Arquitectura de Computadores I

Ejercicios de Autoevaluación - 4

1.- [1 punto] Responde a las siguientes preguntas teniendo en cuenta el procesador sencillo BIRD analizado en la asignatura.

a) Dibuja el registro μIR de la unidad de control microprogramada indicando la funcionalidad de cada uno de sus campos.

b) Se quiere ampliar el conjunto de instrucciones del procesador añadiendo la siguiente: "`addp ri,VARIABLE`". Esta instrucción debe cargar en el registro `pc` la suma del valor del registro `ri` más el valor de la posición de memoria correspondiente a la variable que contiene la instrucción; es decir:

```
pc:= ri+MEM[dirección]
```

Su formato es el siguiente:

6	5	5	16
C.I.		ri	dirección

Escribe la parte correspondiente a la fase de ejecución del microprograma de la mencionada instrucción. Si fuera necesario hacer algún cambio en la Unidad de Proceso, debes indicarlo esquemáticamente.

2.- [2 puntos] Las direcciones de acceso al sistema de memoria de un computador tienen la siguiente estructura:

- **Memoria virtual** paginada, con direcciones lógicas de 20 bits**:**

12 bits	8 bits

- **Memoria principal** entrelazada, con direcciones físicas de 18 bits:

15 bits	2 bits	1 bit

Los tiempos de acceso a la jerarquía de memoria son los siguientes:
TLB: 20 ciclos en caso de fallo, 1 ciclo en caso de acierto
MP: 11 ciclos (1 ciclo desde el buffer de entrelazado)

En este computador se ejecuta el siguiente programa:

```
for (i=0;i<1000;i++)
        B[i]=A[i]*A[i+2];
```

```
        movi r1, #0
        movi r2, #999
bucle:  load r3, A[r1]
        load r4, A[r1+4]
        mul r3, r3, r4
        store r3, B[r1]
        add r1, r1, #2
        subi r2, r2, #1
        bge r2, bucle
```

El programa está cargado a partir de la dirección lógica 0. El vector A comienza en la dirección lógica 1024 y el vector B a partir de la dirección lógica 3040. Las instrucciones y los elementos de los vectores son de tamaño 1 palabra. La tabla de páginas contiene la siguiente información:

Página lógica:	0	4	5	7	11	....
Página física:	1	3	6	0	4	....

Se pide:

a) Esquema de la traducción de direcciones. ¿Cuál es el tamaño máximo de un programa? ¿Cuántas páginas lógicas puede tener como máximo un programa? ¿Cuántas páginas físicas tiene la memoria principal?

b) Traduce las referencias que genera el programa en la primera pasada por el bucle y calcula el tiempo de acceso al sistema de memoria para cada una de esas referencias (tiempo de traducción y tiempo de acceso a memoria). Para ello, indica claramente, mediante una tabla, los diferentes pasos que sigues para el cálculo del tiempo de acceso.

c) Calcula el tiempo de traducción para todas las referencias del programa. ¿Para qué se utiliza el TLB en el proceso de traducción?

3.- [0,5 puntos] Haz un esquema del controlador de interrupciones del PC ¿Qué registros tiene? ¿Para qué se utiliza cada registro? Explica claramente cuáles son los pasos para el tratamiento de una interrupción en el PC. ¿Cuál es el cometido de las señales INT e INTA?

4.- [1 punto] Queremos conectar a nuestro PC un sencillo dispositivo que puede ser de entrada y de salida, y que se va a tratar por encuesta. Consta de cuatro registros visibles a nivel de lenguaje máquina y que no están mapeados en memoria. Dichos registros son los siguientes: un registro de estado (puerto 283H), un registro índice (puerto 280H), un registro de datos (puerto 281H) y un registro de control (puerto 282H). El hecho de que haya un registro índice es debido a que tiene una zona de memoria compuesta de 15 bytes, de forma análoga al controlador del cursor en la pantalla. Cada uno de los bytes está identificado por un número. En concreto, el byte 4 sirve para la introducción de datos al ordenador y el byte 6 para dar a conocer datos del ordenador (de salida).

El registro de estado vale 1 si se ha introducido algún valor en el byte 4 y vale 2 si el controlador está preparado para sacar algún valor en el byte 6. Se utiliza el registro de datos para obtener los datos que vienen de fuera o dar a conocer alguno que haya de mandarse afuera. Por otra parte, cada vez que se accede al registro de datos, hace falta realizar un STROBE en el registro de control en el bit 1.

De acuerdo al funcionamiento del periférico, implementa en C las siguientes funciones:

a) La rutina que hace la encuesta para recibir un valor del exterior y lo deja en la variable global DATOIN:

void Encuesta_Lectura ( ).

b) La rutina de STROBE necesaria: ***void Strobe ()***.

NOTA: Para aclarar el código C, explica lo que hace cada línea de código que escribas. Para hacer este ejercicio, dispones de las siguientes rutinas:

*void **Iret**();* *void **Eoi**();*
*unsigned char **InPort** (puerto);* *void **OutPort** (puerto, valor);*
*unsigned char **LeeByteFis** (unsigned short Seg, unsigned short Desp);*
*void **EscribeByteFis** (unsigned short Seg, unsigned short Desp, unsigned char car);*
*unsigned short **LeePalFis** (unsigned short Seg, unsigned short Desp);*
*void **EscribePalFis** (unsigned short Seg, unsigned short Desp, unsigned short pal);*

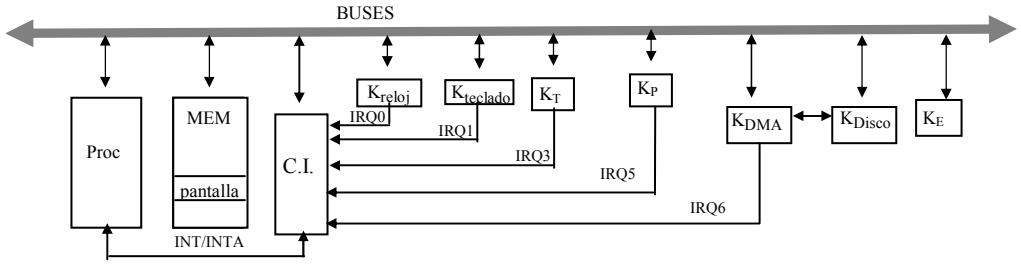
5.- [0,5 puntos] Tenemos un computador cuyo procesador trabaja con un reloj de 2 GHz y queremos transferir un bloque de 32 kB desde un disco duro cuya velocidad de funcionamiento es de 32 MB/s. Calcula la sobrecarga que sufre el procesador por la operación de Entrada/Salida si el sistema realiza las entradas/salidas por DMA. El inicio de una transferencia por DMA requiere 4.000 ciclos de reloj, mientras que la rutina de atención al DMA requiere 2.000 ciclos de reloj.

6.- [1 punto] Contesta las siguientes preguntas sobre buses:

(a) Explica claramente las características de un bus de ciclo partido: ¿cuántas fases se diferencian? ¿cuál es la función de cada una de ellas? ¿quién da inicio a cada una de ellas? Haz un esquema del mismo.

(b) Utilizando un bus síncrono con ancho de banda de 2 MB/s, queremos transferir 2 GB. ¿Cuánto tiempo hará falta para realizar esa transferencia? ¿Y si el bus fuera semisíncrono o asíncrono?

7.- [2,5 puntos] Nos han pedido que diseñemos el **sistema de control de una oficina de atención al público**, de manera que la gente vaya siendo atendida en el mismo orden en que ha ido llegando a la oficina sin necesidad de que formen una cola. Para ello, el sistema cuenta con una máquina expendedora de tarjetas numeradas, situada en la entrada de la oficina, y una ventanilla, en la que será atendida la gente. El sistema dispone de los dispositivos indicados en la figura siguiente:



Los controladores de reloj, teclado, interrupciones y DMA son los mismos que los vistos en la asignatura. En este sistema la sincronización del **teclado** se realiza **por interrupción**. La pantalla está mapeada en memoria a partir de la dirección DIR_PANT. Las características de los controladores de los periféricos novedosos son las siguientes:

K_E: Es el controlador de la máquina expendedora de tarjetas numeradas. La máquina cuenta con un pulsador que la gente debe pulsar para solicitar que se le asigne un número. El controlador posee un registro de estado (REST_K_E) en el que hay un 1 cuando alguien pulsa el pulsador. Posee también dos registros de datos (RDAT1_K_E y RDAT2_K_E) en los que se le debe indicar, respectivamente, el número que debe expender y la hora en que lo expende. Tiene también un registro de control (RCON_K_E) en el que se debe escribir un 1 para indicarle que imprima la tarjeta con el número asignado y la hora, momento en el que el registro de estado pasa a tomar el valor 0 hasta la siguiente solicitud de número. Se sincroniza por **encuesta**.

K_P: Es el controlador de un pulsador situado en la ventanilla. Interrumpirá cuando la persona encargada de la ventanilla lo pulse indicando que la ventanilla está libre.

K_T: Es el controlador de un lector de las tarjetas numeradas expendidas por la máquina expendedora. El controlador genera una petición de interrupción al introducir una tarjeta en el lector de tarjetas. Tiene dos registros de datos (RDAT1_K_T y RDAT2_K_T) en los que aparece la información leída en la tarjeta: el número asignado a la tarjeta y la hora en que ha sido expendida, respectivamente. También tiene un registro de control (RCON_K_T) en el que se debe realizar una secuencia de STROBE cada vez que se le acepta una interrupción.

K_DMA: Es el controlador de DMA y se utiliza para hacer transferencias de memoria a disco. Posee los siguientes registros:

- **registro de control** (RCON_KDMA): cuando se escribe un 1 en él se inicia la transferencia (se pone a 0 automáticamente).
- **registro de dirección** (RDIR_KDMA): dirección de comienzo del bloque a transferir.
- **registro de longitud** (RLON_KDMA): contiene la longitud en bytes del bloque a transferir.
- **registro de estado** (REST_KDMA): cuando su contenido es un 0 indica que la última operación realizada ha sido errónea.

K_DISCO: Es el controlador de disco. Para realizar una transferencia hay que programar este controlador, para lo que tenemos la rutina *ProgramarKDisco()*.

El **funcionamiento** del sistema es el siguiente. Todas las mañanas, antes de abrir la oficina, el personal encargado deberá poner en marcha el sistema, empezándose a expender los números a partir del 1.

Durante el intervalo de tiempo en que está abierta la oficina, los clientes entrarán y, si quieren ser atendidos, deberán pulsar el pulsador de la máquina expendedora de tarjetas numeradas. Ésta expenderá los números consecutivos, conforme vaya siendo detectado su pulsador.

Cuando la persona encargada de atender la ventanilla esté libre, pulsará el pulsador correspondiente, lo que dará lugar a que en una pantalla colocada en la parte superior de la ventanilla aparezca un mensaje indicando qué número de turno se le ha asignado a la ventanilla, para lo que tenemos la función ***escribe_pant(número)***.

La asignación realizada a la ventanilla será válida durante 15 segundos como máximo. Durante ese intervalo, la persona que tiene la tarjeta con el número indicado se deberá acercar a la ventanilla correspondiente y le entregará la tarjeta al empleado, para que éste la introduzca en el lector de tarjetas. Pero si transcurridos los 15 segundos, el empleado correspondiente no ha introducido ninguna tarjeta en el lector de tarjetas (porque no se ha acercado nadie), entonces se le asignará el siguiente número y se volverá a presentar en pantalla la nueva asignación, pero sólo mientras haya números expendidos que todavía no hayan sido atendidos. En el caso en que no haya más números expendidos, el sistema esperará hasta que haya al menos uno.

Cada vez que es atendido alguien, se debe almacenar en memoria la siguiente información: el número de turno asignado, la hora en que ha sido expendido ese número y la hora en la que se le ha empezado a atender —para ello, disponemos de la variable global *HORA*—. Toda esa información ocupa 20 bytes y para escribirla en memoria disponemos de la rutina ***escribmem(núm_turno, HORA$_{expend}$, HORA$_{atención}$)***, la cual se encarga de controlar en qué posiciones concretas de memoria debe escribir: al inicializar el sistema, la información correspondiente a la primera persona atendida se escribe a partir de la dirección DIR_INF y la información posterior se irá almacenando en las posiciones siguientes.

En lo que respecta a la hora, disponemos de dos rutinas: ***InicializarHora()***, que asigna a la variable *HORA* el valor dado por el sistema operativo en el instante concreto, y ***ActualizarHora()***, que actualiza la variable global *HORA* en un segundo.

Con el objetivo de realizar estadísticas, al finalizar la jornada de atención al público, cuando ya no quede gente que atender en la oficina, un operario deberá pulsar la tecla D, de manera que toda la información almacenada en memoria acerca de las personas que han sido atendidas a lo largo del día sea transferida a disco, por medio del DMA. Al finalizar la transferencia de DMA, si ha ocurrido algún error, el sistema deberá intentar realizar la transferencia una y otra vez, hasta que no ocurra ningún error. Para simplificar, supondremos que transferencia habrá acabado correctamente antes del día siguiente. Cuando la transferencia acabe correctamente, el programa deberá finalizar y devolver el control al sistema operativo.

Se pide: Escribe en lenguaje algorítmico todas las rutinas de atención que consideres necesarias, así como el programa principal. Comenta cualquier supuesto que hagas para la resolución del problema. Se valorará representar el comportamiento del sistema mediante un autómata de estados y transiciones.

8.- [1,5 puntos] Queremos calcular la siguiente expresión: X=(A/B-C)*D. Los cálculos se quieren hacer en complemento a 2, con el mínimo número de bits necesario, para el siguiente caso concreto:

A=27, B=6, C=-7, D=-12.

- La división es sin restauración.
- La resta A/B-C se efectúa con una CSA sin RCA.
- Para realizar la multiplicación se utiliza el método de Booth en base 2.

Contesta también a las siguientes preguntas:

a) ¿Hay *Overflow*?

b) ¿Qué valor tiene el resto de la división?

c) ¿Se te ocurre alguna forma para realizar la multiplicación de forma más efectiva en éste caso concreto?

d) ¿De cuántos bits será X?