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

Examen de Fundamentos de Computadores. *Área de Arquitectura y Tecnología de Computadores* **Convocatoria de Junio: 10-06-2008**

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.

Tras el éxito tanto comercial como técnico del sistema de control para un marcador/pantalla de un estadio deportivo diseñado el año pasado, se quiere dotar al sistema de la capacidad para mostrar gráficos en la pantalla/marcador. El marcador tiene las mismas características que el periférico **Pantalla** visto en la asignatura Las dimensiones físicas del dispositivo son las apropiadas para facilitar la visualización de los gráficos desde las gradas. El operario maneja el marcador a través de un periférico de control similar al periférico **Luces** visto en la asignatura. El operario controlará con los interruptores de los 2 bits menos significativos la figura (equis, asterisco, espiral ó triangulo) que quiere mostrar en el marcador. Con los interruptores de los 6 bits menos significativos del byte alto se controlará los atributos de color de texto y de fondo que presentará el gráfico. En la figura siguiente se resumen esta información:

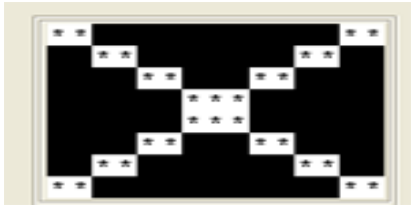
Registro datos periférico Luces

XX	RGB-RGB	XXXXXX	2bits
Sin uso	Atributos fondo-tinta	Sin uso	Figura a dibujar

Para comprobar el funcionamiento del dispositivo se ha escrito el siguiente código:

- **El programa principal:** Establece el vector de interrupción **3 (tres)** asociado al periférico **Luces** y su rutina.
- **rutina_luces:** Rutina que se ejecutará cuando el periférico **Luces** solicite una interrupción. Llamará al procedimiento *"dibuja_figura"* para mostrar en pantalla la figura seleccionada con el atributo marcado.
- **dibuja_figura**: Procedimiento para pintar en la pantalla la figura seleccionada. La figura se formará escribiendo '*****' en las posiciones de pantalla que se indiquen. El asterisco tendrá los atributos de fondo y color que se den como parámetro. Se pasarán a través de la pila y en el orden indicado: **dirección del vector donde se encuentra codificada la figura a mostrar,** y **los atributos** en el byte más significativo de la palabra.

Para codificar la figura que se quiere mostrar en pantalla se usará un vector de 8 elementos. Cada elemento se corresponde con la codificación de una línea de la pantalla y será un número binario de 15 bits (un bit para cada columna de la pantalla) Cada bit indica si hay que escribir en el píxel de esa columna. En la figura se muestra el aspecto de la pantalla al dibujar la equis codificada en la zona de datos.



```
ORIGEN 0500h
INICIO primera
.PILA XXXX

.DATOS
  dir_pantalla VALOR 0ff00h
  dir_luces VALOR 0ffd0h
  dir_ctrl_pantalla VALOR  0ff78h

  p_de_dos_inverso VALOR 16384, 8192
   VALOR 4096, 2048, 1024, 512, 256, 128, 64
   VALOR 32, 16, 8, 4, 2, 1

  equis VALOR   110000000000011b
        VALOR   0011000000001100b
        VALOR   0000110000110000b
        VALOR   0000001111000000b
        VALOR   0000001111000000b
        VALOR   0000110000110000b
        VALOR   0011000000001100b
        VALOR   1100000000000011b

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Datos omitidos intencionadamente.
;
; Son codificaciones de diferentes
;figuras.
;

.CODIGO

PROCEDIMIENTO rutina_luces

    PUSH R0 ;Dirección de inicio figura.
    PUSH R1 ;Atributo.
    PUSH R2 ;Nº figura a mostrar.
    PUSH R3 ;Auxiliar.
    PUSH R4 ;Estado luces.
    PUSH R5 ;Auxiliar.

    ;Leemos el estado del periférico luces en R4.
     MOVH R0,BYTEALTO DIRECCION dir_luces
     MOVL R0,BYTEBAJO DIRECCION dir_luces
              ----- HUECO 1 -----
     MOV R1,[R0]
     MOV R4,[R1]
   ;R4 contiene el atributo y la figura a mostrar.

    ;En R1 ponemos el atributo.
    XOR R3,R3,R3
    MOVH R3,3Fh <----- 2a
    AND R1,R3,R4

;Ponemos en R2 el numero de la figura se tiene que
; mostrar.
    XOR R3,R3,R3
    MOVL R3,3 <----- 2b
```

```
    AND R2,R3,R4

;Ponemos en R0 la dirección de la primera figura.
    MOVL R0,BYTEBAJO DIRECCION equis
    MOVH R0,BYTEALTO DIRECCION equis

    ;Ponemos en R3 nº filas de la pantalla.
    XOR R3,R3,R3
    MOVL R3,8

;Comprobamos si es la primera figura la que
; tenemos que mostrar.
    XOR R5,R5,R5;
    COMP R2,R5
    BRZ es_figura_cero
;En caso contrario avanzamos tantas veces el nº
; columnas hasta llegar al comienzo de la figura
; a mostrar.
seguimos:
    ADD R0,R0,R3
    DEC R2
    BRNZ seguimos
es_figura_cero:

    ;Borramos la pantalla.
     MOVH R3,BYTEALTO DIRECCION dir_ctrl_pantalla
     MOVL R3,BYTEBAJO DIRECCION dir_ctrl_pantalla
     MOV R4,[R3]
     XOR R3,R3,R3
     MOVL R3,3
     MOV [R4],R3

;Llamamos al procedimiento para dibujar la figura.

            ----- HUECO 3 -----
    PUSH R0
    PUSH R1
    CALL dibuja_figura
    INC R7
    INC R7

    POP R5
    POP R4
    POP R3
    POP R2
    POP R1
    POP R0

IRET
FINP

PROCEDIMIENTO dibuja_figura

    PUSH R6
    MOV R6,R7

    PUSH R0 ;Contador para las 15 columnas //
            ; atributo del caracter en la parte
            ; alta.
    PUSH R1 ;Comienzo figura a pintar.
    PUSH R2 ;Dirección pantalla.
    PUSH R3 ;Aux.
    PUSH R4 ;Contador para las 8 filas.
    PUSH R5 ;Dirección potencias de dos
            ; invertidas.
            ;R6 aux.

  ;Recuperamos segundo parámetro, atributo, en R0.
            ----- HUECO 4 -----

    INC R6
    INC R6
    MOV R0,[R6]

    ;Por falta de registros tenemos que
    ; meter en la pila el atributo de
    ;forma temporal,
    ; lo recuperaremos cuando lo necesitemos.
    PUSH R0

  ;Recuperamos primer parámetro, dirección figura,
  ; en R1.
    INC R6
    MOV R1,[R6]

    ;En R2 dirección pantalla.
            ----- HUECO 5 -----
    MOVH R3,BYTEALTO DIRECCION dir_pantalla
    MOVL R3,BYTEBAJO DIRECCION dir_pantalla
    MOV R2,[R3]

    ;En R4 el nº de filas de la pantalla.
    XOR R4,R4,R4
    MOVL R4,8

    ;En R0 el nº de columnas de la pantalla.
    XOR R0,R0,R0
    MOVL R0,15

    ;En R5 la dirección del vector de potencias de
    ; dos en orden inverso.
    MOVH R5,BYTEALTO DIRECCION p_de_dos_inverso
    MOVL R5,BYTEBAJO DIRECCION p_de_dos_inverso

    ;Comenzamos a dibujar la figura, recorreremos
    ; todos los pixeles de la pantalla y
    ; comprobaremos si tenemos que pintarlo o no.
    ;
continua_dibujando:
    ;R3 será una mascara con potencias de dos.
    MOV R3,[R5]
    ;R6 la codificación de la fila de la figura.
    MOV R6,[R1]
   ;Si alguno de los bits de la codificación de la
   ; fila de la figura coincide con la potencia de
   ; dos actual dibujamos el pixel.
    AND R3,R6,R3 <----- 6a
    BRZ no_dibujo_pixel<----- 6b
    ;Dibujamos el pixel actual.
    POP R3 ;Recuperamos de la pila el atributo.
    PUSH R3 ;Volvemos a poner el atributo en
            ; la pila.
    MOVL R3,'*'
    MOV [R2],R3 ;Escribimos en pantalla.

no_dibujo_pixel:
    INC R2  ;Pasamos al siguiente pixel de pantalla.
    INC R5  ;Pasamos a la siguiente potencia de dos.
    DEC R0  ;Decrementamos las columnas tratadas.
    BRNZ continua_dibujando

    ;Hemos terminado de dibujar toda una fila.
    INC R1  ;Pasamos siguiente fila de la figura

    ;Incializamos de nuevo las potencias de dos en
orden inverso.
    MOVH R5,BYTEALTO DIRECCION p_de_dos_inverso
    MOVL R5,BYTEBAJO DIRECCION p_de_dos_inverso
    ;Inicializamos de nuevo la cuenta de columnas.
    XOR R0,R0,R0
    MOVL R0,15

    DEC R4  ;Decremt. contador de filas tratadas.
    BRNZ continua_dibujando

    ;Hemos terminado de dibujar toda la figura.

    ;Quitamos de la pila el atributo y recuperamos
    ; registros
            ----- HUECO 7 -----
    INC R7
    POP R5
    POP R4
    POP R3
    POP R2
    POP R1
    POP R0
            ----- HUECO 8 -----
    POP R6
RET
FINP

primera:

 ;Instalamos la rutina del periferico luces
 ; en el vector 5.
  MOVL R1, BYTEBAJO DIRECCION rutina_luces
  MOVH R1, BYTEALTO DIRECCION rutina_luces
            ----- HUECO 9 -----

  XOR R0, R0, R0
  MOVL R0,5
  MOV [R0], R1
  STI
  JMP -1
FIN
```

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

Examen de Fundamentos de Computadores. *Área de Arquitectura y Tecnología de Computadores* **Convocatoria de Junio: 10-06-2008**

– ¿Qué instrucción o instrucciones faltan en **HUECO 1**? (Usa R1 como registros auxiliares)

```
MOV R1,[R0]

MOV R4,[R1]
```

– Completa las instrucciones marcadas como **2a** y **2b**?

2a: 3Fh

2b: 3

– ¿Qué instrucción o instrucciones faltan en **HUECO 3**?

```
PUSH R0
  PUSH R1
  CALL dibuja_figura
  INC R7
  INC R7
```

– ¿Qué instrucción o instrucciones faltan en **HUECO 4**?

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

– ¿Qué instrucción o instrucciones faltan en **HUECO 5**? (Usa R3 como registros auxiliares)

```
MOVH R3,BYTEALTO DIRECCION
dir_pantalla

MOVL R3,BYTEBAJO DIRECCION
dir_pantalla

    MOV R2,[R3]
```

– Completa las instrucciones marcadas como **6a** y **6b**?

6a: AND

6b: BRZ

– ¿Qué instrucción o instrucciones faltan en **HUECO 7**?

```
INC R7
    POP R5
    POP R4
```

– ¿Qué instrucción o instrucciones faltan en **HUECO 8?**

```
POP R6
RET
```

– ¿Qué instrucción o instrucciones faltan en **HUECO 9?**

```
XOR R0, R0, R0
  MOVL R0,5
  MOV [R0], R1
  STI
```

– ¿Cuál es el tamaño mínimo de pila que debemos reservar en la directiva .PILA para que el programa funcione correctamente? (Expresar en decimal)

19

– La cantidad 80240000h representa un número en formato coma flotante IEEE 754 en precisión simple. ¿Con qué cantidad decimal se corresponde? (Expresar en potencia de dos)

$-(2^{-128}+2^{-131})$

Se sabe que en la CPU teórica un ciclo dura 0,25 milisegundos. Teniendo en cuenta el siguiente fragmento de código:

```
XOR R1, R1, R1
XOR R0, R0, R0
INC R1
COMP R0, R1
BRNZ final
MOVL R3, FFh
final: MOV R5, R2
```

– ¿Cuántos milisegundos empleará en ejecutar dicho código?

8

Se sabe que las entradas de la ALU de 16 bits vista en clase tienen los siguientes valores:

- En el **operando A** se encuentra la combinación de bits que representa el -9 en formato exceso a Z con 16 bits y exceso 120.
- En el **operando B** se encuentran el número positivo más grande que se puede codificar en complemento a 2
- **CarryIn**=0, **Resta**=0, **Op1**=1 y **Op0**=1.

Se recuerda que las ALUs elementales que componen la ALU de 16 bits tienen conectadas las salidas de una puerta AND, de una puerta OR, de una puerta XOR y el resultado de un sumador elemental, a las entradas e_0, e_1, e_2 y e_3 del multiplexador, respectivamente.

– ¿Cuál es el resultado obtenido en la salida? Expresar el resultado con **cuatro dígitos hexadecimales**.

806Eh

Se dispone de una CPU con 32 líneas de datos y 16 líneas de direcciones. El espacio de direccionamiento se divide en 8 zonas iguales, numeradas del 0 (direcciones más bajas) al 7 (direcciones más altas). En las zonas 0 a 2 se instala memoria RAM. En las zonas 3 a 5 se instala memoria ROM. Las zonas 6 y 7 quedan libres para mapear E/S.

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

6000h-BFFFh

– ¿Cuántos chips de organización 4kx8 serán necesarios utilizar para formar el dispositivo de memoria ROM?

24

– ¿A qué banco del dispositivo de memoria ROM corresponderá la dirección de memoria A754h?

4

— ¿Cuáles de las siguientes afirmaciones son **CIERTAS**? (Puede responder "ninguna" si así lo consideras.

a) En las operaciones de resta llevadas a cabo por la CPU elemental, cuando el sustraendo es igual al minuendo se activa el bit CF del registro de estado.

b) La unidad de control de la CPU NO es un sistema digital secuencial.

c) En la CPU elemental, la instrucción PUSH Rx requiere más pasos de ejecución que la instrucción POP Rx

d) Trabajando con 8 bits y exceso a Z=139, el número 117 no es representable.

C,D

— Justo cuando acaba de terminar el paso 3 de una instrucción, los registros PC e IR contienen los siguientes valores:

PC = 0401h
IR = C0FDh

Determina los valores que aparecerán en el bus interno de la CPU en los tres pasos siguientes. Indica los valores en hexadecimal.

Paso 4: 0401 Paso 5: FFFD Paso 6: 03FE

— 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

— Completa los caracteres que faltan en el interior de las casillas, teniendo en cuenta los caracteres que ya hay y sus códigos ASCII.

43h	64h	6Ch	4Ah	31h	35h
C	d	l	J	1	5