**Problema 13.** Continuación anticipada, Transferencia en desorden

Un procesador tiene una cache de primer nivel (que llamaremos L1) con bloques de 32 bytes y está conectado a un segundo nivel de jerarquía de memoria (que llamaremos L2) mediante un bus de 8 bytes de ancho. El primer nivel (L1) tiene un tiempo de acceso de 1 ciclo cuando se produce un acierto. Cuando fallamos en el primer nivel accedemos al segundo nivel (L2) para leer un bloque de datos. Suponemos que solo se realizan lecturas y que nunca hay fallos en L2. El siguiente cronograma ilustra un fallo en L1: se necesitan 5 ciclos de latencia y 4 para transferir los datos (T0-T3). Los datos se cargan en L1 mientras se transfieren (car L1), y una vez tenemos todo el bloque en L1, hay que realizar una lectura en L1 (Lect) para enviar el dato a la CPU (DATO), cosa que ocurre dentro del mismo ciclo.

CLK

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|  |  |  |  |  |  |  |  |  |  | DATO |  |  |  |  |  | |
| MISS |  |  |  |  |  | car L1 | car L1 | car L1 | car L1 | Lect |  |  |  |  |  | |
|  | Latencia | | | | | T0 | T1 | T2 | T3 |  |  |  |  |  |  | |

Ciclo CPU L1 L2

Al ejecutar un programa en un simulador, hemos obtenido que, a una frecuencia de 2GHz el programa tardaría 2 segundos en el caso ideal de que no haya fallos de cache, que se han realizado 109 accesos a memoria y que el 20% de los accesos provocarían un fallo en L1.

1. **Calculad** el tiempo de ciclo y los ciclos que tarda el programa en el caso ideal.
2. **Calculad** los ciclos de penalización de un fallo de L1 y el tiempo de ejecución del programa teniendo en cuenta la penalización debida a los fallos de L1.

Para mejorar el rendimiento, permitimos que el procesador pueda captar el dato en el mismo ciclo que se transfiere de L2 a L1 (continuación anticipada o *early restart*). Mediante simulación sabemos que cuando se produce un fallo en L1 en nuestro programa, en el 70% de los casos el dato deseado se corresponde al que se transfiere en el ciclo T0, mientras que hay una probabilidad del 10% de que se transfiera en cada uno de los ciclos restantes (T1-T3). Sabemos además que nunca se produce un fallo mientras se acaba de transferir el resto del bloque.

1. **Completad** el siguiente cronograma en donde se ilustran las acciones a realizar en caso de fallo en L1, suponiendo que tenemos continuación anticipada y que el dato solicitado se corresponde al byte 12 del bloque.

CLK

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| MISS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

Ciclo CPU L1 L2

1. **Calculad** el tiempo medio de penalización (en ciclos) de un fallo en L1 y el tiempo de ejecución del programa para el procesador con continuación anticipada.

Para mejorar más el rendimiento hemos modificado también L2 de forma que se transfiera el dato solicitado en el primer ciclo (T0) de la transferencia (transferencia en desorden). Sabemos además que nunca se produce un fallo mientras se acaba de transferir el resto del bloque.

1. **Completad** el siguiente cronograma en donde se ilustran las acciones a realizar en caso de fallo en L1, suponiendo que tenemos transferencia en desorden y que el dato solicitado se corresponde al byte 12 del bloque.

CLK

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | | | | | |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| MISS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

Ciclo CPU L1 L2

1. **Calculad** el tiempo medio de penalización (en ciclos) de un fallo en L1 y el tiempo de ejecución del programa para el procesador con continuación anticipada.
2. **Calculad** la ganancia (speedup) del sistema con continuación anticipada y del sistema con transferencia en desorden respecto el sistema sin ninguna mejora.

**Problema 14.**

### Prefetch

Disponemos de un procesador de 16 bits con direcciones de 16 bits que tiene una memoria cache de datos con las siguientes características: 3-asociativa, con algoritmo de reemplazo LRU, 12 bloques y 64 bytes por bloque, política de escritura: *copy back + write allocate*

El contenido inicial de la memoria de etiquetas (tags) es el siguiente:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **conjunto 0** | **DB** | **conjunto 1** | **DB** | **conjunto 2** | **DB** | **conjunto 3** | **DB** |
| 13 | 1 | 13 | 1 | 13 | 0 | 13 | 0 |
| 43 | 1 | 43 | 1 | 43 | 0 | 43 | 0 |
| AC | 0 | AC | 0 | AC | 1 | AC | 1 |

El DB=1 indica que el bloque correspondiente ha sido modificado. La información de reemplazo está implícita en la posición. Las posiciones inferiores corresponden a los bloques que llevan más tiempo sin utilizarse. Las posiciones superiores corresponden a los últimos bloques utilizados. Por ejemplo, en el conjunto 3, el bloque con tag 13 es el último utilizado, y el bloque con tag AC el que lleva más tiempo sin ser utilizado.

1. **Rellenad** la siguiente tabla, indicando para cada referencia, el número de bloque de memoria que le corresponde, la etiqueta (TAG), a qué conjunto de MC va a parar, si es acierto o fallo (A/F), el bloque reemplazado cuando proceda, el número de bytes leídos de MP (si se lee de MP) y el número de bytes escritos en MP (si se escribe en MP)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tipo** | **dirección (hex)** | **bloque de memoria (hex)** | **TAG**  **(hex)** | **conjunto MC** | **¿acierto o fallo? (A/F)** | **bloque reemplazado** | **bytes escritura MP** | **bytes lectura MP** |
| LECT | B12B |  |  |  |  |  |  |  |
| LECT | B145 |  |  |  |  |  |  |  |
| LECT | B1AF |  |  |  |  |  |  |  |
| LECT | B1C4 |  |  |  |  |  |  |  |
| ESCR | 4387 |  |  |  |  |  |  |  |
| LECT | 1108 |  |  |  |  |  |  |  |
| ESCR | 1199 |  |  |  |  |  |  |  |
| LECT | 11AA |  |  |  |  |  |  |  |

A la cache anterior le añadimos un ***buffer de prefetch*** de una entrada. En este *buffer* se hace prebúsqueda hardware del bloque **i+1** cuando se accede (tanto en acierto como en fallo) al bloque **i**, siempre que el **i+1** no esté ya en la cache o en el *buffer*. En este último caso, no se realiza *prefetch*.

1. **Rellenad** la siguiente tabla (mismas referencias que la anterior) indicando, para cada referencia, el número de bloque de memoria que le corresponde, la etiqueta (TAG), a qué conjunto de MC va a parar, si se produce acierto o fallo en la cache (A/F), el número de bytes leídos de MP (si se lee de MP), el número de bytes escritos en MP (si se escribe en MP), el bloque de MP que se encuentra en el *buffer* (si procede), si se produce acierto o fallo (A/F) en el *buffer* y el bloque que se prebusca de MP (si procede).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tipo** | **dirección (hex)** | **bloque de memoria (hex)** | **TAG**  **(hex)** | **conjunto MC** | **Cache**  **¿acierto o fallo?** | **bytes escritura MP** | **bytes lectura MP** | **bloque actual buffer** | **Buffer**  **¿acierto o fallo?** | **bloque prefetch buffer** |
| LECT | B12B |  |  |  |  |  |  |  |  |  |
| LECT | B145 |  |  |  |  |  |  |  |  |  |
| LECT | B1AF |  |  |  |  |  |  |  |  |  |
| LECT | B1C4 |  |  |  |  |  |  |  |  |  |
| ESCR | 4387 |  |  |  |  |  |  |  |  |  |
| LECT | 1108 |  |  |  |  |  |  |  |  |  |
| ESCR | 1199 |  |  |  |  |  |  |  |  |  |
| LECT | 11AA |  |  |  |  |  |  |  |  |  |

# Apéndice: Cronogramas Tema 3

Arquitectura de Computadores

Departamento de Arquitectura de Computadores

Facultad Infomática de Barcelona Universidad Politécnica de Catalunya

39

### Problema 16

**Cronograma 1:** Buffer de 1 entrada.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración | <---------------Iteración 0 > | | | | | | | | | | <--------------------Iteración 1 > | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ciclo | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| movl %eax, a(,%esi,4) | A |  |  |  |  |  |  |  |  |  | - | - | A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| movl %eax, b(,%esi,4) |  | - | - | - | - | - | B |  |  |  |  |  |  | - | - | - | - | - | B |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| incl %esi |  |  |  |  |  |  |  | i |  |  |  |  |  |  |  |  |  |  |  | i |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cmpl $N, %esi |  |  |  |  |  |  |  |  | c |  |  |  |  |  |  |  |  |  |  |  | c |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| jl A |  |  |  |  |  |  |  |  |  | j |  |  |  |  |  |  |  |  |  |  |  | j |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ocupación bus |  | a[0] | | | | |  | b[0] | | | | |  | a[1] | | | | |  | b[1] | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| # Buffer | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[0] |  | a[0] | | | | |  | b[0] | | | | |  | a[1] | | | | |  | b[1] | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

CPI = ................................................................ Ancho de banda = ...........................................

**Cronograma 2:** Buffer de 2 entradas

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ciclo | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| movl %eax, a(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| movl %eax, b(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| incl %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cmpl $N, %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| jl A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ocupación bus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| # Buffer |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[1] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Cronograma 3:** Buffer de 3 entradas

Arquitectura de Computadores

Departamento de Arquitectura de Computadores

Facultad Infomática de Barcelona Universidad Politécnica de Catalunya

40

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ciclo | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| movl %eax, a(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| movl %eax, b(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| incl %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cmpl $N, %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| jl A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ocupación bus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| # Buffer |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[1] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[2] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

CPI = ................................................................ Ancho de banda = ...........................................

**Cronograma 4:** Merge buffer de 3 entradas

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteración |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ciclo | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| movl %eax, a(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| movl %eax, b(,%esi,4) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| incl %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| cmpl $N, %esi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| jl A |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Ocupación bus |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| # Buffer |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[1] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Buffer[2] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |