5.4. Coduri ciclice

Codurile ciclice sînt *coduri bloc în care cele n simboluri care formează un cuvînt sînt considerate ca fiind coeficienții unui polinom de grad n — 1* și anume:

$$v(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} \tag{5.116}$$

Ca și în cazul codurilor grup, simbolurile unui cuvînt de cod, respectiv coeficienții polinomului $v(x)$, pot fi reprezentați și sub formă de matrice linie

$$\mathbf{v} = [a_0\ a_1 \ldots a_{n-1}] \tag{5.117}$$

Codurile ciclice au proprietatea că dacă v este cuvînt cu sens, atunci și orice permutare ciclică a simbolurilor sale este un cuvînt cu sens:

$$[a_i\ a_{i+1} \ldots a_{n-1}\ a_0 \ldots a_{i-1}]$$

Mulțimea tuturor cuvintelor formează o **algebră**, iar *mulțimea cuvintelor cu sens formează un* **ideal.**

5.4.1. Generarea cuvintelor

Mulțimea tuturor cuvintelor, cu sens sau fără sens, este reprezentată de clase de resturi modulo un polinom $p(x)$ de grad n cu coeficienți în $GF(2)$.

Să considerăm inelul tuturor polinoamelor și în acest inel considerăm idealul format din toți multiplii polinomului $p(x)$ de grad n.

Prin mecanismul descris la codurile grup (§ 5.3) formăm clasele de resturi, prima clasă fiind idealul generat de $p(x)$:

$$\begin{array}{llll}
0 & p(x) & xp(x) \ldots 0 + & \text{multiplii lui } p(x) \\
1 & 1 + p(x) & 1 + xp(x) \ldots 1 + & \text{„ „ „} \\
x & x + p(x) & x + xp(x) \ldots x + & \text{„ „ „} \\
1+x & 1 + x + p(x) & 1 + x + xp(x) \ldots 1 + x + & \text{„ „ „} \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \\
a_0 + a_1x + \ldots + a_{n-1}x^{n-1} & \ldots\ldots & \ldots\ldots &
\end{array} \tag{5.118}$$

Clasele de resturi modulo $p(x)$ formează o algebră.

Pentru a specifica că este vorba de clase de resturi se introduce notația

$$\{a_0 + a_1x + \ldots + a_{n-1}x^{n-1}\}$$

sau notația

$$a_0 + a_1X + \ldots + a_{n-1}X^{n-1}$$

în care x a fost înlocuit cu X pentru comoditatea scrisului; cînd nu se pot crea confuzii clasele de resturi vor fi notate cu polinoame în x și nu în X.

În cele ce urmează considerăm că mulțimea tuturor cuvintelor formează o algebră generată de un polinom $p(x)$ de grad n.

Polinomul $p(x)$ poate fi ales arbitrar, dar, pentru a obține unele rezultate similare cu cele obținute la codurile grup, dorim ca produsul $u(X) \cdot v(X) = 0$ unde $u(X)$ și $v(X)$ sînt elemente în algebra modulo $p(x)$, să se exprime sub o formă asemănătoare cu produsul scalar $<u, v> = 0$.

Acest lucru se întîmplă dacă alegem:

$$p(x) = x^n + 1$$

Exemplu

$$p(X) = x^3 + 1$$

$$u(X) = a_0 + a_1X + a_2X^2$$

$$v(X) = b_0 + b_1X + b_2X^2$$

$$u(X) \cdot v(X) = a_0b_0 + (a_0b_1 + a_1b_0)\,X + (a_0b_2 + a_1b_1 + a_2b_0)\,X^2 + (a_1b_2 + a_2b_1)\,X^3 + a_2b_2X^4$$

Dat fiind că polinomul $p(x)$ se găsește în prima clasă de resturi, avem:

$$p(X) = X^3 + 1 = 0$$

de unde $X^3 = 1$, $X^4 = X$

Înlocuind obținem:

$$u(X) \cdot v(X) = a_0b_0 + a_1b_2 + a_2b_1 + (a_0b_1 + a_1b_0 + a_2b_2)\,X + (a_0b_2 + a_1b_1 + a_2b_0)\,X^2$$

Dacă $u(X) \cdot v(X) = 0$ atunci

$$\begin{aligned}
a_0b_2 + a_1b_1 + a_2b_0 &= 0 \\
a_0b_1 + a_1b_0 + a_2b_2 &= 0 \\
a_0b_0 + a_1b_2 + a_2b_1 &= 0
\end{aligned} \tag{5.119}$$

Observăm că prima ecuație este asemănătoare produsului scalar $<u, v> = 0$ în care componentele unui vector sînt scrise în ordine inversă. Mai mult, orice permutare ciclică a coeficienților dă tot un produs scalar nul.

5.4.2. Specificarea cuvintelor cu sens

Mulțimea claselor de resturi modulo $p(x) = x^n + 1$ are 2^n elemente. Din acestea alegem un număr de 2^k cărora le atribuim sens și anume atribuim sens elementelor idealului generat de un polinom $g(x)$ de grad m numit **polinom generator.**

Fiindcă elementul „0" trebuie să facă parte din ideal, rezultă că există un polinom $h(x)$, astfel încît:

$$g(X) \cdot h(X) = 0 = p(X) \tag{5.120}$$

din această relație rezultă că polinomul $g(x)$ trebuie să fie divizor al polinomului $p(x)$

$$g(x) = \frac{p(x)}{h(x)} = \frac{x^n + 1}{h(x)} \tag{5.121}$$

Ținînd seama de cele precedente, cuvintele de cod pot fi specificate în două moduri.

5.4.2.1. Codarea cuvintelor de cod ca elemente în idealul generat de $g(x)$ de grad m

În acest caz $v(x)$ trebuie să fie un multiplu a lui $g(x)$:

$$v(x) = i(x)\, g(x) \tag{5.122}$$

unde $i(x)$ este **polinomul simbolurilor de informație**

$$i(x) = a_m + a_{m+1}x + \ldots + a_{m+k-1}x^{k-1} \tag{5.123}$$

Polinomul generator este notat:

$$g(x) = g_0 + g_1 x + \ldots + g_{m-1}x^{m-1} + x^m \tag{5.124}$$

fiind, în mod obligatoriu, de grad m, $g_m = 1$.

Relația (5.122) dă o metodă de codificare, însă codul astfel obținut nu este cod sistematic.

Pentru a obține un cod sistematic se procedează după cum urmează:

Polinomul simbolurilor de control se notează cu:

$$c(x) = a_0 + a_1 x + \ldots a_{m-1}x^{m-1} \tag{5.125}$$

deci

$$v(x) = c(x) + x^m i(x)$$

Efectuăm împărțirea:

$$\frac{x^m i(x)}{g(x)} = q(x) + \frac{r(x)}{g(x)} \tag{5.126}$$

$r(x)$ fiind restul împărțirii, polinom de grad mai mic decît m.

Relația (5.126) poate fi scrisă

$$\frac{r(x)}{g(x)} + \frac{x^m i(x)}{g(x)} = q(x) \tag{5.127}$$

sau

$$r(x) + x^m i(x) = q(x) \cdot g(x) \tag{5.128}$$

Cuvîntul de cod fiind multiplu al lui $g(x)$ rezultă

$$c(x) = r(x) = rest\, \frac{x^m i(x)}{g(x)} \tag{5.129}$$

adică polinomul simbolurilor de control se obține din divizarea polinomului simbolurilor de informație multiplicat cu x^m, prin polinomul $g(x)$.

În felul acesta $v(x)$ este cuvîntul de cod al unui cod sistematic.

Ținînd seama că:

$$v(x) = q(x)\, g(x) \tag{5.130}$$

unde $q(x) = q_0 + q_1 x + \ldots + q_{k-1}x^{k-1}$ (5.131)

se poate scrie:

$$v(x) = q_0 g(x) + q_1 x g(x) + \ldots + q_{k-1}x^{k-1}g(x) \tag{5.132}$$

cu alte cuvinte $v(x)$ se găsește în spațiul linie al matricei:

$$\mathbf{G} = \begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{k-1}g(X) \end{bmatrix} \tag{5.133}$$

Reamintim că spațiul linie al unei matrici este format din mulțimea combinațiilor liniare ale vectorilor linie ai matricei.

Ori **G**, conform relației (5.124), se poate scrie:

$$\mathbf{G} = \begin{bmatrix} g(X) \\ Xg(X) \\ \vdots \\ X^{k-1}g(X) \end{bmatrix} = \begin{bmatrix} g_0 & g_1 \ldots g_m & 0 \ldots 0 \\ 0 & g_0 \ldots g_m & 0 \ldots 0 \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ 0 \ldots 0 & g_0\ g_1 & \ldots g_m \end{bmatrix} \tag{5.134}$$

unde elementele liniilor sînt coeficienții unui polinom de grad $n - 1$ ($v(x)$ este de grad $n - 1$) iar coeficienții puterilor lui x absente au fost înlocuiți cu zero.

Matricea **G** se numește **matrice generatoare** și are k linii și n coloane; cu ea se poate face codificarea conform relației (5.89):

$$\mathbf{v} = \mathbf{i}\ \mathbf{G}$$

care este echivalentă cu relația (5.122) în sensul că ambele ne dau aceleași simboluri de control.

5.4.2.2. Codarea cuvintelor de cod ca elemente în spațiul nul al idealului generat de $h(x)$ de grad k

Pornind de la relația (5.130), multiplicînd cu:

$$h(x) = h_0 + h_1 x + \ldots h_k x^k \tag{5.135}$$

obținem:

$$v(x)\, h(x) = q(x)\, g(x)\, h(x)$$

respectiv, ținînd seama că polinoamele respective sînt clase de resturi și că avem relația (5.120), se poate scrie:

$$v(X)\, h(X) = q(X)\, g(X)\, h(X)$$

adică

$$v(X)\, h(X) = 0 \tag{5.136}$$

relație care spune că $v(x)$ se găsește în spațiul nul al idealului generat de $h(x)$.

Produsul $v(X)\, h(X)$ prin generalizarea relațiilor (5.119) poate fi scris ca produs scalar cu componentele vectorului $h(x)$ scrise în ordine inversă:

$$\begin{aligned}
&(a_0\ a_1 \dots a_{n-1})\ (0 \dots 0\ \ h_k\ \ h_{k-1} \dots h_0) = 0\\
&(a_0\ a_1 \dots a_{n-1})\ (0 \dots h_k\ \ h_{k-1}\ \dots\ h_0\ 0\ ) = 0\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&(a_0\ a_1 \dots a_{n-1})\ (h_k\ \ h_{k-1} \dots h_0\ 0 \dots 0) = 0
\end{aligned} \tag{5.137}$$

restul ecuațiilor nu prezintă interes fiindcă din cele n ecuații date de permutările ciclice numai $n - k = m$ sînt independente.

Setul de ecuații (5.137) care permit să se determine simbolurile de control în funcție de cele de informație pot fi scrise:

$$\begin{bmatrix} 0 \dots 0\ \ h_k\ \ h_{k-1} \dots h_0 \\ 0 \dots h_k \dots\dots h_0\ 0 \\ \dots\dots\dots\dots \\ \dots\dots\dots\dots \\ h_k \dots h_0\ 0 \dots\dots 0 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \mathbf{0} \tag{5.138}$$

sau notînd:

$$\mathbf{H} = \begin{bmatrix} 0 \dots 0\ \ h_k \dots\ \ h_0 \\ 0 \dots 0\ h_k \dots\ h_0\ 0 \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ h_k \dots\ h_0\ 0 \dots 0\ 0 \end{bmatrix} \tag{5.139}$$

avem:

$$\mathbf{H}\mathbf{v}^T = \mathbf{0} \tag{5.140}$$

deci matricea $\mathbf{H}$ dată de relația (5.139) este matricea de control, relația (5.140) fiind echivalentă cu relația (5.136).

Ținînd seama de relația (5.120), de (5.134) și de (5.139) se poate arăta că

$$\mathbf{G}\mathbf{H}^T = \mathbf{H}\mathbf{G}^T = \mathbf{0} \tag{5.141}$$

5.4.3. Decodarea

Problema decodării constă în *găsirea unei corespondențe între cuvîntul eronat recepționat* $v'(x)$ *și cuvîntul* $\varepsilon(x)$ *care reprezintă erorile introduse de canal.*

$$v'(x) = v(x) + \varepsilon(x) \tag{5.142}$$

Dacă recepționînd $v'(x)$ se poate calcula $\varepsilon(x)$ atunci corecția se realizează prin însumarea modulo doi:

$$v(x) = v'(x) + \varepsilon(x) \tag{5.143}$$

După cum s-a văzut în § 5.3.8 relația (5.106) corectorul se obține prin însumarea simbolurilor de control recepționate $c'(x)$ cu simbolurile de control obținute prin codificarea simbolurilor de informație recepționate

$$c''(x) = rest\ \frac{x^m i'(x)}{g(x)} \tag{5.144}$$

respectiv:

$$z(x) = c'(x) + c''(x) \tag{5.145}$$

sau

$$z(x) = c'(x) + rest\ \frac{x^m i'(x)}{g(x)} \tag{5.146}$$

Polinomul $c'(x)$ fiind de grad mai mic decît m se poate scrie:

$$z(x) = rest\ \frac{c'(x) + x^m i(x)}{g(x)} \tag{5.147}$$

sau

$$z(x) = rest\ \frac{v'(x)}{g(x)} \tag{5.148}$$

sau ținînd seama de relația (5.142):

$$z(x) = rest\ \frac{\varepsilon(x)}{g(x)} \tag{5.149}$$

Polinomul $z(x)$ are gradul cel mult $m - 1$, deci numărul acestor corectori este 2^m (sînt m termeni). Rezultă că din mulțimea configurațiilor posibile de erori, respectiv polinoame $\varepsilon(x)$ în număr de 2^n numai cele care pot fi puse în corespondență biunivocă cu corectorii $z(x)$ adică un număr de 2^m configurații de erori pot fi corectate.

Relația (5.149) este echivalentă cu relația:

$$\mathbf{z} = \mathbf{H}\boldsymbol{\varepsilon}^T \tag{5.150}$$

● **Decodarea pe baza tabloului claselor de resturi.** Prima clasă este formată de idealul generat de $g(x)$. Prima coloană este formată din polinoame pe care le considerăm cuvinte eroare. La urmă rezervăm o coloană pentru corectori:

	Cuvinte recepționate		
	Cuvinte eroare		corectori $z(x)$
Cuvinte cu sens	0	$v_1(x)\ v_2(x) \dots\dots\dots\dots\dots\dots v_s(x)$	
Cuvinte fără sens	$\varepsilon_1(x)$	$\varepsilon_1 + v_1\ \varepsilon_1 + v_2 \dots\dots\dots\dots\dots \varepsilon_1 + v_s$	$z_1 = \text{rest}\ \frac{\varepsilon_1(x)}{g(x)}$
	$\varepsilon_2(x)$	$\varepsilon_2 + v_1\ \varepsilon_2 + v_2 \dots\dots\dots\dots\dots \varepsilon_2 + v_s$	$z_2 = \text{rest}\ \frac{\varepsilon_2(x)}{g(x)}$
	⋮	⋮ ⋮ ⋮	⋮

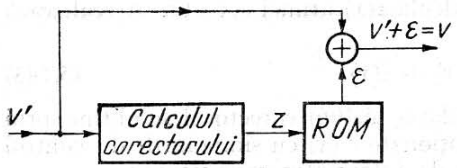

Fig. 5.10. Procedeu de codare bazat pe clase de resturi.

Se observă că pentru toate cuvintele dintr-o clasă de resturi (linie) corectorul este același. Astfel, pentru clasa la care primul element este ε_j, un cuvînt oarecare din această clasă are corectorul:

$$\mathbf{z}_j = \mathbf{H}(\boldsymbol{\varepsilon}_j + \mathbf{v}_k)^T = \mathbf{H}\boldsymbol{\varepsilon}_j^T + \mathbf{H}\mathbf{v}_k^T = \mathbf{H}\boldsymbol{\varepsilon}_j^T \quad k = \overline{1, s} \tag{5.151}$$

Din acest fapt rezultă *procedeul de decodare* și anume:

— la fiecare cuvînt recepționat $\mathbf{v}'$ se calculează corectorul $\mathbf{z} = \mathbf{H}\mathbf{v}'^T$;

— din tabloul claselor de resturi se vede care este cuvîntul eroare $\boldsymbol{\varepsilon}$ corespunzător corectorului;

— se adună $\boldsymbol{\varepsilon}$ cu $\mathbf{v}'$ și se obține $\mathbf{v}$.

Acest procedeu de decodare este valabil numai dacă corectorului calculat nu-i corespunde decît un singur cuvînt eroare. În caz contrar se poate face detecție de erori însă nu se poate face corecție.

În fig. 5.10 este arătat schematic principiul acestei decodări. În memoria ROM care este adresată de către corectorul z se introduc toate cuvintele eroare ε în corespondență biunivocă cu corectorii z.

Pînă în prezent nu s-a specificat ce condiții trebuie să îndeplinească polinomul generator $g(x)$ astfel încît codul să poată corecta un număr impus de erori. Această problemă va fi tratată mai tîrziu.

5.4.4. Realizarea codării și a decodării pentru detecția erorilor prin circuite de multiplicare sau de divizare

După cum s-a văzut în paragraful 5.4.2 codarea poate fi realizată în două moduri:

1. *Prin multiplicare:*

$$v(x) = i(x)\ g(x) \tag{5.152}$$

și în acest caz se obține un cod nesistematic. Demodularea se face prin divizare:

$$z(x) = rest\ \frac{v'(x)}{g(x)} \tag{5.153}$$

2. *Prin divizare:*

$$v(x) = rest\ \frac{x^m i(x)}{g(x)} + x^m i(x) \tag{5.154}$$

în acest caz obținîndu-se un cod sistematic.

Decodarea se face tot prin divizare ca și în cazul precedent (relația (5.153)).

Date fiind avantajele codurilor sistematice, în cele ce urmează se va trata cazul al doilea.

5.4.4.1. Codarea prin circuite de divizare

Fie circuitul secvențial din fig. 5.11 care folosește celule binare notate cu c, circuite de multiplicare cu constantele $g_i \in GF(2)$ și sumatori modulo doi.

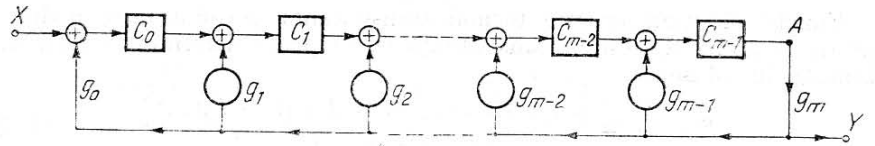

Fig. 5.11. Circuit de divizare.

Pentru analiza circuitului vom folosi un **operator de întîrziere** notat cu D care reprezintă întîrzierea de un tact pe care o introduce o celulă binară.

Dacă în registru se introduc coeficienții în ordinea descrescătoare a indicilor, respectiv:

$$a_n, a_{n-1}, a_{n-2}, \ldots, a_0$$

care corespund polinomului:

$$j(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0$$

atunci secvența aplicată la intrare utilizînd operatorul D poate fi pusă sub forma de transformată Z:

$$a_n + a_{n-1}D + a_{n-2}D^2 + \ldots + a_0 D^n \tag{5.155}$$

cu interpretarea că simbolul a_0 ajunge în prima celulă de intrare după n tacte sau:

$$D^n(a_n D^{-n} + a_{n-1}D^{-(n-1)} + \ldots + a_1 D^{-1} + a_0 I) \tag{5.156}$$

unde:

$$D^0 = I \tag{5.157}$$

reprezintă **operatorul identitate.**

Din cele precedente rezultă că unui polinom în x îi corespunde același polinom în D^{-1}, întîrziat.

Pentru a pune în evidență operația pe care o realizează circuitul, se definește funcția de transfer a circuitului notată cu T ca fiind raportul între secvența de ieșire Y și secvența de intrare X:

$$T = \frac{Y}{X} \tag{5.158}$$

Determinarea funcției de transfer a circuitului se face foarte simplu dacă considerăm graful echivalent al circuitului arătat în fig. (5.12).

În construirea grafului s-a ținut seama de faptul că unei celule binare îi corespunde o ramură de transmitanță D, unui multiplicator cu constanta g_i o ramură de transmitanță g_i, iar unui sumator modulo-doi — un nod, în care însumarea se face modulo doi.

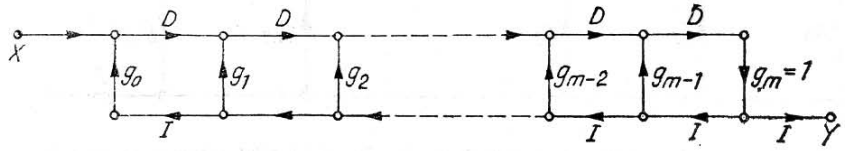

Fig. 5.12. Graful echivalent al circuitului din fig. 5.11.

Funcția de transfer este tocmai transmitanța grafului între nodul de intrare și nodul de ieșire, transmitanță ce se poate determina cu ajutorul formulei lui Mason:

$$T = \frac{Y}{X} = \frac{[(C_1 + C_2 + \ldots)\,(I + L_1)\,(I + L_2)\ldots]^*}{[(I + L_1)\,(I + L_2)\ldots]^*} \tag{5.159}$$

unde:

X este secvența de intrare exprimată în funcție de operatorul D;
Y — secvența de ieșire;
C_i — transmitanța căii i;
L_j — transmitanța buclei j;
I — operatorul identitate.

Asteriscurile indică faptul că din expresiile respective trebuie să fie eliminați termenii ce conțin produse între bucle și căi adiacente și între bucle adiacente. Sumele se fac modulo 2.

Dacă ne referim la graful din fig. 5.12 avem:

$$T = \frac{Y}{X} = \frac{C_{xy}}{I + L_1 + L_2 \ldots + L_m} = \frac{D^m}{I + g_{m-1}D + g_{m-2}D^2 \ldots g_0 D^m} \tag{5.160}$$

unde:

C_{xy} este unica cale între intrare și ieșire;
L_j — transmitanța buclei j

sau

$$T = \frac{Y}{X} = \frac{I}{D^{-m} + g_{m-1}D^{-(m-1)} + \ldots + g_0} \tag{5.161}$$

de unde:

$$Y = \frac{X}{g(D^{-1})} \tag{5.162}$$

relație din care rezultă că circuitul face operația de divizare:

$$y(x) = \frac{j(x)}{g(x)} \tag{5.163}$$

Dacă împărțirea se face cu rest, restul rămîne înmagazinat în celulele registrului.

Dacă secvența de intrare $X' = i(x)$ se aplică la ieșirea registrului (punctul A în fig. 5.11 și 5.13)

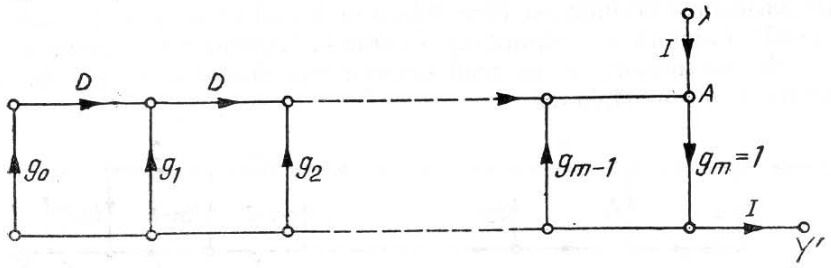

Fig. 5.13. Graful echivalent al circuitului din fig. 5.11 cînd intrarea se face în punctul A.

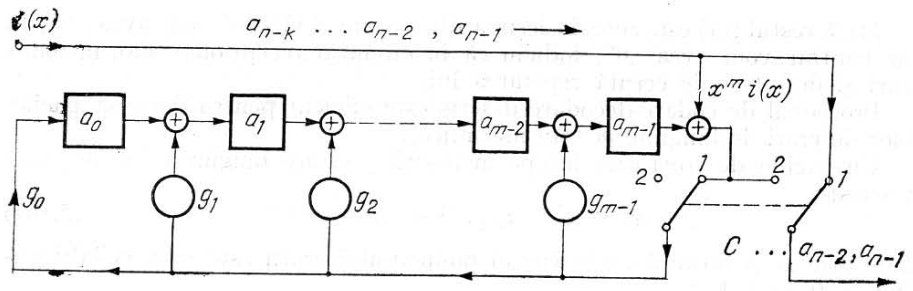

Fig 5.14. Codarea prin circuite de divizare.

atunci funcția de transfer devine:

$$T = \frac{Y'}{X'} = \frac{I}{I + g_{m-1}D + \ldots + g_0 D^m} = \frac{D^{-m}}{g(D^{-1})} \tag{5.164}$$

sau

$$Y' = \frac{D^{-m}X'}{g(D^{-1})} \tag{5.165}$$

respectiv

$$y(x) = \frac{x^m i(x)}{g(x)} \tag{5.166}$$

Această relație arată că circuitul din fig. 5.11 prin introducerea simbolurilor de informație în punctul A poate servi la codificare, în conformitate cu relația (5.154).

Circuitul de codificare este arătat în fig. 5.14.

Inițial cheia C se găsește în poz. 1 și la intrare se introduc coeficienții lui $i(x)$ și fiindcă intrarea se face după ultima celulă, în registru se introduce polinomul $x^m i(x)$. Cîtul diviziunii cu $g(x)$ nu interesează, ci numai restul $c(x)$ care este înmagazinat în registru după k tacte.

După terminarea introducerii simbolurilor de informație cheia C este trecută pe poz. 2 și conținutul $c(x)$ al registrului este evacuat $(a_0 \ldots a_{m-2}a_{m-1})$ completîndu-se astfel la ieșire întregul cuvînt de cod $(a_0 \ldots a_{m-1}, a_m \ldots a_{n-1})$.

5.4.4.2. Decodarea prin circuit de divizare

Decodarea constă în calculul corectorului $z(x) = rest\,\frac{v'(x)}{g(x)}$ realizat cu circuitul din fig. 5.15.

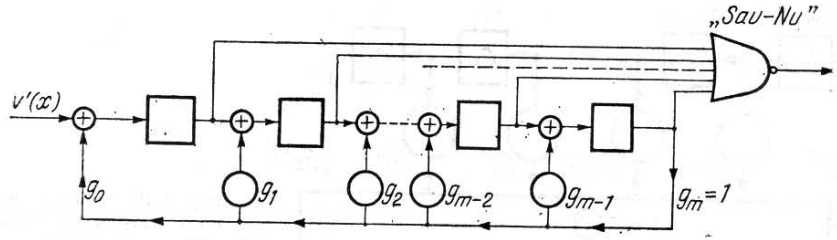

Fig. 5.15. Decodarea prin circuite de divizare.

Dacă restul $z(x)$ este zero, la ieşirea din poarta SAU-NU vom avea „1"; în caz contrar vom avea „0", indiciu că în cuvîntul recepţionat s-au introdus erori şi deci trebuie cerută repetarea lui.

Procedeul de codare-decodare descris este eficient pentru detecţia pachetelor de erori de lungime m (sau mai mică).

Un pachet de erori care începe în poziţia j şi are lungimea $l = m$ poate fi scris:

$$e(x) = x^j + \varepsilon_{j+1}x^{j+1} + \ldots + x^{j+m-1} \tag{5.167}$$

unde primul şi ultimul coeficient al polinomului erorii este unu, ceilalţi putînd fi 0 sau 1.

Pachetul de erori mai poate fi scris:

$$e(x) + x^j [1 + \varepsilon_{j+1}x + \ldots + x^{m-1}] = x^j e_0(x) \tag{5.168}$$

Dacă $$v'(x) = v(x) + e(x) \tag{5.169}$$

atunci $$z(x) = rest\frac{e(x)}{g(x)} = rest\frac{x^j e_0(x)}{g(x)} \tag{5.170}$$

Deoarece $e_0(x)$ este de grad $m - 1$, nu este divizibil cu $g(x)$ iar $g(x)$, fiind polinom ireductibil cum se va vedea ulterior divizor al lui $x^n + 1$, restul $\frac{x^j e_0(x)}{g(x)}$ va fi diferit de zero, deci codul este capabil să detecteze un pachet de erori de lungime m în orice poziţie intervin acestea.