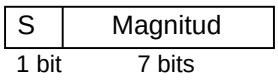
Puntuación: **60%** de la nota final.
Tiempo: **2** horas.

NOTA: Para aprobar el examen es necesario obtener, al menos, 0.5 puntos en cada ejercicio.

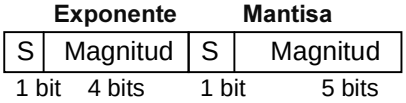
EJERCICIO 1:

Se tiene un computador con los siguientes formatos de representación:

- *Números enteros, representados con 8 bits en signo-magnitud.*



- *Números en coma flotante con las siguientes características:*
  - *Mantisa fraccionaria, normalizada, **con** bit implícito, utilizando 6 bits en signo-magnitud.*
  - *Exponente de 5 bits también representado en signo-magnitud.*



Se pide:

*a. Calcular el rango, ordenado de menor a mayor valor, para la representación en coma fija **(0,25 puntos)**.*

Se sabe que la representación en signo-magnitud tiene un rango simétrico, por tanto, para calcular el rango bastará con conocer el valor máximo y mínimo de la magnitud, $v(x) = \sum_{i=0}^{n-1} bit_i \cdot 2^i$, tal que, n es el número de bits de la mantisa:

- Máxima magnitud representable: 111 1111 ➔ Valor = $2^7 - 1 = 127$
- Mínima magnitud representable: 000 0000 ➔ Valor = 0

Considerando ahora el signo, se obtiene que el rango es:

[-127; -0; 0; 127]

*b. Calcular el rango, ordenado de menor a mayor valor, para la representación en coma flotante **(0,25 puntos)**.*

En la representación de coma flotante, el valor de un número viene dado por $v(x) = \text{valor mantisa} \cdot 2^{\text{valor exponente}}$.

Por tanto, será necesario calcular por separado el rango de la mantisa y el del exponente.

- Rango de la mantisa:

  La representación es en signo-magnitud y fraccionaria, por tanto, bastará con conocer el máximo y mínimo de la magnitud. Considerando, para su cálculo que el número se encuentra normalizado y con bit implícito.

  - Máxima magnitud representable: ,111111= $1 - 2^{-6}$
  - Mínima magnitud representable: ,100000 = 2^{-1}

  El rango de la mantisa será: [$-(1-2^{-6})$; -2^{-1}; 2^{-1} ; $(1-2^{-6})$]

- Rango del exponente (se realizan las mismas consideraciones que en el apartado a.):
  - Máxima magnitud representable: 1111 ➔ Valor = $2^4 - 1 = 15$

    Por tanto, el rango del exponente será: [-15; 15]

Considerando signos y ordenando los valores, se obtiene que el rango, ordenado de menor a mayor, es:

$$[-(1-2^{-6}) \cdot 2^{15};\ -2^{-1} \cdot 2^{-15};\ 2^{-1} \cdot 2^{-15};\ (1-2^{-6}) \cdot 2^{15}]$$

c. Representar el número –240 ***(0,25 puntos)****.*

El número -240 hay que calcularlo utilizando el formato de representación en coma flotante, puesto que dicho número queda fuera del rango de representación permitido para números enteros.

$240 \cdot 10^0$ (base 10 ➔ 1111 0000**,** $\cdot 2^0$ (base 2-coma flotante

Considerando que, en la representación en coma flotante, la coma se sitúa a la izquierda del bit de mayor peso, que en este caso es el bit implícito:

1111 0000**,** $\cdot 2^0$ (base 2-coma flotante ➔ **,**1111 0000 $\cdot 2^8$ (base 2-coma flotante

Siguiendo el segundo formato de representación, el número -240 quedará representado como:

0	1000	1	11100

A la máquina de las características anteriormente mencionadas, le llega la siguiente trama, protegida mediante la técnica del código Hamming:

0 0 0 1 0 0 0 0 1 0 1 1 0 1 0

d.1. Verificar si el número llega correctamente, y en caso negativo corregirlo ***(0,75 puntos)****.*

Al estar el número codificado según el código de Hamming, los bits que ocupan las posiciones correspondientes a potencias de dos: 1, 2, 4 y 8, son los bits de paridad de tal modo que:

- **b1 = bit de paridad**
- **b2 = bit de paridad**
- b3 = b2 + b1
- **b4 = bit de paridad**
- b5 = b4 + b1
- b6 = b4 + b2
- b7 = b4 + b2 + b1
- **b8 = bit de paridad**
- b9 = b8 + b1
- b10 = b8 + b2
- b11 = b8 + b2 + b1
- b12 = b8 + b4
- b13 = b8 + b4 + b1
- b14 = b8 + b4 + b2
- b15 = b8 + b4 + b2 + b1

El bit b1, protegerá a todos aquellos valores en cuya descomposición aparezca un b1. El bit b2, protegerá a todos aquellos valores en cuya descomposición aparezca b2, y así sucesivamente. De tal forma que:

- b1 protege a los bits: b3, b5, b7, b9, b11, b13, b15
- b2 protege a los bits: b3, b6, b7, b10, b11, b14, b15
- b4 protege a los bits: b5, b6, b7, b12, b13, b14, 15
- b8 protege a los bits: b9, b10, b11, b12, b13, b14, b15

De este modo, observamos que:

- b1 tiene paridad par ➔ codificación correcta

- b2 tiene paridad par ➔ codificación correcta
- b4 tiene paridad impar ➔ codificación **incorrecta**. b4 debería ser par → b4=0
- b8 tiene paridad par ➔ codificación correcta

Esto quiere decir que en la transmisión se ha producido un error, por tanto, la codificación Hamming correcta del número: 0 0001 0 11010, es:

000 0000 0101 1010

*d.2. Calcular el valor del número suponiendo que, en la representación en coma flotante descrita anteriormente, **no** se emplea la técnica del bit implícito **(0,25 puntos)**.*

El valor del número recibido, 0 0001 0 11010, es:

- Valor de la mantisa (0 11010) = $2^{-1} + 2^{-2} + 2^{-4}$
- Valor del exponente (0 0001) = 1

Por tanto, el valor del número representado es: $(2^{-1} + 2^{-2} + 2^{-4}) \cdot 2^{1} = 2^{0} + 2^{-1} + 2^{-3} = 1{,}625$.

1,625

*d.3. ¿Qué valor tendría el número si la mantisa fuese fraccionaria, normalizada, **sin** bit implícito pero representada en complemento a 2? **(0,25 puntos)**.*

El valor del número recibido, 0 0001 011010, es:

- Valor de la mantisa (011010) = $2^{-2} + 2^{-3} + 2^{-5}$
- Valor del exponente (0 0001) = 1

Por tanto, el valor del número representado es: $(2^{-2} + 2^{-3} + 2^{-5}) \cdot 2^{1} = 2^{-1} + 2^{-2} + 2^{-4} = 0{,}8125$.

0,8125

EJERCICIO 2:

Dado el siguiente programa escrito en el ensamblador del microprocesador 8086/88 (a la izquierda del código aparece la posición en memoria de la instrucción, seguida de la codificación máquina correspondiente):

```
DATOS SEGMENT
     Acaba          EQU 4Ch
     Incremento     DW  1
     Lista_Datos    DW  0, 0, 0, 0, 0, 0, 0
DATOS ENDS

PILA SEGMENT STACK
    DB 1024 DUP (0)
PILA ENDS
```

```
                                CODIGO SEGMENT
                                     ASSUME CS:CODIGO, DS:DATOS, SS:PILA

                                Principal:
44B5:0    B8 74 44                   mov AX, DATOS
44B5:3    8E D8                      mov DS, AX
44B5:5    33 C0                      xor AX, AX
44B5:7    33 F6                      xor SI, SI
44B5:9    33 C9                      xor CX, CX
44B5:B    B1 07                      mov CL, 07h

                                 Bucle:
44B5:D    C7 84 02 00 FC FF  mov Lista_Datos[SI], 0FFFCh
44B5:13   8B 1E 00 00                mov BX, Incremento
44B5:17   01 9C 02 00                add Lista_Datos[SI], BX
44B5:1B   72 09                      jc Fin
44B5:1D   83 C6 02                   add SI, 2
44B5:20   FF 06 00 00                inc Incremento
44B5:24   E2 E7                      loop Bucle
          Instante B ➔       Fin:
44B5:26   B4 4C                      mov AH, Acaba
44B5:28   CD 21                      int 21h

                                CODIGO ENDS
                                END Principal
```

Los contenidos de los registros y variables de la máquina, en un instante determinado, son los que a continuación se indican:

Reg	Instante ?	Instante B		Instante ?	Instante B		Instante ?	Instante B
AX	*4474 h*	0000 h	**SI**	*0000 h*	0006 h	**DS**	*4474 h*	4474 h
BX	*0000 h*	0004 h	**DI**	*0000 h*	0000 h	**ES**	*4464 h*	4464 h
CX	*0000 h*	0004 h	**SP**	*0400 h*	0400 h	**SS**	*4475 h*	4475 h
DX	*0000 h*	0000 h	**BP**	*0000 h*	0000 h	**CS**	*44B5 h*	44B5 h
			IP	*0005 h*	0026 h			

Variables	Instante ?	Instante B
Lista_Datos	*0..0 0..0 0..0 0..0 0..0 0..0 0..0 (h)*	FFFD FFFE FFFF 0000 0000 0000 0000 (h)
Incremento	*1 h*	4 h

a. *Considerando la información de la tabla anterior, indicar cuál es la próxima instrucción, en lenguaje ensamblador, que se va a ejecutar **(0,1 puntos)**.*

Sabiendo que el registro contador de programa indica la siguiente instrucción a la que se está ejecutando, la próxima instrucción a ejecutar es la que viene indicada en el registro IP, 0005 h:

xor AX, AX

*b. Indicar en qué posición absoluta de memoria se encuentra dicha instrucción **(0,1 puntos)**.*

Para calcularlo basta con ver el segmento de código (CS) y el contador de programa, y realizar los cálculos correspondientes, sabiendo que la memoria del procesador 80x86/88 se encuentra segmentada. Estos son:

CS x 16 d + IP = CS x 10 h + IP = 44B5 h x 10 h + 5 h = 44B55 h:

44B55 h

Observando los tipos de datos declarados al principio del programa, y los contenidos de los registros y variables de la tabla:

*c. Rellenar la tabla anterior, considerando la ejecución del programa **sólo** hasta el instante B **(0,8 puntos)**.*

Ver datos en tabla.

*d. Indicar en qué posición absoluta de memoria se encuentra el elemento Lista_Datos[SI] en el instante B **(0,2 puntos)**.*

Primeramente se calcula la posición de inicio del segmento de datos (DS*10 h), y a continuación, se realizan los cálculos necesarios para obtener la posición del dato.

Observando el segmento de datos definido en el programa, la primera posición es *Incremento,* de 16 bits y tipo DW, el cual ocupa las direcciones 0000 h y 0001 h del segmento de datos. Las siguientes posiciones: [0002 h – 000F h], están ocupadas por Lista_Datos, formada por elementos de 16 bits, definidos con el tipo DW.

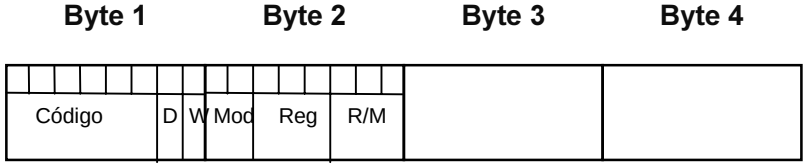
Por tanto, Lista_Datos comienza en la posición relativa 0002 h, y sabiendo que el registro SI contiene un 0006 h, se tiene que:

DS * 16 d = DS * 10 h + Dir. de memoria de comienzo de Lista_Datos + SI =
= 44740 + 0002h + 0006h =

44748 h

e. Calcular el código máquina de la instrucción: mov Lista_Datos[SI], DX, sabiendo que, en este caso, el código de operación de la instrucción mov es 10 0010 b.

*Utilizar las tablas que se muestran a continuación **(0,8 puntos)**.*



D	Información en REG
0	Operando fuente
1	Operando destino

W	Tipos de datos
0	8 bits
1	16 bits

REG	W=0	W=1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

MOD = 11			Cálculo de la dirección efectiva			
R/M	W=0	W=1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	[BX]+[SI]	[BX]+[SI]+D8	[BX]+[SI]+D16
001	CL	CX	001	[BX]+[DI]	[BX]+[DI]+D8	[BX]+[DI]+D16
010	DL	DX	010	[BP]+[SI]	[BP]+[SI]+D8	[BP]+[SI]+D16
011	BL	BX	011	[BP]+[DI]	[BP]+[DI]+D8	[BP]+[DI]+D16
100	AH	SP	100	[SI]	[SI]+D8	[SI]+D16
101	CH	BP	101	[DI]	[DI]+D8	[DI]+D16
110	DH	SI	110	Dirección directa	[BP]+D8	[BP]+D16
111	BH	DI	111	[BX]	[BX]+D8	[BX]+D16

Analizando el formato de las instrucciones del procesador 80x86/88, se observa que el contenido de cada uno de los campos es:

- Código de operación: 1000 10.
- D: 0, puesto que en el campo Reg se introducirá la codificación correspondiente al registro DX, que es el operando fuente.
- W: 1, puesto que se trata de operandos de 16 bits.
- Mod: 10, R/M: 100, puesto que el primer operando es Lista_Datos[SI], es decir, una dirección de memoria determinada por un desplazamiento de 16 bits y por el índice indicado en el registro SI.
- Reg: 010, para especificar el registro DX.

- Los bytes 3 y 4, indican la posición de memoria, relativa al segmento de datos, en la que se encuentra el inicio de Lista_Datos, esto es, 0002 h. Indicando en el byte 3 la parte baja, 02 h (0000 0010 b), y en el byte 4 la parte alta, 00 h (0000 0000 b).

  Quedando como resultado:

C.O.	D	W	Mod	Reg	R/M	Dir_L Lista_Datos	Dir_H Lista_Datos
1000 10	0	1	10	010	100	0000 0010	0000 0000

El código máquina correspondiente a esta instrucción, en hexadecimal, es:

89 94 02 00 h

EJERCICIO 3:

Se dispone de un PC en el cual el bus de direcciones tiene 32 bits y el ancho de palabra es de 64 bits.

El mapa de memoria de este PC tiene dos secciones:

- *el área conocida como memoria convencional (los primeros 640 Kbytes de memoria), utilizada principalmente por el sistema operativo y programas residentes, y*
- *el área entre de memoria superior (situada a continuación, hasta ocupar 1 Mbyte), reservada para adaptadores hardware y la ROM BIOS, entre otros.*

*Grosso modo puede decirse que los primeros 1024 Kbytes de este computador (**128 Kpalabras x 64 bits**) se distribuyen de la siguiente manera:*

- *640 Kbytes de memoria RAM (**80 Kpalabras x 64 bits**)*
- *384 Kbytes de memoria ROM (**48 Kpalabras x 64 bits**)*

Sabiendo que se dispone de módulos de memoria de las siguientes características:

Módulos RAM
2 Kpalabras x 16
32 Kpalabras x 32
64 Kpalabras x 16

Módulos ROM
16 Kpalabras x 8
16 Kpalabras x 16
64 Kpalabras x 1

a. *¿Se podría diseñar un mapa de memoria con el tipo de pastillas disponibles? En el caso de que no sea posible, proponga otra opción **(0,1 puntos)**.*

   El bus de direcciones tiene 32 líneas, lo cual permite direccionar hasta 2^{32} = 4Gpalabras de memoria. En el caso propuesto se trata de realizar una memoria de 128Kpalabras, para lo cual sólo son necesarias 17 líneas, ya que 2^{17} = 128 Kpalabras.

   Por otro lado, también hay que comprobar que el bus de datos proporciona el ancho de palabra suficiente. El bus de datos tiene 64 bits de ancho de palabra y la palabra de memoria es de 64 bits.

Por tanto,

Sí, es posible diseñar dicho mapa de memoria.

*b. Indicar cuántos módulos de memoria y de qué características serían necesarios para diseñar el mapa de memoria, utilizando el menor número de pastillas posible **(0,1 puntos)**.*

- Para la memoria RAM, se dispone de los siguientes módulos de memoria:
  - 2 Kpalabras x 16
  - 32 Kpalabras x 32
  - 64 Kpalabras x 16

Realizando los siguientes cálculos:

$$\frac{80\ Kpalabras}{2\ Kpalabras} x \frac{64}{16} = 40\ x\ 4 = 160 \text{ módulos de memoria}$$

$$\mathbf{\frac{80\ Kpalabras}{32\ Kpalabras} x \frac{64}{32} = 2{,}5\ x\ 2 = 3\ x\ 2 = 6 \text{ módulos de memoria}}$$

$$\frac{80\ Kpalabras}{64\ Kpalabras} x \frac{64}{16} = 1{,}25\ x\ 4 = 2\ x\ 4 = 8 \text{ módulos de memoria}$$

- Para la memoria ROM, se dispone de los siguientes módulos de memoria:
  - 16 Kpalabras x 8
  - 16 Kpalabras x 16
  - 64 Kpalabras x 1

Realizando los siguientes cálculos:

$$\frac{48\ Kpalabras}{16\ Kpalabras} x \frac{64}{8} = 3\ x\ 8 = 24 \text{ módulos de memoria}$$

$$\mathbf{\frac{48\ Kpalabras}{16\ Kpalabras} x \frac{64}{16} = 3\ x\ 4 = 12 \text{ módulos de memoria}}$$

$$\frac{48\ Kpalabras}{64\ Kpalabras} x \frac{64}{1} = 0{,}75\ x\ 64 = 1\ x\ 64 = 64 \text{ módulos de memoria}$$

Utilizando el menor número de módulos posibles, el mapa de memoria quedaría definido por las siguientes pastillas de memoria:

6 RAM de 32 Kpalabras x 32, distribuidos en 3 filas y 2 columnas.

12 ROM de 16 Kpalabras x 16, distribuidos en 3 filas y 4 columnas.

*c. Realizar el diseño del mapa de memoria del modo que indica la siguiente figura **(1 punto)**:*

Mapa de memoria RAM	*0* *80K-1*
Mapa de memoria ROM	*80K* *128K-1*

Para diseñar el mapa de memoria, es preciso averiguar el número de líneas que se necesitan para direccionar cada pastilla de memoria:

- Módulos RAM de 32 Kpalabras x 32:

  32 Kpalabras $\leq 2^n$ ➔ para direccionar 32 Kpalabras se necesitan n=15 bits, es decir, las líneas $A_0..A_{14}$, para cada una de las pastillas de memoria RAM.

  Las pastillas de la última fila de memoria RAM no se utilizarán completamente, sino que sólo será preciso utilizar las primeras 16 Kpalabras de dichos módulos de memoria.

- Módulos ROM de 16 Kpalabras x 16:

  16 Kpalabras $\leq 2^n$ ➔ para direccionar 16 Kpalabras se necesitan n=14 bits, es decir, las líneas $A_0..A_{13}$, para cada una de las pastillas de memoria ROM.

  En este caso se utilizan completamente todos los módulos de memoria.

Además, por el apartado a), sabemos que se necesitan 17 líneas para direccionar el primer Mbyte del mapa de memoria propuesto. Por tanto, el mapa de memoria queda de la siguiente manera:

A_{31} .. A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	...	A_0			
0 ... 0	0	0	0	0	...	0	Mínima dirección	0 K	**1ª fila RAM**
			1	1	...	1	Máxima dirección	32 K-1	
0 ... 0	0	1	0	0	...	0	Mínima dirección	32 K	**2ª fila de RAM**
			1	1	...	1	Máxima dirección	64 K-1	
0 ... 0	1	0	0	0	...	0	Mínima dirección	64 K	**3ª fila de RAM**
			0	1	...	1	Máxima dirección	80 K-1	
Zona no direccinable de módulos RAM								80K 96 K-1	
0 ... 0	1	0	1	0	...	0	Mínima dirección	80 K	**1ª fila de ROM**
				1	...	1	Máxima dirección	96 K-1	
0 ... 0	1	1	0	0	...	0	Mínima dirección	96 K	**2ª fila de ROM**
				1	...	1	Máxima dirección	112 K-1	
0 ... 0	1	1	1	0	...	0	Mínima dirección	112 K	**3ª fila de ROM**
				1	...	1	Máxima dirección	128 K-1	

*d. Dibujar el esquema del mapa de memoria resultante **(0,8 puntos)**.*

A continuación se muestra un mapa de memoria simplificado, puesto que se ha utilizado un solo decodificador, que permite seleccionar únicamente la zona de memoria requerida en el problema. Además, por simplicidad, se han eliminado las líneas de lectura/escritura para la memoria RAM y la línea de lectura para la memoria ROM.