





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

☐ Se ha desarrollado un programa para la CPU teórica que incluye dos procedimientos:

- **Multiplica:** recibe por la pila dos factores y devuelve en R0 su multiplicación interpretándolos como números naturales.
- **PotenciaMem:** recibe por la pila (en este orden) un número que interpreta como base, otro que interpreta como potencia y una dirección de memoria. Calcula el resultado de elevar la base a la potencia y lo guarda en la dirección de memoria. El procedimiento supone todos los números naturales.

El código del programa se muestra a continuación:

```
1  ORIGIN 2000h
2  INICIO main
3  .PILA 20h
4
5  .DATOS
6  resultado1 VALOR 0
7  resultado2 VALOR 0
8
9  .CODIGO
10 PROCEDIMIENTO Multiplica
11     PUSH R6
12     MOV R6, R7
13     PUSH R1
14     PUSH R2
15
16     INC R6
17     INC R6
18     MOV R1, [R6]
19     INC R6
20     MOV R2, [R6]
21
22     ; Si algún factor es 0, devolver 0
23     XOR R0, R0, R0
24     
25     BRZ salirMultiplica
26     
27     BRZ salirMultiplica
28
29     bucleMultiplica:
30     ADD R0, R0, R1
31     DEC R2
32     BRNZ bucleMultiplica
33
34     salirMultiplica:
35     POP R2
36     POP R1
37     POP R6
38     RET
39 FINP
40
```

```
41 PROCEDIMIENTO PotenciaMem
42     PUSH R6
43     MOV R6, R7
44     PUSH R0
45     PUSH R1
46     PUSH R2
47     PUSH R3
48
49     ; Extracción de parámetros
50     
51
52
53
54     MOV R1, [R6]
55     INC R6
56     MOV R2, [R6]
57
58     ; Si la base es 0, el resultado es 0
59     XOR R0, R0, R0
60     COMP R2, R0
61     BRZ salirPotencia
62
63     ; Si el exponente es 0, el resultado es 1
64     COMP R1, R0
65     BRNZ calcularPotencia
66     MOVL R0, 1
67     JMP salirPotencia
68
69     calcularPotencia:
70     MOV R0, R2 ; Incializar el resultado
71     DEC R1 ; Equivale a haber multiplicado una vez
72
73     buclePotencia:
74     PUSH R2
75     PUSH R0
76     CALL Multiplica
77     INC R7
78     INC R7
79     
80
81
82     salirPotencia:
83     MOV [R3], R0
84     POP R3
85     POP R2
86     POP R1
87     POP R0
88     POP R6
89     RET
90 FINP
91
92     main:
93     MOVH R0, 0
94     MOVL R0, 9
95     PUSH R0
96     MOVL R0, 5
97     PUSH R0
98     MOVL R1, BYTEBAJO DIRECCION resultado1
99     MOVH R1, BYTEALTO DIRECCION resultado1
100    PUSH R1
101    CALL PotenciaMem
102    INC R7
103    INC R7
104    INC R7
```

```
105
106    MOVL R0, 2
107    PUSH R0
108    MOVL R0, 3
109    PUSH R0
110    MOVL R1, BYTEBAJO DIRECCION resultado2
111    MOVH R1, BYTEALTO DIRECCION resultado2
112    PUSH R1
113    CALL PotenciaMem
114    INC R7
115    INC R7
116    INC R7
117    FIN
```

— ¿Cuáles son las instrucciones que faltan en las líneas 24 y 26?

— ¿Qué instrucción o instrucciones faltan en el hueco de la línea 50?

— ¿Qué instrucción o instrucciones faltan en el hueco de la línea 79?

— ¿Cuál sería el número más grande que se podría poner en la línea 108 sin que el programa calculase un resultado incorrecto? Responder en decimal.

— Se sabe que la primera vez que la CPU ejecutó una instrucción del programa con código F403h, tardó 1 ns. ¿Cuál es la frecuencia de la CPU? Incluir las unidades en la respuesta.

— ¿Qué valor hay en el bus interno en el paso 2 de la instrucción PUSH R6 del procedimiento Multiplica? Contestar en hexadecimal.

— Se sabe que la primera dirección de memoria que resulta modificada tras comenzar la ejecución del programa es la 2071h. ¿En qué dirección está la instrucción de la línea 116? Contestar en hexadecimal.

— La segunda vez que la instrucción de la línea 83 es ejecutada, ¿qué valor tienen los registros que aparecen en ella? Contestar en hexadecimal.

R0:	R3:
-----	-----

☐ ¿Cuál o cuáles de las siguientes afirmaciones acerca de la ALU de la CPU elemental son **verdaderas**?

- A) A partir del paso de ejecución 4 en adelante, la ALU sólo se es utilizada en las instrucciones aritmético lógicas, como su nombre indica.
- B) La ALU puede operar indistintamente con números naturales o enteros en signo-magnitud, produciendo resultados correctos en ambos casos.
- C) La ALU, además del resultado, genera cinco bits de estado (ZF, OF, CF, SF e IF).
- D) La ALU no tiene memoria, por lo que, en la CPU teórica, necesita registros auxiliares para su correcta operación.

☐ 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  $-2,5$  en el formato anterior? Responde en hexadecimal.

— Si en los números denormales de este formato el exponente implícito es -30, ¿cuál es el número más pequeño, es decir, el positivo más cercano a cero, representable en este formato? Dejar la respuesta indicada como sumas de potencias de dos. ¿Qué código (hexadecimal) tiene?

Valor:	Código:
--------	---------

— ¿Qué número real está representado por el código 0300h? Dejar la respuesta indicada como sumas de potencias de dos.

☐ El siguiente fragmento de código envía datos al dispositivo luces que está mapeado en la dirección F4C0h

```
1  MOVH R0, 0F4h
2  MOVL R0, 0C0h
3  MOVH R1, 00
4  MOVL R1, 03
5  bc:MOV [R0], R1
6  ADD R1, R1, R1
7  BRNC bc
```

— ¿Cuál es el máximo número de bombillas que están luciendo **simultáneamente** al ejecutar el programa?

— ¿Qué valor tendrá R1 cuando se salga del bucle? Contestar en hexadecimal.

☐ Para ilustrar el uso de las interrupciones en el computador elemental se ha escrito el programa que muestra el listado. En él, se muestra una rutina de interrupción que modifica el valor del registro R0 (haciendo que éste varíe entre 0 y 9), y un programa principal (que imprime en la esquina superior izquierda de la pantalla el valor del registro R0 continuamente).

```
1  ORIGIN 2000h
2  INICIO primera
3  .PILA 10h
4  .CODIGO
5
```

```
6  PROCEDIMIENTO rutinaTecla
7  PUSH R1
8  PUSH R2
9  ; se lee pulsación para limpiar buffer
10 MOVH R1, 
11 MOVH R1, 
12 MOV R2, [R1]
13 ; se incrementa R0 y se comprueba que
14 ; su valor está entre cero y nueve
15 INC R0
16 MOVL R1, 10
17 MOVH R1, 0
18 COMP R0, R1
19 BRNZ sigue
20 XOR R0, R0, R0
21 sigue:
22 POP R2
23 POP R1
24 IRET
25 FINP
26
27 primera:
28 ; se inicializan algunos registros
29 ; y se instala la rut. de interrupción
30 XOR R0, R0, R0
31 
32 
33 MOVL R2, 5
34 MOVH R2, 0
35 MOV [R2], R1
36 
37 
38 ; bucle de impresión infinito
39 MOVL R1, 0
40 MOVH R1, 0F0h
41 imprime:
42 MOVL R2, '0'
43 MOVH R2, 7h
44 ADD R2, R2, R0
45 MOV [R1], R2
46 jmp imprime
47 FIN
```

Sabiendo que el teclado está instalado en direcciones de memoria que están justo por encima de la pantalla, y que el simulador del computador elemental comenzará la ejecución del programa con las interrupciones inhabilitadas, contestar las siguientes preguntas:

— ¿Qué valor tomará el registro R1 tras ejecutar las instrucciones en las líneas 10 y 11? Contestar en hexadecimal.

— ¿Cuál es el vector de interrupción asociado al teclado?

- Si la pantalla tiene  $8 \times 15$  caracteres, ¿en qué dirección de memoria debe escribirse para que el valor de R0 aparezca en la esquina inferior izquierda en lugar de en la esquina superior izquierda de la misma? Contestar en hexadecimal.

- ¿Qué instrucciones faltan en las líneas 31 y 32?

- ¿Qué instrucción o instrucciones faltan en la línea 36?

- ☐ Se desea añadir una nueva instrucción para la CPU teórica con el mnemónico `SWAP R0, R1`. Esta instrucción debe intercambiar los valores de R0 y R1 sin modificar ningún otro registro que pueda afectar a los programas. Indicar la secuencia de señales de control necesarias para su ejecución en el menor número de pasos posibles. **[Esta pregunta vale doble]**

Paso	Señales
4	
5	
6	
7	

- ☐ Se quiere emplear la CPU teórica para trabajar con números reales. Puesto que se desea poder operar con ellos de forma sencilla, se opta por un formato de coma fija, del que deben elegirse el número de bits empleados para la parte entera y la parte decimal.

- Si se desea que el rango de números representables sea al menos  $[-8000, 8000]$ , ¿cuál es el mínimo número de bits que se necesitan para representar la parte entera?

- ¿Cuál es la precisión del formato elegido? Contestar en base 10.

- ☐ Se desea conectar un dispositivo de memoria para la CPU teórica, de modo que cubra el 12,5 % del espacio de direcciones. Este módulo se utilizará, entre otras cosas, para almacenar la tabla de vectores de interrupción. En el recuadro inferior dibuja cómo sería su circuito de activación, poniendo nombres a sus líneas de entrada (ej:  $a_{10}, a_{11}, \dots$ ). Escribe también cuál sería el rango de direcciones cubierto por este módulo.

Rango: