

# Architettura degli Elaboratori

Esercitazione



**Barbara Masucci**

UNIVERSITA' DEGLI STUDI DI SALERNO

**DIPARTIMENTO DI INFORMATICA**

**DIPARTIMENTO DI ECCELLENZA**

# Su cosa ci esercitiamo oggi?

- Ancora istruzioni MIPS
  - Istruzioni per prendere decisioni
  - Cicli



# Esercizio 1

- Si consideri il seguente **frammento** in assembler MIPS

```
    slt $t0, $s0, $s1  
    bne $t0, $s1, LABEL  
    add $s0, $s0, $s1  
LABEL: sll $s0, $s0, 1
```

- Si supponga che \$s0 contenga 1 ed \$s1 contenga 20
- Quali sarà il **contenuto** di \$s0 al termine della sequenza?
  - A. 21
  - B. 2
  - C. 42
  - D. Nessuna delle risposte precedenti.



# Esercizio 1: Soluzione

- Si consideri il seguente **frammento** in assembler MIPS

```
slt $t0, $s0, $s1#$t0=1 perché  $1 < 20$   
bne $t0, $s1, LABEL#esegue il salto perché  $1 \neq 20$   
add $s0, $s0, $s1#non eseguita  
LABEL: sll $s0, $s0, 1#$s0 viene shiftato a sx di un posto
```

- Si supponga che \$s0 contenga 1 ed \$s1 contenga 20
- Al termine del frammento di codice, \$s0 contiene 2
- La risposta esatta è **B**
  - A. 21
  - B. 2
  - C. 42
  - D. Nessuna delle risposte precedenti.



# Esercizio 2

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT
      addi $s0, $s0, 1
      j LOOP
EXIT:...
```

- Si supponga che \$s0 contenga 2 ed \$s1 contenga 6
- Quali saranno i **contenuti finali** di \$s0 e di \$s1 al termine del ciclo?
- A. 2 e 6  
B. 6 e 2  
C. 2 e 2  
D. Nessuna delle risposte precedenti.



# Esercizio 2: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT#non esegue il salto perché 2 ≠ 6  
      addi $s0, $s0, 1#$s0 contiene 3  
      j LOOP#torna all'inizio del ciclo  
EXIT:...
```

- Si supponga che \$s0 contenga 2 ed \$s1 contenga 6
- Quali saranno i **contenuti finali** di \$s0 e di \$s1 al termine del ciclo?

A. 2 e 6

B. 6 e 2

C. 2 e 2

D. Nessuna delle risposte precedenti.



# Esercizio 2: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT#non esegue il salto perché 3 ≠ 6  
      addi $s0, $s0, 1#$s0 contiene 4  
      j LOOP#torna all'inizio del ciclo  
EXIT:...
```

- Si supponga che \$s0 contenga 2 ed \$s1 contenga 6
- Quali saranno i **contenuti finali** di \$s0 e di \$s1 al termine del ciclo?
  - A. 2 e 6
  - B. 6 e 2
  - C. 2 e 2
  - D. Nessuna delle risposte precedenti.



# Esercizio 2: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT#non esegue il salto perché 4 ≠ 6
      addi $s0, $s0, 1#$s0 contiene 5
      j LOOP#torna all'inizio del ciclo
EXIT:...
```

- Si supponga che \$s0 contenga 2 ed \$s1 contenga 6
  - Quali saranno i **contenuti finali** di \$s0 e di \$s1 al termine del ciclo?
- A. 2 e 6  
B. 6 e 2  
C. 2 e 2  
D. Nessuna delle risposte precedenti.





# Esercizio 2: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT#non esegue il salto perché 5 ≠ 6
      addi $s0, $s0, 1#$s0 contiene 6
      j LOOP#torna all'inizio del ciclo
EXIT:...
```

- Si supponga che \$s0 contenga 2 ed \$s1 contenga 6
  - Quali saranno i **contenuti finali** di \$s0 e di \$s1 al termine del ciclo?
- A. 2 e 6                      C. 2 e 2  
B. 6 e 2                      D. Nessuna delle risposte precedenti.



# Esercizio 2: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

```
LOOP: beq $s0, $s1, EXIT#esegue il salto perché 6 = 6  
      addi $s0, $s0, 1#non eseguita  
      j LOOP#non eseguita  
EXIT:...
```

- Quindi al termine del ciclo, \$s0 contiene 6 ed \$s1 contiene 6
- La risposta esatta è **D**
  - A. 2 e 6
  - B. 6 e 2
  - C. 2 e 2
  - D. Nessuna delle risposte precedenti.



# Esercizio 3

- Si consideri il seguente **ciclo** in assembler MIPS

```
CICLO: slt $t2, $zero, $t1  
       beq $t2, $zero, FINE  
       addi $t1, $t1, -1  
       addi $s2, $s2, 2  
       j CICLO
```

FINE:

- Si supponga che \$t1 contenga il valore 10
- Quale sarà il **contenuto finale di \$s2**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #all'inizio del ciclo \$t2 contiene 1  
#perché  $0 < 10$

`beq $t2, $zero, FINE` # $\$t2 \neq 0$ , quindi non esegue il  
salto

`addi $t1, $t1, -1` # $\$t1$  contiene 9

`addi $s2, $s2, 2` # $\$s2$  contiene 2

`j CICLO` # $\$$ torna all'inizio del ciclo

FINE:

- Si supponga che  $\$t1$  contenga il valore 10
- Quale sarà il **contenuto finale di  $\$s2$** , supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 9$   
`beq $t2, $zero, FINE` #`$t2` ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 8  
`addi $s2, $s2, 2` #`$s2` contiene 4  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 8$   
`beq $t2, $zero, FINE` #`$t2`  $\neq 0$ , quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 7  
`addi $s2, $s2, 2` #`$s2` contiene 6  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 7$   
`beq $t2, $zero, FINE` #`$t2` ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 6  
`addi $s2, $s2, 2` #`$s2` contiene 8  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` # \$t2 contiene 1 perché  $0 < 6$   
`beq $t2, $zero, FINE` # \$t2 ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` # \$t1 contiene 5  
`addi $s2, $s2, 2` # \$s2 contiene 10  
`j CICLO` # \$torna all'inizio del ciclo

FINE:

- Si supponga che \$t1 contenga il valore 10
- Quale sarà il **contenuto finale di \$s2**, supponendo che il suo valore iniziale sia 0?





# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 5$   
`beq $t2, $zero, FINE` #`$t2` ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 4  
`addi $s2, $s2, 2` #`$s2` contiene 12  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 4$   
`beq $t2, $zero, FINE` #`$t2`  $\neq 0$ , quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 3  
`addi $s2, $s2, 2` #`$s2` contiene 14  
`j CICLO` #`$` torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 3$   
`beq $t2, $zero, FINE` #`$t2` ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 2  
`addi $s2, $s2, 2` #`$s2` contiene 16  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 1 perché  $0 < 2$   
`beq $t2, $zero, FINE` #`$t2`  $\neq 0$ , quindi non esegue il salto

`addi $t1, $t1, -1` #`$t1` contiene 1  
`addi $s2, $s2, 2` #`$s2` contiene 18  
`j CICLO` #`$`torna all'inizio del ciclo

FINE:

- Si supponga che `$t1` contenga il valore 10
- Quale sarà il **contenuto finale di `$s2`**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` # \$t2 contiene 1 perché  $0 < 1$   
`beq $t2, $zero, FINE` # \$t2 ≠ 0, quindi non esegue il salto

`addi $t1, $t1, -1` # \$t1 contiene 0  
`addi $s2, $s2, 2` # \$s2 contiene 20  
`j CICLO` # torna all'inizio del ciclo

FINE:

- Si supponga che \$t1 contenga il valore 10
- Quale sarà il **contenuto finale di \$s2**, supponendo che il suo valore iniziale sia 0?



# Esercizio 3: Soluzione

- Si consideri il seguente **ciclo** in assembler MIPS

CICLO: `slt $t2, $zero, $t1` #`$t2` contiene 0  
      `beq $t2, $zero, FINE` #`$t2=0`, quindi salta a FINE  
      `addi $t1, $t1, -1` #non eseguita  
      `addi $s2, $s2, 2` #non eseguita  
      `j CICLO` #non eseguita  
FINE: #esci dal ciclo

- Il **contenuto finale di \$s2** è 20



# Esercizio 4

- Scrivere il codice C corrispondente al seguente **ciclo** in assembler MIPS

```
        addi $t1, $zero, 0
CICLO: lw $s1, 0($s0)
        add $s2, $s2, $s1
        addi $s0, $s0, 4
        addi $t1, $t1, 1
        slti $t2, $t1, 100
        bne $t2, $zero, CICLO
```

- Si supponga che
  - \$s2 contenga la **variabile intera result**
  - \$s0 contenga l'**indirizzo base del vettore M**



# Esercizio 4: Soluzione

addi \$t1, \$zero, 0 # \$t1 contiene 0  
CICLO: lw \$s1, 0(\$s0) # \$s1 carica in \$s1 la word M[0]  
add \$s2, \$s2, \$s1 # \$s2 contiene result+M[0]  
addi \$s0, \$s0, 4 # \$s0 contiene l'indirizzo della word M[1]  
addi \$t1, \$t1, 1 # \$t1 contiene 1  
slti \$t2, \$t1, 100 # \$t2=1 perché 1 < 100  
bne \$t2, \$zero, CICLO # poichè 1 ≠ 0, torna all'inizio del  
ciclo





# Esercizio 4: Soluzione

CICLO: lw \$s1, 0(\$s0) #\$carica in \$s1 la word M[1]  
add \$s2, \$s2, \$s1 #\$s2 contiene result+M[0]+M[1]  
addi \$s0, \$s0, 4 #\$s0 contiene l'indirizzo della word M[2]  
addi \$t1, \$t1, 1 #\$t1 contiene 2  
slti \$t2, \$t1, 100 #\$t2=1 perché 2 < 100  
bne \$t2, \$zero, CICLO #poichè 1≠0, torna all'inizio del  
ciclo



# Esercizio 4: Soluzione

CICLO: lw \$s1, 0(\$s0) #\$carica in \$s1 la word M[2]  
add \$s2, \$s2, \$s1 #\$s2 contiene result+M[0]+M[1]+M[2]  
addi \$s0, \$s0, 4 #\$s0 contiene l'indirizzo della word M[3]  
addi \$t1, \$t1, 1 #\$t1 contiene 3  
slti \$t2, \$t1, 100 #\$t2=1 perché 3 < 100  
bne \$t2, \$zero, CICLO #poichè 1≠0, torna all'inizio del  
ciclo



# Esercizio 4: Soluzione

CICLO: lw \$s1, 0(\$s0) #\$carica in \$s1 la word M[3]  
add \$s2, \$s2, \$s1 #\$s2 contiene  
result+M[0]+M[1]+M[2]+M[3]  
addi \$s0, \$s0, 4 #\$s0 contiene l'indirizzo della word M[4]  
addi \$t1, \$t1, 1 #\$t1 contiene 4  
slti \$t2, \$t1, 100 #\$t2=1 perché 4 < 100  
bne \$t2, \$zero, CICLO #poichè 1≠0, torna all'inizio del  
ciclo



# Esercizio 4: Soluzione

- In definitiva, ecco il codice C corrispondente al **ciclo** in assembler MIPS

```
        addi $t1, $zero, 0
CICLO: lw $s1, 0($s0)
        add $s2, $s2, $s1
        addi $s0, $s0, 4
        addi $t1, $t1, 1
        slti $t2, $t1, 100
        bne $t2, $zero, CICLO
```

```
for (i=0; i<100; i++) {
    result += M[i];
}
```

