Si consideri il seguente frammento di codice C:

for (i = 0; i < 100; i++) {   
 v4[i] = v1[i]\*v2[i]; v5[i] = v1[i]+v2[i] + (v1[i]\*v3[i]);  
 }

dove i vettori v1[i], v2[i] e v3[i] contengono numeri Floating Point (FP), sono lunghi 100 e sono stati salvati in precedenza nella memoria. Si consideri un processore MIPS64 con le seguenti caratteristiche:   
· L’unità di moltiplicazione FP è un’unità pipelined a 4 stadi   
· L’unità aritmetica FP è un’unità pipelined a 2 stadi   
· L’unità di divisione FP è un’unica unità con una latenza pari a 4 colpi di clock   
· Il branch delay slot è pari ad 1 colpo di clock   
· Il delay slot non è abilitato (ossia, la pipeline viene svuotata se il salto viene preso)   
· Il data forwarding è abilitato.

Con riferimento al programma riportato nel seguito, scritto per l’architettura del processore MIPS64 descritta, ed utilizzando gli spazi a ciò appositamente destinati, si eseguano le seguenti operazioni:   
1) Si calcoli il numero di colpi di clock richiesti per l’esecuzione dell’intero programma.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Clock  cycles |
| V1: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V2: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V3: .double “100 values”  …  V5: .double “100 zeros” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| .text |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| main: daddui r1,r0,0 | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| daddui r2,r0,100 |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| loop: l.d f1,v1(r1) |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| l.d f2, v2(r1) |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| l.d f3,v3(r1) |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| mul.d f4,f1,f2 |  |  |  |  |  | F | D | X | X | X | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |
| s.d f4,v4(r1) |  |  |  |  |  |  | F | D | E | s | s | s | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| add.d f4,f1,f2 |  |  |  |  |  |  |  | F | D | s | s | s | + | + | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| mul.d f2,f1,f3 |  |  |  |  |  |  |  |  | F | s | s | s | D | X | X | X | X | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 |
| add.d f1,f4,f2 |  |  |  |  |  |  |  |  |  |  |  |  | F | D | s | s | s | + | + | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| s.d f1,v5(r1) |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | D | E | s | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddui r1,r1,8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | s | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddi r2,r2,-1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Bnez r2,loop |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| halt |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Total | 6+21\*100=2106 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2106 |

Si calcoli il numero di colpi di clock necessari all’esecuzione di 2 cicli del programma non ottimizzato proposto per un processore MIPS che implementa la strategia multiple-issue con scheduling dinamico e speculazione; si supponga che:   
· si possa eseguire l’issue di 2 istruzioni per colpo di clock   
· in presenza di un’istruzione di salto, venga eseguita una sola issue  
· si possa eseguire il commit di 2 istruzioni per colpo di clock   
· siano disponibili le seguenti unità funzionali indipendenti:  
 i. unità Memory address   
 ii. unità per operazioni intere (ALU)   
 iii. unità per il calcolo dei salti   
 iv. unità di moltiplicazione FP senza pipeline (latenza 4)   
 v. unità di divisione FP senz a pipeline (latenza 4)   
 vi. unità di somma e sottrazione FP no pipelined (latenza 2)

· la previsione sui salti sia sempre corretta   
· le cache non producano mai situazioni di miss   
· siano disponibili due CDB (Common Data Bus)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # iterazione |  | Issue | EXE | MEM | CDB x2 | COMMIT x2 |
| 1 | l.d f1,v1(r1) | 1 | 2M | 3 | 4 | 5 |
| 1 | l.d f2,v2(r1) | 1 | 3M | 4 | 5 | 6 |
| 1 | l.d f3,v3(r1) | 2 | 4M | 5 | 6 | 7 |
| 1 | mul.d f4,f1,f2 | 2 | 6X | - | 10 | 11 |
| 1 | s.d f4,v4(r1) | 3 | 5M | - | - | 11 |
| 1 | add.d f4,f1,f2 | 3 | 6A | - | 8 | 12 |
| 1 | mul.d f2,f1,f3 | 4 | 10X | - | 14 | 15 |
| 1 | add.d f1,f4,f2 | 4 | 15A | - | 17 | 18 |
| 1 | s.d f1,v5(r1) | 5 | 6M | - | - | 18 |
| 1 | daddui r1,r1,8 | 5 | 6I | - | 7 | 19 |
| 1 | daddi r2,r2,-1 | 6 | 7I | - | 8 | 19 |
| 1 | bnez r2,loop | 7 | 9J | - | - | 20 |
| 2 | l.d f1,v1(r1) | 8 | 9M | 10 | 11 | 20 |
| 2 | l.d f2,v2(r1) | 8 | 10M | 11 | 12 | 21 |
| 2 | l.d f3,v3(r1) | 9 | 11M | 12 | 13 | 21 |
| 2 | mul.d f4,f1,f2 | 9 | 14X | - | 18 | 22 |
| 2 | s.d f4,v4(r1) | 10 | 12M | - | - | 22 |
| 2 | add.d f4,f1,f2 | 10 | 13A | - | 15 | 23 |
| 2 | mul.d f2,f1,f3 | 11 | 18X | - | 22 | 23 |
| 2 | add.d f1,f4,f2 | 11 | 23A | - | 25 | 26 |
| 2 | s.d f1,v5(r1) | 12 | 13M | - | - | 26 |
| 2 | daddui r1,r1,8 | 12 | 13I | - | 14 | 27 |
| 2 | daddi r2,r2,-1 | 13 | 14I | - | 15 | 27 |
| 2 | bnez r2,loop | 14 | 16J | - | - | 28 |