**Computer Organization**

### Memory stall cycles

## One-word-wide memory organization

由於Cache只有一層，因此有cache hit與cache miss兩種情況。

1. 若發生cache hit，所需要的執行的動作為：send the address(1)、access single cache content(2)、send a word of data(1)，一次hit所需要的memory stall cycle為1+2+1 = 4 (cycle)。
2. 若發生cache miss，所需要執行的動作為：send the address(1)、[send the address(1) + access memory content(100) + send a word of data(1) + access single cache content(2) ]block size(8 words/block)、access single cache content(2)、send a word of data(1)，因此miss所需要的memory stall cycle為(cycle)。

## Wider memory organization

由於Cache只有一層，因此有cache hit與cache miss兩種情況，且access memory時，一次可以取得8 words。

1. 若發生cache hit，所需要的執行的動作為：send the address(1)、access single cache content(2)、send a word of data(1)，一次hit所需要的memory stall cycle為1+2+1 = 4 (cycle)。
2. 若發生cache miss，所需要執行的動作為：send the address(1)、[send the address(1) + access memory content(100) + send a word of data(1) + access single cache content(2) ]、access single cache content(2)、send a word of data(1)，因此miss所需要的memory stall cycle為(cycle)。

## Two-level memory organization

由於Cache有兩層，因此總共有三種情況，分別為L1 cache hit、L2 cache hit (L1 cache miss)、All miss (L1&L2 cache miss)。

1. 若發生L1 cache hit，所需要的執行的動作為：send the address(1)、access L1 cache content(1)、send a word of data(1)，一次L1 cache hit所需要的memory stall cycle為1+1+1 = 3 (cycle)。
2. 若發生L2 cache hit (L1 cache miss)，所需要執行的動作為：send the address(1)、[send the address(1) + access L2 cache content(10) + send a word of data(1) + access L1 cache content(1) ]block size(4 words/block)、access L1 cache content(1)、send a word of data(1)，因此L2 cache hit (L1 cache miss) 所需要的memory stall cycle為(cycle)。
3. 若發生All miss (L1&L2 cache miss)，所需要執行的動作為：send the address(1)、[send the address(1) + access memory content(100) + send a word of data(1) + access L2 cache content(10) ]block size(32 words/block) 、[send the address(1) + access memory content(10) + send a word of data(1) + access L1 cache content(1) ]block size(4 words/block)、access L1 cache content(1)、send a word of data(1)，因此All miss (L1&L2 cache miss) 所需要的memory stall cycle為(cycle)。

### Difference among the three memory organizations

Memory stall cycles

|  |  |  |  |
| --- | --- | --- | --- |
| Memory organization  Data | One-word-wide | Wider | Two-level |
| a1xb1(4x4 4x4) | 6016 | 1648 | 12144 |
| a2xb2(8x4 4x8) | 19072 | 5968 | 32736 |
| a3xb3(16x16 16x32) | 7626560 | 1068008 | 791552 |
| a4xb4(64x128 128x32) | 251426432 | 35098320 | 998152040 |

* One word-wide memory organization與Wider memory organization會發生miss與hit的次數會相同，因為它們之間的差別在於access memory時可以一次傳比較多的data到cache，因此若是發生cache miss時所需要的memory stall cycles可以少很多。
* Wider memory organization與Two-level memory organization的差別在於後者有2層的cache，使得Two-level memory organization若發生L2 cache hit時所需要的memory stall cycles非常少(55 cycle)，但是若發生All miss的情況時，所需要的cycle非常的多(3639 cycle)。
* 觀察四個測資simulation完的結果中可以發現，只有在a3xb3時Two-level memory organization的stall cycles才小於Wider memory organization，其他的都是大於，這是因為a3xb3所需要的資料很多都可以在第二層cache中找到，不需要到memory裡面找，所以使得Wider memory organization中cache miss的次數遠大於Two-level memory organization發生all miss的次數，才會有這樣的結果。

### Bonus: Performance improvement by software

Memory stall cycles

|  |  |  |  |
| --- | --- | --- | --- |
| Memory organization  Data | One-word-wide | Wider | Two-level |
| a1xb1(4x4 4x4) | 5568 | 1200 | 11808 |
| Performance improvement(cycle) | 448 | 448 | 336 |
| a2xb2(8x4 4x8) | 17280 | 4176 | 31392 |
| Performance improvement(cycle) | 1792 | 1792 | 1344 |
| a3xb3(16x16 16x32) | 7563072 | 1004520 | 763904 |
| Performance improvement(cycle) | 63488 | 63488 | 27648 |
| a4xb4(64x128 128x32) | 250827584 | 33195624 | 1002166580 |
| Performance improvement(cycle) | 598848 | 1902696 | -4014540 |

為了讓memory stall cycle減少，在重寫Assembly code方面主要是減少讀取C矩陣的次數，因為C矩陣只是要放矩陣乘法計算完的結果而已，所以不需要每次在相乘時都將C矩陣的內容讀取出來，只需要將乘法完的結果放進去就好了，因此在Memory stall cycle上面，可以看出明顯的下降。但是在a4xb4中，由於此方法會造成讀取C矩陣的頻率較不頻繁，所以發生All miss的情況略增，但一次的All miss的memory stall cycle又很龐大，使得performance improvement的情況沒有改善，但在其他測資中，這種方法可以明顯改善效能。