Memoria Principal e Interconexión

Organización de computadoras

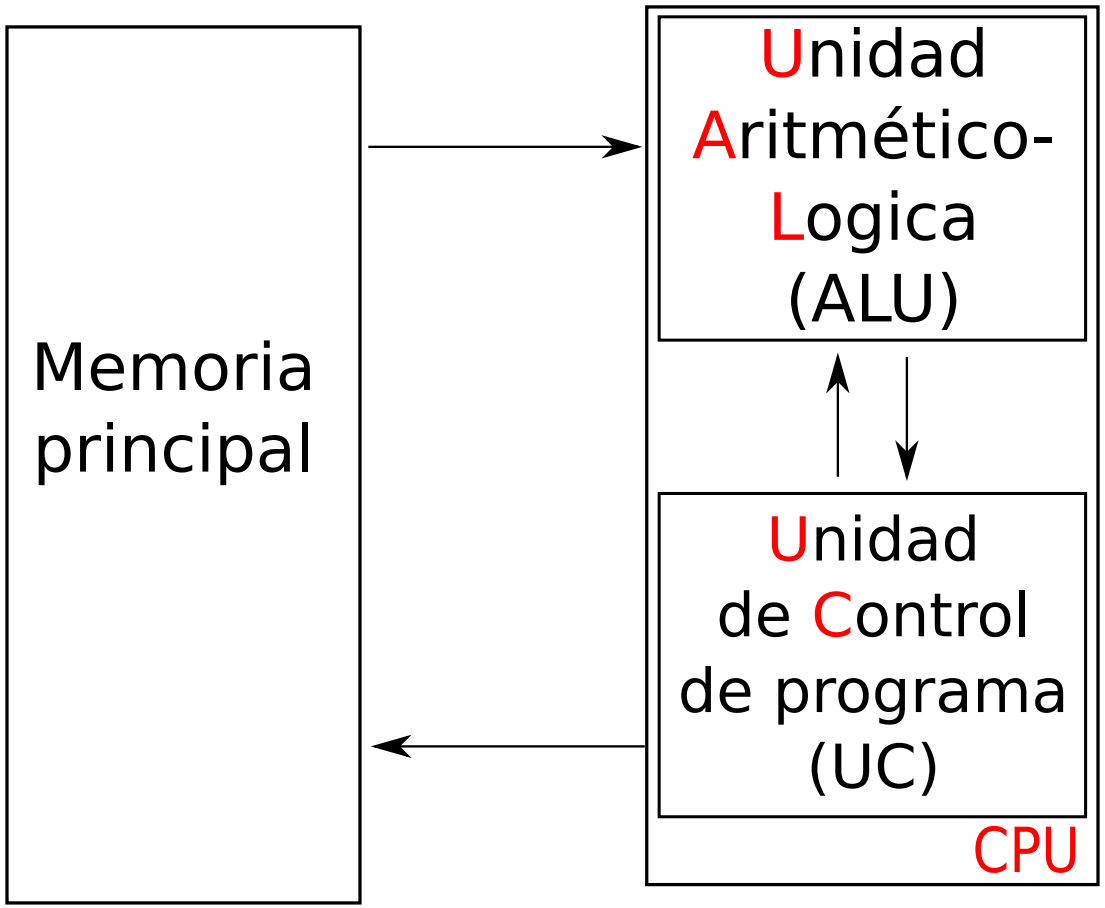
Universidad Nacional de Quilmes

8 de septiembre de 2014

Repaso

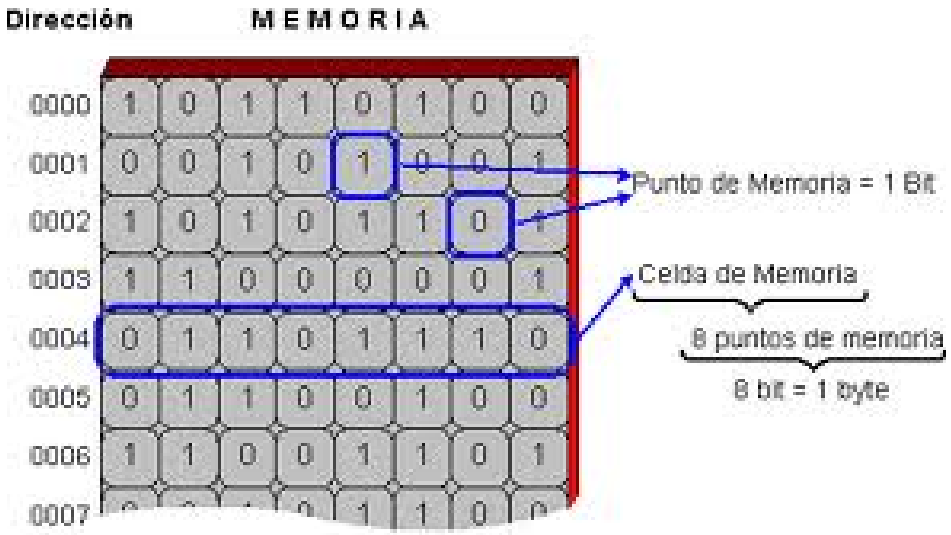
1. Ciclo de vida del programa
2. Dentro de la CPU
    1. Unidad de Control
    2. ALU
    3. Registros
3. Ensamblar y desensamblar: Jugamos a la panadería
4. Ciclo de ejecución de una instrucción: Decodificación
5. Computadora Q1

Arquitectura de Von Neumann



Memoria Principal

Arquitectura de Von Neumann: Memoria Principal



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.

Arquitectura de Von Neumann: Memoria Principal

- Se la conoce también como RAM (Memoria de Acceso Aleatorio)

Arquitectura de Von Neumann: Memoria Principal

- Se la conoce también como RAM (Memoria de Acceso Aleatorio)
- Se utiliza para almacenar temporalmente datos y programas.

Arquitectura de Von Neumann: Memoria Principal

- Se la conoce también como RAM (Memoria de Acceso Aleatorio)
- Se utiliza para almacenar temporalmente datos y programas.
- Es volátil: Pierde su contenido al desconectar la energía eléctrica

Arquitectura de Von Neumann: Memoria Principal

- Se la conoce también como RAM (Memoria de 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!)

Arquitectura de Von Neumann: Memoria Principal

Memoria de Ejemplo

0	0101
1	1010
2	0000
3	1111
4	1100

Arquitectura de Von Neumann: Memoria Principal

Memoria de Ejemplo

0	0101
1	1010
2	0000
3	1111
4	1100

- Direcciones: $\{0, 1, 2, 3, 4\}$
- Celdas de 4 bits

Arquitectura de Von Neumann: Memoria Principal

Memoria de Ejemplo

0	0101
1	1010
2	0000
3	1111
4	1100

- Direcciones: $\{0, 1, 2, 3, 4\}$
- Celdas de 4 bits

Si la memoria recibe una orden de lectura sobre la celda 3 ¿Que responde?

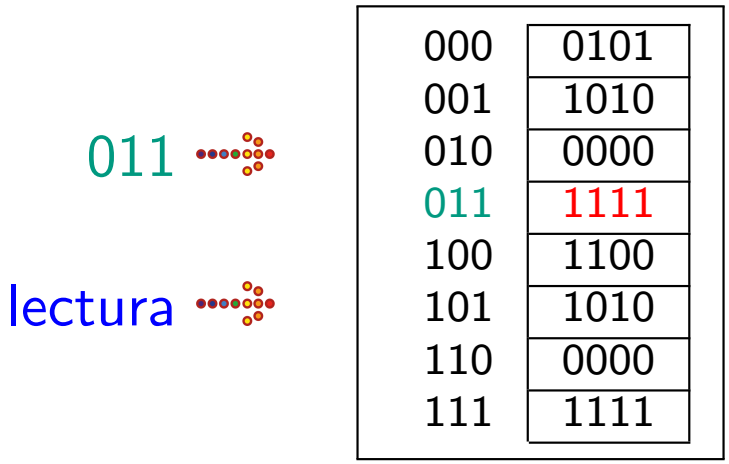
Arquitectura de Von Neumann: Memoria

Funcionamiento: Lectura

1. Recibe la señal de lectura
2. Recibe una dirección
3. Entrega el dato contenido en la celda correspondiente.

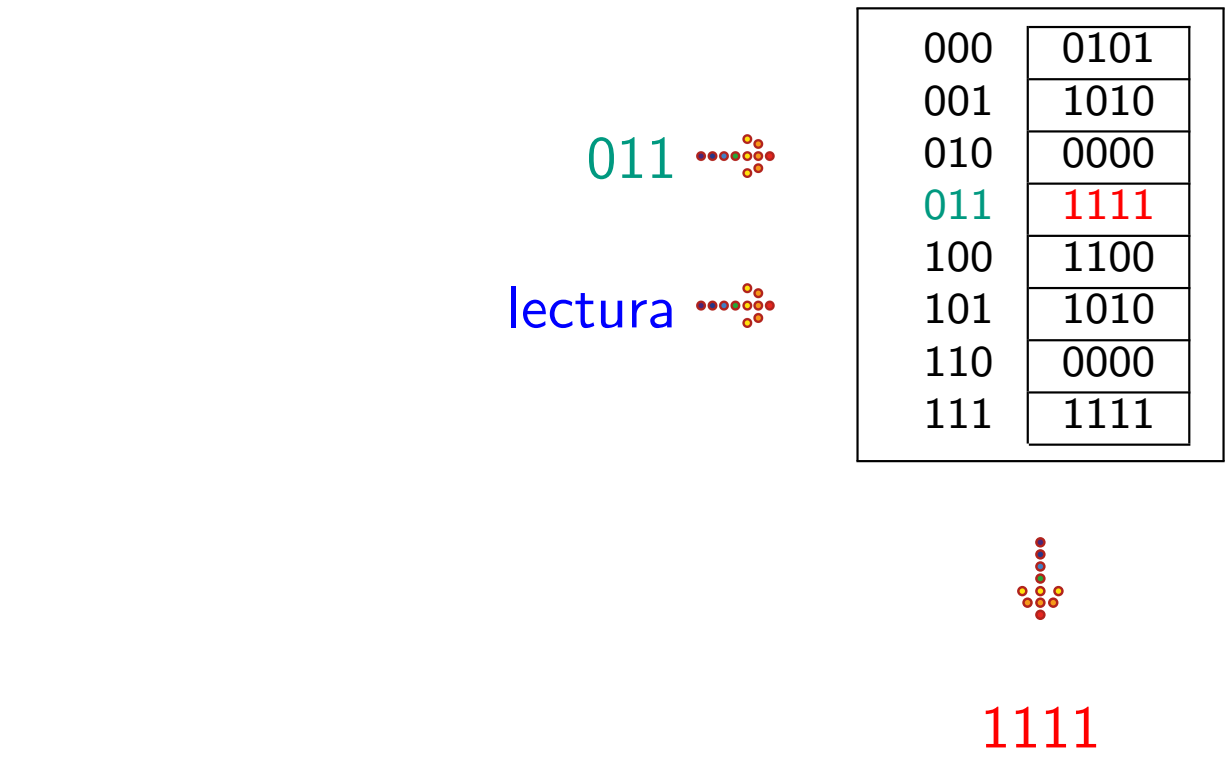
Arquitectura de Von Neumann: Memoria

Funcionamiento: Lectura



Arquitectura de Von Neumann: Memoria

Funcionamiento: Lectura



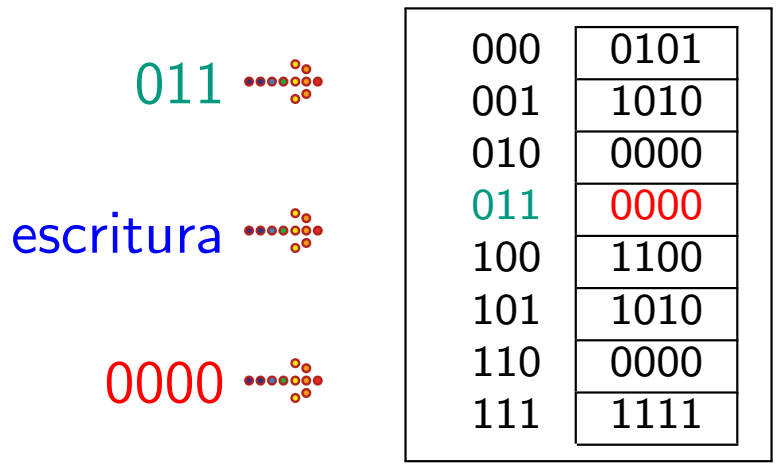
Arquitectura de Von Neumann: Memoria

Funcionamiento: Escritura

1. Recibe la señal de escritura
2. Recibe una dirección
3. Recibe un dato
4. Almacena el dato en la celda correspondiente.

Arquitectura de Von Neumann: Memoria

Funcionamiento: Escritura



Arquitectura de Von Neumann: Memoria

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.

Arquitectura de Von Neumann: Memoria

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

Buses
Interconexión entre la memoria y la CPU

Arquitectura de Von Neumann: Interconexión

Bus

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

Arquitectura de Von Neumann: Interconexión

¿Que se necesita?

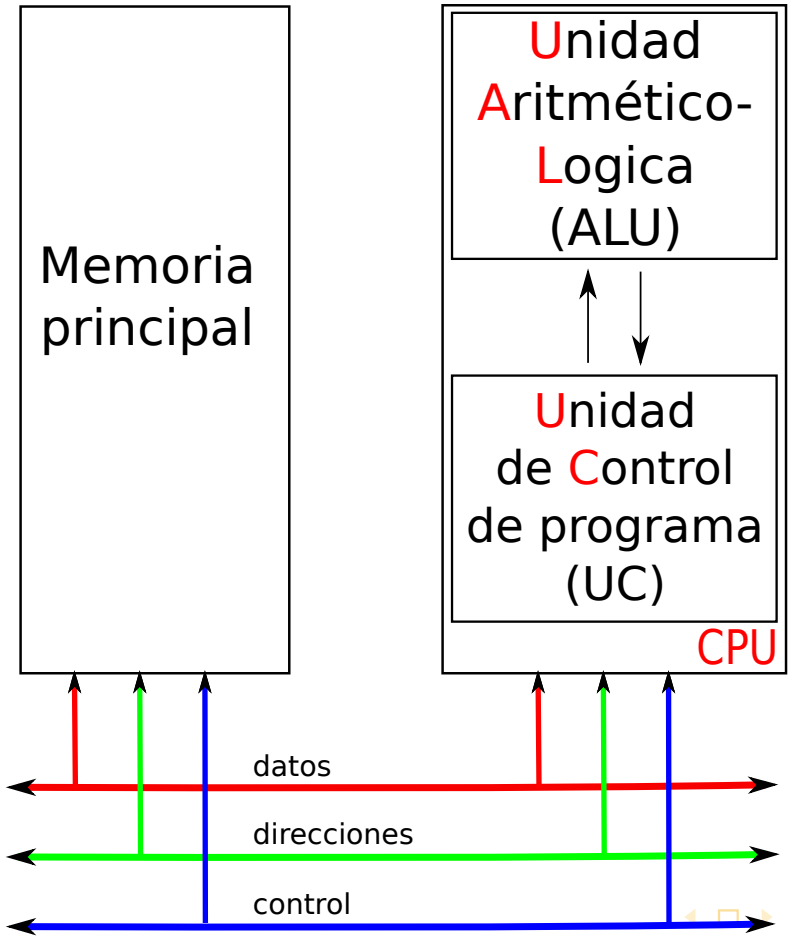
- Transmitir datos desde y hacia la memoria principal
- Transmitir direcciones hacia la memoria principal
- Transmitir señales de control hacia la memoria principal

Arquitectura de Von Neumann: Interconexión

- Cada línea de un bus transmite 1 bit a la vez.
- Ancho del bus: cantidad de líneas

Arquitectura de Von Neumann: Interconexión

- Cada línea de un bus transmite 1 bit a la vez.
- Ancho del bus: cantidad de líneas



Arquitectura de Von Neumann: Interconexión

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

Arquitectura de Von Neumann: Bus de direcciones

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**.

Arquitectura de Von Neumann: Memoria

Revisamos la lectura

(datos)	1111	
(dirección)	011	
(control)	lectura	

000	0101
001	1010
010	0000
011	1111
100	1100
101	1010
110	0000
111	1111

Arquitectura de Von Neumann: Memoria

Revisamos la escritura

(datos)	0000	
(dirección)	011	
(control)	escritura	

000	0101
001	1010
010	0000
011	0000
100	1100
101	1010
110	0000
111	1111

Interconexión: medios compartidos

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

Interconexión: medios compartidos

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

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

Interconexión: medios compartidos

Muchos dispositivos se conectan al bus y la señal transmitida por cualquiera de ellos está disponible para ser leida 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 de Von Neumann: Interconexión

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)

Arquitectura de Von Neumann: Interconexión

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
  ¿El bus está ocupado?/Necesito usar el bus
- Los comandos indican el tipo de operación que debe llevar a cabo la memoria (lectura o escritura)

Arquitectura de Von Neumann: Interconexión

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)
  Lectura/Escritura

Relación entre los buses y memoria principal

0	01010000
1	10010000
2	10101010
3	11001100
4	01010000
5	10010000
6	10101010
7	11001100

Relación entre los buses y memoria principal

0	01010000
1	10010000
2	10101010
3	11001100
4	01010000
5	10010000
6	10101010
7	11001100

- Se tienen 8 celdas

Relación entre los buses y memoria principal

0	01010000
1	10010000
2	10101010
3	11001100
4	01010000
5	10010000
6	10101010
7	11001100

- Se tienen 8 celdas → El bus de direcciones debe tener 3 bits

Relación entre los buses y memoria principal

0	01010000
1	10010000
2	10101010
3	11001100
4	01010000
5	10010000
6	10101010
7	11001100

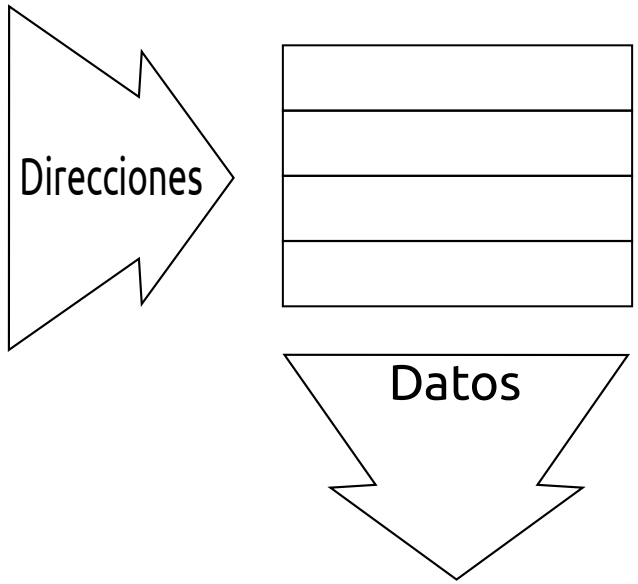
- Se tienen 8 celdas  El bus de direcciones debe tener 3 bits
- Las celdas contienen 8 bits 

Relación entre los buses y memoria principal

0	01010000
1	10010000
2	10101010
3	11001100
4	01010000
5	10010000
6	10101010
7	11001100

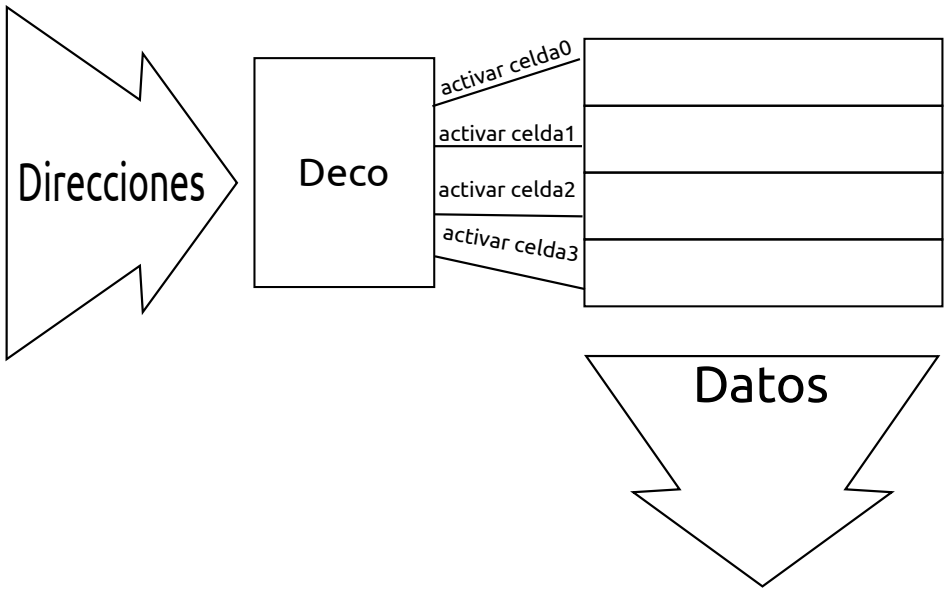
- Se tienen 8 celdas ⇢ El bus de direcciones debe tener 3 bits
- Las celdas contienen 8 bits ⇢ El bus de datos debe tener 8 bits

Acceso a las celdas



¿Cómo se activa una sola celda?

Acceso a las celdas



Usando un circuito decodificador

Actividad Grupal

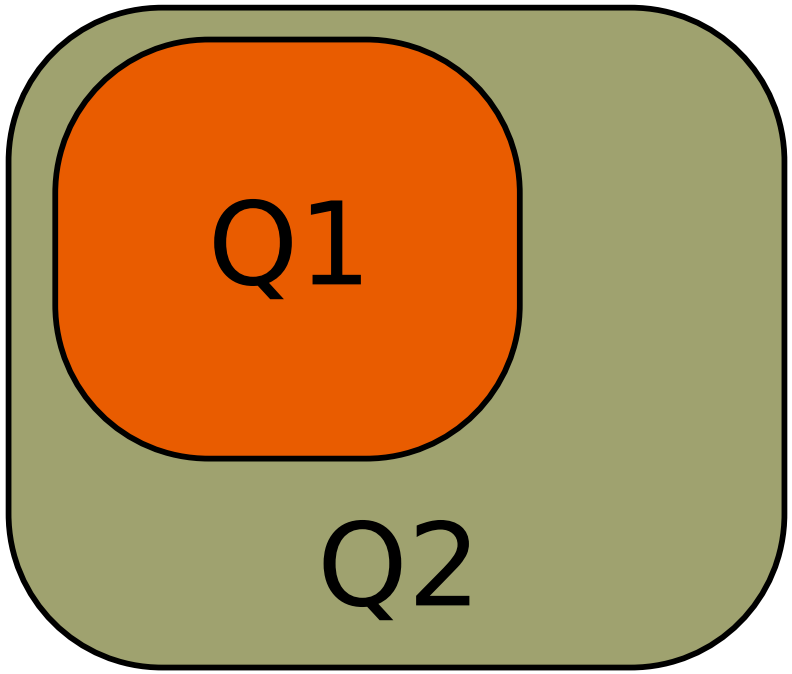
A pensar!

- Suponga que tiene un bus de direcciones de 4 bits con tamaño de celda de 8 bits . ¿Cuántas celdas tiene la memoria ? ¿Qué capacidad total posee dicha memoria?
- Sabiendo que 1 byte equivale a 8 bits, suponga una memoria principal con tamaño de celda de 2 bytes y tamaño total de 64 bytes. ¿Cuántas celdas tiene? ¿Cuántos bits tendrá el bus de direcciones?

De los creadores de Q1 llega...

Arquitectura **Q2**

Arquitectura **Q2**



Arquitectura **Q2**

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

Arquitectura **Q2**

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

Arquitectura **Q2**

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

instrucción	sintaxis	efecto
ADD	`ADD destino, origen`	`destino ← destino+origen`
SUB	`SUB destino, origen`	`destino ← destino - origen`
MUL	`MUL destino, origen`	`(R7,destino) ← destino * origen`
DIV	`DIV destino, origen`	`destino ← destino% origen`
MOV	`MOV destino, origen`	`destino ← origen`

Arquitectura Q2

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

instrucción	sintaxis	efecto
ADD	`ADD destino, origen`	`destino ← destino+origen`
SUB	`SUB destino, origen`	`destino ← destino - origen`
MUL	`MUL destino, origen`	`(R7,destino) ← destino * origen`
DIV	`DIV destino, origen`	`destino ← destino% origen`
MOV	`MOV destino, origen`	`destino ← origen`

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

Arquitectura **Q2**: 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

Arquitectura **Q2**: modos de direccionamiento

Modo de direccionamiento directo

MOV R0, [3456]

ADD [FFEE], [AB01]

ADD [FFEE], 0xAB01

Arquitectura **Q2**: formato de instrucciones

Formato de instrucción
Define la organización de los bits dentro de una instrucción, en términos de las partes que la componen. Debe incluir el **código de la operación** y los **operandos**

Arquitectura **Q2**: formato de instrucciones

Formato de instrucción
Define la organización de los bits dentro de una instrucción, en términos de las partes que la componen. Debe incluir el **código de la operación** y los **operandos**

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

(Idem **Q1**)

Arquitectura **Q2**: Códigos de Operación

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

(Idem **Q1**)

Arquitectura **Q2**: Códigos de los modos de direccionamiento

Modo	**Codificación**
Inmediato	000000
Directo	001000
Registro	100rrr

donde rrr es una codificación (en 3 bits) del número de registro.

Arquitectura **Q2**: formato de instrucciones

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

Los campos de los opendos Destino y Origen...

- no existen (si el modo respectivo es *registro*),
- contienen valores constantes (si el modo respectivo es *inmediato*),
- o contiene una dirección de memoria (si el modo respectivo es *directo*)

Arquitectura **Q2**: Ejemplos

Ejercicio: Ensamblar MOV R1,[0003]

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

Arquitectura **Q2**: Ejemplos

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

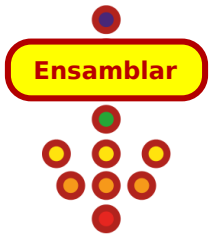
Modo	**Codificación**
Inmediato	000000
Directo	001000
Registro	100rrr

Arquitectura **Q2**: Ejemplos

Ejercicio: Ensamblar MOV R1,[0003]

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

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



0001100001001000000000000000011

Arquitectura **Q2**: Ejemplos

Ejercicio: Ensamblar MOV [AAAA],[0003]

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

Arquitectura **Q2**: Ejemplos

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

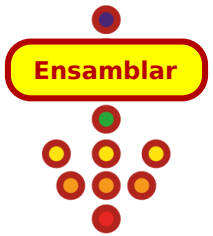
Modo	**Codificación**
Inmediato	000000
Directo	001000
Registro	100rrr

Arquitectura **Q2**: Ejemplos

Ejercicio: Ensamblar MOV [AAAA],[0003]

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

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



0001001000001000101010101010101000000000000011

Computadora **Q2**: Ejemplos

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

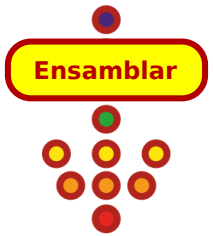
Modo	**Codificación**
Inmediato	000000
Directo	001000
Registro	100rrr

Computadora **Q2**: Ejemplos

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



0001001000100101111100001111000 0

Computadora **Q2**: Ejercicios

Ejercicios

Computadora **Q2**: Ejercicios

Ejercicios

1. Hacer un programa que multiplique por 12 el valor de la celda 0007 .
2. Hacer un programa que sume los valores de las celdas 7654 y 0123, y guarde el resultado en R2

Computadora **Q2**

Si las direcciones son de 16 bits

Computadora **Q2**

Si las direcciones son de 16 bits

y las celdas contienen 16 bits

Computadora **Q2**

Si las direcciones son de 16 bits

y las celdas contienen 16 bits

¿Cuál es el tamaño de la memoria de **Q2**?

Computadora **Q2**: Ejercicios

Ejercicio: Desensamblar

1200 A1A0 0002

1200 A1A1 0003

0208 A1A0 A1A1

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

Modo	**Codificación**
Inmediato	000000
Directo	001000
Registro	100rrr

Computadora **Q2**: Ejercicios

Ejercicio: Desensamblar

1200 A1A0 0002

1200 A1A1 0003

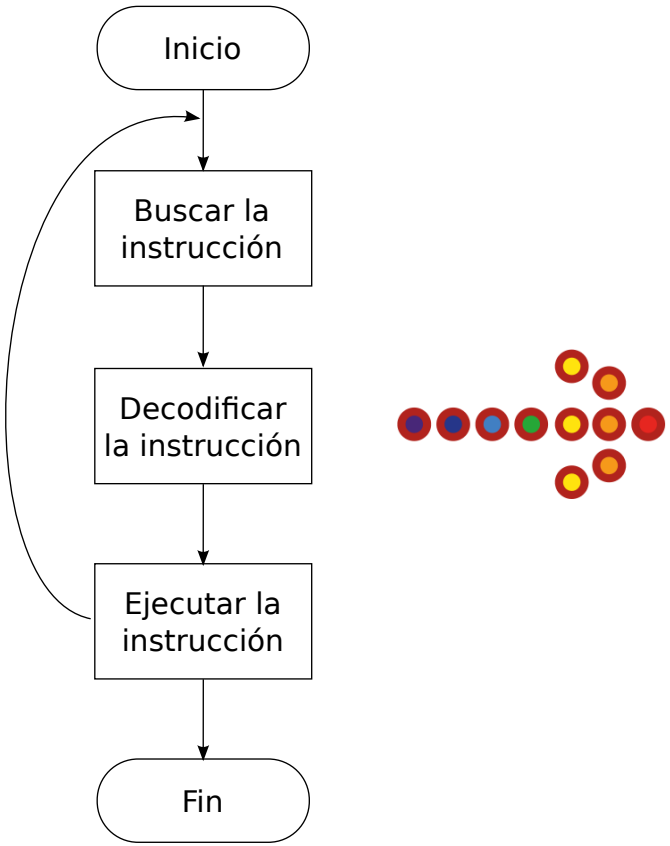
0208 A1A0 A1A1

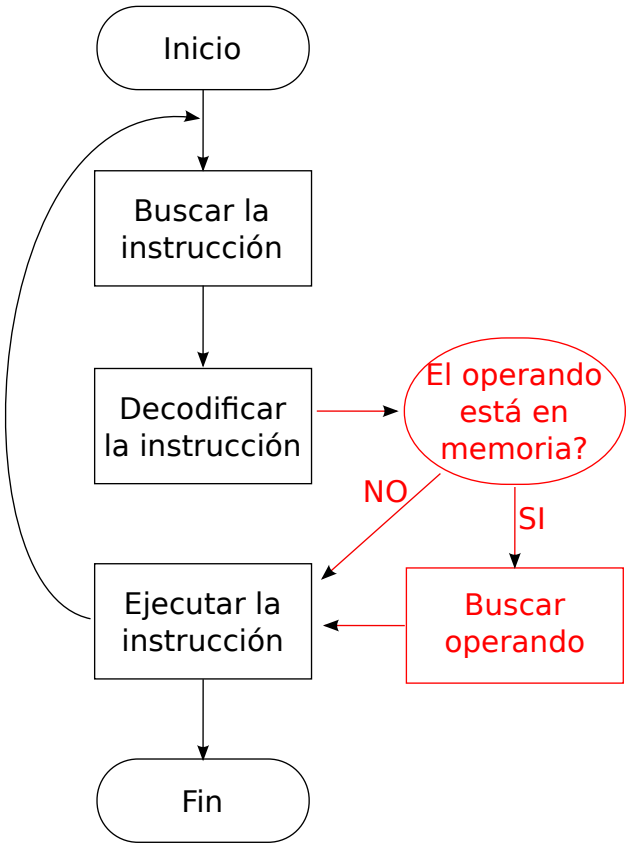
cadena	Codop	Modo Destino	Modo Origen	Origen	Destino
1200 A1A0 0002	MOV	Directo	Inmediato	A1A0	0002
1200 A1A1 0003	MOV	Directo	Inmediato	A1A1	0003
0208 A1A0 A1A1	MUL	Directo	Directo	A1A0	A1A1

Ciclo de ejecución de instrucción

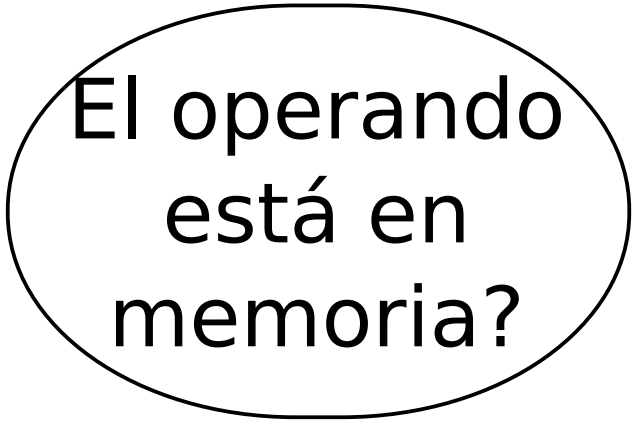
Ciclo de ejecución de instrucción revisado

Revisemos el ciclo para **Q2**





Ciclo de ejecución de instrucción revisado



¿Cómo hace la Unidad de Control para responderlo?

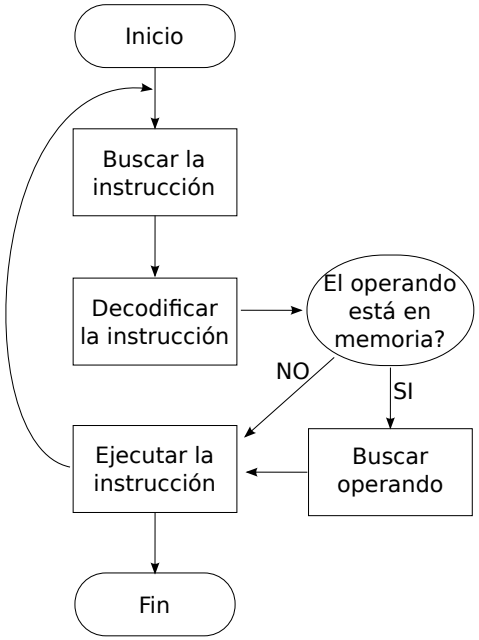
Ciclo de ejecución de instrucción revisado

El operando está en memoria?

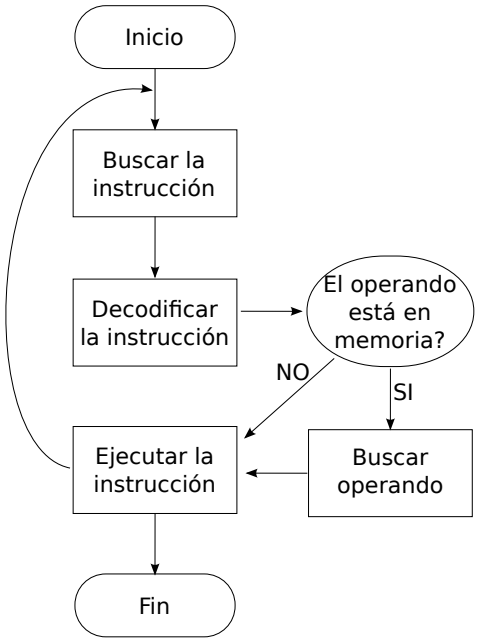
¿Cómo hace la Unidad de Control para responderlo?

Mediante los modos de direccionamiento

¿Cuanto tarda la ejecución de un programa?

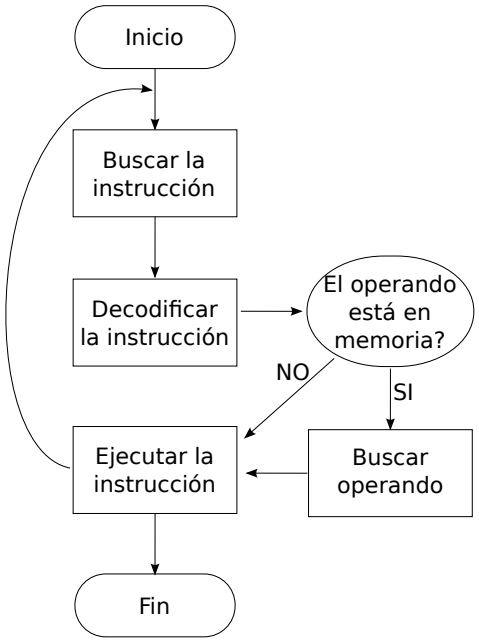


¿Cuánto tarda la ejecución de un programa?



¿Cuánto tarda la ejecución de un programa?

¿De qué depende?



¿Cuánto tarda la ejecución de un programa?

¿De qué depende?

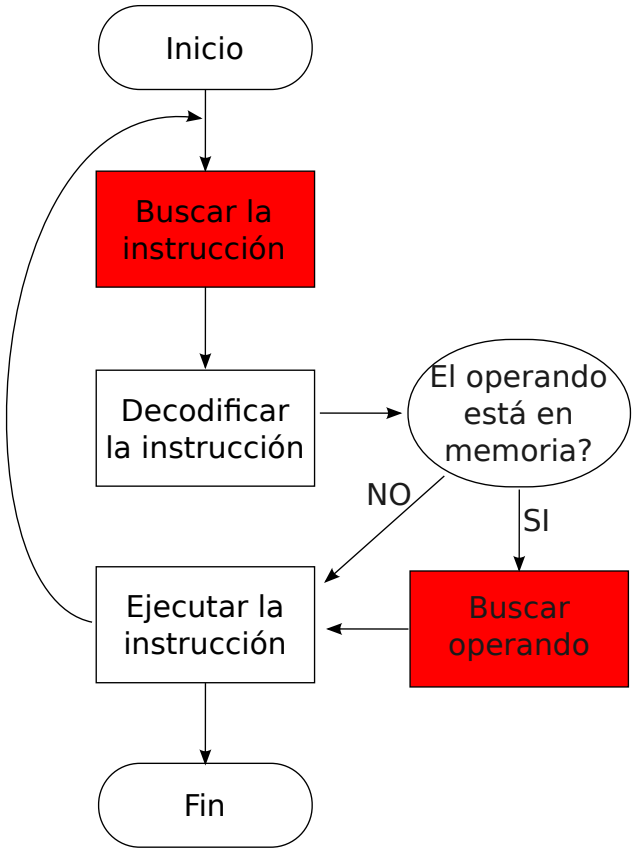
Accesos a memoria + tiempo de ejecución

Accesos a memoria

¿En que momento se accede a memoria?

Accesos a memoria

¿En que momento se accede a memoria?



Cantidad de accesos de una instrucción

¿Cuántos accesos tiene?

MOV R1, R2

Cantidad de accesos de una instrucción

MOV R1, R2

Cantidad de accesos de una instrucción

MOV R1, R2

Búsqueda de la instrucción

Cantidad de accesos de una instrucción

MOV R1, R2

Búsqueda de la instrucción  0001100001100010
(1 celda)

Búsqueda de operandos

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]

Cantidad de accesos de una instrucción

MOV R1, [0001]

Búsqueda de la instrucción

Cantidad de accesos de una instrucción

MOV R1, [0001]

Búsqueda de la instrucción  0001100001001000000000000000

(2 celdas)

Búsqueda de operandos

Cantidad de accesos de una instrucción

MOV R1, [0001]

Búsqueda de la instrucción  0001100001001000000000000000

(2 celdas)

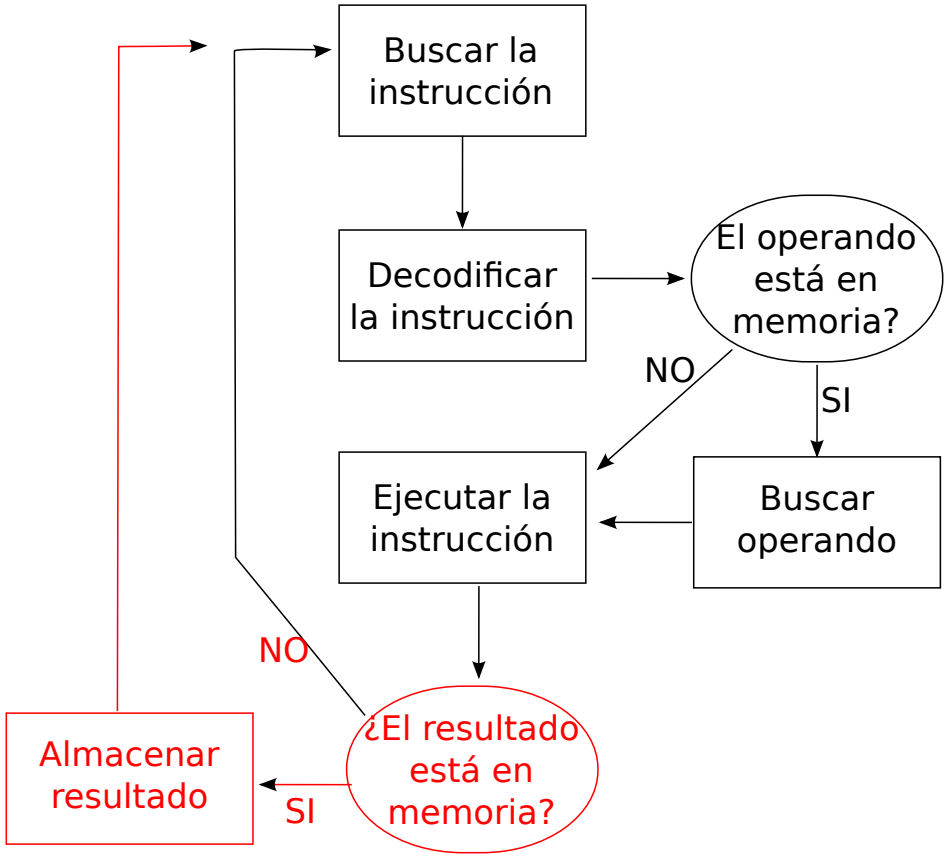
Búsqueda de operandos  Se lee la celda 0001

Cantidad de accesos de una instrucción

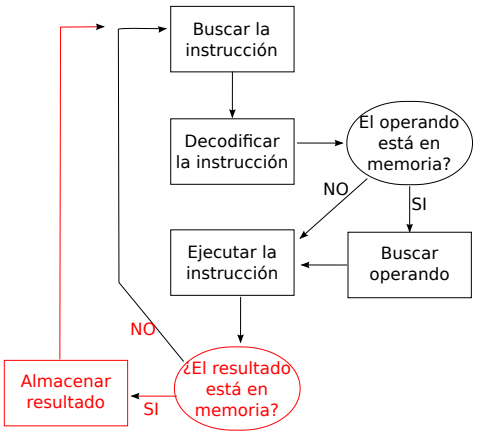
¿Cuantos accesos tiene este programa?

MOV [0001],R1

Cantidad de accesos de una instrucción



Cantidad de accesos de una instrucción



MOV [0001],R1

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