

A

Instrucciones generales para la realización de este examen

La respuesta debe escribirse en el hueco existente a continuación de cada pregunta **con letra clara**.

Cada respuesta correcta suma un punto Cada respuesta incorrecta, ilegible o vacía no suma ni resta. El total de puntos se dividirá entre el total de preguntas y se multiplicará por 10 para obtener la nota del examen.

— El valor 41BD0000h representa un nº real en formato IEEE-754, ¿a qué número decimal corresponde?

23,625

En la CPU elemental se han mapeado los siguientes dispositivos:

• Dispositivo RAM: en el rango C000-DFFF.

• Dispositivo ROM: en el rango 0800 – 0BFF.

• Dispositivo PROM: en el rango F110 – F117.

— ¿Cuál es la organización MxN del dispositivo ROM?

1Kx16

— El dispositivo de memoria RAM está formado por 128 chips organizados en 32 bancos, ¿Cuál es la organización MxN de cada chip?

256x4

— Se desea mapear un nuevo dispositivo de organización 512x16 a partir del dispositivo de memoria RAM. Dibuja su circuito de activación etiquetando correctamente las líneas.

$$a_{15} \cdot a_{14} \cdot a_{13} \cdot \sim a_{12} \cdot \sim a_{11} \cdot \sim a_{10} \cdot \sim a_9$$

— Se desea añadir al juego de instrucciones de la CPU teórica una nueva instrucción que permita decrementar en una unidad el contenido de un dato en memoria, DEC [Ri]. Completar el cuadro siguiente con las señales necesarias para que se ejecute, en el menor número de ciclos de reloj posible, la instrucción DEC [R5]

Pa so	Señales de Control
4	R5-IB, IB-MAR, LEER
5	(ciclo de espera)
6	MDR-IB, TMPE-SET, ADD, ALU-TMPS, ALU-SR
7	TMPS-IB, IB-MDR, ESCRIBIR
8	FIN

— Se sabe que el código que asigna ISO-Latin1 al carácter ‘a’ es 61h, al carácter ‘p’ el 112 (decimal) y al carácter ‘á’ el E1h. Codifica la cadena “papá” en UTF-8. Responder en hexadecimal.

706170C3A1h

— Determina cuál o cuáles de las siguientes afirmaciones son CIERTAS. En caso de que ninguna sea cierta, contesta “ninguna”.

A) Cuando la CPU elemental recibe una interrupción, tarda como mínimo tres pasos de ejecución en atenderla.

B) Al final de la rutina de servicio de una interrupción se tiene que ejecutar la instrucción STI para activar el flag de interrupción que se había puesto a cero durante la fase de aceptación.

C) En un formato coma fija con 3 bits para la parte entera y 2 para la parte fraccionaria la precisión que se obtiene es de 0,25.

D) Durante el proceso de atención a una interrupción, la CPU recibe un número de vector de interrupción a través del registro MDR

C, D

— Una ALU de cuatro bits se introducen el número A= −3 codificado en complemento a 2 y el numero B= −7 codificado en signo-magnitud. Las señales de entrada son: Resta=1, Cin=0, Op1=1 y Op0=1¿Cuáles son los valores de los bits del Registro de Estado después de realizarse la operación?

Z: 0 C: 1 O: 0 S: 1

Determinar cuáles de las siguientes afirmaciones son CIERTAS. Responder TODAS o NINGUNA si se considera que es la respuesta adecuada.

- A) La instrucción JMP Inm8 no modifica los flags del registro de estado.
- B) La instrucción que más rápido se ejecuta en la CPU elemental tarda 3 ciclos de reloj.
- C) La señal de control SUB no se activa en la instrucción NEG.
- D) En el formato IEEE-754 el menor número desnormalizados positivo distinto de cero es 2^{-127}

A

— Justo cuando acaba de terminar el paso 3 de una instrucción, los registros PC e IR contienen los siguientes valores:

PC = 0401h
IR = C0FAh

Determina los valores que aparecerán en el bus interno de la CPU en los tres pasos siguientes. Indica los valores en hexadecimal:

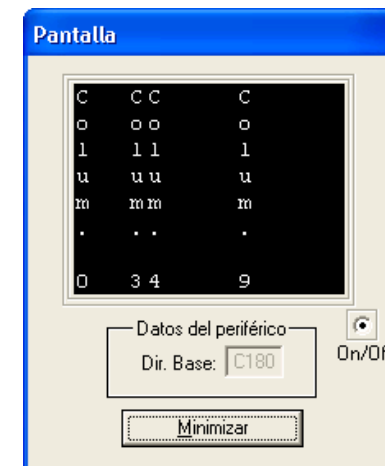
Paso 4: 0401 Paso 5: FFFA Paso 6: 03FB

El siguiente programa para la CPU teórica se encarga de escribir en la pantalla una cadena de texto en una de las 10 primeras columnas de las 15 que se compone la pantalla cuando se pulsa en el teclado un número de columna válido. Son válidas únicamente las teclas correspondientes a los números comprendidas entre 0 y 9.

El código incluido consta de un **programa principal** (cuyo comienzo está marcado con la etiqueta comienzo), de una rutina de servicio de interrupción del teclado (**rutina_teclado**) y de un procedimiento llamado **escribe_cadena**.

- El **programa principal** simplemente instala la rutina de servicio de interrupción de teclado en la tabla de vectores de interrupción.
- **rutina_teclado** lee la tecla pulsada y en el caso de la tecla pulsada sea válida muestra en la pantalla en la columna correspondiente a la tecla pulsada la cadena de texto almacenada en la variable cadena concatenada al código ascii de la tecla pulsada.
- El procedimiento **escribe_cadena** recibe dos parámetros a través de la pila. En el orden en que son apilados antes de la llamada al procedimiento, estos parámetros son: 1) el código ASCII de la tecla pulsada (por ejemplo el código ASCII del número 6 es 36h); 2) la dirección de la cadena a escribir

A continuación se muestra un ejemplo de salida de pantalla para dicho programa, y el código fuente del programa en ensamblador, al cual le faltan algunas instrucciones:



```
ORIGIN 2600h
INICIO main
.PILA XXX
.DATOS
    cadena VALOR "Colum. ",0

.CODIGO

PROCEDIMIENTO escribe_cadena
    PUSH R6
    MOV R6, R7
    PUSH R0
    PUSH R1
    PUSH R2
    PUSH R3
    PUSH R4
    PUSH R5

    ; Recuperamos en R0 la dirección de la cadena
    ; ---1---

    ; Recuperamos en R1 el ASCII de la columna
    INC R6
    MOV R1, [R6]

    ; Cargamos en R2 la posición de comienzo de la
    ; pantalla
    ; ---2---

    ; Obtener en R1 el número de la columna
    ; Usar R3 como registro auxiliar
```

<pre> MOVL R3, 30h MOVH R3, 0h SUB R1, R1, R3 ; Posicion de comienzo de la escritura ADD R2, R2, R1 ; R4 Registro auxiliar XOR R4, R4, R4 ; cargamos en R3 el número de columnas de la pantalla MOVH R3, 0 MOVL R3, 15 ; Escritura de la cadena escritura: MOV R5, [R0] ; ---3--- MOVH R5, 7 MOV [R2], R5 ; ---4--- JMP escritura final: MOV R1, [R6] MOVH R1, 7 MOV [R2], R1 POP R5 POP R4 POP R3 POP R2 POP R1 POP R0 POP R6 RET FINP PROCEDIMIENTO rutina_teclado PUSH R0 PUSH R1 PUSH R2 PUSH R3 MOVL R0, 47h MOVH R0, 0C1h MOVL R1, 0 ; R1 = mascara </pre>	<pre> MOVH R1, 1 bucle_rut: MOV R2, [R0] AND R2, R2, R1 BRZ no_hay_tecla DEC R0 ; R0 = registro datos MOV R2, [R0] ; R2 = tecla pulsada MOVH R2, 0 ; Eliminar cógigo scan ; comprobar tecla pulsada (Nº 0 - 9) ; ---5--- BRC bucle_rut MOVL R3, '9' COMP R2, R3 BRNC comprobarZF JMP correcto comprobarZF: BRNZ bucle_rut correcto: ; ---6--- PUSH R2 CALL escribe_cadena INC R7 INC R7 JMP bucle_rut no_hay_tecla: POP R3 POP R2 POP R1 ; ---7--- main: MOVH R0, 0 MOVL R0, 26 MOVH R1, BYTEALTO DIRECCION rutina_teclado MOVL R1, BYTEBAJO DIRECCION rutina_teclado MOV [R0], R1 STI JMP -1 FIN </pre>	<div> — ¿Qué instrucción o instrucciones falta/n en el hueco —1—? <div> INC R6 INC R6 MOV R0, [R6] </div> </div> <div> — ¿Qué tamaño mínimo debe tener la pila? Contestar en decimal <div> 16 </div> </div> <div> — ¿Qué instrucción o instrucciones falta/n en el hueco —2—? <div> MOVL R2, 80h MOVH R2, 0C1h </div> </div> <div> — ¿Qué instrucción o instrucciones falta/n en el hueco —3—? <div> COMP R5, R4 BRZ final </div> </div> <div> — ¿Qué instrucción o instrucciones falta/n en el hueco —4—? <div> INC R0 ADD R2, R2, R3 </div> </div>
--	---	--



— ¿Qué número de interrupción lleva asociado el teclado? **Contestar en hexadecimal**

001Aha

— Qué instrucción o instrucciones falta/n en el hueco —5—?

MOVL R3, '0'

MOVH R3, 0

COMP R2, R3

— Qué instrucción o instrucciones falta/n en el hueco —6—?

PUSH R2

MOVH R2, BYTEALTO DIRECCION cadena

MOVL R2, BYTEBAJO DIRECCION cadena

— ¿En qué posición de memoria se encuentra mapeado el registro de control de la pantalla? **Contestar en hexadecimal**

C1F8h

— Qué instrucción o instrucciones falta/n en el hueco —7—?

PUSH R0

IRET

FINP

Sin signo	Relación	Con signo
ZF = 1	=	ZF = 1
ZF = 0	≠	ZF = 0
CF=0 AND ZF=0	>	SF=OF AND ZF=0
CF = 0	≥	SF=OF
CF = 1	<	SF ≠ OF
CF=1 OR ZF =1	≤	SF≠OF OR ZF=1