

PREDLOŽI INSTRUKCIJSKI FORMAT TIPA RISC KOJIM BI SE INSTRUKCIJE IZ TRI OSLAVNA KLASA KOJEBIC PUNOCU SMO 16 BITOVA.

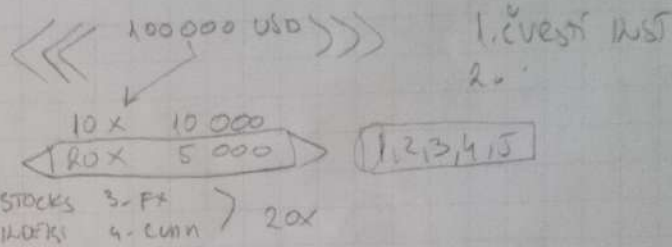
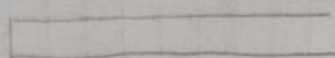
TRI KLASA:

ALU: NPR. ADD, SUB...

UPRAVLJACE: NPR. JAL, J, BNE...

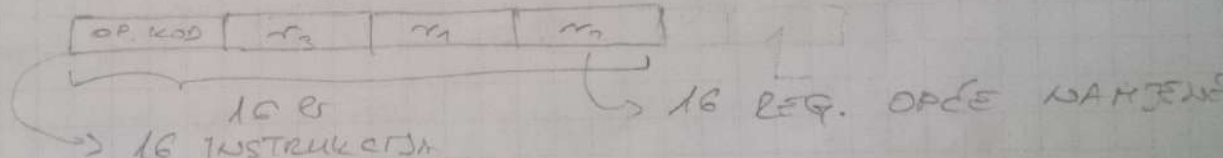
MEMORIJE: NPR. LOAD/STORE

- INSTRUKCIJA = OP. CODE + ARGUMENTI
- TREBA IMAT BAR MJEST ZA 3 OPERANDI ZA ALU OPERACIJE I, R, R3 S+5+5
- NE MOŽEMO 3 REGISTRA!



1) ALU - RIZNISLJAYE

- 1° broj REGISTARA → AKO JE 16 BITA, ⇒ 8 REGISTARA ⇒ 313 BITA
- 2° broj JABOVI (instrukcija) → 16 ⇒ 4 BITA

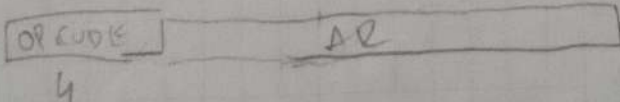


MIJERE "IDEALNOG" RISC

- 32 REGISTRA
- DELAYED BRANCHING
- 64 INSTRUKCIJE (FIRST)
- 4-5 RAZINA PROTOKOLNE STRUKTURE
- 2-3 MJEHA ADRESIRANJA
- SKLOPOVSKA CU (OZICHA IZBOJIM)

ADRESA

2) UPRAVLJACE



a) ABSOLUTNE 4 BITA OPERANDI + 12 BITA ADD

b) RELATIVNE → TREBA IMAT OPERANDI ⇒ 2 REGISTRA

- ZADANA JE 5-SEGMENTNI PUT, SA ZADATIM TRAJANJIMA SVAKOG SEGMENTA

IF: 150 ps EX: 125 ps WB: 100 ps

ID: 100 ps MEM: 150 ps

ODREDI LAGENCIJE OD: LD, ST, SUB, ADD ZA

a) NEPROTOČNO b) PROTOČNO STRUKTURU

a) NEPROTOČNO = SLEDNO.

- LOAD, LD : IF, ID, EX, MEM, WB = $150 + 100 + 125 + 150 + 100 = 625$ ps
- STORE, ST : IF, ID, EX, MEM, ~~WB~~ = 525 ps
- SUB : IF, ID, EX, ~~MEM~~, WB = 475 ps
- ADD : IF, ID, EX, ~~MEM~~, WB = 475 ps

b) PROTOČNO = PROJEKTO

≈ OPTIMALNO MPUTA ZA MSEGMENTA, A REALNO ≈ 67%
(PROVERITI JOŠ)

- PAZI: SEGMENTI NE TRAJU ISTO, MORAMO UZET LAŽEVI!!

- I TADA ON PREDSTAVJA TAKT, PERIOD TRAJAJA SEGMENTA!!

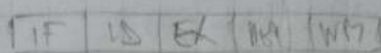
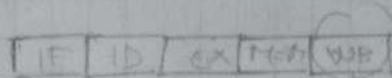
⇒ U 1 TAKTU OD 150 JA MOGU ZA 5 INSTRUKCIJA I SEGMENT (IDENI)

⇒ EFEKTIVNO SAM OBRADI INSTR U 150 ps

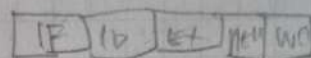
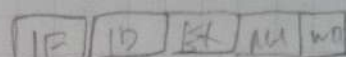
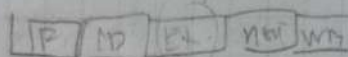
U ZADATKU PROGRAMSKIM ODJEKTO IDENTIFICIRAJ SVUG HAZARDE
(RAW, WAR, WAW).

SHENA

1) LW R1, 40 (R6)



2) ADD R6, R1, R2



3) SW R6, 50 (R1)

- OVAJ SADRŽI R1, DOK LW NE PISCUJE

W → R ⇒ RAW

- PRIGODI ILOD?

4) LW R5, -16 (R5)

- STRUKTURNI KONFLIKT

5) SW R5, -16 (R5)

- ~~STRUKTURNI KONFLIKT~~ (ADD NE KORISTI R5)

- LW, SW ⇒ RAW

6) ADD R5, R5, R5

- STRUKTURNI

- SW, LR ⇒ RAW

ALGO

1) GLEDATI SHENU PROTOKA
VIDJETI KONFLIKTE

2) DODATI PROMJER USTU ILIST

3) POGLEDATI OVISNOST ARGUMENTA IZ
INSTRUKCIJE I PRISTUPITELJE ZA UČAVANJE
PODNEHANU

PRAVILA

STRUKTURNI: KONFLIKT IF-NET (LOAD/STORE U KONFLIKTU SA TADIM IF)

PODNEKOVNI: RAW - READ AFTER WRITE! = AVO OČE ČIAT POJE WRITE ⇒ HAZARD

WAW - WRITE AFTER WRITE = BITKO TAD RIŠE PRAI TKA DRUGI

WAR - WRITE AFTER READ =

UPRAVNAČKI: AVO SE JUNKU

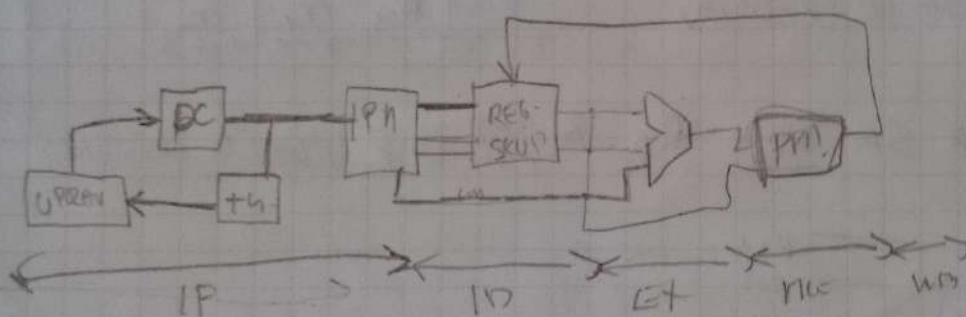
- SKICIRATI DIO PUTA PODATAKA PROTOKLA STRUKTURE MIPS KOJI JE AKTIVAN TIJEKOM IZVOĐENJA INSTRUKCIJA BEZUVJETNOG GRANANJA (RELATIVNOG) IZ OBRAZLOŽITI TAKVU ORGANIZACIJU. NAVESTI SVE AKTIVNOSTI IZ PUTA PODATAKA, SKUPINOM PONA PROTOKLIM SEGMENTIM.

BEZUVJETNO - NE GLEDAT PASTAVICE I SKANJA
RELATIVNO - $EADIS = PC + \text{POMAK (KONSTANTA)}$

NPR. JMP R2

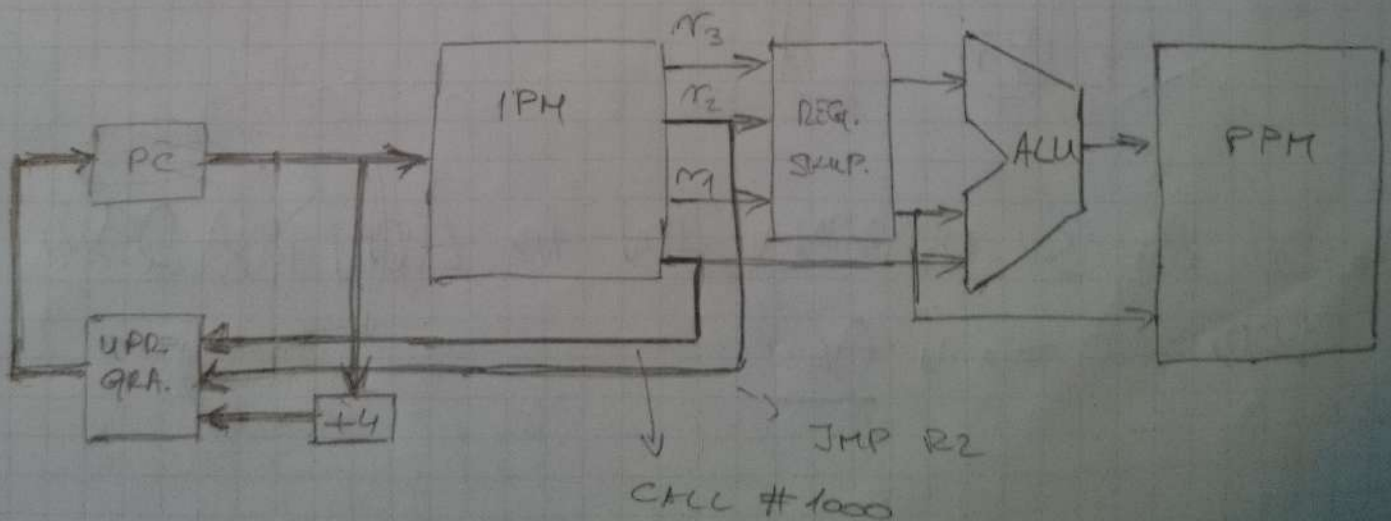
CALL #1000

RELATIVNO = POMAK (KONSTANTA); DIREKTNO = ADRESA

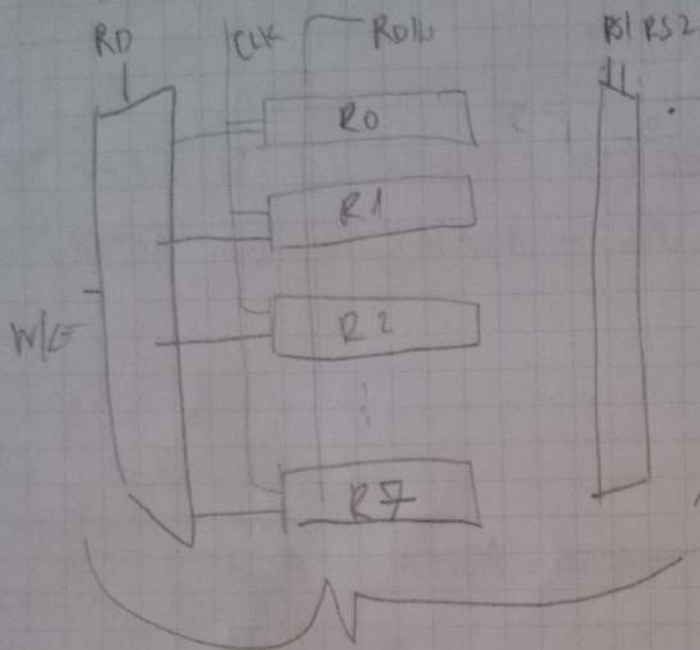


KI REGISTAR, KI KONST

ŠTO JE AKTIVNO??



PREDOŽITI IZVEDBU REGISTARSKOG SKUPA ZA PROCESOR SA 16 BITNIH REGISTARA OPĆE NAMENE. BITNO JE DA JE NOVOČE U SVAKOM CIKLUSU UPISATI I PROČITATI DVA REGISTRA.

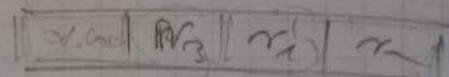


RS1 i RS2 BIRAJU REGISTAR
(KOJI ĆE DOĆI U ALU!!)

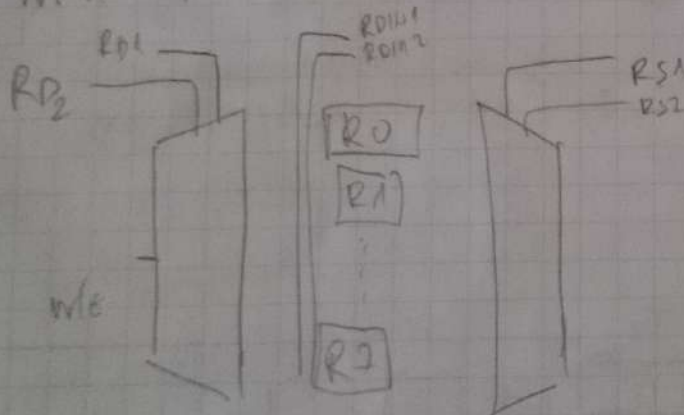
RD ODREĐUJE
ODREĐENI REGISTAR.

npr. add r_3, r_1, r_2
 \downarrow \downarrow \downarrow
 RD RS1 RS2

ORGANIZACIJA SHENA IZ PREDNASTAV



⇒ NAMA TREBA SHENA SA DVA RD-A, TE DVA ULAZA SADRŽAVO

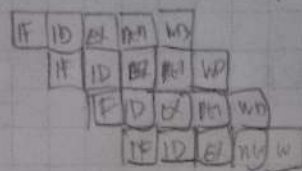


- OVI KOJI SU ODABRALI IDU NA LJEVO I LA DESNO
SADRŽAVO ⇒ U ALU

ZA PROTOČNI RISC PROCESOR IZA SLJEDUĆI PROGRAMSKI ODJEČAK
UKAŽITE I OZNAČITE MJESTO HAZARDA (AKO ON POSTOJI), TE A; B.

- 1) LOAD R1, A
- 2) LOAD R2, B
- 3) ADD R3, R1, R2
- 4) LOAD R4, D

1 2 3 4 5 6 7 8 9 10

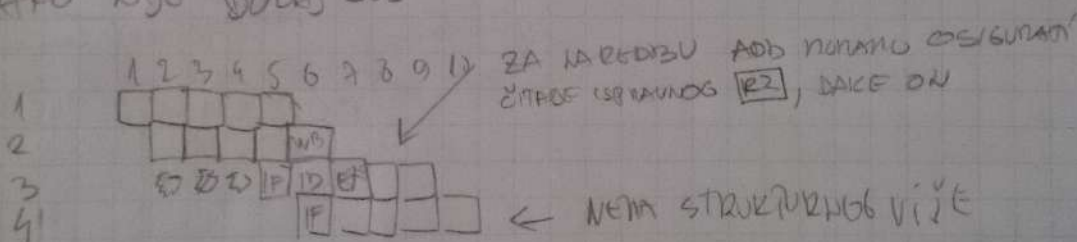


2 WB, 3 R ⇒ RAN
STRUKTURNI 1-4

- ILBO 1) VIDI STRUKTURNI?
2) PODATKULI?

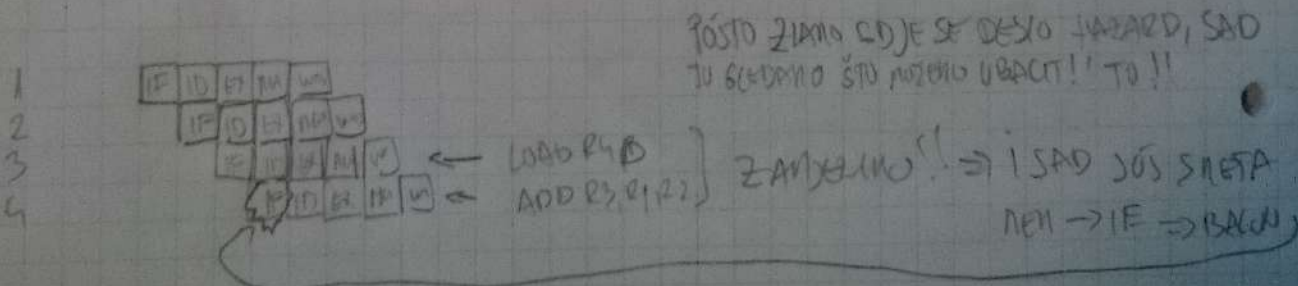
A) POTREBNO PRELAČITI KOD SA JE-OPTIMIRANIM PREVODIČEM.

- ⇒ SMO STAVJA NOP-ove, TANO PRIJE HAZARDA, KOLIKU NOPOVA??
- ⇒ DODAJ NOP I GLEDAČ JER OTIČO HAZARDA.
- ⇒ AKO NJE DODAJ JOŠ.



B) PRELAČITI PROGRAM NA NAČIN KAKO BI TO UČINIO OPTIMIRAJUĆI PREVODIČ.

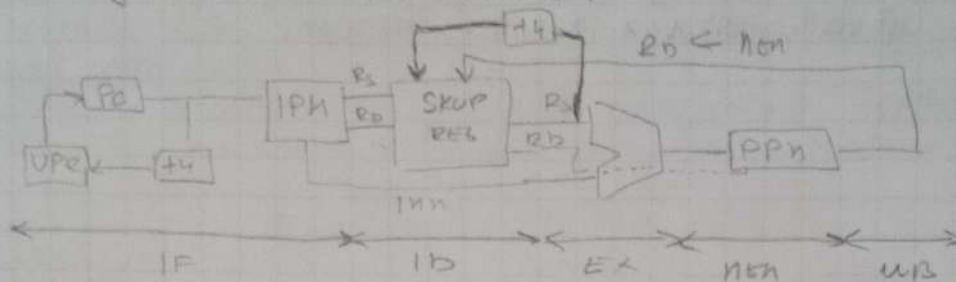
- OPTIMIRAJUĆI PREVODIČ: ON UZIMA "SEGMENTE" ILI OD HAZARDA ILI ON IZA
TE IZ GUPA U PRIKLJUK ZA KASNIJE
- PRIKLJUK ZA KASNIJE: JE OVAJ SEGMENT NEPODRŽAN PRIJE HAZARDA!



PRIKAZATI KAKO BISMO SLJEDEĆE INSTRUKCIJE UKLOPILI U RISC-NIPS. KOJE BI POTREBNO Bilo DODATNO SKLOPOVNE UVRSTITI U PUT PODATAKA?

- a) STWNC RD, OFFSET, RS // $MEM[RS+OFFSET] \rightarrow RD$; $RS = RS+4$
b) ADDN RD, OFFSET, RS // $RD + MEM[RS+OFFSET] \rightarrow RD$

a) NACRTAJMO KRATČAN PUT PODATAKA.



- 1° ZBROJI $RS+OFFSET \Rightarrow EX(RS, OFFSET)$, ID (RS, INH)
- 2° DOHVAĆI IZ MEM $\Rightarrow MEM$
- 3° POVRATI U $RD \Rightarrow WB$

IF: DOHVAĆI $PC+4$

ID: DEKODIRAJ, PRIGAVI $RS, OFFSET$

EX: ZBROJI

MEM: DOHVAĆI SA $MEM[RS+OFFSET]$

WB: POVRATI U WB RD , TE RS

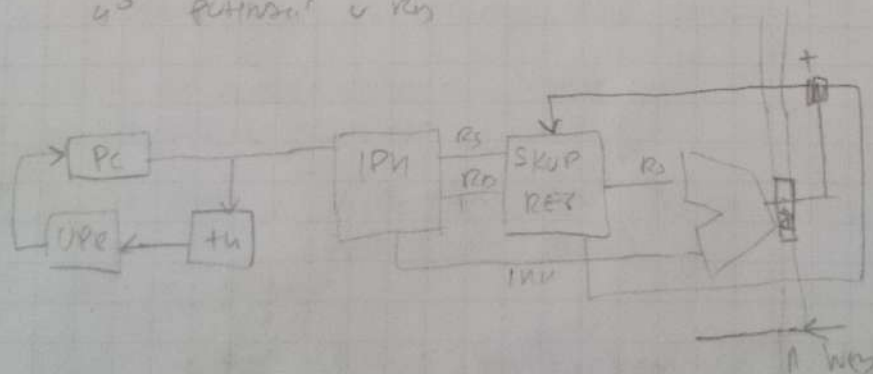
- NO IJAŠ INSTRUKCIJA NORA ODRADITI! $RS = RS+4$!!

- NOVANO DODATI SKLOPOVNE KOJE ĆE U WB FAZI INSTRUKCIJE UPIŠATI U RS

b) ADDN RD, OFFSET, RS

TEK NAKON MEM FAZE SNIJEŽI ZBROJ
JER NEMA SNILA.

- 1° ZBROJI $OFFSET + RS$
- 2° DOHVAĆI IZ MEM
- 3° ZBROJI SA RD
- 4° POVRATI U RD



2.

ZADAN JE NEOPTIMIZIRAN PROGRANSKI ODSJEČAK ZA ARHITEKTURU SA ZAKLAJEM INstrukcija:

MY SUB : ADD R6, R5, R6

LOOP : LW R1, 0, R5

NOP

ADD R1, R1, R7

SW R1, 0(R5)

ADD R5, R5, 4

BNE R5, R6, LOOP

NOP

JR R31

LOAD R_x, 0(R₁) ZNAČI
UČITAJ U R_x IZ R₁ SA
ODMAKOM NULA

JMP
PRINJER

A) KOJU FUNKCIJU OZNAČAVA KOD, NAPIŠI C KOD.

B) PRIKAŽI OPTIMIZIRAN KOD SA JEDNOSTRUKIM IZDAVANJEM (SA PRISILJEM)

A) void mysub (R5, R6)
 R6 = *R5;

DO {
 R1 = R5;
 R1 = R1 + R7;

*R5 = R1;

R5 = R5 + 4;

} WHILE (R5 <> R6)

JEBI = KOD, FUNKCIJE!
TAKO ZNAMO DA JE PETA
DIO MYSUB

STORE →
LOAD ←

TRIAL AND ERROR!

B) ADD LW

IF	ID	EX	MEM	WB
	IF	ID	EX	MEM
		IF	ID	EX
			IF	ID

LW ADD

IF	ID	EX	MEM	WB
	IF	ID	EX	MEM
		IF	ID	EX
			IF	ID

ZADANA JE FUNKCIJA U JEZIKU C:

```
void ADD_SCALAR_TO_VECTOR (int scalar, int *p, int size)
{
    int *end = p + size;
    while (p != end)
    {
        *p = *p + scalar;
        ++p;
    }
}
```

JMP
Printer

NAPISATI NEOPTIMIZIRANI KOD ZA MIPS, ZAKLAJTELO SNAJAV.

BEQ R1, R2, LABELA

PREDLOŽITI RASPORED INstrukcija koji bi se poboljšao performansi

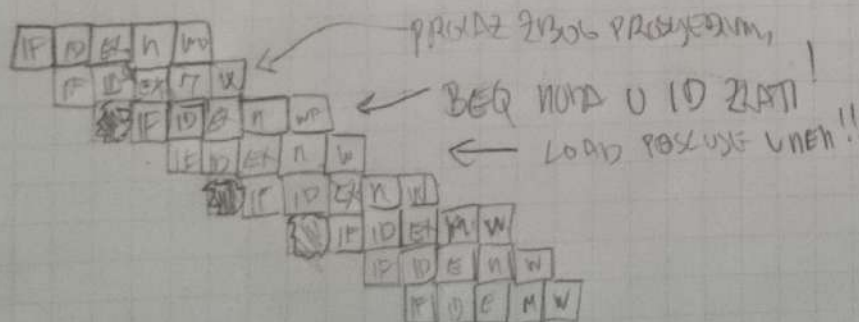
OVO JE KOD ZA HAZARD

```
ADD_SCALAR_TO_VECTOR: MUL R5, R3, 4 ①
                        ADD R4, R4, R5 ②
PETH_P: BEQ R1, R2, KRAJ ③
        LW R6, 0(R2) ④
        ADD R6, R1, R6 ⑤
        SW R6, 0(R2) ⑥
        ADD R2, R2, 4 ⑦
        J PETH_P ⑧
KRAJ: J R31 ⑨
```

SCALAR = R1
P = R2
SIZE = R3
END = R4

POVEĆATI ADRESU ZA 4B! ZASTO?
JE IHT ZAVRŠNA 4B, PA NI NIKAD
NIKAD ZAN

XP LW R6, 0(R2)
LW R6, R2 → 40h



LW MEM
BEQ ID
SW MEM
ADD EX

RASPORED ZA OPTIMIZACIJU: (GLEDAJU SAMO MJESTA HAZARDA!) → 3 5 1 6