

Todas las preguntas tienen la misma puntuación, salvo indicación expresa de lo contrario. Las respuestas erróneas no restan puntos.

- ☐ Se tiene una serie de códigos binarios, que son respectivamente las codificaciones de:

- A) 130 (natural)
B) -15 (complemento a 2)
C) -15 (signo-magnitud)
D) -15 (exceso a 127)

Si un algoritmo intenta ordenar de menor a mayor estos códigos sin saber qué significan, simplemente comparándolos como si fueran naturales, ¿qué ordenación produciría? Ejemplo de respuesta: B) A) C) D)

D) A) C) B)

- ☐ Se diseña un formato de coma flotante análogo al del estándar IEEE-754, pero que usa sólo 16 bits en lugar de 32. Estos bits se distribuyen en la siguiente forma: 1 para el signo, 6 para el exponente y 9 para la mantisa. Al igual que en IEEE-764 se reservan los casos “todo ceros” y “todo unos” del exponente para números “denormales” y “especiales”. En los restantes casos, el exponente se codifica en exceso a Z central menos uno.

— ¿Qué código tendría el número $-5,25$ en el formato anterior? Responde en hexadecimal.

C2A0

— ¿Cuál es el número real más grande representable en este formato? (Puedes dejar la respuesta indicada en forma de fórmula que involucre potencias de dos). ¿Qué código hexadecimal corresponde a ese número?

Valor: $2^{31} - 2^{22}$

Código: 7DFFh

- ☐ Un computador que maneja datos de 12 bits se emplea para representar números en coma fija. Si se sabe que el mayor número representable en el formato de coma fija elegido es 127,9375, ¿cuál es el menor número representable en ese formato? Contestar en decimal.

-128

- ☐ Dada la siguiente secuencia de señales de control:

Paso	Acciones de Control
1	ALU_TMPS, TMPE_CLR, ADD, PC-IB, READ, IB-MAR
2	TMPS-IB, IB-PC
3	MDR-IB, IB-IR
4	IRI-IBI, IBI-R2I, FIN

— ¿A qué instrucción corresponden dichas señales de control?

MOVL R2, ?

— ¿Qué señal falta en el paso 1?

CarryIn

— Si en el paso 4 el valor que aparece en el bus interno (IB) es 0080h ¿Cuál ha sido el valor que apareció en el paso 3?

2280h

- ☐ Se tiene un computador (no necesariamente el Computador Elemental de teoría), y un módulo de memoria que cubre el 50 % de su Espacio de Direcciones. Dicho módulo está compuesto por 4 bancos, y en cada banco hay 8 chips.

— ¿Cuántas líneas de dirección llegarán al Circuito de Activación (CA) de dicho módulo?

1 línea

— ¿Qué tipo (tamaño) de decodificador tendrá el módulo en su interior? (ejemplo de respuesta: 1 : 2)

2:4

— Si cada chip tiene una capacidad de $4K \times 2$ ¿De qué tamaño es el espacio de direcciones de esta CPU? Ejemplo de respuesta: $16K \times 8$

$32K \times 16$

- ☐ Se ha escrito un programa que tiene por cometido copiar un texto que está almacenado en una dirección de la memoria a otra dirección de memoria, pero “dándole la vuelta” durante la copia. Es decir, si el texto original era por ejemplo “HOLA”, el resultado una vez copiado será “ALOH”. La cadena origen llevará un caracter nulo (ASCII 0) al final para indicar dónde termina.

La “copia hacia atrás” la realiza un procedimiento llamado AL_REVES. Este procedimiento recibe dos parámetros que el procedimiento principal debe apilar antes de llamarle y que son:

- La dirección de memoria donde está la cadena a copiar.
- La dirección de memoria donde debe dejarse la cadena copiada “al revés”.

El procedimiento simplemente prepara dos registros, uno de ellos apuntando al final de la cadena que hay que copiar y el otro apuntando al principio de la dirección destino, y va haciendo la copia mientras decreuenta el primer puntero e incrementa el segundo.

Para saber en qué dirección termina la cadena origen, hace uso de otro procedimiento llamado CUANTAS_LETRAS. Este procedimiento simplemente recibe la dirección de una cadena y retorna en el registro R0 cuántas letras tiene esa cadena. Por ejemplo, ante la cadena "HOLA", 0 retornaría 4 (ya que el terminador no lo cuenta). El código fuente de este procedimiento, aunque es trivial, no se muestra. Una vez conocido el número de letras que tiene el texto origen, se suma este número menos 1 a la dirección en que empieza el texto, y así se tiene la dirección de la última letra del mismo.

Seguidamente se muestra parte del código de dicho programa, en el que se han tapado algunas líneas por cuyo contenido se preguntará más adelante.

```

1  ORIGEN █████
2  INICIO start
3  .PILA █████
4  .DATOS
5  texto  VALOR "Prueba",0
6  result VALOR " ",0
7  .CODIGO
8  PROCEDIMIENTO CUANTAS_LETRAS
9  ; El código de este procedimiento se omite
10 ; para mayor brevedad. Véase el enunciado.
11 FINP
12
13 PROCEDIMIENTO AL_REVES
14  PUSH R6
15  MOV  R6, R7
16  PUSH R0
17  PUSH R1
18  PUSH R2
19  PUSH R3
20
```

```
21 INC R6 ; Recuperar parámetros
22 INC R6
23 MOV [R6], [R6]
24 INC R6
25 MOV [R6], [R6]
26 ; Calcular longitud de cadena origen
27 PUSH R2
28 CALL CUANTAS_LETRAS
29 INC R7
30 ; En R0 tenemos el número de letras de la cadena
31 ; Hacer apuntar R2 a la última letra
32 [R6]
33
34
35 ; Hacer la copia
36 ; (repetir tantas veces como indique R0)
37 bucle:
38 MOV R3, [R2]
39 MOV [R1], R3
40 [R6]
41
42 DEC R0
43 BRNZ bucle
44
45 POP R3
46 POP R2
47 POP R1
48 POP R0
49 POP R6
50 RET
51 FINP
52
53 start:
54 ; Probamos el procedimiento con la cadena texto
55 MOVH R0, BYTEALTO DIRECCION texto
56 MOVL R0, BYTEBAJO DIRECCION texto
57 PUSH R0
58 MOVH R0, BYTEALTO DIRECCION result
59 MOVL R0, BYTEBAJO DIRECCION result
60 PUSH R0
61 CALL AL_REVES
62 [R6]
63
64 FIN
```

Cuando está a punto de ejecutarse la primera instrucción de la rutina AL_REVES, el simulador de la CPU nos permite averiguar los siguientes datos. Algunos de estos datos son necesarios para responder a las preguntas que siguen.

- R7 vale 0292h
- En la cima de la pila hay un 0273h
- Justo debajo de la cima de la pila hay un 0237h

— ¿Qué valor se utilizó en la directiva ORIGEN del programa?

0230

— ¿Qué valor se utilizó en la directiva PILA del programa?

0020

— Si la rutina CUANTAS_LETRAS, que no se muestra, empieza en la forma estándar, esto es, con PUSH R6 seguido de MOV R6, R7, ¿qué valor tomará el registro R6 tras el MOV R6, R7?

028Ah

— ¿Qué falta en las líneas 23 y 25 en las que se recuperan los parámetros?

Línea 23: R1

Línea 25: R2

— Al retorno de la función CUANTAS_LETRAS, el registro R0 contiene el número de letras de la cadena origen. Esta información se usa para actualizar el registro que apunta a esta cadena, de modo que pase a apuntar a la última letra de la misma. Esto lo hacen las instrucciones que faltan en la línea 32 y la siguiente ¿qué instrucciones serían éstas?

ADD R2, R2, R0
DEC R2

— La línea 40 y la siguiente actualizan los punteros a las cadenas. Escribe qué instrucciones se han ocultado en estas líneas.

INC R1
DEC R2

— Tras el CALL AL_REVES (línea 61) aparecen dos instrucciones ocultas ¿cuáles serían?

INC R7
INC R7

— En un instante la CPU está ejecutando el MOV de la línea 39 y en ese momento R3 vale 0072h ¿cuál es el valor en ese momento de R0?

2

— Si el valor más bajo que toma R7 durante la ejecución del programa es 0289h ¿Cuántas instrucciones PUSH contiene la rutina CUANTAS_LETRAS?

dos

— Si el procedimiento CUANTAS_LETRAS no mostrado contiene 18 instrucciones, codifica la instrucción CALL CUANTAS_LETRAS de la línea 28. Escribe la respuesta en hexadecimal.

D0E1

- ❑ La CPU acaba de finalizar el paso 3 de una instrucción y se dispone a iniciar el paso 4. En ese instante los valores de algunos de los registros de la CPU y de algunas de las direcciones de la memoria son los que se muestran en las tablas siguientes:

		DIR.	CONT.
R7=1100h PC=20F1h MAR=20F0h MDR=B800h		10FE	7124
		10FF	AB38
		1100	3951
		1101	0000
		1102	5720

— ¿Cuál será el valor que tendrá PC durante el paso 1 de la siguiente instrucción que se ejecute? Responder en hexadecimal.

3951h

— Si la próxima instrucción que se ejecuta es un salto condicional BRZ, ¿cuántos pasos de ejecución tomará la ejecución de este salto?

4

- ❑ Usando la CPU elemental se diseña un computador, al cual se le incorpora un interfaz dedicado a la comunicación entre dicho computador y un periférico de tipo impresora. El interfaz contiene dos registros:

Registro de control. La impresora pone un 1 en este registro para indicar a la CPU que está lista para recibir un nuevo dato. Este registro se sitúa en la posición E800h del espacio de direcciones de la CPU.

Registro de datos. En él la CPU pone los datos que se envían a la impresora. Se mapea en la dirección E801h.

En el listado siguiente se representa un fragmento de un programa cuyo cometido es enviar un dato almacenado en una posición de memoria a la impresora. Adicionalmente, en se proporciona el estado de los registros y del PC justo cuando va a comenzar el primer ciclo de ejecución de la primera instrucción del fragmento de código dado.

```
1  ; Esperar hasta que la impresora esté lista
2  otra_vez: MOV  R1, [R4]
3             COMP R1, R0
4                             
5             ; Enviar dato a la impresora
6             MOV  R1, [R3]
7                             
```

R0=0001	R3=22A1	R6=0000
R1=0000	R4=E800	R7=1300
R2=0000	R5=E801	PC=12B4

A partir de los valores de los registros y de la interpretación que tú hagas del programa, contestar a las siguientes preguntas:

- ¿Qué instrucción es necesaria en el hueco del código correspondiente a la línea 4?

BRNZ otra_vez

- ¿Qué instrucción es necesaria en el hueco del código correspondiente a la línea 7?

MOV [R5], R1

- ¿En qué dirección de memoria se encuentra el dato que se envía a la impresora?

22A1h