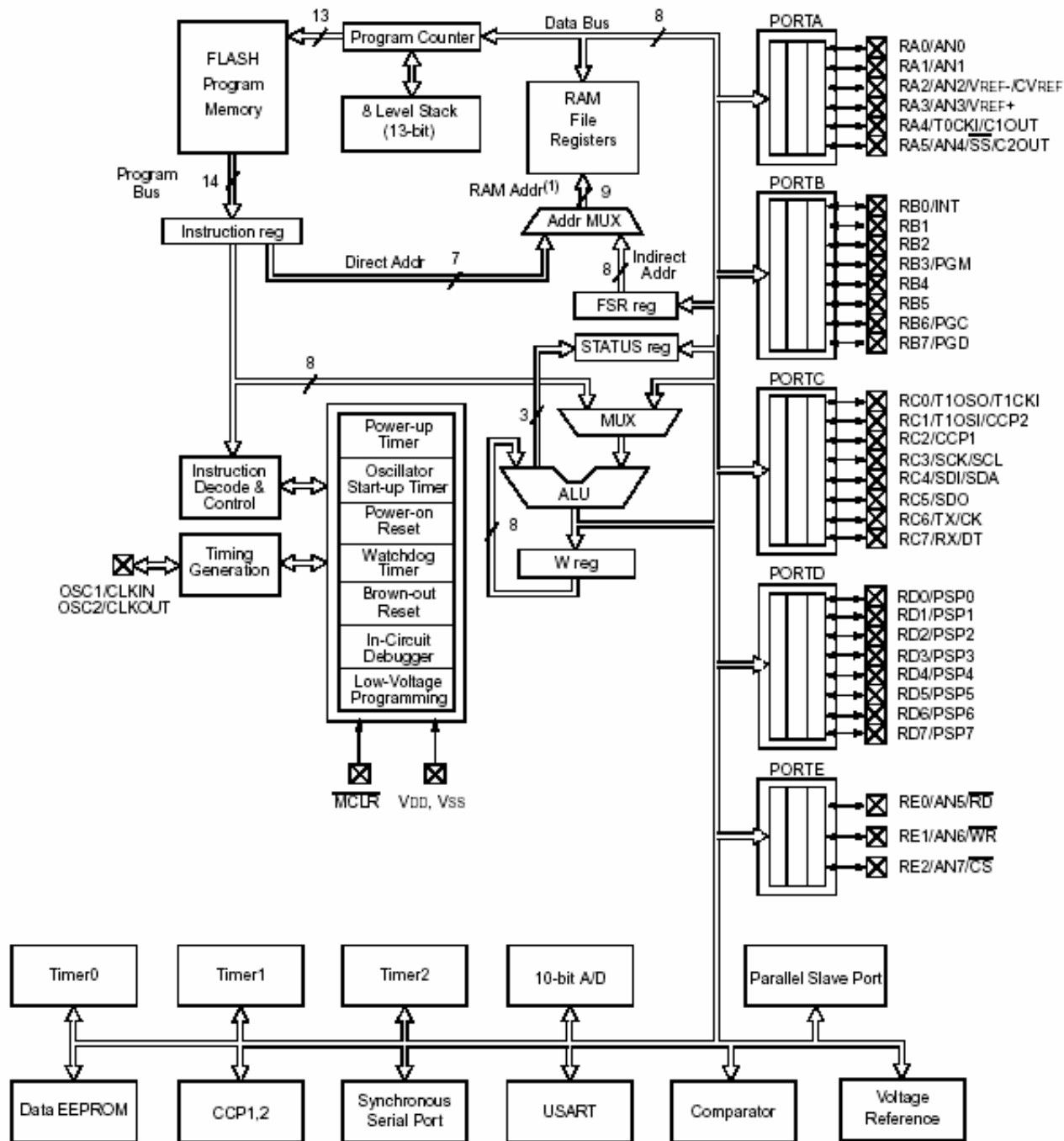
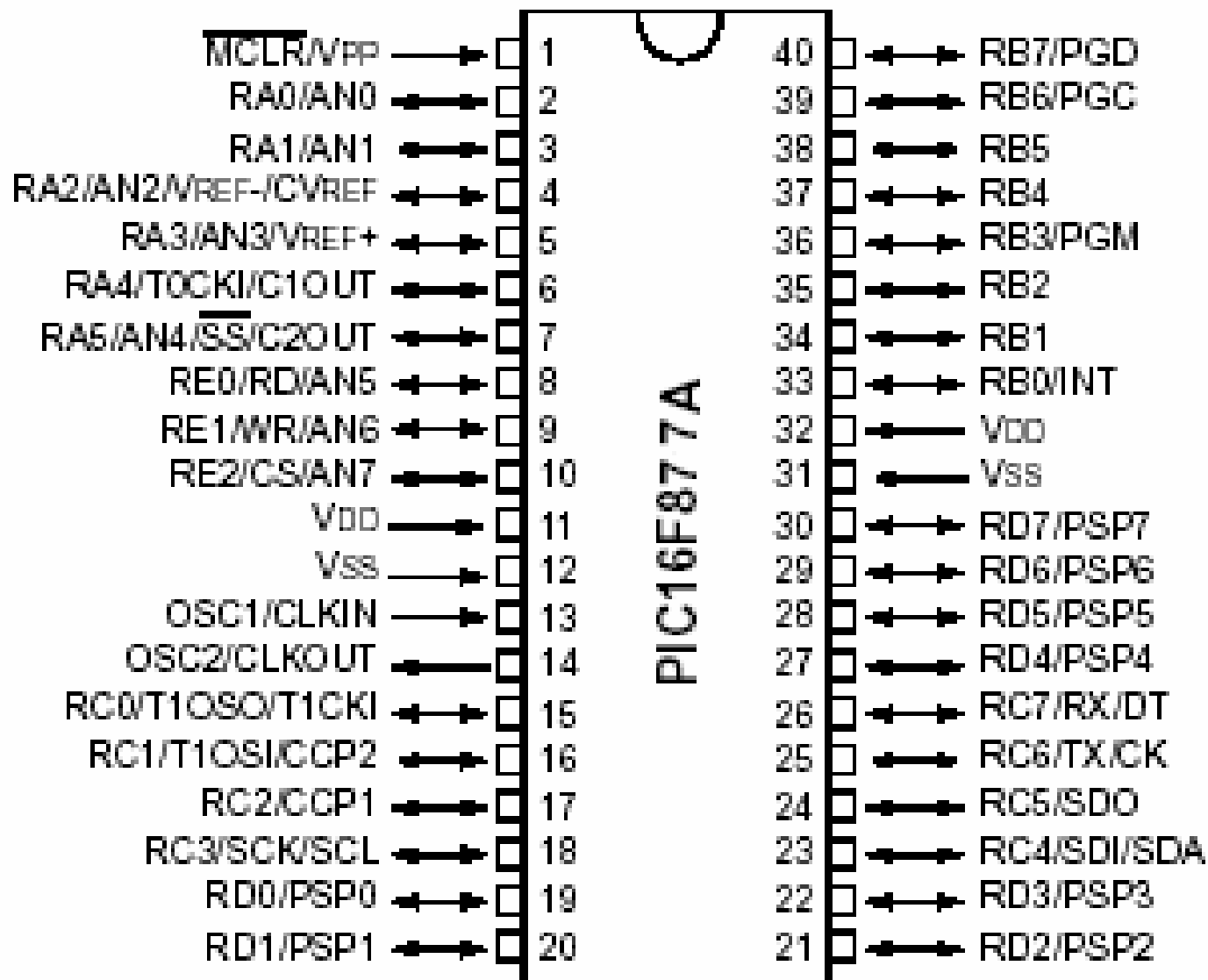


# PIC 16F877A

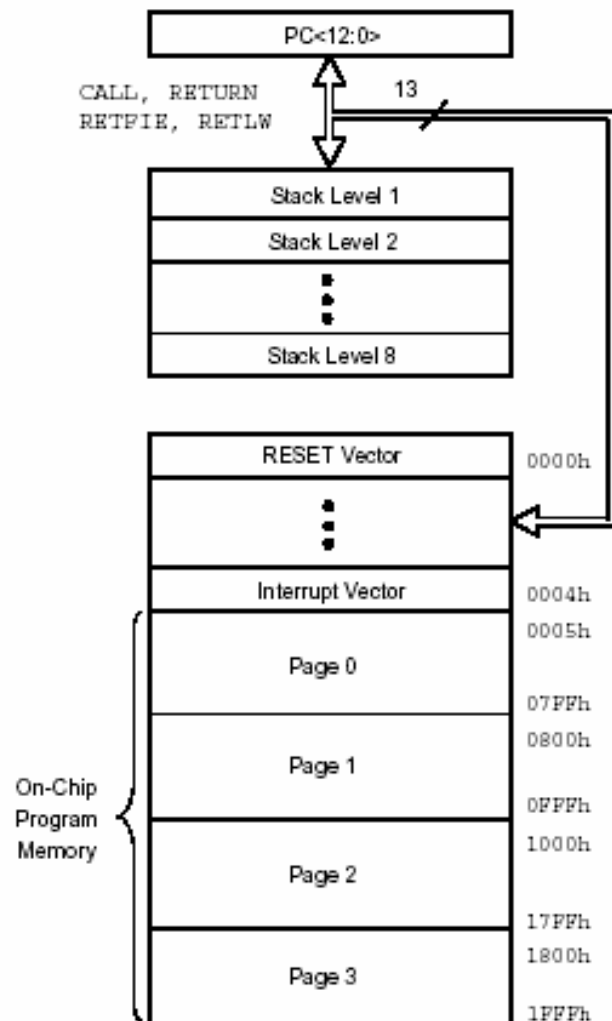
- RISC mikrokrmilnik
- 35 ukazov fiksne dolžine
- 8Kx14bitnih besed programskega pomnilnika (FLASH)
- 368 zlogov podatkovnega pomnilnika (registri)
- 256 zlogov električno zbrisljivega pomnilnika (EEPROM)
- 33 vhodno/izhodnih priključkov
- 3 časovniki (2x8 bit in 1x16 bit)
- 8 analognih vhodov za analogno/digitalno pretvorbo
- 2 analogna primerjalnika
- sinhroni in asinhroni serijski vmesnik (I<sup>2</sup>C, RS232, ...)
- 15 prekinitvenih izvorov





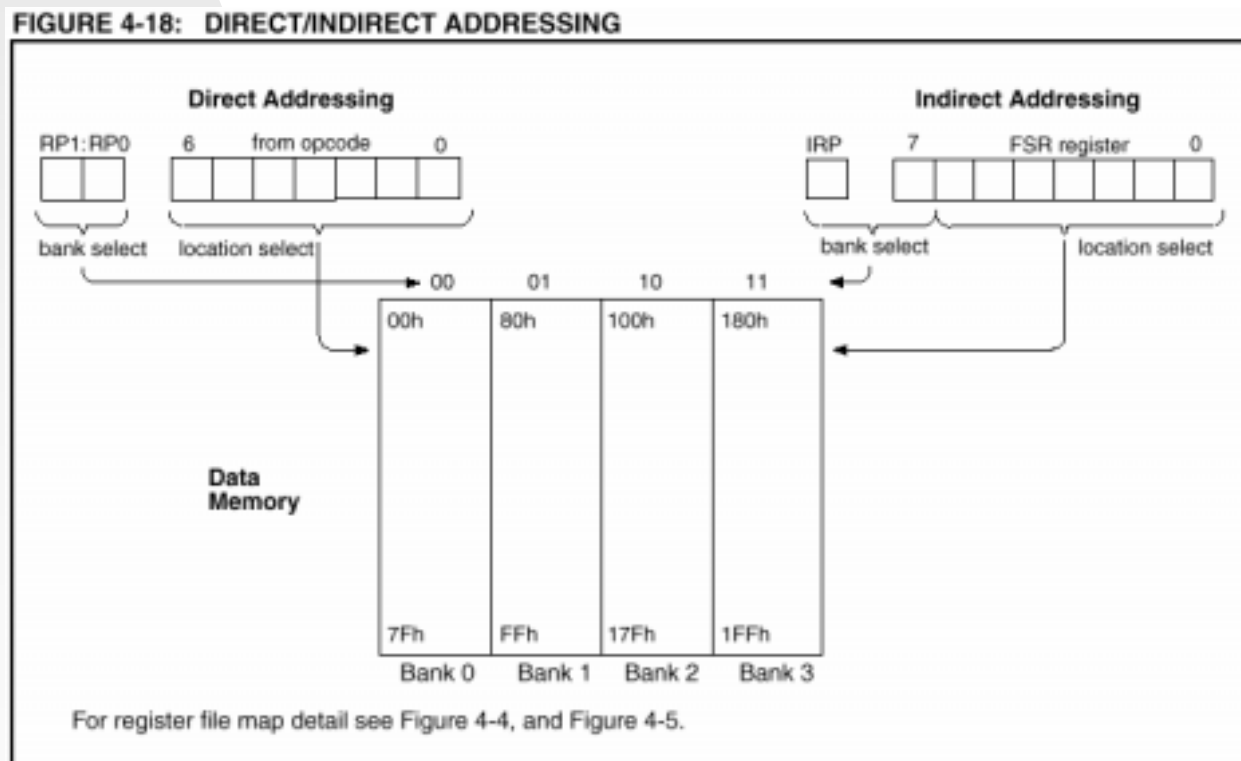
# Organizacija pomnilnika

- Programski pomnilnik



- 13 bitni naslovi (8K)
- 8 vgnezditev klicev podprogramov
- razdeljen na strani po 2K
- reset vektor na lokaciji 0
- prekinitveni vektor na lokaciji 4

- Podatkovni pomnilnik
  - Vsebuje tako registre mikroprocesorja kot tudi registre perifernih vmesnikov (razern registra W)
  - Razdeljen je na štiri banke
  - Omogočen je direktni in posredni dostop do podatkov

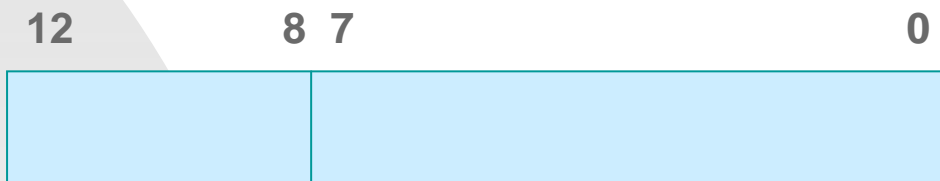




- Osnovni namenski registri

Programski števec

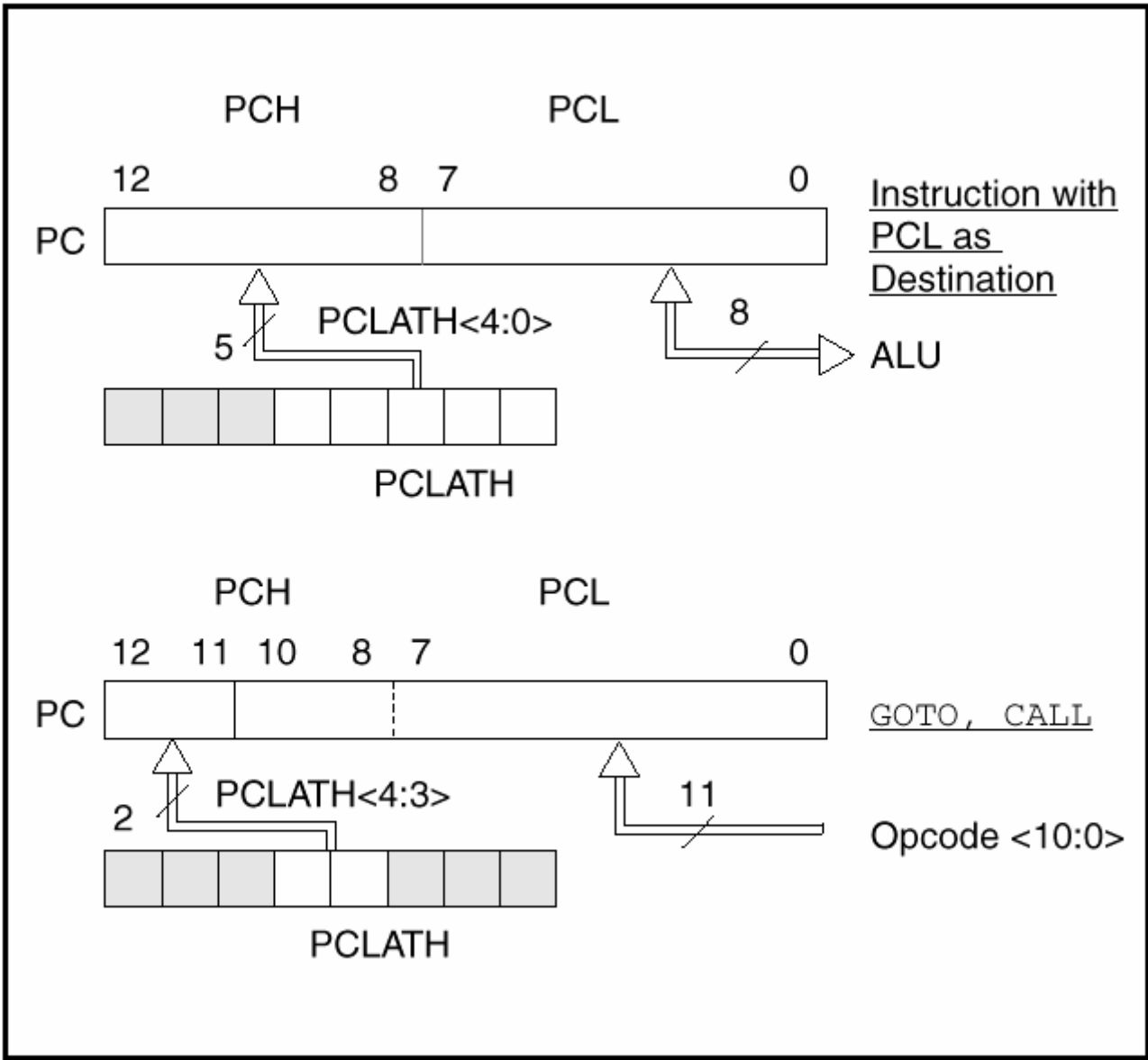
PC equ 2



Zadrževalni register za PŠ

PCLATH equ 0ah

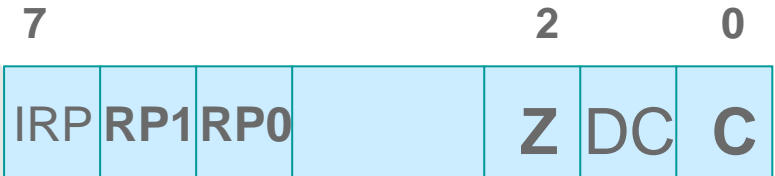






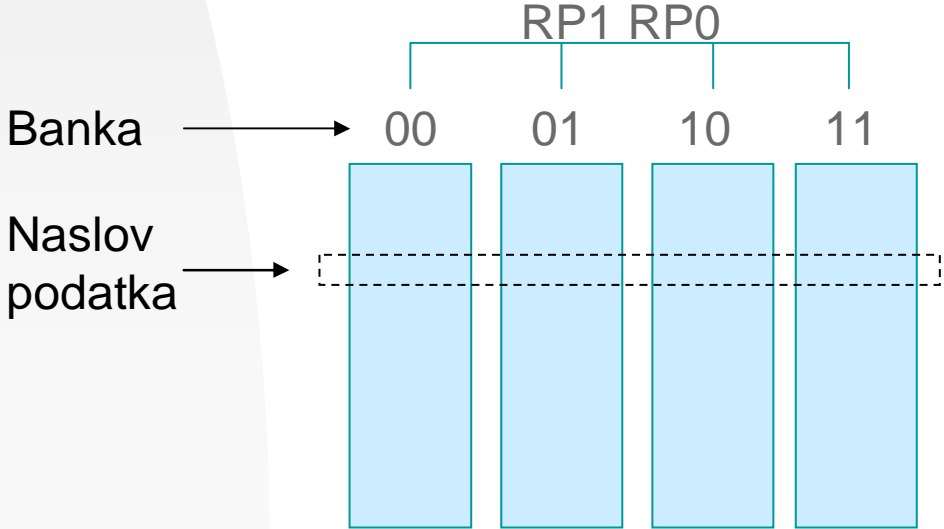
# Statusni register

STATUS equ 3



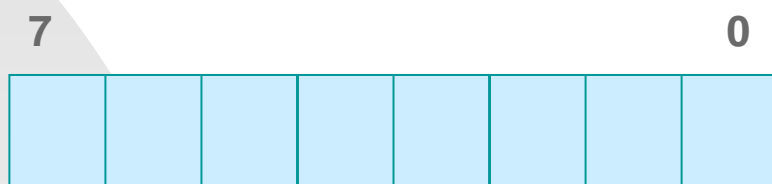
- Z - 1 => rezultat operacije = 0
- C - 1 => pri operaciji je prišlo do prenosa
- DC-1 => prenos med bitoma 3 in 4

IRP,RP1,RP0 - naslavljanje registrov



	00000001
	+11111111
	<hr style="width: 100%;"/>
C =	10000000
DC = 1	Z = 1

Delovni register W



Ni v podatkovnem pomnilniku !

Nastopa kot eden od operandov pri vseh binarnih aritmetičnih in logičnih operacijah.

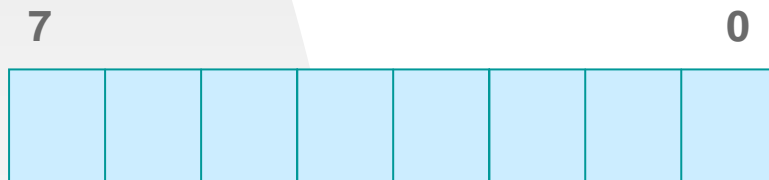
## Vhodno izhodni registri

- Registri smeri (TRISA, TRISB, TRISC, TRISD, TRISE)

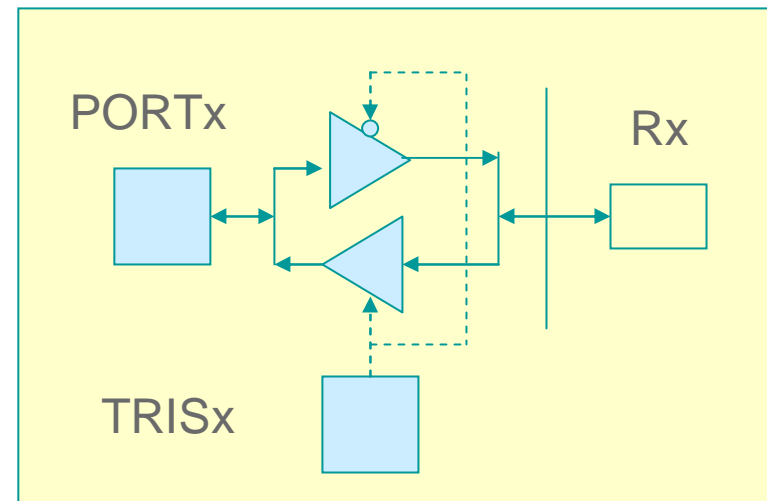


0 - izhod  
1 - vhod

- Vhodno/izhodna vodila (PORTA, PORTB, ..., PORTE)



Večina priključkov ima dvojno funkcije. Ob zagonu so vodila B, C in D konfigurirana kot vhodno/izhodni priključki (vhodi), vodili A in E pa kot analogni vhodi.

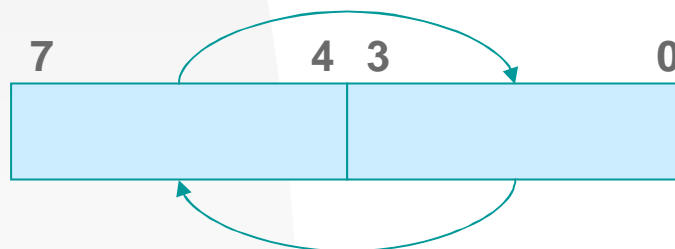


# Nabor ukazov

- Ukazi za premikanje podatkov

MOVF	f,d	$f \rightarrow W$ ali $f \rightarrow f$
MOVWF	f	$W \rightarrow f$
MOVLW	k	$k \rightarrow W$
CLRF	f	$0 \rightarrow f$
CLRW		$0 \rightarrow W$
SWAPF	f,d	zamenja spodnje in zgornje 4 bite

f - številka registra  
d - rezultat operacije  
(0 - W, 1 - reg)  
k - konstanta



- Aritmetični ukazi

ADDWF	f,d	$f+W \rightarrow W$ ali $f+W \rightarrow f$
SUBWF	f,d	$f-W \rightarrow W$ ali $f-W \rightarrow f$
INCF	f,d	$f+1 \rightarrow W$ ali $f+1 \rightarrow f$
DECF	f,d	$f-1 \rightarrow W$ ali $f-1 \rightarrow f$
ADDLW	k	$k+W \rightarrow W$
SUBLW	k	$k-W \rightarrow W$

f - številka registra  
d - rezultat operacije  
(0 - W, 1 - reg)  
k - konstanta

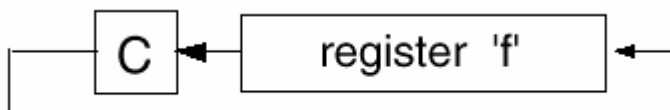
- Logični ukazi

ANDWF	f,d	$f \text{ and } W \rightarrow W$ ali $f \text{ and } W \rightarrow f$
IORWF	f,d	$f \text{ or } W \rightarrow W$ ali $f \text{ or } W \rightarrow f$
XORWF	f,d	$f \text{ xor } W \rightarrow W$ ali $f \text{ xor } W \rightarrow f$
ANDLW	k	$k \text{ and } W \rightarrow W$
IORLW	k	$k \text{ or } W \rightarrow W$
XORLW	k	$k \text{ xor } W \rightarrow W$
COMF	f,d	$\text{not } f \rightarrow W$ ali $\text{not } f \rightarrow f$

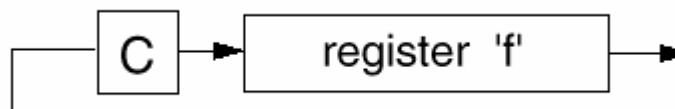
A	B	and	or	xor
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

- Ukazi rotiranja

RLF f,d



RRF f,d



- Ukazi za delo z biti

BCF	f,b	briše bit b v registru f
BSF	f,b	postavi bit b v registru f

b = 0..7

## ■ Ukazi za nadzor izvajanja programa

GOTO	k	brezpogojni skok na lokacijo k
BTFSC	f,b	testira bit b v registru f in preskoči naslednji ukaz, če je bit zbrisan (0)
BTFSS	f,b	testira bit b v registru f in preskoči naslednji ukaz, če je bit postavljen (1)
INCFSZ	f,d	izvede INCF in preskoči naslednji ukaz, če je rezultat 0
DECFSZ	f,d	izvede DECF in preskoči naslednji ukaz, če je rezultat 0
CALL	k	klic podprograma
RETURN		vrnitev iz podprograma
RETLW	k	vrnitev s konstanto
RETFIE		vrnitev iz prekinitvene rutine

- **Sistemiški ukazi**

NOP            porabi čas

CLRWDT        izbriše časovni stražnik (watch-dog timer)  
SLEEP            preklopi krmilnik v stanje pripravljenosti



# Zgledi programov

## ■ Začetek programov in inicializacija

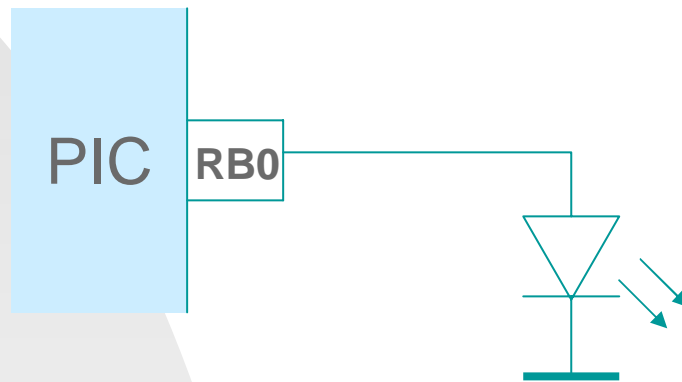
```
STATUS EQU    3        ; definicija simboličnih konstant
PORTB EQU     6
TRISB EQU     6
RP0 EQU       5        ; bita za preklop bank
RP1 EQU       6

                ORG     0h        ; premik prevajanja na začetek EPROM-a
START BSF     STATUS,RP0 ; preklop na banko 1
                BCF     STATUS,RP1

                MOVLW   0fh        ; inicializacija vodila B (00001111)
                MOVWF  TRISB ; priključki RB0-RB3 so vhodi,
                            ; priključki RB4-RB7 so izhodi
                BCF     STATUS,RP0 ; preklop na banko 0
```

## ■ Prižiganje lučke (LED)

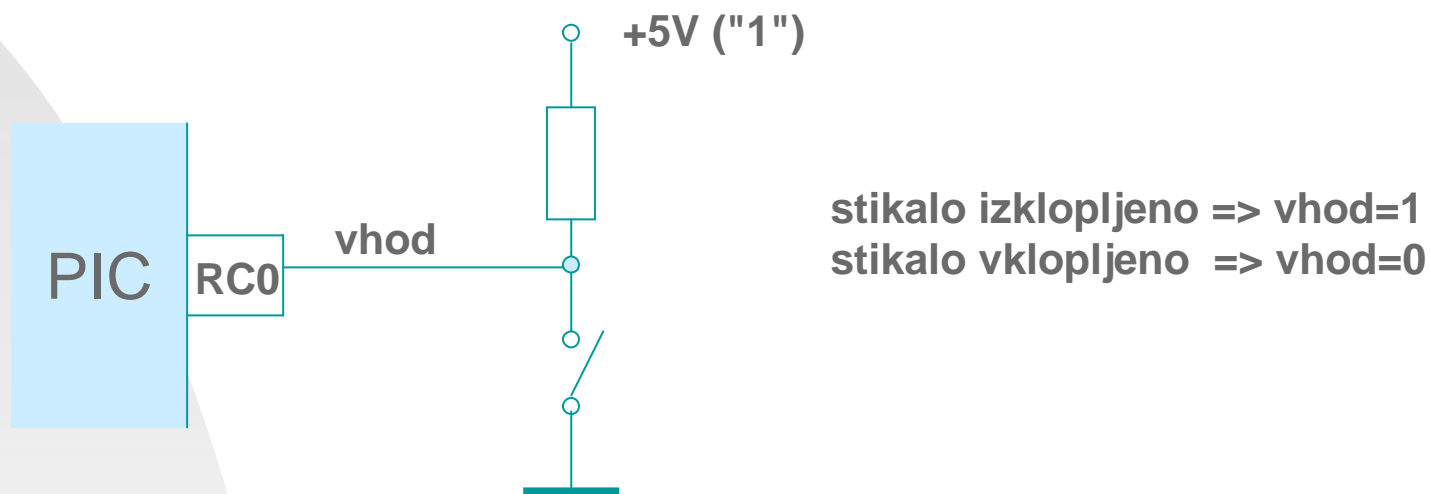
Na priključek RB0 je povezana svetleča dioda (LED)



```
PORTB EQU 6 ; definicija simboličnih konstant
TRISB EQU 6
...
START ...
    MOVLW 0 ; inicializacija vodila B
    MOVWF TRISB ; vse linije vodila B so izhodne
    ...
    BSF PORTB,0 ; prižiganje diode
    ...
    BCF PORTB,0 ; ugašanje diode
```

## ■ Testiranje stikala

Na priključek RC0 je povezano stikalo po naslednji shemi

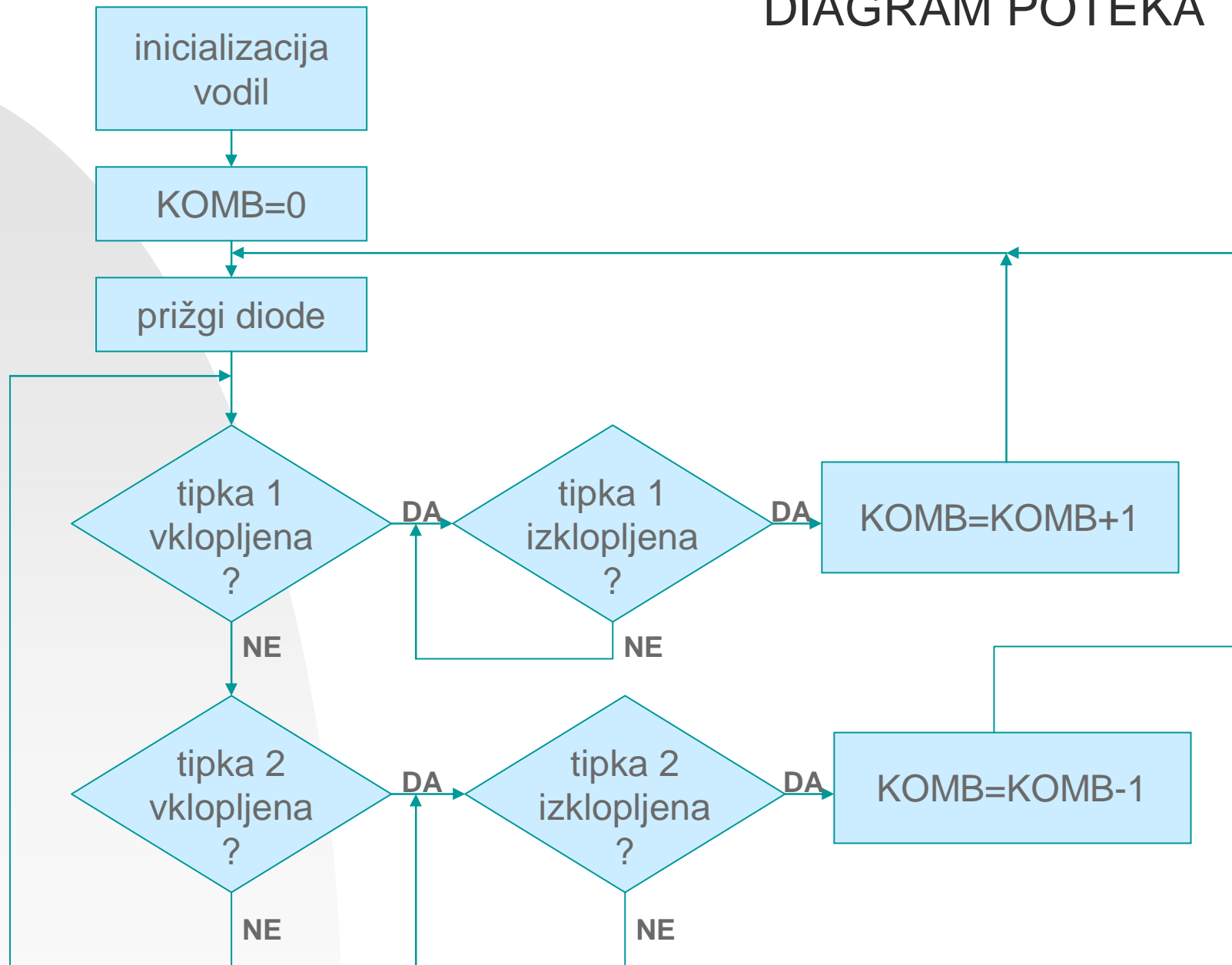


```
PORTC EQU 7 ; definicija simboličnih konstant
...
START ...
    MOVLW 0ffh ; inicializacija vodila C
    MOVWF TRISC ; vse linije vodila C so vhodne
    ...
    BTFSC PORTC,0 ; testira bit in preskoči, če je "0"
    GOTO IZKLOP ; tipka je izklopljena
VKLOP ... ; tipka je vklopljena
```

# Naloga

Na PIC priključite 8 svetlečih diod in dve tipki. Diode naj svetijo v skladu z nekim binarnim številom. Vsakokrat ko pritisnemo prvo tipko naj se binarna kombinacija poveča za ena, vsakokrat ko pritisnemo drugo tipko pa naj se za ena zmanjša. Program naj ne bo odvisen od tega kako dolgo držimo posamezno tipko.

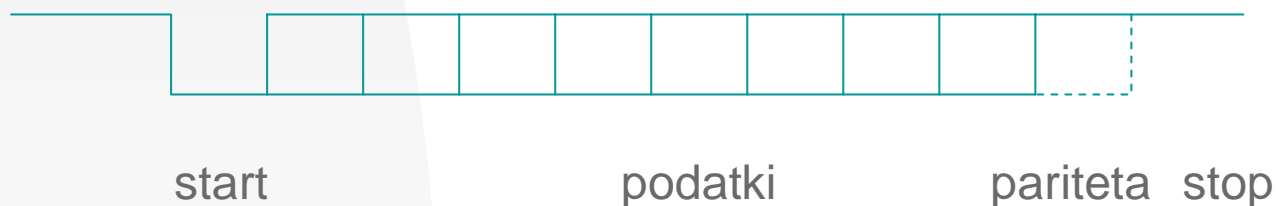
# DIAGRAM POTEKA



# USART

Univerzalni sinhroni/asinhroni sprejemnik in oddajnik

- Omogoča full duplex asinhroni prenos podatkov ter half duplex sinhroni sprejem oz. oddajo podatkov
- Omogoča 8 ali 9 bitni prenos (pariteta)
- Ima vgrajeni generator ure prenosa



## Osnovni registri

Kontrolni in statusni register oddajnika

TXSTA EQU 98h

7

0

0	TX9	TXEN	0	0	0	TRMT	TX9D
---	-----	------	---	---	---	------	------

TX9 8 bitni (0) ali 9 bitni (1) prenos

TXEN oddaja onemogočena (0) ali omogočena (1)

TRMT status oddajnega registra: 1 - prazen, 0 - poln

TX9D 9 bit podatka pri 9 bitnem prenosu

Kontrolni in statusni register sprejemnika

RCSTA EQU 18h



SPEN serijski vmesnik onemogočen (0) ali omogočen (1)  
(vpliv na priključka RC7/RX/DT in RC6/TX/CK)

RX9 onemogoči (0) ali omogoči (1) 9 bitni sprejem

CREN onemogoči (0) ali omogoči (1) sprejem

FERR napaka okvirja

OERR napaka preplavitve

RX9D 9 bit podatka pri 9 bitnem prenosu

Status sprejema

PIR1 EQU 0ch



RCIF 1 - znak je prispel 0 - znaka ni oz. se sprejema



Generator ure (Baud rate generator)

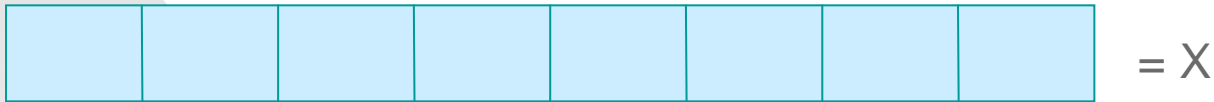
SPBRG

EQU

99h

7

0



Določitev hitrosti prenosa podatkov:

$$\text{Hitrost prenosa} = \frac{F_{\text{osc}}}{64 (X+1)}$$

Oddajni register

TXREG

EQU

19h

Sprejemni register

RCREG

EQU

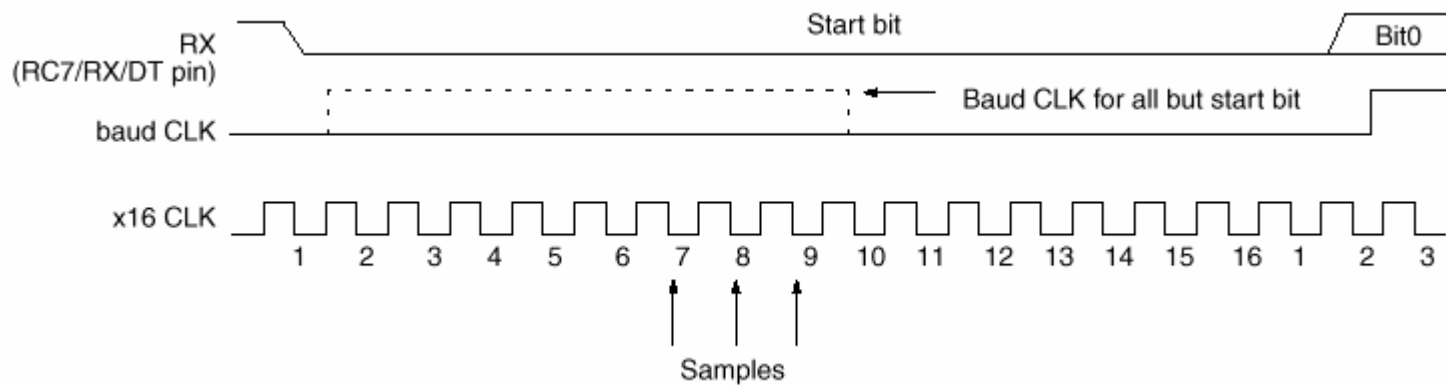
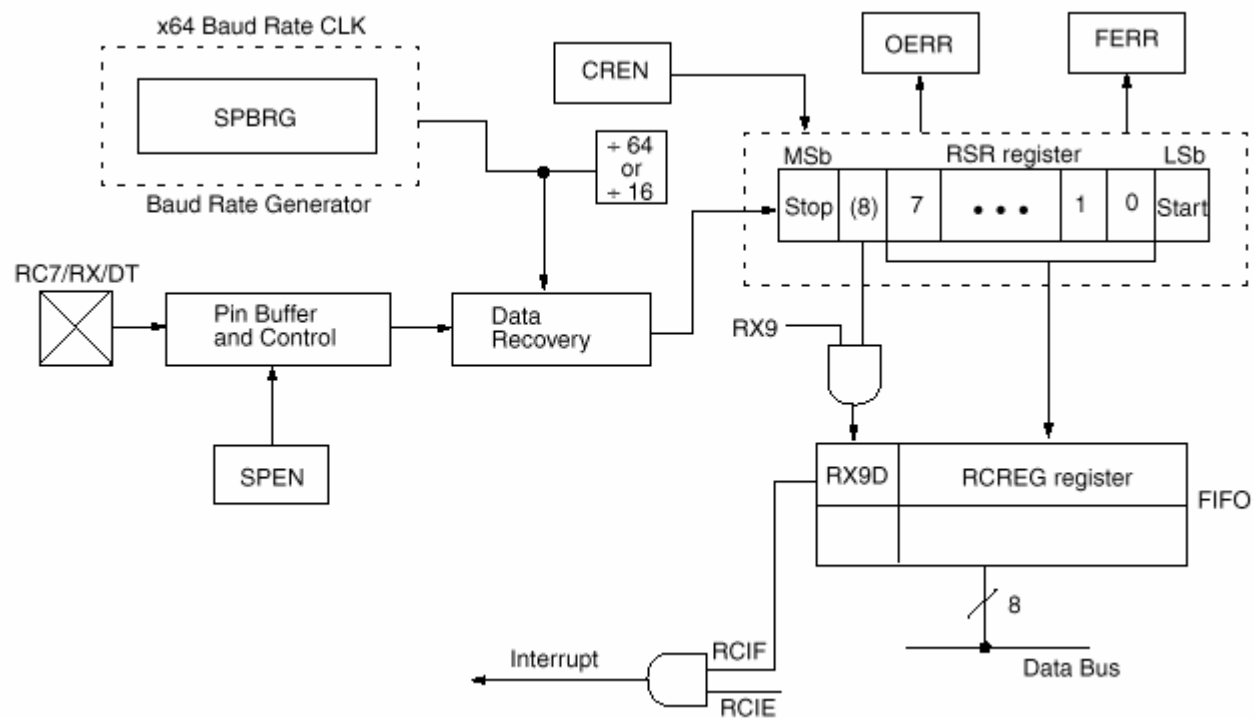
1Ah

7

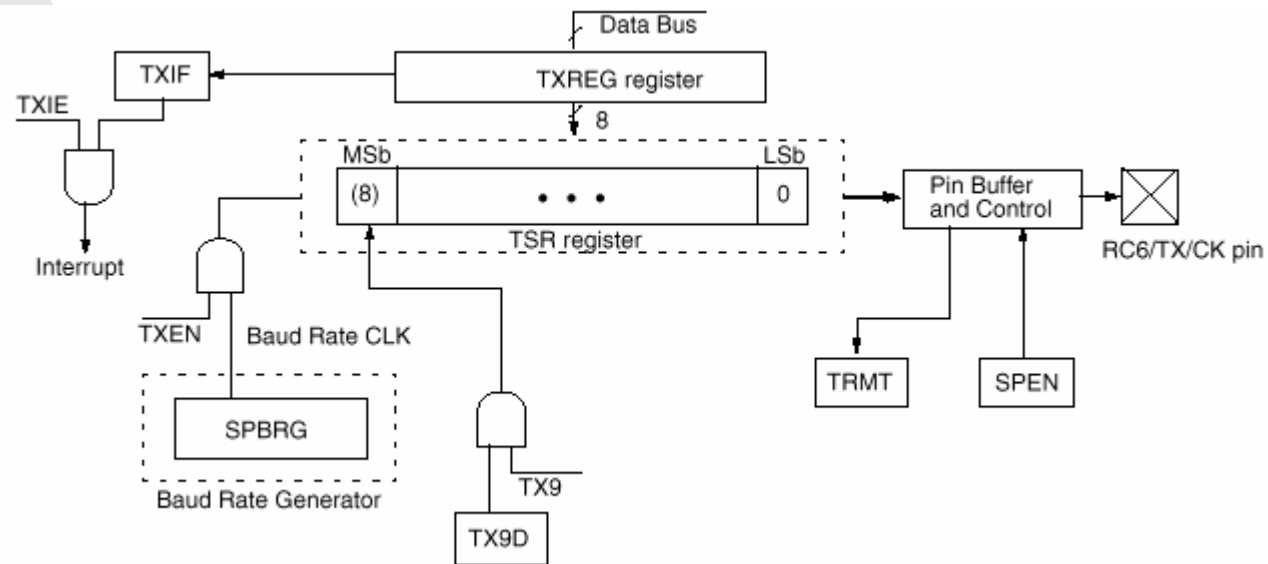
0



# Delovanje sprejemnika



## Delovanje oddajnika



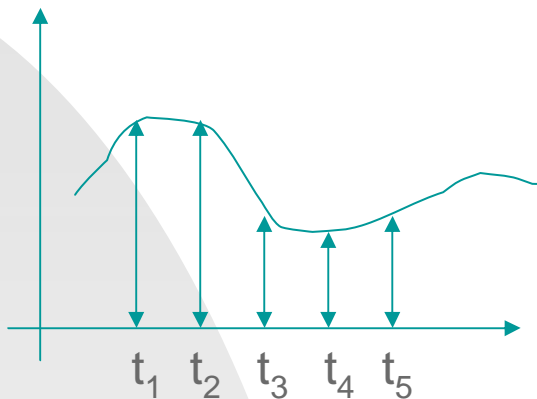
# Naloga

Izdelajte program, ki bo preko serijskega vmesnika sprejemal znake iz PC-ja. Vsak sprejeti znak naj takoj pošlje nazaj na PC, pri čemer vse velike črke zamenja z malimi in male z velikimi. Uporabite kvarčni oscilator s frekvenco 16Mhz in postavite hitrost prenosa na 19200 baudov. Podatke prenašajte brez paritete.

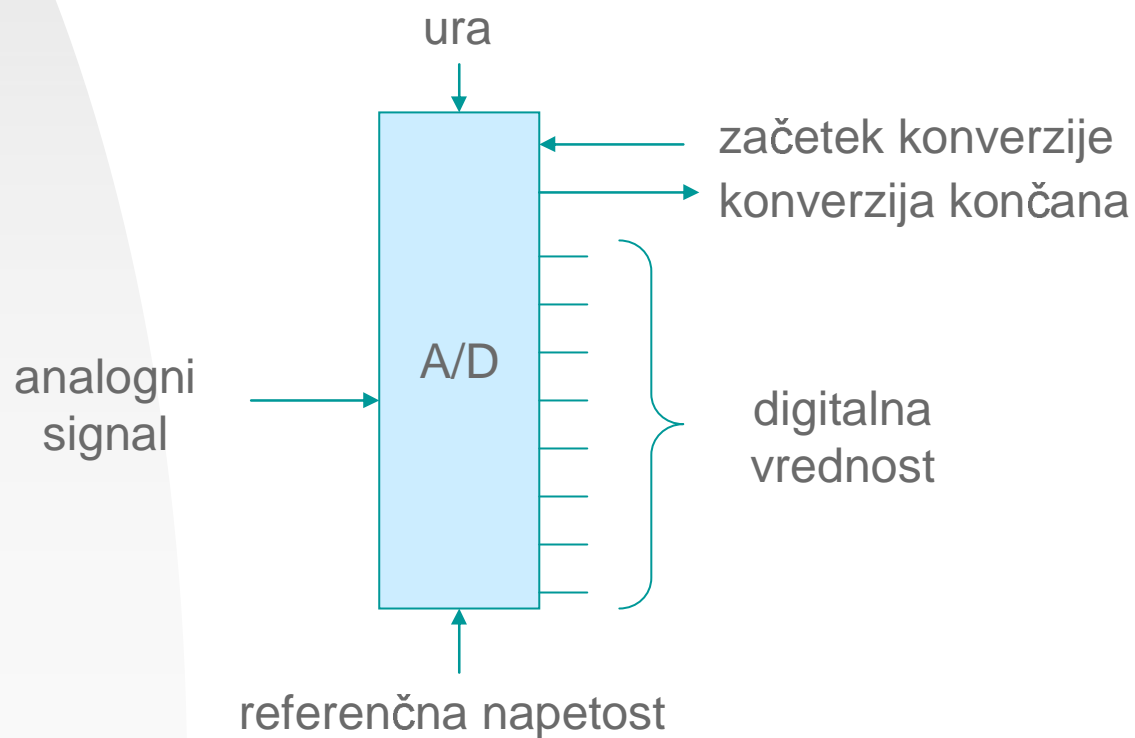
# DIAGRAM POTEKA



# Analogno/digitalni pretvorniki (8 analognih vhodov)



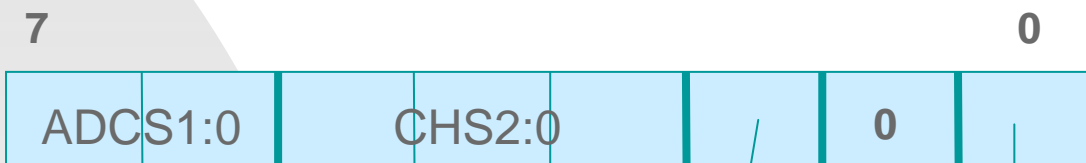
$t_1$	200
$t_2$	200
$t_3$	110
$t_4$	100
$t_5$	115



# Nabor registrov

Kontrolni register AD 0

ADCON0 EQU 1Fh



GO/DONE

ADON

ADCS1:0

ura za konverzijo:  
 00 Fosc/2  
 01 Fosc/8  
 10 Fosc/32  
 11 Interni oscilator

GO/DONE

vpis 1 začne konverzijo  
 1 - v izvajanju  
 0 - končana

CHS2:0

številka kanala:  
 000 RA0 100 RA5  
 001 RA1 101 RE0  
 010 RA2 110 RE1  
 011 RA3 111 RE2

ADON

onemogoči (0)  
 omogoči (1)  
 AD

# Kontrolni register AD 1

ADCON1 EQU 9Fh

7

0



PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	Vss	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	Vss	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	Vss	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	Vss	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	Vss	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	Vss	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0

A = Analog input D = Digital I/O  
 C / R = # of analog input channels / # of A/D voltage references

ADFM – format rezultata konverzije

0 – desno poravnano

1 – levo poravnano



Rezultat konverzije

ADRESH EQU 1Eh  
ADRESL EQU 9Eh

Rezultat A/D pretvorbe



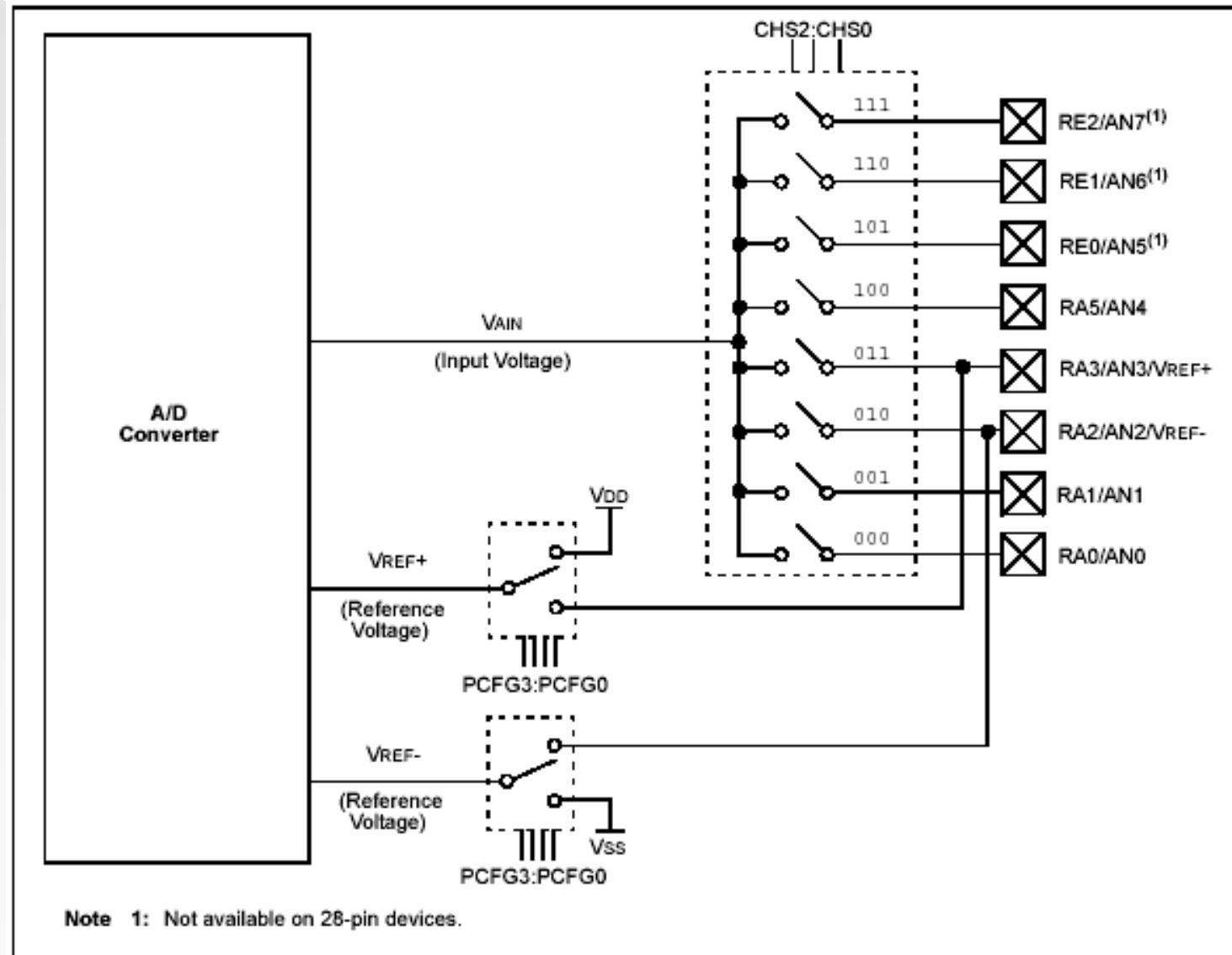
ADRESH

ADRESL









# Način povezave A/D pretvornika

FIGURE 11-1: A/D BLOCK DIAGRAM

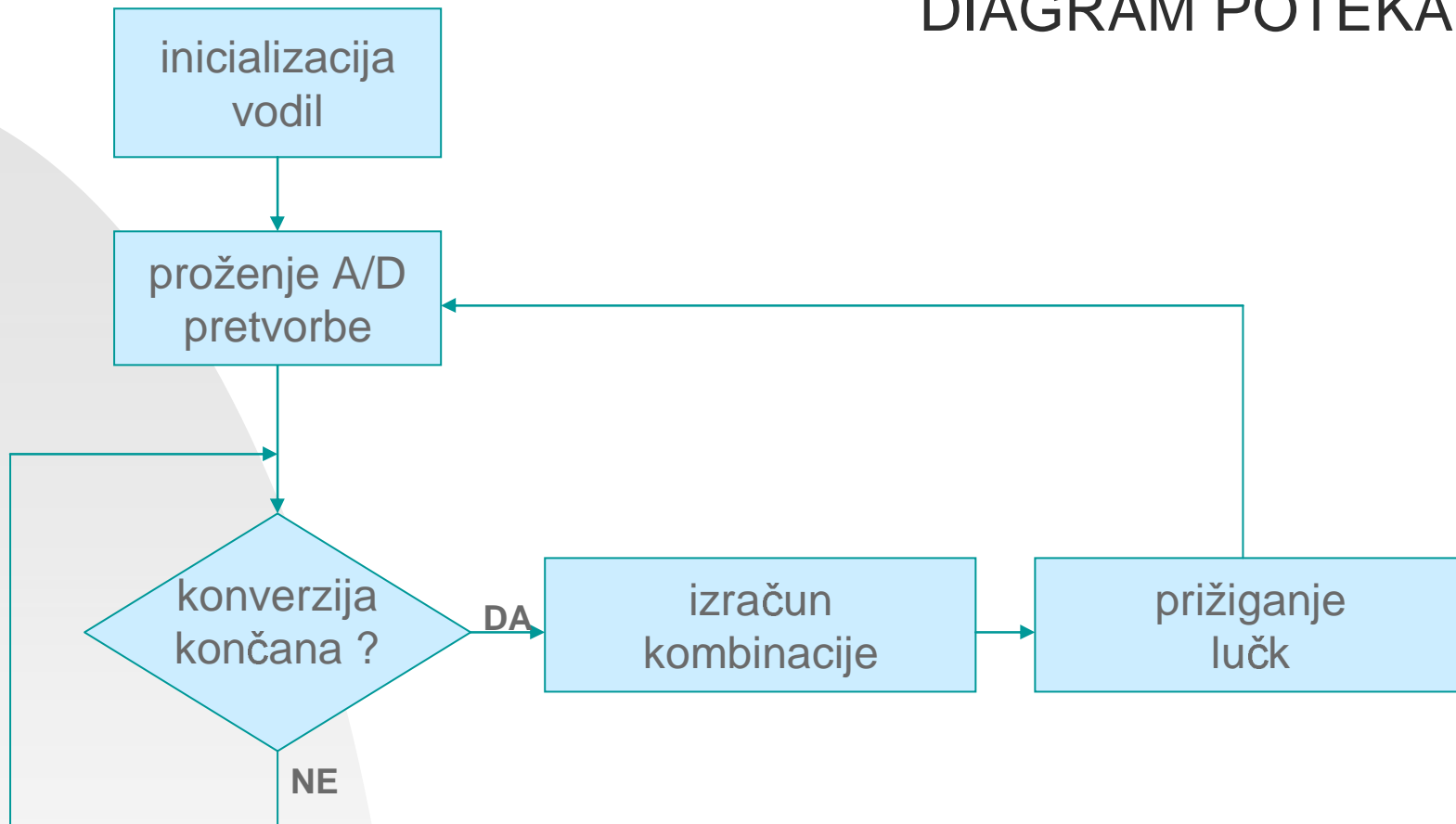


# Naloga

Izdelajte program in skonstruirajte vezje, ki bo omogočilo prikaz trenutne vrednosti vhodnega signala po principu "VU metra". To pomeni da vsaka dioda predstavlja 1/9 vhodne napetosti.

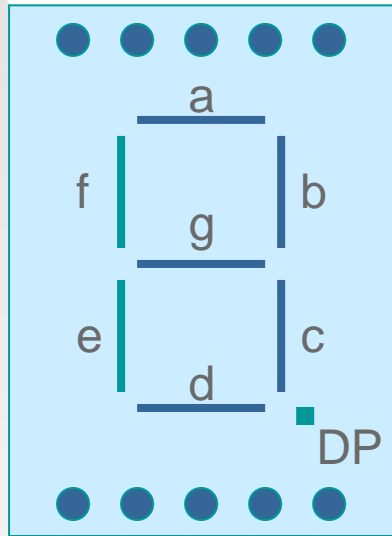
	0V	<0001110000
	0,55V	>=0001110000
	1,11V	>=0011100000
	1,66V	>=0101010000
	2,22V	>=0111000000
...		
	4,44V	>=1110000000

# DIAGRAM POTEKA

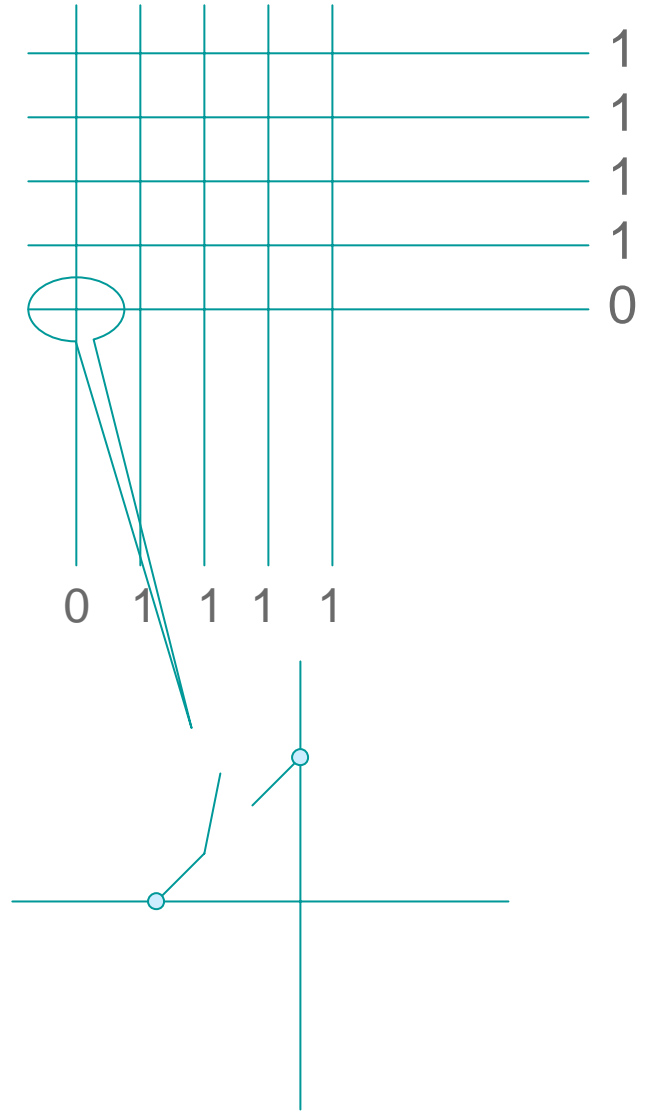


# Naloga

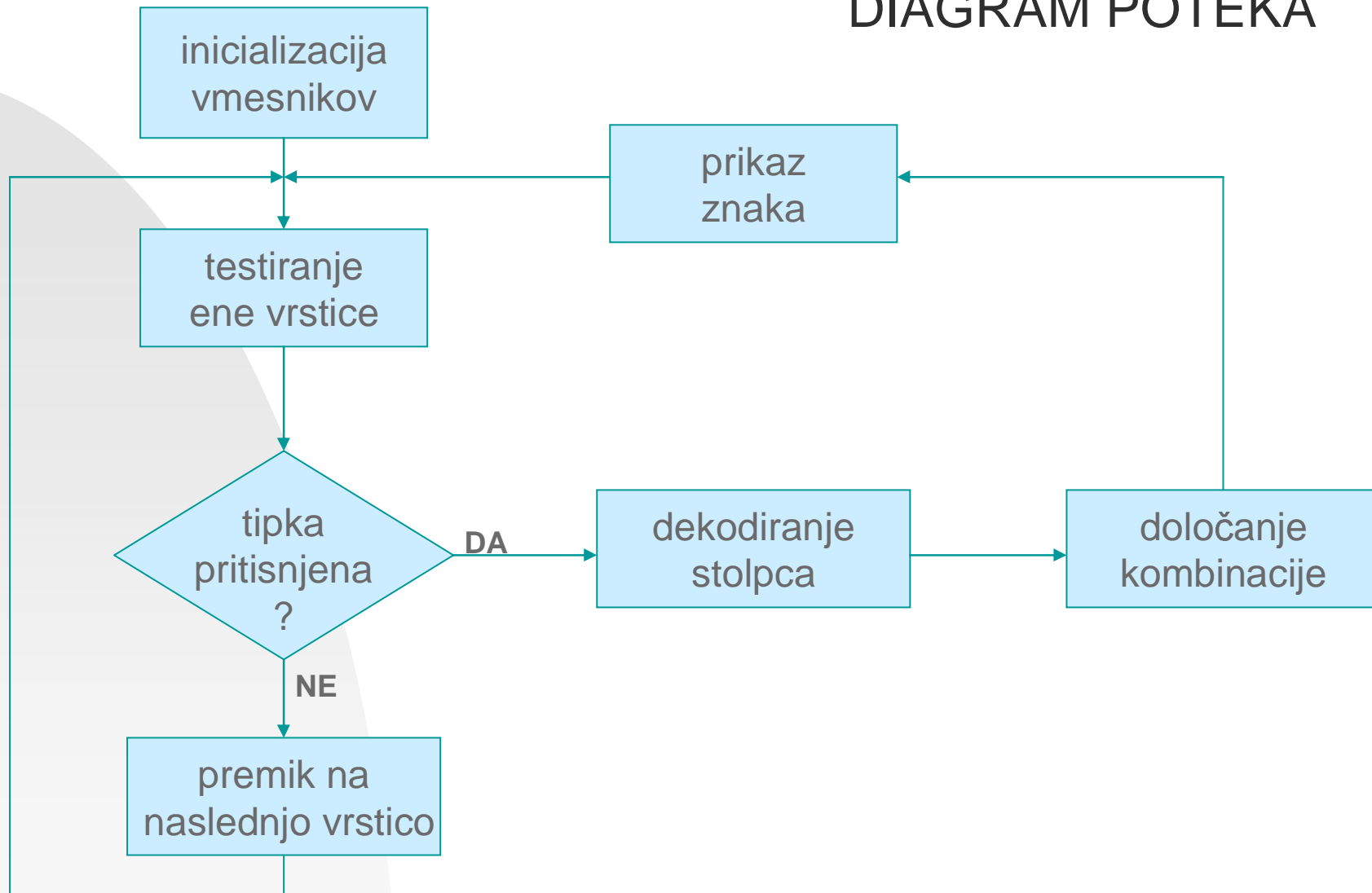
Izdelajte program s katerim boste povezali enostavno tipkovnico in 7 segmentni LED prikazovalnik. Program naj v zanki zazna katera tipka na tipkovnici je pritisnjena, jo dekodira in prikaže na LED prikazovalniku.



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



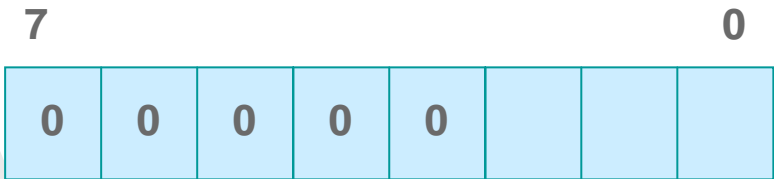
# DIAGRAM POTEKA



# Implementacija časovnika v PIC-u



OPTION



0	0	0	/2
0	0	1	/4
0	1	0	/8
0	1	1	/16
1	0	0	/32
1	0	1	/64
1	1	0	/128
1	1	1	/256

$$F = \text{Ura}/4/\text{preddelilnik}/\text{TMR0} \quad T=1/F$$



## ■ Uporaba časovnika

Želimo doseči zakasnitev za 10ms pri uri 10Mhz.

---

$$T = 10 \text{ ms} = 1/F \rightarrow F = 100 \text{ Hz}$$

$$\text{Ura} = 10 \text{ Mhz} = 10000000 \text{ Hz}$$

$$F = \text{Ura}/4/\text{preddelilnik}/\text{TMR0}$$

$$= 10000000/4/\text{preddelilnik}/\text{TMR0} = 100$$

$$\Rightarrow \text{preddelilnik}/\text{TMR0} = 25000$$

$$\text{če izberemo preddelilnik} = 256$$

$$\Rightarrow \text{TMR0} \approx 97 = 61\text{h}$$

---

```
TMR0 EQU 1 ; definicija simboličnih konstant
OPTION_REG EQU 81h
...
START MOVLW 07h ; inicializacija časovne logike
MOVWF OPTION_REG ; preddelilnik = 256
...
CRLF TMR0 ; časovnik postavimo na 0
CAKAJ MOVF TMR0,0 ; vrednost časovnika prenesemo v W
SUBLW 61h ; in odštejemo 97
BTFSS STATUS,2 ; testiramo zastavico Z
GOTO CAKAJ ; zastavica Z=0 => TMR0<97
... ; Z=1 => TMR0=97 => čas je potekel
```