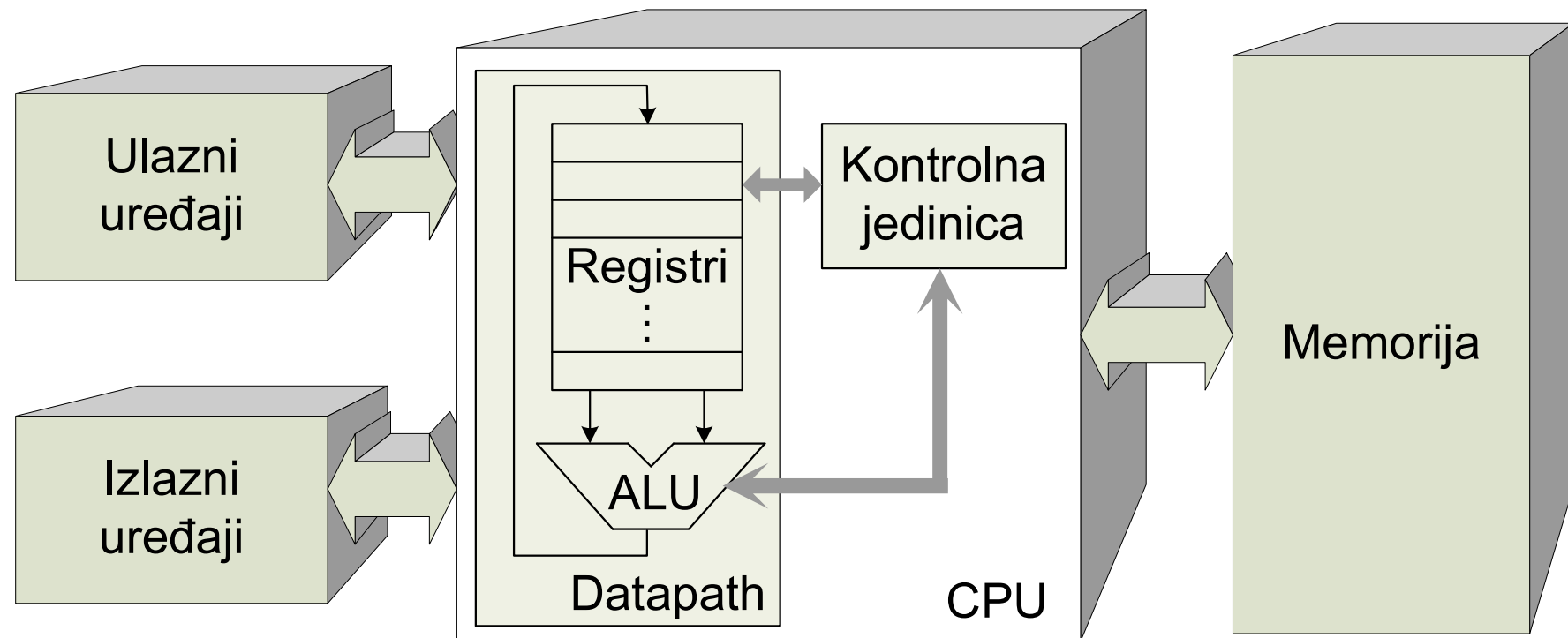




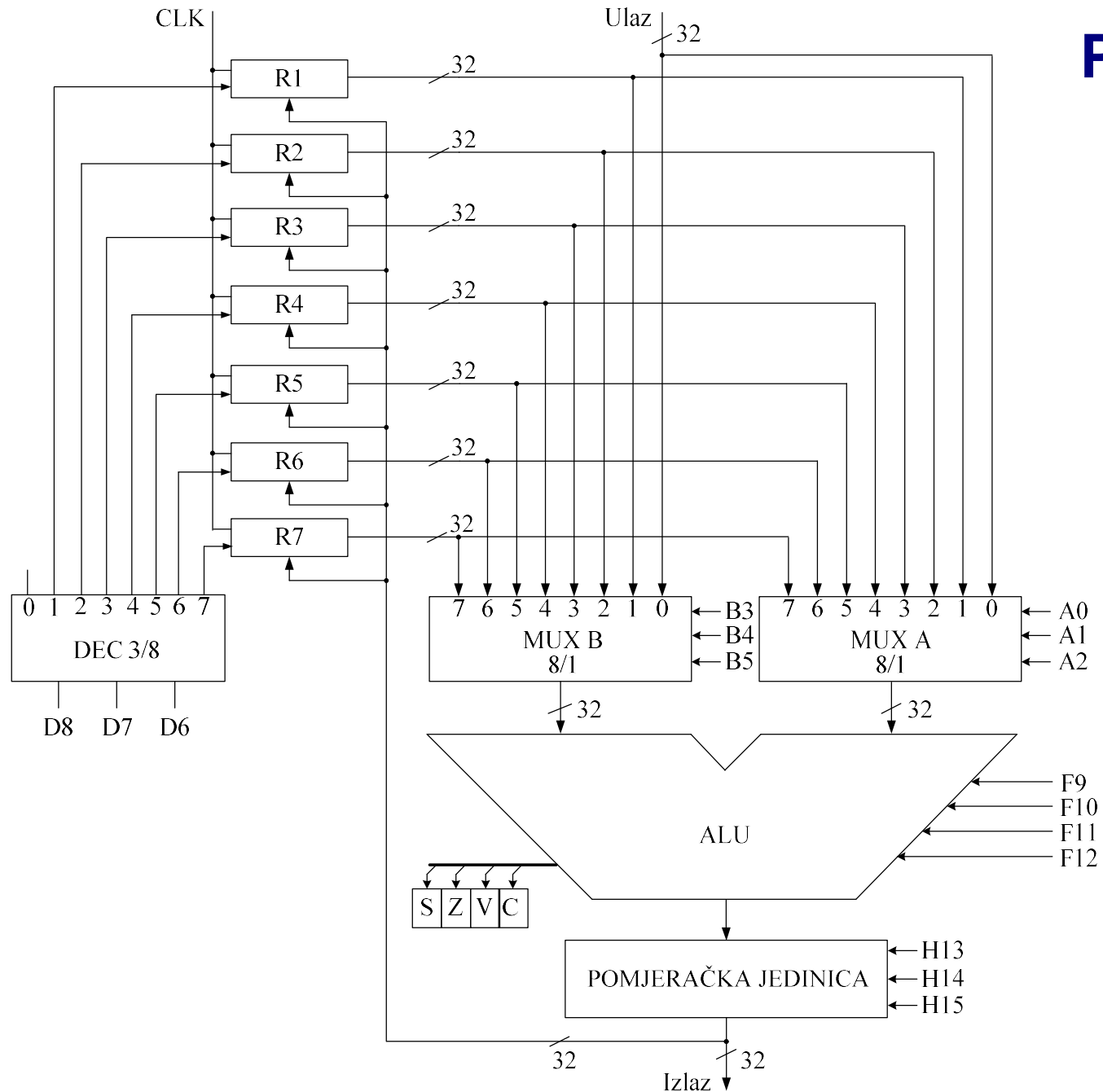
# Osnovi računarstva I

**Centralna procesorska jedinica**  
**II dio: Kontrolna jedinica**

## Podsjećanje:



# Podsjećanje: Procesorska jedinica (Datapath)



# Podsjećanje:

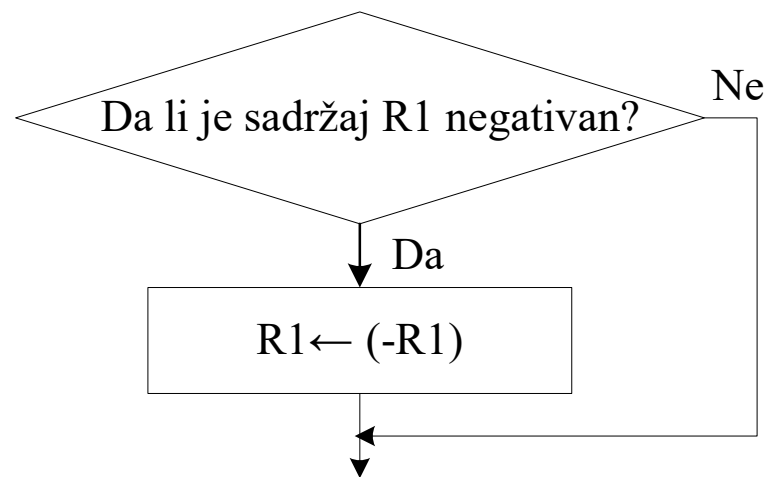
## Kontrolna riječ

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A			B			D			F			H			

Bin. kod	A	B	D	F		H
				$C_{in}=0$	$C_{in}=1$	
000	Ulaz	Ulaz	Izlaz	$F=A$	$F=A+1$	$S \leftarrow F$
001	R1	R1	R1	$F=A+B$	$F=A+B+1$	$S \leftarrow \text{shl}(F)$
010	R2	R2	R2	$F=A + \bar{B}$	$F=A-B$	$S \leftarrow \text{shr}(F)$
011	R3	R3	R3	$F=A-1$	$F=A$	$S \leftarrow 0$
100	R4	R4	R4	$F=A \wedge B$	×	$S \leftarrow \text{rlc}(F)$
101	R5	R5	R5	$F=A \vee B$	×	$S \leftarrow \text{rol}(F)$
110	R6	R6	R6	$F=A \oplus B$	×	$S \leftarrow \text{ror}(F)$
111	R7	R7	R7	$F=\bar{A}$	×	$S \leftarrow \text{rrc}(F)$

# Kontrolna jedinica

- Osim izvršavanja pojedinačnih operacija u ALU, potrebno je uvesti i kontrolu toka (redoslijeda) kojim se pojedinačne operacije izvršavaju
- **Primjer:** U reg. R1 upisati apsolutnu vrijednost njegovog ranijeg sadržaja.
- **Rješenje: Algoritamski**

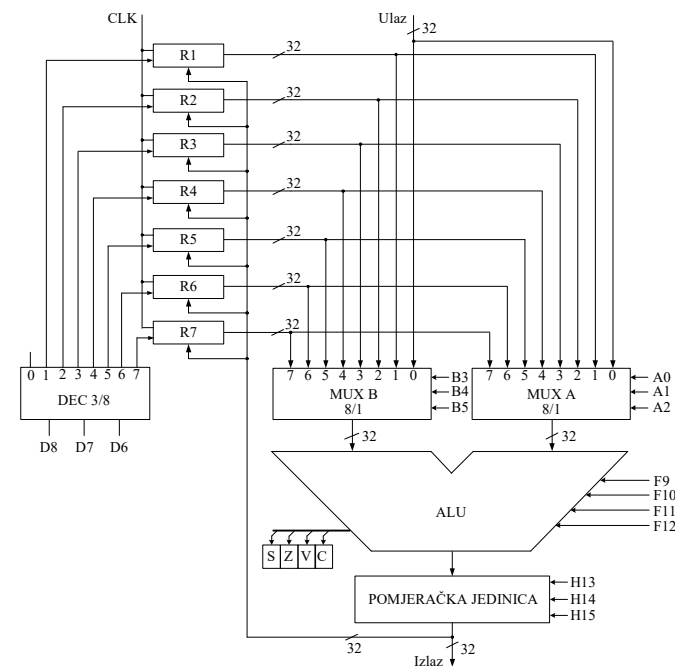


Neophodni koraci u cilju rješavanja problema:

- 1) Utvrđivanje da li je sadržaj registra R1 negativan broj
- 2) Ako jeste, konvertovati ga u pozitivan broj, u suprotnom – ne raditi ništa

# Kontrolna jedinica

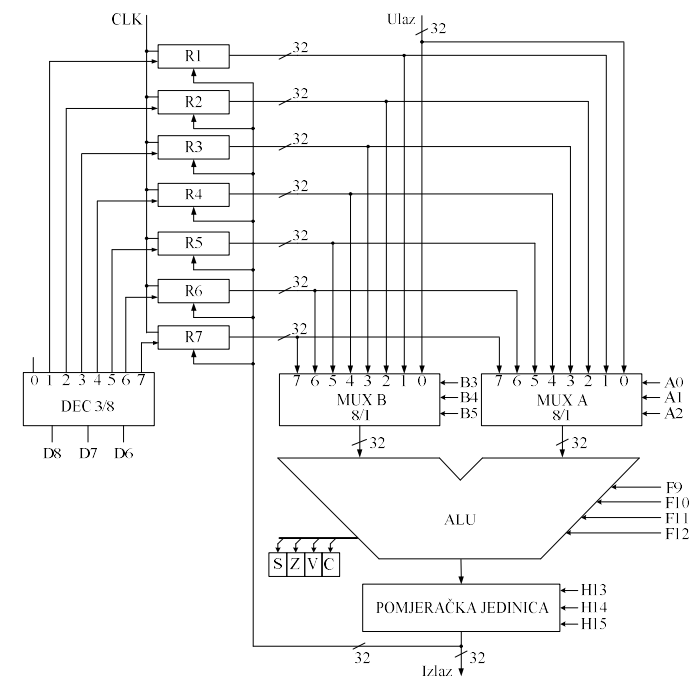
- **Provjera znaka podatka** sadržanog u registru R1 može se izvršiti uz pomoć **statusnog bita S** procesorske jedinice
- Ukoliko se, recimo, u ALU izvrši operacija sabiranja pri čemu se na oba njena ulaza dovede sadržaj reg. R1, rezultat će biti  $2R1$
- Ako statusni bit S, nakon izvršavanja ove operacije, uzme vrijednost logičke jedinice, rezultat sabiranja  $2R1$  (a time i podatak u R1) je negativan (pod uslovom da nije došlo do overflow-a)



Polje	A	B	D	F	H	Mikrooperacija
Simbolički zapis	R1	R1	Izlaz	$A+B$	$S \leftarrow F$	$Izlaz \leftarrow R1 + R1$
Binarni zapis	001	001	000	0010	000	

## Kontrolna jedinica

- Ukoliko je sadržaj registra R1 negativan, potrebno je izvršiti operaciju  $R1 \leftarrow (-R1)$
- $R1 \leftarrow (-R1)$  se može izvršiti u 2 koraka:
  - 1) komplementira se sadržaj registra R1 (u cilju kreiranja 1-nog komplementa (R1)),
  - 2) inkrementira se novi sadržaj registra R1 (u cilju kreiranja 2-nog komplementa (R1))



Mikrooperacija	Polje	A	B	D	F	H
$R1 \leftarrow \text{COM}(R1)$	Simbolički zapis	R1	×	R1	$F = \text{COM}(A)$	$S \leftarrow F$
	Binarni zapis	001	000	001	1110	000
$R1 \leftarrow R1 + 1$	Simbolički zapis	R1	×	R1	$F = A + 1$	$S \leftarrow F$
	Binarni zapis	001	000	001	0001	000

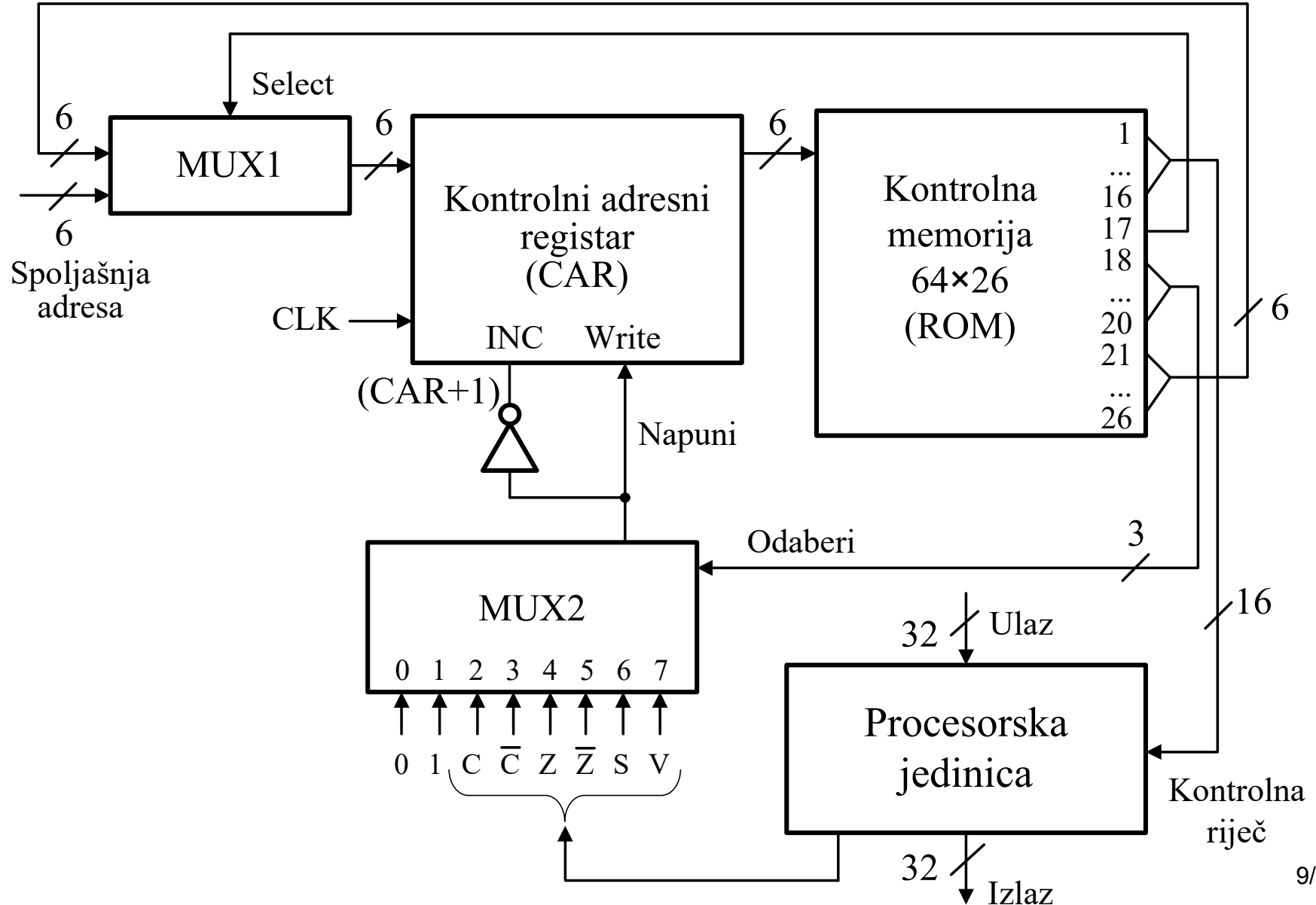


## Kontrolna jedinica

- Pristup rješavanju prethodnog problema asocira na izvršavanje programa
- Ipak, ovdje nije riječ o programu koji izvršava CPU, već o skupu akcija čiji način sprovođenja veoma podsjeća na programski kôd
- **Ovim skupom akcija (mikrooperacija) zapravo se vrši kontrola rada procesorske jedinice**
  - ⇒ **Mikroprogramska kontrolna jedinica**, a skup mikrooperacija koji ona sadrži – *mikroprogram*
- **Neophodno je proširiti kontrolnu riječ u cilju uvođenja kontrole toka (redoslijeda) izvršavanja mikrooperacija**
- **Proširena kontrolna riječ**, koja osim kontrole izvršavanja mikrooperacija omogućava i kontrolu toka izvršavanja sljedećih mikrooperacija (zadatih kontrolnim riječima), naziva se *mikroinstrukcija*



# Kontrolna jedinica



## Podjela mikroinstrukcija i njihova implementacija

Tip instrukcije	Opis implementacije	Write kontrolni signal CAR-a (izlaz MUX 2)	Ulaz CAR-a (izlaz MUX 1)	Interna ads. (za upis u CAR)
Sekvencijalne	Onemogućiti upis adrese, selektirane kroz MUX 1, u CAR, već $CAR \leftarrow CAR+1$	0	Bilo što	Bilo što
Bezuslovni skok	Upisati adresu, selektiranu kroz MUX 1, u CAR	1	0 (interna ads.)	Address
			1 (eksterna ads.)	Bilo što
Uslovni skok	Upisati adresu, selektiranu kroz MUX 1, u CAR, ali samo ukoliko je zadovoljen uslov grananja zadat vrijednošću statusnih bitova S, Z, V, C	C, Z, S, V, $inv(C)$ , $inv(Z)$	Interna adresa	Address



## Format mikroinstrukcije

Bitovi	Polje	Simboli	Funkcija
1 – 3	A	INP, R1 – R7	Prvi ulaz ALU
4 – 6	B	INP, R1 – R7	Drugi ulaz ALU
7 – 9	D	NONE, R1 – R7	Odredište rezultata sa izlaza datapath-a
10 – 13	F	Tabela 7.9	Operacija ALU
14 – 16	H	Tabela 7.10	Operacija pomjeračke jedinice
17	MUX1	INT, EXT	Interna (0) ili eksterna (1) adresa
18 – 20	MUX2	Tabela 7.11	Postavljanje Write kontrolnog ulaza CAR-a
21 – 26	ADS	Adresa	Adresa sljedeće mikroinstrukcije

## Najvažnije operacije ALU (polje F)

Tabela 7.9 Najvažnije operacije ALU (polje F)

Mnemonik	Statusni bitovi				Funkcija
	Z	S	C	V	
TSF	N	N	N	N	Transfer operanda A kroz ALU
INC	D	D	N	N	Povećavanje A za 1 (inkrementiranje)
ADD	D	D	D	D	Sabiranje ( $A + B$ )
SUB	D	D	D	D	Oduzimanje ( $A - B$ )
DEC	D	D	N	N	Smanjivanje A za 1 (dekrementiranje)
TRC	D	D	0	N	Transfer sa postavljanjem statusnih bitova
AND	D	D	N	N	Logičko množenje ( $A \text{ AND } B$ )
OR	D	D	N	N	Logičko sabiranje ( $A \text{ OR } B$ )
XOR	D	D	N	N	Ekskluzivno ILI ( $A \text{ EXOR } B$ )
COM	D	D	N	N	Komplementiranje operanda A

Oznake:

- N – sadržaj statusnog bita ne zavisi od operacije
- D – sadržaj statusnog bita zavisi od operacije
- 0 – sadržaj statusnog bita postavlja se na nulu

## Operacije pomjeranja (polje H)

Tabela 7.10 Operacije pomjeranja (polje H)

Binarni

zapis	Mnemonik	Funkcija
000	NSH	Nema pomjeranja – No SHift
001	SHL	Pomjeranje ulijevo, nula na mjesto bita najniže težine
010	SHR	Pomjeranje udesno, nula na mjesto bita najviše težine
011	ZERO	Na izlazu pomjeračke jedinice su sve nule
100	RLC	Rotiranje ulijevo sa bitom prijenosa (C)
101	ROL	Rotiranje ulijevo
110	ROR	Rotiranje udesno
111	RRC	Rotiranje udesno sa bitom prijenosa (C)

## Sljedeća adresa (polje MUX2)

Tabela 7.11 Sljedeća adresa (polje MUX2)

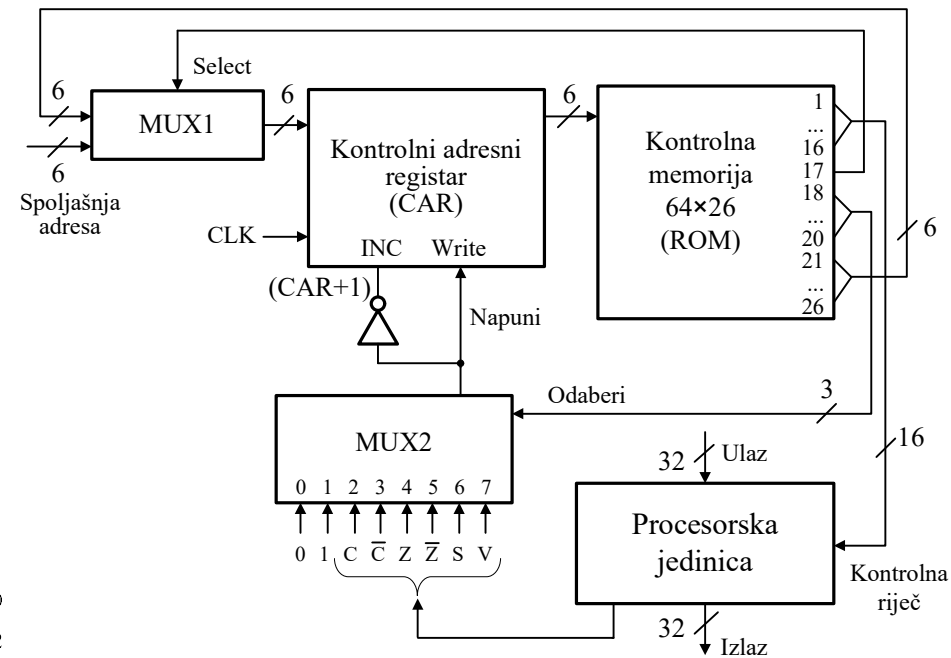
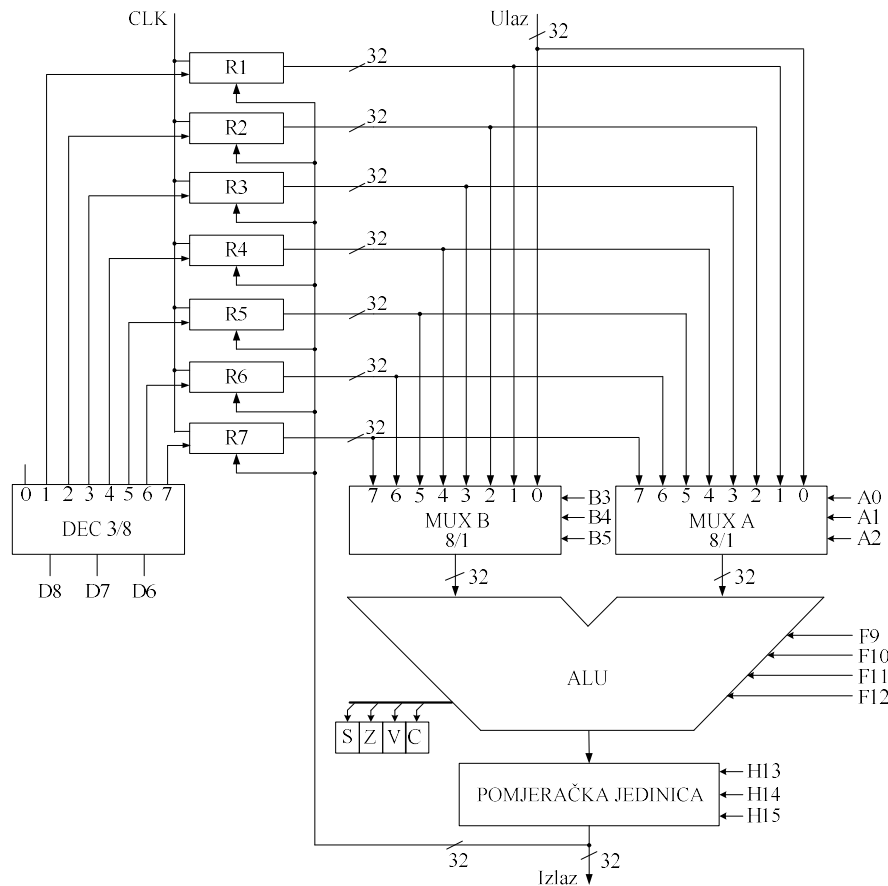
Bin. zapis	Mnemonik	Funkcija
000	NEXT	Sljedeća adresa je CAR+1
001	LAD	Sljedeća adresa je iz MUX1
010	LC	Ako je C=1, sljedeća adresa je iz MUX1; u suprotnom je CAR+1
011	LNC	Ako je C=0, sljedeća adresa je iz MUX1; u suprotnom je CAR+1
100	LZ	Ako je Z=1, sljedeća adresa je iz MUX1; u suprotnom je CAR+1
101	LNZ	Ako je Z=0, sljedeća adresa je iz MUX1; u suprotnom je CAR+1
110	LS	Ako je S=1, sljedeća adresa je iz MUX1; u suprotnom je CAR+1
111	LV	Ako je V=1, sljedeća adresa je iz MUX1; u suprotnom je CAR+1

## Podjela mikroinstrukcija i njihova implementacija

Tip instrukcije	Opis implementacije	Write kontrolni signal CAR-a (izlaz MUX 2)	Ulaz CAR-a (izlaz MUX 1)	Interna ads. (za upis u CAR)
Sekvencijalne	Onemogućiti upis adrese, selektirane kroz MUX 1, u CAR, već $CAR \leftarrow CAR+1$	0	Bilo što	Bilo što
Bezuslovni skok	Upisati adresu, selektiranu kroz MUX 1, u CAR	1	0 (interna ads.)	Address
			1 (eksterna ads.)	Bilo što
Uslovni skok	Upisati adresu, selektiranu kroz MUX 1, u CAR, ali samo ukoliko je zadovoljen uslov grananja zadat vrijednošću statusnih bitova S, Z, V, C	C, Z, S, V, $inv(C)$ , $inv(Z)$	Interna adresa	Address

**Primjer:** Odrediti mikroinstrukciju koja definiše sljedeću mikrooperaciju:  
 $R1 \leftarrow R1 \wedge R2, CAR \leftarrow CAR + 1$

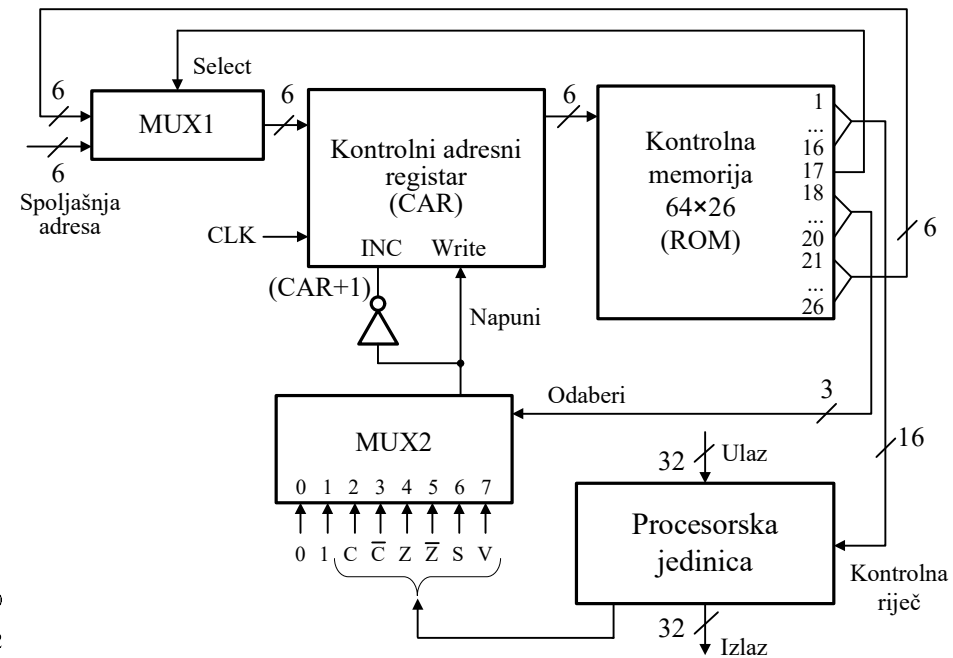
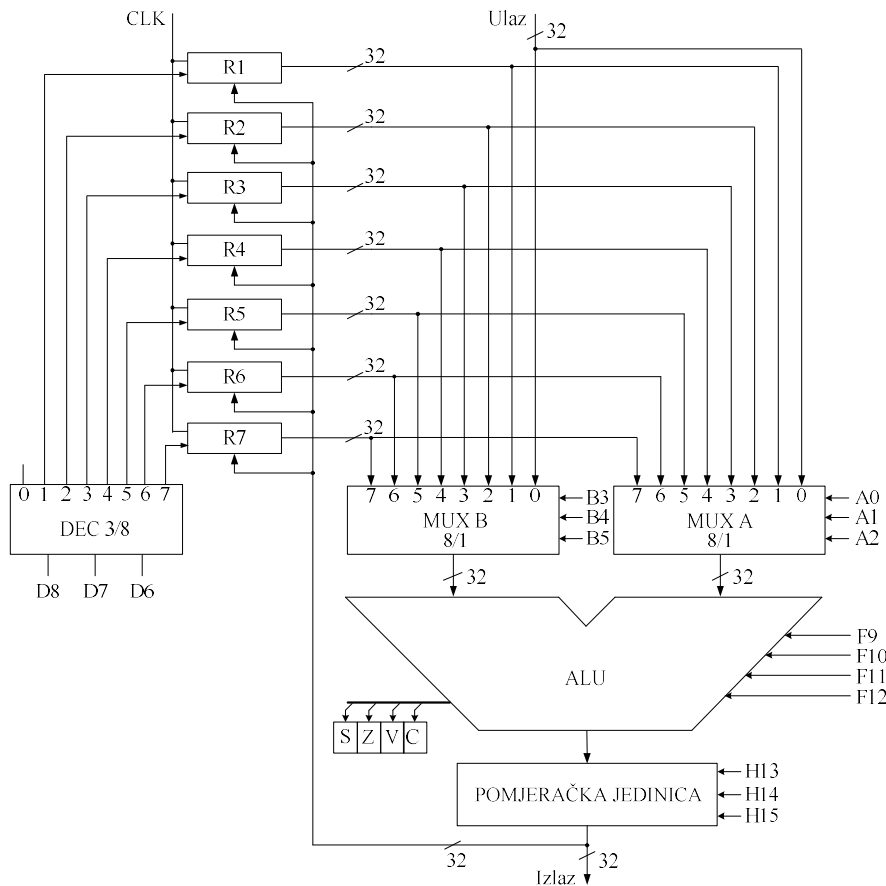
A	B	D	F	H	MUX1	MUX2	ADS
R1	R2	R1	AND	NSH	-	NEXT	-
001	010	001	1000	000	0	000	000000





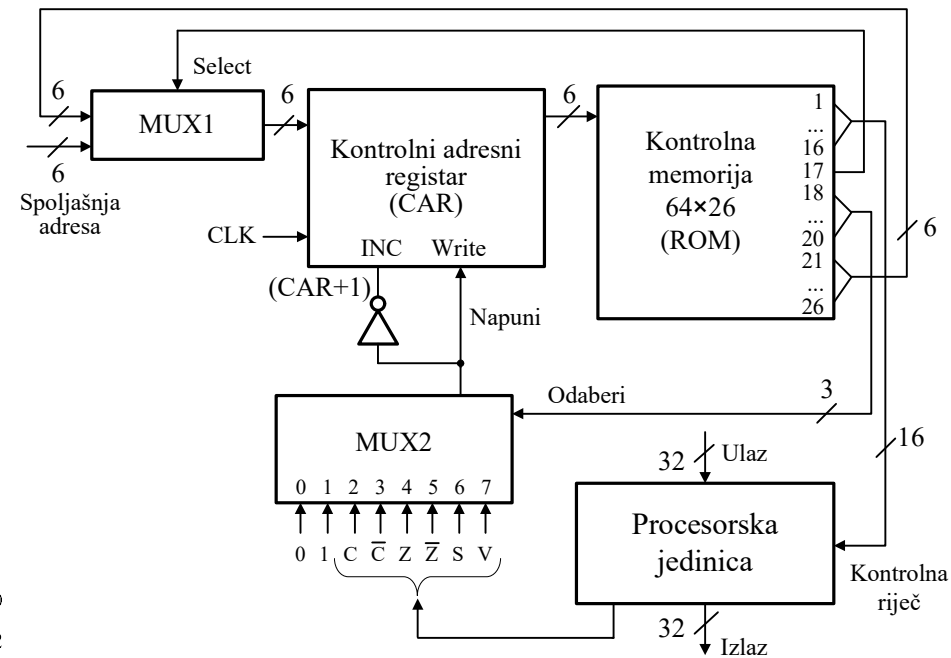
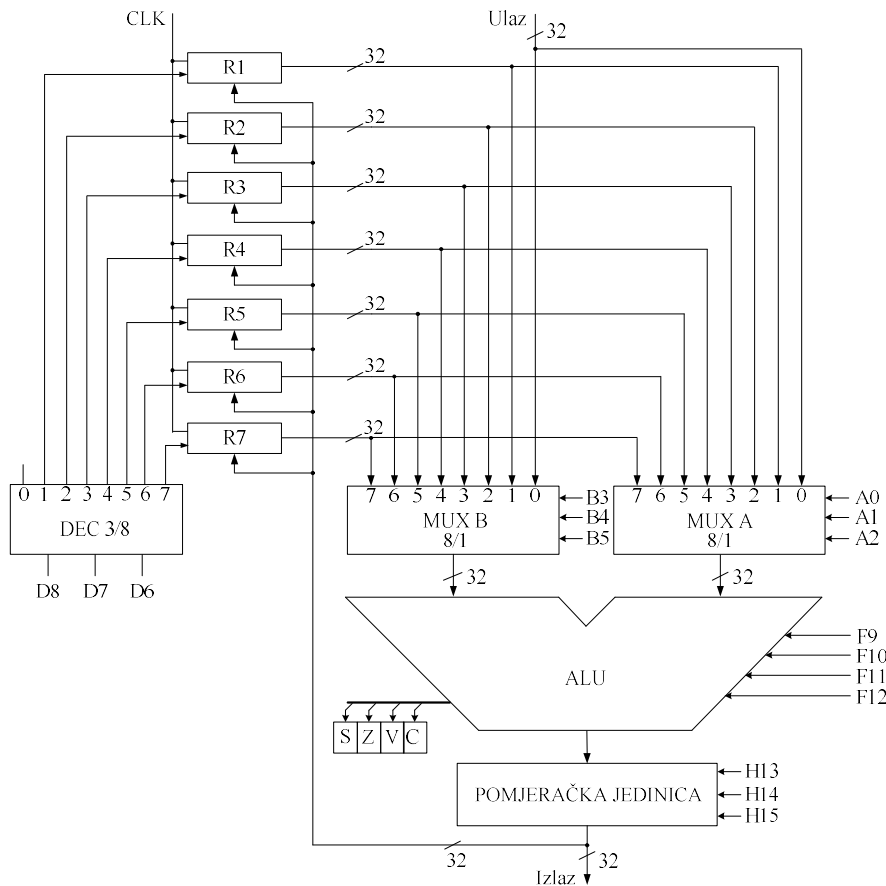
**Primjer:** Odrediti mikroinstrukciju koja definiše sljedeću mikrooperaciju:  
 $R3 \leftarrow R3 - 1, CAR \leftarrow 43$

A	B	D	F	H	MUX1	MUX2	ADS
R3	-	R3	DEC	NSH	INT	LAD	43
011	000	011	0110	000	0	001	101011



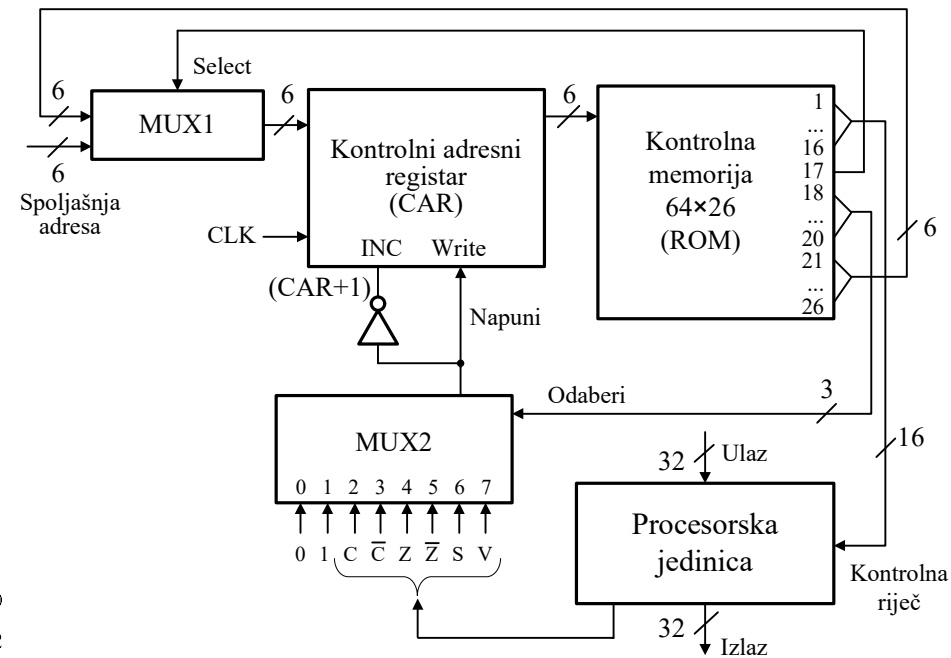
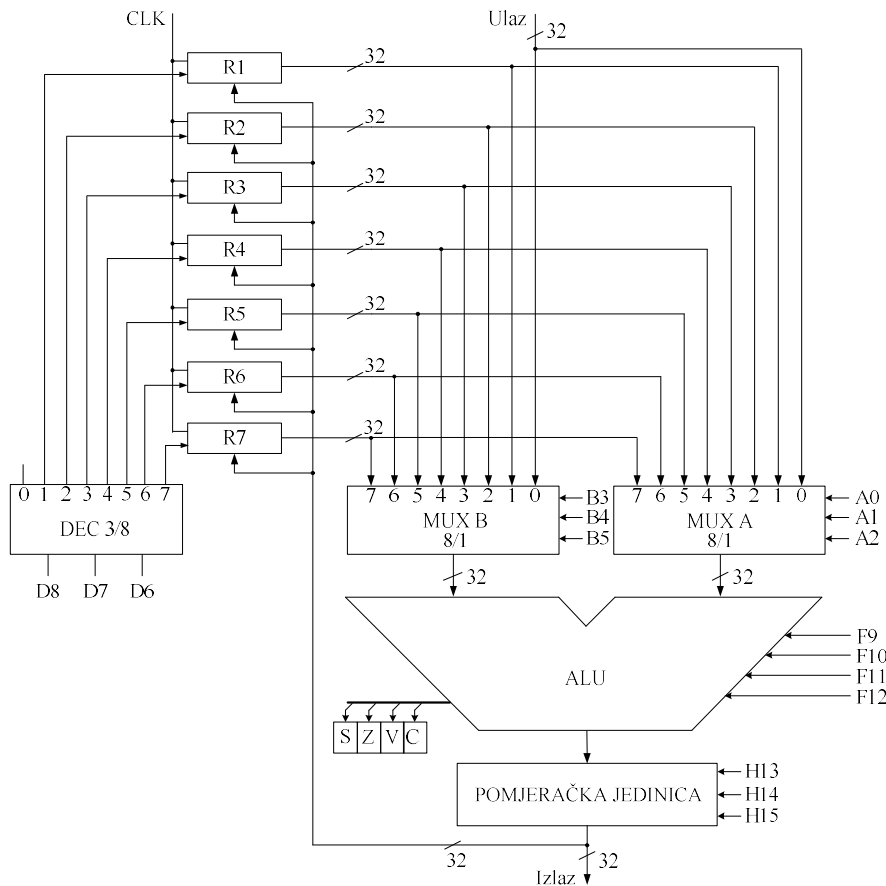
**Primjer:** Odrediti mikroinstrukciju koja definiše sljedeću mikrooperaciju:  
 $R4 \leftarrow 0$ , IF(S=1) THEN  $CAR \leftarrow 37$  ELSE  $CAR \leftarrow CAR + 1$

A	B	D	F	H	MUX1	MUX2	ADS
-	-	R4	TSF	ZERO	INT	LS	37
000	000	100	0000	011	0	110	100101



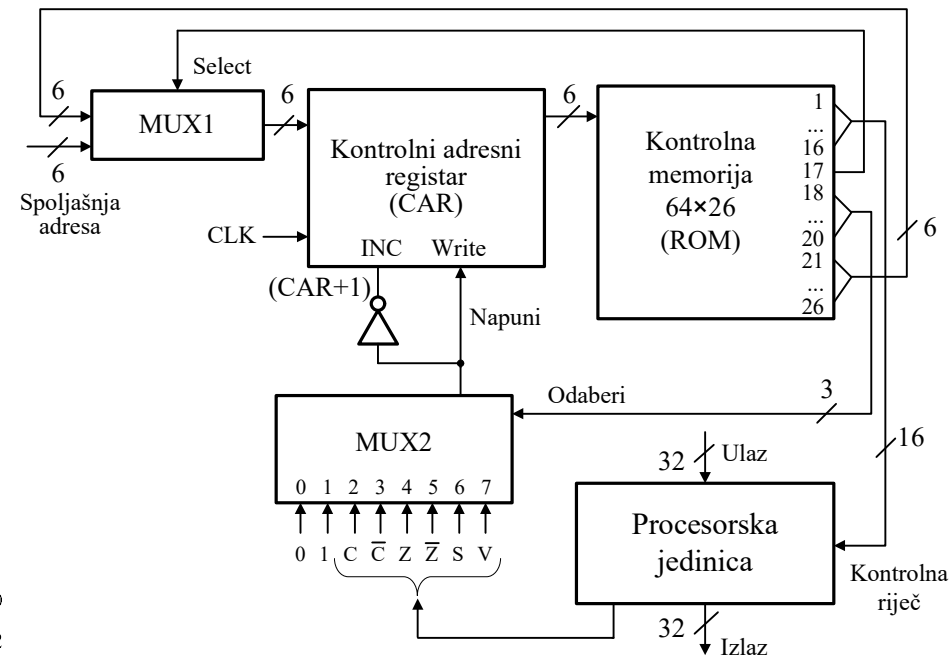
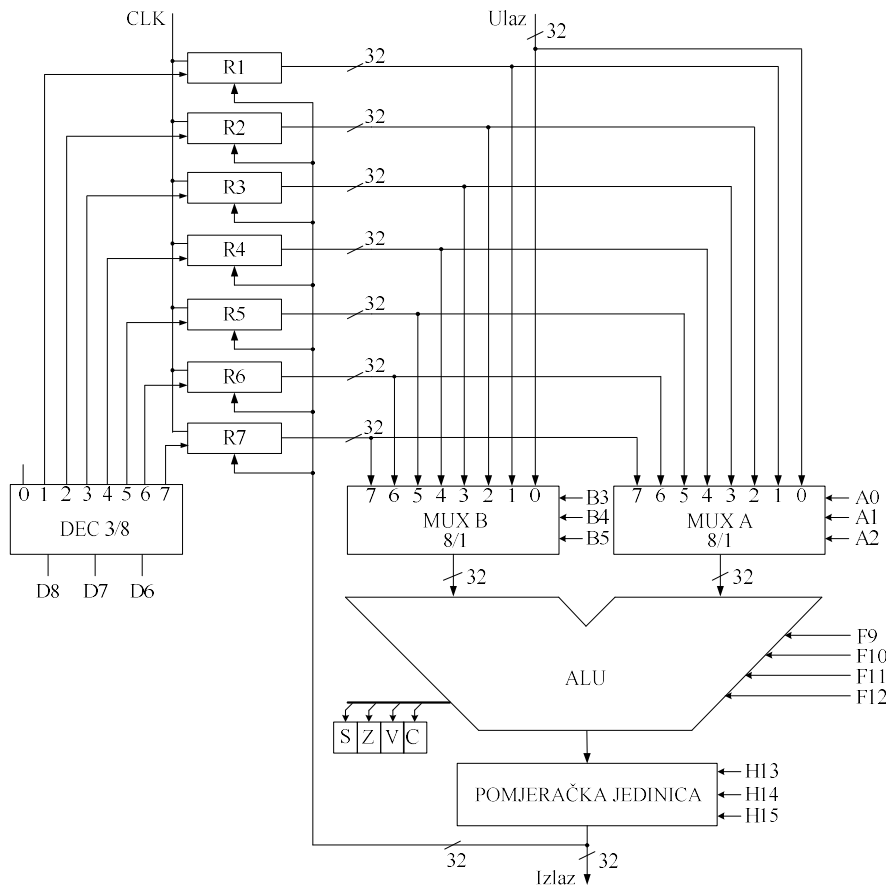
**Primjer:** Odrediti mikroinstrukciju koja definiše sljedeću mikrooperaciju:  
 $R5 \leftarrow \text{shl}(R5)$ , IF(C=0) THEN  $\text{CAR} \leftarrow 62$  ELSE  $\text{CAR} \leftarrow \text{CAR} + 1$

A	B	D	F	H	MUX1	MUX2	ADS
R5	-	R5	TSF	SHL	INT	LNC	62
101	000	101	0000	001	0	011	111110



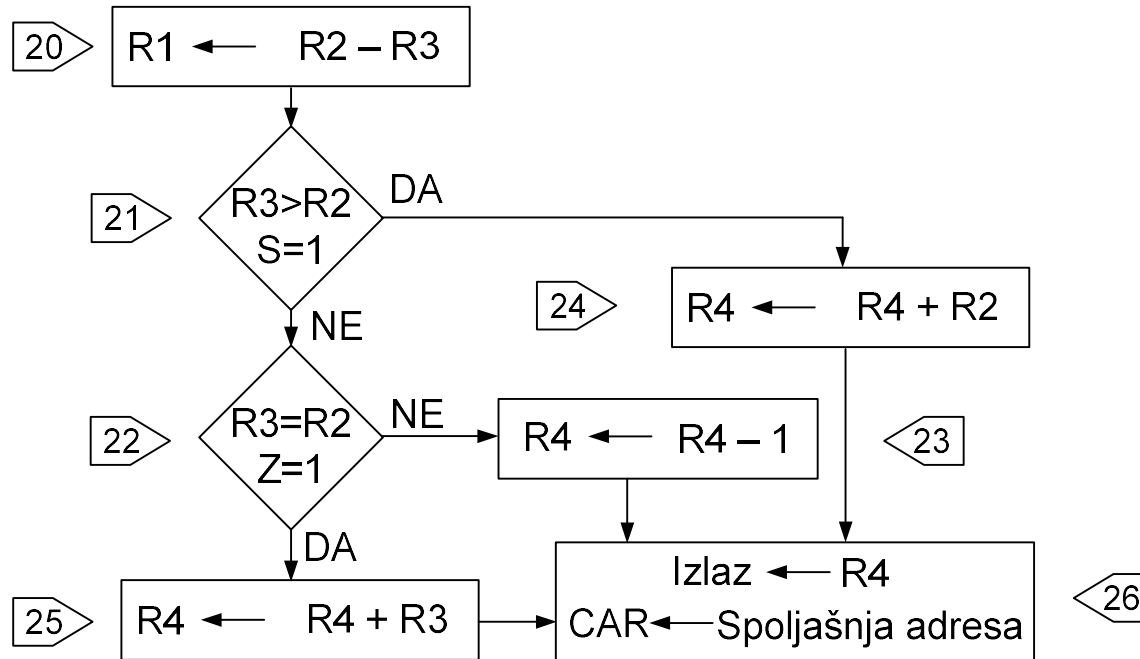
**Primjer:** Odrediti mikroinstrukciju koja definiše sljedeću mikrooperaciju:  
 $R2 \leftarrow R6 + R7$ ,  $CAR \leftarrow$  spoljašnja adresa

A	B	D	F	H	MUX1	MUX2	ADS
R6	R7	R2	ADD	NSH	EXT	LAD	-
110	111	010	0010	000	1	001	000000



# Primjer mikroprograma

**Primjer:** Napisati mikroprogram koji odgovara sljedećem dijagramu toka:



**Rješenje:**

- 20  $R1 \leftarrow R2 - R3, CAR \leftarrow CAR + 1$
- 21 IF(S=1) THEN  $CAR \leftarrow 24$  ELSE  $CAR \leftarrow CAR + 1$
- 22 IF(Z=1) THEN  $CAR \leftarrow 25$  ELSE  $CAR \leftarrow CAR + 1$
- 23  $R4 \leftarrow R4 - 1, CAR \leftarrow 26$
- 24  $R4 \leftarrow R4 + R2, CAR \leftarrow 26$
- 25  $R4 \leftarrow R4 + R3, CAR \leftarrow CAR + 1$
- 26  $Izlaz \leftarrow R4, CAR \leftarrow spoljašnja adresa$

Nastavak:

- 20  $R1 \leftarrow R2 - R3, CAR \leftarrow CAR + 1$
- 21 IF(S=1) THEN  $CAR \leftarrow 24$  ELSE  $CAR \leftarrow CAR + 1$
- 22 IF(Z=1) THEN  $CAR \leftarrow 25$  ELSE  $CAR \leftarrow CAR + 1$
- 23  $R4 \leftarrow R4 - 1, CAR \leftarrow 26$
- 24  $R4 \leftarrow R4 + R2, CAR \leftarrow 26$
- 25  $R4 \leftarrow R4 + R3, CAR \leftarrow CAR + 1$
- 26 Izlaz  $\leftarrow R4, CAR \leftarrow$  spoljašnja adresa

adresa	A	B	D	F	H	MUX1	MUX2	ADS
20	R2	R3	R1	SUB	NSH	-	NEXT	-
	010	011	001	0101	000	0	000	000000
21	-	-	NONE	TSF	NSH	INT	LS	24
	000	000	000	0000	000	0	110	011000
22	-	-	NONE	TSF	NSH	INT	LZ	25
	000	000	000	0000	000	0	100	011001
23	R4	-	R4	DEC	NSH	INT	LAD	26
	100	000	100	0110	000	0	001	011010
24	R4	R2	R4	ADD	NSH	INT	LAD	26
	100	010	100	0010	000	0	001	011010
25	R4	R3	R4	ADD	NSH	-	NEXT	-
	100	011	100	0010	000	0	000	000000
26	R4	-	NONE	TSF	NSH	EXT	LAD	-
	100	000	000	0000	000	1	001	000000