



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.

- Indica las señales de control necesarias para implementar la instrucción INC [R0].

Pasos	Señales de Control
4	R0-IB, IB-MAR, READ
5	(ciclo de espera)
6	TMPE-CLR, CarryIn, MDR-IB, ADD, ALU-TMPS, ALU-SR
7	TMPS-IB, IB-MDR, WRITE
8	FIN

- Se tiene un formato en coma flotante con 8 bits (los más significativos) para la representación de la mantisa (signo-magnitud) y 4 para la representación del exponente en exceso a 4. La normalización utilizada es todo fracción. ¿Cuál es la máxima diferencia entre dos números consecutivos? Responder en decimal.

16

- En el anterior formato representar el número 7.25.

0111 0100 0111

- La secuencia de bytes 65h 34h C7h A7h 25h representa un conjunto de caracteres en formato UTF-8. ¿Cuántos caracteres se encuentran codificados?

4

Se tiene el siguiente programa para la CPU elemental sin ningún cometido en particular.

```

ORIGEN ****
INICIO ini
.PILA 10h
.CODIGO
    IRET
ini:
    XOR R0, R0, R0
    MOVL R1, 00
    MOVH R1, 10h
    MOV [R0], R1
    STI
    JMP -1
FIN
    
```

- Durante la ejecución de la primera instrucción se produce una petición de interrupción. Posteriormente, durante la fase de aceptación de la interrupción aparecen en el registro MDR de forma consecutiva los valores, 11h, 1006h, 0h. ¿Cuál es la dirección de comienzo de la rutina de interrupción? Responder en hexadecimal.

1000h

- ¿Cuál es el valor del registro de estado justo antes de empezar a ejecutar la primera instrucción de la rutina de interrupción? Responder en hexadecimal.

10h

- Se ha construido un dispositivo de memoria de organización 256KB a partir de chips de organización 16KB, ¿qué tipo de decodificador se ha utilizado? Ej: 2:4

4:16

- Dibuja el circuito de activación necesario para mapear el dispositivo anterior en la zona de memoria más alta de un computador con un espacio de direcciones de 2MB.

a20 ^ a19 ^ a18

- En la CPU Elemental,
- ¿Qué valor se produce si todas las entradas de la ALU están a 1 excepto la entrada de resta y la de CarryIn? Responder en decimal interpretado con signo.

-2

- ¿Cuál será el valor del registro de estado (ZCOS) tras la operación anterior? Responder en hexadecimal.

0101 = 5h

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 en 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, **ChequeanSenyal**, 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
- Límite inferior para la validación

Este procedimiento recorre los elementos de la señal que se le indica como parámetro y comprueba si todos los valores de la misma están comprendidos entre los límites dados, es decir, mayores que el límite inferior y menores que el límite superior. Si es así, en la posición de memoria en la que almacena el resultado del chequeo escribirá un "1". En caso contrario, en esa posición escribirá un "0".

El algoritmo para la validación de una soldadura invoca al procedimiento **ChequeaSenyal** una vez para cada una de las señales del proceso de soldadura que deba validar. Los resultados de estos chequeos estarán almacenados en **ResultTmp** y **ResultCrr**. Si el resultado de los chequeos de ambas señales es satisfactorio, la soldadura es correcta, por lo que se escribirá un "1" en la variable **ResultSoldadura**. En caso contrario, la soldadura es defectuosa, por lo que se escribirá un "0" en dicha variable.

Al finalizar la ejecución del programa,

- ¿qué valores habrá almacenados en las siguientes posiciones de memoria? [Responder en hexadecimal].

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

- ¿Qué dos instrucciones faltan en el **Hueco 1a** y qué dos instrucciones faltan en el **Hueco 1b**?

Hueco 1a:

```
MOVL R0, BYTEBAJO DIRECCION ResultTmp
MOVH R0, BYTEALTO DIRECCION ResultTmp
```

Hueco 1b:

```
MOVL R1, BYTEBAJO DIRECCION Temperatura
MOVH R1, BYTEALTO DIRECCION Temperatura
```

- ¿Qué instrucción falta en los huecos etiquetados como **Hueco 2**?

```
ADD R7, R7, R5
```

- ¿Qué instrucción falta en el hueco etiquetado como **Hueco 3**?

```
AND R4, R4, R5
```

- ¿Qué instrucción o instrucciones falta(n) en el hueco etiquetado como **Hueco 4**?

```
INC R6
```

```
MOVR3, [R6]
```

```
MOVR3, [R3]
```

```
INC R6
```

```
MOVR4, [R6]
```

Apellidos _____

Nombre _____

DNI _____


Examen de Fundamentos de Computadores. Área de Arquitectura y Tecnología de Computadores Convocatoria de Septiembre: 15-09-2006

```

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

senal_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
POP R6
RET

FINP

FIN

```

- ¿Qué instrucción falta en el **Hueco 5a** y qué instrucción falta en el **Hueco 5b**?

Hueco 5a:

BRC limite_inferior

Hueco 5b:

BRC limite_superior

- ¿Qué instrucción falta en el hueco etiquetado como **Hueco 6**?

XOR R0, R0, R0

La figura muestra un instante de la ejecución del programa en el simulador de la CPU Elemental. A la vista de la imagen,

- ¿cuál será el valor del registro R7 al finalizar la ejecución del programa? [Responder en hexadecimal].

05B1h

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

- El número 23 no es representable en exceso a Z central con 5 bits.
- El paso de parámetros a un procedimiento a través de registros del procesador es más rápido que a través de la pila de programa.
- Sólo las instrucciones aritmético-lógicas modifican los bits del registro de estado del procesador.
- La sección de pila permite la creación de variables locales en procedimientos.

B, D

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

- El tamaño del espacio direccionable de un computador depende del ancho de su bus de direcciones y del ancho de su bus de datos.
- En la CPU Elemental, la instrucción PUSH R1 tarda tanto tiempo en ejecutarse como la instrucción POP R1.

- El programa en ensamblador más grande para la CPU Elemental puede tener 2^{16} instrucciones.
- La Unidad de Control comprueba el estado de la línea INT en el último paso de ejecución de todas las instrucciones.

C, D

