

Practica 4.- “Tamaño de Instrucción”.

2.1.3 Instrucciones y modo de direccionamiento.

OBJETIVO DEL SUBTEMA. Modificar el programa que se diseñó en el punto 2.1.2. El programa actualizado además, deberá de obtener el tamaño de la instrucción en bytes.

Introducción:

MODOS DE DIRECCIONAMIENTO

Existen 15 modos de direccionamiento.

Todos los modos de direccionamiento, excepto el implícito, generan una dirección de 16 bits que se denomina dirección efectiva.

IMPLÍCITO

Todo lo que se necesita para ejecutar la instrucción se conoce con el código de la instrucción => no tiene operandos o los operandos son registros de la CPU.

INMEDIATO

El operando está contenido en los B que siguen a la instrucción. El número de B corresponde con el tamaño del registro afectado.

La dirección efectiva apunta a los B que siguen a la instrucción.

En el ensamblador se indica precediendo al dato por la almohadilla: #

EXTENDIDO

La dirección efectiva se indica con 2 B que siguen a la instrucción.

En ensamblador se indica poniendo dirección (sin almohadilla).

DIRECTO

El B menos significativo de la dirección se encuentra en el B que sigue a la instrucción. El B más significativo se supone es \$00. De esta manera sólo se puede acceder a las direcciones en el rango \$0000-\$00FF, el cual se denomina página cero.

De esta manera es posible el acceso a la página directa con 1 B menos que con el direccionamiento extendido => ocupa menos memoria y es más rápida (1 fetch menos).

En la página directa se suele colocar la memoria RAM o los registros de E/S.

En ensamblador, al igual que el anterior, se indica poniendo dirección (sin almohadilla). Al programar no tenemos que distinguir estos casos. El ensamblador elegirá el direccionamiento directo cuando sea posible.

RELATIVO

Se utiliza sólo en instrucciones de salto relativo. Se trata de un valor con signo que se suma al PC para dar la dirección de la siguiente instrucción. Son posibles saltos relativos cortos, con desplazamiento de 8 bits (rango -128 a +127) y saltos relativos largos, con desplazamiento de 16 bits (-32K a +32K).

INDEXADOS

Utilizan un registro de 16 bits de la CPU (X,Y, SP ó PC) al cual se le suma un desplazamiento, constante o variable, para dar la dirección efectiva. Todos añaden un Byte, denominado *postbyte*, en el que se indica el tipo y otras informaciones del direccionamiento.

INDEXADO CON DESPLAZAMIENTO CONSTANTE DE 5 BITS

El desplazamiento es constante con signo de 5 bits (rango -16 a +15) y se incluye en el *postbyte*.

INDEXADO CON DESPLAZAMIENTO CONSTANTE DE 9 BITS

El desplazamiento es constante con signo de 9 bits (1 bit en el postbyte y el resto en un byte de extensión), dando un rango de -256 a +255.

INDEXADO CON DESPLAZAMIENTO CONSTANTE DE 16 BITS

El desplazamiento es constante con signo de 16 bits (en 2 bytes de extensión), dando un rango de ± 32 Kb.

En todos estos tres casos, en ensamblador se indica el valor del desplazamiento, una coma y el registro que se utiliza como base:3,X -120,Y -500,SP 15,PC Cuando el desplazamiento es 0 basta con indicar el registro base, es decir, X equivale a 0,X

INDEXADO CON DESPLAZAMIENTO EN ACUMULADOR

La dirección efectiva es la suma de un registro base de 16 bits (X, Y, SP ó PC) y una acumulador (A, B ó D). En ensamblador se pone registro de desplazamiento seguido de registro base: B,X A,PC D,Y

INDEXADO CON PRE/POST INCREMENTO/DECREMENTO

El registro base (en este caso sólo X, Y ó SP) se incrementa/decrementa antes/después de ser utilizado como dirección efectiva del operando. Por lo tanto el registro base queda modificado tras la utilización de este direccionamiento. El valor del incremento/decremento puede estar comprendido entre 1 y 8.

Ejemplo de las 4 posibilidades son:

1,+SP

el SP se incrementa en 1 y su nuevo valor se utiliza como dirección efectiva.

3,-X

el registro X de decrementa en 3 y su nuevo valor se utiliza como dirección efectiva.

 $7, Y_+$

el valor en registro Y se utiliza como dirección efectiva y posteriormente a dicho registro se le suma 7.

2,X-

el valor en registro X se utiliza como dirección efectiva y posteriormente a dicho registro se le resta 2.

INDEXADO INDIRECTO CON DESPLAZAMIENTO CONSTANTE DE 16 BITS

Ahora la suma del registro base y el desplazamiento constante (en 2 bytes de extensión) es utilizado para traer de memoria un dato de 16 bits que es utilizado como dirección efectiva del operando.

En ensamblador se denota utilizando corchetes: $[-4, X]$

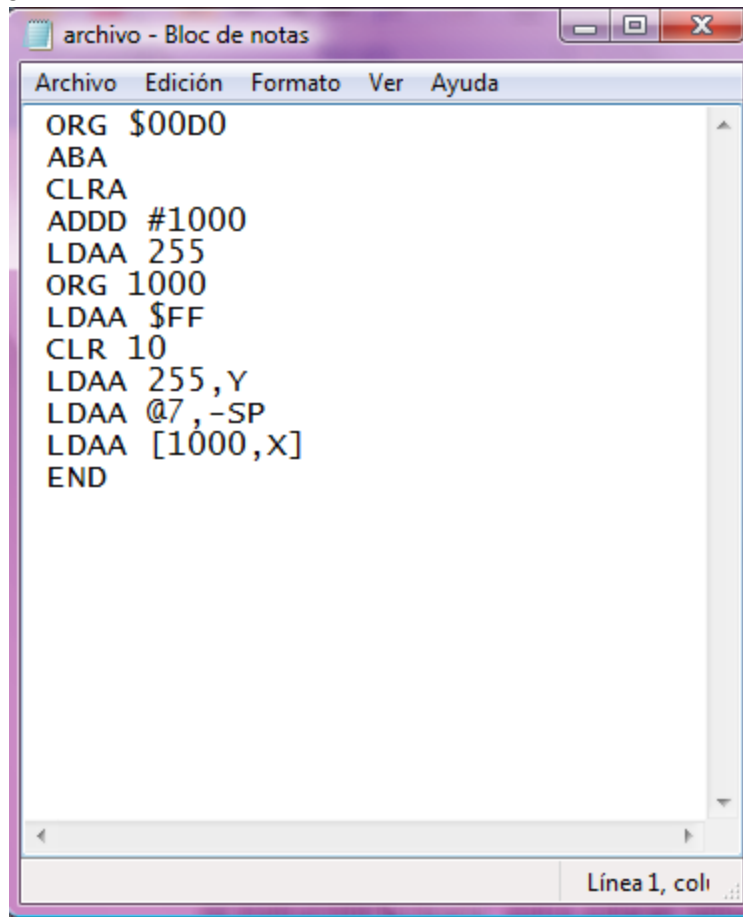
INDEXADO INDIRECTO CON DESPLAZAMIENTO EN ACUMULADOR D

Ídem que el anterior, pero como desplazamiento se utiliza el contenido del acumulador D:
[D,PC]



DESARROLLO:

En un archivo con extensión asm, se introducen las líneas de código del programa en lenguaje ensamblador.



```
archivo - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
ORG $00D0
ABA
CLRA
ADDD #1000
LDAA 255
ORG 1000
LDAA $FF
CLR 10
LDAA 255,Y
LDAA @7,-SP
LDAA [1000,X]
END
Línea 1, col 1
```

Se va analizando línea por línea de el código con lenguaje Ensamblador para que posteriormente el programa separe estas instrucciones y las clasifique de acuerdo al su campo correspondiente ya sea una

Etiqueta

Instrucción

Operando

Comentarios

Tomando en cuenta los operandos y la instrucción, que son consultadas en el tabop, para que pertenezcan realmente al lenguaje ensamblador

En el caso de que si hubiese más de un operando, deberá especificarse cuantos tiene y numerarlos desde 0 hasta “n”, siendo “n” el número de operandos en la línea de instrucción

tecleada, estos estarán separados por comas.

Mostrar cuales fueron los comentarios no es necesario, es opcional si se muestran o no; más no deben de aparecer en el campo de operando cuando el programa muestre la separación de la instrucción.

El primer campo será necesariamente para la etiqueta, seguida de la instrucción, posteriormente los operandos y los comentarios aparecerán al final.

Cada campo estará separado aleatoriamente por cuantos espacios y/o tabulaciones indique el profesor. En caso de omitir algún campo en la línea de instrucción, al momento de que el programa muestre la separación, en el campo omitió deberá aparecer NULL o en blanco.

El programa deberá indicar la cantidad de bytes que requerirá la instrucción si el formato del operando fue el correcto. Si el operando que tenía la instrucción no se apega al formato deberá indicarse un error de modo de direccionamiento no valido.

Nota: en esta práctica solo son obligatorias las instrucciones que soportan un solo modo de direccionamiento.