

Simulazione 1

Sia data la seguente sequenza di istruzioni assembler (ideali) e la corrispondente codifica esadecimale (ideale) con codice operativo ed operandi di ampiezza 1 byte:

```

    mov c, R1      ! B0 0C ! inizializza R1 al valore 12 decimale
X:  mov 4, R2      ! B1 04 ! inizializza R2 al valore 4 decimale
    mov [R1], R3   ! BE 13 ! pone in R3 il contenuto di mem[R1]
    add R2, R3     ! A1 23 ! somma R2 ad R3 ponendo il risultato in R2
    add R4, R2     ! A1 42 ! somma R4 ad R2 ponendo il risultato in R4
    deci R1, 4    ! 90 04 ! decrementa R1 di 4 decimale
    jnz X         ! 75 F9 ! ritorna ad X se R1 non vale 0
    ret          ! 16      ! ritorna al chiamante senza modificare il
                                ! contenuto dei registri R1,R2,R3,R4

```

Si assuma la presenza di una cache dati ed una cache istruzioni, con associazione 2-way (con politica di rimpiazzo LRU) e dimensione di blocco 4B. Si simuli il comportamento delle cache sotto le seguenti condizioni:

- cache istruzioni e cache dati entrambe di 16B
- cache istruzioni di 16B e cache dati di 8B
- cache istruzioni e cache dati entrambe di 8B

Simulazione 2

Sia data la seguente sequenza di istruzioni assembler (ideali) e la corrispondente codifica esadecimale (ideale) con codice operativo ed operandi di ampiezza 1 byte:

```

    mov c, R1      ! B0 0C ! inizializza R1 al valore 12 decimale
X:  mov 4, R2      ! B1 04 ! inizializza R2 al valore 4 decimale
    mov R3, [R1]   ! BF 31 ! scrive il contenuto di R3 in mem[R1]
    add R2, R3     ! A1 23 ! somma R2 ad R3 ponendo il risultato in R2
    add R4, R2     ! A1 42 ! somma R4 ad R2 ponendo il risultato in R4
    deci R1, 2    ! 90 02 ! decrementa R1 di 2 decimale
    jnz X         ! 75 F9 ! ritorna ad X se R1 non vale 0
    ret          ! 16      ! ritorna al chiamante senza modificare il
                                ! contenuto dei registri R1,R2,R3,R4

```

Stesse cache di sopra. Si simuli il comportamento delle cache sotto le seguenti condizioni (sia con write back che con write through):

- cache istruzioni e dati entrambe di 16B
- cache istruzioni di 16B e cache dati di 8B
- cache istruzioni di 8B e cache dati di 16B
- cache istruzioni e dati entrambe di 8B