

 UTN.BA <small>UNIVERSIDAD TECNOLÓGICA NACIONAL FACULTAD REGIONAL BUENOS AIRES</small>	Arquitectura de Computadores	Curso: K1092
	UNIDAD 10	INSTRUCCIONES - DIRECCIONAMIENTO - FORMATOS

TRABAJO PRÁCTICO - UNIDAD Nª 10

Instrucciones - Modos Direcccionamiento -

Formatos

GRUPO Nª 5

ALUMNO	LEGAJO
BUONAMICO, Leandro Elían	1776721
DI NAPOLI, Franco	2047901
PERALTA, Roque Damián	1447580
RODRIGUEZ, Iván Ángel	1782630

TP No10: Instrucciones - Modos Direcccionamiento - Formatos

Responder: (utilizando como bibliografía el Cáp.17: Repertorio de Instrucciones de Angulo)

1. Indique brevemente los 4 tipos de datos que maneja Pentium.
2. Indique brevemente los 3 grupos en los cuales se clasifican los 7 tipos de datos que pueden manejar el coprocesador matemático
3. Indique brevemente las 3 categorías en que se agrupan las formas de referenciar los operandos por el UP Pentium.
4. Explique las instrucciones privilegiadas, la razón de su uso y los 4 grupos.

5. Si AX = 3330 y DX = 4879, al ejecutarse ADD AX, DX como queda AX. Justifique.
6. Si AX = 99FF y DX = 4879, al ejecutarse SUB AX, DX como queda AX. Justifique.
7. Si AX = 99FF y DX = 4879, al ejecutarse CMP AX, DX como queda el reg. EFLAGS. Justifique
8. Explique brevemente qué hacen las instrucciones aritméticas DEC, INC, MUL e IMUL
9. Si AX = 3330 y DX = 4879, al ejecutarse AND AX, DX como queda el reg. EFLAGS. Justifique
10. Si AX = AFE4, al ejecutarse ROL AX, 1 como queda AX. Justifique
11. Explique brevemente qué hacen las instrucciones TEST, SAL y SAR.
12. Si AX = 3330 y DX = 4879, al ejecutarse SHLD AX, DX, 2 como queda AX y DX.
13. Explique cómo queda el registro PC al ejecutarse la instrucción JMP FF456781. Justifique
14. Explique brevemente qué hacen las instrucciones de salto JA, JC y JE
15. Explique brevemente qué hacen las instrucciones de transferencia de datos MOV, IN, POP y PUSH
16. Explique brevemente qué hacen las instrucciones de control de señalizadores CLC y CLI
17. Explique y dé un ejemplo de la instrucción BT
18. Explique brevemente qué hacen las instrucciones multisegmento CALL, RET, INT e IRET
19. Explique brevemente qué hacen las instrucciones HLT, LGDT, LIDT y LLDT
20. Explique brevemente que hacen la instrucción del coprocesador WAIT
21. Explique brevemente qué hacen y para que se usa la instrucción NOP

- 1) Pentium está diseñado para trabajar con todos los tipos de datos que manejan los lenguajes evolucionados, pero hay 4 fundamentales: byte (8 bits), palabra (2 bytes), doble palabra (4 bytes), cuádruple palabra (8 bytes).
- 2) Los 3 grupos en los cuales se clasifican los 7 tipos de datos que pueden manejar el coprocesador matemático son:
 - Enteros
 - Decimal empaquetado
 - Coma flotante
- 3) Las 3 categorías en que se agrupan las formas de referenciar los operandos por el UP Pentium son:
 - Modo de direccionamiento inmediato.
 - Por registro.
 - Modo de direccionamiento en memoria.
- 4) La mayoría de las instrucciones del Pentium se pueden ejecutar en cualquier nivel de privilegio, pero hay unas pocas

que pueden comprometer la integridad del sistema y uso está restringido. Uno de los grupos se denomina “Instrucciones privilegiadas”, dichas instrucciones pueden ejecutarse únicamente con el máximo nivel de privilegio (segmentos que tengan CPL=0).

Se distinguen cuatro grupos:

- 1. Cualquier instrucción que pueda modificar el campo IOPL como IRET, POPF y las relacionadas con la conmutación de tarea, como la instrucción CLTS que pone a cero el señalizador TS de tarea conmutada.
- 2. Instrucciones que afectan a los registros que hacen referencia a tablas que controlan el sistema en modo Protegido.
 - Dentro de este grupo se incluyen las siguientes instrucciones:
 - LGDT-LIDT: Sirven para cargar los registros GDTR e IDTR, respectivamente. Tanto GDTR como IDTR, sólo deben cargarse en la inicialización del sistema.
 - SGDT-SIDT: Almacenan los contenidos de GDTR e IDTR, respectivamente.
 - LLDT-LTR: Cargan a LDT y a TR, respectivamente, El operando que manejan es el selector de un descriptor apropiado, LDTR y TR deben cargarse de forma explícita durante la inicialización del sistema.
 - SLDT:STR: Almacena el valor de LDTR y TR, respectivamente y pueden ejecutarse en cualquier nivel de privilegio.
- 3. Instrucciones que afecten a la MSW, la palabra baja del CRO.

LMSW: Carga en la MSW el valor del operando, Instrucción privilegiada.

SMSW: Almacena el valor de MSW, pudiéndose ejecutar desde cualquier nivel de privilegio.

- 4. Instrucción HL. Esta instrucción lleva al procesador a un estado de parada, bloqueando la ejecución de instrucciones, del que sale mediante un RESET o una interrupción.

5) AX queda con un valor de 7BA9 en hexadecimal.

```

-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 B83033 MOV AX,3330
-t
AX=3330 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 BA7948 MOV DX,4879
-t
AX=3330 BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0106 NU UP EI PL NZ NA PO NC
073F:0106 01D0 ADD AX,DX
-t
AX=7BA9 BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0108 NU UP EI PL NZ NA PE NC
073F:0108 CD20 INT 20
-

```

El valor es debido a que la instrucción ADD suma, en este caso sumó el valor de AX con el valor de DX dando como valor 7B49 en hexadecimal.

6) AX queda con el valor 5186 expresado en Hexadecimal.

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
073F:0100 mov ax,99ff
073F:0103 mov dx,4879
073F:0106 sub ax,dx
073F:0108 int 20
073F:010A
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 B8FF99 MOV AX,99FF
-t
AX=99FF BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 BA7948 MOV DX,4879
-t
AX=99FF BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0106 NU UP EI PL NZ NA PO NC
073F:0106 29D0 SUB AX,DX
-t
AX=5186 BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0108 NU UP EI PL NZ NA PO NC
073F:0108 CD20 INT 20

```

7)

Se modificó OV = Hubo desbordamiento (overflow)

NZ = no es cero la operación

NA = no hubo acarreo auxiliar

PO = paridad

NC = no hubo acarreo

```

073F:0100 mov ax,99ff
073F:0103 mov dx,4879
073F:0106 cmp ax,dx
073F:0108 int 20
073F:010A
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 B8FF99 MOV AX,99FF
-t
AX=99FF BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 BA7948 MOV DX,4879
-t
AX=99FF BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0106 NU UP EI PL NZ NA PO NC
073F:0106 39D0 CMP AX,DX
-t
AX=99FF BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0108 NU UP EI PL NZ NA PO NC
073F:0108 CD20 INT 20

```

8)

DEC, INC, MUL e IMUL son operaciones aritméticas.

DEC: decrementa en 1 el contenido del registro o posición en memoria.

Ejemplo: DEC AX

INC: lo contrario a DEC, incrementa en 1 el contenido del registro o de una posición de memoria.

Ejemplo: INC AX;

MUL realiza una multiplicación sin signo (sin complemento a dos), mientras que **IMUL** la realiza con signo.

9) Se almacena en AX el resultado de la operación AND

AX = 0030

Mientras que la bandera afectada al final de la operación es PE indicando que es par la paridad de bits con valor 1

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
073F:0100 mov ax,3330
073F:0103 mov dx,4879
073F:0106 and ax,dx
073F:0108 int 20
073F:010A
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 B83033 MOV AX,3330
-t
AX=3330 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 B47948 MOV DX,4879
-t
AX=3330 BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0106 NU UP EI PL NZ NA PO NC
073F:0106 21D0 AND AX,DX
-t
AX=0030 BX=0000 CX=0000 DX=4879 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0108 NU UP EI PL NZ NA PE NC
073F:0108 CD20 INT 20
```

10) AX queda con el valor 5FC9

La operación ROL copia el primer 1 al carry y luego lo traslada a la última posición.

AFE4 (16) = 1010111111100100 (2)

Finalmente -> 101111111001001 (2) = 5FC9

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0100 NU UP EI PL NZ NA PO NC
073F:0100 B8E4AF MOV AX,AFE4
-t
AX=AFE4 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0103 NU UP EI PL NZ NA PO NC
073F:0103 D1C0 ROL AX,1
-t
AX=5FC9 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=073F ES=073F SS=073F CS=073F IP=0105 NU UP EI PL NZ NA PO CY
073F:0105 CD20 INT 20
```

11) TEST: Realiza la operación lógica AND de los operandos, sin resultados, solo afecta a los señalizadores.

SAL/SAR : desplazamiento aritmético a izquierda o derecha

12).

AX = 3330

DX = 4879

SHLD AX, DX, 2

AX						1	1	0	1	0	0	0	0	0	0	1	0
----	--	--	--	--	--	---	---	---	---	---	---	---	---	---	---	---	---

DX					1	0	0	1	1	0	0	0	0	1	1	1	1
----	--	--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---

AX				1	1	0	1	0	0	0	0	0	0	1	0	1	0
----	--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---

DX					1	0	0	1	1	0	0	0	0	1	1	1	1
----	--	--	--	--	---	---	---	---	---	---	---	---	---	---	---	---	---

AX = 11010000001010 = 13322

13). La instrucción JMP es una instrucción que se utiliza para desviar el flujo del programa sin tomar en cuenta las condiciones actuales de los datos ni las banderas. En este caso luego de ejecutarla, el IP tomará el valor de FF456781.

14)instrucciones de salto:

JA: Salta si el primer operando es mayor que el segundo . sin signo.

JC : salta si el bit de CARRY del EFLAGS es igual a 1 .

JE : salta si el bit Z del EFLAGS es igual a 1.

15)MOV: transfiere al primer operando el valor del segundo.

IN-OUT: entrada y salida de información desde las puertas de E/S

POP: se transfiere desde la cima de la pila una información a un registro.

PUSH: transfiere un registro a la pila

16) CLC-STC: ponen a 0 o 1 el señalizador del acarreo, respectivamente.

CLI: pone a 0 el señalizador 1. Inhabilita interrupciones enmascarables.

17) BT : asigna al señalizador CF el valor del bit del primer operando, quedando especificada su posición por el segundo operando.

18) CALL: llamada a una rutina o procedimiento.

RET: retorno de un procedimiento

INT: llamada a un programa de manejo de una interrupción

IRET: retorno de un programa de interrupción

19)HLT: detiene la ejecución de un programa

LGDT :carga el GDTR desde una posición de memoria

LIDT: carga el IDTR desde una posición de memoria

LLDT: carga el registro LDTR

20) WAIT : Se detiene el procesador x86 hasta que la patilla BUSY se desactive. Así el procesador espera al coprocesador . Esto se usaba en procesadores anteriores como el 386 que disponía de un coprocesador externo. No se usa en los Pentium que tienen integrada la FPU.

21)Operación: No opera

Operandos: Ninguno

Words: 1 (2 bytes)

Ciclos: 1

Flags: Ninguno

Descripción: NOP realiza un simple ciclo sin operar.