

7. Mikroprogramiranje

- Osnovni pojmovi i Wilkesova izvorna shema
- Faze mikroprogramiranja
- Struktura mikroprogramirane upravljačke jedinice
- Model mikroprogramiranog procesora
- Upravljačke riječi (mikroriječi)
- Primjer mikroprograma (instrukcija CBR)
- Formati upravljačkih riječi
- Mikroprogramiranje i nanoprogramiranje

Mikrooperacija - elementarna (nedjeljiva) operacija izravno i u potpunosti sklopovski podržana;

Primjeri:

- prijenos podataka (sadržaja registra) između dva registra
- aktiviranje sklopa u ALU
- brisanje ili postavljanje bistabila (zastavice)

Odnos između instrukcije u **strojnom jeziku** i **mikrooperacije**:

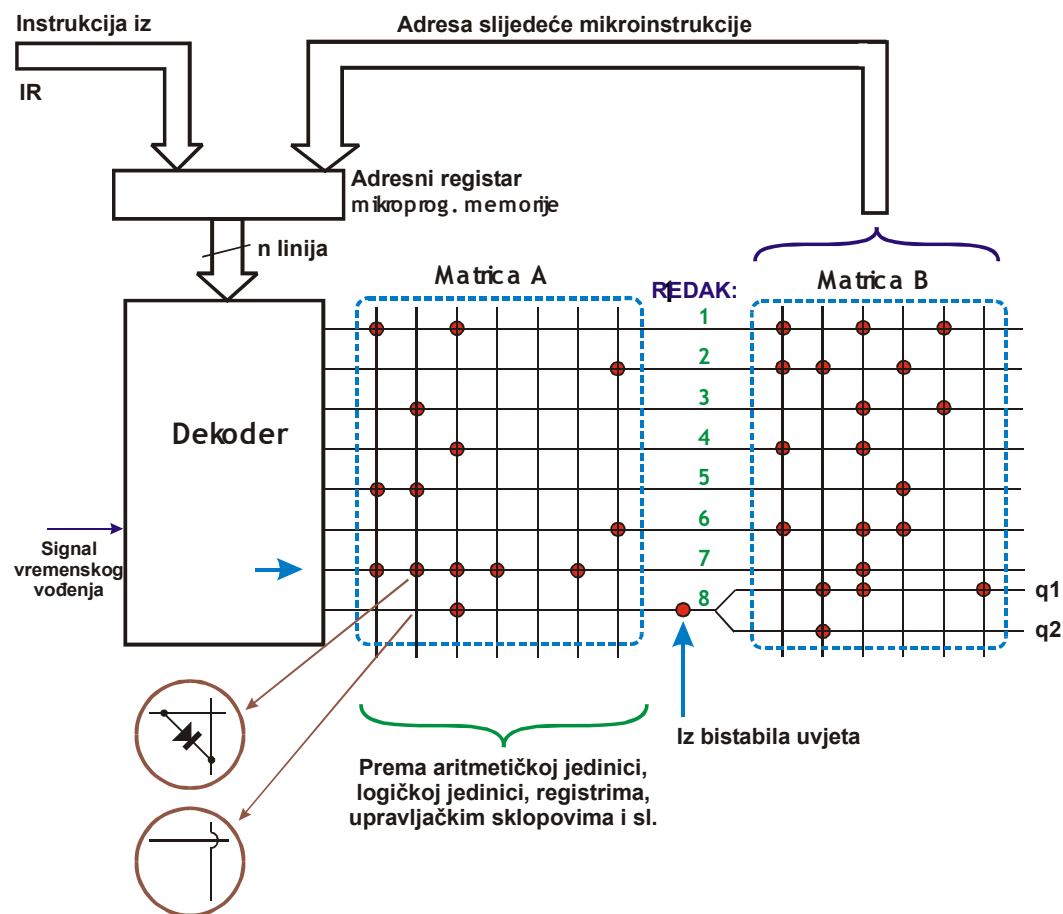
ADD	→	MDR ← M(MAR)
		IR ← MDR
		PC ← PC + 1
		D ← <i>opc</i>
		MAR ← MDR
		...

Mikroinstrukcija – kodirano predstavljena (nizom bitova) **jedna** ili **više mikrooperacija**

Mikroprogram – slijed mikroinstrukcija koje su pohranjene u upravljačkoj (mikroprogramskoj) memoriji

Mikroinstrukcija pohranjena u upravljačkoj memoriji naziva se i **mikroriječ** (engl. Microword) ili **upravljačka riječ** (engl. Control word)

Wilkesova izvorna shema (M. V. Wilkes (1951. godine))



Primjer: Instrukcijski ciklus za ADD

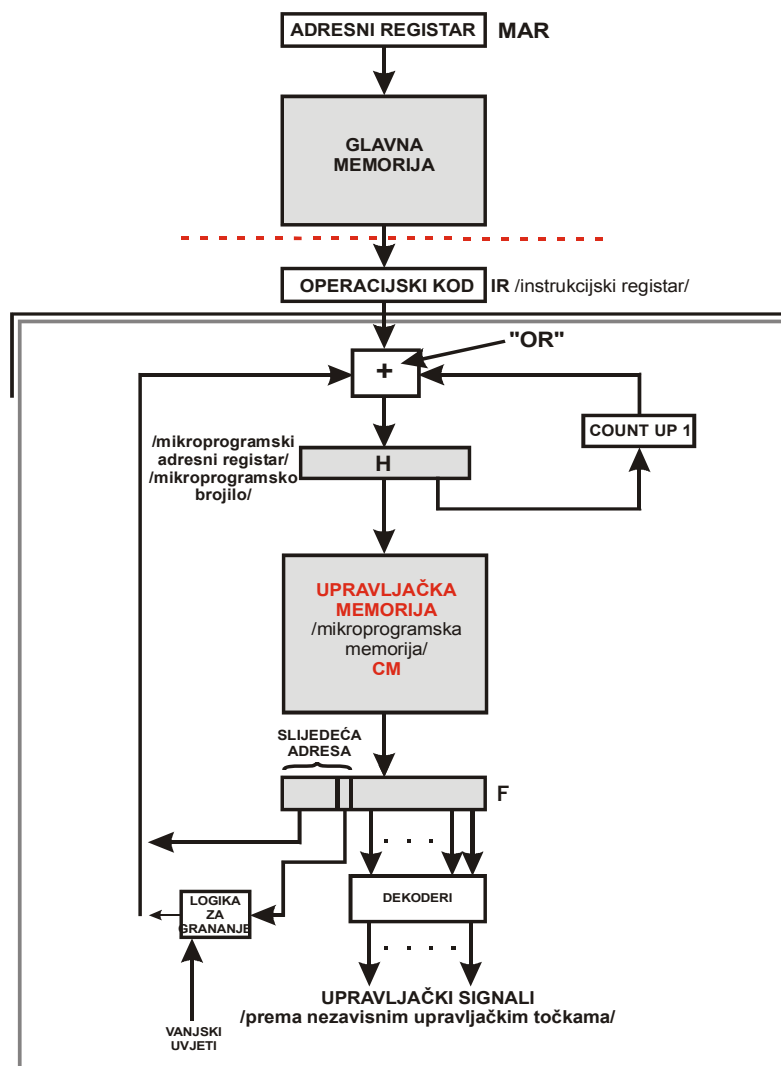
3. mikrooperacija: $\text{MDR} \leftarrow \text{M}(\text{MAR})$
4. mikrooperacija: $\text{IR} \leftarrow \text{MDR}, \text{PC} \leftarrow \text{PC} + 1$
5. mikrooperacija: $\text{D} \leftarrow \text{opc}$
6. mikrooperacija: $\text{MAR} \leftarrow \text{AP}, \text{F} \leftarrow 0, \text{E} \leftarrow 1$
7. mikrooperacija: $\text{TR1} \leftarrow \text{MDR}, \text{TR2} \leftarrow \text{ACC}$
8. mikrooperacija: $\text{ACC} \leftarrow \text{ADDER}$
9. mikrooperacija: $\text{E} \leftarrow 0, \text{F} \leftarrow 1, \text{MAR} \leftarrow \text{PC}$

Tri faze mikroprogramiranja:

- III. faza: uporaba diodnih matrica /vrijeme pristupa $\sim 0,5$ mikrosekundi
vrijeme pristupa memoriji ~ 10 mikrosekundi/
- VII. faza (kasne 50-te godine prošlog stoljeća): /glavna memorija –
feritne jezgrice; diodne matrice nisu
dovoljno brze; mikroprogramiranje –
samo za emulaciju računala
- XII. faza (krajem 60-tih godina): pojava brzih (bipolarnih) memorija za
izvedbu upravljačke memorije

Mikroprogramiranje – najbolja (sustavna) metoda za realizaciju upravljačke jedinice

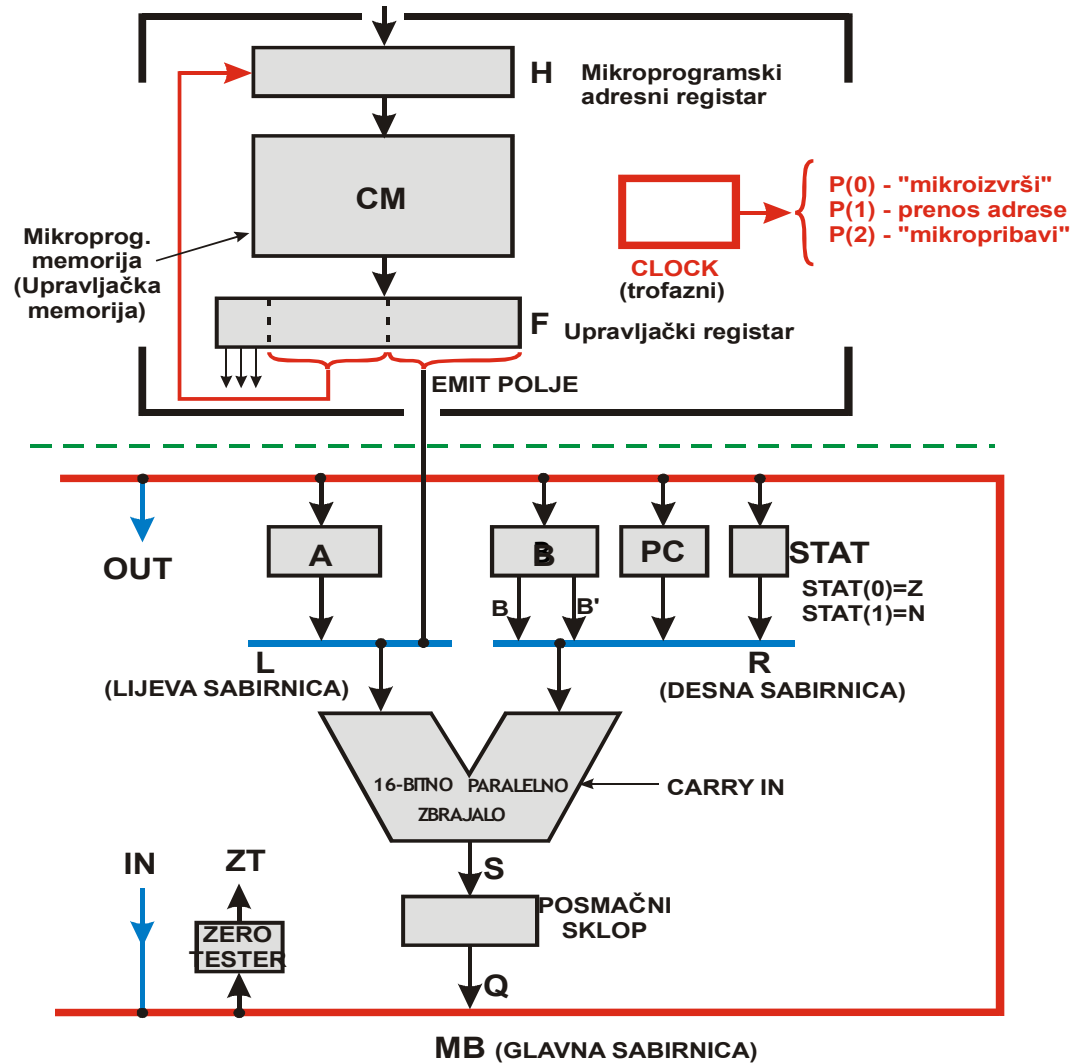
Struktura mikroprogramirane upravljačke jedinice



Načini dobivanja adrese
slijedeće mikroinstrukcije:

4. povećanjem sadržaja mikroprogramskog adresnog registra H (countup 1)
7. prijenosom adresnog polja upravljačke riječi u H
3. uporabom logike za grananje

Model mikroprogramiranog procesora

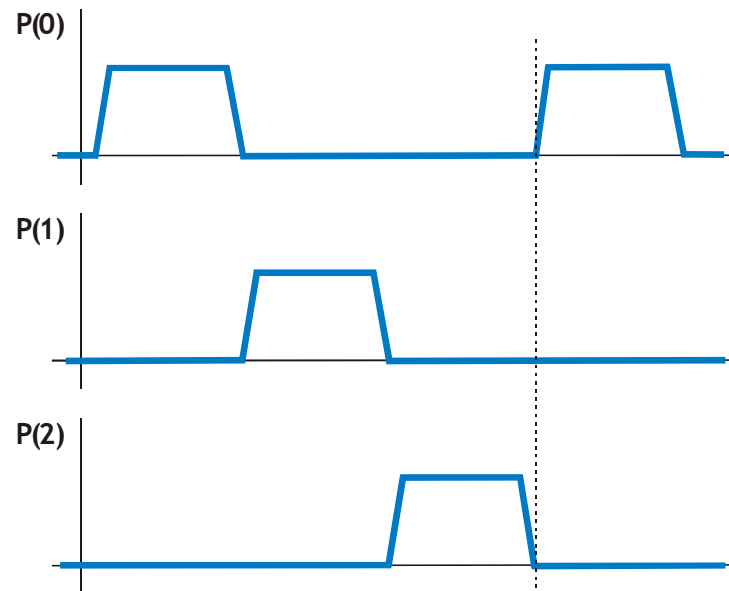


Mikroprogramirana upravljačka jedinica ima **trofazni signal** vremenskog vođenja:

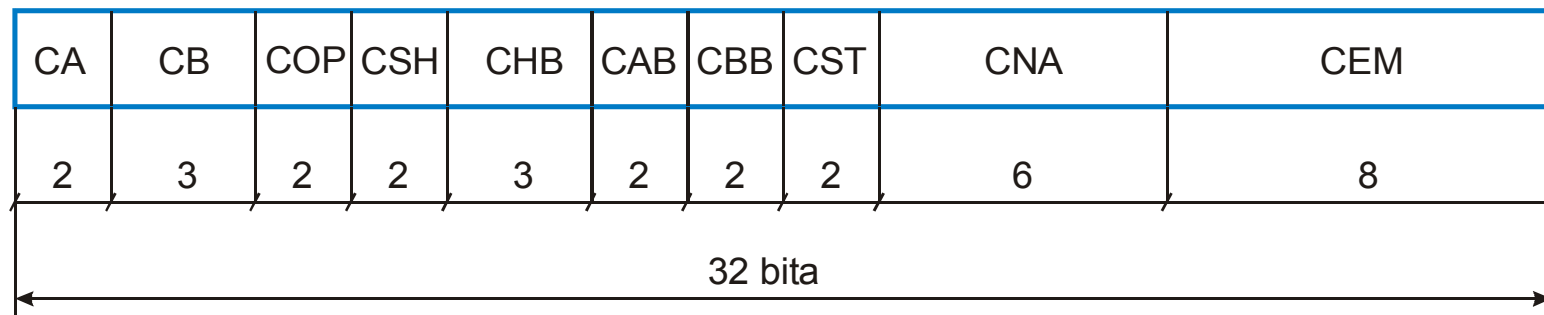
faza P(0) – izvršavanje mikroinstrukcija (“mikroizvrši”)

faza P(1) – prijenos adrese u adresni mikroprogramski registar H

faza P(2) – čitanje mikroinstrukcije iz mikroprogramske memorije (“mikropribavi”)



Format upravljačke (mikroriječi) riječi: **CMB**



CA – veza sa sabirnicom L

CB – veza sa sabirnicom R

COP – mikrooperacije paralelnog 16-bitnog zbrajala

CHS – upravljanje posmačnim sklopom

CMB – veza glavne sabirnice (MB) i ostalih komponenti procesora

CAB – utjecaj na H(6)

CBB – utjecaj na H(7)

CST – utjecaj na STAT

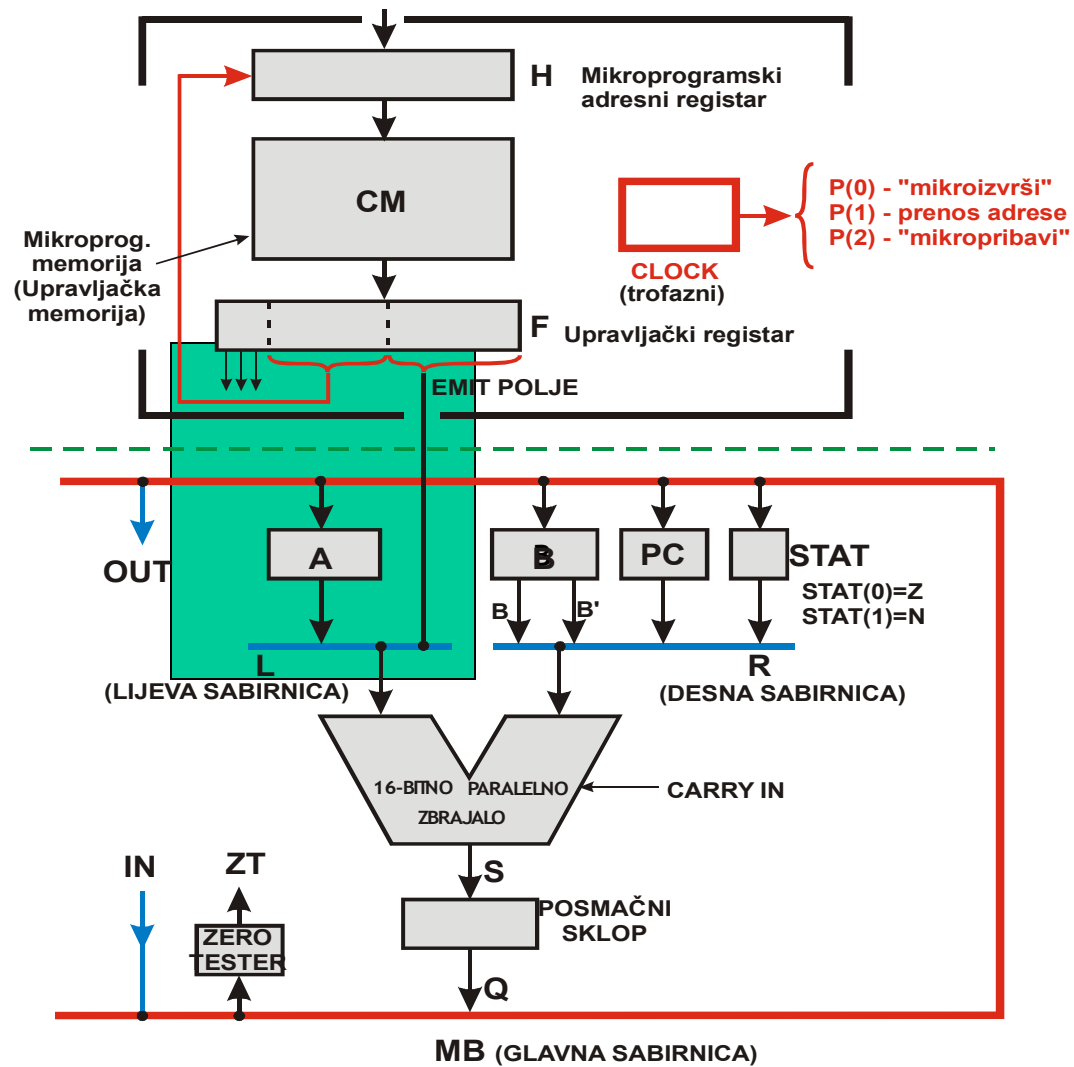
CNA – polje adrese slijedeće mikroinstrukcije

CEM – 8-bitna konstanta /emit polje/

CA – veza sa sabirnicom L

00	ϕ ; nema prijenosa podataka na lijevu sabirnicu (L)
01	$L(0-7,8-15) = 0 - F(EM)$; konstanta se smještava na 8 najne značajnih linija (LSB) sabirnice L
10	$L(0-7, 8-15) = F(EM) - 0$; konstanta se smještava na MSB sabirnice L
11	$L = A$; na L se smještava sadržaj akumulatora A

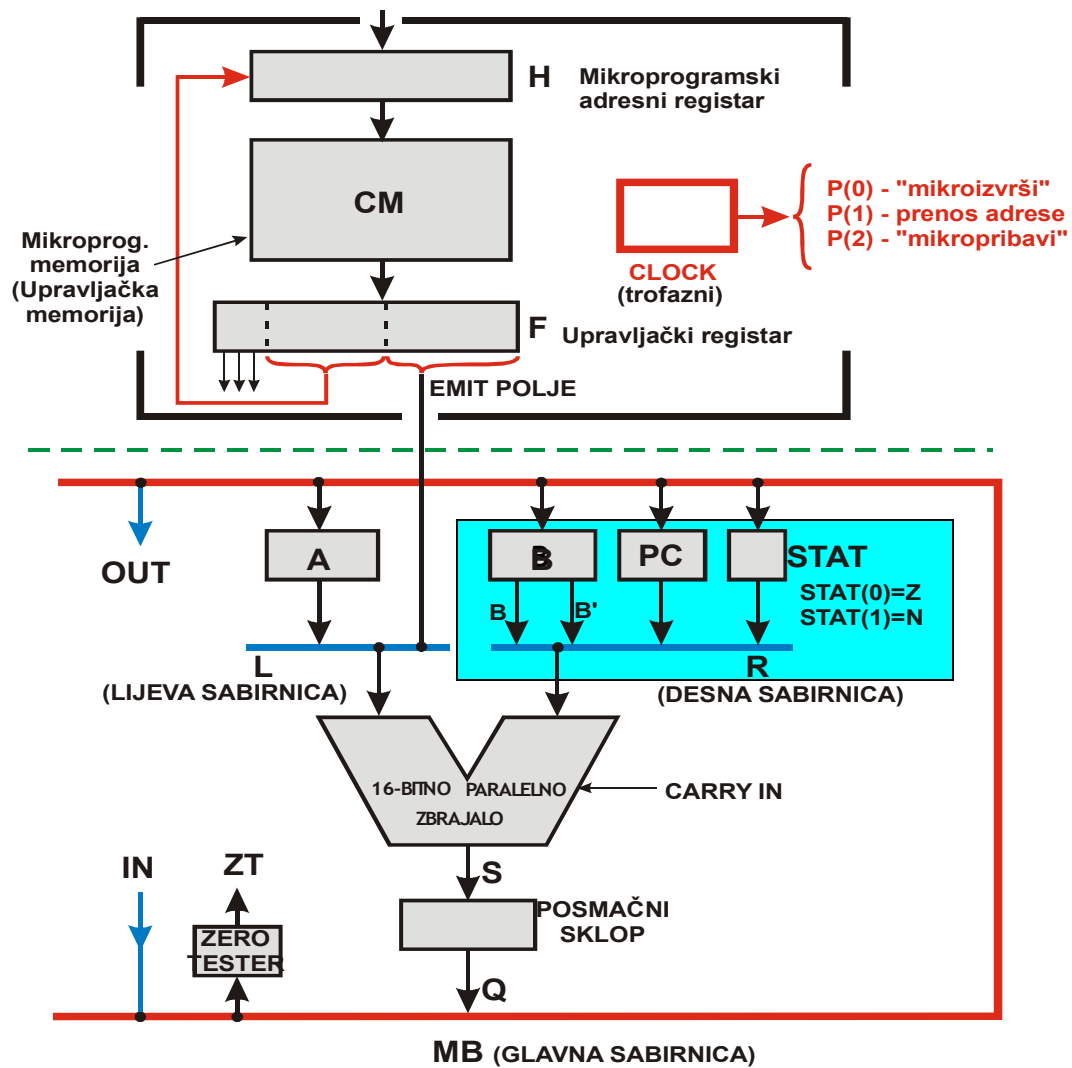
Model mikroprogramiranog procesora – polje CA



CB – veza sa sabirnicom R

000	ϕ ; nema prijenosa podataka
001	$R = B$; prijenos akumulatora B na desnu sabirnicu (R)
010	$R = B'$; prijenos jediničnog komplementa sadržaja akumulatora B
011	$R = PC$
100	$R = STAT$
101	ne koristi se
...	ne koristi se
111	ne koristi se

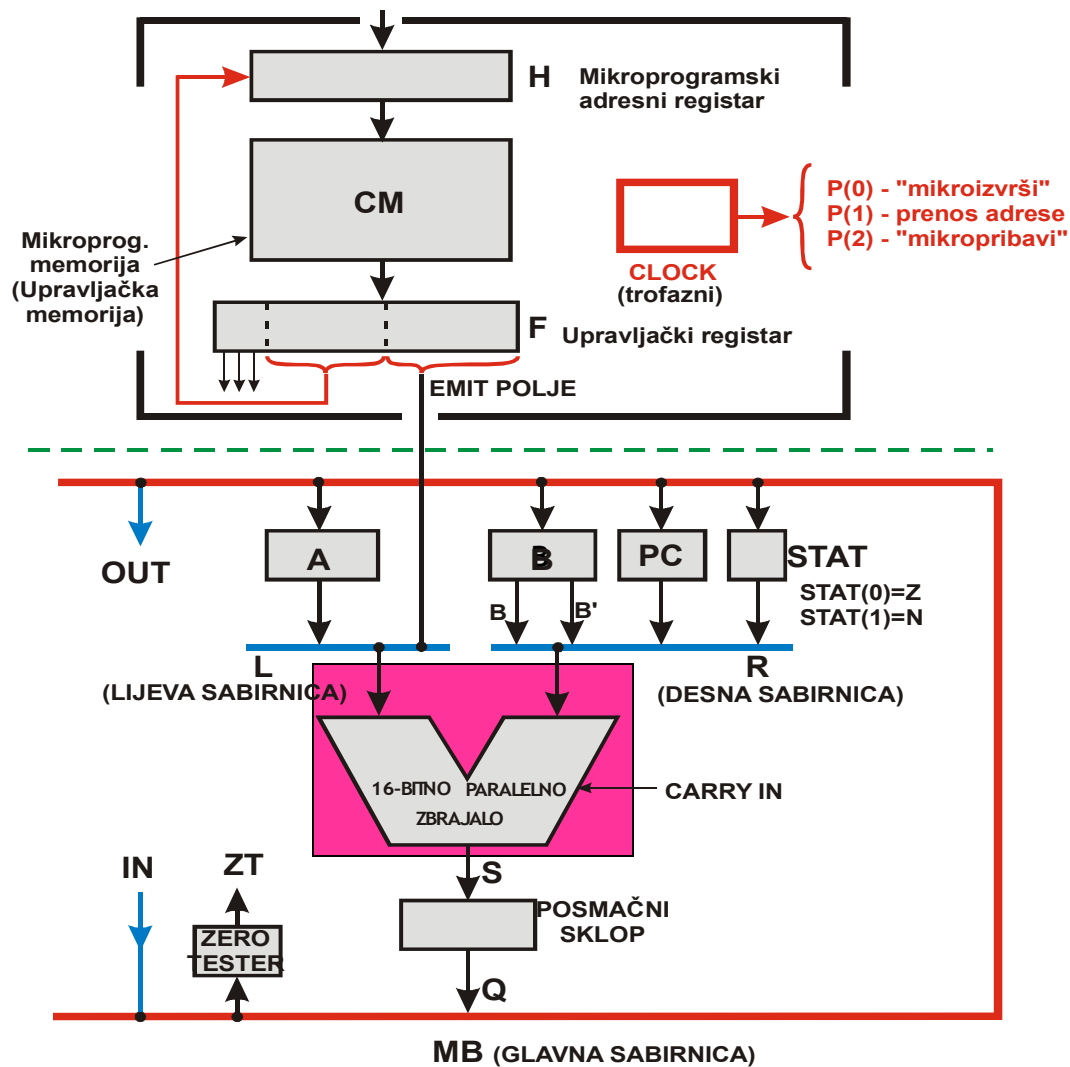
Model mikroprogramiranog procesora – polje CB



COP – upravljanje paralelnim 16-bitnim paralelnim zbrajalom

00	zbroji sa $C_{in} = 0$
01	zbroji sa $C_{in} = 1$
10	ne koristi se
11	ne koristi se

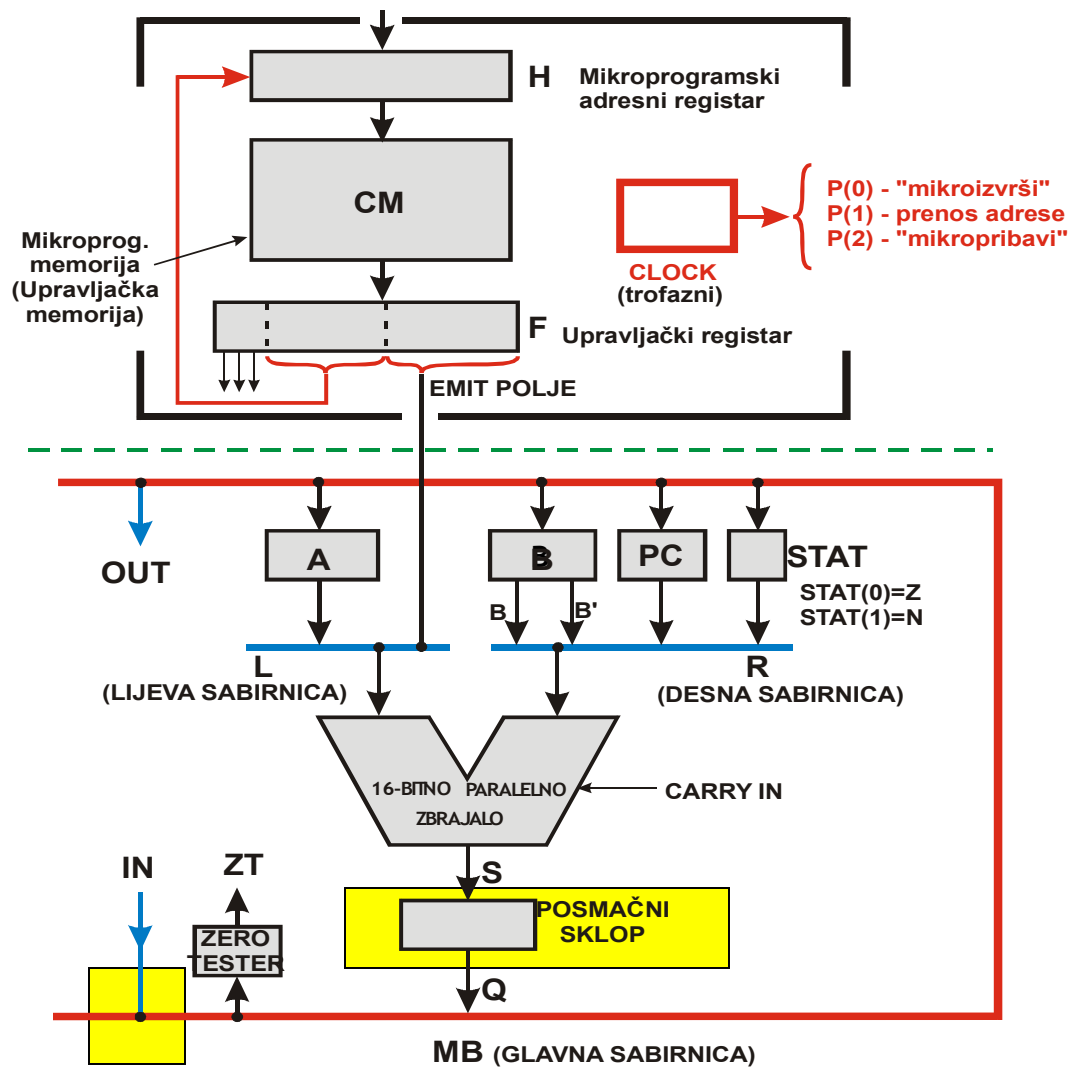
Model mikroprogramiranog procesora – polje COP



CSH – upravljanje posmačnim sklopom

00	$MB = Q, Q = S$; nema posmaka
01	$MB = Q, Q = shr\ S$; posmak udesno
10	$MB = Q, Q = shl\ S$; posmak ulijevo
11	$MB = IN$

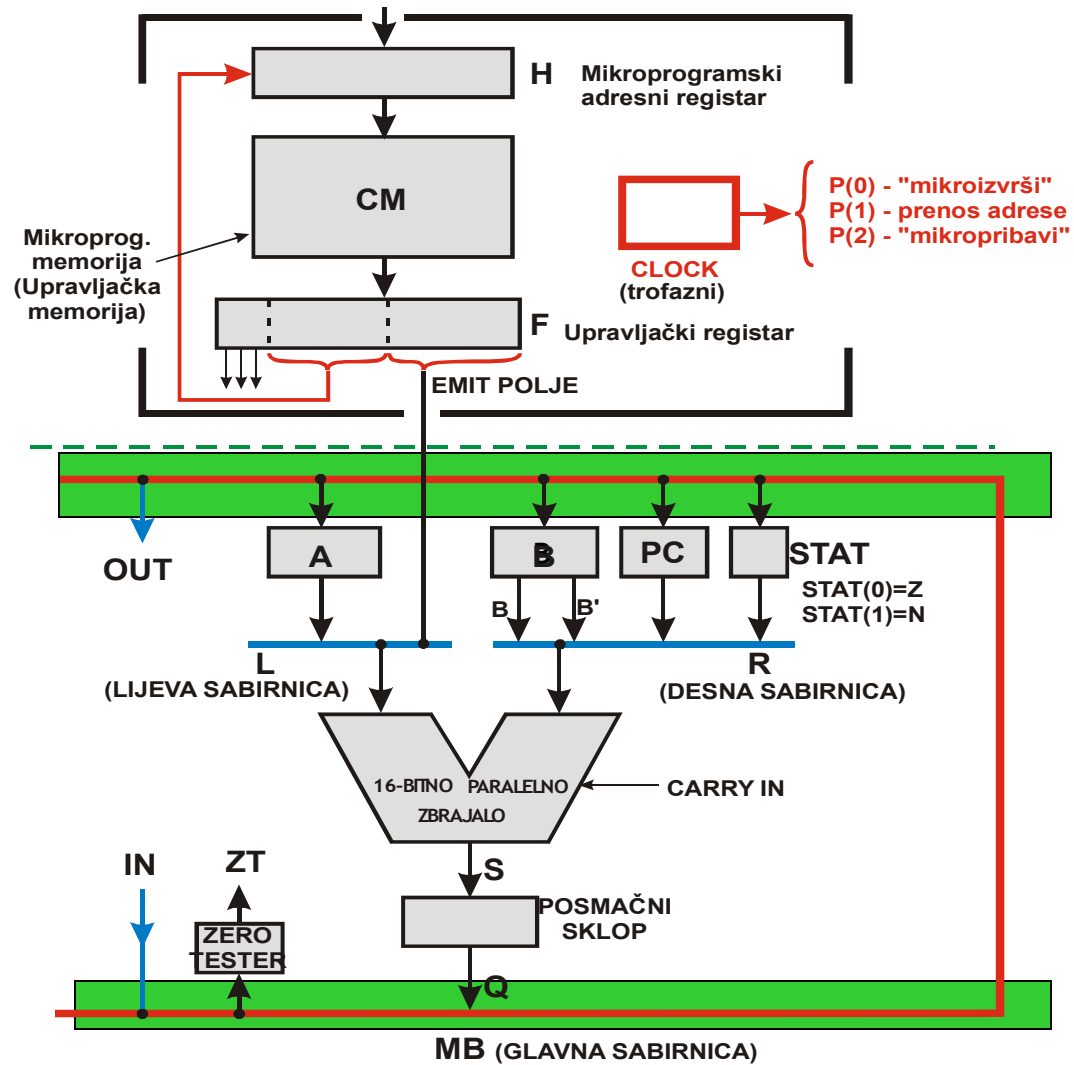
Model mikroprogramiranog procesora – polje CSH



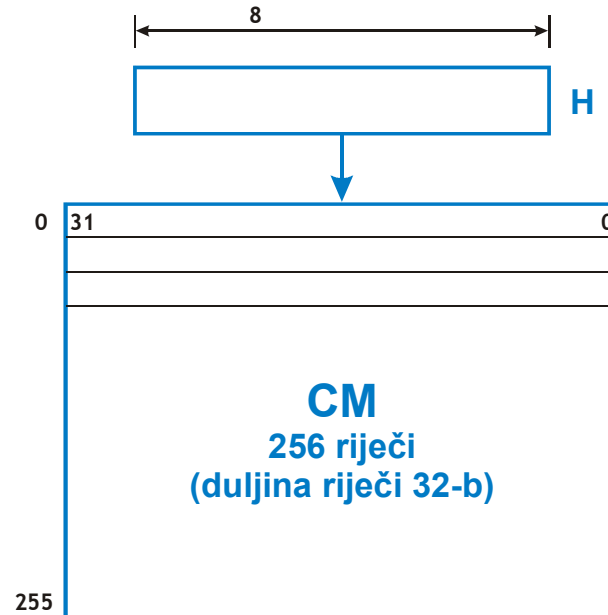
CMB – veza glavne sabirnice (MB) i ostalih komponenti

000	ϕ ; nema prijenosa podataka
001	$A \leftarrow MB$
010	$B \leftarrow MB$
011	$PC \leftarrow MB$
100	$STAT \leftarrow MB$
101	$OUT = MB$
...	ne koristi se
111	ne koristi se

Model mikroprogramiranog procesora – polje CMB

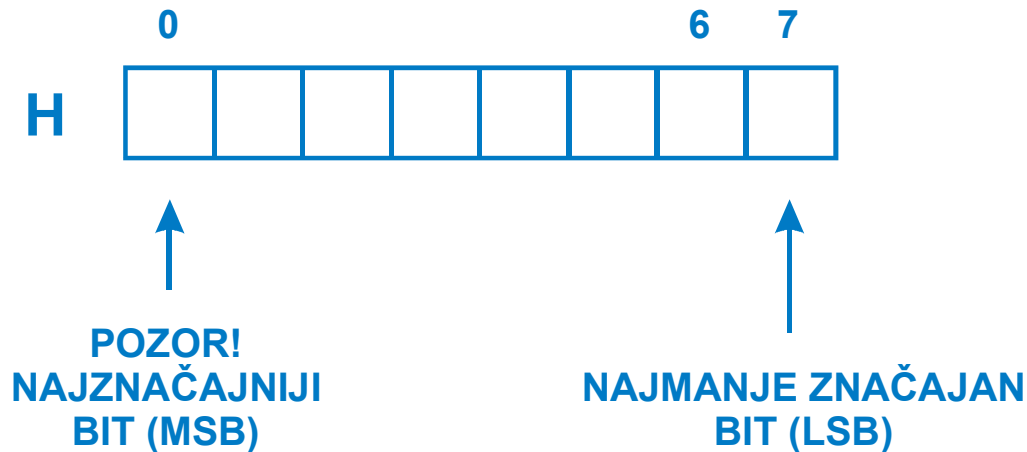


CM – upravljačka, mikroprogramska memorija



Mikroprogramski adresni
registar H

Mikroprogramski adresni registar H



Šest najznačajnijih bitova H(0-5) namijenjeno je polju CNA (polje adrese slijedeće instrukcije)

Bitovi H(6, 7) određuju se poljima CAB i CBB

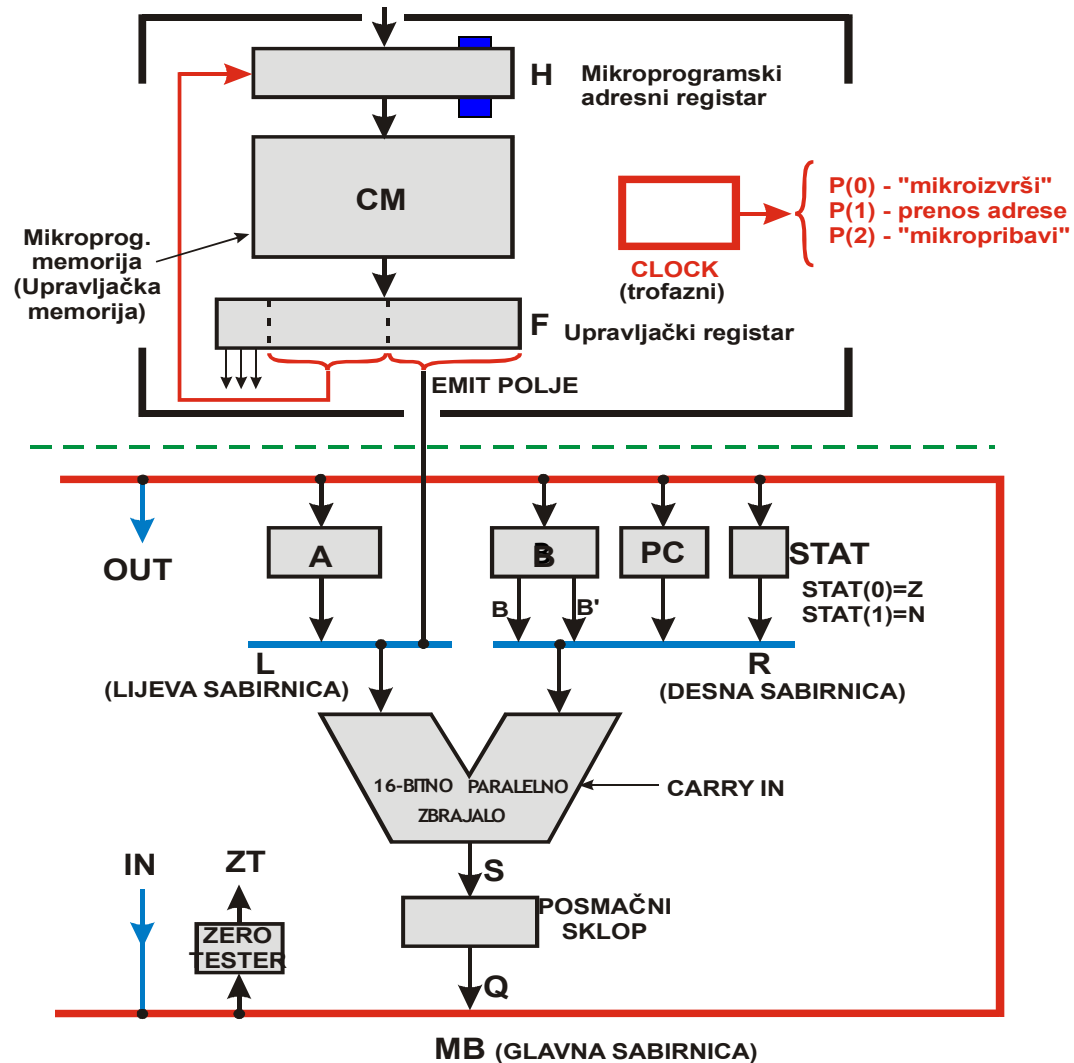
CAB – utjecaj na H(6)

00	$H(6) \leftarrow 0$
01	$H(6) \leftarrow 1$
10	$H(6) \leftarrow \text{STAT}(0)$
11	$H(6) \leftarrow \text{STAT}(1)$

$\text{STAT}(0) = Z$ (zastavica)

$\text{STAT}(1) = N$ (zastavica)

Model mikroprogramiranog procesora – polje CAB



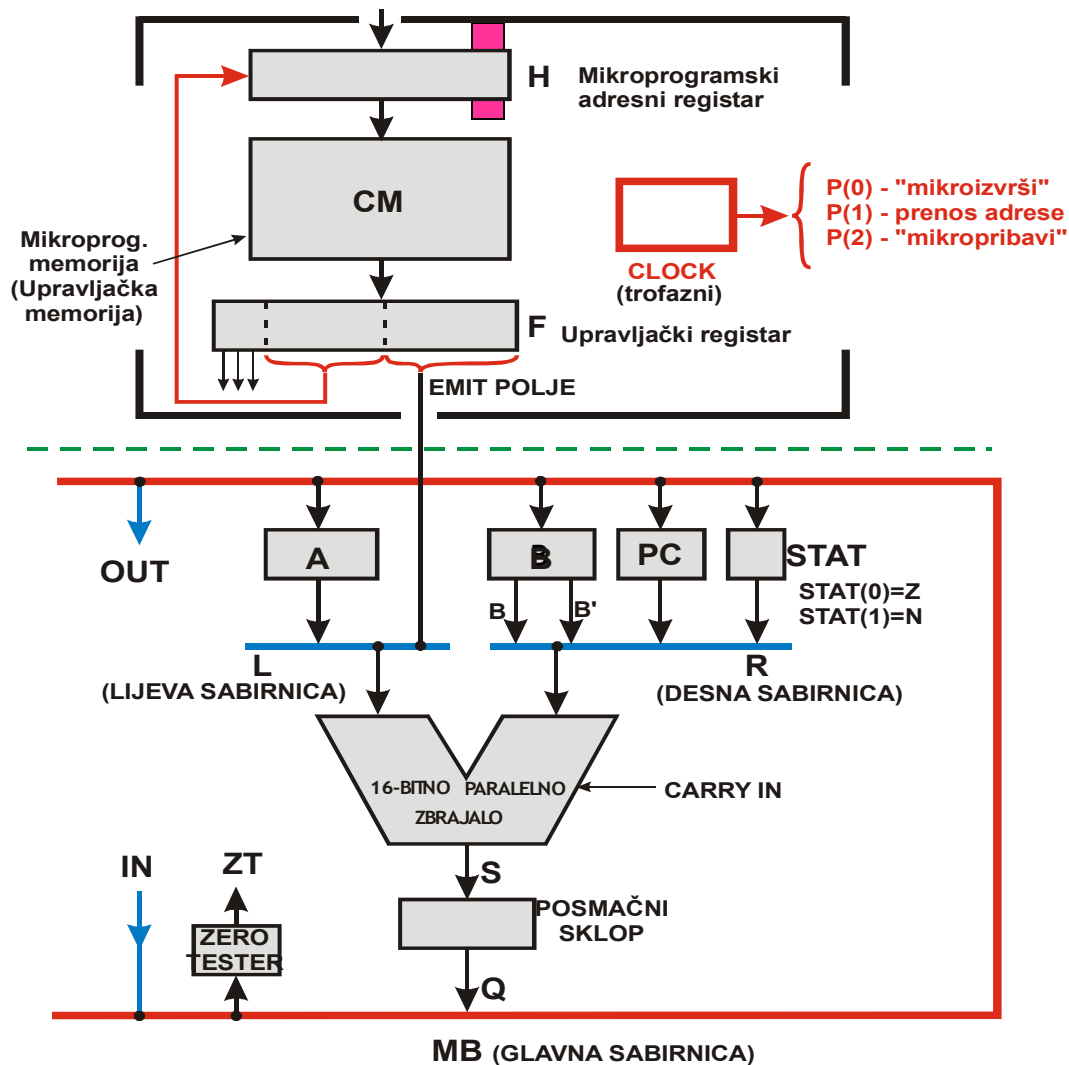
CBB – utjecaj na H(7)

00	$H(7) \leftarrow 0$
01	$H(7) \leftarrow 1$
10	$H(7) \leftarrow \text{STAT}(1)$
11	$H(7) \leftarrow \text{MB}(0)$; MB(0) je najznačajniji bit 16-bitne riječi koja “živi” na sabirnici

$\text{STAT}(0) = Z$ (zastavica)

$\text{STAT}(1) = N$ (zastavica)

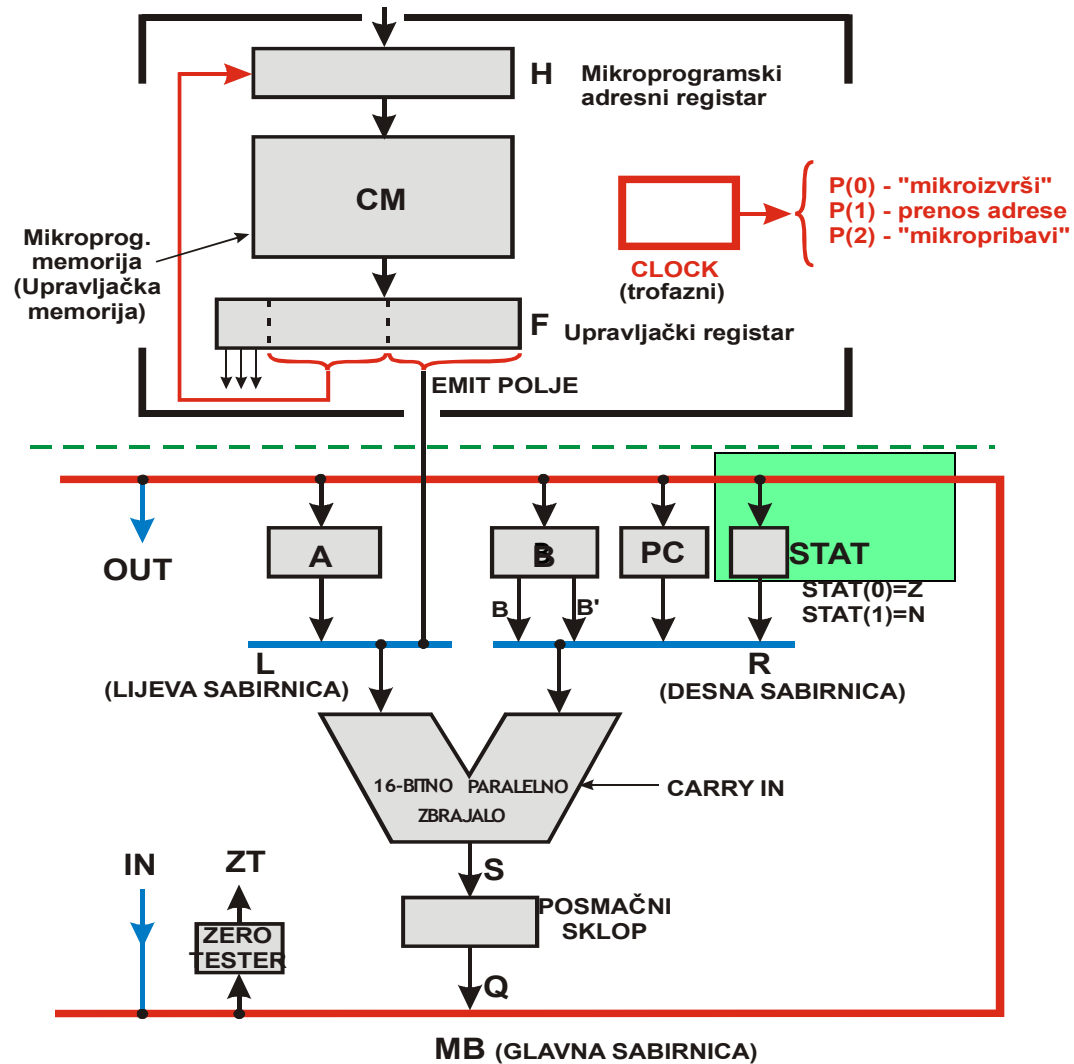
Mikroprogramirani model mikroprocesora – polje CBB



CST – utjecaj na STAT

00	ϕ ; nema utjecaja
01	<i>IF</i> (ZT = 0) <i>THEN</i> (STAT(0) \leftarrow 0) <i>ELSE</i> (STAT(0) \leftarrow 1)
10	STAT(1) \leftarrow MB(0)
11	(<i>IF</i> (ZT = 0) <i>THEN</i> (STAT(0) \leftarrow 0)) <i>ELSE</i> \leftarrow \leftarrow (STAT(0) 1), STAT(1) MB(0))

Model mikroprogramiranog procesora – polje CST



Tablica odnosa ZT' i MB(0)

ZT' (komplementirani izlaz iz sklopa za ispitivanje nule)	MB(0) Najznačajniji bit sabirnice MB	
0	0	rezultat na MB(0) = 0
0	1	Nije dopuštena kombinacija
1	0	rezultat > 0
1	1	rezultat < 0

CNA – polje slijedeće adrese

CNA \longrightarrow H(0-5)

CEM – 8-bitna konstanta

ZADATAK:

Napisati mikroprogram za instrukciju CBR (Conditional Branch) /i to za fazu IZVRŠI/ kojom se uspoređuju dva broja (jedan u registru A a drugi u registru B) uz pretpostavku da su brojevi preočeni u notaciji dvojnog komplementa. Na temelju rezultata uspoređivanja poduzima se jedna od slijedećih akcija:

- a) Ako je broj u registru A veći od broja u registru B, tada nema promjene PC-a /tijekom faze IZVRŠI/
- b) Ako su brojevi jednaki, PC se inkrementira (povećava za 1) /tijekom faze IZVRŠI/
- c) Ako je broj u A manji od broja u B, tada se PC povećava za 2 /tijekom faze IZVRŠI/

Strojna instrukcija: CBR ; instrukcija uvjetnog grananja

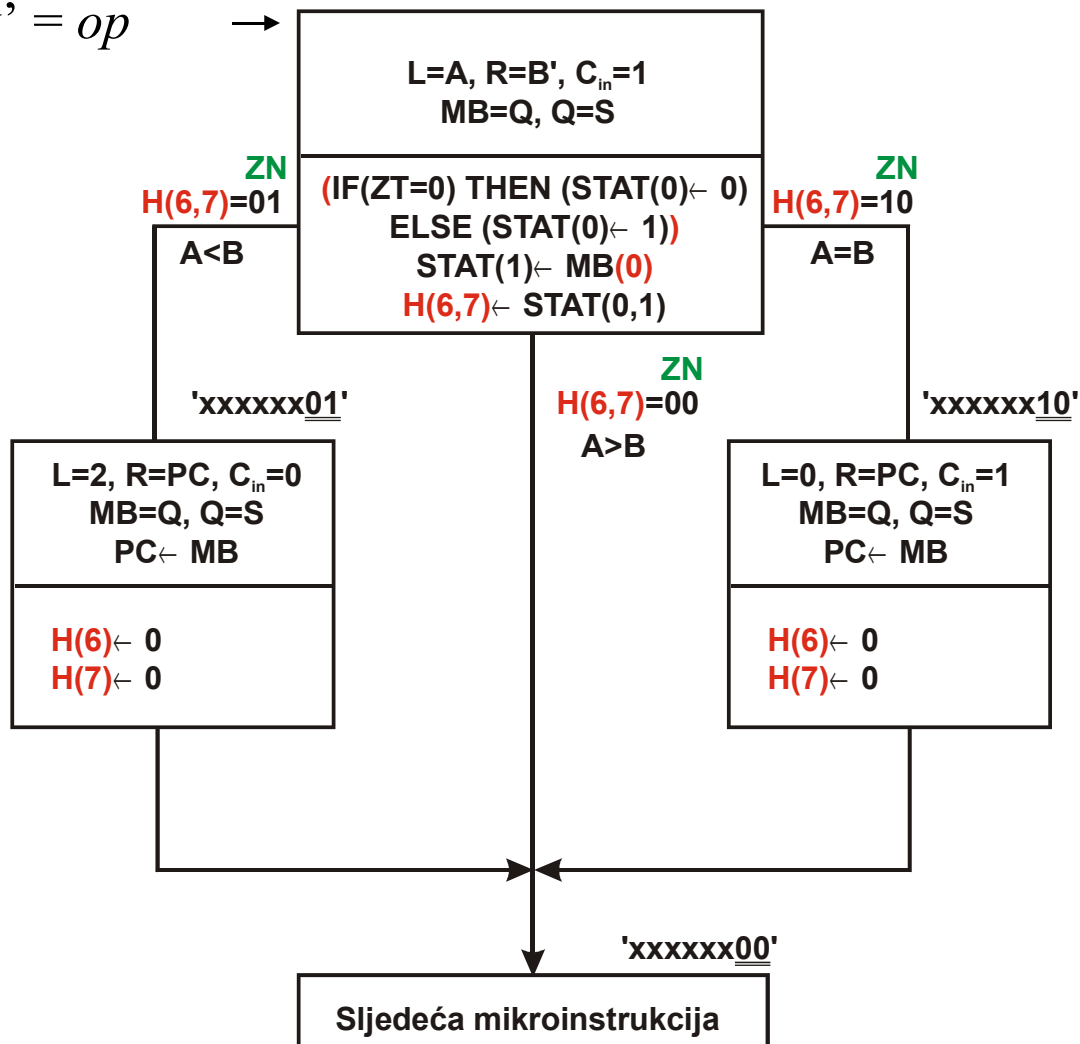
IF A > B THEN NOP (PC = PC)

IF A = B THEN PC = PC + 1

IF A < B THEN PC = PC + 2

Dijagram toka

'yyyyyyyy' = *op*

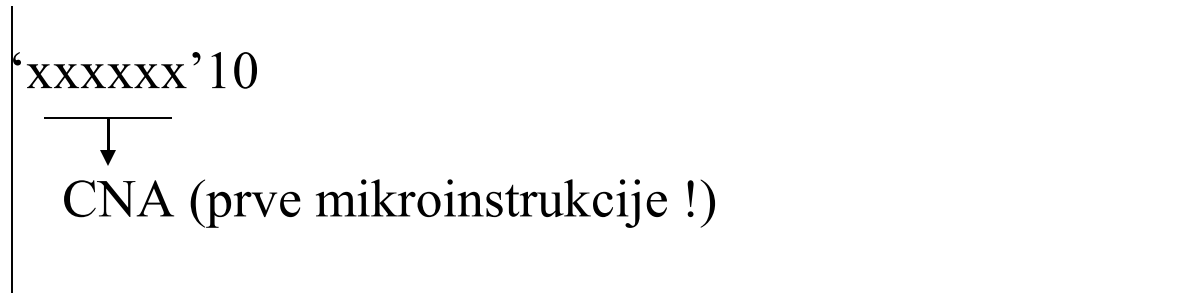


Strojnoj instrukciji CBR (za fazu IZVRŠI) odgovara niz mikrooperacija:

- 1- prijenos sadržaja akumulatora A na lijevu sabirnicu L
- 2- prijenos jediničnog komplementa sadržaja akumulatora B na desnu sabirnicu R
- 3- aktiviranje paralelnog zbrajala ($C_{in} = 1$)
- 4- postavljanje zastavice Z u zavisnosti od rezultata ispitivanja sklopom ZT ($Z \rightarrow 0$ ako je rezultat mikrooperacije 3 nula)
- 5- postavljanje zastavice N (MB(0) određuje stanje zastavice N)
- 6- definiranje adrese slijedeće mikroinstrukcije na temelju polja CNA i $H(6) \leftarrow STAT(o)$ i $H(7) \leftarrow STAT(1) / STAT(0) = Z$; $STAT(1) = N$

OPASKA: ALU nema komparator
mikrooperacije 1-6 određuju 1. mikroinstrukciju

Ako je $A = B$ tada je $STAT(0) = 1$ i $STAT(1) = 0$ te je *adresa slijedeće mikroinstrukcije*:



Mikrooperacije:

- 1' - na lijevu sabirnicu $L = 0$
- 2' - na desnu sabirnicu $R = PC$
- 3' - aktiviranje paralelnog zbrajala ($C_{in} = 1$)
- 4' - $PC + 1$ prenesi sa MB u PC
- 5' - definiranje *adrese slijedeće mikroinstrukcije*:
 'xxxxxx'00 $H(6) \leftarrow 0; H(7) \leftarrow 0;$

Mikrooperacije 1' – 5' određuju 2. mikroinstrukciju

Ako je $A < B$ tada je $STAT(0) = 0$ i $STAT(1) = 1$ te je *adresa slijedeće mikroinstrukcije*:

'xxxxxx'01 ; $H(6) \leftarrow STAT(0), H(7) \leftarrow STAT(1)$

↓
CNA (prve mikroinstrukcije !)

1'' - na lijevu sabirnicu postavi konstantu 2 (iz EMIT polja)

2'' - na desnu sabirnicu postavi PC

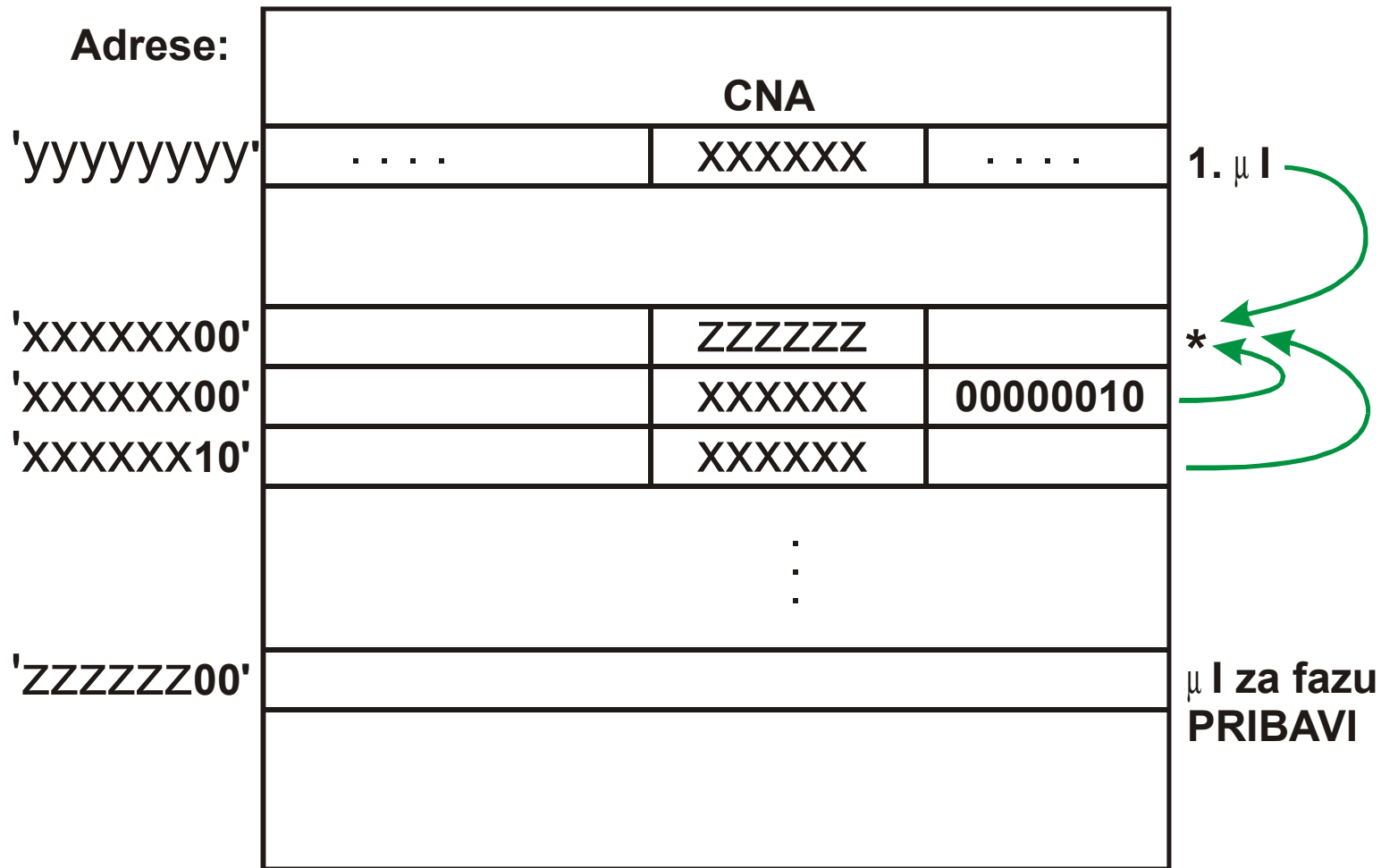
3'' - aktiviraj paralelno zbrajalo ($C_{in} = 0$)

4'' - $PC + 2$ prenesi u PC

5'' - definiranje *adrese slijedeće mikroinstrukcije*:

'xxxxxx'00 $H(6) \leftarrow 0; H(7) \leftarrow 0;$

Mikrooperacije 1'' – 5'' određuju 3. mikroinstrukciju



**MIKROPROGRAMSKA
MEMORIJA
(256 riječi x 32 bita)**

Mikroprogram specificiran u jeziku sličnom CDL-u
(Computer Design Language):

Struktura:

/logički uvjet/ *mikrooperacija1, mikrooperacija2,...* ; komentar

-***ako*** je logički uvjet zadovoljen ***tada*** se izvršava niz mikrooperacija
mikrooperacija1, mikrooperacija2,... i to istodobno

Mikroprogram u jeziku sličnom CDL-u:

/START(ON) * P(0)/	$P_r \leftarrow 0, G \leftarrow 0$; G – upravljački registar koji osigurava pravilan početak sekvence s odgovarajućom adresom u H, P_r – bistabil koji označava fazu PRIBAVI
/G' * P(1)/	$H \leftarrow \text{'yyyyyyyy'}$, $G \leftarrow 1$; P(1) – prijenos adrese u H
/G * P(2)/	$F \leftarrow CM(H)$; P(2) – čitanje iz mikroprogramske memorije /mikroPRIBAVI/
/CA(3) * P(0)/	$L = A$	

/CB(2) * P(0)/ $R = B'$; jedinični komplement
B na desnu sabirnicu

/COP(1) * P(0)/ $C_{in} = 1$; zbroji s $C_{in} = 1$

/CSH(0) * P(0)/ $MB = Q, Q = S$

/CMB(0) * P(0)/ NOP

/CST(3) * P(0)/ (IF(ZT = 0) THEN
(STAT(0) ← 0) ELSE
(STAT(0) ← 1)),
STAT(1) ← MB(0)

$$/ \text{CAB}(2) * \text{P}(1) / \quad \text{H}(6) \leftarrow \text{STAT}(0) \quad ; \text{STAT}(0) = \text{Z}$$

```

/CBB(2) * P(1)/          H(7) ← STAT(1)          ; STAT(1) = N

```

/G * P(1)/ H(0-5) ← F(CNA) ; F(CNA) =
 'XXXXXXXXXX'

/G * P(2)/

$F \leftarrow CM(H)$

; A = B – adresa mikroinstrukcije je ‘xxxxxx10’

/CA(1) * P(0)/

$L = 0$

; L(0-7, 8-15) = 0 –
F(EM)

/CB(3) * P(0)/

$R = PC$

/COP(1) * P(0)/

$C_{in} = 1$

; zbroji s $C_{in} = 1$

/CSH(0) * P(0)/

MB(0) = Q, Q = S

/CMB(3) * P(0)/

PC \leftarrow MB

/CST(0) * P(0)/

NOP

/CAB(0) * P(1)/

H(6) \leftarrow 0

/CBB(0) * P(1)/

H(7) \leftarrow 0

/G * P(1)/

H(0-5) \leftarrow F(CNA)

$$/G * P(2)/$$

$$F \leftarrow CM(H)$$

$$; A < B$$

$$/CA(1) * P(0)/$$

$$L(0-7, 8-15) = 0 - F(EM) \quad ; F(EM) = 2$$

$$/CB(3) * P(0)/$$

$$R = PC$$

$$/COP(0) * P(0)/$$

$$C_{in} = 0$$

$$; \text{zbroji s } C_{in} = 0$$

$$/CMB(3)* P(0)/$$

$$PC \leftarrow MB$$

/CAB(0) * P(1)/

H(6) \leftarrow 0

/CBB(0) * P(1)/

H(7) \leftarrow 0

/G * P(1)/

H(0-5) \leftarrow F(CNA)

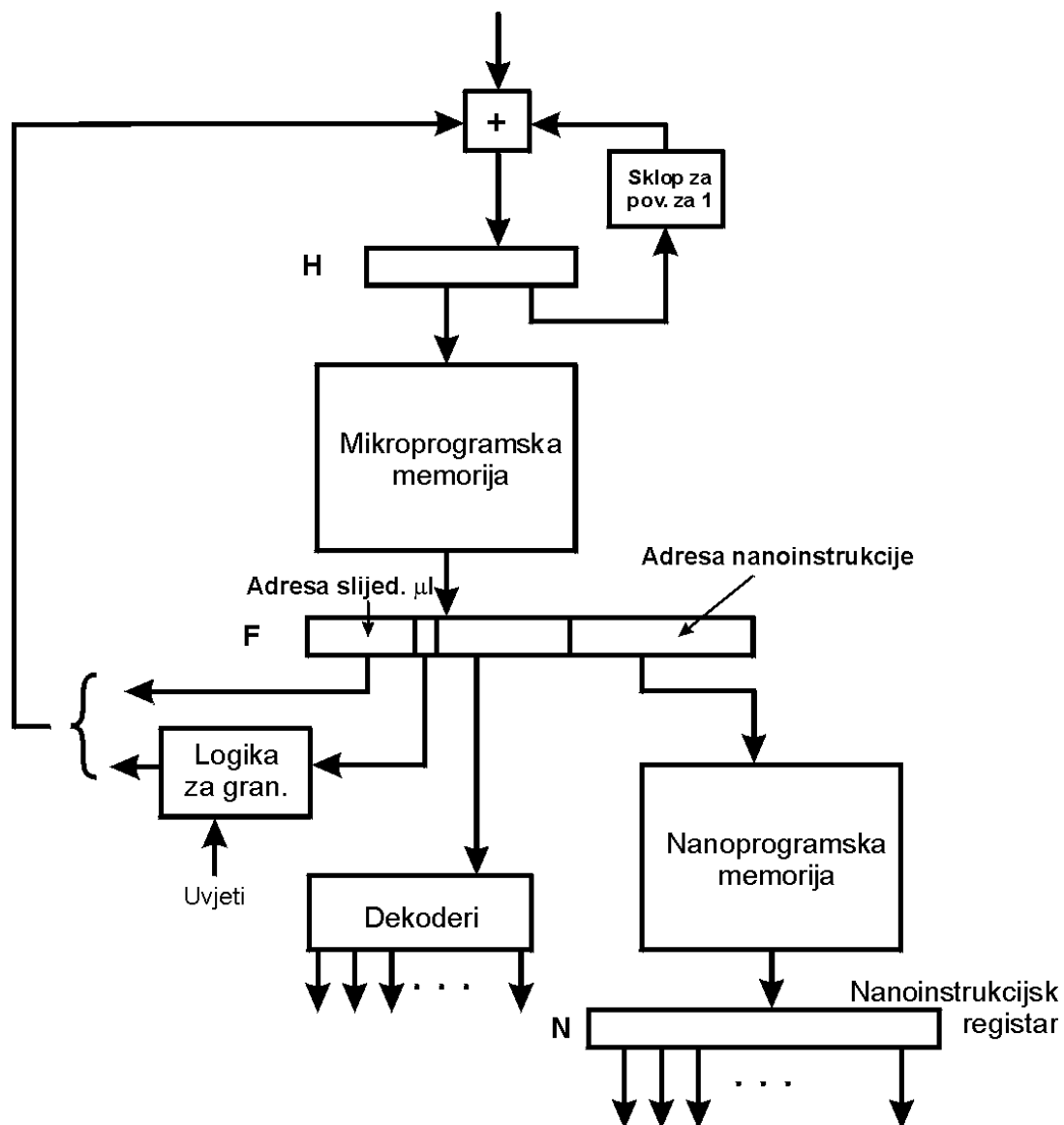
; F(CNA) =
'xxxxxxxx'

/G * P(2)/

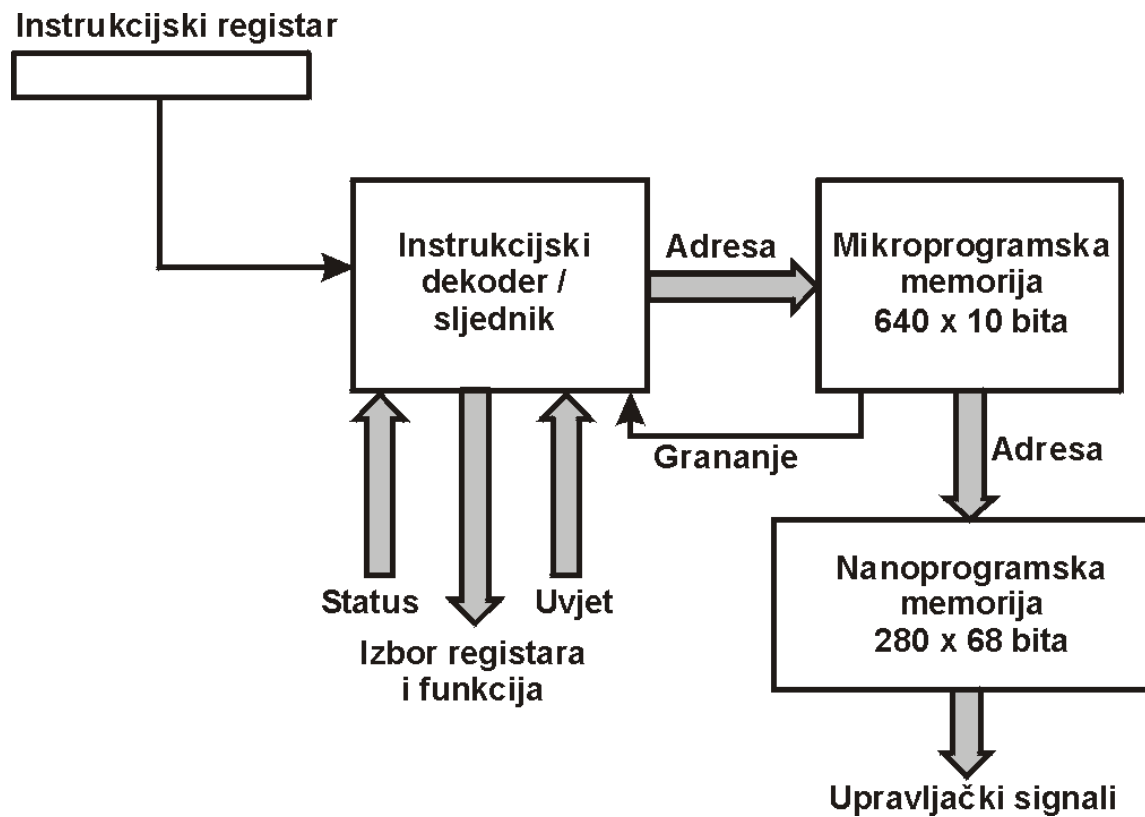
F \leftarrow CM(H)

2	3	2	2	3	2	2	2	CNA 6	8
CA	CB	COP	CSH	CMB	CAB	CBB	CST	X X X X X X	CEM
1 1	0 1 0	0 1	0 0	0 0 0	1 0	1 0	1 1	X X X X X X	0 0 0 0 0 0 0 0
								Z Z Z Z Z Z	CEM
0 1	0 1 1	0 0	0 0	0 1 1	0 0	0 0	0 0	X X X X X X	0 0 0 0 0 0 1 0
0 1	0 1 1	0 1	0 0	0 1 1	0 0	0 0	0 0	X X X X X X	0 0 0 0 0 0 0 0

Organizacija mikroprogramirane jedinice u dvije razine (nanoprogramska jed.)



Organizacija upravljačke jedinice procesora MC 68000



Format upravljačke riječi (mikroriječi)

Upravljačka riječ mora omogućiti:

- 1) grupiranje i dodjeljivanje upravljačkih bitova
- 2) sekvenciranje – “vezivanje” upravljačkih riječi u mikroprogram
- 3) fleksibilnost za reprogramiranje

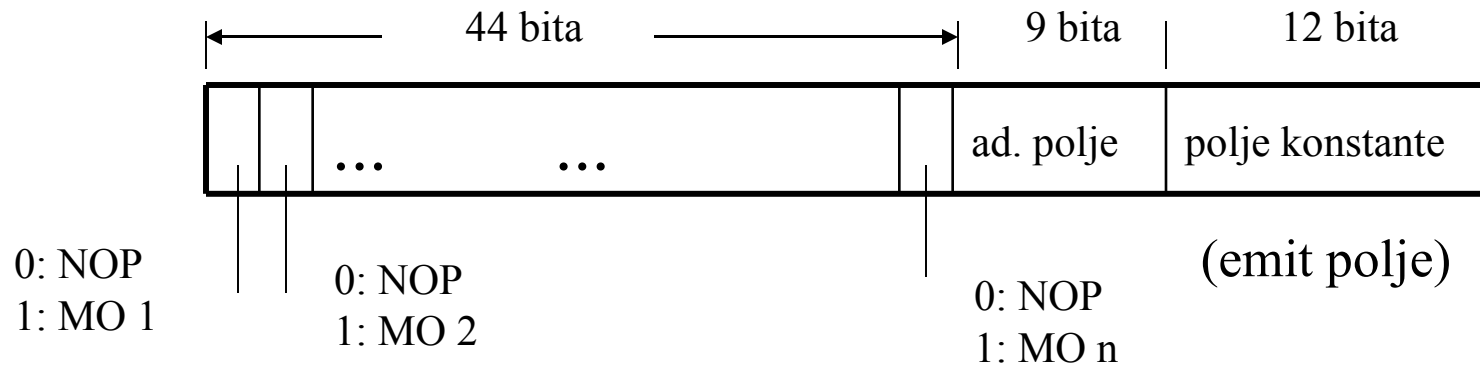
Funkcije trebaju biti obavljene sa:

- a) minimalnim brojem bitova u upravljačkoj riječi
- b) minimalnim brojem riječi u upravljačkoj memoriji
- c) minimalnim vremenom izvođenja

Tehnike (načini) dodjeljivanja upravljačkih bitova – formatiranje upravljačkih riječi

- a) izravno upravljanje
- b) grupiranje bitova
- c) višestruki formati
- d) vertikalno/horizontalno mikroprogramiranje

Izravno upravljanje



- upravljačka jedinica može imati nekoliko stotina mikrooperacija
→ vrlo duge upravljačke riječi /nekoliko stotina bitova/

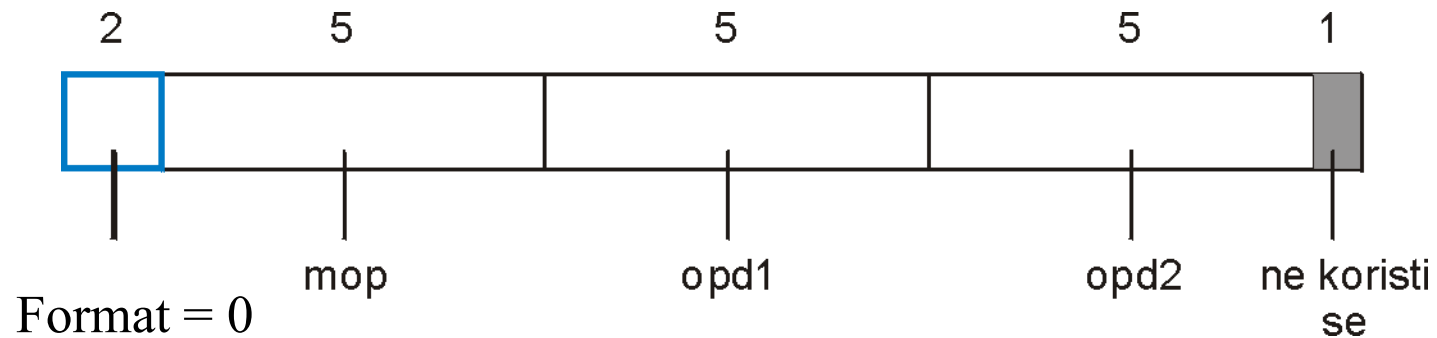
Grupiranje bitova mikroriječi (engl. minimal encoding)

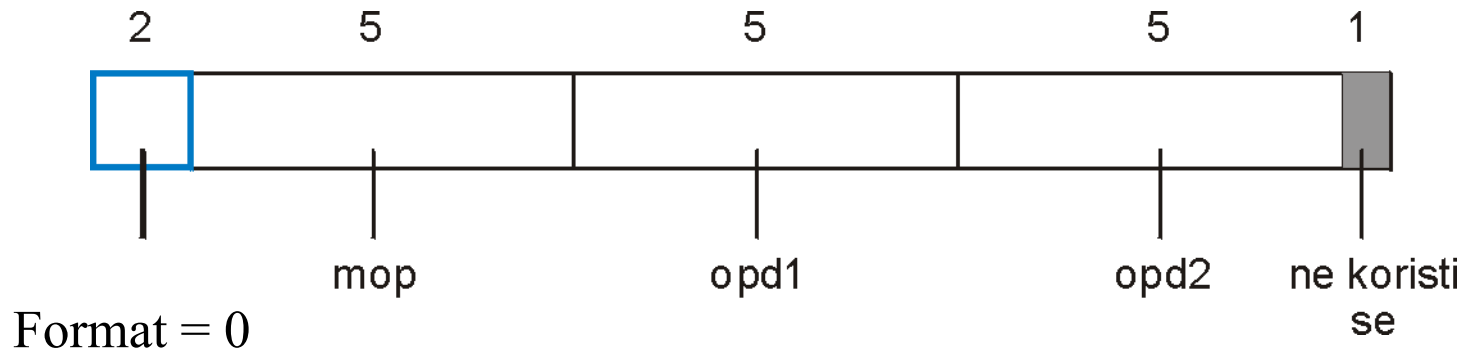
- a) grupiranje mikrooperacija koje se UVIJEK izvode istodobno → jedan upravljački bit upravlja s nekoliko mikrooperacija
- b) grupiranje onih mikrooperacija koje se izvode ISKLJUČIVO jedna u vremenu i to grupiranje se tako izvodi da se mikrooperacije kodiraju poljem bitova odgovarajuće duljine

Višestruki formati

- mikroriječ je kratka: 16 – 32 bita
- svaka mikroinstrukcija specificira jednu ili dvije (maksimalno 4) mikrooperacije

Primjer:





if format = 0

then

$\text{mop} \in \{\text{transfer}\}$ and interpret (**opd1** as source, **opd2** as result)

if format = 1 **then**

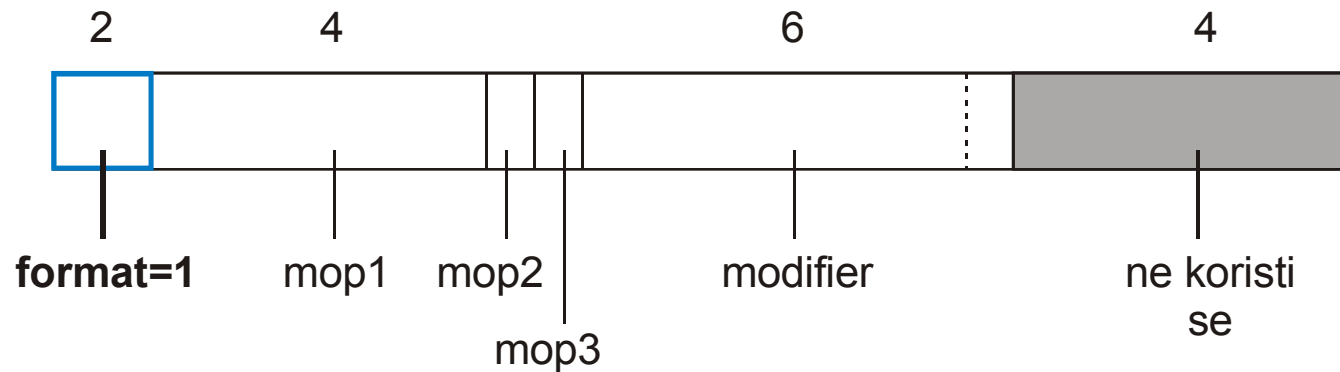
mop1 \in {ADD, SUB, OR, AND, EX_OR, NOT, SHR}

mop2 \in {READ_MM, WRITE_MM}

mop3 \in {INC_PC}

& if mop1 \in {SHL, SHR }

then interpret (modifier as shift amount)



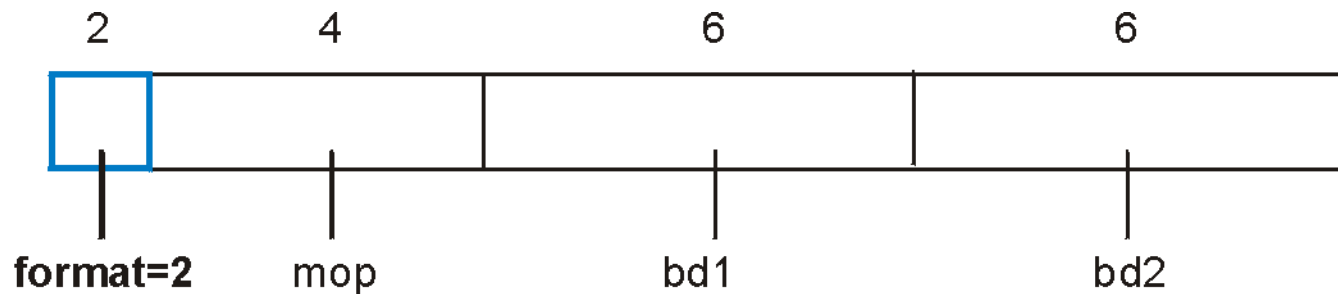
- specifikacija do 3 mo koje se mogu istodobno izvesti

if format = 2

then

mop \in {MASK}

and interpret (bd1 as bound i_1 , bd2 as bound i_2)



AOUT := AIL mask AIR [$i_1 \dots i_2$]

bit pozicije u AIL od i_1 pozicije (ulijevo) i i_2 pozicije (udesno)

prenose se bez izmjene;

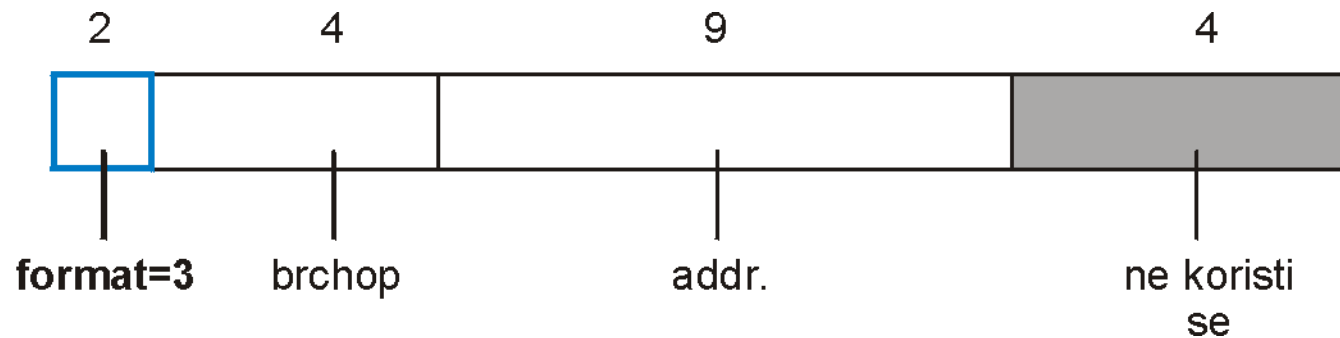
bit pozicije od i_1 do i_2 poprimaju u rezultatu vrijednost AIR registra

if format = 3

then

brchop $\in \{BN, BPZ, BU, BISI, BNISI\}$

and interpret (addr asbranch address)



Horizontalno mikroprogramiranje:

“horizontalna” mikroinstrukcija ima jednu ili više od sljedećih značajki:

4. mikroinstrukcija omogućava različite resurse (npr., funkcionalne jedinice, putove podataka) u računalu tako da se oni upravljaju nezavisno – mikroinstrukcija određuje nekoliko istodobno izvodljivih instrukcija
7. mikroinstrukcija je duga riječ – npr., od 64 do nekoliko stotina bitova
8. horizontalne mikroinstrukcije dopuštaju (mikro)programeru upravljanje na razini pojedine mikrooperacije

Vertikalno mikroprogramiranje

3. mikroinstrukcije obično specificiraju jednu li dvije mikrooperacije; ne dopuštaju (mikro)programeru iskorištavanje potencijalnog paralelizma u procesoru
6. mikroinstrukcije su relativno male duljine – od 16 do 32 bita
7. jedna mikroinstrukcija može prouzrokovati da određen skup ili slijed mikrooperacija bude pobuđen.