

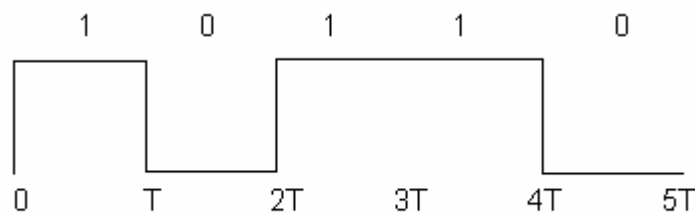
CODURI DE LINIE BINARE

1. GENERALITĂȚI

Datele sunt reprezentate binar, prin biții 1 (M - *Mark*) și 0 (S - *Space*), prin nivelele logice H (*high*) și L (*low*), asociate fie cu nivele de tensiune **polare** ($\pm A$) în așa-numite **transmisii bipolare** sau **dublu-curent**, fie cu nivele de tensiune **unipolare** (0 și A) în **transmisii simplu-curent**.

Pentru fiecare bit 1 se transmite un impuls rectangular, de nivel $+A$ și durată T numită **perioadă de bit**. Frecvența de bit se calculează ca inversa perioadei de bit. De exemplu, perioada de $1 \mu\text{s}$ corespunde frecvenței de 1 MHz.

Bitul 0 este asociat în comunicațiile bipolare cu un impuls rectangular de nivel $-A$ și durată T , astfel că pentru probabilități egale de apariție a biților, componenta de curent continuu (c.c.) a semnalului codat este nulă.



SECVENȚA DE DATE

Secvența de date poate să fie generată de o sursă de informație digitală (de exemplu, computer, CD, DVD etc.), sau poate fi obținută prin conversia analog-digitală a unui semnal analogic (de tip voce, audio, video).

În sistemele de comunicații digitale, cu transmisie pe cablu (linie telefonică, fibră optică etc.) sau prin undă radio, se utilizează coduri de linie care facilitează sincronizarea de bit la recepție prin crearea în spectrul semnalului codat a unor componente spectrale discrete pe frecvența de bit sau pe un submultiplu al acesteia. Menținerea în sincronism a blocului de sincronizare de bit din receptor este facilitată de tranzițiile multiple din semnalul recepționat codat.

Constrângerile impuse duratei maxime pe care nivelul semnalului este constant (pe secvențe de biți identici „0” sau „1”) sunt absolut necesare întrucât se creează o componentă de curent continuu, simultan cu scăderea nivelului componentei discrete utilizate pentru sincronizare.

Sistemele ADSL („*Asynchronous Digital Subscriber Line*”) sunt prevăzute cu coduri de linie binare sau ternare.

Codarea este o operație de prelucrare a datelor în **banda de bază** (BB - „*BaseBand*”), realizată pe baza unui **semnal de tact** sau **de clock**, cu frecvența egală cu cea de bit de la intrarea codorului. Spectrul semnalului codat este centrat pe frecvența zero.

2. DESCRIEREA CODURILOR DE LINIE BINARE

Există mai multe familii de coduri de linie binare (care generează o secvența codată cu două nivele):

- NRZ (*Non-Return-to-Zero*): NRZ-L (*level*); NRZ-M (*mark*); NRZ-S (*space*).
- BIF (*Biphase*): Manchester sau BIF-L; BIF-M; BIF-S.
- RZ (*Return-to-Zero*)
- Miller.

Unele coduri asociază biții de date cu nivele de tensiune și de aceea sunt denumite **coduri de nivel** (*level*), precum NRZ – L, BIF – L, RZ.

Alte legi de codare asociază biții de date cu tranziții din semnalul codat, fiind denumite **coduri prin tranziții**. În această categorie intră codurile NRZ – M, NRZ – S, BIF – M, BIF – S și Miller.

În sistemele de transmisie dublu-curent, codarea prin nivele de tensiune poate genera așa-numita **ambiguitate de fază**, prin inversarea polarităților firelor la recepție, ceea ce conduce la o eroare totală de decodare, obținându-se secvența de date negată.

Eliminarea ambiguității de fază este posibilă în cazul codului Miller.

Codarea prin tranziții nu creează această problemă, întrucât biții nu mai sunt asociați cu nivele absolute de tensiune, ci cu tranzițiile de semnal, adică trecerile HL sau LH din secvența codată.

Legea de codare NRZ – L asociază bitul 1 cu nivelul H, iar bitul 0 cu nivelul L.

Legea de codare BIF – L asociază bitul 1 cu nivelul H pe prima semiperioadă de bit și L pe cea de a doua, adică 1 este codat HL sau 10. Bitul 0 se codează invers prin nivelele LH, sau secvența 01. Rata de codare este în acest caz 1:2. Frecvența de bit la ieșirea codorului este dublul frecvenței de bit de la intrarea acestuia.

Codul RZ are rata de codare 1:2 și codează bitul 1 prin HL sau 10, iar bitul 0 ca LL sau 00. Se folosește în general în varianta unipolară, când se transmite un impuls îngustat pentru bitul 1 și „nimic” pentru bitul 0.

Codul NRZ – M asociază bitul 1 cu o tranziție iar bitul 0 cu lipsa oricărei tranziții.

Similar, **codul NRZ – S** asociază bitul 0 cu o tranziție și bitul 1 este transmis fără tranziție.

Se observă că de fapt codul NRZ – S se obține prin aplicarea legii de codare NRZ – M secvenței de date negate.

Codurile bifazice prin tranziții definesc un bit activ: 1 pentru BIF – M și 0 pentru BIF – S. Celălalt bit este considerat pasiv. Bitul activ este asociat cu două tranziții, una la începutul intervalului de bit și a doua la mijlocul intervalului. Bitul pasiv este asociat cu o singură tranziție, la începutul intervalului de bit. Se observă că tranziția de la începutul intervalului de bit nu diferențiază biții. Decodarea se va face în funcție de cea de la mijlocul intervalului de bit.

Ca și la codurile NRZ, codurile BIF – M și BIF – S se obțin unul din celălalt prin aplicare pe secvența de date negată.

Codarea Miller asociază bitul 1 cu o singură tranziție la mijlocul intervalului de bit și între doi biți 0 apare de asemenea o tranziție.

O stare a codorului este definită prin combinația de biți de intrare/ de ieșire din codor.

De exemplu, în cazul codului RZ sunt definite 2 stări: 1/10 și 0/00. Codul Miller are patru stări: 1/01, 1/10, 0/11 și 0/00.

Descrierea matematică a acestor coduri se face cu ajutorul grafurilor orientate, folosind diagrame de tranziții cu număr finit de stări (FSTD – „*Finite State Transition Diagram*”).

Identificarea tranzițiilor dintre stări se face prin codarea unei secvențe de date aleatoare, suficient de lungă pentru a pune în evidență toate tranzițiile posibile.

Analiza matematică a codurilor de linie se face matricial, pe baza probabilităților de apariție a biților în secvența de date.

Implementarea acestor coduri poate fi realizată în variantă software, ca algoritmi de codare/decodare.

Algoritmii de codare și de decodare se pot implementa în orice limbaj de programare (C, C++) și pot fi rulați folosind procesoare digitale de semnal.

Circuitele de codare/decodare sunt implementate ca automate cu număr finit de stări, folosind porți logice și bistabile.

3. MODUL DE LUCRU

- a. Se deschid programul MATLAB și fișierul *cod_bin.txt* (vezi punctul 5).
- b. Se vizualizează secvența de date și secvențele codate (NRZ-M, NRZ-S, BIF-M, BIF-S, RZ, Miller și Manchester), reprezentate cu diraci.
- c. Se deduce în fiecare caz legea de codare.
- d. Se identifică stările fiecărui codor și se trasează diagrama de stări completă.

e. Se identifică legile de codare (c1, c2, c3, c4, c5, c6, c7) pentru o secvență de date oarecare de 16 biți.

4. EXERCITII ȘI ÎNTREBĂRI

1. Codați secvența de date 1 0 1 1 1 0 0 0 1 0 1 0 0 0 1 1 prin toate legile de codare prezentate în lucrare și reprezentați grafic formele de undă cu impulsuri rectangulare asociate secvențelor codate.

2. Care este rata de codare a codului Miller? Cum sunt codate zerourile singulare prin codul Miller?

3. Care este durata maximă a unui impuls rectangular din secvența codată Miller și pentru ce combinație de biți apare?

4. Decodați secvența codată NRZ – M: 0 1 1 0 1 1 1 0 0 1 0 1 1 1 0 1.

5. Decodați BIF- M secvența: 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1.

6. Decodați Miller secvența: 0 1 1 0 0 1 1 1 0 0 0 1 1 1 1 0.

7. Pe ce frecvență apare o componentă discretă în spectrul secvenței codate NRZ – M dacă la intrarea codorului se aplică un șir lung de biți 1? Idem, pentru codurile BIF – M, RZ și Miller.

8. În ce caz apare componenta de curent continuu în spectrul semnalului codat NRZ-S?

5. ALGORITMI MATLAB

În continuare sunt prezentate listingurile funcțiilor Matlab folosite în această lucrare și al fișierului `cod_bin.txt`.

Funcțiile se găsesc în biblioteca `FUNCTII`:

```
>> cd d:\[cale]\functii
% COD_BIN.TXT
% coduri de linie binare
% date de intrare
a=[1 0 1 1 1 1 0 0 0 1 1 1 0 0 0 1];
l=length(a);
stem(a),axis([0 1 0 1.5])

% codare NRZ-M
y1=nrzm(a);r=1/l;
subplot(2,1,1),stem(a),ylabel('date');
subplot(2,1,2),stem(y1),ylabel('NRZ-M');

% codare NRZ-S
n=1-a;
y2=nrzm(n);r=1/l;
subplot(2,1,1),stem(a),ylabel('date');
subplot(2,1,2),stem(y2),ylabel('NRZ-S');
```

```

% codare BIF-M
y3=bifm(a);r=1/2;
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(y3);ylabel('BIF-M');

% codare BIF-S
n=1-a;
y4=bifm(n);r=1/2;
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(y4);ylabel('BIF-S');

% codare RZ
y5=rz(a);r=1/2;
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(y5);ylabel('RZ');

% codare BIF-L
y6=BIFL(a);r=1/2;
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(y6);ylabel('BIF-L');

% codare Miller
y7=miller(a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(y7);ylabel('Miller');

% coduri binare
d=[1 1 1 0 0 0 1 0]
y1=nrzm(d);
y2=nrzm(1-d);
y3=bifl(d);
y4=bifm(d);
y5=bifm(1-d);
y6=miller(d);
y7=rz(d);
subplot(8,1,1),stem(d(1:8));ylabel('date');
subplot(8,1,2),stem(y1(1:8));ylabel('c1');
subplot(8,1,3),stem(y6(1:16));ylabel('c2');
subplot(8,1,4),stem(y7(1:16));ylabel('c3');
subplot(8,1,5),stem(y3(1:16));ylabel('c4');
subplot(8,1,6),stem(y2(1:8));ylabel('c5');
subplot(8,1,7),stem(y4(1:16));ylabel('c6');
subplot(8,1,8),stem(y5(1:16));ylabel('c7');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=nrzm(x)
% algoritm de codare nrz-m
l=length(x);y(1)=0;
for i=2:l
    y(i)=mod2(y(i-1),x(i));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function y=bifl(x)
% algoritm de codare BIF-L
l=length(x);
for i=1:l
    a(i)=mod(x(i),2);
    if a(i)==1
        y(2*i-1)=1;y(2*i)=0;
    end
    if a(i)==0
        y(2*i-1)=0;y(2*i)=1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function y=bifm(x)
% algoritm de codare bif-m
l=length(x);
y(1)=x(1);y(2)=mod2(x(1),y(1));
for k=2:l
    y(2*k-1)=1-y(2*k-2);
    y(2*k)=mod2(x(k),y(2*k-1));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function y=rz(x)
% algoritm de codare RZ
l=length(x);
for i=1:l
    a(i)=mod(x(i),2);
    if a(i)==1
        y(2*i-1)=1;y(2*i)=0;
    end
    if a(i)==0
        y(2*i-1)=0;y(2*i)=0;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function y=miller(x)
% algoritm de codare MILLER
l=length(x);y(1)=0;y(2)=mod2(y(1),x(1));
for i=2:l
    x(i)=mod(x(i),2);
    if x(i)==1
        y(2*i-1)=y(2*i-2);
        y(2*i)=1-y(2*i-1);
    end
    if x(i)==0
        z(i)=1-mod2(x(i),x(i-1));
        y(2*i-1)=mod2(y(2*i-2),z(i));
        y(2*i)=y(2*i-1);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```