

1 Naredbe projektiranog procesora

Sve naredbe procesora jednake su dužine, 32 bita.

1.1 Aritmetičke i logičke naredbe

Izvede se isključivo nad sadržajima spremnika procesora. Obuhvaćaju dva operanda i rezultat. Jedan operand je uvijek varijabla, dok drugi operand može biti varijabla ili konstanta.

1.1.1 Naredba za zbrajanje dviju varijabli

$$z = x + y$$

gdje su x, y i z spremnici.

Oblik naredbe je:

ADD R[ra],R[rb],R[rc]

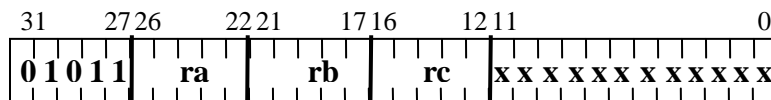
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

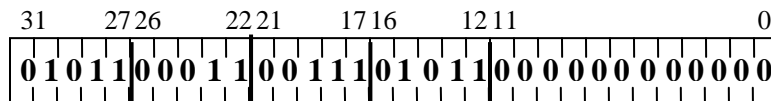
R[rc] spremnik u koji se drugi operand

Operacijski kod naredbe je 11,



Primjer:

ADD R3,R7,R11



1.1.2 Naredba za zbrajanje varijable i konstante

$$z = x + c21$$

gdje su x, z spremnici, a c21 numerička konstanta.

Oblik naredbe je:

ADDI R[ra],R[rb],c2

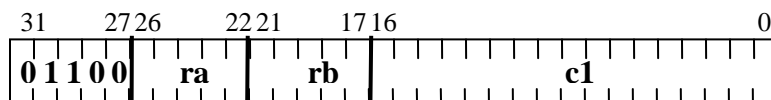
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

c2 numerička konstanta

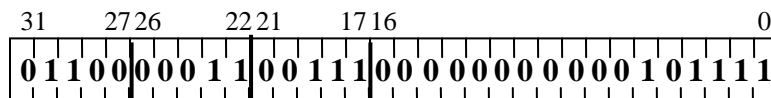
Operacijski kod naredbe je 12,



Konstanta c1 ograničena je na 17 bita i kako može poprimiti i negativnu vrijednost njen iznos ograničen je na -2^{16} do $2^{16}-1$.

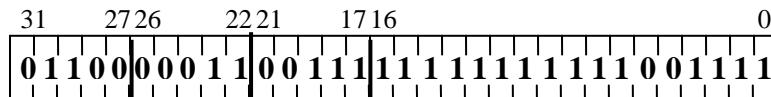
Primjer:

ADDI R3,R7,47

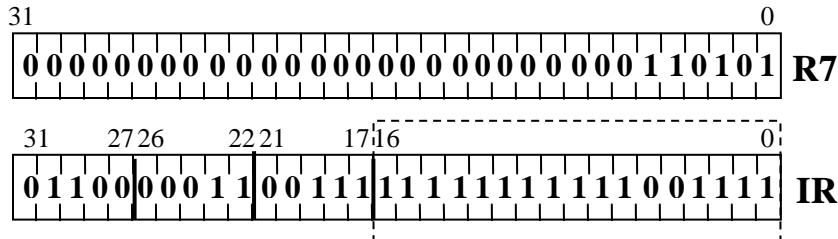


ili

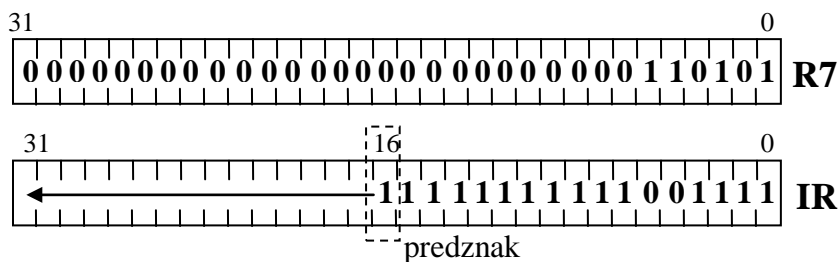
ADDI R3,R7,-47



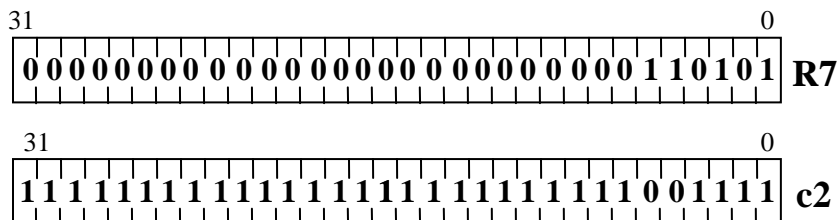
Problem je zbrajanje spremnika 32 bita i konstante c2 koja je sastavni dio IR i veličine je 17 bita. Neka je sadržaj R7 53.



odnosno predznak (16 bit) proširi se na sve preostale bitove od 16 do 31:



te se dobije:



1.1.3 Naredba za oduzimanje dviju varijabli

$$z = x - y$$

gdje su x, y i z spremnici.

Oblik naredbe je:

SUB R[ra],R[rb],R[rc]

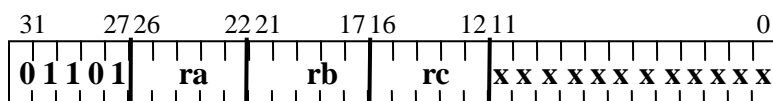
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

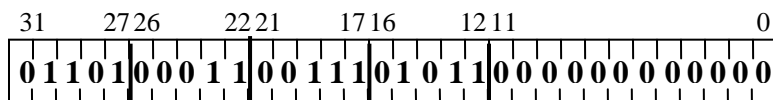
R[rc] spremnik u koji se drugi operand

Operacijski kod naredbe je 13,



Primjer:

SUB R3,R7,R11



1.1.4 Naredba za logičko I dviju varijabli

$$z = x \text{ AND } y$$

gdje su x, y i z spremnici.

Oblik naredbe je:

AND R[ra],R[rb],R[rc]

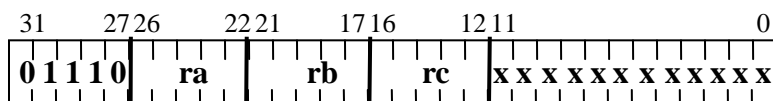
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

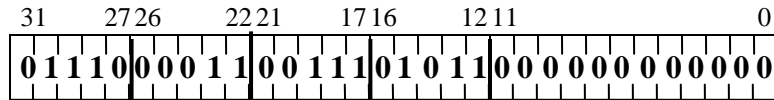
R[rc] spremnik u koji se drugi operand

Operacijski kod naredbe je 14,



Primjer:

AND R3,R7,R11



1.1.5 Naredba za logičko I varijable i konstante

$z = x \text{ AND } c2$

gdje su x, z spremnici, a c2 numerička konstanta.

Oblik naredbe je:

ANDI R[ra],R[rb],c2

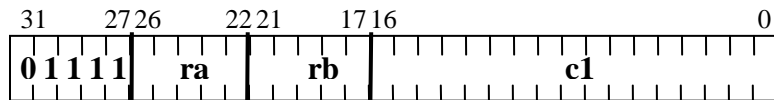
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

c2 numerička konstanta

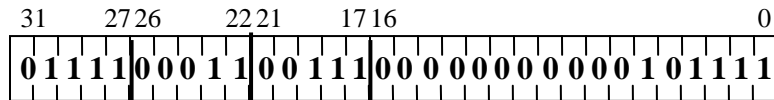
Operacijski kod naredbe je 15,



Konstanta c1 ograničena je na 17

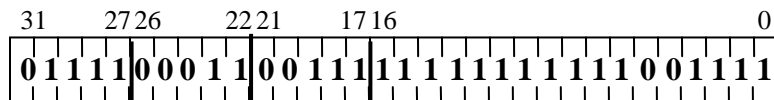
Primjer:

ANDI R3,R7,47



ili

ANDI R3,R7,-47



1.1.6 Naredba za logičko ILI dviju varijabli

$z = x \text{ OR } y$

gdje su x, y i z spremnici.

Oblik naredbe je:

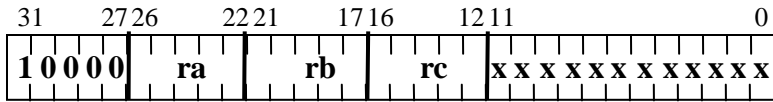
OR R[ra],R[rb],R[rc]

Gdje su:

R[ra] spremnik u koji se sprema rezultat

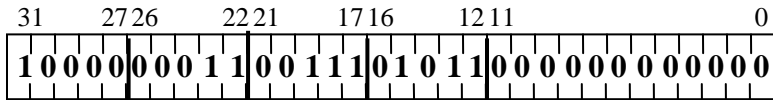
R[rb] spremnik u koji se nalazi prvi operand

R[rc] spremnik u koji se drugi operand
Operacijski kod naredbe je 16,



Primjer:

OR R3,R7,R11



1.1.7 Naredba za logičko ILI varijable i konstante

$$z = x \text{ OR } c_{21}$$

gdje su x , z spremnici, a c_2 numerička konstanta.

Oblik naredbe je:

ORI R[ra],R[rb],c2

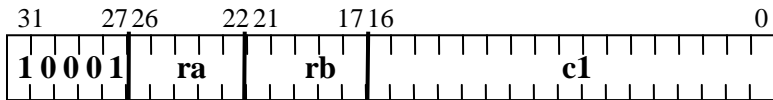
Gdje su:

R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi prvi operand

c2 numerička konstanta

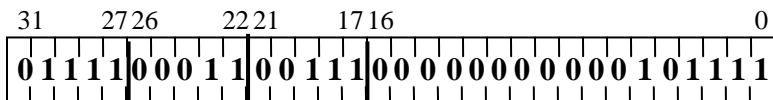
Operacijski kod naredbe je 17,



Konstanta c_1 ograničena je na 17

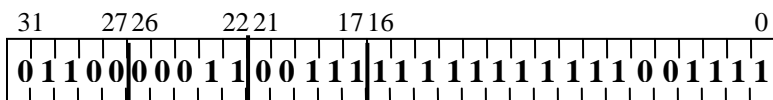
Primjer:

ORI R3,R7,47



ili

ORI R3,R7,-47



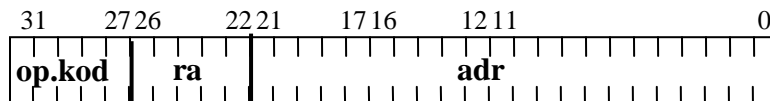
1.2 Naredbe za prijenos podataka

Naredbe za prijenos podataka koriste se za prijenos podataka iz memorije u spremnik, odnosno iz spremnika nazad u memoriju. Naredba mora specificirati smjer prijenosa, spremnik u koji se sprema podatak iz memorije ili iz kojega se podatak prenosi u memoriju, te memorijsku lokaciju koja se koristi u prijenosu podataka.

1.2.1 Naredba za prijenos podatka iz memorije u spremnik

$$R[ra] = M[adr]$$

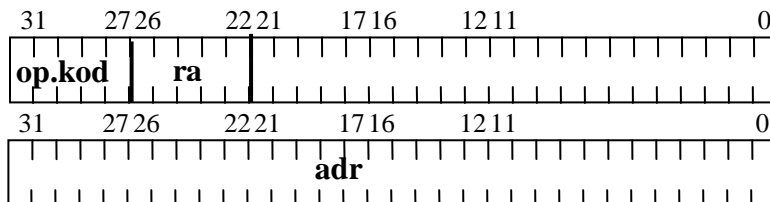
gdje su **ra** spremnik u koji se podatak upisuje,
adr je adresa sa koje se podatak upisuje u spremnik.
Mogući format naredbe bio bi:



odakle slijedi da adresa može biti samo 22 bita, odnosno da je moguće adresirati sveg 4MB memorije, što definitivno nije zadovoljavajuće.

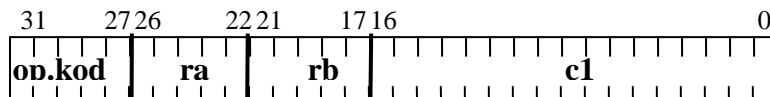
Moguća rješenja su:

- Proširiti naredbu za dodatnih 32 bita



Ovim rješenjem narušava se koncept da su sve naredbe iste dužine.

- Koristiti druge adresne modove osim direktnog adresiranje, npr. indirektno pomoću registra ili bazno pomoću registra (registar + baza)



gdje je $\mathbf{adr} = R[\mathbf{rb}] + c1$

Drugo rješenje puno je bolje te će se koristiti kod projektiranog procesora.

Oblik naredbe je:

LD R[ra],R[rb],c1

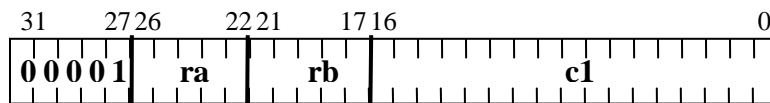
Gdje su:

R[ra] spremnik u koji se sprema rezultat

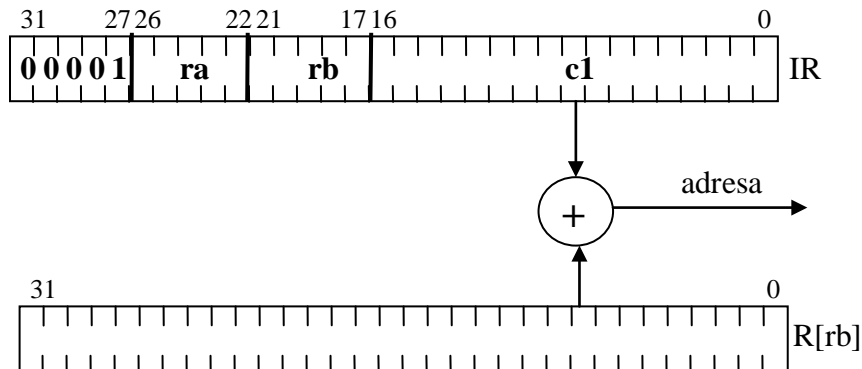
R[rb] spremnik u koji se nalazi baza adrese

c1 konstanta u dvostrukom komplementu koja je indeks adrese.
 $-2^{16} \leq c1 < 2^{16}$

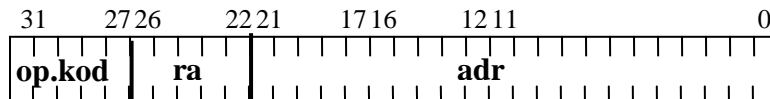
Operacijski kod naredbe je 1.



Postupak generiranja adrese prikazan je sljedećom slikom.

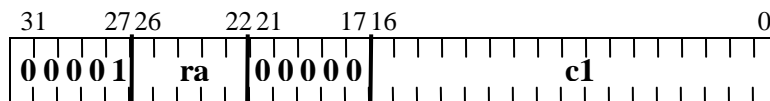


Ponekad je zgodno imati mogućnost upisa direktno s adrese bez proračuna adrese uz ograničenja na adresu kojoj se direktno pristupa. Moguća rješenja su:



gdje je ovo sada nova naredba s novim operacijskim kodom.

Drugo rješenje je da se iskoristi postojeća naredba uz određene modifikacije, na način da se direktno adresira ukoliko je polje **rb** = 0.



Ovakvo ograničenje pretpostavlja da se za indirektno adresiranje ne može koristiti spremnik R[0].

Kod projektiranja procesora koristiti ćemo drugo rješenje.

1.2.2 Naredba za prijenos podatka iz spremnika u memoriju

Oblik naredbe je:

ST R[ra],R[rb],c1

Gdje su:

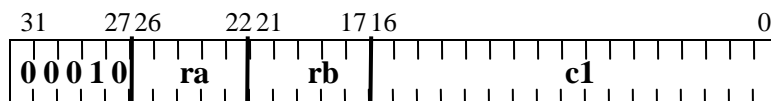
R[ra] spremnik u koji se sprema rezultat

R[rb] spremnik u koji se nalazi baza adrese

c1 konstanta u dvostrukom komplementu koja je indeks adrese.

$$-2^{16} \leq c1 < 2^{16}$$

Operacijski kod naredbe je 2.



1.2.3 Naredba za upis konstante u spremnik

Prethodne dvije naredbe zahtijevaju da se u spremnik R[rb] unese baza. U skladu s navedenim potrebno je uvesti naredbu za upis konstante (bazne adrese) u spremnik. Oblik naredbe je:

LDA R[ra],c2

Gdje su:

R[ra] spremnik u koji se sprema konstanta

c1 konstanta u dvostrukom komplementu koja je indeks adrese.

$$-2^{21} \leq c1 < 2^{21}$$

Operacijski kod naredbe je 3.



1.3 Naredbe za upravljanje programskim tokom

Naredbe za upravljanje programskim tokom mijenjaju slijedno izvođenje naredbi (izvodi se sljedeća naredba u nizu) na način da se izvođenje prebacuje na neku drugu memorijsku adresu (preskače se određeni broj naredbi).

Ove naredbe mijenjaju sadržaj programskog brojila.

Razlikuju se:

- Bezuvjetna grananja
- Uvjetna grananja

Bezuvjetno grananje prebacuje automatski izvođenje na adresu naredbe specificirane naredbom za grananje.

Uvjetne naredbe ispituju uvjet i ukoliko je isti zadovoljen izvodi se grananje na adresu naredbe specificirane naredbom za grananje. Uvjet je neki aritmetički i logički izraz koji se izračuna i kao rezultat pohrani u neki spremnik. Slijedi ispitivanje sadržaja spremnika na jednostavne uvjete:

- Pozitivan (bit predznaka 0)
- Negativan (bit predznaka 1)
- Jednak nula (svi bitovi nula; NILI funkcija svih bitova)
- Različit od nula (barem jedan bit = 1; ILI funkcija svih bitova).

Primjer:

$$a + b < c - 32;$$


```

LDA  R1, adr a
LD   R2, R1, 0    ; a u R2
LD   R3, R1, 4    ; b u R3
LD   R4, R1, 8    ; c u R4
ADD  R5, R2, R3    ; a + b
ADDI R6, R4, -32   ; c - 32
SUB  R5, R5, R6    ; (a + b) - (c - 32)

```

U spremniku R5 rezultat je operacija i ako je sadržaj R5 negativan uvjet je zadovoljen.

U naredbama za grananja moguće je adresu grananja specificirati na dva načina:

- Relativno u odnosu na trenutnu adresu (sadržaj PC)
- Apsolutno

Temeljem navedenog imati ćemo sljedeće naredbe za relativna i apsolutna grananja.

1.3.1 Naredbe za relativna grananja

Naredbe za relativna uvjetna grananja uspoređuju sadržaj spremnika R[ra] s uvjetom i ukoliko je uvjet zadovoljen grana se na naredbu na adresi PC + c2, odnosno:

$$PC = PC + c2$$

Imaju format:



gdje je $-2^{21} \leq c2 < 2^{21}$

Naredba koja izvodi grananje ako je sadržaj spremnika R[ra] pozitivan

BRP R[ra], c2

Operacijski kod je 20

Naredba koja izvodi grananje ako je sadržaj spremnika R[ra] negativan

BRM R[ra], c2

Operacijski kod je 21

Naredba koja izvodi grananje ako je sadržaj spremnika R[ra] nula

BRZ R[ra], c2

Operacijski kod je 22

Naredba koja izvodi grananje ako je sadržaj spremnika R[ra] različit od nule

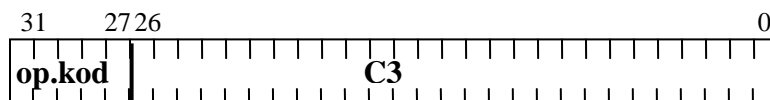
BRNZ R[ra], c2

Operacijski kod je 23

Naredba za relativno bezuvjetno grananje izvodi bezuvjetno grananje od trenutne vrijednosti programskog broja za vrijednost c3, odnosno:

$$PC = PC + c3$$

Imaju format:



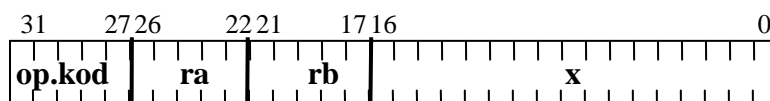
gdje je $-2^{26} \leq c3 < 2^{26}$
 Naredba za bezuvjetno relativno grananje
BR c3
 ima operacijski kod 24.

1.3.2 Naredbe za apsolutna grananja

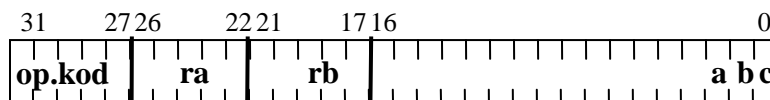
Naredbe za apsolutna uvjetna grananja uspoređuju sadržaj spremnika R[rb] s uvjetom i ukoliko je uvjet zadovoljen grana se na naredbu na adresu koja je upisana u spremnik R[ra], odnosno:

$$PC = R[ra]$$

Koristi se indirektni adresni mod budući kao i kod naredbi za prijenos podataka broj preostalih bita u naredbi nije dovoljan da se adresiraju sve memorijske lokacije. Mogući format naredbe je:



Svaka naredba dobiva svoj operacijski kod.
 Kako u naredbi ima 17 neiskorištenih bita, u cilju uštede operacijskih kodova moguće je sljedeće rješenje:



Sva apsolutna grananja imaju isti operacijski kod 25 te se tip grananja određuje na osnovu a, b i c bitova na način:

a	b	c		
0	0	0	bezuovjetno	BA
0	0	1	pozitivno	BAP
0	1	0	negativno	BAM
0	1	1	nula	BAZ
1	0	0	raz. nula	BANZ