ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

Synchronous Sequential Logic

Computer Organization
502044

Acknowledgement

This slide show is intended for use in class, and is not a complete document. Students need to refer to the book to read more lessons and exercises. Students have the right to download and store lecture slides for reference purposes; Do not redistribute or use for purposes outside of the course.

[1] Morris R. Mano (Author), Michael D. Ciletti, [2019] **Digital Design: With an Introduction to the Verilog HDL,** 5th Edition.

Sequential Circuits

Latches

Flip-flop

Counters

Register

Chapter Objectives

1. Know about sequence elements.
2. Understand the work of Flip-flops.

Syllabus

5.1 Introduction

5.2 Sequential Circuits

5.3 Storage Elements: Latches

5.4 Storage Elements: Flip-Flops

5.5 Registers

5.6 Shift Registers

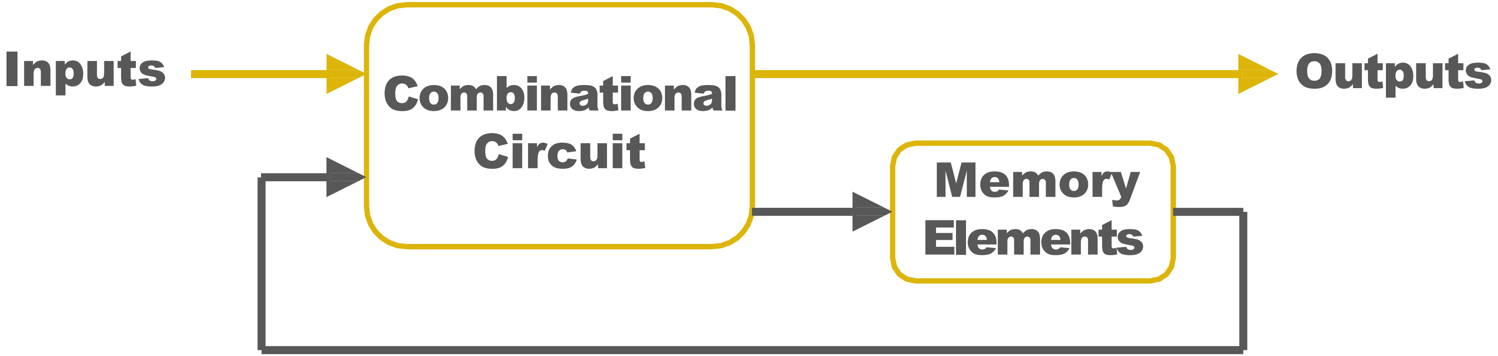
5.7 Ripple Counters

5.8 Random-Access Memory

5.9 Memory Decoding

5.1 Introduction

- Every digital system is likely to have **combinational circuits**.
- Most systems encountered in practice also include **storage elements**, which require that the system be described in terms of **sequential logic**.

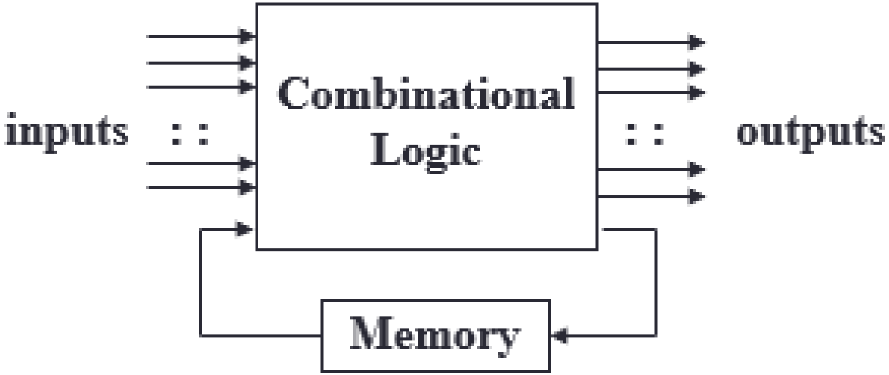


5.2 Sequential Circuits

- The storage elements are devices capable of storing binary information.
- The binary information stored in these elements at any given time defines the state of the sequential circuit at that time.
- The sequential circuit receives binary information from external inputs.
- These inputs, together with the present state of the storage elements, determine the binary value of the outputs.

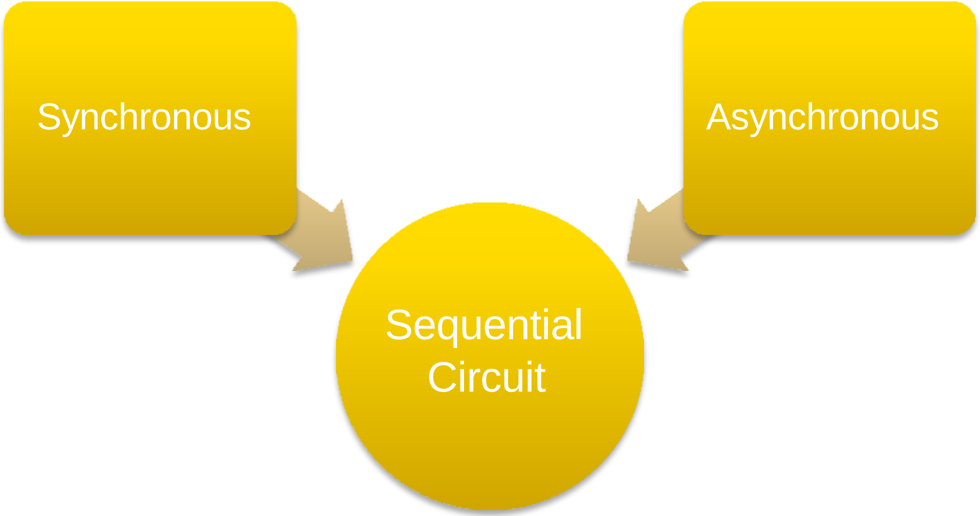
5.2 Sequential Circuits

- They also determine the condition for changing the state in the storage elements.
- A sequential circuit is specified by a time sequence of inputs, output, and internal states.

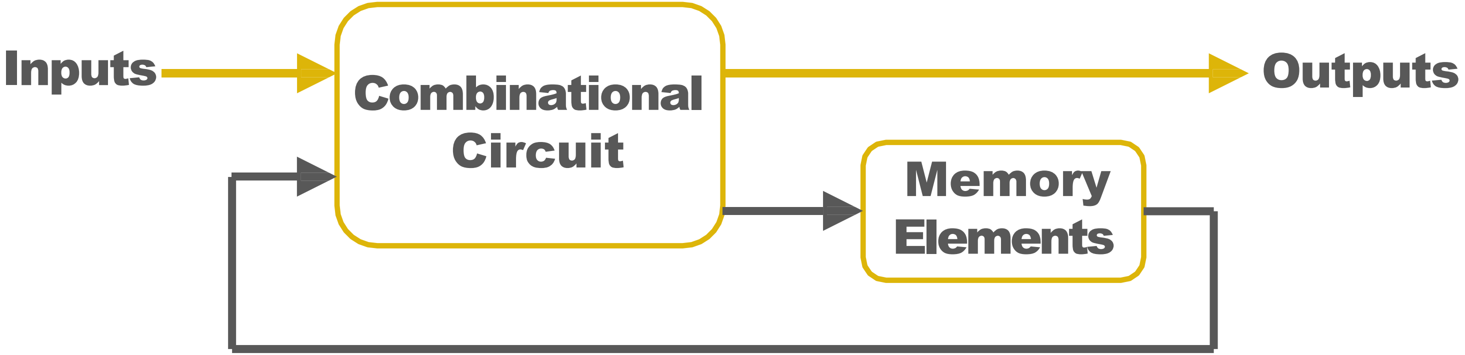


5.2 Sequential Circuits

- There are two main types of sequential circuits.
- Their classification depends on the timing of their signals.



5.2 Sequential Circuits: Asynchronous

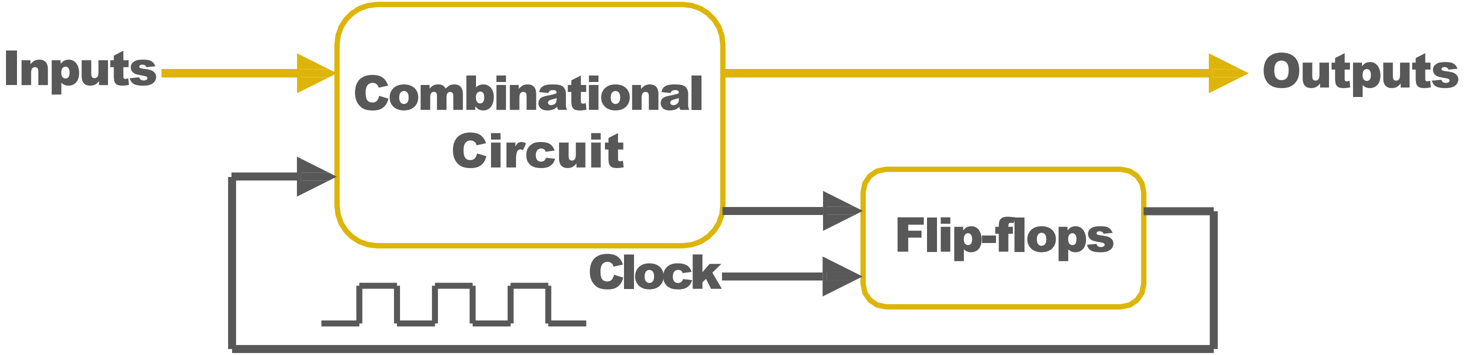


- –   The behaviour of the circuit depends upon the input signals at any instant of time and the order in which the inputs change.

5.2 Sequential Circuits: Asynchronous

- In gate – type asynchronous systems, the storage elements consist of logic gates whose propagation delay provides the required storage.
- Thus, an asynchronous sequential circuit may be regarded as a combinational circuit with feedback.
- Because of the feedback among logic gates, an asynchronous sequential circuit may become unstable at times.

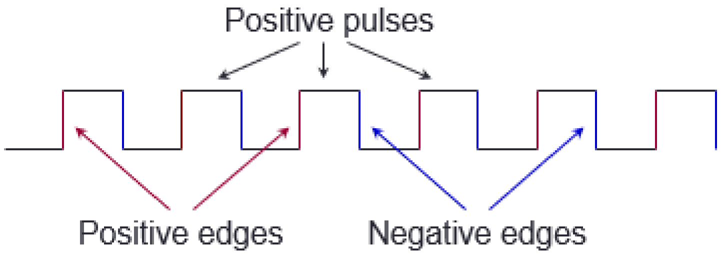
5.2 Sequential Circuits: Asynchronous



The behaviour can be defined from the knowledge of its signals at discrete instants of time.

5.2 Sequential Circuits: Asynchronous

- Synchronous Sequential Circuit
  - Employs signals that affect the storage elements only at discrete instants of time.
  - Synchronisation is achieved by a timing device called a clock generator.
    - Provides a periodic train of clock pulses.
    - Clock pulses are distributed throughout the system in such a way that storage elements are affected only with the arrival of each pulse.

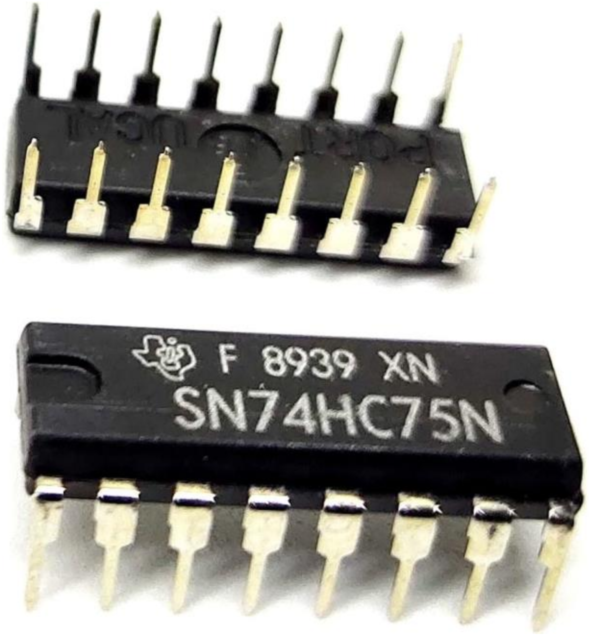


5.2 Sequential Circuits: Asynchronous

- Synchronous Sequential Circuit
    - In practice, the clock pulses are applied with other signals that specify the required change in the storage elements.
- Circuits that use clock pulses in the inputs of storage elements are
- called clocked sequential circuits.
- The storage elements used in clocked sequential circuits are called
- flip – flops.
- A flip – flop is a binary storage device capable of storing one bit of information.

5.3 Latches

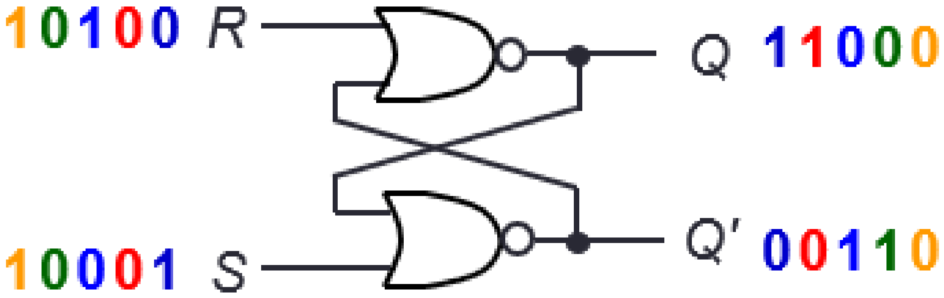
- Latches are the basic circuits from which all flip – flops are constructed.
- Although latches are useful for storing binary information and for the design of asynchronous sequential circuits, they are not practical for use in synchronous sequential circuits.



IC Latch 74HC75, Transparent, Complementary, 11 ns, 5.2 mA

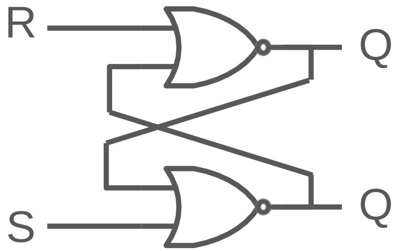
5.3 Latches

- SR Latch

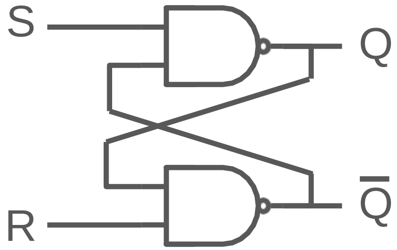


S	R	Q	Q’	
1	0	1	0	Set State
0	0	1	0	Hold State
0	1	0	1	Reset State
0	0	0	1	Hold State
1	1	0	0	Invalid State

5.3 Latches: S-R

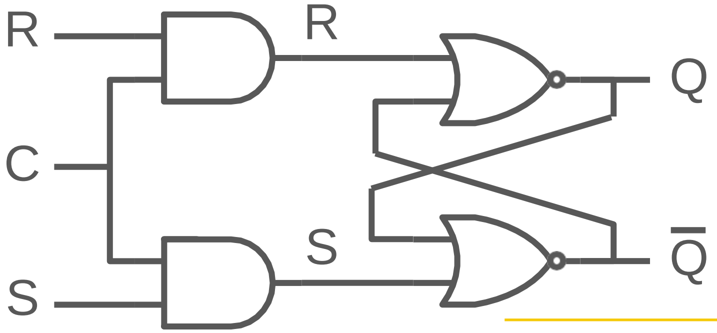


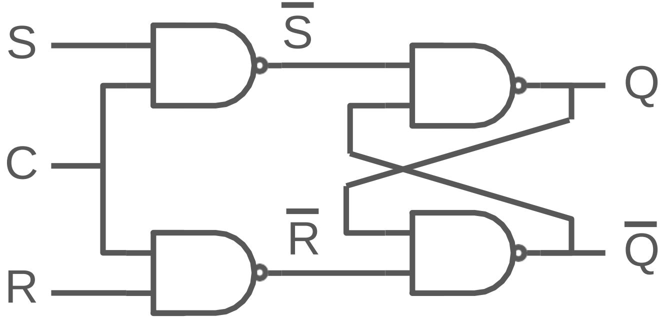
S R	Q	
0 0	Q_0	No change
0 1	0	Reset
1 0	1	Set
1 1	Q=Q'=0	Invalid



S R	Q	
0 0	Q=Q'=1	Invalid
0 1	1	Set
1 0	0	Reset
1 1	Q_0	No change

5.3 Latches: S-R with Enable / Control





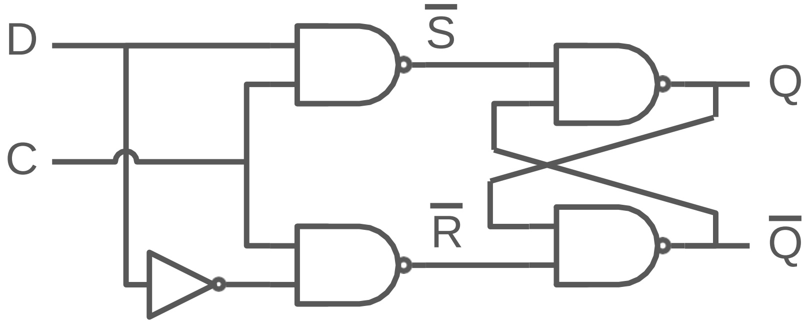
C	S	R	Q	
0	X	X	HOLD	No change
1	0	0	HOLD	No change
1	0	1	Q = 0	Reset
1	1	0	Q = 1	Set
1	1	1	Q = Q’	Invalid

LATCHES

- **D Latch (D = Data)**
  - One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that inputs S and R are never equal to 1 at the same time.
  - D latch has two inputs
    - D (data) - directly goes to the S input and its complement is applied to the R input.
    - C (control)

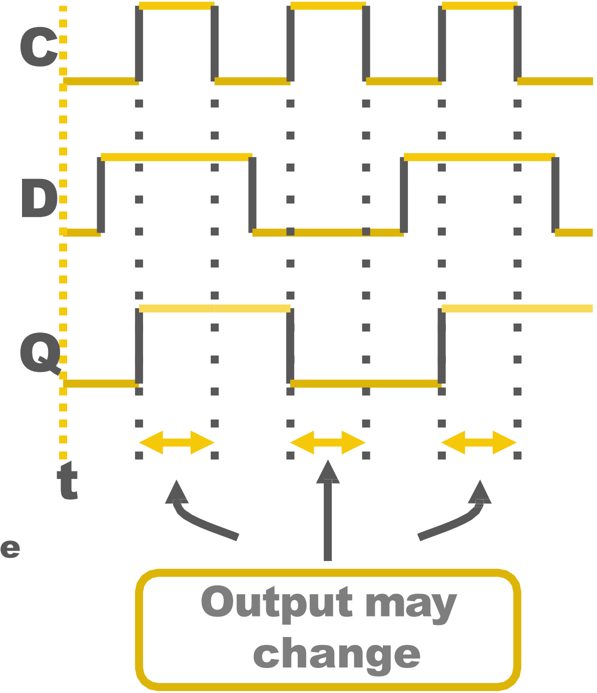
LATCHES

- **D Latch (D = Data)**



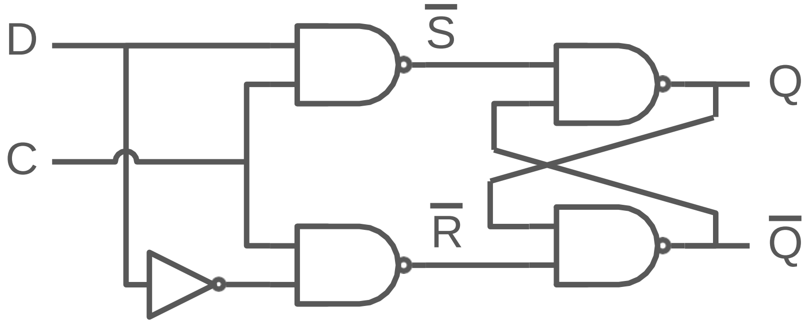
C	D	Q	
0	X	HOLD	No change
1	0	Q = 0	Reset
1	1	Q = 1	Set

Timing Diagram

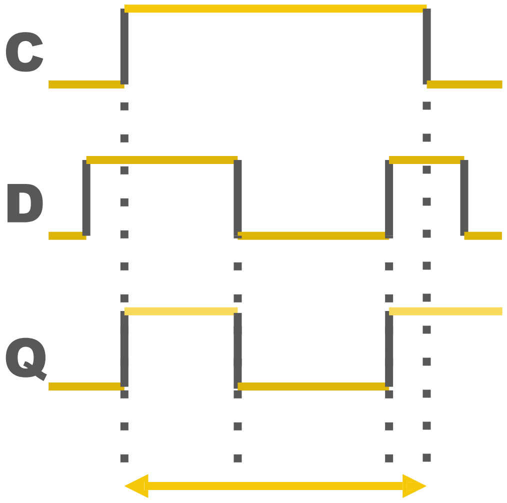


LATCHES

- **D Latch (D = Data)**



Timing Diagram



C	D	Q	
0	X	HOLD	No change
1	0	Q = 0	Reset
1	1	Q = 1	Set

Output may change

LATCHES

- **D Latch (D = Data)**
  - The D latch has an ability to hold data in its internal storage.
  - It is suited for use as a temporary storage for binary information.
  - This circuit is often called **transparent** latch.
    - The output follow changes in the data input as long as the control input is **enabled**.

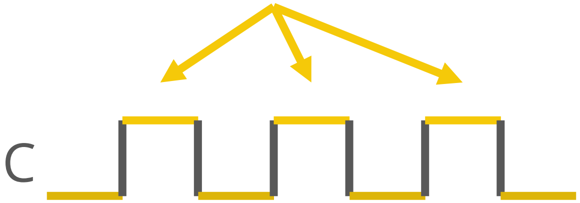
5.3 FLIP - FLOPS

5.4 Flip-flops

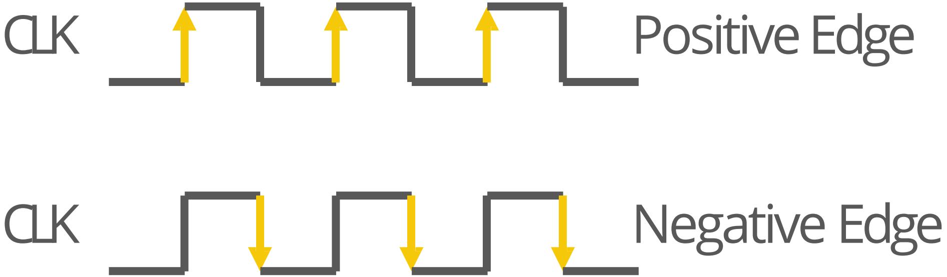
- Flip – flops are constructed in such a way to make D latches operate properly when they are part of a sequential circuit that employs a common clock.
- The problem with the latch is that
  - It responds to a change in the level of a clock pulse.
    - Positive level response in the control input allows changes, in the output when the D input changes while the control pulse stays at logic 1.
- The key to the proper operation of a flip – flop is
  - to trigger it only during a signal transition.

5.4 Flip-flops

- Controlled latches are level – triggered



- Flip-Flops are edge – triggered

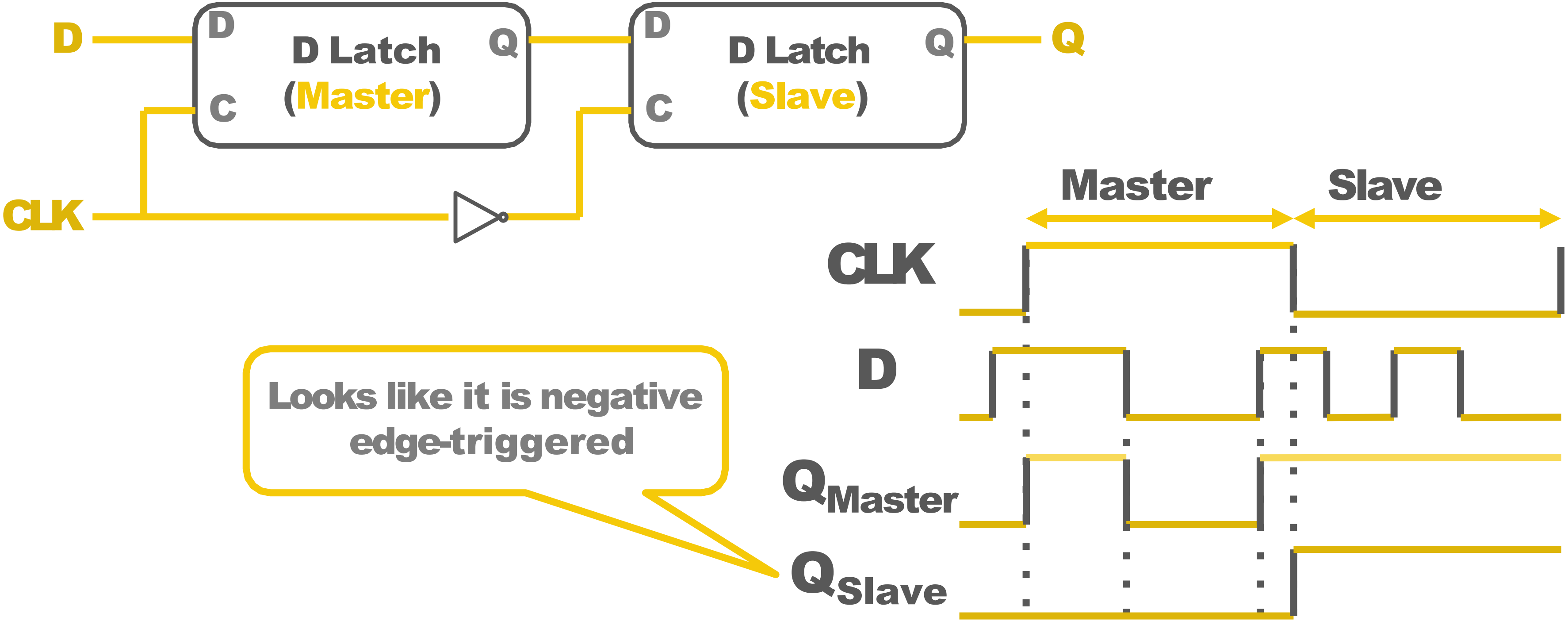


5.4 Flip-flops

- There are two ways that a latch can be modified to form a flip–flop.
    - Employ two latches in a special configuration that
        - isolates the output of the flip – flop from being affected while its input is changing.
    - Produce a flip – flop that triggers only during a signal transition.
        - From 0 to 1 or from 1 to 0 only.
        - Disabled during the rest of the clock pulse duration.

5.4 Flip-flops

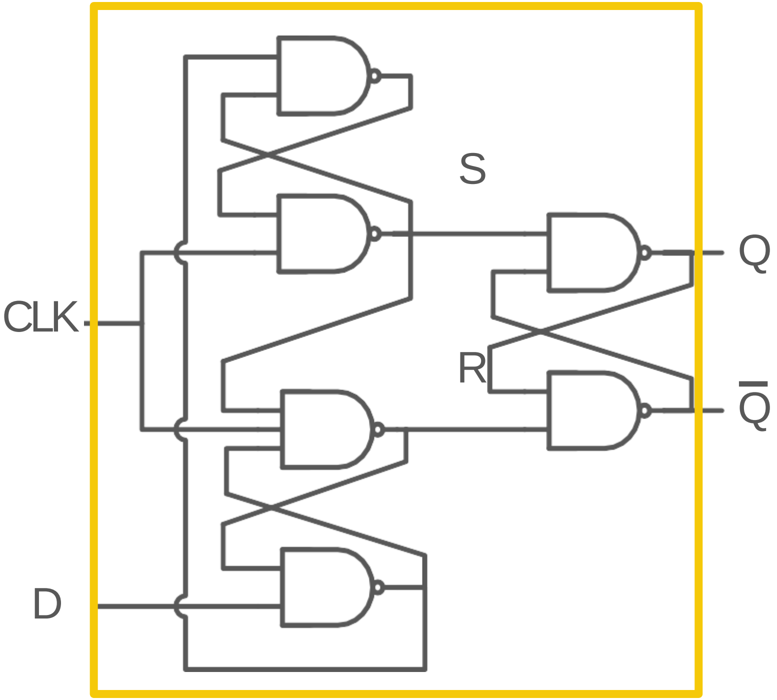
- Master – Slave D flip – flops



Looks like it is negative edge-triggered

5.4 Flip-flops

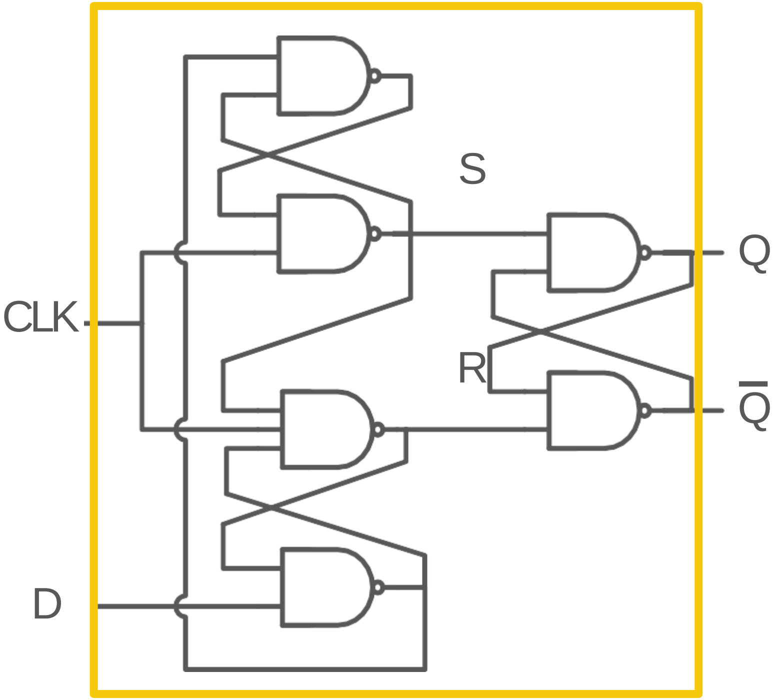
- Edge-Triggered D Flip – Flop



- Two latches respond to the external D (data) and CLK (clock inputs).
- Third latch provides the outputs for the flip – flop.

5.4 Flip-flops

- Edge-Triggered D Flip – Flop



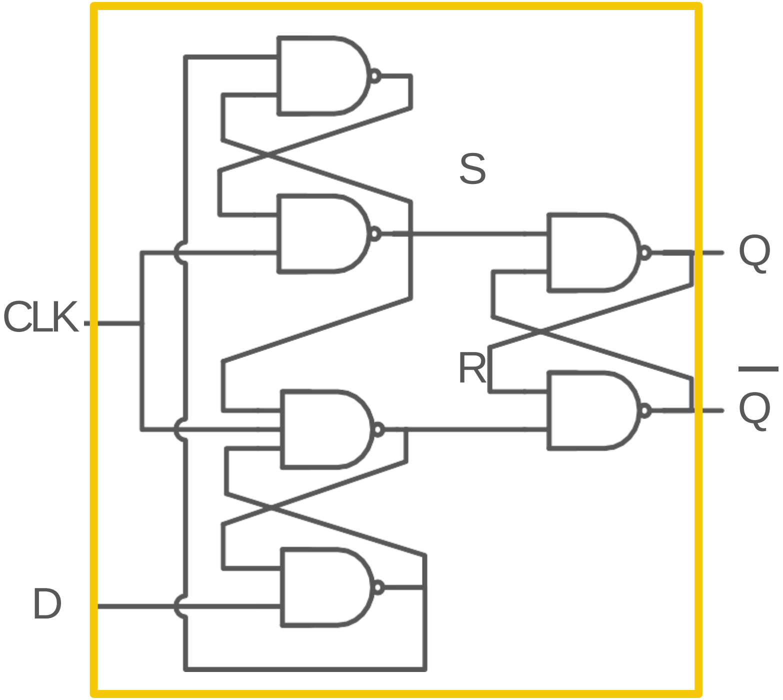
I. When CLK = 0, S = 1 and R =
1.Output = present state.

II. If D = 0, when CLK □ 1

1. R changes to 0
2. Flip – flop goes to the RESET state.
3. Q = 0.

5.4 Flip-flops

- Edge-Triggered D Flip – Flop



III. If D changes when CLK = 1 then

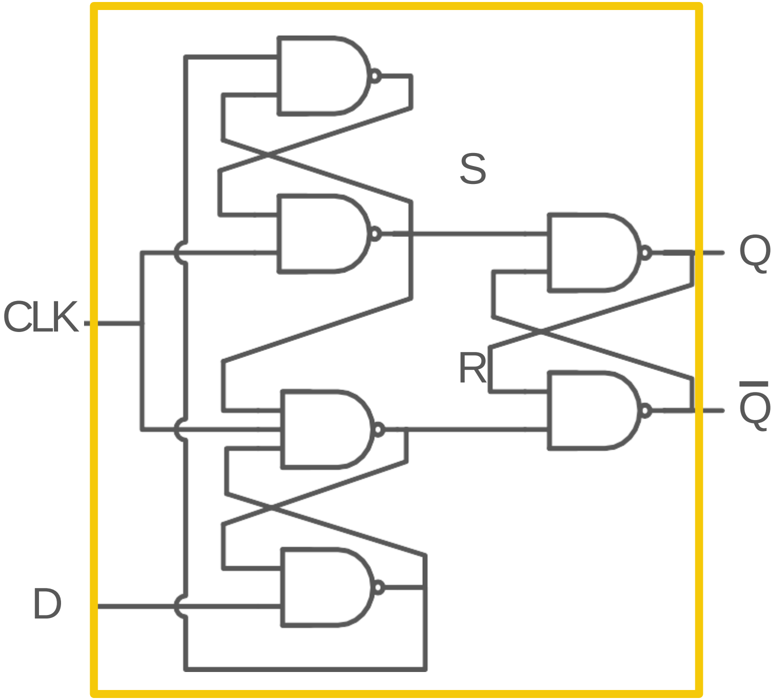
1. R remains at 0.
2. Flip – flop is locked out
3. Unresponsive to further changes in the input.

IV. When CLK □ 0,

1. R □ 1
2. Placing the output latch in the quiescent condition.
3. No change in the output.

5.4 Flip-flops

- Edge-Triggered D Flip – Flop



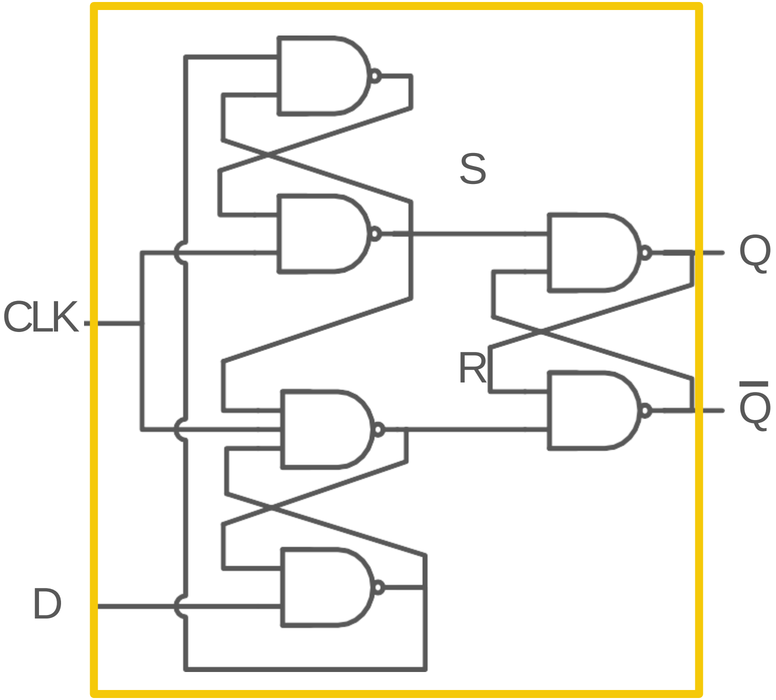
V. If D = 1 when CLK = 0 □ 1,

1. S changes to 0.
2. Circuit goes to SET state
3. Q = 1.
4. Any change in D while CLK = 1

   does not affect the output.

5.4 Flip-flops

- Edge-Triggered D Flip – Flop



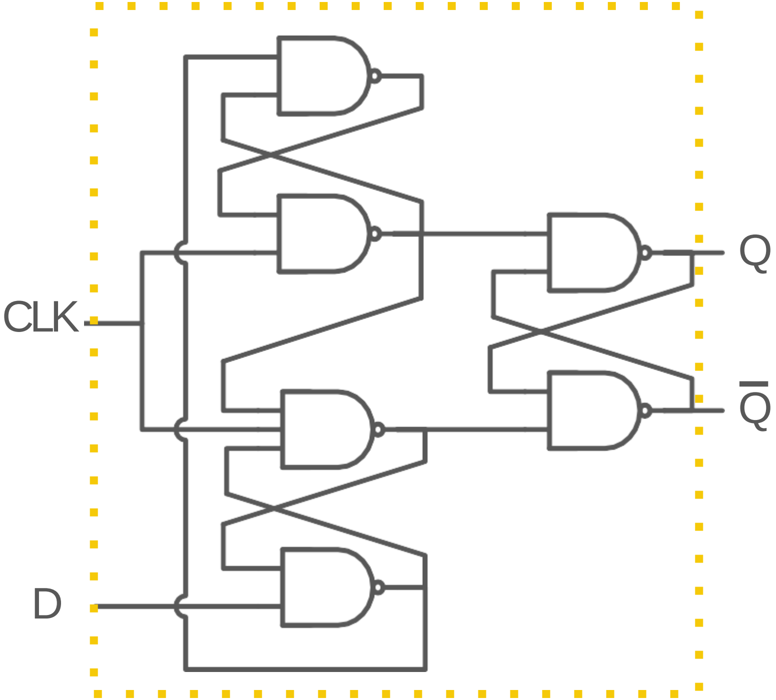
- When CLK in the positive-edge-triggered flip – flop
  - Makes positive transition
    - The value of D is transferred to Q.
  - Makes negative transition
    - Does not affect the output.
  - Steady CLK 1 or 0
    - Does not affect the output.

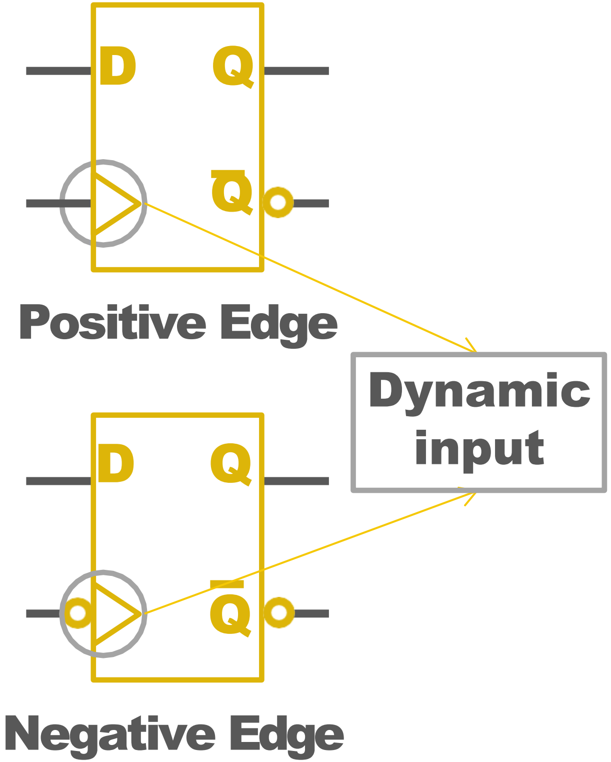
5.4 Flip-flops

- Edge-Triggered D Flip – Flop
  - The timing of the response of a flip – flop to input data and clock must be taken into consideration when using edge – triggered flip - flops.
    - There is a minimum time, called **setup time**, for which the D input must be maintained at a constant value prior to the occurrence of the clock transition.
    - There is a minimum time, called **hold time**, for which the D input must not change after the application of the positive transition of the clock.

5.4 Flip-flops

- Edge-Triggered D Flip – Flop





5.4 Flip-flops

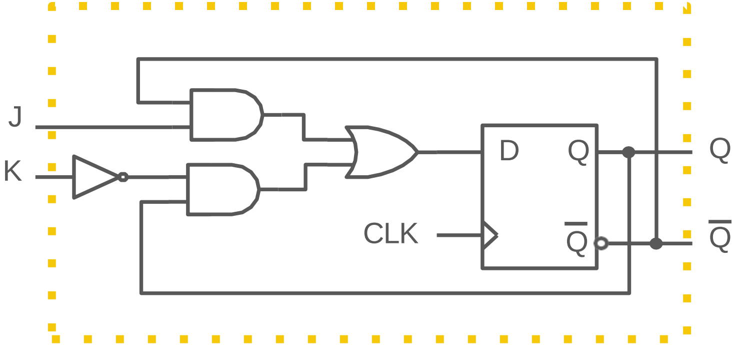
- The most economical and efficient flip – flop constructed is the edge – triggered D flip – flop.
    - It requires smallest number of gates.
- Other types of flip – flops can be constructed by using the D flip – flop and external logic.
    - JK flip – flops
    - T flip - flops

5.4 Flip-flops

- There are three operations that can be performed with a flip – flop:
  - Set it to 1
  - Reset it to 0
  - Complement its output

5.4 Flip-flops

- JK Flip – Flop
  - Performs all three operations.

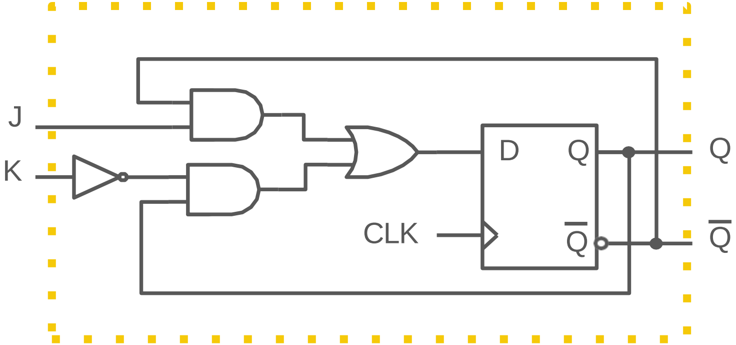


D = JQ' + K'Q

- When J= 1, sets the flip – flop to 1.
- When K = 1, resets the flip – flop to 0.

5.4 Flip-flops

- JK Flip – Flop



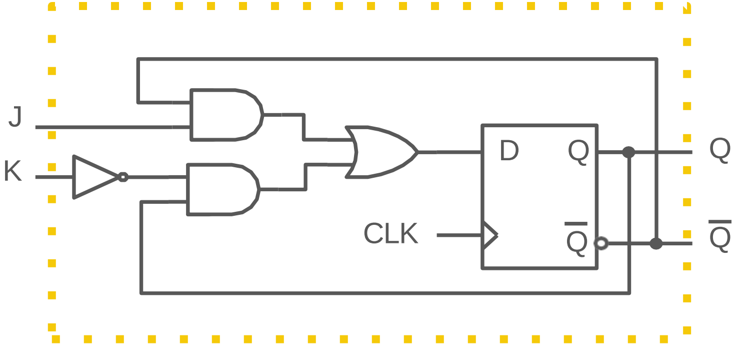
D = JQ' + K'Q

Operation 1

- When J= 1 and K = 0,
  - D = 1.Q' + 1.Q (Post2b)
  - D = Q' + Q (Post5a)
  - D = 1
  - Next clock edge sets the output to 1.

5.4 Flip-flops: J-K

- JK Flip – Flop



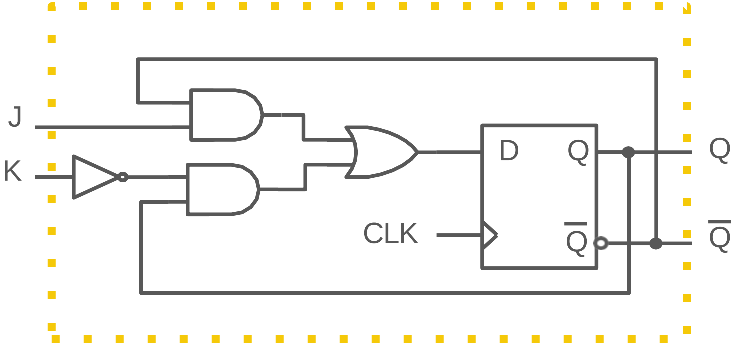
D = JQ' + K'Q

Operation 2

- When J= 0 and K = 1,
  - D = 0.Q' + 0.Q (Theo2b)
  - D = 0 + 0
  - D = 0
  - Next clock edge sets the output to 0.

5.4 Flip-flops: J-K

- JK Flip – Flop



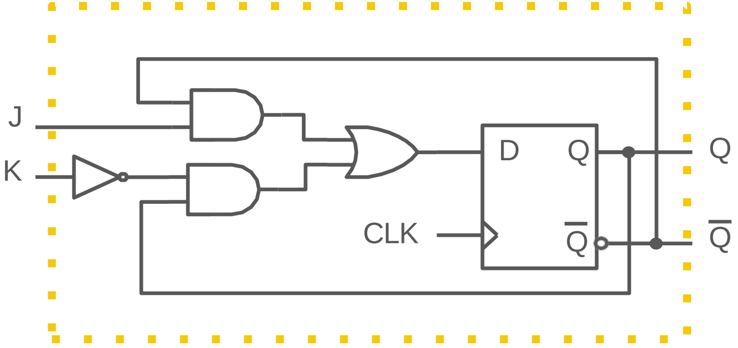
D = JQ’ + K’Q

Operation 3

- When J= 1 and K = 1,
  - D = 1.Q’ + 0.Q (Post2b)
  - D = Q’ + 0 .Q (Theo2b)
  - D = Q’ + 0 (Post2a)
  - D = Q’
  - Next clock edge complements the output.

5.4 Flip-flops: J-K

- JK Flip – Flop

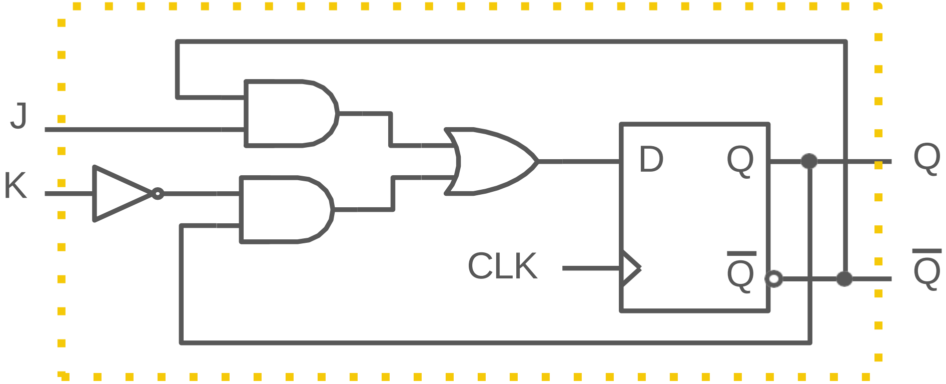


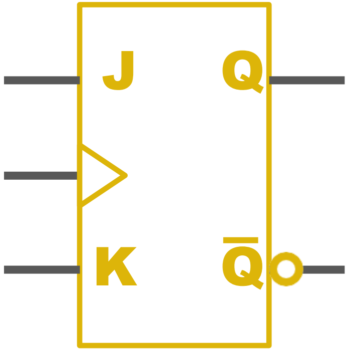
D = JQ' + K'Q

- When J= 0 and K = 0,
  - D = 0.Q' + 1.Q (Theo2b)
  - D = 0 + 1 .Q (Post2b)
  - D = 0 + Q (Post2a)
  - D = Q
  - Next clock edge the output is unchanged.

5.4 Flip-flops: J-K

- JK Flip – Flop

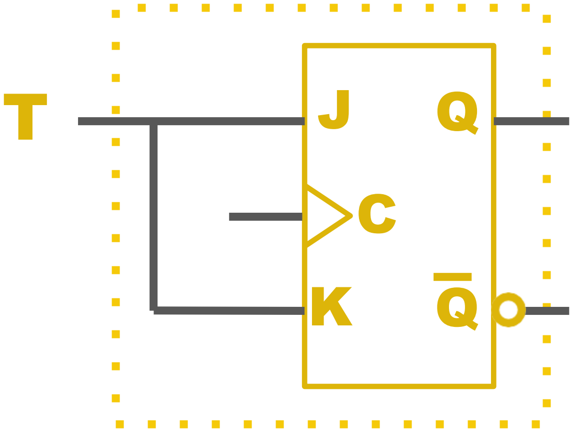




D = JQ' + K'Q

5.4 Flip-flops: J-K

- T (toggle) Flip – Flop
  - Complementing flip – flop.
  - Can be obtained from a JK flip – flop.
  - When inputs Jand K are tied together.
  - Useful for designing binary counters.

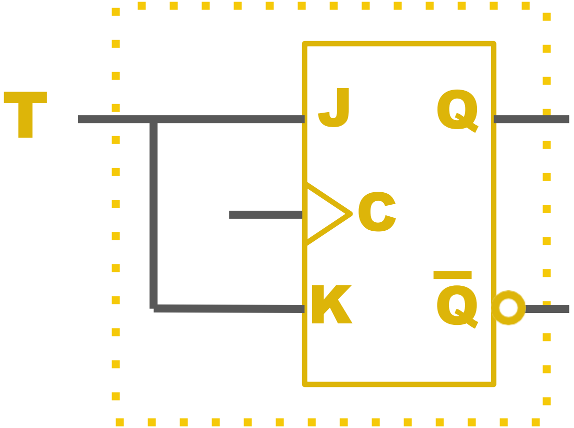


$$D = JQ' + K'Q$$

$$D = TQ' + T'Q = T \,\square\, Q$$

FLIP – FLOPS

- T (toggle) Flip – Flop
  - When T = 0 (J = K = 0)
  - A clock edge does not change the output.
  - When T = 1 (J = K = 1)
  - A clock edge complements the output.

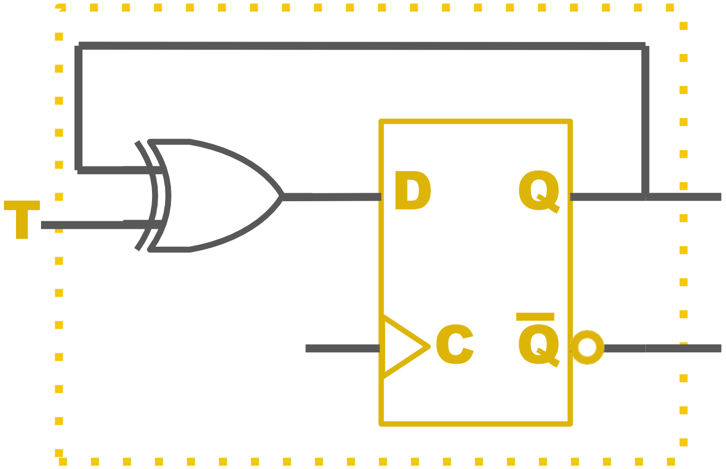


$D = JQ' + K'Q$

$D = TQ' + T'Q = T \;□\; Q$

5.4 Flip-flops: J-K

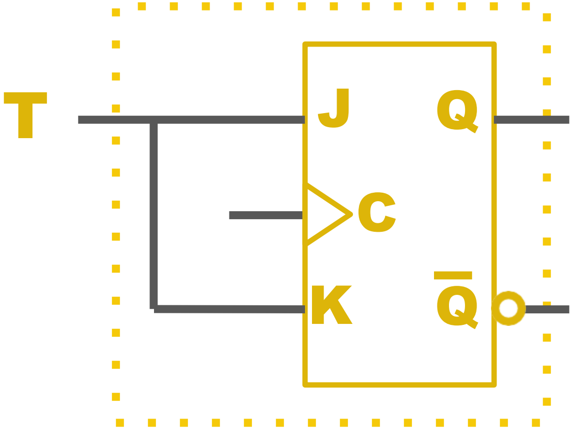
- T (toggle) Flip – Flop
  - Can be constructed with a D flip – flop and an XOR gate.
  - When T = 0 then D = Q
    - No change in the output.
  - When T = 1 then D = Q’
    - Output complements



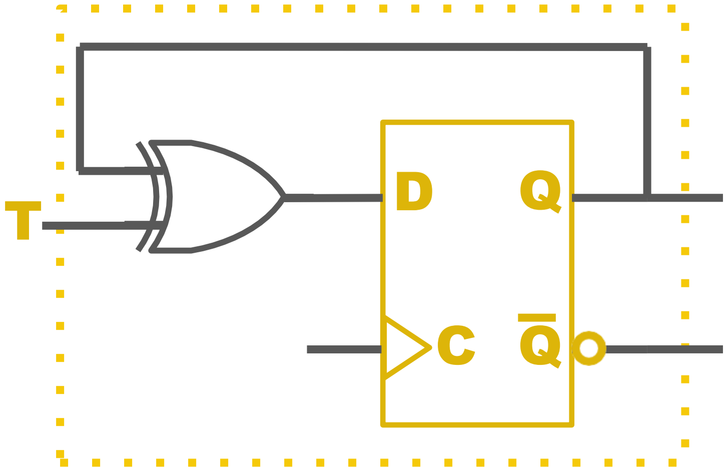
D = TQ’ + T’Q = T □ Q

5.4 Flip-flops: J-K

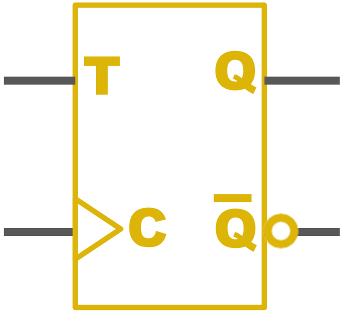
- T (toggle) Flip – Flop



(a) From JK Flip – Flop



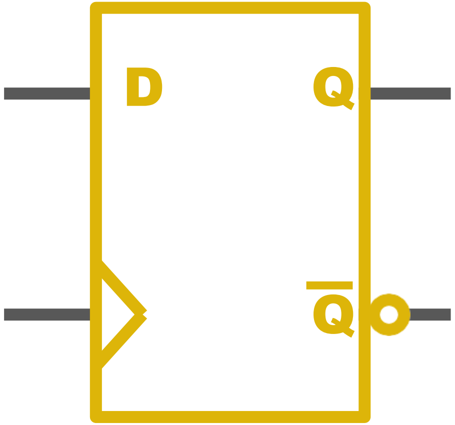
(b) From D Flip – Flop



(c) Graphic Symbol

5.4 Flip-flops: J-K

- Flip – Flop Characteristics Table

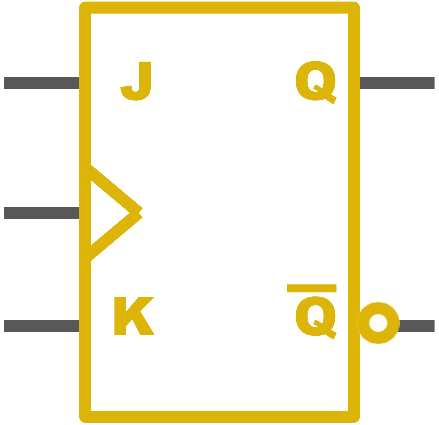


D	Q (t+1)	
0	0	Reset
1	1	Set

$$Q(t+1) = D$$

5.4 Flip-flops: J-K

- Flip – Flop Characteristics Table

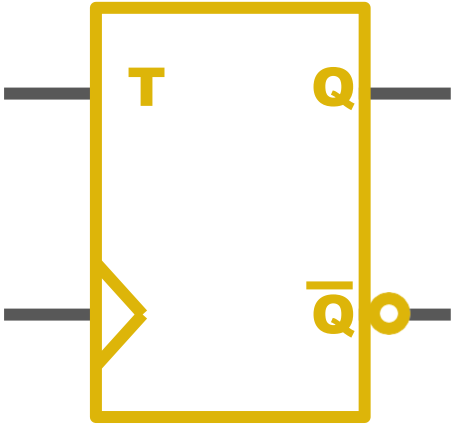


J	K	Q (t+1)	
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Toggle

$$Q(t+1) = JQ' + K'Q$$

5.4 Flip-flops: T

- Flip – Flop Characteristics Table



T	Q (t+1)	
0	Q(t)	No change
1	Q'(t)	Toggle

$$Q(t+1) = T \oplus Q$$

FLIP – FLOPS

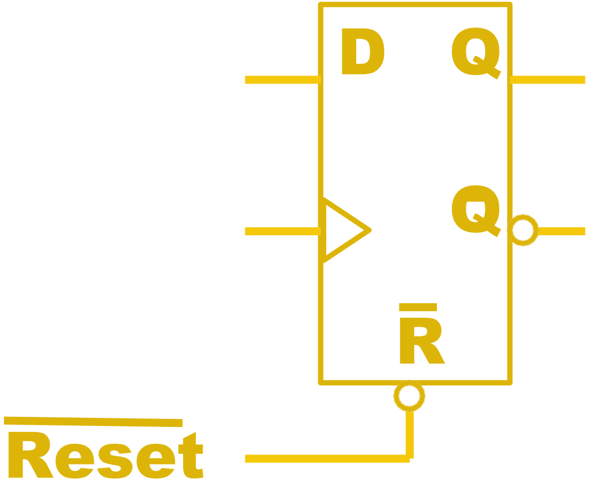
- Some flip – flops have **asynchronous** inputs that are used to force the flip – flop to a particular state **independent** of the clock.
- The input that sets the flip – flop to 1 is called **preset**.
- The input that clears the flip – flop to 0 is called **clear** or **direct reset**.
- When power is on in a digital system, the state of the flip flop is unknown.

FLIP – FLOPS

- When power is on in a digital system, the state of the flip flop is unknown.
- The direct inputs are useful for bringing all flip – flops in the system to a known starting state prior to the clocked operation.

FLIP - FLOPS

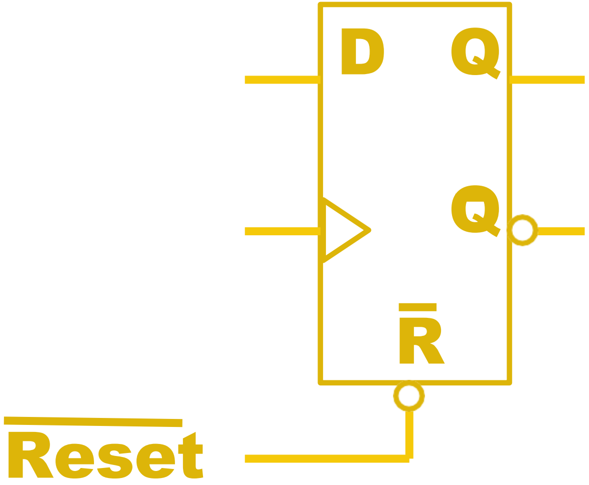
- Asynchronous Reset



R’	D	CLK	Q(t+1)
0	x	x	0

FLIP - FLOPS

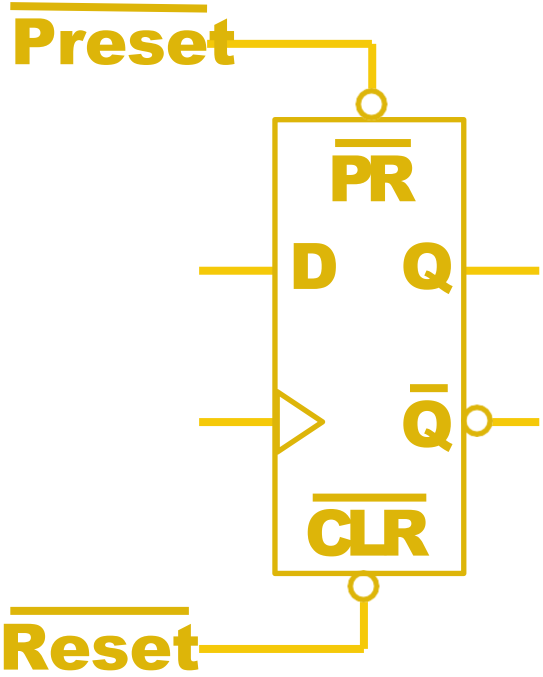
- Asynchronous Reset



R’	D	CLK	Q(t+1)
0	x	x	0
1	0	↑	0
1	1	↑	1

FLIP – FLOPS

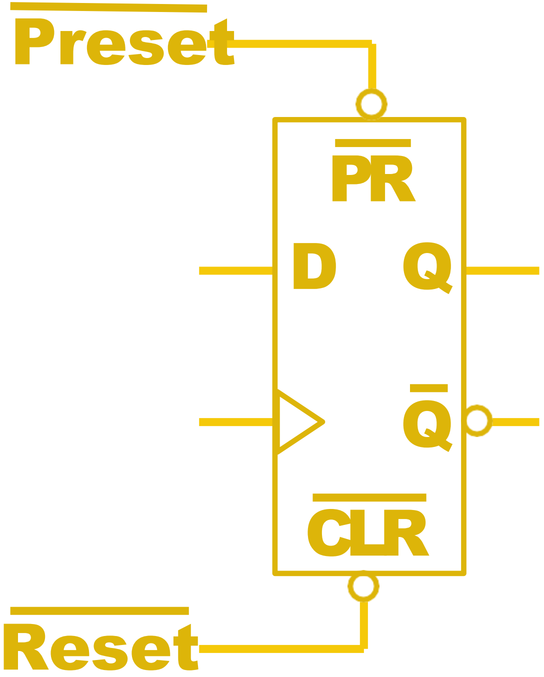
- Asynchronous Preset and Clear



PR’	CLR’	D	CLK	Q(t+1)
1	0	x	x	0

Flip-flops

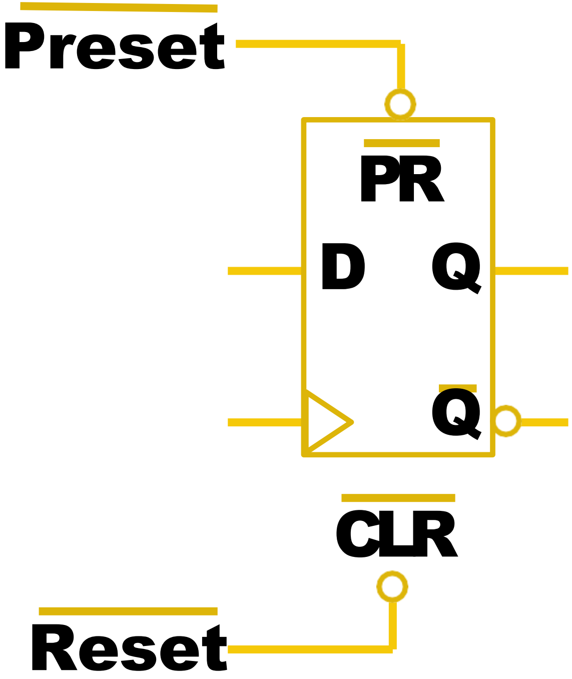
- Asynchronous Preset and Clear



PR’	CLR’	D	CLK	Q(t+1)
1	0	x	x	0
0	1	x	x	1

Flip-flops

- Asynchronous Preset and Clear



PR’	CLR’	D	CLK	Q(t+1)
1	0	x	x	0
0	1	x	x	1
1	1	0	↑	0
1	1	1	↑	1

5.5 Registers

- A *register* is a group of flip-flops, each one of which shares a common clock and is capable of storing one bit of information.
- An *n* -bit register consists of a group of *n* flip-flops capable of storing *n* bits of binary information

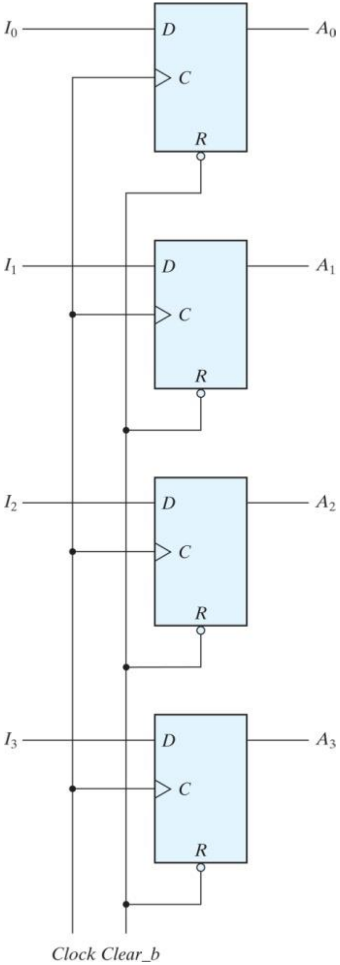


FIGURE 6.1 Four-bit register

5.6 Shift Registers

- The logical configuration of a shift register consists of a chain of flip-flops in cascade, with the **output of one flip-flop** connected to the **input of the next flip-flop**.
- All flip-flops receive **common clock pulses**, which activate the shift of data from one stage to the next.

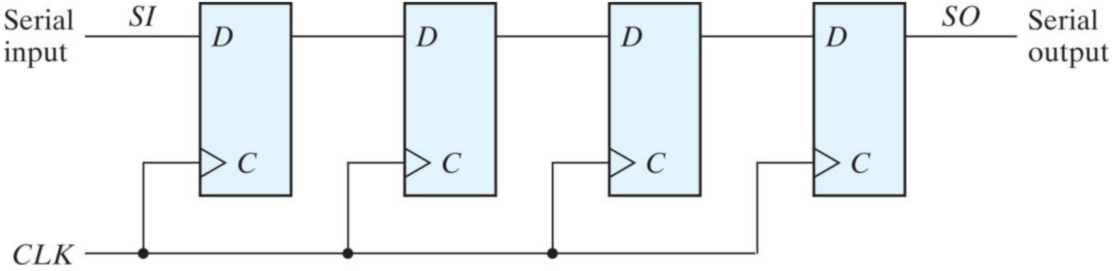
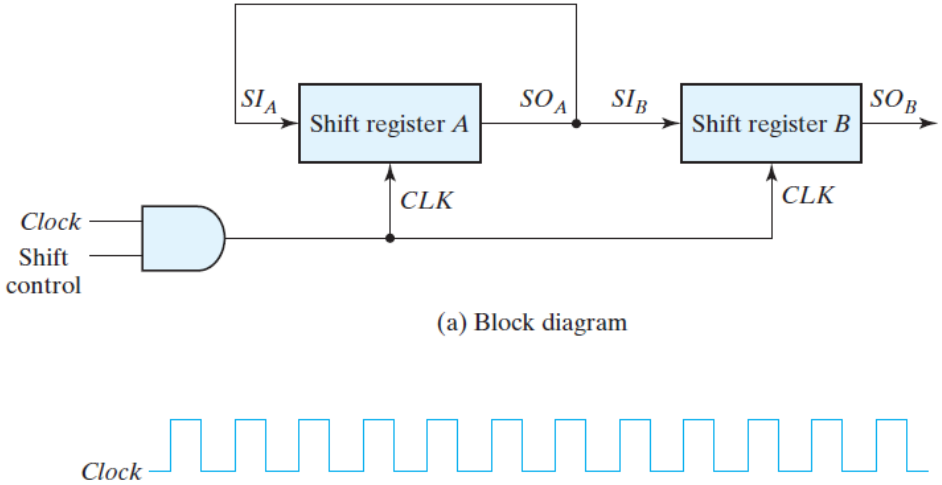


FIGURE 6.3 Four-bit shift register

5.6 Shift Registers: application

- Serial transfer
- Serial addition
- Universal shift register

Fig: Serial transfer from register A to register B



(a) Block diagram

5.7 Counters

- A register that goes through a prescribed sequence of states upon the application of input pulses is called a *counter*.
- The input pulses may be:
    - clock pulses,
    - from external source.
- A *binary counter:*
    - follows the binary number sequence
    - *n* -bit binary counter consists of *n* flip-flops and can count in binary from 0 through 2^n - 1.
- Catalogues:
    - ripple counters: a flip-flop output transition trigger next flip-flops.
    - synchronous counters: receive the common clock.

5.7 Ripple counters

Table 6.8
Binary Count Sequence

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

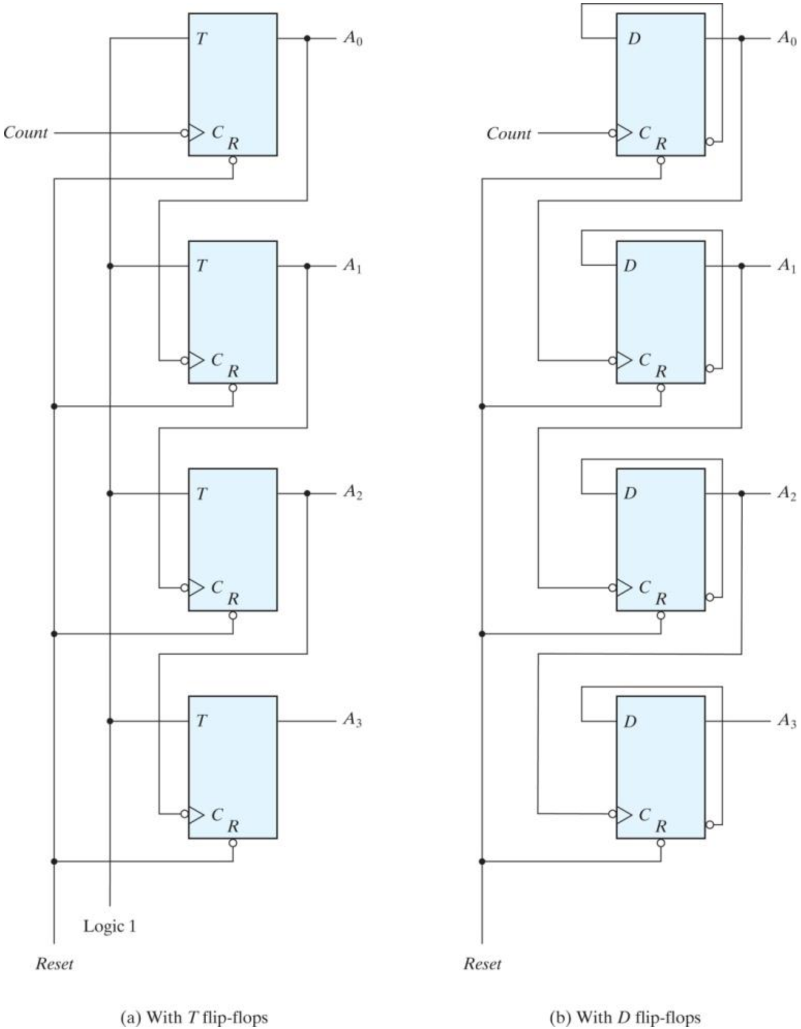


FIGURE 6.12 *Four-bit binary ripple counter*

5.7 Ripple counters: BCD counter

FIGURE 6.13 *State diagram of a decimal BCD counter*

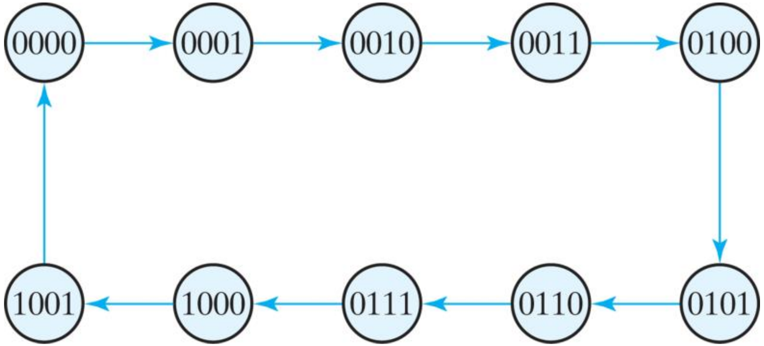
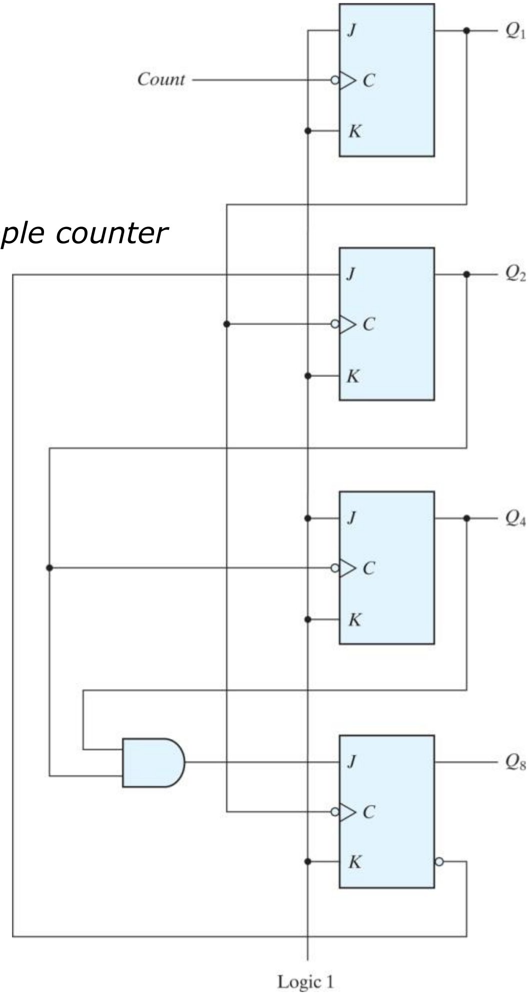


FIGURE 6.14 *BCD ripple counter*



5.7 Ripple counters: BCD counter (more digits)

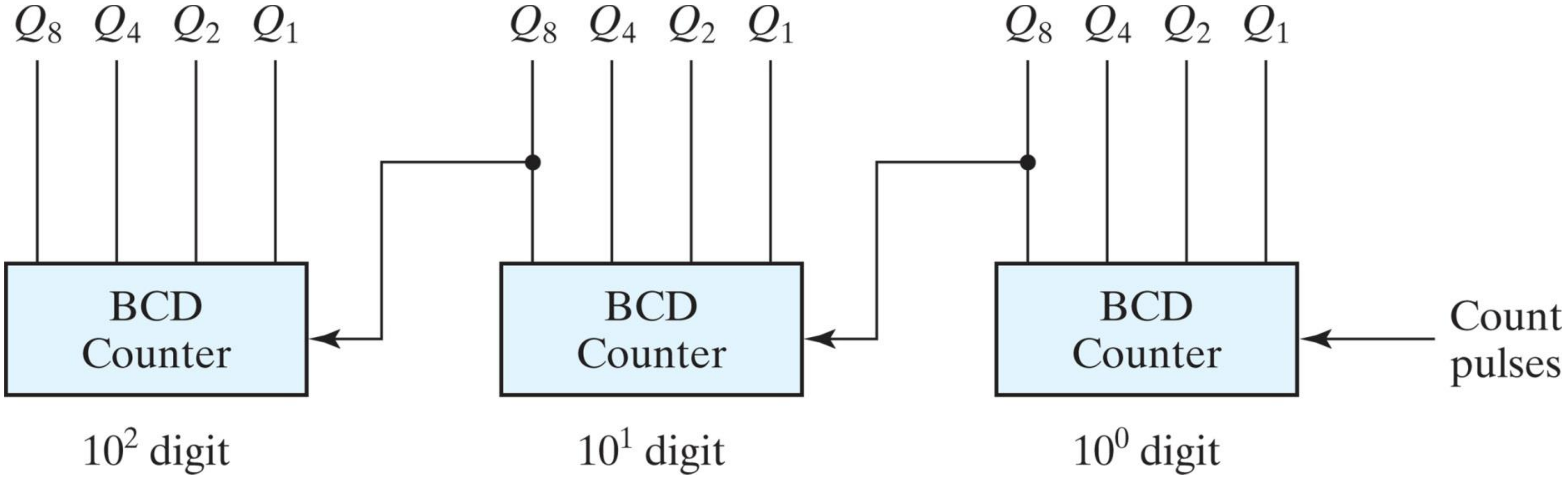
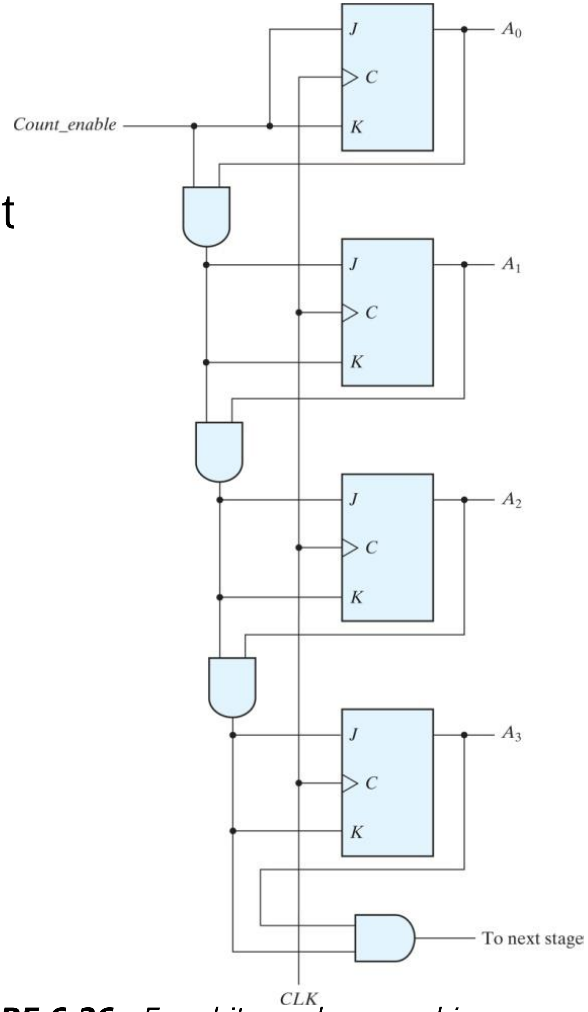
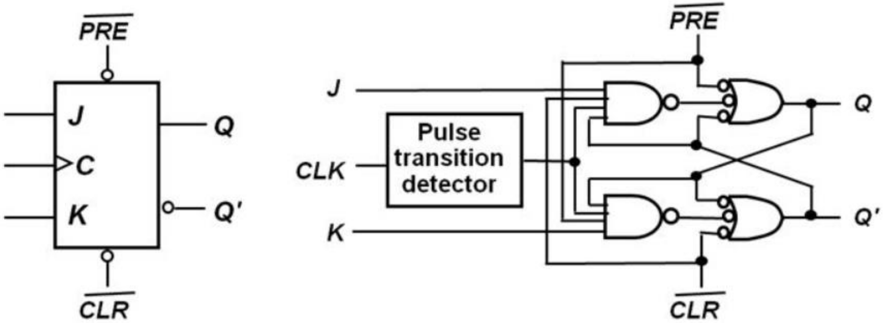


FIGURE 6.15 Block diagram of a three-decade decimal BCD counter

5.7 Synchronous counters:

- Synchronous counters: acommon clock pulse t simultaneously.
  - Binary counter
  - Up–Down Binary Counter
  - BCD Counter
  - Binary Counter with Parallel Load



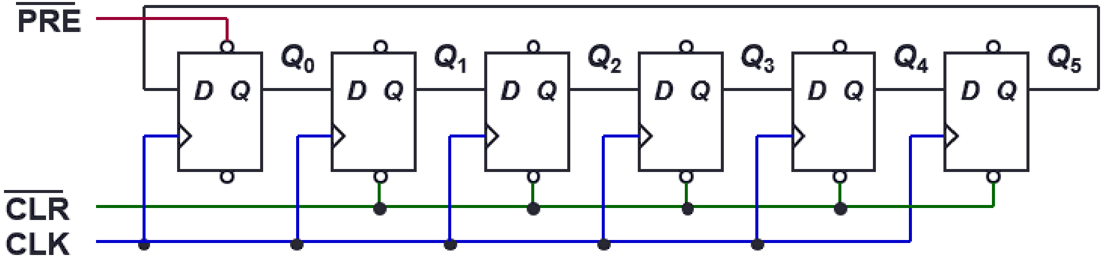


J-K flip-flop with asynchronous Preset / Clear

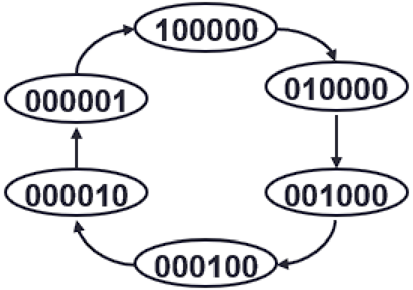
FIGURE 6.26 *Four-bit synchronous binary counter*

5.7 Other counter: Ring counter

- An *n-bit* ring counter cycles through *n* states.
- Example: A 6-bit ring counter (also called mod-6 ring counter)

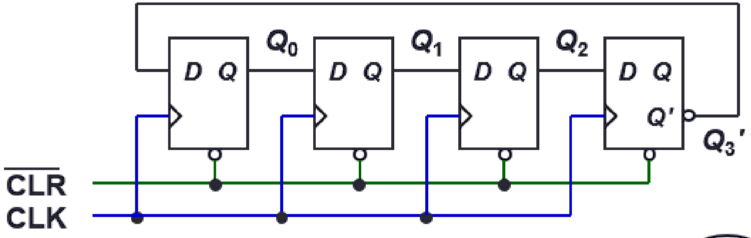


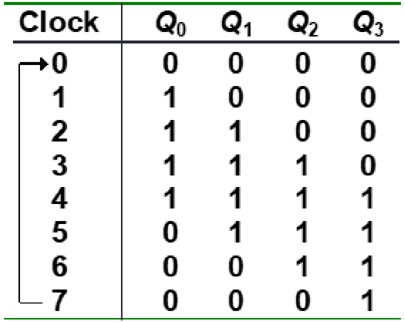
Clock	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5
0	1	0	0	0	0	0
1	0	1	0	0	0	0
2	0	0	1	0	0	0
3	0	0	0	1	0	0
4	0	0	0	0	1	0
5	0	0	0	0	0	1



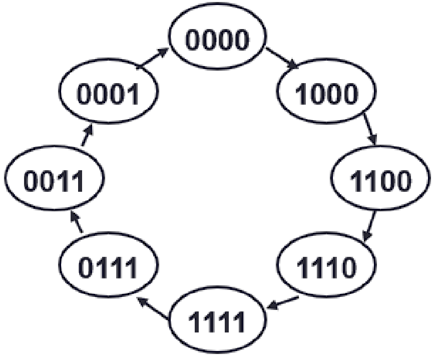
5.7 Other counter: Johnson counter

- An n-bit Johnson counter (twisted-ring counter) cycles through 2*n states.
- Example: A 4-bit John counter (also called mod-8 Johnson counter)





Clock	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1



0000
1000
1100
1110
1111
0111
0011
0001