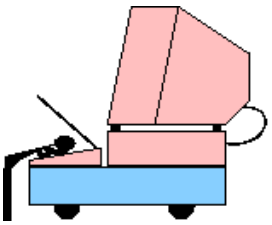
Fundamentos de Ordenadores 2



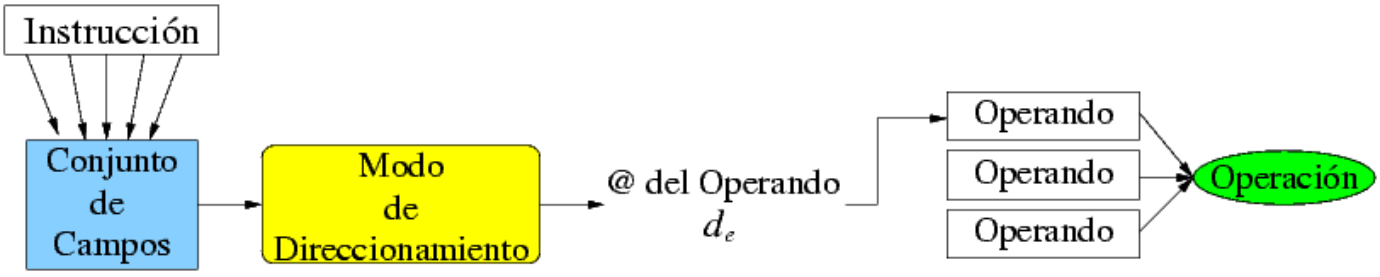
Modos de Direccionamiento del Intel Pentium

Abelardo Pardo
abel@it.uc3m.es

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática

Modos de Direccionamiento

- Las operaciones deben de obtener sus operados, ya sea de los **registros** o de la **memoria**.
- Para acceder a estos operandos se precisa especificar un **modo** de obtenerlos.
- Los procesadores permiten especificar la posición de un operando de múltiples maneras.
- **Modo de Direccionamiento:** Dado uno o varios campos de la instrucción, calcular la **dirección efectiva** (d_e) de un operando.
- La **Dirección Efectiva** no tiene por qué ser una dirección en memoria.



¿Cuántos modos de direccionamiento existen?

- Hay **infinidad** de formas de calcular la dirección de un operando.
- Cada procesador implementa un **subconjunto** de formas posibles.
- Tanto los operandos involucrados en el cálculo como las diferentes formas de calcular esta dirección están todos **codificados en la instrucción**.
- Cuanto más compleja sea la **tarea de cálculo de la dirección efectiva**, o más operandos incluya, más compleja será la decodificación y ejecución de la instrucción.

Terminología

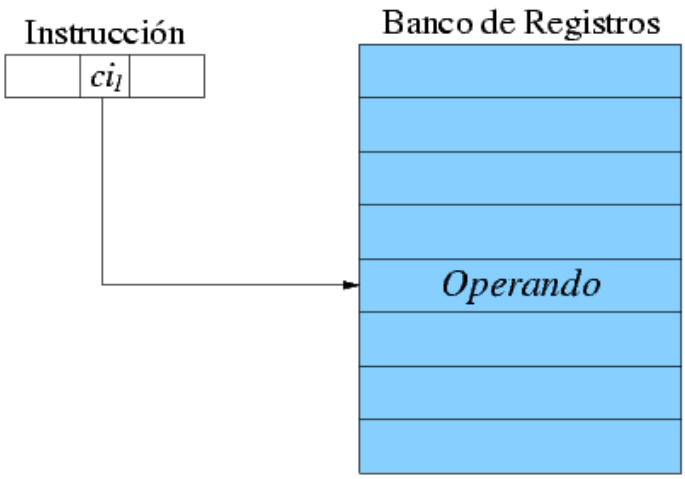
- Existen dos espacios posibles de direcciónes de operandos:

1. **Dirección de Registro:** Para aquellos operandos que están almacenados en registro.
2. **Dirección de Memoria:** Para aquellos operandos que están almacenados en memoria.

- Nos referiremos a los diferentes campos de la instrucción como ci_1, ci_2, etc.
- La expresión (ci_1) significa el contenido del campo ci_1 de la instrucción, o lo que es lo mismo, el número codificado en dicho campo.
- La expresión (R_i), donde R_i es el nombre de un registro, significa el contenido del registro.
- La expresión $dato \rightarrow R$ significa que se carga en el registro R el dato $dato$.

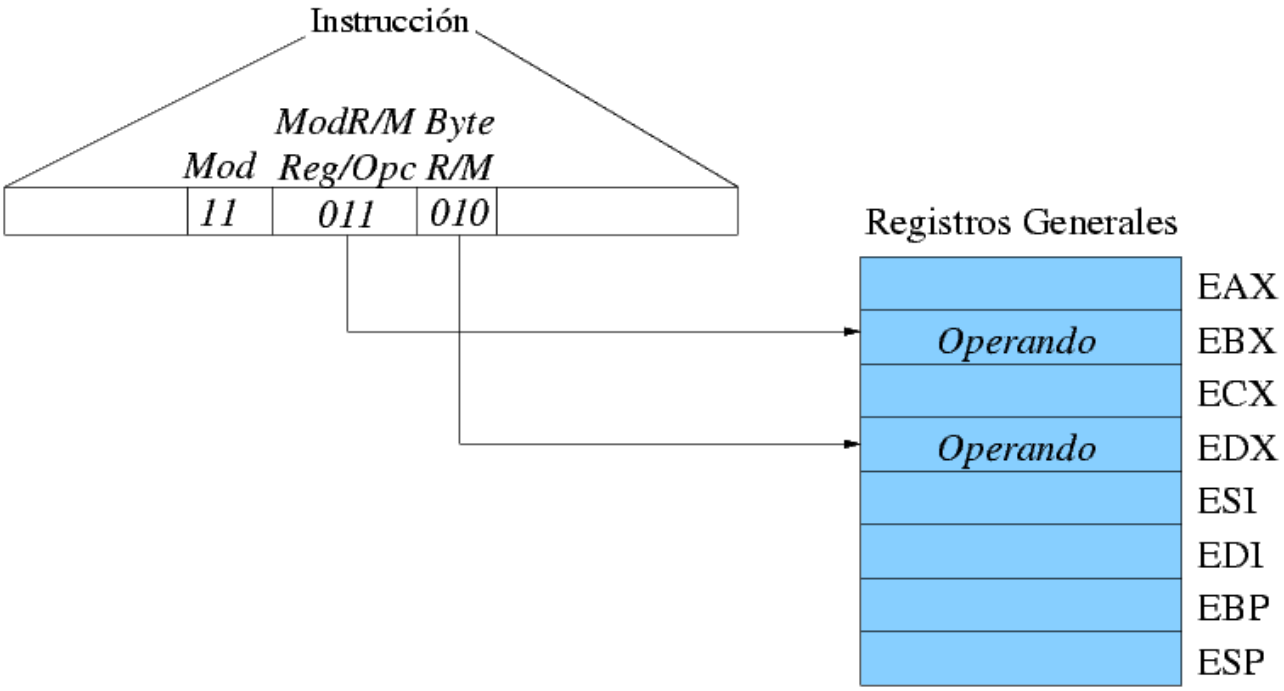
Modo Registro

- Los registros pueden almacenar **operandos**.
- La **dirección de registro** es diferente a la **dirección de memoria**.
- $d_e = (ci)$



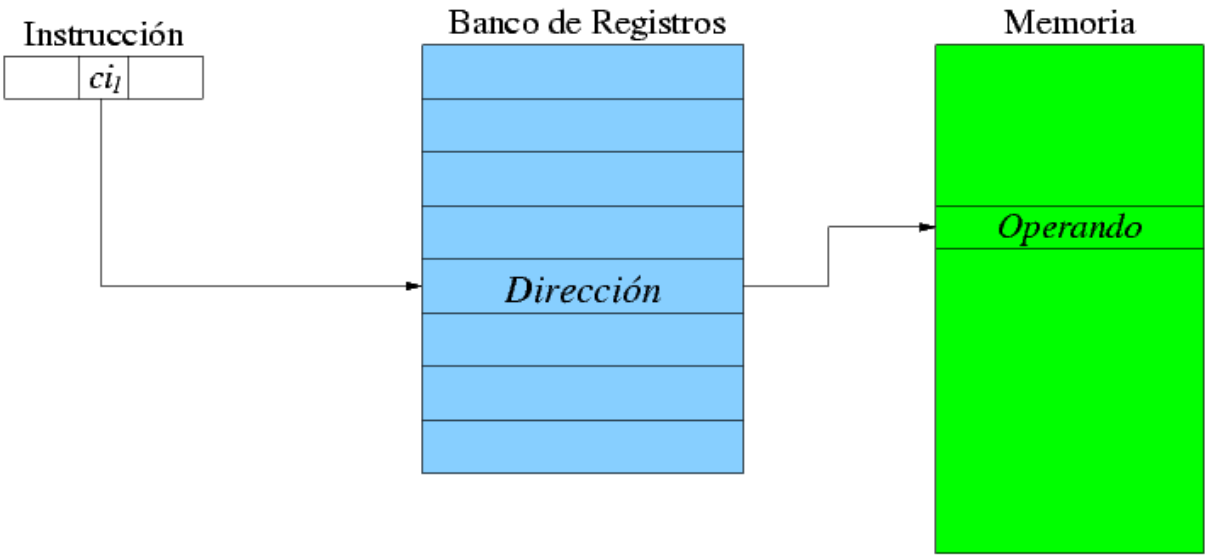
Modo Registro: Ejemplo

- **Ejemplo:** ADD %ebx, %edx
- Los campos que especifican los registros están en el byte **ModR/M**.



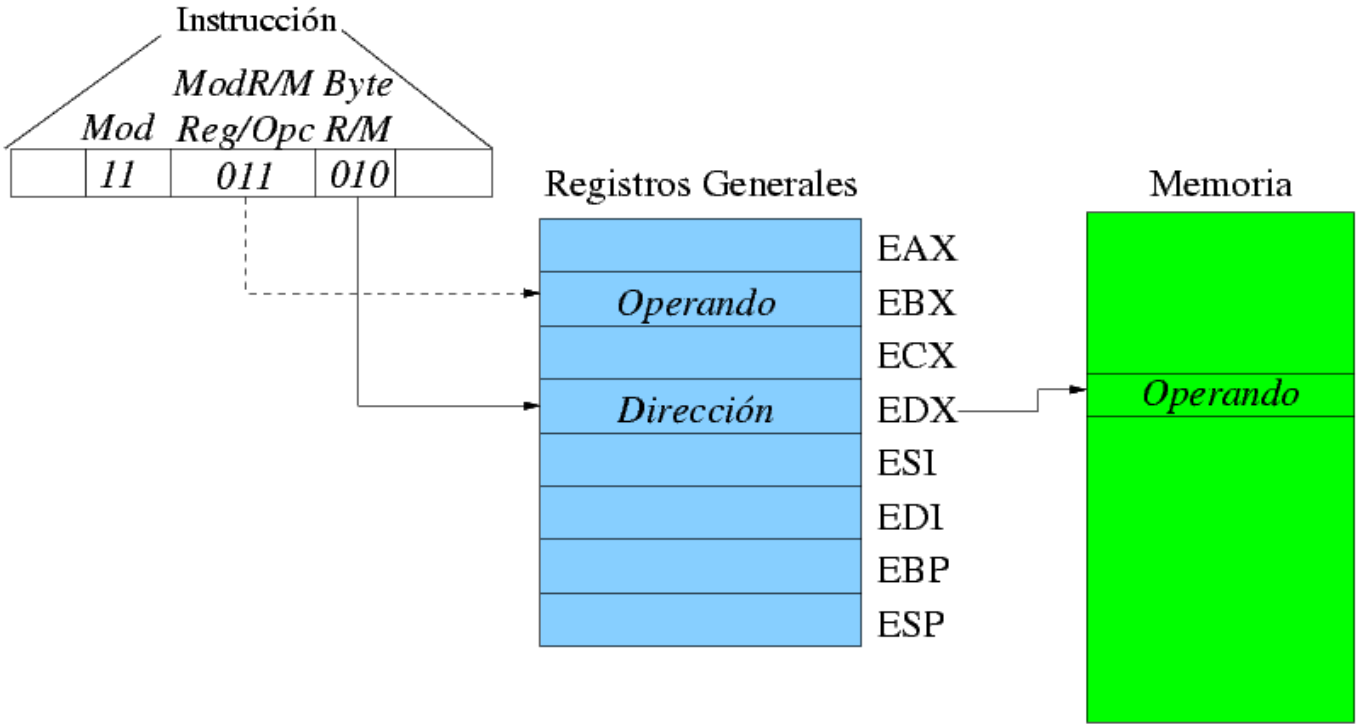
Modo Registro Indirecto

- La dirección efectiva está contenida en un **registro** codificado en la instrucción.
- Útil para acceder a un **array** de elementos.
- $d_e = ((ci))$



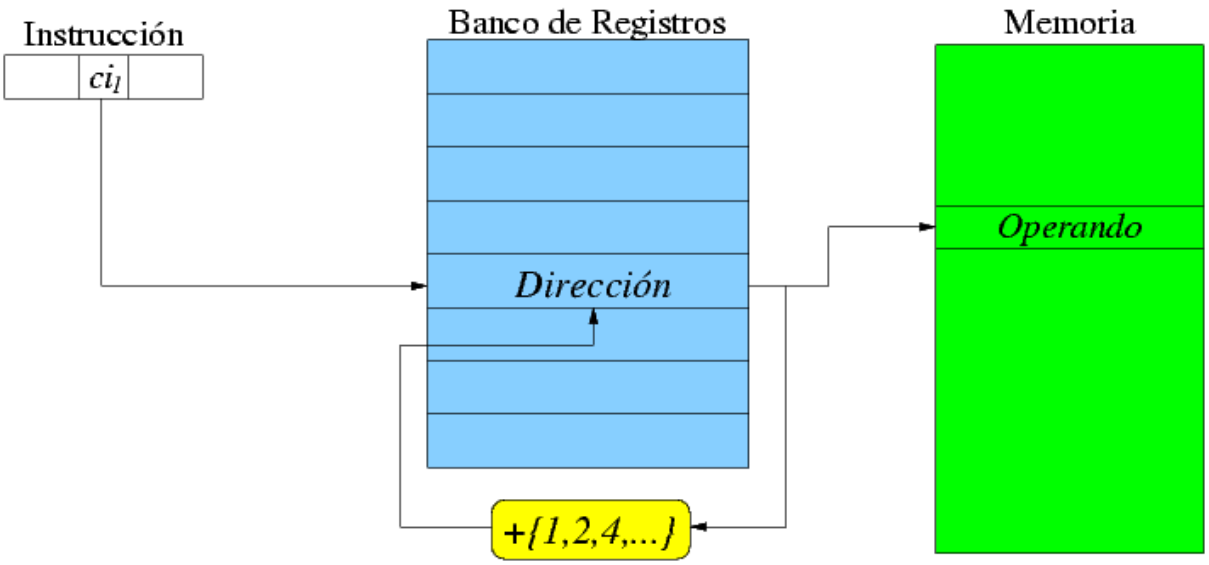
Modo Registro Indirecto: Ejemplo

- **Ejemplo:** ADD %ebx, (%edx)
- Los campos que especifican los registros están en el byte **ModR/M** y en el código de operación.



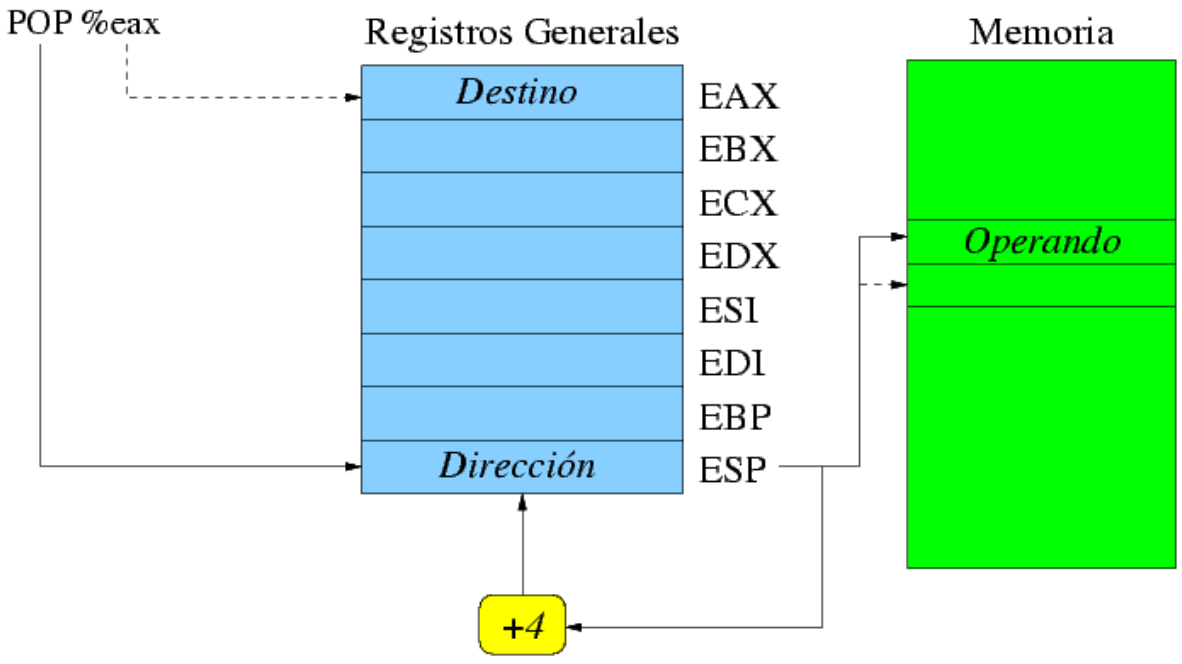
Modo Autoincremento

- El registro involucrado en el cálculo de la dirección efectiva se **incrementa** después de ser utilizado.
- El **tamaño del incremento** (1, 2, 4, etc) está relacionado con el **tamaño del operando** (8, 16, 32 bits, etc)
- $d_e = ((ci));\ (R_i) + \{1, 2, 4\} \rightarrow R_i$, donde R_i es el registro codificado en el campo ci.



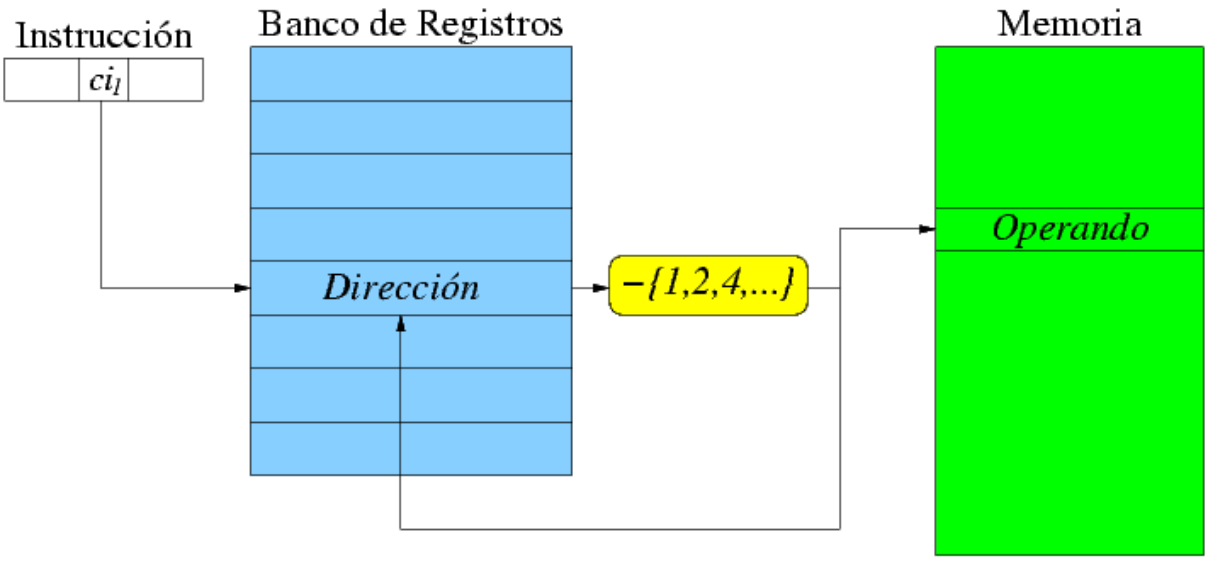
Modo Autoincremento: Ejemplo

- **Ejemplo:** POP %eax
- Como el tamaño del operando es **4 bytes** el incremento es de 4.



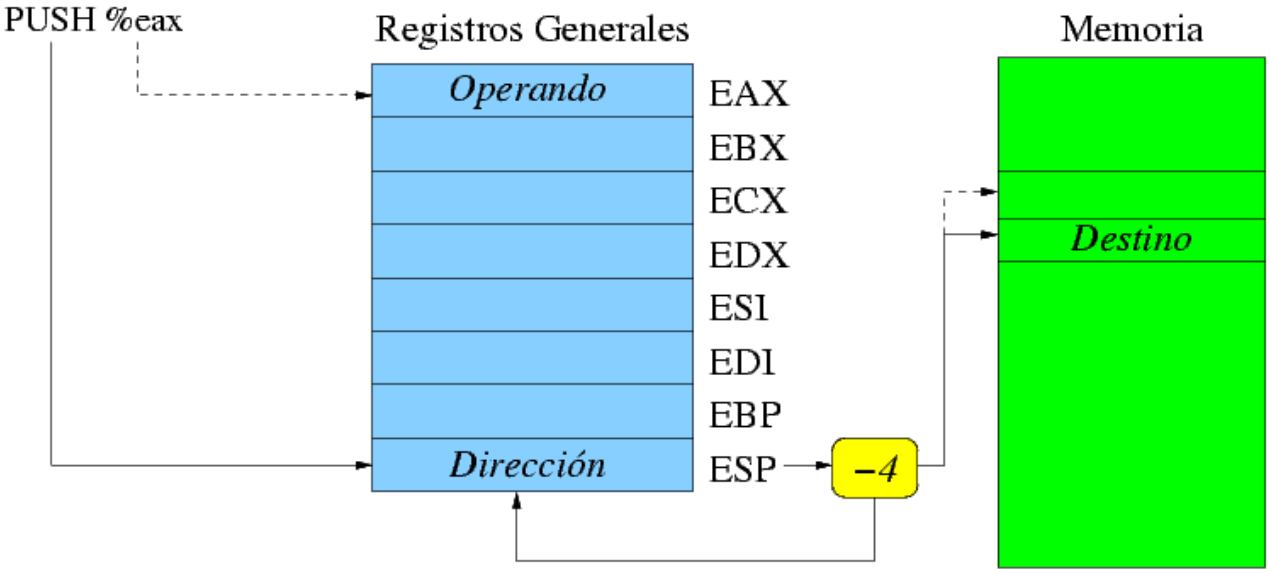
Modo Autodecremento

- El registro involucrado en el cálculo de la dirección efectiva se **decrementa antes** de ser utilizado.
- El **tamaño del decremento** (1, 2, 4, etc) está relacionado con el **tamaño del operando** (8, 16, 32 bits, etc)
- $d_e = ((ci)) - \{1, 2, 4\};\ (R_i) - \{1, 2, 4\} \rightarrow R_i$, donde R_i es el registro codificado en el campo ci.



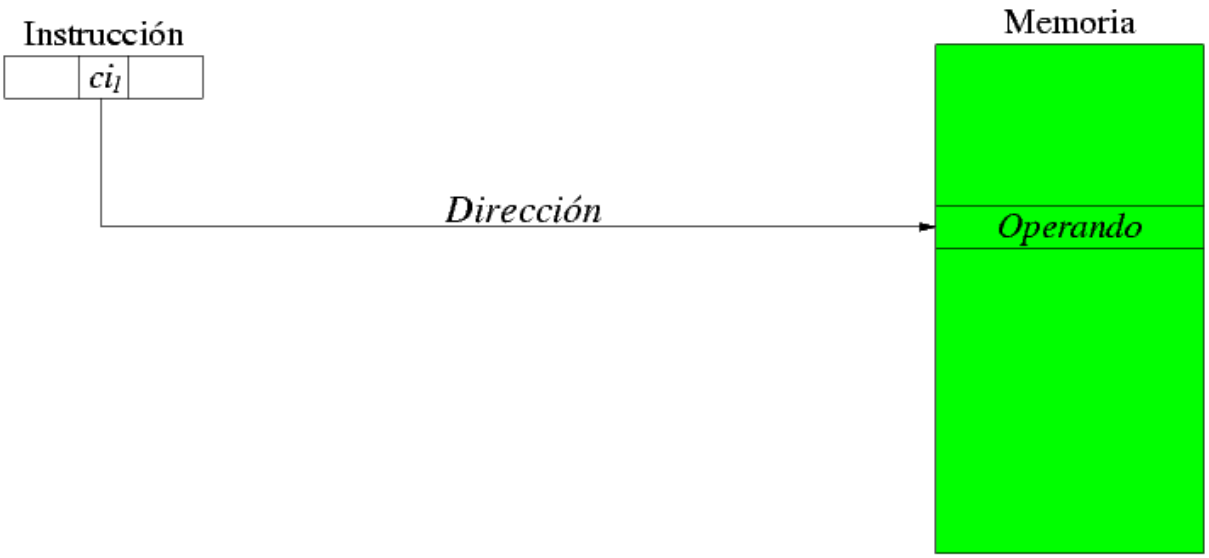
Modo Autodecremento: Ejemplo

- **Ejemplo:** PUSH %eax
- Como el tamaño del operando es **4 bytes** el decremento es de 4.



Modo Absoluto

- La dirección efectiva se encuentra en el **campo de la propia instrucción**.
- Útil para acceder a **constantes** de un programa.
- $d_e = (ci)$

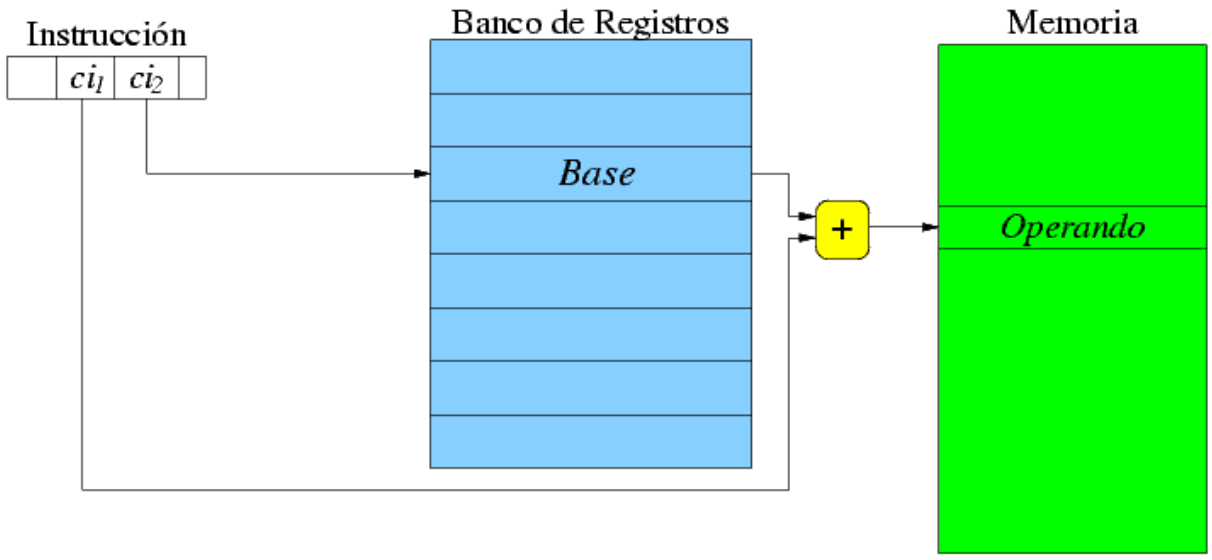


Modo Absoluto: Ejemplo

- **Ejemplo:** MOV mistring, %ebx
- Instrucción que carga el contenido en memoria con etiqueta **mistring** en el registro %ebx.
- El **lenguaje ensamblador** nos permite definir y utilizar estas etiquetas.

Modo Base + Desplazamiento

- La dirección se obtiene sumando el **contenido de un registro** y un valor codificado en la instrucción.
- Útil para acceder a **un campo de un record de datos**.
- $d_e = (ci_1) + ((ci_2))$

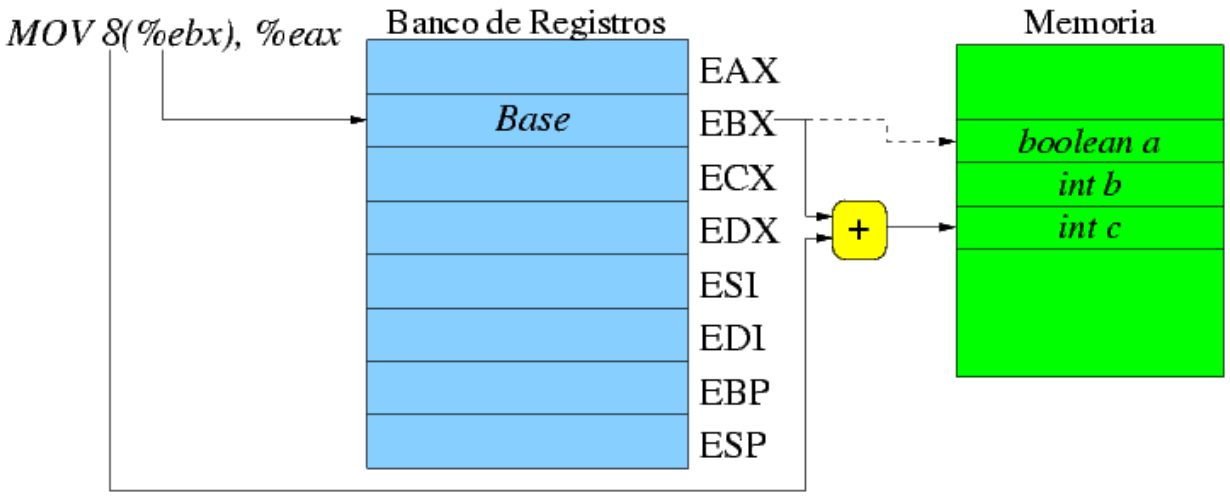


Modo Base + Desplazamiento: Ejemplo

- Supongamos un **record** almacenado en una dirección **contenida en %ebx** y con **campos**:

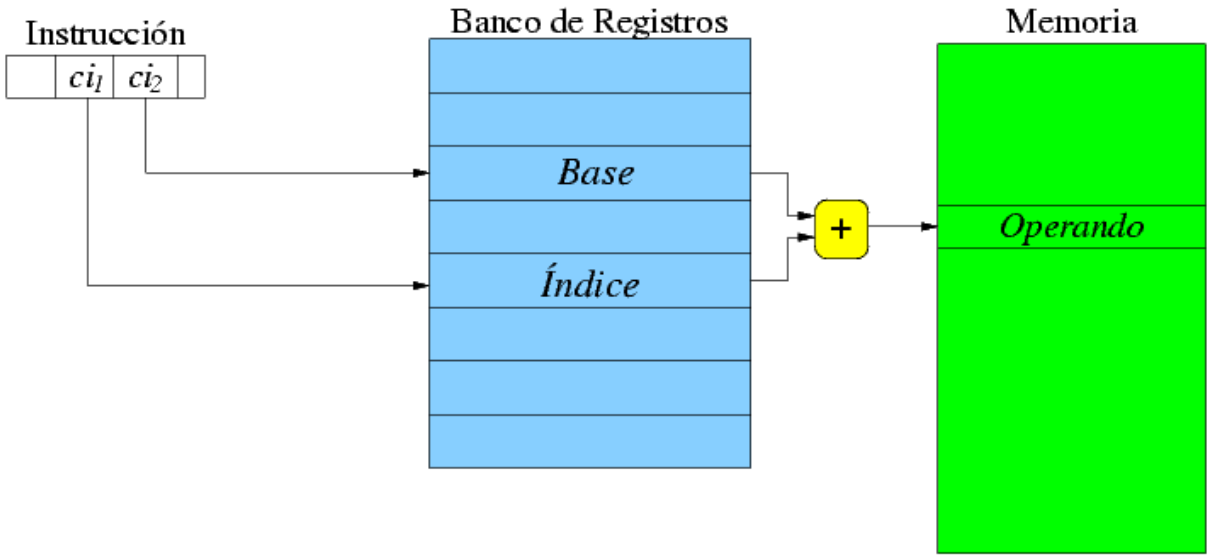
```
boolean a; /* 4 bytes */
int b;     /* 4 bytes */
int c;     /* 4 bytes */
```

- ¿Cómo cargamos el valor del campo **c** en el registro **%eax**?



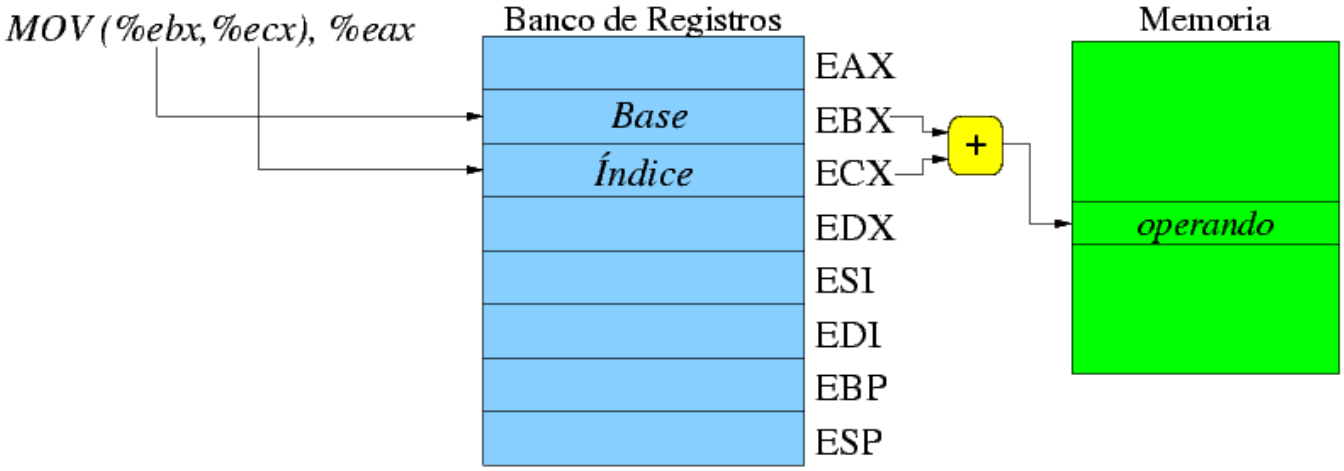
Modo Base + Índice

- La dirección se obtiene sumando el **contenido de dos registros**.
- Útil para acceder a los elementos de un array **secuencialmente**, pues se conserva la dirección de inicio de los datos y el índice.
- $d_e = ((ci_1)) + ((ci_2))$



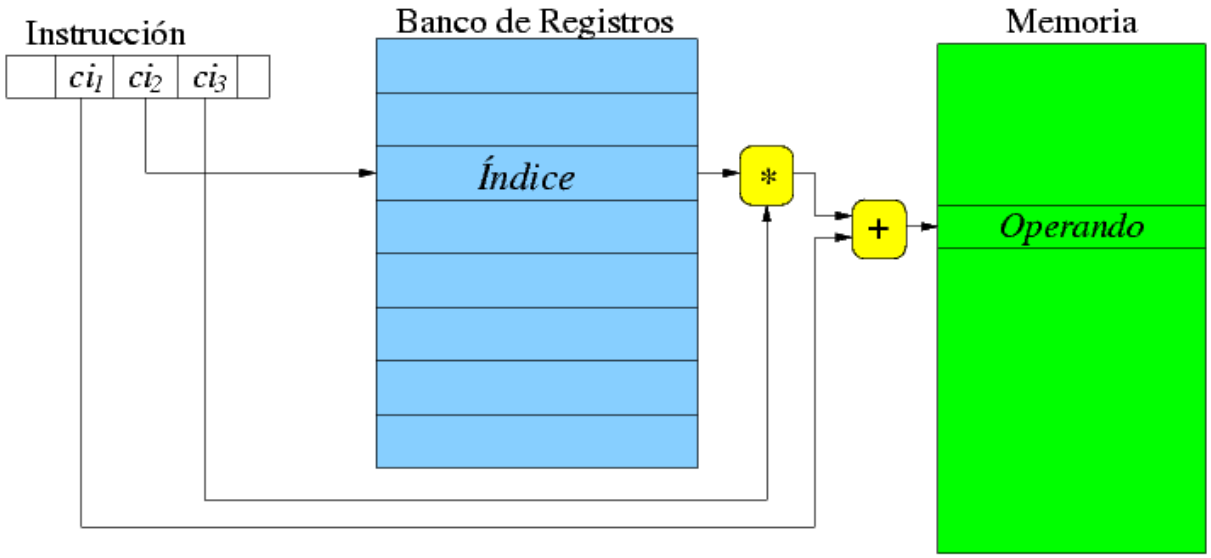
Modo Base + Índice: Ejemplo

- Supongamos el **array de enteros** almacenado en la posición contenida en el registro %ebx.
- ¿Cómo podemos **iterar** sobre **todos** los elementos del array?



Modo Índice Escalado + Desplazamiento

- La dirección se obtiene sumando el registro índice escalado al valor codificado en la instrucción.
- Útil para acceder a **arrays que están en posiciones fijas** y que contienen elementos de tamaños 1, 2, 4 u 8 bytes.
- $d_e = (ci_1) + ((ci_2)) * (ci_3)$; **tal que** $(ci_3) \in \{1, 2, 4, 8\}$

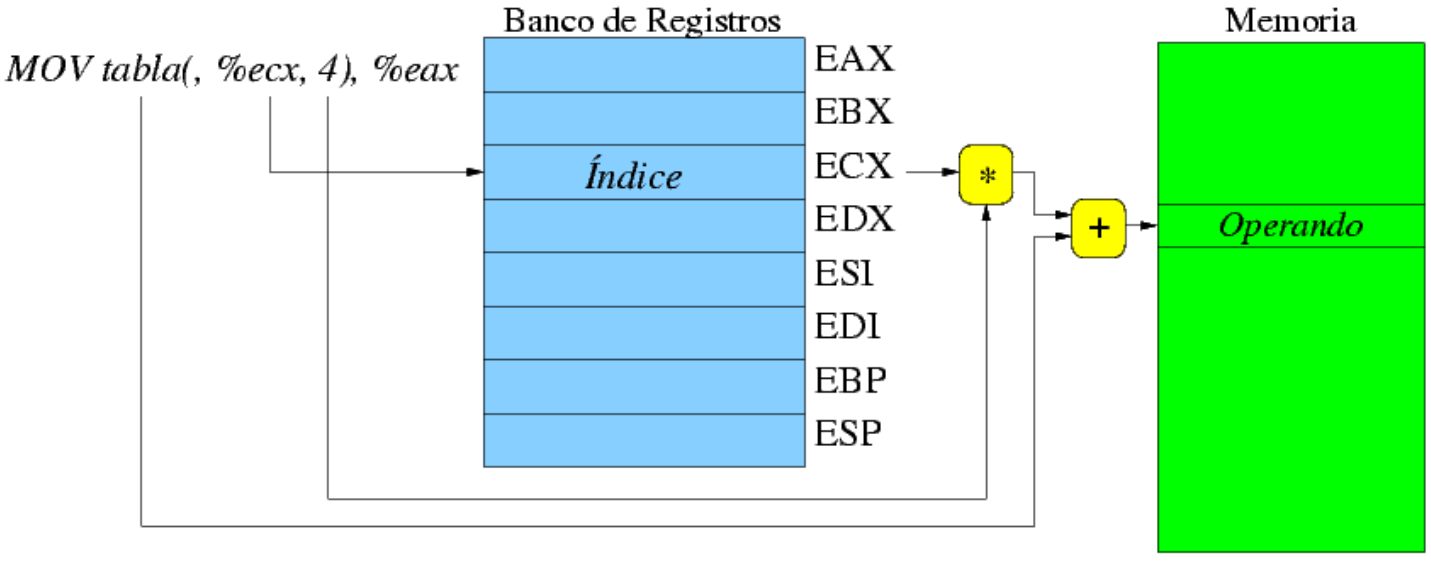


Modo Índice Escalado + Desplazamiento: Ejemplo

- Supongamos un **array de enteros** como variable global del programa con la siguiente estructura:

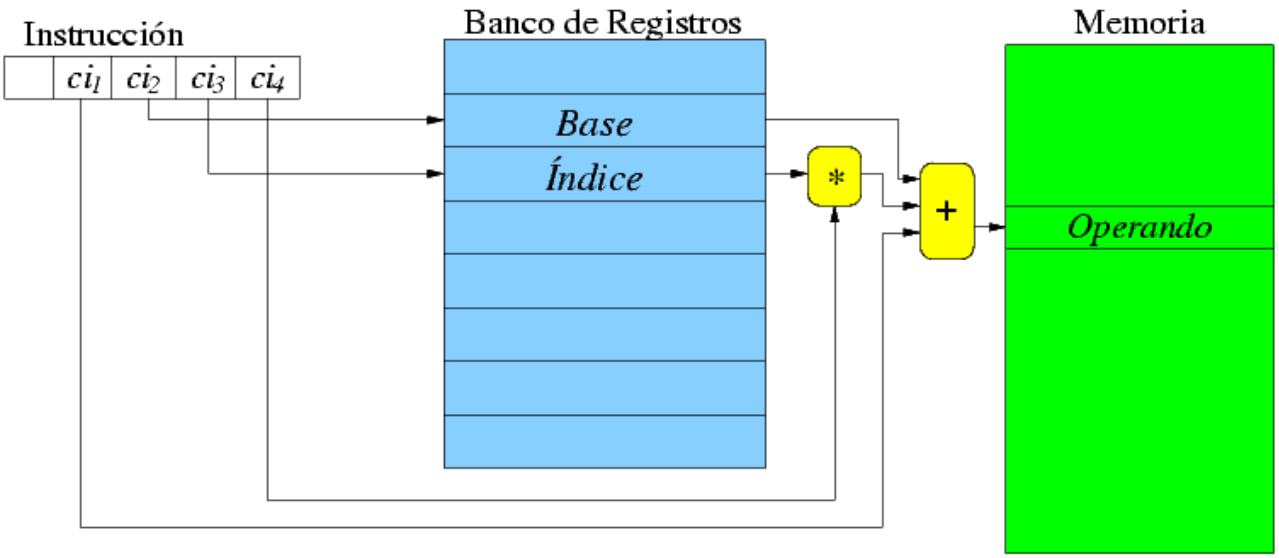
tabla	100	200	300	...

- Para **iterar** sobre los elementos de la tabla **no necesitamos almacenar la dirección base en un registro**.



Modo Base + Índice Escalado + Desplazamiento

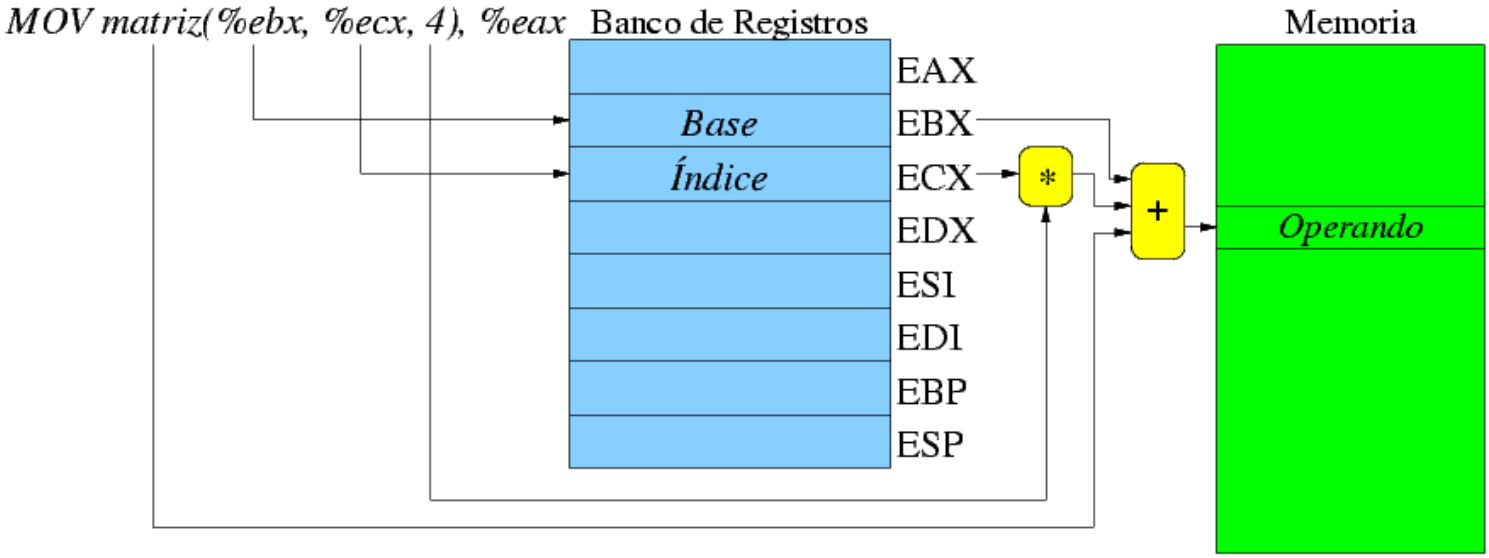
- Combinación de los modos **base + índice** e **índice escalado + desplazamiento**.
- La dirección se obtiene sumando el desplazamiento, el registro base y el registro índice multiplicado por el factor de escala pertinente.
- $d_e = (ci_1) + ((ci_2)) + ((ci_3)) * (ci_4)$; **tal que** $(ci_4) \in \{1, 2, 4, 8\}$



Modo Base + Índice escalado + Desplazamiento: Ejemplo

- Supongamos un array de dos dimensiones (una matriz) con elementos de 4 bytes.
- Para acceder a un elemento se precisan índices de fila y columna. La dirección del elemento en la posición (i, j) es:

$$\text{base} + (i * \text{tamañoFila}) + (j * \text{tamañoElemento}).$$



Resumen de Modos de Direccionamiento

Registro: Operando en regisro. $d_e = (ci)$.

Registro Indirecto: Operando en dirección de memoria en registro. $d_e = ((ci))$.

Autoincremento: Como el Registro indirecto, pero el registro se incrementa **después** de ser usado. $d_e = ((ci));\ (R_i) + \{1, 2, 4\} \rightarrow R_i$, donde R_i es el registro codificado en el campo ci.

Autodecremento: Como el Registro indirecto, pero el registro se decrementa **antes** de ser usado. $d_e = (R_i) - \{1, 2, 4\};\ ((ci)) - \{1, 2, 4\} \rightarrow R_i$, donde R_i es el registro codificado en el campo ci.

Absoluto: La dirección está codificada en la propia instrucción. $d_e = (ci)$.

Modo Base + Índice escalado + Desplamiento: La dirección se puede codificar hasta con cuatro operandos, permitiendose todas las combinaciones (índice y escala se tratan como un parámetro).

$$d_e = (\text{Registro Base}) + [(\text{Registro Índice}) * \text{Escala}] + \text{Desplazamiento}$$