

Organizacija procesora

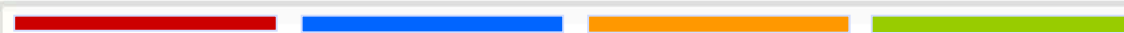


Arhitektura i organizacija procesora

- Procesor je sa jednoadresnim formatom instrukcija.
- Dužina instrukcije je promenljiva i kreće se od 1, 2 do N memorijskih reči.
- Prva reč uvek specificira kod operacije.
- Bezadresne instrukcije su:
 - instrukcija povratka iz potprograma (RTS) i
 - povratka iz prekidne rutine (RTI).
 - Dužina instrukcija je jedna memorijska reč.
- Instrukcije skoka su:
 - instrukcija uslovnog skoka ukoliko je rezultat nula (JZ),
 - bezuslovnog skoka (JMP) i
 - skoka na potprograma (JSR).
 - Kod ovih instrukcija druga i sledećih nekoliko reči specificiraju adresu skoka.

Arhitektura i organizacija procesora (2)

- Adresne instrukcije su:
 - instrukcija prenosa u akumulator (LOAD),
 - instrukcija prenosa iz akumulatora (STORE),
 - aritmetička instrukcija sabiranja (ADD),
 - logička instrukcija logički proizvod (AND) i
 - instrukcija aritmetičkog pomeranja udesno za jedno mesto (ASR).
 - Kod ovih instrukcija druga reč specificira način adresiranja i adresu registra opšte namene. Kod registarskih adresiranja dužina instrukcije je dve reči. Kod memorijskih adresiranja treća i sledećih nekoliko reči specificiraju adresu, pomeraj ili neposrednu veličinu.
- Podaci su celobrojne veličine bez znaka dužine koja odgovara širini memorijske reči.

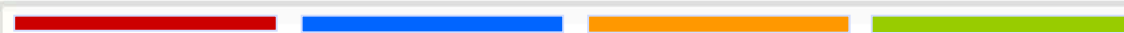


Arhitektura i organizacija procesora (3)

- Načini adresiranja su specificirani sa nekoliko bitova druge reči instrukcije i to:
 - registarsko direktno adresiranje,
 - registarsko indirektno adresiranje,
 - registarsko indirektno adresiranje sa postdekrementiranjem,
 - registarsko indirektno adresiranje sa preinkrementiranjem,
 - memorijsko direktno adresiranje,
 - memorijsko indirektno adresiranje,
 - registarsko indirektno sa pomerajem i
 - neposredno adresiranje.

Arhitektura i organizacija procesora (3)

- Kod adresiranja koja koriste neki od registara opšte namene R adresa registra se specificira preostalim bitovima druge reči instrukcije.
- Kod registarskog direktnog adresiranja, registarskog indirektnog adresiranja, registarskog indirektnog adresiranja sa postdekrementiranjem i registarskog indirektnog adresiranja sa preinkrementiranjem dužina instrukcije je dve reči. Kod memorijskog direktnog i memorijskog indirektnog adresiranja treća i preostale reči instrukcije sadrže adresu memorijske lokacije. Bitovi druge reči koji sadrže adresu registra opšte namene se ne koriste. Kod registarskog indirektnog adresiranja sa pomerajem treća i preostale reči instrukcije sadrže pomeraj. Bitovi druge reči koji sadrže adresu registra opšte namene se koriste. Kod neposrednog adresiranja treća i preostale reči instrukcije sadrže podatak. Bitovi druge reči koji sadrže adresu registra opšte namene se ne koriste.

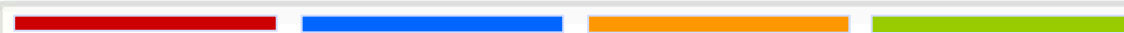


Arhitektura i organizacija procesora (4)

- U procesoru postoji:
 - programski brojač PC,
 - adresni registar memorije MAR,
 - prihvatni registar podatka memorije MBR,
 - prihvatni registar instrukcije IR dužine N memorijskih reči,
 - akumulator A,
 - prihvatni registar podatka B,
 - registri opšte name R,
 - programska statusna reč PSW,
 - ukazivač na vrh steka SP,
 - registar broja ulaza u tabelu sa adresama prekidnih rutina BR i
 - ukazivač na tabelu sa adresama prekidnih rutina IVTP.
 - Svi registri sem registra IR su dužine koja odgovara širini memorijske reči, dok je registar IR dužine 1, 2 do N memorijskih reči.
- Stek raste prema nižim memorijskim lokacijama, a registar SP ukazuje na prvu slobodnu memorijsku lokaciju.

Arhitektura i organizacija procesora (5)

- Zahtevi za prekid dolaze od 4 ulazno/izlazna uređaja po linijama označenim od 0 do 3. Po liniji 0 stiže zahtev za prekid najnižeg, a po liniji 3 najvišeg prioriteta. Broj linije najvišeg prioriteta po kojoj je stigao zahtev za prekid nalazi se u binarnom obliku u registru BR dužine 2 razreda. Adrese prekidnih rutina 4 ulazno/izlazna uređaja koji po linijama označenim od 0 do 3 šalju zahteve za prekid nalaze se u ulazima 0 do 3 tabele sa adresama prekidnih rutina. Sadržaj registra BR predstavlja broj ulaza u tabelu sa adresama prekidnih rutina. Početna adresa tabele sa adresama prekidnih rutina se nalazi u registru IVTP. U okviru hardverskog dela opsluživanja zahteva za prekid na stek sa stavljaju samo registri PC i PSW.

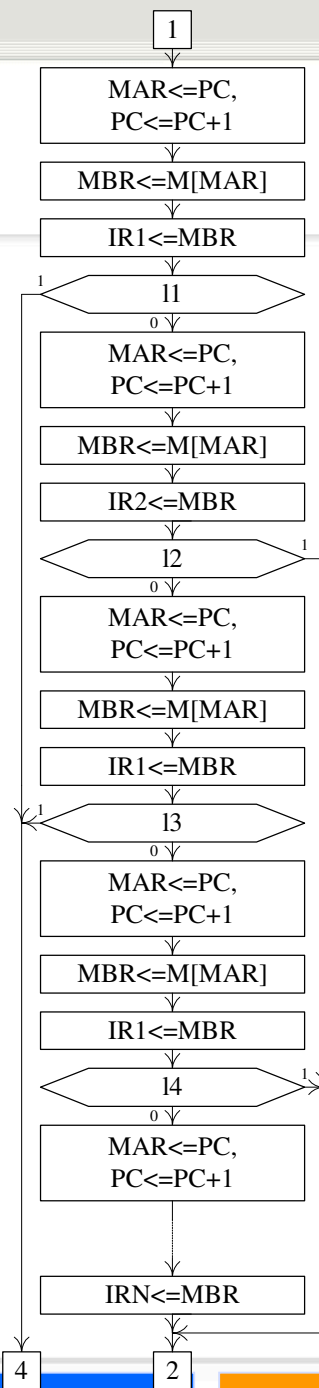


Arhitektura i organizacija procesora (6)

- Najpre treba nacrtati dijagram toka faza izvršavanja instrukcije i to:
 - faze čitanja instrukcije,
 - faze formiranja adrese i čitanja operanda,
 - faza izvršavanja operacija LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTS i RTI i
 - faze opsluživanja zahteva za prekid.

Dijagram toka

- Faza čitanja instrukcije

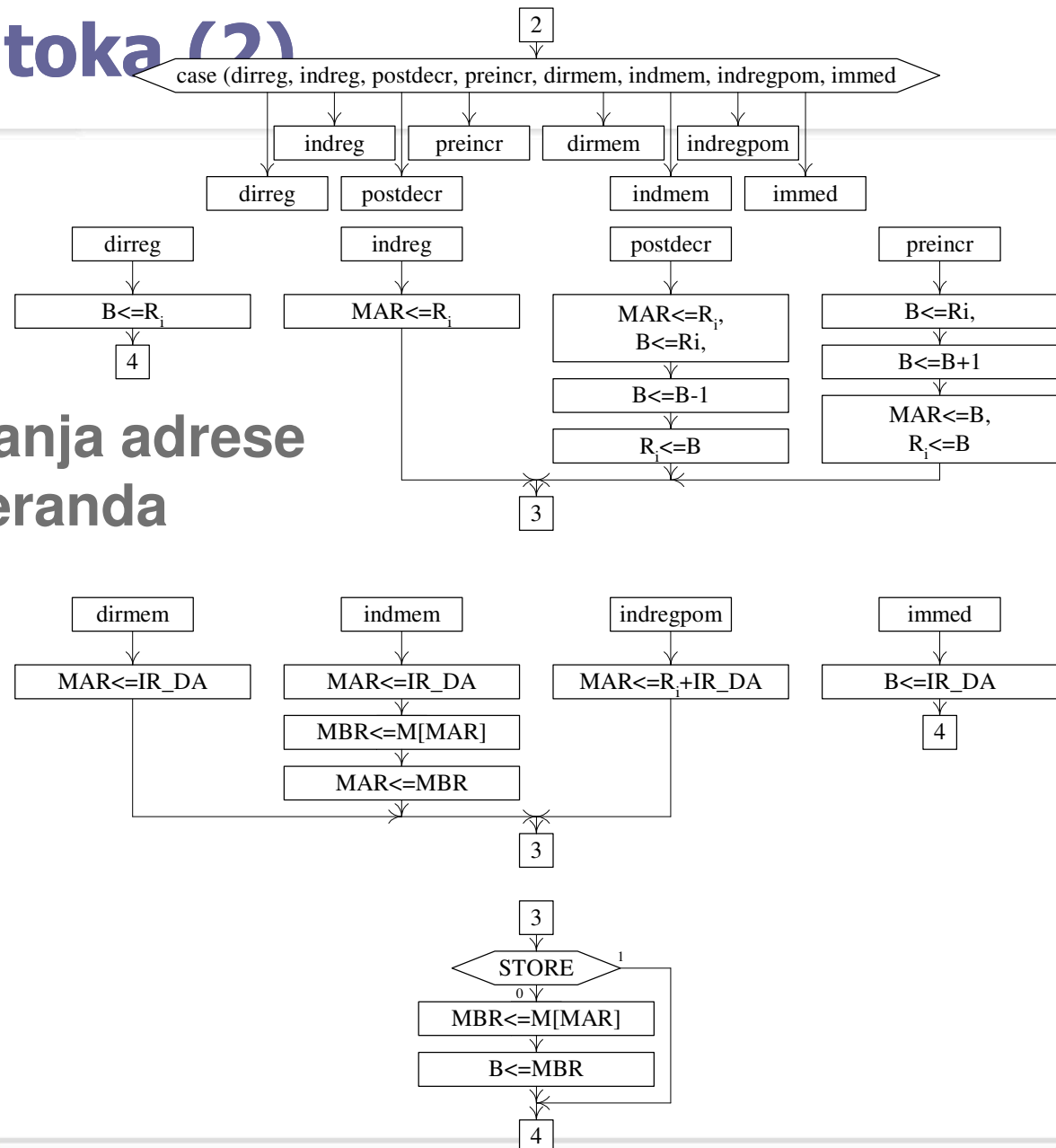


4

2

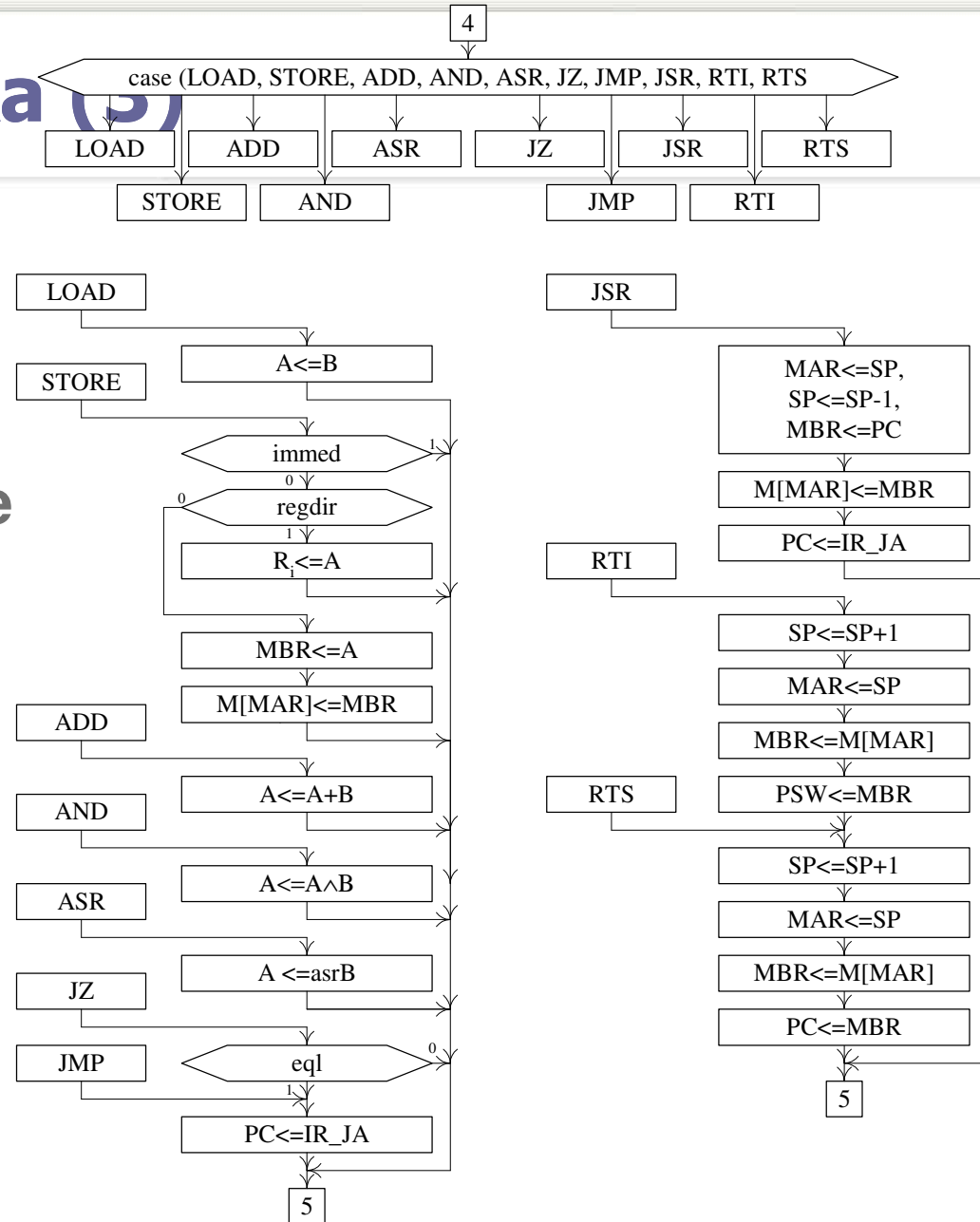
Dijagram toka (2)

- Faza formiranja adrese i čitanja operanda



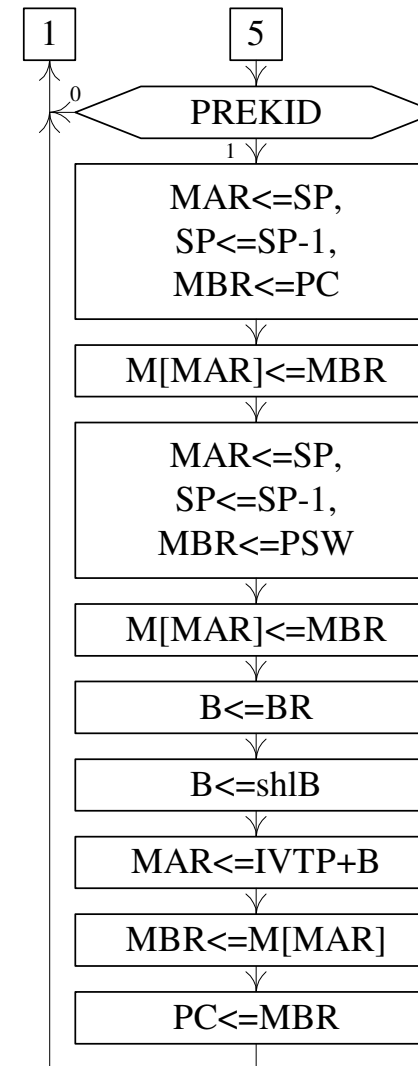
Dijagram toka

- Faza izvršavanje operacija



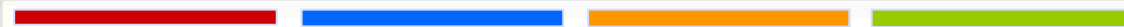
Dijagram toka (4)

- Faza opsluživanje prekida

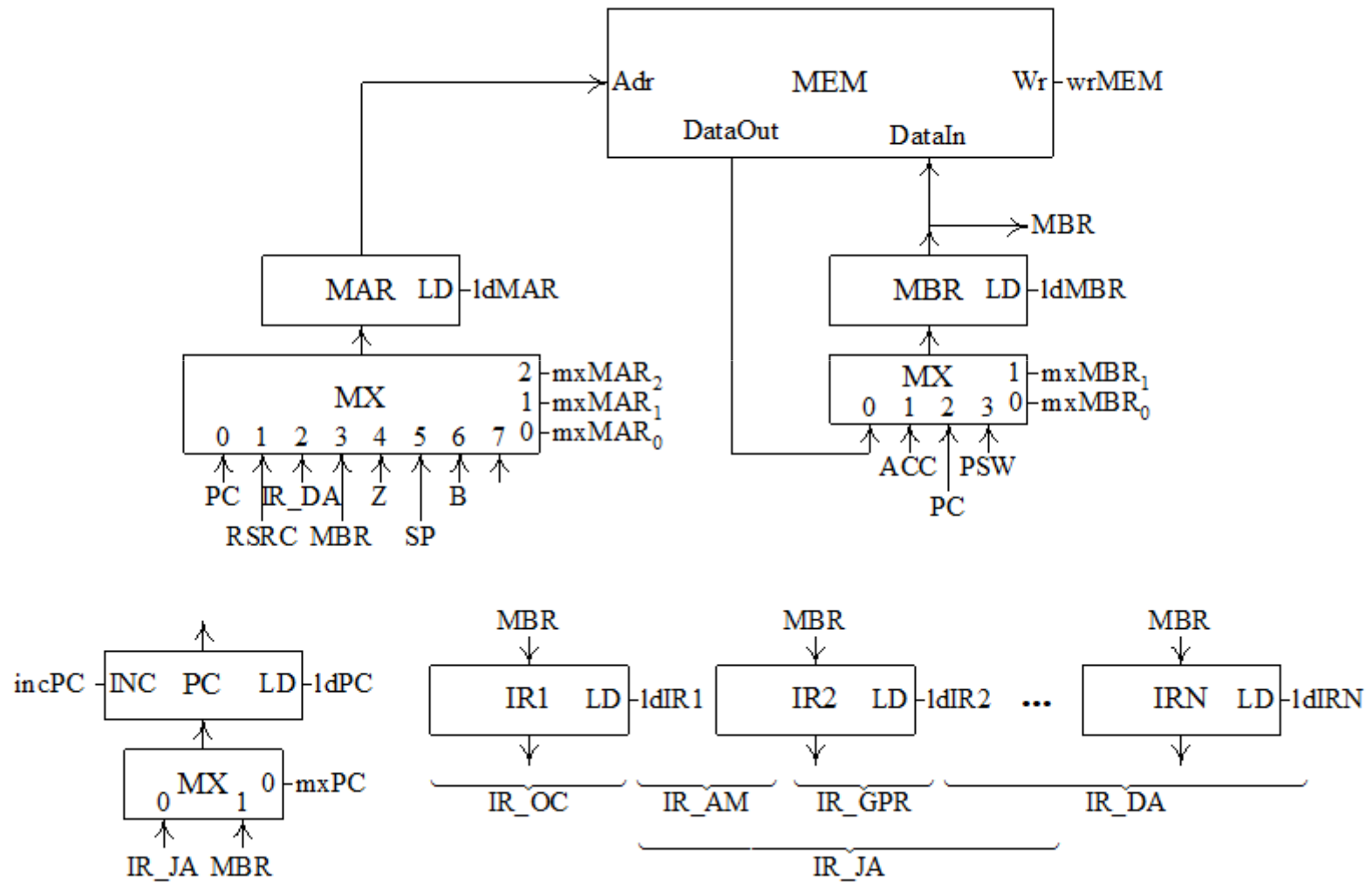


Operaciona jedinica

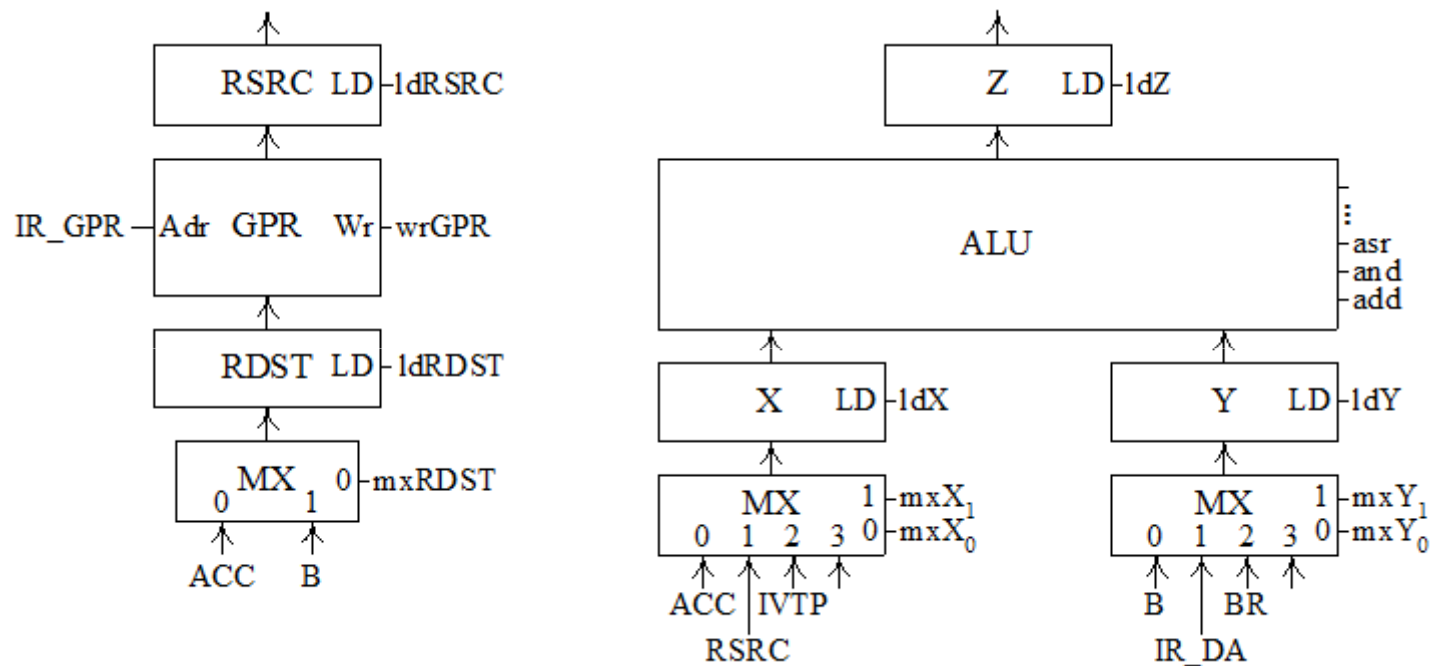
- Sa direktnim vezama
- Sa jednom magistralom
- Sa dve magistrale
- Sa tri magistrale



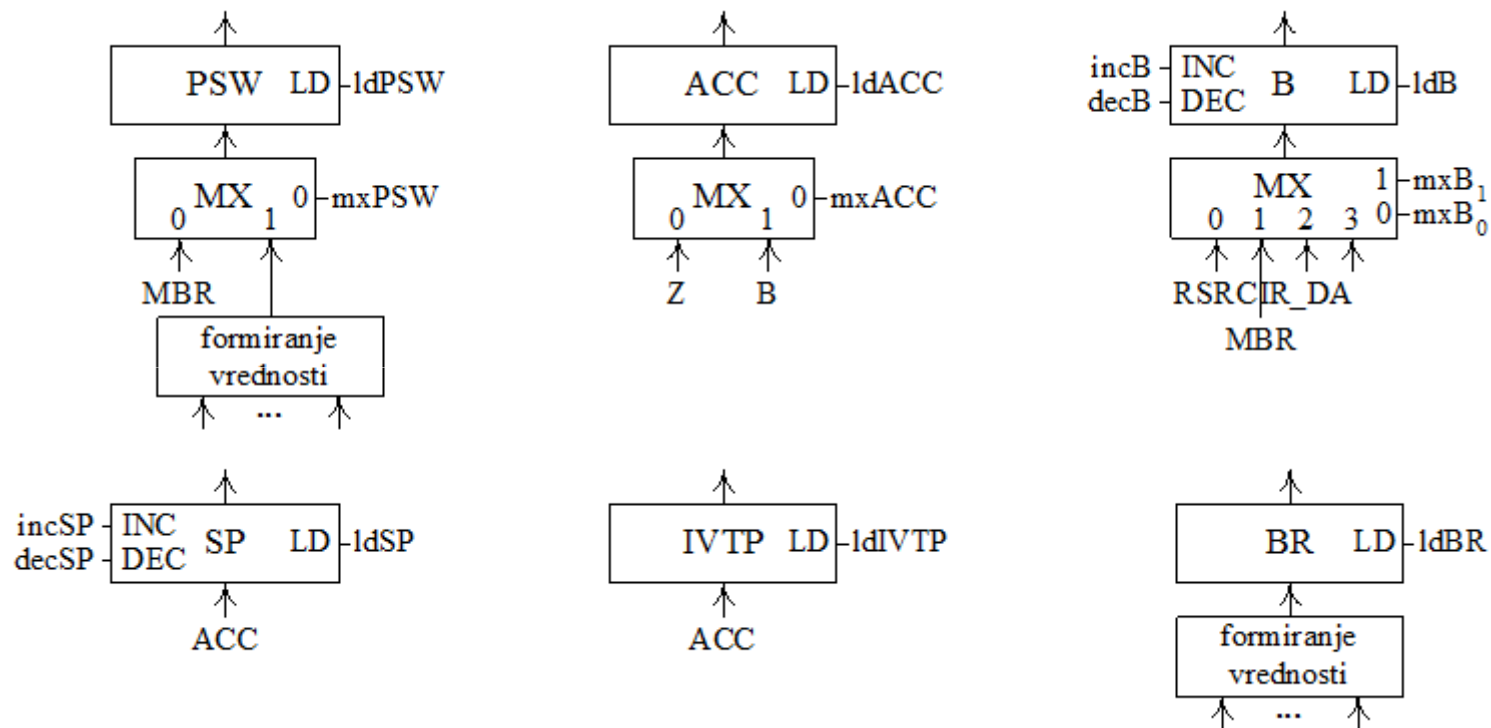
Operaciona jedinica sa direktnim vezama



Operaciona jedinica sa direktnim vezama (2)

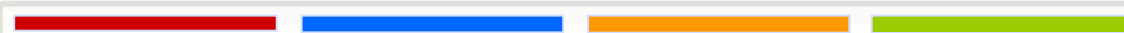


Operaciona jedinica sa direktnim vezama (3)



Algoritam generisanja upravljačkih signala

- Algoritam generisanja upravljačkih signala je formiran na osnovu dijagrama toka operacija i dat u obliku sekvence upravljačkih signala po koracima.
- U sekvenci upravljačkih signala po koracima se koriste iskazi za signale i skokove. Iskazi za signale su oblika
- **signali.**
- Ovaj iskaz sadrži spisak upravljačkih signala operacione jedinice i određuje koji se signali bezuslovno generišu.

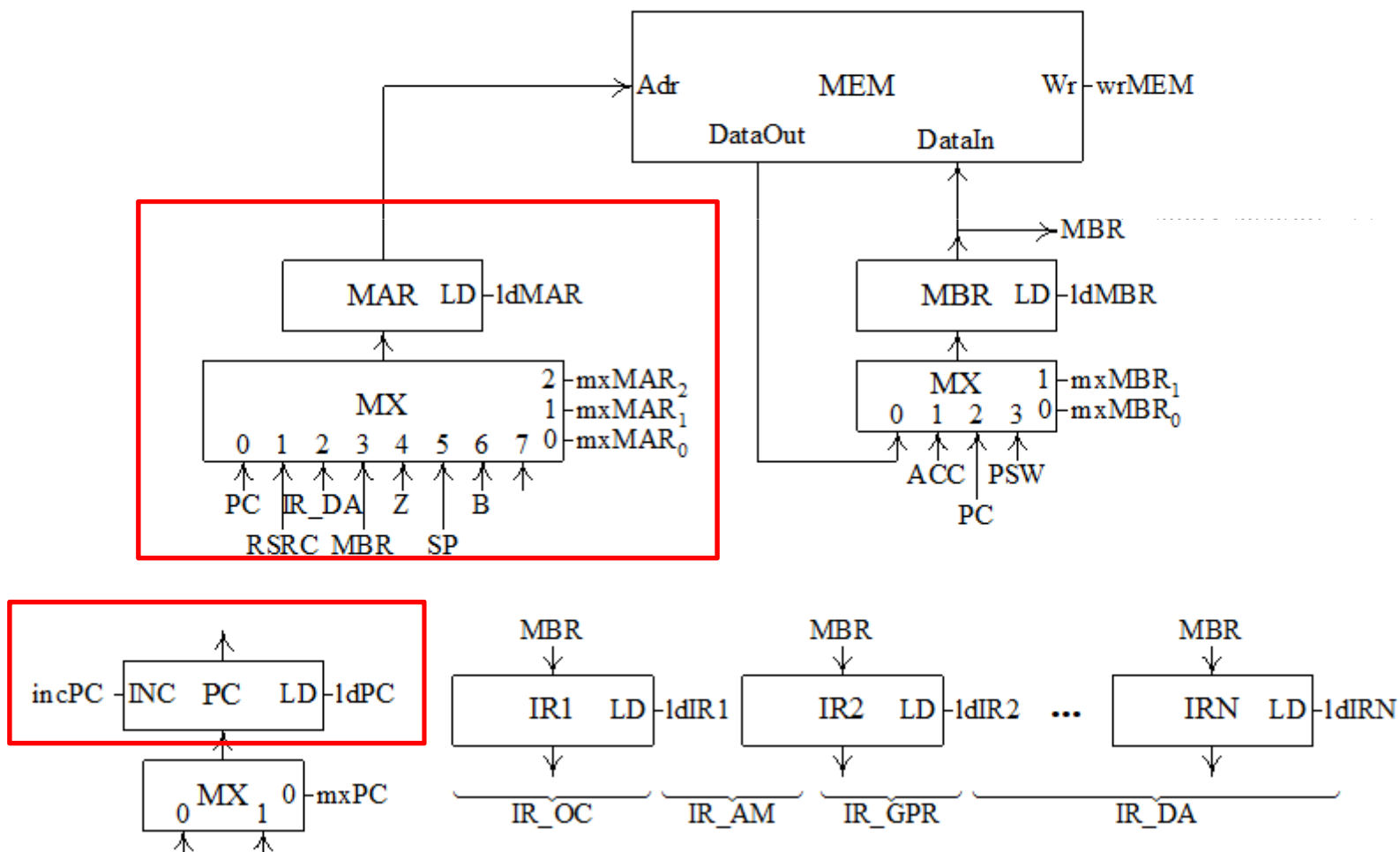
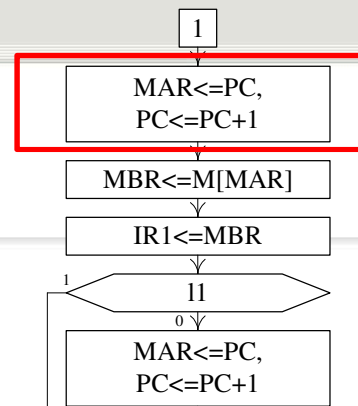


Algoritam generisanja upravljačkih signala (2)

- Iskazi za skokove su oblika
- *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}).
- Prvi iskaz sadrži korak step_A na koji treba bezuslovno preći i u daljem tekstu se referiše kao bezuslovni skok. Drugi iskaz sadrži signal **uslov** i korak step_A i određuje korak step_A na koji treba preći ukoliko signal **uslov** ima aktivnu vrednost i u daljem tekstu se referiše kao uslovni skok. Treći iskaz sadrži signale **uslov**₁, ..., **uslov**_n i korake step_{A1} , ..., step_{An} i određuje na koji od koraka step_{A1} , ..., step_{An} treba preći u zavisnosti od toga koji od signala **uslov**₁, ..., **uslov**_n ima aktivnu vrednost i u daljem tekstu se referiše kao višestruki uslovni skok.

Sekvenca upravljačkih signala

- ! Čitanje instrukcije !
- $step_{00}$ $ldMAR, incPC;$



Sekvenca upravljačkih signala (2)

- ! Formiranje adrese i čitanje operanda !
- $step_{0F}$ *br* (*case* (**dirreg**, **indreg**, **postdec**, **preinc**,
dirmem, **indmem**, **indregpom**, **immed**) *then*
- (**dirreg**, $step_{10}$), (**indreg**, $step_{13}$), (**postdec**, $step_{16}$), (**preinc**,
 $step_{1C}$),
- (**dirmem**, $step_{22}$), (**indmem**, $step_{24}$), (**indregpom**, $step_{28}$),
(**immed**, $step_{30}$));
- ! Direktno registarsko !
- $step_{10}$ **ldRSRC**;
- $step_{11}$ **ldB**;
- $step_{12}$ *br* $step_{31}$;
- ! Indirektno registarsko !
- $step_{13}$ **ldRSRC**;
- $step_{14}$ **mxMAR₀**, **ldMAR**;
- $step_{15}$ *br* $step_{2C}$;

Sekvenca upravljačkih signala (3)

- ! Postdekrement !
- step₁₆ **ldRSRC;**
- step₁₇ **mxMAR₀, ldMAR, ldB;**
- step₁₈ **decB;**
- step₁₉ **mxRDST, ldRDST;**
- step_{1A} **wrGPR;**
- step_{1B} *br step_{2C};*
- ! Preinkrement !
- step_{1C} **ldRSRC;**
- step_{1D} **ldB;**
- step_{1E} **incB;**
- step_{1F} **mxMAR₂, mxMAR₁, ldMAR, mxRDST, ldRDST;**
- step₂₀ **wrGPR;**
- step₂₁ *br step_{2C};*

Sekvenca upravljačkih signala (4)

- ! Direktno memorijsko !
- step₂₂ **mxMAR₁, IdMAR;**
- step₂₃ *br* step_{2C};
- ! Indirektno memorijsko!
- step₂₄ **mxMAR₁, IdMAR;**
- step₂₅ **IdMBR;**
- step₂₆ **mxMAR₁, mxMAR₀, IdMAR;**
- step₂₇ *br* step_{2C};
- ! Indirektno registarsko sa pomerajem !
- step₂₈ **IdRSRC;**
- step₂₉ **mxX₀, IdX, mxY₀, IdY;**
- step_{2A} **add, IdZ;**
- step_{2B} **mxMAR₂, IdMAR;**

Sekvenca upravljačkih signala (5)

- ! Čitanje operanda za memorijska adresiranja !
- $step_{2C}$ *br (if **STORE** then $step_{31}$);*
- $step_{2D}$ **ldMBR;**
- $step_{2E}$ **mxB₀, ldB;**
- $step_{2F}$ *br $step_{31}$;*
- ! Neposredno !
- $step_{30}$ **mxB₁, ldB;**

Sekvenca upravljačkih signala (6)

- ! Izvršavanje operacije !
- $step_{31}$ *br* (*case* (**LOAD**, **STORE**, **ADD**, **AND**, **ASR**, **JZ**, **JMP**, **JSR**, **RTI**, **RTS**) *then*
- (**LOAD**, $step_{32}$), (**STORE**, $step_{34}$),
- (**ADD**, $step_{3C}$), (**AND**, $step_{40}$), (**ASR**, $step_{44}$),
- (**JZ**, $step_{48}$), (**JMP**, $step_{4C}$), (**JSR**, $step_{4A}$), (**RTI**, $step_{4E}$), (**RTS**, $step_{52}$));
- ! LOAD !
- $step_{32}$ **mxACC**, **ldACC**;
- $step_{33}$ *br* $step_{56}$;
- ! STORE !
- $step_{34}$ *br* (*if immed then* $step_{56}$);
- $step_{35}$ *br* (*if regdir then* $step_{39}$);
- $step_{36}$ **mxMBR₀**, **ldMBR**;
- $step_{37}$ **wrMEM**;
- $step_{38}$ *br* $step_{56}$;
- $step_{39}$ **ldRDST**;
- $step_{3A}$ **wrGPR**;
- $step_{3B}$ *br* $step_{56}$;

Sekvenca upravljačkih signala (7)

- ! ADD !
- step_{3C} **ldX, ldY;**
- step_{3D} **add, ldZ;**
- step_{3E} **ldACC;**
- step_{3F} *br step₅₆;*
- ! AND !
- step₄₀ **ldX, ldY;**
- step₄₁ **and, ldZ;**
- step₄₂ **ldACC;**
- step₄₃ *br step₅₆;*
- ! ASR !
- step₄₄ **ldY;**
- step₄₅ **asr, ldZ;**
- step₄₆ **ldACC;**
- step₄₇ *br step₅₆;*

Sekvenca upravljačkih signala (8)

- ! JZ !
- step₄₈ *br* (if **eq** then step_{4C});
- step₄₉ *br* step₅₆;
- ! JSR !
- step_{4A} **mxMAR₂**, **mxMAR₀**, **ldMAR**, **decSP**, **mxMBR₁**, **ldMBR**;
- step_{4B} **wrMEM**;
- ! JMP !
- step_{4C} **ldPC**;
- step_{4D} *br* step₅₆;
- ! RTI !
- step_{4E} **incSP**;
- step_{4F} **mxMAR₂**, **mxMAR₀**, **ldMAR**;
- step₅₀ **ldMBR**;
- step₅₁ **ldPSW**;
- ! RTS !
- step₅₂ **incSP**;
- step₅₃ **mxMAR₂**, **mxMAR₀**, **ldMAR**;
- step₅₄ **ldMBR**;
- step₅₅ **mxPC**, **ldPC**;

Sekvenca upravljačkih signala (9)

- ! Opsluživanje prekida !
- step₅₆ *br (if then step₀₀)*;
- step₅₇ **mxMAR₂, mxMAR₀, IdMAR, decSP, mxMBR₁,
IdMBR**;
- step₅₈ **wrMEM**;
- step₅₉ **mxMAR₂, mxMAR₀, IdMAR, decSP, mxMBR₁,
mxMBR₀, IdMBR**;
- step_{5A} **wrMEM**;
- step_{5B} **mxX₁, IdX, mxY₁, IdY**;
- step_{5C} **add, IdZ**;
- step_{5D} **mxMAR₂, IdMAR**;
- step_{5E} **IdMBR**;
- step_{5F} **mxPC, IdPC**;
- step₆₀ *br step₀₀*;

Spajanje koraka

- Operacioni korak i prvi sledeći upravljački korak u nekim situacijama mogu da se spoje u isti korak. Time se ukupan broj koraka neophodnih za izvršavanje instrukcije smanjuje, čime se povećava brzina izvršavanja instrukcija.
- Ako je upravljački korak bezuslovni skok, tada se dati upravljački korak i prethodni korak koji je operacioni korak mogu spojiti ukoliko se na dati upravljački korak prelazi samo iz prethodnog koraka koji je operacioni korak a ne i iz još nekog koraka koji je upravljački korak. Primeri su koraci step_{11} **ldB** i step_{12} *br* step_{31} , zatim koraci step_{14} **mxMAR₀**, **ldMAR** i step_{15} *br* step_{2C} itd.

Spajanje koraka (2)

- Ako je upravljački korak uslovni skok, tada se dati upravljački korak i prethodni korak koji je operacioni korak mogu spojiti ukoliko signal logičkog uslova koji se konsultuje pri uslovnom skoku ne zavisi od mikrooperacija izvršenih na osnovu upravljačkih signala generisanih u operacionom koraku i ukoliko se na dati upravljački korak prelazi samo iz prethodnog koraka koji je operacioni korak a ne i iz još nekog koraka koji je upravljački korak. U suprotnom slučaju koraci se ne mogu spojiti.
- Ako je upravljački korak višestruki uslovni skok, tada se dati upravljački korak i prethodni korak koji je operacioni korak mogu spojiti ukoliko ni jedan od signala logičkih uslova koji se konsultuju pri višestrukome uslovnom skoku ne zavisi od mikrooperacija izvršenih na osnovu upravljačkih signala generisanih u operacionom koraku i ukoliko se na dati upravljački korak prelazi samo iz prethodnog koraka koji je operacioni korak a ne i iz još nekog koraka koji je upravljački korak. U suprotnom slučaju koraci se ne mogu spojiti.

Spajanje koraka (3)

- Operacioni korak i prvi sledeći upravljački korak ne bi mogli da se spoje u isti korak i u situacijama kada operacioni korak traje više od jedne periode signala takta.

Sekvenca upravljačkih signala sa spajanjem

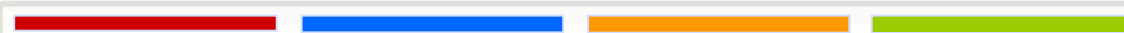
- ! Čitanje instrukcije !
- step₀₀ **ldMAR, incPC;**
- step₀₁ **ldMBR;**
- step₀₂ **ldIR1;**
- step₀₃ *br (if I1 then step_{2A});*
- step₀₄ **ldMAR, incPC;**
- step₀₅ **ldMBR;**
- step₀₆ **ldIR2;**
- step₀₇ *br (if I2 then step_{0F});*
- step₀₈ **ldMAR, incPC;**
- ...
- step_{0E} **ldIRN;**

Sekvenca upravljačkih signala sa spajanjem (2)

- ! Formiranje adrese i čitanje operanda !
- $step_{0F}$ *br* (*case* (**dirreg**, **indreg**, **postdec**, **preinc**,
dirmem, **indmem**, **indregpom**, **immed**) *then*
- (**dirreg**, $step_{10}$), (**indreg**, $step_{12}$), (**postdec**, $step_{14}$), (**preinc**,
 $step_{19}$),
- (**dirmem**, $step_{1E}$), (**indmem**, $step_{1F}$), (**indregpom**, $step_{22}$),
(**immed**, $step_{29}$));
- ! Direktno registarsko !
- $step_{10}$ **IdRSRC**;
- $step_{11}$ **IdB**, *br* $step_{2A}$;
- ! Indirektno registarsko !
- $step_{12}$ **IdRSRC**;
- $step_{13}$ **mxMAR₀**, **IdMAR**, *br* $step_{26}$;

Sekvenca upravljačkih signala sa spajanjem (3)

- ! Postdekrement !
- step₁₄ **ldRSRC;**
- step₁₅ **mxMAR₀, ldMAR, ldB;**
- step₁₆ **decB;**
- step₁₇ **mxRDST, ldRDST;**
- step₁₈ **wrGPR, br step₂₆;**
- ! Preinkrement !
- step₁₉ **ldRSRC;**
- step_{1A} **ldB;**
- step_{1B} **incB;**
- step_{1C} **mxMAR₂, mxMAR₁, ldMAR, mxRDST, ldRDST;**
- step_{1D} **wrGPR, br step₂₆;**
- ! Direktno memorijsko !
- step_{1E} **mxMAR₁, ldMAR, br step₂₆;**



Sekvenca upravljačkih signala sa spajanjem (4)

- ! Indirektno memorijsko!
- step_{1F} **mxMAR₁, IdMAR;**
- step₂₀ **IdMBR;**
- step₂₁ **mxMAR₁, mxMAR₀, IdMAR, br step₂₆;**
- ! Indirektno registarsko sa pomerajem !
- step₂₂ **IdRSRC;**
- step₂₃ **mxX₀, IdX, mxY₀, IdY;**
- step₂₄ **add, IdZ;**
- step₂₅ **mxMAR₂, IdMAR;**
- ! Čitanje operanda za memorijska adresiranja !
- step₂₆ *br (if **STORE** then step_{2A});*
- step₂₇ **IdMBR;**
- step₂₈ **mxB₀, IdB, br step_{2A};**
- ! Neposredno !
- step₂₉ **mxB₁, IdB;**

Sekvenca upravljačkih signala sa spajanjem (5)

- ! Izvršavanje operacije !
- step_{2A} *br (case (LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTI, RTS) then*
- (**LOAD**, step_{2B}), (**STORE**, step_{2C}),
- (**ADD**, step₃₂), (**AND**, step₃₅), (**ASR**, step₃₈),
- (**JZ**, step_{3B}), (**JMP**, step_{3F}), (**JSR**, step_{3D}), (**RTI**, step₄₀), (**RTS**, step₄₄));
- ! LOAD !
- step_{2B} **mxACC**, **ldACC**, *br step₄₈*;
- ! STORE !
- step_{2C} *br (if immed then step₄₈)*;
- step_{2D} *br (if regdir then step₃₀)*;
- step_{2E} **mxMBR₀**, **ldMBR**;
- step_{2F} **wrMEM**, *br step₄₈*;
- step₃₀ **ldRDST**;
- step₃₁ **wrGPR**, *br step₄₈*;

Sekvenca upravljačkih signala sa spajanjem (6)

- ! ADD !
- step₃₂ **ldX, ldY;**
- step₃₃ **add, ldZ;**
- step₃₄ **ldACC, br step₄₈;**
- ! AND !
- step₃₅ **ldX, ldY;**
- step₃₆ **and, ldZ;**
- step₃₇ **ldACC, br step₄₈;**
- ! ASR !
- step₃₈ **ldY;**
- step₃₉ **asr, ldZ;**
- step_{3A} **ldACC, br step₄₈;**
- ! JZ !
- step_{3B} *br (if Z then step_{3F});*
- step_{3C} *br step₄₈;*

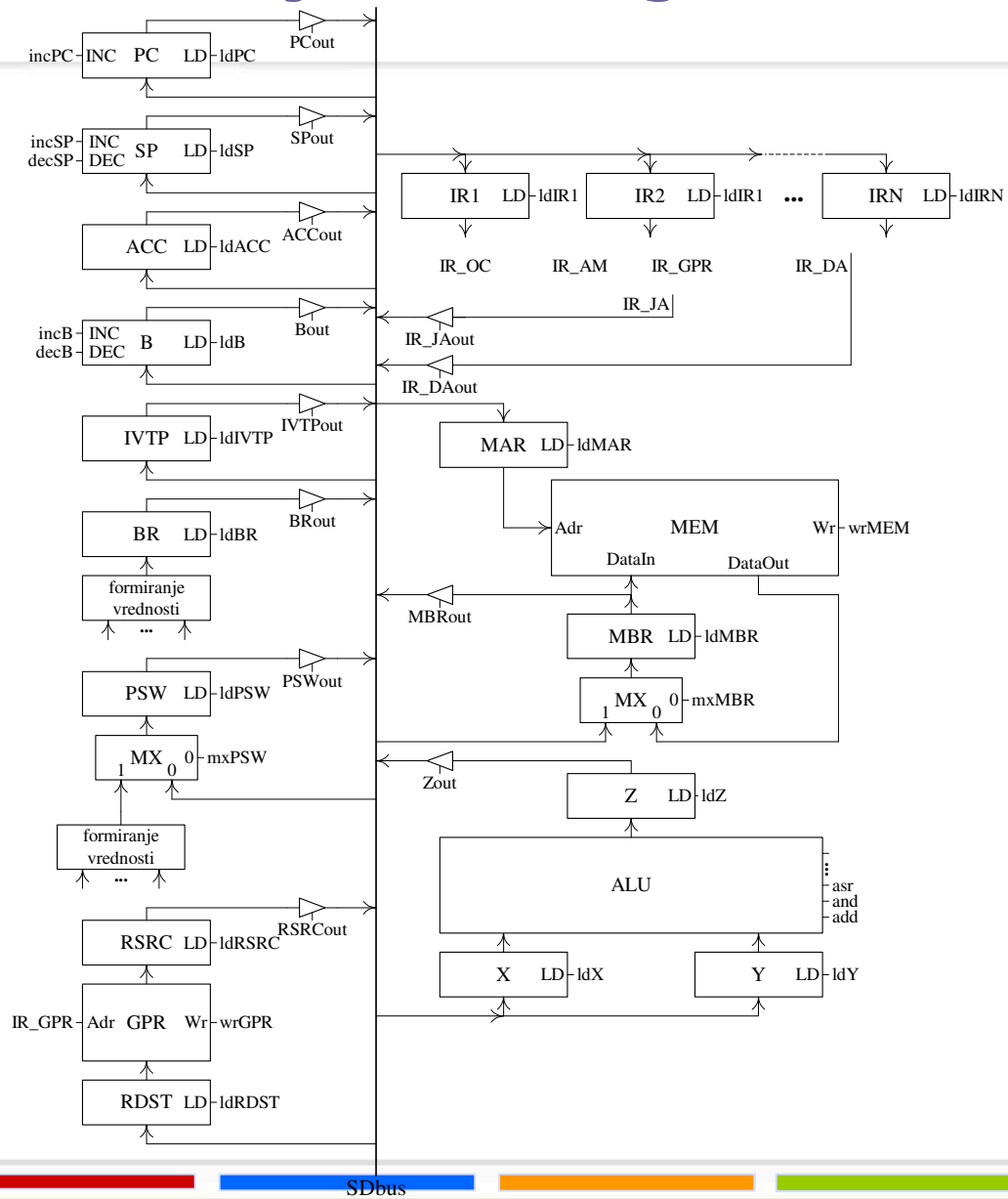
Sekvenca upravljačkih signala sa spajanjem (7)

- ! JSR !
- step_{3D} mxMAR₂, mxMAR₀, IdMAR, decSP, mxMBR₁, IdMBR;
- step_{3E} wrMEM;
- ! JMP !
- step_{3F} IdPC, *br* step₄₈;
- ! RTI !
- step₄₀ incSP;
- step₄₁ mxMAR₂, mxMAR₀, IdMAR;
- step₄₂ IdMBR;
- step₄₃ IdPSW;
- ! RTS !
- step₄₄ incSP;
- step₄₅ mxMAR₂, mxMAR₀, IdMAR;
- step₄₆ IdMBR;
- step₄₇ mxPC, IdPC;

Sekvenca upravljačkih signala sa spajanjem (8)

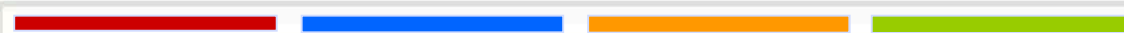
- ! Opsluživanje prekida !
- step₄₈ *br (if then step₀₀)*;
- step₄₉ **mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁, ldMBR;**
- step_{4A} **wrMEM;**
- step_{4B} **mxMAR₂, mxMAR₀, ldMAR, decSP, mxMBR₁,
mxMBR₀,ldMBR;**
- step_{4C} **wrMEM;**
- step_{4D} **mxX₁, ldX, mxY₁, ldY;**
- step_{4E} **add, ldZ;**
- step_{4F} **mxMAR₂, ldMAR;**
- step₅₀ **ldMBR;**
- step₅₁ **mxPC, ldPC, br step₀₀;**

Operaciona jedinica sa jednom magistralom



Sekvenca upravljačkih signala

- ! Čitanje instrukcije !
- step₀₀ **PCout, IdMAR, incPC;**
- step₀₁ **IdMBR;**
- step₀₂ **MBRout, IdIR1;**
- step₀₃ *br (if I1 then step₃₁);*
- step₀₄ **PCout, IdMAR, incPC;**
- step₀₅ **IdMBR;**
- step₀₆ **MBRout, IdIR2;**
- step₀₇ *br (if I2 then step_{0F});*
- step₀₈ **PCout, IdMAR, incPC;**
- ...
- step_{0E} **MBRout, IdIRN;**

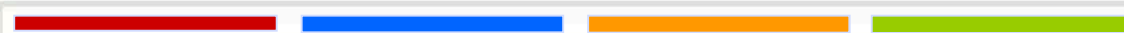


Sekvenca upravljačkih signala (2)

- ! Formiranje adrese i čitanje operanda !
- $step_{0F}$ *br* (*case* (**dirreg**, **indreg**, **postdec**, **preinc**,
dirmem, **indmem**, **indregpom**, **immed**) *then*
- (**dirreg**, $step_{10}$), (**indreg**, $step_{13}$), (**postdec**, $step_{16}$), (**preinc**,
 $step_{1C}$),
- (**dirmem**, $step_{22}$), (**indmem**, $step_{24}$), (**indregpom**, $step_{28}$),
(**immed**, $step_{30}$));
- ! Direktno registarsko !
- $step_{10}$ **IdRSRC**;
- $step_{11}$ **RSRCout**, **IdB**;
- $step_{12}$ *br* $step_{31}$;
- ! Indirektno registarsko !
- $step_{13}$ **IdRSRC**;
- $step_{14}$ **RSRCout**, **IdMAR**;
- $step_{15}$ *br* $step_{2C}$;

Sekvenca upravljačkih signala (3)

- ! Postdekrement !
- step₁₆ **IdRSRC;**
- step₁₇ **RSRCout, IdMAR, IdB;**
- step₁₈ **decB;**
- step₁₉ **Bout, IdRDST;**
- step_{1A} **wrGPR;**
- step_{1B} *br* step_{2C};
- ! Preinkrement !
- step_{1C} **IdRSRC;**
- step_{1D} **RSRCout, IdB;**
- step_{1E} **incB;**
- step_{1F} **Bout, IdMAR, IdRDST;**
- step₂₀ **wrGPR;**
- step₂₁ *br* step_{2C};

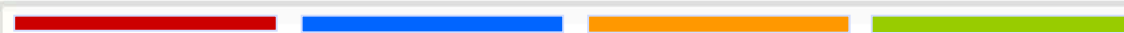


Sekvenca upravljačkih signala (4)

- ! Direktno memorijsko !
- step₂₂ **IR_DAout, IdMAR;**
- step₂₃ *br* step_{2C};
- ! Indirektno memorijsko!
- step₂₄ **IR_DAout, IdMAR;**
- step₂₅ **IdMBR;**
- step₂₆ **MBRout, IdMAR;**
- step₂₇ *br* step_{2C};
- ! Indirektno registarsko sa pomerajem !
- step₂₈ **IR_DAout, IdY, IdRSRC;**
- step₂₉ **RSRCout, IdX;**
- step_{2A} **add, IdZ;**
- step_{2B} **Zout, IdMAR;**

Sekvenca upravljačkih signala (5)

- ! Čitanje operanda za memorijska adresiranja !
- step_{2C} *br* (if **STORE** then step₃₁);
- step_{2D} **IdMBR**;
- step_{2E} **MBRout, IdB**;
- step_{2F} *br* step₃₁;
- ! Neposredno !
- step₃₀ **IR_DAout, IdB**;

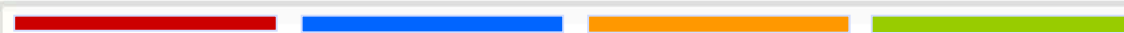


Sekvenca upravljačkih signala (6)

- ! Izvršavanje operacije !
- $step_{31}$ *br* (case (LOAD, STORE, ADD, AND, ASR, JZ, JMP, JSR, RTI, RTS) then
- (LOAD, $step_{32}$), (STORE, $step_{34}$),
- (ADD, $step_{3C}$), (AND, $step_{41}$), (ASR, $step_{46}$),
- (JZ, $step_{4A}$), (JMP, $step_{4F}$), (JSR, $step_{4C}$), (RTI, $step_{51}$), (RTS, $step_{55}$));
- ! LOAD !
- $step_{32}$ **Bout, IdACC;**
- $step_{33}$ *br* $step_{59}$;
- ! STORE !
- $step_{34}$ *br* (if **immed** then $step_{59}$);
- $step_{35}$ *br* (if **regdir** then $step_{39}$);
- $step_{36}$ **ACCout, IdMBR;**
- $step_{37}$ **wrMEM;**
- $step_{38}$ *br* $step_{59}$;
- $step_{39}$ **ACCout, IdRDST;**
- $step_{3A}$ **wrGPR;**
- $step_{3B}$ *br* $step_{59}$;

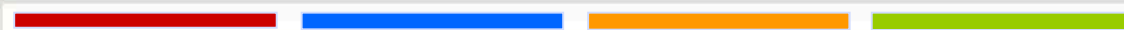
Sekvenca upravljačkih signala (7)

- ! ADD !
- step_{3C} **ACCout, IdX;**
- step_{3D} **Bout, IdY;**
- step_{3E} **add, IdZ;**
- step_{3F} **Zout, IdACC;**
- step₄₀ *br step₅₉;*
- ! AND !
- step₄₁ **ACCout, IdX;**
- step₄₂ **Bout, IdY;**
- step₄₃ **and, IdZ;**
- step₄₄ **Zout, IdACC;**
- step₄₅ *br step₅₉;*



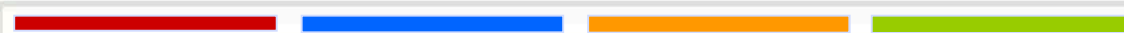
Sekvenca upravljačkih signala (8)

- ! ASR !
- step₄₆ **Bout, IdY;**
- step₄₇ **asr, IdZ;**
- step₄₈ **Zout, IdACC;**
- step₄₉ *br step₅₉;*
- ! JZ !
- step_{4A} *br (if Z then step_{4F});*
- step_{4B} *br step₅₉;*
- ! JSR !
- step_{4C} **SPout, IdMAR, decSP;**
- step_{4D} **PCout, IdMBR;**
- step_{4E} **wrMEM;**



Sekvenca upravljačkih signala (9)

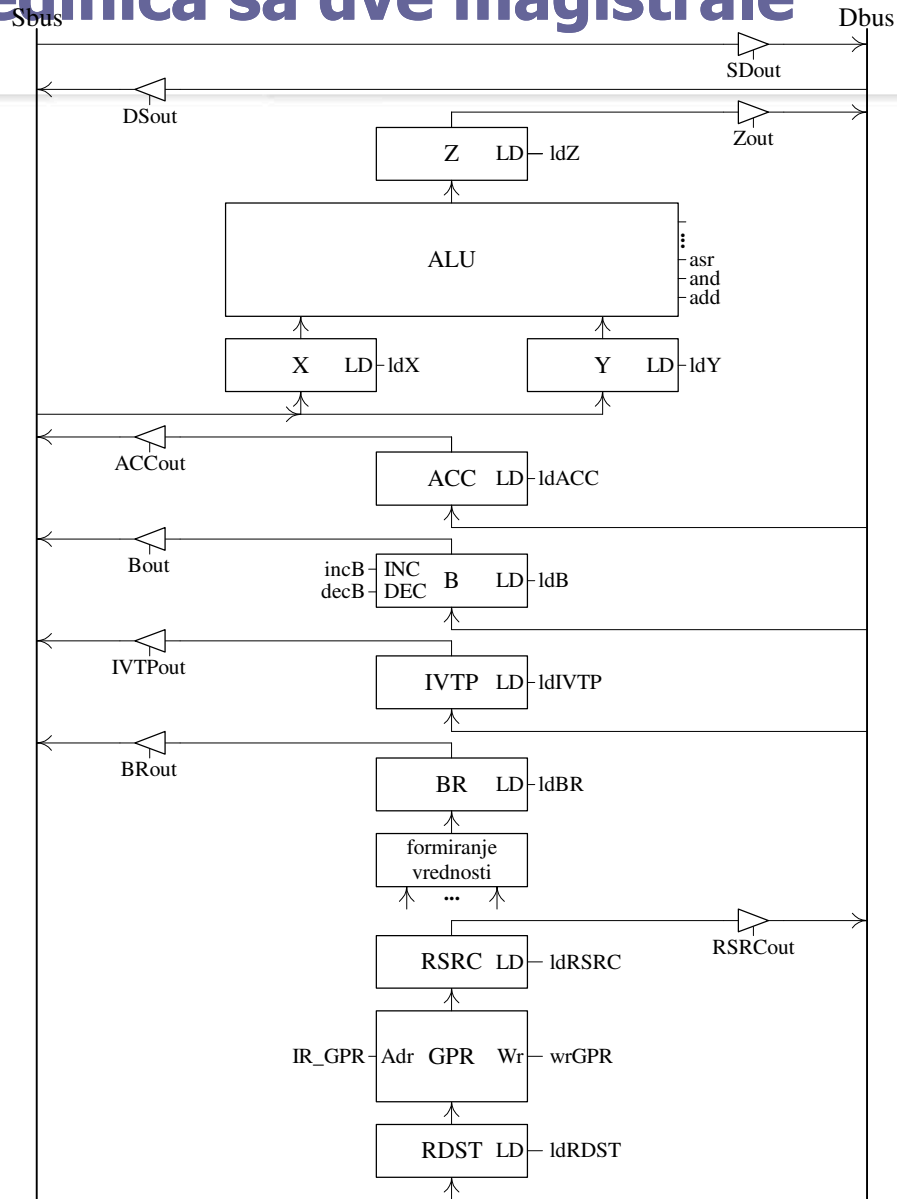
- ! JMP !
- step_{4F} **IR_JAout, IdPC;**
- step₅₀ *br* step₅₉;
- ! RTI !
- step₅₁ **incSP;**
- step₅₂ **SPout, IdMAR;**
- step₅₃ **IdMBR;**
- step₅₄ **MBRout, IdPSW;**
- ! RTS !
- step₅₅ **incSP;**
- step₅₆ **SPout, IdMAR;**
- step₅₇ **IdMBR;**
- step₅₈ **MBRout, IdPC;**



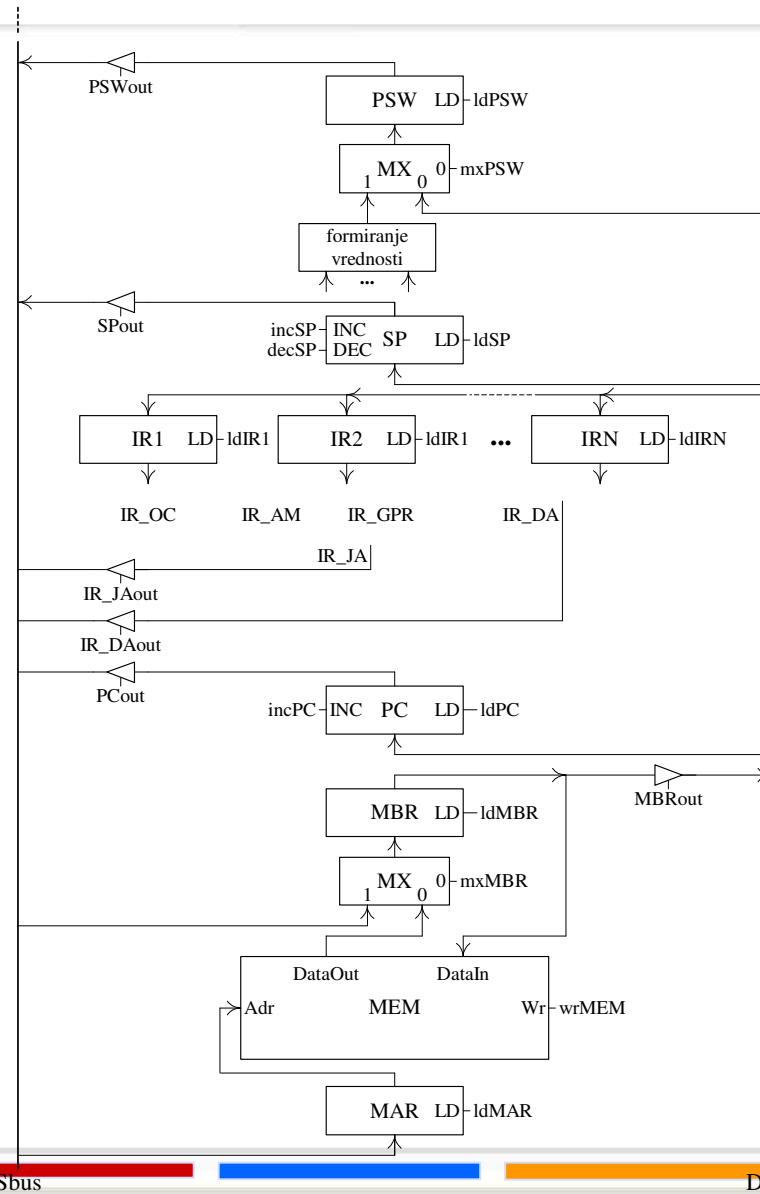
Sekvenca upravljačkih signala (10)

- ! Opsluživanje prekida !
- step₅₉ *br (if then step₀₀);*
- step_{5A} **SPout, IdMAR, decSP;**
- step_{5B} **PCout, IdMBR;**
- step_{5C} **wrMEM;**
- step_{5D} **SPout, IdMAR, decSP;**
- step_{5E} **PSWout, IdMBR;**
- step_{5F} **wrMEM;**
- step₆₀ **IVTPout, IdX;**
- step₆₁ **BRout, IdY;**
- step₆₂ **add, IdZ;**
- step₆₃ **Zout, IdMAR;**
- step₆₄ **IdMBR;**
- step₆₅ **MBRout, IdPC;**
- step₆₆ *br step₀₀;*

Operaciona jedinica sa dve magistrale

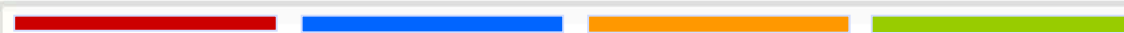


Operaciona jedinica sa dve magistrale (2)



Sekvenca upravljačkih signala

- ! Čitanje instrukcije !
- step₀₀ **PCout, IdMAR, incPC;**
- step₀₁ **IdMBR;**
- step₀₂ **MBRout, IdIR1, PCout, IdMAR;**
- step₀₃ *br (if I1 then step_{2E});*
- step₀₄ **IdMBR, incPC;**
- step₀₅ **MBRout, IdIR2, PCout, IdMAR;**
- step₀₆ *br (if I2 then step_{0C});*
- step₀₇ **IdMBR, incPC;**
- ...
- step_{0B} **MBRout, IdIRN;**

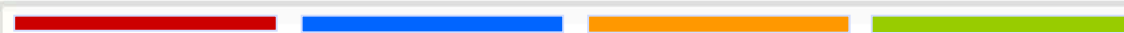


Sekvenca upravljačkih signala (2)

- ! Formiranje adrese i čitanje operanda !
- $step_{0C}$ *br* (*case* (**dirreg**, **indreg**, **postdec**, **preinc**,
dirmem, **indmem**, **indregpom**, **immed**) *then*
- (**dirreg**, $step_{0D}$), (**indreg**, $step_{10}$), (**postdec**, $step_{13}$), (**preinc**,
 $step_{19}$),
- (**dirmem**, $step_{1F}$), (**indmem**, $step_{21}$), (**indregpom**, $step_{25}$),
(**immed**, $step_{2D}$));
- ! Direktno registarsko !
- $step_{0D}$ **IdRSRC**;
- $step_{0E}$ **RSRCout**, **IdB**;
- $step_{0F}$ *br* $step_{2E}$;
- ! Indirektno registarsko !
- $step_{10}$ **IdRSRC**;
- $step_{11}$ **RSRCout**, **DSout**, **IdMAR**;
- $step_{12}$ *br* $step_{29}$;

Sekvenca upravljačkih signala (3)

- ! Postdekrement !
- step₁₃ **IdRSRC;**
- step₁₄ **RSRCout, DSout, IdMAR, IdB;**
- step₁₅ **decB;**
- step₁₆ **Bout, IdRDST;**
- step₁₇ **wrGPR;**
- step₁₈ *br step₂₉;*
- ! Preinkrement !
- step₁₉ **IdRSRC;**
- step_{1A} **RSRCout, IdB;**
- step_{1B} **incB;**
- step_{1C} **Bout, IdMAR, IdRDST;**
- step_{1D} **wrGPR;**
- step_{1E} *br step₂₉;*



Sekvenca upravljačkih signala (4)

- ! Direktno memorijsko !
- step_{1F} **IR_DAout, IdMAR;**
- step₂₀ *br* step₂₉;
- ! Indirektno memorijsko!
- step₂₁ **IR_DAout, IdMAR;**
- step₂₂ **IdMBR;**
- step₂₃ **MBRout, DSout, IdMAR;**
- step₂₄ *br* step₂₉;
- ! Indirektno registarsko sa pomerajem !
- step₂₅ **IR_DAout, IdY, IdRSRC;**
- step₂₆ **RSRCout, DSout, IdX;**
- step₂₇ **add, IdZ;**
- step₂₈ **Zout, DSout, IdMAR;**

Sekvenca upravljačkih signala (5)

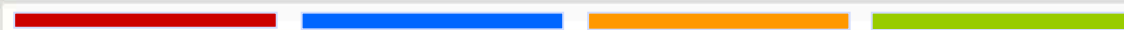
- ! Čitanje operanda za memorijska adresiranja !
- step₂₉ *br* (if **STORE** then step_{2E});
- step_{2A} **ldMBR**;
- step_{2B} **MBRout, ldB**;
- step_{2C} *br* step_{2E};
- ! Neposredno !
- step_{2D} **IR_DAout, SDout, ldB**;

Sekvenca upravljačkih signala (6)

- ! Izvršavanje operacije !
- $step_{2E}$ *br* (*case* (**LOAD**, **STORE**, **ADD**, **AND**, **ASR**, **JZ**, **JMP**, **JSR**, **RTI**, **RTS**) *then*
- (**LOAD**, $step_{2F}$), (**STORE**, $step_{31}$),
- (**ADD**, $step_{39}$), (**AND**, $step_{3E}$), (**ASR**, $step_{43}$),
- (**JZ**, $step_{47}$), (**JMP**, $step_{4C}$), (**JSR**, $step_{49}$), (**RTI**, $step_{4E}$), (**RTS**, $step_{52}$));
- ! **LOAD** !
- $step_{2F}$ **Bout**, **SDout**, **IdACC**;
- $step_{30}$ *br* $step_{56}$;
- ! **STORE** !
- $step_{31}$ *br* (*if immed then* $step_{56}$);
- $step_{32}$ *br* (*if regdir then* $step_{36}$);
- $step_{33}$ **ACCout**, **mxMBR**, **IdMBR**;
- $step_{34}$ **wrMEM**;
- $step_{35}$ *br* $step_{56}$;
- $step_{36}$ **ACCout**, **IdRDST**;
- $step_{37}$ **wrGPR**;
- $step_{38}$ *br* $step_{56}$;

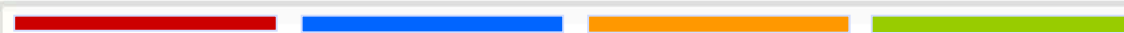
Sekvenca upravljačkih signala (7)

- ! ADD !
- step₃₉ **ACCout, IdX;**
- step_{3A} **Bout, IdY;**
- step_{3B} **add, IdZ;**
- step_{3C} **Zout, IdACC;**
- step_{3D} *br step₅₆;*
- ! AND !
- step_{3E} **ACCout, IdX;**
- step_{3F} **Bout, IdY;**
- step₄₀ **and, IdZ;**
- step₄₁ **Zout, IdACC;**
- step₄₂ *br step₅₆;*



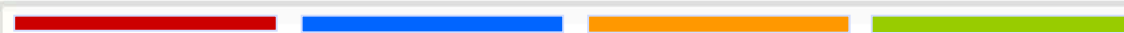
Sekvenca upravljačkih signala (8)

- ! ASR !
- step₄₃ **Bout, IdY;**
- step₄₄ **asr, IdZ;**
- step₄₅ **Zout, IdACC;**
- step₄₆ *br step₅₆;*
- ! JZ !
- step₄₇ *br (if **eq** then step_{4C});*
- step₄₈ *br step₅₆;*
- ! JSR !
- step₄₉ **SPout, IdMAR, decSP;**
- step_{4A} **PCout, mxMBR, IdMBR;**
- step_{4B} **wrMEM;**



Sekvenca upravljačkih signala (9)

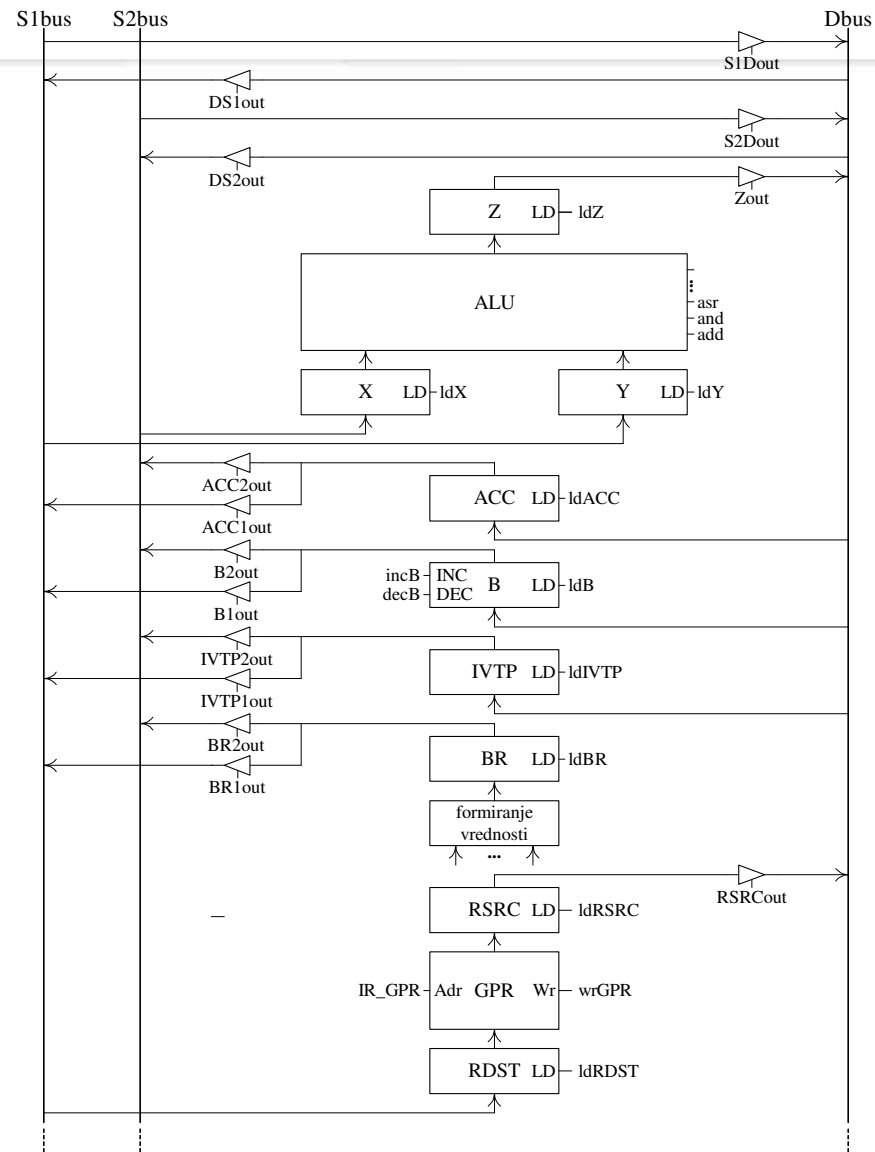
- ! JMP !
- step_{4C} **IR_JAout, SDout, IdPC;**
- step_{4D} *br* step₅₆;
- ! RTI !
- step_{4E} **incSP;**
- step_{4F} **SPout, IdMAR;**
- step₅₀ **IdMBR;**
- step₅₁ **MBRout, IdPSW;**
- ! RTS !
- step₅₂ **incSP;**
- step₅₃ **SPout, IdMAR;**
- step₅₄ **IdMBR;**
- step₅₅ **MBRout, IdPC;**



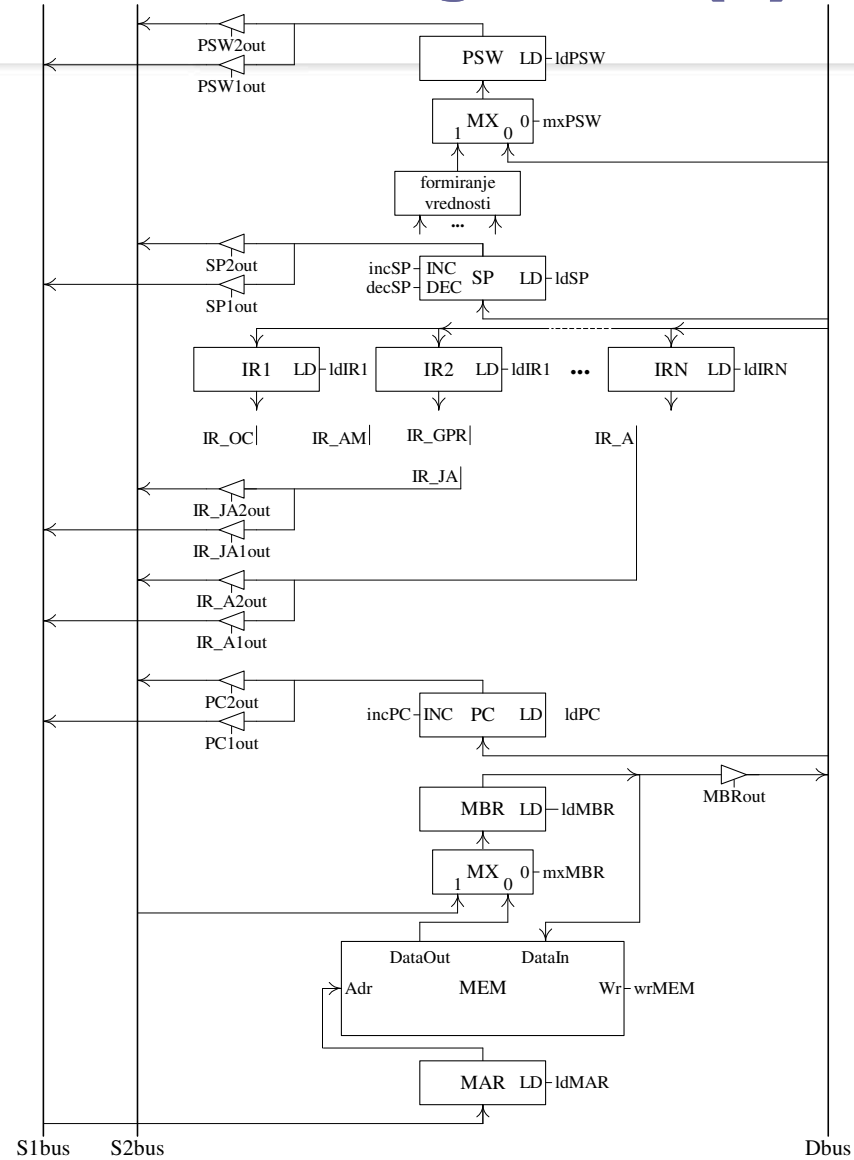
Sekvenca upravljačkih signala (10)

- ! Opsluživanje prekida !
- step₅₆ *br (if then step₀₀);*
- step₅₇ **SPout, IdMAR, decSP;**
- step₅₈ **PCout, mxMBR, IdMBR;**
- step₅₉ **wrMEM;**
- step_{5A} **SPout, IdMAR, decSP;**
- step_{5B} **PSWout, mxMBR, IdMBR;**
- step_{5C} **wrMEM;**
- step_{5D} **IVTPout, IdX;**
- step_{5E} **BRout, IdY;**
- step_{5F} **add, IdZ;**
- step₆₀ **Zout, DSout, IdMAR;**
- step₆₁ **IdMBR;**
- step₆₂ **MBRout, IdPC;**
- step₆₃ *br step₀₀;*

Operaciona jedinica sa tri magistrale

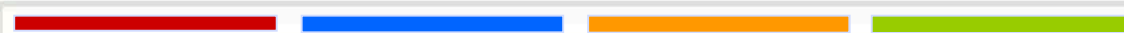


Operaciona jedinica sa tri magistrale (2)



Sekvenca upravljačkih signala

- ! Čitanje instrukcije !
- step₀₀ **PC1out, IdMAR, incPC;**
- step₀₁ **IdMBR;**
- step₀₂ **MBRout, IdIR1, PC1out, IdMAR;**
- step₀₃ *br (if I1 then step_{2E});*
- step₀₄ **IdMBR, incPC;**
- step₀₅ **MBRout, IdIR2, PC1out, IdMAR;**
- step₀₆ *br (if I2 then step_{0C});*
- step₀₇ **IdMBR, incPC;**
- ...
- step_{0B} **MBRout, IdIRN;**

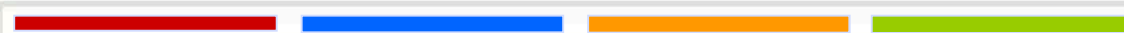


Sekvenca upravljačkih signala (2)

- ! Formiranje adrese i čitanje operanda !
- $step_{0C}$ *br* (*case* (**dirreg**, **indreg**, **postdec**, **preinc**,
dirmem, **indmem**, **indregpom**, **immed**) *then*
- (**dirreg**, $step_{0D}$), (**indreg**, $step_{10}$), (**postdec**, $step_{13}$), (**preinc**,
 $step_{19}$),
- (**dirmem**, $step_{1F}$), (**indmem**, $step_{21}$), (**indregpom**, $step_{25}$),
(**immed**, $step_{2D}$));
- ! Direktno registarsko !
- $step_{0D}$ **IdRSRC**;
- $step_{0E}$ **RSRCout**, **IdB**;
- $step_{0F}$ *br* $step_{2E}$;
- ! Indirektno registarsko !
- $step_{10}$ **IdRSRC**;
- $step_{11}$ **RSRCout**, **DS1out**, **IdMAR**;
- $step_{12}$ *br* $step_{29}$;

Sekvenca upravljačkih signala (3)

- ! Postdekrement !
- step₁₃ **IdRSRC;**
- step₁₄ **RSRCout, DS1out, IdMAR, IdB;**
- step₁₅ **decB;**
- step₁₆ **B1out, IdRDST;**
- step₁₇ **wrGPR;**
- step₁₈ *br step₂₉;*
- ! Preinkrement !
- step₁₉ **IdRSRC;**
- step_{1A} **RSRCout, IdB;**
- step_{1B} **incB;**
- step_{1C} **B1out, IdMAR, IdRDST;**
- step_{1D} **wrGPR;**
- step_{1E} *br step₂₉;*



Sekvenca upravljačkih signala (4)

- ! Direktno memorijsko !
- step_{1F} **IR_DA1out, IdMAR;**
- step₂₀ *br* step₂₉;
- ! Indirektno memorijsko!
- step₂₁ **IR_DA1out, IdMAR;**
- step₂₂ **IdMBR;**
- step₂₃ **MBRout, DS1out, IdMAR;**
- step₂₄ *br* step₂₉;
- ! Indirektno registarsko sa pomerajem !
- step₂₅ **IdRSRC;**
- step₂₆ **RSRCout, DS2out, IdX, IR_DA1out, IdY;**
- step₂₇ **add, IdZ;**
- step₂₈ **Zout, DS1out, IdMAR;**

Sekvenca upravljačkih signala (5)

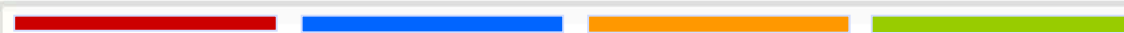
- ! Čitanje operanda za memorijska adresiranja !
- step₂₉ *br (if **STORE** then step_{2E});*
- step_{2A} **ldMBR;**
- step_{2B} **MBRout, ldB;**
- step_{2C} *br step_{2E};*
- ! Neposredno !
- step_{2D} **IR_DA1out, S1Dout, ldB;**

Sekvenca upravljačkih signala (6)

- ! Izvršavanje operacije !
- $step_{2E}$ *br* (*case* (**LOAD**, **STORE**, **ADD**, **AND**, **ASR**, **JZ**, **JMP**, **JSR**, **RTI**, **RTS**) *then*
- (**LOAD**, $step_{2F}$), (**STORE**, $step_{31}$),
- (**ADD**, $step_{39}$), (**AND**, $step_{3D}$), (**ASR**, $step_{41}$),
- (**JZ**, $step_{45}$), (**JMP**, $step_{49}$), (**JSR**, $step_{47}$), (**RTI**, $step_{4B}$), (**RTS**, $step_{4F}$));
- ! **LOAD** !
- $step_{2F}$ **B1out**, **S1Dout**, **ldACC**;
- $step_{30}$ *br* $step_{53}$;
- ! **STORE** !
- $step_{31}$ *br* (*if immed then* $step_{53}$);
- $step_{32}$ *br* (*if regdir then* $step_{36}$);
- $step_{33}$ **ACC2out**, **mxMBR**, **ldMBR**;
- $step_{34}$ **wrMEM**;
- $step_{35}$ *br* $step_{53}$;
- $step_{36}$ **ACC1out**, **ldRDST**;
- $step_{37}$ **wrGPR**;
- $step_{38}$ *br* $step_{53}$;

Sekvenca upravljačkih signala (7)

- ! ADD !
- step₃₉ **ACC2out, IdX, B1out, IdY;**
- step_{3A} **add, IdZ;**
- step_{3B} **Zout, IdACC;**
- step_{3C} *br step₅₃;*
- ! AND !
- step_{3D} **ACC2out, IdX, B1out, IdY;**
- step_{3E} **and, IdZ;**
- step_{3F} **Zout, IdACC;**
- step₄₀ *br step₅₃;*
- ! ASR !
- step₄₁ **B1out, IdY;**
- step₄₂ **asr, IdZ;**
- step₄₃ **Zout, IdACC;**
- step₄₄ *br step₅₃;*

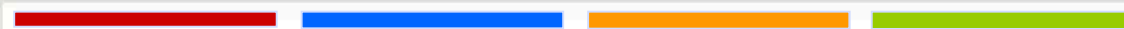


Sekvenca upravljačkih signala (8)

- ! JZ !
- step₄₅ *br (if eql then step₄₉);*
- step₄₆ *br step₅₃;*
- ! JSR !
- step₄₇ **SP1out, IdMAR, decSP, PC2out,**
mxMBR, IdMBR;
- step₄₈ **wrMEM;**
- ! JMP !
- step₄₉ **IR_JA1out, S1Dout, IdPC;**
- step_{4A} *br step₅₃;*

Sekvenca upravljačkih signala (9)

- ! RTI !
- step_{4B} **incSP;**
- step_{4C} **SP1out, IdMAR;**
- step_{4D} **IdMBR;**
- step_{4E} **MBRout, IdPSW;**
- ! RTS !
- step_{4F} **incSP;**
- step₅₀ **SP1out, IdMAR;**
- step₅₁ **IdMBR;**
- step₅₂ **MBRout, IdPC;**



Sekvenca upravljačkih signala (10)

- ! Opsluživanje prekida !
- step₅₃ *br (if then step₀₀)*;
- step₅₄ **SP1out, IdMAR, decSP, PC2out, mxMBR,**
IdMBR;
- step₅₅ **wrMEM;**
- step₅₆ **SP1out, IdMAR, decSP, PSW2out, mxMBR,**
IdMBR;
- step₅₇ **wrMEM;**
- step₅₈ **IVTP2out, IdX, BR1out, IdY;**
- step₅₉ **add, IdZ;**
- step_{5A} **Zout, DS1out, IdMAR;**
- step_{5B} **IdMBR;**
- step_{5C} **MBRout, IdPC;**
- step_{5D} *br step₀₀*;

Питања?

Електротехнички Факултет
Универзитет у Београду

