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 6 clock cycles   + branch delay slot: 1 clock cycle, and the branch delay slot disabilitato | * + forwarding abilitato   + 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 comletamente il programma.

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

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| .data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Clock  cycles |
| V1: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V2: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V3: .double “100 values”  …  V5: .double “100 zeros” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| .text |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| main: daddui r1,r0,8\*100 | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| loop: l.d f1,v1(r1) |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| l.d f2,v2(r1) |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| div.d f5,f1,f2 |  |  |  | F | D | s | / | / | / | / | / | / | 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,f5,f3 |  |  |  |  |  |  |  | F | D | s | s | s | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| add.d f6,f6,f4 |  |  |  |  |  |  |  |  | F | s | s | s | D | s | s | s | s | s | + | + | + | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 |
| s.d f6,v5(r1) |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | s | s | D | E | s | S | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddui r1,r1,-8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **F** | **D** | **s** | **S** | **E** | **M** | **W** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **1** |
| bnez r1,loop |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **F** | **s** | **S** | **s** | **D** | **E** | **M** | **W** |  |  |  |  |  |  |  |  |  |  |  |  | **2** |
| halt |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **F** | **-** | **-** | **-** | **-** |  |  |  |  |  |  |  |  |  |  |  | **1** |
| Total | 5+23\*100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2305 |

Considerato 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
  + Esegui 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: 8 stages
    5. 1 FP divider unit, which is not pipelined: 8 clock cycles
    6. 1 FP Arithmetic unit, which is pipelined: 4 stages
  + La predizione di salto è sempre corretta
  + Non ci sono cache misses
  + Essitono 2 CDB (Common Data Bus).
* Si complete la tabella mostrando il comportamento del processore durante le 3 iniziali iterazioni

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # iterazione | Instruction | ISSUE | EXE | MEM | CDBx2 | COMMITx2 |
| 1 | l.d f1,v1(r1) | 1 | 2ea | 3 | 4 | 5 |
| 1 | l.d f2,v2(r1) | 1 | 3ea | 4 | 5 | 6 |
| 1 | mul.d f1,f1,f1 | 2 | 5m-12m |  | 13 | 14 |
| 1 | mul.d f2,f2,f2 | 2 | 6m-13m |  | 14 | 15 |
| 1 | div.d f5,f1,f2 | 3 | 15d-22d |  | 23 | 24 |
| 1 | s.d f5,v3(r1) | 3 | 4ea |  |  | 24 |
| 1 | daddui r1,r1,-8 | 4 | 5i |  | 6 | 25 |
| 1 | bnez r1,loop | 5 | 7j |  |  | 25 |
| 2 | l.d f1,v1(r1) | 6 | 7ea | 8 | 9 | 26 |
| 2 | l.d f2,v2(r1) | 6 | 8ea | 9 | 10 | 26 |
| 2 | mul.d f1,f1,f1 | 7 | 10m-17m |  | 18 | 27 |
| 2 | mul.d f2,f2,f2 | 7 | 11m-18m |  | 19 | 27 |
| 2 | div.d f5,f1,f2 | 8 | 23d-30d |  | 31 | 32 |
| 2 | s.d f5,v3(r1) | 8 | 9ea |  |  | 32 |
| 2 | daddui r1,r1,-8 | 9 | 10i |  | 11 | 33 |
| 2 | bnez r1,loop | 10 | 12j |  |  | 33 |
| 3 | l.d f1,v1(r1) | 11 | 12ea | 13 | 14 | 34 |
| 3 | l.d f2,v2(r1) | 11 | 13ea | 14 | 15 | 34 |
| 3 | mul.d f1,f1,f1 | 12 | 15m-22m |  | 23 | 35 |
| 3 | mul.d f2,f2,f2 | 12 | 16m-23m |  | 24 | 35 |
| 3 | div.d f5,f1,f2 | 13 | 31d-38d |  | 39 | 40 |
| 3 | s.d f5,v3(r1) | 13 | 14ea |  |  | 40 |
| 3 | daddui r1,r1,-8 | 14 | 15i |  | 16 | 41 |
| 3 | bnez r1,loop | 15 | 17j |  |  | 41 |