

Considerare la seguente architettura MIPS64:

- 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:

.data

ACC: .double 0

.text

1.d f3,v3(r1)

[illegible]

A1

3106

[illegible]

Name, Student ID

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

- Issue di 2 istruzioni per clock cycle
- Istruzioni jump richiedono 1 issue
- Esegue il commit di 2 istruzioni per clock cycle
- Le unità funzionali hanno le seguenti caratteristiche:
 - i. 1 Memory address 1 clock cycle
 - ii. 1 Integer ALU 1 clock cycle
 - iii. 1 Jump unit 1 clock cycle
 - iv. 1 FP multiplier unit, which is pipelined: 6 stages
 - v. 1 FP divider unit, which is not pipelined: 8 clock cycles
 - vi. 1 FP Arithmetic unit, which is pipelined: 3 stages
- La predizione di salto è sempre corretta
- Non ci sono cache misses
- Esistono 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	2	3	4	5
1	l.d f2,v2(r1)	1	3	4	5	6
1	mul.d f5,f1,f2	2	6	—	12	13
1	l.d f3,v3(r1)	2	4	5	6	13
1	l.d f4,v4(r1)	3	5	6	7	14
1	div.d f6,f3,f4	3	8	—	16	17
1	mul.d f5,f5,f6	4	17	—	23	24
1	l.d f7, ACC(r0)	4	6	7	8	24
1	add.d f7, f5, f7	5	24	—	27	28
1	s.d f7,v5(r1)	5	7	—	—	28

1 Luglio 2020 – ASE – Architetture moderne

A1

Name, Student ID

1	s.d f7,ACC(r0)	6	8	—	—	29
1	daddui r1,r1,8	6	7	—	8	29
1	daddi r2,r2,-1	7	8	—	9	30
1	bnez r2,loop	8	10	—	—	30
2	l.d f1,v1(r1)	9	10	11	12	31
2	l.d f2,v2(r1)	9	11	12	13	31
2	mul.d f5,f1,f2	10	14	—	20	32
2	l.d f3,v3(r1)	10	12	13	14	32
2	l.d f4,v4(r1)	11	13	14	15	33
2	div.d f6,f3,f4	11	16	—	24	33
2	mul.d f5,f5,f6	12	25	—	31	34
2	l.d f7, ACC(r0)	12	14	15	16	34
2	add.d f7, f5, f7	13	32	—	35	36
2	s.d f7,v5(r1)	13	15	—	—	36
2	s.d f7,ACC(r0)	14	16	—	—	37
2	daddui r1,r1,8	14	15	—	17	37
2	daddi r2,r2,-1	15	17	—	18	38
2	bnez r2,loop	16	19	—	—	38