

Soluzioni degli esercizi del Capitolo 10

Questo documento contiene le soluzioni ad un numero selezionato di esercizi del Capitolo 10 del libro “Calcolatori Elettronici - Architettura e organizzazione”, Mc-Graw Hill 2017.

Coloro che avessero sviluppato soluzioni alternative a quelle qui proposte, o soluzioni a esercizi non compresi tra quelli qui trattati, sono invitati a trasmetterle all’indirizzo sotto riportato. Serviranno a migliorare e tenere aggiornati i contenuti di questo sito.

L’autore sarà grato nei confronti di coloro che segnaleranno errori di qualunque genere, sia nella parte che segue sia nel libro menzionato.

giacomo.bucci@unifi.it

Aggiornato il 19 aprile 2017

10.1 Risposta alla prima domanda

Se la pipeline ha la forma di Figura 9.4 b) del testo, essa è una pipeline lineare con 6 stadi (IF, ID, EX1, EX2, ME, WB), tutte le istruzioni attraversano i 6 stadi. L’istruzione MUL richiede l’elaborazione di EX1 e EX2, mentre le altre passano attraverso EX2 perdendo solo tempo.

In queste ipotesi, la Tabella 10.1 mostra il contenuto dei 6 stadi della pipeline riportando l’istruzione in essi contenuta ai differenti clock. Il segno “–” sta per contenuto irrilevante o bolla.

L’istruzione 3 subisce 3 stalli per R1, in attesa del completamento istruzione 2; gli stalli sono 3 perché, per l’ipotesi di clock halving l’istruzione 3 può eseguire in ID la lettura del contenuto del registro R1 mentre l’istruzione 2 è ancora in WB. L’istruzione 5 subisce 2 stalli per R3 (completamento istruzione 3). Incidentalmente osserviamo che

clock	IF	ID	EX1	EX2	ME	WB
1	1	–	–	–	–	–
2	2	1	–	–	–	–
3	3	2	1	–	–	–
4	4	3	2	1	–	–
5	4	3	–	2	1	–
6	4	3	–	–	2	1
7	4	3	–	–	–	2
8	5	4	3	–	–	–
9	6	5	4	3	–	–
10	6	5	–	4	3	–
11	6	5	–	–	4	3
12	–	6	5	–	–	4
13	–	–	6	5	–	–
14	–	–	–	6	5	–
15	–	–	–	–	6	5
16	–	–	–	–	–	6

Tabella 10.1 (Esercizio 10.1) Risposta alla prima domanda.

se la macchina avesse avuto un memoria non divisa, con possibilità di conflitto tra la

fase IF e la fase ME, la sequenza precedente non avrebbe introdotto alcun stallo. Infatti, le istruzioni che usano effettivamente ME sono la 2, la 5 e la 6; quando la 2 è in ME, lo stadio IF è già in stallo per il motivo visto e quindi non c'è conflitto; per la 5 e la 6 occorrerebbe sapere quali sono le istruzioni che seguono.

Risposta alla seconda domanda

Se la pipeline assume la forma di Figura 9.4 c) essa non è più lineare, in quanto lo stadio di esecuzione è ora fatto di due percorsi paralleli:

- un percorso composto dal solo stadio EX1a, che richiede un solo periodo di clock, attraverso il quale passano tutte le istruzioni eccetto la MUL
- uno percorso composto da due stadi EX1b e EX2, che richiede due periodi di clock, attraversato dalla sola MUL.

In altre parole la pipeline assume la forma di Figura 10.1. Ovviamente si fa l'ipotesi che venga emessa dallo stadio ID verso la dovuta unità di esecuzione solo un'istruzione alla volta.

Si noti che in questa pipeline possono succedere fenomeni come quelli descritti al nel testo. Per esempio, se non si prendessero provvedimenti, l'istruzione 2 tenderebbe ad uscire dalla sua stazione di esecuzione (EX1a) contemporaneamente all'istruzione 1 (che esce da EX2). Dunque si avrebbe un conflitto sul percorso (unico) che dallo stadio di esecuzione porta allo stadio ME. Facciamo l'ipotesi che la macchina sia dotata del meccanismo del completamento in ordine (attraverso RSR). Ciò comporta che al ciclo 4, quando l'istruzione 1 (MUL) è in EX2, l'istruzione 2 (LD) deve essere mantenuta in ID, in quanto diversamente configgerebbe con 1 per il bus verso ME.

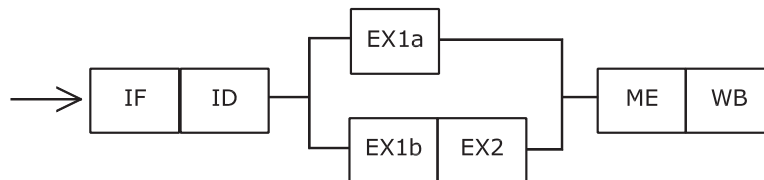


Figura 10.1 (Esercizio 10.1) Struttura della pipeline per la seconda domanda.

Il contenuto della pipeline all'avanzare del clock è quello di Tabella 10.2

L'istruzione 3 (ADD) ha una dipendenza dati dalla 2 per via di R1, dunque il fetch dei registri per l'istruzione 3 può essere fatto solo quando la 2 è in WB (cioè al clock 7), in virtù della sovrapposizione.

Sul clock 9 l'istruzione 3 è in Me, mentre la 4 è in EX1b. L'istruzione 5 (ST) può leggere i registri (sfruttando la sovrapposizione ID-WB) solo quando la 3 è in WB e cioè sul clock 10.

Si noti infine che, a causa dei conflitti, si sono risparmiati solo 2 clock rispetto al caso precedente, non 4 come ci si poteva aspettare dalla riduzione di 1 clock su 4 istruzioni.

clock	IF	ID	EX1a		ME	WB
			EX1b	EX2		
1	1	—	—	—	—	—
2	2	1	—	—	—	—
3	3	2	1	—	—	—
4	3	2	—	1	—	—
5	4	3	2	—	1	—
6	4	3	—	—	2	1
7	4	3	—	—	—	2
8	5	4	3	—	—	—
9	6	5	4	—	3	—
10	6	5	—	4	—	3
11	—	6	5	—	4	—
12	—	—	6	—	5	4
13	—	—	—	—	6	5
14	—	—	—	—	—	6

Tabella 10.2 Esercizio 10.1. Risposta alla seconda domanda. Si ricordi che c'è sovrapposizione ID-WB.

10.3 Questo esercizio prevede tre risposte.

1. Senza rete di bypass:

- L'istruzione ADD inizia sul clock n.1 e si conclude al clock 5.
- L'istruzione LD inizia sul clock n.2 e si conclude al clock 6.
- L'istruzione MUL ha una dipendenza dalla precedente: la sua fase ID può aver luogo solo dopo il completamento della precedente e cioè sul clock 7. Poiché richiede due cicli EX, essa si conclude al clock 11.
- L'istruzione ADDF non ha dipendenze; pertanto la sua fase IF inizia sul clock 7; la fase WB è sul clock 12 (richiedendo due cicli EX).
- L'istruzione DIVF non ha dipendenze; pertanto la sua fase IF inizia sul clock 8; la fase WB è sul 14 (richiedendo tre cicli EX).

Dunque il numero di clock complessivo è 14.

2. Con rete di bypass:

- L'effetto della dipendenza tra la MUL e la LD può essere alleviato dalla rete di bypass ma non completamente eliminato. Infatti alla fase EXE di MUL può essere anticipato il valore di R4 solo dopo che l'istruzione LD ha completato il suo stadio ME, cioè al clock 6. Ciò determina l'eliminazione di due stalli. Il numero complessivo di clock diventa quindi 12.

3. Col riordinamento statico:

- Non essendoci dipendenze tra le ultime 3 istruzioni, la MUL può essere spostata in fondo alla sequenza. Si può verificare che in tal caso la DIVF termina sul clock 10. La fase di ID della MUL verrebbe a coincidere con la fase WB della LD (clock 6) e ciò richiede uno stallo alla MUL, in modo da avere l'ID di quest'ultima al clock 7. Tenuto conto che la MUL passa 2 cicli in EX (clock 8 e 9), la MUL termina sul clock 11. Il numero totale di clock è dunque 11. Come mostrato in Figura 10.2.
- Se ora osserviamo che da un punto di vista logico non c'è nessuna ragione per avere la LD dietro la ADD e invertiamo ADD e LD (in aggiunta alla modifica precedente, la

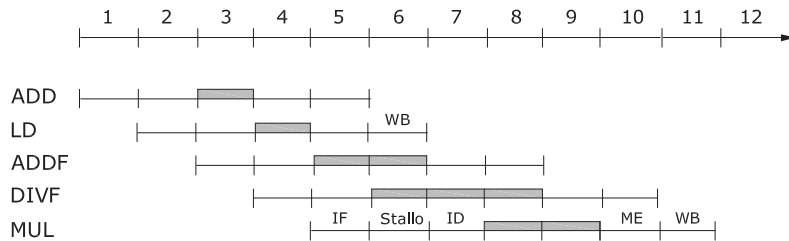


Figura 10.2 Esercizio 10.3. Profilo di esecuzione dopo il riordinamento con la MUL in ultima posizione. I tratti in grigio evidenziano le fasi di esecuzione, di diversa lunghezza a seconda dell'istruzione.

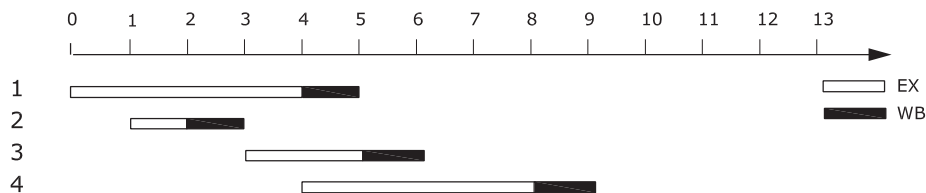
MUL non ha uno stallo per lo stadio ID, ma deve comunque aspettare la conclusione della DIVF, per cui il conto totale dei clock resta 11.

- Si a partire dalla situazione ora descritta, si porta la MUL a precedere la DIVF non ci sarebbe sovrapposizione per il WB della MUL con quello di altre istruzioni, ma la DIVF (quella che ha la fase di EX più lunga) farebbe avanzare il suo WB fino al clock 12. In effetti questa soluzione non sfrutta per niente la possibilità che istruzioni lunghe “coprano” il lavoro di istruzioni più brevi.

10.4 Qui di seguito riportiamo la sequenza data.

1	(100)	mul	f1,f12,f12	4 cicli	FPU
2	(104)	add	r2,r4,r5	1 ciclo	IU1
3	(108)	mul	r3,r10,r11	2 cicli	IU2
4	(112)	addf	f4,f6,f7	4 cicli	FPU

Si può verificare che il profilo di esecuzione è quello della figura sotto riportata. La durata è 9 periodi di clock.



10.5 Si fa riferimento al modello di Figura 10.38 del testo. Indichiamo con UI1 l'unità per interi da 1 ciclo, con UI3 l'unità per interi da 3 cicli e con UF l'unità in virgola mobile da 4 cicli.

Notare che la seconda istruzione deve attendere che la prima sia completata prima di poter essere messa in esecuzione; ciò comporta uno stallo nello stadio ID di 3 clock.

Analogamente, la quarta istruzione deve attendere il completamento della terza; La sesta istruzione deve attendere il completamento della quinta; in questo caso la bolla è di 6 cicli (4 perché l'istruzione `mul` attraversa lo stadio EX e 2 per gli stadi ME e WB).

Lasciamo al lettore la stesura dei disegni che mostrano lo stato del ROB, accontentandoci di elencare, qui di seguito, gli eventi ai differenti clock (il clock 1 è quello successivo all'emissione dell'istruzione 1, ovvero quello in cui questa istruzione è in EX)

1. L'istruzione 1 è in EX (in UI1).
2. L'istruzione 1 è in ME, la 2 in stallo in ID in attesa della conclusione della fase di WB dell'istruzione 1 e della conseguente disponibilità di r1.
3. L'istruzione 1 è in WB, la 2 in stallo in ID.
4. L'istruzione 2 completa la fase ID e viene emessa verso EX.
5. L'istruzione 2 è in EX (primo slot di UI3).
6. L'istruzione 2 è in EX (secondo slot di UI3); l'istruzione 3 è in EX (in UI1).
7. L'istruzione 2 è in EX (terzo slot di UI3); l'istruzione 3 è nel ROB; l'istruzione 4 è in stallo in ID in attesa che sia completata l'istruzione 3.
8. L'istruzione 2 è in ME; l'istruzione 3 è nel ROB; l'istruzione 4 è in stallo in ID.
9. L'istruzione 2 è in WB; l'istruzione 3 è in ME; l'istruzione 4 è in stallo in ID.
10. L'istruzione 3 è in WB; l'istruzione 4 è in stallo in ID.
11. L'istruzione 4 completa la fase ID e viene emessa verso EX.
12. L'istruzione 4 è in EX (primo slot di UI3).
13. L'istruzione 4 è in EX (secondo slot di UI3); l'istruzione 5 è in EX (primo slot di UF);
14. L'istruzione 4 è in EX (terzo slot di UI3); l'istruzione 5 è in EX (secondo slot di UF); l'istruzione 6 è in stallo in ID in attesa del risultato dell'istruzione 5.
15. L'istruzione 4 è in ME; l'istruzione 5 è in EX (terzo slot di UF); l'istruzione 6 è bloccata in ID.
16. L'istruzione 4 è in WB; l'istruzione 5 è in EX (quarto slot di UF); l'istruzione 6 è bloccata in ID.
17. L'istruzione 5 è in EX (quarto slot di UF); l'istruzione 6 è bloccata in ID.
18. L'istruzione 5 è in ME; l'istruzione 6 è bloccata in ID.
19. L'istruzione 5 è in WB; l'istruzione 6 è ancora bloccata in ID.
20. L'istruzione 6 completa la sua fase ID e viene emessa.
21. L'istruzione 6 è in EX (primo slot di UF).
22. L'istruzione 6 è in EX (secondo slot di UF); l'istruzione 7 è in EX (in UI1).
23. L'istruzione 6 è in EX (terzo slot di UF); l'istruzione 7 è nel ROB.
24. L'istruzione 6 è in EX (quarto slot di UF); l'istruzione 7 è nel ROB.
25. L'istruzione 6 è in ME; l'istruzione 7 è nel ROB.
26. L'istruzione 6 è in WB; l'istruzione 7 è in ME.

Come si vede ci corrono 26 periodi di clock tra quello in cui la prima istruzione entra in esecuzione e quello in cui l'ultima esce dal ROB.

Le prime quattro istruzioni non possono essere riordinate tra loro. Eventualmente la loro sequenza può essere inframezzata con altre istruzioni, purché il loro ordine resti immutato. L'istruzione 5 deve comunque precedere la 6. L'istruzione 7, non dipendendo da alcuna delle precedenti può essere spostata ovunque.

Dunque si può riordinare nel modo seguente (nella colonna di sinistra c'è il nuovo ordine, in quella accanto il numero originale):

1 ¹	1	add	r1,r2,r3	1 ciclo
2 ¹	5	mulf	f12,f8,f10	4 cicli
3 ¹	7	add	r20,r21,r22	1 ciclo
4 ¹	2	div	r5,r4,r1	3 ciclo
5 ¹	3	add	r1,r6,r7	1 ciclo
6 ¹	4	mul	r8,r9,r1	3 cicli
7 ¹	6	addf	f18,f12,f16	4 cicli

In questo modo, si avrebbe un unico stallo di 3 clock sull'istruzione 6¹ (mul) per via di r1 atteso dalla 5¹ (add). Questo bloccherebbe anche l'istruzione 7¹ che ha un lungo tempo di esecuzione. Convienne portare la addf a precedere la mul, in modo da poterla avviare prima possibile (compatibilmente col fatto che la mulf sia completata). Convienne quindi questo ordine

1 ²	1 ¹	1	add	r1,r2,r3	1 ciclo
2 ²	2 ¹	5	mulf	f12,f8,f10	4 cicli
3 ²	3 ¹	7	add	r20,r21,r22	1 ciclo
4 ²	4 ¹	2	div	r5,r4,r1	3 ciclo
5 ²	5 ¹	3	add	r1,r6,r7	1 ciclo
6 ²	7 ¹	6	addf	f18,f12,f16	4 cicli
7 ²	6 ¹	4	mul	r8,r9,r1	3 cicli

La precedente sequenza di 26 passi si riduce a questa:

1. L'istruzione 1² è in EX (in UI1).
2. L'istruzione 1² è in ME; la 2² in EX (primo slot di UF).
3. L'istruzione 1² è in WB; la 2² in EX (secondo slot di UF); la 3² in EX (in UI1).
4. L'istruzione 2² è in EX (terzo slot di UF); la 3² è nel ROB; la 4² in EX (primo slot di UI3);
5. L'istruzione 2² è in EX (quarto slot di UI3). la 3² è nel ROB; la 4² è in EX (secondo slot di UI3); la 5² è in EX (in UI1)
6. L'istruzione 2² è in ME; la 3² è nel ROB; la 4² è in EX (terzo slot di UI3); la 5² è nel ROB; la 6² in stallo in ID in attesa del completamento della 2².
7. L'istruzione 2² è in WB; la 3² in ME; la 4² è nel ROB; la 5² è nel ROB; la 6² in stallo in ID.
8. L'istruzione 3² è in WB; la 4² in ME; la 5² è nel ROB; la 6 in EX (primo slot di UF).
9. L'istruzione 4² è in WB; la 5² è in ME; la 6² è in EX (secondo slot di UF); la 7² in stallo in ID per il completamento della 5.
10. L'istruzione 5² è in WB; la 6² è in EX (terzo slot di UF); la 7² in stallo in ID.
11. L'istruzione 6² è in EX (quarto slot di UF); la 7² è in EX (primo slot di UI3).
12. L'istruzione 6² è in ME; la 7² è in EX (secondo slot di UI3).
13. L'istruzione 6² è in WB; la 7² è in EX (terzo slot di UI3).
14. L'istruzione 7² è in ME.

Il numero di clock è sceso a 14.

Con la sovrapposizione ID-WB si risparmiano ancora due cicli nella sequenza precedente, arrivando a 12. Se ora si suppone di avere anche il bypass, si risparmiano i due ultimi stalli rimanenti e si arriva a 10 clock.

10.6 La Tabella 10.3 illustra lo svolgimento dell'esecuzione, riportando lo stato delle unità e del ROB.

Clock	Unit:	Istruzioni	Istruzioni nel ROB
0	U1:	–	
	U2:	–, –	
	U3:	–, –, –, –	
1	U1:	3	3,2,1
	U2:	1, –	
	U3:	2, –, –, –	
2	U1:	5	5,4,3,2,1
	U2:	4, 1	
	U3:	–, 2, –, –	
3	U1:	6	7, 6, 5,4,3,2,1
	U2:	7, 4	
	U3:	–, –, 2, –	
4	U1:	10	10, 9, 8, 7, 6, 5,4,3,2
	U2:	8, 7	
	U3:	9, –, –, 2	
5	U1:	–	10, 9, 8, 7, 6, 5,4,3,2
	U2:	–, 8	
	U3:	–, 9, –, –	
6	U1:	–	10, 9, 8, 7, 6, 5
	U2:	–, –	
	U3:	–, –, 9, –	
7	U1:	–	10, 9, 8
	U2:	–, –	
	U3:	–, –, –, 9	
8	U1:	–	10, 9
	U2:	–, –	
	U3:	–, –, –, –,	
8	U1:	–	
	U2:	–, –	
	U3:	–, –, –, –,	

Tabella 10.3 Esercizio 10.6. Evoluzione del contenuto delle unità operative e del ROB. Per ipotesi la macchina è in grado di emettere verso le unità operative fino a 3 istruzioni alla volta e di ritirarne fino a 3 dal ROB.

10.7 In Figura 10.4 è illustrato il contenuto delle differenti unità funzionali e del ROB all'avanzamento del clock. Per meglio seguire gli eventi, nel ROB si riportano le istruzioni e non i nomi dei registri (e/o i valori in essi contenuti).

Notiamo subito che la dipendenza della istruzione 2 (**add**) dalla precedente consente di emettere al primo clock la sola istruzione 1. L'emissione della 2 potrà avvenire solo al clock 4 dopo che la 1 ha completato (e scritto il risultato in ROB). (Si tenga conto che c'è il bypass che consente di inviare direttamente a UI2 quello che esce da ULS.). Al clock 4 assieme alla 2 vengono emesse in parallelo la 3 e la 4 verso le rispettive unità.

Al clock 5 viene emessa solo l'istruzione 5, perché la successiva dipende da essa. Notare che al clock 5 (il risultato de) l'istruzione completata in precedenza è stato ritirato dal ROB.

Sul clock 6 e sul clock 7 non viene emessa alcuna istruzione perché la 6 attende il completamento della 5. Sul clock 6 ha termine l'istruzione 4, mentre la 2 è già stata ritirata dal ROB. Sul clock 7 terminano la 3 e la 5, per cui sul clock 8 possono essere emesse la 6 e la 7. Sul medesimo clock vengono ritirate le tre istruzioni completate in precedenza, cioè 3, 4 e 5.

Sul clock 9 viene emessa l'istruzione 8 che si conclude sullo stesso clock assieme alla 7. Queste due istruzioni vengono estratte dal ROB al clock successivo.