Apellidos _____ **Nombre** _____ **DNI** _____ **A**

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Final de Junio 25-06-2002**

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.

— Los dígitos hexadecimales C3414000h representan un número codificado en el formato de coma flotante IEEE-(P)754 de precisión simple. Indica a qué cantidad decimal corresponde.

-193,25

— Se dispone de un formato de coma flotante con las siguientes características:

Mantisa: Formato signo-magnitud y normalizada de la misma forma que en el formato IEEE-(P)754

Exponente: Formato exceso a Z central

Se desea saber cuántos bits debe tener el exponente para que el número 256 sea representable en este formato

5

— Se sabe que en el registro R0 de la CPU elemental se encuentra almacenado el código ASCII de un dígito numérico, es decir, un carácter entre el '0' y el '9'. Escribe una secuencia de tres instrucciones que transforme el código ASCII del dígito en el valor que representa, quedando dicho valor almacenado en R0. Utiliza como registro auxiliar R1.

```
MOVL R1, 0Fh
MOVH R1, 0
AND R0, R0, R1
```

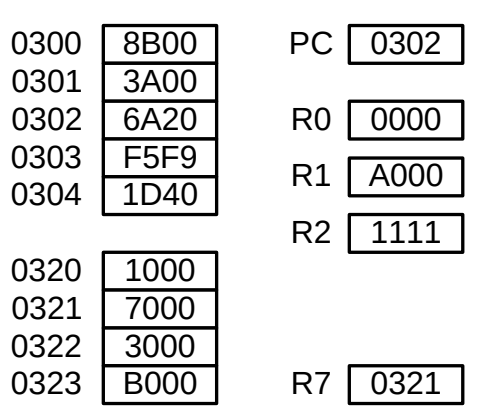
— Se pretende diseñar una nueva instrucción para la CPU elemental, cuyo objetivo es cargar una dirección en un vector de interrupción. La instrucción trabajará sobre dos registros. El registro fuente contendrá la dirección que se va a cargar en el vector, y el registro destino, el número de vector que se va a modificar. El mnemónico elegido para esta instrucción es CVI (Carga Vector de Interrupción). Escribe la secuencia de señales de control que llevará a cabo la CPU para ejecutar CVI R0, R1.

Paso	Acciones de control
4	R0-IB, IB-MAR
5	R1-IB, IB-MDR, WRITE
6	FIN

— Escribe la secuencia de señales de control necesarias para llevar a cabo la instrucción RET

Paso	Acciones de control
4	R7-IB, IB-MAR, READ, TMPE-CLR CARRY_IN, ADD, ALU-TMPS
5	TMPS-IB, IB-R7
6	MDR-IB, IB-PC, FIN

En la figura adjunta se muestra el estado de algunos registros y de un conjunto de posiciones de memoria del computador elemental justo cuando acaba de ejecutarse el **paso 4º** de una instrucción.



A partir de la información de la figura contesta a las preguntas A, B y C.

— A) ¿cuál será el contenido del registro MAR en el instante representado en la figura?

0321

— B) Determina el valor de los bits del registro de estado justo después de la ejecución de la instrucción 6A20 (almacenada en la dirección 0302).

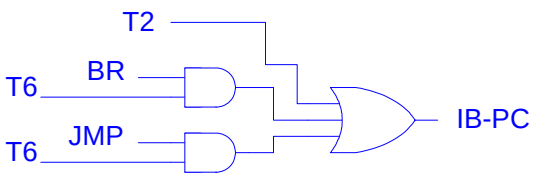
CF = 1 OF = 1 SF = 1 ZF = 0

— C) ¿Qué valor contendrá el BUS interno durante la ejecución del paso 5º de la instrucción F5F9 (almacenada en la dirección 0303)? Contesta con 4 dígitos hexadecimales.

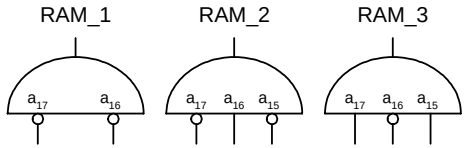
FFF9

— Diseña el circuito combinacional necesario para generar la señal de control IB-PC de la CPU elemental. Se supone que dicho circuito combinacional formará parte de la unidad de control cableada de la CPU. Para hacer el diseño utiliza

puertas OR y AND de las entradas que necesites. Utiliza también las entradas que consideres oportunas entre las que se proporcionan a continuación: T1, T2, T3, T4, T5, T6, ADD, INC, BR, JMP, PUSH y POP. (Las entradas T1... T6 son generadas por el contador de pasos y el resto por el decodificador de instrucciones)



Un computador tiene un bus de direcciones de 18 líneas y un bus de datos de 8 líneas. Se pretende conectar a este computador tres dispositivos de memoria RAM, denominados RAM_1, RAM_2 y RAM_3. En la figura siguiente se muestran los circuitos de activación utilizados para ubicar estos dispositivos de RAM en diferentes regiones del espacio de direcciones.



La organización interna de los dispositivos es la siguiente:

RAM_1: 8 bancos de 8 chips

RAM_2: 2 bancos de 4 chips

RAM_3: 8 bancos de 2 chips

Teniendo en cuenta toda esta información contesta a las preguntas A, B y C.

— A) Indica los rangos de direcciones cubiertos por los dispositivos RAM_1 y RAM_3. (Ejemplo de solución para indicar un rango: 30000h ─ 3FFFFh)

Rango RAM_1: 00000h ─ 0FFFFh

Rango RAM_3: 28000h ─ 2FFFFh

— B) Determina el número de líneas de dirección y de datos de los chips que forman el dispositivo RAM_3

Línea de dirección: 12 Líneas de datos: 4

— C) Si la CPU emite la dirección 1343Ah, ¿qué dispositivo de memoria RAM y qué banco serán accedidos? (Ejemplo de contestación: RAM_1, banco 6).

RAM_2, banco 0

— Determina cuál o cuáles de las siguientes afirmaciones son CIERTAS. En caso de que ninguna sea cierta, contesta "ninguna".

A) Cuando la CPU elemental recibe una interrupción, tarda como mínimo tres pasos de ejecución en atenderla.
B) Siempre que el interfaz de un periférico activa la señal INT, después la CPU activa la señal INTA.
C) Durante la ejecución de una instrucción IRET el registro R7 se incrementa en 2.
D) Durante el proceso de atención a una interrupción, la CPU recibe un número de vector de interrupción a través del registro MDR

C, D

— Determina cuál o cuáles de las siguientes afirmaciones son CIERTAS. En caso de que ninguna sea cierta, contesta "ninguna".

A) El número positivo más pequeño que se puede representar en el formato IEEE(P)-754 normalizado es el 2^{-126}.
B) En un formato de enteros en exceso a Z con n=4 y Z=5, el entero más grande representable es 7.
C) En un formato de complemento a 2 necesitamos utilizar como mínimo 10 bits para que el 512 sea representable.
D) En el formato YE^2 el número 126,1 puede representarse con un error menor que 0,25

A, D

Se ha conectado una cámara digital al computador elemental. La cámara tiene un registro de control de 16 bits en el que se indica, en milisegundos, cada cuanto tiempo se toma una foto. Este registro está mapeado en la dirección B000h. En la dirección B001h se mapea otro registro de control que sirve para indicar, en sus cuatro bits de menor peso, a qué resolución captura la cámara entre 16 posibles resoluciones. Los cuatro bits siguientes sirven para indicar el número de colores empleados. Ambos registros de control son de escritura y lectura. Teniendo en cuenta esta información contesta a las preguntas Ay B.

— A) Escribe las instrucciones necesarias para configurar la cámara para que capture imágenes cada 25 milisegundos. Utiliza como registros auxiliares R1 y R2.

```
MOVL R1, 0
MOVH R1, 0B0h
MOVL R2, 19h
MOVH R2, 0
MOV [R1], R2
```

— B) Escribe las instrucciones necesarias para almacenar en R5 la resolución a la que está trabajando la cámara. Además de R5, utiliza como registros auxiliares R3 y R4.

```
MOVL R3, 15
MOVH R3, 0
MOVL R4, 01h
MOVH R4, 0B0h
MOV R5, [R4]
AND R5, R5, R3
```

A continuación se muestra el listado de un programa cuyo objetivo es procesar una cadena de caracteres y enviarlas a la pantalla. Para ello el programa utiliza dos procedimientos, denominados *transforma* e *imprime*. A continuación se describe brevemente cada uno de ellos, así como el programa principal.

Procedimiento *transforma*

El objetivo del procedimiento *transforma* es procesar una cadena de caracteres, que debe estar finalizada por un determinado carácter que hace de terminador de cadena. *transforma* Recibe tres parámetros a través de la pila en el orden que se indica a continuación: 1) el terminador de cadena, 2) la dirección de memoria en la que se encuentra la cadena a procesar, y 3) la dirección de memoria a partir de la cual se almacenará la nueva cadena obtenida después del procesamiento. El parámetro 1 quiere decir el primero en apilarse y así sucesivamente.

El procedimiento *transforma* procesa uno a uno los caracteres de la cadena cuya dirección se le pasa como parámetro, de modo que si el carácter a procesar es una letra minúscula lo deja como está, pero si es cualquier otro carácter lo sustituye por un '*'. Cada carácter, una vez procesado, es copiado en una nueva ubicación, cuya dirección de comienzo ha sido pasada al procedimiento en el parámetro 3.

Procedimiento *imprime*

El objetivo de este procedimiento es imprimir una cadena de caracteres en la pantalla del computador elemental. La cadena a imprimir debe estar finalizada con un carácter que haga de terminador. Este procedimiento recibe a través de la pila tres parámetros: 1) el terminador de cadena, 2) la posición de la pantalla a partir de la que se desea imprimir la cadena (el 0 indica la esquina superior izquierda, los números siguientes indican posiciones sucesivas desplazándonos de izquierda a derecha y de arriba abajo), y 3) la dirección de la cadena a imprimir.

La pantalla del computador elemental se mapea a partir de la dirección E000h.

Programa principal

Llama al procedimiento *transforma* para procesar la *cadena1* almacenada en la sección de datos, y dejar el resultado en *cadena2*. Después llama al procedimiento *imprime* para escribir *cadena2* en pantalla.

Información adicional

Dirección etiqueta *ini*: 1084h
Código ASCII de la 'a' = 61h

Teniendo en cuenta toda esta información y los comentarios presentes en el programa contesta a las siguientes preguntas:

— Indica las instrucciones que son necesarias en el recuadro 1.

```
INC R6
INC R6
MOV R0, [R6]
```

— Indica la instrucción o instrucciones que son necesarias en el recuadro 2.

```
BRNZ no_es_minuscula
JMP es_minuscula
```

— Indica las instrucciones que son necesarias en el recuadro 3.

```
MOVL R2, 00h
MOVH R2, 0E0h
ADD  R2, R2, R1
```

— Determina el contenido de la dirección de memoria 1012h durante la ejecución del programa. Contesta en hexadecimal.

0064h

— Determina el valor de R7 justo cuando termine de ejecutarse la instrucción *call transforma*.

10AAh

— Determina el contenido de la dirección de memoria 1087h durante la ejecución del programa

2000h

— Determina el valor existente en el bus interno de la CPU, durante la ejecución del paso 4 de la instrucción marcada con '←←←', cuando esta instrucción se ejecuta por última vez.

1035h

— Si durante la ejecución del segundo paso de la instrucción *call imprime* se produce una interrupción, y ésta es aceptada por la CPU, ¿cuántos pasos se ejecutarán hasta que una nueva instrucción sea cargada en el registro de instrucciones de la CPU? (Nota: no contar el paso 2). Contesta en decimal.

1 ó 19

```
ORIGEN 1000h
INICIO ini
.PILA  10h
.DATOS
cadena1  VALOR  "ejemplo-de-11,0-cade(ER)na$"
cadena2  VALOR 30 VECES 0

.CODIGO
PROCEDIMIENTO transforma
  push r6
  mov  r6, r7
  push r0
  push r1
  push r2
  push r3
  push r4
  push r5

  ; Cargar el tercer parametro en R0
```

Instrucciones Ocultas

```
  ; Cargar el segundo parametro en R1
```

Instrucciones Ocultas

```
  ; Cargar el primer parametro en R2
```

Instrucciones Ocultas

```
  ; Cargar R4 y R5 con la 'a' y la 'z'
  movl r4, 'a'
  movh r4, 0
  movl r5, 'z'
  movh r5, 0

  ; Bucle de procesamiento
bucle1:
  mov  r3, [r1]
  comp r3, r2
  brz  fuera1

  ; Cambiar caracter por '*' si no es una
  ; minuscula
  comp r3, r4
  ; Salta si R3 < R4  (CF=1)
  brc no_es_minuscula

  comp r3, r5
  ; Salta si R3 > R5 (CF=0 and ZF=0)
  brnc mira_ZF
  jmp  es_minuscula
mira_ZF:
```

2

```
no_es_minuscula:
  movl r3, '*'

es_minuscula:
  mov  [r0], r3
  inc  r0
  inc  r1
  jmp  bucle1

fuera1:
  ; Copiar terminador de cadena
  mov  [r0], r3

  pop  r5
  pop  r4
  pop  r3
  pop  r2
  pop  r1
  pop  r0
  pop  r6
  ret
FINP

PROCEDIMIENTO imprime
  push r6
  mov  r6, r7
  push r0
  push r1
  push r2
  push r3
  push r4

  ; Cargar el tercer parametro en R0
```

1

```
  ; Cargar el segundo parametro en R1
```

Instrucciones Ocultas

```
  ; Cargar el primer parametro en R3
```

Instrucciones Ocultas

```
  ; Preparar R2 para apuntar a la posicion
  ; requerida de la memoria de video
```

3

```
  ; Bucle de procesamiento
bucle2:
  mov  r4, [r0] ←←←
  comp r4, r3
  brz  fuera2
  ; Cargar atributo
  movh r4, 7

  mov  [r2], r4

  inc  r0
  inc  r2
  jmp  bucle2

fuera2:
  pop  r4
  pop  r3
  pop  r2
  pop  r1
  pop  r0
  pop  r6
  ret

FINP

ini:
  ; Paso de parametros y llamada al procedimiento
  ; transforma
  movl r0, '$'
  movh r0, 0
  push r0

  movl r0, BYTEBAJO DIRECCION cadena1
  movh r0, BYTEALTO DIRECCION cadena1
  push r0

  movl r0, BYTEBAJO DIRECCION cadena2
  movh r0, BYTEALTO DIRECCION cadena2
  push r0

  call transforma

  inc r7
  inc r7
  inc r7

  ; Paso de parametros y llamada al procedimiento
  ; imprime
  movl r0, '$'
  movh r0, 0
  push r0

  movl r0, 16
  movh r0, 0
  push r0

  movl r0, BYTEBAJO DIRECCION cadena2
  movh r0, BYTEALTO DIRECCION cadena2
  push r0

  call imprime

  inc r7
  inc r7
  inc r7

FIN
```