

SERIJSKE MAGISTRALE

=====

Prema nacinu transfera, magistrale se dele na SERIJSKE i PARALELNE magistrale. Serijske megistrale se sastoje iz jedne linije preko koje se podaci prenose bit po bit. Paralelne se sastoje iz vise linija preko kojih se podaci prenose rec po rec.

Iako su u ranijem periodu dominirale paralelne magistrale koje su u vecini slucajeva bile visestruko brze od serijskih (sto deluje logicno na prvi pogled), danas se sve vise koriste serijske magistrale koje su jednostavnije i jeftinije za implementaciju, a u mnogim situacijama omogucavaju i znatno vece brzine prenosa od paralelnih. Naime, postoje dva osnovna problema koja se javljaju pri daljem povecavanju brzine prenosa kod paralelnih magistrala:

-) Prilikom prenosjenja reci, ne stizu svi bitovi na odrediste u isto vreme (pojava poznata kao i ISKRIVLJENOST, engl. skew). Ovo je zato sto nisu sve linije iz kojih se sastoji paralelna magistrala identicne duzine i provodljivosti. Ove vremenske razlike su veoma male, ali pri velikim frekvencijama prenosa dolaze do izrazaja i otezavaju prijemnom uredjaju da ispravno ocita vrednosti sa magistrale.
-) Pri velikim frekvencijama prenosa dolazi do veoma brzog smenjivanja jedinica i nula (tj. naponskih nivoa) na linijama magistrale. Ovo dovodi do elektromagnetne indukcije, sto dovodi do toga da signali sa razlicitih paralelnih linija uticu jedni na druge (pojava poznata i kao INTERFERENCIJA). Ovo moze dovesti do promene vrednosti prenetih bitova ili nemogucnosti da se isti ocitaju pravilno.

Sa druge strane, kod serijskih magistrala takvih problema nema, pa je kod njih moguće visestruko povecavati frekvenciju prenosa (broj transfera u sekundi), sto ih u praksi cini brzim od paralelnih magistrala.

Dodatno, serijske magistrale mogu biti i dvosmerne (engl. full-duplex) a ne samo jednosmerne (engl. half-duplex). Kod dvosmernih imamo dve linije, od kojih se jedna koristi za prenos u jednom smeru, a druga za prenos u drugom smeru. Na taj nacin se podaci mogu istovremeno prenositi u oba smera. Sa druge strane, paralelne magistrale su po pravilu jednosmerne, jer bi realizacija dvosmerne paralelne komunikacije bila preskupa.

Zbog svega ovoga, danas se na razlicitim nivoima racunarskog sistema umesto ranijih paralelnih (PCI, PATA, eksterni paralelni port) koriste serijske magistrale (PCI-EXPRESS, SATA, USB, ...). Izuzetak je MEMORIJSKA MAGISTRALA koja povezuje procesor sa operativnom memorijom koja je i dalje paralelna (razdvojena) magistrala.

NAPOMENA: Serijske magistrale po pravilu podrazumevaju point-to-point veze izmedju uredjaja, tj. po pravilu nemaju zaista topologiju magistrale. Termin "magistrala" se i ovde koristi po inerciji, iako nije bas sasvim adekvatan.

Serijske magistrale se u velikoj meri razlikuju od paralelnih magistrala, s obzirom da se kod njih sve informacije (komande, adrese

i podaci) salju preko jedne komunikacione linije, bit po bit. U ovom odeljku se ukratko upoznajemo sa nekim aspektima serijske komunikacije.

-) ELEKTRICNE KARAKTERISTIKE SERIJSKIH SIGNALA

Uobicajeno je da se logicka vrednost 1 predstavlja pozitivnim naponom (npr. +5V), a da se logicka vrednost 0 predstavlja naponom 0V. Ovaj napon se racuna relativno u odnosu na masu (koja se obicno oznacava sa GND, od ground). Stoga je za prenosenje bitova preko serijske magistrale, teorijski, potrebna samo jedna zica preko koje se prenosi odgovarajuci potencijal [pod pretpostavkom da su oba subjekta koja komuniciraju povezani na istu masu; u slucaju udaljenih uredjaja to obicno nije slucaj, pa zato postoji i druga zica koja predstavlja GND, kako bi oba uredjaja racunala potencijal u odnosu na istu masu]. Medjutim, ovakav nacin prenosjenja signala je dosta podlozan smetnjama, s obzirom da očitavanje ispravne vrednosti zavisi od potencijala jedne zice koji se moze promeniti usled elektromagnetnih uticaja okoline. Da bi se uticaj smetnji smanjio, cesto se koristi drugacija tehnika prenosa: umesto jedne zice za prenos se koristi par zica (oznacimo ih sa D+ i D-). Logicka jedinica se kodira tako sto se na D+ dovede pozitivan napon (npr. +5V), a na D- se dovede negativan napon (npr. -5V). Logicka nula se kodira obrnuto: na D+ se dovede -5V a na D- se dovede +5V. Dakle, vrednosti se kodiraju odgovarajucom razlikom potencijala izmedju dve zice (koja je pozitivna za logicku jedinicu a negativna za logicku nulu), a ne potencijalom jedne zice u odnosu na masu. Ova tehnika je poznata i kao DIFERENCIJALNO SIGNALIZIRANJE. Ova tehnika je siroko prihvacena kod serijskih magistrala (koriste je sve aktuelne serijske magistrale). Sa druge strane, realizacija ovakvog pristupa kod paralelnih magistrala bi bila preskupa, jer bismo time duplirali i onako veliki broj zica koje cine magistralu. Koriscenje ove tehnike kod serijskih magistrala omogucava pouzdan prenos na veoma visokim frekvencijama.

-) DVOSMERNA I JEDNOSMERNA KOMUNIKACIJA

Za razliku od paralelnih magistrala kod kojih nije moguće istovremeno slati i primati podatke (kazemo i da su jednosmerne, ili half-duplex), kod serijskih magistrala je izvodljivo obezbediti i dvosmernu komunikaciju (full-duplex). Ovo se ostvaruje tako sto umesto jednog para zica imamo dva para zica, po jedan par za svaki smer komunikacije. Ovako nesto znacajno povecava brzinu protoka, s obzirom da vecina protokola magistrale podrazumeva da subjekti razmenjuju poruke prilikom komunikacije koje idu u oba smera. Vecina savremenih brzih serijskih magistrala podrzavaju dvosmernu komunikaciju (PCI-EXPRESS, QPI, DMI). Kod paralelnih magistrala ovako nesto bi bilo preskupo, jer bi dupliralo i onako preveliki broj linija.

-) KODIRANJE PODATAKA KOD SERIJSKIH MAGISTRALA

Serijske magistrale najcesce podrazumevaju sinhroni prenos. Ovo znaci da postoji casovnik u odnosu na koji se sinhronizuje slanje bitova preko serijske linije (npr. na svakoj uzlaznoj ivici casovnika po jedan bit). Ukoliko su subjekti koji komuniciraju povezani na isti casovnik, tada je sinhronizaciju magistrale relativno jednostavno izvesti. U slucaju da imamo dva udaljena uredjaja koji komuniciraju putem serijske magistrale (npr. kada racunar i stampac komuniciraju putem USB interfejsa), tada oni nisu povezani na isti casovnik, pa je potrebno obezbediti sinhronizaciju njihovih casovnika. Cak i kada su

subjekti koji komuniciraju povezani na isti casovnik, pri veoma visokim frekvencijama je veoma tesko obezbediti da signal casovnika dolazi istovremeno do oba uredjaja (pojava poznata i kao ISKRIVLJENOST CASOVNIKA, engl. clock skew), pa je ponovo potrebno obezbediti sinhronizaciju signala casovnika subjekata koji komuniciraju. Ovo se obezbedjuje tako sto se sam niz bitova koji se prenosi kodira tako da u sebi sadrzi informacije koje omogucavaju sinhronizaciju casovnika. U osnovi, postoje dva nacina kodiranja niza bitova koji se prenosi:

-) povratak na nulu (return-to-zero, RZ): podsetimo se da se logicka jedinica kodira pozitivnim naponom izmedju para zica, a logicka jedinica negativnim naponom. Kod RZ kodiranja, nakon svakog prenetog bita se napon izmedju zica na neko vreme vraca na nulu pre nego sto se zapocne slanje sledeceg bita. Ovi periodi "odmora" omogucavaju sinhronizaciju, jer prijemni uredjaj moze jasno da vidi u kom trenutku zapocinje prenos bita, a u kom se zavrшава. Medjutim, ovakav nacin kodiranja produzava duzinu ciklusa i samim tim ogranicava frekvenciju prenosa.

-) bez povratka na nulu (non-return-to-zero NRZ): izmedju prenosa bitova nema "odmora", tj. napon se ne vraca na nulu, vec se odmah zapocinje sa prenosom sledeceg bita. Ovim se omogucavaju vece frekvencije i samim tim vece brzine prenosa. Medjutim, problem nastaje kod sinhronizacije: ukoliko imamo dugacak niz istih bitova koje prenosimo, nece biti nikakve promene napona u duzem vremenskom periodu, pa prijemni uredjaj moze da izgubi sinhronizaciju sa casovnikom predajnika.

Opisani problem kod NRZ kodiranja se resava tako sto se ulazni niz bitova modifikuje tako da se obezbedi da se promene vrednosti signala desavaju dovoljno cesto, kako bi prijemni uredjaj mogao da "uhvati ritam" casovnika predajnika. Neki od nacina da se to postigne su:

-) NRZI kodiranje (non-return-to-zero-inverted): umesto da se logicka nula kodira negativnim naponom, a logicka jedinica pozitivnim naponom, primenjujemo drugaciji pristup: logicku jedinicu kodiramo tako sto zadržavamo isti naponski nivo signala, a logicku nulu kodiramo tako sto invertujemo naponski nivo signala. Na primer:

ORIGINALNI NIZ BITOVA:	1001101101110001111100001100000001
KODIRANI NIZ BITOVA:	0100011100001011111101011101010100

Prvi niz bitova je originalni sadržaj koji zelimo da prenesemo preko komunikacionog kanala. Umesto da prenosimo njega, mi prenosimo drugi niz bitova koji nastaje NRZI kodiranjem. Kodirani niz se dobija tako sto, pocevsi npr. od 0, za svaku jedinicu u originalnom nizu zadržavamo istu vrednost u kodiranom nizu, a za svaku nulu u originalnom nizu invertujemo vrednost bita u kodiranom nizu. Na ovaj nacin se obezbedjuje da cak i prilikom prenosjenja dugackih nizova nula (sto se cesto desava) mi zapravo imamo promenu vrednosti na komunikacionom kanalu pri svakom bitu, pa ce prijemni uredjaj moci da se sinhronizuje. Ostaje jos problem dugackih nizova jedinica (tada ce se u dugom periodu zadržavati isti naponski nivo na komunikacionoj liniji, pa prijemni uredjaj moze izgubiti sinhronizaciju). Ovaj problem se obicno resava tako sto se nakon svakog niza od nekoliko uzastopnih jedinica (tipicno 5 ili 6) u originalnom nizu dodaje jedna nula, cime se obezbedjuje da ce izlazni signal imati promenu u tom trenutku. Ovakav pristup koristi npr. USB 2.0.

-) 8b/10b kodiranje: pored problema sa sinhronizacijom, dodatni problem kod serijskog prenosa moze biti ukoliko broj jedinica i nula koje se prenose nije ujednacen (npr. uglavnom prenosimo nule i tek po neku jedinicu). Ovakvi signali nemaju dobre elektricne karakteristike na mestu prijema, pa se zato pokusava ostvariti tzv. DC-balans (tj. situacija u kojoj je broj prenetih nula i jedinica priblizno jednak). Ovo je narocito znacajno pri visokim frekvencijama prenosa. Kodiranje 8b/10b pokusava da istovremeno resi i problem sa sinhronizacijom, kao i problem DC-balansa. Ideja je da se svaka sekvenca od 8 uzastopnih bitova (tj. jedan bajt) u originalnom nizu bitova kodira kao neka 10-bitna rec (tj. niz od 8 bitova se zamenjuje nizom od 10 bitova). Pritom, sam kod se bira tako da zadovoljava postavljene zahteve (tj. da kodirani niz bitova ima dovoljno ceste promene vrednosti bita, kao i da broj nula i jedinica u kodiranom nizu bitova bude priblizno jednak). S obzirom da 10-bitnih reci ima 4 puta vise od 8-bitnih reci, ovakav kod je moguce pronaci. Kodiranje ne mora biti jednoznacno, tj. jednom te istom bajtu mogu se po potrebi dodeljivati vise razlicitih 10-bitnih kodova, kako bi se odrzao DC-balans. Za detalje kodiranja citalac moze konsultovati odgovarajucu literaturu. Ovaj nacin kodiranja koriste npr. PCI-E 2.0, USB 3.0, SATA 2.0 i 3.0.

-) 64b/66b, 128b/132b, 128b/130b: problem sa 8b/10b kodiranjem je u velikom broju dodatnih bitova (na svakih 8 bitova dodajemo jos 2 dodatna bita, sto je 25%). Kako bi se povecao protok korisnih podataka, primenjuju se druga kodiranja sa slicnom idejom, ali manjim procentom dodatnih bitova (npr. u 64b/66b kodiranju se na svakih 64 bita dodaju jos dva dodatna bita, sto je 3.12%). Ova kodiranja ne garantuju DC-balans i dobru sinhronizaciju, ali su tako osmisljena da statisticki gledano (dakle, sa velikom verovatnocom) daju odlicne rezultate po oba pitanja. Ovakve nacine kodiranja koriste npr. PCI-E 3.0, USB 3.1, SATA 3.2.