



Instituto Universitario Politécnico

“Santiago Mariño”

Unidad N° II – Capacidad de la unidad central de procesamiento

Organización de instrucción de máquina

Cada instrucción requiere la información que necesita el CPU para su ejecución.

El formato de las instrucción es un conjunto de especificaciones que indican como debe ser interpretado el patrón de bits de una instrucción de máquina para lograr su ejecución dentro del computador.

El formato de la instrucción nos indica cual es el código de operación y cuales los operandos que la instrucción especifica, tanto explícita como implícitamente .

Una instrucción está conformada por: Código de Operación, referencia a operandos fuente, referencia al operando resultado, referencia a la siguiente instrucción.

Código de Operación (OpCode)

Es el primer campo de cada instrucción, esta indica la instrucción a realizar. Por ejemplo, ADD (sumar). Muchas instrucciones tienen un campo adicional que especifica el operando. Por ejemplo, las instrucciones que accesan a una variable local necesitan un campo para indicar cuál variable.

En los formatos de instrucciones comunes puede haber cero, una, dos o tres direcciones presentes.

Código de Operación (OpCode)

Código de Operación

Cód. Op

Dirección

Cód. Op.

Dirección 1

Dirección 2

Cód. Op.

Dir1

Dir2

Dir3

Clasificación de las instrucciones de máquina

Tipo de instrucción	Ejemplos
Transferencia de datos	Movimiento (move), alteración de datos (clear, inc, dec), rotación bits (shift, rotate)
Aritméticas	Add, Sub, Mult, Div
Lógicas	And, Or, Xor
Booleanas	Set bit, Clear bit, jump if bit set, jump if bit clear
De salto	Control (Jump, Conditional jumps), relacionada con subrutinas (Push, Pull), Relacionadas con interrupción (Retorno de int.)

Direccionamiento

Una instrucción consiste en un código de operación (opcode) generalmente acompañado con información adicional como la dirección de los operandos, el destino de los resultados y la dirección de la siguiente. El tema general de especificar dónde están los operandos (es decir, sus direcciones) se llama direccionamiento.

Considere una instrucción ADD que requiere la especificación de tres operandos: dos fuentes y un destino. De alguna manera, la instrucción ADD debe indicar dónde están los operandos y dónde debe poner se el resultado. Si las direcciones de memoria tienen 32 bits, la especificación de esta instrucción requiere direcciones de 32 bits además del código de operación. Las direcciones ocupan muchos más bits que el código de operación.

Modos de Direccionamiento

Cada instrucción debe contener la información que necesita el CPU para su ejecución:

- **Código de operación**
- **Referencia a operandos fuente (dirección)**
- **Referencia a operando resultado (dirección)**
- **Referencia a la siguiente instrucción.**

Los operandos fuente y resultado pueden estar en la memoria principal, el registro del CPU o en un dispositivo de E/S.

Modos de Direccionamiento

El direccionamiento toma en cuenta dos cosas:

1. **Cómo se especifica la dirección de un operando o referencia.**
2. **Cómo se organizan los bits de una instrucción para definir las direcciones de los operandos y la operación que realiza dicha instrucción.**

Es por ello que se emplean diversas técnicas de direccionamiento. Todas ellas implican algún compromiso entre el rango de direcciones y/o flexibilidad de direccionamiento de una parte, y por otro lado, el número de referencias a memoria y/o la complejidad del cálculo de las direcciones.

Modos de Direccionamiento

Las técnicas de direccionamiento más comunes son las siguientes:

- Inmediato
- Directo
- Indirecto
- Registro
- Indirecto con registro
- Con desplazamiento
- Pila

La unidad de control a menudo utiliza los códigos de operación para indicar el modo de direccionamiento o también se emplean bits del formato de instrucción para indicar el modo.

Modos de Direccionamiento

Suponga que estamos trabajando con una arquitectura que posee 4 registros (\$R0,\$R1,\$R2 y \$R3) y la memoria es direccionada por palabras. El conjunto de instrucciones para esta arquitectura está conformado por las siguientes operaciones:

add op1, op2

op1 = op1 + op2

sub op1, op2

op1 = op1 – op2

move op1, op2

op1 = op2

\$R0	1
\$R1	0
\$R2	2
\$R3	0

Registros
Del CPU

0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

A	5
B	8
C	3

Tabla de
símbolos

Modo Inmediato

El operando está presente en la propia instrucción. Es decir el campo dirección en la instrucción contiene el operando.



Operando = A

Este modo puede utilizarse para definir y usar constantes, o para fijar valores iniciales de variables.

Ventaja: No requiere referencia a memoria para obtener el operando.

Desventaja: El tamaño del número está restringido a la longitud del campo de direcciones.

Modo Inmediato

El modo inmediato representado con el símbolo #

Ejemplo: *move \$R1 , #2*

\$R0	1
\$R1	0
\$R2	2
\$R3	0

Registros
Del CPU

\$R0	1
\$R1	2
\$R2	2
\$R3	0

Registros
Del CPU

Modo Directo

El campo de dirección contiene la dirección efectiva (EA) del operando.



Características:

- La instrucción contiene la EA del operando
- Un acceso adicional es requerido para obtener el operando
- El rango de direcciones está limitado por el ancho del campo que contiene la dirección del operando.
- La dirección es una constante a tiempo de ejecución pero el operando en sí puede variar en tiempo de ejecución.

Modo Directo

Modo directo representado por: la etiqueta o número.

Ejemplo: *move \$R0 , 2*

⌘R0	≠	10
⌘R1		2
⌘R2		2
⌘R3		0

Registros
Del CPU



0	3
1	4
2	10
3	9
4	2
5	0
6	1
7	4
8	7
9	3
10	4
...	

Memoria

Modo Indirecto

El modo directo presenta el problema de que la longitud del campo de direcciones es normalmente menor que la longitud de una palabra, limitando de esta manera el rango de direcciones. Una solución es hacer que el campo de direcciones referencie la dirección de una palabra de memoria que contiene la dirección completa del operando. A esto se le conoce como modo indirecto.

Cod. Op.

A

...

EA = (A)

Operando = ((A))

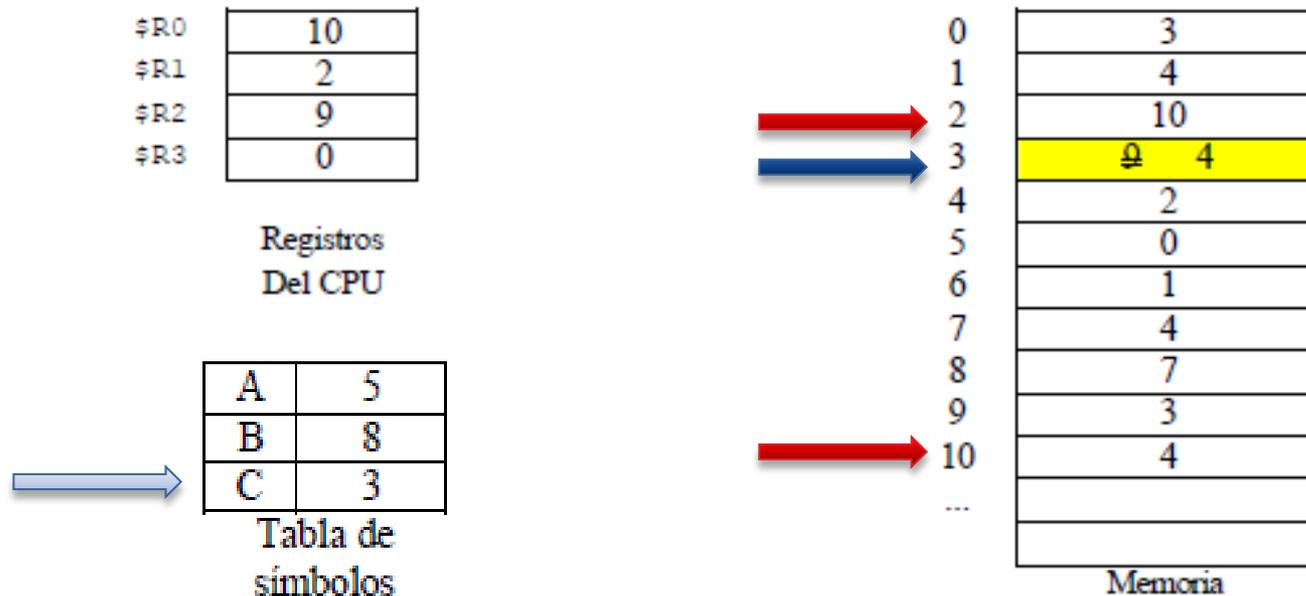
Ventajas: Permite acceder el espacio de direcciones. No hay limitación como en modo directo.

Modo Indirecto

Desventaja: La ejecución de la instrucción requiere de dos referencias a memoria para capturar el operando: una para captar su dirección y otra para obtener su valor.

El modo indirecto será representado por el símbolo @

Ejemplo: *move C, @2*



Modo Registro

Modo similar al directo. La única diferencia es que el campo de dirección referencia un registro en lugar de una dirección de memoria principal.



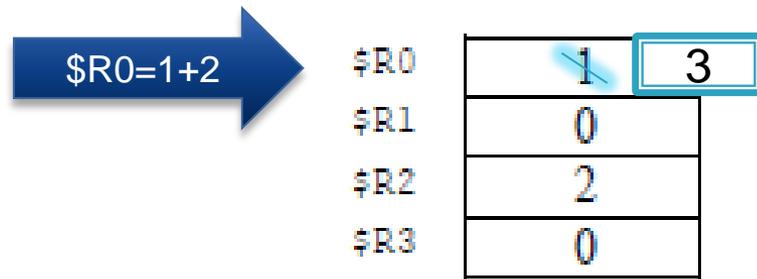
Ventajas: Sólo es necesario un campo pequeño de direcciones en la instrucción y no se requieren referencias a memoria. El tiempo de acceso a un registro interno al CPU es mucho menor que el tiempo necesario para acceder memoria principal.

Desventaja: El espacio de direcciones está muy limitado dado al pequeño número de registros que poseen las arquitecturas.

Modo Registro

Modo registro está representado por: $\$R_i$, donde i es el número de registro.

Ejemplo: *add \$R0, \$R2*



Registros
Del CPU

Modo Indirecto con Registro

Similar al indirecto. La diferencia está en que en el indirecto el campo de dirección hace referencia a un posición de memoria que contiene la dirección del operando y en el indirecto con registro el campo de dirección hace referencia a un registro que contiene la dirección efectiva del operando

Cod. Op.

Rn

...

EA = (Rn)

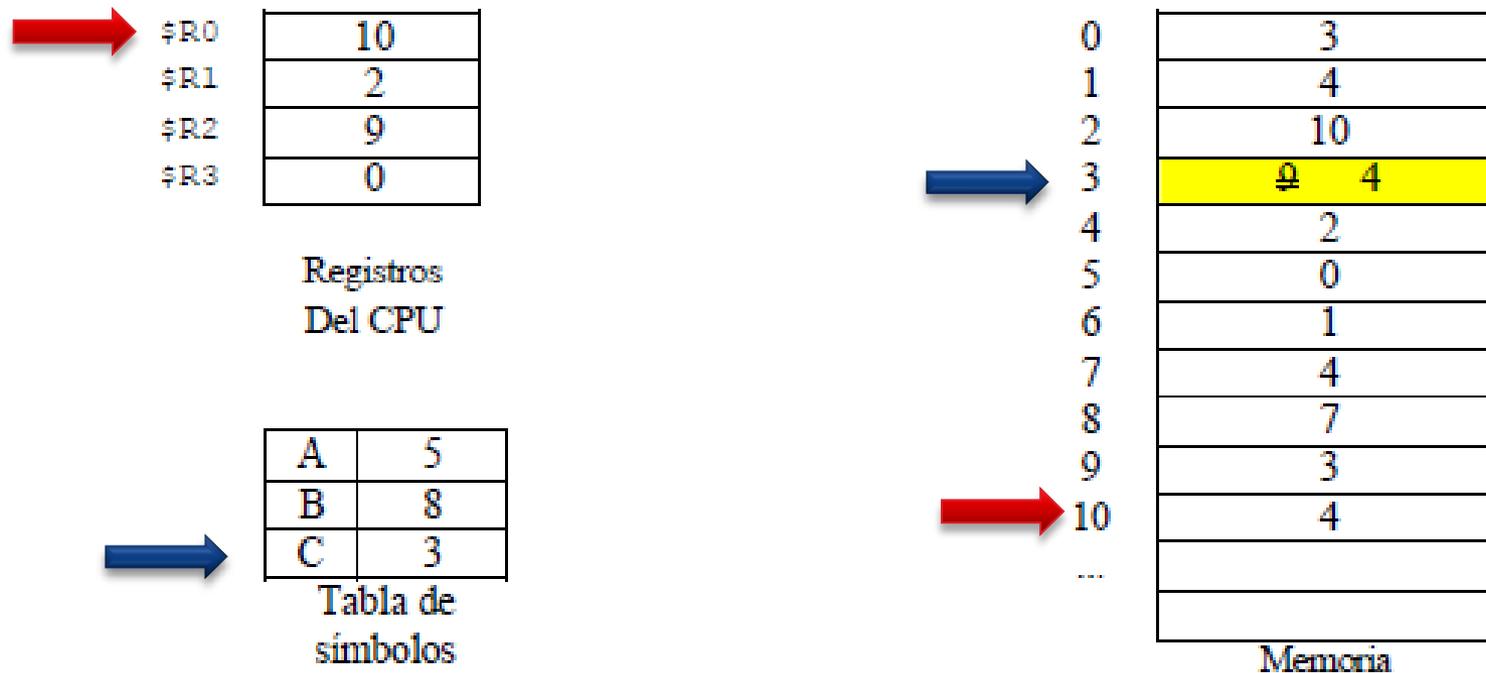
Operando = ((Rn))

Ventajas: Emplea una referencia menos a memoria que el modo indirecto.

Desventaja: La ejecución de la instrucción requiere una referencia a memoria para capturar el operando.

Modo Indirecto con Registro

Ejemplo: *move C, @\$R0*



Modo con Desplazamiento

Este modo combina las posibilidades de los modos directo e indirecto con registro. Requiere que las instrucciones tengan dos campos de dirección donde al menos uno de ellos es explícito. El valor contenido en uno de los campos de dirección se utiliza directamente. El otro campo de dirección o referencia implícita definida por el código de operación se refiere a un registro cuyo contenido se suma al primer campo de dirección para generar la dirección efectiva del operando.

Cod. Op.

Rn

A

$$EA = A + (Rn)$$

$$\text{Operando} = (A + (Rn))$$

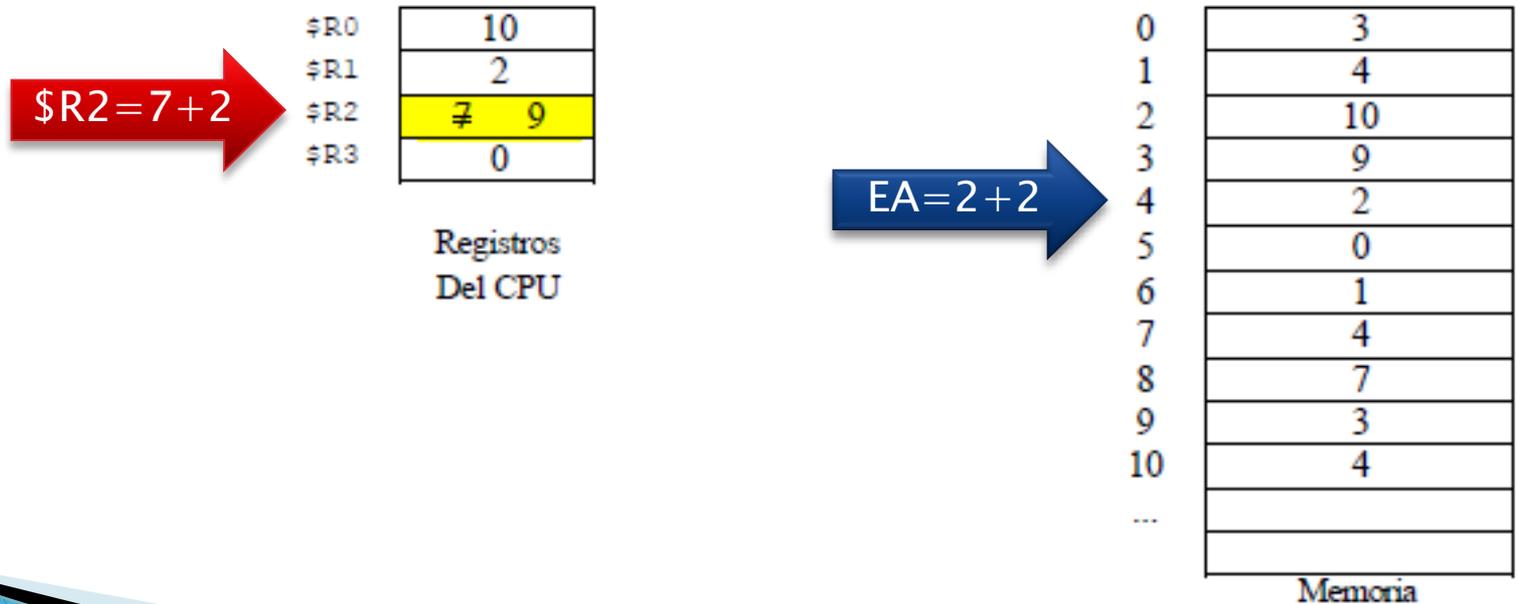
Hay tres formas distintas de modos con desplazamiento: desplazamiento relativo, desplazamiento con registro base e indexado.

Modo con Desplazamiento

Se representa el modo desplazamiento así: $n(\$Ri)$

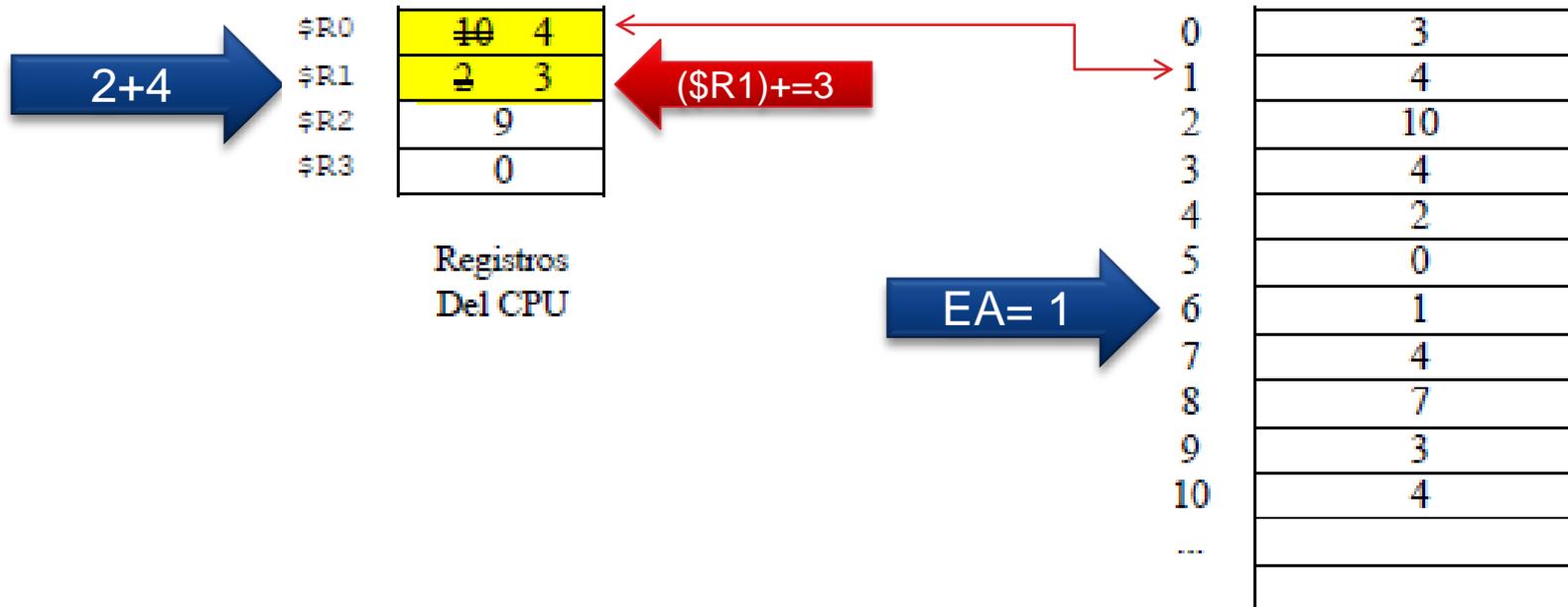
Para el modo desplazamiento autoincremento: $n(\$Ri) +$ (Después de calcular la dirección del operando incrementa en 1 $\$Ri$).

Ejemplo de modo de desplazamiento: $add \$R2, 2(\$R1)$



Modo con Desplazamiento

Ejemplo de modo desplazamiento indirecto con incremento: `move $R0 , @4 ($R1) +`



Modo de Pila

La pila es una matriz lineal de posiciones. A veces denominada lista de apilamiento o cola LIFO. La pila es un bloque de posiciones reservado en memoria. Los elementos se añaden en la cabecera de manera que, en cualquier instante, el bloque está parcialmente lleno. La pila tiene asociado un puntero (SP) que hace referencia a la cabecera o tope de la pila. Dicho puntero se usa como puntero de direccionamiento. Tanto la cabecera como el puntero pueden residir en registros de procesador.

Este modo es una forma de direccionamiento implícito. Las instrucciones máquina no necesitan incluir referencia a memoria sino que operan implícitamente con la cabecera de la pila.