

Esercizio 1

- n Sapendo che i campi `codice operativo` e `funct` della `sll` contengono tutti zeri, determinare il codice assembly corrispondente al seguente frammento di programma in linguaggio macchina

00010101011 00101 0000000000000011
00000001011 10000 0110000000100000
00000000000 01100 0101000011000000
10101110001 01010 0000000000001000
00000001000 00000 0000000000001000
00100010000 00010 0000000000000100

Convenzioni per i formati

n **load / store :**

- **base nel primo campo**
- **dato nel secondo campo**

n **add (e altre con 3 registri)**

- **2 operandi sorgente nei primi due campi registri**
- **risultato nel terzo campo registro**

n **addi, slti (e altre con immediato)**

- **sorgente nel primo dei due campi registro**
- **risultato nel secondo**

n **beq, bne**

- **2 registri nei due campi (stesso ordine)**

n **sll, slr**

- **0 nel primo campo registro**
- **sorgente nel secondo campo registro**
- **risultato nel terzo campo registro**

Soluzione dell'esercizio 1

<i>I</i>	000101	01011	00101	0000000000000011		
<i>R</i>	000000	01011	10000	01100	00000	100000
<i>R</i>	000000	00000	01100	01010	00011	000000
<i>I</i>	101011	10001	01010	0000000000001000		
<i>R</i>	000000	01000	00000	00000	00000	001000
<i>I</i>	001000	10000	00010	0000000000000100		

```

bne $t3,$a1,L
add $t4,$t3,$s0
sll $t2,$t4,3
sw $t2,8($s1)
L: jr $t0
addi $v0,$s0,4
    
```

000101 = bne formato *I*
 101011 = sw formato *I*
 001000 = addi formato *I*
 000000 = ? formato *R*

100000 = add
 000000 = sll
 001000 = jr

Esercizio 2

- n Sapendo che i campi codice operativo e funct della sll contengono tutti zeri, determinare il codice assembleativo corrispondente al seguente frammento di programma in linguaggio macchina

10001110000 01000 00000000000001100
001010 01000 01001 0000000000001010
000100 01001 00000 0000000000000010
000000 00000 01000 0100000010000000
000000 01000 10010 0001000000100000
000000 11111 00000 0000000000001000

Convenzioni per i formati

- n **load / store :**
 - **base nel primo campo**
 - **dato nel secondo campo**
- n **add (e altre con 3 registri)**
 - **2 operandi sorgente nei primi due campi registri**
 - **risultato nel terzo campo registro**
- n **addi, slti (e altre con immediato)**
 - **sorgente nel primo dei due campi registro**
 - **risultato nel secondo**
- n **beq, bne**
 - **2 registri nei due campi (stesso ordine)**
- n **sll, slr**
 - **0 nel primo campo registro**
 - **sorgente nel secondo campo registro**
 - **risultato nel terzo campo registro**

Soluzione dell'esercizio 2

I	100011	10000	01000	00000000000001100		
I	001010	01000	01001	00000000000001010		
I	000100	01001	00000	00000000000000010		
R	000000	00000	01000	01000	00010	000000
R	000000	01000	10010	00010	00000	100000
R	000000	11111	00000	00000	00000	001000

```

lw    $t0,12($s0)
slti  $t1,$t0,10
beq   $t1,$zero,L
sll   $t0,$t0,2
add   $v0,$t0,$s2
L:    jr    $ra
    
```

100011 = lw formato **I**
 001010 = slti formato **I**
 000100 = beq formato **I**
 000000 = ? formato **R**

100000 = add
 000000 = sll
 001000 = jr