資訊工程學系 計算機結構 期末考(106 下)

姓名： 學號：

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

1. (5 pnts) 請說明何謂 hardware speculation。

當還不確定 branch 指令的 outcome 時，以預測的方式猜測可能的 outcome，之後繼續執行這個路徑上的指令。

Overcome control dependence by hardware speculating on outcome of branches and executing program as if guesses were correct

2. (6 pnt) 為了將原本的 Tomasulo 機制擴展成可以達到 hardware-based speculation 運作，需要額外在原 Tomasulo 機制增加(a)甚麼階段與增加(b)甚麼硬體? (c)新的階段所要做的任務是甚麼?

(a) Commit

(b) Reorder buffer

(c) 當指令進到 commit 階段時，經判定該指令是必須執行的，則將記錄在 ROB entry 中的結果存入記憶體或是暫存器，否則需要清除此 entry 之後的所有 entry

3. (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。

If an instruction caused an interrupt, we could simply wait until it reached the head of the ROB and take the interrupt, flushing any other pending instructions from the ROB. Because instruction commit happens in order, this yields a precise exception.

4. (4 pnt) 給予可以使得 CPI<1 的兩個機制。

Superscalar processors

VLIW (very long instruction word) processors

5. (10 pnt) (a)What step is added to Tomasulo Algorithm in order to implement the hardware-based speculation? (b)Besides, what hardware is needed to prevent any irrevocable action? (c) What interval does the hardware supply operands in? (d) What conditions are satisfied if an instruction achieves the added step? (e) What is done in the step?

(a) Commit 1

(b) Reorder buffer 1

(c) between completion of instruction execution and instruction commit 2

(d) Result is made and the instruction is at head 3

(e) Updeate the register or memory with the result in ROB 3

6. (6 pnts) (a)請說明 Branch Target Buffer (BTB)機制 (b)與 Branch Prediction Buffer 相較，BTB 的好處與額外增加的成本是甚麼?

(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)減少計算下個指令的時間，需要多額外空間做 PC 紀錄

[Multiprocessors and Thread-Level Parallelism] (49 pnts)

7. (5 pnts) 請說明何謂 Cache Coherency problem。

在不同處理器的 cache 中，所看到的同一變數之值不同

8. (6 pnts) What is the coherence miss? The coherence miss can be divided into two subtypes: true sharing misses and false sharing misses. Please explain false sharing misses. And give a solution to false sharing misses.

Miss caused due to the coherence protocol 3

when a block is invalidated because some word in the block, other than the one being read, is written into 2

One-word size 1

9. (6 pnts) We can use the write invalidate protocol or write update protocol in implement the coherence protocol. Please explain them. And which is better? Please give the reason.

Write invalidate protocol

A. To ensure that a processor has exclusive access to a data item before it writes that item.
B. It invalidates other copies on a write.

Write update or write broadcast protocol:

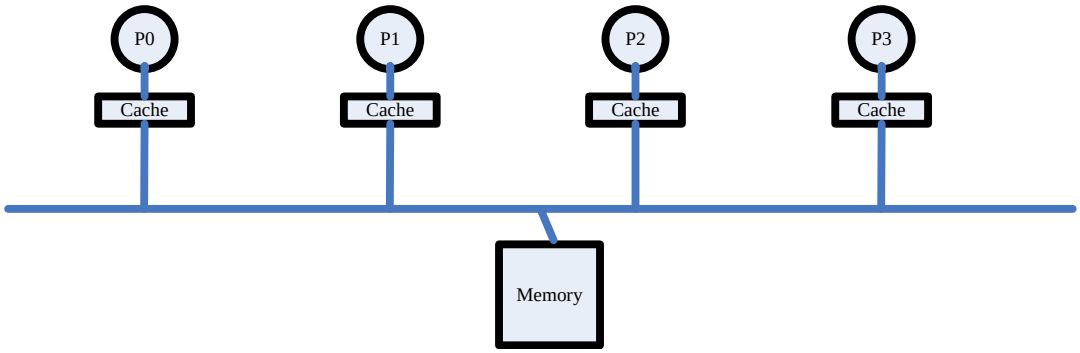
C. To updates all the cached copies of a data item when that items is written.
D. Because a write update protocol must broadcast all writes to shared cache lines, it consumes considerably more bandwidth.

由於前者只傳送 invalidation 訊息，並不會將新的資料廣播給所有處理器知道，因此較節省頻寬

10. (6 pnts) 請解釋 directory-based protocol 中的三種 node：local、home、remote nodes。.

A. **Local node** where a request originates
B. **Home node** where the memory location and the directory entry of an address reside
C. **Remote node** has a copy of a cache block, whether exclusive or shared

11. (11 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) P1 writes 1 to X1 (2 pnts)

In P1, cache block: I→ E X1=1 write miss

Memory block: X1=0

(2) P2 reads X1 (3 pnts)

In P2, cache block: I→S X1=1 read miss

In P1, cache block: E→S write back the dirty block, abort the memory access

Memory block: X1=1

(3) P0 writes 2 to X1 (3 pnts)

In P0, cache block: I→E X1=2 write miss

In P1, cache block: S→I

In P2, cache block: S→I

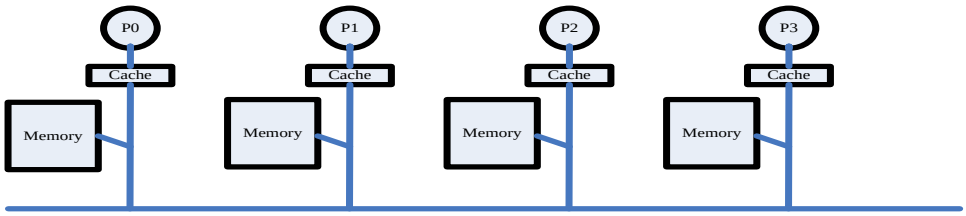
Memory block: X1=1

(4) P2 writes 3 to X2 (3 pnts)

In P2, cache block: I→ E X2=3 write miss

Memory block: X1=1 X2=0

12. (15 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) P1 reads X1 (3 pnts)

In P1, cache block: I→ S X1=0 Read miss

In P2, X1 memory block: U→ S Sharer={P1} X1=0, Data Value Reply

(2) P2 reads X1 (3 pnts)

In P2, cache block: I→ S X1=1 ~~Read miss~~

In P1, cache block: S→S

In P2, X1 memory block: S→S Sharer={P1,P2} X1=1, ~~Data Value Reply~~ (home node=local node)

(3) P1 writes 2 to X1 (3 pnts)

In P1, cache block: S →E X1=2 Invalidate

In P2, cache block: S →I

In P2, X1 memory block: S→E, Sharer={P1} X1=0, Invalidate

(4) P3 writes 3 to X1 (3 pnts)

In P3, cache block: I→E X1=3 Write miss

In P1, cache block: E→I

In P2, X1 memory block: E→E Sharer={P3} X1=2, Fetch/Invalidate, Data Value Reply

(5) P1 reads X1 (3 pnts)

In P1, cache block: I→S X1=3 Read miss

In P3, cache block: E→S X1=3

In P2, X1 memory block: E→S Sharer={P1,P3} X1=3, Fetch, Data Value Reply

[Data-Level Parallelism in Vector, SIMD, and GPU Architectures] (25 pnts)

13. (5 pnts) 如果驗一個 vector processor 系統中有 4 個 memory bank，當欲從記憶體載入資料到一個 64 個 element 的暫存器時，資料在 bank 中如何放置時，會有最大的讀取時間。

如果欲載入的資料皆是從同一個 bank 中讀取，將會有此情況

14. (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，請描述其差異。

前者為 function unit 為 deeper pipeline 設計

後者為 function unit 有多個

15. (10 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[i+1] = X[i]+1;
}

a=1, b=1, c=1, and d=0

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

(b) for (i=0; i<100; i=i+1) {
  X[2*i] = X[2*i-1] *2;
}

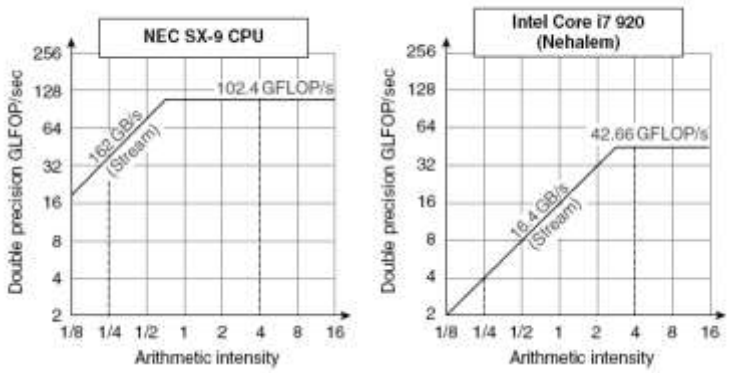
a=2, b=0, c=2, and d=-1

GCD(c,a)=2

d-b=-1

GCD(c,a)＝2 無法整除(d-b)=-1，所以不存在相依性

16. (5 pnts) 請解釋為何不同的處理器在效能呈現上會有如以下的相同趨勢？



當 Arithmetic Intensity 逐漸增加時，每秒能夠完成的浮點數指令逐漸增加，但是受限於記憶體頻寬的限制，之後就算 Arithmetic Intensity 繼續提升，也無法讓每秒指令執行個數增加。