資訊工程學系 計算機結構 期末考(107 下)

姓名：　　　　　　　　　　學號：

[Instruction Level Parallelism and Its Exploitation] (36 pnts)

1. (15 pnts) 請說明何謂 Hardware-Based Speculation，為何需要用到 Reorder buffer? 何時指令才可以完成 commit 步驟? 萬一猜測錯誤，應該如何處理?

在某些結果(如 branch 的 outcome 或是 load 的 value)尚未確定的情況下，以預測方式繼續執行指令，在還不確定指令是否該執行的情況下，若該指令已完成，則先將結果存放在 Reorder buffer 中，等確定該指令是應該執行之後，才將 Reorder buffer 中的值更新到相關位置。

指令結果已產生並且已在 ROB 的最前端

將預測執行的指令從 ROB 中刪除

2. (5 pnts) Reorder Buffer 是如何確保 precise interrupt model?

因為 ROB 可以讓指令 In order commit，當指令執行階段發生 interrupt 或是 exception 時，相關訊息將會先被記錄但不處理，直到指令到達 commint 階段時，才處理。

3. (6 pnt) 如果 load 與 store 指令存取相同記憶體位置的資料，將產生 hazard。應該如何做，才能避免 hazard？

load 要被執行時，應該先判斷是否在此之前有其他要寫入同一位置的 store 指令尚未完成

store 要執行時，應該先判斷是否在此之前有其他要讀取或是寫入同一位置的 load 或是 store 指令尚未完成

4. (5 pnts) Speculation 預測下執行並不是對於任何 branch 之後的指令都是如此，請問會考量甚麼因素?

為避免猜錯時過度影響效能，如果需要花較多執行時間的指令，將先不執行

5. (5 pnts) (a)請說明 Branch Target Buffer (BTB)機制 (b)如果想再加快速度，有甚麼方式?

(a)每個 entry 中紀錄了目前指令的 PC 與所預測之接下來指令位置

Next PC prediction buffer, indexed by current PC

(b)直接紀錄上次的執行指令

[Multiprocessors and Thread-Level Parallelism] (44 pnts)

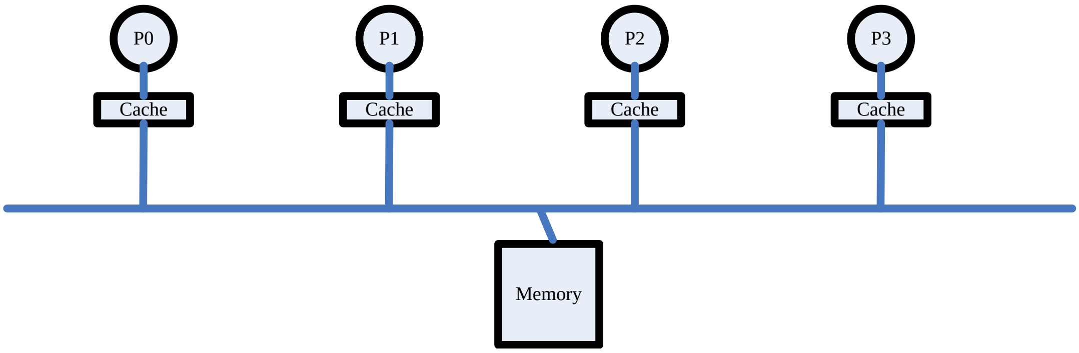
6. (10 pnts) 請說明何謂 Cache Coherency problem。並給予一個例子。

在不同處理器的 cache 中，所看到的同一變數之值不同

7. (3 pnts) 當處理器越多時，哪一種 miss 會增加？

Coherence misses

8. (10.5 pnts) A write-back cache use the write-invalidate cache coherence protocol. In the snoopy protocol, the cache controller must maintain the state transitions for each cache block. We assume that the memory addresses X1 and X2 are in different memory blocks and map to the same cache block in different processors. Their initial values are 0. The initial stats of cache block is Invalid (I). (Cache state: Invalid (I), Shared (S), or Exclusive (E))



Please describe

甲、the stat change of each corresponding cache block

乙、the values of X1 and X2 in the memory and cache

丙、what message is placed in the bus

When the following requests are issued under the snoopy protocol.

1. P0 reads X1 (2 pnts)

Bus: place read miss

In P0, cache block: I→S, X1=0

Memory block: X1=0

2. P1 reads X1 (3 pnts)

Bus: place read miss

In P0, cache block: S, X1=0

In P1, cache block: I→S, X1=0

Memory block: X1=0

3. P2 writes 2 to X1 (3 pnts)

Bus: place write miss

In P0, cache block: S→I

In P1, cache block: S→I

In P2, cache block: I→E, X1=2

Memory block: X1=0

4. P3 writes 3 to X1 (2.5 pnts)

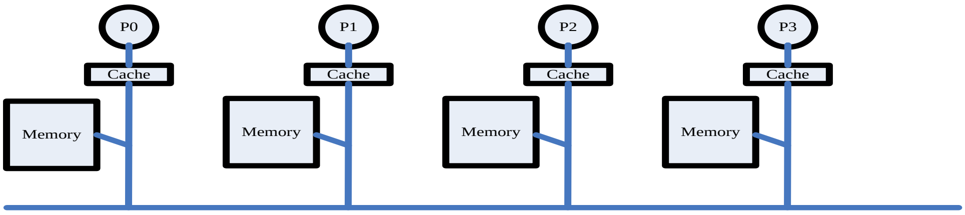
Bus: place write miss

In P2, cache block: E→I

In P3, cache block: I→E X1=3

Memory block: X1=2

9. (15.5 pnts) A write-back cache use the write-invalidate cache coherence protocol. In the directory protocol, the cache controller must maintain the state transitions for each cache block and each memory block. Besides, we assume that the memory addresses X1 and X2 are in different memory blocks on the processor P2 and map to the same cache block. Their initial values are 0. The initial stats of each cache block and each memory block are Invalid and Uncached, respectively. States of a memory block: Uncached(U), Exclusive(E), and Shared(S). States of a cache block: Invalid(I), Exclusive(E), and Shared(S).



Please describe

甲、the stat change of each corresponding cache block and memory bock

乙、the sharer set for the memory block

丙、the values of X1 and X2 in the memory and cache

丁、what message is placed in the bus

When the following requests are issued under the directory protocol. (Hint: home node, local node, and remote node)

1. P0 writes 1 to X1 (3 pnts)

Bus: write miss

In P0, cache block: I→E X1=1

In P2, X1 memory block: U→E Sharer={P0} X1=0

2. P0 writes 2 to X2 (5 pnts)

Bus: data write back, write miss,

In P0, cache block: E→E X2=2

In P2, X1 memory block: E→U Sharer={} X1=1

X2 memory block: U→E Sharer={P0} X2=0

3. P1 reads X2 (4 pnts)

Bus: read miss, Fetch

In P0, cache block: E→S X2=2

In P1, cache block: I→S X2=2

In P2, X2 memory block: E→S Sharer={P0,P1} X2=2

4. P1 writes 3 to X2 (3.5 pnts)

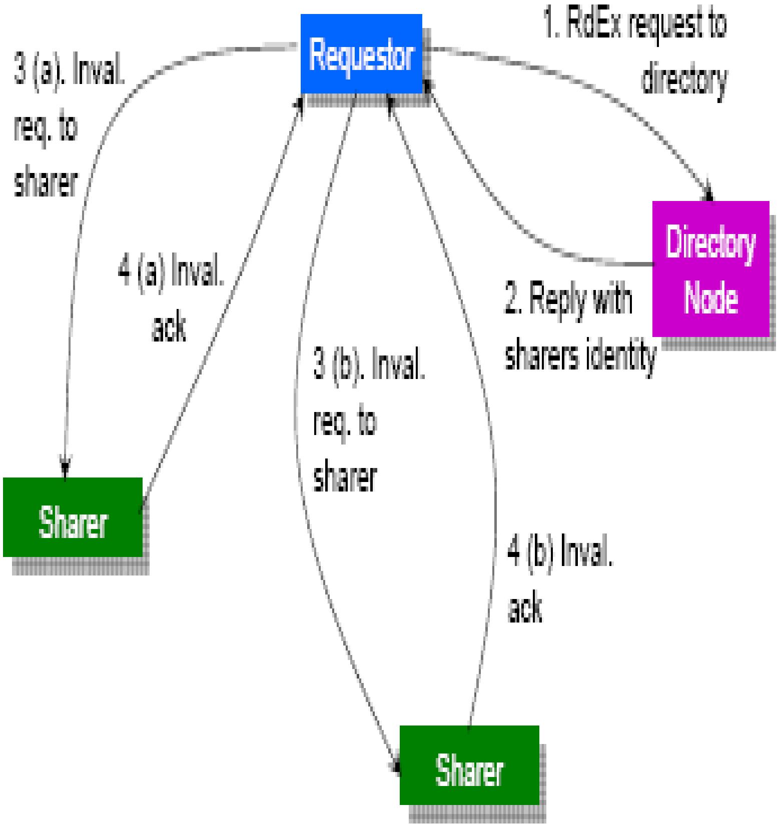
Bus: Invalid

In P0, cache block: S→I

In P1, cache block: S→E X2=3

In P2, X2 memory block: S→E Sharer={P1} X2=2

10. (5 pnts) In the original directory protocol, a home node can be the performance bottleneck. Please propose an idea to improve the performance for the case where a locale node has a write hit for a clean block.


1. RdEx request to directory
3 (a). Inval. req. to sharer
Requestor
Directory Node
2. Reply with sharers identity
4 (a) Inval. ack
3 (b). Inval. req. to sharer
Sharer
4 (b) Inval. ack
Sharer

The local node gets the remote owner IDs from the home node. The local node directly sends the invalidations to the remote nodes.

[Data-Level Parallelism in Vector, SIMD, and GPU Architectures] (20 pnts)

11. (5 pnts) 在 Vector processor 中的 vector register 一次可以存放多的資料，請問以下指令執行時，執行行為如何?

```
LV          V1,Rx          ;load vector X
MULVS.D     V2,V1,F0       ;vector-scalar multiply
LV          V3,Ry          ;load vector Y
ADDVV.D     V4,V2,V3       ;add two vectors
SV          Ry,V4          ;store the sum
```

不需等待整個 vector 暫存器中的所有 Element 都載入完畢，如有載入完成，即可開行執行乘法動作，只要完成乘法動作的，即可繼續往下做加法，依此類推。

12. (5 pnts) Vector processor 與 GPU 都可以在同一個時間點執行多個指令，但是架構上完全不同，請描述兩者 function unit 設計上的差異。

前者為 function unit 為 deeper pipeline 設計

後者為 function unit 有多個

13. (5 pnts) 為何會有 Vector Mask Registers 的需求? (可以以例子說明)

並不是 vector register 中的所有 element 都要做相同的運算，有時會依據條件決定是否要執行運算，不須運算者會在 Mask Registers 註明 disable。

```
for (i = 0; i < 64; i=i+1)
            if (X[i] != 0)
                  X[i] = X[i] – Y[i];
```

14. (5 pnts) 課堂上講述過 GCD test: If a dependency exists, GCD(c,a) must evenly divide (d-b)。請利用 GCD test 判斷以下程式迴圈的 iteration 之間是否有相依性。

(a)
```
for (i=0; i<100; i=i+1) {
    X[3*i+2] = X[2*i+1]+1;
}
```

a=3, b=2, c=1, and d=1

GCD(c,a)=1

d-b=1

GCD(c,a)＝1 可以整除(d-b)=1，所以可能存在相依性