MODOS DE DIRECCIONAMIENTO

A algunos estudiantes no les parece muy importante el tema de los modos de direccionamiento. Si el objetivo último y principal es procesar de alguna manera (sumar, restar, desplazar, etc) unos datos, se da por supuesto que es ahí (en dichas operaciones) donde está la única dificultad en las tareas del computador. Se supone que si debemos operar con unos operandos, es porque ya los tenemos. En nuestros cálculos humanos es así. Los datos los tenemos a la vista en papel, o en nuestra mente. Pero el computador no sabe por sí solo dónde están. El programador debe indicarle cómo llegar hasta ellos. El estudiante debe ser consciente de que la importancia de los modos de direccionamiento está basada en estos dos motivos:

- ✔ En el momento de redactar un programa en lenguaje ensamblador, debe conocer el conjunto de modos disponibles para usar los más adecuados en la optimización del programa.
- ✔ El motivo más radical es que el repertorio de modos de direccionamiento está estrechamente ligado a la arquitectura interna del microprocesador. La decisión de incluir o no un modo puede determinar la construcción o no de algunos componentes electrónicos (registros, contadores, operadores especiales, buses, etc) para acelerar la ejecución de dicho modo.

El texto base trata los modos de direccionamiento en dos capítulos. El capítulo 10 hace un estudio en general de los modos de direccionamiento habituales de los microprocesadores más conocidos, con una nomenclatura unificada. El capítulo 13 hace un estudio particularizado para el microprocesador M68000, con su nomenclatura propia. El estudiante se ve confundido por estos dos hechos:

- ✔ Nomenclaturas diferentes (en algunos casos contradictorias: *directo relativo a registro base* en el capítulo 10 e *indirecto a registro con desplazamiento* en el capítulo 13).
- ✔ Algunos modos generales no están presentes en el M68000; y lo que es más desconcertante: algunos modos del M68000 no habían sido contemplados en el estudio general.

Para evitar esta confusión podemos hacer un estudio comparado de los modos de direccionamiento. Primero recordemos los nombres:

EN GENERAL (tema 10)	**EN EL MICROPROCESADOR M68000 (tema 13)**
Inmediato	Inmediato
Directos:	
Absolutos:	
De registro	En registro de datos o de direcciones
De memoria	Absoluto (corto y largo)
De página base	–
Relativos:	
–	Indirecto a registro
Relativo a un registro base	Indirecto a registro con desplazamiento
Relativo al PC	Relativo al PC con desplazamiento
Relativo a pila	–
Relativos a un registro índice:	
Con preincremento	–
Con predecremento	Indirecto a registro con predecremento
Con posincremento	Indirecto a registro con posincremento
Con posdecremento	–
–	Indirecto a registro con índice y desplazamiento
–	Relativo al PC con índice y desplazamiento
Indirecto	–
–	Implícito

Tabla 1. Correspondencia entre los modos presentados en los temas 10 y 13:

DIRECCIONAMIENTO INMEDIATO

El dato está contenido en la instrucción. Pueden ser usadas una o varias palabras de memoria. En este último caso se precisan varios accesos a memoria.
Ventaja: Rapidez.
Inconveniente: El dato debe ser constante para todas las ejecuciones del mismo programa. Su tamaño debe caber en el campo del RI.
Sintaxis en el MC68000: El dato es precedido por el símbolo `#`.

Ejemplos:

```
move.w    #37,D1
addi      #1,D0
```

DIRECCIONAMIENTO DIRECTO ABSOLUTO

MEDIANTE REGISTRO DE DATOS O DE DIRECCIONES

La instrucción contiene el código de identificación de uno de los registros del procesador.
Ventaja: Rapidez.
Inconveniente: Si se utiliza este modo, hay que procurar hacerlo con los datos más usados, pues no suele haber suficientes registros para todos los datos del programa.
Sintaxis en el MC68000: En la instrucción se indica el nombre del registro.

EjemploS:

```
move.w      D0,D1
movea.w     $300,A1
```

ABSOLUTO

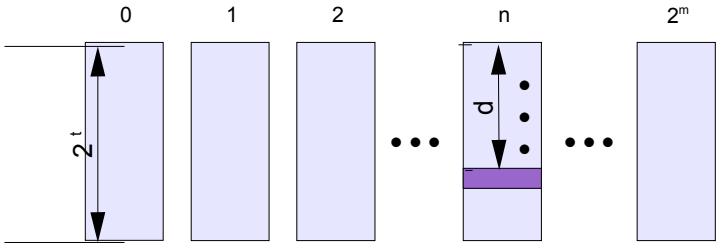
La instrucción contiene la dirección de memoria donde está almacenado el dato.
Ventaja: No se necesitan operaciones para calcular la dirección.
Inconveniente: Si el campo de la instrucción que contiene la dirección tiene una longitud de *n* bits, y la memoria tiene un tamaño superior a 2^n bytes, no es posible acceder a todas la posiciones de memoria con este modo.
Sintaxis en el MC68000: La dirección del dato se expresa de forma directa.

Ejemplo:

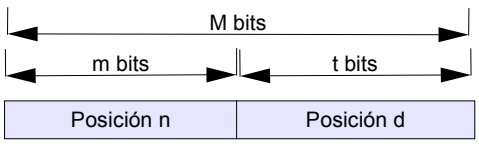
```
move.b    $300A,D2
```

DE PÁGINA BASE

Organización de la memoria:
d: Posición de la palabra dentro de la página
n: Número de página
2^t: Tamaño de la página
2^m: Número de página
2^M: Memoria total



Descomposición de las direcciones de memoria:



DIRECCIONAMIENTO DIRECTO RELATIVO

INDIRECTO A REGISTRO

La dirección donde está el dato está contenida en un registro de direcciones.
Utilidad: Punteros a datos simples (escalares).
Sintaxis en el MC68000: El nombre del registro de direcciones está entre paréntesis.
Ejemplo:

```
move.w   (A0),D0
```

INDIRECTO A REGISTRO CON DESPLAZAMIENTO

Utilidad: El procesamiento de formaciones unidimensionales y en sistemas operativos multiusuarios para trabajar con bloques de código o datos reubicables.
Sintaxis en el MC68000: Se indica el nombre del registro entre paréntesis precedido por una constante (el desplazamiento).
Ejemplo:

```
move.l   $20(A1),D1  ;  La operación efectuada es: ( (A1) + $20 )   D1
```

RELATIVO AL PC CON DESPLAZAMIENTO

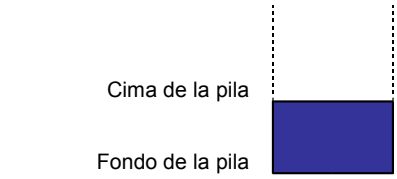
Utilidad: Es muy utilizado cuando el dato está cerca de la instrucción actual, por ejemplo en saltos, para obtener una nueva instrucción, que no está situada a continuación de la actual, pero sí está cerca.
Sintaxis en el MC68000: El PC puede referenciarse con las letras PC; o con el símbolo '*'.
Ejemplos:

```
add.w    *+$10,D1    ; 8 palabras ($10=16 bytes) más allá de la posición apuntada por el PC
move.b   5(PC),D0
```

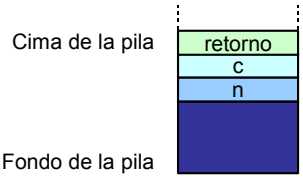
RELATIVO AL PUNTERO DE PILA

Es muy utilizado por los compiladores al traducir las llamadas a procedimientos en lenguaje de alto nivel a subrutinas en ensamblador.
Ejemplo:

```
PROCEDURE      Procedimiento( x1 : INTEGER; x2 : CHAR ) : REAL ; (* Declaración *)
.  .  .
Procedimiento(n, c) ;          (* Llamada *)
.  .  .
```



Estado de la pila fuera del procedimiento



Estado de la pila durante del procedimiento

RELATIVOS A UN REGISTRO CON ÍNDICE

En esencia son iguales que el direccionamiento indirecto a registro con desplazamiento. La dirección del dato en memoria se obtiene tras la suma del contenido de un registro más otra cantidad. La diferencia está en que el contenido del registro se actualiza con dicha suma; y es posible utilizar incrementos automáticos (sin necesidad de indicar explícitamente la suma) de ciertas cantidades enteras (positivas o negativas). Según sea el momento y signo de dichas cantidades, se clasifican en:

- Indirecto a registro con preincremento.
- Indirecto a registro con posdecremento.
- Indirecto a registro con predecremento.
  Sintaxis en el MC68000: Nombre del registro entre paréntesis precedido por el símbolo '-'.
- Indirecto a registro con posincremento.
  Sintaxis en el MC68000: Nombre del registro entre paréntesis precediendo al símbolo '+'.

Utilidades: Procesar arrays unidimensionales en un bucle. Ejemplo:

```
* Mueve 32 bytes desde la posición $2000 hasta la $3000, invirtiendo el orden de almacenamiento.

            org     $1000
            move.b  #32,D1          ; Inicializa el contador a 32
            movea.l #$2000,A1
            movea.l #$3001,A2       ; Fija las direcciones iniciales de transferencia.

InicioBucle move.b  (A1)+,-(A2)     ; Núcleo del programa
            subi.b  #1,D1           ; Decrementa el contador
            bne     InicioBucle     ; Continúa hasta que D1=0

            end
```

Estos dos últimos modos pueden combinarse para la síntesis por parte del usuario de pilas adicionales. El puntero de estas pilas puede ser cualquier registro de propósito general. Si la pila sintetizada crece hacia las direcciones bajas el predecremento es para la introducción de datos; y el posincremento para la extracción.

INDIRECTO A REGISTRO CON ÍNDICE Y DESPLAZAMIENTO

Sintaxis: Desplazamiento (Nombre del registro base, Nombre del registro índice)

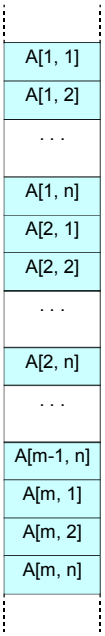
Ejemplo:

```
move.b   5(A0,D0),D1    ; La operación efectuada es: ( (A0) + (D0) + 5 )   D1
```

Utilidad: Procesamiento de formaciones bidimensionales:

Sea la matriz $A_{mn} = \begin{pmatrix} A_{[1,1]} & A_{[1,2]} & \cdots & A_{[1,n]} \\ A_{[2,1]} & A_{[2,2]} & \cdots & A_{[2,n]} \\ \cdots & \cdots & \cdots & \cdots \\ A_{[m,1]} & A_{[m,2]} & \cdots & A_{[m,n]} \end{pmatrix}$

Sus elementos están dispuestos en memoria de esta manera:



El algoritmo para recorrer los elementos de la matriz es:

```
            movea.l     #$1000,A0
            clr.l       D0                          ; Contador de filas, lo inicializa a 0
            .   .   .
InicioBucle < Procesar el elemento contenido en 0(A0,D0) >
            < Procesar el elemento contenido en 1(A0,D0) >
            .   .   .
            < Procesar el elemento contenido en (n-1)(A0,D0) >

            addi.l      #n,D0                       ; Avanza hasta el comienzo de la siguiente fila
            cmpi        #(n*m),D0
            bne         InicioBucle
```

RELATIVO AL CONTADOR DE PROGRAMA CON ÍNDICE Y DESPLAZAMIENTO

Sintaxis: Desplazamiento(PC, Nombre del registro índice)

Ejemplo:

```
move.b   5(PC,D0),D1    ; La operación efectuada es: ( (PC) + (D0) + 5 )   D1
```

DIRECCIONAMIENTO IMPLÍCITO

La ubicación del dato está determinada pora la propia instrucción.

Ejemplo:

```
rts
```

DIRECCIONAMIENTO INDIRECTO

La instrucción contiene la dirección donde está almacenada la dirección del dato. Características:

- No se necesitan operaciones aritméticas para calcular la dirección final.
- Son necesarios dos ciclos de lectura en memoria.
- Permite una gran capacidad de direccionamiento. Pueden ser utilizados todos los bits de la palabra leída.

Este modo es muy utilizado en :

- Sistemas operativos multiusuario.
- Bancos de datos. Se construye una tabla de punteros, en la que cada uno apunta al comienzo de una serie de registros.