

CAPITOLUL VIII
SINCRONIZAREA ÎN
SISTEMELE DIGITALE DE COMUNICAȚII

VIII.1 Aspecte generale ale sincronizării

Canalele de comunicații introduc schimbări mai mult sau mai puțin previzibile în semnalul transmis.

Semnalul recepționat este o replică a celui emis, întârziată, defazată, atenuată, distorsionată, afectată de zgomote. Demodulatorul digital extrage secvența de date transmisă și reface forma impulsului elementar inițial (în general, rectangular). În varianta coerentă de demodulare, este nevoie de semnale purtătoare în concordanță de fază cu cele din semnalul recepționat. Acestea pot fi generate local sau pot fi extrase fie din semnalul recepționat, dacă acesta conține în spectru anumite componente discrete, fie de pe un canal secundar de transmisie.

Apar mai multe etape de sincronizare în cazul sistemelor digitale.

Sincronizarea de purtătoare elimină incertitudinea de fază din semnalul recepționat. Ca și în cazul sistemelor analogice de comunicații, aceasta se realizează fie cu **circuite de tip PLL** dacă se transmite periodic purtătoarea intercalată cu secvența de date sau dacă există în spectru componente discrete pe frecvența purtătoare sau pe subpurtătoare, fie cu **circuite neliniare** (buclă Costas, buclă pătratică, DFL) lucrând pe principiul estimatorului de maximă probabilitate, pentru cazul 'modulației totale' (cu purtătoare suprimată). Transmisia suplimentară a semnalului de sincronizare determină creșterea puterii de emisie, eventual a lățimii benzii utile ori scăderea vitezei efective de transfer a datelor.

Sincronizarea de bit (simbol) realizează o ajustare fină în timp a semnalului de tact, respectiv a momentelor de eșantionare în receptor, pe baza impulsului formator al datelor, independent de secvența de date transmisă.

Sincronizarea de cuvânt (bloc) urmărește împărțirea corectă a șirului de date în blocuri, cu lungime fixă sau variabilă. În acest scop, în funcție de modul de transmisie adoptat (sincron sau asincron), se caută combinațiile specifice de biți care marchează începutul sau sfârșitul cuvintelor de cod (de exemplu, secvențe de control STX, ETX, SOH, EOH, DLE STX, DLE ETX, EOB, START, STOP etc) ori se detectează momentele respective pe baza anumitor proprietăți ale codului folosit. În cazul sistemelor de comunicații cu spectru extins este necesară sincronizarea de bloc a secvenței pseudoaleatoare generată local în receptor cu cea utilizată la emisie.

VIII.2 Sincronizarea de bit

Pentru decodarea corectă a datelor este necesară stabilirea momentelor optime de eșantionare a semnalului recepționat.

Semnalul de tact poate fi transmis pe un canal separat de cel al datelor cu condiția ca acestea să aibă aceleași caracteristici de frecvență și să prezinte aceleași perturbații. Metoda este simplă, ieftină dar apare un consum suplimentar de energie precum și riscul diferențierii condițiilor de transmisie a tactului față de semnalul de date.

Informația de sincronizare poate fi extrasă prin metode de rezonanță (filtre RLC acordate pe frecvența de bit, circuite PLL) din semnalul transmis cu condiția ca acesta să prezinte suficiente tranziții pentru funcționarea sigură a blocului de sincronizare.

De asemeni, tactul poate fi generat local în receptor iar faza acestuia se ajustează periodic, comparând tranzițiile acestuia cu cele din semnalul recepționat.

În cazul în care nu se transmite separat semnalul de sincronizare, printr-o precodare adecvată a datelor și folosind anumite metode neliniare tactul poate fi obținut pe baza secvenței recepționate. Se utilizează coduri de linie (NRZ-M; NRZ-S; RZ; BIF-M; BIF-S; BIF-L; Miller) care crează în spectru componente discrete pe frecvența de bit sau la jumătatea acesteia.

Exemple:

1. Pentru o secvență codată Miller, secvențele de biți identici întretin în spectru o componentă discretă pe $0,5f_{bit}$ (perioadă $2T_{bit}$).

date: 1 1 1 1 1 1 0 0 0 0 0 0.....

semnal codat Miller: 0110011001100111001100110011...

2. Conversia semnalului NRZ-M în semnal RZ se obține cu circuitul din figură, asemănător unui filtru derivator .

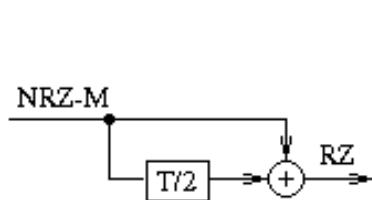


Fig.VIII.1 Circuit neliniar de codare

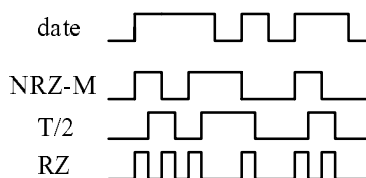


Fig.VIII.2 Forme de undă

În sistemele de comunicații asincrone, tactul este generat local în receptor iar faza acestuia se ajustează odată cu detecția cu un bistabil (setat pe front descrescător) a tranziției HL de START din cadrul de date asincron. Evident, există riscul unor pseudostarturi sau a nerecunoașterii impulsului de START cu pierdere de informații (erori de încadrare - *framing error*). Reducerea erorii de fază este posibilă prin utilizarea unui oscilator pe un multiplu al frecvenței de bit ($2^k f_{\text{bit}}$) care eșantionează semnalul recepționat de mai multe ori pe durata bitului de START. Astfel se limitează eroarea de fază la maximum $2\pi/2^k$ radiani.

Întrucât semnalul recepționat demodulat este distorsionat (filtrat trece-jos) față de unda rectangulară transmisă, este necesar ca UART să corecteze forma semnalului prin eșantionare pe mijlocul intervalelor de bit și cuantizare binară.

Apare deci problema de **detecție a momentelor optime de eșantionare**.

Se aplică următorul **algoritm** pentru determinarea mijlocului bitului START:

Pas 1: Se așteaptă apariția tranziției HL;

Pas 2: După detectarea tranziției de START se eșantionează semnalul cu o rată de k ori mai mare decât frecvența de bit (fig.VIII.3);

Pas 3: Se testează eșantioanele din a doua jumătate a perioadei de bit (după $k/2$ fronturi active din tactul de eșantionare):

- dacă se detectează, pe eșantionul de valoare minimă, valoarea logică '0' se consideră că s-a recepționat bitul de START și se continuă eșantionarea semnalului o dată în fiecare interval de bit începând cu momentul respectiv;

- dacă apare numai valoarea logică '1' se consideră că tranziția detectată nu reprezintă secvență de START (alarmă falsă) și se abandonează recepția.

Minimizarea erorii de sincronizare (*jitter*) se poate face prin aplicarea unor factori de multiplicare de 16, 32 sau 64.

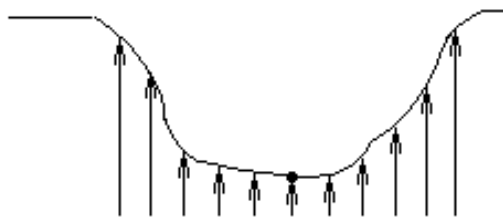


Fig.VIII.3 Detecția momentului optim de eșantionare

VIII.3 Sincronizarea de bloc

Rolul sincronizării de bloc este de a împărți corect șirul de simboluri (biți) în blocuri pentru a se putea efectua decodarea.

În acest sens, se utilizează fie secvențe speciale de delimitare a blocurilor de date (*marker*) fie impulsuri de o anumită formă, cu **proprietăți optime de corelație**.

Secvențele de delimitare a blocurilor de simboluri au funcții de autocorelație cu un maxim pronunțat în origine, astfel încât prin corelarea a două secvențe de același tip generate cu circuite diferite, să se poată corecta defazajul existent între ele. **Secvențele Barker** (Tabelul VIII.1) au proprietăți optime de corelație. Aceste secvențe binare au un anumit număr de biți. Cu cât lungimea secvenței este mai mare, cu atât eroarea maximă de fază este mai redusă.

Tabelul VIII.1

Secvențe Barker uzuale

Lungimea secvenței	Secvență Barker
3	110
4	1101
5	11101
7	1110010
11	11100010010
13	1111100110101

Exemplu: Să calculăm valorile funcției de autocorelație a secvenței Barker de ordin 5, reprezentată dublucurent (+++ - +), periodizată (Fig.VIII.4).

$$R(k) = \frac{1}{N} \sum_{i=1}^N a_i a_{n+i} \rightarrow R(0) = 1 = \text{maxim}; R(1) = R(2) = R(3) = R(4) = 1/5 .$$

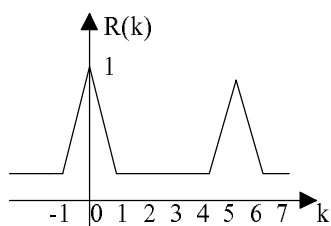


Fig.VIII.4 Funcția de autocorelație secvenței Barker de ordin 5

În general, valoarea maximă este 1 iar cea minimă $1/N$ (N - perioada secvenței).

Sincronizarea de cuvânt este absolut necesară în cazul codării-bloc a datelor. Foarte utile sunt codurile de tip prefix prin capacitatea lor de autosincronizare. Mai general, există coduri **autosincronizante** (*comma-free codes*), la care prin reunirea oricăror două cuvinte de cod nu rezultă alt cuvânt de cod. Aceste coduri sunt mai puternice decât metodele de marcare a momentelor de început și sfârșit a blocurilor de simboluri, întrucât elimină posibilitatea falsei sincronizări și permit creșterea vitezei de transmisie a informației propriu-zise. Totuși până în prezent, algoritmi respectivi de codare/decodare sunt insuficient studiați și dezvoltati.

Codurile-bloc corectoare de erori, cu lungime fixă a lungimii cuvântului de cod (Hamming, Reed-Solomon ș.a.), nu intră în categoria codurilor autosincronizante, deoarece prin permutarea simbolurilor unui cuvânt se obține alt cuvânt de cod. Există posibilitatea transformării acestor coduri în așa-numitele **coduri coset** în vederea asigurării autosincronizării.

Se alege un vector **h** de aceeași lungime cu cea a cuvintelor codului-bloc ales. Vectorul datelor **d** se codează prin vectorul de cod **c** care se sumează cu vectorul **h**, în câmpul de definiție (GF) a codului inițial. Se obține astfel un cod coset. La recepție, se extrage **h** din vectorul recepționat **r** în vederea sincronizării de bloc.

Exemplu: Codul coset RS(n;k)

Cuvintele de cod RS se sumează cu vectorul de n biți:

$$\bar{h} = [0 \ 0 \ \dots \ 0 \ 1].$$

Deci se transmite vectorul:

$$\bar{c}' = [c_0 \ c_1 \ \dots \ c_{n-2} \ (c_{n-1} + 1)].$$

Dacă șirul de simboluri recepționat:

$$c_0 \ c_1 \ \dots \ c_{r-1} [c_r \ \dots \ (c_{n-1} + 1) \ c_n \ \dots \ c_{n+r-1}] \ \dots$$

este greșit împărțit în cuvinte (eroare de sincronizare de r simboluri),

prin extragerea vectorului **h** se obține:

$$[c_r \ \dots \ (c_{n-1} + 1) \ c_n \ \dots \ (c_{n+r-1} + 1)]$$

cu erori pe cele două poziții. Prin algoritmul de corecție a erorilor se detectează pozițiile erorilor rezultând astfel valoarea factorului **r** de desincronizare.