Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2343 – Arquitectura de Computadores (II/2014)

Solución Interrogación 3

Pregunta 1

a) Describa y compare entre ellos los conceptos de i) interrupción, ii) interrupción de software y iii) excepción. **(1 pto.)**
Solución: Una interrupción es un mecanismo que tienen los dispositivos de I/O para solicitar la atención de la CPU, sin necesidad de que ella inicie la acción. La interrupciones son generadas por hardware. Las interrupciones de software son interrupciones generadas por un programa, que llaman a funcionalidades específicas, generalmente del sistema operativo. Es análogo a llamar a una subrutina que no es parte del programa. A diferencia de las interrupciones tradicionales, no hay hardware involucrado. Finalmente, las expcepciones son un tipo especial de interrupción que puede ser originada por software o hardware. La excepción se gatilla cuando ocurre algún evento especial, que está especificado previamente, por ejemplo, una división por cero. En caso de ocurrir, el control de la CPU se pasa a una ISR especial para su manejo, generalmente perteneciente al sistema operativo.

b) El algoritmo de reemplazo **MRU** (Most Recently Used), a diferencia de LRU, descarta primero los elementos que han sido ocupados más recientemente. ¿En que casos podría ser útil el uso de este esquema? **(1 pto.)**
Solución: Este esquema podría ser útil cuando se lee un arreglo de manera secuencial y repetidamente, *i.e.*, una vez que se termina de leer, se comienza nuevamente a leer desde el principio.

c) ¿Cuáles son las ventajas y desventajas de tener un controlador de DMA en un computador? **(1 pto.)**
Solución: La ventaja es la liberación de la CPU de la responsabilidad de copiar datos entre distintas posiciones de memoria. Una desventaja es los problemas de consistencia que deben solucionarse cuando se utiliza una caché tipo write-back.

d) ¿Que características recomendaría para la memoria caché de primer nivel (L1), en un procesador que tiene dos núcleos? Justifique su respuesta. **(1 pto.)**
Solución: Para maximizar el rendimiento, se debería usar una caché distinta para cada núcleo, además de que cada una de estas cachés debiese ser del tipo split.

e) Describa el proceso booting de un computador y las partes de éste que están involucradas. **(1 pto.)**
Solución: Al momento de encender el computador, se carga en la memoria principal el programa de inicio, BIOS o UEFI, que se encuentra en una memoria ROM y se modifica el PC para que apunte al principio del programa. Este programa se encarga de inicializar los dispositivos y de verificar el estado de todos los componentes del computador. Una vez concluido este proceso, el programa de inicio actualiza el PC y carga en memoria principal el sector de booteo del disco duro, que contiene la información de como iniciar el sistema operativo.

f) Describa al menos dos posibles soluciones para el problema de consistencia de memoria que se genera al tener un esquema de escritura de caché write-back **(1 pto.)**
Solución: La primera solución consiste en agregar un bit extra a cada palabra de la memoria RAM, que indique si esta ha sido modificada en la caché. Si es este el caso, el programa que esté accediendo a esta palabras debe ir a buscarla a la caché o actualizar la memoria principal. La segunda solución es más simple y consiste en que todo acceso a memoria se haga a través de la memoria caché, más especificamente, el DMA realiza las copias de memoria viendo la memoria caché y no la RAM.

Pregunta 2

a) ¿Cómo afecta el tamaño de las líneas de la caché al rendimiento? Compare y analice distintos casos para el tamaño y relaciónelo con los principios de localidad. **(1 pto.)**
Solución: Al aumentar el tamaño de las líneas, se da prioridad al principio de localidad espacial, ya que se tendrá una mayor cantidad de elementos cercanos. El problema con esto es que si el tamaño de la caché se mantiene constante, se reduce la cantidad de líneas de esta, lo que tiene un efecto negativo desde el punto de vista de la localidad temporal. Además, puede ser que un tamaño grande de línea no corresponda al uso real que se le da a la memoria, por lo que finalmente puede traer elementos a la caché que nunca serán utilizado.
Si por el contrario, se disminuye el tamaño de las líneas, se necesitarán muchos más accesos a memoria, ya que se priorizará sólo la localidad temporal.

b) El tiempo de acceso promedio a memoria (TAPM) puede ser modelado usando la siguiente expresión:

$$\text{TAPM} = \textit{hit-time} + \textit{miss-rate} \times \textit{miss-penalty}$$

Nombre y explique brevemente, para cada uno de los componentes de la fórmula, una técnica para disminuir el tiempo de acceso promedio a memoria. **(1 pto.)**
Solución: Para disminuir el hit-time se podrían ocupar cachés simple, por ejemplo directly-mapped, de manera que se minimize el tiempo para encontrar la línea. Además, si mantenemos el costo fijo, una caché pequeña tendrá un hit-time menor que una más grande.
Para disminuir el miss-rate, se pueden usar cachés más grandes y con mayores asociatividades, de manera de cumplir de mejor maneras los principios de localidad.
Finalmente, para disminuir el miss-penalty, se podrían usar jerarquías de caché multinivel, de manera que bajar para buscar un dato tenga un menor costo.

c) Considere dos computadores idénticos, que sólo difieren en su organización de la memoria caché. El primer computador utiliza una caché 2-way associative y el segundo una caché 4-way associative. Además, en el primer computador, un ciclo del clock toma 1.25ns y la caché tiene un miss rate de 1.0 %. Por otro lado, un ciclo del clock en la segunda máquina toma 1.4ns. Asumiendo que las instrucciones toman en promedio 2 ciclos del clock cuando la caché funciona de manera perfecta, que estas tienen en promedio 1.5 referencias a memoria cada una, la caché tiene un miss-penalty de 75ns y un hit-time de 1 ciclo del clock, determine el miss-rate del segundo computador, de manera que tenga un TAPM menor que el del primer computador. **(2 ptos.)**
Solución:

$$\begin{aligned} TAPM_1 &= 1{,}25 + (0{,}01 \times 75) = 2{,}0ns \\ TAPM_2 &= 1{,}4 + (x \times 75) \\ TAPM_2 < TAPM_1 &\implies 1{,}4 + 75x < 2{,}0 \\ x &< \frac{2{,}0 - 1{,}4}{75} = 0{,}008 = 0{,}8\,\% \end{aligned}$$

d) En un computador, el tiempo que toma un ciclo del clock depende de la asociatividad N de la memoria caché, donde $N = 1$ es una caché con mapeo directo, $N = 2$ una caché 2-way associative, etc. La expresión que describe la dependencia entre la duración de un ciclo del clock y la asociatividad N esta dada por:

$$\text{Duración\_Ciclo\_Clock}(N) = 1{,}0 + 0{,}02 \times N^2$$

De manera similar, el miss-rate también depende de N y está dado por la siguiente expresión:

$$\text{miss-rate}(N) = 0{,}01\text{– }0{,}002 \times N$$

Teniendo en cuenta que un hit de la memoria caché toma 1 ciclo del clock y el miss-penalty es de 75ns, construya una expresión para el TAPM en función de la asociatividad N. Luego, encuentre el valor entero para N que minimize el TAPM. **(2 ptos.)**
Solución:

$$\begin{aligned}\text{TAPM}(n) &= 1{,}0 + 0{,}02n^2 + (0{,}01\text{–}0{,}002n) \times 75 \\ &= 1{,}0 + 0{,}02n^2 + 0{,}75\text{–}0{,}15n \\ &= 0{,}02n^2 - 0{,}15n + 1{,}75\end{aligned}$$

Luego, el valor óptimo se obtiene derivando la expresión anterior e igualando a 0.

$$\begin{aligned}\frac{d}{dn}\text{TAPM}(n) &= 0{,}04n - 0{,}15 = 0 \\ n &= 3{,}75\end{aligned}$$

De esta manera, el valor entero de n que minimiza el TAPM es 4.

Pregunta 3

A un computador x86 se le conecta un dispositivo para mostrar imágenes. Este dispositivo se encuentra mapeado a memoria y demora 1 segundo en mostrar imágenes en escala de grises con una resolución arbitraria. Para configurar su funcionamiento, el dispositivo posee un registro de comandos y un registro de argumentos para los comandos, ambos de 16 bits, donde se debe ingresar la resolución de la imagen a mostrar. El dispositivo posee además un buffer de tamaño K, donde se escribe el contenido de una fila de la imagen que luego es dibujada. Para comunicarse con el computador, el dispositivo genera una interrupción cada vez que va a comenzar a dibujar un fila de la imagen, *i.e.*, cada $\frac{1}{M}$ segundos, donde M es la cantidad de filas de la imagen. Teniendo esto en consideración, responda las siguientes preguntas.

a) Describa los comandos que requiere el dispositivo para su funcionamiento y un posible mapeo de memoria de éste. Explique además cómo conectaría el dispositivo al controlador de interrupciones, de manera que la ISR asociada siempre sin retrasos, *i.e.*, que sea llamada cada $\frac{1}{M}$ segundos. **(1 pto.)**
Solución: Para controlar el dispositivo, se necesitan tres comendos básicos. Uno para setear la resolución horizontal, no para la vertical y uno para dar la orden de inicio del proceso. El comando para el inicio del proceso será $0x0001$, para la resolución horizontal $0x0002$ y para la resolución vertical $0x0004$. Para los comandos de seteo de resolución, antes de enviar el comando de control, se debe escribir en el registro de parámetros la resolución, dada por la cantidad de pixeles. El mapeo de memoria queda entonces de la siguiente manera:

Dirección	Función asociada
0-15	Vector de interrupciones de hardware
16-17	Registro de comandos del dispositivo
18-19	Registro de argumentos del dispositivo
20-(K+20)	Buffer del dispositivo

Finalmente, asegurar que la interrupción no tenga restrasos, es necesario que esta no sea enmascarable y que tenga la prioridad más alta posible para que no sea encolada, lo que implica que debe conectarse en la señal IRQ0.

b) Asumiendo que todas las instrucciones del computador toman un ciclo del clock y que el procesamiento de una petición de interrupción es instantáneo, *i.e.*, el tiempo gastado en todo el proceso de interrupción es sólo el tiempo que toma la ISR, y que los datos son transferidos mediante un esquema PIO, estime la resolución horizontal máxima que puede mostrar el dispositivo, dada una resolución vertical M, si el clock del computador corre a X Hz. **(2 ptos.)**
Solución: Dado que el dispositivo interrumpe cada $\frac{1}{M}$ segundos para dibujar una línea, este es el tiempo máximo que puede tomar la ISR. Luego, si el clock corre a X Hz y cada instrucción toma un ciclo del clock, o sea, $\frac{1}{X}$ segundos, la cantidad de instrucciones que pueden ejecutarse en una ISR es de $\frac{1/M}{1/X} = \frac{X}{M}$. A continuación es necesario definir la cantidad de instrucciones necesarias para dibujar un pixel. Existen multiples opciones, por lo que acá se describirá una básica basada en un ciclo *do while* y direccionamiento con registro base y registro índice. Dado que se utiliza el esquema PIO, se necesitan al menos 2 instrucciones para copiar elementos entre 2 posiciones de memoria. Si asumimos que las imágenes utilizan pixeles de 1 byte, se necesitan 2 instrucciones para dibujar 2 pixeles, si usamos los registros de 16 bits. A esto hay que agregar las instrucciones necesarias para incrementar dos contadores, la instrucción de comparación y la instrucción de salto. Luego, se necesitan $\frac{6}{2} = 3$ instrucciones por pixel. De esta manera, la resolución horizontal máxima es de $\lfloor \frac{X}{3M} \rfloor$ pixeles. Si evaluamos la fórmula con valores típicos para una imagen, como por ejemplo 640x480 pixeles, se obtiene que la frecuencia del clock debe ser de al menos 921.6KHz

c) Escriba un programa que configure el dispositivo para mostrar una imagen que tiene un tamaño válido y una ISR que escriba en la memoria del dispositivo los valores de los pixeles. Asuma que el inicio del contenido de la imagen se encuentra en la variable *imagen*. **(2 ptos.)**
Solución: Asumimos que la frencuencia del clock es de 768KHz.

```
        JMP   main
        reg_comandos dw 0x0010
        reg_params dw 0x0012
        res_hor dw 0x0280 ;640px
        res_ver dw 0x01E0 ;480px

    main:
        MOV BX, reg_params
        MOV [BX], res_hor
        MOV BX, reg_comandos
        MOV [BX], 0x0002

        MOV BX, reg_params
        MOV [BX], res_ver
        MOV BX, reg_comandos
        MOV [BX], 0x0004

        LEA BX, imagen
        MOV DI, 0

        MOV AX, reg_comandos
        MOV [AX], 0x0000

    wait:
        CMP DI, res_ver
        JNE wait

        RET

    ISR0:
        MOV SI, 0
        INC DI
    start:
        MOV AX, [BX]
        INC BX
        MOV [SI+20], AX
        INC SI
        CMP SI, res_hor
        JLT start
        IRET
```

d) ¿Es posible aumentar notablemente la resolución de las imágenes a mostrar, si se utilizara DMA en vez de PIO como esquema de escritura, asumiendo que el controlador de DMA es capaz de copiar una palabra por ciclo del clock? Argumente su respuesta. **(1 pto.)**
Solución: En el caso de la solución acá propuesta, el DMA ofrecería un gran aumento en la resolución, ya que el costo de pintar cada pixel sería sólo 1 ciclo, a diferencia de las 3 instrucciones necesarias por pixel en caso de usar PIO.