



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

### **PROBLEMA. 1**

**(2.5 Puntos)**

Represente, en la tabla que aparece a continuación, microinstrucciones, microórdenes y operaciones básicas de la secuencia de instrucciones:

**SHR A, JNZ etiqueta (si no se cumple la condición) y NOP.**

Instrucción	μInstrucciones	μÓrdenes	Operación Básica



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

### PROBLEMA. 1

(cont.)

### SOLUCIÓN

Instrucción	μInstrucciones	μÓrdenes	Operación Básica
SHR A	PC→ABUS	salpc	F
	(ABUS)→DBUS	mem, rd	
	DBUS→CO	ckco	
	PC++	reg1, ckpch, ckpcl	
	A >sa>→A	alu3, alu2, alu1, cka	I <sub>talu</sub>
JNZ addr (Si F <sub>Z</sub> =1)	PC→ABUS	salpc	F
	(ABUS)→DBUS	mem, rd	
	DBUS→CO	ckco	
	PC++	reg1, ckpch, ckpcl	
	PC++	reg1, ckpch, ckpcl	I <sub>id16</sub>
NOP	PC→ABUS	salpc	F
	(ABUS)→DBUS	mem, rd	
	DBUS→CO	ckco	
	PC++	reg1, ckpch, ckpcl	



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

### PROBLEMA. 2

(2.5 Puntos)

Inicialmente el registro A del Indalo III contiene cierto número  $a_7a_6a_5a_4a_3a_2a_1a_0$ .

1. Desarrolla, en lenguaje simbólico, la secuencia de operaciones necesarias para que se active el flag de cero ( $F_Z = 1$ ) si y solo si se cumplen simultáneamente las dos igualdades siguientes:

$$a_4 = a_5 \text{ y } a_2 \neq a_3$$

2. Obtén un programa en lenguaje ensamblador del Indalo equivalente al proceso anterior. Después de cada instrucción en ensamblador que afecte a algún registro, deberás indicar el contenido del registro destino.  
No es necesario utilizar la memoria

### SOLUCIÓN

- a) A pesar de tratarse de un único operando tendremos que utilizar dos registros

Lenguaje simbólico	Valor de A
$A \rightarrow B^*$	$a_7a_6a_5a_4a_3a_2a_1a_0$
$A \gg s \rightarrow A$	$0a_6a_5a_4a_3a_2a_1a_0$
$A \oplus B \rightarrow A$	$a_7(a_7 \oplus a_6)(a_6 \oplus a_5)(a_5 \oplus a_4)(a_4 \oplus a_3)(a_3 \oplus a_2)(a_2 \oplus a_1)(a_1 \oplus a_0)$
$A \wedge 14H \rightarrow A$	$000(a_5 \oplus a_4)0(a_3 \oplus a_2)00$
$A \oplus 4H \rightarrow A$	$000(a_5 \oplus a_4)0(a_3 \oplus a_2)00$

\*B no varía desde la primera carga.

- b) El siguiente fragmento de código saltaría a la etiqueta **eti** si se cumple la condición propuesta.

```
MOV B, A
SHR A
XOR A, B
AND A, 14H
XOR A, 4
JZ eti
```



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

---

### **PROBLEMA. 3**

**(2.5 Puntos)**

Dado el siguiente programa:

```
ORG      1000H
datos:   DB      08H, 00, N
programa: DB      56H, 00H, 10H, 6EH, 00H, 71H, 7EH, 02H, A3H
bucle:   MOV     A, (X)
         ADD     A,B
         MOV     (X), A
         MOV     A, (X+1)
         ADC     A, 0
         MOV     (X+1), A
         DEC     C
         JNZ     bucle
         HLT
```

- Indique claramente la equivalencia en hexadecimal de cada etiqueta (datos, programa, bucle).
- Obtener el contenido conocido de la memoria desde la dirección 1000H (incluyendo el ensamblado del código), **antes de ejecutar el programa**. Especifique dirección y contenido utilizando la tabla siguiente.
- Suponiendo que los contenidos de memoria son los obtenidos en el apartado anterior, y que **inicialmente el registro PC contiene 1003H**, explique brevemente la función que realiza el programa.
- Si la frecuencia del reloj del sistema es de 1GHz, calcule **N** para que el tiempo de ejecución del programa sea de 141 nsg.

NOTAS: Suponga que **N** es un entero sin signo de 8 bits  
nsg = nanosegundo =  $10^{-9}$  sg.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

### PROBLEMA. 3

(cont.)

### SOLUCIÓN

a y b)

Dirección	Contenido	Valor de las etiquetas
1000H	08	
1001H	00	<b>datos = 1000H</b>
1002H	N	<b>programa =1003H</b>
1003H	56	<b>bucle = 100CH</b>
1004H	00	
1005H	10	
1006H	6E	
1007H	00	
1008H	71	
1009H	7E	
100AH	02	
100BH	A3	
100CH	6E	
100DH	00	
100EH	8A	
100FH	49	
1010H	00	
1011H	6E	
1012H	01	
1013H	94	
1014H	00	
1015H	49	
1016H	01	
1017H	A3	
1018H	17	
1019H	F2	
101AH	0E	



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

- c) Como PC vale inicialmente 1003H, la ejecución comenzará en la posición identificada con la etiqueta **programa**. Desensamblando la parte codificada (en negrita), el código en ensamblador quedaría:

	CÓDIGO	BYTES	CICLOS	VECES
programa:	<b>MOV X, 1000H</b>	3	15	1
	<b>MOV A, (X)</b>	2	13	1
	<b>MOV B, A</b>	1	7	1
	<b>MOV C, (X+2)</b>	2	13	1
	<b>DEC C</b>	1	7	1
bucle:	MOV A, (X)	2	13	N-1
	ADD A,B	1	7	N-1
	MOV (X), A	2	13	N-1
	MOV A, (X+1)	2	13	N-1
	ADC A, 0	2	10	N-1
	MOV (X+1), A	2	13	N-1
	DEC C	1	7	N-1
	JNZ bucle	2	7/12	1/(N-2)
	HLT	1	3	1
		24	53+88·(N-1)	TOTAL

El programa pretende obtener el producto del valor almacenado en **datos** por **N**, en tamaño WORD.

- d) En la tabla anterior se incluyen también el número de bytes, ciclos y veces que se ejecuta cada instrucción, así como el total de bytes y el de ciclos, éste último en función de **N**. A partir de ese resultado: **N=2**.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

### TEST

**(2.5 Puntos)**

En las siguientes preguntas de tipo test, sólo existe una respuesta correcta. Cada pregunta tiene un valor de 0,25 ptos. Cada respuesta incorrecta descuenta 0,1 ptos. La pregunta que deje en blanco no sumará ni restará puntos.

**Notas:** • Se debe poner una X sobre la letra de la **tabla "SOLUCIÓN"** (que encontrará en la hoja siguiente) correspondiente a la respuesta correcta.

• La tabla "CALIFICACIÓN" no deberá utilizarla, es para su posterior calificación.

• No se tendrán en cuenta las respuestas que no se encuentren en dicha tabla "SOLUCIÓN"

1.- En una máquina donde el flag de *carry* se comporte como *borrow* en restas, al realizar una operación, si son uno los dos últimos acarreos, se activarán los flags:

- a)  $F_C$  y  $F_O$ .
- b) Sólo  $F_O$ .
- c) Sólo  $F_C$ .
- d) Dependerá de la operación realizada.

2.- Sobre los ciclos de máquina del Indalo 3.0 podemos decir:

- a) Todo ciclo de máquina necesita de un único ciclo de reloj.
- b) Un ciclo de máquina siempre supone un salto a subrutina.
- c) Todos requieren la utilización del bus de datos.
- d) Necesariamente utilizan la memoria.

3.- ¿Qué modo de direccionamiento permite el INDALO 1.0?:

- a) Indexado.
- b) Directo por valor (o directo).
- c) Inmediato.
- d) Indirecto por valor.

4.- Sobre las instrucciones de salto podemos decir:

- a) El salto incondicional se ensambla siempre en tres bytes.
- b) El salto condicional se ensambla siempre en tres bytes.
- c) El salto condicional necesita, al menos, 7 pulsos de reloj para su ejecución.
- d) El salto incondicional es siempre un salto corto.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **30 de Enero de 2004 (Extraordinaria)**

---

### TEST

(cont.)

5.- Para el INDALO 3.0, en la fase de **ejecución** de la instrucción ADC A, (3F3DH), además de *saldir* y *mem*, se activan las microórdenes:

- a) *rd, alu1, cka.*
- b) *wr, alu1, cka.*
- c) *rd, cka.*
- d) *rd, alu1.*

6.- Indicar cuál de las siguientes instrucciones carece de sentido en el INDALO 3.0:

- a) JMP AF.
- b) JMP X.
- c) DEC SP.
- d) PUSH AF.

7.- Señalar la frase correcta:

- a) Antes de ejecutar una llamada a una subrutina con la instrucción CALL, tenemos que salvar en la pila el contenido del PC.
- b) Después de ejecutar la instrucción POP, hay que incrementar el puntero de pila SP.
- c) La instrucción RET no incrementa el PC después de ser leída.
- d) La instrucción PUSH incrementa el registro SP.

8.- De las interrupciones del INDALO 3.0 puede decirse que ...:

- a) ... son todas enmascarables.
- b) ... utilizan como vector la propia dirección de comienzo de la rutina de servicio.
- c) ... se gestionan con la ayuda de la microorden **io**.
- d) Ninguna de las anteriores respuestas es correcta.

9.- Para leer el código de operación de la instrucción SHR A se necesita una operación:

- a) F.
- b) F<sub>FLAG</sub>.
- c) F<sub>ALU</sub>.
- d) F<sub>NOINC</sub>.





UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 30 de Enero de 2004 (Extraordinaria)

**TEST**

**(cont.)**

10.- ¿Cuál de las siguientes expresiones **no** es una microinstrucción?:

- a) DBUS  $\rightarrow$  (ABUS).
- b) DIR ++.
- c) (SP)  $\rightarrow$  DBUS.
- d) 0  $\rightarrow$  F<sub>C</sub>.

**SOLUCIÓN**

1	2	3	4	5	6	7	8	9	10
a	a	a	a	a	a	a	a	a	a
b	b	b	b	b	b	b	b	b	b
c	c	c	c	c	c	c	c	c	c
d	d	d	d	d	d	d	d	d	d

La respuesta correcta se indicará tachando la letra correspondiente con una **X**.

Para anular una respuesta dada, se rodeará con una circunferencia  $\otimes$ , de modo que se entenderá:

$\otimes$	respuesta dada por el alumno.
$\otimes$	respuesta anulada por el alumno.

**CALIFICACIÓN**

Correctas		x 0,25 =	
Erróneas		x (- 0,1) =	-
Total			



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### **PROBLEMA. 1**

**(2.5 Puntos)**

Sea la siguiente secuencia de microinstrucciones:

Microinstrucciones	Microórdenes	Operaciones básicas	Instrucciones
PC → ABUS			
(ABUS) → DBUS			
DBUS → CO			
PC ++ → PC			
0 → Fc			
PC → ABUS			
(ABUS) → DBUS			
DBUS → DIRI			
PC ++ → PC			
X → ABUS			
ABUS+DIRI → ABUS			
(ABUS) → DBUS			
A+DBUS → A			
PC → ABUS			
(ABUS) → DBUS			
DBUS → CO			
PC ++ → PC			
B → DBUS			
A-DBUS → A			
PC → ABUS			
(ABUS) → DBUS			
DBUS → CO			
PC ++ → PC			
PC ++ → PC			

- Indicar las microórdenes que se activan en cada caso. *(1,2 ptos.)*
- Indicar que tipo de operación se realiza en cada conjunto de microórdenes. *(0,7 ptos.)*
- Indicar de que instrucción o instrucciones se trata. Si en algún caso no está totalmente definida, indicarlo y especificar lo más posible. *(0,6 ptos.)*



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
 ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
 Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 7 de junio de 2004. (Mañana)

### PROBLEMA. 1

(cont.)

### SOLUCIÓN

Microinstrucciones	Microórdenes	Operaciones básicas	Instrucciones
PC → ABUS	<i>salpc</i>	<i>F<sub>flag</sub></i>	<i>ADD A,(X+rel8)</i>
(ABUS) → DBUS	<i>mem,rd</i>		
DBUS → CO	<i>ckco</i>		
PC ++ → PC	<i>reg1,ckpcl,ckpch</i>		
0 → Fc	<i>quitfc</i>		
PC → ABUS	<i>salpc</i>	<i>F</i>	
(ABUS) → DBUS	<i>mem,rd</i>		
DBUS → DIRI	<i>ckdirI</i>		
PC ++ → PC	<i>reg1,ckpcl,ckpch</i>		
X → ABUS	<i>salx</i>	<i>R<sub>alu</sub></i>	
ABUS+DIRI → ABUS	<i>sumdir</i>		
(ABUS) → DBUS	<i>mem,rd</i>		
A+DBUS → A	<i>alu1,ckfl,ckfc,cka</i>		
PC → ABUS	<i>salpc</i>	<i>F</i>	<i>SBB A,B</i>
(ABUS) → DBUS	<i>mem,rd</i>		
DBUS → CO	<i>ckco</i>		
PC ++ → PC	<i>reg1,ckpcl,ckpch</i>		
B → DBUS	<i>salb</i>	<i>I<sub>talu</sub></i>	
A-DBUS → A	<i>alu3,alu1,alu0,cka,ckfl,ckfc</i>		
PC → ABUS	<i>salpc</i>	<i>F</i>	<i>Salto condicional en el que no se cumple la condición</i>
(ABUS) → DBUS	<i>mem,rd</i>		
DBUS → CO	<i>ckco</i>		
PC ++ → PC	<i>reg1,ckpcl,ckpch</i>		
PC ++ → PC	<i>reg1,ckpcl,ckpch</i>	<i>I<sub>id16</sub></i>	



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### **PROBLEMA. 2**

**(2.5 Puntos)**

En una CPU Indalo.3 se están ejecutando las siguientes instrucciones:

```
ORG 2000  
MOV B, ( X+ 0F3H)  
INT 07  
XOR A, A  
JNZ ETIQUETA
```

Las condiciones iniciales del problema son las siguientes:

- PC apunta a la dirección de la primera instrucción
- SP = 0FF00 H
- Fi = 1
- X = 1000 H

Además se conoce el contenido de las siguientes posiciones de memoria:

posición	contenido
09	00
0A	21 H
0B	0AF H
0C	52 H
0D	24 H
0E	23 H
0F	10 H
10	00
11	01
-----	-----
0FF3	32 H

Se pide:

- Dirección de la rutina de atención de INT 07 así como los valores de SP, PC, X y contenido de la pila cuando se va a empezar a ejecutar la rutina de atención a la interrupción 07
- Si cuando se está ejecutando la instrucción INT 07 se produce una interrupción con vector 05, repetir el apartado anterior pero al comenzar la rutina de atención a la interrupción 05
- Rellenar un cuadro donde aparezcan reflejados, para cada instrucción del fragmento, los ciclos de máquina, tipo de cada ciclo, así como los contenidos de los buses durante ellos. En la tabla no debe tenerse en cuenta la situación planteada en el apartado b), ni la rutina de atención a la interrupción INT 07.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### PROBLEMA. 2

(cont.)

### **SOLUCIÓN**

a) La dirección la leemos de la tabla adjunta de la siguiente forma:

$$\begin{aligned} v*2 = 14 = 0E H &\Rightarrow (0E H) = 23 H \\ v*2 + 1 = 15 = 0FH &\Rightarrow (0F H) = 10 H \end{aligned}$$

Estos dos contenidos forman la dirección 1023 H, y es en esta dirección donde empieza la rutina de atención a la interrupción.

Cuando se ejecuta INT 07 se guarda la dirección de retorno en la pila, para lo cual se decrementa dos veces el puntero de pila con lo cual

$$SP = 0FEFE H$$

Se guarda en el PC la dirección donde empieza la rutina de atención a la interrupción

$$PC = 1023 H$$

La pila queda

04	0FEFE
20	0FEFF

Se guarda la dirección de la siguiente instrucción, es decir, de la instrucción

XOR A,A

El registro X no se modifica por lo que se queda con

$$X = 1000H$$

b) Se termina de ejecutar la instrucción en curso que es INT 07.

Valor del PC

$$05*2 = 10 = 0A H \Rightarrow (0A H) = 21H$$

$$05*2 + 1 = 11 = 0B H \Rightarrow (0B H) = 0AF H$$

$$PC = 0AF21 H$$

El puntero de pila se decrementa dos veces

$$SP = 0FEFC H$$

X queda igual y la pila



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### PROBLEMA. 2

(cont.)

21	0FEFC
AF	0FEFD
04	0FEFE
20	0FEFF

c)

Instruccion	Abus	Dbus	Tipo ciclo	Acción
MOVB,(X+0F3)	2000	76	F	Lectura CO
	2001	F3	F	Lectura rel8
	0FF3	32	Ralu	(0FF3)→ A
INT 07	2002	10	F	Lectura CO
	2003	07	F	Lectura V
	FEFF	20	W	Dir.de retorno a la pila
	FEFE	04	W	
	0E	23	R	V*2
	0F	10	R	V*2+1
Rutina de atención a la interrupción				
XOR A, A	2004	F9	F	Lectura CO
JNZ ETIQUETA	2005	17	F	Lectura CO



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### PROBLEMA. 3

**(2.5 Puntos)**

Se dispone de un programa en ensamblador para el INDALO 3.0 que busca el mayor de cuatro números **sin signo** almacenados en memoria, guardándolo en el registro B:

```
ORG 1000H
JMP Inicio
Numeros: DB 1AH, 02H, 0F3H, 8DH
Inicio:  MOV X, Numeros
        MOV C, 4
        MOV B, 0
Bucle:  MOV A, (X)
        CMP A, B
        JC  Continua
        MOV B, (X+1)
Continua: INC X
        DEC C
        JNZ Bucle
HLT
```

a) Ensamblar el programa, indicando la dirección de memoria donde comienza a almacenarse cada línea del mismo

*(1,25 puntos)*

b) En el programa anterior hay una línea con un error (no sintáctico). Indicar cuál es dicha línea y en qué consiste el error.

*(0,5 puntos)*

c) Se desea cambiar el programa anterior, de forma que en lugar de tener inicialmente los cuatro datos en memoria, éstos se reciban desde el puerto **30H** y se vayan guardando en esas mismas posiciones. Para ello se intercala en la línea con la etiqueta **Inicio** una llamada a una subrutina encargada de ese cometido. Redactar dicha subrutina en lenguaje ensamblador de Indalo 3.0.

*(0,75 puntos)*



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### PROBLEMA. 3

(cont.)

### **SOLUCIÓN**

a)

	POSICIÓN	CONTENIDO
Numeros: Inicio:	1000H	12H, 07H, 10H
	1003H	1AH, 02H, 0F3H, 8DH
	1007H	56H, 03H, 10H
	100AH	7CH, 04H
Bucle:	100CH	74H, 00H
	100EH	6EH, 00H
	1010H	82H
	1011H	11H, 02H
Continua:	1013H	76H, 01H
	1015H	35H
	1016H	A3H
	1017H	17H, 0F5H
	1019H	0EH

b) Deberíamos cambiar `MOV B, (X+1)` por `MOV B, A` pues en el caso de que el resultado de la comparación sea  $A > B$ , el último dato leído de memoria será el mayor de los manejados hasta el momento y habrá que almacenarlo en B.

c) Teniendo en cuenta que el programa empieza en la posición 1000H y que la instrucción `JMP Inicio` ocupa 3 bytes, el primer dato habrá que emplazarlo en la posición 1003H, por lo que la subrutina pedida será:

```
Leer:  MOV  X, 1003H  ;Apunta a la primera celda
        MOV  B, 4    ;Inicializa el contador del bucle
        MOV  C, 30H  ;Especifica el puerto
        IN   A, C    ;Lectura desde el puerto
        MOV  (X), A  ;Almacena el dato leído
        INC  X       ;Se pasa a la siguiente posición
        DEC  B       ;Decrementa el contador
        JNZ  Leer    ;Se repite hasta leer el cuarto n°
        RET
```





# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### TEST

**(2.5 Puntos)**

En las siguientes preguntas de tipo test, sólo existe una respuesta correcta. Cada pregunta tiene un valor de 0,25 ptos. Cada respuesta incorrecta descuenta 0,1 ptos. La pregunta que deje en blanco no sumará ni restará puntos.

**Notas:** • Se debe poner una X sobre la letra de la **tabla "SOLUCIÓN"** (que encontrará en la hoja siguiente) correspondiente a la respuesta correcta.

• La tabla **"CALIFICACIÓN"** no deberá utilizarla, es para su posterior calificación.

• No se tendrán en cuenta las respuestas que no se encuentren en dicha tabla **"SOLUCIÓN"**

- Suponga que la frecuencia de reloj de un equipo basado en el microprocesador Indalo 3.0 es de 100 (MHz), de forma que su periodo es de 10(ns). Suponga, además, que el tiempo de acceso ( $T_A$ ) de la memoria principal disponible es de 50 (ns). Entonces, la ejecución de una operación básica  $R_{alu}$  requerirá, al menos, de:
  - 1 ciclo de WAIT.
  - 2 ciclos de WAIT.
  - 3 ciclos de WAIT.
  - 4 ciclos de WAIT.
- La memoria entrelazada y la memoria caché, son dos tipos de memoria cuya función es, fundamentalmente:
  - Aumentar la capacidad del subsistema de memoria.
  - Reducir los costes del subsistema de memoria.
  - Aumentar el rendimiento del subsistema de memoria, reduciendo el número de ciclos de WAIT en los accesos a memoria.
  - Ninguna de las anteriores es correcta.
- Se sabe que, en un determinado momento de la ejecución de un programa del Indalo 3.0, el contenido del registro A es 9EH, y que  $F_C=0$ . A continuación, el secuenciador de su unidad de control ejecuta la instrucción RCL A. Entonces, después de ejecutada la instrucción...
  - $A=3CH$ ,  $F_C=1$ ,  $F_O=1$ ,  $F_S=0$ ,  $F_Z=1$  y  $F_P=1$ .
  - $A=9EH$ ,  $F_C=1$ ,  $F_O=1$ ,  $F_S=1$ ,  $F_Z=0$  y  $F_P=1$ .
  - $A=3CH$ ,  $F_C=1$ ,  $F_O=0$ ,  $F_S=0$ ,  $F_Z=0$  y  $F_P=1$ .
  - $A=3CH$ ,  $F_C=1$ ,  $F_O=1$ ,  $F_S=0$ ,  $F_Z=0$  y  $F_P=1$ .



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### CONTINUACIÓN TEST

4. Se sabe que, en un determinado momento de la ejecución de un programa del Indalo 3.0, el contenido de los registros A y B de la unidad de proceso es, respectivamente, A=7FH y B=90H. A continuación, se ejecuta la instrucción CMP A, B. Sabiendo que el programa maneja los operandos A y B como enteros sin signo de 8 bits, ¿cómo quedarán los flags  $F_Z$  y  $F_C$  tras la ejecución de la instrucción?
- (a)  $F_Z=1$ ,  $F_C=0$ , por lo que  $A=B$ .
  - (b)  $F_Z=0$ ,  $F_C=0$ , por lo que  $A>B$ .
  - (c)  $F_Z=0$ ,  $F_C=1$ , por lo que  $A<B$ .
  - (d) Ninguna de las anteriores es correcta.
5. Suponga ahora que, tras ejecutar el secuenciador de la unidad de control del Indalo 3.0 la instrucción CMP A, B -en la que A y B representan enteros con signo-, el contenido de los flags es  $F_Z=0$ ,  $F_O=1$  y  $F_S=0$ . Esto indica que...
- (a)  $A<B$ .
  - (b)  $A=B$ .
  - (c)  $A>B$ .
  - (d) Lo que indiquen estos flags no es significativo, puesto que habría que conocer el valor de  $F_C$ .
6. Examine el siguiente fragmento de código ensamblador del Indalo 3.0:
- ```
ORG 9E4FH
MOV A, 3FH
MOV B, 20H
MOV C, 00H
BUCLE: SUB A, B
      JC  FIN
      INC C
      JMP BUCLE
FIN:   HLT
      ...
```
- ¿Cuántas veces se ejecuta la instrucción JMP BUCLE?
- (a) 1 vez.
  - (b) 2 veces.
  - (c) 3 veces.
  - (d) 4 veces.
7. Sigamos con el fragmento de código ensamblador del Indalo 3.0 anterior. ¿Qué valor asignará el programa ensamblador a la etiqueta Bucle?
- (a) 9E4FH.
  - (b) 9E51H.
  - (c) 9E53H.
  - (d) 9E55H.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

---

### CONTINUACIÓN TEST

---

8. Observe la instrucción JC FIN, y su ubicación dentro del programa anterior. Una vez ensamblada, dicha instrucción ocupará dos bytes, uno para su código de operación – 11H- y otro para el valor de *rel8*, puesto que las instrucciones de salto condicional emplean como modo de direccionamiento el *relativo al PC*. ¿Qué valor calculará el programa ensamblador para *rel8* en este caso?
- (a) 0FCH.
  - (b) 0FBH.
  - (c) 04H.
  - (d) 05H.
9. La división en operaciones básicas de la instrucción JMP BUCLE del programa anterior es:
- (a) F, F, F, I<sub>T16</sub>.
  - (b) F, F, F, I<sub>ID16</sub>.
  - (c) F, F, F<sub>NOINC</sub>, I<sub>ID16</sub>.
  - (d) F, F, F<sub>NOINC</sub>, I<sub>T16</sub>.
10. Una instrucción característica del Indalo 3.0, que no existe en el Indalo 2.0, es precisamente una instrucción de manejo de la pila o “stack”. Hablamos de la instrucción POP BC. ¿Cuántos ciclos de máquina se producen durante la lectura y posterior ejecución de dicha instrucción?
- (a) Uno.
  - (b) Dos.
  - (c) Tres.
  - (d) Cuatro.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Mañana)**

### CONTINUACIÓN TEST

#### SOLUCIÓN

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|----|
| a | a | a | a | a | a | a | a | a | a  |
| b | b | b | b | b | b | b | b | b | b  |
| c | c | c | c | c | c | c | c | c | c  |
| d | d | d | d | d | d | d | d | d | d  |

La respuesta correcta se indicará tachando la letra correspondiente con una **X**.

Para anular una respuesta dada, se rodeará con una circunferencia  $\otimes$ , de modo que se entenderá:

|           |                                  |
|-----------|----------------------------------|
| $\otimes$ | respuesta dada por el alumno.    |
| $\otimes$ | respuesta anulada por el alumno. |

#### CALIFICACIÓN

|           |  |             |   |
|-----------|--|-------------|---|
| Correctas |  | x 0,25 =    |   |
| Erróneas  |  | x (- 0,1) = | - |
| Total     |  |             |   |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

---

### **PROBLEMA. 1**

**(2.5 Puntos)**

En la tabla que aparece a continuación se representan las microinstrucciones y microórdenes correspondientes a la ejecución completa de una secuencia de instrucciones en un Indalo 3.0.

Se pide:

1. Complete las columnas de microinstrucciones y microórdenes con los valores que correspondan (*1 pto.*).
2. Identifique a qué operaciones básicas corresponden cada conjunto de microinstrucciones. Agrúpelas y use para nombrarlas la simbología empleada en la asignatura (*0,5 ptos.*).
3. Complete las casillas en blanco de las columnas ABUS y DBUS (contenido de cada BUS en el instante correspondiente) (*0,5 ptos.*).
4. Indique cuál es la secuencia de instrucciones que se ha ejecutado (Debe especificar lo más posible cada instrucción y, si corresponde, comentar en qué condiciones se produce la ejecución de la misma) (*0,5 ptos.*).

NOTA: El conjunto de los datos proporcionados corresponden a la ejecución completa de una o varias instrucciones.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 7 de junio de 2004. (Tarde)

### PROBLEMA. 1

(cont)

| Instrucción | µInstrucciones               | µÓrdenes                     | Op. Básica | ABUS  | DBUS |
|-------------|------------------------------|------------------------------|------------|-------|------|
|             | PC → ABUS                    |                              |            | 21A0H |      |
|             | (ABUS) → DBUS                |                              |            |       |      |
|             | DBUS → CO                    |                              |            |       |      |
|             | PC++                         |                              |            |       |      |
|             | 0 → F <sub>C</sub>           |                              |            |       |      |
|             | BC → ABUS                    |                              |            |       |      |
|             | (ABUS) → DBUS                |                              |            | 1234H | AAH  |
|             |                              | alu3, alu1, alu0, ckfl, ckfc |            |       |      |
|             | PC → ABUS                    |                              |            |       |      |
|             | (ABUS) → DBUS                |                              |            |       | 1BH  |
|             | DBUS → CO                    |                              |            |       |      |
|             | PC++                         |                              |            |       |      |
|             | PC → ABUS                    |                              |            |       |      |
|             | (ABUS) → DBUS                |                              |            |       | 5DH  |
|             | DBUS → DIR <sub>L</sub>      |                              |            |       |      |
|             | PC++                         |                              |            |       |      |
|             | PC → ABUS                    |                              |            |       |      |
|             | ABUS+DIR <sub>L</sub> → ABUS |                              |            |       |      |
|             | ABUS → PC                    |                              |            |       |      |
|             | PC → ABUS                    |                              |            |       |      |
|             | (ABUS) → DBUS                |                              |            |       |      |
|             | DBUS → CO                    |                              |            |       |      |
|             | PC++                         |                              |            |       |      |
|             | 0 → F <sub>i</sub>           |                              |            |       |      |
|             | PC → ABUS                    |                              |            |       |      |
|             | (ABUS) → DBUS                |                              |            |       |      |
|             | DBUS → CO                    |                              |            |       |      |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
 ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
 Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 7 de junio de 2004. (Tarde)

### PROBLEMA. 1

(cont)

| Instrucción       | µInstrucciones                       | µÓrdenes                     | Op. Básica               | ABUS         | DBUS       |
|-------------------|--------------------------------------|------------------------------|--------------------------|--------------|------------|
| <b>CMP A,(BC)</b> | PC → ABUS                            | <i>salpc</i>                 | <b>F<sub>flag</sub></b>  | 21A0H        |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          |              | <b>85H</b> |
|                   | DBUS → CO                            | <i>ckco</i>                  |                          |              |            |
|                   | PC++                                 | <i>reg1,ckpcl,ckpch</i>      |                          |              |            |
|                   | 0 → F <sub>C</sub>                   | <i>quitfc</i>                |                          |              |            |
|                   | BC → ABUS                            | <i>salbc</i>                 | <b>R<sub>alu</sub></b>   |              |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          | 1234H        | AAH        |
|                   | <b>A + Ca2(DBUS+FC) → act. flags</b> | alu3, alu1, alu0, ckfl, ckfc |                          |              |            |
| <b>JZ 2200H</b>   | PC → ABUS                            | <i>salpc</i>                 | <b>F</b>                 | <b>21A1H</b> |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          |              | 1BH        |
|                   | DBUS → CO                            | <i>ckco</i>                  |                          |              |            |
|                   | PC++                                 | <i>reg1,ckpcl,ckpch</i>      |                          |              |            |
|                   | PC → ABUS                            | <i>salpc</i>                 | <b>F</b>                 | <b>21A2H</b> |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          |              | 5DH        |
|                   | DBUS → DIR <sub>L</sub>              | <i>ckdirl</i>                |                          |              |            |
|                   | PC++                                 | <i>reg1,ckpcl,ckpch</i>      |                          |              |            |
|                   | PC → ABUS                            | <i>salpc</i>                 | <b>I<sub>t16</sub></b>   |              |            |
|                   | ABUS+DIR <sub>L</sub> → ABUS         | <i>sumdir</i>                |                          |              |            |
|                   | ABUS → PC                            | <i>ckpcl,ckpch</i>           |                          |              |            |
|                   |                                      |                              |                          |              |            |
| <b>CLI</b>        | PC → ABUS                            | <i>salpc</i>                 | <b>F<sub>flag</sub></b>  | <b>2200H</b> |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          |              | <b>0AH</b> |
|                   | DBUS → CO                            | <i>ckco</i>                  |                          |              |            |
|                   | PC++                                 | <i>reg1,ckpcl,ckpch</i>      |                          |              |            |
|                   | 0 → F <sub>I</sub>                   | <i>quitfi</i>                |                          |              |            |
| <b>HLT</b>        | PC → ABUS                            | <i>salpc</i>                 | <b>F<sub>noinc</sub></b> |              |            |
|                   | (ABUS) → DBUS                        | <i>mem,rd</i>                |                          | <b>2201H</b> | <b>0EH</b> |
|                   | DBUS → CO                            | <i>ckco</i>                  |                          |              |            |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

### PROBLEMA. 2

**(2.5 Puntos)**

Sea el siguiente fragmento de programa:

```
MOV B,A
SHL A
JC sigue1
AND A,10000000B
JMP sigue
sigue1: OR A,01111111B
AND A,01000000B
sigue:  MOV C,A
        MOV A,B
        SHL A
        SHL A
        AND A,11000000B
        XOR A,B
```

Suponemos que inicialmente el registro A contiene cierto número  $a_7a_6a_5a_4a_3a_2a_1a_0$ .

- a) Después de ejecutada la última instrucción (XOR A,B) el flag de cero se activa si se cumplen simultáneamente ciertas condiciones. Obtend dichas condiciones (1,5 pto.).
- b) Obtenga, en función del valor de los bits de A, el tiempo de ejecución del fragmento (1 pto.).

SOLUCION:

a) Inicialmente  $A = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$

**MOV B,A** ;  $B = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$

**SHL A** ;  $A = a_6 a_5 a_4 a_3 a_2 a_1 a_0 0$

**JC SIGUE1** ;  $F_c = a_7$

**1)  $a_7 = 1$**

**OR A,01111111B** ;  $A = a_6 1 1 1 1 1 1 1$

**AND A,01000000B** ;  $A = 0 1 0 0 0 0 0 0$

**MOV C,A** ;  $C = 0 1 0 0 0 0 0 0$

**2)  $a_7 = 0$**

**AND A, 10000000B** ;  $A = a_6 0 0 0 0 0 0 0$

**MOV C;A**  $C = a_6 0 0 0 0 0 0 0$

**MOV A,B** ;  $A = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$

**SHL A** ;  $A = a_6 a_5 a_4 a_3 a_2 a_1 a_0 0$

**SHL A** ;  $A = a_5 a_4 a_3 a_2 a_1 a_0 0 0$

**AND A,11000000B** ;  $A = a_5 a_4 0 0 0 0 0 0$

Finalmente se efectúa la xor de  $A = a_5 a_4 0 0 0 0 0 0$  con  $B = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$ . El resultado será cero si  $a_7 = a_5$ ,  $a_6 = a_4$ ,  $a_5 = a_4 = a_3 = a_2 = a_1 = a_0 = 0$





UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

**PROBLEMA. 2**

**(cont.)**

**b)** El tiempo que transcurre en uno y otro caso será :

$a_7 = 1$

MOV B,A .....(7 Pulsos)  
SHL A.....(7 Pulsos)  
JC sigue1.....(12 Pulsos)  
OR A,01111111B....(10 Pulsos)  
AND A,01000000B....(10 Pulsos)  
MOV C,A.....(7 Pulsos)  
MOV A,B.....(7 Pulsos)  
SHL A.....(7 Pulsos)  
SHL A.....(7 Pulsos)  
AND A,11000000B..(10 Pulsos)  
XOR A,B.....(7 Pulsos)  
TOTAL..... 91 Pulsos

$a_7 = 0$

MOV B,A .....(7 Pulsos)  
SHL A.....(7 Pulsos)  
JC sigue1.....(7 Pulsos)  
AND A,10000000B...(10 Pulsos)  
JMP sigue.....(15 Pulsos)  
MOV C,A.....(7 Pulsos)  
MOV A,B.....(7 Pulsos)  
SHL A.....(7 Pulsos)  
SHL A.....(7 Pulsos)  
AND A,11000000B..(10 Pulsos)  
XOR A,B.....(7 Pulsos)  
TOTAL..... 91 Pulsos



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

### PROBLEMA. 3

**(2.5 Puntos)**

En un determinado sistema gobernado por un INDALO 3.0 se pretende diseñar la interfaz hardware y software de una impresora Centronics con las siguientes características:

- La señal de petición de interrupción hardware está conectada de forma que se activa tanto con la señal ACK (es decir, porque la impresora ya haya recogido el carácter anterior y esté pidiendo el siguiente), como por la señal PE (falta papel) o porque SELECT esté desactivada (impresora *off line* —fuera de línea—).
- La rutina de interrupción, que se muestra más abajo, deberá leer el puerto de estado de la impresora, que tiene la dirección 10H, y comprobar si su bit 5 es 0 (significaría que hay papel) y su bit 7 es 1 (indicaría que está *on line*), y en ese caso llamar a la subrutina **enviar**, que se encargaría de enviar el nuevo carácter, mientras que si no se cumple la condición de bits, hay que llamar a la subrutina **avisar**, que visualizaría por pantalla un mensaje de error.

```
int_imp:  MOV     C,10H
          IN      A,C
          .....
          .....
          .....
          .....
          .....
          CALL    avisar
          RET
sigue:    CALL    enviar
          RET
```

*;Fragmento que debe comprobar  
;si el bit 5 de A es un 0 y  
;el bit 7 un 1, y en ese caso  
;tiene que saltar a la  
;etiqueta sigue.*

Se pide:

- A) Codificar en ensamblador el fragmento que falta en la rutina de atención a la interrupción. (No incluir las subrutinas **avisar** ni **enviar**; sólo el fragmento que analiza los bits). (0,7 ptos.)

- B) Si la interrupción de impresora es la de vector 02 y se conocen además los siguientes contenidos en memoria, ¿qué dirección le corresponderá a la etiqueta **int\_imp**? (0,5 ptos.)

| Dirección | Contenido |
|-----------|-----------|
| 0000      | 1AH       |
| 0001      | 2BH       |
| 0002      | 3CH       |
| 0003      | 4DH       |
| 0004      | 5EH       |
| 0005      | 6FH       |
| 0006      | 70H       |
| 0007      | 81H       |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

### **PROBLEMA. 3**

**(cont.)**

- C) Suponiendo que se produce una interrupción de impresora por falta de papel mientras que se está ejecutando la instrucción **ADD A,B** que está situada en la dirección **9F7CH** (esta instrucción pertenece al programa interrumpido, que no se muestra en este enunciado), y que en ese momento **SP=F708H**, indicar el contenido del **SP** y los valores guardados en la tabla cuando se entra en la subrutina **avisar**, especificando también en que direcciones se guardan. (0,7 ptos.)  
NOTA: Si no se hubiera contestado a los apartados A o B, supóngase que la etiqueta **int\_imp** se corresponde con la dirección **1000H** y que el fragmento pedido en el apartado A ocupa **10H bytes**, pero **indíquese que se ha partido de esta suposición**.
- D) Ensamblar (obtener su código máquina) la instrucción de salto condicional que se haya incluido en el fragmento del apartado A. Si no se hubiera contestado a ese apartado, suponer que la última instrucción del fragmento es un salto condicional (es indiferente qué condición se use, pero hay que indicarlo). (0,6 ptos.)

### **SOLUCIÓN**

#### **Apartado A**

**AND A,0A0H**  
**XOR A,80H**  
**JZ sigue**

#### **Apartado B**

**6F5EH**

#### **Apartado C**

**SP=F704H porque se han guardado dos direcciones de retorno que ocupan 2 words.**

| <b>DIRECCIÓN</b> | <b>CONTENIDO</b> | <b>COMENTARIOS</b>                                                       |
|------------------|------------------|--------------------------------------------------------------------------|
| <b>F707H</b>     | <b>9FH</b>       | <b>Dirección de la instrucción siguiente a ADD A,B</b>                   |
| <b>F706H</b>     | <b>7DH</b>       |                                                                          |
| <b>F705H</b>     | <b>6FH</b>       | <b>Dirección de la instrucción siguiente a CALL avisar<sup>1 2</sup></b> |
| <b>F704H</b>     | <b>6AH</b>       |                                                                          |

<sup>1</sup> El cálculo figura en la página siguiente.

<sup>2</sup> Si se hubiera basado en la suposición indicada en el enunciado, el resultado hubiera sido: (F705H)=10H y (F706H)=16H



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 7 de junio de 2004. (Tarde)

**PROBLEMA. 3**

**(cont.)**

|                                  | <b>BYTES<br/>OCUPADOS</b> | <b>DIRECCIÓN<br/>INICIAL</b> |
|----------------------------------|---------------------------|------------------------------|
| <i>int_imp:</i> <b>MOV C,10H</b> | <b>2</b>                  | <b>6F5EH</b>                 |
| <b>IN A,C</b>                    | <b>1</b>                  | <b>6F60H</b>                 |
| <b>AND A,0A0H</b>                | <b>2</b>                  | <b>6F61H</b>                 |
| <b>XOR A,80H</b>                 | <b>2</b>                  | <b>6F63H</b>                 |
| <b>JZ sigue</b>                  | <b>2</b>                  | <b>6F65H</b>                 |
| <b>CALL avisar</b>               | <b>3</b>                  | <b>6F67H</b>                 |
| <b>RET</b>                       | <b>1</b>                  | <b>6F6AH</b>                 |
| <i>sigue:</i> <b>CALL enviar</b> |                           | <b>6F6BH</b>                 |
| <b>RET</b>                       |                           |                              |

**Apartado D**

**El código de operación es 1BH y  $rel8 = 6F6BH - 6F67H = 04$ . Por lo tanto la instrucción desensamblada será:**  
**1BH,04<sup>3</sup>**

<sup>3</sup> Si se hubiera hecho la suposición del enunciado, el primer byte pudiera ser distinto (11H o de 13H a 1BH) dependiendo de la condición elegida, pero el segundo seguiría siendo 04.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_

ASIGNATURA: **Fundamentos de Computadores (Test)**

CURSO: **1º**

GRUPO: \_\_\_\_\_

Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

### **TEST**

**(2.5 Puntos)**

En las siguientes preguntas de tipo test, sólo existe una respuesta correcta. Cada pregunta tiene un valor de 0,25 ptos. Cada respuesta incorrecta descuenta 0,1 ptos. La pregunta que deje en blanco no sumará ni restará puntos.

**Notas:** • Se debe poner una X sobre la letra de la tabla "SOLUCIÓN" (que encontrará en la hoja siguiente) correspondiente a la respuesta correcta.

• La tabla "CALIFICACIÓN" no deberá utilizarla, es para su posterior calificación.

• No se tendrán en cuenta las respuestas que no se encuentren en dicha tabla "SOLUCIÓN"

1) Cuántos ciclos de máquina se necesitan para leer y ejecutar la instrucción CALL 4500H

- a) Tres (los tres de tipo F)
- b) Cuatro (tres de tipo F y un W)
- c) Cinco (tres de tipo F y dos W)
- d) Seis (tres de tipo F y otros tres W)

2) En el Indalo 3.0, indicar cuál de estas sentencias es correcta

- a) En una división por dos (desplazamiento a la derecha), el flag de carry es el resto.
- b) Hay vectores de interrupción de dos bytes.
- c) No hay instrucciones que ocupen tres bytes
- d) El flag de overflow indica desbordamiento en enteros sin signo..

3) ¿Cuántos ciclos de máquina se precisan para leer la instrucción cuyo código fuente es JNO dirección, sabiendo que el FO = 0?

- a) Un ciclo de F y otro de Fnoinc
- b) Tres ciclos de F
- c) Dos ciclos de F
- d) Dos ciclos de F y uno de Fnoinc

4) El contador de programa del Indalo 3.0 es un registro que:

- a) Se incrementa siempre al final de cada ciclo de Falu
- b) Puede cambiarse discrecionalmente su contenido con una instrucción MOV
- c) No puede incrementarse.
- d) Se incrementa siempre al final de cada ciclo de Fnoinc.

5) Sabiendo que SP = 8000H, después de efectuar estas dos instrucciones PUSH X y PUSH BC, el SP valdrá...

- a) El SP = 7FFEh
- b) EL SP = 8004H
- c) El SP = 7FFCh
- d) EL SP = 7996H



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_

ASIGNATURA: **Fundamentos de Computadores (Test)**

CURSO: **1º**

GRUPO: \_\_\_\_\_

Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

---

### **CONTINUACIÓN TEST**

---

6) Se sabe que los registros SP= 8000H , BC= 0345H, F = 0FH, que en la dirección 2350H se encuentra la instrucción SUB A, A que acabamos de leer y estamos ejecutando, y en ese momento ocurre una interrupción HW cuyo vector de interrupción sabemos que vale 50H. ¿Qué ocurre a continuación:

- a) No se atiende la interrupción y se continúa con la instrucción contenida en 2352H
- b) Se atiende la interrupción y se salta a la rutina de servicio contenida en las direcciones 100H y 101H
- c) No se atiende la interrupción y se continúa con la instrucción contenida en 2351H
- d) No se atiende la interrupción y se continúa con la instrucción contenida en 8000H

7) ¿Cuántas operaciones básicas se precisan para leer y ejecutar la instrucción INT 45H?

- a) Dos de tipo F, dos de tipo W y tres de tipo Iid16
- b) Dos de tipo F dos de tipo R , una de I2\*DIRL y dos de tipo Iid16
- c) Dos de tipo F, dos de tipo W, dos de tipo R y dos de tipo Iid16
- d) Dos de tipo F, dos de tipo W, dos de tipo R, una de I2\*DIRL y tres Iid16

8) Cuando se produce una interrupción Hardware,

- a) El  $\mu$ P pone en el ABUS la dirección del vector.
- b) El vector debe multiplicarse por 2 si el FI está activado
- c) La tabla de direcciones de las rutinas de atención a las interrupciones ocupa las posiciones de la 00H a la FFH
- d) Se atiende si el FI vale 0

9) La tabla de direcciones de las rutinas de servicio a las interrupciones (también llamada tabla de vectores de interrupción) tiene un tamaño máximo de:

- a) 256 bytes
- b) 512bytes
- c) 756 bytes
- d) 1024 bytes

10) En el direccionamiento de un puerto

- a) Se usa el valor del registro B para identificar el puerto
- b) Se utilizan los 8 bits de menos peso del ABUS
- c) Se utilizan los 8 bits del DBUS para identificar el puerto
- d) Se utilizan los 16 bits del registro X



**UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID**  
**Departamento de Electrónica y Comunicaciones**

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

---

**CONTINUACIÓN TEST**

---

**SOLUCIÓN**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|----|
| a | a | a | a | a | a | a | a | a | a  |
| b | b | b | b | b | b | b | b | b | b  |
| c | c | c | c | c | c | c | c | c | c  |
| d | d | d | d | d | d | d | d | d | d  |

La respuesta correcta se indicará tachando la letra correspondiente con una **X**.

Para anular una respuesta dada, se rodeará con una circunferencia  $\otimes$ , de modo que se entenderá:

|           |                                  |
|-----------|----------------------------------|
| $\otimes$ | respuesta dada por el alumno.    |
| $\otimes$ | respuesta anulada por el alumno. |

**CALIFICACIÓN**

|                  |  |                    |   |
|------------------|--|--------------------|---|
| <b>Correctas</b> |  | <b>x 0,25 =</b>    |   |
| <b>Erróneas</b>  |  | <b>x (- 0,1) =</b> | - |
|                  |  | <b>Total</b>       |   |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
 ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
 Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### PROBLEMA. 1

**(2.5 Puntos)**

a) Para el fragmento de programa adjunto, indicar **razonadamente** qué condiciones deben cumplir los bits del operando A con respecto a los de B para que se ejecute el salto condicional de la última línea.

```
AND  A, 4CH
XOR  A, 24H
SHL  A
MOV  C, A
MOV  A, B
AND  A, C8H
XOR  A, C
XOR  A, 80H
JZ   Etiqueta
```

**NOTA:** Considérense A y B como números de 8 bits,  $A = (a_7, \dots, a_0)$  y  $B = (b_7, \dots, b_0)$ , inicialmente almacenados en los registros de igual nombre.

*(1,25 puntos)*

### SOLUCIÓN

a)

|            | OPERANDO A                           |            | OPERANDO B                               |
|------------|--------------------------------------|------------|------------------------------------------|
| AND A, 4CH | $0, a_6, 0, 0, a_3, a_2, 0, 0$       | MOV A, B   | $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$ |
| XOR A, 24H | $0, a_6, 1, 0, a_3, \bar{a}_2, 0, 0$ | AND A, C8H | $b_7, b_6, 0, 0, b_3, 0, 0, 0$           |
| SHL A      | $a_6, 1, 0, a_3, \bar{a}_2, 0, 0, 0$ |            |                                          |
| MOV C, A   | $a_6, 1, 0, a_3, \bar{a}_2, 0, 0, 0$ |            |                                          |

|            | RESULTADO                                                              |
|------------|------------------------------------------------------------------------|
| XOR A, C   | $(a_6 \oplus b_7), \bar{b}_6, 0, a_3, (\bar{a}_2 \oplus b_3), 0, 0, 0$ |
| XOR A, 80H | $(a_6 \oplus b_7), \bar{b}_6, 0, a_3, (\bar{a}_2 \oplus b_3), 0, 0, 0$ |

Para que se produzca el salto, debe activarse el flag  $F_Z$ , por lo que las condiciones exigidas son:

$$\begin{aligned}
 (\overline{a_6 \oplus b_7}) = 0 &\Rightarrow (a_6 \oplus b_7) = 1 \Rightarrow a_6 \neq b_7 \\
 \bar{b}_6 = 0 &\Rightarrow b_6 = 1 \\
 a_3 &= 0 \\
 (\bar{a}_2 \oplus b_3) = 0 &\Rightarrow \bar{a}_2 = b_3 \Rightarrow a_2 \neq b_3
 \end{aligned}$$





# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### PROBLEMA. 1

(cont.)

b) Sean el programa y volcado de memoria siguientes:

|        |     |       |       |      |
|--------|-----|-------|-------|------|
|        | ORG | 1000H | 2000H | 0A3H |
|        | XOR | A, A  | 2001H | 4EH  |
|        | POP | BC    | 2002H | 09H  |
| Bucle: | INC | B     | 2003H | 10H  |
|        | MOV | A, B  | 2004H | 0EH  |
|        | CMP | A, C  | 2005H | 0EFH |
|        | JNZ | Bucle | 2006H | 4CH  |
|        | HLT |       | 2007H | 57H  |

Considerando que inicialmente SP = 2003H, rellenar la tabla adjunta indicando la evolución de los flags y los registros A, B y C (éstos últimos en notación hexadecimal) para cada una de las instrucciones del programa hasta llegar a HLT (inclúyanse todas y cada una de las ejecuciones del bucle).

(1,25 puntos)

| F <sub>C</sub> | F <sub>O</sub> | F <sub>S</sub> | F <sub>P</sub> | F <sub>Z</sub> | A | B   | C   | INSTRUCCIÓN |
|----------------|----------------|----------------|----------------|----------------|---|-----|-----|-------------|
|                |                |                |                |                |   | 00H | 00H | XOR A, A    |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |
|                |                |                |                |                |   |     |     |             |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### PROBLEMA. 1

(cont.)

### SOLUCIÓN

b) Puesto que  $SP = 2003H$ , lo que se guarda en BC al leer de la pila con la instrucción POP BC es  $0E10H$ , ya que  $C=(2003H)=10H$  y  $B=(2004H)=0EH$ .

El programa va incrementa B para a continuación compararlo con C, repitiendo el proceso hasta que, como resultado de esa comparación, se active el flag  $F_Z$ , es decir,  $B=C$ .

| $F_C$ | $F_O$ | $F_S$ | $F_P$ | $F_Z$ | A   | B   | C   | INSTRUCCIÓN |
|-------|-------|-------|-------|-------|-----|-----|-----|-------------|
| 0     | 0     | 0     | 1     | 1     | 00H | 00H | 00H | XOR A, A    |
| 0     | 0     | 0     | 1     | 1     | 00H | 0EH | 10H | POP BC      |
| 0     | 0     | 0     | 1     | 0     | 00H | 0FH | 10H | INC B       |
| 0     | 0     | 0     | 1     | 0     | 0FH | 0FH | 10H | MOV A, B    |
| 1     | 0     | 1     | 1     | 0     | 0FH | 0FH | 10H | CMP A, C    |
| 1     | 0     | 1     | 1     | 0     | 0FH | 0FH | 10H | JNZ Bucle   |
| 1     | 0     | 0     | 0     | 0     | 0FH | 10H | 10H | INC B       |
| 1     | 0     | 0     | 0     | 0     | 10H | 10H | 10H | MOV A, B    |
| 0     | 0     | 0     | 1     | 1     | 10H | 10H | 10H | CMP A, C    |
| 0     | 0     | 0     | 1     | 1     | 10H | 10H | 10H | JNZ Final   |
| 0     | 0     | 0     | 1     | 1     | 10H | 10H | 10H | HLT         |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

---

### **PROBLEMA. 2**

**(2.5 Puntos)**

El siguiente fragmento de código corresponde a un programa en el ensamblador del microprocesador INDALO 3.0.

```
                ORG 7FFFH
                JMP comienzo
dato:           DB ?
comienzo:       MOV A, (dato)
                AND A, 0FH
                MOV B, A
                MOV A, (dato)
                AND A, 0F0H
                CMP A, B
                JNZ fin
                JMP seguir
fin:            HLT
seguir:         ...
```

Se pide:

- (a) Código máquina del programa, incluyendo la equivalencia en hexadecimal de las etiquetas del mismo y el número total de bytes que ocupa. (1.25 puntos)
- (b) Suponiendo que el contenido de la dirección de memoria DATO es 12H, esto es, (DATO)=12H, y que la frecuencia de reloj del sistema es de 166 MHz. Obtenga, para cada instrucción, las operaciones básicas desarrolladas y el tiempo total (especifíquelo en ciclos y en unidades de segundo) de ejecución del fragmento. (1.75 puntos)



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### PROBLEMA. 2

(cont.)

### SOLUCIÓN

(a) Construimos una tabla con las direcciones de memoria en su primera columna y el código máquina en la segunda, como se muestra a continuación:

| Dirección (Etiqueta) | Código Máquina |
|----------------------|----------------|
| 7FFFH                | 12H, 03H, 80H  |
| 8002H=DATO           | ?              |
| 8003H=COMIENZO       | 68H, 02H, 80H  |
| 8006H                | 9CH, 0FH       |
| 8008H                | 71H            |
| 8009H                | 68H, 02H, 80H  |
| 800CH                | 9CH, 0F0H      |
| 800EH                | 82H            |
| 800FH                | 17H, 03H*      |
| 8011H                | 12H, 15H, 80H  |
| 8014H=FIN            | 0EH            |
| 8015H=SEGUIR         | ...            |
| Nº Total Bytes       | 22             |

\* Es el valor de rel8 en la instrucción JNZ FIN, y se calculará como la diferencia entre la dirección a la que saltará la instrucción si se cumple la condición, esto es, FIN=8014H, y el valor del PC después de leer la instrucción, esto es, PC=8011H.

Entonces,

$$\begin{array}{rcl} & \text{FIN=} & 8014\text{H} \\ - & \text{PC=} & -8011\text{H} \\ \hline \text{(Después de haber leído JNZ FIN)} & \text{rel8=} & 0003\text{H} \end{array}$$



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### PROBLEMA. 2

(cont.)

### SOLUCIÓN

(b) Podemos hacer una tabla similar a la anterior en la que figure la instrucción en la primera columna, sus operaciones básicas en la segunda y el número de ciclos de cada operación en la tercera.

| Instrucción  | Operaciones Básicas                   | Ciclos de CK | Comentario                                    |
|--------------|---------------------------------------|--------------|-----------------------------------------------|
| JMP comienzo | F, F, F <sub>NOINC</sub> , IT16       | 15           | -                                             |
| MOV A,(dato) | F, F, F, R <sub>ALU</sub>             | 18           | A=12H                                         |
| AND A,0FH    | F, F <sub>ALU</sub>                   | 10           | A=02H                                         |
| MOV B, A     | F, I <sub>TALU</sub>                  | 7            | B=02H                                         |
| MOV A,(dato) | F, F, F, R <sub>ALU</sub>             | 18           | A=12H                                         |
| AND A, 0F0H  | F, F <sub>ALU</sub>                   | 10           | A=10H                                         |
| CMP A, B     | F <sub>FLAG</sub> , I <sub>TALU</sub> | 7            | F <sub>Z</sub> =0                             |
| JNZ fin      | F, F, I <sub>T16</sub>                | 12           | Se efectúa el salto, porque F <sub>Z</sub> =0 |
| JMP seguir   | -                                     | -            | No se ejecuta                                 |
| HLT          | F <sub>NOINC</sub>                    | 3            | -                                             |

Nº Total Ciclos CK= 100

Entonces:

$$f_{CK} = 133(\text{MHz}) \cdot 10^6(\text{Hz}) / 1(\text{MHz}) \cdot 1(\text{ciclo/s}) / 1(\text{Hz}) = 133 \cdot 10^6(\text{ciclos/s})$$

De ahí que:

$$1(\text{ciclo}) = 1 / [133 \cdot 10^6(\text{s}^{-1})] \approx 7.5 \cdot 10^{-9}(\text{s})$$

Finalmente, pues:

$$\boxed{\text{Tiempo de ejecución} = 100(\text{ciclos}) \cdot 7.5 \cdot 10^{-9}(\text{s/ciclo}) \approx 750 \cdot 10^{-9}(\text{s}) = 750(\text{ns}) \approx 0.8(\mu\text{s})}$$



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### **PROBLEMA. 3**

**(2.5 Puntos)**

Las instrucciones de un programa para el Indalo 3 comienzan a partir de la dirección 2000H. El código máquina es el que aparece en el siguiente volcado de memoria:

| Dirección    | Contenido           |
|--------------|---------------------|
| <b>2000H</b> | 56H 90H 34H 6EH 00H |
| <b>2005H</b> | 84H FFH 1BH 08H 9CH |
| <b>200AH</b> | 03H 49H 00H 35H 12H |
| <b>200FH</b> | 03H 20H 0EH         |

Los datos se encuentran en las direcciones 3490H hasta la 3499H (ambas inclusive):

| Dirección | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 | 3497 | 3498 | 3499 |
|-----------|------|------|------|------|------|------|------|------|------|------|
|           | H    | H    | H    | H    | H    | H    | H    | H    | H    | H    |
| Contenido | 4FH  | 49H  | 6AH  | 75H  | 86H  | 0ABH | 00H  | D8H  | 0FFH | 46H  |

Se pide:

- Desensamblar el programa (0,50 pts.)
- ¿Cuál es el contenido de las direcciones 3490H hasta la 3499H, al finalizar el programa? ¿Cuáles son los valores finales de los registros A y X ? (0,75 pts.)
- ¿Qué hace el programa?. Explicar la funcionalidad del programa en menos de 7 líneas. (0,75 pts.)
- Calcular el tiempo que tarda en ejecutarse el programa sabiendo que la frecuencia de la CPU es de 1 GHz. Expresar la respuesta en nanosegundos (0,50 pts.)

*Nota. Contestar cada apartado en el espacio reservado para tal fin. No se corregirá cualquier respuesta que no esté situada en el sitio indicado*



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### **PROBLEMA. 3**

**(cont.)**

### **SOLUCIÓN**

**Código fuente:**

| Dirección | Instrucción (código fuente) |
|-----------|-----------------------------|
| 2000 H    | MOV X, 3490H                |
| 2003H     | MOV A, (X)                  |
| 2005H     | CMP A, 0FFH                 |
| 2007H     | JZ 2011H                    |
| 2009H     | AND A, 03H                  |
| 200BH     | MOV (X), A                  |
| 200DH     | INC X                       |
| 200EH     | JMP 2003H                   |
| 2011H     | HLT                         |

- a) ¿Cuál es el contenido de las direcciones 3490H hasta la 3499H, al finalizar el programa? Valor de los registros A y X al terminar el programa.

| Dirección | 3490 | 3491 | 3492 | 3493 | 3494 | 3495 | 3496 | 3497 | 3498 | 3499 |
|-----------|------|------|------|------|------|------|------|------|------|------|
|           | H    | H    | H    | H    | H    | H    | H    | H    | H    | H    |
| Contenido | 03H  | 01H  | 02H  | 01H  | 02H  | 03H  | 00H  | 00H  | 0FFH | 46H  |

**A = 0FFH**  
**X = 3498H**

- b) ¿Qué hace el programa?. Explicar la funcionalidad del programa en menos de 7 líneas

El programa recorre mediante el registro X, que utiliza como puntero, una lista de números a los que procesa uno a uno, hasta encontrar el 0FFH, en cuyo caso el programa concluye. El proceso consiste en hacer un AND del número con la máscara 03. Es decir, deja pasar los dos últimos bits y anula los demás. (Ello equivale al resto de la división del número entre 4. Esto se denomina clase de equivalencia, módulo 4) El número obtenido, se sobrescribe sobre el dato original.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### **PROBLEMA. 3**

**(cont.)**

- c) Calcular el tiempo que tarda en ejecutarse el programa sabiendo que la frecuencia de la CPU es de 1 GHz.

| Instrucción (código fuente) | Ciclos de Reloj | Numero de veces que se ejecuta | Totales parciales |
|-----------------------------|-----------------|--------------------------------|-------------------|
| MOV X, 3490H                | 15              | 1                              | 15                |
| MOV A, (X)                  | 13              | 9                              | 117               |
| CMP A, 0FFH                 | 10              | 9                              | 90                |
| JZ 2011H                    | 7/12            | 8/1                            | 68                |
| AND A, 03H                  | 10              | 8                              | 80                |
| MOV (X), A                  | 13              | 8                              | 104               |
| INC X                       | 7               | 8                              | 56                |
| JMP 2003H                   | 15              | 8                              | 120               |
| HLT                         | 3               | 1                              | 3                 |
|                             |                 | TOTAL FINAL:                   | <b>653</b>        |

$$\text{tiempo} = 653 \cdot \text{Periodo} = 653 \cdot \frac{1}{1 \cdot 10^{-9}} \text{ seg} = 653 \text{ nseg. (nanosegundos)}$$

**Tiempo de ejecución = 653 nseg.**





# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### TEST

**(2.5 Puntos)**

En las siguientes preguntas de tipo test, sólo existe una respuesta correcta. Cada pregunta tiene un valor de 0,25 pts. Cada respuesta incorrecta descuenta 0,1 pts. La pregunta que deje en blanco no sumará ni restará puntos.

**Notas:** • Se debe poner una X sobre la letra de la **tabla "SOLUCIÓN"** (que encontrará en la hoja siguiente) correspondiente a la respuesta correcta.

• La tabla **"CALIFICACIÓN"** no deberá utilizarla, es para su posterior calificación.

• No se tendrán en cuenta las respuestas que no se encuentren en dicha tabla **"SOLUCIÓN"**

1).- Si partiendo de una situación inicial en la que los contenidos de los registros A y B son respectivamente 80H y 88H, se ejecuta la instrucción **ADD A,B**:

1a).- El contenido final del registro A será 108H.

1b).- El contenido final del registro A será forzosamente el mismo que si se hubiera ejecutado la instrucción **ADC A,B**.

1c).- El contenido final del registro A será forzosamente inferior al que se hubiera obtenido si se hubiera ejecutado la instrucción **ADC A,B**.

1d).- Ninguna de las respuestas anteriores es cierta.

2).- En los ciclos de escritura, se entiende por tiempo de acceso de una memoria al que transcurre:

2a).- Desde el momento en el que se selecciona con la señal **CS**, hasta que se realiza la transferencia del dato.

2b).- Desde el momento en el que está preparado todo (la dirección y el dato en los buses correspondientes, y además, las señales **CS** y **WE** activas), hasta que la memoria recoge el dato del bus.

2c).- Desde el momento en el que la dirección está en el bus, hasta que la memoria está preparada para capturar el dato.

2d).- Ninguna de las respuestas anteriores es correcta.

3).- Con relación a la instrucción **INT vector** del INDALO 3.0, puede decirse que...

3a).- ...se produce tanto en las interrupciones hardware como en las software.

3b).- ...sólo se ejecuta si el flag I es igual a 1.

3c).- ...al igual que en el caso de las interrupciones hardware, puede interrumpir al programa que se esté ejecutando en cualquier momento, y no es predecible en que instante se producirá.

3d).- ...se diferencia de **CALL address** únicamente en el modo de direccionamiento.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

---

### CONTINUACIÓN TEST

---

4).- La instrucción **ADD A,(X)**...

- 4a).- ... no existe como tal. Se simula por medio de 2 instrucciones: **CLC** y **ADC A,(X)**.
- 4b).- ...para no sumar nada a X no activa durante su ejecución la microorden *sumdir*.
- 4c).- ...produce durante su ejecución la microorden *quitfc*.
- 4d).- ...se lee mediante un único ciclo de *fetch*.

5).- El *ADM* o *DMA*...

- 5a).- ...puede enmascarse mediante un biestable interno del micro.
- 5b).- ...se produce cuando se ejecutan instrucciones que emplean direccionamiento directo para acceder a memoria.
- 5c).- ...fuerza al micro a guardar la dirección de retorno en la pila.
- 5d).- Ninguna de las anteriores es cierta.

6).- Con relación a las instrucciones de salto condicional del INDALO 3.0, puede decirse que...

- 6a).- ...si el valor del segundo byte es 0FEH, se produce un bucle infinito.
- 6b).- ...dan lugar siempre a 2 ciclos de *fetch*.
- 6c).- ...ocupan un solo byte si no se cumple la condición, y 2 si se cumple.
- 6d).- Ninguna de las anteriores es cierta.

7).- Si tras una instrucción **CMP A,B** se obtienen los valores de los flags:  $F_C=1$ ,  $F_S=0$ ,  $F_O=1$  y  $F_Z=0$ , entonces:

- 7a).-  $A < B$  en enteros con signo.
- 7b).-  $A > B$  en enteros con signo.
- 7c).-  $A > B$  en enteros sin signo.
- 7d).- Ninguna de las anteriores es cierta.

8).- En la CPU INDALO 3.0 ...

- 8a).- ... la pila crece hacia posiciones de memoria decrecientes.
- 8b).- ... la pila crece si crece SP.
- 8c).- ... la pila se direcciona mediante direccionamiento absoluto.
- 8d).- ... el programa funciona más lento, cuanto menor sea la pila.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Test)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Mañana)**

### CONTINUACIÓN TEST

9).- Cuando se produce una interrupción Hardware

- 9a).- El  $\mu P$  pone en el ABUS la dirección del vector.
- 9b).- En el Indalo 3.0 el vector debe sufrir un desplazamiento o *shift* de 1 bit a la izquierda.
- 9c).- La tabla de direcciones de las rutinas de atención a las interrupciones ocupa las posiciones de la 00H a la FFH.
- 9d).- No hay diferencia entre una interrupción y una CALL *addr*.

10).- Si en un sistema basado en el INDALO 3.0 necesitamos conectar 25 puertos de entrada, 30 de salida y 16 de entrada/salida...

- 10a).- ...tiene que realizarse forzosamente una decodificación incompleta de las direcciones de los puertos.
- 10b).- ...la decodificación de las direcciones de los puerto puede ser incompleta de 6 bits.
- 10c).- ...la decodificación de las direcciones de los puerto debe ser forzosamente distribuida.
- 10d).- ...la decodificación de las direcciones de los puerto no puede ser incompleta.

### SOLUCIÓN

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|----|
| a | a | a | a | a | a | a | a | a | a  |
| b | b | b | b | b | b | b | b | b | b  |
| c | c | c | c | c | c | c | c | c | c  |
| d | d | d | d | d | d | d | d | d | d  |

La respuesta correcta se indicará tachando la letra correspondiente con una **X**.

Para anular una respuesta dada, se rodeará con una circunferencia  $\otimes$ , de modo que se entenderá:

|           |                                  |
|-----------|----------------------------------|
| $\otimes$ | respuesta dada por el alumno.    |
| $\otimes$ | respuesta anulada por el alumno. |

### CALIFICACIÓN

|           |  |             |   |
|-----------|--|-------------|---|
| Correctas |  | x 0,25 =    |   |
| Erróneas  |  | x (- 0,1) = | - |
| Total     |  |             |   |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Tarde)**

### **PROBLEMA. 1**

**(2.5 Puntos)**

Dado el siguiente fragmento de programa...:

```
MOV B, A
ROR A
AND A, 50H
XOR A,B
AND A, 5EH
XOR A,12H
JZ ETIQUETA
```

Se pide:

- Especificar todas las condiciones que deben cumplir los bits del registro A para que se produzca el salto a la dirección ETIQUETA. (1 pto.)
- En el caso de que el valor inicial de A sea 0D2H, ¿se produciría el salto a la dirección ETIQUETA?. Especificar las condiciones que se cumplen o no. (0,3 ptos.).
- Contestar al apartado b) suponiendo que el valor inicial de A es 23H (0,3 ptos.).
- Contestar al apartado b) suponiendo que el valor inicial de A es 32H (0,3 ptos.).
- ¿Cuántos bytes ocupará este fragmento en código máquina? (no se pide ensamblarlo, sólo el numero de bytes) (0,6 ptos.).

*Nota.- Denominar los bits del registro A como  $a_7a_6a_5a_4a_3a_2a_1a_0$*

*Para contestar el apartado a) se sugiere el desarrollo del programa mediante lenguaje simbólico, analizando los valores binarios de A y B*

*Contestar cada apartado en el espacio reservado para ello.*

### **SOLUCIÓN**

**a)**

| Leng. simbólico              | Leng. Ensambl. | Valor de A                                           | Valor de B                 |
|------------------------------|----------------|------------------------------------------------------|----------------------------|
| $A \rightarrow B$            | MOV B, A       | $a_7a_6a_5a_4a_3a_2a_1a_0$                           | $a_7a_6a_5a_4a_3a_2a_1a_0$ |
| $A > r \rightarrow A$        | ROR A          | $a_0a_7a_6a_5a_4a_3a_2a_1$                           | No varía                   |
| $A \cap 50H \rightarrow A$   | AND A, 50H     | $0a_70a_50000$                                       | No varía                   |
| $A \oplus B \rightarrow A$   | XOR A, B       | $a_7(a_7 \oplus a_6)a_5(a_5 \oplus a_4)a_3a_2a_1a_0$ | No varía                   |
| $A \cap 5EH \rightarrow A$   | AND A, 5EH     | $0(a_7 \oplus a_6)0(a_5 \oplus a_4)a_3a_2a_10$       | No varía                   |
| $A \oplus 12H \rightarrow A$ | XOR A, 12H     | $0(a_7 \oplus a_6)0(a_5 \oplus a_4)a_3a_2a_10$       | No varía                   |
| Salto condicional            | JZ ETIQUETA    | No varía                                             | No varía                   |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_

ASIGNATURA: **Fundamentos de Computadores (Prob.)**

CURSO: **1º**

GRUPO: \_\_\_\_\_

Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Tarde)**

### PROBLEMA. 1

(cont)

A la vista de lo anterior, las condiciones que deben cumplirse para que el Fz se active, y se produzca el salto a la dirección ETIQUETA, son :

$$a_7 = a_6, \quad a_5 \neq a_4, \quad a_3 = a_2 = 0 \quad \text{y} \quad a_1 = 1$$

b) Si A = 0D2H, su valor binario es  $a_7a_6a_5a_4a_3a_2a_1a_0$  :

| a7 | a6 | a5 | a4 | a3 | a2 | a1 | a0 |
|----|----|----|----|----|----|----|----|
| 1  | 1  | 0  | 1  | 0  | 0  | 1  | 0  |

Se cumplen todas las condiciones señaladas en el apartado anterior y por tanto, **se produce el salto** a la dirección ETIQUETA

c) Si A = 23H H, su valor binario es  $a_7a_6a_5a_4a_3a_2a_1a_0$  :

| a7 | a6 | a5 | a4 | a3 | a2 | a1 | a0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 1  | 0  | 0  | 0  | 1  | 1  |

Se cumplen todas las condiciones señaladas en el apartado anterior y por tanto, **se produce el salto** a la dirección ETIQUETA

d) Si A = 32H H, su valor binario es  $a_7a_6a_5a_4a_3a_2a_1a_0$  :

| a7 | a6 | a5 | a4 | a3 | a2 | a1 | a0 |
|----|----|----|----|----|----|----|----|
| 0  | 0  | 1  | 1  | 0  | 0  | 1  | 0  |

Se cumplen todas las condiciones, salvo la condición  $a_5 \neq a_4$  y por tanto, **NO se produce el salto** a la dirección ETIQUETA

e) Tamaño del fragmento en código máquina

| Instrucción  | Tamaño (bytes)  |
|--------------|-----------------|
| MOV B, A     | 1               |
| ROR A        | 1               |
| AND A, 50H   | 2               |
| XOR A,B      | 1               |
| AND A, 5EH   | 2               |
| XOR A,12H    | 2               |
| JZ ETIQUETA  | 2               |
| <b>TOTAL</b> | <b>11 bytes</b> |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 14 de septiembre de 2004 (Tarde)

### PROBLEMA. 2

(2.5 Puntos)

Dado el siguiente programa de un INDALO 3.0 que funciona a 5MHZ:

```
ORG      1000H
datos: DB  3,0,0
MOV      X,datos
MOV      B,(X)
bucle: CALL subrut1
DEC      B
JZ        fin
CALL     subrut1
JMP      bucle
fin:     HLT
```

Se pide:

- A) Calcular el tiempo de ejecución sin incluir el que se tarda en la subrutina **subrut1**, y sabiendo que inicialmente PC=1003H y que la subrutina no modifica el contenido del registro B. (0,7 ptos.)
- B) Indicar cuantas veces se llama a esta subrutina. (0,3 ptos.)
- C) Desensamblar la subrutina **subrut1**, que en lenguaje máquina es la siguiente:  
6EH, 01, 8AH, 49H, 01, 6EH, 02, 94H, 00, 49H, 02, 08.  
(0,7 ptos.)
- D) ¿Qué hace la subrutina, independientemente del programa principal?. (0,4 ptos.)
- E) ¿Qué quedará en **datos** después de ejecutar el programa completamente?. (0,4 ptos.)

### SOLUCIÓN

#### Cálculos

|             |         | Ciclos<br>de reloj | Veces que<br>se ejecuta | Ciclos<br>totales |
|-------------|---------|--------------------|-------------------------|-------------------|
| ORG         | 1000H   |                    |                         |                   |
| datos: DB   | 3,0,0   |                    |                         |                   |
| MOV         | X,datos | 15                 | 1                       | 15                |
| MOV         | B,(X)   | 13                 | 1                       | 13                |
| bucle: CALL | subrut1 | 27                 | 3                       | 81                |
| DEC         | B       | 7                  | 3                       | 21                |
| JZ          | fin     | 7/12               | 2/1                     | 26                |
| CALL        | subrut1 | 27                 | 2                       | 54                |
| JMP         | bucle   | 15                 | 2                       | 30                |
| fin: HLT    |         | 3                  | 1                       | 3                 |



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 14 de septiembre de 2004 (Tarde)

### PROBLEMA. 2

(cont.)

#### Solución apartado A:

$15+13+81+21+26+54+30+3=243$  ciclos de reloj.  
 $243/5*10^6=48,6 \mu\text{seg.}$

#### Solución apartado B:

5 veces

#### Solución apartado C:

| LENGUAJE MÁQUINA | ENSAMBLADOR |
|------------------|-------------|
| 6EH, 01          | MOV A,(X+1) |
| 8AH              | ADD A,B     |
| 49H, 01          | MOV (X+1),A |
| 6EH, 02          | MOV A,(X+2) |
| 94H, 00          | ADC A,0     |
| 49H, 02          | MOV (X+2),A |
| 08               | RET         |

#### Solución apartado D:

Suma el byte que ocupa la dirección (datos) a la word que ocupa las direcciones (datos+1) y (datos+2).

#### Solución apartado E:

datos DB 3,9,0

Ya que a la word que ocupa las direcciones (datos+1) y (datos+2) se le suman consecutivamente los valores 3, 2, 2, 1 y 1. De hecho, aunque esto obviamente no se pedía, el programa lo que hace es calcular el cuadrado del número que hay en la dirección (datos) usando el conocido algoritmo de sumar los "n" primeros números impares, y dejar el resultado en la word que ocupa las direcciones (datos+1) y (datos+2).



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Tarde)**

---

### **PROBLEMA. 3**

**(2.5 Puntos)**

Sea el siguiente programa en ensamblador:

```
ORG 2FF0H
MOV C,70H
IN A,C
SHL A
JC sigue
MOV C,71H
IN A,C
SHL A
JC sigue1
JMP sigue2
sigue: OUT C,A
JMP fin
sigue1: OUT C,A
MOV C,70H
OR A,80H
OUT C,A
JMP fin
sigue2: MOV A,0FFH
fin: HLT
```

- 1) Ensambladlo (Obtened el mismo programa en código máquina) (1Pto)
- 2) Si la frecuencia del reloj del sistema es de 1GHz y los contenidos de los puertos 70H y 71 H son 80h y 90h respectivamente ,obtened el tiempo que tarda en ejecutarse el programa.
- 3) Si los contenidos de los puertos 70h y 71h son 60h y 35h respectivamente, obtened el valor final del registro A





# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **14 de septiembre de 2004 (Tarde)**

### PROBLEMA. 3

(cont.)

### SOLUCIÓN

1) A partir de las tablas de instrucciones se obtienen los siguientes valores:

|                    | Dirección               | Código Máquina | Nº de Ciclos |
|--------------------|-------------------------|----------------|--------------|
| ORG 2FF0H          | <i>No genera código</i> |                |              |
| MOV C,70H          | 2FF0H                   | 7C 70          | 10           |
| IN A,C             | 2FF2H                   | 0F             | 8            |
| SHL A              | 2FF3H                   | E0             | 7            |
| JC sigue           | 2FF4H                   | 11 09          | 7/12         |
| MOV C,71H          | 2FF6H                   | 7C 71          | 10           |
| IN A,C             | 2FF8H                   | 0F             | 8            |
| SHL A              | 2FF9H                   | E0             | 7            |
| JC sigue1          | 2FFAH                   | 11 07          | 7/12         |
| JMP sigue2         | 2FFCH                   | 12 0C 30       | 15           |
| sigue: OUT C,A     | 2FFFH                   | 20             | 8            |
| JMP fin            | 3000H                   | 12 0E 30       | 15           |
| sigue1: OUT C,A    | 3003H                   | 20             | 8            |
| MOV C,70H          | 3004H                   | 7C 70          | 10           |
| OR A,80H           | 3006H                   | C4 80          | 10           |
| OUT C,A            | 3008H                   | 20             | 8            |
| JMP fin            | 3009H                   | 12 0E 30       | 15           |
| sigue2: MOV A,0FFH | 300CH                   | 6C FF          | 10           |
| fin: HLT           | 300EH                   | 0E             | 3            |

2) Siguiendo la ejecución del código, en función de los datos del enunciado, resulta:

| Instrucciones ejecutadas | Resultado            | Ciclos    |
|--------------------------|----------------------|-----------|
| MOV C,70H                | C=70H                | 10        |
| IN A,C                   | puerto70H=A=80H      | 8         |
| SHL A                    | A=00 „ Fc=1          | 7         |
| JC sigue                 | salta a <i>sigue</i> | 12        |
| sigue: OUT C,A           | puerto70H=00         | 8         |
| JMP fin                  | salta a <i>fin</i>   | 15        |
| fin: HLT                 | parada               | 3         |
| <b>TOTAL=</b>            |                      | <b>63</b> |

Como la frecuencia es de 1GHz, el tiempo de ejecución será:  **$t_{\text{ejecución}} = 63 \text{ ns}$**



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: Fundamentos de Computadores (Prob.) CURSO: 1º GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: 14 de septiembre de 2004 (Tarde)

**PROBLEMA. 3**

**(cont.)**

3) Siguiendo la ejecución del código, en función de los datos del enunciado, resulta:

| Instrucciones ejecutadas   | Resultado                |
|----------------------------|--------------------------|
| MOV C,70H                  | C=70H                    |
| IN A,C                     | Puerto70H=A=60H=01100000 |
| SHL A                      | A=11000000=C0H,, Fc=0    |
| JC sigue                   | No salta                 |
| MOV C,71H                  | C=71H                    |
| IN A,C                     | Puerto71H=A=35H=00110101 |
| SHL A                      | A=01101010=6AH,, Fc=0    |
| JC sigue1                  | No salta                 |
| JMP sigue2                 | Salta a <i>sigue2</i>    |
| <i>sigue2</i> : MOV A,0FFH | A=FFH                    |
| HLT                        | parada                   |

De modo que el valor resultante del acumulador es: **A=FFH**.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_

ASIGNATURA: **Fundamentos de Computadores (Prob.)**

CURSO: **1º**

GRUPO: \_\_\_\_\_

Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

### **TEST**

**(2.5 Puntos)**

En las siguientes preguntas de tipo test, sólo existe una respuesta correcta. Cada pregunta tiene un valor de 0,25 pts. Cada respuesta incorrecta descuenta 0,1 pts. La pregunta que deje en blanco no sumará ni restará puntos.

**Notas:** • Se debe poner una X sobre la letra de la tabla "SOLUCIÓN" (que encontrará en la hoja siguiente) correspondiente a la respuesta correcta.

• La tabla "CALIFICACIÓN" no deberá utilizarla, es para su posterior calificación.

• No se tendrán en cuenta las respuestas que no se encuentren en dicha tabla "SOLUCIÓN"

1.- Las señales *int* e *inta* afectan a:

- a) Las interrupciones software.
- b) Las interrupciones internas o excepciones.
- c) Las interrupciones hardware.
- d) Ningún tipo de interrupciones.

2.- ¿Qué instrucción incrementa el PC tras leerse su último byte de la memoria?:

- a) JMP.
- b) RET.
- c) HLT.
- d) CALL.

3.- La principal diferencia entre las instrucciones CALL e INT radica en que:

- a) CALL no incrementa el PC al ser leída.
- b) INT modifica el *flag* F<sub>1</sub> al ser ejecutada.
- c) INT se produce de forma asíncrona con el programa en ejecución.
- d) CALL no tiene que calcular la dirección de comienzo de la subrutina.

4.- Considerando que Dato1 es una etiqueta, ¿cuál de las siguientes instrucciones sería correcta?:

- a) MOV A, Dato1
- b) MOV BC, (Dato1)
- c) MOV A, (Dato1)
- d) MOV X, (Dato1)

5.- En el Indalo 3.0, ¿dónde se guarda la dirección de retorno de una subrutina?:

- a) En el registro PC.
- b) En el registro SP.
- c) En el registro X.
- d) Ninguna de las respuestas anteriores es correcta.



# UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID

## Departamento de Electrónica y Comunicaciones

APELLIDOS Y NOMBRE: \_\_\_\_\_

ASIGNATURA: **Fundamentos de Computadores (Prob.)**

CURSO: **1º**

GRUPO: \_\_\_\_\_

Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

---

### CONTINUACIÓN TEST

---

6.- La señal CS (*Chip Select*) actúa sobre:

- a) El decodificador externo de la memoria.
- b) El decodificador interno de los circuitos integrados de memoria.
- c) Sobre el *driver* bidireccional de los circuitos integrados de memoria.
- d) Sobre el circuito que gestiona los ciclos de espera.

7.- En la ALU que utiliza la familia INDALO, se activa el flag ...:

- a)  $F_C$  si, al restar, hay acarreo final.
- b)  $F_O$  si, al desplazar, el bit que sale por un extremo es diferente al que entra por el otro.
- c)  $F_O$  si, al desplazar, sale un bit a 1 por la derecha.
- d)  $F_C$  si, al restar, el minuendo es menor que el sustraendo.

8.- En el INDALO 3.0, el vector de interrupción ...:

- a) Se almacena en la pila.
- b) Sólo puede ser suministrado por la instrucción INT.
- c) Se almacena en el registro  $DIR_L$ .
- d) Se guarda en la tabla de vectores que comienza en la posición 0000H.

9.- En el INDALO 3.0, los puertos ...:

- a) Utilizan direcciones de 8 bits que viajan por el bus de datos (DBUS).
- b) Utilizan direcciones de 16 bits almacenadas en el registro C.
- c) Se direccionan a través del registro X.
- d) Trabajan con direcciones de 8 bits, pero que son enviadas por el bus de direcciones (ABUS).

10.- ¿Qué instrucción de salto utiliza para su lectura y ejecución la secuencia de operaciones F, F,  $F_{noinc}$ ,  $I_{t16}$ ?:

- a) JMP 3F2FH.
- b) JNP 3F2FH.
- c) JMP BC.
- d) JNP BC.



**UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID**  
**Departamento de Electrónica y Comunicaciones**

APELLIDOS Y NOMBRE: \_\_\_\_\_  
ASIGNATURA: **Fundamentos de Computadores (Prob.)** CURSO: **1º** GRUPO: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ CONVOCATORIA: **7 de junio de 2004. (Tarde)**

---

**CONTINUACIÓN TEST**

---

**SOLUCIÓN**

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|----|
| a | a | a | a | a | a | a | a | a | a  |
| b | b | b | b | b | b | b | b | b | b  |
| c | c | c | c | c | c | c | c | c | c  |
| d | d | d | d | d | d | d | d | d | d  |

La respuesta correcta se indicará tachando la letra correspondiente con una **X**.

Para anular una respuesta dada, se rodeará con una circunferencia  $\otimes$ , de modo que se entenderá:

|           |                                  |
|-----------|----------------------------------|
| $\otimes$ | respuesta dada por el alumno.    |
| $\otimes$ | respuesta anulada por el alumno. |

**CALIFICACIÓN**

|                  |  |                    |   |
|------------------|--|--------------------|---|
| <b>Correctas</b> |  | <b>x 0,25 =</b>    |   |
| <b>Erróneas</b>  |  | <b>x (- 0,1) =</b> | - |
|                  |  | <b>Total</b>       |   |