

1 DIV f_2, f_9, f_3
 2 MUL f_1 , f_9, f_3
 3 ADD f_4, f_9, f_3
 4 DIV f_5, f_1, f_2
 5 ADD f_5, f_3, f_1
 6 DIV f_4, f_5, f_3

RAW f_2, f_1

RAW f_1

RAW f_5

Esempio Esercizio

$1 \rightarrow 4$

$2 \rightarrow 4$

$2 \rightarrow 5 \rightarrow 6$



Esercizio

- **Pipeline non bloccante con più unità funzionali multiciclo**
 - Un RV con $T_{ck} = T$ dispone di **tre** unità funzionali, A, M e D “multiciclo” capaci di eseguire le seguenti istruzioni su operandi in virgola mobile:
 - **A: fadd (in 2T)**
 - **M: fmul (in 3T)**
 - **D: fdiv (in 4T).**
- } *tempistiche per l'esecuzione.*
- Si faccia l'ipotesi che in T1 si abbia $f_i = i$ per ogni valore di i compreso tra 0 e 31. Si consideri il seguente frammento di codice:

1. fadd **f20**, f2, f1
2. fdiv **f2**, f2, **f20**
3. fmul f6, f1, **f2**
4. fmul f25, f25, f3
5. fmul **f1**, f9, f10
6. fadd f9, f3, **f1**

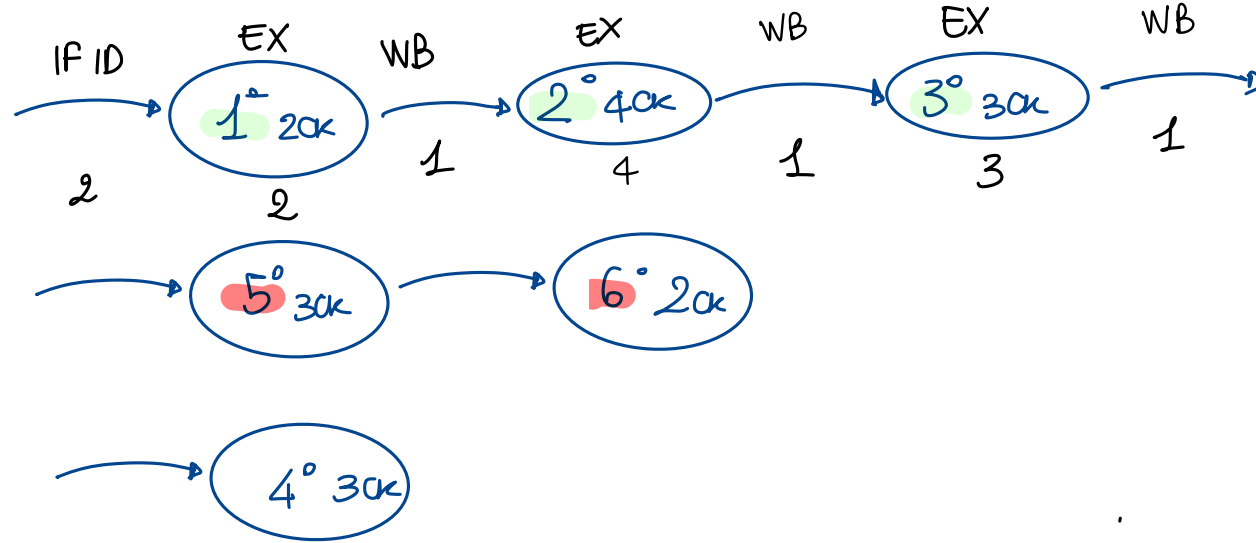
Quesiti

- a) • Stimare il numero di colpi di clock al di sotto del quale non è possibile scendere nell'esecuzione del codice assegnato, qualunque sia il numero di RS, CRB e stadi di Fetch e Decode disponibili, e si motivi la risposta (*punti 2*) *a 1 dipendenze di dato a 2 dipendenze strutturali*
- b) • mostrare la dinamica dell'esecuzione nel caso di 1 CRB, uno stadio IF e uno stadio ID, e **2 RS** per ogni unità funzionale (*punti 4*)
↓
common data bus
- c) • Si supponga ora di poter apportare le seguenti modifiche all'architettura:
1. aumento di RS a una o più unità funzionali
 2. raddoppio del CRB per poter eseguire due fasi di WB per clock
 3. aggiunta di un'altra unità funzionale con due RS
 4. raddoppio degli stadi IF e ID per poter decodificare contemporaneamente due istruzioni.
- d) • Quali modifiche converrebbe apportare? Nella soluzione scelta si mostri quanti clock si risparmierebbero rispetto al punto 2 (*a tal fine si disegni la dinamica della pipeline risultante*) e si indichi quale CPI medio si otterrebbe (*punti 4*)
→ ridurre il tempo di esecuzione

Affinché l'istruzione 3 possa terminare devono già essere terminate le 1 e le 2.

Q.1 Considero infinite risorse di unità funzionali (ma dipendenti)

1. fadd **f20**, f2, f1
2. fdiv **f2**, f2, **f20**
3. fmul f6, f1, **f2**
4. fmul f25, f25, f3
5. fmul **f1**, f9, f10
6. fadd f9, f3, **f1**



$$T_{MIN} = 14 T_{ck}$$

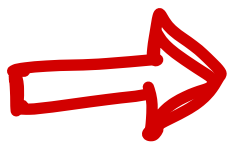
$$T_{MIN} = 9 T_{ck}$$

$$T_{MIN} = 6 T_{ck}$$

Q.2 Considero il numero reale di unità fu ma nessuna dipendenza di dato

ci sono 3 istruzioni che hanno bisogno del moltiplicatore, abbiamo che in EX

$$\begin{aligned} M: 3 \times 3 &= 9 + 1 + 1 + 1 = 12 T_{ck-min} \\ D: 1 \times 4 &= 4 = 7 T_{ck-min} \\ A: 2 \times 2 &= 4 = 7 T_{ck-min} \end{aligned}$$



- a) • considerando solo dipendenze di dato: $14 T_{ck}$
 • considerando solo le dipendenze strutturali: $12 T_{ck}$

Nota: si è fatta l'assunzione, i dati viaggiano nel common data bus nel primo semiperiodo di ck .

Nel 2° semiperiodo di ck possiamo assumere che i dati siano stati recapiti.

2° assunzione: l'ID risolve le albe strutturali nel 1° semiperiodo e accede al RF nel 2° semiperiodo.

1. fadd **f20**, f2, f1
2. fdiv **f2**, f2, **f20**
3. fmul f6, f1, **f2**
4. fmul **f25**, **f25**, f3
5. fmul **f1**, f9, f10
6. fadd f9, f3, **f1**

Quindi dal secondo semiperiodo, la RSS ha $busy = 0$, però l'ID ha osservato la RSS nel 1° semiperiodo, quindi la vede busy. Quindi servirà un altro ciclo di ID per trovare la RSS libera.

	B	OP	Q_j	V_j	Q_k	V_k
A0			0	2	0	1

	Q
0	A0
	R.F.

1. fadd **f20**,f2,f1
2. fdiv **f2**,f2,**f20**
3. fmul f6,f1,**f2**
4. fmul f25, f25, f3
5. fmul **f1**,f9,f10
6. fadd f9, f3, **f1**