

# CONFLITTI DI DATI

sub \$2, \$1, \$3 # il registro \$2 è scritto

and \$12, \$2, \$5 # il primo operando (\$2) dipende dalla sub

or \$13, \$6, \$2 # il secondo operando (\$2) dipende dalla sub

add \$14, \$2, \$2 # entrambi gli operandi dipendono dalla sub

sw \$15, 100(\$2) # l'indice (\$2) dipende dalla sub

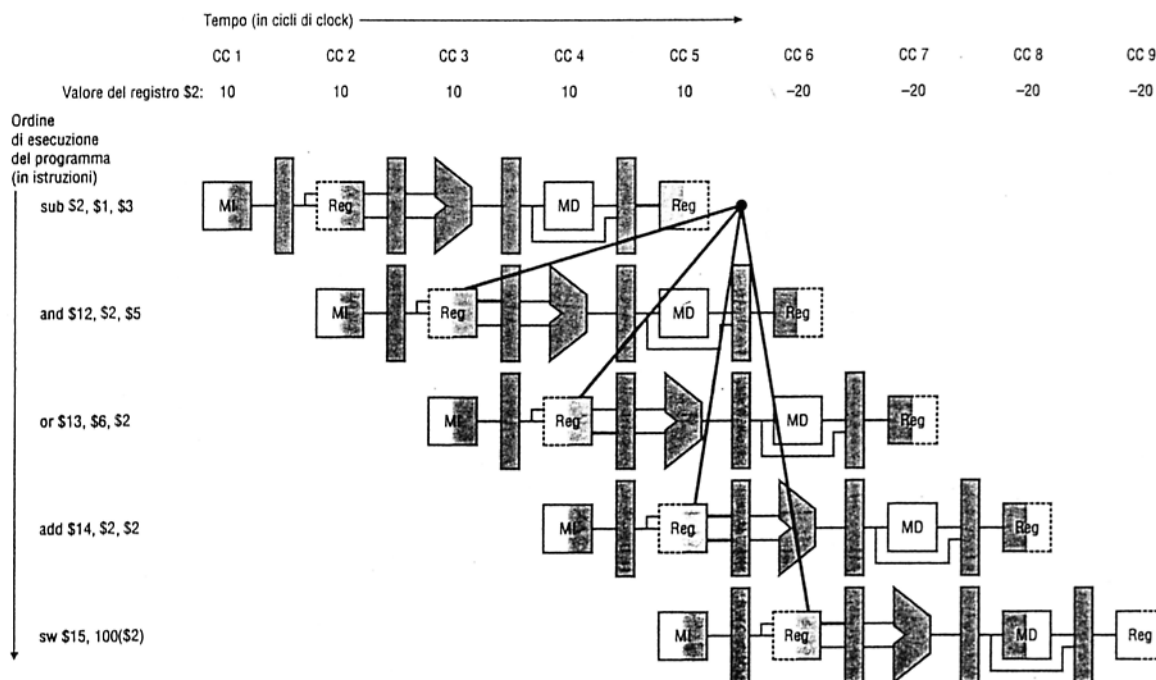


FIGURA 6.30 Dipendenze nell'organizzazione a pipeline nella sequenza di cinque istruzioni; si usa un'unità di calcolo semplificata per mettere in evidenza le dipendenze. Tutte le azioni dipendenti sono evidenziate con il colore e «CC/» nella parte superiore della figura indica il ciclo di clock  $i$ . La prima istruzione scrive nel registro \$2, le istruzioni successive leggono \$2. Il registro viene scritto nel ciclo di clock 5, quindi il valore corretto non è disponibile prima del ciclo di clock 6. Le linee colorate dall'unità di calcolo in alto verso quelle in basso mostrano le dipendenze. Quelle che devono andare verso cicli di clock precedenti si chiamano *conflitti di dati della pipeline*. Si noti che i registri hanno la metà di lettura e la metà di scrittura.

# SOLUZIONE SOFTWARE PER EVITARE CONFLITTI DI DATI

sub \$2, \$1, \$3

nop

nop

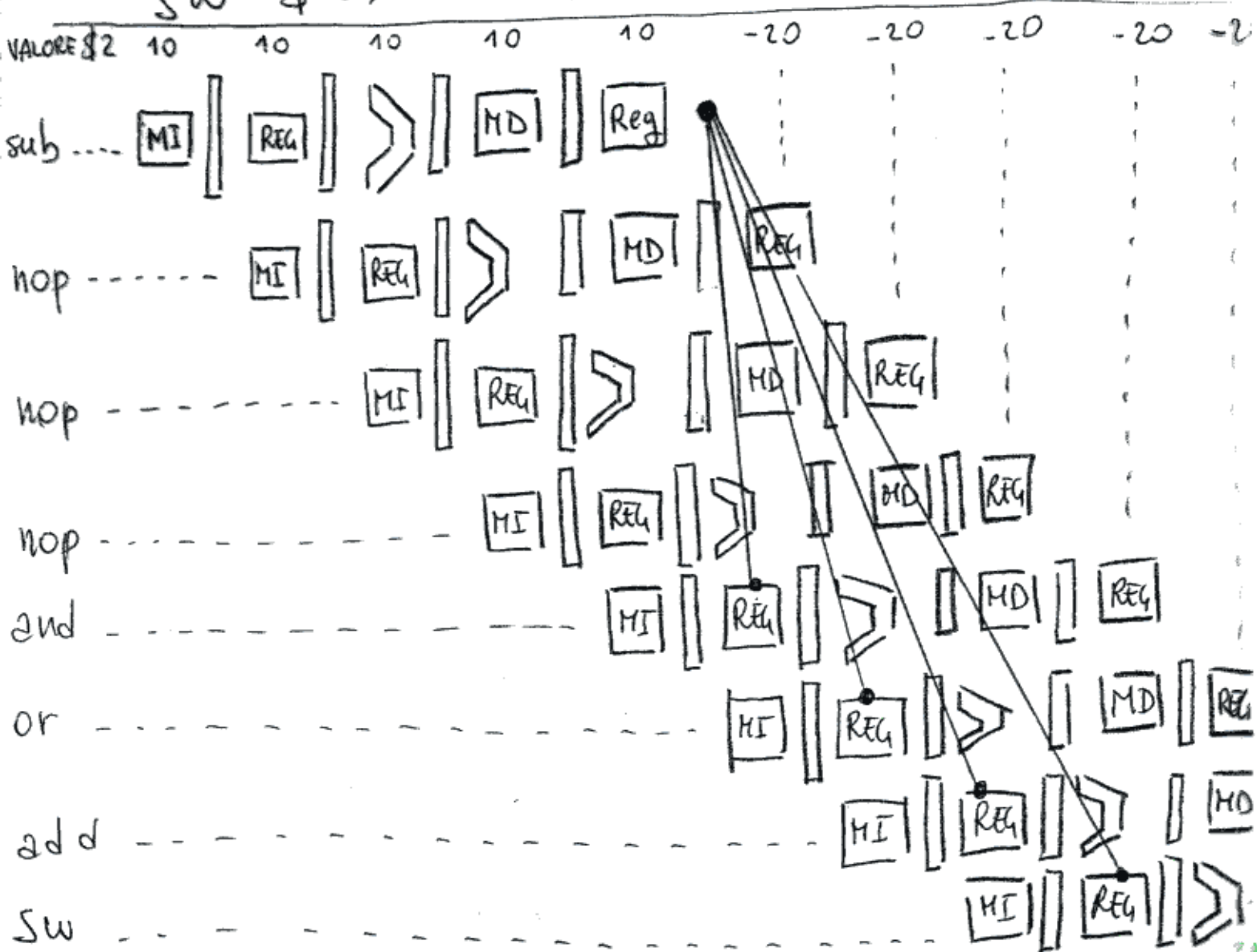
nop

and \$12, \$2, \$5

or \$13, \$6, \$2

add \$14, \$2, \$2

sw \$15, 100(\$2)



# SOLUZIONE HARDWARE PER EVITARE CONFLITTI DI DATI : INSERIMENTO DI "BOLLE"

- ▶ LE BOLLE METTONO IN STALLO LO STADIO DELLA PIPE LINE IN CUI SONO PRESENTI
- ▶ PER REALIZZARE UNA BOLLA E' SUFFICIENTE PORRE A ZERO TUTTI I CAMPI DI CONTROLLO DEL REGISTRO DI PIPELINE ID/EX, CHE POI SI PROPAGA NEI REGISTRI EX/MEM & MEM/WB

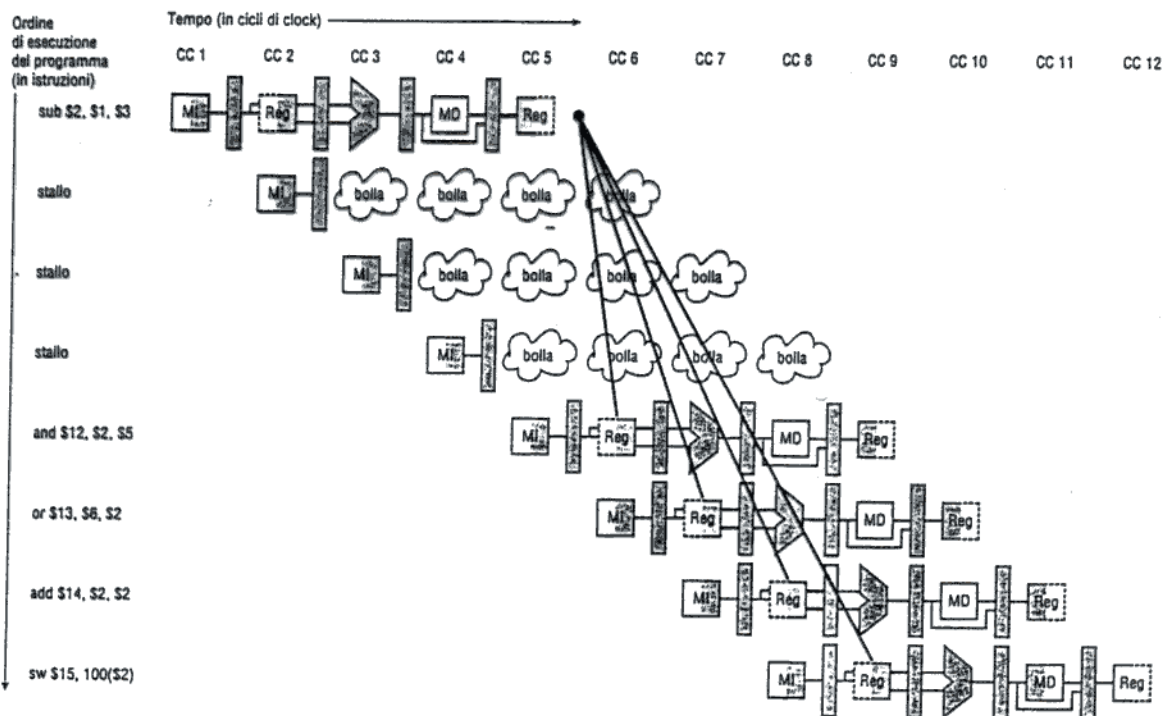


FIGURA 6.33 Una versione della figura 6.32 che mostra il modo in cui gli stalli vengono realmente inseriti nella pipeline. Poiché le dipendenze sono verso stadi successivi, non ci sono conflitti. Si noti che i tre cicli di clock di stallo, mostrati con bolle nella figura 6.30, sono simili a istruzioni `nop` inserite dal compilatore nell'esempio della rubrica «Interfaccia hardware-software» a pagina 330.

# RIVELAZIONI DI CONFLITTI A RUN-TIME

condizioni:

1a. ID/EX. Reg SCRITTURA = IF/ID. Reg LETTURA 1

1b ID/EX. Reg SCRITTURA = IF/ID. Reg LETTURA 2

2a. EX/MEM. Reg SCRITTURA = IF/ID. Reg LETTURA 1

2b. EX/MEM. Reg SCRITTURA = IF/ID. Reg LETTURA 2

3a MEM/WB. Reg SCRITTURA = IF/ID. Reg LETTURA 1

3b MEM/WB. Reg SCRITTURA = IF/ID. Reg LETTURA 2

Reg. SCRITTURA = { "campo"  $R_d$  se istr. tipo R  
 "campo"  $R_d$  se istr. tipo lw

RegDst = 1      RegDst = 0

### 1. Conflitto EX:

ID/EX.RegWrite and  
 ((ID/EX.RegDst = 0 and ID/EX.RegistroScritturaRt = IF/ID.RegistroLettura1) or  
 (ID/EX.RegDst = 1 and ID/EX.RegistroScritturaRd = IF/ID.RegistroLettura1) or  
 (ID/EX.RegDst = 0 and ID/EX.RegistroScritturaRt = IF/ID.RegistroLettura2) or  
 (ID/EX.RegDst = 1 and ID/EX.RegistroScritturaRd = IF/ID.RegistroLettura2))

— SEGNALI DI CONTROLLO

### 2. Conflitto MEM:

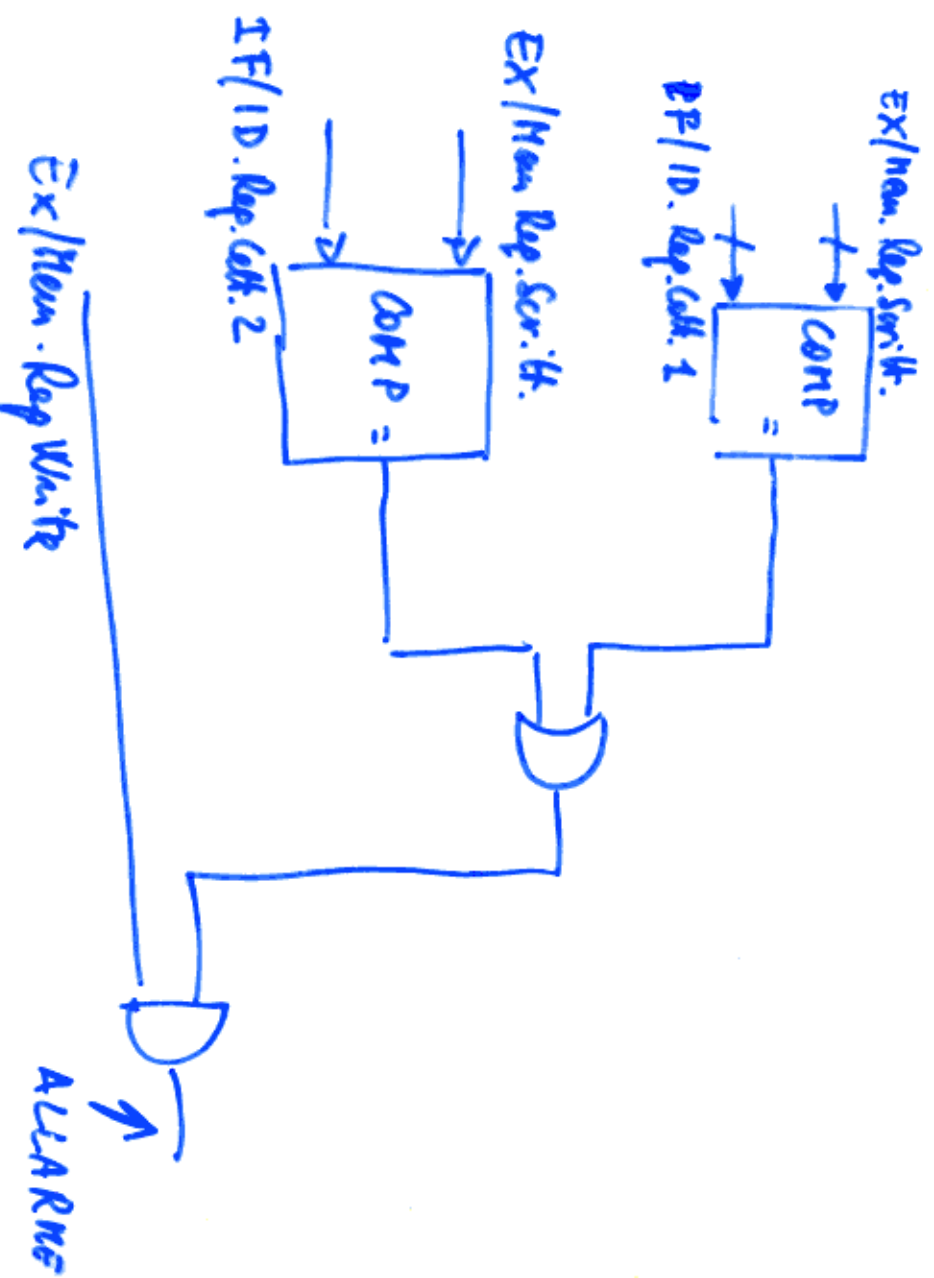
EX/MEM.RegWrite and  
 ((EX/MEM.RegistroScrittura = IF/ID.RegistroLettura1) or  
 (EX/MEM.RegistroScrittura = IF/ID.RegistroLettura2))

### 3. Conflitto WB:

MEM/WB.RegWrite and  
 ((MEM/WB.RegistroScrittura = IF/ID.RegistroLettura1) or  
 (MEM/WB.RegistroScrittura = IF/ID.RegistroLettura2))

ID/EX. Reg Write  
 EX/ID. Reg Write  
 MEM/WB. Reg Write  
 ID

EX/MEM. Reg Write  
 EX/MEM. Reg. Scr. th.



ID/EX

# Architettura semplificata di SCO - SCA con Unità di Rivelazione di Conflitti

IF/ID. Registro Lettura 1  
IF/ID Registro Lettura 2

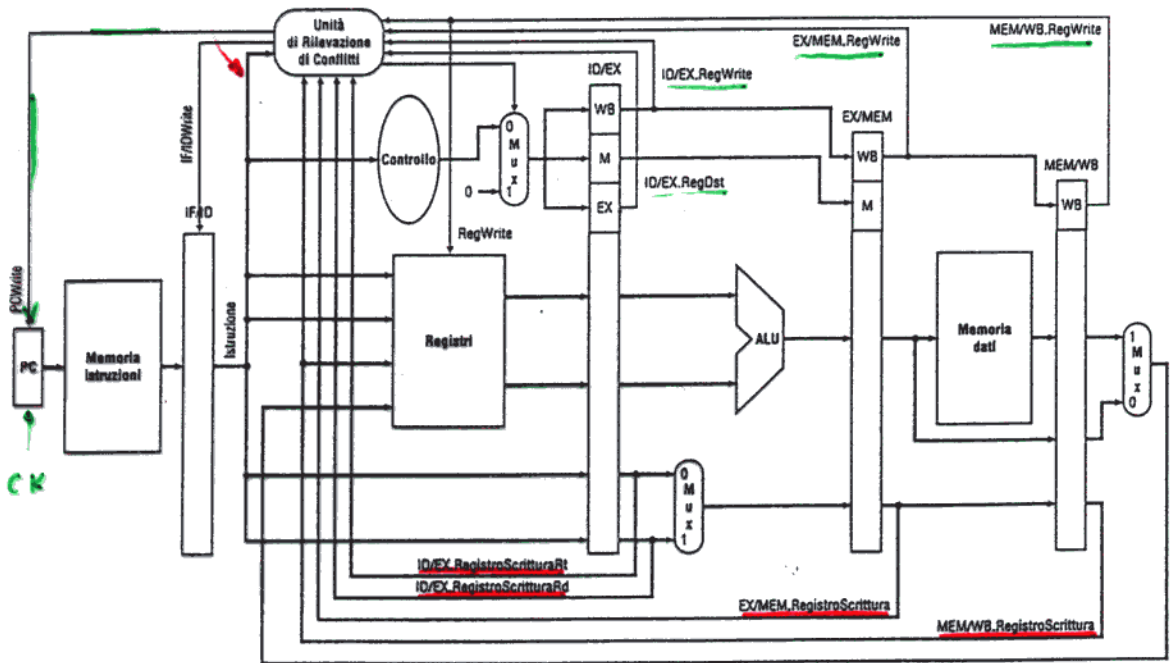


FIGURA 6.34 L'Unità di Rivelazione di Conflitti impedisce che il PC e il registro di pipeline IF/ID vengano scritti e pone a 0 i valori di controllo nel registro di pipeline ID/EX se rileva un conflitto. Utilizza per determinare i conflitti il numero del registro di destinazione, i segnali di scrittura dei registri dai tre ultimi registri di pipeline, l'istruzione corrente nello stadio ID, e inoltre il bit del registro di pipeline ID/EX che seleziona il registro di destinazione.

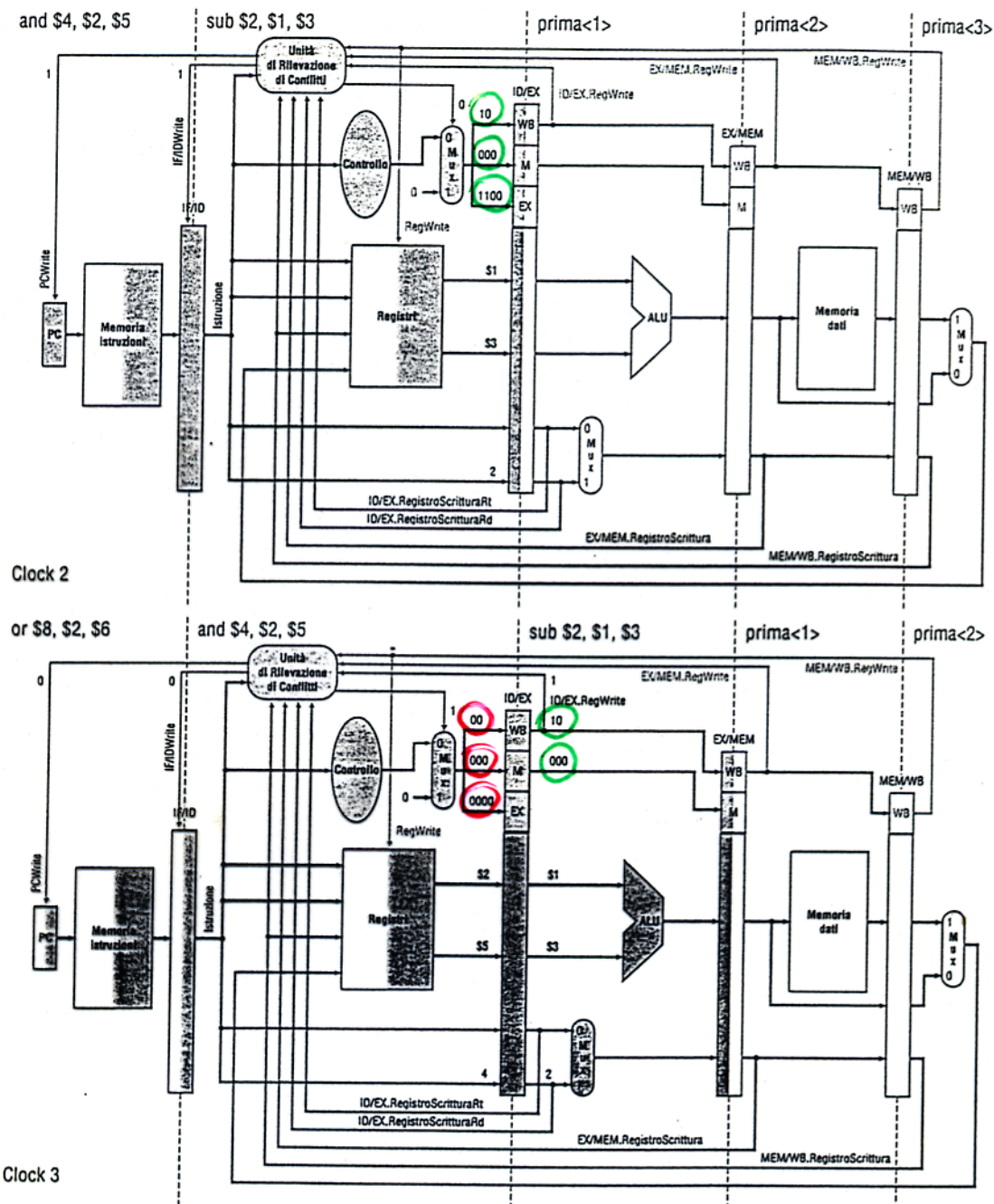


FIGURA 6.35 Cicli di clock 2 e 3 della sequenza di istruzioni dell'esempio. I valori delle linee di controllo significative, i registri, e i numeri dei registri sono tutti etichettati nelle figure. L'istruzione `and` intende leggere il valore generato dall'istruzione `sub`, quindi l'Unità di Rilevazione di Conflitti pone in stato di stallo le istruzioni `and` e `or` nel ciclo di clock 3 fino a quando il conflitto viene risolto nel ciclo di clock 6 (si veda la figura 6.37).

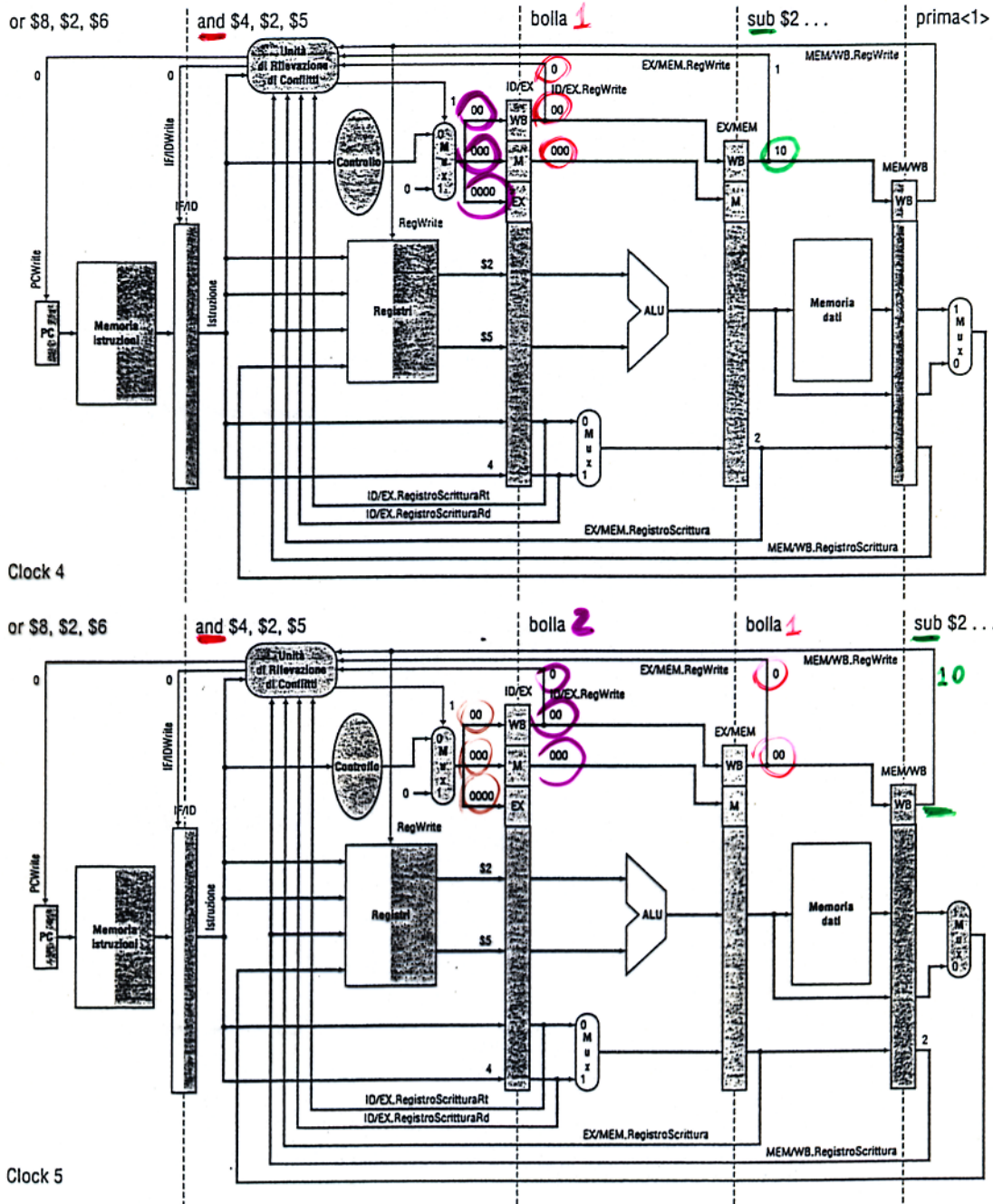


FIGURA 6.36 Cicli di clock 4 e 5 della sequenza di istruzioni dell'esempio. Lo stallò continua in questi due cicli di clock come risultato del conflitto. Si noti che sebbene il valore corretto del registro \$2 venga scritto alla fine del ciclo di clock 5, il valore letto durante il ciclo di clock e caricato nel registro di pipeline ID/EX è quello vecchio. La pipeline deve quindi essere posta in stato di stallò per un altro ciclo di clock per consentire al valore corretto di essere caricato nel registro di pipeline.

or \$8, \$2, \$6

and \$4, \$2, \$5

bolla 3

bolla 2

bolla 1

Clock 6

add \$9, \$4, \$2

or \$8, \$2, \$6

and \$4, \$2, \$5

bolla 3

bolla 2

Clock 7

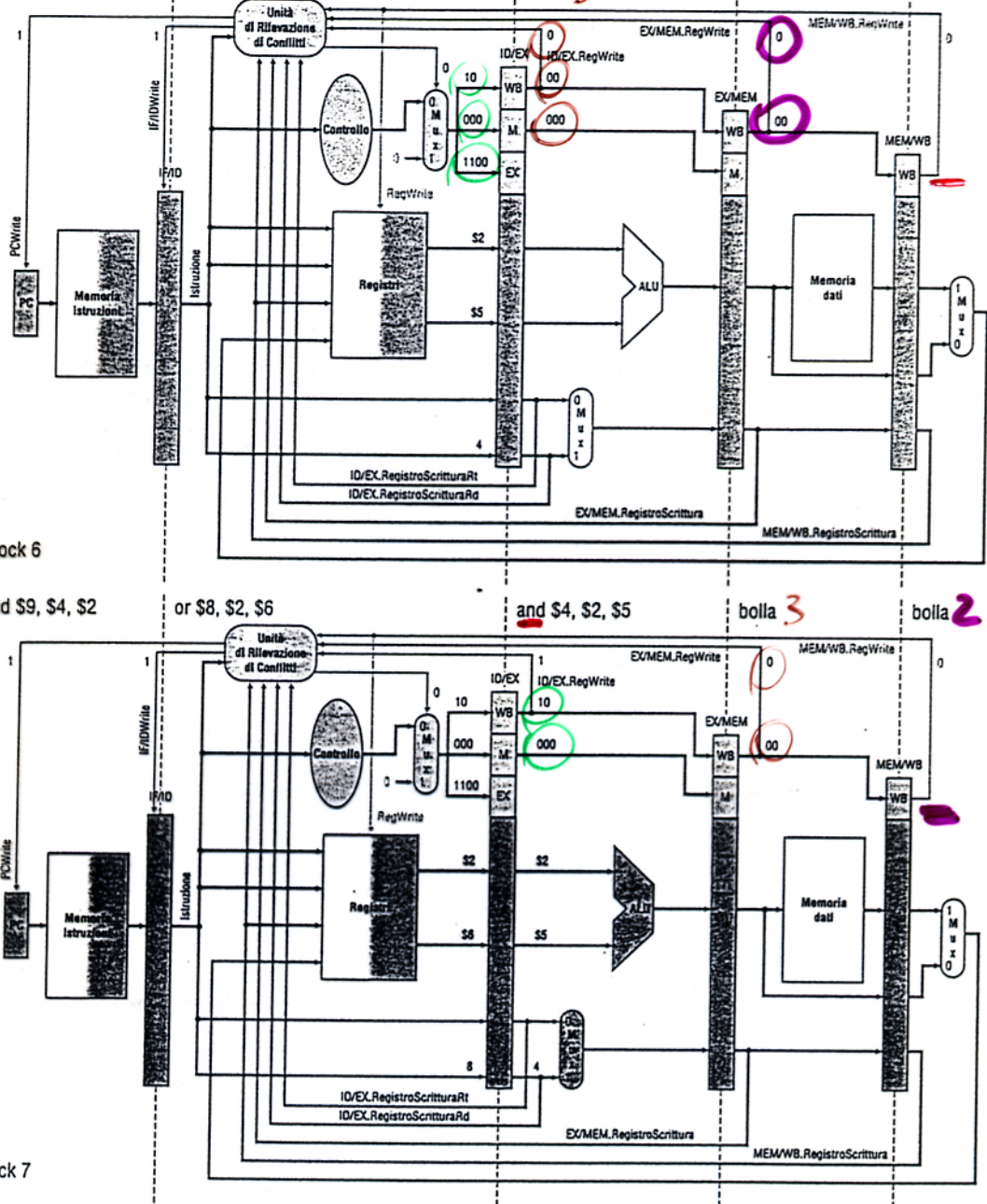


FIGURA 6.37 Cicli di clock 6 e 7 della sequenza di istruzioni dell'esempio. L'istruzione **and** può procedere nel ciclo di clock 6, mentre le restanti istruzioni proseguono fino a che l'Unità di Rilevazione di Conflitti individua un nuovo conflitto. Occorre notare che non ci sono più istruzioni nell'unità di calcolo che scrivono nel registro \$2, quindi l'istruzione **or** può procedere.

conflitto su \$4

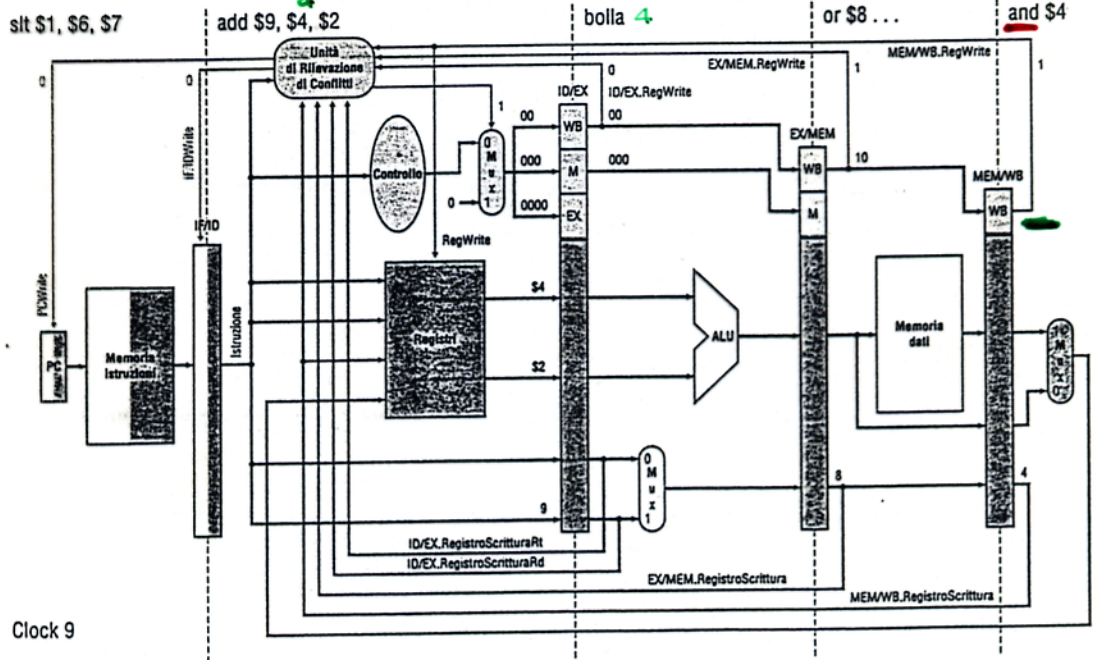
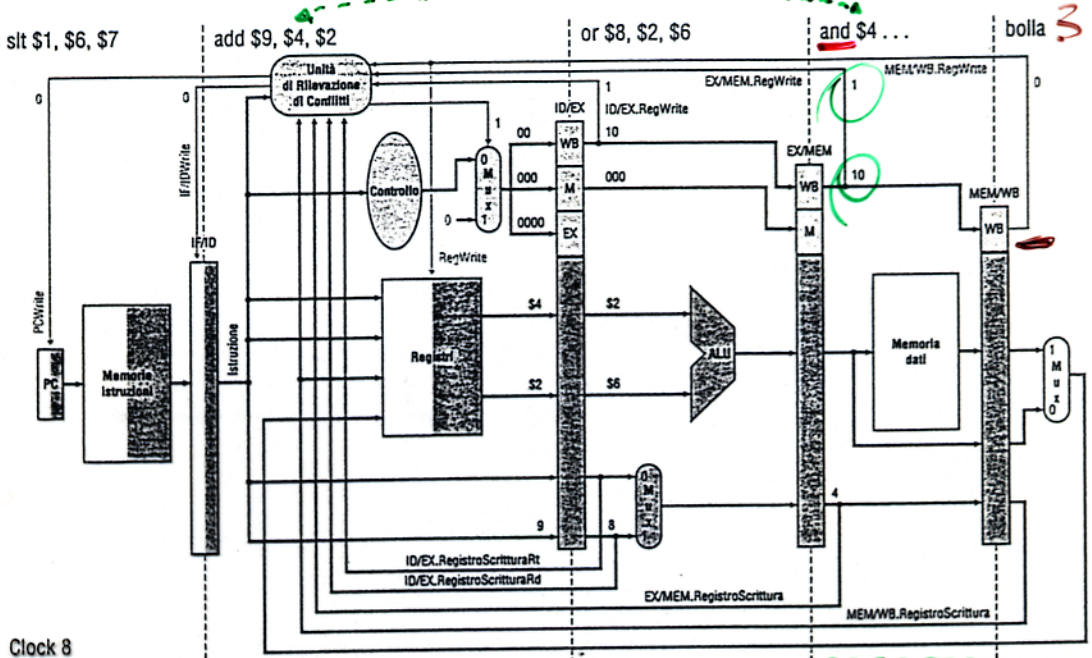


FIGURA 6.38 Cicli di clock 8 e 9 della sequenza di istruzioni dell'esempio. Nel ciclo di clock 8, il registro \$4 è motivo di conflitto tra l'istruzione `and` nello stadio MEM e l'istruzione `add` nello stadio EX, quindi l'Unità di Rilevazione di Conflitti pone in stato di stallo le istruzioni `add` e `sllt` nel ciclo di clock 9.



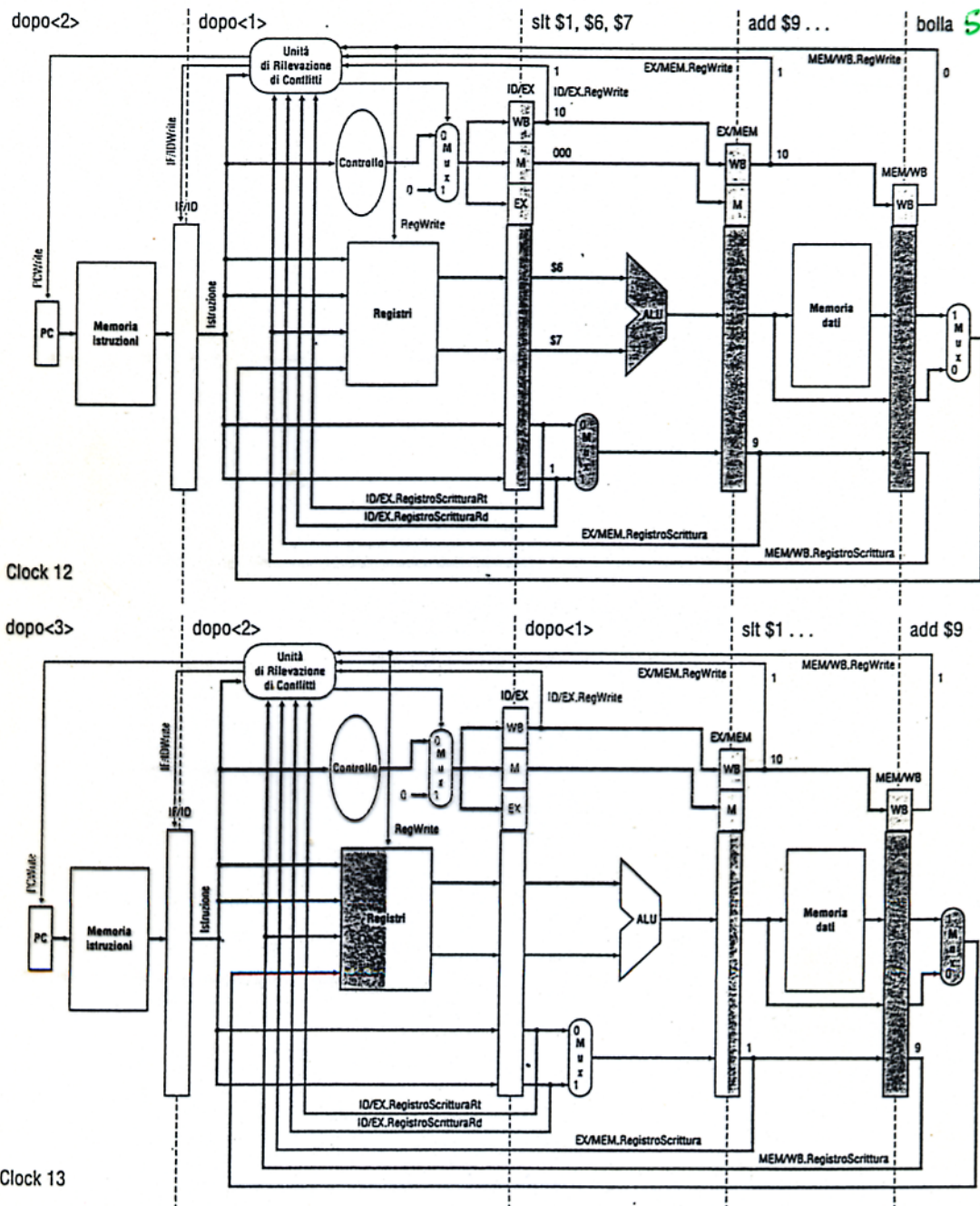


FIGURA 6.40 Cicli di clock 12 e 13 della sequenza di istruzioni dell'esempio. L'istruzione `add` posta in stallo nella figura 6.39 giunge al termine nel ciclo di clock 13.