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 2 stages   + FP divider unit: not pipelined unit that requires 7 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:

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

v5[i] = v1[i]/v2[i];

v6[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” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V4: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V5: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| V6: .double “100 values” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| .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 |
| l.d f4,v4(r1) |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| div.d f5,f1,f2 |  |  |  |  |  |  | F | D | / | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |
| s.d f5,v5(r1) |  |  |  |  |  |  |  | F | D | E | s | s | s | s | s | S | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| mul.d f6,f1,f2 |  |  |  |  |  |  |  |  | F | D | \* | \* | \* | \* | \* | \* | S | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| div.d f7,f3,f4 |  |  |  |  |  |  |  |  |  | F | D | S | S | S | S | / | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |
| add.d f1,f6,f7 |  |  |  |  |  |  |  |  |  |  | F | s | s | s | s | D | s | s | s | s | s | s | + | + | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| s.d f1,v6(r1) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | 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 | - | - | - | - |  |  |  |  |  |  |  | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Total | 6+100\*26 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2606 |

**Question 2**

Considerando il programma precedente e l’architettura del processore superscalare descritto in seguito; completare la tabella relativa alle prime 2 iterazioni.

Processor architecture:

* + Issue 2 instructions per clock cycle
  + jump instructions require 1 issue
  + handle 2 instructions commit per clock cycle
  + timing facts for the following separate functional units:
    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: 7 clock cycles
    6. 1 FP Arithmetic unit, which is pipelined: 2 stages
  + Branch prediction is always correct
  + There are no cache misses
  + There are 2 CDB (Common Data Bus).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # iteration |  | Issue | EXE | MEM | CDB x2 | COMMIT x2 |
| 1 | l.d f1,v1(r1) | 1 | 2ea | 3 | 4 | 5 |
| 1 | l.d f2,v2(r1) | 1 | 3ea | 4 | 5 | 6 |
| 1 | l.d f3,v3(r1) | 2 | 4ea | 5 | 6 | 7 |
| 1 | l.d f4,v4(r1) | 2 | 5ea | 6 | 7 | 8 |
| 1 | div.d f5,f1,f2 | 3 | 6d-12d | / | 13 | 14 |
| 1 | s.d f5,v5(r1) | 3 | 6ea | / | / | 15 |
| 1 | mul.d f6,f1,f2 | 4 | 6m-11m | / | 12 | 15 |
| 1 | div.d f7,f3,f4 | 4 | 13d-19d | / | 20 | 21 |
| 1 | add.d f1,f6,f7 | 5 | 21a-22a | / | 23 | 24 |
| 1 | s.d f1,v6(r1) | 5 | 7ea | / | / | 25 |
| 1 | daddui r1,r1,8 | 6 | 7i | / | 8 | 25 |
| 1 | daddi r2,r2,-1 | 6 | 8i | / | 9 | 26 |
| 1 | bnez r2,loop | 7 | 10j | / | / | 26 |
| 2 | l.d f1,v1(r1) | 8 | 9ea | 10 | 11 | 27 |
| 2 | l.d f2,v2(r1) | 8 | 10ea | 11 | 12 | 27 |
| 2 | l.d f3,v3(r1) | 9 | 11ea | 12 | 13 | 28 |
| 2 | l.d f4,v4(r1) | 9 | 12ea | 13 | 14 | 28 |
| 2 | div.d f5,f1,f2 | 10 | 20d-26d | / | 27 | 29 |
| 2 | s.d f5,v5(r1) | 10 | 13ea | / | / | 29 |
| 2 | mul.d f6,f1,f2 | 11 | 13m-18m | / | 19 | 30 |
| 2 | div.d f7,f3,f4 | 11 | 27d-33d | / | 34 | 35 |
| 2 | add.d f1,f6,f7 | 12 | 35a-36a | / | 37 | 38 |
| 2 | s.d f1,v6(r1) | 12 | 14ea | / | / | 39 |
| 2 | daddui r1,r1,8 | 13 | 14i | / | 15 | 39 |
| 2 | daddi r2,r2,-1 | 13 | 15i | / | 16 | 40 |
| 2 | bnez r2,loop | 14 | 17j | / | / | 40 |