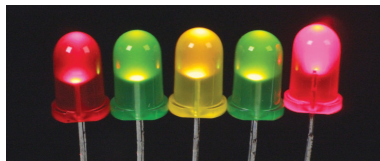


## 8 Namjenska korisnička sučelja

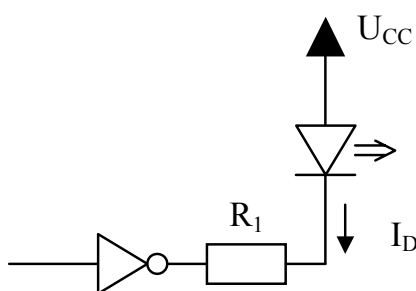
- pokazne jedinice
- jedinice za unos

### 8.1 Svjetleće diode

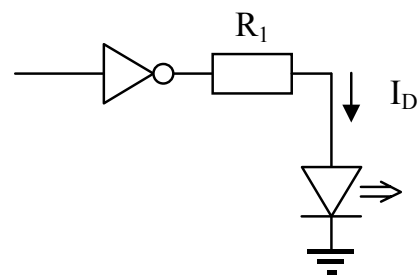
- izgled



- pogon logičkim sklopom



a)



b)

- slučaj a)

$$I_D = \frac{U_{CC} - U_D - U_{OL}}{R_1}$$

$U_{OL} \Rightarrow$  napon na izlazu logičkog sklopa u stanju 0, kod zadane struje

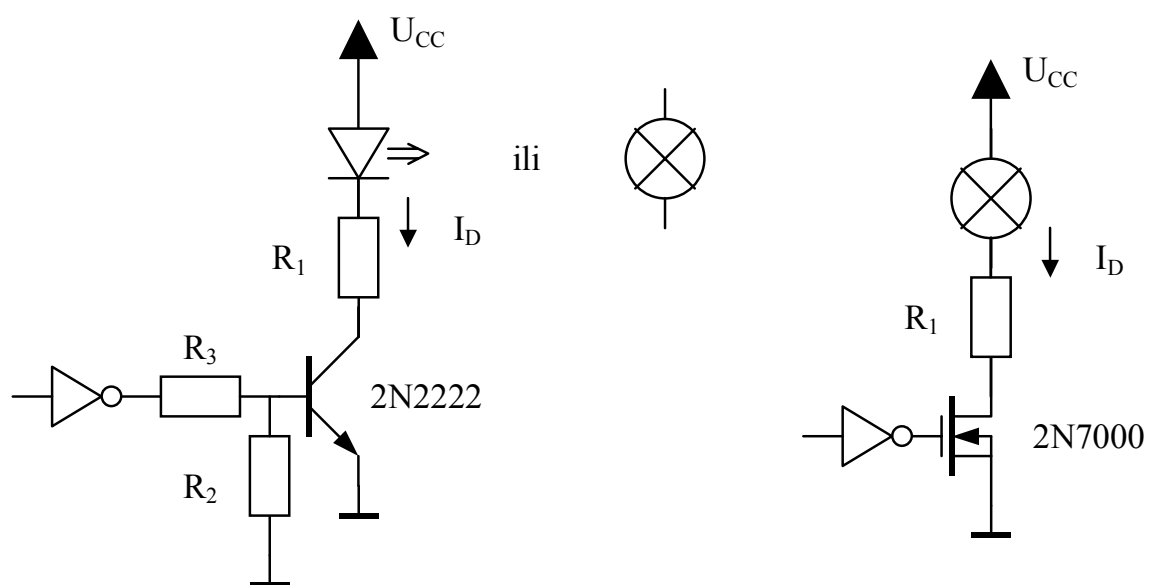
$U_D \Rightarrow$  za svjetleće diode iznosi oko 1.8V

$I_D \Rightarrow$  u praksi iznosi 3mA do 10mA

$I_{OL} \Rightarrow$  ograničenje sklopa

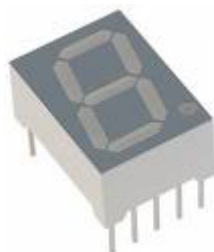
$I_{OL} > I_{OH}$

- veće struje => dodaje se tranzistor

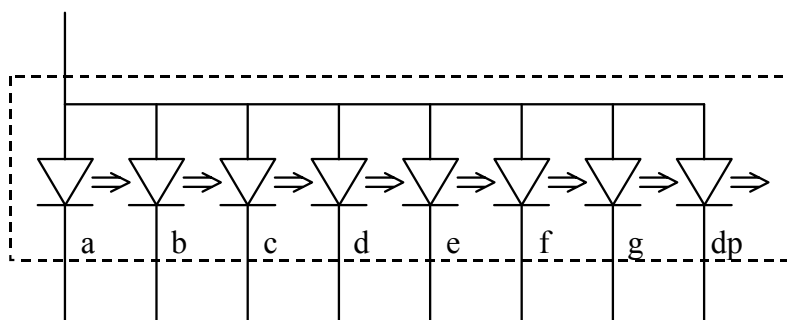


## 8.2 Osnovni način spajanja 7-segmentnih pokaznika

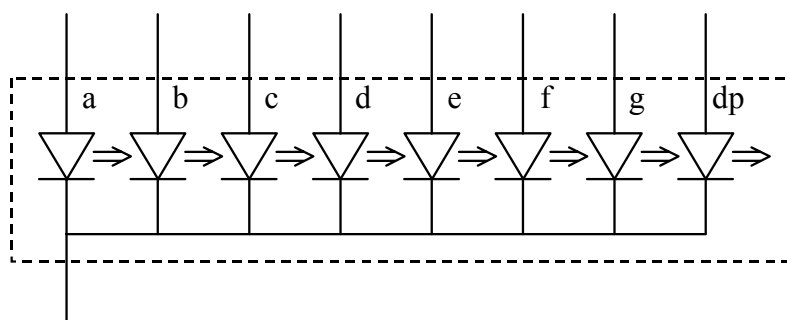
- izgled



- pokaznik sa svjetlećim diodama u spoju zajedničke anode

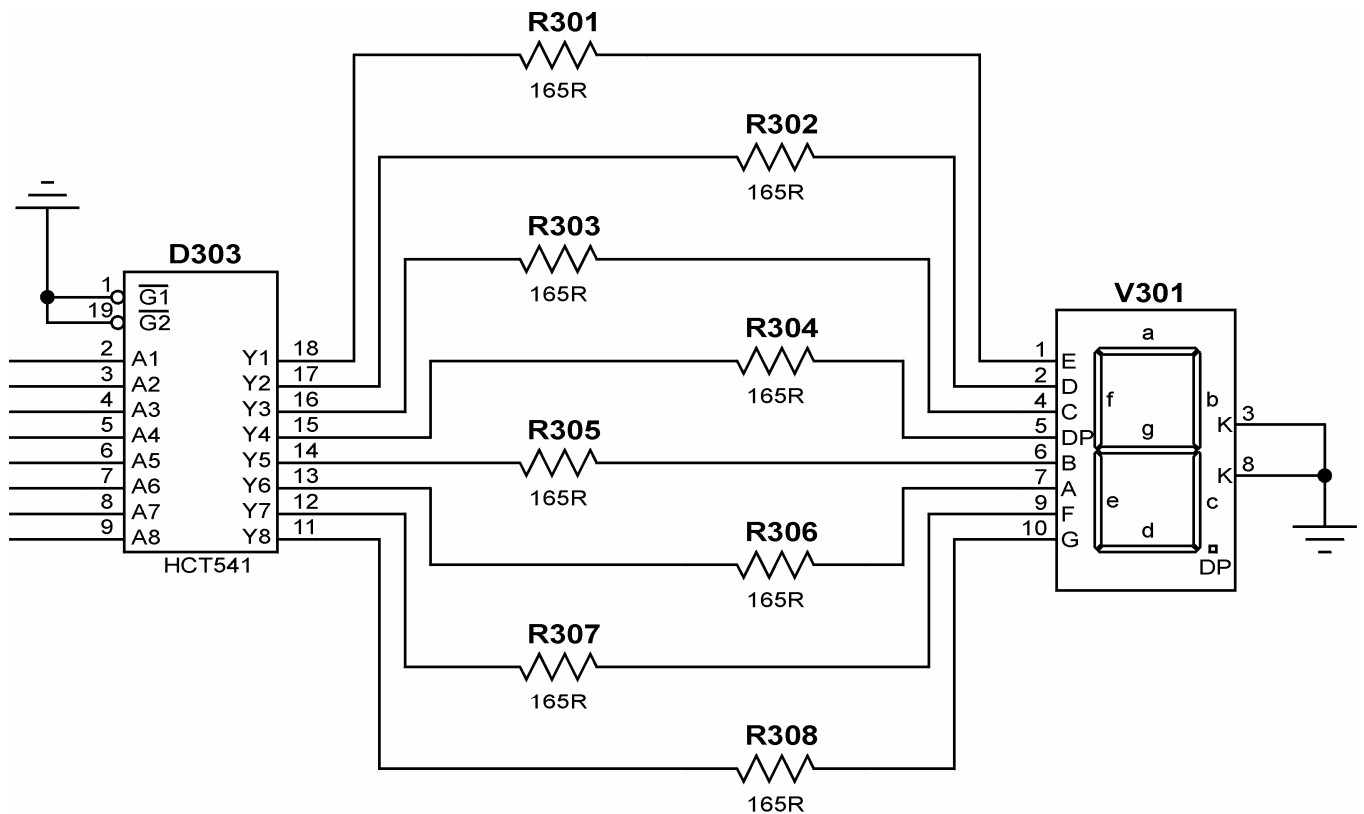


- pokaznik sa svjetlećim diodama u spoju zajedničke katode



## Primjer

- spajanje jedne znamenke indikatora

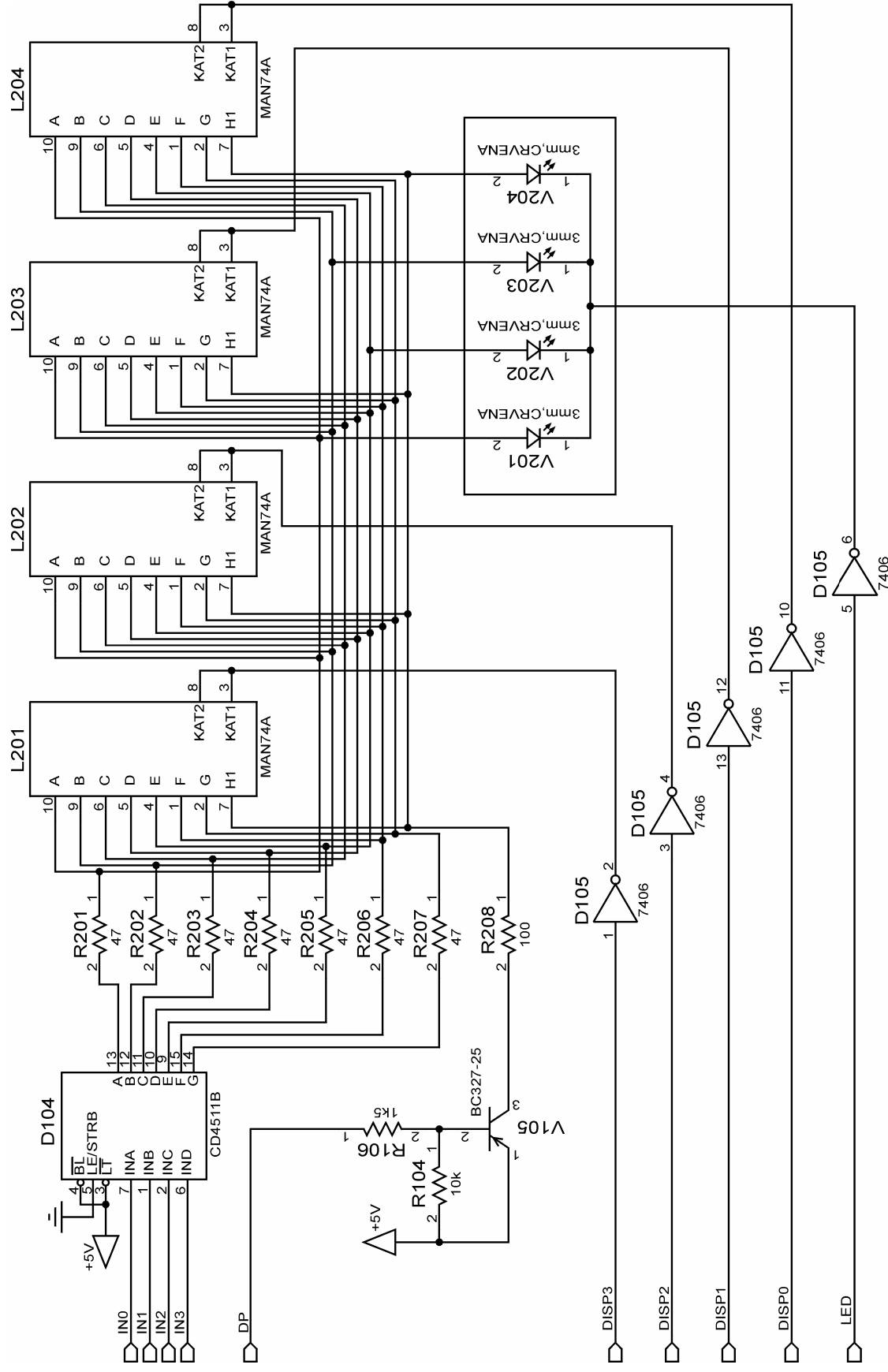


- koristi se za spoj 1 do 2 indikatora
- više od 2 indikatora  $\Rightarrow$  pogon u vremenskom multipleksu

### **8.3 Spajanja 7-segmentnih pokaznika u preklapajućem načinu rada**

- ideja
  - svaka znamenka svijetli dio vremena
  - zbog tromosti oka znamenke prividno svijetle cijelo vrijeme
- ciklus osvježavanja
  - vrijeme u kojem se redom uključe sve znamenke
- frekvencija osvježavanja
  - ne smije se vidjeti treperenje
  - ne smije se potrošiti previše procesorskog vremena
  - tipična frekvencija osvježavanja je 50-75 Hz

## Primjer

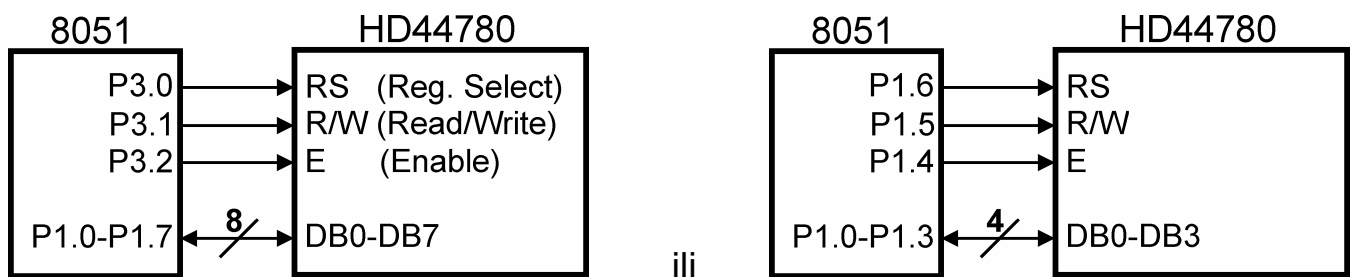


## 8.4 LCD pokaznik

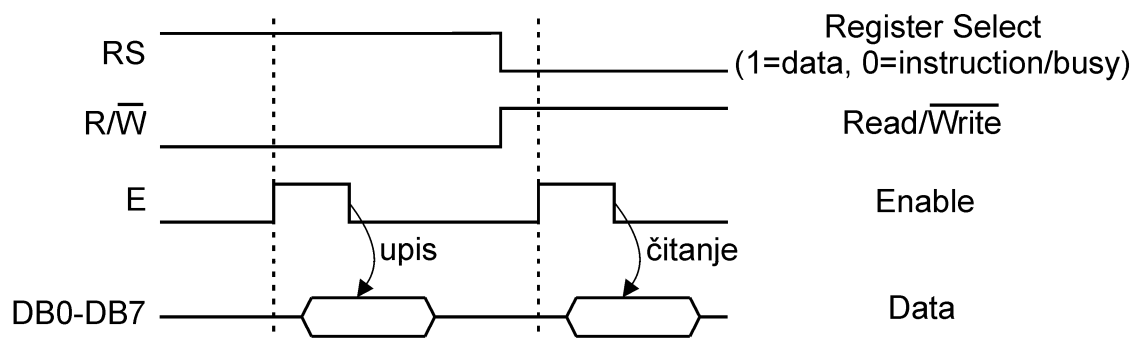
- izgled



- sadrže kontroler, npr. Hitachi HD44780
- primjer spajanja pokaznika s mikrokontrolerom 8051



- primjer komunikacije s pokaznikom



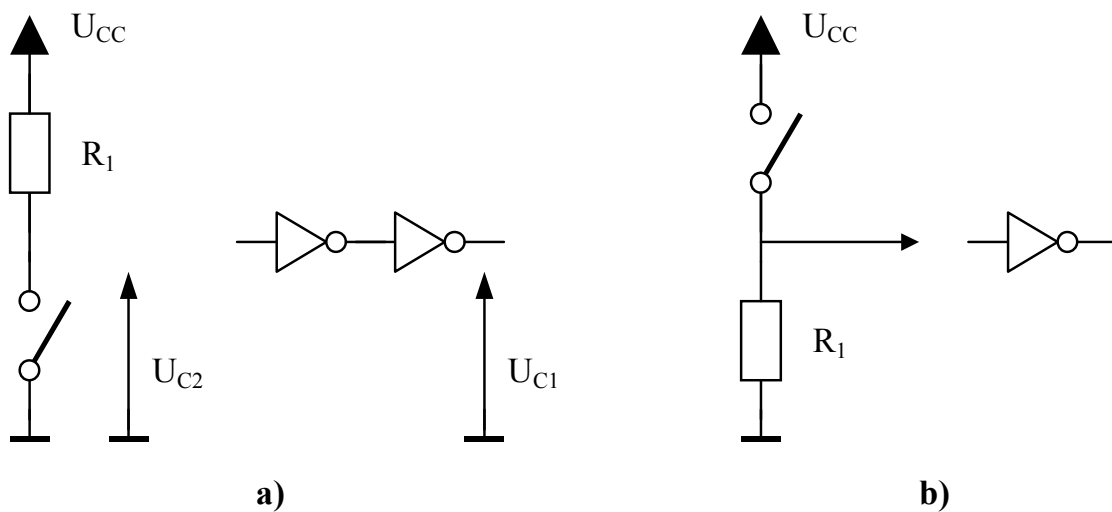
- komande
  - ⇒ upis znaka, brisanje pokaznika, pomicanje kursora, itd.
- kako programski emulirati sučelje
  - ⇒ vidi sličan primjer u daljnjim poglavljima (kod I2C sabirnice)

## 8.5 Spajanje tipke

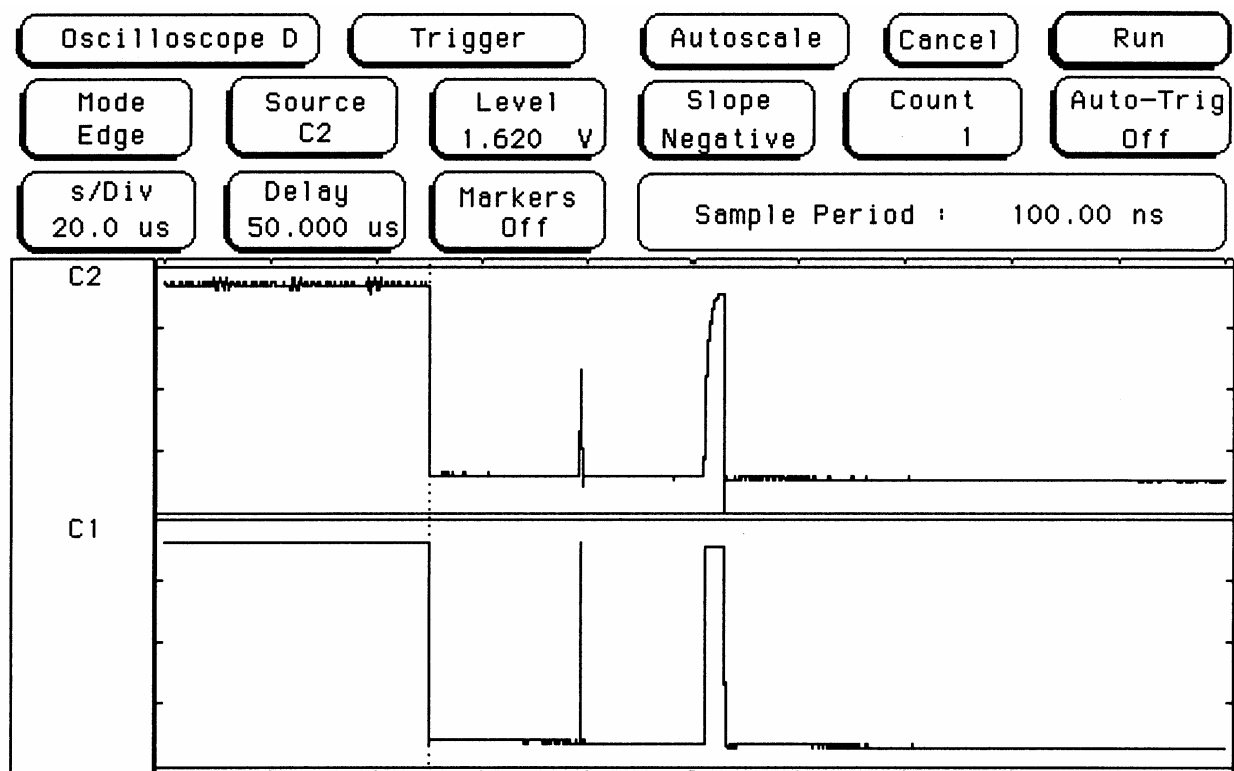
- izgled



- osnovni spoj tipke

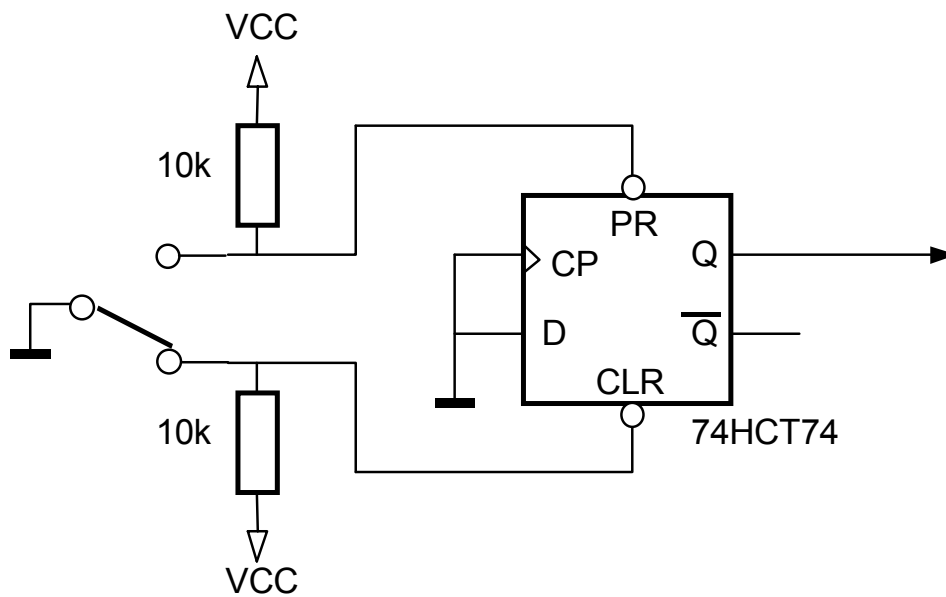


- "istitrvanje"

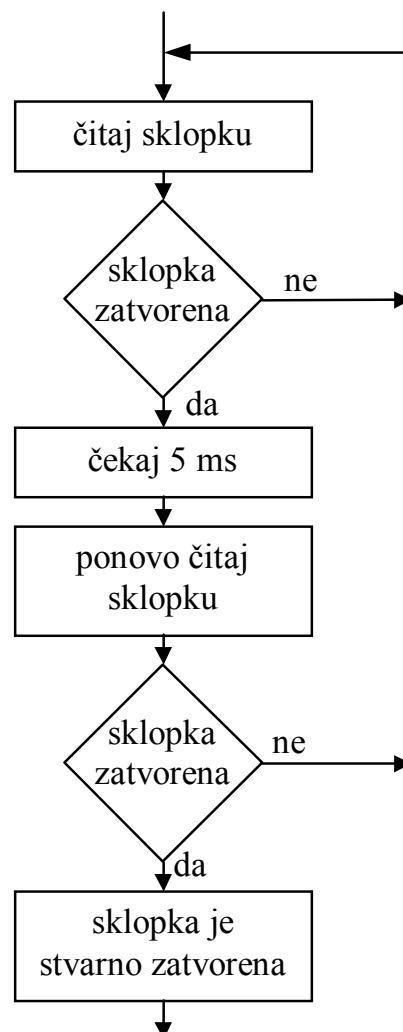




- uklanjanje istitravanja sklopke pomoću bistabila



- Programsko uklanjanje istitravanja

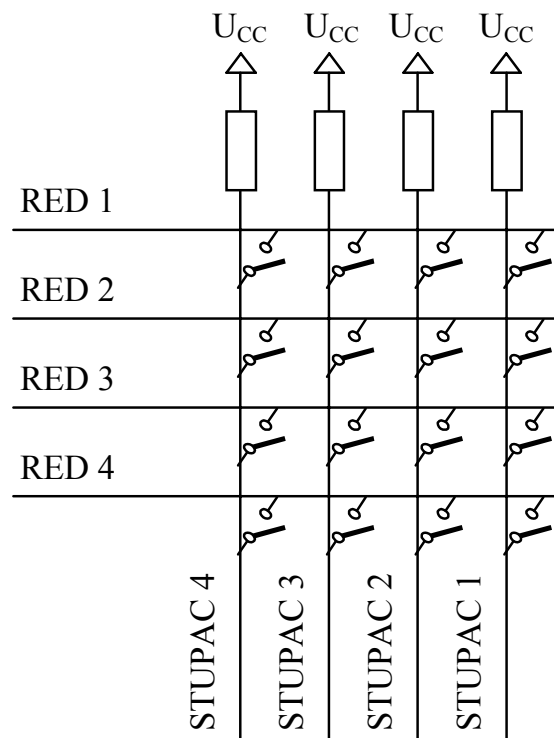


## 8.6 Matrična tipkovnica

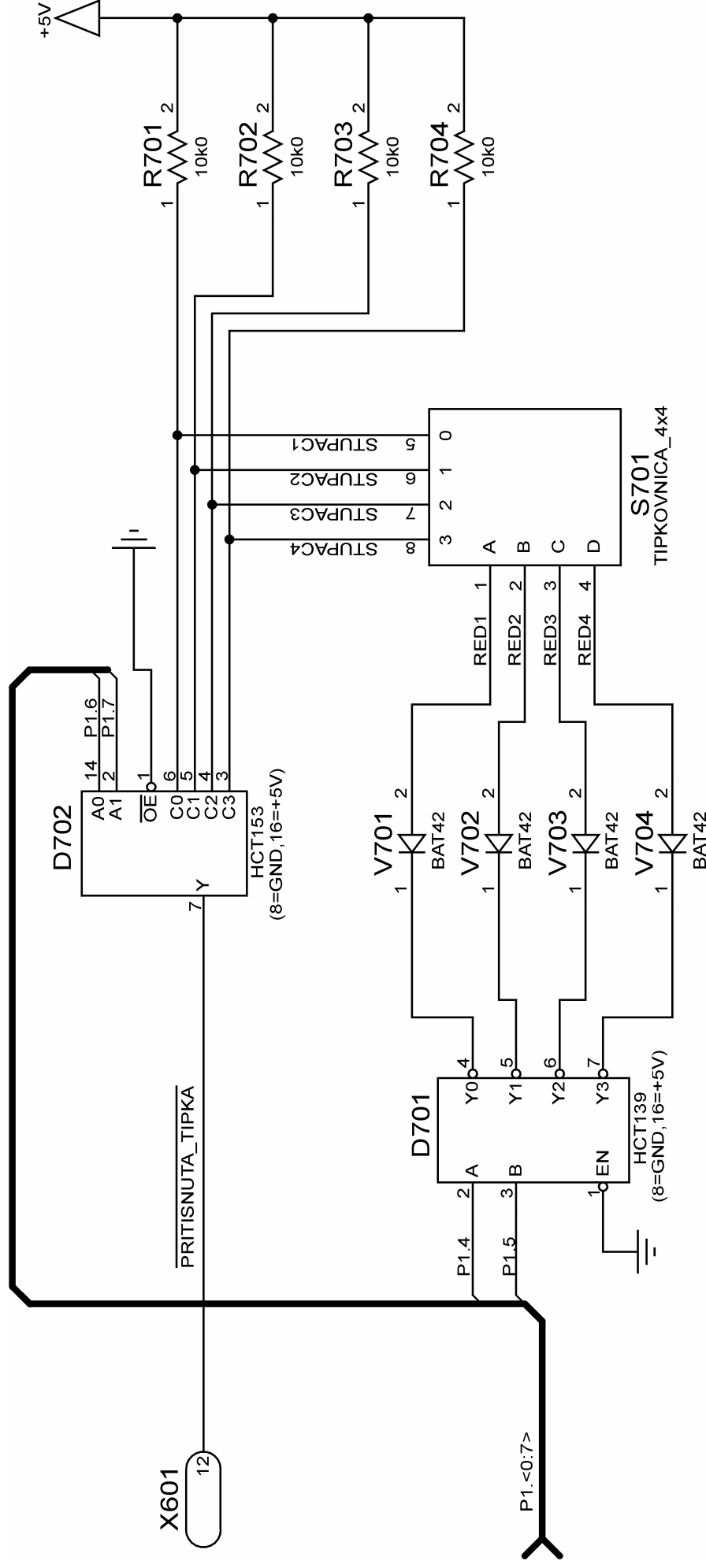
- izgled



- spoj 16 tipki u matrici



## 8.7 Primjer spajanja matične tipkovnice



```

/*=====*/
/*                                PRIMJER CITANJA TIPKOVNICE                                */
/*=====*/

#include <reg51.h>
#include <stdio.h>

void uzmi_tipku(void); /* prekidna rutina za pretrazivanje      */
/* tipkovnice (prekid uzrokuje brojilo T0) */

sbit P3_4=0x0B0^4;      /* signal koji kaze da je tipka pritisnuta */
/* (vidi elektricnu shemu) */

unsigned char data prosli_put=0xFF; /* pomocna varijabla za "debouncing" */
unsigned char data tipka=0xFF; /* varijabla koja sadrzi kod pritisnute tipke */

main()
{
/* inicijalizacija serije: */

    TMOD=0x21;          /* brojilo T0 u nacin rada 1 (za prekid) */
                        /* brojilo T1 u nacin rada 2 (za seriju) */
    TH1=0xF4;           /* preset vrijednost za brojila T1, 2400 bitova/s */
    TR1=1;              /* brojilo T1 enable */
    SCON=0x52;          /* 0 - SM0 Serial Mode 1 */
                        /* 1 - SM1 Serial Mode 1 */
                        /* 0 */
                        /* 1 - REN Receive ENable */
                        /* 0 */
                        /* 0 */
                        /* 1 - TI Transmit Interrupt flag */
                        /* 0 */

/* inicijalizacija brojila T0 (tipkovnica) */

    TH0=0x4B;           /* brojilo T0 broji od 0x4BFF do 0xFFFF => cca 50 ms */
    TL0=0xFF;
    TR0=1;              /* brojilo T0 enable */

/* prekidi */

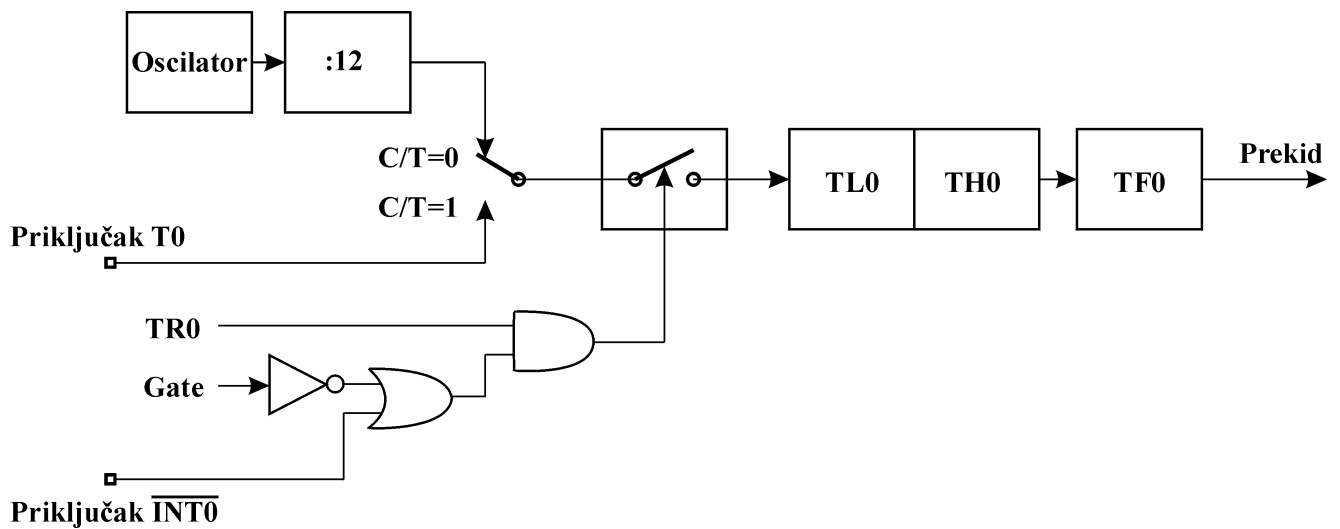
    ET0=1;              /* omoguci prekid brojila T0 */
    EA=1;              /* omoguci prekide (global interrupt enable) */

/* ispis tipke na seriju */

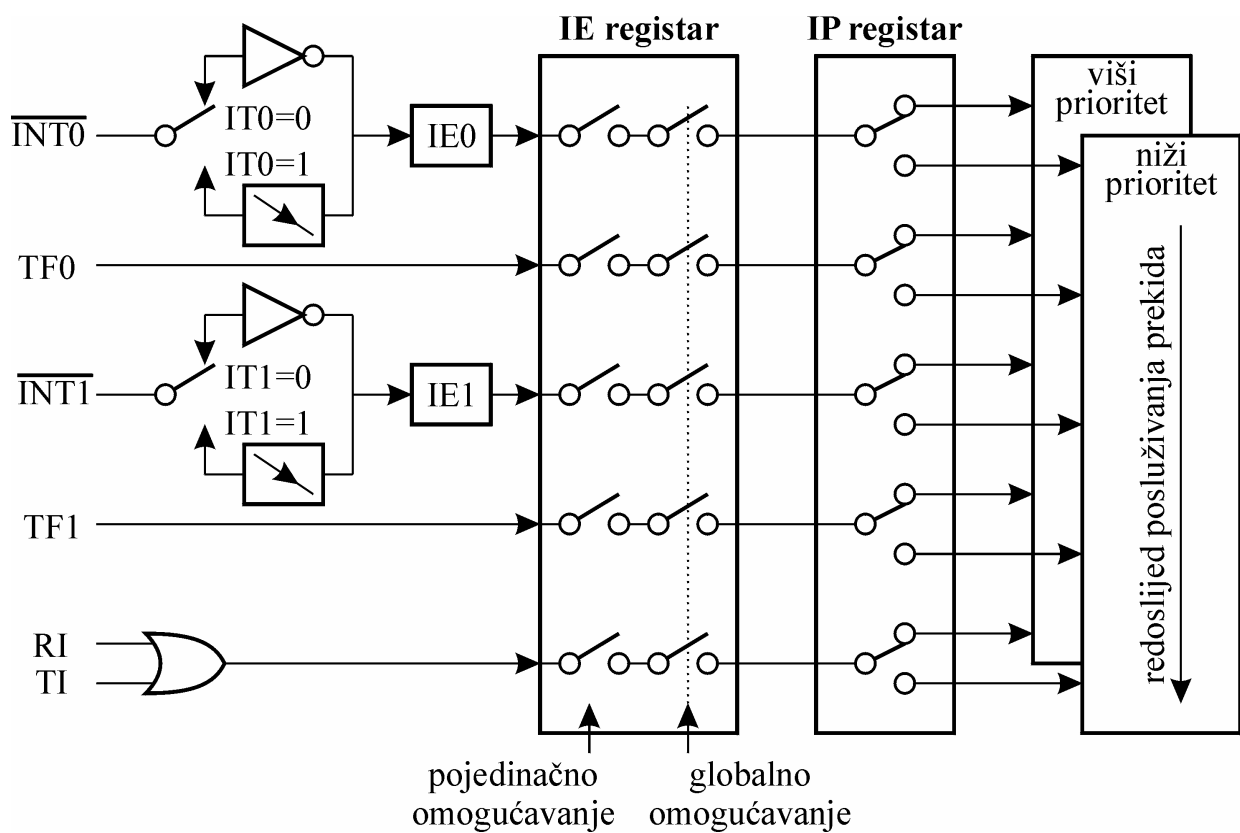
    while(1)
    {
        while(tipka==0xFF); /* cekaj dok se pritisne neka tipka */
        printf("\n Tipka=%u", (unsigned int)tipka);
        while(tipka!=0xFF); /* cekaj dok se tipka otpusti */
    }
}

```

- Brojilo T0  $\Rightarrow$  način rada 1



- sustav za upravljanje prekidima



```
/*=====*/
/*          PREKIDNA RUTINA ZA PRETRAZIVANJE TIPKOVNICE          */
/*=====*/

void uzmi_tipku(void) interrupt 1 using 1
{
    unsigned char data i;          /* brojac za pretrazivanje tipkovnice */
    unsigned char data priprema; /* pomocna varijabla          */

    TH0=0x4B; /* ponovno punjenje brojila T0 koje radi u nacinu rada 1 */
    TL0=0xFF;

    /* Pronadi da li je pritisnuta neka tipka */

    for(i=0;i<16;i++)
    {
        priprema=i<<4;          /* stavi broj tipke na gornja 4 bita */
        priprema=priprema+0x0F; /* neka donja 4 bita budu 1      */

        P1=priprema;

        if(P3_4==0)             /* da li je prozvana tipka pritisnuta ? */
        {
            if(prosli_put==i) tipka=i; /* Ako je tipka bila pritisnuta i */
                                     /* prije 50 ms onda je stvarno pritisnuta */
                                     /* a ne radi se o istitravanju.      */

            prosli_put=i;        /* Zapamti koja je tipka pritisnuta */
                                /* (za slijedeci prolaz prekidne rutine) */

            return;
        }
    }

    /* Slucaj kad niti jedna tipka nije pritisnuta */

    if(prosli_put==0xFF) tipka=0xFF; /* Ako tipka nije bila pritisnuta */
                                    /* prije 50 ms onda stvarno nije pritisnuta */
                                    /* a ne radi se o istitravanju.      */

    prosli_put=0xFF;                /* Zapamti da nije pritisnuto nista */
                                    /* (za slijedeci prolaz prekidne rutine) */

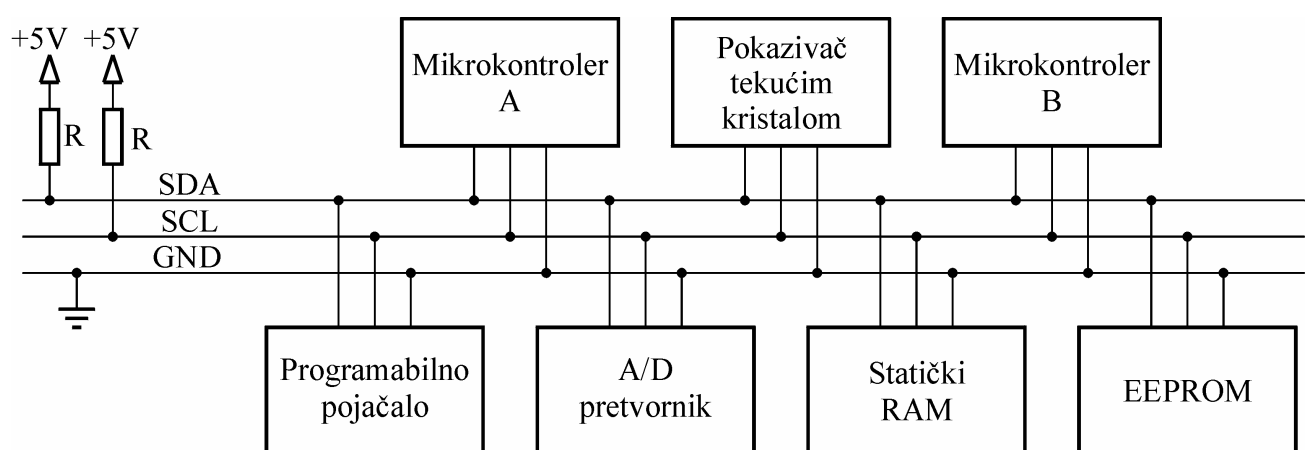
}
```

## 9 I<sup>2</sup>C sabirnica

- Inter IC => IIC => I<sup>2</sup>C
- povezuje dijelove sustava:
  - npr.  $\mu$ C, LCD indikatori, EEPROM, RAM, TV komponente i sl.
- sinkrona, dvosmjerna, serijska sabirnica

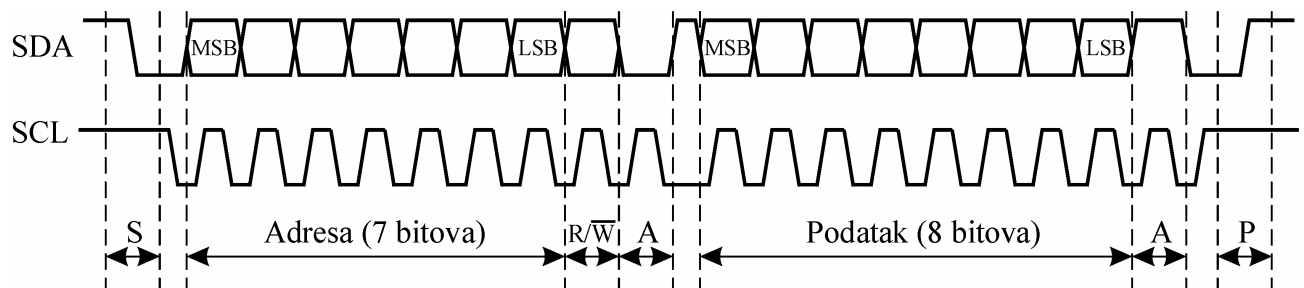
### 9.1 Opis I<sup>2</sup>C sabirnice

- linije SDA, SCL
- otvoreni kolektor
- brzina prijenosa
  - standardni način rada => do 100 kbit/s
  - brzi način rada => do 400 kbit/s
- ne treba "glue logic"
- najveći broj jedinica na sabirnici
  - ograničen jedino kapacitetom na liniji (najviše 400 pF)



## 9.2 Opis komunikacije

- pojmovi
  - upravljač (master) - izvršilac (slave)
  - prijemnik (receiver) - odašiljač (transmitter)
  - arbitracija
- prijenos podataka
  - start bit
  - stop bit
  - 7 (10) bitna adresa
  - read/write
  - potvrda (acknowledge)





### 9.3 Protokol kod čitanja i pisanja

- upravljač piše u izvršioca

	adresa	R/ $\overline{W}$	A	podatak	A	podatak	A	
S	1010110	0	0	10101101	0	01001101	0	P



upravljač (*master*)



izvršilac (*slave*)

- upravljač čita iz izvršioca

	adresa	R/ $\overline{W}$	A	podatak	A	podatak	A	
S	1010110	1	0	10101101	0	01001101	1	P



upravljač (*master*)



izvršilac (*slave*)

- kombinacija čitanja i pisanja

	adresa	R/ $\overline{W}$	A	podatak	A		adresa	R/ $\overline{W}$	A	podatak	A	
S	1010110	0	0	10101101	0	S	1010110	1	0	01001101	1	P



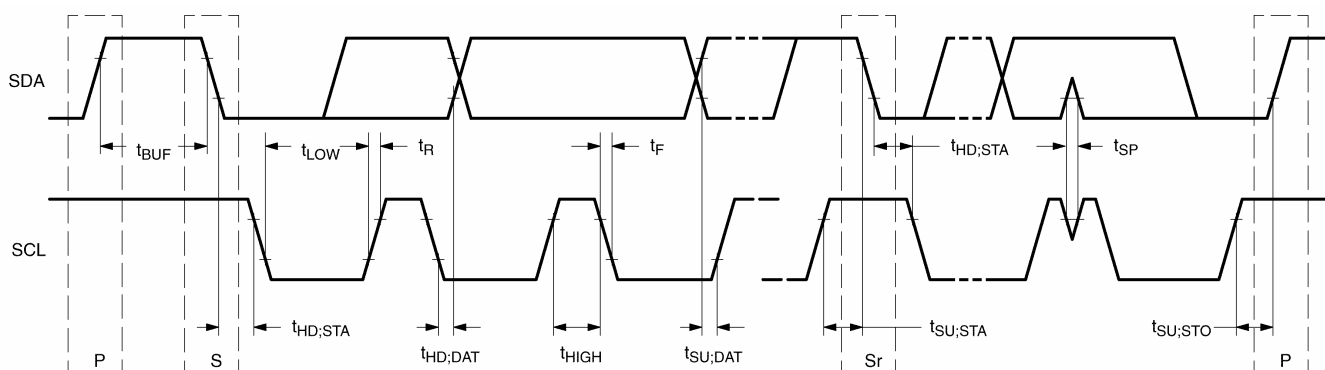
upravljač (*master*)



izvršilac (*slave*)

## 9.4 Električke specifikacije I<sup>2</sup>C sabirnice

- električke specifikacije neke sabirnice obuhvaćaju:
  - opis signala
  - naponske i strujne karakteristike, zaključenja i sl.
  - vremenske dijagrame



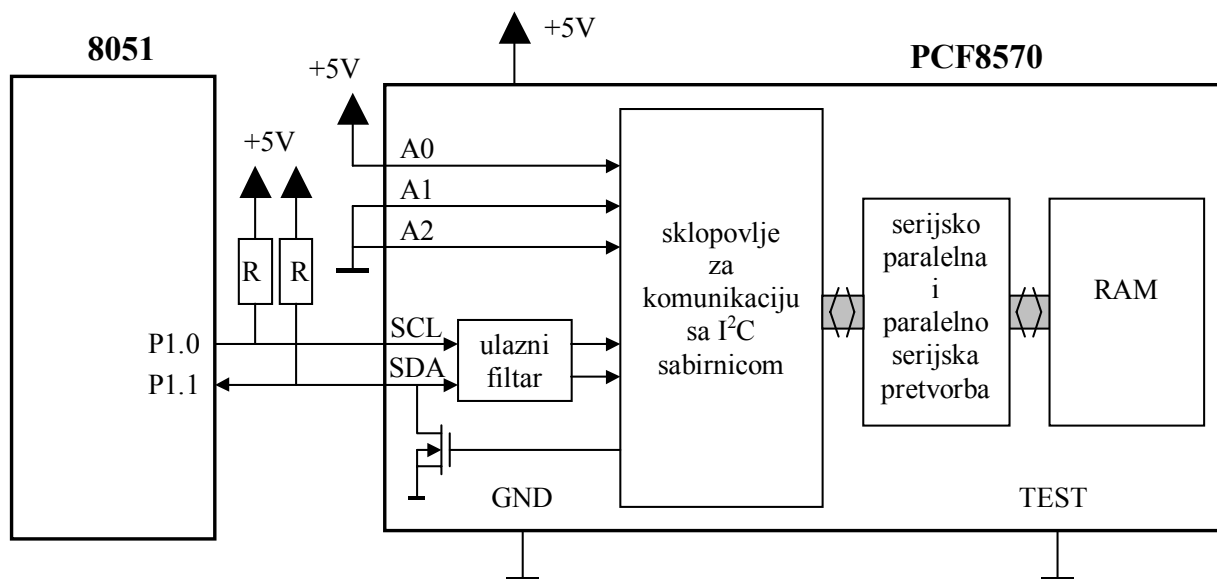
PARAMETER	SYMBOL	STANDARD-MODE I <sup>2</sup> C-BUS		FAST-MODE I <sup>2</sup> C-BUS		UNIT
		Min.	Max.	Min.	Max.	
SCL clock frequency	$f_{SCL}$	0	100	0	400	kHz
Bus free time between a STOP and START condition	$t_{BUF}$	4.7	–	1.3	–	$\mu s$
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD;STA}$	4.0	–	0.6	–	$\mu s$
LOW period of the SCL clock	$t_{LOW}$	4.7	–	1.3	–	$\mu s$
HIGH period of the SCL clock	$t_{HIGH}$	4.0	–	0.6	–	$\mu s$
Set-up time for a repeated START condition	$t_{SU;STA}$	4.7	–	0.6	–	$\mu s$
Data hold time: for CBUS compatible masters (see NOTE, Section 9.1.3) for I <sup>2</sup> C-bus devices	$t_{HD;DAT}$	5.0 0 <sup>1)</sup>	– –	– 0 <sup>1)</sup>	– 0.9 <sup>2)</sup>	$\mu s$ $\mu s$
Data set-up time	$t_{SU;DAT}$	250	–	100 <sup>3)</sup>	–	ns
Rise time of both SDA and SCL signals	$t_r$	–	1000	$20 + 0.1C_b^{4)}$	300	ns
Fall time of both SDA and SCL signals	$t_f$	–	300	$20 + 0.1C_b^{4)}$	300	ns
Set-up time for STOP condition	$t_{SU;STO}$	4.0	–	0.6	–	$\mu s$
Capacitive load for each bus line	$C_b$	–	400	–	400	pF

## 9.5 Primjer spajanja sklopa na I<sup>2</sup>C sabirnicu

- spajanje I2C RAM memorije s mikrokontrolerom 8051
- I2C RAM PCF8570 (Philips)
  - 256 bajtova
  - kućište s 8 priključaka (DIL 8 i SO8L)
  - radi u standardnom načinu rada (brzina manja od 100 kbit/s)

### 9.5.1 I<sup>2</sup>C sklopovlje

- shema



- adresa

1	0	1	0	A2	A1	A0
---	---	---	---	----	----	----

- - iznos otpora otpornika za zaključanje sabirnice

$$R = \frac{t_r}{C_{BUS}}$$

## 9.5.2 Programska emulacija I<sup>2</sup>C sučelja

### Primjer

- Protokol za upis dva bajta u RAM

	adresa RAM-a	$\overline{R/\overline{W}}$	A	početna adresa u RAM-u	A	prvi podatak	A	drugi podatak	A	
S	1010001	0	0	00000010	0	10100000	0	10101111	0	P



8051



PCF8570

- Protokol za čitanje dva bajta iz RAM-a

	adresa RAM-a	$\overline{R/\overline{W}}$	A	početna adresa u RAM-u	A		adresa RAM-a	$\overline{R/\overline{W}}$	A	prvi podatak	A	drugi podatak	A	
S	1010001	0	0	00000010	0	S	1010001	1	0	10100000	0	10101111	1	P



8051



PCF8570

```

/*****
/*          PROGRAMSKA EMULACIJA I2C SUCELJA          */
*****/

```

```

#define I2CADRESA 0x51      /* I2C adresa memorije */

```

```

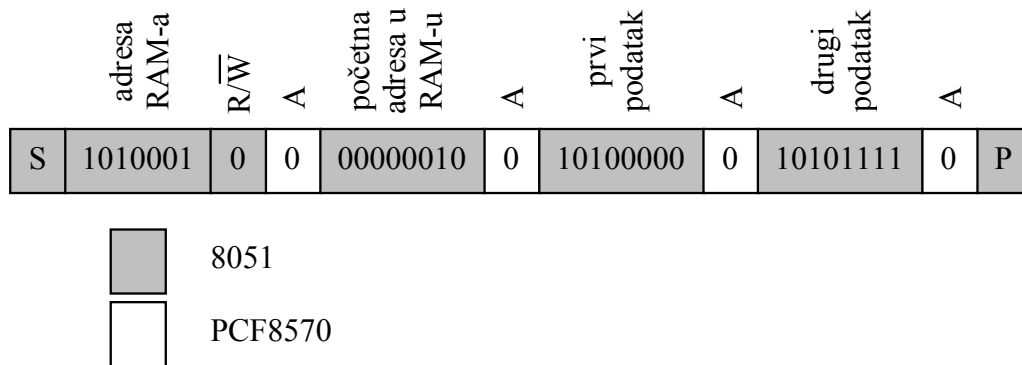
sbit SCL=P1^0;
sbit SDA=P1^1;

```

```

/*=====
/*      Funkcija za upis jednog bajta u memoriju na zadanoj I2C adresi      */
/*=====

```



```

/* Ako je izvršilac primio I2C adresu, adresu i podatak, vraća 0 */

```

```

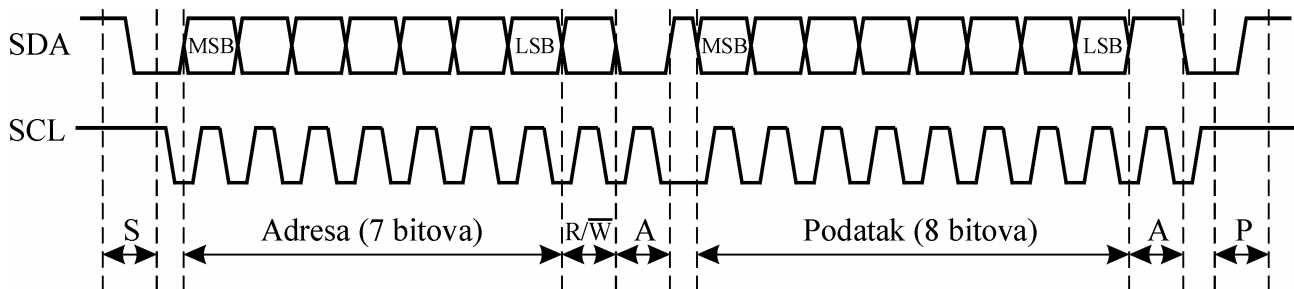
bit PisiBajt(unsigned char i2c,unsigned char Adresa,unsigned char Podatak)
{
    startbit();
    slanje8( (i2c<<1) + 0 );      /* 7 bitna adresa + bit za R/_W */
    if (potvrda())                /* ako nije stigla potvrda ... */
    {
        stopbit();
        return(1);
    }
    slanje8(Adresa);              /* adresa bajta u memoriji */
    if (potvrda())                /* ako nije stigla potvrda ... */
    {
        stopbit();
        return(1);
    }
    slanje8(Podatak);           /* podatak koji se upisuje u memoriju */
    if (potvrda())                /* ako nije stigla potvrda ... */
    {
        stopbit();
        return(1);
    }
    stopbit();
    return(0);
}

```

```

/*=====*/
/*          Pomocna funkcija za slanje start bita          */
/*=====*/

```

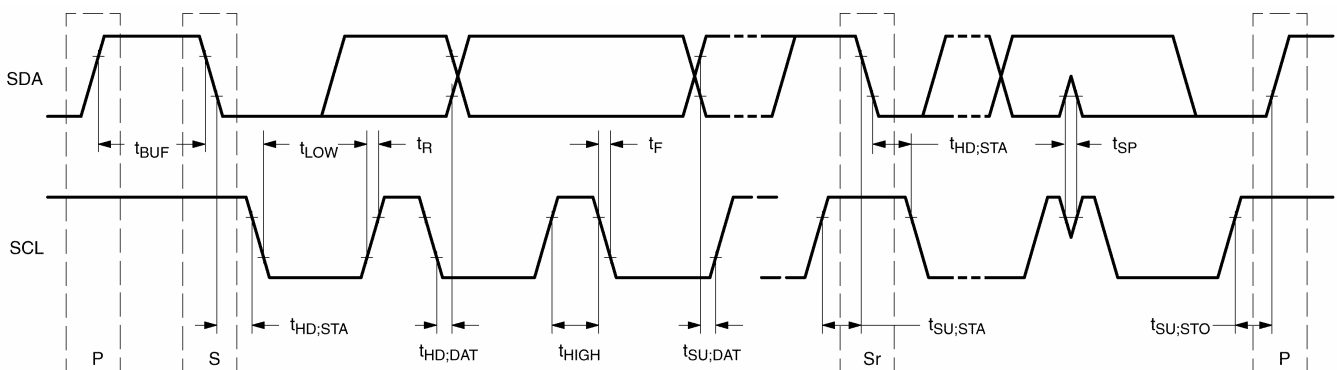


```

void startbit(void)
{
    SDA=1;
    SCL=1;
    pauza(); /* pauza jer je tSU;STA=4.7us, min */

    SDA=0; /* start bit */
    pauza(); /* pauza jer je tHD;SDA=4us, min */
    SCL=0; /* sve funkcije osim stopbit zavsavaju sa SCL=0 */
    pauza(); /* pauza jer je tLOW=4.7us, min */
}

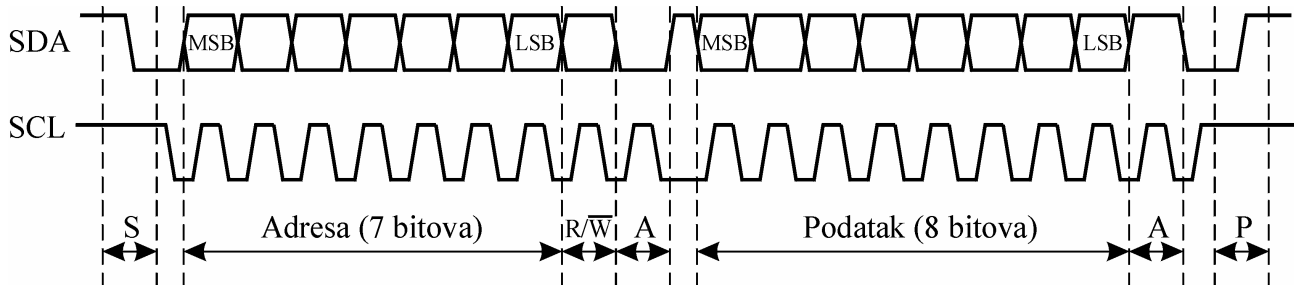
```



```

/*=====*/
/*          Pomocna funkcija za slanje jednog bajta          */
/*=====*/

```

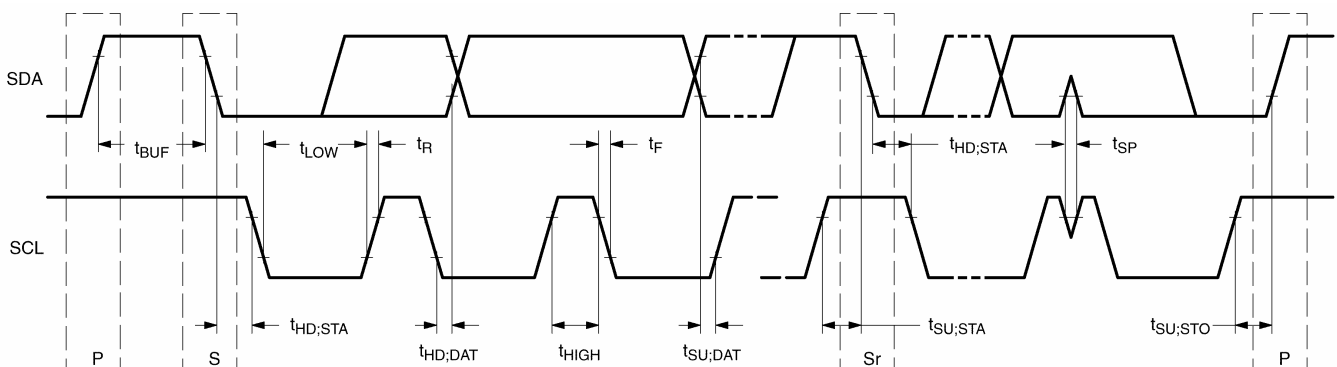


```

void slanje8(unsigned char bajt)
{
    char data j;          /* indeks u petlji */

    for (j=7;j>=0;j--)    /* -- jer najprije ide MSB */
    {
        SDA=(bit)(bajt&(1<<j)); /* u praksi ovdje ne treba cast-operator */
        /* ne treba pauza jer je tSU;DAT=250ns, min */
        SCL=1;
        pauza(); /* pauza jer je tHIGH=4us, min */
        SCL=0;
        pauza(); /* pauza jer je tLOW=4.7us, min */
    }
    SDA=1; /* da se oslobodi sabirnica */
}

```



```

/*
itd.
.
.
.
*/

```

## 10 Zaštita intelektualnog vlasništva

Ta planeta ima - ili, bolje rečeno, imala je - jedan problem, i to ovaj: većina ljudi koji su živjeli na njoj bila je nesretna veći dio vremena. Mnoga rješenja su predlagana za ovaj problem, ali većina od njih se uglavnom bavila kretanjem nekakvih malih, zelenih komada papira, što je čudno obzirom da nisu ti maleni, zeleni komadi papira bili nesretni.

- 
- 
- 

Douglas Adams

*Vodič kroz galaksiju za autostopere*



## Ideja zaštite

- "reverse engineering" - primjer PC računala
- krađa razvoja => otežaj život kradljivcu => zaštita
- cilj zaštite - produžiti vrijeme da se uređaj "klonira"
- tipovi zaštite
  - zaštita sklopovlja
  - zaštita programa
    - od kopiranja
    - od mijenjanja

## Primjeri zaštite sklopovlja

- brisanje oznaka integriranih krugova
- PAL s uključenim sigurnosnim bitom
  - ⇒ bolje registarski neko kombinacijski
- XILINX
  - slaba točka je punjenje
  - rješenje
    - ⇒ baterijsko držanje sheme
    - ⇒ fiksno ožičeni integrirani sklop

## Primjeri zaštite programske podrške

- zaštita od kopiranja
  - "dongl"
  - zaključavanje programske podrške za računalo ⇒ Internet
  - "single chip" mikrokontroler
- zaštita od mijenjanja koda
  - šifriranje teksta - protiv krađe iz druge države
  - ROM sa izmiješanim adresama (slaba zaštita)

## ***Samostalni rad***

- priprema ispita

## ***Zadaci za vježbu***

- Potrebno je sa web stranice predmeta preuzeti domaću zadaću za drugi ciklus predavanja, riješiti zadaću, te predati rješenje u skladu s priloženim uputama.

## ***Pitanja za provjeru znanja***

1. Nacrtati sklopovlje i pokazati kako se proračunavaju komponente sklopa za pogon svjetleće diode logičkim sklopom. Nacrtati sklop za pogon trošila pomoću tranzistora i MOSFET-a. Opisati način spajanja jedne znamenke 7-segmentnog pokazivača.
2. Nacrtati blokovsku shemu i pripadajući vremenski dijagram spoja 7-segmentnog pokazivača sa četiri znamenke koji radi u preklapajućem načinu rada. Što je ciklus osvježavanja. Koliko u praksi iznosi frekvencija osvježavanja? Zašto?
3. Nacrtati shemu osnovnog spoja tipke. Što je to istitravanje kontakata kod tipke? Koliko traje ta pojava? Kakve probleme izaziva i kako se ti problemi rješavaju? Dati primjer sklopovskog rješenja. Nacrtati dijagram toka programa osnovnog postupka za uklanjanje istitravanja.
4. Nacrtati blokovsku shemu spoja matrične tipkovnice. Nacrtati dijagram toka programa koji rješava problem istitravanja pomoću prekida brojila T0.
5. Objasniti razliku između sinkronog i asinkronog serijskog prijenosa. Opisati I<sup>2</sup>C sabirnicu (sklopovlje, standardne brzine, osnovne pojmove, itd.). Nacrtati vremenski dijagram i opisati ciklus prijenosa podataka na sabirnici.
6. Opisati protokol na primjeru pisanja u I<sup>2</sup>C memoriju. Opisati protokol na primjeru čitanja iz I<sup>2</sup>C memorije. Kad se koristi kombinacija čitanja i pisanja?
7. Nacrtati i opisati shemu spajanja I<sup>2</sup>C memorije na sustav s mikrokontrolerom. Objasniti kako se memorija adresira tokom komunikacije.
8. Koji postupci zaštite razvoja se u praksi koriste za zaštitu sklopovlja, a koji za zaštitu programske podrške.