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 9 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++) {

; v4[i] = ((v1[i]\*v2[i])/v3[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” |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| .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 | m | m | m | m | m | m | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |
| div.d f4,f4,f3 |  |  |  |  |  |  | F | D | - | - | - | - | - | / | / | / | / | / | / | / | / | / | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 9 |
| s.d f4,v4(r1) |  |  |  |  |  |  |  | F | - | - | - | - | - | D | - | - | - | - | - | - | - | - | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddi r2,r2,-1 |  |  |  |  |  |  |  |  |  |  |  |  |  | F | - | - | - | - | - | - | - | - | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| daddui r1,r1,8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| bnez r2,loop |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |
| Halt |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Total |  |  |  |  | TOT = 6 + 22\*100 = **2206** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |  |

**Domanda 2**

Considerando il programma precedente, e in particolare la copia di istruzioni:

mul.d f4,f1,f2

div.d f4,f4,f3

come viene attivato e qual è il cammino di forwarding che partecipa alla loro esecuzione? Motivare la risposta

Il cammino di forwarding permette di risolvere problemi di data hazard. In questo caso le due istruzioni sono legate dal dato contenuto in f4, il cui valore viene prodotto al termine della mul e che serve come operando della div. La dipendenza tra queste istruzioni impedisce l’esecuzione delle istruzioni out-of-order, in quanto la div non può iniziare prima che la mul abbia prodotto il valore di f4.

In questo caso la presenza del forwarding permette di rendere disponibile alla div il valore di f4 subito dopo la fase di esecuzione, senza dover attendere il write-back, riducendo i ritardi legati alle dipendenze tra dati.

**Domanda 3**

Considerando il programma precedente e l’architettura del processore superscalare descritto in seguito; completare la tabella relativa alle prime 3 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: 9 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 | 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 | 6\*(+6) |  | 12 | 13 |
| 1 | div.d f4,f4,f3 | 3 | 13d |  | 22 | 23 |
| 1 | s.d f4,v4(r1) | 3 | 5m |  |  | 23 |
| 1 | daddi r2,r2,-1 | 4 | 5i |  | 6 | 24 |
| 1 | daddui r1,r1,8 | 4 | 6i |  | 7 | 24 |
| 1 | bnez r2,loop | 5 | 7j |  |  | 25 |
| 2 | l.d f1,v1(r1) | 6 | 8m | 9 | 10 | 25 |
| 2 | l.d f2,v2(r1) | 6 | 9m | 10 | 11 | 26 |
| 2 | l.d f3,v3(r1) | 7 | 10m | 11 | 12 | 26 |
| 2 | mul.d f4,f1,f2 | 7 | 12\* |  | 18 | 27 |
| 2 | div.d f4,f4,f3 | 8 | 22d |  | 31 | 32 |
| 2 | s.d f4,v4(r1) | 8 | 11m |  |  | 32 |
| 2 | daddi r2,r2,-1 | 9 | 10i |  | 11 | 33 |
| 2 | daddui r1,r1,8 | 9 | 11i |  | 13 | 33 |
| 2 | bnez r2,loop | 10 | 12j |  |  | 34 |
| 3 | l.d f1,v1(r1) | 11 | 14m | 15 | 16 | 34 |
| 3 | l.d f2,v2(r1) | 11 | 15m | 16 | 17 | 35 |
| 3 | l.d f3,v3(r1) | 12 | 16m | 17 | 18 | 35 |
| 3 | mul.d f4,f1,f2 | 12 | 18\* |  | 25 | 36 |
| 3 | div.d f4,f4,f3 | 13 | 31d |  | 40 | 41 |
| 3 | s.d f4,v4(r1) | 13 | 17m |  |  | 41 |
| 3 | daddi r2,r2,-1 | 14 | 15i |  | 16 | 42 |
| 3 | daddui r1,r1,8 | 14 | 16i |  | 17 | 42 |
| 3 | bnez r2,loop | 15 | 17j |  |  | 43 |

**Domanda 4**

Considerando il segmento di codice presentato nella tabella precedente, se assumiamo che il ROB ha una dimensione di 8 elementi, qual è la prima istruzione che dovrebbe stallare durante la esecuzione del programma? motivare la risposta.

|  |  |  |
| --- | --- | --- |
| 2 | l.d f2,v2(r1) | 6 |

* Le prime 8 istruzioni non hanno problemi.
* Successivamente si liberano due posizioni grazie ai commit che avvengono al colpo di clock 5 e 6