



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Buenos Aires

ARQUITECTURA de COMPUTADORES [08-2022]

–2023–

DOCENTE: PROF. ROBERTO TENUTA

TRABAJO PRÁCTICO GRUPAL INSTRUCCIONES Y DIRECCIONAMIENTO

«TRABAJO PRÁCTICO GRUPAL INSTRUCCIONES Y DIRECCIONAMIENTO»

Localización [MEDRANO]					Curso: K1029											
Integrantes del equipo: Herzkovich Agustín, Punta Máximo, Palazzesi Tomás, Adorno Elías, Stamati Gad																
Legajo: 213.787-2					Herzkovich Agustín											
Legajo: 214.033-0					Punta Máximo											
Legajo: 214.166-8					Stamati Gad											
Legajo: 213.961-3					Palazzesi Tomás											
Legajo: 213.445-7					Adorno Elías											
Entrega / Revisión					1		2		3							
Fecha de entrega																
Fecha de calificación																
Calificación					A	B	C	D	A	B	C	D	A	B	C	D
Firma del Docente																
OBSERVACIONES:																

TRABAJO PRÁCTICO INSTRUCCIONES Y DIRECCIONAMIENTO

1) Con una dirección de 32 bits, considerando que la memoria es direccionable al byte: ¿Cuál es el potencial espacio de direccionamiento?

2) Para la línea de código presentada a continuación

5555:0103 MOV AX, [0555]

Calcular las direcciones efectivas de la instrucción y del operando, suponiendo que CS=DS.

3) ¿Si una instrucción de salto JMP 15B, se aloja en el desplazamiento 150 del segmento de código cuantos bytes de código estará salteando cuando se ejecute?

4) Para la línea de código presentada a continuación: AB55:0103 MOV AX, [0333]

Calcular las direcciones efectivas: de la instrucción y del operando, suponiendo que CS=DS.

5) Para la línea de código presentada a continuación: AB55:0103 MOV AH, [0333]

¿Si la siguiente instrucción a la descrita se aloja en la dirección física AB656 Cual es el tamaño de la instrucción y cuantos bytes le corresponden al código de operación?

6) Si la instrucción es MOV EAX, [EBP+ESI*4] indique

- a) ¿qué registro indica la base de la estructura de dato del origen?
- b) ¿qué registro actúa de puntero a un elemento de la estructura?
- c) ¿para qué sirve el factor de escala?

7) Dado el siguiente código, conteste las preguntas

Dirección Segmentada	Código de Máquina	Simbólico de Máquina
0CED:0100	8A260002	MOV AH, [0200]
0CED:0104	8B260102	MOV BH, [0201]
0CED:0108	00FC	ADD AH, BH
0CED:010A	88260202	MOV [0202], AH

Referido a la primera instrucción Assembler

- a. ¿Cuál es la dirección segmentada que apunta al primer byte en donde, esta instrucción, esta almacenada?
- b. ¿Cuántos bytes ocupa la instrucción en código de maquina?
- c. ¿Qué significan los corchetes y que significa el numero 0200?
- d. ¿Cómo clasificaría esta instrucción?
- e. Indique el nombre del registro destino en su formato de 16 y de 32 bits.
- f. Deduzca y exprese en hexadecimal el código de operación de esta instrucción.
- g. ¿Por qué el valor del campo data de la instrucción expresada en Assembler y de la instrucción expresada en código de maquina están invertidos de a byte?
- h. Comparando las instrucciones 1 y 2 indique el motivo por el cual cambia su código de operación

- i. Comparando las instrucciones 1 y 4 indique el motivo por el cual cambia su código de operación
- j. Si tuviese que comprobar el resultado almacenado en [0202] ¿qué dirección o direcciones segmentada/s examinaría la memoria?

8) Para los siguientes registros visibles de un procesador

AX=0800 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=0CD7 ES=0CD7 SS=0CD7 CS=0CD7 IP=0102

- a) indique los registros de cálculo, registros punteros y registros de segmento.
- b) Para cada registro de segmento indique el lugar donde se aloja el respectivo desplazamiento (Código, datos y de Stack)
- c) ¿Por qué digo que la máxima longitud de un segmento es de 64 K cuando observo una dirección segmentada como las presentadas?
- d) Exprese en hexadecimal el desplazamiento asociado al primer y al último byte (suponiendo su tamaño máximo) de un segmento de este tipo
- e) Expréselo en binario.
- f) Expréselo en potencias de dos.

9) ¿Cuántos bits debe tener el desplazamiento para permitir el acceso al último byte de un segmento de 4GB?

10) ¿A qué tipo de segmento le asignaría usted un derecho de acceso “no write”?

11) i) Para el siguiente algoritmo expresado en pseudocódigo indique para cada línea el TIPO de instrucción, la instrucción y el modo de direccionamiento.

ii) Si el contenido inicial de la variable var1 es 20 ¿Cuál es su valor final?

Comienzo:

- a) mueva el valor 10 a un registro llamado r1
- b) mueva el contenido de la variable var1 a otro registro llamado r2

Calcular:

- c) aumente el valor del registro r2 en una unidad
- d) disminuya el valor de r1 en una unidad
- e) Si el valor de r1 es igual a cero, entonces vaya a Calcular, si no, siga con la próxima pseudoinstrucción

f) Mueva el contenido de r2 a la variable var1

12) Para la línea de código presentada a continuación y considerando el cálculo de la dirección física como Registro de Segmento * 10 + Desplazamiento

3BB0:0103 MOV AX, [35B0]

- a) ¿Si el valor del DS es AB55 cuál es la dirección física del operando involucrado en esta instrucción?
- b) Si el valor del SS es 3700 y el valor del SP es FFEE cuál será la dirección segmentada asociada al byte de la pila luego de una instrucción PUSH
- c) Si la instrucción siguiente a la descrita se aloja en la dirección física 3BC07, ¿Cuál es el tamaño de la instrucción y cuantos bytes le corresponden al código de operación?

13) Para el siguiente código en Assembler responder a las siguientes preguntas, teniendo en cuenta el estado inicial de los registros que también se muestra a continuación.

Código Assembler:

```
0C98:0100 90      NOP
0C98:0101 A10002   MOV    AX, [0200]
0C98:0104 BB3000   MOV    BX,0030
0C98:0107 01D8     ADD    AX, BX
0C98:0109 A30003   MOV    [0300], AX
0C98:010C CD20     INT    20
```

AX=10A1 BX=0F48 CX=95CD DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=0C98 ES=0C98 SS=0C98 CS=0C98 IP=0100 NV UP EI PL NZ NA PO NC

- a) ¿Cuál es el tamaño total del código de este programa en la memoria?
- b) ¿Qué valor toman los registros AX y BX luego de la ejecución de las instrucciones: **(Suponer que en la posición de memoria DS:0200 y en la DS:0201 están almacenados los valores 50 y 01 respectivamente)**

```
MOV    AX, [0200]
MOV    BX,0030
```

- c) Para la instrucción MOV [0300], AX calcular la dirección efectiva de la instrucción en la memoria
- d) ¿En qué instrucciones dentro de este código hay acceso a la memoria?
- e) Para la instrucción MOV BX,0030 indicar el tamaño total de la instrucción en memoria ¿Cuántos bits corresponden a la parte DATA, y al COP (OpCode)? (En base a las líneas de código de este programa.

14) Observe el contenido de los siguientes registros de CPU y responda las consignas.

CX=000A BX=0B01 IP=0008 CS=0032 SS=0090 SP=4532

- a) ¿Cuál es el valor del registro destino luego de la ejecución de una instrucción NOT CH
- b) ¿Si este, es el estado de los registros de CPU en un momento dado, cuál es el valor de la dirección segmentada de la instrucción que se está ejecutando?
- c) Si consideramos datos enteros signados para BX y CX, ¿Cuál es el valor final de BX luego de la ejecución de la instrucción SUB-BX, FFFF?
- d) Considerando el valor actual del registro CX indique cuantas veces la instrucción LOOP decrementa el registro hasta que se sale de la estructura repetitiva.

15) Suponga que un microprocesador tiene las siguientes fases de ejecución de instrucciones: 1) Fase Fetch o Búsqueda de la instrucción, 2) Decodificación, 3) Búsqueda de Operandos y 4) Ejecución.

Suponer que las fases de acceso a Memoria tardan 60 nanosegundos y las de procesamiento interno de la CPU (no acceso a memoria) tardan 20 nanosegundos.

Indicar el tiempo que tarda en ejecutarse la siguiente instrucción SUB-BX, [FF00]

Respuestas

1) Con una dirección de 32 bits, y memoria direccionable al byte, el potencial espacio de almacenamiento es 4 Gigabytes. Ya que, con una dirección de 32 bits, se tienen 2^{32} locaciones, lo que equivale a 4 Giga locaciones, y aplicando la siguiente cuenta:

$$\text{Capacidad de Memoria} = \text{Cantidad de locaciones} * \text{Tamaño de cada una}$$

$$\text{Capacidad de Memoria} = 4G * 1 \text{ Byte}$$

$$\text{Capacidad de Memoria} = 4GB$$

2)

$$\text{Dirección efectiva de la instrucción} = 5555 * 10_h + 0103$$

$$\text{Dirección efectiva de la instrucción} = 55550 + 0103$$

$$\text{Dirección efectiva de la instrucción} = 55653$$

$$\text{Dirección efectiva del operando} = 5555 * 10_h + 0555$$

$$\text{Dirección efectiva del operando} = 55550 + 0555$$

$$\text{Dirección efectiva del operando} = 55AA5$$

3) Estará salteando 11 Bytes, ya que salta desde la posición 150 a la 15B.

4)

$$\text{Dirección efectiva de la instrucción} = AB55 * 10_h + 0103$$

$$\text{Dirección efectiva de la instrucción} = AB550 + 0103$$

$$\text{Dirección efectiva de la instrucción} = AB653$$

$$\text{Dirección efectiva del operando} = AB55 * 10_h + 0333$$

$$\text{Dirección efectiva del operando} = AB550 + 0333$$

$$\text{Dirección efectiva del operando} = AB883$$

5)

$$AB656 = AB55 * 10_h + \text{Desplazamiento}$$

$$\text{Desplazamiento} = AB656 - AB550$$

$$\text{Desplazamiento} = 0106$$

El tamaño de la instrucción es $0106 - 0103 = 3$ bytes.

El campo dato ocupa 2 bytes, entonces al código de operación le corresponde 1 byte.

6)

a) La base de la estructura de dato del origen la indica el registro EBP.

b) El registro que actúa de puntero a un elemento de la estructura es ESI.

c) El factor de escala sirve para ajustar la dirección de memoria a la que se accede, en este caso $ESI * 4$, porque se supone que cada elemento dentro de la estructura de

datos ocupa 4 bytes en memoria. Por lo tanto, al multiplicar ESI por 4, se obtiene el desplazamiento adecuado dentro de la estructura para acceder al elemento correcto.

7)

a) La dirección segmentada que apunta siempre al primer byte es 0CED:0100 (CS:IP).

b) Esta instrucción en código de máquina ocupa 4 bytes.

c) Al utilizar corchetes, y dentro de estos un número en el campo de datos de una instrucción, nos indica que se está accediendo a una dirección de memoria, en este caso, específicamente a la dirección 0200 ya que se encuentra dentro de los corchetes.

d) Si clasificamos la instrucción según el modo de direccionamiento, esta es directa debido a que el dato de origen se encuentra en una dirección de memoria. Es una instrucción de transferencia.

e) El nombre del registro destino es:

- AX para 16 bits.
- EAX para 32 bits.

f) El código de operación en hexadecimal es 8A26.

g) Esto se debe a que, en la memoria, los datos se almacenan en Little Endian.

h) La diferencia en el código de máquina es porque el registro en el cual está almacenando los valores son diferentes. En el primer caso se está guardando en el registro AH y en la segunda instrucción se lo está guardando en el registro BH. Y, también en el primer caso la información se busca en la posición de memoria 0200 y en el segundo caso la información la trae de la posición de memoria 0201.

i) La diferencia entre la primera instrucción y la cuarta radica en que, para la primera instrucción se mueve al registro AH los datos de la dirección de memoria 0200, mientras que en la cuarta se envía a la dirección de memoria 0202 los datos que se encuentran en el registro AH.

j) Examinaría la dirección DS:0202.

8)

a)

Registros de cálculo:

- AX.
- BX.
- CX.
- DX

Registros punteros:

- SP.
- BP.
- SI.
- DI.
- IP.

Registros de segmento:

- DS.
- ES.
- SS.
- CS

- b) CS:IP
DS:[mem]
ES:[mem]
SS:SP
- c) Porque en modo real, los desplazamientos son registros de 16 bits.
- d) 0000 primer valor y FFFF ultimo valor
- e) En binario primer valor: 0000 0000 0000 0000 y el último valor: 1111 1111 1111 1111
- f) En potencia de 2 el primero es 0 y el último $2^{16} - 1$

9) Debe tener 32 bits.

10) Se lo asignaría al Segmento de Código para evitar que el código del programa sea modificado durante su ejecución.

11)

i)

- a) `mov r1,10` //instrucción de transferencia, direccionamiento inmediato
- b) `mov r2,var1` //instrucción de transferencia, direccionamiento directo por variable
- c) `inc r2` //instrucción de incremento
- d) `dec r1` //instrucción de decremento
- e) `jz Calcular` //instrucción de salto incondicional
- f) `mov var1,r2` //instrucción de transferencia, direccionamiento directo por registro

ii) Valor final de la variable var1: 21.

12)

a)

$$\text{Dirección física} = AB55 * 10_h + 35B0$$

$$\text{Dirección física} = AEB00$$

b)

$$\text{Dirección segmentada} = 3700:FFFC$$

c)

$$3BC07 = 3BB0 * 10_h + \text{Desplazamiento}$$

$$\text{Desplazamiento} = 0107$$

La instrucción es de $0107 - 0103 = 4$ bytes. Al código de operación le corresponden 2 bytes.

13)

a) El tamaño total del código de este programa en la memoria es de 14 bytes.

b) AX = 0180, BX = 0030

c)

$$\text{Dirección física} = 0C98 * 10_h + 0109$$

$$\text{Dirección física} = 0CA89$$

d) Las instrucciones dentro de este código con acceso a memoria son:

- MOV AX,[0200]
- MOV [0300], AX

e) La instrucción tiene un tamaño total de 3 bytes. A la parte DATA le corresponden 16 bits, y a la parte OP CODE le corresponden 8 bits.

14)

a) CH = FF.

b) 0032:0008.

c) El valor final de BX es 0B02.

d) La instrucción LOOP decreuenta el registro 10 veces.

15) Tiempo que tarda para cada fase

Fase Fetch: 60 nanosegundos

Fase de Decodificación: 20 nanosegundos

Fase de Búsqueda de Operandos: 60 nanosegundos

Fase de Ejecución: 20 nanosegundos

La instrucción tarda 160 nanosegundos en ejecutarse.