

PROFILUL TEHNIC SPECILALIZAREA:
TEHNICIAN OPERATOR TEHNICĂ DE CALCUL

TRANSMISIA – RECEPȚIA SINCRONĂ

2008

CUPRINS

Cap. I	Argument.....	3
Cap. II	Transferul datelor digitale	4
Cap. II. 1.	Noțiuni de bază	4
Cap. II. 2.	Moduri de transfer pentru semnale digitale	5
Cap. II. 3.	Transmisia – recepția sincronă	6
Cap. III.	Controlul transmisiei sincrone	7
Cap. III. 1.	Circuite pentru controlul transmisiei	7
Cap. III. 2.	Sincronizarea datelor	11
Cap. III. 3.	Detectarea erorilor	13

Cap. I ARGUMENT

În această epocă a calculului distribuit, rețelele sunt prezente în aproape toate mediile de lucru. O *rețea* este un mecanism care permite calculatoarelor distincte și utilizatorilor acestora să comunice și să partajeze resurse.

Rețelele au fost inițial soluții de conectivitate brevetate, care erau parte integrată a unui pachet de soluții informatice, în aceeași măsură brevetat. Companiile care automatizau procesarea de date sau funcțiile de contabilitate în epoca de dinaintea calculatoarelor personale erau obligate să se adreseze unui singur comerciant pentru a obține o soluție la cheie.

În aceste medii, aplicațiile software erau executate doar pe un calculator cu un unic sistem de operare. Sistemul de operare putea fi executat numai pe produse hardware ale aceluiași distribuitor. Chiar și echipamentele terminale și conexiunile la calculator făceau parte din aceeași soluție integrată a unui singur producător.

În timpul domniei soluțiilor integrate ale unui singur producător, au apărut două direcții de dezvoltare tehnologică, ce au schimbat cursul viitor al informaticii. În primul rând, au început să apară strămoșii PC-urilor de astăzi. Aceste dispozitive erau inovatoare prin aceea că plasau puterea de calcul chiar pe birou.

În al doilea rând, oamenii de știință de la Xerox Palo Alto Research Center (PARC) au început să caute modalități de îmbunătățire a productivității proprii. Au căutat în special un mijloc de îmbunătățire a partajării datelor și fișierelor între stațiile de lucru inteligente pe care le aveau.

Metoda existentă, de partajare a dischetelor, era problematică și consuma timp. Soluția lor a fost o rețea locală (LAN), pe care au numit-o *ethernet*. Aceasta era o rețea LAN rudimentară care se baza, pentru o mare parte a definirii și comportării sale, pe protocoale de nivel superior pentru inter-rețele. Potențialul comercial al acestei tehnologii a devenit imediat evident.

Împreună, dispozitivele inteligente ale utilizatorilor și rețelele locale vor da naștere unui nou model: prelucrarea deschisă, distribuită, în rețea a datelor.

Cap. II TRANSFERUL DATELOR DIGITALE

Cap. II. 1. Noțiuni de bază.

Definiție: Prin comunicație de date se înțelege schimbul de informație numerică codificată între două DTE.

Trebuie făcută distincția între termenii "dată" și "informație".

Termenul "dată" este folosit pentru a desemna un set sau un bloc de caractere numerice sau alfabetic codificate ce sunt schimbate între două echipamente. În cadrul comunicației de date în afara transferului acestui tip de date este de asemenea necesar ca cele două echipamente să schimbe și diverse mesaje de control (de exemplu pentru a preveni sau corecta erorile de transfer). De aceea termenul de informație este folosit cu un înțeles mai larg desemnând atât date cât și mesaje de control.

Comunicația de date se ocupă nu numai cu modul de transmisie a datelor printr-un mediu de transmisie fizic ci și cu tehnicile ce trebuie folosite pentru detectarea și corectarea erorilor de transmisie, cu controlul ratei de transfer a datelor și stabilirea formatului datelor ce trebuie transferate.

Din punct de vedere al numărului de linii ce interconectează două echipamente se deosebesc două tipuri de conexiuni:

1. modul de transfer paralel presupune folosirea câte unui fir pentru fiecare bit de date (al unui cuvânt). Aceasta înseamnă că mai multe fire sunt folosite pentru interconectarea a două DTE. Din acest motiv modul de transfer paralel nu se folosește decât în cazul în care distanța între DTE este mică.
2. modul de transfer serial presupune folosirea unei singure perechi de fire pentru interconectarea echipamentelor.

La un moment dat pe linie se transmite un singur bit, pentru fiecare bit fiind alocat un interval de timp fix. Viteza de transfer este mai mică decât în cazul 1 dar distanța între DTE poate fi mult mai mare.

Cele două moduri de operare sunt reprezentate în figura 3.1.

Comunicația de date între două echipamente se poate realiza în trei moduri:

1. simplex: presupune transmisia datelor într-o singură direcție.
2. half-duplex: presupune transferul de date alternativ între cele două echipamente. Când unul din echipamente se află în starea de emisie celălalt se află în recepție.
3. duplex (full-duplex): presupune schimbul de date în ambele direcții simultan.

Cap. II. 2 Moduri de transfer pentru semnalele digitale

Semnalele digitale pot fi tratate în sistemele de comunicații și de transmisie în mod sincron (STM - Synchronous Transfer Mode) sau în mod asincron (ATM – Asynchronous Transfer Mode).

STM – mod de transfer sincron (Synchronous Transfer Mode)

Sistemele cu mod de transfer sincron (STM) utilizează transmisii bazate pe ierarhii digitale plesiocrone (PDH – Plesiochronous Digital Hierarchy) sau ierarhii digitale sincrone (SDH – Synchronous Digital Hierarchy). Capacitatea de transmisie accesibilă pe link este divizată în cadre în PDH sau în containere digitale (CV) în SDH.

PDH utilizează semnale digitale care sunt organizate în cadre periodice cu perioada fixă de 125 μ s. În multiplexul de ordinal 1 (primar) fiecare cadru este divizat în 32 de intervale de timp (IT) egale ca durată, fiecare IT permițând transmiterea unui octet de informație. Transmiterea cadrelor este sincronă. Sincronizarea este asigurată prin ITO, care marchează de asemenea, și începutul cadrului. În interiorul cadrului, identificarea unui IT este realizată prin poziția acestuia în raport cu ITO.

Pentru o conexiune se realizează o alocare fixă a unui IT (interval de timp), prin care se transmit periodic informații. Rutarea este implicită, ea fiind definită de IT alocat în interiorul cadrului pentru conexiune. Comutarea poate asigura conectarea informației dintr-un IT al unei intrări la oricare dintre ieșiri. Rezultă că traseul de conexiune dintre două terminale este definit de succesiunea de IT (căi temporale) alocate în multiplexurile temporale între nodurile de comutație care participă la conexiune. Această tehnică, numită conexiune mod circuit, asigură utilizatorului o cale prin rețea cu debit fix de 64 kbit/s (8 biți / 125 μ s).

Datele ce sunt transferate între două DTE sunt formate din unități de lungime fixă, de obicei de câte 8 biți. De exemplu când un terminal comunică cu un calculator fiecare caracter tastat este codificat într-o valoare binară de 8 biți, întregul mesaj fiind format dintr-un șir de astfel de caractere codificate. Deoarece fiecare caracter este transmis serial, echipamentul receptor pentru a decodifica și interpreta corect biții transmiși trebuie să cunoască:

- rata de emisie a biților (durata unei celule bit);
- începutul și sfârșitul fiecărui caracter (octet);
- începutul și sfârșitul fiecărui mesaj complet (bloc).

Acești trei factori sunt cunoscuți sub numele de **sincronism la nivel de bit, sincronism la nivel de caracter și sincronism la nivel de bloc**.

Din acest punct de vedere comunicația între două echipamente poate fi de două tipuri:

- 1) **asincronă** - dacă ceasul receptorului este independent de cel al emițătorului.
- 2) **sincronă** - dacă ceasurile emițătorului și receptorului sunt sincrone.

În cazul în care datele ce trebuie transmise sunt formate din caractere separate de intervale de timp de lungime aleatoare atunci fiecare caracter este transmis independent și receptorul se

sincronizează la începutul fiecărui nou caracter primit. Pentru acest tip de comunicație se folosește transmisia asincronă.

În cazul în care datele ce trebuie transmise sunt formate din blocuri conținând mai multe caractere (octeți) fiecare, ceasurile emițătorului și receptorului trebuie să se afle în sincronism pentru mai mult timp și de aceea se folosește transmisia sincronă.

Cap. II. 3. Transmisia – recepția sincronă

În cazul transmisiei asincrone folosirea biților adiționali (de start și stop) este ne semnificativă datorită intervalelor mari de timp între două caractere.

Uneori este necesară însă transmisia unor blocuri de date de lungime mare (transmisia de fișiere între două calculatoare). În acest caz folosirea biților suplimentari la fiecare caracter devine supărătoare.

Totodată, datorită mecanismului de sincronizare folosit de schema asincronă, aceasta nu poate funcționa fără erori decât până la aprox. 19200 bps.

Alternativa eficientă în aceste situații este transmiterea unui bloc complet ca o singură entitate, adică transmisia sincronă.

Pentru a permite echipamentului receptor să se sincronizeze trebuie respectate condițiile:

- fluxul de biți transmis să fie astfel codificat încât receptorul să poată fi menținut în sincronism la nivel de bit;
- toate blocurile transmise să fie precedate de unul sau mai multe caractere speciale astfel încât la recepție să poată fi delimitați corect octeții (sincronism la nivel de caracter);
- conținutul fiecărui bloc să fie delimitat de o pereche de caractere speciale.

Ultima condiție permite receptorului să determine începutul unui nou bloc atunci când a primit un caracter special (de start) după o perioadă liberă.

În intervalul de timp dintre două blocuri, fie sunt transmise continuu caractere de sincronizare (pentru a întreține sincronismul receptorului la nivel de bit și byte), fie blocurile sunt precedate de unul sau mai mulți octeți de sincronizare (permițând astfel receptorului să revină în sincronism).

În cadrul transmisiei sincrone este necesar să se asigure că octeții sau caracterele de sincronizare să fie unice adică să nu fie prezente și în conținutul blocului ce se transmite.

Cap. III CONTROLUL TRANSMISIEI SINCRONE

Cap. III. 1. Circuite pentru controlul transmisiei

Așa cum s-a arătat, între două DTE datele sunt transmise serial (elemente de câte 8 biți) folosind fie modul sincron fie pe cel asincron (figura 3.4). În interiorul DTE fiecare element este memorat și transferat într-o formă paralelă. Din acest motiv circuitele de control al transmisiei din cadrul fiecărui DTE ce reprezintă de fapt interfața între DTE și legătura serială trebuie să realizeze următoarele funcții:

1. conversie paralel-serie în vederea pregătirii elementului pentru a fi transmis pe linie;
2. conversie serie-paralel a fiecărui element recepționat în vederea memorării și prelucrării sale în interiorul DTE;
3. folosirea metodei adecvate la recepție pentru a realiza sincronismul la nivel de bit, caracter sau bloc;
4. generarea unor biți cu scopul detectării erorilor de transmisie și eventual detectarea acestor erori dacă apar.

Pentru a satisface aceste cerințe au fost proiectate circuite integrate speciale.

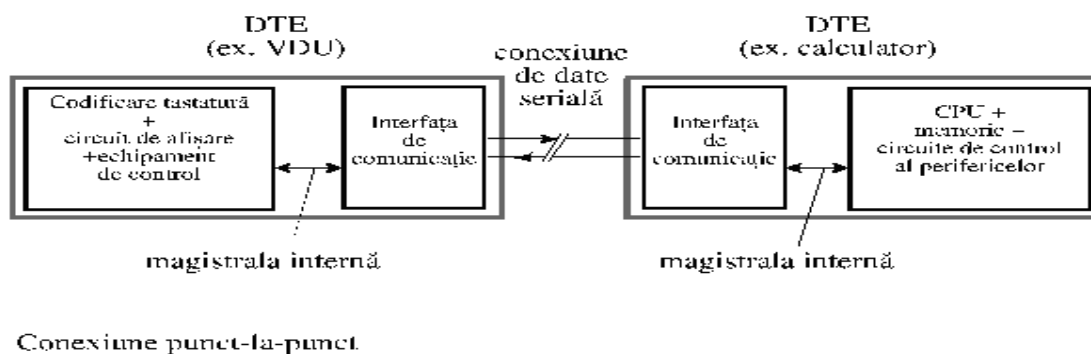


Figura 3.1

Transmisia sincronă

De multe ori transmisia sincronă este deosebită de cea asincronă prin natura elementelor transmise (blocuri sau caractere). De fapt deosebirea majoră între cele două tipuri de comunicație este reprezentată de sincronizarea sau nesincronizarea ceasurilor la emisie și recepție.

Transmisia sincronă ar putea fi realizată prin folosirea unei linii de legătură suplimentare prin care să se emită ceasul de sincronizare $T_x\text{Clk}$. Astfel, echipamentul receptor poate determina exact momentul în care a fost emis un nou bit. În practică nu se folosește această linie suplimentară, informația de ceas fiind conținută în unda transmisă. Prin această metodă tactul de eșantionare la recepție trebuie extras din fluxul de date primit cu ajutorul unui circuit specializat.

Există două moduri de realizare a comunicației sincrone:

1. orientată pe caracter;
2. orientată pe bit.

Diferența majoră între cele două metode constă în modul în care este detectat începutul și sfârșitul unui element transmis. În cadrul metodei orientate pe bit receptorul poate detecta sfârșitul elementului la orice bit (nu numai la biți multipli de 8). Aceasta face ca elementul să poată avea o lungime de N biți nu neapărat multiplu de 8. În practică această metodă este mai puțin folosită deoarece majoritatea aplicațiilor folosesc elemente formate din octeți.

Transmisia sincronă orientată pe caracter

În cazul acestei metode elementele ce trebuie transmise sunt formate din caractere de 7 sau 8 biți ce sunt emise sub forma unui șir continuu de biți fără întârzieri între ei.

Echipamentul receptor pentru a realiza sincronizarea trebuie să fie în stare să:

- detecteze începutul și sfârșitul fiecărui caracter (sincronism la nivel de caracter);
- detecteze începutul și sfârșitul fiecărui element (bloc) complet (sincronism la nivel de bloc).

Pentru realizarea acestor lucruri au fost oferite diverse soluții, obiectivul principal fiind de a face procesul de sincronizare independent de conținutul blocului de date.

Schema cea mai răspândită este cea folosită de protocolul de control sincron numit Basic Mode. Acest protocol este folosit pentru transferul informației alfanumerice între terminale inteligente și calculator.

În cadrul protocolului Basic Mode sincronizarea la nivel de caracter se realizează prin transmisia a două sau mai multe caractere de sincronizare (SYN) imediat înaintea fiecărui bloc de date.

Receptorul, la pornire sau după o perioadă liberă, urmărește bit cu bit fluxul recepționat până când detectează caracterul de sincronizare cunoscut. În acest moment receptorul a realizat sincronizarea la nivel de caracter, în continuare șirul de biți fiind tratat ca o secvență continuă de caractere de 7 sau 8 biți (după cum a fost programat UART- ul).

În protocolul Basic Mode caracterul de sincronizare SYN (00010110) este unul din caracterele rezervate din setul de coduri de caractere definit de ISO. Din acest set fac parte și caracterele de început (STX) și sfârșit (ETX) de bloc.

Toate caracterele sosite după un caracter STX sunt comparate cu codul ETX. În cazul în care caracterul recepționat nu este ETX atunci el este memorat. Dacă este un caracter ETX recepția blocului se încheie putându-se trece la prelucrarea informației recepționate. Această variantă este satisfăcătoare atunci când informația transmisă este formată din caractere tipăribile (de exemplu introduse de la tastatură). În acest caz nu este posibilă prezența accidentală a unui caracter ETX în interiorul blocului. Dacă acest lucru se întâmplă, recepția se termină anormal.

În unele aplicații însă conținutul blocului poate fi un fișier binar. În acest caz trebuie făcute unele operații suplimentare pentru a putea identifica în mod corect sfârșitul de bloc. Acesta este modul "data transparent" (independent de date) și folosește o pereche de caractere pentru identificarea atât a începutului cât și a sfârșitului de bloc.

Detecția incorectă a sfârșitului de bloc se elimină în felul următor: de câte ori emițătorul întâlnește în interiorul blocului un caracter DLE, el inserează după acesta încă un caracter DLE. La recepție, acest al 2-lea caracter va fi eliminat. Receptorul determină astfel sfârșitul de bloc prin secvența unică DLE-ETX.

În varianta de transmisie orientată pe bloc de informație, erorile sunt detectate pe baza unor biți suplimentari calculați pe baza conținutului blocului și transmiși după sfârșitul de bloc. Pentru a menține independența față de conținutul blocului, caracterele pentru verificarea erorilor sunt transmise după încheierea secvenței de bloc.

Transmisia sincronă orientată pe bit

În varianta orientată pe bit, fiecare bloc transmis poate conține un număr arbitrar de biți care nu este neapărat multiplu de 8.

Indicatorul de început și sfârșit de bloc este același. Pentru a asigura independența față de date este necesar ca acest indicator să nu poată fi prezent în conținutul blocului. Acest lucru este realizat prin tehnica inserării unui bit "0" de câte ori echipamentul emițător detectează în conținutul blocului un șir continuu de 5 biți "1". În acest fel secvența 01111110 nu poate fi niciodată transmisă între începutul și sfârșitul blocului.

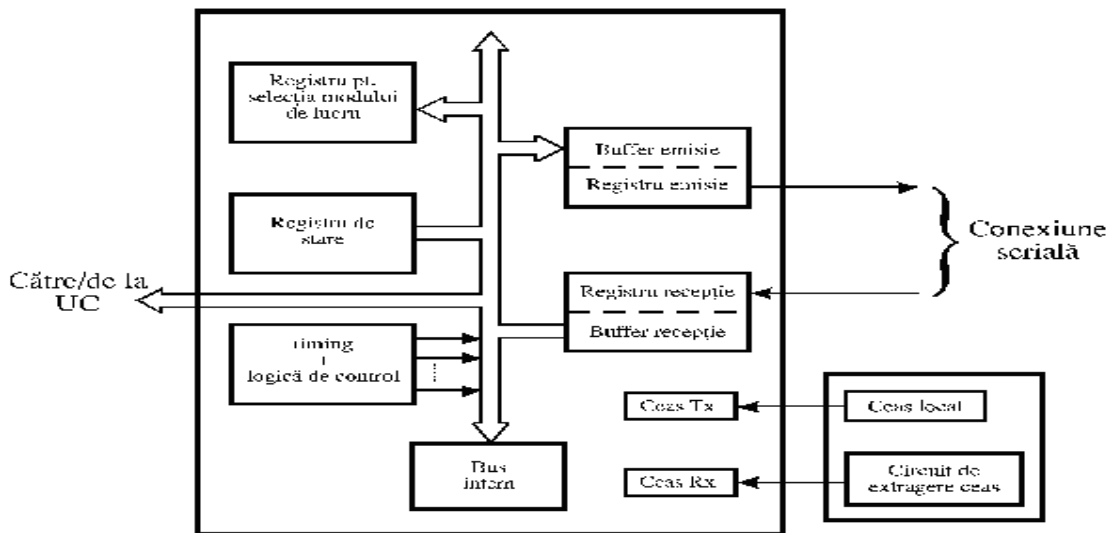
Receptorul, după detectarea indicatorului de început al blocului, contorizează biții "1" consecutivi și în cazul în care după 5 biți "1" urmează un bit "0" acesta este eliminat.

În varianta orientată pe byte fiecare bloc conține la sfârșit biți suplimentari pentru detectarea erorilor. Biții "0" inserați și eliminați nu sunt incluși în procesul de detectare a erorilor.

C. Universal Synchronous Receiver and Transmitter USRT

Circuitele de interfață folosite pentru controlul comunicației sincrone orientate pe caracter sunt cunoscute sub numele USRT. Termenul "universal" este folosit deoarece modulul este programabil și caracteristicile sale de lucru pot fi modificate de către utilizator.

În figura 3.2 se prezintă schematic structura unui USRT.



USRT - schema bloc

Figura 3.2

Pentru a folosi un astfel de modul, în primul rând trebuie selectate caracteristicile de funcționare prin încărcarea unei valori corespunzătoare în registrul de selecție mod.

Biții pentru selecția lungimii și parității au același sens ca în cazul UART-ului.

Bitul SCS permite utilizatorului să selecteze fie unul, fie două caractere de sincronizare SYN care preced fiecare bloc transmis.

Unitatea de control determină starea curentă a USRT prin citirea conținutului registrului de stare și testarea unor biți specifici.

La începutul transmisiei unitatea de control inițiază emiterea unor caractere de sincronizare pentru a permite receptorului să poată realiza sincronismul la nivel de caracter. Caracterele de sincronizare SYN sunt încărcate în buffer-ul de emisie atunci când T_xBE devine "1".

După emisia caracterelor de sincronizare se trece la emisia blocului de date. Acesta este transferat caracter cu caracter în bufferul de emisie, operația fiind controlată de starea bitului T_xBE .

După ce a fost transmis și ultimul caracter al blocului, USRT automat începe să transmită caractere de sincronizare SYN până când unitatea de control este gata să transmită un nou bloc. În acest fel receptorul poate menține sincronismul între două blocuri succesive.

La echipamentul receptor unitatea de control setează USRT-ul în modul urmărire ceea ce face ca logica de control a acestuia să compare conținutul buffer-ului de recepție cu caracterul de sincronizare după fiecare nou bit recepționat. Când a fost detectată o coincidență este setat bitul SYNDET care indică obținerea sincronismului la nivel de caracter.

În continuare receptorul așteaptă un caracter STX care să indice începutul recepției unui bloc de date. Fiecare caracter al blocului este apoi preluat de unitatea de control sub controlul bitului R_xBF până când este detectat caracterul ETX.

În modul sincron toate datele sunt emise și recepționate cu o rată determinată de ceasurile de emisie și recepție. Ceasul de recepție se obține din unda recepționată prin intermediul unui circuit de extragere a ceasului.

Cap. III. 2. Sincronizarea datelor

Sincronizarea la nivel de bit

S-a arătat că în cazul comunicației asincrone se folosește un ceas separat la recepție a cărui frecvență este de câteva ori mai mare decât rata de comunicație. Apoi, la detectarea primei tranziții a bitului de start al fiecărui caracter, receptorul, pe baza ceasului local, estimează centrul fiecărei celule bit. Această metodă este acceptabilă pentru transmisia asincronă din două motive:

- rata de transfer maximă folosită este relativ mică (aprox. 19,2 Kbps);
- metoda de codificare asigură garanția sincronizării la începutul fiecărui caracter.

În cazul transmisiei sincrone, biții de start și stop nu sunt folosiți. Fiecare bloc este transmis ca un flux continuu de cifre binare. De aceea este necesar să se utilizeze o metodă diferită de sincronizare la nivel de bit.

O soluție este evident de a folosi două perechi de linii între emițător și receptor : una pentru fluxul de date și cealaltă pentru semnalul de ceas asociat. Această soluție este însă foarte rar aplicată în practică deoarece într-o rețea telefonică este de obicei disponibilă o singură pereche de linii.

Din acest motiv pentru realizarea sincronizării la nivel de bit s-au propus alte două soluții:

- informația reprezentând semnalul de ceas este inclusă în fluxul de biți și este extrasă din aceasta de către receptor;
- informația ce trebuie transmisă este astfel codificată încât există suficiente tranziții sigure în fluxul transmis pentru a sincroniza un circuit de ceas la receptor.

Fluxul de biți ce trebuie transmis este codificat astfel încât "1" binar este reprezentat printr-un impuls pozitiv în timp ce "0" binar este reprezentat printr-un impuls negativ. Această metodă de codificare este cunoscută sub numele de codificare bipolară.

În cazul metodei de codificare bipolară fiecare celulă bit conține și informația de ceas care poate fi extrasă din unda transmisă printr-un simplu circuit de corecție și întârziere. Deoarece semnalul revine în zero după fiecare bit codificat el se numește cu întoarcere în zero -return to zero RZ-. Semnalul RZ necesită trei nivele de amplitudine pentru a reprezenta fluxul de biți.

Unda rezultată este numită fără întoarcere în zero -non return to zero NRZ- iar metoda de codificare se numește codificare în fază - phase (Manchester) encoding PE -.

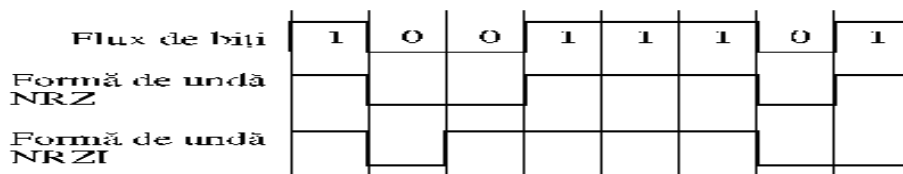
Circuitul pentru extragerea ceasului folosit în cazul metodei PE este ceva mai complicat și se bazează pe existența unei tranziții pozitive sau negative la mijlocul fiecărei celule bit.

În cazul metodei bipolare ceasul extras este folosit pentru eșantionarea fluxului recepționat la mijlocul fiecărei celule bit, în timp ce în cazul metodei PE fluxul de biți este eșantionat în a doua jumătate a fiecărei celule bit.

O a doua soluție este de a utiliza o sursă de tact stabilă a receptorului care să fie menținută în sincronism cu fluxul ce se recepționează. Deoarece în cazul transmisiei sincrone nu există biți de start și stop, informația trebuie să fie astfel codificată încât să existe suficiente tranziții care să permită resincronizarea ceasului receptorului la anumite intervale de timp. Pentru rezolvarea acestei probleme există două soluții:

1. datele ce trebuie transmise sunt trecute printr-un circuit de amestec ce are rolul de a înlătura șiruri continue de 1 sau 0.
2. datele sunt astfel codificate încât să fie garantată prezența naturală permanentă a tranzițiilor.

În figura 3.3 se prezintă modul de codificare NRZI (non return to zero inverted) comparativ cu NRZ.



Codificare NRZI (diferențială)

Figura 3.3

În cazul modului de codificare NRZI (ce se mai numește și codificare diferențială) nivelul semnalului se modifică în cazul transmiterii unui bit 0 și rămâne nemodificat pe bit 1. Aceasta înseamnă că un semnal NRZI va conține întotdeauna tranziții cu excepția cazului în care fluxul transmis conține un șir continuu de 1. Pentru a înlătura această situație se poate adopta metoda inserării de zero-uri după fiecare 5 biți "1" consecutivi. Unda rezultată va conține în mod sigur tranzițiile necesare pentru sincronizarea receptorului.

Circuitul folosit pentru menținerea sincronizării la nivel de bit este cunoscut sub numele de DPLL (digital phase-locked loop). DPLL funcționează pe baza unui oscilator controlat de cuarț având o frecvență suficient de stabilă care nu necesită decât mici ajustări la intervale de timp neregulate. De obicei frecvența oscilatorului este de 32 ori mai mare decât rata de transfer.

Presupunând că fluxul de biți transmiși și ceasul local sunt în sincronism, starea semnalului de pe linie va fi determinată prin eșantionare la centrul fiecărei celule bit. Perioada de eșantionare este de 32 ori mai mare decât perioada ceasului local.

În cazul în care fluxul de biți receptionați și ceasul local ies din sincronism reglarea momentelor de eșantionare.

Dacă pe linie nu sunt tranziții DPLL generează câte un impuls de eșantionare după 32 perioade de ceas. În momentul în care este detectată o tranziție (1□□ 0 sau 0□□ 1) DPLL compară momentul apariției tranziției cu momentul estimat de DPLL. Pentru a realiza acest lucru fiecare perioadă bit este împărțită în 4 părți notate în figură A,B,C,D. Fiecare parte are o durată de 8 perioade de ceas. În cazul în care tranziția este detectată pe timpul sfertului A înseamnă că ultimul impuls de eșantionare a fost dat prea târziu și deci perioada pentru următorul impuls va fi scurtată la 30 perioade de ceas.

În acest fel, prin ajustări succesive, impulsurile de eșantionare sunt generate în apropierea mijlocului fiecărei celule bit.

Când se folosește un DPLL, înaintea transmiterii primului bloc pe linie de obicei se transmit câteva caractere care să asigure minimum 12 tranziții bit (două caractere compuse numai din 0 asigură 16 tranziții în cod NRZI). Astfel DPLL în momentul primirii indicatorului de început de bloc va genera corect impulsurile de eșantionare.

Cap. III. 3. Detectarea erorilor

Metode de detectare a erorilor

În cazul comunicației de date între două DTE se întâmplă uneori ca semnalele electrice reprezentând fluxul de biți transmis să fie modificate de interferențe electromagnetice datorate unor echipamente electrice vecine. Aceasta înseamnă că semnalul reprezentând un 1 binar poate fi interpretat de receptor ca un 0 binar.

Pentru a exista o mare probabilitate ca informația recepționată să fie identică cu cea transmisă, sunt necesare metode prin care receptorul să stabilească dacă informația primită conține sau nu erori. În plus, în cazul detectării erorilor este necesar un mecanism prin care să se obțină informația corectă.

Pentru realizarea acestui lucru există două metode:

1. controlul anticipat al erorii: fiecare caracter sau bloc transmis conține informații adiționale (redundante) pe baza cărora receptorul nu numai că depistează prezența erorilor, dar reface din fluxul de biți recepționat informația pe care o presupune a fi corectă.
2. controlul posterior al erorii: fiecare caracter sau bloc conține numai informațiile adiționale ce permit receptorului să detecteze prezența erorilor (fără a le putea elimina). Informația eronată va fi retransmisă în speranța că operația se va efectua corect de această dată.

În cazul primei metode numărul de biți adiționali necesari pentru controlul anticipat al erorii crește rapid odată cu creșterea numărului de biți ai informației. Din acest motiv în practică este mult mai răspândită a doua metodă. Aceasta poate fi împărțită în două părți:

- tehnicile ce pot fi folosite pentru detectarea erorilor;
- algoritmi de control asociați schemelor de retransmisie.

În continuare se vor prezenta tehnicile uzuale folosite pentru detectarea prezenței erorilor.

Cea mai răspândită metodă folosită pentru detectarea erorilor atunci când numărul de biți de informație este mic și când probabilitatea prezenței unei erori este mică, este folosirea unui singur bit adițional de paritate pentru fiecare element transmis.

Biții de date ai fiecărui caracter sunt examinați de echipamentul emițător pe baza lor fiind calculat bitul de paritate. Acesta este apoi adăugat astfel încât numărul total de 1 în întregul element este fie par, fie impar, în funcție de tipul de paritate folosit (pară sau impară). Receptorul recalculează bitul de paritate pentru caracterul recepționat determinând astfel apariția erorilor de comunicație.

Eficiența unei anumite metode de detectare a erorii depinde foarte mult de tipul erorilor ce pot să apară. Astfel, metoda parității este eficientă în cazul în care un singur bit al unui caracter este interpretat greșit la recepție. În cazul în care doi biți sunt modificați, eroarea nu va putea fi detectată pe baza bitului de paritate.

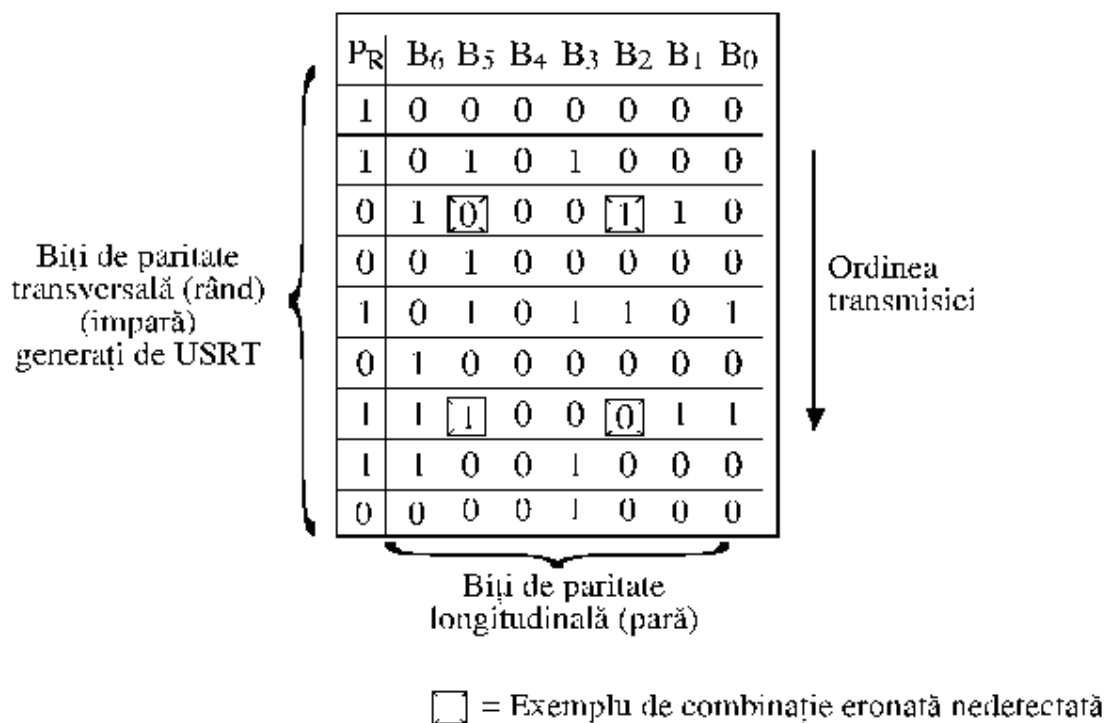
Deoarece paritatea este folosită atât în cazul comunicației asincrone cât și în cazul comunicației sincrone orientată pe caracter atât UART-urile cât și USRT-urile conțin facilități pentru:

- calculul și inserarea bitului de paritate în fiecare caracter la emisie;
- recalcularea parității la recepție pentru fiecare caracter primit și semnalarea apariției unei erori.

Posibilitățile de detectare a erorilor pot fi extinse în cazul folosirii unui singur bit de paritate pe caracter prin folosirea unui set adițional de biți de paritate calculați pe baza întregului set de caractere din bloc. Prin această metodă fiecărui caracter îi este asociat un bit de paritate și în plus este generat un bit suplimentar de paritate pentru fiecare poziție de bit (coloană) din întregul bloc. Setul de biți de paritate rezultați se numește sumă de control. Un exemplu se prezintă în figura 3.4.

Biții de paritate generați de USRT pentru fiecare caracter se numesc biți de paritate transversală, iar biții suplimentari de paritate generați pentru fiecare coloană se numesc biți de paritate longitudinală.

Deoarece biții suplimentari de paritate (pe coloană) se calculează ca sumă modulo 2 a biților din fiecare coloană, caracterul final rezultat se numește sumă de control a blocului.



Exemplu de sumă de control

Figura 3.4

Exemplul prezentat în figura 3.4 folosește paritatea impară pentru rânduri și paritate pară pentru coloane. De exemplu, se poate observa că doi biți eronați într-un caracter nu pot fi detectați cu bitul de paritate transversală dar eroarea poate fi semnalată cu biții de paritate longitudinală. Acest lucru nu este valabil atunci când apar doi biți eronați pe aceeași coloană (ca în figură). Deoarece probabilitatea de apariție a acestui caz este mică metoda este folosită în cazul transmisiei (orientată pe bloc) pe linii cu probabilitate de erori mică. Pentru linii mai zgomotoase se folosesc metode mai riguroase de detectare a erorilor.

Când încearcă să comunice între ele, două dispozitive trebuie să aibă o modalitate de a controla fluxul de date, astfel încât să înțeleagă unde încep și unde se termină caracterele trimise. Șirul de date care este trimis efectiv prin modem celuilalt capăt al conexiunii poate fi transmis utilizând una din două forme de coordonare. Un mod de a controla sincronizarea semnalelor trimise și recepționate la oricare dintre capete prin trimiterea datelor *asincron*. Comunicația asincronă este cea mai răspândită formă utilizată de modemurile convenționale. În comunicația asincronă, informația (caracter, literă, număr sau simbol) care este trimisă de un dispozitiv la altul este reprezentată utilizând un șir de biți. Fiecare șir de biți este separat de celelalte printr-un bit de start și un bit de stop. Utilizând acești biți de start și de stop pentru

fiecare caracter transmis, fiecare dispozitiv știe când trimite sau primește un caracter și nu trebuie să fie prezente semnale de sincronizare externe pentru a controla fluxul de date.

Una dintre nemulțumirile privind comunicațiile asincrone este că aproximativ 20 până la 25 la sută din datele transmise sunt utilizate pentru controlul informației, pentru “sincronizare” conversației dintre dispozitive. Soluția alternativă la comunicația asincronă este *comunicația sincronă*.

În comunicația sincronă trebuie să fie prezent un semnal de sincronizare pentru a controla transmiterea blocurilor de caractere, numite *cadre*. Nu sunt utilizați în transmisie biți de start și de stop. Pentru a începe o transmisiune și pentru a verifica acuratețea acesteia sunt utilizate caractere de sincronizare.

Protocoalele utilizate la transmisiile sincrone realizează funcții care nu sunt realizate de protocoalele asincrone. Iată câteva exemple:

- Verificarea acurateții informației trimise
- Formatarea datelor în cadre
- Adăugarea unor informații de control

Protocoalele sincrone sunt utilizate în medii digitale. Lumea analogică utilizează în mod tipic comunicațiile asincrone. Majoritatea comunicațiilor de rețea sunt realizate asincron. Binary Synchronous Communication Protocol (bisynch), Synchronous Data Link Control (SDLC – control sincron al legăturii de date) și High Level Data Link Control (HDLC – control de nivel înalt al legăturii de date) sunt câteva dintre protocoalele sincrone obișnuite.

BIBLIOGRAFIE

- **Peter Norton, „ Calculatorul Personal”, Editura Teora, 2000**
- **Peter Norton „ Rețele de calculatoare”, Editura Teora, 2000**
- **P. Borza, M. Dascălul, C. Gavrilescu, „Calculatorul personal”
Editura Tehnică, București, 1999**