

Fakultet tehničkih nauka

# Mikroprocesorski upravljački uređaji

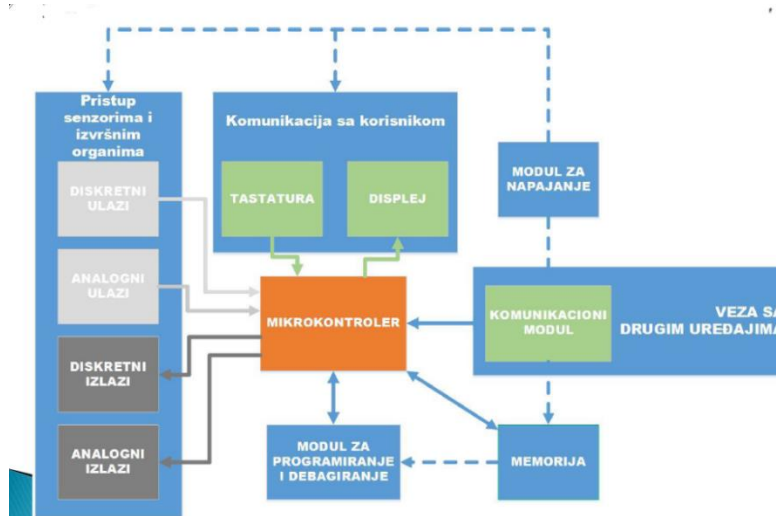
Skripta za usmeni ispit

## 1. Struktura mikroprocesorskog upravljačkog uređaja i primeri

Mikroprocesorski/mikrokontrolerski upravljački uređaji su sistemi bazirani na mikroprocesorima/mikrokontrolerima koji se primjenjuju u sistemima opšte ili posebne namjene.

Upravljački uređaji mogu biti kontrolni i regulacioni.

Blok dijagram strukture:



Centralni dio je mikrokontroler ili mikroprocesor (komunicira sa svim ostalim periferijama).

Pristup sensorima i izvršnim organima se vrši pomoći digitalnih i analognih ulaza i izlaza:

- digitalni ulazi se povezuju sa davačima
- analogni ulazi se povezuju sa sensorima (moraju imati A/D konvertor koji pravi digitalnu reprezentaciju analognog signala)
- A/D konvertore karakteriše rezolucija (broj nivoa na koje se može podijeliti opseg signala, obično je 8-bitna ili 16-bitna)
- analogni signal može biti strujni (4-20 mA) i naponski (0-5 V)
- izlazi prenose informaciju u spoljašnji svijet: DO prenose diskretnu info (logička 0 ili logička 1), a AO info iz odgovarajućeg opsega vrijednosti

Memorija predstavlja prostor za smještanje programa namijenjenog za rad upravljačkih uređaja.

Modul za programiranje služi za prebacivanje upravljačkog programa u memoriju – neophodan. Modul za debugiranje služi za nadzor izvršavanja programa – nije neophodan.

Modul za napajanje na osnovu izvora napajanja obezbjeđuje odgovarajuće logičke nivoe za funkcionisanje ostalih modula – neophodan.

Komunikacioni modul služi za komunikaciju sa nekim drugim uređajima, npr. sa PLC-om ili nekim drugim upravljačkim uređajima.

Primjeri:

**1) PLC (programabilni logički kontroler)**

- klasa industrijskih uređaja
- kontrolni – upravljački uređaj
- radi i sa A i sa D signalima
- na njima se izvršava program koji na osnovu stanja ulaza upravlja izvršnim organima

**2) INDUSTRIJSKI RAČUNARI**

- kompleksniji od PLC-a
- robusni na smetnje (elektro-statička pražnjenja, zračenja, prašinu)
- bazirani na mikroprocesoru
- na njima se izvršava poseban industrijski operativni sistem (sistem namijenjen za rad u realnom vremenu, npr. Windows CE)

**3) INDUSTRIJSKI REGULATORI**

- regulacioni upravljački uređaji
- namjenski
- regulišu tačno određene klase upravljačkih promjenljivih (npr. temperaturu, pritisak...)
- na njima se izvršavaju neki jednostavniji regulacioni krugovi gdje se regulacija vrši primjenjivanjem nekog od postojećih regulacionih zakona upravljanja (P, PI, PID...)

**4) INDUSTRIJSKI BROJAČI**

- služi za brojanje određenih događaja u pogonu
- ima tastaturu za podešavanje vrijednosti do koje broji i izlaze čije stanje zavisi od brojanja
- scan ciklus predstavlja ciklus u kom se provjeravaju stanja ulaza, izvršava program i ažurira (update-je) stanje izlaza
- dužina scan ciklusa definiše brzinu promjena koje je moguće detektovati (npr. scan ciklus traje 100ms => nemoguće detektovati promjene brže od 100ms)
- koriste se u kombinaciji sa PLC-om kada je potrebno registrovanje brzih promjena od trajanja scan ciklusa PLC-a

**5) FREKVENCIJSKI REGULATORI**

- ispravljaju industrijske mašine – trofazne asinhronne motore
- bazirani na komponentama energetske elektronike
- imaju komunikacioni interfejs za PLC
- postoji vektorski i skalarni zakon upravljanja

**6) CNC MAŠINE**

- industrijske mašine za obradu metala i drugih materijala
- pogodne i za grubu i za finu obradu
- iz bezbjednosnih razloga moraju imati pristup hlađenju

**7) INDUSTRIJSKI TERMINALI**

- predstavljaju interfejs između korisnika i upravljačkih uređaja
- pomoću njih se unose parametri sistema
- obično imaju displej sa touch screen-om
- nemaju mogućnost upravljanja

## 2. Diskretni ulazi i izlazi – šeme, naponski nivoi, primeri

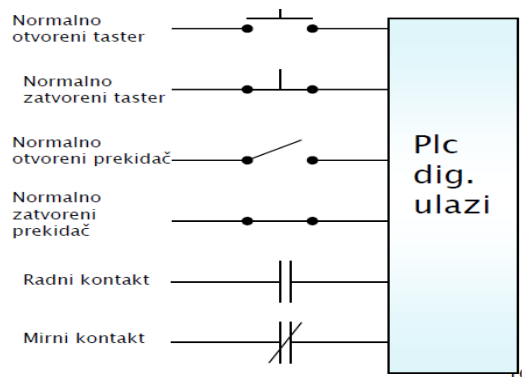
Diskretni ulazi mogu prikupiti samo 2 stanja iz spoljašnjosti, tj. pogona:

- aktivno  $\equiv$  uključi  $\equiv$  logička 1  $\equiv$  ON  $\equiv$  true  $\equiv$  visok naponski nivo (24V) ;
- neaktivno  $\equiv$  isključi  $\equiv$  logička 0  $\equiv$  OFF  $\equiv$  false  $\equiv$  nizak naponski nivo (0 V) ;

Na diskretne ulaze se povezuju **davači** (komponente koje daju 2 stanja: logička 0 ili logička 1).

Vrste davača: taster, prekidač i kontakt.

Šeme davača:

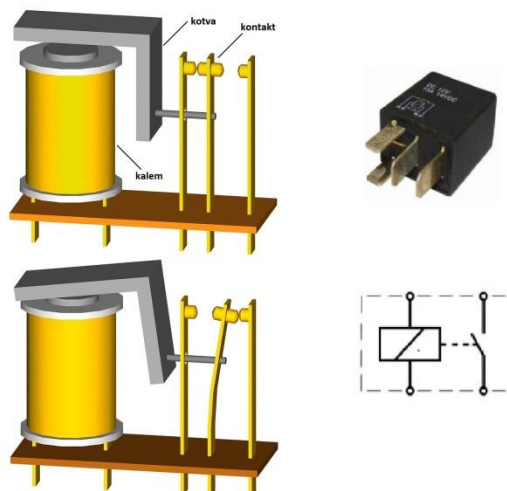


NO (normalno otvoreni): ne postoji električna veza između kontakata dok nema dejstva sile na davač ;

NC (normalno zatvoreni): električna provodnost postoji sve dok se ne djeluje na taster ;

Razlika tastera i prekidača je u tome što taster mijenja stanje samo dok se drži pritisnut, dok prekidači čuvaju stanje sve dok se ono ponovo ne promijeni.

Primjer kontakta je relejni kontakt. Releji predstavljaju strujno kontrolisanu elektro-mehaničku komponentu. Sastoji se od kotve, kalema i kontakta. Prilikom prolaska struje kotva prebacuje kontakte. Releji mogu imati jedan ili više parova kontakata (npr. jedan par predstavlja ulaz, a drugi par signalnu LE diodu).



Diskretni izlazi mogu imati ON (logička 1) i OFF (logička 0) stanje.

Za svaki ulaz i izlaz PLC-a neophodno je izvršiti **galvansku izolaciju**, tj. izolaciju između mikrokontrolera i spoljašnjeg svijeta (spoljašnji svijet predstavlja sva mjesta koja su u kontaktu sa pogonom i rukovaocem opreme).

Galvanska izolacija između dvije komponente znači nepostojanje električne veze između istih. Ugrađuje se radi zaštite mikrokontrolera od prenapona usljed mogućeg kvara.

Realizuje se pomoću pomenutih releja ili pomoću optokaplera. Optokapler predstavlja elektronsku komponentu koja se sastoji iz dva dijela: sa jedne strane se nalazi fotodioda, a sa druge fototranzistor. Prilikom prolaska struje kroz diodu, dioda propušta svjetlo i na taj način pobuđuje fototranzistor.

### 3. Analogni ulazi i izlazi – šeme, naponski i strujni opsezi i primjeri

Analogni ulazi (AI – analog input) prikupljaju vrijednosti iz odgovarajućeg opsega, dok analogni izlazi (AO – analog output) mogu generisati bilo koju vrijednost iz njihovog opsega.

Koriste se za predstavljanje promjenljivih veličina (primjeri: AI/AO za brzinu, temperaturu, težinu...).

Standardni opsezi:

- naponski (0-10 V)
- strujni (4-20 mA)

Za strujni opseg uobičajeno je da se koristi donja granica od 4mA jer transmiteri (senzori) koriste struju od 4mA za napajanje i da bi se mogao detektovati kvar (prekid žice) kada je struja 0mA.

Svaki modul analognog ulaza i izlaza vrši A/D konverziju. A/D konvertore karakterišie **rezolucija** koja govori o minimalnom kvantu (vrijednosti napona/struje) koju je moguće detektovati (u slučaju AI) ili generisati (u slučaju AO).

Na primjer:

- opseg signala je od 0-10 V, a rezolucija konvertora 8-bitna, slijedi da je

$$\text{minimalna vrijednost napona} = \frac{10V-0V}{2^8} = \frac{10}{256} = 40 \text{ mV}$$

PLC sa analognog ulaza nema mogućnost čitanja vrijednosti struje/napona, već on daje vrijednosti u opsegu rezolucije (od 0 do 255 za 8-bitnu). Vrijednosti iz opsega rezolucije se nazivaju A/D jedinice.

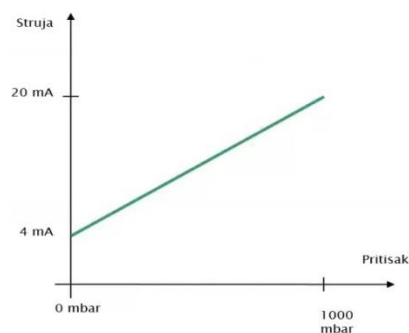
Pročitano A/D jedinicu sa ulaza je neophodno konvertovati u odgovarajuću veličinu da bi dobili intuitivnije vrijednosti za korisnika.

U praksi ne postoji strujni A/D konvertor, umjesto toga koristi se naponski A/D konvertor i to tako što se dodavanjem otpornika pri prolasku struje mjeri pad napona na njemu.

Svaki senzor ima karakteristiku na osnovu koje se vrši konverzija iz jednog opsega u drugi.

Primjer:

- senzor pritiska ulaznog opsega [0-1000]mbar i izlaznog strujnog opsega [4-20]mA
- 8-bitni A/D konvertor sa naponom od 5V



Izgled karakteristike senzora pritiska koji na svom izlazu daje standardni strujni signal 4 – 20mA

Dodajemo otpornik vrijednosti  $R = \frac{5V}{20mA} = 250 \Omega$  ;

Minimalna vrijednost struje  $I = 4mA$  odgovara padu napona na otporniku od  $U_R = 4mA * 250\Omega = 1V$  ;

Maksimalna vrijednost struje  $I = 20mA$  odgovara padu napona na otporniku od  $U_R = 20mA * 250\Omega = 5V$  ;

Opseg napona [1-5] V odgovara opsegu struje [4-20] mA ;

Mana je što se **ne koristi kompletan dinamički opseg** ! U ovom primjeru to je opseg od 0-4 mA, tj od 0-1 V. Sve vrijednosti ispod 1V na otporniku će se smatrati kao prekid žice.

S obzirom na to da je konvertor 8-bitni, tj. opsega od 0-255, vrijednosti napona od 0-1V odgovaraju vrijednostima A/D jedinica od 0 do 51.2. Slijedi da sve vrijednosti A/D jedinica u tom opsegu predstavljaju prekid žice.

[0-4) mA	[4-8) mA	[8-12) mA	[12-16) mA	[16-20) mA
[0-1) V	[1-2) V	[2-3) V	[3-4) V	[4-5) V
[0-51.2) b	[51.2-102.4) b	[102.4-153.6) b	[153.6-204.8) b	[204.8-255] b

#### 4. Mikroprocesor – blok šema i princip rada

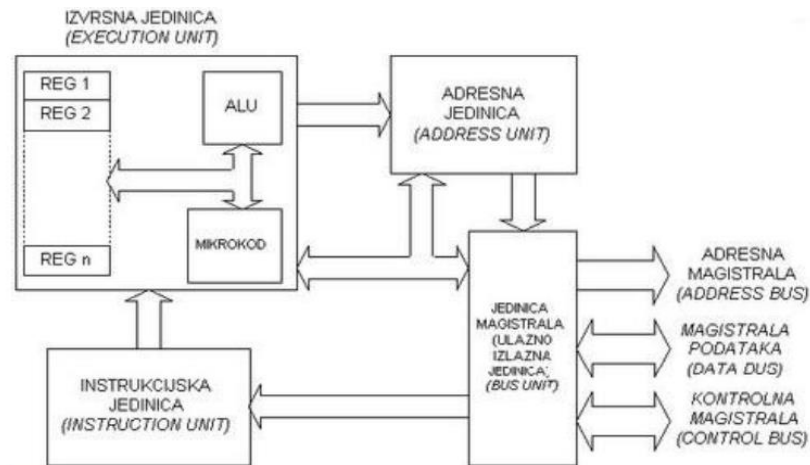
Mikroprocesor predstavlja integrisano kolo visokog stepena integracije koji izvršava mašinski kod. Visok stepen integracije podrazumijeva ogroman broj implementiranih poluprovodničkih komponenti.

Zadatak: izvršavanje mašinskog koda (programa sastavljenog od 0 i 1 u binarnoj reprezentaciji koji predstavljaju adrese memorijskih lokacija kojim mikroprocesor pristupa i kodove operacija koje treba izvršiti nad operandima).

Mikroprocesor se može predstaviti i kao mašina stanja koja može izvršavati više operacija u isto vrijeme.

Mikroprocesor nema memoriju, A/D konvertor, portove i ostale periferije. Svrha mikroprocesora je projektovanje mikroprocesorskih sistema. Memorija se nalazi izvan (predstavlja periferiju), a sa njom komunicira putem magistrala.

Šema mikroprocesora (može se proširiti raznim modulima):



Četiri funkcionalne jedinice mikroprocesora su:

- 1) **IZVRŠNA**
- 2) **ADRESNA (programski brojač)**
- 3) **INSTRUKCIJSKA (dekoder)**
- 4) **jedinica MAGISTRALA**

Magistrale:

- 1) **ADRESNA**
  - unidirekciona
  - služi da MCU postavi adresu periferije kojoj želi pristupiti
- 2) **magistrala PODATAKA (data magistrala)**
  - bidirekciona
  - putem nje MCU prima ili šalje podatke od određene periferije
  - govori da li šaljemo ili čitamo podatke
- 3) **KONTROLNA**
  - bidirekciona
  - putem nje MCU govori da li hoćemo ili nećemo poslati/primiti podatak sa periferije

*Primjer:*

### 8-bitni mikroprocesor

- Svaka instrukcija širine 8 bita (1 bajt  $\equiv$  256 asemblerskih instrukcija) ;
- Instrukcija se sastoji od 2 bajta
  1. KOD OPERACIJE (1. bajt) – na parnim lokacijama (0,2,4...)
  2. OPERAND (2. bajt) – na neparnim lokacijama (1,3,5...)

- Parne lokacije  $\equiv$  kod (ono šta treba izvršiti) ;
- Neparne lokacije  $\equiv$  operand (ono nad čim se izvršava operacija) ;
- npr. asemblerska instrukcija “MOV A, 21H” u mašinskoj predstavi je “B0H 21H”, gdje B0H mašinska predstava instrukcije MOV
  - B0H je kod operacije i nalazi se na parnoj lokaciji
  - 21H je operand i nalazi se na neparnoj lokaciji
- Prva faza rada mikroprocesora pri dovođenju napajanja je RESET (inicijalizacija)
  - postavljanje ADRESNE JEDINICE (programskog brojača) na 0
  - 0 sa ADRESNE JEDINICE odlazi na ADRESNU magistralu (prvi pristup memoriji će biti na memorijskoj lokaciji 0)
- U sljedećem stanju na magistrali PODATAKA se da naredba READ i u tom trenutku se otvara “kanal” između magistrale podataka i memorijske lokacije 0 na periferiji ;  
Na drugoj ivici, podatak sa nulte memorijske lokacije se pojavljuje na jedinici MAGISTRALA (na jednoj ivici takta je data naredba READ, a već na drugoj ivici podatak se pojavio) ;
- Podatak iz jedinice MAGISTRALA odlazi u INSTRUKCIJSKU jedinicu ;
- INSTRUKCIJSKA jedinica dekodira podatak (kod operacije) i zatim putem IZVRŠNE jedinice inkrementira brojač (ADRESNU jedinicu) – sada se u njoj nalazi 1 ;
- 1 sa ADRESNE jedinice se upisuje u ADRESNU MAGISTRALU, magistrala PODATAKA izvršava READ i na sledećoj ivici podatak sa memorijske lokacije 1 se pojavljuje na jedinici MAGISTRALA ;
- podatak sa jedinice MAGISTRALA direktno odlazi u IZVRŠNU jedinicu ;

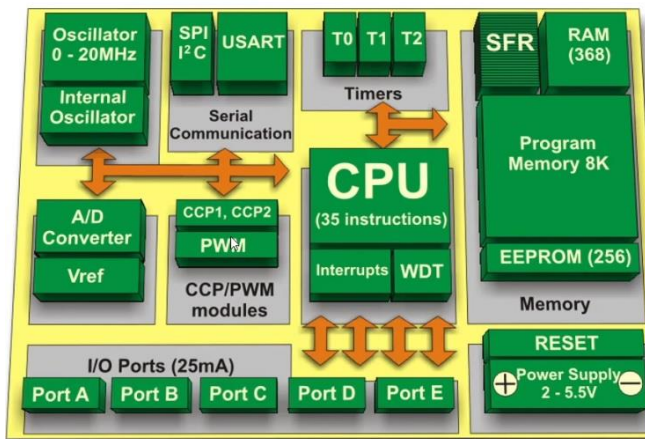
PARNE lokacije	ADRESNA jedinica » jedinica MAGISTRALA » INSTRUKCIJSKA jedinica » IZVRŠNA jedinica
NEPARNE lokacije	ADRESNA jedinica » jedinica MAGISTRALA » IZVRŠNA jedinica

- Zaključak: podatak sa parnih lokacija mora proći kroz INSTRUKCIJSKU jedinicu (dekodiranje), dok podatak sa neparnih lokacija iz jedinice MAGISTRALNE direktno ide u IZVRŠNU JEDINICU gdje se aktivira MIKROKOD koji šalje signal ALU (aritmetičko logičkoj jedinici) da prebaci operand u akumulator (registar A), potom IZVRŠNA jedinica šalje signal ADRESNOJ jedinici (programskom brojaču) da se inkrementira ;
- Inkrementiranjem programskog brojača pokazuje se adresa instrukcije koja sljedeća treba da se izvrši ;
- Ovaj ciklus se ciklično ponavlja sve dok INSTRUKCIJSKA jedinica ne dekodira END instrukciju (F4H) prilikom čega se prekida inkrementiranje programskog brojača i proces se zaustavlja (ali mikroprocesor nastavlja sa radom) ;



## 5. Mikrokontroler –princip rada i osnovne jedinice

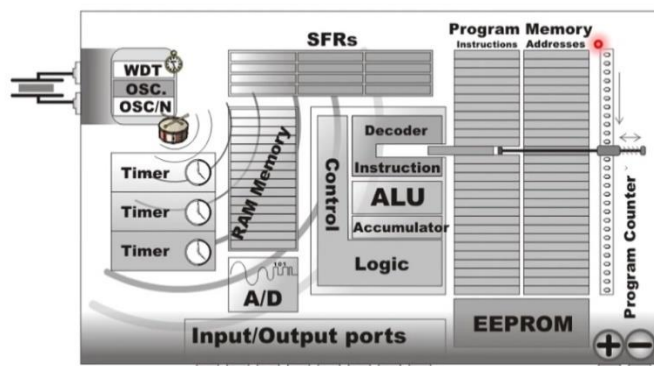
Mikrokontroler je integrirano kolo visokog stepena integracije koje u sebi sadrži: procesor (CPU), memoriju, I/O portove, tajmere, brojače, A/D i D/A konvertore ...



CPU (mikroprocesor) je povezan sa svim ugrađenim periferijama putem **internih magistrala**. Zahvaljujući većim brojem internih magistrala mašinski program se izvršava brže nego na mikroprocesorima (koji ima samo 3 magistrale).

Mikrokontroler predstavlja skup mikroprocesora sa svim periferijama koje se nalaze u njemu. On ima svoj ROM, što znači da nikakve periferije nisu neophodne da se napravi funkcionalan sistem.

Princip rada je isti kao kod mikroprocesora, jedina razlika je što se kod mikrokontrolera sve nalazi na jednoj ploči.



SFR (specijalni funkcijski registri) predstavljaju posebnu memoriju (slično RAM-u, ali nije ista funkcija).

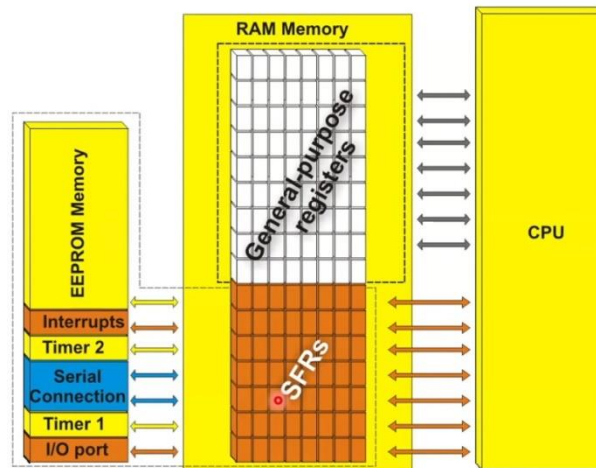
RAM služi za smještanje varijabli i adresa za praćenje rada programa, dok pomoću SFR-ova mikrokontroler komunicira sa periferijama.

EEPROM memorija je memorija koja čuva sadržaj i nakon prekida napajanja.

Registri opšte namjene služe za definisanje promjenljivih.

Periferije sa kojim se komunicira putem SFR-ova su:

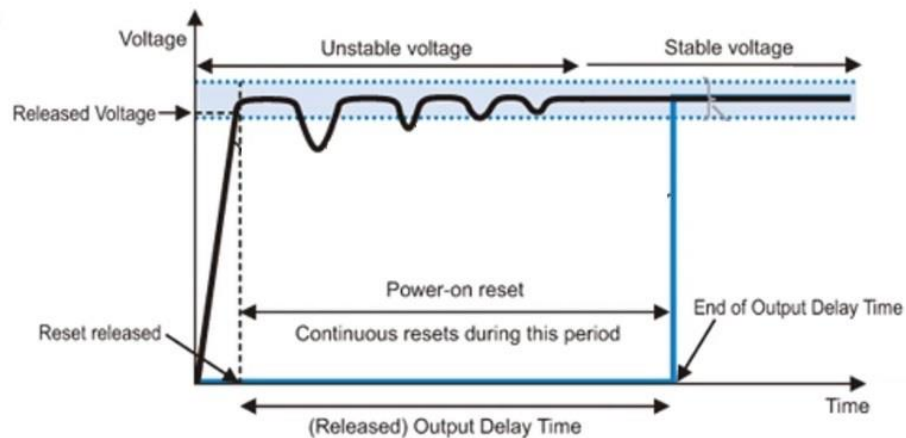
- EEPROM
- prekidi
- tajmeri
- I/O portovi
- serijska komunikacija (UART i SPI)



## 6. Reset kolo mikrokontrolera – princip rada i svrha upotrebe

Reset kolo je kolo čiji je zadatak da resetuje mikrokontroler, tj. postavi sve registre, memorijske lokacije i periferije mikrokontrolera u inicijalno stanje (poznato stanje od kojeg može da se počne sa normalnim radom). U slučaju da ne dođe do inicijalizacije, izvršavanje programa postaje nepredvidljivo.

Reset je neophodan radi **stabilizacije napona napajanja** mikrokontrolera (napajanje predstavlja dovođenje napona na nožice mikrokontrolera).

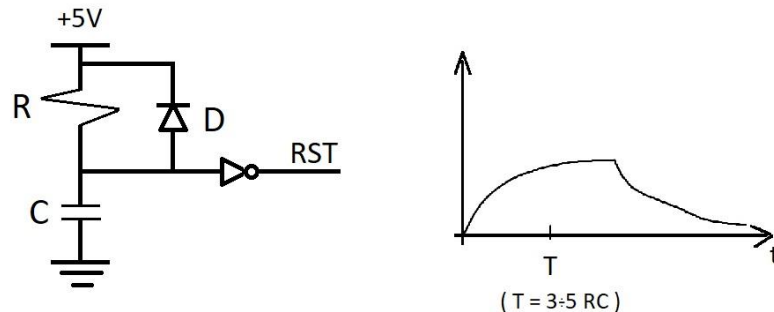


Pri trenutku aktivacije napon napajanja je u linearnom rastu. Linearan rast zahtjeva povlačenje velike struje iz izvora (u cilju punjenja kapacitivnih komponenti sistema).

Ukoliko podizanje napona traje predugo mikrokontroler može doći u nedefinisano stanje. Da bi se to spriječilo prilikom aktiviranja napona napajanja mikrokontroler se postavlja u reset. U trenutku kada se napon smiri mikrokontroler izlazi iz RESET-a. Ovaj mehanizam je poznatiji kao **power-on reset**. Dužina trajanja power-on reseta zavisi od mikrokontrolera.

U slučaju da mikrokontroler nema ugrađeno RESET kolo, neophodno ga je dodati.

Projetoвање RESET kola:



- pomoću pasivne mreže (kondenzatora C i otpornika R) ;
- vremenska konstanta punjenja i pražnjenja  $\tau = RC$  ;
- dioda D se dodaje antiparalelno da bi obezbjedila brže pražnjenje kondenzatora pri gubitku napajanja ;
- brže pražnjenje kondenzatora se mora obezbjediti zbog slučajeva kada je gubitak napajanja kraći od  $\tau$ , jer u suprotnom neće doći do reseta mikrokontrolera ;

## 7. Kolo za detekciju propadanja napona napajanja – Brown out

Propadanje napona napajanja se može desiti prilikom uključivanja nekih komponenti koje povuku veliku struju pri aktivaciji.

Propadanje  $\equiv$  brown out može izazvati pogrešan rad mikrokontrolera i pogrešno izvršavanje programa.

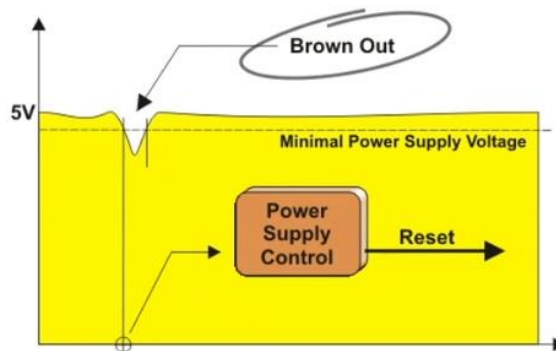
Osobine propadanja:

- 1) **dubina** (plitka ili niska)
- 2) **trajanje** (kratko ili dugo)

Da bi se spriječilo propadanje uvodi se **brown out detektor** kolo čiji je zadatak detektovanje pada napona napajanja.

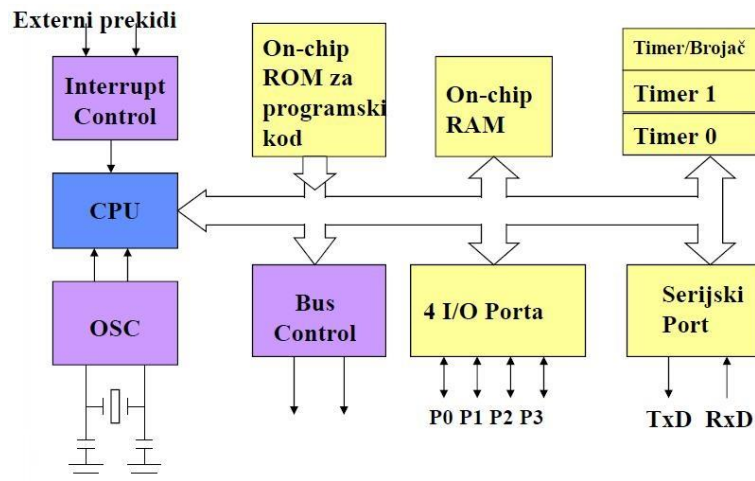
Princip detekcije:

- postoji komparator koji poredi referentni napon (granicu dozvoljene dubine) i trenutni napon napajanja ;
- ukoliko dođe do propadanja napona ispod referentnog nivoa pali se tajmer koji počinje odbrojavati dužinu trajnja propadanja ;
- ako je trajanje kraće od unaprijed definisanog dozvoljenog trajanja propadanja ne dešava se ništa, ali ako se desi da propadanje traje duže, brown out detektor kolo šalje signal reset kolu da izvrši reset ;
- Brown out detektor kolo ne izvršava reset, ono samo šalje signal reset kolu kada da se aktivira ;



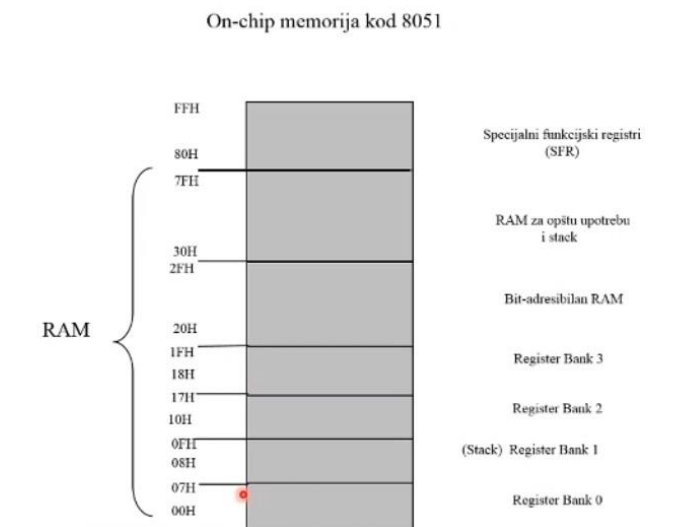
## 8. Mikrokontroler 8051 – osnove arhitekture, resursi, osnovni registri, specijalni funkcijski registri

Mikrokontroler 8051 sadrži procesor koji je internim magistralama povezan na sljedeće periferije: memoriju (ROM i RAM), tajmere, I/O portove, serijski port, kontrolnu magistralu, eksterni oscilator i modul za prekide.



Eksterni prekidi su događaji (signali na ulazu) koji prekidaju normalno izvršavanje koda i dovode do neke posebne rutine izvršavanja u što kraćem mogućem roku.

	8051
ROM (u bajtovima)	4K
RAM (u bajtovima)	128
Tajmeri	2
I/O pinova	32
Serijski port	1
Izvori prekida	5
SPI	NE



Pinovi:

- VCC (5 V)
- GND
- XTAL1 i XTAL2 (za eksterni oscilator)
- RST
- I/O portovi (P0, P1, P2, P3) – svaki port ima 8 pinova i svi su dvosmjerni
- PSEN (za pristup eksternom ROM-u)
- EA i ALE (za pristup eksternom ROM-u i RAM-u)

SFR (specijalni funkcijski registri) služe za konfigurisanje resursa (periferija) mikrokontrolera i za očitavanje podataka (bajtova) sa tih periferija.

Neki od SFR-ova: P0 (pin), SP (stack pointer), TCON, TMOD, TH0, TL0, TH1, TL1, AUXR (za XRAM), SCON i SBUF (za UART), PSW (carry flag, auxiliary flag, overflow flag, parity flag) ...

Osnovni registri:

- 8-bitni:
  - **A registar**
    - akumulator
    - služi za smještanje rezultata i operanada većine mašinskih instrukcija
  - **B registar**
    - sličan A registru ali služi samo za operacije množenja i dijeljenja

- **Rn registri** (  $n \in [0,7]$  )
  - pomoćni registar u velikom broju operacija
  - služi za smještanje međurezultata
- 16-bitni:
  - **DPTR** (gornjih osam bita DPH, a donjih DPL)
    - data pointer, tj. pokazivač na podatke (na adresi eksterne memorije)
    - jedini 16-bitni registar kojem se može pristupiti
  - **PC** (program counter)
    - govori gdje se u memoriji nalazi sljedeća instrukcija koja treba da se izvrši
    - uvijek kreće od 0000H i inkrementira se za 1, 2 ili 3
    - pristupa mu se samo indirektnim adresiranjem što znači da je nemoguće direktno pročitati njegovu vrijednost ili ju direktno upisati

## 9. Mikrokontroler 8051 – vrste memorije, način adresiranja memorije

Vrste memorije: ROM, RAM, SFR, eksterni ROM i RAM.

ROM služi za smještanje programskog koda.

RAM služi za smještanje varijabli i adresa za praćenje rada programa. U RAM-u se nalazi stack (poseban dio memorije) koji služi za smještanje rezultata operacija, kao i za čuvanje memorijskih lokacija (vrijednosti programskog brojača) na koje želimo da se vratimo nakon obavljanja neke druge rutine. Registar za pristup stack-u je SP (stack pointer), a upisom vrijednosti u SP konfiguriramo u kojem dijelu RAM-a će se stack nalaziti. Stack predstavlja LIFO strukturu podataka. Korištenjem instrukcija PUSH i POP moguće je podatke smješati i izbacivati sa stack-a.

SFR (specijalni funkcijski registri) predstavljaju posebnu memoriju (sličnu RAM-u) pomoću koje se pristupa periferijama mikrokontrolera.

Eksterni ROM i RAM je moguće ugraditi kao periferije ukoliko mikrokontroler nema dovoljno memorije.

Načini adresiranja:

### 1) NEPOSREDNI

- broj koji se obrađuje se nalazi neposredno u ROMu iza koda operacije
- najbrži način
- primjer: **MOV A, #20h**

### 2) PREKO REGISTARA

- primjer: **MOV R1, A** (greška: ~~MOV R1, R3~~)

### 3) DIREKTNI

- direktnim adresiranjem je moguć pristup SFR-ovima
- direktno se daje adresa sa koje će se preuzeti podatak
- primjer: **MOVA, 30h**

### 4) INDIREKTNI

- primjer: **MOVA, @R0**
- u ovom primjeru u registar A se upisuje podatak sa adrese koja je upisana u R0
- indirektnim adresiranjem je nemoguć pristup SFR-ovima
- u proširenoj arhitekturi (AT89C51RC2) ovim adresiranjem se pristupa XRAM-u

### 5) ADRESIRANJE EKSTERNE MEMORIJE

- prvo je neophodno postaviti adresu u DPTR (16 bita) ili u R0 i R1 (8 bita)
- koristi se posebna naredba MOVX
- primjer: **MOVX A, @DPTR**

### 6) ADRESIRANJE ROM-a

- unutar ROM-a moguće je napraviti lookup tabele (nepromjenljive tabele podataka)
- koristi se posebna naredba MOVC
- moguće je samo čitanje podataka !
- primjer: **MOVC A, @A+DPTR**

## 10. Mikrokontroler 8051 – rad sa I/O portovima, mašinski ciklus

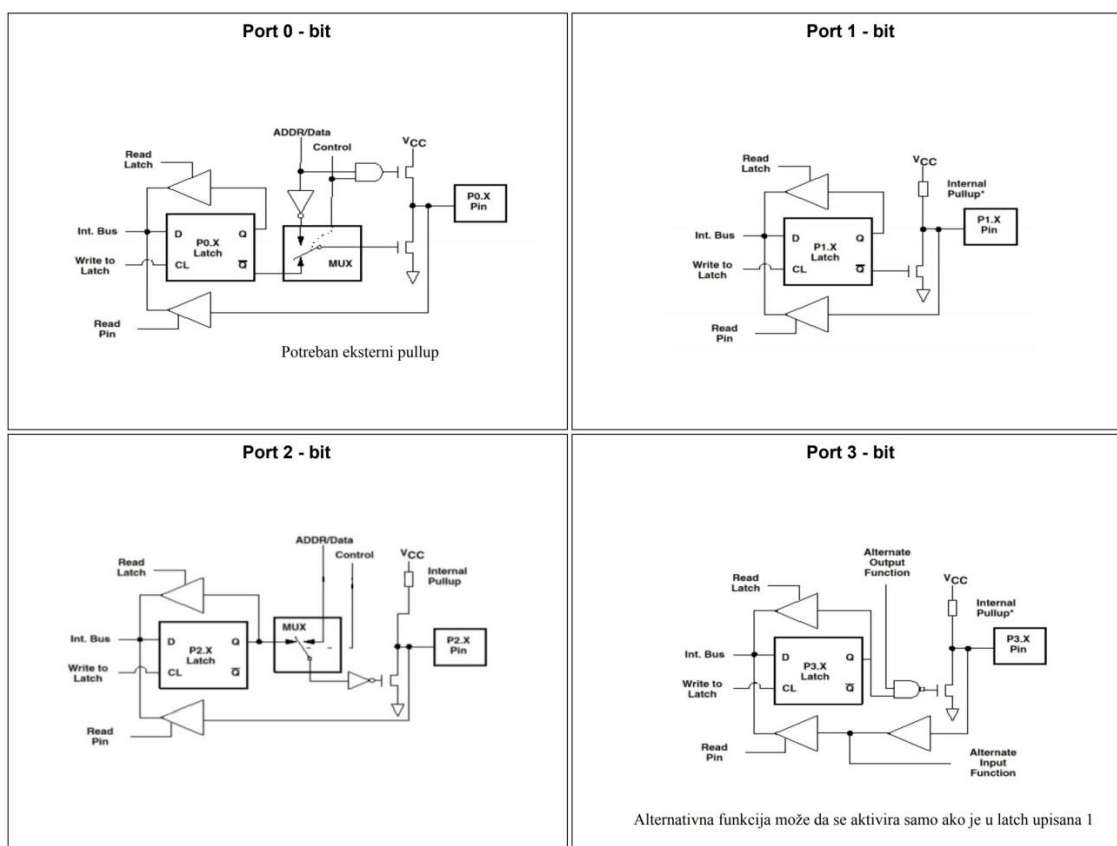
Mikrokontroler 8051 ima 4 porta (svaki port ima po 8 pinova). Portovi predstavljaju interfejs između mikrokontrolera i okoline.

Svaki pin je dvosmjernan što znači da može biti i ulaz i izlaz opšte namjene.

Specijalne namjene portova:

- Port 0: pristup eksternom ROMu i RAMu (niži bajt adrese ili podaci)
- Port 2: pristup eksternom ROMu i RAMu (viši bajt adrese)
- Port 3:
  - 0 - RxD: serijski ulaz
  - 1 - TxD: serijski izlaz
  - 2 - INT0: eksterni prekid 0
  - 3 - INT1: eksterni prekid 1
  - 4 - T0: tajmer 0 (eksterni ulaz)
  - 5 - T1: tajmer 1 (eksterni ulaz)
  - 6 - WR
  - 7 - RD

Da bi se pin porta koristio kao **ULAZ**, u njegov **latch** mora da se upiše **1 !!!**



Samo Port 0 nema interni pullup što znači da je neophodno ugraditi eksterni pullup.

Kod Porta 0 i Porta 2 ako je Control = 1 port se koristi za adresiranje eksterne memorije, u suprotnom kao ulaz/izlaz opšte namjene.

**Mašinski ciklus** (MC) predstavlja osnovnu jedinicu za vrijeme unutar mikrokontrolera. Jedan MC ima 6 stanja. Za svako stanje su neophodna 2 takta eksternog oscilatora, što znači da 6 stanja ima ukupno 12 perioda (taktova).

Eksterni oscilator se dovodi na pinove XTAL1 i XTAL2 i on obezbjeđuje osnovni takt.

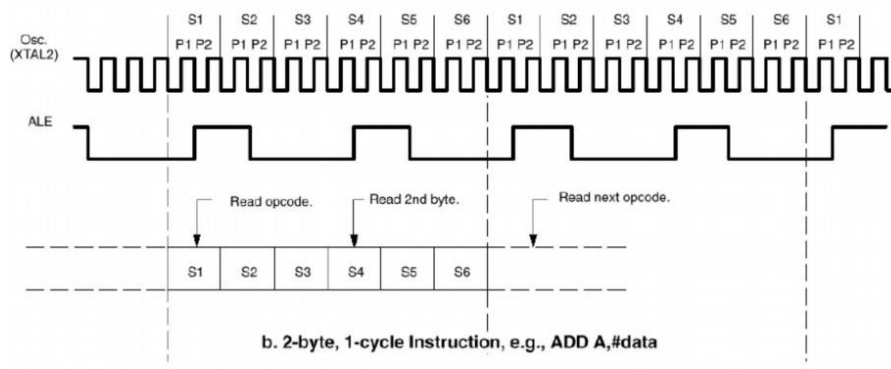
Nijedna instrukcija ne može da se izvrši brže od 1 MC (većina traju 1 MC, neke traju 2 (npr: JMP, JE), a MUL i DIV traju čak 4 MC-a).

Za vrijeme jednog mašinskog ciklusa (tj. 6 stanja) pročita se kod naredbe, dekodira, pročita operand, izvrši instrukcija i proslijedi njen rezultat.

Trajanje mašinskog ciklusa se može izračunati na osnovu frekvencije eksternog oscilatora.



## Izvršavanje instrukcije

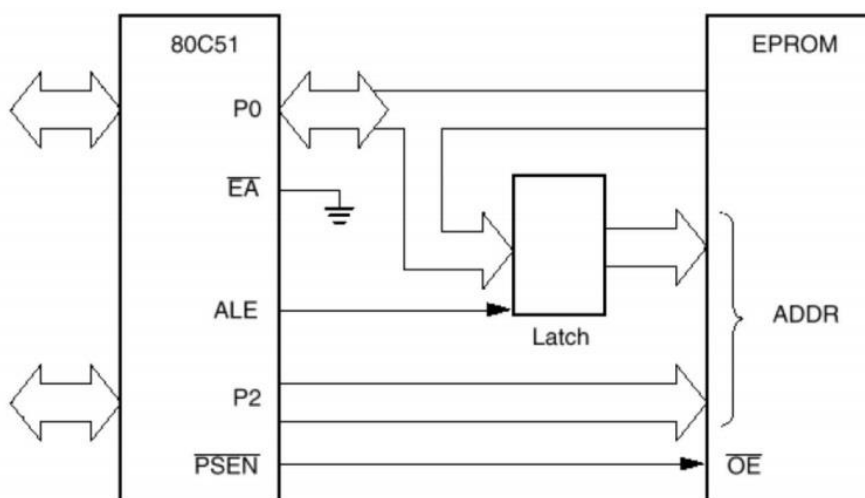


Primjer: XTAL  $\equiv$  f = 11,0592 MHz

$$T_{MC} = \frac{1}{\frac{f}{12}} = 12 \frac{1}{11,0592 \text{ MHz}} = 1,085 \mu s$$

## 11. Mikrokontroler 8051 – pristup eksternoj programskoj memoriji

Program se smješta na eksterni ROM u slučaju da 4K internog ROMa nije dovoljno (ROM – read-only memory).



Pristupi:

- 1) čitav program se smješta u eksterni ROM (ukoliko je  $EA = 0$  )
- 2) u eksterni ROM samo ono što ne može stati u interni (  $EA = 1$  )

P0 služi za adresiranje nižeg bajta i za čitanje podataka eksterne memorije (*ROM je moguće samo čitati !*) ;

P2 služi za adresiranje višeg bajta eksterne memorije ;

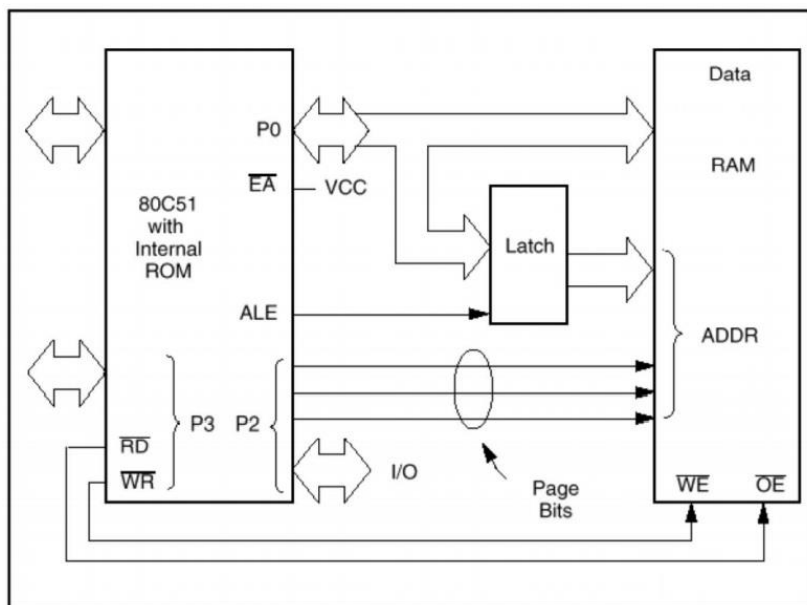
Prilikom adresiranja eksterne memorije viši bajt sa P2 se upisuje direktno, a niži bajt sa P0 preko latch-a. Da bi se P0 oslobodio za čitanje podatka, niži bajt adrese se mora sačuvati u latch-u. To se realizuje tako što se ALE postavi na 1 (  $ALE = 1$  ).

Nakon postavljanja nižeg bajta adrese u latch sve je spremno za čitanje podatka što se realizuje postavljanjem pinova:

1.  $ALE = 0$
2.  $PSEN = 0$

## 12. Mikrokontroler 8051 – pristup eksternoj radnoj memoriji

Pristup eksternom RAMu je sličan kao pristup eksternom ROMu. Razlika je u tome što kod RAMa, pored čitanja moguće je vršiti i upisivanje podataka u njega.



Setovanjem pinova RD (read) i WR (write) govori se da li se podaci čitaju ili upisuju. Ako su oba setovana funkcionalnost je onemogućena.

Adresiranje se vrši na isti način kao kod eksternog ROMa (P0 predstavlja niži bajt i mora da se upisuje preko latch-a, dok P2 predstavlja viši bajt memorijske lokacije kojoj se pristupa na eksternom RAMu).

Ukoliko se ne pristupa čitavom eksternom RAMu, moguće je dio porta P2 osloboditi, a koristiti samo onoliko pinova koliko nam je potrebno.

Za pristup eksternoj memoriji koristi se naredba MOVX.

### **13. Mikrokontroler AT89C51RC2 – najvažnije razlike u odnosu na osnovnu arhitekturu 8051**

	8051	AT89C51RB2/C2
ROM (u bajtovima)	4K	16K/32K
RAM (u bajtovima)	128	256+1024
Tajmeri	2	3
I/O pinova	32	32
Serijski port	1	1
Izvori prekida	5	9
SPI	NE	DA

#### **Najvažnije razlike u odnosu na osnovnu arhitekturu 8051**

- ☐ 256 bajtova RAM memorije (128 bajtova direktno adresabilnih iz osnovne arhitekture 8051 + 128 dodatnih bajtova dostupnih samo indirektnim adresiranjem)
- ☐ 1024 bajta dodatne RAM memorije (eXpanded RAM - XRAM)
- ☐ 32 KB ROM
- ☐ X2 mod rada
- ☐ Mogućnost skaliranja osnovnog takta kontrolera
- ☐ Tri tajmera (Tajmer 0 i Tajmer 1 iz osnovne arhitekture 8051 + Tajmer 2)
- ☐ SPI interfejs (sinhrona serijska komunikacija)
- ☐ Watchdog tajmer
- ☐ Programmable Counter Array
- ☐ Keyboard interface
- ☐ Boot ROM

#### 14. Mikrokontroler AT89C51RC2 – XRAM memorija

Pristup XRAM memoriji se vrši pomoću naredbe MOVX. Izbor da li se pomoću MOVX naredbe vrši pristup internoj XRAM memoriji ili eksternoj RAM memoriji, vrši se pomoću specijalnog funkcijskog registra AUXR.

##### AUXR (8Eh)

3	XRS1	veličina XRAM memorije XRS1 XRS0 0 0 256 bajtova (default)		
2	XRS0	0 1 512 bajtova 1 0 768 bajtova 1 1 1024 bajta		
1	EXTRAM	 0 - pristup XRAMu 1 - pristup eksternoj memoriji ( naredbom MOVX )		

#### 15. Mikrokontroler AT89C51RC2 – X2 mod rada


X2 mod rada predstavlja mod u kome je rad mikrokontrolera duplo brži.

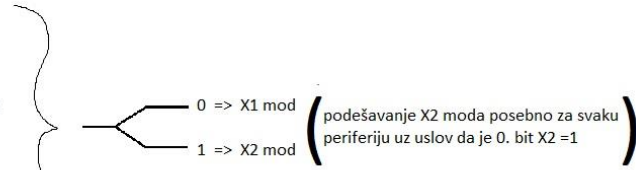
U X2 modu osnovni takt se dijeli sa 6 (ne sa 12), što znači da je za 1 MC potrebno 6 osnovnih taktova. Jedan MC ima 6 stanja, a svako stanje ima samo jedan osnovni takt.

X2 je moguće izabrati za CPU (mikroprocesor), a i za svaku od periferija posebno.

Izbor se vrši pomoću SFR-a **CKCON0**.

##### CKCON0 (8Fh)

Bit	Ime	
7	-	
6	WDX2	Watchdog timer
5	PCAX2	
4	SIX2	Serijska komunikacija
3	T2X2	
2	T1X2	Tajmeri
1	T0X2	
0	X2	 0 => X1 mod 1 => X2 mod


0 => X1 mod  
1 => X2 mod (podešavanje X2 moda posebno za svaku periferiju uz uslov da je 0. bit X2 =1)

## 16. Mikrokontroler AT89C51RC2 – skaliranje osnovnog takta

Dodatnim skaliranjem osnovnog takta se smanjuje brzina rada mikrokontrolera. Smanjivanjem brzine rada smanjuje se i potrošnja mikrokontrolera, što je pogodno za baterijska napajanja kod nekih jednostavnijih sistema.

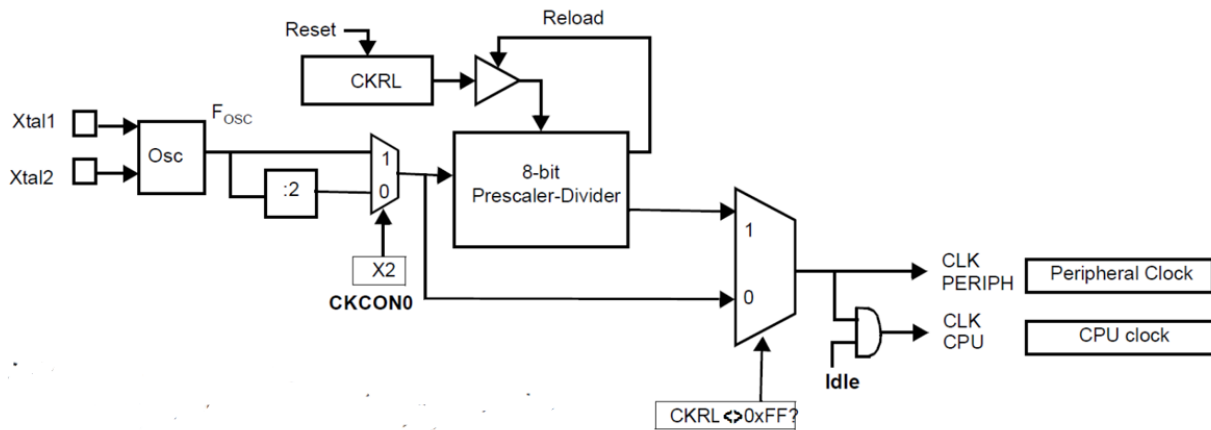
Skaliranje osnovnog takta se vrši upisivanjem vrijednosti u **CKRL**.

Inicijalna vrijednost u CKRL je FFh (to odgovara maksimalnoj frekvenciji, tj. najbržem režimu). Ukoliko u CKRL upišemo 00h dobijamo minimalnu frekvenciju, tj. nasporniji režim.

Skaliranje takta:

$$X1 \text{ mod: } F_{CLK} = \frac{F_{OSC}}{4 (255 - CKRL)}$$

$$X2 \text{ mod: } F_{CLK} = \frac{F_{OSC}}{2 (255 - CKRL)}$$



## 17. Mikrokontroler AT89C51RC2 – prekidi

Prekidi se pokreću pri određenim događajima sa kojim su povezani.

Pri dešavanju događaja koji izaziva prekid, mikrokontroler prestaje sa normalnim izvršavanjem i odlazi na posebnu memorijsku lokaciju gdje se nalazi poseban dio koda koji je vezan za obradu tog prekida.

Nakon obrade prekida mikrokontroler se vraća na memorijsku lokaciju gdje bio prije događaja i vraća se na normalno izvršavanje programa.

Događaji koji mogu uzrokovati prekid su:

### Osnovna arhitektura 8051

- Timer 0 Overflow
- Timer 1 Overflow
- Prijem/Slanje bajta serijskom komunikacijom.
- Externi događaj 0
- Externi događaj 1

### Dodatni resursi u odnosu na 8051

- Timer 2 Overflow
- Programmable Counter Array
- Keyboard interface
- SPI – sinhrona serijska komunikacija

Svaki prekid ima flag u odgovarajućem SFR-u koji kada se setuje dolazi do obrade tog prekida. Neki flagovi se resetuju harverski, a neke je potrebno resetovati softverski (programski).

Broj prekida	Prekid	Flag
0	Externi 0	IE0
1	Timer 0	TF0
2	Externi 1	IE1
3	Timer 1	TF1
4	Serijski	RI/TI
5	Timer 2	TF2+EXF2
6	PCA	CF + CCFn (n = 0-4)
7	Keyboard	KBDIT
8	SPI	SPIF

Omogućavanje prekida se vrši setovanjem odgovarajućih bitova registara **IEN0** i **IEN1**. Pored bitova za omogućavanje svakog prekida pojedinačno, postoji i **EA** (enable all) bit koji je neophodno setovati da bi mehanizam prekida uopšte radio, jer on predstavlja globalnu dozvolu prekida.

IEN0(A8h)

Bit	Ime
7	EA
6	EC
5	ET2
4	ES
3	ET1
2	EX1
1	ET0
0	EX0

IEN1(B1h)

Bit	Ime
7	-
6	-
5	-
4	-
3	-
2	ESPI
1	-
0	KBD

**Prioriteti prekida** predstavljaju mehanizam kojim je moguće upravljati redom izvršavanja prekida. Za svaki prekid moguće je postaviti prioritet od 0 do 3 (3 je najveći prioritet).

Prekidi višeg prioriteta (ukoliko dođe do događaja koji ih izazivaju) zaustavljaju obradu prekida nižeg prioriteta (ukoliko se takav prekid trenutno obrađuje) i “preko reda” dolazi do obrade prekida višeg prioriteta.

Prioriteti se podešavaju postavljanjem odgovarajućih vrijednosti u SFR-ove: **IPH0**, **IPLO** (čine jedan par), **IPH1** i **IPL1** (drugi par).

Ukoliko se prioriteti ne definišu obrada se dešava na osnovu **sekvence provjere prekida** koja je specifična za mikrokontroler.

## 18. Mikrokontroler AT89C51RC2 – tajmeri

AT89C51RC2 ima tri tajmera: T0, T1 i T2.

### Tajmeri T0 i T1:

- Funkcije :
  - 1) brojanje vremena i/ili brojanje razmaka između 2 događaja
  - 2) brojanje događaja
  - 3) generisanje baud rate-a za serijski port (pri asinhronoj serijskoj komunikaciji)
- Radom T0 i T1 se upravlja pomoću SFR-ova:

Ime SFR	
TH0	u njima se nalazi sadržaj koji se inkrementira na osnovu mašinskog ciklusa ili eksternih događaja
TL0	
TH1	
TL1	
TCON	konfigurišu upravljanje radom T0 i T1
TMOD	

TMOD (89h) SFR:

Bit	Ime	
7	GATE1	sve je isto samo su ovo bitovi za T1
6	C/T1#	
5	T1M1	
4	T1M0	
3	GATE0	0 => tajmer radi ako je $TR0=1$ 1 => rad tajmera se kontroliše signalom sa eksternog pina (uz $TR0$ )
2	C/T0#	
		bitovi za T0
1	T0M1	služe za izbor moda
0	T0M0	

0 => brojanje mašinskih ciklusa  
1 => brojanje eksternih događaja sa pina

Izbor moda rada timera:

TxM1	TxM0	Timer Mode	Opis moda
0	0	0	13-bit Timer.
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

TCON (88h)

Bit	Ime	
7	TF1	T1
6	TR1	
5	TF0	---> flag koji se setuje (kada dođe do overflow-a u TH0 i TL0) i pokreće prekid
4	TR0	0 => T0 isključen 1 => T0 uključen

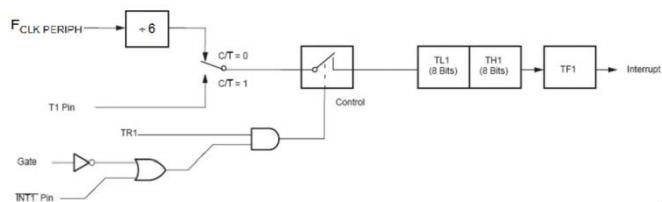
Mod 1 i mod 2 su najčešće korišteni.

Mod 1 predstavlja 16-bitni mod (TH1 i TL1 se inkrementiraju do 0xFFFFh, a na sljedećem inkrementu setuju TF1 flag i generišu prekid).

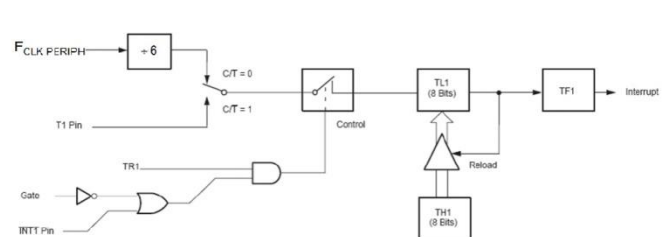
Mod 2 predstavlja 8-bitni autoreload mod (nakon što se TL1 inkrementira do 0xFFh i generiše prekid, TL1 neće krenuti da broji od 0x00h, nego od vrijednosti koja je upisana u TH1 – autoreload vrijednosti).

Kada je tajmer u modu 3 oba tajmera se koriste za prekide i nemoguće ih je koristiti za nešto drugo.

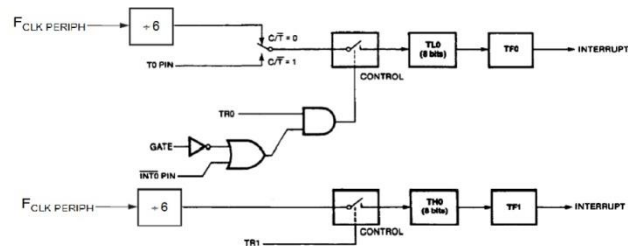
**Princip rada tajmera1 u modu 1**



**Princip rada tajmera 1 u modu 2**



**Princip rada tajmera 0 i 1 u modu 3**



### **Tajmer T2:**

- Uvijek radi kao 16-bitni brojač
- Modovi rada:
  - 1) Autoreload
  - 2) Capture
- + mod rada kada se T2 koristi kao generator programabilnog clock izlaza
- Radom T2 se upravlja pomoću SFR-ova **TH2**, **TL2**, **T2MOD**, **T2CON**, **RCAP2H** i **RCAP2L**



T2CON (C8h)

Bit	Ime
7	TF2
6	EXF2
5	RCLK
4	TCLK
3	EXEN2
2	TR2
1	C/T2#
0	CP/RL2#

--> overflow flag (mora se programski resetovati)

--> poseban flag koji može generisati prekid u nekim modovima

koriste se za podešavanje brzine prijema i slanja podataka preko asinhronne serijske komunikacije

--> ima različite funkcije za svaki mod

0 => T2 OFF  
1 => T2 ON

0 => brojanje mašinskih ciklusa  
1 => brojanje eksternih događaja

0 => autoreload mod  
1 => capture mod

T2MOD (C9h)

Bit	Ime
7-2	-
1	T2OE
0	DCEN

0 => pin P1.0 (T2) se koristi kao ulaz tajmera 2 ili kao I/O opšte namjene  
1 => pin P1.0 se koristi kao clock izlaz

0 => samo increment tajmera  
1 => omogućen decrement u autoreload modu

### T2 autoreload mod:

- da bi radio mora biti **TR2 = 1** ;
- može brojati MC ili eksterne događaje (**C/T2#**) i moguće je dekrementirati sadržaj TH2 i TL2;
- u T2H i T2L vrijednosti se reloaduju iz RCAP2H i RCAP2L ;

#### *I. kada je DEC = 0 (decrement onemogućen)*

- EXEN2 = 0 => T2EX pin je nebitan i samo flag TF2 može generisati prekid (flag EXF2 ne može generisati prekid)
- EXEN2 = 1 => omogućena funkcionalnost pina T2EX, tj. na silaznu ivicu signala sa T2EX setuje se flag EXF2 koji generiše prekid

#### *II. kada je DEC = 1 (moguće decrementiranje)*

- T2EX = 0 => TH2 i TL2 se inkrementiraju
- T2EX = 1 => TH2 i TL2 se dekrementiraju (sve do vrijednosti koje su upisane u RCAP2h i RCAP2L, a prilikom sljedećeg dekrementa dolazi do tzv. *underflowa* koji generiše prekid; nakon underflowa u oba registra TH2 i TL2 se autoreloaduju vrijednosti 0xFFh)

### T2 capture mod:

- da bi radio mora biti **TR2 = 1** i **CP/RL=1** ;
- pri capture-u sadržaj koji se trenutno nalazi u TH2 i TL2 se prepisuje u RCAP2H i RCAP2L usljed nekog događaja (silazne ivice signala) na T2EX pinu ;

#### *I. kada je EXEN2= 0*

- capture ne radi (tj. ne čuvaju se vrijednosti u RCAP2H i RCAP2L)
- prekid se dešava samo na overflow T2H i T2L i setovanje TF2

#### *II. kada je EXEN2= 1*

- capture omogućen
- prekid moguć setovanjem TF2 i EXF2 flagova (TF2 se setuje na overflow događaj, a EXF2 na silaznu ivicu signala sa pina T2EX prilikom čega dolazi do capture-a)

## 19. Mikrokontroler AT89C51RC2 – eksterni prekidi

Eksterni prekidi su prekidi koje generiše neki od ulaznih pinova mikrokontrolera. Služe da bi se što prije odreagovalo na neki signal sa ulaznih pinova.

Na AT89C51RC2 postoje 2 eksterna prekida. Da bi se omogućili eksterni prekidi neophodno je setovati SFR-ove **EA** i **EX0** (za eksterni prekid 0) i/ili **EX1** (za eksterni prekid 1).

Konfigurišu se pomoću SFR-a **TCON**:

TCON (88h)

Bit	Ime
3	IE1
2	IT1
1	IE0
0	IT0

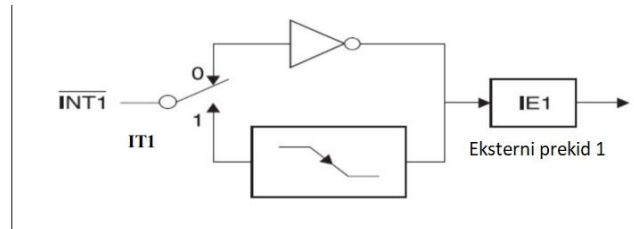
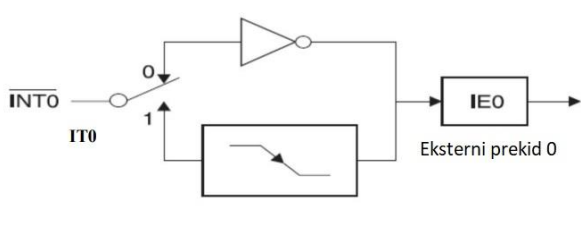
} eksterni prekid 1 (podešavanje isto kao za eksterni prekid 0)

--> flag koji generiše prekid kada je setovan

0 => eksterni prekid se dešava na **logičku 0** signala sa pina  
1 => eksterni prekid se dešava na **silaznu ivicu** signala sa pina

} bitovi za eksterni prekid 0

Šeme eksternih prekida:



## 20. Mikrokontroler AT89C51RC2 – Watchdog tajmer

Watchdog tajmer služi da omogući povratak sistema u normalan rad u slučaju kada sistem prestane raditi zbog neke softverske greške (blokira onaj dio programa, tj. koda gdje se nalazi greška).

Radi na taj način što resetuje kontroler i počinje izvršavanje programa iz početka ukoliko vrijeme proteklo od kada je program posljednji put bio na referentnom mjestu u kodu prevazilazi definisano vrijeme watchdog tajmera.

Konfigurisanje se vrši pomoću SFR-ova **WDTRST** (za pokretanje i reset) i **WDTPRG** (za definisanje vremena).

Pokretanje watchdog tajmera se vrši upisom vrijednosti **0x1Eh**, a zatim **0xE1h** u WDTRST (jedna za drugom). Nakon pokretanja tajmer počinje brojati MC (mašinske cikluse) pomoću 14-bitnog brojača.

Da bi se resetovanje mikrokontrolera spriječilo prije nego što tajmer odbroji definisano vrijeme (tj.  $2^{14} - 1$  mašinskih ciklusa) u nekom dijelu programa (koji predstavlja referentno mjesto) potrebno je u registar WDTRST upisati vrijednosti **0xE1h**, a potom **0x1Eh** (isto zaredom, ali u obrnutom redoslijedu).

Broj mašinskih ciklusa koji se odbrojava je moguće povećati, a za to se koristi registar WDTPRG. Željeni broj se podešava upisom u nulti, prvi i drugi bit WDTPRG-a.

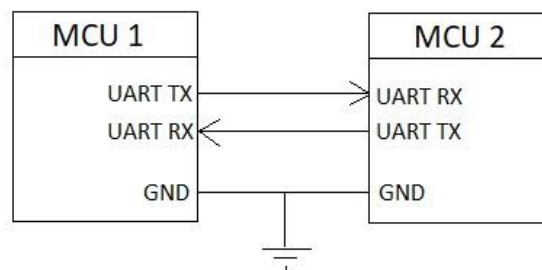
WDTPRG(A7h)

Bit	Ime
7-3	-
2	S2
1	S1
0	S0

S2	S1	S0	Time-out watchdog tajmera
0	0	0	$(2^{14}-1)$ mašinskih ciklusa
0	0	1	$(2^{15}-1)$ mašinskih ciklusa
0	1	0	$(2^{16}-1)$ mašinskih ciklusa
0	1	1	$(2^{17}-1)$ mašinskih ciklusa
1	0	0	$(2^{18}-1)$ mašinskih ciklusa
1	0	1	$(2^{19}-1)$ mašinskih ciklusa
1	1	0	$(2^{20}-1)$ mašinskih ciklusa
1	1	1	$(2^{21}-1)$ mašinskih ciklusa

## 21. Mikrokontroler AT89C51RC2 – UART

UART (universal asynchronous receiver,transmitter) služi za asinhronu serijsku komunikaciju između mikrokontrolera. Putem UARTa omogućeno je i primanje i slanje podataka. Podaci se šalju u bajtima (bajt po bajt).



UART nema posebnu liniju za clock koja određuje vrijeme primanja i slanja podataka, već je unaprijed poznata brzina kojom će se podaci razmijenjivati između mikrokontrolera.

Ne postoji mogućnost da jedan mikrokontroler javlja drugom kada će mu poslati podatke.

Za uspješan prenos podataka UARTom **neophodno je da se zna** kada kreće **početak** komunikacije! Komunikacija počinje **START bitom** koji ne predstavlja podatak za prenos. START bit traje (postavljen je na logičku 0) onoliko koliko traje razmjena 1 bita.

Prijem podataka u odnosu na slanje kasni 0,5 takta.

Nakon slanja posljednjeg (8.) bita linija se postavlja na logičku jedinicu što predstavlja **STOP bit**. STOP bit treba da traje koliko traje slanje 1 bita. Ukoliko je STOP bit različit od 1 sigurno je došlo do greške pri komunikaciji.

Da bi došlo do razmjene podataka potrebno je definisati mod rada i brzinu (baud rate). Nakon konfiguracije sve što je potrebno za razmjenu je da se vrijednost upiše u odgovarajući SFR (**SBUF**), ako se šalje, odnosno da se pročita iz istog SFRA, kada se prima.

Konfiguracija moda UARTa se vrši pomoću **SCON** i **PCON** registara. Konfiguracija brzine (baud rate-a) moguća je pomoću INTERNOG baud rate generatora i preko overflowa tajmera T1 i T2. Kada se tajmeri koriste kao baud rate generatori neophodno ih je podesiti u atoreload mod i isključiti njihov prekid!

Za podešavanje brzine internog baud rate generatora koristi se registar **BDRCON** i **BRL**.

#### SCON (98h)

Bit	Ime	
7	SM0/FE	--> zavisi od SMOD0 bita registra PCON <ul style="list-style-type: none"> <li>SMOD0 = 0 =&gt; SM0 (koristi se za podešavanje moda)</li> <li>SMOD0 = 1 =&gt; FE (flag koji se setuje kada je primljen neispravan STOP bit)</li> </ul>
6	SM1	--> za podešavanje moda
5	SM2	--> za multiprocesorsku komunikaciju (NEBITAN)
4	REN	--> receive enable (da bi se podatak mogao primiti mora biti setovan)
3	TB8	} kontrolni (parity) bit za slanje/prijem 9. bita u modu 2 i 3 (9-bitnim modovima)
2	RB8	
1	TI	--> transmit flag (setovan kada je bajt uspješno poslat)
0	RI	--> receive flag (setovan kada je bajt uspješno primljen)

} generišu isti prekid (moraju se programski resetovati)

Modovi rada:

SM0	SM1	Mod rada	Objašnjenje	Baud Rate
0	0	0	8-bit Shift Register	$F_{osc}/12$ (X1 mod) ili $F_{osc}/6$ (X2 mod)
0	1	1	8-bitni UART	Promjenjiv
1	0	2	9-bitni UART	$F_{osc}/64$ (X1 mod) ili $F_{osc}/32$ (X2 mod) (*)
1	1	3	9-bitni UART	Promjenjiv

#### PCON (87h)

Bit	Ime	
7	SMOD1	--> obezbjeđuje 2x brži baud rate u modu 1, 2 i 3
6	SMOD0	<ul style="list-style-type: none"> <li>0 =&gt; SCON[7] = SM0</li> <li>1 =&gt; SCON[7] = FE</li> </ul>

( SCON[7] - 7. bit registra SCON )

## BDRCON

Bit	Ime
4	BRR
3	TBCK
2	RBCK
1	SPD
<del>0</del>	<del>SRG</del>

0 => interni baud rate OFF  
1 => interni baud rate ON

izbor da li se interni baud rate koristi za slanje ili primanje podataka

0 => spori interni baud rate generator  
1 => brzi interni baud rate generator

--> nebitan

Šema i način računanja baud rate-a u modovima 1 i 3:

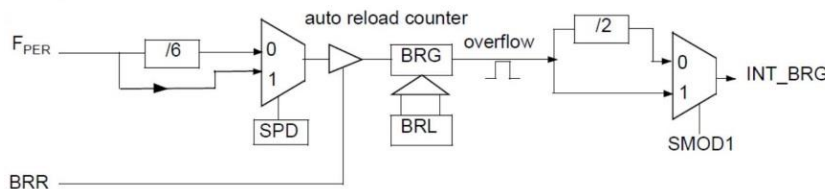
### a) definisanje pomoću tajmera T1 i T2



$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \cdot \left( \frac{F_{\text{PER}}}{6 \cdot [256 - \text{TH1}]} \right)$$

$$\left( \text{TH1} = 256 - \frac{2^{\text{SMOD1}} \cdot F_{\text{PER}}}{32 \cdot 6 \cdot \text{Baud Rate}} \right)$$

### b) definisanje pomoću internog baud rate generatora



$$\text{Baud Rate} = \frac{2^{\text{SMOD1}}}{6^{(1-\text{SPD})} \cdot 32} \cdot \left( \frac{F_{\text{PER}}}{[256 - \text{BRL}]} \right)$$

$$\left( \text{BRL} = 256 - \frac{2^{\text{SMOD1}} \cdot F_{\text{PER}}}{6^{(1-\text{SPD})} \cdot 32 \cdot \text{Baud Rate}} \right)$$

UPIS i ČITANJE sa serijskog porta se vrši pomoću SFRa **SBUF**!

- slanje: `MOV SBUF, #9`
- čitanje: `MOV A, SBUF`

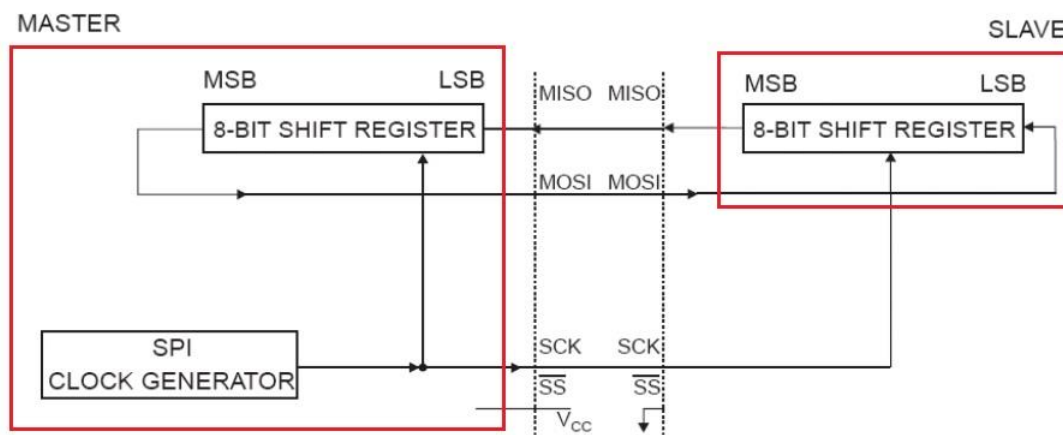
AT89C51RC2 nema ni ulazni ni izlazni bafer za serijsku komunikaciju što znači da ih je neophodno programski napraviti. U baferu se čuvaju podaci koji stižu sa serijske komunikacije. Obično se implementira *kružni bafer*.

Za asinhronu serijsku komunikaciju postoje *protokoli* koji definišu način razmjene podataka. Mogu biti standardni i custom, a na osnovu tipa podataka koji se šalju ASCII i binarni.

## 22. Mikrokontroler AT89C51RC2 – SPI

SPI omogućava razmjenu podataka sinhronom serijskom komunikacijom sa velikom brzinom između mikrokontrolera i periferija ili između više mikrokontrolera. Osnovne karakteristike SPI interfejsa su sljedeće:

- ❑ Full-Duplex sinhronizovani prenos podataka sa 3 žice (moguće istovremeno primiti i slati podatke)
- ❑ Master ili Slave mod rada
- ❑ Velika brzina
- ❑ Prenos podataka sa prvim LSB ili prvim MSB (AT89C51RC2 je sa prvim most significant bitom - MSB)
- ❑ Više programabilnih brzina prenosa
- ❑ Generisanje prekida na kraju prenosa



SFR koji se koriste su: **SPCON**, **SPSTA**, **SPDAT** (analogna funkcija SBUF kod asinhronne komunikacije) i za podešavanje prekida EA (globalna dozvola), ESPI (dozvola SPI prekida) i IPH1 i IPL1 za podešavanje prioriteta SPI prekida.

Pinovi koji se koriste za žice su P1.1 (SS – slave select), P1.5 (MISO – master input, slave output), P1.6 (SCK – SPI clock) i P1.7 (MOSI – master output, slave input).

**SPCON (C3h)**

Bit	Ime	
7	SPR2	--> uz SPR0 i SPR1 služi za podešavanje brzine
6	SPEN	--> kada se setuje pinovi se povezuju na žice za SPI (neophodna)
5	SSDIS	--> slave select disable (kada je 1 SS se ne koristi ni za slave)
4	MSTR	<div> <div>0 =&gt; mikrokontroler je slave</div> <div>1 =&gt; mikrokontroler je master</div> </div>
3	CPOL	--> clock polarity } konfiguriraju način rada
2	CPHA	
1	SPR1	} podešavanje brzine
0	SPR0	

CPOL  $\begin{cases} 0 \Rightarrow \text{prva ivica na početku SPI komunikacije je uzlazna} \\ 1 \Rightarrow \text{prva ivica na početku SPI komunikacije je silazna} \end{cases}$  } mora biti isto podešeno na masteru i slaveu

CPHA  $\begin{cases} 0 \Rightarrow \text{očitavanje vrijednosti na 1. ivici, a postavljanje na 2. ivici clock signala} \\ 1 \Rightarrow \text{postavljanje vrijednosti na 1. ivici, a očitavanje na 2. ivici clock signala} \end{cases}$  } (SS je neophodan za sinhronizaciju, jer se bit postavlja na njegovoj silaznoj ivici da bi se pojavio prije očitavanja)  
(Sinhronizacija preko clock signala, SS nije neophodan)

Podešavanje brzine:

SPR2	SPR1	SPR0	SCK
0	0	0	$F_{CLK PERIPH}/2$
0	0	1	$F_{CLK PERIPH}/4$
0	1	0	$F_{CLK PERIPH}/8$
0	1	1	$F_{CLK PERIPH}/16$
1	0	0	$F_{CLK PERIPH}/32$
1	0	1	$F_{CLK PERIPH}/64$
1	1	0	$F_{CLK PERIPH}/128$
1	1	1	Invalid



SPSTA je statusni registar. 7. bit SPSTA registra je **SPIF** flag koji generiše prekid. SPIF se ne resetuje ni harverski, ni softverski nego automatski nakon što se pročita sadržaj SPSTA i pristupi SPDAT registru (čitanje ili upis u SPDAT) !

SPSTA (C4h)	
Bit	Ime
7	SPIF
6	WCOL
5	SSERR
4	MODF

--> setuje se nakon prenosa jednog bajta (generiše prekid ako je omogućen)

} statusni flagovi koji se setuju ako se desi određena greška pri SPI komunikaciji

Vrste konfiguracija mastera sa više slave-ova:

1) regularna

- master uvijek razmjenjuje podatke samo sa 1 slave-om ;
- za svaki slave postoji poseban SS (slave select) pin ;

2) Daisy chain

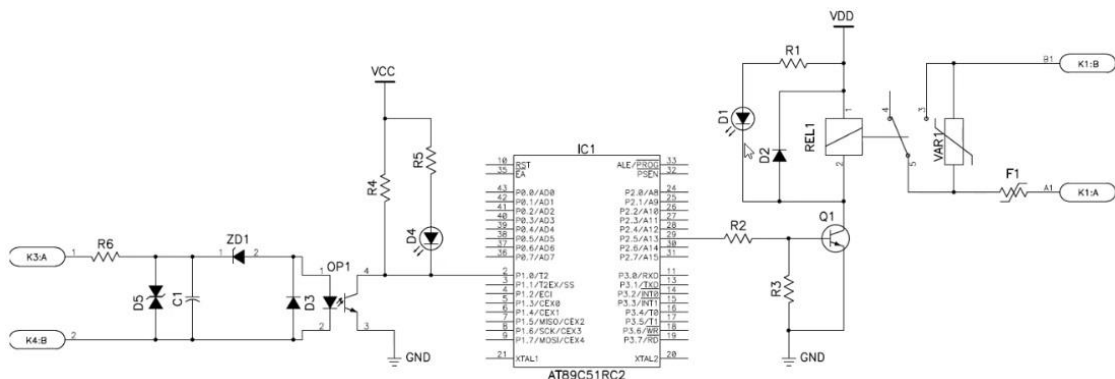
- master ima samo jedan SS pin ;
- izlaz mastera je ulaz 1. slave-a, izlaz 1. slave-a je ulaz 2. slave-a ... do posljednjeg slave-a čiji je izlaz povezan na ulaz mastera ;

## 23. Diskretni ulazi i izlazi – direktno povezivanje

Šema ulaza i izlaza (moduli uređaja) se povezuju direktno na pinove mikrokontrolera. Šema veze direktnog povezivanja je veoma jednostavna (mikrokontroler sa 4 porta sa po 8 linija na svakom, može da se poveže na maksimalno 32 ulaza i izlaza ukupno). Takođe je jednostavno i *programiranje* (čitanjem pina/porta dobijamo stanje ulaza, a upisom vrijednosti na pin/port mijenjamo stanje na izlazima mikrokontrolera). To znači da se i čitanje i pisanje realizuje jednom asemblerskom instrukcijom.

Prednosti direktnog povezivanja: brzina čitanja i pisanja i jednostavnost.

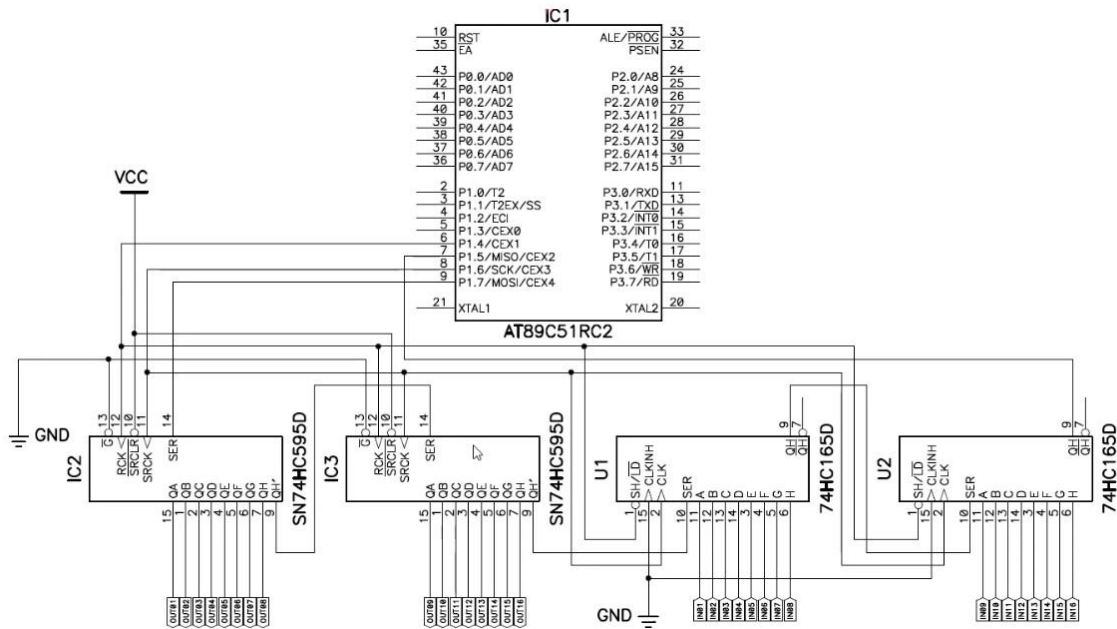
Nedostaci: problem pri proširenju željenog broja ulaza i izlaza (za veći broj I/O potreban je fizički veći MCU) i korištenjem velikog broja pinova smanjuje se broj slobodnih pinova za ostale periferije (tastaturu, displej...)



## 24. Diskretni ulazi i izlazi –povezivanje preko SPI

Pri povezivanju preko SPI-a koriste se MOSI, MISO, SCK i LOAD linije. LOAD linija služi za učitavanje stanja u periferije.

Koristi se za srednji broj I/O (do 100, maksimalno 200). Uglavnom za PLC-ove.





Nedostaci:

- način projektovanja je kompleksniji ;
- da se promijeni samo 1 bit na izlazu ili pročita samo 1 pin ulaza mora se “protjerati” čitav SPI lanac ;
- porastom broja shift registara povećava se dužina linije clock signala (povećanjem linije SCK-a linija prestaje biti nulte otpornosti i počinje se ponašati kao karakteristični vod, zbog toga može doći do refleksije signala i gubitka podataka; problem se rješava smanjenjem brzine komunikacije ili postavljanjem hardverskih drajvera koji regulišu strujne nivoe signala) ;

Primjer:

- za 4 shift registra potrebno je slati 5 bajtova
- dodatni bajt predstavlja **CONTROL** bajt (MCU provjerava da li je poslao kontrolni bajt jednak primljenom)

- sekvenca slanja bajtova iz MCU-a:

1. bajt	2. bajt	3. bajt	4. bajt	5. bajt
CTRL	dummy1	dummy2	data_for_IC3	data_for_IC2

- prije početka komunikacije je neophodno stanje ulaza prepisati u shift registre (U1 i U2) !
- sadržaj izlaznih shift registara (IC2 i IC3) na početku komunikacije je prazan

- tok slanja bajtova:

korak	IC2	IC3	U2	U3	IC1 (MCU)
0	-	-	data_from_U2	data_from_U3	-
1	CTRL	-	data_from_U2	data_from_U3	-
2	dummy1	CTRL	data_from_U2	data_from_U3	-
3	dummy2	dummy1	CTRL	data_from_U2	data_from_U3
4	data_for_IC3	dummy2	dummy1	CTRL	data_from U2 data_from_U3
5	data_for_IC2	data_for_IC3	dummy2	dummy1	CTRL data_from U2 data_from_U3

- ako je primljeni CTRL bajt jednak poslatom tada se iz latch-eva IC2 i IC3 podaci (data\_for\_IC2 i data\_for\_IC3) prebacuju na fizičke izlaze (tako što LOAD linija generiše signal)

## 25. Diskretni ulazi i izlazi – povezivanje preko adresnog prostora

Služi za velike sisteme sa velikim brojem ulaza i izlaza (preko 1000). To su tzv. **“Wrack”** sistemi kod kojih postoji centralni procesorski sistem sa razvučenim magistralama na backplane-u.

Backplane je štampana ploča sa naslaganim konektorima na koje se dodaju periferije u njegov adresni prostor.

Implementacija se vrši putem **eksternog** adresnog prostora (eksterne memorije) – MOVX naredbom.

Portovi za adresiranje eksternih uređaja su **P0** i **P2**.

P0 predstavlja multipleksiranu magistralu adrese i magistralu podataka. To znači da se u jednom dijelu mašinskog ciklusa na P0 nalazi adresa, a u drugom podatak. P2 se koristi kao adresni port (samo za adresu).

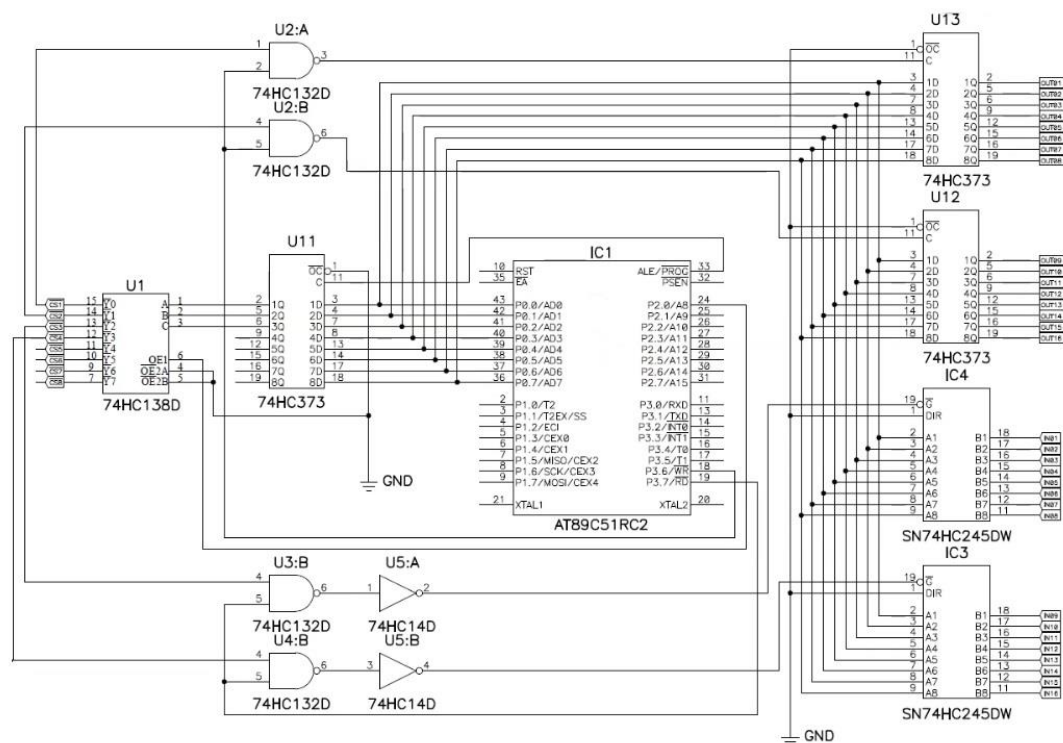
Putem P0 i P2 moguće je adresirati do 16 bita što odgovara 65536 memorijskih lokacija veličine 8 bita. MCU na ovaj način može napraviti 65536\*8 ulaza i izlaza.

Na multipleksiranoj adresnoj i data magistrali (**na P0**) se u **prvom ciklusu nalazi adresa, a u drugom podatak**. Adresu iz P0 je potrebno upisati u latch. Putem signala na pinu ALE (address latch enable) latch dobija informaciju kada da sačuva donji dio adrese sa P0.

U drugom dijelu ciklusa u P0 se nalazi podatak. Ukoliko je on izlazni, nalaziće se na ulazu svih periferija (integrisanih kola). Sva integrisana kola imaju svoju adresu putem kojih ih mikrokontroler raspoznaje i izvršava čitanje podataka ili upis podataka na njih.

Setovanjem odgovarajućeg CS-a (chip select-a) bira se periferija (tj. čip), a setovanjem RD ili WR pina govori se da li se podatak čita iz periferije ili se upisuje na nju.

Dok se čita sa jednog čipa (periferije, integrisanog kola) svi ostali čipovi se nalaze u stanju visoke impedanse što znači da se sistem ponaša kao da ne postoje.

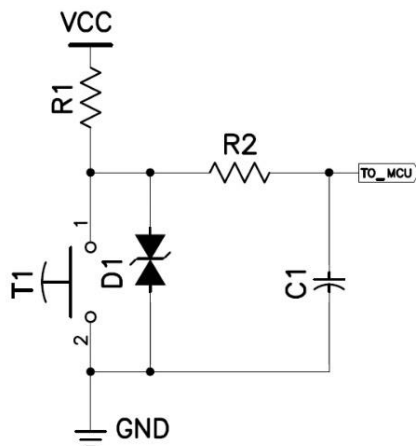


## 26. Taster – povezivanje, bounce i debounce

Tastatura predstavlja skup 1 ili više tastera (1 taster može imati više funkcionalnosti u zavisnosti od npr. dužine trajanja držanja tastera).

Taster je elektro-mehanička komponenta. Sastoji se od 2 kontakta između kojih dolazi do kratkog spoja pilikom pritiskanja. Na tasteru se nalazi opruga koja ima koeficijent elastičnosti  $k$ . Prilikom puštanja tastera opruga povlači nazad pločicu tastera i drži ju odmaknutu od kontakata, čime onemogućava protok struje (ovo je slučaj za NO taster – normalno otvoreni, za NC taster logika je obrnuta).

Povezivanje:



Taster se uglavnom povezuje na masu (jedan kraj na GND, a drugi na ulaz MCU-a).

Kraj koji je povezan na ulaz MCU-a se putem pullup (R1) otpornika povezuje na napajanje. R1 obezbjeđuje stabilno stanje na ulazu MCU-a kada taster nije pritisnut, tj. sprječava da se ulaz ponaša kao prijemna antena.

Tastatura je obično fizički dislocirana od mikrokontrolera, dok se pullup otpornik uglavnom nalazi tik uz pin mikrokontrolera.

Zbog koeficijenta elastičnosti sistem se ponaša kao sistem 2. reda (ima konjugovano-kompleksne polove). Koeficijent prigušenja predstavlja koeficijent elastičnosti. Zbog toga, prilikom pritiskanja tastera dolazi do treperenja pločice, tj. javlja se tzv. **bounce** (treperenje). Pojava bounce-a predstavlja prvi problem kod tastera.

Drugi problem nastaje pri rukovanju jer može doći do elektro-statičkog pražnjenja između rukovaoca i izolacije na tasteru. Da bi se ova pojava spriječila, na ulaz MCU-a postavlja se TRANSIL (D1).

Tehnika otklanjanja bounce-a (oscilacija usljed prisustva koeficijenta elastičnosti) se naziva **debounce**.

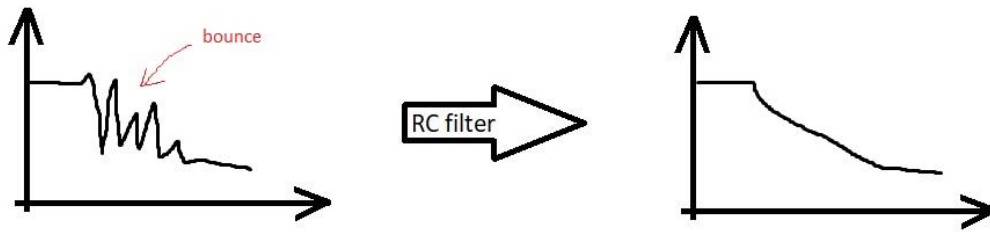
Vrste implementacija debounce-a:

### 1) softverski

- zasniva se na ispitivanju stanja tastera u određenom vremenskom intervalu i startovanja tajmera koji mjeri dužinu trajanja tog stanja sve dok se ono ne promijeni (vrijeme provjere je obično 1-10ms)

## 2) hardverski

- realizuje se dodavanjem RC kola koje predstavlja filter visokofrekvencijskih smetnji
- ulaz mora biti tipa šmit triger (šmit triger je kolo koje ima histerzis čime se povećava imunizacija na smetnje i izbjegava lažno ukidanje ulaza)



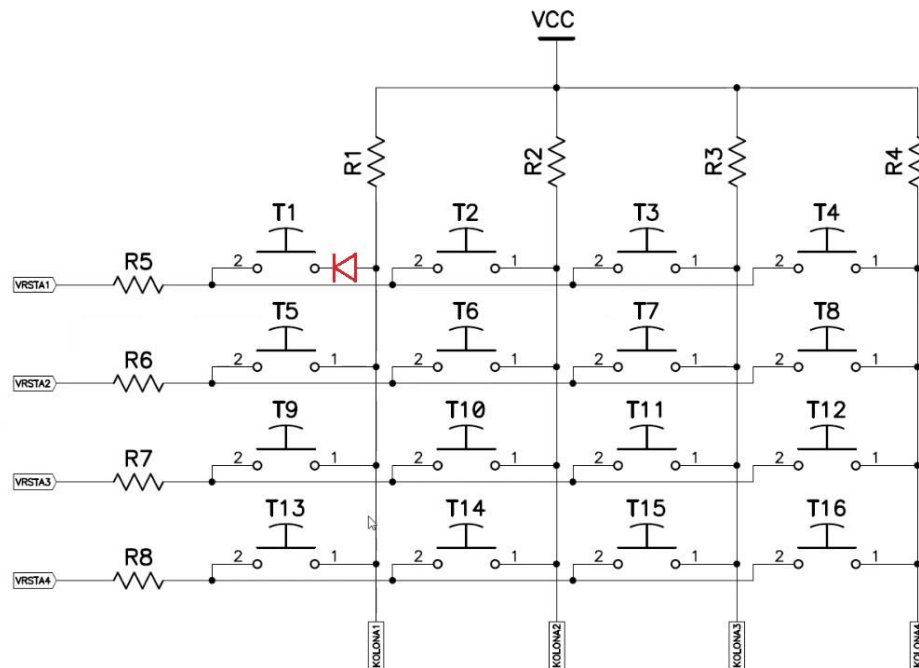
## 3) hibridni

- kombinacija softverskog i hardverskog

## 27. Matrična tastatura

Kada želimo da povežemo veći broj tastera na MCU, a da nam pri tom ostane više slobodnih pinova - koristi se matrična tastatura.

Ako sa  $m$  predstavimo broj vrsta, a sa  $n$  broj kolona matrične tastature, tada za  $m \cdot n$  tastera, matrična tastatura omogućuje smanjenje potrebnih pinova sa  $m \cdot n$  na  $m+n$ .



R1, R2, R3 i R4 predstavljaju pullup otpornike. VRSTE predstavljaju ulazne pinove, a KOLONE izlazne pinove.

Selektovanje vrste se vrši upisom logičke 0 u odgovarajući pin vrste, a logičke 1 u sve pinove za preostale vrste. Nakon što smo npr. selektovali vrstu 1 (VRSTA1 = 0), čitanjem stanja kolona - čitamo vrijednost tastera iz selektovane vrste i kolone koju trenutno čitamo.

Pri završetku čitanja vrste prvo je potrebno deselektovati vrstu, a tek potom izabrati sljedeću vrstu.

**Scan ciklus** matrične tastature obuhvata selekciju svih vrsta.

**Prednosti:**

- jednostavnost ;
- ne zahtjeva mnogo dodatnih komponenti za fizičku implementaciju ;

**Mane:**

- podložnost elektro-statičkom pražnjenju pri rukovanju (rješenje: dodavanjem transila na portove MCU-a) ;
- **fantomska detekcija tastera** (nema softversko rješenje, može isključivo hardverskom izmjenom dodavanjem diode na red sa svakim tasterom) ;

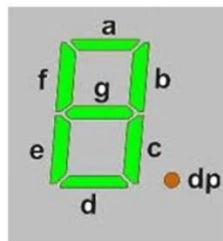
## 28. Sedmosegmentni LED displej – povezivanje sa MCU – zajednička anoda i zajednička katoda – normalan režim rada

LED displej je displej čiji je osnovni element funkcionalnosti LE dioda. LE dioda emituje svjetlost u vidljivom spektru (pad napona na diodi iznosi od 1,6 do 2,8 V).

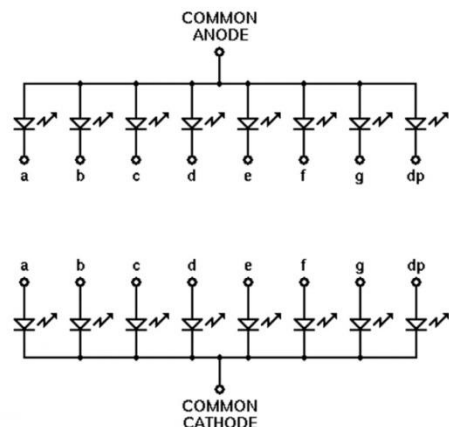
Postoje uskopojasne i širokopojasne LED. Za displeje se koriste uskopojasne.

Konfiguracije LED displeja:

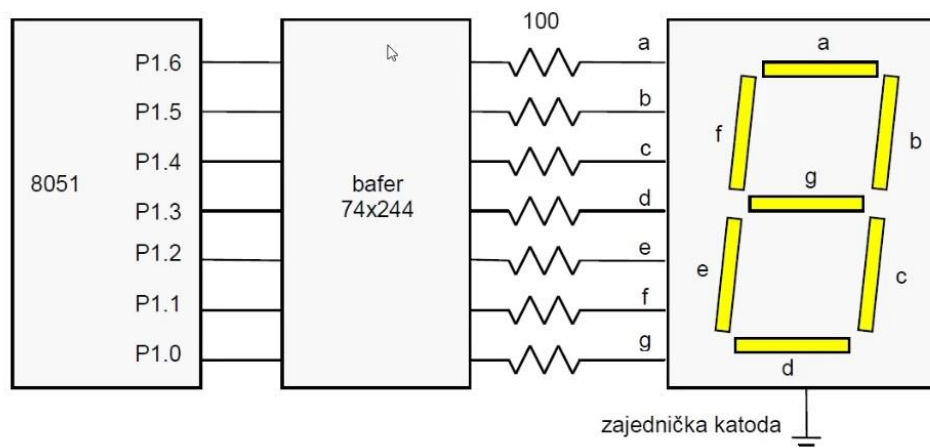
- 1) sa zajedničkom ANODOM (Vcc) i
- 2) sa zajedničkom KATODOM (GND)



7 segmenata  
(a,b,c,d,e,f,g)  
+  
1 podrazumijevani  
(dp - tačka)



Aktivacija segmenata kod konfiguracije sa zajedničkom anodom se vrši upisivanjem logičke 0 u odgovarajući segment, a kod konfiguracije sa zajedničkom katodom upisivanjem logičke 1.



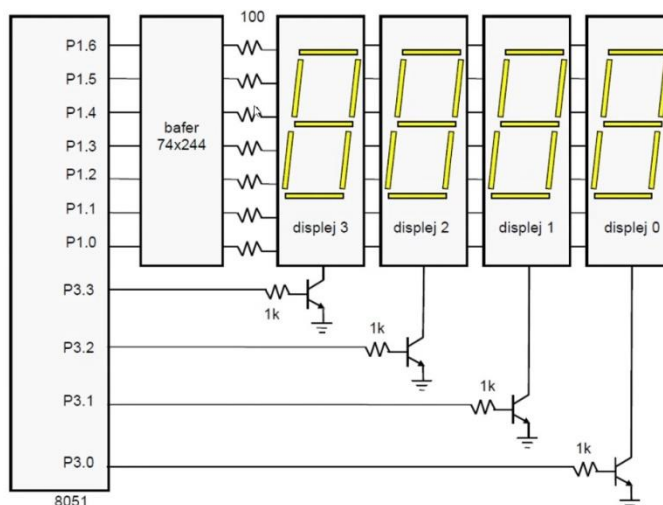
Uloga bafera (koji je kod displeja digitalna komponenta – ima samo stanja logičke 0 i 1) je povećavanje strujnog kapaciteta porta mikrokontrolera.

Otpornici na segmentima ograničavaju i kontrolišu struju koja teče kroz njih. Jačina struje odgovara jačini osvijetljaja segmenata.

*Mana:* za veći broj LED displeja mora se zauzeti veliki broj portova mikrokontrolera.

## 29. Sedmosegmentni LED displej – povezivanje sa MCU – zajednička anoda i zajednička katoda – multiprocesorski režim rada

Multipleksirani režim rada se koristi da bi se uštedio broj portova mikrokontrolera koji se koriste. Zasniva se na perzistenciji vida (tromosti ljudskog oka). Sve što gledamo, gledamo sa niskopropusnim filterom oka učestanosti od približno 50 Hz (što znači da ne vidimo promjene brže od 50 Hz).



Segmenti svih cifara su kratko spojeni (npr. "a" segment 1. cifre je spojen na "a" segmente druge, treće i četvrte cifre).

U svakom trenutku je uključena jedna od cifara. Selekcija cifre koja je uključena se vrši putem kanala za selekciju (NPN tranzistora).

Scan ciklus osvježavanja predstavlja vrijeme osvježavanje displeja. Vrijeme osvježavanja displeja je jednako zbiru vremena osvježavanja svih cifara.

Primjer: frekvencija displeja je 100 Hz

- $f = 100 \text{ Hz} \Rightarrow t = 10 \text{ ms}$  (vrijeme osvježavanja displeja)
- 4 cifre za prikaz  $\Rightarrow$  vrijeme osvježavanja cifre  $= t/4 = 2,5 \text{ ms}$
- vrijeme osvježavanja cifre  $<$  vrijeme osvježavanja displeja

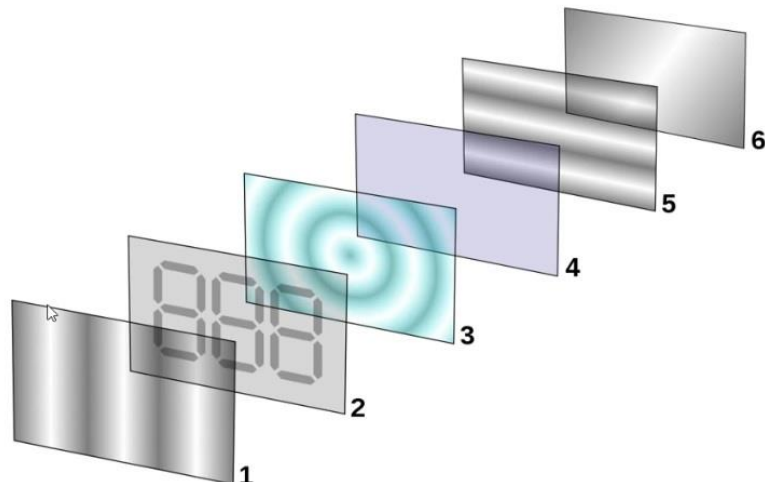
### 30. Princip rada displeja sa tečnim kristalima – LCD

LCD – Liquid Crystal Display (bazira se na tehnologiji tečnih kristala).

Dijelovi:

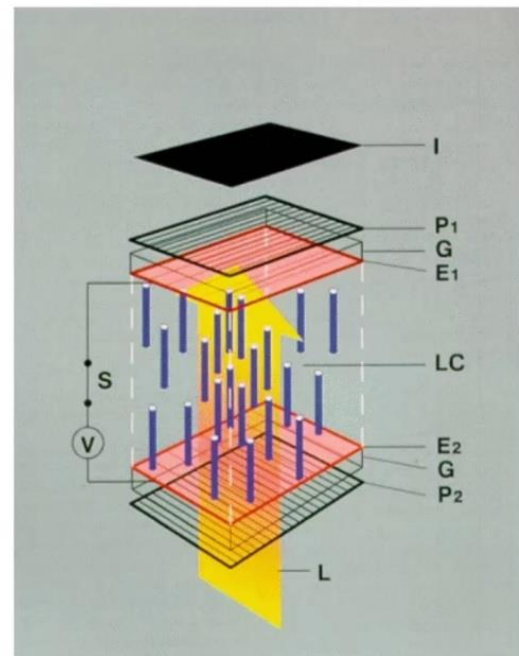
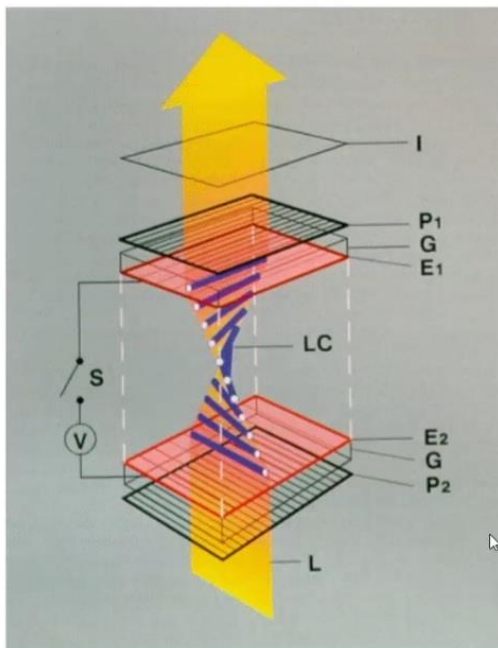
- 1) gornji **polarizator** svjetlosti (vertikalni) – polarizator propušta svjetlost samo u 1 smjeru,
- 2) gornje **providne elektrode**,
- 3) **nematički** tečni kristal (predstavlja treći polarizator),
- 4) donje **providne elektrode**,
- 5) donji **polarizator** svjetlosti (horizontalni) i
- 6) donje **ogledalo** (providna staklena ploča).

Svjetlost prolazi u smjeru od 1 -> 6 ili od 6 -> 1.



### Princip rada:

- tečni kristal je molekularna struktura koja je sa oba kraja zakačena za **elektrode** ;
- iznad elektrode se nalazi **vertikalni** ili **horizontalni polarizator** ;
- sam **tečni kristal vrši polarizaciju svjetlosti u nepolarizovanom obliku** (dok nema napajanja na elektrodama on u nepolarizovanom obliku vrši polarizaciju iz horizontalne u vertikalnu ili obrnuto)
  - **DOK NEMA NAPAJANJA** i pozadinsko svjetlo krene da teče, ono prvo prolazi kroz horizontalni polarizator, zatim teče kroz tečni kristal gdje se vrši polarizacija (uvijanje) iz horizontalnog u vertikalno ;  
vertikalna svjetlost prolazi kroz vertikalni polarizator napolje i ono se vidi kao osvijetljaj (osvijetljaj predstavlja uključeno stanje pixela) ;
  - **KADA SE NAPAJANJE DOVEDE NA ELEKTRODE** dolazi do depolarizacije tečnog kristala, mijenja se kristalna rešetka samog kristala i više ne dolazi do polarizacije iz horizontalne u vertikalnu svjetlost;  
na taj način prodor svjetlosti je spriječen i **pixel je isključen** (crn);
- **kvalitet displeja** se ogleda u tome koliko je crna boja pixela stvarno crna ;
- problem sa LCD displejom je to što može doći do njegovog oštećenja tokom vremena, jer napon koji se dovodi na elektrode nije sinusoida (nego tzv. balansirani pravougani signal) ;





### 31. Princip rada LCD modula i povezivanje sa MCU – 8-bitni režim rada

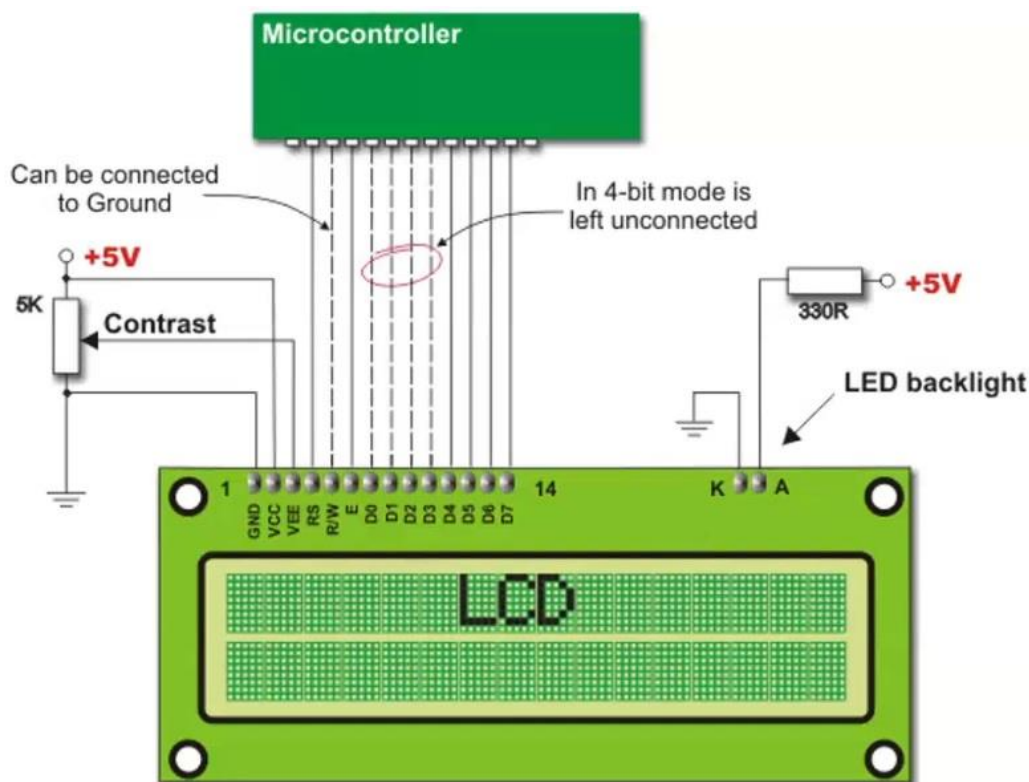
Displej ima data magistralu (magistalu podataka) od 8 linija (**D0 – D7**). Osam linija za korištenje magistrale podataka znači da je **jedna jedinica transakcije 8 bita**, tj. 1 bajt.

Ostale linije koje se koriste za komunikaciju između displeja i mikrokontrolera su **upravljačke linije** (linije putem kojih MCU upravlja registrima LCD modula).

Upravljačku liniju čine 3 linije:

- 1) **RS** (register select)
  - govori da li da pristupamo **registru za kontrolu displeja** ili **registorima za upis podataka** koje treba prikazati na displeju ;
- 2) **ENABLE**
  - govori da li su podaci na magistrali podataka stabili i validni ;
- 3) **R/W**
  - opcionalna linija ;
  - govori da li čitamo ili upisujemo podatak na displej ;
  - ako se ne koristi povezuje se na GND i podrazumijeva se da se samo upisuje na displej ;

Preostale linije služe za **napajanje** i **pozadinsko osvjetljenje displeja** (pomoću LE diode).



### **32. Princip rada LCD modula i povezivanje sa MCU – 4-bitni režim rada**

U slučaju da je mikrokontroler previše opterećen brojem zauzetih linija (pinova) i ne može obezbjediti 8 slobodnih linija za LCD, moguće je koristiti samo gornje 4 linije data magistrale displeja (D4 – D7). U tom slučaju 4 donje linije se povezuju na Vcc.

Ostatak rada 4-bitnog režima je isti kao 8-bitni osim što se bajt (podatak za upis na displej) sa mikrokontrolera na LCD sada mora slati u 2 dijela (po 4 bita). Zbog toga mora postojati sinhronizacija između mikrokontrolera i LCD-a da bi znali da je displej prihvatio i prvu i drugu polovinu bajta.

### **33. Inicijalizacija LCD modula**

Inicijalizacija LCD modula se vrši i za 4-bitni i za 8-bitni režim rada.

Koraci inicijalizacije (8-bitni režim):

- 1) paljenje
- 2) pauza (15ms) za inicijalizaciju registara
- 3) inicijalizacija u 8-bitnom režimu
- 4) pauza (4.1ms)
- 5) 8-bitni režim
- 6) pauza (100μs)
- 7) 8bitni režim
- 8) konfiguracija željenog rada displeja

Inicijalizacija u 4-bitnom režimu takođe kreće u 8-bitnom režimu, a tek kada dođe do konfiguracije rada displeja (8. korak) prebacuje se u 4-bitni režim. Kada se prebaci u 4-bitni režim tada se komande šalju iz 2 dijela (prvo viši dio bajta, zatim niži dio bajta).

### **~~34. RS232 komunikacija~~**

### 35. RS485 komunikacija – karakteristike, konfiguracija i terminacija

RS485 je standardna komunikacija koja se koristi u procesnoj industriji na nivou komunikacije između više mikrokontrolera i između mikrokontrolera i senzora.

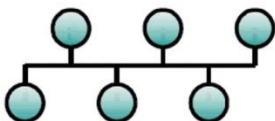
RS485 nije protokol, nego se na njemu implementiraju protokoli.

#### Karakteristike:

- 1) definiše samo **električne karakteristike** (tj. definiše način prenosa podataka, način korištenja medijuma – kablova, konektora i drajvera na hardverskom nivou koji su zaduženi za komunikaciju) ;
  - drajver predstavlja hardversku komponentu koja vrši konverziju signala koji generiše mikrokontroler u signal koji je definisan RS485 standardom ;
- 2) ima **balansiran (diferencijalni) interfejs** (razvlače se diferencijalni parovi, tj. 2 žice koje predstavljaju diferencijalne signale: 1 pozitivna i 1 negativna) ;
  - predajnik šalje pozitivan i negativan signal, prijemnik ih prima, od njih pravi diferencijalne signale, računa njihovu razliku i konvertuje ju u signal referenciran u odnosu na masu ;
- 3) **multipoint rad** (u RS485 mreži može komunicirati veći broj uređaja) ;
- 4) javlja se **signal zajedničkog moda** koji predstavlja signal smetnje između svake od linija u komunikaciji i mase (GND-a) ;
  - javlja se usljed interferencije okoline u kojoj se nalazi ;
  - postoje određene zaštite koje obezbjeđuju nesmetanu komunikaciju i uz ovaj signal smetnje ;
  - zaštitu je neophodno implementirati ukoliko naponski nivo signala zajedničkog moda izlazi iz opsega od -7 V do +12 V ;
- 5) na jednoj liniji može biti **maksimalno 32 uređaja** ;
- 6) **maksimalna brzina komunikacije** je do **10Mb/s** za dužinu kabla do 12m (za veće dužine kabla maksimalna brzina je manja) ;
- 7) **maksimalna dužina kabla** je **1200m** (dosta niska brzina komunikacije  $\approx 9600\text{b/s}$ ) ;

#### Konfiguracija:

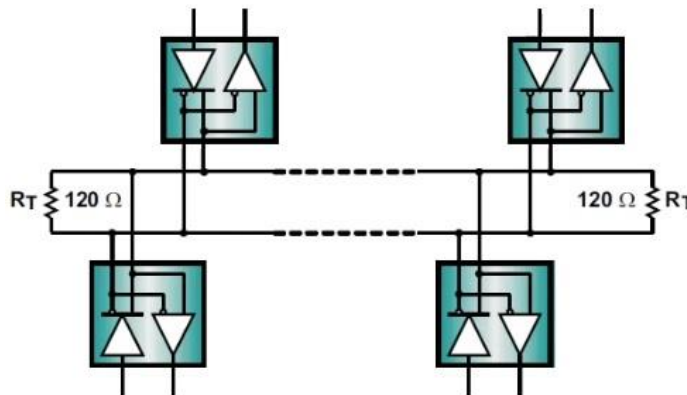
- RS485 predlaže **daisy chain** način mreže koji podrazumijeva postojanje glavnog komunikacionog voda na koji se povezuju svi uređaji putem odgovarajućih priključaka ;



- Unutar jednog drajvera postoji prijemni i predajni drajver, a pored te dvije linije postoje dvije linije koje definišu smjer drajvera: predajni ili prijemni smjer ;  
 - dva uređaja ne smiju slati podatke u isto vrijeme jer će doći do sudara podataka na magistrali ;  
 - RS485 je **half duplex** što znači da u jednom trenutku drajver može samo da prima ili samo da šalje podatke ;
- Ako je neophodno da se u isto vrijeme primaju i šalju podaci korsiti se RS485 **full duplex** ;
- Drajveri koji ne učestvuju u komunikaciji se isključuju tako što se postavljaju u stanje visoke impedanse ;

### Terminacija:

- Vrš se u cilju obezbjeđivanja **konstantne karakteristične impedanse** komunikacionog voda i **spriječavanja refleksije** signala ;
- Realizuje se postavljanjem terminacionih otpornika na krajevima komunikacionog voda ( $R_T = 120\Omega$ ). Ovaj način je poznatiji kao Tevenenov način ;
- Preporučuje se da se terminacioni otpornici uvijek implementiraju, a terminacija je neophodna kada je dužina komunikacionog voda veća od 3 talasne dužine brzine komunikacije ;
- Terminacija osigurava liniju sa tačno definisanim logičkim nulama i jedinicama ;

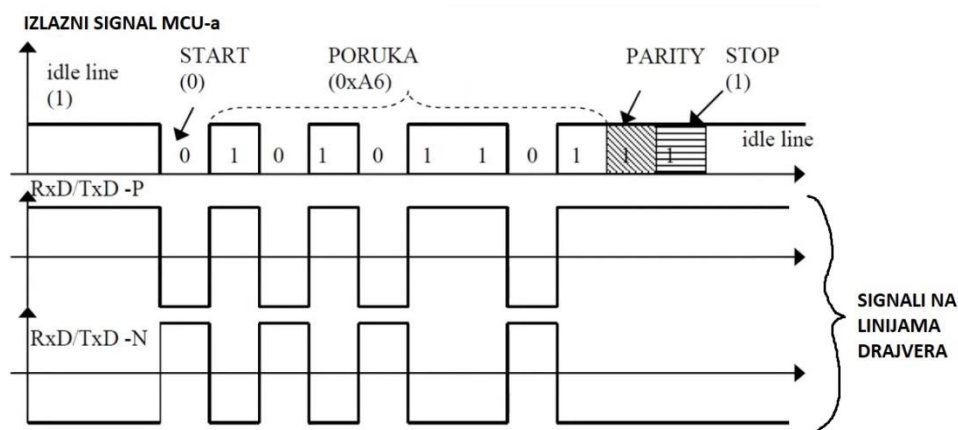


### Polarizacija:

- Polarizacija se koristi za drajvere koji ne učestvuju u komunikaciji i koji se nalaze u stanju visoke impedanse u cilju sprečavanja toga da se prijemna linija drajvera ponaša kao prijemna antena ;
- Realizuje se na taj način što se pozitivna linija komunikacionog voda postavlja na logičku jedinicu ( $V_{cc}$ ), a negativna na logičku nulu ( $GND$ ) ;

### 36. RS485 komunikacija – diferencijalni prenos podataka

Mikrokontroler šalje izlazni signal koji se dovodi na ulaz RS485 drajvera. Drajver ima 2 linije: P - pozitivnu i N - negativnu (+2 linije za smjer koje govore da li se ponaša kao prijemnik ili predajnik).



Početak UART poruke kreće od logičke 1 (neaktivnog stanja). Za logičku jedinicu sa MCU-a na P liniji drajvera se javlja pozitivan napon, a na N liniji negativan. Prijemnik drajvera vrši razliku P-N i dobija pozitivan signal. Izračunati signal šalje drugom mikrokontroleru.

Start bit je logička 0. Start bit se na P liniji drajvera pojavljuje kao negativan napon, a na N liniji kao pozitivan napon. Diferencijalni prijemnik računa P-N i dobija negativan signal što predstavlja logičku 0 kada je signal referenciran u odnosu na GND.

Drajver za RS485 ima dvostruku ulogu:

- generiše diferencijalni signal od signala referenciranog u odnosu na masu (od signala sa MCU-a),
- vrši razliku i od diferencijalnog signala generiše signal referenciran u odnosu na masu koji je pogodan za ulaz drugog MCU-a.

Kod diferencijalne komunikacije javlja se **signal zajedničkog moda** koji predstavlja signal smetnje. On se javlja i na pozitivnoj i na negativnoj liniji drajvera. Prilikom računanja razlike signala P-N, dolazi do brisanja i signala smetnje ukoliko je on u opsegu od -7 V do +12 V. To znači da je diferencijalna komunikacija imuna na smetnje za naponske nivoe u tom opsegu i da posebne zaštite u tom slučaju nisu potrebne.

RS485 definiše upotrebu kablova i konektora koji su imuni na elektro-statičke i elektro-magnetne smetnje u okruženju (koriste se kablovi sa *uvrnutim paricama*, tj. sa uvijenim žicama unutar kabla).

### **37. Izolovani RS485**

Između 2 uređaja koji imaju različite izvore napajanja javlja se tzv. “**ground loop**”, tj. petlja između uzemljenja ta dva uređaja.

Pojavom ground loop-a može se desiti da protekne struja od pozitivnijeg uzemljenja, ka drugom (negativnijem). Struja koja krene da teče može oštetiti drajver RS485. Zbog ove pojave neophodno je izvršiti **galvansku izolaciju** uređaja.

Galvanska izolacija se realizuje **izolovanjem UART linije** između drajvera i mikrokontrolera i **izolovanjem izvora napajanja** mikrokontrolera.

Izolovani komunikacioni moduli su dosta skuplji od običnih.

### **~~38. PROFIBUS – karakteristike i primer korišćenja~~**

### **~~39. PROFIBUS – OSI model~~**

### **~~40. PROFIBUS – fizički nivo PA~~**