



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.

Se ha construido para la CPU teórica un programa que utiliza el periférico luces a modo de teclado. El funcionamiento del programa es el siguiente:

El usuario coloca una combinación en los interruptores del periférico Luces y activa la generación de una interrupción, la rutina de tratamiento de esa interrupción muestra en la pantalla el código ASCII del número codificado por los 8 interruptores menos significativos en la pantalla, los 8 interruptores más significativos codifican el atributo de color de representación

COLOR				ASCII			

Por ejemplo si la combinación de entrada ha sido 0738h, la rutina mostrará en la pantalla el carácter correspondiente al ASCII 38h (carácter ‘8’) con el atributo de color 07h.

Además la rutina debe controlar que el usuario no coloque en los interruptores una combinación de un código ASCII no imprimible, en cuyo caso no se mostrará carácter alguno en la pantalla.

La variable *posición* controla la posición de escritura en la pantalla, y tendrá un valor entre 0 y 119. La rutina es la encargada de escribir el carácter correspondiente en la posición de la pantalla marcada por dicha variable

Para escribir la combinación leída del periférico luces en pantalla se utiliza el procedimiento *Escribir_en_pantalla* el cual recibe dos parámetros: la posición de escritura final en

pantalla y el código a escribir.

```
ORIGEN 300h
INICIO inicial
.PILA 40h
```

```
.DATOS
posicion VALOR 0
```

```
.CODIGO
inicial:
; Instalación rutina Luces
```

-- Hueco 1 --

```
MOVL R1, 12h
MOVH R1, 0h
MOV [R1], R0
```

```
; Activación de interrupciones
STI
```

```
forever:
jmp forever
```

```
PROCEDIMIENTO Luces
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
```

```
; Se carga la dirección del registro
; de datos del periférico luces
MOVL R0, 0FEh
MOVH R0, 8Fh
```

```
; Se lee el dato de los pulsadores
MOV R1, [R0]
```

```
; Se convierte a ASCII el dato leído
; aplicando la máscara adecuada
MOVL R2, 0FFh
MOVH R2, 0h
AND R3, R1, R2
```

```
; Si no es imprimible acabo
MOVL R4, 21h
MOVH R4, 0h
COMP R3, R4
BRS no_imprimible
```

```
MOVL R4, 7Dh
MOVH R4, 0h
COMP R3, R4
BRNS no_imprimible ; ++++++++
```

```
; Se carga la dirección base de la pantalla
MOVL R0, 0h
MOVH R0, 80h
```

```
; Se lee la posición de escritura
MOVL R2, BYTEBAJO DIRECCION posicion
MOVH R2, BYTEALTO DIRECCION posicion
MOV R3, [R2]
```

```
; Se comprueba si es necesario inicializar
; la posicion de pantalla
MOVL R4, 120
MOVH R4, 0
COMP R3, R4
BRNZ no_inicializo ; ++++++++
```

```
XOR R3, R3, R3
```

```
no_inicializo:
; Se calcula la posición en pantalla para
; escribir
ADD R0,R0,R3
```

```
; Se cargan los parametros
PUSH R0
PUSH R1
```

```
; Se llama al procedimiento
; Escribir_en_pantalla
CALL Escribir_en_pantalla
```

-- Hueco 2 --

```
; Incremento posición de pantalla
INC R3
; Guardo posición de pantalla
MOV [R2], R3
```

```
no_imprimible:
POP R4
POP R3
POP R2
POP R1
POR R0
```

-- Hueco 3 --

```
FINP
```

PROCEDIMIENTO Escribir_en_pantalla

```
PUSH R6
MOV R6, R7
PUSH R0
PUSH R1
; Recupero parametro 1
```

-- Hueco 4 --

```
MOV R1, [R6]
; Recupero parametro 2
INC R6
MOV R0, [R6]
```

```
; Escribo en pantalla
MOV [R0], R1
POP R1
POP R0
POP R6
RET
FINP
```

FIN

- ¿A qué vector de la tabla de vectores de interrupción está asociada la interfaz de luces? **Contesta en decimal.**

18

- ¿A qué rango de direcciones de memoria responde la interfaz de vídeo? ¿Y la de luces? **Responde en hexadecimal.** Ejemplo de un rango de direcciones: **00FFh-1245h**. Pista: El número de direcciones a los que responden ambas interfaces es una potencia de 2.

Rango direcciones interfaz de vídeo: 8000h - 807Fh

Rango direcciones interfaz de luces: 8FFEh – 8FFFh

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

MOVL R0, BYTEBAJO DIRECCION Luces

MOVH R0, BYTEALTO DIRECCION Luces

[R6] R2, 0F0h

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

INC R7

INC R7

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

IRET

- 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 luces). **ATENCIÓN: Hay un hueco para usar hasta 8 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-0FFFh.**

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

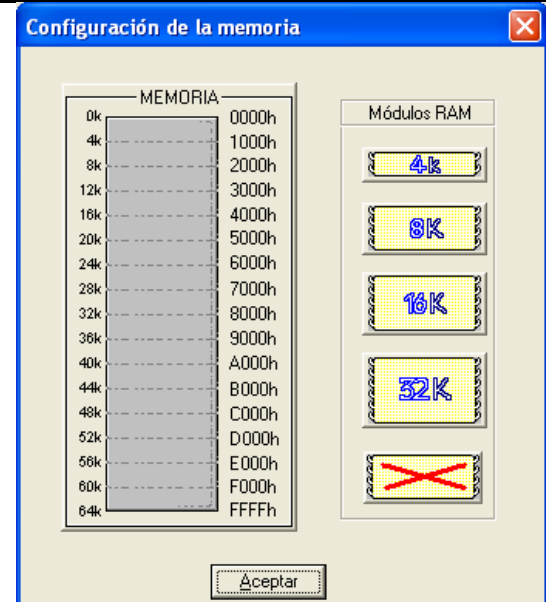
Disp. 2 → Tamaño: 4K Rango dir: 9000h-9FFFh

Disp. 3 → Tamaño: 8K Rango dir: A000h-BFFFh

Disp. 4 → Tamaño: 16K Rango dir: C000h-FFFFh

Disp. 5 → Tamaño: Rango dir:

Disp. 6 → Tamaño: Rango dir:



- ¿Qué instrucción o instrucciones falta/n en el
-- Hueco 4 --?

INC R6

INC R6

- Como puedes comprobar, el programador ha puesto un tamaño de pila grande para evitar problemas. Pero, ¿cuál sería el tamaño mínimo de pila con el que funcionaría el programa? **Responde en hexadecimal con dos dígitos.**

0Dh

- ¿Cuál es la codificación de la instrucción **BRNZ no_inicializo** de la rutina luces (la que aparece a la izquierda de un comentario con varios asteriscos)? **Responde en hexadecimal.**

F501h

- ¿Qué señales están activas en el último paso de ejecución de la última de las instrucciones **BRNS no imprimible** de la rutina *lucos* (a que aparece a la izquierda de un comentario con varios signos de suma), si sabemos que cuando se ejecutó la instrucción anterior R3 valía 007Fh? ¿De qué número de paso se trata?

Señales activas: TMPS-IB, IB-PC, FIN

Número de paso: 6

- ¿Cuál o cuáles de las siguientes afirmaciones son **ciertas**? Si ninguna es cierta contesta NINGUNA.

- La resta de los códigos ASCII de los caracteres '9' (minuyendo) y '0' (cero, sustraendo) es igual a 9.
- El número de direcciones a las que puede acceder la CPU elemental depende del número de líneas del bus de datos del sistema.
- La ALU es utilizada por toda instrucción del juego de instrucciones de la CPU elemental en alguno de sus pasos de ejecución.
- En la CPU elemental, la instrucción PUSH Rx requiere más pasos de ejecución que la instrucción POP Rx

A, C y D

Se dispone de un computador con un bus de direcciones de 16 líneas y un bus de datos de 16 líneas.

- Se desea conectar al computador indicado un dispositivo de memoria que ocupará el 6,25 % del espacio de direcciones. Para ello se dispone de chips de memoria de 8 líneas de dirección y 4 de datos. ¿Cuántos chips serán necesarios para construir este dispositivo?

64

- Indica el rango de direcciones en el que se ubica el banco 8 del dispositivo de memoria al que se hace referencia en la pregunta anterior, teniendo en cuenta que la dirección 2615h pertenece a dicho dispositivo de memoria. (Los bancos están numerados como banco 0, banco 1...), y la dirección dada puede pertenecer o no al banco de memoria indicado. (Formato de solución para el rango: XXXXh – YYYYh)

2800h – 28FFh

Se pretende confeccionar una nueva instrucción para la CPU teórica, que permite efectuar la operación lógica NOR sobre dos operandos ubicados en memoria, dejando el resultado en un registro. El mnemónico de la nueva instrucción es:

NOR Rd, [Ri₁], [Ri₂] ; Rd ← [Ri₁] NOR [Ri₂]

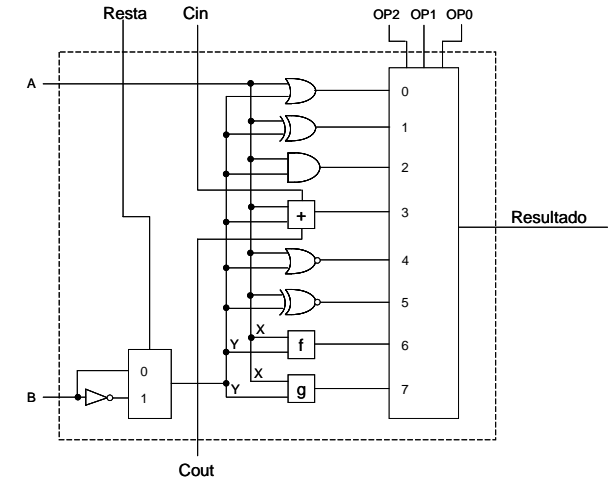
La siguiente tabla muestra las señales de control necesarias en cada paso para implementar esta instrucción, tomando como ejemplo la instrucción NOR R0, [R1], [R2]:

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	MDR-IB, OR, ALU-TMPS
10	
11	R0-IB, TMPE-SET, XOR, ALU-SR, ALU-TMPS
12	TMPS-IB, IB-R0, FIN

- ¿Qué señales se activan en el paso 10?

TMPS-IB, IB-R0

Si se modifica la ALU de 1 bit de la ALU de la CPU teórica, implementándola como se muestra en la figura:



y donde las funciones f y g implementan la tabla de verdad mostrada en las siguientes tablas:

X	Y	f	X	Y	g
0	0	0	0	0	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	1	1	1	0

- Calcular el resultado devuelto por la ALU (en hexadecimal) y los valores de los bits de estado si el operando 1 (ó A) de la ALU tiene el valor F45Eh, el operando 2 (ó B) tiene el valor 0FFFh, OP2=1, OP1=1, OP0=0, RESTA=1 y Cin=0.

Resultado: F000h

Z = 0 C = 0 O = 0 S = 1



Cuando la CPU se encuentra ejecutando el paso 1 de la instrucción **COMP R0, R1** el valor de algunos registros de la CPU es el que muestra la tabla siguiente:

R0=0001h	R1=0000h	R2=0000h	R3=10F1h
R7=1100h	PC=10F0h	MAR=10F0h	TMPS=10F1h

También se sabe que cierta zona de la memoria del equipo contiene la siguiente información:

DIR.	CONT.
0000	7124
0001	AB38
0002	3951
0003	2000
0004	5720
0005	B4FF

En ese instante, un periférico asociado al vector de interrupción 3 solicita una interrupción, que será aceptada. Se sabe que la rutina de tratamiento de interrupción tiene seis instrucciones y que permite continuar la ejecución normal del programa en curso tras su ejecución.

— ¿Qué valor tomará el registro MAR tras finalizar la ejecución de la instrucción en curso? Contestar en hexadecimal.

10FFh

— ¿Cuál es el contenido de la dirección de memoria 10FFh durante la ejecución de la rutina de tratamiento de la interrupción? Contestar en hexadecimal.

0001h

— ¿Cuál es el máximo valor que ha tomado el registro PC durante la ejecución de la rutina de interrupción? Contestar en hexadecimal.

2006h



— Un computador que maneja datos de 12 bits se emplea para representar números en coma fija. Si se sabe que el mayor número representable en el formato de coma fija elegido es **127.9375**, ¿cuál es el menor número representable en ese formato? Contestar en decimal.

100000000000b = -128

— La señal de reloj que llega a la unidad de control tiene una frecuencia de 20MHz. ¿Cuánto tiempo tardan en ejecutarse las dos instrucciones que inicializan R0? Contestar en microsegundos

4*2/20 = 0,4 microsegundos