

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 ecualizador es un dispositivo que procesa señales de audio, es el elemento que permite modificar la curva de respuesta en frecuencia de un sistema de audio. Los más habituales son los ecualizadores gráficos que presentan un conjunto de potenciómetros deslizantes, colocados de tal manera que permiten una visualización de la compensación realizada. Pueden tener un nº variable de controles para que cada uno controle un ancho de banda concreto del espectro audible. Casi todos los componentes multimedia actuales disponen de ecualizadores más o menos sofisticados. Vamos a utilizar la CPU teórica para desarrollar un sistema de visualización y control de un ecualizador.

Usaremos el periférico **Pantalla** para mostrar las barras del ecualizador gráfico. Tendremos 8 controles deslizantes que se corresponderán con las 8 filas de la pantalla. Cada control gestionará un trozo del ancho de banda audible. Cada control podrá tomar 15 posiciones discretas, correspondiéndose con las 15 columnas de las que dispone la pantalla. Para indicar la posición en la que se encuentra el control se dibujará en la fila correspondiente

a ese control tantas 'I' como sean necesarias.

Usaremos el periférico **Teclado** para interactuar con el ecualizador. Podremos desplazarnos por los diferentes controles usando las teclas 'q' y 'a' (la 'q' permitirá subir por los diferentes controles y la 'a' bajar) y usaremos las teclas 'o' y 'p' para decrementar o incrementar el control activo. Para identificar el control activo usaremos un **asterisco** rojo sobre fondo blanco para sustituir la última 'I' del control. En la siguiente figura se muestra el aspecto que presenta la pantalla del ecualizador.



El programa para gestionar este display contiene el siguiente código:

- **El programa principal:** Establece el vector de interrupción 3 asociado al periférico **Teclado** y su rutina.
- **rutina\_PulsacionTecla:** Rutina que se

ejecutará cuando el periférico **Teclado** solicite una interrupción. En ella se analizará si la tecla pulsada es alguna de las de función 'q,a,o,p' y reaccionará ante ellas de la manera descrita anteriormente para interactuar con el ecualizador. Esta rutina usará una serie de procedimientos para facilitar la tarea.

- **DibujaEcualizador:** Procedimiento que leerá de memoria principal el estado del ecualizador desde la variable **barras\_ecualizador**, vector de 8 posiciones en el que se almacena la posición de cada uno de los 8 controles de que disponemos y dibujará una representación en pantalla de esos valores.
- **DibujaCursor:** Procedimiento que leerá de memoria la posición actual del cursor desde la variable **posicion\_cursor** y dibujará un asterisco en la fila que indica el control activo, usará igual que el procedimiento anterior la variable **barras\_ecualizador**.
- **LeeVector:** Procedimiento que permite leer el valor de un vector en una determinada posición. Se pasarán a través de la pila y en el orden indicado: **dirección del vector a leer** y **posición del vector a leer**. El valor leído será devuelto usando el registro **R5**. Los índices del vector comienzan a numerarse desde 1.
- **EscribeVector:** Procedimiento que permite escribir un valor en una posición determinada de un vector. Se pasarán a través de la pila y en el orden indicado: **dirección del vector**, **posición del vector a escribir** y **el valor a escribir**. Los índices del vector comienzan a numerarse desde 1.



```

        JMP continua

Es_una_o:

; debemos decrementar la barra de ecualización
;seleccionada
        XOR R2,R2,R2
        MOVL R2,1

; leemos en R5 el valor de la barra de
; ecualización seleccionada
        ----- HUECO 5 -----

        PUSH R6 ;vector a leer
        PUSH R4 ;posicion a leer
        CALL LeeVector
        INC R7
        INC R7
        COMP R2,R5
        BRZ cursor_abajo
;decrementamos la barra de ecualizacion
        DEC R5
        JMP continua_op

cursor_abajo:
        MOVL R5,15
        JMP continua_op

;;;;;;;;;;;;;;;;;;;;;;;;;;
; CÓDIGO OMITIDO INTENCIONADAMENTE
;
; código similar para actuar ante la tecla pulsada
;
; ...
;;;;;;;;;;;;;;;;;;;;;;;;;;

continua_op:

;actualizamos en memoria la barra de ecualización
        PUSH R6 ; vector a escribir
        PUSH R4 ; posicion a escribir
        PUSH R5 ; valor a escribir
        CALL EscribeVector
        INC R7
        INC R7
        INC R7

continua_qa:
; actualizamos en memoria el valor actual del
;cursor
        MOV [R3],R4

;borramos la pantalla
        MOVB R2,0F0h
        MOVL R2,78h
        XOR R5,R5,R5
        MOVL R5,3
        MOV [R2],R5

```

```

CALL DibujaEcuallizador
CALL DibujaCursor

continua: ;no pulso ninguna tecla de función
POP R6
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0

----- HUECO 6 -----

IRET
FINP

PROCEDIMIENTO DibujaEcuallizador

    PUSH R0 ; R0 dirección pantalla
    PUSH R1 ; R1 dirección vector barra ecuallización
    PUSH R2 ; R2 contador de barras ecuallizador
    PUSH R5 ; R5 aux función leer_vector
    PUSH R3 ; R3 aux
    PUSH R4 ; R4 contiene el carácter 'I'

    ;En R0 dirección de p. pantalla
    MOVL R0,00
    MOVH R0,0F0h

;avanzamos el puntero de pantalla hasta su última
;fila ya que pintaremos desde abajo hacia arriba
    XOR R3,R3,R3
    MOVL R3,69h ;7x15=105= 69h
    ADD R0,R0,R3

    ;En R1 dirección barras_ecuallizador
    MOVL R1,BYTEBAJO DIRECCION barras_ecuallizador
    MOVH R1,BYTEALTO DIRECCION barras_ecuallizador

    ;En R2 n° filas de pantalla
    XOR R2,R2,R2
    MOVL R2,8

; R4 carácter I que actuará como línea de la
;columna del ecuallizador
    MOVH R4, 07h
    MOVL R4,49h

    ;Pintamos todas las barras del ecuallizador
Continua_SiguienteBarra:

    PUSH R1 ; vector a leer
    PUSH R2 ; posicion a leer
    CALL LeeVector
    INC R7
    INC R7

;En R5 tenemos el valor de la barra de
; ecuallización seleccionada

```





- ¿Qué instrucción o instrucciones faltan en **HUECO 4** sabiendo que el periférico teclado está mapeado justo a continuación de la pantalla?

MOVL R1,80h

MOVH R1,0F0h

MOV R0,[R1]

- ¿Qué instrucción o instrucciones faltan en **HUECO 5**?

PUSH R6 ;vector a leer

PUSH R4 ;posicion a leer

CALL LeeVector

INC R7

INC R7

- ¿Qué instrucción o instrucciones faltan en **HUECO 6**?

IRET

FINP

INC R7



- ¿Qué instrucción o instrucciones faltan en **HUECO 7**? (Usar el registro R3 como registro auxiliar)

XOR R3,R3,R3

MOVL R3,0Fh ;1x15=15= Fh

SUB R0,R0,R3

- ¿Qué instrucción o instrucciones faltan en **HUECO 8**?

BRNZ Continua\_SiguienteBarra

- Con el código mostrado y con la información que se da en las partes omitidas ¿cuál es el tamaño mínimo de pila que debemos reservar en la directiva .PILA para que el programa funcione correctamente? Se supone que la limpieza de parámetros de pila se hace siempre con INC(Expresar en decimal)

24

- ¿Cuál es el código scan y el código ascii de la tecla 'p' (minúscula)? Expresar en decimal

Scan: 13h

ASCII: 70h

- ¿Cuál o cuáles de las siguientes afirmaciones son **CIERTAS** (puedes responder “ninguna” o “todas” si así lo consideras)?

- A) La instrucción MOV [R0], [R1] no se puede ejecutar en la CPU elemental debido a que ésta sólo tiene un bus de datos y en este caso necesitaría dos.
- B) En el formato de representación de números enteros en exceso a Z, a medida que aumenta el valor de Z aumenta la precisión de la representación.
- C) El ancho de una palabra de un dispositivo de memoria es igual al ancho del bus de datos del computador.
- D) El registro R7, al inicio de la ejecución de un programa en la CPU elemental, *apunta* a la primera posición de memoria de la sección de pila del programa.

C

Se desea mapear un dispositivo de memoria en una CPU basada en la CPU elemental. El rango elegido es el 8000-FFFFh.

- ¿Cuál es la ecuación lógica que deberá implementar el circuito de activación? Ejemplo de respuesta:  $\sim a_{24} \wedge a_{20}$

EA=a<sub>15</sub>

- ¿Cuántos chips de organización 1024x4 serán necesarios para construir el dispositivo anterior?

128

- Completa las señales de control necesarias en los pasos mostrados para que la CPU elemental sea capaz de ejecutar la instrucción CHANGE R0, R1. Esta instrucción intercambia el contenido de los registros R0 y R1, sin utilizar ningún otro registro de propósito general.

4	R0-MDR
5	R1-IB, IB-R0
5	MDR-IB, IB-R1, FIN

- Codifica el número real ( $2^{-95} + 2^{-98}$ ) en formato IEEE-754 precisión simple. Responder en hexadecimal.

10100000h

- ¿Cuál es el mínimo número de bits que es necesario utilizar en un formato de coma fija con signo para conseguir representar tanto el número 16.0 como el número 0.25 sin errores de representación?

8

Se ha realizado el siguiente programa en lenguaje ensamblador de la CPU elemental sin ningún cometido en particular. Se sabe que durante la ejecución del programa sólo se realizan dos peticiones de interrupción, justo en el momento en el que la CPU se encuentra ejecutando por primera vez la instrucción DEC R2.

```

ORIGEN 1000h
INICIO ini
.PILA 10h
.CODIGO

PROCEDIMIENTO RutInt
    IRET
FINP

ini:
    XOR R0, R0, R0
    MOVL R1, BYTEBAJO DIRECCION RutInt
    MOVH R1, BYTEALTO DIRECCION RutInt
    MOVL R2, 0
    MOVH R2, 1

bucle:
    MOV [R0], R1
    INC R0
    DEC R2
    BRNZ bucle

    STI
    JMP -1

FIN
    
```

- ¿Cuáles serán las siguientes DOS instrucciones que se ejecutarán tras la ejecución de la instrucción STI? Responder con los mnemónicos.

IRET, JMP -1

- ¿Cuál será la dirección de retorno de la rutina de interrupción? Responder en hexadecimal.

100Bh

- ¿Qué valor habrá almacenado en la dirección de memoria 00FFh al final de la ejecución del programa?

1000h

- ¿Cuál es el mínimo tamaño que podría tener la pila para que el programa siga funcionando correctamente?

2