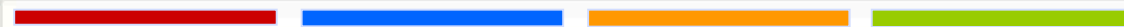
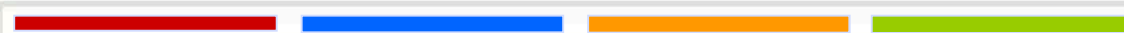


Organizacija procesora



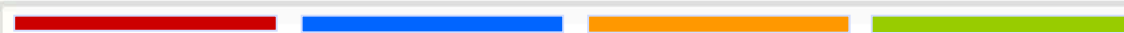
Upravljačka jedinica

- Upravljačka jedinica se u opštem slučaju realizuje kao sekvencijalna mreža sa onoliko stanja koliko ima koraka u sekvenci upravljačkih signala po koracima
- Svakom koraku se dodeljuje posebno stanje
- Stanja dodeljena operacionim koracima se koriste za generisanje upravljačkih signala operacione jedinice, a stanja dodeljena upravljačkim koracima se koriste za realizaciju skokova



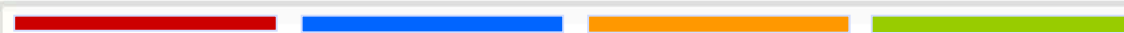
Upravljačka jedinica (2)

- U zavisnosti od toga kako se stanja sekvencijalne mreže koriste za generisanje upravljačkih signala operacione jedinice i realizaciju skokova u sekvenci upravljačkih signala po koracima, razlikuju se dve osnovne tehnike realizacija upravljačke jedinice i to
 - ožičena realizacija upravljačke jedinice i
 - mikroprogramska realizacija upravljačke jedinice
- Biće prikazane na primeru operacione jedinice sa direktnim vezama



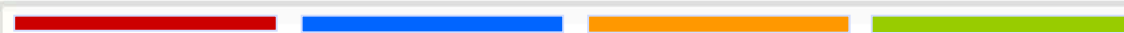
Ožičena realizacija

- Upravljačka jedinica se sastoji iz:
 - brojača koraka
 - dekodera stanja
 - kombinacione mreže za generisanje upravljačkih signala i
 - kombinacione mreže za generisanje nove vrednosti brojača koraka



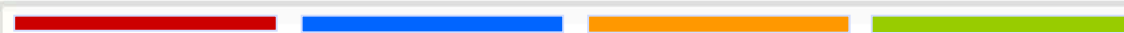
Ožičena realizacija (2)

- Posebno stanje brojača koraka se dodeljuje svakom od koraka u sekvenci upravljačkih signala po koracima
- Na osnovu vrednosti brojača koraka na izlazima dekodera koraka se dobija aktivna vrednost jednog signala koraka
- Kombinatorna mreža za generisanje upravljačkih signala na osnovu signala koraka generiše dve grupe signala i to:
 - upravljačke signale operacione jedinice i
 - upravljačke signale upravljačke jedinice



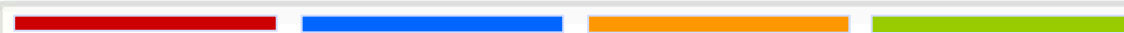
Ožičena realizacija (3)

- Upravljački signali operacione jedinice obezbeđuju izvršavanje odgovarajućih mikrooperacija u operacionoj jedinici
- Upravljački signali upravljačke jedinice obezbeđuju da se sadržaj brojača koraka ili inkrementira ili da se preko kombinacione mreže za generisanje nove vrednosti brojača koraka generiše nova vrednost i upiše u brojač koraka i time realizuje skok u sekvenci upravljačkih signala po koracima



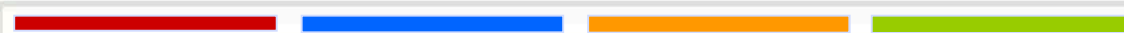
Ožičena realizacija (4)

- Upravljački signali se generišu kao unija signala dekodovanih stanja brojača koraka dodeljenih koracima u kojima se odgovarajući upravljački signali operacione jedinice pojavljuju i koracima u kojima upravljački signali upravljačke jedinice treba da realizuju bezuslovne, uslovne i višestruke uslovne skokove



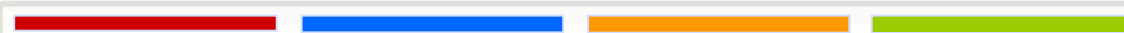
Upravljačka jedinica bez spajanja koraka

- Upravljački signali operacione jedinice se mogu generisati na osnovu sekvence upravljačkih signala po koracima
- Za svaki upravljački signal operacione jedinice treba krenuti kroz sekvencu upravljačkih signala po koracima i tražiti korake sa iskazima za signale u kojima se pojavljuje dati signal
- Za svaki takav korak treba uzeti signal dekodovanog stanja brojača koraka i formirati njihovu uniju



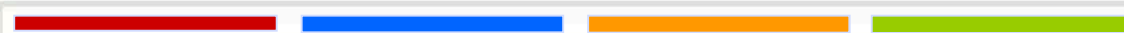
Upravljačka jedinica bez spajanja koraka (2)

- Upravljački signali upravljačke jedinice se ne mogu generisati na osnovu sekvence upravljačkih signala po koracima, jer se u njoj ne pojavljuju upravljački signali upravljačke jedinice, već samo iskazi za skokove
- Zbog toga je potrebno na osnovu sekvence upravljačkih signala po koracima formirati sekvencu upravljačkih signala za upravljačku jedinicu ožičene realizacije



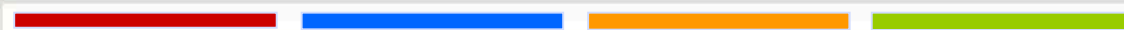
Upravljačka jedinica bez spajanja koraka (3)

- U njoj treba da se pored upravljačkih signala operacione jedinice pojave i upravljački signali upravljačke jedinice neophodni za realizaciju bezuslovnih, uslovnih i višestrukih uslovnih skokova specificiranih iskazima za skokove
- Prilikom njenog formiranja primenjuje se različit postupak za upravljačke signale operacione jedinice i za upravljačke signale upravljačke jedinice



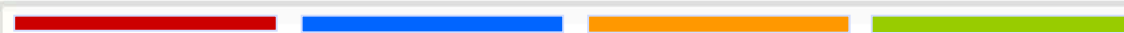
Upravljačka jedinica bez spajanja koraka (4)

- Za upravljačke signale operacione jedinice treba staviti iskaze za signale onako kako se javljaju u sekvenci upravljačkih signala po koracima
- Za upravljačke signale upravljačke jedinice treba u sekvenci upravljačkih signala po koracima tražiti iskaze: *br* step_A , *br* (*if* **uslov** *then* step_A) i *br* (*case* (**uslov**₁, ..., **uslov**_n) *then* (**uslov**₁, step_{A1}), ..., (**uslov**_n, step_{An}))



Upravljačka jedinica bez spajanja koraka (5)

- Umesto iskaza $br\ step_A$ treba staviti signal bezuslovnog skoka koji određuje da se bezuslovno prelazi na korak $step_A$ i signal val_A koji određuje da treba formirati binarnu vrednost A za upis u brojač koraka
- Simbolička oznaka signala bezuslovnog skoka je **bruncnd**



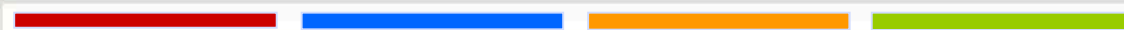
Upravljačka jedinica bez spajanja koraka (6)

- Koraci step_A , simboličke oznake signala val_A i vrednosti A za sve korake ovog tipa koji se javljaju u sekvenci upravljačkih signala po koracima, dati su u tabeli

step_A	val_A	A
step_{00}	val_{00}	00
step_{2C}	val_{2C}	2C
step_{31}	val_{31}	31
step_{56}	val_{56}	56

Upravljačka jedinica bez spajanja koraka (7)

- Umesto iskaza *br* (*if uslov then step_A*) treba staviti signal uslovnog skoka koji određuje signal **uslov** koji treba da bude aktivan da bi se realizovao prelaz na korak $step_A$ i signal **val_A** koji određuje da treba formirati binarnu vrednost A za upis u brojač koraka u slučaju da je signal **uslov** aktivan



Upravljačka jedinica bez spajanja koraka (8)

- Simboličke oznake signala uslovnih skokova i signala uslova za sve iskaze ovog tipa koji se javljaju u sekvenci upravljačkih signala po koracima, dati su u tabeli

signal uslovnog skoka	signal uslova
brl1	l1
brl2	l2
brSTORE	STORE
brimmed	immed
brregdir	regdir
breql	eql
brnotPREKID	<u>PREKID</u>

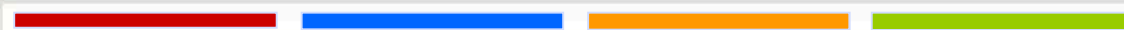
Upravljačka jedinica bez spajanja koraka (9)

- Koraci step_A , simboličke oznake signala val_A i vrednosti A za sve korake ovog tipa koji se javljaju u sekvenci upravljačkih signala po koracima dati su u tabeli

step_A	val_A	A
step_{00}	val_{00}	00
step_{0F}	val_{0F}	0F
step_{31}	val_{31}	31
step_{39}	val_{39}	39
step_{4C}	val_{4C}	4C
step_{56}	val_{56}	56

Upravljačka jedinica bez spajanja koraka (10)

- Umesto iskaza *br* (*case* (**uslov**₁, ..., **uslov**_n) *then* (**uslov**₁, step_{A1}), ..., (**uslov**_n, step_{An}) treba staviti signal višestrukog uslovnog skoka koji određuje signale **uslov**₁, **uslov**₂, ..., **uslov**_n od kojih jedan treba da bude aktivan da bi se realizovao prelazak na jedan od koraka step_{A1}, step_{A2}, ..., step_{An}



Upravljačka jedinica bez spajanja koraka (11)

- Simboličke oznake signala višestrukog uslovnog skoka za sve iskaze ovog tipa koji se javljaju u sekvenci upravljačkih signala po koracima, date su u tabeli

korak	signal višestrukog uslovnog skoka
step _{0F}	bradr
step ₃₁	bropr

Upravljačka jedinica bez spajanja koraka (12)

- Vrednosti koje treba upisati u brojač koraka i signali uslova koji određuju koju od tih vrednosti treba upisati u brojač koraka za iskaz ovog tipa koji se javlja u koraku $step_{0F}$, dati su u tabeli

signal uslova	vrednost
dirreg	10
indreg	13
postdec	16
preinc	1C
dirnem	22
indnem	24
indregpom	28
innmed	30

Upravljačka jedinica bez spajanja koraka (13)

- Vrednosti koje treba upisati u brojač koraka i signali uslova koji određuju koju od tih vrednosti treba upisati u brojač koraka za iskaz ovog tipa koji se javlja u koraku $step_{31}$, dati su u tabeli

signal uslova	vrednost	signal uslova	vrednost
LOAD	32	JZ	48
STORE	34	JMP	4A
ADD	3C	JSR	4C
AND	40	RTI	4E
ASR	44	RTS	52

Sekvenca upravljačkih signala – čitanje instrukcije

Po koracima	Za uprav. jed.
step ₀₀ ldMAR, incPC;	T ₀₀ ldMAR, incPC;
step ₀₁ ldMBR;	T ₀₁ ldMBR;
step ₀₂ ldIR1;	T ₀₂ ldIR1;
step ₀₃ br (if I1 then step ₃₁);	T ₀₃ brI1, val ₃₁ ;
step ₀₄ ldMAR, incPC;	T ₀₄ ldMAR, incPC;
step ₀₅ ldMBR;	T ₀₅ ldMBR;
step ₀₆ ldIR2;	T ₀₆ ldIR2;
step ₀₇ br (if I2 then step _{0F});	T ₀₇ brI2, val _{0F} ;
step ₀₈ ldMAR, incPC;	T ₀₈ ldMAR, incPC;
...	...
step _{0E} ldIRN;	T _{0E} ldIRN;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda

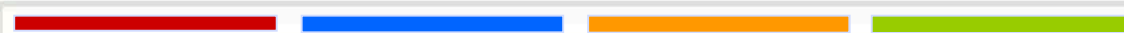
Po koracima	Za uprav. jed.
$step_{0F}$ <i>br</i> (<i>case</i> (dirreg , indreg , postdec , preinc , dirmem , indmem , indregpom , immed) <i>then</i> (dirreg , $step_{10}$), (indreg , $step_{13}$), (postdec , $step_{16}$), (preinc , $step_{1C}$), (dirmem , $step_{22}$), (indmem , $step_{24}$), (indregpom , $step_{28}$), (immed , $step_{30}$));	T_{0F} bradr ;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – direktno registarsko

Po koracima	Za uprav. jed.
step ₁₀ ldRSRC;	T ₁₀ ldRSRC;
step ₁₁ ldB;	T ₁₁ ldB;
step ₁₂ <i>br</i> step ₃₁ ;	T ₁₂ bruncnd, val₃₁;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – indirektno registarsko

Po koracima	Za uprav. jed.
step ₁₃ IdRSRC;	T ₁₃ IdRSRC;
step ₁₄ mxMAR₀, IdMAR;	T ₁₄ mxMAR₀, IdMAR;
step ₁₅ <i>br</i> step _{2C} ;	T ₁₅ bruncnd, val_{2C};



Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – postdekrement

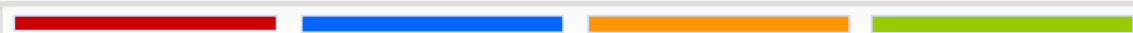
Po koracima	Za uprav. jed.
step ₁₆ ldRSRC;	T ₁₆ ldRSRC;
step ₁₇ mxMAR₀, ldMAR, ldB;	T ₁₇ mxMAR₀, ldMAR, ldB;
step ₁₈ decB;	T ₁₈ decB;
step ₁₉ mxRDST, ldRDST;	T ₁₉ mxRDST, ldRDST;
step _{1A} wrGPR;	T _{1A} wrGPR;
step _{1B} <i>br</i> step _{2C} ;	T _{1B} bruncnd, val_{2C};

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – preinkrement

Po koracima	Za uprav. jed.
step _{1C} ldRSRC;	T _{1C} ldRSRC;
step _{1D} ldB;	T _{1D} ldB;
step _{1E} incB;	T _{1E} incB;
step _{1F} mxMAR₂, mxMAR₁, ldMAR, mxRDST, ldRDST;	T _{1F} mxMAR₂, mxMAR₁, ldMAR, mxRDST, ldRDST;
step ₂₀ wrGPR;	T ₂₀ wrGPR;
step ₂₁ <i>br</i> step _{2C} ;	T ₂₁ bruncnd, val_{2C};

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – direktno memorijsko

Po koracima	Za uprav. jed.
step ₂₂ mxMAR₁, IdMAR; step ₂₃ <i>br</i> step _{2C} ;	T ₂₂ mxMAR₁, IdMAR; T ₂₃ bruncnd, val_{2C};



Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – indirektno memorijsko

Po koracima	Za uprav. jed.
step ₂₄ mxMAR₁, IdMAR;	T ₂₄ mxMAR₁, IdMAR;
step ₂₅ IdMBR;	T ₂₅ IdMBR;
step ₂₆ mxMAR₁, mxMAR₀, IdMAR;	T ₂₆ mxMAR₁, mxMAR₀, IdMAR;
step ₂₇ <i>br</i> step _{2C} ;	T ₂₇ bruncnd, val_{2C};

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – indirektno registarsko sa pomerajem

Po koracima	Za uprav. jed.
step ₂₈ IdRSRC;	T ₂₈ IdRSRC;
step ₂₉ mxX₀, IdX, mxY₀, IdY;	T ₂₉ mxX₀, IdX, mxY₀, IdY;
step _{2A} add, IdZ;	T _{2A} add, IdZ;
step _{2B} mxMAR₂, IdMAR;	T _{2B} mxMAR₂, IdMAR;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – čitanje operanda za mem. adr.

Po koracima	Za uprav. jed.
step _{2C} <i>br</i> (if STORE then step ₃₁);	T _{2C} brSTORE , val ₃₁ ;
step _{2D} ldMBR ;	T _{2D} ldMBR ;
step _{2E} mxB₀ , ldB ;	T _{2E} mxB₀ , ldB ;
step _{2F} <i>br</i> step ₃₁ ;	T _{2F} bruncnd , val ₃₁ ;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – neposredno

Po koracima	Za uprav. jed.
step ₃₀ mxB₁, IdB;	T ₃₀ mxB₁, IdB;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – izvršavanje operacije

Po koracima	Za uprav. jed.
<p>step₃₁ <i>br</i> (<i>case</i> (LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTI, RTS) <i>then</i></p> <p>(LOAD, step₃₂), (STORE, step₃₄),</p> <p>(ADD, step_{3C}), (AND, step₄₀), (ASR, step₄₄),</p> <p>(JZ, step₄₈), (JMP, step_{4C}),</p> <p>(JSR, step_{4A}), (RTI, step_{4E}), (RTS, step₅₂));</p>	<p>T₃₁ bropr;</p>

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – LOAD

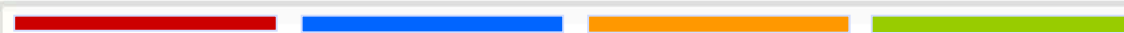
Po koracima	Za uprav. jed.
step ₃₂ mxACC, IdACC; step ₃₃ <i>br</i> step ₅₆ ;	T ₃₂ mxACC, IdACC; T ₃₃ bruncnd, val₅₆;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – STORE

Po koracima	Za uprav. jed.
step ₃₄ <i>br (if immed then step₅₆);</i>	T ₃₄ brimmed, val₅₆;
step ₃₅ <i>br (if regdir then step₃₉);</i>	T ₃₅ brregdir, val₃₉;
step ₃₆ mxMBR₀, IdMBR;	T ₃₆ mxMBR₀, IdMBR;
step ₃₇ wrMEM;	T ₃₇ wrMEM;
step ₃₈ <i>br step₅₆;</i>	T ₃₈ bruncnd, val₅₆;
step ₃₉ IdRDST;	T ₃₉ IdRDST;
step _{3A} wrGPR;	T _{3A} wrGPR;
step _{3B} <i>br step₅₆;</i>	T _{3B} bruncnd, val₅₆;

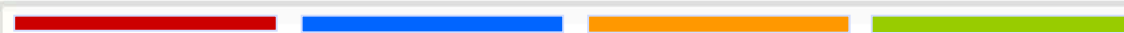
Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – ADD

Po koracima	Za uprav. jed.
step _{3C} ldX, ldY;	T _{3C} ldX, ldY;
step _{3D} add, ldZ;	T _{3D} add, ldZ;
step _{3E} ldACC;	T _{3E} ldACC;
step _{3F} <i>br</i> step ₅₆ ;	T _{3F} bruncnd, val₅₆;



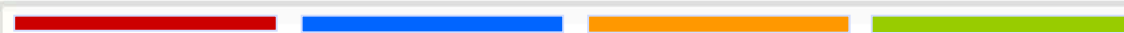
Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – AND

Po koracima	Za uprav. jed.
step ₄₀ ldX, ldY;	T ₄₀ ldX, ldY;
step ₄₁ and, ldZ;	T ₄₁ and, ldZ;
step ₄₂ ldACC;	T ₄₂ ldACC;
step ₄₃ <i>br</i> step ₅₆ ;	T ₄₃ bruncnd, val₅₆;



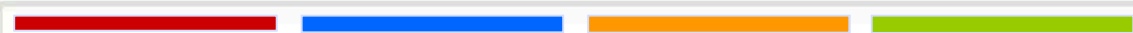
Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – ASR

Po koracima	Za uprav. jed.
step ₄₄ ldY;	T ₄₄ ldY;
step ₄₅ asr, ldZ;	T ₄₅ asr, ldZ;
step ₄₆ ldACC;	T ₄₆ ldACC;
step ₄₇ <i>br</i> step ₅₆ ;	T ₄₇ bruncnd, val₅₆;



Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – JZ

Po koracima	Za uprav. jed.
step ₄₈ <i>br</i> (if eq l then step _{4C});	T ₄₈ breql, val_{4C} ;
step ₄₉ <i>br</i> step ₅₆ ;	T ₄₉ bruncnd, val₅₆ ;

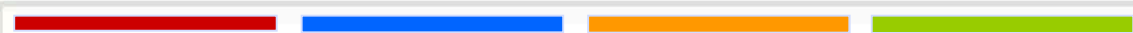


Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – JSR

Po koracima	Za uprav. jed.
step _{4A} mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR; step _{4B} wrMEM;	T _{4A} mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR; T _{4B} wrMEM;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – JMP

Po koracima	Za uprav. jed.
step _{4C} IdPC; step _{4D} <i>br</i> step ₅₆ ;	T _{4C} IdPC; T _{4D} bruncnd, val₅₆;

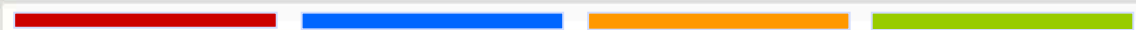


Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – RTI

Po koracima	Za uprav. jed.
step _{4E} incSP;	T _{4E} incSP;
step _{4F} mxMAR₂, mxMAR₀, ldMAR;	T _{4F} mxMAR₂, mxMAR₀, ldMAR;
step ₅₀ ldMBR;	T ₅₀ ldMBR;
step ₅₁ ldPSW;	T ₅₁ ldPSW;

Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – RTS

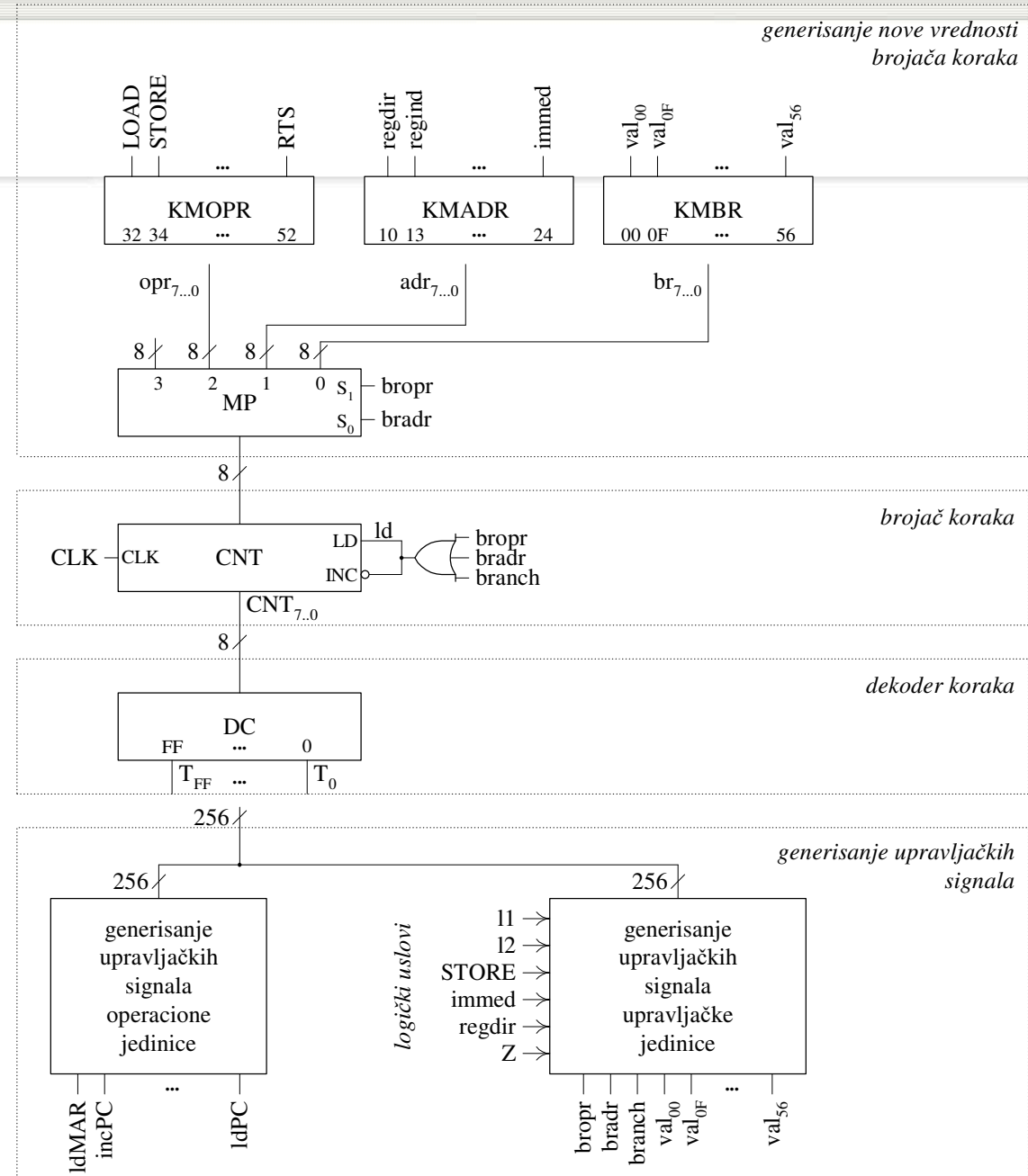
Po koracima	Za uprav. jed.
step ₅₂ incSP;	T ₅₂ incSP;
step ₅₃ mxMAR₂, mxMAR₀, ldMAR;	T ₅₃ mxMAR₂, mxMAR₀, ldMAR;
step ₅₄ ldMBR;	T ₅₄ ldMBR;
step ₅₅ mxPC, ldPC;	T ₅₅ mxPC, ldPC;



Sekvenca upravljačkih signala – formiranje adrese i čitanje operanda – opsluživanje prekida

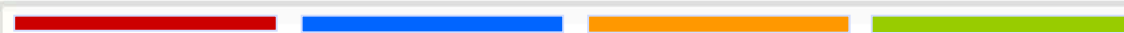
Po koracima	Za uprav. jed.
step ₅₆ <i>br</i> (if then step ₀₀);	T ₅₆ brnotPREKID, val₀₀ ;
step ₅₇ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR ;	T ₅₇ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR ;
step ₅₈ wrMEM ;	T ₅₈ wrMEM ;
step ₅₉ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, mxMBR₀, ldMBR ;	T ₅₉ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, mxMBR₀, ldMBR ;
step _{5A} wrMEM ;	T _{5A} wrMEM ;
step _{5B} mxX₁, ldX, mxY₁, ldY ;	T _{5B} mxX₁, ldX, mxY₁, ldY ;
step _{5C} add, ldZ ;	T _{5C} add, ldZ ;
step _{5D} mxMAR₂, ldMAR ;	T _{5D} mxMAR₂, ldMAR ;
step _{5E} ldMBR ;	T _{5E} ldMBR ;
step _{5F} mxPC, ldPC ;	T _{5F} mxPC, ldPC ;
step ₆₀ <i>br</i> step ₀₀ ;	T ₆₀ bruncnd, val₀₀ ;

Struktura upravljačke jedinice ožičene realizacije



Generisanje upravljačkih signala operacione jedinice

- Upravljački signali operacione jedinice se generišu na sledeći način:
 - $\text{ldMAR} = T_{00} + T_{04} + T_{08} + T_{14} + T_{17} + T_{1F} + T_{22} + T_{24} + T_{26} + T_{2B} + T_{4A} + T_{4F} + T_{53} + T_{57} + T_{59} + T_{5D}$
 - $\text{mxMAR}_0 = T_{14} + T_{17} + T_{1F} + T_{26} + T_{4A} + T_{4F} + T_{53} + T_{57} + T_{59}$
 - $\text{wrGPR} = T_{1A} + T_{20} + T_{3A}$
- Na identičan način se generišu i preostali upravljački signali operacione jedinice



Generisanje upravljačkih signala upravljačke jedinice

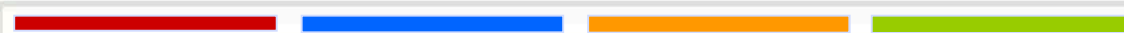
- Upravljački signali upravljačke jedinice se generišu na sledeći način:
 - $\text{bropr} = T_{31}$
 - $\text{bradr} = T_{0F}$
 - $\text{branch} = \text{bruncnd} + \text{brl1} * I1 + \text{brl2} * I2 + \text{brSTORE} * \text{STORE} + \text{brimmed} * \text{immed} + \text{brregdir} * \text{regdir} + \text{breql} * \text{eql} + \text{brnotPREKID} * \text{!PREKID}$
 - $\text{val00} = T_{56} + T_{60}$
 - $\text{val0F} = T_{07}$
 - $\text{val2C} = T_{15} + T_{1B} + T_{21} + T_{23} + T_{27}$
 - $\text{val31} = T_{03} + T_{12} + T_{2C} + T_{2F}$
 - $\text{val39} = T_{35}$
 - $\text{val4C} = T_{48}$
 - $\text{val56} = T_{33} + T_{34} + T_{38} + T_{3B} + T_{3F} + T_{43} + T_{47} + T_{49} + T_{4D}$

Generisanje upravljačkih signala upravljačke jedinice (2)

- Signali koji se javljaju u izrazu za signal **branch** se generišu na sledeći način:
 - $\text{bruncnd} = T_{12} + T_{15} + T_{1B} + T_{21} + T_{23} + T_{27} + T_{2F} + T_{33} + T_{38} + T_{3B} + T_{3F} + T_{43} + T_{47} + T_{49} + T_{4D} + T_{60}$
 - $\text{brl1} = T_{03}$
 - $\text{brl2} = T_{07}$
 - $\text{brSTORE} = T_{2C}$
 - $\text{brimmed} = T_{34}$
 - $\text{breql} = T_{48}$
 - $\text{brnotPREKID} = T_{56}$
- Pri generisanju signala **branch** koriste se sledeći signali logičkih uslova koji dolaze iz operacione jedinice:
 - I1, I2, STORE, immed, regdir, eql i PREKID

Upravljačka jedinica sa spajanjem koraka

- Upravljačka jedinica sa spajanjem koraka se realizuje istim postupkom kao i upravljačka jedinica bez spajanja koraka
- Samo se koristi sekvenca upravljačkih signala po koracima sa spajanjem koraka

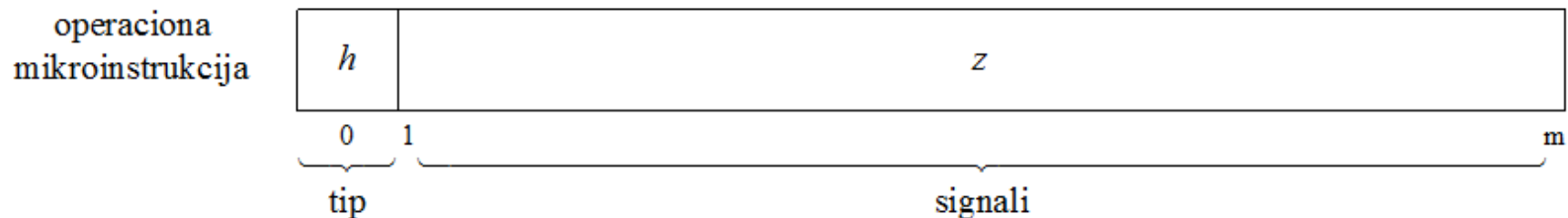


Mikroprogramska realizacija

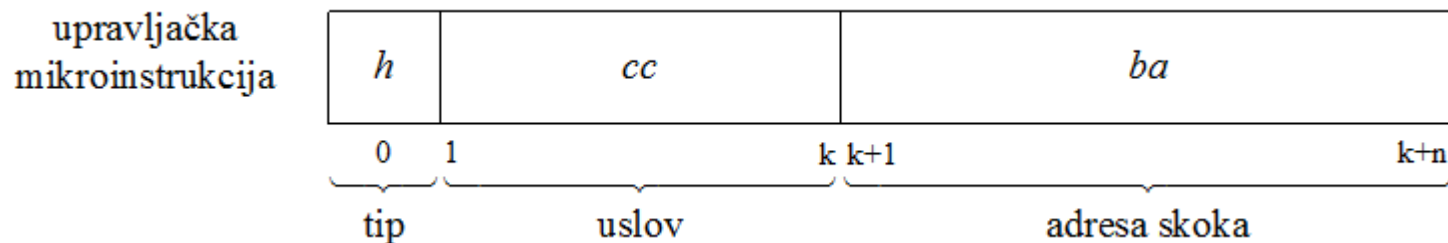
- Kod mikroprogramske realizacije upravljačke jedinice treba razmotriti:
 - sa koliko tipova mikroinstrukcija se realizuje:
 - sa jednim tipom mikroinstrukcija,
 - sa dva tipa mikroinstrukcija,
 - način kodiranja upravljačkih signala:
 - horizontalno kodiranje,
 - vertikalno kodiranje,
 - mešovito kodiranje.

Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- U sekvenci upravljačkih signala po koracima bez spajanja koraka se svakom operacionom koraku, u kome se generišu upravljački signali operacione jedinice, pridružuje binarna reč

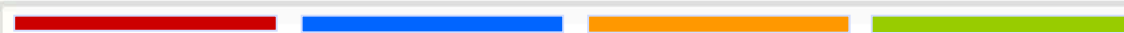


- i svakom upravljačkom koraku, u kome se realizuju skokovi, pridružuje binarna reč



Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Te binarne reči se nazivaju mikroinstrukcijama, mikronaredbama ili mikrokomandama.
- Mikroinstrukcije pridružene operacionim koracima nazivaju se operacione mikroinstrukcije, dok se mikroinstrukcije pridružene upravljačkim koracima nazivaju upravljačke mikroinstrukcije.
- Uređeni niz mikroinstrukcija pridruženih operacionim koracima i upravljačkim koracima, naziva se mikroprogram.



Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Upravljačka jedinica se sastoji iz:
 - mikroprogramske memorije,
 - mikroprogramskog brojača,
 - prihvatnog registra mikroinstrukcije,
 - kombinacione mreže za generisanje upravljačkih signala i kombinacione mreže za generisanje nove vrednosti mikroprogramskog brojača.

Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Operaciona mikroinstrukcija

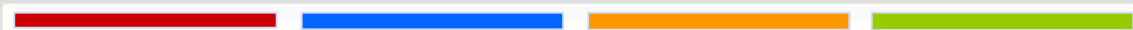
0	1	2	3	4	5	6	7
0	ldMAR	mxMAR ₂	mxMAR ₁	mxMAR ₀	wrMEM	ldMBR	mxMBR ₁

8	9	10	11	12	13	14	15
mxMBR ₀	ldPC	incPC	mxPC	incSP	decSP	ldIR ₁	ldIR ₂

16	17	18	19	20	21	22	23
ldACC	mxACC	incB	decB	ldB	mxB ₁	mxB ₀	ldPSW

24	25	26	27	28	29	30	31
mxPSW	ldX	mxX ₁	mxX ₀	ldY	mxY ₁	mxY ₀	add

32	33	34	35	36	37	38	39
and	asr	ldZ	ldRSRC	wrGPR	ldRDST	mxRDST	/



Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Upravljačka mikroinstrukcija

0	1	2	3	4	5	6	7
1	/	/	/	<i>cc</i>			

8	9	10	11	12	13	14	15
<i>ba</i>							

16	17	18	19	20	21	22	23
/	/	/	/	/	/	/	/

24	25	26	27	28	29	30	31
/	/	/	/				

32	33	34	35	36	37	38	39
/	/	/	/	/	/	/	/

Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Signal bezuslovnog skoka

<i>cc</i>	signal bezuslovnog skoka
01	brmcd

- Signali uslovnih skokova

<i>cc</i>	signal uslovnog skoka	signal uslova
02	brl1	l1
03	brl2	l2
04	brSTORE	STORE
05	brimmed	immed
06	breql	eql
07	brnotPREKID	$\overline{\text{PREKID}}$

- Signali višestrukih uslovnih skokova

<i>cc</i>	korak	signal višestrukog uslovnog skoka
08	step _{0F}	bradr
09	step ₃₁	bropr

Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Koraci $step_A$, adrese $madr_A$ i vrednosti A za bezuslovne skokove

$step_A$	$madr_A$	A
$step_{00}$	$madr_{00}$	00
$step_{2C}$	$madr_{2C}$	2C
$step_{31}$	$madr_{31}$	31
$step_{56}$	$madr_{56}$	56

- Koraci $step_A$, adrese $madr_A$ i vrednosti A za uslovne skokove

$step_A$	$madr_A$	A
$step_{00}$	$madr_{00}$	00
$step_{0F}$	$madr_{0F}$	0F
$step_{31}$	$madr_{31}$	31
$step_{39}$	$madr_{39}$	39
$step_{56}$	$madr_{56}$	56

Mikroprogramska realizacija sa dva tipa mikroinstrukcija

- Signali uslova i vrednosti za upis u mikroprogramski brojač za višestruki uslovni skok u koraku $step_{0F}$

signal uslova	vrednost
dirreg	10
indreg	13
postdec	16
preinc	1C
dirmem	22
indmem	24
indregpom	28
immed	30

- Signali uslova i vrednosti za upis u mikroprogramski brojač za višestruki uslovni skok u koraku $step_{31}$

signal uslova	vrednost	signal uslova	vrednost
LOAD	32	JZ	48
STORE	34	JMP	4A
ADD	3C	JSR	4C
AND	40	RTI	4E
ASR	44	RTS	52

Mikroprogram – čitanje instrukcije

Po koracima	Za uprav. jed.
step ₀₀ ldMAR, incPC;	madr00 ldMAR, incPC;
step ₀₁ ldMBR;	madr01 ldMBR;
step ₀₂ ldIR1;	madr02 ldIR1;
step ₀₃ br (if I1 then step ₃₁);	madr03 cnt, br11, madr31;
step ₀₄ ldMAR, incPC;	madr04 ldMAR, incPC;
step ₀₅ ldMBR;	madr05 ldMBR;
step ₀₆ ldIR2;	madr06 ldIR2;
step ₀₇ br (if I2 then step _{0F});	madr07 cnt, br12, madr0F;
step ₀₈ ldMAR, incPC;	madr08 ldMAR, incPC;
...	...
step _{0E} ldIRN;	madr0E ldIRN;

Mikroprogram – formiranje adrese i čitanje operanda

Po koracima	Za uprav. jed.
$step_{0F}$ <i>br</i> (<i>case</i> (dirreg , indreg , postdec , preinc , dirmem , indmem , indregpom , immed) <i>then</i> (dirreg , $step_{10}$), (indreg , $step_{13}$), (postdec , $step_{16}$), (preinc , $step_{1C}$), (dirmem , $step_{22}$), (indmem , $step_{24}$), (indregpom , $step_{28}$), (immed , $step_{30}$));	$madr_{0F}$ cnt, $bradr$;

Mikroprogram – formiranje adrese i čitanje operanda – direktno registarsko

Po koracima	Za uprav. jed.
step ₁₀ ldRSRC;	madr10 ldRSRC;
step ₁₁ ldB;	madr11 ldB;
step ₁₂ <i>br</i> step ₃₁ ;	madr12 cnt, bruncnd, madr31;

Mikroprogram – formiranje adrese i čitanje operanda – indirektno registarsko

Po koracima	Za uprav. jed.
step ₁₃ ldRSRC;	madr13 ldRSRC;
step ₁₄ mxMAR₀, ldMAR;	madr14 mxMAR0, ldMAR;
step ₁₅ <i>br</i> step _{2C} ;	madr15 cnt, bruncnd, madr2C;

Mikroprogram – formiranje adrese i čitanje operanda – postdekrement

Po koracima	Za uprav. jed.
step ₁₆ ldRSRC;	madr16 ldRSRC;
step ₁₇ mxMAR₀, ldMAR, ldB;	madr17 mxMAR0, ldMAR, ldB;
step ₁₈ decB;	madr18 decB;
step ₁₉ mxRDST, ldRDST;	madr19 mxRDST, ldRDST;
step _{1A} wrGPR;	madr1A wrGPR;
step _{1B} <i>br</i> step _{2C} ;	madr1B cnt, bruncnd, madr2C;

Mikroprogram – formiranje adrese i čitanje operanda – preinkrement

Po koracima	Za uprav. jed.
step _{1C} ldRSRC;	madr1CldRSRC;
step _{1D} ldB;	madr1DldB;
step _{1E} incB;	madr1EincB;
step _{1F} mxMAR₂, mxMAR₁, ldMAR, mxRDST, ldRDST;	madr1F mxMAR2, mxMAR1, ldMAR, mxRDST, ldRDST;
step ₂₀ wrGPR;	madr20 wrGPR;
step ₂₁ <i>br</i> step _{2C} ;	madr21 cnt, bruncnd, madr2C;

Mikroprogram – formiranje adrese i čitanje operanda – direktno memorijsko

Po koracima	Za uprav. jed.
step ₂₂ mxMAR₁, IdMAR; step ₂₃ <i>br</i> step _{2C} ;	madr22 mxMAR1, IdMAR; madr23 cnt, bruncnd, madr2C;

Mikroprogram – formiranje adrese i čitanje operanda – indirektno memorijsko

Po koracima	Za uprav. jed.
step ₂₄ mxMAR₁, IdMAR;	madr24 mxMAR1, IdMAR;
step ₂₅ IdMBR;	madr25 IdMBR;
step ₂₆ mxMAR₁, mxMAR₀, IdMAR;	madr26 mxMAR1, mxMAR0, IdMAR;
step ₂₇ <i>br</i> step _{2C} ;	madr27 cnt, bruncnd, madr2C;

Mikroprogram – formiranje adrese i čitanje operanda – indirektno registarsko sa pomerajem

Po koracima	Za uprav. jed.
step ₂₈ ldRSRC;	madr28 ldRSRC;
step ₂₉ mxX₀, ldX, mxY₀, ldY;	madr29 mxX0, ldX, mxY0, ldY;
step _{2A} add, ldZ;	madr2A add, ldZ;
step _{2B} mxMAR₂, ldMAR;	madr2B mxMAR2, ldMAR;

Mikroprogram – formiranje adrese i čitanje operanda – čitanje operanda za mem. adr.

Po koracima	Za uprav. jed.
step _{2C} <i>br</i> (if STORE then step ₃₁);	madr2CbrSTORE, madr31;
step _{2D} ldMBR ;	madr2DldMBR;
step _{2E} mxB₀, ldB ;	madr2EmxB0, ldB;
step _{2F} <i>br</i> step ₃₁ ;	madr2Fcnt, bruncnd, madr31;

Mikroprogram – formiranje adrese i čitanje operanda – neposredno

Po koracima	Za uprav. jed.
step ₃₀ mxB₁, IdB;	madr30 mxB1, IdB;

Mikroprogram – formiranje adrese i čitanje operanda – izvršavanje operacije

Po koracima	Za uprav. jed.
<pre>step₃₁ br (case (LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTI, RTS) then (LOAD, step₃₂), (STORE, step₃₄), (ADD, step_{3C}), (AND, step₄₀), (ASR, step₄₄), (JZ, step₄₈), (JMP, step_{4C}), (JSR, step_{4A}), (RTI, step_{4E}), (RTS, step₅₂));</pre>	<pre>madr31 cnt, bropr;</pre>

Mikroprogram – formiranje adrese i čitanje operanda – LOAD

Po koracima	Za uprav. jed.
step ₃₂ mxACC, IdACC; step ₃₃ <i>br</i> step ₅₆ ;	madr32 mxACC, IdACC; madr33 cnt, bruncnd, madr56;

Mikroprogram – formiranje adrese i čitanje operanda – STORE

Po koracima	Za uprav. jed.
step ₃₄ <i>br (if immed then step₅₆);</i>	madr34 cnt, brimmed, madr56;
step ₃₅ <i>br (if regdir then step₃₉);</i>	madr35 cnt, brregdir, madr39;
step ₃₆ mxMBR₀, IdMBR;	madr36 mxMBR0, IdMBR;
step ₃₇ wrMEM;	madr37 wrMEM;
step ₃₈ <i>br step₅₆;</i>	madr38 cnt, bruncnd, madr56;
step ₃₉ IdRDST;	madr39 IdRDST;
step _{3A} wrGPR;	madr3A wrGPR;
step _{3B} <i>br step₅₆;</i>	madr3B cnt, bruncnd, madr56;

Mikroprogram – formiranje adrese i čitanje operanda – ADD

Po koracima	Za uprav. jed.
step _{3C} ldX, ldY;	madr3CldX, ldY;
step _{3D} add, ldZ;	madr3Dadd, ldZ;
step _{3E} ldACC;	madr3EldACC;
step _{3F} <i>br</i> step ₅₆ ;	madr3F cnt, bruncnd, madr56;

Mikroprogram – formiranje adrese i čitanje operanda – AND

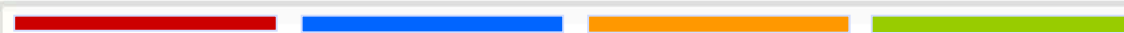
Po koracima	Za uprav. jed.
step ₄₀ ldX, ldY;	madr40 ldX, ldY;
step ₄₁ and, ldZ;	madr41 and, ldZ;
step ₄₂ ldACC;	madr42 ldACC;
step ₄₃ <i>br</i> step ₅₆ ;	madr43 cnt, bruncnd, madr56;

Mikroprogram – formiranje adrese i čitanje operanda – ASR

Po koracima	Za uprav. jed.
step ₄₄ ldY ;	madr44 ldY;
step ₄₅ asr, ldZ ;	madr45 asr, ldZ;
step ₄₆ ldACC ;	madr46 ldACC;
step ₄₇ <i>br</i> step ₅₆ ;	madr47 cnt, bruncnd, madr56;

Mikroprogram – formiranje adrese i čitanje operanda – JZ

Po koracima	Za uprav. jed.
step ₄₈ <i>br (if eql then step_{4C});</i> step ₄₉ <i>br step₅₆;</i>	madr48 cnt, breql, madr4C; madr49 cnt, bruncnd, madr56;

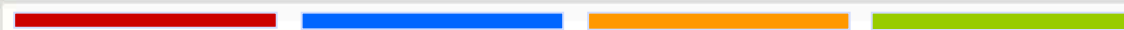


Mikroprogram – formiranje adrese i čitanje operanda – JSR

Po koracima	Za uprav. jed.
step _{4A} mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR; step _{4B} wrMEM;	madr4A mxMAR2, mxMAR0, ldMAR, decSP, mxMBR1, ldMBR; madr4B wrMEM;

Mikroprogram – formiranje adrese i čitanje operanda – JMP

Po koracima	Za uprav. jed.
step _{4C} ldPC;	madr4CldPC;
step _{4D} <i>br</i> step ₅₆ ;	madr4Dcnt, bruncnd, madr56;



Mikroprogram – formiranje adrese i čitanje operanda – RTI

Po koracima	Za uprav. jed.
step _{4E} incSP;	madr4E incSP;
step _{4F} mxMAR₂, mxMAR₀, ldMAR;	madr4F mxMAR2, mxMAR0, ldMAR;
step ₅₀ ldMBR;	madr50 ldMBR;
step ₅₁ ldPSW;	madr51 ldPSW;

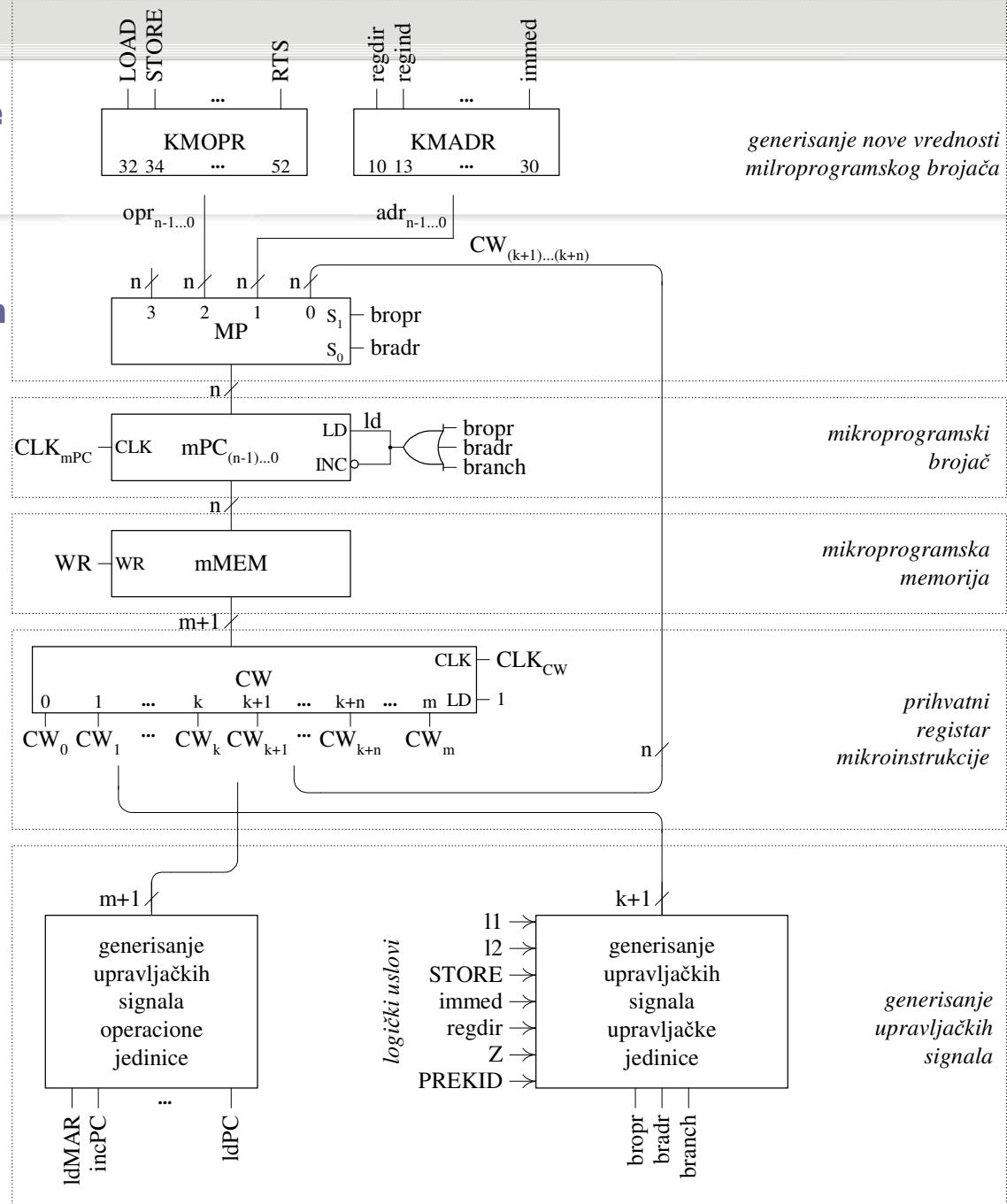
Mikroprogram – formiranje adrese i čitanje operanda – RTS

Po koracima	Za uprav. jed.
step ₅₂ incSP;	madr52 incSP;
step ₅₃ mxMAR₂, mxMAR₀, ldMAR;	madr53 mxMAR2, mxMAR0, ldMAR;
step ₅₄ ldMBR;	madr54 ldMBR;
step ₅₅ mxPC, ldPC;	madr55 mxPC, ldPC;

Mikroprogram – formiranje adrese i čitanje operanda – opsluživanje prekida

Po koracima	Za uprav. jed.
step ₅₆ <i>br (if then step₀₀);</i>	madr56 cnt, brnotPREKID, madr00;
step ₅₇ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR;	madr57 mxMAR2, mxMAR0, ldMAR, decSP, mxMBR1, ldMBR;
step ₅₈ wrMEM;	madr58 wrMEM;
step ₅₉ mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, mxMBR₀,ldMBR;	madr59 mxMAR2, mxMAR0, ldMAR, decSP, mxMBR1, mxMBR0,ldMBR;
step _{5A} wrMEM;	madr5A wrMEM;
step _{5B} mxX₁, ldX, mxY₁, ldY;	madr5B mxX1, ldX, mxY1, ldY;
step _{5C} add, ldZ;	madr5C add, ldZ;
step _{5D} mxMAR₂, ldMAR;	madr5D mxMAR2, ldMAR;
step _{5E} ldMBR;	madr5E ldMBR;
step _{5F} mxPC, ldPC;	madr5F mxPC, ldPC;
step ₆₀ <i>br step₀₀;</i>	madr60 cnt, bruncnd, madr00;

Struktura upravljačke jedinice mikroprogramske realizacije sa horizontalnim mikroprogramiranjem



Generisanje upravljačkih signala operacione jedinice

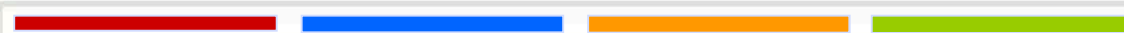
- Upravljački signali operacione jedinice se generišu na sledeći način:

$$\text{ldMAR} = \overline{\text{CW}}_0 \cdot \text{CW}_1$$

$$\text{mxMAR}_0 = \overline{\text{CW}}_0 \cdot \text{CW}_4$$

$$\text{wrGPR} = \overline{\text{CW}}_0 \cdot \text{CW}_{36}$$

- Na identičan način se generišu i preostali upravljački signali operacione jedinice



Generisanje upravljačkih signala upravljačke jedinice

- Upravljački signali upravljačke jedinice se generišu na sledeći način:

$$\text{bropr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{bradr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot \overline{CW_7}$$

$$\text{branch} = \text{bruncnd}$$

$$+ \text{brl1} * \text{l1} + \text{brl2} * \text{l2} + \text{brSTORE} * \text{STORE} + \text{brimmed} * \text{immed} \\ + \text{brregdir} * \text{regdir} + \text{breql} * \text{eql} + \text{brnotPREKID} * \overline{\text{PREKID}}$$

$$\text{bruncnd} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{brl1} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brl2} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot CW_7$$

$$\text{brSTORE} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot \overline{CW_7}$$

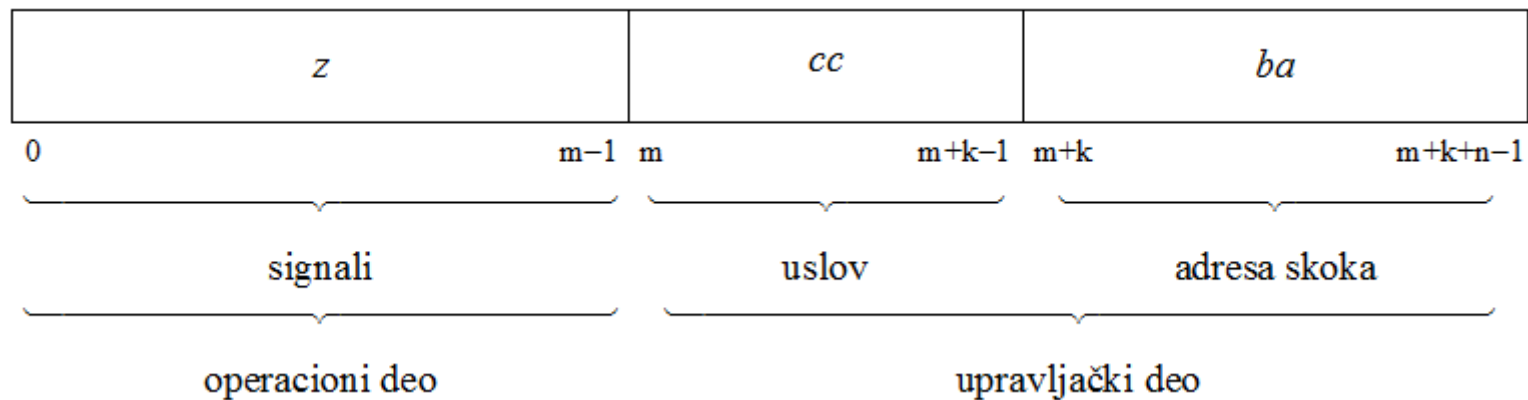
$$\text{brimmed} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot CW_7$$

$$\text{breql} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brnotPREKID} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot CW_7$$

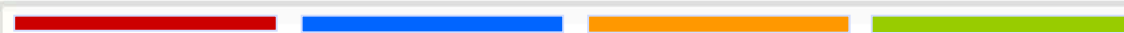
Mikroprogramska realizacija sa jednim tipom mikroinstrukcija

- Upravljačka jedinica sa spajanjem koraka se realizuje istim postupkom kao i upravljačka jedinica bez spajanja koraka
- U slučaju spajanja koraka postoji samo jedan tip mikroinstrukcije



Vertikalno kodiranje

- U slučaju vertikalnog kodiranja upravljačkih signala operacione jedinice jedna binarna vrednost se dodeljuje određenoj kombinaciji upravljačkih signala operacione jedinice neophodnoj da se u jednom koraku realizuje jedna mikrooperacija.
- Vertikalno kodiranje upravljačkih signala operacione jedinice se realizuje na isti način bez obzira na to da li su oni specificirani posebnim operacionim mikroinstrukcijama ili operacionim delom mikroinstrukcije.



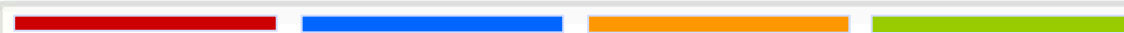
Vertikalno kodiranje

- Kombinacije upravljačkih signala operacione jedinice i simboličke oznake kodova – nisu svi kodovi prikazani

Kombinacija signala	Oznaka koda
/	V ₀₀
ldMAR	V ₀₁
mxMAR ₀ , ldMAR	V ₀₂
mxMAR ₁ , ldMAR	V ₀₃
mxMAR ₁ , mxMAR ₀ , ldMAR	V ₀₄
mxMAR ₂ , ldMAR	V ₀₅
mxMAR ₂ , mxMAR ₀ , ldMAR	V ₀₆
mxMAR ₂ , mxMAR ₁ , ldMAR	V ₀₇
ldMBR	V ₀₈
mxMBR ₀ , ldMBR	V ₀₉
mxMBR ₁ , ldMBR	V _{0A}

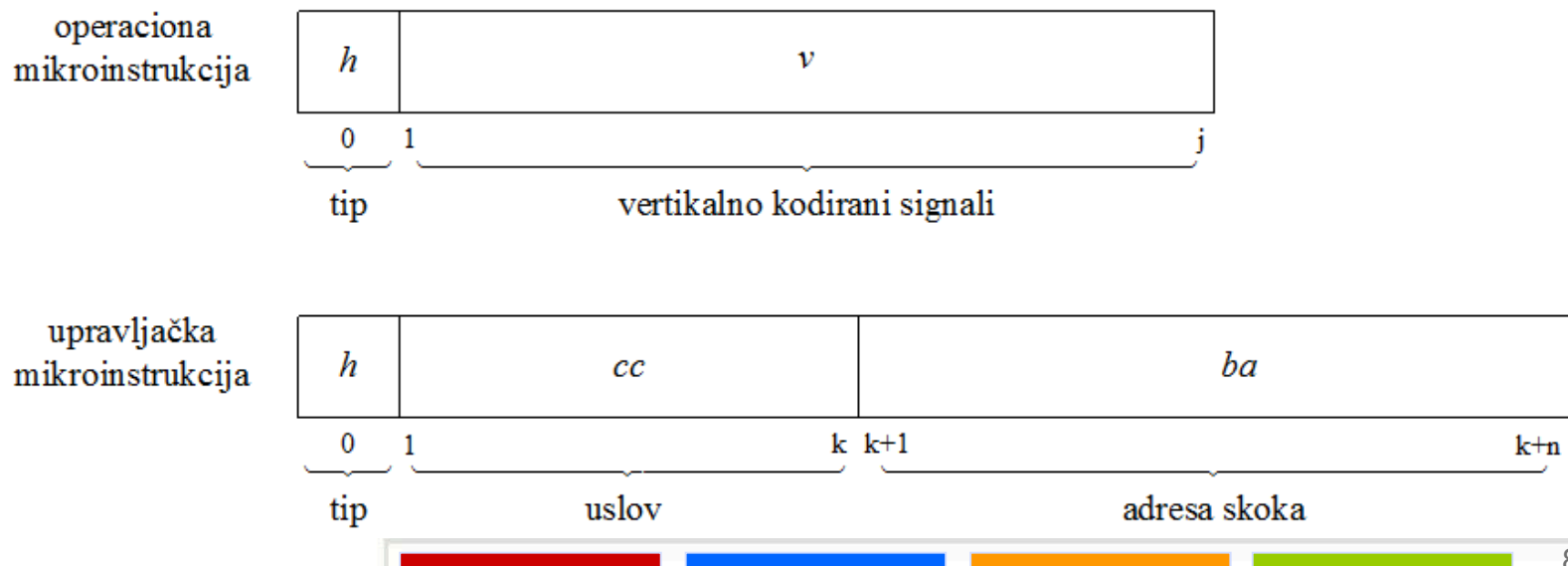
Vertikalno kodiranje

- Kombinacije upravljačkih signala su tako odabrane da njima mogu da se pokriju sve situacije iz sekvence upravljačkih signala po koracima.
- U ovom slučaju je za kodiranje kombinacija upravljačkih signala operacione jedinice potrebno 44 koda, pa je za kodiranje polja *kombinacija upravljačkih signala* operacionih mikroinstrukcija dovoljno 6 bita.



Vertikalno kodiranje

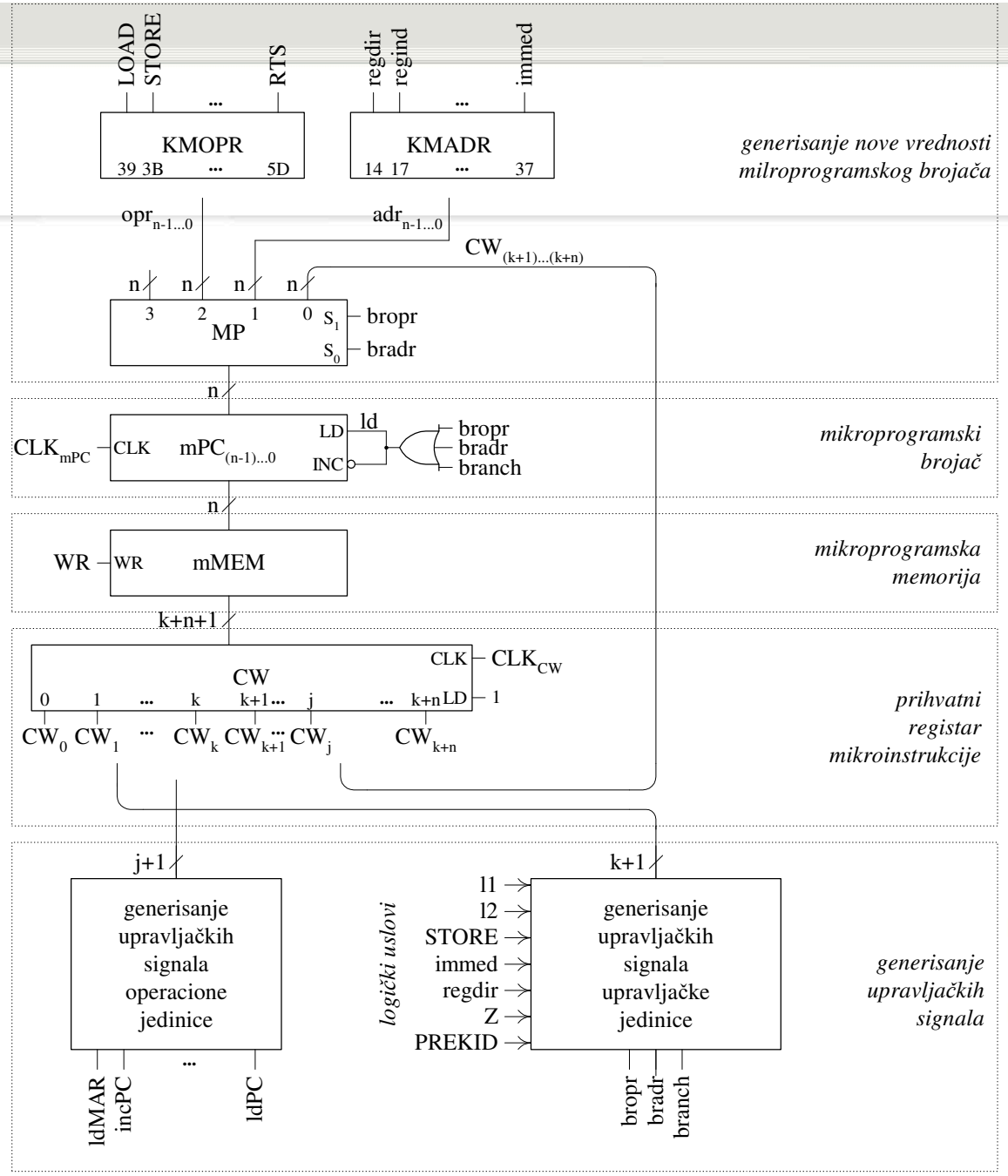
- Razlika je samo u poljima z i v operacionih mikroinstrukcija. U slučaju horizontalnog kodiranja upravljačkih signala operacione jedinice poseban bit polja z je dodeljen svakom signalu operacione jedinice, dok su u slučaju vertikalnog kodiranja upravljačkih signala operacione jedinice u polju v pojavljuju kodovi koji određuju kombinacije aktivnih vrednosti upravljačkih signala operacione jedinice neophodnih da se u jednom koraku realizuje jedna mikrooperacija.



Mikroprogram – primer

Po koracima		Za uprav. jed.	
step ₀₀	ldMAR;	madr00 V01;	! ldMAR !
step ₀₁	incPC;	madr01 V12;	! incPC !
step ₀₂	ldMBR;	madr02 V08;	! ldMBR !

Struktura upravljačke jedinice mikroprogramske realizacije sa vertikalnim mikroprogramiranjem



Generisanje upravljačkih signala operacione jedinice

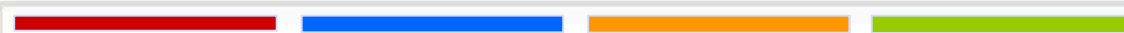
- Upravljački signali operacione jedinice se generišu na sledeći način:

$$\text{ldMAR} = V_{01} + V_{02} + V_{03} + V_{04} + V_{05} + V_{06} + V_{07}$$

$$\text{mxMAR}_0 = V_{02} + V_{04} + V_{06}$$

$$\text{wrGPR} = V_{15}$$

- Na identičan način se generišu i preostali upravljački signali operacione jedinice



Generisanje upravljačkih signala upravljačke jedinice

- Upravljački signali upravljačke jedinice se generišu na sledeći način:

$$\text{bropr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{bradr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot \overline{CW_7}$$

$$\text{branch} = \text{bruncnd}$$

$$+ \text{brl1} * \text{l1} + \text{brl2} * \text{l2} + \text{brSTORE} * \text{STORE} + \text{brimmed} * \text{immed} \\ + \text{brregdir} * \text{regdir} + \text{breql} * \text{eql} + \text{brnotPREKID} * \overline{\text{PREKID}}$$

$$\text{bruncnd} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{brl1} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brl2} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot CW_7$$

$$\text{brSTORE} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot \overline{CW_7}$$

$$\text{brimmed} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot CW_7$$

$$\text{breql} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brnotPREKID} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot CW_7$$

Mešovito kodiranje

- U slučaju mešovitog kodiranja upravljačkih signala operacione jedinice postoji više grupa signala, pri čemu je unutar grupe binarno kodiranje signala.
- Time su kombinovani pozitivni efekti horizontalnog i vertikalnog kodiranja.
- Na nivou grupa je horizontalno, a unutar grupe vertikalno kodiranje.
- Prilikom određivanja broja grupa i raspoređivanja signala po grupama treba omogućiti da se svi signali koji se u sekvenci upravljačkih signala po koracima javljaju u istom koraku mogu da pojave u istom koraku.

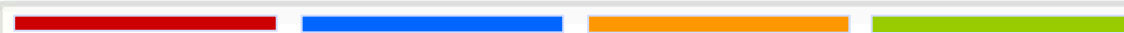
Mešovito kodiranje

- Kodiranje upravljačkih signala operacione jedinice i simboličke oznake kodova – nisu svi kodovi prikazani

polje M1		polje M2		polje M3		polje M4	
M1 ₀	–	M2 ₀	–	M3 ₀	–	M4 ₀	–
M1 ₁	mxMAR ₂	M2 ₁	mxMAR ₁	M3 ₁	mxMAR ₀	M4 ₁	lMAR
M1 ₂	add	M2 ₂	mxB ₁	M3 ₂	mxB ₀	M4 ₂	ldB
M1 ₃	and	M2 ₃	mxX ₁	M3 ₃	mxX ₀	M4 ₃	ldX
M1 ₄	asr	M2 ₄	mxPC	M3 ₄	wrGPR	M4 ₄	ldZ
M1 ₅	/	M2 ₅	ldIR1	M3 ₅	mxACC	M4 ₅	incB
M1 ₆	/	M2 ₆	ldIR2	M3 ₆	mxPSW	M4 ₆	decB
M1 ₇	/	M2 ₇	ldIRN	M3 ₇	wrMEM	M4 ₇	ldIVTP

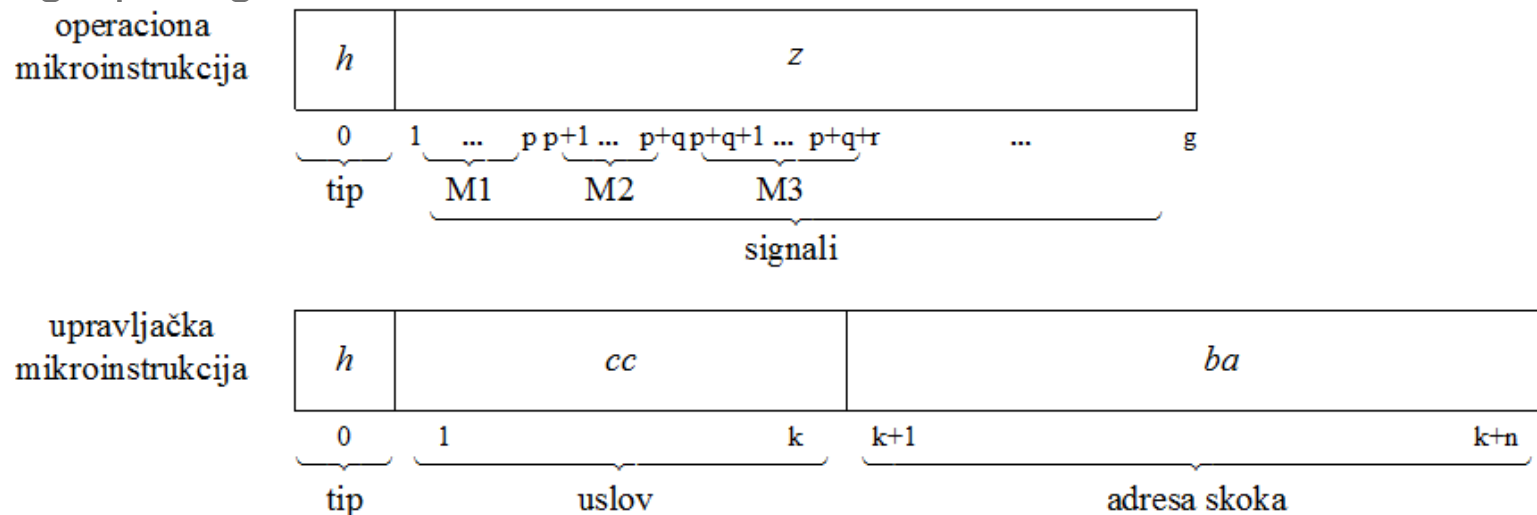
Mešovito kodiranje

- Kodom $M1_1$ se definiše da je vrednost polja M1 jedan i da je, iz grupe signala $mxMAR_2$, add, and i asr koji mogu da se specificiraju poljem M1, signal $mxMAR_2$ aktivan a ostali su neaktivni.
- Na sličan način se označavaju i vrednosti preostalih polja i time određuje po jedan signal iz svih preostalih grupa koji može da bude aktivan sa signalom $mxMAR_2$ iz grupe M1.
- Ukoliko za neke od grupa ni jedan od signala iz date grupe signala ne treba da bude aktivan, vrednost polja date grupe treba da bude nula.



Mešovito kodiranje

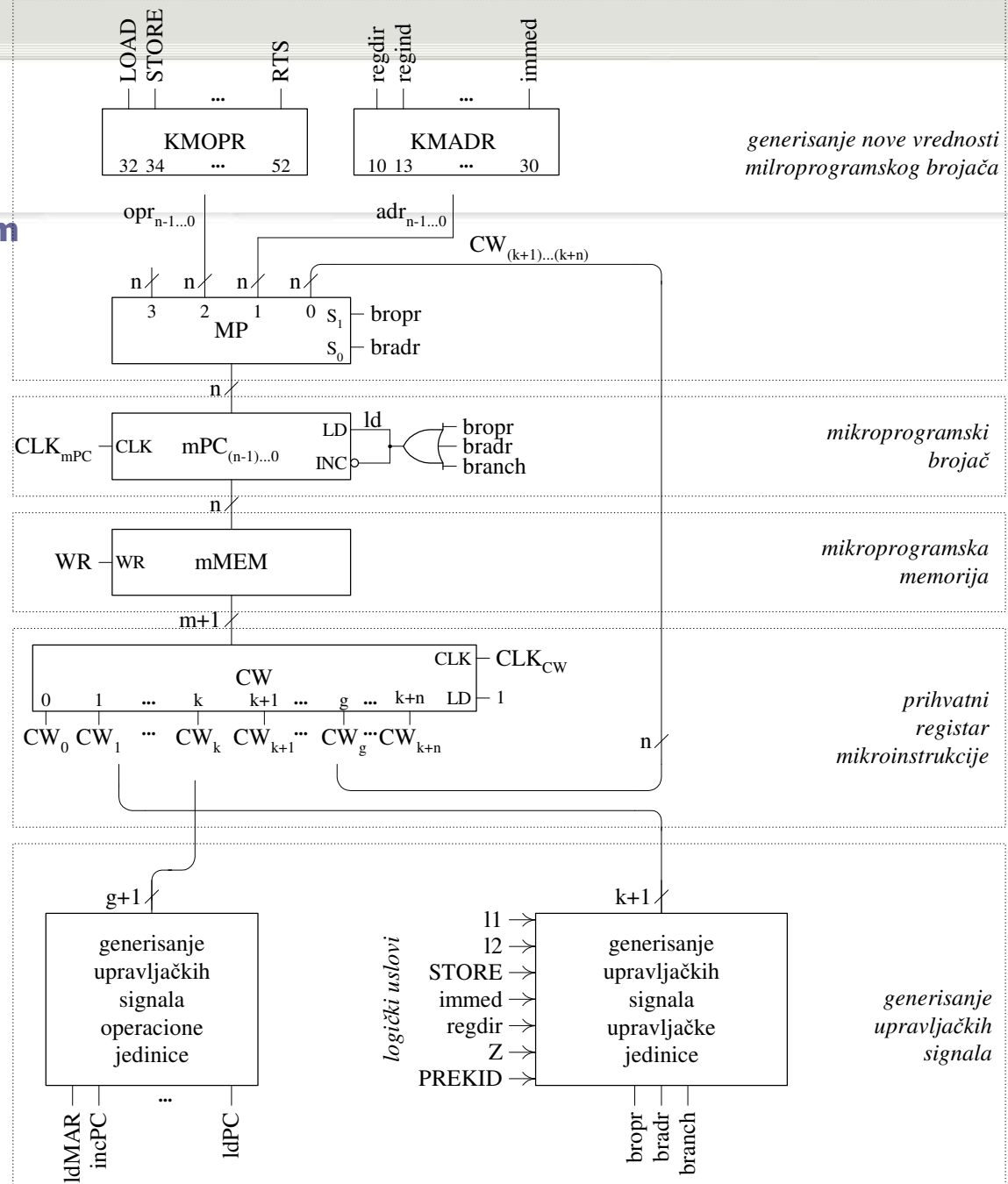
- Razlika je samo u poljima z operacionih mikroinstrukcija. U slučaju horizontalnog kodiranja upravljačkih signala operacione jedinice poseban bit polja z je dodeljen svakom signalu operacione jedinice, dok je u slučaju mešovitog kodiranja upravljačkih signala operacione jedinice polje z dužine g bita podeljeno na onoliko potpolja M1, M2, M3 itd. dužine p, q, r itd. bita, koliko ima grupa upravljačkih signala operacione jedinice. Binarnim vrednostima potpolja kodiraju se signali iz svake od grupa signala.



Mikroprogram – primer

Po koracima	Za uprav. jed.
step ₀₀ ldMAR, incPC;	madr00 M4 ₁ , M5 ₄ ; ! ldMAR, incPC !
step ₀₁ dMBR;	madr01 M8 ₁ ; ! ldMBR !
step ₀₂ ldIR1;	madr02 M2 ₅ ; ! ldIR1 !

Struktura upravljačke jedinice mikroprogramske realizacije sa mešovitim mikroprogramiranjem



Generisanje upravljačkih signala operacione jedinice

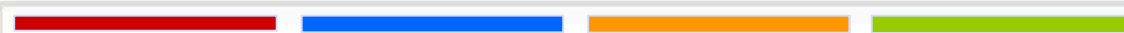
- Upravljački signali operacione jedinice se generišu na sledeći način:

$$\text{ldMAR} = \overline{CW_0} \cdot \overline{CW_{10}} \cdot \overline{CW_{11}} \cdot CW_{12}$$

$$\text{mxMAR}_0 = \overline{CW_0} \cdot \overline{CW_7} \cdot \overline{CW_8} \cdot CW_9$$

$$\text{wrGPR} = \overline{CW_0} \cdot CW_7 \cdot \overline{CW_8} \cdot \overline{CW_9}$$

- Na identičan način se generišu i preostali upravljački signali operacione jedinice



Generisanje upravljačkih signala upravljačke jedinice

- Upravljački signali upravljačke jedinice se generišu na sledeći način:

$$\text{bropr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{bradr} = CW_0 \cdot CW_4 \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot \overline{CW_7}$$

$$\text{branch} = \text{bruncnd}$$

$$+ \text{brl1} * \text{l1} + \text{brl2} * \text{l2} + \text{brSTORE} * \text{STORE} + \text{brimmed} * \text{immed} \\ + \text{brregdir} * \text{regdir} + \text{breql} * \text{eql} + \text{brnotPREKID} * \overline{\text{PREKID}}$$

$$\text{bruncnd} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot \overline{CW_6} \cdot CW_7$$

$$\text{brl1} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brl2} = CW_0 \cdot \overline{CW_4} \cdot \overline{CW_5} \cdot CW_6 \cdot CW_7$$

$$\text{brSTORE} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot \overline{CW_7}$$

$$\text{brimmed} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot \overline{CW_6} \cdot CW_7$$

$$\text{breql} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot \overline{CW_7}$$

$$\text{brnotPREKID} = CW_0 \cdot \overline{CW_4} \cdot CW_5 \cdot CW_6 \cdot CW_7$$

Питања?

Електротехнички Факултет
Универзитет у Београду

