Considerare la seguente architectura MIPS64:

|  |  |  |
| --- | --- | --- |
| * + Integer ALU: 1 clock cycle   + Data memory: 1 clock cycle   + FP multiplier unit: pipelined 6 stages | * + FP arithmetic unit: pipelined 3 stages   + FP divider unit: not pipelined unit that requires 8 clock cycles   + branch delay slot: 1 clock cycle, and the branch delay slot disabled | * + forwarding enabled   + è possibile completare lo stage EXE di una istruzion in modo out-of-order. |

* Facendo riferimento al frammento di codice riportato, si mostrino le tempistiche relative all’esecuzione ciascuna istruzione e si calcoli il numero totale di clock cycles necessari per eseguire completamente il programma:

acc = 0.0;

for (i = 0; i < 100; i++) {

v5[i] = acc + ((v1[i]\*v2[i]) \* (v3[i]/v4[i]));

acc = v5[i]; }

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Clock  cycles |
| V1: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V2: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V3: .double “100 values”  …  V5: .double “100 zeros” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V4: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V5: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ACC: .double 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| .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 |
| mul.d f5,f1,f2 |  |  |  |  | F | D | s | x | x | x | x | x | x | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |
| l.d f3,v3(r1) |  |  |  |  |  | F | s | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| l.d f4,v4(r1) |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| div.d f6,f3,f4 |  |  |  |  |  |  |  |  | F | D | s | : | : | : | : | : | : | : | : | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| mul.d f5,f5,f6 |  |  |  |  |  |  |  |  |  | F | s | D | s | s | s | s | s | s | s | x | x | x | x | x | x | M | W |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| l.d f7, ACC(r0) |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | s | s | s | s | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| add.d f7, f5, f7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | s | s | s | s | + | + | + | M | W |  |  |  |  |  |  |  |  |  | 3 |
| s.d f7,v5(r1) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | s | D | s | s | E | M | W |  |  |  |  |  |  |  |  | 1 |
| s.d f7,ACC(r0) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | D | E | M | W |  |  |  |  |  |  |  | 1 |
| daddui r1,r1,8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  | 1 |
| daddi r2,r2,-1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  | 1 |
| bnez r2,loop |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | D | E | M | W |  |  |  | 2 |
| halt |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  | 1 |
| Total | 6+31\*100=3106 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 3106 |

Condiderando un programma basato su loop, ed assumendo che il processore utilizzato sia un MIPS64 che implementa multiple-issue e speculation:

* + Issue di 2 instruzioni per clock cycle
  + Instruzioni jump richiedono 1 issue
  + Esegue il commit di 2 istruzioni per clock cycle
  + Le unità funzionali hanno le seguenti caratteristiche:
    1. 1 Memory address 1 clock cycle
    2. 1 Integer ALU 1 clock cycle
    3. 1 Jump unit 1 clock cycle
    4. 1 FP multiplier unit, which is pipelined: 6 stages
    5. 1 FP divider unit, which is not pipelined: 8 clock cycles
    6. 1 FP Arithmetic unit, which is pipelined: 3 stages
  + La predizione di salto è sempre corretta
  + Non ci sono cache misses
  + Esitono 2 CDB (Common Data Bus).
* Si completi la tabella mostrando il comportamento del processore durante le 2 iniziali iterazioni

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # iterazione | Instruction | ISSUE | EXE | MEM | CDBx2 | COMMITx2 |
| 1 | l.d f1,v1(r1) | 1 | 2M | 3 | 4 | 5 |
| 1 | l.d f2,v2(r1) | 1 | 3M | 4 | 5 | 6 |
| 1 | mul.d f5,f1,f2 | 2 | 6X | - | 12 | 13 |
| 1 | l.d f3,v3(r1) | 2 | 4M | 5 | 6 | 13 |
| 1 | l.d f4,v4(r1) | 3 | 5M | 6 | 7 | 14 |
| 1 | div.d f6,f3,f4 | 3 | 8D | - | 16 | 17 |
| 1 | mul.d f5,f5,f6 | 4 | 17X | - | 23 | 24 |
| 1 | l.d f7, ACC(r0) | 4 | 6M | 7 | 8 | 24 |
| 1 | add.d f7, f5, f7 | 5 | 24A | - | 27 | 28 |
| 1 | s.d f7,v5(r1) | 5 | 7M | - | - | 28 |
| 1 | s.d f7,ACC(r0) | 6 | 8M | - | - | 29 |
| 1 | daddui r1,r1,8 | 6 | 7I | - | 8 | 29 |
| 1 | daddi r2,r2,-1 | 7 | 8I | - | 9 | 30 |
| 1 | bnez r2,loop | 8 | 10J | - | - | 30 |
| 2 | l.d f1,v1(r1) | 9 | 10M | 11 | 12 | 31 |
| 2 | l.d f2,v2(r1) | 9 | 11M | 12 | 13 | 31 |
| 2 | mul.d f5,f1,f2 | 10 | 14X | - | 20 | 32 |
| 2 | l.d f3,v3(r1) | 10 | 12M | 13 | 14 | 32 |
| 2 | l.d f4,v4(r1) | 11 | 13M | 14 | 15 | 33 |
| 2 | div.d f6,f3,f4 | 11 | 16D | - | 24 | 33 |
| 2 | mul.d f5,f5,f6 | 12 | 25X | - | 31 | 34 |
| 2 | l.d f7, ACC(r0) | 12 | 14M | 15 | 16 | 34 |
| 2 | add.d f7, f5, f7 | 13 | 32A | - | 35 | 36 |
| 2 | s.d f7,v5(r1) | 13 | 15M | - | - | 36 |
| 2 | s.d f7,ACC(r0) | 14 | 16M | - | - | 37 |
| 2 | daddui r1,r1,8 | 14 | 15I | - | 17 | 37 |
| 2 | daddi r2,r2,-1 | 15 | 17I | - | 18 | 38 |
| 2 | bnez r2,loop | 16 | 19j | - | - | 38 |