Guía para el Trabajo Práctico Número 7

Arquitectura de Computadoras 27/05/2026

Problemas Propuestos

2) *[Acotado]* Dada la mezcla de instrucciones para gcc vista en prácticos anteriores, se intentará calcular la influencia del uso de una memoria caché en el CPI del programa.

Tipo de Instrucción	% gcc	$\overline{CPI}$
Aritméticas	48 %	1,0
Transferencias de datos	33 %	1,4
Saltos condicionales	17 %	1,7
Saltos incondicionales	2 %	1,2

a) Calcule el CPI promedio de gcc corriendo en una máquina con un caché, donde la tasa de fallos es del 2 % para las instrucciones y del 4 % para datos, y la penalización por fallos es de 20 ciclos en ambos casos.
b) Calcule cuánto más rápida es la máquina sin fallos que se venía viendo en los prácticos con respecto a la máquina del apartado anterior.
c) Suponga que ahora Ud. realiza mejoras a la arquitectura interna del procesador de manera tal que reduce los CPI de cada instrucción a la mitad. Repita los apartados anteriores y saque conclusiones.

4) *[Acotado]* Considere un caché de mapeo directo, con 16 palabras por bloque, las cuales se cargan de una memoria SDRAM que requiere un ciclo para enviar el dato, un ciclo para enviar el tamaño del bloque, 5 ciclos de latencia y un ciclo por cada palabra a transferir. Calcule el tiempo de penalización por fallos para las siguientes situaciones, tomando siempre como referencia los porcentajes de gcc:
a) Fallo de Lectura en un caché de instrucciones.
b) Fallo de Lectura en un caché de instrucciones con Early Restart (considerando que la mitad de los saltos condicionales son tomados).
c) Fallo de Lectura en un caché de datos con política de escritura write-back, suponiendo que el 20 % de las veces, el bloque a reemplazar ha sido modificado.
d) Fallo de escritura en un caché de datos con política write-through sin buffer.
e) Fallo de escritura en un caché de datos con política write-through con buffer que nunca desborda.

7) *[Acotado]* Una jerarquía de memoria de un procesador está compuesta por dos cachés L1 separados (para instrucciones y datos) y por un caché L2 unificado. El caché L1 de instrucciones es de 16 KiB, posee una tasa de fallos del 3 % y un tiempo de acierto de 1 ns. El caché L1 de datos es idéntico, pero con una tasa de fallos del 8 %. El caché L2 es de 64 KiB, posee un tiempo medio de acceso de memoria de 12 ns. La frecuencia de trabajo del procesador es de 2 GHz.
Luego de aplicar sobre el caché L2 varias técnicas vistas en la asignatura, se pudo reducir a la mitad su tiempo medio de acceso a memoria, sin aumentar su tamaño y sin que sea necesario agregar un caché L3.
a) Sabiendo que el 75 % de los accesos a memoria se corresponden con accesos a instrucciones y el resto son accesos a datos, calcule el tiempo medio de acceso a memoria del sistema original.
b) Mencione dos técnicas que pueden haber sido utilizadas para reducir el AMAT del caché L2.
c) Calcule el tiempo medio de acceso a memoria del sistema con el caché L2 mejorado.
d) A un integrante del equipo se le ocurre que, dado el caché L2 mejorado, sería posible reducir el tamaño de los cachés L1. Por ejemplo, para cachés de 4 KiB se obtiene una tasa de fallos del 8 % para instrucciones y del 19 % para datos, mientras que el tiempo de acierto se reduce a la mitad. Calcule el tiempo medio de acceso a memoria para el sistema con esta modificación propuesta.
e) ¿Tiene sentido la modificación propuesta en el apartado anterior? Justifique su respuesta describiendo dos ventajas y/o desventajas de la propuesta, si las hubiere.

Soluciones Completas

2)

a) La idea de este ejercicio es mostrar la influencia de los fallos en el cálculo del CPI. Hasta ahora habíamos supuesto siempre una memoria ideal, y habíamos obtenido en el mejor de los casos un CPI de 1,255 en el procesador segmentado. Pero cuando las memorias dejan de ser ideales, y se convierten en cachés, su tiempo de acceso sigue siendo de un ciclo, pero sólo cuando hay un acierto.
Cuando ocurre un fallo de caché, el procesador no puede continuar realizando ninguna tarea, y se detiene completamente hasta que el fallo es resuelto. Por este motivo es que la cantidad de ciclos que el procesador pierde a causa de los fallos de caché debe sumarse al CPI.

$$CPI = CPI_{ideal} + Paradas\ por\ fallos$$

Las paradas por fallos se calculan como la cantidad de ciclos que pierdo cada vez que tengo un fallo (penalidad) multiplicado por la cantidad de veces que este evento ocurre (tasa de fallos).

$$Paradas\ por\ fallos = m \cdot Pf$$

Además, no debemos olvidar que el procesador segmentado posee dos cachés separados, uno para instrucciones y uno para datos, y que cada uno de ellos puede influir de diferente manera, según la cantidad de veces que es accedido, pero ambos provocarán paradas.

$$Paradas\ por\ fallos = \%ref_{Instr} \cdot m_{Instr} \cdot Pf_{Instr} + \%ref_{Datos} \cdot m_{Datos} \cdot Pf_{Datos}$$

Con estas ecuaciones, simplemente reemplazamos las variables con los datos proporcionados por el enunciado:

$$Paradas\ por\ fallos = 100\,\% \cdot 2\,\% \cdot 20 + 33\,\% \cdot 4\,\% \cdot 20 = 0.4 + 0.264 = 0.664$$

$$CPI = 1.255 + 0.664 = 1.919$$

Es muy importante destacar que este resultado implica que los fallos de caché incrementan en más de un 50 % el CPI promedio de un procesador, y esto es considerando penalidades que no son tan reales.

4)

a) En el caso de producirse un fallo en la caché de instrucciones, es necesario traer todo un bloque, por lo que la penalización será la suma de los ciclos para enviar el dato (1 ciclo), más los ciclos para enviar el tamaño del bloque (1 ciclo), más los ciclos de latencia (5 ciclos), más la cantidad para transferir todo el bloque (16 ciclos, 1 por cada palabra). El tiempo total será entonces:

$$T = 1 + 1 + 5 + 16 \cdot 1 = 23\ ciclos$$

b) Early restart es una técnica que se utiliza para disminuir la penalización por fallos en cachés que tienen bloques de más de una palabra. Básicamente, consiste en que se empieza a llenar el bloque por la primera palabra y se devuelve el control al procesador apenas llega la palabra solicitada, no cuando termine de llenarse todo el bloque, como es usual.
De esta manera, suponiendo accesos secuenciales (normales en las instrucciones), cuando pida la primera palabra de un bloque será un fallo, pero cuando pida la segunda tendré un acierto. Si extendemos este supuesto, vemos que sólo se fallará al solicitar la primera palabra de un bloque, por lo tanto la penalidad sería la de traer solamente la primera palabra.
Pero no todas las instrucciones son secuenciales. Las secuenciales son las aritméticas (48 %) y las de transferencia de datos (33 %). Los saltos incondicionales no son secuenciales, y suponemos que la mitad de los saltos condicionales sí lo son. Esto nos da que el 89.5 % de las instrucciones son secuenciales, por lo que les corresponde la penalidad de traer la primera palabra de un bloque. Esta sería de:

$$1 + 1 + 5 + 1 = 8\ ciclos$$

Para el resto de instrucciones (10.5 %), les puede tocar cualquier palabra dentro del bloque (incluida la primera), por lo que tomamos un promedio:

$$\frac{8+9+10+\ldots+23}{16} = \frac{8+23}{2} = 15,5 \; ciclos$$

Finalmente, tendremos que:

$$T = 0.895 \cdot 8 + 0.105 \cdot 15.5 = 8.7875 \; ciclos$$

c) En un caché de tipo write-back, los bloques que son modificados no se escriben en memoria hasta que les llega el momento de ser reemplazados. Entonces, en el caso de que se produzca un fallo de caché la penalidad puede verse incrementada si es que el bloque que va a ser reemplazado ha sido modificado previamente. En este caso, podría decirse que la penalidad es doble: se paga una vez para escribir en memoria el bloque modificado, y luego otra vez para traer el bloque solicitado.
Entonces, el 80 % de las veces la penalidad es la de traer todo un bloque: 23 ciclos. Y el 20 % restante la penalidad es doble, por lo que el tiempo total sería:

$$T = 0.8 \cdot 23 + 0.2 \cdot 2 \cdot 23 = 27.6 \; ciclos$$

La ecuación anterior puede reducirse algebraicamente, y expresarse como:

$$T = (1 + 0.2) \cdot 23$$

Lo cual resulta una manera más sencilla de calcular el tiempo.

7)

a) Lo primero que se debe considerar es que la frecuencia del procesador es de 2 GHz, y por lo tanto el periodo de la señal de reloj será de 0,5 ns. Esto permite expresar los tiempos de acierto de los cachés en ciclos de reloj del procesador. Los valores quedarían como 2T para los caché L1 y 24T para el caché L2 original. Con estos valores, y recordando que la penalidad de los caché L1 es igual al tiempo medio de acceso del caché L2, los cálculos del AMAT para cada uno de los caché quedarían de la siguiente forma:

$$AMAT_{L1I} = T_{Hit} + Tf_{L1I} \cdot AMAT_{L2} = 2T + 0.03 \cdot 24T = 2.72T$$

$$AMAT_{L1D} = T_{Hit} + Tf_{L1D} \cdot AMAT_{L2} = 2T + 0.08 \cdot 24T = 3.92T$$

Finalmente para calcular el tiempo medio de acceso memoria del sistema se debe tener en cuenta que el 75 % de los accesos son a instrucciones y el 25 % restante a instrucciones, por lo que el cálculo del AMAT quedaría como:

$$AMAT = F_I \cdot AMAT_{L1I} + (1 - F_I) \cdot AMAT_{L1D} = 0.75 \cdot 2.72T + 0.25 \cdot 3.92T = 3.02T$$