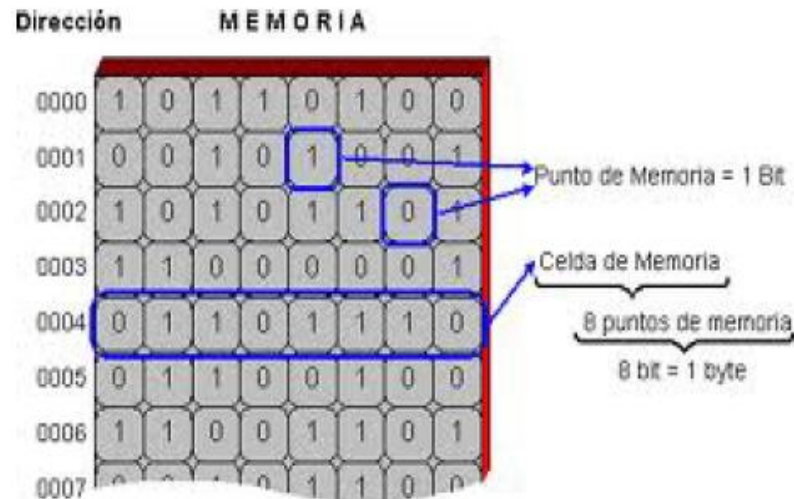


# MEMORIA PRINCIPAL

---



Lic. Denise Pari



## Memoria principal

- Conjunto de celdas, todas del mismo tamaño (cantidad de bits).
- Cada celda se accede a través de su **dirección**.
- Permite leer o escribir celdas.
- La celda es la **unidad direccionable** mas pequeña: cada una está asociada a una dirección diferente.

# MEMORIA PRINCIPAL

## RAM (ACCESO ALEATORIO)

- Se utiliza para almacenar temporalmente datos y programas.
- Es **volátil**: Pierde su contenido al desconectar la energía eléctrica

### **Memoria de Acceso Aleatorio**

Es posible acceder a cualquier celda con el mismo consumo de tiempo (**¡No es azar!**)

## Espacio direccionable

Conjunto de todas las direcciones de las celdas de memoria

- Si la memoria tiene  $2^m$  celdas, se necesitan **m** bits para expresar las direcciones  $[0 : 2^m - 1]$
- Las celdas **se agrupan en palabras**. La palabra es la unidad *natural* de organización de la memoria. El tamaño de la palabra suele coincidir con lo necesario para representar números, y puede ser de 1 celda.
- Unidad de transferencia: cantidad de bits que se transmiten al mismo tiempo.

LAS DIRECCIONES NO SE GUARDAN EN MEMORIA!

# EJECUCIÓN DE UNA INSTRUCCIÓN

REGISTROS QUE INTERVIENEN:



PC \_Program Counter

IR\_ Instruction Register

MAR\_Memory Address Register

MBR\_Memory Buffer Register

- **PC:** Contiene la dirección de la instrucción a buscar
- **IR:** Contiene la instrucción buscada mas recientemente. Muchas veces este registro tiene mas capacidad que los demás, a los fines de almacenar la instrucción completa.
- **MAR:** Contiene la dirección de una posición de memoria.
- **MBR:** Contiene el dato a escribir en una posición de memoria o el dato contenido en una posición de memoria leído mas recientemente.

# FUNCIONAMIENTO LECTURA

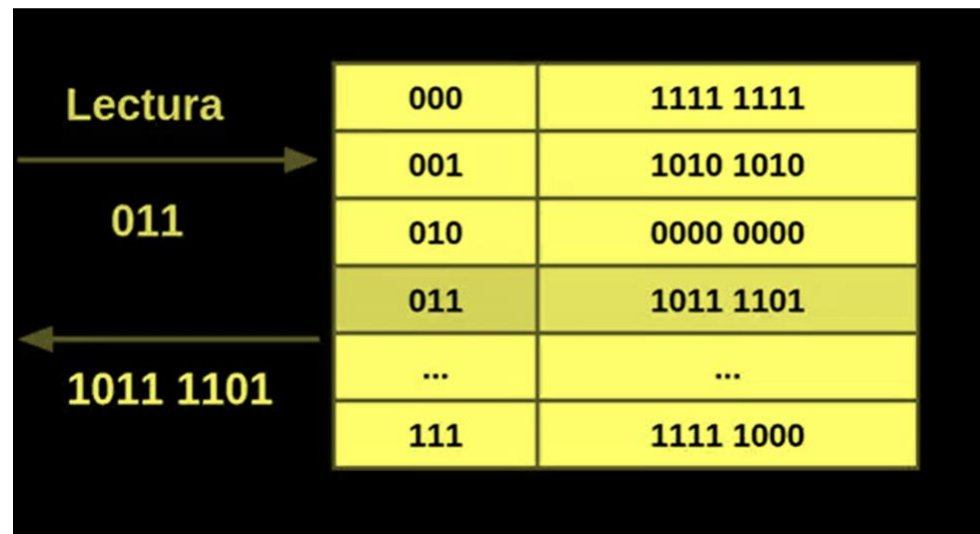
1) UC Solicita dato almacenado en Memoria

2) UC pone dirección en el MAR y **activa señal de lectura**  
( se activa señal de lectura en el BUS DE CONTROL)

3) MEMORIA activa un deco que elige la celda específica y copia contenido en BUS DE DATOS.

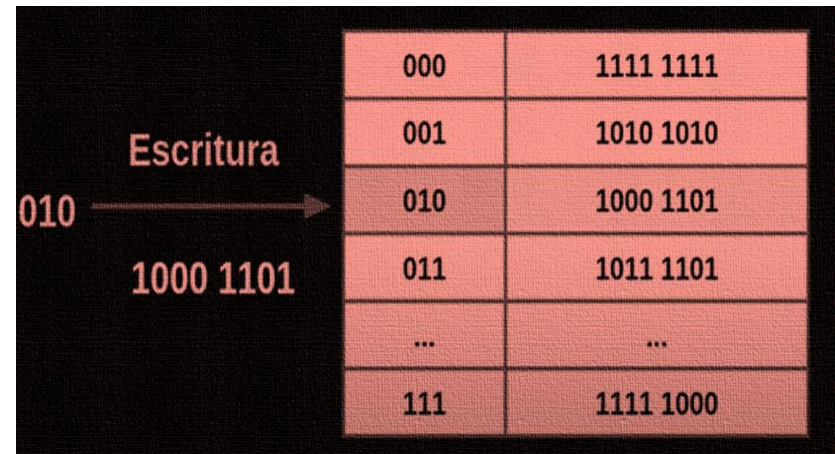
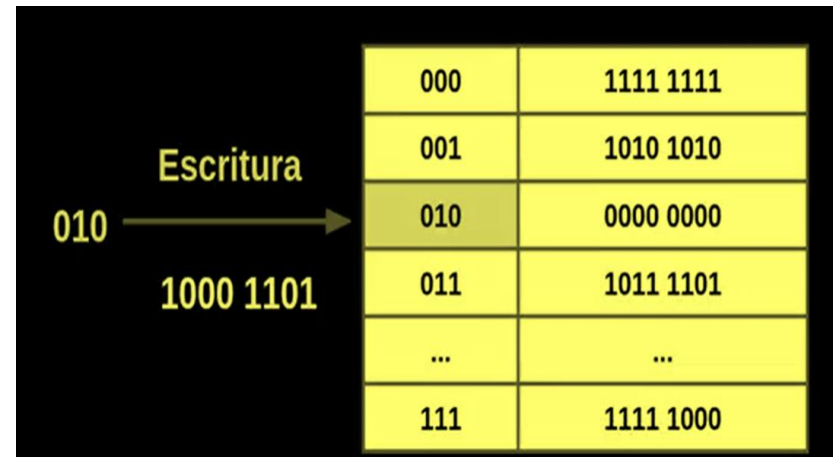
4) Dato en la MBR de la CPU.

5) BUS DE CONTROL activa señal de que el dato está DISPONIBLE.



# FUNCIONAMIENTO DE ESCRITURA

- 1) UC necesita de almacenar un DATO en una celda de MEMORIA.
- 2) La UC pone el dato en el registro MBR y una dirección en el registro MAR.
- 3) Luego activa una señal de escritura en el bus de control para indicar la operación de escritura.
- 4) La Memoria activa un circuito decodificador que permite elegir la celda específica, pero además toma el dato del bus de datos y lo copia en dicha celda.
- 5) Mediante el bus de control se indica a la UC que la operación ha finalizado.





# ENTONCES:

## Memoria

- Lectura

Se envía:

- Señal de lectura

- Dirección

Recibe:

- Valor

- Escritura

Se envía:

- Señal de escritura

- Dirección

- Valor

# INTERCONEXIÓN

La **memoria principal** y la **unidad de control** son circuitos que deben comunicarse para intercambiar datos y direcciones.

¿Cómo se comunican?

A través de un medio de transmisión compartido entre la CPU y la Memoria Principal que se denomina Bus del sistema.

# Bus de sistema

Es un conjunto de señales que permite:

- transmitir datos desde y hacia la memoria
- transmitir direcciones hacia la memoria
- transmitir señales de control desde y hacia la memoria principal

El bus de sistema está formado por tres buses para cumplir los diferentes objetivos y reciben los nombres:

Bus de direcciones

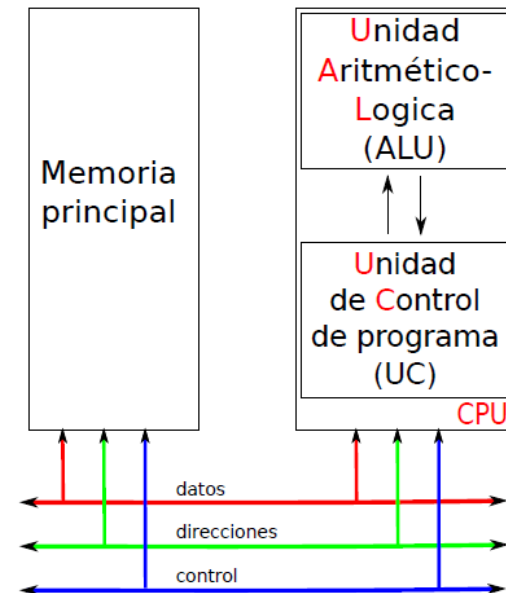
Bus de datos

Bus de control

# ¿Cómo le llegan los **datos**/las **direcciones**/las **señales** a la memoria principal?

¿Que se necesita?

- Transmitir **datos** desde y hacia la memoria principal
- Transmitir **direcciones** hacia la memoria principal
- Transmitir **señales de control** hacia la memoria principal



## Bus

- Medio de transmisión compartido entre 2 o mas dispositivos
- Conjunto de señales (cables) agrupadas con un determinado objetivo

# Características:

Cada bus tiene una determinada cantidad de líneas (ancho del bus) y cada línea transmite un bit a la vez.

Entonces el ancho del bus determina cuántos bits se pueden transmitir en paralelo.

En el caso del bus de direcciones, el ancho determina el espacio direccionable, y en el caso del bus de datos determina el tamaño de palabra.

# BUS DE DIRECCIONES

DEBE NOMBRAR CADA CELDA (EN ESTE CASO 8  
DIFERENTES  $2^3$ )  
(MENOS BITS QUE LA CANT TOTAL DE CELDAS)

Bus de direcciones

8 celdas

— — —

Ancho de bus: 3 bits

000	1111 1111
001	1010 1010
010	0000 0000
011	1011 1101
...	...
111	1111 1000

# BUS DE DATOS

- DEBO PODER PASAR CADA UNA DE LAS DIFERENTES COMBINACIONES QUE PUEDO OBTENER CON 8 BITS

<b>Bus de datos</b>	
<b>8 bits</b>	
-----	
<b>Ancho de bus: 8 bits</b>	
000	1111 1111
001	1010 1010
010	0000 0000
011	1011 1101
...	...
111	1111 1000

# ENTONCES...

## Ancho de Bus

- **Direcciones** hacia la memoria  
Con  $2^n$  celdas:  $n$
- **Datos** desde y hasta la memoria  
Con  $m$  bits por celda:  $m$
- Tamaño de la memoria:  $2^n \times m$



# EJEMPLO:

## Ancho de Bus

- Si tengo una memoria de 32 celdas y la capacidad de cada celda es 8 bits:
  - ¿Cuántas líneas debe tener el bus de datos? 8
  - ¿Cuántas líneas debe tener el bus de direcciones ? 5 ( $2^5 = 32$ )
  - ¿Cual es el tamaño de mi memoria? 256 ( $2^5 \times 8 = 256$ )

## Bus de control

Transmite señales de temporización y de comando hacia la memoria.

- La temporización indica la validez de los datos y direcciones transmitidos en los otros buses
- Los comandos indican el tipo de operación que debe llevar a cabo la memoria (lectura o escritura)



## EJEMPLO

El bus está ocupado  
Necesito usar el bus  
Listo!

## ENTONCES PODEMOS RESUMIR EL CONCEPTO DE BUSES:

### **Bus de datos**

Transporta datos entre los módulos. El ancho del bus determina cuantos bits pueden transmitirse simultáneamente (en paralelo)

### **Bus de direcciones**

Indica el destino o el origen del dato que está en el bus de datos. El ancho de este bus determina el **espacio direccionable**.

# INTERCONEXIÓN MEDIOS COMPARTIDOS

Muchos dispositivos se conectan al bus y la señal transmitida por cualquiera de ellos está disponible para ser leída por cualquier otro.

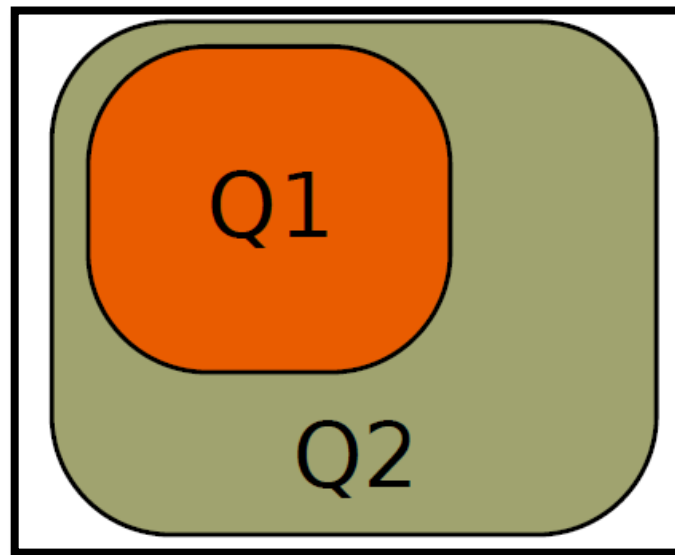


Si mas de un dispositivo transmite al mismo tiempo sus señales colisionan



Se necesita algún mecanismo de control y sincronización para asegurar que solo uno transmita al mismo tiempo.

# ARQUITECTURA Q2



## Memoria (Q2)

Contenido de Celdas de Memoria (16 bits)



Dirección  
16 bits



0x0000	0xFFFF
0x0001	0xAAAA
0x0002	0x0880
0x0004	0x0E80
0x0005	0x0000
...	...

- Tiene 8 registros de uso general de 16 bits: R0..R7
- Tiene instrucciones de 2 operandos:

instrucción	sintaxis	efecto
ADD	ADD destino, origen	$\text{destino} \leftarrow \text{destino} + \text{origen}$
SUB	SUB destino, origen	$\text{destino} \leftarrow \text{destino} - \text{origen}$
MUL	MUL destino, origen	$(R7, \text{destino}) \leftarrow \text{destino} * \text{origen}$
DIV	DIV destino, origen	$\text{destino} \leftarrow \text{destino} \% \text{origen}$
MOV	MOV destino, origen	$\text{destino} \leftarrow \text{origen}$

- Tiene direcciones de 16 bits.
- Los operandos pueden estar en registros, ser constantes o estar en direcciones de memoria.

# MODOS DE DIRECCIONAMIENTO

**Q2** permite 3 modos de direccionamiento:

- modo registro: el valor buscado está en un registro
- modo inmediato: el valor buscado está codificado dentro de la instrucción
- modo **directo**: el valor buscado está contenido en una celda de memoria

Modo	Codificación
Inmediato	000000
<b>Directo</b>	<b>001000</b>
Registro	100rrr



# Formato de Instrucciones

## Ejercicio: Ensamblar MOV R1,[0003]

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Operación	CodOp
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Codificación
Inmediato	000000
Directo	001000
Registro	100rrr

## Ejercicio: Ensamblar MOV R1,[0003]

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Efecto	$R1 \leftarrow [0003]$
Código de operación	0001
Modo Destino	R1 está en modo registro: 100rrr
Modo Origen	[0003] está en modo directo: 001000



00011000010010000000000000000011

## Ejercicio: Ensamblar MOV [AAAA],[0003]

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Operación	CodOp
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Codificación
Inmediato	000000
Directo	001000
Registro	100rrr

## Ejercicio: Ensamblar MOV [AAAA],[0003]

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Efecto	$[AAAA] \leftarrow [0003]$
Código de operación	0001
Modo Destino	[AAAA] está en modo directo: 001000
Modo Origen	[0003] está en modo directo: 001000



0001001000001000101010101010100000000000000011

## Ejercicio: Ensamblar MOV [F0F0],R5

Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Operación	CodOp
MUL	0000
MOV	0001
ADD	0010
SUB	0011
DIV	0111

Modo	Codificación
Inmediato	000000
Directo	001000
Registro	100rrr

## Ejercicio: Ensamblar MOV [F0F0],R5

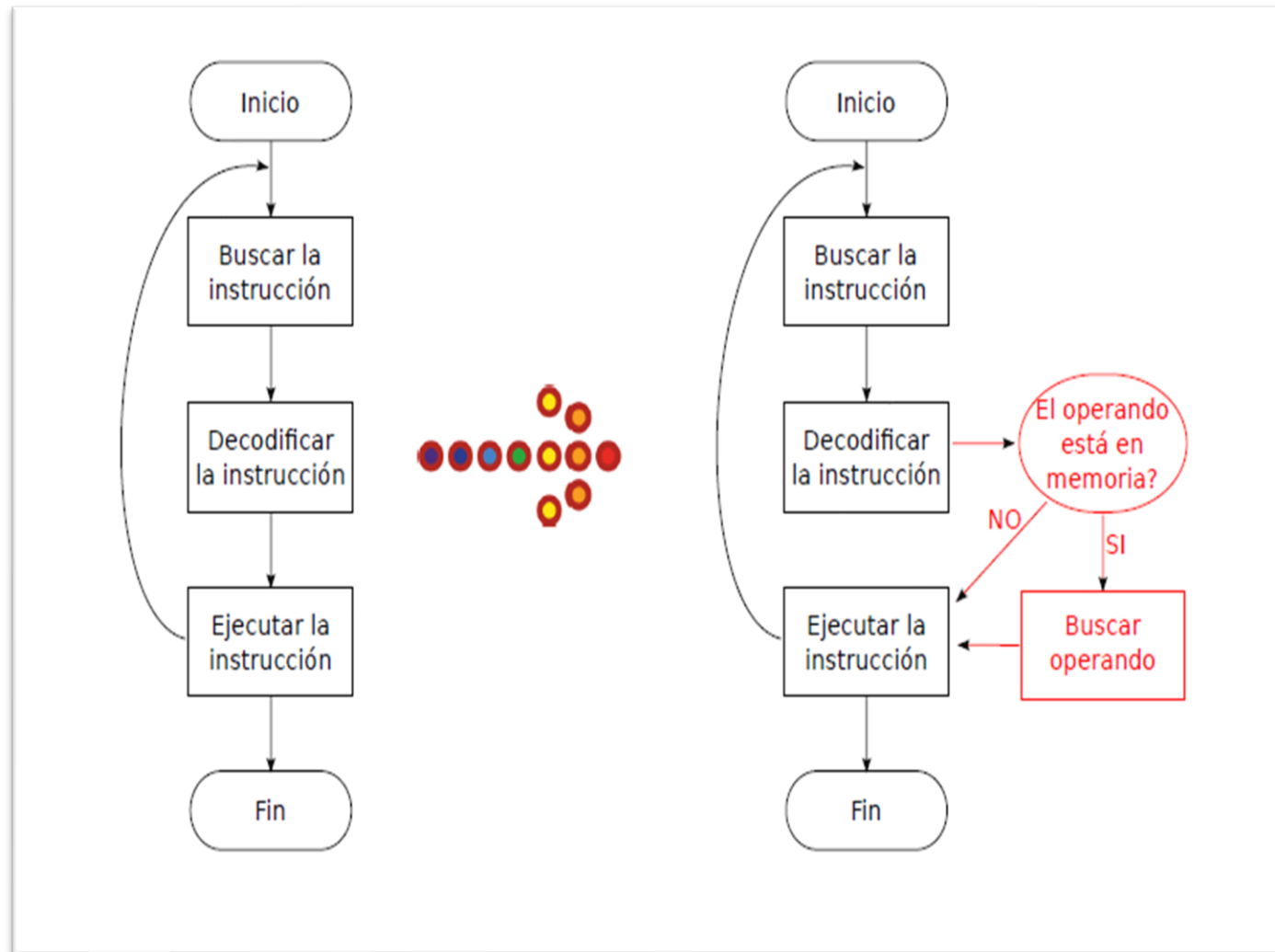
Cod_Op (4b)	Modo Destino (6b)	Modo Origen (6b)	Operando Destino (16b)	Operando Origen (16b)
----------------	----------------------	---------------------	---------------------------	--------------------------

Efecto	[F0F0] ← R5
Código de operación	0001
Modo Destino	[F0F0] está en modo directo: 001000
Modo Origen	R5 está en modo registro: 100101



0001001000100101111000011110000

# CICLO Q2



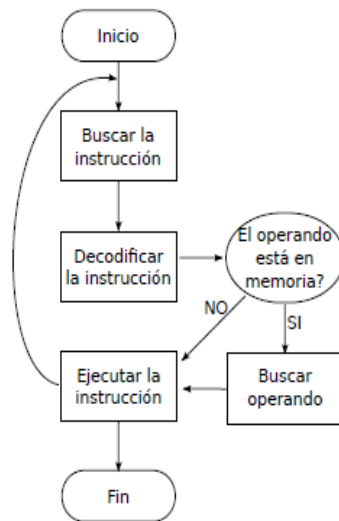


El operando  
está en  
memoria?

¿Cómo hace la Unidad de Control para responderlo?



Mediante los modos de direccionamiento



¿Cuánto tarda la ejecución de un programa?



¿De qué depende?



Accesos a memoria + tiempo de ejecución

# Cantidad de accesos de una instrucción

MOV R1, R2



Búsqueda de la instrucción



0001100001100010  
(1 celda)

Búsqueda de operandos



No tiene operandos en memoria

# Cantidad de accesos de una Instrucción

MOV R1, [0001]



Búsqueda de la instrucción



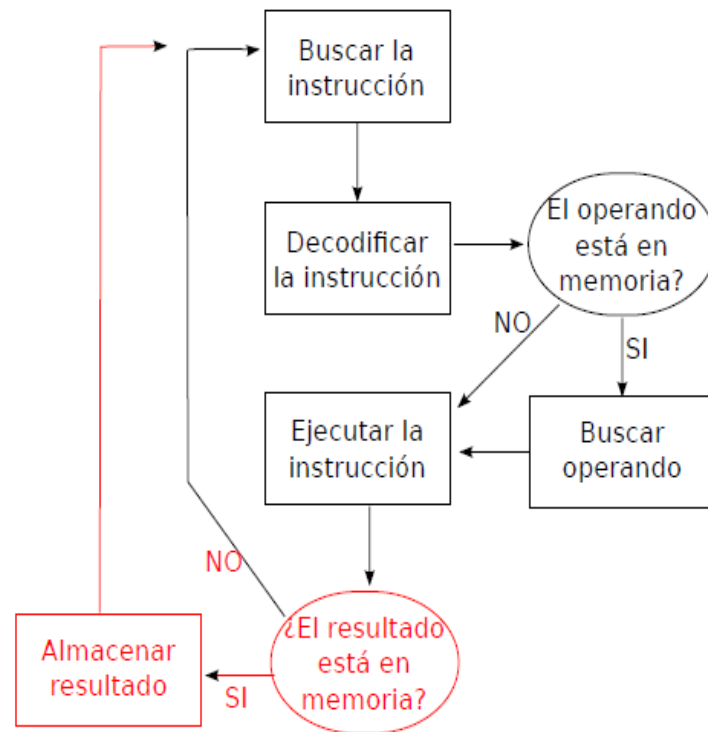
000110000100100000000000000000

(2 celdas)

Búsqueda de operandos

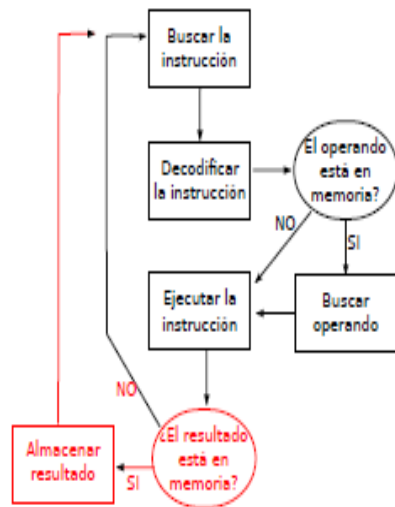


Se lee la celda 0001



¿Cuántos accesos tiene este programa?

MOV [0001],R1



MOV [0001],R1

- Búsqueda de instrucción: 2 celdas
- Búsqueda de operandos: 0 celdas
- Almacenamiento de operandos: 1 celda