MODOS DE DIRECCIONAMIENTO

Principios de Computadores

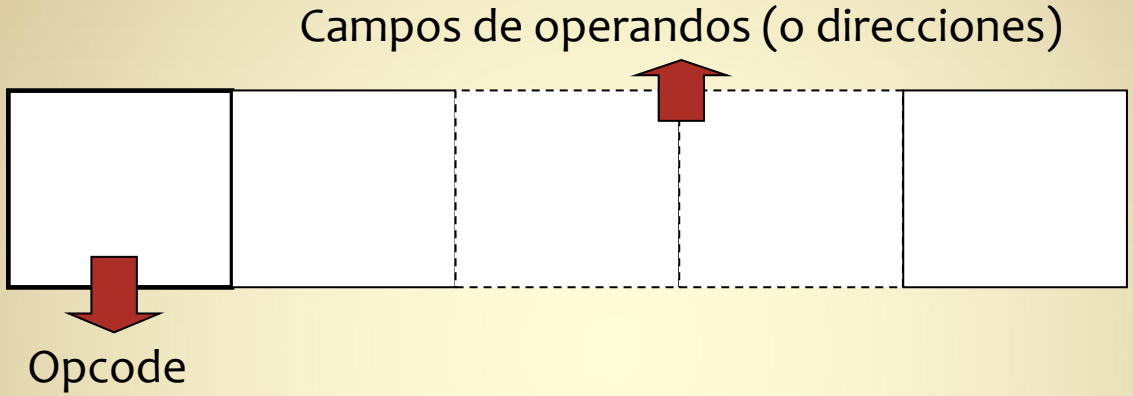
- ¿Qué es el direccionamiento?
- Modos de direccionamiento

1. Direccionamiento inmediato
2. Direccionamiento por registros
3. Direccionamiento directo
4. Direccionamiento indirecto
5. Direccionamiento relativo a registro base
6. Direccionamiento relativo a contador de programa
7. Direccionamiento indexado
8. Direccionamiento implícito

- Comparación de modos de direccionamiento

¿Qué es el direccionamiento?

- Formato de una instrucción:



- El OPCODE indica el tipo de operación que debe hacerse sobre los operandos. Los demás campos de la instrucción indican donde están esos operandos.

- Los **modos de direccionamiento** son las diferentes maneras de especificar un operando dentro de una instrucción.

Modos de direccionamiento

- Las instrucciones pueden clasificarse según el número de direcciones que usan. Las más comunes son de 1, 2 y 3 direcciones.
- En muchas máquinas se hacen operaciones aritméticas con una sola dirección (donde está un operando). El otro se encuentra en un registro especial de la ALU que se llama ACUMULADOR:

acumulador := acumulador + memoria [m]

- Las instrucciones de dos direcciones usan una de las direcciones como fuente y la otra como destino:

destino:= destino + fuente

- Las instrucciones de tres direcciones especifican dos fuentes y un destino. Ej: las dos fuentes se suman y el resultado se guarda en el destino.

Modos de direccionamiento

Los bits del campo de dirección indican dónde encontrar los operandos. Esto es el **DIRECCIONAMIENTO**, y se puede hacer de distintos modos:

1) Direccionamiento inmediato.

2) Direccionamiento por registros.

3) Direccionamiento directo.

4) Direccionamiento indirecto.

5) Direccionamiento relativo a registro base.

6) Direccionamiento relativo a PC.

7) Direccionamiento indexado.

8) Direccionamiento implícito.

1. Direccionamiento inmediato

- El modo más simple de especificar un operando es que la parte de la dirección de la instrucción contenga ya el operando y no su dirección. Este operando se llama **OPERANDO INMEDIATO**.

- Ventaja: no requiere referencias adicionales a memoria para extraer operandos. Es muy rápido. Desventaja: el operando debe ser un número que quepa en el campo de dirección.

- Por ejemplo, en MIPS:

```
addi $sp, $sp, 12

j 1000
```

2. Direccionamiento por registros

- El campo de dirección contiene el número del registro donde está almacenado el operando.

- Las máquinas con registros se utilizan por dos motivos:

1) Los registros son más rápidos que la memoria principal.

2) El número de bits necesarios para especificar un registro es mucho más pequeño que el necesario para especificar una dirección de memoria. Esto implica instrucciones más cortas.

- Pero como hay pocos registros (8-16-32…) la programación es complicada: se debe decidir donde poner los operandos y los resultados intermedios de las operaciones, en el limitado número de registros o en la memoria principal.

- Por ejemplo, en MIPS:

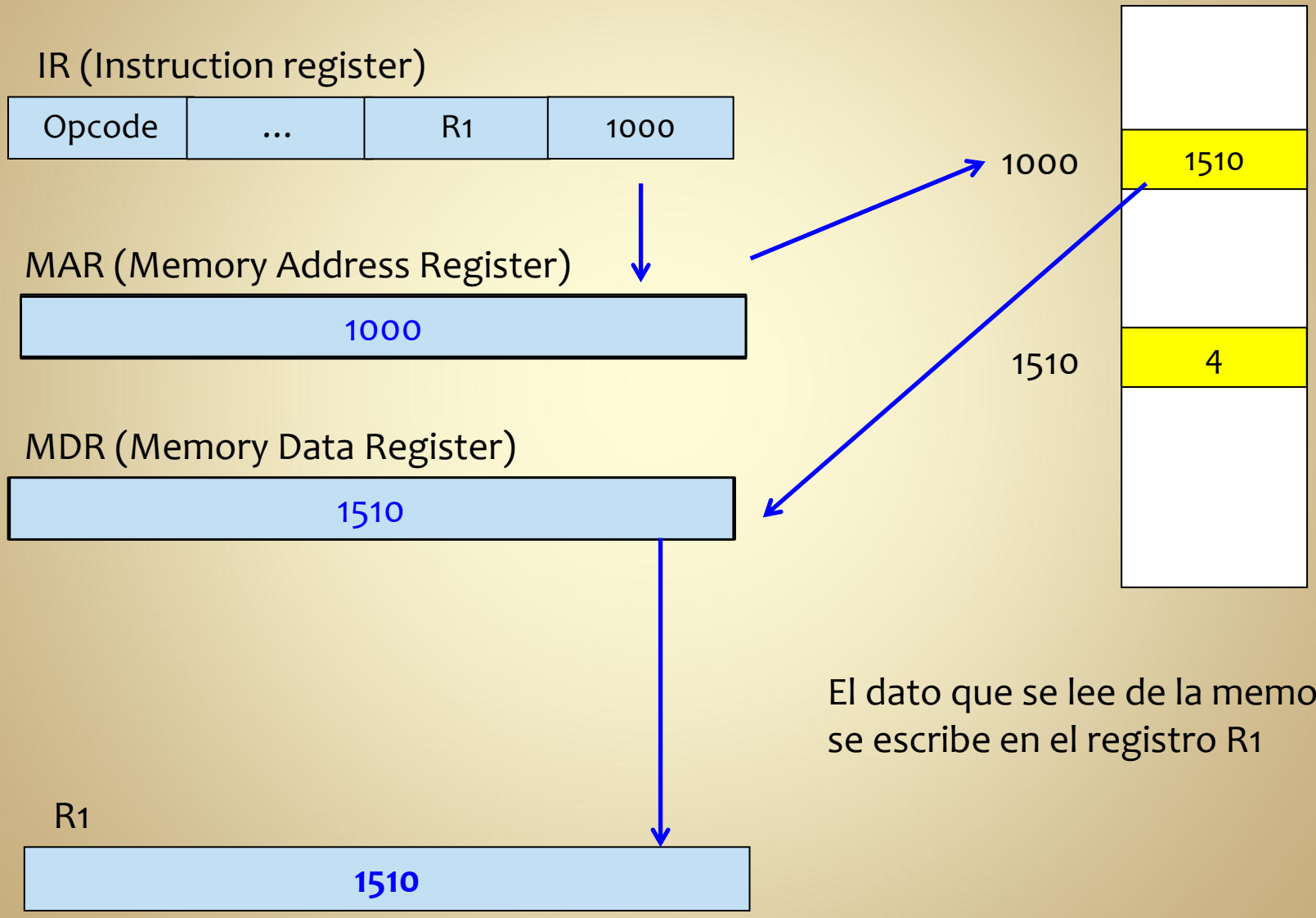
```
add $t0, $s0, $s1
```

3. Direccionamiento directo

El campo de dirección contiene la dirección de memoria donde está guardado el operando. Implica **un acceso a memoria.**

Ej: Instrucción: “Cargar R1 de 1000” (direccionamiento directo)

Direccionamiento directo



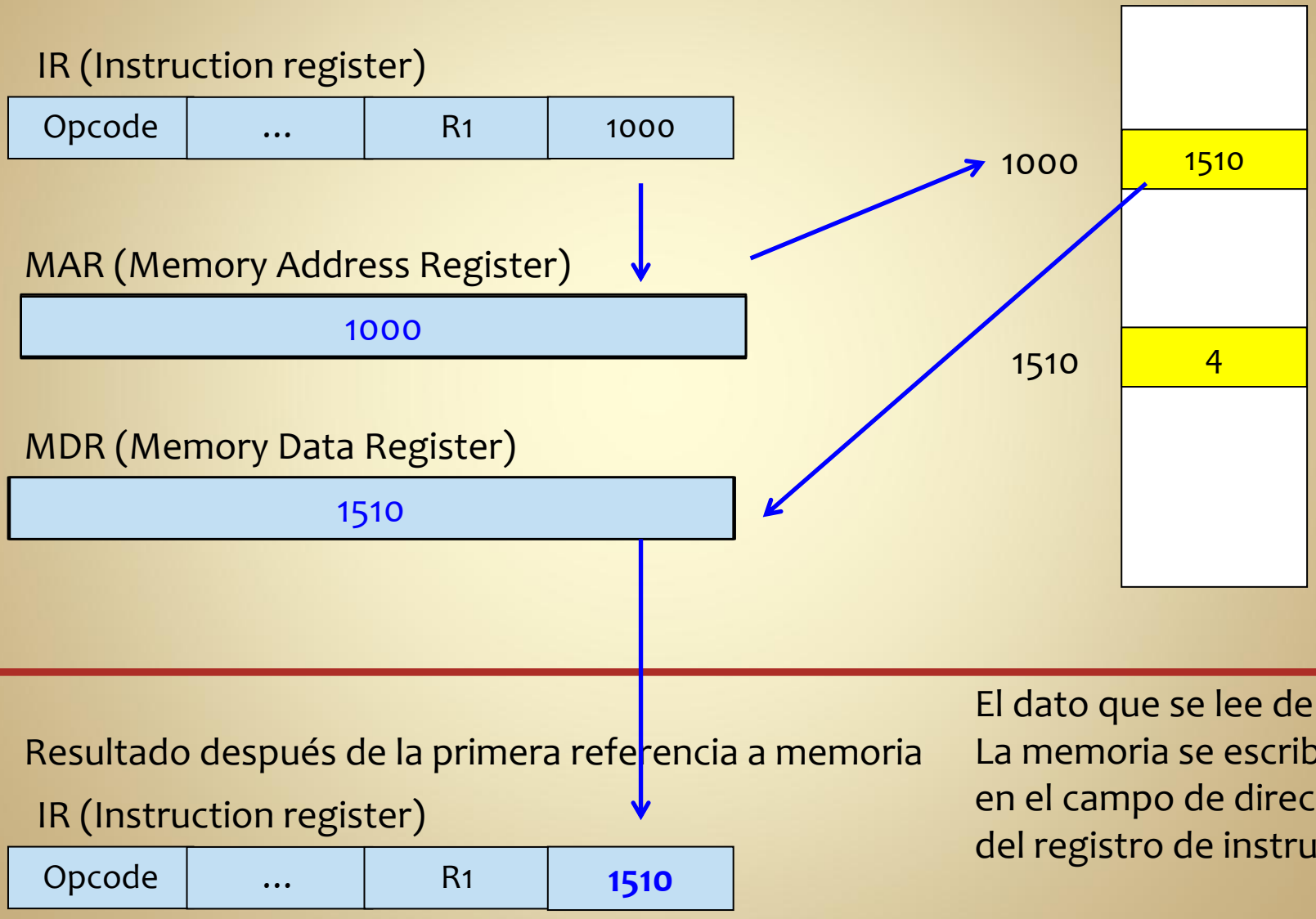
El dato que se lee de la memoria se escribe en el registro R1

4. Direccionamiento indirecto

El campo de dirección contiene la dirección de memoria del sitio donde está guardada la dirección del operando (no el operando en sí). Implica **dos accesos a memoria**.

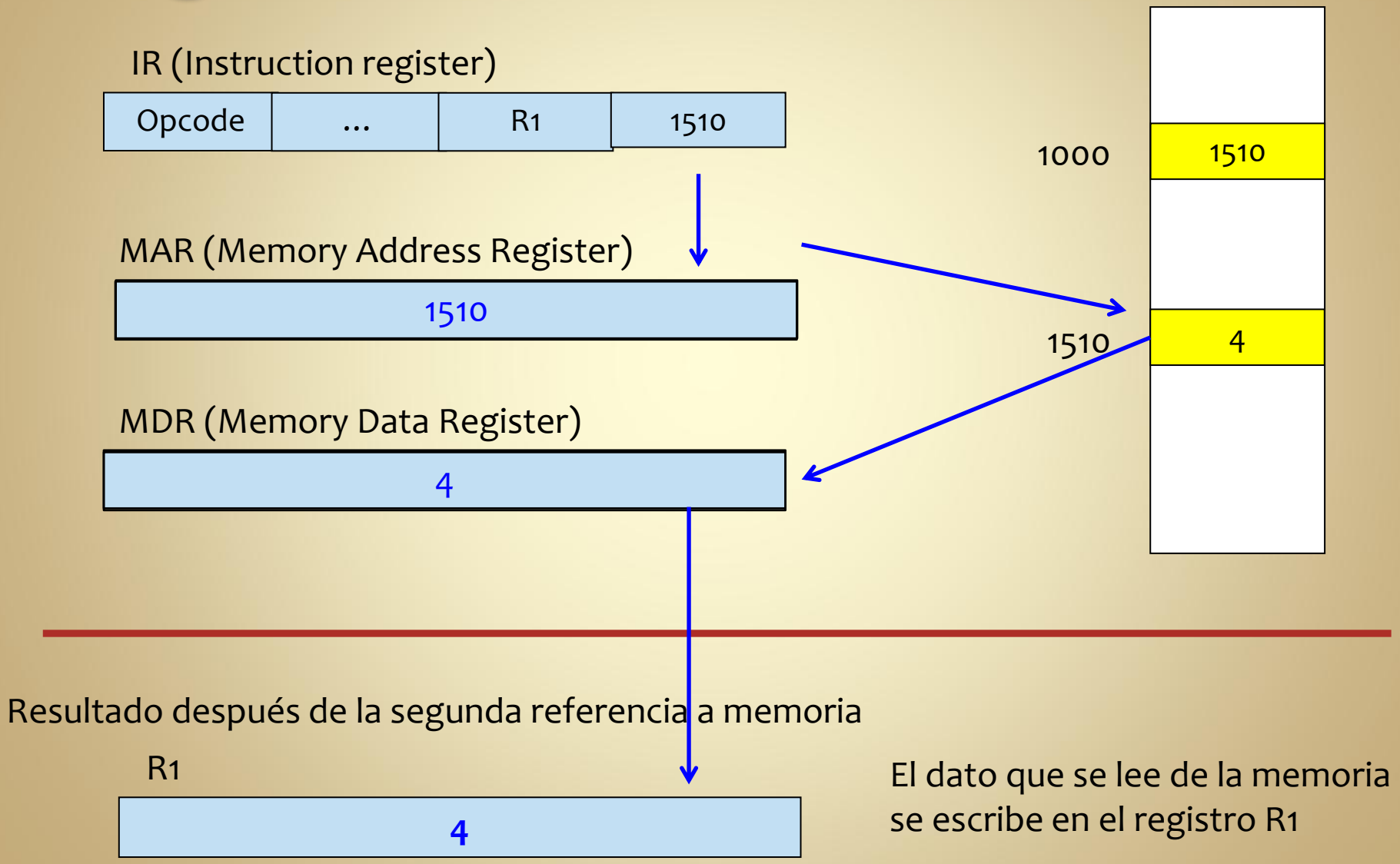
Ej: Instrucción: “Cargar R1 de 1000” (direccionamiento indirecto)

Primera referencia a memoria



El dato que se lee de
La memoria se escribe
en el campo de dirección
del registro de instrucción

Segunda referencia a memoria



Direccionamiento indirecto multinivel

- El direccionamiento **inmediato** no necesita referencias a memoria ni a registros.

- El direccionamiento **directo** necesita una referencia a memoria.

- El direccionamiento **indirecto** necesita dos referencias a memoria (una para el puntero, otra para el operando).

- El direccionamiento **INDIRECTO MULTINIVEL** exige al menos 3 referencias a memoria (dos o más para punteroes y una para el operando). Es un modo de direccionamiento que casi no se usa.

5. Direccionamiento relativo a registro base

- La dirección efectiva del operando se calcula como la suma del contenido de un registro (**registro base**) y un **desplazamiento** (contenido en la instrucción).

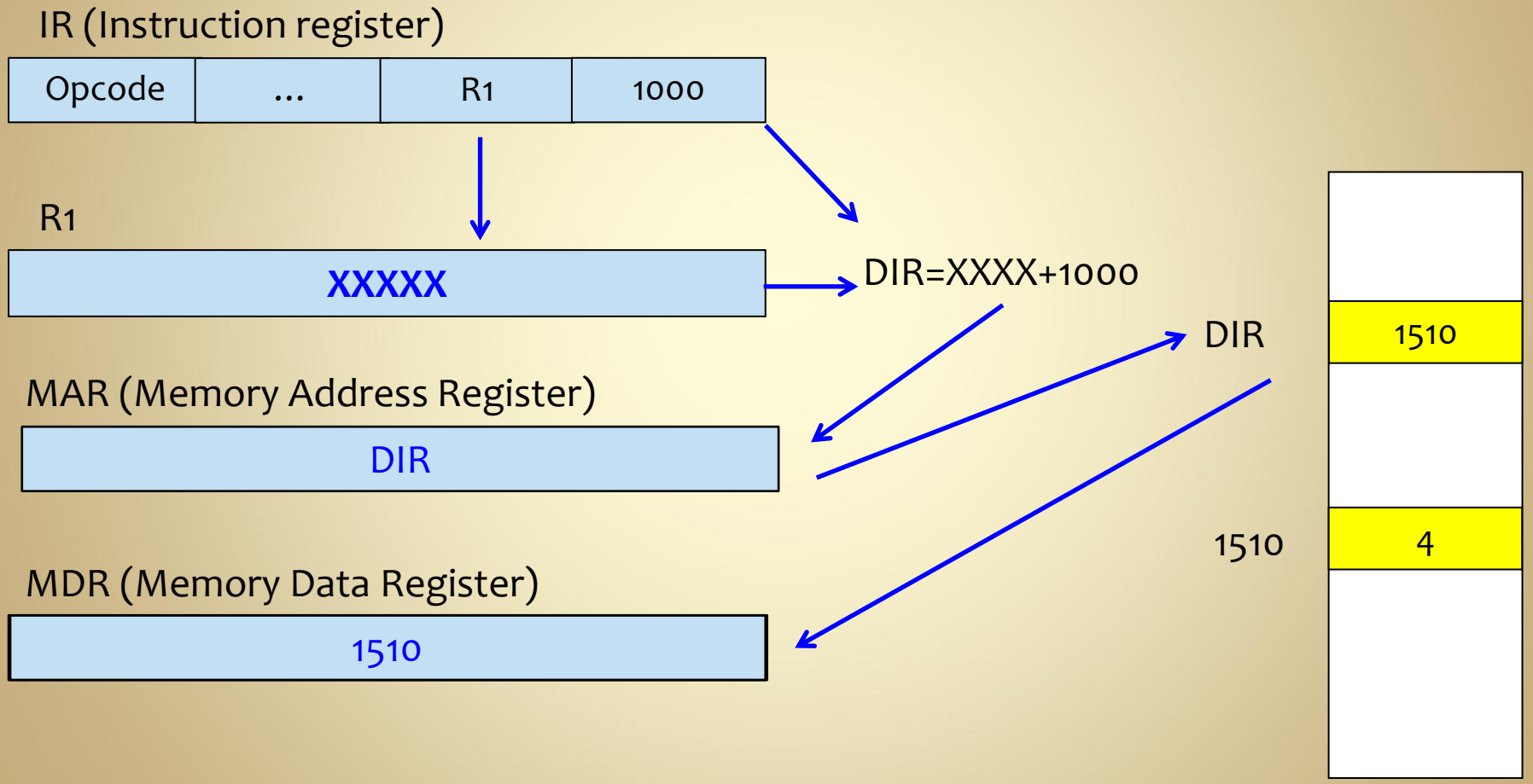
- Esta técnica permite acceder de forma fácil y rápida a posiciones cercanas de memoria.

- Por ejemplo, en MIPS:

```
lw $st0, 400($s1)
sw $st0, 400($s1)
```

- En estos ejemplos, el registro base es un registro cualquiera del banco de registros.

Direccionamiento relativo a registro base



6. Direccionamiento relativo a contador de programa (PC)

- Es un direccionamiento relativo a registro base, donde el registro base es el CONTADOR DE PROGRAMA (PC).

- Este direccionamiento se emplea para acceder a instrucciones o para referenciar datos cercanos al código.

7. Direccionamiento indexado

Muchos algoritmos exigen realizar operaciones sobre una secuencia de datos estructurados, almacenados en posiciones de memoria consecutivas.

Ej.: Bloque de n palabras en posiciones A, A+1, …, A+n-1 deben ser trasladadas a posiciones B, B+1, …, B+n-1.

Instrucción: MOVE A B (pasar el contenido de A a B)

Se puede modificar esta instrucción y hacer: MOVE A+1, B+1, y así n veces, hasta copiar las n palabras.

Pero muchas máquinas ofrecen un modo de direccionamiento para facilitar este tipo de operaciones: el direccionamiento indexado.

Direccionamiento indexado

En el **direccionamiento indexado** la dirección de memoria efectiva se calcula sumando el contenido de un registro de la CPU con el contenido de un campo de direcciones de la instrucción.

REPASO: En el direccionamiento relativo a registro base, la dirección de memoria efectiva se calculaba sumando el contenido del registro base (un registro de la CPU) con el desplazamiento (contenido en un campo de direcciones de la instrucción).

Por lo tanto, el **direccionamiento indexado** es similar al **direccionamiento relativo a registro base** con la diferencia de que *el desplazamiento no es un valor inmediato sino que está en un registro, que se llama* ***REGISTRO ÍNDICE***. Este registro índice se va incrementando/decrementando para recorrer los datos uno a uno.

Direccionamiento indexado

Este direccionamiento se usa tanto que muchas máquinas cuentan con registros índice especiales que automáticamente se incrementan/ decrementan. Esta modificación automática del registro índice se llama **AUTOINDEXACIÓN**:

- ***Direccionamiento autoincremental***: la dirección del operando se encuentra en un registro, y éste se incrementa automáticamente después de acceder al operando, en el tamaño del mismo.
- ***Direccionamiento autodecremental***: la dirección del operando se encuentra en un registro, y éste se decrementa automáticamente después de acceder al operando, en el tamaño del mismo.

Direccionamiento indexado

Direccionamiento autoincremental:

1. El direccionamiento post-incremental es útil para **manejar vectores y matrices.**
2. También es útil para **extraer datos de pilas,** ya que, si se aplica al puntero de pila, después de la operación de extracción, el puntero señalará al siguiente elemento de la pila.

Direccionamiento autodecremental:

1. Es útil para **introducir datos en pilas,** ya que, si se aplica sobre el puntero de pila conseguimos que antes de efectuar el acceso, el puntero señale al siguiente hueco libre de la pila.

8. Direccionamiento implícito

En este modo de direccionamiento el operando se especifica en la misma definición de la instrucción (en el OPCODE). El modo implícito se usa en dos casos:

- Cuando el OPCODE se refiere **a un registro especia**l. El operando está situado en este registro especial, no hace falta especificar donde está, es evidente que está ahí, no hay otro registro especial igual. Ej.: instrucciones que usan el ***acumulador***.

- Cuando el OPCODE se refiere a la **pila**. El operando está situado en la cima de la pila, no hace falta especificar dónde está. Ej.: push, pop.

La instrucción más corta es la que no tiene direcciones explícitas, únicamente aparece en ella el código de operación (OPCODE). Las direcciones de los operandos están implícitas en la definición del opcode.

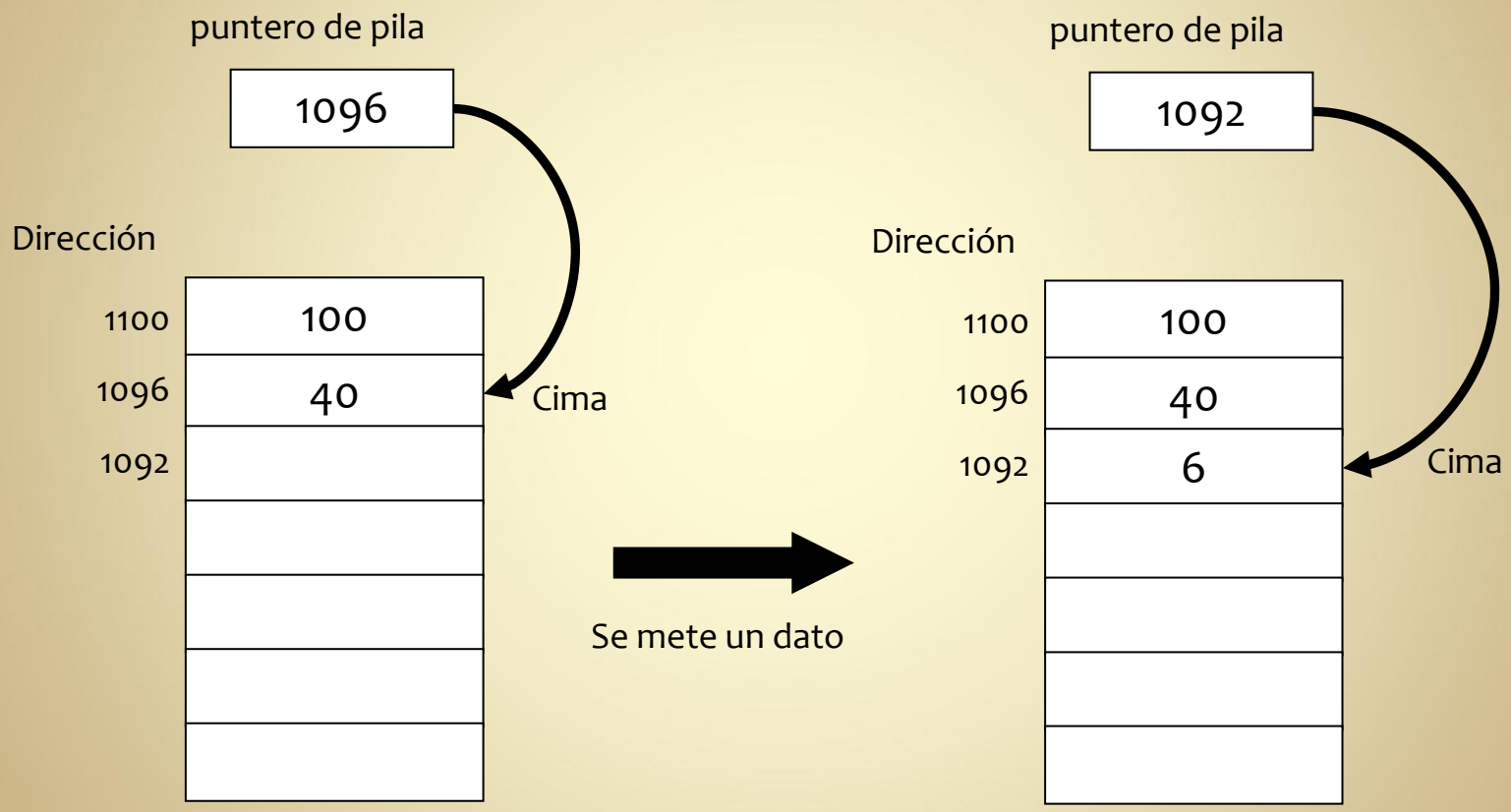
Direccionamiento implícito en pilas

- Una **PILA** es una estructura que consta de datos elementales almacenados en orden consecutivo en la memoria. La pila puede estar implementada en memoria o en registros.

- El primer dato introducido en la pila está en el ***fondo*** o ***base de la pila***. El dato introducido más recientemente está en la ***cima de la pila***. Hay un registro que contiene la dirección de la cima de la pila: el **PUNTERO DE PILA**. Por ejemplo, en MIPS, el puntero de pila es el registro $sp

NOTA: En MIPS la pila crece de direcciones mayores a direcciones menores

Direccionamiento implícito en pilas



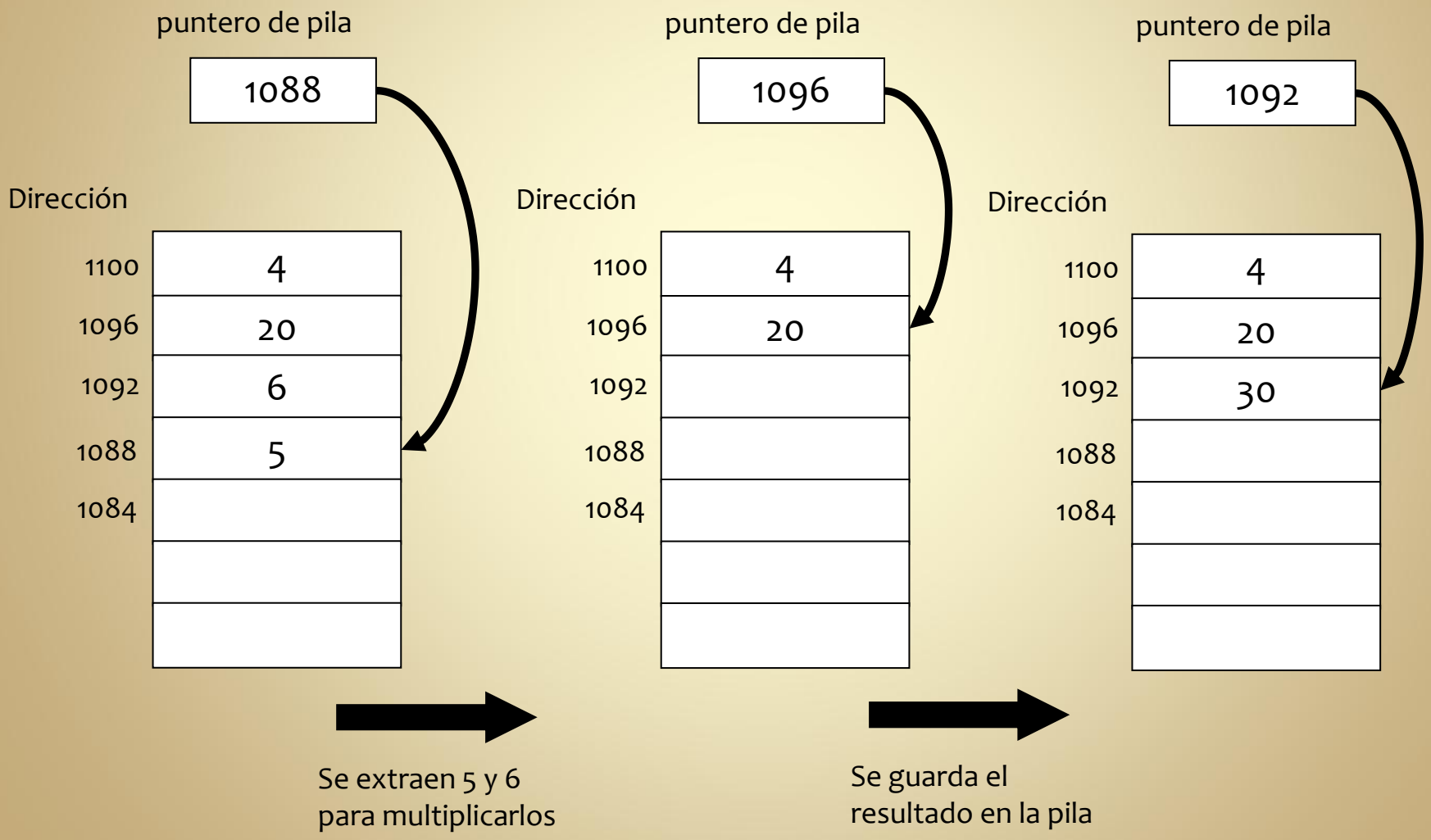
Direccionamiento implícito en pilas

- Las computadoras orientadas a pila tienen una instrucción para apilar datos (**PUSH**). Esta instrucción *copia el dato en la pila e incrementa el puntero de la pila*.

- Del mismo modo, también tienen una instrucción para extraer datos de la pila (**POP**). *Esta instrucción decrementa el puntero de pila, extrae el dato de la cima y lo guarda en un registro o en memoria.*

- Esta forma de direccionamiento implica que los operandos de una instrucción se "desapilan" uno detrás de otro, se realiza la operación, y el resultado se vuelve a introducir en la pila.

Direccionamiento implícito en pilas



Comparación de modos de direccionamiento

No todos los modos de direccionamiento se usan de igual modo. En general los compiladores de lenguajes de alto nivel usan:

1) **Autoindexado**: apilado/desapilado de parámetros de procedimiento.
2) **Directo**: acceso a variables globales.
3) **Inmediato**: operaciones con constantes.
4) **Indexado**: acceso a vectores, matrices y cadenas.
5) **De registro**: almacenamiento de variables locales.
6) **Indirecto**: almacenamiento de punteros a estructuras.

Comparación de modos de direccionamiento

Para comprender mejor los modos de direccionamiento es necesario comprender cómo se calcula la **dirección efectiva** (dirección donde se encuentra el operando):

- **Implícito:** La dirección efectiva se encuentra en el OPCODE.
  - Si se refiere a un registro especial (ej.: acumulador), el operando se encuentra en ese registro especial.
  - Si se refiere a PILA, el operando se encuentra en la PILA.

- **Inmediato:** No hace falta dirección, ya que el operando se encuentra directamente incluido en la palabra de instrucción.

- **Por registros:** El operando se encuentra en el registro direccionado en la palabra de instrucción.

  `Dirección Efectiva = dirección del registro`

Comparación de modos de direccionamiento

- **Directo a memoria:**
  Dirección Efectiva = dirección contenida en el campo de la instrucción correspondiente

- **Relativo a registro base:**
  Dirección Efectiva = Contenido del registro base + desplazamiento (valor inmediato)

- **Relativo a contador de programa:**
  Dirección Efectiva = Contenido del PC + desplazamiento (valor inmediato)

- **Directo Indexado:**
  Dirección Efectiva = Contenido del registro base + Contenido del registro índice (desplazamiento)