



**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 siguiente programa para la CPU teórica dibuja un asterisco en pantalla y lo mueve hacia a la izquierda cuando se pulsa la tecla ‘o’ y hacia la derecha cuando se pulsa la ‘p’. Para ello utiliza tres elementos:

- La variable **pos**: Indica la posición actual donde está el asterisco.
- El procedimiento **mueve\_asterisco**: Recibe por la pila un valor que indica el incremento que hay sumarle a **pos** para volver a pintar el asterisco. Antes de pintarlo, lo borra de la posición anterior.
- La rutina de servicio **rut\_teclado**: Comprueba qué teclas se han pulsado. Si se pulsa una ‘o’, llama a **mueve\_asterisco** con parámetro -1; si se pulsa una ‘p’, lo llama con parámetro 1; si es otra tecla no hace nada.

```
ORIGEN 400h
INICIO main
.PILA XXXXh ; Eliminado intencionadamente
.DATOS
    pos VALOR 0F000h
.CODIGO

PROCEDIMIENTO mueve_asterisco
    PUSH R6
    MOV R6, R7
    PUSH R0
    PUSH R1
    PUSH R2

    ; Recuperar en R0 el parámetro
    INC R6
    INC R6
    MOV R0, [R6]

    ; Recuperar posición actual en R1
    ; -- 1 --
    MOV R1, [R6] ; R1 = pos actual
```

```
; Escribir en blanco en la pos actual
MOVL R2, ' '
MOVH R2, 7
MOV [R1], R2

; Modificar pos y escribir el asterisco
ADD R1, R1, R0
MOVL R2, '*'
MOVH R2, 7
MOV [R1], R2
MOV [R6], R1 ; Actualizar pos

POP R2
POP R1
POP R0
POP R6
RET
FINP

PROCEDIMIENTO rut_teclado
    PUSH R0
    PUSH R1
    PUSH R2
    PUSH R3
    PUSH R6

    MOVL R1, 1h ; R1 = reg. control tecl.
    MOVH R1, 0FEh
    MOVL R2, 0 ; Hacer R2 = máscara
    MOVH R2, 1

bucle: ; Hacerlo mientras queden teclas
    MOV R3, [R1]
    AND R3, R3, R2
    BRZ no_hay_tecla

    DEC R1 ; R1 = reg. de datos
    MOV R3, [R1] ; R3 = tecla pulsada
    INC R1 ; R1 = reg. de control

    ; Comprobar la tecla pulsada
    MOVL R6, 'o'
    MOVH R6, 12h ; scan de la 'o'
    COMP R6, R3
    BRNZ no_es_o
    XOR R6, R6, R6
    DEC R6 ; Poner un -1 en R6
    PUSH R6
    CALL mueve_asterisco
    INC R7
    JMP bucle

no_es_o:
```

```
MOVL R6, 'p'
MOVH R6, 13h ; scan de la 'p'
COMP R6, R3
; --2-- Saltar si es necesario
MOVL R6, 1
MOVH R6, 0
PUSH R6
CALL mueve_asterisco
INC R7
JMP bucle

no_hay_tecla:
    POP R6
    POP R3
    POP R2
    POP R1
    POP R0
    ; --3--
FINP

main:
    XOR R0, R0, R0
    INC R0
    MOVL R1, BYTEBAJO DIRECCION rut_teclado
    MOVH R1, BYTEALTO DIRECCION rut_teclado
    MOV [R0], R1
    STI

    JMP -1
FIN
```

La figura de la página siguiente muestra un momento de la ejecución del programa.

- Después de haber generado una interrupción y de recibir de la CPU la señal INTA, ¿cuál es el primer número que la interfaz del teclado envía por el bus de datos?

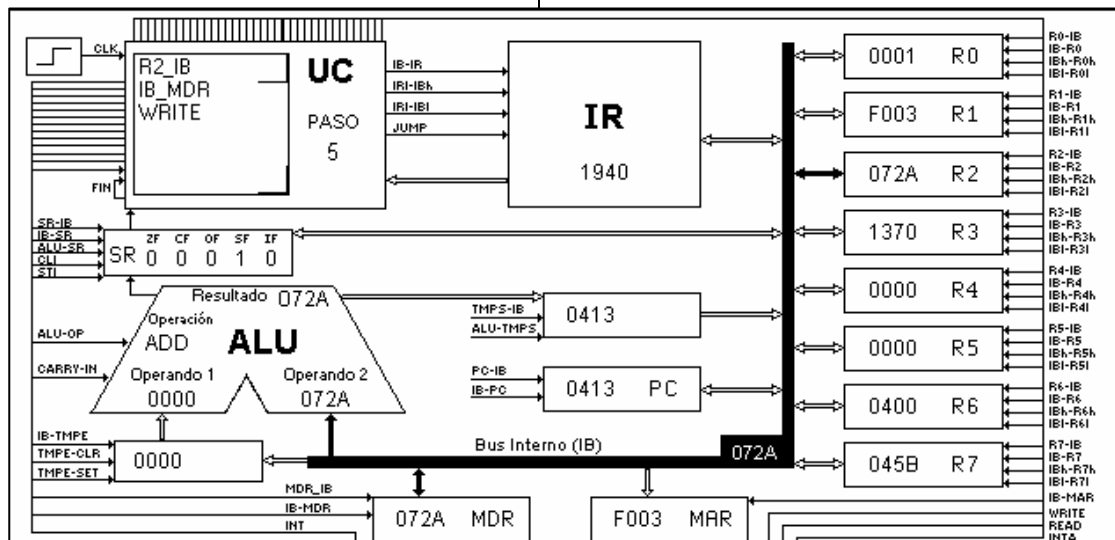
0001h

- ¿Cuál es el código ASCII de la ‘p’? **Contestar en hexadecimal.**

70h

- ¿Qué instrucción o instrucciones falta/n en -- 2 --?

BRNZ bucle



— ¿Qué instrucción o instrucciones falta/n en -- 1 --?

MOVL R6, BYTEBAJO DIRECCION pos

MOVH R6, BYTEALTO DIRECCION pos

— Sabiendo que no se ha pulsado ninguna vez la tecla 'o', ¿cuántas veces como mínimo se habrá pulsado la tecla 'p' antes de llegar al instante mostrado en la figura?

3

— ¿Qué instrucción o instrucciones falta/n en -- 3 --?

IRET

— ¿Qué tamaño mínimo debe tener la pila? **Contestar en hexadecimal.**

0Dh

— Rellena el espacio de direcciones con dispositivos de memoria para que el programa funcione correctamente. Dispones de dispositivos de tamaño 4K, 8K, 16K, 32K y 64K. El espacio de direcciones deberá estar tan lleno de dispositivos de memoria como sea posible y deberá ser usado el mínimo número posible de éstos. Ten en cuenta el rango de direcciones ocupado por las interfaces de los dispositivos de E/S (pantalla y teclado). **ATENCIÓN: Hay un hueco para usar hasta 6 dispositivos de memoria, pero no quiere decir que no se puedan usar menos. Como ayuda, se ofrece también el diálogo de configuración de memoria del simulador VonNeumann. Ejemplo de tamaño: 4K. Ejemplo de rango: 0000h-7FFFh.**

Disp. 1 → Tamaño: 32K Rango dir: 0000h-7FFFh

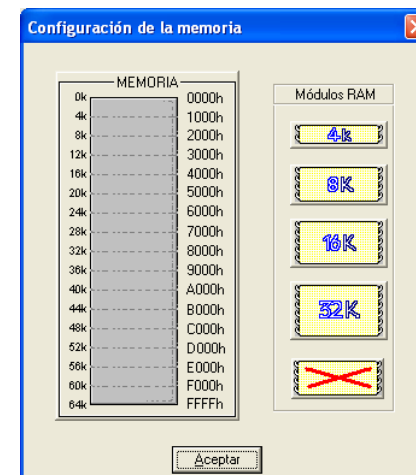
Disp. 2 → Tamaño: 16K Rango dir: 8000h-BFFFh

Disp. 3 → Tamaño: 8K Rango dir: C000h-DFFFh

Disp. 4 → Tamaño: 4K Rango dir: E000h-EFFFh

Disp. 5 → Tamaño: Rango dir:

Disp. 6 → Tamaño: Rango dir:



Se dispone de una CPU con doble número de líneas de datos e igual número de líneas de dirección que la CPU teórica. El espacio de direccionamiento se divide en 8 zonas iguales, numeradas del 0 (direcciones más bajas) al 7 (direcciones más altas). En las zonas 0 a 3 se instala memoria RAM, en las zonas 4 y 5 se instala memoria ROM, y en las zonas 6 y 7 nuevamente memoria RAM.

— ¿Qué rango de direcciones abarca la memoria ROM? Expresar el resultado en el formato XXXXh-YYYYh.

8000h-BFFFh

— ¿Cuántos chips de tamaño 4Kx8 serán necesarios para construir las dos zonas de memoria RAM?

48

— ¿Cuál es el menor salto entre dos números normalizados consecutivos en el formato de coma flotante IEEE-754? Responder en forma de potencias de dos.

$2^{-149}$

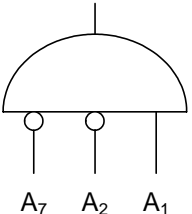
— ¿Cuáles de las siguientes afirmaciones son **CIERTAS**? (Puedes responder "ninguna" si así lo consideras.)

- a) La unidad de control de la CPU es un sistema digital secuencial.
- b) El número de direcciones a las que puede acceder la CPU elemental depende del número de líneas del bus de datos del sistema.
- c) El número de bancos en un dispositivo de memoria conectado al computador elemental depende del número de líneas del bus de datos.
- d) La CPU teórica accede a los dispositivos de entrada salida empleando las instrucciones MOV Rd, [Ri] y MOV [Ri], Rs.

a) y d)

Si se conecta un dispositivo a una CPU con un bus de direcciones de 20 líneas, y se ha visto que el dispositivo cubre el rango de direcciones C0000h-DFFFFh.

- Dibujar la puerta AND que debe actuar como circuito de activación del dispositivo. (En el ejemplo de respuesta se muestran tres líneas de dirección, pero puede ser cualquier otro número; indicar su peso y si entran negadas o no a la puerta.)

Ejemplo de respuesta	Responder
	<div>1</div>

Se pretende confeccionar una nueva instrucción para la CPU teórica, que permite efectuar la operación lógica NAND sobre dos operandos ubicados en memoria, dejando el resultado en un registro. El mnemónico de la nueva instrucción es:

**NAND Rd, [Ri<sub>1</sub>], [Ri<sub>2</sub>]** ; Rd ← [Ri<sub>1</sub>] NAND [Ri<sub>2</sub>]

La siguiente tabla muestra las señales de control necesarias en cada paso para implementar esta instrucción, tomando como ejemplo la instrucción NAND R0, [R1], [R2]:

4	R1-IB, IB-MAR, READ
5	Ciclo de espera
6	MDR-IB, IB-TMPE
7	R2-IB, IB-MAR, READ
8	Ciclo de espera
9	
10	TMPS-IB, IB-R0
11	R0-IB, TMPE-SET, XOR, ALU-SR, ALU-TMPS
12	TMPS-IB, IB-R0, FIN

— ¿Qué señales se activan en el paso 9?

MDR-IB, AND, ALU\_TMPS

Se sabe que las entradas de la ALU de 16 bits vista en clase tienen los siguientes valores:

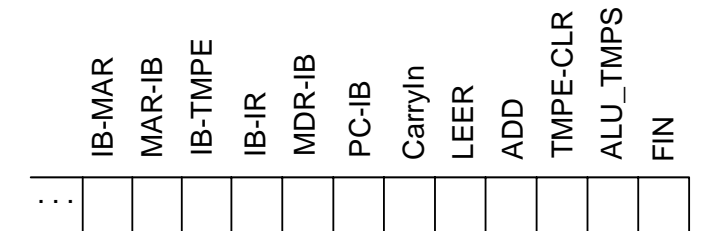
- En el **operando A** se encuentra la combinación de bits que representa el -3 en formato exceso a Z con 16 bits y exceso central.
- En el **operando B** se encuentran los 16 bits más altos de la codificación del número  $2^{107}(1+2^{-7})$  en formato IEEE-754.
- **CarryIn** = 0, **Resta** = 1, **Op1** = 1 y **Op0** = 1.

Se recuerda que las ALUs elementales que componen la ALU de 16 bits tienen conectadas las salidas de una puerta AND, de una puerta OR, de una puerta XOR y el resultado de un sumador elemental, a las entradas e<sub>0</sub>, e<sub>1</sub>, e<sub>2</sub> y e<sub>3</sub> del multiplexador, respectivamente.

- ¿Cuál es el resultado obtenido en la salida? Expresar el resultado con **cuatro dígitos hexadecimales**.

75FCh

Se dispone de una unidad de control microprogramada para la CPU teórica, cuyas palabras de control se interpretan como se indica en la siguiente figura (se muestran sólo los 12 bits menos significativos).



En un instante dado, se sabe que la señal IB-IR está activa.

- ¿Cuál es la palabra de control que está generando? Contestar con tres dígitos hexadecimales.

...0001 1000 0000 = 180h

Se lleva a cabo una modificación del formato de coma flotante IEEE-754 para representar números reales utilizando 16 bits. El nuevo formato y las reglas de representación utilizadas son idénticas a las del formato de 32 bits, pero la magnitud de la mantisa se recorta a 7 bits, en vez de los 23 utilizados en el formato original. Contesta a las siguientes preguntas relativas a este nuevo formato.

- Indica el error absoluto que se cometería al representar el número 625,625. Contesta en decimal.

1,625

- Indica qué número representa en este formato los cuatro dígitos hexadecimales 3EC0h. Contesta en decimal.

0.375

Se sabe que la CPU teórica emplea 5 milisegundos en ejecutar el siguiente fragmento de código:

```
XOR R3, R3, R2  
MOV R2, [R4]  
MOVL R3, FFh  
MOV R5, R2
```

— ¿Cuántos milisegundos dura un ciclo?

0,25

— ¿Cuáles de las siguientes afirmaciones son **CIERTAS**?  
(Puedes responder "ninguna" si así lo consideras.)

- a) El sistema UNICODE de representación de caracteres, al utilizar conjuntos distintos para distintas lenguas, permite la asignación del mismo código a distintos caracteres, siempre y cuando pertenezcan a conjuntos diferentes.
- b) Las CPUs con unidades de control microprogramadas reservan una parte de su espacio de direccionamiento en la memoria del sistema para almacenar la memoria de microprograma.
- c) La entrada/salida basada en muestreo periódico obliga a la CPU a derrochar tiempo en la consulta del estado de los periféricos.
- d) En la CPU teórica, el usuario, a través de un programa, NO puede cargar en el registro PC el valor que desee.

C