Profesores: Néstor González – Leo Medina
Fecha: 7 de septiembre de 2018
Duración: 90 minutos

Pregunta 1 (Alternativa 1; escoja solo una). Excepciones (1 pt.)

Considere el camino de datos con control para el manejo de excepciones mostrado en la Figura 1.

A) (0,3 pts.) ¿Qué excepción puede gatillar la instrucción "`add $1, $1, $2`"? Identifique la etapa del *pipeline* en que la excepción es detectada.

"*Overflow*" aritmético, que se detecta en etapa EX.

B) (0,2 pts.) ¿Para esta instrucción en particular, qué problema, desde el punto de vista del programador, puede ocurrir si la ejecución no es detenida en la etapa identificada en A)?

Como el resultado de la suma se escribe en el registro $1 que también es un operando, se corre el riesgo que el programador pierda el valor original de $1 que ayudó a causar el "*overflow*".

C) (0,2 pts.) ¿Qué instrucción entra en la etapa "IF" en el ciclo siguiente de haberse detectado la excepción?

La instrucción que está en la dirección de memoria 80000180_{hex}.

D) (0,3 pts.) Describa cómo implementar en el *pipeline* de la Figura 1 un mecanismo de manejo de excepciones basado en una tabla de direcciones de manejo de excepciones o "*Interrupt Vector Table (IVT)*".

Este enfoque requiere hacer "*fetch*" de la dirección del manejador desde memoria. Debemos agregar el código de la excepción a la dirección de la tabla vector de excepciones, leer la dirección del manejador desde memoria, y saltar a esa dirección. Una forma de hacer esto es tratarla como una instrucción especial que calcula la dirección en EX, carga la dirección en MEM y asigna el PC en WB.

Pregunta 1 (Alternativa 2; escoja solo una). Buses y Periféricos (1 pt.)

A) (0,2 pts) Para administrar los dispositivos periféricos se necesita manejar, según el tipo de dispositivo, todos o algunos de los siguientes recursos: i) IRQ; ii) intervalo de E/S; ii) Intervalo de Memoria y iv) DMA. Explique en qué consiste y para qué sirve cada uno de éstos.

I. IRQ es la señal de interrupción que genera un dispositivo externo, a partir de ella, el PIC se comunica con el procesador, a través de la línea IRQ y le envía un Byte con un identificador del dispositivo que está provocando la interrupción.
II. Intervalo de E/S. Se trata de un intervalo de memoria predefinido para que se almacene información necesaria para la atención de la interrupción. Es un espacio que varía según el dispositivo, pero corresponde a unos cuantos bytes. Se conoce también como puerto.
III. Es un intervalo de memoria predefinido para almacenar información necesaria para atender una interrupción. Este intervalo de memoria es mayor que el de E/S porque debe almacenar más información, por ejemplo, la información de todos los pixeles a ser mostrados en una pantalla.
IV. DMA. Es Acceso Directo a la Memoria. Es una técnica usada en los buses que permite a un dispositivo acceder directamente a la memoria sin la intervención de la CPU.

B) (0,2 pts) Dé 2 ejemplos concretos de casos en que una tecnología de bus paralelo o de puerto paralelo se ha convertido en serial, explicando en qué consisten los cambios y su impacto en la construcción de un computador. Explique además por qué la tecnología de transmisión serial constituye una buena solución para arquitecturas actuales y futuras. Indique también en qué parte de un computador se sigue usando la transmisión paralela y porqué.

Un caso es ATA, la interfaz para Discos y otros dispositivos. De la primera versión paralela pasó a una versión serial SATA y luego a e-SATA. Otro caso es PCI que ha evolucionado de paralelo a serial. El uso de transmisión serial impacta fuertemente en el diseño de la placa madre, de los conectores y de los cables, ya que se requiere menos espacio al usar solo unas pocas líneas.

C) (0,3 pts) Dibuje una estructura de computador con jerarquía de buses basados en buses PCI y explique por qué se diseña la estructura de esa manera (a qué se debe y qué justifica la estructura jerárquica). Indique algunos dispositivos que se conectan al NorthBridge y algunos que se conectan al SouthBridge.

Ver una estructura en los apuntes de clase. La estructura jerárquica se justifica porque la CPU es de muy alta velocidad, también se requiere alta velocidad en el acceso a Memoria y tarjetas gráficas, por ejemplo, pero hay dispositivos periféricos de baja velocidad, como el teclado. Luego, la jerarquía permite unir estos dispositivos ajustando las velocidades cuando es requerido. Los dispositivos del puente norte son los de alta velocidad y los del puente sur son los de baja velocidad.

D) (0,3 pts) Considere las siguientes tecnologías (sus versiones más actuales): USB, FireWire, SATA, PCI-Express, Infiniband, EIDE, QuickPath, Thunderbolt. Complete la tabla siguiente:

Tecnología	Uso principal	¿Con cuál compite?	¿Es un bus?	¿Serial o Paralelo?	Velocidad de Transmisión
USB	Conexión de variados dispositivos externos	Firewire, Thunderbolt	Si	Serial	USB3.1 = 10Gb/s
FireWire	Conexión de variados dispositivos externos	USB	SI	Serial	Aprox 1 Gbps
PCI-Express	Conexión de variados dispositivos internos y externos	¿SATA?	SI	Serial	250MB/S por carril
SATA	Orientado a conexión de DD	¿PCI?	Si	Serial	600MB/s
Infiniband	Conexiones internas y externas	--	Si	Serial	Hasta 56Gb/s
EIDE	Antiguo conector para DD	--	Si	Paralelo	¿? Baja
QuickPath	Conexiones internas en arquitectura Intel	--	Conexión punto a punto	Serial	6,4 GT/s
Thunderbolt	Conexión de variados dispositivos externos	USB	Si	Serial	10GB/s

Pregunta 2. Rendimiento de Procesadores Paralelos (2 pts.)

A) (1 pt.) Suponga que quiere ejecutar un programa 90 veces mas rápido incrementando el número de procesadores de 1 a 100. ¿Qué porcentaje del programa puede ser secuencial para permitir esta aceleración con este número de procesadores?

Recordar la ley de Amdahl: $aceleración = \frac{1}{(1-f_a)+\frac{f_a}{m}}$, donde f_a es la fracción de tiempo afectado y m es la mejora. En este caso, queremos aceleración=90. Además, $m=100$. Reemplazando y despejando para f_a obtenemos que f_a =0,999, que corresponde a la fracción del programa que sería paralelo. En otras palabras, solo el 0,1% del programa puede ser secuencial.

B) (1 pt.) Suponga que quiere realizar dos sumas: una es una suma de 10 variables escalares, y la otra, una suma matricial de dos matrices de 10x10. Asuma que sólo la suma de matrices es paralelizable. ¿Qué aceleración se lograría con 10 versus 40 procesadores? Además, calcule la aceleración asumiendo que las matrices son de 20x20.

Asumiendo que las sumas escalares son secuenciales, y que la suma matricial de 10x10=100 elementos, se puede realizar en paralelo, entonces el tiempo de ejecución para 10 procesadores sería $100t/10 + 10t=20t$, donde t es el tiempo para realizar una suma. Esto representa una aceleración de $110t/20t = 5,5$ (notar que con un procesador el tiempo es $110t$ pues se realizarían 110 sumas secuencialmente). Para 40 procesadores, el tiempo sería $100t/40 + 10t=12,5t$, con una aceleración de 8,8.

Para el caso de matrices de 20x20, del mismo modo se obtienen aceleraciones de 8,2 y 20,5 para 10 y 40 procesadores, respectivamente.

Pregunta 3. Disco duro (1,5 pt.)

Considere los discos duros con las siguientes características.

Disco	Tiempo promedio de "*seek*" (ms)	RPM	Tasa de transferencia del disco	*Overhead* del controlador de disco
"Eastern Digital"	9	7500	100 MB/s	0,2 ms
"Rivergate Technology"	6	15000	50 MB/s	0,2 ms

A) (0,7 pts.) Calcule el tiempo promedio para leer o escribir un sector de i) 1024 bytes, y otro de ii) 4096 bytes, para cada disco.

$$Tiempo\ promedio = tiempo\ seek + \frac{0{,}5}{RPM} + tiempo\ transf. + overhead\ controlador$$

i1) Eastern Digital, 1024 bytes: $T = 9 + \frac{0{,}5\ vueltas}{\frac{7500}{60E3} vueltas/ms} + \frac{1KiB}{100 \cdot 1024KiB/1E3ms} + 0{,}2 = 13{,}21ms$

i2) Rivergate Tech, 1024 bytes: $T = 6 + \frac{0{,}5\ vueltas}{\frac{15000}{60E3} vueltas/ms} + \frac{1KiB}{50 \cdot 1024KiB/1E3ms} + 0{,}2 = 8{,}22ms$

ii1) Eastern Digital, 4096 bytes: $T = 9 + \frac{0{,}5\ vueltas}{\frac{7500}{60E3} vueltas/ms} + \frac{4KiB}{100 \cdot 1024KiB/1E3ms} + 0{,}2 = 13{,}24ms$

ii2) Rivergate Tech, 4096 bytes: $T = 6 + \frac{0{,}5\ vueltas}{\frac{15000}{60E3} vueltas/ms} + \frac{4KiB}{50 \cdot 1024KiB/1E3ms} + 0{,}2 = 8{,}28ms$

B) (0,4 pts.) Para cada disco, determine el factor dominante de rendimiento. Específicamente, si pudiera introducir una mejora en algún aspecto del disco, ¿qué escogería? Si no hay factor dominante, explique por qué.

En ambos casos, el factor dominante o tiempo mayor es el tiempo de "*seek*", y por tanto, la mejora debería ser introducida en el tiempo de posicionamiento del cabezal sobre la pista.

C) (0,4 pts.) ¿En qué condiciones los tiempos promedios de lectura o escritura del "Eastern Digital" y del "Rivergate Technology" podrían ser iguales?

$$9 + \frac{0{,}5\ vueltas}{\frac{7500}{60E3} vueltas/ms} + \frac{DatosKiB}{100 \cdot 1024KiB/1E3ms} + 0{,}2$$

$$= 6 + \frac{0{,}5\ vueltas}{\frac{15000}{60E3} vueltas/ms} + \frac{DatosKiB}{50 \cdot 1024KiB/1E3ms} + 0{,}2$$

$$Datos = 0{,}5\ MiB$$

Si se transfiere medio MiB de datos, entonces los tiempos de transferencia son los mismos.

Pregunta 4. Multithreading (1,5 pts.)

Considere las siguientes organizaciones de CPU:

- CPU SS: Un microprocesador de 2 *cores*, superescalar, que provee la capacidad de enviar instrucciones fuera de orden (*out-of-order*) en dos unidades funcionales. Un solo hilo (*thread*) corre en cada *core* al mismo tiempo.
- CPU MT: Procesador con *multithreading* de granularidad fina que permite que 2 instrucciones corran concurrentemente (hay dos unidades funcionales), pero solo instrucciones de un solo hilo pueden ser enviadas en un solo ciclo.
- CPU SMT: Un procesador SMT ("*Simultaneous Multi-Threading*") que permite que 2 hilos corran concurrentemente (hay dos unidades funcionales), e instrucciones de cualquiera de los hilos puede enviarse a ejecutar en cualquier ciclo.

Asuma que hay dos hilos, X e Y, para correr en estas CPU que incluyen las siguientes operaciones:

Hilo X	**Hilo Y**
A1: toma 3 ciclos en ejecutar	B1: toma 3 ciclos en ejecutar
A2: sin dependencias	B2: conflicto por una unidad funcional con B1
A3: conflicto por una unidad funcional con A1	B3: depende del resultado de B1
A4: depende del resultado de A3	B4: sin dependencias y toma 3 ciclos en ejecutarse

Asuma que todas las instrucciones toman un ciclo de reloj en ejecutarse a menos que se diga explícitamente lo contrario o que se encuentren con un *hazard*. Para los siguientes casos, muestre en qué orden se ejecutan las instrucciones de ambos hilos, separándolas por *core* y/o unidad funcional, para un número suficiente de ciclos de reloj que permita terminar la ejecución de ambos hilos:

A) (0,5 pts) Para una CPU SS.

Core 1	**Core 2**
A3	B1, B4
A1, A2	B1, B4
A1, A4	B1, B4
A1	B2, B3

B) (0,5 pts) Para una CPU MT.

FU1	**FU 2**
A1	A2
A1	
A1	
B1	
B1	
B1	
A3	
B2	B3
A4	
B4	
B4	
B4	

C) (0,5 pts) Para una CPU SMT.

FU1	FU 2
A1	B1
A1	B1
A1	B1
A2	B2
A3	B3
A4	B4
	B4
	B4