

Apellidos _____

Nombre _____

DNI _____


Examen de Fundamentos de Computadores. Área de Arquitectura y Tecnología de Computadores
Convocatoria de Junio: 13-06-2006
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.

Sea un computador basado en la CPU Elemental que utiliza una Unidad de Control microprogramada. El significado de los 14 bits inferiores de las palabras de control es el siguiente:

TMPE-SET	CARRY_IN	TMPS_IB	SR-IB	ADD	IB-TMPE	ALU_TMPS	PC-IB	IB-PC	JUMP	TMPS_IB	TMPE-CLR	R7-IB	FIN
...													

A continuación, se muestra una tabla con los 14 bits inferiores de las palabras de control que se ejecutaron en los últimos tres pasos, junto con el valor que se encontraba en el bus interno (IB) en cada uno de estos pasos:

Paso	Palabra de control	Contenido de IB
4	0140h	1000h
5	0290h	FFFAh
6	0029h	0FFAh

- ¿Cuál es la codificación de la instrucción que se acaba de ejecutar si se sabe que el bit de peso 13 de dicha codificación es igual a 0? Responder en hexadecimal.

C0FAh

La cantidad C2874000h representa un número en formato coma flotante IEEE 754 en precisión simple.

- ¿Con qué cantidad decimal se corresponde? Responde con un número decimal donde no se incluya ninguna potencia de dos.

-67,625

En un computador basado en la CPU Elemental se mapea sólo un dispositivo de memoria RAM a partir de la dirección C000h, que cubre el 25% del espacio direccionable. El dispositivo está construido a partir de chips que almacenan 2^{12} palabras de 1 byte cada una. Los chips se comienzan a numerar en 0 por la esquina superior izquierda.

- Si una nota del fabricante advierte que el chip 5 del dispositivo está defectuoso, ¿qué rango de direcciones de memoria se verá afectado? Responder en hexadecimal.

[E000h–EFFFh]

- ¿Cuántos dispositivos periféricos que soliciten la atención de la CPU a través de una interrupción para realizar operaciones de E/S se pueden conectar a este computador?

Ninguno

- ¿Cuál será el contenido de la posición CF55h si se sabe que almacena el número 251,375 en formato coma fija sin signo utilizando la mitad superior de la palabra para representar la parte entera y la mitad inferior para la parte fraccionaria? Responder en hexadecimal.

FB60h

En un sumador capaz de operar con cantidades de 7 bits, 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? Responder en hexadecimal.

NOTA: el código ASCII de la 'a' es 61h. Además, se sabe que si se realiza una operación XOR bit a bit del código ASCII de la 'Z' con el de la 'z' el resultado es 20h.

41h



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

- A) El circuito detector de *Overflow* de la CPU Elemental sólo necesita conocer, además de la operación aritmética que se realiza, el signo de cada uno de los operandos de entrada para generar el bit O del registro de estado.
- B) Los siguientes tres bytes, 4FCAA3h, expresados en hexadecimal, representan 3 caracteres codificados en formato Unicode UTF-8.
- C) La arquitectura VonNeumann define un modelo de computador con sistemas de memoria diferenciados para instrucciones y para datos de programa.
- D) En la CPU Elemental, una instrucción de salto que utiliza direccionamiento inmediato sólo podrá modificar el valor del contador de programa en el rango [PC-128, PC+127].

D

- Completa la siguiente rutina en lenguaje C que debe calcular la media de dos valores A y B. Además, B es siempre mayor o igual que A. El cálculo debe evitar la posible aparición de overflow. Ejemlos A=1 y B=1 Res=1, A=2 y B=4 Res=3, A=2 y B=3 Res=2 ó 3. Sólo se puede utilizar operaciones enteras. Recordad que 1/2 es cero.

```
int PuntoMedio(int A, int B)
{
    return A + (B - A)/2;
}
```

Durante la ejecución de una de las instrucciones del siguiente fragmento de código en la CPU Elemental,

```
MOVH R0, 0
MOVL R0, 16
Bucle: MOV R2, [R1]
        ADD R4, R4, R2
        INC R1
        DEC R0
        BRNZ Bucle
```

los registros de la CPU contienen los siguientes valores:

R0	000A	R4	001A
R1	7030	R5	0000
R2	0034	R6	1FFA
R3	0000	R7	7422
PC	7141	IR	4488

- Si la instrucción en ejecución acaba de finalizar su paso 3, ¿cuál será el próximo valor que aparecerá en el registro TMPS? Responder en hexadecimal.

004Eh

- ¿Qué valor hay almacenado en la dirección de memoria 713Eh? Responder en hexadecimal.

2010h



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

- A) El rango de números enteros representable en formato exceso a Z central, coincide con el rango de números enteros representable en formato complemento a 2.
- B) La Unidad de Control de la CPU Elemental está gobernada por una señal de reloj de forma que cada paso de ejecución corresponde a un período de dicha señal.
- C) La rutina de servicio de una interrupción puede recibir parámetros a través de la pila.
- D) La resta del código ASCII de las letras "m" y "e" produce el mismo resultado que la resta del código ASCII de las letras "M" y "E".

A, B, D

Se ha realizado un programa en ensamblador de la CPU elemental que se utiliza para dibujar sobre la pantalla. El programa tiene tres listas en la sección de datos que contienen las posiciones en el eje X (ListaX) e Y (ListaY) de los caracteres a representar (ListaC). Por ejemplo el primer elemento de las tres listas indica que en la posición (0, 5) se debe representar el carácter '/'. Las tres listas tienen el mismo número de elementos y terminan en -1. Además se ha instalado una rutina de interrupción para que cuando se pulse una tecla se cambie el color del texto con el que se realiza el dibujo. Tras terminar de dibujar la información contenida en las tres listas, se borra la pantalla y se comienza de nuevo.

<p>– ¿Cuál es el color de fondo con el que se realiza el dibujo sobre la pantalla? Responde “Cambia” si consideras que el color de fondo varía durante la ejecución del programa dependiendo de las teclas pulsada. Los colores que representan cada tripleta RGB son lo siguientes:</p> <p>000 = Negro, 001 = Azul, 010 = Verde, 011 = Amarillo, 100 = Rojo, 101 = Cian, 110 = Gris, 111 = Blanco</p> <div>Negro</div> <p>– Tras pulsar 9 teclas, y una vez fuera de la rutina de interrupción, ¿Qué valor habrá almacenado en la dirección de memoria referenciada por la etiqueta de datos <code>AtributoColor?</code> Responder en hexadecimal.</p> <div>0300h</div> <p>– ¿Qué TRES instrucciones faltan en el hueco 2?</p> <div> <div>MOV R5, [R3]</div> <div>COMP R4, R5</div> <div>BRZ comienza</div> </div> <p>– ¿Cual es el número de posiciones de memoria en la pila que necesita la rutina de interrupción para poder ejecutarse?</p> <div>7</div>	<p>– Completa la implementación del procedimiento <code>multiplica</code> que se muestra a continuación</p> <div> <pre> PROCEDIMIENTO Multiplica ; Multiplica dos valores x*y ; Recibe dos paramatros a traves de la pila: ; el valor de X, el valor de Y ; Deja en resultado en R0 ; Tanto X como Y son numeros naturales ; Y es mayor que 0 PUSH R6 MOV R6, R7 PUSH R1 PUSH R2 INC R6 ; Acceso a los parametros INC R6 MOV R1, [R6] ; El valor x se pone en R1 INC R6 MOV R2, [R6] ; El valor y se pone en R2 XOR R0, R0, R0 ; Incializacion del acumulador sigue_sumando: ; Comienzo del hueco ----- ADD R0, R0, R1 DEC R2 BRNZ sigue_sumando ; Fin del hueco ----- POP R2 ; Se desapilan los registros POP R1 POP R6 RET FINP </pre> </div> <p>– Sabiendo que la pantalla tiene 8 filas y 15 columnas, ¿Cuál es la posición de memoria en la que se mapea el registro de control de la pantalla? Responder en hexadecimal.</p> <div>F078h</div>	<p>– Sabiendo que la instrucción <code>CALL PintaCaracterXY</code> está almacenada en la dirección 659h, y que su codificación es D09Ch, ¿Cuál es la dirección donde se almacena la instrucción <code>CALL CalculaDireccion?</code> En el hueco 1 hay cuatro instrucciones. Responder en hexadecimal.</p> <div>0608h</div> <p>– ¿Qué CUATRO instrucciones faltan en el hueco 1?</p> <div> <div>INC R6</div> <div>INC R6</div> <div>MOV R0, [R6] ; Posición X de la pantalla</div> <div>INC R6</div> </div> <p>– Durante la ejecución de la primera instrucción del programa se pulsa una tecla, ¿Cuál es el valor almacenado en la dirección de memoria apuntada por R7 justo después de ejecutar la primera instrucción de la rutina de interrupción por primera vez?</p> <div>3</div>
--	---	---



```
BASE_PANTALLA EQU 0F000h
BASE_TECLADO EQU 0F100h
INTE_TECLADO EQU 0003h

ORIGEN 500h
INICIO ini
.PILA 20h
.DATOS
ListaX VALOR 0, 0, 0,... ..., -1
ListaY VALOR 5, 6, 11,... ..., -1
ListaC VALOR '/', ';', ';',... ..., -1
AtributoColor VALOR 0100h

.CODIGO
PROCEDIMIENTO Multiplica
; Multiplica dos valores X*Y. Recibe dos paramatros
; a traves de la pila: el valor de X, el valor de Y
; Deja en resultado en R0
...
FINP

PROCEDIMIENTO CalculaDireccion
; Devuelve en R0 la dirección de la coordenada
; (X, Y) a partir de la dirección base.
; X, Y y DirBase son recibidos por la pila
PUSH R6
MOV R6, R7
PUSH R1
PUSH R2
PUSH R3

INC R6
INC R6
MOV R0, [R6] ; Posición X de la pantalla
INC R6
MOV R1, [R6] ; Posición Y de la pantalla
INC R6
MOV R2, [R6] ; Dirección base

; Número de columnas de la pantalla
MOVH R3, 0
MOVL R3, 15

PUSH R3
PUSH R0
CALL Multiplica
INC R7
INC R7
ADD R0, R0, R1
ADD R0, R0, R2

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

```
PROCEDIMIENTO PintaCaracterXY
; Pinta un carácter en la coordenada (X, Y)
; El carácter, Y, y X se reciben por la pila
PUSH R6
MOV R6, R7
PUSH R0
PUSH R1
PUSH R2
PUSH R3
```

<< HUECO 1 >>

```
MOV R1, [R6] ; Posición Y de la pantalla
INC R6
MOV R2, [R6] ; Caracter a pintar
```

```
MOVH R3, BYTEALTO BASE_PANTALLA
MOVL R3, BYTEBAJO BASE_PANTALLA
```

```
PUSH R3
PUSH R1
PUSH R0
CALL CalculaDireccion
INC R7
INC R7
INC R7
```

```
MOV [R0], R2
```

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

FINP

PROCEDIMIENTO BorraPantalla

...

FINP

PROCEDIMIENTO RutinaInt

```
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
```

```
MOVH R1, BYTEALTO BASE_TECLADO
MOVL R1, BYTEBAJO BASE_TECLADO
MOV R0, [R1]
```

```
MOVH R2, BYTEALTO DIRECCION AtributoColor
MOVL R2, BYTEBAJO DIRECCION AtributoColor
MOVL R4, 0
MOVH R4, 7H
MOV R3, [R2]
COMP R3, R4
BRZ reinicia_color
```



```
XOR R4, R4, R4
MOVH R4, 1
ADD R3, R4, R3
JMP continua
reinicia_color:
MOVH R3, 1
continua:
MOV [R2], R3

POP R4
POP R3
POP R2
POP R1
POP R0
IRET

FINP

ini:
MOVH R0, BYTEALTO INTE_TECLADO
MOVL R0, BYTEBAJO INTE_TECLADO
MOVH R1, BYTEALTO DIRECCION RutinaInt
MOVL R1, BYTEBAJO DIRECCION RutinaInt
MOV [R0], R1
STI

MOVH R4, 0FFh
MOVL R4, 0FFh ; -1
comienza:
CALL BorraPantalla
MOVH R1, BYTEALTO DIRECCION ListaX
MOVL R1, BYTEBAJO DIRECCION ListaX
MOVH R2, BYTEALTO DIRECCION ListaY
MOVL R2, BYTEBAJO DIRECCION ListaY
MOVH R3, BYTEALTO DIRECCION ListaC
MOVL R3, BYTEBAJO DIRECCION ListaC
sigue_pintando:
```

<< HUECO 2 >>

```
MOVH R6, BYTEALTO DIRECCION AtributoColor
MOVL R6, BYTEBAJO DIRECCION AtributoColor
MOV R6, [R6]
OR R5, R5, R6
PUSH R5
MOV R5, [R2]
PUSH R5
MOV R5, [R1]
PUSH R5
```

```
CALL PintaCaracterXY
INC R7
INC R7
INC R7
```

```
INC R1
INC R2
INC R3
```

JMP sigue_pintando

FIN