

## Practica 7.- “Direccionamientos Relativos”.

### 2.2.2.- Modos de direccionamiento indizados y relativos.

*OBJETIVO DEL SUBTEMA. Obtener el código máquina de direccionamientos indizados y los relativos.*

#### *DIRECCIONAMIENTO RELATIVO*

En este modo el contenido del contador de programa se suma a la parte de dirección de la instrucción para obtener la dirección efectiva. La parte de dirección de la instrucción por lo general es un número con signo (en representación de complemento a 2 el cual puede ser positivo o negativo. Cuando se suma este número al contenido del contador de programa, el resultado produce una dirección efectiva cuya posición en la memoria es relativa a la dirección de la siguiente instrucción.

Este modo de direccionamiento es el empleado en las instrucciones de bifurcación (Branch):

- Si la condición de salto es cierta, el dato de 8 bits con signo suministrado con el COP de la instrucción se suma al contador de programa (PC) para obtener la dirección de la próxima instrucción a ejecutar.
- Si la condición es falsa se ejecuta la instrucción siguiente a la de salto condicional.

*“La dirección efectiva se obtiene de la suma con signo del contenido del contador de programa más el operando del código máquina.”*

En la práctica 7 genera los mismos archivos que la practica 6 generando el código máquina calculado, pero además agrega el direccionamiento relativo. Este modo de direccionamiento es utilizado exclusivamente por las instrucciones de salto condicional (branch). La dirección de memoria a la que se transfiere el control del programa se obtiene sumando el valor del contador de programa al segundo byte de la instrucción, denominado "offset". Este offset es un número en complemento a-2, con lo que se pueden efectuar saltos de hasta 127 posiciones hacia adelante o 128 hacia atrás. El primer paso reserva las localidades de memoria (contloc) y TABSIM (referencias).

En el direccionamiento relativo a la dirección proporcionada por el registro de instrucciones se le añade una dirección de referencia, normalmente esta dirección de referencia es el contador de programa PC.

La formula es  $EA-PC=rr$

EA es el operando de la instrucción.

PC es la dirección de memoria que le corresponderá al código máquina de la siguiente línea de instrucción.

Ejemplo : Programa

```
ORG $4000
BNE $4000
BNE $4000
END
```

4000 26 rr BNE \$4000	4000 - 4002=FE	4000 26FE BNE \$4000
4002 26 rr BNE \$4000	4000 - 4004=FC	4002 26FC BNE \$4000
		4004 END

En el desarrollo de la práctica utilizo la practica anterior además de los archivos generados de la practica posterior en esta se calculan las instrucciones de direccionamiento relativo de 8 a 16 bits se utilizara una resta que muestre el resultado de rr.

```

C:\Documents and Settings\Usuario\Escritorio\practicaprogramacion\practicaprogramacion\practica_7\prac7.exe
Direcccionamiento Relativo
Taller de Programacion de Sistemas
Ceballos Cruz Adriana

ARCHIVO EN USO
ETIQUETA:      NULL
INSTRUCCION:   ORG
OPERANDO 0 :   $4000

LINEA DE CODIGO COMPLETA:      ORG      $4000
INSTRUCCION VALIDA

ETIQUETA:      INICIO
INSTRUCCION:   ldaa
OPERANDO 0 :   #$30

LINEA DE CODIGO COMPLETA:      INICIO      ldaa      #$30

```

```

;A<=30H
IMM 86ii 2

ETIQUETA:      dato1
INSTRUCCION:   adda
OPERANDO 0 :   50

LINEA DE CODIGO COMPLETA:      dato1      adda      50
DIR opr8a 9Bdd 2

ETIQUETA DUPLICA

ETIQUETA:      NULL
INSTRUCCION:   bra
OPERANDO 0 :   $4000

LINEA DE CODIGO COMPLETA:      bra      $4000
REL rel18 20rr 2

```

```

ETIQUETA:      NULL
INSTRUCCION:   lbra
OPERANDO 0 :   $7000

LINEA DE CODIGO COMPLETA:      lbra      $7000
REL rel16 1820qqrr 4

ETIQUETA:      NULL
INSTRUCCION:   lbne
OPERANDO 0 :   INICIO

LINEA DE CODIGO COMPLETA:      lbne      INICIO
REL rel16 1826qqrr 4

```

```

ETIQUETA:      NULL
INSTRUCCION:   rol
OPERANDO 0 :   10

LINEA DE CODIGO COMPLETA:      rol      10
EXT   opr16a 75hh11 3

ETIQUETA:      NULL
INSTRUCCION:   END
OPERANDO 0 :

LINEA DE CODIGO COMPLETA:      END
NO EXISTE INSTRUCCION

ETIQUETA:      NULL
INSTRUCCION:   END
OPERANDO 0 :

LINEA DE CODIGO COMPLETA:      END
NO EXISTE INSTRUCCION

```

En el archivo .lst se muestra los relativos ya calculados

```

0000                                ORG      $4000
4000 86 3D      INICIO      ldaa      #$30      ;A<=30H
4002 9B 32      dato1      adda      50
4004 20 FA                                bra      $4000
4006 1820 2FF6                                lbra      $7000
400A 1826 FFF2                                lbne      INICIO
400E 75 000A                                rol      10
4011                                END

```

En la línea de la dirección de memoria 4004 la instrucción es:

**bra        \$4000**

y según el manual esta instrucción es:

**BRA   rel8   20 rr**

rel8 es cualquier valor entre 0 y 65535

en la instrucción el operando es \$4000, el símbolo “\$” indica que se trata de un numero hexadecimal, entonces 4000H en decimal sería el número 16384 que está dentro del rango, por lo tanto el operando “\$4000” es el operando rel8, y

**rr = rel8 – CONLOC (siguiente)**

si el código maquina es 20 rr entonces este requiere 2 bytes y

**CONLOC (siguiente) = 4004H + 2H = 4006H**  
**rel8 = 4000H**

$$rr = 4000H - 4004H = FFFAH$$

Que es el complemento a 2 del 4H en 16 bits. Lo cual representa el número decimal -4. Queda dentro del rango de rr que debe estar entre -128 y 127, en ese caso rr serán los dos últimos dígitos hexadecimales ( FA). Si el valor se saliera de rango se marcaría como error.

Para calcular qq rr se consideran los 4 dígitos del resultado de la resta.