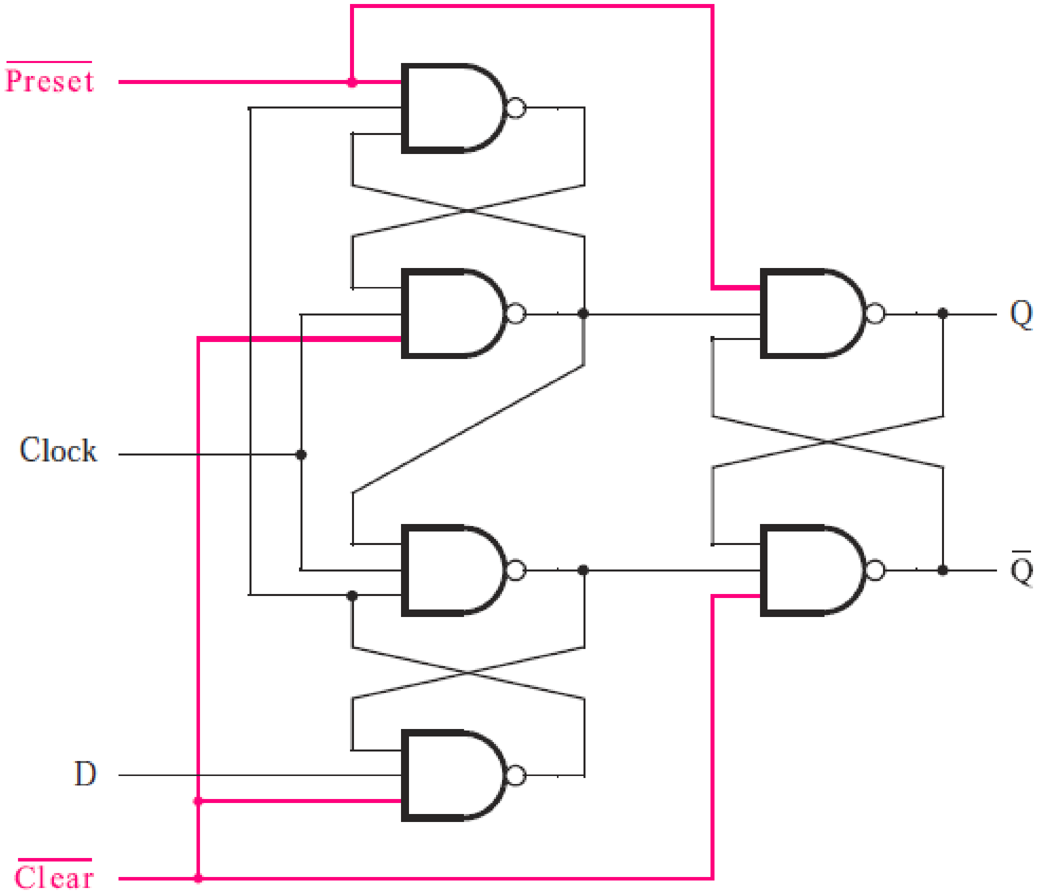
Master-slave D flip-flop with CLEAR and PRESET

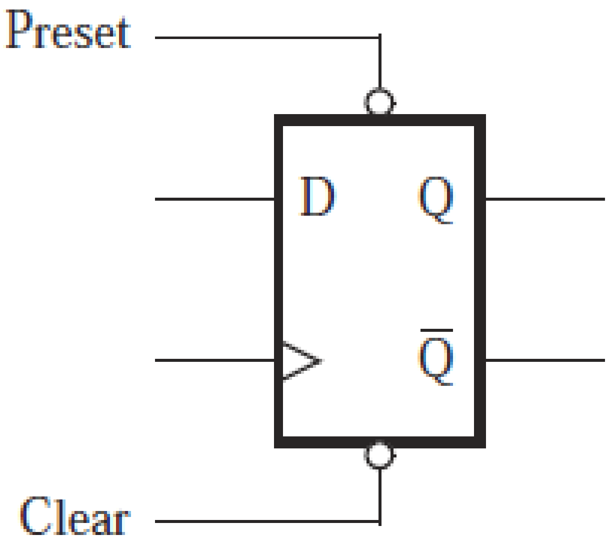
- Circuit: modified master-slave D flip-flop



Edge-triggered D flip-flop with CLEAR and PRESET

- Circuit and circuit symbol (positive edge)



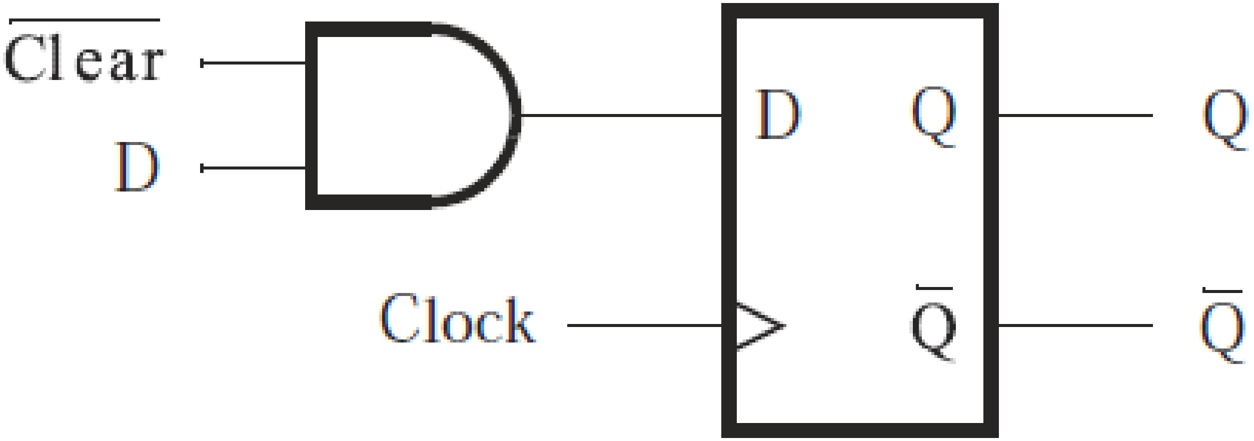


Asynchronous vs. synchronous CLEAR

- Asynchronous CLEAR
  - If CLEAR=0, flip-flop will be cleared to Q=0 immediately without regard to clock
- Synchronous CLEAR
  - If CLEAR=0, flip-flop will be cleared to Q=0 at the next clock edge (positive or negative)

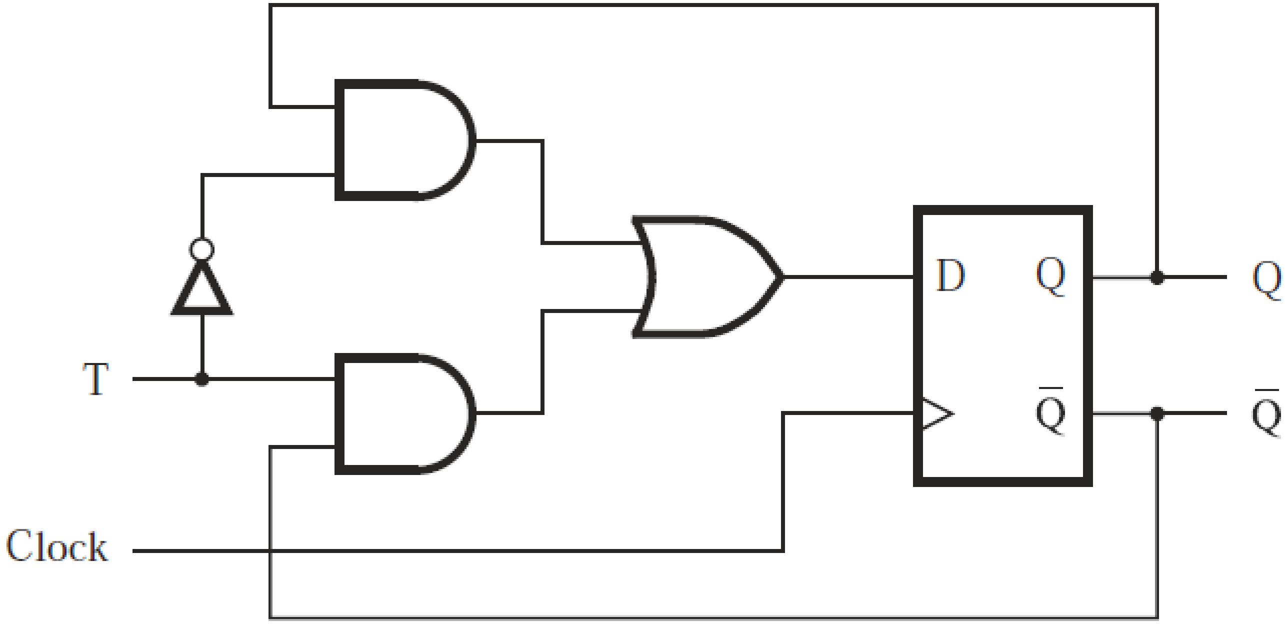
Synchronous CLEAR

- Circuit



T flip-flop

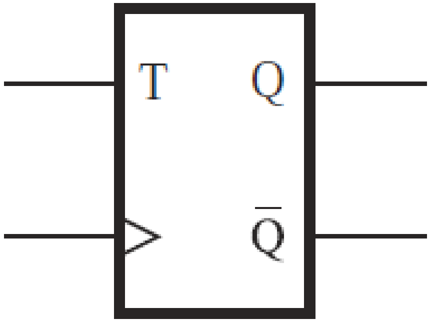
- T flip-flop
  - T=0: state does not change
  - T=1: state is reversed
- Circuit



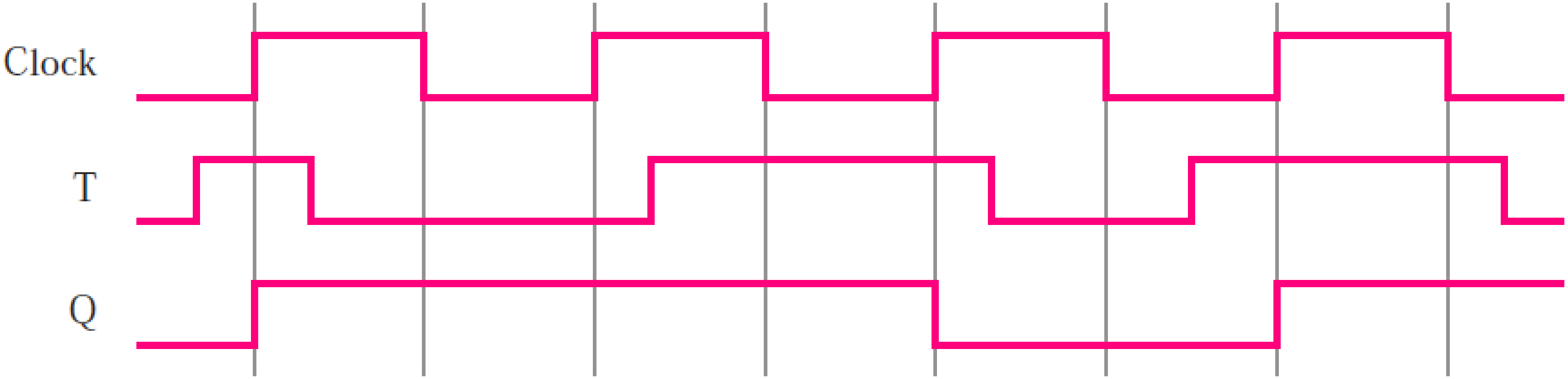
T flip-flop

- Characteristic table and circuit symbol

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

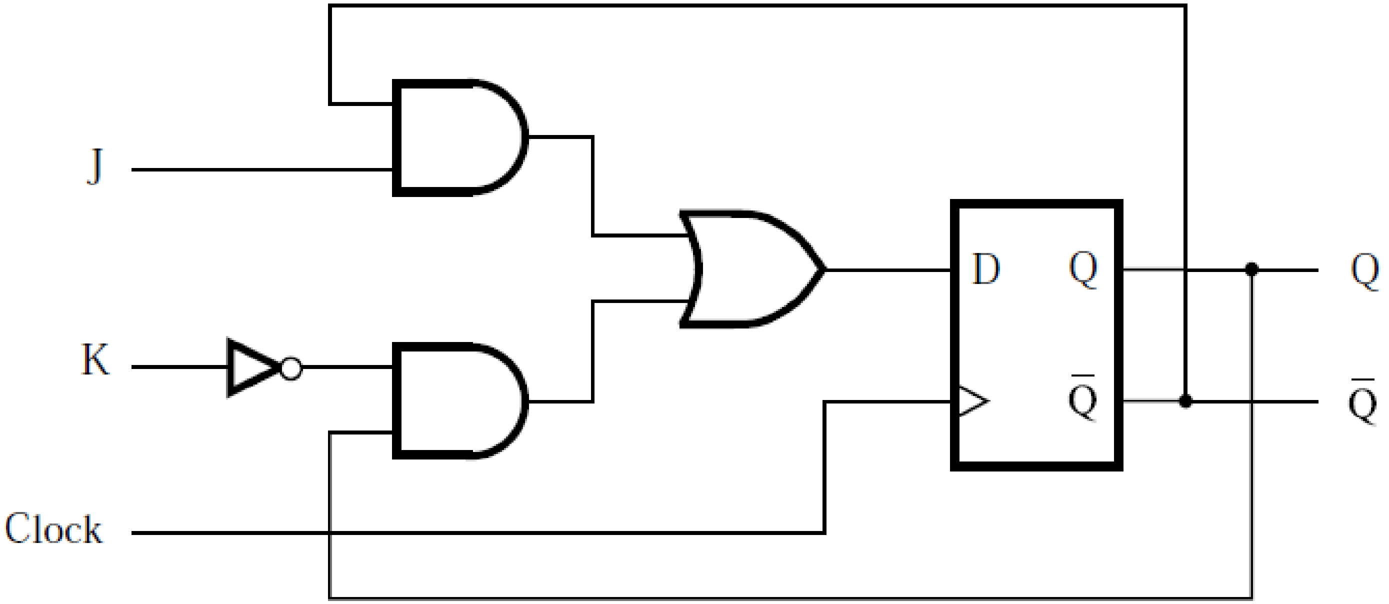


- Timing diagram

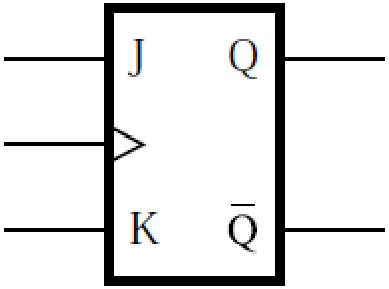


JK flip-flop

- JK flip-flop
  - Combines behaviors of basic SR latch and T flip-flop
  - J: set, K: reset
- Circuit, characteristic table, and circuit symbol

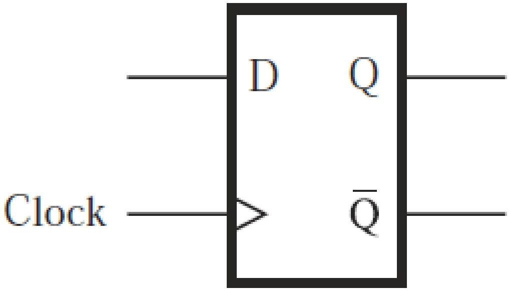


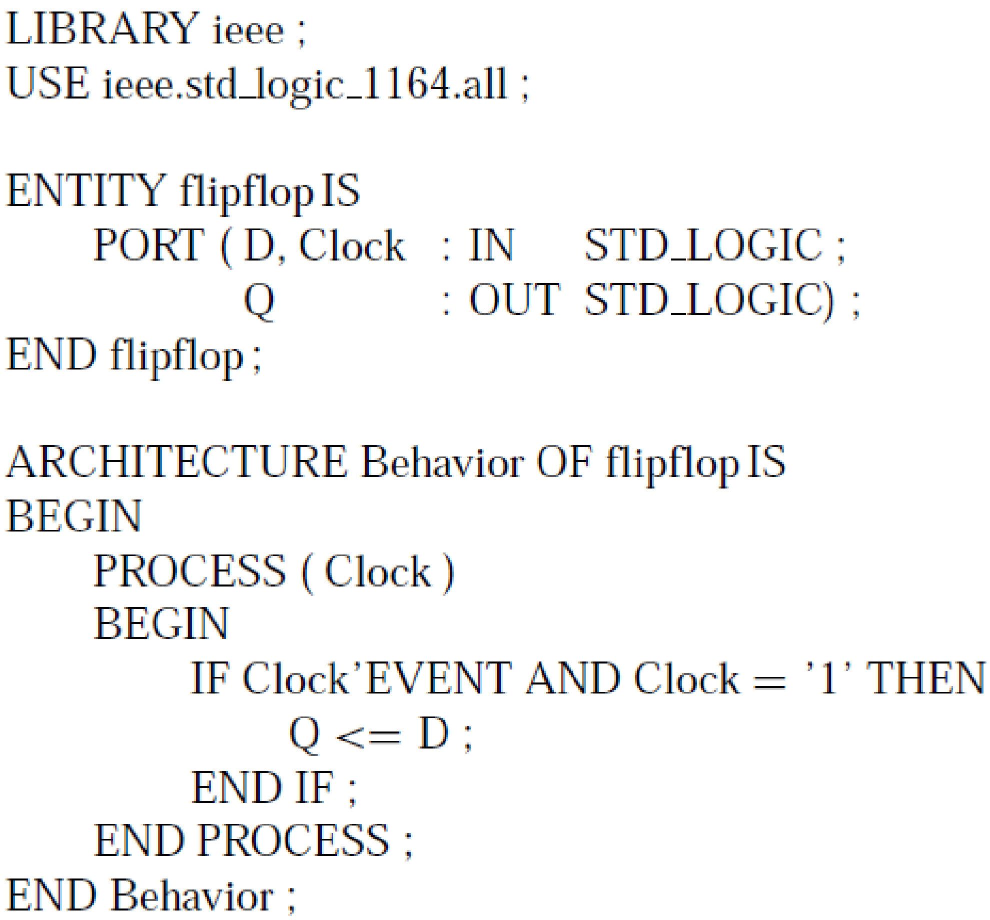
J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q}(t)$



Code for D flip-flop (positive edge)

- Code
  - Sensitivity list: Clock
  - ’EVENT: signal attribute
  - Clock’EVENT: any change in the Clock signal
  - Clock’EVENT AND Clock=‘1’: clock signal value has changed and the value is now 1 (positive edge)



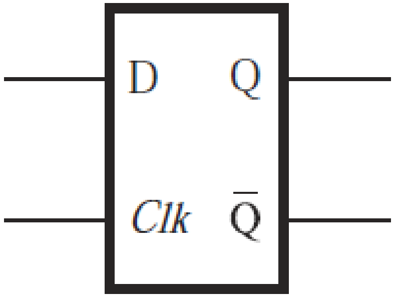


```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY flipflop IS
    PORT ( D, Clock  : IN    STD_LOGIC ;
              Q      : OUT  STD_LOGIC) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS ( Clock )
    BEGIN
        IF Clock'EVENT AND Clock = '1' THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Comparison: VHDL code for gated D latch



Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

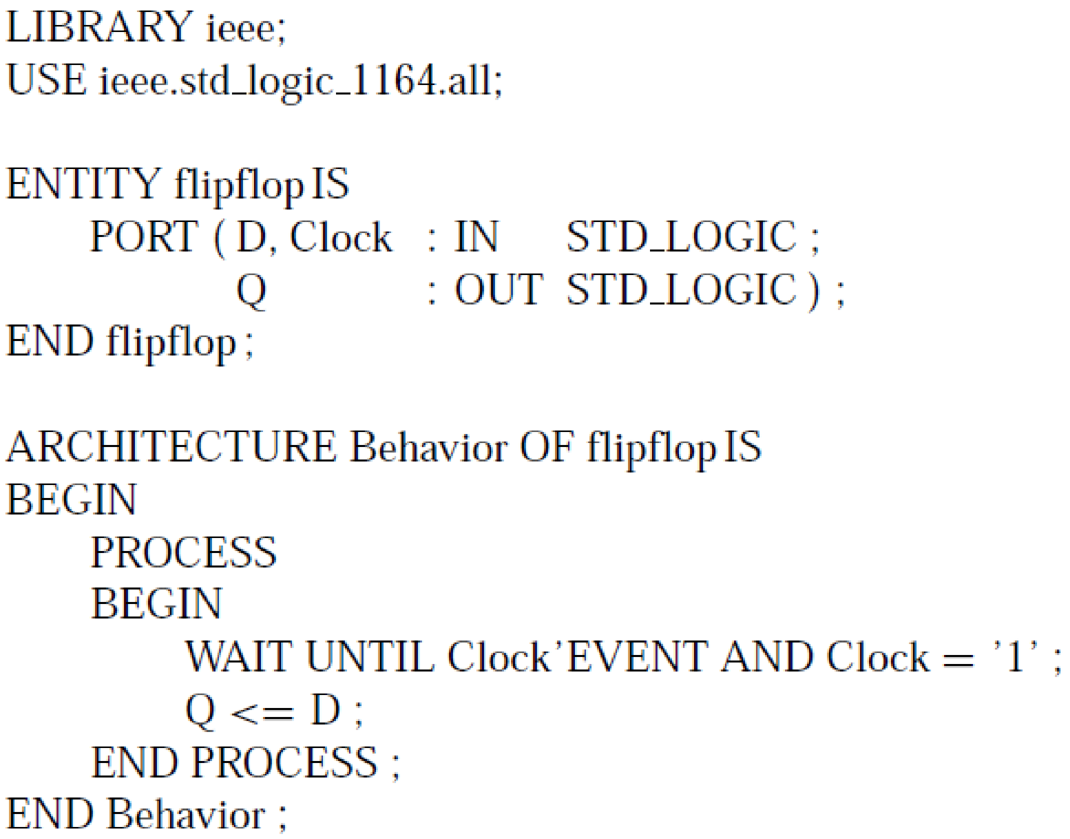
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY latch IS
    PORT ( D, Clk  : IN   STD_LOGIC ;
           Q       : OUT  STD_LOGIC) ;
END latch ;

ARCHITECTURE Behavior OF latch IS
BEGIN
    PROCESS ( D, Clk )
    BEGIN
        IF Clk = '1' THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Alternative code for D flip-flop (positive edge)

- Alternative code
  - Sensitivity list: omitted
  - WAIT UNTIL: implying that sensitivity list only includes clock signal
  - Some VHDL tools accept a simplified WAIT UNTIL statement (i.e., Clock’EVENT is redundant and removed): WAIT UNTIL Clock=‘1’



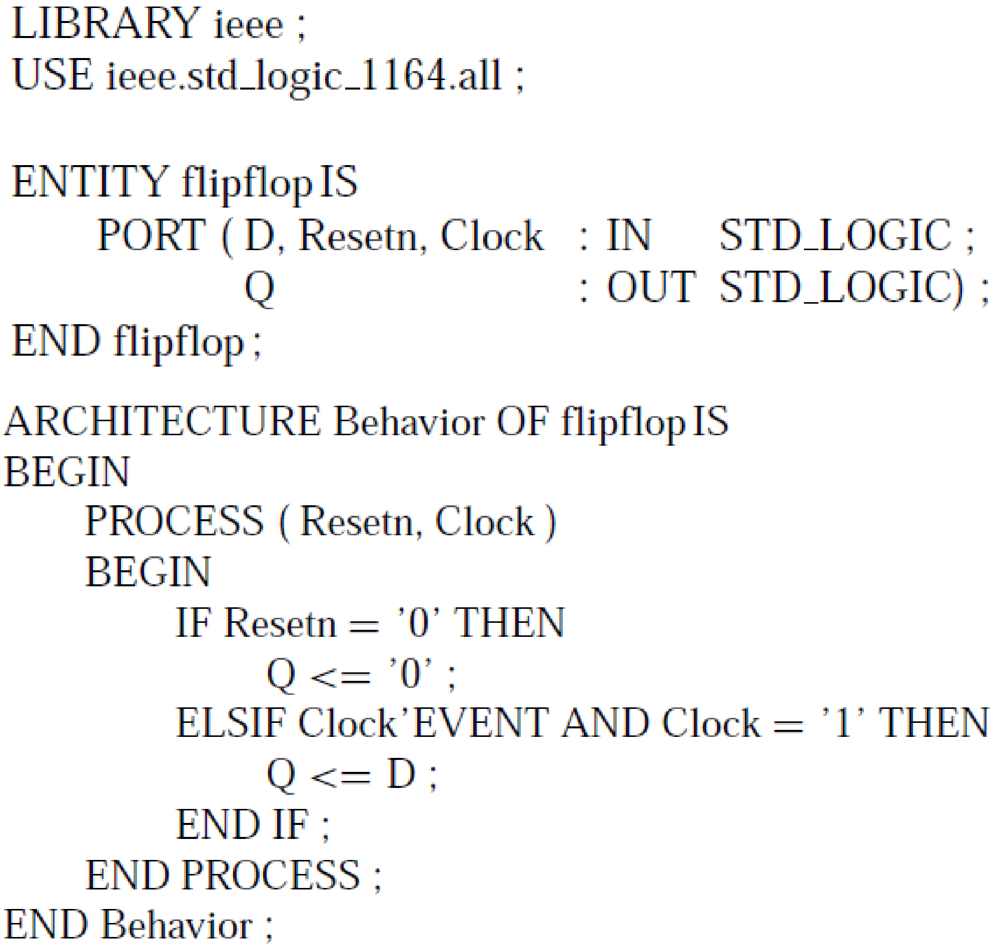
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY flipflop IS
    PORT ( D, Clock   : IN     STD_LOGIC ;
           Q          : OUT    STD_LOGIC ) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS
    BEGIN
        WAIT UNTIL Clock'EVENT AND Clock = '1' ;
        Q <= D ;
    END PROCESS ;
END Behavior ;
```

Code for D flip-flop with asynchronous reset (clear)

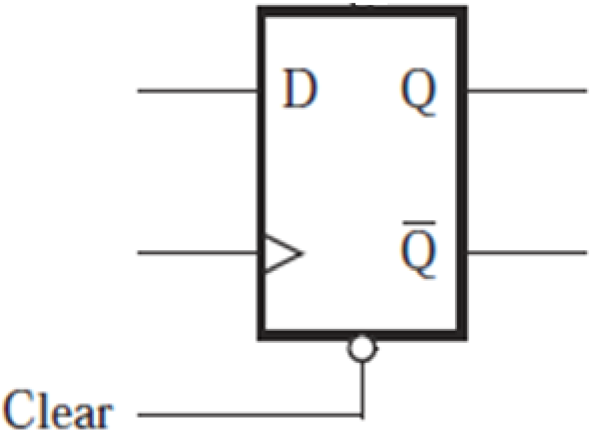
- Asynchronous reset (clear): active low



```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY flipflop IS
    PORT ( D, Resetn, Clock   : IN     STD_LOGIC ;
           Q                  : OUT  STD_LOGIC) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
        IF Resetn = ’0’ THEN
            Q <= ’0’ ;
        ELSIF Clock’EVENT AND Clock = ’1’ THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```



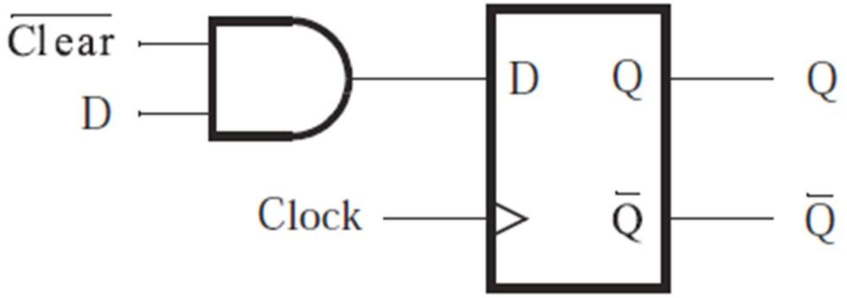
Code for D flip-flop with synchronous reset (clear)

- Synchronous reset (clear) acts at clock edge

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY flipflop IS
    PORT ( D, Resetn, Clock   : IN    STD_LOGIC ;
           Q                  : OUT   STD_LOGIC) ;
END flipflop ;

ARCHITECTURE Behavior OF flipflop IS
BEGIN
    PROCESS
    BEGIN
        WAIT UNTIL Clock’EVENT AND Clock = ’1’ ;
        IF Resetn = ’0’ THEN
            Q <= ’0’ ;
        ELSE
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behavior ;
```



$\overline{\text{Clear}}$
D
Clock
D
Q
$\overline{Q}$
Q
$\overline{Q}$