



# ISA e linguaggio macchina

Prof. Alberto Borghese  
Dipartimento di Scienze dell'Informazione  
[borgese@di.unimi.it](mailto:borgese@di.unimi.it)  
Università degli Studi di Milano

Riferimento sul Patterson: capitolo 4.2 , 4.4, D1, D2.

A.A. 2014-2015 1/44 http:\\borgese.di.unimi.it\



# Introduzione alla CPU

- Istruzioni di salto dell'ISA MIPS
- I diversi tipi di istruzioni: Istruzioni di tipo R
- I diversi tipi di istruzioni: Istruzioni di tipo I
- I diversi tipi di istruzioni: Istruzioni di tipo J

A.A. 2014-2015 2/44 http:\\borgese.di.unimi.it\



## Tipi di istruzioni



```

for (i=0; i<N; i++)           // Istruzioni di controllo
{
  elem = i*N + j;           // Istruzioni aritmetico-logiche
  s = v[elem];             // Istruzioni di accesso a memoria
  z[elem] = s;             // Istruzioni di accesso a memoria
}

```

A.A. 2014-2015 3/44 http://borghese.di.unimi.it/



## Alcune istruzioni dell'ISA



- Aritmetico logiche:
  - add
  - addi
  - addu
  - sub
  - mult
  - or
  - and
  - slt
  - ....
- Accesso alla memoria:
  - lw
  - sw
  - lb
  - lh

A.A. 2014-2015 4/44 http://borghese.di.unimi.it/

## I registri del register file

Nome	Numero	Utilizzo
\$zero	0	costante zero
\$at	1	riservato per l'assemblatore
\$v0-\$v1	2-3	valori di ritorno di una procedura
\$a0-\$a3	4-7	argomenti di una procedura
\$t0-\$t7	8-15	registri temporanei (non salvati)
\$s0-\$s7	16-23	registri salvati
\$t8-\$t9	24-25	registri temporanei (non salvati)
\$k0-\$k1	26-27	gestione delle eccezioni
\$gp	28	puntatore alla global area (dati)
\$sp	29	stack pointer
\$s8	30	registro salvato (fp)
\$ra	31	indirizzo di ritorno

A.A. 2014-2015 5/57 <http://borghese.di.unimi.it/>

## Istruzioni di salto condizionato

- Salti condizionati relativi:
  - **beq** *r1*, *r2*, *L1* (*branch on equal*)
  - **bne** *r1*, *r2*, *L1* (*branch on not equal*)
- Salti condizionati assoluti:
  - Il flusso sequenziale di controllo cambia solo se la condizione è vera. (*beq*)
  - Il calcolo del valore dell'etichetta **L1 (indirizzo di destinazione del salto)** avviene a partire dal Program Counter (PC).
  - Indirizzamento del tipo Base (PC) + Spiazzamento

A.A. 2014-2015 6/44 <http://borghese.di.unimi.it/>

## I salti incondizionati

Salti incondizionati assoluti (j, jal...) – j Etichetta    Esempio: j 80000  
 Il salto viene sempre eseguito.  
 L'indirizzo di destinazione del salto è un indirizzo assoluto di memoria.  
 L'indirizzo di destinazione del salto è un numero sempre positivo.

2 Gbyte

256Mbyte

4Mbyte

0

Stack

↓

↑

Dati Dinamici

Dati Statici

Testo

**Riservata S.O.**

} Segmento dati

} Segmento testo

A.A. 2014-2015
7/44
<http://borghese.di.unimi.it/>

## Introduzione alla CPU

- Istruzioni di salto dell'ISA MIPS
- **I diversi tipi di istruzioni: Istruzioni di tipo R**
- I diversi tipi di istruzioni: Istruzioni di tipo I
- I diversi tipi di istruzioni: Istruzioni di tipo J

A.A. 2014-2015
8/44
<http://borghese.di.unimi.it/>




## Formato delle istruzioni di tipo R

Contiene:

- Un codice operativo su 6 bit
- Un registro source, rs, su 5 bit
- Un registro target, rt, su 5 bit
- Un registro destinazione, rd, su 5 bit
- Un numero di posizioni di shift (shift amount, shamt), su 5 bit
- Un codice di funzione (cf. selettore ALU), su 6 bit

6-bit	5-bit	5-bit	5-bit	5-bit	6-bit	
R	op	rs	rt	rd	shamt	funct

A.A. 2014-2015

9/44

<http://borghese.di.unimi.it/>




## Istruzioni di tipo R: esempio

`add $t0, $s1, $s2`

0	17	18	8	0	32
---	----	----	---	---	----

↓

000000	10001	10010	01000	00000	100000
--------	-------	-------	-------	-------	--------

**0x02324020**

A.A. 2014-2015

10/44

<http://borghese.di.unimi.it/>



## Istruzioni di tipo R: esempi



Nome campo	op	rs	rt	rd	shamt	funct
Dimensione	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
<b>sub \$t0, \$s1, \$s2</b>	000000	10001	10010	01000	00000	100010

Nome campo	op	rs	rt	rd	shamt	funct
Dimensione	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
<b>and \$s1, \$s2, \$s3</b>	000000	10010	10011	10001	00000	100100

Nome campo	op	rs	rt	rd	shamt	funct
Dimensione	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
<b>sll \$s1, \$s2, 3</b>	000000	X	10010	10001	00011	000000
<i>s1 = s2*2<sup>3</sup> Se s2 contiene 20 (0000....0010100) =&gt; s1 conterrà = 160 (0000....0010100000)</i>						

Nome campo	op	rs	rt	rd	shamt	funct
Dimensione	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
<b>srl \$s1, \$s2, 6</b>	000000	X	10010	10001	00110	000010
<i>s1 = s2*2<sup>-6</sup></i>						

A.A. 2014-2015
11/44
<http://borgese.di.unimi.it/>



## Altre istruzioni di tipo R



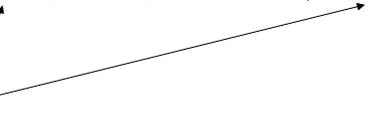
  

Funzioni di salto indiretto (ad esempio per accedere alla zona di memoria istruzioni superiore ai 2GByte)

0	rs	0	0	0	8
---	----	---	---	---	---



**jr rs**



(jump register con **formato R**)

- Salta all'indirizzo di memoria **assoluto** contenuto nel registro **rs** (spazio di  $2^{32}$  Word cioè  $2^{34}$  byte = 8 Gbyte > intero spazio di memoria)

A.A. 2014
<http://borgese.di.unimi.it/>




## Introduzione alla CPU

- Istruzioni di salto dell'ISA MIPS
- I diversi tipi di istruzioni: Istruzioni di tipo R
- **I diversi tipi di istruzioni: Istruzioni di tipo I**
- I diversi tipi di istruzioni: Istruzioni di tipo J

A.A. 2014-2015 13/44 <http://borgese.di.unimi.it/>




## Formato istruzioni di tipo I

o	rs	rt	costante
p 6 bit	5 bit	5 bit	16 bit

- In questo caso, i campi hanno il seguente significato:
  - **op** identifica il tipo di istruzione;
  - **rs** indica il registro sorgente. Nel caso di una lw contiene il registro base;
  - **rt** indica il registro target. Nel caso di una lw, contiene il registro destinazione dell'istruzione di caricamento;
  - **costante**. Nel caso di una lw riporta lo spiazzamento (offset).

A.A. 2014-2015 14/44 <http://borgese.di.unimi.it/>

## Istruzioni di tipo I: accesso a memoria

Con questo formato una istruzione **lw** (**sw**) può indirizzare byte nell'intervallo  $-2^{15}$  ( $-32K$ ) +  $+2^{15}-1$  ( $32K-1$ ) rispetto all'indirizzo base:  $\text{indirizzo} = \text{indirizzo\_base} + \text{offset}$ .

**lw \$t0, 32(\$s3)**

0x8E680020

0

A.A. 2014-2015 15/44 <http://borghese.di.unimi.it/>

## Istruzioni di tipo I: accesso memoria

Nome campo	op	rs	rt	indirizzo
Dimensione	6-bit	5-bit	5-bit	16-bit
<b>lw \$t0, 32 (\$s3)</b>	100011	10011	01000	0000 0000 0010 0000

Nome campo	op	rs	rt	indirizzo
Dimensione	6-bit	5-bit	5-bit	16-bit
<b>sw \$t0, 32 (\$s3)</b>	101011	10011	01000	0000 0000 0010 0000

A.A. 2014-2015 16/44 <http://borghese.di.unimi.it/>

 **Versione I di istruzioni aritmetico-logiche** 

Nome campo	op	rs	rt	costante			
Dimensione	6-bit	5-bit	5-bit	16-bit			
<code>addi \$t0, \$s3, 64</code>	001000	10011	01000	0000	0000	0100	0000

Nome campo	op	rs	rt	costante			
Dimensione	6-bit	5-bit	5-bit	16-bit			
<code>addi \$s1, \$s2, 4</code>	001000	10010	10001	0000	0000	0000	0100

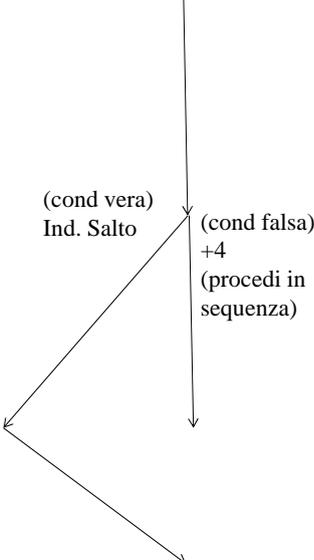
Nome campo	op	rs	rt	costante			
Dimensione	6-bit	5-bit	5-bit	16-bit			
<code>slti \$t0, \$s2, 8</code>	001010	10010	01000	0000	0000	0000	1000

# \$t0 = 1 if \$s2 < 8

A.A. 2014-2015 17/44 http://borghese.di.unimi.it/

 **Istruzioni di salto condizionato** 

- Salti condizionati relativi:
  - `beq r1, r2, L1` (*branch on equal*)
  - `bne r1, r2, L1` (*branch on not equal*)
- Salti condizionati relativi:
  - Il flusso sequenziale di controllo cambia solo se la condizione è vera (`beq`)
  - Il calcolo del valore dell'etichetta **L1 (indirizzo di destinazione del salto)** avviene a partire dal Program Counter (PC).
  - Indirizzamento del tipo Base (PC) + Spiazzamento.



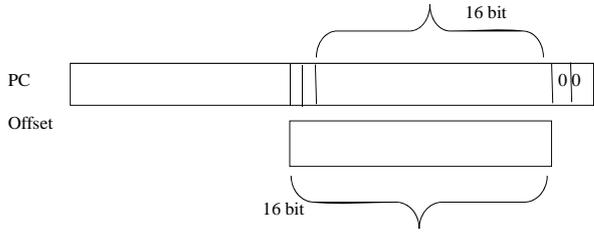
A.A. 2014-2015 18/44 http://borghese.di.unimi.it/




## Allargamento dello spazio di indirizzamento relativo

0000	0	0
0100	1	4
1000	2	8
1100	3	12

Calcolo lo spaziammento in numero di parole invece che di Byte.  
 Considero 64Mword (64M istruzioni) invece di 64Kbyte. Lo spazio indirizzabile all'interno del segmento di testo è di 64Kword \* 4 = 256Kbyte.  
 Moltiplicare per 4 vuol dire operare uno shift a sinistra di due posizioni dell'offset



**La costante su 16 bit rappresenta l'offset in termini di numero di istruzioni**

A.A. 2014-2015
19/44
<http://borghese.di.unimi.it/>




## Calcolo dell'indirizzo di salto

	0x400	addi \$t1, \$zero, 10	# N=10
for (i=0; i++; i<10)	0x404	addi \$t0,\$zero,0	# i =0;
{ ....	0x408	loop: addi \$t0, \$t0,1	# i++
	0x40C	....	
	0x420	bne \$t0, \$t1, loop	
}			

Quanto vale il campo costante da inserire nella bne?

A.A. 2014-2015
20/44
<http://borghese.di.unimi.it/>




## Calcolo dell'indirizzo di salto

```

0x400      addi $t1, $zero, 10 # N=10
0x404      addi $t0,$zero,0   # i =0;
0x408  loop:  addi $t0, $t0,1   # i++
0x40C      ....

0x420      bne $t0, $t1, loop
  
```

L'indirizzo di destinazione è 0x408 (indirizzo dell'etichetta loop)

Lo spiazzamento del salto in byte è pari a:  $(0x408 - 0x424) = -28$  Byte  
 Lo spiazzamento del salto in numero di istruzioni è pari a  $-7$  istruzioni

Prova: Indirizzo di salto = Indirizzo PC+4 + Offset (#istruzioni) \* 4  
 Loop = 0x408 = 0x424 + -7 \* 4

5	8	9	-7 = 1111 1111 1111 1001
---	---	---	--------------------------

A.A. 2014-2015
21/44
<http://borghese.di.unimi.it/>




## Istruzioni di tipo I - Branch

Nome campo	op	rs	rt	indirizzo
Dimensione	6-bit	5-bit	5-bit	16-bit
<code>beq \$s1, \$s2, L1</code>	000100	10001	10010	0000 0000 0001 1001

L1 = PC+4 + 100 byte    Codifica su 18 bit: (00)000 0000 0001 1001(00) in binario.

Nome campo	op	rs	rt	indirizzo
Dimensione	6-bit	5-bit	5-bit	16-bit
<code>beq \$s1, \$s2, L1</code>	000100	10001	10010	1111 1111 1110 0111

L1 = PC+4 -100 byte    Codifica su 18 bit: (11)111 1111 1110 0111(00) in binario.

A.A. 2014-2015
22/44
<http://borghese.di.unimi.it/>




## Formato R ed operazioni logico-matematiche

Non tutte le operazioni logico-matematiche, sono di tipo R.

Le operazioni logico-matematiche di tipo R hanno codice operativo 0.

Non tutte le operazioni con codice operativo 0 sono logico-matematiche (ad esempio ci sono le istruzioni di *jr*, *syscall*...).

Occorre distinguere il funzionamento dell'istruzione elementare dalla sua codifica.

- Codifiche simili (e.g. Tipo R) possono essere condivise da istruzioni di tipo diverso (e.g. aritmetico-logiche, salto).
- Codifiche diverse (e.g. Tipo I e Tipo R) possono essere condivise da istruzioni dello stesso tipo (e.g. add ed addi)

Non c'è corrispondenza 1 a 1, tra tipi strutturali e tipi funzionali.

A.A. 2014-2015 23/44 <http://borghese.di.unimi.it/>




## Introduzione alla CPU

- Istruzioni di salto dell'ISA MIPS
- I diversi tipi di istruzioni: Istruzioni di tipo R
- I diversi tipi di istruzioni: Istruzioni di tipo I
- **I diversi tipi di istruzioni: Istruzioni di tipo J**

A.A. 2014-2015 24/44 <http://borghese.di.unimi.it/>

## Formato istruzioni di tipo J

- E' il formato usato per le istruzioni di salto incondizionato (*jump*):

op	indirizzo
6 bit	26 bit

- In questo caso, i campi hanno il seguente significato:
  - **op** indica il tipo di operazione;
  - **indirizzo** (composto da **26-bit**) riporta una parte (26 bit su 32) dell'indirizzo **assoluto** di destinazione del salto.
- I 26-bit del campo **indirizzo** rappresentano un indirizzo di parola (**word address**)  $2^{26}$  parole.

PC

		00
4 bit (invariati)	26 bit	2 bit

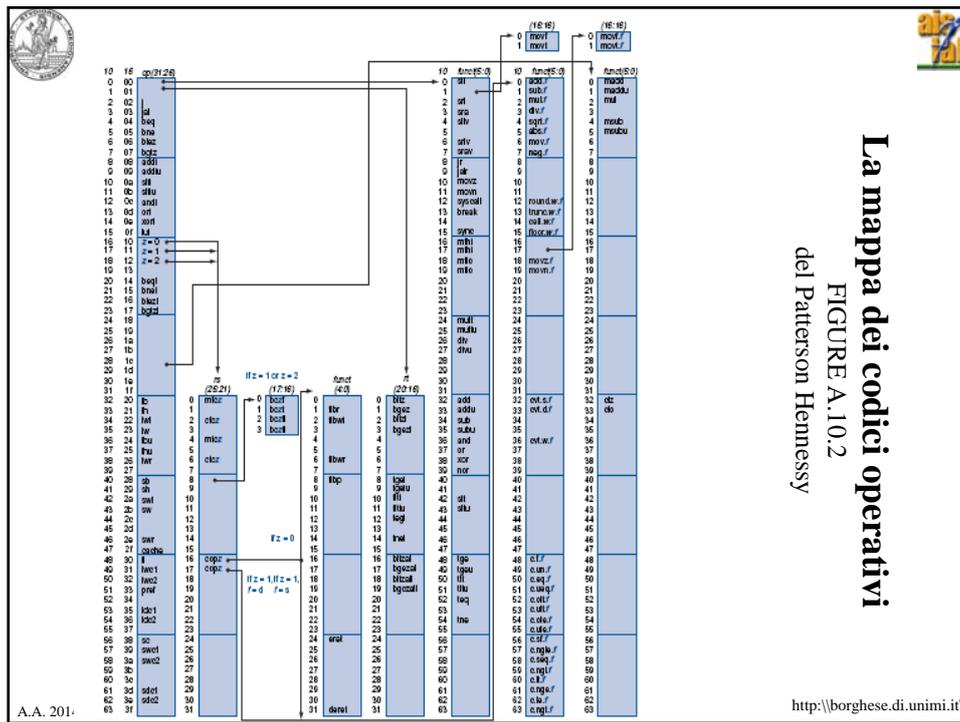
A.A. 2014-2015 25/44 http://borghese.di.unimi.it/

## Codifica delle istruzioni

- Tutte le istruzioni MIPS hanno la **stessa** dimensione (**32 bit**) – **Architettura RISC**.
- I 32 bit hanno un significato diverso a seconda del formato (o tipo) di istruzione
  - il tipo di istruzione è riconosciuto in base al valore di alcuni bit (**6 bit**) più significativi (**codice operativo - OPCODE**)
- Le istruzioni MIPS sono di **3** tipi (formati):
  - **Tipo R (register)** – **Lavorano su 3 registri.**
    - Istruzioni aritmetico-logiche.
  - **Tipo I (immediate)** – **Lavorano su 2 registri. L'istruzione è suddivisa in un gruppo di 16 bit contenenti informazioni + 16 bit riservati ad una costante.**
    - Istruzioni di accesso alla memoria o operazioni contenenti delle costanti.
  - **Tipo J (jump)** – **Lavora senza registri: codice operativo + indirizzo di salto.**
    - Istruzioni di salto incondizionato.

	6-bit	5-bit	5-bit	5-bit	5-bit	6-bit
R	op	rs	rt	rd	shamt	funct
I	op	rs	rt	Indirizzo / costante		
J	op	Indirizzo / costante				

A.A. 2014-2015 26/44 http://borghese.di.unimi.it/



## Introduzione alla CPU

- Istruzioni di salto dell'ISA MIPS
- I diversi tipi di istruzioni: Istruzioni di tipo R
- I diversi tipi di istruzioni: Istruzioni di tipo I
- I diversi tipi di istruzioni: Istruzioni di tipo J

A.A. 2014-2015 28/44 <http://borgnese.di.unimi.it/>