Considerando il processore MIPS64 e l’architettura descritta in seguito:

|  |  |  |
| --- | --- | --- |
| * + 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 6 clock cycles   + branch delay slot: 1 clock cycle, and the branch delay slot disabled | * + forwarding enabled   + it is possible to complete instruction EXE stage in an out-of-order fashion. |

Usando il frammento di codice riportato, si calcoli il tempo di esecuzione dell’intero programma in colpi di clock e si completi la seguente tabella.

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

; v5[i] = v1[i]\*v2[i];

; v6[i] = ((v2[i]/v1[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 |
| mul.d f5,f1,f2 |  |  |  |  |  |  | F | D | \* | \* | \* | \* | \* | \* | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| s.d f5,v5(r1) |  |  |  |  |  |  |  | F | D | E | s | s | s | s | S | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| div.d f5,f2,f1 |  |  |  |  |  |  |  |  | F | D | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| mul.d f6,f3,f4 |  |  |  |  |  |  |  |  |  | F | D | \* | \* | \* | \* | \* | \* | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| div.d f7,f5,f6 |  |  |  |  |  |  |  |  |  |  | F | D | s | s | s | s | s | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| s.d f7,v6(r1) |  |  |  |  |  |  |  |  |  |  |  | F | D | E | s | s | s | s | s | s | s | s | s | S | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddui r1,r1,8 |  |  |  |  |  |  |  |  |  |  |  |  | F | D | s | s | s | s | s | s | s | s | s | S | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddi r2,r2,-1 |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | s | s | s | s | s | s | s | s | S | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| bnez r2,loop |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | s | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 |
| halt |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | - | - | - | - |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Total |  |  |  |  | 6+25\*100 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2506 |

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: 6 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).

ea = effective address per load e store (exe)

mem = solo nella load, le alter vuote (anche store)

cdb = tutte apparte store e branch (perchè non scrivono su registro)

per la dipendenza vedi dove I dati sono messi nel cdb e parti da clock dopo

commit = segue l’ordine (ROB), non ci sono Buchi

posso fare 6d perchè prendo da cdb e non da dove sono arrivato (qua div f2, f1, basta che prendo f2 dal cdb che è 5, quindi faccio 6)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # 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 | mul.d f5,f1,f2 | 3 | 6m-11m | / | 12 | 13 |
| 1 | s.d f5,v5(r1) | 3 | 6ea | / | / | 13 |
| 1 | div.d f5,f2,f1 | 4 | 6d-11d | / | 12 | 14 |
| 1 | mul.d f6,f3,f4 | 4 | 8m-13m | / | 14 | 15 |
| 1 | div.d f7,f5,f6 | 5 | 15d-20d | / | 21 | 22 |
| 1 | s.d f7,v6(r1) | 5 | 7ea | / | / | 22 |
| 1 | daddui r1,r1,8 | 6 | 7i | / | 8 | 23 |
| 1 | daddi r2,r2,-1 | 6 | 8i | / | 9 | 23 |
| 1 | bnez r2,loop | 7 | 10j | / | / | 24 |
| 2 | l.d f1,v1(r1) | 8 | 9ea | 10 | 11 | 24 |
| 2 | l.d f2,v2(r1) | 8 | 10ea | 11 | 13 | 25 |
| 2 | l.d f3,v3(r1) | 9 | 11ea | 12 | 13 | 25 |
| 2 | l.d f4,v4(r1) | 9 | 12ea | 13 | 14 | 26 |
| 2 | mul.d f5,f1,f2 | 10 | 14m-19m | / | 20 | 26 |
| 2 | s.d f5,v5(r1) | 10 | 13ea | / | / | 27 |
| 2 | div.d f5,f2,f1 | 11 | 21d-26d | / | 27 | 28 |
| 2 | mul.d f6,f3,f4 | 11 | 15m-20m | / | 21 | 28 |
| 2 | div.d f7,f5,f6 | 12 | 27d-32d | / | 33 | 34 |
| 2 | s.d f7,v6(r1) | 12 | 14ea | / | / | 34 |
| 2 | daddui r1,r1,8 | 13 | 14i | / | 15 | 35 |
| 2 | daddi r2,r2,-1 | 13 | 15i | / | 16 | 35 |
| 2 | bnez r2,loop | 14 | 17j | / | / | 36 |

Quale istruzione stalla nel ROB se ho 16 posti (linee)? Dovrei prendere la 16^ istruzione (la 16^ riga della tabella), la 17^ va in stallo, solo se la sua ISSUE avviene prima del commit della 1^ istruzione delle 16 da mettere nel ROB. Qui per esempio dovrei prendere la riga blu, ma il suo issue avviene a 9 e le istruzioni precedenti nel frattempo fanno commit ed escono dal rob, quindi ne escono 4; a quel punto vado sotto di 4 istruzioni e ricontrollo e vedo che la issue avviene a 11, mentre la prima commit delle istruzioni avviene a 13 quindi ok)