

# Fizička razina

---

## 1. Osnovna svojstva fizičke razine

Fizička razina je prvi i najniži sloj računalne mreže i odnosi se na fizičku vezu između dva susjedna uređaja. Zadatak fizičke razine je slanje i primanje podataka kao signala koji se prenose komunikacijskim medijem koji povezuje dva susjedna uređaja. To znači da prilikom slanja podataka fizička razina dobije od podatkovne razine bitove i bajtove koje fizička razina pretvori u signal i šalje kanalom, dok u prijemu fizička razina detektira signal na mediju i pretvara ga u primljene bitove i bajtove koje onda isporuči podatkovnoj razini prijemnika. Kako bi ovo bilo moguće, standardi fizičke razine propisuju:

- mehaničke karakteristike fizičke komunikacije
- električne karakteristike fizičke komunikacije
- funkcionalne karakteristike fizičke komunikacije

Mehaničke karakteristike se odnose na izgled konektora na uređaju i kanalu, raspored kontakata, karakteristike kanala koji se koristi (npr. maksimalna duljina bakrenog vodiča u žičanoj mreži, broj zavrtaja po metru UTP kabela, minimalni radijus prilikom savijanja kabela).

Električne karakteristike se odnose na način stvaranja galvanske veze između prijemnika i predajnika, napon koji predajnik mora biti u stanju generirati te napon koji prijemnik mora biti u stanju izdržati, raspon napona u kojemu signal ne smije biti (prijelazno područje) te maksimalno dopušteno trajanje prolaska signala kroz prijelazno područje i sl.

Funkcionalne karakteristike se odnose na ulogu pojedinih signala u fizičkoj komunikaciji. Predajnik i prijemnik su obično povezani s nekoliko žica (npr. u UTP Cat5 kabeu imaju 4 parice, tj. 8 žica), pa standard fizičke razine definira koje se žice koriste za prijenos podataka, a koje za upravljanje i nadzor prijenosa te kako se to radi.

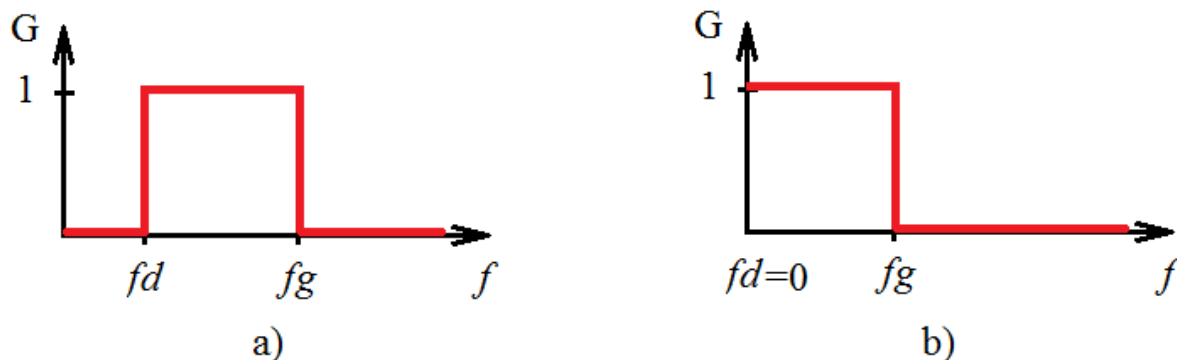
## 2. Informacijski volumen

Informacijski volumen je količina informacije (izraženo u bitovima) koju možemo prenijeti u vremenu  $T$  preko  $K$  istovjetnih paralelnih komunikacijskih sustava, a iznosi  $V=2BDTK$ .

U ovoj formuli  $B$  označava **širinu frekvencijskog pojasa** koji se može koristiti za signalizaciju, a  $2B$  označava broj simbola koji se mogu prenijeti u sekundi. Naime, komunikacijski kanali i primopredajni uređaji su takvi da se njima mogu prenositi, primati i slati samo valovi koji pripadaju nekom određenom rasponu frekvencija.

Na donjoj slici (a) prikazana je općenita frekvencijska karakteristika nekog komunikacijskog sustava, pri čemu je na osi  $y$  označeno pojačanje ( $G=0$  znači da je signal potpuna zagušen,  $G=1$  znači da je nepromijenjen), a na osi  $x$  frekvencija. Sustav s frekvencijskom karakteristikom sa slike ne može prenositi valove čija je frekvencija manja od  $f_d$  (donja granična frekvencija) ili veća od  $f_g$  (gornja granična frekvencija). Širina frekvencijskog pojasa je  $B = f_g - f_d$ .

U žičanim komunikacijama vrlo često je slučaj da je  $f_d = 0$ , tj. da je sustav niskopropusni pa može prenositi sve frekvencije od 0 do neke gornje granične frekvencije (slika b).



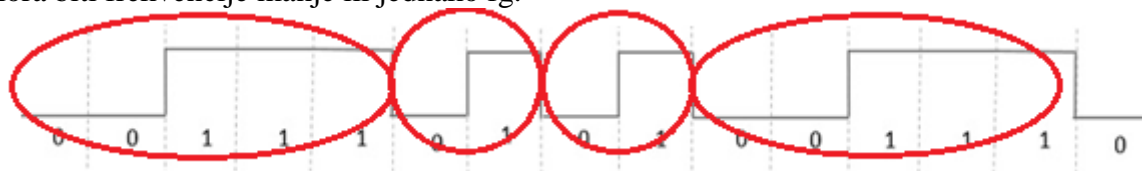
Slika 1: Frekvencijska karakteristika komunikacijskog sustava a) općeniti b) niskopropusni

Pretpostavimo da preko nekog niskopropusnog sustava s gornjom graničnom frekvencijom  $f_g$  želimo prenijeti sljedeći signal:



Slika 2: Primjer nekog signala

U primjeru se podaci prenose kao niz pravokutnih valova određenih frekvencija (niže frekvencije na početku i na kraju slike, više frekvencije u sredini). Da bi se signal mogao uspješno prenijeti preko našeg sustava, on ne smije sadržavati valove frekvencije veće od  $f_g$ . Dakle, u ovom signalu onaj centralni dio u kojemu imamo brze izmjene iz 0 u 1 i obratno mora biti frekvencije manje ili jednako  $f_g$ .



Slika 3: Pravokutni valovi u signalu na slici

Kako bismo imali maksimalnu brzinu prijenosa podataka, tako ćemo slati bitove tako da uzastopna izmjena nula i jedinica formira val frekvencije jednake upravo  $f_g$ . Kako u jednome takvom valu mi prenesemo dva simbola (jedan poluperiod predstavlja nulu, a drugi jedinicu), to znači da možemo prenijeti  $2 \cdot f_g$  simbola u sekundi.

Za sustav kod kojega je  $f_d > 0$  mogli bismo na sličan način doći do zaključka da možemo prenijeti maksimalno  $2 \cdot (f_g - f_d) = 2B$  simbola po sekundi.

D iz formule za informacijski volumen ( $2BDTK$ ) označava **dinamiku**, tj. koliko bitova informacije nosi svaki preneseni simbol.

U gornjem primjeru postojala su samo dva različita simbola, pa je jedan od njih simbolizirao nulu, a drugi jedinicu, tj. svaki simbol je nosio jedan bit informacije. Ako bismo imali sustav u kojemu možemo prenositi 4 različita simbola, onda bi svaki simbol prenosio 2 bita informacije.

Na primjer, neka koristimo takav sustav u kojem predajnik može prijemu poslati 4 slova: A, B, C i D. Tada bi se moglo specificirati da svaki simbol predstavlja 2 bita i to:

A – 00

B – 01

C – 10

D – 11

Ako pošiljatelj želi poslati ovaj niz bitova 110110110000, on bi poslao DBCDAA, i za slanje 12 bitova informacije bilo bi mu dovoljno samo 6 simbola. To je zato jer svaki simbol predstavlja 2 bita, tj. dinamika je 2. Općenito, ako se sustavom može razmjenjivati R različitih simbola, dinamika je  $D = \lg(R)$ , gdje je  $\lg$  oznaka za logaritam po bazi 2.

Dakle, 2B nam govori koliko simbola možemo prenijeti po sekundi, a D govori koliko bitova informacije nosi svaki simbol, pa umnožak 2BD govori koliko bitova informacije možemo prenijeti po sekundi i naziva se **kapacitet**. Ovo je ona veličina o kojoj se priča kada se kaže da je „mreža brza 100Mbps“, misli se na kapacitet mreže. Na kapacitet potpuno jednako utječe povećanje (ili smanjenje) pojasne širine npr. za 3 puta kao i povećanje broja simbola 8 puta (jer se tako dinamika poveća 3 puta). U oba slučaja kapacitet mreže bi porastao 3 puta. Ako bi se moglo povećati i jedno i drugo, kapacitet bi narastao 9 puta.

T iz formule 2BDTK je vrijeme koliko se prenose podaci, a K je broj **paralelnih sustava**. Naime, bitovi se mogu prenijeti serijski preko jednog sustava brzinom X ili preko 3 paralelna sustava, brzinom X/3 po svakome. Očito bi u slučaju paralelnog prijenosa mogli imati 3 puta manji kapacitet pojedinog kanala (npr. 3 puta manju frekvencijsku širinu kanala) u odnosu na prijenos preko jednog kanala, a da u vremenu T prenesemo jednaku količinu informacija. Tada bi kod paralelnog prijenosa svaki simbol na kanalu trajao 3 puta duže nego kod prijenosa jednim kanalom. Ako je brzina kanala (jednog serijskog i pojedinačnih paralelnih) jednaka, onda bi paralelnim prijenosom preko 3 kanala mogli prenijeti 3 puta veću količinu informacija.

Međutim, paralelni prijenos ima svojih nedostataka. Naime, kod paralelnog prijenosa podataka preko više kanala potrebno je osigurati da simboli stižu na odredište u istom ili približno istom trenutku, kako bi prijemu mogao posložiti primljene simbole ispravnim redoslijedom. Kako kanali nikada nisu savršeno iste duljine, uvijek će kašnjenje signala na paralelnim kanalima biti različito, što prijemu otežava sinkronizaciju primljenih podataka. Ovo je posebno izraženo kod prijenosa podataka na velike udaljenosti, kod kojega će i razlike u kašnjenju po kanalima biti veće. Stoga se može dogoditi da signali po jednom kanalu kasne toliko da neki bit stigne do prijema tek kada je po ostalim kanalima već stigla sljedeća grupa bitova, zbog čega bi prijemu primljene bitove pogrešno posložio te prijenos ne bi bio uspješan.

Može se zaključiti da se prijenosom podataka paralelnim kanalima može povećati ukupna brzina prijenosa, ali je potrebno riješiti probleme sa sinkronizacijom podataka po kanalima, što na velikim udaljenostima može biti i nemoguće.

### 3. Sinkronizacija

Jedan od zadataka fizičke razine je obaviti sinkronizaciju po bitu ili po bitu i po oktetu (tj. po riječi). Kod sinkronizacije po bitu zadatak predajnika i prijema je odrediti u kojem trenutku trebaju poslati (za predajnik) tj. detektirati (za prijemu) novi bit podataka na liniji. Kod sinkronizacije po riječi njihov je zadatak odrediti u kojem trenutku na liniji počinje, a u kojem završava jedan blok podataka koji se naziva riječ. Riječ se vrlo često sastoji od 8 bitova

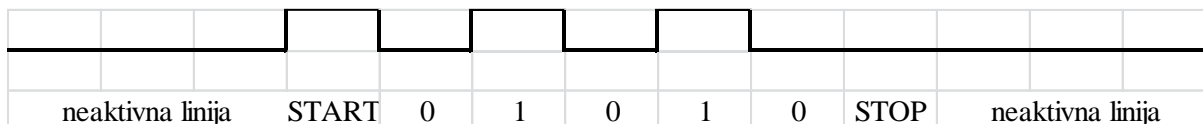
Po tome kako se obavlja sinkronizacija na fizičkoj razini razlikujemo asinkronu i sinkronu komunikaciju.

Kod asinkrone komunikacije predajnik i prijemnik razmjenjuju podatke bez zajedničkog taktnog signala koji bi im služio za sinkronizaciju. Brzina komunikacije (u bitovima po sekundi) se unaprijed dogovori i ona definira koliko jedan bit traje na liniji. Kod nekih malo naprednijih oblika asinkrone komunikacije brzina signalizacije ne mora biti unaprijed dogovorena, već prijemnik može iz primljenog signala sam zaključiti koliko traje jedan bit. Kako bi prijemnik mogao odrediti gdje mu počinje, a gdje završava primljena riječ, pošiljalac prije svake riječi pošalje START bit, a nakon svake riječi pošalje STOP bit. Start bit ima vrijednost suprotnu od neaktivne linije (napon na liniji dok se ne šalju podaci), dok STOP bit ima vrijednost jednaku neaktivnoj liniji.

neaktivna linija			0	1	0	1	0	neaktivna linija	

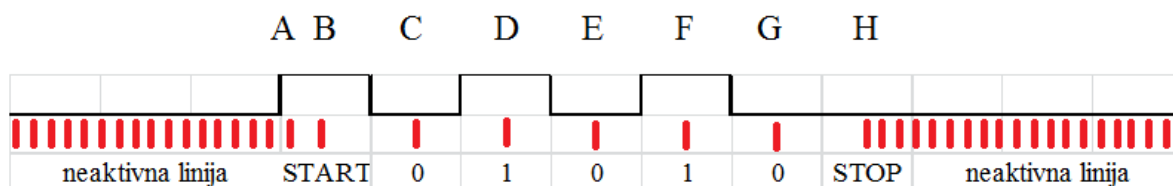
neaktivna linija				1	0	1	0	0	neaktivna linija		
neaktivna linija				0	0	1	0	1	neaktivna linija		

Da bi se ovo izbjeglo, koriste se START (suprotan od neaktivne linije) i STOP bitovi (jednak neaktivnoj liniji) na početku i kraju riječi.

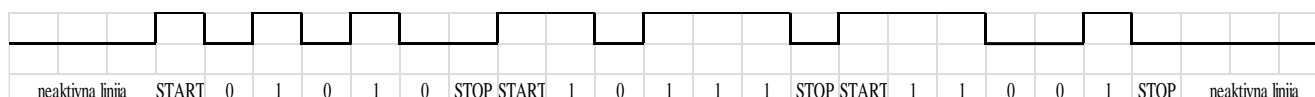


Uz korištenje START i STOP bitova i poznatu brzinu signalizacije (2bps) i veličinu riječi (5 bitova) prijemnik sada može uspješno detektirati koju riječ je primio:

- dok je linija neaktivna prijemnik učestalo ispituje liniju da vidi je li došlo do promjene (na donjoj slici su crvenom bojom označeni trenuci u kojima prijemnik ispituje stanje na liniji)
- kada prijemnik detektira da je linija prešla iz neaktivnog stanja u aktivno zaključuje da je to početak START bita (trenutak A)
- prijemnik zna da svaki bit traje 0,5s pa nakon 0,25s opet provjeri liniju jer zna da je to sredina START bita (B)
- prijemnik zna da će nakon 0,5s na liniji biti sredina prvog bita riječi, pa opet nakon 0,5s sredina drugog bita riječi itd, pa u trenucima C-G primi pet bitova podataka, tj. jednu riječ
- prijemnik nakon 0,5s očekuje sredinu STOP bita (trenutak H), nakon čega opet počne učestalo ispitivati liniju čekajući neki novi START bit



Ako bi predajnik želio poslati nekoliko uzastopnih riječi (npr. 01010, 10111, 11001), signal na liniji bi izgledao ovako:



Možemo se zapitati jesu li STOP i START bitovi zaista potrebni između uzastopnih riječi, ili bi bilo dovoljno nakon prvog START bita samo poslati bitove podataka jedan za drugim. Prijemnik bi onda brojao tako primljene bitove i slagao ih 5 po 5 u riječi.

Problem kod ovoga bi bila situacija u kojoj predajnik želi poslati npr. 3000 uzastopnih riječi 11111. Kako prijemnik i predajnik nemaju neki zajednički takt prema kojem bi se sinkronizirali, moguće je da bitovi koje predajnik šalje traju malo duže ili kraće od 0,5s ili da prijemnik ispituje liniju malo češće ili rjeđe od jednom svakih pola sekunde. U tom slučaju, po primitku 3000 riječi 11111 prijemniku bi bilo poprilično teško odrediti da je primio baš 15000 jedinica, a ne 14999 ili 50001 jedinicu. Stoga STOP i START bitovi između svake riječi služe za sinkronizaciju predajnika i prijemnika na početku svake nove riječi, a riječi su dovoljno kratke da se do kraja riječi ta sinkronizacija neće izgubiti.

Dobra strana asinkrone komunikacije je što nam je za slanje podataka s predajnika na prijemnik dovoljna samo jedna linija. Nedostaci su to što se brzina signalizacije mora unaprijed dogovoriti i to što se osim samih podataka linijom prenose i START i STOP bitovi. Ako bi signalizacija bila 100 bps, a predajnik šalje prijemniku 5-bitovne riječi, onda bi se prijenos podataka odvijao brzinom  $5/7 \cdot 100 \text{ bps} = 71 \text{ bps}$ , dok bi se 29 bps trošilo na prijenos

START i STOP bitova. Naime u svakih 7 prenesenih bitova mi ćemo imati samo 5 bitova podataka.

### 3.2. Sinkrona komunikacija

Druga opcija na fizičkoj razini je korištenje sinkrone komunikacije. U njoj su predajnik i prijemnik sinkronizirani zajedničkim taktom. Postoje različite metode za sinkronizaciju predajnika i prijemnika, a najjednostavnija opcija je kada su osim podatkovnom linijom (kojom se šalju podaci) oni povezani i dodatnom taktom linijom kojom se prenosi takt koji određuje kada se na liniji nalazi novi bit podataka.

Ako predajnik generira ovaj takt i želi poslati prijemniku niz bitova 01010 signal na linijama bi izgledao ovako:



Dakle, svaki put kada predajnik na podatkovnu liniju stavi novi bit podataka, malo nakon toga (tj. u sredini trajanja bita) predajnik na taktnoj liniji generira impuls kojim govori prijemniku da treba primiti novi bit. Prijemnik cijelo vrijeme ispituje taktnu liniju i kada na njoj detektira impuls, tek tada ispita koji bit je primio na podatkovnoj liniji.

Nedostatak ovakve komunikacije je u tome što je u predajnik i prijemnik potrebno povezati dodatnom linijom (za taktni signal), a prednost je što se podatkovna linija može koristiti samo za prijenos podataka (bez START i STOP bitova) te što se brzina slanja podataka može lako mijenjati, jednostavnim ubrzavanjem ili usporavanjem takta na taktnoj liniji.

Postoje i druge metode sinkronizacije predajnika i prijemnika, kod kojih nije potrebna dodatna taktna linija. Tako predajnik može slati unaprijed dogovoreni niz sinkronizacijski bitova podatkovnom linijom dok je ona neaktivna, npr. 000101100001011000010110. Prema detektiranom signalu prijemnik može odrediti koliko točno traje jedan bit. Kada predajnik počne slati podatke (bilo što osim unaprijed dogovorenog sinkronizacijskog niza), prijemnik će biti sinkroniziran s predajnikom i znati će svakih koliko će primiti novi bit podataka. Ako je niz podataka kojeg predajnik šalje predug, onda opet može doći do toga da predajnik i prijemnik izgube sinkronizaciju pa da prijemnik ne detektira neki bit ili neki bit primi dvaput. Stoga se prilikom slanja dugih nizova podataka mora s vremena na vrijeme obaviti re-sinkronizacija predajnika i prijemnika, slanjem sinkronizacijskog niza.

U modernim mrežama nerijetko je slučaj da se u sinkronoj komunikaciji taktni signal šalje istom linijom skupa s podacima, kao npr. u Ethernet lokalnoj mreži. U tom slučaju kažemo da se takt utisne u podatke. I u ovom slučaju nam je onda dovoljna jedna linija za komunikaciju, ali je nedostatak to što se dio kapaciteta koristi za prijenos takta (a ne samo podataka).

Zaključno, kod asinkronog prijenosa sinkronizacija predajnika i prijemnika se obavlja korištenjem START i STOP bitova, čime se obavi i sinkronizacija po riječi, dok se sinkronizacija po bitu temelji (uglavnom) na unaprijed dogovorenoj brzini signalizacije.

S druge strane, kod sinkronog prijenosa se sinkronizacija po bitu (uglavnom) obavlja korištenjem taktnog signala koji se prenosi uz podatke (bilo posebnom linijom ili utisnut u

podatke), a sinkronizaciju po riječi (tj. po bajtu) je potrebno obaviti nekim dodatnim tehnikama, često na podatkovnoj razini.

#### 4. Linijski kodovi

























































































Linijski kod je oblik napona, struje ili fotona koji se koristi za predstavljanje digitalnih podataka na komunikacijskom kanalu. Najjednostavniji kod je onaj u kojem bit 1 predstavljamo nekim naponom (ili strujom ili nekom trećom veličinom) i taj napon se održava sve dok traje bit 1, a bit 0 se predstavlja drugim naponom i opet se taj napon održava sve dok traje bit 0. Ovakvo kodiranje nula i jedinica naziva se **NRZ (non return to zero)**, jer napon traje za vrijeme cijelog trajanja nekog bita. Ako bi se nakon signaliziranja jedinice (na početku tog bita) napon vratio na nulu do sljedećeg bita, onda bi to bio **RZ (return to zero)** kod.

Kodovi mogu biti **unipolarni ili bipolarni**. Kod unipolarnih kodova donja naponska razina je 0 V, a gornja je neki pozitivan napon (npr. 5 V). Kod bipolarnih kodova gornja naponska razina je pozitivan, a donja negativan napon, najčešće jednak kao i pozitivni (npr. 5 V i -5V).

Kod unipolarnih kodova prosječni napon na liniji je neka pozitivna vrijednost. Na primjer ako je gornja naponska razina 5 V, a donja 0 V, onda će prosječni napon na liniji biti oko 2,5 V. Ovo prijemniku otežava detekciju primljenih bitova. Naime, ovu istosmjernu komponentu signala će na prijemniku blokirati kondenzatori i transformatori, pa će prijemnik za jedinicu detektirati napon od 2,5 V (od prosjeka do 5V), a za nulu negativan napon od -2,5V. Ako prijemnik primi dugi niz jedinica, istosmjerna komponenta signala će narasti s 2,5V prema 5V, pa će prijemnik za jedinicu detektirati puno manji porast napona (npr. ako je prosječni napon u tom dugom nizu porastao na 4,3V onda će za novu jedinicu prijemnik detektirati porast napona od samo 0,7V pa bi mogao pomisliti da je to nula). Zbog ovoga su unipolarni kodovi podložni greškama, posebno prilikom prijema dugih nizova jedinica.

Dobra karakteristika bipolarnih kodova je to što im je prosječni napon na liniji 0 V, tako da nema istosmjerne komponente koju bi blokirali kondenzatori i transformatori prijemnika. Ipak, i kod ovih kodova postoji opasnost da prosječni napon poraste ako predajnik šalje dugi niz jedinica, uslijed čega bi se moglo dogoditi da prijemnik neku jedinicu krivo primi kao nulu. Ipak, ovo je puno rjeđe nego kod unipolarnih kodova.

Primjeri nekih linijskih kodova navedeni su na slici ispod, za niz bitova 11010010

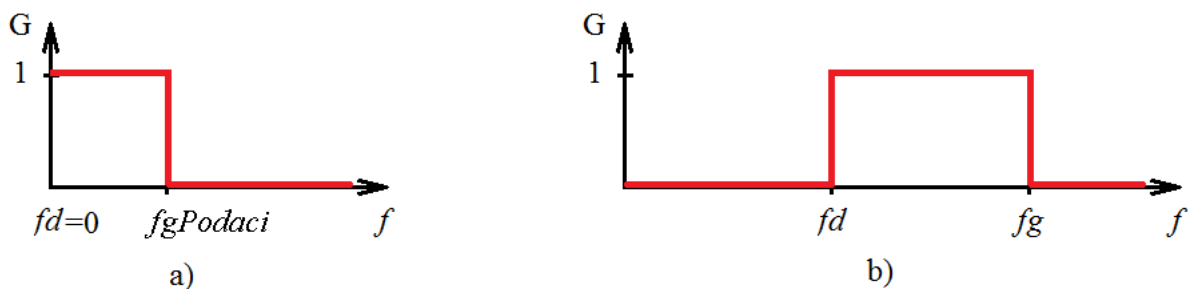
Kod	1	1	0	1	0	0	1	0	Širina pojasa min-max
NRZ-L									DC-R/2
NRZ-M									DC-R/2
NRZ-S									DC-R/2
RZ-M									DC-R
FM-M									R-2R
MFM									R/2-R
BI-L (Manchester 2)									R/2-R
BI-M (Manchester 1)									R/2-R
BI-S									R/2-R
MILLER									R/4-R/2
CMI									R/2-R



	1	0	komentar
NRZ-L	visoka naponska razina	niska naponska razina	non return to zero-level, standardni format signala u pozitivnoj logici. Jednostavan za implementirati, efikasno koristi pojasnu širinu, ima veliku DC komponentu
NRZ-M	promjena napona na početku bita	bez promjene	non return to zero-mark
NRZ-S	bez promjene	promjena napona na početku bita	non return to zero-space
RZ-M	pola perioda visoki napon nakon čega se vraća u niski	ostaje niski napon	return to zero, potreban mu je dvostruko veći pojas od NRZ koda, ali ima dvostruko manju DC komponentu
FM-M	impuls na početku i u sredini bita	impuls na početku bita	koristio se za zapisivanje podataka na diskete
MFM	impuls na sredini bita	bez promjene za prvu nulu u nizu, impuls na početku druge i svih sljedećih uzastopnih nula	koristio se za zapisivanje podataka na diskete i tvrde diskove
BI-L (Manchester 2)	pozitivna promjena napona na sredini bita	negativna promjena napona na sredini bita	dodatna promjena između dva uzastopna ista bita (00 ili 11), koristi se u Ethernetu, takti signal utisnut s podacima, ali potreban mu je dvostruko veći frekvencijski pojas u odnosu na NRZ
BI-M (Manchester 1)	promjena na početku i na sredini bita	promjena na početku bita	
BI-S	promjena na početku bita	promjena na početku i na sredini bita	koristi se u token ring lokalnim mrežama
MILLER	promjena na sredini bita	bez promjene za prvu nulu u nizu, promjena na početku druge i svih sljedećih uzastopnih nula	jako efikasan, ali podložan greškama
CMI	bez promjene za prvu jedinicu u nizu, promjena na početku druge i svih sljedećih uzastopnih jedinica	<b>pozitivna</b> promjena u sredini bita	coded mark inversion, potreban mu je dvostruko veći frekvencijski pojas u odnosu na NRZ, ali u podatke je utisnut i takt

## 5. Modulacija

Kod prijenosa podataka bakrenom žicom karakteristika komunikacijskog sustava je najčešće takva da propušta frekvencije od 0 do neke  $f_g$  (niskopropusni sustav), tako da je potrebno samo paziti da podaci kodirani linijskim kodom koji se koristi u sebi nemaju frekvencije veće od  $f_g$ . Međutim, nerijetko je slučaj da komunikacijski sustav nije niskopropusni, nego pojasnopropusni, tj. da propušta frekvencije od  $f_d$  do  $f_g$ , pri čemu je  $f_d > 0$ . Kako je „prirodno“ frekvencijsko područje kodiranih podataka od 0 Hz do neke  $f_g$  Podaci, komunikacijski sustav neće moći prenijeti podatke u njihovom kodiranom obliku. Jednostavno, predajnik ne može generirati tako niske frekvencije niti ih prijemnik može detektirati. U računalnim mrežama ovo je slučaj kod bežičnog prijenosa podataka, u kojem sustav može prenositi samo valove u jako visokom frekvencijskom području (nekoliko GHz).



Frekvencijska karakteristika a) podataka kodiranih linijskim kodom b) komunikacijskog sustava

Stoga je prije slanja podataka kanalom potrebno prebaciti podatke iz njihovog originalnog frekvencijskog područja u područje koje se može koristiti u komunikacijskom sustavu. Ovo originalno frekvencijsko područje signala se naziva osnovni pojas (engl. baseband). Postupak prebacivanja signala iz osnovnog u pomaknuto frekvencijsko područje naziva se modulacija. Modulirane podatke predajnik može poslati, oni mogu nezagušeni proći kanalom te ih prijemnik može uspješno detektirati i nakon toga demodulirati, tj. vratiti ih iz frekvencijskog područja u koje su bili prebačeni u njihovo originalno područje. Iz demoduliranih podataka se zatim detektiraju simboli (nule i jedinice). Modulaciju obavlja modulator, a demodulaciju demodulator. Uređaj koji obavlja i modulaciju (prilikom slanja) i demodulaciju (prilikom prijema) naziva se modem.

Postoji više tehnika za moduliranje podataka.

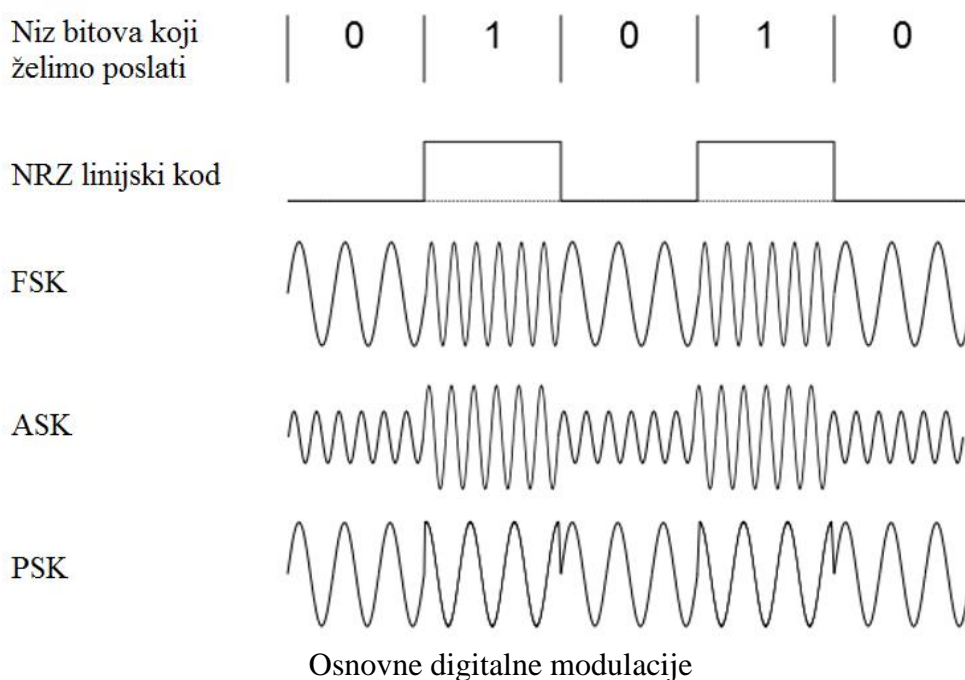
Kod **amplitudne modulacije** različiti simboli se predstavljaju istim valom, ali različitim amplitudama. Frekvencija ovog vala mora biti u rasponu frekvencija koje se mogu prenijeti komunikacijskim sustavom. Ako sustav može generirati i prenijeti samo dvije različite amplitude, onda će jedna amplituda predstavljati nulu, a druga jedinicu. U nekom složenijem sustavu predajnik i prijemnik će možda moći generirati i detektirati 4 različite amplitude pa bi u tom sustavu postojala 4 različita simbola i svaki simbol bi u sebi nosio 2 bita informacije. Varijanta amplitudne modulacije koja se koristi za prijenos digitalnih podataka naziva se amplitude shift keying (ASK).

Kod **frekvencijske modulacije** različiti simboli se predstavljaju valom iste amplitude, ali različitih frekvencija. Ove frekvencije moraju biti u rasponu frekvencija koje se mogu prenijeti komunikacijskim sustavom. Ako sustav može generirati i prenijeti samo dvije različite frekvencije, onda će jedna frekvencija predstavljati nulu, a druga jedinicu. U nekom složenijem sustavu predajnik i prijemnik će možda moći generirati i detektirati 4 različite frekvencije pa bi u tom sustavu postojala 4 različita simbola i svaki simbol bi u sebi nosio 2 bita informacije. Varijanta frekvencijske modulacije koja se koristi za prijenos digitalnih podataka naziva se frequency shift keying (FSK).

Kod **fazne modulacije** različiti simboli se predstavljaju valom iste amplitude i frekvencije, ali s različitim pomakom u fazi. Frekvencija vala mora biti u rasponu frekvencija koje se mogu prenijeti komunikacijskim sustavom. Ako sustav može generirati i prenijeti samo dva različita pomaka u fazi, onda bi se nula mogla predstaviti kao originalni val, bez pomaka u fazi (tj. s pomakom  $0^\circ$ ), a jedinica kao originalni val pomaknut u fazi za  $180^\circ$ . Naravno, i u ovom slučaju ako bi primopredajnici mogli raditi s više od 2 pomaka u fazi onda bi svaki pomak nosio više bitova informacije. Fazna modulacija koja se koristi za prijenos digitalnih podataka naziva se phase shift keying (PSK). Nedostatak PSK je u tome što prijemnik mora primiti val

iste frekvencije i faze kakav je generiran na predajniku, što nije uvijek moguće (npr. ako se neki uređaj pomiče, Dopplerov efekt bi doveo do promjene frekvencije detektiranog vala). Stoga se u faznoj modulaciji često koristi DPSK (diferencijalna PSK) modulacija, u kojoj se različiti simboli kodiraju različitim pomakom u fazi u odnosu na prethodni simbol, a ne na originalni val. Tako se kod DPSK nula može kodirati kao bez promjene faze na početku simbola (val nastavlja normalno nakon prethodnog simbola), a jedinica kao skok u fazi od  $180^\circ$  nakon prethodnog simbola.

U bežičnim računalnim mrežama često se koristi QAM modulacija koja se može shvatiti kao kombinacija amplitudne (ASK) i fazne (PSK) modulacije. U QAM modulaciji modemi mogu generirati i detektirati simbole s različitim i amplitudama i faznim pomacima, tako da svaki simbol ima svoju amplitudu i svoj pomak. Budući da se ovakvom kombinacijom dobije veći broj simbola, onda i svaki simbol u sebi nosi veći broj bitova informacije. Tako se u bežičnim lokalnim mrežama često koristi 64-QAM i 256-QAM modulacija (64 tj. 256 različitih simbola, svaki nosi 6 tj. 8 bitova informacije), a u najnovijima i 1024-QAM (1024 simbola, 10 bitova informacije po simbolu). Povećavanjem broja različitih simbola povećava se brzina prijenosa informacija (zbog porasta dinamike D), ali se povećava i mogućnost greške.



## 6. Osnovni i izvedeni kanali

Ako se kao kanal za prijenos informacija koristi kapacitet fizičkog kanala u potpunosti, to nazivamo osnovni kanal. On podrazumijeva neprekidno korištenje cijelog frekvencijskog pojasa kanala (od  $f_d$  do  $f_g$ ).

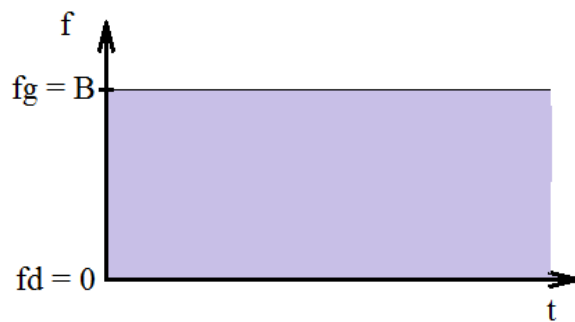
Ukoliko se informacijski volumen kanala podijeli na više korisnika, dobiju se izvedeni kanali. Tri su osnovna načina za kreiranje izvedenih kanala:

- podjela po vremenu (TDM – time domain multiplexing): vrijeme korištenja osnovnog kanala se podijeli na periode te za svaki period odredi kojem izvedenom kanalu taj

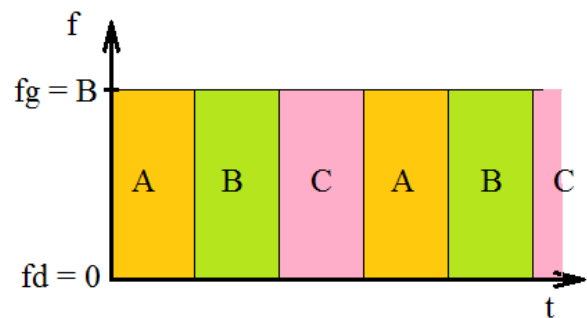
period pripada. Tijekom pripadajućeg perioda izvedeni kanal koristi cijeli kapacitet osnovnog kanala (tj cijeli njegov frekvencijski pojas)

- podjela po frekvenciji (FDM – frequency domain multiplexing): frekvencijski pojas kanala se podijeli na intervale koji se dodijele izvedenim kanalima. Svaki izvedeni kanal može prenositi podatke cijelo vrijeme, ali smije koristiti samo one frekvencije (tj onaj dio pojasa osnovnog kanala) koje su mu dodijeljene podjelom
- kombinirana podjela (TDM+FDM): osnovni kanal se podijeli i po vremenu i po frekvenciji, tako da neki izvedeni kanal može koristiti dodijeljeni dio frekvencijskog pojasa osnovnog kanala samo u dodijeljenom intervalu (tj. intervalima)

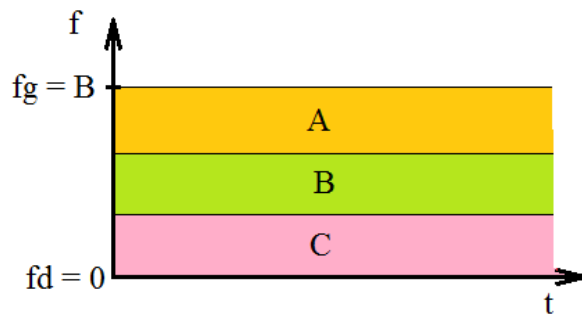
Na donjim slikama prikazana je podjela osnovnog kanala na izvedene kanale. Os x predstavlja vrijeme, a os y predstavlja dostupne frekvencije osnovnog kanala.



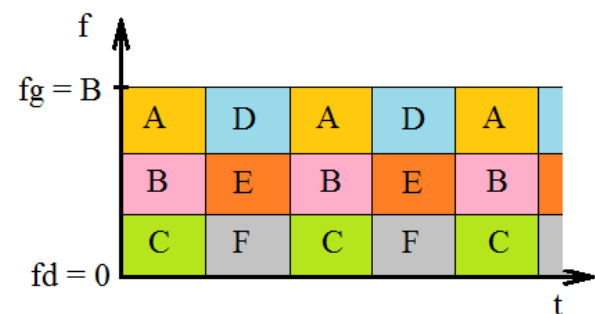
a)



b)



c)

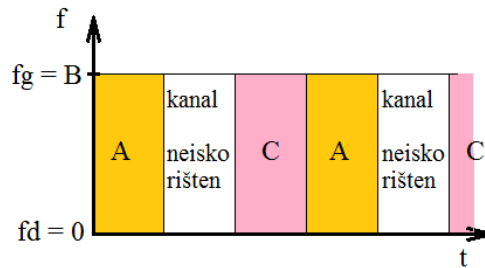


d)

Osnovni i izvedeni kanali: a) osnovni kanal b) TDM, 3 izvedena kanala c) FDM, 3 izvedena kanala d) TDM+FDM, 6 izvedenih kanala

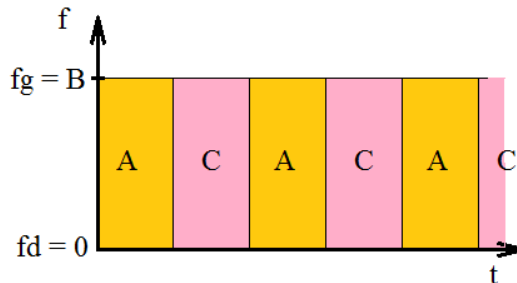
Podjela osnovnog kanala u vremenu na izvedene kanale može biti fiksna ili statistička. Kod fiksne podjele (**fiksno multipleksiranje** u vremenu), neki uređaj A prenosi podatke kanalom dok traje njegov vremenski odsječak, a nakon toga prestaje s radom te sljedeći uređaj (B) počinje slati svoje podatke. Uređaji mogu koristiti isključivo svoje dodijeljene intervale. Ako neki uređaj nema podataka za slanje, kanal tijekom njegovog intervala ostaje neiskorišten.

Na primjer, ako na gornjoj slici b) uređaj B ne bi imao podataka za slanje, a koristi se fiksno multipleksiranje, korištenje kapaciteta osnovnog kanala bi izgledao kao na donjoj slici:



Fiksno multipleksiranje u vremenu, 3 kanala, jedan od njih neaktivan

Očito je da bi u ovom slučaju ukupni kapacitet kanala ostao neiskorišten trećinu vremena tj. koristilo bi se samo 67% dostupnog kapaciteta. Stoga je bolja opcija (ali i malo kompliciranija) korištenje **statističkog multipleksiranja**, kod kojega vremenski odsječci nisu fiksno dodijeljeni, već neki uređaj može koristiti vremenske odsječke drugih uređaja ako su oni neaktivni. U gornjem primjeru, u kojem samo uređaji A i C šalju podatke, dok je uređaj B neaktivan, TDM bi izgledao kao na donjoj slici.



Statističko multipleksiranje u vremenu, 3 kanala, jedan od njih neaktivan

Postoje razne metode kojima se kanal dodjeljuje nekome od zainteresiranih (aktivnih) uređaja, a mogu se podijeliti u dvije grupe:

- centralizirani pristup kanalu: postoji neki uređaj koji upravlja kanalom i određuje koji će od zainteresiranih uređaja u kojem trenutku dobiti pravo korištenja kanala
- distribuirani pristup kanalu: zainteresirani uređaji se međusobno ravnopravno nadmeću za korištenje kanala u slobodnom vremenskom odsječku, nema neke centralne stanice koja upravlja pristupom kanalu