

Tecnica del salto ritardato: Esercizi

Tecnica del salto ritardato

Istruzione di salto

prima istruzione successiva
seconda istruzione successiva
...
n-esima istruzione successiva

Destinazione del salto

**SLOT DI
RITARDO**

*Istruzioni eseguite
indipendentemente dal
successo del salto*

... di solito lo slot di ritardo può contenere 1 sola istruzione

Comportamento del salto ritardato

Caso 1

Istruzione i di salto eseguito
Istruzione $i+1$ nello slot di ritardo
Destinazione del salto
Destinazione del salto + 1
Destinazione del salto + 2

Caso 2

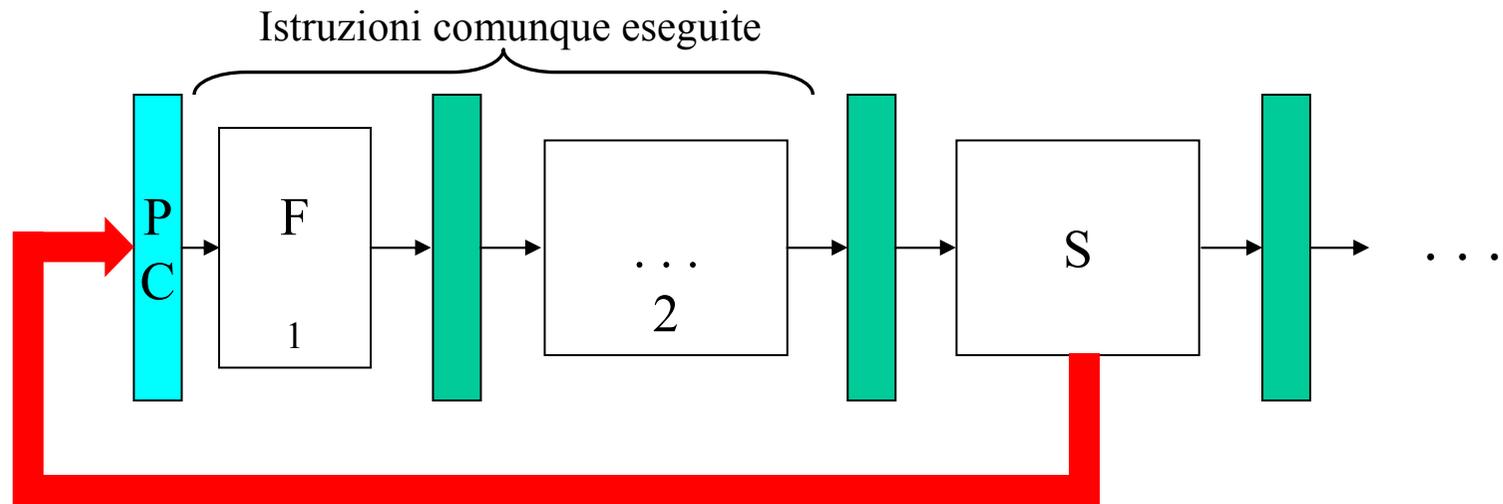
Istruzione i di salto non eseguito
Istruzione $i+1$ nello slot di ritardo
Istruzione $i+2$
Istruzione $i+3$
Istruzione $i+4$

NB: si è ipotizzato un solo slot di ritardo

Stadio di esecuzione e slot

- Il numero di slot di ritardo è legato direttamente alla posizione dello stadio in cui il salto viene eseguito (ovvero: PC aggiornato che determina il fetch istruzione destinazione nel ciclo successivo)
- In particolare:
 - stadio D (nr. 2) \Rightarrow 1 slot di ritardo
 - stadio E (nr. 3) \Rightarrow 2 slot di ritardo
 - ecc. ecc.

Infatti

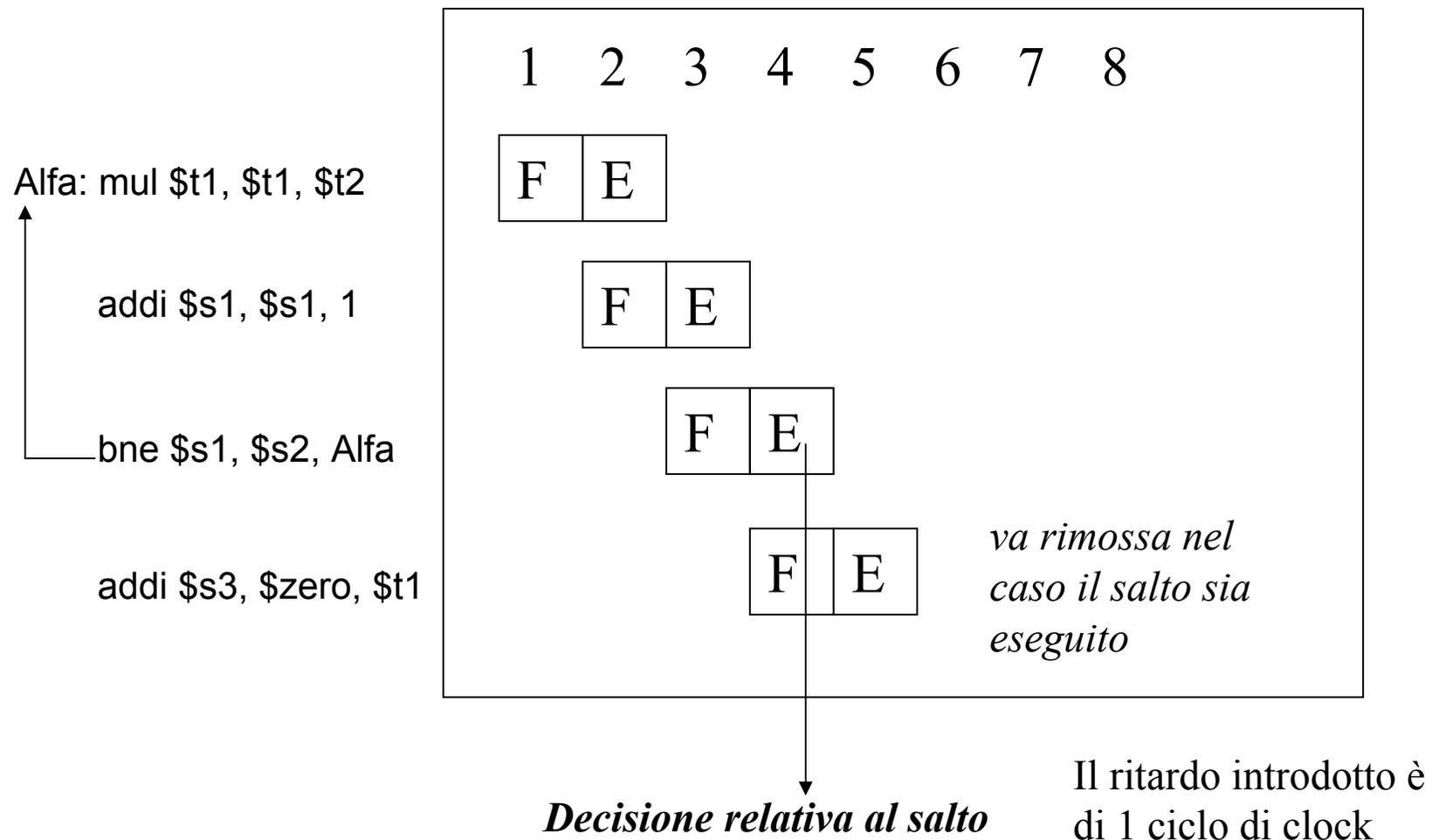


Esempio (1)

- Segmento di codice in assembler MIPS
[realizza $\$s3=2^{\$s2}$]

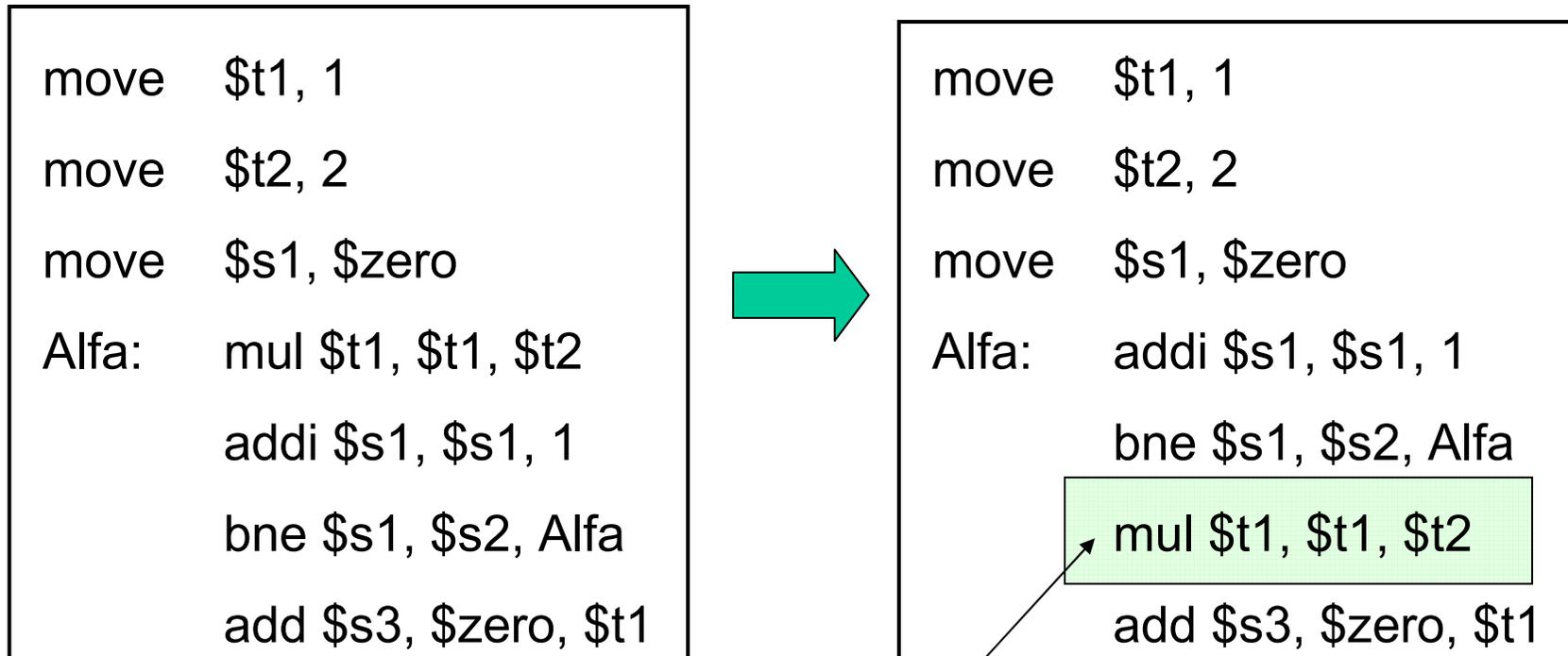
```
        move        $t1, 1           // prodotto parz.
        move        $t2, 2           // base=2
        move        $s1, $zero       // indice ciclo
Alfa:   mul         $t1, $t1, $t2
        addi        $s1, $s1, 1
        bne         $s1, $s2, Alfa
        add         $s3, $zero, $t1
```

Diagramma temporale dell'esecuzione senza salto ritardato (pipeline a 2 stadi) -prediz. “salto non eseguito”



Riordino delle istruzioni

- L'architettura prevede il salto ritardato



Istruzione inserita nello slot di ritardo

Diagramma temporale dell'esecuzione con salto ritardato: salto eseguito

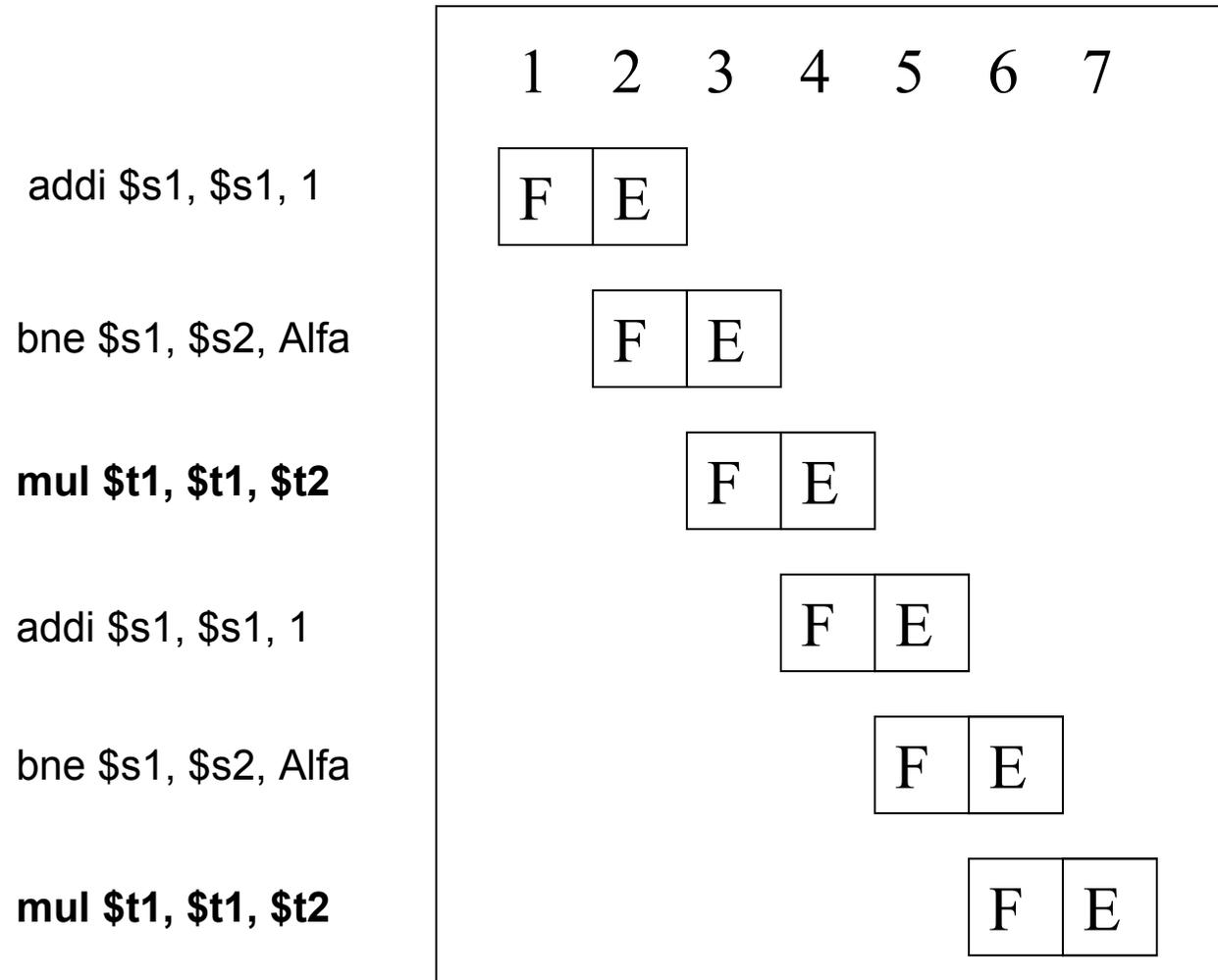
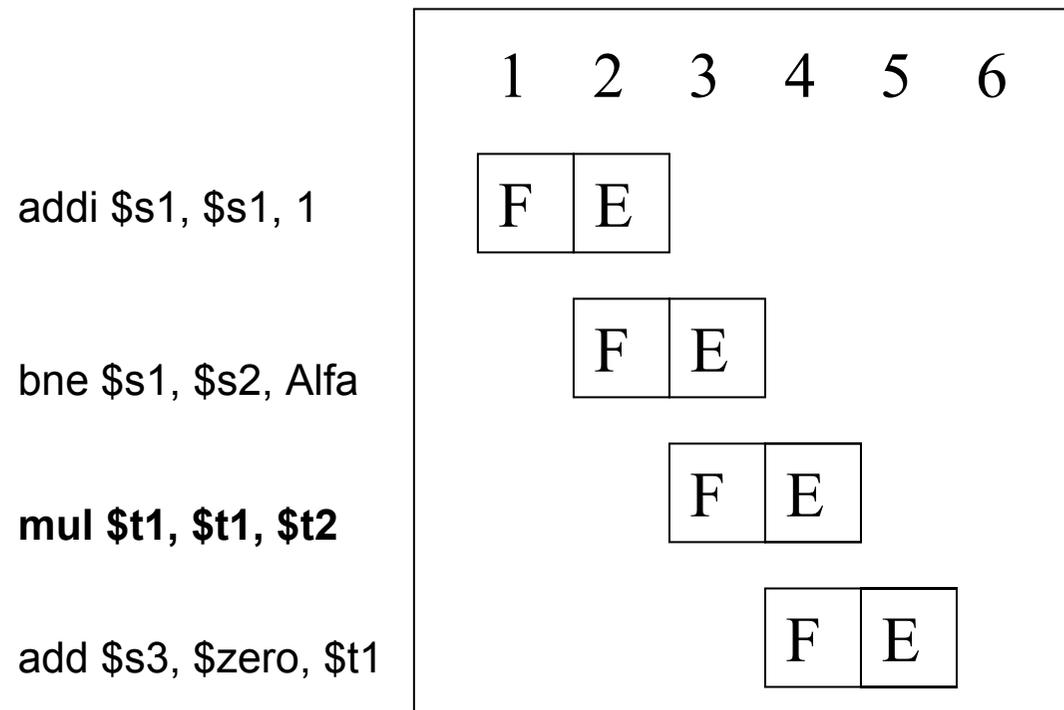


Diagramma temporale dell'esecuzione con salto ritardato: salto non eseguito



ESERCIZIO:

Supponiamo ora di rifare lo stesso esempio
(diagramma temporale con o senza salto ritardato)
ipotizzando:

- pipeline a 5 stadi (F-D-E-M-W)
- 1 slot di ritardo

NB: dalle specifiche si capisce che l'esecuzione del salto avviene necessariamente nello stadio D [1 slot di ritardo]

Diagramma temporale dell'esecuzione senza salto ritardato (pipeline a 5 stadi)

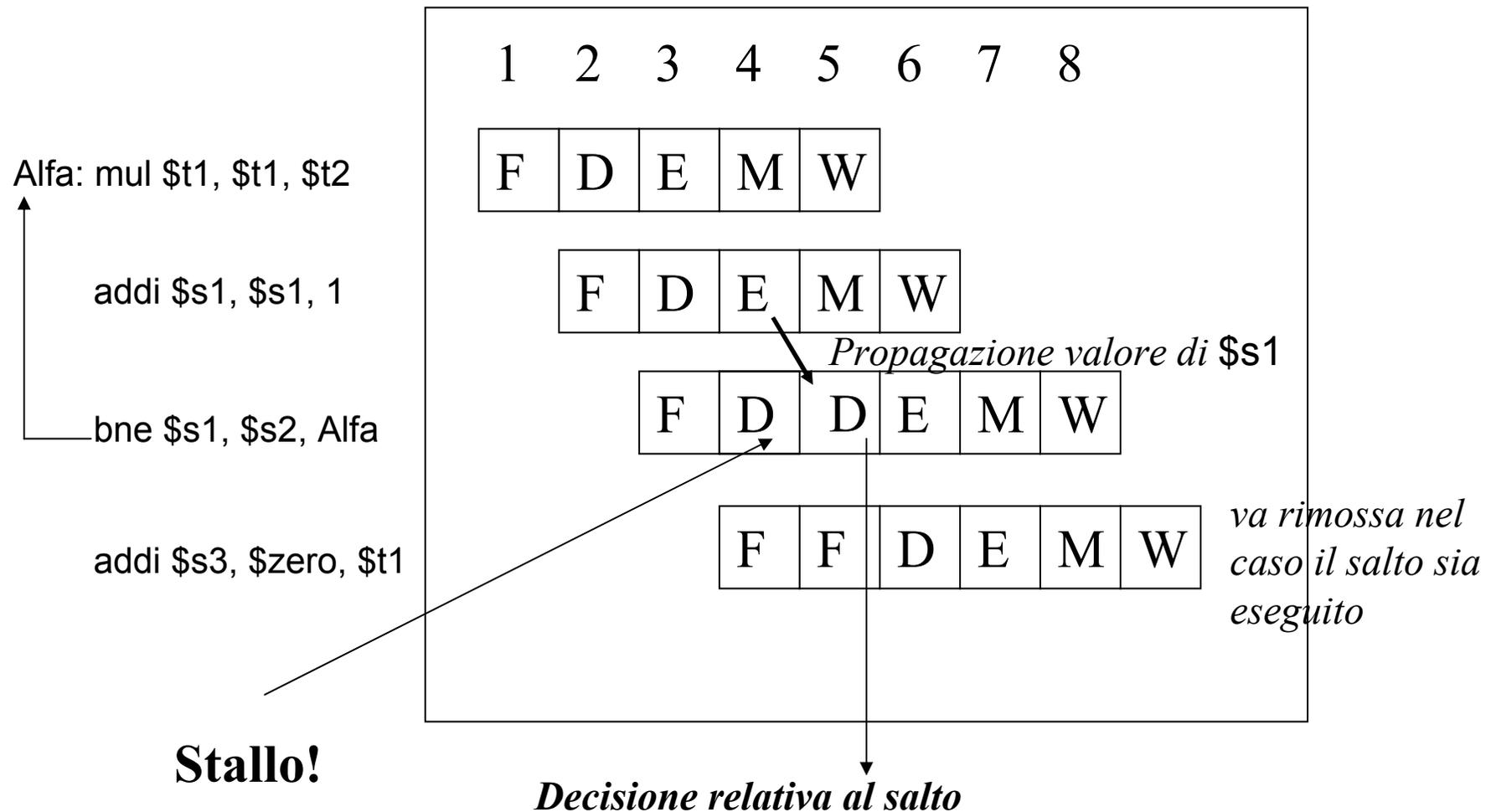


Diagramma temporale dell'esecuzione con salto ritardato: caso 1 (salto eseguito)

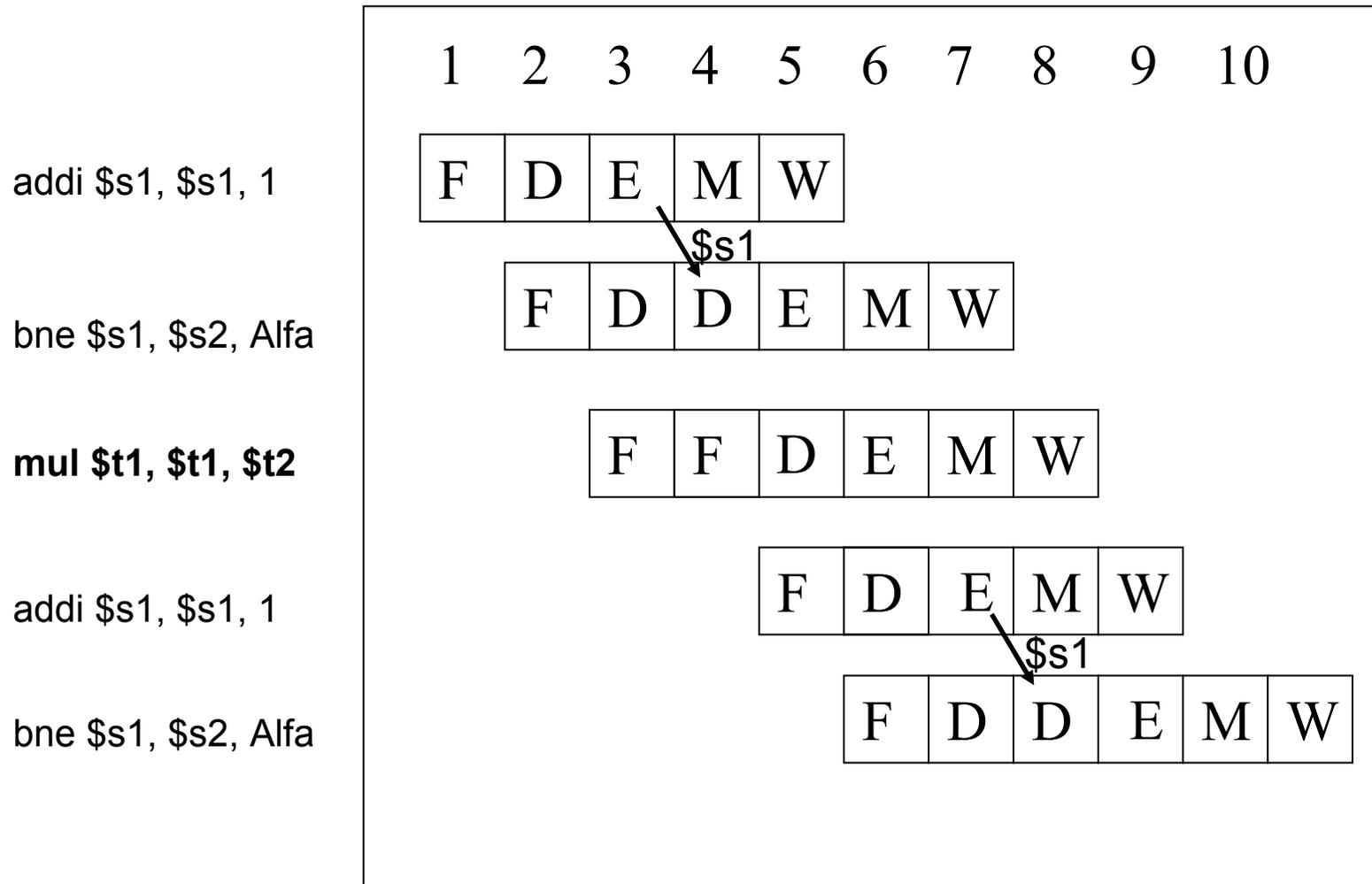
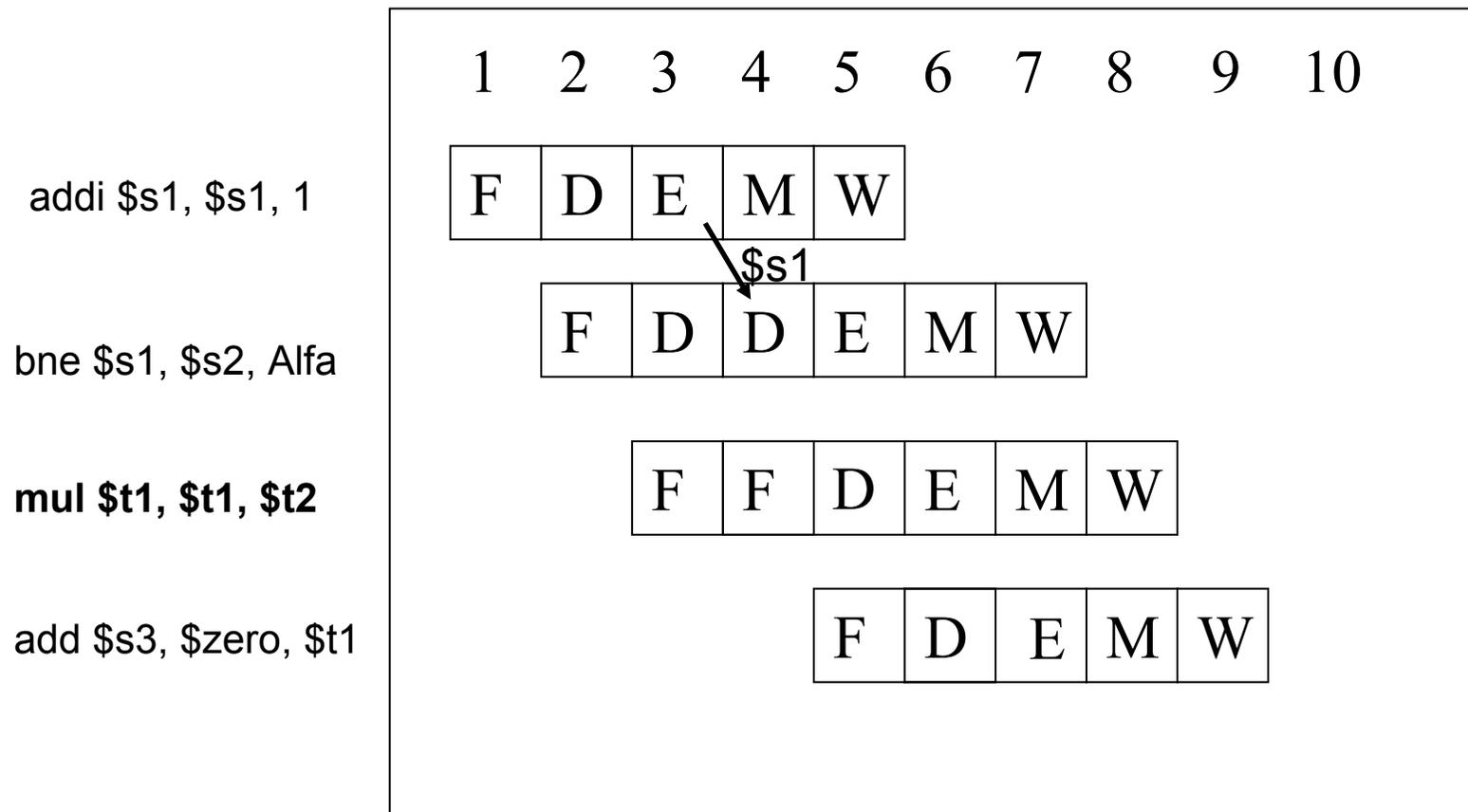


Diagramma temporale dell'esecuzione con salto ritardato: caso 2 (salto non eseguito)



ESERCIZIO:

Supponiamo che l'esecuzione del salto avvenga nello stadio M (come nella prima implementazione del MIPS) anziché nello stadio D...

Diagramma temporale dell'esecuzione senza salto ritardato (pipeline a 5 stadi)

Decisione relativa al salto nello stadio M

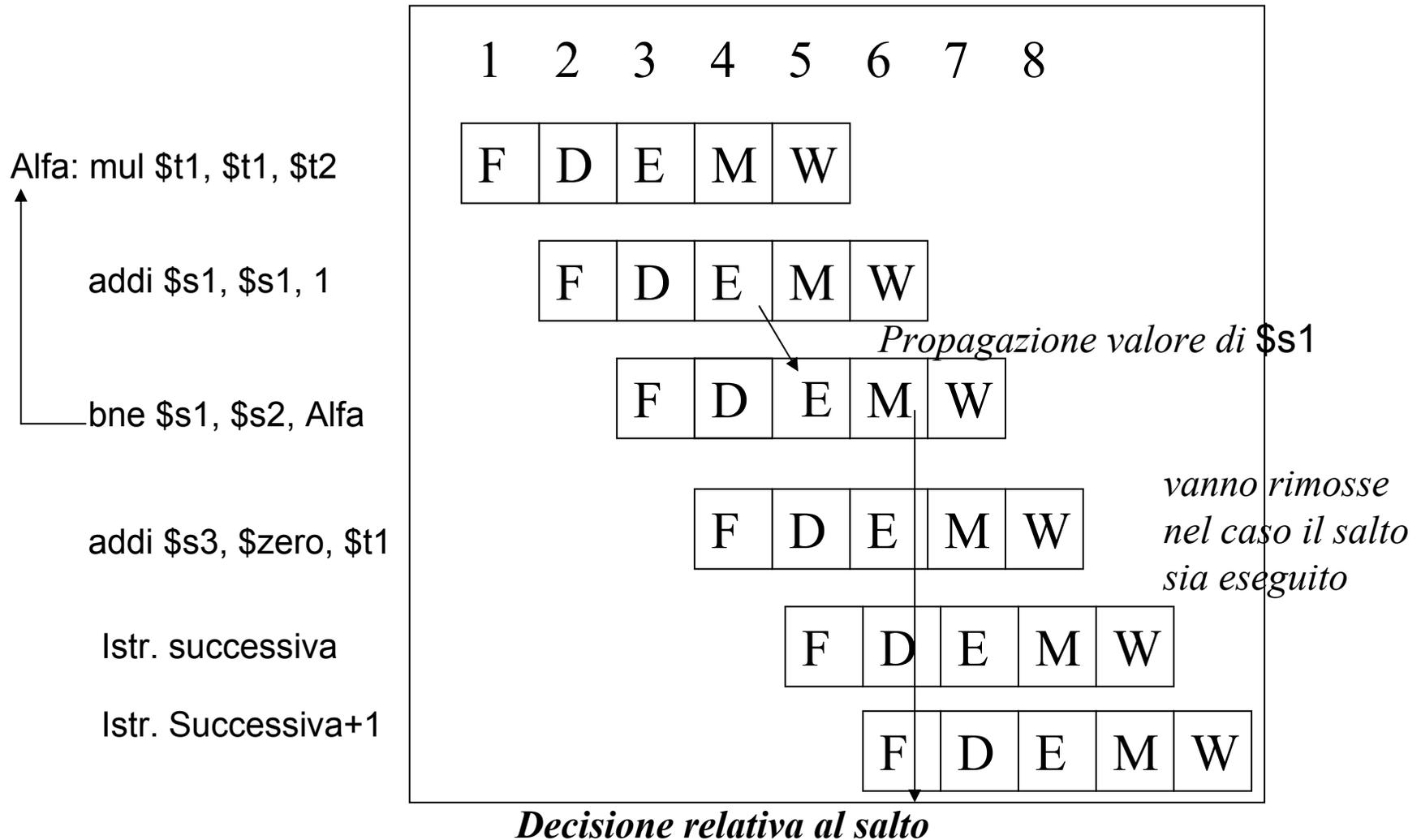


Diagramma temporale dell'esecuzione con salto ritardato: caso 1 (salto eseguito)

Decisione relativa al salto nello stadio M

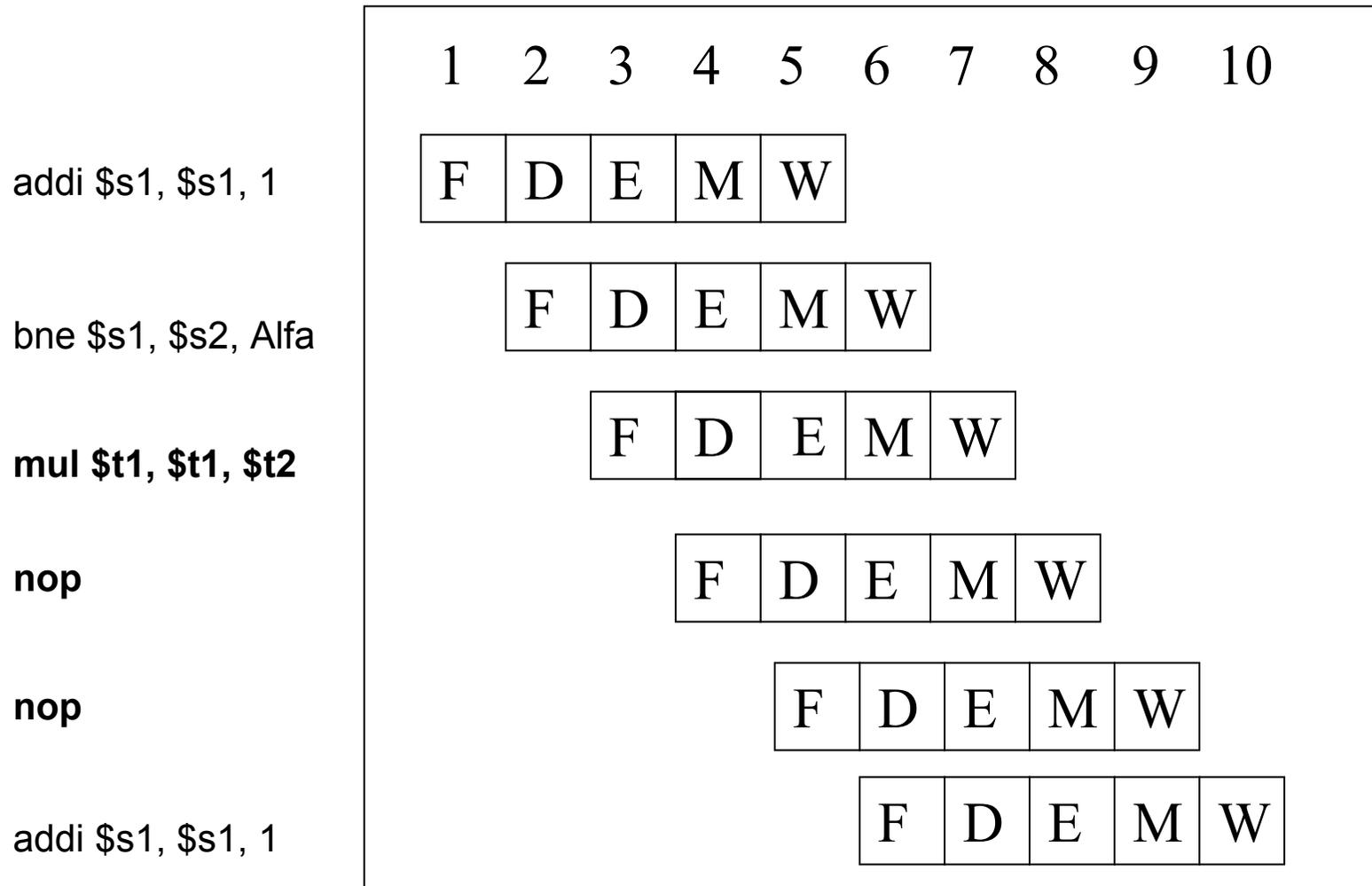
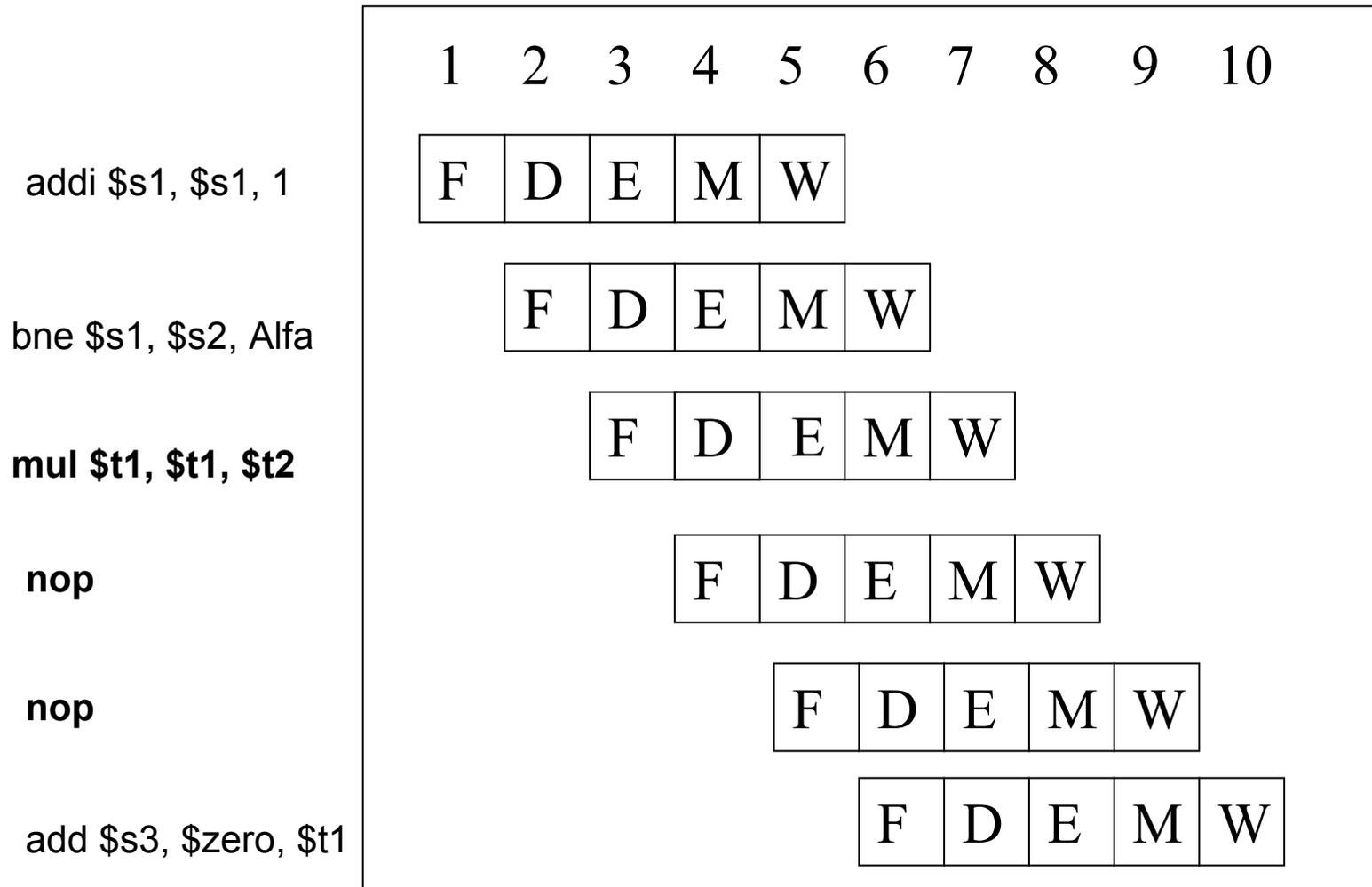


Diagramma temporale dell'esecuzione con salto ritardato: caso 2 (salto non eseguito)



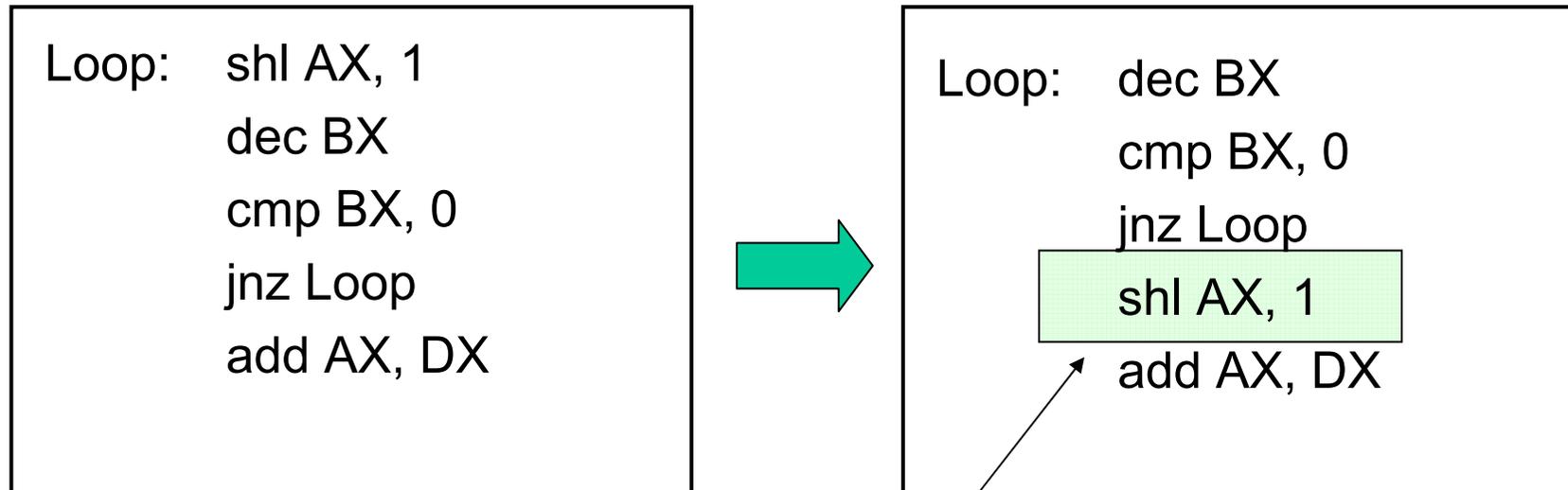
Esempio (2)

- Segmento di codice in assembler Intel

```
Loop: shl      AX, 1
      dec      BX
      cmp      BX, 0
      jnz      Loop
      add      AX, DX
```

NB: fa lo shift a sinistra di AX per un numero BX volte

Riordino delle istruzioni nel caso di salto ritardato



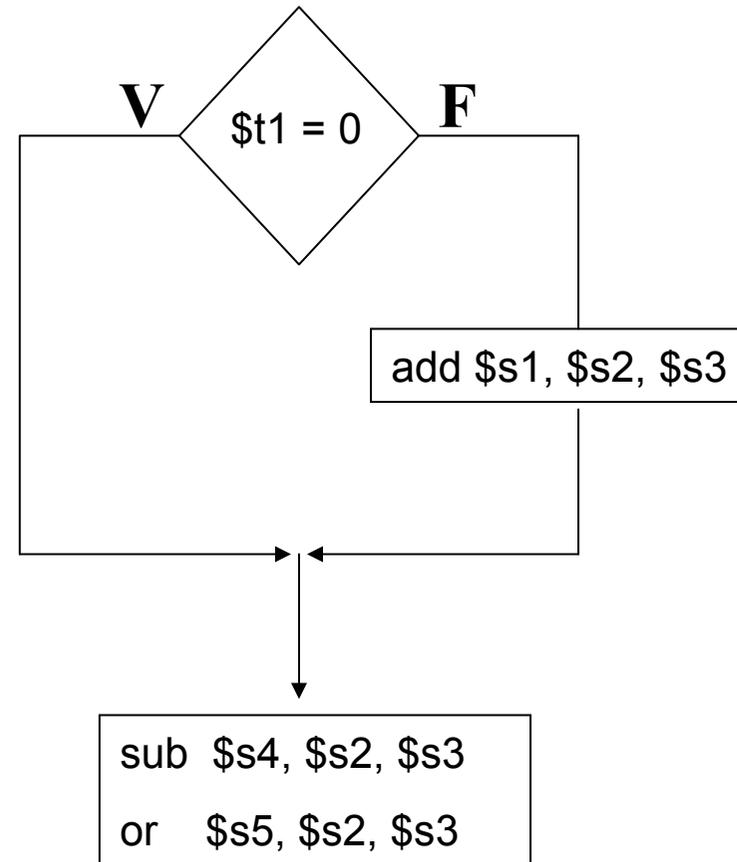
*Istruzione inserita nello slot di ritardo
(viene sempre eseguita)*

Esercizio: analizzare l'esecuzione del codice su una pipeline a 2 stadi

Esempio (3)

- Segmento di codice in assembler MIPS

```
Alfa:    beq    $t1, $zero, Alfa
         add    $s1, $s2, $s3
         sub    $s4, $s2, $s3
         or     $s5, $s2, $s3
```



Esempio (3) con salto ritardato

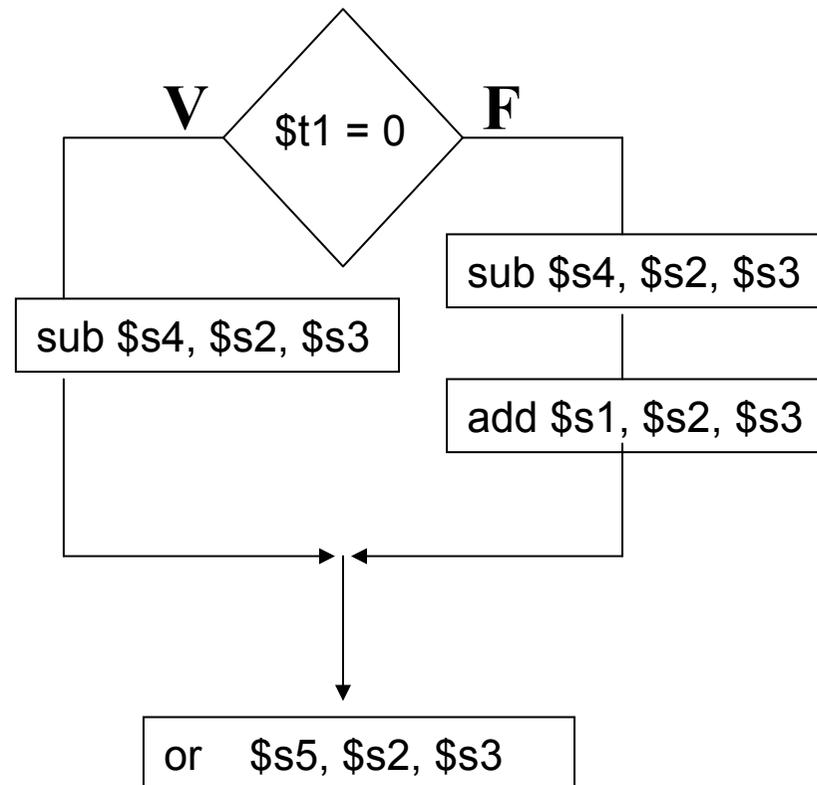
- Segmento di codice in assembler MIPS

```
beq    $t1, $zero, Alfa
```

```
sub    $s4, $s2, $s3
```

```
add    $s1, $s2, $s3
```

```
Alfa:  or    $s5, $s2, $s3
```



Esercizio: analizzare l'esecuzione del codice su una pipeline a 2 stadi e a 5 stadi

Calcolo di prestazioni

- Si ipotizzi che il 20% delle istruzioni eseguite da un calcolatore siano istruzioni di salto
- Il calcolatore adotta la tecnica del salto ritardato con 1 intervallo di ritardo
- Determinare il CPI nel caso in cui il compilatore sia in grado di utilizzare l'85% degli intervalli di ritardo (si assuma che non esistano altre cause di stallo oltre ai salti)

Soluzione:

$$\text{CPI} = 1 + \underbrace{0,20 \times 0,15 \times 1}_{\text{Nel 15\% dei salti non viene utilizzato l'intervallo di ritardo, e quindi 1 ciclo viene perso}} = 1,3$$

Nel 15% dei salti non viene utilizzato l'intervallo di ritardo, e quindi 1 ciclo viene perso

Calcolo di prestazioni

- Si ipotizzi che il 20% delle istruzioni eseguite da un calcolatore siano istruzioni di salto
- Il calcolatore adotta la tecnica del salto ritardato con 2 intervalli di ritardo
- Determinare il CPI supponendo che il compilatore sia in grado di riempire il primo intervallo nell'85% delle volte, mentre riempie il secondo solo nel 20% dei casi (si assuma che non esistano altre cause di stallo oltre ai salti)

Soluzione:

nel 15% dei casi perde 2 cicli, nel 65% perde 1 ciclo (riempie un solo slot di ritardo), nel 20% non perde nulla

$$\Rightarrow \text{CPI} = 1 + 0,20 \times \underbrace{(0,15 \times 2)}_{\substack{\text{nel 15\% dei casi} \\ \text{perde 2 cicli}}} + \underbrace{0,65 \times 1}_{\substack{\text{nel 65\% dei casi} \\ \text{perde 1 ciclo}}} = 1,19$$