

Aritmetičke i logičke instrukcije

add Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x20	Rd=Rs+Rt
Sumu registara Rs i Rt smešta u registar Rd.							
addu Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x21	Rd=Rs+Rt
Sumu registara Rs i Rt smešta u registar Rd. Bez prekoračenja.							
addi Rt, Rs, Imm	8	Rs	Rt	Imm			Rt=Rs+signex(Imm)
Sumu registra Rs i sign-extended konstante smešta u registar Rt.							
addiu Rt, Rs, Imm	9	Rs	Rt	Imm			Rt=Rs+signex(Imm)
Sumu registra Rs i sign-extended konstante smešta u registar Rt. Bez prekoračenja.							
and Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x24	Rd=Rs AND Rt
Logičko I između registara Rs i Rt smešta u Rd.							
andi Rt, Rs, Imm	0xC	Rs	Rt	Imm			Rt=Rs+zeroex(Imm)
Logičko I između registra Rs i konstante proširene nulama smešta u registar Rt.							
nor Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x27	Rd=Rs NOR Rt
Logičko NILI između registara Rs i Rt smešta u Rd.							
or Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x25	Rd=Rs OR Rt
Logičko ILI između registara Rs i Rt smešta u Rd.							
ori Rt, Rs, Imm	0xD	Rs	Rt	Imm			Rt=Rs+zeroex(Imm)
Logičko ILI između registra Rs i konstante proširene nulama smešta u Rt.							
sll Rd, Rt, Sa	0	Rs	Rt	Rd	Sa	0	logičko šiftovanje ulevo
Pomeranje sadržaja registra Rt ulevo za iznos dat u konstanti Sa i smeštanje rezultata u Rd.							
sllv Rd, Rt, Rs	0	Rs	Rt	Rd	0	4	logičko šiftovanje ulevo
Pomeranje sadržaja registra Rt ulevo za iznos dat u registru Rs i smeštanje rezultata u Rd.							
sra Rd, Rt, Sa	0	Rs	Rt	Rd	Sa	3	aritmetičko šiftovanje udesno
Pomeranje (aritmetičko) sadržaja registra Rt udesno za iznos dat u konstanti Sa i smeštanje rezultata u Rd.							
srav Rd, Rt, Rs	0	Rs	Rt	Rd	0	7	aritmetičko šiftovanje udesno
Pomeranje (aritmetičko) sadržaja registra Rt udesno za iznos dat u registru Rs i smeštanje rezultata u Rd.							
srl Rd, Rt, Sa	0	Rs	Rt	Rd	Sa	2	logičko šiftovanje udesno
Pomeranje (logičko) sadržaja registra Rt udesno za iznos dat u konstanti Sa i smeštanje rezultata u Rd.							
srlv Rd, Rt, Rs	0	Rs	Rt	Rd	0	6	logičko šiftovanje udesno
Pomeranje (logičko) sadržaja registra Rt udesno za iznos dat u registru Rs i smeštanje rezultata u Rd.							
sub Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x22	Rd = Rs - Rt
Razlika registara Rs i Rt smešta se u Rd.							
subu Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x23	Rd = Rs - Rt
Razlika registara Rs i Rt smešta se u Rd. Bez prekoračenja.							
xor Rd, Rs, Rt	0	Rs	Rt	Rd	0	0x26	Rd = Rs XOR Rt
Ekskluzivno ILI između registara Rs i Rt smešta u Rd.							

xori Rt, Rs, Imm

0xE	Rs	Rt	Imm
-----	----	----	-----

 Rt=Rs+zeroex(Imm)
Ekskluzivno ILI između registra Rs i konstante proširene nulama smešta u Rt.

Instrukcije poredjenja

slt Rd, Rs, Rt

0	Rs	Rt	Rd	0	0x2A
---	----	----	----	---	------

 set less than
Upisuje 1 u registar Rd ako je Rs < Rt, a 0 ako nije.

sltu Rd, Rs, Rt

0	Rs	Rt	Rd	0	0x2B
---	----	----	----	---	------

 set less than unsigned
Upisuje 1 u registar Rd ako je Rs < Rt, a 0 ako nije. Najviši bit ne posmatra kao bit znaka nego kao težinski bit.

slti Rt, Rs, Imm

0xA	Rs	Rt	Imm
-----	----	----	-----

 slt signex(Imm)
Upisuje 1 u registar Rt ako je Rs manje od sign-extended konstante, a 0 ako nije.

sltiu Rt, Rs, Imm

0xB	Rs	Rt	Imm
-----	----	----	-----

 slt signex(Imm) unsigned
Upisuje 1 u registar Rt ako je Rs manje od "sign-extended" konstante, a 0 ako nije. Najviši bit ne posmatra kao bit znaka nego kao težinski bit.

Instrukcije uslovnog i bezuslovnog skoka

beq Rs, Rt, label

4	Rs	Rt	Offset
---	----	----	--------

 branch on equal
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs=Rt

bgez Rs, label

1	Rs	1	Offset
---	----	---	--------

 branch if Rs ≥ 0
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs ≥ 0.

bgtz Rs, label

7	Rs	0	Offset
---	----	---	--------

 branch if Rs > 0
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs > 0.

blez Rs, label

6	Rs	0	Offset
---	----	---	--------

 branch if Rs ≤ 0
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs ≤ 0.

bltz Rs, label

1	Rs	0	Offset
---	----	---	--------

 branch if Rs < 0
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs < 0.

bne Rs, Rt, label

5	Rs	Rt	Offset
---	----	----	--------

 branch if Rs ≠ Rt
Skočiti za onoliko instrukcija koliko pokazuje Offset, ako je Rs ≠ Rt.

j label

2	Target
---	--------

 jump
Skočiti na instrukciju na lokaciji Target.

jal label

3	Target
---	--------

 jump and link
Skočiti na instrukciju na lokaciji Target. Sačuvati adresu sledeće instrukcije u registru 31.

jr Rs

0	Rs	0	8
---	----	---	---

 jump register
Skočiti na instrukciju čija se adresa nalazi u registru Rs.

Instrukcije transfera podataka

lb Rt, address

0x20	Rs	Rt	Offset
------	----	----	--------

 load byte
Učitati bajt sa lokacije Rs+Offset u registar Rt. Bajt je "sign-extended".

lbu Rt, address

0x24	Rs	Rt	Offset
------	----	----	--------

 load unsigned byte
Učitati bajt sa lokacije Rs+Offset u registar Rt. Bajt je proširen nulama.

lh Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x21</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x21	Rs	Rt	Offset	load halfword
0x21	Rs	Rt	Offset			
	Učitati 16 bita (pola reči) sa lokacije Rs+Offset u registar Rt. Tih 16 bita su "sign-extended".					
lhu Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x25</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x25	Rs	Rt	Offset	load unsigned halfword
0x25	Rs	Rt	Offset			
	Učitati 16 bita (pola reči) sa lokacije Rs+Offset u registar Rt. Tih 16 bita su prošireni nulama.					
lw Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x23</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x23	Rs	Rt	Offset	load word
0x23	Rs	Rt	Offset			
	Učitati reč sa lokacije Rs+Offset u registar Rt.					
sb Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x28</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x28	Rs	Rt	Offset	store byte
0x28	Rs	Rt	Offset			
	Smestiti najniži bajt iz registra Rt na lokaciju sa adresom Rs+Offset.					
sh Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x29</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x29	Rs	Rt	Offset	store halfword
0x29	Rs	Rt	Offset			
	Smestiti nižih 16 bita iz registra Rt na lokaciju sa adresom Rs+Offset.					
sw Rt, address	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 100px;">0x2B</td><td style="width: 50px;">Rs</td><td style="width: 50px;">Rt</td><td style="width: 150px;">Offset</td></tr></table>	0x2B	Rs	Rt	Offset	store word
0x2B	Rs	Rt	Offset			
	Smestiti reč iz registra Rt na lokaciju sa adresom Rs+Offset.					