



# Técnicas Digitales II

## Diseño de una CPU

Dr.Ing.Steiner Guillermo



# Introducción

## **Sistemas Digitales**

Conjunto de dispositivos que permiten la generación, procesamiento y almacenamiento de datos digitales

Se dividen en:

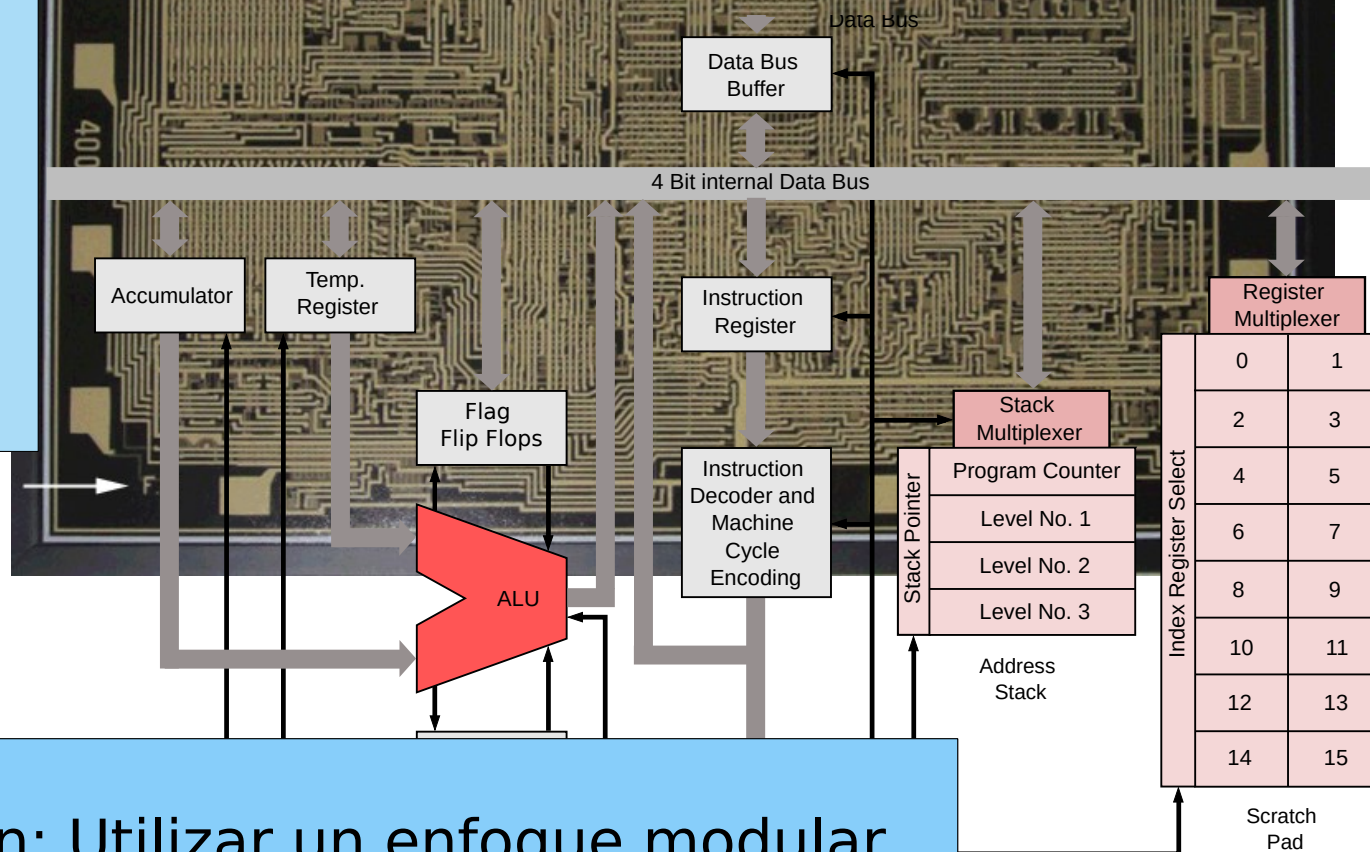
**Sistemas digitales combinacionales:** la salida depende de la entrada.

**Sistemas digitales secuenciales:** la salida depende de la entrada y de estados anteriores.

# Introducción

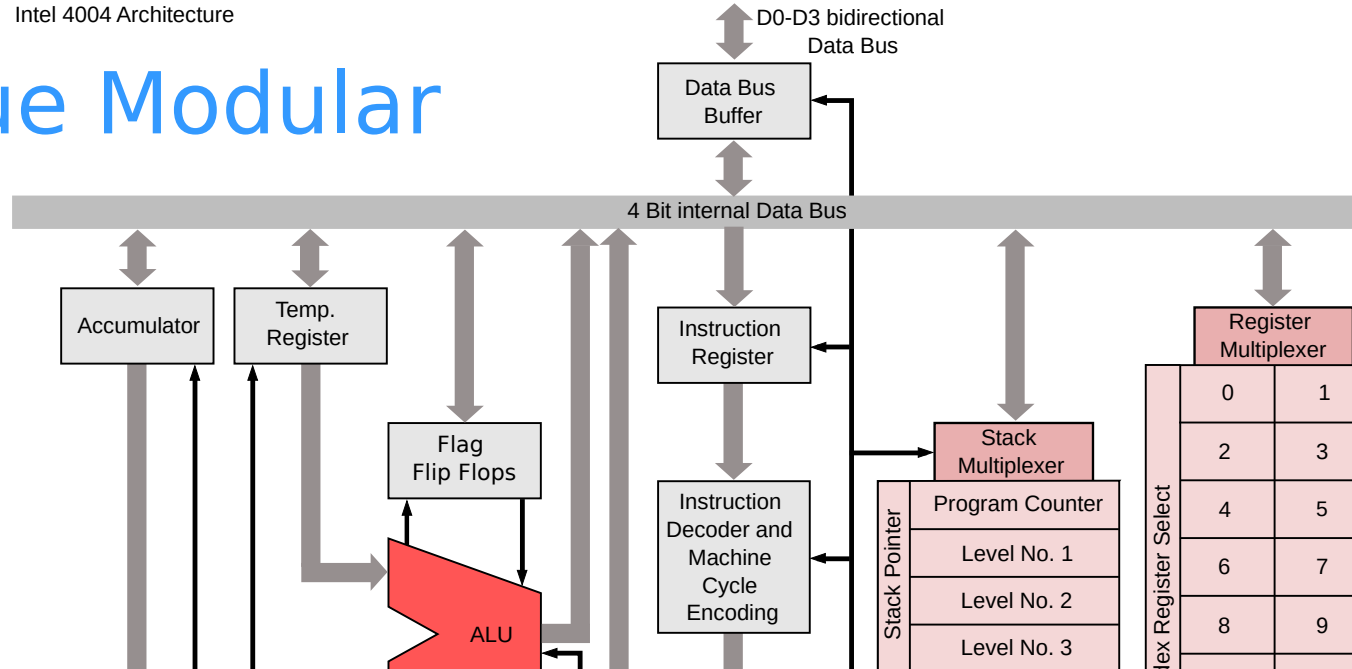
- Un sistema digital secuencial puede ser representado por tablas de estados
- Para sistemas grandes, el método se vuelve poco práctico.

**Intel 4004 1971**  
**1er Microprocesador en un chip**  
\* 2300 transistores  
\* Clock 740kHz  
\* 46 instrucciones  
\* 16 registros de 4 bits



**Solución: Utilizar un enfoque modular.**

# Enfoque Modular

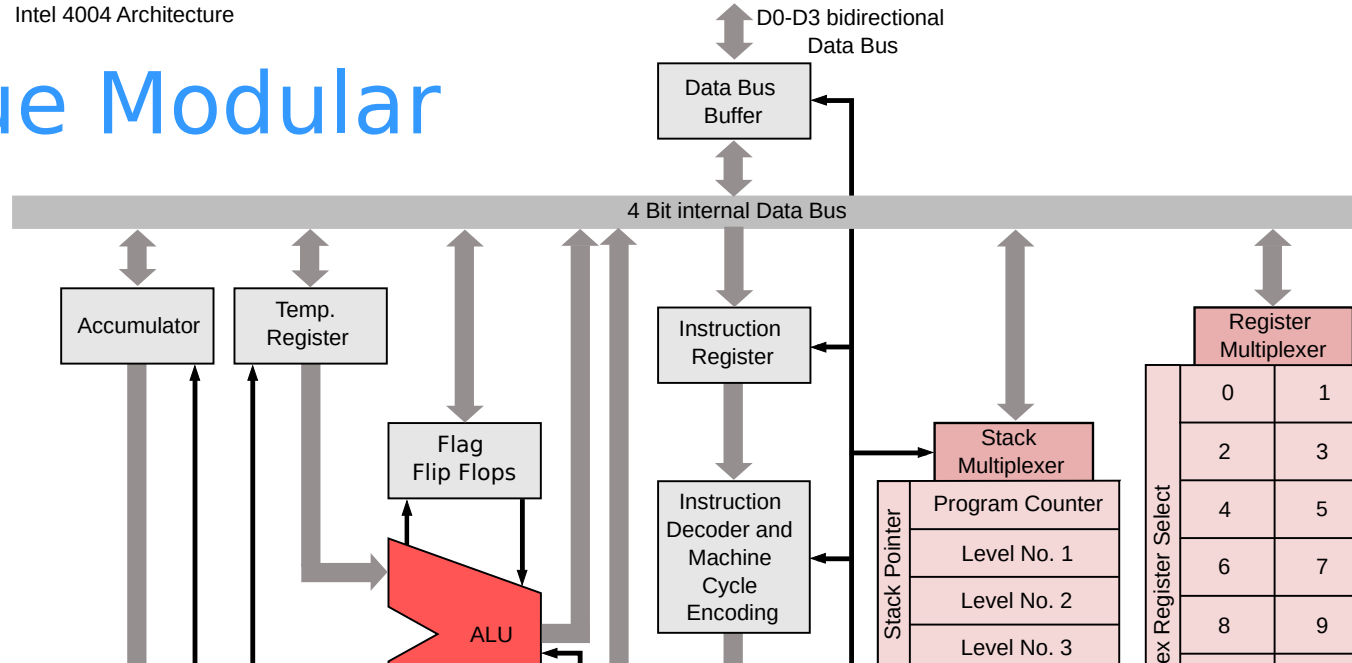


- En este método el sistema es dividido en:
  - Módulos que realizan alguna tarea específica (ALU, Registro de corrimientos, etc).
  - Líneas de comunicación de datos y control entre los módulos.



Todo el conjunto forma un computadora digital

# Enfoque Modular

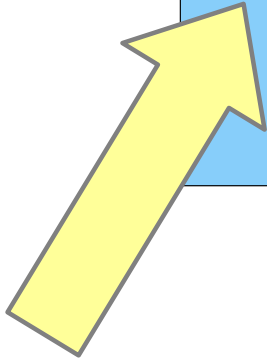


Estos Módulos, son construidos a partir de dispositivos digitales simples

- registros,
- contadores, multiplexores,
- decodificadores,
- elementos aritméticos
- lógica de control.
- Etc.

# Módulos Digitales

- Independientemente de los dispositivos que los forman, podemos distinguir dos elementos que lo definen:
  - Los registros que contienen.
  - Las operaciones que se realizan con el valor digital contenido en ellos. (Corrimiento, conteo, reset, carga de un valor)

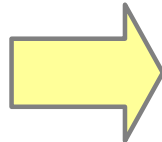


- NOTA: Mas adelante veremos que no necesariamente debe poseer los dos elementos, una ALU por ejemplo solo posee las operaciones entre registros (2do pto)

# Módulos Digitales

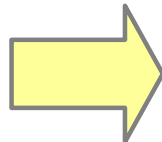
- Entonces, un Módulo Digital o en definitiva un sistema digital lo podemos resumir en un conjunto de

Registros



Componente básico de cualquier sistema digital

Operaciones que se realizan con el contenido de los registros.



Estas operación junto con el flujo de información reciben el nombre de **Transferencia de Registros.**



# Transferencia de Registros

- Las transferencias de registros se definen por medio de los siguientes componentes
  - 1) El conjunto de registros utilizados.
  - 2) Las operaciones que se efectúan con la información almacenada en los registros.
  - 3) El control que supervisa la secuencia de operaciones.



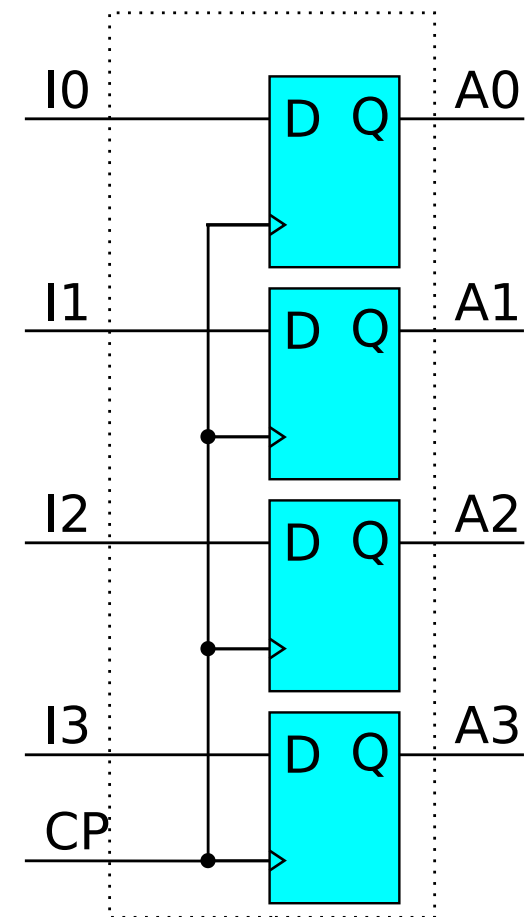
# Registros

Conjunto de multivibradores biestables o flip flop que almacenan información binaria y tienen la posibilidad de realizar una o varias operaciones elementales.

## Ejemplos

- Un contador es un registro que realiza la operación de incremento.
- Un shift register, es un registro que permite la operación de desplazamiento.
- Los flip-flops y compuertas asociadas a cualquier circuito secuencial son registros.

Registro en su forma más sencilla

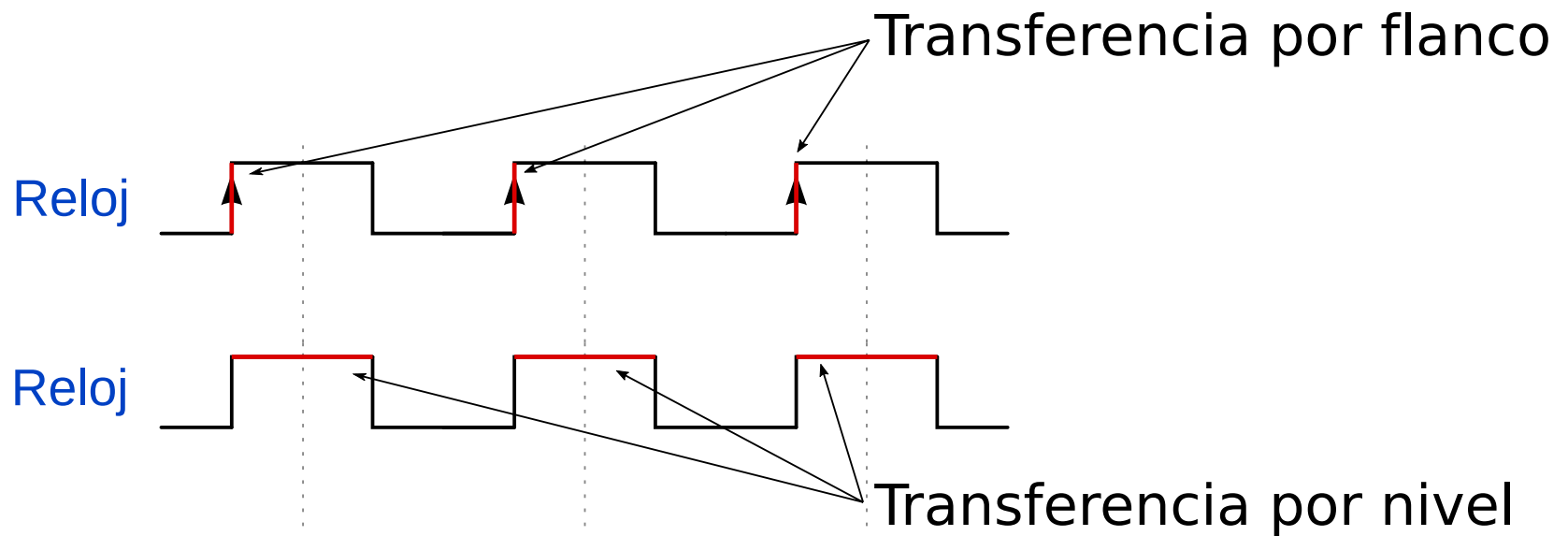


# Registros - Reloj

Todo sistema digital secuencial, posee un reloj que marca el ritmo en el sistema

Los Flip-Flop puede grabar la entrada por nivel o por flanco.

Los Flip-Flop por flanco son los denominados registros y son los usados en los sistemas digitales gobernados por un reloj, relegando el uso de los Flip-Flop por nivel a casos particulares.





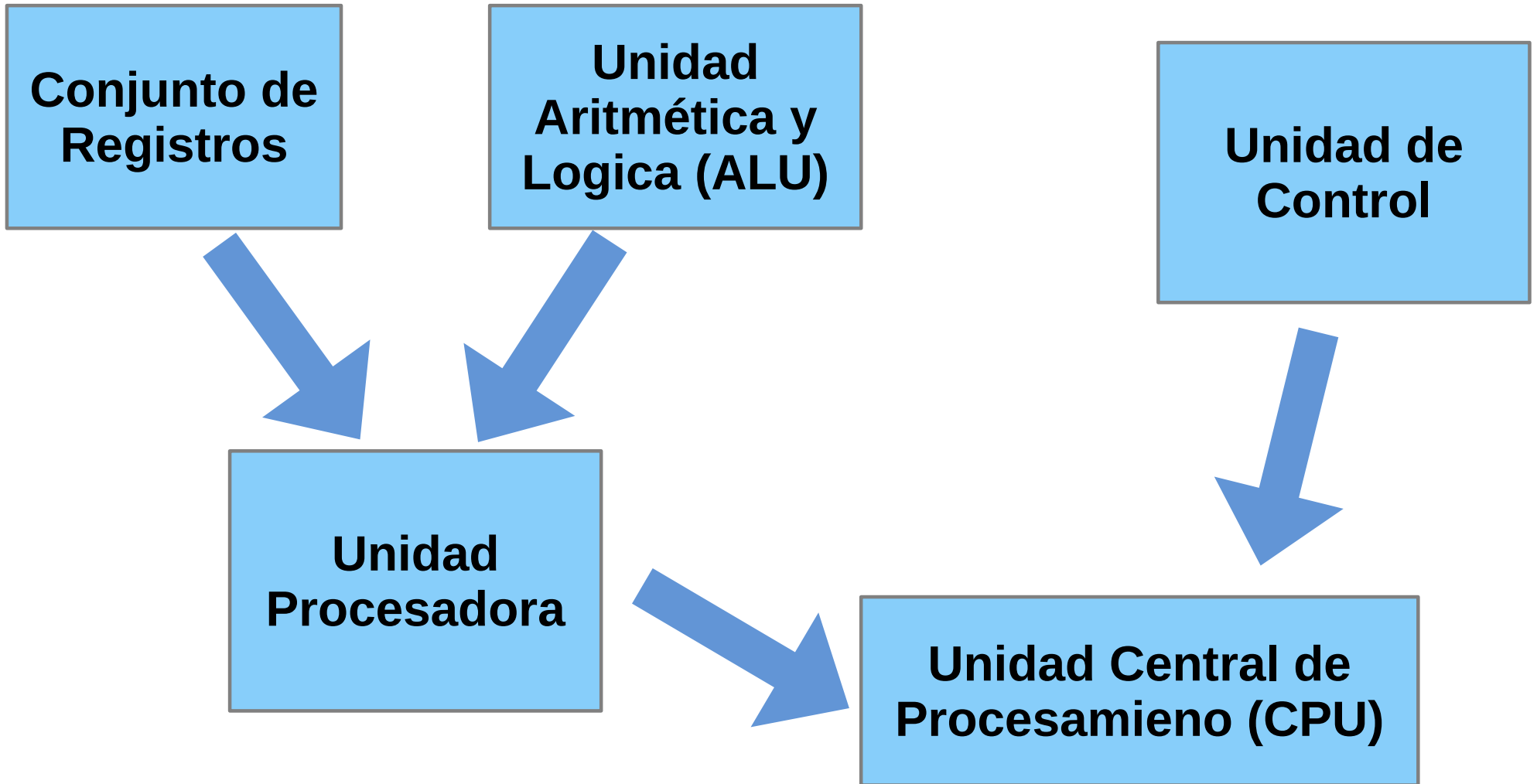
# Microoperaciones

- Son las operaciones que se realizan con la información almacenada en los registros.
- Esta operación se realiza en un ciclo de reloj.
- En general estas operaciones modifican el valor de un registro o transfiere el mismo a otro registro.

En un sistema digital, en lugar de tener registros que realizan microoperaciones individuales, se emplean un conjunto de registros de almacenamiento junto a una unidad operacional común denominada Unidad Aritmética y Lógica (ALU).

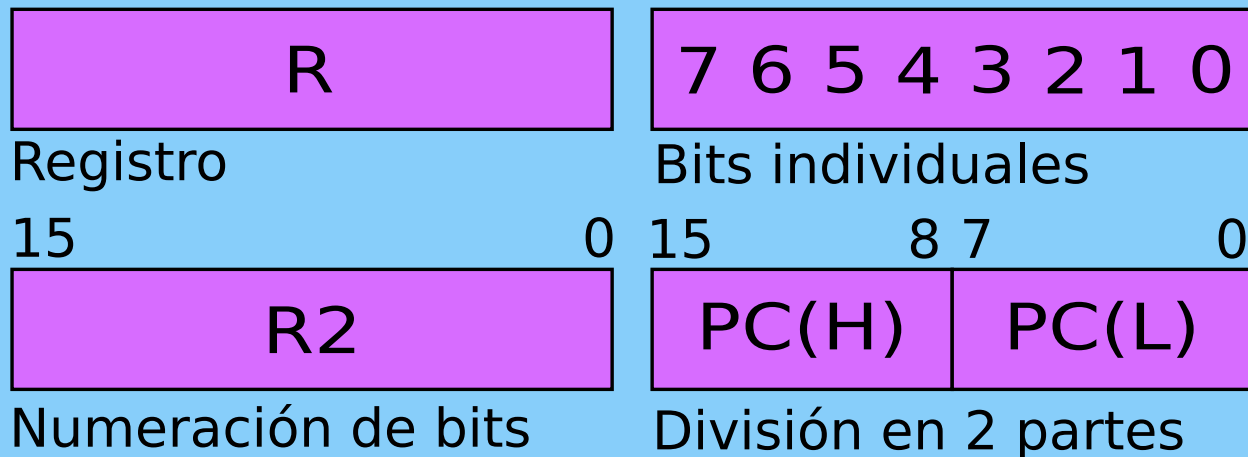
En general los desplazamiento se encuentran en una unidad separada, que a veces se considera parte de la ALU.

# Finalmente la CPU



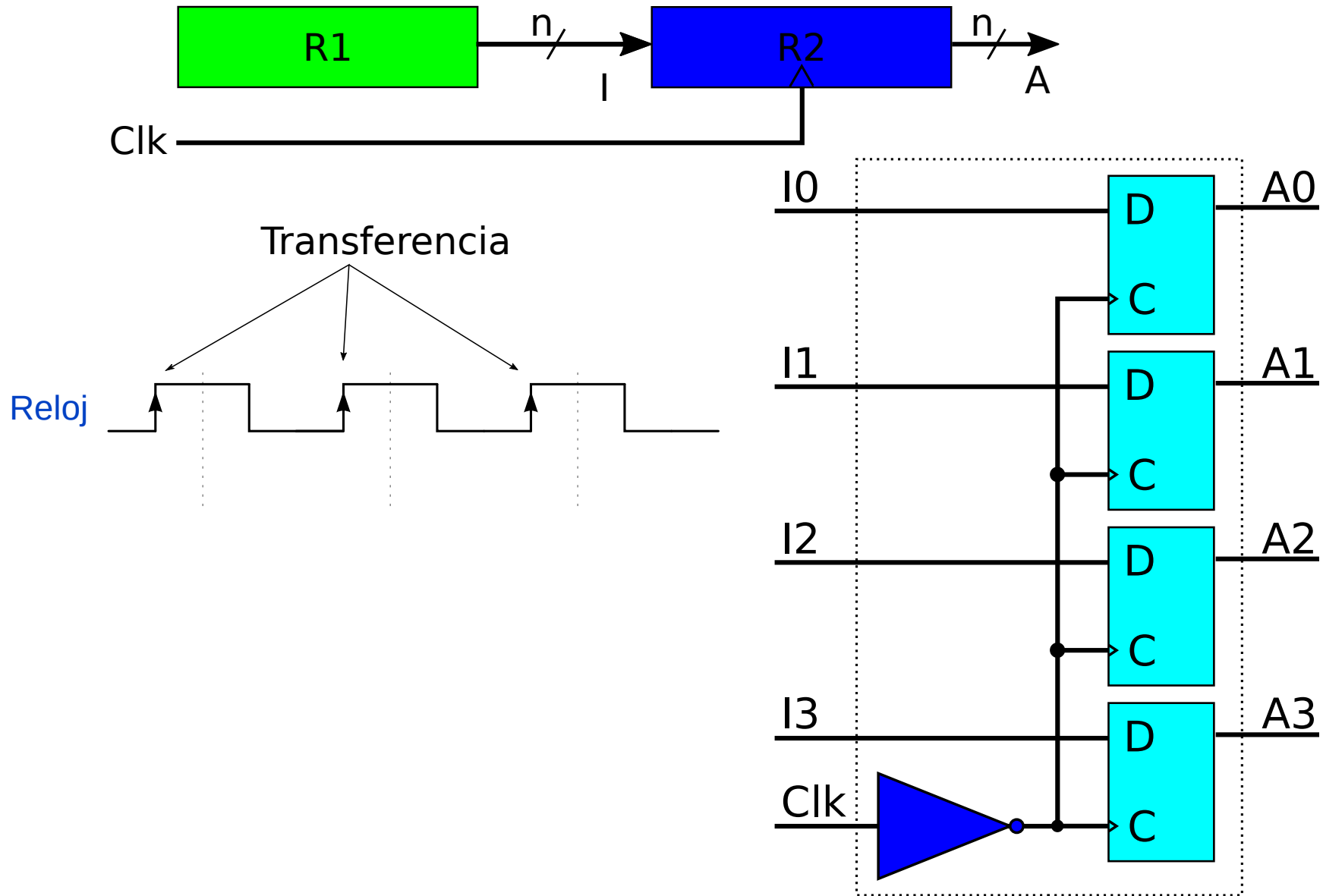
# Transferencia de Registros

- Los registros se designan con letras mayúsculas que denota la función del registro.
  - Registro de dirección de memoria AR
  - Contador de programa PC o IP
  - Registro genérico R1,R2



La transferencia de una registro a otro se designa en forma simbólica como:  $R2 \leftarrow R1$

# Transferencia de Registros



# Transferencia de Registros

- Si la actualización es requerida en una condición determinada, se utiliza una instrucción condicionada

$$\textit{Si}(T_1=1)\textit{ entonces}(R2 \leftarrow R1)$$

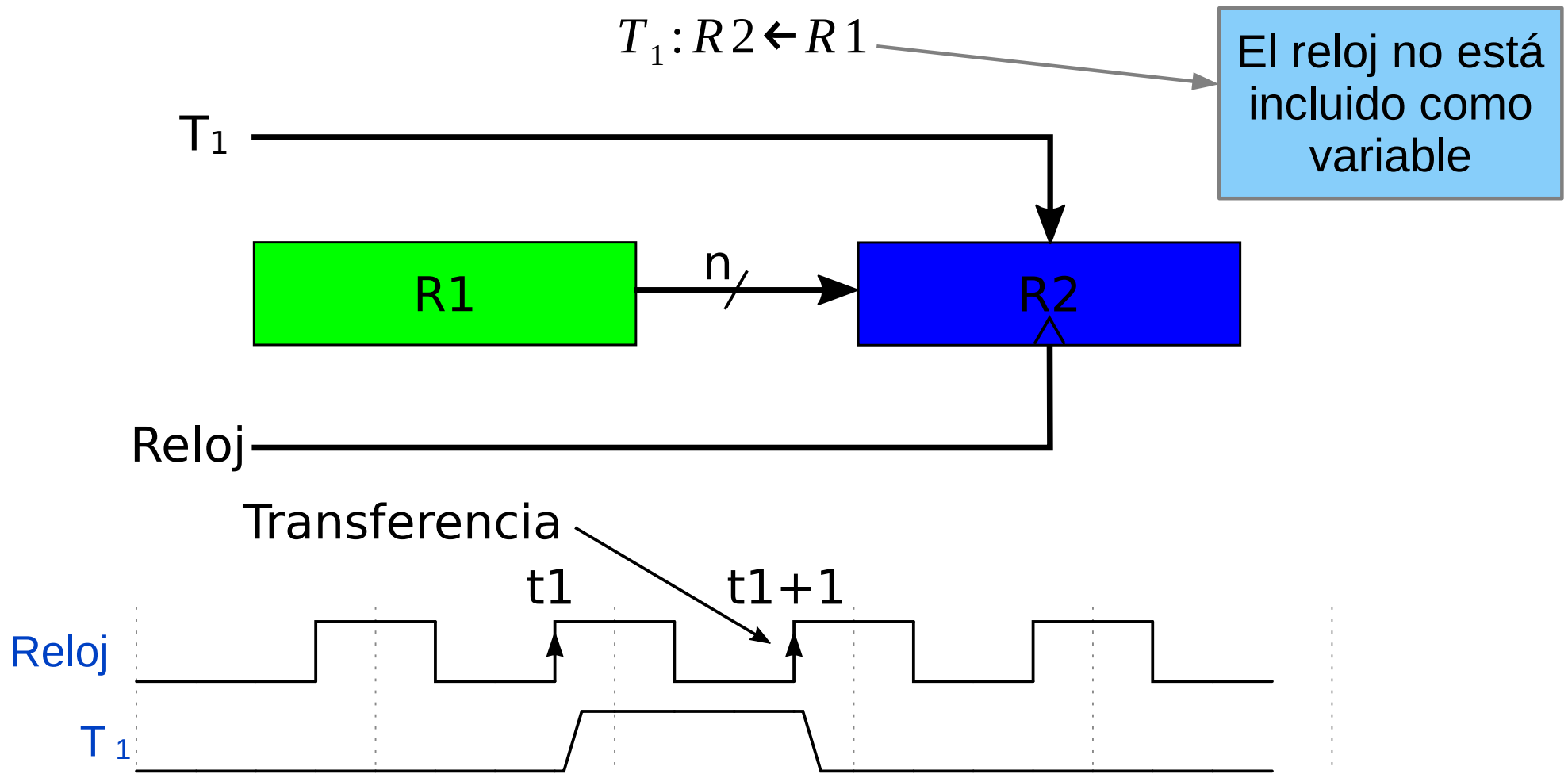
- $T_1$  es una señal de control generada en la sección de control

$$T_1 : R2 \leftarrow R1$$

- La actualización de R2 está condicionada a que  $T_1 = 1$

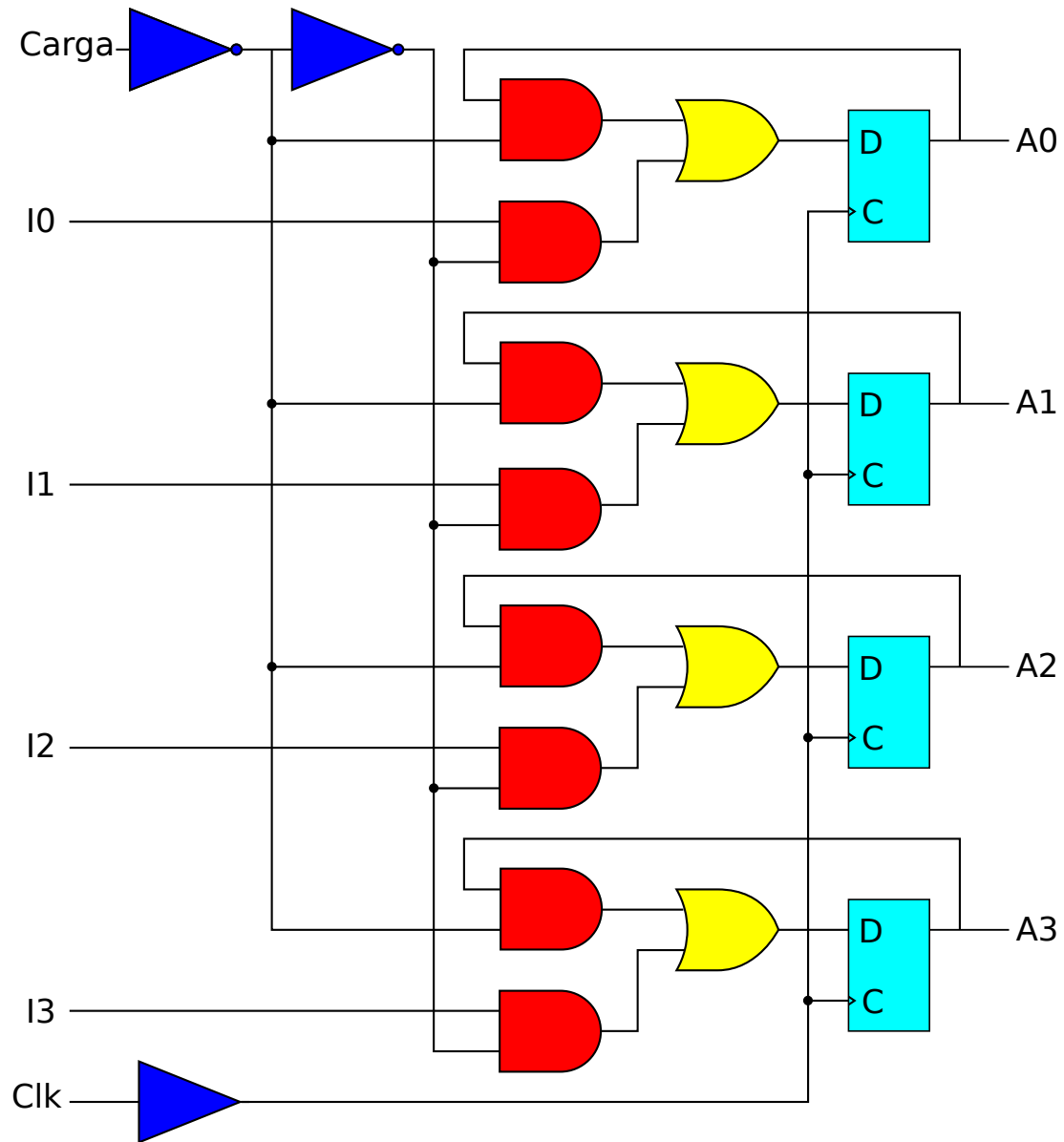
# Transferencia de Registros

Representación en Hardware de la transferencia de un registro condicionado por  $T_1$





# Transferencia de Registros



# Transferencia de Registros

## Símbolos Básicos

Símbolo	Descripción	Ejemplos
Letras y numerales	Denota un registro	$AR, R2, PC$
Paréntesis ( )	Denota una parte de un registro	$R2(0-7), R2(L)$
Flecha ←	Denota transferencia de información	$R2 \leftarrow R1$
Coma ,	Separa dos microoperaciones	$R2 \leftarrow R1, R1 \leftarrow R2$
Corchetes [ ]	Especifica Dirección de la memoria	$DR \leftarrow M[AR]$

# Transferencia de Registros

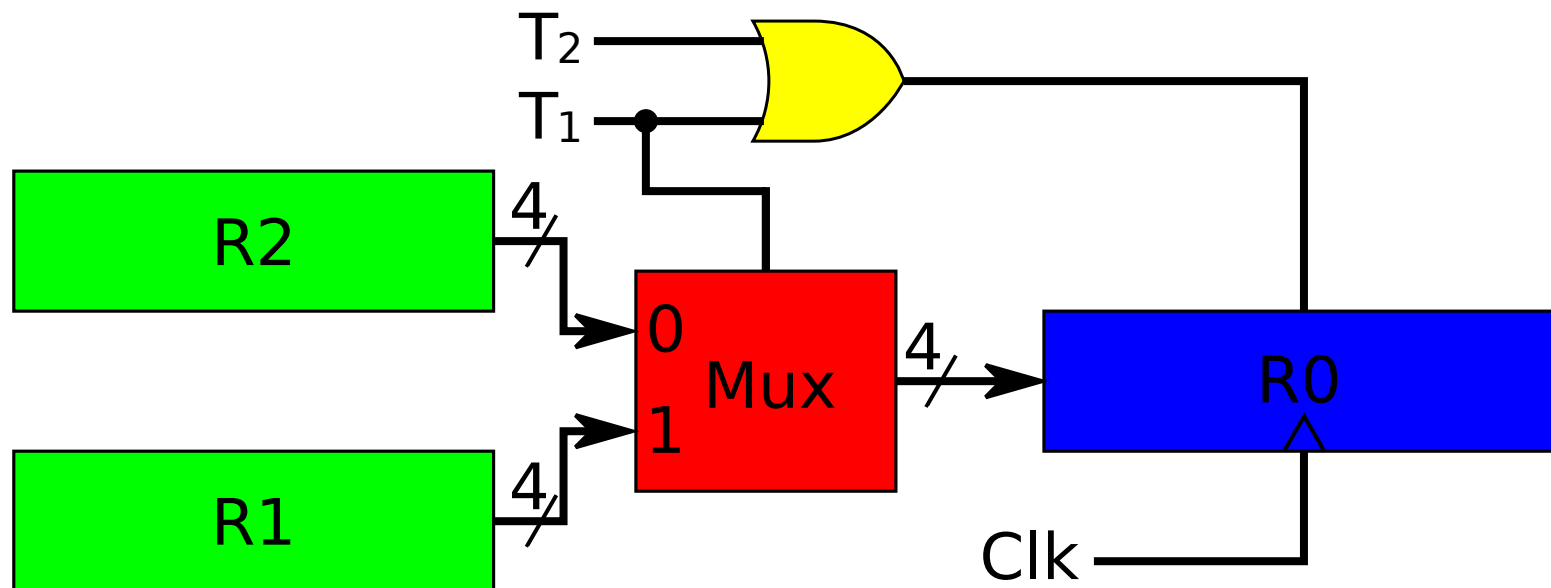
- Multiple transferencias simultáneas

$$T_3: R2 \leftarrow R1, R1 \leftarrow R2$$

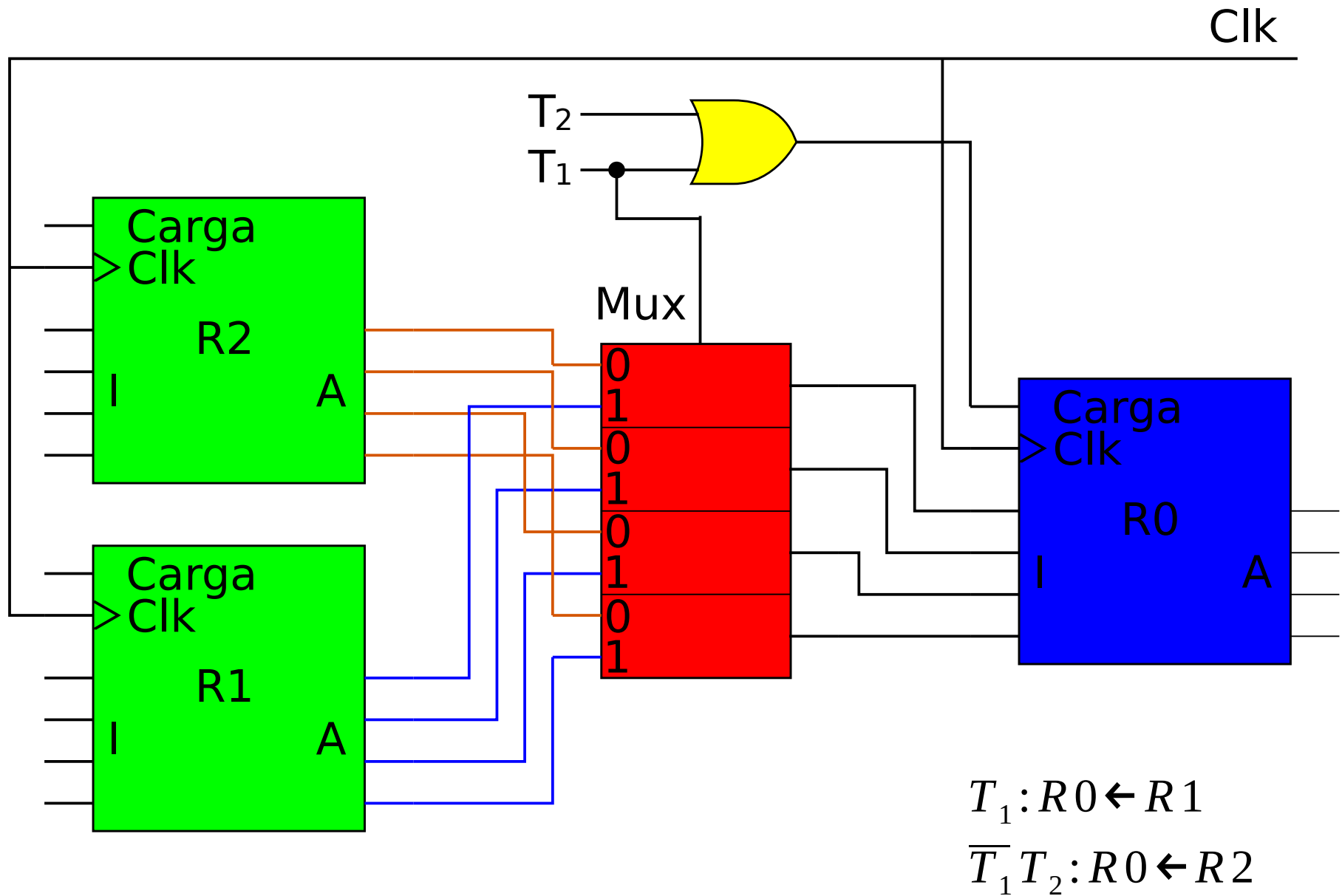
- Fuentes distintas

$$T_1: R0 \leftarrow R1$$

$$\overline{T_1} T_2: R0 \leftarrow R2$$



# Transferencia de Registros





# Microoperaciones

- Operación elemental, realizada con los datos almacenados en los registros, se dividen en:

No cambia el contenido de la información

- Transferencias de registros

El contenido de la información es cambiado.

- Operaciones Aritméticas
- Operaciones Lógicas
- Operaciones de Corrimiento

# Microoperaciones Aritméticas

- Operaciones básicas  
suma , resta, incremento y disminución

- ~~• El corrimiento aritmético~~

Es una operación aritmética pero se incluye en las microoperaciones de corrimiento.

- ~~• Multiplicación y División~~

No se incluyen como operaciones básicas.  
Solo pueden considerarse microoperación si se realiza en un ciclo de reloj y no con un conjunto de microoperaciones (microcódigo)

# Microoperaciones Aritméticas

Microoperación de suma

$$R0 \leftarrow R1 + R2$$

Microoperación de resta

$$R0 \leftarrow R1 - R2$$

y utilizando el complemento a 2 con la suma

$$R0 \leftarrow R1 + \overline{R2} + 1$$

Microoperación de incremento y disminución

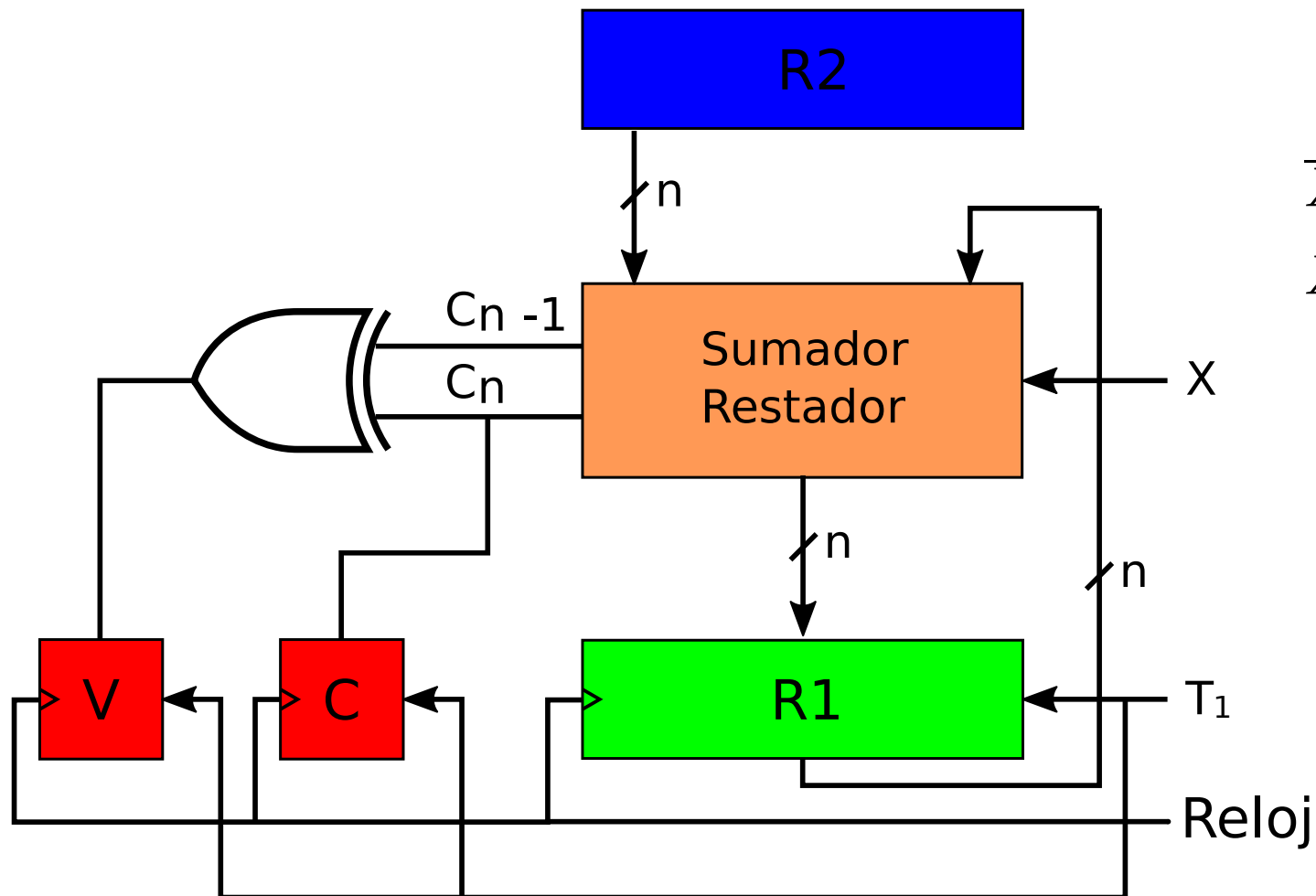
$$R0 \leftarrow R1 + 1 \quad R0 \leftarrow R1 - 1$$

# Microoperaciones Aritméticas

Símbolo	Descripción
$R0 \leftarrow R1 + R2$	Contenido de R1 más R2 transferido a R0
$R0 \leftarrow R1 - R2$	Contenido de R1 menos R2 transferido a R0
$R0 \leftarrow \overline{R2}$	Complemento del contenido de R2 (Complemento a 1)
$R0 \leftarrow \overline{R2} + 1$	Complemento a 2 del contenido de R2
$R0 \leftarrow R1 + \overline{R2} + 1$	R1 más el complemento a 2 de R2 (sustracción)
$R0 \leftarrow R1 + 1$	Incremento del contenido de R1
$R0 \leftarrow R1 - 1$	Decremento del contenido de R1



# Microoperaciones Aritméticas



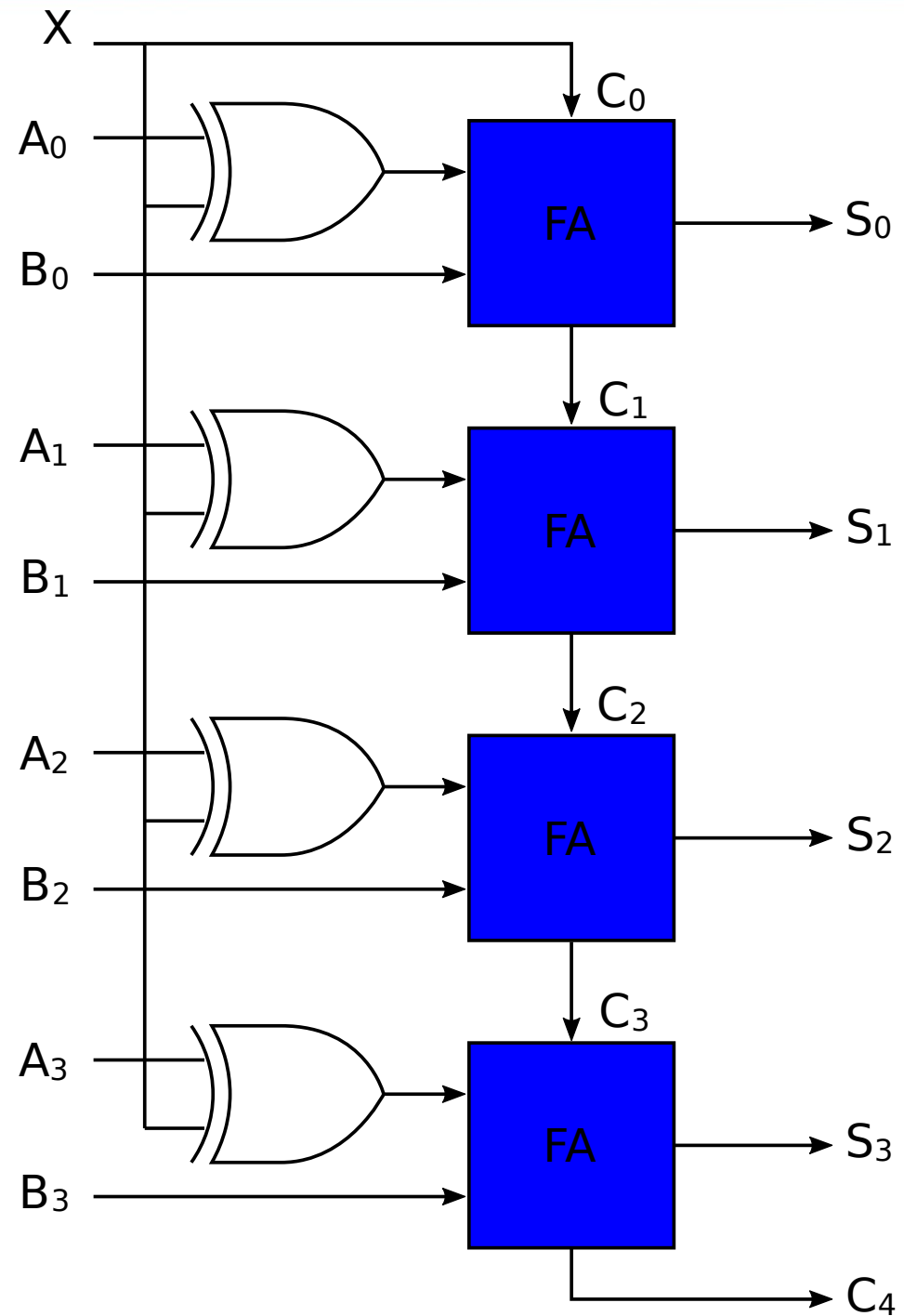
$$\overline{X}T_1: R1 \leftarrow R1 + R2$$
$$XT_1: R1 \leftarrow R1 + \overline{R2} + 1$$

# Microoperaciones Aritméticas

Módulo Sumador Restador

$$\bar{X}T_1: R1 \leftarrow R1 + R2$$

$$XT_1: R1 \leftarrow R1 + \bar{R2} + 1$$

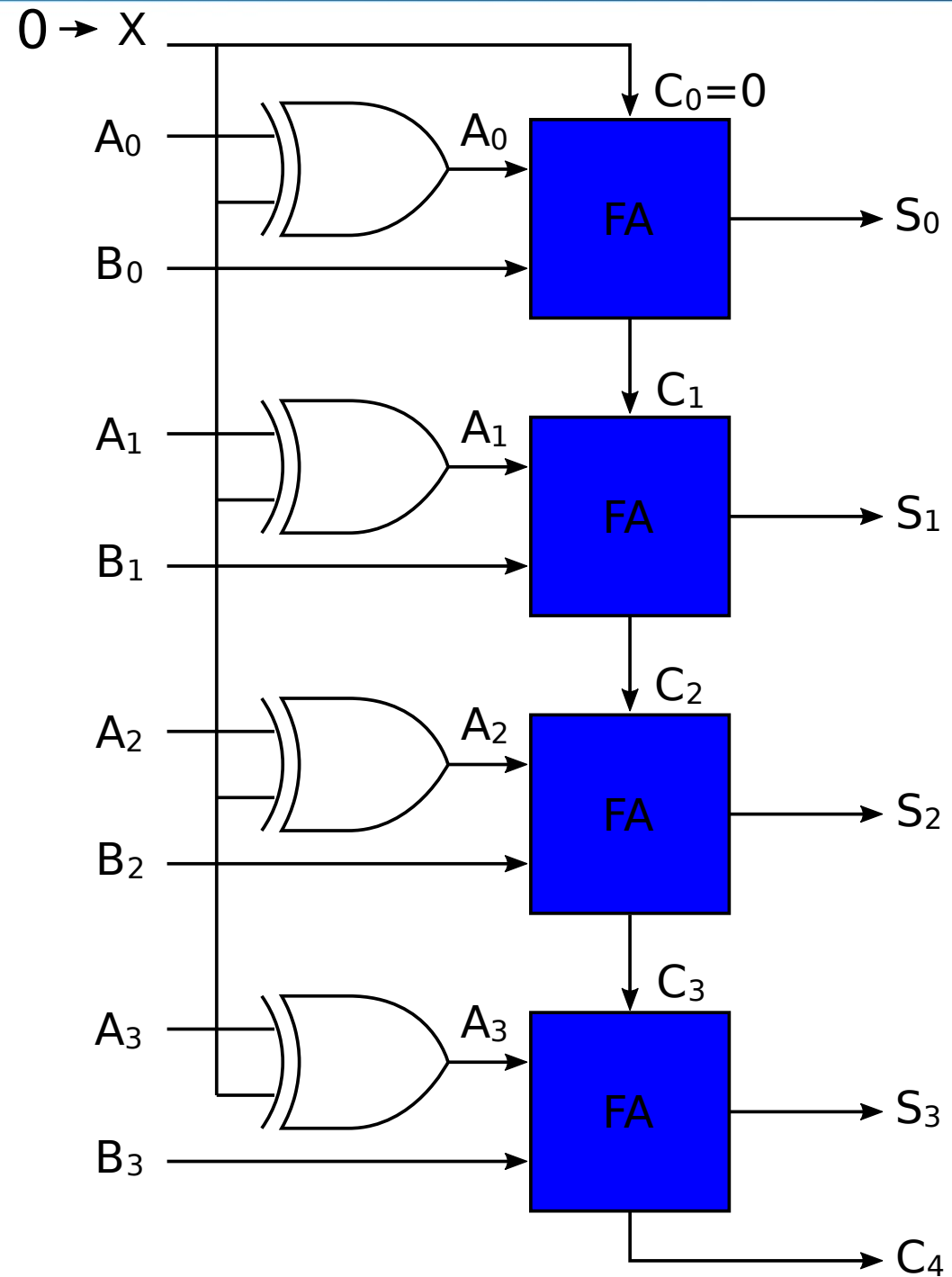


# Microoperaciones Aritméticas

Módulo Sumador Restador

$$\bar{X}T_1: R1 \leftarrow R1 + R2$$

$$XT_1: R1 \leftarrow R1 + \bar{R2} + 1$$

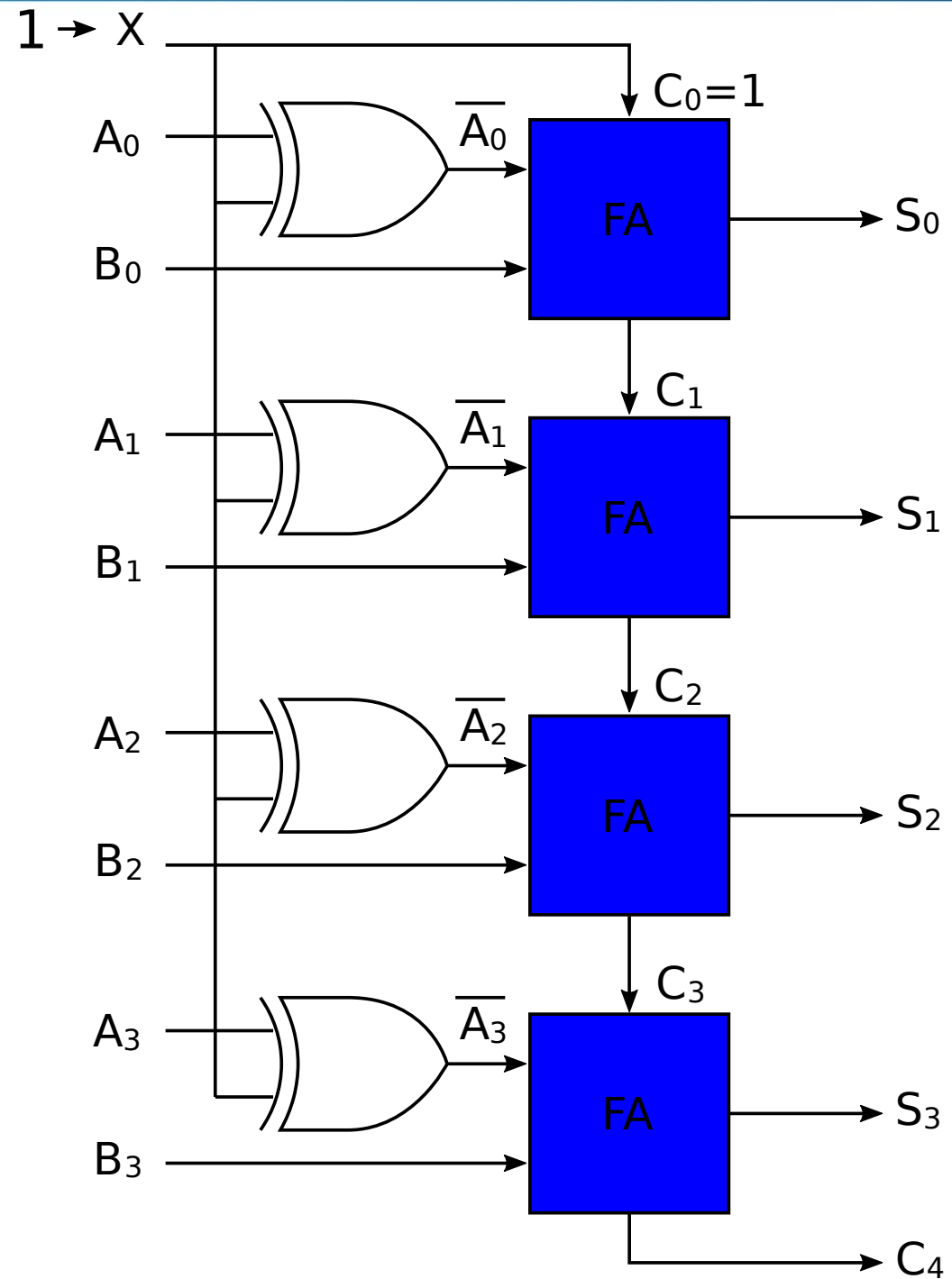


# Microoperaciones Aritméticas

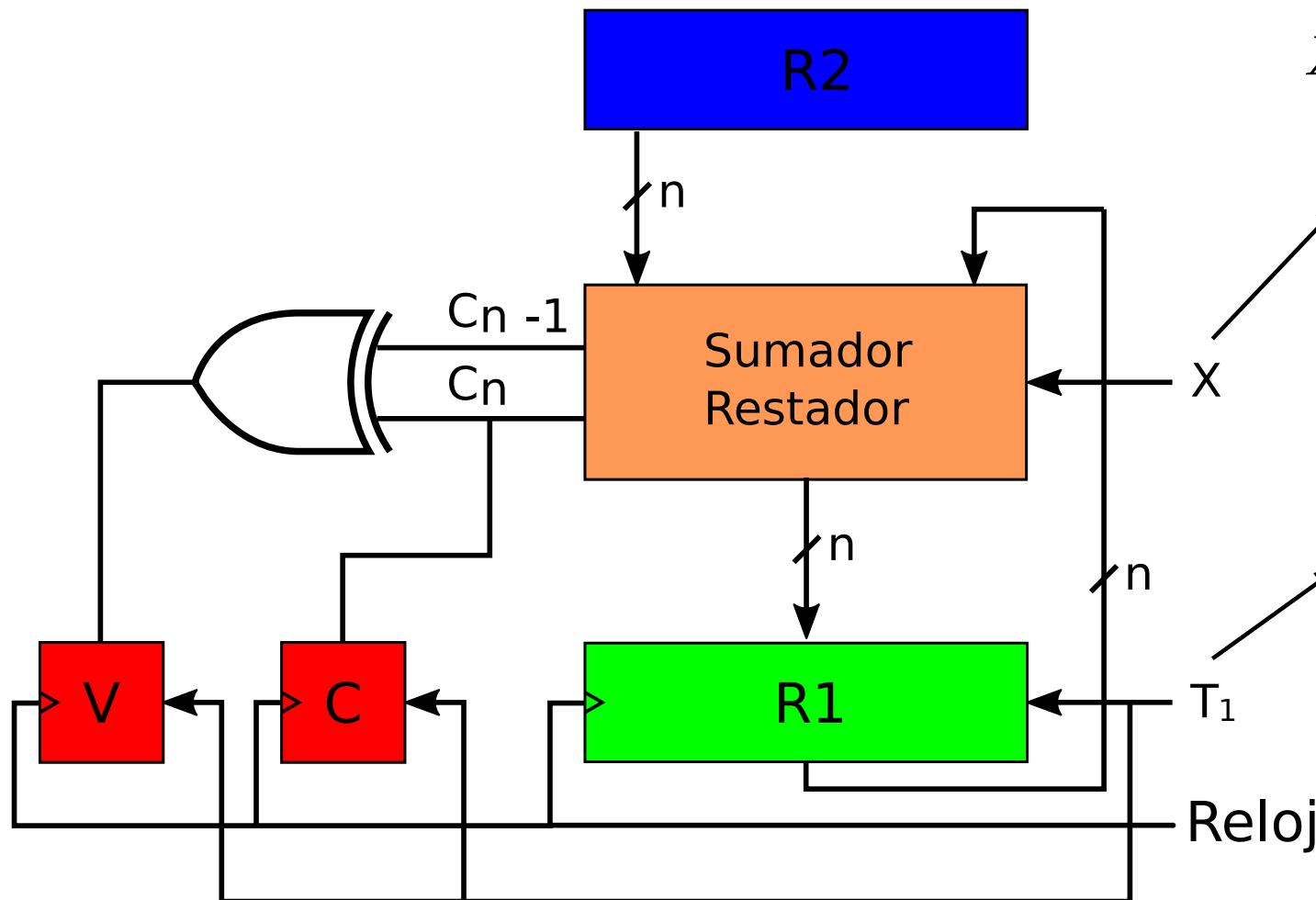
Módulo Sumador Restador

$$\bar{X}T_1: R1 \leftarrow R1 + R2$$

$$XT_1: R1 \leftarrow R1 + \bar{R2} + 1$$



# Microoperaciones Aritméticas



$$\bar{X}T_1: R1 \leftarrow R1 + R2$$

$$XT_1: R1 \leftarrow R1 + \bar{R2} + 1$$

Si  $X=0$  suma y  
si  $X=1$  resta

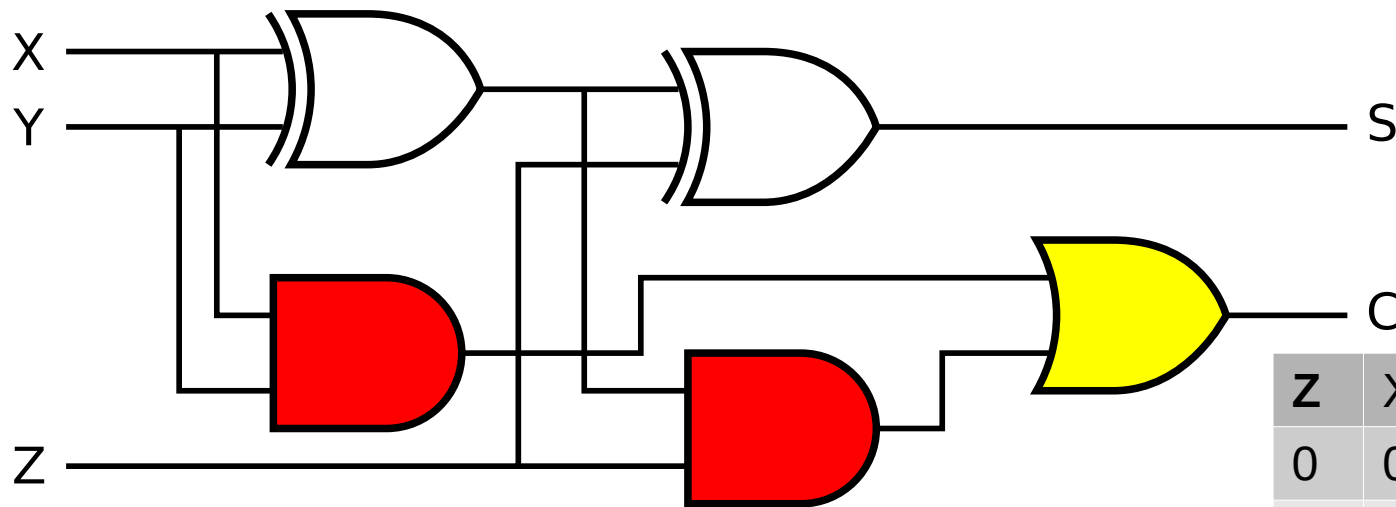
Carga en R1 si  
 $XT_1=1$   $\bar{X}T_1=1$   
entonces

$$\bar{X}T_1 + XT_1 = 1$$

$$(\bar{X} + X)T_1 = T_1 = 1$$

# Microoperaciones Aritméticas

## Módulo Sumador Completo



Z	X	Y	S	C
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

$$S = X \oplus Y \oplus Z$$

$$C = XY + Z(X \oplus Y)$$

# Microoperaciones Aritméticas- Desborde

Distintos, hay desborde

Acarreo 0 1  
+70 0 1000110  
+80 0 1010000  
~~+150 1 0010110~~

Acarreo 1 0  
-70 1 0111010  
-80 1 0110000  
~~-150 0 1101010~~

Acarreo en nro sin signo

Acarreo 0 0  
+70 0 1000110  
+20 0 0010100  
~~+90 0 1011010~~

Acarreo 1 0  
70 0 1000110  
190 1 0111110  
~~260 0 0000100~~

# Microoperaciones Lógicas

Permite manipular bits almacenados en los registros

- Complemento  $R \leftarrow \bar{R}$
- AND lógica  $R0 \leftarrow R1 \wedge R2$
- OR lógica  $R0 \leftarrow R1 \vee R2$
- XOR lógica  $R0 \leftarrow R1 \oplus R2$



# Microoperaciones Lógicas

**AND**

```
10101101 10101011 R1
00000000 11111111 R2
00000000 10101011
R0 ← R1 ∧ R2
```

**OR**

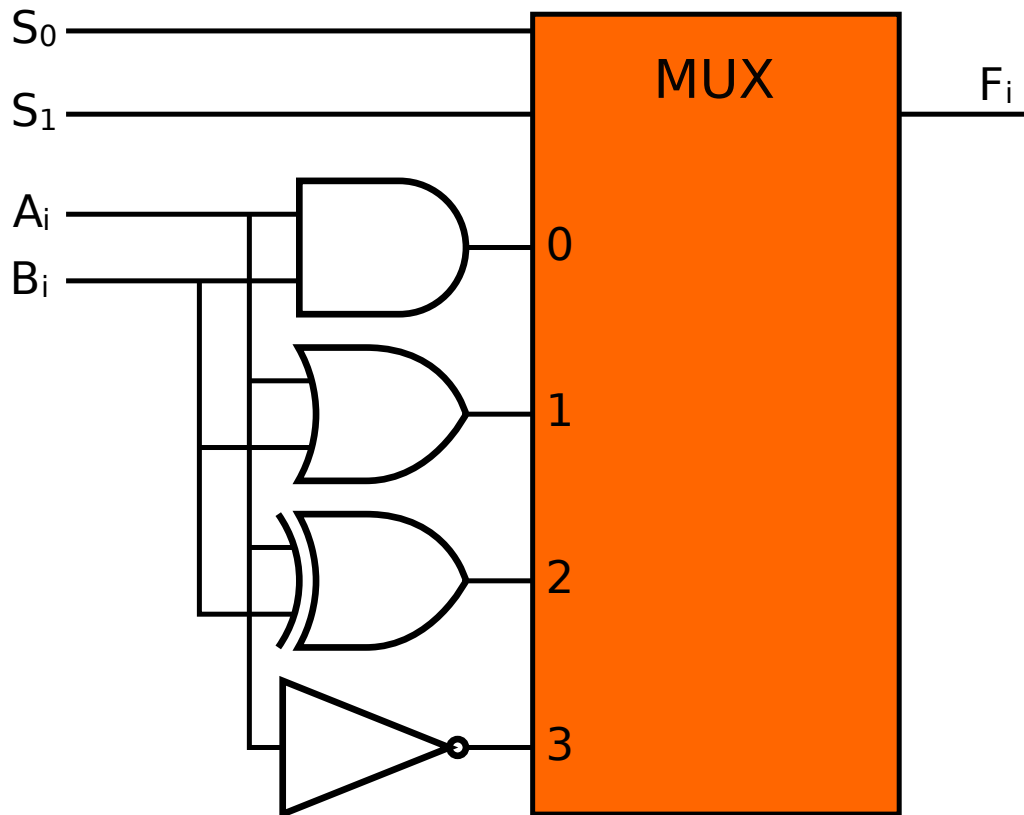
```
10101101 10101011 R1
11111111 00000000 R2
11111111 10101011
R0 ← R1 ∨ R2
```

**XOR**

```
10101101 10101011 R1
11111111 00000000 R2
01010010 10101011
R0 ← R1 ⊕ R2
```

# Microoperaciones Lógicas

Etapa de un circuito lógico



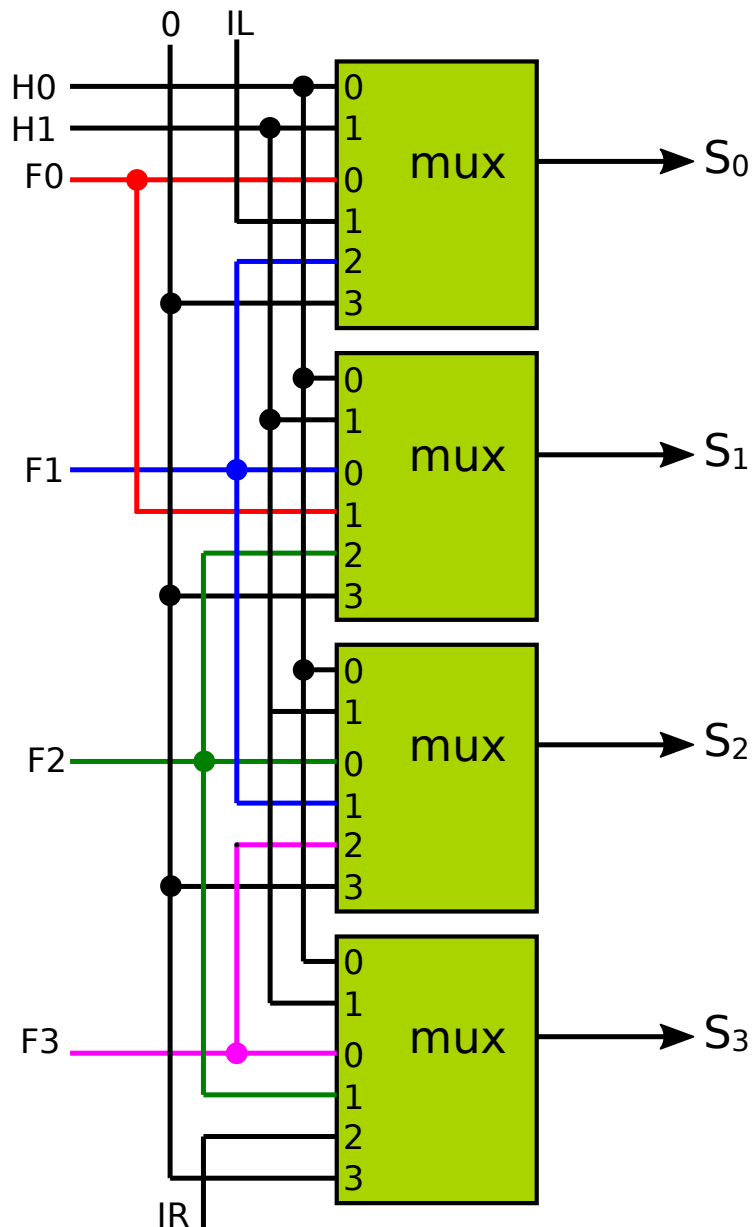
$S_0$	$S_1$	Función
0	0	AND
0	1	OR
1	0	XOR
1	1	Complemento

# Microoperaciones de Corrimiento

Se utilizan para manipular datos en serie y para realizar operaciones aritméticas, lógicas y de control.

- Corrimiento a la izquierda  $R \leftarrow shl R$
- Corrimiento a la derecha  $R \leftarrow shr R$
- Rotación a la izquierda  $R \leftarrow rol R$
- Rotación a la derecha  $R \leftarrow ror R$
- Corrimiento aritmético a la izquierda  $R \leftarrow asl R$
- Corrimiento aritmético a la derecha  $R \leftarrow asr R$

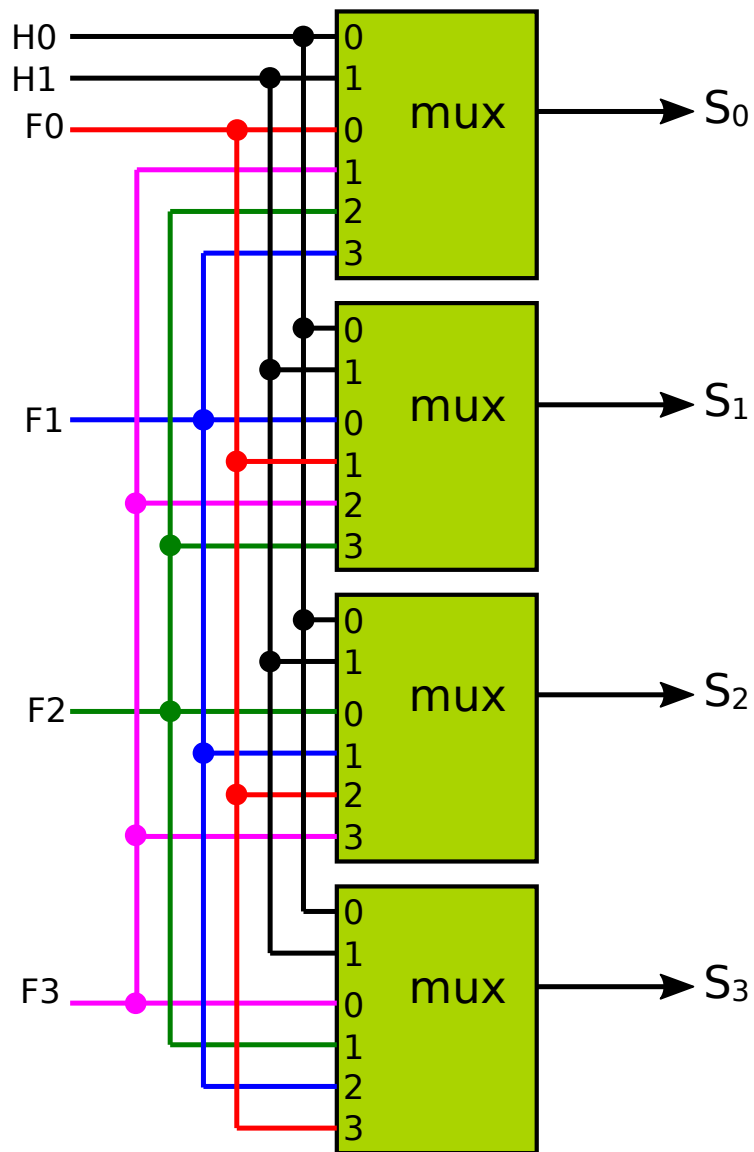
# Microoperaciones de Corrimiento



**Unidad de Corrimiento de 4 bits**

H1	H0	Función
0	0	Transferencia
0	1	Corrimiento a la izquierda
1	0	Corrimiento a la derecha
1	1	Transferencia de 0

# Microoperaciones de Corrimiento



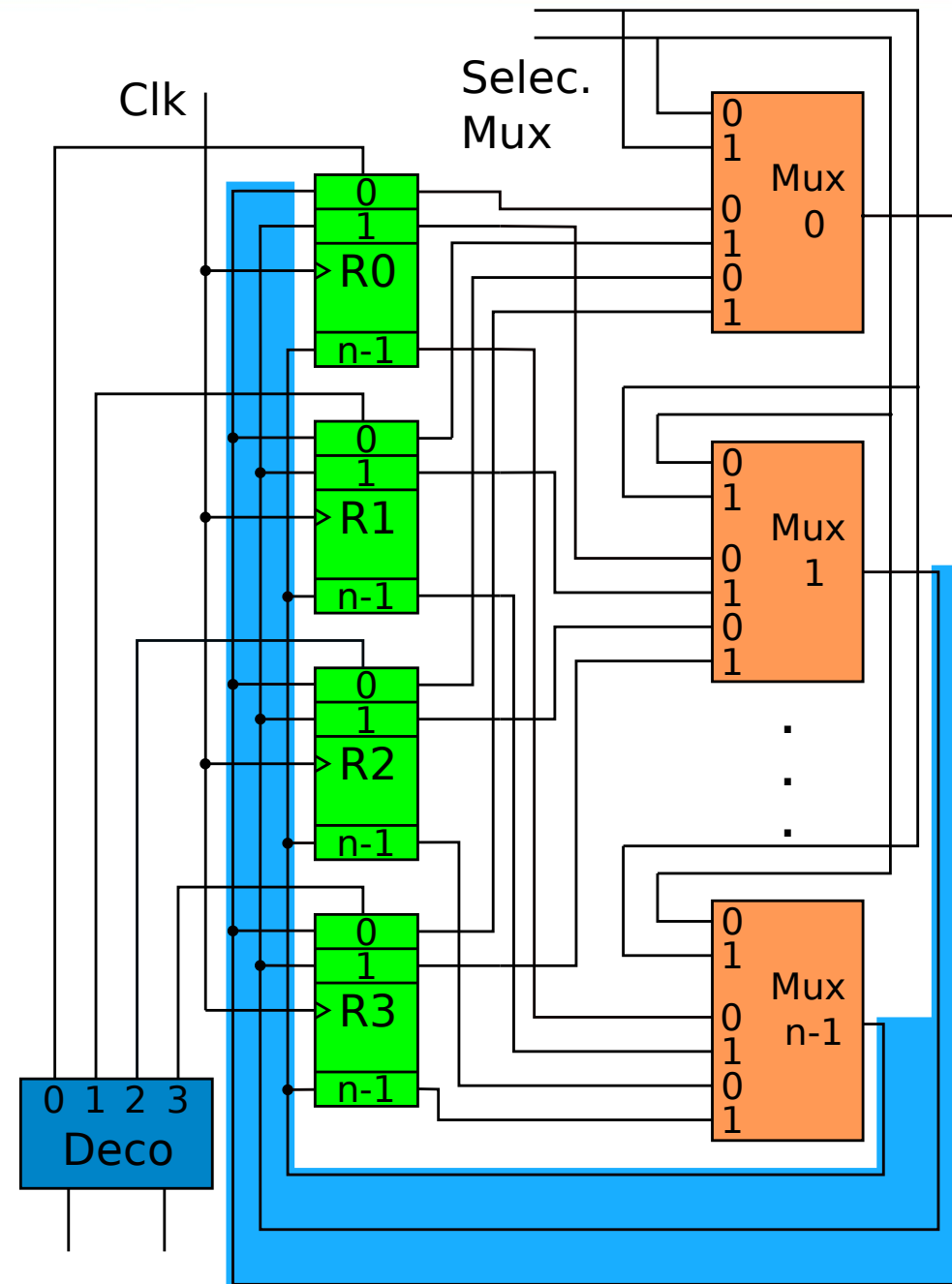
**Unidad de Corrimiento de tambor de 4 bits**

H1	H0	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Función
0	0	F3	F2	F1	F0	Sin corrimiento
0	1	F2	F1	F0	F3	Rotación 1 bits
1	0	F1	F0	F3	F2	Rotación 2 bits
1	1	F0	F3	F2	F1	Rotación 3 bits

# Bus

- Esquema mas eficiente para transferir información
- Se utiliza multiplexores

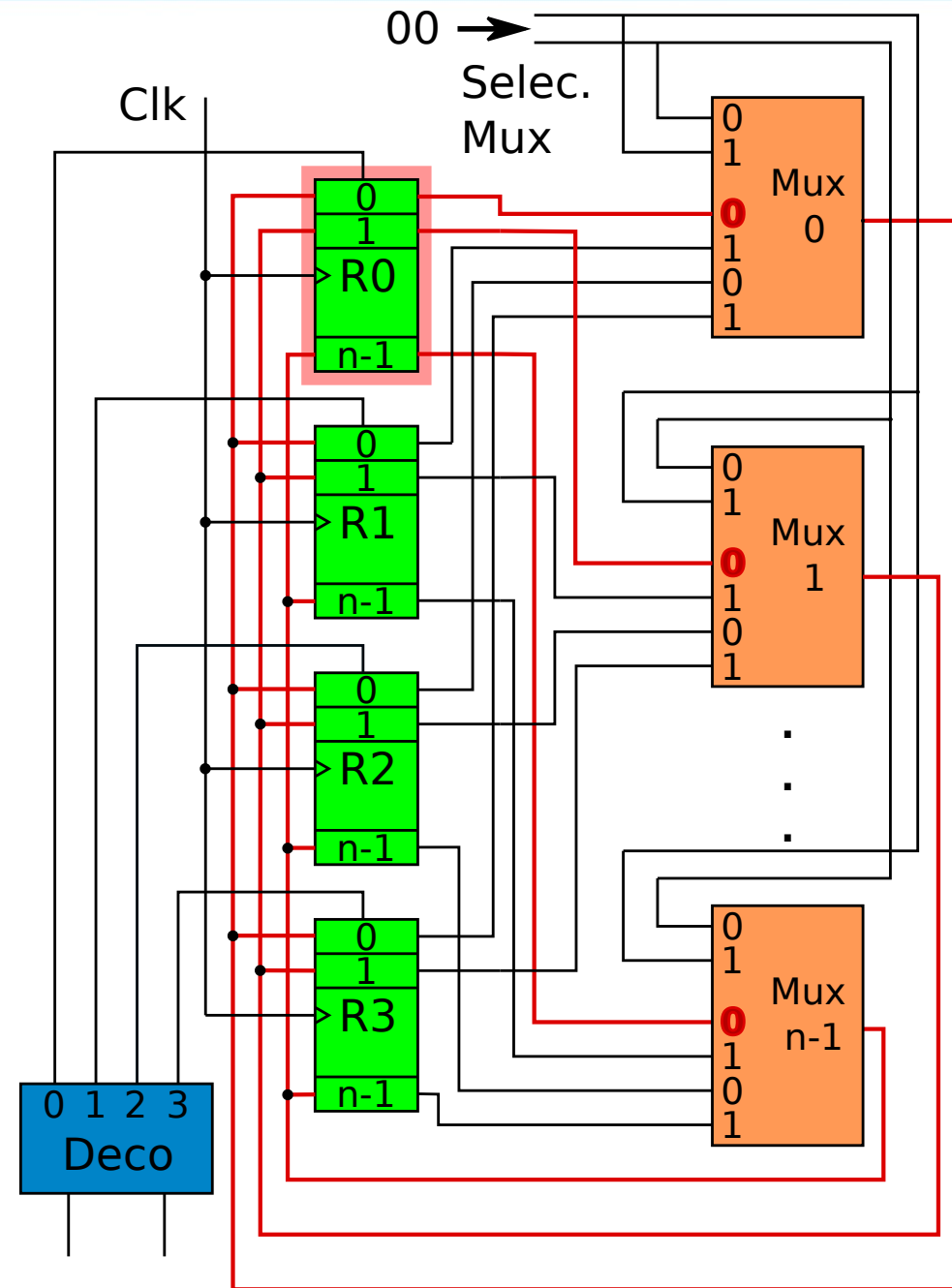
$$R2 \leftarrow R0$$



# Bus

- Esquema mas eficiente para transferir información
- Se utiliza multiplexores

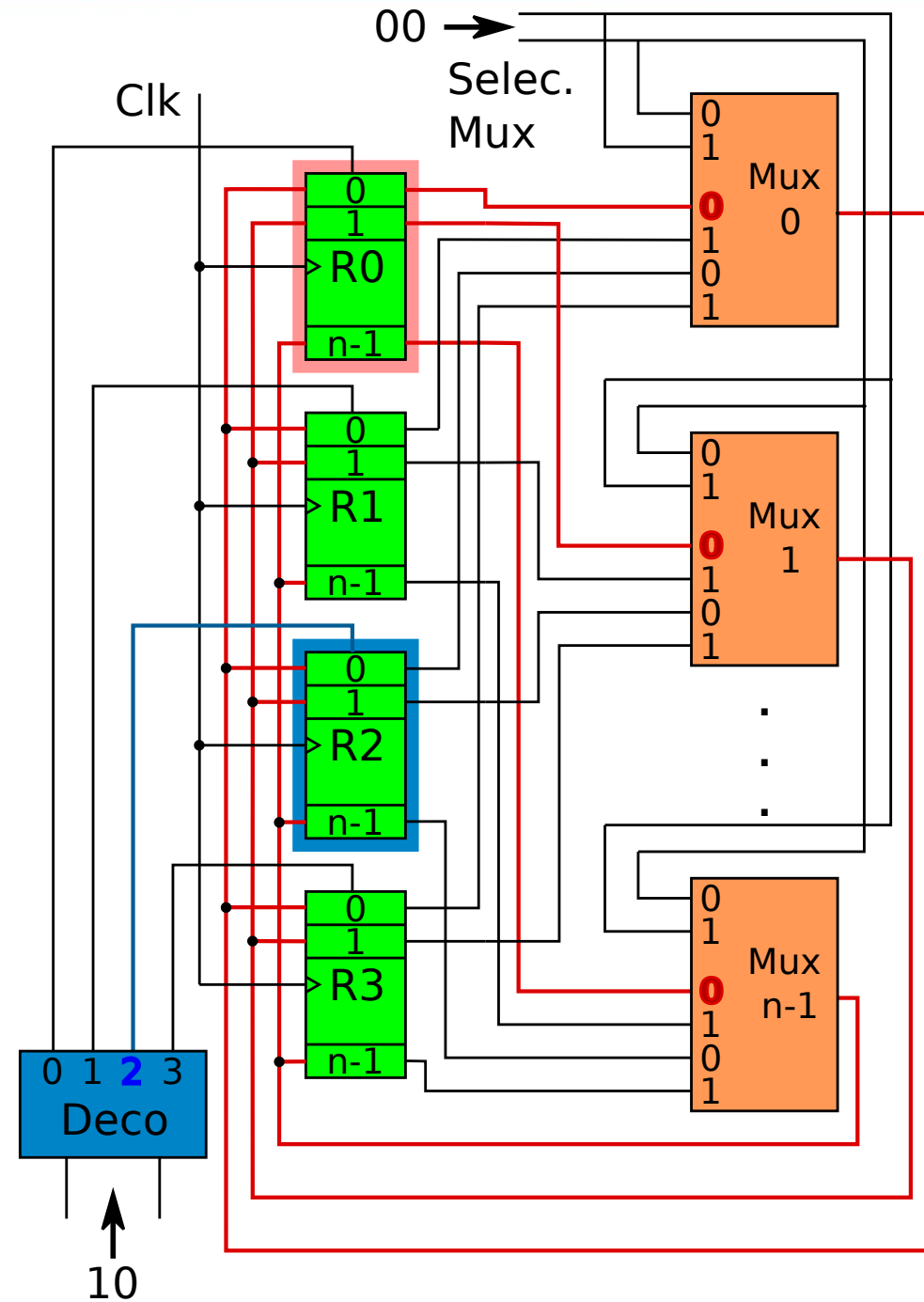
$$R2 \leftarrow R0$$



# Bus

- Esquema mas eficiente para transferir información
- Se utiliza multiplexores

$$R2 \leftarrow R0$$

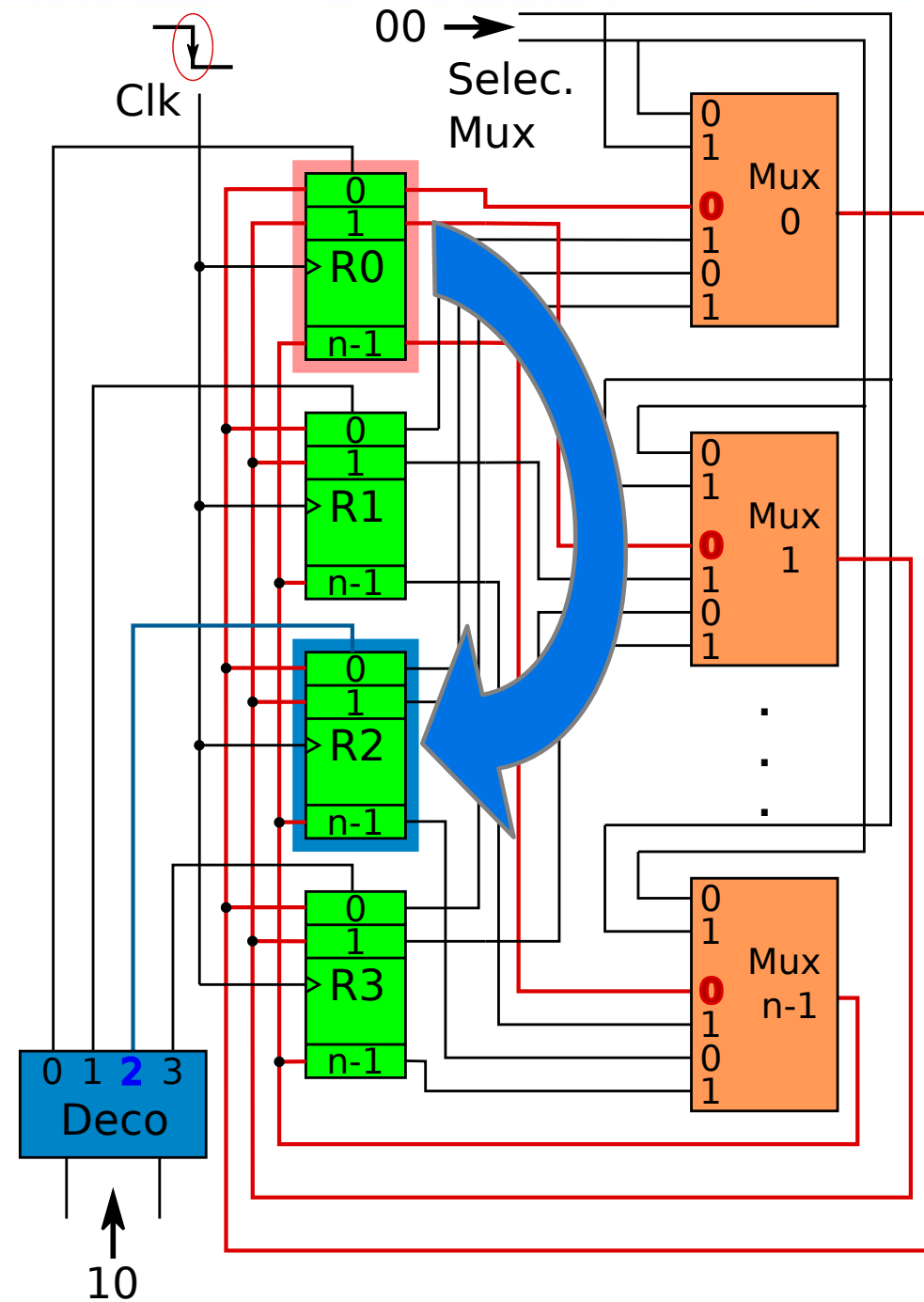




# Bus

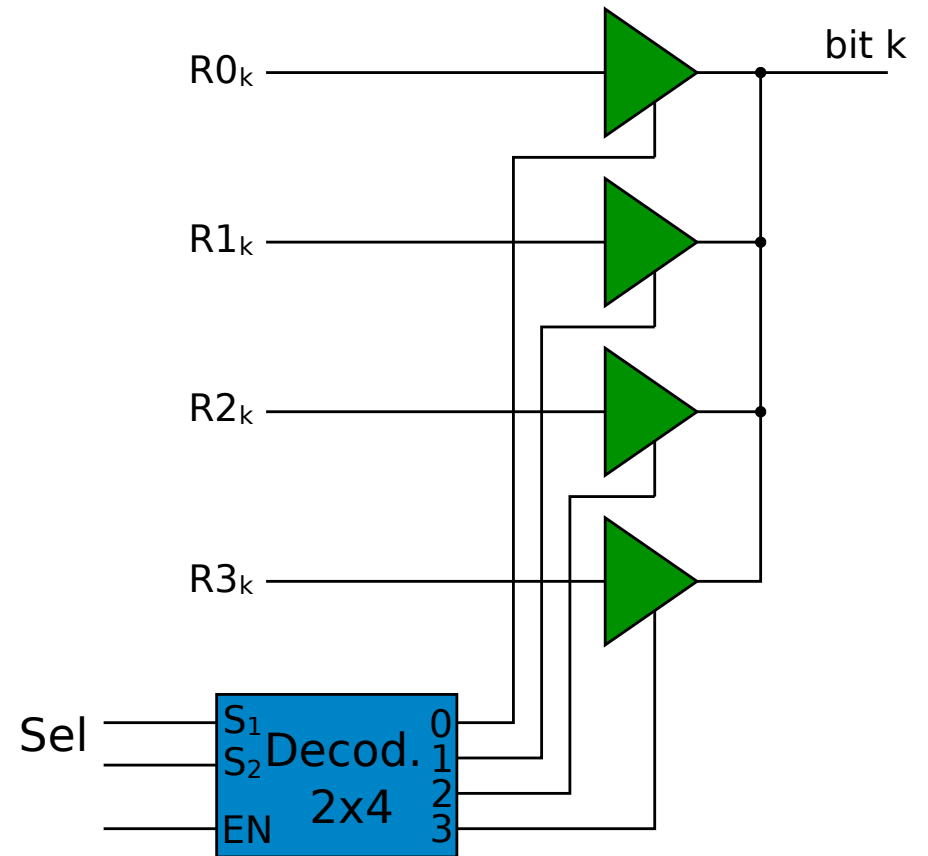
- Esquema mas eficiente para transferir información
- Se utiliza multiplexores

$$R2 \leftarrow R0$$



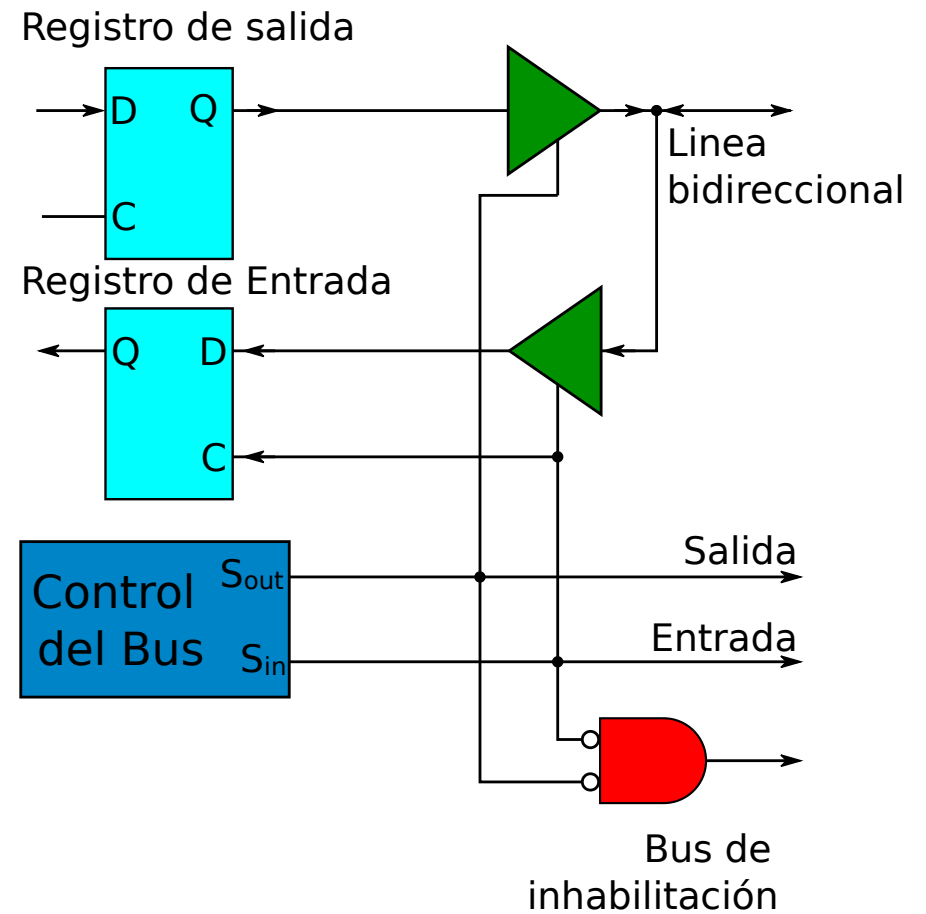
# Bus de tres estados

- Reemplaza los multiplexores por compuertas de tres estados.
- Las salidas no cargan a la red



# Bus bidireccional

- Permite transferir datos binarios en ambas direcciones



# Transferencia de la Memoria

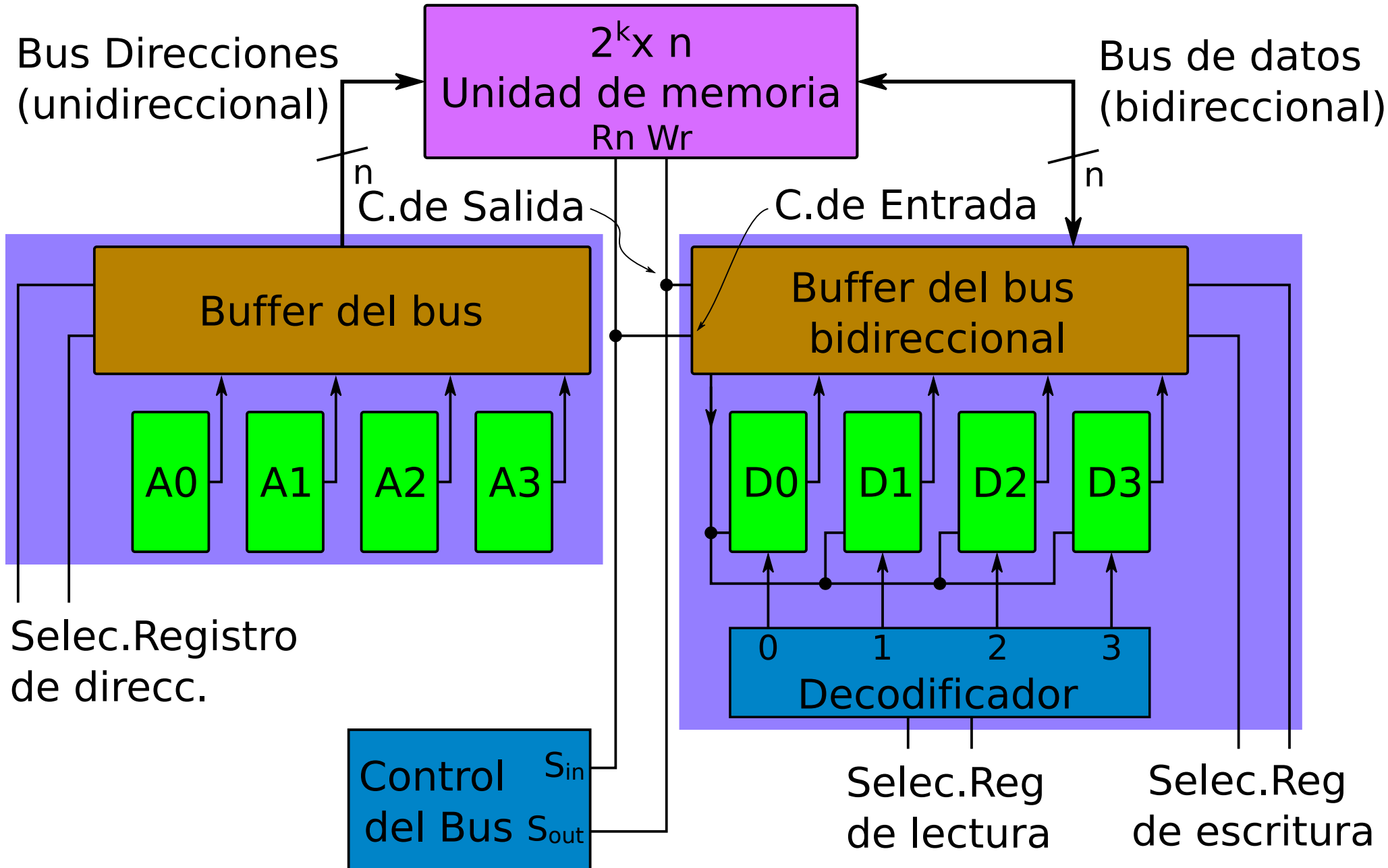
- Son operaciones desde y hacia una unidad de memoria
- Lectura: transferencia desde una unidad de memoria

$$DR \leftarrow M[AR]$$

- Escritura: transferencia hacia una unidad de memoria

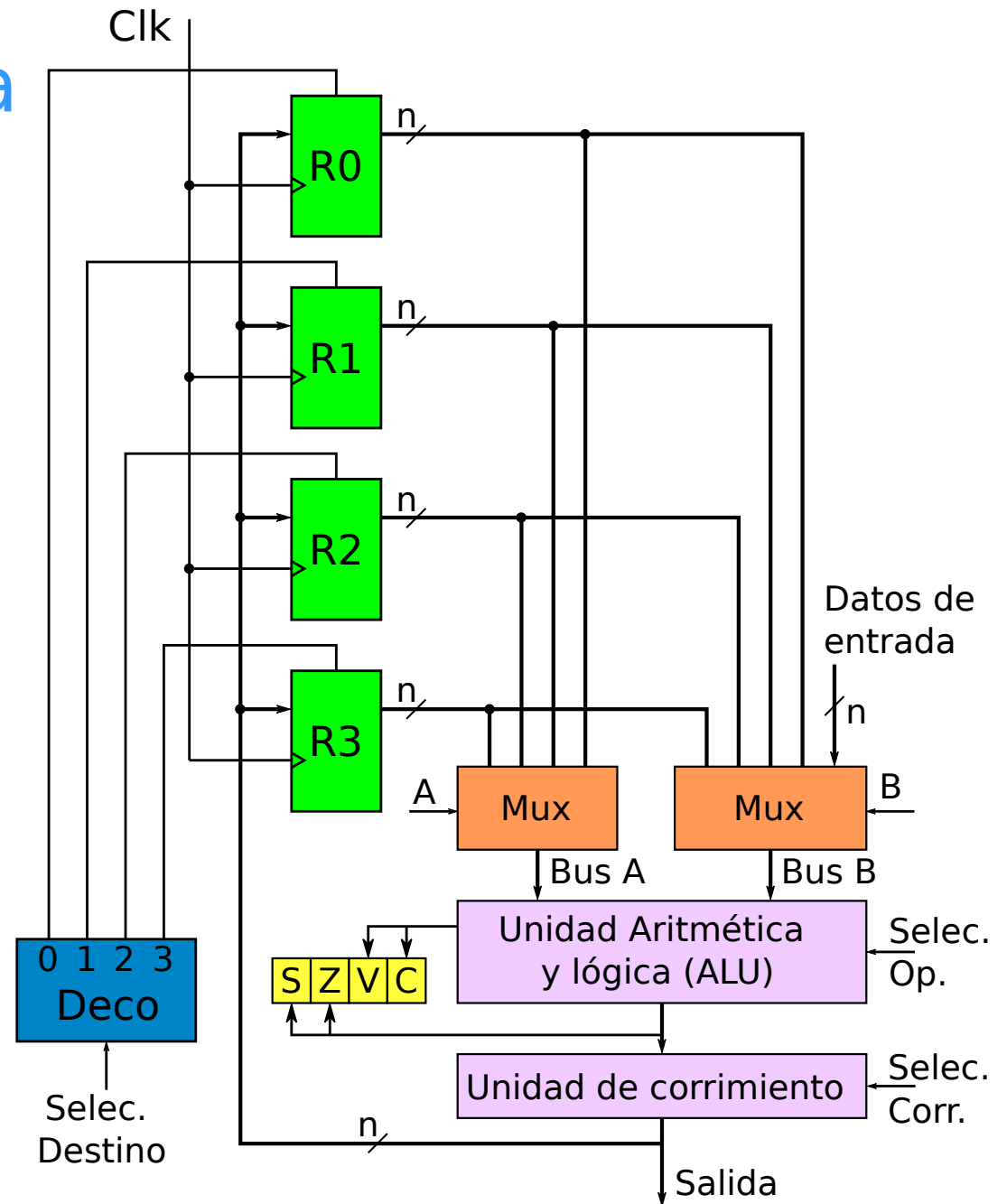
$$M[AR] \leftarrow DR$$

# Transferencia de la Memoria



# Unidad Procesadora

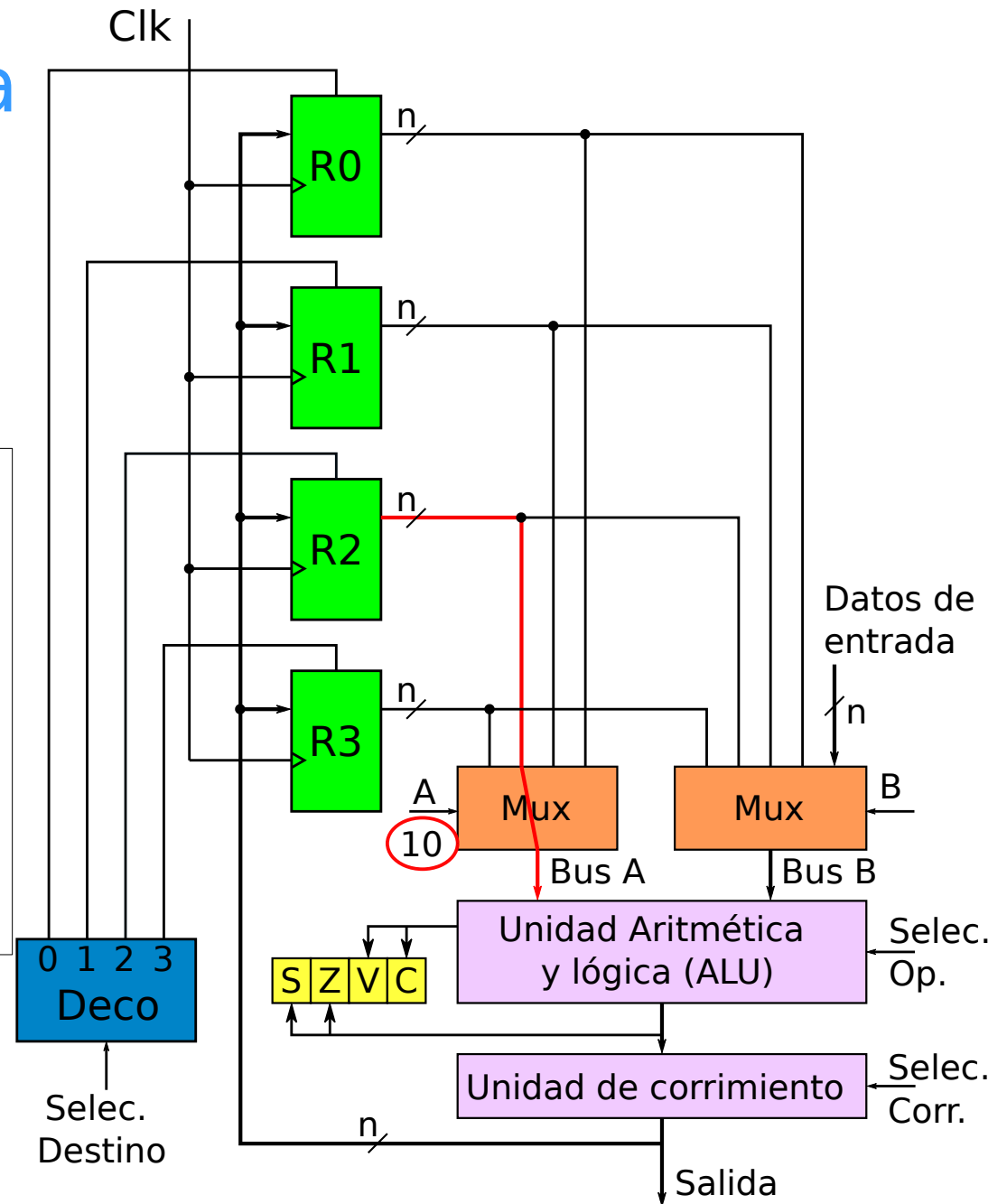
$$R1 \leftarrow R2 + R3$$



# Unidad Procesadora

$$R1 \leftarrow R2 + R3$$

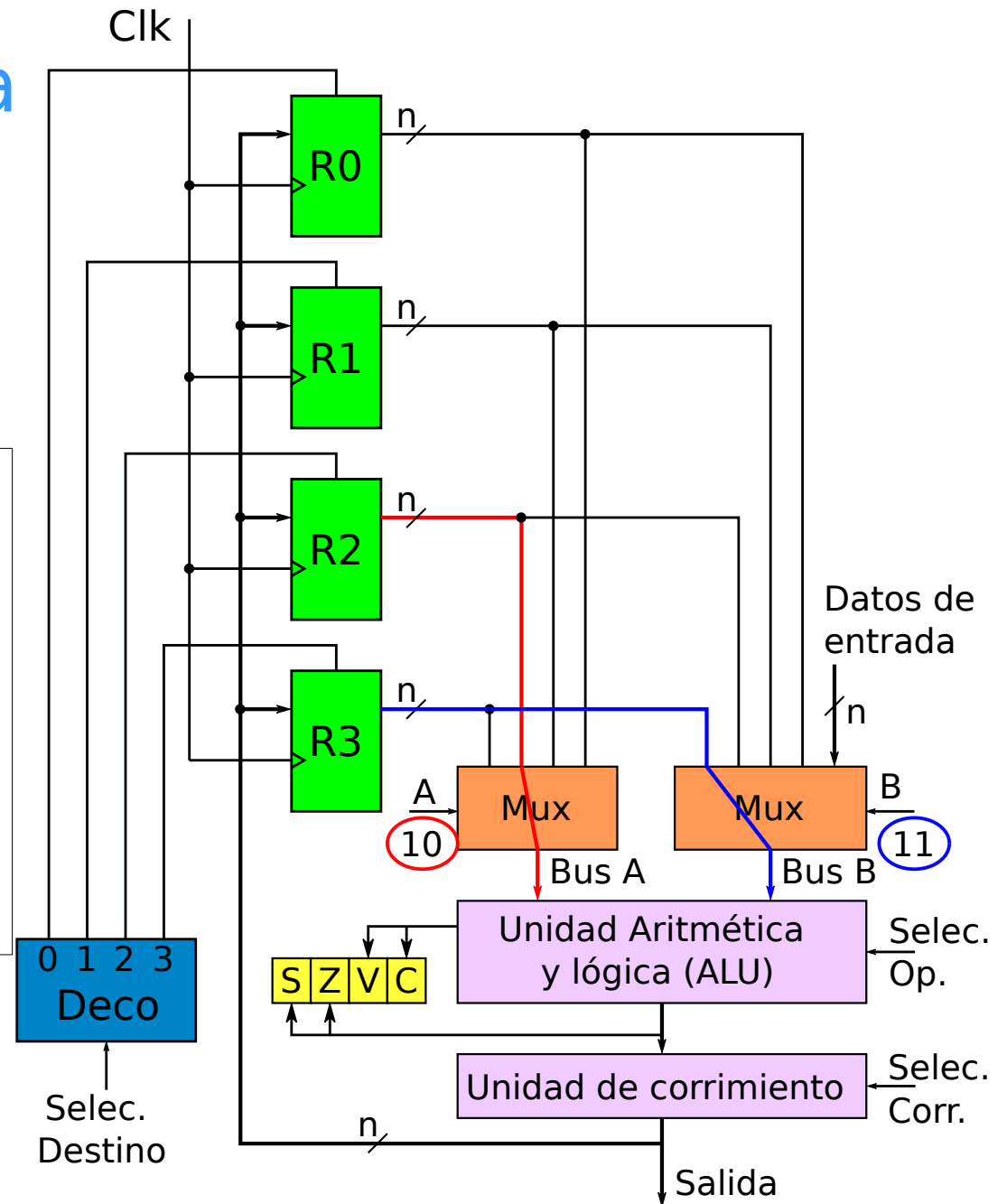
- MUX A Selec R2 en bus A



# Unidad Procesadora

$$R1 \leftarrow R2 + R3$$

- MUX A Selec R2 en bus A
- MUX B Selec R3 en bus B

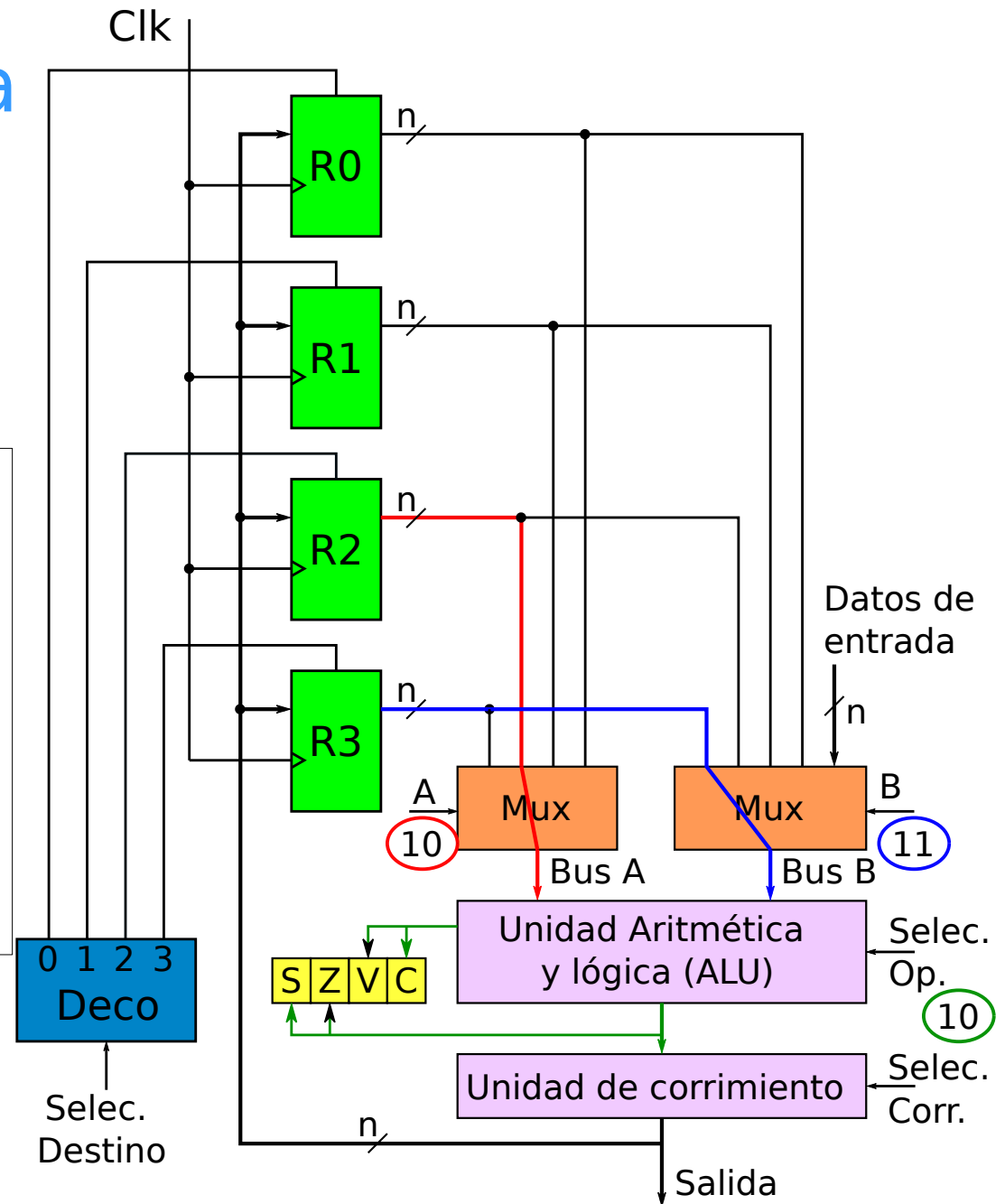




# Unidad Procesadora

$$R1 \leftarrow R2 + R3$$

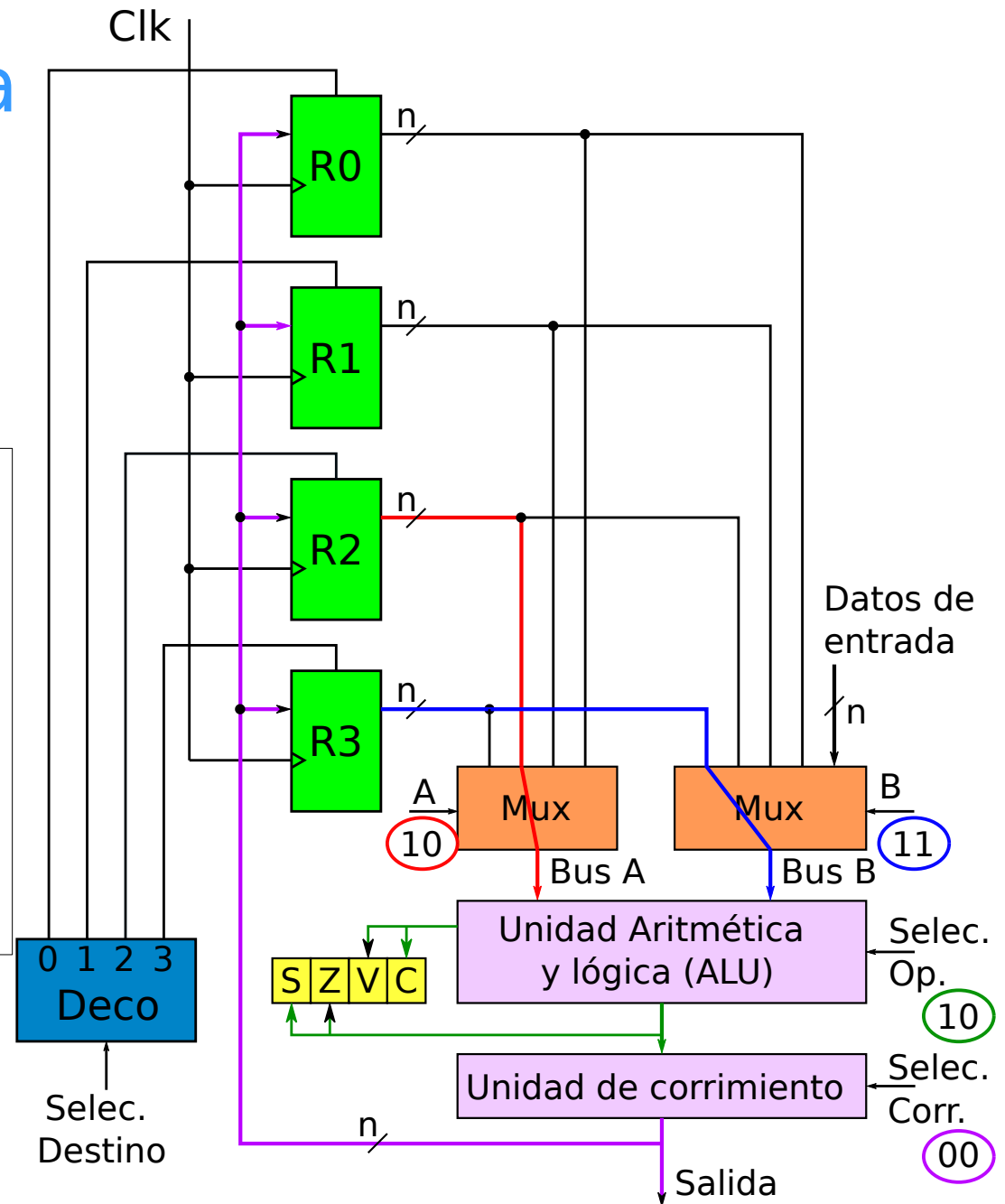
- MUX A Selec R2 en bus A
- MUX B Selec R3 en bus B
- ALU selec op. A+B



# Unidad Procesadora

$$R1 \leftarrow R2 + R3$$

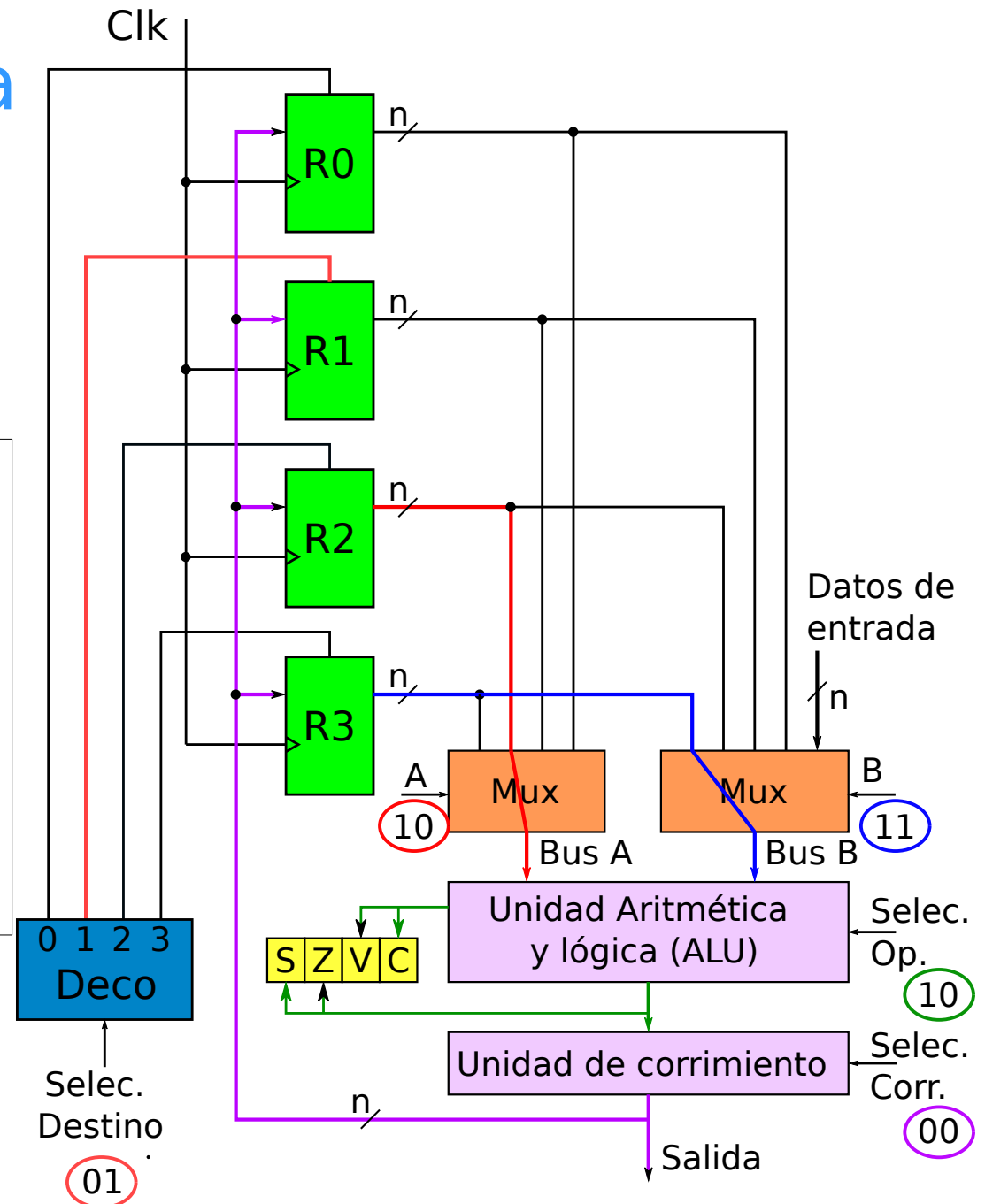
- MUX A Selec R2 en bus A
- MUX B Selec R3 en bus B
- ALU selec op. A+B
- Selec de Corrimiento para solo transferencia



# Unidad Procesadora

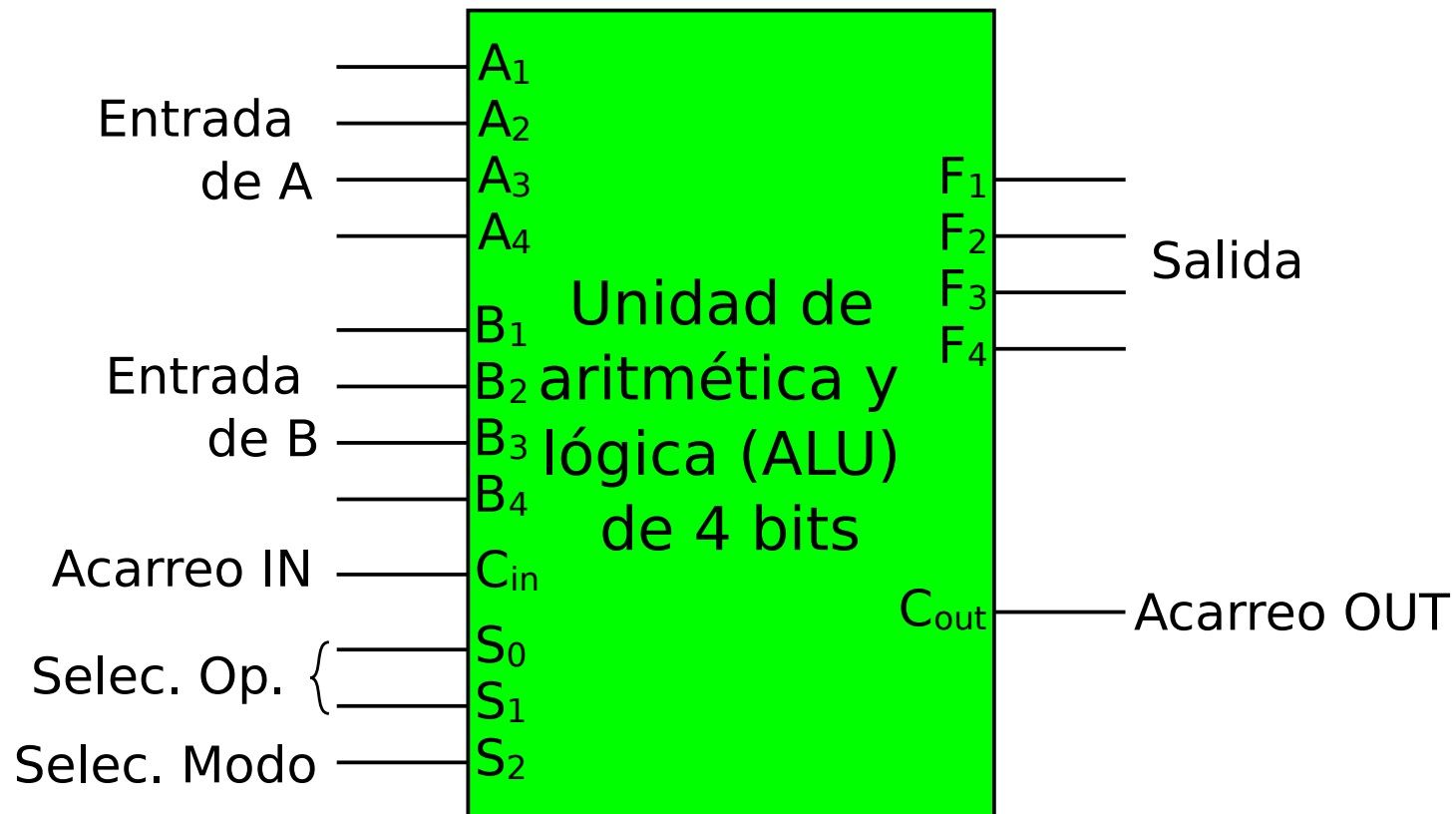
$$R1 \leftarrow R2 + R3$$

- MUX A Selec R2 en bus A
- MUX B Selec R3 en bus B
- ALU selec op. A+B
- Selec de Corrimiento para solo transferencia
- Selec Destino R1



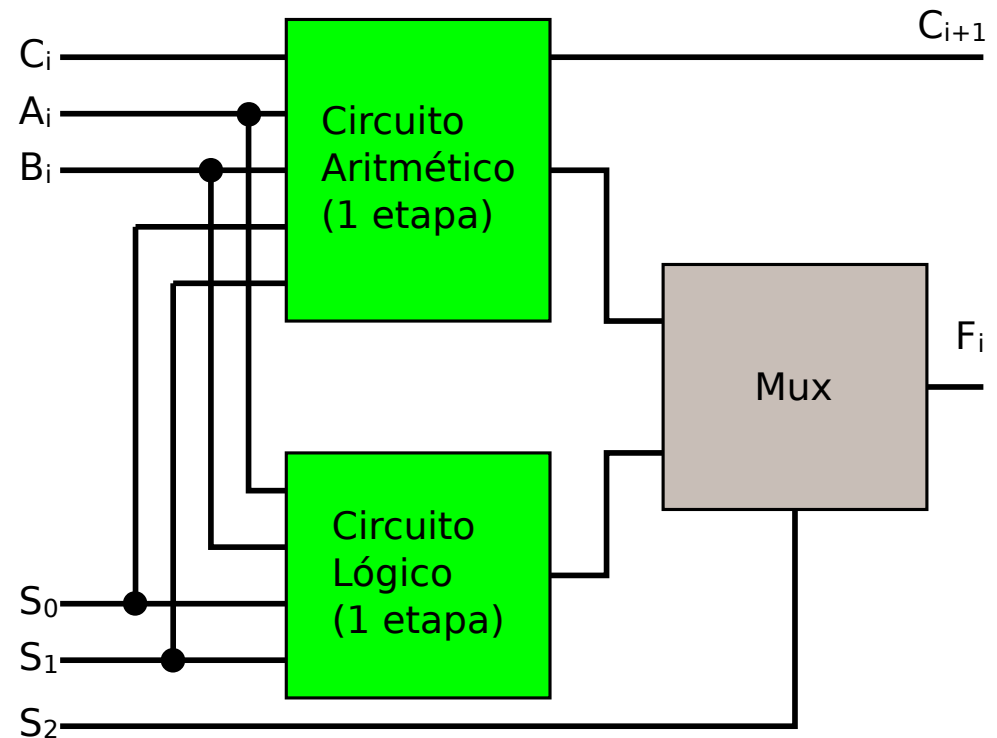
# Unidad Aritmética y Lógica

- Circuito combinatorio que realiza un conjunto de microoperaciones aritméticas y lógicas.



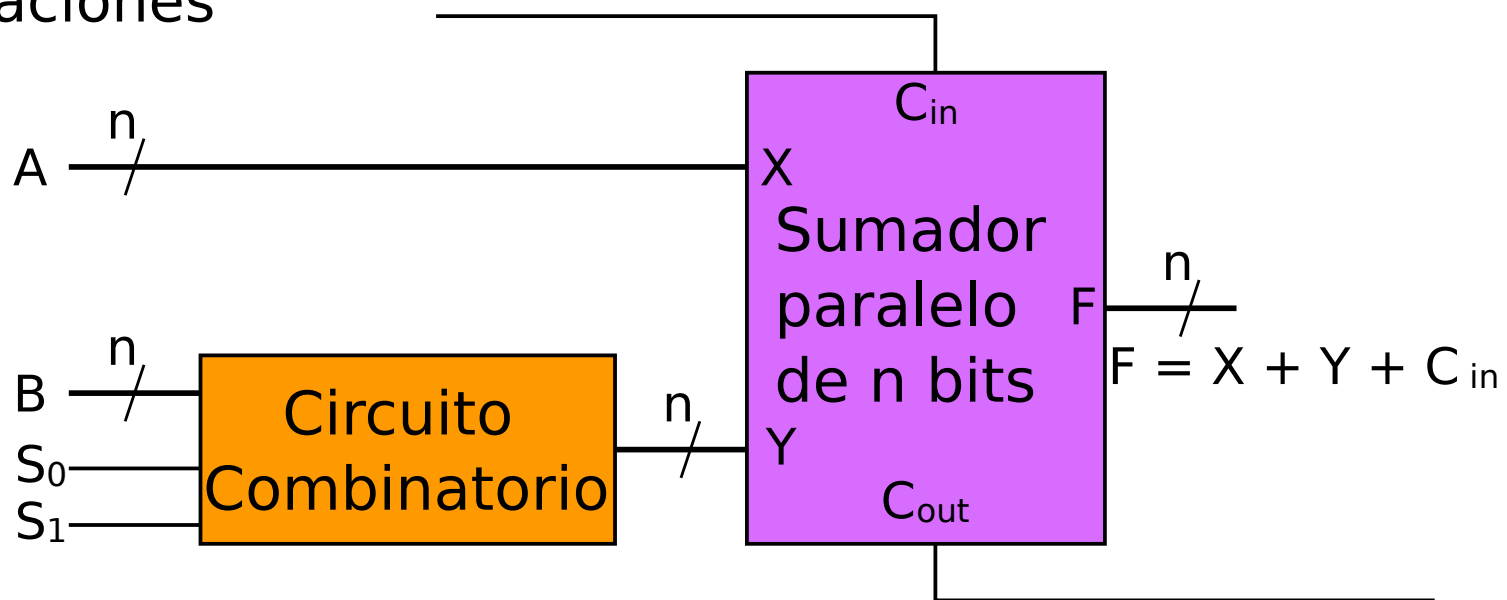
# Unidad Aritmética y Lógica

- El modelo típico se realiza en 3 etapas
  - Sección Aritmética
  - Sección Lógica
  - Combinación de ambas secciones



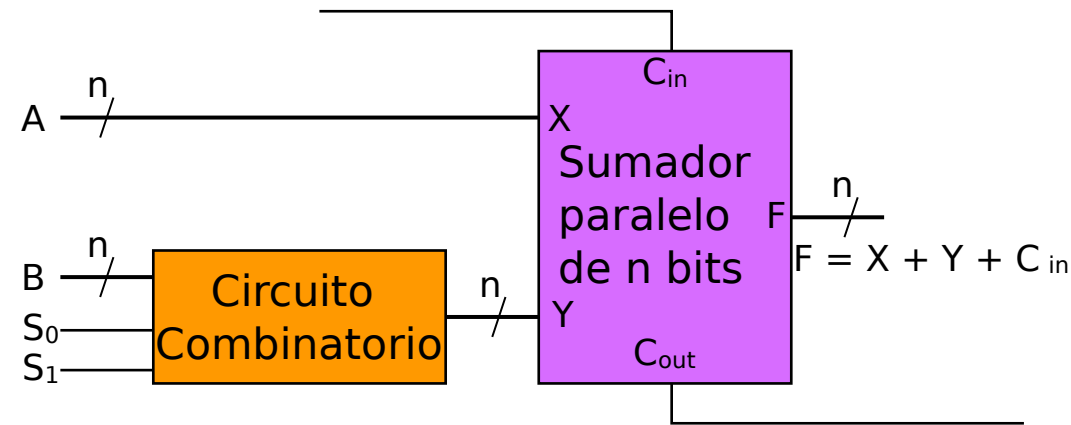
# ALU - Sección Aritmética

- Formado por un conjunto de sumadores en paralelo
- Mediante el control de las entradas a ese sumador, es posible obtener diferentes operaciones.
- En el diagrama las entradas son A y B
- Modificando la entrada B es posible obtener distintas operaciones



# ALU - Sección Aritmética

- La salida del sumador es  $F = X + Y + C_{in}$
- Donde X es A e Y puede tener  $2^2$  posibles salida.
- Si consideramos  $C_{in}$  como otra entrada que puede ser 0 o 1



$$F = A + Y + C_{in}$$

S0	S1	Entrada en Y	F para $C_{in} = 0$		F para $C_{in} = 1$	
0	0	0	$F = A$	transferencia	$F = A + 1$	incremento
0	1	B	$F = A + B$	suma	$F = A + B + 1$	
1	0	$\bar{B}$	$F = A + \bar{B}$		$F = A + \bar{B} + 1$	resta
1	1	1	$F = A - 1$	decremento	$F = A$	transferencia

# ALU - Sección Aritmética

- El circuito puede ser resuelto como:
  - n multiplexores con n sumadores
  - Diseño del circuito combinatorio en cada etapa, para reducir el número de compuertas

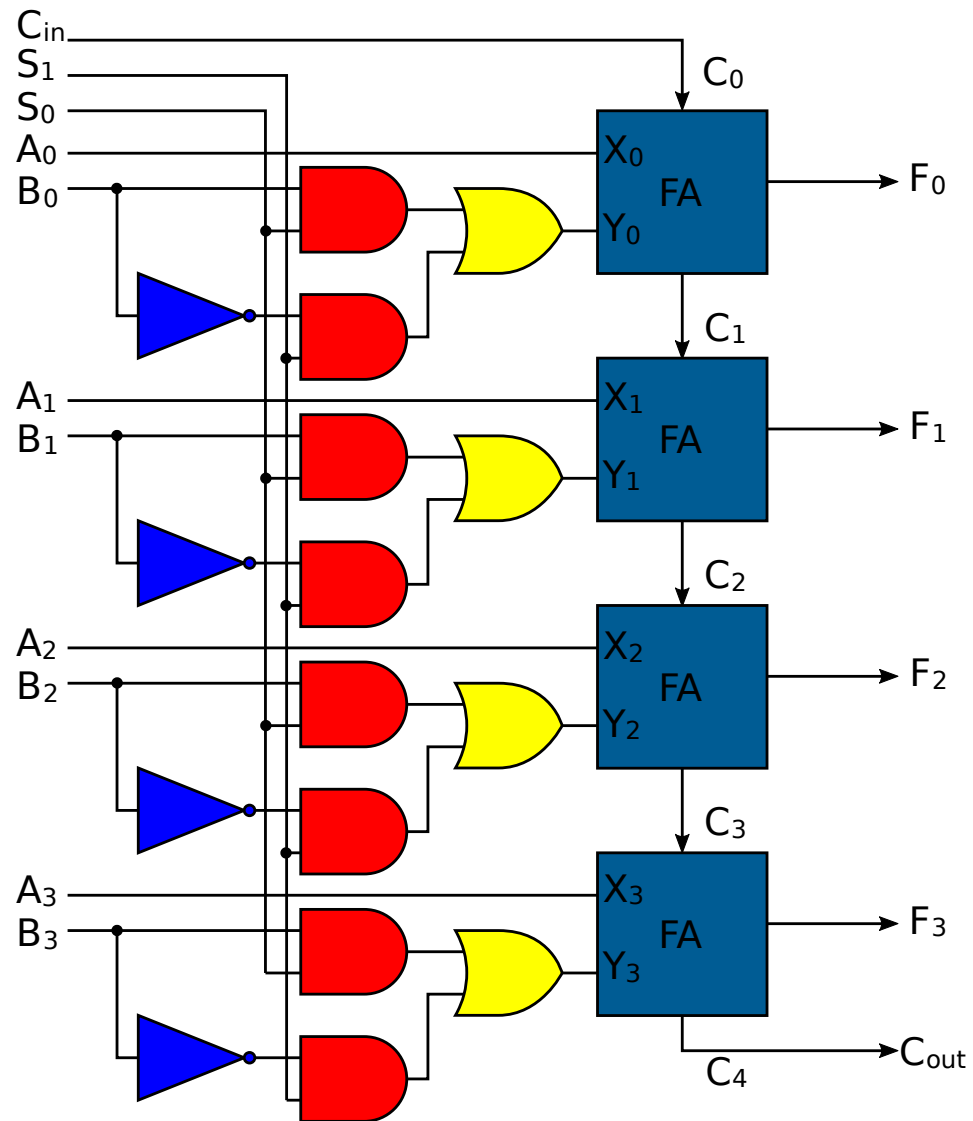
$$Y_i = B_i S_0 + \bar{B}_i S_1$$

Entradas			Salidas	
$S_1$	$S_0$	$B_i$	$Y_i$	
0	0	0	0	$Y_i=0$
0	0	1	0	
0	1	0	0	$Y_i=B_i$
0	1	1	1	
1	0	0	1	$Y_i=\bar{B}_i$
1	0	1	0	
1	1	0	1	$Y_i=1$
1	1	1	1	

$S_1 \backslash S_0 B_i$	00	01	11	10
0			1	
1	1		1	1

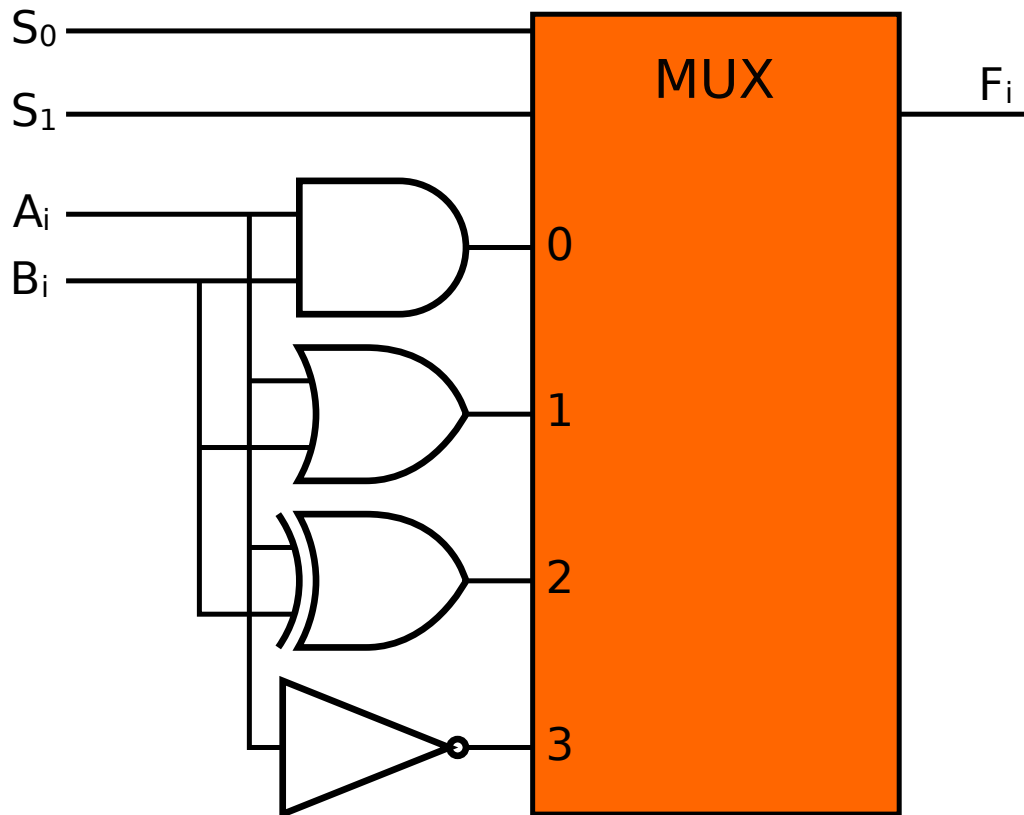


# ALU - Sección Aritmética



# Etapa Lógica

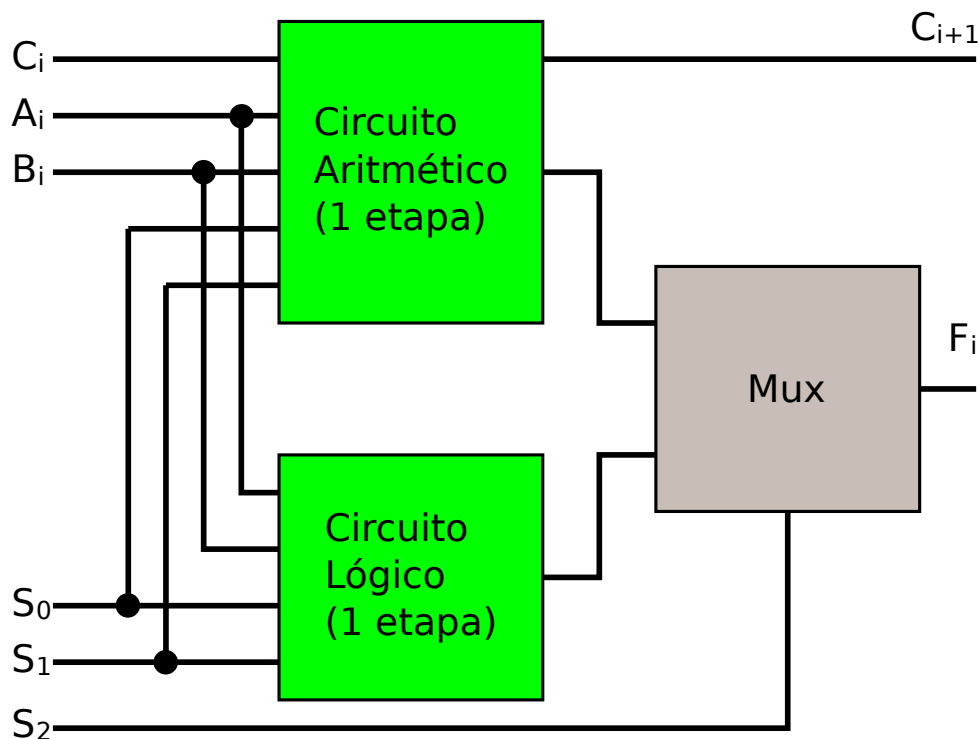
## Etapa de un circuito lógico



$S_0$	$S_1$	Salida	Función
0	0	$F_i = A_i \wedge B_i$	AND
0	1	$F_i = A_i \vee B_i$	OR
1	0	$F_i = A_i \oplus B_i$	XOR
1	1	$F_i = \bar{A}_i$	Complemento

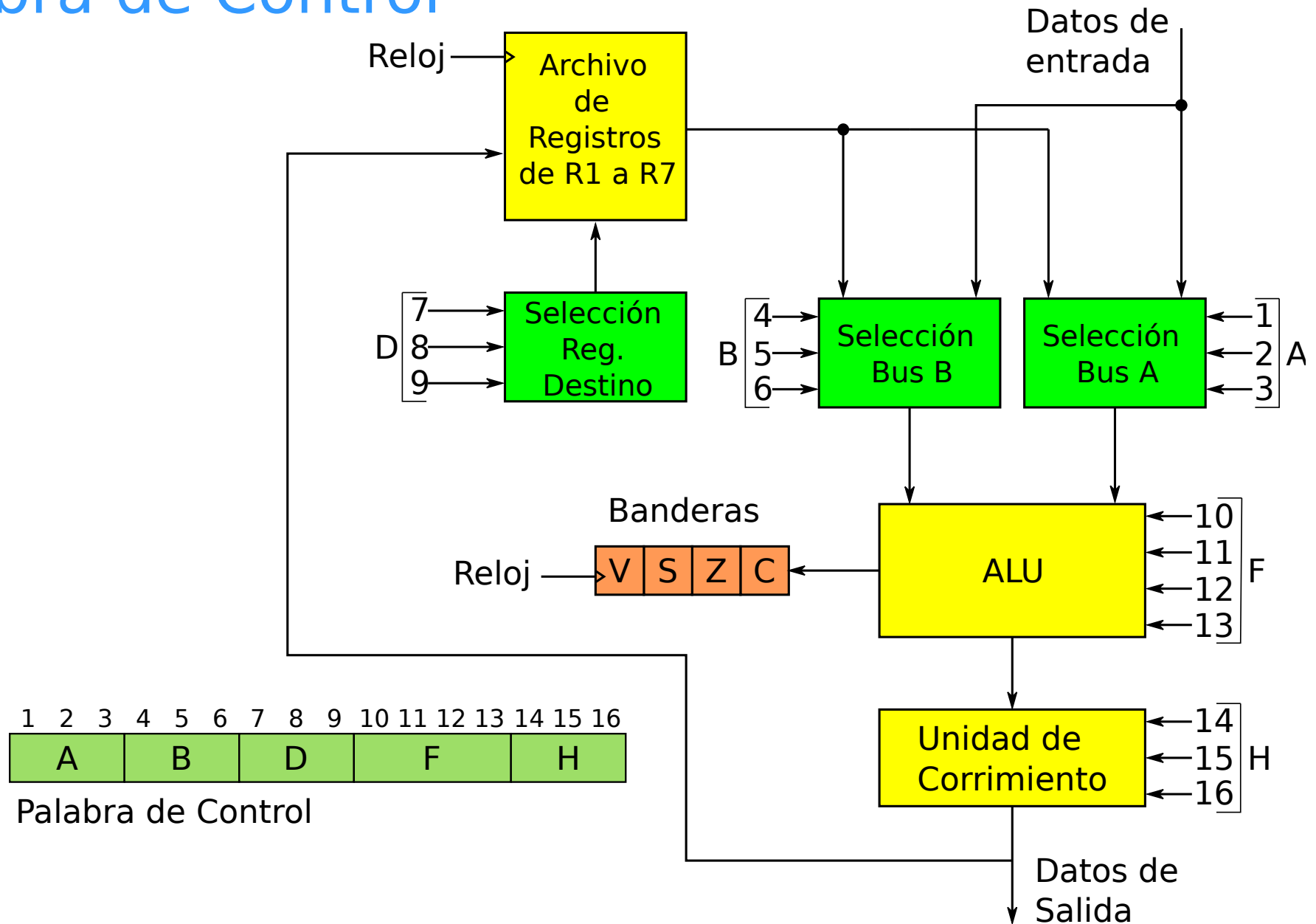
# Unidad Aritmética y Lógica

- La combinación del circuito lógico con el aritmético forma la ALU.
- Un multiplexor y una tercera variable de selección, permite decidir que etapa se utilizará



S2	S1	S0	Cin	Función
0	0	0	0	$F = A$
0	0	0	1	$F = A + 1$
0	0	1	0	$F = A + B$
0	0	1	1	$F = A + B + 1$
0	1	0	0	$F = A + \bar{B}$
0	1	0	1	$F = A + \bar{B} + 1$
0	1	1	0	$F = A - 1$
0	1	1	1	$F = A$
1	0	0	0	$F = A \wedge B$
1	0	1	0	$F = A \vee B$
1	1	0	0	$F = A \oplus B$
1	1	1	0	$F = \bar{B}$

# Palabra de Control



# Ejemplos de palabras de control

Código	A	B	A	F con Cin=0	F con Cin=1	H
000	Entrada	Entrada	Ninguno	$F=A$	$F=A + 1$	Sin corr
001	R1	R1	R1	$F=A + B$	$F=A + B + 1$	SHL
010	R2	R2	R2	$F=A + \bar{B}$	$F=A - B$	SHR
011	R3	R3	R3	$F=A - 1$	$F=A$	Bus = 0
100	R4	R4	R4	$F = A \wedge B$		
101	R5	R5	R5	$F = A \vee B$		ROL
110	R6	R6	R6	$F = A \oplus B$		ROR
111	R7	R7	R7	$F = \bar{A}$		

# Ejemplos de palabras de control

Microoperación	A SoA	B SoB	D Tar	F ALU	H Shift	A	B	D	F	H
<b>R1 ← R2 - R3</b>	R2	R3	R1	F=A-B	Sin Corr	010	011	001	0101	000
<b>R4 ← shr (R2 + R5)</b>	R2	R6	R4	F=A+B	SHR	010	110	100	0010	010
<b>R7 ← R7 + 1</b>	R7		R7	F=A+1	Sin Corr	111	000	111	0001	000
<b>R2 ← R1</b>	R1		R2	F=A	Sin Corr	010	000	001	0000	000
<b>Salida ← R3</b>	R3		Nin	F=A	Sin Corr	011	000	000	0000	000
<b>R4 ← rol (R4)</b>	R4		R4	F=A	ROL	100	000	100	0000	101
<b>R4 ← 0</b>			R5		Bus=0	000	000	101	0000	111



- Bibliografía

Morris Mano. Ingeniería computacional: diseño del hardware.  
Prentice Hall, 1991. Capítulo 7



**¿ Preguntas ?**