Guía para el Trabajo Práctico Número 8

Arquitectura de Computadoras 02/06/2025

Problemas Propuestos

4) Dados los siguientes datos sobre el sistema de memoria de un procesador, se pide:

Datos	Alpha 21264
Dirección virtual	48 bits
Dirección física	41 bits
Tamaño de página	8 KiB
TLB	MD, 256 entradas
Caché L1	MD, 8 KiB, bloques de 64 B
Caché L2	MD, 4 MiB, bloques de 64 B

a) Indique en qué campos se descompone la dirección virtual, y los tamaños de cada uno de ellos.
b) Indique en qué campos se descompone la dirección física, y los tamaños de cada uno de ellos.
c) El caché de primer nivel, ¿es accedido con direcciones virtuales o reales?
d) Indique el tamaño de la etiqueta del TLB.
e) Indique el tamaño de una entrada completa del TLB.
f) Indique en qué campos se descompone la dirección del caché de segundo nivel.

7) Considere un procesador de 1 GHz con 2 niveles de caché, 1 nivel de DRAM y un disco para memoria virtual. El procesador posee un caché de nivel 1 para instrucciones y otro para datos. El sistema de memoria posee los siguientes parámetros:

Componente	Hit Time	Miss Rate	Block Size
Caché L1	1 ciclos	1.5 %	32 Bytes
Caché L2	7 ciclos + 1 ciclo / 32 bits	5 %	256 Bytes
DRAM	16 ns + 3 ns / 4 bytes	0.7 %	4 KiB
DISCO	8.5 ms + 5 ms / 512 bytes	0 %	4 KiB

Por último considere que el caché de nivel L1 trabaja con direcciones reales, es decir que primero se accede a un TLB que tiene un tiempo de acceso de 1 ciclo, que falla 0,1 % de las veces y que requiere una lectura de 4 bytes de la DRAM para actualizarse (estos accesos no pueden fallar).
a) ¿Cuál es el tiempo medio de acceso a memoria para instrucciones y para datos?
b) Suponiendo que todos los accesos a datos son lecturas, ¿cuánto es el CPI promedio de gcc?

Soluciones Completas

4)
a) La dirección virtual se descompone en dos campos: el número de página virtual y el Offset de página. Como las páginas son de 8 KiB son necesarios 13 bits para el Offset, y el resto (35 bits) se usan para el número de página.
b) La dirección física se descompone en dos campos: el número de marco y el Offset de página. El Offset es el mismo que el de la dirección virtual, por lo que también es de 13 bits, quedando 28 bits para el número de marco.
c) Para saber si el caché L1 puede ser accedido con direcciones virtuales, tengo que encontrar cómo se descompone su dirección, y controlar que la cantidad de bits necesarios para identificar un bloque (o sea el índice más el desplazamiento) sea menor o igual que la cantidad de bits del Offset de página. Si no se cumple esta condición el caché L1 sólo podrá ser accedido con direcciones reales.
Como el caché tiene bloques de 64 bytes, son necesarios 6 bits para indicar el desplazamiento. Como es de mapeo directo, tendrá en total (8 KiB / 64 B) = 128 bloques, por lo que se necesitan 7 bits para indicar el índice. O sea que en total son necesarios 7 + 6 = 13 bits para indexar un bloque de caché, y como dijimos anteriormente el Offset de página es de 13 bits, por lo que puede accederse al caché con direcciones virtuales.

d) Teniendo en cuenta los resultados obtenidos en el apartado a, el número de página de la dirección virtual tiene 35 bits. Además como el TLB es de mapeo directo, y tiene 256 entradas, son necesarios 8 bits para su índice y el resto de los bits se usan para la etiqueta, por lo que la misma es de 35 – 8 = 27 bits.

e) Una entrada completa de TLB posee tres campos: bits de estado, la etiqueta a comparar y el número de marco a devolver en caso de acierto. Por lo tanto considerando los resultados obtenidos en los apartados anteriores, y que usamos 4 bits de estado, su tamaño total es de 4 + 27 + 28 = 59 bits.

f) Al caché L2 se accede con direcciones reales (de 41 bits), y a esta dirección se la divide en tres campos: etiqueta, índice y desplazamiento. Como los bloques son de 64 bytes, se necesitan 6 bits para el desplazamiento. Como el tamaño es de 4 MiB, y es de mapeo directo, tiene en total (4 MiB / 64 B) = 64K bloques, por lo que se necesitan 16 bits de índice. El resto de los 41 bits se utiliza como etiqueta, o sea 41 – 16 – 6 = 19 bits. Por lo tanto podemos descomponer la dirección del caché L2 como etiqueta (19 bits), índice (16 bits) y desplazamiento (6 bits).

7) Sabiendo que el procesador es de 1 GHz podemos calcular el período T que será de 1 ns.

a) Para poder calcular el tiempo medio de acceso al sistema de memoria completo debemos comenzar por el último nivel, en este caso el disco. Esto se debe a que los tiempos de acceso de los niveles inferiores se van agrupando y formando la penalización de los niveles superiores.

Para calcular los tiempos de acceso generalmente debemos considerar en primer lugar el tiempo de hit, el mismo esta compuesto por un tiempo fijo y un tiempo de transferencia de datos. Este último es el tiempo necesario para que la unidad que se esta analizando envíe los datos al nivel superior y se calcula teniendo en cuenta la tasa de transferencia del nivel actual y el tamaño de bloque de la unidad del nivel anterior. Además para las unidades que tienen un nivel por debajo, y por lo tanto tienen una tasa de fallos distinta de cero, debemos adicionar la tasa de fallos por la penalidad, la misma será igual al tiempo de acceso del nivel que está por debajo del nivel analizado.

De este modo para calcular el tiempo de acceso del disco debemos tener en cuenta los 8.5 ms que serían el tiempo de acceso fijo, y además, sabiendo que debemos transferir un bloque completo a la DRAM, y que los bloques de la DRAM tienen 4 KiB, adicionamos el tiempo de transferencia del disco para enviar esta información teniendo en cuenta que la tasa de transferencia es 512 bytes cada 5 ms. Dado que el disco no tiene un nivel por debajo, la tasa de fallos es cero por lo que no se adiciona ninguna penalidad.

$$AMAT_{Disco} = 8.5\ ms + 5\ ms * \frac{4\ KiB}{512\ bytes} = 8.5\ ms + 5\ ms * 8 = 48.5\ ms$$

Una vez calculado el tiempo de acceso del disco, subimos un nivel para calcular el tiempo de acceso de la DRAM. En este caso, debemos tener en cuenta los 16 ns de acceso, más el tiempo de transferencia de un bloque de caché L2 a una tasa de transferencia de 4 bytes cada 3 ns. Como la DRAM posee un nivel por debajo (el disco) la misma tiene una tasa de fallos distinta de cero, por lo que deberemos adicionar la tasa de fallos por la penalidad de fallo, teniendo en cuenta que esta penalidad será igual al tiempo de acceso del nivel inferior, es decir, del disco.

$$AMAT_{DRAM} = 16\ ns + 3\ ns * \frac{256\ bytes}{4\ bytes} + 0.007 * 48.5\ ms$$

$$AMAT_{DRAM} = 208\ ns + 339500\ ns = 339708 ns = 339708\ T$$

Al analizar el tiempo de acceso del caché L2 contamos los 7 ciclos de acceso más las transferencia de 32 bytes al caché L1 con una tasa de transferencia de 32 bits (4 bytes) cada ciclo. También adicionamos la penalidad (tiempo de acceso de la DRAM) por la tasa de fallos del caché L2.

$$AMAT_{L2} = 7\ T + 1\ T * \frac{32\ bytes}{4\ bytes} + 0.05 * 339708\ ns$$

$$AMAT_{L2} = 15\ T + 16985\ T = 17000\ T$$

El último nivel de memoria es la caché L1 y el tiempo de acceso a la misma será igual el tiempo de acceso (1T) más la tasa de fallos por la penalidad de fallos (tiempo de acceso

a caché L2). Para este caso en particular en el cual se accede primero al TLB para la traducción de las direcciones, debemos adicionar el retardo provocado tanto por el acceso al TLB como por la falla del mismo, el cual será igual al tiempo requerido por la DRAM para transferir 4 bytes, teniendo en cuenta que la misma no puede fallar.

$$AMAT_{L1} = 1\ T + 0.015 * 17000\ T + 1 + 0.001 * (16\ ns + 3\ ns)$$

$$AMAT_{L1} = 1\ T + 255\ T + 1 + 0.001 * 19\ T = 257\ T$$

Por lo tanto el tiempo medio de acceso a memoria tanto para instrucciones como para datos en el sistema de memoria completo será igual a 257 T.

b) Para calcular el CPI promedio de gcc, tomamos como base el CPI ideal calculado en prácticos anteriores 1,255 y al mismo le adicionamos los ciclos medios requeridos para acceder a memoria. Si tenemos en cuenta que todas las instrucciones acceden a memoria de instrucciones y que el 33% de las instrucciones (instrucciones de transferencia) acceden a memoria de datos (en este caso con la simplificación de que todos estos accesos son lecturas), y considerando el tiempo medio de acceso calculado en el apartado anterior podemos calcular el nuevo CPI promedio.

$$CPI = CPI_{ideal} + ref * (AMAT_{L1} - HitTime_{L1})$$

$$CPI = 1,255 + 1,33 * (257 - 1) = 1,255 + 340,48 = 341,735$$

Por lo tanto el CPI promedio para gcc considerando los accesos a memoria será 341,735. Como se puede observar, este nuevo CPI es casi 300 veces mayor que el ideal, así que pueden darse cuenta del tremendo impacto en la performance que agrega la jerarquía de memoria, y cuánta importancia toma la multiprogramación para poder realizar otras tareas mientras se resuelven los fallos de página.