

Repertorios de instrucciones: modos de direccionamiento y formatos

MSc. Elvin Rojas Ramírez

DIRECCIONAMIENTO

El campo o campos de direcciones en un formato de instrucción usual están bastante limitado. Sería deseable poder **referenciar un rango elevado de posiciones de memoria principal o, en algunos sistemas, de memoria virtual**. Para conseguir este objetivo se han empleado diversas técnicas de direccionamiento. Las más comunes son:

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

DIRECCIONAMIENTO ...

Todas las arquitecturas de computadoras ofrecen más de uno de estos modos de direccionamiento.

Tabla 10.1. Modos de direccionamiento básicos

Modo	Algoritmo	Principal ventaja	Principal desventaja
Inmediato	Operando = A	No referencia a memoria	Operando de magnitud limitada
Directo	EA = A	Es sencillo	Espacio de direcciones limitado
Indirecto	EA = (A)	Espacio de direcciones grande	Referencias a memoria múltiples
Registro	EA = R	No referencia a memoria	Número limitado de registros
Indirecto con registro	EA = (R)	Espacio de direcciones grande	Referencia extra a memoria
Con desplazamiento	EA = A + (R)	Flexibilidad	Complejidad
Pila	EA = cabecera de la pila	No referencia a memoria	Aplicabilidad limitada

Direccionamiento Inmediato

El operando esta en realidad presente en la propia instrucción:

OPERANDO A

Se utiliza para definir y utilizar constantes, o para fijar valores iniciales de variables.

La **ventaja** del direccionamiento inmediato es que, una vez captada la instrucción, **no se requiere una referencia a memoria** para obtener el operando, ahorrándose pues un ciclo de memoria o de cache en el ciclo de instrucción.

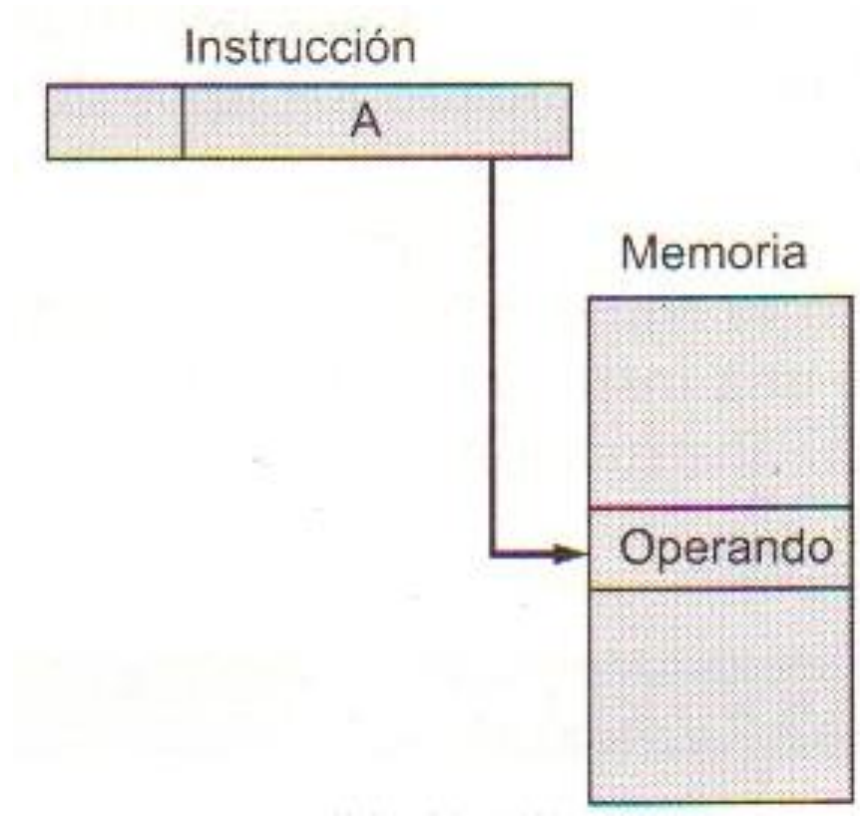
Direccionamiento Directo

En el que el campo de direcciones contiene la dirección efectiva del operando:

$$EA = A$$

Sólo requiere una referencia a memoria, y no necesita ningún cálculo especial. La **limitación** obvia, mencionada con anterioridad, es que proporciona un **espacio de direcciones restringido**.

Direccionamiento Directo

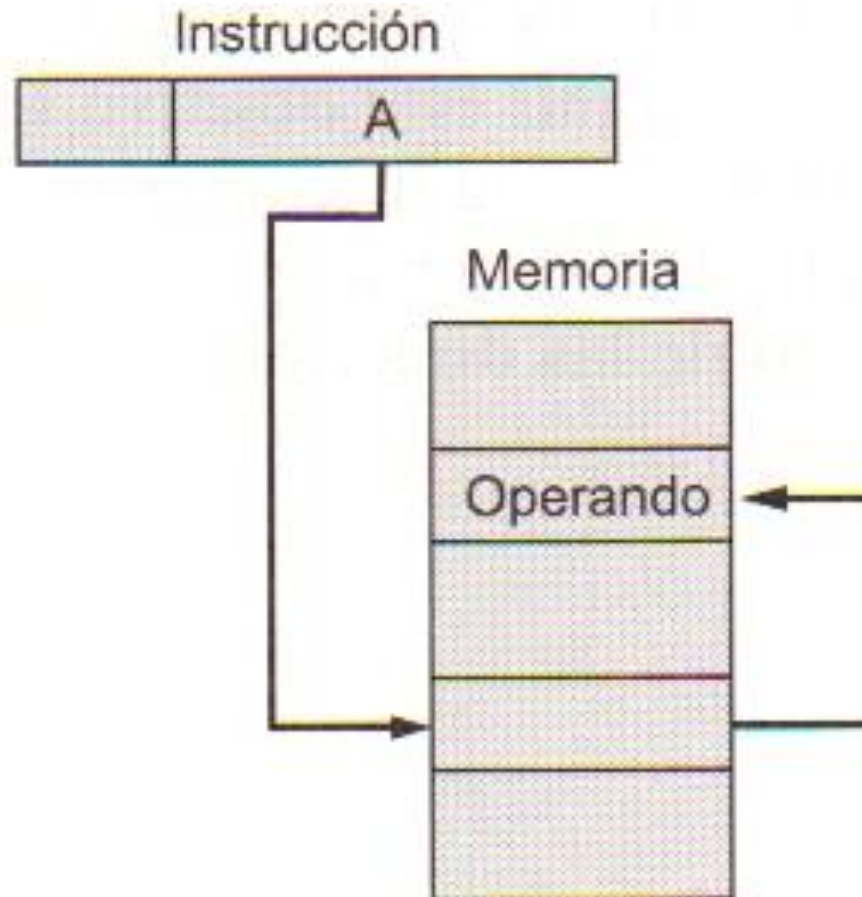


Direcccionamiento Indirecto

El problema del direccionamiento DIRECTO es que la longitud del campo de direcciones es, normalmente, menor que la longitud de la palabra, limitando pues el rango de direcciones. **Una solución** es hacer que el campo de direcciones referencie la dirección de una palabra de memoria que contenga la dirección completa del operando. Esto es lo que se denomina ***direccionamiento indirecto***.

$$EA = (A)$$

Direccionamiento Indirecto



Direccionamiento Indirecto ...

El paréntesis se interpreta como **[contenido de]**.
La **ventaja** obvia de esta aproximación es que para una longitud de palabra de N bits, se dispone ahora de un espacio de direcciones de 2 elevado a la N . La **desventaja** es que la ejecución de la instrucción requiere **dos referencias a memoria** para captar el operando.

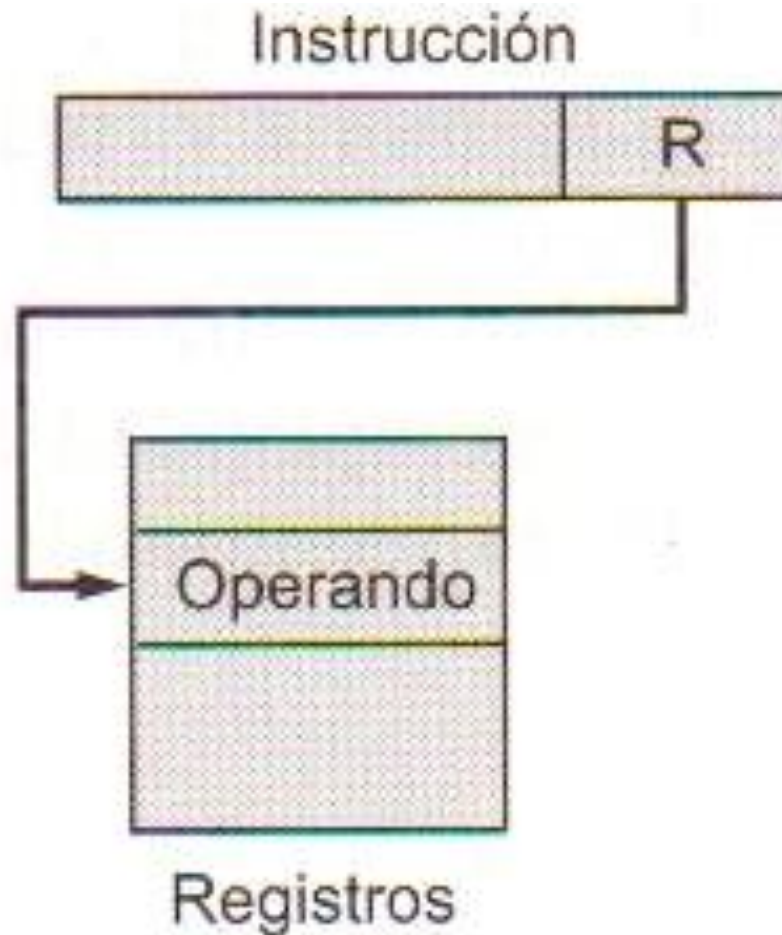
Direccionamiento de Registros

El direccionamiento de registros es **similar al directo**. La única diferencia es que el campo de direcciones referencia un registro, en lugar de una dirección de memoria principal:

$$EA = R$$

Las **ventajas** del direccionamiento de registros son que (1) sólo es necesario un campo pequeño de direcciones en la instrucción, (2) no se requieren referencias a memoria. El tiempo de acceso a un registro interno de la CPU es mucho menor que para la memoria principal. La **desventaja** del direccionamiento a registros es que el espacio de direcciones está muy limitado.

Direccionamiento de Registros

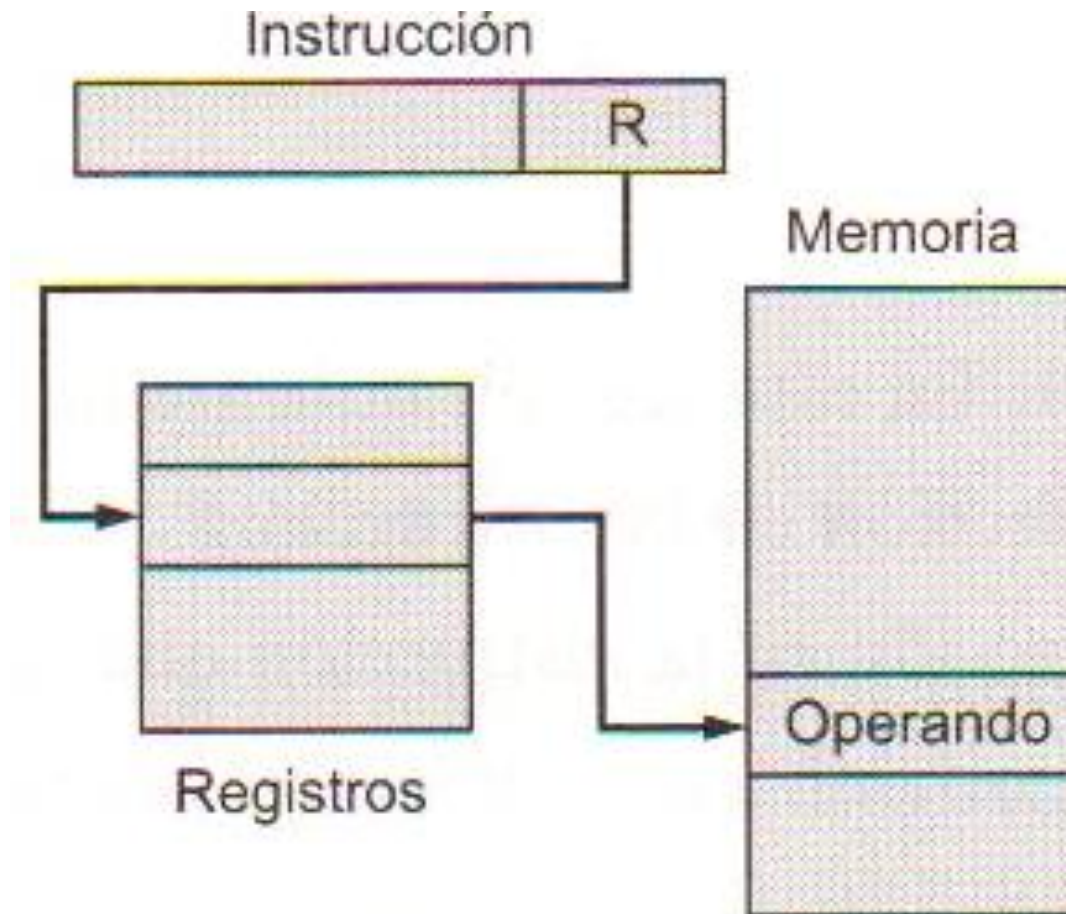


Direccionamiento Indirecto con Registro

$$EA = (R)$$

Las ventajas y limitaciones del direccionamiento indirecto con registro son básicamente las mismas que se tienen para el direccionamiento indirecto. Además del direccionamiento indirecto con registro **emplea una referencia menos a memoria** que el direccionamiento indirecto.

Direccionamiento Indirecto con Registro



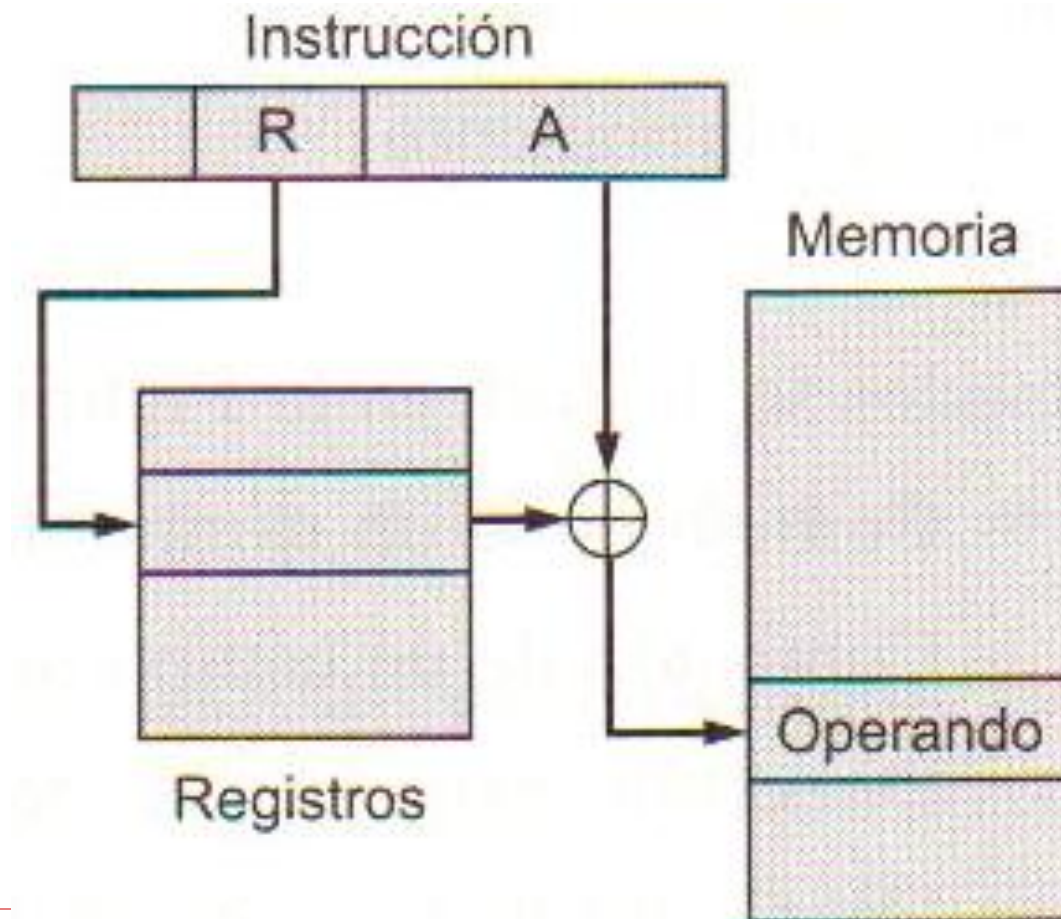
Direccionamiento con Desplazamiento

Combina las posibilidades de los direccionamientos directo, e indirecto con registro.

$$EA = A + (R)$$

Requiere que las instrucciones tengan dos campos de direcciones, al menos uno de ellos explícito. El valor contenido en uno de los campos de direcciones (valor = A) se utiliza directamente; el otro campo de direcciones, o una referencia implícita definida por el código de operación, **se refiere a un registro cuyo contenido se suma a A para generar la dirección efectiva.**

Direccionamiento con Desplazamiento



Direccionamiento con Desplazamiento ...

- Describiremos tres de los usos más comunes del direccionamiento con desplazamiento:
 - Desplazamiento relativo(PC)
 - Direccionamiento con registro base
 - Indexado
-

Direccionamiento con Desplazamiento ...

Direccionamiento relativo

Para el direccionamiento relativo, el registro referenciado implícitamente es el **contador de programa (PC)**. Es decir, la dirección de instrucción actual se suma al campo de direcciones para producir el valor EA. En consecuencia, la dirección efectiva es un desplazamiento relativo a la dirección de la instrucción.

Direccionamiento con Desplazamiento ...

Direccionamiento con registro-base

La interpretación del direccionamiento con registro-base es la siguiente: el registro referenciado contiene una dirección de memoria, y el campo de dirección contiene un desplazamiento desde dicha dirección.

MOV AX, [BX+7]

Direccionamiento con Desplazamiento ...

Indexado

La interpretación usual del indexado es la siguiente: el campo de dirección referencia una dirección de memoria principal, y el registro referenciado contiene un desplazamiento positivo desde esa dirección.

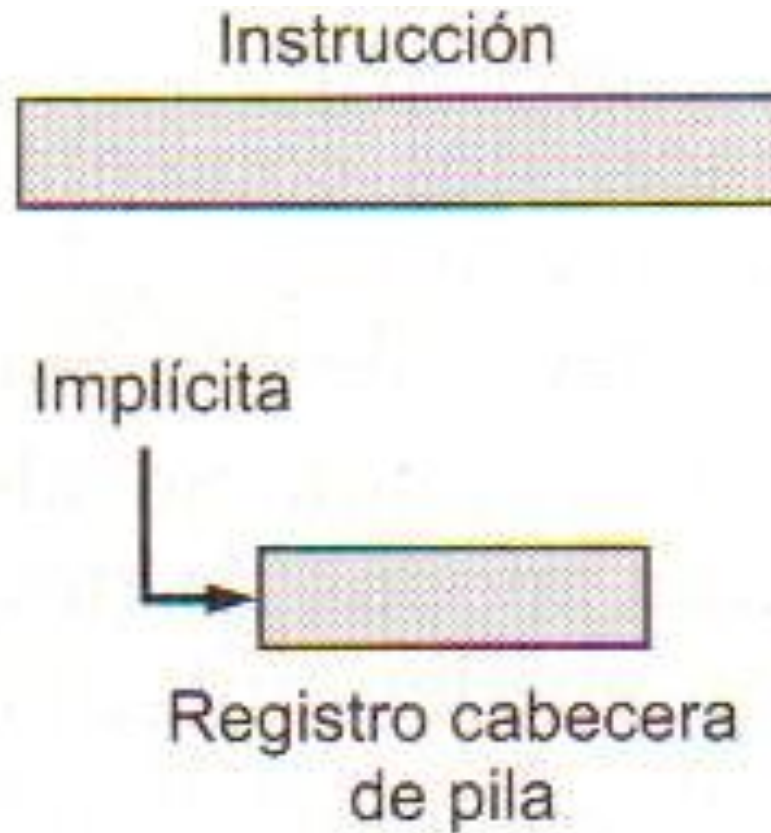
- **MOV AX, [SI+7]**
 - **MOV [SI+1], "B"**
 - **INC SI**
 - **ADD SI, 1**
-

Direccionamiento de Pila

Una pila es una matriz lineal de direcciones. La pila tiene asociado un puntero, cuyo valor es la dirección de la cabecera de la pila. El puntero de pila se mantiene en un registro. Así, las **referencias a posiciones de la pila en memoria son, de hecho, DIRECCIONES DE ACCESO INDIRECTO CON REGISTRO.**

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

Direcccionamiento de Pila



MODOS DE DIRECCIONAMIENTO DEL PENTIUM II

El mecanismo de traducción de direcciones del Pentium II produce una dirección, denominada **<<dirección virtual o efectiva>>, que es un desplazamiento dentro de un segmento.** La suma de la dirección de comienzo del segmento y la dirección efectiva produce una dirección lineal.

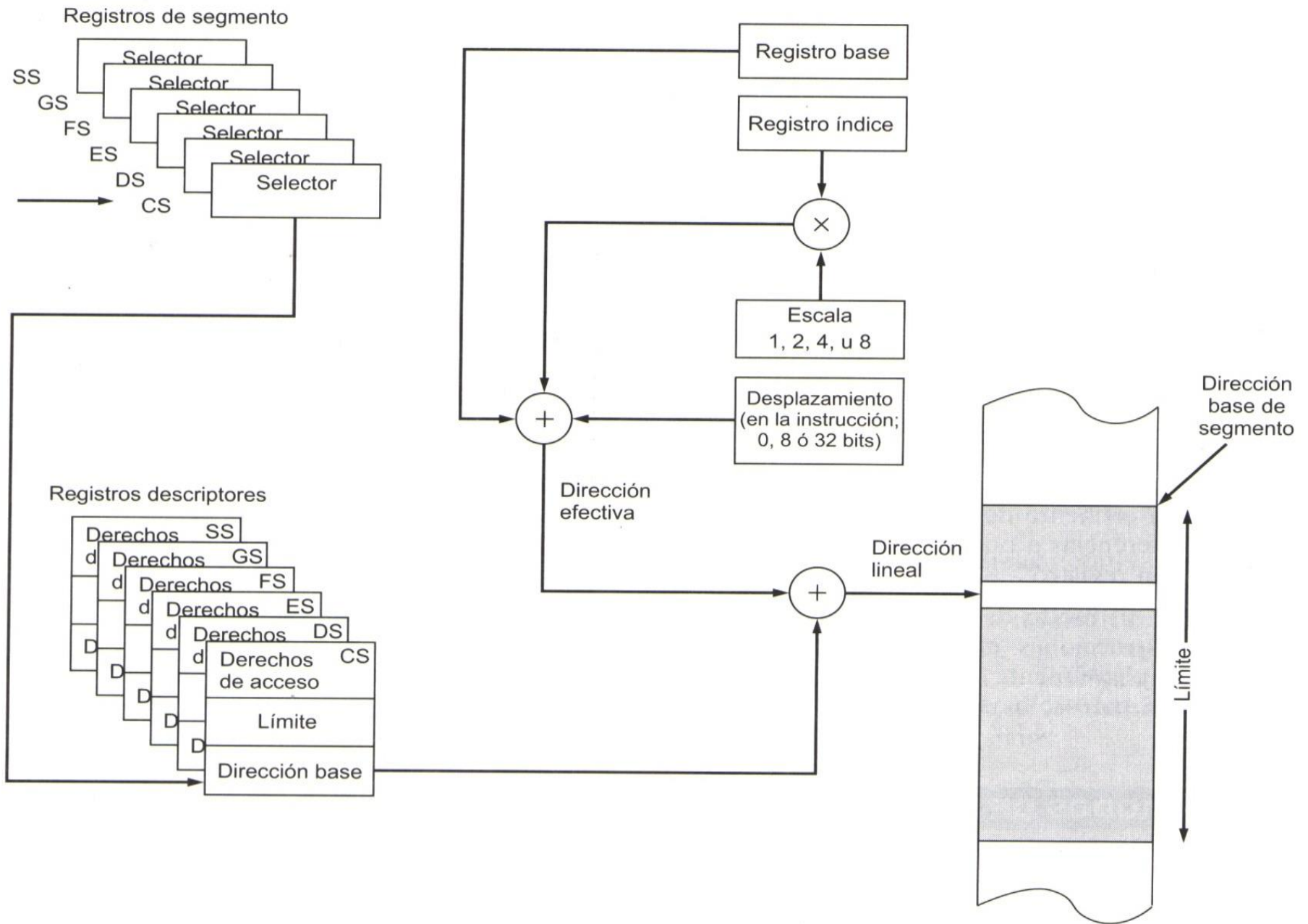


Figura 10.2. Cálculos en el modo de direccionamiento del Pentium II.

MODOS DE DIRECCIONAMIENTO DEL PENTIUM II ...

Cada **registro de segmento** retiene la dirección de comienzo del correspondiente segmento. Asociado con cada registro de segmento hay un registro descriptor de segmentos (no visible al programador), que registra los derechos de acceso para el segmento, así como la dirección de comienzo y el límite (longitud) del segmento.

Tabla 10.2. Modos de direccionamiento del Pentium II

Modo	Algoritmo
Inmediato	Operando = A
Registro	LA = R
Con desplazamiento	LA = (SR) + A
Base	LA = (SR) + (B)
Base con desplazamiento	LA = (SR) + (B) + A
Índice escalado con desplazamiento	LA = (SR) + (I) × S + A
Base con índice y desplazamiento	LA = (SR) + (B) + (I) + A
Base con índice escalado y desplazamiento	LA = (SR) + (I) × S + (B) + A
Relativo	LA = (PC) + A

LA = dirección lineal

(X) = contenido de X

SR = registro de segmento

PC = contador de programa

A = contenido de un campo de dirección de la instrucción

R = registro

B = registro base

I = registro índice

S = factor de escala

MODOS DE DIRECCIONAMIENTO DEL PENTIUM II ...

En el **modo inmediato**, el operando se incluye en la instrucción

En el **modo operando con registro**, el operando está situado en un registro. Para instrucciones de tipo general, tales como instrucciones de transferencia de datos, aritméticas, y lógicas, el operando puede ser uno de los registros generales de 32 bits (EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP), uno de los registros generales de 16 bits (AH, BH, CH, DH, AL, BL, CL, DL). Hay también algunas instrucciones que hacen referencia a los registros de segmento (CS, DS, ES, SS, FS, GS).

En el **modo de desplazamiento**, el desplazamiento de operando (la dirección efectiva en la Figura 10.2) está incluido, formando parte de la instrucción, como desplazamiento de 8, 16 o 32 bits.

MODOS DE DIRECCIONAMIENTO DEL PENTIUM II ...

El **modo base** especifica que uno de los registros de 8, 16 o 32 bits, contiene la dirección efectiva. Esto es equivalente a lo que hemos denominado <<direccionamiento indirecto con registro>>.

En el **modo base con desplazamiento**, la instrucción incluye un desplazamiento que hay que sumar un registro base, que puede ser cualquiera de los registros de uso general.

En el **modo con índice <<escalado>>**, la instrucción incluye un desplazamiento a sumar a un registro, llamado en este caso <<registro índice>>.

MODOS DE DIRECCIONAMIENTO DEL PENTIUM II ...

El **modo base con índice y desplazamiento** suma los contenidos de los registros base e índice, y un desplazamiento, para formar la dirección efectiva.

El **modo base con índice escalado y desplazamiento** suma el contenido de registro índice, multiplicado por un factor de escala, con el contenido del registro base y el desplazamiento.

Finalmente, el **direccionamiento relativo** puede emplearse en instrucciones de transferencia del control (control de flujo). Se suma un desplazamiento al valor del contador de programa, que apunta a la instrucción siguiente.

FORMATOS DE INSTRUCCIONES

Un formato de instrucciones **define la descripción en bits de una instrucción en términos de las distintas partes que la componen.** Un formato de instrucciones debe incluir un código de operación (**codop**) e, implícita o explícitamente, ninguno o algunos operandos.

Cada operando explícito se referencia utilizando uno de los modos de direccionamiento descritos.

En la mayoría de los repertorios de instrucciones se emplea más de un formato de instrucción.

Longitud de Instrucción

El aspecto de diseño más básico a considerar en el formato es la longitud de la instrucción. **Esta decisión afecta, y se ve afectada, por el tamaño de la memoria, su organización, la estructura de buses, la complejidad de la CPU, y la velocidad de la CPU.**

El programador desea más codops, más operandos, más modos de direccionamiento y mayor rango de direcciones. Más codops y más operandos facilitan el trabajo del programador, ya que puede redactar programas más cortos para resolver las mismas tareas.

Longitud de Instrucción

Debiera cumplirse o que el **tamaño de la instrucción fuera igual al tamaño de las transferencias a memoria** o que uno fuera un múltiplo del otro, pero sin embargo importante, es que la longitud de la instrucción debiera ser **múltiplo de la longitud de un carácter**, que normalmente es 8 bits. Ya que una forma común de datos es el carácter, sería deseable que una palabra almacenara un número entero de caracteres. Si no, se perderían bits en cada palabra cuando se almacenan múltiples caracteres, o habría algunos caracteres partidos entre dos palabras.

Asignación de los Bits

Para una longitud de instrucción dada existe claramente un compromiso entre el número de codops y la capacidad de direccionamiento. Los siguientes factores, relacionados entre sí, afectan a la definición del uso dado a los bits de direccionamiento.

- **Número de modos de direccionamiento**
 - **Número de operandos**
 - **Registros frente a memoria**
 - **Número de conjuntos de registros:** La tendencia actual ha sido pasar de un solo banco de registros de uso general a un grupo de dos o más conjuntos especializados (por ejemplo, para datos y para desplazamientos)
 - **Rango de direcciones:** Relacionado con el número de bits de direccionamiento
-

PDP-8

Utiliza **instrucciones de 12 bits** y opera con palabras de 12 bits. Hay **un solo registro de uso general**, el acumulador.

La Figura 10.4 muestra el formato de instrucción del PDP-8. Hay un **codop de 3 bits y 3 tipos de instrucciones**. Para los codops 0 a 5, el formato consiste en una instrucción con una sola referencia a memoria, incluyendo un bit de página y un bit de indirección. Así pues, **hay sólo 6 operaciones básicas**. Para ampliar el grupo de operaciones, **el codop 7 define una referencia a registro o *microinstrucción***. El codop 6 es la operación de E/S; se emplean 6 bits para seleccionar uno de entre 64 dispositivos, y 3 bits especifican una orden particular de E/S.

El formato de instrucción del PDP-8 es bastante eficiente. Permite direccionamiento indirecto, direccionamiento con desplazamiento e indexado.

Instrucciones con referencia a memoria

Codop	D/I	Z/C	Desplazamiento
0	2	3	4 5 11

Instrucciones de entrada/salida

1 1 0	Dispositivo	Codop
0	2 3	8 9 11

Instrucciones con referencia a registro

Microinstrucciones del grupo 1

1 1 1 0	CLA	CLL	CMA	CML	RAR	RAL	BSW	IAC
0 1 2 3	4	5	6	7	8	9	10	11

Microinstrucciones del grupo 2

1 1 1 1	CLA	SMA	SZA	SNL	RSS	OSR	HLT	0
0 1 2 3	4	5	6	7	8	9	10	11

Microinstrucciones del grupo 3

1 1 1 1	CLA	MQA	0	MQL	0	0	0	1
0 1 2 3	4	5	6	7	8	9	10	11

Nemotécnicos

CLA = Acumulador a cero

CLL = Enlace a cero

CMA = Complementar acumulador

CML = Complementar enlace

RAR = Rotar a derecha acumulador

RAL = Rotar a izquierda acumulador

BSW = Intercambiar Byte

IAC = Incrementar acumulador

SMA = Salto si acumulador negativo

SZA = Salto si acumulador cero

SNL = Salto si enlace distinto de cero

RSS = Invertir sentido del salto

OSR = OR con registro conmutador

HLT = Parar

MQA = Multiplicador / cociente en el acumulador

MQL = Cargar multiplicador / cociente

Figura 10.4. Formatos de instrucciones del PDP-8.



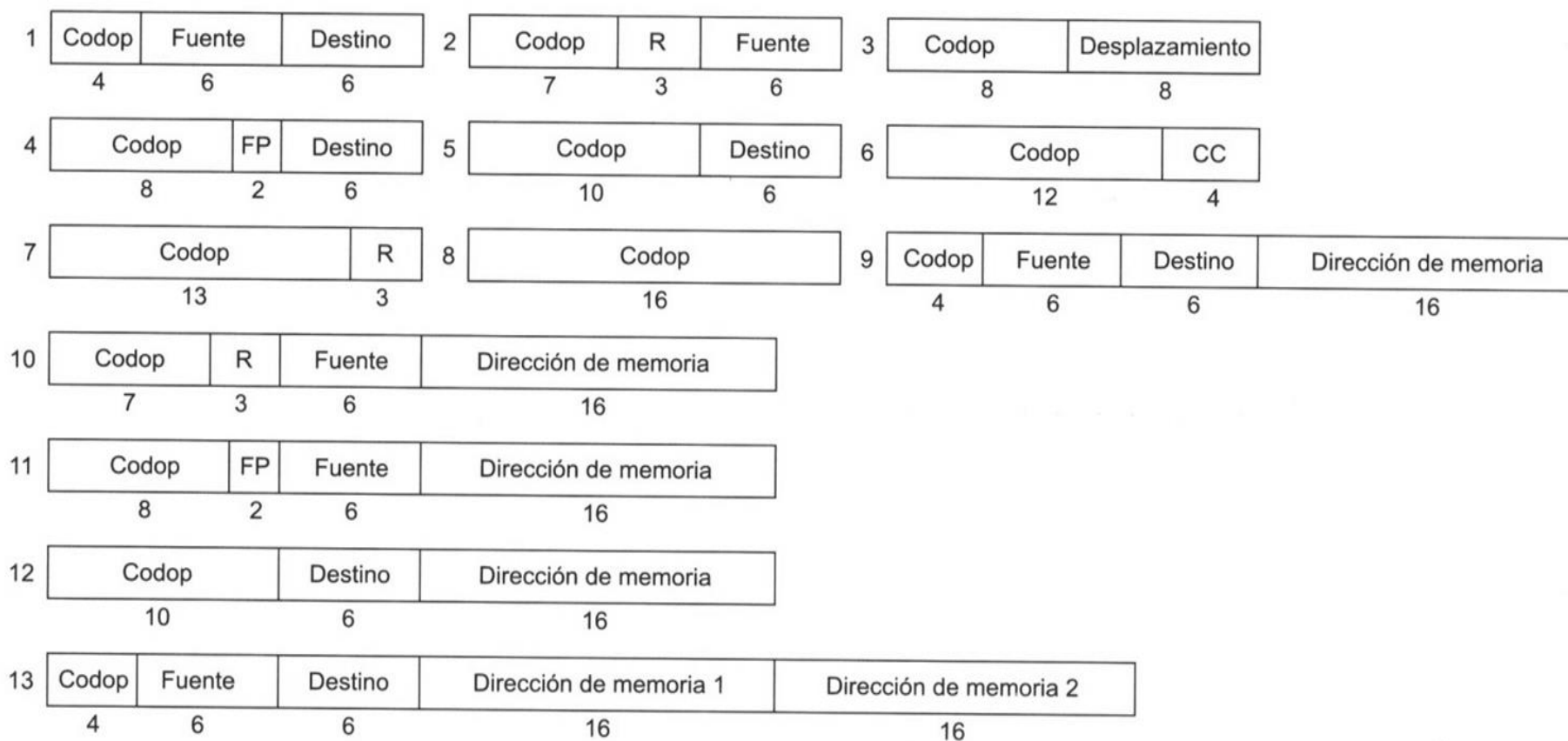
Instrucciones de longitud Variable

Los ejemplos vistos hasta ahora hacen uso de una única longitud de instrucción fija e, implícitamente, hemos discutido decisiones adoptadas en ese contexto. No obstante, los diseñadores pueden optar por utilizar varios formatos de instrucción de longitudes diferentes. **Esta táctica hace fácil proporcionar un amplio repertorio de codops de longitud variable. El direccionamiento puede ser más flexible, con varias combinaciones de referencias a registros y a memoria, así como de modos de direccionamiento.** El precio a pagar por las instrucciones de longitud variable es el aumento de complejidad de la CPU.

PDP-11

El PDP-11 emplea un conjunto de **8 registros de uso general de 16 bits**. Dos de estos registros tienen una función adicional: uno se emplea como puntero de pila en operaciones específicas con la pila, y el otro se emplea como contador de programa.

Se **usan 13 formatos, compaginando instrucciones de ninguna, una, y dos direcciones**. La longitud del codop puede variar de 4 a 16 bits. Para las referencias a registros se emplean 6 bits. Tres bits identifican el registro, y los otros tres restantes indican el modo de direccionamiento.



Fuente y destino contienen un campo de modo de direccionamiento de 3 bits y un número de registro de 3 bits.

FP es uno de los registros de punto flotante 0, 1, 2 o 3.

R es uno de los registros de uso general.

CC es el campo de códigos de condición.

Figura 10.6. Formatos de instrucciones utilizados en el PDP-11; los números indican longitudes de campos.



FORMATOS DE INSTRUCCIÓN DEL PENTIUM II

El Pentium II está equipado con varios formatos de instrucciones. **De los elementos descritos anteriormente, sólo el campo codop está siempre presente.** La Figura 10.8 ilustra el formato de instrucción general. Las instrucciones se componen de entre cero y cuatro prefijos de instrucción general. Las instrucciones se componen de entre cero y cuatro prefijos de instrucción opcionales, un codop de uno o dos bytes, una especificación de dirección opcional que consta del byte <<Mod r/m>> y del byte de índice de escala (<<Scale Index>>), un desplazamiento opcional, y un campo inmediato opcional.

FORMATOS DE INSTRUCCIÓN DEL PENTIUM II

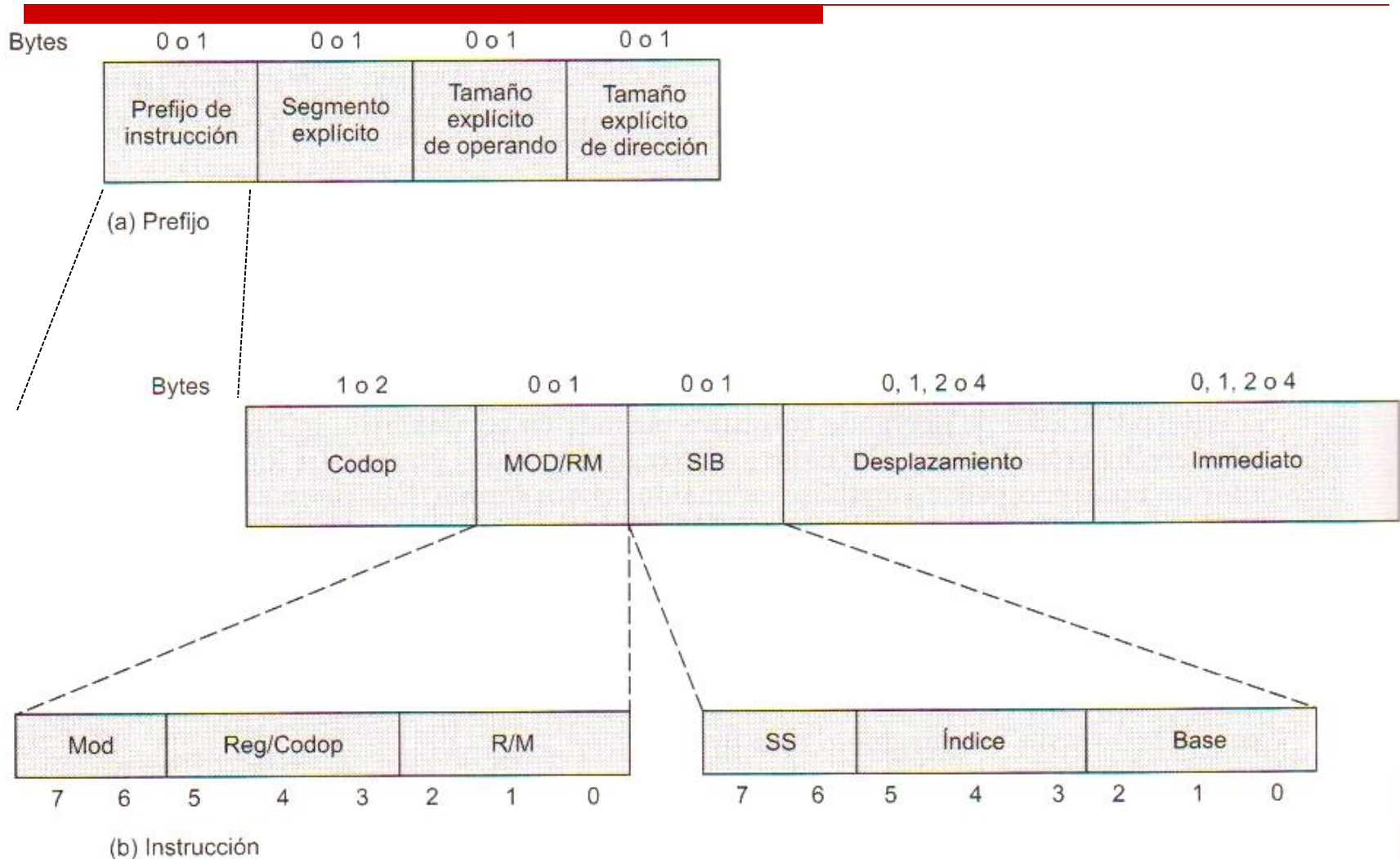


Figura 10.8. Formato de instrucción del Pentium II.

FORMATOS DE INSTRUCCIÓN DEL PENTIUM II

La propia instrucción incluye los siguientes campos:

Codop: Codop de uno o dos bytes. El codop puede incluir también bits que indican si el operando es de 1 byte o de un tamaño completo (16 o 32 bits dependiendo del contexto), la dirección de la operación con los datos (a, o desde, memoria), y si el campo de dato inmediato debe o no extenderse con el signo.

Mod r/m: Este byte y el siguiente aportan información de direccionamiento. **El byte <<Mod r/m>> indica si un operando está en un registro o en memoria.** El byte <<Mod r/m>> consta de tres campos. El campo Mod (2 bits) se combina con el campo r/m para formar 32 valores posibles: 8 a registro y 24 modos de indexado; el campo Reg/Codop (3 bits) especifica o bien un número de registro, o bien tres bits más de información del codop; el campo r/m (3 bits) puede especificar un registro como posición de un operando o puede formar parte de la codificación del modo de direccionamiento, en combinación con el campo Mod.

FORMATOS DE INSTRUCCIÓN DEL PENTIUM II

SIB: Cierta codificación del byte <<Mod r/m>> indica la inclusión del byte SIB para especificar por completo el modo de direccionamiento. El byte SIB consta de tres campos. El campo SS (2 bits) especifica el factor de escala para el indexado escalado; el campo de índice (3 bits) especifica el registro índice; el campo de base (3 bits) especifica el registro base

Desplazamiento: Cuando el indicador de modo de direccionamiento especifica que se emplea desplazamiento

Inmediato: Proporciona el valor de un operando de 8, 16 o 32 bits.

FORMATOS DE INSTRUCCIÓN DEL PENTIUM II

Ya que solo un operando puede tener información de modo de dirección, en una instrucción solo puede referenciarse un operando de memoria. En contraste, el VAX lleva la información de modo de dirección con cada operando, permitiendo operaciones de memoria a memoria.

Como puede verse, la codificación del conjunto de instrucciones del Pentium II es muy compleja. Esto se debe, en parte a la necesidad de ser compatible con el 8086, y en parte al deseo de algunos diseñadores de suministrar todas las ayudas posibles al diseñador del compilador para que produzca código eficiente.
