

# Instrukcijski set



## TEME

- ✓ Aritmetičke instrukcije
- ✓ Logičke instrukcije
- ✓ Pomeračke instrukcije
- ✓ Instrukcije prenosa
- ✓ Instrukcije skoka

# Uvod

**Instrukcijski set** je skup instrukcija kojima se specificiraju operacije koje procesor može da izvrši.

- ❑ Izbor instrukcija zavisi od namene računara.
- ❑ Procesori opšte namene uglavnom podržavaju standardne instrukcije, dok procesori specijalne namene rade i sa nestandardnim instrukcijama

Primeri nestandardnih instrukcija:

- instrukcije nad celobrojnim veličinama promenljive dužine
- string instrukcije...

# Standardni instrukcijski set

---

- ✓ Aritmetičke instrukcije
- ✓ Logičke instrukcije
- ✓ Pomeračke instrukcije
- ✓ Instrukcije prenosa
- ✓ Instrukcije skoka



# Aritmetičke instrukcije

---

Realizuju standardne aritmetičke operacije:

- sabiranje
- oduzimanje
- množenje
- deljenje



# Instrukcije sabiranja

Kod operacije: **ADD** (sabiranje)

<b>ADD</b> <i>a, b, c</i>	3-adr.	<i>a</i> i <i>b</i> izvor.	<i>c</i> odr.	<i>c</i> nije neposr.
<b>ADD</b> <i>a, b</i>	2-adr.	<i>a</i> i <i>b</i> izvor.	<i>a</i> (ili <i>b</i> ) odr.	<i>a</i> nije neposr.
<b>ADD</b> <i>a</i>	1-adr.	ACC izvor.	<i>a</i> izvor.	ACC odr.
<b>ADD</b>	0-adr.	SP izvor.	SP izvor.	SP odr.

Kod operacije: **INC** (inkrementiranje = sabiranje sa 1)

<b>INC</b> <i>a, b</i>	2-adr.	<i>a</i> izvor.	<i>b</i> odr.	<i>b</i> nije neposr.
<b>INC</b> <i>a</i>	1-adr.	<i>a</i> izvor.	<i>a</i> odr.	<i>a</i> nije neposr.
<b>INC</b>	0-adr.	SP izvor.	SP odr.	

Legenda

*n*-adr. = *n*-adresna instrukcija  
izvor. = izvorišni operand  
odr. = odredišni operand

ACC – akumulator  
SP – vrh steka  
neposr. – neposredno zadat u instrukciji

# Instrukcije oduzimanja

Kod operacije: **SUB** (oduzimanje)

<b>SUB</b> <i>a, b, c</i>	3-adr.	<i>a</i> i <i>b</i> izvor.	<i>c</i> odr.	<i>c</i> nije neposr.
<b>SUB</b> <i>a, b</i>	2-adr.	<i>a</i> i <i>b</i> izvor.	<i>a</i> (ili <i>b</i> ) odr.	<i>a</i> nije neposr.
<b>SUB</b> <i>a</i>	1-adr.	ACC izvor.	<i>a</i> izvor.	ACC odr.
<b>SUB</b>	0-adr.	SP izvor.	SP izvor.	SP odr.

Kod operacije: **DEC** (dekrementiranje = oduzima se 1)

<b>DEC</b> <i>a, b</i>	2-adr.	<i>a</i> izvor.	<i>b</i> odr.	<i>b</i> nije neposr.
<b>DEC</b> <i>a</i>	1-adr.	<i>a</i> izvor.	<i>a</i> odr.	<i>a</i> nije neposr.
<b>DEC</b>	0-adr.	SP izvor.	SP odr.	

Legenda

*n*-adr. = *n*-adresna instrukcija  
izvor. = izvorišni operand  
odr. = odredišni operand

ACC – akumulator  
SP – vrh steka  
neposr. – neposredno zadat u instrukciji

# Instrukcije oduzimanja

Kod operacije: **CMP** (aritmetičko upoređivanje)

<b>CMP</b> <i>a, b, c</i>	3-adr.	<i>a</i> i <i>b</i> izvor.	<i>c</i> se ne koristi
<b>CMP</b> <i>a, b</i>	2-adr.	<i>a</i> i <i>b</i> izvor.	
<b>CMP</b> <i>a</i>	1-adr.	ACC izvor.	<i>a</i> izvor.
<b>CMP</b>	0-adr.	SP izvor.	SP izvor.

Izvršavanje **CMP** instrukcije:

- od prvog izvorišnog operanda se oduzima drugi i dobija rezultat
- rezultat operacije se nigde ne upisuje, već se samo vrši njegova provera i postavljaju indikatori u *PSW* registru

Indikatori dobijaju vrednost 1 ako je:

**N** (rezultat negativan)

**Z** (rezultat nula)

**C** (bilo pozajmice)

**V** (perkoračen opseg)

# Instrukcije oduzimanja

Kod operacije: **CMP** (aritmetičko upoređivanje)

- ❑ Svrha postavljanja flegova je da se omogući rad mehanizma za realizaciju uslovnih skokova u programu.
- ❑ Instrukcija uslovnog skoka proverava flegove i utvrđuje da li je uslov za skok ispunjen ili nije.
- ❑ Ako je uslov ispunjen, ostvaruje se skok, a ako nije, nastavlja se sa sekvencijalnim izvršavanjem instrukcija.

Instrukcija	Relacija	Uslov
BEQL	=	Z = 1
BGRTU	>	C v Z = 0
BLSSU	<	C = 1

Instrukcija	Relacija	Uslov
BNEQ	≠	Z = 0
BGREU	≥	C = 0
BLEQU	≤	C v Z = 1



# Instrukcije množenja i deljenja

Kod operacije: **MUL** (množenje)

**MUL**  $a, b, c$

**MUL**  $a, b$

**MUL**  $a$

**MUL**

Kod operacije: **DIV** (deljenje)

**DIV**  $a, b, c$

**DIV**  $a, b$

**DIV**  $a$

**DIV**

- ☐ Eksplicitna i implicitna specifikacija izvorišnih i odredišnih operanada je ista kao kod instrukcije sabiranja.



# Logičke instrukcije

Realizuju standardne logičke operacije: I, ILI, ekskluzivno ILI i NE.

Kod operacije: **AND**

**AND**  $a, b, c$

**AND**  $a, b$

**AND**  $a$

**AND**

Kod operacije: **OR**

**OR**  $a, b, c$

**OR**  $a, b$

**OR**  $a$

**OR**

Kod operacije: **EXOR**

**EXOR**  $a, b, c$

**EXOR**  $a, b$

**EXOR**  $a$

**EXOR**

Kod operacije: **NOT**

**NOT**  $a, b$

**NOT**  $a$

**NOT**

# Logičke instrukcije

Kod operacije: **TST** (logičko upoređivanje)

<b>TST</b> <i>a, b, c</i>	3-adr.	<i>a</i> i <i>b</i> izvor.	<i>c</i> se ne koristi
<b>TST</b> <i>a, b</i>	2-adr.	<i>a</i> i <i>b</i> izvor.	
<b>TST</b> <i>a</i>	1-adr.	ACC izvor.	<i>a</i> izvor.
<b>TST</b>	0-adr.	SP izvor.	SP izvor.

Izvršavanje **TST** instrukcije:

- izvršava se / operacija nad operandima i dobija rezultat
- rezultat operacije se nigde ne upisuje, već se samo vrši njegova provera i postavljaju indikatori u *PSW* registru

Indikatori se postavljaju na sledeći način:

**N** (najstariji bit rezultata)

**Z** (1 ako je rezultat nula)

**C** (fiksna vrednost 0)

**V** (fiksna vrednost 0)

# Logičke instrukcije

Kod operacije: TST (logičko upoređivanje)

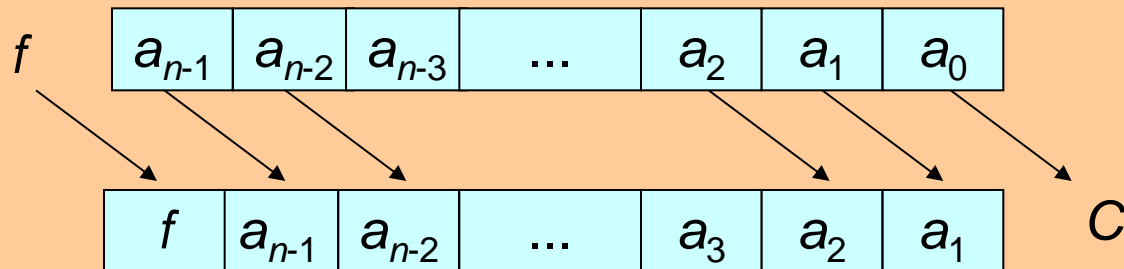
- ❑ Svrha postavljanja flegova je da se omogući rad mehanizma za realizaciju uslovnih skokova u programu.
- ❑ Od instrukcija uslovnog skoka, imaju smisla samo: jednako i nejednako (rezultat je 0 ili različit od 0).

Relacija	Uslov
=	Z = 1
≠	Z = 0

# Pomeračke instrukcije

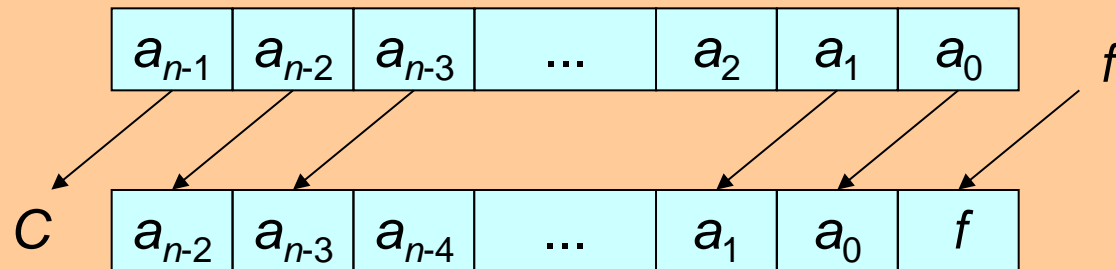
Realizuju pomeranje binarne reči za jedno mesto ulevo ili udesno.

*pomeranje udesno*



$f$	Instrukcija
$a_{n-1}$	aritmetičko pomer.
0	logičko pomeranje
$a_0$	rotiranje
$C$	rotiranje kroz ind. $C$

*pomeranje ulevo*



$f$	Instrukcija
0	aritmetičko pomer.
0	logičko pomeranje
$a_{n-1}$	rotiranje
$C$	rotiranje kroz ind. $C$

Napomena:  $C$  je indikator PSW registra.

# Pomeračke instrukcije

Kod operacije: **ASR** (aritmetičko pomeranje udesno)

<b>ASR a, b, c</b>	3-adr.	a izvor.	b odr.	c se ne koristi
<b>ASR a, b</b>	2-adr.	a izvor.	b odr.	
<b>ASR a</b>	1-adr.	a izvor.	a odr.	
<b>ASR</b>	0-adr.	SP izvor.	SP odr.	

Izvršavanje **ASR** instrukcije: izvorišni operand se pomeri udesno i smesti na odredište

Na osnovu dobijenog rezultata postavljaju se indikator *PSW* registra:

**N** (najstariji bit rezultata)

**Z** (1 ako je rezultat 0)

**C** (najniži bit izvorišnog operanda)

**V** (fiksna vrednost 0)

# Pomeračke instrukcije

Kod operacije: LSR (logičko pomeranje udesno)

LSR  $a, b, c$

LSR  $a, b$

LSR  $a$

LSR

Kod operacije: ROR (rotacija udesno)

ROR  $a, b, c$

ROR  $a, b$

ROR  $a$

ROR

Kod operacije:

RORC (rotacija udesno kroz ind. C)

RORC  $a, b, c$

RORC  $a, b$

RORC  $a$

RORC

# Pomeračke instrukcije

Kod operacije: **ASL** (aritmetičko pomeranje ulevo)

<b>ASL a, b, c</b>	3-adr.	a izvor.	b odr.	c se ne koristi
<b>ASL a, b</b>	2-adr.	a izvor.	b odr.	
<b>ASL a</b>	1-adr.	a izvor.	a odr.	
<b>ASL</b>	0-adr.	SP izvor.	SP odr.	

Izvršavanje **ASL** instrukcije: izvorišni operand se pomeri ulevo i smesti na odredište

Na osnovu dobijenog rezultata postavljaju se indikator *PSW* registra:

**N** (najstariji bit rezultata)

**Z** (1 ako je rezultat 0)

**C** (najviši bit izvorišnog operanda)

**V** (fiksna vrednost 0)



# Pomeračke instrukcije

Kod operacije: LSL (logičko pomeranje ulevo)

LSL *a, b, c*

LSL *a, b*

LSL *a*

LSL

Kod operacije: ROL (rotacija ulevo)

ROL *a, b, c*

ROL *a, b*

ROL *a*

ROL

Kod operacije:

ROLC (rotacija ulevo kroz ind. C)

ROLC *a, b, c*

ROLC *a, b*

ROLC *a*

ROLC

# Pomeračke instrukcije

## Primer 1

R1

1	0	0	1	1	0	1	0
---	---	---	---	---	---	---	---

(R1) = 9A

PSW

				1			
--	--	--	--	---	--	--	--

C = 1

ASR R1

R1

1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

(R1) = CD

ROR R1

R1

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

(R1) = 4D

LSL R1

R1

0	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

(R1) = 34

ROL R1

R1

0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

(R1) = 35

# Instrukcije prenosa

Realizuju prenos podataka sa jednog mesta u računaru na drugo.

Podaci se mogu naći na različitim mestima:

- ✓ u memorijskim lokacijama (*ML*)
- ✓ u procesorskim registrima (*PR*)
- ✓ u instrukciji (neposredno adresiranje) (*NA*)
- ✓ u registrima kontrolera periferija
  - registri imaju isti tretman kao memorijske lokacije (*RKP*)  
(U/I adresni prostor je memorijski preslikan)
  - registri se posebno tretiraju (*RKR*)  
(U/I i memorijski adresni prostori su razdvojeni)
- ✓ u akumulatoru (kod 1-adresnih instrukcija) (*AK*)
- ✓ na steku (kod 0-adresnih instrukcija) (*ST*)

# Instrukcije prenosa

---

Instrukcija: **MOV** *a, b*

*a* izvor. (*ML, PR, RKP, NA*)

*b* odr. (*ML, PR, RKP*)

Izvršavanje instrukcije: izvorišni operand se prenosi na odredište



# Instrukcije prenosa

Instrukcije: kod 2-adr. procesora sa razdvojenim U/I i M adr.prostorima

IN *regper, regproc*

OUT *regproc, regper*

prenos *RKR* → *PR*

prenos *PR* → *RKR*

adresa registra kontrolera periferije

adresa procesorskog registra

Instrukcije: kod 1-adr. procesora sa razdvojenim U/I i M adr.prostorima

IN *regper*

OUT *regper*

prenos *RKR* → *AK*

prenos *AK* → *RKR*

# Instrukcije prenosa

Instrukcije: kod 1-adr. procesora

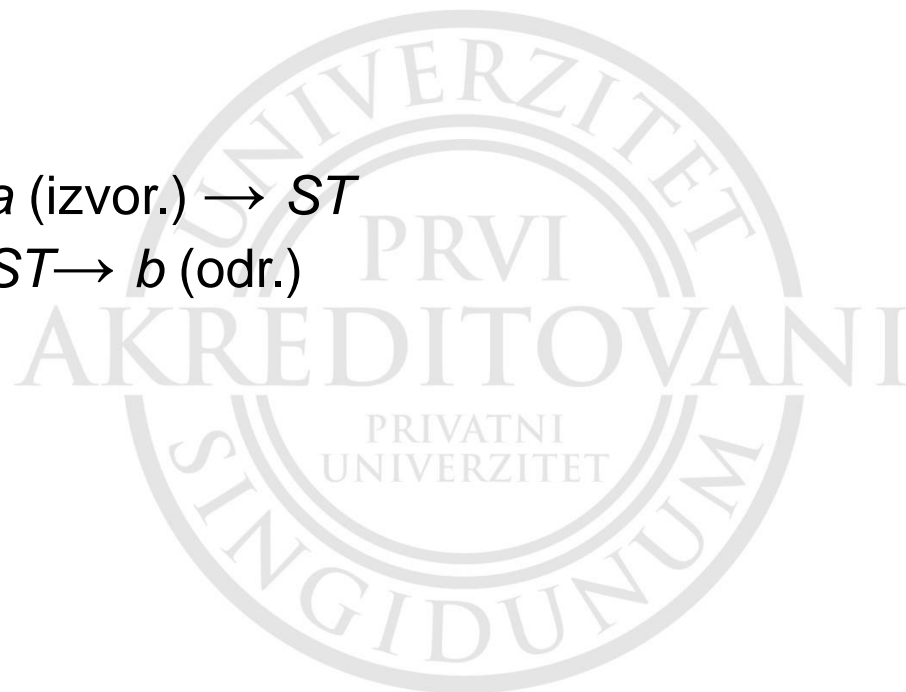
<b>LOAD</b> <i>a</i>	prenos <i>a</i> (izvor.) $\rightarrow$ <i>AK</i>
<b>STORE</b> <i>b</i>	prenos <i>AK</i> $\rightarrow$ <i>b</i> (odr.)

Instrukcije: kod 0-adr. procesora

<b>PUSH</b> <i>a</i>	prenos <i>a</i> (izvor.) $\rightarrow$ <i>ST</i>
<b>POP</b> <i>b</i>	prenos <i>ST</i> $\rightarrow$ <i>b</i> (odr.)

Za sve navedene instrukcije važi:

<i>a</i>	( <i>ML</i> , <i>PR</i> , <i>RKP</i> , <i>NA</i> )
<i>b</i>	( <i>ML</i> , <i>PR</i> , <i>RKP</i> )



# Instrukcije skoka

---

- ❑ Program se izvršava sekvencijalno ukoliko u njemu nema instrukcija skoka.
- ❑ Kada treba odstupiti od sekvencijalnosti, koristi se instrukcija skoka.
- ❑ Instrukcija skoka modifikuje sadržaj programskog brojača (*PC*).

## Klasifikacija instrukcija skoka

- ❑ instrukcije bezuslovnog skoka
- ❑ instrukcije uslovnog skoka
- ❑ instrukcije skoka na potprogram (sa povratkom)
- ❑ instrukcije skoka na prekidnu rutinu (sa povratkom)

# Instrukcije bezuslovnog skoka

Instrukcija:      **JMP** *adr*

- ❑ *adr* je adresa mem.lok. u kojoj je instrukcija na čije izvršavanje treba preći (koristi se kada je u vreme kompajliranja ova adresa poznata)

Izvršavanje instrukcije: vrednost *adr* se upisuje u *PC*

Instrukcija:      **JMPIND** *a*

- ❑ *a* je polje sa specifikacijom adrese memorijske lokacije u kojoj se nalazi instrukcija na čije izvršavanje treba preći (koristi se kada u vreme kompajliranja nije poznata adresa instrukcije na koju treba skočiti, već se ona računa tokom izvršavanja programa)

Izvršavanje instrukcije: računa se adresa i rezultat upisuje u *PC*



# Instrukcije uslovnog skoka

Opšti oblik instrukcije uslovnog skoka je: **KO** *p*

- ☐ **KO** je kod operacije instrukcije uslovnog skoka.
- ☐ *p* je pomeraj sa kojim treba napraviti relativan skok u odnosu na tekući sadržaj *PC*, ukoliko je uslov za skok ispunjen.
- ☐ Pomeraj *p* je u drugom komplementu, pa skok može biti unapred ili unazad.
- ☐ Uslov za skok se specificira kodom operacije, a proverava osnovu vrednosti indikatora *PSW* registra (*N*, *Z*, *C* i *V*).

Izvršavanje instrukcije:

- sadržaj *PC* se sabira sa *p* i izračunava adresa memorijske lokacije
- na osnovu indikatora *PSW*, proverava se uslov za skok
- ako je uslov ispunjen, izračunata adresa se upisuje u *PC*

# Instrukcije uslovnog skoka

Instrukcija	Opis	Uslov
<b>BEQL</b> <i>p</i>	skok na jednako	$Z = 1$
<b>BNEQ</b> <i>p</i>	skok na nejednako	$Z = 0$
<b>BGRTU</b> <i>p</i>	skok na veće od (bez znaka)	$C \vee Z = 0$
<b>BGREU</b> <i>p</i>	skok na veće od ili jednako (bez znaka)	$C = 0$
<b>BLSSU</b> <i>p</i>	skok na manje od (bez znaka)	$C = 1$
<b>BLEQU</b> <i>p</i>	skok na manje od ili jednako (bez znaka)	$C \vee Z = 1$
<b>BGRT</b> <i>p</i>	skok na veće od (sa znakom)	$(N \oplus V) \vee Z = 0$
<b>BGRE</b> <i>p</i>	skok na veće od ili jednako (sa znakom)	$N \oplus V = 0$
<b>BLSS</b> <i>p</i>	skok na manje od (sa znakom)	$N \oplus V = 1$
<b>BLEQ</b> <i>p</i>	skok na manje od ili jednako (sa znakom)	$(N \oplus V) \vee Z = 1$
<b>BNEG</b> <i>p</i>	skok na $N = 1$	$N = 1$
<b>BNNG</b> <i>p</i>	skok na $N = 0$	$N = 0$
<b>BOVF</b> <i>p</i>	skok na $V = 1$	$V = 1$
<b>BNVF</b> <i>p</i>	skok na $V = 0$	$V = 0$

# Instrukcije za rad sa potprogramom

Instrukcija skoka na potprogram: **JSR *adr***

- ❑ *adr* je adresa memorijske lokacije u kojoj se nalazi prva instrukcija potprograma

Izvršavanje instrukcije:

- tekući sadržaj *PC* se stavlja na stek
- vrednost *adr* se upisuje u *PC*

Instrukcija povratka iz potprograma: **RTS**

- ❑ instrukcija **RTS** mora da bude poslednja instrukcija potprograma

Izvršavanje instrukcije:

- sadržaj sa vrha steka se upisuje u *PC*
- ovaj sadržaj mora da odgovara vrednosti koju je instrukcija **JSR** stavila na stek

# Instrukcija skoka na prekidnu rutinu

Instrukcija:      **INT** *adr*

- ❑ *adr* je adresa memorijske lokacije u kojoj se nalazi prva instrukcija prekidne rutine

Izvršavanje instrukcije:

- tekući sadržaj *PC* i *PSW* se stavlja na stek
- vrednost *adr* se upisuje u *PC*
- ❑ U nekim realizacijama prekidnog mehanizma, adrese prekidnih rutina se nalaze u tabeli prekida. U tom slučaju, *adr* je broj ulaza u tabeli prekida iz koga treba pročitati adresu prekidne rutine i smestiti je u *PC*.

# Instrukcija povratka iz prekidne rutine

---

Instrukcija :     **RTI**

☐ instrukcija **RTI** mora da bude poslednja instrukcija prekidne rutine

Izvršavanje instrukcije:

- uzimanje sadržaja sa steka i upis u *PSW* i *PC*
- ovaj sadržaj mora da odgovara vrednostima koje je instrukcija **INT** stavila na stek

