

CODURI DE LINIE TERNARE

1. GENERALITĂȚI

Datele de intrare în codorul ternar sunt reprezentate binar, prin biții 1 și 0, dar la ieșirea acestuia se generează 3 nivele: 0 și $\pm A$, adică alfabetul codului este ternar.

Pentru reprezentarea secvenței codate ternar se folosesc simbolurile ternare: $-$, 0 și $+$.

Impulsul rectangular asociat unui bit este de durată T egală cu **perioada de bit** din intrarea codorului și cu **perioada unui simbol** ternar de la ieșirea acestuia. Codarea ternară este realizată sincron cu **semnalul de tact** sau **de clock**, cu frecvența egală cu cea de bit de la intrarea codorului.

Rata de codare este 1: 1 când unui bit i se asociază un simbol ternar. În acest caz viteza de transmisie este nemodificată prin codarea datelor.

Există și coduri ternare cu rată de codare supraunitară, de exemplu 4:3, caz în care viteza de transmisie crește prin codarea ternară a datelor, ceea ce constituie un avantaj al tehnicilor de codare ternară.

În general, codurile ternare sunt astfel proiectate încât să aibă componenta de curent continuu nulă („*dc-free codes*”).

Ca dezavantaj al acestor coduri, trebuie menționat faptul că decodorul ternar va folosi în procesul de decizie două tensiuni de prag ($\pm A/2$) fiind astfel mai complex decât un decodor binar.

Tehnicile de codare ternară sunt folosite atât în comunicațiile „pe fir” (cablu optic, cablu torsadat etc.), cât și în cele „wireless” cu transmisie prin unde radio. Ele creează în spectrul semnalului codat componente discrete utile pentru sincronizarea blocurilor din receptor.

2. DESCRIEREA CODURILOR DE LINIE TERNARE

Există mai multe familii de coduri de linie ternare:

- BIPOLARE: BIP-1 sau AMI (*Alternative Mark Inversion*); în general, BIP-n.
- HDBn (*High Density Bipolar Codes*) – coduri bipolare de densitate ridicată.
- CHDBn (*Complementary HDBn*) – coduri HDB complementare.

- mBnT (m – Binary, n - Ternary), adică m simboluri binare sunt codate prin n simboluri ternare, pe baza unui tabel de codare.
- MSmn (*Mark Space m-n*) - codarea secvențelor de m biți prin secvențe de n simboluri ternare, conform unui tabel de codare.
- PST (*Pair Selected Ternary*), adică perechi de biți selectate și codate ternar.

Legea de codare AMI codează biții 1 alternativ prin + și –, iar bitul 0 prin 0.

Legea de codare BIPn inversează polaritatea impulsului de ieșire la fiecare n biți 1. Bitul 0 se codează numai prin 0. Rata de codare este în acest caz 1:1.

Exemplu:

Date:	1 0 1 1 0 0 0 1 1 1 0 1 0 1
Codat AMI:	+ 0 - + 0 0 0 - + - 0 + 0 -
Codat BIP2:	+ 0 + - 0 0 0 - + + 0 - 0 -

Se observă faptul că toate codurile bipolare au memorie nenulă, fiind memorate una sau mai multe polarități anterioare.

Decodorul bipolar reface secvența de date prin redresarea semnalului codat.

Toate codurile bipolare au dezavantajul că pe durata unei secvențe lungi de biți 0 receptorul se poate desincroniza. De aceea, s-a limitat lungimea secvențelor de simboluri ternare 0 consecutive, în cazul codurilor HDBn și CHDBn.

Codul HDBn codează bitul 1 bipolar (B) conform legii AMI, prin alternarea polarităților, iar biții 0 prin 0, cu excepția celui de al $(n+1)$ -lea bit de 0 consecutiv care va fi asociat cu un impuls + sau –, cu aceeași polaritate cu cea a ultimei polarități memorate, ceea ce constituie o violare a legii de codare bipolare (V).

Întrucât se urmărește anularea c.c., se aplică două convenții de codare a secvențelor de $(n+1)$ biți consecutivi de 0, astfel încât două simboluri codate consecutiv prin V să aibă polarități opuse:

Convenția I: 0 0 0 ... 0 V.

Convenția II: B 0 0 ... 0 V.

Exemplu:

Date:	1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1
Convenția HDB3:	B 0 0 B B 0 0 0 B 0 B 0 0 0 V B 0 0 V B
Semnal codat HDB3:	+ 0 0 - + 0 0 0 - 0 + 0 0 0 + - 0 0 - +.

Decodorul HDBn va lucra cu memorie, revenind în cazul convenției II de codare, asupra bitului asociat simbolului codat prin legea B, care în mod normal este decodat ca 1, dar care va fi trecut în 0 dacă se găsește pe poziția a n -a înaintea unui simbol codat prin legea V.

Din această cauză, algoritmul de decodare HDBn depinde de dimensiunea n a codului.

Pentru a putea folosi un singur algoritm de decodare, independent de dimensiunea n a codului și cu memorie redusă, au fost propuse codurile CHBn, complementare celor HDBn, care păstrează regulile de codare HDBn, cu excepția convenției II aplicate unei secvențe de $(n+1)$ biți de 0 consecutivi care devine: 0 0 0 ... B 0 V.

Exemplu:

Date: 1 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1
 Convenția CHDB3: B 0 0 B B 0 0 0 B 0 B 0 0 0 V 0 B 0 V B
 Semnal codat CHDB3: + 0 0 - + 0 0 0 - 0 + 0 0 0 + 0 - 0 - +.

Astfel decodorul CHDBn va reveni asupra valorii penultimului bit codat prin B înaintea unui bit codat prin V, indiferent de dimensiunea n a codului. Se implementează astfel un singur decodor CHDBn, universal, cu memorie de 2 simboluri.

Codurile mBnT și MSmn sunt descrise în tabele de codare, pe baza **sumei digitale curente** (RDS – „*Running Digital Sum*”).

Tabel 1. CODUL MS43

Date	Suma curentă				
	-2	-1	0	1	2
0000			-0+		
0001			-+0		
0010			0-+		
0011			0+-		
0100			+ -0		
0101			+0-		
0110		+++			+ - -
0111		00+			00-
1000		0+0			0-0
1001		+ -+			-+-
1010		+00			-00
1011		++-			--+
1100	0++			0- -	
1101	+0+			-0-	
1110	++0			--0	
1111	+++			---	

Suma digitală curentă este obținută prin sumarea simbolurilor din secvența codată ternar, inițializată cu zero, și este limitată la o anumită valoare.

Codul PST este un cod bloc, care codează perechile de biți:

11 prin +-, 00 prin -+, 01 prin 0B, 10 prin B0.

(B reprezintă convenția bipolară de inversare a ultimei polarități din secvența codată).

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

O stare a codorului ternar este definită prin combinația de biți de intrare, simboluri de ieșire din codor și memoria codului (una sau mai multe polarități anterioare).

De exemplu, în cazul codului AMI sunt definite 4 stări: 1/+-, 1/-+ și 0/0+, 0/0-.

Codurile ternare au în general un număr mare de stări, fiind utilă folosirea tabelor de codare în locul diagramelor de tranziție.

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 ternare se face matricial, pe baza probabilităților de apariție a biților în secvența de date de la intrarea codorului.

Algoritmii de codare și de decodare se pot implementa în orice limbaj de programare (C, C++) și pot fi testați în diverse programe matematice (de exemplu, Matlab).

Circuitele de codare/decodare ternare se implementează folosind registre de deplasare și sumatoare algebrice.

3. MOD DE LUCRU

- a. Se deschid programul MATLAB și fișierul *cod_tern.txt* (vezi punctul 5).
- b. Se vizualizează secvența de date și secvențele codate (Bipolare, HDB, CHDB, PST) urmărindu-se în fiecare caz legea de codare.
- c. Se identifică stările și se trasează diagramele de stări ale codurilor AMI și BIP2.

4. EXERCITII ȘI ÎNTREBĂRI

1. Codați AMI, BIP2, BIP3, PST, HDB3, CHDB3, HDB4, CHDB4 și MS43 secvența de date 1 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 și reprezentați grafic formele de undă cu impulsuri rectangulare asociate secvențelor codate. Calculați suma digitală curentă pentru fiecare secvență codată.
2. Care este valoarea maximă absolută a sumei digitale curente pentru codurile BIPn?
3. Decodați secvența codată HDB3: - 0 0 0 - + - + 0 0 + - 0 0 - + - + - 0 0 0 -.

5. ALGORITMI MATLAB

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

Funcțiile se găsesc în biblioteca `FUNCTII`:

```
>> cd d:\[cale]\functii
% COD_TERN.TXT
% coduri de linie ternare
a=[0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1];

% codare AMI
t1=ami(a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t1);ylabel('AMI');
% codare BIPOLAR2
t2=bipolar(2,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t2);ylabel('BIP2');
% codare BIPOLAR3
t3=bipolar(3,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t3);ylabel('BIP3');
% codare BIPOLAR4
t4=bipolar(4,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t4);ylabel('BIP4');
% codare HDB3
t5=hdb(3,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t5);ylabel('HDB3');
% codare HDB4
t6=hdb(4,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t6);ylabel('HDB4');
% codare HDB5
t7=hdb(5,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t7);ylabel('HDB5');
% codare CHDB3
t8=chdb(3,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t8);ylabel('CHDB3');
% codare CHDB4
t9=chdb(4,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t9);ylabel('CHDB4');
% codare CHDB5
t10=chdb(5,a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t10);ylabel('CHDB5');
% codare PST
t11=pst(a);
subplot(2,1,1),stem(a);ylabel('date');
subplot(2,1,2),stem(t11);ylabel('PST');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% analizati algoritmul de codare PST
help pst
```

```

function S=bipolar(m,x)
% codare bipolară numărul m a vectorului x
n=length(x);
for i=1:n; if x(i)>=1, x(i)=1;end;end
l=0; k=1;
for i=1:n;
    if x(i)==0;
        S(i)=0;
    end
    if x(i)==1;
        if l<=m
            S(i)=k;l=l+1;
        end
        if l>=m+1
            l=1;k=-k;S(i)=k;
        end
    end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function y=hdb(m,x)
% y=HDB(m,x)
% codare HDBm
n=length(x);
if m<2;m=2;end
for i=1:n
    if x(i)>=1/2;x(i)=1;end
    if x(i)<1/2;x(i)=0;end
end
z=zeropad(x,m);
for i=n+1:n+m
    z(i)=1;
end
k=-1;
v=-1;
s=zeros(1,m);
for i=m+1:n
    s(i)=x(i);
    for j=1:m
        s(i)=s(i)+z(i-j);
    end
end
c=ami(x(1:m));
contor(1)=1-x(1);
for i=2:n
    contor(i)=0;
    if x(i)==1; contor(i)=0;else
        contor(i)=(1-x(i-1))*contor(i-1)+1;
        contor(i)=modulo(contor(i),m+1);
    end
end
end
mem=0;
for i=m+1:n
    if x(i)==1; c(i)=-k;k=-k;v=-k;end
    if x(i)==0
        if contor(i)>=1;c(i)=0; end
        if contor(i)==0; c(i)=-v;mem=mem+1;k=-v; end
        if mem>=2;mem=0;c(i-m)=-k;c(i)=-k;k=-k;v=k;end
    end
end
y=c;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% EOF %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```