

國立臺灣科技大學答案卷

National Taiwan University of Science and Technology Answer Sheet

姓名/Name 林怡薰

學號/Student ID B10902116

班級/Class 四電子四乙

科目/Course title 計算機組識

日期/Date 12/22

評分 Score	教師簽章 Signature of Lecturer
91	

記分欄 從此處開始寫起。試卷用紙務須節用，非經主試認可不得續用其他紙張作答。/Please write from here.

4

1. 當遇到工作量少、簡單的程式時，例如：看新聞、看影片等，選擇使用小核心。而遇到 AI 學習等重度使用下，選擇用大核心來執行。因為大、小核的電容不一樣，電容越小，能量消耗越小。因此，根據使用場景不同選擇大、小核的使用，可以降低功耗。

7

2. 因為如果控制訊號是拿前一階段執行過的結果來操作，那永遠是拿到錯的控制訊號，執行錯誤操作。因此，控制訊號需要跟 pipeline 的 stage 傳遞下去，下一階段需要做什麼操作，再把控制訊號抓出來使用。具體

10

3. cache 使用 write-back 的方式，只會改 cache 內的資料，並用 dirty bit 標示哪些資料修改過。只有當 dirty bit 為 1，也就是資料已經修改過的那些資料要被換走時，才會寫回 DRAM。如此一來，就不用 cache 被改，DRAM 也跟著要改，就可以降低 DRAM 被寫入的次數。✓

10

4. 因為有時候會有 hazard、cache miss 等情況發生，需要 stall 程式，等待資料都處理好了，才可以繼續執行。但 dynamic scheduling 可以讓 CPU 執行指令時不用按照順序，可以在需要暫停時插入不相干的指令執行，類似取代 stall，讓效率增加。但寫回暫存器時還是要按照順序寫入。✓

9

5. 因為 2-bit 的 predictor 只有在連續猜錯 2 次的情况下，才會猜相反的答案。因此若要跳出 ^{inner}for 迴圈時，整體會少錯一次，就可以減少猜測的錯誤率。

10

6. I-cache 的 cache miss ratio 總是比 D-cache 低是因為指令的順序比較固定，通常都是固定取下一道指令 PC+4，只有少部份情況會需要 branch。相較之下，Data 的順序比較不固定，所以，I-cache 的 cache miss ratio 總是比 D-cache 還低。✓

5

7. speculation 是在不確定資訊下進行猜測。例如：CPU 在執行指令時，會推測結果，儘管結果還不確定，再根據這個結果繼續執行程式。如果推測正確，就繼續執行；推測錯誤就 flush 掉再重新執行。而 speculation 與 prediction 的差異是在於是否有歷史的數據、資料。speculation 不會根據歷史記錄來猜測，而 prediction 會根據以往的經驗去推測，例如 branch prediction。

4

8. early restart 及 critical-word-first 是先傳需要的那個 word，先讓 pipeline 重新啟動，先繼續執行，後續剩下的 word 再慢慢傳輸就好。(。·。)

5

9. TLB 有 good locality 是因為 cache 一個 block 才只有幾個 word，而 Page 有 4k Bytes，相當於一千行指令。因為指令非常多，所以 hit rate 很高，miss rate 很低，TLB 不需要很大，因此就有 good locality。

3

10. Superscalar 是複製好幾個 pipeline，同時有好幾個 pipeline 一起執行，一個 cycle 執行好幾道指令。Superscalar 如下：



可轉頁再寫。
一個 cycle 同時執行 3 道指令 ✓

10 11. write-through: 當 cache 資料一被改, DRAM 資料也要跟著改, 否則會有資料不一致的問題。但因為 DRAM 太慢, 所以有 write-buffer 來存 Cache 的資料。cache 一被改, 就會把資料寫到 write-buffer 後繼續執行, 再由 write-buffer 慢慢寫到 DRAM。
write-back: 只會先改 cache 的資料, 並用 dirty bit 標示哪些資料有被改過, 只有當被改過的資料要被換掉時, 才會寫進 write-buffer 慢慢寫給 DRAM。如果要換掉的資料沒被改過, 那直接蓋掉即可。
在 cache read miss 時會使用到 write-back 的方式 ✓

5 12. TLB miss, Page Table miss, Cache hit 的情況不可能發生, 因為 cache 是子集合, 不可能子集合有的資料, 卻不存在在大的集合裡。不可能 cache 有的資料, Page 卻不在 memory 裡。因此, 這種情況不可能發生。✓

5 13. fully associative cache 不會發生 conflict misses。因為 conflict misses 只會在競爭 set 內的空間時才會發生。而 fully associate 是所有空間都可以存取, 因此, fully associative cache 不會有 conflict misses。✓

4 14. 4 核心以上的 CPU 最好使 4-way 以上的 associative, 因為如果使用太小的 associative, 會有較大的 block size, 雖然 miss rate 降低, 但同時會有更多資料要競爭同一個 set, miss rate 又被提高。並且 block size 變大, 一但要寫回 DRAM, 就需要寫回更多資料, miss penalty 增加。所以, 核心數量越多, 就需要增加 associativity, 讓 miss rate 降低, 減少 hit time。✓