

The background is a dark green field filled with intricate, glowing yellow-green circuit traces that resemble a printed circuit board. These traces are interconnected by numerous small yellow circular dots, some of which are arranged in vertical columns on the left and right sides of the image. The overall effect is a high-tech, digital aesthetic.

Serijska komunikacija

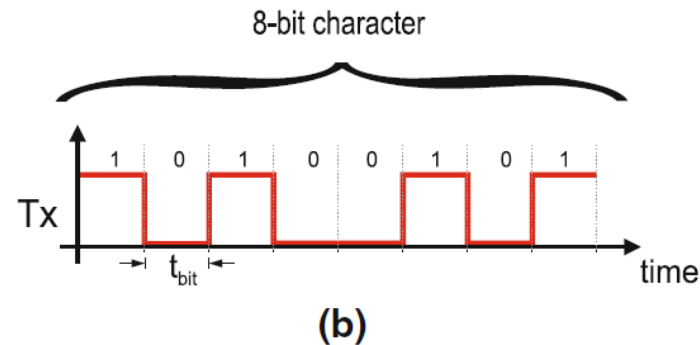
Katedra za elektroniku

Osnove serijskih komunikacionih sistema

- **Serijski komunikacioni kanali** su danas bez sumnje najzastupljeniji vid komunikacije koja se koristi unutar digitalnih sistema
- Različiti oblici serijskih komunikacionih formata i protokola koriste se u aplikacijama počevši od **kratkih komunikacionih linkova** koji povezuju module unutar istog ili odvojenih integrisanih kola, pa sve do komunikacionih linkova koji obezbeđuju vezu sa svemirskim sondama koje putuju na udaljene planete
- Praktično **svi oblici komunikacije** koji se danas koriste u potrošačkoj elektronici baziraju na **serijskoj komunikaciji**
- Neki od najpoznatijih serijskih komunikacionih protokola korišćenih danas su:
 - RS-232, USB (Universal Serial Bus), Bluetooth, FireWire
 - Ethernet, WiFi
 - I²C, SPI
 - PCIe
 - SATA

Osnove serijskih komunikacionih sistema

- **Paralelni komunikacioni kanal**, prilikom prenosa n -bitnog podatka, koristi n **simultanih signalnih linija**, jednu za **svaki bit** podatka koji se prenosi
- Serijski komunikacioni kanal, prilikom prenosa n -bitnog podatka, koristi samo **JEDNU** signalnu liniju, pri čemu se n bita podatka prenosi **sekvencijalno**, jedan za drugim
- Svaki bit koji se prenosi serijski zahteva unapred definisanu količinu vremena, t_{bit} , tako da je za prenos n -bitnog podatka potrebno ukupno $n \cdot t_{bit}$ sekundi
- Na slici desno prikazan je prenos jednog 8-bitnog podatka (0xA5) preko paralelnog, a), i serijskog, b), kanala



Osnove serijskih komunikacionih sistema

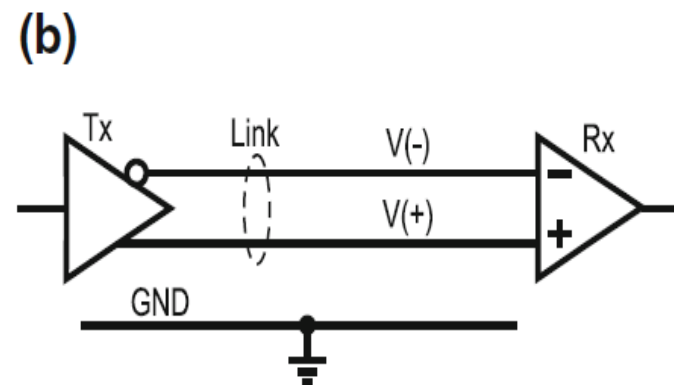
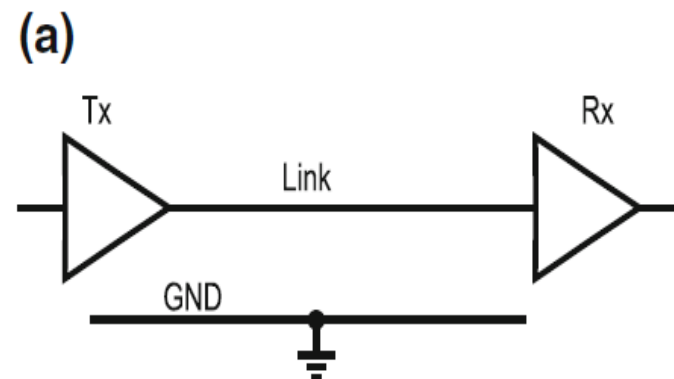
- **Brzina prenosa podataka** preko serijskog kanala određena je količinom vremena potrebnog za prenos **jednog bita** informacije (t_{bit})
- U praksi se koriste dve metrike za iskazivanje brzine prenosa podataka preko serijskih kanala:
 - Bitska brzina (**bit rate**)
 - Simbolska brzina (**baud rate**)
- **Bitska brzina** (**bps**) pokazuje koliko je bita moguće preneti tokom jedne sekunde preko serijskog kanala
- Na primer, serijski kanal sa bitskim vremenom $t_{bit} = 10 \text{ ns}$ ima bitsku brzinu od 100 Mbit-a u sekundi, ili 100 Mbps, 100 Mbit/s

Osnove serijskih komunikacionih sistema

- **Simbolska brzina (baud rate)**, odnosi se na broj karaktera (**simbola**) koji se mogu preneti preko serijskog kanala tokom jedne sekunde
- Ukoliko je svaki simbol predstavljen pomoću jednog bita, simbolska i bitska brzina su jednake
- Ukoliko se koriste složenije modulacione tehnike, moguće je kodirati veći broj bita za svaku promenu signala na serijskom kanalu
- Na primer, ukoliko se koristi **fazna modulacija** (*Phase-Shift Keying, PSK*), moguće je kodirati dva ili više bita u svaku promenu faze signala koji putuje preko serijskog komunikacionog linka
- U ovom slučaju bitska brzina će biti veća od simbolske brzine

Osnove serijskih komunikacionih sistema

- Komunikacioni kanali mogu biti realizovani na razne načine
- **Ožičeni kanali** mogu da koriste (single-ended) ili diferencijalne veze (žice) za prenos podataka
- U slučaju **single-ended veze**, kanal se sastoji iz jednog fizičkog voda (žice) i referente linije (ground)
- U slučaju **n-bitnog paralelnog single-ended** kanala bilo bi potrebno ukupno $n+1$ žica za ostvarivanje veze: po jedna žica za svaki bit plus referentna linija mase
- U slučaju **n-bitnog serijskog single-ended** kanala dovoljne bi bile samo dve žice: jedna signalna i jedna referentna



Osnove serijskih komunikacionih sistema

- U slučaju diferencijalne veze, informacija koja se prenosi kodirana je kao naponska razlika između dve fizičke žice
- Ova vrsta veze je otpornija na uticaj smetnji, jer se uticaj aditivnih smetnji međusobno poništava prilikom prenosa podataka preko diferencijalne veze
- **Paralelni diferencijalni n -bitni** kanal zahtevao bi $2n+1$ žica za ostvarivanje veze: po dve žice za svaki bit, plus jedna žica za prenos referentnog napona (mase)
- U slučaju **serijskog diferencijalnog kanala** bile bi potrebne samo 3 žice: dve za prenos informacija i jedna za prenos referentnog napona

Osnove serijskih komunikacionih sistema

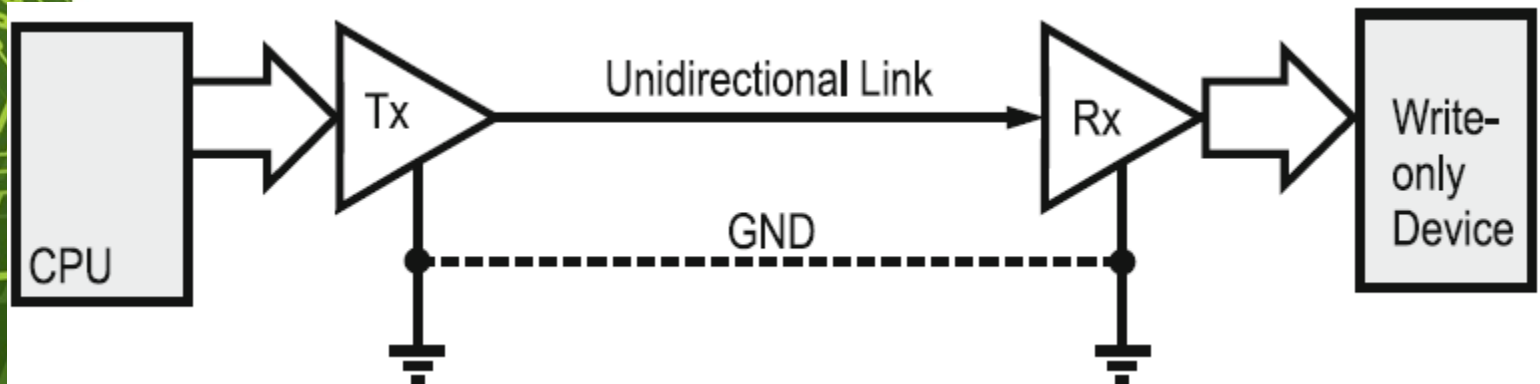
- Bežični kanali koriste bežične načine komunikacije za prenos podataka bazirane na različitim fizičkim fenomenima:
 - **Optičke**, u slučaju infracrvenih ili laserskih primopredajnika
 - **Akustičke**, u slučaju podvodnih komunikacija
 - **Elektromagnetne**, u slučaju radio komunikacija kao što je WiFi
- U bilo kom od ovih slučajeva nema potrebe za zajedničkom referencom, ali bi u slučaju paralelnog prenosa i dalje bilo potrebno **n bežičnih linkova**, dok bi u slučaju serijskog prenosa bio dovoljan samo jedan ovakav link.
- Na osnovu prethodne analize može se naslutiti glavna prednost serijskih kanala u odnosu na paralelne: **njihova cena**
- Mnogo je jeftinije i jednostavnije imati samo **jedan serijski link** za prenos podataka nego **n paralelnih**, sinhronizovanih linkova u slučaju paralelnog prenosa

Osnove serijskih komunikacionih sistema

- Ukoliko pretpostavimo **jednaka bitska vremena** unutar serijskog i paralelnog kanala, **paralelna komunikacija bila bi n puta brža** od serijske
- Ovo je uvek tačno, međutim kako rastojanje na kojem se vrši komunikacija i brzina kojom se prenose podaci rastu, talasni efekti na komunikacionim vodovima, preslušavanje i uticaj šuma počinju da uzimaju sve veći danak u slučaju paralelnih komunikacionih kanala, ograničavajući njihovu praktičnu primenu
- Ovo je bio jedan od najvažnijih razloga za **masovno korišćenje serijskih komunikacionih kanala** u savremenim elektronskim uređajima i sistemima

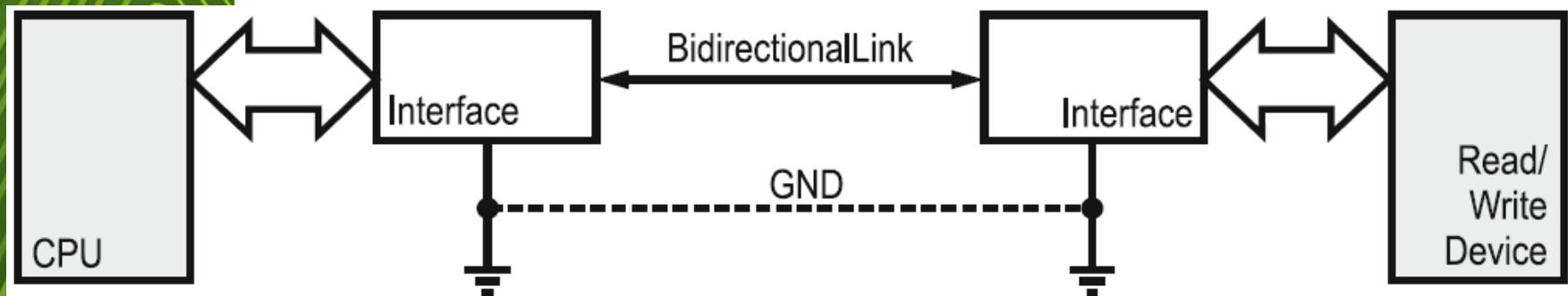
Vrste serijskih komunikacionih kanala

- Na osnovu mogućnosti prenosa podataka između predajnika i prijemnika, serijski kanali dele se u sledeće grupe:
 - Unidirekcioni (simplex)
 - Half-dupleks
 - Bidirekcioni (full-duplex)
- Simplex serijski kanal prenosi podatke samo u jednom smeru koristeći odgovarajući komunikacioni link. Na jednom kraju nalazi se predajnik, dok se na drugom kraju nalazi prijemnik.
- Simplex kanali nemaju način za potvrdu i verifikaciju ispravnog prijema podataka od strane prijemnika
- Primeri simplex kanala su radio i TV predajni sistemi, izlazne periferije poput štampača ili displeja, ulazne periferije poput tastera i tastatura



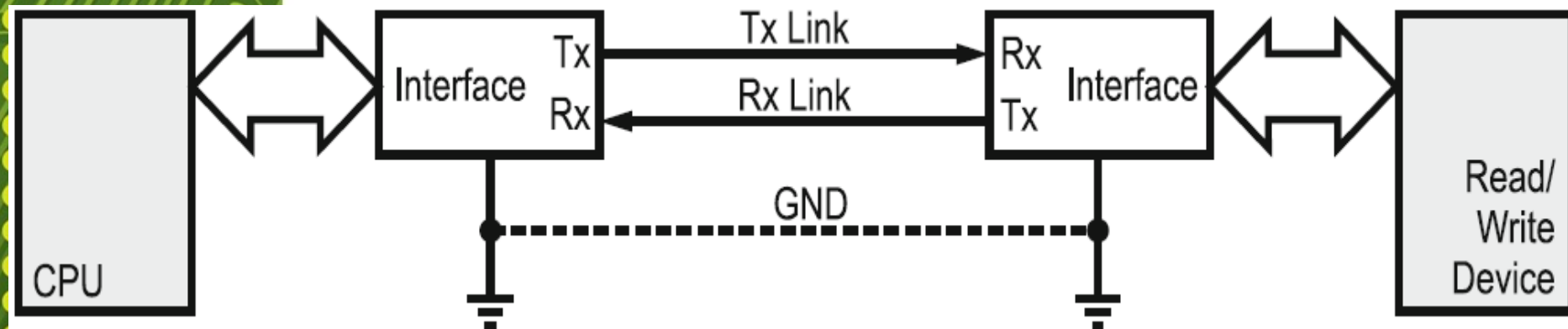
Vrste serijskih komunikacionih kanala

- **Half-duplex serijski kanali** koriste **jedan komunikacioni link**, ali se preko njega podaci mogu prenositi u **oba smera** pri čemu se u svakom trenutku podaci mogu prenositi samo u jednom smeru
- Kod half-duplex sistema na oba kraja nalaze se **serijski primopredajnici** (serial transceiver), koji imaju mogućnost slanja i prijema podataka
- Svaki put kada je potrebno promeniti smer prenosa, primopredajnici na oba kraja menjaju modove rada
- Ova promena zahteva postojanje odgovarajućih pravila kako bi se izbegla situacija da na oba kraja primopredajnici pokušavaju da pošalju podatke
- Ovaj skup pravila koji određuje ponašanje primopredajnika na oba kraja kanala i način na koji oni organizuju i interpretirau podatke koji putuju kanalom naziva se **komunikacioni protokol**



Vrste serijskih komunikacionih kanala

- **Full-duplex serijski kanali** koriste dva posebna komunikaciona linka, jedan za prenos a drugi za prijem podataka, omogućavajući **istovremeni prenos podataka u oba smeru**.
- Uređaji na krajevima kanala imaju mogućnost istovremenog prijema i predaje podataka, ne zahtevajući nikakvu promenu u režimu rada, na taj način omogućavajući **neprekidni bidirekcionni prenos podataka**.
- **Većina savremenih serijskih komunikacionih kanala** koji se koriste u embeded sistemima koriste **full-duplex kanale**.



Vrste serijskih komunikacionih kanala

- **Komunikacioni linkovi** koji se koriste u simplex, half-duplex i full-duplex serijskim komunikacionim kanalima mogu biti **žični** ili **bežični**
- U slučaju žičnih linkova, obe strane kanala moraju koristiti isti referentni napon, te je stoga potrebna dodatna linija (žica) između dva kraja kanala
- U slučaju bežičnih kanala nema potrebe za ovom referentnom linijom
- Komunikacioni kanali prikazani na prethodnim slajdovima predstavljaju takozvane *point-to-point* topologije, kod kojih kroz **jedan kanal** komuniciraju tačno **dva uređaja**
- Postoje i složenije topologije koje dozvoljavaju postojanje **većeg broja uređaja** koji međusobno komuniciraju koristeći jedan, **zajednički kanal**. Ovakve topologije poznate su pod imenom **multi-point** ili **multi-drop** topologije.

Sinhronizacija unutar serijskih kanala

- Svi serijski komunikacioni kanali zahtevaju postojanje stabilnog izvora referentnog **sinhronizacionog signala (clock)** na osnovu kojega se mogu odrediti **brzine predaje i prijema podataka** i izvršiti unutrašnja sinhronizacija prijemnika i predajnika
- U zavisnosti od načina na koji se obezbeđuje klok signal, kanal može biti:
 - Asinhroni
 - Sinhroni
- **Asinhroni kanali** koriste **nezavisne generatore sinhronizacionog signala** na prijemnoj i predajnoj strani
- **Sinhroni kanala** distribuiraju jedan, **zajednički sinhronizacioni signal**, zajedno sa podacima koji se prenose. Ovaj zajednički sinhronizacioni signal generiše se na jednom mestu, najčešće na predajnoj strani.

Sinhronizacija unutar serijskih kanala

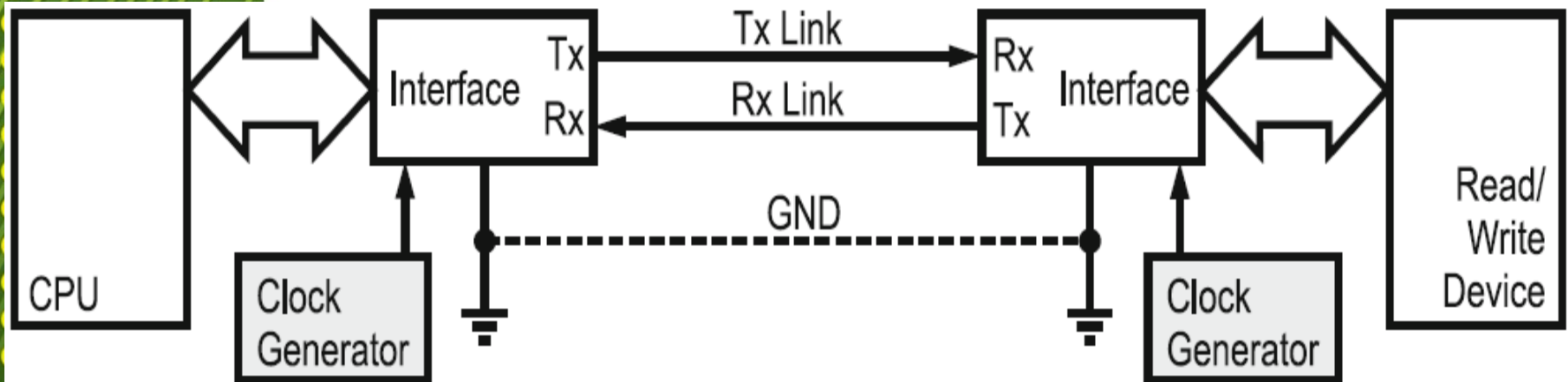
- I asinhroni i sinhroni serijski komunikacioni protokoli dele poruke koje prenose preko kanala na osnovne komponente koji se nazivaju **paketi podataka** ili **datagrami**
- Svaki paket sastoji se iz tri dela:
 - **zaglavlja (header)**
 - **tela (body)**
 - **repa (footer)**
- Header i footer započinju, odnosno završavaju, svaki paket i sadrže sinhronizaciona informaciona polja koja su dodata od strane **komunikacionog protokola** kako bi se obezbedio pouzdan prenos podataka kroz komunikacioni kanal

Sinhronizacija unutar serijskih kanala

- **Header paketa** sadrži:
 - Polje koje označava **početak paketa**
 - Opciona polja koja **sadrže adresne informacije** neophodne u slučaju postojanja većeg broja potencijalnih prijemnika (multi-point topologija)
 - Opciona polja koja sadrže **informaciju o dužini i tipu paketa**
- **Footer paketa** sadrži:
 - Polje koje označava **kraj paketa**
 - Opciono polje koje sadrži informacije na osnovu kojih se može proveriti **ispravnost primljenog paketa** (frame/packet check sequence)
- Header i footer delovi paketa razlikuju se po dužini i formatu u slučaju asinhronih i sinhronih kanala kao i od komunikacionog protokola koji se koristi

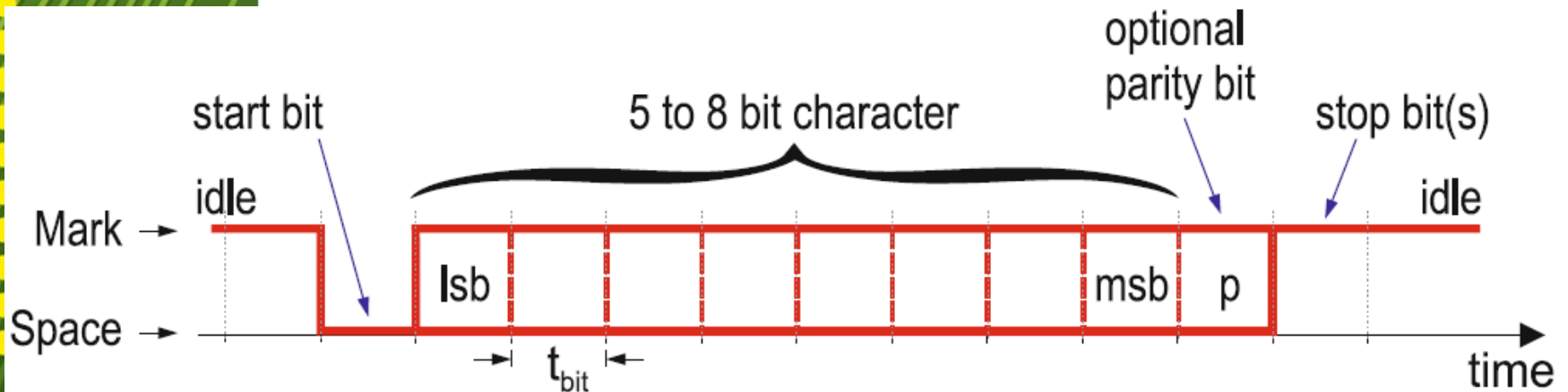
Asinhrona serijska komunikacija

- U asinhronim komunikacionim kanalima, na svakom kraju kanala postoji **nezavisni izvor sinhronizacionog signala**
- Da bi se obezbedila pouzdana komunikacija, uređaji na oba kraja kanala konfigurisani su na takav način da generišu podatke **istom brzinom** i koriste **isti protokol** za razmenu paketa.
- Slika ispod prikazuje topologiju **asinhronog, full-duplex serijskog kanala** sa jasno vidljivim nezavisnim izvorima sinhronizacionog signala.



Format asinhronog paketa

- Asinhroni paketi podataka imaju vrlo jednostavnu strukturu
- Njihov **header** sadrži samo jedan bit, koji se zove **start bit**. Vrednost start bita jednaka logičkoj nuli ("space" simbolu) označava početak paketa.
- **Telo paketa** sadrži karakter, kodovan pomoću 5 do 8 bita organizovanih od bita najmanje značajnosti (LSB) do bita najveće značajnosti (MSB)
- **Footer** sadrži opcioni **bit parnosti**, koji omogućava detekciju 1-bitnih grešaka koje su se javile prilikom prenosa kao i **jedan ili više stop bita** koji označavaju kraj paketa. Vrednost stop bita jednaka logičkoj jedinici ("mark" simbolu) označava kraj paketa.



Format asinhronog paketa

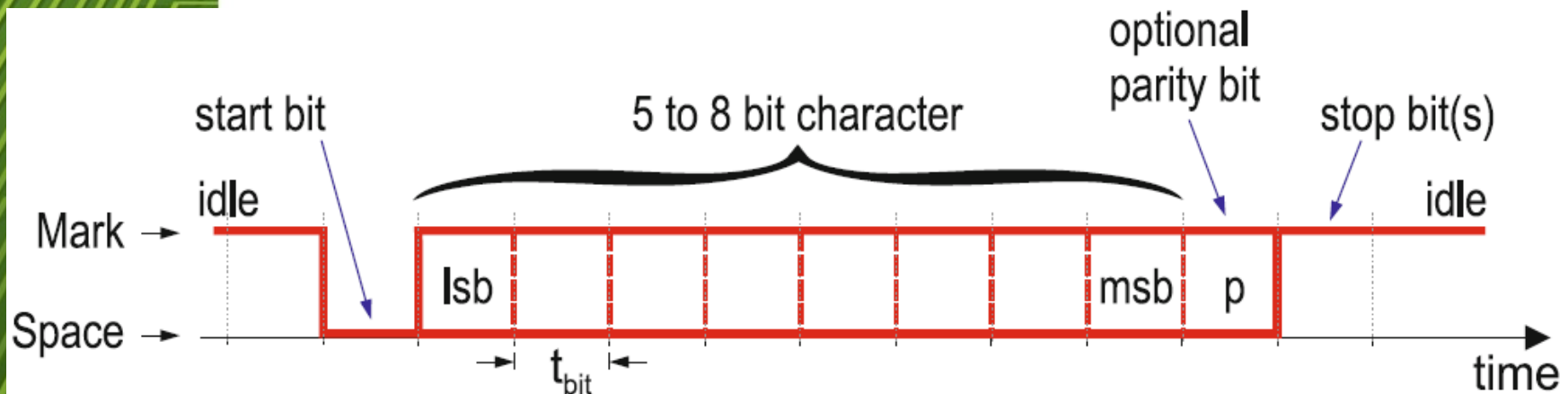
- Kada je **asinhroni kanal u neaktivnom stanju** (*idle state*, nema prenosa podataka) nalazi se u stanju **logičke jedinice**
- Start bit ukazuje na tranziciju kanala iz **neaktivnog** u **aktivno** stanje
- Validni start bit treba da ima trajanje od **jednog bitskog vremena**, t_{bit}
- Kada prijemnik detektuje tranziciju sa logičke jedinice na logičku nulu, dok se nalazi u neaktivnom stanju, on je shvata kao početak **novog paketa podataka**
- Da bi **potvrdio** činjenicu da je u toku prijem start bita, prijemnik obično proverava vrednost asinhronne komunikacione linije još jednom, nakon $t_{bit}/2$ vremena na **sredini bitskog vremena**, i ukoliko je vrednost linije i dalje na logičkoj nuli to je potvrda da je u toku prijem start bita

Format asinhronog paketa

- Nakon izvršene validacije start bita, prijemnik očitava (sample) asinhronu liniju u **regularnim vremenskim intervalima** dužine t_{bit} sekundi, kako bi preuzeo preostale bitove tekućeg paketa
- Na **predajnoj strani**, brzina slanja individualnih bitova paketa određena je na osnovu **lokalnog izvora sinhronizacionog signala**
- Na prijemnoj strani **lokalni sinhronizacioni signal prijemnika** određuje brzinu prijema podataka
- Iako su predajni i prijemni izvori sinhronizacionog signala međusobno nezavisni, obe strane moraju da se “dogovore” o korišćenju iste **brzine slanja/prijema podataka**
- Uređaji na oba kraja linka obično koriste **delitelje učestanosti** kako bi generisali potrebno bitsko vreme na osnovu njihovih lokalnih izvora sinhronizacionog signala

Format asinhronog paketa

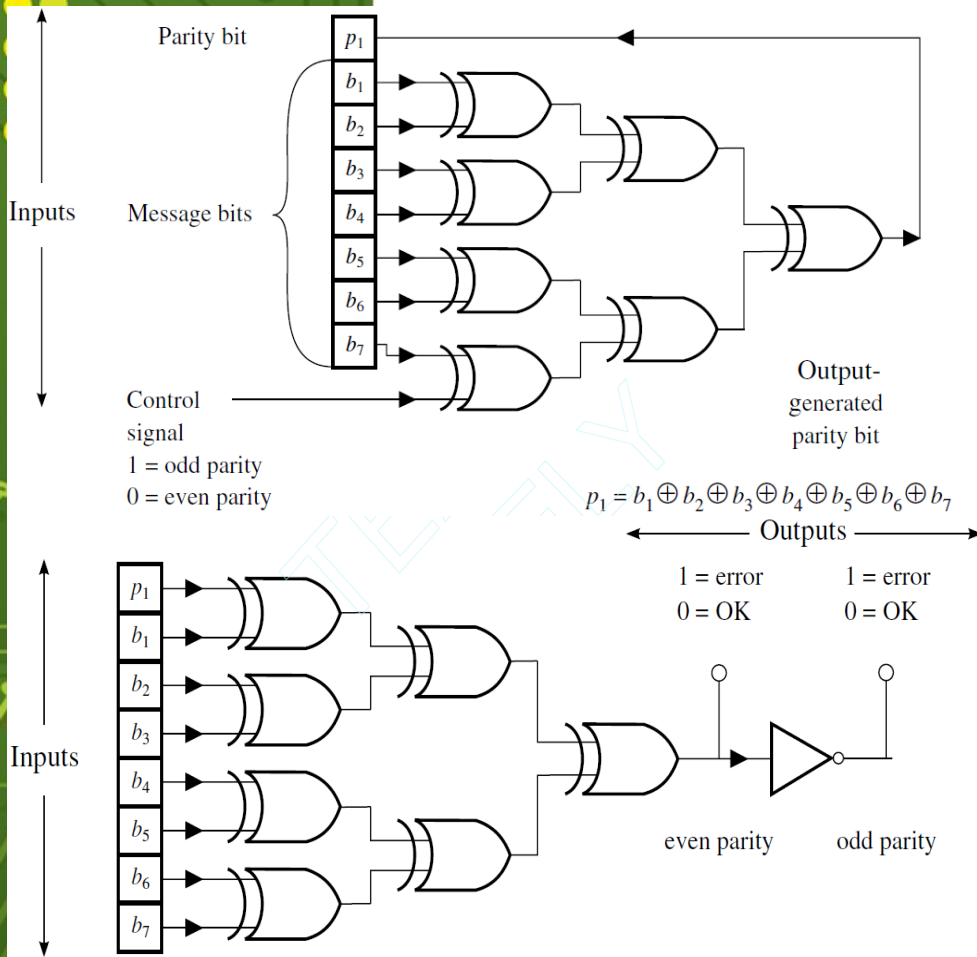
- Usled **drifta** koji se obično javlja kada se koriste nezavisni izvori sinhronizacionog signala, dužina asinhronog paketa se mora držati **malom**, a start bit je neophodan da bi se izvršila **resinhronizacija** predajnika i prijemnika
- Takođe, da bi se obezbedila **pouzdana detekcija** vrednosti individualnih bitova, asinhrona linija se obično očitava (sempluje) sa učestanošću koja je k puta (16, 32, 64) veća od bitske brzine koja se koristi



Detekcija greške prilikom asinhronog prenosa

- **Detekcija grešaka** prilikom asinhronog prenosa podataka ostvaruje se pomoću **bita parnosti**, koji je opcioni deo footer dela paketa
- Bit parnosti predstavlja elementarni mehanizam detekcije grešaka pomoću kojega je moguće detektovati **1-bitne greške** koje se mogu javiti prilikom prenosa podataka
- Bit parnosti predstavlja indicaciju parnog (**even parity**) ili neparnog (**odd parity**) broja logičkih jedinica u tekućem karakteru, plus bit parnosti, koji se prenosi
- U slučaju **even parity** varijante, bit parnosti se postavlja na nulu ili jedinicu kako bi ukupan broj jedinica u karakteru koji se prenosi i bitu parnosti bio **paran**
- U slučaju **odd parity** varijante, bit parnosti se postavlja na potrebnu vrednost koja će obezbediti da broj jedinica u karakteru zajedno sa bitom parnosti bude **neparan**

Detekcija greške prilikom asinhronog prenosa



- **Predajnik**, prilikom slanja tekućeg karaktera **računa** i potrebnu vrednost bita parnosti, u zavisnosti od odabrane **varijante (even ili odd)** koju šalje u okviru footer dela paketa
- Prijemnik ponavlja istu operaciju prilikom prijema karaktera
- Ukoliko se izračunata vrednost bita parnosti na prijemnoj strani ne poklapa sa vrednošću koja je bila poslata, došlo je do pojave 1-bitne greške
- **Logičke mreže** za računanje bita parnosti su vrlo jednostavne i sastoje se iz odgovarajućeg **broja XOR logičkih kapija**, kao što je prikazano na slici desno

Generisanje paketa

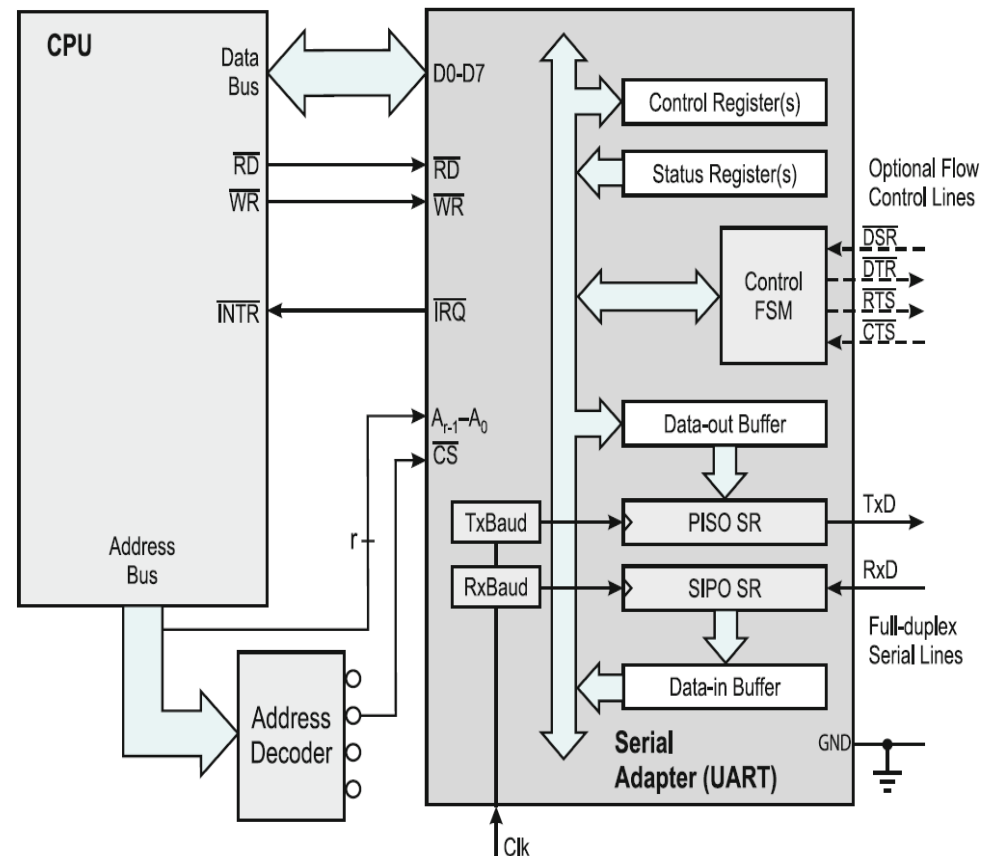
- Start bit, bit parnosti i stop bit automatski se **dodaju od strane modula za serijsku komunikaciju na predajnoj strani**
- Na prijemnoj strani ovi bitovi se takođe **automatski odstranjuju**, pri čemu se vrši detekcija potencijalnih grešaka nastalih prilikom prenosa ukoliko se koristi bit parnosti
- Korisnički softver ne mora da brine o ovim detaljima, jedino treba da obezbedi karakter koji je potrebno poslati preko asinhronog serijskog linka ili obradi primljeni karakter
- U slučaju detekcije greške, korisnički softver će odrediti eventualne neophodne korake za oporavak od greške
- **Modul za asinhronu serijsku komunikaciju** poznat je pod nazivom **UART** (Universal Asynchronous Receiver and Transmitter)

Generisanje paketa

- Da bi se obezbedila komunikacija između dva uređaja pomoću asinhronog serijskog komunikacionog kanala oba uređaja moraju programirati njihove UART module na sledeći način:
 - Da koriste istu bitsku brzinu
 - Da koriste isti broj bita prilikom prenosa karaktera
 - Da li će koristiti detekciju grešaka prilikom prenosa, i ako hoće, koji tip detekcije će biti korišćen (even parity ili odd parity)
 - Koliki broj stop bita će biti korišćen za označavanje kraja tekućeg paketa

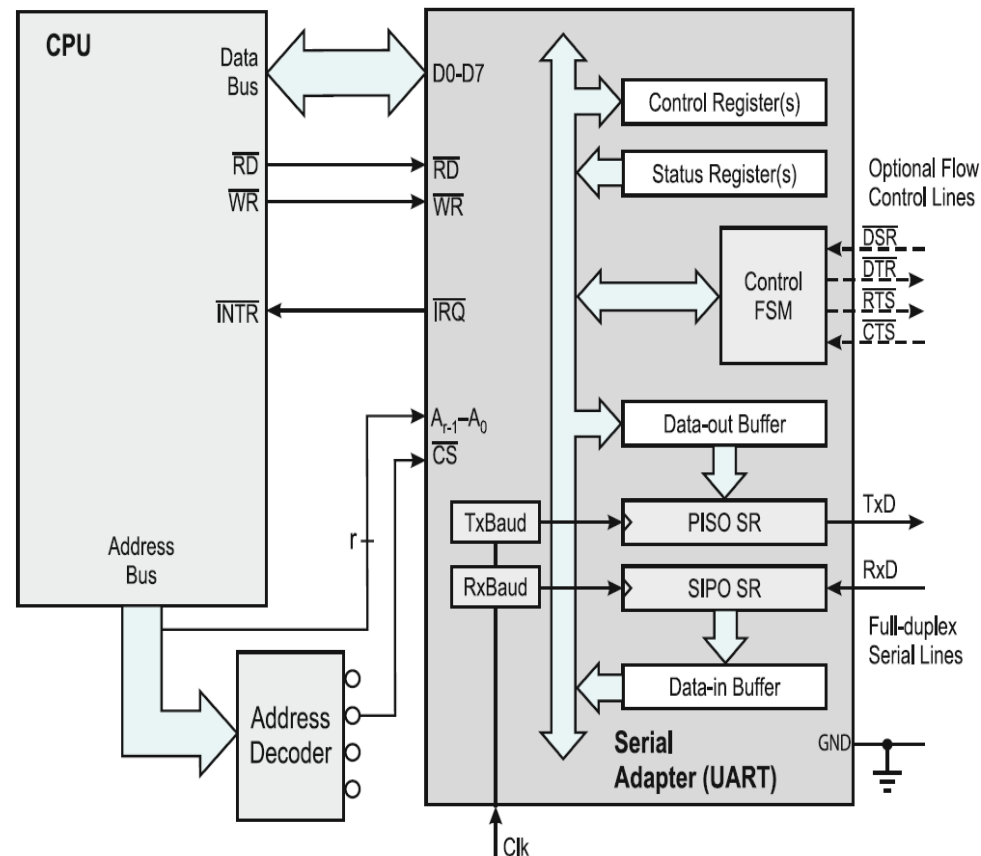
Struktura standardnog UART modula

- UART kombinuje funkcionalnost **predajnika i prijemnika** unutar jednog modula sa ciljem obezbeđivanja serijskog interfejsa **sa full-duplex mogućnošću**
- Unutrašnje komponente UART-a centrirane su oko dva **pomeračka registra**:
 - PISO** (Parallel-Input Serial-Output) registra unutar predajnika
 - SIPO** (Serial-Input Parallel-Output) registra unutar prijemnika
- Da bi inicirao prenos, CPU upisuje podatak (karakter) koji je potrebno preneti u **Data-out bafer**
- Svaki karakter, jedan po jedan, se zatim okružuje sa odgovarajućim header i footer poljima, što je funkcija koju realizuje **upravljački automat (Control FSM)** i upisuje u PISO registar



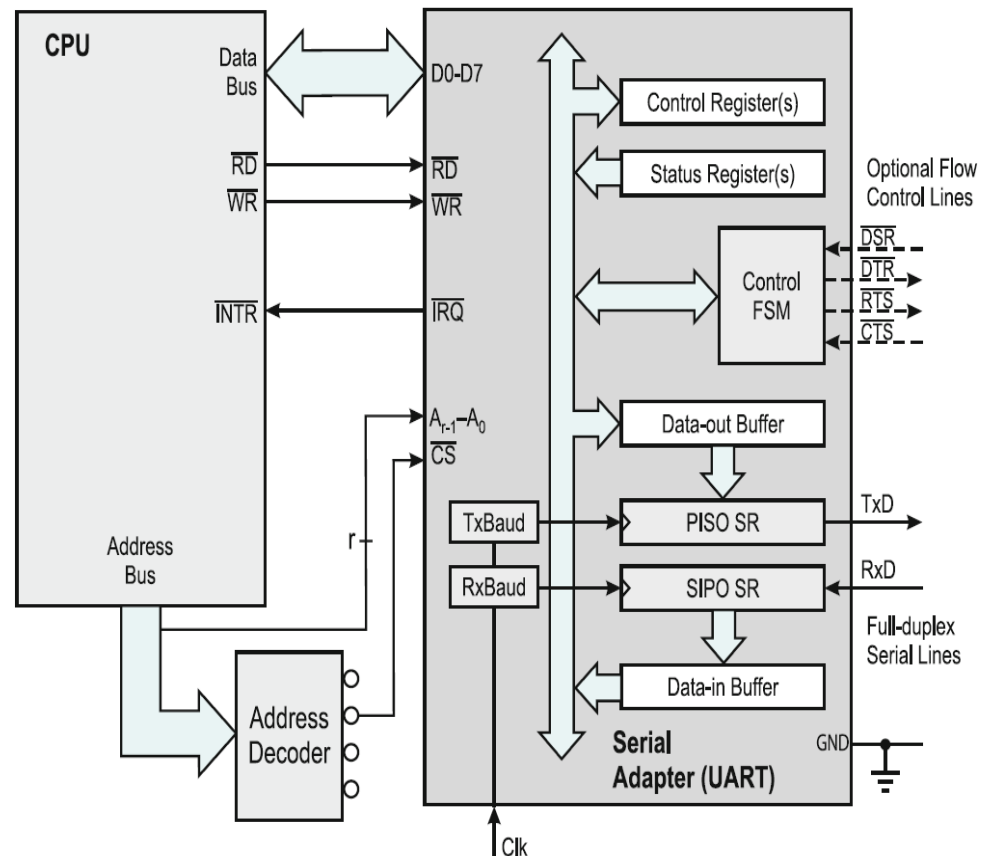
Struktura standardnog UART modula

- U ovom trenutku upravljački automat takođe računa i vrednost bita parnosti (ukoliko se on koristi)
- Sadržaj PISO registra** se zatim šalje preko serijskog linka brzinom koju određuje baud-rate generator predajnika (**TxBaud**)
- Kada se svi karakteri iz Data-out bafera pošalju, UART podiže **TxReady** zastavicu, označavajući da je spreman za prijem novih karaktera koje je potrebno poslati preko serijskog linka
- Upis u Data-out bafer automatski briše TxReady zastavicu
- Jednostavni UART moduli imaju Data-out bafer u koji se može smestiti samo jedan karakter, dok je kod složenijih UART-a ovaj bafer veći i dostiže veličine i od nekoliko kilobajta



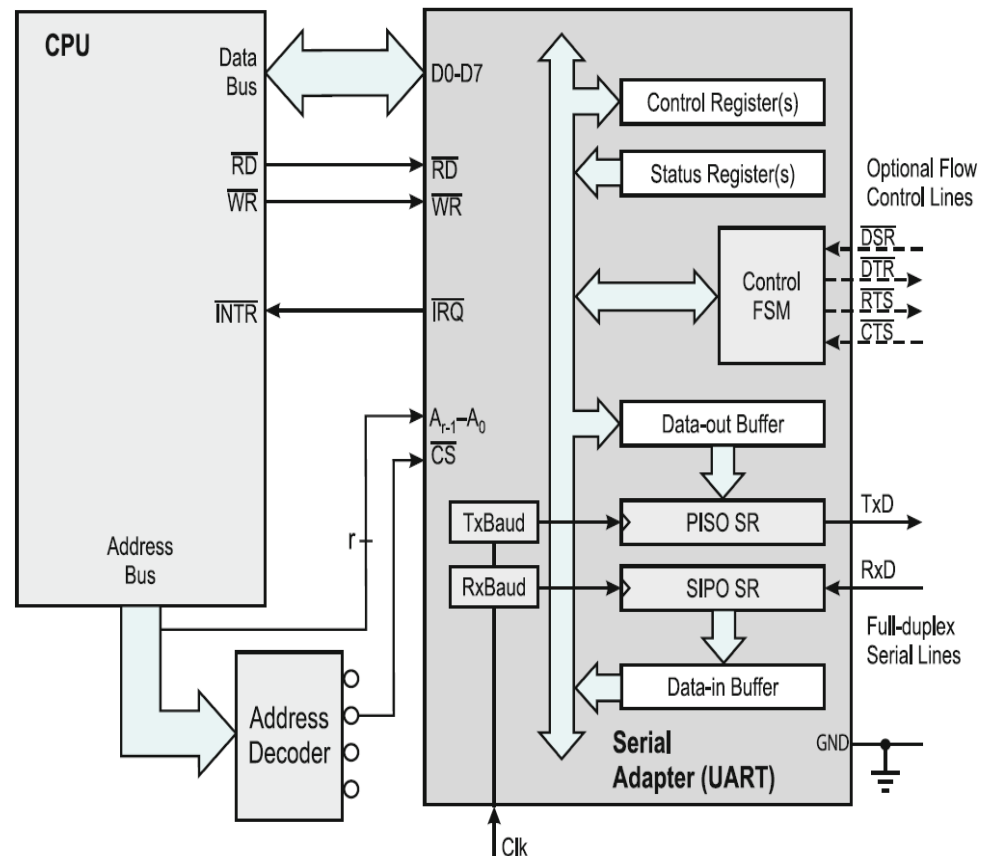
Struktura standardnog UART modula

- Prilikom prijema, validni start bit **resetuje SIPO** registar i započinje upis bitova koji se primaju preko serijskog linka u SIPO registar
- Kada su svi bitovi smešteni u SIPO registar, **upravljački automat** uklanja start i stop bitove, izvodi proveru parnosti (ukoliko je neophodna) i uklanja bit parnosti, postavlja primljeni karakter u Data-in bafer i podiže **RxReady zastavicu**
- Kada detektuje da je RxReady zastavica aktivirana, CPU čita primljeni karakter iz Data-in bafera
- Čitanje Data-in bafera obično briše RxReady zastavicu
- U slučaju da su detektovane greške parnosti ili je paket nepravilno formiran, UART signalizira novonastalu situaciju podizanjem odgovarajućih zastavica unutar **Status registra**



Struktura standardnog UART modula

- Ukoliko se prilikom prijema Data-in bafer **puni brže** nego što CPU iz njega čita karaktere, može doći do **gubitka** nekih od **karaktera** koji su bili poslati
- Ovakva situacija poznata je pod nazivom **“overrun error”**
- Da bi se izbegla ova situacija u praksi se koriste Data-in baferi koji imaju mogućnost prihvatanja **većeg broja karaktera**, na taj način oslobađajući CPU od potrebe za brzom reakcijom nakon indikacije prijema podataka
- PISO i SIPO registri rade na učestanosti **predajnog i prijemnog takta** koji određuju i brzine predaje i prijema podataka
- Svi indikatori koji označavaju trenutni status predajnika, prijemnika i eventualno postojanje grešaka dostupni su preko odgovarajućih **statusnih registara**

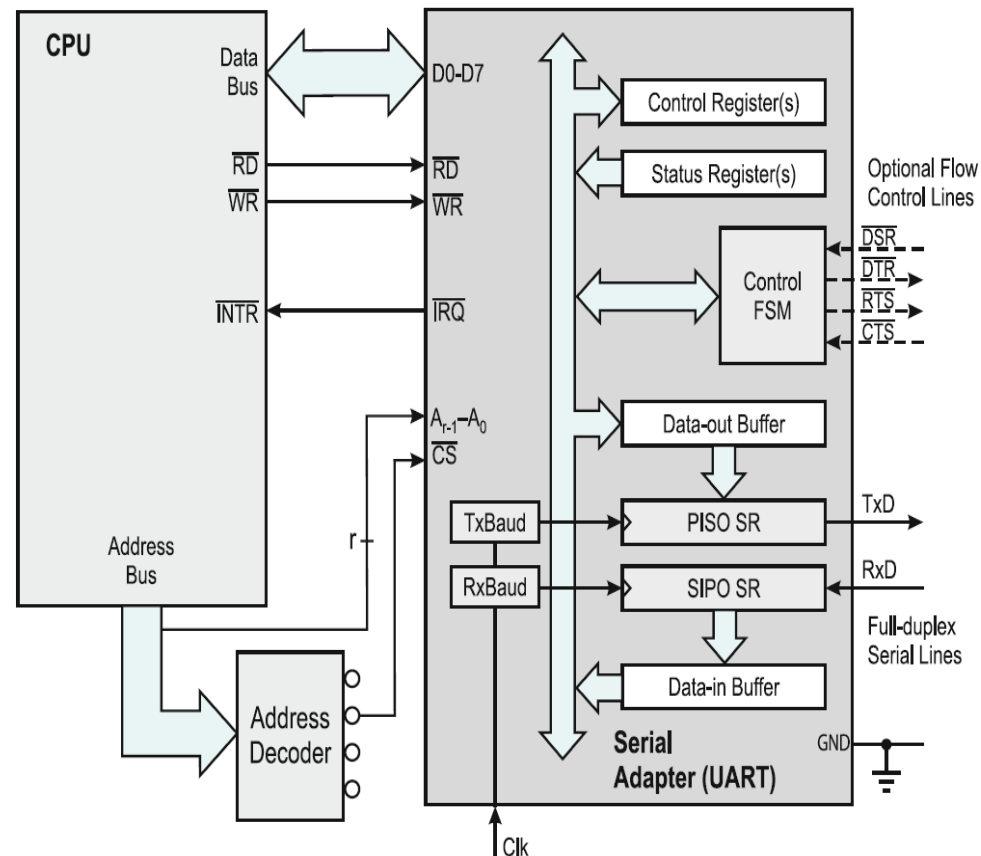


Interfejs standardnog UART modula

- Interfejs UART modula sastoji se iz tri celine:
- **CPU interfejsa** – koji omogućava povezivanje UART modula sa procesorom preko njegove **adresne, kontrolne i magistrale podataka**
- **Klok interfejsa** – koji obezbeđuje **referentni izvor sinhronizacionog signala** na osnovu kojega će biti određene bitske brzine predajnika i prijemnika
- **Interfejsa serijskog kanala** – preko kojega se vrši povezivanje UART modula sa **fizičkim serijskim linkom**

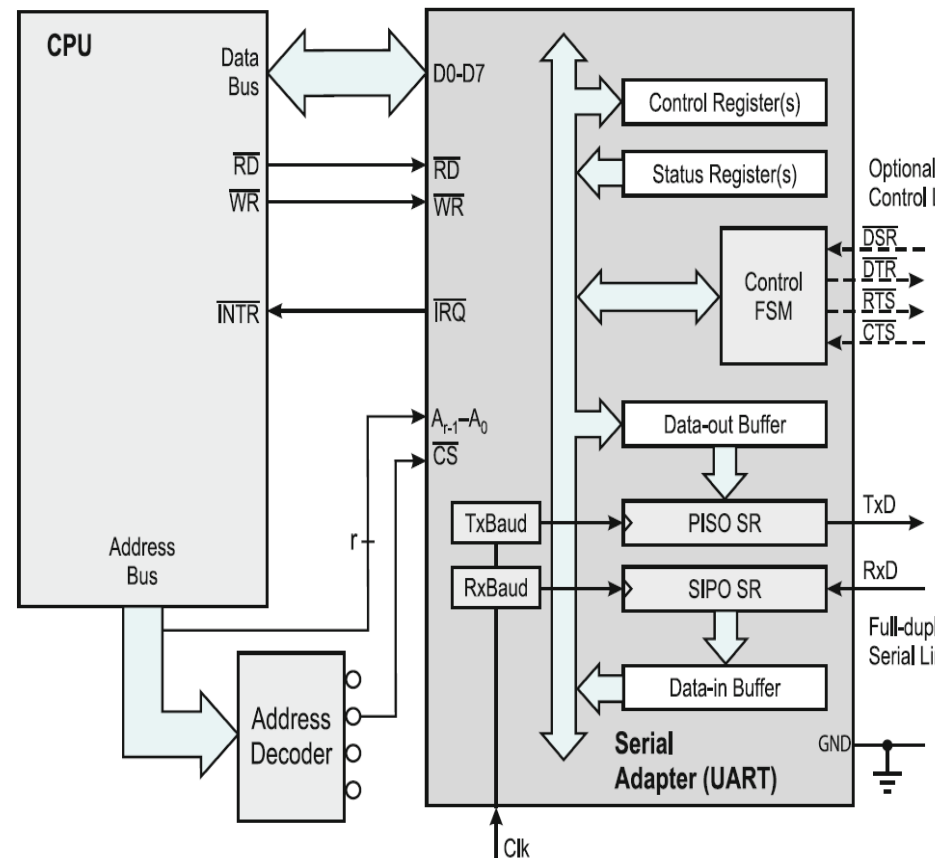
CPU interfejs

- UART se povezuje sa procesorom preko sledećih magistrala:
- **Magistrale podataka** – skupa bidirekcionih linija podataka (obično je reč o 8-bitnoj magistrali podataka) koje služe za prenos **podataka od i ka UART modulu**
- **Read/write kontrolnih signala** – koji određuju smer toka podataka
- **CS signala** – preko kojega se aktivira UART modul
- **Adresne magistrale** – skupa unidirekcionih selekcionih linija ($A_{r-1} - A_0$), preko kojih je moguće jednoznačno odrediti kojem **unutrašnjem registru** se želi pristupiti



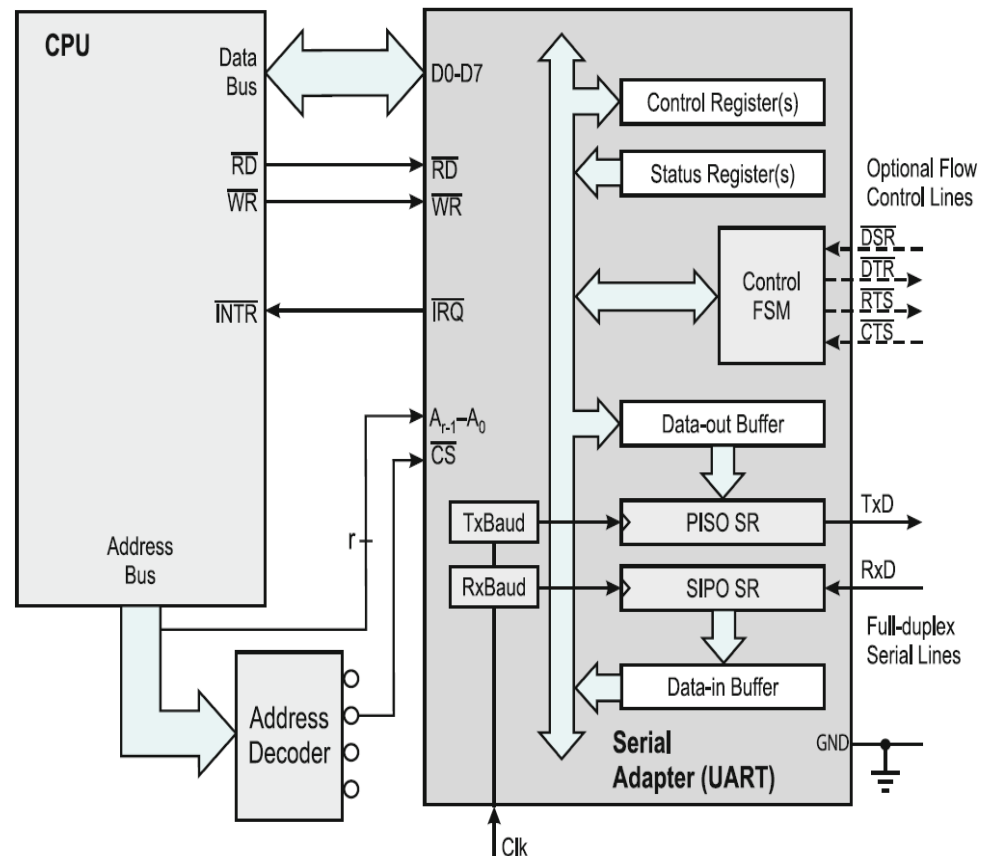
CPU interfejs

- CPU interfejs obično uključuje i jednu ili više **linija za generisanje zahteva za prekidom (IRQ)** pomoću kojih UART modul može da generiše zahtev za opsluživanjem od strane procesora
- U zavisnosti od modela UART-a postoji jedna ili više nezavisnih IRQ linija
- U slučaju postojanja samo jedna IRQ linije, indikacija o uspešnoj predaji i prijemu podataka kao i indikacija o eventualnim greškama prilikom predaje i prijema signalizirala bi se preko iste IRQ linije
- Prilikom obrade zahteva za prekidom CPU bi prvo morao da utvrdi **razlog za prekidom** (proveravajući sadržaj statusnih registara) kako bi utvrdio redosled akcija potreban za obradu prekida
- Postojanje višestrukih IRQ linija omogućuje razdvajanje individualnih izvora prekida



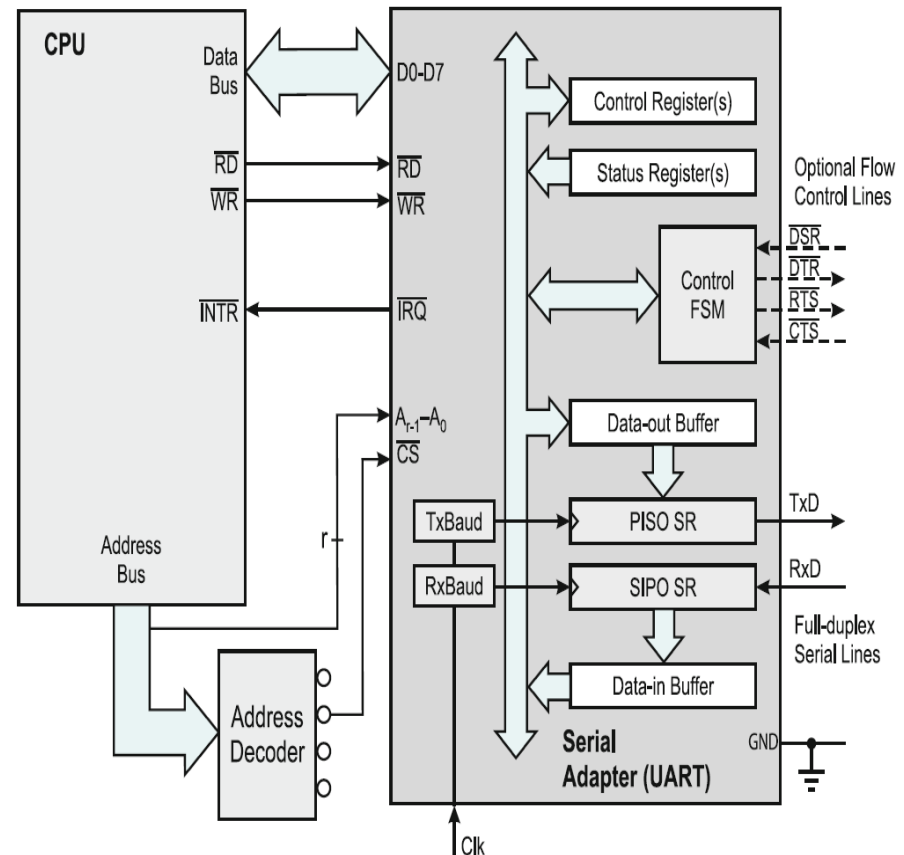
Klok interfejs

- Svaki UART modul poseduje jednu ili više **ulaznih klok linija** koje su povezane na unutrašnje generatore bitskih brzina predajnika i prijemnika
- Ovi klok ulazi obezbeđuju **referentni sinhronizacioni signal** učestanosti f_{clk} koji se zatim deli pomoću unutrašnjih **delitelja kloka (TxBaud i RxBaud)** kako bi se generisale bitske brzine predajnika i prijemnika
- Većina mikrokontrolera koji na sebi imaju integrisani UART modul, generiše potrebne bitske brzine na osnovu **sistemskog kloka**, mada kod nekih postoji mogućnost korišćenja posebnih **spoljašnjih izvora** referentnog klok signala



Interfejs serijskog kanala

- Glavne **signalne linije interfejsa** ka serijskom kanalu su TxD and RxD, koje zajedno sa linijom referentnog napona (GND) prenose dolazne i odlazne serijske tokove podataka ka i od UART modula
- TxD (Transmitted Data) predstavlja serijski izlazni port koji je direktno povezan na izlaz **PISO registra**
- RxD (Received Data) predstavlja serijski ulazni port koji je direktno povezan na ulaz **SIPO registra**
- Kod većine mikrokontrolera TxD i RxD signali su multipleksirani sa GPIO signalima ili nekim drugim signalima tako da je potrebno izvršiti odgovarajuća konfigurisanja mikrokontrolera ukoliko se žele koristiti

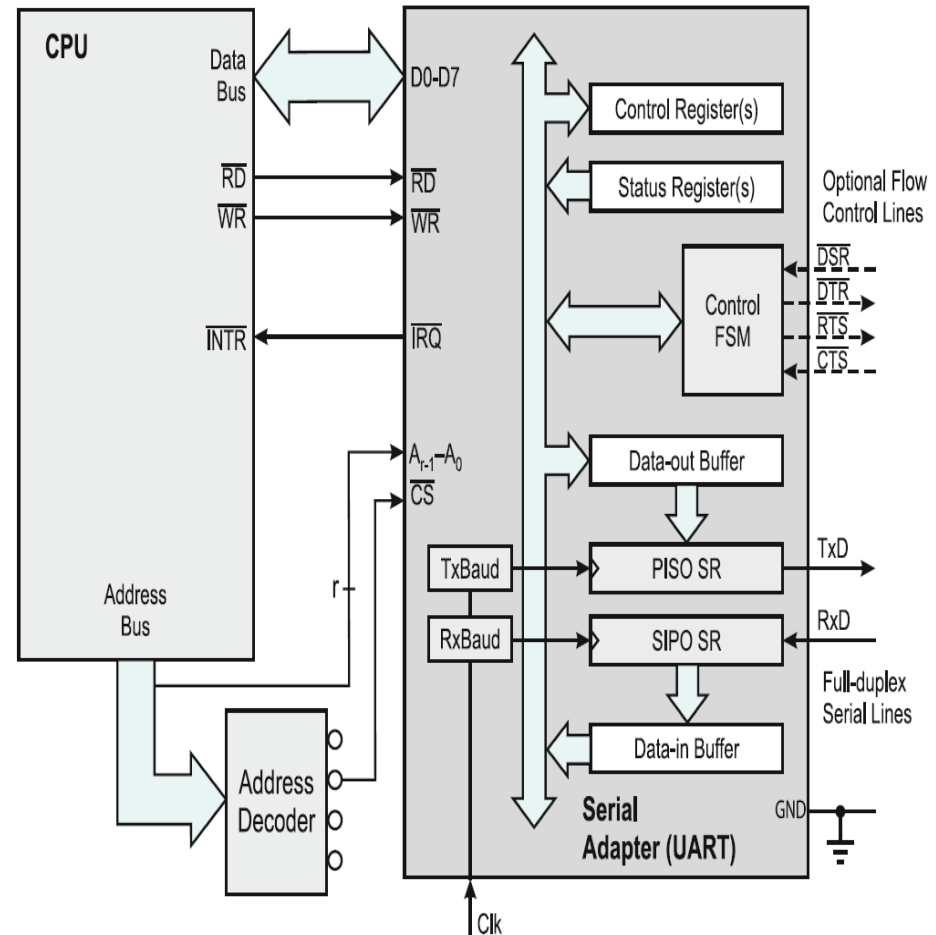


Interfejs serijskog kanala

- Neki UART moduli mogu uključiti i **dodatne signale unutar interfejsa** serijskog kanala pomoću kojih je moguće obezbediti hardversku kontrolu toka podataka (**hardware flow control**):

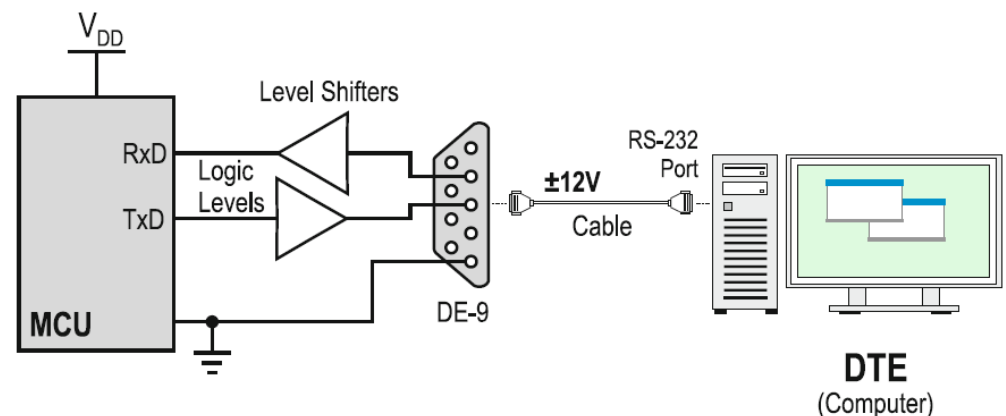
- **DSR** – **Data Set Ready**, ulazni signal UART modula koji predstavlja indicaciju da je spoljašnji komunikacioni modul spreman za rad
- **DTR** – **Data Terminal Ready**, izlazni signal UART modula koji predstavlja indicaciju da je UART modul spreman za rad
- **RTS** – **Ready to Send**, izlazni signal UART modula koji obaveštava spoljašnji uređaj da želi da izvrši slanje podataka
- **CTS** – **Clear to Send**, ulazni signal UART modula koji predstavlja indicaciju da je spoljašnji komunikacioni modul spreman da primi podatke

- Ovi dodatni signali, zajedno sa RI (Ring Indicator) i DCD (Data Carrier Detected) čine skup signala koji definiše dobro poznati **RS-232 komunikacioni interfejs**, međutim većina savremenih mikrokontrolera ih ne podržava



Interfejs serijskog kanala

- Fizički standardi za serijsku komunikaciju obično koriste **veliki dinamički opseg naponskih nivoa i/ili odgovarajuće** signalizacione mehanizme u cilju povećanja “otpornosti” na uticaje šuma prilikom prenosa podataka kroz kanal
- Na primer, u slučaju RS-232 kanala, naponski nivoi koji se koriste za prenos “nule” i “jedinice” su -12 V i +12 V
- Ovi naponski nivoi nisu direktno kompatibilni sa logičkim nivoima signala koje očekuje ili generiše UART modul
- Zbog toga je u većini ovakvih sistema potrebno korišćenje **naponskih translatora (Level Shifter)** kako bi se ovo povezivanje ostvarilo na način koji će obezbediti pouzdan rad i neće dovesti do oštećenja komunikacione opreme





Konfiguracija i rad sa UART modulom

- Korišćenje UART modula sa stanovišta softvera je relativno jednostavan zadatak koji se sastoji iz sledećih koraka:
- **Konfiguracija UART modula**
- Konfiguracija UART modula sastoji se iz sledećih koraka:
- **Izbor izvora klok signala** - poznate učestanosti f_{clk} , za generatore bitskih brzina
- Konfigurisanje generatora bitskih brzina – kada je poznata učestanost referentnog klok signala, f_{clk} , odgovarajuća **bitska brzina BR** postiže se delenjem referentnog kloka sa **faktorom N** koji se računa prema sledećem izrazu
$$N = \frac{f_{clk}}{BR}$$
- Kod većine UART modula za svrhu ovog deljenja koristi se **poseban tajmer**
- Poželjno je odabrati takvu vrednost f_{clk} koja će kada se podeli sa željenom bitskom brzinom BR rezultovati u celobrojnoj vrednosti faktora N

Konfiguracija i rad sa UART modulom

- **Izbor i konfigurisanje mehanizma detekcije grešaka** – ukoliko se želi koristiti **mehanizam za detekciju grešaka** prilikom prenosa karaktera prijemnik i predajnik moraju se konfigurisati tako da omoguće generisanje i proveru bita parnosti prilikom predaje i prijema karaktera
- U slučaju da je korišćenje bita parnosti uključeno na obe strane mora se konfigurisati i način njegovog generisanja (even ili odd mehanizam)
- **Konfigurisanje formata karaktera i dužine stop bita** – većina UART modula može da radi sa različitim dužinama karaktera, najčešće od 5 do 8 bita. Ukoliko je ovo konfigurisanje moguće, onda se prijemnik i predajnik moraju konfigurisati da koriste reprezentaciju karaktera sa istim brojem bita
- Slično, UART moduli sa **konfigurabilnom dužinom stop bita**, moraju se konfigurisati tako da koriste istu dužinu stop bita

Konfiguracija i rad sa UART modulom

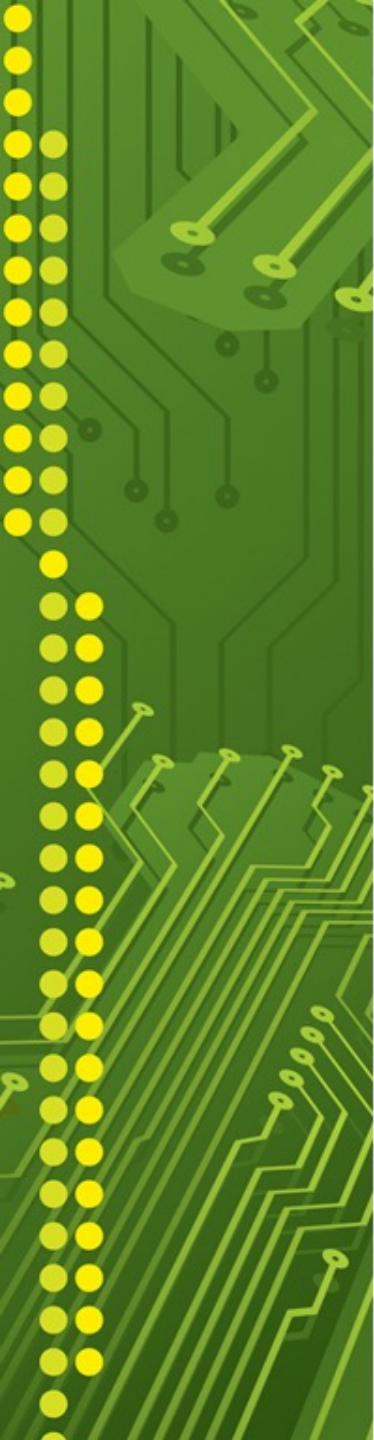
▪ Aktiviranje

- U zavisnosti od smera komunikacije i toga da li će rad UART biti kontrolisan **sistemom prozivke** ili **sistemom prekida**, potrebno je izvršiti nekoliko aktivacija:
- Neki od UART modula zahtevaju eksplicitno aktiviranje predajnika i prijemnika pojedinačno ili zajedno kako bi UART počeo sa svojim radom
- U slučaju korišćenja sistema prekida, odgovarajući **zahtevi za prekidom** predajnika (Tx) i prijemnika (Rx) moraju se **dozvoliti**
- Ukoliko različite **detektovane greške** prilikom rada UART modula treba da **generišu zahteve za prekidom** moraju se aktivirati odgovarajuće biti **koji dozvoljavaju generisanje ovih zahteva**

Konfiguracija i rad sa UART modulom

▪ Rad

- Sam rad sa UART modulom, nakon što je pravilno konfigurisan i aktiviran je relativno jednostavan
- U slučaju korišćenja sistema prozivke softver mora **periodično proveravati spremnost** predajnika za prenos sledećeg karaktera (**TxReady**) i status prijemnika (**RxReady**) pre nego što se izvrši bilo kakva operacija nad ulaznim i izlaznim baferom UART modula
- Nakon prijema svakog karaktera moraju se proveriti biti grešaka kako bi se utvrdilo da li nije došlo do neke greške prilikom prijema tekućeg karaktera
- U slučaju da se utvrdi da je došlo do greške prilikom prijema karaktera softver mora da izvrši odgovarajuće korektivne akcije

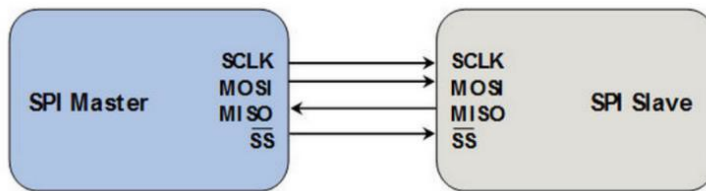


Konfiguracija i rad sa UART modulom

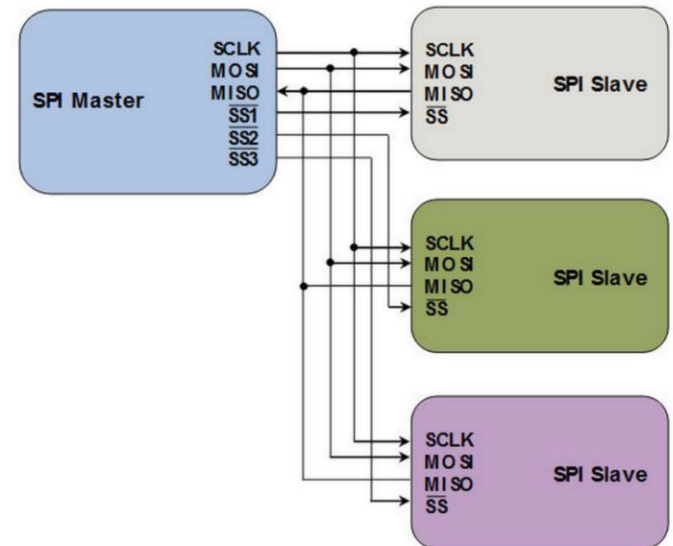
- U slučaju korišćenja **sistema prekida**, softver za upravljanje UART modulom je još jednostavniji
- Aktiviranje zahteva za prekidom od strane UART prijemnog modula znači da je **primljen novi karakter** te je u prekidnom potprogramu potrebno izvršiti **preuzimanje primljenog karaktera** i proveriti da li je bilo grešaka prilikom njegovog prijema
- U slučaju da je **UART predajnik generisao** zahtev za prekidom, prekidni potprogram treba da upiše naredni podatak koji je potrebno poslati u predajni bafer (Data-out buffer) UART modula

SPI (serijski periferni interfejs)

- SPI (engl. Serial Peripheral Interface) je protokol namenjen komunikaciji između integriranih kola, kao i za relativno sporu komunikaciju sa perifernim jedinicama koje se nalaze na istoj štampanoj ploči.
- Arhitektura SPI magistrale je takva da postoji jedan glavni (engl. *master*) uređaj i jedan ili više sporednih (engl. *slave*) uređaja.
- SPI interfejs koristi 4 linije, sa sledećim funkcionalnostima:
 1. **SCLK** (engl. *Serial CLock*) je taktni signal koji uvek generiše master i vodi se ka svim slave uređajima. Prenos podataka putem SPI magistrale sinhronizovan je u odnosu na ovaj signal.
 2. **SS_n** signal ima ulogu odabira slave uređaja (engl. *Slave Select*). Aktivnim stanjem SS signala odabira se jedan od povezanih slave uređaja sa kojim će master ostvariti komunikaciju.
 3. **MOSI** (engl. *Master Out - Slave In*) je linija za prenos podataka od mastera ka svim slave uređajima.
 4. **MISO** (engl. *Master In - Slave Out*) je linija za prenos podataka od slave uređaja ka master uređaju.

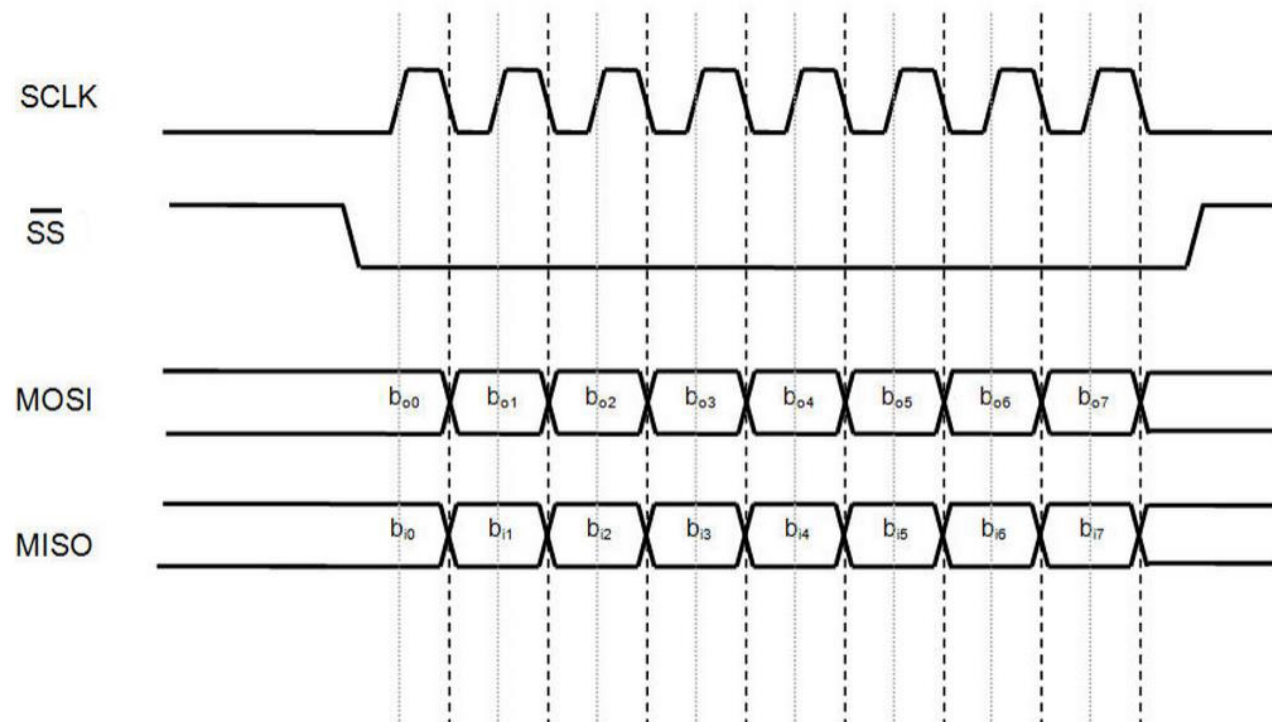


Jedan master - jedan slave uređaj



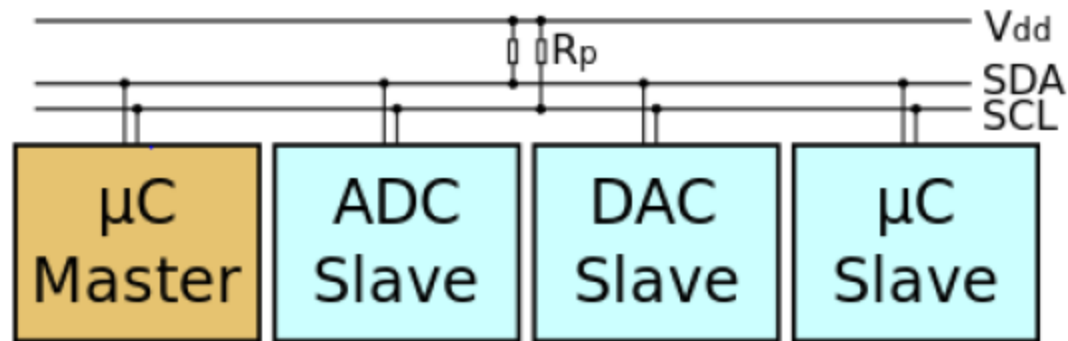
Jedan master - više slave uređaja

- Master inicira svaku komunikaciju sa ostalim uređajima na SPI magistrali. Kada SPI master želi da šalje podatak slave-u, ili da prihvati neki podatak od slave-a, on će ga adresirati tako što odgovarajuću SS liniju obori na nizak naponski nivo i aktivira taktni signal.
- Master šalje podatke putem MOSI linije, dok ih prima preko MISO linije.
- 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).
- Primer na slici pokazuje mod u kojem se ažuriranje bita informacije na MISO i MOSI liniji vrši na silaznu ivicu SCLK signala (tj. i master i slave tada postavljaju nove bite), dok se na rastućoj ivici učitavaju vrednosti na komunikacionim linijama.



I²C komunikacija

- I²C (engl. *Inter-Integrated Circuit*) je protokol za komunikaciju između integrisanih kola koji koristi svega dve linije:
 - **SDA** (engl. *Serial Data*) - Linija za prenos podataka
 - **SCL** (engl. *Serial Clock*) - Linija za takt
- Arhitektura I²C magistrale je **multi master-multi slave**, što znači da se na istoj magistrali može naći više master i više slave uređaja (do 128). Uloge ovih uređaja su:
 - **Master** - Generiše takt i komunicira sa slave uređajima
 - **Slave** - Prima takt i odgovara kada je adresiran od strane mastera



- Uređaji se povezuju na I²C magistralu preko bidirekcionih linija sa open-collector (open-drain) izlaznim stepenom. Između linija SDA i SCL i napona napajanja nalaze se pull-up otpornici. Kada je magistrala **neaktivna**, ovi otpornici postavljaju stanje linija na **logičku jedinicu**.

Opis I²C protokola

- I²C protokol podrazumeva:
 - 7-bitno adresiranje slave uređaja, pri čemu svaki uređaj povezan na I²C magistralu ima jedinstvenu adresu. Na ovaj način **broj uređaja** na magistrali ograničen je na **128**.
 - podatke podeljene u bajtove (8 bita informacije); prilikom slanja svakog bajta, biti se šalju redom od **MSB ka LSB**
 - 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**. Time se signalizira svim uređajima koji se nalaze na I²C magistrali da krenu da osluškiju 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 (signal potvrde), kako bi master znao da je uređaj sa traženom adresom pronađen. Nakon prijema ACK signala master nastavlja sa **slanjem ili čitanjem slave** uređaja i po završetku komunikacije generiše **STOP uslov**.
- U slučaju da master želi da **šalje** podatke slave uređaju, nakon 7 bita adrese šalje se logička **nula**, dok se logička **jedinica** šalje u slučaju kada je potrebno **čitati** podatke poslate od strane slave-a.
- Nakon svakog primljenog bajta, prijemni uređaj koji može biti master ili slave šalje **ACK signal (logička nula na SDA liniji)** nakon čega se nastavlja sa sledećim bajtom. Prilikom prijema više bajta podataka od strane slave-a, master šalje NACK (logičko 1 na SDA liniji) nakon prijema poslednjeg bajta i time daje do znanju slave-u da je transfer završen.

Format I²C poruka

- Sledeća tabela prikazuje 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 od mastera ka slave** uređaju, podaci se generišu u skladu sa sledećom tabelom. Signali koje generiše master su osenčeni, dok su u ćelijama sa belom pozadinom prikazani signali koje generiše slave.

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

- Situacija u **slučaju prijema** je prikazana je u narednoj tabeli (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

I²C: START uslov, STOP uslov i vremenski dijagram transfera podataka

- START i STOP signali se generišu kao specijalna kombinacija naponskih nivoa na SDA i SCL liniji.
- Prilikom postavljanja podataka na SDA liniju (osim START i STOP karaktera) koriste se pravila prikazana na vremenskom dijagramu:
 - Podaci **su validni i ne smeju se** menjati tokom intervala vremena kada je stanje naponskog signala na SCL liniji stabilno na **visokom naponskom nivou** (označeno zelenom bojom na dijagramu)
 - **Sve promene** naponskog nivoa na SDA moraju se izvršiti tokom intervala kada je **naponski nivo na SCL** liniji nizak (označeno plavom bojom na dijagramu).
- START uslov generiše se kao opadajuća ivica SDA signala u trenutku kada je napon na SCL liniji visok (označeno slovom „S“ na dijagramu).
- Slično, STOP uslov predstavljen je rastućom ivicom SDA signala u trenutku kada je napon na SCL visok visok (označeno slovom „P“ na dijagramu).

