

University of Ljubljana  
Faculty of Computer and  
Information Science



# Uvod v računalništvo

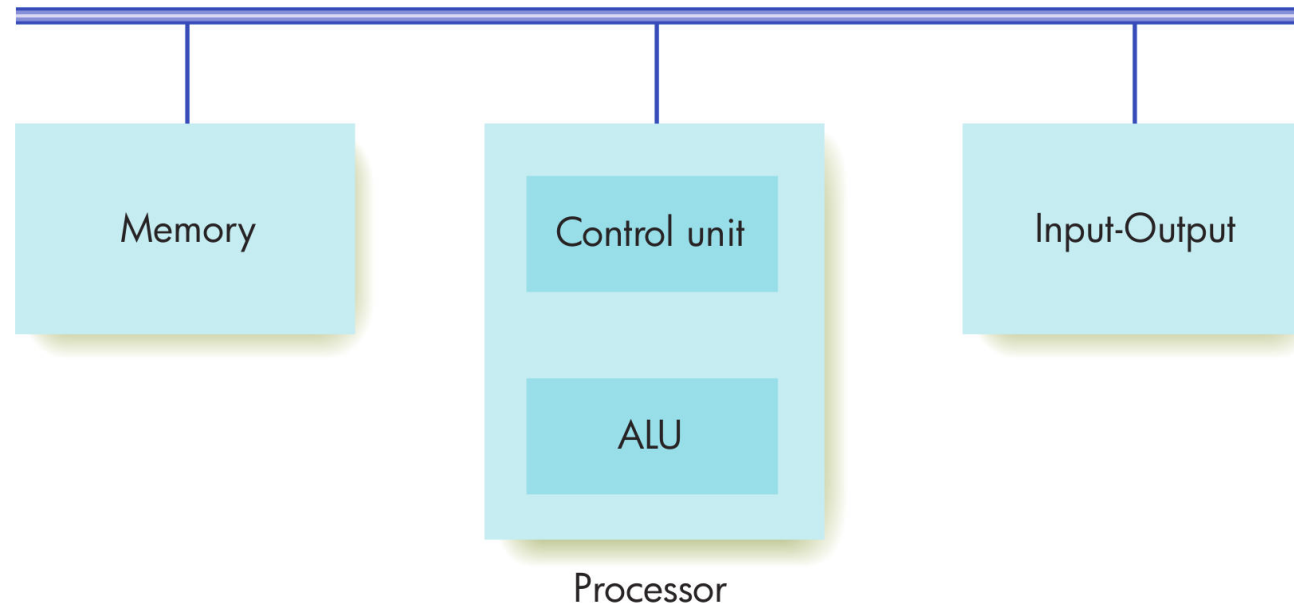
Vaje 3

25.10 -  
1.11.2021



## Računalnik kot stroj (quick recap)

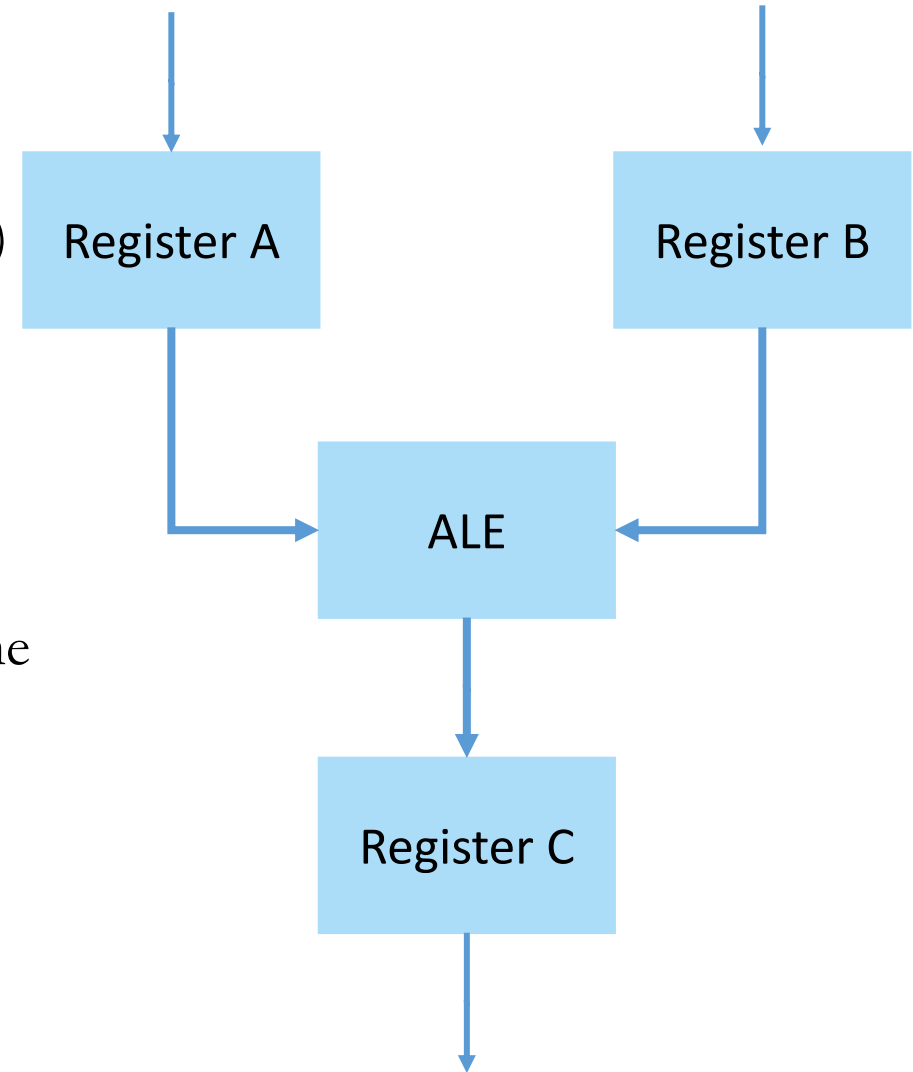
- Računalniki (prenosni/namizni računalnik, pametni telefon, ...) imajo isto zasnovo.
- Von Neumannova arhitektura – 3 karakteristike:
  - Štirje glavni podsistemi: pomnilnik, vhod/izhod, **aritmetično-logična enota (ALE)**, **krmilna enota**.
  - Program shranjen v pomnilniku.
  - Zaporedno izvajanje ukazov.





## Aritmetično-logična enota

- ALE je del procesorja (skupaj s krmilno enoto)
- Vsebuje vezja
  - za aritmetiko: +, -, \*, /
  - za primerjanje in logiko: =, IN, ALI, NE
- Vsebuje registre
  - izredno hitre namenske pomnilniške enote povezane z vezjem ALE
- Podatkovna pot (Data path) kako potuje informacija v ALE
  - iz registrov na vezja
  - iz vezij nazaj na registre





## Strojni ukazi

- Format ukaza v strojnem jeziku:
  - koda operacije
  - naslovi pomnilniških lokacij z operandi

	Operacijska koda	Naslovno polje 1	Naslovno polje 2	...
primer				
št. bitov:	8	16	16	

- **Primer:**
  - op code: 9
  - naslov X: 99
  - naslov Y: 100
  - ADD X,Y: sešteje števili in zapiše rezultat nazaj na lokacijo Y
  - ADD X,Y: 00001001 0000000001100011 0000000001100100





## Strojni ukazi

### ▪ 1. Prenos podatkov

- prenos podatkov
  - pomnilniška lokacija -> register ALE
  - register ALE -> pomnilniška lokacija
  - ena pomnilniška lokacija -> druga pomnilniška lokacija
  - en register ALE -> drugi register ALE
- **Primeri:**
  - LOAD X naloži vsebino pomnilniške lokacije v register R
  - STORE X shrani vsebino registra R na pomnilniško lokacijo X
  - MOVE X,Y kopiraj vsebino pomn. lokacije X na pomn. lokacijo Y

### 2. Aritmetika

- Aritmetične in logične operacije v ALE: +, -, \*, /, IN, AND, OR, NOT
- **Primeri**
  - ADD X,Y,Z  $vr(Z)=vr(X)+vr(Y)$   
tro-naslovni ukaz
  - ADD X,Y  $vr(Y)=vr(X)+vr(Y)$   
dvo-naslovni ukaz
  - ADD X R  $=vr(X)+R$  eno-naslovni ukaz



## Strojni ukazi

### ▪ 3. Primerjanje

- rezultat primerjanja postavi vrednosti bitov pogojnih kod
- primer:
  - COMPARE X,Y primerja vrednosti pomn. lokacij X in Y in postavi vrednosti pogojnih kod:
  - $vr(X) > vr(Y)$  GT=1, EQ=0, LT=0
  - $vr(X) = vr(Y)$  GT=0, EQ=1, LT=0
  - $vr(X) < vr(Y)$  GT=0, EQ=0, LT=1

### ▪ 4. Vejitve

- spreminjanje normalnega zaporednega toka ukazov
- tipično po ukazu za primerjanje
- primeri:
  - JUMP X vzemi naslednji ukaz s pomn. lokacije X
  - JUMPGT X skoči samo, če je indikator GT postavljen na 1
  - JUMPGE X skoči, če sta GT ali EQ indikatorja postavljeni na 1
  - HALT ustavi izvajanje programa4.



## Strojni ukazi

BINARY OP CODE	OPERATION	MEANING
0000	LOAD X	$CON(X) \rightarrow R$
0001	STORE X	$R \rightarrow CON(X)$
0010	CLEAR X	$0 \rightarrow CON(X)$
0011	ADD X	$R + CON(X) \rightarrow R$
0100	INCREMENT X	$CON(X) + 1 \rightarrow CON(X)$
0101	SUBTRACT X	$R - CON(X) \rightarrow R$
0110	DECREMENT X	$CON(X) - 1 \rightarrow CON(X)$
0111	COMPARE X	if $CON(X) > R$ then $GT = 1$ else 0 if $CON(X) = R$ then $EQ = 1$ else 0 if $CON(X) < R$ then $LT = 1$ else 0
1000	JUMP X	Get the next instruction from memory location X.
1001	JUMPGT X	Get the next instruction from memory location X if $GT = 1$ .
1010	JUMPEQ X	Get the next instruction from memory location X if $EQ = 1$ .
1011	JUMPLT X	Get the next instruction from memory location X if $LT = 1$ .
1100	JUMPNEQ X	Get the next instruction from memory location X if $EQ = 0$ .
1101	IN X	Input an integer value from the standard input device and store into memory cell X.
1110	OUT X	Output, in decimal notation, the value stored in memory cell X.
1111	HALT	Stop program execution.



## Naloga 3.1

Predpostavite, da register R ter pomnilniške celice 80 in 81 vsebujejo naslednje vrednosti:

- R: 20,
- pomnilniška celica 80: 43,
- pomnilniška celica 81: 97.

Te vrednosti se nahajajo v registru oz. obeh pomnilniških celicah pred izvedbo vsakega izmed spodnjih ukazov zbirnega jezika. Ugotovite, kakšne so vrednosti v registru in obeh pomnilniških celicah po izvedbi vsakega izmed ukazov.

- (a) LOAD 80
- (b) STORE 81
- (c) COMPARE 80
- (d) ADD 81
- (e) IN 80
- (f) OUT 80

	R	M(80)	M(81)
(a)			
(b)			
(c)			
(d)			
(e)			
(f)			





## Naloga 3.1

Predpostavite, da register R ter pomnilniške celice 80 in 81 vsebujejo naslednje vrednosti:

- R: 20,
- pomnilniška celica 80: 43,
- pomnilniška celica 81: 97.

Te vrednosti se nahajajo v registru oz. obeh pomnilniških celicah pred izvedbo vsakega izmed spodnjih ukazov zbirnega jezika. Ugotovite, kakšne so vrednosti v registru in obeh pomnilniških celicah po izvedbi vsakega izmed ukazov.

- LOAD 80
- STORE 81
- COMPARE 80
- ADD 81
- IN 80
- OUT 80

(a)	R:43	M(80):43	M(81):97
(b)	R:20	M(80):43	M(81):20
(c)	R:20	M(80):43 GT=1	M(81):97
(d)	R:117	M(80):43	M(81):97
(e)	R:20	M(80): vhodna vrednost	M(81):97
(f)	R:20	M(80): 43 izpiše se vrednost 43	M(81):97



## Naloga 3.2

Predpostavite, da register R ter pomnilniške celice 80 in 81 vsebujejo naslednje vrednosti:

- R: 100,
- pomnilniška celica 50: 90,
- pomnilniška celica 55: 35.

Te vrednosti se nahajajo v registru oz. obeh pomnilniških celicah pred izvedbo vsakega izmed spodnjih ukazov zbirnega jezika. Ugotovite, kakšne so vrednosti v registru in obeh pomnilniških celicah po izvedbi vsakega izmed ukazov.

- (a) STORE 50
- (b) ADD 55
- (c) CLEAR 55
- (d) INCREMENT 55
- (e) DECREMENT 50
- (f) SUBSTRACT 50

	R	M(50)	M(55)
(a)			
(b)			
(c)			
(d)			
(e)			
(f)			



## Naloga 3.2

Predpostavite, da register R ter pomnilniške celice 80 in 81 vsebujejo naslednje vrednosti:

- R: 100,
- pomnilniška celica 50: 90,
- pomnilniška celica 55: 35.

Te vrednosti se nahajajo v registru oz. obeh pomnilniških celicah pred izvedbo vsakega izmed spodnjih ukazov zbirnega jezika. Ugotovite, kakšne so vrednosti v registru in obeh pomnilniških celicah po izvedbi vsakega izmed ukazov.

- (a) STORE 50
- (b) ADD 55
- (c) CLEAR 55
- (d) INCREMENT 55
- (e) DECREMENT 50
- (f) SUBSTRACT 50

(a)	R:100	M(50):100	M(55):35
(b)	R:135	M(50):90	M(55):35
(c)	R:100	M(50):90	M(55):0
(d)	R:100	M(50):90	M(55):36
(e)	R:100	M(50):89	M(55):35
(f)	R:10	M(50):90	M(55):35



## Naloga 3.3

Predpostavite, da pomnilniška celica 50 vsebuje vrednost 4, oznaka L pa ustreza pomnilniški lokaciji. Katero vrednost v register R naloži vsak izmed naslednjih ukazov LOAD?

- (a) LOAD 50
- (b) LOAD 4
- (c) LOAD L
- (d) LOAD L+1 (predpostavimo da je operacija dovoljena)





## Naloga 3.3

Predpostavite, da pomnilniška celica 50 vsebuje vrednost 4, oznaka L pa ustreza pomnilniški lokaciji. Katero vrednost v register R naloži vsak izmed naslednjih ukazov LOAD?

- (a) LOAD 50
- (b) LOAD 4
- (c) LOAD L
- (d) LOAD L+1 (predpostavimo da je operacija dovoljena)

(a)	4
(b)	Kopija vsebine pomnilnike celice 4
(c)	Oznaka L je ekvivalentna naslovu 50 , zato je ta ukaz ekvivalenten ukazu LOAD 50
(d)	Kopija vsebine pomnilniške celice 51. Ukaz izvaja aritmetiko na naslovih, ne sami vsebini.



## Naloga 3.4

Dan je naslednji program v strojnem jeziku, tabela simbolov in tabela kod ukazov.

Operacijska koda ukaza je 4 bitna, naslovi 12 bitni. Kakšen je ustrezen program v zbirniku?

(UR P45 Računalnik kot stroj.pptx, sl.49)

(a) 0101001100001100

(b) 0011000000000111





## Naloga 3.4

Dan je naslednji program v strojnem jeziku, tabela simbolov in tabela kod ukazov.

Operacijska koda ukaza je 4 bitna, naslovi 12 bitni. Kakšen je ustrezen program v zbirniku?

(UR P45 Računalnik kot stroj.pptx, sl.49)

(a) 0101001100001100

(b) 0011000000000111

\*Koda ukaza = 4 biti

\*Naslov = 12 bitov

(a) **0101001100001100**

Koda ukaza: 0101

Naslov: 001100001100

→ SUBTRACT  $2^9 + 2^8 + 2^3 + 2^2$

→ SUBTRACT 780

(b) **0011000000000111**

Koda ukaza: 0101

Naslov: 001100001100

→ ADD  $2^2 + 2^1 + 2^0$

→ ADD 7



## Naloga 3.5

Z uporabo nabora ukazov s predavanj prevedite naslednje algoritmične operacije v zbirni jezik. Zapišite tudi vse potrebne psevdo-ukaze .DATA

Če  $x > 50$

izpiši vrednost  $x$

sicer

preberi novo vrednost  $x$

\*Predpostavimo da je  $X$ -u predhodno bila prirejena vrednost







## Naloga 3.5

Z uporabo nabora ukazov s predavanj prevedite naslednje algoritmične operacije v zbirni jezik. Zapišite tudi vse potrebne psevdo-ukaze .DATA

Če  $x > 50$

izpiši vrednost  $x$

sicer

preberi novo vrednost  $x$

\*Predpostavimo da je  $X$ -u predhodno bila prirejena vrednost

```
LOAD FIFTY    % konstanta 50
COMPARE X
JUMPGT THEN
IN X
JUMP DONE
THEN: OUT X
DONE: naslednji ukaz gre sem
```

```
X:      .DATA 0
FIFTY:  .DATA 50
```



## Naloga 3.6

Recimo, da posamezen program med izvajanjem 50% svojega časa porabi za čakanje na zaključek vhodno/izhodnih operacij.

- a) Kolikšen je odstotek časa, v katerem procesor opravlja koristno delo (oz. kolikšen je t.i. izkoristek procesorja), če so v pomnilniku naloženi trije programi?
- b) Koliko programov bi morali imeti v pomnilniku, če bi želeli doseči vsaj 95% izkoristek procesorja?





## Naloga 3.6

Recimo, da posamezen program med izvajanjem 50% svojega časa porabi za čakanje na zaključek vhodno/izhodnih operacij.

- Kolikšen je odstotek časa, v katerem procesor opravlja koristno delo (oz. kolikšen je t.i. izkoristek procesorja), če so v pomnilniku naloženi trije programi?
- Koliko programov bi morali imeti v pomnilniku, če bi želeli doseči vsaj 95% izkoristek procesorja?

3 programi:  $\frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{1}{8}$

Verjetnost, da vsi trije programi čakajo na zaključek operacij je  $\frac{1}{8}$ , oz. 12,5%

Torej, procesor opravlja koristno delo:  $1 - \frac{1}{8} = \frac{7}{8}$ , oz. 87,5%

4 program:  $1 - (\frac{1}{2})^4 = 1 - \frac{1}{16} = \frac{15}{16} = 93,75\%$  časa

5 programov:  $1 - (\frac{1}{2})^5 = 1 - \frac{1}{32} = \frac{31}{32} = 96,875\%$  časa



## Naloga 3.7

Predpostavite da register R ter pomnilniške celice 50 in 55 vsebujejo naslednje vrednosti:

- R: 45,
- pomnilniška celica 50: 115,
- pomnilniška celica 55: 23.

Te vrednosti se nahajajo v registru oz. obeh pomnilniških celicah pred izvedbo vsakega izmed spodnjih ukazov zbirnega jezika. Ugotovite, kakšne so vrednosti v registru in obeh pomnilniških celicah po izvedbi vsakega izmed ukazov.

- LOAD 50
- INCREMENT 55
- COMPARE 55
- ADD 50
- DECREMENT 50
- OUT 55



## Naloga 3.7

Predpostavite da register R ter pomnilniške celice 50 in 55 vsebujejo naslednje vrednosti:

- R: 45,
- pomnilniška celica 50: 115,
- pomnilniška celica 55: 23.

- (a) LOAD 50
- (b) INCREMENT 55
- (c) COMPARE 55
- (d) ADD 50
- (e) DECREMENT 50
- (f) OUT 55

(a)	R:115	M(50):115	M(55):23
(b)	R:45	M(50):115	M(55):24
(c)	R:45	M(50):115	M(55):23; LT: 1
(d)	R:160	M(50):115	M(55):23
(e)	R:45	M(50):114	M(55):23
(f)	R:45	M(50):115	M(55):23 izpiše 23



## Naloga 3.8

Dan je naslednji program v strojnem jeziku, tabela simbolov in tabela kod ukazov.

Operacijska koda ukaza je 4 bitna, naslovi 12 bitni. Kakšen je ustrezen program v zbirniku?

(UR P45 Računalnik kot stroj.pptx, sl.49)

(a) 0100001000001001

(b) 0111001101111100





## Naloga 3.8

Dan je naslednji program v strojnem jeziku, tabela simbolov in tabela kod ukazov.

Operacijska koda ukaza je 4 bitna, naslovi 12 bitni. Kakšen je ustrezen program v zbirniku?  
(UR P45 Računalnik kot stroj.pptx, sl.49)

(a) 0100001000001001

INCREMENT 521

(b) 0111001101111100

COMPARE 892



## Naloga 3.9

Z uporabo nabora ukazov s predavanj prevedite naslednje algoritmične operacije v zbirni jezik. Zapišite tudi vse potrebne psevdo-ukaze .DATA

vsota = 0

I = 0

Dokler  $I < 50$ , ponavljaj

    vsota = vsota + I

    I = I + 1







## Naloga 3.9

Z uporabo nabora ukazov s predavanj prevedite naslednje algoritmične operacije v zbirni jezik. Zapišite tudi vse potrebne psevdo-ukaze .DATA

$vsota = 0$

$I = 0$

Dokler  $I < 50$ , ponavljaj

$vsota = vsota + I$

$I = I + 1$

```
LOAD ZERO % naloži 0 v R
STORE SUM % inicializiraj SUM na 0
STORE I   % inicializiraj I na 0
LOOP: LOAD FIFTY   % naloži 50 v R
      COMPARE I
      JUMPEQ DONE
      LOAD SUM
      ADD I
      STORE SUM
      INCREMENT I
      JUMP LOOP
DONE: naslednji ukaz gre sem
SUM:      .DATA 0
I:        .DATA 0
ZERO:     .DATA 0
FIFTY:    .DATA 50
```



Hvala za pozornost!

Petar.Kochovski@fri.uni-lj.si

