

## Vježbe IX

### Kontrola procesorske jedinice

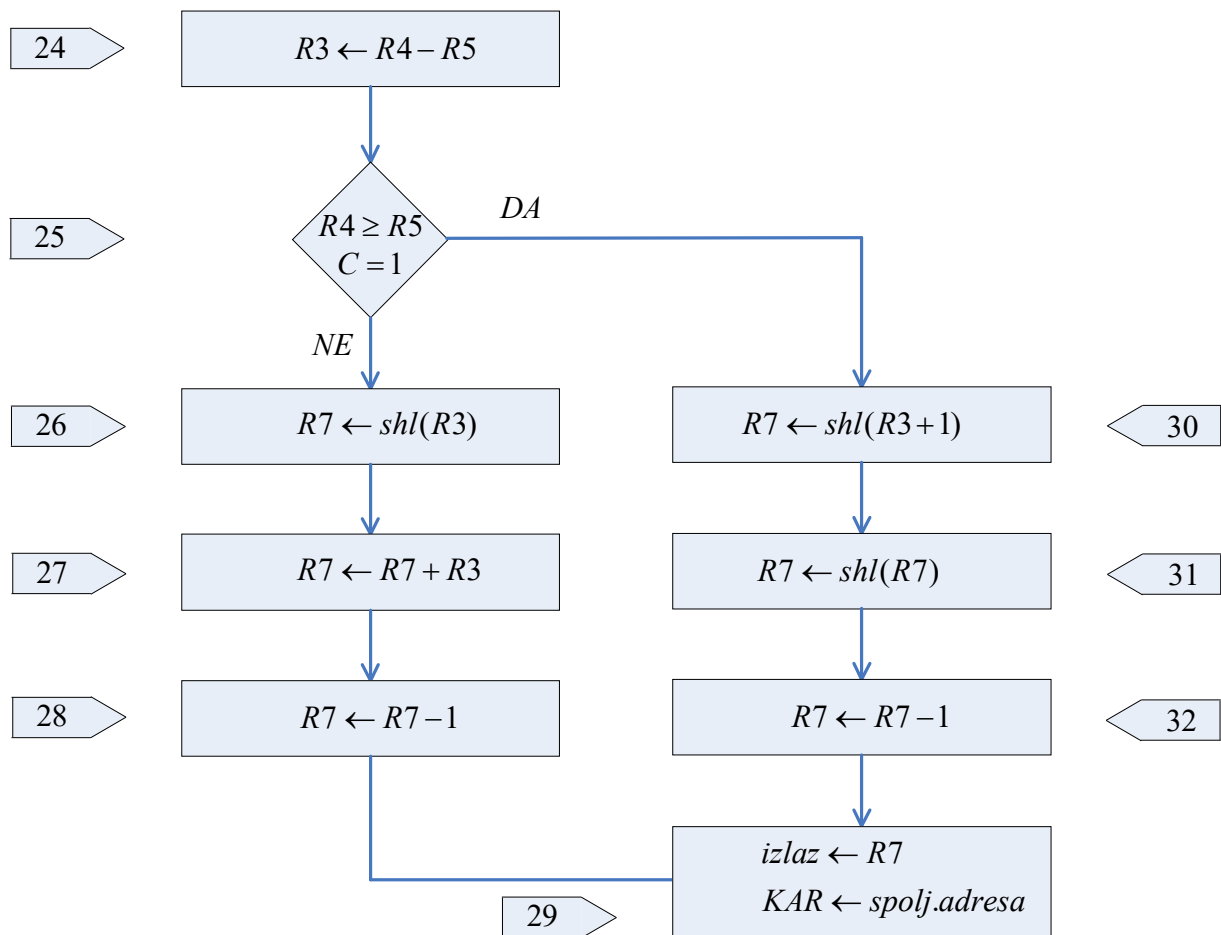
1. Napisati mikroprogram u simboličkom i mnemoničkom obliku kojim se realizuje slijedeća funkcija  $R7 \leftarrow f(R4 - R5)$ , pri čemu je:

$$f(x) = \begin{cases} 4x + 3, & x \geq 0 \\ 3x - 1, & x < 0 \end{cases}$$

Početna mikroinstrukcija se nalazi na lokaciji 24.

Imajući u vidu da nam je za ovaj slučaj  $x = R4 - R5$ , funkciju možemo zapisati:

$$R7 \leftarrow \begin{cases} 4(R4 - R5) + 3, & R4 - R5 \geq 0, R4 \geq R5 \\ 3(R4 - R5) - 1, & R4 - R5 < 0, R4 < R5 \end{cases}$$



Funkcija zavisi od promjenljive  $x$ , pri čemu je zadatkom dato da je  $x = R4 - R5$ . Smjestimo promjenljivu u pomoćni registar  $R3$ ,  $R3 \leftarrow R4 - R5$ .

Da bi crtali algoritam, neophodno je da ispitamo vrijednost promjenljive. Ukoliko je promjenljiva veća ili jednaka nuli ( $\geq$ ), funkcija uzima vrijednost  $4(R4 - R5) + 3$ ; ako je promjenljiva manja od nule ( $<$ ) funkcija je jednaka  $3(R4 - R5) - 1$ . Ispitivanje vrijednosti promjenljive se vrši korišćenjem bita  $C$ . Ukoliko je  $C=1$ , tada je  $R4 - R5 \geq 0$ , odnosno  $R4 \geq R5$ . Ako je, pak,  $C=0$  važi  $R4 - R5 < 0$ , odnosno  $R4 < R5$ .

Posmatrajmo „DA“ granu ( $C=1$ ). Funkcija je, u tom slučaju,  $4(R4 - R5) + 3$ . Imajući u vidu da je množenje sa 2 operacija shift left (*shl*), i da možemo istovremeno sabirati samo po jednu jedinicu (inkrementiranje), željenu vrijednost dobijamo tako što:

1. U izlazni registar R7 smjestimo  $shl(R3+1)$ , što predstavlja  $2((R4 - R5) + 1)$ .
2. U izlazni registar R7 smjestimo  $shl(R7)$ , što predstavlja  $2R7 = 2 * 2((R4 - R5) + 1) = 4(R4 - R5) + 4$ .
3. Kako nam je potrebna vrijednost  $4(R4 - R5) + 3$ , a u registru R7 imamo  $4(R4 - R5) + 4$ , to je potrebno izvršiti oduzimanje jedinice, odnosno  $R7 \leftarrow R7 - 1$ , što predstavlja  $4(R4 - R5) + 4 - 1 = 4(R4 - R5) + 3$ .

Kad smo u registru R7 dobili željenu vrijednost funkcije, prosljedimo je na izlaz,  $izlaz \leftarrow R7$ , a za KAR uzmemo spoljašnju adresu.

Posmatrajmo „NE“ granu (nije ispunjen uslov  $R4 - R5 \geq 0$ , što automatski znači da je  $R4 - R5 < 0$ ). Funkcija je u tom slučaju  $3(R4 - R5) - 1$ , i dobijamo je tako što:

1. U izlazni registar R7 smjestimo  $shl(R3)$ , što predstavlja  $2(R4 - R5)$ .
2. U izlazni registar R7 smjestimo  $R7 + R3$ , što je  $2(R4 - R5) + (R4 - R5) = 3(R4 - R5)$ .
3. Kako nam je potrebna vrijednost  $3(R4 - R5) - 1$ , a u registru R7 imamo  $3(R4 - R5)$ , to je potrebno izvršiti oduzimanje jedinice, odnosno  $R7 \leftarrow R7 - 1$ .

U registru R7 smo, opet, dobili željenu vrijednost funkcije, koju prosljedujemo na izlaz,  $izlaz \leftarrow R7$ , a za KAR uzmemo spoljašnju adresu.

Numerisanje lokacija se vrši redom, pri čemu se početna adresa (25 u ovom slučaju) dodjeljuje prvoj operaciji. Kad se naiđe na grananje u algoritmu, prvo se vrši numeracija „NE“ grane do kraja, nakon čega se vraćamo na „DA“ granu.

Mnemonički oblik programa se dobija prepisivanjem operacija iz algoritma. Pri tome se vodi računa o lokacijama, odnosno o tome da li se operacije izvršavaju uzastopno ( $KAR \leftarrow KAR + 1$ ) ili se prelazi na neku, tačno specificiranu lokaciju (npr.  $KAR \leftarrow 31$ ). Potrebno je voditi računa o operacijama koje uključuju ispitivanje vrijednosti određenih bita (bita C u ovom slučaju). Tada se odgovarajuća mikroinstrukcija piše u obliku **if** petlje, na slijedeći način:

**if** (uslov) **then** (adresa na koju se prelazi ako je uslov zadovoljen) **else** (adresa na koju se prelazi ako uslov nije zadovoljen)

### Simbolički oblik

<u>ADS</u>	<u>Mikroinstrukcije</u>
24	$R3 \leftarrow R4 - R5 ; KAR \leftarrow KAR + 1$
25	$if (C=1) then ( KAR \leftarrow 30 ) else ( KAR \leftarrow KAR + 1 )$
26	$R7 \leftarrow shl(R3) ; KAR \leftarrow KAR + 1$
27	$R7 \leftarrow R7 + R3 ; KAR \leftarrow KAR + 1$
28	$R7 \leftarrow R7 - 1 ; KAR \leftarrow KAR + 1$
29	$izlaz \leftarrow R7 ; KAR \leftarrow spolj.adresa$
30	$R7 \leftarrow shl(R3+1) ; KAR \leftarrow KAR + 1$
31	$R7 \leftarrow shl(R7) ; KAR \leftarrow KAR + 1$
32	$R7 \leftarrow R7 - 1 ; KAR \leftarrow 29$

Simbolički oblik mikroprograma podrazumijeva definisanje 8 polja:

**A i B:** polja prvog i drugog operanda (registri nad čijim sadržajem se vrše operacije; npr.: R3, R7... ili – ako nema operanda)

**D:** polje rezultata (registar u kojem se bilježi rezultat; npr. R4, R6... ili NONE ako nema rezultata)

**F:** polje aritmetičke funkcije; najčešće aritmetičke funkcije:

Sabiranje (sadržaja dva registra) ADD                      Oduzimanje broja 1 (dekrementiranje) DEC  
 Oduzimanje (sadržaja dva registra) SUB                    Komplementiranje COM  
 Sabiranje sa 1 (inkrementiranje) INC                      Transfer (prepisivanje vrijednosti) TSF

**H:** polje pomjeračke funkcije; najčešće pomjeračke funkcije:

Množenje sa 2 (shift left) SHL                              Nema pomjeračke funkcije (no shift) NHS  
 Dijeljenje sa 2 (shift right) SHR                            Upisivanje nule u registar ZERO

**MUX1:** Interna adresa (poznata tačna lokacija) INT, Spoljašnja adresa EXT, u ostalim slučajevima –

**MUX2:** Prelazak na susjednu lokaciju NEXT, Prelazak pod uslovom da je C=1 LC, Bezulsovni prelazak LAD

**Adresa:** Brojačana vrijednost ako je poznata, u suprotnom –

**NAPOMENA:** Navedene su vrijednosti koje se najčešće srijeću; kompletne tabele za svako od polja se nalaze u skripti!

**Mnemonički oblik:**

<u>KAR</u>	<u>A</u>	<u>B</u>	<u>D</u>	<u>F</u>	<u>H</u>	<u>MUX1</u>	<u>MUX2</u>	<u>Adresa</u>
24	R4	R5	R3	SUB	NSH	-	NEXT	-
25	-	-	NONE	TSF	NSH	INT	LC	30
26	R3	-	R7	TSF	SHL	-	NEXT	-
27	R7	R3	R7	ADD	NSH	-	NEXT	-
28	R7	-	R7	DEC	NSH	-	NEXT	-
29	R7	-	NONE	TSF	NSH	EXT	LAD	-
30	R3	-	R7	INC	SHL	-	NEXT	-
31	R7	-	R7	TSF	SHL	-	NEXT	-
32	R7	-	R7	DEC	NSH	INT	LAD	29

2. Napisati mikroprogram u simboličkom i mnemoničkom obliku, kojim se realizuje slijedeća funkcija  $R1 \leftarrow f(R3 - R5)$ , pri čemu je  $f(x) = |x|$ . Početna mikroinstrukcija je na lokaciji 45.

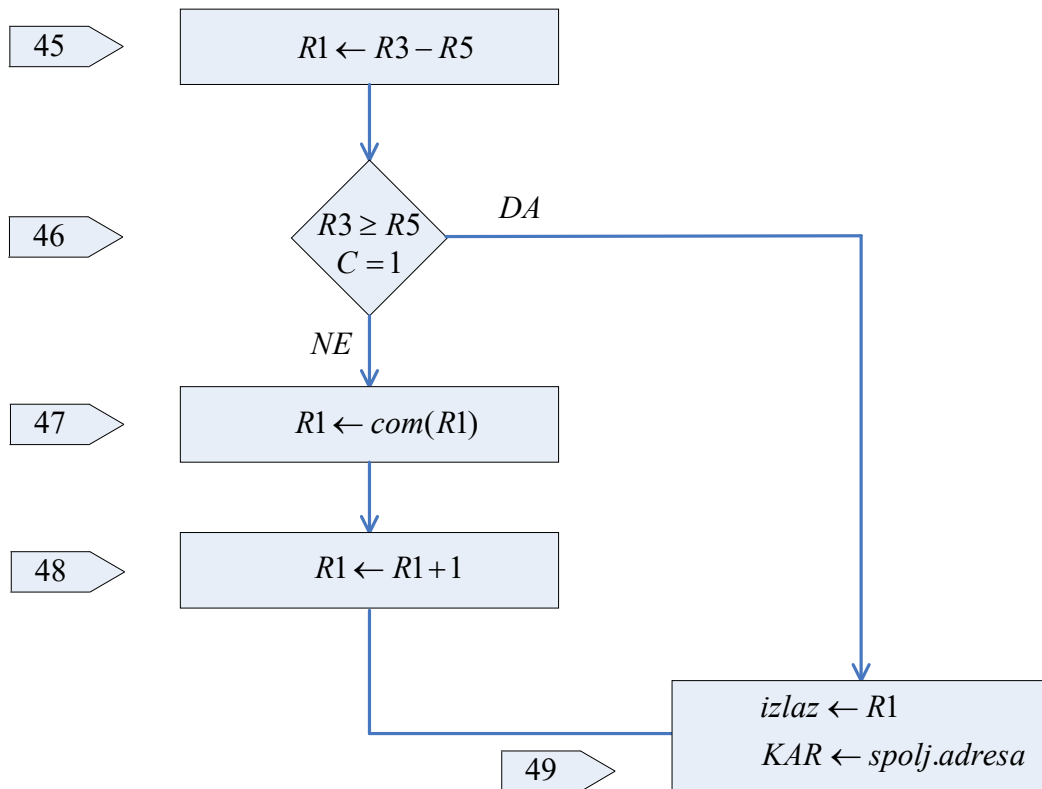
Znajući da je  $f(x) = |x| = \begin{cases} x, & x \geq 0 \\ -x, & x < 0 \end{cases}$ , zaključujemo:  $R1 \leftarrow \begin{cases} (R3 - R5), & R3 - R5 \geq 0, R3 \geq R5 \\ -(R3 - R5), & R3 - R5 < 0, R3 < R5 \end{cases}$

Funkcija zavisi od promjenljive  $x$ , koja je u ovom slučaju  $x = R3 - R5$ . Smjestimo promjenljivu u pomoćni registar  $R1$ ,  $R1 \leftarrow R3 - R5$ .

Ukoliko je promjenljiva veća ili jednaka nuli ( $\geq$ ) funkcija uzima vrijednost  $(R3 - R5)$ ; ako je promjenljiva manja od nule ( $<$ ) funkcija je jednaka  $-(R3 - R5)$ . Ispitivanje vrijednosti promjenljive se vrši korišćenjem bita C. Ukoliko je C=1, tada je  $R3 - R5 \geq 0$ , odnosno  $R3 \geq R5$ . Ako je, pak, C=0 važi  $R3 - R5 < 0$ , odnosno  $R3 < R5$ .

Posmatrajmo „DA“ granu (C=1). Funkcija je u tom slučaju jednaka promjenljivoj, te je potrebno samo prosljediti promjenljivu na izlaz.

Posmatrajmo „NE“ granu (nije ispunjen uslov  $R3 - R5 \geq 0$ , što automatski znači da je  $R3 - R5 < 0$ ). Tada je na izlaz potrebno prosljediti negativnu vrijednost promjenljive. Negativan broj predstavlja zapravo dvojni komplement originalne vrijednosti (jedinični komplement plus 1).



**Simbolički oblik:**

<u>ADS</u>	<u>Mikroinstrukcije</u>
45	$R1 \leftarrow R3 - R5; KAR \leftarrow KAR + 1$
46	if (C=1) then ( $KAR \leftarrow 49$ ) else ( $KAR \leftarrow KAR + 1$ )
47	$R1 \leftarrow com(R1); KAR \leftarrow KAR + 1$
48	$R1 \leftarrow R1 + 1; KAR \leftarrow KAR + 1$
49	$izlaz \leftarrow R1; KAR \leftarrow spolj.adresa$

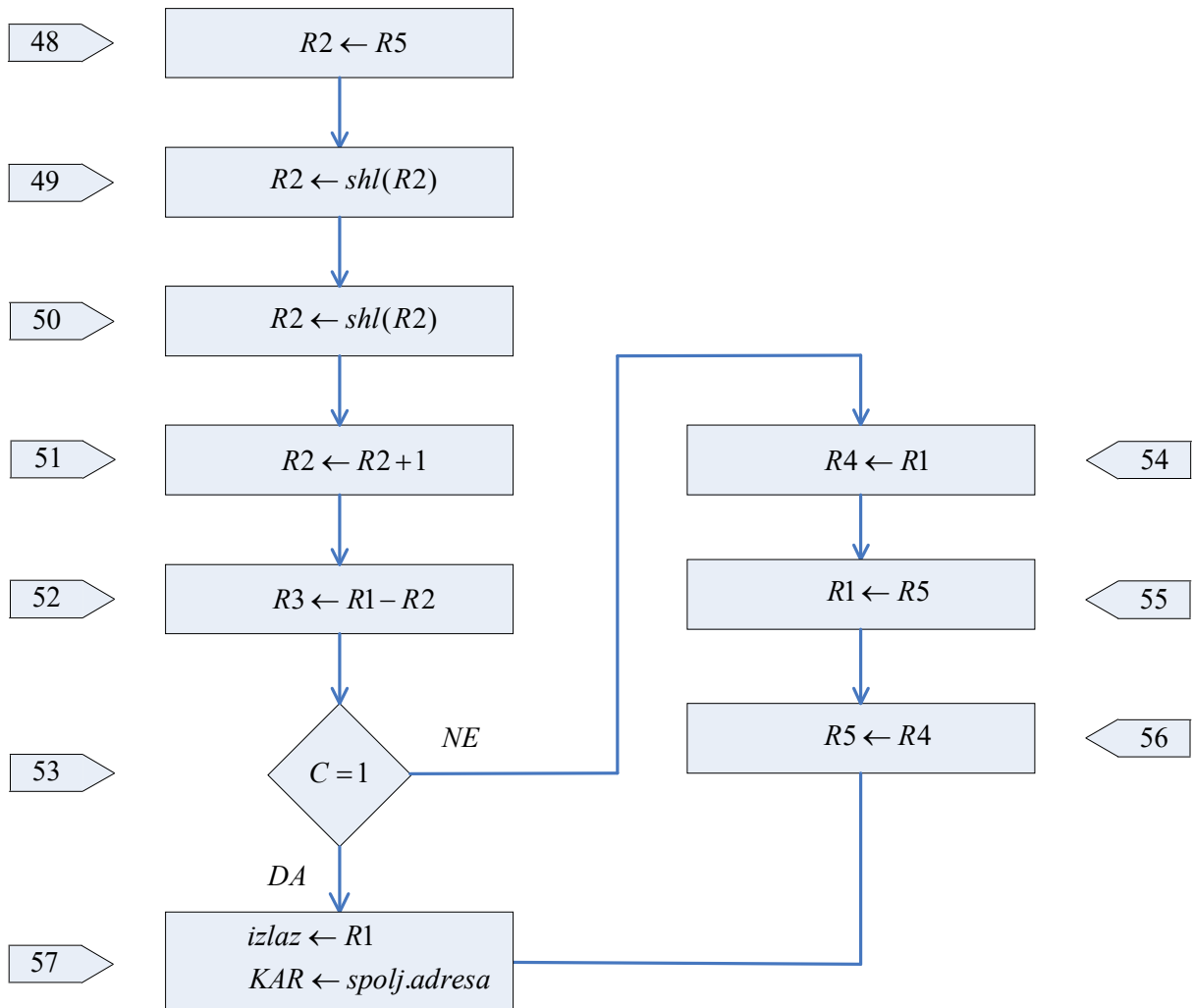
**Mnemonički oblik:**

<u>KAR</u>	<u>A</u>	<u>B</u>	<u>D</u>	<u>F</u>	<u>H</u>	<u>MUX1</u>	<u>MUX2</u>	<u>Adresa</u>
45	R3	R5	R1	SUB	NSH	-	NEXT	-
46	-	-	NONE	TSF	NSH	INT	LC	49
47	R1	-	R1	COM	NSH	-	NEXT	-
48	R1	-	R1	INC	NSH	-	NEXT	-
49	R1	-	NONE	TSF	NSH	EXT	LAD	-

3. Napisati program u simboličkom i mnemoničkom obliku kojim se realizuje slijedeća operacija: u registrima R1 i R5 su smješteni cijeli brojevi; ukoliko je ispunjen uslov da je  $R1 \geq 4R5 + 1$ , brojevi ostaju na prvobitnim pozicijama, u suprotnom mijenjaju mjesta. Početna mikroinstrukcija je na lokaciji 48.

Da bismo mogli utvrditi da li je zadovoljena nejednakost  $R1 \geq 4R5 + 1$ , potrebno je najprije vrijednost  $4R5 + 1$  smjestiti u pomoćni registar. To, očigledno, ne možemo obaviti u jednom koraku, već:

1. U pomoćni (proizvoljni) registar  $R2$  smještamo vrijednost  $R5$ .
2. U  $R2$  smještamo  $shl(R2)$  što predstavlja  $2 R2=2 R5$ .
3. U  $R2$  smještamo  $shl(R2)$  što predstavlja  $2 R2=2(2 R5)=4R5$ .
4. U  $R2$  smještamo  $R2+1$ , što predstavlja  $R2+1=4R5+1$ .

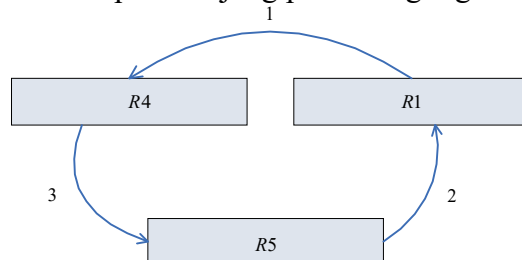


Na taj način smo u registar  $R2$  smjestili vrijednost  $4R5+1$ , koju dalje poredimo sa vrijednošću registra  $R1$ .

Ukoliko je zadovoljena nejednakost  $R1 \geq 4R5+1$ , odnosno  $C=1$ , vrijednosti registara se ne mijenjaju, te jednostavno prosljeđujemo vrijednost registra  $R1$  (ili  $R5$ , pošto nije precizno navedeno u zadatku) na izlaz.

Ukoliko, pak, nejednakost nije zadovoljena, potrebno je da vrijednosti registara  $R1$  i  $R5$  zamjene mjesta. Ako bismo direktno uputili  $R1 \leftarrow R5$  postigli bismo da se vrijednost registra  $R5$  nađe u registru  $R1$ , ali bismo izgubili informaciju o stvarnoj vrijednosti registra  $R1$  koju treba da smjestimo u registar  $R5$ . Slično bi se dogodilo ukoliko bismo uradili slijedeće:  $R5 \leftarrow R1$ . Zaključujemo da na ovaj način ne možemo izvršiti zamjenu vrijednosti koje se čuvaju u registrima.

Zamjenu ćemo postići upotrebom proizvoljnog pomoćnog registra, na slijedeći način:



1. U pomoćni registar smjestimo vrijednost registra R1.
2. U registar R1 smjestimo vrijednost registra R5.
3. U registar R5 smjestimo sadržaj pomoćnog registra koji čuva vrijednost registra R1.

Na taj način postizemo da se u registru R1 nalazi vrijednost registra R5 i obrnuto.

Sada možemo pisati mnemonički, odnosno simbolički oblik odgovarajućeg mikroprograma:

### Simbolički oblik:

<u>ADS</u>	<u>Mikroinstrukcije</u>
48	$R2 \leftarrow R5 ; KAR \leftarrow KAR + 1$
49	$R2 \leftarrow shl(R2) ; KAR \leftarrow KAR + 1$
50	$R2 \leftarrow shl(R2) ; KAR \leftarrow KAR + 1$
51	$R2 \leftarrow R2 + 1 ; KAR \leftarrow KAR + 1$
52	$R3 \leftarrow R1 - R2 ; KAR \leftarrow KAR + 1$
53	if (C=1) then ( $KAR \leftarrow 57$ ) else ( $KAR \leftarrow KAR + 1$ )
54	$R4 \leftarrow R1 ; KAR \leftarrow KAR + 1$
55	$R1 \leftarrow R5 ; KAR \leftarrow KAR + 1$
56	$R5 \leftarrow R4 ; KAR \leftarrow KAR + 1$
57	$izlaz \leftarrow R1 ; KAR \leftarrow spolj.adresa$

### Mnemonički oblik:

<u>KAR</u>	<u>A</u>	<u>B</u>	<u>D</u>	<u>F</u>	<u>H</u>	<u>MUX1</u>	<u>MUX2</u>	<u>Adresa</u>
48	R5	-	R2	TSF	NSH	-	NEXT	-
49	R2	-	R2	TSF	SHL	-	NEXT	-
50	R2	-	R2	TSF	SHL	-	NEXT	-
51	R2	-	R2	INC	NSH	-	NEXT	-
52	R1	R2	R3	SUB	NSH	-	NEXT	-
53	-	-	NONE	TSF	NSH	INT	LC	57
54	R1	-	R4	TSF	NSH	-	NEXT	-
55	R5	-	R1	TSF	NSH	-	NEXT	-
56	R4	-	R5	TSF	NSH	-	NEXT	-
57	R1	-	NONE	TSF	NSH	EXT	LAD	-

4. Napisati mikroprogram u simboličkom i mnemoničkom obliku, kojim se realizuje slijedeća operacija:

- ako je broj koji se nalazi u registru R2 paran, izračunati broj nula binarne riječi koja je smještena u registru R2;

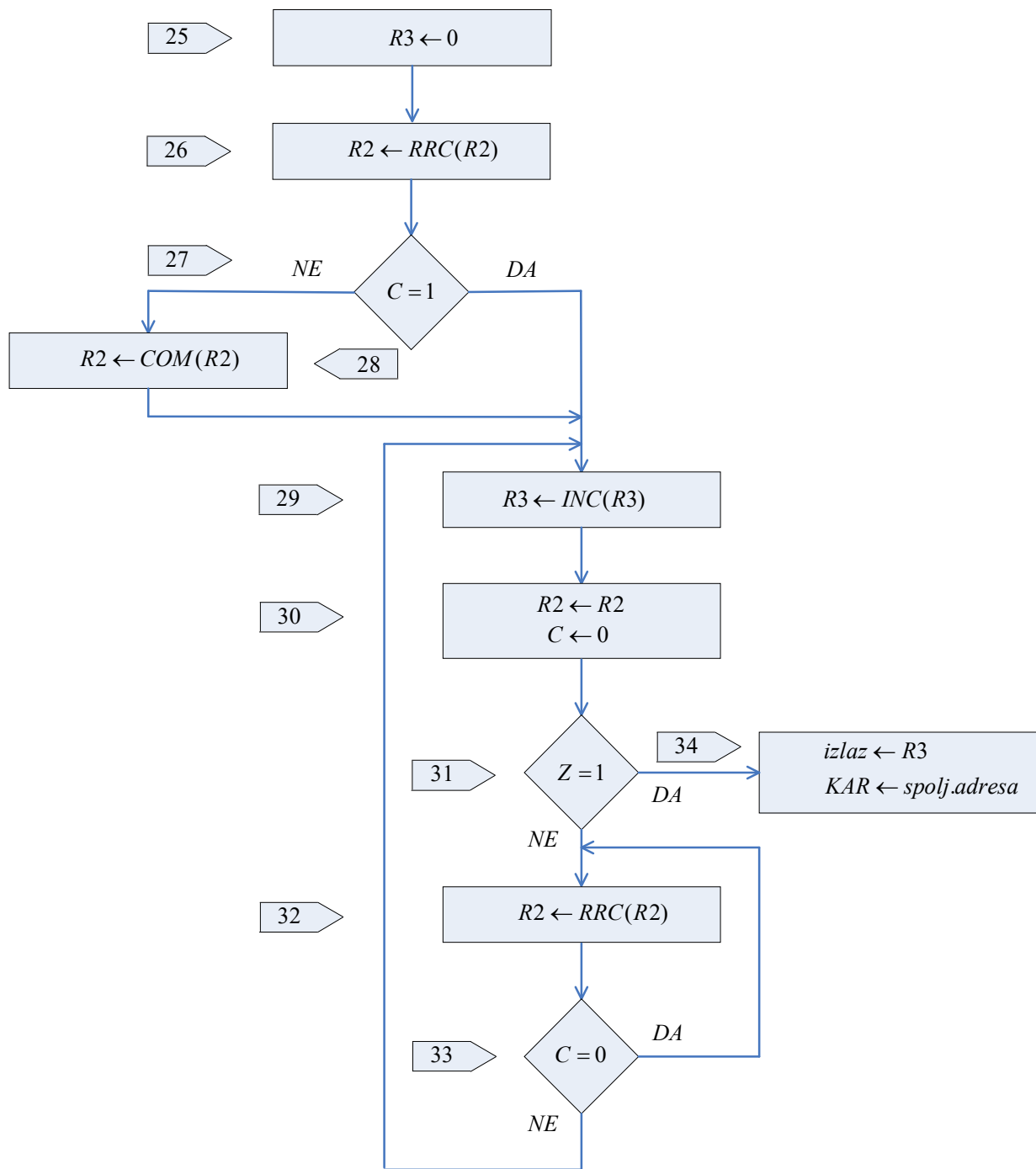
- ako je broj koji se nalazi u registru R2 neparan, izračunati broj jedinica binarne riječi koja je smještena u registru R2.

Dobijeni rezultat smjestiti u registar R3. Početna mikroinstrukcija je na lokaciji 25.

RRC (*rotate right through carry*) je logička operacija kojom se rotira sadržaj registra za po jedno mjesto u desno, pri čemu se bit koji je ispao iz registra zadržava u C (*carry*).

Ovu operaciju možemo iskoristiti za određivanje parnosti broja. Naime, ukoliko je u registru smješten neparan broj posljednja cifra će biti 1 ( $2^0=1$ ; ostali biti su stepeni dvojke, pa ukoliko su sabrani sa jedinicom daće neparan broj); u suprotnom će biti paran. Dakle, prilikom prvog korišćenja operacije RRC određujemo parnost broja.

Za slučaj neparne vrijednosti potrebno je prebrojati koliko jedinica sadrži binarna riječ. Koristeći RRC koja u svakom taktu izbacuje po jedan bit u C, provjeravamo da li je taj broj 1 ili 0, te brojati, odnosno inkrementirati sadržaj registra koji koristimo kao brojač.



**Simbolički oblik:**

<b>ADS</b>	<b>Mikroinstrukcije</b>
25	$R3 \leftarrow 0; KAR \leftarrow KAR + 1$
26	$R2 \leftarrow RRC(R2); KAR \leftarrow KAR + 1$
27	if (C=1) then ( $KAR \leftarrow 29$ ) else ( $KAR \leftarrow KAR + 1$ )
28	$R2 \leftarrow COM(R2); KAR \leftarrow KAR + 1$
29	$R3 \leftarrow INC(R3); KAR \leftarrow KAR + 1$
30	$R2 \leftarrow R2; C \leftarrow 0; KAR \leftarrow KAR + 1$
31	if (Z=1) then ( $KAR \leftarrow 34$ ) else ( $KAR \leftarrow KAR + 1$ )
32	$R2 \leftarrow RRC(R2); KAR \leftarrow KAR + 1$
33	if (C=0) then ( $KAR \leftarrow 32$ ) else ( $KAR \leftarrow 29$ )
34	$izlaz \leftarrow R3; KAR \leftarrow spolj.adresa$

Da bi problem brojanja nula za slučaj parnog broja sveli na brojanje jedinica, dovoljno je komplementirati broj za koji smo utvrdili da je paran i brojati njegove jedinice (kojih je tačno onoliko koliko je nula u originalnoj vrijednosti broja).

Nakon svakog rotiranja broja potrebno je provjeriti da li se registar ispraznio, odnosno da li su u registru preostale samo nule. Ukoliko to jeste slučaj brojanje je završeno. U suprotnom kompletna proceduda brojanja se ponavlja.

### Mnemonički oblik:

<u>KAR</u>	<u>A</u>	<u>B</u>	<u>D</u>	<u>F</u>	<u>H</u>	<u>MUX1</u>	<u>MUX2</u>	<u>Adresa</u>
25	-	-	R3	TSF	ZERO	-	NEXT	-
26	R2	-	R2	TSF	RRC	-	NEXT	-
27	-	-	NONE	TSF	NSH	INT	LC	29
28	R2	-	R2	COM	NSH	-	NEXT	-
29	R3	-	R3	INC	NSH	-	NEXT	-
30	R2	-	R2	TSF	NSH	-	NEXT	-
31	-	-	NONE	TSF	NSH	INT	LZ	34
32	R2	-	R2	TSF	RRC	-	NEXT	-
33	-	-	NONE	TSF	NSH	INT	LNC	32
34	R3	-	NONE	TSF	NSH	EXT	LAD	-

5. Napisati mikroprogram u simboličkom i mnemoničkom obliku, kojim se realizuje funkcija  $R4 \leftarrow f(R1 - R5)$ , pri čemu je:

$$f(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & 0 < x < 1 \\ x, & x \geq 1 \end{cases}$$

Početna mikroinstrukcija je na lokaciji 25.

$$\text{To znači da je: } f(R1 - R5) = \begin{cases} -1, & R1 - R5 < 0, \quad R1 < R5 \\ 0, & R1 - R5 = 0, \quad R1 = R5 \\ 1, & 0 < R1 - R5 < 1 \\ R1 - R5, & R1 - R5 > 1 \end{cases}$$

Promjenljivu smjestimo u pomoćni registar (ovdje smo izabrali izlazni registar, jer funkcija ne zahtijeva veće transformacije promjenljivih; ipak, to nije obavezno, moguće je izabrati proizvoljni registar, pa na kraju algoritma rezultat premjestiti dobijeni rezultat u izlazni registar) i počeli sa ispitivanjem vrijednosti promjenljive.

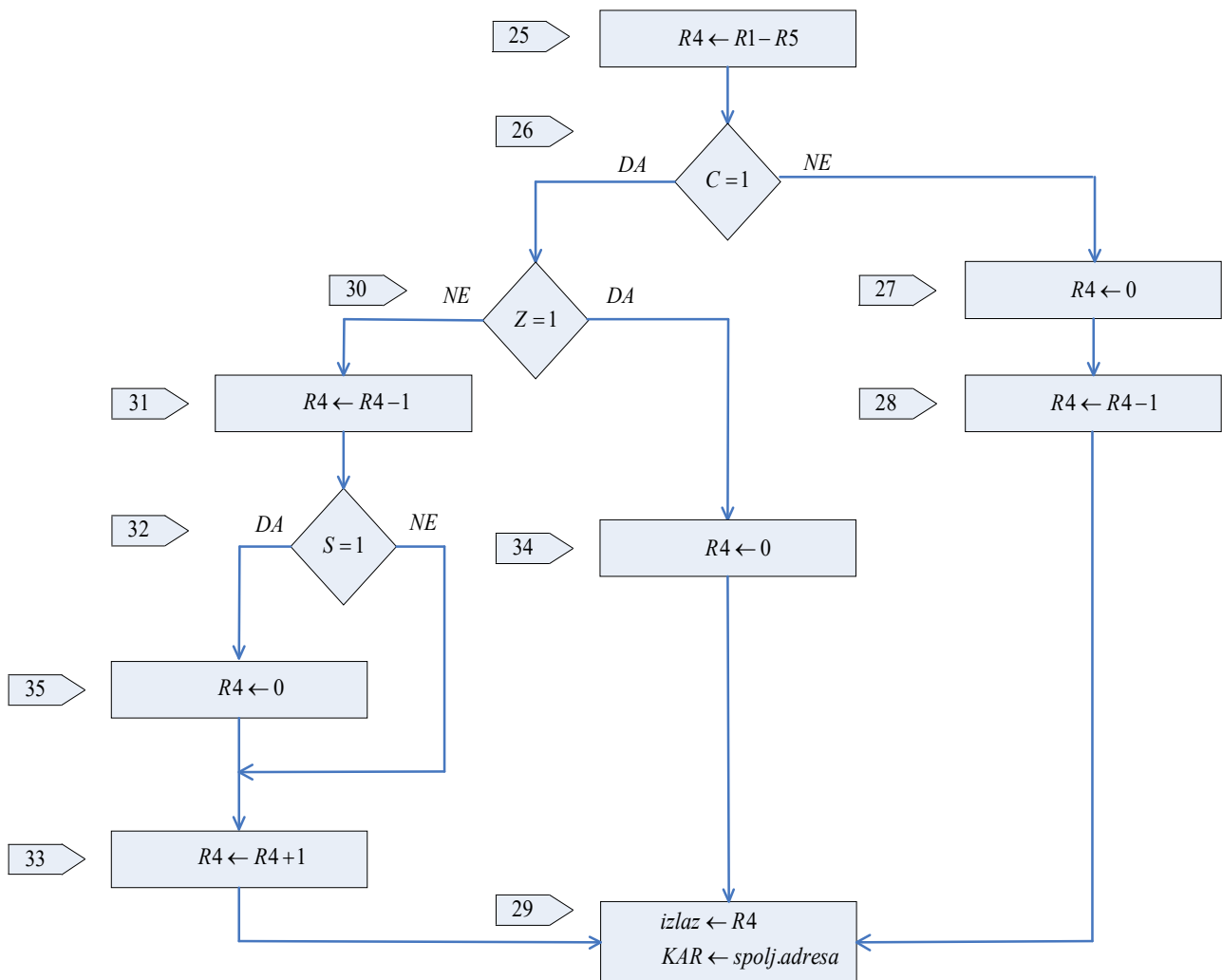
Naprije pomoću C bita utvrđujemo da li je promjenljiva  $\geq 0$ , odnosno  $< 0$ . Za slučaj da je manja od nule, na izlaz prosljeđujemo, kao što je definisano zadatkom, broj -1. Ovu vrijednost dobijamo tako što registar postavimo na 0 u prvom koraku, a zatim ga dekrementiramo.

Ukoliko je promjenljiva  $\geq 0$  razdvajamo slučajeve da je  $> 0$  i da je  $= 0$ , ispitivanjem bita Z. Za slučaj da je promjenljiva  $= 0$  na izlaz prosljeđujemo 0, kao što je određeno samom funkcijom.

Ukoliko je promjenljiva  $> 0$ , moramo ispitati da li je veća i od 1, ili je u osegu od 0 do 1. To ćemo uraditi tako što ćemo od promjenljive oduzeti broj 1, i ispitati da li smo dobili negativan broj pomoću S bita. Ako je broj negativan to znači da je veći od 0 i manji od 1, pa funkcija uzima vrijednost 1, koju dobijamo tako što u registar smjestimo 0 koju u sledećem koraku



inkrementiramo. U suprotnom broj je veći i od 0 i od 1, pa je potrebno prosljediti samo vrijednost promjenljive. Vodimo računa o tome da smo promjenljivu umanjili za 1 kako bi ispitali njen opseg, pa je potrebno sabrati sa tom jedinicom kako bi se očuvala njena originalna vrijednost.



**Simbolički oblik:**

<b>ADS</b>	<b>Mikroinstrukcije</b>
25	$R4 \leftarrow R1 - R5; KAR \leftarrow KAR + 1$
26	if (C=1) then ( $KAR \leftarrow 30$ ) else ( $KAR \leftarrow KAR + 1$ )
27	$R4 \leftarrow 0; KAR \leftarrow KAR + 1$
28	$R4 \leftarrow R4 - 1; KAR \leftarrow KAR + 1$
29	$izlaz \leftarrow R4; KAR \leftarrow spolj.adresa$
30	if (Z=1) then ( $KAR \leftarrow 34$ ) else ( $KAR \leftarrow KAR + 1$ )
31	$R4 \leftarrow R4 - 1; KAR \leftarrow KAR + 1$
32	if (S=1) then ( $KAR \leftarrow 35$ ) else ( $KAR \leftarrow KAR + 1$ )
33	$R4 \leftarrow R4 + 1; KAR \leftarrow 29$
34	$R4 \leftarrow 0; KAR \leftarrow 29$
35	$R4 \leftarrow 0; KAR \leftarrow 33$

**Mnemonički oblik:**

<u>KAR</u>	<u>A</u>	<u>B</u>	<u>D</u>	<u>F</u>	<u>H</u>	<u>MUX1</u>	<u>MUX2</u>	<u>Adresa</u>
25	R1	R5	R4	SUB	NSH	-	NEXT	-
26	-	-	NONE	TSF	NSH	INT	LC	30
27	-	-	R4	TSF	ZERO	-	NEXT	-
28	R4	-	R4	DEC	NSH	-	NEXT	-
29	R4	-	NONE	TSF	NSH	EXT	LAD	-
30	-	-	NONE	TSF	NSH	INT	LZ	34
31	R4	-	R4	DEC	NSH	-	NEXT	-
32	-	-	NONE	TSF	NSH	INT	LS	35
33	R4	-	R4	INC	NSH	INT	LAD	29
34	-	-	R4	TSF	ZERO	INT	LAD	29
35	-	-	R4	TSF	ZERO	INT	LAD	33