5' Esercitazione

https://politecnicomilano.webex.com/meet/gianenrico.conti

Gian Enrico Conti

MIPS architecture OPCODES

Outline

- **Formato Istruzioni**
  - Istruzioni per formato

Formato istruzioni

- **Esistono 3 tipi di formati istruzioni nel processore MIPS**
  - Istruzioni R
  - Istruzioni I
  - Istruzioni J

Nome	Campi						Commenti
Dimensione del campo	6 bit	5 bit	5 bit	5 bit	5 bit	6 bit	Tutte le istruzioni MIPS sono a 32 bit
Formato R	codop	rs	rt	rd	shamt	funz	Formato delle istruzioni aritmetiche
Formato I	codop	rs	rt	indirizzo / costante			Formato delle istruzioni di trasferimento dati di salto condizionato e immediate
Formato J	codop	indirizzo di destinazione					Formato delle istruzioni di salto incondizionato

- **I campi definiti sono**
  - codop — codice operativo
  - rs — primo registro sorgente
  - rt — secondo registro sorgente
  - rd — registro destinazione
  - shamt — shift amount
  - funz — codice funzione
  - costante — valore costante
  - indirizzo — indirizzo di destinazione

Istruzioni per formato

Istruzioni MIPS	Nome	Formato
add	add	R
subtract	sub	R
add immediate	addi	I
load word	lw	I
store word	sw	I
load half	lh	I
load half unsigned	lhu	I
store half	sh	I
load byte	lb	I
load byte unsigned	lbu	I
store byte	sb	I
load linked	ll	I
store conditional	Sc	I
load upper immediate	lui	I
and	and	R
or	or	R
nor	nor	R
and immediate	andi	I
or immediate	ori	I
shift left logical	sll	R
shift right logical	srl	R
branch equal	beq	I
branch not equal	bne	I
set less than	slt	R
set less than immediate	slti	I
set less than immediate unsigned	sltiu	I
jump	j	J
jump register	jr	R
jump and link	jal	J

Pseudoistruzioni MIPS	Nome	Formato
move	move	R
multiply	mult	R
multiply immediate	multi	I
load immediate	li	I
branch less than	blt	I
branch less than or equal	ble	I
branch greater than	bgt	I
branch greater than or equal	bge	I

MIPS Reference Sheet

(https://uweb.engr.arizona.edu/~ece369/Resources/spim/MIPSReference.pdf)

Instruction Encodings

Register	000000ss sssttttt dddddaaa aafffffff
Immediate	oooooooss sssttttt iiiiiiii iiiiiiii
Jump	oooooooii iiiiiiii iiiiiiii iiiiiiii

Opcode Table

Instruction	Opcode/Function	Syntax
add	100000	ArithLog
addu	100001	ArithLog
addi	001000	ArithLogI
addiu	001001	ArithLogI
and	100100	ArithLog
andi	001100	ArithLogI
div	011010	DivMult
divu	011011	DivMult
mult	011000	DivMult
multu	011001	DivMult
nor	100111	ArithLog
or	100101	ArithLog
ori	001101	ArithLogI
sll	000000	Shift
sllv	000100	ShiftV
sra	000011	Shift
srav	000111	ShiftV
srl	000010	Shift
srlv	000110	ShiftV
sub	100010	ArithLog
subu	100011	ArithLog
xor	100110	ArithLog
xori	001110	ArithLogI
lhi	011001	LoadI
llo	011000	LoadI

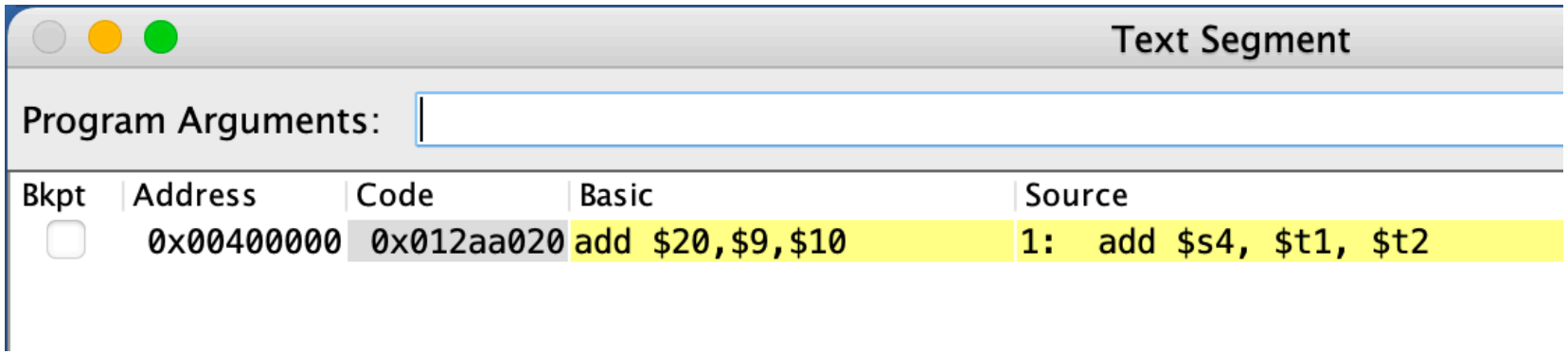
Instruction	Opcode/Function	Syntax
slt	101010	ArithLog
sltu	101001	ArithLog
slti	001010	ArithLogI
sltiu	001001	ArithLogI
beq	000100	Branch
bgtz	000111	BranchZ
blez	000110	BranchZ
bne	000101	Branch
j	000010	Jump
jal	000011	Jump
jalr	001001	JumpR
jr	001000	JumpR
lb	100000	LoadStore
lbu	100100	LoadStore
lh	100001	LoadStore
lhu	100101	LoadStore
lw	100011	LoadStore
sb	101000	LoadStore
sh	101001	LoadStore
sw	101011	LoadStore
mfhi	010000	MoveFrom
mflo	010010	MoveFrom
mthi	010001	MoveTo
mtlo	010011	MoveTo
trap	011010	Trap

Some samples..

EXAMPLE: R-type

add $s4, $t1, $t2

In Mars:



Rd: 20

Rs:9

Rt: 10

Op = 00000

E dalla tabella prec.:

Instruction	Opcode/Function	Syntax
add	100000	ArithLog

EXAMPLE: R-type

add $s4, $t1, $t2

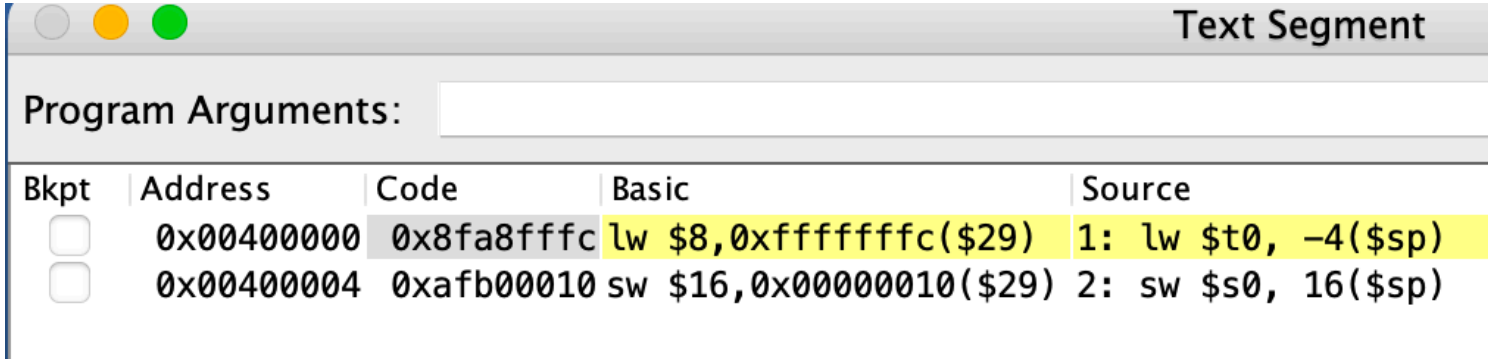

```
add $s4, $t1, $t2
```

```
Rd: 20          10100
Rs:  9          01001
Rt: 10          01010
```

EXAMPLE: lw, sw

```
.text
lw $t0, -4(sp)
sw $s0, 16(sp)
```

In Mars:



Rd: --
Rs: (SP)$29 = 29
Rt: $8
    $10

lw	100011
sb	101000
sh	101001
sw	101011

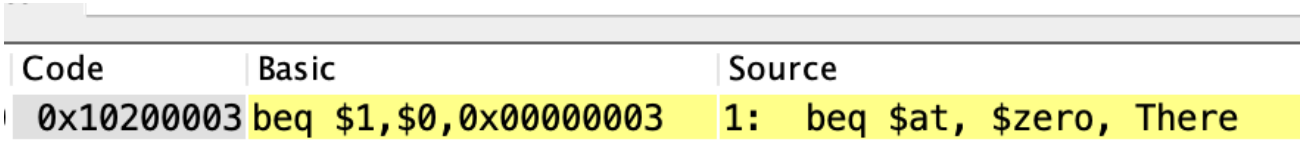
EXAMPLE: salto

beq $at, $zero, There

addi $at, $zero, 10 #dont care
addi $at, $zero, 20 #dont care

There: sub $at, $at,30 #dont care

In Mars:



Code	Basic	Source
0x10200003	beq $1,$0,0x00000003	1: beq $at, $zero, There

Rd: 0
Rs: 1

Si noti salto di 3 (3 istruz. = 12 celle)

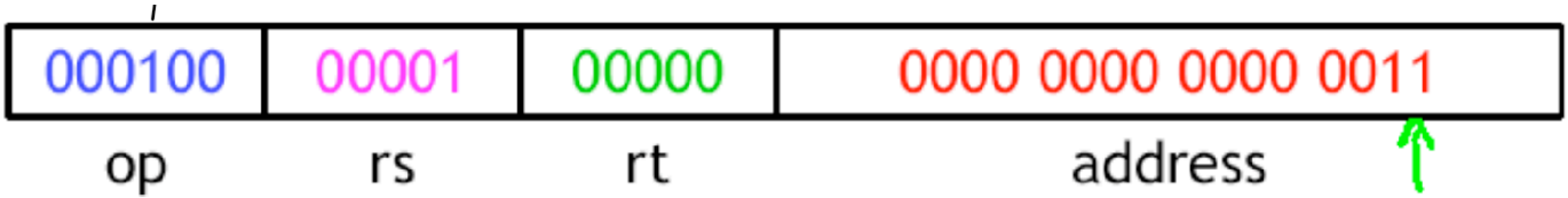
EXAMPLE: lw, sw

```
beq $at, $zero, There
li $t1,10
li $t2,20
add $t1, $t1, $t2
There: sub $at, $at,30
```

sltiu	001001	ArithLogI
beq	000100	Branch
bgtz	000111	BranchZ
blez	000110	BranchZ

Rd: 0

Rs: 1



Offset= 3x4=12

How to...

- https://www.eg.bucknell.edu/~csci320/mips_web/
-