ESTRUCTURA DE COMPUTADORES

Examen Segundo Parcial *12/Junio/2012*

Apellidos y Nombre	DNI	Grupo

1.- (*3 puntos*) Supóngase el siguiente código ejecutándose en un procesador segmentado en cinco etapas (como el visto en clase) que dispone de todos los cortocircuitos de la ALU y de la Memoria para resolver los riesgos por dependencia de datos. También cuenta con la posibilidad de insertar ciclos de parada en aquellos casos en que no pueda aplicarse alguno de los cortocircuitos disponibles. Además, dicho procesador posee una latencia de salto de 2 ciclos. Los riesgos de control se resuelven mediante la inserción de ciclos de parada.

```
(1)           .text 0x00040000
(2)   __start: addi $t0, $zero, 8
(3)            lui $t1, 0x1080
(4)            ori $t1, $t1, 0xA0
(5)     bucle: lw $t2, 0($t1)
(6)            srl $t2, $t2, 1
(7)            sw $t2, 0($t1)
(8)            addi $t1, $t1, 4
(9)            addi $t0, $t0, -1
(10)           bne $t0, $zero, bucle
```

a) Identificar los riesgos por dependencia de datos e indicar la solución dada a los mismos. En caso de emplear cortocircuitos, indíquese cuáles de ellos se emplean. En caso de emplear ciclos de parada, deberá indicarse su número. Nótese que el bucle se ejecuta 8 veces. **(0,7 puntos)**

	Registro	Número de instrucción en que se escribe	Número de instrucción en que se lee	Solución
Riesgo 1				
Riesgo 2				
Riesgo 3				
Riesgo 4				
Riesgo 5				
……				

b) Calcúlese los ciclos que se emplearían para ejecutar el código y el CPI resultante. **(0,6 puntos)**

Instrucciones ejecutadas (I)	
Ciclos de parada (P)	
Ciclos totales de ejecución (T)	
CPI	

c) ¿Qué ventajas reportaría, en términos de CPI, el empleo de **salto retardado**? En su caso ¿qué instrucciones seleccionarías para **rellenar el hueco de salto** y conseguir así una solución óptima? **(0,7 puntos)**

d) Suponiendo que el procesador se halla segmentado en cinco etapas, cuyos retardos son 45ns, 25ns, 35ns, 45ns y 30ns, y que el retardo de los registros de segmento es de 5ns, indíquese cuál sería el periodo de reloj del procesador y su *speedup* (aceleración) respecto al procesador no segmentado. **(0,4 puntos)**

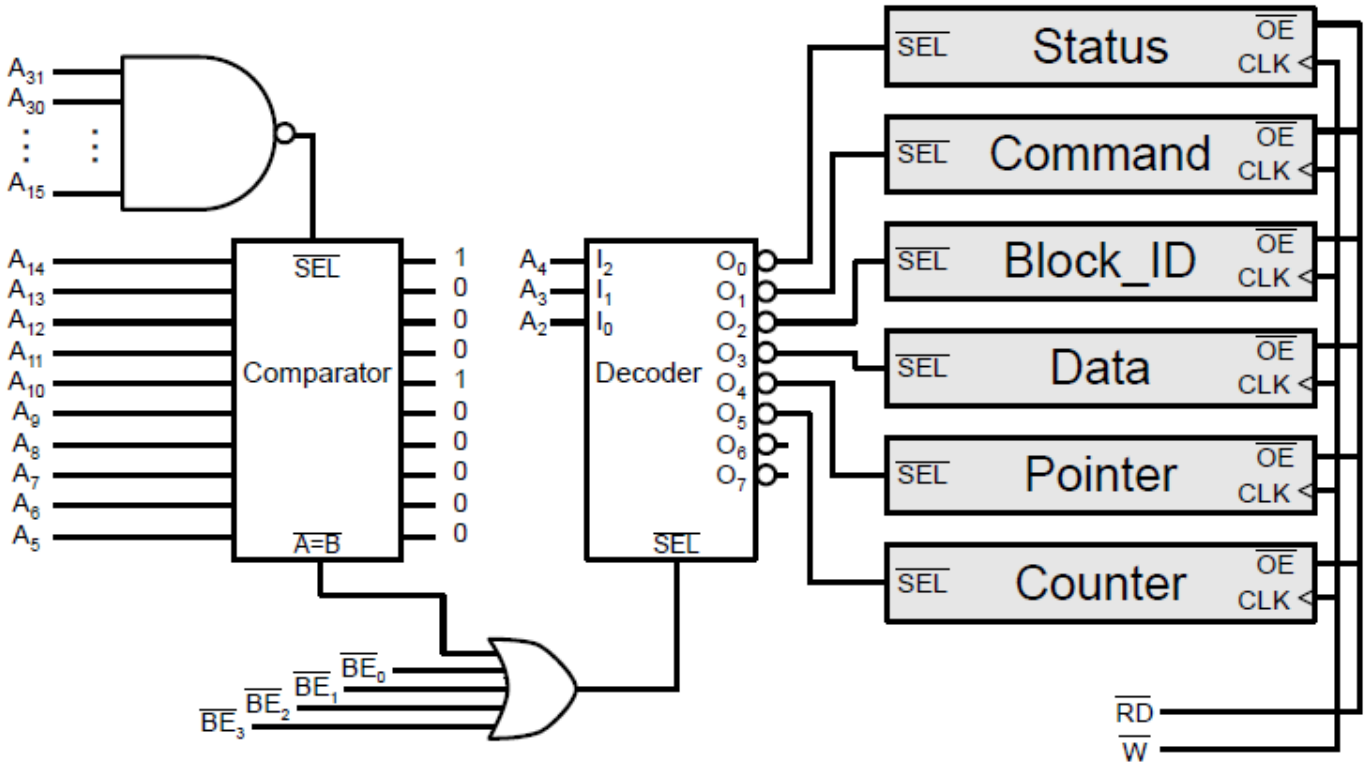
Ciclo de reloj =

Speedup =

e) ¿Cuál sería la **productividad real** del procesador, expresada en MIPS, al ejecutar el código anterior? *Nota: Partir directamente del resultado del apartado b* **(0,3 puntos)**

f) ¿Qué se podría esperar en términos de productividad máxima del empleo de un procesador superescalar de 4 vías trabajando a una frecuencia de reloj la mitad de la del procesador anterior? **(0,3 puntos)**

2.- ***(2,5 puntos)*** La siguiente figura muestra el adaptador de cierto dispositivo de bloques, el cual emplea el mecanismo de DMA para realizar las transferencias de datos:



Todas las líneas de dirección comprendidas entre la A_{31} y la A_{15} se conectan directamente a la puerta NAND (entradas no invertidas). Asumiendo que **los seis registros del adaptador son de 32 bits de tamaño**, se pide:

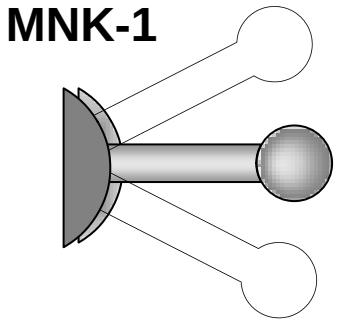
a) ¿Cuál es la dirección base (DB) de este dispositivo? **(0,5 puntos)**

b) Examinando las conexiones de las líneas A4, A3 y A2, indíquense las direcciones en que se ubican cada uno de los 6 registros del adaptador, expresándolas en términos de DB+<desplazamiento>. **(0,4 puntos)**

c) Supóngase que el registro de Estado contiene un bit READY localizado en el bit 0. Este bit se establece a 1 por el hardware cuando hay disponible un nuevo dato en el registro de Datos. Escríbase el código para consultar el dispositivo hasta que esté preparado, y entonces proceder a transferir la palabra del registro de Datos a la variable `Nuevo_Dato` del programa **(0,8 puntos)**

d) Escríbase el código para programar una transferencia DMA desde el dispositivo a memoria (transferencia de lectura). La transferencia se inicia cuando los bits 0 y 1 del registro de Command se establecen a 1. Se desea leer el bloque cuyo identificador (ID) es 0x33331111. El tamaño del bloque es 512 bytes, aunque téngase en cuenta que en cada ciclo que se transfiere una palabra completa de 32 bits. Asimismo, se desea almacenar el bloque en la dirección de memoria etiquetada como `Mem_Block`. *Nota: El contador se decrementa en 1 con cada ciclo de transferencia* **(0,8 puntos)**

3.- ***(3 puntos)*** Considérese el periférico MNK-1, el cual consiste en una palanca que dispone de tres posiciones: una central, de reposo, otra hacia arriba y otra hacia abajo, como se aprecia en la figura



Address	Name	Key
B	Status	R
B+4	Command	E C*
B+8	Data	D U

La interfaz del dispositivo está conectada a un MIPS R2000, actúa sobre la línea de interrupción *Int2*, se ubica en la dirección base B=0xFFFF9000 y dispone de tres registros:

- Registro de **Datos** (sólo de lectura) en la dirección B+8 y que cuenta con dos bits significativos:
  U (bit 0): U=1 si la palanca está desplazada hacia arriba; U=0 en cualquier otro caso
  D (bit 2): D=1 si la palanca está desplazada hacia abajo; D=0 en cualquier otro caso

- Registro de **Comandos** (sólo de escritura) en la dirección B+4 y que cuenta con dos bits significativos:
  C* (bit 0): hacer C*=0 cancela el periféricos haciendo R=0
  E (bit 4): hacer E=1 habilita la interrupción, mientras que E=0 la deshabilita

- Registro de **Estado** (sólo de lectura) en la dirección B y que cuenta con un bit significativo:
  R (bit 0): R=1 cuando la palanca cambia de posición; R=0 cuando el programa cancela el periférico escribiendo C*=0 en el registro de Comandos.

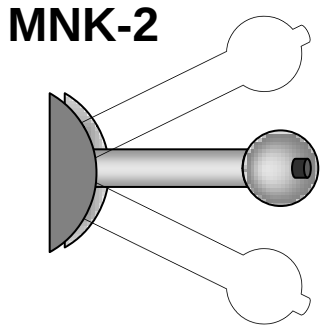
El sistema mantiene una variable `Count`, declarada en el segmento de datos del manejador:

```
            .kdata
Count:      .word 0
```

a) Analizar el código de tratamiento de la interrupción Int2 e indicar lo que hace **(0,8 puntos)**

```
            .ktext
Int2:  la $t0, 0xFFFF9000
       li $t1, 0x10
       sb $t1, 4($t0)
       lb $t1, 8($t0)
L0:    andi $t2, $t1, 1
       bnez $t2, L2
       andi $t2, $t1, 4
       beqz $t2, retexc
L1:    lw $t0, Count
       addi $t0, $t0, -1
       sw $t0, Count
       j retexc
L2:    lw $t0, Count
       addi $t0, $t0, 1
       sw $t0, Count
       j retexc
```

b) Se ha sustituido MNK-1 por un nuevo modelo MNK-2, el cual incorpora un pulsador (ver figura)



Address	Name	Key
B	Status	R
B+4	Command	E C*
B+8	Data	D P U

La interfaz de MNK-2 tiene un nuevo bit P en su registro de Datos descrito en los siguientes términos:

- **P** (bit 1): P=1 si el botón está presionado; P=0 si el botón está liberado

Obsérvese que el actuar sobre el pulsador no deja al periférico preparado, es decir, no hace R=1.

Modificar la rutina de tratamiento del apartado anterior para que mientras el botón está liberado, el comportamiento siga siendo el mismo que antes, pero que cuando el botón esté presionado, el desplazamiento de la palanca hacia arriba o hacia abajo haga Count=0 y el desplazamiento hacia la posición central de reposo no tenga ningún efecto. Escribir el nuevo código a partir de la etiqueta

`Int2`. Supóngase que el código de salida del manejador comienza en la etiqueta `retexc`. **(0,8 puntos)**

```
            .ktext
Int2:       la $t0, 0xFFFF9000
            li $t1, 0x10
            sb $t1, 4($t0)
            lb $t1, 8($t0)









L0:         andi $t2, $t1, 1
            bnez $t2, L2
            andi $t2, $t1, 4
            beqz $t2, retexc
L1:         lw $t0, Count
            addi $t0, $t0, -1
            sw $t0, Count
            j retexc
L2:         lw $t0, Count
            addi $t0, $t0, 1
            sw $t0, Count
            j retexc
```

c) Si la causa de excepción es SYSCALL y el `$v0=205`, el manejador salta al siguiente fragmento de código:

```
FunX:  la $t0, 0xFFFF9000
       sb $zero, 4($t0)
       j retexc
```

Si un programa ejecuta la instrucción `syscall` con `$v0= 205` ¿qué efecto tendrá? ¿Qué le pasará a la variable Contador si el usuario desplaza posteriormente la palanca? **(0,4 puntos)**

d) Escríbase el código de tratamiento de las dos funciones del sistema siguientes: **(0,5 puntos)**

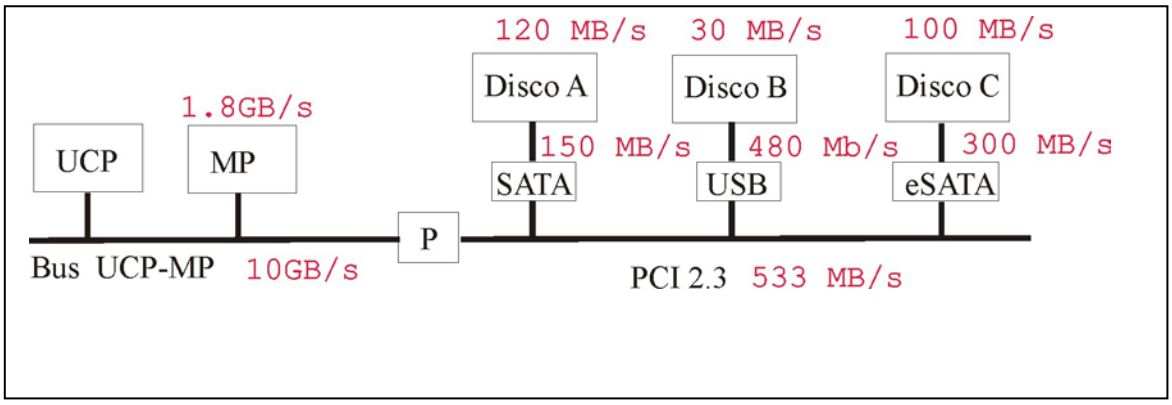
Nombre	$v0	Argumento	Retorno	Descripción
Set-MNK	120	$a0=valor	--	Hace Contador=0
Get-MNK	121	--	$v0=valor	Retorna el valor de Contador

e) ¿Qué pasos o acciones se llevan a cabo desde el momento en que se activa la interrupción *Int2* a la entrada del procesador MIPS R2000 hasta que se inicia la ejecución de la rutina de tratamiento asociada? Indíquese claramente si la acción la realiza directamente el **procesador** (hardware) o el **manejador** (software) **(0,5 puntos)**

Acción	Quién la realiza

4.- *(1,5 puntos)* Considere el sistema de la figura, formado por un subsistema procesador (UCP) y memoria (MP), y un bus de expansión en el que hay instalados tres dispositivos de almacenamiento (discos A, B y C). Las velocidades de cada elemento son las siguientes:

- MP: 1.8 GB/s
- Bus SATA: 150 MB/s
- Bus USB: 480 Mb/s
- Bus eSATA: 300 MB/s
- Bus UCP-MP: 10 GB/s
- Bus PCI 2.3: 533 MB/s
- Disco A: 120 MB/s
- Disco B: 30 MB/s
- Disco C: 100 MB/s



Considerando solo las limitaciones impuestas por el ancho de banda de los buses y los propios dispositivos de almacenamiento (sin tener en cuenta la interferencia del procesador), responda a las siguientes cuestiones:

a) Tiempo mínimo teórico en transferir un archivo de 2GB desde el disco A al disco C. **(0,5 puntos)**

b) Porcentaje de utilización del bus PCI 2.3 para la transferencia del apartado anterior. **(0,5 puntos)**

c) Si la transferencia fuera desde el disco A a otra carpeta de este mismo disco, ¿cuál sería el tiempo mínimo teórico de la transferencia? **(0,5 puntos)**