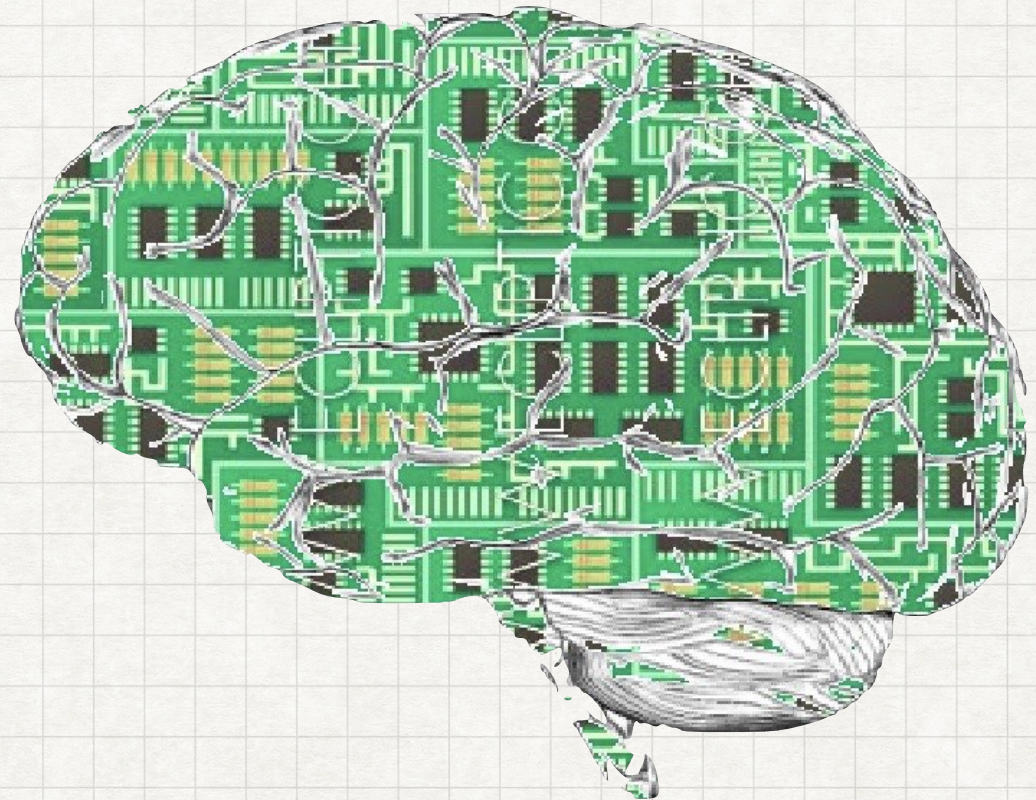


DEPARTAMENTO DE  
COMPUTACIÓN  
UNRC

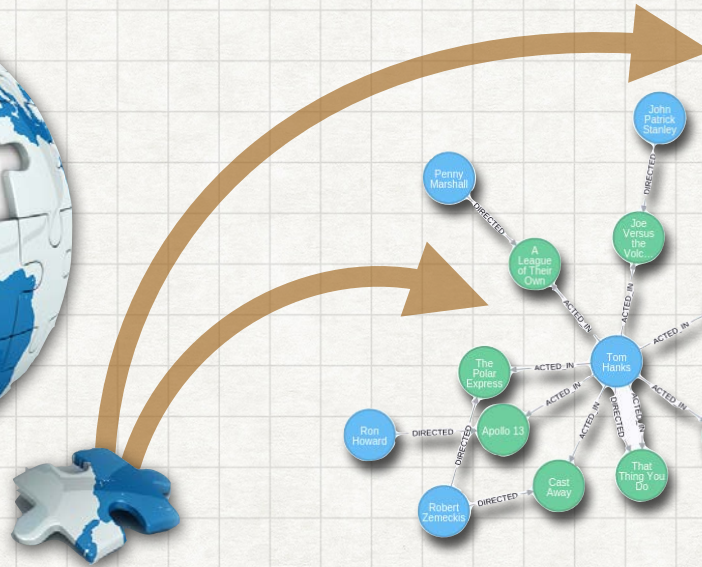
# ORGANIZACIÓN DEL PROCESADOR





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN Y MANIPULACIÓN DE INFORMACIÓN



```
public class TcpClientSample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input, stringData;
        TcpClient server;
        try{
            server = new TcpClient(" . . . . ", port);
        }catch (SocketException){
            Console.WriteLine("Unable to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recvd = ns.Read(data, 0, data.Length);
        stringData = Encoding.ASCII.GetString(data, 0, recvd);
        Console.WriteLine(stringData);
        while(true){
            input = Console.ReadLine();
            if (input == "exit") break;
            newchild.Properties["ou"].Add("Auditing Department");
            newchild.CommitChanges();
            newchild.Close();
        }
    }
}
```

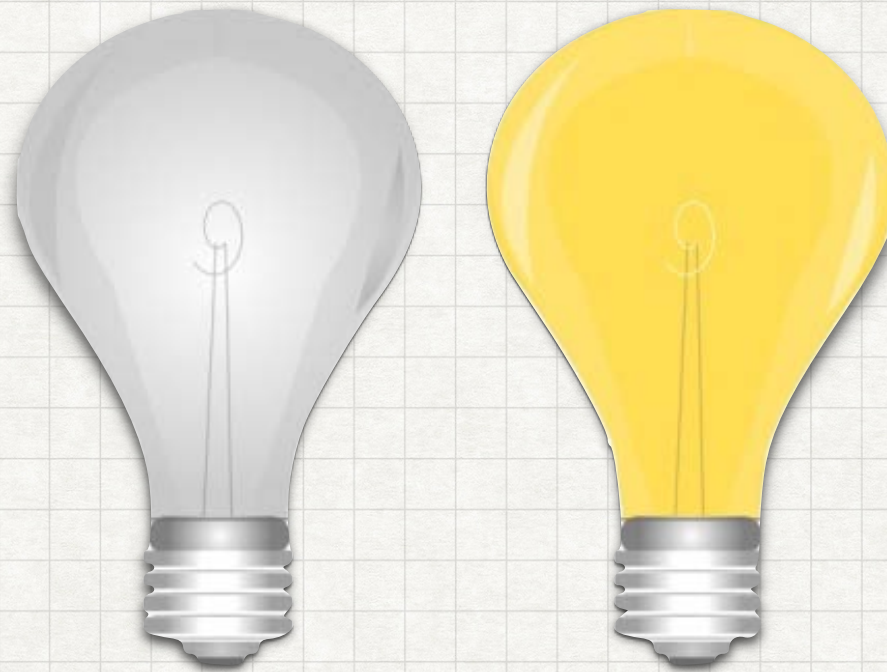




# ORGANIZACIÓN DEL PROCESADOR

## SISTEMAS DE NUMERACIÓN

Para representar la información utilizaremos el sistema binario (base 2)





# ORGANIZACIÓN DEL PROCESADOR

## SISTEMAS DE NUMERACIÓN

Para representar la información utilizaremos el sistema binario (base 2)



**0**  
**low**



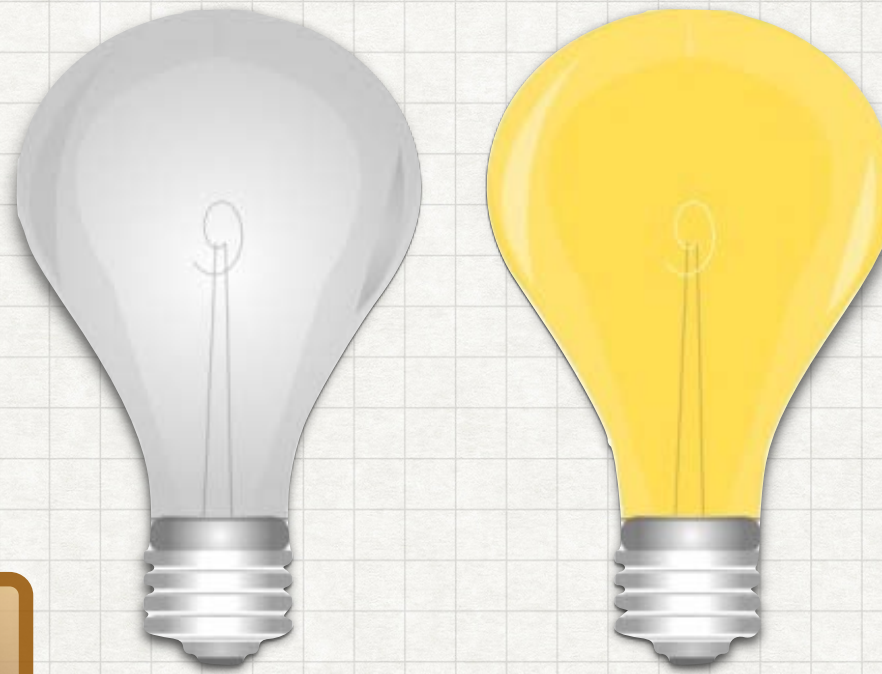
**1**  
**high**



# ORGANIZACIÓN DEL PROCESADOR

## SISTEMAS DE NUMERACIÓN

Para representar la información utilizaremos el sistema binario (base 2)



**BASE 2**

*(dos símbolos para  
identificar cosas)*

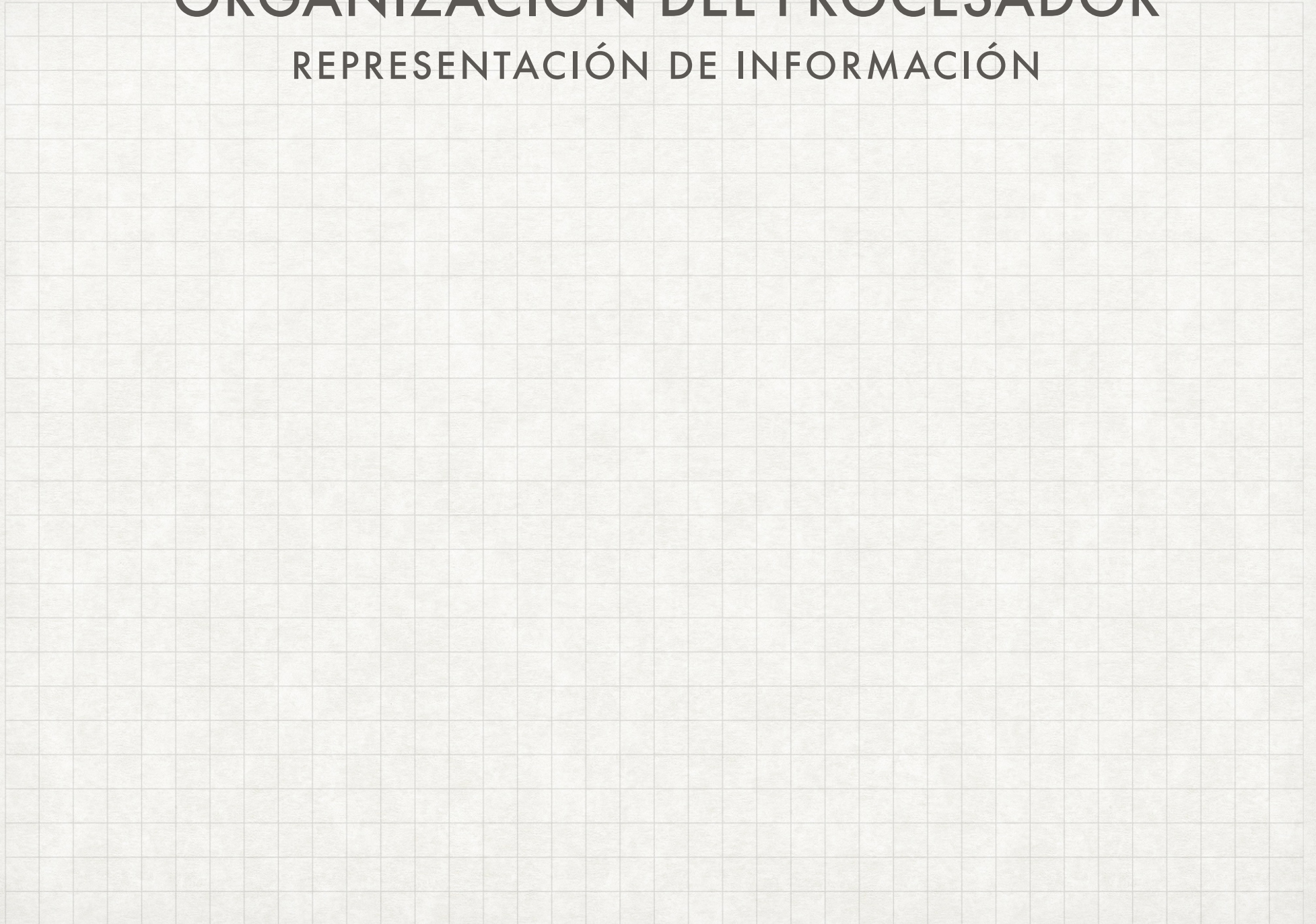
**0**  
**low**

**1**  
**high**



# ORGANIZACIÓN DEL PROCESADOR

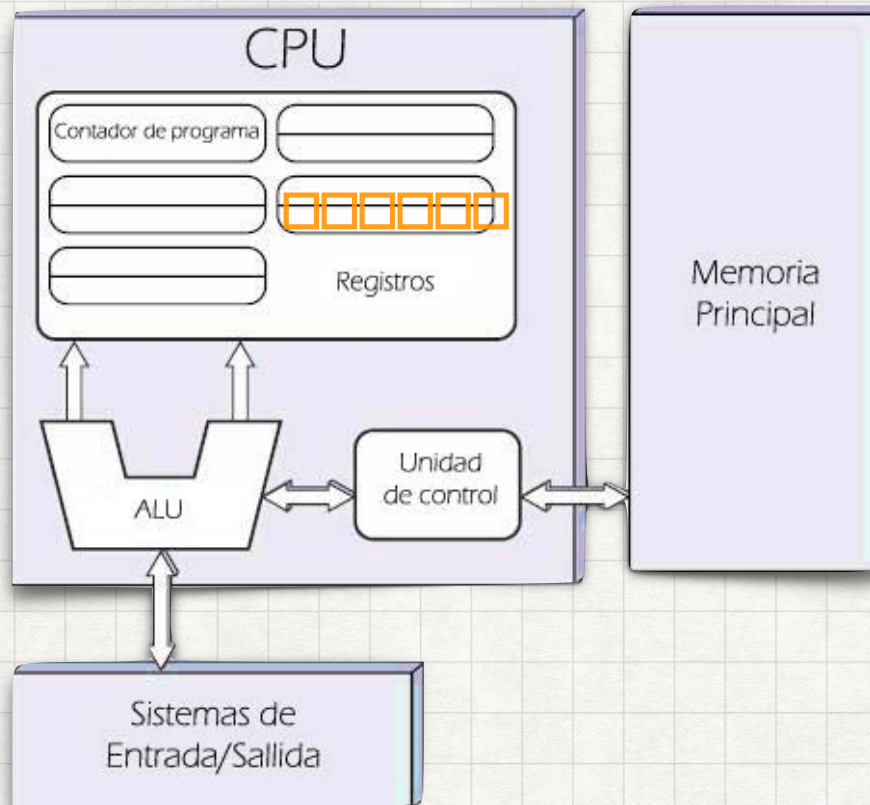
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

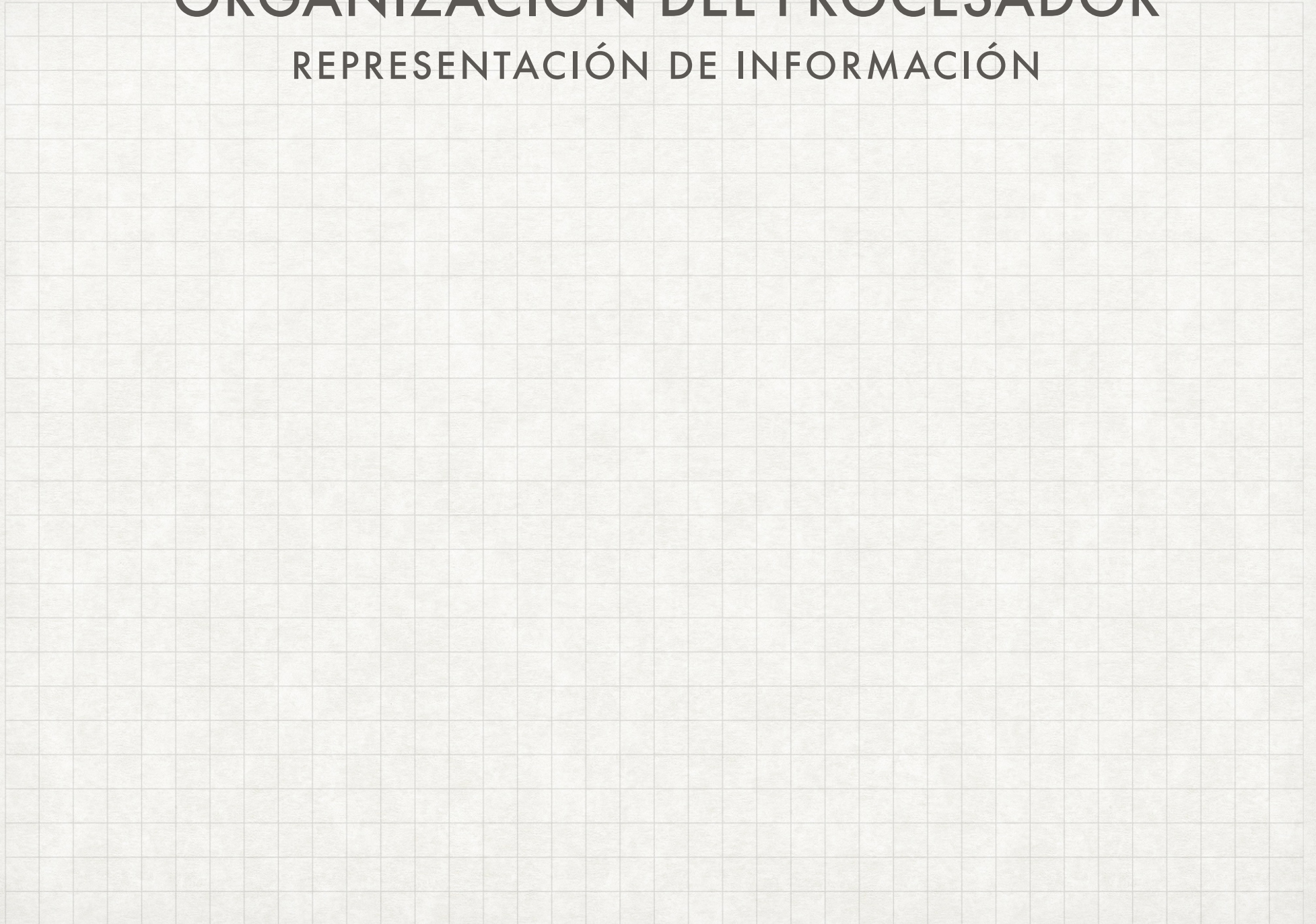
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

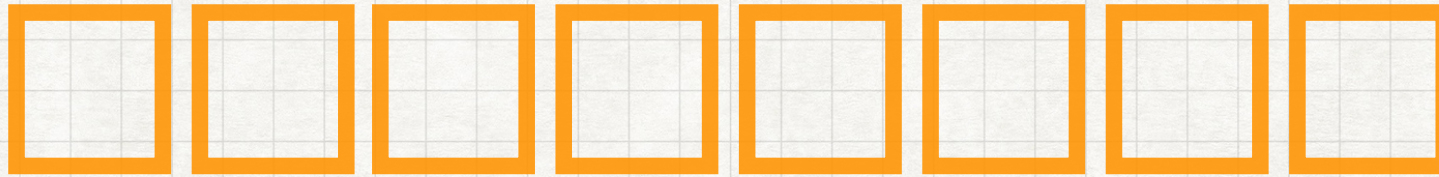
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

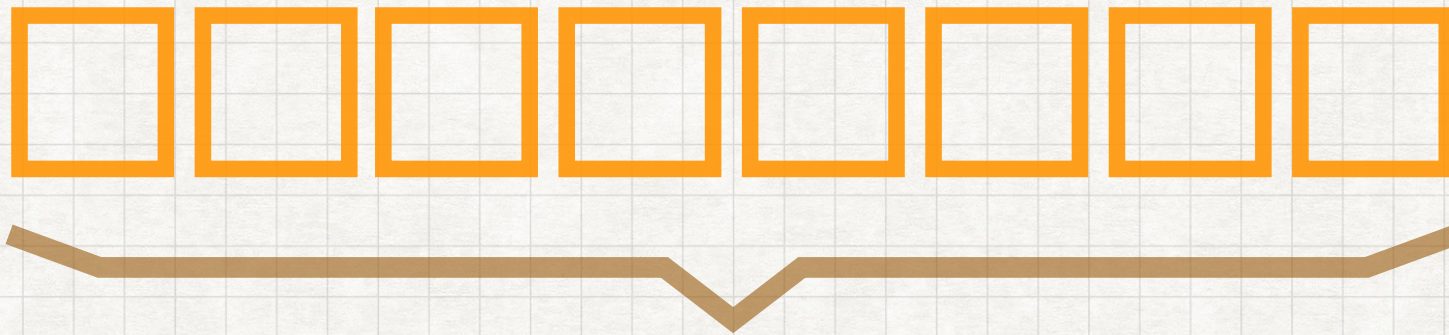
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

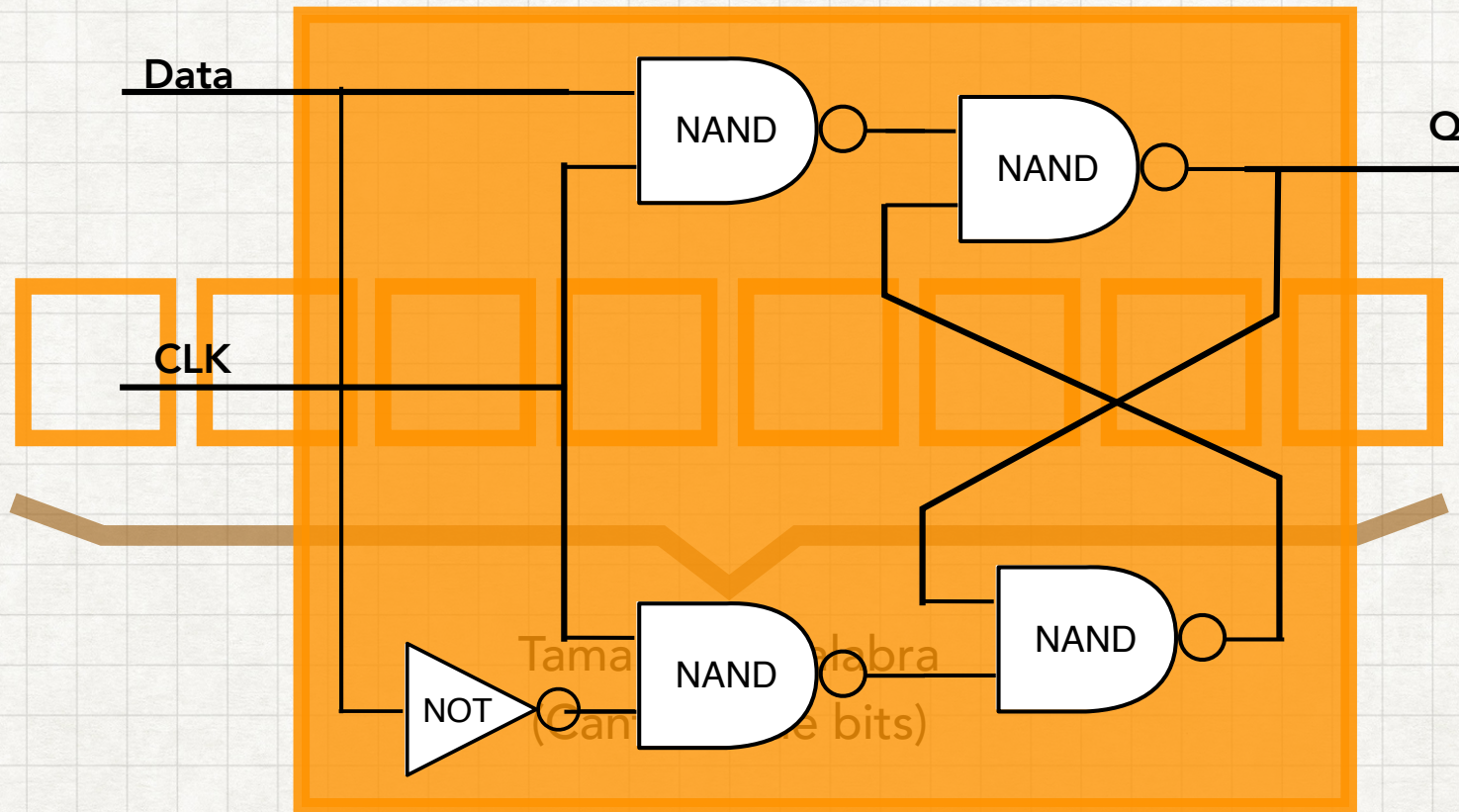


Tamaño de Palabra  
(Cantidad de bits)



# ORGANIZACIÓN DEL PROCESADOR

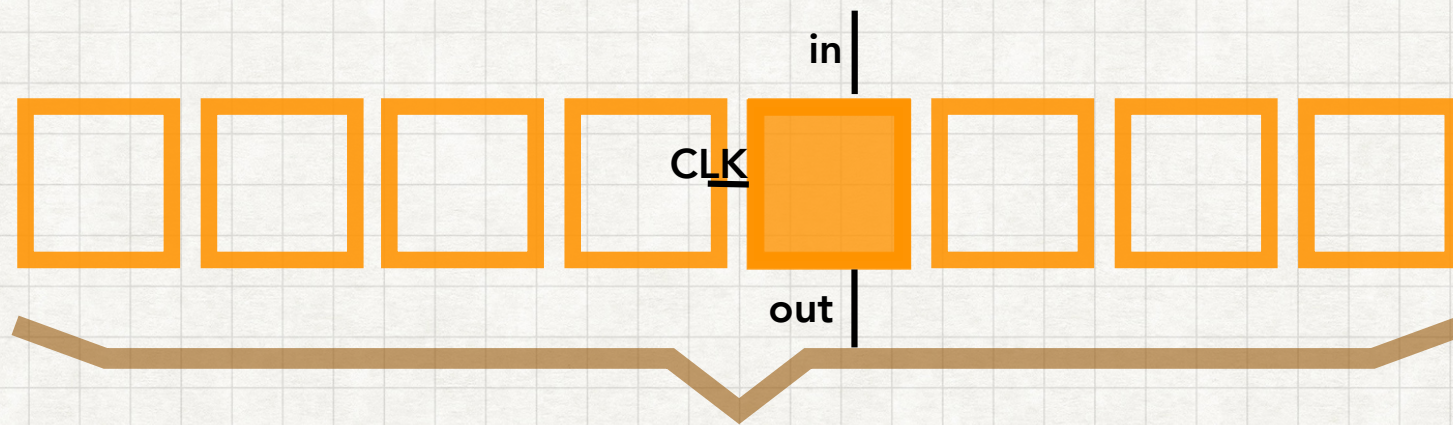
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

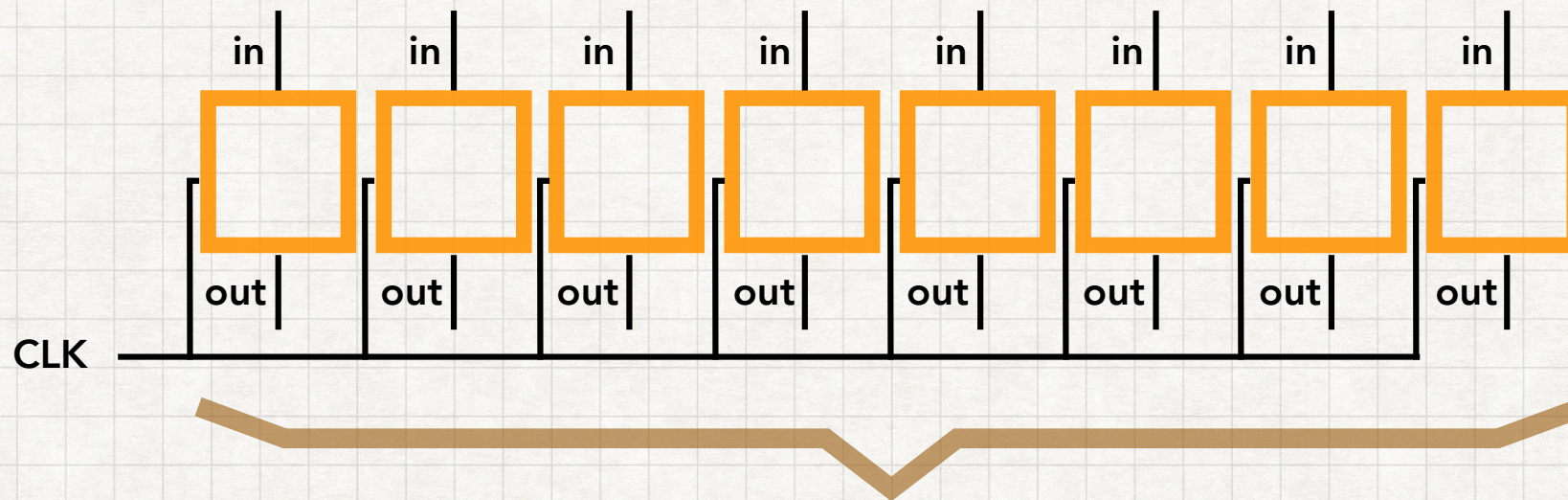


Tamaño de Palabra  
(Cantidad de bits)



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN



Tamaño de Palabra  
(Cantidad de bits)

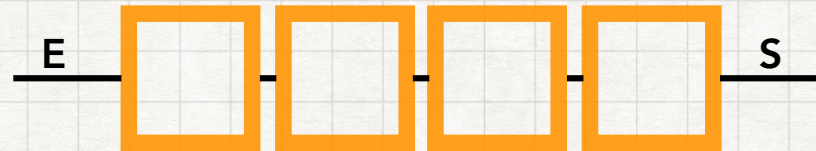


# ORGANIZACIÓN DEL PROCESADOR

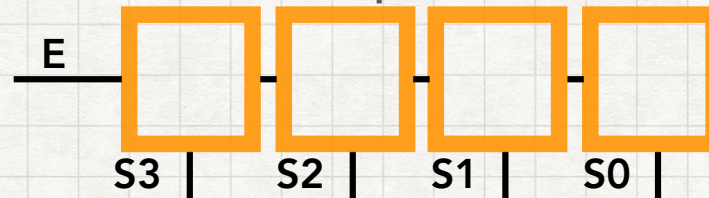
## REPRESENTACIÓN DE INFORMACIÓN

Existen diferentes tipos de registros según su forma de insertar y obtener el valor de su contenido:

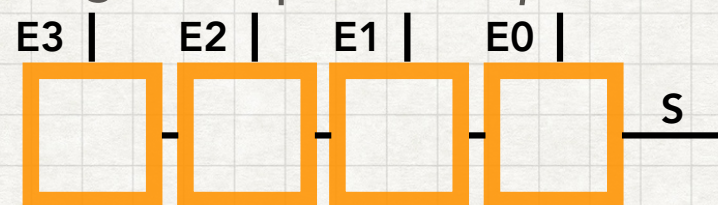
**SISO:** Single Input Single Output (*una entrada una salida*)



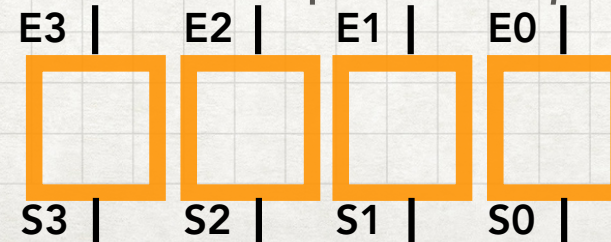
**SIPO:** Single Input Parallel Output (*una entrada múltiples salidas*)



**PISO:** Parallel Input Single Output (*múltiples entradas una salida*)



**PIPO:** Parallel Input Parallel Output (*múltiples entradas múltiples salida*)





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

CLOCK

CLOCK

*tiempo*

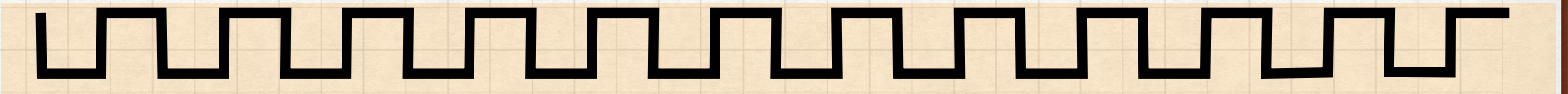


# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

CLOCK

CLOCK



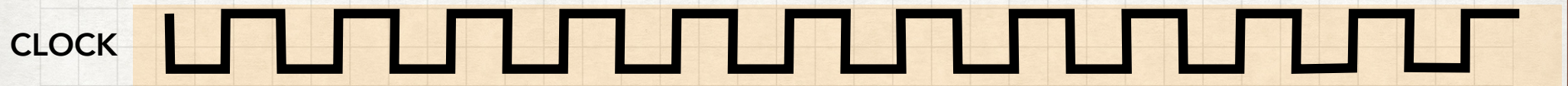
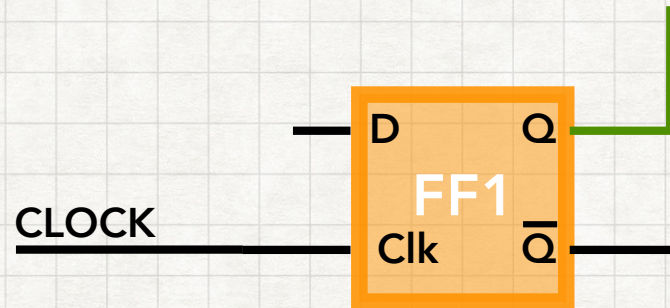
tiempo





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN



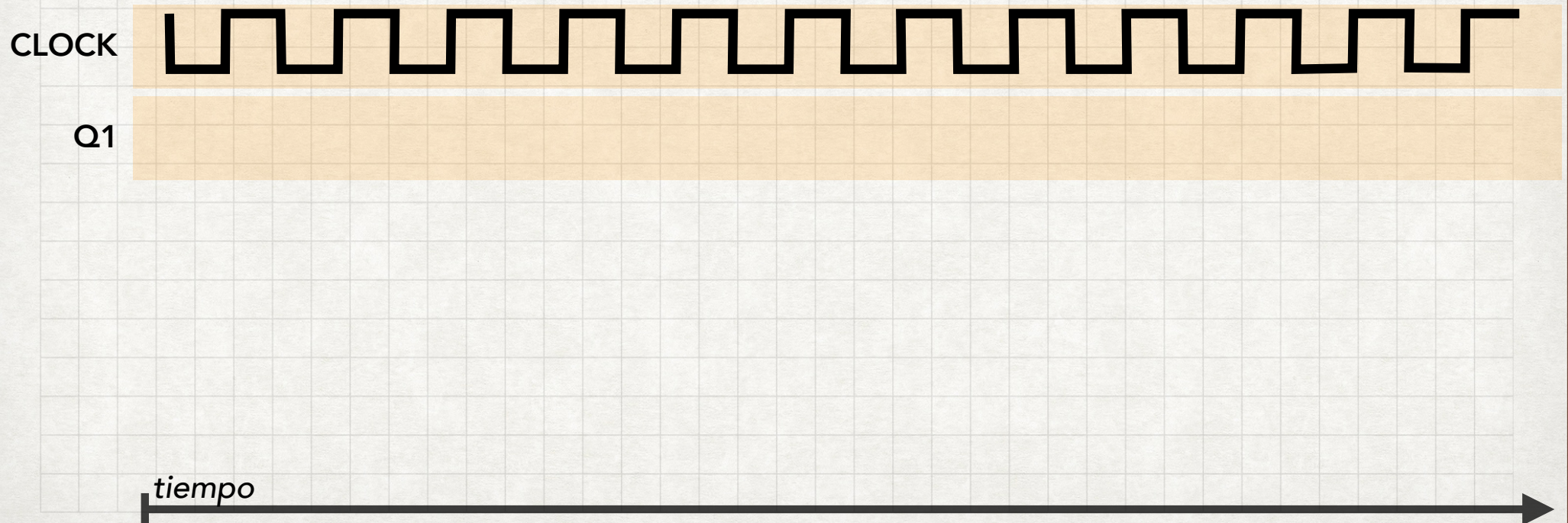
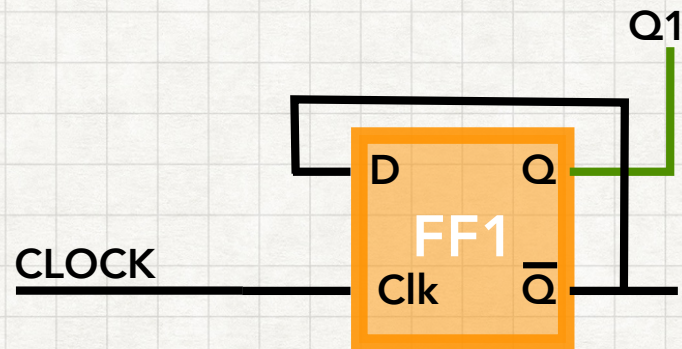
tiempo





# ORGANIZACIÓN DEL PROCESADOR

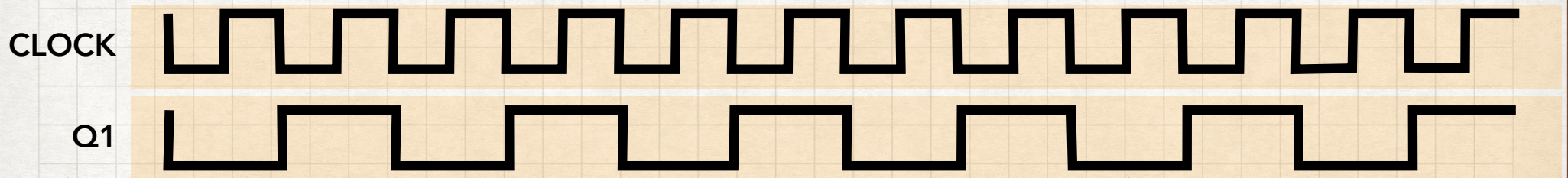
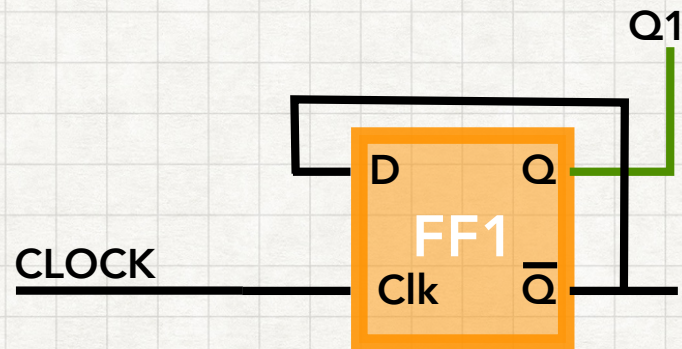
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

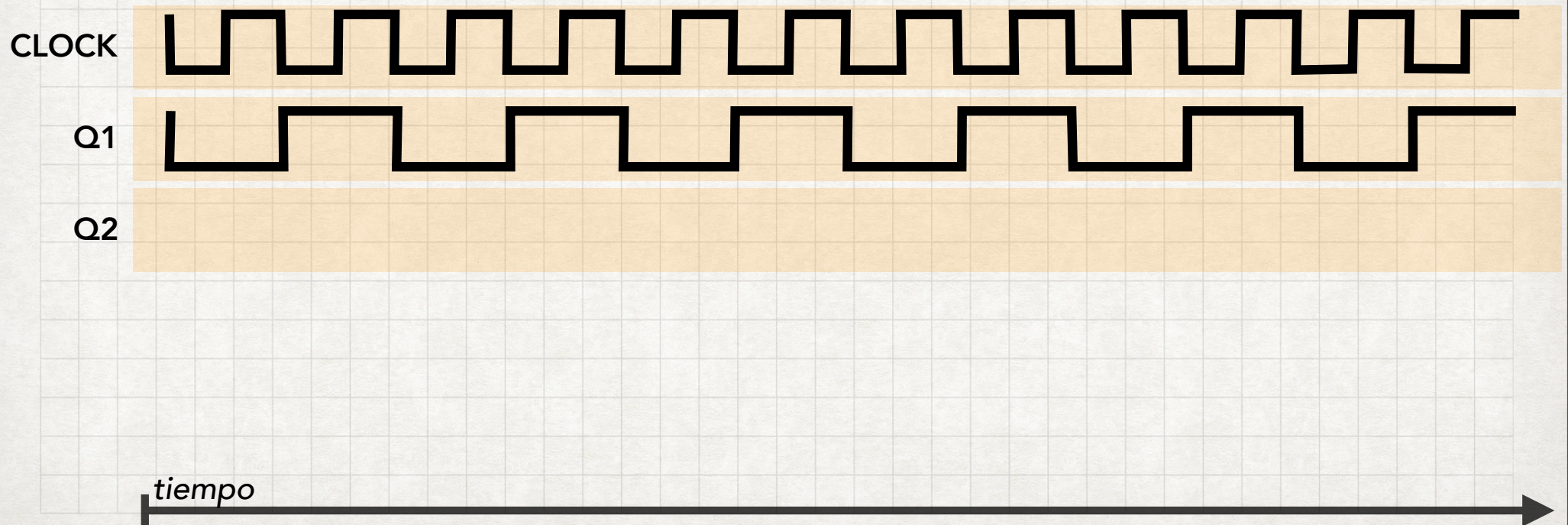
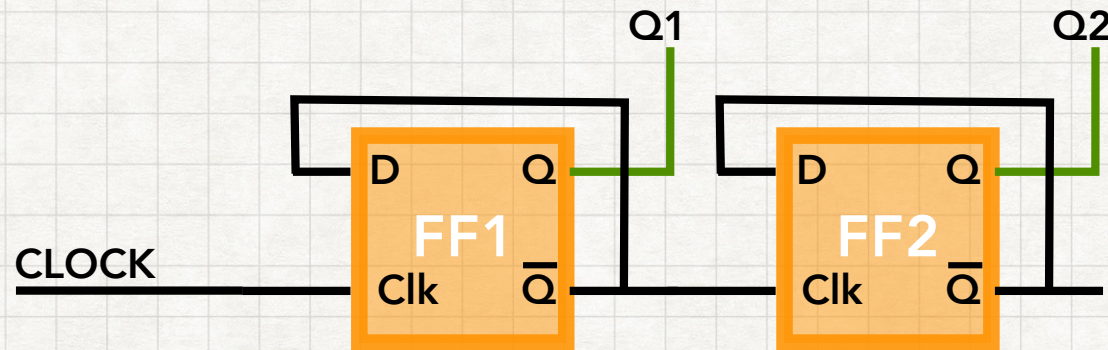


tiempo



# ORGANIZACIÓN DEL PROCESADOR

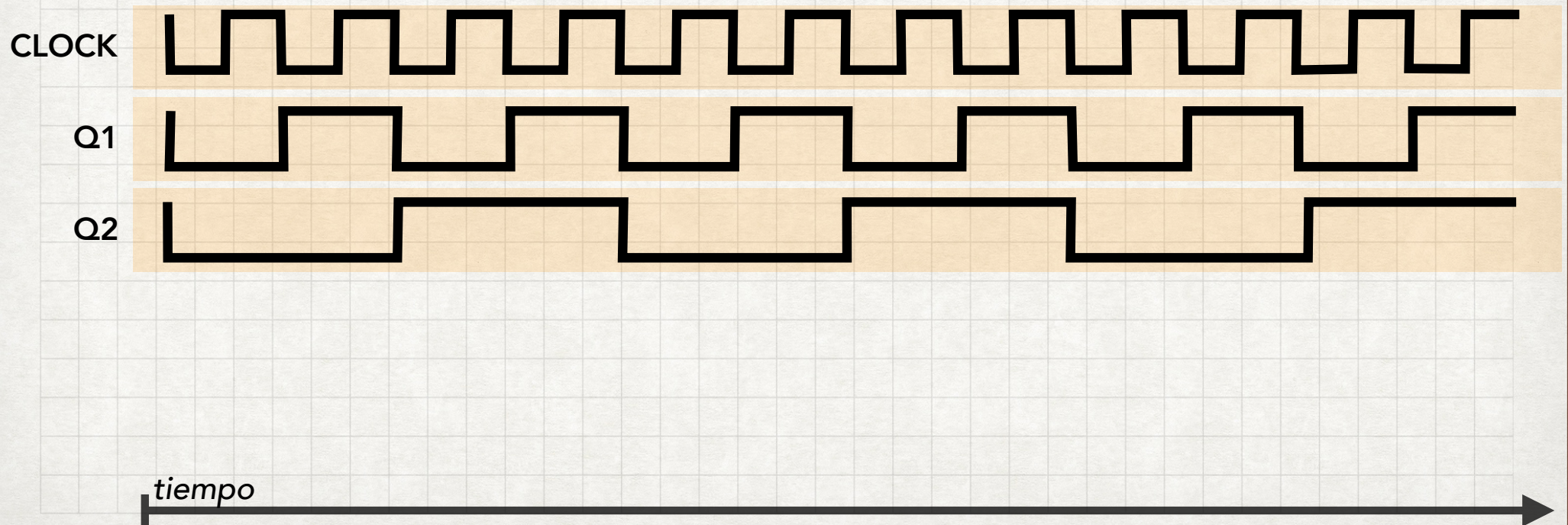
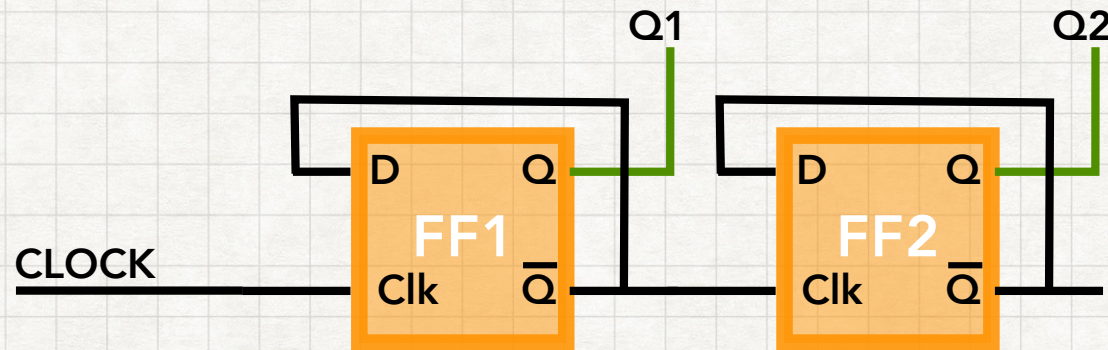
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

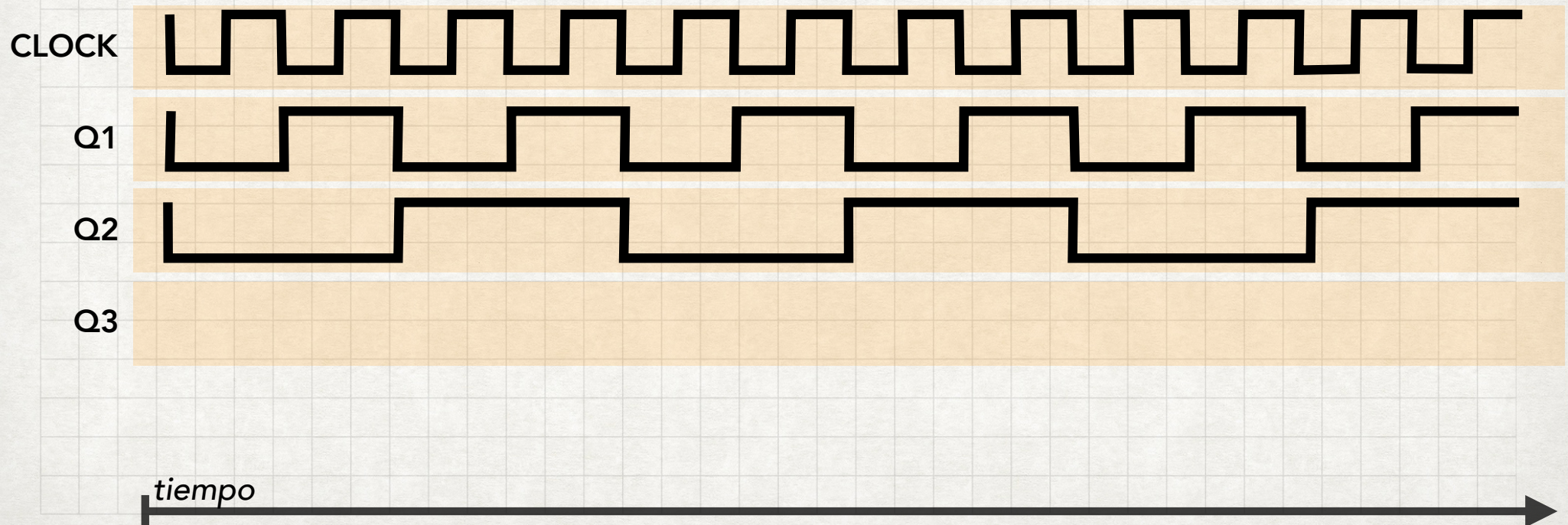
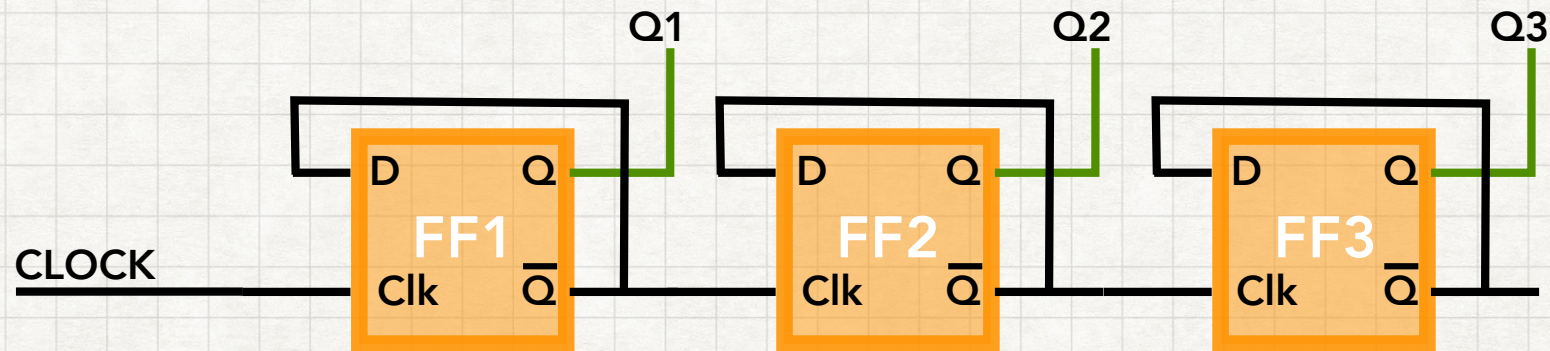
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

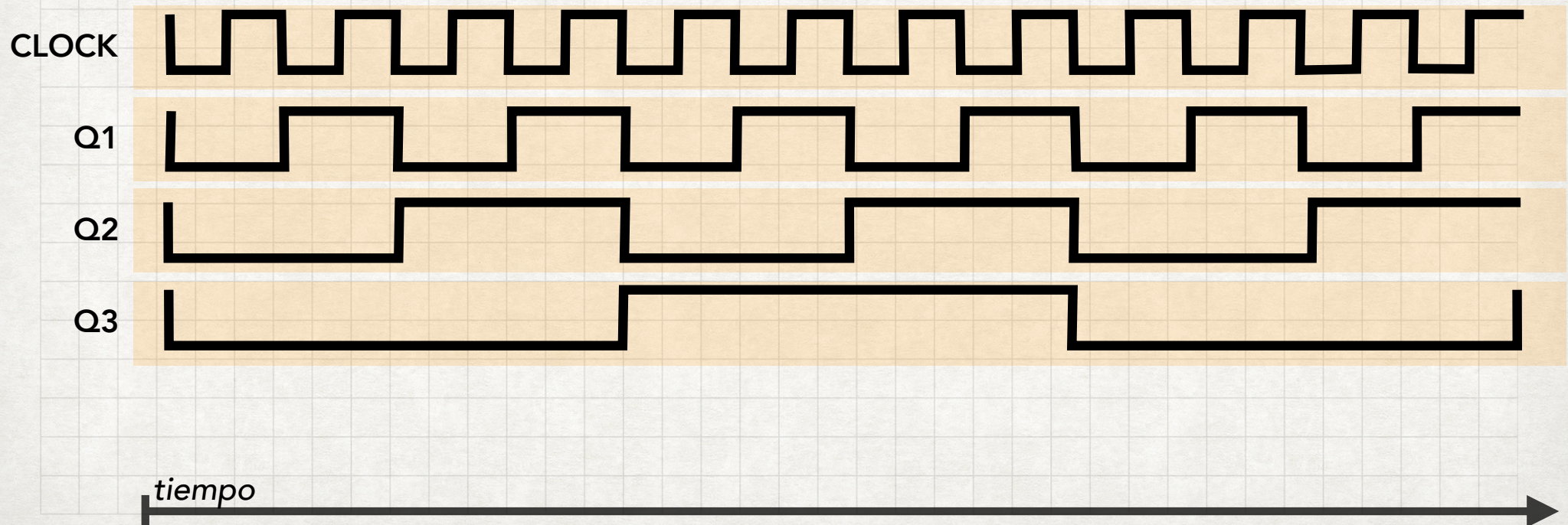
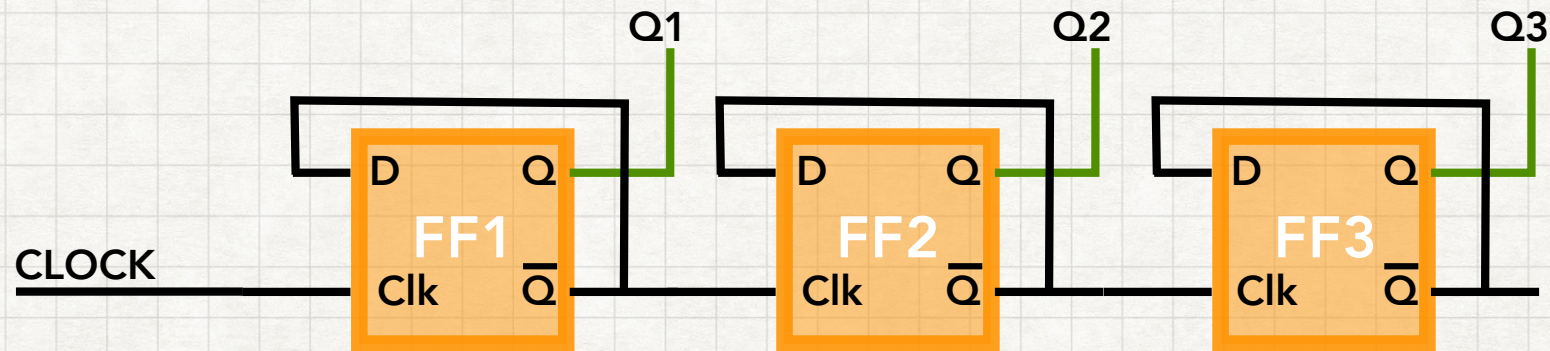
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

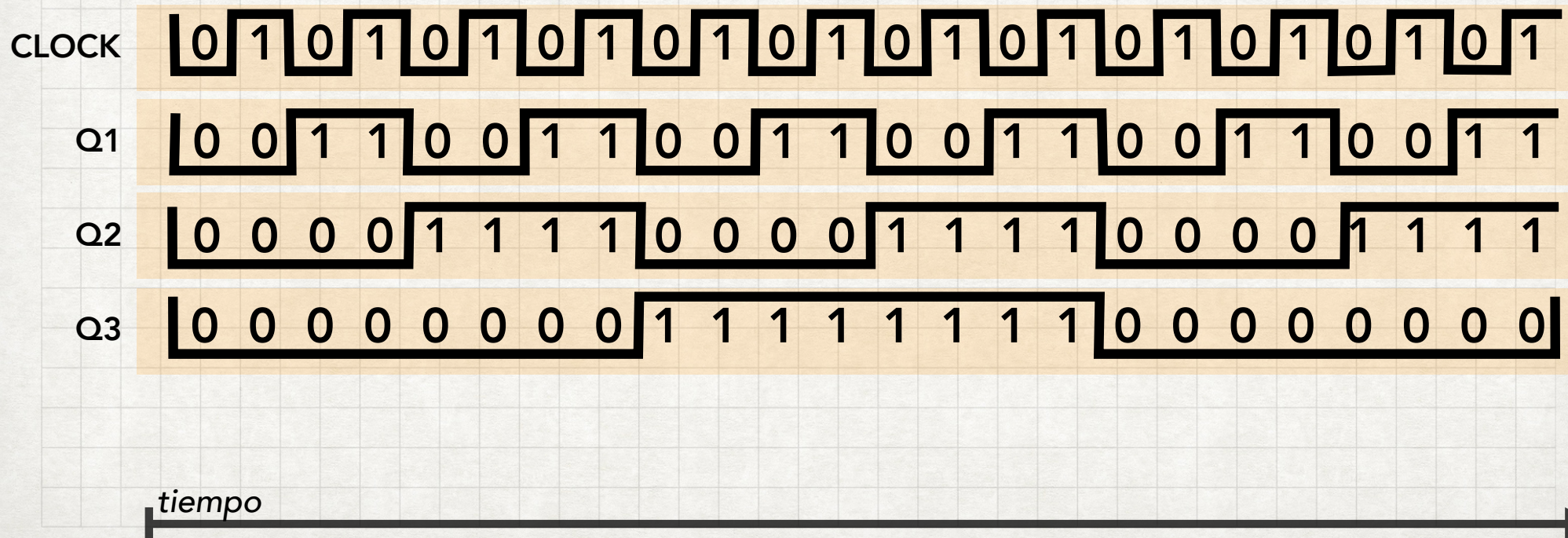
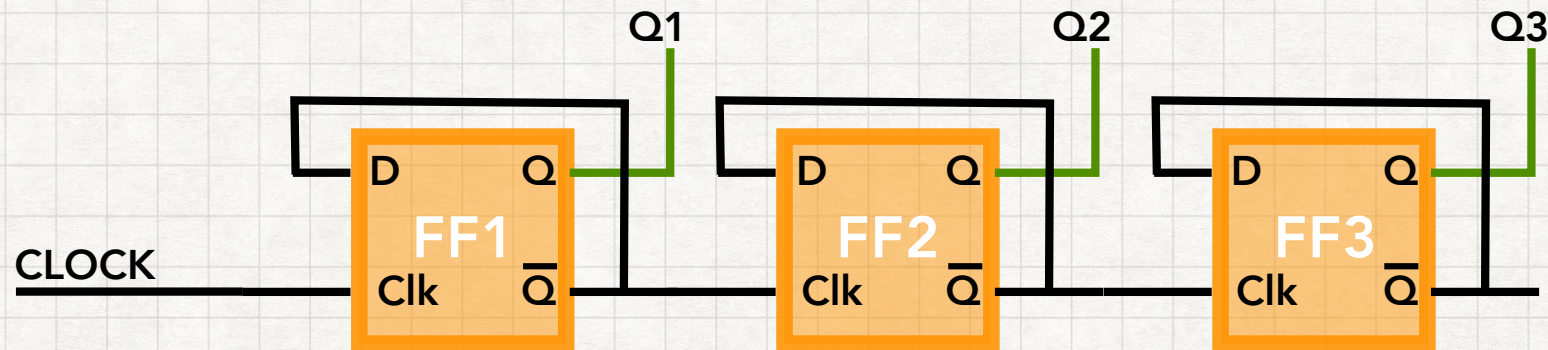
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

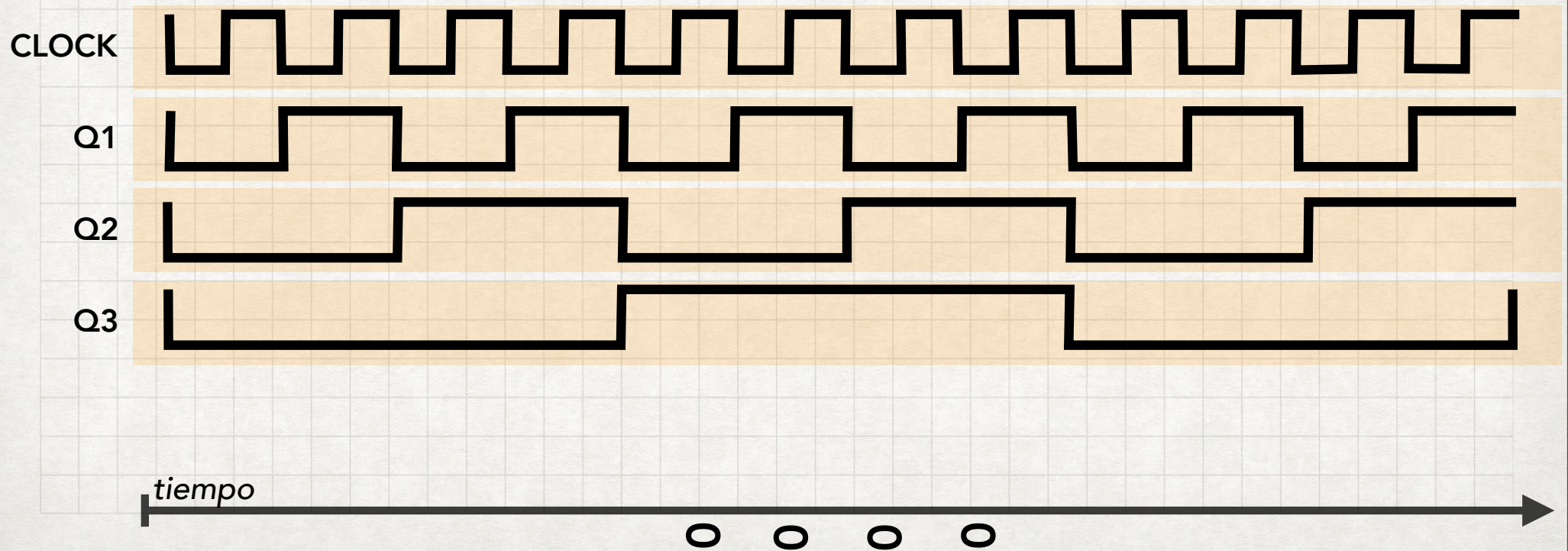
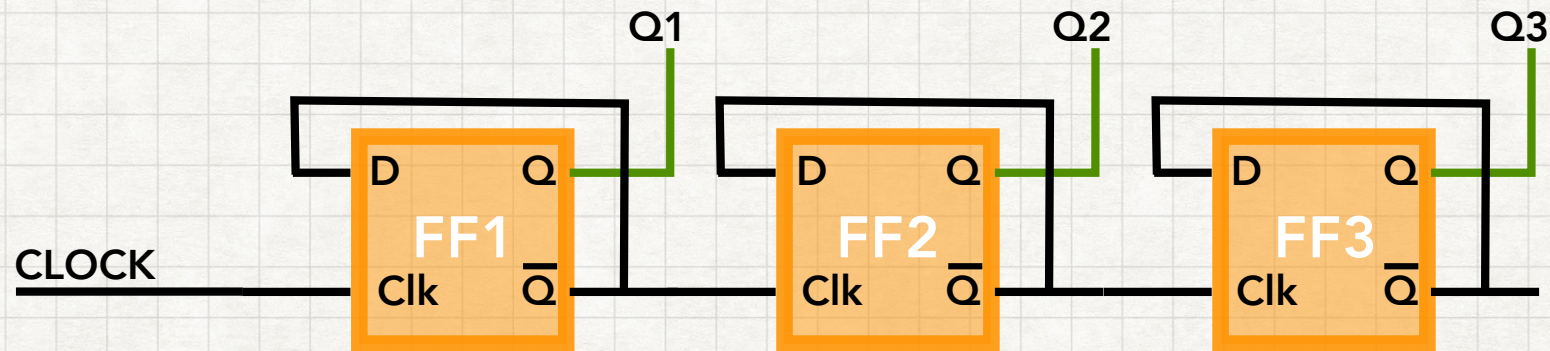
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

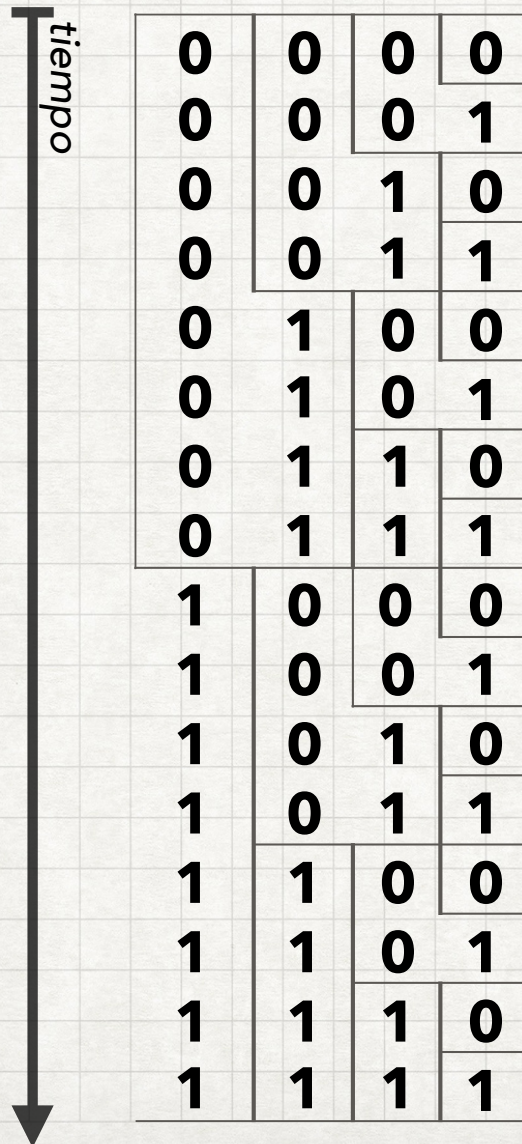


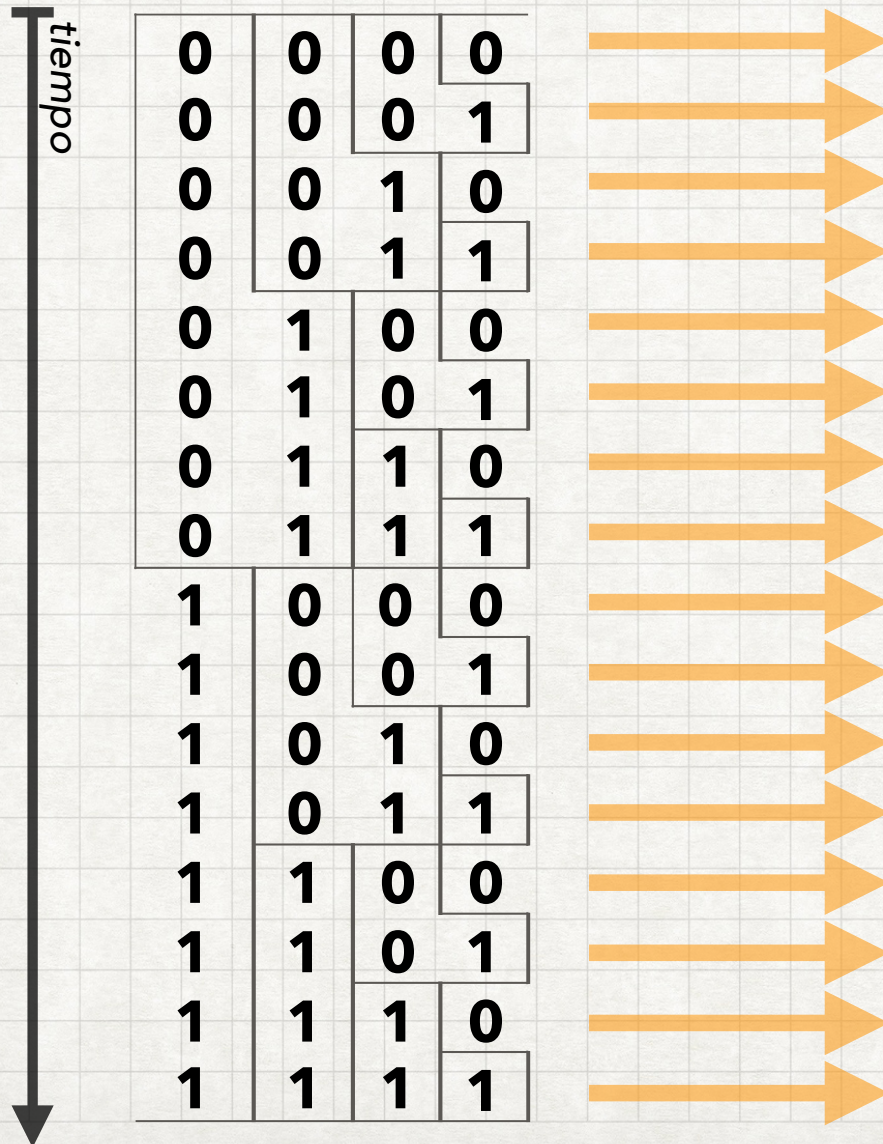
Diagram illustrating the representation of information over time, showing a sequence of 16 time steps (rows) and 4 data columns. The vertical axis is labeled "tiempo" (time).

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



# ORGANIZACIÓN DEL PROCESADOR

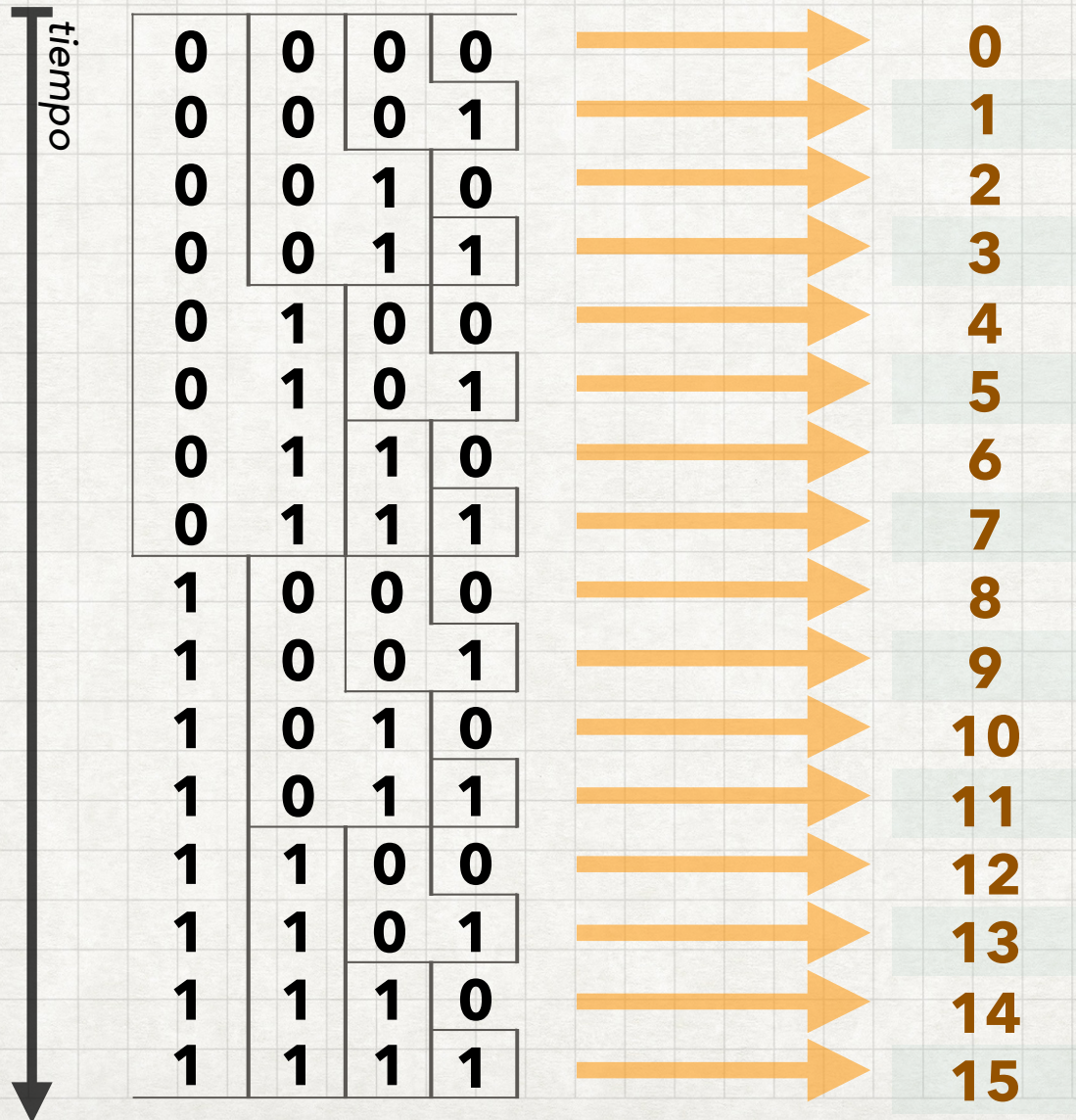
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

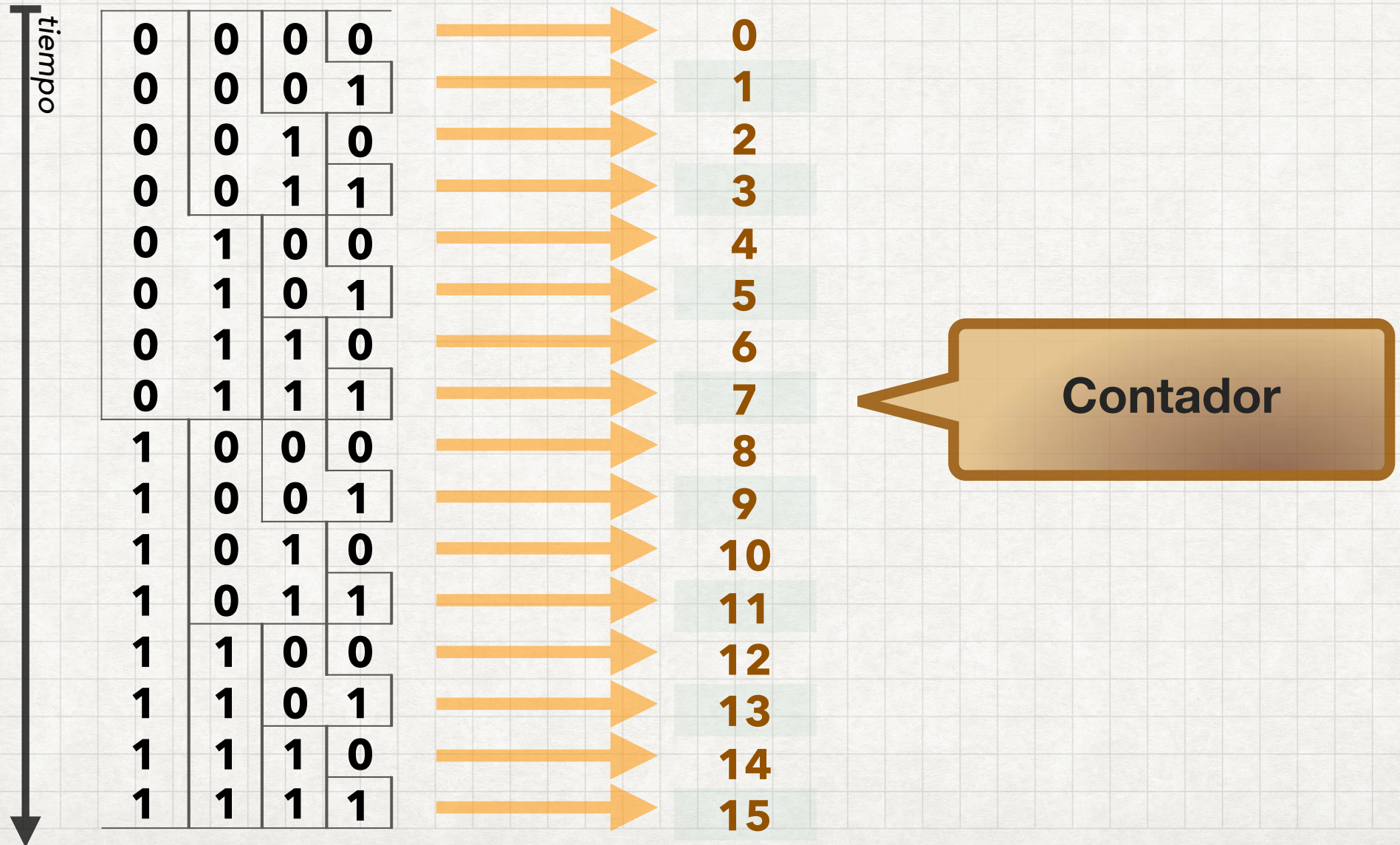
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

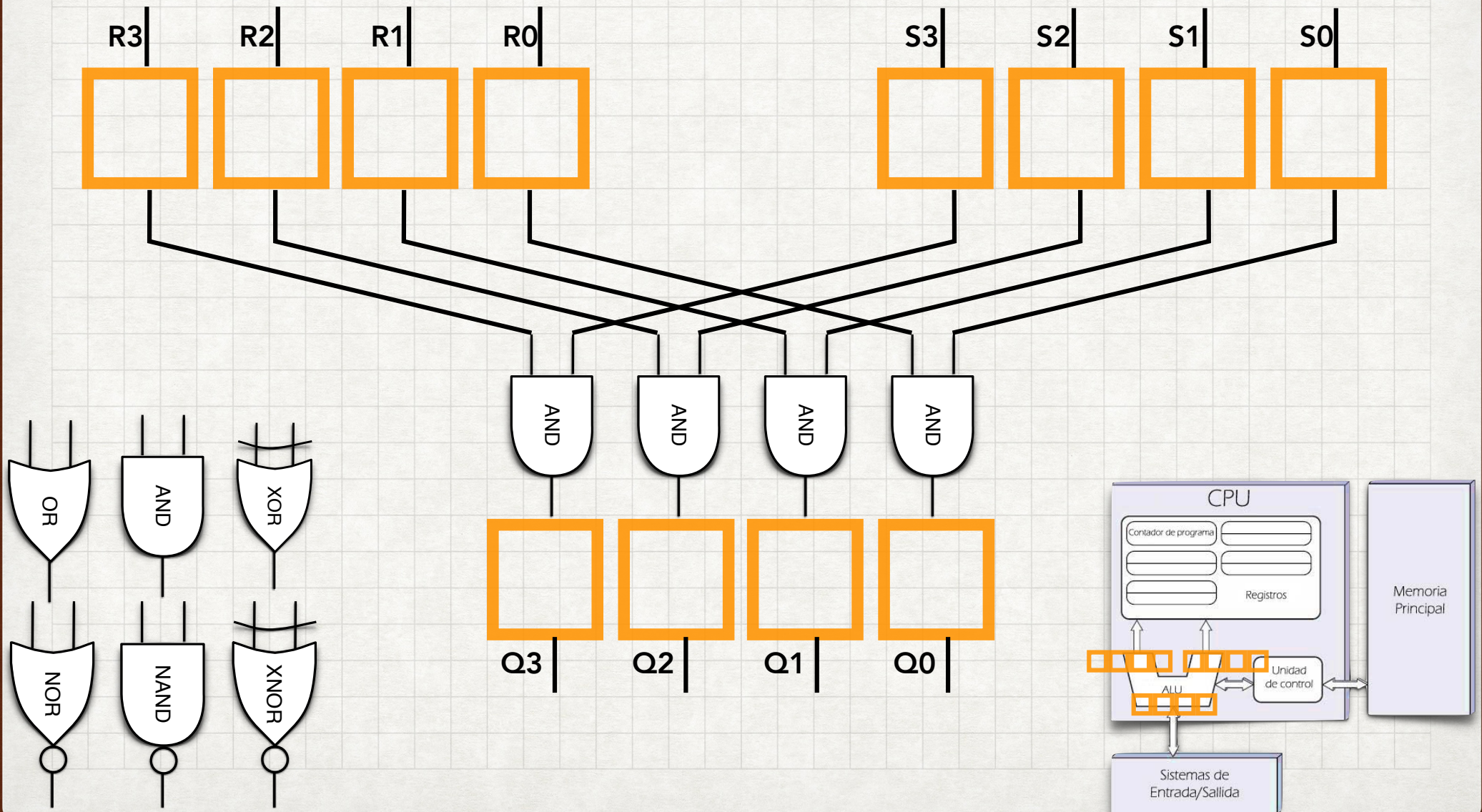
## REPRESENTACIÓN DE INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

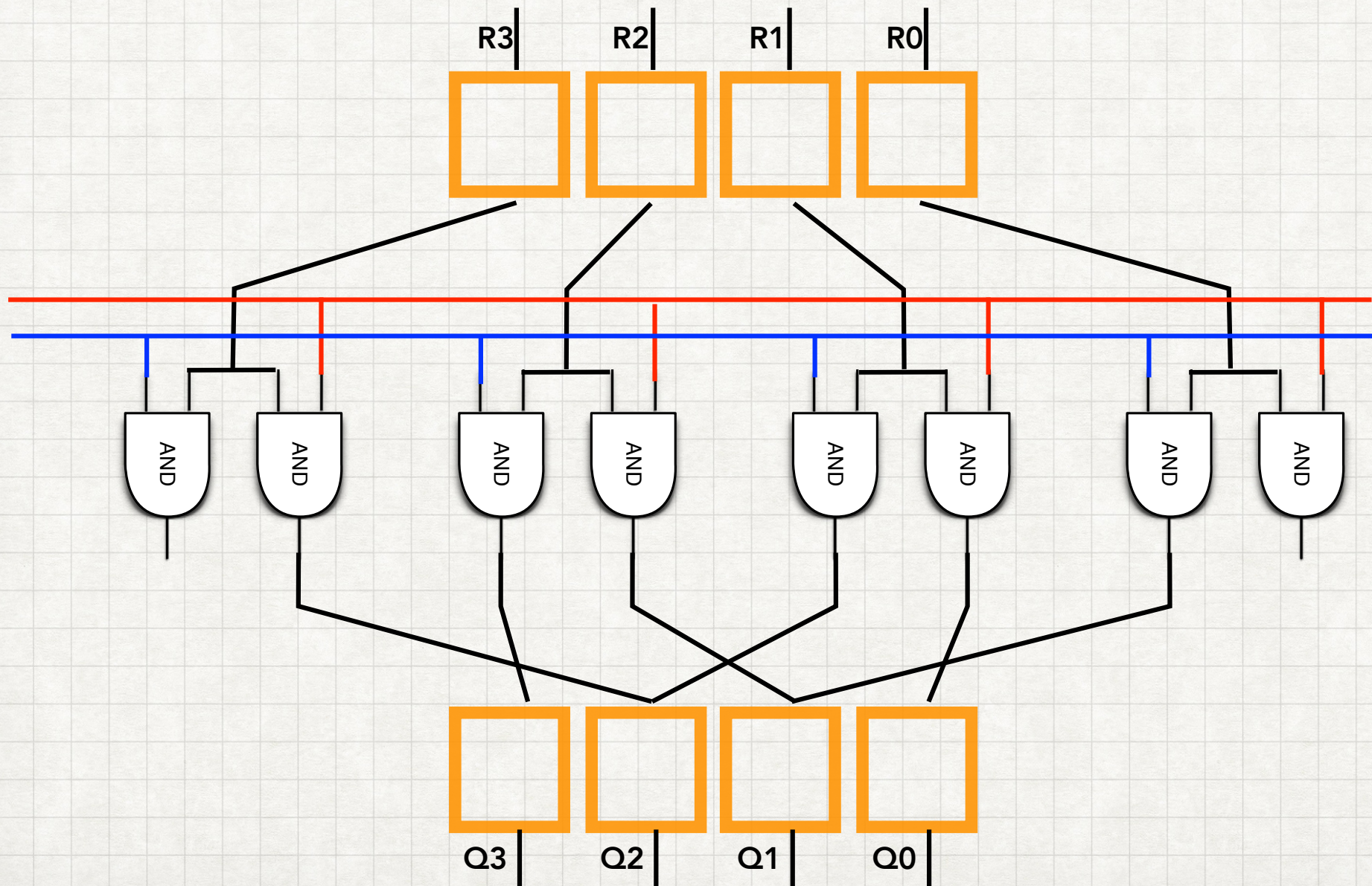
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE INFORMACIÓN

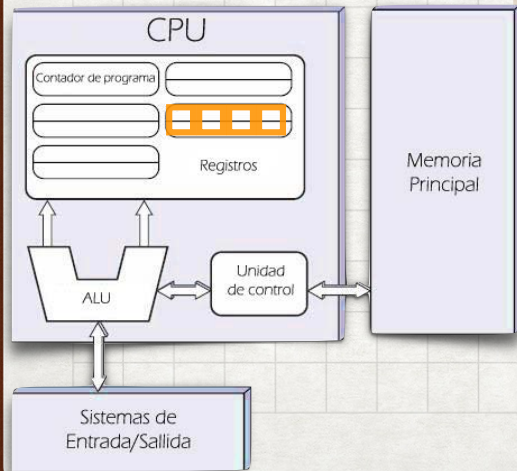
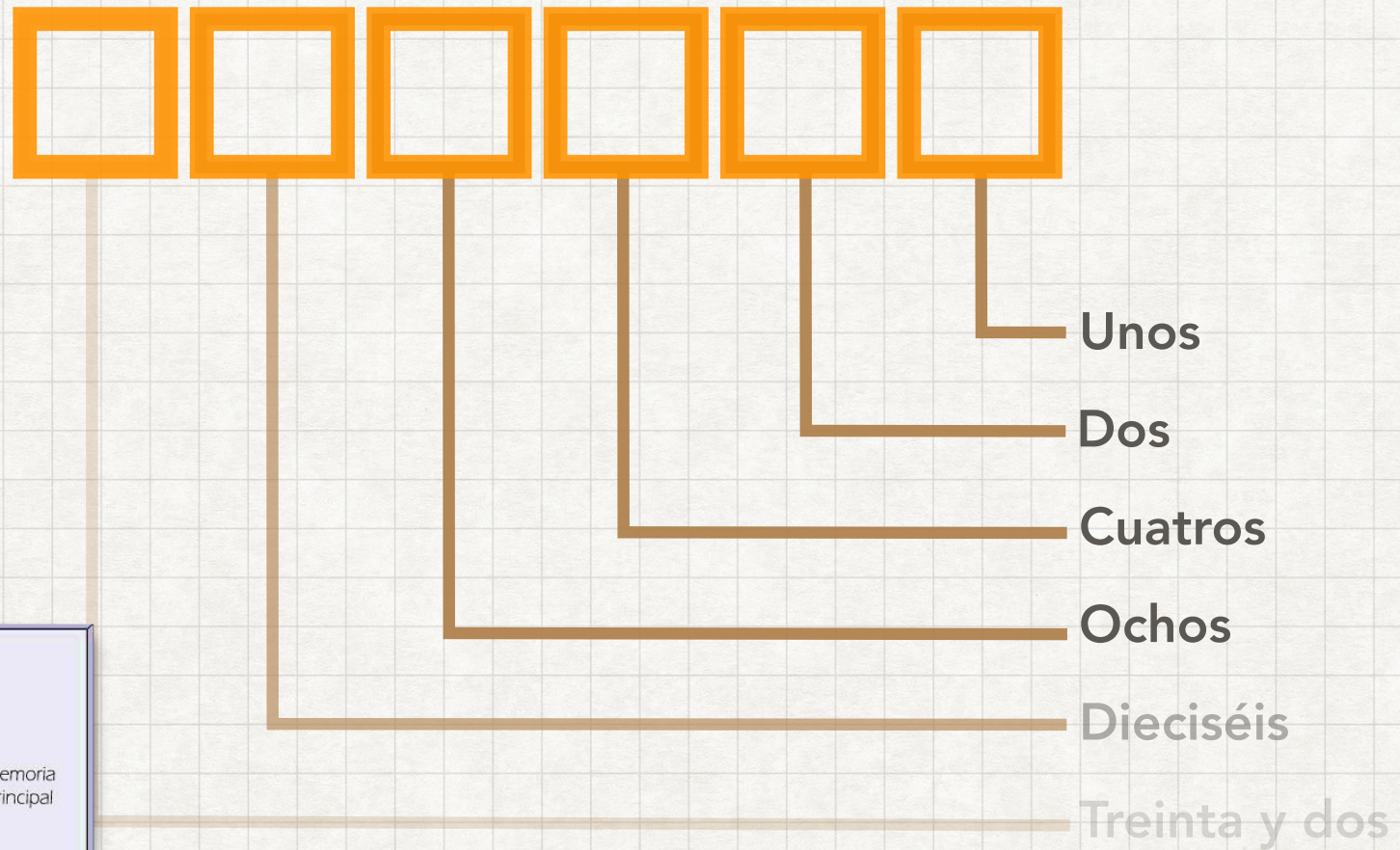
Dependiendo de cada procesador (arquitectura) tenemos diferentes registro en el procesador y con diferentes características:

- Intel 8086/88 (1976): 4 registros de propósito general de 16 bits (AX,BX,CX,DX) pero que también pueden ser utilizados como de 8 bits.
- Intel Pentim III (1999): nuevas instrucciones SSE 8 registros de 128 bits.



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

¿Hasta qué número puedo representar con  $n$  bits?

Cantidad de bits	Puedo identificar
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

$$\begin{array}{rcccc} & 0 & 1 & 1 & 1 \\ + & 0 & 1 & 1 & 0 \\ \hline & \square & \square & \square & \square \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

$$\begin{array}{r} 0111 \\ + 0110 \\ \hline \phantom{0} \phantom{0} \phantom{0} 1 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

$$\begin{array}{r} 1 \\ 0111 \\ + 0110 \\ \hline \phantom{0} \phantom{0} 01 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

$$\begin{array}{r} \phantom{0}1 \phantom{0}1 \\ 0111 \\ + 0110 \\ \hline \phantom{0}101 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

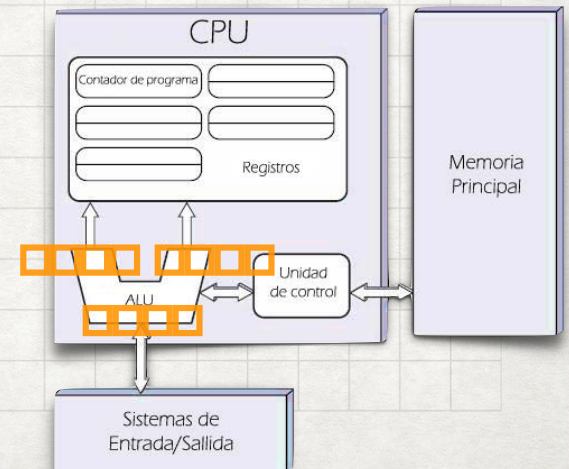
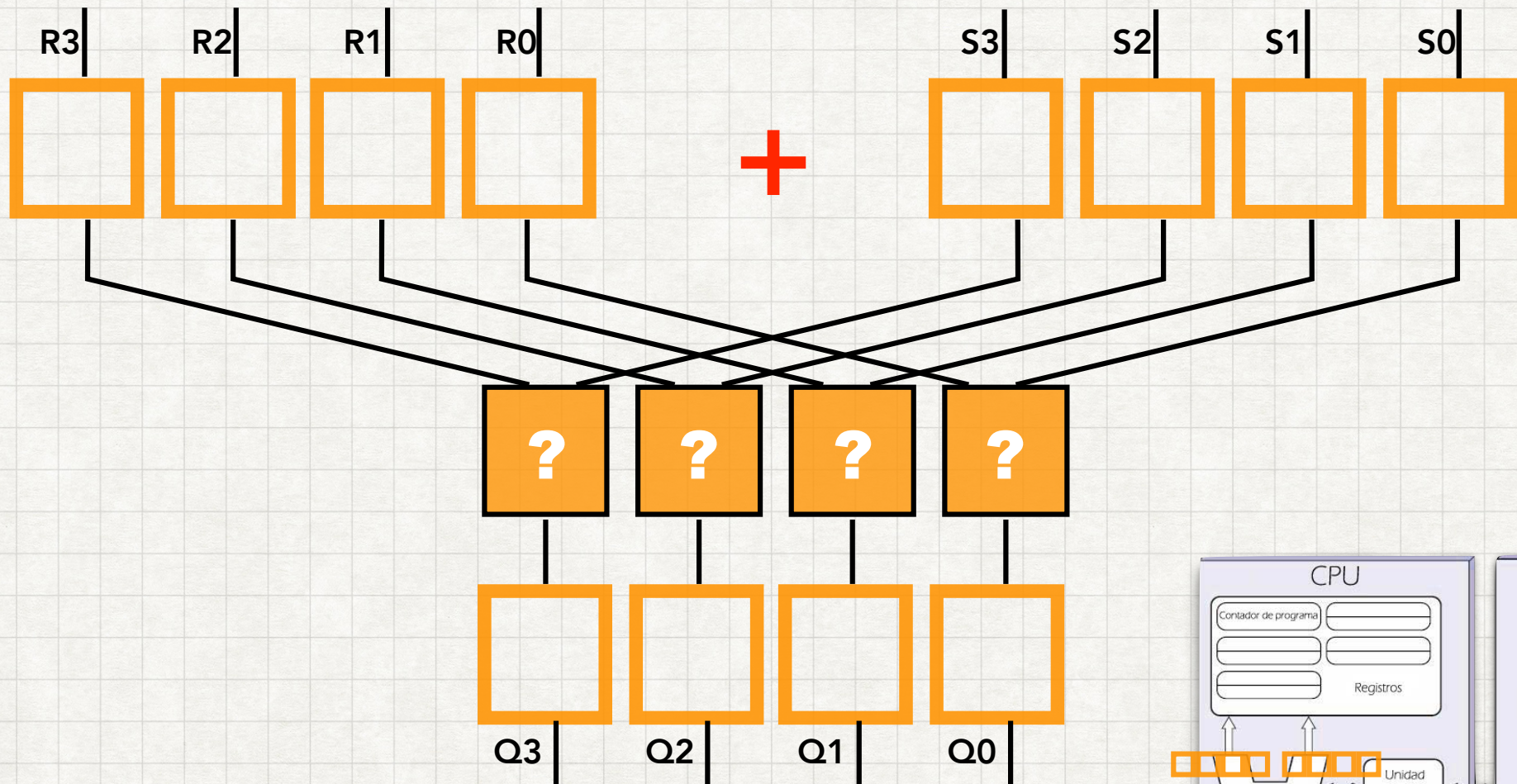
## OPERACIONES CON NÚMEROS ENTEROS

$$\begin{array}{r} \phantom{00}1\phantom{00}1 \\ \phantom{00}\boxed{0}\boxed{1}\boxed{1}\boxed{1} \\ + \phantom{00}\boxed{0}\boxed{1}\boxed{1}\boxed{0} \\ \hline \phantom{00}\boxed{1}\boxed{1}\boxed{0}\boxed{1} \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

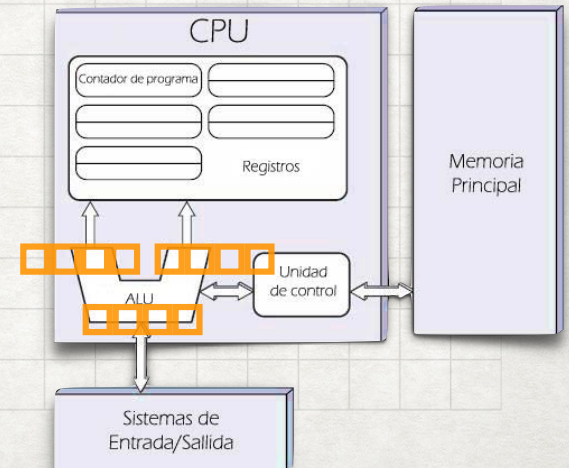
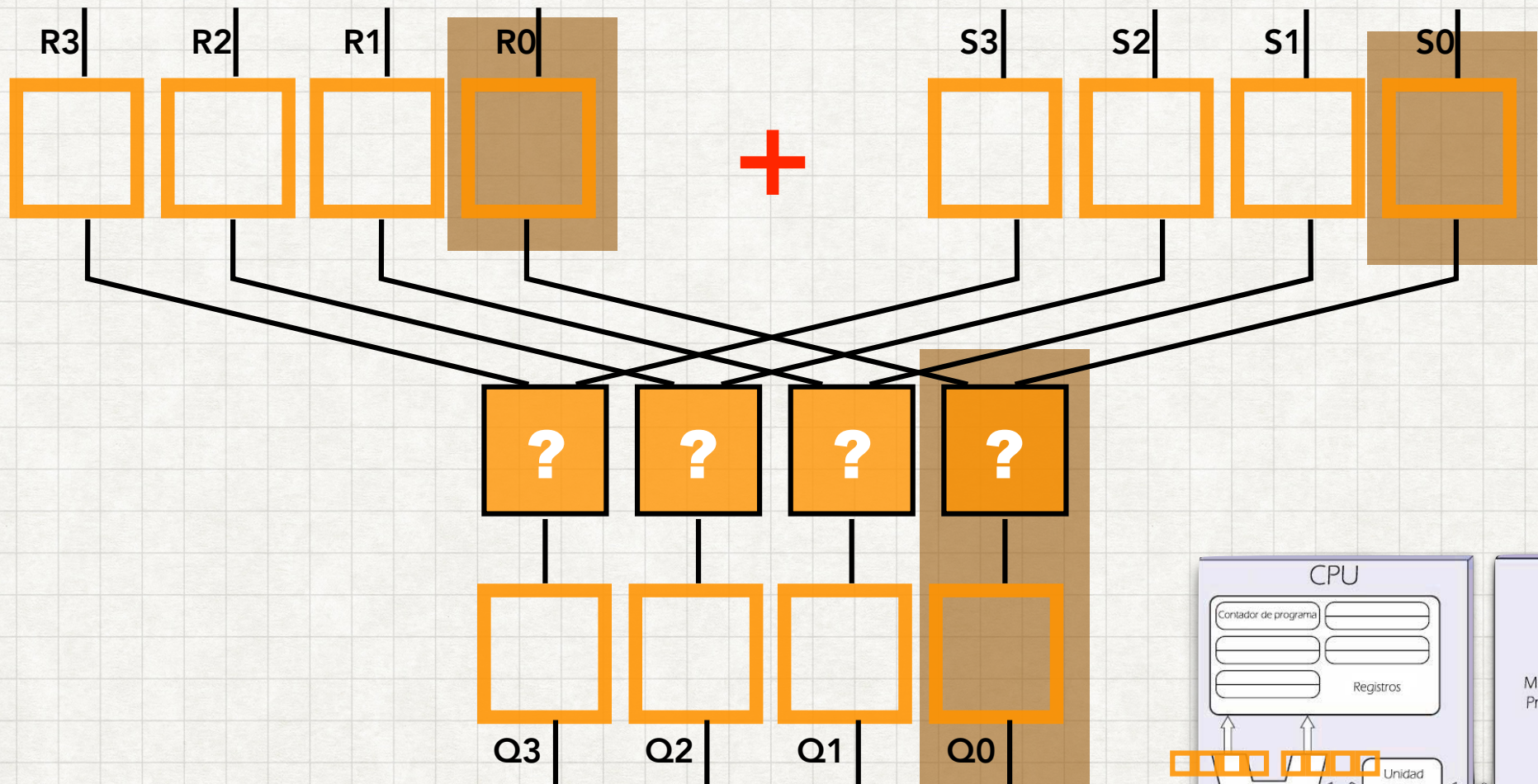
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

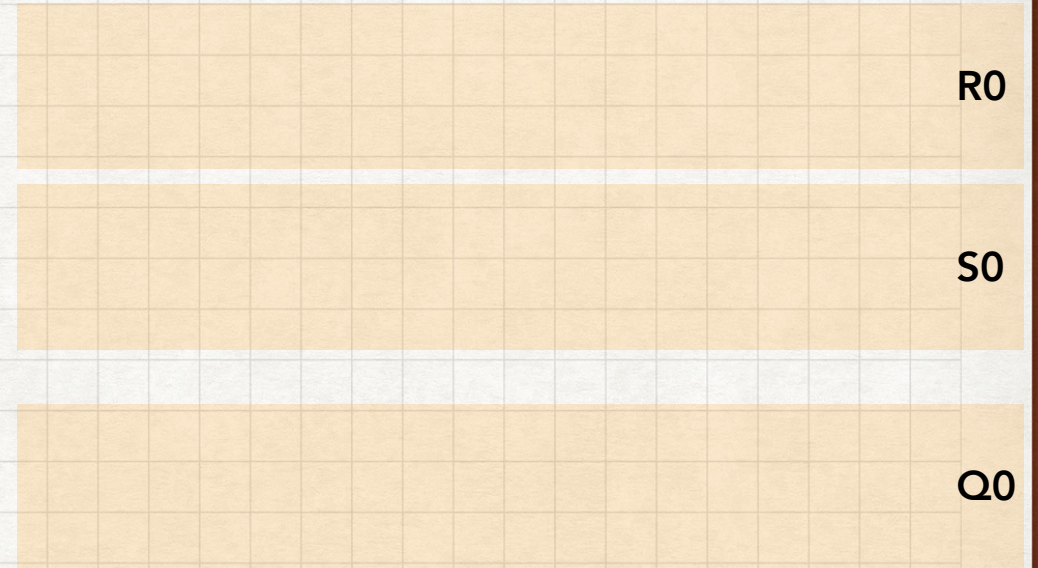
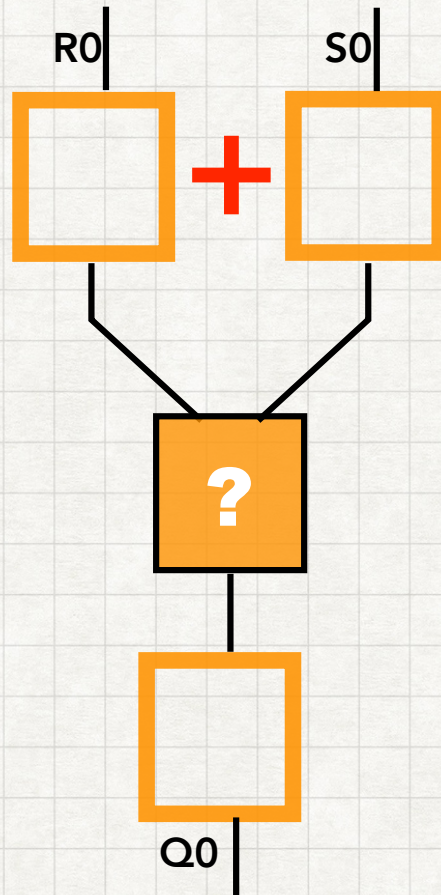
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

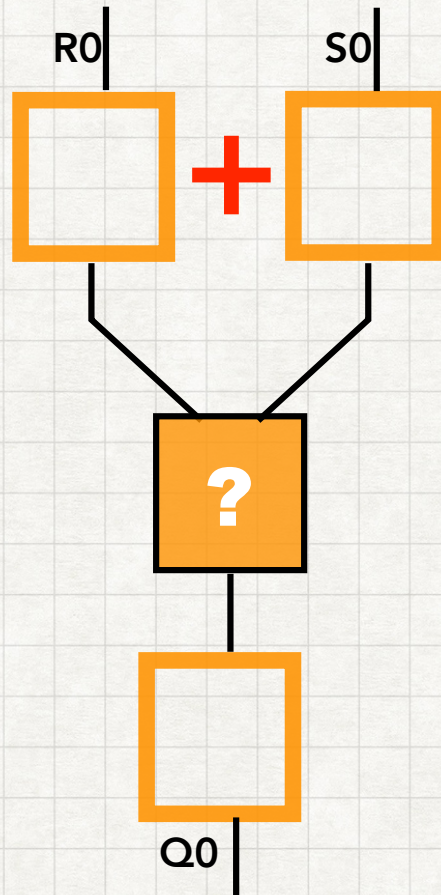
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN

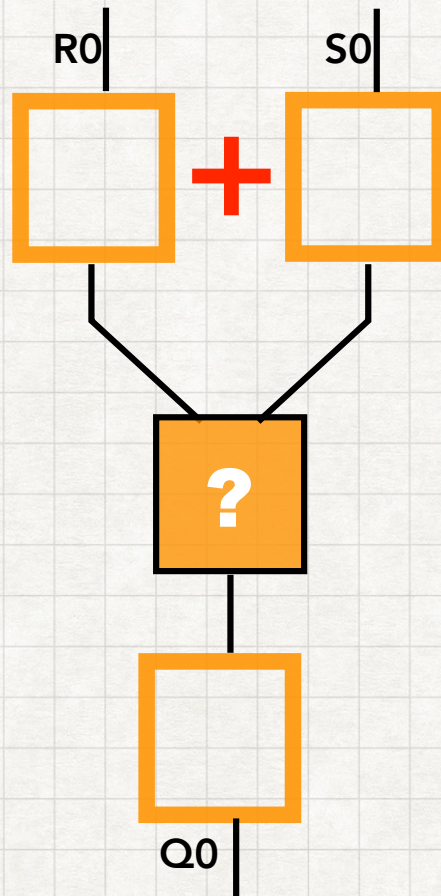


	0	R0
	0	S0
	<hr/>	
	0	Q0



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN

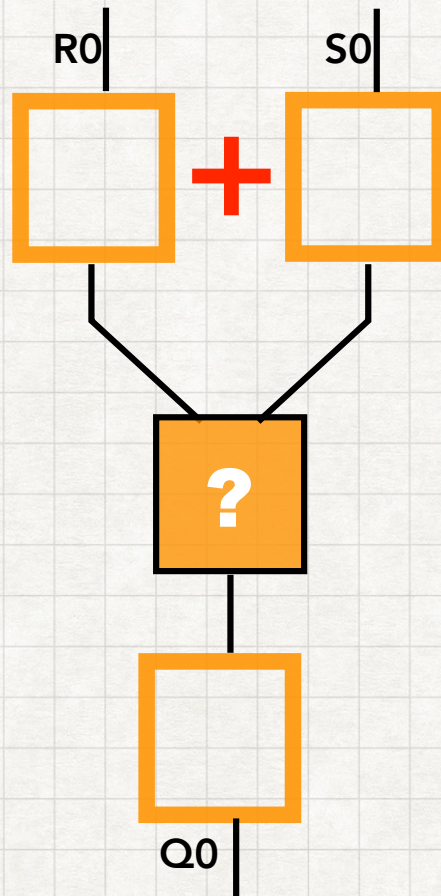


	0	0	$R0$
	1	0	$S0$
	<hr/>	<hr/>	
	1	0	$Q0$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN

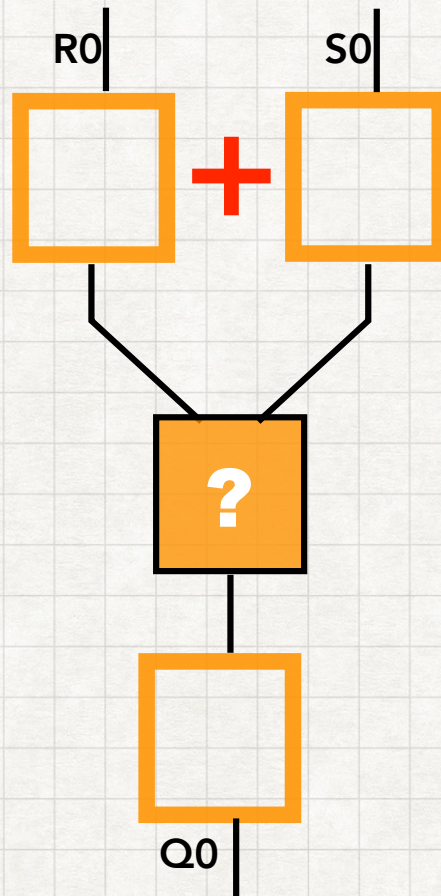


1	0	0	R0
0	1	0	S0
1	1	0	Q0



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN

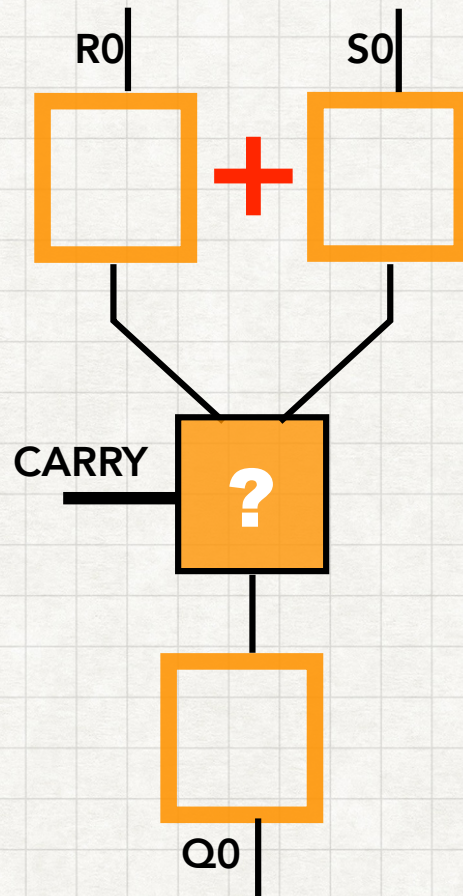


1	1	0	0	$R0$
1	0	1	0	$S0$
<hr/>				
0	1	1	0	$Q0$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN

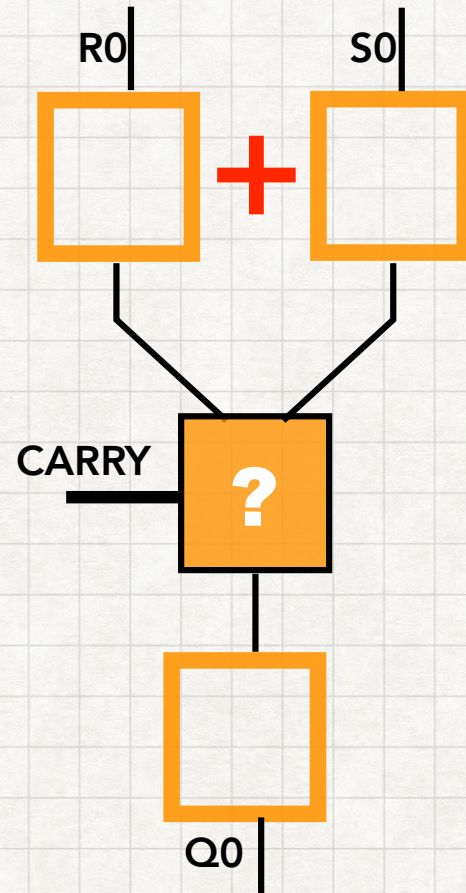


	1	1	0	0	R0
	1	0	1	0	S0
	<hr/> <hr/>				
1	0	1	1	0	Q0



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



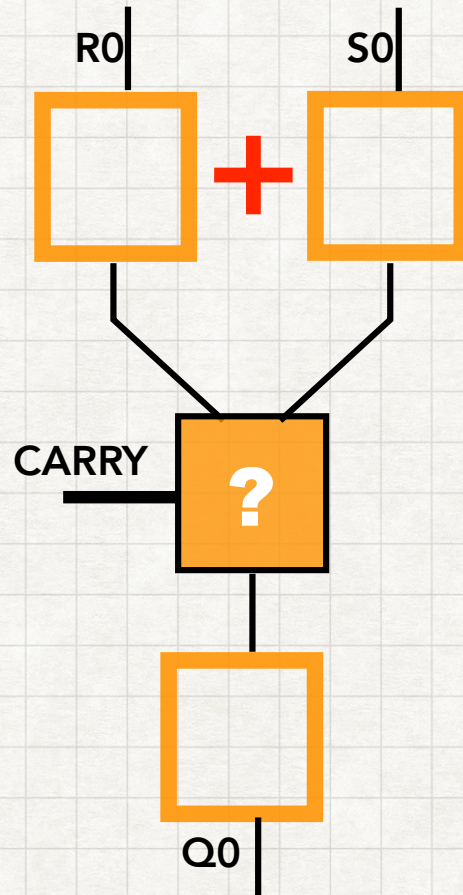
1	1	0	0	R0
1	0	1	0	S0
1	0	1	1	Q0

R0	S0	Q0
F	F	F
F	T	T
T	F	T
T	T	F



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



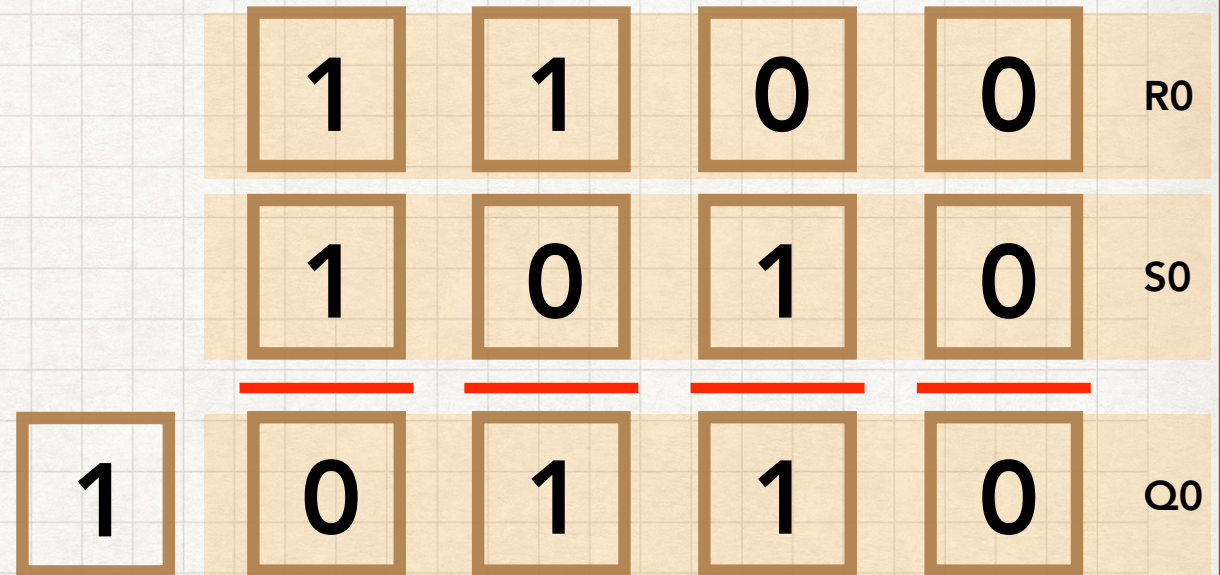
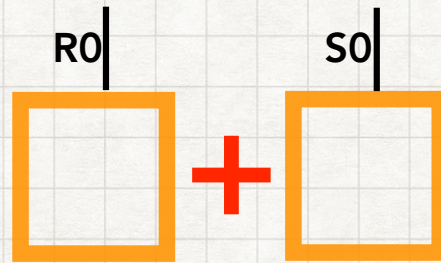
1	1	0	0	R0
1	0	1	0	S0
1	0	1	1	Q0

R0	S0	Q0	CARRY
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T

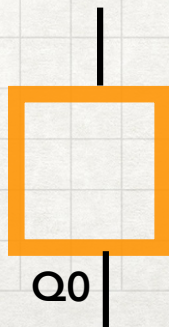


# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



CARRY

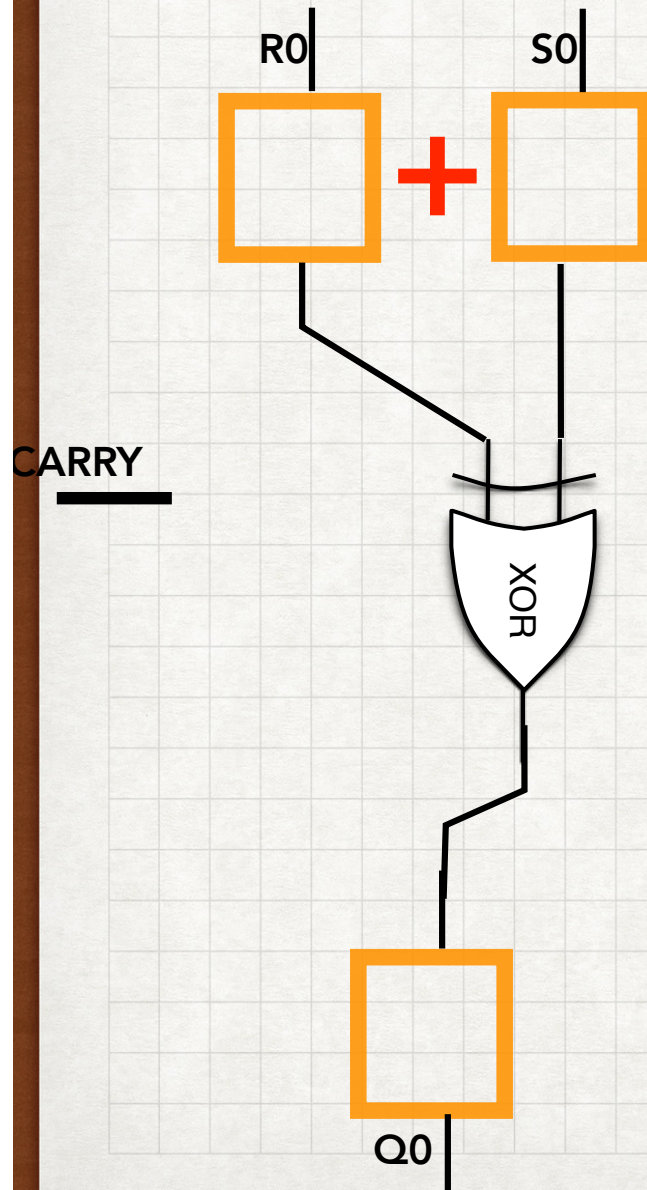


R0	S0	Q0	CARRY
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



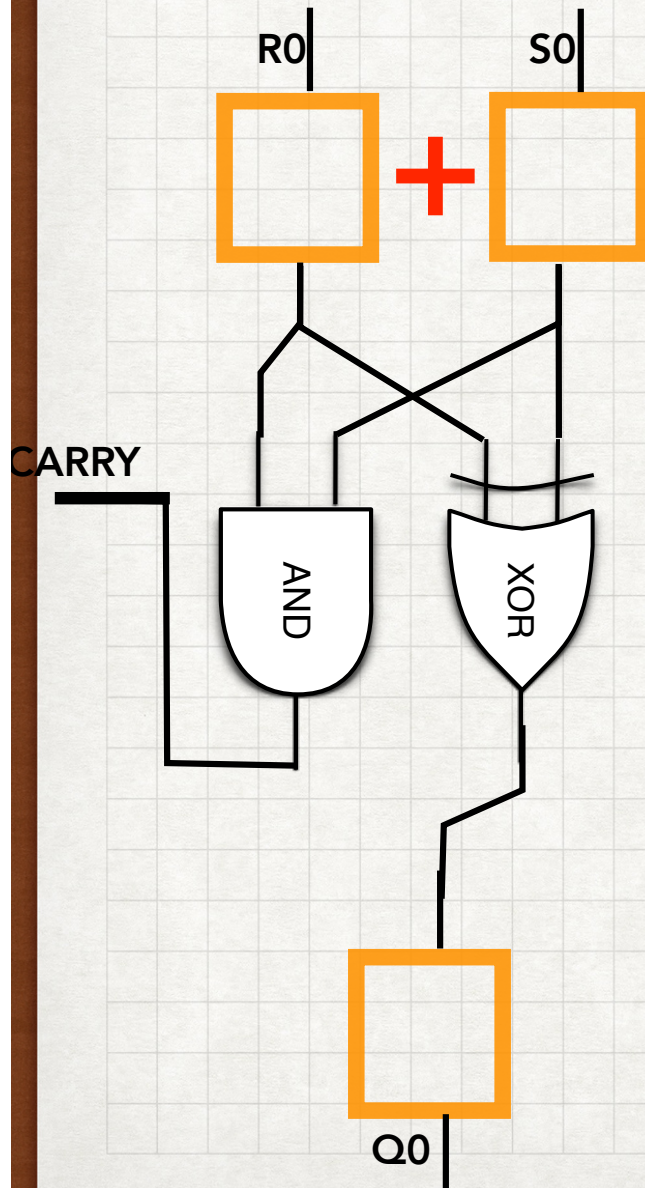
1	1	0	0	R0
1	0	1	0	S0
1	0	1	1	Q0
0	1	1	0	

R0	S0	Q0	CARRY
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



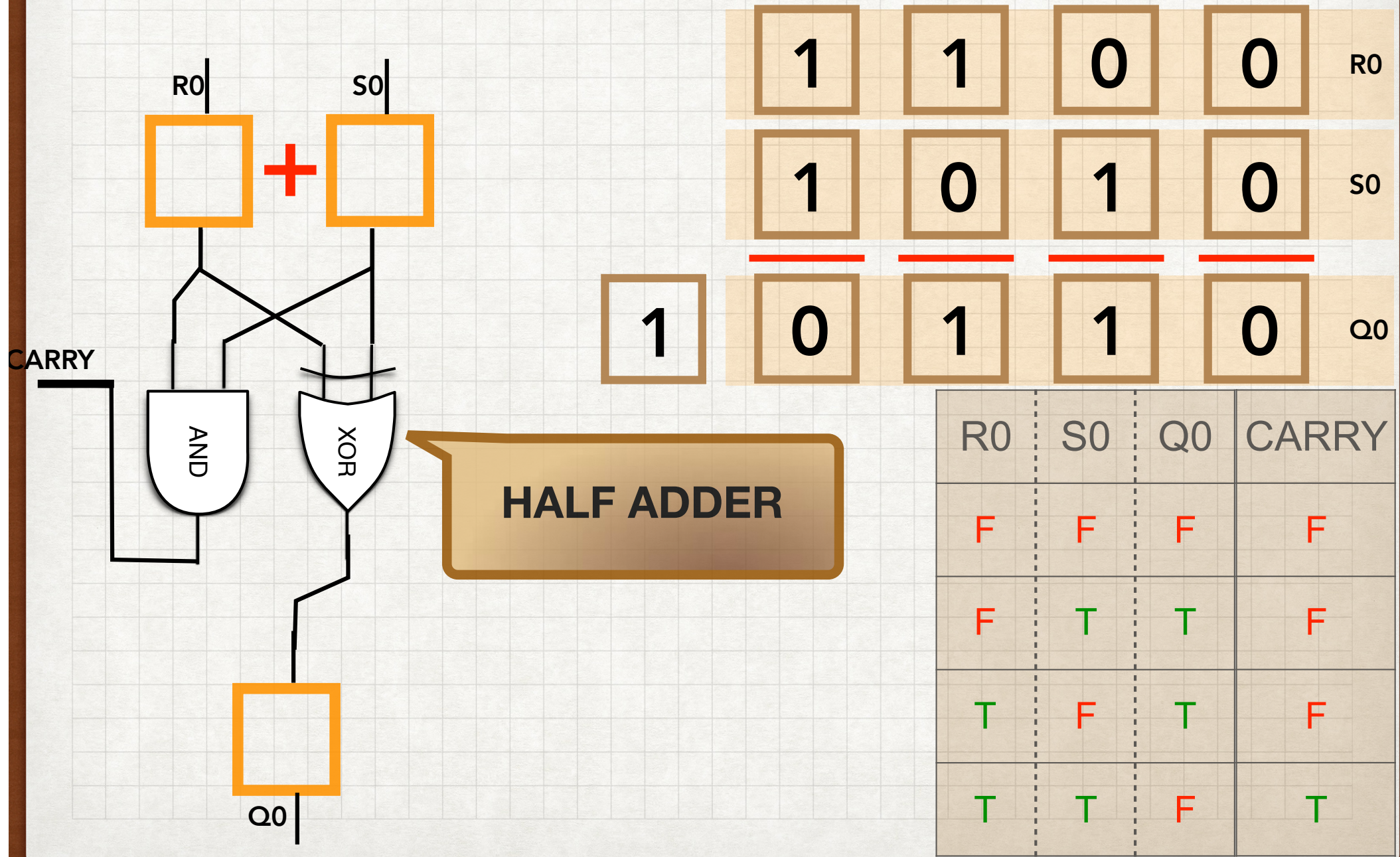
1	1	0	0	R0
1	0	1	0	S0
1	0	1	1	Q0

R0	S0	Q0	CARRY
F	F	F	F
F	T	T	F
T	F	T	F
T	T	F	T



# ORGANIZACIÓN DEL PROCESADOR

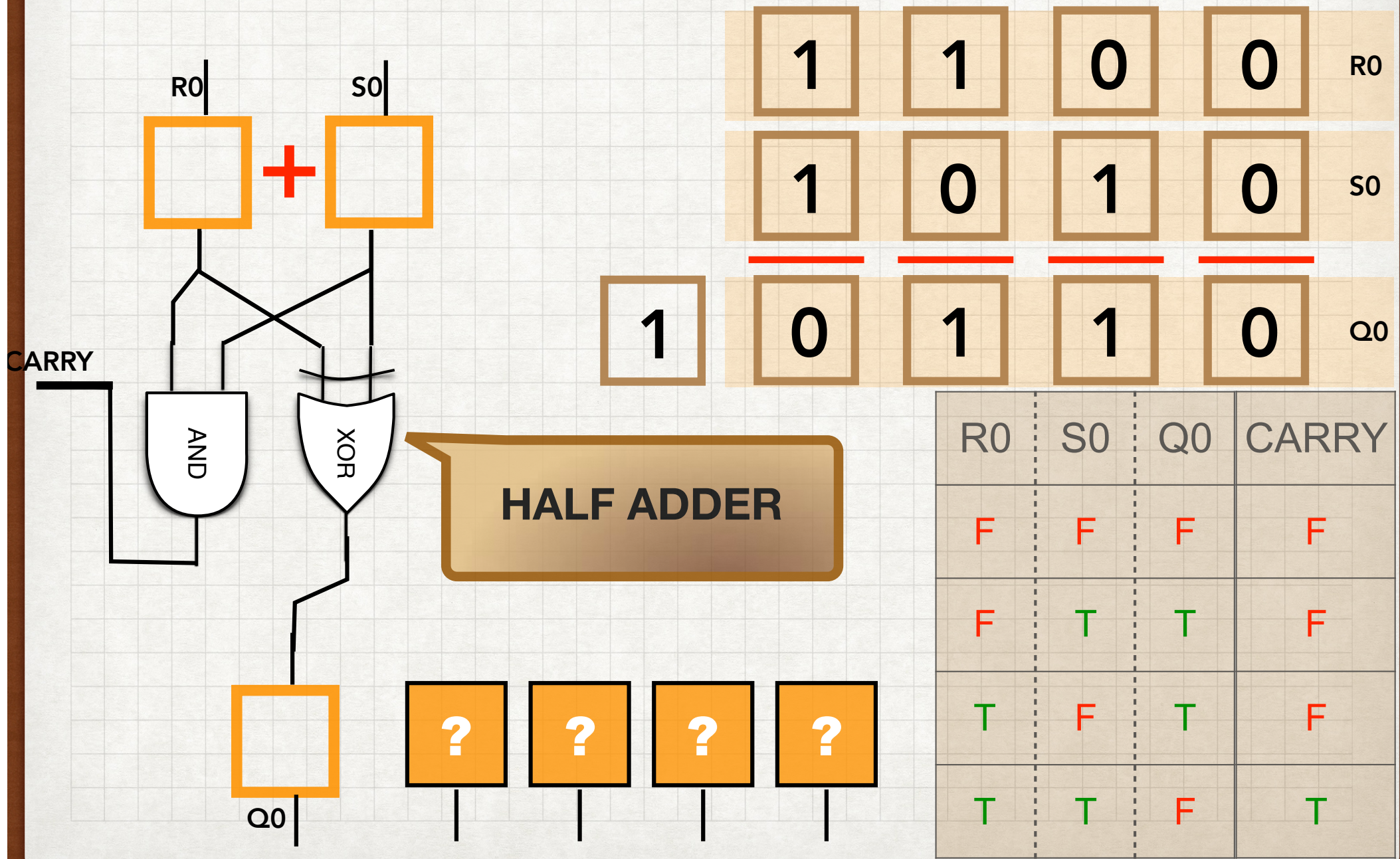
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

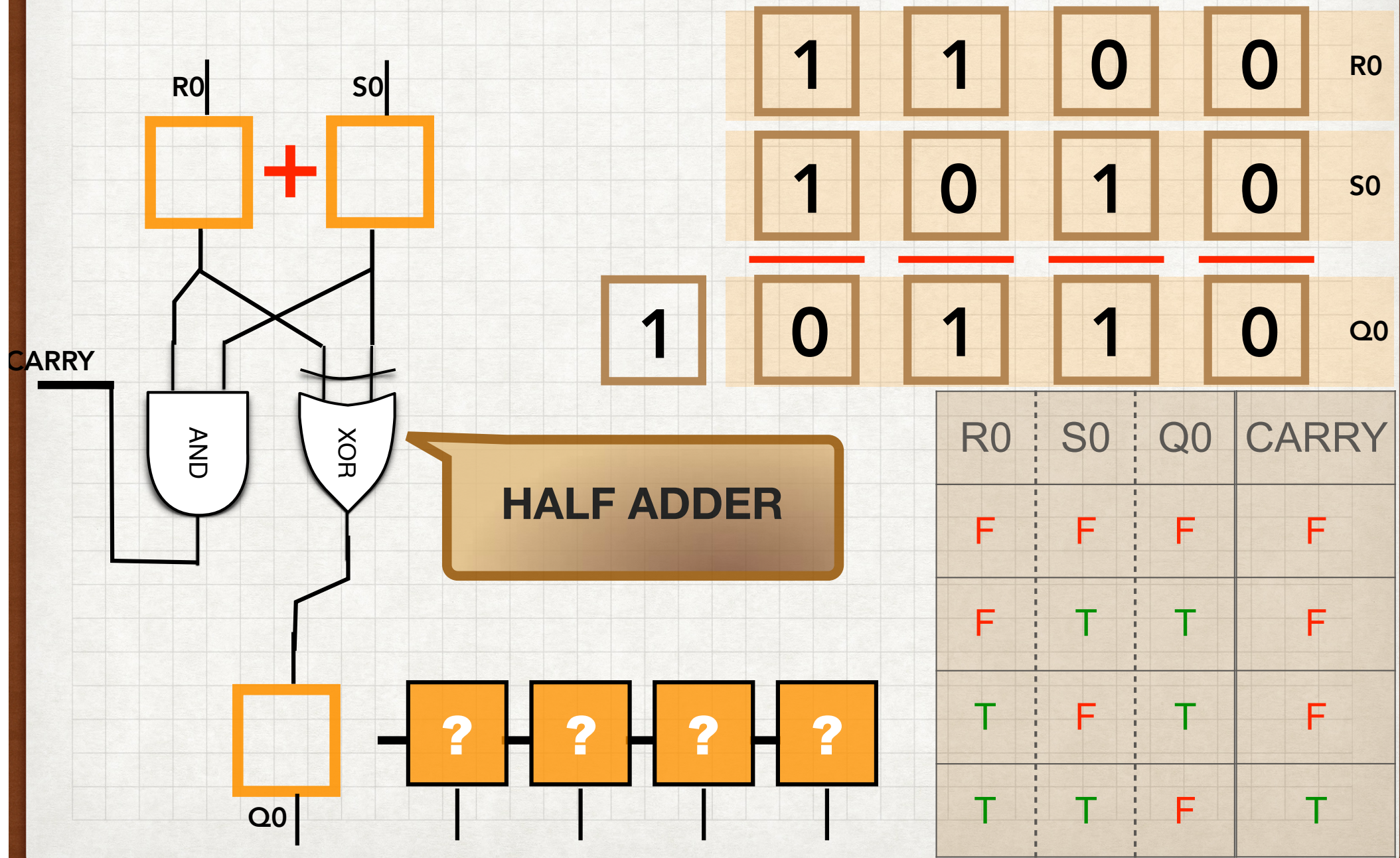
## OPERACIONES CON INFORMACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON INFORMACIÓN



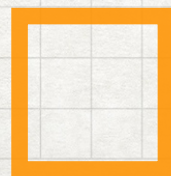


# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS



entrada			salida	
R0	S0	Cin	Q0	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

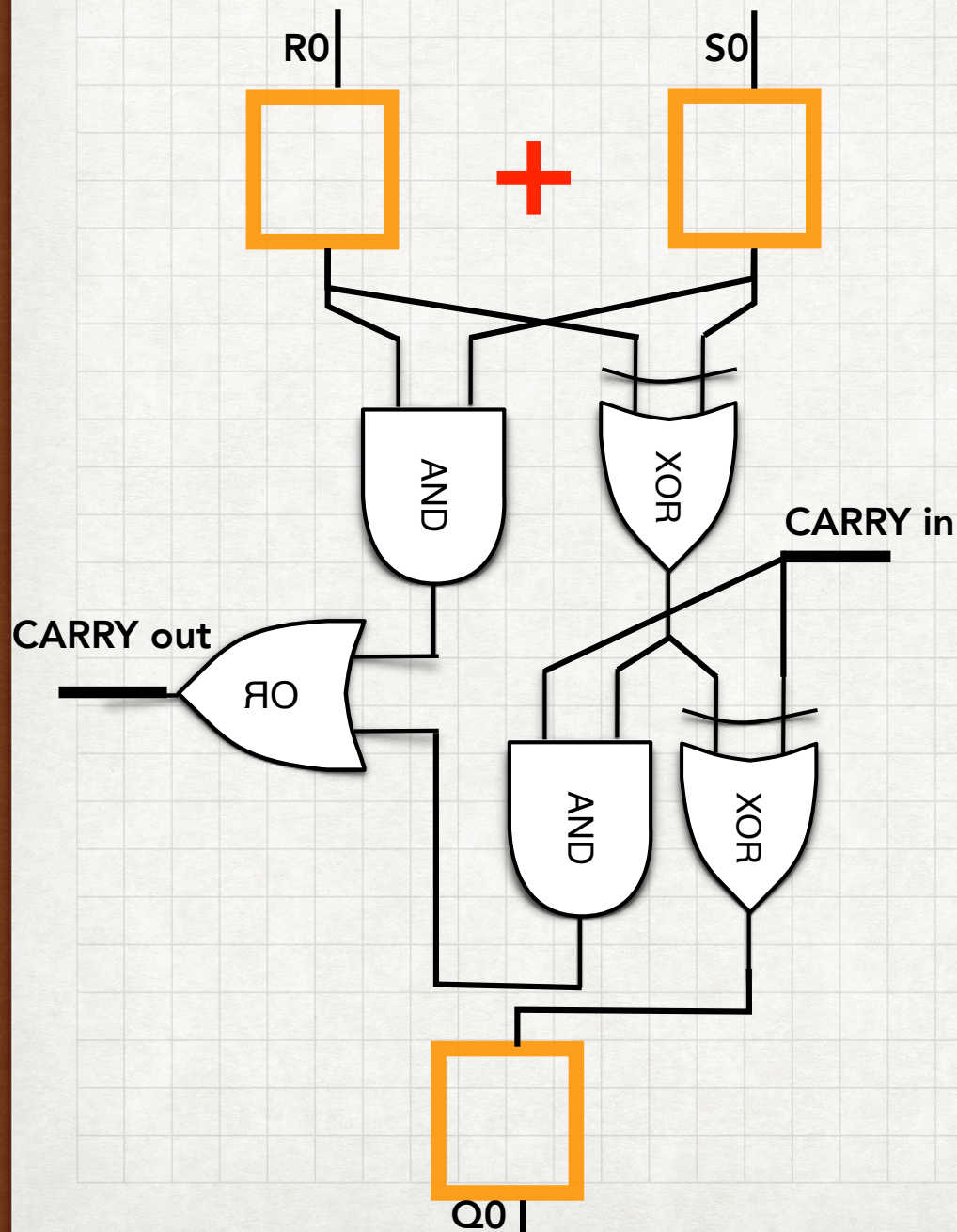


Q0



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

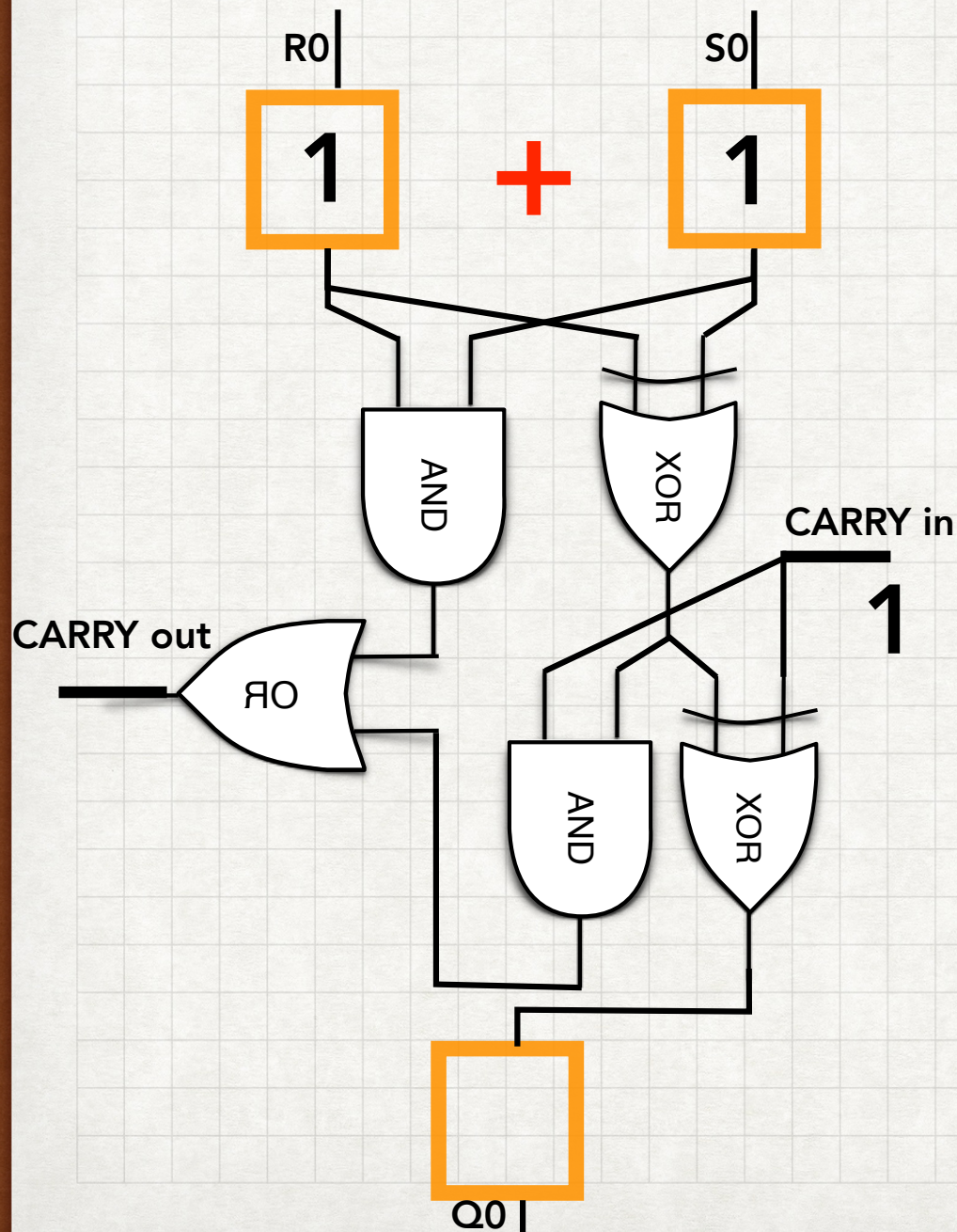


entrada			salida	
R0	S0	Cin	Q0	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

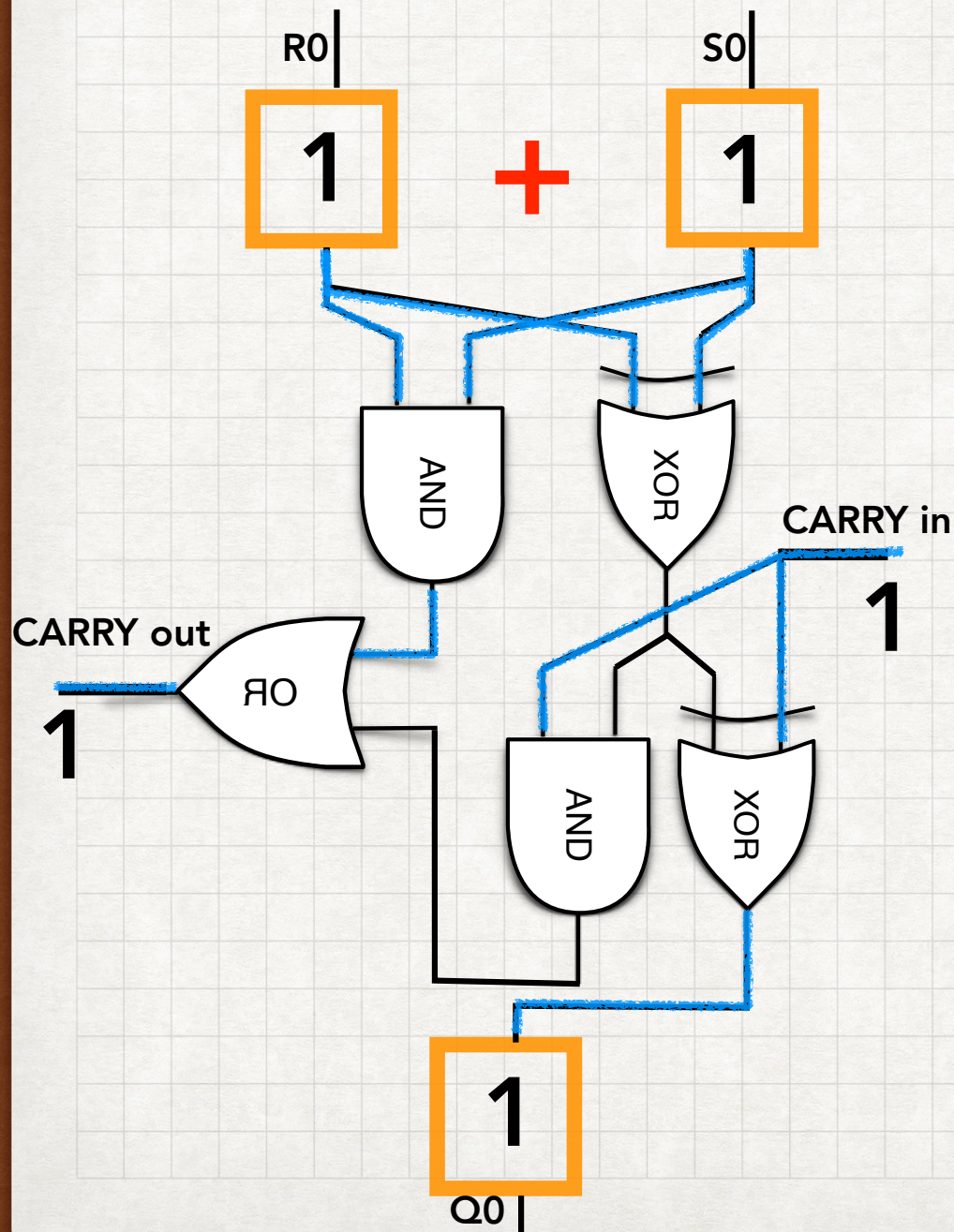


entrada			salida	
R0	S0	Cin	Q0	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS

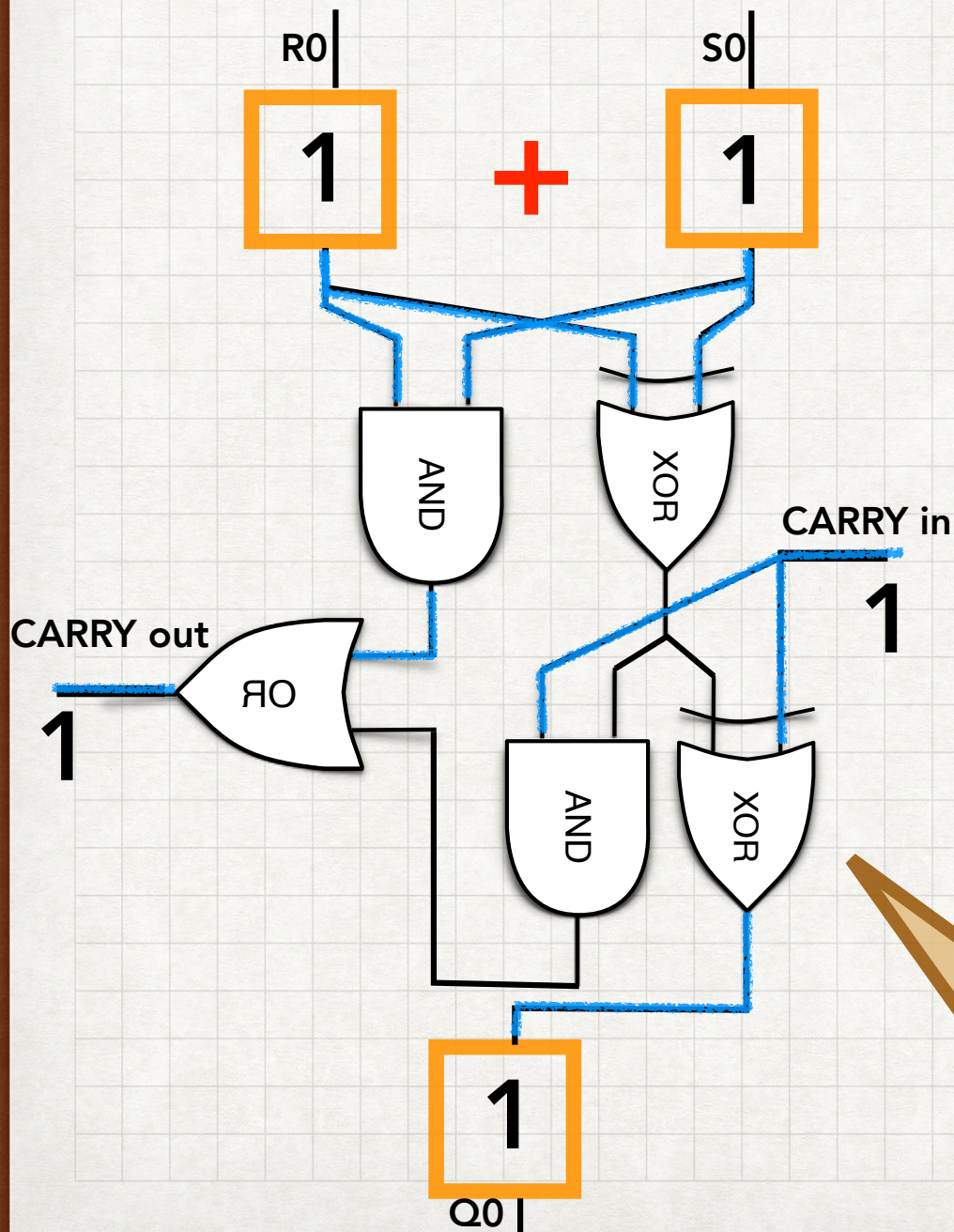


entrada			salida	
R0	S0	Cin	Q0	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS



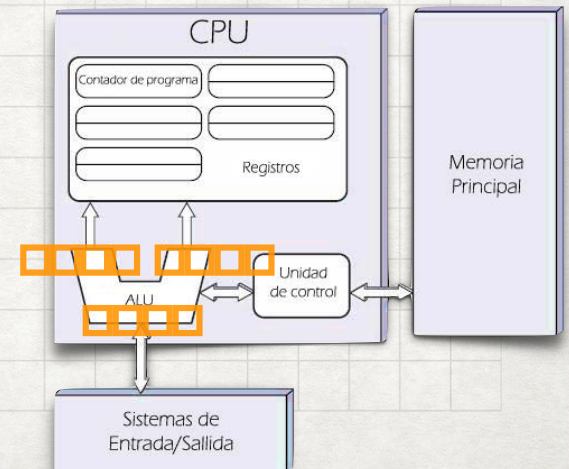
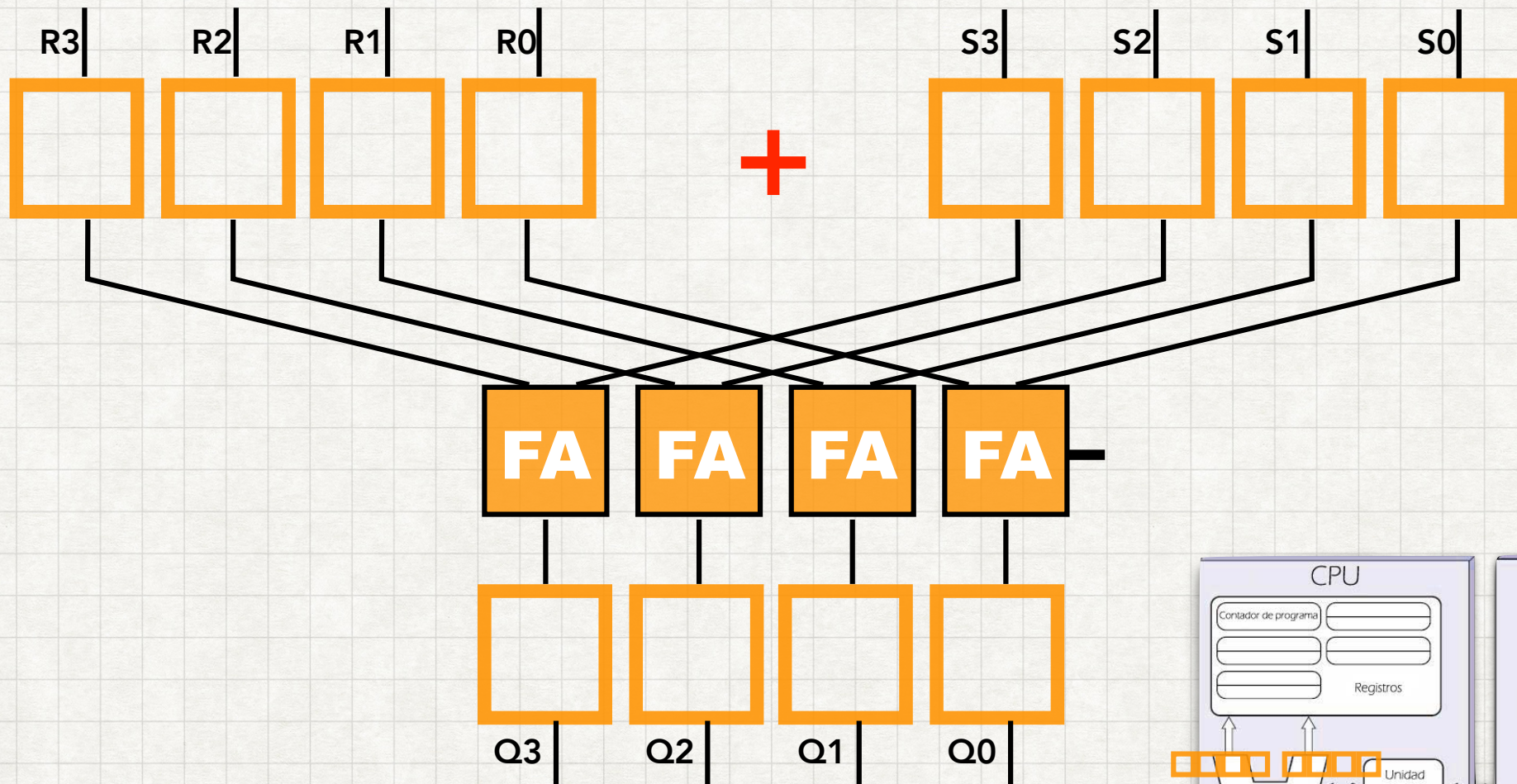
entrada			salida	
R0	S0	Cin	Q0	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**FULL ADDER**



# ORGANIZACIÓN DEL PROCESADOR

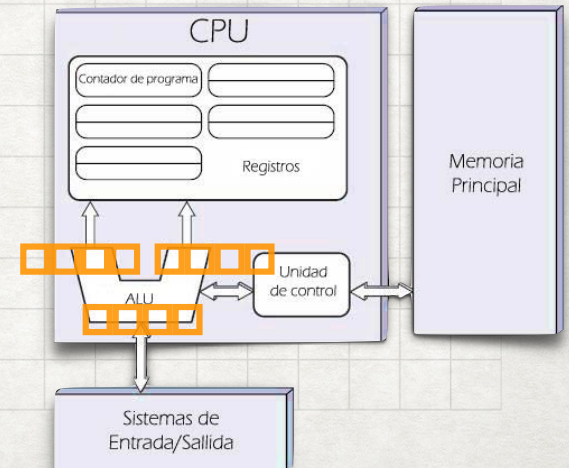
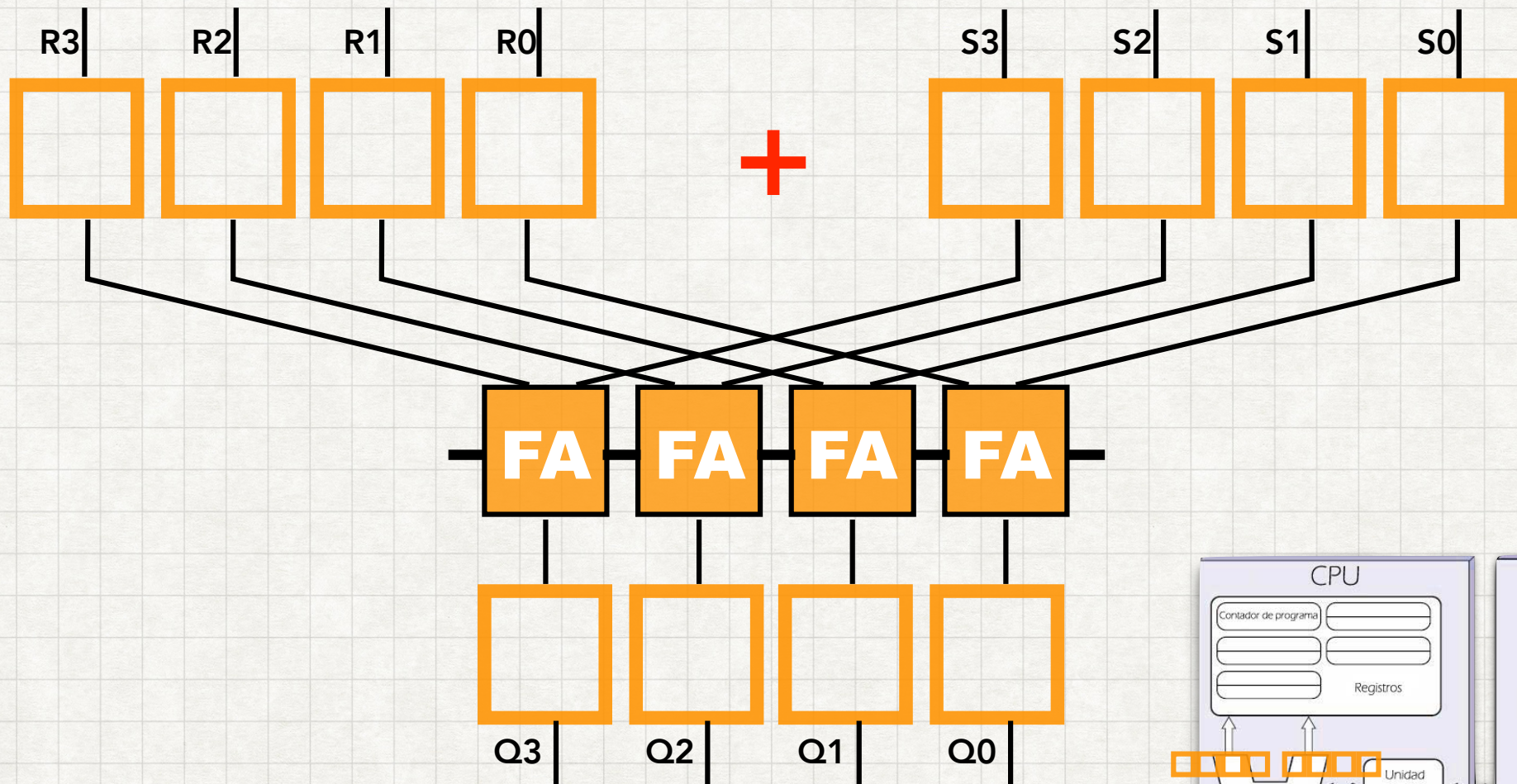
## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

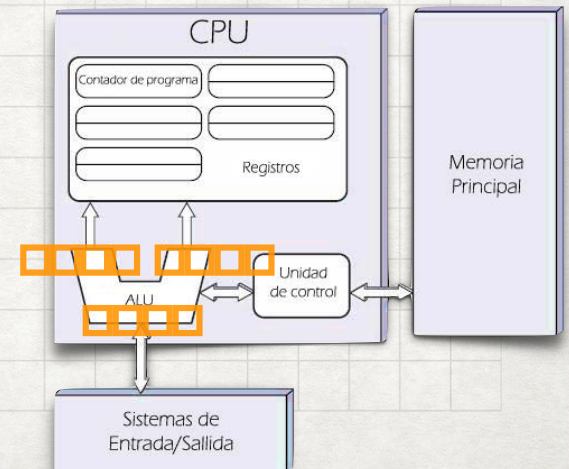
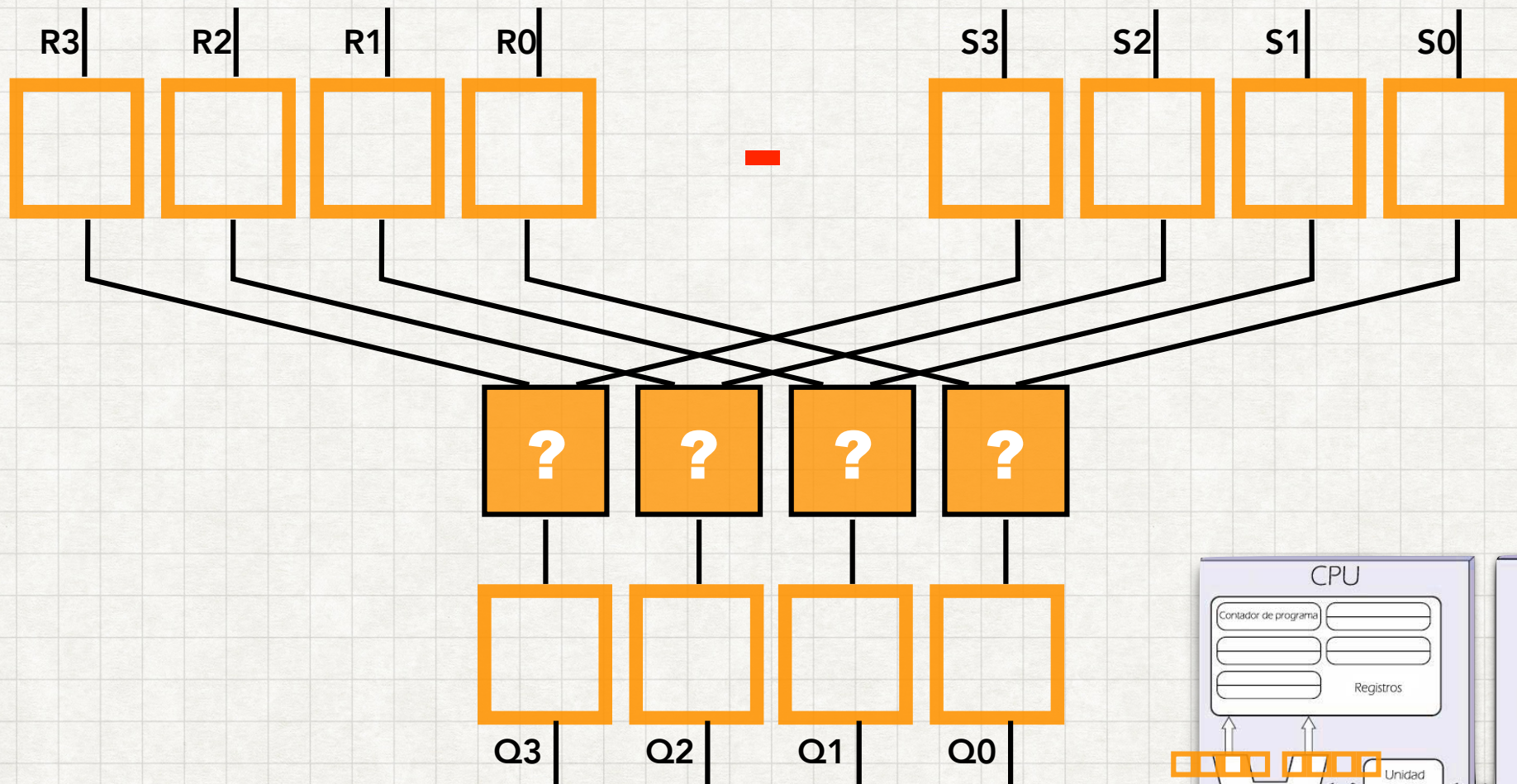
## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES - RESTA

$$\begin{array}{r} 1101 \\ - 0110 \\ \hline \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES - RESTA

$$\begin{array}{r} 1101 \\ - 0110 \\ \hline \phantom{00} \phantom{00} \phantom{00} 1 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES - RESTA

$$\begin{array}{r} 1011 \\ - 0110 \\ \hline \phantom{0} \phantom{0} \phantom{0} 1 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES - RESTA

$$\begin{array}{r} 1011 \\ - 0110 \\ \hline \phantom{0} \phantom{0} 11 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES - RESTA

$$\begin{array}{r} 0101 \\ - 0110 \\ \hline 0111 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

¿cómo represento el siguiente número?

**- 78**

0	1	0	0	1	1	1	0
128	64	32	16	8	4	2	1



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

¿cómo represento el siguiente número?

**- 78**





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

¿cómo represento el siguiente número?

**- 78**





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

¿cómo represento el siguiente número?

- 78



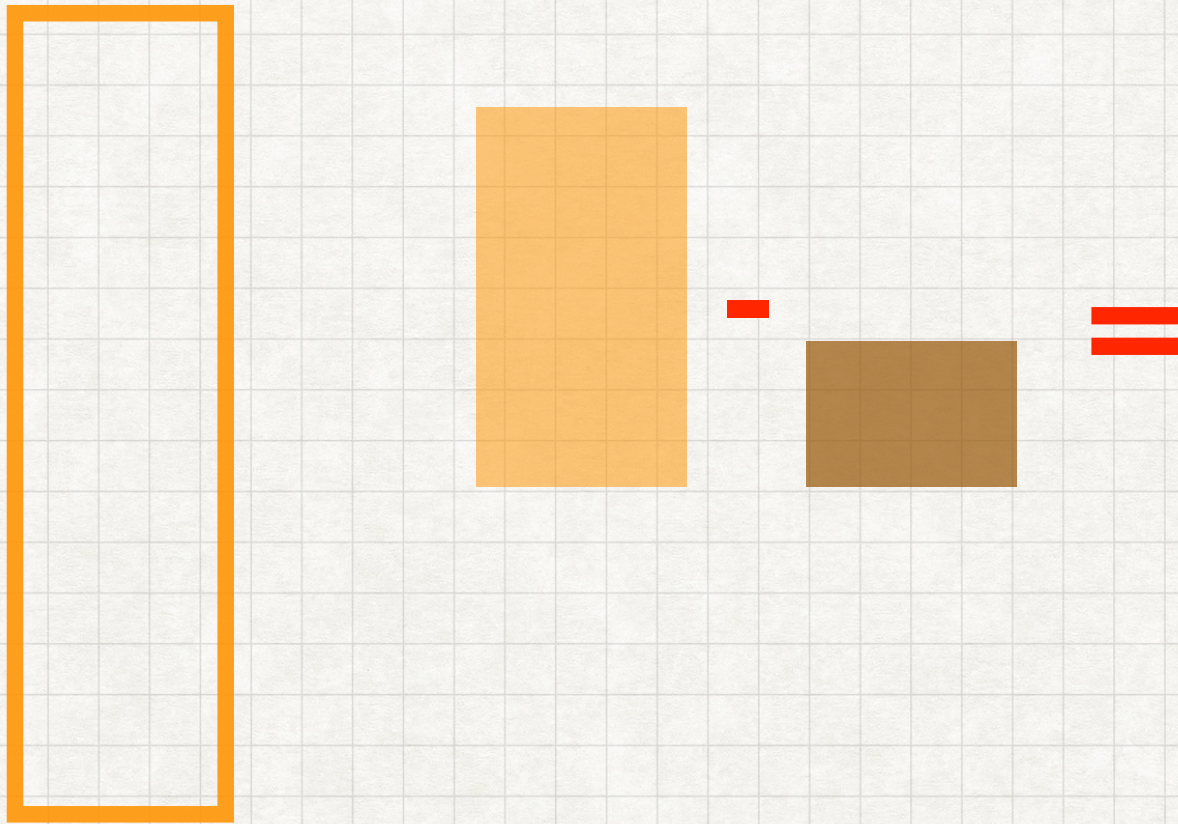
### Representación con Signo

- No ayuda mucho para operar con resta
- Tenemos 2 representaciones del 0 (+, -)



# ORGANIZACIÓN DEL PROCESADOR

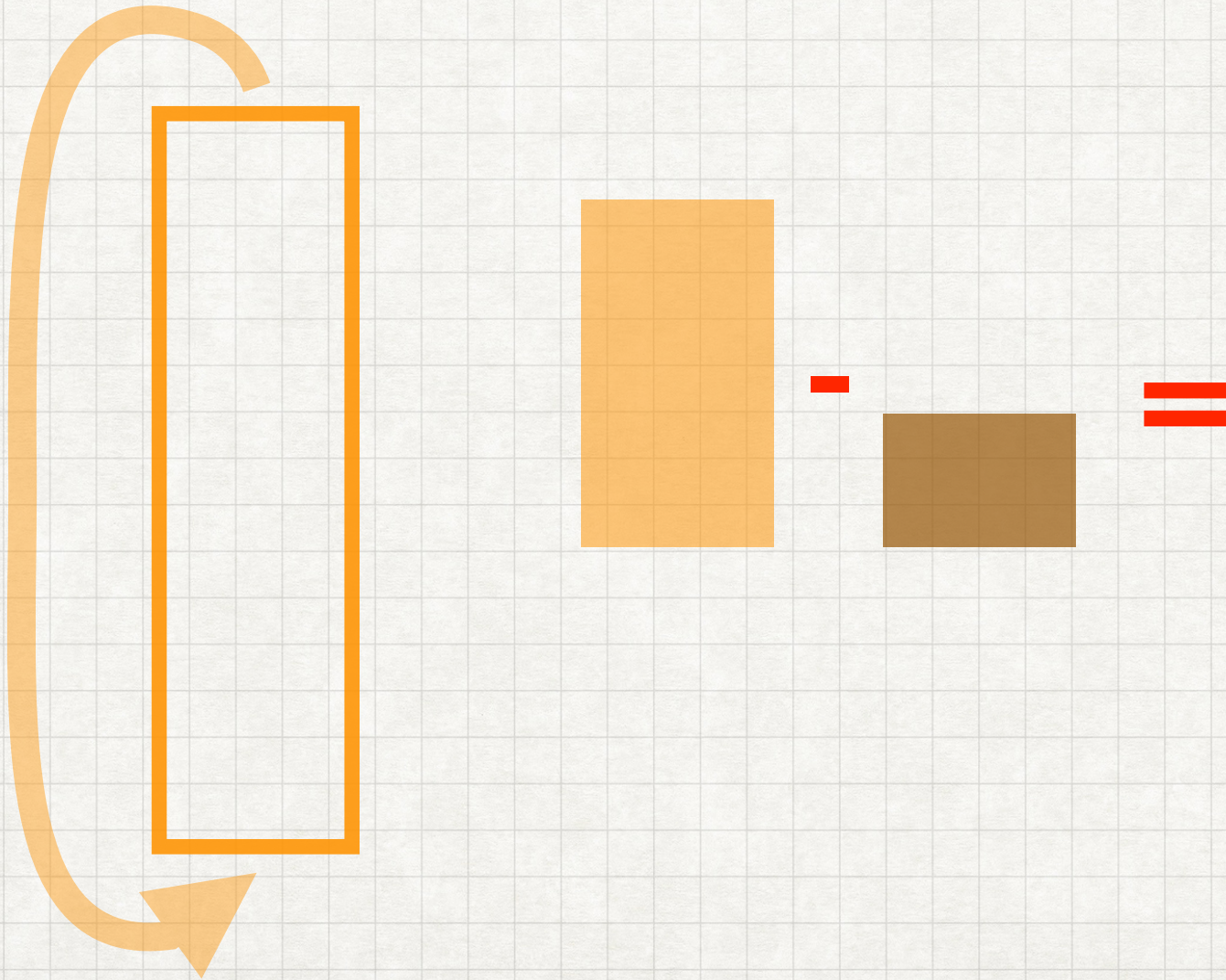
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

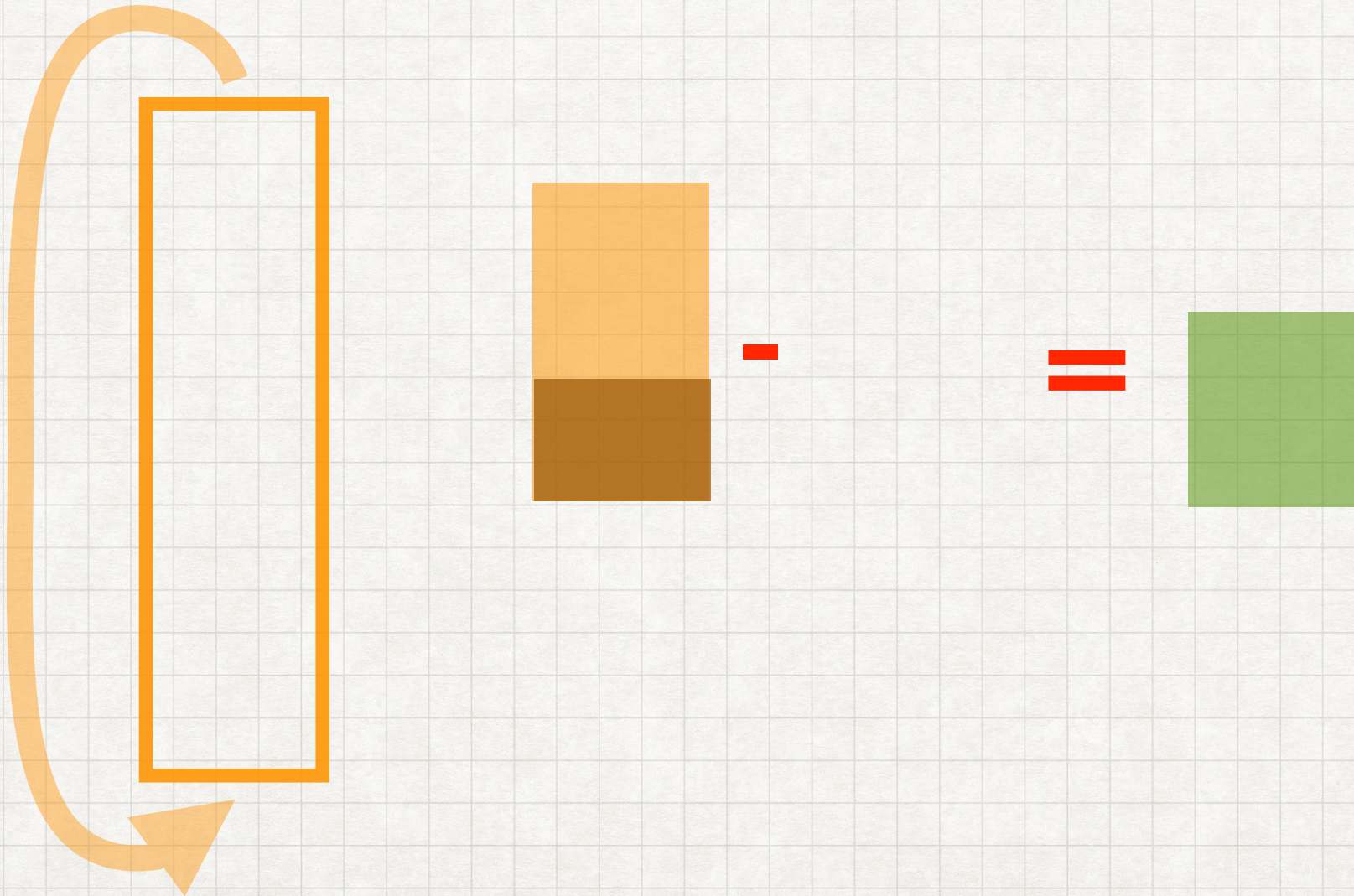
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

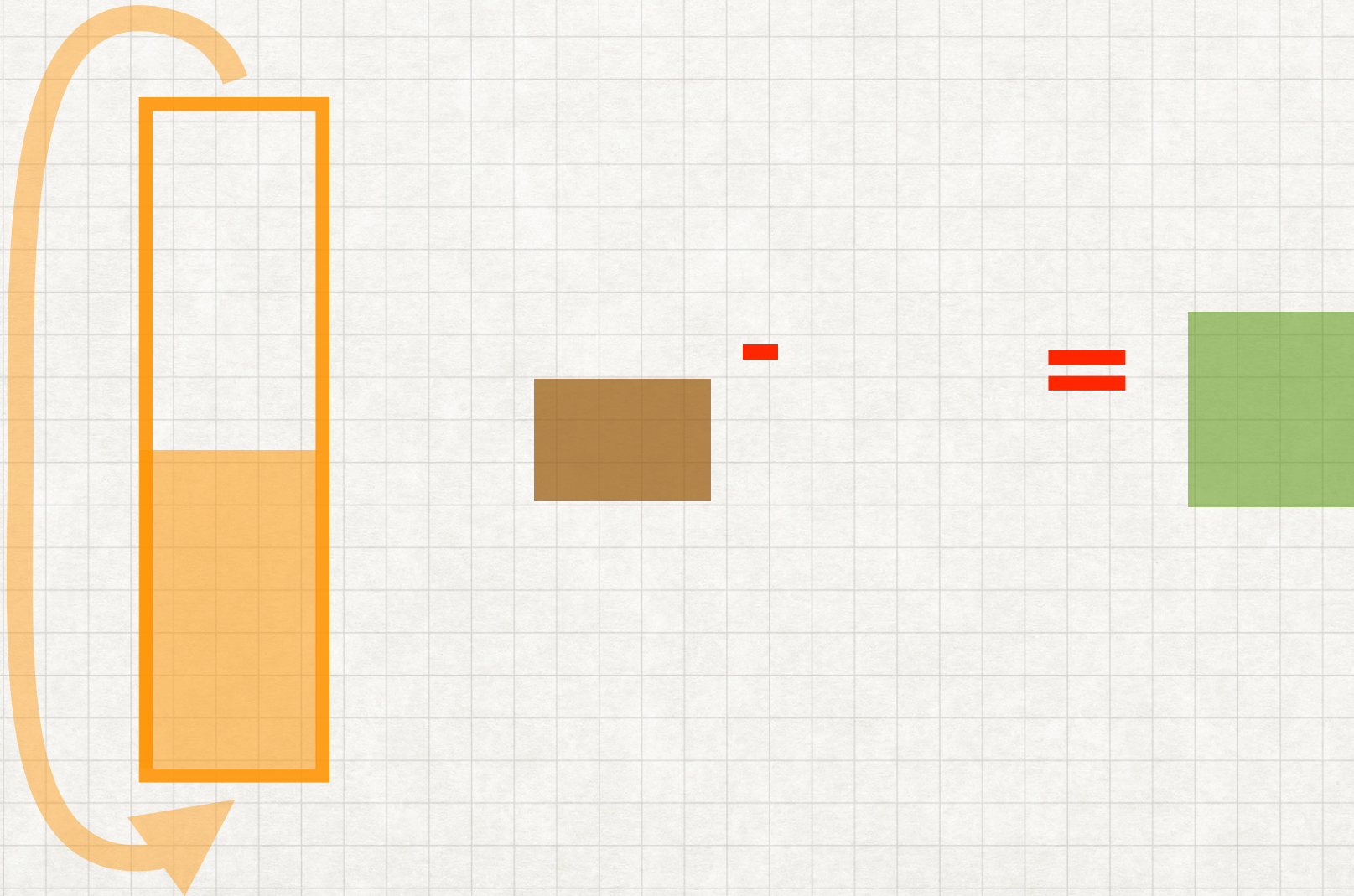
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

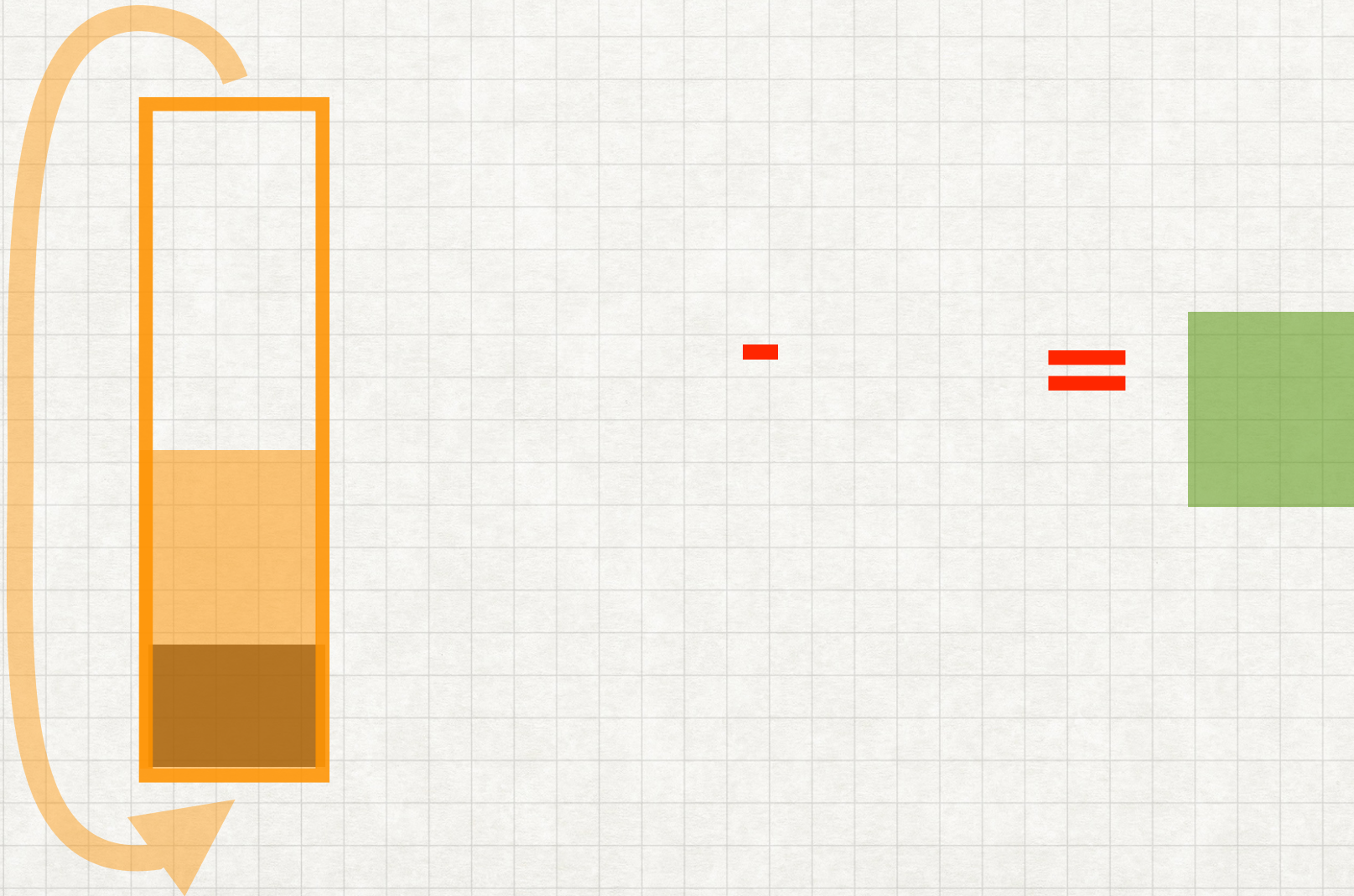
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

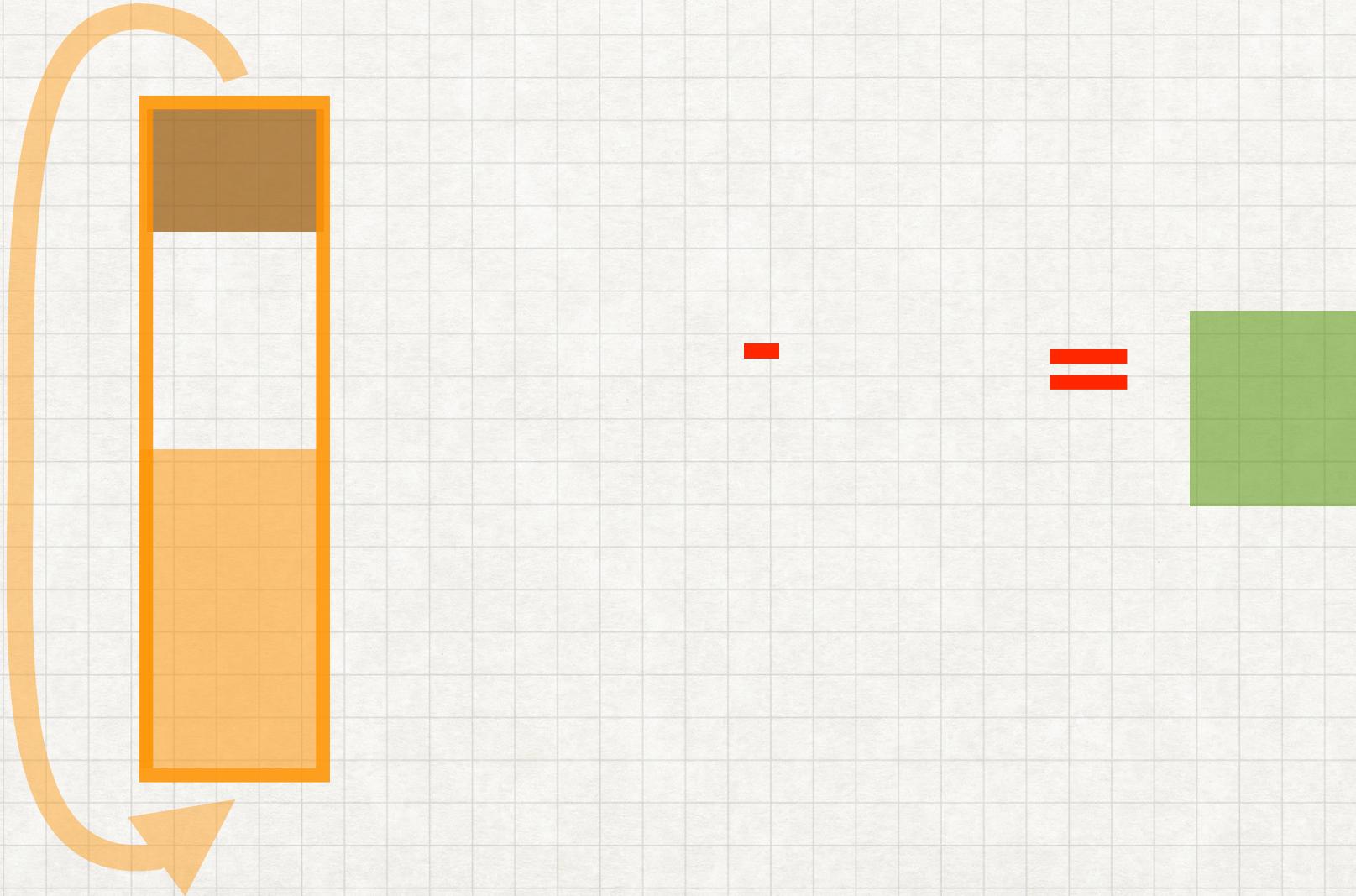
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

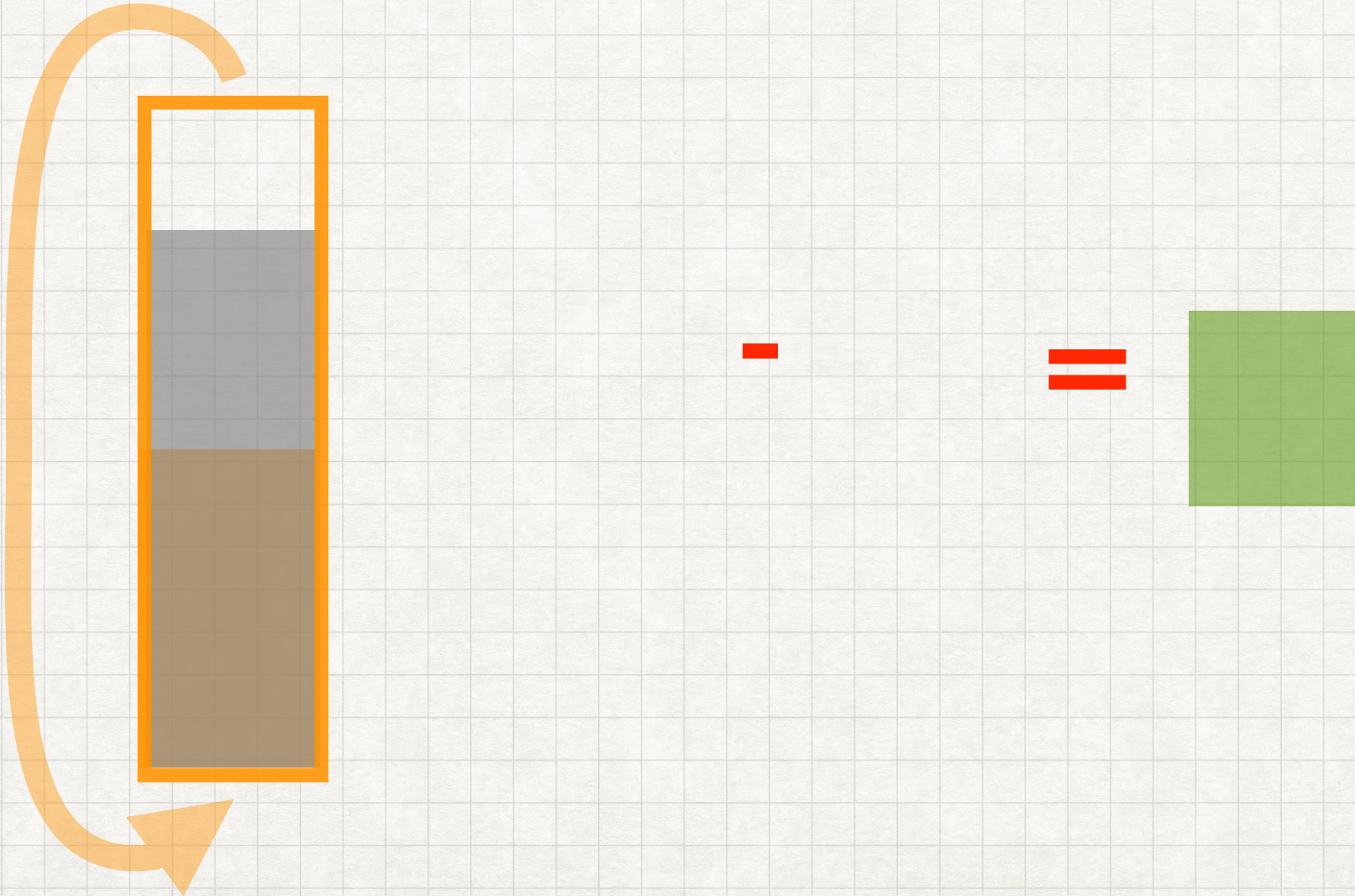
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

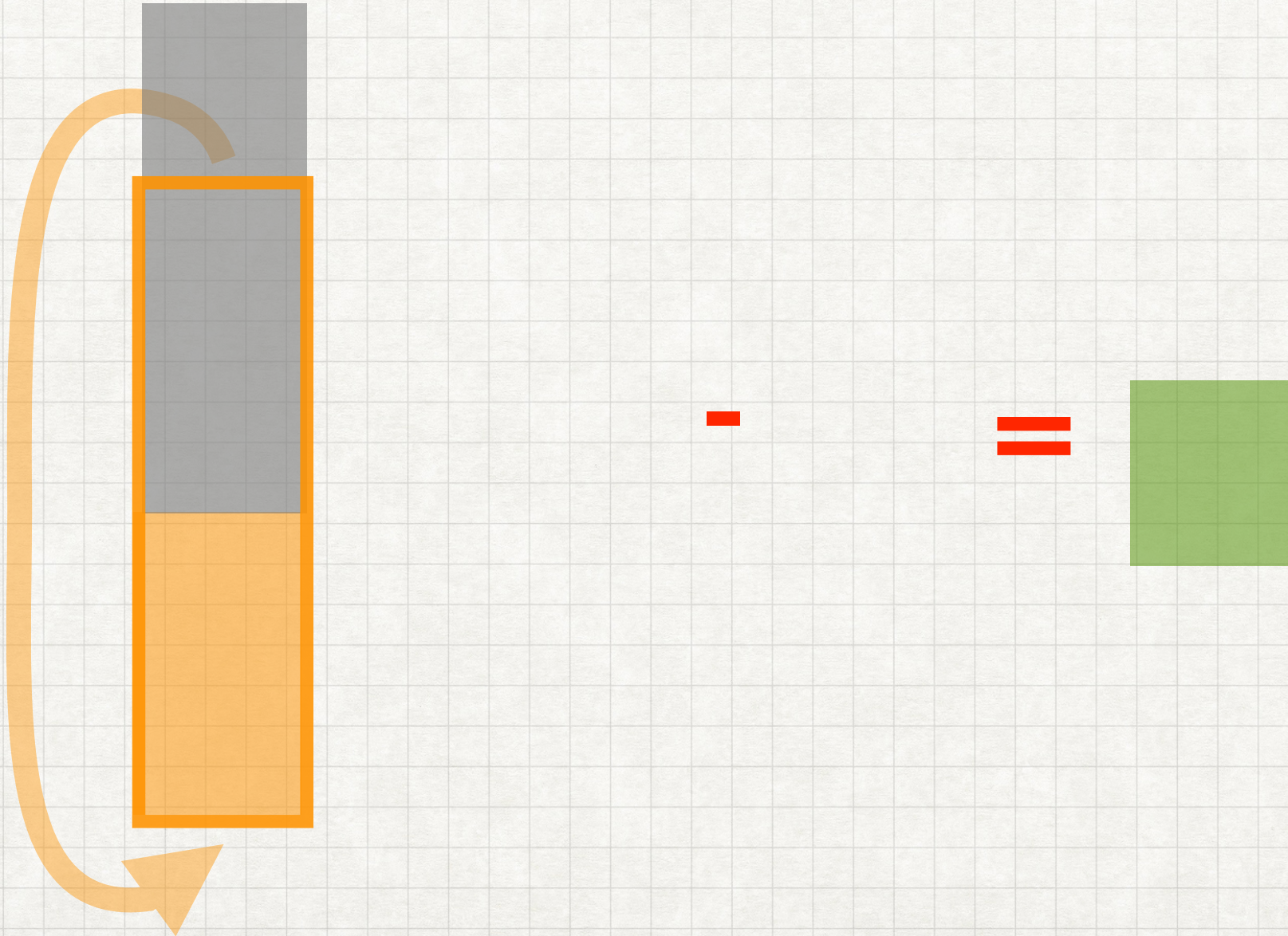
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

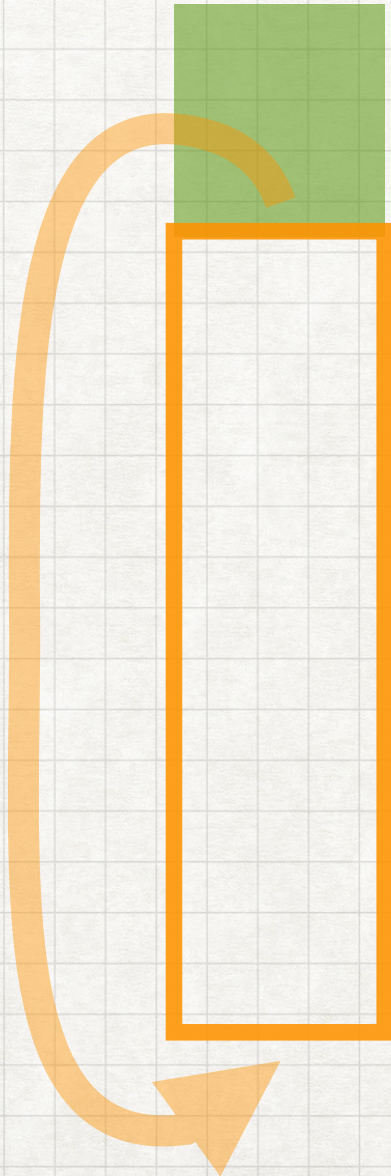
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS



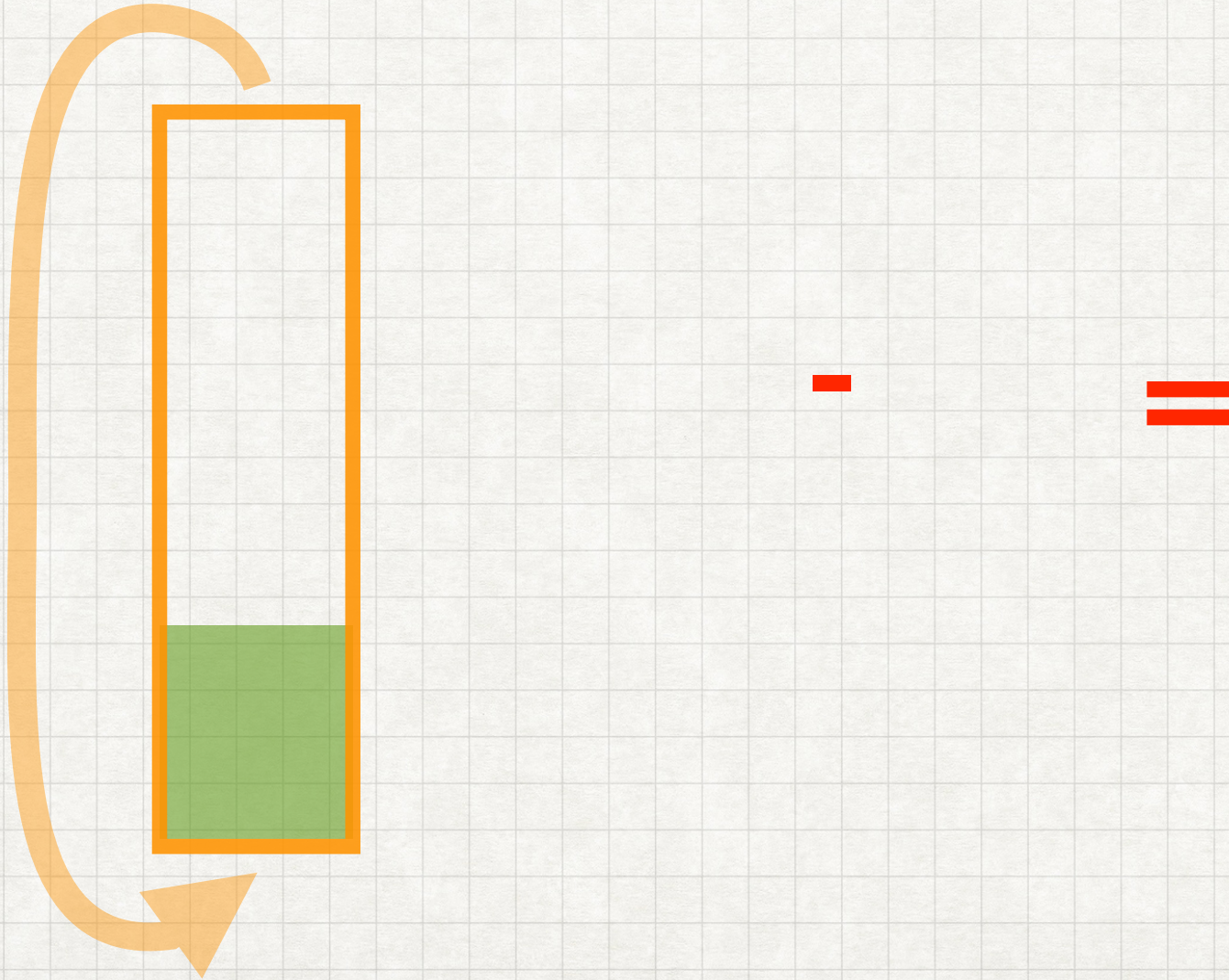
-

=



# ORGANIZACIÓN DEL PROCESADOR

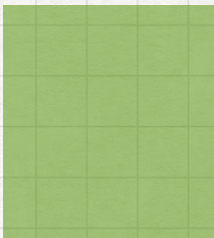
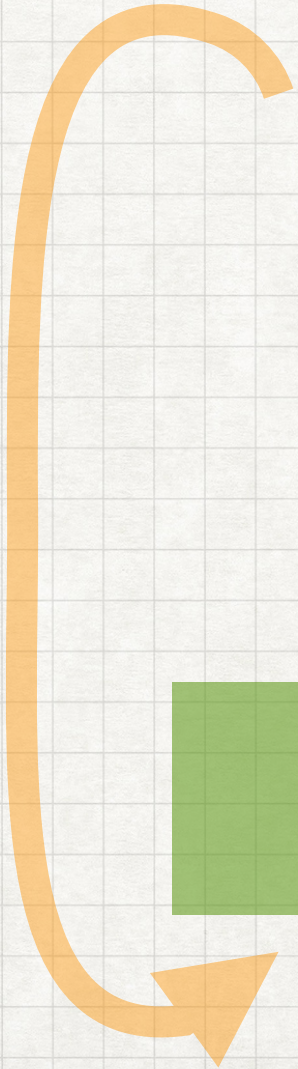
## REPRESENTACIÓN DE NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS



-

=



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS - COMPLEMENTO

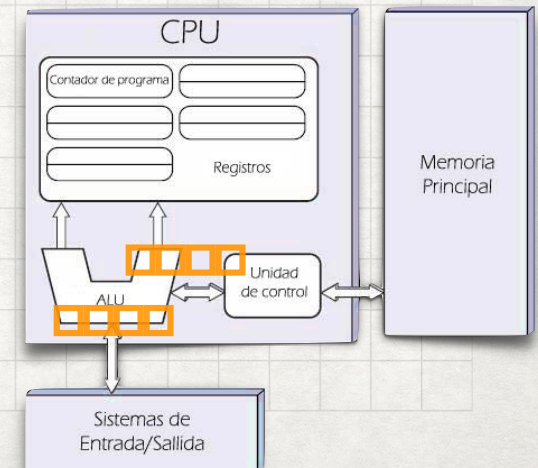
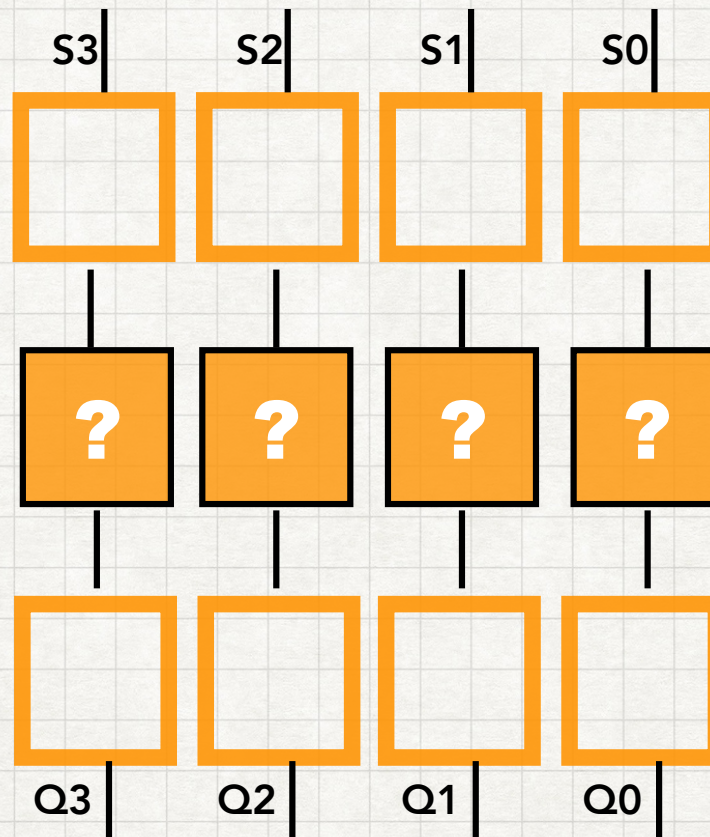
- Representación con Complemento a la base -1

$$-N = (\text{base}^{\text{tamaño}} - 1) - N$$



# ORGANIZACIÓN DEL PROCESADOR

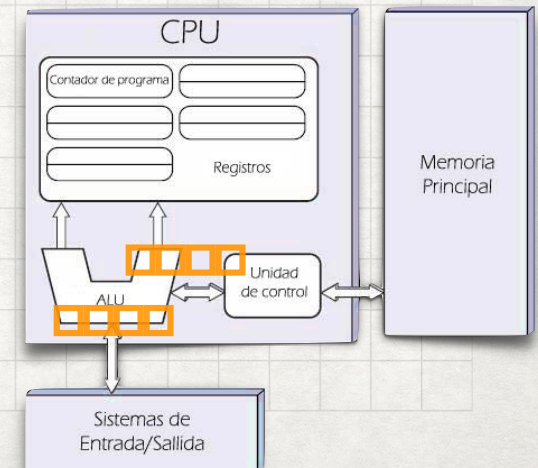
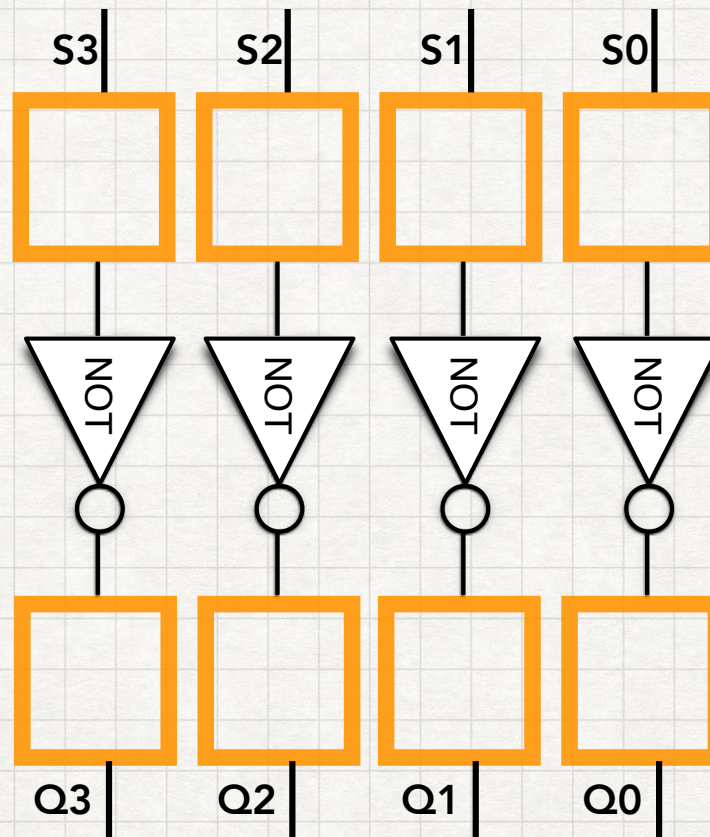
## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

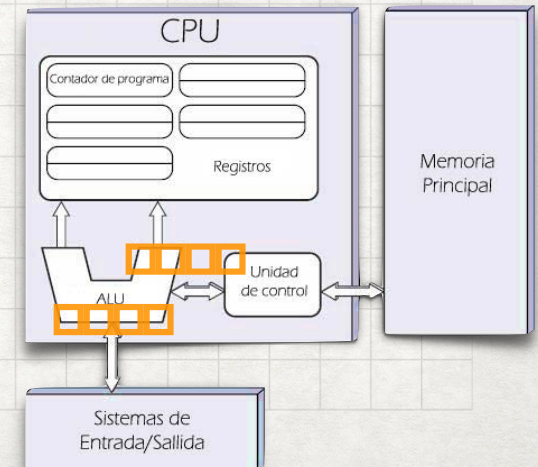
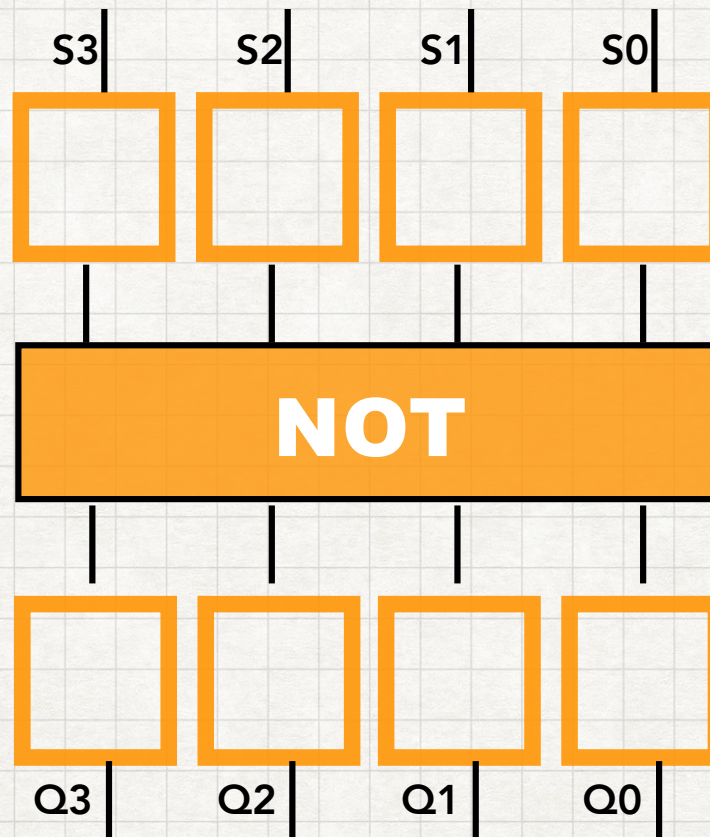
## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS - COMPLEMENTO

- Representación con Complemento a la base -1

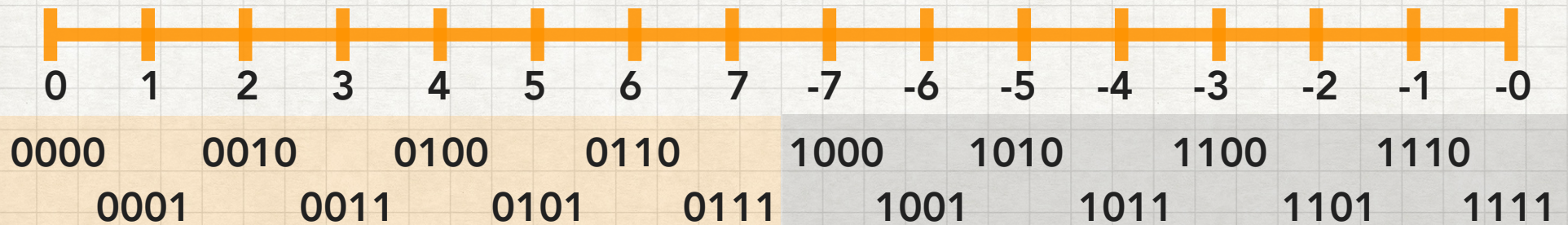
$$-N = (\text{base}^{\text{tamaño}} - 1) - N$$



# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$4 + 4 = 8$$





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$4 + 4 = 8$$

**Overflow**

¿cómo lo detectamos?





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$4 + 4 = 8$$

**Overflow**

¿cómo lo detectamos?

4





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$4 + 4 = 8$$

**Overflow**

¿cómo lo detectamos?

4

4



0000	0010	0100	0110	1000	1010	1100	1110
0001	0011	0101	0111	1001	1011	1101	1111



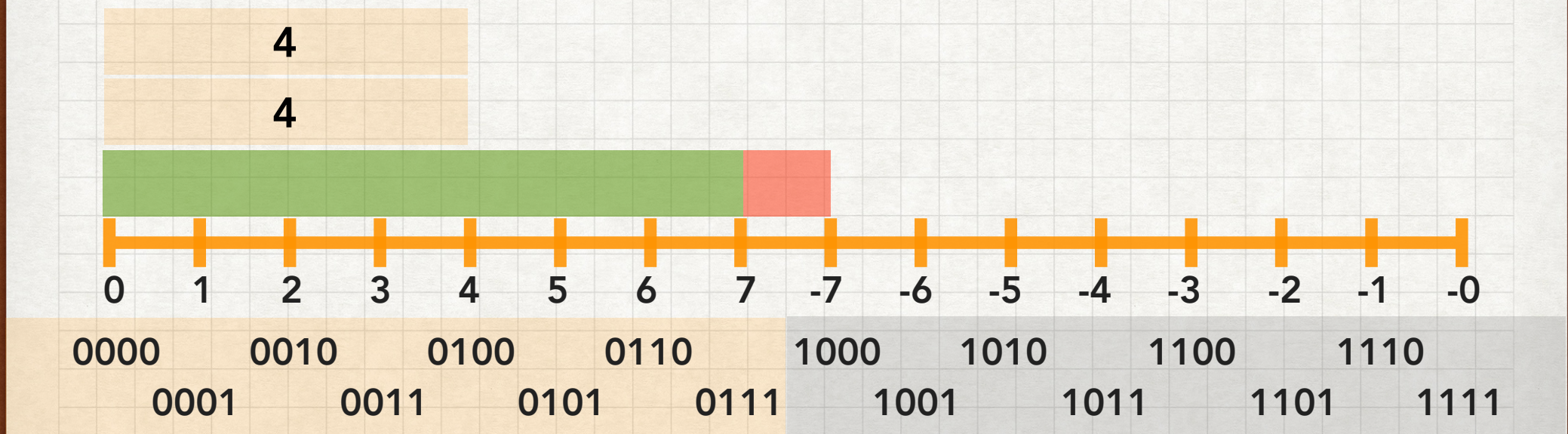
# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$4 + 4 = 8$$

**Overflow**

¿cómo lo detectamos?





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$(-4) + (-5) = -9$$

**Overflow**

¿cómo lo detectamos?

-4

-5





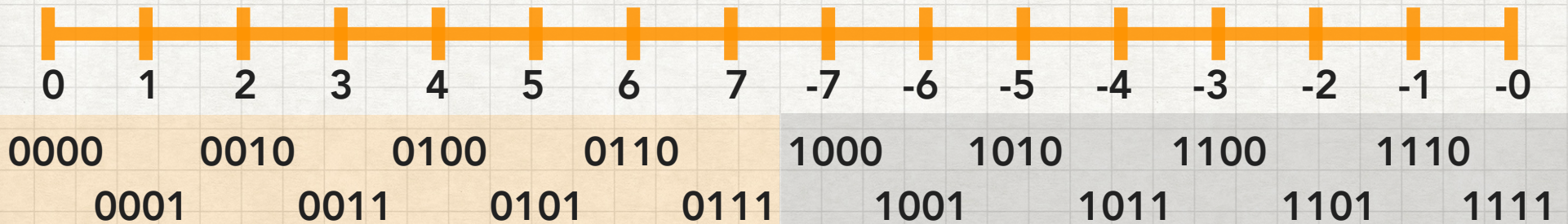
# ORGANIZACIÓN DEL PROCESADOR

# REPRESENTACIÓN DE NÚMEROS ENTEROS

$$(-4) + (-5) = -9$$

# Overflow

## ¿cómo lo detectamos?





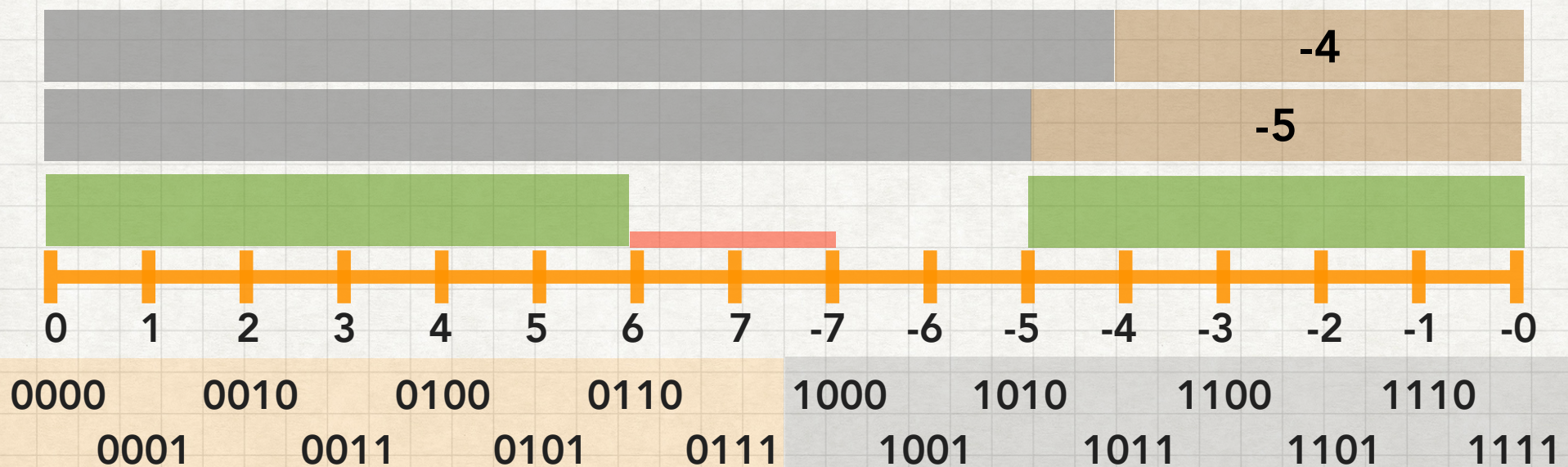
# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$(-4) + (-5) = -9$$

**Overflow**

¿cómo lo detectamos?





# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS

$$(-4) + (-5) = -9$$

**Overflow**

¿cómo lo detectamos?

**Overflow**

Si el CARRY en el bit de signo difiere del  
CARRY out, tenemos overflow !!.

-4

-5

0

1

2

3

4

5

6

7

-7

-6

-5

-4

-3

-2

-1

-0

0000

0010

0100

0110

1000

1010

1100

1110

0001

0011

0101

0111

1001

1011

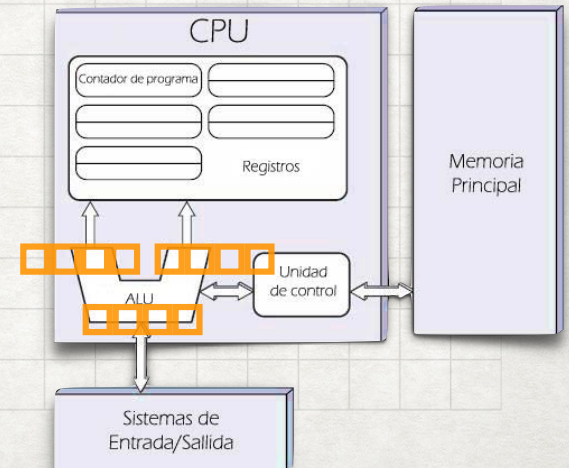
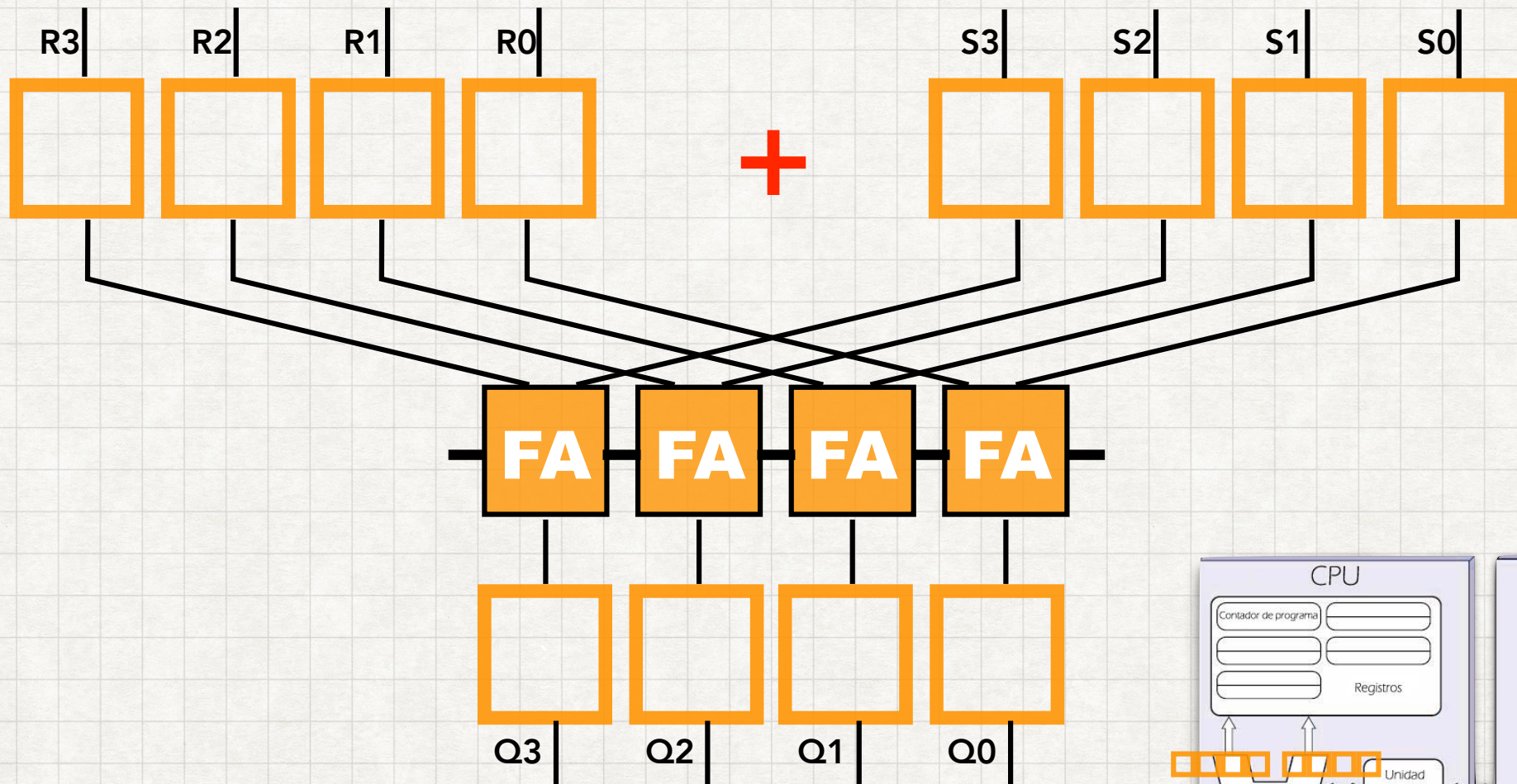
1101

1111



# ORGANIZACIÓN DEL PROCESADOR

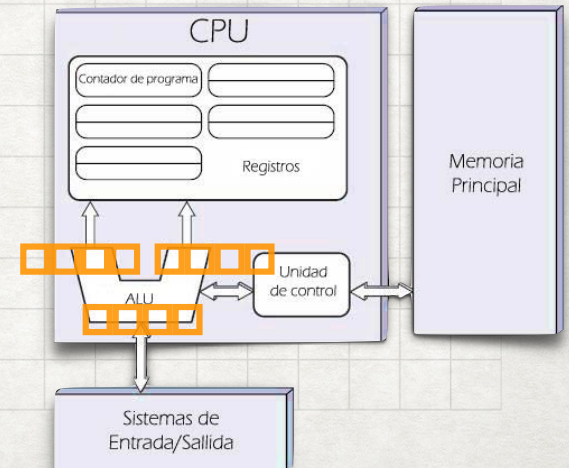
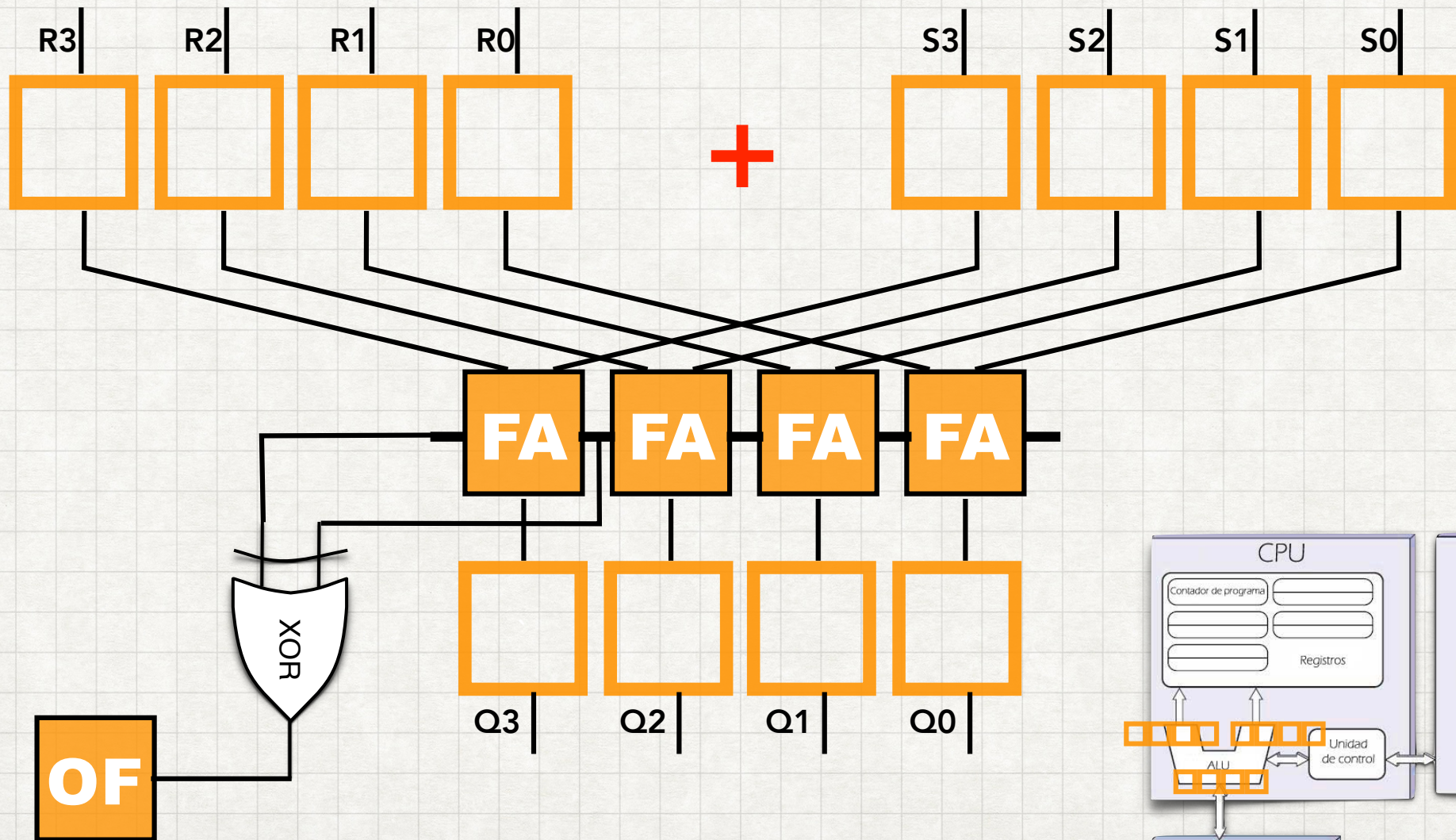
## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS



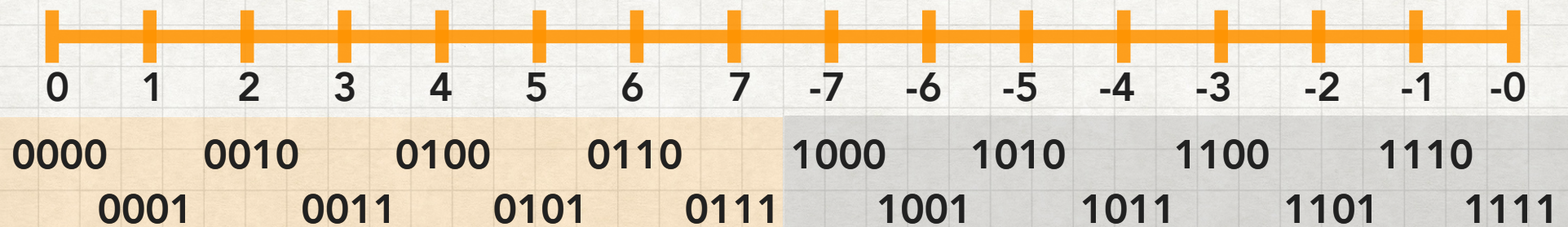


# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS COMPLEMENTO A LA BASE

- Representación con Complemento a la base

$$-N = (\text{base}^{\text{tamaño}} - 1) - N + 1$$



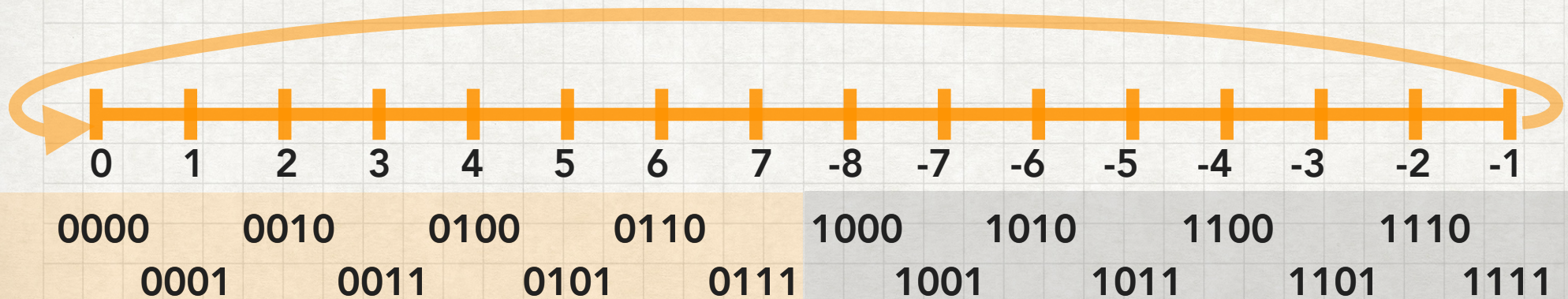


# ORGANIZACIÓN DEL PROCESADOR

## REPRESENTACIÓN DE NÚMEROS ENTEROS COMPLEMENTO A LA BASE

- Representación con Complemento a la base

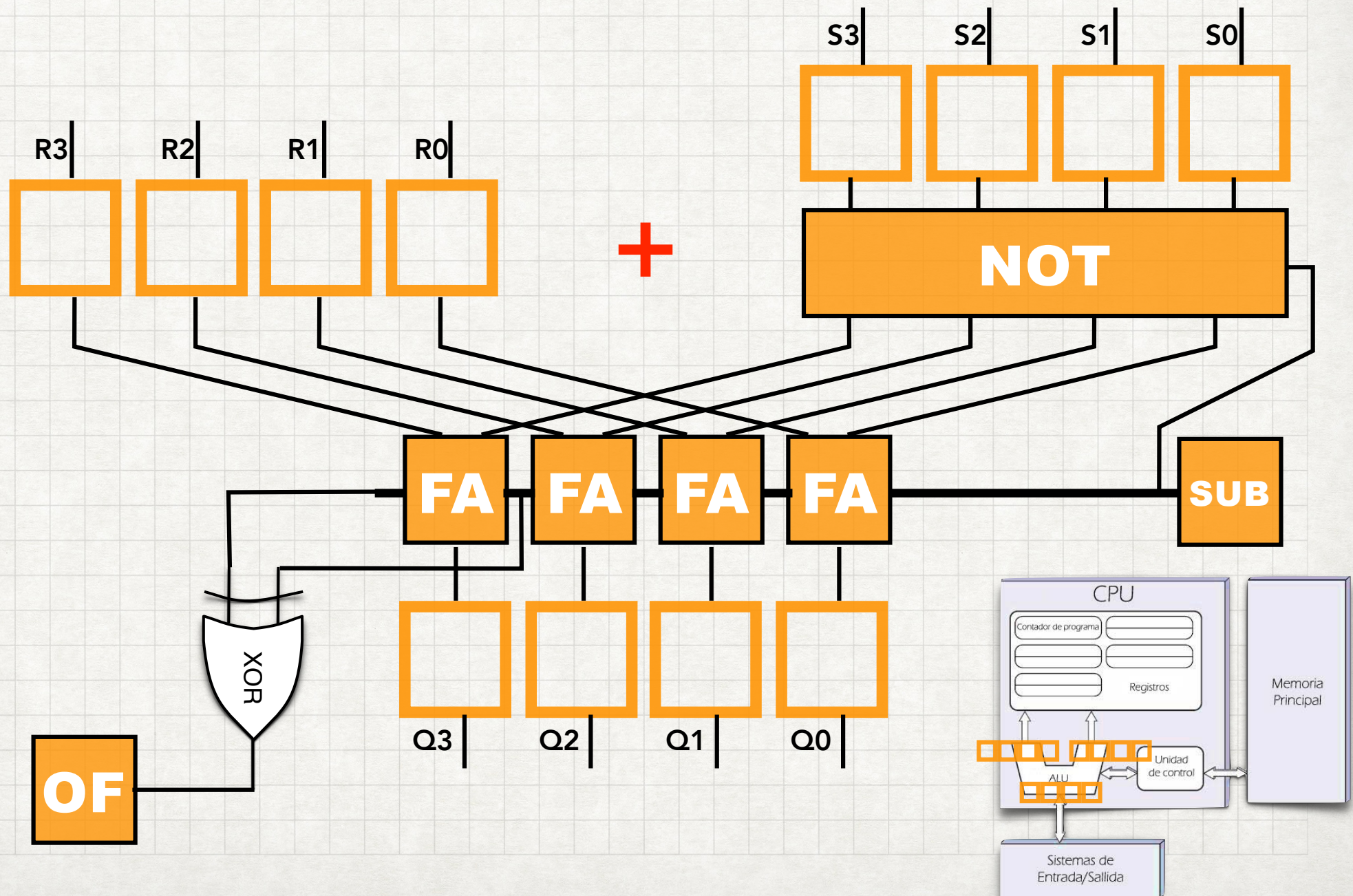
$$-N = (\text{base}^{\text{tamaño}} - 1) - N + 1$$





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

$$\begin{array}{r} * \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \end{array} \begin{array}{r} 11 \\ 7 \end{array}$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

Multiplicando

\*

Multiplicador

1

0

1

1

0

1

1

1

11

7



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

Multiplicando

Multiplicador

\*

1	0	1	1
0	1	1	1
<hr/>			
1	0	1	1

11

7



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

Multiplicando

\*

Multiplicador

	1	0	1	1
	0	1	1	1
	1	0	1	1
1	0	1	1	

11

7



# OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN



1	0	1	1
0	1	1	1

11

7

1 0 1 1

1 0 1 1

1 0 1 1



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

Multiplicando

Multiplicador

\*

1

0

1

1

11

0

1

1

1

7

1

0

1

1

1

0

1

1

1

0

1

1

0

0

0

0



# OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN



7

11

Four square tiles, each containing the digit 0. The tiles are arranged horizontally and are part of a larger grid. Each tile has a light gray background with a faint grid pattern and a brown border.



# ORGANIZACIÓN DEL PROCESADOR

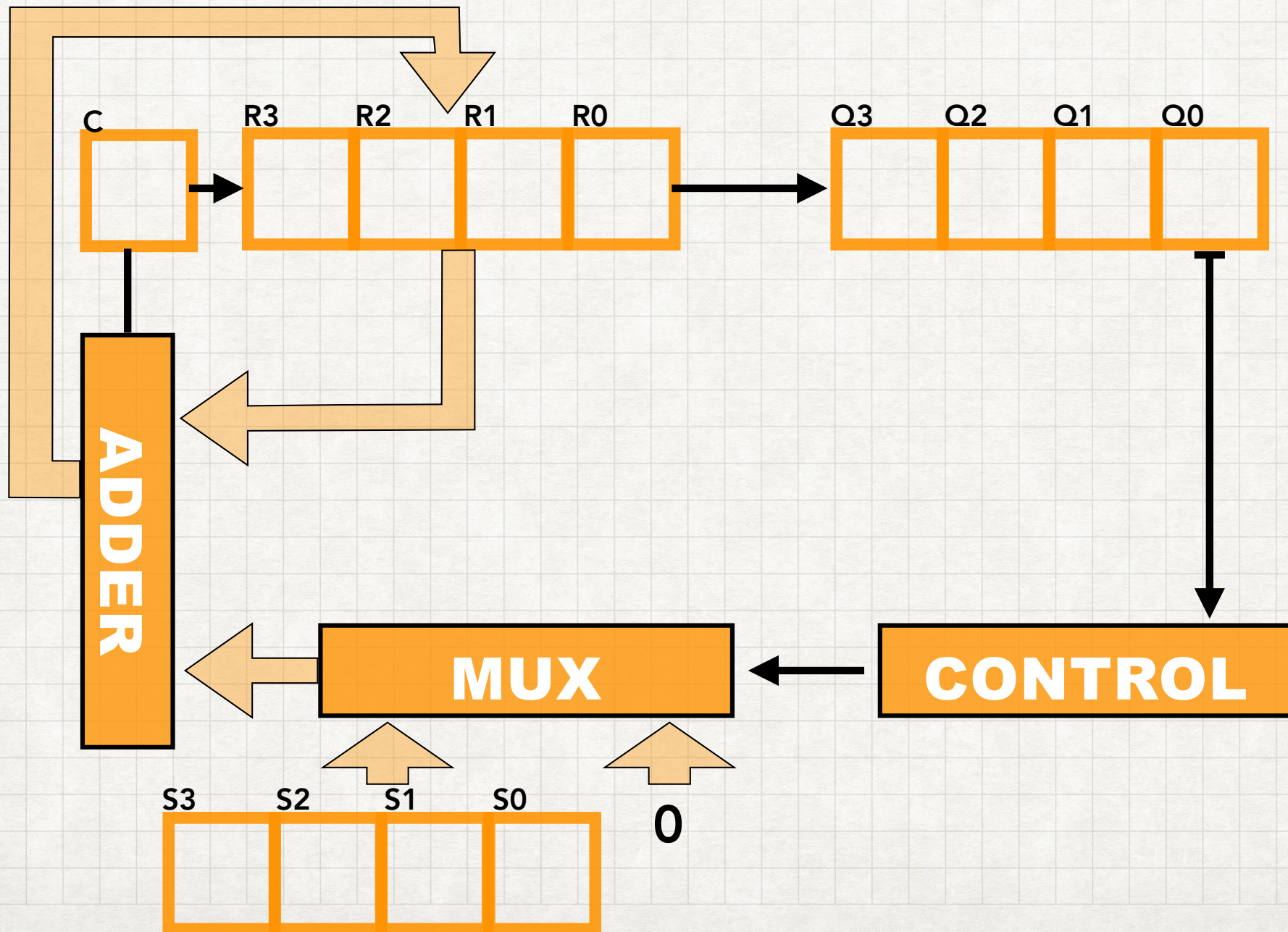
## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

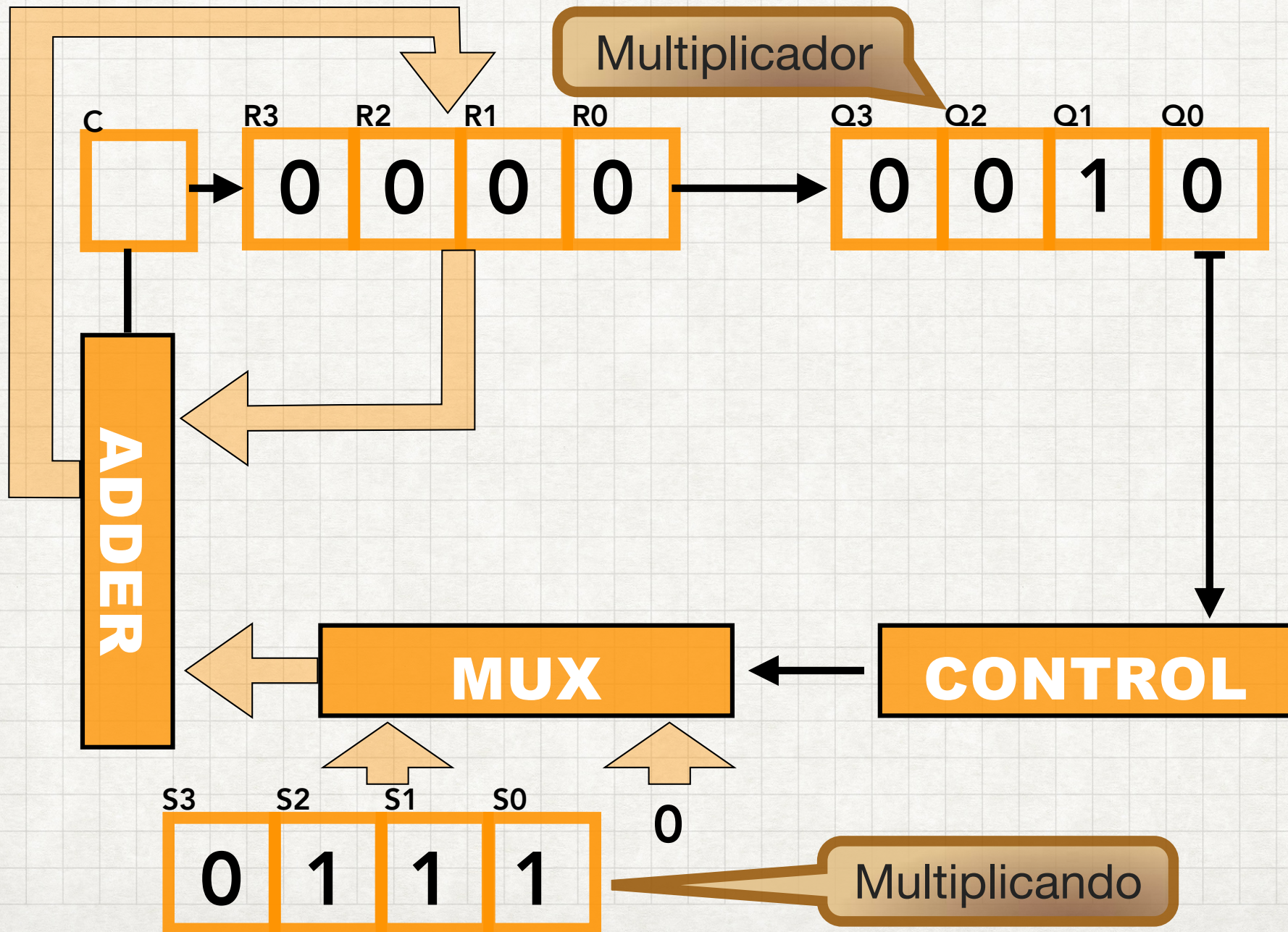
## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

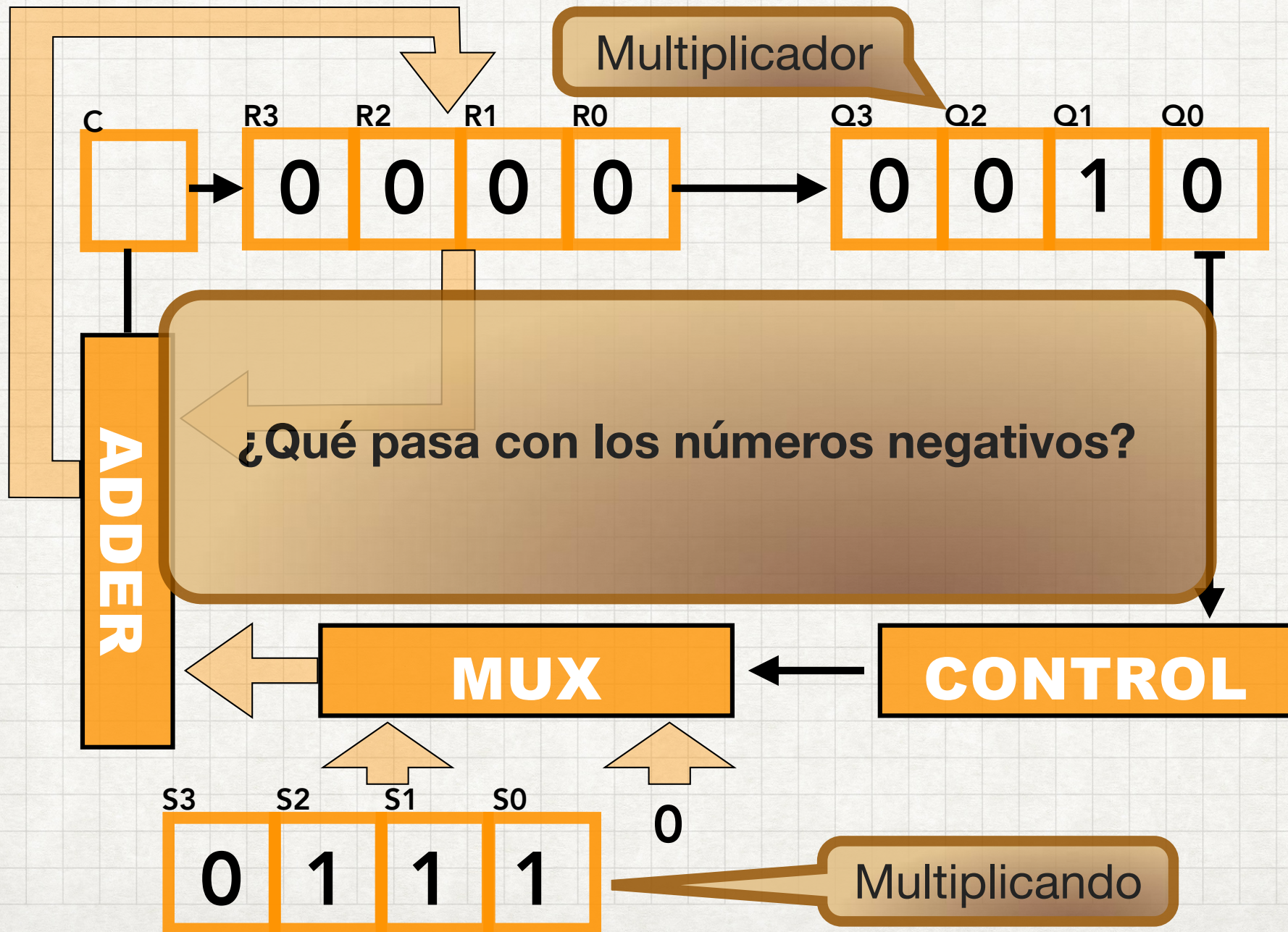
## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

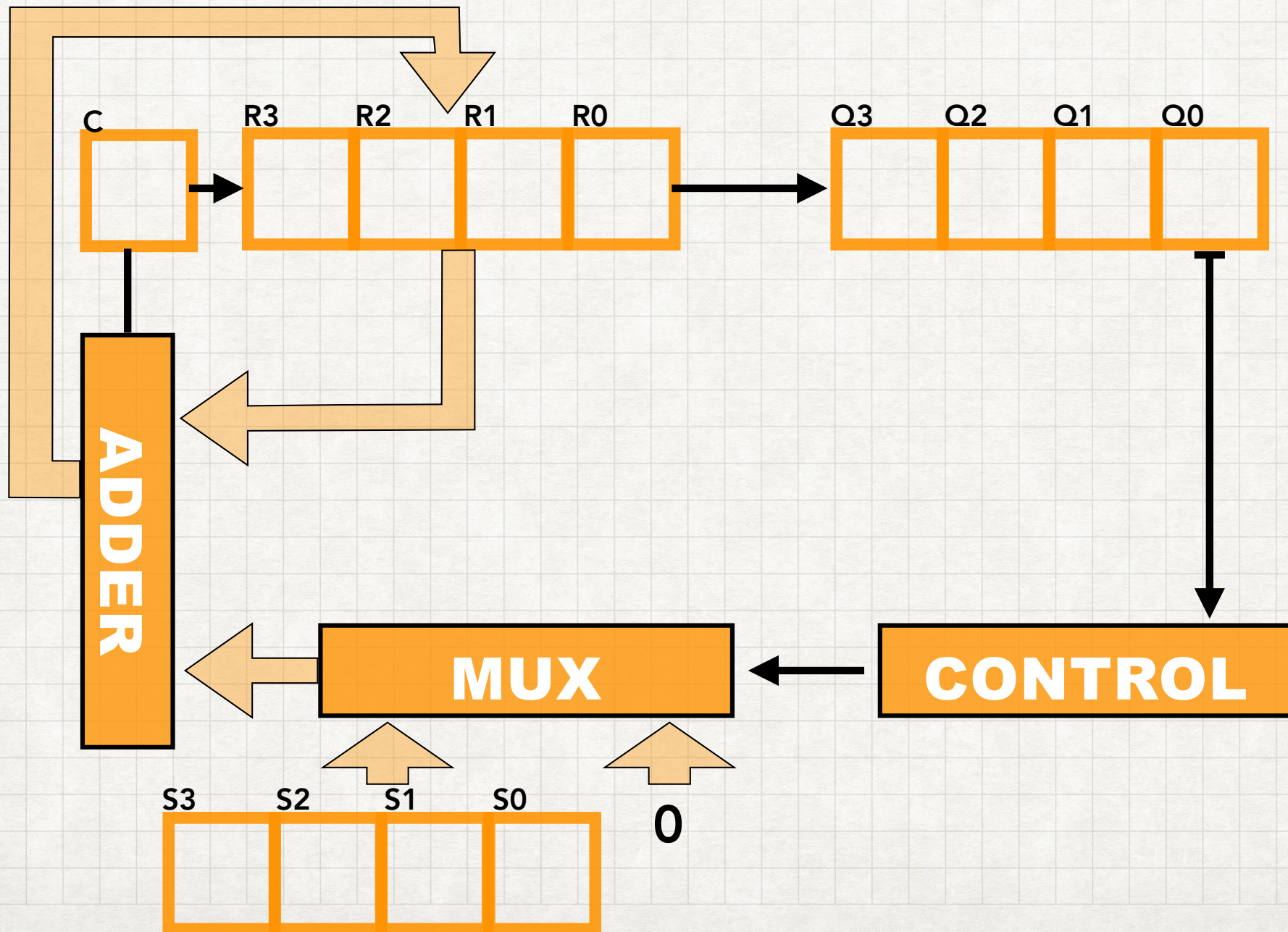
## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

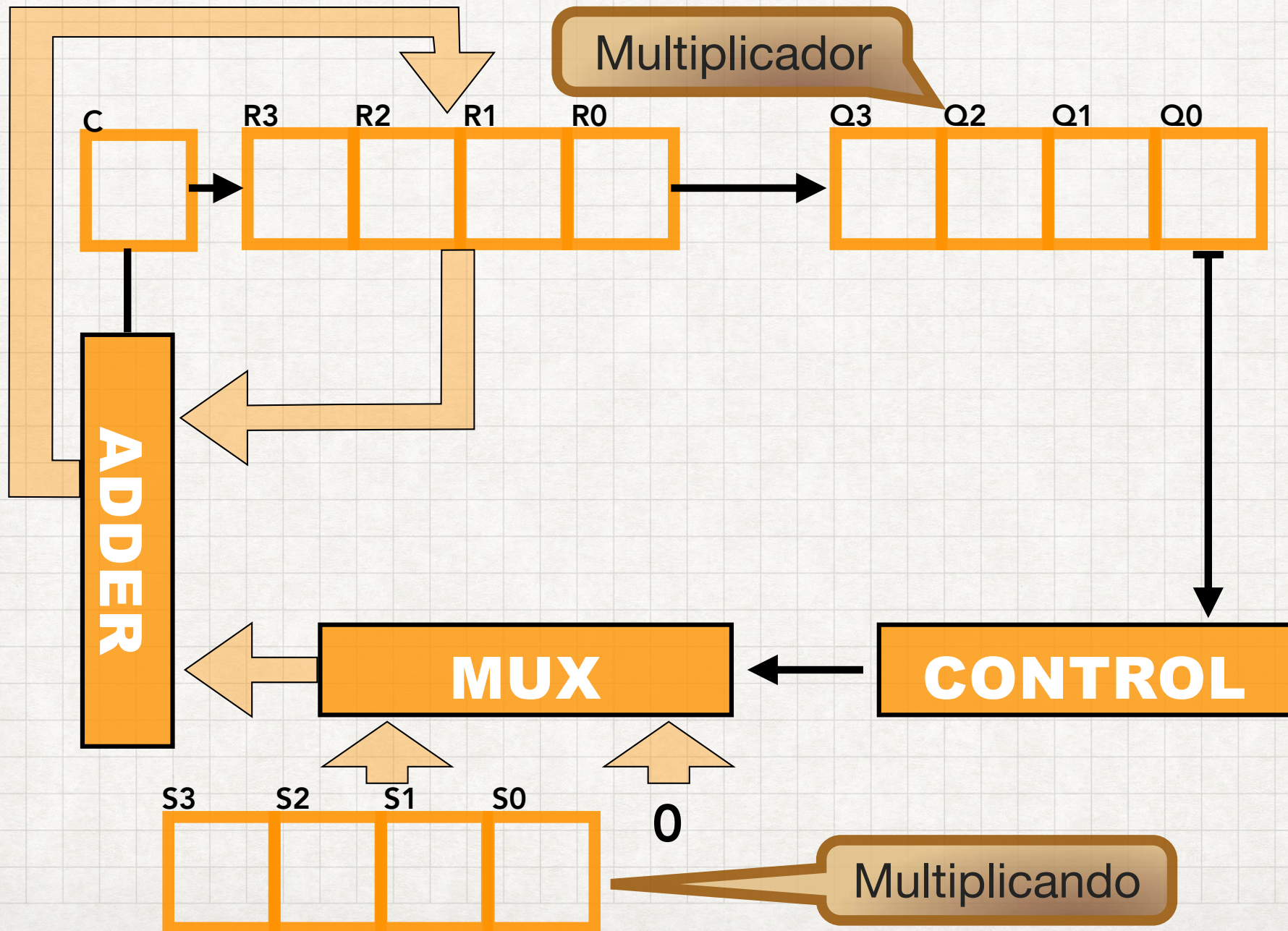
## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

### Multiplicación de enteros general (positivos y negativos):

- El resultado necesita un tamaño de bits equivalente a la suma de los operandos.
- El proceso calcula productos parciales y depende de los siguientes casos:
  - (a) **+** Multiplicando **\*** **+** Multiplicador : Se comienza con el producto parcial igual a 0 y el bit de *carry out* en 0. Se recorre el multiplicador de derecha a izquierda. Si es un 0 se *shiftea con el carry out a derecha* el producto parcial. Si es un 1, al producto parcial se le suma el multiplicando y luego se *shiftea con el carry out a derecha* el producto parcial.
  - (b) **-** Multiplicando **\*** **+** Multiplicador : El proceso es igual que el caso anterior, pero el bit de *carry out* comienza con 1, y en cada operación de *shift a derecha* se mantiene el *carry out* con 1 (se extiende el complemento del número)
  - (c) **+** Multiplicando **\*** **-** Multiplicador : se complementan ambos números y se procede como en (b)
  - (d) **-** Multiplicando **\*** **-** Multiplicador : se complementan ambos números y se procede como en (a)



# ORGANIZACIÓN DEL PROCESADOR

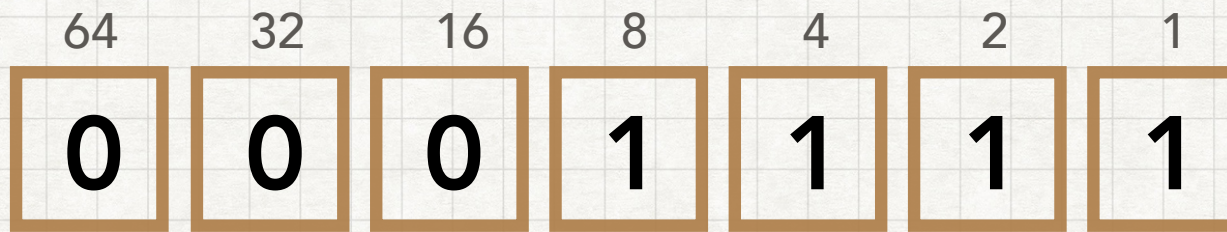
## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

La idea del algoritmo de Booth es la de no operar por casos y la de tratar de realizar la menor cantidad posible de sumas (es más costoso que *shiftear*)



# ORGANIZACIÓN DEL PROCESADOR

OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH





# ORGANIZACIÓN DEL PROCESADOR

OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

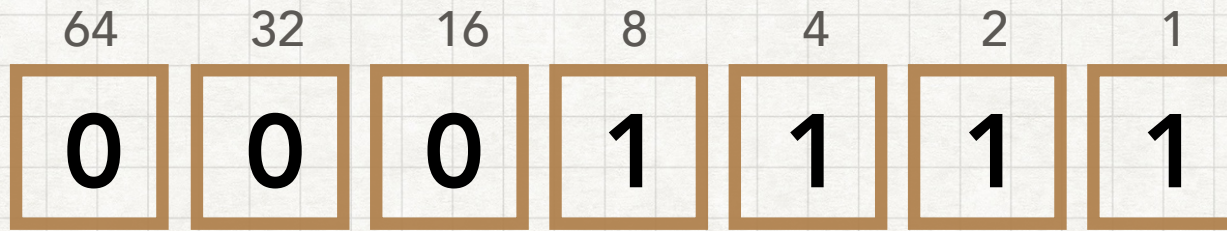


$$8 + 4 + 2 + 1 = 15$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH



$$8 + 4 + 2 + 1 = 15$$

$$16 + 0 + 0 + 0 - 1 = 15$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

64	32	16	8	4	2	1
0	0	0	1	1	1	1

$$8 + 4 + 2 + 1 = 15$$

$$16 + 0 + 0 + 0 - 1 = 15$$

0	0	1	0	0	0	-1
---	---	---	---	---	---	----



# ORGANIZACIÓN DEL PROCESADOR

OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH





# ORGANIZACIÓN DEL PROCESADOR

OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH



$$16 + 8 + 4 + 2 + 0 = 30$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

64	32	16	8	4	2	1
0	0	1	1	1	1	0

$$16 + 8 + 4 + 2 + 0 = 30$$

$$32 + 0 + 0 + 0 - 2 + 0 = 30$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

64	32	16	8	4	2	1
0	0	1	1	1	1	0

$$16 + 8 + 4 + 2 + 0 = 30$$

$$32 + 0 + 0 + 0 - 2 + 0 = 30$$

0	1	0	0	0	-1	0
---	---	---	---	---	----	---



# ORGANIZACIÓN DEL PROCESADOR

OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	1

$$64 + 32 + 16 + 0 + 4 + 2 + 1 = 119$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	1

$$64 + 32 + 16 + 0 + 4 + 2 + 1 = 119$$

$$128 + 0 + 0 - 16 + 8 + 0 + 0 - 1 = 119$$



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULT. BOOTH

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	1

$$64 + 32 + 16 + 0 + 4 + 2 + 1 = 119$$

$$128 + 0 + 0 - 16 + 8 + 0 + 0 - 1 = 119$$

1	0	0	-1	1	0	0	-1
---	---	---	----	---	---	---	----

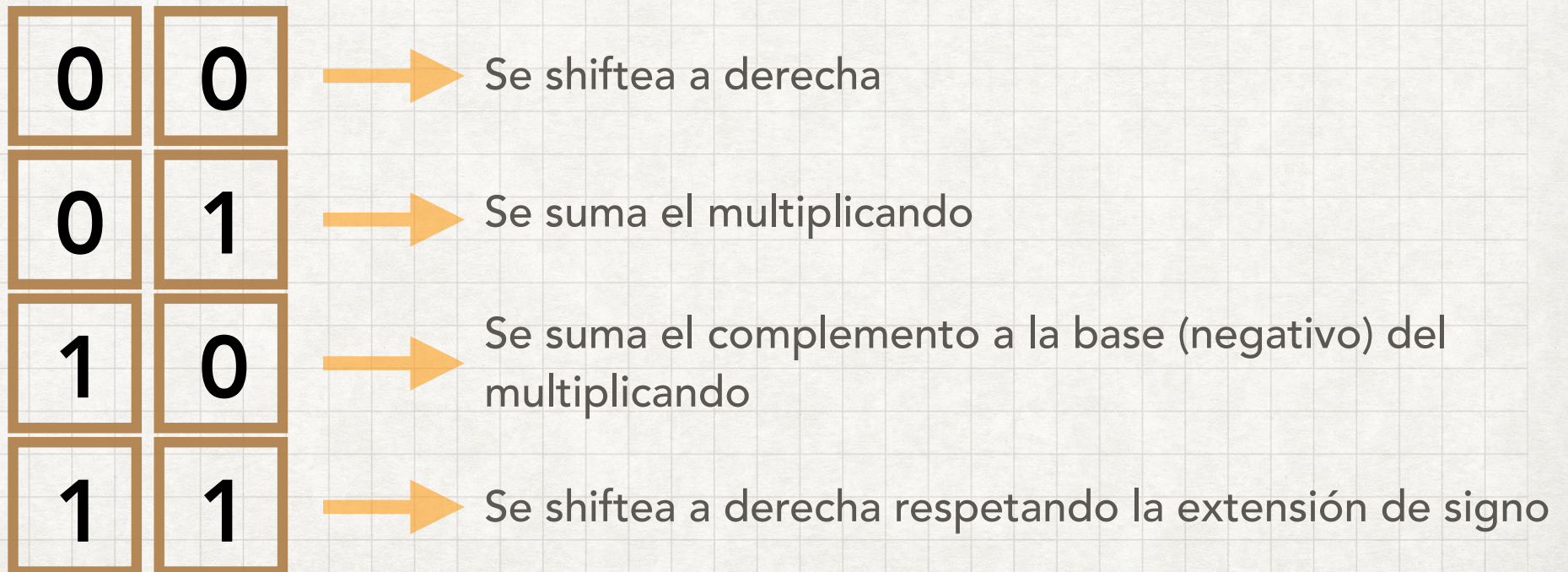


# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

### Algoritmo de Booth:

- El resultado necesita un tamaño de bits equivalente a la suma de los operandos.
- El proceso calcula productos parciales de manera similar que el algoritmo general pero en vez de mirar el último bit del **multiplicador** mira los últimos 2 y de acuerdo a los siguientes casos opera sobre el producto parcial:





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - MULTIPLICACIÓN

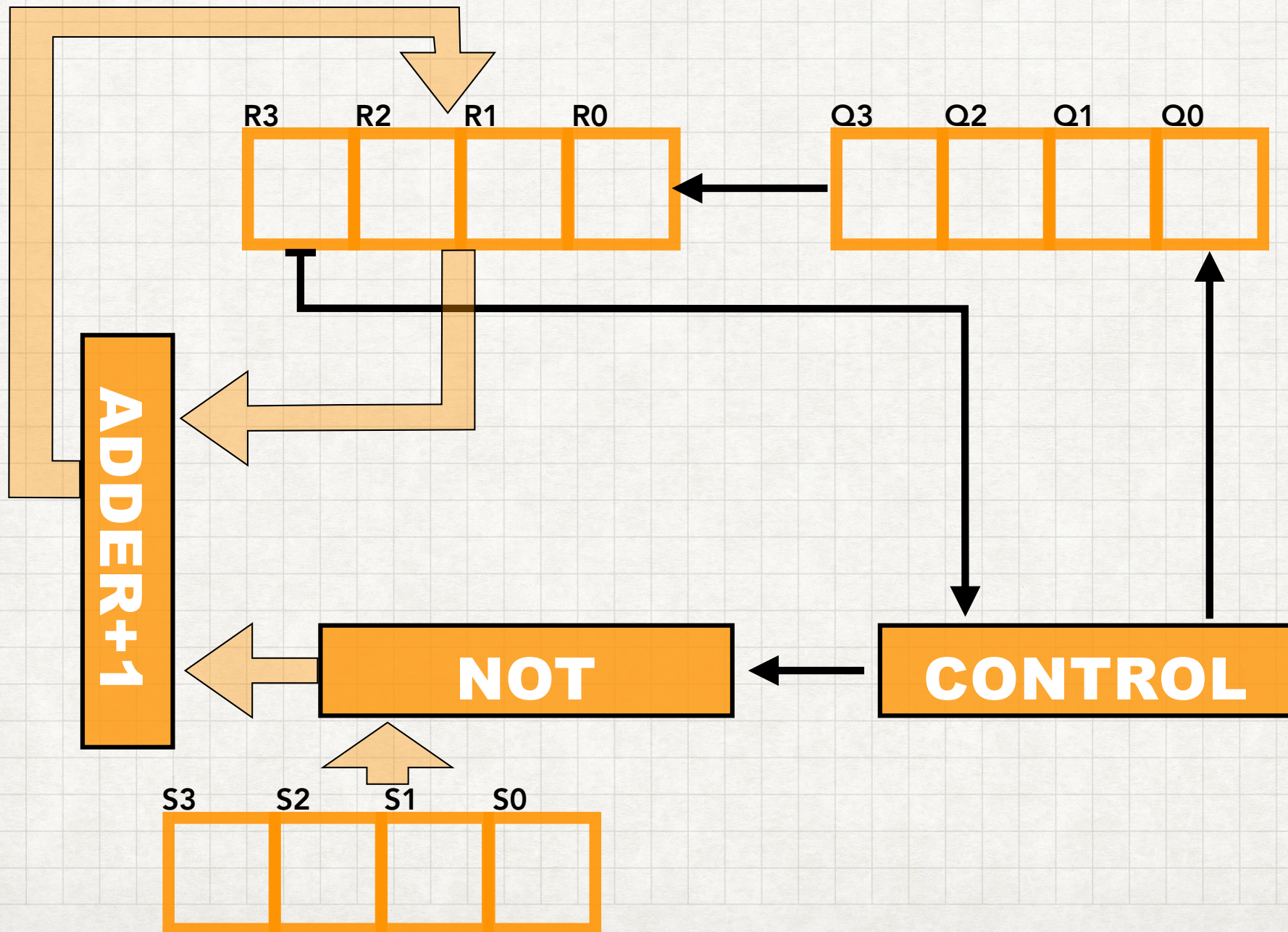
### Algoritmo de Booth:

- La cantidad de operaciones (sumas) optimizadas es proporcional a la cantidad de secuencias contiguas de 1 en la representación del multiplicador.
- Existe un caso donde empeora.



# ORGANIZACIÓN DEL PROCESADOR

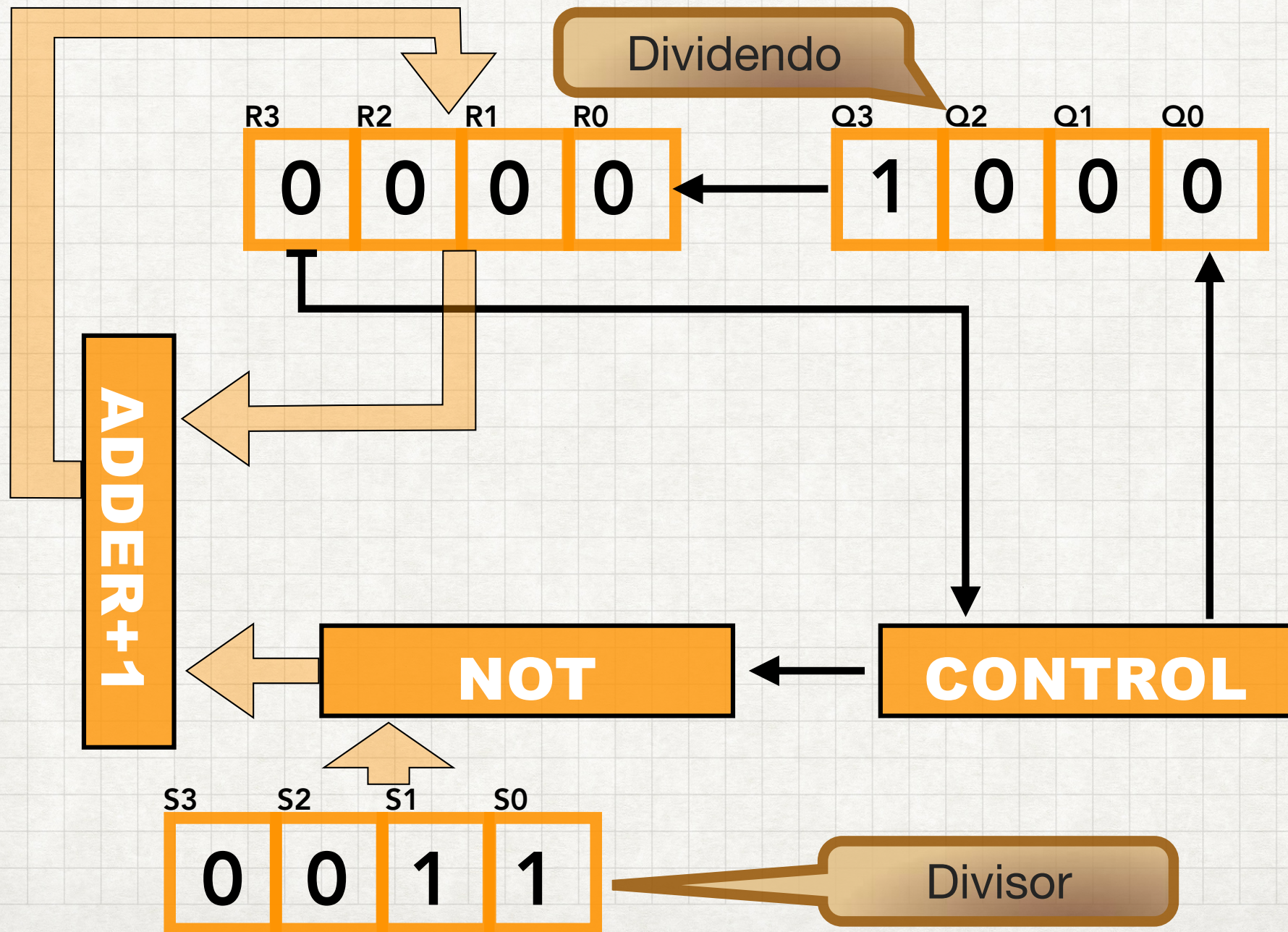
## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN





# ORGANIZACIÓN DEL PROCESADOR

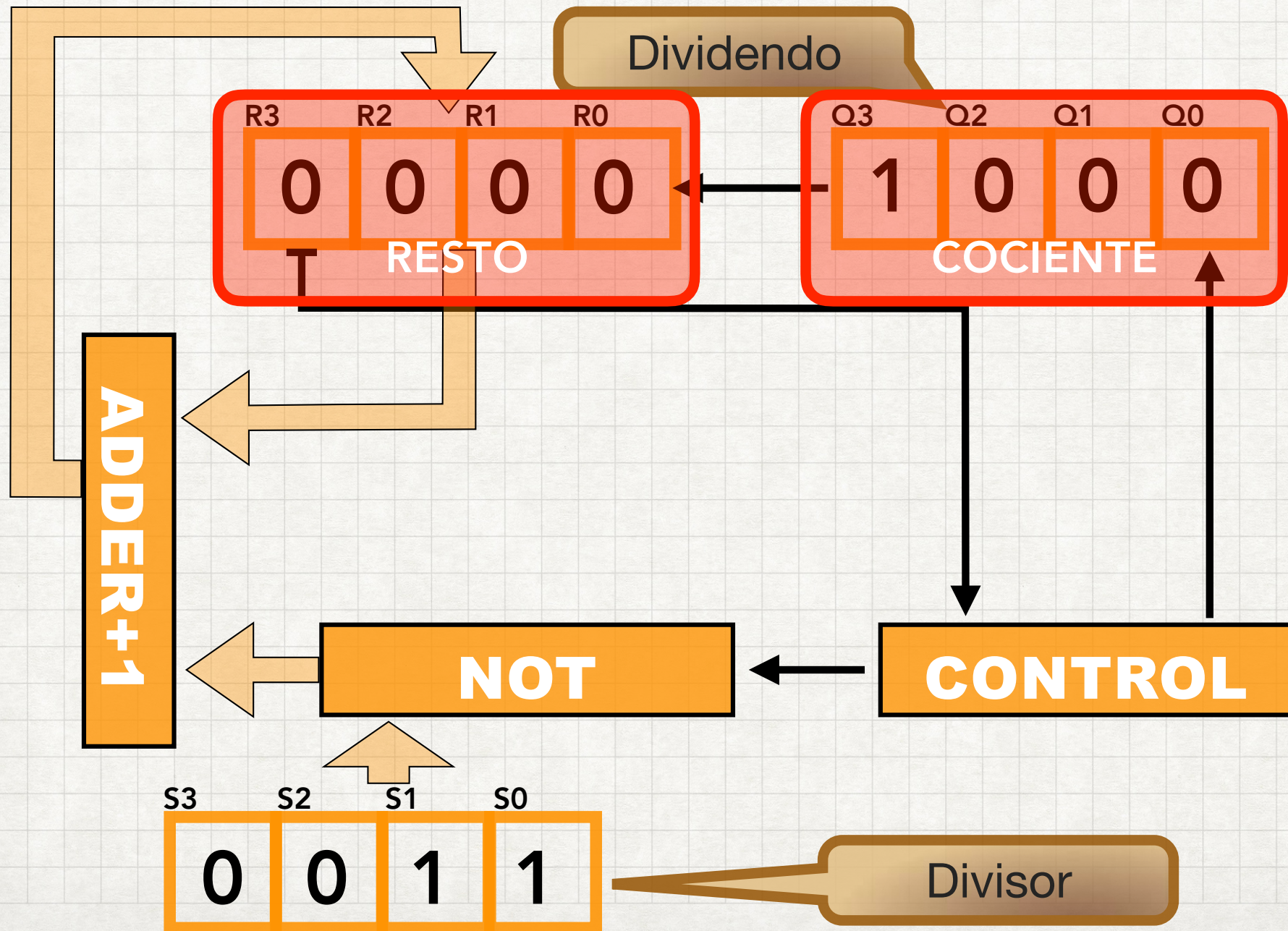
## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN





# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN

### Algoritmo de la división:

Repetir  $n$  (*cantidad de bits*) veces los siguientes pasos:

- (1) *Shiftear a izquierda* una vez los registro correspondientes al **Dividendo** (Q) con 0 y el **registro auxiliar** (R) con el bit más significativo de Q.
- (2) Dejar en el registro auxiliar el resultado de restarle al mismo el **Divisor** (S). Notar que se suma el **Comp.Base** de (S)
- (3) Si el signo (*bit más* significativo del registro auxiliar R) es 1 (negativo), cambiar el *bit menos* significativo de **Dividendo** ( $Q_0$ ) a 0 y restaurar el **registro auxiliar** sumando el **Divisor** (S), sino cambiar el *bit menos* significativo de **Dividendo** ( $Q_0$ ) a 1.



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN

### Algoritmo de la división: (restauración)

Repetir  $n$  (*cantidad de bits*) veces los siguientes pasos:

- (1) *Shiftear a izquierda* una vez los registro correspondientes al **Dividendo** (Q) con 0 y el **registro auxiliar** (R) con el bit más significativo de Q.
- (2) Dejar en el registro auxiliar el resultado de restarle al mismo el **Divisor** (S). Notar que se suma el **Comp.Base** de (S)
- (3) Si el signo (*bit más* significativo del **registro auxiliar** R) es 1 (negativo), cambiar el *bit menos* significativo de **Dividendo** ( $Q_0$ ) a 0 y restaurar el **registro auxiliar** sumando el **Divisor** (S), sino cambiar el *bit menos* significativo de **Dividendo** ( $Q_0$ ) a 1.



# ORGANIZACIÓN DEL PROCESADOR

## OPERACIONES CON NÚMEROS ENTEROS - DIVISIÓN

**Algoritmo de la división: (sin restauración)**

Repetir  $n$  (*cantidad de bits*) veces los siguientes 2 pasos:

- (1) Si el signo de registro auxiliar (R) es 0. *Shiftear* (R) y (Q) y dejar en R el resultado de  $R-S$ . Sino *shiftear* (R) y (Q) y dejar en R el resultado de  $R+S$ .
- (2) Si el signo (*bit más* significativo del registro auxiliar R) es 0 (negativo), cambiar el *bit menos* significativo de Dividendo ( $Q_0$ ) a 1, sino cambiarlo a 0.

Finalmente, si el signo de R es 1 (negativo). Sumo S a R.

**Si el signo de (R) es 0, hacemos  $2R-S$ . Si es 1 hacemos  $2R+S$ . Cambiando en cada caso el correspondiente  $bit_0$  de Q.**