

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.

En la página siguiente se muestra el listado de un programa, denominado **Pruebas**, cuyo objetivo es probar el funcionamiento de un procedimiento denominado *scroll*. Dicho procedimiento sirve para hacer el scroll de cualquier pantalla conectada al computador elemental. La operación *scroll* desplaza el contenido de toda la pantalla una línea hacia arriba y rellena la línea inferior con espacios en blanco. El procedimiento *scroll* recibe como parámetro a través de la pila la dirección de comienzo de la memoria de vídeo. Este procedimiento se programa mediante un bucle que copia primero la fila 2 de la pantalla sobre la fila 1, después la fila 3 sobre la fila 2 y así sucesivamente hasta que se copia la fila 8 sobre la fila 7. Después mediante otro bucle (al que se hace referencia en el listado con la etiqueta *bucle_3*) se rellena la fila 8 con caracteres en blanco.

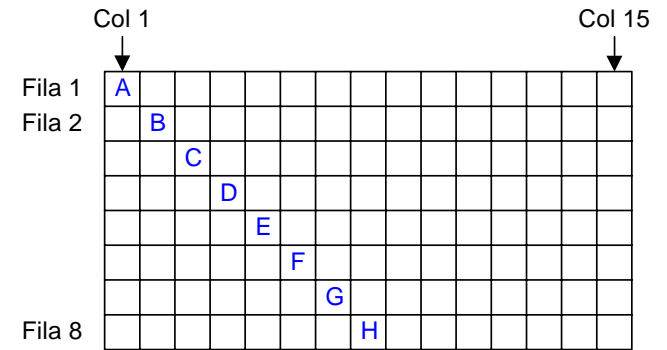
Para probar el procedimiento se diseña un programa principal que escribe un conjunto de caracteres en la pantalla y, después, llama al procedimiento *scroll*. Finalmente el programa queda detenido en un bucle infinito.

Información adicional
Dirección de comienzo de la memoria de vídeo = E000h
Código ASCII de la 'A' = 41h
Dirección de comienzo del procedimiento = 0313h
Número total de instrucciones del programa = 51 (decimal)

Teniendo en cuenta toda esta información y los comentarios presentes en el programa contesta a las siguientes preguntas:

— La figura siguiente representa la pantalla del computador elemental, que se organiza en 8 filas y 15 columnas.

Indica sobre la figura el estado que presenta la pantalla justo antes de la llamada al procedimiento *scroll* (indica sólo los caracteres que aparecen en las diferentes posiciones sin tener en cuenta el atributo). Se supone que justo antes de comenzar el programa la pantalla se encontraba totalmente en blanco.



— Indica el contenido de la dirección de memoria 302h. (Contesta con 4 dígitos hexadecimales)

2341h

— Indica las instrucciones necesarias en el recuadro 1. (Ten en cuenta el comentario del programa).

```
MOVL R0, 0
MOVH R0, 0E0h
PUSH R0
CALL scroll
```

— Codifica la instrucción *BRNZ bucle_2*, marcada en el listado con los caracteres '**←←←**'.

F5FAh

— Indica el mínimo valor que alcanzará el registro R7 durante la ejecución del programa

033Ch

— Escribe las instrucciones necesarias en el recuadro 2.

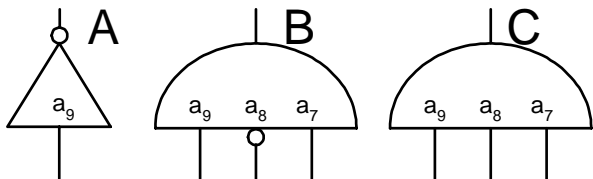
```
MOV [R3], R1
INC R3
COMP R3, R0
BRNZ bucle_3
```

— Cuando se ejecuta la instrucción *RET* del procedimiento se modifican los registros PC y R7. Determina los valores que tendrán estos registros justo después de completarse la ejecución de la instrucción.

PC = 0311h
R7 = 0342h

Listado programa Pruebas		
<pre> ORIGEN 300h INICIO ini .PILA 10h .DATOS .CODIGO ini: ; R0 contiene la dirección de la mem. de ; video MOVL R0, 0 MOVH R0, 0E0h *** ; R3 contiene la letra a imprimir y su ; atributo (blanco sobre negro = 7) MOVL R3, 'A' MOVH R3, 7 ; R2 contiene el contador del bucle MOVL R2, 8 MOVH R2, 0 ; R4 contiene desplazamiento de impresión MOVL R4, 16 MOVH R4, 0 ; Bucle de inicialización de pantalla bucle_1: MOV [R0], R3 ADD R0, R0, R4 INC R3 DEC R2 BRNZ bucle_1 ; *** Paso de parámetro y llamada al ; procedimiento scroll *** MOVL R0, 0 MOVH R0, 0E0h push R0 call scroll ; Eliminar parámetro de la pila INC R7 ; Terminar en bucle infinito para_siempre: JMP para_siempre PROCEDIMIENTO scroll PUSH R6 MOV R6, R7 PUSH R0 </pre>	<pre> PUSH R1 PUSH R3 PUSH R4 ; Cargamos el parámetro en R3 ; Contiene la dirección de fila 1 INC R6 INC R6 MOV R3, [R6] ; Cargamos R4 con la dirección de fila 2 MOVL R1, 15 MOVH R1, 0 ADD R4, R3, R1 ; Cargamos R0 con la dirección E078h ; (Dirección siguiente a la última de ; la pantalla) ; Se usa este valor para detectar el ; final de bucle_2 MOVL R0, 078h MOVH R0, 0E0h bucle_2: MOV R1, [R4] MOV [R3], R1 INC R3 INC R4 COMP R4, R0 BRNZ bucle_2 <<<< ; Rellenamos la última fila con el ; carácter blanco MOVL R1, ' ' MOVH R1, 7 ; atributo bucle_3: MOV [R3], R1 INC R3 COMP R3, R0 BRNZ bucle_3 POP R4 POP R3 POP R1 POP R0 POP R6 RET FINP FIN </pre>	<p>Imagina que el computador elemental que ejecuta el programa Pruebas tiene conectado un periférico. Dicho periférico tiene asociado el vector N° 5, pero no hay una rutina de interrupción disponible para atenderlo. Sin embargo, con objeto de que las posibles interrupciones generadas por el periférico no corrompan otros programas se le asigna una rutina vacía, constituida solo por la instrucción IRET. Supón que este periférico solicita una interrupción en algún momento durante la ejecución de la segunda instrucción del programa (marcada con *** en el listado) y que la CPU acepta la interrupción. Teniendo en cuenta esta situación contesta a las siguientes preguntas.</p> <p>— Valor de R7 justo antes de que comience el ciclo 1 de la instrucción IRET de la rutina vacía.</p> <div>0341h</div> <p>— Número de ciclos de CPU ejecutados desde el comienzo del programa hasta el instante indicado en la pregunta anterior. Contesta en decimal.</p> <div>18</div> <p>— Valor del PC justo después de ejecutar la instrucción IRET de la rutina vacía.</p> <div>0302h</div>

Se dispone de un computador con un bus de direcciones de 10 líneas y un bus de datos de 8 líneas. A este computador se encuentran conectados tres dispositivos de memoria, utilizando como circuitos de activación para dichos dispositivos las puertas que se muestran a continuación:



Teniendo en cuenta esta información contesta a las preguntas A, B y C:

- A) Si el dispositivo de memoria activado por la puerta C está constituido por 8 bancos de chips, indica el rango de direcciones del espacio de direcciones del computador que le correspondería al banco 5. (Formato de solución para el rango: XXXh – YYYh).

3D0h – 3DFh

- B) Se desea conectar al computador indicado un cuarto dispositivo de memoria que ocupará el 12,5 % del espacio de direcciones. Para ello se dispone de chips de memoria de 4 líneas de dirección y 4 de datos. ¿Cuántos chips serán necesarios para construir este dispositivo?

16

- C) Indica los dos rangos de direcciones en los que se podría ubicar el dispositivo de memoria al que se hace referencia en la pregunta anterior. (Formato de solución para los rangos: XXXh – YYYh)

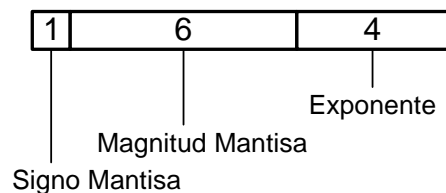
Posibilidad_1: 200h – 27Fh

Posibilidad_2: 300h – 37Fh

- Codifica el número $2^{-128} + 2^{-130} + 2^{-132}$ en el formato IEEE(P)754. Contesta con 8 dígitos hexadecimales.

002A0000

A continuación se muestra un formato de coma flotante con 7 bits para la mantisa (incluyendo el signo) y 4 para el exponente. La mantisa se codifica en signo-magnitud, todo fracción. El primer bit de la mantisa, que será siempre 1, se representa, es decir, no hay bit oculto. El exponente se codifica en exceso Z = 10.



- Determina cuál es el número más grande representable en este formato. Responde en decimal.

31,5

- Calcula la diferencia mínima entre dos números positivos sucesivos en este formato (contestar en forma de potencia de dos)

2^{-16}

- Determina cuál o cuáles de las siguientes afirmaciones son CIERTAS. Si ninguna es cierta contesta NINGUNA.

- A) Dado un número fijo de bits n , la cantidad de números a representar trabajando en exceso a Z depende de Z.
- B) Trabajando con 5 bits y exceso a Z = 25, el número 7 no es representable.
- C) El número decimal 0,2 puede representarse sin cometer ningún error (error absoluto = 0) en el formato IEEE(P)754.

- D) Los 128 primeros caracteres del código ASCII extendido coinciden con los 128 primeros caracteres del código ISO Latin 1.

B, D

- Se dispone de un periférico conectado al computador elemental que tienen asociado el vector de interrupción numero 5. En un momento dado el bit IF de la CPU se encuentra a ‘1’ (por lo que la CPU acepta interrupciones) y el periférico solicita interrupción activando la línea INT. Se desea saber cuál o cuáles de las acciones que se indican a continuación tendrán lugar durante la fase de desencadenamiento de la interrupción. Si ninguna ocurre, contesta NINGUNA.

- A) El registro R7 se decrementa en uno.
- B) Se activa la línea INTA.
- C) Se envía al periférico el número 0005h a través del registro MDR.
- D) Se realiza una lectura de la dirección de memoria 0005h.

B, D

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

PC = 0401h
IR = C0FDh

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: FFFD Paso 6: 03FE

- Se desea diseñar una nueva instrucción de suma para la CPU elemental que tenga la siguiente estructura:

ADD $R_x, [R_y]$

- El operando [Ry] es de tipo memoria y actúa como fuente. El operando Rx es de tipo registro y actúa como fuente y destino. Así, la operación realizada por esta instrucción se describe de la siguiente forma:

$$R_X \leftarrow R_X + [R_Y]$$

Teniendo en cuenta el funcionamiento de esta nueva instrucción, indica las señales de control necesarias para la ejecución de la instrucción ADD R0, [R1]

Paso 4	R1-IB, IB-MAR, READ
Paso 5	R0-IB, IB-TMPE
Paso 6	MDR-IB, ADD, ALU-TMPS, ALU-SR
Paso 7	TMPS-IB, IB-R0, FIN

- Dibuja el esquema de la ALU elemental para cantidades de un bit. (Se recomienda contestar a lápiz).

Este diagrama ilustra la implementación de un sumador de 4 bits con control de flujo de datos. Las entradas A y B se conectan a tres puertas lógicas: una AND de 2 entradas y dos OR de 3 entradas. Las salidas de estas puertas lógicas se dirigen a un multiplexor (MUX) y a un sumador elemental. El MUX selecciona entre las salidas de las puertas lógicas y la salida de un sumador elemental para producir el resultado final. El sumador elemental toma como entrada las salidas de las puertas lógicas y la entrada de acarreo C_{in}. El resultado del sumador elemental se conecta al MUX y al sumador elemental para la siguiente etapa. El MUX también recibe las señales de control Op₁ y Op₀ para seleccionar la operación correcta. La salida final del MUX es el Resultado.

BRZ salto

Bloque de código

salto: