Repertorios de Instrucciones: Modos de Direccionamiento y Formatos

Daniel Araya Román

La referencia a un operando en una instrucción contiene o bien su valor (inmediato) o una referencia a la dirección del operando. Diversos repertorios de instrucciones utilizan diferentes modos de direccionamiento, como lo son, direccionamiento directo, indirecto, a registro, indirecto con registro, etc.

En el capítulo anterior, se basa en *qué* hace una instrucción, en este capítulo se aborda en *cómo* especificar los operandos y las operaciones de las instrucciones. Teniendo en cuenta, como **especificar la dirección de un operando** y **como se organizan los bits de una instrucción para definir las direcciones de los operandos y la operación que realiza la instrucción.** [1]

Modos de Direccionamiento

Un concepto importante es el **campo de direcciones**, el cual en un formato de instrucción, puede estar bastante limitado. Sería conveniente poder referenciar un rango elevado de posiciones de memoria principal o virtual. Para este motivo, se utlizan los **modos de direccionamiento**. Implicando compromiso entre el rango de direcciones o la flexibilidad de direccionamiento. Como también, el número de referencias a memoria y la complejidad de cálculo de las direcciones.

Direccionamiento Inmediato

Es la forma más sencilla de direccionamiento, en el que el operando está presente en la propia instrucción.

```
Operand = A
mov ax, 05H
```

Se puede utilizar para definir constantes, o fijar valores iniciales de variables. Naturalmente, el número se almacena en complemento a dos. La **ventaja** es que una vez captada la instrucción, no requiere de una referencia a memoria paraobtener el operando, ahorrando un ciclo de memoria o caché, en el ciclo de instrucción. La **desventaja** es que el tamaño del número está restringido a la longitud del **campo de direcciones**. Este en la mayoría de los repertorios de instrucciones es pequeño comparado a una longitud de palabra (16 bits).

Direccionamiento Directo

Otra forma sencilla de direccionamiento, en el que el campo de direcciones contiene la dirección efectiva del operando.

```
EfectiveAddress = A
message db 'Hola, mundo!', '$'
mov dx, offset message
```

La **ventaja** es que solamente requiere una referencia a memoria, la **desventaja** es que proporciona un espacio de direcciones limitado.

Direccionamiento Indirecto

El problema del direccionamiento directo es que la longitud del campo de direcciones es normalmente menor que la longitud de palabra, limitando el rango de direcciones. El direccionamiento indirecto permite que el campo de direcciones referencie la dirección de una palabra de memoria. La cual contiene la dirección completa del operando.

```
EfectiveAddress = [A]
message db 'Hola, mundo!', '$'
mov ax, message
mov dx, [ax]
```

La **ventaja** obvia es que para una longitud de palabra de n *bits*, se dispone ahora de un espacio de direcciones de 2^n. La **desventaja** es que la ejecución de la instrucción requiere de dos referencias a memoria, una para obtener la dirección del operando y otra para obtener el operando en sí. Una variante del direccionamiento indirecto poco común es el direccionamiento indirecto multinivel, o en cascada.

```
EfectiveAddress = [...[A]...]
message db 'Hola, mundo!', '$'
mov ax, message
mov bx, [ax]
mov dx, [bx]
```

No ofrece ninguna ventaja particular, y la desventaja es que requiere de dos o más referencias a memoria para obtener el operando.

Direccionamiento a Registro

Es análogo al direccionamiento directo, la diferencia es que el campo de direcciones referencia un registro en lugar de una dirección de memoria principal.

```
EfectiveAddress = R
.model small
.data
    value1 dw 10
    value2 dw 20
    result dw ?

.code
main proc
    mov ax, value1
    add ax, value2
    mov result, ax

    mov ah, 4Ch
    int 21h
main endp
end main
```

Las **ventajas** son que solo es necesario un campo pequeño de direcciones en la instrucción, y que no se requieren referencias a memoria. Siendo el tiempo de acceso a un registro interno es mucho menor que para la memoria principal. Las **desventajas** es que el espacio de direcciones es limitado y que el número de registros es limitado.

Si se hace un uso masivo del direccionamiento a registros en un repertorio de instrucciones, los registros del procesador se emplearán intensamente. Depende del programador decidir qué valores deben mantenerse en registros y cuáles deben almacenarse en memoria principal. La mayoría de los procesadores modernos emplean múltiples registros de uso general, cargando al programador en lenguaje ensamblador (cuando se desarrolla compiladores, por ejemplo) con la responsabilidad de conseguir una ejecución eficiente.

Direccionamiento Indirecto a Registro

Igual que el direccionamiento de registro es análogo al direccionamiento directo, el direccionamiento indirecto a registro es análogo al direccionamiento indirecto. La diferencia estriba en si el campo de direcciones hace referencia a una posición de memoria principal o a un registro.

```
EfectiveAddress = [R]
.model small
.data
    value       dw 10
    value_ptr   dw offset value
    result      dw ?

.code
main proc
    mov ax, value_ptr
    mov ax, [ax]
    add ax, 5
    mov result, ax

    mov ah, 4Ch
    int 21h
main endp
end main
```

Las **ventajas** y limitaciones son las mismas que para el direccionamiento indirecto. Em ambos casos la limitación es que el campo de direcciones, se supera haciendo que dicho campo referencie una posición de memoria que contenga la dirección del operando. Además, el direccionamiento indirecto a registro requiere de una referencia menos a memoria para obtener el operando.

Direccionamiento con Desplazamiento

Un modo de direccionamiento que combina las posibilidades de los direccionamientos directo e indirecto con registro.

```
EfectiveAddress = A + (R)
.model small
.data
    array   dw 1, 2, 3, 4, 5
    index   dw 2
    result  dw ?

.code
main proc
    mov ax, index
    mov bx, offset array

    mov cx, 2
    mov ax, [bx + ax * cx]
    mov result, ax

    mov ah, 4Ch
    int 21h
main endp
end main
```

Requiere que tenga dos campos de direcciones, al menos uno de los cuales es explícito. El valor contenido en uno de los campos de direcciones se utiliza directamente (valor = A),

el otro campo de direcciones, o una referencia implícita, se refiere a un registro (valor = R), cuyo contenido se suma a A para obtener la dirección efectiva del operando. Los tres usos más comunes de este modo de direccionamiento son:

- **Desplazamiento base.** El registro referenciado contiene una dirección de memoria, y el campo de dirección contiene un desplazamiento desde esa dirección base. Siendo la referencia explícita o implícita.
- **Desplazamiento indexado.** El campo de dirección referencia una dirección de memoria, y el registro referenciado contiene un desplazamiento positivo desde esa dirección. Obsérvese que es justo lo opuesto de la interpretación del desplazamiento base. Puede ser preindexado o postindexado.
- **Desplazamiento relativo.** También llamado direccionamiento relativo al PC, el registro referenciado implícitamente es el contador de programa (PC). La dirección de la instrucción actual se suma al campo de direcciones para obtener la dirección efectiva del operando.

Direccionamiento de Pila

La pila tiene un puntero asociado, cuya valor es la dirección de la cabecera o tope de la pila. Este puntero se mantiene en un registro, el cual se denomina **Stack Pointer (SP)**. El direccionamiento de pila es un caso especial, porque 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 la referencia a memoria, sino que se operan con la cabecera de la pila.

Modos de direccionamiento del Pentium

El mecanismo de traducción de direcciones del Pentium produce una dirección denominada 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.**

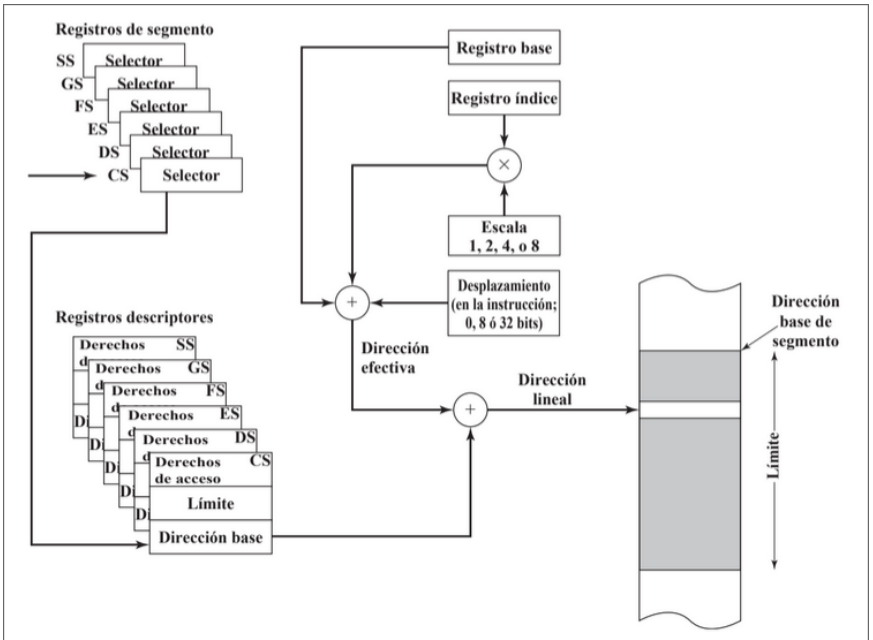

Figura 11.2. Cálculos en el modo de direccionamiento del Pentium.

Figure 1: Modos de direccionamiento del Pentium [1]

El pentium contiene seis registros de segmento, cada uno de los cuales contiene la dirección de comienzo de un segmento. Además de dos registros que pueden emplearse para construir una dirección, el **registro base (BX)** y el **registro índice (SI)**.

Tabla 11.2. Modos de direccionamiento del pentium.

Modo	Algoritmo
Inmediato	Operando = A
Registro	LA = R
Con deplazamiento	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

Figure 2: Modos de direccionamiento del Pentium [1]

Formatos de Instrucciones

Un formato de instrucciones define la descripción en bits de una instrucción. Debe de incluir un **código de operación (codop)**, e incluir cero o más operandos. Cada operando explícito se referencia utlizando uno de los **modos de direccionamiento**. En la mayoría de los repertorios de instrucciones se emplean más de un formato de instrucción.

Longitud de Instrucción

El aspecto más básico a considerar es la longitud de la instrucción. Afecta y se ve afectada por el tamaño de la memoria, su organización, la estructura de buses, la complejidad del procesador y la velocidad del mismo. Esta decisión define la riqueza y la flexibilidad de la máquina desde un punto de vista del programador en lenguaje ensamblador.

El programador desea más *codops*, más operandos, más modos de direccionamiento y mayor rango de direcciones. Todo esto requiere de bits, y empuja hacia longitudes de instrucción mayores. No obstante una instrucción mayor puede ser improcedente. **No porque una instrucción de 64 bits ocupe el doble de espacio que una de 32 bits, significa que sea el doble de útil.** La longitud de palabra de memoria, **(16 bits)**, es, en cierto sentido, la unidad natural de organización.

Asignación de los Bits

Un aspecto igualmente difícil es cómo asignar los bits en un formato de instrucción. Para una longitud de instrucción dada, existe un compromiso entre el número de *codops* y la capacidad de direccionamiento. Un mayor número de *codops* implica más bits en el campo de codop. Reduciendo el número de bits disponibles para direccionamiento. Los siguientes factores influyen en la asignación de bits:

- **Número de modos de direccionamiento.** Pueden a veces indicarse de forma implícita. En otros casos, deben de especificarse explícitamente. requiriendo uno o más bits de modo.

- **Número de operandos.** Las instrucciones típicas de las máquinas actuales permiten dos operandos. Cada dirección de operando podría requerir de un indicador de modo dentro de la instrucción.
- **Registros frente a memoria.** Una máquina debe de disponer de registros para traer datos al procesador a fin de procesarlos. En el caso de solo tener un registro visible (el acumulador), la dirección del operando está implícita y no consume bits. Incluso con varios registros, se ocupan pocos bits para especificar el registro. Normalmente se disponen de 8 a 32 registros visibles para el usuario.
- **Número de conjuntos de registros.** Distintas máquinas tienen un conjunto de registros de uso general, entre 8 y 16 registros. Una ventaja que ofrece este enfoque es que, requiere menos bits por instrucción. Así el codop determina de forma implícita qué conjunto de registros se está referenciando.
- **Rango de direcciones.** Para referencias a memoria, este rango está relacionado con el número de bits disponibles para direccionamiento. Dado que esto impone limitaciones, rara vez se utiliza el direccionamiento directo, normalmente se emplea el direccionamiento con desplazamiento.
- **Granularidad de las direcciones.** Para direcciones referenciando a memoria en lugar de registros, otro factor que afecta es su granularidad de direccionamiento. El direccionamiento por bytes es conveniente manipular caracteres pero requiere, para un tamaño de memoria dado, más bits de dirección.

PDP-8

Un diseño sencillo para un computador de uso general fue el del PDP-8. Este utiliza instrucciones de 12 bits, opera con palabras de 12 bits, y solo tiene un registro de uso general, el acumulador. El formato de instrucción del PDP-8 es bastante eficiente. Permite direccionamiento indirecto, direccionamiento con desplazamiento, e indexado. Dispone de un total de 35 instrucciones.

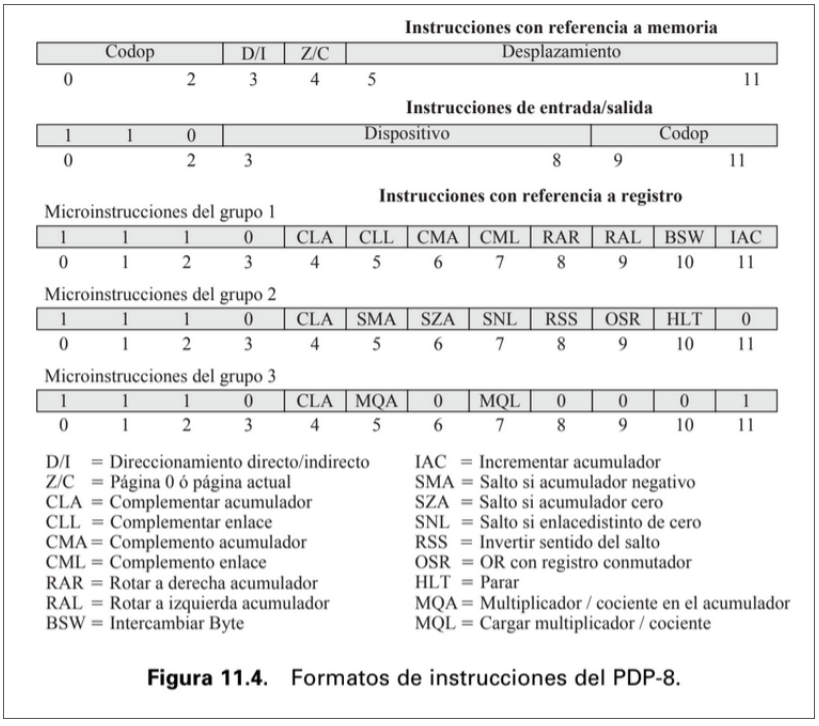

Figura 11.4. Formatos de instrucciones del PDP-8.

Figure 3: Formato de instrucciones del PDP-8 [1]

PDP-10

Tiene una longitud de instrucción de 36 bits, y una longitud de palabra de 36 bits. Permite 512 operaciones, con 365 instrucciones diferentes. Con 16 registros de uso general.

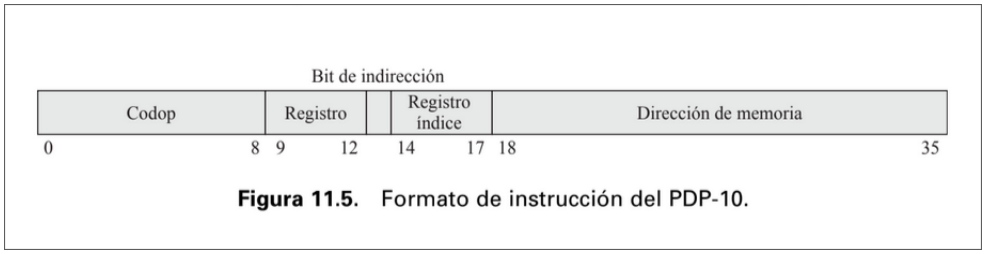

Figura 11.5. Formato de instrucción del PDP-10.

Figure 4: Formato de instrucción del PDP-10 [1]

Instrucciones de Longitud Variable: PDP-11

El PDP-11 se diseñó para proporcinar un repertorio de instrucciones flexible dentro de las limitaciones de un microprograma de 16 bits. Este emplea 8 registros de uso general de 16 bits, teniendo punteros de pila. Una ventaja que posee es el modo de direccionamiento con operando, en lugar de con el codop. Esta independencia se denomina *ortogonalidad*. Tiene 13 formatos de instrucción diferentes, con una longitud de instrucción de 16 bits.

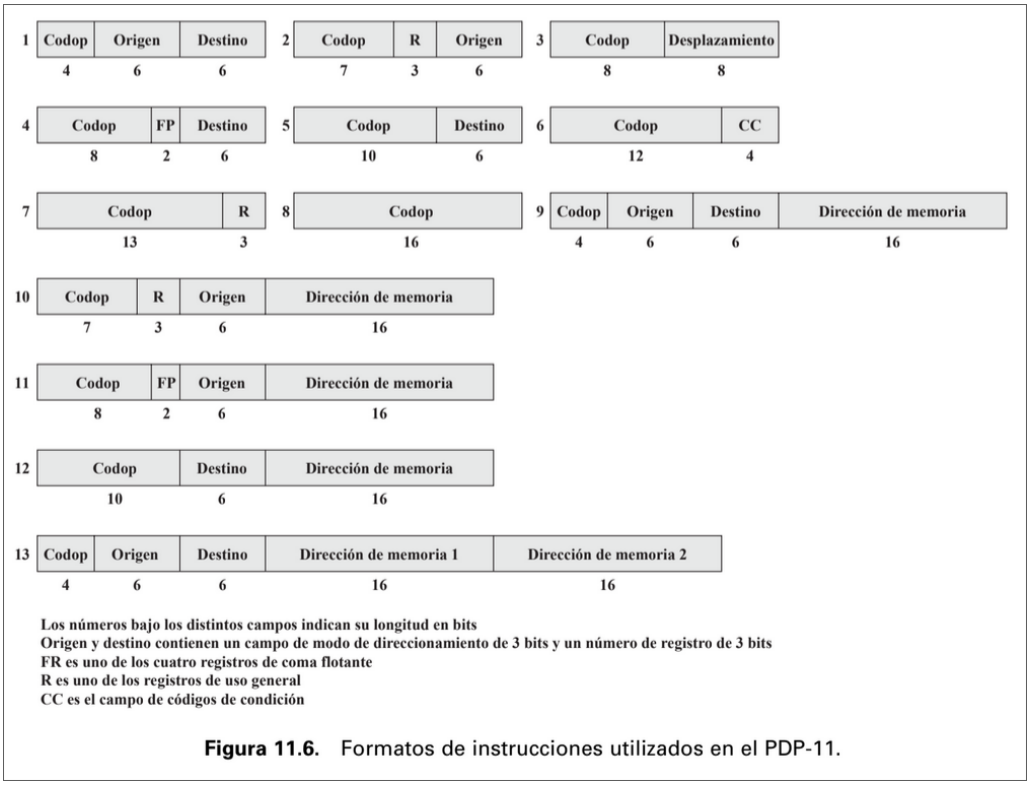

Figura 11.6. Formatos de instrucciones utilizados en el PDP-11.

Figure 5: Formatos de instrucciones del PDP-11 [1]

Formatos de instrucción del Pentium

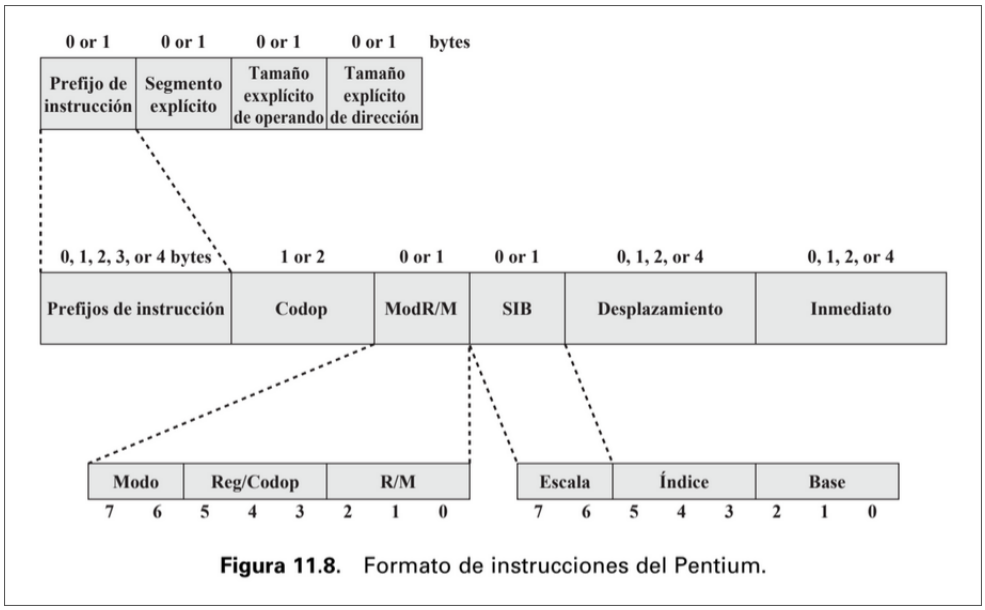

Figure 6: Formatos de instrucción del Pentium [1]

References

[1] W. Stallings. *Organización y arquitectura de computadores.* Fuera de colección Out of series. Pearson Educación, 2006.

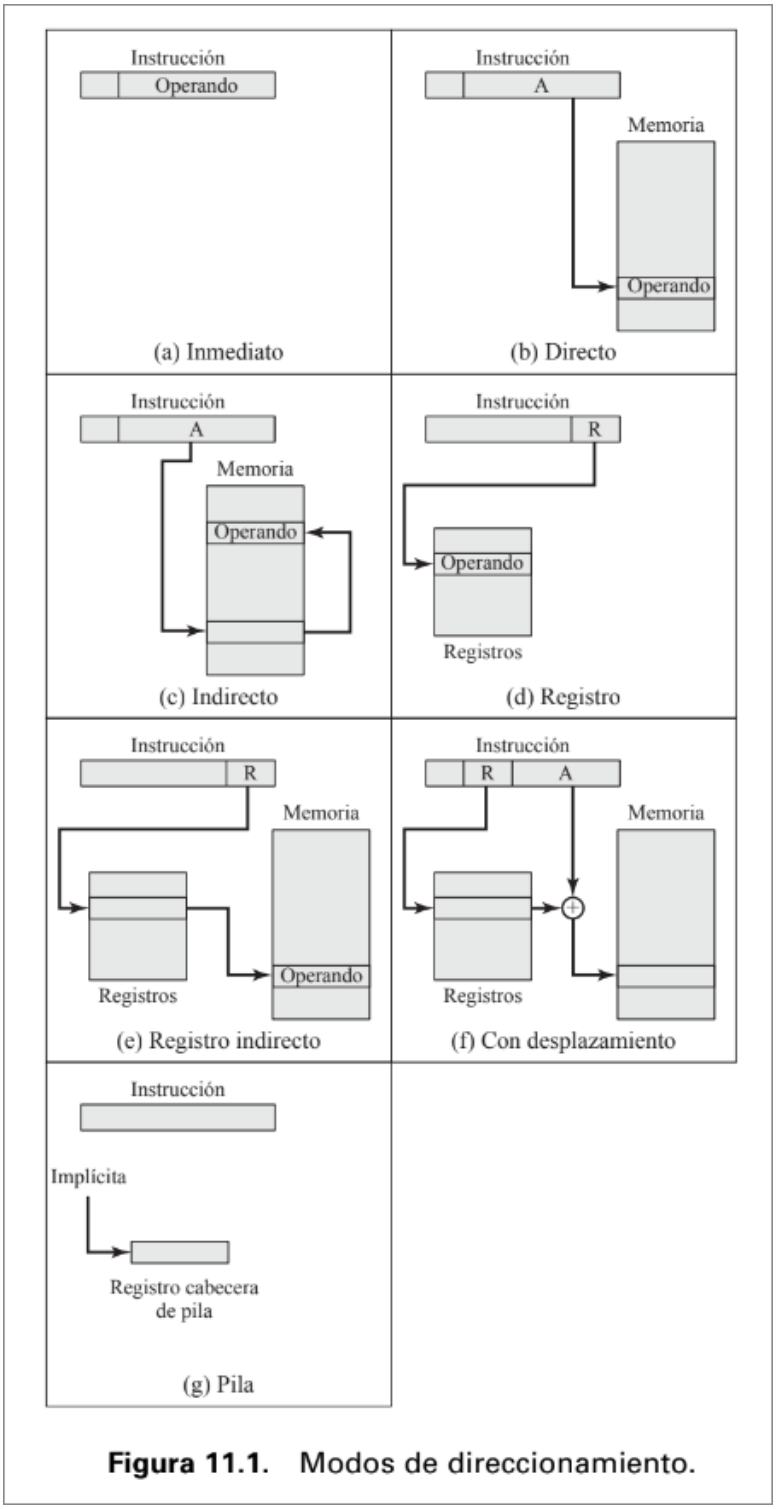

Figura 11.1. Modos de direccionamiento.

Figure 7: Modos de direccionamiento [1]