

# Parte IV

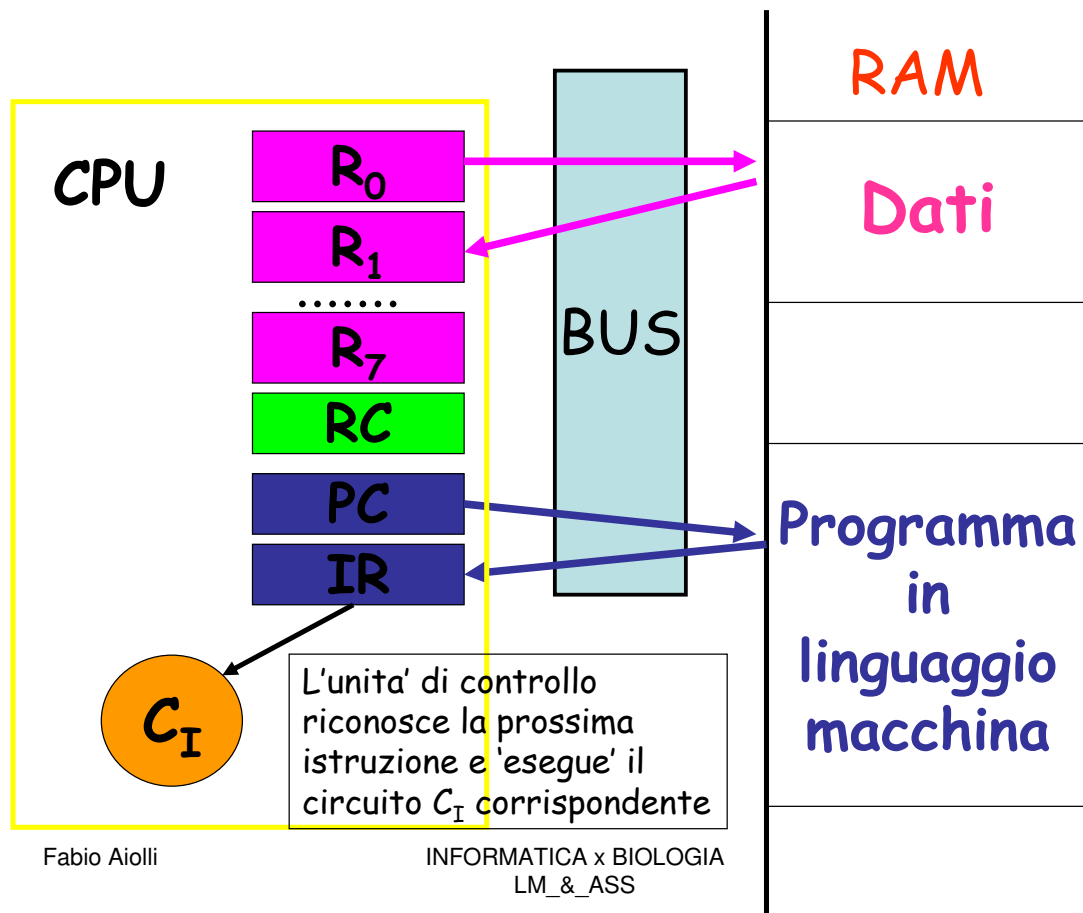
## Linguaggio Macchina e Assembler

### LINGUAGGIO MACCHINA

Descriveremo una CPU "MINIMA" dotata di un certo insieme di istruzioni  $I$  ciascuna realizzata da un corrispondente circuito  $C_I$ .

Questo insieme di istruzioni della CPU "MINIMA" costituisce il linguaggio macchina di "MINIMA".

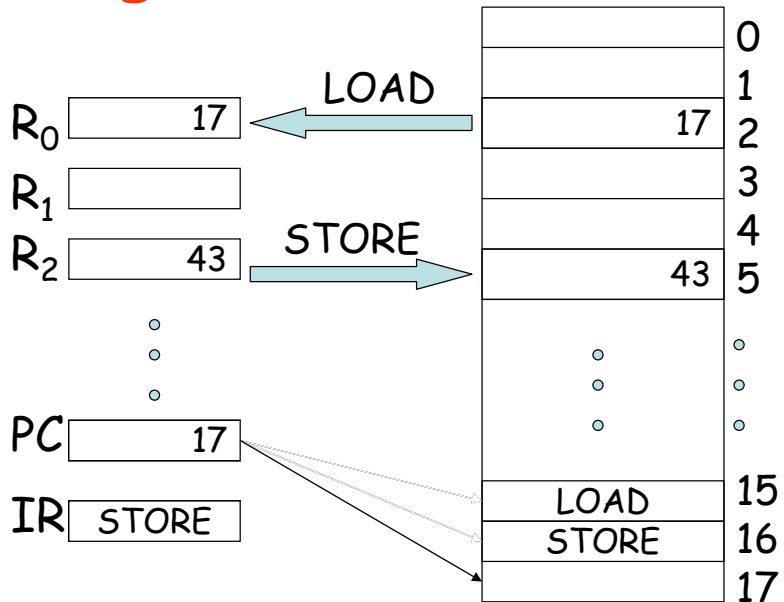
Semplificheremo l'approccio considerando che ogni istruzione sia memorizzata in una parola di memoria di 32 bit.



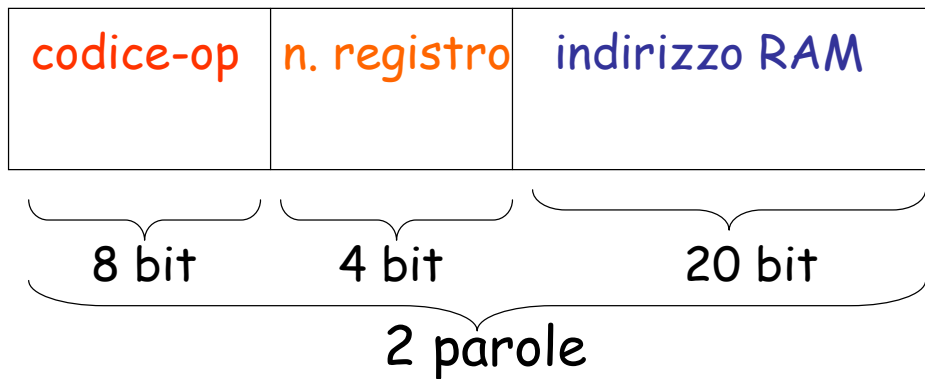
## 3 tipi di istruzioni macchina

- 1) **trasferimento** tra RAM e registri di calcolo della CPU
- 2) **operazioni aritmetiche**: somma, differenza, moltiplicazione e divisione
- 3) **operazioni di controllo**: confronto, salto e stop

# Istruzioni di trasferimento: registri $\Leftrightarrow$ RAM



## Formato delle istruzioni di trasferimento



**Codici:**

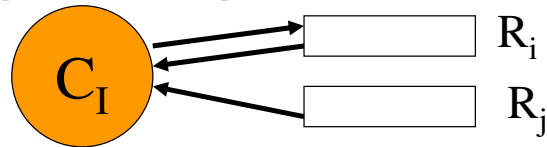
LOAD	00000000
STORE	00000001

**Esempio:**

00000000	0000	0000000000000000000010
00000001	0010	00000000000000000000101

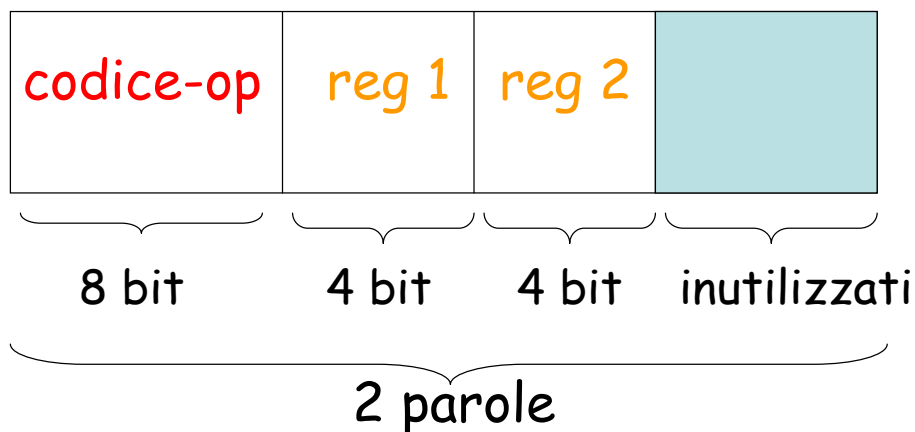
# Istruzioni aritmetiche

Eseguono somma, differenza, moltiplicazione e divisione usando i registri come operandi. Memorizzano il risultato nel primo registro argomento.



<b>ADD</b>	<b>00000010</b>	<b>FADD</b>	<b>00000011</b>
<b>SUB</b>	<b>00000100</b>	<b>FSUB</b>	<b>00000101</b>
<b>MULT</b>	<b>00000110</b>	<b>FMULT</b>	<b>00000111</b>
<b>DIV</b>	<b>00001000</b>	<b>FDIV</b>	<b>00001001</b>
<b>MOD</b>	<b>00001010</b>		

## Formato delle istruzioni aritmetiche

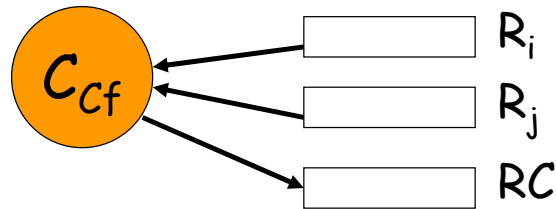


Esempio: **00000010** **0011** **0001** **xxxxxxxxxxxxxxxxxxxx**  
**00000011** **0011** **0001** **xxxxxxxxxxxxxxxxxxxx**

# Istruzione di confronto

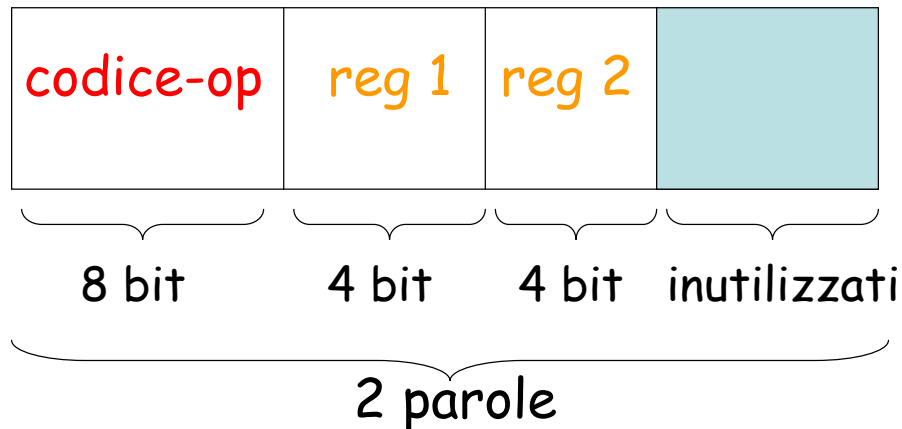
Paragona il contenuto di 2 registri  $R_i$  e  $R_j$

- se  $R_i < R_j$  memorizza -1 nel registro RC
- se  $R_i = R_j$  memorizza 0 in RC
- se  $R_i > R_j$  memorizza 1 in RC



Codici:            COMP    00100000  
                  FCOMP    00100001

## Formato dell'istruzione di confronto



Esempio: 00100000 0010 0101 xxxxxxxxxxxxxxxxxxxx  
          00100001 0010 0101 xxxxxxxxxxxxxxxxxxxx

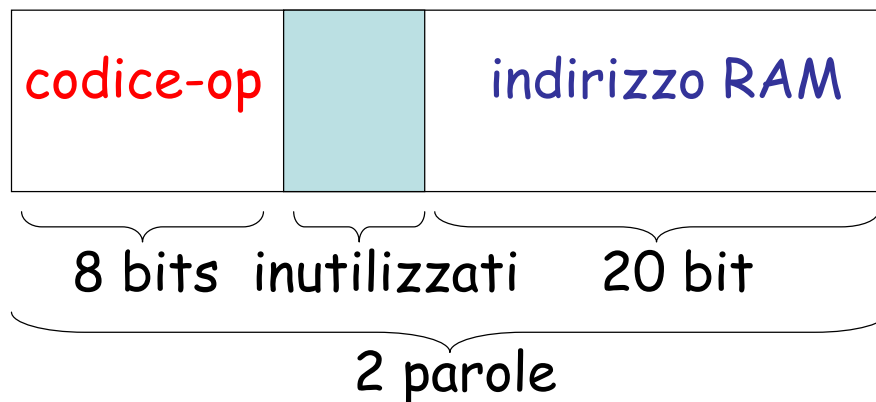
# Istruzione di salto

Permette di "saltare" ad un'altra istruzione del programma a seconda del contenuto del registro RC (cioè a seconda del risultato di un confronto)

<b>BRLT</b> (RC=-1)	01000001	<b>BRNE</b> (RC≠0)	01000100
<b>BRLE</b> (RC<1)	01000010	<b>BRGE</b> (RC>-1)	01000110
<b>BREQ</b> (RC=0)	01000011	<b>BRGT</b> (RC=1)	01000101
	<b>BRANCH</b>	10000000	

↑  
**Salto incondizionato!**

## Formato dell'istruzione di salto



**Esempio:** 01000001 xxxx 000000000000000000001001  
10000000 xxxx 000000000000000000001010

L'effetto di una istruzione di salto ad un indirizzo  $M$  è quindi quello di memorizzare  $M$  nel registro PC se si verifica la relativa condizione nel registro RC



# Esempio

Scriviamo un programma in linguaggio macchina che:

- trasferisce il contenuto delle 2 parole della RAM di indirizzi 64 e 68 nei registri  $R_0$  e  $R_1$
- somma i contenuti dei registri  $R_0$  ed  $R_1$
- trasferisce il risultato nella parola della RAM all'indirizzo 60

