

# El Procesador II:

## Modos de direccionamiento y formato de las instrucciones

# Ejemplo: Cálculo de $Y = (A-B)/(C+D \cdot E)$

## Con instrucciones de tres direcciones

Instrucción	Comentario
SUB Y, A, B	$Y \leftarrow A - B$
MPY T, D, E	$T \leftarrow D \times E$
ADD T, T, C	$T \leftarrow T + C$
DIV Y, Y, T	$Y \leftarrow Y / T$

## Con instrucciones de una dirección

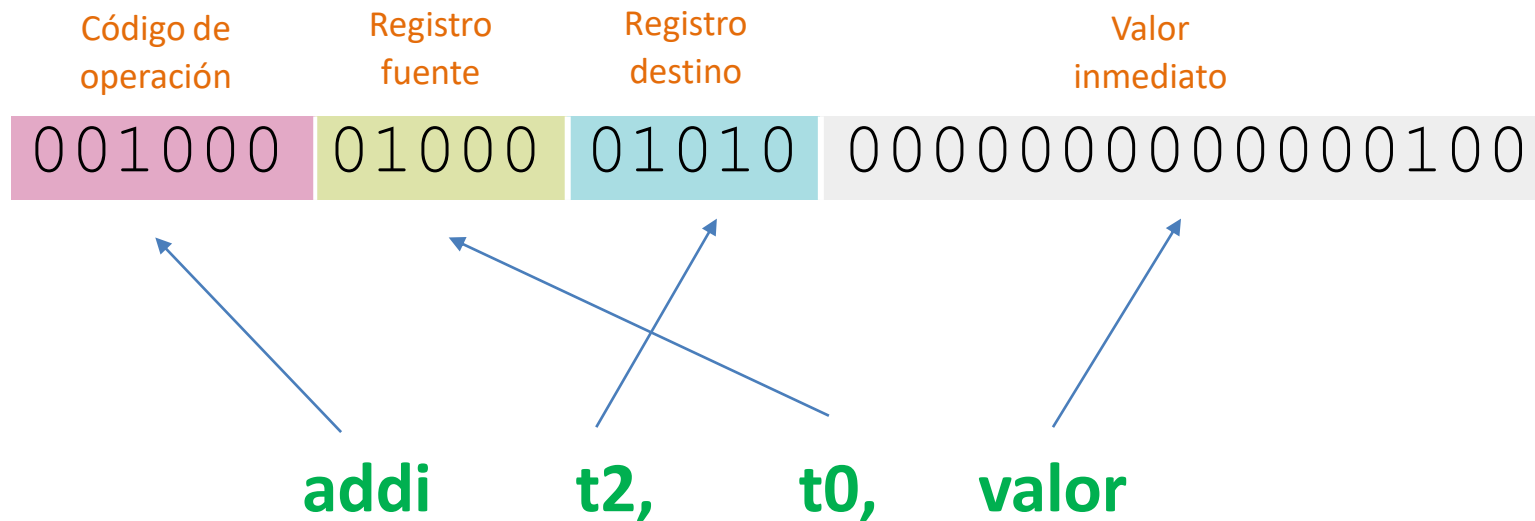
Instrucción	Comentario
LOAD D	$AC \leftarrow D$
MPY E	$AC \leftarrow AC * E$
ADD C	$AC \leftarrow AC + C$
STOR Y	$Y \leftarrow AC$
LOAD A	$AC \leftarrow A$
SUB B	$AC \leftarrow AC - B$
DIV Y	$AC \leftarrow AC / Y$
STOR Y	$Y \leftarrow AC$

## Con instrucciones de dos direcciones

Instrucción	Comentario
MOVE Y, A	$Y \leftarrow A$
SUB Y, B	$Y \leftarrow Y - B$
MOVE T, D	$T \leftarrow D$
MPY T, E	$T \leftarrow T * E$
ADD T, C	$T \leftarrow T + C$
DIV Y, T	$Y \leftarrow Y / T$

## Formato de las instrucciones

## Ejemplo de **instrucción máquina (binario)** y su **equivalente en ensamblador**



- Y se divide en campos:
  - Operación a realizar
  - Operandos a utilizar
  - Puede haber operandos implícitos (indicados por el CO opcode)

# Campos que puede tener una instrucción

- Ejemplos de instrucciones que realizan la operación suma utilizando un número distinto de operandos

ADD_SP	Sin uso		
--------	---------	--	--

Ejemplo de una instrucción con cero operandos

ADDR_SP	7	Sin uso	
---------	---	---------	--

Ejemplo de una instrucción con un operando

ADD	4	2	Sin uso
-----	---	---	---------

Ejemplo de una instrucción con dos operandos

ADDR	4	2	7	Sin uso
------	---	---	---	---------

Ejemplo de una instrucción con tres operandos

# Introducción a los modos de direccionamiento

Supongamos que tenemos los siguientes valores guardados en las siguientes direcciones de memoria principal (MP) y queremos hacer su suma:

a		...
	1	0x10B2
	3	0x10B3
	2	0x10B4
	4	0x10B5
	...	...
	10	$0x10B2 + n - 1$
	12	$0x10B2 + n$

## CARACTERÍSTICAS DEL EJEMPLO

- Máquina de 3 operandos (add reg, dir1, dir2)
- Suponemos que debemos sumar N valores
- En este caso, el array recibirá el nombre de «a» → int a[N]

## PROBLEMAS

- ¿Debemos codificar las N instrucciones por separado para acceder a cada una de las posiciones de memoria?
- ¿Una vez codificado, el programa se puede utilizar para otro array de datos que se encuentre en posiciones de memoria distintas (reubicable)?
- ¿Los datos deben estar colocados a partir de una posición de memoria concreta?
- Si otro array tiene más posiciones de memoria, ¿podemos acceder a ellas con nuestro programa actual?

# Modos de direccionamiento

El modo de direccionamiento es un procedimiento que permite determinar la ubicación (dirección efectiva) de un operando o una instrucción

## Puede ser:

- Inmediato
- Directo por memoria
- Directo por registro
- Indirecto por memoria
- Indirecto por registro
- Relativo
  - A registro base
  - A registro índice
  - A contador de programa

# Modos de direccionamiento

El modo de direccionamiento es un procedimiento que permite determinar la ubicación (dirección efectiva) de un operando o una instrucción

Los **operandos** pueden ubicarse:

- En la propia instrucción
- En los registros del procesador
- En memoria principal
- En unidades de Entrada/Salida (I/O)

# Modos de direccionamiento

**Inmediato:** El operando es parte de la instrucción.

## Instrucción

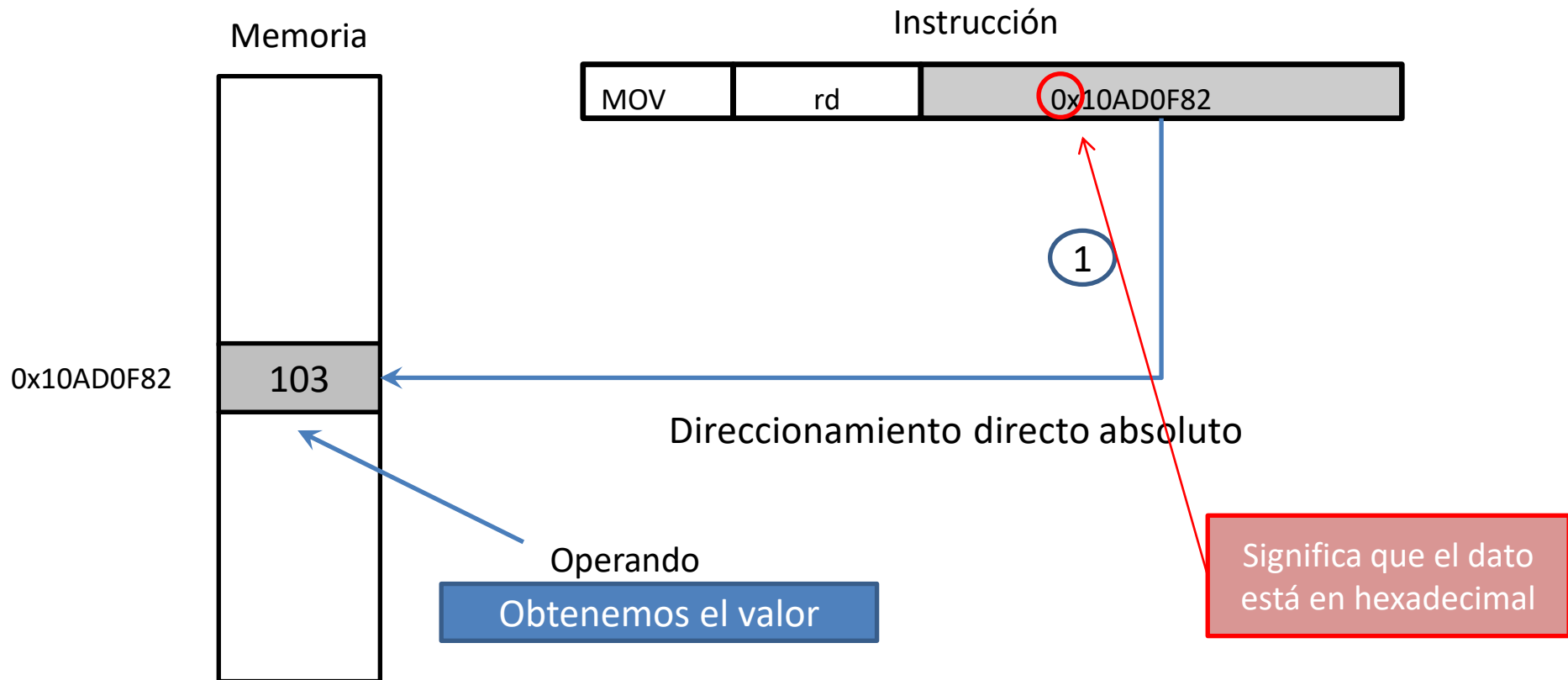
addi	Rd	Rs1	103
------	----	-----	-----

**Operando**



# Modos de direccionamiento

**Directo por memoria:** El operando se encuentra en memoria, y la dirección está codificada en la instrucción



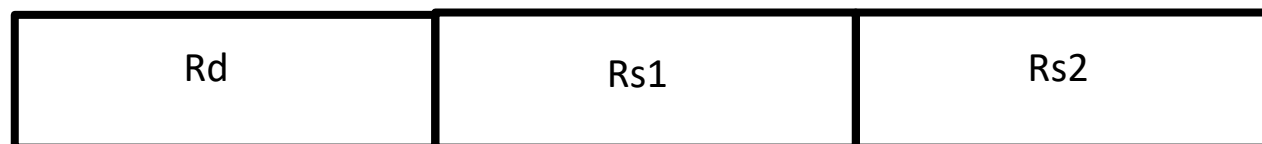
# Modos de direccionamiento

**Directo por registro:** El operando se encuentra en el registro

## Instrucción



## Banco de registros



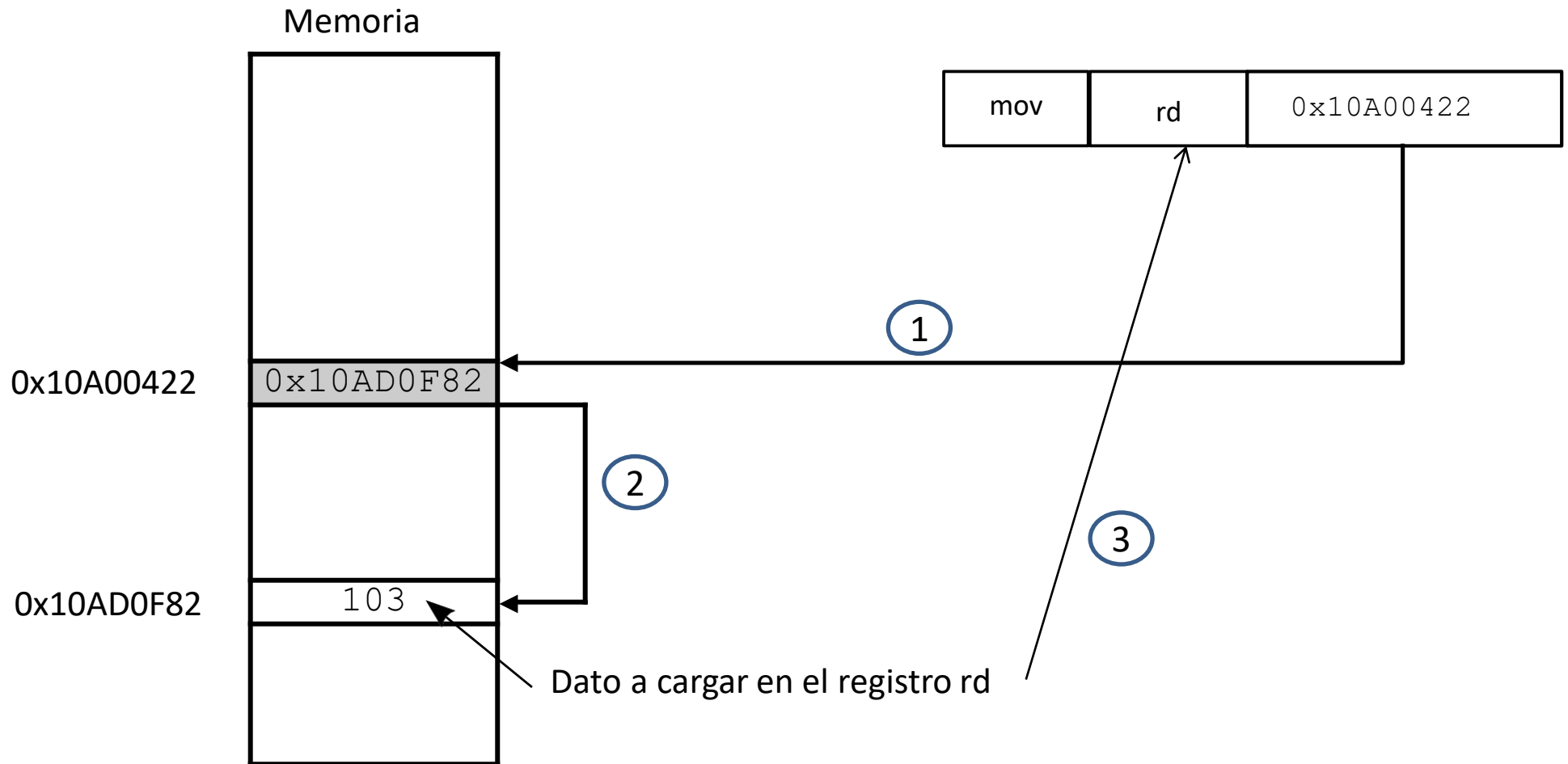
↑  
Escritura

↓  
Lectura

↓  
Lectura

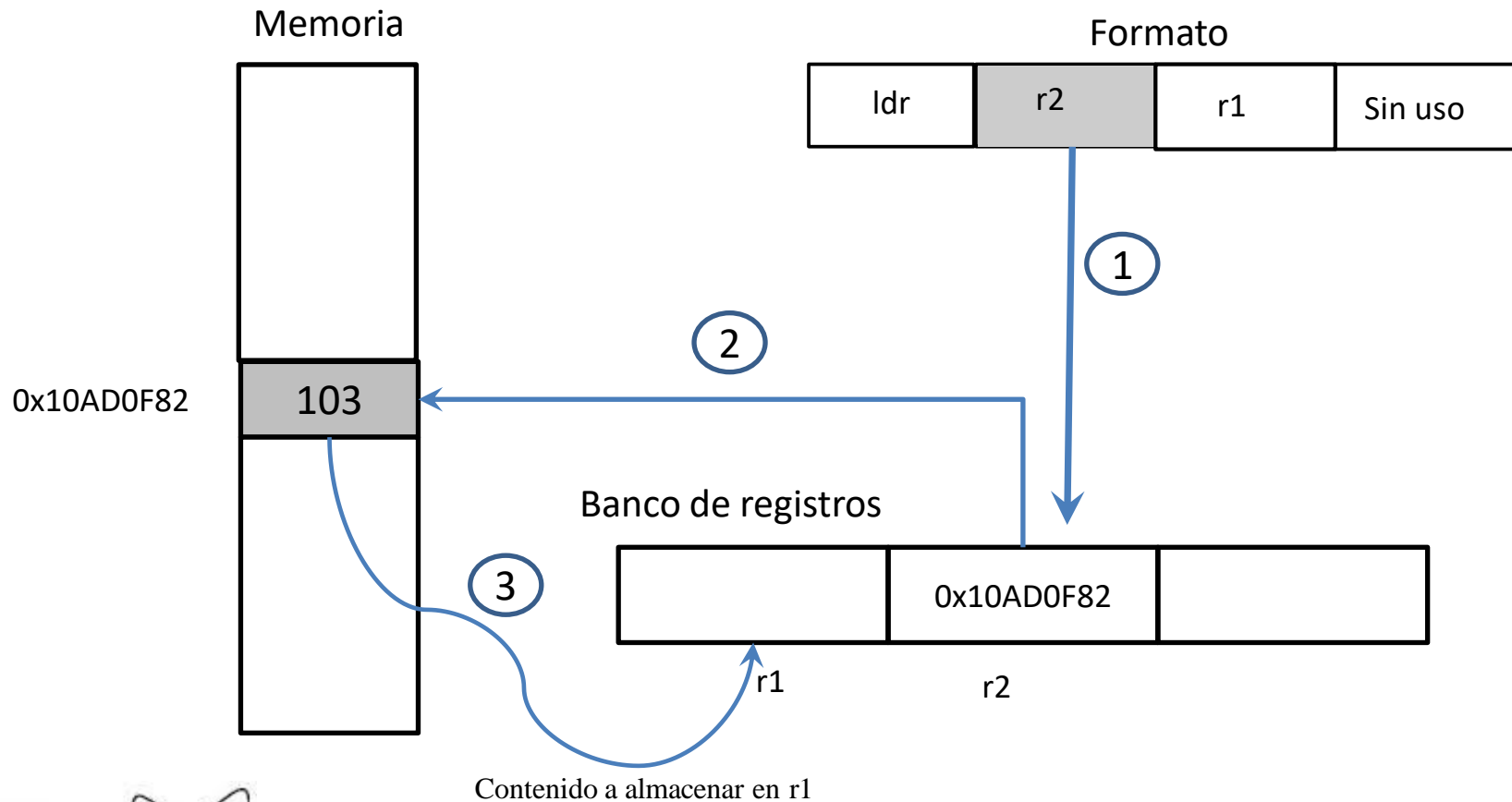
# Modos de direccionamiento

## Indirecto por memoria



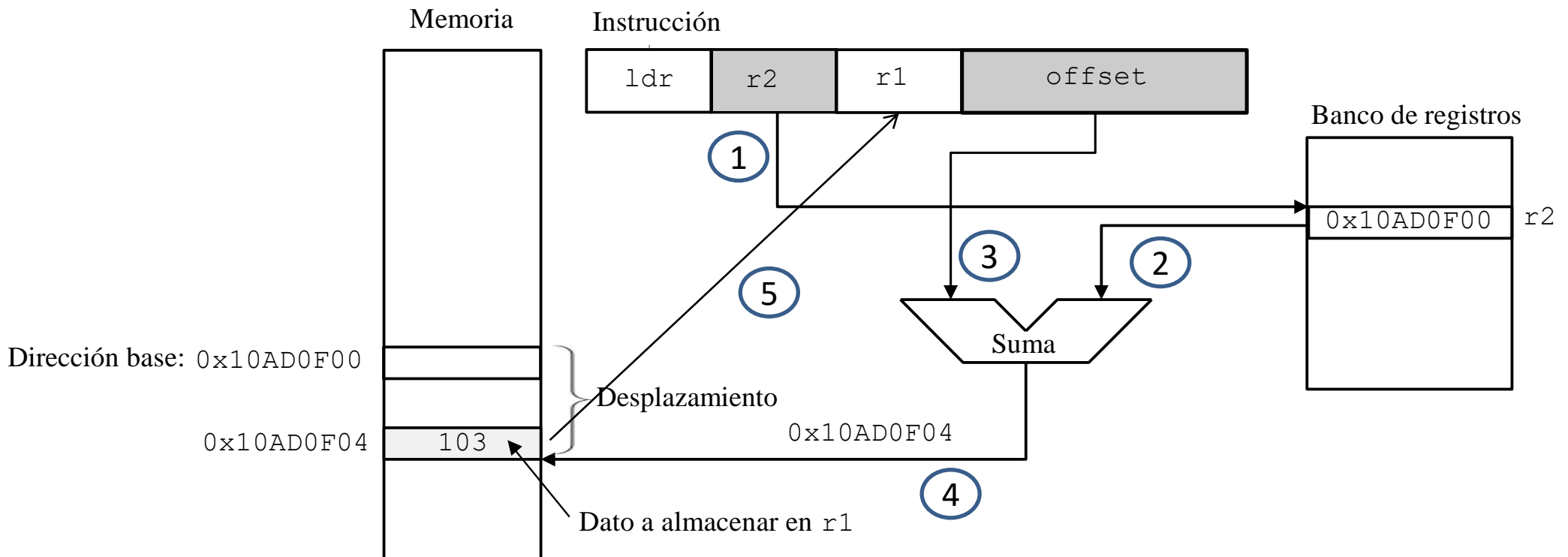
# Modos de direccionamiento

**Indirecto por registro:** Se indica en la instrucción el registro que almacena la dirección del operando



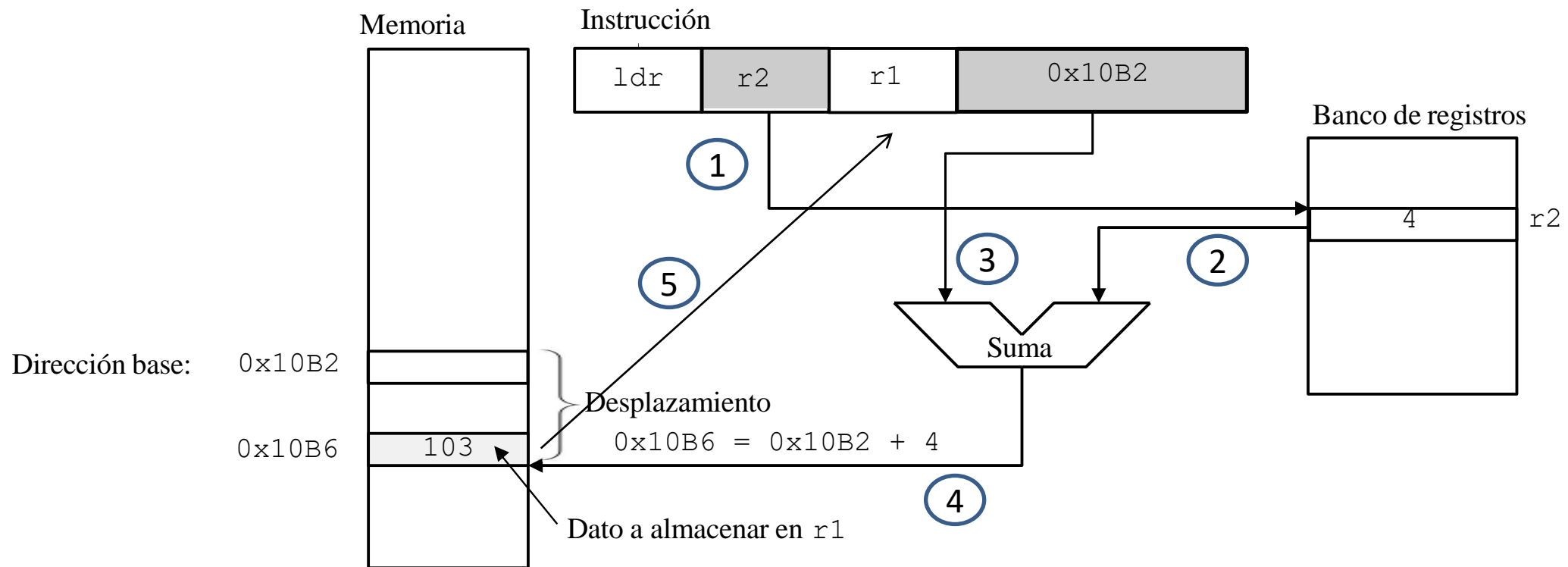
# Modos de direccionamiento

## Direccionamiento relativo a registro base



# Modos de direccionamiento

## Direccionamiento relativo a registro índice



# Modos de direccionamiento

## Direccionamiento relativo a contador de programa

Instrucción



Salta 64 posiciones de memoria hacia delante si r0 = 0

Contador de programa



Operando



0x10AD0F64



# Aplicación de sumar con los distintos modos de direccionamiento un array de N posiciones

Ejemplos			
Modo de direccionamiento directo por registro	Modo de direccionamiento indirecto	Modo de direccionamiento relativo a registro índice	Modo de direccionamiento relativo a registro base
<p><b>add Rd, Rs1, Rs2</b></p> <p>Se debe ir cargando cada uno de los valores del array en los registros fuente para después poder sumarlos. En este caso no es práctico</p>	<p><b>ldr r1,dir</b></p> <p>Ldr R0, 0x10B2  Ldr R1, 0x10B3  Ldr R2, 0x10B4  Ldr R3, 0x10B5  Ldr R4, 0x10B6  Ldr R5, 0x10B7  ...  Add R0, R0, R1  Add R0,R0, R2  ...</p>	<p><b>ldr r1, 0x10B2(r2)</b></p> <p>Mv R2, #0  Mv R0, #0</p> <p>Bucle:  Ldr R1, 0x10B2(R2)  Addi R2, R2, #1  Add R0, R0, R1  B bucle</p> <p>R2 → registro para incrementar el índice  R0 → registro para almacenar el resultado</p>	<p><b>ldr r1, [r2, #offset]</b></p> <p>Mv R2, 0x10B2  Mv R0, #0  Ldr R1, [R2, #0]  Add R0, R0, R1</p> <p>Bucle:  Ldr R1, [R2, #1]  Add R0, R0, R1  Addi R2, R2, #1  B bucle</p>



# Formato de las instrucciones

El formato de una instrucción indica los campos y su tamaño:

- Tamaño de un campo limita los valores que puede codificar
- La longitud del formato viene determinada por los siguientes factores:
  - a) El número de operandos de la instrucción
  - b) El modo de direccionamiento utilizado
  - c) El número de registros existentes y el espacio direccionado