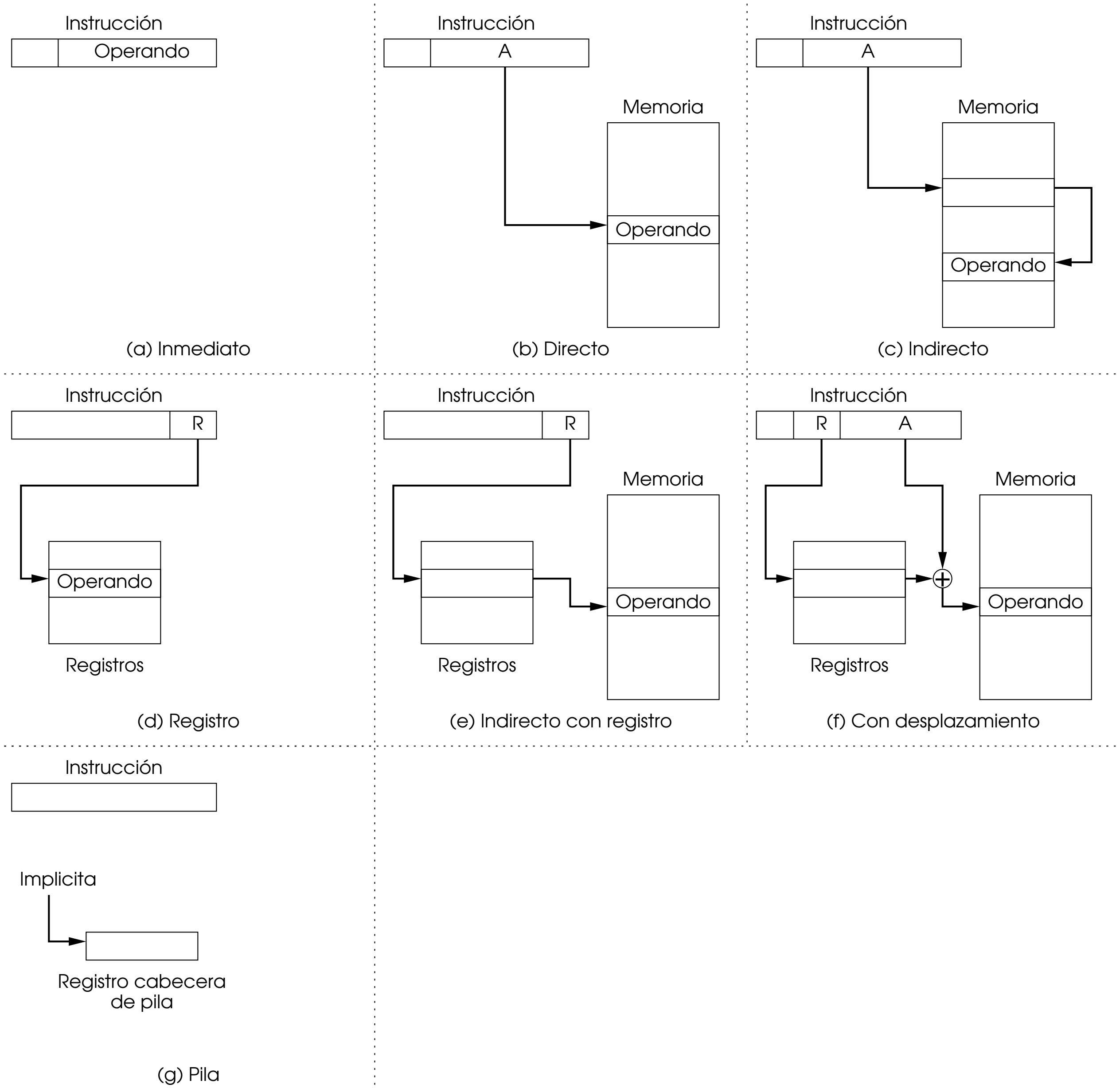
Introducción

En la siguiente presentación se ve la forma de *cómo* especificar los operandos o dar la dirección de un operando y las operaciones de las instrucciones, es decir, la forma cómo se organizan los bits de una instrucción para definir las direcciones de los operandos y la operación que realiza dicha instrucción.

Direccionamiento

El campo o campos de dirección en un formato de instrucción está limitado. Se desea direccionar un rango grande de posiciones de memoria principal o memoria virtual. Las técnicas de direccionamiento implican un compromiso entre el rango de direcciones y/o flexibilidad de direccionamiento y por otra, el número de referencias a memoria y/o la complejidad de cálculo de las direcciones. Las más comunes son:

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


Instrucción
Operando
(a) Inmediato
Instrucción
A
Memoria
Operando
(b) Directo
Instrucción
A
Memoria
Operando
(c) Indirecto
Instrucción
R
Operando
Registros
(d) Registro
Instrucción
R
Memoria
Registros
Operando
(e) Indirecto con registro
Instrucción
R
A
Memoria
Registros
Operando
(f) Con desplazamiento
Instrucción
Implicita
Registro cabecera de pila
(g) Pila

Cálculos hechos para cada modo de direccionamiento

Modo	*Algoritmo*	*Ventaja Principal*	*Desventaja Principal*
Inmediato	Operando $= A$	No referencia a memoria	Operando de magnitud limitada
Directo	$EA = A$	Sencillo	Espacio de direcciones limitado
Indirecto	$EA = (A)$	Espacio de direcciones grande	Referencias a memoria múltiples
Registro	$EA = R$	No referencia a memoria	Espacio de direcciones limitado
Indirecto con registro	$EA = (R)$	Espacio de direcciones grandes	Referencia extra a memoria
Con desplazamiento	$EA = A + (R)$	Flexibilidad	Complejidad
Pila	$EA =$ cabezera	No referencia a memoria	Aplicabilidad limitada

A contenido del campo de direcciones de la instrucción

EA dirección real que contiene el operando

(X) contenido de la posición X

Como las arquitecturas de las computadoras ofrecen varias modos de direccionamiento, la forma como determina la unidad de control que modo de direccionamiento se está empleando, puede ser usando diferentes codops para cada modo, o usando uno o más bits del formato de instrucción como *campo de modo*.

La dirección efectiva (EA “effective address”) en un sistema sin memoria virtual es una dirección de memoria principal o un registro. En un sistema con memoria virtual es un dirección virtual o un registro. La correspondencia real con una dirección física dependerá del mecanismo de paginación y no es visible al programador.

Direccionamiento Inmediato

Es la forma más sencilla y el operando está presente en la propia instrucción:

$$\text{OPERANDO} = A$$

Se usa para definir y utilizar constantes o fijar valores iniciales de variables. El número se almacena en complemento a dos.

La ventaja es que no se requiere una referencia a memoria para obtener el operando, la desventaja es el tamaño del número que está restringido a la longitud del campo de direcciones.

Direccionamiento Directo

El campo de direcciones contiene la dirección efectiva del operando

$$EA = A$$

Fue común en las primeras generaciones de computadoras y es usado aún en un número pequeño de sistemas. La desventaja es que se tiene un espacio de direcciones reducido, ya que la longitud del campo de direcciones es por lo general menor que la longitud de palabra.

Direccionamiento indirecto

Se hace que el campo de direcciones referencie la dirección de una palabra de memoria, la cual tiene la dirección completa del operando.

$$EA = (A)$$

El paréntesis indica el "contenido de". La ventaja es que para una longitud de palabra de N bits se tiene un espacio de direcciones de 2^N. La desventaja es que requiere dos referencias a memoria (una para captar la dirección y otra para obtener el valor).

Aunque el número de palabras que pueden direccionarse es 2^N, el número de direcciones efectivas diferentes está limitado a 2^K, donde K es la longitud del campo de direcciones.

Una variante es el direccionamiento multinivel o en cascada
$EA = (\ldots(A)\ldots)$

Direccionamiento de registros

Parecido al directo, en donde el campo de direcciones referencia un registro.

$$EA = R$$

Como un campo de direcciones que referencia a registros tiene 3 ó 4 bits, se pueden usar de 8 a 16 registros de uso general.

Las ventajas son: sólo es necesario un campo pequeño de direcciones en la instrucción y no se requieren referencias a memoria. Además el tiempo de acceso es menor que a la memoria principal.

Para conseguir un uso óptimo el operando de registro se debe mantener en uso para varias operaciones, por ejemplo, en los resultados intermedios de un cálculo.

Direccionamiento Indirecto con Registro

Es análogo al direccionamiento indirecto.

$$EA = (R)$$

Las ventajas y limitaciones son básicamente las mismas que las del direccionamiento indirecto. La ventaja del direccionamiento indirecto con registro es que se emplea una referencia menos a memoria.

Direccionamiento con desplazamiento

Es una forma potente de direccionamiento que combina las posibilidades de los direccionamientos directo e indirecto con registro. Se conoce con varios nombres dependiendo del contexto, pero el mecanismo básico es mismo. Se usará la denominación de *direccionamiento con desplazamiento*.

$$EA = A + (R)$$

Se requiere que las instrucciones tengan dos campos de direcciones, al menos uno de los cuales explícito.

Los uso más comunes del direccionamiento con desplazamiento:

- Desplazamiento relativo
- Direccionamiento con registro base
- Indexado

Direccionamiento relativo

El registro referenciado implícitamente es el contador de programa (PC). La dirección de instrucción actual se suma al campo de direcciones para producir el valor EA. El campo de direcciones se trata como un número en complemento a dos, por lo que se da un desplazamiento relativo a la dirección de la instrucción. Se pueden ahorrar bits de direcciones por el uso de localidad.

Direccionamiento con registro base

El registro referenciado contiene una dirección de memoria, y el campo de dirección contiene un desplazamiento (representación entera sin signo) desde dicha dirección.

Hace uso de la localidad para las referencias a memoria. Ayuda a implementar la segmentación. En algunas implementaciones se emplea un solo registro de base de segmento que es usado implícitamente, y en otras, el programador puede seleccionar un registro.

Direccionamiento con desplazamiento – Indexado

El campo de dirección referencia una dirección de memoria principal, y el registro referenciado contiene un desplazamiento positivo desde esa dirección.

Es un mecanismo eficiente para ejecutar operaciones iterativas. Los registros índice se usan normalmente para tareas iterativas, en donde se incrementan o decrementan tras cada referencia, lo cual algunos sistemas lo hacen automáticamente como parte del ciclo de instrucción, que se conoce como *autoindexado*

Algunas máquinas disponen de direccionamiento tanto indirecto como indexado, y es posible emplear ambos en la misma instrucción.

Si el indexado se realiza después de la "indirección" se denomina *postindexado*

$$EA = (A) + (R)$$

El contenido del campo de direcciones se usa para acceder la posición de memoria que contiene una dirección directa, la cual se indexa mediante el valor del registro. Es útil para acceder a uno entre un número de bloque de datos con un formato fijo.

Con el *preindexado*, el indexado se realiza antes de la “indirección”

$$EA = (A + (R))$$

La dirección se calcula como el indexamdo simple, donde la dirección calculada no contiene el operando, sino la dirección del operando. Se usa esta técnica para construir tablas de bifurcación multirama. Puede establecerse una tabla de direcciones que comience en la posición A, y mediante indexado en esta tabla, encontrar la posición requerida.

Direccionamiento de pila

Es una matriz lineal de posiciones, se le conoce también como *lista de apilamiento* o *cola*. La pila tiene asociado un puntero cuyo valor es la dirección de la cabecera o tope de la pila, el cual se mantiene en un registro, con lo que las referencia a posiciones de la pila en memoria son direcciones de acceso indirecto con registro. Las instrucciones máquina no necesitan incluir una referencia a memoria sino que operan implícitamente con la cabecera.

Modos de direccionamiento del Pentium

Modo	*Algoritmo*
Inmediato	Operando $= A$
Registro	$LA = R$
Con desplazamiento	$LA = (SR) + A$
Base	$LA = (SR) + (B)$
Base con desplazamiento	$LA = (SR) + (B) + A$
Indice escalado con desplazamiento	$LA = (SR) + (I) \times S + A$
Base con índice y desplazamiento	$LA = (SR) + (B) + (I) + A$
Base con índice escalado y desplazamiento	$LA = (SR) + (I) \times S + (B) + A$
Relativo	$LA = (PC) + A$

$LA =$ dirección lineal $\quad (X) =$ contenido de X

$SR =$ registro de segmento $\quad PC =$ contador de programa

$A =$ contenido de un campo de dirección de la instrucción

$R =$ registro $\quad B =$ registro base

$I =$ registro índice $\quad S =$ factor de escala

Modos de Direccionamiento del Pentium

Modo inmediato. El operando se incluye en la instrucción, pudiendo ser un byte, una plabra, o una palabra doble de datos.

Modo de operando en registro. El operando está situado en un registro. En instrucciones generales (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), de 16 bits (AX, BX, CX, DX, SI, DI, SP, BP), o uno de los registros generales de 8 bits (AH, BH, CH, DH, AL, BL, CL, DL). Operaciones en punto flotante, los operandos de 64 bits se hacen con 2 registros de 32 bits.

Modo de desplazamiento. Es útil para referenciar variables globales. El desplazamiento del operando está incluido, formando parte de la instrucción, como desplazamiento de 8, 16 ó 32 bits.

Modo base. Especifica que uno de los registros de 8, 16 ó 32 bits, contiene la dirección efectiva, que es equivalente al direccionamiento indirecto con registro.

Modo base con desplazamiento La instrucción incluye un desplazamiento que hay que sumar a un registro de uso general, conocido como base. Ejemplos de uso son:

1. Utilización por un compilador para apuntar al comienzo de la zona de varialbes locales.

2. Utilización para indexar en una matriz cuando el tamaño de elemento no es 2, 4 ó 8 bytes, ya que no se puede indexar con un registro índice.

3. Utilización para acceder a un campo de un registro.

Modo desplazamiento con índice “escalado”. La instrucción incluye un desplazamiento a sumar a un registro, llamado registro índice. El registro puede ser cualquiera de los registros de uso general excepto el ESP. El calculo de la dirección efectiva es igual al contenido del registro índice multiplicado por un factor de escala (1, 2, 4 ó 8), más un desplazamiento. Se usa para indexar matrices.

Modo base con índice y desplazamiento Se suman los contenidos de los registros base, índice y un despalzamiento para formar la dirección efectiva. Puede emplearse para acceder a una matriz local en un marco de pila, o par manejar una matriz bidimensional donde el despalzamiento apunta al inicio de la matriz, y cada registro gestiona una dimensión.

Modo base con índice escalado y desplazamiento Se suma el contenido de registro índice multiplicado por un factor de escala, con el contenido del registro base, y el desplazamiento. Se usa cuando una matriz esta alamacenada en un marco de pila, donde los elementos de la matriz tienen una longitud de 2, 4 u 8 bytes.

Direccionamiento relativo Se emplea en instrucciones de transferencia del control. Se suma un desplazamiento al valor del contador de programa, que apunta a la instrucción siguiente. El desplazamiento se trata como un byte, una palabra, o una palabra dble numérica con signo, cuyo valor incrementa o decrementa al PC.

Formato de Instrucciones

El formato de instrucciones define la descripción en bits de una instrucción. El formato incluye un código de operación *codop* e, implícita o explícitamente, ningún o algunos operandos. El formato debe indicar el modo de direccionamiento para cada operando. En la mayoría de los repertorios de instrucciones se emplea más de un formato.

Longitud de instrucción Es el aspecto de diseño mas básico a considerar en el formato, ya que afecta y se ve afectada por, el tamaño de la memoria, su organización, la estructura de buses, complejidad del CPU, y velocidad del CPU. Define la riqueza y flexibilidad de la máquina desde el punto de vista del programdor en lenguaje ensamblador.

El compromiso básico es disponer de un repertorio de instrucciones máquina potente y la necesidad de ahorrar espacio. Existen otras consideraciones que se deben tenerse en cuenta, por ejemplo, se debe cumplir, que el tamaño de la instrucción sea igual al tamaño de las transferencias a memoria (en un sitema con ancho del bus de direcciones y datos de igual tamaño), o que uno fuera múltiplo del otro. En caso contrario, no se tendrá un número entero de instrucciones durante un ciclo de captación. Otra consideración es la velocidad de transferencia de la memoria, que no ha crecido a la par de la velocidad de los procesadores, pero se puede resolver con el usa de un cache.

Asignación de los Bits

Un aspecto complejo es como asignar los bits en un formato de instrucción. Hay un compromiso entre el número de codops y la capacidad de direccionamiento. Donde un número mayor de codops usa más bits, reduciéndose los bits para direccionamiento. Otra aproximación, es usar codops de longitud variable, donde hay una longitud mínima.

Los factores que influyen en la definición del uso dado a los bits de direccionamiento son:

- *Número de modos de direccionamiento*
- *Número de operandos*
- *Registros frente a memoria*
- *Número de conjuntos de registros*
- *Rango de direcciones*
- *Granularidad de las direcciones*

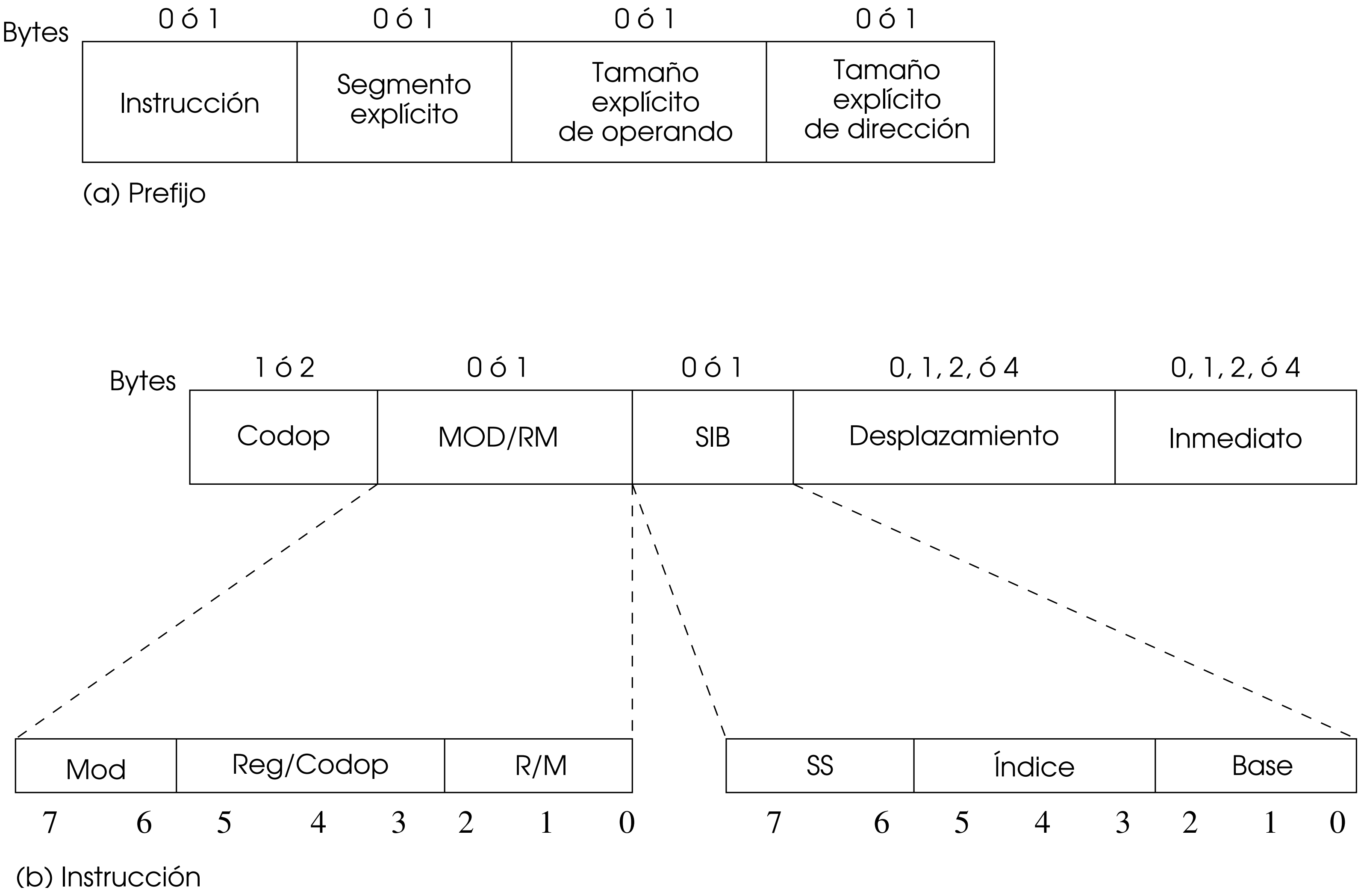
Instrucciones de Longitud Variable

Si se usan varios formatos de instrucción de longitudes diferentes, se puede tener un amplio repertorio de codops de longitud variable. También se logra un direccionamiento más flexible, con varias combinaciones de referencias a registros y a memoria, y modos de direccionamiento, las cuales se dan en una forma eficiente y compacta.

El costo que se tiene por usar instrucciones de longitud variable es el aumento de complejidad del CPU, el cual es un costo pequeño, debido a la reducción del costo del hardware, el uso de microprogramación y más conocimientos de diseño.

Como el CPU no conoce la longitud de la instrucción que va a captar, una estrategia usal es captar un número de bytes o de palabras igual o al menos, al tamaño de la instrucción más larga.

Formato de Instrucciones del Pentium



Formatos de Instrucciones del Pentium

Está equipado con varios formatos. Las instrucciones se componen de:

- *Prefijos* Entre cero y cuatro prefijos.
  - *De Instrucción*. Consta del prefijo LOCK usado para asegurar el uso exclusivo de la memoria compartida en entornos multiprocesador o de prefijos de repetición para una operacion en una cadena que son: REP, REPE, REPZ, REPNE, y REPNZ. Para el primero (absoluto) en el registro CX se indica el número de repeticiones, y los otros (condicionales) se repite la instrucción hasta que se cumpla la condición.
  - *Segmento Explícito*. Indica el registro de segmento que deberá utilizar la instrucción.
  - *Tamaño de la Dirección*. Indica si la dirección es de 16 ó 32 bits.

  - *Tamaño de operando*. Indica el tamaño del operando 16 ó 32 bits.

- *Instrucción*
  - *codop*. De 1 o 2 bytes. Puede incluir bits que indiquen si el operando es de 8 (byte), 16 ó 32 (tamaño completo) bits; la dirección de la operación (a o desde memoria), y si se debe expandir el campo de dato con el signo.
  - *Mod r/m*. Indica si el operando está en un registro o memoria. Si esta en memoria los campos indican el modo de direccionamiento a utilizar
  - *SIB*. Dependiendo del códgio del byte Mod r/m puede indicar la inclusión del byte SIB para indicar por completo el modo de direccionamiento (factor de escala, registro índice, registro base)
  - *Desplazamiento*. Si el indicador de modo de direcciona-

miento indica desplazamiento, se suma un campo con signo de 8, 16 ó 32 bits.

– *Inmediato*. Da el valor de un operando de 8, 16 ó 32 bits.