



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 1** **(2.5 puntos)**

Rellenar la tabla siguiente indicando en cada caso la instrucción que se ejecuta, las microinstrucciones y microórdenes necesarios para ello, así como las operaciones básicas. Completar también los espacios en blanco de las columnas ABUS y DBUS.

Microinstrucciones	Microórdenes	Ops. básicas	Instrucciones	ABUS	DBUS
PC → ABUS				3000 H	
(ABUS) → DBUS					
DBUS → CO					
PC++					
0 → Fc					
B → DBUS					
A-DBUS-Fc → A					
	salpc				
	mem, rd				
	ckco				
	reg1, ckpcl, ckpch				
	salpc				
	mem, rd				05H
	ckdir1				
	reg1, ckpcl, ckpch				
	reg1, reg0, ckspl, cksph				
	salsp				
	salpch				
	mem, wr				
	reg1, reg0, ckspl, cksph				
	salsp				
	salpcl				
	mem, wr				
	reg0, ckdir1, ckdirh				
	saldir				
	mem, rd				
	reg0, ckpcl				
	reg1, ckdir1, ckdirh				
	saldir				
	mem, rd				
	reg0, ckpch				

**NOTA:** Téngase en cuenta que la parcelación de las columnas no es completa, por lo que pueden estar sujetas a nuevas subdivisiones durante la realización del ejercicio. Háganse todas aquellas que se consideren necesarias.



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

**FUNDAMENTOS DE COMPUTADORES (Problemas. Examen Final. Mañana)**

APELLIDOS Y NOMBRE: \_\_\_\_\_  
 Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 1: Solución (y 1)**

Microinstrucciones	Microórdenes	Ops. básicas	Instrucciones	ABUS	DBUS	
PC → ABUS	<i>salpc</i>	<i>F<sub>flag</sub></i>	<i>SUB A,B</i>	3000 H		
(ABUS) → DBUS	<i>mem, rd</i>					
DBUS → CO	<i>ckco</i>				<i>0F2H</i>	
PC++	<i>reg1,ckpcl,ckpch</i>					
0 → Fc	<i>quitfc</i>					
B → DBUS	<i>salb</i>	<i>I<sub>alu</sub></i>				
A-DBUS-Fc → A	<i>alu3, alu1, alu0, ckfl, ckfc, cka</i>					
<i>PC → ABUS</i>	salpc	<i>F</i>	<i>INT 05H</i>	<i>3001H</i>		
<i>(ABUS) → DBUS</i>	mem, rd					
<i>DBUS → CO</i>	ckco				<i>10H</i>	
<i>PC++</i>	reg1, ckpcl, ckpch	<i>F</i>				
<i>PC → ABUS</i>	salpc					
<i>(ABUS) → DBUS</i>	mem, rd				05H <sup>1</sup>	
<i>DBUS → DIR<sub>L</sub></i>	ckdirl	<i>I<sub>id16</sub></i>				
<i>PC++</i>	reg1, ckpcl, ckpch					
<i>SP--</i>	reg1,reg0,ckspl, cksph	<i>W</i>				
<i>SP → ABUS</i>	salsp					
<i>PC<sub>H</sub> → DBUS</i>	salpch				<i>30H<sup>2</sup></i>	
<i>DBUS → (ABUS)</i>	mem, wr	<i>I<sub>id16</sub></i>				
<i>SP--</i>	reg1,reg0,ckspl, cksph					
<i>SP → ABUS</i>	salsp	<i>W</i>				
<i>PC<sub>L</sub> → DBUS</i>	salpcl					
<i>DBUS → (ABUS)</i>	mem, wr	<i>I<sub>2dirl</sub></i>				
<i>2*DIR<sub>L</sub> → DIR</i>	reg0, ckdirl, ckdirh					
<i>DIR → ABUS</i>	saldir	<i>R</i>		<i>000AH<sup>3</sup></i>		
<i>(ABUS) → DBUS</i>	mem, rd					
<i>DBUS → PC<sub>L</sub></i>	reg0, ckpcl					
<i>DIR++</i>	reg1, ckdirl, ckdirh	<i>I<sub>id16</sub></i>				
<i>DIR → ABUS</i>	saldir					
<i>(ABUS) → DBUS</i>	mem, rd	<i>R</i>				
<i>DBUS → PC<sub>H</sub></i>	reg0, ckpch					

<sup>1</sup> Vector de interrupción.

<sup>2</sup> Parte alta de la dirección de retorno.

<sup>3</sup> Parte baja de la entrada de la tabla de vectores correspondiente a la interrupción de vector 05H.



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2** **(2.5 puntos)**

Se dispone de la siguiente subrutina (`subr`), la cual trabaja sobre la información almacenada en una tabla de datos (`tabla`).

```
tabla:    DB      2CH, 0B7H, 70H, 69H, 34H, 00H
          ...
subr:     MOV     X, tabla
          MOV     C, 70H
bucle:    MOV     A, (X)
          CMP     A, C
          JZ      final
          INC     X
          AND     A, 0
          CMP     A, (X)
          JZ      final
          JMP     bucle
final:    RET
```

- a) Ensamblar la subrutina suponiendo que ésta se encuentra almacenada en memoria a partir de la posición 1000H y que los datos de la tabla que maneja comienzan en 0800H. Haga constar el resultado en la tabla siguiente, utilizando una fila para cada línea de programa **a partir de la primera instrucción** (`MOV X, tabla`) e indicando, además, la posición donde comienza.

(1 punto)

INSTRUCCIÓN	DIRECCIÓN	CÓDIGO MÁQUINA
MOV X, tabla	1000H	
MOV C, 70H		
MOV A, (X)		
CMP A, C		
JZ final		
INC X		
AND A, 0		
CMP A, (X)		
JZ final		
JMP bucle		
RET		



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2**

(continuación)

- b) Realizar un seguimiento de dicha subrutina sobre la siguiente tabla, indicando en ella el número de iteraciones que realiza el programa y, para cada una de ellas, especificando el número de ciclos de cada instrucción **que se ejecute**. En la última columna de la derecha, deberá anotarse el total de ciclos que utiliza cada instrucción considerando todas las iteraciones indicadas. En la casilla recuadrada del final de la tabla debe figurar la suma de las cantidades que se encuentren encima suyo, es decir, la totalidad de los ciclos que requiere la subrutina para ejecutarse.

(1.5 puntos)

			Iterac. 1	Total
subr:	MOV	X, tabla		
	MOV	C, 70H		
bucle:	MOV	A, (X)		
	CMP	A, C		
	JZ	final		
	INC	X		
	AND	A, 0		
	CMP	A, (X)		
	JZ	final		
	JMP	bucle		
final:	RET			
TOTAL				



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2: Solución (1)**

**Apartado a)**

INSTRUCCIÓN	DIRECCIÓN	CÓDIGO MÁQUINA
MOV X, tabla	1000H	56H, 00H, 08H
MOV C, 70H	1003H	7CH, 70H
MOV A, (X)	1005H	6EH, 00H
CMP A, C	1007H	83H
JZ final	1008H	1BH, 0AH
INC X	100AH	35H
AND A, 0	100BH	9CH, 00H
CMP A, (X)	100DH	86H, 00H
JZ final	100FH	1BH, 03H
JMP bucle	1011H	12H, 05H, 10H
RET	1014H	08H

NOTA: Para aquellos puntos donde figuren etiquetas y/o saltos condicionales, téngase en cuenta que *tabla*=0800H, *final*=1014H y *bucle*=1005H.

**Apartado b)**

Mediante una estructura de bucle, la subrutina recorre con ayuda del registro X una lista de datos almacenados en memoria desde la posición etiquetada como "tabla" y compara el contenido de esas posiciones con el del registro C, cuyo valor siempre es 70H. Si se da esa coincidencia de valores, se procede a salir del bucle y a dar por finalizada la subrutina.

También se comprueba dentro del bucle si X llega al final de la lista de datos (00H), aunque esto último nunca produce, pues en la tercera iteración hay coincidencia con el dato 70H, saliendo así del bucle.

Según lo expuesto, la tabla pedida queda como sigue:



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2: Solución (y 2)**

			Iterac. 1	Iterac. 2	Iterac. 3	Total
subr:	MOV	X, tabla	15			15
	MOV	C, 70H	10			10
bucle:	MOV	A, (X)	13	13	13	39
	CMP	A, C	7	7	7	21
	JZ	final	7	7	12	26
	INC	X	7	7		14
	AND	A, 0	10	10		20
	CMP	A, (X)	13	13		26
	JZ	final	7	7		14
	JMP	bucle	15	15		30
final:	RET				13	13
TOTAL						228





UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Mañana*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3**

**(2.5 puntos)**

Se dispone de la siguiente subrutina para el INDALO 3.0:

```
subr:    PUSH BC
         PUSH X
         MOV X, 2000H
         MOV C, 30H
bucle:   MOV B, (X)
         IN A, C
         CMP A, B
         JZ final
         INC X
         JMP bucle
final:   POP X
         POP BC
         RET
```

También se conoce el contenido de algunos registros, posiciones de memoria y puertos de E/S **antes de empezar a ejecutar ese código**:

Puerto de E/S	Contenido	Dirección	Contenido	Registro	Contenido
10H	34H	0038H	33H	PC	1000H
20H	0EFH	0039H	22H	SP	0500H
2AH	0ABH	0040H	2CH	BC	2030H
2FH	00H	0041H	1FH	X	1C2DH
30H	25H	0042H	49H		
34H	10H	...	...		
3FH	22H	2000H	0DDH		
		2001H	0EH		
		2002H	25H		
		2003H	00H		
		2004H	52H		

Suponiendo que las interrupciones están habilitadas, responda **breve y razonadamente** a las siguientes preguntas:

- a) ¿Cuál es el contenido del registro SP después de ejecutar la instrucción PUSH X? ¿Qué contendrá la celda de memoria señalada por ese registro?

**(0.5 puntos)**



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

**FUNDAMENTOS DE COMPUTADORES** (*Problemas. Examen Final. Mañana*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

---

**PROBLEMA 3**

**(continuación)**

- b) Con los datos del programa, calcule dónde empieza la rutina de servicio de la interrupción 20H.  
(0.5 puntos)
- c) En la primera vez que se ejecuta el bucle que incluye la subrutina se produce la interrupción 20H mientras se está procesando la instrucción JZ final, ¿cuál es el valor de SP y PC cuando vaya a leerse la primera instrucción de la rutina de servicio de esa interrupción? ¿Cuál es la dirección de retorno que se ha guardado en la pila?  
(0.8 puntos)
- d) ¿Qué valores contendrán los registros A, B, C y SP cuando se termine de ejecutar la instrucción RET? ¿Cuántas veces se ejecuta el bucle?  
(0.7 puntos)





UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (Problemas. Examen Final. Mañana)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3: Solución (1)**

Mediante una estructura de bucle, la subrutina recorre una lista de datos almacenados en memoria desde la posición 2000H con ayuda del registro X y compara el contenido de esas posiciones con el del puerto de E/S de dirección 30H, cuyo valor siempre se vuelca sobre el acumulador A. Si se da esa coincidencia de valores, se procede a salir del bucle, dejar los registros X Y BC como estaban antes de comenzar la ejecución de la subrutina y a retornar al punto del programa donde fue invocada. En caso contrario, se incrementa X para avanzar por la lista de datos y realizar nuevas comparaciones.

Dado que el contenido del puerto 30H y el de la celda de dirección 2002H son iguales (25H), en la tercera iteración del bucle se produce la coincidencia comentada.

**Apartado a)**

Las dos primeras instrucciones PUSH de la subrutina guardan en pila el contenido inicial de los registros BC y X, que son los dados en el enunciado (BC = 2030H y X = 1C2DH). Teniendo en cuenta, además, que inicialmente SP = 0500H, la pila y este último registro quedarán de la siguiente manera:

Dirección	Contenido	Comentario
SP = 04FCH	2DH	Contenido de X
04FDH	1CH	
04FEH	30H	Contenido de BC
04FFH	20H	
0500H	?	

**Apartado b)**

$$2 * \text{vector} = 2 * 20H = 40H = 0040H \Rightarrow \text{addr}_L = (0040H) = 2CH$$

$$2 * \text{vector} + 1 = 0041H \Rightarrow \text{addr}_H = (0041H) = 1FH$$

$$\text{addr} = 1F2CH$$



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (Problemas. Examen Final. Mañana)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3: Solución (2)**

**Apartado c)**

La interrupción no se atiende hasta completar la ejecución de la instrucción en curso. Puesto que estamos en la primera iteración del bucle, no hay coincidencia en la comparación realizada, por lo que el salto condicional (JZ final) no se produce. En consecuencia, después de ejecutar esa instrucción, el registro PC apunta a la siguiente, que es INC X y que comienza en la posición 100DH (téngase en cuenta que la primera instrucción de la subrutina comienza donde esté apuntando el PC antes de iniciarse la ejecución de dicha subrutina, es decir, 1000H según el enunciado). Es entonces ésa la dirección de retorno que debe guardarse en la pila antes de invocar a la rutina de servicio de la interrupción. Después de eso, se carga en PC la dirección de inicio de dicha rutina, la cual se ha calculado en el apartado previo. Así:

Dirección	Contenido	Comentario
SP = 04FAH	0DH	Dirección de retorno (Contenido de PC)
04FBH	10H	
04FCH	2DH	Contenido de X
04FDH	1CH	
04FEH	30H	Contenido de BC
04FFH	20H	
0500H	?	

PC = addr = 1F2CH

**Apartado d)**

Antes de RET, previamente se han ejecutado las dos instrucciones POP, por lo que los registros X y BC contendrán lo que tenían antes de empezar la ejecución de la subrutina. Así:

B = 20H

C = 30H

Por otro lado, RET restaura la dirección de retorno de la subrutina. Eso implica incrementar dos veces el valor de SP después de ejecutar las instrucciones POP, luego:

SP = 0502H

En cuanto al registro A, siempre contiene el dato almacenado en el puerto de E/S de dirección 30H para cada una de las comparaciones (téngase en cuenta lo comentado al principio de esta solución):

A = 25H

Por último, y como ya se ha dicho al principio, el bucle se ejecuta **tres veces**.



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 1** (2.5 puntos)

Rellenar la tabla adjunta indicando en cada caso la instrucción que se ejecuta, las microinstrucciones y microórdenes necesarias para ello, así como las operaciones básicas. Completar también los espacios en blanco de las columnas ABUS y DBUS.

Instrucciones	µInstrucciones	µÓrdenes	Ops. básicas	ABUS	DBUS
	PC → ABUS			2000H	
	(ABUS) → DBUS				
	DBUS → CO				7EH
	PC++				
	PC → ABUS				
	(ABUS) → DBUS				
	DBUS → DIRL				00H
	PC++				
	X → ABUS				
	ABUS+DIRL → ABUS				
	(ABUS) → DBUS				
	DBUS → C				
	PC → ABUS				
	(ABUS) → DBUS				
	DBUS → CO				
	PC++				
	0 → Fc				
		alu3,alu0,cka,ckfl,ckfc			
	PC → ABUS				
	(ABUS) → DBUS				
	DBUS → CO				
	PC++				

**NOTA:** Téngase en cuenta que la parcelación de las columnas no es completa, por lo que pueden estar sujetas a nuevas subdivisiones durante la realización del ejercicio. Háganse todas aquellas que se consideren necesarias.



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 1: Solución (y 1)**

Instrucciones	$\mu$ Instrucciones	$\mu$ Órdenes	Ops. básicas	ABUS	DBUS
<b>MOV C,(X)</b>	PC $\rightarrow$ ABUS		<b>F</b>	2000H	
	(ABUS) $\rightarrow$ DBUS				
	DBUS $\rightarrow$ CO				7EH
	PC++				
	PC $\rightarrow$ ABUS		<b>F</b>		
	(ABUS) $\rightarrow$ DBUS				
	DBUS $\rightarrow$ DIRL			<b>2001H</b>	00H
	PC++				
	X $\rightarrow$ ABUS		<b>Ralu</b>		
	ABUS+DIRL $\rightarrow$ ABUS				
	(ABUS) $\rightarrow$ DBUS				
	DBUS $\rightarrow$ C				
<b>SHL A</b>	PC $\rightarrow$ ABUS		<b>Fflag</b>	<b>2002H</b>	
	(ABUS) $\rightarrow$ DBUS				
	DBUS $\rightarrow$ CO				<b>E0H</b>
	PC++				
	0 $\rightarrow$ Fc		<b>Italu</b>		
	<b>A &lt; rc <math>\leftrightarrow</math> A</b>	alu3,alu0,cka,ckfl,ckfc			
<b>NOP</b>	PC $\rightarrow$ ABUS		<b>F</b>		
	(ABUS) $\rightarrow$ DBUS				
	DBUS $\rightarrow$ CO				<b>00H</b>
	PC++				



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2**

**(2.5 puntos)**

Un microprocesador basado en el Indalo 3.0, está ejecutando el siguiente fragmento de programa

```
ORG      2000H
MOV      B, (2500H)
MOV      C, (2501H)
XOR      A, A
JZ       ETQ
```

Además, se conocen los siguientes datos iniciales:

PC = 2000H, SP = 0FF00H

y los contenidos de las siguientes posiciones de memoria

Posición	Contenido
0005H	23H
0006H	10H
0007H	21H
0008H	30H
-----	-----
2007H	1BH
2008H	05H
2009H	9DH

En estas condiciones, se pide:

- Determinar el espacio que ocupa en memoria el fragmento de programa. ¿Cuál es la dirección correspondiente a la etiqueta ETQ? ¿Cuál sería la instrucción que habría a continuación del fragmento de programa?  
(0.8 puntos)
- Cuando se está ejecutando la instrucción JZ ETQ se produce una interrupción con vector 03H. Indicar el contenido de la pila **antes de leer** el código de operación de la primera instrucción de la rutina de atención a la interrupción, así como los valores de PC y SP en ese momento.  
(0.8 puntos)
- Escribir una tabla en la que se indiquen los ciclos de máquina que se ejecutan desde que se recibe la interrupción hasta **justo antes de leer** la primera instrucción de la rutina de atención. Especificar el tipo de cada ciclo, así como los contenidos de los buses durante cada uno de ellos.  
(0.9 puntos)



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2: Solución (1)**

**Apartado a)**

<b>Instrucción</b>	<b>Nº Bytes</b>
MOV B,(2500H)	3
MOV C,(2501H)	3
XOR A,A	1
JZ ETQ	2

Por lo tanto, este fragmento de programa ocupa:

9 Bytes

Según el problema, la primera instrucción empieza en la dirección 2000H, por lo tanto, la instrucción JZ ETQ ocupará las posiciones 2007H y 2008H y, según la tabla de memoria:

(2007H) = 1BH → Código de operación de la instrucción JZ

(2008H) = 05H → Cantidad que hay que sumar al PC para obtener la dirección de salto

Dirección = PC + 05H = 2009H + 05H = 200EH

ETQ = 200EH

El contenido de la siguiente posición de memoria es 9DH. Este es el código de operación de la siguiente instrucción:

9DH = 10011101 → Código 10011op8

Esto se corresponde con la instrucción AND A,op8 donde op8 = 101 = (BC). Así, la instrucción es:

AND A, (BC)





UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 2: Solución (y 2)**

**Apartado b)**

Después de la ejecución de  $XOR\ A,A$ , el resultado es 0, lo que implica que  $F_z=1$ , luego se cumple la condición de salto y, así, al terminar de ejecutarse la instrucción, el contenido del PC será:

$$PC = 200EH$$

Esta es la dirección que se guardará en la pila, quedando como sigue:

Dirección	Contenido
0FEFEH	0EH
0FEFFH	20H
0FF00H	?

donde el puntero SP toma el valor:

$$SP = 0FEFEH$$

En el PC se cargará la dirección donde empieza la rutina de atención:

$$V*2 = 06H \rightarrow \text{De la tabla de memoria: } (0006H) = 10H$$

$$V*2+1 = 07H \rightarrow (0007H) = 21H$$

Por lo tanto:

$$PC = 2110H$$

**Apartado c)**

TIPO	ABUS	DBUS
RI	-----	03H
W	0FEFFH	20H
W	0FEFEH	0EH
R	0006H	10H
R	0007H	21H



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

APELLIDOS Y NOMBRE: \_\_\_\_\_  
Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3**

**(2.5 puntos)**

El siguiente programa implementa un reloj en tiempo real. El funcionamiento es el siguiente: A partir de un cristal de cuarzo, se genera una frecuencia que, a través de divisores se convierte en una señal de 1 Hz. Esta señal produce interrupciones cada segundo que actualizan los registros que contiene los minutos y los segundos.

```
ORG      2000H
XOR      A, A
MOV      A, B
CMP      A, 59
JZ       ETQ1
INC      B
JMP      FIN
ETQ1:    MOV      B, 0
MOV      A, C
CMP      A, 59
JZ       ETQ2
INC      C
JMP      FIN
ETQ2:    MOV      C, 0
FIN:     HLT
```

Sabiendo que la frecuencia de la CPU es de 100 MHz, se pide:

- Código máquina del programa. (0.8 puntos)
- Espacio que ocupa en memoria. (0.4 puntos)
- Analizar el programa. Indique qué registro almacena los segundos y cuál los minutos. (0.5 puntos)
- ¿Cuánto tiempo tarda en ejecutarse el programa en el caso de que  $B = 59$ ? ¿Y en el caso de que  $B < 59$ ? (0.8 puntos)



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3: Solución (1)**

DIRECCIÓN	INSTRUCCIÓN	CÓDIGO	Nº DE CICLOS
2000H	XOR A,A	F9H	7
2001H	MOV A,B	6AH	7
2002H	CMP A,59	84H, 3BH	10
2004H	JZ ETQ1	1BH, 04H	7/12
2006H	INC B	AAH	7
2007H	JMP FIN	12H, 17H, 20H	15
200AH = ETQ1	MOV B,0	74H, 00H	10
200CH	MOV A,C	6BH	7
200DH	CMP A,59	84H, 3BH	10
200FH	JZ ETQ2	1BH, 04H	7/12
2011H	INC C	ABH	7
2012H	JMP FIN	12H, 17H, 20H	15
2015H = ETQ2	MOV C,0	7CH, 00H	10
2017H = FIN	HLT	0EH	3

**Apartado a)**

De la tabla anterior, y viendo los bytes que ocupa cada instrucción, podemos deducir que el espacio que ocupa el programa en memoria es:

24 Bytes

**Apartado b)**

Como se indica en el enunciado, se trata de un reloj en tiempo real, por lo tanto, el primer registro en actualizarse será el de los segundos, y después el de los minutos. Siguiendo la secuencia del programa, vemos que, en primer lugar, se comprueba si B ha alcanzado el valor 59. Si no es así, se incrementa este registro y se sale del programa hasta la siguiente interrupción. Cuando B = 59, se habrá completado un minuto, con lo cual se pasará a actualizar el registro de los minutos de forma similar a como se ha hecho con el de los segundos (previamente se habrá puesto a cero el registro B). El programa termina cada vez que se actualiza el registro B o, si B = 59, cada vez que se actualiza el registro C. Entonces:

B → segundos

C → minutos



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (Problemas. Examen Final. Tarde)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

**PROBLEMA 3: Solución (2)**

**Apartado c)**

- Si  $B < 59$ , se ejecutan las siguientes instrucciones:

Instrucción	Nº Ciclos
XOR A,A	7
MOV A,B	7
CMP A,59	10
JZ ETQ1	<u>7</u>
INC B	7
JMP FIN	15
HLT	3

Todas las instrucciones se ejecutan una vez. El número total de ciclos de reloj será la suma de todos ellos, lo que da un total de 56 ciclos y el tiempo será:

$$T = \text{Nº Ciclos} * \text{Tiempo de ciclo} = 56 * (1/100 \text{ MHz}) = 56 * 10^{-8} \text{ s} = 0,56 * 10^{-6} \text{ s} = \boxed{0,56 \mu\text{s}}$$

- Si  $B = 59$ , pueden ocurrir dos cosas:
  - Que  $C = 59$  o no. Si  $C < 59$ , se ejecutan las siguientes instrucciones:

Instrucción	Nº Ciclos
XOR A,A	7
MOV A,B	7
CMP A,59	10
JZ ETQ1	<u>12</u>
MOV B,0	10
MOV A,C	7
CMP A,59	10
JZ ETQ2	<u>7</u>
INC C	7
JMP FIN	15
HLT	3

El número de ciclos es 95 y el tiempo es:

$$\boxed{T = 0,95 \mu\text{s}}$$



UNIVERSIDAD PONTIFICIA DE SALAMANCA EN MADRID  
Departamento de Electrónica y Comunicaciones

FUNDAMENTOS DE COMPUTADORES (*Problemas. Examen Final. Tarde*)

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

Nº DE EXPEDIENTE: \_\_\_\_\_ GRUPO: \_\_\_\_\_ CONVOCATORIA: 16 de Junio de 2008

---

**PROBLEMA 3: Solución (y 3)**

---

- Si  $C = 59$ , se ejecutan las siguientes instrucciones

<b><i>Instrucción</i></b>	<b><i>Nº Ciclos</i></b>
<i>XOR A,A</i>	7
<i>MOV A,B</i>	7
<i>CMP A,59</i>	10
<i>JZ ETQ1</i>	<b><u>12</u></b>
<i>MOV B,0</i>	10
<i>MOV A,C</i>	7
<i>CMP A,59</i>	10
<i>JZ ETQ2</i>	<b><u>12</u></b>
<i>MOV C,0</i>	10
<i>HLT</i>	3

El número total de ciclos es de 88 y el tiempo será:

$T = 0,88 \mu s$
------------------