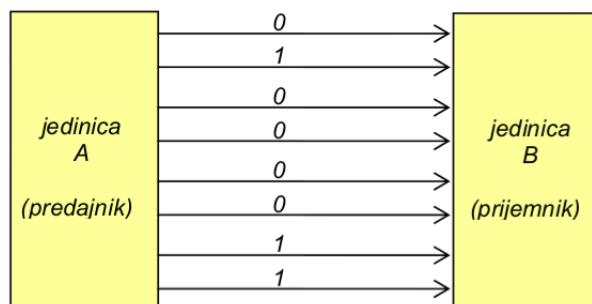


# Serijski komunikacioni protokoli

U savremenim mikroračunarskim sistemima sve češće postoji potreba povezivanja više uređaja u cilju komunikacije i razmenjivanja informacija. Postoji više načina povezivanja uređaja i prenosa podataka, a jedan od najčešće korišćenih je serijski prenos podataka.

## 1 Serijska komunikacija

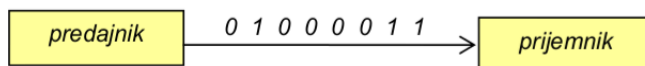
U dosadašnjim razmatranjima upoznali smo se sa paralelnim prenosom podataka kod koga se između dve jedinice istovremeno prenosi dva ili više bita. Na sledećoj slici ilustrovan je paralelni prenos bajtova (8 bita) od jedinice A (predajnik) do jedinice B (prijemnik). Kod paralelnog prenosa postoji poseban provodnik za prenos svakog od ukupno 8 bita u jednom bajtu. Na primer, bajt 43<sub>16</sub> (odnosno 0100 0011 u binarnom brojnem sistemu) od jedinice A do jedinice B prenosi se tako što za svaki bit postoji poseban provodnik tako da svi biti od jedinice A do jedinice B stižu istovremeno, videti primer na slici 1.



Slika 1: Paralelni prenos podataka

Paralelni prenos podataka pogodan je za velike brzine prenosa i relativno kratka rastojanja između jedinica. Međutim, već kod rastojanja veća od nekoliko metara, cena provodnika počinje da raste i paralelni prenos nije praktičan. Cilj serijskog prenosa je da se smanji cenu provodnika tako što se prenosi samo jedan bit u jednom trenutku, odnosno koristi samo jedan provodnik za prenos podataka, slika 2.

Pošto je u jednom trenutku moguće preneti samo jedan bit, očigledan nedostatak serijskog prenosa je u tome što je sporiji u odnosu na paralelni prenos. Intuitivno, paralelni prenos sa  $n$



Slika 2: Serijski prenos podataka

provodnika brži je  $n$  puta u odnosu na serijski prenos, naravno pod pretpostavkom da su svi ostali parametri prenosa isti.

Kod serijskog prenosa podataka na predajnoj strani mora da se obavi paralelno–serijska, a na prijemnoj strani serijsko–paralelna konverzija. Na slici 3 levo prikazan je princip paralelno-serijske, a desno serijsko-paralelne konverzije. Obe vrste konverzije zasnovane su na primeni pomeračkog registra sa serijskim i paralelnim ulazima i izlazima, o kojima je bilo reči ranije.

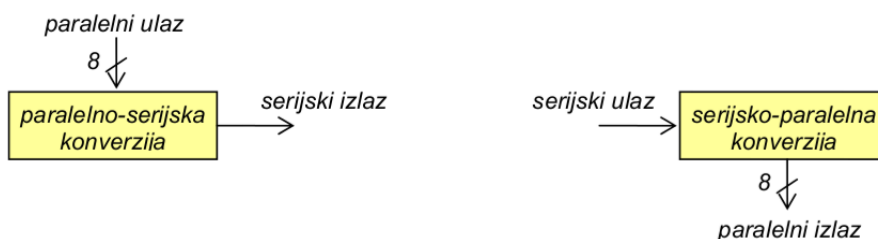
Pomerački registar sa paralelnim ulazom i serijskim izlazom koristi se na predajnoj strani za konverziju paralelnih podataka u serijski niz bita. Na primer, ako se bajt 0x43 (0100 0011 u binarnom brojnem sistemu) paralelno upiše u registar, onda će se u ritmu sinhronizacionog signala biti pomerati u desnu stranu tako da će se serijskom izlazu prvo pojaviti najmanje značajan bit (u navedenom primeru logička 1), zatim sledeći bit (u primeru logička 1), i tako dalje. Na kraju će se preneti najznačajniji bit (u primeru je to logička 0).

Pomerački registar sa serijskim paralelnim ulazom i paralelnim izlazom koristi se na prijemnoj strani za suprotnu konverziju, odnosno konverziju ulaznih bita koji se serijski primaju u paralelni podatak. U prethodnom primeru, ako pomerački registar prvo dobije logičku 1, zatim logičku 1, pa logičku 0, i tako dalje, i na kraju logičku 0, onda će se ovi biti pomerati za po jedno mesto dok se u registru ne formira paralelni podatak 0x43 koji je dostupan u paralelnom obliku na paralelnom izlazu pomeračkog registra.

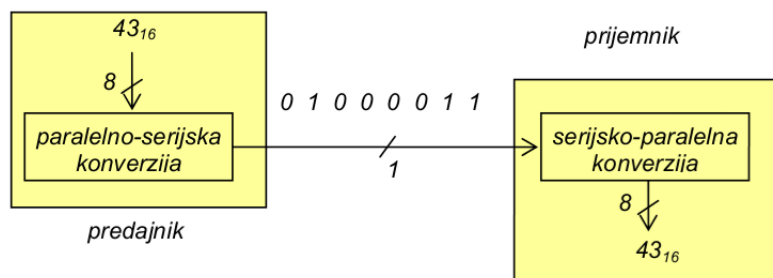
Koncept serijskog prenosa prikazan je na slici 4. U predajniku se paralelni podatak upisuje u paralelno-serijski registar, zatim se biti pomeraju udesno i jedan po jedan preko serijske veze prenose do prijemnika. Na prijemnoj strani serijski primljeni biti dolaze na serijski ulaz serijsko-paralelnog registra, pomeraju udesno i asemblira originalni podatak koji je u paralelnom obliku dostupan na paralelnom izlazu registra.

Kod serijskog prenosa ima nekoliko pitanja od čijih odgovora zavisi tip veze i način serijskog prenosa:

- Da li je serijska veza jednosmerna ili dvosmerna i ako je veza dvosmerna, da li može da se istovremeno obavlja predaja i prijem?
- Kako se vrši sinhronizacija rada predajnika i prijemnika?



Slika 3: Serijsko-paralelna i paralelno-serijska konverzija



Slika 4: Koncept serijskog prenosa

- Da li postoji i kako se obavlja kontrola ispravnosti prenosa podataka?
- Da li se podaci prenose reč po reč ili u blokovima?

U pogledu smera veze, razlikujemo jednosmernu (simpleks), polu-dvosmernu (polu-dupleks) i dvosmernu (dupleks) vezu. Kod jednosmerne veze između predajnika i prijemnika postoje dva provodnika, jedan za prenos serijskih podataka (na slici dole obezbežen sa TXD) i drugi za masu (na slici obezbežen sa GND) videti sliku 5.

Kod dvosmerne veze postoje tri provodnika, po jedan za vezu u oba smera i jedan zajednički provodnik za masu. Polu-dvosmernu vezu ima dva provodnika, jedan za masu i drugi koji, po dogovoru između predajnika i prijemnika, u jednom periodu vremena može da se koristi za serijski prenos u jednom, a u drugom periodu vremena za serijski prenos u drugom smeru.

Sinhronizacija rada između predajnika i prijemnika je važna jer je osnova za ispravan serijski prenos podataka. Sinhronizacija pre svega podrazumeva da predajnik i prijemnik rade na istoj brzini prenosa, odnosno da je brzina prenosa izražena u broju bita u sekundi ista na predajnoj i prijemnoj strani.

Sinhronizacija rada može da se vrši i razmenom upravljačkih signala, odnosno rukovanjem (engleski *handshaking*) između predajnika i prijemnika. Naravno, rukovanje zahteva dodatne provodnike za prenos upravljačkih signala između predajnika i prijemnika.

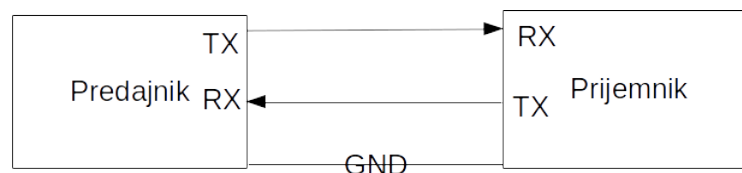
Kontrola ispravnosti prenosa podataka može da se obavlja na različitim nivoima. Na primer, tačnost prenosa podataka može da se obavi dodavanjem bita parnosti na predaji i proverom bita parnosti na prijemu.

Podaci mogu da se prenose reč po reč ili u blokovima u kojima se nalazi više reči. Prenos reč po reč obično se naziva asinhroni prenos, kod koga se za svaku reč prvo šalje početni (start) a na kraju reči završni (stop) bit. Kod sinhronog prenosa obično se prenosi blok podataka u kome postoji početna i završna reč, a pored serijskih podataka prenosi se i sinhronizacioni signal od predajnika do prijemnika.

U nastavku ćemo više pažnje posvetiti najčešće korišćenim serijskim komunikacionim protokolima: Asinhrona serijska komunikacija, SPI i IIC.



Slika 5: Jednosmerna veza



Slika 6: Povezivanje UART predajnika i prijemnika kod Asinhronne serijske komunikacije

## 2 Asinhrona serijska komunikacija

Asinhrona serijska komunikacije podrazumeva komunikaciju između dva UART kontrolera (engleski Universal Asynchronous Receiver Transmitter), uređaja koji podržavaju prijem i predaju serijskih podataka u asinhronom načinu rada.

RX linija se koristi za prijem podataka, dok se podaci šalju putem TX linije (slika 6).

Serijski kontroler koji podržava i sinhroni način rada na Engleskom se naziva Universal Synchronous/Asynchronous Receiver Transmitter ili USART ali o njemu ovde neće biti reči.

Kod predaje, UART kontroler prihvata podatke u paralelnom obliku od mikroprocesora, pretvara ih u serijski oblik, ubacuje dodatne bite i u serijskom obliku šalje poruku na izlaznu liniju.

Signali koje koriste UART kontroleri obično su kompatibilni sa TTL standardom, pa tako napon na izlazu koji je između 0 V i 0.4 V predstavlja logičku 0, a napon između 2.6 V i 5 V predstavlja logičku 1.

Sličan postupak je kod prijema podataka, ali ovoga puta u suprotnom smeru. Od strane UART kontrolera podatak se iz serijskog prevodi u paralelni oblik i u tom obliku predaje mikroprocesoru.

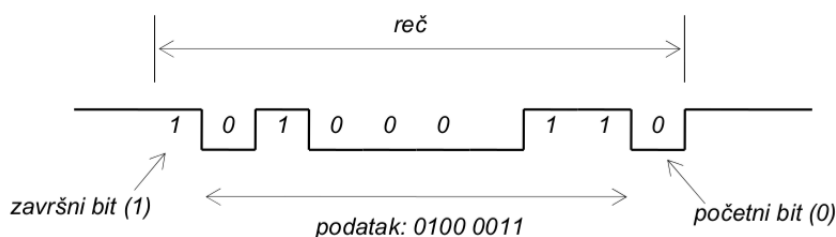
Kod asinhronog prenosa podaci se prenose reč po reč (engleski character).

Kada je linija neaktivna, na njoj je visok naponski nivo, tj pre početka serijskog prenosa i između prenosa dve uzastopne reči, provodnik za serijski prenos drži se u stanju logičke 1. Opadajuća ivica na komunikacionoj liniji je signal prijemnoj strani da predajna strana počinje sa slanjem reči.

Reč koja se šalje, sastoji se od:

1. početnog bita (engleski *start*) koji je logička 0
2. 7 ili 8 *bita podataka* (ovo je zapravo informacija koja se prenosi)
3. Opcionog bita za proveru parnosti. Bit parnosti predstavlja najjednostavniju kontrolu ispravnosti primljene reči i postavlja se na "1" ukoliko je broj jedinica u okviru bita podataka (stavka 2) paran (ili neparan, što se može konfigurisati). Na prijemnoj strani se tada proverava koliko ima jedinica medju bitima podataka i proverava se da li se to slaže sa bitom parnosti koji je primljen. Očigledno je da se na ovaj način može detektovati greška na jednom bitu, ali na ovaj način nije moguće detektovati koji bit je pogrešno primljen, niti prepoznati grešku u slučaju pogrešnog prijema parnog broja bita
4. 1 do 2 završna (stop) bita koji su uvek logičke "1".

U slučaju prenosa binarnog podatka 0100 0011 od 8 bita, ako se ne koristi bit za proveru parnosti i ukoliko postoji 1 završni bit, onda bi se preko provodnika za serijski prenos preneo niz bita prikazan na slici 7. Treba obratiti pažnju da se u prikazu formata reči sa slike podrazumeva da se sa desne



**Slika 7:** Format reči kod asinhronog prenosa

strane nalaze biti koji se prvi šalju tokom prenosa. U skladu sa tim, krajnje desno na slici je prikazan početni (START) bit, dok se završni (STOP) bit nalazi sa krajnje leve strane.

Da bi prijemnik i predajnik uspešno komunicirali, potrebno je konfigurisati i prijemnik i predajnik na identičan način. Ova konfiguracija minimalno podrazumeva konfigurisanje:

1. Brzine serijske komunikacije (eng. *baud rate*) izražen u broju bita u sekundi. Tipične brzine asinhronne serijske komunikacije su 9600b/s, 19200b/s, 38400b/s, 57600b/s i 115200b/s
2. Broj bita podataka 7 ili 8
3. Da li se koristi bit parnosti i ako se koristi da li će on biti setovan za paran ili neparan broj jedinica u okviru bita podataka
4. Broj stop bita koji se koristi (1 ili 2)

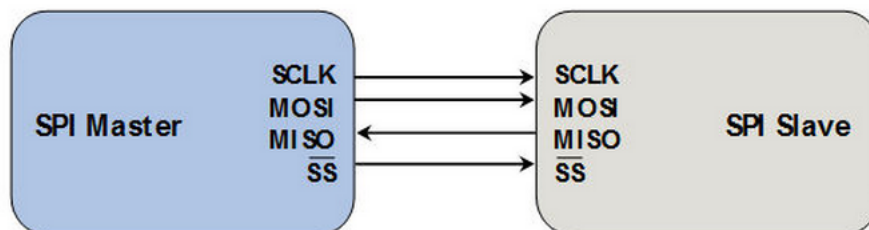
Nakon što su prijemnik i predajnik konfigurisani, prijemnik poruke prolazi kroz sledeća stanja čekajući na poruku:

1. Čeka početak poruke pojavom *start* bita (opadajuća ivica na prijemnoj RX liniji)
2. Nakon detekcije početka poruke, prima jedan po jedan bit poruke tako što očitava napon na RX liniji svakih  $kT + T/2$ ,  $k=0,1,\dots,8$ ,  $T=1/\text{baud rate}$ . Da bi ovo bilo moguće, neophodno je da i prijemnik i predajnik “znaju” brzinu komunikacije, jer nema nikakve sinhronizacije (prijemnik nema nikakvu indikaciju kada je na liniji dostupan “novi” bit)
3. Nakon prijema cele poruke, proverava da li bit parnosti u skladu sa bitovima podataka (u slučaju da je konfigurisano korišćenje bita parnosti). Ukoliko jeste, podataka je prihvaćen, u suprotnom primljeni podatak se odbacuje i neophodno ga je ponovo primiti.

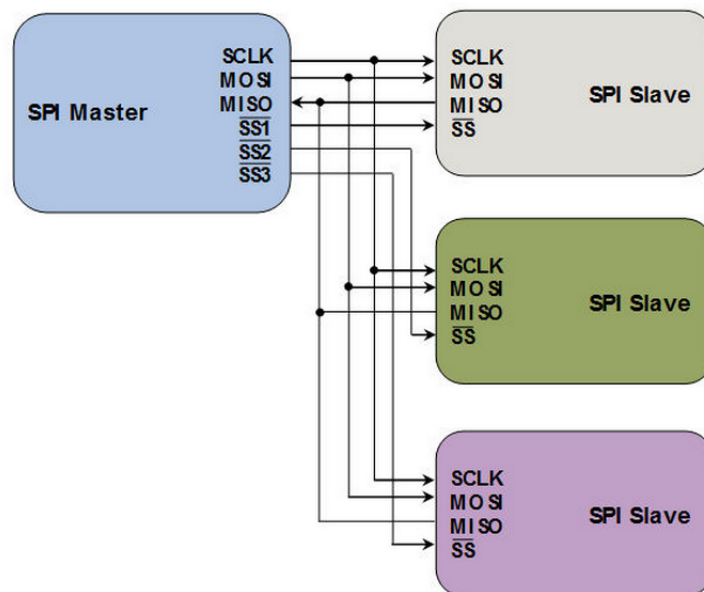
Kao što je već naglašeno ranije, u paraleli sa prijemom poruke, moguće je istovremeno slati poruku korišćenjem nezavisne komunikacione linije TX.

### 3 SPI komunikacioni protokol

Naziv SPI je skraćenica od Serial Peripheral Interface i namenjen je komunikaciji između integrisanih kola, kao i za relativno sporu komunikaciju sa perifernim jedinicama koje se nalaze na istoj štampanoj ploči. SPI protokol je osmislila i patentirala Motorola. Inicijalno SPI je



Slika 8: Jedan master - jedan slave

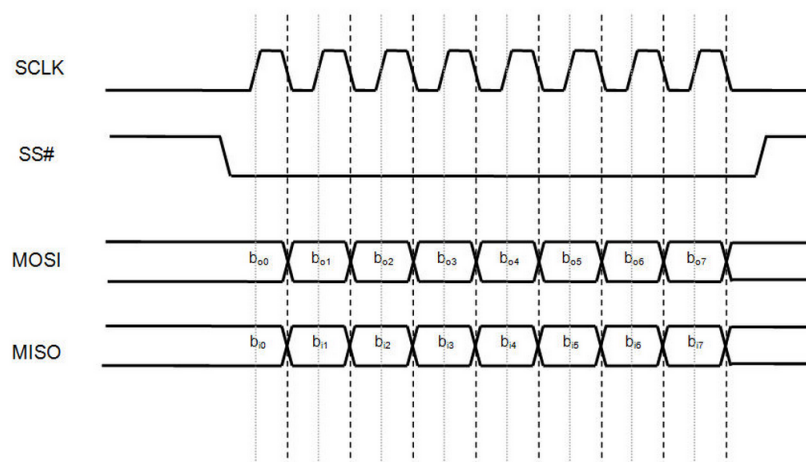


Slika 9: Jedan master - više slave uređaja

osmišljen kako bi povezoao mikrokontroler sa periferijama koristeći 4 žice i prilično je jednostavan. SPI komunikacija podrazumeva da uvek u sistemu postoji jedinstveni *master*, dok je moguće postojanje većeg broja *slave* uređaja:

1. SCLK signal je taktni signal koji uvek generiše master i vodi se ka svim slave uređajima. Svi SPI signali su sinhronizovani u odnosu na ovaj signal;
2. SS<sub>n</sub> signal predstavlja odabir slave-a (engl. *Slave Select*) služi da odabere jedan od povezanih slave uređaja u datom trenutku, sa kojim će master da komunicira;
3. MOSI (Master Out Slave In) je linija podataka od mastera ka svim slave uređajima;
4. MISO (Master In Slave Out) je linija podataka od slave uređaja ka master uređaju.

Na slici 8 je prikazano povezivanje jednog master uređaja sa jednim slave uređajem, dok je na slici 9 prikazan način povezivanja jednog master-a sa više slave-ova.



Slika 10: Signali na SPI magistrali

Kao što je već naglašeno, SPI je komunikacioni protokol koji omogućava postojanje samo jednog *master* uređaja u sistemu. Ovo znači da jedan centralni uređaj inicira svaku komunikaciju sa ostalim učesnicima u komunikaciji. Kada SPI master želi da šalje podatak slave-u ili da prihvati neki podatak od slave-a, on će ga odabrati tako što odgovarajuću SS liniju obori na nizak naponski nivo i aktivira taktni signal. Master generiše podatke za slanje na MOSI liniji, dok čita bit po bit koji se pojavljuje na MISO liniji (slika 10).

Sa slike se može videti da se, u ovoj konfiguraciji, na silaznoj ivici SCLK signala ažuriraju informacija na MISO i MOSI linijama (i master i slave postavljaju “nove” bite), dok se na rastućoj ivici učitavaju vrednosti na komunikacionim linijama. Postoje četiri moda (MODE 0, 1, 2 i 3) koji definišu na kojoj ivici SCLK signala se menja stanje na komunikacionim linijama, na kojoj ivici se očitavaju trenutne vrednosti signala na komunikacionim linijama, i u kom stanju se nalazi SCLK linija kada je u “neaktivnom” stanju (nizak ili visok naponski nivo).

Master/slave par mora da koristi isti skup konfiguracionih parametara kako bi komunikacija bila uspešna: SCLK frekvencija, podešavanja za promenu stanja na linijama, kao i očitavanja stanja sa linija. Ukoliko se koristi više slave uređaja, master mora da se rekonfiguriše pre svake komunikacije sa različitim slave uređajem.

Za kraj, nekoliko karakteristika SPI komunikacionog interfejsa:

1. SPI protokol ne definiše maksimalnu brzinu komunikacije (za razliku od asinhronne serijske komunikacije, kod SPI je sve sinhronizovano sa SCLK signalom)
2. SPI ne omogućava nikakvu potvrdu prijema podatka, kao ni kontrolu toka komunikacije
3. SPI master čak nije svestan ni da li je slave uređaj uopšte prisutan na SPI magistrali

## 4 I<sup>2</sup>C komunikacioni protokol

I<sup>2</sup>C je skraćenica od *Inter Integrated Circuit* (ili IIC) i razvijen je od strane kompanije Philips 1972. godine. Osnovna namena ovog protokola jeste bila sprega mikroprocesora sa integrisanim kolima u okviru TV uređaja. Do tada je tipičan način sprege ovog tipa podrazumevao spajanje integrisanog

kola na adresnu magistralu i magistrali podataka mikroprocesora. Problem kod ovakvog pristupa je u rutiranju i usložnjava se sa povećanjem broja integrisanih kola koja na ovaj način moraju biti povezana sa mikroprocesorom. U cilju smanjenja troškova, Philips je osmislio komunikacioni protokol koji zahteva postojanje samo dve žice. Maksimalna brzina komunikacije u početku je bila 100kbps (kilo bits per second), nakon promene specifikacije 1995. godine, podignuta je na 400kbps, a od 1998. maksimalna brzina je 3.4Mbps (mega bits per second).

I2C je multi-master protokol koji koristi dve signalne linije:

1. SDA (Serial DAta) za podatke
2. SCL (Serial CLock) za takt

Ne postoji potreba za dodatnim signalom koji bi omogućio CS funkcionalnost kao u slučaju SPI komunikacije. Teorijski, proizvoljan broj master uređaja i proizvoljan broj slave uređaja mogu da se nađu na I2C magistrali. Ovo je omogućeno protokolom koji definiše:

- 7-bitne adrese slave uređaja pri čemu svaki uređaj povezan na I2C magistralu mora imati svoju jedinstvenu adresu;
- podatke podeljene u bajtove (8 bita informacije);
- nekolicinu kontrolnih bita koji određuju početak i kraj komunikacije, kao i smer toka podataka i potvrdu prijema.

Na početku komunikacije master uvek generiše START uslov (kasnije ćemo reći koji je START uslov). Ovo daje signal svim uređajima koji se nalaze na I2C magistrali da krenu da oslušuju poruke.

Nakon START uslova, master generiše adresu. Svi uređaji proveravaju adresu i porede sa svojom adresom. Samo uređaj koji ima datu adresu nastavlja da komunicira sa master uređajem, dok ostali ignorišu ostatak poruke, sve dok ne stigne STOP uslov na magistrali. Uređaj koji ima traženu adresu mora da generiše ACK (acknowledge signal) kako bi master znao da je uređaj sa traženom adresom pronađen. Nakon prijema ACK signala master nastavlja sa slanjem ili čitanjem slave uređaja i kada završi sa tim generiše STOP signal.

U slučaju da master želi da šalje podatke slave uređaju nakon 7 bita adrese generiše se logička nula, dok se logička jedinica generiše u slučaju kada je potrebno čitati podatke poslate od strane slave-a.

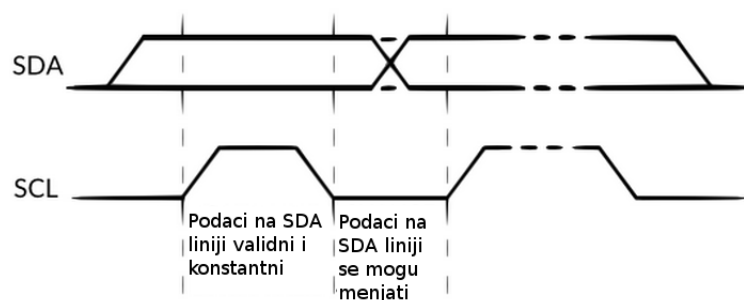
Nakon svakog primljenog (poslatog) bajta master (slave) generiše ACK signal (logička nula na SDA liniji) nakon čega se nastavlja sa sledećim bajtom (slanje ili prijem). Prilikom prijema višebajtnog podatka od strane slave-a, master generiše NACK (logičko 1 na SDA liniji) nakon prijema poslednjeg bajta i time daje do znanju slave-u da je transfer završen.

U sledećoj tabeli je format poruka koje se razmenjuju između mastera i slave-a, u slučaju slanja ili prijema 2 bajta podataka.

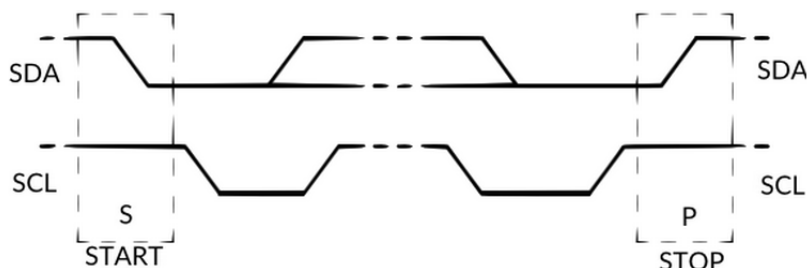
| START | Adresa slave-a | $R/\overline{W}$ | ACK   | Podaci | ACK   | Podaci | ACK/NACK | STOP  |
|-------|----------------|------------------|-------|--------|-------|--------|----------|-------|
| 1 bit | 7 bita         | 1 bit            | 1 bit | 8 bita | 1 bit | 8 bita | 1 bit    | 1 bit |

U slučaju slanja ka slave uređaju, podaci se generišu u skladu sa sledećom tabelom. Pri tome treba imati u vidu da su signali koje generiše master posivljeni, dok su u ćelijama sa belom pozadinom prikazani signali koje generiše slave.





Slika 11: Promena stanja na liniji SDA (osim u slučaju START i STOP karaktera)



Slika 12: START i STOP signali

|       |                |       |       |        |       |        |       |       |
|-------|----------------|-------|-------|--------|-------|--------|-------|-------|
| START | Adresa slave-a | 0     | 0     | Podaci | 0     | Podaci | 0     | STOP  |
| 1 bit | 7 bita         | 1 bit | 1 bit | 8 bita | 1 bit | 8 bita | 1 bit | 1 bit |

U slučaju prijema, situacija je malo drugačija (obratiti pažnju na NACK signalizaciju od strane master-a nakon poslednjeg primljenog bajta):

|       |                |       |       |        |       |        |       |       |
|-------|----------------|-------|-------|--------|-------|--------|-------|-------|
| START | Adresa slave-a | 1     | 0     | Podaci | 0     | Podaci | 1     | STOP  |
| 1 bit | 7 bita         | 1 bit | 1 bit | 8 bita | 1 bit | 8 bita | 1 bit | 1 bit |

START i STOP signali se generišu kao specijalna kombinacija naponskih nivoa na SDA i SCL liniji. U slučaju postavljanja podataka na SDA liniju (osim START i STOP karaktera) koriste se pravila prikazana na slici 11. Podaci su validni i mogu se smatrati nepromenljivim samo u intervalu vremena kada je stanje naponskog signala na SCL liniji stabilno i kada je to visok naponski nivo. Sve promene naponskog nivoa na SDA liniji se moraju izvršiti u periodima kada je naponski nivo na SCL liniji nizak.

U skladu sa tim, START karakter podrazumeva opadajuću ivicu SDA signala u trenutku kada je napon na SCL liniji još uvek visok. Slično, STOP karakter podrazumeva rastuću ivicu SDA signala u trenutku kada je napon na SCL liniji još uvek visok (slika 12).