

**Examen de Fundamentos de Computadores - Septiembre 2008**

Cada respuesta correcta suma un punto. Cada respuesta incorrecta, ilegible o vacía no suma ni resta. Para obtener la nota del examen, el total de puntos se dividirá entre el total de preguntas y se multiplicará por 10.

Se ha escrito un programa en ensamblador de la arquitectura x86-32 que genera los números de Fibonacci. Estos números, cantidades enteras y positivas, son miembros de una sucesión en la que cada número es la suma de los dos anteriores, es decir,  $F_i = F_{i-1} + F_{i-2}$ . Los primeros dos números están definidos igual a uno, es decir,  $F_1 = F_2 = 1$ . La generación de la sucesión de números se realiza mediante el procedimiento `GenerarListaF` al que se le pasan dos parámetros por la pila en el siguiente orden:

- 1) El número de elementos de la sucesión.
- 2) La dirección de la lista que los va a contener.

```
386
.MODEL flat, stdcall
ExitProcess PROTO, ExitCode:DWORD
.DATA
    ListaF DD 10 DUP (0)
.CODE
Inicio:
    PUSH 10
    PUSH OFFSET ListaF
    CALL GenerarListaF
    ; Terminar retornando al S.O.
    PUSH 0
    CALL ExitProcess
END Inicio
```

**Listado1.asm - Programa Principal**

```
GenerarListaF PROC
    PUSH ebp
    MOV ebp, esp
    PUSH eax
    PUSH ebx
    PUSH ecx
    PUSH edx
    PUSH edi
    PUSH esi
```

```
; Carga dir. de la lista de números a generar
***** HUECO 1 *****

; Carga numero de elementos a generar
***** HUECO 2 *****

    XOR esi, esi
    ; Carga primer elemento
    MOV [edi+4*esi], DWORD PTR 1h
    MOV eax, [edi+4*esi]
    INC esi
    ; Carga segundo elemento
    MOV [edi+4*esi], DWORD PTR 1h
    MOV ebx, [edi+4*esi]
    SUB ECX, 2
Bucle1:
    MOV edx, eax
    ADD eax, ebx
    MOV ebx, edx
    ; Cargar numero generado en lista.
    ***** HUECO 3 *****

    LOOP Bucle1
    POP esi
    POP edi
    POP edx
    POP ecx
    POP ebx
    POP eax
    POP ebp
    RET 8
GenerarListaF ENDP
```

**Listado2.asm – Procedimiento GenerarListaF**

- ¿Qué instrucción o instrucciones faltan en el **Hueco 1** y **Hueco 2** del procedimiento `GenerarListaF`?

**MOV edi, [ebp+8]**

**MOV ecx, [ebp+12]**

- ¿Qué instrucción o instrucciones faltan en el **Hueco 3** del procedimiento `GenerarListaF`?

**INC esi**

**MOV [edi+4\*esi], eax**

- Justamente después de llamar al procedimiento `GenerarListaF` el registro `ESP` vale `0012FFB4h`, ¿cuál es el valor inicial del puntero de pila? Responder en hexadecimal.

**0012FFC0h**

En la dirección de memoria `004010CAh` se encuentra codificada la primera instrucción del programa principal. La codificación, con un total de 10 bytes, es la siguiente:

`C7 05 DirListaF(4 bytes) OperandoInm(4 bytes)`

- ¿Cuál es el valor que contendrá la dirección `004010D0h`? Responder en hexadecimal.

**01 h**

- ¿Cuál es la máxima diferencia entre dos números positivos consecutivos en el formato de coma flotante IEEE-754? (Responder en forma de potencia de dos).

**$2^{104}$**

- En la CPU teórica, el valor que se encuentra en el `IB` es `0001h`, se activan las señales `TMPE_SET`, `ADD` y `ALU_SR`. ¿Qué valor tiene el registro `ZCOS`?

**Z=1 C=1 O=0 S=0**

**Pig Latin** es un juego de idioma usado principalmente en inglés. El **Pig Latin** lo usan los niños para divertirse o para conversar secretamente sobre adultos u otros niños. Se ha escrito un programa en ensamblador de la CPU elemental al que se le introduce la palabra por un teclado y nos muestra en pantalla la palabra traducida a **Pig Latin**. Consideraremos una versión simplificada para el castellano. Palabras que comienzan por los sonidos vocales “a”, “e”, “o”. En este caso se añade al final de la palabra original la letra “E”. (AMO → AMOE)

Las palabras se introducen por teclado indistintamente en mayúsculas o minúsculas. El procedimiento LeerPalabra recibe como parámetros la dirección de la palabra a codificar y la dirección de la posición de memoria de la longitud de dicha palabra.

El procedimiento ProcesarPalabra identifica el tipo de palabra y aplica el algoritmo de codificación. El procedimiento EscribirPalabra recibe la longitud de la palabra codificada, la dirección de la lista y la dirección donde comienza a escribir en pantalla. El periférico pantalla es un *display* de una sola fila y 15 columnas. El *display* se mapea en la dirección FA10h. Datos:

- ASCII de la letra “e”: 45.
- ASCII de la letra “E”: 65.

```
ORIGEN 500h
INICIO ini
.PILA 20h
.DATOS
Palabra VALOR 10 VECES 0
Alabrape VALOR 10 VECES 0
Longitud VALOR 0
.CODIGO
ini:
MOVL R0, BYTEBAJO DIRECCION Palabra
MOVH R0, BYTEALTO DIRECCION Palabra
MOVL R1, BYTEBAJO DIRECCION Alabrape
MOVH R1, BYTEALTO DIRECCION Alabrape
MOVL R2, BYTEBAJO DIRECCION Longitud
MOVH R2, BYTEALTO DIRECCION Longitud
PUSH R2
PUSH R0
CALL LeerPalabra
INC R7
INC R7
PUSH R0
```

```
PUSH R1
PUSH R2
CALL ProcesarPalabra
INC R7
INC R7
INC R7
MOV R3, [R2]
PUSH R3 ; Longitud modificada.
PUSH R1
***** HUECO 1 *****
```

```
CALL EscribirPalabra
INC R7
INC R7
INC R7
JMP ini
Fin
```

### Listado1.ens – Programa Principal

```
PROCEDIMIENTO LeerPalabra
PUSH R6
MOV R6, R7
PUSH R0
PUSH R1
PUSH R2
PUSH R3
PUSH R4
PUSH R5
; Ini. de registros para uso del teclado
MOVH R0, 0F1h
MOVL R0, 0
MOVH R1, 0F1h
MOVL R1, 1
; Dirección de memoria de "Palabra".
INC R6
INC R6
MOV R3, [R6]
; Se espera pulsacion
bucle:
; Mascara para detectar caracter tecleado
MOVH R2, 1
MOVL R2, 0
MOV R5, [R1]
AND R5, R5, R2
BRZ bucle
MOV R5, [R0]
PUSH R1
MOVH R1, 00h
MOVL R1, 30h
XOR R5, R5, R1
BRZ FinPalabra
POP R1
```

```
; Pasar a Mayúsculas.
***** HUECO 2 *****

MOV [R3], R5
; FinPalabra (se pulsó 0)
POP R1
INC R3
INC R4 ; Longitud de la Palabra
JMP bucle
FinPalabra:
; Acceso a Dirección Longitud.
INC R6
MOV R3, [R6]
; Almacena en memoria la long. de Palabra
MOV [R3], R4
POP R5
POP R4
POP R3
POP R2
POP R1
POP R0
POP R6
RET
FINP
```

### Listado2.ens – Procedimiento LeerPalabra

— ¿Qué instrucción o instrucciones faltan en el **Hueco 1** del programa principal?

**MOVH R0, 0FAh**

**MOVL R0, 10h**

**PUSH R0**

— ¿Qué instrucción o instrucciones faltan en el **Hueco 2** del procedimiento LeerPalabra?

**MOVL R2, 0DFh**

**MOVH R2, 00h/FFh**

**AND R5, R5, R2**



Apellidos: \_\_\_\_\_ Nombre: \_\_\_\_\_ DNI: \_\_\_\_\_

## Examen de Fundamentos de Computadores - Septiembre 2008

- Se está ejecutando la instrucción `MOVL R1, 30h` correspondiente al procedimiento `LeerPalabra`. ¿Qué valor aparecerá durante el paso 3 en el bus interno (IB) de la CPU? Responder en hexadecimal.

2130h

- ¿Cuál es la función lógica que implementa el circuito de activación necesario para *mapear* el dispositivo de visualización (*display*) en la CPU teórica? Respuesta tipo ...*a<sub>0</sub>*·*a<sub>1</sub>*·*a<sub>2</sub>*·...

*a<sub>15</sub>*·*a<sub>14</sub>*·*a<sub>13</sub>*·*a<sub>12</sub>*·*a<sub>11</sub>*·...·*a<sub>10</sub>*·*a<sub>9</sub>*·...·*a<sub>8</sub>*·...·*a<sub>7</sub>*·...·*a<sub>6</sub>*·...·*a<sub>5</sub>*·*a<sub>4</sub>*

- ¿Cuál es la codificación de la instrucción `BRZ` Bucle del procedimiento `LeerPalabra`?

F4FBh

Durante la ejecución de una de las instrucciones del procedimiento `LeerPalabra` (*Listado2.ens*) se observa cómo aparecen secuencialmente en el IB los siguientes valores:

Paso 1	0598
Paso 2	0599
Paso 3	8E00
Paso 4	05D3
Paso 5	05D4

- ¿De qué instrucción se trata?

INC R6

- ¿Qué aparecerá en MAR en el primer paso de la siguiente instrucción?

0599h

- Se ha detectado una cierta lentitud en la traducción de las palabras. Al objeto de mejorar el rendimiento, se decide sustituir el muestreo periódico por un sistema de interrupciones. Se ha escrito una rutina de interrupción, `RutinaInt`. ¿Qué instrucciones son necesarias para instalar la rutina de interrupción en el vector 5?

`MOVL R0, 5`

`MOVH R0, 0`

`MOVL R1, BYTEBAJO DIRECCION RutinaInt`

`MOVH R1, BYTEALTO DIRECCION RutinaInt`

`MOV [R0], R1`

`STI`

- ¿En qué paso de los pasos de interrupción desencadenados por la Unidad de Control tendrá el IB el valor del Vector de Interrupción?

I-10

- Un formato de coma flotante se representa por 6 bits para la mantisa (incluyendo signo) y 4 bits para el exponente, y en el orden descrito. La mantisa se codifica en signo magnitud, todo fracción y bit implícito. El exponente se codifica en Exceso a Z Central. ¿Qué error se ha cometido al representar el número 16,25? Responder en decimal.

0,25

- En un ordenador basado en una CPU teórica, se tiene el siguiente mapa de memoria:

- Dos dispositivos de memoria RAM a partir de la posición de memoria 0000h y tamaño 16K cada dispositivo.
- Un dispositivo de memoria ROM de 8K justo a continuación.

¿Cuál es la primera dirección libre para poder mapear un nuevo dispositivo? Responder en hexadecimal.

A000 h

- Indica cuál o cuales de las siguientes afirmaciones son ciertas. Contesta “Ninguna” si crees que ninguna es cierta.

- Una característica importante del método de codificación UTF es que su longitud es constante.
- La precisión de números reales codificados en coma flotante aumenta al aumentar el valor del exponente y disminuye con la disminución del exponente.
- La celda básica de una memoria ROM es un biestable.
- El tipo de memoria DRAM se utilizan principalmente para la memoria principal.

d

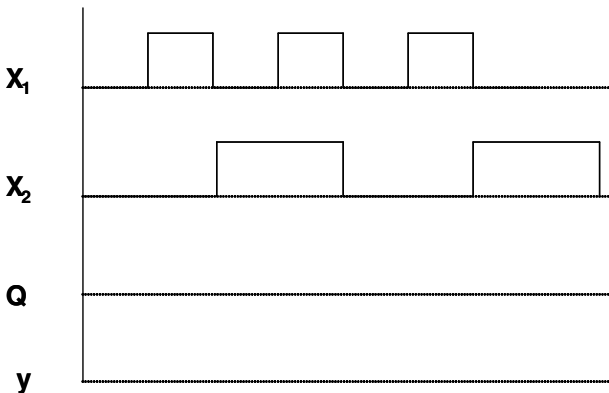
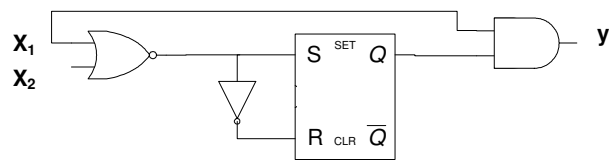
- En un ciclo determinado aparece en IR el valor 13C0h. ¿Qué señales de control se activarán en los ciclos posteriores?

Paso	R6-IB,IB-MAR,READ
Paso	Ciclo de Espera
Paso	MDR-IB,IB-R3,FIN

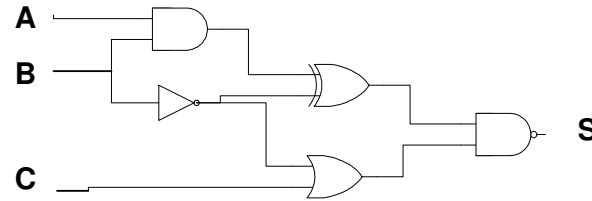
- Un sumador para cantidades de 8 bits opera con los siguientes operandos: número 13 codificado en Exceso a Z (Exceso Central) y número -11 codificado en Complemento a 2. ¿Cuál es el resultado interpretado en Signo-Magnitud?

-2

- Completar el cronograma correspondiente al siguiente circuito. Valor inicial de Q=1.



- Determinar la función lógica en forma canónica del siguiente circuito.



**Forma Canónica:**  $\overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C}$

- Para simplificar la operación de la suma en caso de utilizar un proceso de acumulación se define una nueva instrucción con la siguiente sintaxis:

**ADDA R<sub>d</sub>,R<sub>s</sub> (R<sub>d</sub> = R<sub>d</sub> + R<sub>s</sub>)**

¿De qué señales de control se compondría la instrucción? Omitir los tres primeros pasos.

Paso	Señales
4	Rd-IB, IB-TMPE
5	Rs-IB, ADD, ALU-SR, ALU-TMPS
6	TMPS-IB, IB-Rd, FIN

- Indica cuál o cuales de las siguientes afirmaciones son ciertas. Contesta “Ninguna” si crees que ninguna es cierta:

- El vector de interrupción determina la dirección de comienzo de la tabla de vectores de interrupción.
- Un dispositivo de memoria de 1Mx8 formado por chips de 256kx2 necesita 4 chips por banco para tener 8 líneas de datos.
- El código máquina de la directiva ORIGEN consta de 16 bits.
- La instrucción que se encuentra en el registro IR de la CPU teórica es siempre la que se está ejecutando, independientemente del paso en que se encuentre dicha ejecución.

b)