1. ***(3 puntos)*** Un sistema basado en procesador MIPS R2000 posee una cache L1 dual configurada como sigue:

* **Cache de Instrucciones**: 8 KB, correspondencia directa, tamaño de bloque de **32 Bytes**.
* **Cache de Datos**: 1 KB, correspondencia asociativa por conjuntos de 2 vías, tamaño de bloque de **64 bytes**. Emplea ubicación en escritura (*write-allocate*), actualización posterior (*write-back*) y LRU para reemplazos.

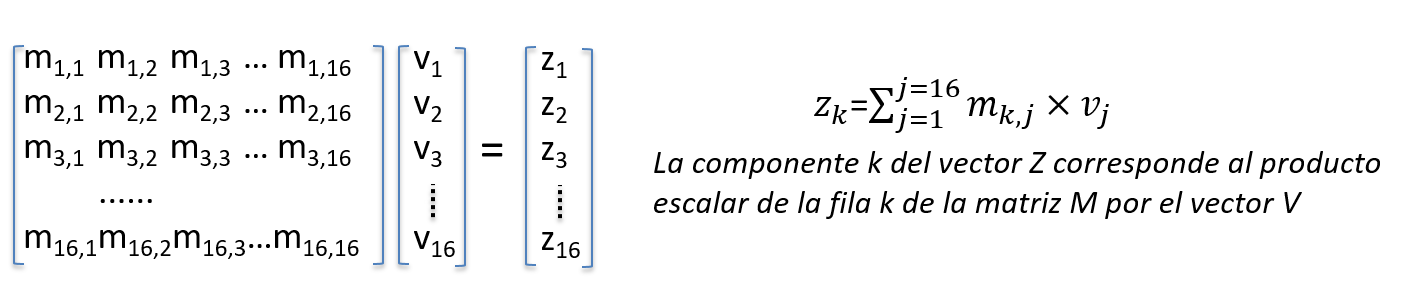
1. *(0.5 puntos)* Indique el número de bits de los campos de la dirección de memoria para ambas caches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cache de Instrucciones** | |  | **Cache de Datos** | |
| **Etiqueta** |  |  | **Etiqueta** |  |
| **Línea** |  |  | **Conjunto** |  |
| **Desplazamiento** |  |  | **Desplazamiento** |  |

1. *(0.5 puntos)* Calcule el tamaño de la memoria de control requerido para la cache de datos

|  |  |
| --- | --- |
|  | **Cache de Datos** |
| **Número de líneas en la memoria de control** |  |
| **Número de bits de cada entrada (indique el nombre de los campos)** |  |
| **Tamaño total de la memoria de control (en bits)** |  |

1. El siguiente programa realiza la multiplicación de la matriz M por el vector V, dando como resultado el vector Z. Todos los elementos corresponden a enteros de 4 bytes. La matriz M está constituida por 256 elementos, organizados en 16 filas y 16 columnas. Los elemenos de la matriz se hallan almacenados en memoria de forma consecutiva, empezando por los de la fila 1 y terminando con los de la fila 16.

**

.data **0x20000000**

M: .word 1,2,3,4, ... 256 # *Matriz 256 enteros (16 filas x 16 columnas)*

*.*data **0x20002000**

V: .word 1,2,3,4, ... 16 # *Vector columna de 16 valores enteros*

*.*data **0x20004000**

Z: .space 64 # *Vector resultado de 16 valores enteros*

.text 0x00400000

\_\_start: lui $t0,0x2000 # *Puntero a matriz M en $t0*

ori $t2,$t0,0x4000 # *Puntero al vector resultado Z en $t2*

li $t3,16 # *Inicializa contador de iteraciones en k*

L0:ori $t1,$t0,0x2000 # *Puntero al vector V en $t1*

li $t6,16 # *Inicializa contador de iteraciones en j*

ori $s0,$zero,0 # *Inicializa registro acumulador $s0*

L1: lw $t4,0($t0) # *Lee M[k,j] (k=1..16, j=1..16)*

lw $t5,0($t1) # *Lee V[j]*

mul $t4,$t5 # *M[k,j]* × *V[j]*

mflo $t4 # *Acumula resultado multiplicación en $t4*

add $s0,$s0,$t4 # *Acumula suma parcial en $s0*

addiu $t0,$t0,4 # *Incrementa puntero a matriz M[k,j+1]*

addiu $t1,$t1,4 # *Incrementa puntero a vector V[j+1]*

addi $t6,$t6,-1 # *Decrementa contador de iteraciones j*

bne $t6, $zero,L1 # *Sigue iterando mientras contador > 0*

sw $s0,0($t2) # *Almacena producto parcial en Z[k]*

addi $t3,$t3,-1 # *Decrementa contador de iteraciones k*

bne $t3, $zero,L0 # *Sigue iterando mientras contador > 0*

.end

c.1) *(0.5 puntos)* Obtenga, para la **cache de instrucciones**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de bloques de código** | **Primer número de bloque** | **Último número de bloque** | **Total de FALLOS de código** |
|  |  |  |  |
| **Total de ACCESOS a código (Indique el cálculo)** |  | | |
| **Tasa de aciertos (Con cuatro dígitos decimales. Indique el cálculo)** |  | | |

c.2) *(0.5 puntos)* Indique los números de bloque del primer y último bloque correspondientes a la matriz y a los dos vectores, así como los conjuntos en los que se almacenan en la **cache de datos (en hex)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Num. de bloques de datos** | **Primer bloque** | | **Último bloque** | | **Primer conjunto** | | **Último conjunto** | |
| **M** |  | |  | |  | |  | |  | |
| **V** |  | |  | |  | |  | |  | |
| **Z** |  | |  | |  | |  | |  | |

c.3) *(0.7 puntos)* Calcule, para la **cache de datos**:

|  |  |
| --- | --- |
| **Total de ACCESOS a datos** |  |
| **Total de FALLOS de datos** | **Fallos de inicio:**  **Fallos de colisión/capacidad:** |
| **Tasa de aciertos** |  |
| **Número de reemplazos de bloque** |  |
| **Número de escrituras a memoria *(palabras de 64 bits)*** |  |

c.4) *(0.3 puntos)* ¿Cuál sería el número de fallos de la cache de datos si se empleara una correspondencia directa en lugar de asociaciativa de 2 vías? Justifique la respuesta.

1. **(3 puntos)** Se dispone de un cajero automático, el cual se halla controlado por un MIPS R2000. Actualmente, dicho cajero solo procesa tarjetas tipo MONEDERO y permite seleccionar únicamente tres importes para retirar: 50€, 100€ y 300€. El sistema está constituido, entre otros, por tres módulos de interfaz: el **TARJETERO,** el **TECLADO** y el **EXPENDEDOR DE BILLETES**, definidos más abajo. Asimismo, existe la variable de sistema **sesion**, cuya función se explica más adelante junto a su definición. La operación en el cajero se inicia introduciendo una tarjeta en el tarjetero, que verifica su clave y obtiene su tipo y saldo. Un proceso se encarga de gestionar de forma continuada el Tarjetero por consulta de estado.

|  |  |
| --- | --- |
|  | El TECLADO dispone de teclas especiales para cada uno de los tres importes: <50>, <100> y <300>, además de la tecla <CANCEL>, todas ellas gestionadas por la interrupción INT1\*. También dispone de las teclas numéricas <0-9> y <ENTER> para introducir el código clave, pero estas teclas no generan interrupción y se procesan al margen por el hardware. El EXPENDEDOR DE BILLETES incluye una impresora para imprimir el justificante del importe retirado en €. Dicho importe debe ser almacenado previamente en el correspondiente registro de su interfaz. |

**TARJETERO** (Dir. Base 0XFF000000) Este dispositivo se controla mediante sincronización por consulta de estado

* Registro de **ESTADO** (solo lectura, 8 bits, DB+0)
  + Bit 0 – **R**: (bit Ready) La interfaz lo pone a 1 cada vez que se introduce una tarjeta y se valida su clave.
  + Bits 7,6: **TIPO-** Tipo de tarjeta:

00: DEBITO

01: CREDITO

11: MONEDERO

* Registro de **CONTROL** (solo escritura, 8 bits, DB+2)
  + Bit 2 – **EX**: (bit de Expulsión) Poniendo EX=1 se ordena la expulsión de la tarjeta
  + Bit 6 – **CL**: (bit Cancel) Se pone a 1 para poner a cero (cancelar) el bit R
* Registro de **SALDO** (lectura y escritura, 32 bits, DB+4): Importe del saldo en € disponible en la tarjeta

**TECLADO** (Dir. Base 0XFF010000) Este dispositivo se controla mediante sincronización por interrupción

* Registro de **CANTIDAD** (solo lectura, 32 bits, DB+4) Almacena cantidad € según tecla 50/100/300 pulsada
* Registro de **ESTADO** (lectura y escritura según bits, 8 bits, DB+8)
  + Bit 0 – **R**: (bit Ready – solo lectura) La interfaz lo pone a 1 con cada vez que se pulsa alguna de las siguientes teclas clave: <CANCEL>, <50>, <100> y <300>. Además, si E=1, entonces se activa la interrupción **INT1\*** del MIPS. R se pone a 0 automáticamente tras leer el registro de ESTADO.
  + Bit 3 – **E**: (bit de Enable – solo escritura) Se pone a 1 para habilitar la interrupción en la interfaz y a 0 para inhibirla.
  + Bits 7 – **TECLA** (bit de tecla especial pulsada **–** solo lectura):

0: 50/100/300

1: CANCEL

**EXPENDEDOR DE BILLETES** (Dir. Base 0XFF020000) Este dispositivo emplea E/S Directa

* Registro de **CONTROL** (solo escritura, 8 bits, DB+0)
  + Bit 2 – **P**: (bit de Print) Se pone a 1 para imprimir justificante del importe retirado.
  + Bit 3 – **EB**: (bit de Expulsión Billetes) Se pone a 1 para abrir trampilla de entrega de billetes.
* Registro de **ENTREGA** (solo escritura, 32 bits, DB+8): Importe a entregar en €

Se supone la existencia de la siguiente variable del sistema:

.kdata

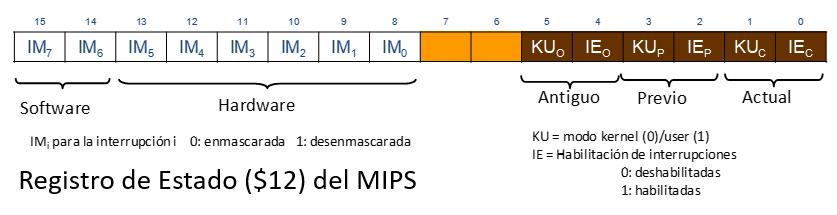
sesion: .byte 0 ;sesión ‘open’(=1) o ‘closed’(=0)

Se pide:

1. *(0.75 puntos)* Programe la siguiente función del sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Índice** | **Argumentos** | **Resultado** |
| *inicializar* | $v0 = 600 | ----- | Inicializa la variable sesion a cero. Habilita la interrupción en el periférico de TECLADO, desenmascara la interrupción INT1 y deja las interrupciones globalmente habilitadas al retornar. |

*Inicializar:*



1. *(1 punto)* Programe el código del proceso del Kernel encargado de gestionar el TARJETERO. Dicho código debe estar a la espera de que se introduzca una tarjeta con clave correcta. Una vez introducida, deberá identificar su tipo. Si la tarjeta es de tipo MONEDERO, se iniciará sesión poniendo la variable sesion a ‘Open’ (=1), en caso contrario se ordenará su expulsión. Este código **debe estar en permanente ejecución**.
2. *(1.25 puntos)* Programe el código de la rutina de servicio de interrupción INT1\* del TECLADO, encargada de gestionar las teclas especiales <50>, <100>, <300> y <CANCEL>. Dicha rutina deberá comenzar comprobando si sesion=’open’ (=1), en cuyo caso procederá a averiguar cuál de las teclas especiales se ha pulsado y a actuar de acuerdo a la misma, tal y como muestra el pseudocódigo a continuación. Las teclas numéricas normales <0-9> sirven solo para introducir el código y se procesan independientemente por el hardware, siendo pues ignoradas.

si sesion == 1 entonces ;consulta si sesion está ‘open’

si TECLA == CANCEL entonces ;expulsar tarjeta, cerrar sesión y salir

sino ; TECLA == 50/100/300

si SALDO > CANTIDAD entonces

;actualizar SALDO, anotar CANTIDAD en reg. ENTREGA del EXPENDEDOR DE BILLETES

;entregar billetes, imprimir justificante, expulsar tarjeta, cerrar sesión y salir

sino ;ignorar y salir

sino ;ignorar y salir

*Int1:*

1. ***(2,5 puntos)*** Considere un escáner conectado a un MIPS. Este periférico digitaliza imágenes rectangulares desde la esquina superior izquierda con una altura *Y* y anchura *X* dadas en píxeles, haciendo que el escáner genere un total de *X*×*Y* píxeles en cada operación. Si el escáner opera en monocromo, cada píxel se codifica en 8 bits; si lo hace en color, cada píxel se representa en 16 bits.

La interfaz está ubicada en la dirección base (AB) 0xFFFF8080 y se ha conectado a la línea de interrupción *int5\**. Puede operar en modo PIO o en modo ADM. Contiene los registros que se describen a continuación:

Macintosh HD:Users:alvaro:Library:Mobile Documents:com~apple~CloudDocs:Faena:ETC:Parcial 2 2019:auxiliar:interfície escànner cas.pdf

* **CR:** Registro de órdenes (AB+0, sólo escritura).
  + Bit 0: **S** (start): al escribir un 1 comienza la operación del escáner
  + Bit 1: **C** (color) a 1 indica color (16 bits por píxel) y a 0 monocromo (8 bits por píxel). Sólo es útil si S=1.
  + Bit 4: **T** (modo de transferencia): a 1 indica ADM, a 0 indica PIO
  + Bit 5: **E** (habilitación de interrupción): mientras el bit R del registro de estado vale 1, activa la línea de interrupción
* **SR:** Registro de estado (AB+4, lectura/escritura)
  + Bit 1: **R** (preparado) Depende del modo de funcionamiento. En modo PIO, vale 1 cuando el registro de datos DR contiene un nuevo píxel de la imagen. En modo ADM, vale 1 cuando el escáner ha transferido toda la imagen. **Es necesario escribir un 0 en este bit para cancelarlo.**
* **X** e **Y:** Coordenadas *X* (AB+12, escritura) e *Y* (AB+16, escritura):

Especifican la anchura y la altura del rectángulo que se escanea, respectivamente.

* **DR:** Registro de datos (AB+20, lectura):

Sólo es útil en modo PIO. Contiene los datos de un píxel. Si opera en monocromo, son válidos los bits del 7 al 0; si lo hace en color, son válidos los bits del 15 al 0.

* **AR:** Registro de dirección (AB+24, escritura):

Sólo es útil en modo ADM. Contiene la dirección inicial del buffer.

Note que la interfaz no dispone de contador para operar en modo ADM. La lógica del adaptador deduce internamente el número de transferencias de píxel multiplicando el contenido de los registros *X* e *Y*.

La intención es ofrecer dos funciones del sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Función** | **Índice** | **Argumentos** | **Resultado** |
| MonoScan | $v0= 1035 | $a0 dirección del buffer  $a1 anchura de la imagen  $a2 altura de la imagen | Escanea en monocromo |
| ColorScan | $v0= 1036 | $a0 dirección del buffer  $a1 anchura de la imagen  $a2 altura de la imagen | Escanea en color |

El código inicial de la función *MonoScan* es el siguiente:

|  |  |
| --- | --- |
| 1 | fun1035: la $t0,0xFFFF8080 |
| 2 | sw $a1,12($t0) # *Paràmetro X* |
| 3 | sw $a2,16($t0) # *Paràmetro Y* |
| 4 | mult $a1,$a2 |
| 5 | mflo $t2 # *$t2 = número de píxeles* |
| 6 | li $t1,1 |
| 7 | sw $t1,0($t0) # *Orden de escanear* |
| 8 | sinc: lw $t1,4($t0) # *Sincronización* |
| 9 | andi $t1,$t1,2 |
| 10 | beqz $t1, sinc |
| 11 | sw $zero,0($t0) # *Cancelación* |
| 12 | lbu $t1,20($t0) # *Transferencia* |
| 13 | sb $t1,0($a0) |
| 14 | addi $a0,$a0,1 # *Avanza puntero y contador* |
| 15 | addi $t2,$t2,-1 |
| 16 | bgtz $t2,sinc |
| 17 | j retxec |

1. *(0,2 puntos)* ¿Es conveniente esta implementación de la función *MonoScan* para que el computador trabaje con un sistema operativo multiprogramado? Razone la respuesta.

1. *(0,3 puntos)* Escriba el código de la función *ColorScan* utilizando las mismas técnicas de sincronización y de transferencia que aplica la versión inicial de *MonoScan*. No es necesario que escriba todo el código; señale solamente las diferencias con el código de *MonoScan* indicando la línea modificada, como muestra el ejemplo.

(1) **fun1036**: la $t0,0xFFFF8080

1. *(1 punto)* Implemente de nuevo la función *MonoScan* con sincronización por interrupciones y transferencia ADM. Escriba los tratamientos de la nueva función *MonoScan*, así como de la interrupción *int5*. Utilice las llamadas a *suspende\_este\_proceso* y a *activa\_procesos\_en\_espera* donde sea necesario.

fun1035:

Int5:

1. *(1 punto)* El circuito de selección de la interfaz y de sus registros tiene la estructura siguiente:

Macintosh HD:Users:alvaro:Library:Mobile Documents:com~apple~CloudDocs:Faena:ETC:Parcial 2 2019:auxiliar:Selecció escàner 3 castellà.pdf

d1) *(0,2 puntos)* ¿Qué líneas del bus de direcciones habrá que conectar al comparador *CMP*?

d2) *(0,2 puntos)* ¿Cuál es la constante *K*? Escríbala en binario.

d3) *(0,2 puntos)* ¿Cuántas entradas y salidas ha de tener el decodificador *DEC*?

d4) *(0,4 puntos)* Complete la lógica de selección del registro *SR*. Tiene que dibujar un circuito lógico con tres entradas (*RD\**, *WR\** y el terminal del descodificador *Si* que indique) y las dos salidas que se conectarán a los terminales OE\* y Clk del registro *SR*.

Macintosh HD:Users:alvaro:Library:Mobile Documents:com~apple~CloudDocs:Faena:ETC:Parcial 2 2019:auxiliar:Selecció SR enunciat cas.pdf

1. ***(1,5 puntos)*** Un servidor ha de suministrar vídeo en tiempo real a través de la red. Los clientes se conectan a él por la red y piden la transmisión de un vídeo. Cada retransmisión implica la lectura de un archivo de vídeo y la transferencia en tiempo real de su contenido por la red. Todos los vídeos están codificados a una velocidad de datos de 800 kbps.

El servidor tiene los periféricos y buses siguientes:

* Bus PN a 10 GBps y NS a 4 GBps.
* Bus MN (memoria) a 12.8 GBps.
* **2 discos internos**, cada uno de 500 GB (500×109 B) de capacidad, con velocidad media de transferencia de 100 MB/s. Cada disco está conectado mediante un adaptador SATA que admite transferencias de 1,5 Gbps con codificación 8b/10b.
* 1 tarjeta de red que opera a 1 Gbps efectivo.
* **1 disco externo** de 1 TB (1000×109 B) de capacidad, con velocidad media de 100 MB/s, conectado a un puerto USB 3.0 a 4,8 Gbps con codificación 8b/10b.

Macintosh HD:Users:alvaro:Library:Mobile Documents:com~apple~CloudDocs:Faena:ETC:Parcial 2 2019:auxiliar:servidor streaming cas.pdf

1. *(0,5 puntos)* ¿Cuántas transmisiones simultáneas puede mantener el servidor si todos los vídeos implicados están en el mismo disco interno? Suponga que el resto del tráfico dentro del computador es despreciable.

¿Cuál será el porcentaje de ancho de banda consumida en el bus NS en este caso?

1. *(0,25 puntos)* ¿Cuántas transmisiones simultáneas puede mantener el servidor si los vídeos implicados están repartidos por igual entre los dos discos internos?
2. *(0,25 puntos)* Asumiendo que ambos discos internos están ocupados al 80% de su capacidad. ¿Cuántas horas de vídeo contendrían? Desprecie el espacio ocupado por el resto de archivos.
3. *(0,5 puntos)* Si se desconecta el servidor de la red ¿cuál sería el mínimo tiempo para hacer una copia de seguridad del contenido de los dos discos internos (ocupados al 80%) en el disco externo?