

FACULTAD:	Tecnología Informática		
CARRERA:	Analista programador		
ALUMNO/A:	Mendoza Maria Florencia		
SEDE:	A distancia	LOCALIZACIÓN:	Buenos Aires
ASIGNATURA:	Sistemas de Computación II		
COMISIÓN:	B	TURNO:	A distancia
PROFESOR:	Raul Oscar Romero	FECHA:22/11/22	
TIEMPO DE RESOLUCIÓN:	120 min	EXAMEN PARCIAL N°:	2 (dos)
MODALIDAD DE RESOLUCIÓN:		A Distancia Sincrónico - Escrito	
CALIFICACIÓN:			
RESULTADOS DE APRENDIZAJE ESPERADO: Asegurar la comprensión de Interrupciones, manejo de la pila y programación en Assembler.			

Dado el siguiente estado de la UCP (los registros que no aparecen no son usados por el programa que se está ejecutando):

AX, IP, RE, SP, CS, DS, SS (tomar los valores de la planilla provista por el profesor)

Alumno										Vec:	Vec:
	AX	IP	RE	SP	CS	DS	SS	CALL	INT	IP	CS
	2345	1400	7572	7A89	8000	8000	C100	3455	66	5718	A0A0

PUNTO 1:

Si la siguiente instrucción a ejecutar es CALL nnnn que ocupa 3 bytes, **indicar:**

1.a Mediante las componentes XXXX:YYYY de la memoria, en qué dirección de memoria está la instrucción CALL nnnn y cuál es la dirección que aparecerá en el bus de direcciones al buscar esa instrucción.

CALL 3455

Los componentes son CS:IP = 8000:1400

La dirección que aparecerá en el bus de direcciones al buscar la instrucción es:

$$80000 + 1400 = 81400$$

1.b Qué movimientos de información ocurren durante la ejecución de CALL nnnn y con qué valores quedan los registros involucrados.

Los movimientos de información que ocurren durante la ejecución CALL 3455 y los valores que quedan son:

- $IP = IP + 3 = 1400 + 3 = 1403$ (IP apunta a la próxima instrucción, la instrucción que le sigue a call 3455. Se le suma 3 porque la instrucción call 3455 ocupa 3 bytes)
- $SP \leftarrow SP - 2$ ($SP = 7A89 - 2 = 7A87$) La nueva cima de la pila es 7A87
- $[SP] \leftarrow IP$ [$7A87/8$] = 1403. Se guarda el ip desde la dirección 7A87/8
- $IP \leftarrow 3455$ ($IP = 3455$). El IP apunta a la dirección 3455, que es donde comienza la subrutina

1.c Cómo queda la pila luego de la ejecución de CALL nnnn.

SS:SP	C100:7A87	03	(IP)
	7A88	14	(IP)

PUNTO 2:

Si la subrutina empieza con PUSH AX, **indicar:**

2.a Mediante las componentes XXXX:YYYY, donde se encuentra la instrucción PUSH AX y cuál es la dirección que aparecerá en el bus de direcciones al buscarse la instrucción PUSH AX.

Los componentes son CS:IP = 8000:3455

La dirección que aparecerá en el bus de direcciones al buscar la instrucción es:

$80000 + 3455 = 83455$

2.b Qué movimientos de información ocurren durante la ejecución de PUSH AX y con qué valores quedan los registros involucrados.

Los movimientos de información que ocurren durante la ejecución PUSH AX y los valores que quedan son:

- $SP \leftarrow SP - 2$ ($SP = 7A87 - 2 = 7A85$). La nueva cima de la pila es 7A85
- $[SP] \leftarrow AX$ [$7A85/6$] = 2345. En la dirección 7A85/6 se guarda el valor del registro AX

2.c Cómo queda la pila luego de la ejecución de PUSH AX

SS:SP	C100:7A85	45	(AX)
	7A86	23	(AX)
	7A87	03	(IP)
	7A88	14	(IP)

PUNTO 3:

Si a PUSH AX (de un byte de longitud) sigue la instrucción INT nn (de dos bytes de longitud), cuyo vector de interrupciones contiene las componentes CS=CSIN e IP=IPIN, **indicar:**

3.a Mediante las componentes XXXX:YYYY, en que posición de memoria se encuentra INTnn y cómo aparecerá la misma en el bus de direcciones.

INT 66

CS=A0A0

IP=5718

La instrucción PUSH AX ocupa 1 byte de memoria. La instrucción INT 66 ocupa 2 bytes de memoria. La instrucción PUSH AX comienza en CS:IP = 8000:3455. La instrucción INT 66 comenzara en 8000:3456

La dirección que aparecerá en el bus de direcciones al buscar la instrucción es:

80000 + 3456 = 83456

3.b Qué movimientos de información se realizan durante su ejecución para poder resguardar información en la Pila.

- $SP \leftarrow SP - 2$ ($SP=7A85-2 = 7A83$). La nueva cima de la pila es 7A83
- $[SP] \leftarrow RE$ [$7A83/4$] = 7572. En la dirección 7A83/4 se guarda el valor del registro RE
- $SP \leftarrow SP - 2$ ($SP=7A83-2 = 7A81$). La nueva cima de la pila es 7A81
- $[SP] \leftarrow CS$ [$7A81/2$] = 8000. En la dirección 7A81/2 se guarda el valor del registro CS
- $IP \leftarrow IP+2$ ($3456+2=3458$). INT 66 ocupa 2 bytes de memoria y cuando se pidió INT 66, el IP estaba en 3456. Ahora el IP apunta a la próxima instrucción que sigue a la interrupción, a la cual se debe retomar cuando finalice la subrutina de SO llamada por INT 66.
- $SP \leftarrow SP - 2$ ($SP=7A81-2 = 7A7F$). La nueva cima de la pila es 7A7F
- $[SP] \leftarrow IP$ [$7A7F/0$] = 3458. En la dirección 7A7F/0 se guarda el valor del IP

3.c Cómo queda la pila luego de los movimientos de información que dispara la ejecución de INTnn.

SS:SP	C100:7A8F	58	(IP)
	7A80	34	(IP)
	7A81	00	(CS)
	7A82	80	(CS)
	7A83	72	(RE)
	7A84	75	(RE)
	7A85	45	(AX)
	7A86	23	(AX)
	7A87	03	(IP)
	7A88	14	(IP)

3.d Cuáles son las acciones que faltan realizar para concluir con la ejecución de INTnn en relación con el flag I y la localización de la subrutina del SO que atiende INTnn. Describir la ubicación del vector de interrupción dentro de la Tabla de vectores.

Mientras se guardó en la pila el contenido del registro RE que existía antes de la ejecución INT 66, durante la ejecución, RE va a cambiar ya que el flag I que forma parte de RE, debe pasar de 1 a 0, deshabilitando cualquier interrupción por hardware que ocurra mientras se ejecuta la subrutina de INT 66. Para localizar a INT 66, está compuesta por el segmento 0000 (al igual que cualquier interrupción) y en el componente derecho es calculado de la siguiente manera:

$$66 \times 4 = 198.$$

Ósea que la dirección del vector es: 0000:0198

Y de acuerdo a los datos otorgados en la planilla, en la zona de vectores se tendrá:

0000: 0198 18 (IP)

0000: 0199 57 (IP)

0000: 019A A0 (CS)

0000: 019B A0 (CS)

Entonces CS:IP = A0A0:5718. Es la dirección donde empieza la subrutina del SO que atiende INT 66

3.e Con qué instrucción termina esta rutina que atiende la Interrupción por software, y qué movimientos de información genera su ejecución.

La instrucción que atiende la interrupción INT 66 debe terminar con IRET

Los movimientos de información que genera su ejecución son:

- $IP \leftarrow [SP]$ (IP = 3458)
- $SP \leftarrow SP + 2$ (SP=7A7F + 2 = 7A81). La nueva cima de la pila es 7A81
- $CS \leftarrow [SP]$ (CS = 8000)
- $SP \leftarrow SP + 2$ (SP=7A81+2 = 7A83). La nueva cima de la pila es 7A83
- $RE \leftarrow [SP]$ (RE = 7572)
- $SP \leftarrow SP + 2$ (SP=7A83+2 = 7A85). SP apunta al contenido de AX guardado en la pila

Como ahora los valores de CS:IP son 8000:3458, la próxima instrucción a ejecutar será la que sigue a INT 66. Luego de la ejecución de IRET, debe ser I = 1 para que las interrupciones por hardware sigan su curso.

3.f Cómo queda la pila cuando finaliza la ejecución de esta subrutina del SO.

SS:SP	C100:7A85	45	(AX)
	7A86	23	(AX)
	7A87	03	(IP)
	7A88	14	(IP)

3.g Se puede enmascarar este tipo de interrupciones?

Int 66 al ser una interrupción por software, no puede enmascararse ya que es una instrucción que ejecuta la UC.

PUNTO 4:

Puesto que la subrutina empezó con PUSH AX e INTnn, **indicar:**

4.a Con qué instrucciones debe terminar.

Las instrucciones con las que debe terminar son:

POP AX

RET

4.b Qué movimientos ocurren cuando se ejecuta cada una, y cómo queda la pila luego de cada ejecución.

Ejecución POP AX

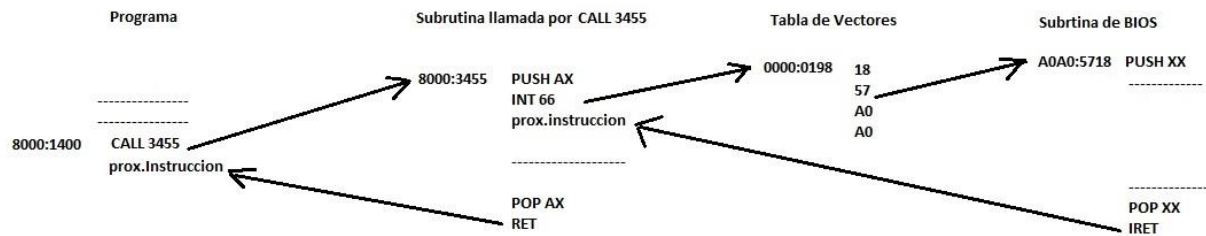
- $AX \leftarrow [SP]$ (Se guarda en AX lo que está apuntando SP, es decir $AX = 2345$)
- $SP \leftarrow SP + 2$ ($SP = 7A85 + 2 = 7A87$). La nueva cima de la pila

Ejecución RET

- $IP \leftarrow [SP]$ (Se guarda en IP lo que está apuntando SP, es decir $IP = 1403$). La próxima instrucción a ejecutar que le sigue a CALL 3455
- $SP \leftarrow SP + 2$ ($SP = 7A87 + 2 = 7A89$). La nueva cima de la pila, que es donde comenzó el ejercicio

PUNTO 5:

Realizar un esquema de la memoria, graficando el proceso llevado a cabo entre los Puntos 1 y 4 de este examen, arrancando en la posición de memoria donde se encuentra el llamado a subrutina del Punto 1, y mostrando qué recorrido realiza el procesador buscando instrucciones en la memoria, hasta volver a ese mismo punto.



PUNTO 6:

Realizar un programa en lenguaje ensamblador, que tome como dato un vector de números NATURALES/ENTEROS de 8/16 bits cuya cantidad de elementos se proporciona en la planilla (elem). El vector dato inicia en (Orig). A cada elemento se le deben sumar (sumar), para formar un vector resultado que se guarda a partir de la dirección (Dest). En caso de que la operación de error se deberá interrumpir la operación y guardar el Código de error en la dirección de error.

Tomar los datos de este programa de la planilla provista por el profesor

Alumno	Vec:Orig	Vec:Dest	Elem	bits	TIPO	Sumar	Código Error	Dir Error
	4040	5040	3	8	NAT	B0	FF	BB00

Descripción del programa:

Vectores de números naturales, de 8 bits, cantidad de elementos del vector: 3. Iniciando desde la posición 4040

A cada elemento del vector se le debe sumar B0, donde el resultado de esa suma será guardado a partir de la dirección 5040.

Si la operación da error, se interrumpe y se guardara el código de error: FF en la dirección BB00.

Inicio del programa en la posición de memoria: **2040**

Inicio de área de datos del vector en la posición de memoria: **4040**

Inicio de área de datos del vector destino en la posición de memoria: **5040**

“n” elementos del vector en la posición de memoria: **1500**

Las variables se ubicarán en las posiciones:

4040: 11

4041: 22

4042: 66

Se usarán los registros:

SI: Apuntara al elemento del vector que se le sumara B0

DI: Apuntara a la lista donde se guarda los nuevos vectores, resultantes de la suma del vector con B0

CL: Es la parte baja del CX y representa los n elementos de un vector

AL: es el registro en donde se ira guardando los resultados de las sumas de cada vector con B0

Detalle del programa cargado en DOSBOX:

- MOV SI, 4040 = SI apuntara al comienzo del vector
- MOV DI, 5040 = DI apuntara al comienzo del vector resultante de la suma de B0 por cada elemento del vector
- MOV CL, [1500] = se carga en CL (parte baja del registro CX), la cantidad de elementos del vector guardada en la posición 1500
- MOV AL, [SI] = mover el valor que apunta el registro SI a la parte baja del registro AX, es decir, se mueve el valor del vector al que este apuntando SI
- ADD AL, B0 = se le suma el valor del elemento del vector con B0
- JC 2052 = al resultado de esa suma, si carry dio 1 (se están sumando naturales), saltar a la posición 2052
- MOV [DI], AL = en caso de no dar carry la suma del elemento del vector con B0, continuar con esta instrucción. Se moverá el resultado de la suma a la otra lista apuntada por DI
- INC DI = incrementar en 1 DI (son vectores de 1 byte por elemento), para que apunte a la próxima posición y preparar esta lista para guardar otro elemento (en caso que no haya generado error en su suma ese próximo elemento)
- INC SI = incrementar en 1 SI (son vectores de 1 byte por elemento), para que apunte a la próxima posición del vector principal y continuar con el programa.
- DEC CL = decrementar CL, ya que se continua con el siguiente elemento del vector
- JNZ 204A = mientras esa decrementacion no de 0, volver a la instrucción MOV AL, [SI]
- INT 21 = finalización del programa
- MOV AL, FF = se llega hasta aquí cuando la suma del elemento del vector con B0 dio carry dio 1, se guarda en la parte baja del registro AX, el código de error FF
- JMP 2060 = y se salta a la posición 2060
- MOV [BB00], AX = mover todo el registro AX a la posición BB00, dirección que guarda el error de la suma.
- INT 21 = se finaliza el programa

Capturas del DOSBOX:

Desde **2040** se carga el programa:

New colors set.

C:\>DEBUG

-A 2040

0792:2040 MOV SI,4040

0792:2043 MOV DI,5040

0792:2046 MOV CL,[1500]

0792:204A MOV AL,[SI]

0792:204C ADD AL,B0

0792:204E JC 205C

0792:2050 MOV [DI],AL

0792:2052 INC DI

0792:2053 INC SI

0792:2054 DEC CL

0792:2056 JNZ 204A

0792:2058 INT 21

0792:205A

-A 205C

0792:205C MOV AL,FF

0792:205E JMP 2060

0792:2060

-A 2060

0792:2060 MOV [BB00],AX

0792:2063 INT 21

0792:2065 _

Desde 4040 se carga los datos:

0792:2053 INC SI

0792:2054 DEC CL

0792:2056 JNZ 204A

0792:2058 INT 21

0792:205A

-A 205C

0792:205C MOV AL,FF

0792:205E JMP 2060

0792:2060

-A 2060

0792:2060 MOV [BB00],AX

0792:2063 INT 21

0792:2065

-

-E 4040

0792:4040 00.11 00.22 66.66

-E 4040

0792:4040 11. 22. 66.

-E 1500

0792:1500 7E.03

-E 1500

0792:1500 03.

Se apunta el IP hacia el comienzo del programa:

```
-
-E 4040
0792:4040 00.11 00.22 66.66

-E 4040
0792:4040 11. 22. 66.

-E 1500
0792:1500 7E.03

-E 1500
0792:1500 03.

-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0792 ES=0792 SS=0792 CS=0792 IP=0100 NU UP EI PL NZ NA PO NC
0792:0100 0000 ADD [BX+SI],AL DS:0000=CD
-R IP
IP 0100
:2040
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0792 ES=0792 SS=0792 CS=0792 IP=2040 NU UP EI PL NZ NA PO NC
0792:2040 BE4040 MOV SI,4040
--
```

Se ejecuta el programa:

```
DS=0792 ES=0792 SS=0792 CS=0792 IP=0100 NU UP EI PL NZ NA PO NC
0792:0100 0000 ADD [BX+SI],AL DS:0000=CD
-R IP
IP 0100
:2040
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=0792 ES=0792 SS=0792 CS=0792 IP=2040 NU UP EI PL NZ NA PO NC
0792:2040 BE4040 MOV SI,4040
-T
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=4040 DI=0000
DS=0792 ES=0792 SS=0792 CS=0792 IP=2043 NU UP EI PL NZ NA PO NC
0792:2043 BF4050 MOV DI,5040
-T
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=2046 NU UP EI PL NZ NA PO NC
0792:2046 8A0E0015 MOV CL,[1500] DS:1500=03
-T
AX=0000 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=204A NU UP EI PL NZ NA PO NC
0792:204A 8A04 MOV AL,[SI] DS:4040=11
--
```

```

AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=2046  NU UP EI PL NZ NA PO NC
0792:2046 8A0E0015      MOV     CL,[1500]      DS:1500=03
-T

```

```

AX=0000 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=204A  NU UP EI PL NZ NA PO NC
0792:204A 8A04      MOV     AL,[SI]      DS:4040=11
-T

```

```

AX=0011 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=204C  NU UP EI PL NZ NA PO NC
0792:204C 04B0      ADD     AL,B0
-T

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=204E  NU UP EI NG NZ NA PO NC
0792:204E 720C      JB      205C
-T

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=2050  NU UP EI NG NZ NA PO NC
0792:2050 8805      MOV     [DI],AL      DS:5040=6C
-

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=2050  NU UP EI NG NZ NA PO NC
0792:2050 8805      MOV     [DI],AL      DS:5040=6C
-T

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5040
DS=0792 ES=0792 SS=0792 CS=0792 IP=2052  NU UP EI NG NZ NA PO NC
0792:2052 47      INC     DI
-T

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4040 DI=5041
DS=0792 ES=0792 SS=0792 CS=0792 IP=2053  NU UP EI PL NZ NA PE NC
0792:2053 46      INC     SI
-T

```

```

AX=00C1 BX=0000 CX=0003 DX=0000 SP=00FD BP=0000 SI=4041 DI=5041
DS=0792 ES=0792 SS=0792 CS=0792 IP=2054  NU UP EI PL NZ NA PE NC
0792:2054 FEC9      DEC     CL
-T

```

```

AX=00C1 BX=0000 CX=0002 DX=0000 SP=00FD BP=0000 SI=4041 DI=5041
DS=0792 ES=0792 SS=0792 CS=0792 IP=2056  NU UP EI PL NZ NA PO NC
0792:2056 75F2      JNZ     204A
-

```

AX=00C1	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=2056	NU	UP	EI PL NZ NA PO NC
0792:2056 75F2		JNZ		204A			
-T							
AX=00C1	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=204A	NU	UP	EI PL NZ NA PO NC
0792:204A 8A04		MOV		AL,[SI]		DS:4041=22	
-T							
AX=0022	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=204C	NU	UP	EI PL NZ NA PO NC
0792:204C 04B0		ADD		AL,B0			
-T							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=204E	NU	UP	EI NG NZ NA PE NC
0792:204E 720C		JB		205C			
-T							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=2050	NU	UP	EI NG NZ NA PE NC
0792:2050 8805		MOV		[DI],AL		DS:5041=38	
-							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=2050	NU	UP	EI NG NZ NA PE NC
0792:2050 8805		MOV		[DI],AL		DS:5041=38	
-T							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5041
DS=0792	ES=0792	SS=0792	CS=0792	IP=2052	NU	UP	EI NG NZ NA PE NC
0792:2052 47		INC		DI			
-T							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4041	DI=5042
DS=0792	ES=0792	SS=0792	CS=0792	IP=2053	NU	UP	EI PL NZ NA PE NC
0792:2053 46		INC		SI			
-T							
AX=00D2	BX=0000	CX=0002	DX=0000	SP=00FD	BP=0000	SI=4042	DI=5042
DS=0792	ES=0792	SS=0792	CS=0792	IP=2054	NU	UP	EI PL NZ NA PE NC
0792:2054 FEC9		DEC		CL			
-T							
AX=00D2	BX=0000	CX=0001	DX=0000	SP=00FD	BP=0000	SI=4042	DI=5042
DS=0792	ES=0792	SS=0792	CS=0792	IP=2056	NU	UP	EI PL NZ NA PO NC
0792:2056 75F2		JNZ		204A			
-							

AX=00D2 BX=0000 CX=0002 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2054 NU UP EI PL NZ NA PE NC
0792:2054 FEC9 DEC CL

-T

AX=00D2 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2056 NU UP EI PL NZ NA PO NC
0792:2056 75F2 JNZ 204A

-T

AX=00D2 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204A NU UP EI PL NZ NA PO NC
0792:204A 8A04 MOV AL,[SI] DS:4042=66

-T

AX=0066 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204C NU UP EI PL NZ NA PO NC
0792:204C 04B0 ADD AL,B0

-T

AX=0016 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204E NU UP EI PL NZ NA PO CY
0792:204E 720C JB 205C

-

AX=00D2 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204A NU UP EI PL NZ NA PO NC
0792:204A 8A04 MOV AL,[SI] DS:4042=66

-T

AX=0066 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204C NU UP EI PL NZ NA PO NC
0792:204C 04B0 ADD AL,B0

-T

AX=0016 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=204E NU UP EI PL NZ NA PO CY
0792:204E 720C JB 205C

-T

AX=0016 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=205C NU UP EI PL NZ NA PO CY
0792:205C B0FF MOV AL,FF

-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=205E NU UP EI PL NZ NA PO CY
0792:205E EB00 JMP 2060

-

```

AX=0016 BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=205C  NV UP EI PL NZ NA PO CY
0792:205C B0FF          MOV     AL,FF
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=205E  NV UP EI PL NZ NA PO CY
0792:205E EB00          JMP     2060
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2060  NV UP EI PL NZ NA PO CY
0792:2060 A300BB          MOV     [BB00],AX          DS:BB00=0000
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2063  NV UP EI PL NZ NA PO CY
0792:2063 CD21          INT     21
-T

```

Se muestra que en la posición queda guardado los nuevos elementos del resultado de la suma con B0 (dos elementos no dieron erro) y en BB00 el código de error de la suma del tercer elemento (el tercer elemento si dio carry 1)

```

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=205E  NV UP EI PL NZ NA PO CY
0792:205E EB00          JMP     2060
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2060  NV UP EI PL NZ NA PO CY
0792:2060 A300BB          MOV     [BB00],AX          DS:BB00=0000
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00FD BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=0792 IP=2063  NV UP EI PL NZ NA PO CY
0792:2063 CD21          INT     21
-T

AX=00FF BX=0000 CX=0001 DX=0000 SP=00F7 BP=0000 SI=4042 DI=5042
DS=0792 ES=0792 SS=0792 CS=F000 IP=14A0  NV UP DI PL NZ NA PO CY
F000:14A0 FB          STI
-
-E 5040
0792:5040 C1.      D2.

-E BB00
0792:BB00 FF.      00._

```