Unit 4.
Sequential Systems

Contents

- Basic Concepts
- Latches and flip-flops
- Registers
- Counters
- Sequential System Design

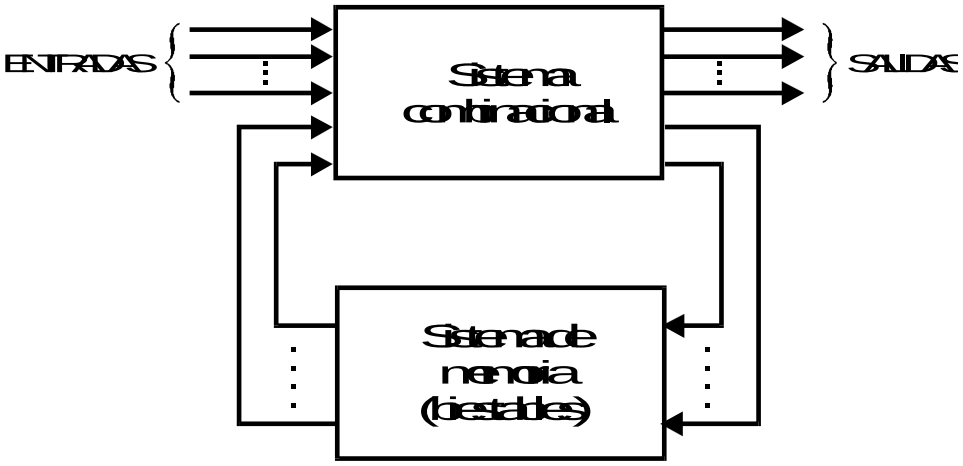
Bibliography

- Digital fundamentals.
  Thomas Floyd. Prentice-Hall.
- Digital Design.
  M. Morris Mano. Prentice-Hall
- Introduction to Digital Logic Design.
  John P. Hayes. Addison-Wesley

Basic concepts

Sequential circuit. Circuit in which the outputs in a concrete instant are function of the inputs in that instant and the state of the circuit, i.e., they store information

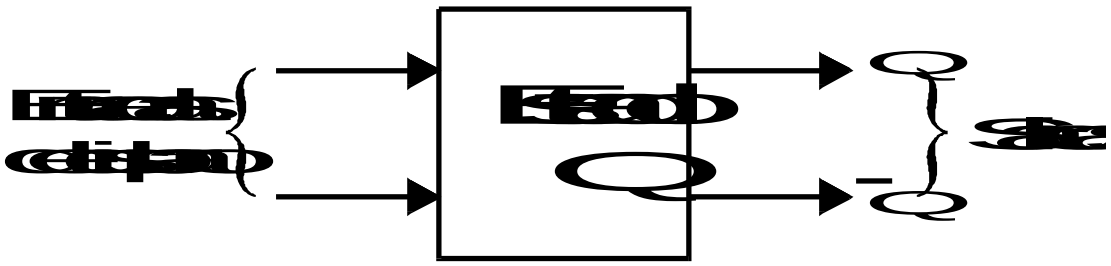
- Therefore a sequential system is formed by two different blocks: a combinational system to process information and a memory system to store it.



Generally, feedback circuits are present in sequential systems.

Latches and flip-flops (I)

- Information is stored in binary, and the basic memory elements are **latches** and **flip-flops** which store just one bit of information.
- They are elementary logical circuits that can remain in one of the two possible states (Q=0 or Q=1) and switch among them depending on the triggering inputs.
- There are many types, but the general scheme is:



Latches and flip-flops (II)

Classification:

Depending on **triggering method:**

R-S **J-K** **D** **T**

Depending on **triggering synchronization:**

- **Synchronous.** Switching among states occurs in synchrony with a clock signal
- **Asynchronous.** Switching among states can occurs in any moment; it just depend on the triggering inputs.

Depending on the form of the **triggering signal:**

- **Level-triggered.** Triggering and change of state occurs when a low or high level is detected in the inputs.
- **Edge-triggered (synchronous flip-flops):** Triggering and change of state occurs just when the clock changes from low to high (rising edge) or from high to low (falling edge).

Latches and flip-flops (III)

R-S NOR

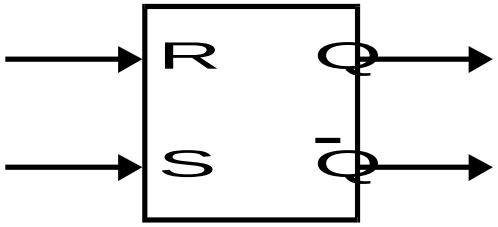
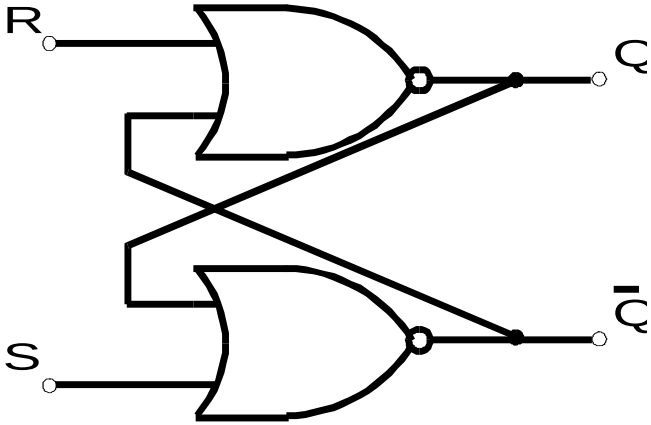


Tabla de excitación		
R_t	S_t	Q_{t+1}
0	0	Q_t
0	1	1
1	0	0
1	1	I



R=S=1: Q is not determined.
Not valid input

Latches and flip-flops (IV)

RS-NAND

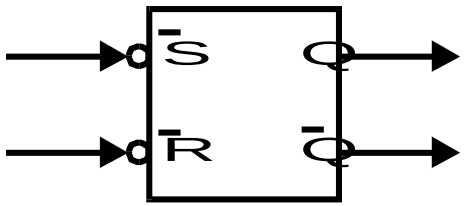
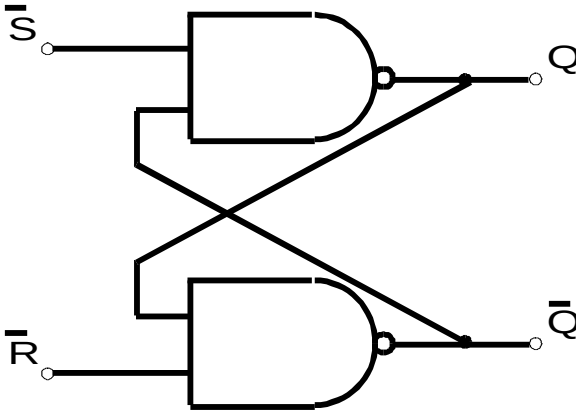


Tabla de excitación		
R_n	S_n	Q_{n+1}
0	0	I
0	1	0
1	0	1
1	1	Q_n

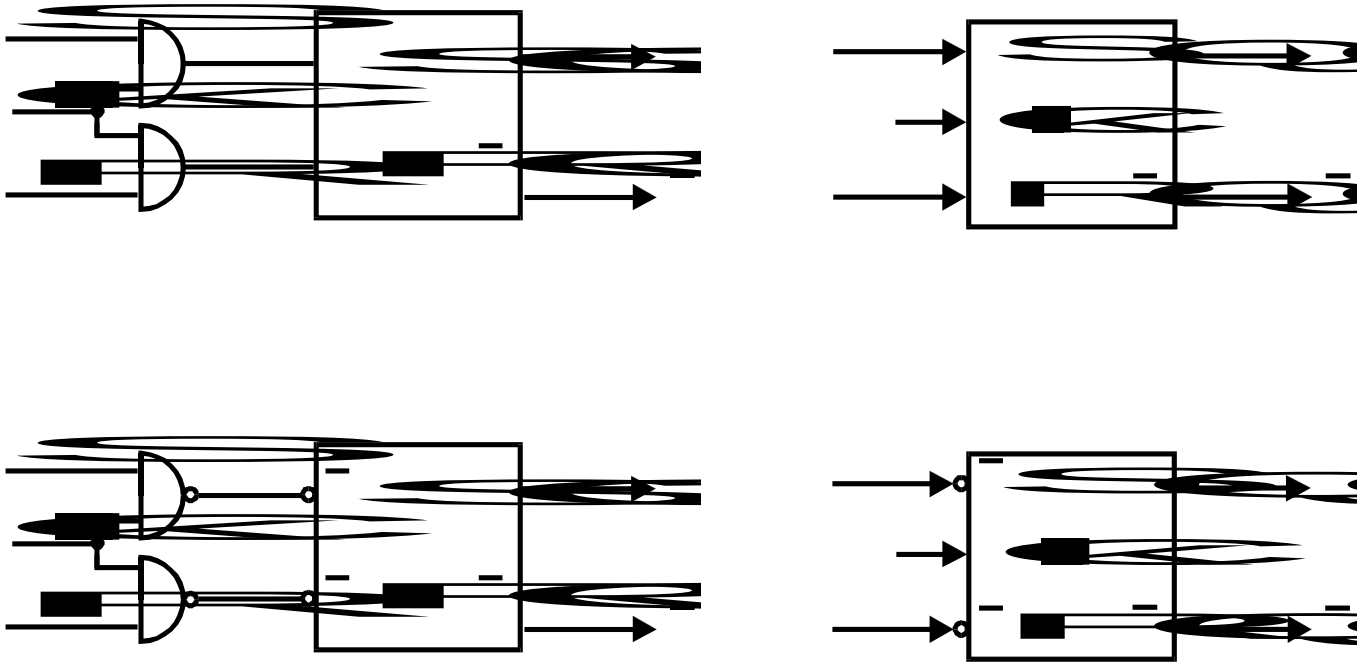


R=S=0: Q is not determined.
Not valid input

Latches and flip-flops (V)

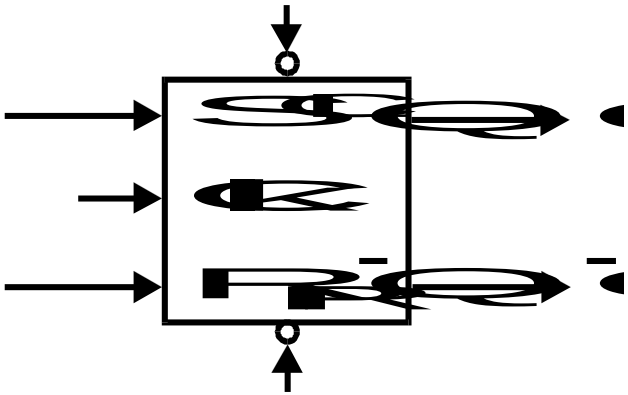
R-S synchronous

- Level-triggered



Latches and flip-flops (VI)

R-S synchronous with asynchronous inputs CL and PR

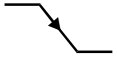


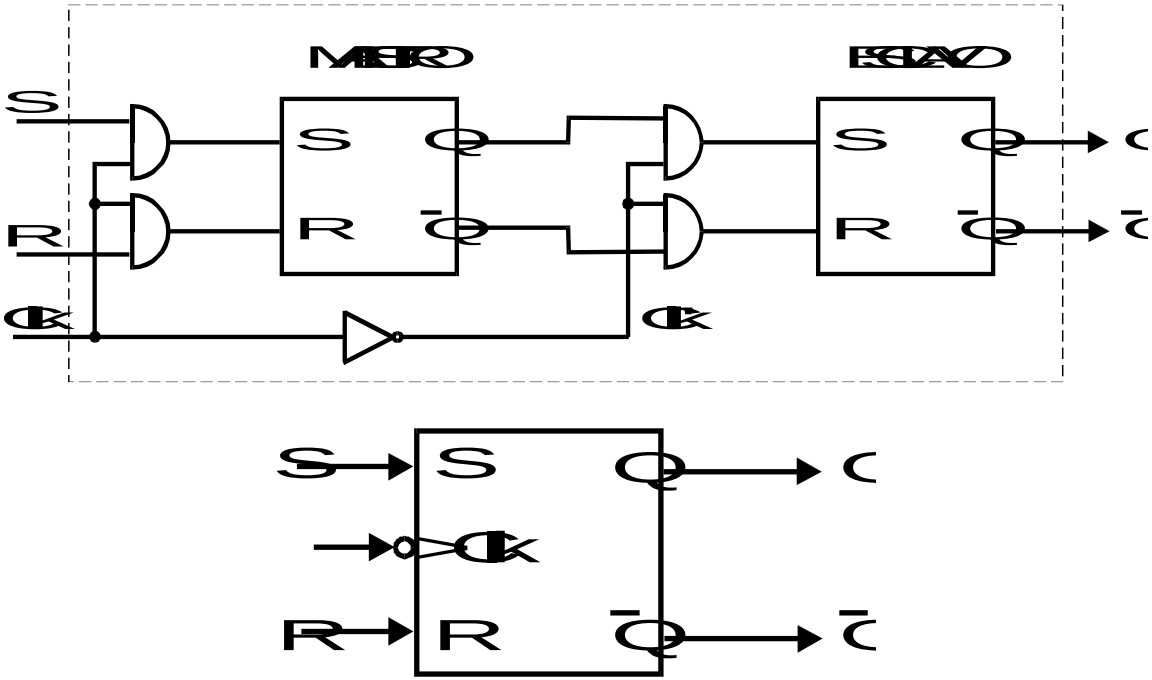
PR	CL	S	R	CK	Q^{t+1}
0	1	X	X	X	1
1	0	X	X	X	0
0	0	X	X	X	X
1	1	0	0	⊓	Q^t
1	1	1	0	⊓	1
1	1	0	1	⊓	0
1	1	1	1	⊓	X

Latches and flip-flops (VII)

R-S Master-Slave

It solves timing problems that can give rise to wrong outputs by reducing the switching moment of the flip-flop to transitions of the clock (rising or falling edges)

- Example: falling edge 

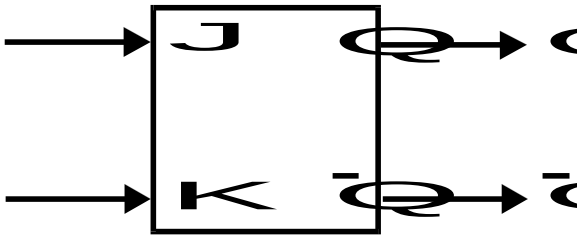


Latches and flip-flops (VIII)

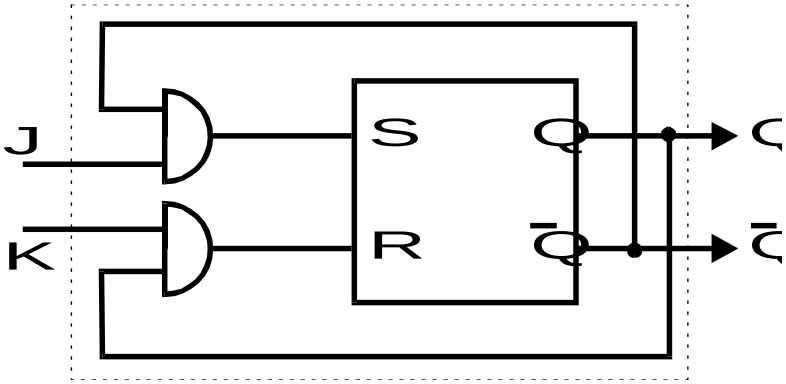
Asynchronous J-K Flip-flop

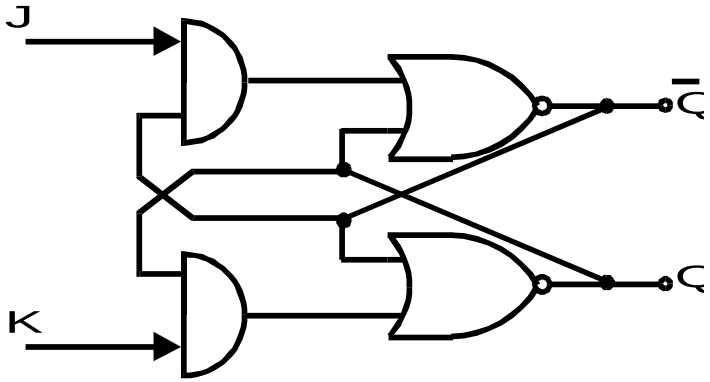
Like R-S, but removing undetermined situations using feedback

J ~ S y K ~ R.



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

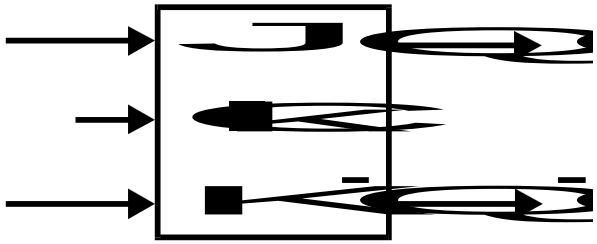


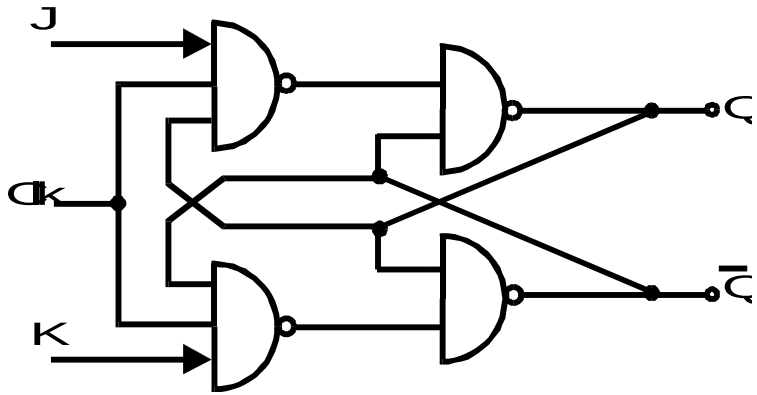


Latches and flip-flops (IX)

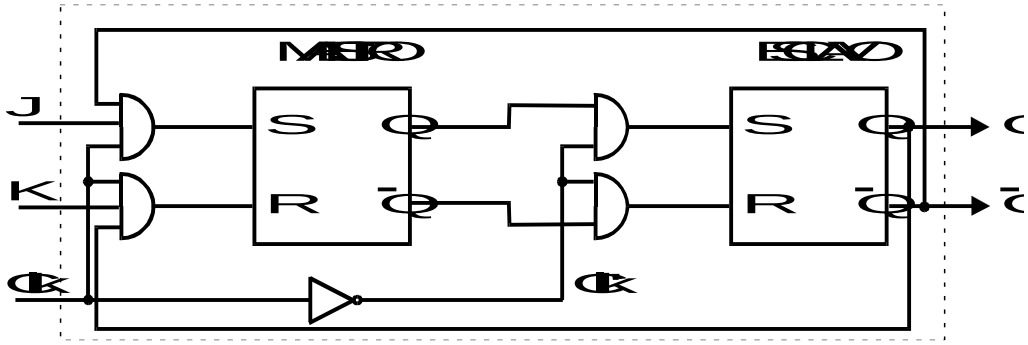
Synchronous J-K Flip-flop

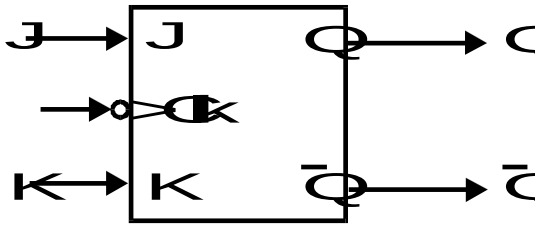
- Level-triggered





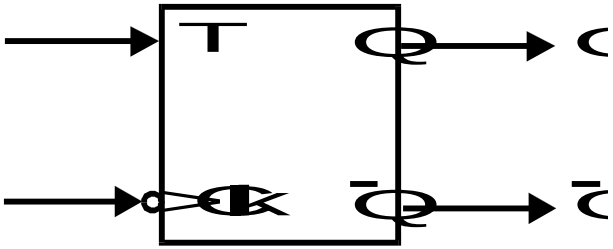
- Edge-triggered

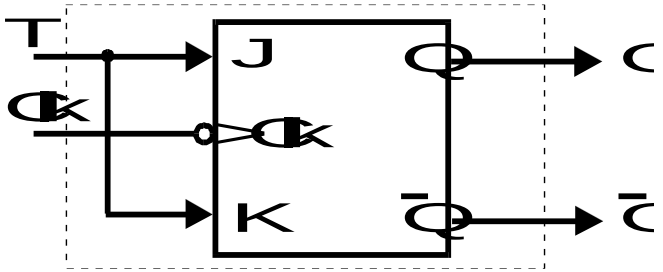




Latches and flip-flops (X)

T flip-flop

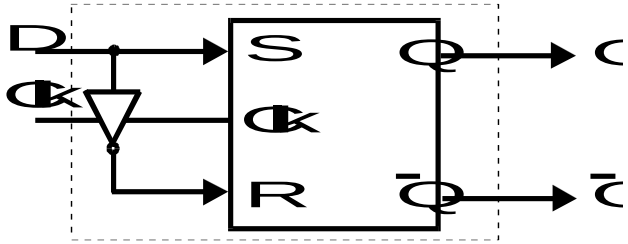




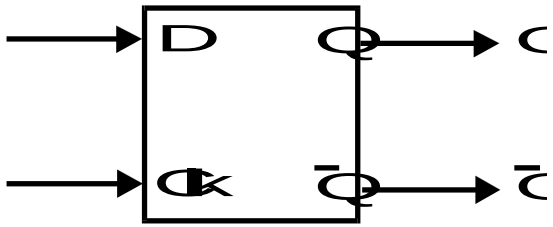
T	Clk	Q(t+1)
0	↓	Q(t)
1	↓	Q̄(t)

Latches and flip-flops (XI)

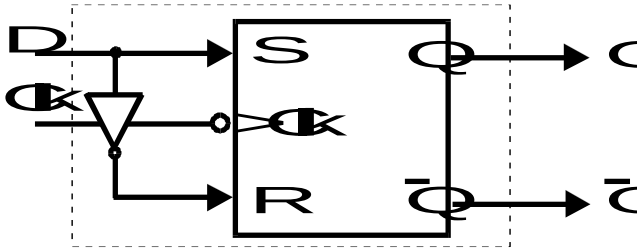
- **D latch** (level-triggered)



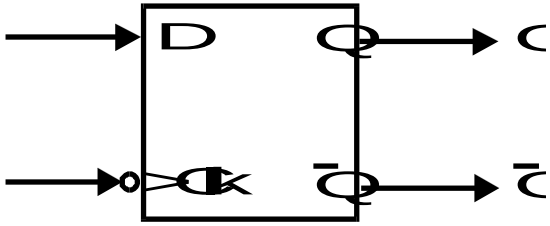
D	C	Q(t+1)
X	0	Q(t)
D	1	D



- **D flip-flop** (edge-triggered)



D	C	Q(t+1)
D	↘	D



Registers (I)

Register: Circuit that can store binary information, generally a *word* (*n* bits: 4, 8, 16, 32, 64…).

It is formed by flip-flops connected by different ways depending on the type:

Basic types:

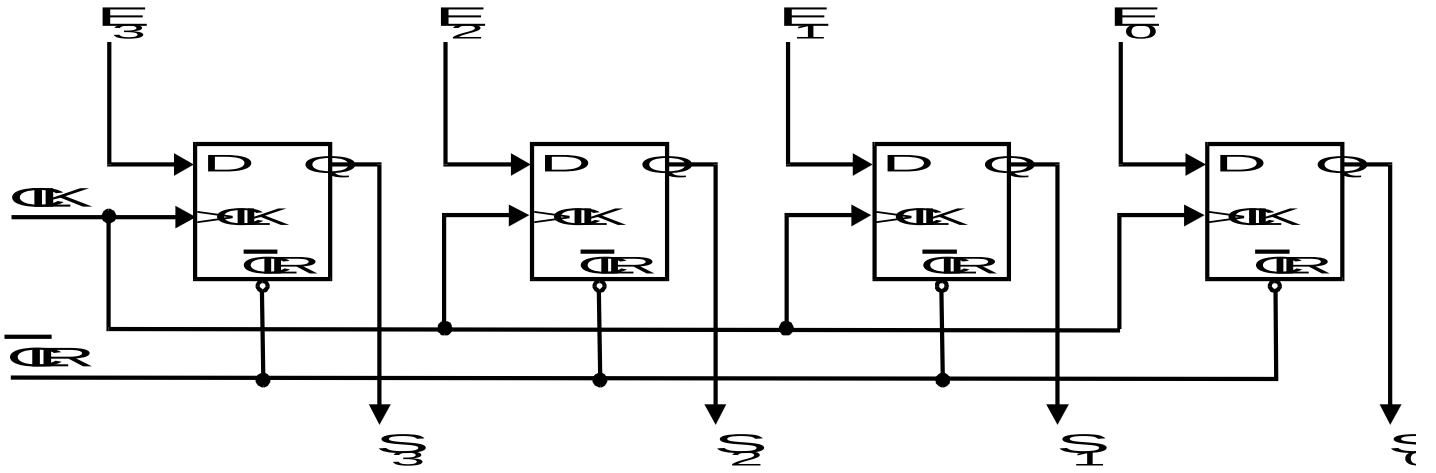
- Storing registers
- Shift registers
- Counters

Registers (II)

Storing Register: it works like a small memory; just stores bits.
Operations: read and write

Depending on the triggering:

- Latches
- Flip-flops

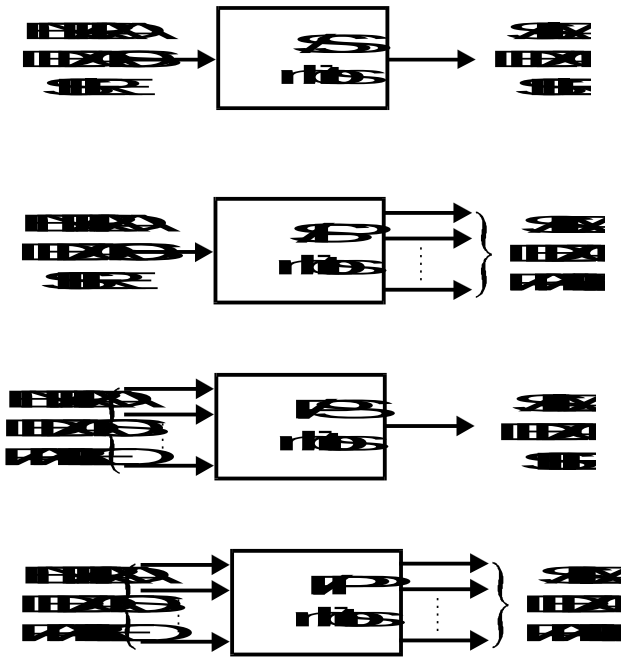


Registers (III)

Shift register. Besides storing information, it can shift it by moving bits between connected flip-flops or latches

Types (depending on input-output)

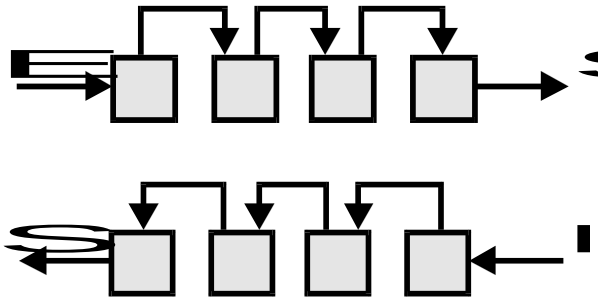
- Serial Input - Serial Output
- Serial Input - Parallel output
- Parallel Input - Serial Output
- Parallel Input - Parallel output



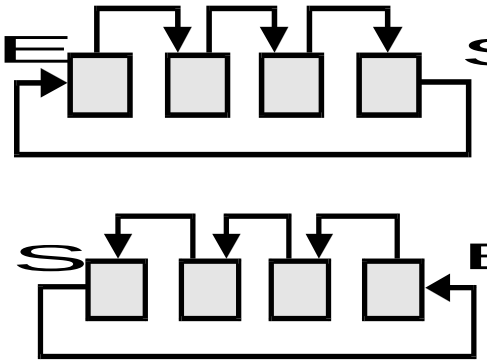
Registers (IV)

Types (depending on shifting):

- **Open**
  - Right shift
  - Left shift

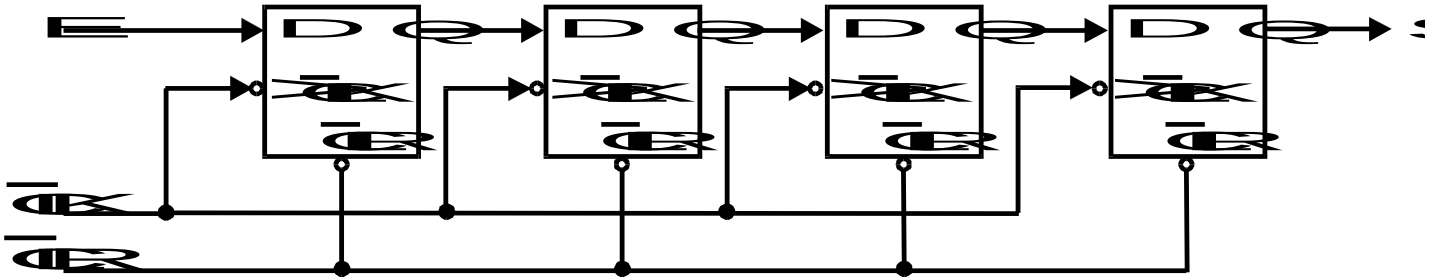


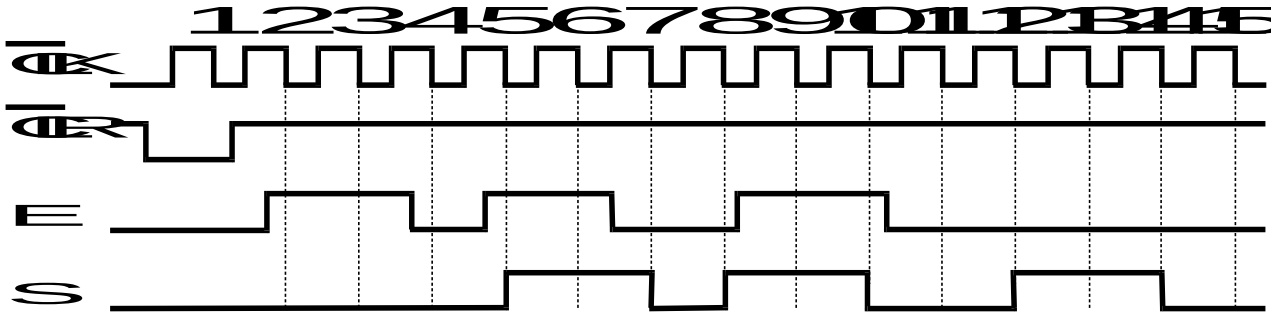
- **Ring**
  - Right shift
  - Left shift



Registers (V)

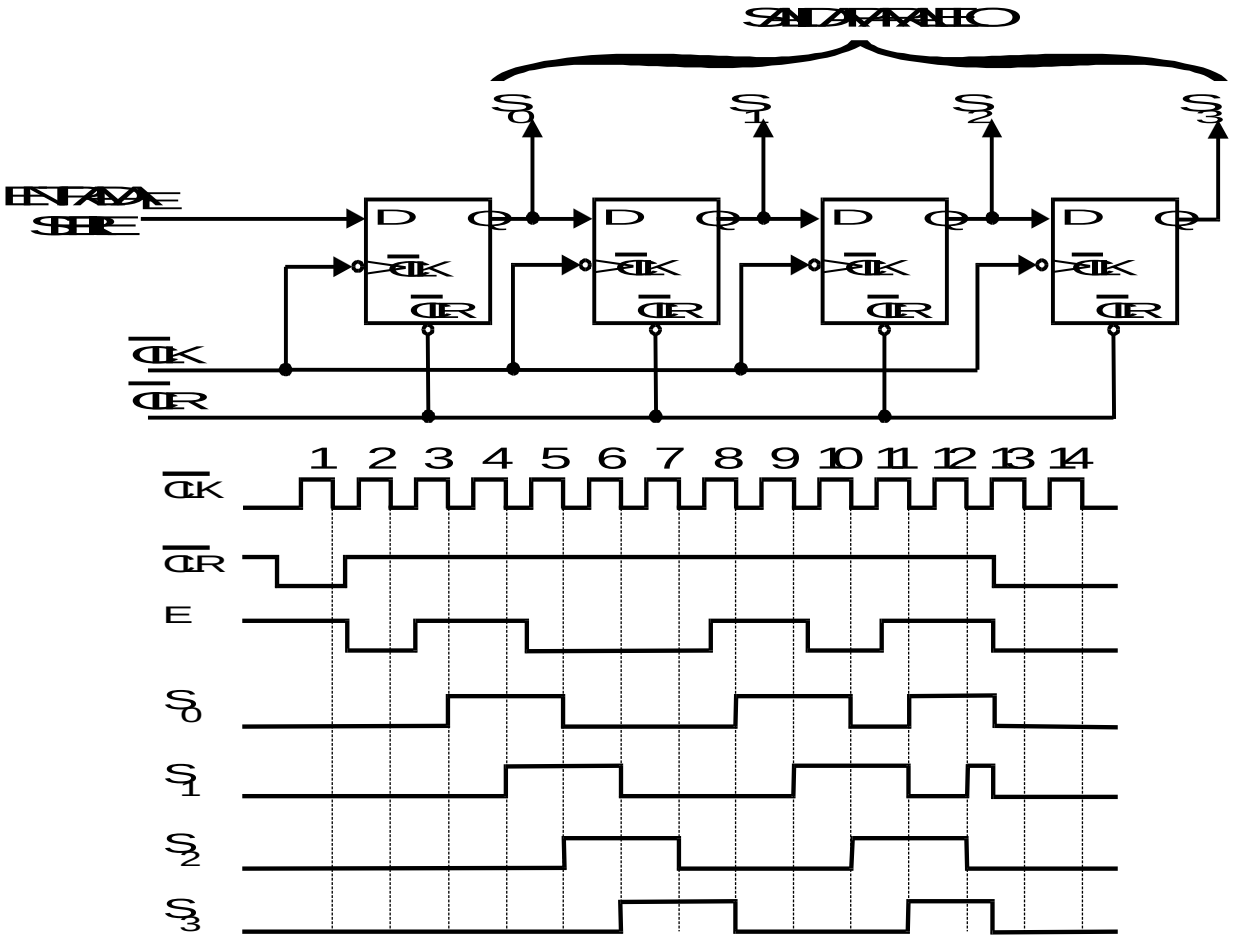
Shift register with serial input – serial output





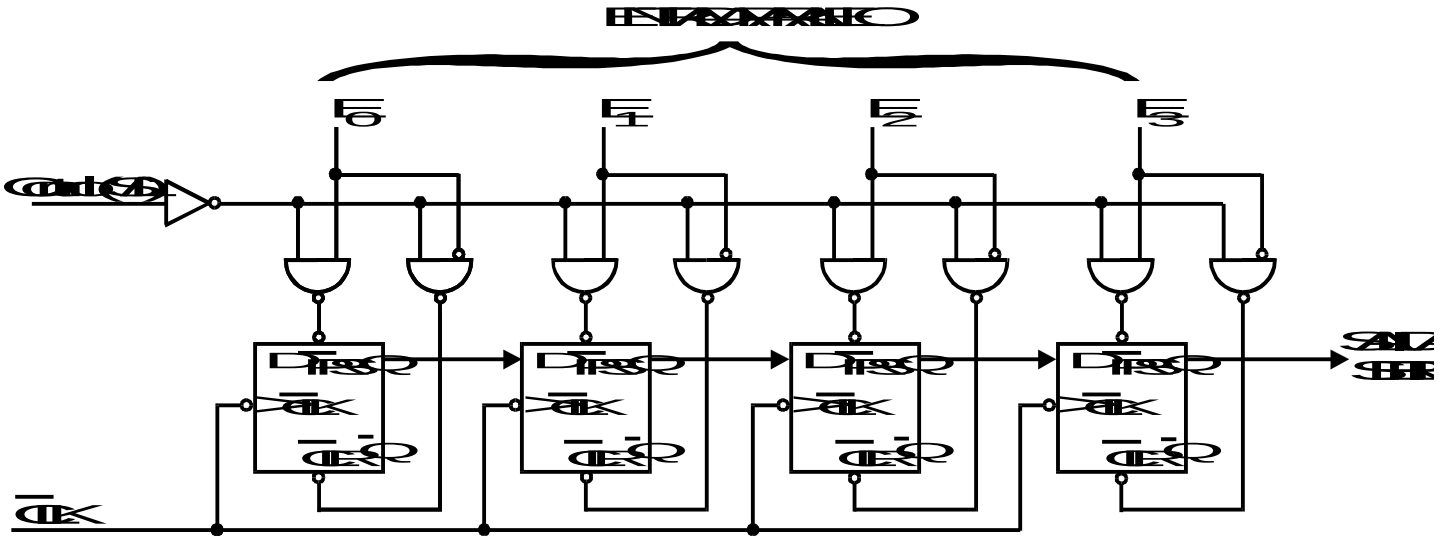
Registers (VI)

Shift register with serial input – parallel output



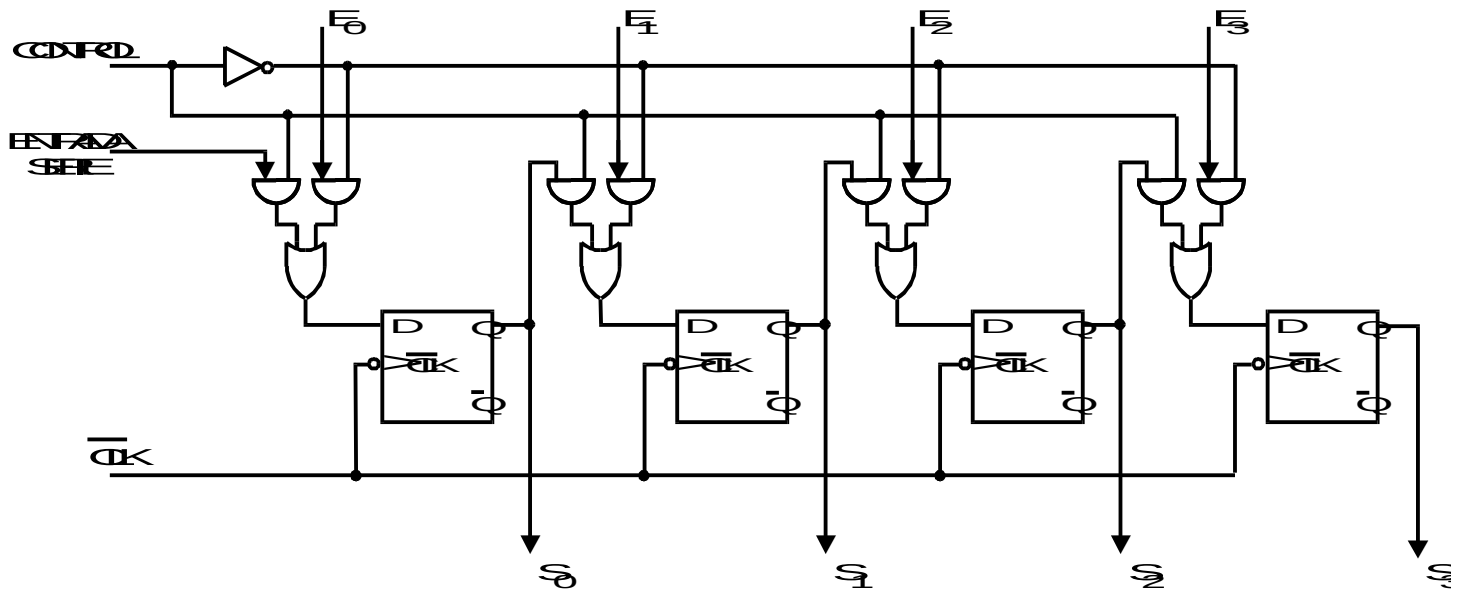
Registers (VII)

Shift register with parallel input – serial output



Registers (VIII)

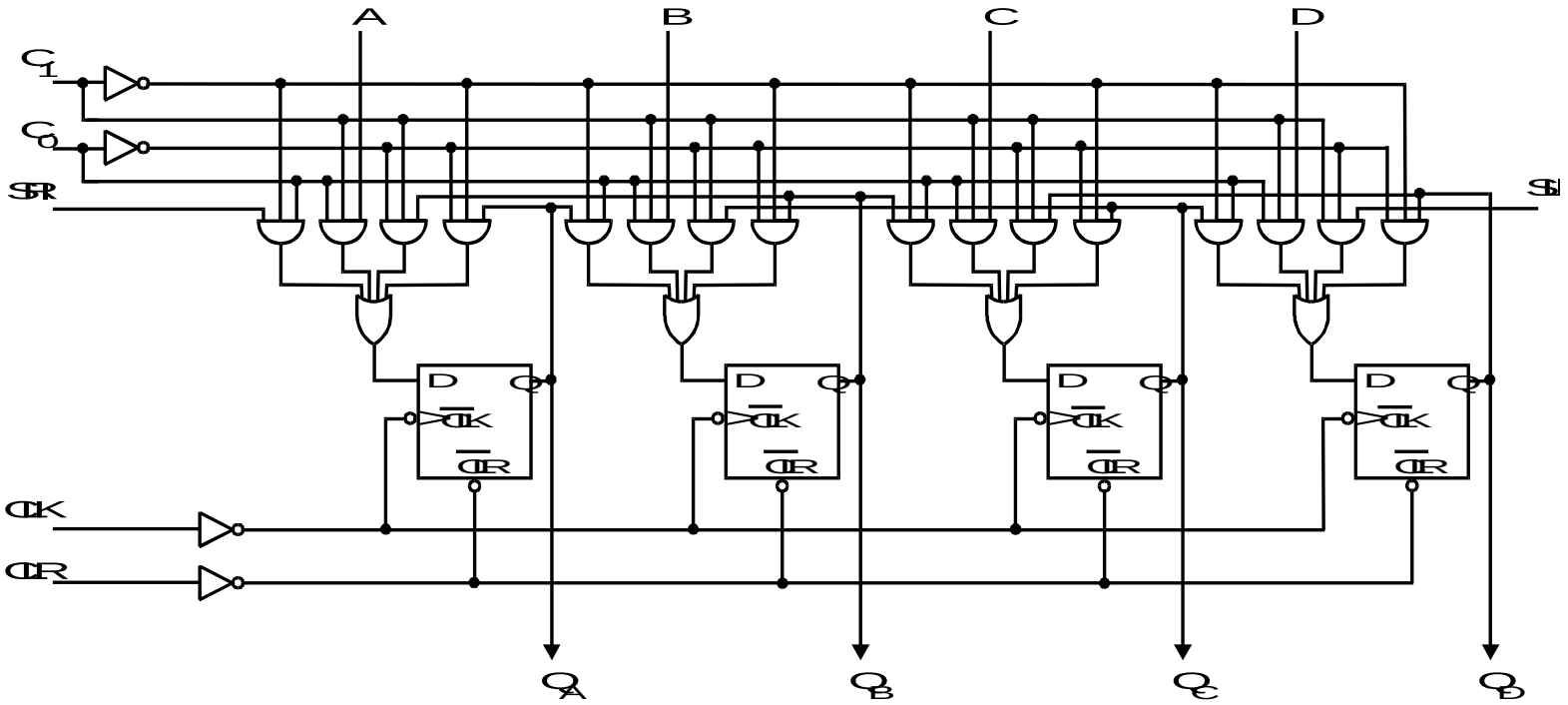
Shift register with parallel input - parallel output



Registers (IX)

Universal shift register

C_1	C_0	Operation
0	0	Keeps the state. No op
0	1	Right shift
1	0	Left shift
1	1	Load



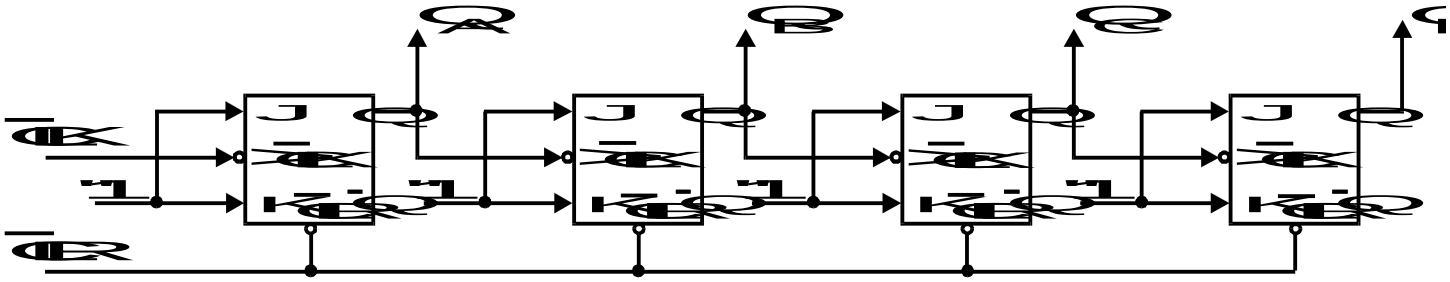
Counters (I)

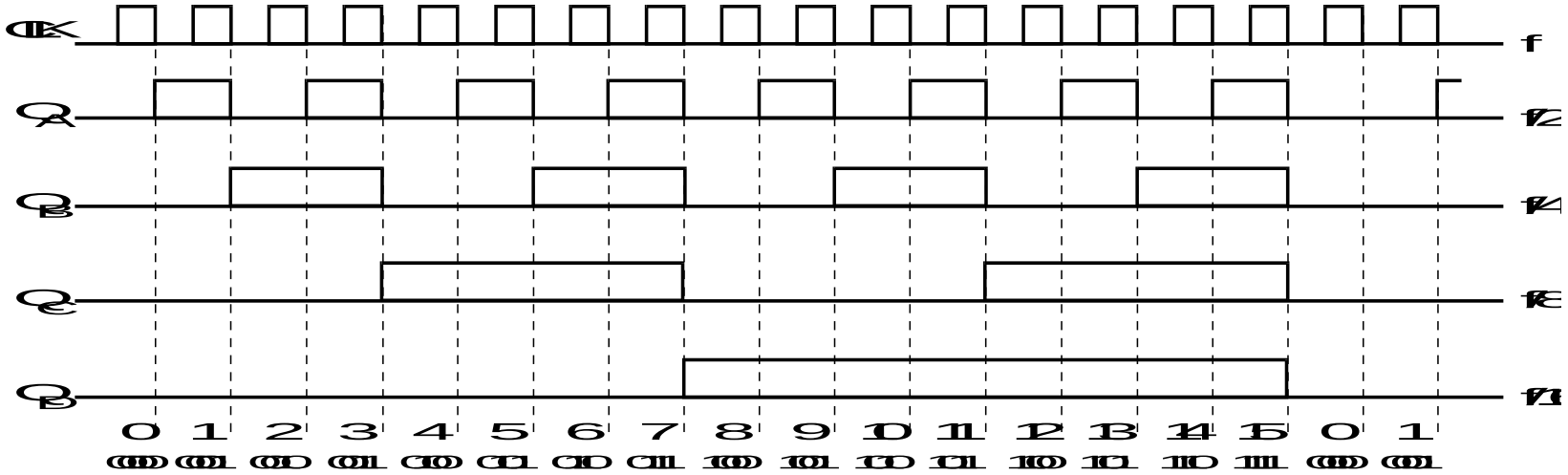
- Circuit that “counts” and “remembers” the number of pulses it receives from an external signal or clock
- It is formed by a chain of flip-flops whose *n* outputs represent the count in binary

- Classifications:
  - Synchronism:
    - Asynchronous / synchronous
  - Counting way:
    - Up / Down
  - Maximum count:
    - Binary / N-modulus

Counters (II)

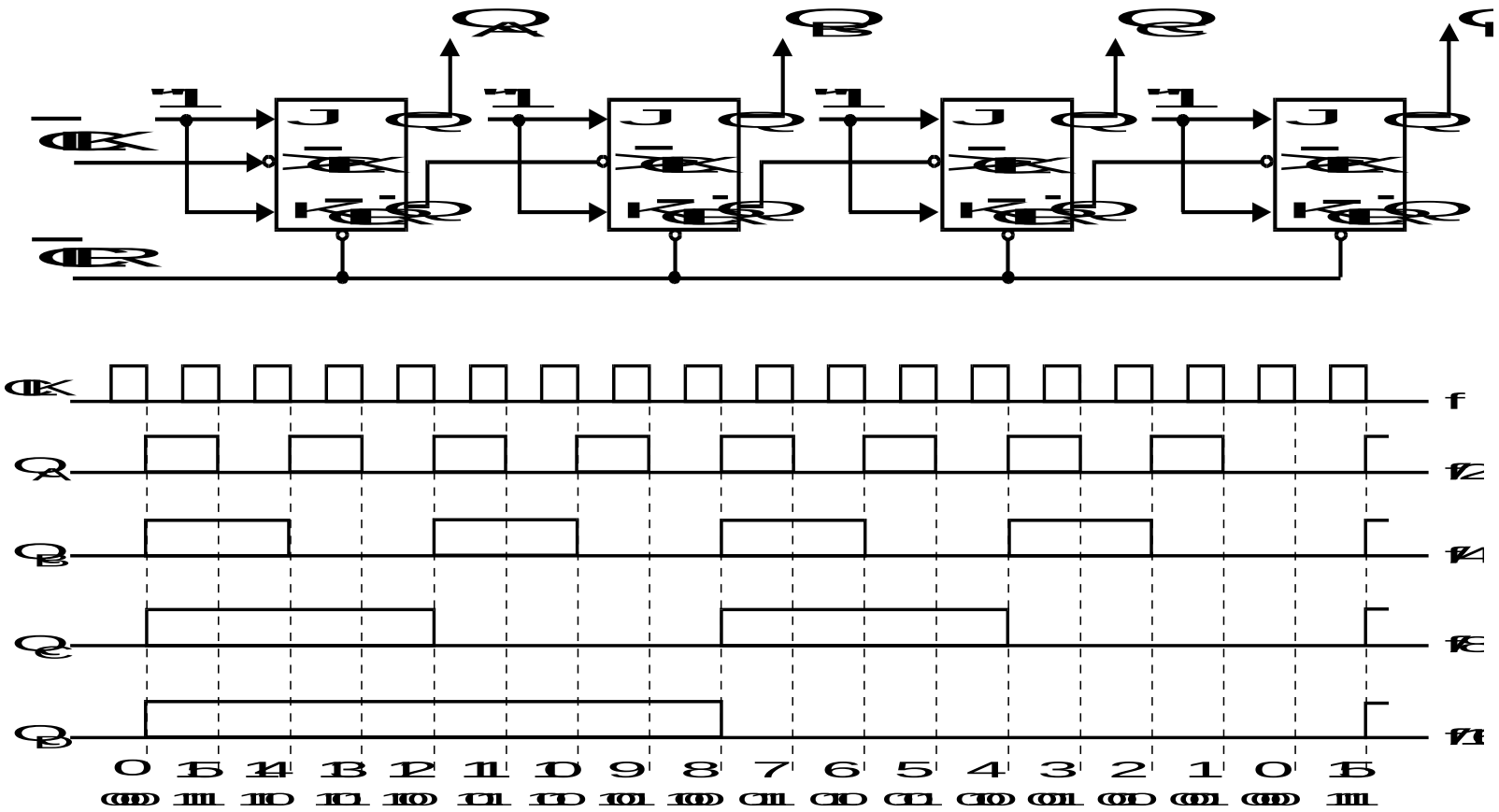
Asynchronous up binary counter





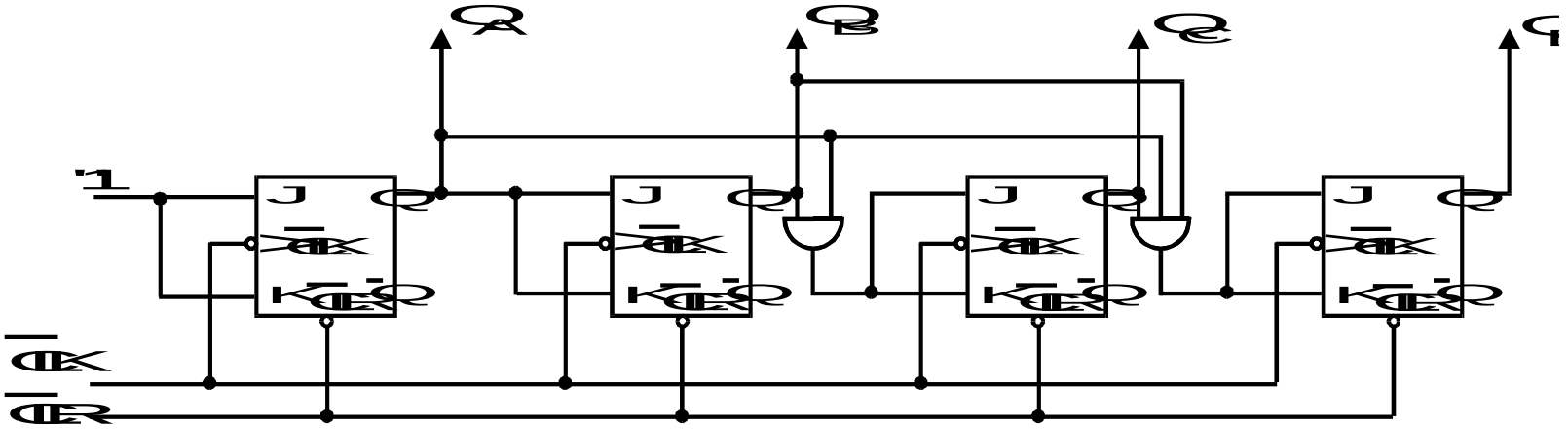
Counters (III)

Asynchronous down binary counter



Counters (IV)

Synchronous up binary counter



Counters (V)

10-modulus counter N (Up, Asynchronous)

Q_D	Q_C	Q_B	Q_A
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
1	0	0	1
1	**0**	**1**	**0**
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Puesta a cero del contador

Detección del 10