

UvR

Dodatne naloge

1. del

Naloga 1

Recimo, da je zgradba ukaznega registra naslednja:

op code	address-1	address-2
6 bitov	18 bitov	18 bitov

1. Kakšno je maksimalno število različnih kod ukazov, ki jih lahko prepozna in izvede procesor na tem računalniku?
2. Kakšna je maksimalna velikost pomnilnika na tem računalniku?
3. Koliko bajtov je potrebnih za vsako operacijo?

Naloga 2

Imamo glavni pomnilnik velikosti 16 GB in z dostopnim časom 24 ns.

- (a) Kako velik je naslovni register?
- (b) Povprečni dostopni čas želimo z uporabo predpomnilnika zmanjšati na 18 ns. Kakšen dostopni čas mora imeti predpomnilnik, če vemo, da je verjetnost zadetka enaka 75%?

Naloga 3

Dostopni čas do glavnega pomnilnika je 30 ns. Povprečni čas dostopa smo znižali na 13 ns, tako da smo vgradili predpomnilnik in dosegli 90 % verjetnost zadetka. Kolikšen je dostopni čas predpomnilnika?

Naloga 4

Jaka ima računalnik opremljen s pomnilnikom s povprečnim časom dostopa 50ns in desetkrat hitrejšim predpomnilnikom. Kakšen je skupni povprečni čas dostopa, če je verjetnost zadetka v predpomnilniku enaka 80%? Jaka bi rad kupil nov pomnilnik, da bi se skupni povprečni čas dostopa prepolovil. Kolikšen mora biti povprečni čas dostopa do novega predpomnilnika? [10 točk]

Naloga 5

Imamo dva računalniška sistema. Prvi ima vgrajen glavni pomnilnik z dostopnim časom 60 ns ter predpomnilnik, pri katerem smo izmerili 60 % verjetnost zadetka. Drugi sistem ima vgrajen glavni pomnilnik z dostopnim časom 120 ns ter predpomnilnik z enakim dostopnim časom kot prvi sistem. Kolikšno verjetnost zadetka v predpomnilniku mora doseči drugi sistem, da bosta oba v povprečju enako hitro dostopala do podatkov?

Naloga 6

Z uporabo nabora ukazov s predavanj napišite program v zbirnem jeziku (vključno z vsemi potrebnimi psevdo-ukazi), ki na vhodu bere števila in obenem šteje, koliko teh števil je že prebral. Branje se nadaljuje, dokler se na vhodu ne pojavi negativno število. Takrat program izpiše, koliko pozitivnih vhodnih števil je bilo prebranih in se ustavi. Če so vhodni podatki na primer 42, 108, 99, 60, 1, 42, 3, -27 , naj bi program torej izpisal vrednost 7.

Naloga 7

Z uporabo nabora ukazov s predavanj napišite program v zbirnem jeziku (vključno z vsemi potrebnimi psevdo-ukazi), ki prebere zaporedje celih števil, eno za drugim, in izpiše največje in najmanjše izmed njih. Predpostavite, da se na vhodu vedno pojavi natanko 100 celih števil.

Naloga 8

Napišite program v zbirnem jeziku, ki s standardnega vhoda bere števila, dokler ne vpišemo nič in na koncu izpiše vsoto vseh vpisanih negativnih števil. [20 točk]

Naloga 9

5. Napišite program v zbirniku, ki na ekranu prikazuje številke, ki jih vnaša uporabnik. To stori tako, da na vходу prebere vneseno številko, potem pa na izhod pošlje njeno ASCII kodo. Program naj se ustavi, ko uporabnik vnese število, ki je večje kot 9. Primeri za vхod in izхod: [10 točk]

1 2 9 7 8 45 -> 49 50 57 55 56

0 0 3 120 -> 48 48 51

ASCII	znak
48	'0'
49	'1'
50	'2'
...	...
57	'9'

Naloga 10

5. Dan je naslednji program v zbirniku. Operacijska koda ukaza je 4 bitna, naslovi pa 8 bitni. Prevedite program v strojni jezik in napišite psevdokodo programa ter tudi z besedami opišite kakšen rezultat vrne program. Predpostavimo da je STEVILO>X. [10 točk]

Opis:

naslov	strojni ukaz	zbirnik
0000		IN STEVILO
0001		IN X
0010		LOAD STEVILO
0011		LOOP: COMPARE X
0100		JUMPEQ YES
0101		JUMPGT NO
0110		SUBTRACT X
0111		JUMP LOOP
1000		YES: OUT ONE
1001		HALT
1010		NO: OUT ZERO
1011		HALT
1100		STEVILO: .DATA 0
1101		X: .DATA 0
1110		ONE: .DATA 1
1111		ZERO: .DATA 0

Psevdokoda:

UKAZ	Op. koda
LOAD	0000
STORE	0001
CLEAR	0010
ADD	0011
INCREMENT	0100
SUBTRACT	0101
DECREMENT	0110
COMPARE	0111
JUMP	1000
JUMPGT	1001
JUMPEQ	1010
JUMPLT	1011
JUMPNEQ	1100
IN	1101
OUT	1110
HALT	1111

Naloga 11

4. Dan je naslednji algoritem, tabela simbolov in tabela kod ukazov:

```
Vnesi vrednost X
Vnesi vrednost Y
if  $X \geq Y$  then
    Izpiši vrednost X
else
    Izpiši vrednost Y
```

Dopolni programa v zbirniku in strojnem jeziku. Operacijska koda ukaza je 4 bitna, naslovi so 6 bitni. [15 točk]

naslov	strojni ukaz	zbirnik
0000		
0001		
0010		
0011		
0100	1001	
0101		
0110		
0111		
1000		
1001	0000	
1010	000000	

		UKAZ	Op. koda
		LOAD	0000
		STORE	0001
		CLEAR	0010
		ADD	0011
		INCREMENT	0100
		SUBTRACT	0101
		DECREMENT	0110
		COMPARE	0111
		JUMP	1000
		JUMPGT	1001
		JUMPEQ	1010
		JUMPLT	1011
		JUMPNEQ	1100
		IN	1101
		OUT	1110
		HALT	1111

simbol	naslov (DEC)
PRINTY	7
DONE	8
X	9
Y	10

REŠITVE

Rešitev - Naloga 1

Maksimalno število različnih kod ukazov, ki jih lahko prepozna in izvede procesor na tem računalniku je $2^6 = 64$.

Maksimalna velikost pomnilnika na tem računalniku je 2^{18}B

Za vsako operacijo je potrebnih 6 bajtov: $6+18+18 = 42 \text{bitov}$ (5B ne zadostuje).

Rešitev - Naloga 2

(a) $2^4 \cdot 2^{30} \text{ B} = 2^{24} \text{ B} \longrightarrow 24 \text{b naslovni register}$

(b) $0,75 \cdot x + 0,25 \cdot (x + 24 \text{ns}) = 18 \text{ ns}$

$$0,75 \cdot x + 0,25 \cdot x + 0,25 \cdot 24 = 18$$

$$x + 6 = 18$$

$$x = 12 \text{ ns}$$

Rešitev - Naloga 3

$$0.9x + 0.1 \cdot (x + 30 \text{ ns}) = 13 \text{ ns}$$

$$0.9x + 0.1x + 3 \text{ ns} = 13 \text{ ns}$$

$$x = 10 \text{ ns}$$

Rešitev - Naloga 4

$$0,8 \cdot 5 + 0,2 \cdot (5 + 50) = 15 \text{ns}$$

$$0,8 \cdot 5 + 0,2 \cdot (5 + x) = 7,5, \quad X = 12,5 \text{ns}$$

Rešitev - Naloga 5

$$\begin{aligned}t_1 &= 0,6 \cdot t_p + 0,4(t_p + 60\text{ns}) \\t_2 &= x \cdot t_p + (1 - x)(t_p + 120\text{ns}) \\t_1 &= t_2 \\0,6 \cdot t_p + 0,4(t_p + 60\text{ns}) &= x \cdot t_p + (1 - x)(t_p + 120\text{ns}) \\0,6 \cdot t_p + 0,4(t_p + 60\text{ns}) - x \cdot t_p - (1 - x)(t_p + 120\text{ns}) &= 0 \\0,6 \cdot t_p + 0,4 \cdot t_p + 0,4 \cdot 60 - x \cdot t_p - t_p + t_p x - 120 + 120x &= 0 \\24 - 120 + 120x &= 0 \\120x &= 120 - 24 \\x &= 96/120 \\x &= 0.8\end{aligned}$$

Rešitev - Naloga 6

Rešitev

```
.BEGIN                % Prva vrstica programa
IN NUMBER             % Preberi prvo število
LOOP: LOAD ZERO
COMPARE NUMBER        % Ali je število < 0 ?
JUMPLT DONE           % Število je negativno
INCREMENT COUNT       % Število je nenegativno (povečaj števec)
IN NUMBER             % Preberi naslednje število
JUMP LOOP             % Ponovi zanko
DONE: OUT COUNT        % Izpiši končno število pojavitev
HALT                  % Ta ukaz zaključuje izvajanje programa
COUNT: .DATA 0       % Število pojavitev nenegativnih števil
ZERO: .DATA 0          % Konstanta 0 ki jo uporabljamo za primerjavo
NUMBER: .DATA 0        % Prostor za shranjevanje vhodne vrednosti
.END                  % Zadnja vrstica programa
```

Rešitev - Naloga 7

```
IN MAX          % Preberi prvo vrednost kot MAX
LOAD MAX
STORE MIN       % Shrani prvo vrednost kot MIN
AGAIN: INCREMENT COUNTER % Branje naslednjega števila
IN N
LOAD N
COMPARE MAX
JUMPLT LARGER   % Če MAX < N, N je novi MAX
COMPARE MIN
JUMPGT SMALLER  % MIN > N, N je novi MIN
JUMP TEST       % N ni novi MAX ali MIN
LARGER: STORE MAX % Shrani N v MAX
JUMP TEST
SMALLER: STORE MIN % Shrani N v MIN
TEST: LOAD COUNTER
COMPARE HUNDRED
JUMPGT AGAIN    % Manj kot 100 vrednosti prebranih
DONE: OUT MAX   % Izpiši MAX in MIN
OUT MIN
HALT
N: .DATA 0
MAX: .DATA 0
MIN: .DATA 0
COUNTER: .DATA 1
HUNDRED: .DATA 100
```

Rešitev - Naloga 8

Rešite sami!

Rešitev - Naloga 9

```
.BEGIN
LOOP: IN N
LOAD N
COMPARE NINE
JUMPLT KONEC
ADD FT
STORE N
OUT N
JUMP LOOP
KONEC: HALT
N: .DATA 0
NINE: .DATA 9
FT: .DATA 48
.END
```


Rešitev - Naloga 10

naslov	strojni ukaz	zbirnik
0000	1101 00001100	IN STEVILO
0001	1101 00001101	IN X
0010	0000 00001100	LOAD STEVILO
0011	0111 00001101	LOOP: COMPARE X
0100	1010 00001000	JUMPEQ YES
0101	1001 00001010	JUMPGT NO
0110	0101 00001101	SUBTRACT X
0111	1000 00000011	JUMP LOOP
1000	1110 00001110	YES: OUT ONE
1001	1111 00000000	HALT
1010	1110 00001111	NO: OUT ZERO
1011	1111 00000000	HALT
1100	000000000000	STEVILO: .DATA 0
1101	000000000000	X: .DATA 0
1110	000000000001	ONE: .DATA 1
1111	000000000000	ZERO: .DATA 0

Psevdokoda:

```

read STEVILO, X
while(STEVILO > X):
    STEVILO = STEVILO - X
if (stevalo==x):
    print 1
else:
    print 0

```

UKAZ	Op. koda
LOAD	0000
STORE	0001
CLEAR	0010
ADD	0011
INCREMENT	0100
SUBTRACT	0101
DECREMENT	0110
COMPARE	0111
JUMP	1000
JUMPGT	1001
JUMPEQ	1010
JUMPLT	1011
JUMPNEQ	1100
IN	1101
OUT	1110
HALT	1111

Rešitev - Naloga 11

naslov	strojni ukaz	zbirnik
0000	1101 001001	IN X
0001	1101 001010	IN Y
0010	0000 001001	LOAD X
0011	0111 001010	COMPARE Y
0100	1001 000111	JUMPGT PRINTY
0101	1110 001001	OUT X
0110	1000 001000	JUMP DONE
0111	1110 001010	PRINTY: OUT Y
1000	1111 000000	DONE: HALT
1001	0000 000000	X: .DATA 0
1010	0000 000000	Y: .DATA 0