

4. Mikroprogramiranje

- Osnovni pojmovi i Wilkesova izvorna shema
- Model mikroprogramiranog procesora
- Primjer mikroprograma (instrukcija CBR)
- Formati upravljačkih riječi

Strojna instrukcija - element instrukcijske arhitekture računala:

- građevna jedinica korisničkih programa, npr: `inc $05ff`

Mikrooperacija - element unutrašnje organizacije (mikroarhitekture):

- elementarna (nedjeljiva) operacija
- izravno i u potpunosti sklopovski podržana
- građevna jedinica strojnih instrukcija
- primjeri:
 - prijenos podataka (sadržaja registara) između dva registra
 - aktiviranje sklopa u ALU
 - brisanje ili postavljanje bistabila (zastavice)

Odnos između strojne instrukcije **ADD** i pripadnih mikrooperacija:

ADD: \longrightarrow MAR \leftarrow PC
 MDR \leftarrow M(MAR)
 IR \leftarrow MDR(I)
 PC \leftarrow PC + 1
 MAR \leftarrow MDR(A)
 MDR \leftarrow M(MAR)
 . . .

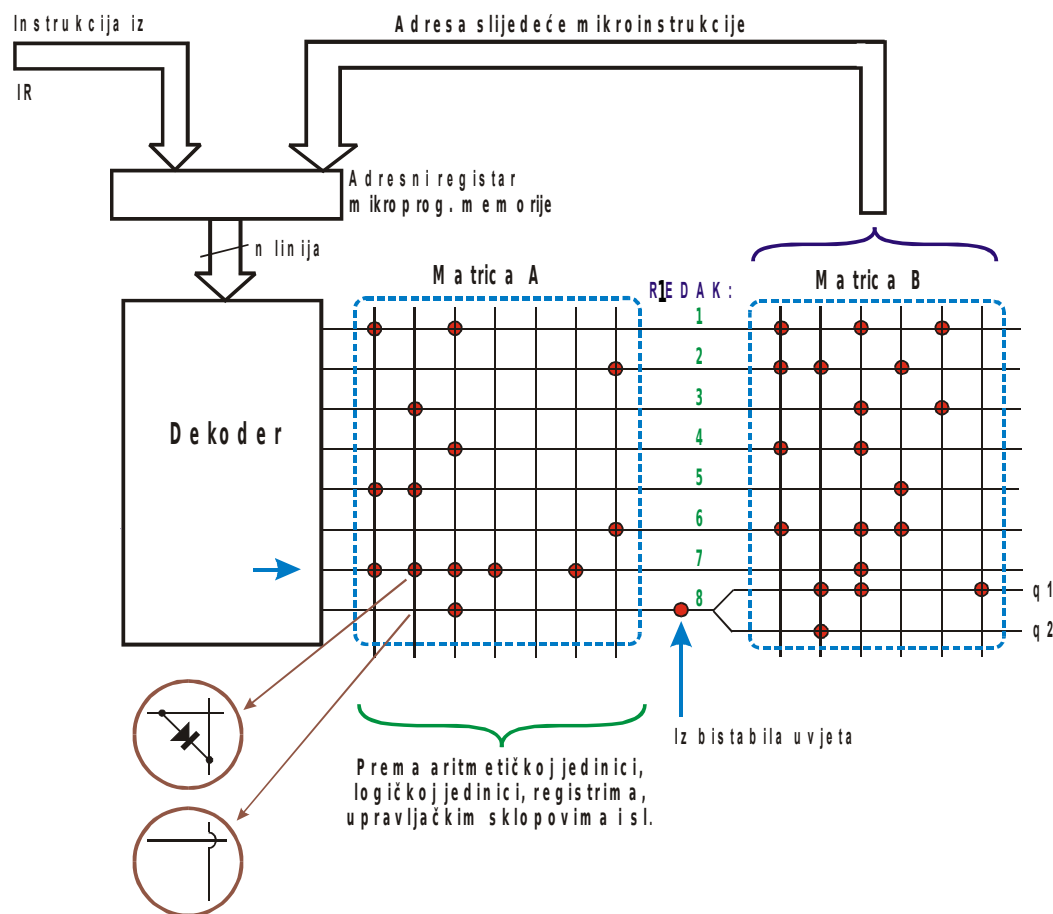
Mikroprogramiranje – pristup izvedbi upravljačke jedinice

- posebno prikladno za CISC procesore sa složenim instrukcijama:
 - **MC68000**: DBGT D3, label
 - **x86**: LOOP label
 - **x86**: REP MOVSB
 - **VAX**: poly X, #2, PTABLE
 - **VAX**: movc5 <srclen>, <srcaddr>, <fill>, <dstlen>, <dstaddr>
- **osnovna ideja**: dinamički prevoditi instrukcije u niz **mikroinstrukcija**
- sustavna metoda prijelaza s instrukcijske arhitekture na mikroarhitekturu
- **neizrečena pretpostavka**: razvoj arhitekture ide prema premošćivanju semantičkog jaza između strojnog i višeg programskog jezika (**krivo!**)
- vrijeme je pokazalo da se najbolja performansa postiže favoriziranjem brzine izvođenja najjednostavnijih instrukcija (za koje se pokazalo da su **najčešće**)

Malo terminologije:

- **Mikroinstrukcija** – kodna riječ koja predstavlja jednu ili više mikrooperacija
- **Mikroprogram** – slijed mikroinstrukcija
- pohranjen u brznoj upravljačkoj (mikroprogramskoj) memoriji (ROM ili RAM)
- alternativni nazivi elemenata mikroprograma (mikroinstrukcija)
 - **mikroriječ** (engl. Microword)
 - **upravljačka riječ** (engl. Control word)
- Mikroprograme klasificiramo kao **firmware** (ugrađena programska podrška)
- **firmware**: programi koji izravno upravljaju radom sklopovlja

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



Povijesni razvoj i perspektiva:

Mikroprogramiranje: dominantan pristup izvedbe upravljanja 1960-1990

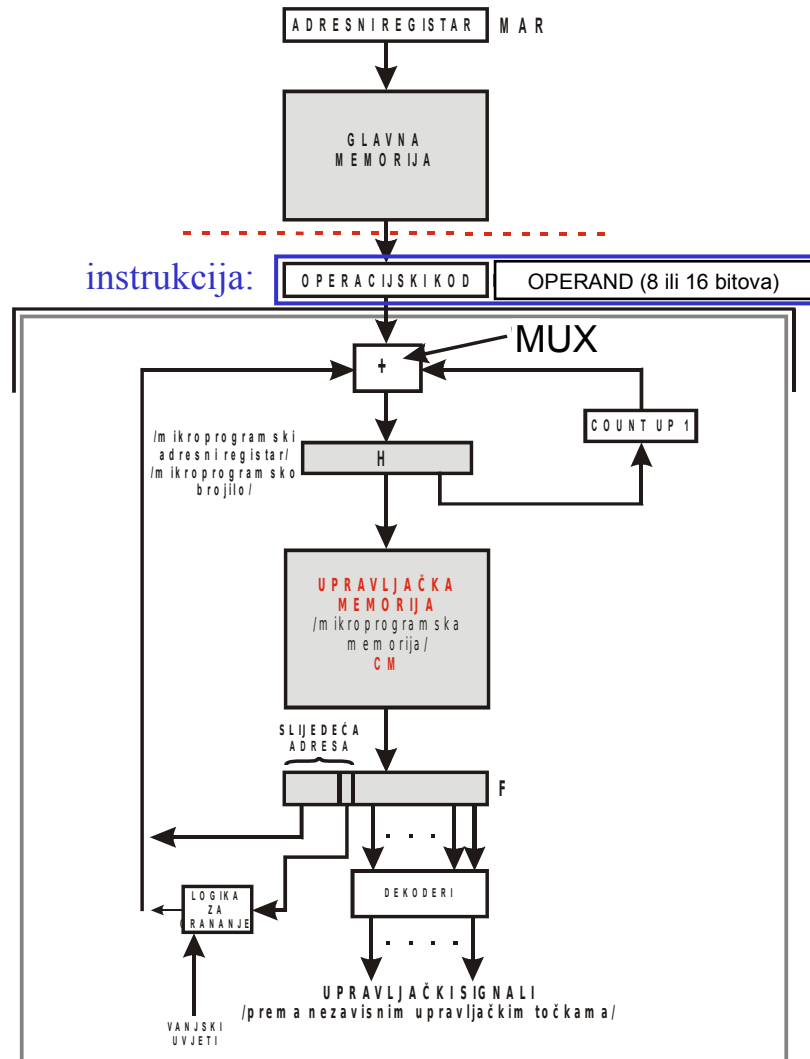
- rane 1950, diodne matrice: $0,5 \mu\text{s}$ vs $10 \mu\text{s}$ RAM (magnetni bubanj)
- kasne 1950, diodne matrice prespore: $0,5 \mu\text{s}$ vs $1 \mu\text{s}$ RAM (feritna jezgra)
- 1960: prodor zbog pojave brzih bipolarnih memorija (ECL)
- 1990: povlačenje zbog
 - RISC filozofije (MIPS, SPARC, DEC, ARM, P6)
 - CAD programa,
 - brzih priručnih memorija

Ipak, neke instrukcije modernih procesora implementirane mikroprogramom (npr, x86: REP MOVSB; CUID)

Instrukcije RISC i VLIW (Itanium) su srodne mikroinstrukcijama (!)

- izvode se u jednom periodu signala takta
- instrukcije RISC su srodne *vertikalnim* mikroinstrukcijama
- instrukcije VLIW su srodne *horizontalnim* mikroinstrukcijama

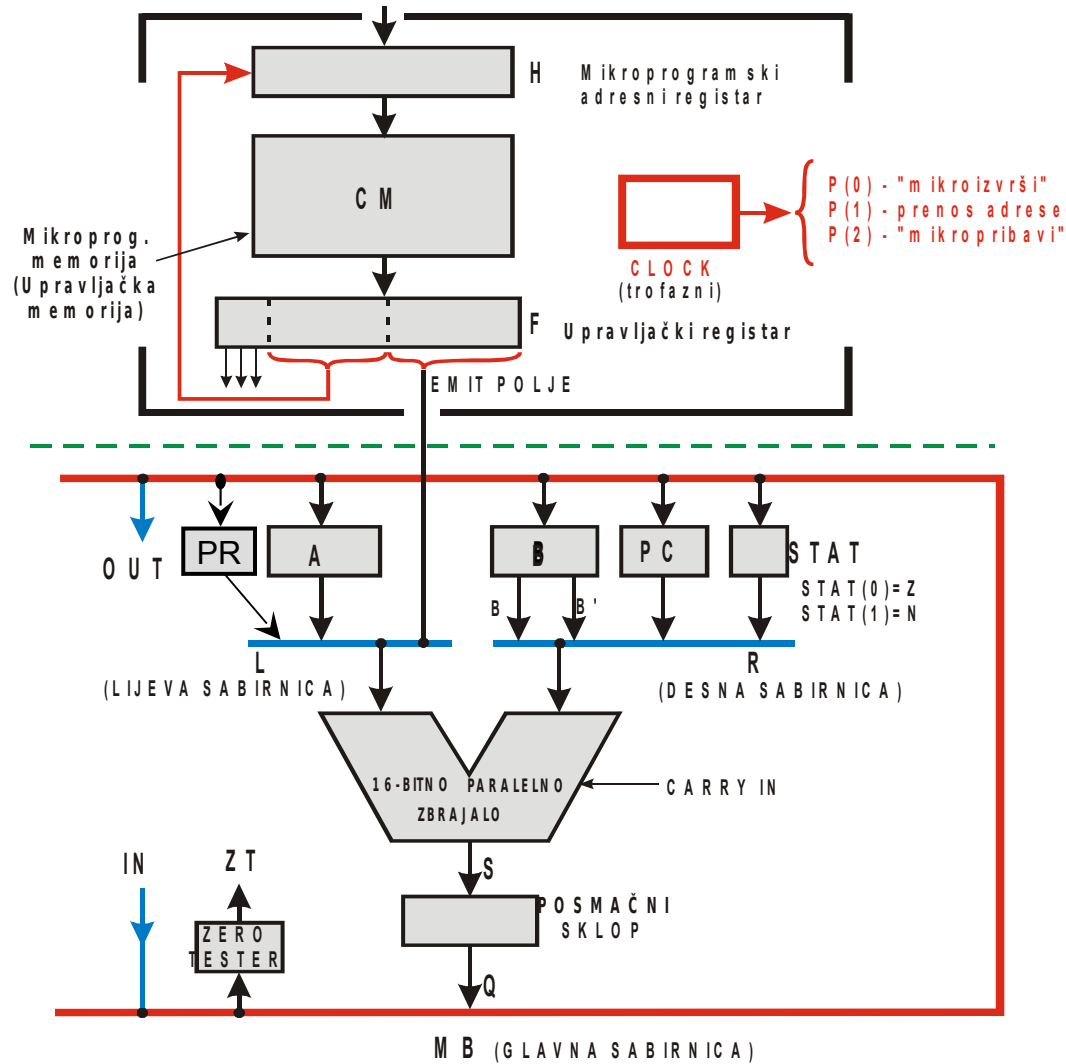
Struktura mikroprogramirane upravljačke jedinice (cca 1970)



Načini dobivanja adrese sljedeće mikroinstrukcije:

1. povećanjem sadržaja μ prog. adresnog registra H (countup 1)
2. prijenosom adresnog polja upravljačke riječi (CNA) u H
 - uz eventualno korištenje logike za grananje
2. izravnim upisom operacijskog koda makroinstrukcije

Model mikroprogramiranog procesora

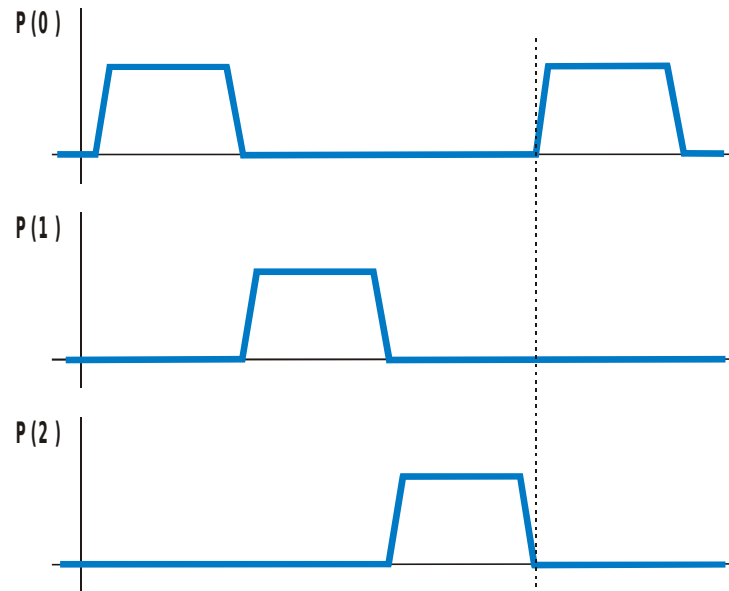


Mikroprogramirana upravljačka jedinica ima **trofazni signal** vremenskog vođenja:

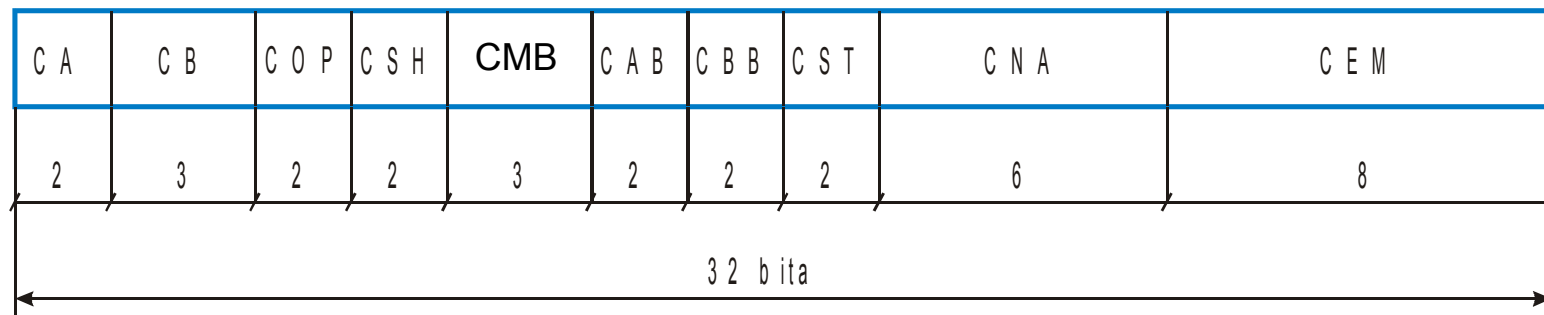
faza P(0) – izvršavanje mikroinstrukcija (“mikroizvrši”)

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

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



Format upravljačke (mikroriječi) riječi:



CA – veza sa sabirnicom L

CB – veza sa sabirnicom R

COP – mikrooperacije paralelnog 16-bitnog zbrajala

CSH – 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

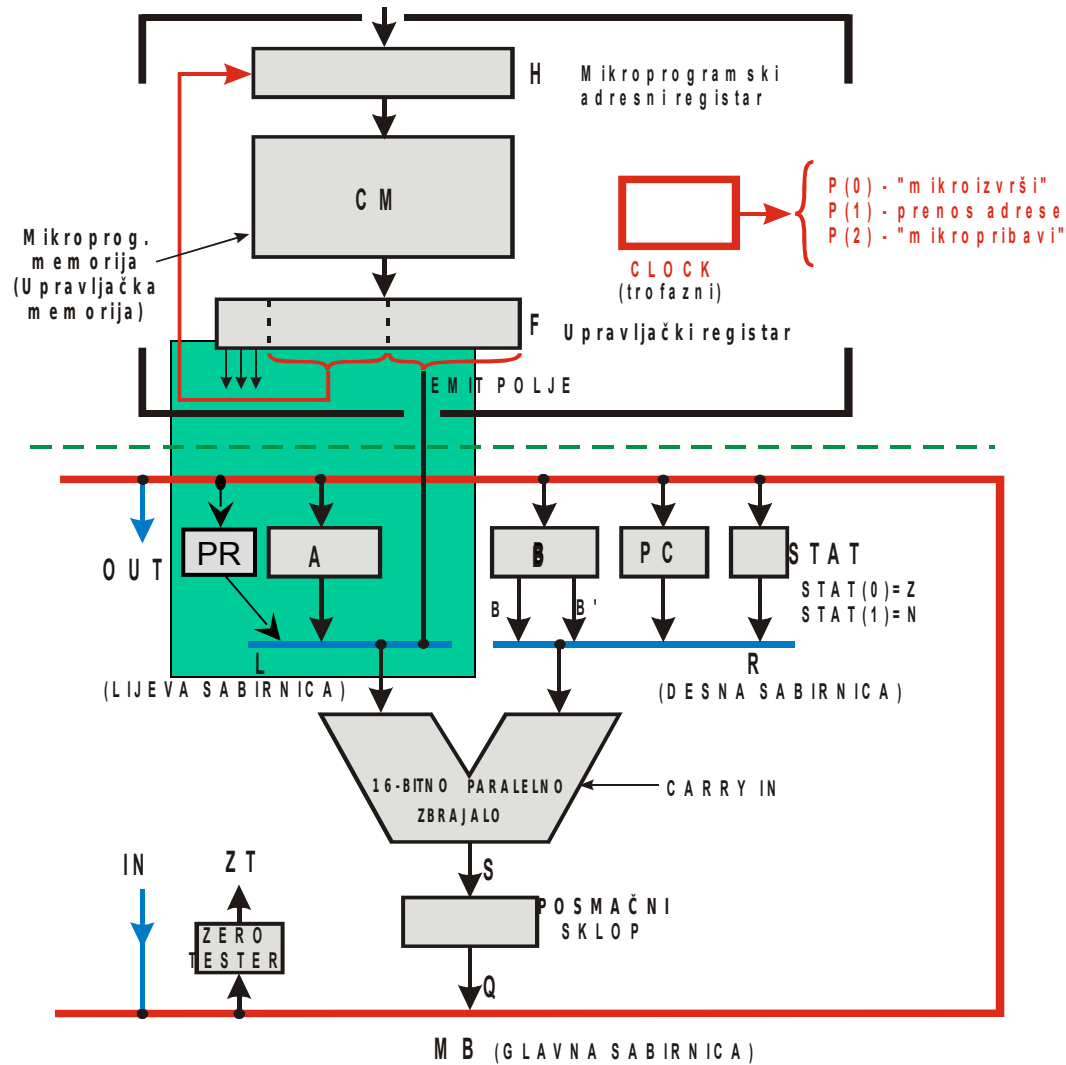
CNA – polje adrese slijedeće mikroinstrukcije

CEM – 8-bitna konstanta /emit polje/

CA – veza sa sabirnicom L

00	$L \leftarrow PR$; na L se smješta privremeni registar PR (u PR se prosljeđuje 16-bitni izravni operand instrukcije , ako ga ima)
01	$L(0-7,8-15) \leftarrow 0 - F(EM)$; mikroinstrukcijska konstanta se smješta na 8 manje značajnih linija (LSB) sabirnice L
10	$L(0-7, 8-15) \leftarrow F(EM) - 0$; mikroinstrukcijska konstanta se smješta na 8 značajnijih linija (MSB) sabirnice L
11	$L \leftarrow A$; na L se smješta sadržaj akumulatora A

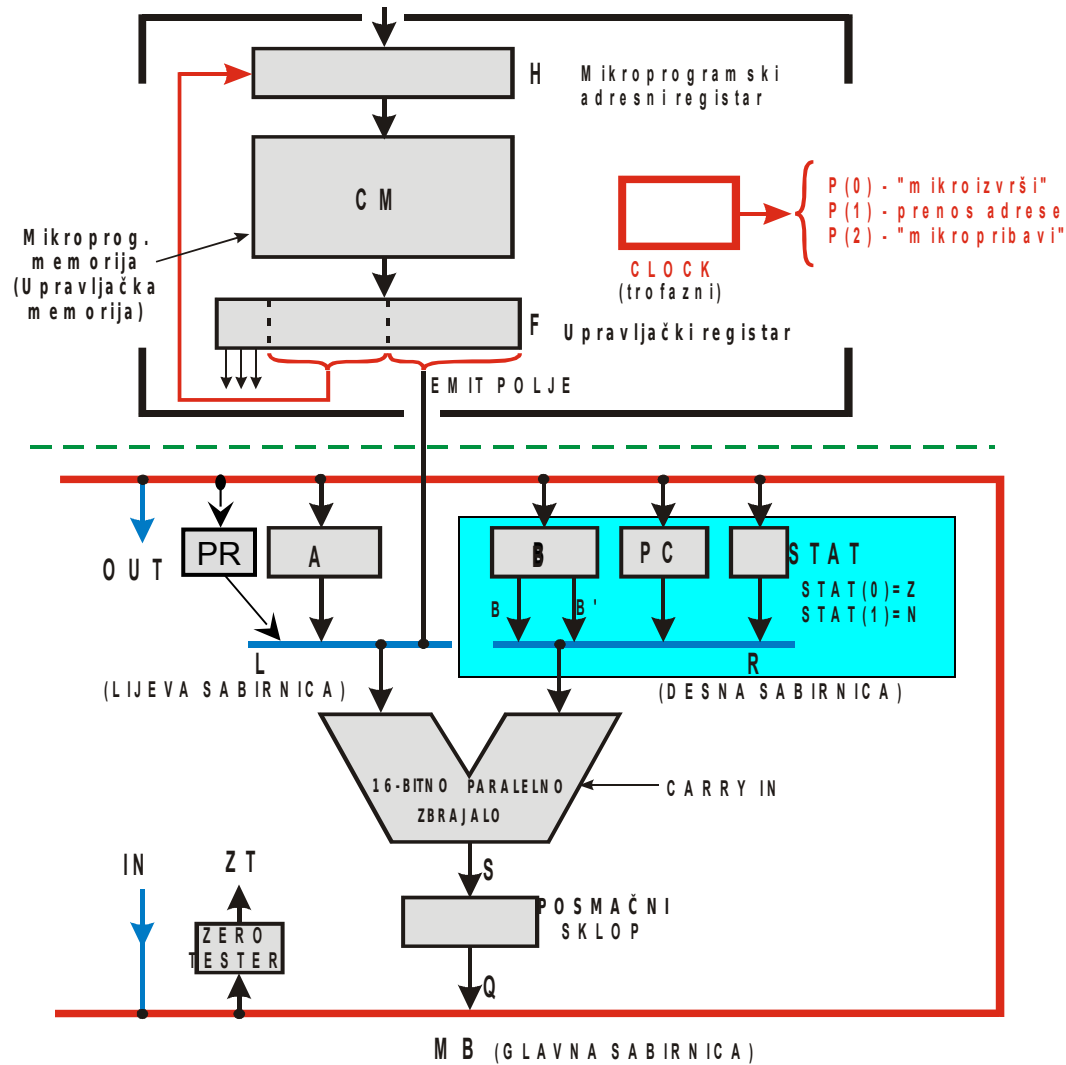
Model mikroprogramiranog procesora – polje CA



CB – veza sa sabirnicom R

000	$R \leftarrow 0$
001	$R \leftarrow B$; prijenos akumulatora B na desnu sabirnicu (R)
010	$R \leftarrow B'$; prijenos jediničnog komplementa sadržaja akumulatora B
011	$R \leftarrow PC$
100	$R \leftarrow STAT$
101	ne koristi se
...	ne koristi se
111	ne koristi se

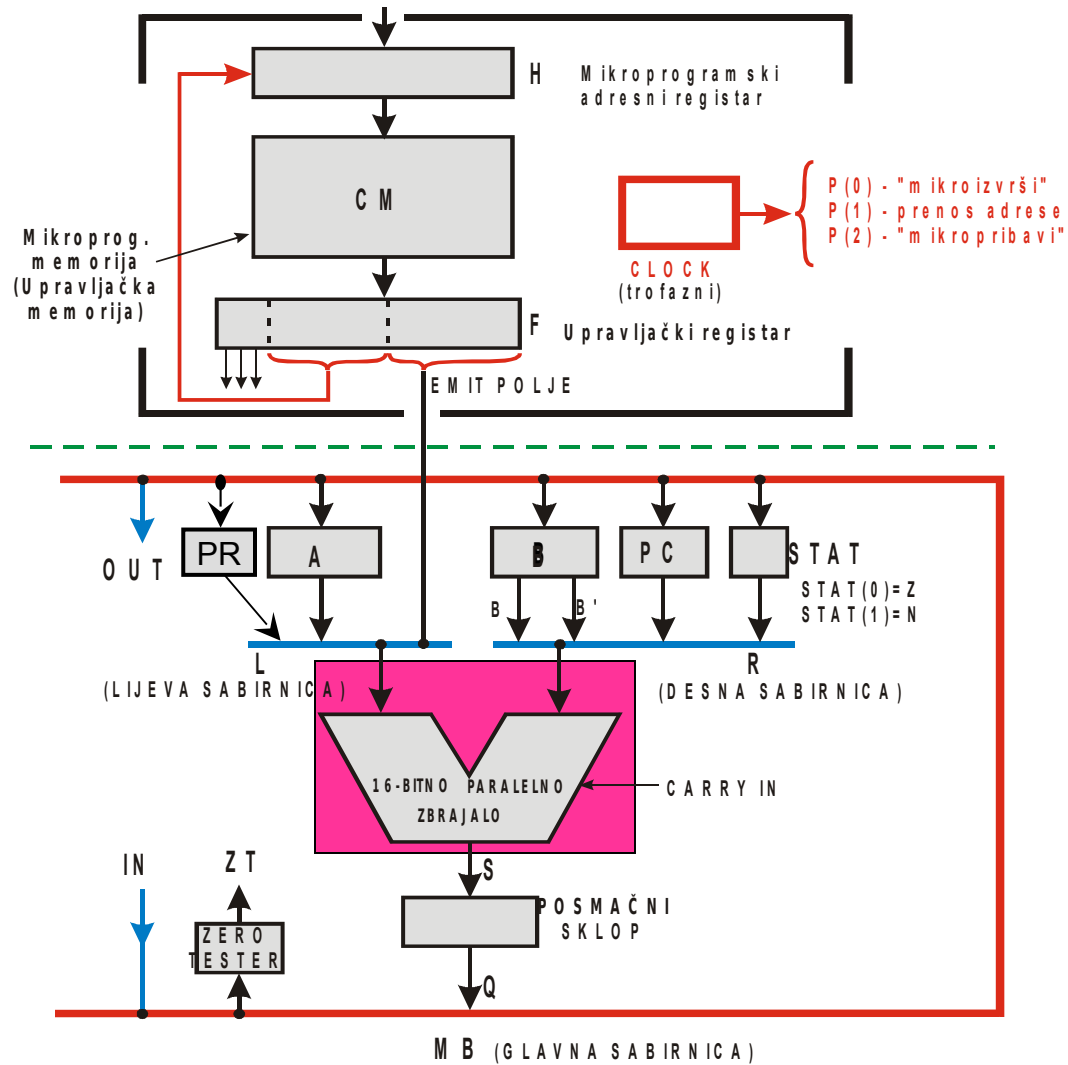
Model mikroprogramiranog procesora – polje CB



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

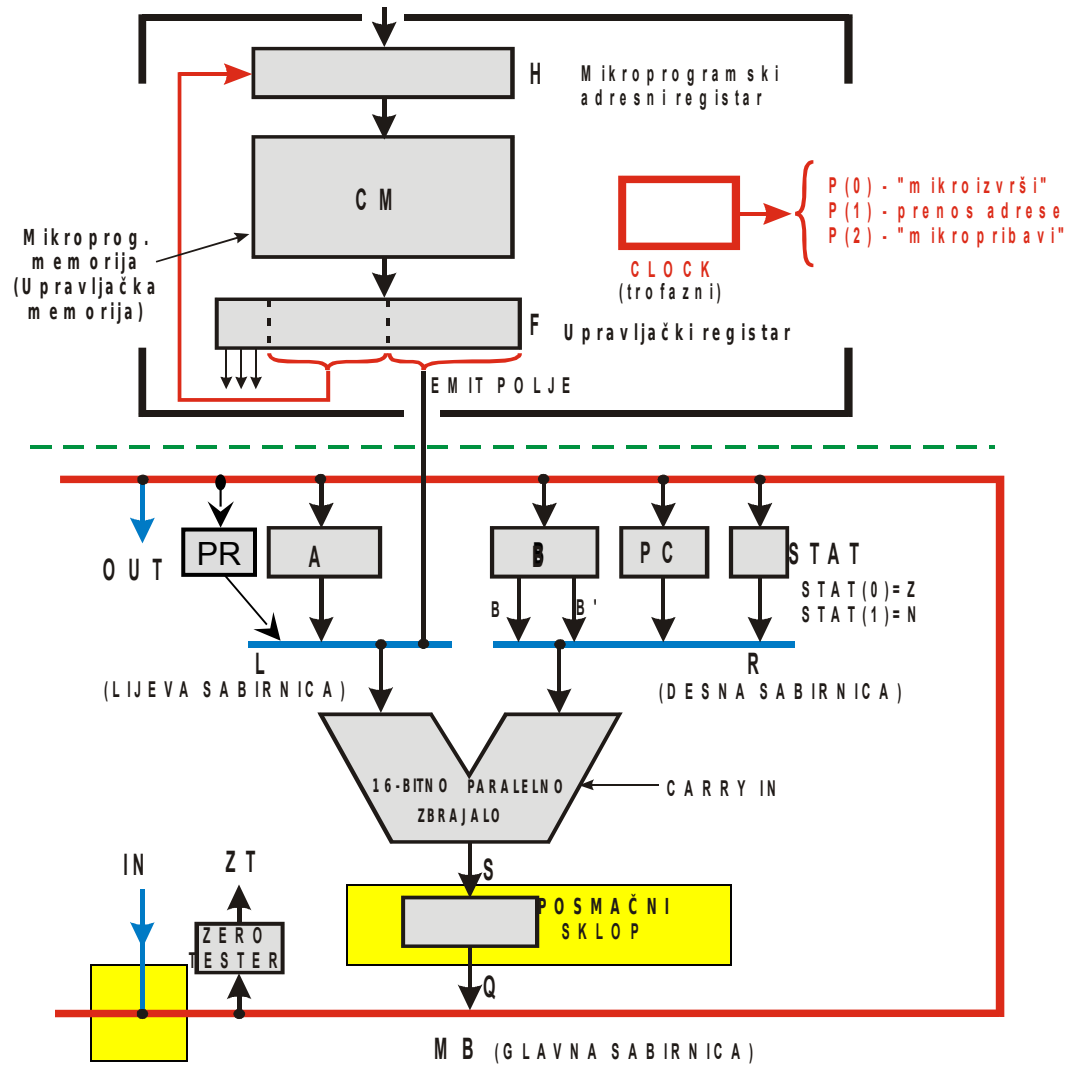
Model mikroprogramiranog procesora – polje COP



CSH – upravljanje pristupom glavnoj sabirnici

00	$MB \leftarrow Q, Q \leftarrow S$; nema posmaka
01	$MB \leftarrow Q, Q \leftarrow \text{shr } S$; posmak udesno
10	$MB \leftarrow Q, Q \leftarrow \text{shl } S$; posmak ulijevo
11	$MB \leftarrow IN$

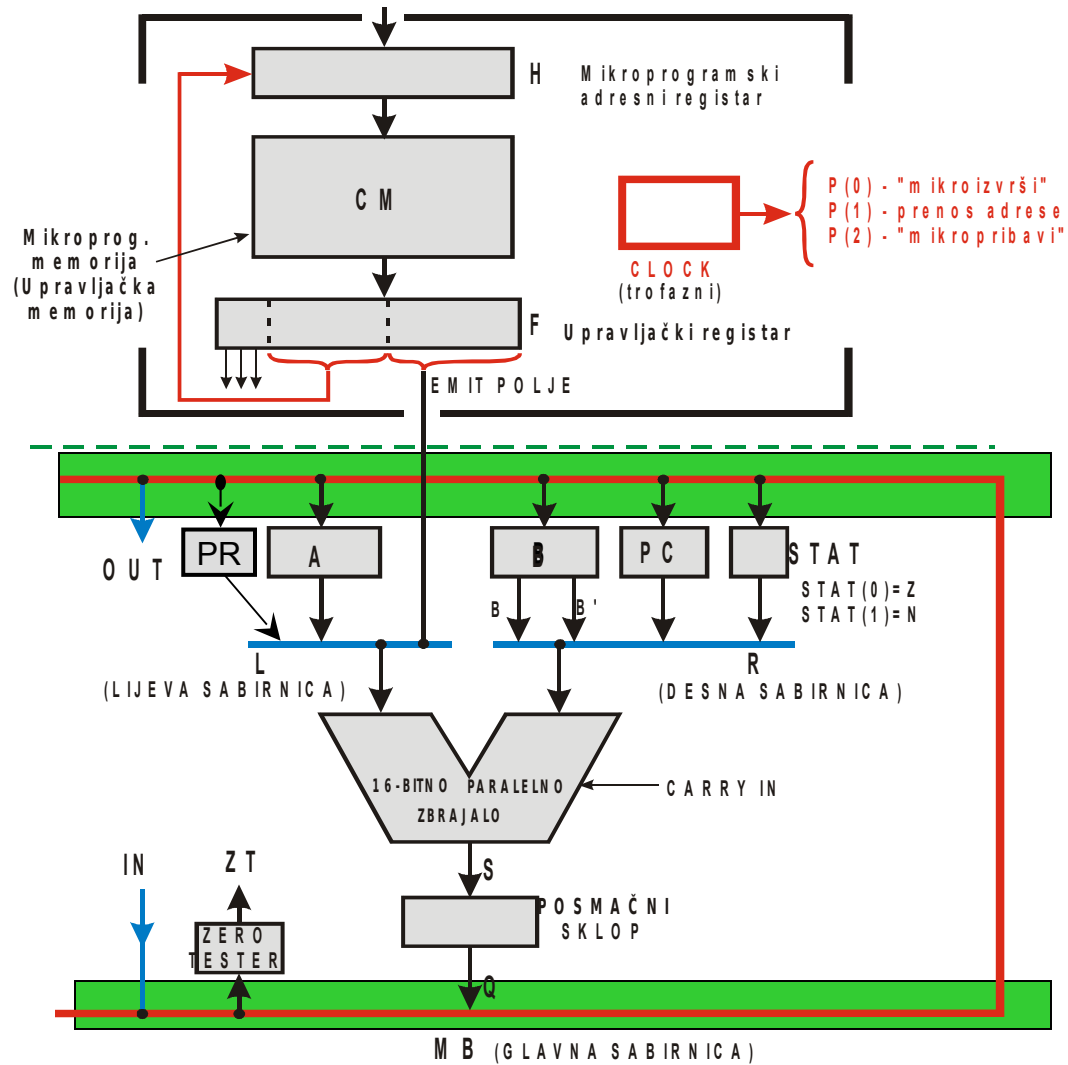
Model mikroprogramiranog procesora – polje CSH



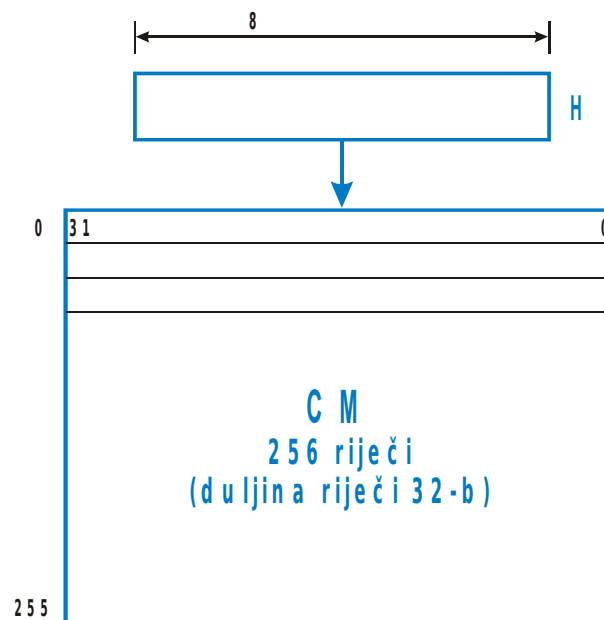
CMB – veza glavne sabirnice (MB) i ostalih komponenti

000	nema prijenosa
001	$A \leftarrow MB$
010	$B \leftarrow MB$
011	$PC \leftarrow MB$
100	$STAT \leftarrow MB$
101	$OUT \leftarrow MB$
110	$PR \leftarrow MB$
111	ne koristi se

Model mikroprogramiranog procesora – polje CMB

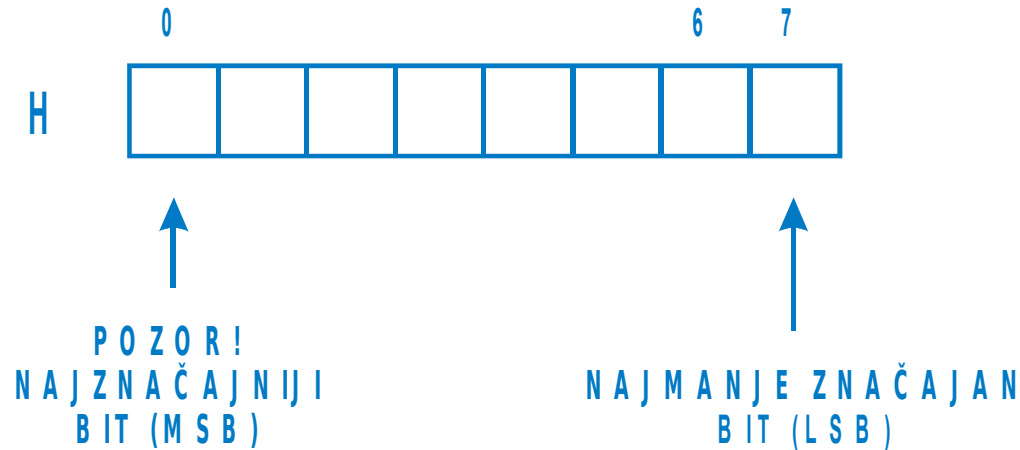


CM – upravljačka, mikroprogramska memorija



Mikroprogramski adresni
registar H

Mikroprogramski adresni registar H



Šest najznačajnijih bitova H(0-5) određeni su poljem CNA (polje adrese slijedeće instrukcije)

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

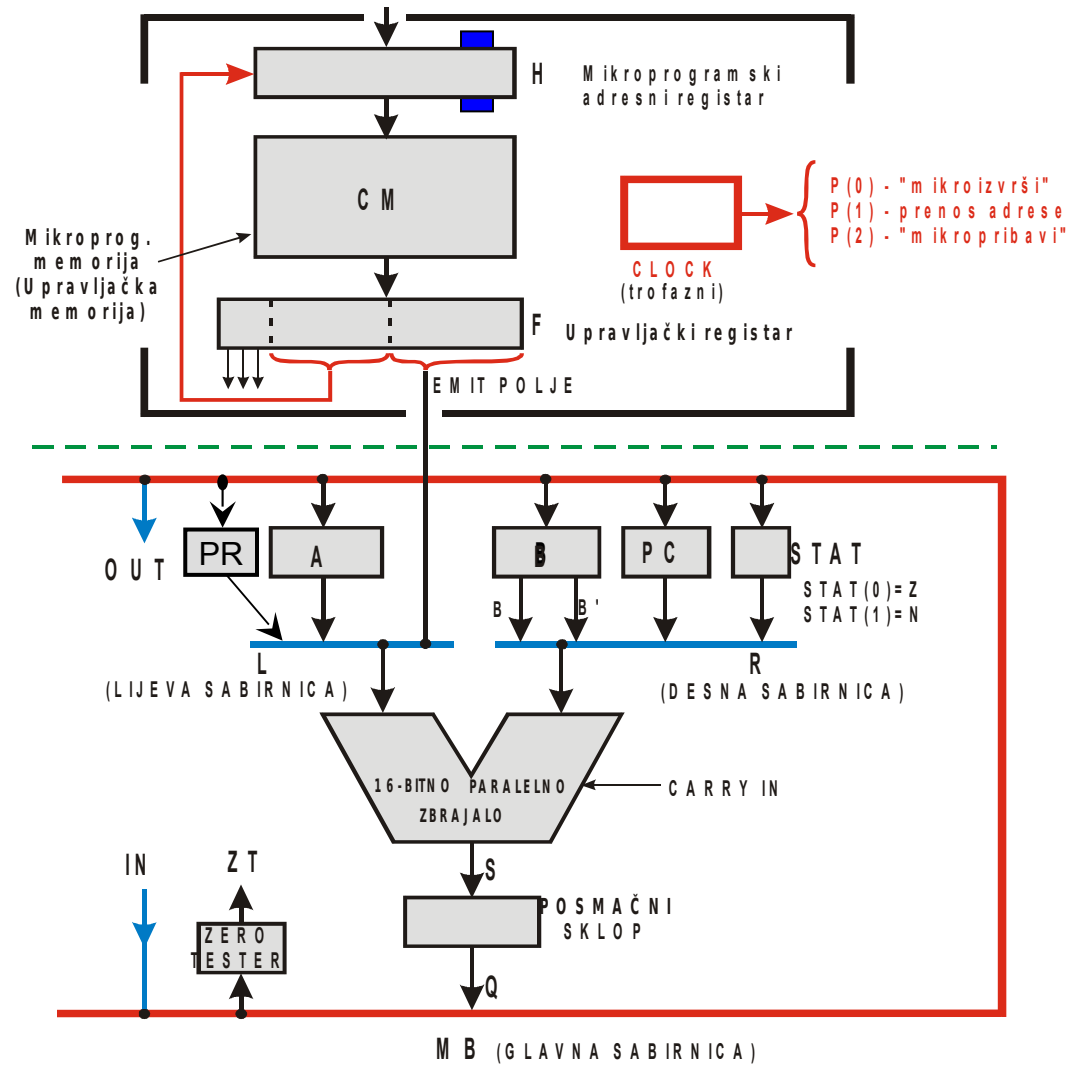
CAB – utjecaj na H(6)

00	$H(6) \leftarrow 0$
01	$H(6) \leftarrow 1$
10	$H(6) \leftarrow \text{STAT}(0)$
11	$H(6) \leftarrow \text{MB}(15)$; MB(15) je najmanje značajni bit 16-bitne riječi na sabirnici MB

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

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

Model mikroprogramiranog procesora – polje CAB



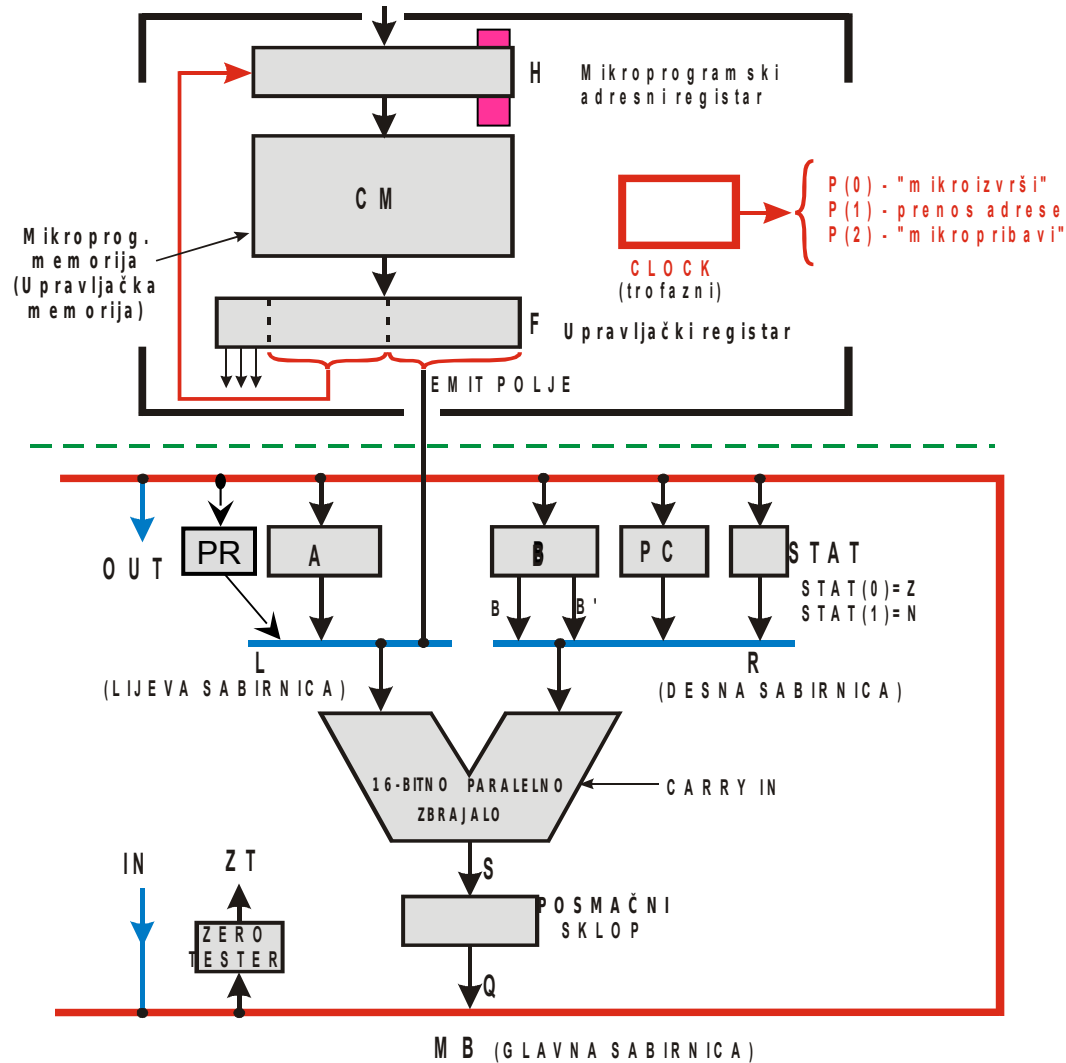
CBB – utjecaj na H(7)

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 na sabirnici MB

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

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

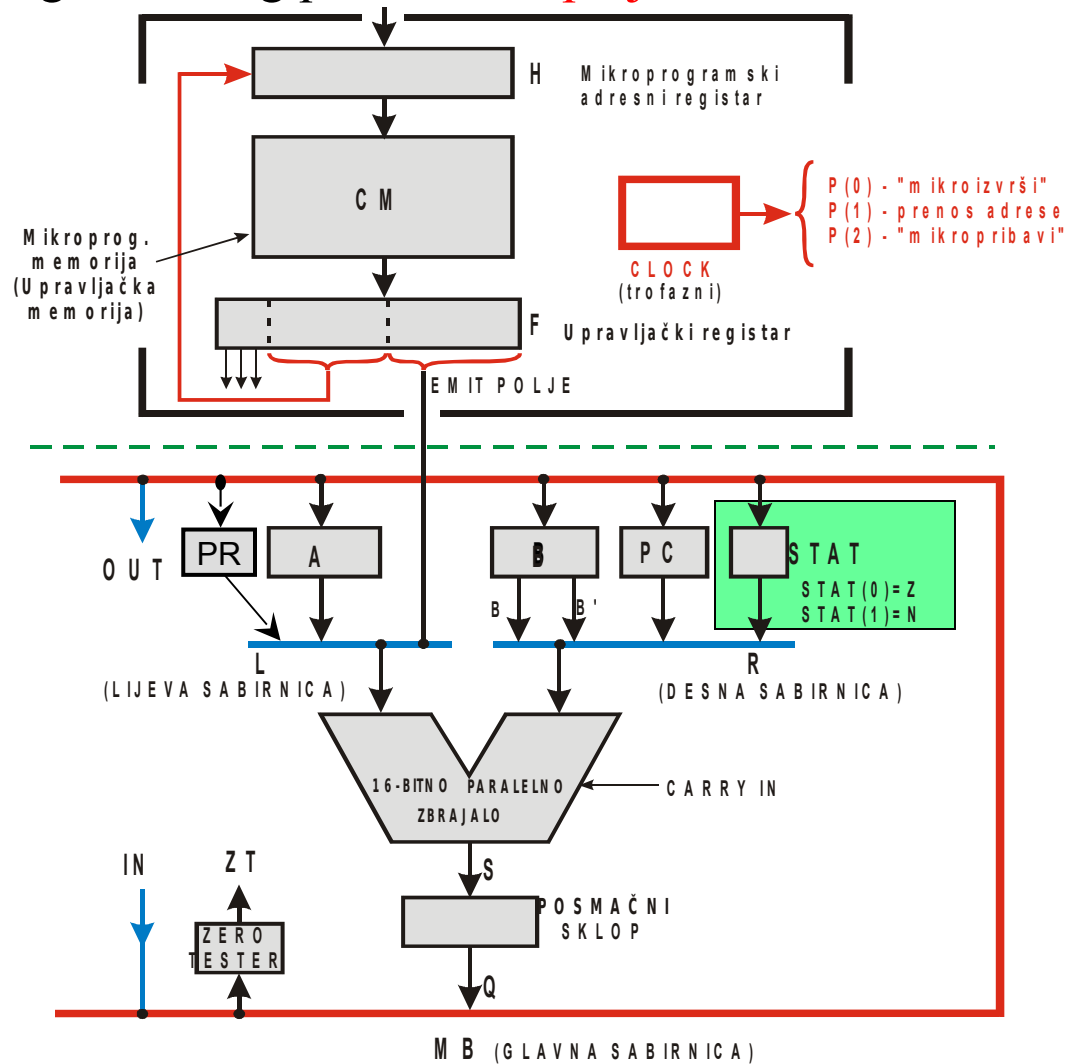
Mikroprogramirani model mikroprocesora – polje CBB



CST – utjecaj na STAT

00	nema utjecaja
01	$\text{STAT}(0) \leftarrow \text{ZT}$
10	$\text{STAT}(1) \leftarrow \text{MB}(0)$
11	$\text{STAT}(0) \leftarrow \text{ZT}$ $\text{STAT}(1) \leftarrow \text{MB}(0)$

Model mikroprogramiranog procesora – polje CST



Tablica vrijednosti bitova ZT i MB(0) u ovisnosti o rezultatu:

ZT (izlaz iz sklopa za ispitivanje nule)	MB(0) Najznačajniji bit sabirnice MB	
0	0	rezultat > 0
0	1	rezultat < 0
1	0	rezultat = 0
1	1	nemoguća kombinacija

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 predloženi 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/

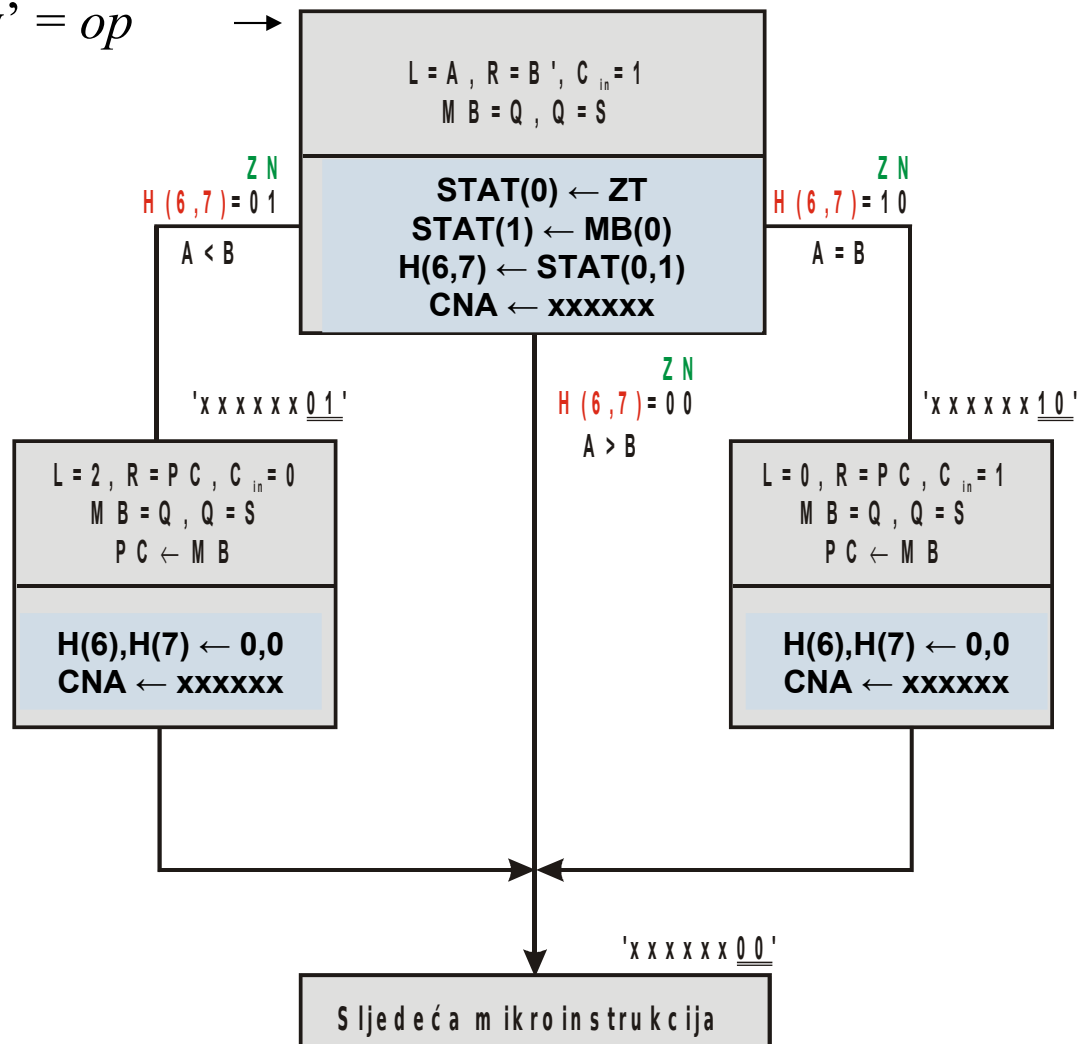
Strojna instrukcija: CBR ; instrukcija uvjetnog grananja

if $A = B$: $PC \leftarrow PC + 1$

if $A < B$: $PC \leftarrow PC + 2$

Dijagram toka

'yyyyyyyy' = *op*

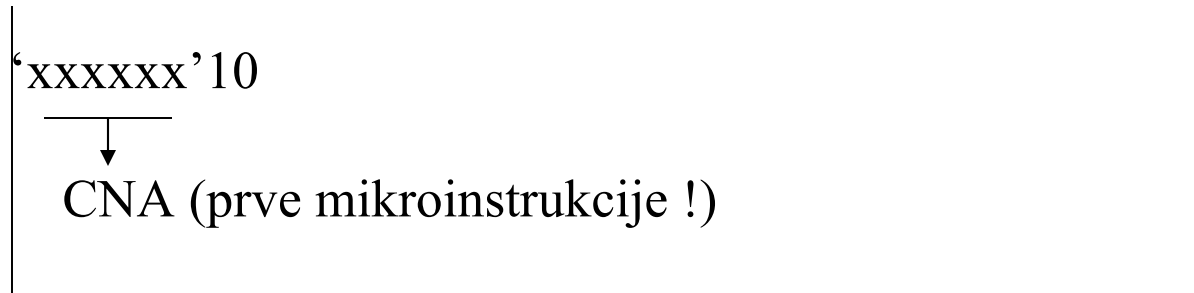


Strojnoj instrukciji CBR (za fazu IZVRŠI) odgovara niz mikrooperacija:

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

Napomena: 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

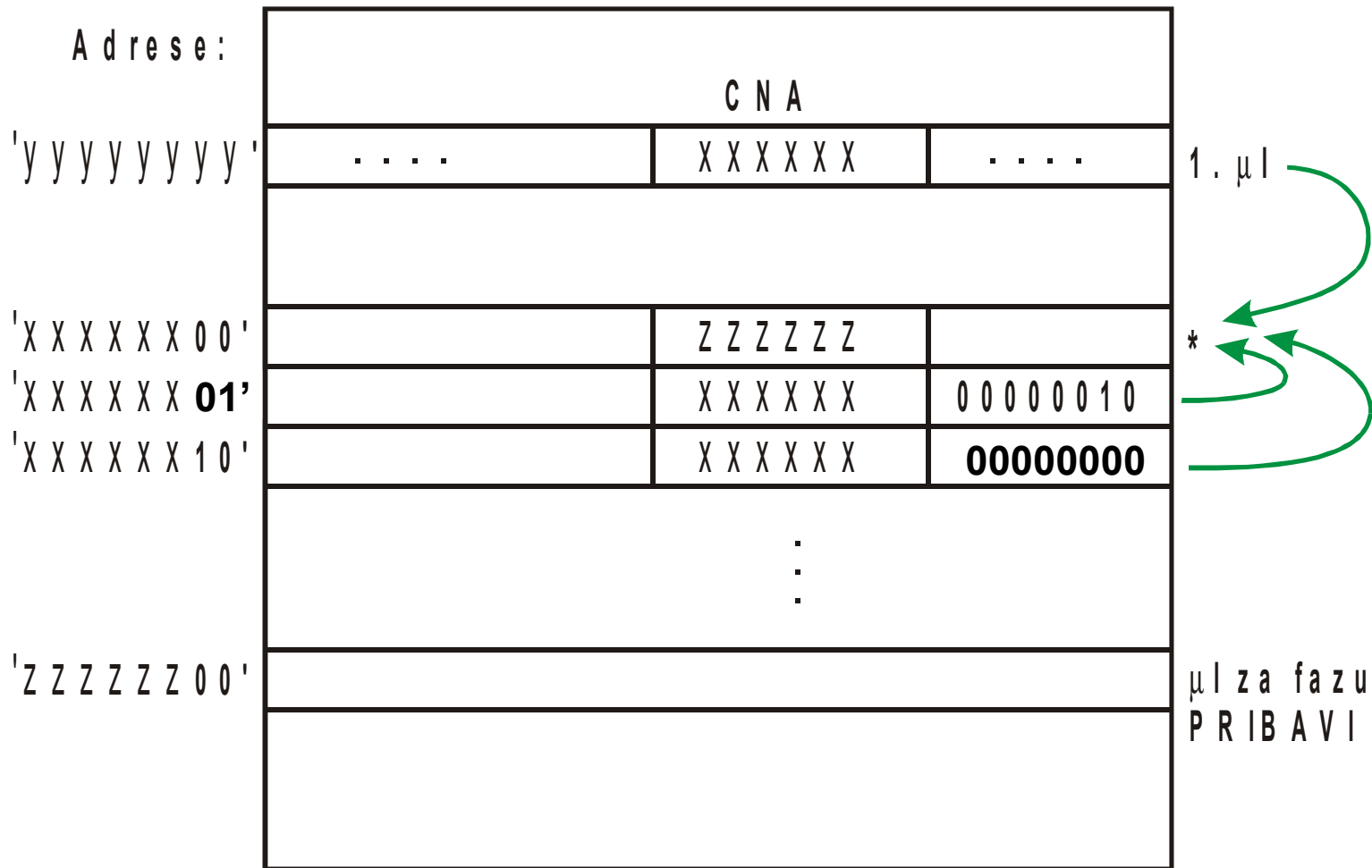
3'' - aktiviraj paralelno zbrajalo ($C_{in} = 0$)

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)/ $STAT(0) \leftarrow ZT, STAT(1) \leftarrow MB(0)$

/ CAB(2) * G * P(1)/ H(6) ← STAT(0) ; STAT(0) = Z

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

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

/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) * G * P(1)/

H(6) \leftarrow 0

/CBB(0) * G * P(1)/

H(7) \leftarrow 0

/G * P(1)/

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

/G * P(2)/

F ← CM(H)

; A < B

/CA(1) * P(0)/

L(0-7, 8-15) = 0 – F(EM) ; F(EM) = 2

/CB(3) * P(0)/

R = PC

/COP(0) * P(0)/

C_{in} = 0

; zbroji s C_{in} = 0

/CMB(3)* P(0)/

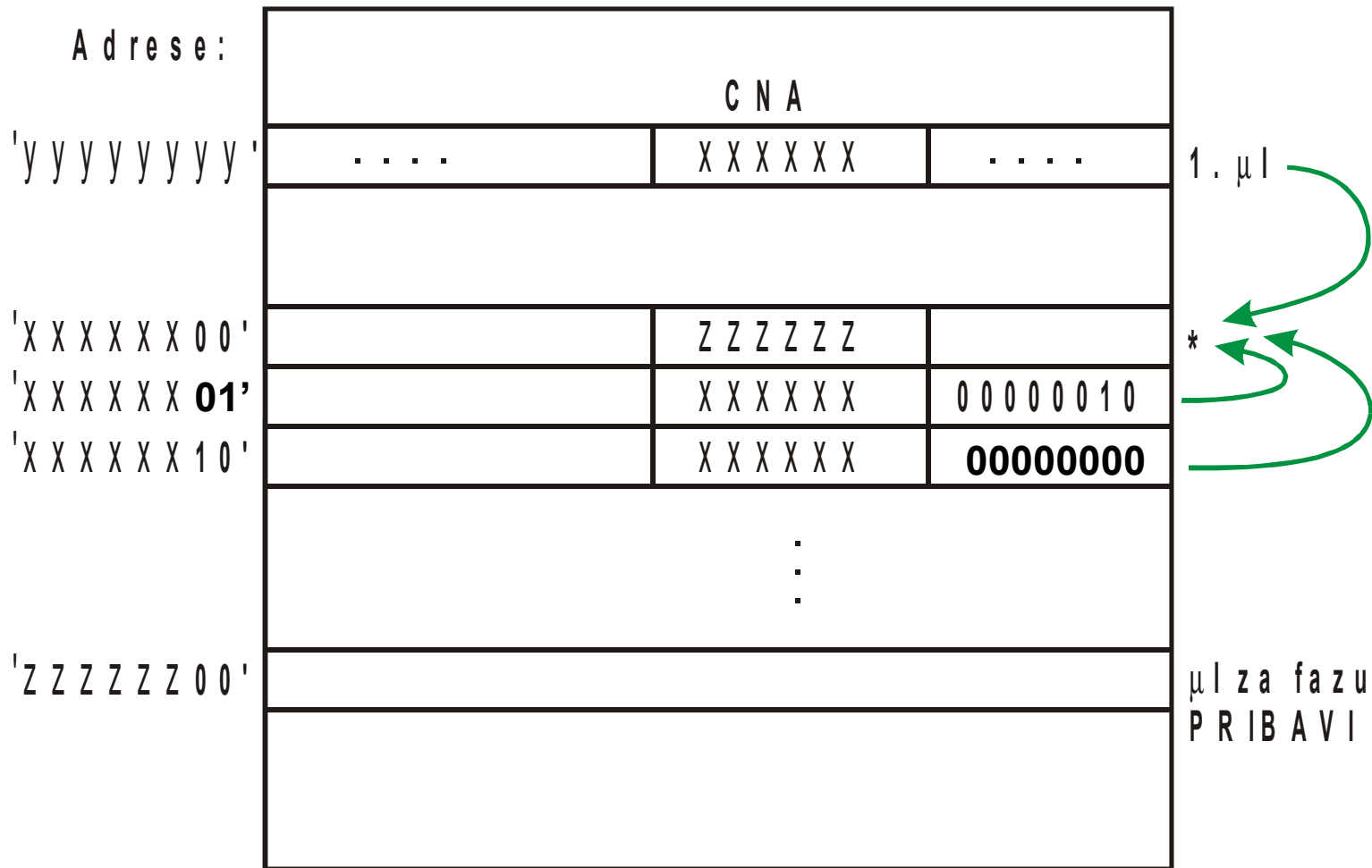
PC ← MB

/CAB(0) * G * P(1)/ $H(6) \leftarrow 0$

/CBB(0) * G * P(1)/ $H(7) \leftarrow 0$

/G * P(1)/ $H(0-5) \leftarrow F(CNA)$; $F(CNA) = \text{'xxxxxx'}$

/G * P(2)/ $F \leftarrow CM(H)$



MIKROPROGRAMSKA
MEMORIJA
(256 riječi x 32 bita)

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

redosljed μ instrukcija:

yyyyyyyy

xxxxxx00

xxxxxx01

xxxxxx10

Format upravljačke riječi (mikroriječi)

Zahtjevi pri oblikovanju upravljačke riječi:

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

Tehnički oblikovni kriteriji (međusobno ovisni!):

- a) minimalna veličina upravljačke memorije (broj riječi \times širina riječi)
- b) minimalno vrijeme izvođenja makroinstrukcija

Tehnike oblikovanja upravljačkih riječi (načini dodjeljivanja upravljačkih bitova):

1. horizontalno mikroprogramiranje
(nezavisno upravljanje resursima)
2. vertikalno mikroprogramiranje
(slijedno izvođenje ili grupiranje mikrooperacija)
3. mikroinstrukcije s promjenljivim formatom

“Horizontalna” mikroinstrukcija ima jednu ili više od sljedećih **značajki**:

1. mikroinstrukcija omogućava nezavisno upravljanje sklopovskim resursima (npr., funkcionalne jedinice, putevi podataka)
2. velika duljina upravljačke riječi: do nekoliko stotina bitova
3. horizontalne mikroinstrukcije dopuštaju upravljanje na razini pojedine mikrooperacije

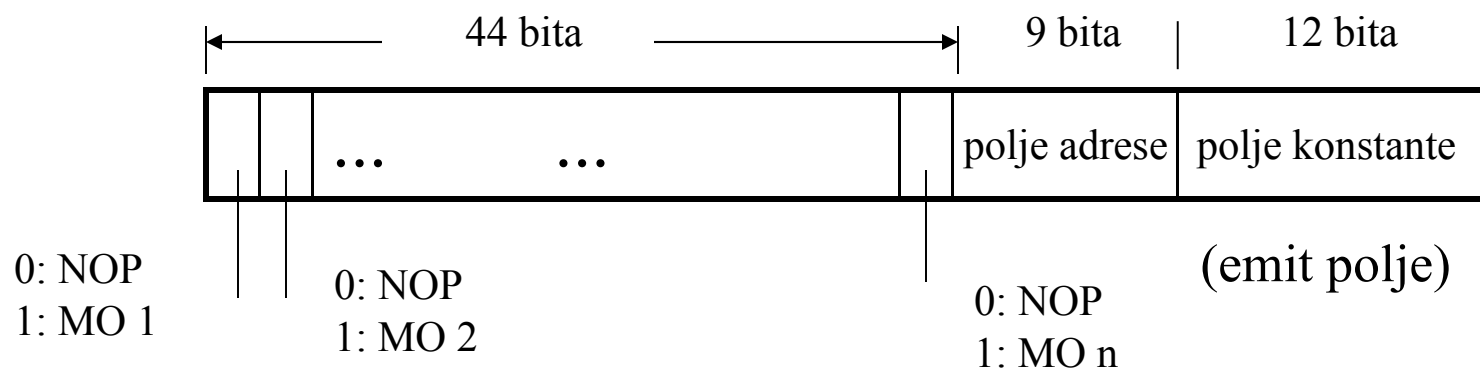
Primjer horizontalne mikroinstrukcijske arhitekture:

- upravljačka jedinica predstavljena u prethodnom predavanju (uobičajen kompromis: grupiranje isključivih μ o, npr kod polja CA)

Svojstva:

- potpuno iskorištenje sklopovskih resursa, dobra performansa makroprograma
- inspiracija za instrukcijsku arhitekturu VLIW

Izravno dodjeljivanje bitova mikroriječi upravljačkim signalima:



- najčešće se koristi kod horizontalnog programiranja

Značajke vertikalne mikroprogramske organizacije

1. mikroinstrukcije pobuđuju ili jednu mikrooperaciju ili preprogramirani skup mikrooperacija
2. ne dopušta se potpuno iskorištavanje paralelizma u procesoru
3. mikroinstrukcije su relativno male duljine – npr. 16 bitova

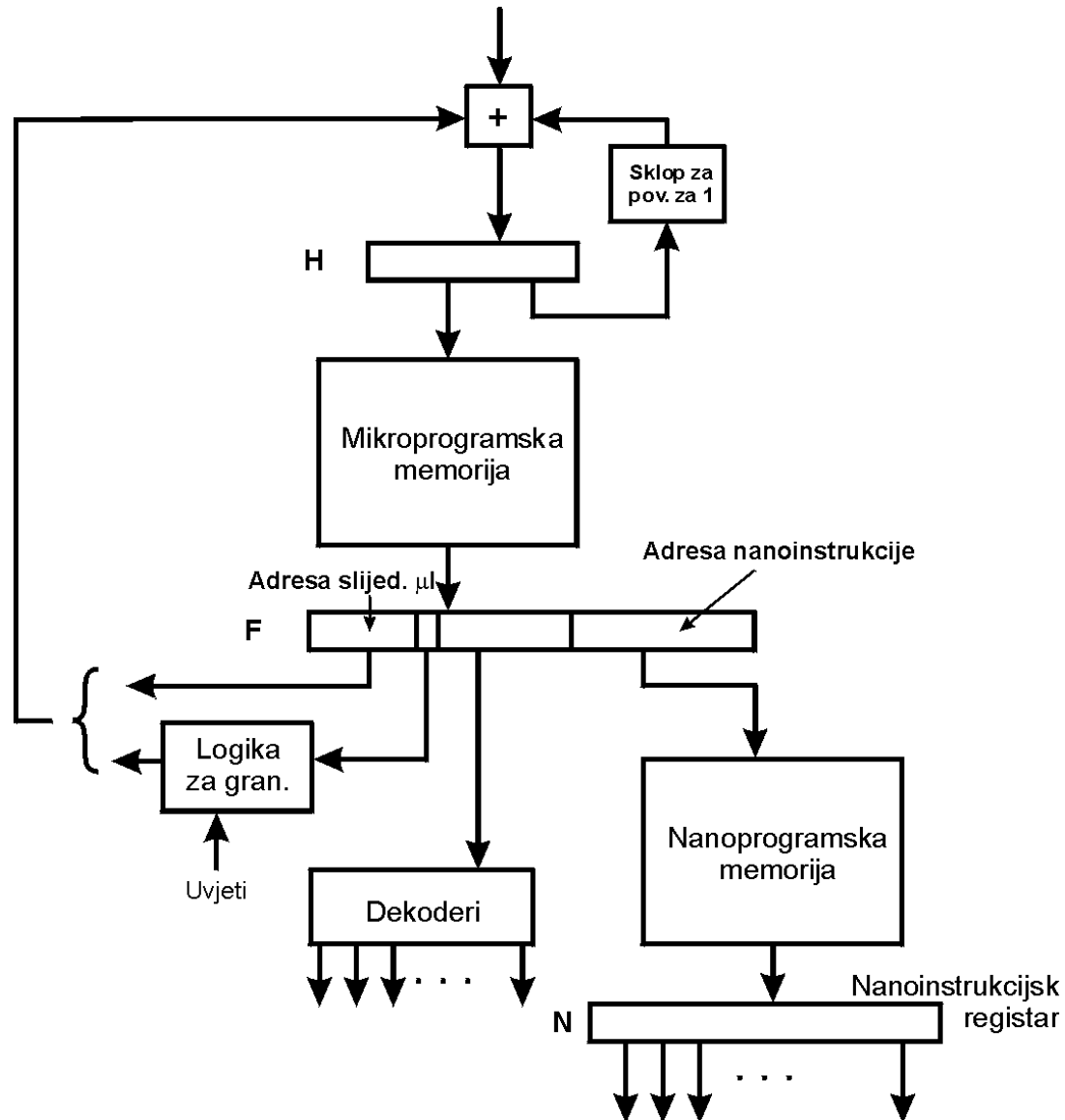
Primjeri vertikalnih mikroinstrukcijskih formata:

- pojedinačne mikrooperacije; npr: ADD MDR, A
- dvorazinska organizacija: kodirajući dio mikroinstrukcije adresira nanoprogramku memoriju u kojoj su horizontalne mikroinstrukcije (nanoriječi se referenciraju iz više mikroriječi, MC68000)
- mikroprogramirani sustav Intel 3000 (laboratorijske vježbe)

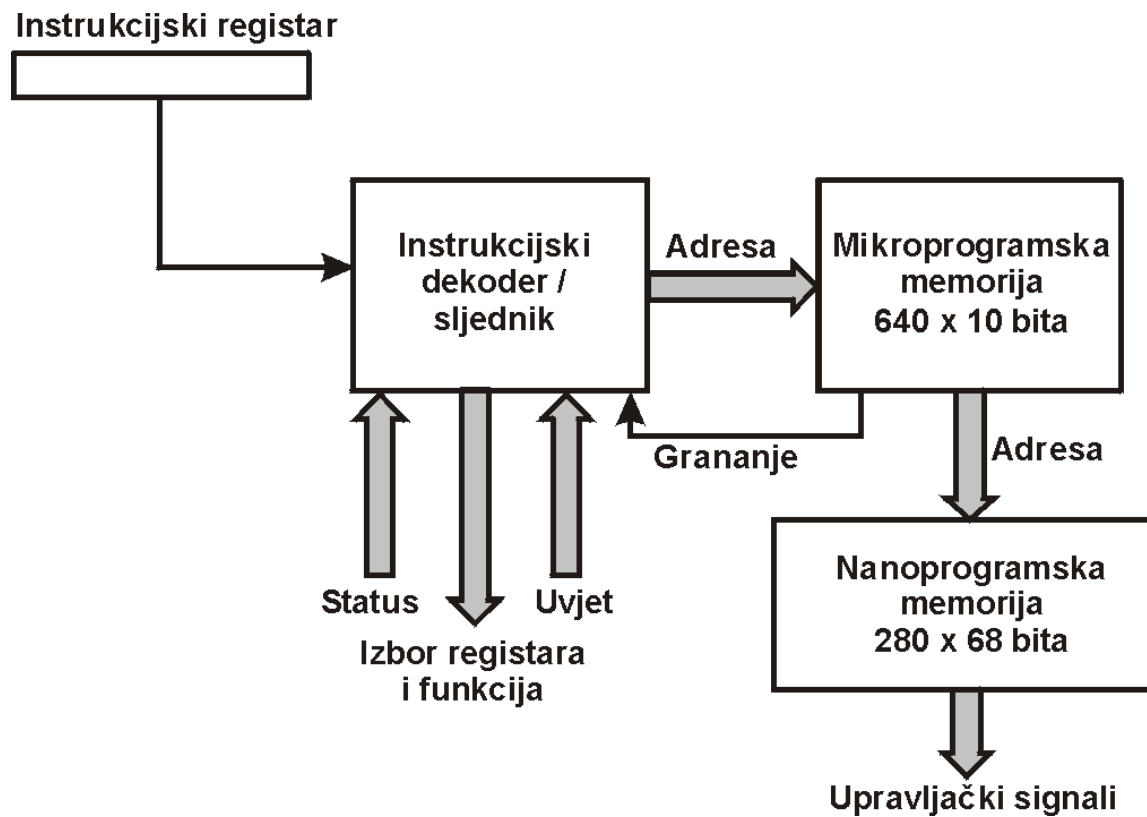
Svojstva:

- manja mikroprogramska memorija
- sporiji odziv zbog potrebe za dodatnim dekodiranjem
- grupiranje μ operacija i višestruki formati μ instrukcija: inspiracija za instrukcijsku arhitekturu tipa RISC

Organizacija mikroprogramirane jedinice u dvije razine (nanoprogramska jed.)



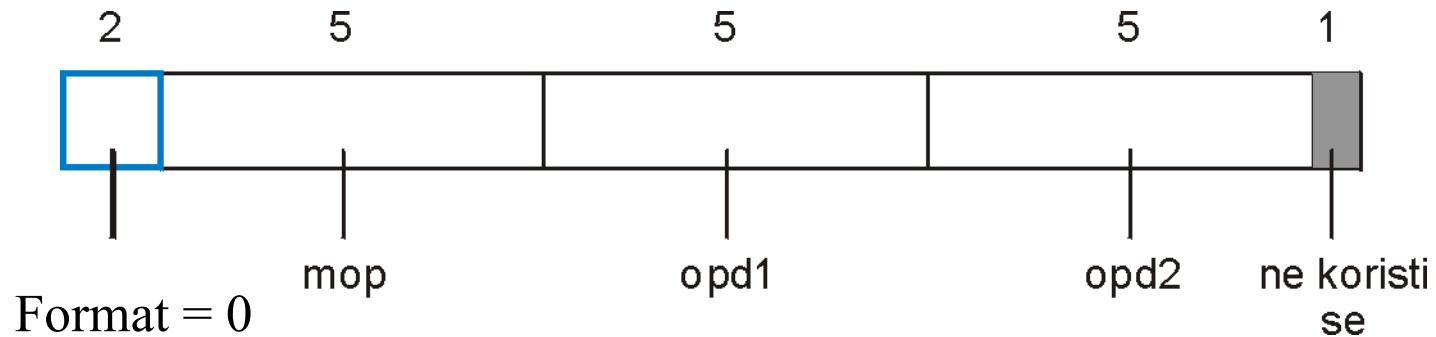
Organizacija upravljačke jedinice procesora MC 68000

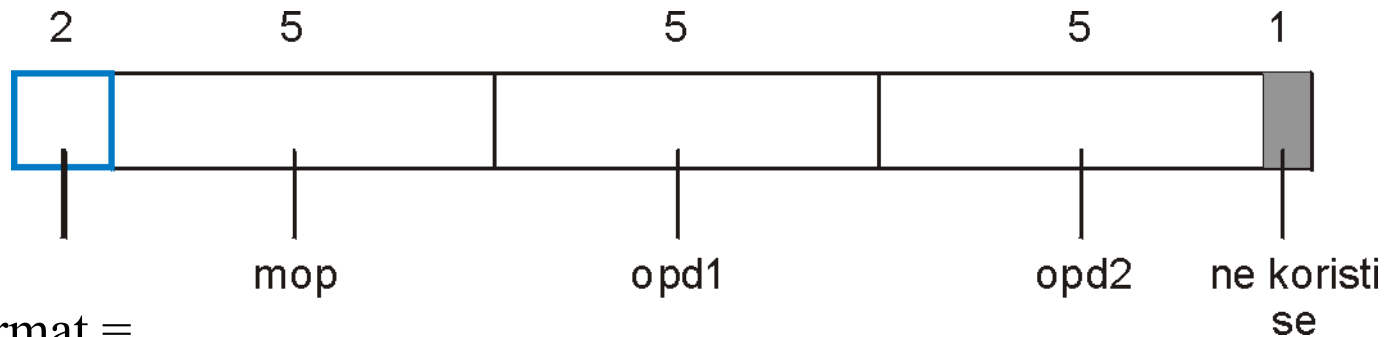


Mikroinstrukcije s promjenljivim formatom

- mikroriječ je kratka: 16 – 32 bita
- mikroinstrukcije specificiraju nekoliko mikrooperacija (1-4)

Primjer:





Format =
0

if format = 0
then

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

if format = 1 **then**

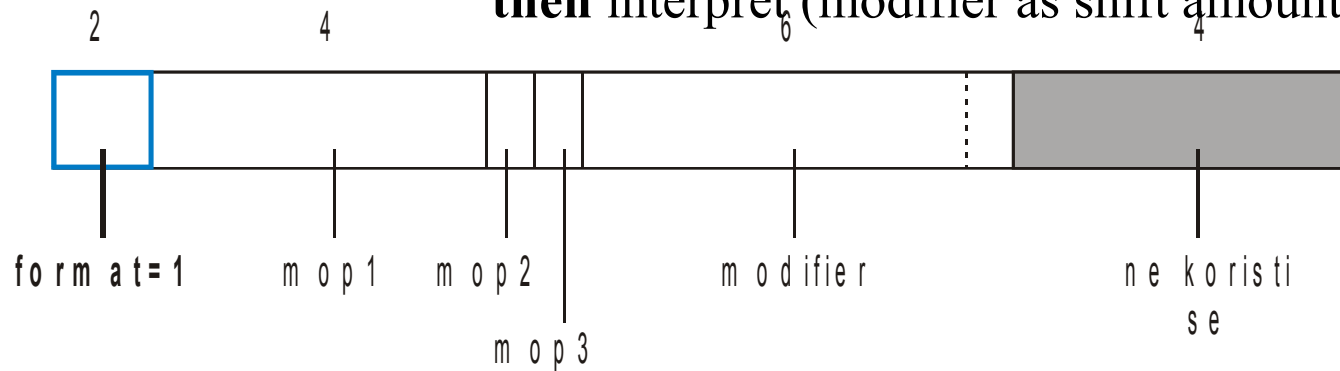
mop1 \in {ADD, SUB, OR, AND, EX_OR, NOT, SHR}

mop2 \in {READ_MM, WRITE_MM}

mop3 \in {INC_PC}

if mop1 \in {SHL, SHR }

then interpret (modifier as shift amount)



Specifikacija do 3 μ o koje se mogu istodobno izvesti:

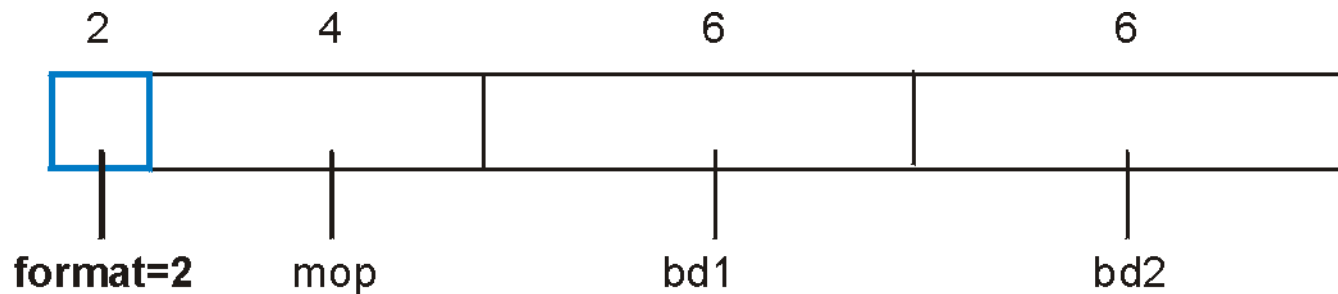
1. AOUT \leftarrow AIL mop1 AIR
 AIL, AIR: arithmetic input left, right
 AOUT: arithmetic output
2. PC \leftarrow PC+1
3. MDR \leftarrow M(MAR) **if** mop2 = READ_MM
 MDR \leftarrow M(MAR) **if** mop2 = WRITE_MM

if format = 2

then

mop \in {MASK}

and interpret (bd1 as bound i, bd2 as bound j)



AOUT := AIL mask AIR [i..j]

- AIL, AIR: arithmetic input left, right
- AOUT: arithmetic output
- $AOUT \leftarrow AIL[0..i] \parallel (AIL[i..j] \& AIR[i..j]) \parallel AIL[j..n]$

if format = 3

then

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

and interpret (addr asbranch address)

