

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.

- ¿Cuál o cuáles de las siguientes afirmaciones son CIERTAS (puedes responder “ninguna” o “todas” si así lo consideras)?
- A) El valor de un vector de interrupción indica la prioridad de la interrupción asociada a ese vector.
- 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) La rutina de servicio de una interrupción puede recibir parámetros a través de la pila.
- D) La línea **INTA** nunca se puede activar si no se ha activado la línea **INT** previamente.

D

- ¿Cuál o cuáles de las siguientes afirmaciones son CIERTAS (puedes responder “ninguna” o “todas” si así lo consideras)?
- A) Con n bits se pueden representar la misma cantidad de números enteros, utilizando el convenio signo-magnitud que utilizando el convenio exceso a Z.
- B) En el formato IEEE-754 el menor número desnormalizados positivo distinto de cero

- es 2^{-127}
- 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) En el código ASCII extendido la codificación de las letras mayúsculas y minúsculas se diferencian en 1 bit.

C

- En una UC microprogramada para la CPU teórica, las palabras de control se interpretan como se indica en la figura (solamente se muestra los 11 bits menos significativos de la palabra de control).

⋮	R3-IB	IB-R3	R5-IB	IB-R5	IB-TMPE	TMPE-CLR	SUB	TMPS-IB	IB-PC	ALU-SR	FIN
...											

Para una determinada instrucción se generan las señales de control para las siguientes palabras de control:

Paso 4: 001 0100 0000
Paso 5: 100 0001 0011

- ¿Cuál es la codificación de la instrucción que se ejecuta con esas palabras de control?
Expresar el resultado en hexadecimal

6D60h (COMP R5,R3)

- ¿Cuál sería la representación según el formato IEEE-(P)754 de la cantidad binaria $-0,101x2^{-130}$? (Contestar en hexadecimal)

80050000h

- Con objeto de mejorar la funcionalidad del juego de instrucciones de la CPU elemental se implementa una versión para la operación ADD que hace uso de direccionamiento indirecto a memoria de varios de sus operandos. La nueva versión:

ADD [R0],R1,[R2]

Almacena en la dirección de memoria apuntada por R0 la suma del contenido de dirección de memoria apuntada por R2 y el contenido de R1.

Las acciones de control necesarias para llevar a cabo dicha instrucción se muestran parcialmente a continuación. Completa las señales de control que faltan en los pasos 4,6 y 8

Pasos	Acciones de Control
4	R2-IB, IB_MAR, READ
5	R1-IB, IB-TMPE
6	MDR-IB,ADD, ALU_TMPS, ALU_SR
7	R0-IB,IB-MAR
8	TMPS-IB, IB-MDR,WRITE
9	FIN

- En 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



Se tiene un formato de coma flotante con 1 bit para el signo de la mantiza, 5 para la magnitud de la mantisa y 3 bits para el exponente (tal como se muestra en la figura). La mantisa se representa en signo-magnitud, todo fracción y se representa el primer bit. El exponeente se representa en exceso a Z. Se sabe que el número 110010101 es este formato es el -2.25 en decimal.

1	5	3
Signo	Magnitud mantisa	Exponente

– ¿Cuánto vale Z?

Z=3

En un computar basado en la CPU elemental se tiene el siguiente mapa de memoria:

- Dispositivos de memoria RAM a partir de la posición de memoria 0000h y tamaño 36K
- Dispositivos de memoria ROM a partir de la posición de memoria C800h y tamaño 14K

¿Qué tamaño del espacio de direcciones queda libre para poder mapear dispositivos de Entrada/Salida? ¿A partir de qué dirección (en hexadecimal)?

Tamaño: 14K

Dirección comienzo: 9000h

En el espacio de direcciones descrito anteriormente la memoria RAM está formada por 9 dispositivos de memoria cuya organización es del tipo 4Kx16. y cada uno está formado por 4

bancos de 8 chips

– ¿Cuál es la organización interna de los Chips? (contestar en el formato NxM; por ejemplo 32Kx1)

1Kx2

– ¿Cuántas líneas de dirección se conectarían al decodificador de cada dispositivo 4Kx16 anterior? ¿Cuántas líneas de dirección entrarían a cada dispositivo de memoria?

Nº líneas al decodificador: 2

Nº líneas al dispositivo: 12

La CPU elemental está ejecutando el siguiente fragmento de código:

```
Leer: MOV R1, [R0]
      ADD R3, R1, R1
      INC R0
      DEC R5
      BRNZ Leer
      MOVL R2, 3
```

En un instante dado, la CPU se halla ejecutando una de las instrucciones anteriores, y en ese instante los valores de sus registros son:

R0	7030h	R4	6576h
R1	0034h	R5	0100h
R2	0000h	R6	1FFAh
R3	0000h	R7	7422h
PC	7141h	IR	4324h

– Si en el instante aludido el valor de registro TMPS es 0068h ¿qué ciclo de ejecución acaba de finalizar?

El ciclo 5



Se pretende desarrollar un módulo software capaz de validar soldaduras entre diferentes piezas de acero realizadas en una planta siderúrgica.

Las soldaduras que se pretenden validar se realizan utilizando un proceso de soldadura eléctrica. Una soldadura de este tipo se puede validar a partir del análisis de dos señales analógicas adquiridas durante el proceso:

- Temperatura que alcanza el cordón de soldadura
- Corriente eléctrica aplicada al material a soldar

Se desarrolla un primer prototipo del módulo de validación de soldaduras en lenguaje ensamblador para la CPU Elemental. En este primer prototipo las señales analógicas se simplifican y se representan con dos listas de números naturales:

- una lista de cinco elementos, **Temperatura**, que simula la señal de temperatura del proceso
- una lista de cinco elementos, **Corriente**, que simula la señal de corriente eléctrica del proceso

Los límites máximo y mínimo que pueden tomar cada una de las señales durante el proceso de soldadura se almacenan en las variables **TmpSuperior**, **TmpInferior**, **CrrSuperior** y **CrrInferior**.

Se dispone de un procedimiento, **ChqueanSenyal**, que determina si los valores de una señal se encuentran dentro de unos límites mínimo y máximo. Para realizar esta tarea, el procedimiento recibe a través de la pila (y en este orden) los siguientes parámetros:

- Dirección de memoria en la que almacenará el resultado del chequeo
- Dirección de memoria del primer elemento de la señal a chequear
- Límite superior para la validación

Dirección	Contenido
050E	0000h
050F	0001h
0510	0000h
051D	D017h

12



```
ORIGEN      500h
INICIO      ini_programa
.PILA       50h

.DATOS

Temperatura VALOR  773, 937, 900,
              997, 777
Corriente   VALOR  14575, 15278,
              15787, 15001, 14857
TmpSuperior VALOR  950
TmpInferior  VALOR  750
CrrSuperior  VALOR  16000
CrrInferior  VALOR  14500
ResultTmp    VALOR  -1
ResultCrr    VALOR  -1
ResultSoldadura VALOR -1

.CODIGO

ini_programa:
; Cargar parámetros en la pila
---- HUECO 1a ----
PUSH R0
---- HUECO 1b ----
PUSH R1
MOVL R2, BYTEBAJO DIRECCION TmpSuperior
MOVH R2, BYTEALTO DIRECCION TmpSuperior
PUSH R2
MOVL R2, BYTEBAJO DIRECCION TmpInferior
MOVH R2, BYTEALTO DIRECCION TmpInferior
PUSH R2

; Chequear la señal de temperatura
CALL ChequearSenyal
MOVL R5, 04h
MOVH R5, 00h
---- HUECO 2 ----

; Cargar parámetros en la pila
INC R0
PUSH R0
INC R5
ADD R1, R1, R5
DEC R5
PUSH R1
INC R2
PUSH R2
INC R2
PUSH R2
```

```
;Chequear la señal de corriente
CALL ChequearSenyal
---- HUECO 2 ----

; R4 <-- ResultTmp
; R5 <-- ResultCrr
DEC R0
MOV R4, [R0]
INC R0
MOV R5, [R0]

; Calcular resultado de soldadura en R4
---- HUECO 3 ----

; Almacenar resultado del análisis
INC R0
MOV [R0], R4
JMP -1

PROCEDIMIENTO ChequearSenyal
PUSH R6
MOV R6, R7

; Salvar registros generales
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5

; Acceso a los parámetros
; R2 <-- Límite Inferior Señal
; R3 <-- Límite Superior Señal
; R4 <-- Dirección de Señal
; R5 <-- Dirección de Resultado
INC R6
INC R6
MOV R2, [R6]
MOV R2, [R2]

---- HUECO 4 ----

INC R6
MOV R5, [R6]

XOR R0, R0, R0
MOVL R1, 05h
MOVH R1, 00h
```

```
bucle:
; R0 <-- VALOR de la señal en la
; iteración actual
MOV R0, [R4]
COMP R0, R2
; Si VALOR es menor que ímite Inferior
; ir a limite_inferior
---- HUECO 5a ----

---- HUECO 5b ----
; Si VALOR es mayor que Límite Superior
; ir a limite_superior
BRC limite_superior

; Preparar la siguiente iteración del
; bucle
INC R4
DEC R1
BRNZ bucle
JMP senyal_correcta

limite_inferior:
limite_superior:
---- HUECO 6 ----
JMP resultado

senyal_correcta:
MOVL R0, 01h
MOVH R0, 00h

resultado:
; Escribir resultado del chequeo de la
; señal
MOV [R5], R0

; Restaurar registros generales
; y retornar al programa principal
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
---- HUECO 7 ----

FINP

FIN
```