

## **-MODOS DE DIRECCIONAMIENTO**

Los temas anteriores mostraron las ventajas de la Lógica Programada sobre la Lógica Cableada. Esta ventaja está basada en que con un único sistema físico (hardware) y bajo el control de programas (software) se pueden realizar las más variadas funciones. Tales programas están constituidos por componentes elementales, de la misma manera que la parte física, éstos componentes básicos son las denominadas Instrucciones. Una Instrucción es una combinación de dígitos binarios, que debe ser interpretada por la lógica de control del Procesador para definir lo que debe hacer el sistema (tipo de operación) y cual es el dato o los datos con que debe trabajar. Es decir, la instrucción debe especificar dos componentes fundamentales:

- a) El código de operación, que informa sobre la naturaleza o tipo de la Instrucción.
- b) El operando (o los operandos), la información a procesar.

En cuanto al código de operación (OP), cuantos más tipos diferentes de Instrucciones realice el sistema, más bits se necesitarán para especificarlas. Los Microprocesadores (MP) actuales tienen un conjunto de Instrucciones (repertorio básico) muy potente, son comunes para los MP de 8 bits de la primera generación (8080, 6800 por ejemplo) un número de más de 70 Instrucciones, en algunos más actuales el número de Instrucciones es bastante mayor llegando a 150 o más. Por esto generalmente basta con 8 bits para especificar el código de operación, aunque en procesadores más potentes se utilizan hasta 2 bytes.

La referencia al operando debe hacerse en una o más palabras adicionales al código de operación, las cuales indican al sistema donde ubicar el dato a procesar.

Para el caso particular de un procesador que posea aproximadamente unas 70 instrucciones, parece excesivo el hecho de emplear 8 bits (1 byte) para especificar el código de operación. El hecho de ocupar un número mayor de bits para definir la instrucción, se basa en que existe otro factor a considerar en el código de operación, es la posibilidad de utilizar distintos modos para ubicar el dato con el cual debe operar y esto obliga a utilizar algunos bits, dentro del campo del código de operación, para indicar el modo de direccionamiento o modo de ACCESO al operando.

En las instrucciones de bifurcación y llamados a subrutina también se debe considerar un modo de acceso para poder obtener la próxima Instrucción a ejecutar.

La potencia de cualquier procesador está basada en su facilidad o habilidad para acceder a Memoria, los modos de direccionamiento son los que le proveen esta capacidad, ya que estos definen la forma en la cual una instrucción obtiene el dato requerido para su ejecución.

## **MODOS DE DIRECCIONAMIENTO**

En esta parte del curso, se detallarán las distintas formas usadas por los MP para obtener las direcciones que le permitan acceder a los operandos. Existe una gran variedad de Modos de Direccionamiento como se verá a continuación. Debe aclararse que, en general, ningún procesador utiliza todos los Modos de Direccionamiento, sino que solo hace uso de algunos de ellos, esto está definido por su Arquitectura y por la potencialidad que el fabricante le desea dar.

## DIRECCIONAMIENTO DE OPERANDOS

Para realizar una determinada tarea (Instrucción) el computador debe operar con datos almacenados en memoria. Las Instrucciones que procesan datos contienen implícita o explícitamente Direcciones que le permiten al procesador localizar el o los datos a utilizar en la instrucción.

Como se sabe la cantidad de líneas del bus de direcciones indica la cantidad máxima de memoria direccionable por el procesador; en general, procesadores con 16 líneas en el bus de direcciones pueden direccionar 64 Kbytes de memoria. Los últimos procesadores como por ejemplo el 68.000 y Z8000, por su arquitectura de 32 y 31 líneas de direcciones respectivamente, pueden direccionar más de 2 M byte de memoria.

Existen varias maneras de especificar la dirección del dato, comúnmente llamada Dirección Efectiva (DE), como parte de una instrucción. El manejo de Direcciones requiere en algunas aplicaciones el uso de determinados modos de direccionamiento, por lo tanto el programador deberá elegir entre los posibles el más adecuado a la aplicación específica.

La finalidad de los modos de direccionamiento es el de proveer la Dirección Efectiva (DE) de una Instrucción. En instrucciones con manejo de datos (+,-, AND,OR, etc) la DE es la dirección del dato utilizado por la instrucción. En una instrucción de salto, la DE es la dirección de la Instrucción a la cual se debe producir el salto.

Los modos de direccionamiento se pueden clasificar en dos grandes grupos.

### a) Directos

### b) Indirectos

En los directos, la DE es tomada Directamente de la Instrucción o calculada mediante un valor de algún registro. Este es el modo más común sobre todo considerando los procesadores más simples. En los modos Indirectos o Diferidos, la dirección obtenida de la Instrucción o de algún cálculo corresponde a la dirección de Memoria en que se encuentra almacenada la DE. En algunos procesadores como el 6809 y la PDP-11 cada modo de direccionamiento directo tiene su correspondiente indirecto, otros procesadores, tales como el Z8000 y el 8086 solo tienen un modo de direccionamiento Indirecto (Indirecto vía registro).

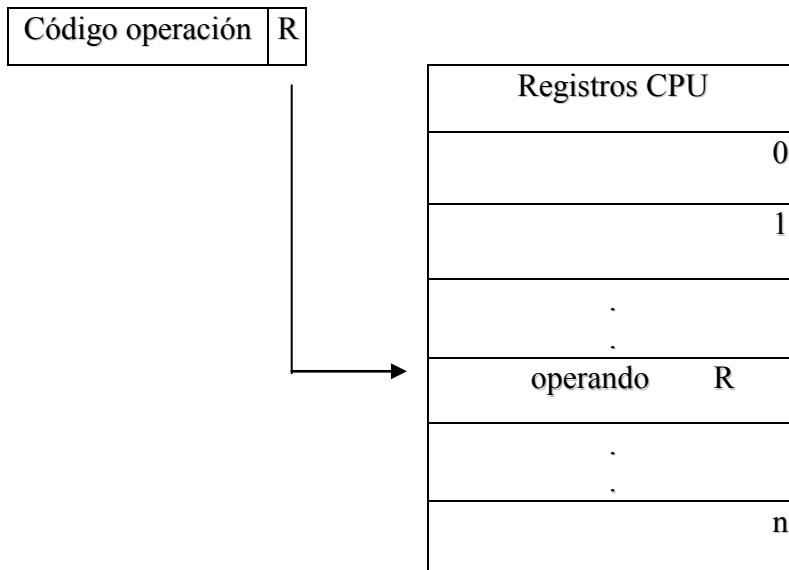
## DIRECCIONAMIENTO IMPLICITO o INHERENTE

Se refiere en general a instrucciones que utilizan un solo byte. El operando asociado a la Instrucción está contenido en un registro de CPU, y por ello resulta implícito en el código de operación y no es necesario hacer referencia a ninguna dirección. El código de operación identifica a la operación a ejecutar y el o los registros que intervienen (generalmente se refieren al acumulador y/o registros de uso general).

Ejemplo: Incrementa el acumulador A en 1.

$(AC) + 1 \rightarrow (AC)$

### Diagrama esquemático



Las ventajas de este tipo de acceso son:

- a- rapidez, el operando está en CPU y no se requieren ciclos de búsqueda de direcciones u operandos en memoria.
- b- basta una sola palabra para especificar toda la instrucción (economía en la memoria de programa).

En general todos los MP tienen este modo de direccionamiento.

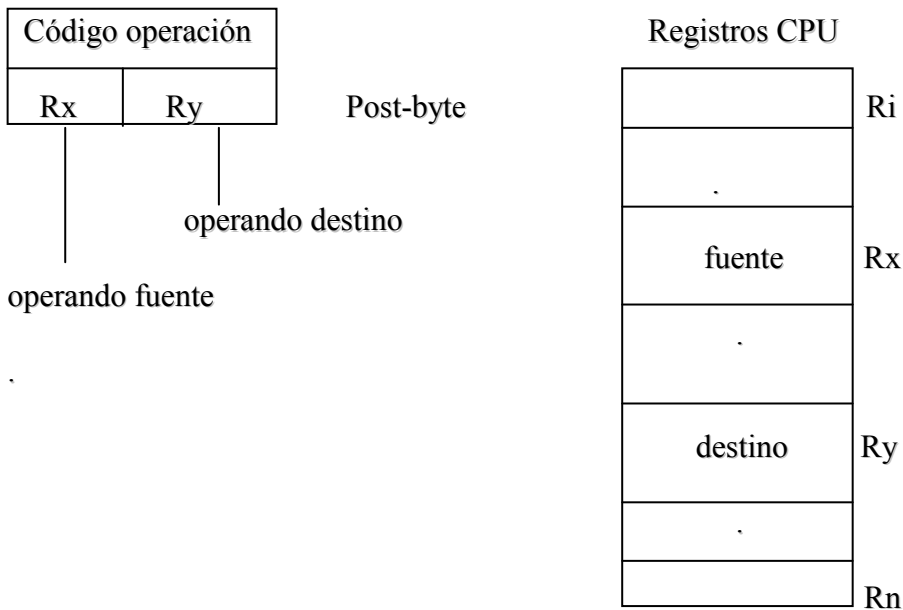
### **MODO REGISTRO**

Este modo de direccionamiento se puede interpretar como una extensión del anterior. Ya que el o los datos están en registros de CPU. Este modo lo poseen aquellos procesadores con una gran cantidad de Registros y/o de Modos de Direccionamiento, por ejemplo el 6809, en el cual con un byte no es suficiente para indicar la operación y el o los registros que intervienen, por lo tanto se hace necesaria la utilización de un post-byte para individualizar el o los Registros.

Ejemplo: (Reg.X) >(Reg.Y)

Transfiere el contenido de un Registro (fuente) a otro Registro (destino); el código de operación identifica que la operación a realizar es una transferencia, en el 2do byte se especifica cual es el Registro fuente y cual es el destino.

## Diagrama esquemático.



## **DIRECCIONAMIENTO INMEDIATO**

El operando forma parte de la instrucción es decir la constante especificada a continuación del código de operación es el dato, algunos procesadores permiten solo un byte para el operando (por ejemplo el SC/MP y el 6502). La gran mayoría de MP permite 1 o 2 byte dependiendo si el Registro que interviene es de 8 o 16 bits (8080, 6800, 6809, PDP-11, etc). Los microprocesadores Z8000 y 68000 de 16 bytes pueden tener operandos inmediatos de hasta 4 bytes de longitud, (palabra de longitud doble). El programador utiliza este modo de direccionamiento cuando necesita:

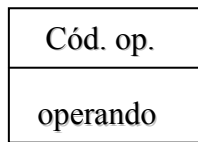
- \* constantes para inicializar variables.
- constantes para operar con variables mediante operaciones aritméticas lógicas.

Ejemplo:  $(AC) + 5 \longrightarrow (AC)$

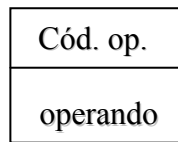
- \* Direcciones o datos constantes para ser comparados con variables.
- \* Direcciones fijas a cargar en Registros para luego usar en otros modos de direccionamiento.

El direccionamiento inmediato no es realmente una característica necesaria de la arquitectura de un procesador, ya que las constantes pueden ser almacenadas en lugares arbitrarios de memoria y acceder a ellos usando direccionamiento directo o algún otro.

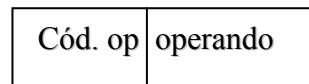
Algunos procesadores proveen operandos Inmediatos de 8 bits que son extendidos a 16 o 32 bits manteniendo el signo (sign-extended). El Z8000 y el 68000 proveen un modo inmediato corto, por ejemplo "Incrementar un registro en n", donde n es un valor de 4 bits contenido en la misma palabra que el código de operación.



A



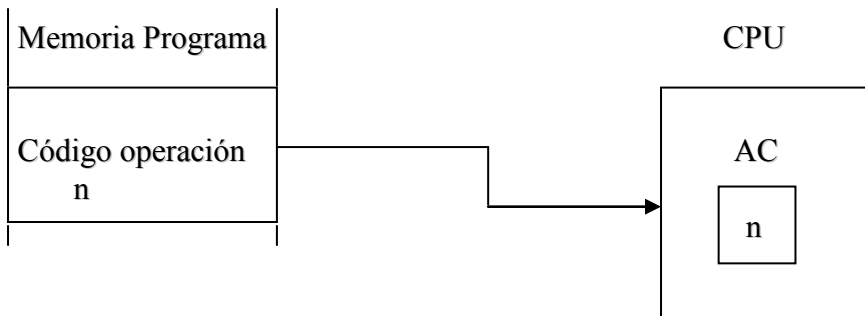
B



C

- A- Direccionamiento Inmediato. Longitud operando igual a la del código de operación.
- B- Direccionamiento Inmediato de longitud doble.
- C- Direccionamiento Inmediato corto.

Ejemplo:



Código de operación: Carga en forma inmediata el Acumulador.  
 n → (AC)

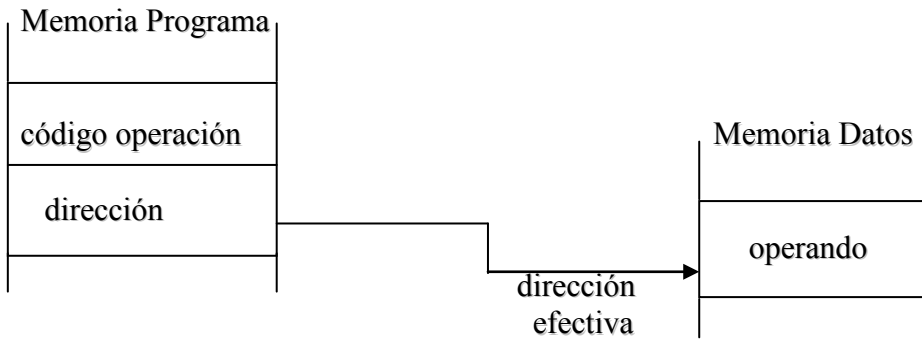
## DIRECCIONAMIENTO ABSOLUTO

Este modo de direccionamiento también llamado por los fabricantes de microprocesadores como Directo Pleno o Extendido, es sin ninguna duda la forma más simple de especificar un lugar dentro de la memoria.

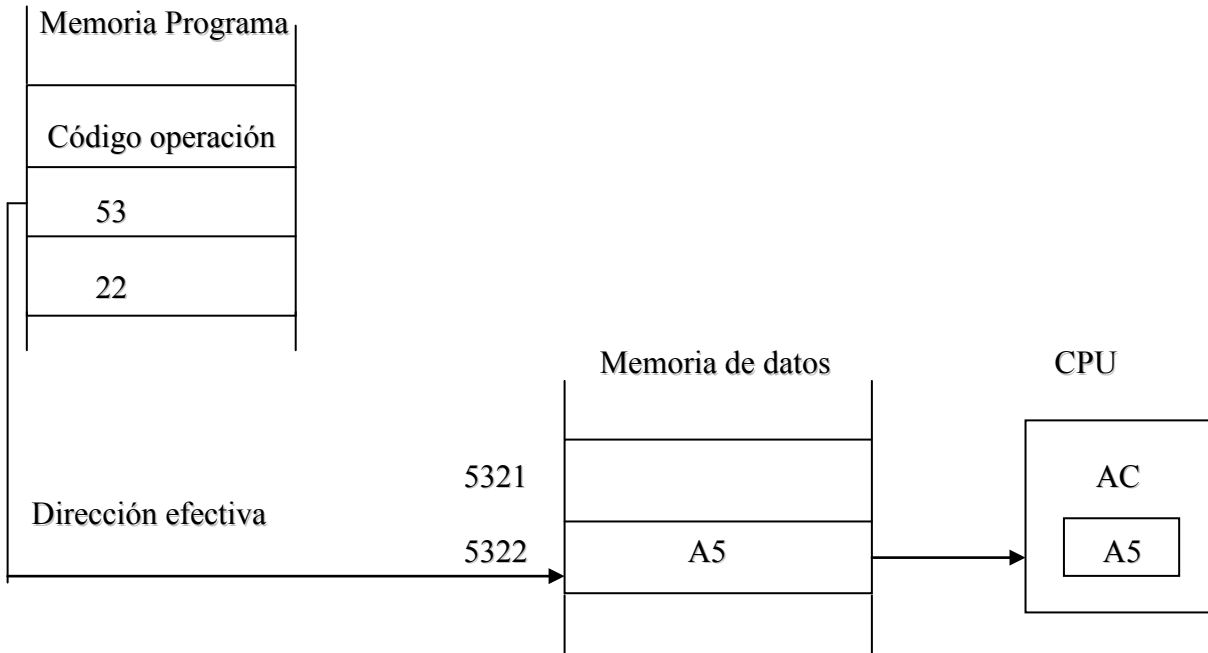
Una instrucción con este modo de direccionamiento está compuesta por el código de operación y a continuación la dirección del operando, directamente especificada en uno o más bytes según el alcance del campo de direcciones.

En general el modo absoluto tiene la desventaja de que ocupa muchos bytes. En MP con 16 líneas de direcciones ocupa 2 bytes y el número de bytes ocupados crece con la capacidad de direccionamiento del procesador, así por ejemplo el 68000 con 32 bits de direcciones debe ocupar 4 bytes, debido a esto, son necesarios varios ciclos de memoria para ejecutar la Instrucción.

Es evidente que el programador utilizará este modo de direccionamiento cuando conoce la dirección absoluta de la variable a procesar, sin conocer el valor de la variable.



Ejemplo: cargar el acumulador, con direccionamiento directo pleno.



En general todos los MP poseen este modo de direccionamiento (salvo el SC/MP).

**DIRECCIONAMIENTO DIRECTO PAGINADO**

Este modo de direccionamiento se deriva del modo directo pleno y tiene como finalidad eliminar por lo menos un byte de la instrucción y un acceso a memoria, lo que hace más práctica su utilización.

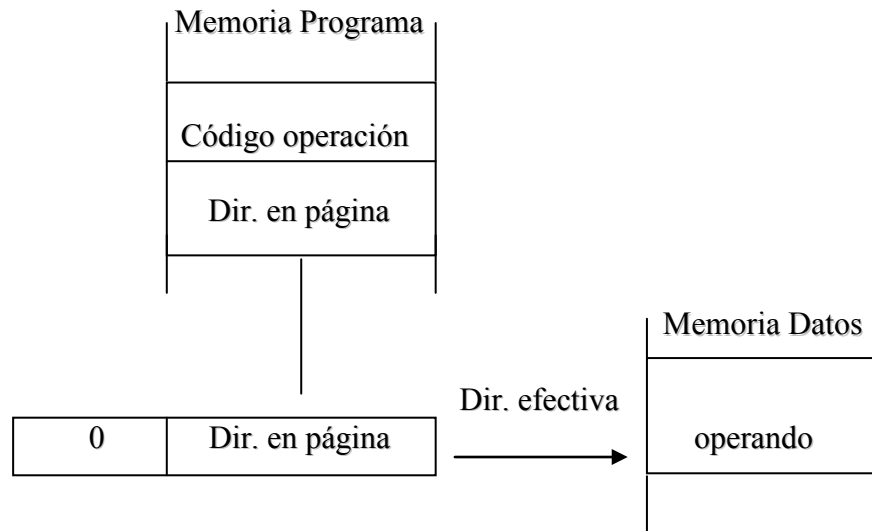
El modo se basa en que la memoria de un computador puede ser dividida en páginas, por ejemplo: en una memoria direccionada con 16 líneas (64 K) los 8 bits más significativos identifican una de las 256 páginas posibles y con los 8 menos significativos la ubicación dentro de la página, es decir que la instrucción deberá especificar la dirección dentro de la página.

Existen tres formas del Direccionamiento Directo paginado:

a)- Direccionamiento a página base.

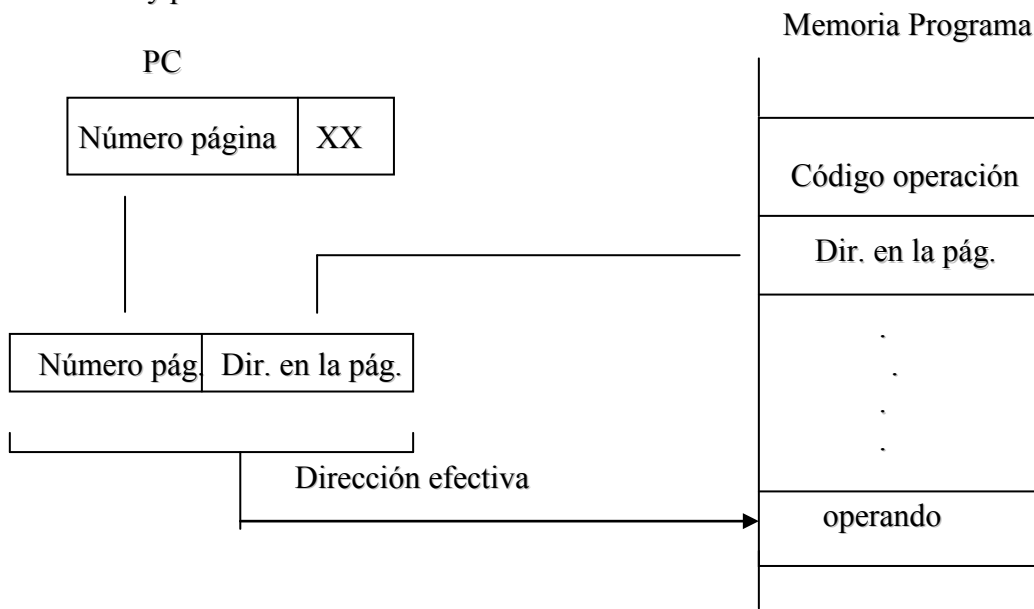
En esta forma del direccionamiento directo paginado, la página base es fija, generalmente la página cero, este tipo de direccionamiento es común a la familia del 6800 y a la del 6502. Estos fabricantes le llaman a este modo de direccionamiento directo y al directo pleno, extendido.

Es evidente, que para un mejor aprovechamiento de la máquina, en un sistema que posea este modo de direccionamiento, la memoria de datos (RAM) deberá estar en las direcciones bajas del mapeado de Memoria. El programador deberá utilizar siempre que le sea posible este modo de direccionamiento y no el extendido.



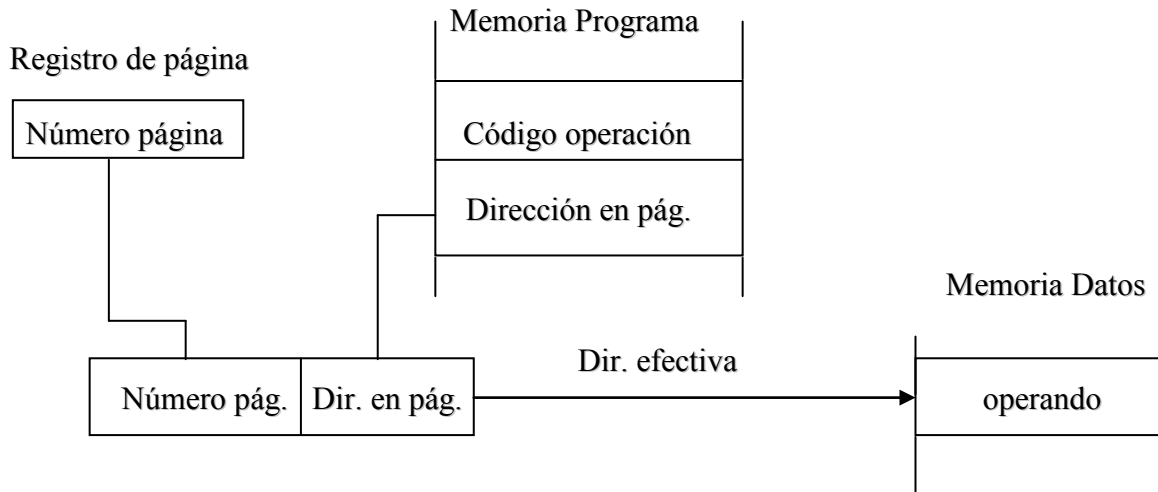
b)- Direccionamiento a página presente

También llamado a página corriente, en este caso el número de la página está definida en la parte alta del PC, con este modo se lo puede direccionar en la misma página en la que se encuentra la Instrucción, este modo es muy poco utilizado.

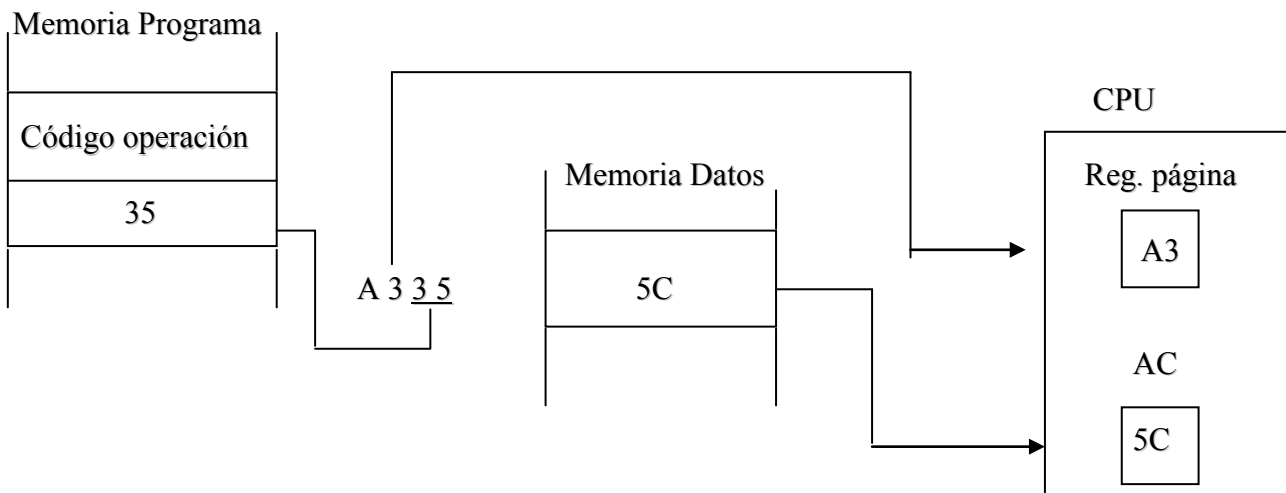


### c)- Direccionamiento con Registro de Página

Como en el direccionamiento directo a página cero, en la instrucción solo se especifica la dirección dentro de la página. La identificación de la página (número) está definida por un registro especial de CPU llamado puntero de página, el que debe ser cargado previamente por programa. El procesador 6809 de Motorola posee este modo de direccionamiento. Este modo de direccionamiento puede ser utilizado por el programador para obtener programas cuyos datos sean independientes de la dirección, con la condición de que el Registro Puntero de página sea cargado previamente con un valor que depende de la zona del programa en ejecución.



Ejemplo: cargar el AC directo con Registro de Página.



### **DIRECCIONAMIENTO INDIRECTO VIA REGISTRO**

En este modo, también llamado modo Registro Indirecto, la dirección efectiva del operando está contenida en un Registro simple o par (según el procesador) de la CPU, la Instrucción; con este modo de direccionamiento, está formada por una sola palabra (código de operación), el cual indicará la operación a realizar y el registro que contiene la dirección del operando. Se debe recordar que previamente se debe cargar el registro puntero con una determinada dirección.

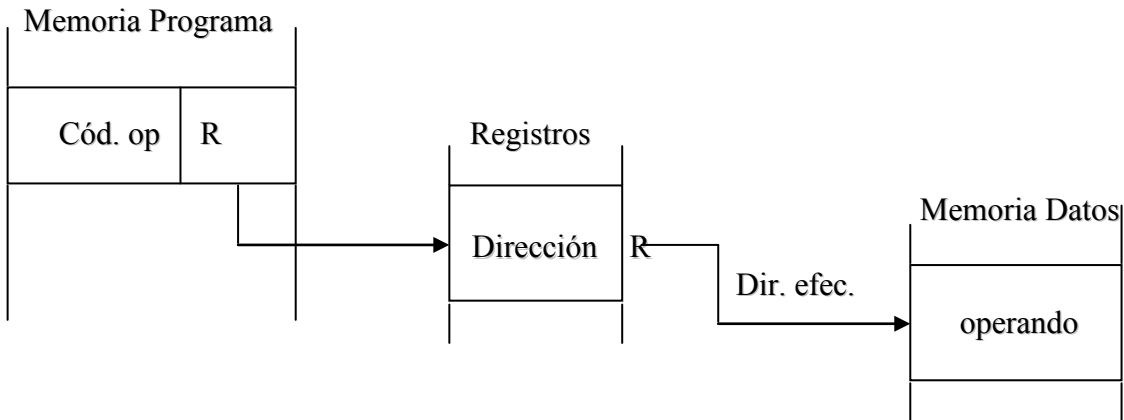
La ventaja del uso de este modo, como en general de todos los Indirectos o Diferidos, no es precisamente hacer más rápido el acceso a operandos, sino que en el manejo de tablas de datos con muchas entradas, generalmente la dirección absoluta del operando no es conocida en el momento en que el programador



realiza el programa sino que debe ser calculada durante su ejecución. Como no es práctico modificar palabras de la Instrucción (imposible pues están en ROM), se usa algún lugar de Memoria o Registro, como en este caso, para resolver el problema.

El MP 8080 y 8085 poseen este modo de Direccionamiento cuando se refiere al lugar de Memoria Direccionado por alguno de sus registros pares, también en el MP 6809 y en el microcomputador 68705 en su modo llamado Indexado sin desplazamiento.

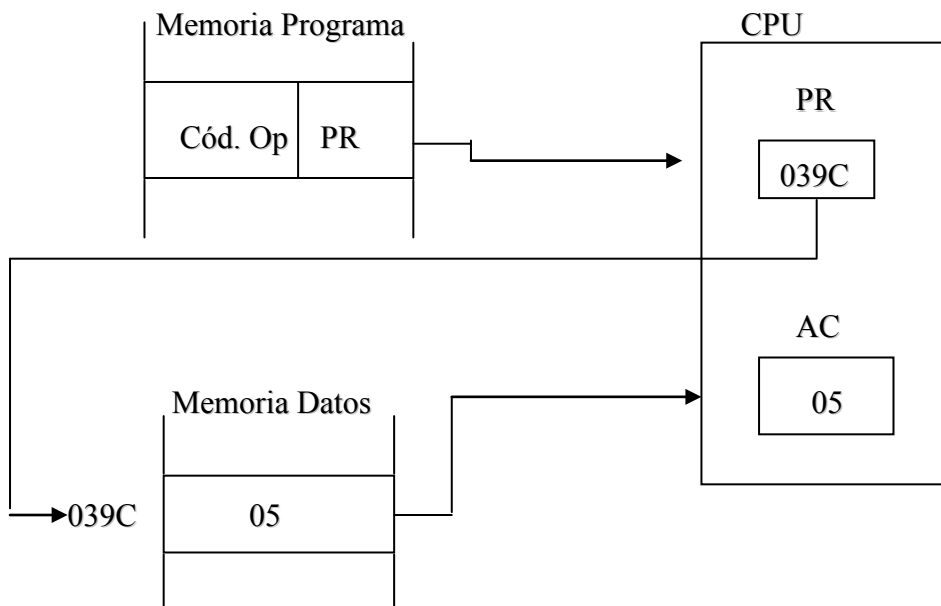
El Procesador del minicomputador PDP-11 también posee este modo y en general la mayoría de los procesadores de 16 bits.



Ejemplo: carga del acumulador, con direccionamiento Indirecto vía Registro puntero (PR)

$((PR)) \longrightarrow (AC) ; (PR)=039C$

$((PR)) = (039C) \longrightarrow (AC)=05$



## INDIRECTO VIA REGISTRO CON AUTO INCREMENTO Y AUTO DECREMENTO

Se dijo que el Modo Indirecto vía registro se utiliza para manejar tablas, listas de datos etc. Generalmente en esta tarea se direcciona cada dato con el registro y luego es necesario saltar al siguiente, incrementando o decrementando el puntero con una Instrucción, con el auto incremento o auto decremento la actualización del puntero en la tabla se hace, como su nombre lo indica, en forma automática. Este

Incremento o Decremento depende de la longitud del operando, por ejemplo será de 1 para apuntar al siguiente byte, de 2 en el caso de palabra (16 bits) y de 4 en el caso de palabra doble (32 bits). La PDP-11, 6809 y TI 9900 permiten solo por 1 o 2, el 68000 es más flexible y permite por 1, 2 y 4. En el procesador SC/MP (uno de los primeros y más simples procesadores) se permiten auto Incrementos y decrementos de hasta 8 bits en representación Complemento 2 (-127 a + 127), a este modo su fabricante (National) lo llama auto Indexado.

La diferencia entre el auto Incremento y el auto decremento es la siguiente:

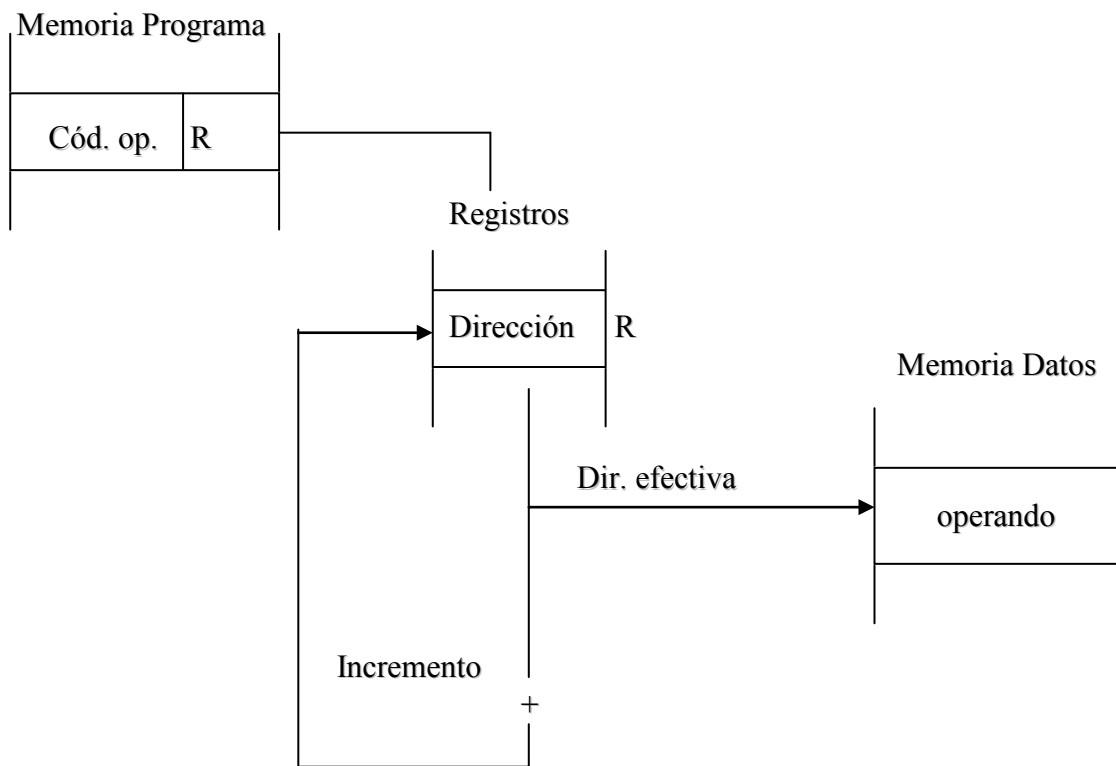
En el auto incremento la dirección efectiva es el contenido del puntero, una vez obtenido el dato se actualiza el puntero. En el auto decremento se invierte el orden de las operaciones, primero se actualiza el puntero para recién obtener la dirección efectiva.

Esta es la forma de trabajo del Stack Pointer para su empleo en pilas tipo LIFO (último en entrar, primero en salir).

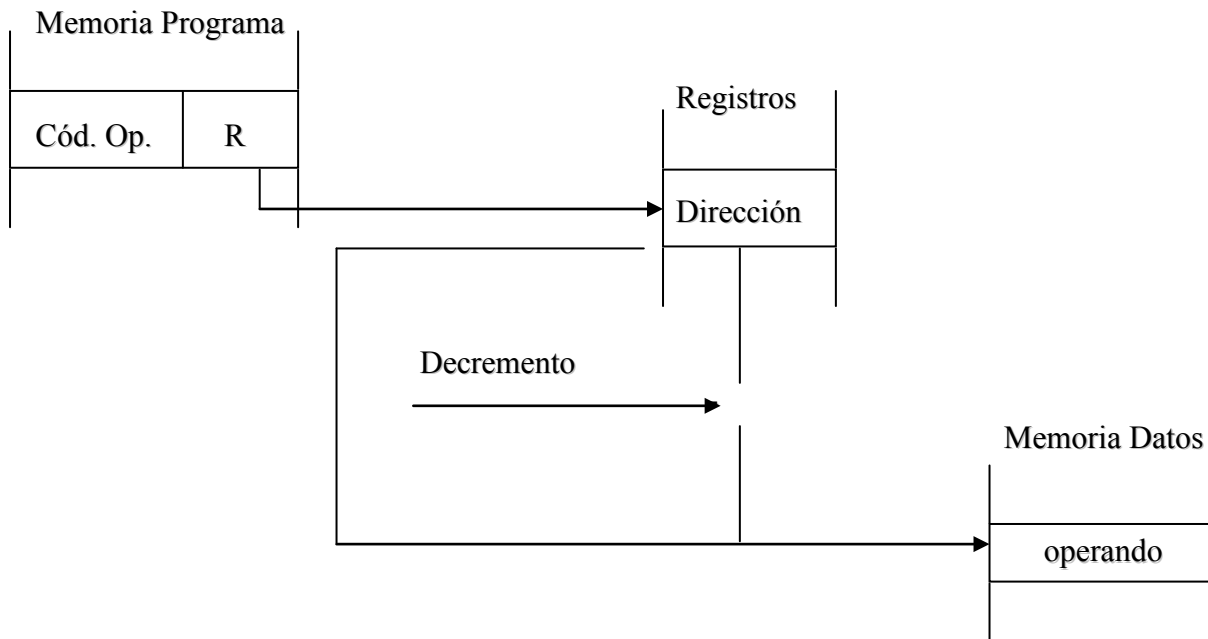
En muchos casos se enfatiza en Lenguajes Assemblers el auto pre-decremento y el auto post-incremento por ejemplo en el 6809.

Auto-post incremento.

STA - (X)  
STA (X) +



### Auto pre-Decremento.



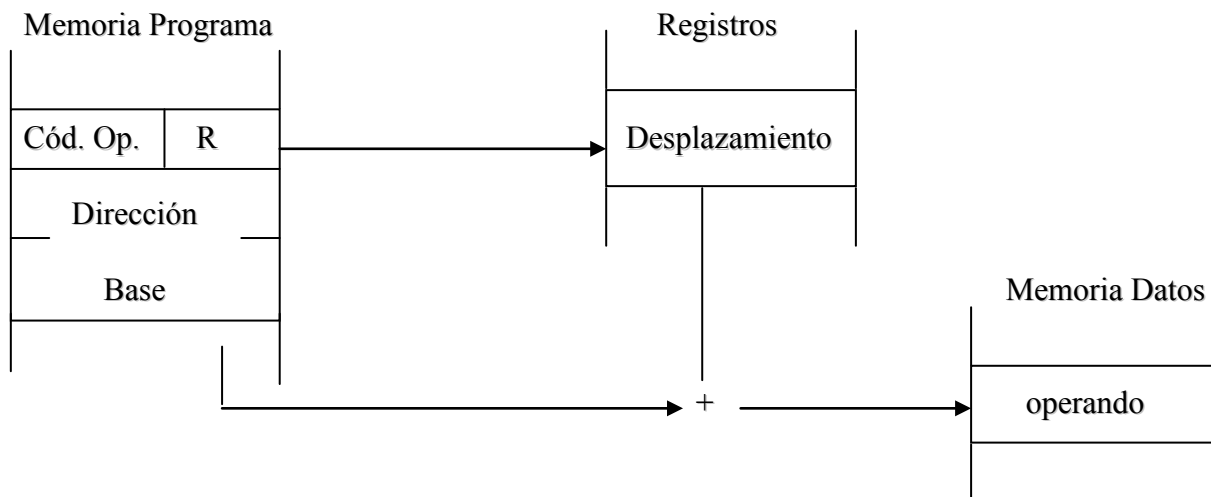
### **DIRECCIONAMIENTO INDEXADO**

El modo de Direccionamiento Indexado, utiliza dos elementos que deben sumarse para encontrar la Dirección efectiva. La instrucción se forma con el código de operación, seguido por una Dirección Base, como en el direccionamiento absoluto o extendido. Esta Dirección Base es sumada con un desplazamiento especificado en un registro para obtener la Dirección efectiva.

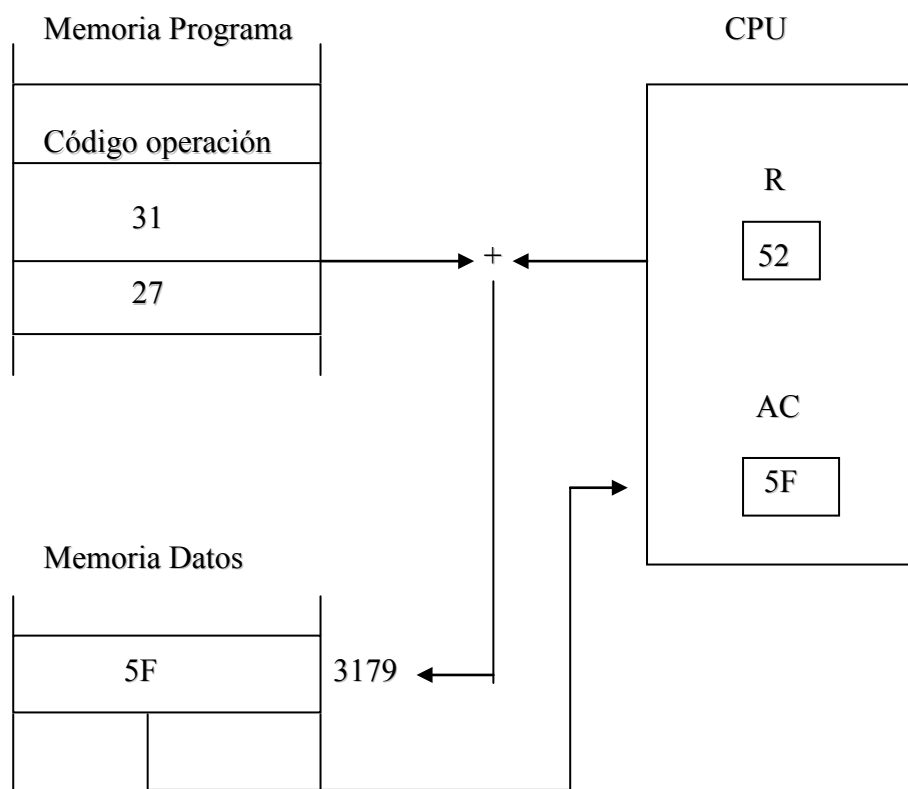
Cabe la siguiente aclaración al lector con algunos conocimientos sobre microprocesadores, muchos fabricantes; la gran mayoría, denominan Direccionamiento Indexado a uno que no cumple con la definición dada anteriormente. En uno de los próximos temas (Direccionamiento Base) se aclarará más sobre esta disparidad de criterios.

El Direccionamiento Indexado es el apropiado para acceder a arreglos y tablas, en este caso la Dirección Base explícita en la Instrucción corresponde a la dirección base del Arreglo, mientras que el valor contenido en el Registro, corresponde al desplazamiento (Indexación) del componente en el arreglo. Este modo de direccionamiento lo poseen la mayoría de los últimos procesadores, por ejemplo el 6502, 68705, Z8000, etc.

El desplazamiento puede ser un número con signo o sin signo, según el procesador, cuando es con signo, para realizar la suma de un número de 8 bit con uno de 16 debe pasar el número de 8 bits a 16 manteniendo el signo (esto es: los bits 8 a 15 del número ampliado deben ser iguales al bit 7 del número original de 8 bits).



Ejemplo: cargar el acumulador, con Direccionamiento Indexado.



### **MODO DE DIRECCIONAMIENTO BASE O CON DIRECCION BASE**

Este modo de direccionamiento es similar al Indexado y frecuentemente se confunden, especialmente en las denominaciones dadas por los fabricantes.

Una dirección base es siempre una dirección plena, de igual longitud que el bus de direcciones, mientras que el desplazamiento puede ser corto o largo 8 a 16 por ejemplo.

Si recordamos que en el Direccionamiento Indexado la instrucción contenía la Dirección Base y un Registro contenía el desplazamiento, el Direccionamiento Base es precisamente lo opuesto, la Instrucción contiene un desplazamiento y un registro de direcciones (registro base) contiene la dirección base.

De esto se pueden sacar algunas conclusiones:

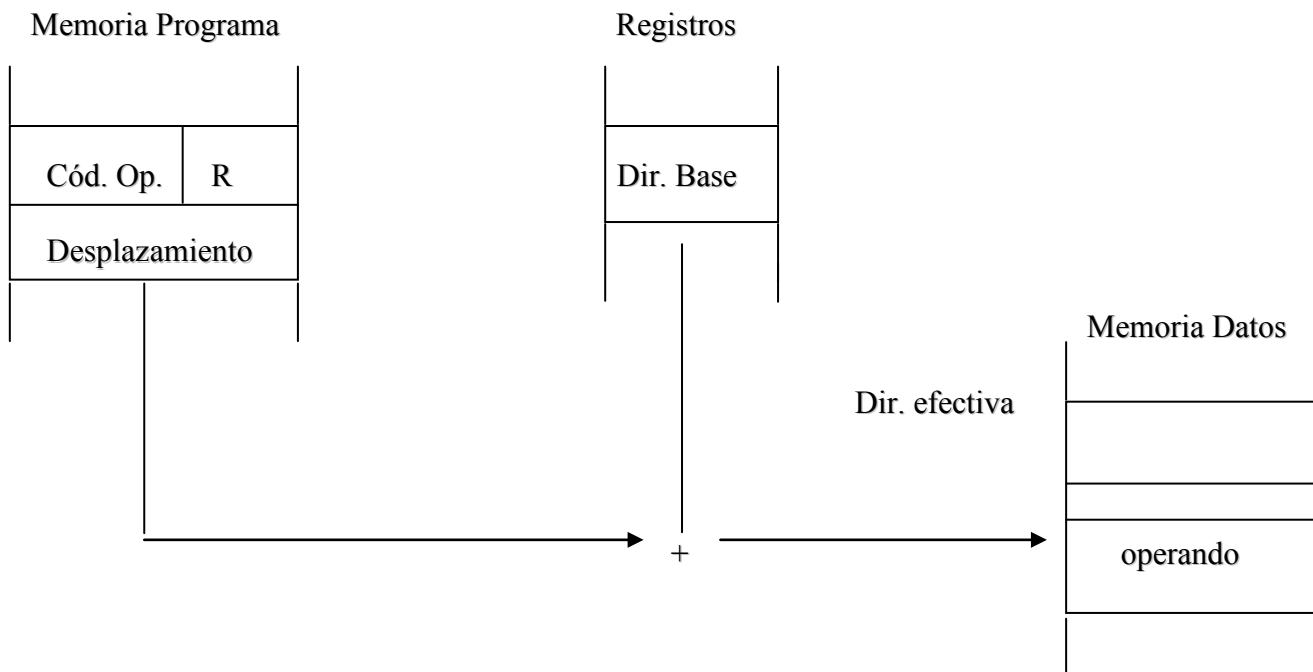
a)- Si el desplazamiento y la Dirección Base son de igual longitud, los dos modos Indexado y Base no se pueden distinguir, por ejemplo PDP-11, 6809 (con desplazamiento largo) etc.

b)- La diferencia entre los dos modos se manifiesta cuando son usados desplazamientos cortos; por ejemplo el 6809 con desplazamiento corto.

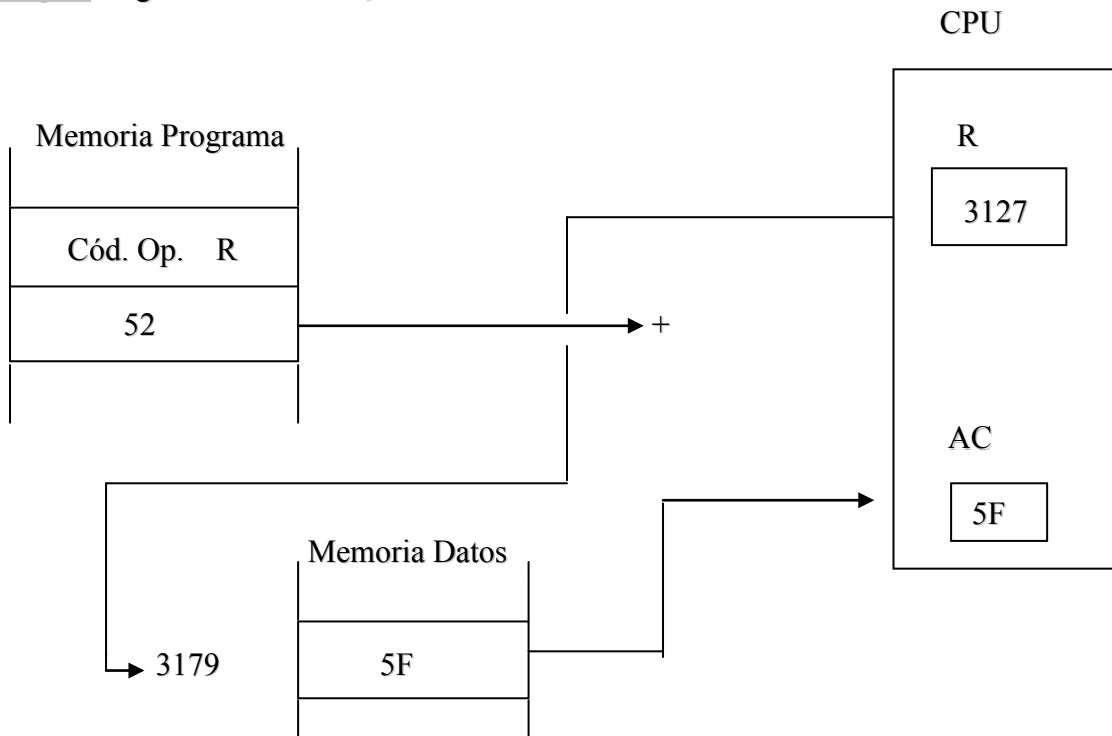
c)- Los direccionamientos llamados Indexados por los fabricantes de los Microprocesadores son realmente con Dirección Base, por ejemplo SC/MP, 6800, Z8000, 6808 con 5 y 8 bit de desplazamiento.

d)- En el procesador Z8000 que posee ambos modos, el Indexado y el Base, se observa que como el desplazamiento y la Dirección Base son de 16 bits no se distinguen, pero ocurre que tienen distintos códigos de operación aunque la operación realizada sea la misma.

e)- Con respecto al signo del desplazamiento es válido lo expresado para direccionamiento Indexado.



Ejemplo: cargar el acumulador, con Direccionamiento Base.



El direccionamiento Indexado es usado cuando la Dirección Base de una estructura de datos es conocida en el momento en que el programador realiza el programa, pero el acceso a un elemento cualquiera se debe realizar cuando el programa se está ejecutando. Recordemos que en el Direccionamiento Indexado la Dirección Base forma parte de la Instrucción.

El direccionamiento base es usado cuando la posición relativa de un elemento en la estructura de datos es conocida cuando se realiza el programa, pero la dirección de comienzo de la estructura no está definida sino que se fijará durante su ejecución.

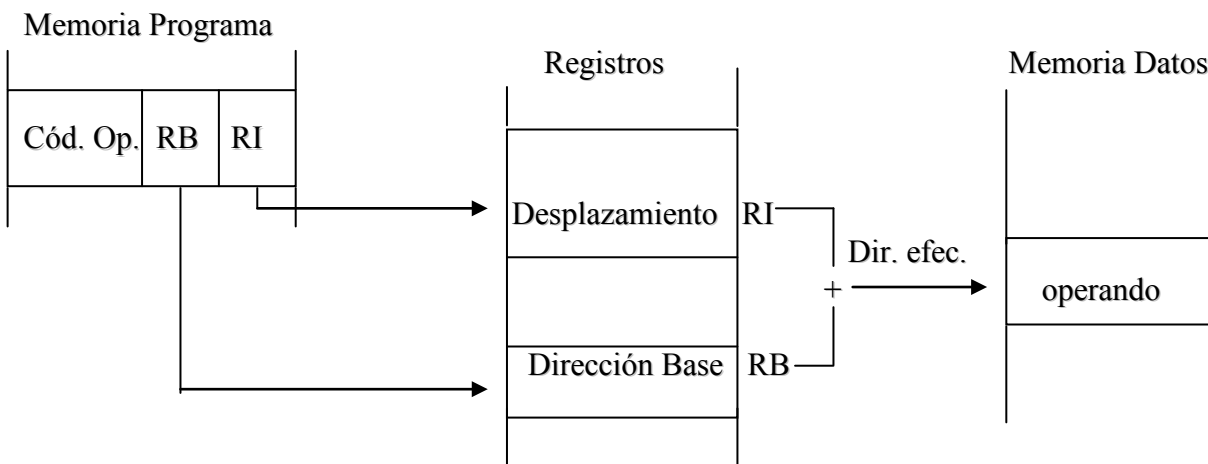
Entre las aplicaciones típicas del Direccionamiento Base podemos citar:

- \* acceso a datos en una zona de parámetros cuando la dirección base es transferida a una subrutina.
- \* acceso a datos y bifurcaciones a direcciones en un programa independiente de la posición, cuando el registro base es cargado con la dirección comienzo del programa.
- \* acceso a un campo particular de un bloque completo, cargando el registro base con la dirección de comienzo del bloque.

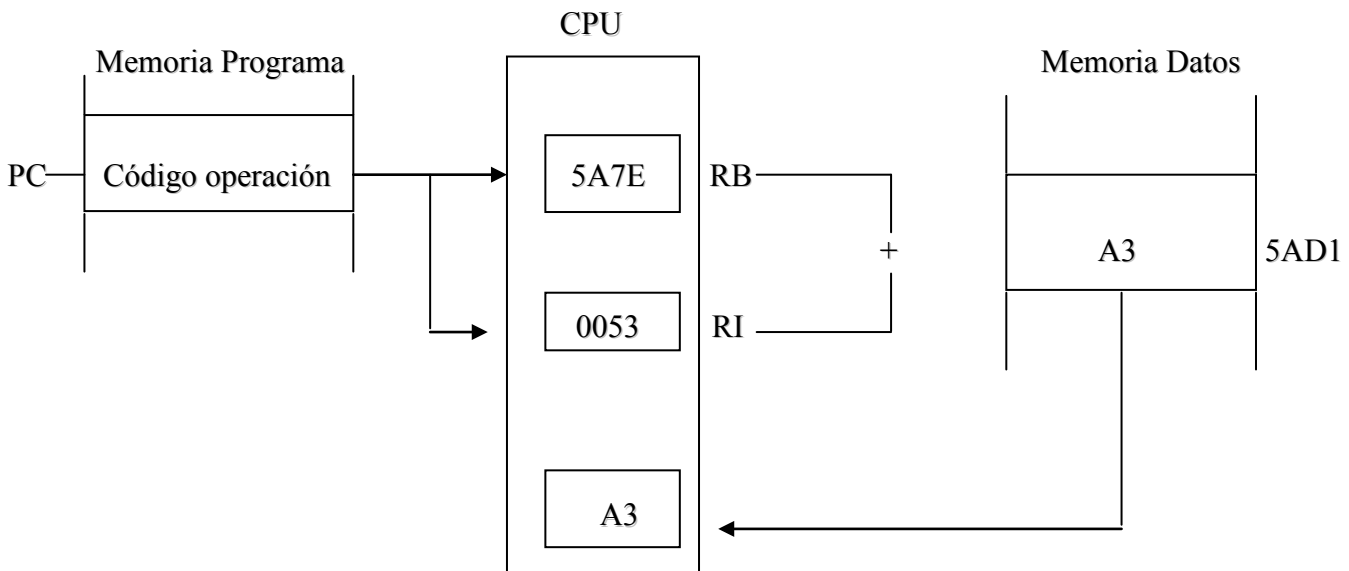
## DIRECCIONAMIENTO BASE-INDEXTADO

Este modo de direccionamiento obtiene, como los anteriores, la dirección efectiva como la suma de una Dirección Base y un desplazamiento, con la diferencia de que en este caso ambos, desplazamiento y Dirección Base, están en Registros de CPU, de esta forma la Instrucción está formada solo por el Código de operación. Por ejemplo el 6809 usa un Registro de 16 bits como Registro base (llamado por el fabricante Índice), y permite un desplazamiento de 8 bits del Acumulador (A o B) o un desplazamiento de 16 bits del Acumulador Doble (A y B). El 68000, el 8086 y el Z8000 también poseen el modo de Direccionamiento Base Indexado.

Este modo permite que tanto la dirección base de un arreglo como el desplazamiento de un elemento puedan calcularse durante la ejecución del programa.



Ejemplo: cargar el Acumulador usando Direccionamiento Base Indexado.

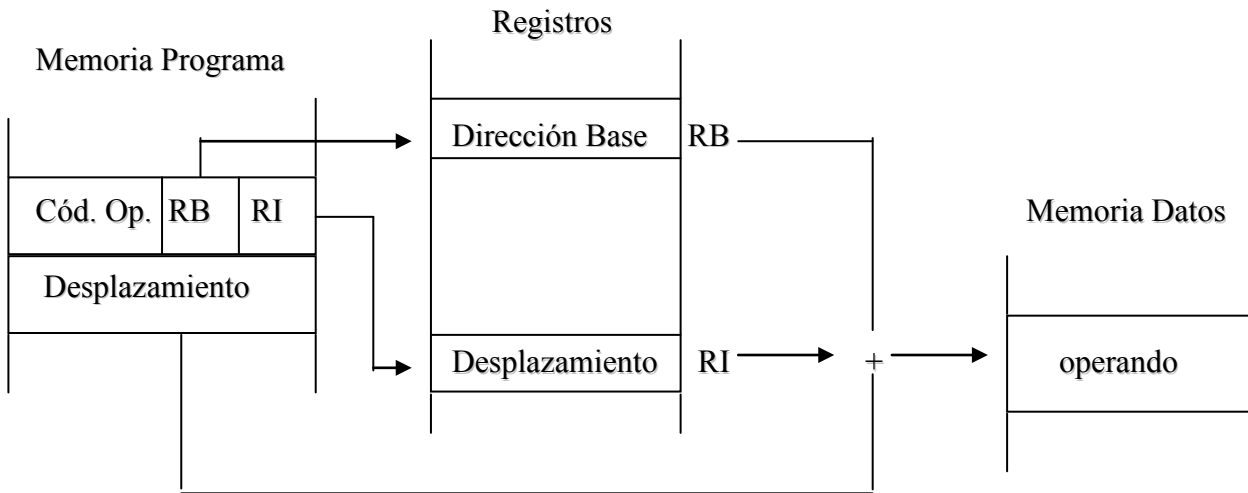


## BASE INDEXADO CON DESPLAZAMIENTO

Este es una extensión del modo anterior y lo poseen los procesadores 8086 y 8088. En este modo la Dirección efectiva se obtiene como la suma de

- Una Dirección Base contenida en un Registro Base.
- Un desplazamiento contenido en un Registro Índice.

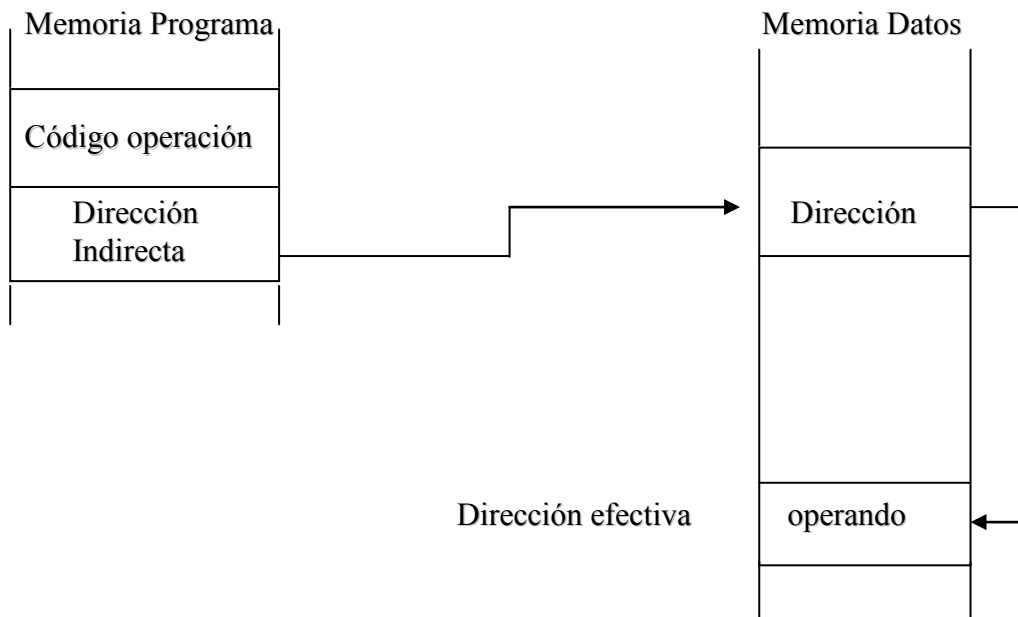
c) Un desplazamiento que sigue al código de operación en la Memoria de Programa.



### DIRECCIONAMIENTO EXTENDIDO INDIRECTO

La dirección que figura en la Instrucción, a continuación del código de operación, es la dirección de Memoria donde se encuentra la dirección del dato. En otras palabras se le ha agregado un grado de Indirección al Direccionamiento Extendido, también pueden existir varios niveles de Indirección (por ejemplo el microprocesador MOSTEK MK 5065P).

El Direccionamiento con un solo grado de Indirección lo poseen el 6809 y el 6502 aunque éste solo para las instrucciones de bifurcación.



Este modo es utilizado cuando se deben manejar arreglos sin conocer la dirección absoluta de él, pero conociendo una dirección (Dirección Indirecta) de Memoria donde se encuentra almacenada su dirección absoluta.



## DIRECCIONAMIENTO BASE INDIRECTO

También puede llamarse Base Diferido, aunque los fabricantes comúnmente lo denominaban Indexado Indirecto debido a la incertidumbre entre el Direccionamiento Base e Indexado explicada anteriormente.

Algunos procesadores permiten agregar un nivel de Indirección a los Direccionamientos Base. Dentro de los Indexados Diferidos podemos diferenciar dos tipos:

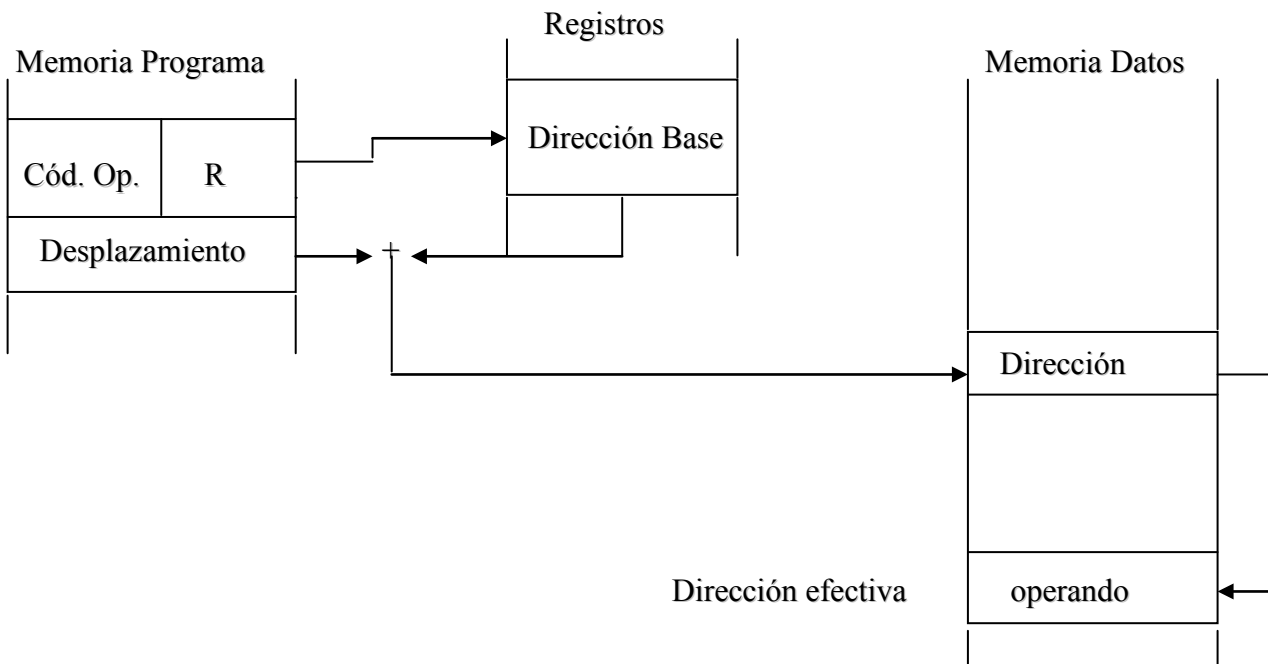
a- Direccionamiento Pre-Indexado Indirecto.

b- Direccionamiento Post-Indexado Indirecto.

La diferencia entre un modo y otro se basa en considerar a quien se le debe sumar el desplazamiento, si al Registro Base o al contenido de la dirección indicada en dicho Registro.

## DIRECCIONAMIENTO PRE-INDEXADO INDIRECTO

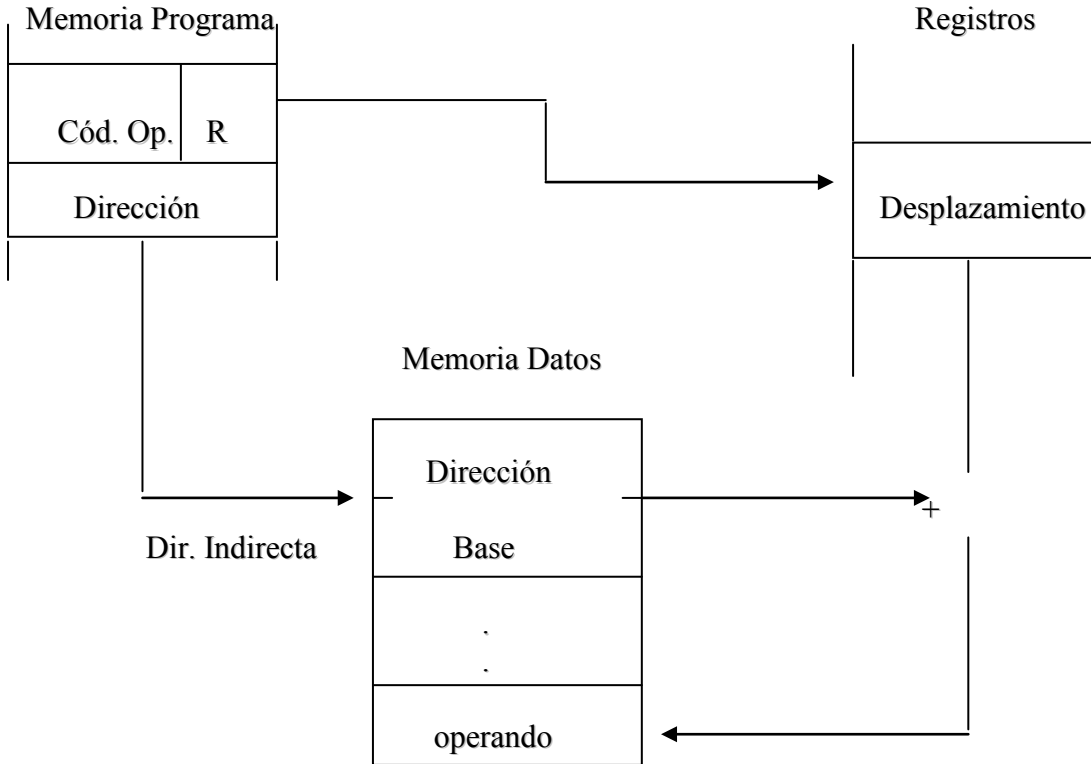
En este modo el Desplazamiento especificado como segundo byte de la Instrucción es sumado al contenido del Registro base para obtener la Dirección Indirecta donde se encuentra almacenada la dirección del operando.



Los procesadores 6809 permite agregarle este modo Indirecto a alguno de sus Modos Base o Indexado. La minicomputadora PDP-11 también posee este modo, así como el procesador 6502.

## DIRECCIONAMIENTO POST-INDEXADO INDIRECTO

Uno de los únicos procesadores que posee éste Modo de direccionamiento es el 6502, éste posee un Registro puntero de 8 bit y en la instrucción el byte que sigue al código de operación es también de 8 bits por lo que en este caso no existe diferencia entre Indexado o Base. Por el modo de operar, considera que la Información que trae la Instrucción es una dirección en la página cero, se trata de un Direccionamiento Indexado Indirecto. La Información contenida en el segundo byte de la Instrucción contiene una dirección de los primeros 256 bytes de Memoria. Esta Dirección y la siguiente contienen una dirección la cual es sumada al contenido de un Registro Índice para obtener la dirección efectiva.



Las diferencias entre el Modo de Direccionamiento Pre-Indexado Indirecto y el Post-Indexado Indirecto son las siguientes:

- 1- En el Pre-Indexado Indirecto se realiza primero la Indexación y luego la Indirección mientras que en el otro la Indirección se realiza antes de la Indexación.
- 2- El Modo de Direccionamiento Pre-Indexado Indirecto es ventajoso utilizar para elegir una de varias direcciones Indirectas a usar.
- 3- El Modo de Direccionamiento Post-Indexado Indirecto es útil cuando se debe acceder a un arreglo o tabla cuya dirección base es obtenida en forma indirecta.
- 4- En el caso del 6502 que tiene 2 Registros punteros de 8 bits cada uno X e Y el pre-indexado Indirecto usa el X mientras que el post-Indexado Indirecto usa el Y.

## MODO DE DIRECCIONAMIENTO RELATIVO

Este modo es similar al Base lo único que el Registro Base utilizado es el contador de Programa (PC).

En éste modo se obtiene la dirección efectiva como la suma de un desplazamiento contenido como segundo byte de la Instrucción con el contenido del PC.

En general existen dos tipos de direccionamientos Relativos: 8 y 16 bits de desplazamiento respectivamente.

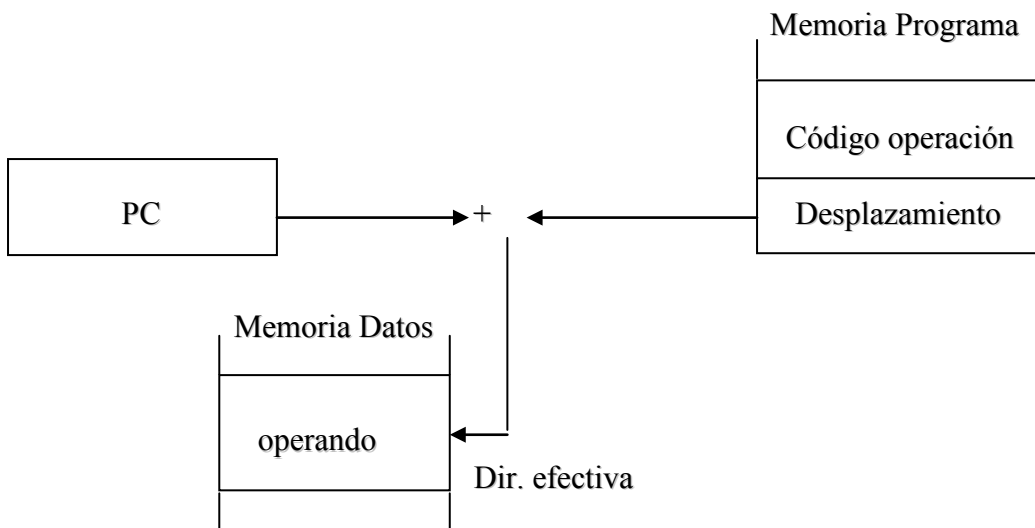
En el caso de desplazamiento de 8 bits contenido en la Instrucción éste en algunos procesadores es extendido a 16 bits conservando el signo, lo cual permite pasar de una página a otra sin ninguna dificultad caso del 6800, 6809, 8086, Z80, etc.

En el caso de Microprocesador SC/MP permite desplazamiento de 8 bits pero no es posible cambiar de página ya que la suma se realiza solo en los 8 bits menos significativos sin tener en cuenta el acarreo sobre la parte alta de la dirección.

Algunos procesadores; caso 6809, PDP-11, 6502, 8086, entre otros, permiten desplazamientos de 16 bits por lo tanto se puede mover dentro de los 64 K mientras que en el caso de 8 bits de desplazamiento solo se puede mover -128 a +127.

El uso más común de éste modo de direccionamiento es para las instrucciones de salto y bifurcación, aunque en algunos procesadores, caso 8086, 6809, PDP-11 y SC/MP este modo puede utilizarse con cualquier otra instrucción, por ejemplo, Instrucciones aritméticas y lógicas.

Este modo de Direccionamiento permite implementar programas independientes de la posición, Programas Re-ubicables.

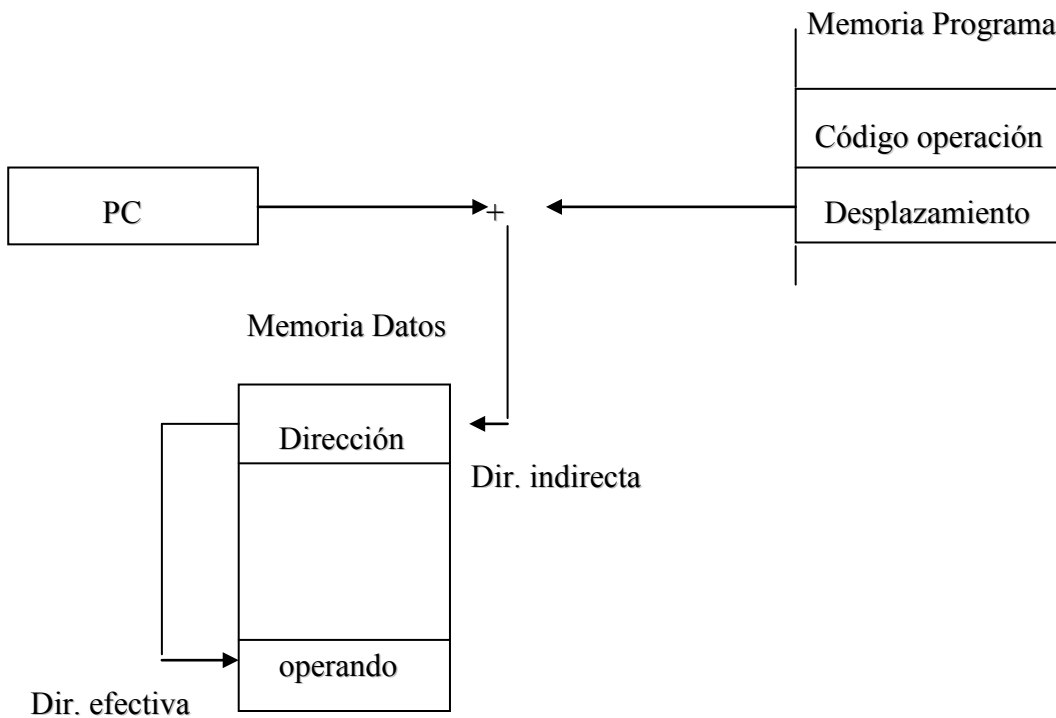


## DIRECCIONAMIENTO RELATIVO INDIRECTO

El modo de direccionamiento Relativo Indirecto o Relativo diferido, se basa en adicionar un orden de indirección al direccionamiento Relativo.

En éste modo la dirección obtenida como suma del contador de programa y el desplazamiento contenido en la Instrucción es la dirección Indirecta, donde se encuentra almacenada la dirección efectiva.

Solo algunos procesadores, por ejemplo el 6809 y PDP-11 permiten este modo de direccionamiento.

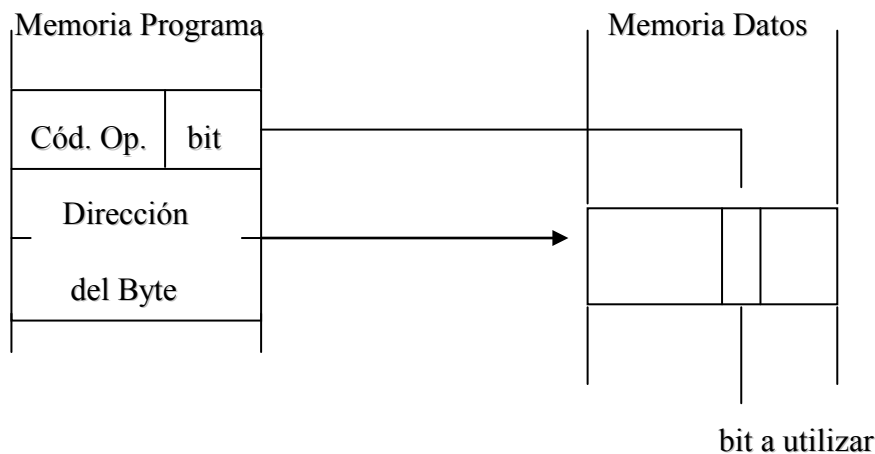


## DIRECCIONAMIENTO DE BIT

Algunos procesadores caso Z80 y la familia de los microcomputadores del 68705 poseen instrucciones con las cuales es posible acceder al bit.

Estas Instrucciones de Manipulación de bits utilizan una combinación de Modos de Direccionamiento, Directo, Registro, Registro Indirecto o Indexado para individualizar la posición de memoria o el registro donde se encuentra el byte a considerar y un código de 3 bits dentro del código de operación de la Instrucción especifica el bit en cuestión.

Estas Instrucciones tienen como ventaja un mejor aprovechamiento de la memoria, tanto la del programa como la de datos, ya que con una sola Instrucción el programador podrá colocar en 1 o 0 un determinado bit o realizar un salto si ese bit está en 1 o 0. El programador se dará cuenta que si no posee este modo de acceso al bit necesitaría 3 instrucciones del programa, visto está el aprovechamiento de la memoria RAM ya que el programador no tendrá ninguna dificultad en utilizar como bandera a un bit y no desaprovechar la memoria usando un byte como bandera. Supongamos un acceso directo pleno al bit.



## DIRECCIONAMIENTO DE INSTRUCCIONES

Cuando la Instrucción en cuestión es de ruptura de la secuencia, condicional o incondicional, se debe especificar de algún modo con que nueva dirección se debe cargar el contador de Programa en caso de que se realice la transferencia en el Control de Programa (salto o llamado a subrutina).

Los tipos de acceso a la Instrucción son semejantes a los accesos a operando, ya en algunos ejemplos se adelantó el direccionamiento en las Instrucciones de Bifurcación. La única diferencia que se puede citar es que en el caso de direccionamiento de operandos la dirección efectiva era la dirección del operando en este caso la dirección efectiva es la dirección de la próxima Instrucción a ejecutar. En el caso de direccionamiento de Instrucciones algunos tipos de direccionamientos vistos no tienen sentido, por ejemplo el Direccionamiento Inmediato, el Direccionamiento de bit, los auto incremento y decremento, siendo los modos de Direccionamiento más usados el Directo pleno, Directo a página presente, Indexados, Base, Indirecto, Relativo, Relativo Indirecto etc.

## CONJUNTO DE INSTRUCCIONES DE UN MICROPROCESADOR

Cada microprocesador o familia de microprocesadores posee un conjunto de Instrucciones, llamado por el fabricante como set de Instrucciones, este conjunto de Instrucciones está definido por el fabricante.

Estas Instrucciones no son otra cosa que las distintas operaciones que es capaz de realizar el procesador, de las cuales el usuario elegirá las necesarias para elaborar una serie de operaciones sucesivas que constituyen el programa.

Las Instrucciones de los procesadores se pueden clasificar de acuerdo a su naturaleza, en los siguientes tipos de Instrucciones:

- a- De Transferencia de Información.
- b- Aritmético-lógicas.
- c- De manejo de contenidos de Registros y Memoria.
- d- De control de Programa.
- e- De control de sistema.
- f- De entrada-salida.

Como se estudió previamente, cada Instrucción puede abarcar uno o mas bytes, ya que se debe especificar la operación y el operando, este último teniendo en cuenta el modo de direccionamiento utilizado. Vamos a dar una mayor generalidad al concepto de operando, distinguiendo entre operandos datos, los cuales son los datos propiamente dicho, y operandos de control que son a los que se hace referencia en las Instrucciones

de control de programa. Por lo tanto cualquier Instrucción en general es la formada por una configuración binaria para especificar el código de operación y otra para especificar al operando, aunque se tienen Instrucciones de un solo byte.

Es de hacer notar que para los procesadores de 8 bits, sobre los cuales basaremos nuestro estudio, cada uno de los bytes que forman una Instrucción es el contenido de una dirección de la Memoria de Programa, tanto las direcciones como sus contenidos se suelen expresar en código hexadecimal, la cual como se vio es una manera de compactar los números binarios. Sin embargo la codificación, en lenguaje absoluto (la única que comprende el sistema) es muy incómoda para el programador.

Para simplificar el proceso de escribir el programa, depurar los errores e interpretarlo con mayor facilidad se ha introducido la codificación simbólica o mnemotécnica, donde cada código de operación es representado por 3 o 4 letras. Estos símbolos se relacionan con la operación que se desea realizar. El fabricante al fijar la lista de Instrucciones, define también el código simbólico de cada una de las Instrucciones. En cuanto al operando, también se adopta para explicitarlo, una referencia simbólica en varias letras que el usuario puede elegir respetando convenciones dadas por el fabricante.

La codificación simbólica (programación en lenguaje Assembler) es muy cómoda para el programador, pero ésta no es "entendida" por la computadora, que solo comprende el lenguaje absoluto binario. Por ello, es necesario traducir el código simbólico a su equivalente hexadecimal o binario, esta tarea es realizada por el Programa ensamblador o armado. Con respecto a los operandos utilizados por las Instrucciones se pueden distinguir 3 tipos:

- a) operando implícito
- b) operando inmediato
- c) dirección efectiva

En el primer caso no hace falta ninguna indicación especial sobre el operando ya que éste está implícito en el código de operación. En el segundo, el operando forma parte de la Instrucción y en el tercero está en una dirección efectiva, que puede formarse en alguno de los Modos de Direccionamiento (directo, indexado, indirecto, etc).

## **INSTRUCCIONES DE TRANSFERENCIA DE INFORMACION**

Permiten las transferencias de datos entre elementos del sistema, por ejemplo entre un registro y otro, entre un registro y memoria y viceversa. Estas transferencias normalmente la copia del contenido de un elemento origen o fuente sobre un elemento denominado Destino, destruyendo el contenido anterior de este último, pero dejando inalterado el Registro fuente. Cuando el elemento origen es la memoria y el destino un Registro de CPU, la operación es cargar (load); en el caso Inverso se trata de almacenar (store).

Algunos procesadores poseen Instrucciones de intercambio de Información, en las cuales el contenido del Registro fuente va al Registro destino y viceversa sin perderse ninguno de los contenidos sino que se Intercambian.

## **INSTRUCCIONES ARITMETICO-LOGICAS**

Son aquellas que utilizan la Unidad Aritmético-lógica (ALU), en los procesadores de una dirección y trabajándose en general con dos operandos, uno de éstos es en la gran mayoría de los casos el acumulador. El otro podrá ser:

- a) Otro acumulador o Registro de uso general.
- b) Memoria de datos, debiéndose indicar el modo de direccionamiento en la Instrucción.

Una vez ejecutada la Instrucción, el Acumulador almacena el resultado.

Los procesadores de 8 bit tradicionales 8080, 6800, 6802, Z80 solo permiten operaciones aritméticas de adición o sustracción. Procesadores mas complejos y modernos como el 6809, 68000, 8086 y en general todos los de 16 bits permiten operaciones de multiplicación o multiplicación y división por hardware. Esto último le da al procesador una mayor potencia de cálculo.

Las operaciones antes mencionadas se realizan en el formato punto fijo. Todavía no se ha implementado dentro de un procesador (CPU) una unidad aritmética de punto flotante. Aunque existen coprocesadores que realizan únicamente operaciones aritméticas en punto flotante. Estos coprocesadores o unidades aritméticas realizan las operaciones elementales suma, resta, multiplicación, división, raíz, potenciación, funciones trigonométricas, logaritmo, etc; entre éstas podemos citar el procesador aritmético Am 9511 o el coprocesador de la familia del 8086 el 8087. También la familia del Microcomputador PDP-11 posee un procesador aritmético. Si nos remitimos a los procesadores de 8 bit, éstos representan los números con signo en el sistema de representación de complemento a 2. La unidad aritmética y lógica de un procesador realiza todas sus operaciones en binario, no obstante algunos MP, la gran mayoría, tienden a facilitar la programación de operaciones en decimal codificado en binario (BCD) con Instrucciones especiales como el ajuste decimal que lo poseen la mayoría de los procesadores, el SC/MP contempla dentro de su repertorio la Instrucción de suma en decimal, el procesador 6502 y el Z8000 tiene una bandera y de acuerdo al estado de ésta realiza las operaciones (+ o -) en binario o BCD.

En los microprocesadores de 8 bits, en general las operaciones suma o resta se realizan en 8 bits, cuyo bit mas significativo indica el signo (tener en cuenta el sistema de representación de complemento a 2). Si se desea expresar los números con una mayor cantidad de bits para mayor precisión, se encadenan por programa dos o mas bytes. Considerando como bit de signo el correspondiente a la palabra mas significativa, los demás bytes se consideran sin signo, pero el bit más significativo puede transmitir en la suma un acarreo o transporte (carry), el cual es necesario tomar en cuenta. Igual ocurre en la resta, donde se genera un "pido" (borrow). Para poder realizar operaciones en múltiple precisión el procesador deberá poseer Instrucciones especiales que permitan sumar o restar teniendo en cuenta el transporte generado en la operación anterior.

Cuando se suman dos números de igual signo o se restan dos de distinto signo, existe la posibilidad de desborde (overflow); el resultado excede la capacidad disponible en la palabra. Este caso debe ser indicado por el procesador, para que el programador tome alguna decisión al respecto, ya que el resultado obtenido es erróneo. Esta indicación también se debe realizar para el signo, transporte y resultado nulo. Estos indicadores se encuentran en un registro llamado Registro de Estado o Registro de Códigos de condición o registro de Indicadores ("Status Register", "Condition Code Register").

Este registro está formado por un cierto número de flip-flops independientes, colocándose en 1 cuando se ha producido la condición que cada uno debe indicar.

El número de Indicadores o banderas ("flags") varían en tipo y cantidad según el procesador, entre los más comunes podemos citar:

a- **Flag de cero**, se activa cuando el resultado de la última operación es cero o cuando en el acumulador u otro registro se ha transferido un dato nulo.

b- **Flag de signo**, se activa cuando el resultado de la última operación es negativo.

c- **Indicador de carry**, toma su valor activo cuando ocurre un transporte o pedido desde el bit más significativo.

d- **Flag de carry auxiliar**, se activa cuando se produce un acarreo desde el 4º al 5º bit.

e- **Flag de overflow**, se activa cuando se produce desborde.

f- **Flag de paridad del acumulador**, se activa si el número de unos del acumulador es par.

Los fabricantes indican en el set de instrucciones para cada operación que flags son afectados en el caso de operaciones lógicas, (and, or, or-exclusivo, inversión) estas operan bit a bit, por ejemplo, se considera la intersección del contenido del acumulador con el de un lugar de memoria, cada bit del primero será sometido al producto lógico con el bit correspondiente (de igual orden) del segundo.

Otras instrucciones de este grupo son las de prueba de datos. Estas implican una operación aritmética o lógica con la particularidad de que su resultado no es retenido ni se alteran los operandos; solo afecta a los Indicadores (flags).

Por ejemplo la Instrucción compare la cual realiza una resta sin obtener resultado, la Instrucción bit test la cual realiza una operación, la Instrucción test que le resta 00 a un dato, como puede verse, lo único que hace es actualizar las banderas.

## **INSTRUCCIONES DE MANEJO DE CONTENIDOS EN MEMORIA Y REGISTROS**

Estas operaciones afectan generalmente a un elemento de almacenamiento de CPU (registro, flip-flop de estado) o a una palabra de la memoria de datos y en la mayoría de los casos no producen transferencia de información de un lugar a otro; únicamente producen el cambio de un contenido binario.

Entre las instrucciones de este tipo encontramos las de poner en cero (Clear), incrementar (Increment), decrementar (Decrement); desplazar (Shift), el contenido a izquierda o a derecha, rotar (rotate) a izquierda o derecha, poner en uno un indicador (set condition flip-flop), poner en cero (reset, clear, condition, flip-flop), complementar a uno (complement), complementar a 2 (negate), y ajuste decimal (Decimal Adjust) la cual produce el ajuste del resultado de una operación cuando los operandos están en BCD.

Un caso particular de este grupo de instrucciones es la que posee el microprocesador SC/MP de incremento y carga del acumulador y decremento y carga la cual podría considerarse como una instrucción compuesta.

## **INSTRUCCIONES DE CONTROL DEL PROGRAMA**

Quizás la más importante función de un sistema de cómputo sea la de poder ser programado para tomar decisiones. Se podría afirmar que un programa útil debe tener decisiones.

Sin esas decisiones el programa consistiría en un cierto número de instrucciones que serían ejecutadas secuencialmente en el orden en que han sido almacenadas en memoria, la capacidad de decisión permite estructurar el programa en base a un conjunto de ramas cada una de las cuales se ejecutará si se cumple una cierta condición. Se parecerá entonces, a un árbol con ramificaciones y lazos, mas que a una sucesión lineal de operaciones, lo cual exige que exista una estructura de control en el software, además de la estructura de datos.

Es evidente que la estructura de control está íntimamente ligada a los contenidos del contador del programa (PC), pues este registro apunta siempre a la instrucción que debe ejecutarse. Si la secuencia de operaciones sigue el orden sucesivo normal, del PC se incrementará gradualmente; si por el contrario una circunstancia cualquiera, como una decisión impone la ruptura de esta secuencia para pasar a una ramificación, el PC deberá alterar su contenido indicando una dirección no consecutiva.

Las instrucciones de control del programa, también llamadas de ruptura de secuencia o transferencia de control, son las encargadas de formar el nuevo contenido del PC, estas instrucciones llevan asociado un operando de control según el modo de direccionamiento utilizado, que le permitirá obtener la dirección de la próxima instrucción a ejecutar.



Estas instrucciones se diferencian de todas las demás en que sus operandos están vinculados a la memoria del programa y no a la de datos y también porque en última instancia todas hacen la misma operación: carga del contador del programa, sin afectar datos ni transmitirlos.

El código de operación de las instrucciones de transferencia de control siempre implica la carga del PC, con un nuevo valor. Si ésta debe producirse sin restricciones (si o si), diremos que la transferencia es incondicional, ésta se usa por ejemplo cuando el programa debe salir de un lazo. Por otra parte, está la capacidad de decisión; la realización o no de la transferencia puede estar supeditada a alguna prueba ("test"). Diremos entonces que la transferencia es condicional. En este caso el mecanismo es siempre el siguiente: si la condición planteada se verifica, el PC es cargado con la nueva dirección saliendo de la secuencia normal; caso contrario, es incrementado como corresponde a dicha secuencia.

La forma más lógica de realizar las pruebas de cuyo resultado depende una transferencia condicional, es mediante los indicadores o banderas que ya se han mencionado.

Condiciones tales como resultado cero o negativo, transporte y desborde, son fáciles de aplicar en un programa, y son normalmente usadas en los microprocesadores.