Digitalna vezja
UL, FRI

Vaja 12, MIPS procesor

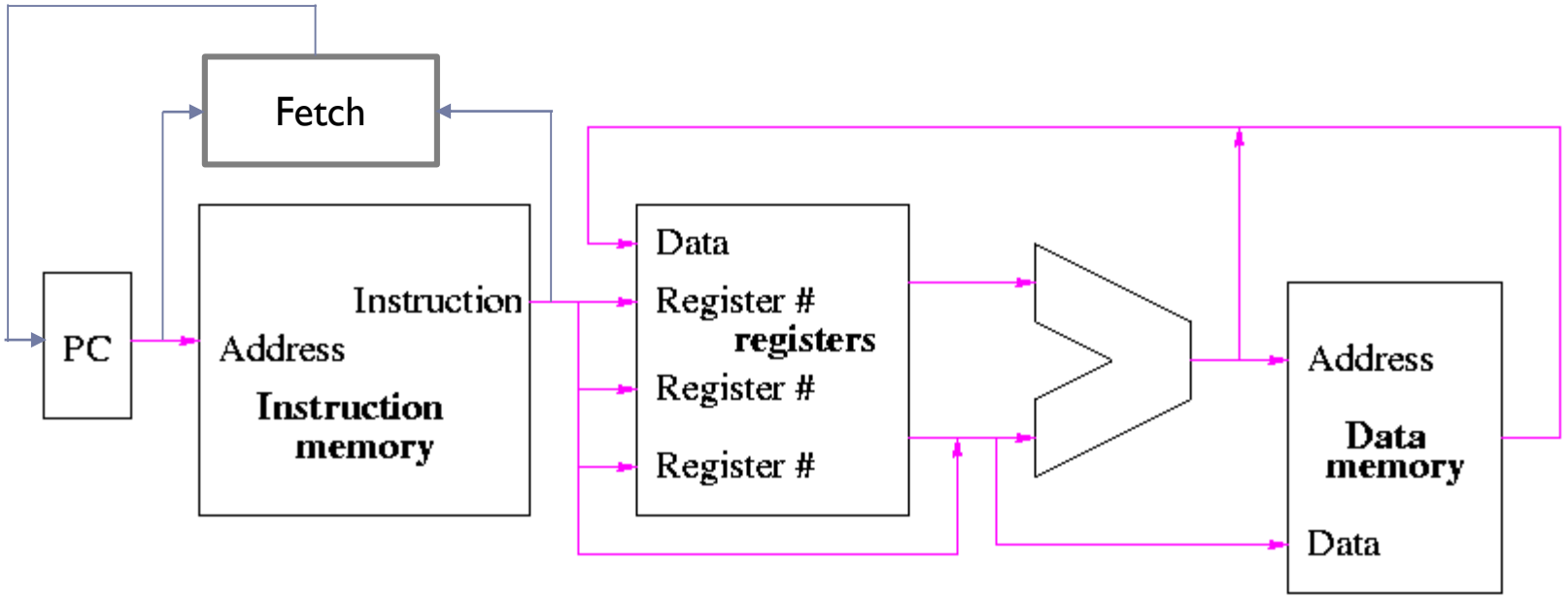
MIPS – podatkovne poti (enojni cikel)

Pomnilnik:

- Ukazni
- Podatkovni

Literatura: Computer Organization and Design, 3rd edition, 2005

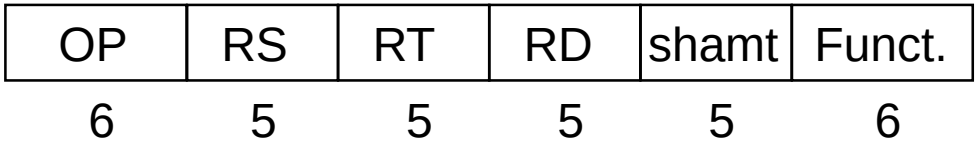
http://dcsuop.blogspot.com/2013/02/computer-organization-and-design-3rd-ed.html



Procesor MIPS: ukazi

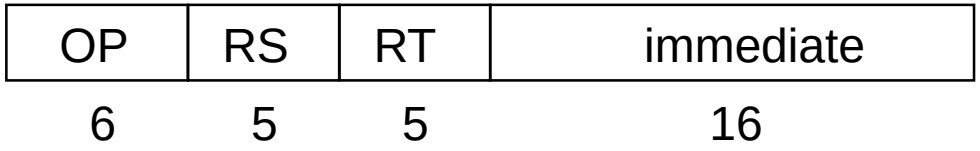
Aritmetično-logični ukazi (R-tip):

Add Unsigned addu R[RD]=R[RS] + R[RT] $0/21_H$



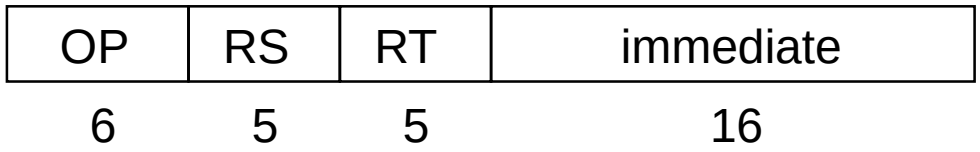
Pomnilni ukazi (I-tip):

Load Word lw R[RT]=M[R[RS]+SgnExtImm] 23_H



Pogojna vejitev (I-tip):

Branch on not equal bne if(R[RS] != R[RT])
PC=PC+4+BrAddr 5_H



RTL jezik ('Register Transfer Language')

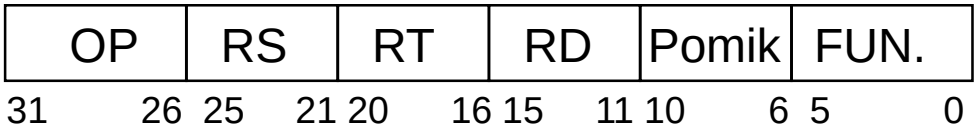
ZAPIS: je določen kot: Dst <= Src ali Src in operacija

Primeri:

- Vpis v samostojen register (IR, PC):
  - IR <= Pom[PC]
  - PC <= PC+ 4
- Branje iz registra in vpis v register (Registrski niz):
  - A(RD1) <= Reg[IR(25-21)] (data out <= address)
  - B(RD2) <= Reg[IR(20-16)]
  - B <= SgnExt[IR(15-0)] (ALE in <= data out)
  - Reg[IR(15-11)] <= ALEizh (data in <= data out)
  - Reg[IR(20-16)] <= Pom[ALEizh]
- Določanje izhoda ALE:
  - ALEizh <= A(RD1)+SgnExt[IR(15-0)]
- Zapis pogojev:
  - if(A-B==0) then PC <= ALEizh

R- tip: Add Unsigned

Ime	Add Unsigned
Mnemonik	addu
Format	R
Operacija	R[rd] = R[rs] + R[rt]
OPCODE/FUNCT	$0/21_H$



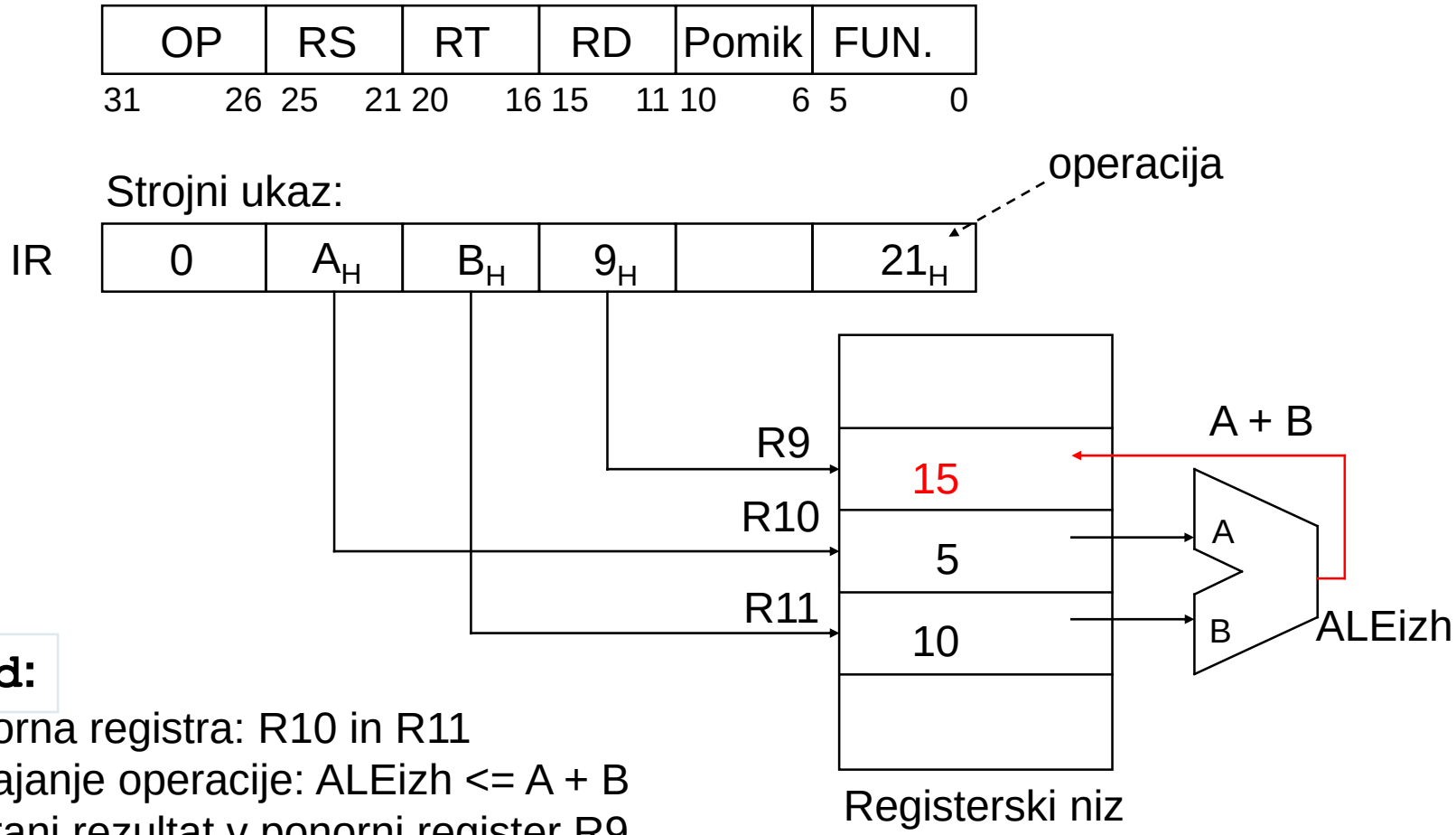
Opis: 0000 00ss ssst tttt dddd d000 0010 0001

Zapis ukaza: OP RD, RS, RT

Zbirnik: addu $t9, $t10, $t11

0000 0001 0010 1011 0101 0000 0010 0001

addu $t9, $t10, $11



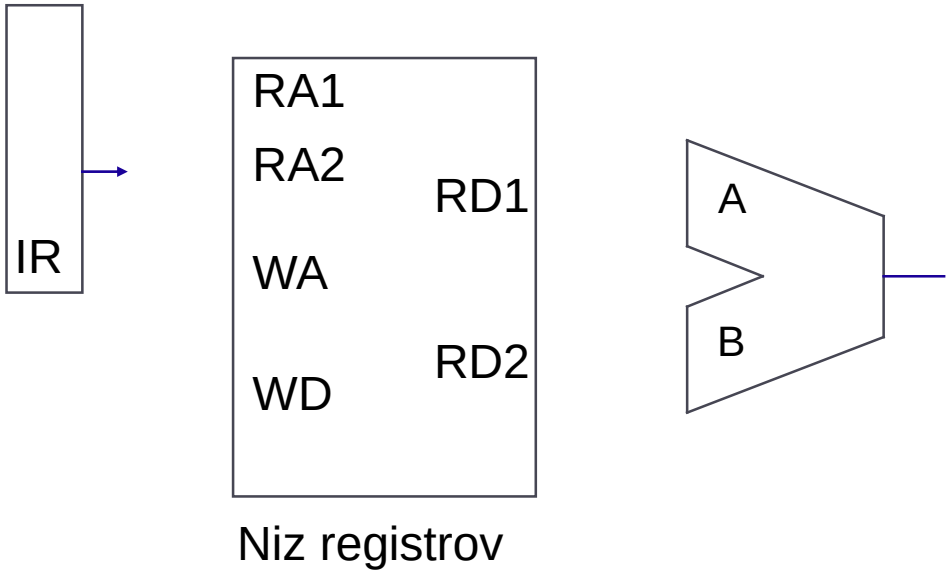
add:

Izvorna registra: R10 in R11

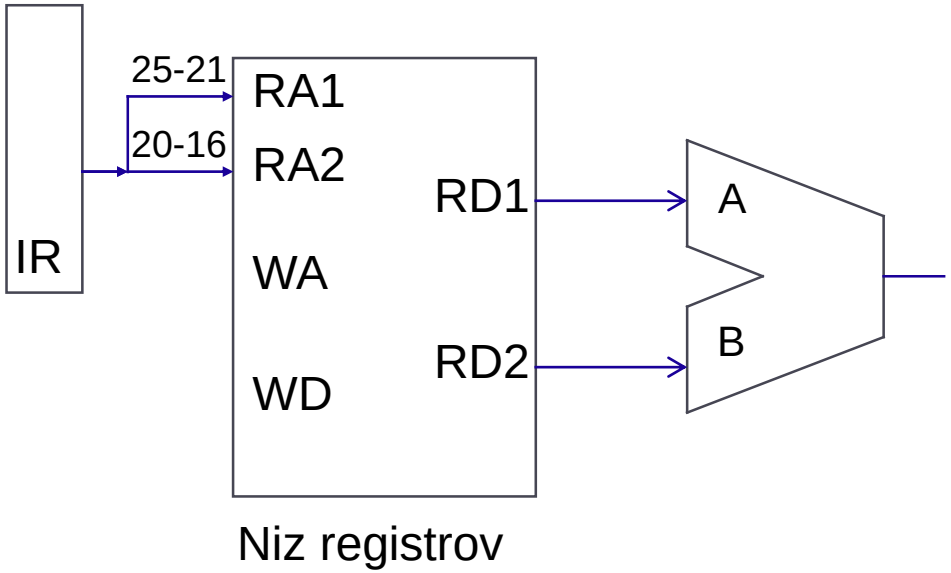
Izvajanje operacije: ALEizh <= A + B

Shrani rezultat v ponorni register R9

Podatkovne poti - gradniki

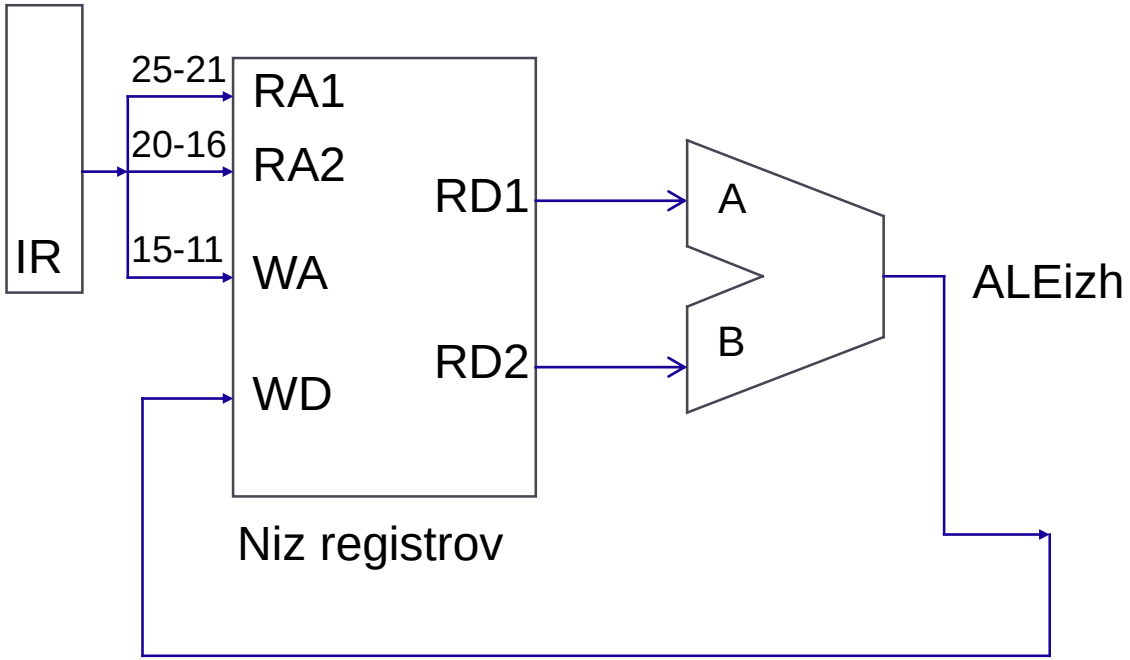


Naslavljanje registrov RS in RT, podatki na ALE



RS: A(RD1) <= Reg[IR(25-21)]
RT: B(RD2) <= Reg[IR(20-16)]

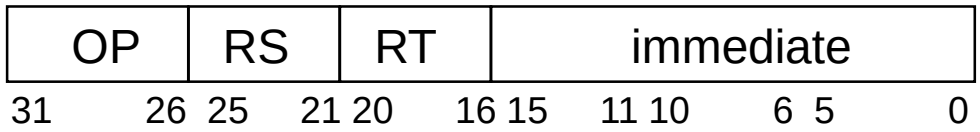
Seštevanje, naslavljanje registra RD in vpis rezultata



RS: A(RD1) <= Reg[IR(25-21)]
RT: B(RD2) <= Reg[IR(20-16)]
ALEizh <= A + B
RD: Reg[IR(15-11)] <= ALEizh

I- tip: Load Word

Ime	Load Word
Mnemonik	ld
Format	I
Operacija	R[rt] = M[R[rs] + SignExtImm]
OPCODE	23_H



Opis: 1000 11ss ssst tttt iiii iiii iiii iiii

SignExtImm: { 16 {immediate[15]} , immediate }

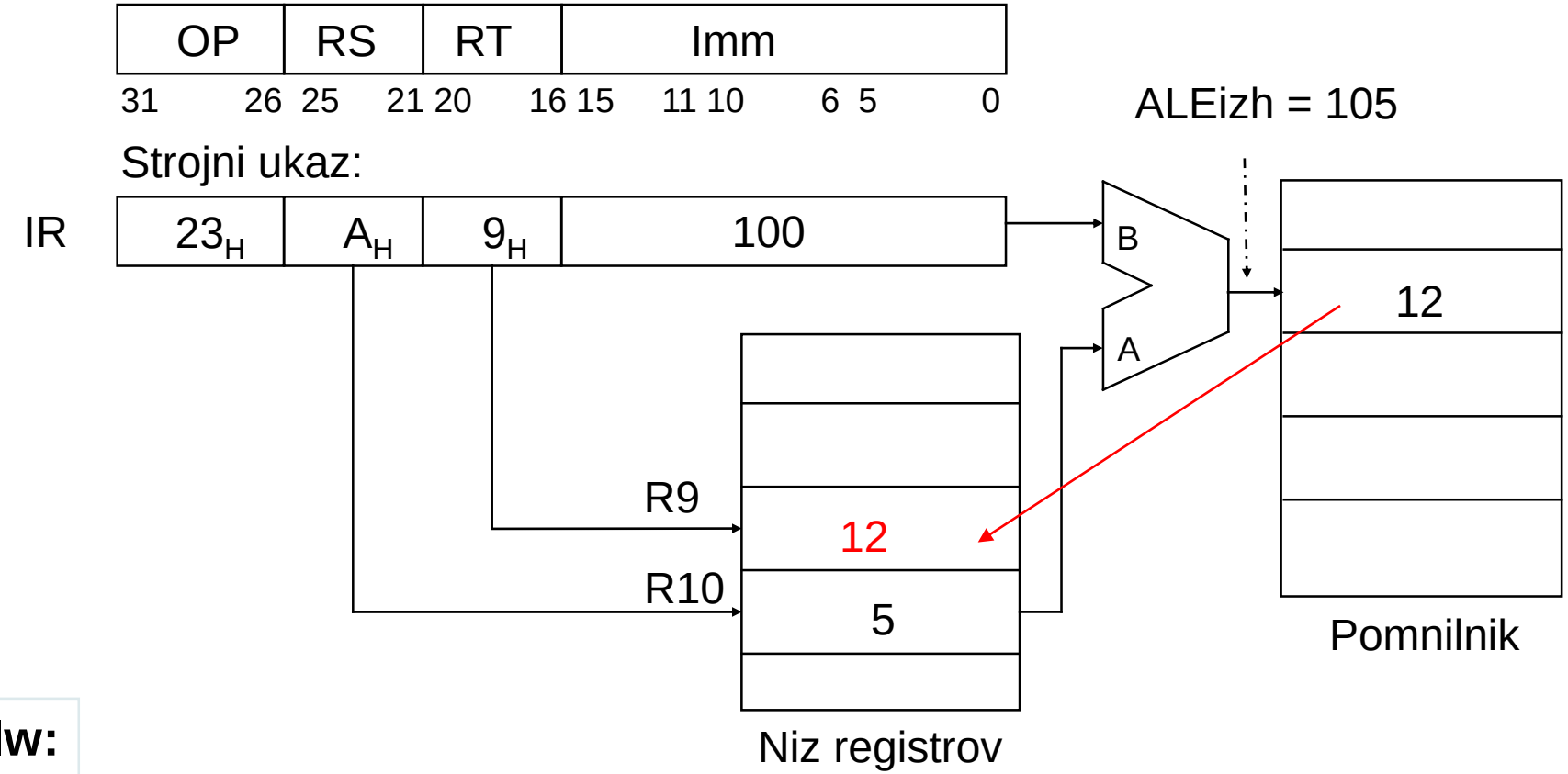
Zapis ukaza: OP rt, Imm(rs)

Zbirnik: lw $t9, 100($t10)

1000 1101 0100 1001 0000 0000 0110 0100

SignExtImm: 0000 0000 0000 0000 0000 0000 0110 0100

lw $t9, 100($t10)



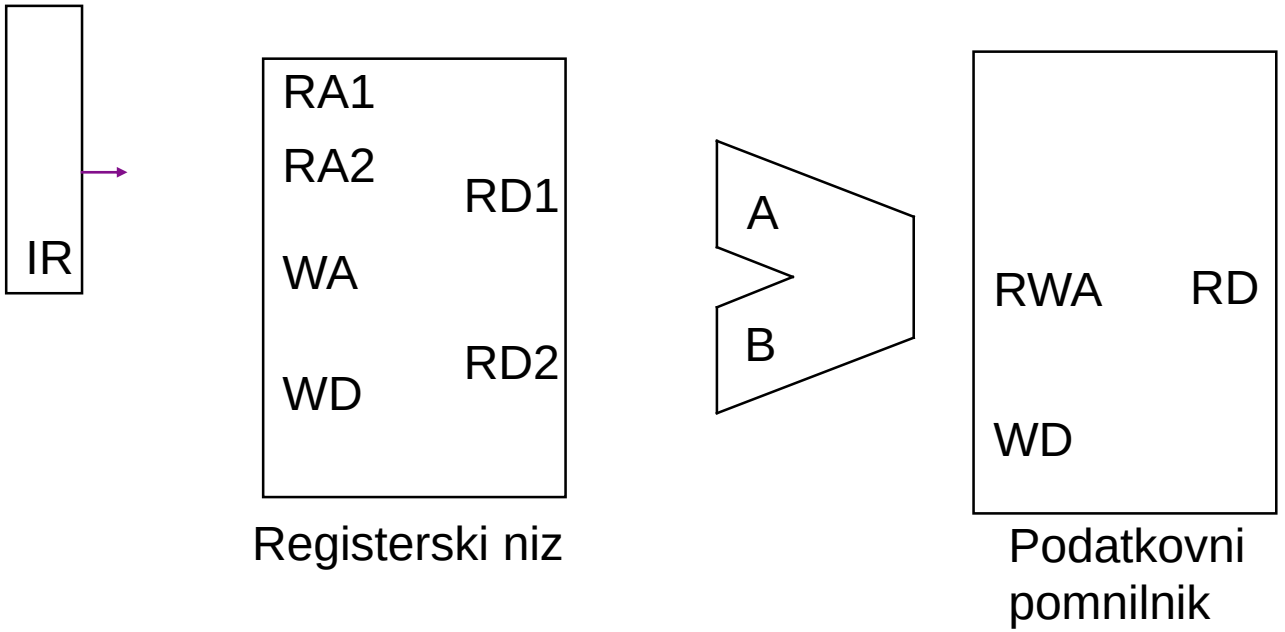
lw:

Izvorni register: R10

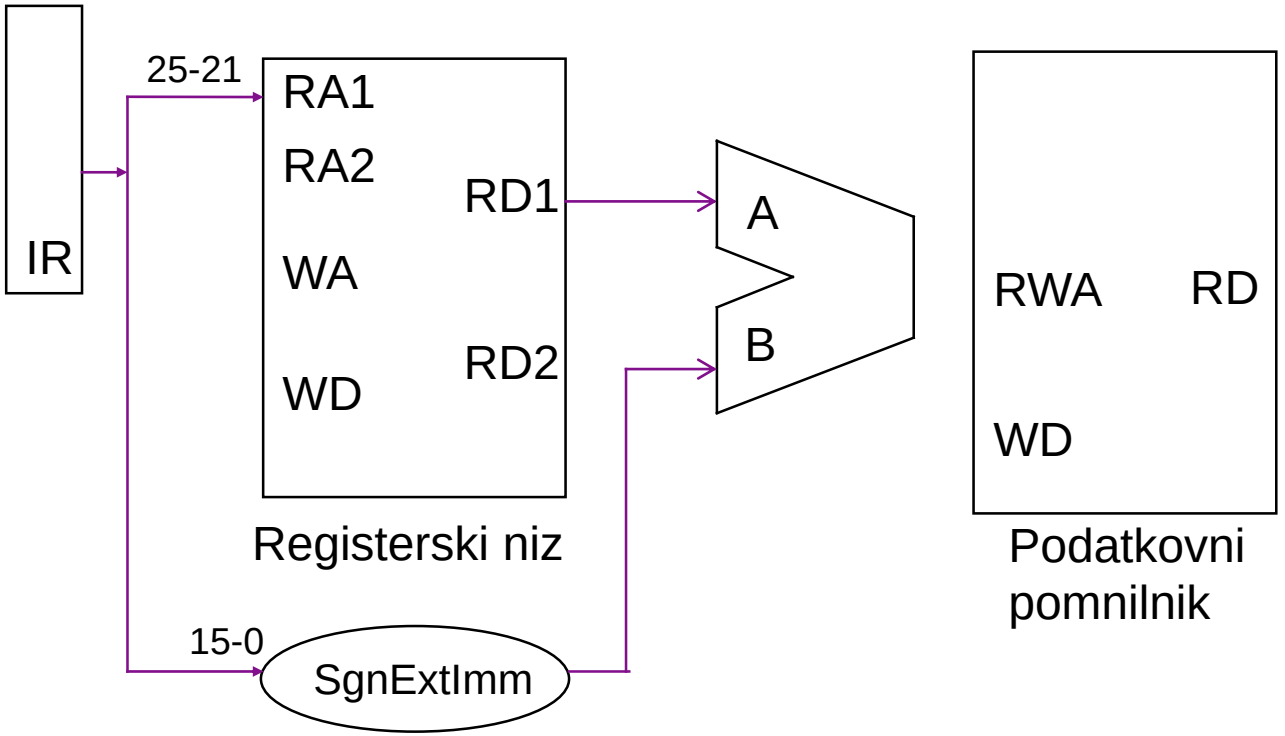
Izračun naslova: ALEizh <= A + B(SgnExt(IR(15-0)))

Branje iz pomnilnika, naslov 105 in vpis podatka v register R9

Podatkovne poti - gradniki

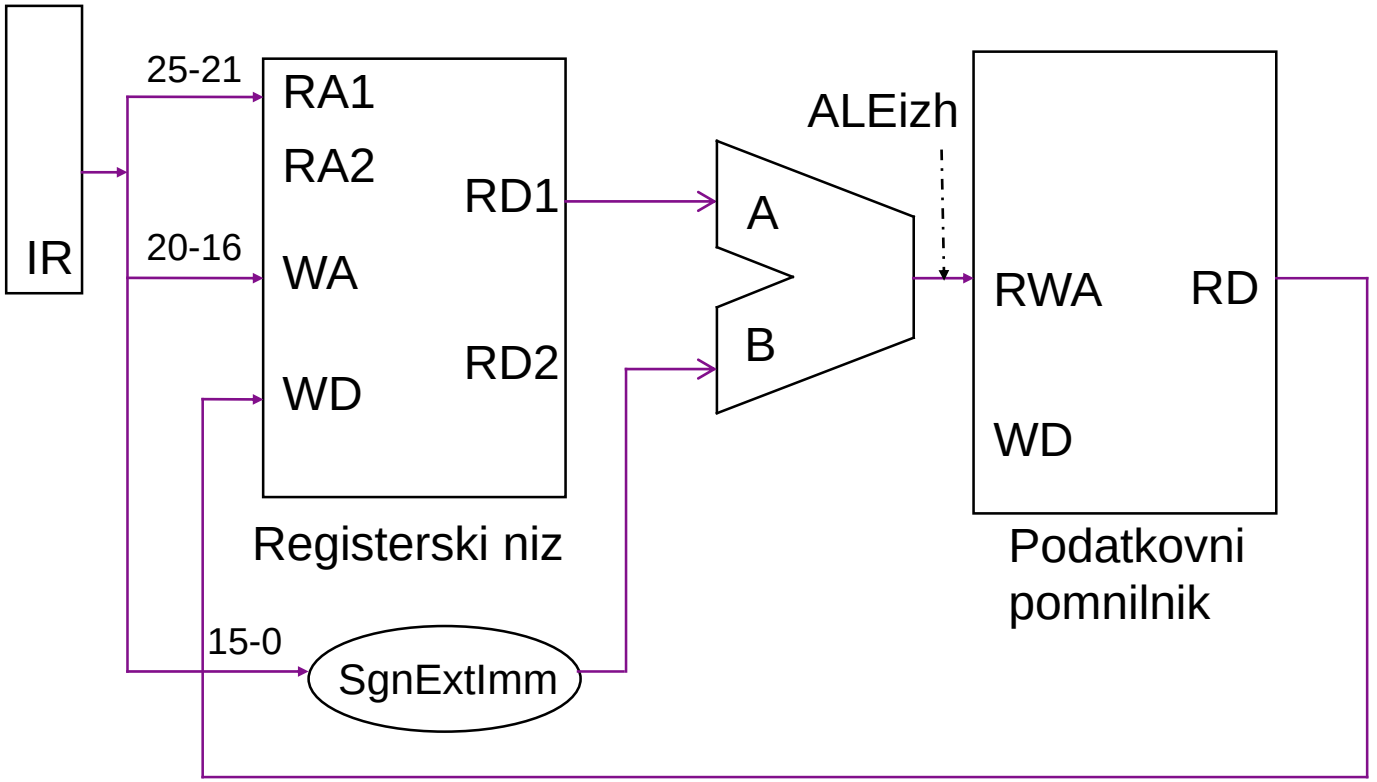


Naslavljanje registra RS, SgnExtImm, podatki na ALE



RS: A(RD1) <= Reg[IR(25-21)]
B(SgnExt) <= IR(15-0)

Naslavljanje registra RT, seštevanje in naslavljanje pomnilnika, podatki v register RT

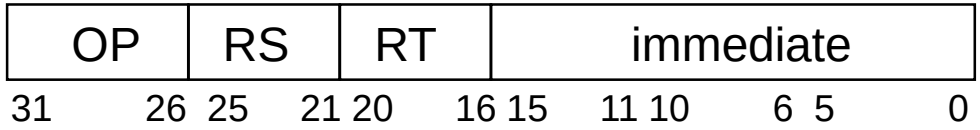


RS: A(RD1) <= Reg[IR(25-21)]
B(SgnExt) <= IR(15-0)
ALEizh = A + B

RT: Reg[IR(20-16)] <= Pom[ALEizh]

I- tip: Branch on not Equal

Ime	Branch on not Equal
Mnemonik	bne
Format	I
Operacija	if (R[rs] != R[rt]), PC = PC + 4 + BrAddr
OPCODE	5_H



Opis: 0001 01ss ssst tttt iiii iiii iiii iiii

BrAddr: { 14 {immediate[15]}, immediate, 00 }

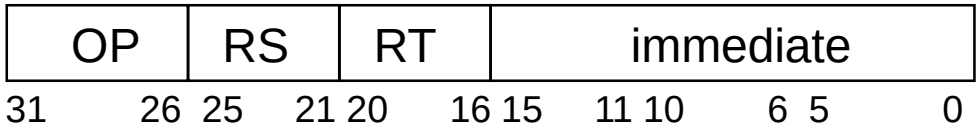
Zapis ukaza: OP rs, rt, Imm

Zbirnik: bne $t10, $t9, 25

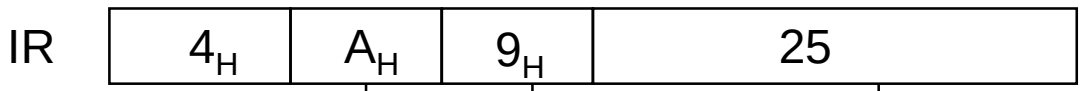
0001 0101 0100 1001 0000 0000 0001 1001

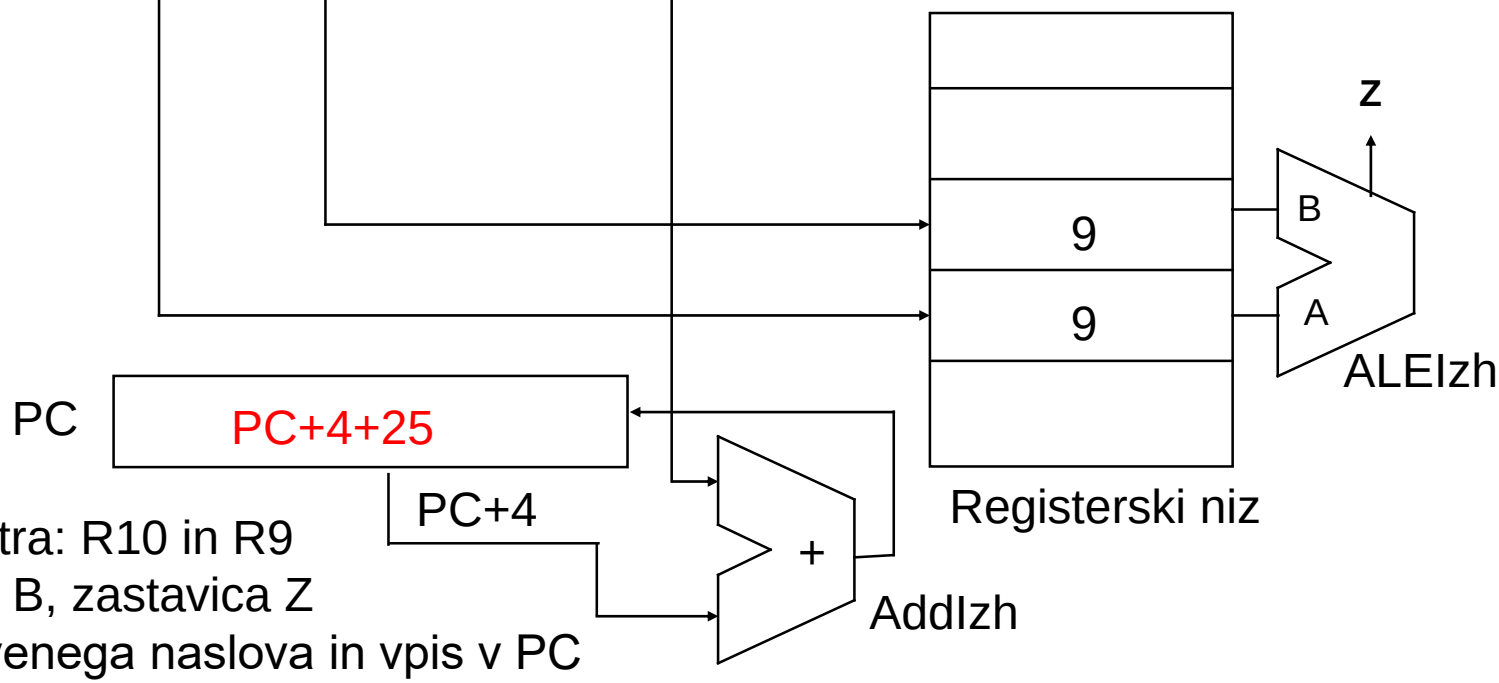
BrAddr: 0000 0000 0000 0000 0000 0000 0001 1001

bne $t10, $t9, 25



Strojni ukaz:

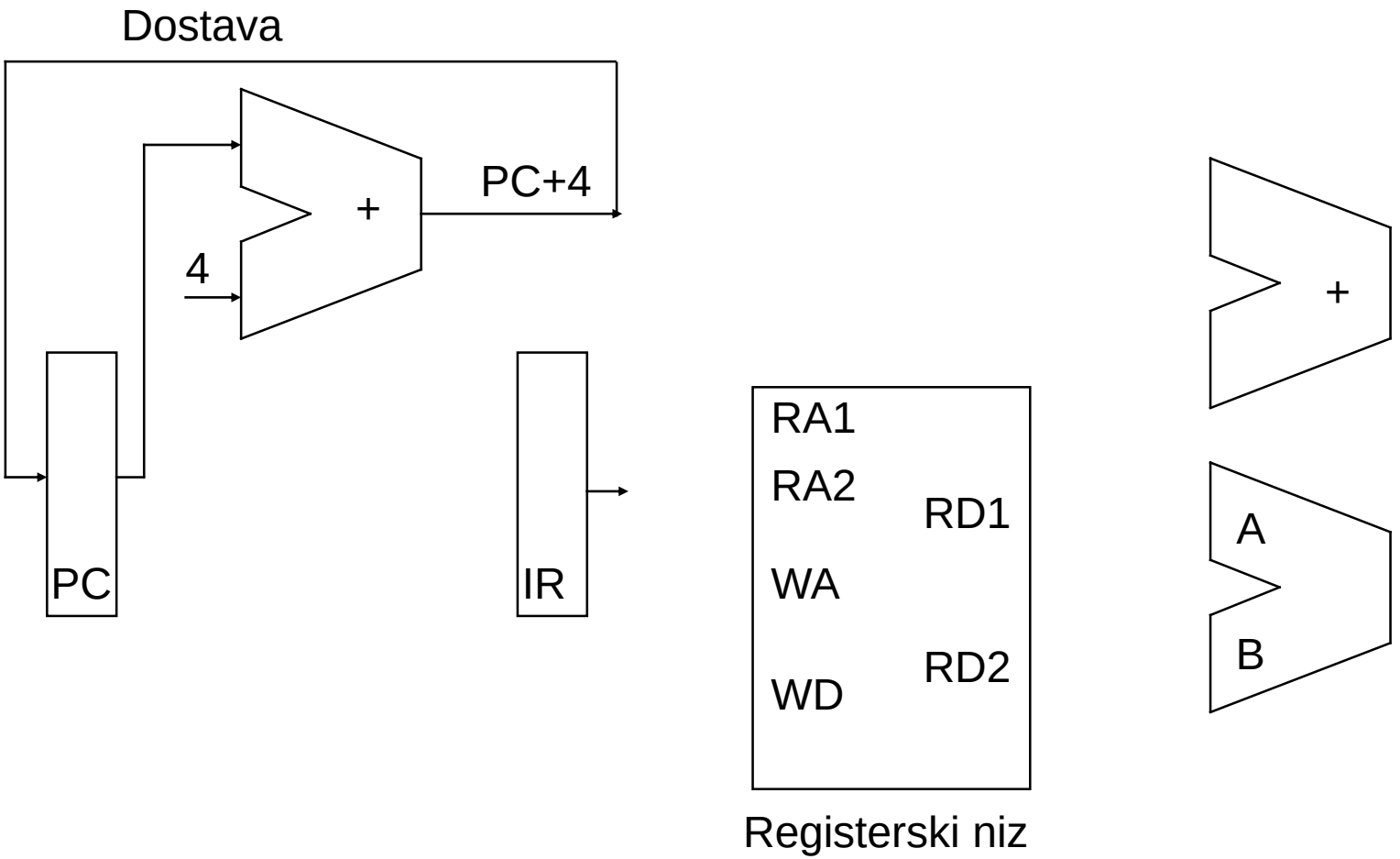




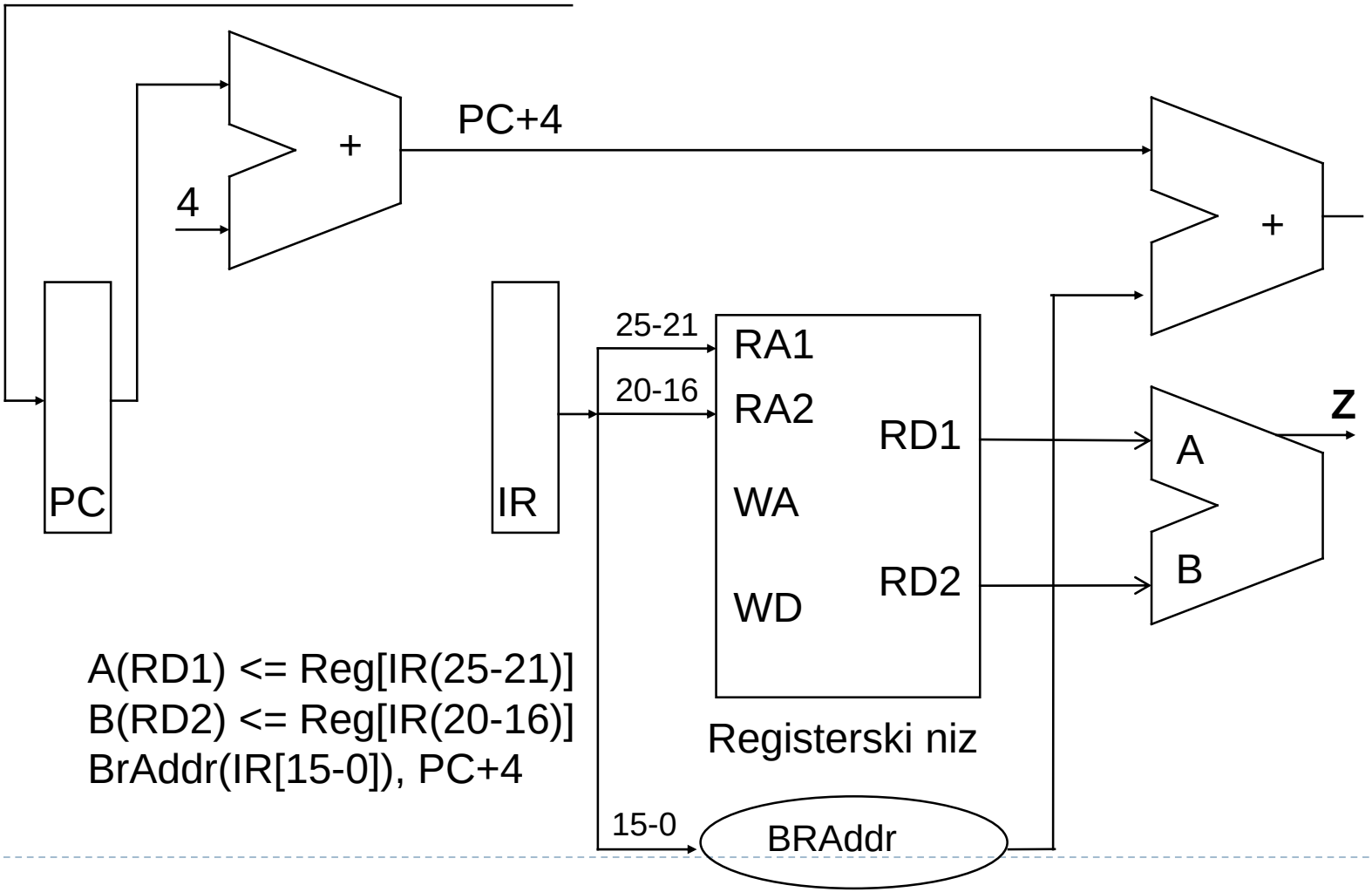
bne:

Izvorna registra: R10 in R9
ALEIzh = A – B, zastavica Z
Izračun vejitvenega naslova in vpis v PC
AddIzh <=PC+4+BrAddr(IR[15-0]); if Z=0 PC <= AddIzh

Podatkovne poti - gradniki

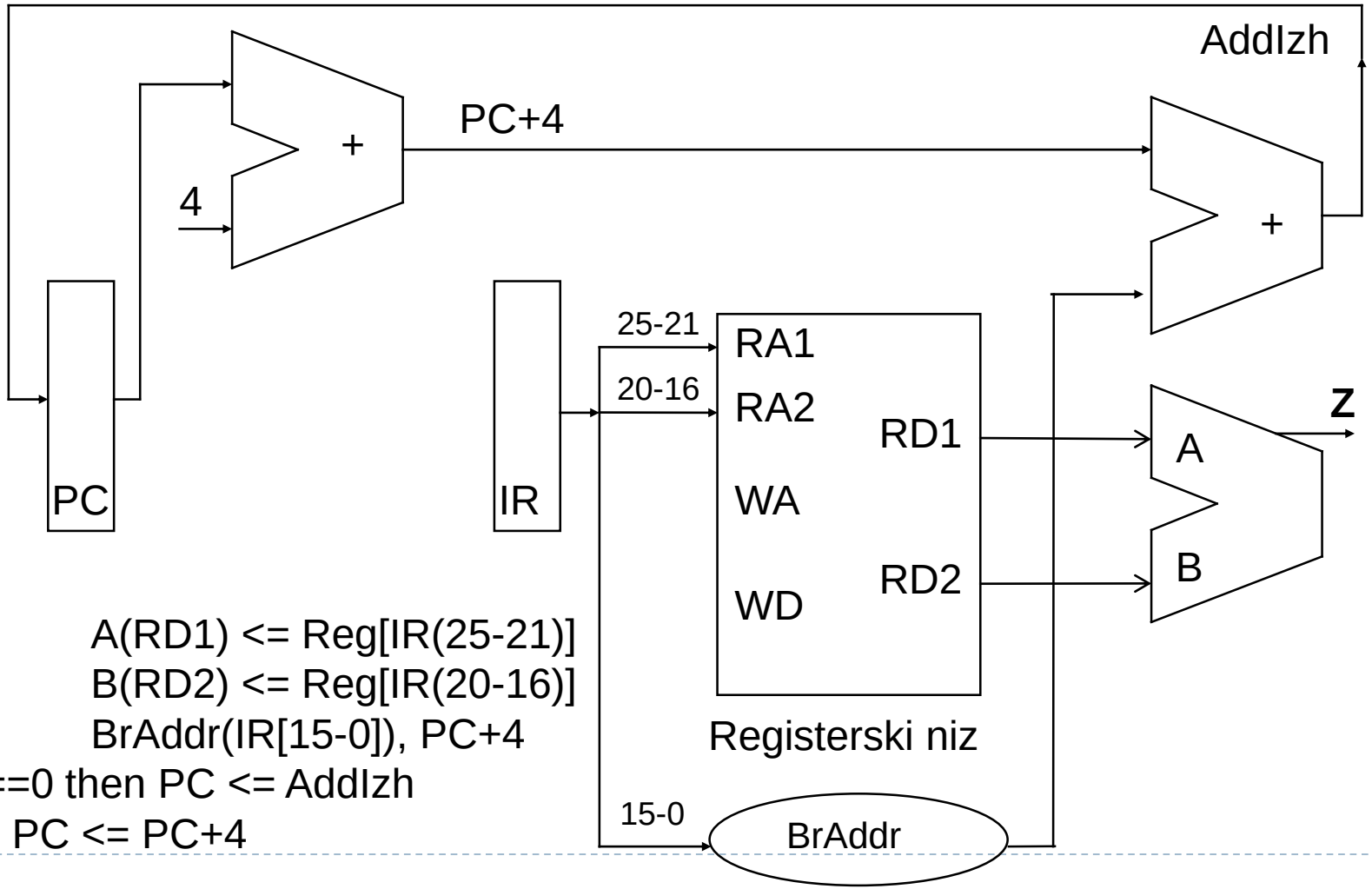


Naslavljanje registrov RS, RT, BrAddr, podatki na ALE in seštevalnik



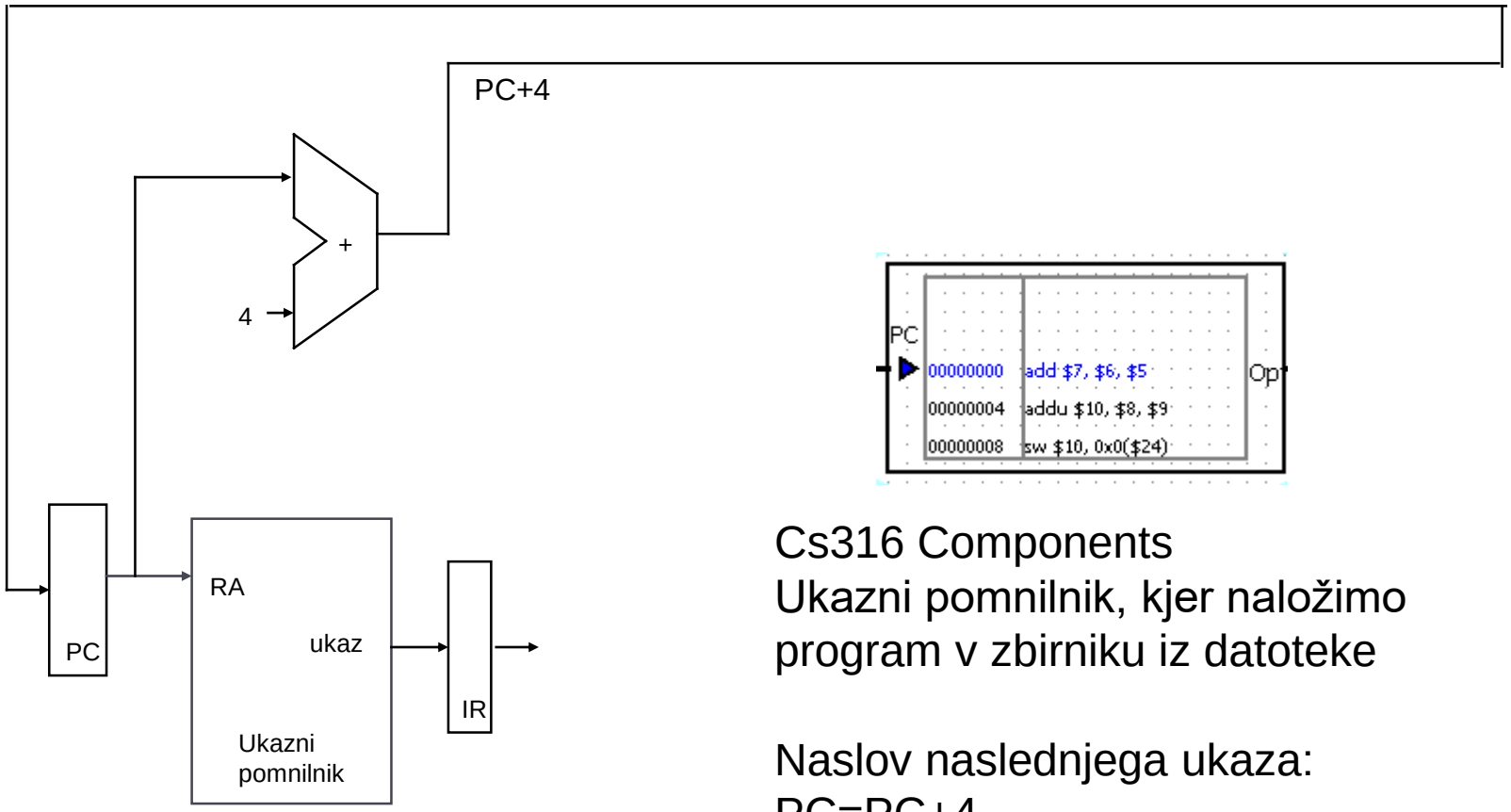
RS: A(RD1) <= Reg[IR(25-21)]
RT: B(RD2) <= Reg[IR(20-16)]
BrAddr(IR[15-0]), PC+4

Postavitev zastavice Z, izračun vejitvenega naslova in vpis v PC if Z=0



RS: A(RD1) <= Reg[IR(25-21)]
RT: B(RD2) <= Reg[IR(20-16)]
BrAddr(IR[15-0]), PC+4
if Z==0 then PC <= AddIzh
else PC <= PC+4

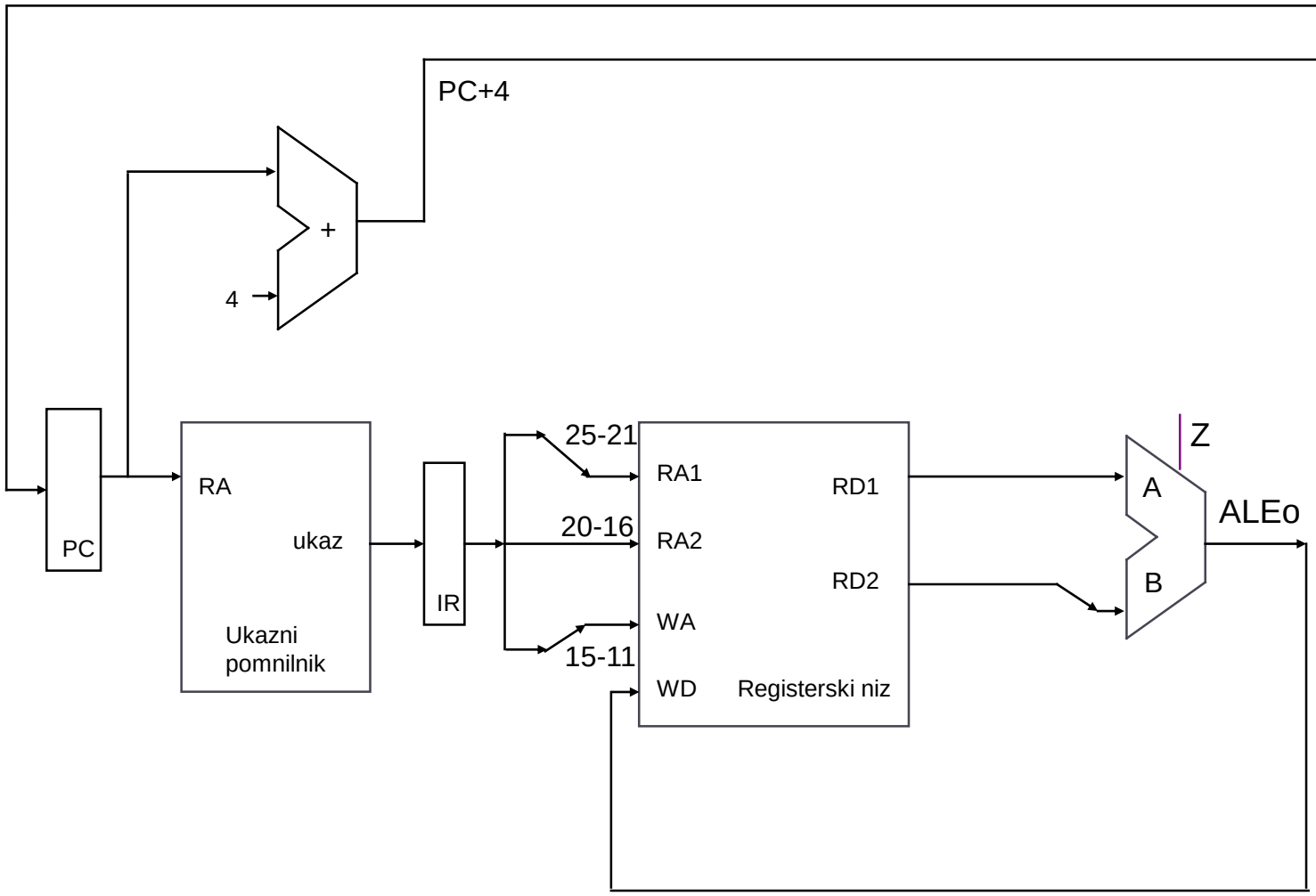
Procesor MIPS: dostava



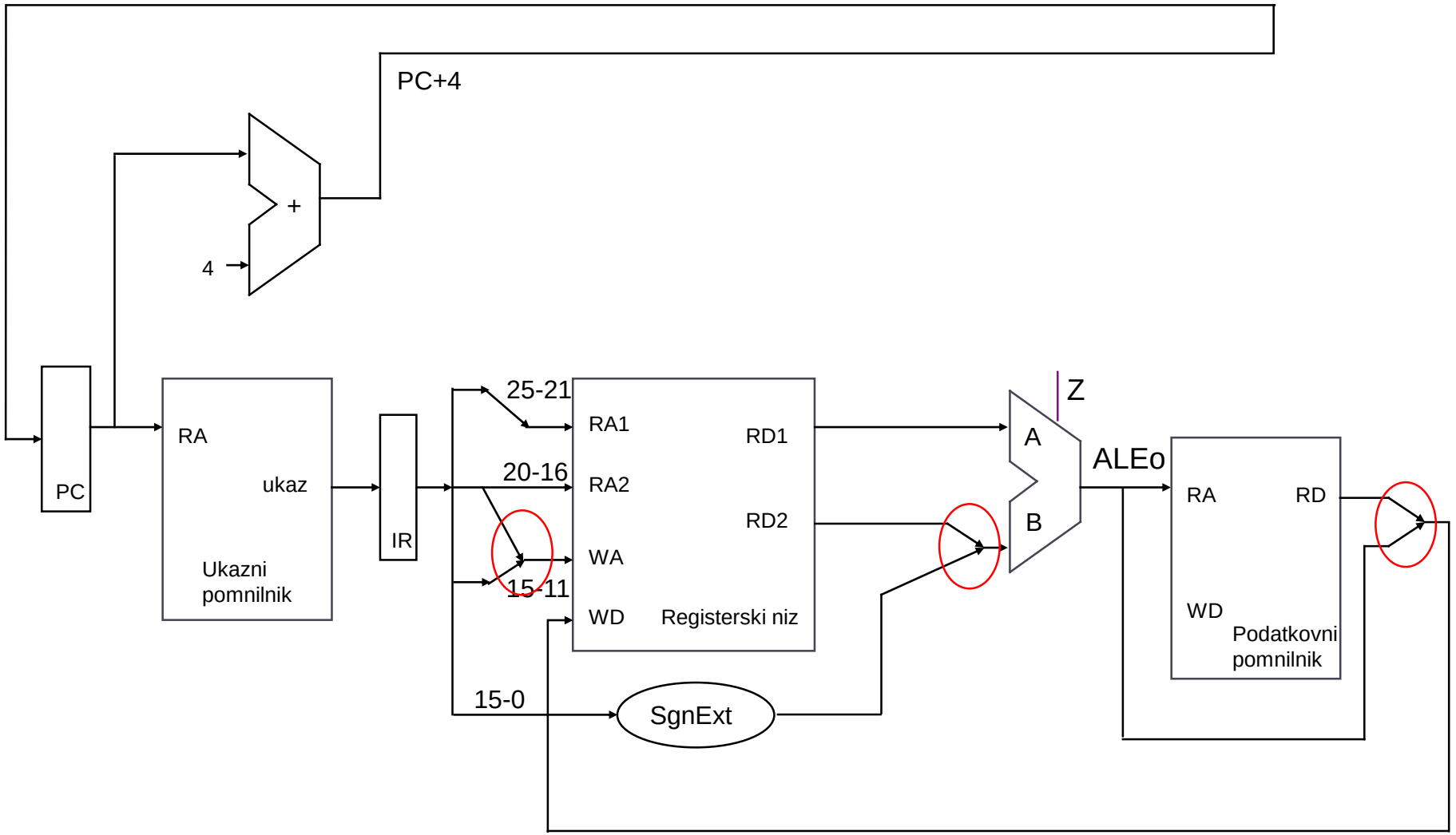
Cs316 Components
Ukazni pomnilnik, kjer naložimo program v zbirniku iz datoteke

Naslov naslednjega ukaza:
PC=PC+4

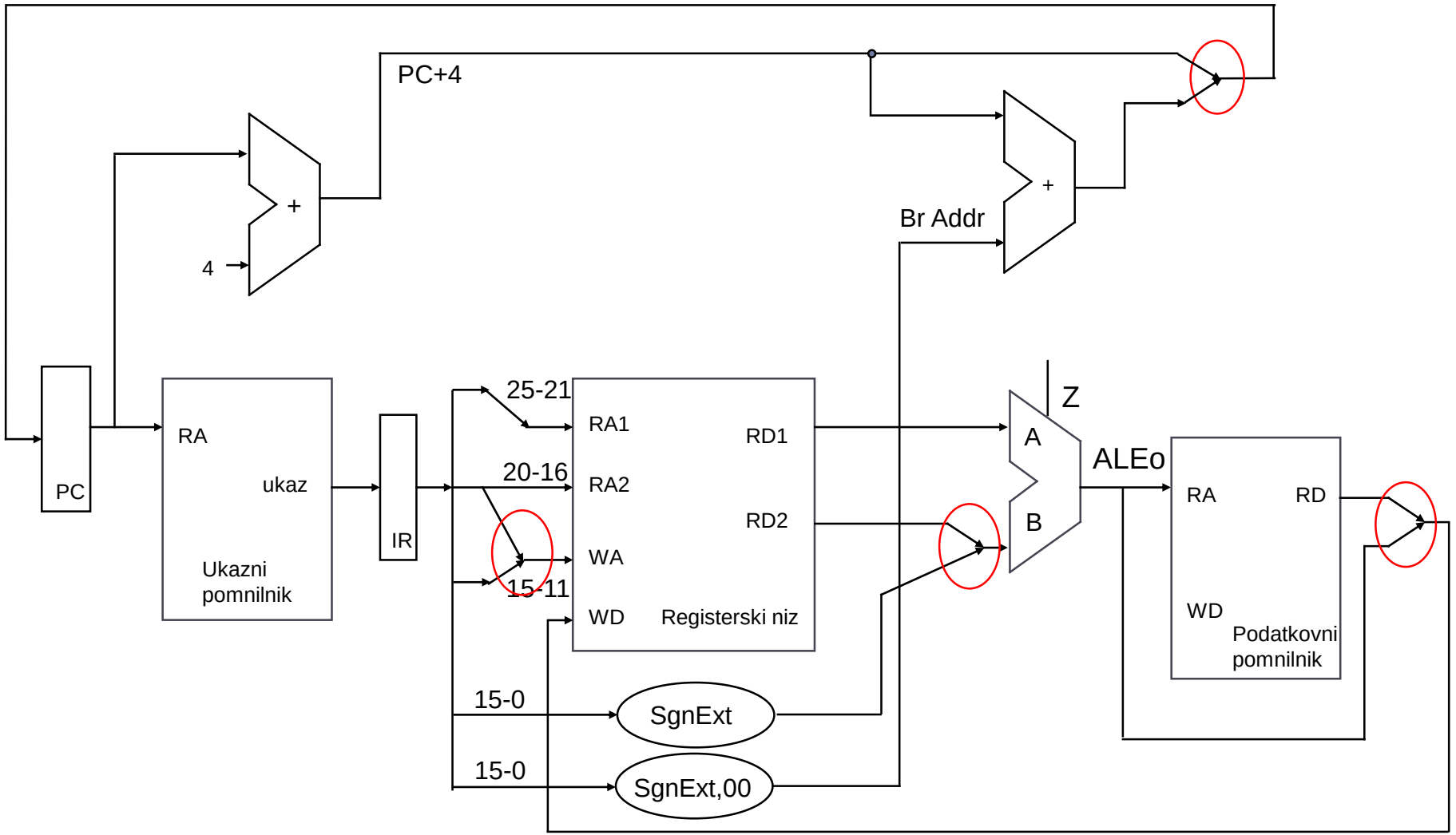
Procesor MIPS: R-tip (addu)



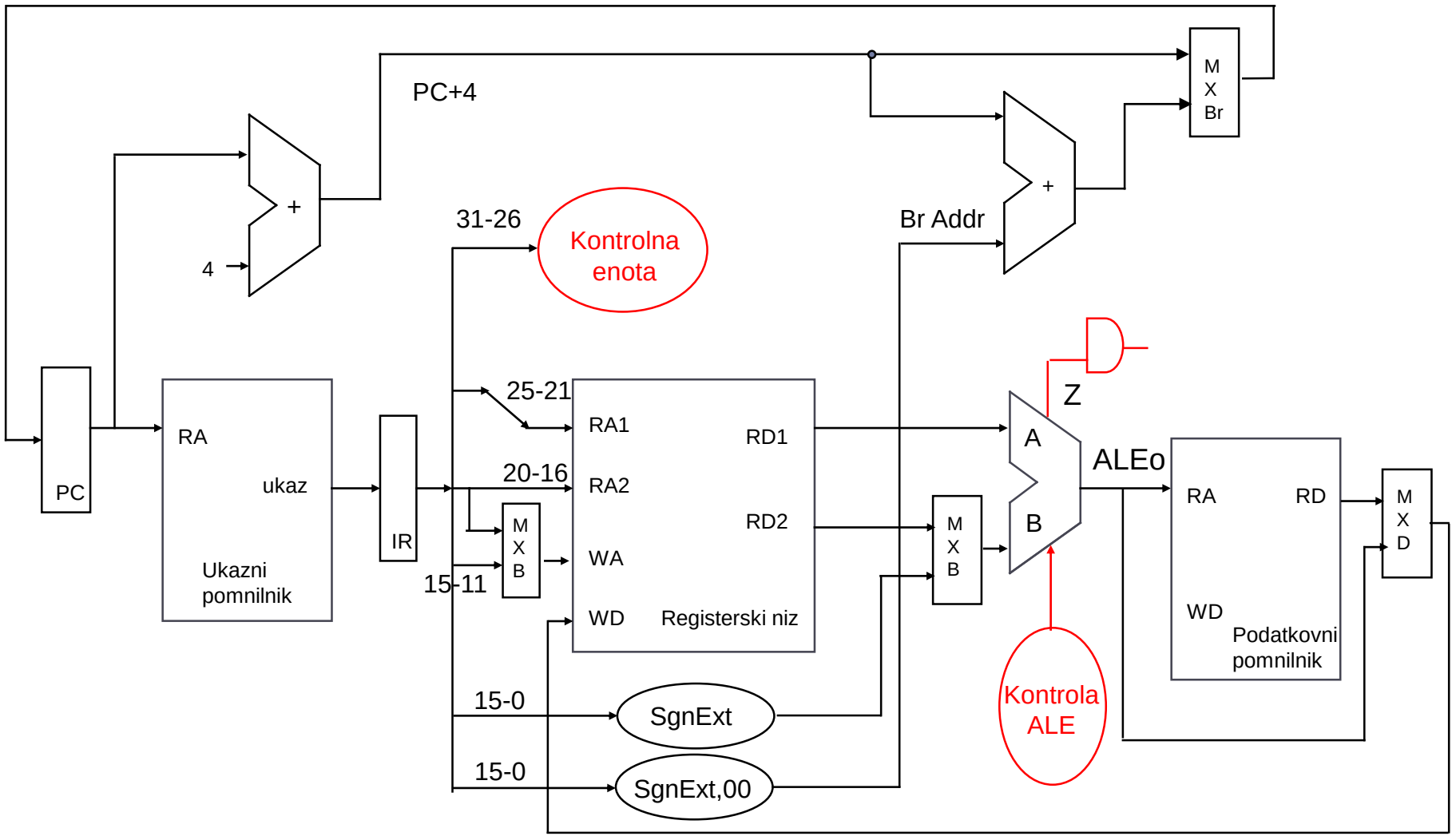
Procesor MIPS: addu + I-tip (lw)



Procesor MIPS: addu + lw + I-tip (bne)



Procesor MIPS: addu + lw + bne



Procesor MIPS - krmilni signali

MXB: **SrcB** (1 signal)
- 0: RD2
- 1: SgnExt

MXW: **Dst** (1 signal)
- 0: IR[20-16]
- 1: IR[15-11]

MXD: **RegW** (1 signal)
- 0: ALEo
- 1: RD

Registerski niz:
- **R-RW** (1 signal)
- 0: ni vpisa v register
- 1: vpis v register

Podatkovni pomnilnik:
- **P-RW** (1 signal)
- 0 - PW
- 1 - PR

MXBr: **Br** (1signal)
- 0: PC+4
- 1: Br Addr

ALE:

ALEop (2 signala)
- 00: R-tip (IR(5-0)) + ALEf
- 01: seštevanje (`lw`)
- 10: odštevanje (`bne`)

ALEf: (3 signali)
- 000: X+Y,
- 001: X+1,
- 010: X-Y,
- 011: X*Y
- 100: X/Y,
- 101: << X -pomik levo,
- 110: X>> -pomik desno

Krmilna enota za ALE

(IR(31-26)) - $ALEop_1$, $ALEop_0$ – določena na osnovi operacijske kode

000000 - addu

IR(5-0) – aritmetično/logična funkcija za ukaze R-tip

21H = 100001 - addu

100011 - lw

000101 - bne

	Ukaz	IR(31-26)	Operacija	ALE operacija	ALEop (2)	IR(5-0) funkcija	ALE (3) $f_2f_1f_0$
R-tip	addu	000000	Seštevanje	ADD	00	100001	000
I-tip	lw	100011	Naloži besedo	ADD	01	xxxxxx	000
I-tip	bne	000101	Primerjava	SUB	10	xxxxxx	010

ALE – kontrolni signali:

Ukaz	$ALEop_1$	$ALEop_0$	IR_5	IR_4	IR_3	IR_2	IR_1	IR_0	f_2	f_1	f_0
addu	0	0	1	0	0	0	0	1	0	0	0
lw	0	1	x	x	x	x	x	x	0	0	0
bne	1	0	x	x	x	x	x	x	0	1	0

Minimizacija funkcij za krmiljenje ALE:

$f_2 = 0$

$f_1 = ALEop_1.\sim ALEop_0$

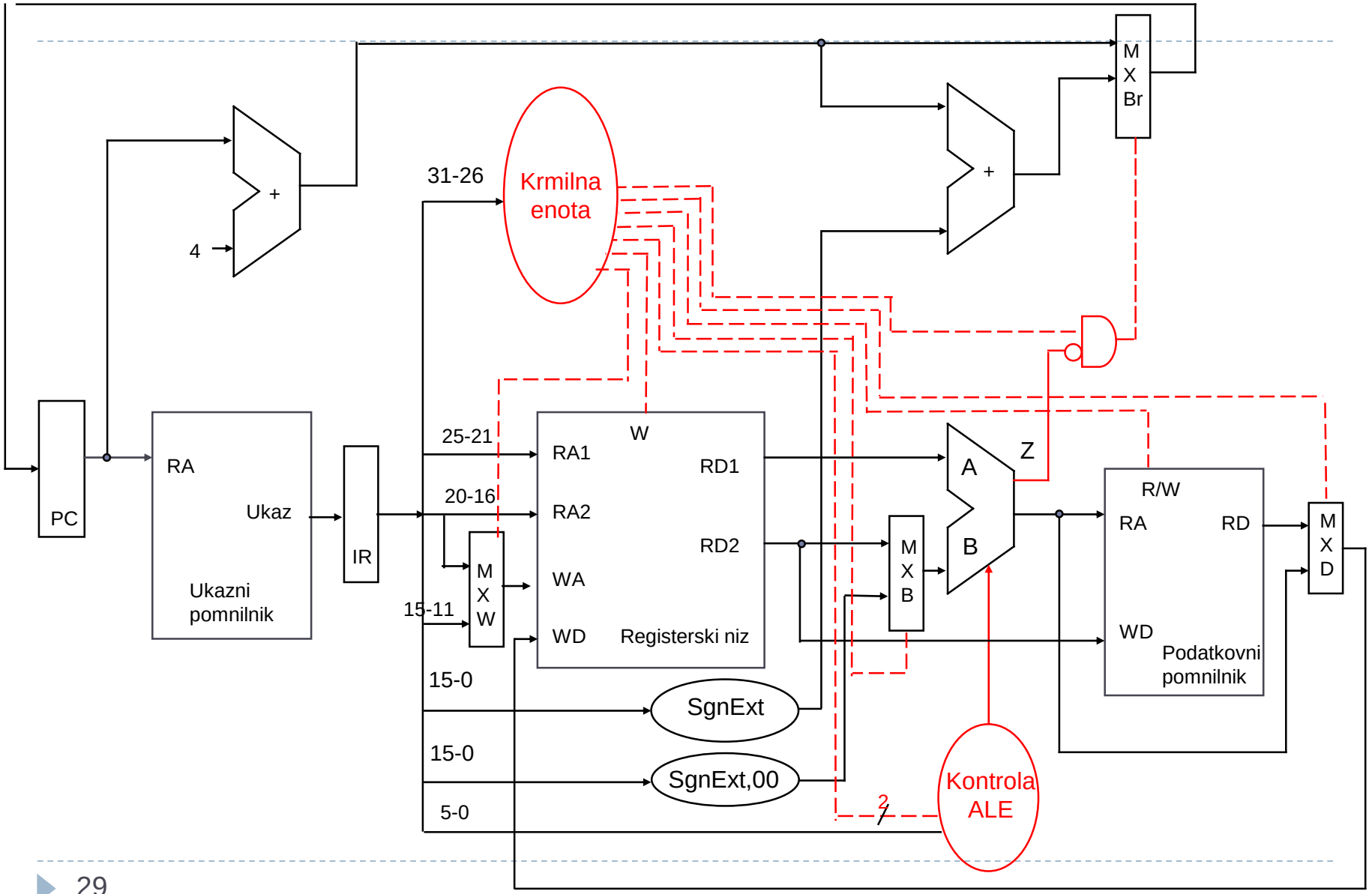
$f_0 = 0$

Krmilna enota – krmilni signali (ROM)

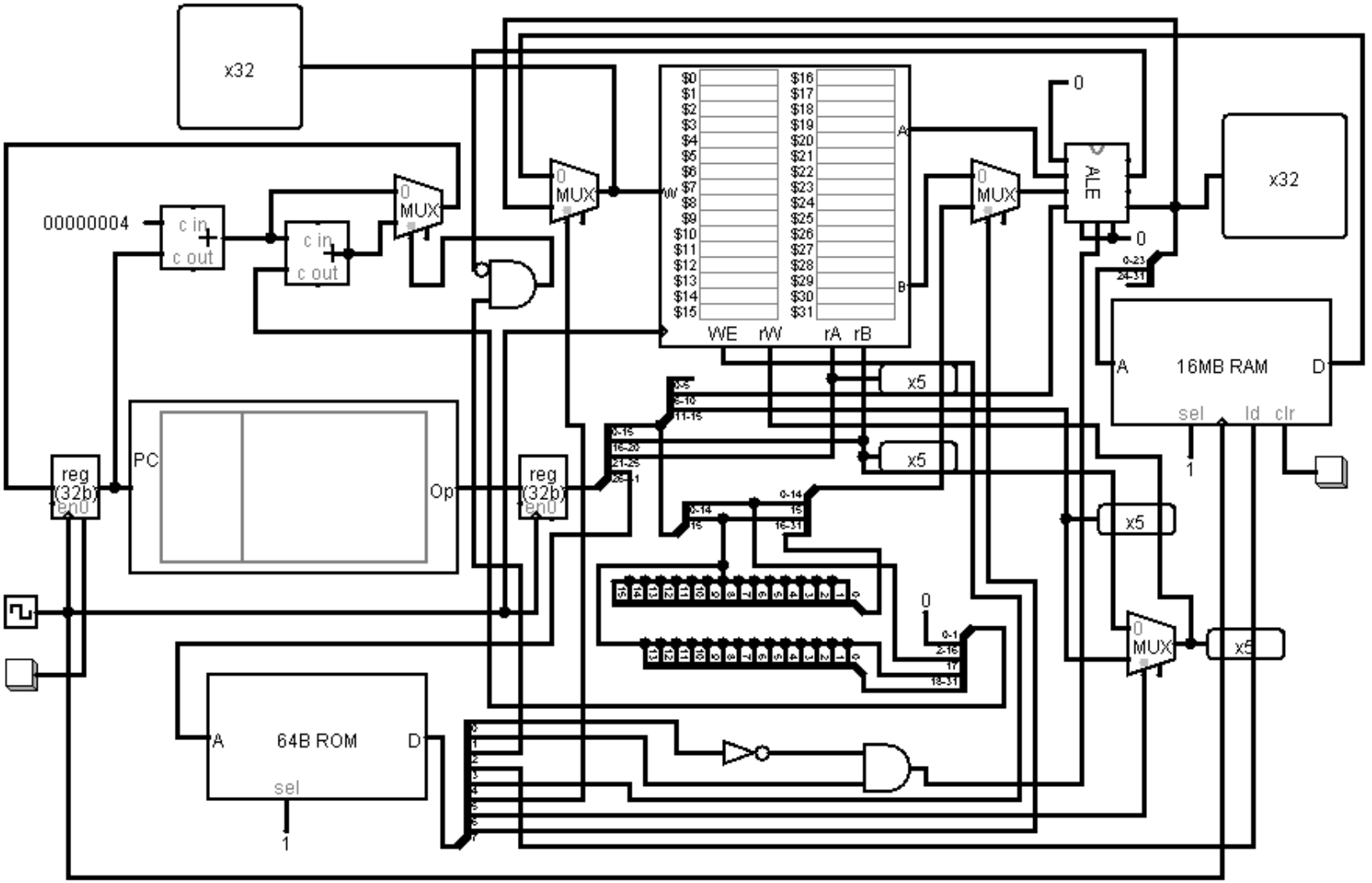
Ukaz	IR_{31} IR_{30} IR_{29} IR_{28} IR_{27} IR_{26}	**Src B**	**Dst**	**Reg W**	**R-RW**	**P-RW**	**Br**	**ALE op1**	**ALE op0**
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
addu	000000	0	1	1	1	1	0	0	0
lw	100011	1	0	0	1	1	0	0	1
bne	000101	x	x	x	0	1	1	1	0

Naslov ROM	IR_{31} IR_{30} IR_{29} IR_{28} IR_{27} IR_{26}	**SrcB**	**Dst**	**Reg W**	**R-RW**	**P-RW**	**Br**	**ALE op1**	**ALE op0**	**ROM podatek**
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
00	000000	0	1	1	1	1	0	0	0	78
23	100011	1	0	0	1	1	0	0	1	99
05	000101	x=0	x=0	x=0	0	1	1	1	0	0E

Procesor MIPS (shema)



Testiranje vezja – preveriti delovanje



Testni program (zbirnik)

- Preveri spodnji testni program.

```
# Test_SC
# R-tip: addu
# I-tip: lw, bne

# Registri:
# R2= 0x2
# R5= 0x1
# R6= 0x2
# R8= 0x5
# R9= 0x5
# R24= 0x0
```

```
#Pomnilnik
# 0: 2
# 1: 3

start:
    lw $1, 0($24)
enak:
    bne $2,$1, razlicen
    addu $2, $6, $5
    addu $10, $8, $9
    bne $2,$1, enak
razlicen:
    lw $2, 1($24)
```