Architettura degli elaboratori lezione 8

📚 Appunti di Davide Scarlata 2024/2025

👩‍🏫 **Prof:** Claudio Schifanella
✉️ **Mail:** claudio.schifanella@unito.it
📌 **Corso:** C
🔗 **Moodle Unito**
🗓️ **Data:** 24/03/2025

---

esempio moltiplicazione per 10 senza mul

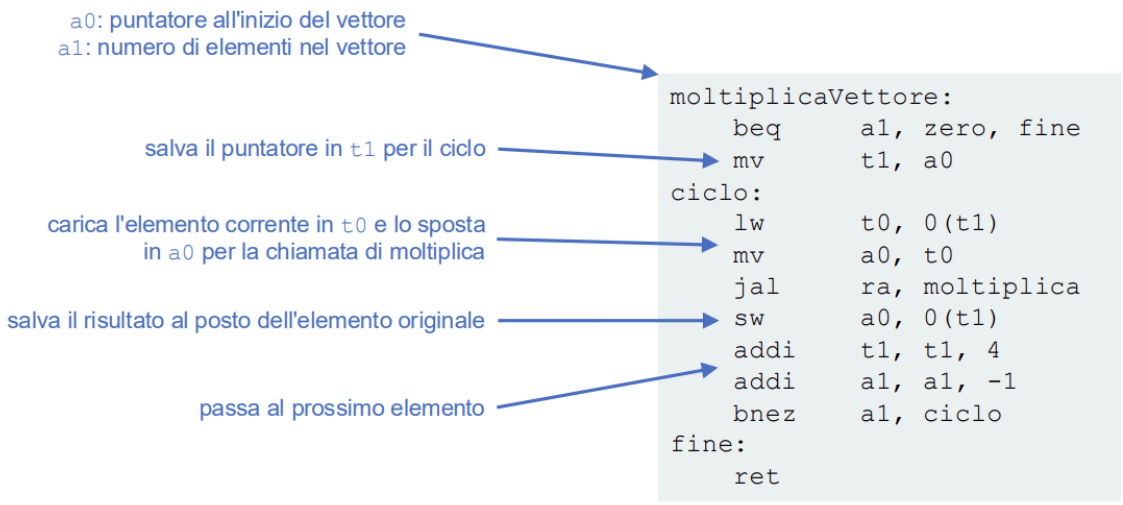
```
moltiplica:
slli t0, a0, 3 # t0 = a0 * 8
slli t1, a0, 1 # t1 = a0 * 2
add a0, t0, t1
ret
```

è una funzione foglia in quanto non chiama altre funzioni , rispetta le convenzioni di chiamata

---

• Scrivere una funzione moltiplicaVettore che
• Dato un vettore di interi (word)
• Restituisce il vettore sostituendo ogni elemento con il suo prodotto per 10

Una possibile soluzione: chiediamo a ChatGPT



perchè è sbagliato ?

- P1 - La funzione non è foglia ma il registro ra non viene salvato e ripristinato
- P2 - Convenzioni di chiamata non rispettate
- Il registro t1 viene modificato dall'invocazione di moltiplica
- Il registro a1 potrebbe essere modificato da moltiplica

come risolviamo?

p1)

```
moltiplicaVettore:
addi sp,sp,-4              risolve p1
sw ra,0(sp)                risolve p1
beq a1, zero, fine
mv t1, a0
ciclo:
lw t0, 0(t1)
addi sp, sp, -8          risolve p2
sw t1, 0(sp)             risolve p2
sw a1, 4(sp)             risolve p2
mv a0, t0
jal ra, moltiplica
lw t1, 0(sp)           risolve p2
lw a1, 4(sp)           risolve p2
addi sp, sp, 8         risolve p2
sw a0, 0(t1)
addi t1, t1, 4
addi a1, a1, -1
bnez a1, ciclo
fine:
```

```
lw ra, 0(sp)        risolve p1
addi sp, sp, 4      risolve p1
ret
```

è corretto ma inefficiente perchè salva a1 e t1 che non vengono modificati da moltiplica come risolvere?

```
moltiplicaVettore:
    addi sp, sp, -12
    sw s2, 8(sp)
    sw s1, 4(sp)
    sw ra, 0(sp)
    beq a1, zero, fine
    mv s1, a0
    mv s2, a1
ciclo:
    lw t0, 0(s1)
    mv a0, t0
    jal ra, moltiplica
    sw a0, 0(s1)
    addi s1, s1, 4
    addi s2, s2, -1
    bnez s2, ciclo
fine:
    lw ra, 0(sp)
    lw s1, 4(sp)
    lw s2, 8(sp)
    addi sp, sp, 12
ret
```

Operandi immediati ampi

• Problema: è possibile caricare in un registro una costante a 32 bit? Supponiamo di voler caricare nel registro x5 il valore 0x12345678

• si introduce una nuova istruzione lui (load upper immediate, tipo U) che carica i 20 bit più significativi della costante nei bit da 12 a 31 di un registro

• Con una operazione di or immediato si impostano i 12 bit meno significativi rimasti

```
lui x5,0x12345
```

x5 ..... 00010010 00110100 01010000 00000000

```
ori x5,x5,0x678
```

x5 ..... 00010010 00110100 01010110 01111000

Salti con offset più grandi

• Come per le costanti, anche i salti possono essere eseguiti anche ad istruzioni più lontane • RISC-V introduce la possibilità di salto in un intervallo pari a 2^32

- Nuova istruzione

```
auipc rd, offset
```

Add Upper Immediate PC

Tipo U in linguaggio macchina

- Inserisce nel registro `rd` l'indirizzo di PC + (offset << 12)

Esempio: `auipc x5, 0x12345` ⟶ `x5 = PC + 0x12345000`

- Se usiamo `auipc` con i 20 bit più significativi dell'offset, allora possiamo aggiungere questa istruzione una istruzione che calcola
  - PC = rd + offset[31..0]
- Otteniamo come risultato un salto incondizionato con offset più esteso
- L'istruzione da considerare è `jalr`! Ricapitolando:

```
auipc rd, offset[31..12]
jalr x0, offset[0..11](rd)
```

- Realizza un salto incondizionato a `PC + offset[31..0]`

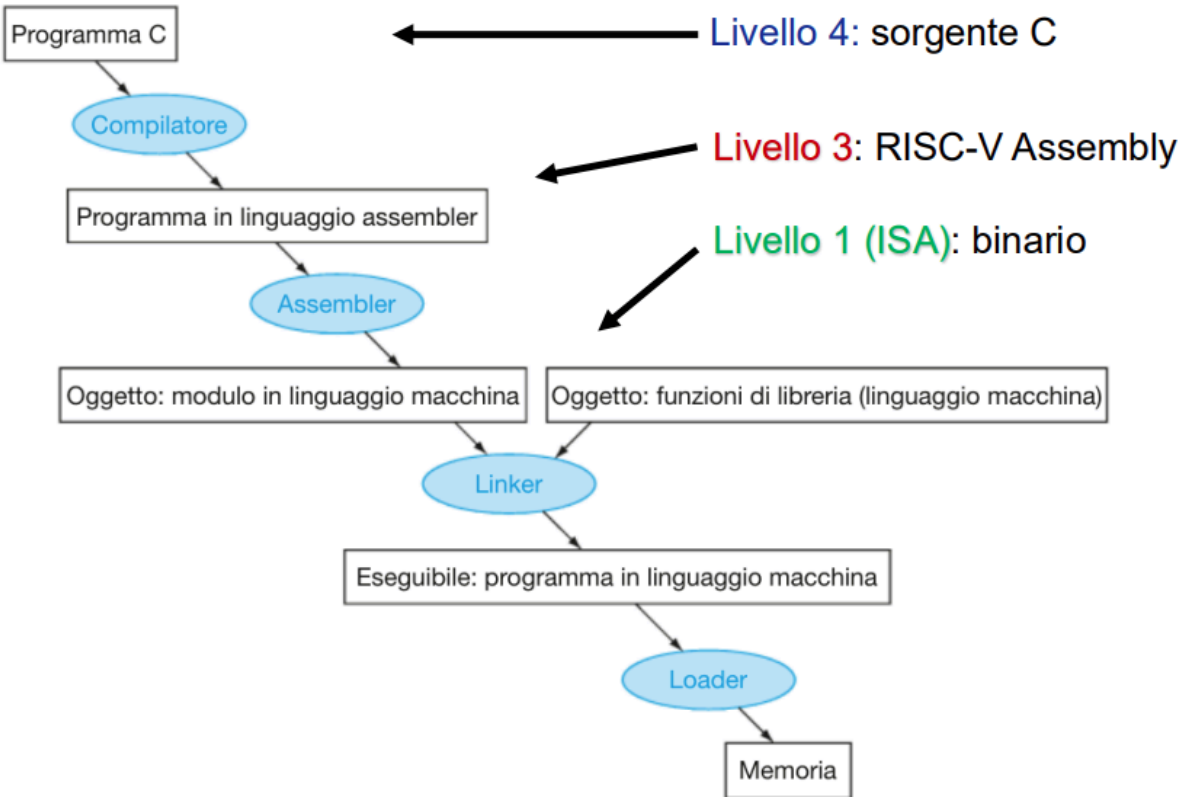
Riassunto dei formati delle istruzioni RISC-V

Nome (dimensione del campo)	Campi						Commenti
	7 bit	5 bit	5 bit	3 bit	5 bit	7 bit	
Tipo R	funz7	rs2	rs1	funz3	rd	codop	Istruzioni aritmetiche
Tipo I	Immediato[11:0]		rs1	funz3	rd	codop	Istruzioni di caricamento dalla memoria e aritmetica con costanti
Tipo S	immed[11:5]	rs2	rs1	funz3	immed[4:0]	codop	Istruzioni di trasferimento alla memoria (store)
Tipo SB	immed[12, 10:5]	rs2	rs1	funz3	immed[4:1,11]	codop	Istruzioni di salto condizionato
Tipo UJ	immediato[20, 10:1, 11, 19:12]				rd	codop	Istruzioni di salto incondizionato
Tipo U	immediato[31:12]				rd	codop	Formato caricamento stringhe di bit più significativi

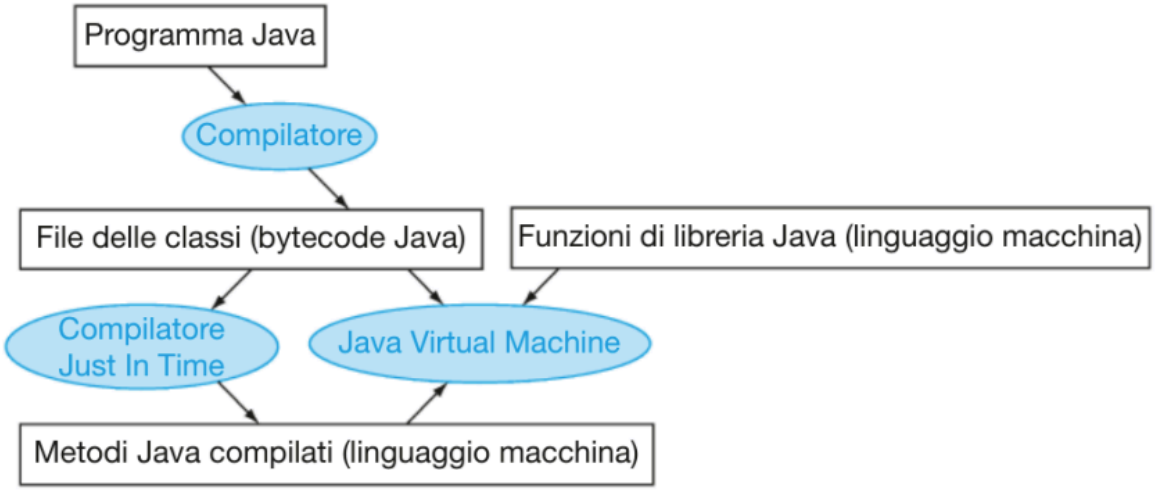
---

Il livello del Linguaggio Assembly

Sequenza di passi di traduzione per il C

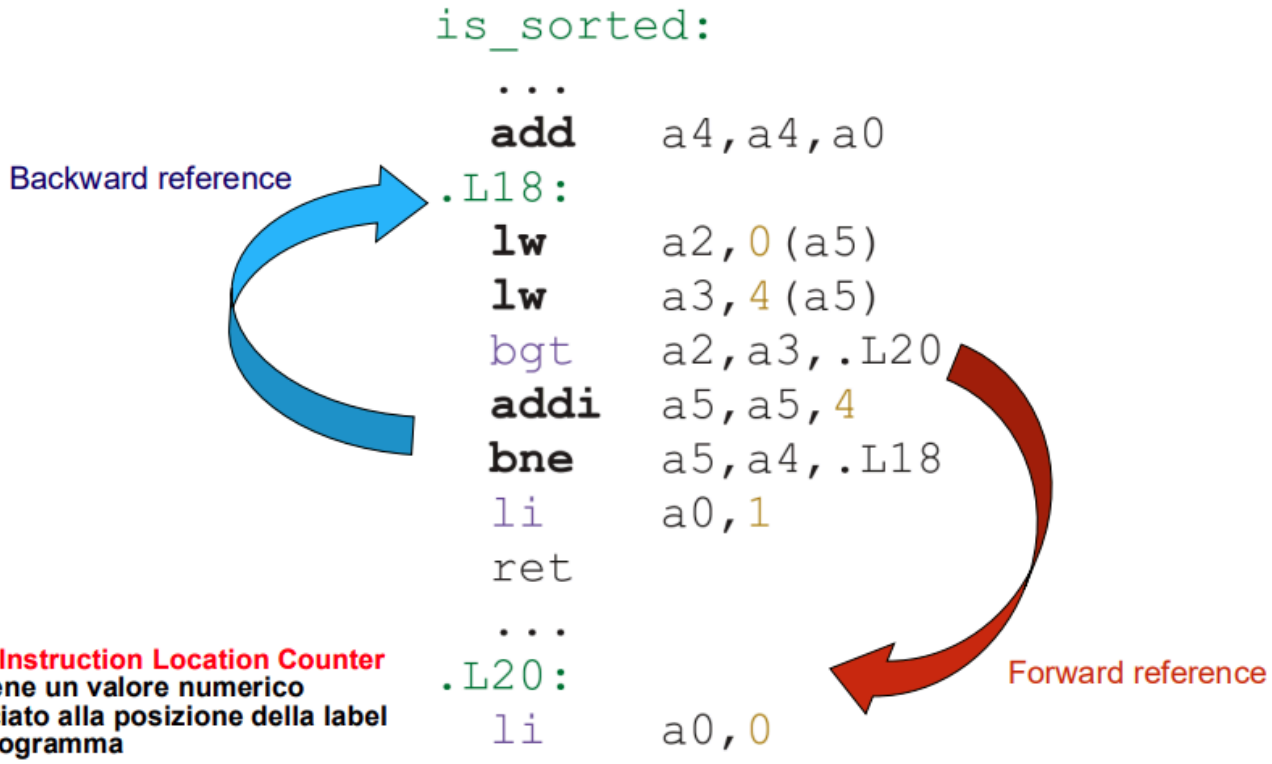


Traduzione del codice Java



- Il programma Java è eseguito da un interprete (Java Virtual Machine)
- **La JVM può invocare il compilatore Just In Time (JIT), che compila i metodi del linguaggio Java nel linguaggio macchina del calcolatore sul quale è in esecuzione**

Dal sorgente al modulo oggetto



non riesco a leggere tutte le istruzioni mi serve un'altro passo

ASSEMBLATORE a due PASSI

Problema delle forward reference !

1° Passo :

▪ individuazione di tutti i nomi (le etichette) che compaiono come riferimento simbolico di dati o di istruzioni

▪ creazione di una Symbol Table che contiene le etichette con la loro posizione relativa all'interno del programma

2° Passo :

▪ traduzione dei codici mnemonici delle istruzioni, degli operandi e delle etichette, mediante la consultazione della Symbol Table costruita nel 1° passo.