

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_

DNI \_\_\_\_\_



## Examen de Fundamentos de los computadores. Área de Arquitectura y Tecnología de Computadores

Febrero. Gijón: 13-2-04

## 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.

- Expresar **en hexadecimal** el número 234,75 en formato IEEE-754 simple.

436AC000

- ¿Cuál o cuáles de las siguientes afirmaciones son **ciertas**?

- La resta de los códigos ASCII de los caracteres '9' (minuendo) 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.
- Se denomina registro índice a todo registro de propósito general cuyo valor es interpretado como una dirección de memoria a la que se accederá para leer o escribir un dato.

A, C y D

- Se pretende obtener una memoria 32Kx8 y se dispone de 100 chips de memoria 4Kx1 para confeccionarla. ¿Cuántos chips nos sobrarán?

36

La siguiente porción de programa para la CPU elemental escribe un carácter espacio en blanco en la esquina inferior derecha del dispositivo pantalla:

```
XOR R0, R0, R0
MOVL R0, 00100000b
MOVL R1, 77h
MOVH R1, 36
MOV [R1], R0
```

- Dibuja el circuito de activación del dispositivo pantalla. Hay que indicar el peso de cada línea utilizada.

$$\sim a_{15} \sim a_{14} \cdot a_{13} \sim a_{12} \sim a_{11} \cdot a_{10} \sim a_9 \sim a_8 \sim a_7$$

- ¿Cuál o cuáles de las siguientes afirmaciones son **ciertas**?

- La unidad de control de la CPU es un sistema digital secuencial.
- La unidad aritmético lógica de la CPU teórica trabaja con operandos de 15 bits cuando estos están codificados en complemento a 2 y con operandos de 16 bits cuando están codificados en binario natural.
- El contador de microprograma de una unidad de control microprogramada contiene la dirección de memoria principal en la cual se encuentra la siguiente palabra de control.
- La CPU teórica accede a los dispositivos de entrada salida empleando las instrucciones MOV Rd, [Ri] y MOV [Ri], Rs.

a) y d)

En un sumador capaz de operar con cantidades de 7bits, por una de sus entradas se introduce el número -4 codificado en complemento a 2. Por la otra se introduce el código ASCII de la letra 'E'.

- ¿Cuál será el resultado de la suma? Escríbelo en hexadecimal. Datos: el código ASCII de la 'a'=61h. Además, se sabe que si se lleva a cabo la operación XOR bit a bit del código ASCII de la 'Z' con el de la 'z' el resultado es 20h.

41h

Se dispone de una CPU con 32 líneas de datos y 16 líneas de direcciones. El espacio de direccionamiento se divide en 16 zonas iguales, numeradas del 0 (direcciones más bajas) al 15 (direcciones más altas). En las zonas 0 a 12 se instala memoria RAM. En las zonas 13 a 15 se instala memoria ROM.

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

0000h-CFFFh

- En un sumador elemental de 8 bits se sumaron el número positivo más grande que se puede representar en complemento a 2 con 8 bits y el número cero representado en exceso a Z central con 8 bits. ¿Cuál ha sido el resultado obtenido, interpretado en complemento a 2 y expresado en decimal?

-1

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

- A) En la arquitectura Von Neumann se conectan entre sí CPU, sistema de memoria y sistema de E/S solamente a través del bus de direcciones y del bus de datos.
- B) En la CPU elemental, la instrucción MOV [R0], R1 tarda tanto en ejecutarse como la instrucción MOV R0, [R1].
- C) A partir del código ASCII de la letra 'f', se puede deducir el código ASCII de todas las letras mayúsculas y minúsculas.
- D) El número de bancos en un dispositivo de memoria conectado al computador elemental depende del número de líneas del bus de datos.

B,C

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, [Ri<sub>1</sub>], [Ri<sub>2</sub>]** ; Rd ← [Ri<sub>1</sub>] NOR [Ri<sub>2</sub>]

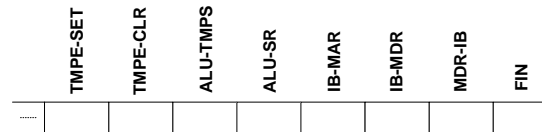
La siguiente tabla muestra las señales de control necesarias en cada paso para implementar esta instrucción, tomando como ejemplo la instrucción NOR R0, [R1], [R2]:

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	
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, OR, ALU\_TMPS

Una unidad de control microprogramada genera palabras de control que se interpretan tal y como se muestra en la figura (sólo se muestran los 8 bits inferiores de la palabra de control):



En un instante dado, se sabe que el byte inferior de la palabra de control contiene el valor 01h, que el paso de ejecución de instrucción es el 4 y que si se aplica una máscara de valor 0700h sobre la codificación de la instrucción utilizando la función AND produce el valor 0700h. Además, se sabe que no es una instrucción de salto.

— ¿Qué posible/s instrucción/es podría/n estar ejecutándose?

MOV R7, Rs; MOVL R7, Inm\_8; MOVH R7, Inm\_8

El código mostrado en la página siguiente pertenece a un programa para la CPU teórica que muestra el mensaje OK en la esquina superior derecha de la pantalla si la última tecla pulsada es la barra espaciadora y que muestra la pantalla vacía si la última pulsación corresponde a otra tecla. Para ello, se utiliza la rutina de servicio de interrupción de teclado *servicio\_teclado*, encargada de comprobar si la tecla pulsada es la correcta, escribiendo el valor 0 (no se pulsó la barra espaciadora) ó el valor 1 (se pulsó la barra espaciadora) en la variable global *hayespacio* y borrando la pantalla en el primero de los dos casos. El programa principal examina la variable global *hayespacio*, imprimiendo el mensaje OK si la variable está a 1. Además, se sabe que el teclado responde a la señal INTA enviada por la CPU escribiendo el valor 5 en el bus de datos del sistema.

— ¿Cuál es el valor mínimo que puede seguir a la directiva .PILA en el **HUECO 1** ?

8

— ¿Qué instrucción/instrucciones falta/n en el **HUECO 2** ?

MOVL R0, BYTEBAJO DIRECCION servicio\_teclado

MOVH R0, BYTEALTO DIRECCION servicio\_teclado

MOVL R1, 5

MOVH R1, 0

— ¿Qué instrucción/instrucciones falta/n en el **HUECO 3** ?

INC R7

INC R7

— ¿Qué señales de control están activas en el último paso de la instrucción inmediatamente anterior al **HUECO 3** (CALL imprime\_mensaje) ? ¿De qué número de paso se trata?

Nº de paso: 8

Señales de control activas en el paso:

TMPS\_IB, IB\_PC, FIN

— Si las 6 instrucciones posteriores al **HUECO 2** (desde MOV [R1], R0 hasta XOR R4, R4, R4; ambas incluidas) tardan en ejecutarse 14 milésimas de segundo, ¿cuál es la frecuencia (en **KHz**) de la señal de reloj CLK?

2

— ¿Cuál es la última posición del espacio de memoria de la CPU elemental mapeada por el dispositivo teclado? **Responder en hexadecimal.**

F078h

Apellidos \_\_\_\_\_

Nombre \_\_\_\_\_

DNI \_\_\_\_\_



Examen de Fundamentos de los computadores. Área de Arquitectura y Tecnología de Computadores

Febrero. Gijón: 13-2-04

```
ORIGEN 9000h
INICIO ini

.PILA (-----HUECO 1-----)
```

```
.DATOS
mensaje VALOR "OK"
hayespacio VALOR 0
```

```
.CODIGO
```

```
ini:
```

```
CLI
```

```
; Inicializar el vector de interrupcion
; asociado al teclado
```

```
(-----HUECO 2-----)
```

```
MOV [R1], R0
```

```
; Inicializar posicion de memoria
; de la variable global hayespacio
MOVL R2, BYTEBAJO DIRECCION hayespacio
MOVH R2, BYTEALTO DIRECCION hayespacio
```

```
; Valor auxiliar
MOVL R3, 1
MOVH R3, 0
```

```
; Valor auxiliar
XOR R4, R4, R4
```

```
; Código en el buffer de teclado
; correspondiente a la barra espaciadora
MOVL R5, 20h
MOVH R5, 32h
```

```
; Almacenar dirección del mensaje "OK"
MOVL R0, BYTEBAJO DIRECCION mensaje
MOVH R0, BYTEALTO DIRECCION mensaje
```

```
; Tamaño del mensaje
MOVL R1, 2
MOVH R1, 0
```

```
STI
```

```
esperando_tecla:
MOV R6, [R2]
COMP R6, R3
BRNZ esperando_tecla
CLI
CALL limpiar_pantalla
PUSH R0
PUSH R1
CALL imprime_mensaje
```

```
(-----HUECO 3-----)
```

```
MOV [R2], R4
STI
JMP esperando_tecla
```

```
; Rutina de servicio de interrupción de teclado
; que indica en la variable global hayespacio
; si se ha pulsado la barra espaciadora.
; La rutina deja en la posición de memoria
; asociada a esa variable un 1 en caso afirmativo,
; y un cero en caso contrario. En este último
; caso, borra además la pantalla.
```

```
PROCEDIMIENTO servicio_teclado
PUSH R0
PUSH R1
```

```
; Leer de teclado
MOVL R0, 0
MOVH R0, 0F0h
MOV R1, [R0]
```

```
; Comprobar que la tecla pulsada es la
; barra espaciadora, escribiendo en hayespacio
; el valor apropiado; en caso contrario,
; borrar pantalla
COMP R1, R5
BRNZ borrar
MOV [R2], R3
JMP salir_servicio
```

```
borrar:
MOVL R1, 120
MOVH R1, 0F1h
; Mascara para borrar pantalla
MOV R0, [R1]
OR R0, R0, R3
MOV [R1], R0
```

```
salir_servicio:
POP R1
POP R0
IRET
FINP
```

```
PROCEDIMIENTO imprime_mensaje
PUSH R6
MOV R6, R7
PUSH R0
PUSH R1
PUSH R2
PUSH R3
```

```
INC R6
INC R6
MOV R0, [R6] ; Tamaño
INC R6
MOV R1, [R6] ; Direccion
```

```
; Apuntar a esquina superior izquierda
; de la pantalla
MOVL R2, 0
MOVH R2, 0F1h
```

```
imprimir_actual:
MOV R3, [R1]
MOVH R3, 7
MOV [R2], R3
INC R1
INC R2
DEC R0
BRNZ imprimir_actual
```

```
POP R3
POP R2
POP R1
POP R0
POP R6
RET
FINP
```

```
PROCEDIMIENTO limpiar_pantalla
PUSH R0
PUSH R1
```

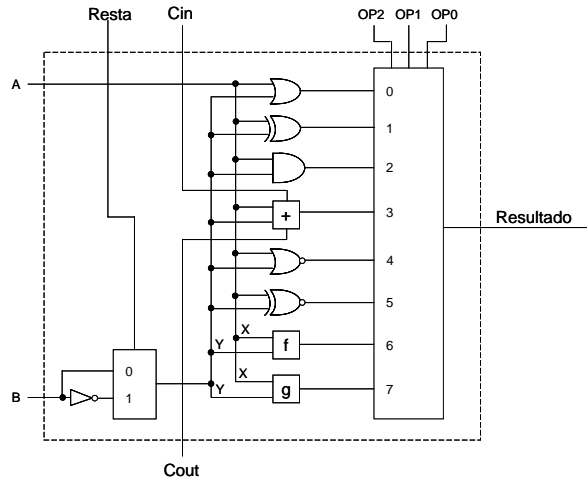
```
MOVL R1, 120
MOVH R1, 0F1h
MOVL R0, 3
MOVH R0, 0
```

```
MOV R0, [R1]
OR R0, R0, R3
MOV [R1], R0
```

```
POP R1
POP R0
RET
FINP
```



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
0	0	0
0	1	1
1	0	0
1	1	1

X	Y	bg
0	0	1
0	1	0
1	0	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
------------------

$$Z=0 \quad C=0 \quad O=0 \quad S=1$$

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

- 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=0,  
OP0=1, RESTA=1 y Cin=0.

Resultado: FBA1h

$$Z=0 \quad C=0 \quad O=0 \quad S=1$$

- Durante los pasos de ejecución de una determinada instrucción en la CPU de ejemplo, en el bus interno se han visto los valores que se muestran en la siguiente tabla. ¿Cuáles deberían ser los valores que faltan en los huecos?

Ciclo 1	C020
2	C021
3	C0F5
4	C021
5	FFF5
6	C016