

Domanda 10 (8 punti)

Si consideri il seguente frammento di codice C:

```
for (i = 0; i < 100; i++) {
    v5[i] = (v1[i]*v2[i]*v3[i])/v4[i];
}
```

dove i vettori $v1[]$, $v2[]$, $v3[]$ e $v4[]$ ($v4[]$ non è mai 0) contengono numeri Floating Point (FP), sono lunghi 100 e sono stati salvati in precedenza nella memoria. Inoltre è stato allocato in memoria il vettore vuoto $v5[]$.

Si eseguano le seguenti operazioni:

- Con riferimento al programma riportato nel seguito, scritto per l'architettura del processore MIPS64 (le cui caratteristiche sono riportate sotto), ed utilizzando gli spazi a ciò appositamente destinati, si calcoli il numero di colpi di clock richiesti per l'esecuzione dell'intero programma. L'architettura da considerare ha le seguenti caratteristiche:
 - L'unità di moltiplicazione FP è un'unità pipelined a 8 stadi
 - L'unità aritmetica FP è un'unità pipelined a 2 stadi
 - L'unità di divisione FP è un'unico blocco con una latenza pari a 12 colpi di clock
 - Il branch delay slot è pari ad 1 colpo di clock
 - Il delay slot non è abilitato (ossia, la pipeline viene svuotata se il salto viene preso)
 - Il data forwarding è abilitato.
- Con riferimento all'architettura di un processore MIPS che implementa la strategia multiple-issue con speculazione (descritta sotto), si calcoli il numero di colpi di clock necessari all'esecuzione di 2 cicli del programma proposto. L'architettura da considerare ha le seguenti caratteristiche:
 - può eseguire l'issue di 2 istruzioni per colpo di clock
 - in presenza di un'istruzione di salto, viene eseguita una sola issue
 - può eseguire il commit di 2 istruzioni per colpo di clock
 - dispone delle seguenti unità funzionali indipendenti:
 - unità Memory address
 - unità per operazioni intere (ALU)
 - unità per il calcolo dei salti
 - unità di moltiplicazione FP pipelined (latenza 8)
 - unità di divisione FP non pipelined (latenza 12)
 - unità di somma e sottrazione FP no pipelined (latenza 2)
 - la previsione sui salti è sempre corretta
 - le cache non producono mai situazioni di miss
 - sono disponibili due CDB (Common Data Bus).

Punto 1)

		Commenti	Colpi di clock
;			
.data			
V1: .double "100 valori"			
V2: .double "100 valori"			
V3: .double "100 valori"			
V4: .double "100 valori"			
V5: .double "100 zeri"			
.text			
main:	daddui r1,r0,0	$r1 \leftarrow \text{puntatore}$	
	daddui r2,r0,100	$r2 \leftarrow 100$	
loop:	l.d f1,v1(r1)	$f1 \leftarrow v1[i]$	
	l.d f2,v2(r1)	$f2 \leftarrow v2[i]$	
	mul.d f6,f1,f2	$f6 \leftarrow v1[i]*v2[i]$	
	l.d f3,v3(r1)	$f3 \leftarrow v3[i]$	
	mul.d f5,f6,f3	$f5 \leftarrow v1[i]*v2[i]*v3[i]$	
	l.d f4,v4(r1)	$f4 \leftarrow v4[i]$	
	div.d f5,f5,f4	$f5 \leftarrow v1[i]*v2[i]*v3[i]/v4[i]$	
	s.d f5,v5(r1)		
	daddui r1,r1,8	$r1 \leftarrow r1 + 8$	
	daddi r2,r2,-1	$r2 \leftarrow r2 - 1$	
	bnez r2,loop		
	halt		
totale			

Punto 2)

# iterazione		Issue	EXE	MEM	CDB x2	COMMIT x2
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 f6,f1,f2	2	6		14	1
1	l.d f3,v3(r1)	2	4	5	6	7
1	mul.d f5,f6,f3	3	15		23	24
1	l.d f4,v4(r1)	3	5	6	7	24
1	div.d f5,f5,f4	4	25		37	38
1	s.d f5,v5(r1)	4	6			38
1	daddui r1,r1,8	5	6		7	39
1	daddi r2,r2,-1	5	7		8	39
1	bnez r2,loop	6	9			40
2	l.d f1,v1(r1)	7	8	9	10	41
2	l.d f2,v2(r1)	7	9	10	11	41
2	mul.d f6,f1,f2	8	12		20	42
2	l.d f3,v3(r1)	8	10	12	13	42
2	mul.d f5,f6,f3	9	21		29	43
2	l.d f4,v4(r1)	9	11	12	13	43
2	div.d f5,f5,f4	10	38		50	51
2	s.d f5,v5(r1)	10	12			52
2	daddui r1,r1,8	11	12		13	14
2	daddi r2,r2,-1	11	13		14	15
2	bnez r2,loop	12	15			16

	unit	latency
m	Memory address unit	1
x	FP Multiplier	8
a	FP Adder	2
d	FP Divider	12
i	integer ALU	1
j	jump unit	1

I primi 2 cicli sono eseguiti in _____ colpi di clock.