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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Septiembre: 2004**

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.

Se ha programado un procedimiento en ensamblador de Intel x86 que recibe una cadena y devuelve otra que es igual a la primera pero insertando un espacio entre cada uno de sus caracteres. Por ejemplo:

Entrada: "Hola, mundo"

Salida: "H o l a , m u n d o"

El procedimiento recibe por la pila como primer parámetro la dirección de la cadena a transformar y como segundo la dirección donde se debe dejar el resultado. Se supone que hay suficiente memoria reservada para esta segunda cadena. El final de las cadenas se indica con el ASCII cero.

A continuación se muestra el código del procedimiento:

```
inserta_espacios PROC
  PUSH EBP
  MOV EBP, ESP
  PUSH EAX
  PUSH EBX
  PUSH ECX
  PUSH EDX

  ; Sacar de la pila la dir. de la cad.
  ; de origen a EAX y la de destino a EBX
  ;--1--

  MOV CL, ' ' ; CL = espacio en blanco

bucle:
  MOV DL, [EAX] ; Traer el carácter actual

  ; Comprobar si se ha llegado al final
  ; y en ese caso salir
  ;--2--

  MOV [EBX], DL ; Copiar el carácter actual
  INC EBX ; Sgte. car. en la cad. destino
  MOV [EBX], CL ; Meter el espacio
  INC EAX ; Sgte. car. en la cad. origen
  INC EBX ; Sgte. car. en la cad. destino
  JMP bucle

  MOV EBX, 0 ; Finalizar cad. destino

salir:
  POP EDX
  POP ECX
  POP EBX
  POP EAX
  POP EBP
  RET 8
ENDP
```

Este procedimiento se utiliza dentro de un programa. Se sabe que en una de las ejecuciones de este programa, justo antes de ejecutar el PUSH EAX del procedimiento, EBP vale 0063FE28h. Además se tienen las siguientes dos capturas en ese instante del panel de datos del Turbo Debugger:

```
:0063FE20 67 01 00 00 67 01 00 00 g☺  g☺
:0063FE28 78 FF 63 00 1B 10 40 00 x c ←▸@
:0063FE30 1A 20 40 00 09 20 40 00 → @ ○ @
:0063FE38 05 10 40 00 37 B5 F8 BF ♣▸@ 7╡°┐
:0063FE40 00 00 00 00 B0 AE 94 81     ░«öü
```

```
:00402000 6B 65 6D 63 69 77 6E 6B kemciwnk
:00402008 61 77 64 6B 66 69 65 6E awdkfien
:00402010 63 71 69 64 38 33 6D 64 cqid83md
:00402018 6A 00 6C 61 6B 6C 64 66 j lakldf
:00402020 69 63 6D 77 65 69 69 6A icmweiij
```

– ¿En qué rango de direcciones se encuentra la codificación de la instrucción CALL que se ejecutó para llamar al procedimiento? Recordar que la codificación de un CALL ocupa 5 bytes. Ejemplo de respuesta: 3B44F230h-3B44F234h.

00401016h-0040101Ah

– ¿Cuál es el primer carácter de la cadena de origen? Ejemplo de respuesta: t.

w

– ¿Qué instrucción o instrucciones falta/n en **-- 1 --**?

MOV EAX, [EBP+12]

MOV EBX, [EBP+8]

– ¿Qué 2 instrucciones faltan en **-- 2 --**?

CMP DL, 0

JE/JZ salir

– ¿Cuál es la representación del número –2.1 en formato IEEE-754? **Contestar en hexadecimal.**

C006 6666h

– Se tiene un formato para números positivos de coma fija con 4 bits para la parte entera y 4 bits para la parte fraccionaria. ¿Cuál es el error cometido al representar el número 2.1 en este formato? **Contestar en decimal.**

0.0375

– En una ALU de 3 bits, análoga a la ALU de la CPU teórica, se introduce por el operando A el número –1 codificado en complemento a 2 y por el operando B el número más grande que se puede representar en exceso a 4. ¿Cuál es el valor de los bits de estado al realizar la suma?

Z: 0 C: 1 O: 0 S: 1

– Se ha añadido a la CPU teórica una unidad de desplazamiento (UD). Esta unidad consiste en un registro llamado UDN de 16 bits para el número al que hay que aplicar el desplazamiento, un registro llamado UDD de 8 bits para el número de bits a desplazar (codificado en complemento a 2) y un circuito lógico que realiza el desplazamiento. Tiene las siguientes señales de control:

- IB-UDN: Toma el contenido de IB y lo introduce en el registro UDN.
- UDN-IB: Vuelca el contenido del registro UDN a IB.
- IBl-UDD: Introduce la parte baja del bus interno en el registro UDD.
- UDD-IBl: Vuelca el registro UDD a la parte baja de IB.
- DESP: Realiza el desplazamiento.

Para realizar un desplazamiento hay que cargar los dos registros y activar la señal DESP. El desplazamiento se efectúa en el mismo ciclo que se activa la señal DESP. En el siguiente ciclo ya está disponible en el UDN el número con la operación de desplazamiento realizada.

Basándose en la funcionalidad anterior, se ha diseñado una nueva instrucción para realizar desplazamientos con el mnemónico **DES Rd/s, num**, que realiza el desplazamiento de **num** bits (a la izquierda si **num** es positivo y a la derecha si es negativo) del contenido del registro **Rd/s**. El número **num** se codifica en los 8 bits más bajos de la instrucción.

— Completar los pasos con las señales de control para esta instrucción teniendo en cuenta que el paso 6 ya está completo y que puede tener cualquier número de pasos entre 6 y 8.

Paso	Señales de Control
4	Rd/s-IB, IB-UDN
5	RIl-IBl, IBl-UDD
6	DESP
7	UDN-IB, IB-Rd/s, FIN
8	

Se están probando dos CPUs (A y B), que tienen idéntica arquitectura y juego de instrucciones. Se diferencian sólo en dos cosas:

1) La frecuencia de su reloj. La CPU A tiene una frecuencia de reloj de 2 KHz, mientras que la de la CPU B es de 2,5 KHz.

2) Las instrucciones de la CPU B requieren un ciclo de reloj más que las de la CPU A.

Se ha medido que un fragmento de código con 5000 instrucciones tarda 10 segundos en ejecutarse en la CPU A.

— ¿Cuánto tardará en la CPU B? **Incluir unidades de tiempo en la respuesta.**

10 segundos

– Si se hiciese una unidad de control cableada para la CPU teórica, ¿cuál sería la señal de control (de aquellas que se activan en los **pasos 4 en adelante**) que requeriría más puertas AND para ser generada?

FIN

— ¿Cuáles de las siguientes afirmaciones son **FALSAS**? (Puedes responder "ninguna" si así lo consideras.)

a) Con 64 bits, la cantidad de números representables en complemento a 2 es mayor que en exceso a Z central.

b) En la ALU de la CPU teórica se realiza la resta a partir de una suma y de la transformación de uno de los operandos.

c) En la arquitectura Von Neumann hay dos memorias separadas: la memoria principal para los datos y la memoria de microprograma para los programas.

d) El ancho de la memoria de microprograma debe coincidir con el ancho de los registros de la CPU.

A, C, D

— ¿Cuáles de las siguientes afirmaciones son **FALSAS**? (Puedes responder "ninguna" si así lo consideras.)

a) Si en un dispositivo de memoria se dobla el ancho de palabra de los chips que utiliza, se deberá reducir a la mitad el número de bancos para que no varíe la organización del dispositivo.

b) Tras ejecutar IRET en la CPU teórica al final de una rutina de servicio de interrupción correctamente implementada, el flag IF siempre se pone a 1.

c) Los datos de las instrucciones de Intel se almacena en memoria principal en formato *big endian.*

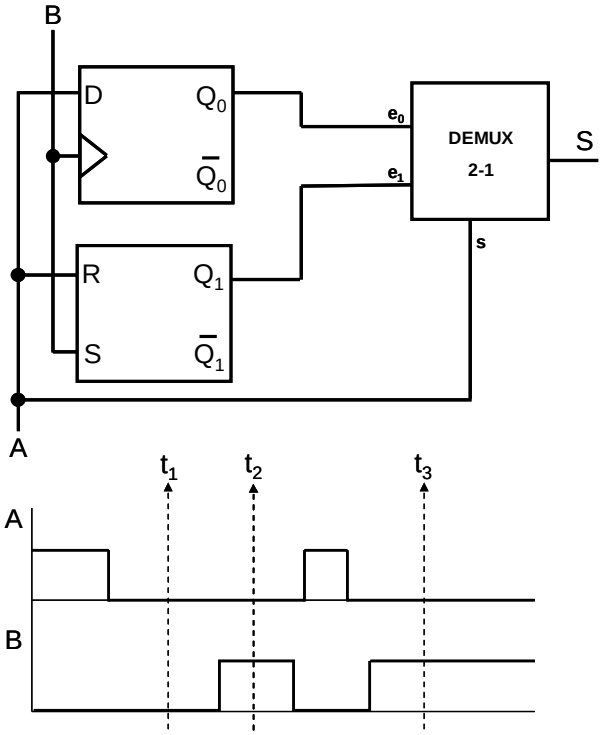
d) Si EAX contiene 8B039582h y EBX contiene A0000000h, al sumarlos se activará el bit de carry.

A, C

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

Examen de Fundamentos de los computadores. *Área de Arquitectura y Tecnología de Computadores* **Septiembre: 2004**

Dado el circuito y el cronograma mostrados a continuación:



– ¿Cuál es el valor de la salida S en los instantes t_1, t_2 y t_3 teniendo en cuenta de que el valor inicial de Q_0 es 1?

t_1: 1 t_2: 0 t_3: 0

– Expresa, utilizando la notación basada en minterms, la fórmula booleana que describe un circuito combinacional que indica si una palabra de 3 bits es par. Este circuito proporciona un valor '1' en su única salida S cuando la palabra de 3 bits de la entrada es un valor par (interpretando la cantidad en binario natural). NOTA: el cero es par. **Ejemplo de respuesta: $S=m_0+m_4+m_7$**

$S = m_0+m_2+m_4+m_6$

El siguiente programa para la CPU teórica se encarga de mostrar en el periférico **pantalla** los datos de personas almacenadas en una "base de datos" en memoria. La "base de datos" contiene los datos de tres personas y estos datos son su nombre y su primer apellido. Existe en la sección de datos del programa una lista de 3 nombres y otra lista de 3 apellidos. Al pulsar las teclas 1, 2 ó 3, aparece en la primera línea de la pantalla el nombre y apellido de la persona seleccionada. En el programa, cuyo listado aparece a continuación, las tres personas cuyos datos están almacenados en la base de datos son Pepe Perez, Lolo Gomez y Juan Lopez. Tanto la lista con los tres nombres como la lista con los tres apellidos tienen la misma estructura: son cadenas de caracteres terminadas con el código ASCII cero. En el caso de los nombres, por simplicidad, las cadenas tienen siempre cuatro caracteres de longitud; en el caso de los apellidos, de la misma forma, la longitud de los mismos es de cinco caracteres.

El programa consta de un procedimiento principal (cuyo comienzo está marcado con la etiqueta *comienzo*), de una rutina de servicio de interrupción del **teclado** llamada ***rutina_teclado*** y de un procedimiento llamado ***imprime_cadena***, que es llamado por *rutina_teclado*.

- El **programa principal** simplemente instala la rutina de servicio de interrupción de teclado.
- **rutina_teclado** lee la tecla pulsada y muestra el nombre y apellido de la primera, segunda o tercera persona en el caso de que se hayan pulsado las teclas 1, 2 ó 3 (respectivamente) del teclado alfanumérico. Si se pulsase otra tecla, no haría nada. Además, procesa todas las pulsaciones pendientes en el buffer de teclado (esto es, si al servir una interrupción existiese más de una pulsación en el buffer, trataría todas las pulsaciones).
- El procedimiento **imprime cadena** recibe tres parámetros a través de la pila. En el orden en que son apilados antes de la llamada al procedimiento, estos parámetros son: 1) dirección de la cadena a imprimir; 2) longitud de la cadena a imprimir; 3) posición de la memoria de video a partir de la cual imprimir la cadena.

A continuación se muestra el código fuente del programa en ensamblador, al cual le faltan algunas instrucciones.

```
ORIGEN 1000h
INICIO comienzo
.PILA 30
.DATOS
lista_nombres
       VALOR "Pepe",0,"Lolo",0,"Juan",0
lista_apellidos
       VALOR "Perez",0,"Gomez",0,"Lopez",0

.CODIGO
PROCEDIMIENTO rut_teclado
  PUSH R0
  PUSH R1
  PUSH R2
  PUSH R3
  PUSH R4

  MOVL R1, 1h    ; R1 = reg. control tecl.
  MOVH R1, 0FEh
  MOVL R2, 0     ; R2 = máscara deteccion
  MOVH R2, 1     ;  pulsaciones pendientes
  MOVH R4, 0   ; Codigo ASCII tecla 1 del
  MOVL R4, 49  ; teclado alfanumerico
  MOVH R0, 0   ; Codigo ASCII carácter cero
  MOVL R0, '0'

bucle:   ; Repetir mientras haya teclas
  MOV R3, [R1]
  AND R3, R3, R2
  BRZ no_hay_tecla

  DEC R1       ; R1 = reg. de datos
  MOV R3, [R1] ; R3 = tecla pulsada
  INC R1       ; R1 = reg. de control

  MOVH R3, 0   ; Quitar cod. SCAN
  ; Ver si se pulsaron las teclas 1, 2 o 3
  COMP R3, R4
  BRZ imprimir
  INC R4
  COMP R3, R4
  BRZ imprimir
  INC R4
  COMP R3, R4
  BRZ imprimir
  JMP bucle
```

```
imprimir:
; Elegir primero el nombre a imprimir
  MOV R2, R3
  SUB R2, R2, R0 ;R2=Nº de nombre a imprimir
  MOVH R4, BYTEALTO DIRECCION
lista_nombres
  MOVL R4, BYTEBAJO DIRECCION
lista_nombres
  XOR R5, R5, R5
  MOVL R5, 5  ; Desplazamiento sgte nombre

  DEC R2
  BRZ imprimir_nombre
bucle_acceso_nombre:
  ADD R4, R4, R5
  DEC R2
  BRNZ bucle_acceso_nombre

imprimir_nombre:
  PUSH R4   ; Apilar inicio cadena
  DEC R5
  PUSH R5   ; Apilar longitud cadena
  MOVH R6, 0F0h
  MOVL R6, 0
  PUSH R6   ; Apilar pos. video
  CALL imprime_cadena
   ß HUECO 1 à

; Elegir el apellido a imprimir
  MOV R2, R3
  SUB R2, R2, R0  ;R2=Nº de nombre a imprimir
  MOVH R4, BYTEALTO DIRECCION
lista_apellidos
  MOVL R4, BYTEBAJO DIRECCION
lista_apellidos
  XOR R5, R5, R5
  MOVL R5, 6  ; Desplazamiento sgte apell.
  DEC R2
  BRZ imprimir_apellido
bucle_acceso_apellido:
  ADD R4, R4, R5
  DEC R2
  BRNZ bucle_acceso_apellido

imprimir_apellido:
  PUSH R4  ; Apilar inicio cadena
  DEC R5
  PUSH R5  ; Apilar long. cadena
  ADD R6, R6, R5 ; Avanzar para imprimir
                 ; el apellido tras nombre
  PUSH R6  ; Apilar pos. video
  CALL imprime_cadena
   ß HUECO 1 à
```

```
  JMP bucle ; Por si hay mas de una
            ; pulsacion en el buffer

no_hay_tecla:
  POP R4
  POP R3
  POP R2
  POP R1
  POP R0
  IRET
FINP

PROCEDIMIENTO imprime_cadena
  PUSH R6
  MOV R6, R7
  PUSH R1
  PUSH R2
  PUSH R3
  PUSH R4

  INC R6
  INC R6
  MOV R3, [R6]  ;R3 = Pos. Mem. Video
  INC R6
  MOV R2, [R6]  ;R2 = Long. cadena
  INC R6
  MOV R1, [R6] ; R1 = Dir. cadena

bucle_impresion:
  MOV R4, [R1]  ; Traer caracter actual
  MOVH R4, 7      ; Atributo de impresion
  MOV [R3], R4  ; Imprimir caracter
  INC R1  ; Apuntar a sgte caracter
  INC R3  ; Apuntar sgte pos. video
  DEC R2 ; Queda 1 car. menos
  BRNZ bucle_impresion

  POP R4
  POP R3
  POP R2
  POP R1
   ß HUECO 3 à
  FINP

comienzo:
 CLI
 ; Instalacion en la TVI de la direccion
 ; de inicio de rutina de servicio de
tecl.
 XOR R0, R0, R0
 MOVL R0, 6
 ß HUECO 2 à
```

```
 MOV [R0], R1
 STI
 JMP -1
FIN
```

– Después de haber generado una interrupción y de recibir de la CPU la señal INTA, ¿cuál es el primer número que la interfaz del teclado envía por el bus de datos? **Contestar en decimal.**

6

– Si tras ejecutarse la instrucción JMP -1 por primera vez, se pulsa la tecla 2 del teclado alfanumérico, ¿cuál es el primer valor que se asigna a R1 dentro del procedimiento imprime_cadena la primera vez que dicho procedimiento es llamado? **Contestar en hexadecimal.**

1005h

– ¿Qué instrucción/instrucciones falta/n en el **HUECO 1**?

INC R7

INC R7

INC R7

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

MOVH R1, BYTEALTO DIRECCION rut_teclado

MOVL R1, BYTEBAJO DIRECCION rut_teclado

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

POP R6

RET

– ¿Qué tamaño mínimo debe tener la pila? **Contestar en decimal.**

16

– ¿A qué posición del espacio de direcciones de la CPU elemental habría que acceder para escribir un carácter en la esquina inferior derecha de la pantalla? **Contestar en hexadecimal.**

F077h

– ¿Cuál es la dirección base de la interfaz de teclado? ¿Y la última dirección mapeada por la interfaz? **Contestar en hexadecimal.**

Dir. Base: FE00h Última dir.: FE01h

Se desea diseñar un dispositivo de memoria que cubra la primera mitad del espacio de direcciones de la CPU teórica. El dispositivo tiene un decodificador 3:8.

– ¿De cuántos bancos se compone el dispositivo?

8

– Sabiendo que el ancho de palabra de los chips es de 1 bit, ¿cuál es la organización de los chips? (Ejemplo de formato de respuesta: 2Kx32).

4Kx16