

Centro Asociado Palma de Mallorca

**Arquitectura
de
Ordenadores**

Tutor: Antonio Rivero Cuesta

Unidad

Didáctica 2

Estructura de un Computador

Tema

7

**Modos
de
Direccionamiento**

Tanto las instrucciones que forman los programas, como los datos sobre los que éstos operan, se encuentran almacenados en la memoria del computador o en registros de la CPU.

El computador debe saber en todo momento cómo localizar un dato determinado que necesite, o una instrucción a la que el programador quiera que salte para proseguir con la ejecución del programa en curso.

Éste es el *objetivo* de los modos de direccionamiento.

Los modos de direccionamiento especifican reglas que permiten interpretar adecuadamente el campo de dirección de la instrucción que se esté ejecutando en cada momento.

De esta forma, se consigue referenciar al objeto, bien sea un dato o una dirección, con una mayor eficiencia.

Puesto que permiten al programador escribir programas en ensamblador más rápidos y que aprovechen mejor los recursos del computador.

Justificación de los Modos de Direccionamiento

A la hora de abordar el estudio de los modos de direccionamiento, hay que tener presente que el número de bits que forman una dirección determina la máxima cantidad de memoria a la que se puede tener acceso:

- Con 8 bits se pueden direccionar hasta 256 bytes, o lo que es lo mismo, 0,25 kbytes.
- Con 16 bits hasta 64 kbytes.
- Con 24 bits hasta 16Mbytes.
- Con 32 bits podemos llegar hasta los 4 Gbytes.

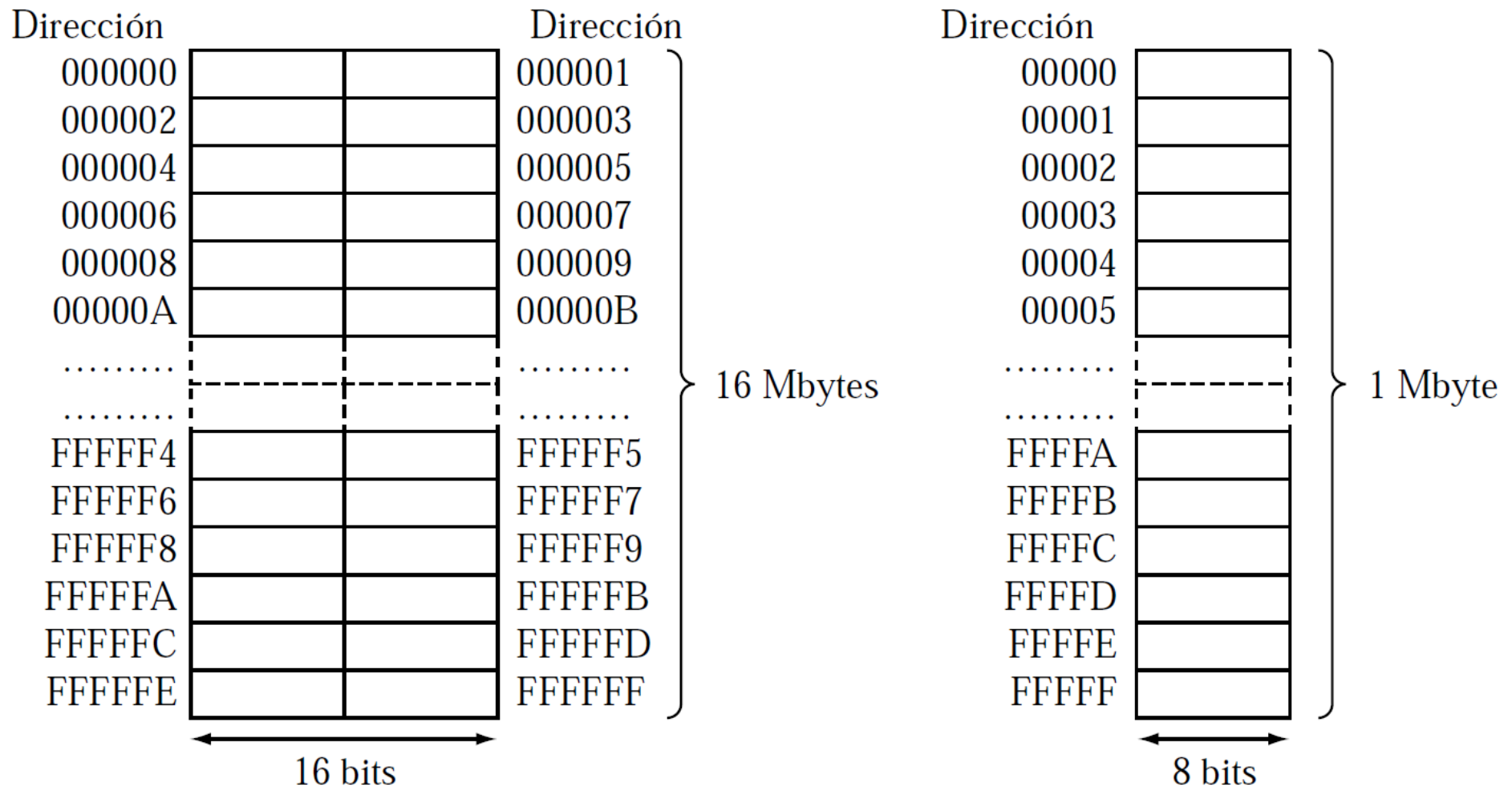


FIGURA 10.1. Disposición de la memoria principal en distintos modelos de la familia del 68000: (a) MC68000 y MC68010; (b) MC68008. Las posiciones de memoria se han numerado en hexadecimal. Con 6 dígitos hexadecimales (que equivalen a 24 binarios) se consigue direccionar un total de 16Mbytes de memoria; con 5 dígitos un total de 1 Mbyte.

De entre las ventajas que proporcionan los modos de direccionamiento, cabe destacar las siguientes:

- Reducción del espacio en memoria.
- Capacidad para reubicar código.
- Facilidad para manejar estructuras de datos.

Reducción Espacio Memoria

Se reduce el número de bits en el campo de dirección de la instrucción.

Con lo que también se reduce el espacio que ocupa el programa cuando se almacena en memoria.

Capacidad Reubicar Código

Permiten la reubicación de programas en memoria para su ejecución.

Para conseguirlo, se trabaja con direcciones relativas de memoria a partir de una dirección dada, sin necesidad de tener que modificar el código del programa.

Facilidad para Manejar Estructuras de Datos

La utilización de determinados modos de direccionamiento facilita enormemente las tareas asociadas al acceso y manipulación de vectores, matrices, listas, colas, etc.

En un procesador determinado, el número de bits necesarios para especificar una dirección de la memoria principal y el tamaño de los registros de propósito general que maneja la CPU están relacionados.

De este modo se consigue que el programador pueda almacenar direcciones en los registros o en pares de registros.

Por otro lado, los mismos registros pueden ser usados para guardar datos y direcciones.

Un dato de 2 bytes se conoce como *word* o palabra.

Un dato de 4 bytes se conoce como *longword* o palabra larga.

No todos los procesadores están preparados para aceptar todos los modos de direccionamiento.

El propósito de los modos de direccionamiento es proporcionar una dirección efectiva para el operando u operandos de las instrucciones que forman los programas.

Cuando se trata de una instrucción que manipula datos, la dirección efectiva es la dirección del dato que se desea manipular.

Cuando se trata de una instrucción de salto, la dirección efectiva indica la dirección de la instrucción a la que se desea saltar para proseguir con la ejecución del programa.

Descripción de los Modos de Direccionamiento

El modo de direccionamiento más simple es el direccionamiento inmediato, en el cual realmente no se direcciona nada.

La instrucción contiene directamente los datos que necesita.

El resto de modos de direccionamiento, se pueden clasificar como directos o indirectos.

En los modos de direccionamiento directos, la dirección efectiva se toma directamente de la instrucción o bien se calcula combinando el valor que aparece en el campo de dirección de la instrucción con otro valor contenido ya de antemano en un registro.

En los modos indirectos, la dirección que se obtiene a partir de la instrucción, o bien se calcula a partir de ésta, indica la posición de memoria que contiene la dirección efectiva del objeto.

Por este motivo, se puede decir que en los modos de direccionamiento indirectos, se precisa de un acceso adicional a memoria con respecto a los modos directos.

Direccionamiento Inmediato

El objeto, que normalmente será un operando se especifica como parte de la propia instrucción.

Esto quiere decir que la instrucción, en lugar de contener un campo de dirección, contiene un campo de operando.

Donde viene especificado el dato real sobre el que se debe realizar la operación que indique el código de operación de la instrucción.

No es un modo realmente necesario a la hora de planificar la arquitectura de un procesador.

Siempre se puede recurrir a guardar las constantes en posiciones de memoria y acceder más tarde a éstas.

Para ello, se puede recurrir a algún otro modo de direccionamiento que permita calcular como dirección efectiva la dirección en la que previamente se guardaron las constantes.

El direccionamiento inmediato presenta algunas ventajas.

Entre las cuales está reducir el tamaño de los programas y aumentar la velocidad de ejecución de los mismos eliminando, en muchas ocasiones, los accesos a memoria para llegar a las constantes.

El operando inmediato forma parte de la instrucción, pero tal como muestra la Figura 10.3, éste puede formar parte de la misma palabra que contiene el código de operación, o bien ir especificado en la palabra o palabras inmediatamente siguientes en memoria.

El número de bits del formato de instrucción que se dedican a codificar el campo de operando, limita el tamaño del operando que se puede especificar.

Cod. Operación	Operando
----------------	----------

(a)

Código de Operación
Operando

(b)

Código de Operación
Operando

(c)

En la mayor parte de los casos las constantes que se manejan son números enteros con signo.

Si suponemos que la memoria con la que se trabaja tiene un ancho de palabra de 16 bits y utilizamos el sistema de complemento a dos para la representación de números enteros, según la figura 3 en el caso (a) tendríamos 8 bits para especificar el operando, con lo cual los valores de dicho operando podrían estar comprendidos entre -128 y 127 .

En el caso (b) tendríamos 16 bits para especificar el operando y los valores representados estarían entre -32.768 y 32.767 .

En el caso (c) tendríamos 32 bits para especificar el operando y los valores que caerían dentro del rango de representación del formato mostrado en la figura estarían comprendidos entre $-2.147.483.648$ y $2.147.483.647$.

Este modo de direccionamiento es muy utilizado en la inicialización de registros, operaciones numéricas y operaciones de comparación (para el control de saltos condicionales):

- Permite especificar constantes para inicializar registros y variables en memoria que más tarde serán usadas por el resto del programa.
- Proporciona un método sencillo para llevar a cabo operaciones aritméticas o lógicas en las que intervengan constantes y variables del programa. Se utiliza en la manipulación de registros que se usan para contener punteros, en este caso, permite indicar la cantidad que es necesario

avanzar para apuntar a los distintos campos de una estructura de datos.

- Por último, se usa para desarrollar operaciones de comparación de variables numéricas o alfanuméricas, como es el caso de las tareas de comparación que permiten llevar a la práctica bifurcaciones condicionales dentro de la secuencia de instrucciones de un programa.

Direccionamiento Directo

La dirección real del objeto aparece en la instrucción.

La instrucción contiene un campo de dirección donde se especifica la dirección donde se encuentra el dato.

Este tipo de direccionamiento resulta útil cuando se pretende acceder a datos estáticos.

Direccionamiento Directo Absoluto

La dirección efectiva del objeto aparece en el campo de dirección de la instrucción.

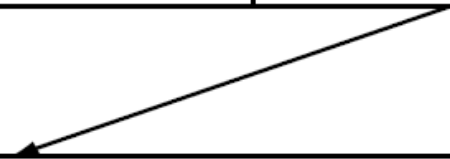
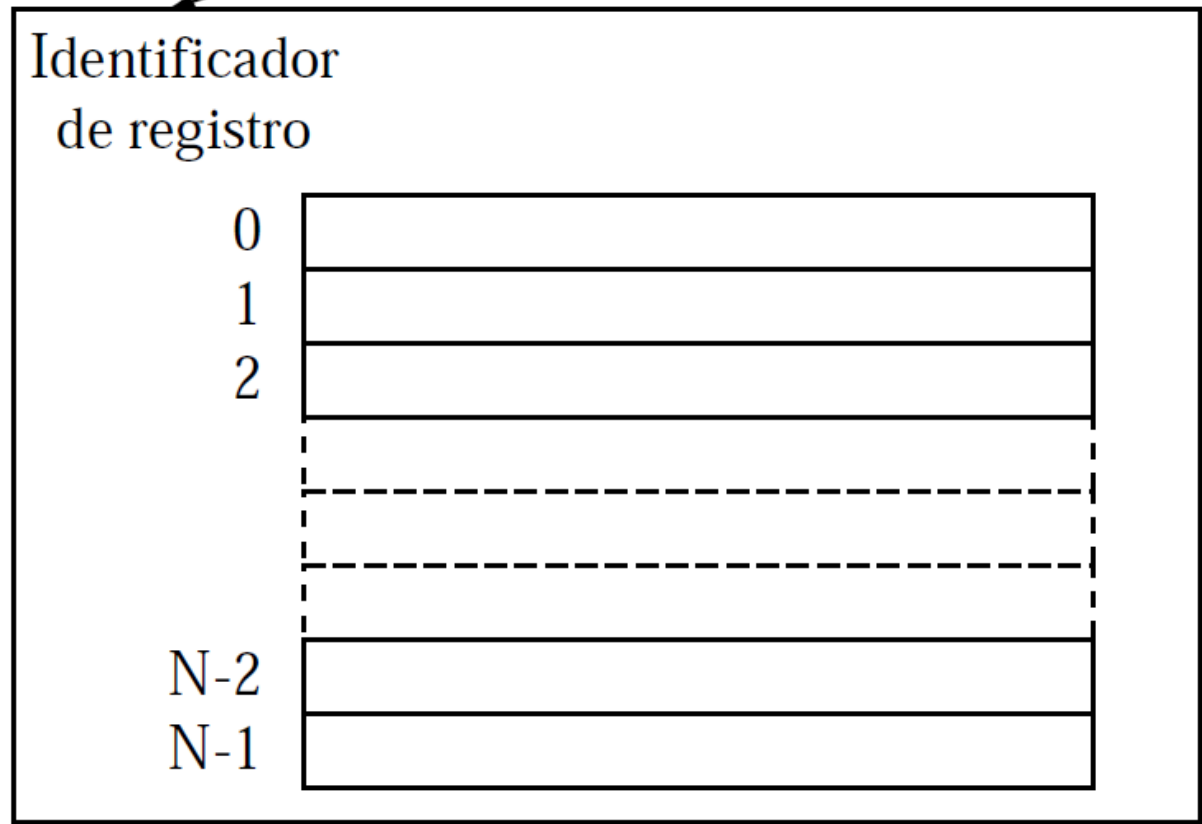
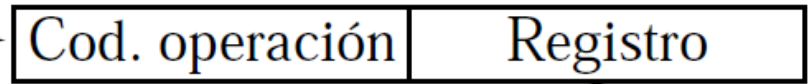
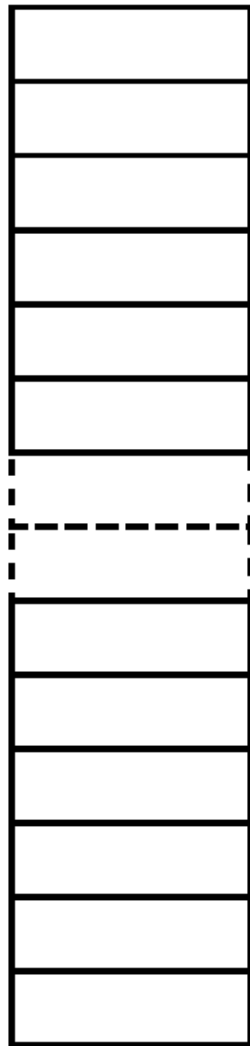
Para ello se distingue entre tres posibilidades en cuanto a cómo se especifica la dirección efectiva:

- Direccionamiento de registro.
- Direccionamiento de memoria.
- Direccionamiento de página base.

Direccionamiento de Registro

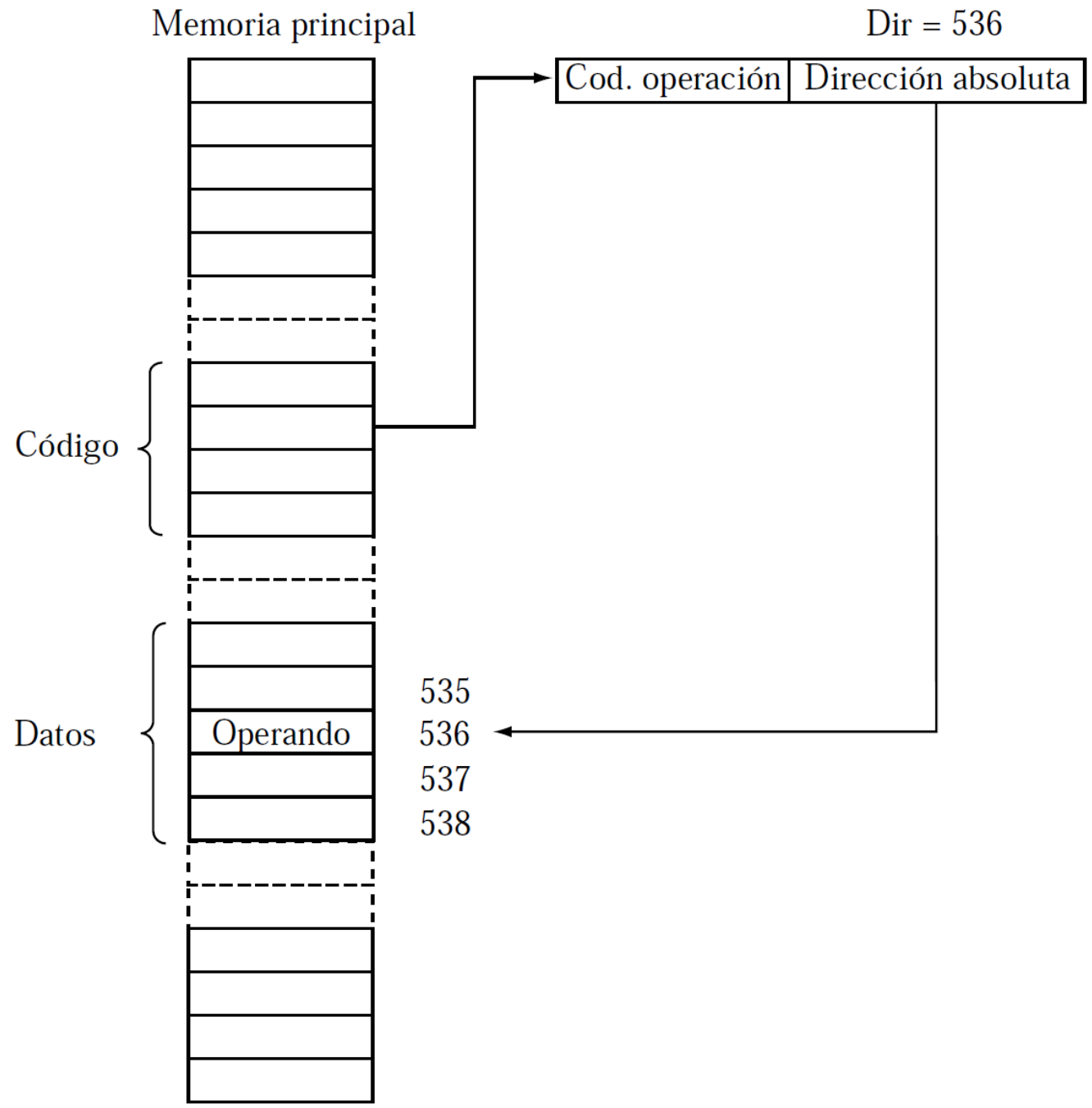
El campo de dirección de la instrucción se limita a identificar un registro de la CPU de propósito general (ver Figura 10.4).

Memoria principal



Direccionamiento de Memoria

El campo de dirección de la instrucción identifica una posición de la memoria principal (ver Figura 10.5).



Direccionamiento de Página Base

El campo de dirección de la instrucción se refiere únicamente a una parte del mapa de memoria denominada página base.

De esta forma, al tratarse de una parte de la memoria, se necesitan menos bits para referenciarla, con lo cual se consigue reducir el tamaño de la instrucción, pero en contrapartida, se limita el acceso de la memoria principal a una zona en particular de la misma.

Direccionamiento Directo Relativo

El campo de dirección de la instrucción contiene un desplazamiento que habrá que sumar a la dirección contenida en un puntero para determinar la dirección efectiva del objeto.

El puntero contiene la dirección de referencia del objeto, de modo que para calcular la dirección efectiva del mismo, se modifica la dirección de referencia en función de lo que indique el campo de dirección de la instrucción, que en este caso es más bien un campo de desplazamiento.

Puesto que lo normal será tener que acceder a la dirección contenida en el puntero varias veces, en la mayoría de los casos se opta por almacenarlo en un registro, que tiene la ventaja de consumir menos tiempo de acceso.

En cuanto al campo de desplazamiento, lo habitual es que permita indicar tanto desplazamientos positivos como negativos a partir de la dirección contenida en el puntero.

Para ello, se suele emplear la representación del desplazamiento en el sistema de complemento a 2.

Según esto, el radio de alcance definido por la memoria que se puede direccionar, abarca una zona de posiciones crecientes y decrecientes respecto a la dirección de referencia, característica que resulta muy conveniente para recorrer determinadas estructuras de datos.

La principal ventaja que presenta este modo de direccionamiento es que, en función del tamaño en bits del registro que contiene el puntero, podemos acceder a zonas altas de memoria utilizando instrucciones en las que el campo de dirección se codifique mediante un número pequeño de bits.

Como contrapartida, cabe destacar que se produce un retardo en la ejecución de la instrucción con respecto al direccionamiento directo absoluto, puesto que antes de realizar la lectura en memoria del dato en cuestión es necesario llevar a cabo una operación de suma.

Sin embargo, este retraso es suficientemente pequeño como para que la utilización del direccionamiento relativo a registro quede suficientemente justificada.

En función del registro que se emplea como puntero, existen varias posibilidades para el direccionamiento directo relativo que se describen a continuación:

- Relativo al contador de programa.
- Relativo a un registro base.
- Relativo a un registro índice.
- Relativo a pila.

Relativo al Contador de Programa

En este modo de direccionamiento el contenido del registro contador de programa, es el que actúa como puntero.

Por tanto, para calcular la dirección efectiva se suma el contenido del registro contador de programa con el contenido del campo de dirección de la instrucción.

Hay que tener en cuenta que una vez que se ha leído una instrucción de la memoria y ésta pasa al registro

de instrucción, el CP apunta a la instrucción del programa inmediatamente siguiente.

De modo que si en un momento dado una instrucción determinada contiene un desplazamiento positivo de 5 posiciones con respecto al CP, en realidad estaría apuntando a la sexta instrucción siguiendo el orden creciente del programa contando a partir de la instrucción en curso.

Se utiliza frecuentemente en instrucciones que representan saltos condicionales en la secuencia de ejecución del programa, especialmente cuando la dirección de salto se encuentra próxima a la posición que ocupa dentro del programa la instrucción de bifurcación.

Relativo a un Registro Base

En este caso el registro que actúa como puntero es un registro base (de propósito general) de la CPU.

El campo de dirección de la instrucción proporciona un desplazamiento, que sumado con la dirección de referencia contenida en el registro base, permite calcular la dirección efectiva del objeto. Ver figura 10.6.

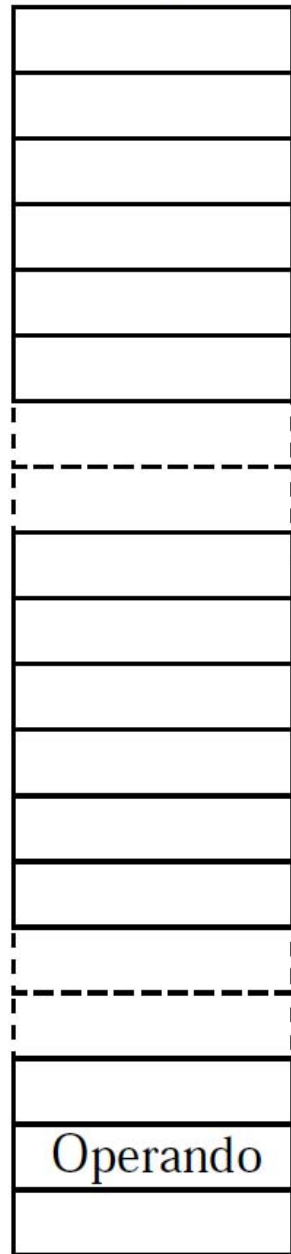
Se utiliza para acceder a una zona de datos en memoria, donde, por ejemplo, se hayan almacenado los datos de una tabla.

De esta forma, se guardaría en el registro base la dirección de la posición de memoria que ocupara el primer dato de la tabla, y a partir de aquí solo se tendrían en cuenta los desplazamientos relativos contenidos en la instrucción, esto es, la posición dentro de la tabla a la que queremos acceder.

También se utiliza este modo de direccionamiento cuando se mueven datos y programas de unos segmentos de memoria a otros, en tal caso, se guardaría en el registro base el valor de comienzo de la nueva zona de memoria, y a partir de aquí, se recalcularían las direcciones a las que se hace

referencia en el programa, sin necesidad de volver a cambiar todo el programa para ello.

Memoria principal

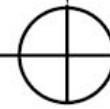


391 \$
392 \$
393 \$

Operando

Reg = 2

Desp = 10



Identificador
de registro

0

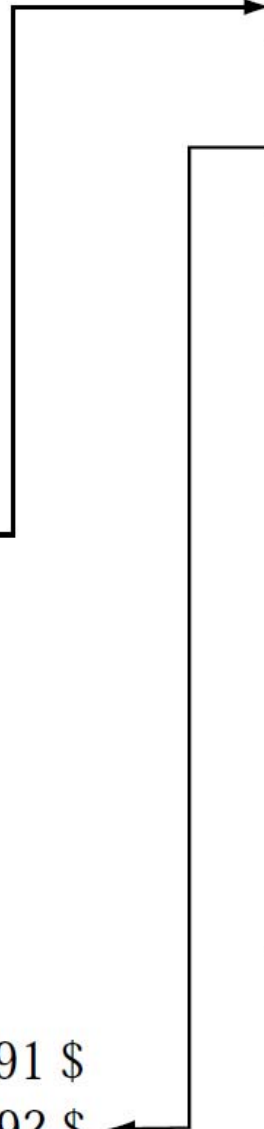
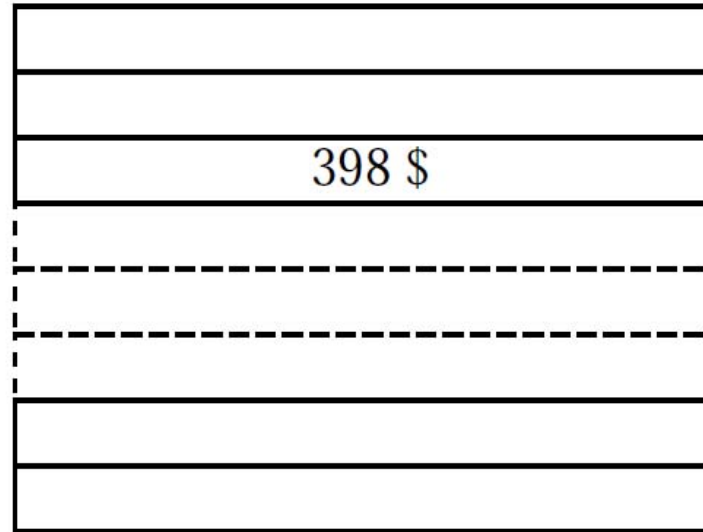
1

2

398 \$

N - 2

N - 1



Relativo a un Registro Índice

Este modo de direccionamiento es muy similar a los anteriores en cuanto a la forma en que se calcula la dirección efectiva.

Con la salvedad de que en esta ocasión, el registro que interviene se denomina registro índice.

Sin embargo, el uso que de él hace el computador es un poco distinto.

A pesar de ser un modo de direccionamiento útil en muchas ocasiones, su principal aplicación está destinada al direccionamiento de arrays.

En este caso, el campo de dirección de la instrucción contiene la dirección inicial del array de datos en memoria, mientras que el registro índice contiene un valor índice que nos permite ir recorriendo el array.

En la mayor parte de las ocasiones en que es utilizado este modo de direccionamiento, está permitido el incremento o decremento del registro índice de forma automática para facilitar aún más las labores de direccionamiento.

En función de esta última propiedad distinguimos cuatro posibles casos:

- Preautoincremento.
- Preautodecremento.
- Postautoincremento.
- Postautodecremento.

Preautoincremento

El registro índice se incrementa y luego se suma al valor que marca el campo de dirección de la instrucción para obtener la dirección efectiva.

Preautodecremento

El registro índice se decrementa y luego se suma al valor que marca el campo de dirección de la instrucción para obtener la dirección efectiva.

Postautoincremento

En primer lugar, se calcula la dirección efectiva sumando al valor del campo de dirección de la instrucción, el contenido del registro índice, y a continuación y para finalizar la operación, se incrementa este último.

Postautodecremento

En primer lugar, se calcula la dirección efectiva sumando al valor del campo de dirección de la instrucción el contenido del registro índice, y a continuación para finalizar, se decrementa este último.

Relativo a Pila

Cuando se utiliza este modo de direccionamiento el registro que se emplea como referencia es un puntero de pila, denominado SP.

El valor que se guarda en este registro es la posición que señala la cima de la pila.

Una pila es una estructura de datos que sigue la idea LIFO (Last In First Out), o lo que es lo mismo, el último dato que entra es el primero en salir.

Puede crecer hacia direcciones de memoria crecientes o decrecientes, hecho que habrá que tener en cuenta cuando se deposite o retire un dato de la pila para modificar el puntero en consonancia.

Si la pila aumenta hacia valores de memoria decrecientes y se quiere insertar un nuevo dato, se necesitará un modo de direccionamiento relativo a pila con preautodecremento.

Si lo que se quiere es extraer un dato de la pila, se necesitará un direccionamiento relativo a SP con posdecremento, ya que siempre el valor del registro SP debe estar apuntando al último valor depositado en la pila, esto es, la cima de la pila.

Normalmente existe una pila denominada pila del sistema que es la que utiliza el computador para llevar a cabo sus funciones mientras se ejecutan los programas.

Esta pila será la que contenga las direcciones de retorno de las subrutinas o las interrupciones, por ejemplo.

Además de esta pila, el usuario puede optar por crearse su propia pila para almacenar datos, o incluso varias pilas, en función de las necesidades del programa que esté diseñando.

Direccionamiento Indirecto

Se denomina así porque lo que se maneja en primera instancia no es el objeto, como ocurre en el direccionamiento directo, sino la dirección efectiva del mismo, por tanto, para determinar el objeto, necesitaremos realizar un nuevo acceso a memoria.

Este modo de direccionamiento es útil cuando se trabaja con una tabla de punteros para acceder a los datos.

Pueden aparecer situaciones en las que se manejen varios niveles de indirección, aunque no suele ser muy frecuente, ya que se ralentiza el acceso a los datos al aumentar el número de lecturas en memoria.

Sin embargo, lo que sí suele suceder es que se combine el modo de direccionamiento indirecto con alguno de los direccionamientos relativos descritos anteriormente.

Memoria principal

Dir = 542

