Unidad

4

###### Nota: Los ejercicios con la siguiente leyenda se presentan resueltos:✍.

El Microprocesador

* Objetivo general del trabajo práctico

El objetivo de este Trabajo práctico es proporcionar al alumno una experiencia en el diseño de una programación en bajo nivel utilizando la familia de microprocesadores de INTEL.

El trabajo práctico está orientado a los alumnos de primer año de la carrera de ingeniería en sistemas de información y específicamente para la asignatura de Arquitectura de las Computadoras de manera que los contenidos vertidos aquí son los básicos para el aprendizaje sobre manejo de modos de direccionamiento, datos, operandos, etc. de microprocesadores y de la herramienta informática debug.

* Objetivos específicos
* Comprender el concepto de microprocesador y su entorno.
* Comprender a través de la experiencia de la programación cómo funciona el hardware de un sistema basado en microprocesador.
* Comprender cómo se integran el hardware y el software en los dispositivos digitales como memorias e interfaces de I/O en los sistemas basado en microprocesadores.
* Comprender como trabaja un microprocesador en relación al sistema de memoria de una computadora.
* Obtener experiencia práctica en el diseño aplicado de software de bajo nivel.
* Entender las herramientas y técnicas utilizadas por los ingenieros en Sistemas de Información para diseñar, ejecutar y depurar los sistemas basados en microprocesadores.
* Para tener en cuenta
* Los problemas propuestos en el presente TP pueden ser resueltos en forma individual o grupal.
* De acuerdo a la complejidad creciente que esta asignatura presenta, desde la unidad número uno y hasta esta, el planteo o formulación de los problemas debe ser realizado basándose en lo desarrollado en las clases teóricas referidas a esta unidad y a todas las unidades anteriores ya desarrolladas así como también de aquellas asignaturas que son correlativas de esta.
* Es fundamental para un correcto aprendizaje que el alumno controle sus soluciones con la herramienta DEBUG. En el Anexo II de esta guía se puede recurrir a un resumen y ejemplificación de cada uno de sus comandos.
* Se recomienda disponer y realizar una lectura general del conjunto de instrucciones de la familia Intel.
* Informe Final del Trabajo Práctico

Los alumnos deberán presentar por escrito el desarrollo de cada uno de los problemas con las salidas respectivas de DEBUG.

* Conceptos necesarios de entrada
* Sistemas de Numeración: particularmente, binario, octal y hexadecimal.
* Algebra de Boole.
* Aritmética Digital.
* Circuitos lógicos Secuenciales.
* Memoria del sistema (estáticas, dinámicas, etc.).
* Memorias RAM y ROM. Memorias Caché.
* Memorias auxiliares: discos duro (HD), cintas, discos flexibles.
* Dispositivo de entrada y salida. Generalidades.
* Notas acerca de la nomenclatura

Nota 1: En la nomenclatura a lo largo del trabajo práctico se encuentra con:

**179A:0107 0000 ADD [BX+SI],AL**

**:**

**:**

**179A:011D 00891700 ADD [BX+DI+0017],CL**

Significa que se han omitido el resto de líneas de ejecución con el objeto de simplificar el trabajo.

Nota 2: **179A:0080**

Los primeros cuatro dígitos hexadecimales corresponden a DS (segmento de dato) en caso que se esté visualizando un volcado de memoria (comando D), o bien CS (segmento de código) si se trata de línea de ejecución. Los siguientes cuatro dígitos corresponden al desplazamiento dentro del segmento de referencia.

Ejercicios del Microprocesador

✍ Ejercicio 1: **Modo de direccionamiento inmediato y por registro.**

a) Escribir una secuencia de instrucciones que permita cargar (mover) la parte baja del registro AX (AL) con el valor F0H, la parte baja del registro BX (BL) con 3FH, sumar ambos registros y dejar el resultado en el registro AX.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

✍ Ejercicio 2: **Modo de direccionamiento extendido y por registro.**

a) Cargar el acumulador (AX) con el contenido de la posición 0030H, el reg. (BX) con el contenido de 0032H, sumar ambos registros y dejar el resultado en el acumulador AX. Las posiciones de memoria hacen referencia al bloque de memoria dentro del Segmento de Datos DS. (opcional dejar el resultado en la posición de memoria 0034H)

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

✍ Ejercicio 3: **Modo de direccionamiento indexado y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan limpiar (poner a cero) el contenido de diez posiciones de memorias consecutivas comenzando desde la posición 0040H

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

✍ Ejercicio 4: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan sumar un bloque de memoria que comienza en 0030H hasta 0040H depositar el resultado de 0040H. Ver datos de la tabla N°1

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

|  |  |
| --- | --- |
| **Posición de memoria** | **Contenido** |
| 0041 | 00 |
| 0040 | 78 |
| 003F | 0F |
| 003E | 0E |
| 003D | 0D |
| 003C | 0C |
| 003B | 0B |
| 003A | 0A |
| 0039 | 09 |
| 0038 | 08 |
| 0037 | 07 |
| 0036 | 06 |
| 0035 | 05 |
| 0034 | 04 |
| 0033 | 03 |
| 0032 | 02 |
| 0031 | 01 |
| 0030 | 00 |

✍ Ejercicio 5: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan sumar uno a uno los bytes de dos bloques de memoria que van desde la posición de memoria 0000H hasta la 0007H y desde la posición de memoria 0008H hasta la 000FH depositando las sumas respectivas en el bloque que va desde la posición 0010H hasta la 0017H. Almacenar el resultado de 0010H

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

|  |  |
| --- | --- |
| **Posición de memoria** | **Contenido** |
| 0017 | 01 |
| 0016 | 78 |
| 0015 | AB |
| 0014 | FE |
| 0013 | A2 |
| 0012 | 4E |
| 0011 | 10 |
| 0010 | EA  +  **=** |
| 000F | 03 |
| 000E | 8A |
| 000D | 11 |
| 000C | FE |
| 000B | 03 |
| 000A | 4F |
| 0009 | F0 |
| 0008 | 1D |
| 0007 | FE |
| 0006 | EE |
| 0005 | 9A |
| 0004 | 00 |
| 0003 | 9F |
| 0002 | FF |
| 0001 | 20 |
| 0000 | CD |

✍ Ejercicio 6: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan mover un bloque de memoria de 80H bytes ubicados desde la posición 0000H a la 007FH a las posiciones 0200H a 027FH.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

✍ Ejercicio 7: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan cargar con una tabla descendente comenzando con el valor FFH y dese la posición de memoria 0000H en adelante

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

✍ Ejercicio 8: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan cargar con una tabla descendente comenzando con el valor FFH y desde la posición de memoria 0000H en adelante.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 9:

a) Escribir un programa que intercambie los contenidos de las posiciones de memoria desde la A000H hasta la A0FFH. El contenido de la primera dirección deberá colocarse en la última y el de la última en la primera.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 10:

a) Dada una tabla que comienza en la posición de memoria B000H y cuya longitud es de 50H byte, determinar cuántos bytes son mayores a F0H, cuántos menores y cuantos iguales. Depositar el resultado en las posiciones C000H, C001H y C002H respectivamente.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 11:

a) Dada una tabla que comienza en la posición de memoria B0A0H y cuya longitud está indicada en la posición de memoria C001H, determinar cuántos byte son mayores a FBH, cuántos menores y cuantos iguales. Depositar el resultado en las posiciones D000H, D001H y D002H respectivamente.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 12:

a) Dada una tabla que comienza en la posición de memoria B0A0H y cuya longitud está indicada en la posición de memoria C001H, determinar cuántos byte son mayores al contenido especificado por la posición A000H, cuántos menores y cuantos iguales. Depositar el resultado en las posiciones D000H, D001H y D002H respectivamente.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 13:

a) Dada una tabla que comienza en la posición de memoria C000H y cuya longitud es de 100H bytes ordenarla de mayor a menor con el método de la burbuja.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Ejercicio 14:

a) Dada una tabla que comienza en la posición de memoria A0A0H y cuya longitud es de 10H byte borrar todas las posiciones de memoria si en la posición de memoria C200H existe el dato FFH. Caso contrario dejar como está.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

**Ejercicio 15**

a) Dadas dos tablas una comienza en la dirección de memoria A000H, la otra en la dirección memoria B000H y cuyas longitudes son de FFH bytes, indicar si ambas tablas contienen la misma información colocando un FFH en la posición C000H, caso contrario colocar un 00H

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

**Ejercicio 16**

**a**) Escribir un programa para cargar las direcciones de memorias desde la A009H hasta la AFFFH inclusive desde una tabla que se encuentra en la posición D000H y tiene una longitud de 50H byte. La carga deberá hacerse en forma ascendente.

**b)** Escriba el mismo programa utilizando la herramienta DEBUG.

**c)** Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

**Ejercicio 17**

a) Ídem problema 16 pero en forma descendente.

**Ejercicio 18**

a) Escribir un programa que realice la multiplicación de dos números binarios de 1 byte, sin signo, dando un resultado de dieciséis bits.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

**Ejercicio 19**

a) Suponga que dispone de un display de siete segmentos y una señal BCD de cuatro líneas. Escribir un programa para convertir la señal BCD al código de siete segmentos.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado

* SOLUCIONES

✍ Ejercicio 1: **Modo de direccionamiento inmediato y por registro.**

**a)** Escribir una secuencia de instrucciones que permita cargar (mover) la parte baja del registro AX (AL) con el valor F0H, la parte baja del registro BX (BL) con 3FH, sumar ambos registros y dejar el resultado en el registro AX.

**b)** Escriba el mismo programa utilizando la herramienta DEBUG.

**c)** Utilice los comandos R,A,U,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico (el código en negrita es el cargado):

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A **IP=0100** NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

MD Inmediato

-**A**

**179A:0100 MOV AL,F0**

MD Inmediato

**179A:0102 MOV BL,3F**

**179A:0104 ADD AX,BX**

**179A:0106 INT 3**

MD por Registro

179A:0107

-**U**

**179A:0100 B0F0 MOV AL,F0**

**179A:0102 B33F MOV BL,3F**

**179A:0104 01D8 ADD AX,BX**

**179A:0106 CC INT 3**

**179A:0107 0000 ADD [BX+SI],AL**

**:**

**:**

**179A:011D 00891700 ADD [BX+DI+0017],CL**

Se ejecuta paso a paso (T) para ver la evolución de los registros:

-**R**

**AX=0000** **BX=0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A **IP=0100** NV UP EI PL NZ NA PO NC

**179A:0100 B0F0 MOV AL,F0**

-**T**

**AX=00F0 BX=0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A **IP=0102** NV UP EI PL NZ NA PO NC

**179A:0102 B33F MOV BL,3F**

-**T**

**AX=00F0 BX=003F** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A **IP=0104** NV UP EI PL NZ NA PO NC

**179A:0104 01D8 ADD AX,BX**

-**T**

**AX=012F BX=003F** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A **IP=0106** NV UP EI PL NZ NA PO NC

**179A:0106 CC INT 3**

Solamente se carga la parte baja de los registros pares AX y BX, se suma ambos para evitar truncamiento de la parte alta, vemos qué sucede si sumamos solo la parte baja de los registros.

-**A104**

MD por Registro, solo se suman la parte baja

**179A:0104 ADD AL,BL**

179A:0106

-**U100**

**179A:0100 B0F0 MOV AL,F0**

**179A:0102 B33F MOV BL,3F**

**179A:0104 00D8 ADD AL,BL**

**179A:0106 CC INT 3**

**:**

**:**

**179A:011B 0034 ADD [SI],DH**

**179A:011D 00891700 ADD [BX+DI+0017],CL**

-**RAX**

Se pone a cero los registros pares

AX 012F

:**0**

-**RBX**

BX 003F

:**0**

-R

**AX=0000 BX=0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A **IP=0106** NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

-**G=100**

**AX=002F BX=003F** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO **CY**

**179A:0106 CC INT 3**

CY=1 (acarreo) auxiliar

✍ Ejercicio 2: **Modo de direccionamiento extendido y por registro.**

a) Cargar el acumulador (AX) con el contenido de la posición 0030H, el reg. (BX) con el contenido de 0032H, sumar ambos registros y dejar el resultado en el acumulador AX. Las posiciones de memoria hacen referencia al bloque de memoria dentro del Segmento de Datos DS. (opcional dejar el resultado en la posición de memoria 0034H)

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico (el código en negrita es el cargado):

-**R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0100** NV UP EI PL NZ NA PO NC

**179A:0100 0000 ADD [BX+SI],AL DS:0000=CD**

-**A**

**179A:0100 MOV AX,[30]**

**179A:0103 MOV BX,[32]**

**179A:0107 ADD AX,BX**

**179A:0109 INT 3**

179A:010A

-**U100**

**179A:0100 A13000 MOV AX,[0030]**

**179A:0103 8B1E3200 MOV BX,[0032]**

**179A:0107 01D8 ADD AX,BX**

**179A:0109 CC INT 3**

**179A:010A 0000 ADD [BX+SI],AL**

**:**

**:**

**179A:011E 8917 MOV [BX],DX**

Carga de las posiciones de memoria con los datos a ingresar en los registros: (AX=0E3F BX=0912)

|  |  |
| --- | --- |
| Posición de memoria | **Contenido** |
| 0033 | **09** |
| 0032 | **12** |
| 0031 | **0E** |
| 0030 | **3F** |

-**E30**

179A:**0030** BE. **3F**  16. **0E** 14. **12** 00. **09**

-D30

179A:0030 **3F 0E 12** **09** 18 00 9A 17-FF FF FF FF 00 00 00 00 ?...............

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

:

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

Se ejecuta paso a paso (T) para ver la evolución de los registros:

**-R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A IP=**0100** NV UP EI PL NZ NA PO NC

**179A:0100 A13000 MOV AX,[0030] DS:0030=0E3F**

-**T**

AX=**0E3F** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0103** NV UP EI PL NZ NA PO NC

**179A:0103 8B1E3200 MOV BX,[0032] DS:0032=0912**

-**T**

AX=**0E3F** BX=**0912** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0107** NV UP EI PL NZ NA PO NC

**179A:0107 01D8 ADD AX,BX**

-**T**

AX=**1751** BX=**0912** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0109 NV UP EI PL NZ AC PO NC

**179A:0109 CC INT 3**

Se dejará el resultado de la suma de las dos posiciones de memoria dentro de la posición 0034H y 0035H:

-**RAX**

Se pone a cero los registros pares

AX 1751

:**0**

-**RBX**

BX 0912

:**0**

-**RIP**

IP 0109

:**100**

-**R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0100**  NV UP EI PL NZ AC PO NC

**179A:0100 A13000 MOV AX,[0030] DS:0030=0E3F**

-**A109**

Se transfiere el resultado a la posición de memoria 0034 y consecutiva

**179A:0109 MOV [34],AX**

**179A:010C INT 3**

179A:010D

-**U100**

**179A:0100 A13000 MOV AX,[0030]**

**179A:0103 8B1E3200 MOV BX,[0032]**

**179A:0107 01D8 ADD AX,BX**

**179A:0109 A33400 MOV [0034],AX**

**179A:010C CC INT 3**

179A:010D 0000 ADD [BX+SI],AL

:

:

179A:011D 00891700 ADD [BX+DI+0017],CL

-**D30**

**179A:0030 3F 0E 12 09** **18 00** 9A 17-FF FF FF FF 00 00 00 00 ?...............

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

:

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

-**R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0100** NV UP EI PL NZ AC PO NC

**179A:0100 A13000 MOV AX,[0030] DS:0030=0E3F**

**-G=100**

AX=**1751** BX=**0912** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010C NV UP EI PL NZ AC PO NC

**179A:010C CC INT 3**

-**D30**

**179A:0030 3F 0E 12 09 51 17** 9A 17-FF FF FF FF 00 00 00 00 ?...Q...........

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

:

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

✍ Ejercicio 3: **Modo de direccionamiento indexado y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan limpiar (poner a cero) el contenido de diez posiciones de memorias consecutivas comenzando desde la posición 0040H

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico

**-R**

AX=**0000** BX=**0000** CX=**0000** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0100** NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

**-A**

179A:0100 **XOR AX,AX**

179A:0102 **XOR CX,CX**

179A:0104 **MOV BX,40**

179A:0107 **MOV [BX],AL**

179A:0109 **INC CX**

179A:010A **INC BX**

179A:010B **CMP CX,0A**

179A:010E **JNE 107**

179A:0110 **INT 3**

179A:0111

**-U100**

**179A:0100 31C0 XOR AX,AX**

**179A:0102 31C9 XOR CX,CX**

**179A:0104 BB4000 MOV BX,0040**

**179A:0107 8807 MOV [BX],AL**

**179A:0109 41 INC CX**

**179A:010A 43 INC BX**

**179A:010B 83F90A CMP CX,+0A**

**179A:010E 75F7 JNZ 0107**

**179A:0110 CC INT 3**

179A:0110 0000 ADD [BX+SI],AL

:

:

179A:011E 8917 MOV [BX],DX

Se examina el contenido de las posiciones 0040 a 004A y se reemplaza cada una por FFH

**-D40**

179A:0040 **05 00 00 00 00 00 00 00-00 00 00** 00 00 00 00 00 ................

:

:

179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

**-E40**

**179A:0040 05. FF 00.FF 00.FF 00.FF 00.FF 00.FF 00.FF 00.FF**

**179A:0048 00.FF 00.FF 00.FF 00.FF 00.FF 00.FF 00.FF 00.FF**

**-D40**

**179A:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF ................**

179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........

:

:

179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

**-R**

AX=**0000** BX=**0000** CX=**0000** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A**  ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL ZR NA PE NC

**179A:0100 31C0 XOR AX,AX**

**-G=100**

AX=**0000** BX=**004A** CX=**000A** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0110** NV UP EI PL ZR NA PE NC

**179A:0110 CC INT 3**

**-D40**

**179A:0040 00 00 00 00 00 00 00 00-00 00** FF FF FF FF FF FF ................

179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........

:

:

179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

✍ Ejercicio 4: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan sumar un bloque de memoria que comienza en 0030H hasta 0040H depositar el resultado de 0040H. Ver datos de la tabla N°1

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico:

**-R**

AX=**0000**  BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

**-A100**

179A:0100 **MOV BX,30**

179A:0103 **XOR AX,AX**

179A:0105 **ADD AL,[BX]**

179A:0107 **INC BX**

179A:0108 **CMP BX,41**

179A:010B **JNE 105**

179A:010D **DEC BX**

179A:010E **MOV [BX],AX**

179A:0110 **INT 3**

179A:0111

**-U100**

**179A:0100 BB3000 MOV BX,0030**

**179A:0103 31C0 XOR AX,AX**

**179A:0105 0207 ADD AL,[BX]**

**179A:0107 43 INC BX**

**179A:0108 83FB41 CMP BX,+41**

**179A:010B 75F8 JNZ 0105**

**179A:010D 4B DEC BX**

**179A:010E 8907 MOV [BX],AX**

**179A:0110 CC INT 3**

:

:

179A:011E 8917 MOV [BX],DX

**-D30**

**179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00** ................

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

:

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

**-E30**

179A:0030 BE.**00** 16.**01** 14.**02** 00.**03** 18.**04** 00.**05** 9A.**06** 17.**07**

179A:0038 FF.**08** FF.**09** FF.**0A** FF.**0B** 00.**0C** 00.**0D** 00.**0E** 00.**0F**

**-D30**

**179A:0030 00 01 02 03 04 05 06 07-08 09 0A 0B 0C 0D 0E 0F**  ................

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

:

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

**-R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL ZR NA PE NC

**179A:0100 BB3000 MOV BX,0030**

**-G=100**

AX=**0078** BX=**0040**  CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=**0110** NV UP EI PL NZ NA PO NC

**179A:0110 CC INT 3**

**-D30**

**179A:0030 00 01 02 03 04 05 06 07-08 09 0A 0B 0C 0D 0E 0F** ................

**179A:0040 78 00** 00 00 00 00 00 00-00 00 00 00 00 00 00 00 x...............

:

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

✍ Ejercicio 5: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan sumar uno a uno los bytes de dos bloques de memoria que van desde la posición de memoria 0000H hasta la 0007H y desde la posición de memoria 0008H hasta la 000FH depositando las sumas respectivas en el bloque que va desde la posición 0010H hasta la 0017H. depositar el resultado de 0010H

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico

**-A100**

179A:0100 **MOV BX,0**

179A:0103 **XOR AX,AX**

179A:0105 **MOV AL,[BX]**

179A:0107 **ADD AL,[BX+08]**

179A:010A **MOV [BX+10],AL**

179A:010D **INC BX**

179A:010E **CMP BX,08**

179A:0111 **JNE 105**

179A:0113 **INT 3**

179A:0114

**-U100**

**179A:0100 BB0000 MOV BX,0000**

**179A:0103 31C0 XOR AX,AX**

**179A:0105 8A07 MOV AL,[BX]**

**179A:0107 024708 ADD AL,[BX+08]**

**179A:010A 884710 MOV [BX+10],AL**

**179A:010D 43 INC BX**

**179A:010E 83FB08 CMP BX,+08**

**179A:0111 75F2 JNZ 0105**

**179A:0113 CC INT 3**

:

:

179A:011E 8917 MOV [BX],DX

**-R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL NZ NA PO NC

**179A:0100 BB0000 MOV BX,0000**

**-D0**

**179A:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03** . ........O.....

**179A:0010 FE 11 17 03 FE 11 28 07**-01 01 01 00 02 FF FF FF ......(.........

:

:

179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........

**-G**

AX=**0001** BX**=0008** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0113** NV UP EI PL ZR NA PE NC

**179A:0113 CC INT 3**

**-D0**

**179A:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03** . ........O.....

**179A:0010 EA 10 4E A2 FE AB 78 01**-01 01 01 00 02 FF FF FF ..N...x.........

:

:

179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........

✍ Ejercicio 6: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan mover un bloque de memoria de 80H bytes ubicados desde la posición 0000H a la 007FH a las posiciones 0200H a 027FH.

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico

**-A100**

179A:0100 **MOV BX,0**

179A:0103 **MOV AX,[BX]**

179A:0105 **MOV [BX+200],AL**

179A:0109 **INC BX**

179A:010A **CMP BX,80**

179A:010E **JNE 103**

179A:0110 **INT 3**

179A:0111

**-U100**

**179A:0100 BB0000 MOV BX,0000**

**179A:0103 8B07 MOV AX,[BX]**

**179A:0105 88870002 MOV [BX+0200],AL**

**179A:0109 43 INC BX**

**179A:010A 81FB8000 CMP BX,0080**

**179A:010E 75F3 JNZ 0105**

**179A:0110 CC INT 3**

:

:

179A:011D 00891700 ADD [BX+DI+0017],CL

**-D0** (contenido de los 80H bytes de memoria a mover)

**179A:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 . ........O.....**

**179A:0010 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........**

**179A:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.**

**179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................**

**179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........**

**179A:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....**

**179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........**

**-D200** (contenido de los 80H bytes de memoria destino)

**179A:0200 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0220 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0230 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0250 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0260 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0270 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**-R**

AX=**0000** BX=**0000** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL ZR NA PE NC

**179A:0100 BB0000 MOV BX,0000**

**-G**

AX=**0000** BX=**0080** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=179A IP=**0110** NV UP EI PL ZR NA PE NC

**179A:0110 CC INT 3**

**-D200** (contenido actual de los 80H bytes de memoria destino)

**179A:0200 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 . ........O.....**

**179A:0210 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........**

**179A:0220 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.**

**179A:0230 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................**

**179A:0240 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0250 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........**

**179A:0260 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....**

**179A:0270 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........**

✍ Ejercicio 7: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan cargar con una tabla descendente comenzando con el valor FFH y dese la posición de memoria 0000H en adelante

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico:

**-A100**

179A:0100 **MOV BX,0**

179A:0103 **MOV CL,FF**

179A:0105 **MOV [BX],CL**

179A:0107 **DEC CX**

179A:0108 **INC BX**

179A:0109 **CMP BX,100**

179A:010D **JNE 105**

179A:010F **INT 3**

179A:0110

**-U100**

**179A:0100 BB0000 MOV BX,0000**

**179A:0103 B1FF MOV CL,FF**

**179A:0105 880F MOV [BX],CL**

**179A:0107 49 DEC CX**

**179A:0108 43 INC BX**

**179A:0109 81FB0001 CMP BX,0100**

**179A:010D 75F6 JNZ 0105**

**179A:010F CC INT 3**

:

:

179A:011E 8917 MOV [BX],DX

**-d0** (volcado de la memoria antes de ejecutar el programa)

**179A:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 . ........O.....**

**179A:0010 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........**

**179A:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.**

**179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................**

**179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........**

**179A:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....**

**179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........**

**-d** (volcado de la memoria antes de ejecutar el programa)

**179A:0080 00 0D 42 4C 41 53 54 45-52 3D 41 32 32 30 20 49 ..BLASTER=A220 I**

**179A:0090 35 20 44 31 20 50 33 33-30 20 54 33 0D 74 2E 65 5 D1 P330 T3.t.e**

**179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............**

**179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**179A:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

**-R**

AX=0000 BX=**0000** CX=**0000** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**0100** NV UP EI PL NZ NA PO NC

**179A:0100 BB0000 MOV BX,0000**

**-G**

AX=0000 BX=**0100** CX=**FFFF** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=**179A** ES=179A SS=179A CS=**179A** IP=**010F** NV UP EI PL ZR NA PE NC

**179A:010F CC INT 3**

**-D0**

**179A:0000 FF FE FD FC FB FA F9 F8-F7 F6 F5 F4 F3 F2 F1 F0 ................**

**179A:0010 EF EE ED EC EB EA E9 E8-E7 E6 E5 E4 E3 E2 E1 E0 ................**

**179A:0020 DF DE DD DC DB DA D9 D8-D7 D6 D5 D4 D3 D2 D1 D0 ................**

**179A:0030 CF CE CD CC CB CA C9 C8-C7 C6 C5 C4 C3 C2 C1 C0 ................**

**179A:0040 BF BE BD BC BB BA B9 B8-B7 B6 B5 B4 B3 B2 B1 B0 ................**

**179A:0050 AF AE AD AC AB AA A9 A8-A7 A6 A5 A4 A3 A2 A1 A0 ................**

**179A:0060 9F 9E 9D 9C 9B 9A 99 98-97 96 95 94 93 92 91 90 ................**

**179A:0070 8F 8E 8D 8C 8B 8A 89 88-87 86 85 84 83 82 81 80 ................**

**-D**

**179A:0080 7F 7E 7D 7C 7B 7A 79 78-77 76 75 74 73 72 71 70 .~}|{zyxwvutsrqp**

**179A:0090 6F 6E 6D 6C 6B 6A 69 68-67 66 65 64 63 62 61 60 onmlkjihgfedcba`**

**179A:00A0 5F 5E 5D 5C 5B 5A 59 58-57 56 55 54 53 52 51 50 \_^]\[ZYXWVUTSRQP**

**179A:00B0 4F 4E 4D 4C 4B 4A 49 48-47 46 45 44 43 42 41 40 ONMLKJIHGFEDCBA@**

**179A:00C0 3F 3E 3D 3C 3B 3A 39 38-37 36 35 34 33 32 31 30 ?>=<;:9876543210**

**179A:00D0 2F 2E 2D 2C 2B 2A 29 28-27 26 25 24 23 22 21 20 /.-,+\*)('&%$#"!**

**179A:00E0 1F 1E 1D 1C 1B 1A 19 18-17 16 15 14 13 12 11 10 ................**

**179A:00F0 0F 0E 0D 0C 0B 0A 09 08-07 06 05 04 03 02 01 00 ................**

✍ Ejercicio 8: **Modo de direccionamiento indexado, por registro y de control de proceso.**

a) Escribir una secuencia de instrucciones que permitan cargar con una tabla descendente comenzando con el valor FFH y dese la posición de memoria 0000H en adelante

b) Escriba el mismo programa utilizando la herramienta DEBUG.

c) Utilice los comandos R,A,U,D,T,E y G para controlar y ejecutar el programa. Verifique el resultado esperado.

Carga de lenguaje nemotécnico:

**AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000**

**DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC**

**179A:0100 0000 ADD [BX+SI],AL DS:0000=CD**

**-A100**

179A:0100 **MOV BX,40**

179A:0103 **XOR AL,AL**

179A:0105 **MOV [BX],AL**

179A:0107 **INC BX**

179A:0108 **MOV CL,[BX]**

179A:010A **INC BX**

179A:010B **MOV AL,[BX]**

179A:010D **RCR AL,1**

179A:010F **JNC 119**

179A:0111 **MOV AL,[40]**

179A:0114 **INC AL**

179A:0116 **MOV [40],AL**

179A:0119 **DEC CL,1**

179A:011B **JNZ 10A**

179A:011D **INT 3**

179A:011E

**-U100**

**179A:0100 BB4000 MOV BX,0040**

**179A:0103 30C0 XOR AL,AL**

**179A:0105 8807 MOV [BX],AL**

**179A:0107 43 INC BX**

**179A:0108 8A0F MOV CL,[BX]**

**179A:010A 43 INC BX**

**179A:010B 8A07 MOV AL,[BX]**

**179A:010D D0D8 RCR AL,1**

**179A:010F 7308 JNB 0119**

**179A:0111 A04000 MOV AL,[0040]**

**179A:0114 FEC0 INC AL**

**179A:0116 A24000 MOV [0040],AL**

**179A:0119 FEC9 DEC CL**

**179A:011B 75ED JNZ 010A**

**179A:011D CC INT 3**

**179A:011E 8917 MOV [BX],DX**

**-D40**

**179A:0040 FF 06 01 02 03 04 05 06-00 00 00 00 00 00 00 00 ................**

:

:

179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

Se ejecutará la secuencia de instrucciones con el comando paso a paso para que el alumno pueda observar los cambios en los registros generales, acumulador, registro de estado y posiciones de memorias involucradas.

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL ZR NA PE NC

**179A:0100 BB4000 MOV BX,0040**

**-T**

AX=0000 BX=**0040** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL ZR NA PE NC

**179A:0103 30C0 XOR AL,AL**

**-T**

AX=0000 BX=0040 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0105 NV UP EI PL ZR NA PE NC

**179A:0105 8807 MOV [BX],AL DS:0040=FF**

**-T**

AX=0000 BX=0040 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0107 NV UP EI PL ZR NA PE NC

**179A:0107 43 INC BX**

**-T**

AX=0000 BX=0041 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0108 NV UP EI PL NZ NA PE NC

**179A:0108 8A0F MOV CL,[BX] DS:0041=06**

**-T**

AX=0000 BX=0041 CX=**0006** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PE NC

**179A:010A 43 INC BX**

**-T**

AX=0000 BX=**0042** CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PE NC

**179A:010B 8A07 MOV AL,[BX] DS:0042=01**

**-T**

AX=00**01** BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PE NC

**179A:010D D0D8 RCR AL,1**

**-T**

AX=0000 BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F NV UP EI PL NZ NA PE **CY**

**179A:010F 7308 JNB 0119**

**-T**

AX=0000 BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0111 NV UP EI PL NZ NA PE CY

**179A:0111 A04000 MOV AL,[0040] DS:0040=00**

**-T**

AX=**0000** BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0114 NV UP EI PL NZ NA PE CY

**179A:0114 FEC0 INC AL**

**-T**

AX=00**01** BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0116 NV UP EI PL NZ NA PO CY

**179A:0116 A24000 MOV [0040],AL DS:0040=00**

**-T**

AX=0001 BX=0042 CX=0006 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 NV UP EI PL NZ NA PO CY

**179A:0119 FEC9 DEC CL**

**-T**

AX=0001 BX=0042 CX=00**05** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL NZ NA PE CY

**179A:011B 75ED JNZ 010A**

**-T**

AX=0001 BX=0042 CX=0005 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PE CY

**179A:010A 43 INC BX**

**-T**

AX=0001 BX=**0043** CX=0005 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PO CY

**179A:010B 8A07 MOV AL,[BX] DS:0043=02**

**-T**

AX=00**02** BX=0043 CX=0005 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PO CY

**179A:010D D0D8 RCR AL,1**

**-T**

AX=00**81** BX=0043 CX=0005 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F OV UP EI PL NZ NA PO **NC**

**179A:010F 7308 JNB 0119**

**-T**

AX=0081 BX=0043 CX=0005 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 OV UP EI PL NZ NA PO NC

**179A:0119 FEC9 DEC CL**

**-T**

AX=0081 BX=0043 CX=00**04** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL NZ NA PO NC

**179A:011B 75ED JNZ 010A**

**-T**

AX=0081 BX=0043 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PO NC

**179A:010A 43 INC BX**

**-T**

AX=0081 BX=**0044** CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PE NC

**179A:010B 8A07 MOV AL,[BX] DS:0044=03**

**-T**

AX=00**03** BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PE NC

**179A:010D D0D8 RCR AL,1**

**-T**

AX=00**01** BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F NV UP EI PL NZ NA PE **CY**

**179A:010F 7308 JNB 0119**

**-T**

AX=0001 BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0111 NV UP EI PL NZ NA PE **CY**

**179A:0111 A04000 MOV AL,[0040] DS:0040=01**

**-T**

AX=00**01** BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0114 NV UP EI PL NZ NA PE CY

**179A:0114 FEC0 INC AL**

**-T**

AX=00**02** BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0116 NV UP EI PL NZ NA PO CY

**179A:0116 A24000 MOV [0040],AL DS:0040=01**

**-T**

AX=0002 BX=0044 CX=0004 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 NV UP EI PL NZ NA PO CY

**179A:0119 FEC9 DEC CL**

**-T**

AX=0002 BX=0044 CX=00**03**  DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL NZ NA PE CY

**179A:011B 75ED JNZ 010A**

**-T**

AX=0002 BX=0044 CX=0003 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PE CY

**179A:010A 43 INC BX**

**-T**

AX=0002 BX=00**45** CX=0003 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PO CY

**179A:010B 8A07 MOV AL,[BX] DS:0045=04**

**-T**

AX=00**04** BX=0045 CX=0003 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PO **CY**

**179A:010D D0D8 RCR AL,1**

**-T**

AX=00**82** BX=0045 CX=0003 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F OV UP EI PL NZ NA PO **NC**

**179A:010F 7308 JNB 0119**

**-T**

AX=0082 BX=0045 CX=0003 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 OV UP EI PL NZ NA PO NC

**179A:0119 FEC9 DEC CL**

**-T**

AX=0082 BX=0045 CX=00**02** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL NZ NA PO NC

**179A:011B 75ED JNZ 010A**

**-T**

AX=0082 BX=0045 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PO NC

**179A:010A 43 INC BX**

**-T**

AX=0082 BX=**0046** CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PO NC

**179A:010B 8A07 MOV AL,[BX] DS:0046=05**

**-T**

AX=00**05** BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PO NC

**179A:010D D0D8 RCR AL,1**

**-T**

AX=00**02** BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F NV UP EI PL NZ NA PO **CY**

**179A:010F 7308 JNB 0119**

**-T**

AX=0002 BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0111 NV UP EI PL NZ NA PO CY

**179A:0111 A04000 MOV AL,[0040] DS:0040=02**

**-T**

AX=00**02** BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0114 NV UP EI PL NZ NA PO CY

**179A:0114 FEC0 INC AL**

**-T**

AX=00**03** BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0116 NV UP EI PL NZ NA PE CY

**179A:0116 A24000 MOV [0040],AL DS:0040=02**

**-T**

AX=00**03** BX=0046 CX=0002 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 NV UP EI PL NZ NA PE CY

**179A:0119 FEC9 DEC CL**

**-T**

AX=0003 BX=0046 CX=00**01** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL NZ NA PO CY

**179A:011B 75ED JNZ 010A**

**-T**

AX=0003 BX=0046 CX=0001 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010A NV UP EI PL NZ NA PO CY

**179A:010A 43 INC BX**

**-T**

AX=0003 BX=00**47** CX=0001 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010B NV UP EI PL NZ NA PE CY

**179A:010B 8A07 MOV AL,[BX] DS:0047=06**

**-T**

AX=00**06** BX=0047 CX=0001 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010D NV UP EI PL NZ NA PE CY

**179A:010D D0D8 RCR AL,1**

**-T**

AX=00**83** BX=0047 CX=0001 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=010F OV UP EI PL NZ NA PE **NC**

**179A:010F 7308 JNB 0119**

**-T**

AX=0083 BX=0047 CX=0001 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0119 OV UP EI PL NZ NA PE **NC**

**179A:0119 FEC9 DEC CL**

**-T**

AX=0083 BX=0047 CX=00**00** DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011B NV UP EI PL ZR NA PE NC

**179A:011B 75ED JNZ 010A**

**-T**

AX=0083 BX=0047 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=011D NV UP EI PL ZR NA PE NC

**179A:011D CC INT 3**

**-D40**

**179A:0040 03** 06 01 02 03 04 05 06-00 00 00 00 00 00 00 00 ................

:

:

**179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................**

Anexo I – Set de Instrucciones

**INSTRUCCIONES DE MOVIMIENTO DE DATOS**

MOV destino, origen (MOVe, mover)

XCHG destino, origen (eXCHanGe, intercambiar)

**OPERACIONES ARITMETICAS**

ADD destino, origen (ADDition, sumar) {O,S,Z,A,P,C}

ADC destino, origen (ADdition with Carry, sumar con acarreo) {O,S,Z,A,P,C}

STC (SeT Carry)

CLC (CLear Carry)

INC destino (INCrement, incrementar) {O,S,Z,A,P}

SUB destino, origen (SUBstract, resta) {O,S,Z,A,P,C}

SBB destino, origen (SuBstract with Borrow, restar con llevada) {O,S,Z,A,P,C}

DEC destino (DECrement, decrementar) {O,S,Z,A,P}

IMUL origen (Integer MULtiplication, multiplicación entera con signo) {O,C}

MUL origen (MULtiplication, multiplicación entera sin signo) {O,C}

IDIV origen (Integer DIVide, división entera con signo)

DIV origen (DIVide, división entera sin signo)

**INSTRUCCIONES DE SOPORTE ARITMÉTICO**

CWD (Convert Word to Doubleword, convertir palabra a palabra doble)

CQD (Convert Doubleword to Quad-word, convertir palabra doble a palabra cuádruple)

CBW (Convert Byte to Word, convertir byte a palabra)

CWDE (Convert Word to Doubleword Extended, convertir palabra a palabra doble extendida)

MOVSX destino, origen (Move with Sign Extension, mover con extensión de signo)

MOVZX destino, origen (Move with Zero Extension, mover con extensión de ceros)

NEG destino (NEGate, negar){O,S,Z,A,P,C}

**INSTRUCCIONES LÓGICAS**

AND destino, origen

OR destino, origen

XOR destino, origen

NOT destino

**DESPLAZAMIENTOS Y ROTACIONES**

SAL destino, origen (Shift Arithmetic Left, desplazamiento aritmético a la izquierda) {O,S,Z,P,C}

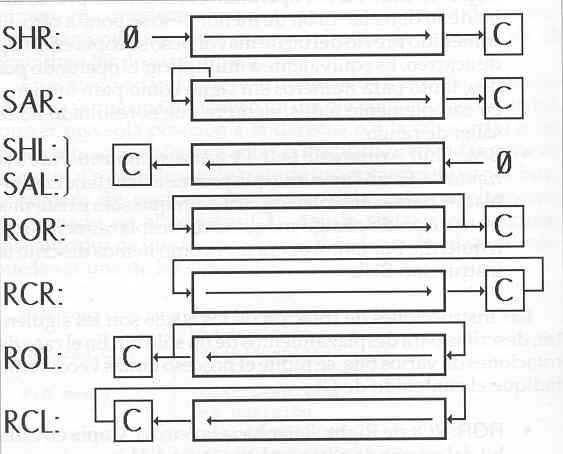
SAR destino, origen (Shift Arithmetic Right, desplazamiento aritmético a la derecha) {O,S,Z,P,C}

SHL destino, origen (SHift logic Left, desplazamiento lógico a la izquierda) {O,S,Z,P,C}

SHR destino, origen (SHift logic Right, desplazamiento lógico a la derecha) {O,S,Z,P,C}

ROL destino, origen (ROtate Left, rotación a la izquierda) {O,C}

ROR destino, origen (ROtate Right, rotación a la derecha) {O,C}

****

**INSTRUCCIONES DE COMPARACIÓN**

CMP operando1, operando2 (CoMPare, comparar) {O,S,Z,A,P,C}

TEST operando1, operando2 (TEST, comprobar) {O,S,Z,A,P,C}

**INSTRUCCIONES DE SALTO**

JMP dirección (JuMP, saltar)

JCC dirección

|  |  |  |  |
| --- | --- | --- | --- |
| Instrucción |  |  | Condición |
| JZ | Jump if Zero | salta si cero | ZF=1 |
| JNZ | Jump if Not Zero | salta si no cero | ZF=0 |
| JC | Jump if Carry | salta si acarreo | CF=1 |
| JNC | Jump if Not Carry | salta si no acarreo | CF=0 |
| JO | Jump if Overflow | salta si overflow | OF=1 |
| JNO | Jump if Not Overflow | salta si no overflow | OF=0 |
| JS | Jump if Sign | salta si signo | SF=1 |
| JNS | Jump if Not Sign | salta si no signo | SF=0 |
| JP/JPE | Jump if Parity (Parity Even) | salta si paridad (Paridad Par) | PF=1 |
| JNP/JPO | Jump if Not Parity (Parity Odd) | salta si no paridad (Paridad Par) | PF=0 |
| JA | Jump if Above | salta si por encima | A>B (sin signo) |
| JAE | Jump if Above or Equal | salta si por encima o igual | A>=B (sin signo) |
| JB | Jump if Below | salta si por debajo | A<B (sin signo) |
| JBE | Jump if Below or Equal | salta si por debajo o igual | A<=B (sin signo) |
| JE | Jump if Equal | salta si igual | A=B |
| JG | Jump if Greater | salta si mayor | A>B (con signo) |
| JGE | Jump if Greater or Equal | salta si mayor o igual | A>=B (con signo) |
| JL | Jump if Less | salta si menor | A<B (con signo) |
| JLE | Jump if Less or Equal | salta si menor o igual | A<=B (con signo) |

LOOP dirección (LOOP, ciclo)

LOOPZ/LOOPE

LOOPNZ/LOOPNE

**MANEJO DE LA PILA**

PUSH origen (empujar)

POP destino (sacar).

PUSHA y POPA (de PUSH All y POP All)

PUSHAD y PUSHF y POPF (PUSH Flags y POP Flags)

**MANEJO DE SUBRUTINAS**

CALL dirección (CALL, llamar)

RET, IRET (RETurn, regresar)

RETN (Near)

RETF (Far);

**INSTRUCCIONES PARA OBTENER DIRECCIONES**

LEA destino, origen (Load Effective Address, cargar dirección efectiva)

LDS destino, origen (Load pointer using DS, cargar puntero usando DS)

LES, LFS, LGS y LSS

**INSTRUCCIONES DE MANEJO DE BITS**

BT/BTS/BTR/BTC registro, operando (Bit Test/Test&Set/Test&Reset/Test&Complement

**INSTRUCCIONES DE CADENA**

STD (SeT Direction flag) hacemos DF=1,

CLD (CLear Direction flag) DF=0

LODSB y LODSW (LOaD String, Byte y LOad String, Word

STOSB y STOSW (STOre String, Byte/Word)

MOVSB y MOVSW (MOV String, Byte/Word)

SCASB y SCASW (SCAn String)

CMPSB y CMPSW (CoMPare String)

REPE/REPZ y REPNE/REPNZ

STOSD, LODSD, MOVSD, SCASD, CMPSD.

**INSTRUCCIONES BCD**

AAA (ASCII Adjust AX After Addition) {A,C}

DAA (Decimal Adjust AL after Addition) {S,Z,A,P,C}

AAS (Adjust AX After Subtraction) {A,C}

DAS (Decimal Adjust AL after Subtraction) {S,Z,A,P,C}

AAM (ASCII Adjust AX After Multiply) {S,Z,P}

AAD (ASCII Adjust AX Before Division) {S,Z,P}

**MISCELÁNEA**

HLT (HaLT, parada)

NOP (No OPeration, no hacer nada)

INT inmediato

INTO

Anexo II – Comandos Básicos de DEBUG.EXE

**?** Ayuda de debug

**R** (Examinar el contenido de los Registros)

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

**R [registro]** (Cambia el contenido del registro)

**-RAX**

AX 0000 (Valor actual del registro solicitado)

:**1FED** (Se carga manualmente)

**-R**

AX=**1FED** BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

**D [dirección]** (examina contenido de las 128 posiciones de memoria consecutivas)

**-D0**

179A:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 . ........O.....

179A:0010 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........

179A:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.

179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........

179A:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....

179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........

**-D**

179A:0080 00 0D 42 4C 41 53 54 45-52 3D 41 32 32 30 20 49 ..BLASTER=A220 I

179A:0090 35 20 44 31 20 50 33 33-30 20 54 33 0D 74 2E 65 5 D1 P330 T3.t.e

179A:00A0 78 65 0D 00 00 00 00 00-00 00 00 00 00 00 00 00 xe..............

179A:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

**E + dirección** (Carga manualmente el contenido de la memoria)

**-R**

AX=1FED BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=11

**-D0**

179A:0000 11 20 FF 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 . ........O.....

179A:0010 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........

179A:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.

179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........

179A:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....

179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........

**-E0**

179A:0000 11.**11** 20.**11**  FF.**31**  9F.

Contenido Actual

Nuevo valor

Barra espaciadora pasa al siguiente byte

**-D0**

179A:0000 11 11 31 9F 00 9A EE FE-1D F0 4F 03 FE 11 8A 03 ..1.......O.....

179A:0010 FE 11 17 03 FE 11 28 07-01 01 01 00 02 FF FF FF ......(.........

179A:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 9E 11 4E 01 ..............N.

179A:0030 BE 16 14 00 18 00 9A 17-FF FF FF FF 00 00 00 00 ................

179A:0040 05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

179A:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!...........

179A:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....

179A:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 ........

**A [dirección] (Carga Assembler Lenguaje Nemotécnico)**

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 0000 ADD [BX+SI],AL DS:0000=CD

**-A**

179A:0100 **MOV AX,0**

179A:0103

**-A1FF**

179A:01FF **MOV BX,FF**

179A:0202

**U [dirección] (examina Assembler con sus códigos de maquina óptimos)**

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 B80000 MOV AX,0000

**-U**

179A:0100 **B80000 MOV AX,0000**

179A:0103 0000 ADD [BX+SI],AL

:

:

179A:011D 00891700 ADD [BX+DI+0017],CL

**G [=dirección] (ejecuta desde donde apunta el IP)**

**-U100**

179A:**0100 B8EEFF MOV AX,FFEE**

179A:**0103 CC** **INT 3**

179A:0104 0000 ADD [BX+SI],AL

:

:

179A:011E 8917 MOV [BX],DX

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 **B8EEFF MOV AX,FFEE**

**-G**

AX=FFEE BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL NZ NA PO NC

179A:0103 **CC INT 3**

**-U100**

179A:0100 **B8FFFF MOV AX,FFFF**

179A:0103 **BBFFFF MOV BX,FFFF**

179A:0106 **CC INT 3**

179A:0107 0000 ADD [BX+SI],AL

:

:

179A:011D 00891700 ADD [BX+DI+0017],CL

**-RAX**

AX FFFF

**:0**

**-RBX**

BX FFFF

**:0**

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

**-G=100**

AX=FFFF BX=FFFF CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

**T [=dirección] (ejecuta paso a paso)**

AX=FFFF BX=FFFF CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL NZ NA PO NC

179A:0103 **BBFFFF MOV BX,FFFF**

**-RAX**

AX FFFF

**:0**

**-RBX**

BX FFFF

**:0**

**-RIP**

IP 0103

**:100**

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0100 NV UP EI PL NZ NA PO NC

179A:0100 **B8FFFF MOV AX,FFFF**

**-T**

AX=FFFF BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL NZ NA PO NC

179A:0103 **BBFFFF MOV BX,FFFF**

**-T**

AX=FFFF BX=FFFF CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL NZ NA PO NC

179A:0103 **BBFFFF MOV BX,FFFF**

**-T**

AX=**FFFF** BX=**FFFF**  CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

**-RAX**

AX FFFF

**:0**

**-RBX**

BX FFFF

**:0**

**-R**

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

**-T=100**

AX=FFFF BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0103 NV UP EI PL NZ NA PO NC

179A:0103 **BBFFFF MOV BX,FFFF**

**-T**

AX=FFFF BX=FFFF CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=179A ES=179A SS=179A CS=179A IP=0106 NV UP EI PL NZ NA PO NC

179A:0106 **CC INT 3**

**Q (salir de Debug)**